Cryptology
Lecture 3
DES:
Data Encryption Standard

Joseph Phillips

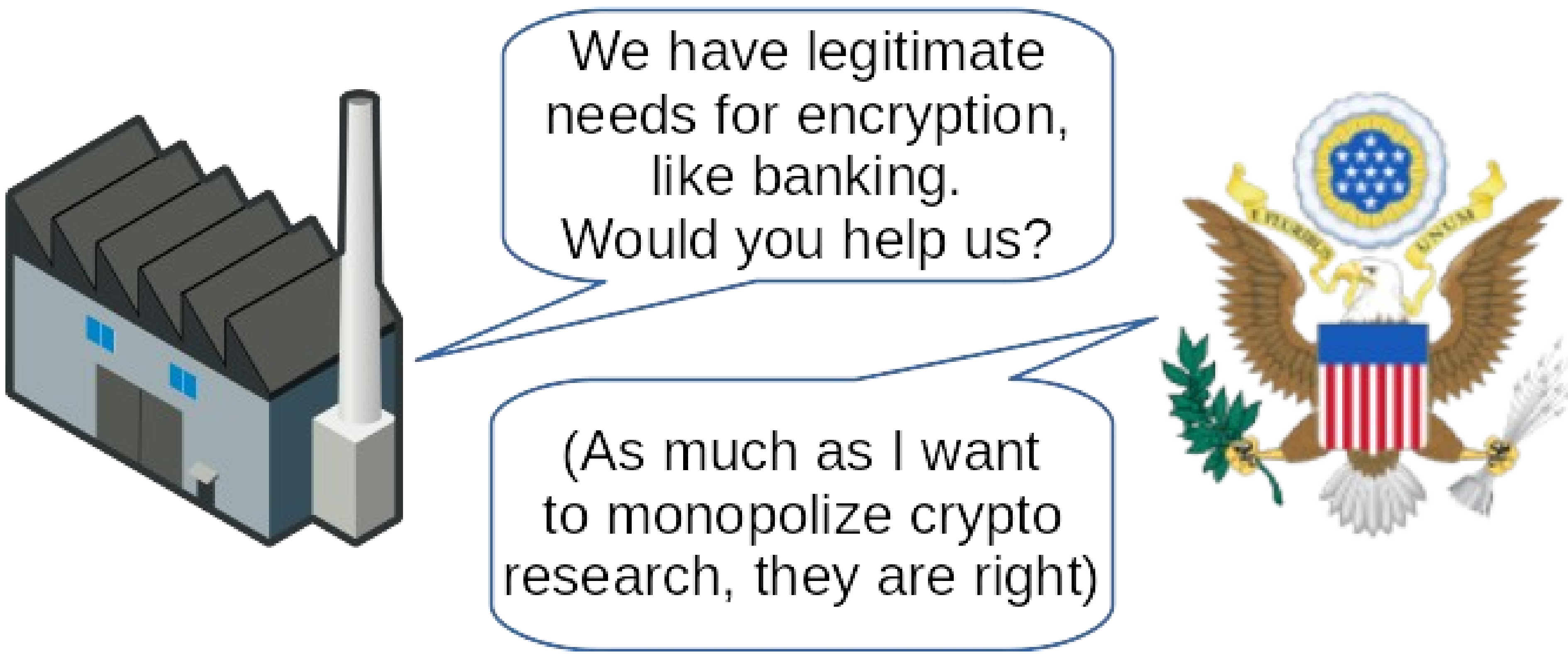
Last modified 2020 April 13

Topics

- History
- Design Motivation
- Details
- Decryption
- Attacks
- Alternatives

- Reading: “*Chapter 3: The Data Encryption Standard (DES) and Alternatives*” of Christof Paar and Jan Pelzl “*Understanding Cryptolography: A Textbook for Students and Practitioners*”

History: Early 1970s
US industry wants crypto



History: Reluctantly, US Govt agrees But does not want to give away the good stuff! (NIST then called NBS)

You boys take charge of this, we don't want the NSA to give away strong crypto techniques!

NIST

Okay.

History: So NBS asks industry what they have

NIST

So industry,
what do you have?

IBM

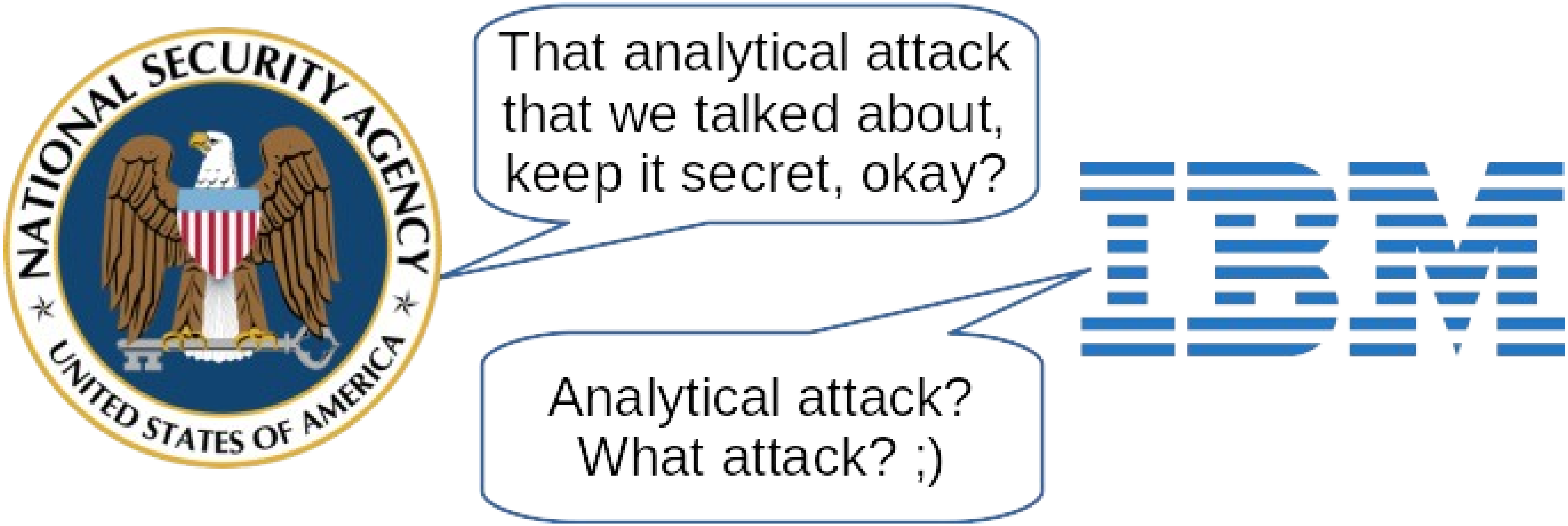
Crypto? We’ve been
working on that!
Check out Lucifer!

History: NBS asks NSA about it

Crypto is your specialty what do you think of IBM’s Lucifer?

Let’s make it stronger against analytical attack, but weaker against brute-force

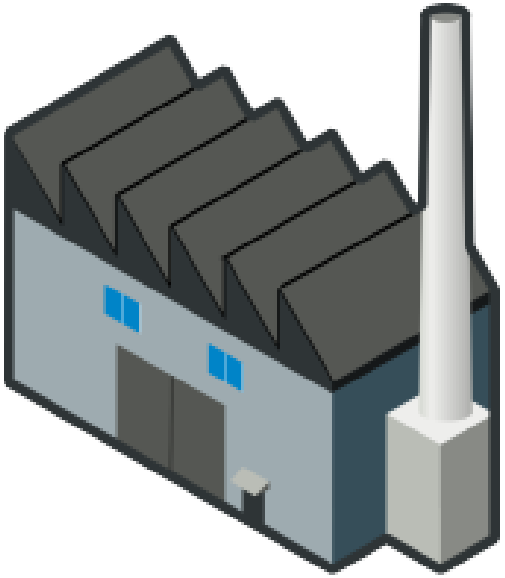
History: Stronger S-boxes but smaller key (128 bits => 56)



