Block ciphers

Myrto Arapinis
School of Informatics
University of Edinburgh

January 27, 2014

Block ciphers

A block cipher with parameters k and ℓ is a pair of deterministic algorithms (E, D) such that

- Encryption $E : \{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$
- Decryption $D : \{0,1\}^k \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$

Examples:

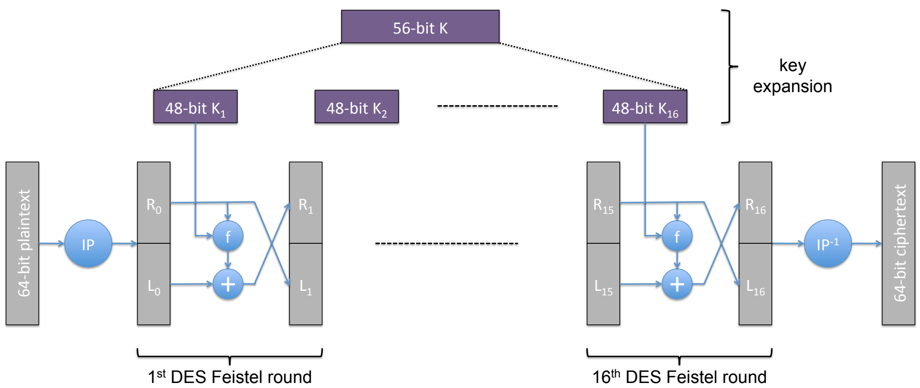
$3DES : \ \ell = 64, \ k = 56$
$AES : \ \ell = 64, \ k = 128, 192, 256$

Data Encryption Standard (DES)

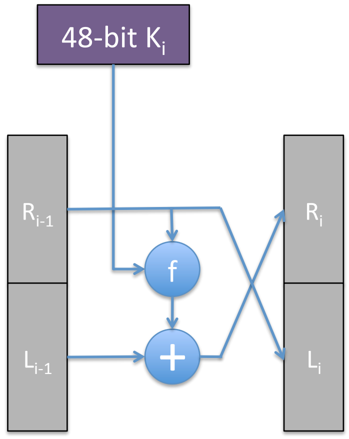
- Early 1970s: Horst Feistel designs Lucifer at IBM
  $k = 128$ bits, $\ell = 128$ bits
- 1973: NBS calls for block cipher proposals.
  $\longrightarrow$ IBM submits a variant of Lucifer.
- 1976: NBS adopts *DES* as a federal standard
  $k = 56$ bits, $\ell = 64$ bits
- 1997: *DES* broken by exhaustive search
- 2001: NIST adopts *AES* to replace *DES*
  $k = 128, 192, 256$ bits, $\ell = 128$ bits

