The Target Cipher

- We want to break this cipher:

```
ADQYS MIUSB OXKKT MIBHK IZOOO
EQOOG IFBAG KAUMF VVTAA CIDTW
MOCIO EQOOG BMBFV ZGGWP CIEKQ
HSNEW VECNE DLAAV RWKXS VNSVP
HCEUT QOIOF MEGJS WTPCH AJMOC
HIUIX
```

First Tool: Kaskski’s Method

- Kaskski: if characters of the key appear over the same characters in the plaintext, repetitions in the ciphertext will occur

key	VIGVIGVIGVIGVIGV
plain	THEBOYHASTHEBALL
cipher	OPKWWECIYOPKWIRG

  - Distance between repetitions is 9, so the period must be a factor of 9 (that is, 1, 3, or 9)
  - Will the ciphertext contain the same repetition in the following two cases?

key	VIGVIGVIGVIGVIGVI	key	VIGJVIGJVIGJVIGJ
plain	THEBOOYHASTHEBALL	plain	THEBOYHASTHEBALL

Repetitions in Example

Letters	***Start***	***End***	***Distance***	***Prime Factors***
MI	5	15	10	2, 5
OO	22	27	5	5
OEQOOG	24	54	30	2, 3, 5
FV	39	63	24	2, 2, 2, 3
AA	43	87	44	2, 2, 11
MOC	50	122	72	2, 2, 2, 3, 3
QO	56	105	49	7, 7
PC	69	117	48	2, 2, 2, 2, 3
NE	77	83	6	2, 3
SV	94	97	3	3
CH	118	124	6	2, 3

Estimate of Period

- The longest repetition *OEQOOG* is probably not a coincidence
  - Distance is 30
- The second longest is *MOC*
  - Distance is 72
- GCD of 30 and 72 is 6
- Others
  - (7/10) have 2 in their factors
  - (6/10) have 3 in their factors
- 6 is a probable period

Splitting Into Alphabets

```
ADQYS MIUSB OXKKT MIBHK IZOOO
EQOOG IFBAG KAUMF VVTAA CIDTW
MOCIO EQOOG BMBFV ZGGWP CIEKQ
HSNEW VECNE DLAAV RWKXS VNSVP
HCEUT QOIOF MEGJS WTPCH AJMOC
HIUIX
```

alphabet 1: AIKHOIATTOBGEEERNEOSAI

alphabet 2: DUKKEFUAWEMGKWDWSUFWJU

alphabet 3: QSTIQBMAMQBWQVLKVTMTMI

alphabet 4: YBMZOAFCOOFPHEAXPQEPOX

alphabet 5: SOIOOGVICOVCSVASHOGCC

alphabet 6: MXBOGKVDIGZINNVVCIJHH

- Step 2 done; now we are dealing with 6 Caesar ciphers!

Frequency Examination

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	3	1	0	0	4	0	1	1	3	0	1	0	0	1	3	0	0	1	1	2	0	0	0	0	0	0
2	1	0	0	2	2	2	1	0	0	1	3	0	1	0	0	0	0	0	1	0	4	0	4	0	0	0
3	1	2	0	0	0	0	0	0	2	0	1	1	4	0	0	0	4	0	1	3	0	2	1	0	0	0
4	2	1	1	0	2	2	0	1	0	0	0	0	1	0	4	3	1	0	0	0	0	0	0	2	1	1
5	1	0	5	0	0	0	2	1	2	0	0	0	0	0	5	0	0	0	3	0	0	2	0	0	0	0
6	0	1	1	1	0	0	2	2	3	1	1	0	1	2	1	0	0	0	0	0	0	3	0	1	0	1

- Letter frequencies in English (H high, M medium, L low)

  HMMMHMMHHMMMMHHMLHHHMLLLLL

- #1 matches – the key is *0*

Begin Decryption

- #3 matches if the key is 8 (A-->I)
- #6 matches if the key is 21 (A-->V)
- Substitute into ciphertext (bold are substitutions)

AD**I**YS **RI**U**K**B O**CK**K**L** MI**GH**K **A**ZO**TO** E**I**OO**L**
IF**T**AG **PA**U**E**F V**AT**A**S** CI**IT**W **E**OC**NO** E**I**OO**L**
BM**T**FV **EG**G**O**P C**NE**K**I** HS**SE**W **N**EC**SE** D**D**AA**A**
RW**C**XS **AN**S**N**P H**HE**U**L** QO**NO**F **E**EG**OS** W**L**PC**M**
AJ**E**OC **MI**U**A**X

