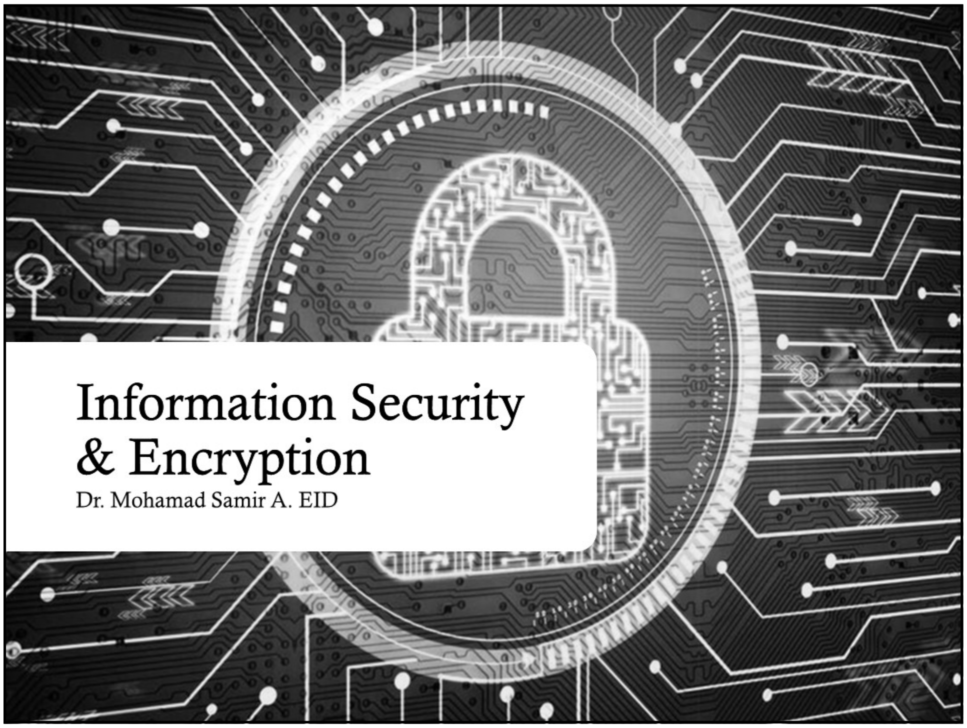
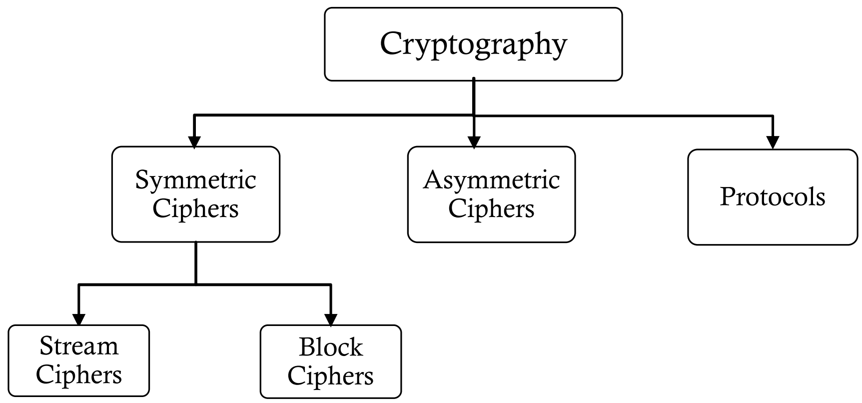