History: 1977 D.E.S. (Data Encryption Standard) introduced

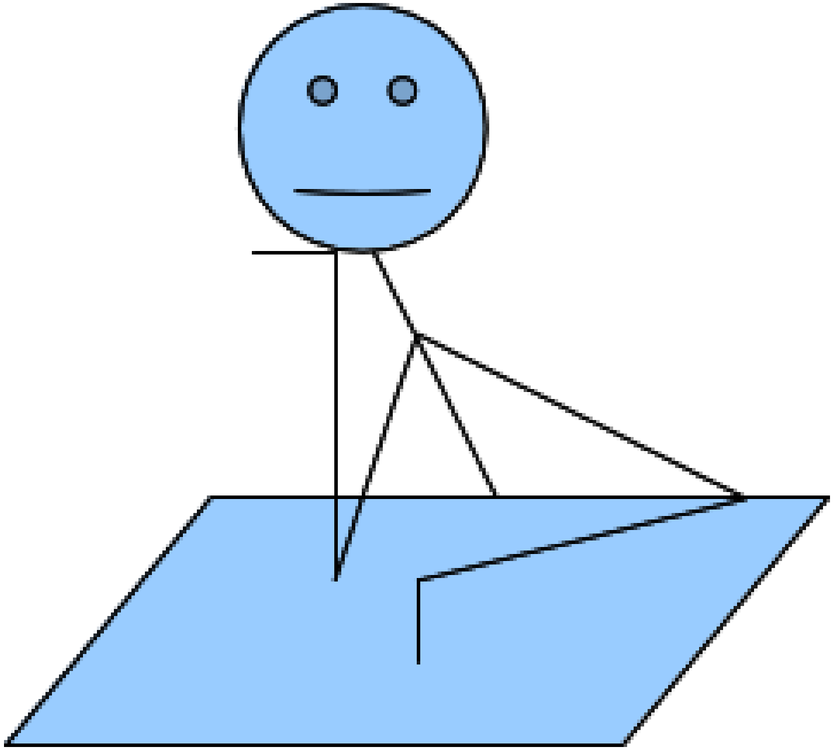
NIST

You want crypto?
Here it is! ***D.E.S.***
Data Encryption
Standard!



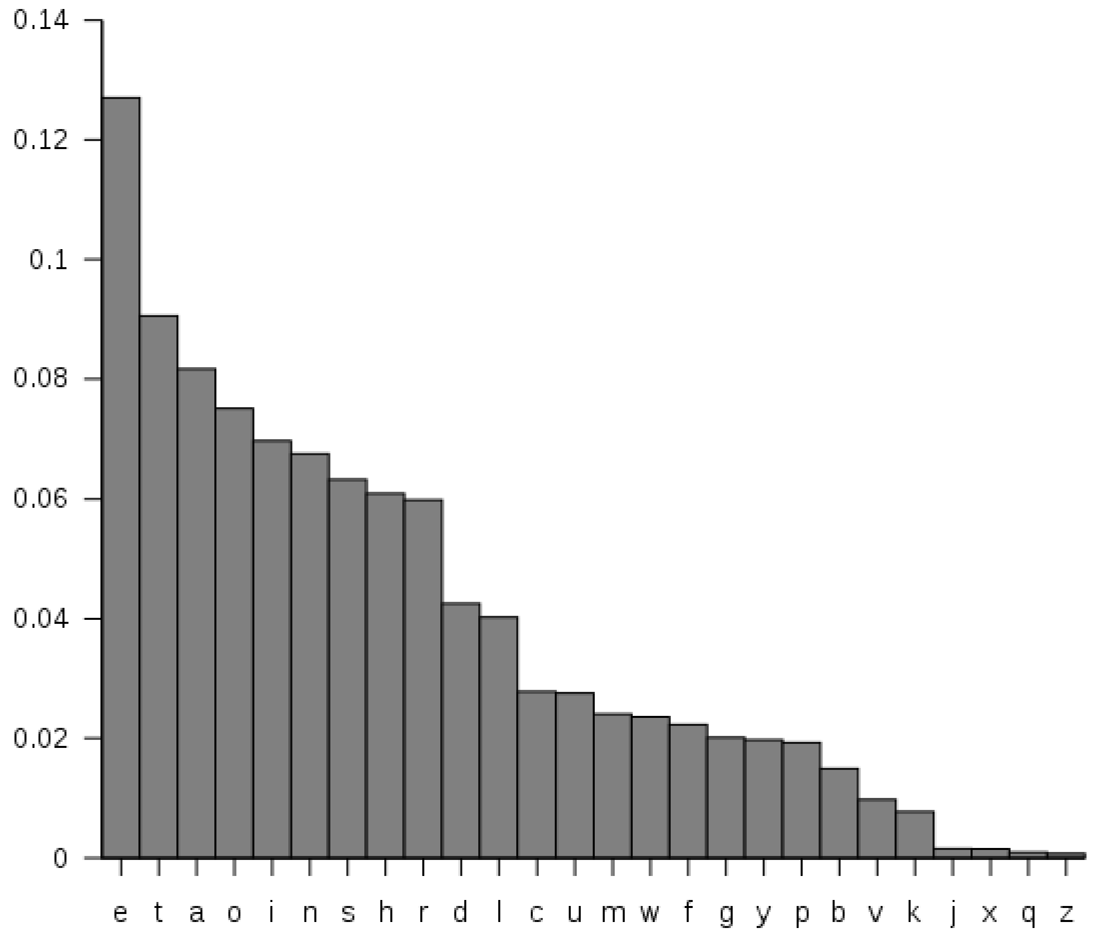
History: A Skeptical Public

"***Input from the super-secret United States <u>NSA</u>? There must be a <u>trick</u>!***"



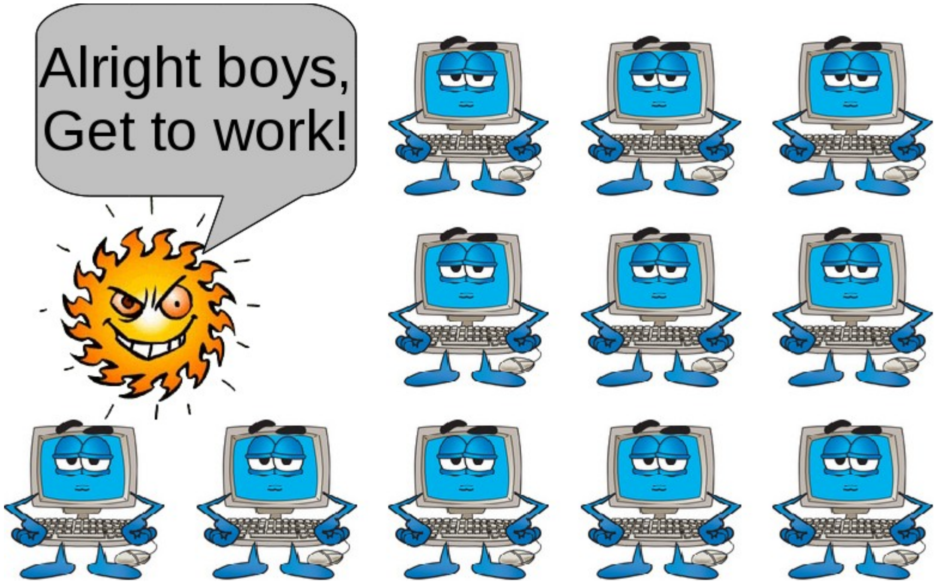
History: Skeptical Public, cont'd

"***A <u>trap-door</u> allowing them to easily decode!***"



