CS 578 – Cryptography

Prof. Michael Backes

Attacks on Block Ciphers, Modes of Operations

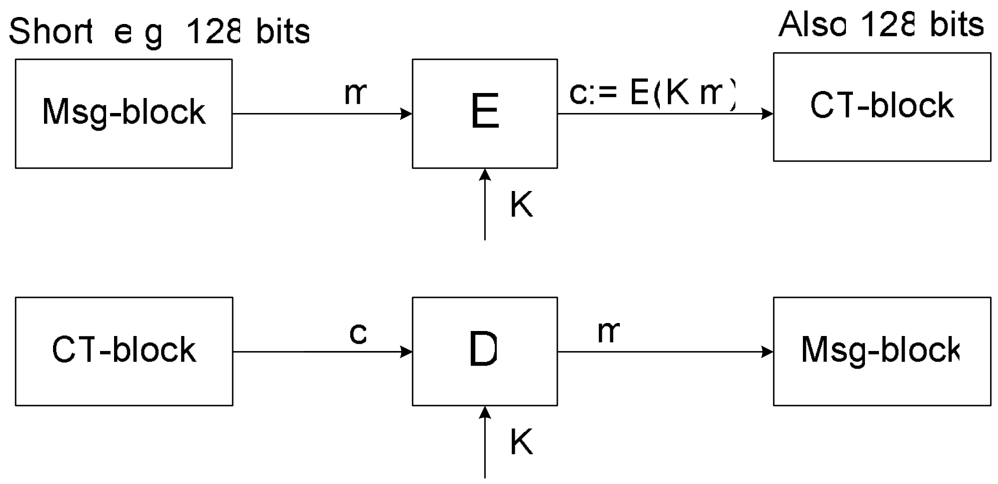
May 2, 2006

Administrative Announcements

- Handouts today:
  - Lecture notes, next exercise sheet
- Practical classes:
  - Start tomorrow, several requests for changes, …
- Quizzes:
  - Start tomorrow, last 15 min.
  - Quizzes written in English
  - Tomorrow's quiz on Lectures 1 + 2
- Discussion board
  - Please register as announcements on the course/exercises/quizzes, etc. will be given there
  - http://infsec.cs.uni-sb.de/wbb2/

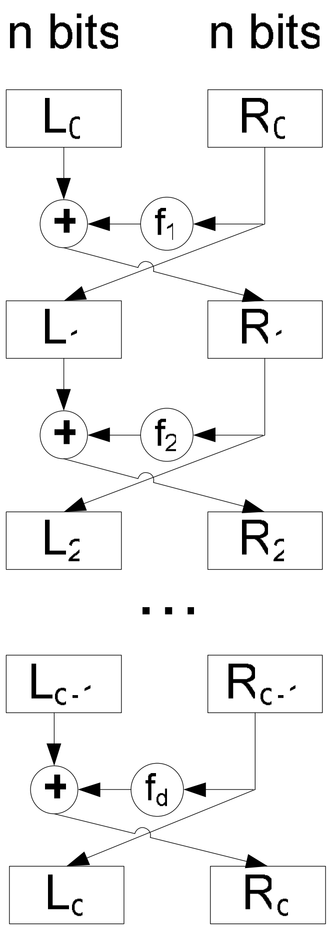
Recall: Stream- and block ciphers

- Ciphers as pair (E,D) of algorithms defined over $(\mathcal{K},\mathcal{M},\mathcal{C})$ such that for all K,m: D(K,E(K,m)) = m.
- Stream Ciphers (PRG): RC4, CSS (bad),…
- Block Ciphers:
  - DES, IDEA, … (Feistel-based)
  - AES, …(not Feistel-based)