AJE in last line suggests "are", meaning #2 key is *18*, then we get:

Look For Clues

```
ALIYS RICKB OCKSL MIGHS AZOTO  MIOOL INTAG
PACEF VATIS CIITE  EOCNO MIOOL BUTFV EGOOP
CNESI  HSSEE NECSE LDAAA RECXS ANANP HHECL
QONON EEGOS ELPCM AREOC  MICAX
```

MICAX in last line suggests “mical” (a common ending for an adjective), meaning #4 key is O :

```
ALIMS RICKP OCKSL AIGHS ANOTO MICOL INTOG PACET
VATIS QIITE ECCNO MICOL BUTTV EGOOD CNESI VSSEE
NSCSE LDOAA RECLS ANAND HHECL EONON ESGOS ELDCM
ARECC MICAL
```

QI means that U maps into I, as Q is always followed by U

Got It!

After several rounds, we can fully decrypt it.

ALIME RICKP ACKSL AUGHS ANATO MICAL
INTOS PACET HATIS QUITE ECONO MICAL
BUTTH EGOOD ONESI VESEE NSOSE LDOMA
RECLE ANAND THECL EANON ESSOS ELDOM
ARECO MICAL

A LIMERICK PACKS LAUGHS ANATOMICAL
INTO SPACE THAT IS QUITE ECONOMICAL
BUT THE GOOD ONES IVE SEEN SO
SELDOM ARE CLEAN AND THE CLEAN ONES
SO SELDOM ARE COMICAL

One-Time Random Password

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable; Why?
  - Consider ciphertext DXQR. Equally likely to correspond to
    - plaintext DOIT (key AJIY) and
    - plaintext DONT (key AJDY) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

DES - History

- The Data Encryption Standard (DES) was developed in the 1970s by the National Bureau of Standards with the help of the National Security Agency.
- Its purpose is to provide a standard method for protecting sensitive commercial and unclassified data.
- IBM created the first draft of the algorithm, calling it LUCIFER. DES officially became a federal standard in November of 1976.

DES - History

- In May 1973, and again in Aug 1974 the NBS (now NIST) called for possible encryption algorithms for use in unclassified government applications.
- Response was mostly disappointing, however, IBM submitted their Lucifer design
- Following a period of redesign and comment it became the Data Encryption Standard (DES)

DES - As a Federal Standard

- DES was adopted as a (US) federal standard in November 1976, published by NBS as a hardware only scheme in January 1977 and by ANSI for both hardware and software standards in ANSI X3.92-1981 (also X3.106-1983 modes of use)
- Subsequently DES was widely adopted and published
- FIPS 46-3

http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

DES - Usage in Industry

- One of the largest users of the DES was the banking industry
- It is for this use that the DES has primarily been standardized.
- DES has been withdrawn as a standard by the National Institute of Standards and Technology