1

Quiz 1

2

Main Areas of Cryptography

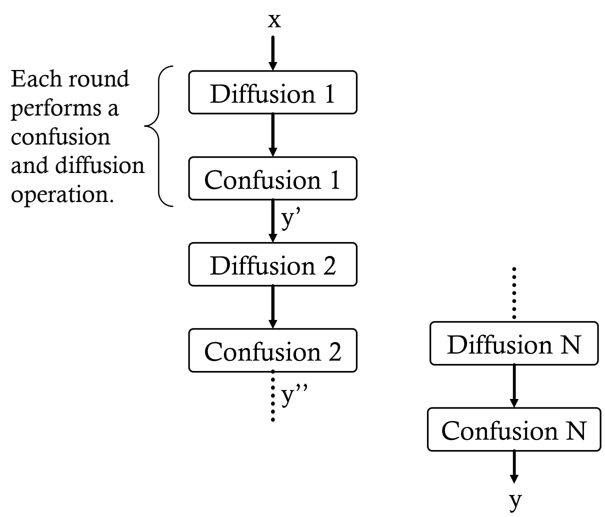


Strong Block Encryption

In 1945, Claude Shannon defined two basic operations to achieve strong encryption:

- **Confusion:** an encryption operation where the relationship between key and ciphertext is hidden.
- **Diffusion:** an encryption operation where the influence of one plaintext bit is spread over many ciphertext bits.

Strong Block Encryption



"Principle of an N-round product cipher"

 5

5

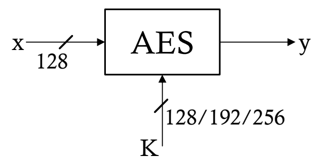
Today

- Motivation.
- Intro to Finite fields (needed for understanding AES).
- Intro to AES.

 6

6

Motivation



Found by every web browser, in banking machines, WiFi routers, Bitlocker, etc ..

How does it work?

All internal operations of AES are based on **Finite Fields**.

Finite Fields (Galois Fields)

- Agenda:
  - Introduction to Finite Fields.
  - Prime Fields.
  - Extension Fields.

Intro to Finite Fields

What's a **Field**?

Abstract (modern) algebra consists of three basic elements:
- Group
- Ring
- Field

Group {G,+,−}: a set of elements, such that the following axioms are obeyed:

Note: the generic operator ∘ denotes either + or −

A1. Closure:
If a and b belong to G, then a ∘ b is also in G.

A2. Associativity:
a ∘ (b ∘ c) = (a ∘ b) ∘ c for all a, b, c in G

A3. Identity element:
There is an element 0 in G such that a ∘ 0 = 0 ∘ a = a for all a in G

A4. Inverse element:
For each a in G there is an element -a in G such that a + (-a) = (-a) + a = 0

A5. Commutativity:
a ∘ b = b ∘ a for all a, b in G

But we're interested in more than just +, −

 9

9

Intro to Finite Fields

Ring {R,+,−,×}: a set of elements such that the following axioms are obeyed:

A1~A5.

M1. Closure under multiplication:
If a and b belong to R, then ab is also in R

M2. Associativity of multiplication:
a(bc) = (ab)c for all a, b, c in R

M3. Distributive laws:
a(b + c) = ab + ac for all a, b, c in R
(a + b)c = ac + bc for all a, b, c in R

M4. Commutativity of multiplication:
ab = ba for all a, b in R

M5. Multiplicative identity:
There is an element 1 in R such that a1 = 1a = a for all a in R

M6. No zero divisors:
If a, b in R and ab = 0, then either a = 0 or b = 0

Still, we're interested in more than just +, −,×

 10

10

Intro to Finite Fields

Field {F,+, −,×,()$^{-1}$}: a set of elements, such that the following axioms are obeyed:

A1~A5.

M1~M6.

M7. Multiplicative inverse:
For each a in F, except 0,
there is an element a^{-1} in F such that $aa^{-1}=(a^{-1})a=1$.

Simply, it's a set of numbers which we can add, subtract, multiply, and invert, that obey A1~A5 & M1~M7.

Example: Which of the following are Fields? $\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$

Intro to Finite Fields

In crypto, we almost always need <u>finite</u> sets.

m: positive integer

Theorem: A finite field only exists if it has p^m elements.

p: prime integer

Order or **cardinality** of the field: number of elements in GF.

