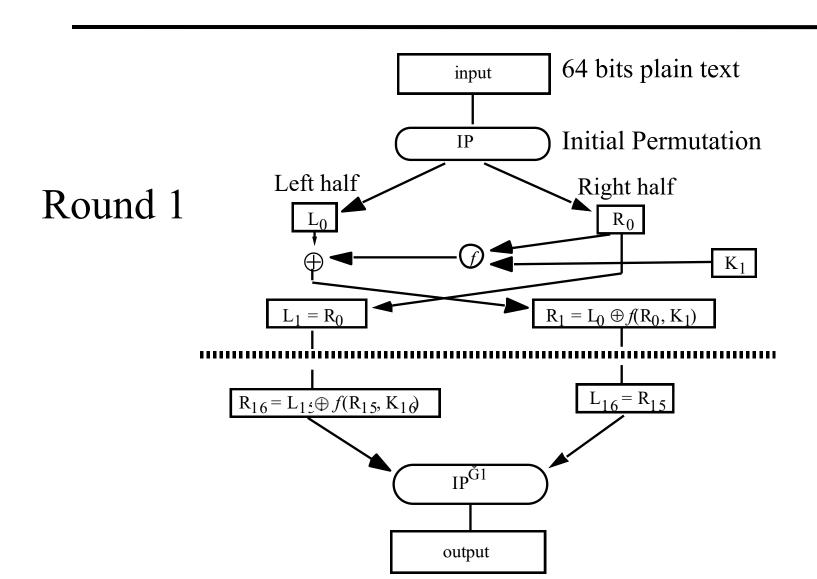
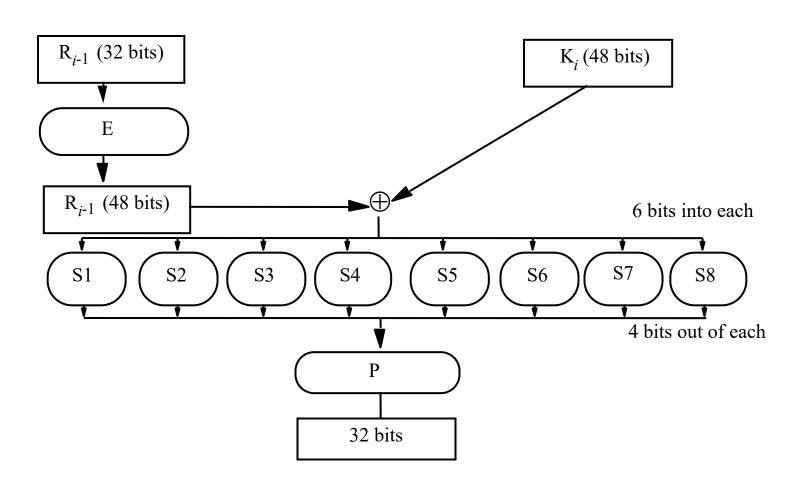
Overview of the DES

- A Symmetric Key Scheme
- A block cipher:
 - encrypts blocks of 64 bits using a 64 bit key
 - outputs 64 bits of ciphertext
 - A product cipher
 - performs both substitution and transposition (permutation) on the bits
 - basic unit is the bit
- Cipher consists of 16 rounds (iterations), each with a round key generated from the user-supplied key

Encipherment Illustration



The f Function



DES Modes

- Electronic Code Book Mode (ECB)
 - Encipher each block independently

ECB Problem

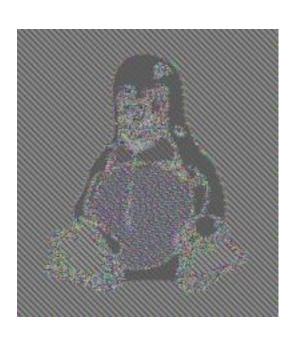
- Problem: identical plaintext blocks produce identical ciphertext blocks
 - Example: two database records
 - MEMBER: HOLLY INCOME \$100,000
 - MEMBER: HEIDI INCOME \$100,000
 - Encipherment:
 - ABCQZRME GHQMRSIB CTXUVYSS RMGRPFQN
 - ABCQZRME ORMPABRZ CTXUVYSS RMGRPFQN
 - Fails to hide patterns in plaintext

