Cristina Nita-Rotaru

CS355: Cryptography

Lecture 9: Encryption modes. AES

Encryption modes: ECB

- Message is broken into independent blocks of *block_size* bits;
- Electronic Code Book (ECB): each block encrypted separately.
- **Encryption: $c_i = E_k(x_i)$**
- **Decryption: $x_i = D_k(c_i)$**

Properties of ECB

Deterministic: the same data block gets encrypted the same way, reveals patterns of data when a data block repeats.

Malleable: reordering ciphertext results in reordered plaintext.

Errors in one ciphertext block do not propagate.

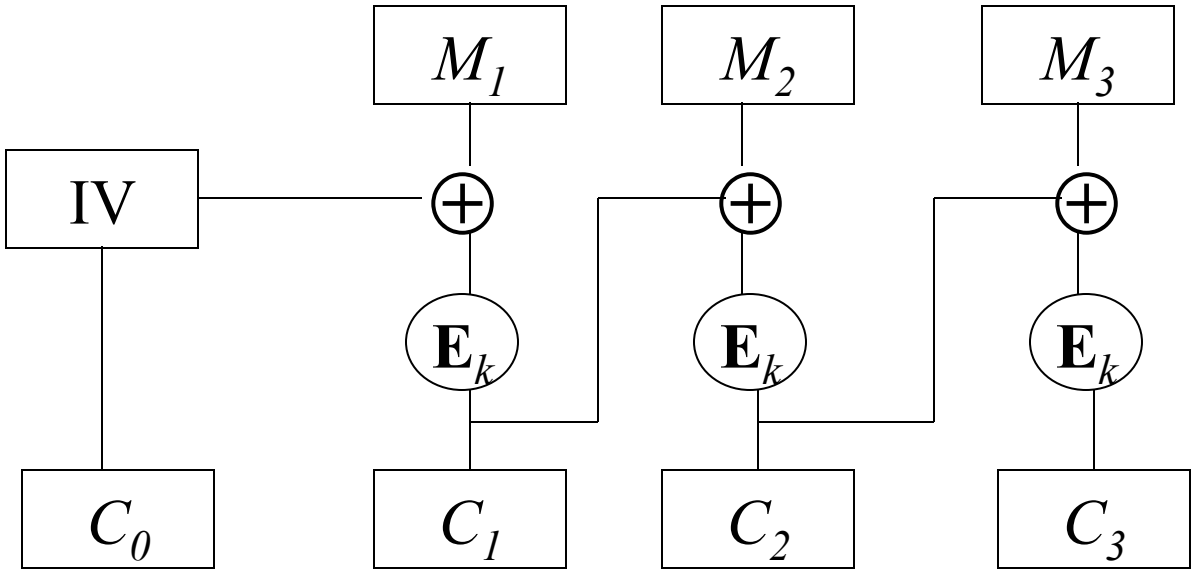
- Usage: not recommended to encrypt more than one block of data.

