Lecture 5
Block Diagrams
Modes of Operation of Block Ciphers
Modes of Operation of Block Ciphers
Block vs. stream ciphers

Every block of ciphertext is a function of only one corresponding block of plaintext.

Every block of ciphertext is a function of the current block of plaintext and the current internal state of the cipher.
Typical stream cipher

Sender

key

initialization vector (seed)

Pseudorandom Key Generator

k_i

keystream

m_i

plaintext

Receiver

key

initialization vector (seed)

Pseudorandom Key Generator

k_i

keystream

c_i

ciphertext

m_i

plaintext
# Standard modes of operation of block ciphers

<table>
<thead>
<tr>
<th>Block cipher</th>
<th>Block cipher turned into a stream ciphers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECB mode</td>
<td>Counter mode</td>
</tr>
<tr>
<td></td>
<td>CFB mode</td>
</tr>
<tr>
<td></td>
<td>CBC mode</td>
</tr>
</tbody>
</table>
ECB (Electronic CodeBook) mode
Electronic CodeBook Mode – ECB Encryption

\[ C_i = E_K(M_i) \quad \text{for } i=1..N \]
Electronic CodeBook Mode – ECB
Decryption

\[ M_i = D_K(C_i) \quad \text{for } i=1..N \]
Electronic CodeBook Mode – ECB
(simplified block diagram)

\[ C_i = E_K(M_i) \]
\[ M_i = D_K(C_i) \]
Electronic CodeBook Mode – ECB
(combined block diagram)

bdi = $M_i$ for Encryption
$C_i$ for Decryption

bdo = $C_i$ for Encryption
$M_i$ for Decryption
Counter Mode
Counter Mode - CTR Encryption

\[ c_i = m_i \oplus k_i \]
\[ k_i = E_K(IV+i-1) \quad \text{for } i=1..N \]
Counter Mode - CTR
Decryption

\[ m_i = c_i \oplus k_i \]

\[ k_i = E_K(IV+i-1) \quad \text{for } i=1..N \]
Counter Mode – CTR
(simplified block diagram)

\[ IS_1 = IV \]
\[ c_i = E_K(IS_i) \oplus m_i \]
\[ IS_{i+1} = IS_i + 1 \]

\[ IS_1 = IV \]
\[ m_i = E_K(IS_i) \oplus c_i \]
\[ IS_{i+1} = IS_i + 1 \]
Counter Mode – CTR
(combined block diagram)

bdi = m_i for Encryption

bdo = c_i for Encryption

bdo = c_i for Decryption

m_i for Decryption
CFB (Cipher FeedBack) Mode
Cipher Feedback Mode - CFB
Encryption

\[ c_i = m_i \oplus k_i \]
\[ k_i = E_K(c_{i-1}) \quad \text{for } i=1..N, \text{ and } c_0 = IV \]
Cipher Feedback Mode - CFB
Decryption

\[ m_i = c_i \oplus k_i \]
\[ k_i = E_K(c_{i-1}) \quad \text{for } i=1..N, \text{ and } c_0 = IV \]
Cipher Feedback Mode – CFB
(simplified block diagram)

\[
\begin{align*}
\text{IS}_1 &= \text{IV} \\
c_i &= E_K(\text{IS}_i) \oplus m_i \\
\text{IS}_{i+1} &= c_i
\end{align*}
\]
Cipher Feedback Mode – CFB
(combined block diagram)

IV

register

IN

E

OUT

K

bdi = m_i for Encryption
c_i for Decryption

bdo = c_i for Encryption
m_i for Decryption
CBC (Cipher Block Chaining) Mode
Cipher Block Chaining Mode - CBC Encryption

\[ c_i = E_K(m_i \oplus c_{i-1}) \quad \text{for } i=1..N \quad c_0=\text{IV} \]
Cipher Block Chaining Mode - CBC Decryption

\[ m_i = D_K(c_i) \oplus c_{i-1} \text{ for } i=1..N \quad c_0=IV \]
Cipher Block Chaining Mode – CBC
(simplified block diagram)

\[ \text{IS}_1 = \text{IV} \]
\[ c_i = E_K(\text{IS}_i \oplus m_i) \]
\[ \text{IS}_{i+1} = c_i \]

\[ \text{IS}_1 = \text{IV} \]
\[ m_i = D_K(c_i) \oplus \text{IS}_i \]
\[ \text{IS}_{i+1} = c_i \]
Cipher Block Chaining Mode – CBC
(combined block diagram)
Advanced Encryption Standard (AES)
Pseudocode
AES Encryption

Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]

    state = in

    AddRoundKey(state, w[0, Nb-1])  // See Sec. 5.1.4

    for round = 1 step 1 to Nr-1
        SubBytes(state)  // See Sec. 5.1.1
        ShiftRows(state)  // See Sec. 5.1.2
        MixColumns(state)  // See Sec. 5.1.3
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

    out = state
end
AES Decryption

EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
begin
  byte state[4,Nb]

  state = in

  AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])

  for round = Nr-1 step -1 downto 1
    InvSubBytes(state)
    InvShiftRows(state)
    InvMixColumns(state)
    AddRoundKey(state, dw[round*Nb, (round+1)*Nb-1])
  end for

  InvSubBytes(state)
  InvShiftRows(state)
  AddRoundKey(state, dw[0, Nb-1])

  out = state
end
AES: Symbols, Block Diagrams, Interfaces
AES_Enc

- Encryption Only
- Key scheduling done as a part of initialization
Symbol

AES_Enc

start

init

rst

din

key

done

done_init

ready

dout

128

128

128
Note: Bold line represents a 128-bit bus unless specified otherwise.
Block Diagram – Round

```
Subject
  input
  output

ShiftRows
  input
  output

MixColumns
  input
  output

dout_fdb  rkey  dout

Note: Bold line represents a 128–bit bus unless specified otherwise.
```
Note: All buses are 32-bit wide unless specified otherwise
AES_Enc: Interface with the Division into the Datapath and Controller