Example of ECB failure

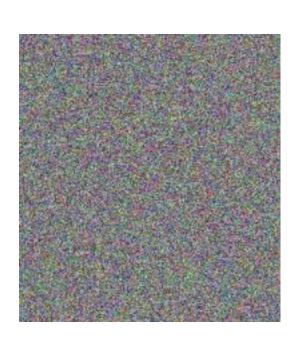
• Pixelmap image of Tux encoded by ECB, and not



Tux



ECB encoding



Non-ECB encoding

DES Modes

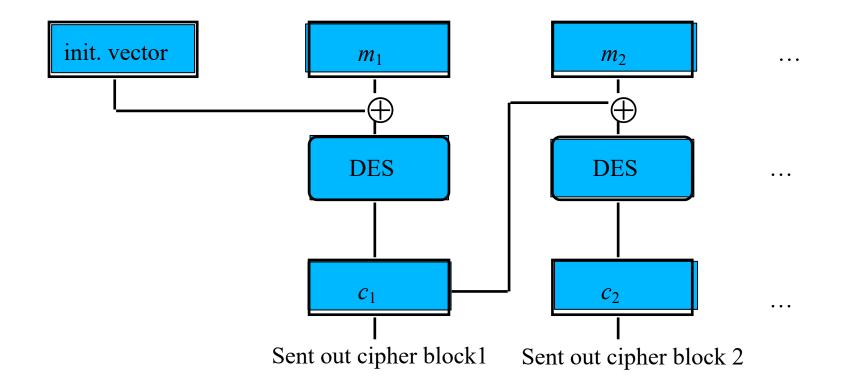
- Electronic Code Book Mode (ECB)
 - Encipher each block independently
- Cipher Block Chaining Mode (CBC)
 - Xor each block with previous ciphertext block
 - Requires an initialization vector for the first one

CBC Mode Encryption

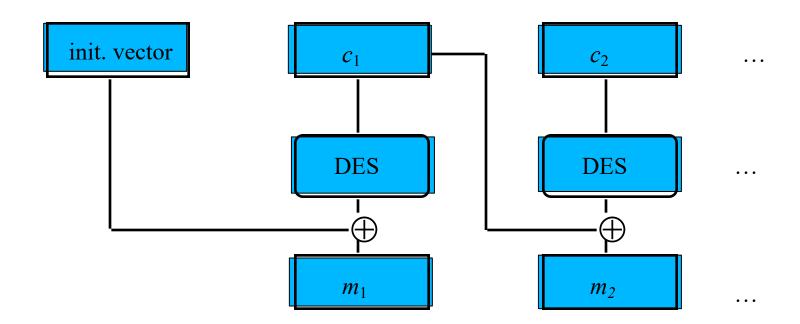
Cipher Block Chaining Mode (CBC)

Xor each block with previous ciphertext block

Requires an initialization vector for the first one



CBC Mode Decryption



CBC mode Self-Healing Property

- Initial message
- Received Ciphertext as (underlined 4c should be 4b)
 - ef7c4cb2b4ce6f3b f6266e3a97af0e2c 746ab9a6308f4256 33e60b451b09603d
- Which decrypts to
 - efca61e19f4836f1 3231333336353837
 3231343336353837 3231343336353837
 - Incorrect bytes underlined
 - Plaintext "heals" after 2 blocks

How does self-healing work?

• Suppose C_i becomes corrupted in transmission (e.g. after encryption) Note that

$$m_i = D_k(c_i) \oplus c_{i-1}$$
 $m_{i+1} = D_k(c_{i+1}) \oplus c_i$
 $m_{i+2} = D_k(c_{i+2}) \oplus c_{i+1}$

- so the ith and (i+1)st message blocks are corrupted. The (i+2)nd block is free from the corrupted ciphertext
- What about C_i is corrupted during the encryption, before c_{i+1} is calculated?