Current Status of DES

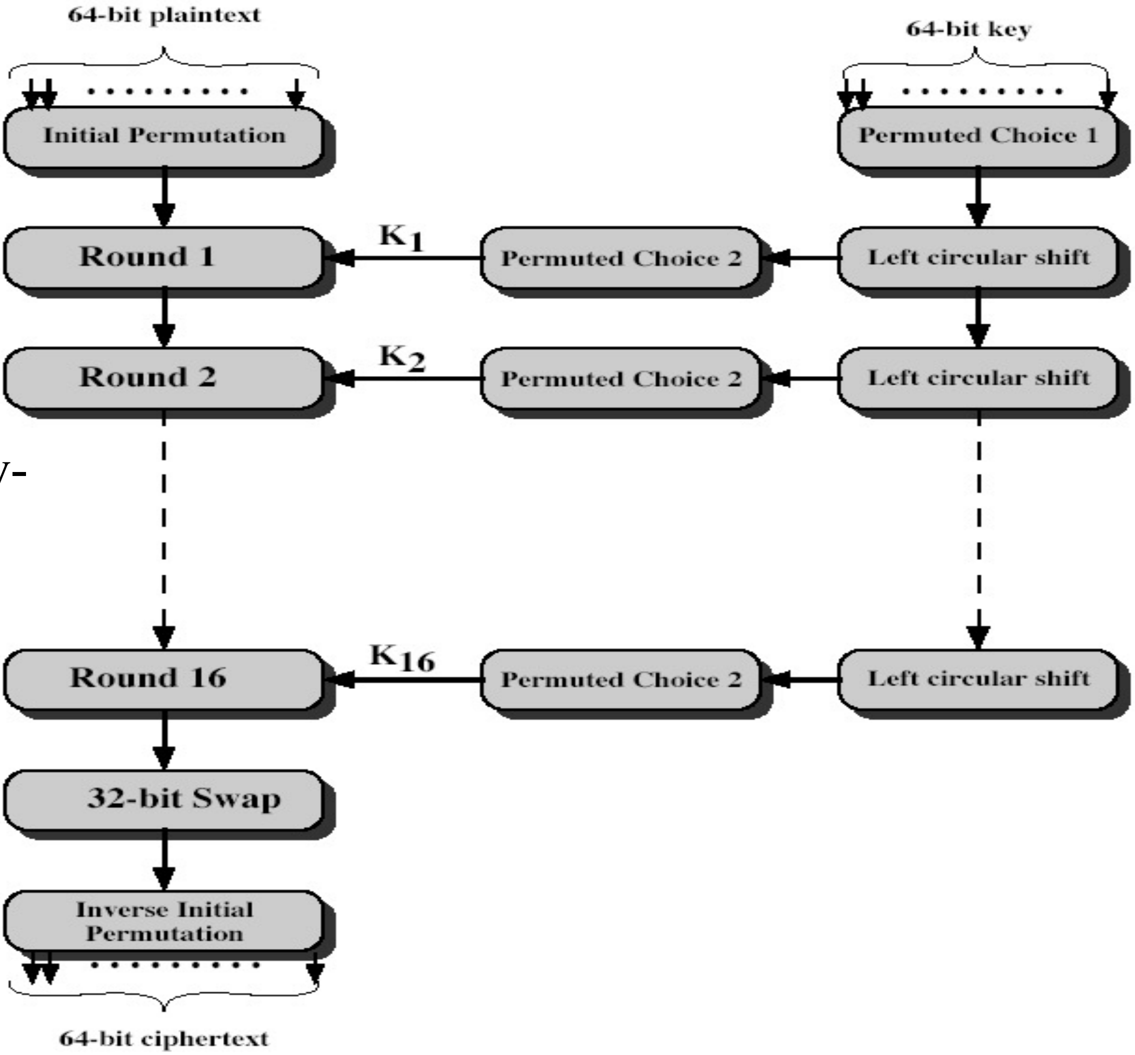
- Design for computer system, associated software that could break any DES-enciphered message in a few days was published in 1998
- In January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes
- NIST selected an Advanced Encryption Standard, successor to DES
  - Designed to withstand attacks that were successful on DES

Overview of the DES

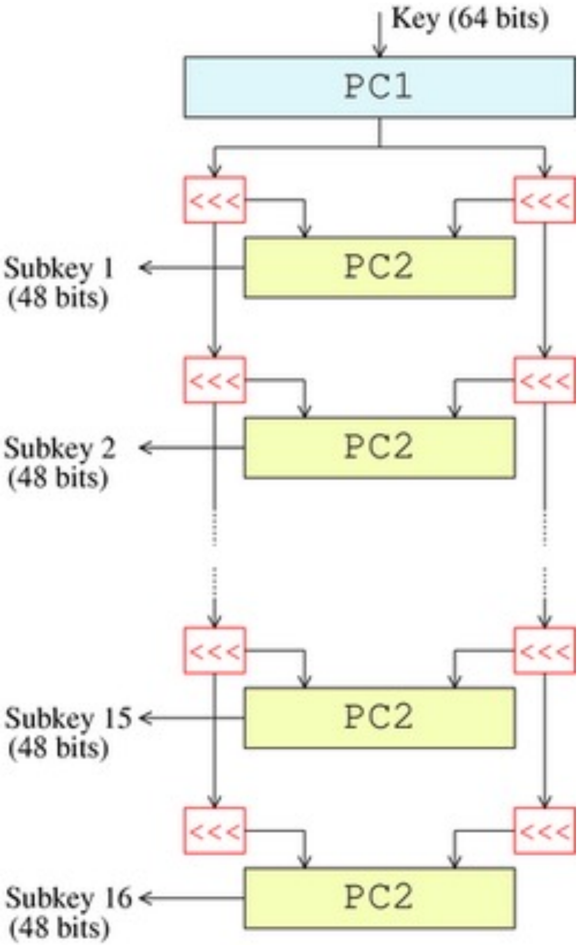
- A Symmetric Key Scheme
- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
  - A product cipher
    - performs both substitution and transposition (permutation) on the bits
  - basic unit is the bit
- Cipher consists of 16 rounds (iterations), each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