Note: Bold line represents a 128-bit bus unless specified otherwise
AES_Enc_KOF

- Encryption Only
- Key scheduling done On the Fly
Symbol

128

128

din

key

128

dout

128

128

start
done

rst

ready

AES_Enc_KOF
Block Diagram – AES_Enc_KOF

Note: Bold line represents a 128-bit bus unless specified otherwise.
Block Diagram – Round

```
<table>
<thead>
<tr>
<th></th>
<th>din</th>
<th>SubBytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>input</td>
<td>output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>input</th>
<th>ShiftRows</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>input</td>
<td>output</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>input</th>
<th>MixColumns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>input</td>
<td>output</td>
</tr>
</tbody>
</table>

Note: Bold line represents a 128-bit bus unless specified otherwise
Note: All buses are 32-bit wide unless specified otherwise.
AES_Enc_KOF: Interface with the Division into the Datapath and Controller

Note: Bold line represents a 128-bit bus unless specified otherwise
AES_EncDec

- Encryption and Decryption
- Key scheduling done as a part of initialization
Block Diagram – AES_EncDec

Note: Bold line represents a 128-bit bus unless specified otherwise.
Block Diagram – Round

Note: Bold line represents a 128-bit bus unless specified otherwise.
Block Diagram – InvRound

**InvSubBytes**

**InvShiftRows**

**InvMixColumns**

Note: Bold line represents a 128-bit bus unless specified otherwise.
Note: All buses are 32-bit wide unless specified otherwise.
AES_EncDec: Interface with the Division into the Datapath and Controller

EncDec Datapath

EncDec Control

Note: Bold line represents a 128–bit bus unless specified otherwise
Example of a Hierarchical Block Diagram

JH hash function
Top Level
R8/R6

R8 : y = 1024
R6 : y = 256

y is the bit size of r

PERMUTE

rp
Example of a Hierarchical Block Diagram

BLAKE hash function
Top Level

BLAKE-32 : b=512, h=256
BLAKE-64 : b=1024, h=512
Permute8

Note:
hi and low denotes top and bottom half of the permutation table

\[ m = m[0..15] \quad c = c[0..15] \]

\[ cm_{2i} = m_{2i} \oplus c_{2i+1} \]

\[ cm_{2i+1} = m_{2i+1} \oplus c_{2i} \]

BLAKE−32 : b=512, w=32
BLAKE−64 : b=1024, w=64
BLAKE−32 : b=512, w=32
BLAKE−64 : b=1024, w=64
G_mod

BLAKE−32 : w=32
BLAKE−64 : w=64

A → CM_{2i} → CM_{2i+1} → A'

B → CM_{2i} → CM_{2i+1} → B'

C → CM_{2i} → CM_{2i+1} → C'

D → CM_{2i} → CM_{2i+1} → D'
Interface of CipherCore Datapath
## Input Ports

<table>
<thead>
<tr>
<th>Port</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>npub (nonce, IV)</td>
<td>out</td>
<td>NPUB_SIZE [Optional] Public message number (Npub). This port is inactive if PLAINTEXT_MODE = 1 or 2.</td>
</tr>
<tr>
<td>nsec (only few candidates)</td>
<td>out</td>
<td>NSEC_SIZE [Optional] Secret message number (Nsec). This port is inactive if NSEC_ENABLE = 0.</td>
</tr>
<tr>
<td>key</td>
<td>out</td>
<td>KEY_SIZE [Optional] Key data. Note: Port is disabled if RDKEY_ENABLE = 1.</td>
</tr>
<tr>
<td>rdkey (we will not use it)</td>
<td>out</td>
<td>RDKEY_SIZE [Optional] Round key data. Note: Port is disabled if RDKEY_ENABLE = 0.</td>
</tr>
<tr>
<td>bdi (AD, M, C)</td>
<td>out</td>
<td>DBLK_SIZE Input block data</td>
</tr>
<tr>
<td>exp_tag</td>
<td>out</td>
<td>TAG_SIZE Expected tag data. This output is valid for authenticated decryption operation.</td>
</tr>
<tr>
<td>len_a</td>
<td>out</td>
<td>CTR_AD_SIZE [SEGMENT INFO] Length of associated data in bytes (used in some algorithms)</td>
</tr>
<tr>
<td>len_d</td>
<td>out</td>
<td>CTR_D_SIZE [SEGMENT INFO] Length of data in bytes (used in some algorithms)</td>
</tr>
</tbody>
</table>

[SEGMENT INFO]. Auxiliary signal that remains valid for the current segment. The value changes when a new segment is received via the PDI data bus. For length information, the values are reset for every new block of data.
Timeline
Project Timeline: Draft Block Diagrams

Thursday 10/15, 12 noon:
First draft of block diagrams (Blackboard)

Thursday 10/15, 1:00-4:30pm

Friday 10/16, 1:00-8:00pm
Discussion of draft block diagrams
(30 minutes per person,
60 minutes per group,
electronic sign-up using Doodle)
Project Timeline: Revised Block Diagrams

Thursday 10/22, 12 noon
Revised block diagrams due (Blackboard)

Thursday 10/22, 1:00-4:30pm
Friday 10/23, 1:00-8:00pm
Discussion of revised block diagrams
(30 minutes per person,
60 minutes per group,
electronic sign-up using Doodle)