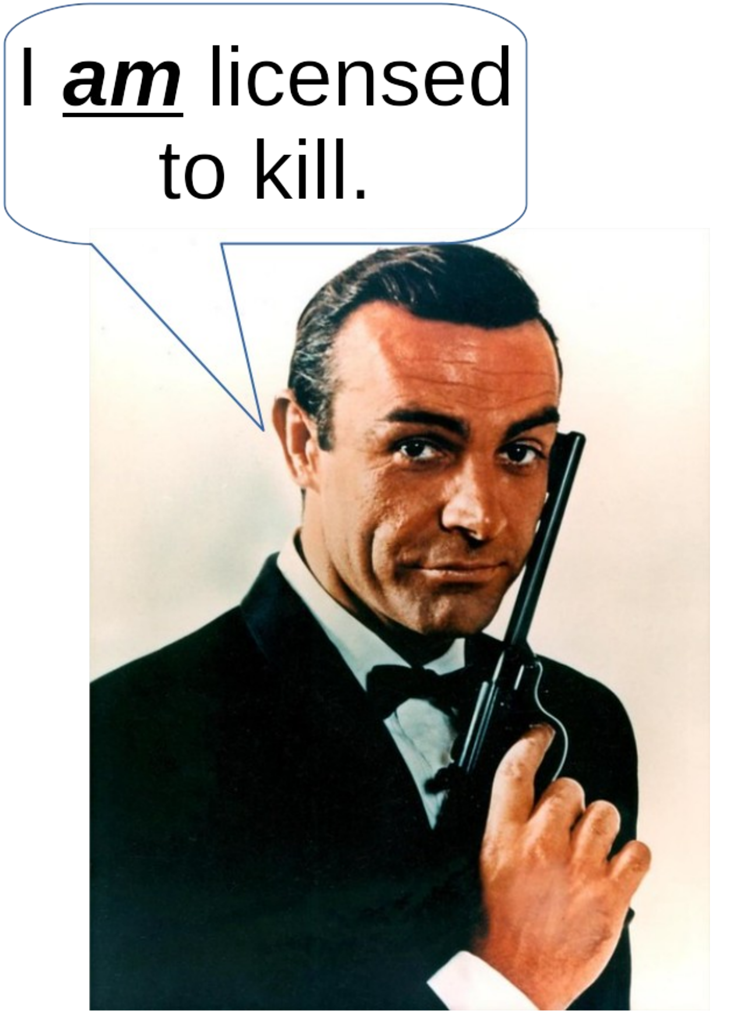
History: Skeptical Public, cont'd

"***Only 56 bits: they can <u>brute-force</u> attack it easier than original 128 bits***"



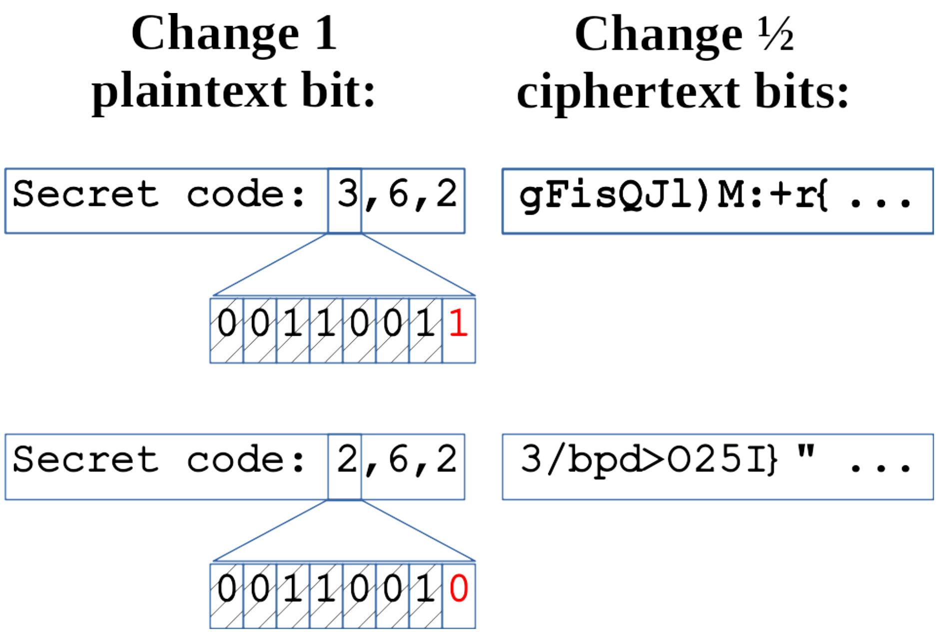
History: Analysis of D.E.S.

- And so people have looked for weaknesses
  - 1970s: birth of academic cryptology
- No serious ***analytical*** flaw published since, thus
  1. Truly as secure as they could make it, or
  2. US Gov't "disappeared" anyone who cracked it!
- (Nowadays everyone knows 56 bits can be brute-force attacked)



Design Motivation

- Desired criteria
  - On average, changing one plaintext bit changes ½ ciphertext bits
  - Difficult to statistically attack



Design Motivation

IQ IFCC VQQR FB RDQ VFLLCQ

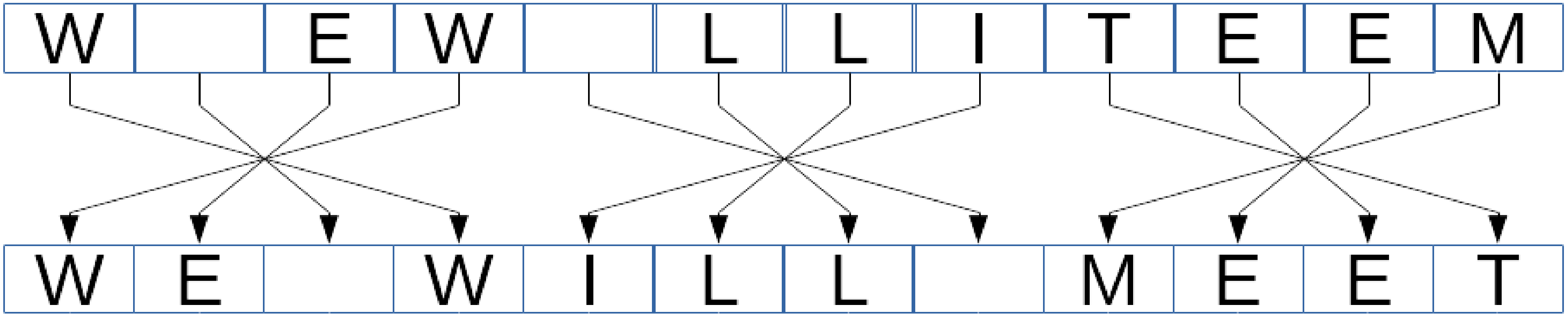
Q = *E*	C = *L*	H = *A*
R = *T*	D = *H*	I = *W*
F = *I*	B = *N*	N = *O*
V = *M*	L = *D*	A = *F*
W = *R*	J = *B*	Z = *S*
S = *G*	E = *S*	

WE WILL MEET IN THE MIDDLE

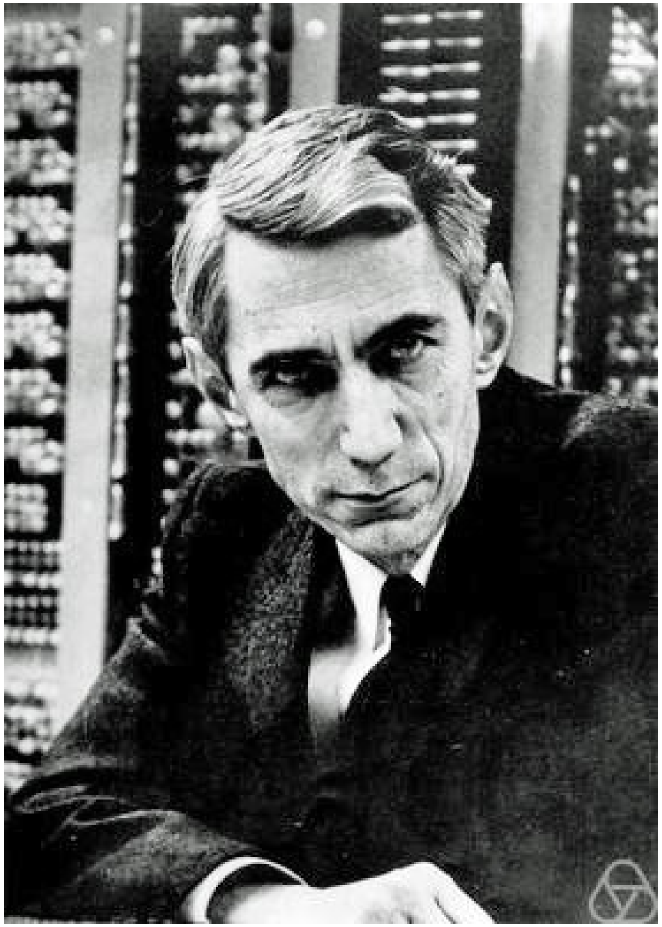
- Confusion
  - Obscure relationship between key and ciphertext
  - E.g. substitution
  - E.g. Shift and Affine ciphers
- Attack thru statistics!

Design Motivation

- Diffusion
  - Move bits around between plaintext and ciphertext
  - Still attack thru statistics!