Examples:

1) There's a finite field with 11 elements. GF(11)
2) There's a finite field with 81 elements. GF(81)= GF(3^4)
3) There's a finite field with 256 elements. GF(256) = GF(2^8) ← The Galois field specified in the AES standard.
4) Is the field with 12 elements a finite field?

Types of Finite Fields

$GF(p^m)$

$m=1$ → $GF(p)$ *i.e., prime fields*

$m>1$ → $GF(p^m)$ *i.e., extension fields*

Note: $GF(2^m)$ is of special importance in crypto.

Prime Field Arithmetic

The elements of a prime field GF(p) are the integers {0, 1, . . ., p−1}

a) Add, subtract, multiply:

$a \circ b \equiv c \bmod p$

Note:
the generic operator ∘ here denotes either +, − , or ×

b) Inversion:

$a \in GF(p)$; the inverse a^{-1} must satisfy $a \cdot a^{-1} \equiv 1 \bmod p$

a^{-1} can be computed using the Extended Euclidian Algorithm.

Extension Field GF(2^m) Arithmetic

Used in AES.

The elements of GF(2^m) are polynomials.

$a_{m-1}x^{m-1} + \ldots + a_1x + a_0 = A(x) \in GF(2^m)$

Coefficients $a_i \in GF(2) = \{0, 1\}$

Example:

$GF(2^3) = GF(8)$

$A(x) = a_2x^2 + a_1x + a_0 = (a_2, a_1, a_0)$

$GF(2^3) = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$

Extension Field GF(2^m) Arithmetic

a) Add and subtract in GF(2^m):

$C(x) = A(x) \circ B(x) = \sum_{i=0}^{m-1} c_i x^i, ci \equiv ai + bi \bmod 2$

Note:
the generic operator ∘ here denotes either +, −

Example: In GF(2^3), $A(x) = x^2 + x + 1$, $B(x) = x^2 + 1$
Compute $A(x) + B(x)$

$GF(2^3) = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$

$$A(x) + B(x) = (1+1)x^2 + x + (1+1)$$
$$= 0x^2 + x + 0$$
$$= x = A(x) - B(x)$$

Note:
Addition and subtraction in GF(2^m) are the same operations.

Extension Field GF(2^m) Arithmetic

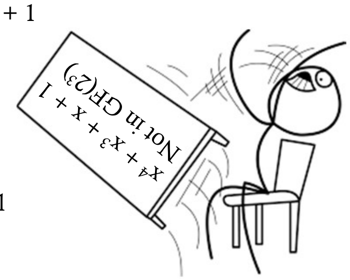
b) Multiplication in GF(2^m):

Example: In GF(2^3), $A(x) = x^2 + x + 1$, $B(x) = x^2 + 1$
Compute $A(x) \times B(x)$

GF(2^3)= { 0, 1, x, x+1, x^2, x^2+1,x^2+x, x^2+x+1}

$$A(x) \times B(x) = (x^2 + x + 1)(x^2 + 1)$$
$$= x^4 + x^3 + x^2 + x^2 + x + 1$$
$$= x^4 + x^3 + (1+1)x^2 + x + 1$$
$$= x^4 + x^3 + x + 1$$

Wait a second . .



So, call this result $x^4 + x^3 + x + 1 = C'(x)$

Solution: Reduce C'(x) modulo a polynomial that behaves like a prime.
i.e., a polynomial that cannot be factored.
i.e., an irreducible polynomial.

In the next example..

 17

17

Extension Field GF(2^m) Arithmetic

b) Multiplication in GF(2^m):

$C(x) \equiv A(x) \times B(x) \bmod P(x)$, where P(x) is an irreducible polynomial.