Encryption modes: CBC

- Cipher Block Chaining (CBC): next input depends upon previous output

  Encryption: $C_i = E_k(M_i \oplus C_{i-1})$, **with** $C_0 = IV$

  Decryption: $M_i = C_{i-1} \oplus D_k(C_i)$, **with** $C_0 = IV$



Properties of CBC

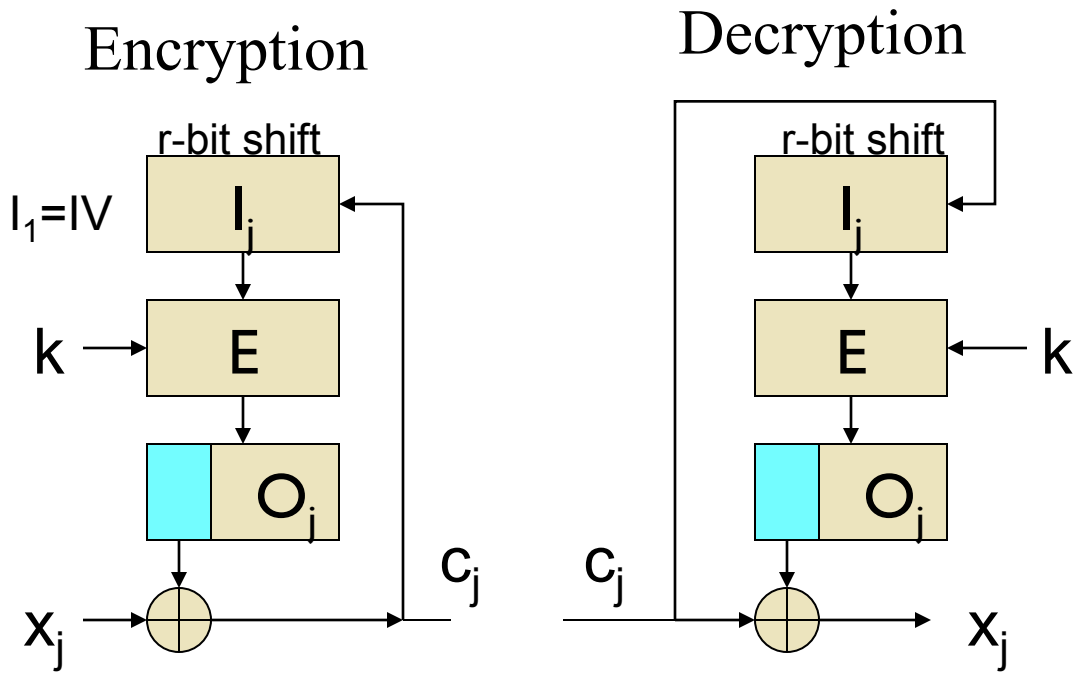
- Randomized encryption: repeated text gets mapped to different encrypted data.
  - can be proven to be "secure" assuming that the block cipher has desirable properties and that random IV's are used
- A ciphertext block depends on all preceding plaintext blocks; reorder affects decryption
- Errors in one block propagate to two blocks
  - one bit error in C_j affects all bits in M_j and one bit in M_{j+1}
- Sequential encryption, cannot use parallel hardware
- Usage: chooses random IV and protects the integrity of IV
- Observation: if $C_i=C_j$ then $\mathbf{E_k}\ (M_i \oplus C_{i-1}) = \mathbf{E_k}\ (M_j \oplus C_{j-1})$; **thus** $M_i \oplus C_{i-1} = M_j \oplus C_{j-1}$; **thus** $M_i \oplus M_j = C_{i-1} \oplus C_{j-1}$

Use DES to construct stream ciphers

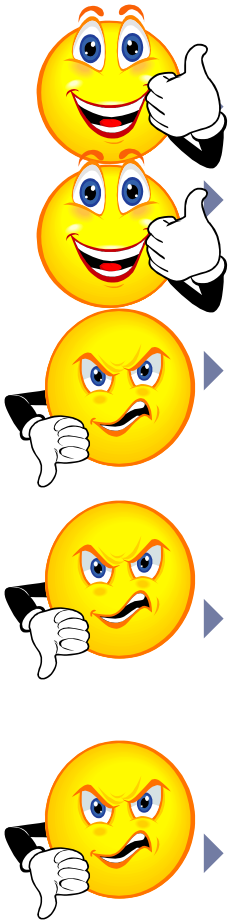
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)
- Common properties:
  - Uses only the encryption function of the cipher both for encryption and for decryption
  - Malleable: possible to make predictable bit changes

Encryption modes: CFB

- Cipher Feedback (CFB): the message is XORed with the feedback of encrypting the previous block