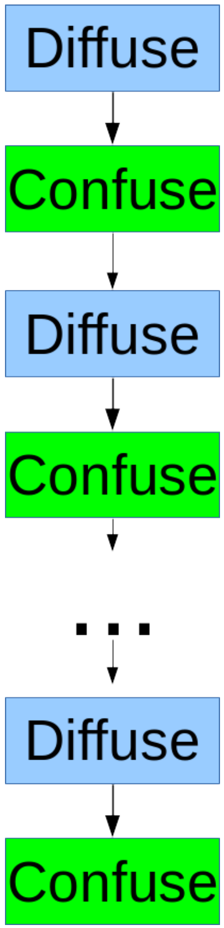
Design Motivation



- Claude Shannon
  - American mathematician, electrical engineer, cryptographer
  - Father of information theory
  - "***Dude, don't do one or the other. <u>Do them both!</u>***"
- Product ciphers:
  - do more than one type of encryption

Modern Product Ciphers

- Several rounds of
  - Diffuse
  - Confuse
  - Diffuse
  - Confuse
  - etc.

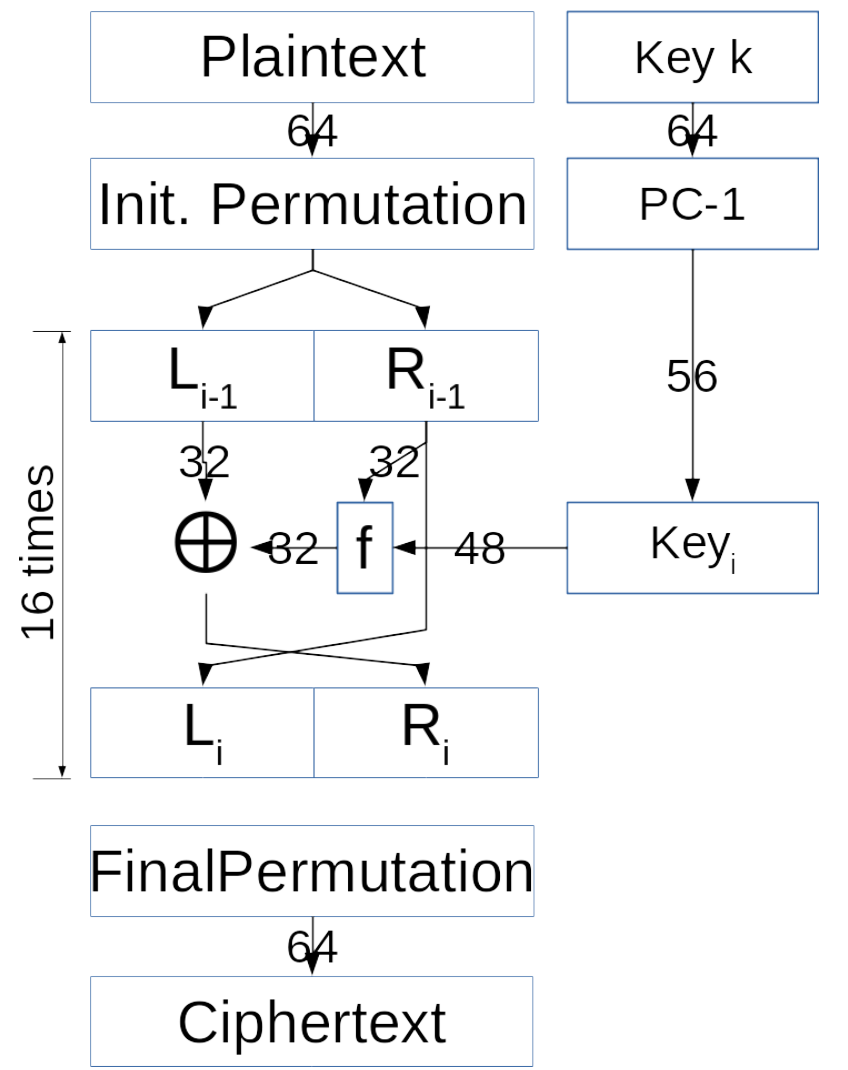


Horst Feistel

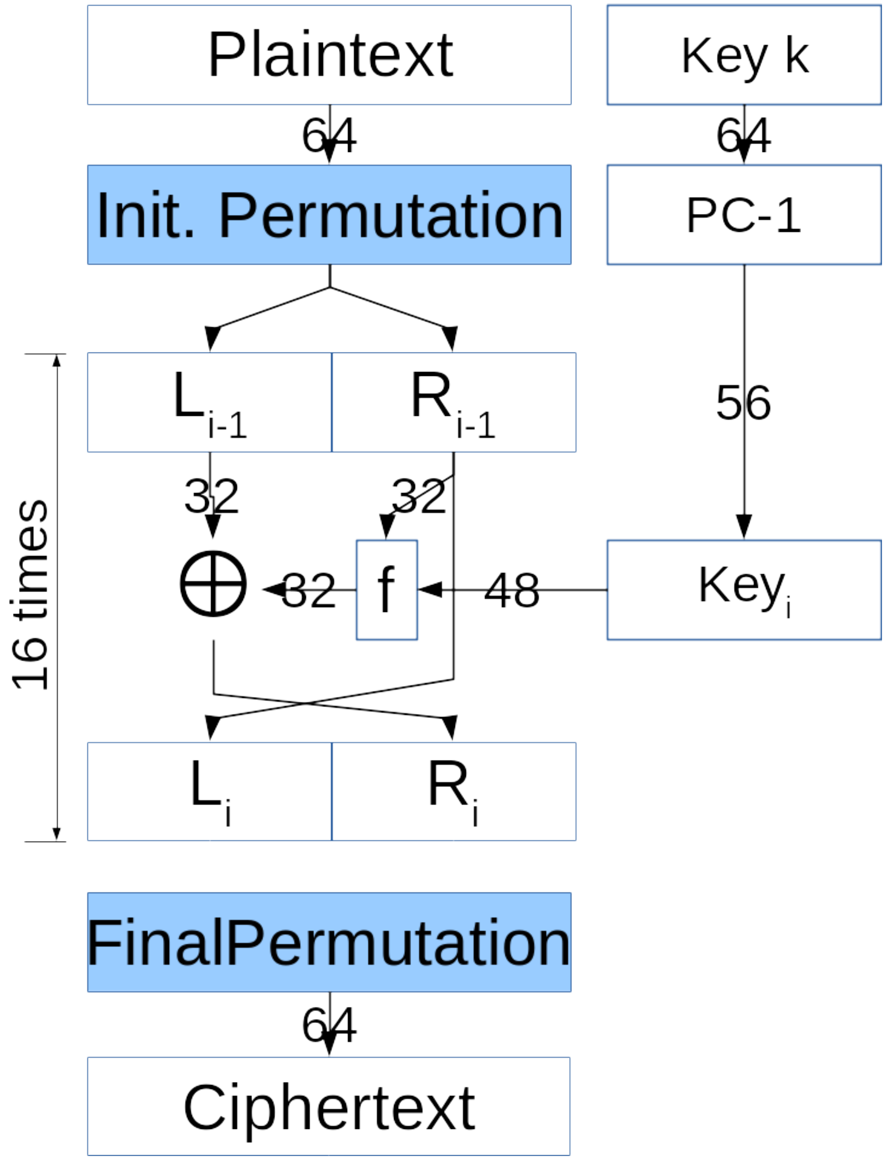
- German-American cryptographer
  - Fled Nazi Germany for USA in 1934
  - During WWII, under house arrest until 1944
  - Worked on Identification-Friend-or-Foe (IFF)
  - MIT → MITRE → IBM
- Worked on Lucifer at IBM