DES Modes

- Electronic Code Book Mode (ECB)
 - Encipher each block independently
- Cipher Block Chaining Mode (CBC)
 - Xor each block with previous ciphertext block
 - Requires an initialization vector for the first one
- Encrypt-Decrypt-Encrypt Mode (2 keys: *k*, *k* ')
 - $-c = DES_k(DES_k^{-1}(DES_k(m)))$
- Encrypt-Encrypt Mode (3 keys: k, k', k'')
 - $-c = DES_k(DES_{k'}(DES_{k'}(m)))$

Advanced Encryption Standard (AES) Background

- Clearly a replacement for DES was needed
- US NIST issued call for ciphers in 1997
 - 15 candidates accepted in Jun 98
 - 5 were short-listed in Aug-99
- Rijndael was selected as AES in Oct-2000
 - issued as FIPS PUB 197 standard in Nov-2001
 - http://csrc.nist.gov/publications/fips/fips197/ fips-197.pdf

AES Requirements

- Private key symmetric block cipher
 - 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- Provide full specification & design details
- Both C & Java implementations
- NIST have released all submissions & unclassified analyses

AES Evaluation Criteria

- Final criteria
 - general security
 - software & hardware implementation ease
 - defence against attacks
 - flexibility

AES Shortlist

- Shortlist August-99:
 - MARS (IBM) -complex, fast, high security margin
 - RC6 (USA) -v. simple, v. fast, low security margin
 - Rijndael(Belgium) -clean, fast, good security margin
 - Serpent (Euro) -slow, clean, v. high security margin
 - Twofish(USA) -complex, v. fast, high security margin
- Saw contrast between algorithms with
 - few complex rounds verses many simple rounds
 - refined existing ciphers verses new proposals

The AES Cipher - Rijndael

- Designed by Rijmen-Daemenin Belgium
 - Has 128/192/256 bit keys, 128 bit data
- An iterative cipher
 - treats data in 4 groups of 4 bytes
 - 4x4 matrix in column major order
 - operates an entire block in every round

AES Overview

- 128 bit block represented by a 4x4 byte matrix
- Processing on each block comprised of several rounds
 - 10 for 128-bit key, 12 for 192-bit key, 14 for 256-bit key

AES: State array

 $S_{0,3}$

 $S_{1,3}$

 $S_{2,3}$

 $S_{3,3}$

	input	bytes	
in _o	in ₄	in ₈	in ₁₂
in ₁	in₅	in ₉	in ₁₃
in ₂	in ₆	in ₁₀	in ₁₄
in ₃	in_7	in ₁₁	in ₁₅

 in_0

 in_1

 in_2

 in_3

npui	t bytes		State array					
in ₄	in ₈	in ₁₂		S _{0,0}	S _{0,1}	S _{0,2}		
in₅	in ₉	in ₁₃	_	S _{1.0}	S _{1,1}	S _{1,2}		
in.	in.	in.	7	Sac	San	Saa		

 $S_{3,0}$

 $S_{3,1}$

 $S_{3,2}$

 in_{15}

 in_{11}

 in_7

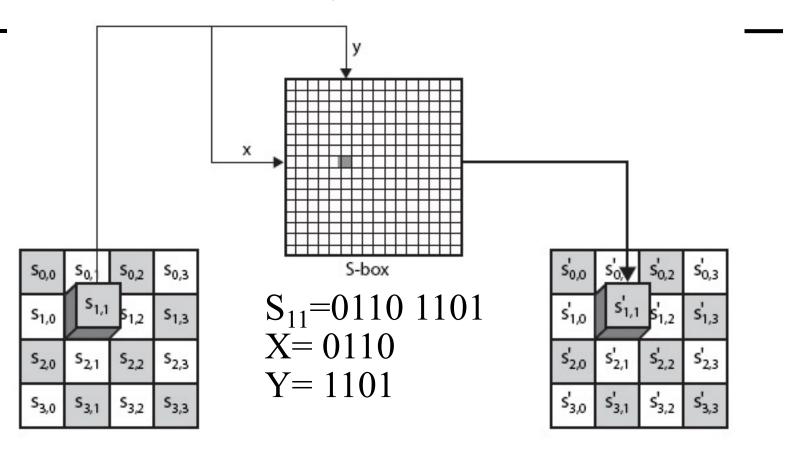
	output	t bytes						
out ₀	out ₄	out ₈	out ₁₂					
out ₁	out ₅	out ₉	out ₁₃					
out ₂	out ₆	out 10	out ₁₄					
out ₃	out,	out ₁₁	out ₁₅					
output bytes								
out _o	out ₄	out ₈	out ₁₂					
out ₁	out ₅	out ₉	out ₁₃					
out ₂	out ₆	out 10	out ₁₄					