Widely deployed in banking (ATM machines) and commerce

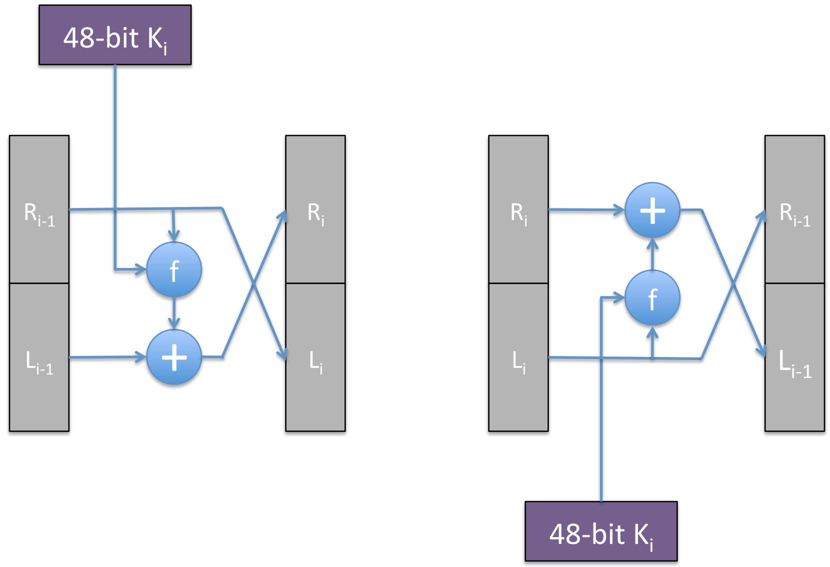
DES: encryption circuit



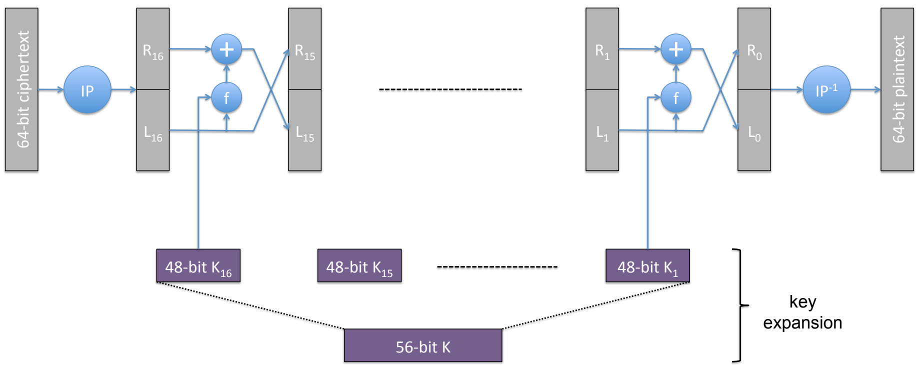
Each *DES* Feistel round is invertible



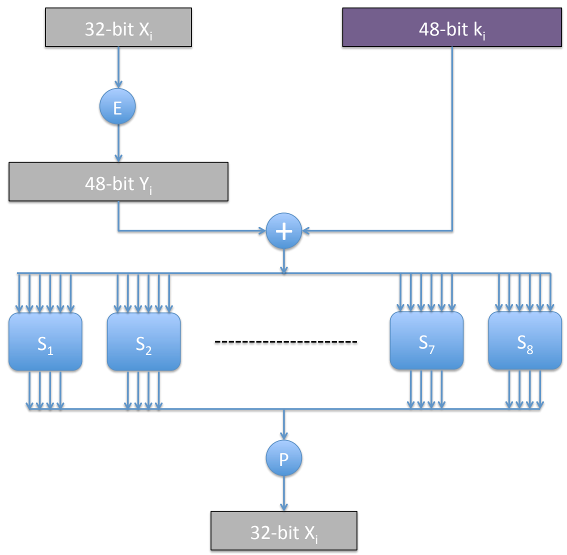
Each *DES* Feistel round is invertible



DES: decryption circuit



DES: the function *f*



DES: S_5-box

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

$S_5 : \{0,1\}^6 \rightarrow \{0,1\}^4$
(source: Wikipedia)

$\longrightarrow$ Note that S_5 is not reversible as it maps 6 bits to 4 bits.

Attacks on *DES*

- **Exhaustive search:** it takes 2^{56} to do an exhaustive search over the key space
  $\longrightarrow$ COBACOBANA (120 FPGAs, $\sim$ 10K\$): 7 days
- **Linear cryptanalysis:** found affine approximations to *DES*
  $\longrightarrow$ can find 14 key bits in time 2^{42}
  brute force the remaining 56-14=42 in time 2^{42}
  $\Rightarrow$ total attack time $\approx 2^{43}$

$\Rightarrow$ *DES* is badly broken! Do not use it in new projects!!

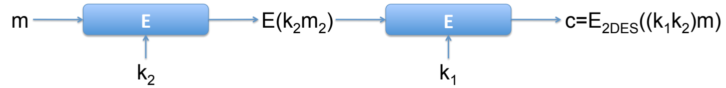
Triple *DES* (3*DES*)

- Goal: build on top of *DES* a block cipher resistant against exhaustive search attacks
- Let $DES = (E_{DES}, D_{DES})$. We build $3DES = (E_{3DES}, D_{3DES})$ as follows
  - $E_{3DES} : (\{0,1\}^k)^3 \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$
    $E_{3DES}((K_1, K_2, K_3), M) = E_{DES}(K_1, D_{DES}(K_2, E_{DES}(K_3, M)))$
    $\longrightarrow K_1 = K_2 = K_3 \Rightarrow DES$
  - $D_{3DES} : (\{0,1\}^k)^3 \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell$
    $D_{3DES}((K_1, K_2, K_3), C) = D_{DES}(K_3, E_{DES}(K_2, D_{DES}(K_1, C)))$

  $\longrightarrow$ 3 times as slow as *DES*!!
- key-size $= 3 \times 56 = 168$ bits
  $\Rightarrow$ Exhaustive search attack in 2^{168}
- simple (meet in the middle) attack in time 2^{118}

What about double DES (2DES)?