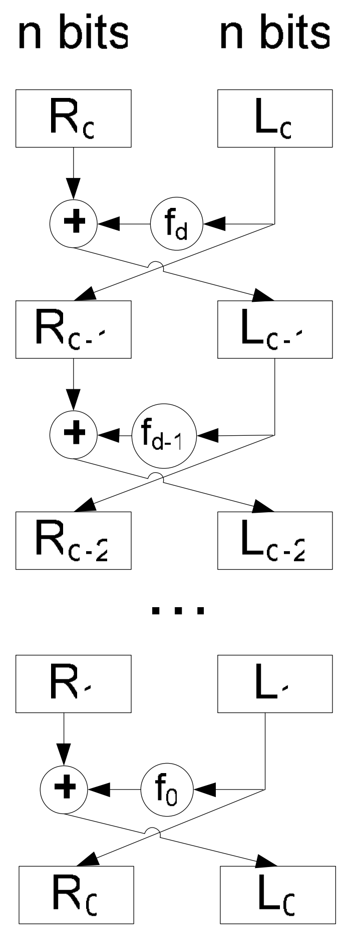


Recall: Feistel Networks

Encryption

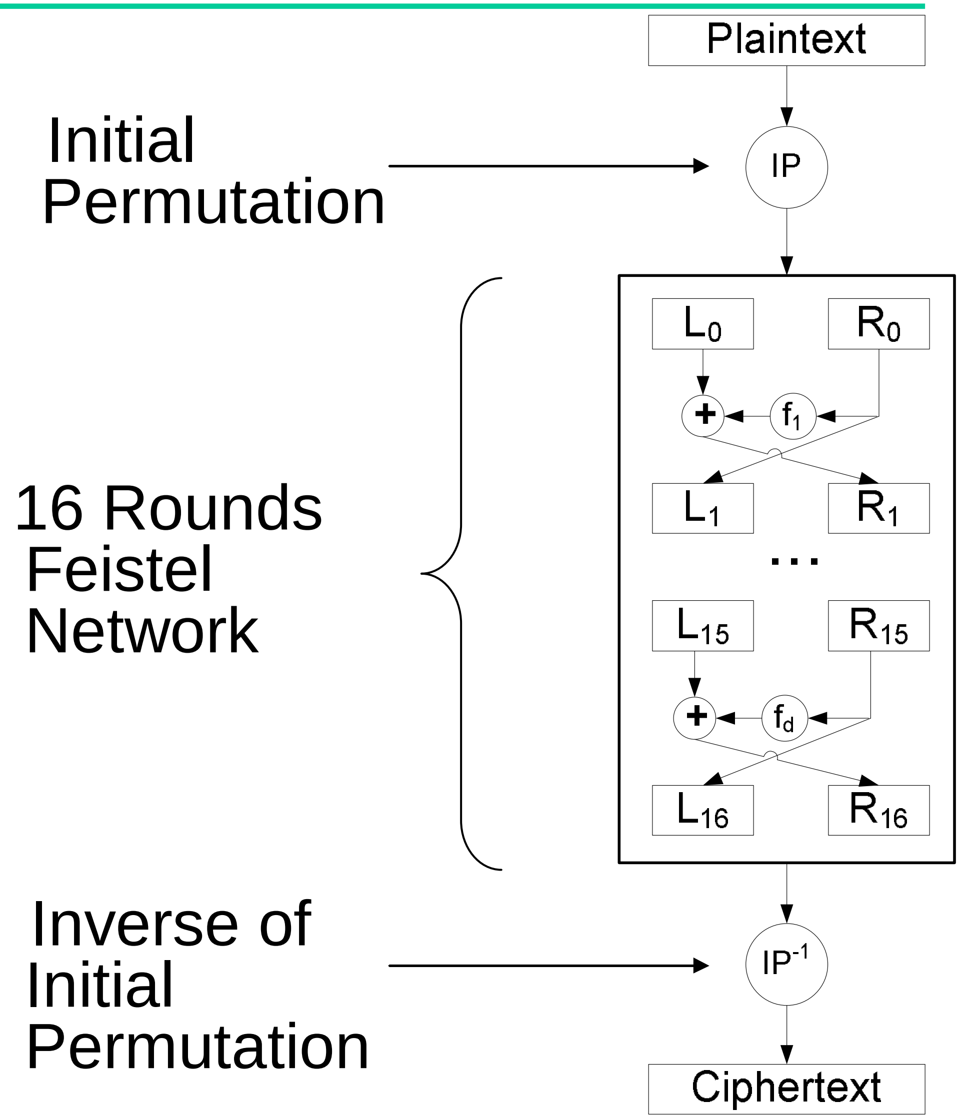


Decryption

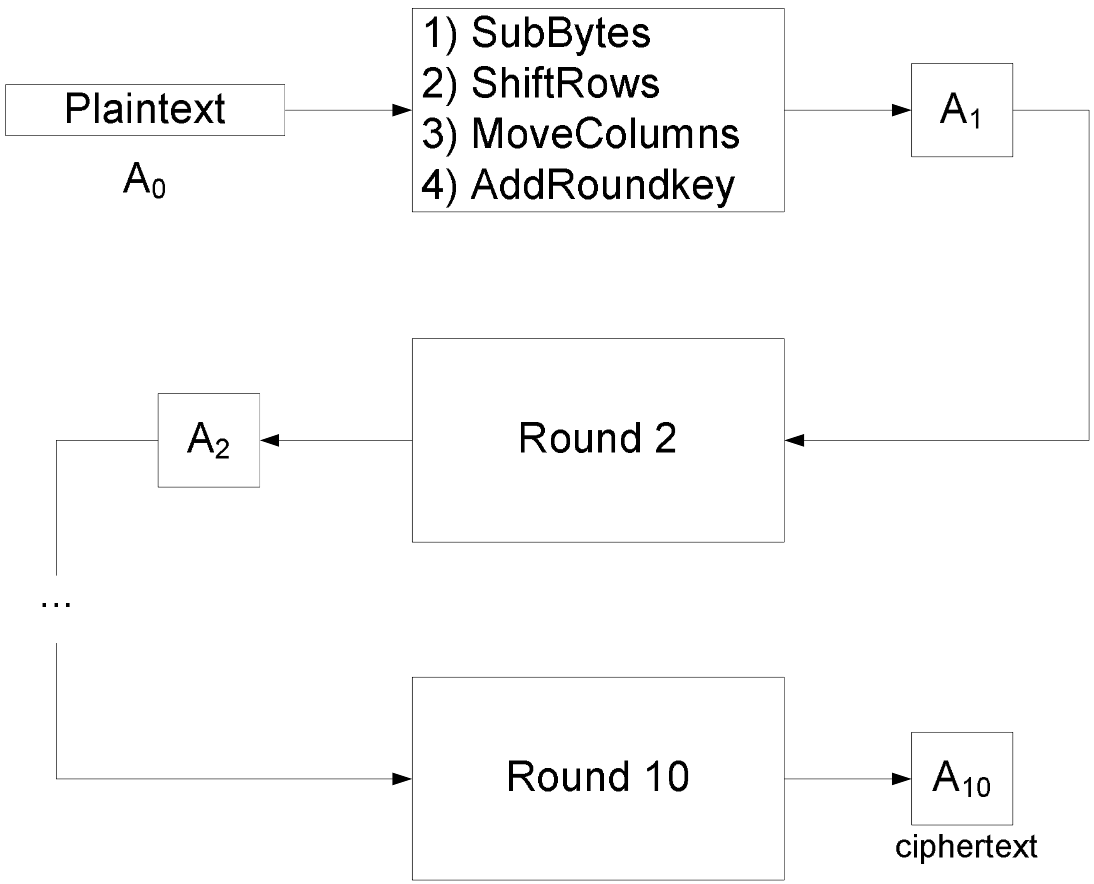


Recall: DES

- DES: 16-round Feistel Network:
  - $f_1,\ldots,f_{16}$: $\{0,1\}^{32} \rightarrow \{0,1\}^{32}$



Recall: AES


Plaintext
A0
1) SubBytes
2) ShiftRows
3) MoveColumns
4) AddRoundkey
A1
Round 2
A2
...
Round 10
A10
ciphertext

Recall: DES and AES Parameters

- DES: n (block-length) = 64 bits, k = 56 bits
- AES: n = 128 bits, k = 128, 192, 256 bits
- AES much faster than DES (AES is software-tailored)
- Only for small blocks! Encrypting large messages requires specific way of combining message blocks (modes of operation, today)

Performance of DES and AES

Crypto++ 5.2.1 Benchmarks [by Wei Dei]

	Algorithm	Megabytes(2^20 bytes) Processed	Time Taken	MB/Second
Stream ciphers	**RC4**	512	4.517	113.350
	SEAL	1.024	3.485	293.831
	BBS 512	0.25	4.096	0.070
Block ciphers	**DES**	128	5.998	21.340
	DES-X	128	6.159	20.783
	3-DES	64	6.499	9.848
	IDEA	64	3.375	18.963
	Rijndael (128-bit key)	256	4.196	61.010
	Rijndael (192-bit key)	256	4.817	53.145