out3 out7 out11 out15

Four Steps in Each Round

- Substitute Bytes single byte based
- Shift Rows row-wise permutation
- Mix Columns column-wise mixing
- Add Round Keys

AES: SubBytes() (S-Box)



A simple substitution of each byte Uses one table of 16x16 bytes containing a permutation of all 256 8-bit values

AES S-Box

00	01	02	03	04	05	06	07	08	09	0a	0b	0с	0d	0e	Of
63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
ca	82	с9	7d	fa	59	47	fO	ad	d4	a2	af	9с	a4	72	c0
b7	fd	93	26	36	3f	f7	СС	34	a5	e5	f1	71	d8	31	15
04	с7	23	сЗ	18	96	05	9a	07	12	80	e2	eb	27	b2	75
09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	еЗ	2f	84
53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3с	9f	a8
51	аЗ	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
cd	0c	13	ес	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
e0	32	За	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
70	Зе	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9е
e1	f8	98	11	69	d9	8e	94	9b	1e	87	e 9	се	55	28	df
8c	a1	89	0d	bf	e6	42	68	41	99	2d	Of	b0	54	bb	16
	63 ca b7 04 09 53 d0 51 cd 60 e0 e7 ba 70 e1	63 7c ca 82 b7 fd 04 c7 09 83 53 d1 d0 ef 51 a3 cd 0c 60 81 e0 32 e7 c8 ba 78 70 3e e1 f8	63 7c 77 ca 82 c9 b7 fd 93 04 c7 23 09 83 2c 53 d1 00 d0 ef aa 51 a3 40 cd 0c 13 60 81 4f e0 32 3a e7 c8 37 ba 78 25 70 3e b5 e1 f8 98	63 7c 77 7b ca 82 c9 7d b7 fd 93 26 04 c7 23 c3 09 83 2c 1a 53 d1 00 ed d0 ef aa fb 51 a3 40 8f cd 0c 13 ec 60 81 4f dc e0 32 3a 0a e7 c8 37 6d ba 78 25 2e 70 3e b5 66 e1 f8 98 11	63 7c 77 7b f2 ca 82 c9 7d fa b7 fd 93 26 36 04 c7 23 c3 18 09 83 2c 1a 1b 53 d1 00 ed 20 d0 ef aa fb 43 51 a3 40 8f 92 cd 0c 13 ec 5f 60 81 4f dc 22 e0 32 3a 0a 49 e7 c8 37 6d 8d ba 78 25 2e 1c 70 3e b5 66 48 e1 f8 98 11 69	63 7c 77 7b f2 6b ca 82 c9 7d fa 59 b7 fd 93 26 36 3f 04 c7 23 c3 18 96 09 83 2c 1a 1b 6e 53 d1 00 ed 20 fc d0 ef aa fb 43 4d 51 a3 40 8f 92 9d cd 0c 13 ec 5f 97 60 81 4f dc 22 2a e0 32 3a 0a 49 06 e7 c8 37 6d 8d d5 ba 78 25 2e 1c a6 70 3e b5 66 48 03 e1 f8 98 11 <	63 7c 77 7b f2 6b 6f ca 82 c9 7d fa 59 47 b7 fd 93 26 36 3f f7 04 c7 23 c3 18 96 05 09 83 2c 1a 1b 6e 5a 53 d1 00 ed 20 fc b1 d0 ef aa fb 43 4d 33 51 a3 40 8f 92 9d 38 cd 0c 13 ec 5f 97 44 60 81 4f dc 22 2a 90 e0 32 3a 0a 49 06 24 e7 c8 37 6d 8d d5 4e ba 78 25 2e 1c a6	63 7c 77 7b f2 6b 6f c5 ca 82 c9 7d fa 59 47 f0 b7 fd 93 26 36 3f f7 cc 04 c7 23 c3 18 96 05 9a 09 83 2c 1a 1b 6e 5a a0 53 d1 00 ed 20 fc b1 5b d0 ef aa fb 43 4d 33 85 51 a3 40 8f 92 9d 38 f5 cd 0c 13 ec 5f 97 44 17 60 81 4f dc 22 2a 90 88 e0 32 3a 0a 49 06 24 5c e7 c8 37	63 7c 77 7b f2 6b 6f c5 30 ca 82 c9 7d fa 59 47 f0 ad b7 fd 93 26 36 3f f7 cc 34 04 c7 23 c3 18 96 05 9a 07 09 83 2c 1a 1b 6e 5a a0 52 53 d1 00 ed 20 fc b1 5b 6a d0 ef aa fb 43 4d 33 85 45 51 a3 40 8f 92 9d 38 f5 bc cd 0c 13 ec 5f 97 44 17 c4 60 81 4f dc 22 2a 90 88 46 e0 32	63 7c 77 7b f2 6b 6f c5 30 01 ca 82 c9 7d fa 59 47 f0 ad d4 b7 fd 93 26 36 3f f7 cc 34 a5 04 c7 23 c3 18 96 05 9a 07 12 09 83 2c 1a 1b 6e 5a a0 52 3b 53 d1 00 ed 20 fc b1 5b 6a cb d0 ef aa fb 43 4d 33 85 45 f9 51 a3 40 8f 92 9d 38 f5 bc b6 cd 0c 13 ec 5f 97 44 17 c4 a7 e0 81 4f	63 7c 77 7b f2 6b 6f c5 30 01 67 ca 82 c9 7d fa 59 47 f0 ad d4 a2 b7 fd 93 26 36 3f f7 cc 34 a5 e5 04 c7 23 c3 18 96 05 9a 07 12 80 09 83 2c 1a 1b 6e 5a a0 52 3b d6 53 d1 00 ed 20 fc b1 5b 6a cb be d0 ef aa fb 43 4d 33 85 45 f9 02 51 a3 40 8f 92 9d 38 f5 bc b6 da cd 0c 13 ec 5f 97	63 7c 77 7b f2 6b 6f c5 30 01 67 2b ca 82 c9 7d fa 59 47 f0 ad d4 a2 af b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 04 c7 23 c3 18 96 05 9a 07 12 80 e2 09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 53 d1 00 ed 20 fc b1 5b 6a cb be 39 d0 ef aa fb 43 4d 33 85 45 f9 02 7f 51 a3 40 8f 92 9d 38 f5 bc b6 da	63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 51 a3 40 8f	63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c d0 ef aa fb 43 4d 33 85 45 fp 02 7f	63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 d0 ef aa fb 43 4d