DES

- The basic process in enciphering a 64-bit data block consists of:
  - An initial permutation (IP)
  - 16 rounds of a complex key-dependent calculation
  - A final permutation, being the inverse of IP



DES – Key Schedule

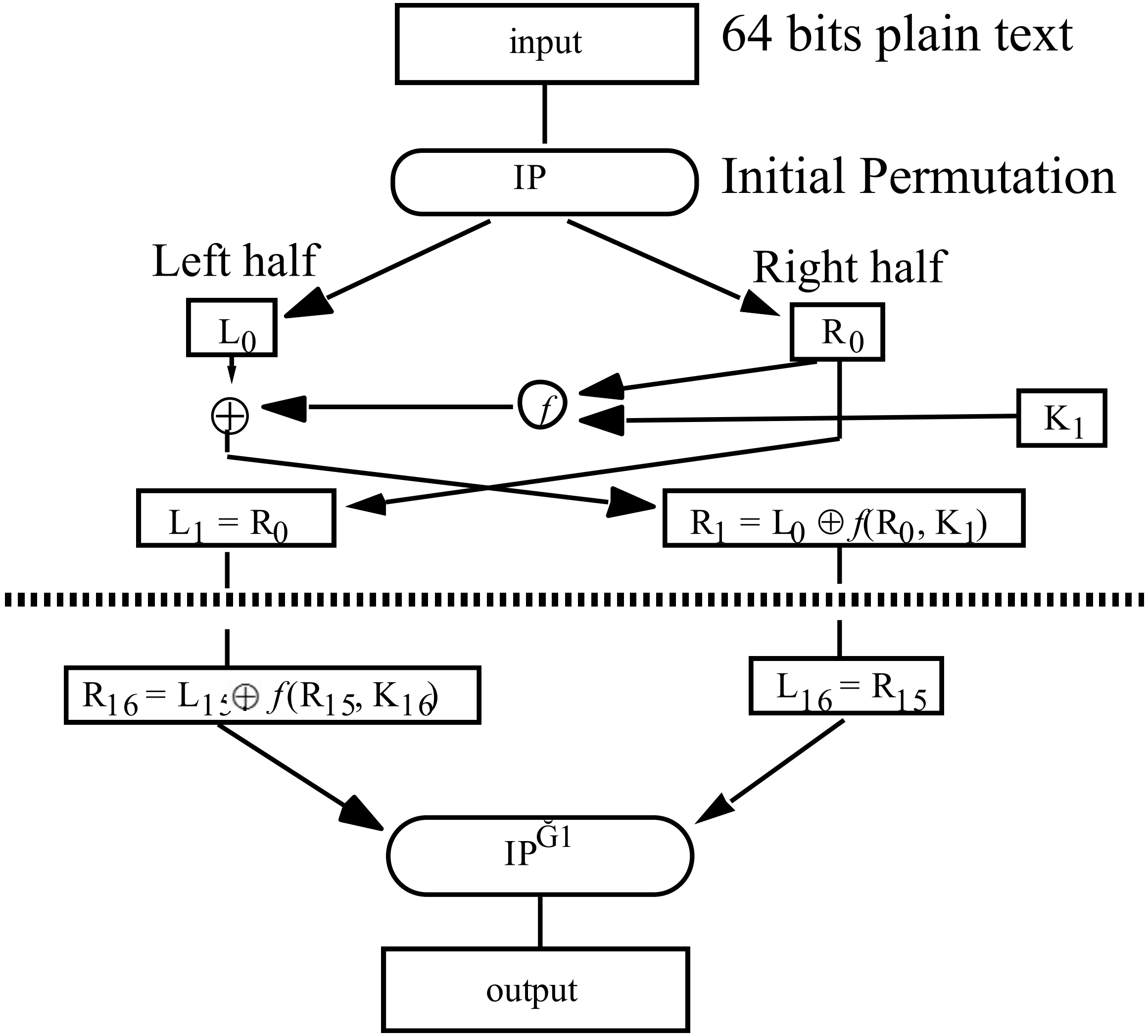


DES – 16 rounds

- The 64-bit block being enciphered is broken into two halves.
- The left half goes through one DES round, and the result becomes the new right half.
- The old right half becomes the new left half, and will go through the next round.
- This goes on for 16 rounds, but after the last round the left and right halves are not swapped, so that the result of the 16th round becomes the final right half, and the result of the 15th round (which became the left half of the 16th round) is the final left half.