	Rijndael (256-bit key)	256	5.308	48.229

Exhaustive Search Attacks

- Most simple attack conceivable
- Given:
  - a few PT/CT pairs (m_1,c_1), (m_2,c_2), …, i.e., $c_i = E(K,m_i)$ for i=1,2,… and m_i random elements from $\{0,1\}^n$
- Goal: Total break, i.e., find K such that $c_i = E(K,m_i)$ for all i.
- Note: No stream ciphers would resist this setting: multiple encryptions with the same key!

Exhaustive Search Attacks for DES

- How many PT/CT pairs until K is uniquely determined?
- Theorem: For DES, given one random PT/CT pair (m,c), there is a unique K such that E(K,m)=c with very high prob. (≥ 1- 1/256).
- "Proof" (only heuristic by idealizing DES into an ideal cipher: collection of 2^{56} random permutations on $\{0,1\}^{64}$; done in all proofs of block ciphers):

  [on the board]

- Consequence: Exhaustive search is possible on DES given only one PT/CT pair

DES Challenge

- Exhaustive Search Challenge set by RSA Security
- msg = “The unknown message is: ----------”
- CT = c_1 c_2 c_3 c_4 c_5
- Originally 10.000$ for solving this challenge

DES Challenge (cont'd)

- 1997: Internet search: 3 month
- 1998: EFF (3 days), spent 250K$
- 1999: 22 hours