The column is determined by the least significant nibble, and the row by the most significant nibble. For example, the value $9a_{16}$ is converted into $b8_{16}$.

Example:

itution

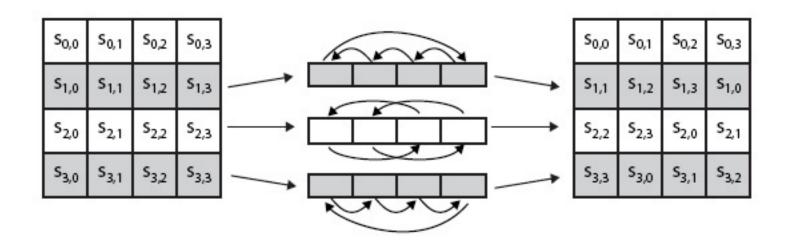
EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

Shift Rows

- A circular byte shift in each row
 - − 1st row is unchanged
 - -2^{nd} row does 1 byte circular shift to left
 - 3rd row does 2 byte circular shift to left
 - 4th row does 3 byte circular shift to left
- Decrypt does shifts to right

AES: ShiftRows()



97

EC

C3

95

87	F2	4D	97		87	F2	4D
EC	6E	4C	90		6E	4C	90
4A	C3	46	E7	-	46	E7	4A
8C	D8	95	A6		A6	8C	D8