DES

- Initial permutation
- For i = 1 to 16:
  - Feistel Round
- Final permutation



Initial and final permutations

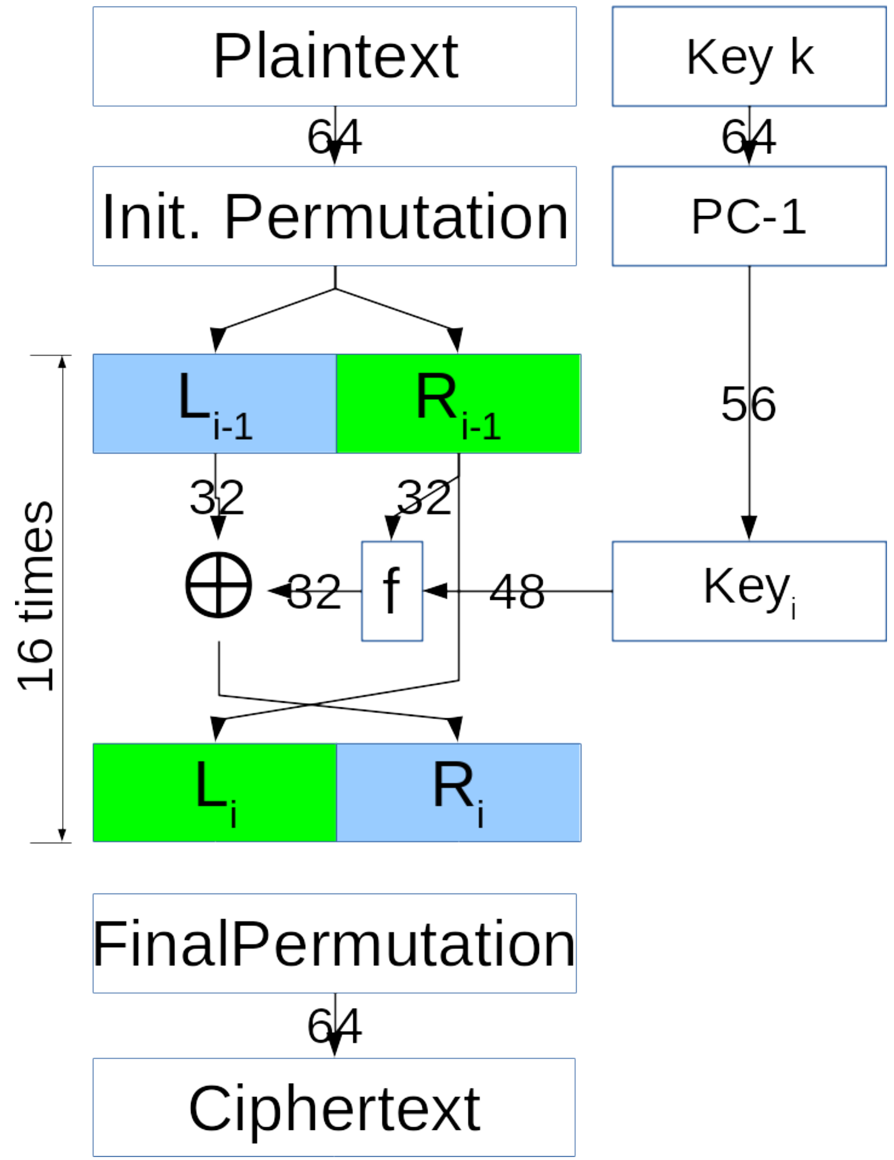


Initial and final permutations

- IP
  - Initial bit rewiring
- IP^{-1}
  - Final bit unrewiring
- Speed
  - Fast in hardware
  - Slow in software
- Author:
  - “Does not add security”
  - “Done to increase speed on 8-bit busses?”
- Joe
  - There might be a hardware reason
  - All even bits first
  - All odd bits last
  - Some regular patterns

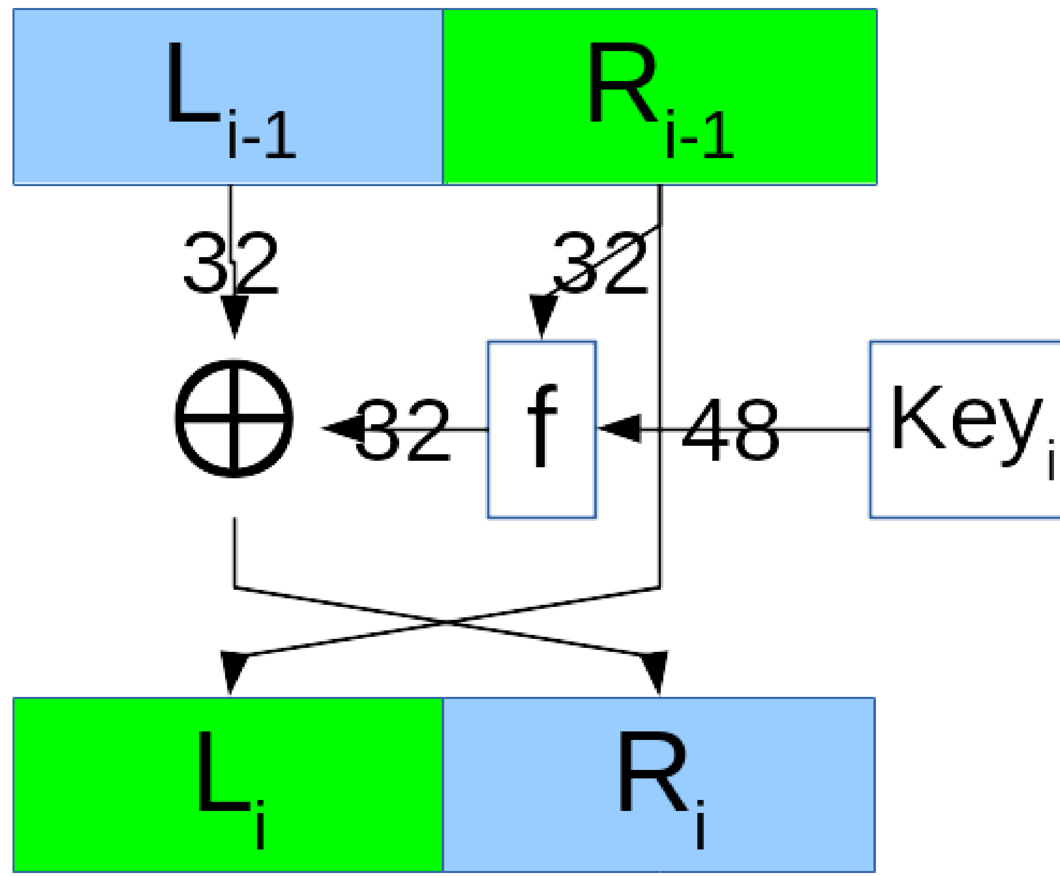
IP (lowest bit = 1)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

DES: Feistel Round

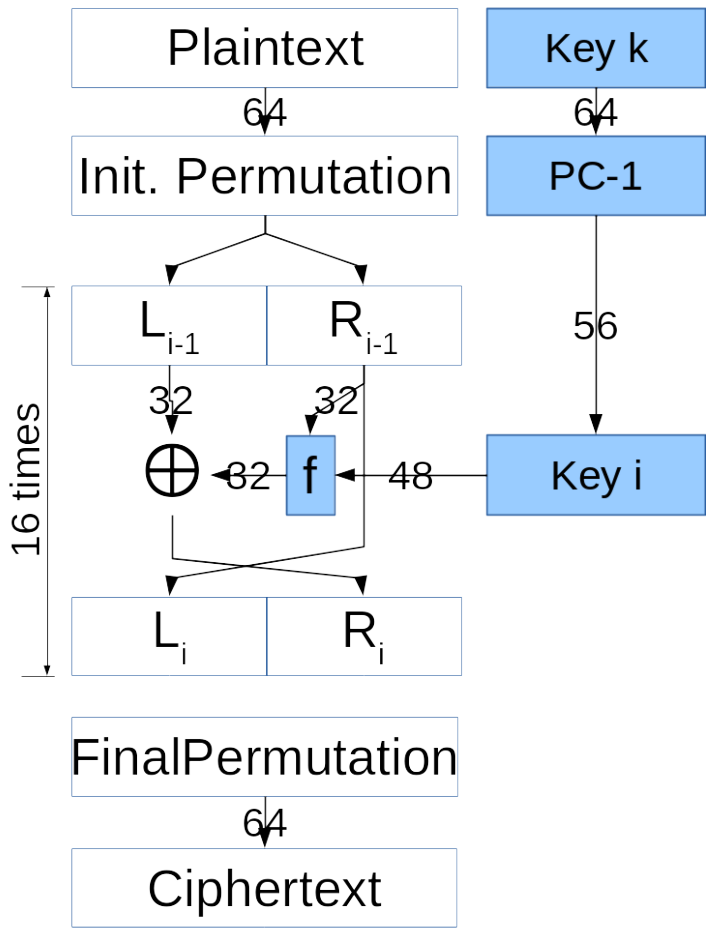


DES: Feistel Round

- Blocks are 64 bits long
  - Left = 32 bits,
  - Right = 32 bits
- Feistel Operation:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1}$ XOR $f(R_{i-1},k_i)$