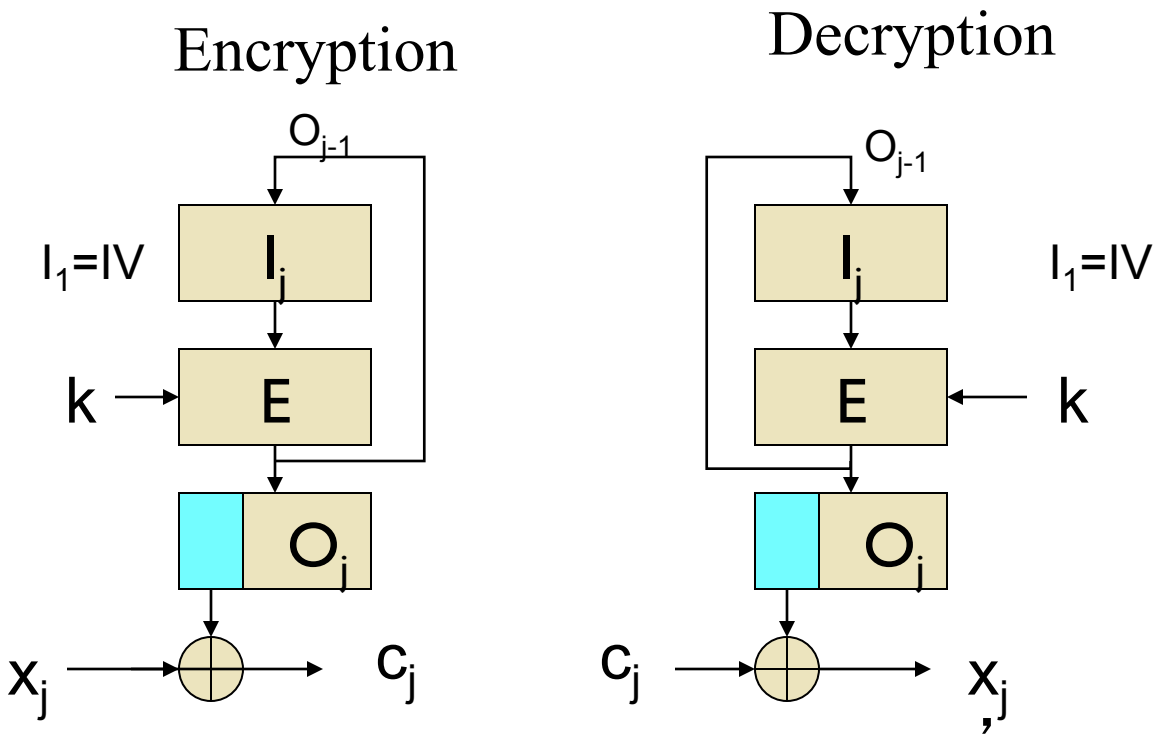
Properties of CFB



- Randomized encryption
- A ciphertext block depends on all preceding plaintext blocks; reorder affects decryption
- Errors propagate for several blocks after the error, but the mode is self-synchronizing (like CBC).
- Decreased throughput.
  - Can vary the number of bits feed back, trading off throughput for ease of use
- Sequential encryption

Encryption modes: OFB

- Output feedback (OFB):
  - construct a PRNG using DES
  - y_0=IV $y_i = E_k[y_{i-1}]$



Properties of OFB

- Randomized encryption
- Sequential encryption, but pre-processing possible
- Error propagation limited
- Subject to limitation of stream cipher

Encryption modes:CTR

- Counter Mode (CTR): Another way to construct PRNG using DES
  - $y_i = E_k[\text{counter}+i]$
  - Sender and receiver share: counter (does not need to be secret) and the secret key.

Properties of CTR

Software and hardware efficiency: different blocks can be encrypted in parallel.

Preprocessing: the encryption part can be done offline and when the message is known, just do the XOR.

Random access: decryption of a block can be done in random order, very useful for hard-disk encryption.

Messages of arbitrary length: ciphertext is the same length with the plaintext (i.e., no IV).

Summary so far

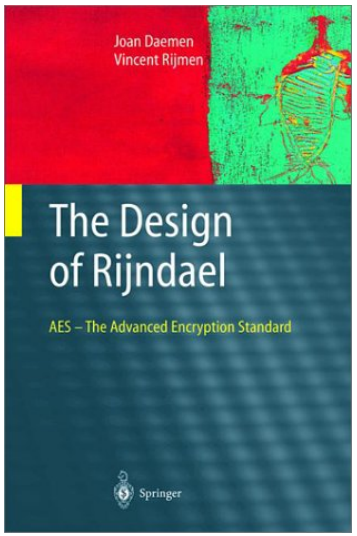
- Block ciphers must be used with encryption modes when encrypting larger messages: CBC or CTR modes

How to Improve Block Ciphers

- Variable key length
- Mixed operators: use more than one arithmetic and/or Boolean; this can provide non-linearity
- Data dependent rotation
- Key-dependent S-boxes
- Lengthy key schedule algorithm
- Variable plaintext/ciphertext block length
- Variable number of rounds
- Operation on both data halves each round
- Variable F function (varies from round to round
- Key-dependent rotation