Example: Given the irreducible polynomial for GF(2^3) $P(x) = x^3 + x + 1$
$A(x) = x^2 + x + 1$, $B(x) = x^2 + 1$
Compute $A(x) \times B(x) \bmod P(x)$

$$A(x) \times B(x) = x^4 + x^3 + x + 1 = C'(x)$$

$$\begin{array}{r|l} & x + 1 \\ \hline x^3 + x + 1 & x^4 + x^3 + x + 1 \\ & \underline{x^4 + x^2 + x} \\ & x^3 + x^2 + 1 \\ & \underline{x^3 + x + 1} \\ & x^2 + x \quad \equiv A(x) \times B(x) \bmod P(x) \equiv C(x) \end{array}$$

 18

18

Extension Field $GF(2^m)$ Arithmetic

Where did P(x) come from in the previous example?

For every finite field $GF(2^m)$, there are several irreducible polynomials.

So, for a given finite field (e.g., $GF(2^3)$), the computation result depends on P(x).

So, multiplication can't be done unless the irreducible polynomial is specified.

It must be..

The AES standard specifies the irreducible polynomial:
$P(x) = x^8 + x^4 + x^3 + x + 1$

What about $()^{-1}$?

How to test whether a P(x) is reducible or not?
https://www.youtube.com/watch?v=pHQ73N3n-ZU

Extension Field $GF(2^m)$ Arithmetic

c) Inversion in $GF(2^m)$:

The inverse $A^{-1}(x)$ of an element $A(x) \in GF(2^m)$ must satisfy:
$A(x) \times A^{-1}(x) \equiv 1 \bmod P(x)$

Extended Euclidian Algorithm.

The Advanced Encryption Standard (AES)

21

AES

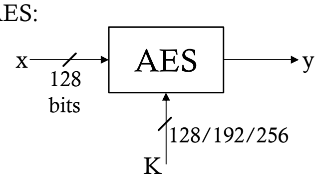
- Agenda:
  - Intro to AES.
  - Structure of AES.
  - Internals of AES.

22

Intro to AES

AES is by now the most important symmetric encryption algorithm in the world.

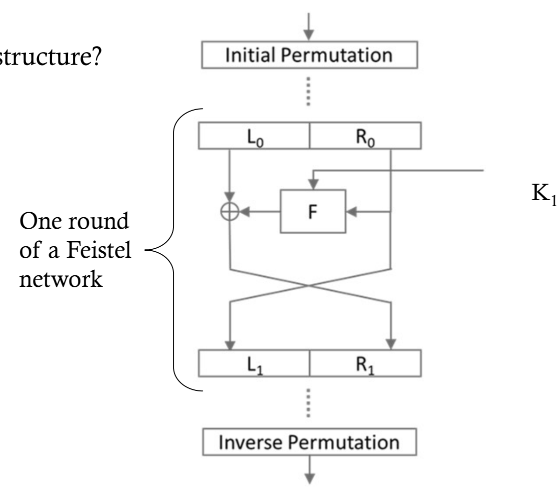
High level view of AES:



NSA uses AES for classified data with 192 or 256 key.

Structure of AES

Remember the Feistel structure?
(e.g., SDES, DES)



AES does **NOT** use the Feistel structure.

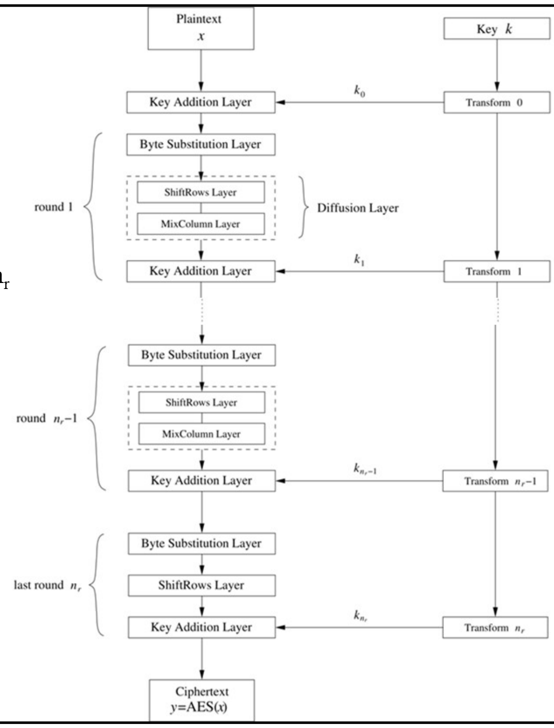
AES: Structure

AES encrypts all 128 bits of the data path in each round.