DES: f-function

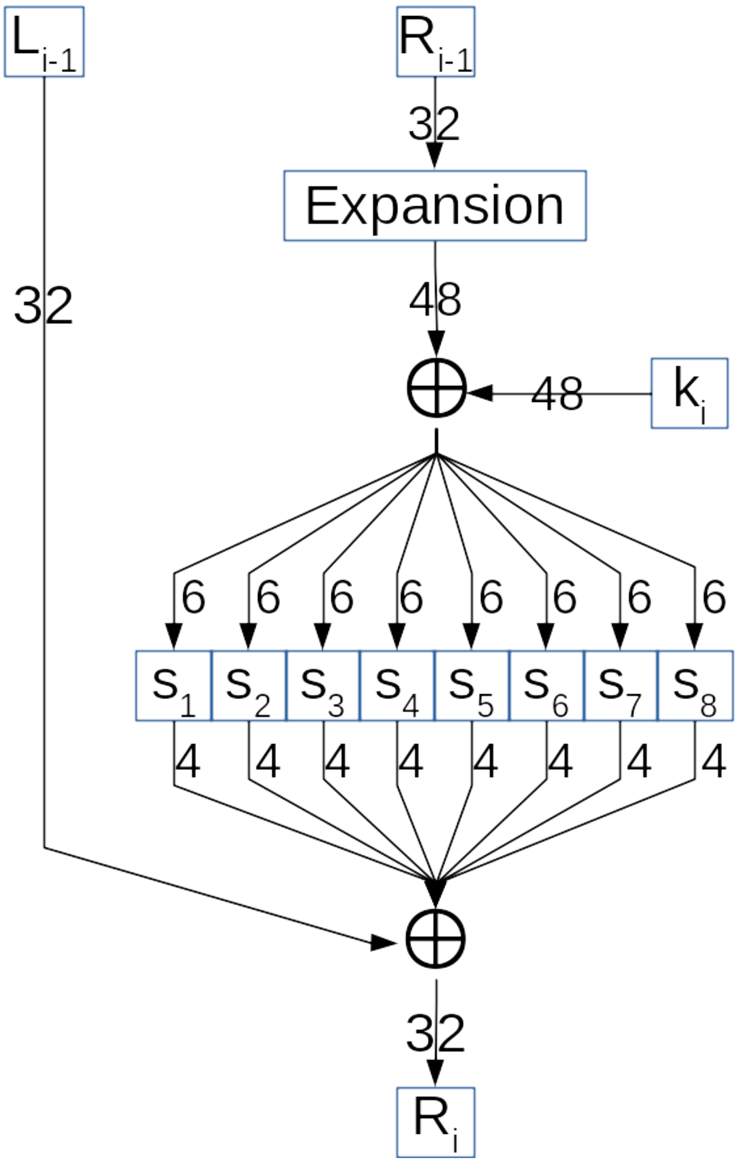


THIS IS THE MOST IMPORTANT PART!

THIS HAS THE SECURITY!

f-function: Expansion ("E") Boxes and Substitution ("S") Boxes

1. Expand R_{i-1} with E-box
   - 32 bits to 48
2. XOR with key_i
3. Substitute with S-box
   - 6 bits to 4
4. XOR with L_{i-1} to make R_i



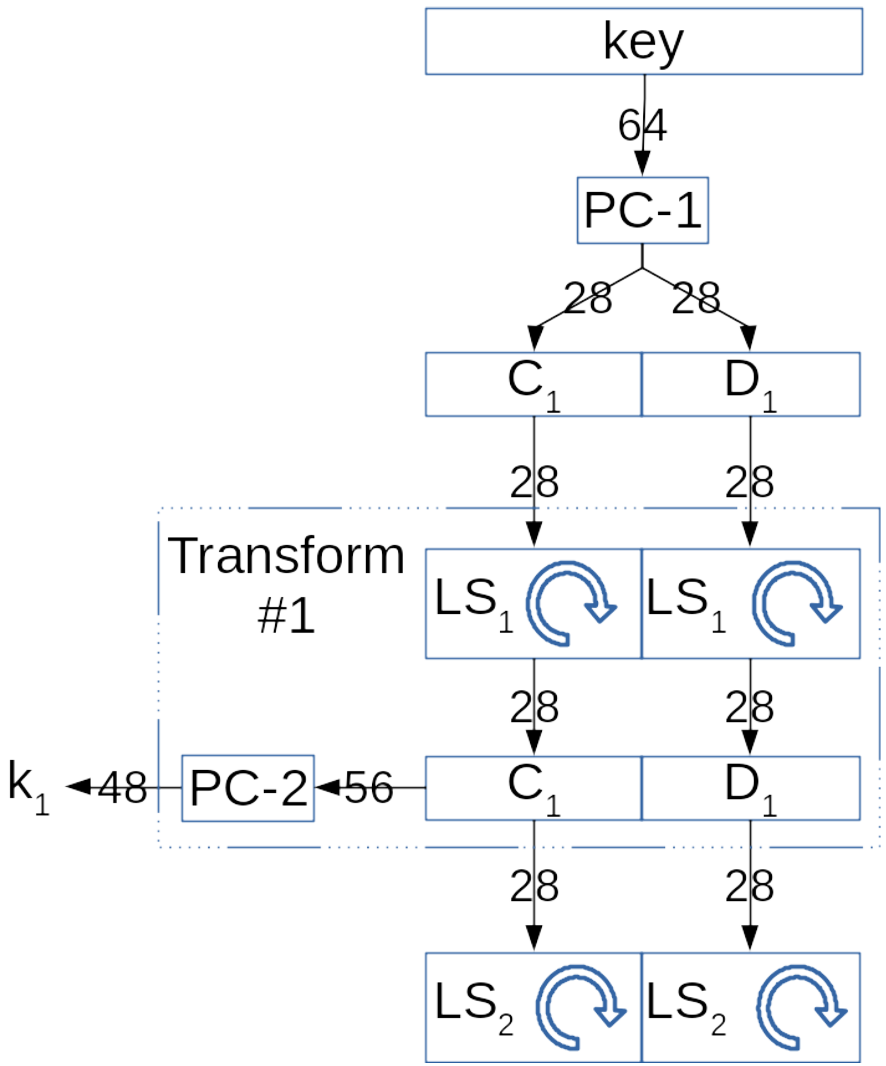
1. Expand with E-box

- 4 bit blocks to 6 bit blocks
- ½ bits appear twice
  - Each bit only once in 6 bit blocks

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

2. XOR with Key

- "Hey, where did that k_i come from?"
  - Permutation of original key
- PC-1
  - "Permuted choice one"
- Split into 2 halves
  - C_0 and D_0, both 28 bits
- Rounds:
  - 1,2, 9 and 16: Rotate left 1 bit
  - All others: Rotate left 2 bits
  - 4*1 + 12*2 = 28,
  - So: $C_0 = C_{16}$ and $D_0 = D_{16}$
- ***HOLD THAT THOUGHT!***
  - $C_0 = C_{16}$ and $D_0 = D_{16}$



2. XOR with key$_i$

- PC-1
  - "Permuted choice one"
- DES says "64 bit key"
- Every 8th bit is a parity bit
  - Effectively 56 bit key

PC-1							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

2. XOR with key_i, cont'd

- C_1 and D_1, both 28 bits
  - Together 56 bits
- Send into PC-2
  - It ignores 8 bits
- Remaining 48 bits used as f for XOR

PC-2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

3. Substitute with S-box

- ***This is where the strength comes from!***
  - Designed to be ***non-linear***
  - Provide confusion
- 8 boxes: S_1 to S_8 (S_1 given below)
- Read in funky fashion:
  - Input b=(***1***<u>0010</u>***1***)$_2$ Output (row ***11***$_2$=3, column <u>0010</u>$_2$ = 2) = 08

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

3. S-box criteria (revealed in 1990)

1. Each has 6 input and 4 output bits
2. No single output bit should be too close to linear combo of input bits
3. If the lowest and highest bits are fixed, and middle varied, each possible 4 bit output appears exactly once
4. If two inputs differ in exactly one bit, outputs must differ in 2 bits

3. S-box criteria, cont'd (revealed in 1990)

5. If two inputs to an S-box differ in the middle two bits, their outputs differ in at least two bits
6. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, then the two outputs must be different
7. For any non-zero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same difference.
8. A collision (zero output difference) at the 32-bit output of the 8 S-boxes is only possible for three adjacent S-boxes.