- $E_{2DES}((K_1, K_2), M) = E_{DES}(K_1, E_{DES}(K_2, M))$



  $\Rightarrow$ For m and c such that $E_{2DES}((k_1, k_2), m) = c$ we have that $E_{DES}(k_2, m) = D_{DES}(k_1, m)$
- 2*DES* admits a meet in the middle attack that reduces the time for key recovery from 2^{112} for an exhaustive search to 2^{56}. Given $M = (m_1, \dots, m_{10})$ and $C = (E_{2DES}((k_1, k_2), m_1), \dots, E_{2DES}((k_1, k_2), m_{10}))$

$$\left.\begin{array}{l}\text{- For all possible } k_2, \text{compute} E_{DES}(k_2, M) \\ \text{- Sort table according to the resulting } E_{DES}(k_2, M)\end{array}\right\} 2^{56}\log(56)$$

$$\left.\begin{array}{l}\text{- For each possible } k_1, \text{compute} D_{DES}(k_1, C) \\ \text{- Look up in the table if } D_{DES}(k_1, C) = E_{DES}(k_2, M)\end{array}\right\} 2^{56}\log(56)$$

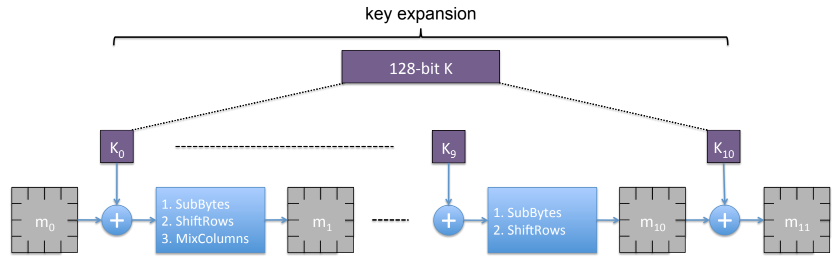
$\Rightarrow$ time $< 2^{63}$

- Similar attack on 3*DES* in time 2^{118}

The Advenced Encryption Standard (AES)

- Goal: replace 3*DES* which is too slow (3*DES* is 3 times as slow as *DES*)
- 2001: NIST adopts Rijndael as AES
- Block size $\ell = 128$ bits, Key size $k = 128, 192, 256$ bits
- *AES* is Substitution-Permutation network (not a Feistel network)

AES: encryption circuit



- m_i : 4×4 byte matrix, K_i: 128-bit key
- m_0: plaintext, m_{11}: ciphertext
- at the last round MixColumns is not applied

$\longrightarrow$ As *AES* is not a Feistel network, each step needs to be reversible!