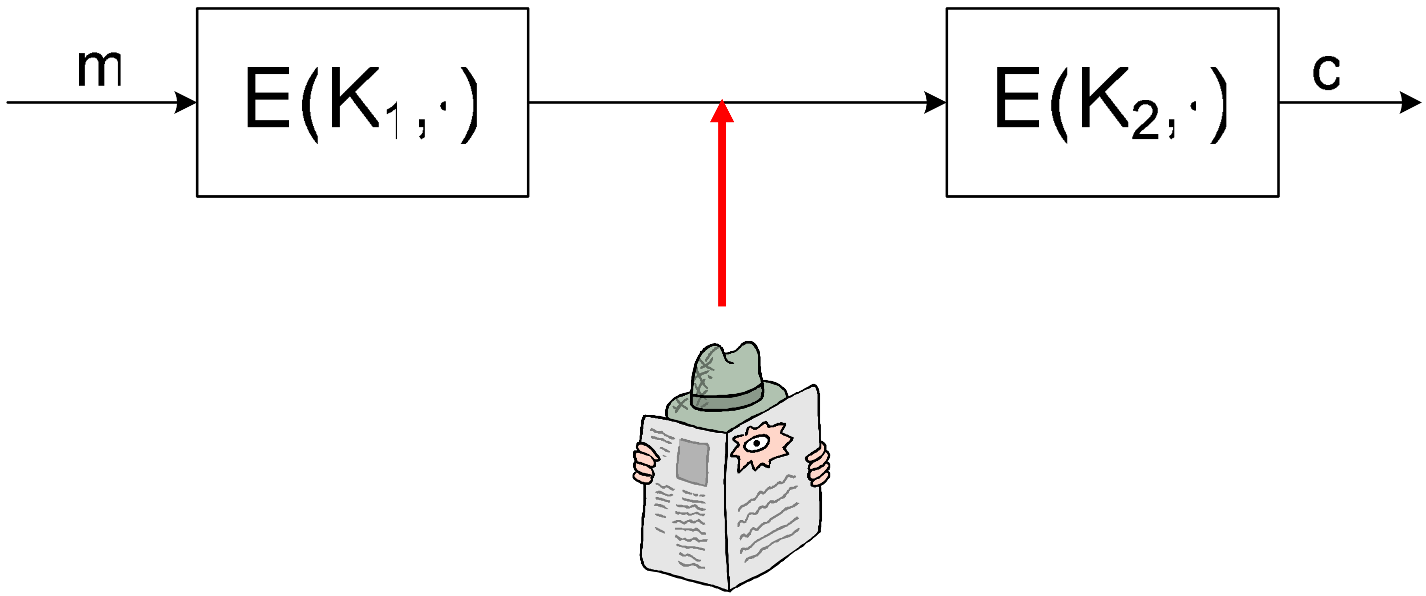
- For 128 bits AES:
  time = 2^{128-56} * time(DES) $\approx 10^{24}$ days

Some ways of saving DES: Triple DES

- Avoiding Exhaustive Search: Triple DES (3DES)
- General Method: Let (E,D) be a cipher
  - Let $TE((K_1,K_2,K_3), m) := E(K_1, D(K_2, E(K_3,m)))$
- Why not 3 times E? → backwards compatibility
- Problem: 3 times slower than E
- Key size: 3*56 = 168 bits

Why not Double DES (2DES)?

- $DE((K_1,K_2), m) := E(K_1, E(K_2, m))$
- Attack by “meet-in-the-middle”



Meet-in-the-middle Attack

- Given PT/CT pair (m,c), $c = E(K_1, E(K_2, m))$

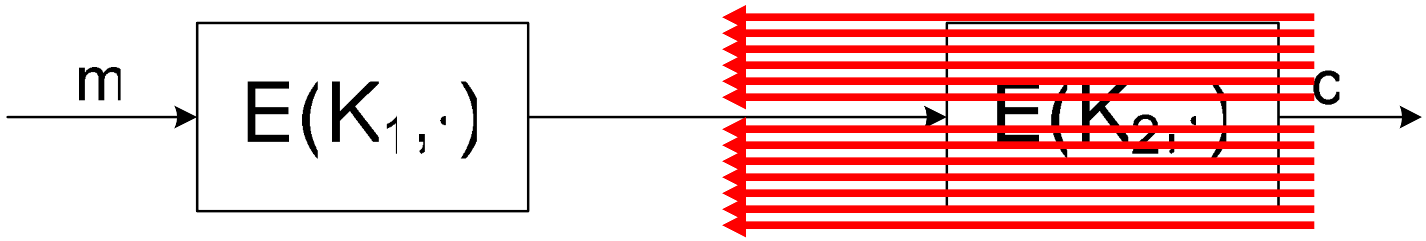
1. Set up the following table:

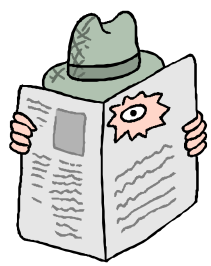
K_1^1	$D(K_1^1, c)$
K_1^2	$D(K_1^2, c)$
K_1^3	$D(K_1^3, c)$
….	….
$K_1^{2^{56}}$	$D(K_1^{2^{56}}, c)$

  - Takes time 2^{56}
  - Then sort right column of the table

Why not Double DES (2DES)?

- $DE((K_1,K_2), m) := E(K_1, E(K_2, m))$
- Attack by “meet-in-the-middle”