AES number of rounds depend on the key length:

key length	# rounds = n_r
128 bit	10
192 bit	12
256 bit	14

A subkey is added at the beginning and at the end of encryption (**key whitening**)



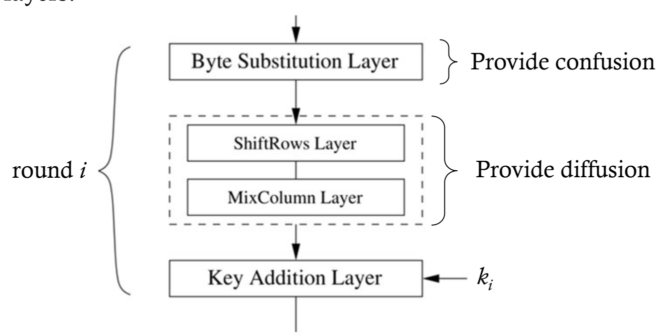
 25

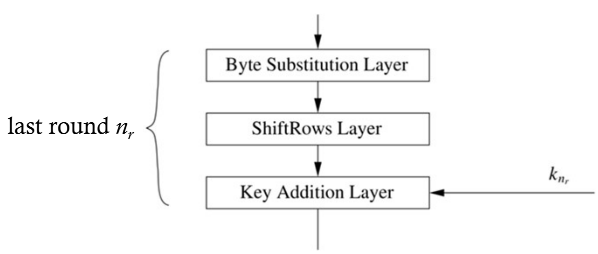
25

AES: Structure

Each round consists of 4 layers:

1) ByteSub
2) ShiftRow
3) MixColumn
4) Key Addition





Except in the last round, there's no MixColumn layer

 26

26

AES: Internals

Note: AES is byte oriented. The 128 bit data block (data path) is split into 16 bytes.

A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}

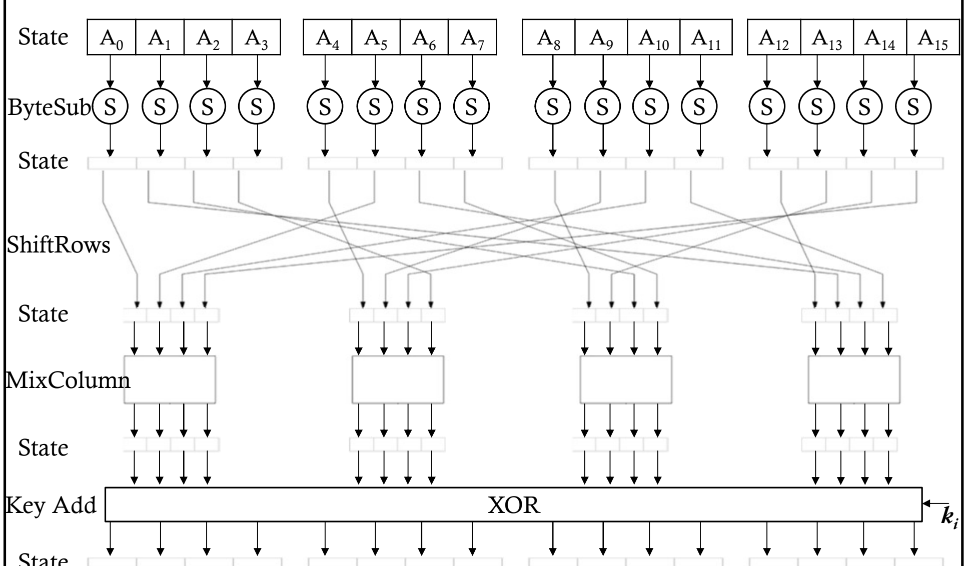
Bytes A_0 , A_1 , . . . , A_{15} are arranged in a four-by-four byte matrix called the state.