Differential Attack: Attack against block-cipher S-boxes

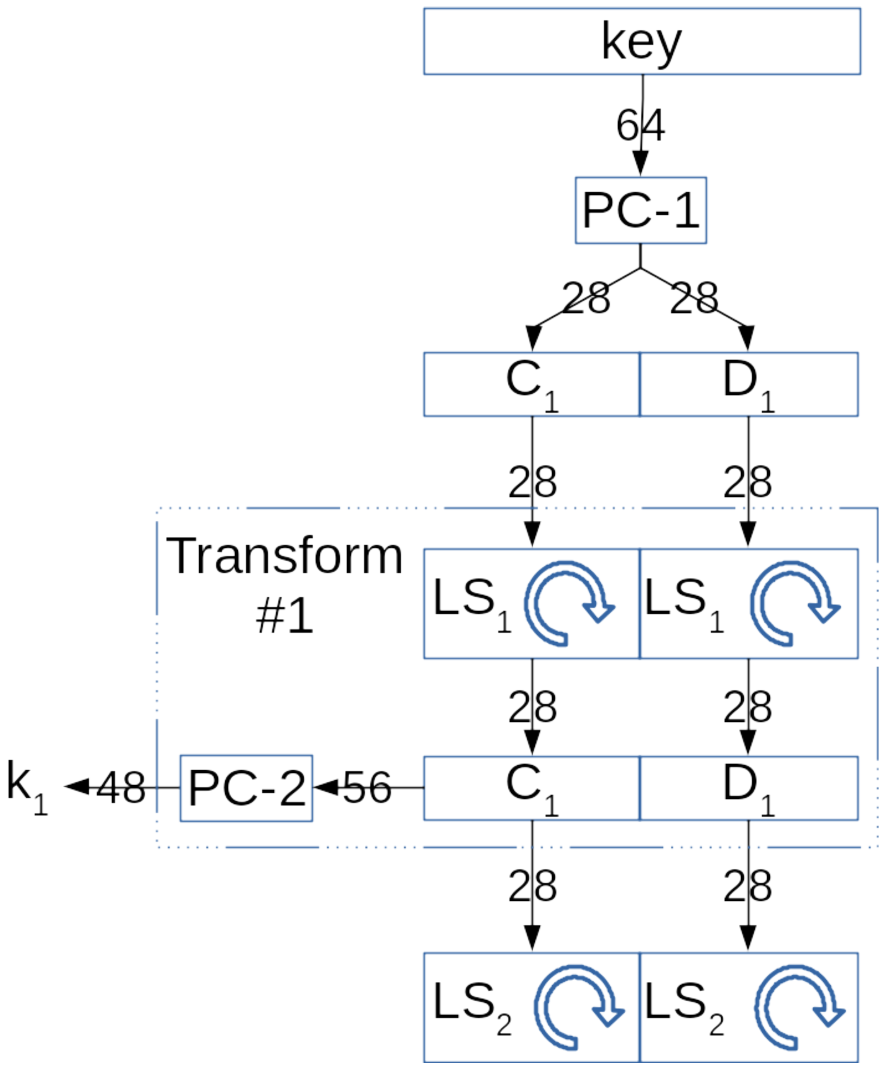
1. Get pairs of plain text related by constant difference
   - (e.g. XOR difference)
2. Get corresponding ciphertext
3. Compute differences in ciphertext
   - Look for statistical patterns in their distribution

Differential Attack: Attack against block-cipher S-boxes

- Published by Eli Biham (top) and Adi Shamir (bottom) in 1980s
  - Israeli cryptographers
  - Found that DES is resistant to that attack
- Known by IBM and NSA (independent of each other) in 1970s
- DES S-boxes designed to be non-linear
  - $S(a) \oplus S(b) \neq S(a \oplus b)$

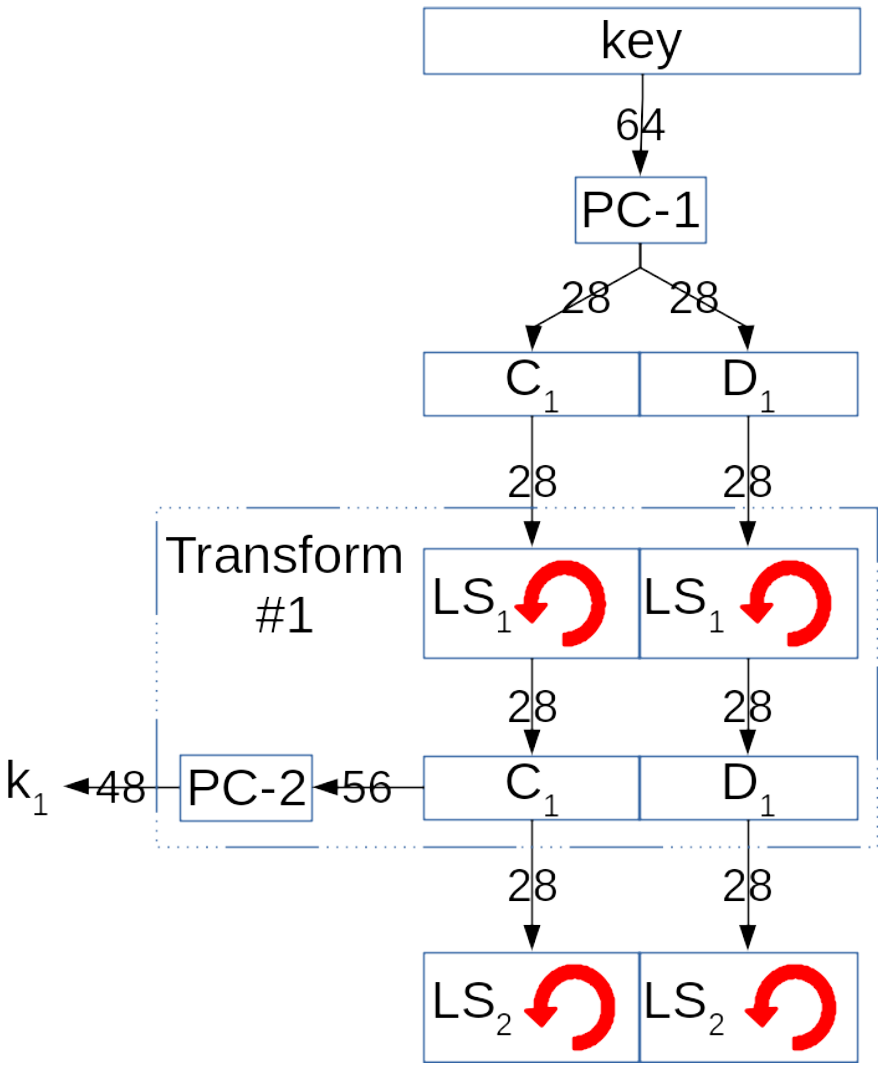
Decryption

- ***Did you remember?***
  - $C_0 = C_{16}$ and $D_0 = D_{16}$
- We ***rotated the key halves left*** to make k_i to encode