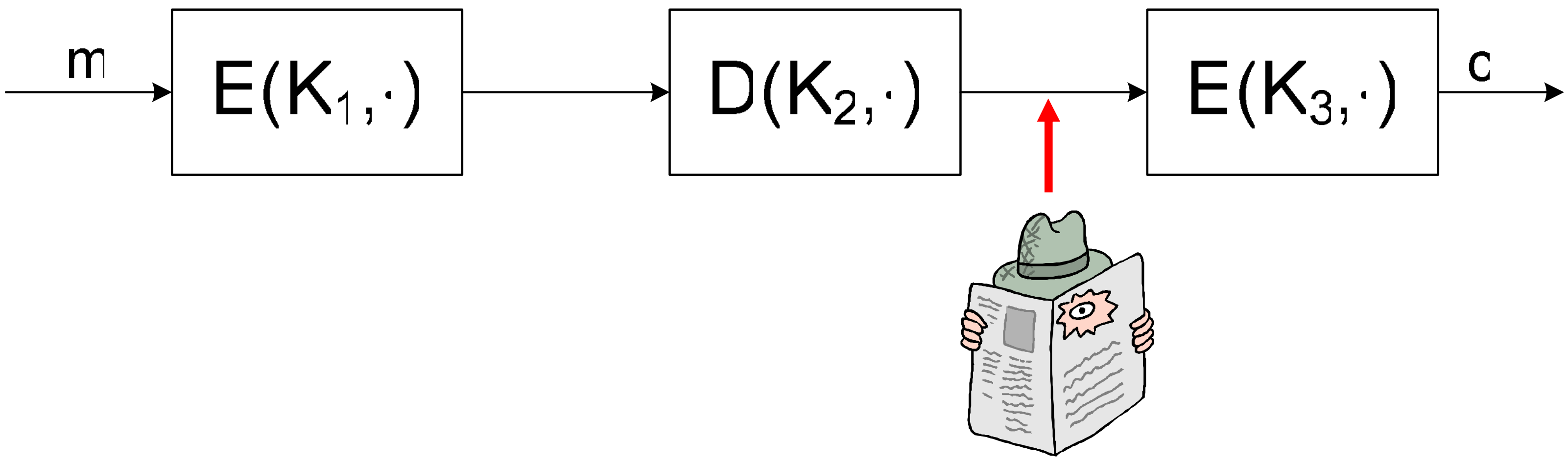
Meet-in-the-middle (cont'd)

2. For each K of $\{0,1\}^{56}$:
   - Test if E(K,m) is in the right column of the table
   - If in column, then E(K,m) = $D(K_1^j, c)$ for some j
     → Key = (K, K_1^j)

- Total time for exhaustive search (ignoring log-factors): $2^{56} + 2^{56} = 2^{57}$
- Effective key length less than 57 bits

Meet-in-the-Middle on 3-DES

- Can we do meet-in-the-middle for 3-DES?



- Time for meet-in-the-middle on 3-DES: 2^{112}
- Effective key length of 3-DES ≤ 112 bits

Two-key Triple DES

- Only 112 bits effective key length → can we get away with shorter keys initially?
- General Method: Let (E,D) be a cipher
  - Let $TE((K_1,K_2,K_3), m) := E(K_1, D(K_2, E(K_1,m)))$
- Standard considers this a suitable option
- Problem: Only as good as DES…

Another ways of saving DES: DESX

- Getting better effective key length: DESX
- General Method: Let (E,D) be a cipher
  - $EX((K_1,K_2,K_3), m) := K_1 \oplus E(K_2, m \oplus K_3)$
- Key length = 64 + 56 + 64 = 184 bits
- As fast as DES!
- Theorem (Kilian & Rogaway ’98): If E is an ideal cipher, then effectivekeylen(EX) ≥ keysize - blocksize -1
- Effective key length of DESX ≥ 119 bits (equality because of meet-in-the-middle)

Sophisticated Attacks on BC

1. Linear and differential cryptanalysis

- Basic idea of linear cryptanalysis:
  Suppose for random m, K and c = E(K,m):
  $$\Pr[\underbrace{m_{i1} \oplus m_{i2} \oplus \ldots \oplus m_{ir}}_{\text{r bits of msg}} \oplus \underbrace{c_{j1} \oplus \ldots \oplus c_{jv}}_{\text{v bits of CT}} \oplus$$
  $$\underbrace{K_{l1} \oplus \ldots \oplus K_{lu}}_{\text{u bits of key}} = 1] \geq ½ + \varepsilon$$

- E.g., the 5^{th} S-box of DES has bias $\varepsilon = 2^{-21}$

Sophisticated Attacks on BC

1. Linear and differential cryptanalysis

- Basic idea of linear cryptanalysis:
  - Suppose for random m, K and c = E(K,m):
    $\Pr[m_{i1} \oplus m_{i2} \oplus \ldots \oplus m_{ir} \oplus c_{j1} \oplus \ldots \oplus c_{jv} \oplus K_{l1} \oplus \ldots \oplus K_{lu} = 1] \geq ½ + \varepsilon$
    (holds for DES with $\varepsilon = 2^{-21}$)
  - Then it holds:
    $\Pr[m_{i1} \oplus m_{i2} \oplus \ldots \oplus m_{ir} \oplus c_{j1} \oplus \ldots \oplus c_{jv} = K_{l1} \oplus \ldots \oplus K_{lu}]$
    $\geq ½ + \varepsilon$
- Theorem: Given $1/\varepsilon^2$ PT/CT pairs. Then
  $K_{l1} \oplus \ldots \oplus K_{lu} = MAJ_{PT/CT} [m_{i1} \oplus m_{i2} \oplus \ldots \oplus m_{ir} \oplus c_{j1} \oplus \ldots \oplus c_{jv}]$
  will hold with probability ≥ 97.7%

Linear Cryptanalysis on DES

- For DES: $\varepsilon = 2^{-21}$
- Given $1/\varepsilon^2 = 2^{42}$ PT/CT pairs, we get $K_{l1} \oplus \ldots \oplus K_{lu}$
- In the same way, we can deduce 14 “bits” of the key using various other relations
- Then exhaustive search on the remaining $2^{56}/2^{14} = 2^{42}$ bits
- Time needed:
  - 2^{42} steps for using linearity to deduce 14 bits
  - 2^{42} steps for exhaustive search on remaining key space