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

 27

27

AES Encryption Round



 28

28

AES Internals

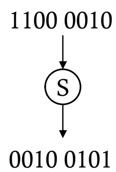
a) The Byte Substitution Layer:

All S-Boxes are identical.

Example:

$A_i = C2_{16} = (xy)$

$B_i = S(A_i) = 25_{16}$



AES S-Box: Substitution values in hexadecimal notation for input byte (xy)

$x \backslash y$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

To learn how the S-Box entries were constructed, see page 102. (Hint: GF)

AES Internals

b) Shift Rows Layer:

Outputs of 16 S-Boxes are rolled in a 4x4 state matrix:

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

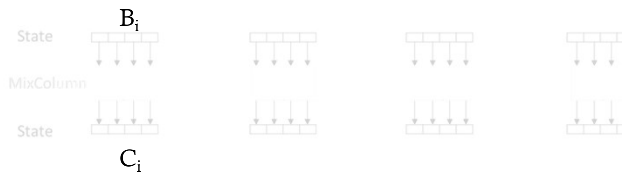
After the Shift Rows operation, the new state matrix becomes as follows:

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	⟵ one position left shift
B_{10}	B_{14}	B_2	B_6	⟵ two positions left shift
B_{15}	B_3	B_7	B_{11}	⟵ three positions left shift

So, the input to the next layer becomes:

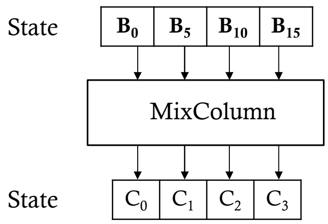
B_0	B_5	B_{10}	B_{15}	B_4	B_9	B_{14}	B_3	B_8	B_{13}	B_2	B_7	B_{12}	B_1	B_6	B_{11}

AES Internals



c) Mix Column Layer:

Example: 1st MixColumn Box (other three are identical)



$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

This way, one-bit flip in any of the input bytes affects C_0, C_1, C_2, C_3.

Note: The multiplications and additions for each C_i is done in $GF(2^8)$ with $P(x) = x^8 + x^4 + x^3 + x + 1$

See example in Paar page 105.

Useful Resources

A flash animation of the AES encryption:

http://www.formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng.swf

A Stick Figure Guide to AES:
http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

AES official specs:
https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

Further Reading

See AES-NI: AES instruction set for Intel processors

https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni/

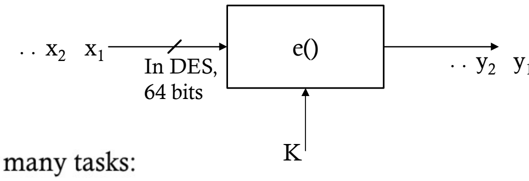
Software Library for AES Encryption and Decryption by Atmel

http://ww1.microchip.com/downloads/en/appnotes/atmel-42508-software-library-for-aes-128-encryption-and-decryption_applicationnote_at10764.pdf

Modes of Operation for Block Ciphers

- Introduction.
- Electronic Codebook Mode (ECB).
- Cipher Block Chaining Mode (CBC).
- Cipher Feedback Mode (CFB).