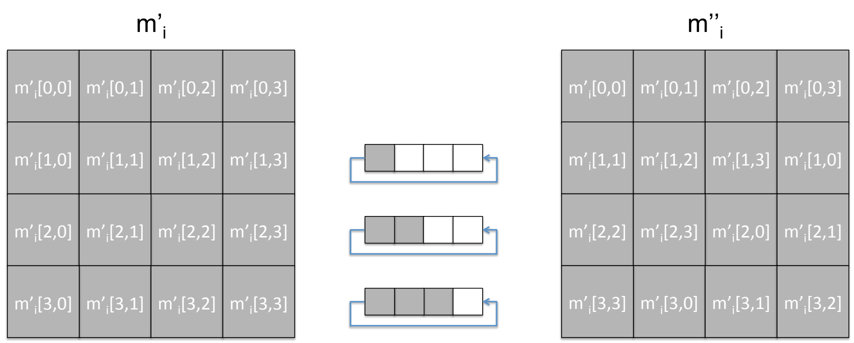
AES: SubBytes

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

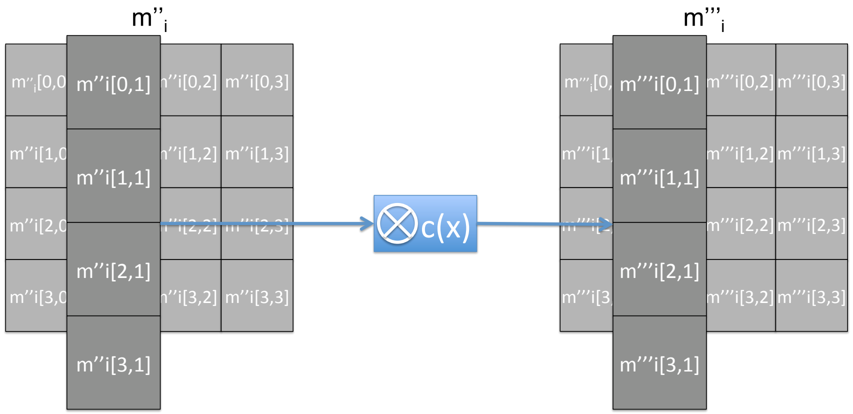
- $\forall j, k.\ m'_i[j, k] = S[m_i[j, k]]$
- rows: most significant 4 bits
- columns: least significant 4 bits

$\longrightarrow$ Note that SubBytes is reversible

AES: ShiftRows



AES: MixColumns



Attacks on *AES*

- **Related-key attack** on the 192-bit and 256-bit versions of *AES*: exploits the *AES* key schedule [A. Biryukov, D. Khovratovich (2009)]
  $\longrightarrow$ key recovery in time $\sim 2^{99}$
- First **key-recovery attack** on full *AES* [A. Bogdanov, D. Khovratovich, C. Rechberger (2011)]
  $\longrightarrow$ 4 times faster than exhaustive search

$\Rightarrow$ Existing attacks on $AES-128$ are still not practical, but use $AES-192$ and $AES-256$ in new projects!

Using block ciphers

Myrto Arapinis
School of Informatics
University of Edinburgh

Goal: Encrypt M using a block cipher operating on blocks of length ℓ when $|M| > \ell$

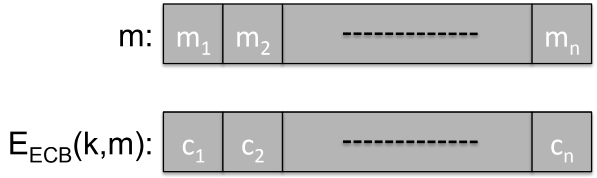
Electronic Code Book (ECB) mode

(E, D) a block cipher.

To encrypt message M under key K using ECB mode:

- M is padded:
  $\Rightarrow \quad M' = M||P$ such that $|M'| = m \times \ell$
- M' is broken into m blocks of length ℓ
  $\Rightarrow \quad M' = M_1 \,||\, M_2 \,||\, \ldots \,||\, M_m$
- Each block M_i is encrypted under the key K using the block cipher
  $\Rightarrow \quad C_i = E(K, M_i)$ for all $i \in \{1, \ldots, m\}$
- The ciphertext corresponding to M is the concatenation of the C_is
  $\Rightarrow \quad C = C_1 \,||\, C_2 \,||\, \ldots \,||\, C_m$

Weakness of ECB



Problem: $\forall i, j.\ m_i = m_j \ \Rightarrow\ c_i = E(k, m_i) = E(k, m_j) = c_j$

$\Rightarrow$ Weak to frequency analysis!