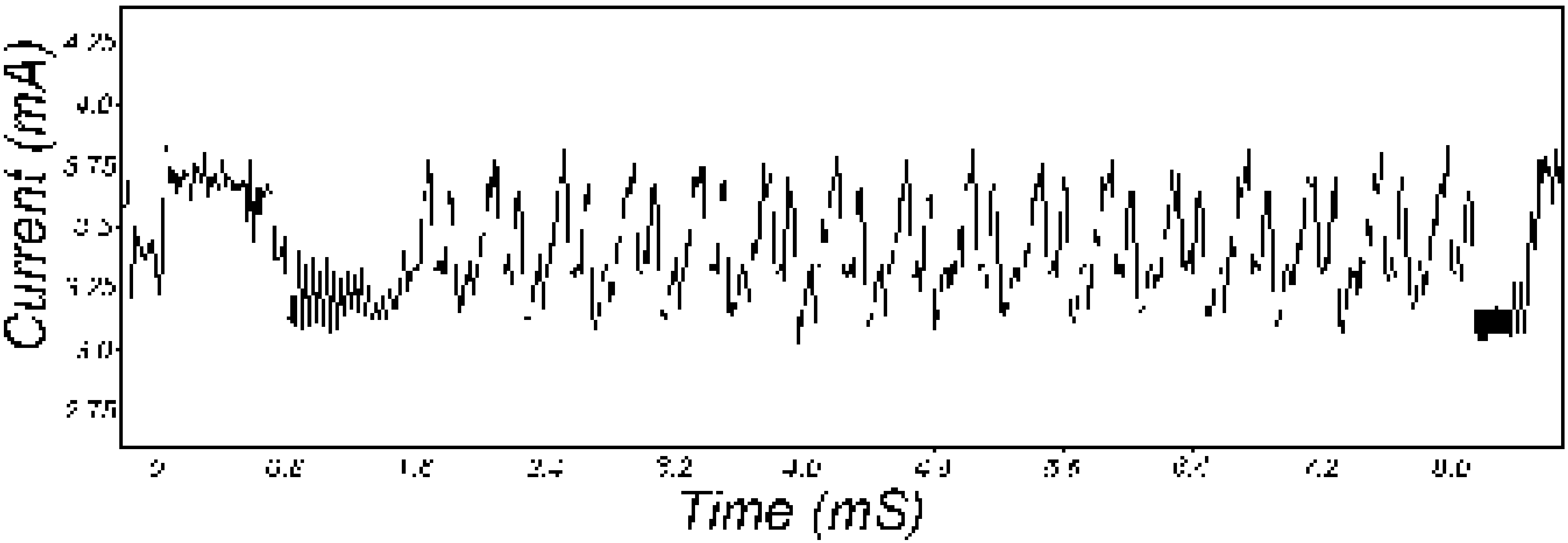
  → 2^{43} steps total
- Conclusion: Don’t design block ciphers yourself!

Sophisticated Attacks on BC (cont’d)

2. Implementation attack (side channel attack)

- Power cryptanalysis

Power-Consumption of DES



Sophisticated Attacks on BC (cont'd)

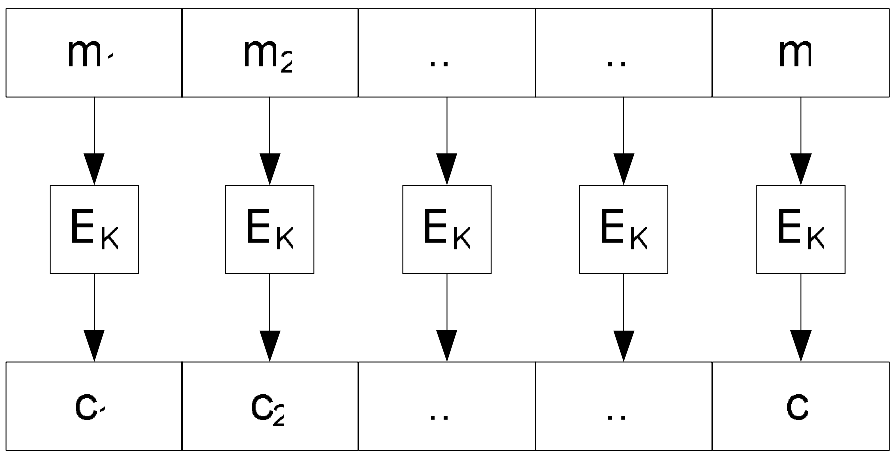
2. Implementation attack (side channel attack)

- Power cryptanalysis
- Electromagnetic emanation
- Timing
- Sound

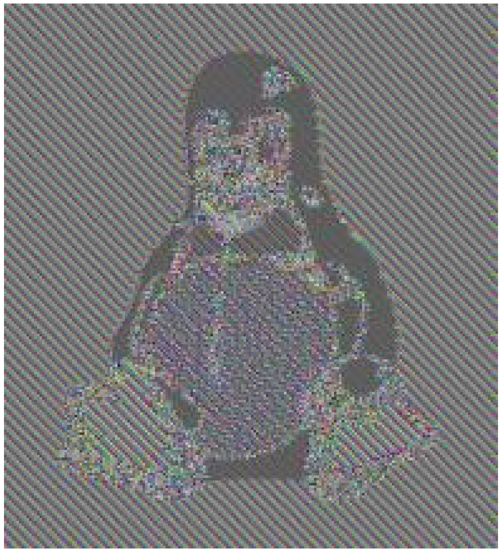
→ Do not even implement ciphers!