Introduction



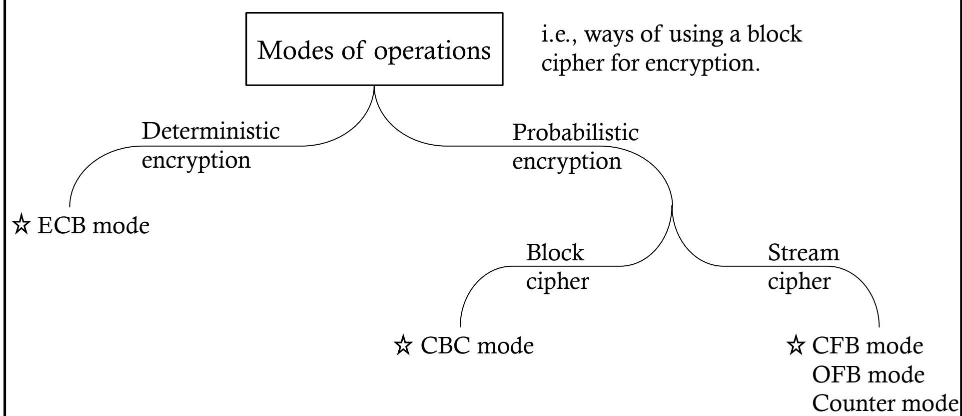
Block ciphers can be used for many tasks:

Today
- Different encryption schemes. ECB and CBC modes.
- Stream ciphers CFB mode.
- PRNG
- Hash function
- MACs
- . . .

Deterministic vs Probabilistic Encryption

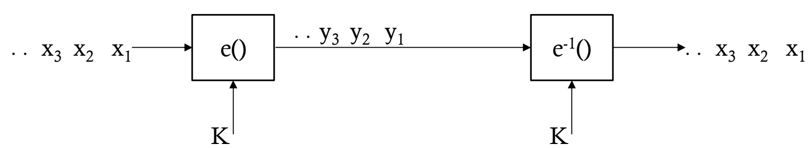
- In a deterministic encryption scheme, a particular plaintext is mapped to a fixed ciphertext, if the key is unchanged.
- A probabilistic encryption scheme is non-deterministic. i.e., if the same plaintext is encrypted twice, different ciphertexts are obtained.

Introduction



☆ i.e., today.

Electronic Codebook Mode (ECB)

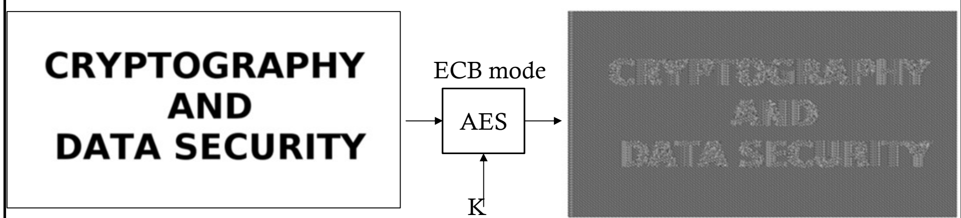


$y_i = e(x_i)$ $x_i = e^{-1}(y_i)$

Seems like the natural way of doing encryption...

But . . . Not a very good way, as we're going to see.

ECB Weakness:
Encryption of Bitmaps in ECB Mode



Bitmap image before encryption | 256-bit key | Encrypted bitmap image using ECB

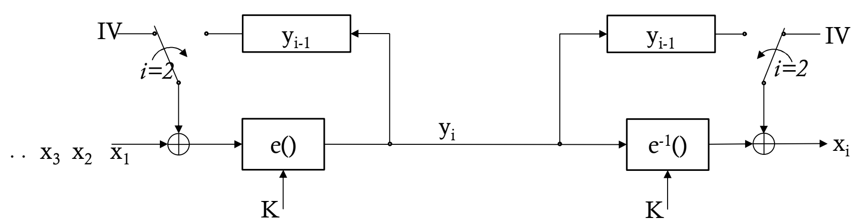
Simply because ECB is <u>deterministic</u>.

Identical plaintext blocks are mapped into identical cyphertext blocks.

Cipher Block Chaining Mode (CBC)

Main goal: Make the encryption <u>probabilistic</u>.