Rijndael Features

- Designed to be efficient in both hardware and software across a variety of platforms.
- Uses a variable block size, **128,192, 256-bits**, key size **of 128-, 192-, or 256-bits.**
- 128-bit round key used for each round (Can be pre-computed and cached for future encryptions).
- Note: AES uses a 128-bit block size.
- Variable number of rounds (10, 12, 14):
  - 10 if B = K = 128 bits
  - 12 if either B or K is 192 and the other is ≤ 192
  - 14 if either B or K is 256 bits



Rijndael Design

- Operations performed on State (4 rows of bytes).
- The 128 bit key is expanded as an array of 44 32bits words; 4 distinct words serve as a round key for each round; key schedule relies on the S-box
- Algorithms composed of three layers
  - Linear diffusion
  - Non-linear diffusion
  - Key mixing

Rijandael: High-Level Description

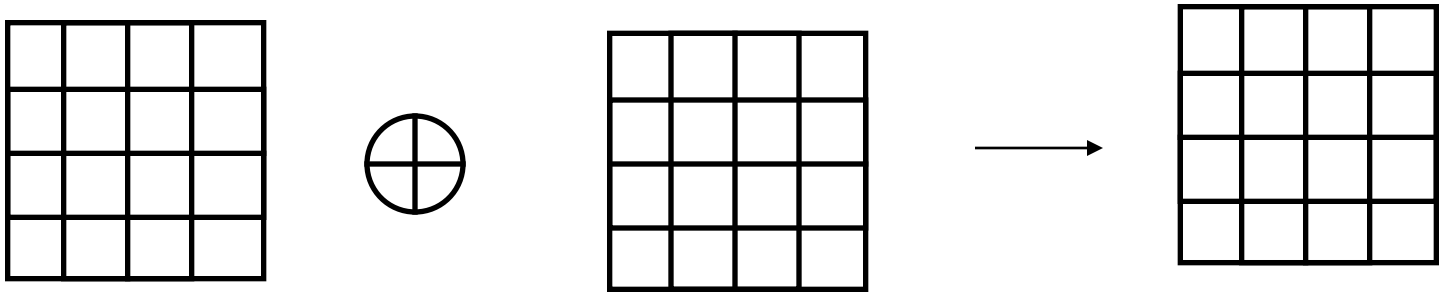
State = X
AddRoundKey(State, Key_0)
for r = 1 to Nr - 1
    SubBytes(State, S-box)
    ShiftRows(State)
    MixColumns(State)
    AddRoundKey(State, Key_r)
endfor
SubBytes(State, S-box)
ShiftRows(State)
AddRoundKey(State, Key_{Nr})
Y = State

AddRound Key

State is represented as follows (16 bytes):

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

AddRoundKey(State, Key):