Outlook: How to use Block Ciphers

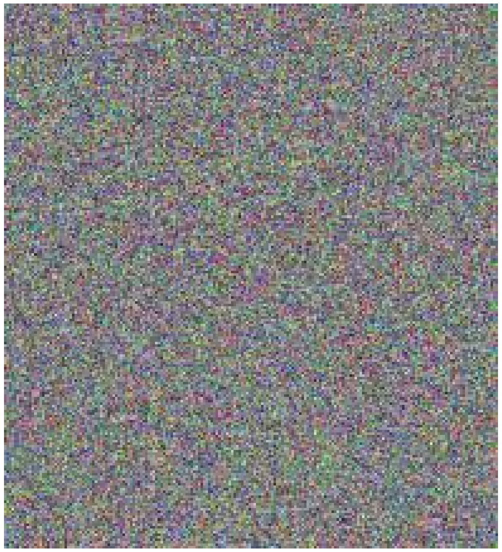
- Construction 1: Electronic Codebook (ECB)
  - Intuitive but naïve way (how not to do it)



ECB Reveals Patterns



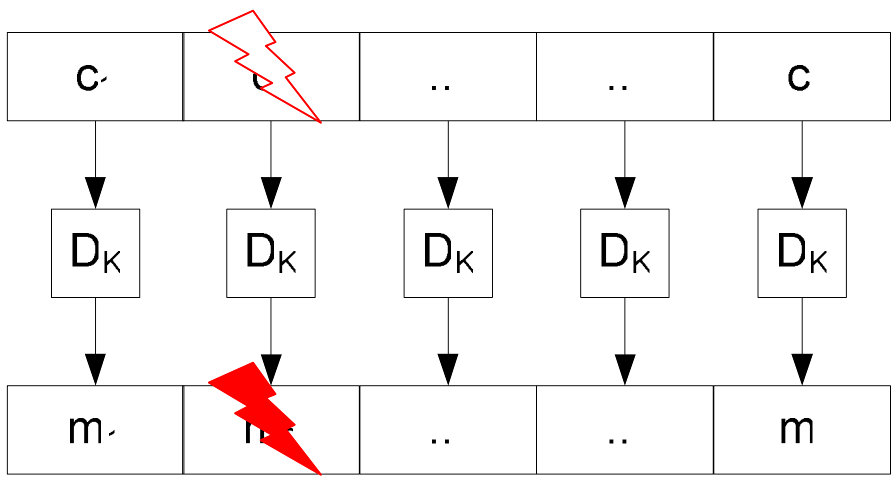
ECB Encryption

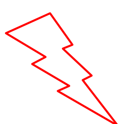


Other mode of operation

Outlook: How to use Block Ciphers

- Construction 1: Electronic Codebook (ECB)
  - Intuitive but naïve way (how not to do it)
  - At least self-synchronizing (if block length are tolerated)