Idea: Use the ciphertext from the previous block, to impact the current block.



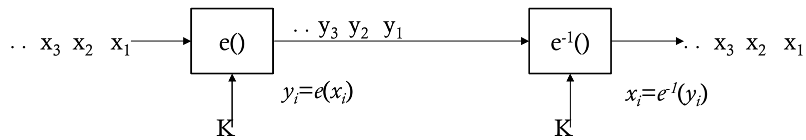
$y_1 = e(x_1 \oplus IV)$
$y_i = e(x_i \oplus y_{i-1})$, $i \geq 2$

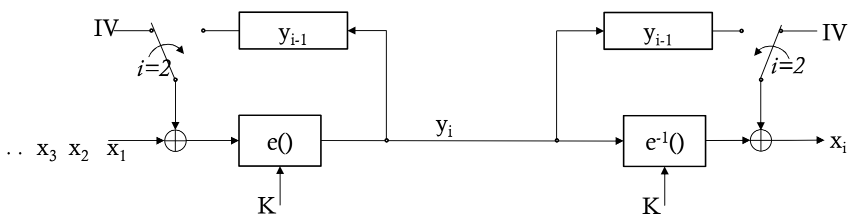
$x_1 = e^{-1}(y_1) \oplus IV$
$x_i = e^{-1}(y_i) \oplus y_{i-1}$, $i \geq 2$

IV: Initialization Vector.
Doesn't have to be a secret.
Should be a nonce, i.e., number used only once.
Could be from a TRNG, PRNG, counter, etc.

Review Question

Is **ECB** mode equivalent to **CBC** with zeros IV?





$y_1 = e(x_1 \oplus IV)$

$y_i = e(x_i \oplus y_{i-1})$, $i \geq 2$

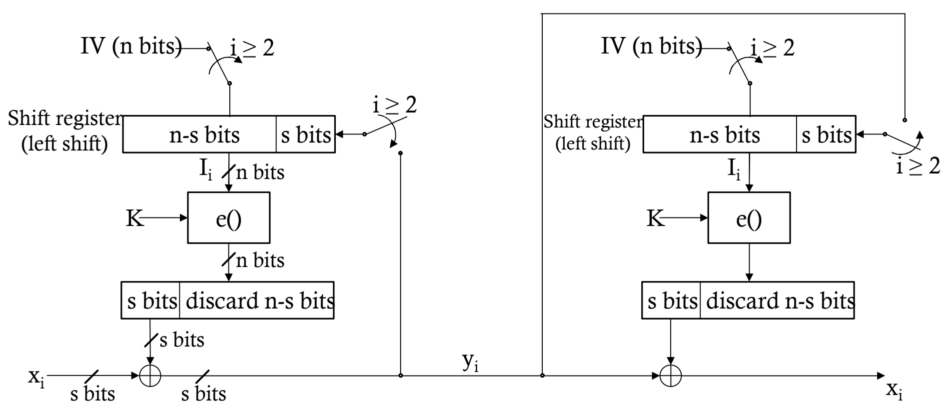
$x_1 = e^{-1}(y_1) \oplus IV$

$x_i = e^{-1}(y_i) \oplus y_{i-1}$, $i \geq 2$

Cipher Feedback Mode (CFB)

Goal: Generate an unpredictable key stream for stream cipher.

Idea: Construct the key stream generator using a block cipher.



$y_i = x_i \oplus MSB_S(e(I_i))$

$I_1 = IV$

$I_i = LSB_{n-s}(I_{i-1}) \,||\, y_{i-1}$, $i \geq 2$

$x_i = y_i \oplus MSB_S(e(I_i))$

$I_1 = IV$

$I_i = LSB_{n-s}(I_{i-1}) \,||\, y_{i-1}$, $i \geq 2$

Textbook

Paar:

- Chapter 4 (till section 4.4.3)
- Sections 5.1.1, 5.1.2, 5.1.4