Encipherment Illustration

Round 1



input

64 bits plain text

IP

Initial Permutation

Left half

Right half

L_0

R_0

f

K_1

$L_1 = R_0$

$R_1 = L_0 \oplus f(R_0, K_1)$

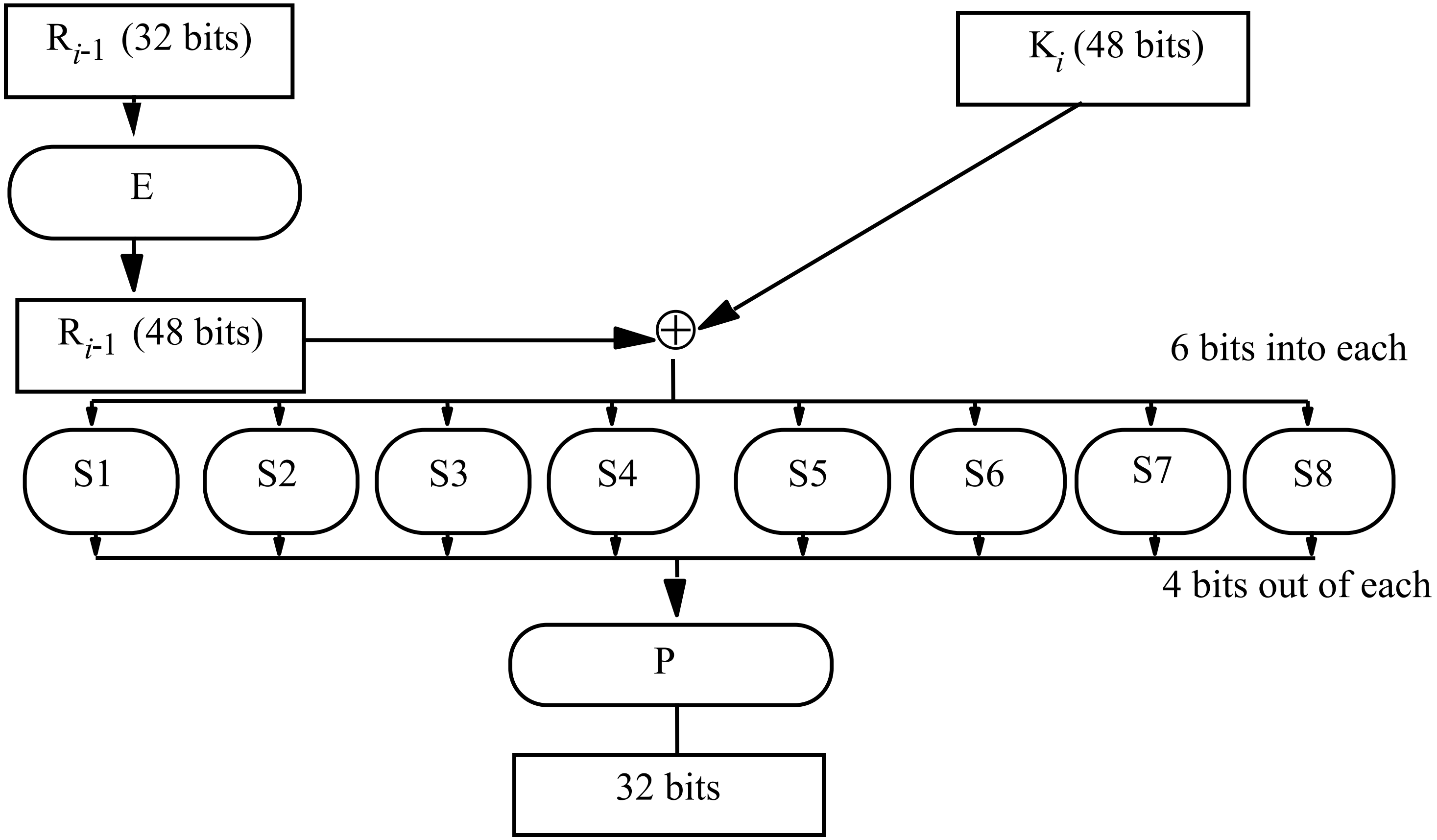
$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$

$L_{16} = R_{15}$

$\text{IP}^{\text{Ğ1}}$

output

The *f* Function



R_{i-1} (32 bits)
K_i (48 bits)
E
R_{i-1} (48 bits)
⊕
6 bits into each
S1
S2
S3
S4
S5
S6
S7
S8
4 bits out of each
P
32 bits