key state state

SubBytes

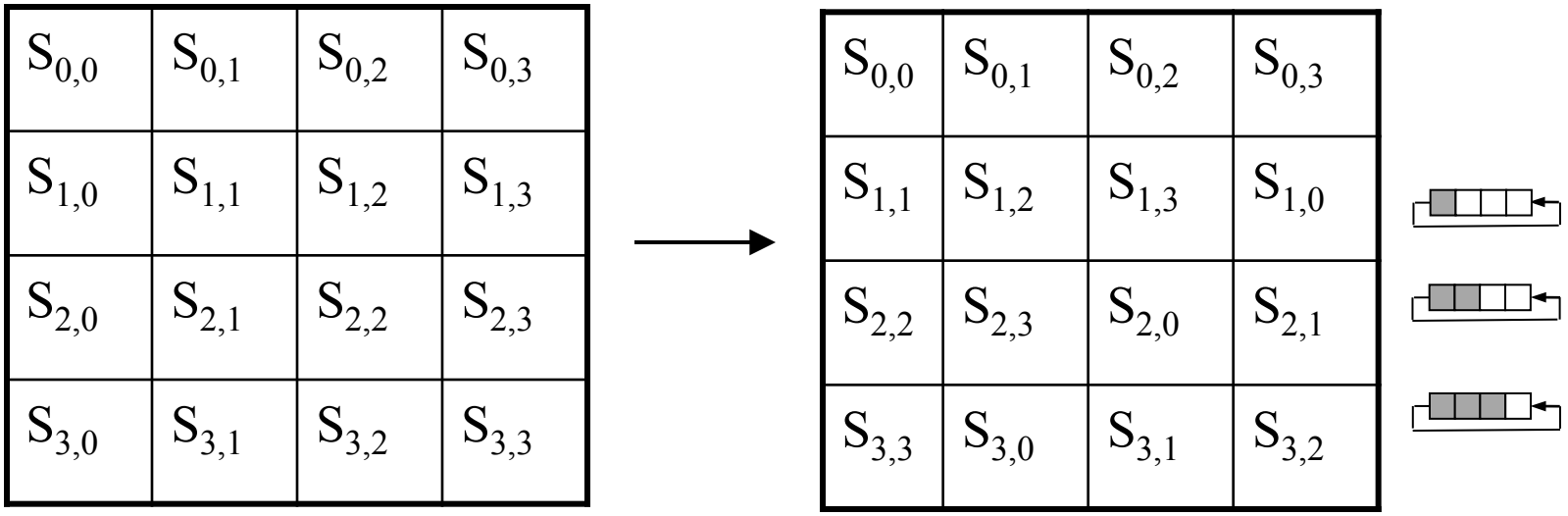
- Byte substitution using non-linear S-Box (independently on each byte).
- S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits
- 8 bytes replaced as follows: 8 bytes defines a hexadecimal number rc, then $s_{r,c}$ = binary(S-box(r, c))
- How is AES S-box different from DES S-box?
  - Only one S-box
  - S-boxes based on modular arithmetic with polynomials, can be defined algebraically, not random
  - Easy to analyze, prove attacks fail

S-box Table

	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	3	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

Example: hexa 53 is replaced with hexa ED

ShiftRows


$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,1}$ $S_{1,2}$ $S_{1,3}$ $S_{1,0}$
$S_{2,2}$ $S_{2,3}$ $S_{2,0}$ $S_{2,1}$
$S_{3,3}$ $S_{3,0}$ $S_{3,1}$ $S_{3,2}$

MixColumns

- Interpret each column as a vector of length 4.
- Each column of State is replaced by another column obtained by multiplying that column with a matrix in a particular field.

Decryption

- The decryption algorithm is not identical with the encryption algorithm, but uses the same key schedule.
- There is also a way of implementing the decryption with an algorithm that is equivalent to the encryption algorithm (each operation replaced with its inverse), however in this case, the key schedule must be changed.

Rijandel Cryptanalysis

Resistant to linear and differential cryptanalysis

- Academic break on weaker version of the cipher, 9 rounds
- Requires 2^{224} work and 2^{85} chosen *related-key* plaintexts.
- Attack not practical.

Rijandel Cryptanalysis as of 2012

2009, attack against 11-round AES-256 that requires 2^{70} time

- Attack can not be extended to AES-128
- It is against 11-round AES-256, AES-256 requires 14 rounds
- Exploits weaknesses in the key schedule for AES-256-bit
- Requires the cryptanalyst to have access to plaintexts encrypted with multiple keys that are related in a specific way.
- **Attacks are not practical**

Rijandel Cryptanalysis as of 2012

- New attacks published in 2011 that do not need related keys
  - The first key recovery attack on the full AES-128 with computational complexity $2^{126.1}$.
  - The first key recovery attack on the full AES-192 with computational complexity $2^{189.7}$.
  - The first key recovery attack on the full AES-256 with computational complexity $2^{254.4}$.
  - Attacks with lower complexity on the reduced-round versions of AES not considered before, including an attack on 8-round AES-128 with complexity $2^{124.9}$.
- **Attacks are not practical.**

So were are we?

- New attacks show that AES safety margin with respect to attacks is not as large as believed

- Can be fixed by increasing the number of rounds. Schneier suggest
  - *AES-128 use 16 rounds*
  - *AES-192 use 20 rounds*
  - *AES-256 use 28 rounds*

Summary

- There are no attacks against any AES variants that are better than brute force
- All existing attacks are against reduced-round variants
- *Stay tuned: attacks never get worse, they only get better.*