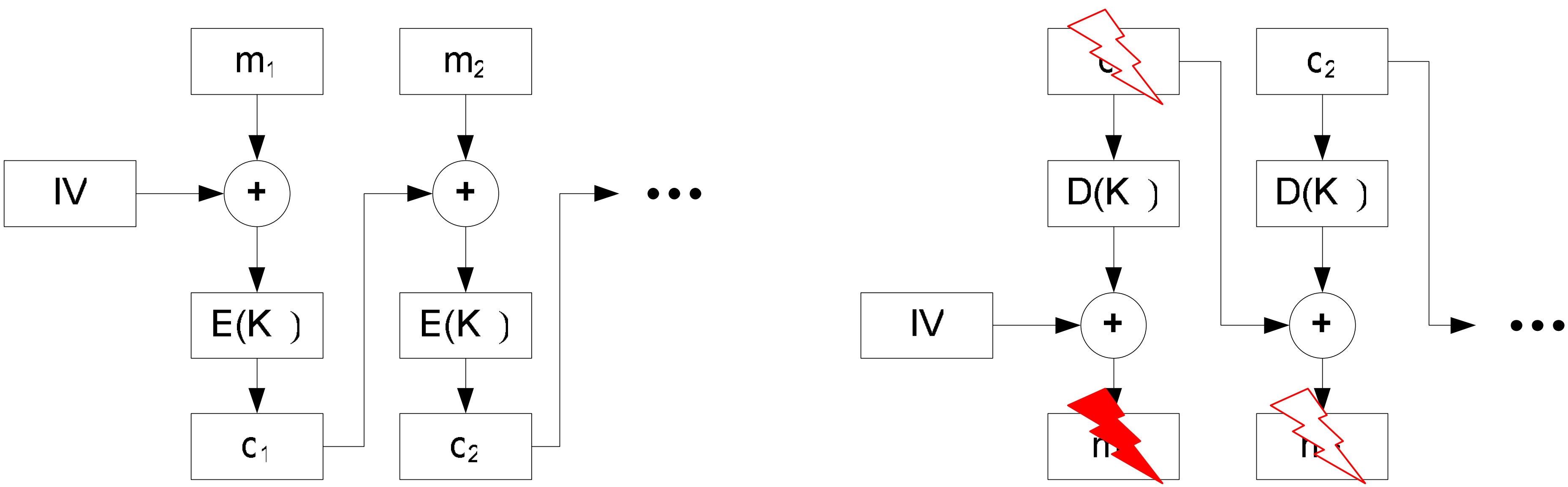


 = Failure of 1 bit

 = Failure of complete block

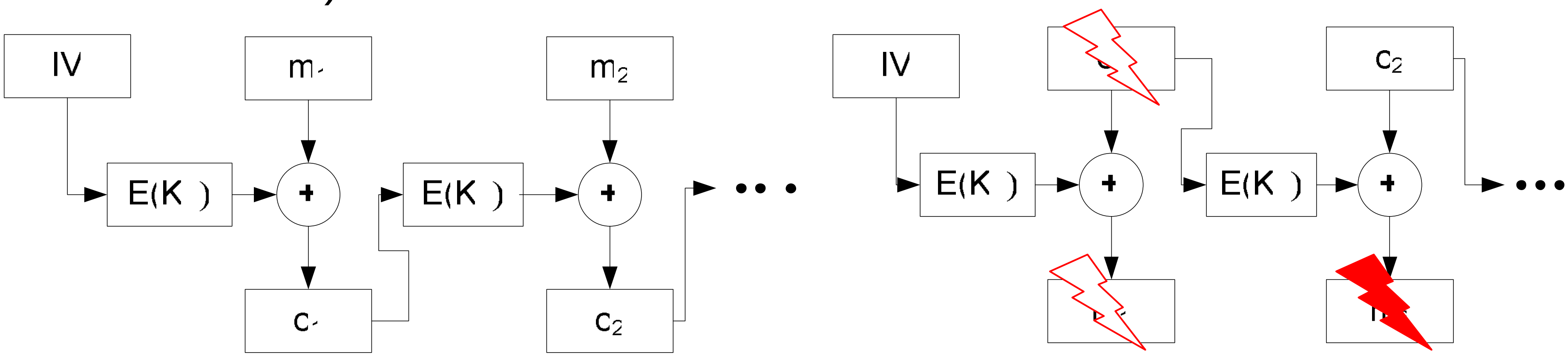
Outlook: How to use Block Ciphers

- Construction 2: Cipherblock Chaining (CBC)
  - Very often used, but some problems:
    Sequential, no integrity for ciphertexts (next week)
  - Self-synchronizing after two blocks (if block length ok)



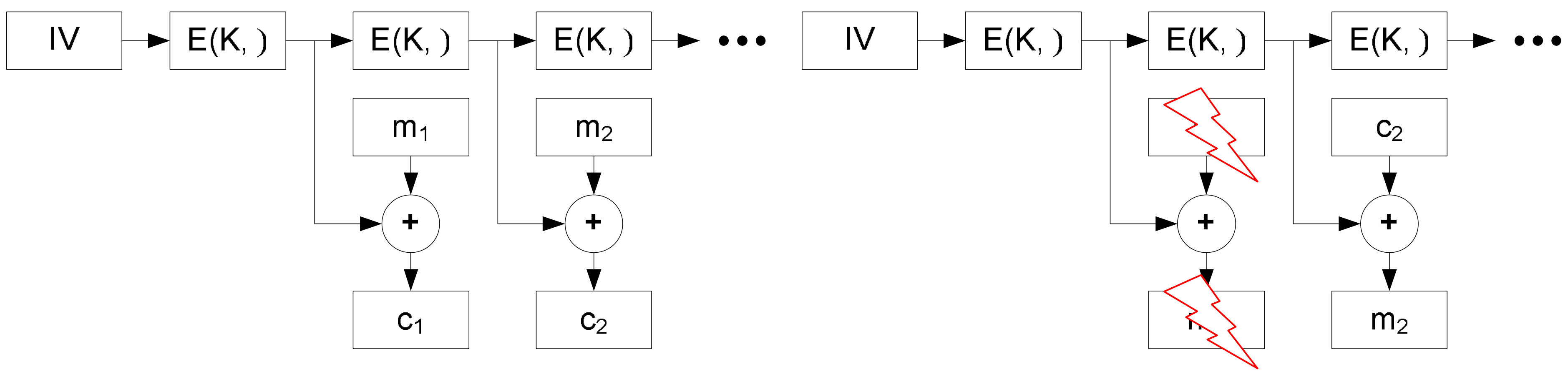
Outlook: How to use Block Ciphers

- Construction 3: Cipher Feedback (CFB)
  - CFB similar to stream ciphers
  - Note: No need for decryption here
  - Also self-synchronizing after two blocks (if block length ok)



Outlook: How to use Block Ciphers

- Construction 4: Output Feedback (OFB)
  - OFB similar to stream ciphers as well
  - Note: No need for decryption here
  - Strongly self-synchonizing (but loss of block border dramatical)



Outlook: How to use Block Ciphers

- Construction 5: Countermode (CTR)
  - Countermode also similar to stream ciphers
  - Note: No need for decryption here
  - Later: Better security than CBC