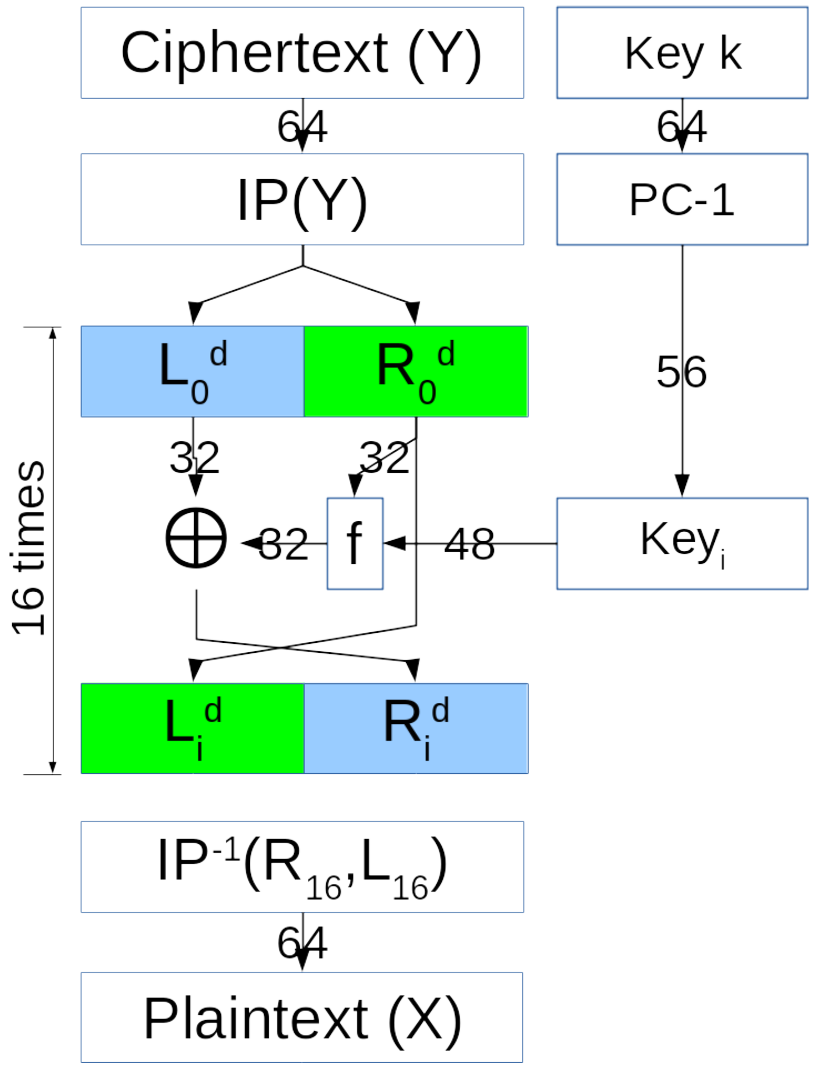
Decryption, cont'd

- To decode the only change is ***rotate key halves right***
- k_{16} =
  = PC-2(C_{16},D_{16})
  = PC-2(C_{16},D_{16})
- k_{15} =
  = PC-2(C_{15},D_{15})
  = PC-2(RS_2(C_{16}),RS_2(D_{16}))
- *Etc*.
- Round:
  - 1: no rotation
  - 2, 9, 16: right 1 bit
  - All others: right 2 bits



Decryption:
Feistel Network encrypts & decrypts

- Decrypt round 1 reverses encrypt round 16
- Decrypt round 2 reverses encrypt round 15
- Etc.
- Undo final permutation
  - Superscript ^{d} means "decryption"
- $(L_0^d, R_0^d) =$
  - $= IP(Y)$
  - $= IP(IP^{-1}(R_{16}, L_{16}))$
  - $= (R_{16}, L_{16})$



Decryption: Feistel Network encrypts & decrypts

1. Now consider it backwards, initially
   - $L_0^d = R_{16}$
   - $R_0^d = L_{16} = R_{15}$

2. Recursively:

- $R_1^d =$

$$= L_0^d \oplus f(R_0^d, k_{16})$$

$$= R_{16} \oplus f(L_{16}, k_{16})$$

$$= [L_{15} \oplus f(R_{15}, k_{16})] \oplus f(R_{15}, k_{16})$$

$$= L_{15} \oplus [f(R_{15}, k_{16}) \oplus f(R_{15}, k_{16})]$$

$$= L_{15}$$

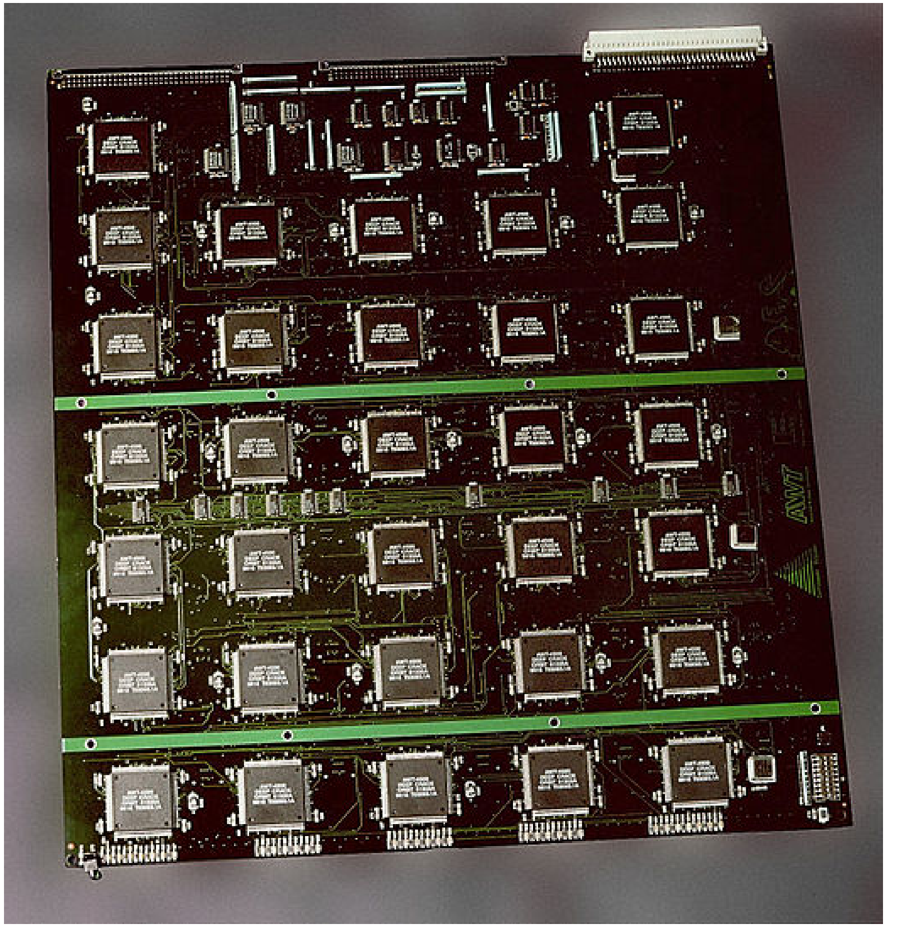
Analytical Attacks

We ***told*** you we made it strong!

- 1980s: Biham and Shamir
  - Try differential cryptanalysis
  - It was resistant
- Mitsuru Matsui
  - Try linear cryptanalysis
  - Also resistant
- Attacker needs
  - 2^{47} (plaintext,ciphertext) <u>*chosen*</u> pairs
  - 2^{55} (plaintext,ciphertext) <u>*random*</u> pairs
  - Of course, neither are realistic
- RSA Security sponsors an attack challenge

Brute-Force Attacks

- 1970s: Whitfield Diffie and Martin Hellman:
  - Estimate US$20,000,000 for dedicated cracker
- 1998: Electronic Frontier Foundation "Deep Crack"
  - US$250,000 of custom hardware
  - Average time = 15 days
  - Shown on right ===>
- 2006: COPACOBANA
  - US$10,000
  - Average time < 7 days



Modern Alternatives to DES

- 3DES: 3*56 = 168 bit key
  - Encrypt with 56 bits
  - Decrypt with 56 bits
  - Encrypt again with 56 bits
- A.E.S: Advanced Encryption Standard
  - *Stay tuned!*

- And other block ciphers:
  - Mars (royalty free)
  - RC6
  - Serpent (royalty free)
  - Twofish (royalty free)

References:

- "*Chapter 3: The Data Encryption Standard (DES) and Alternatives*" of Christof Paar and Jan Pelzl "*Understanding Cryptolography: A Textbook for Students and Practitioners*"
- "EFF DES Cracker Machine Brings Honesty to Crypto Debate" https://web.archive.org/web/20130517060659/http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_descracker_pressrel.html (Downloaded 2020 April 11)
- Biham, Eli; Shamir, Adi "*Differential Cryptanalysis of the Data Encryption Standard*" Springer Verlag, 1993.
- Coppersmith, Don (May 1994) "The Data Encryption Standard (DES) and its strength against attacks" *IBM Journal of Research and Development*, 38 (3): 243.
- Konheim, Alan G. "Horst Feistel: the inventor of LUCIFER, the cryptographic algorithm that changed cryptology" in "*Journal of Cryptographic Engineering*", Vol 9, pg 85-100 (2019)