$\longrightarrow$ Playstation attack

Weakness of ECB in pictures

Original image

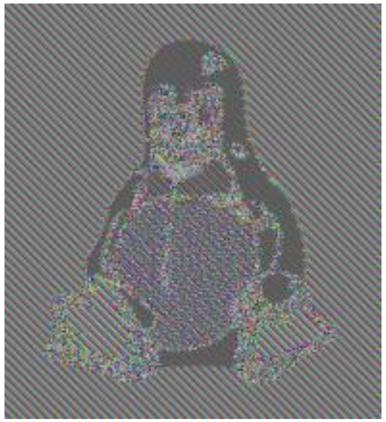
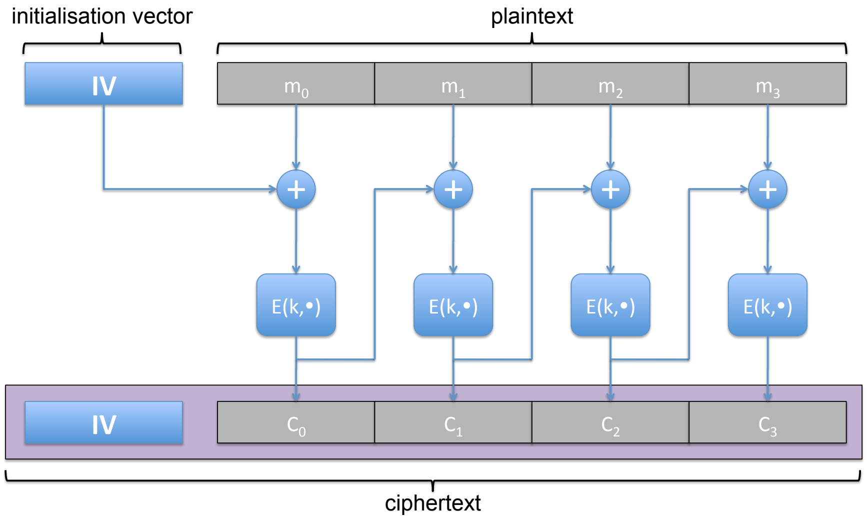


Image encrypted using ECB mode

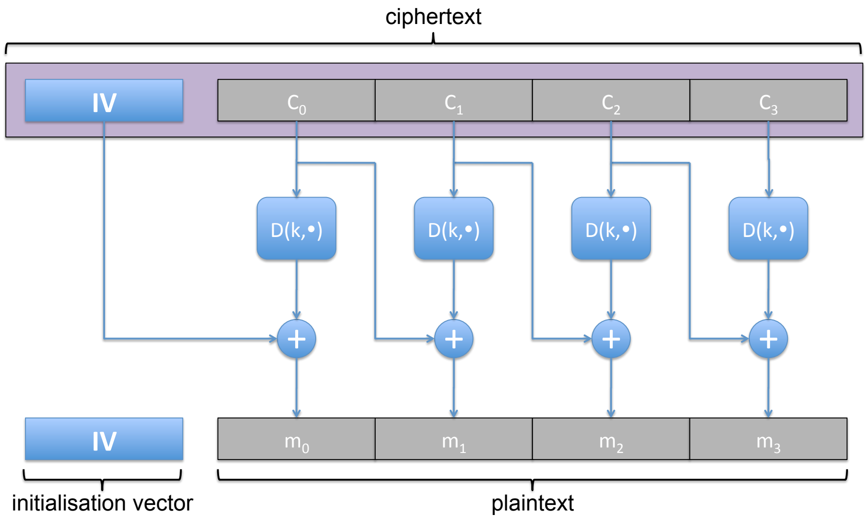
Cipher-block chaining (CBC) mode: encryption

(E, D) a block cipher that manipulates blocks of size ℓ.



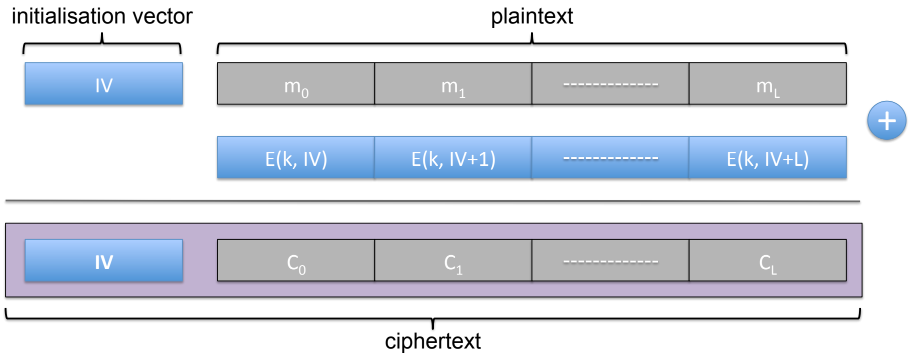
IV chosen at random in $\{0, 1\}^\ell$

Cipher-block chaining (CBC) mode: decryption



Counter (CTR) mode

(E, D) a block cipher that manipulates blocks of size ℓ.



IV chosen at random in $\{0, 1\}^{\ell}$