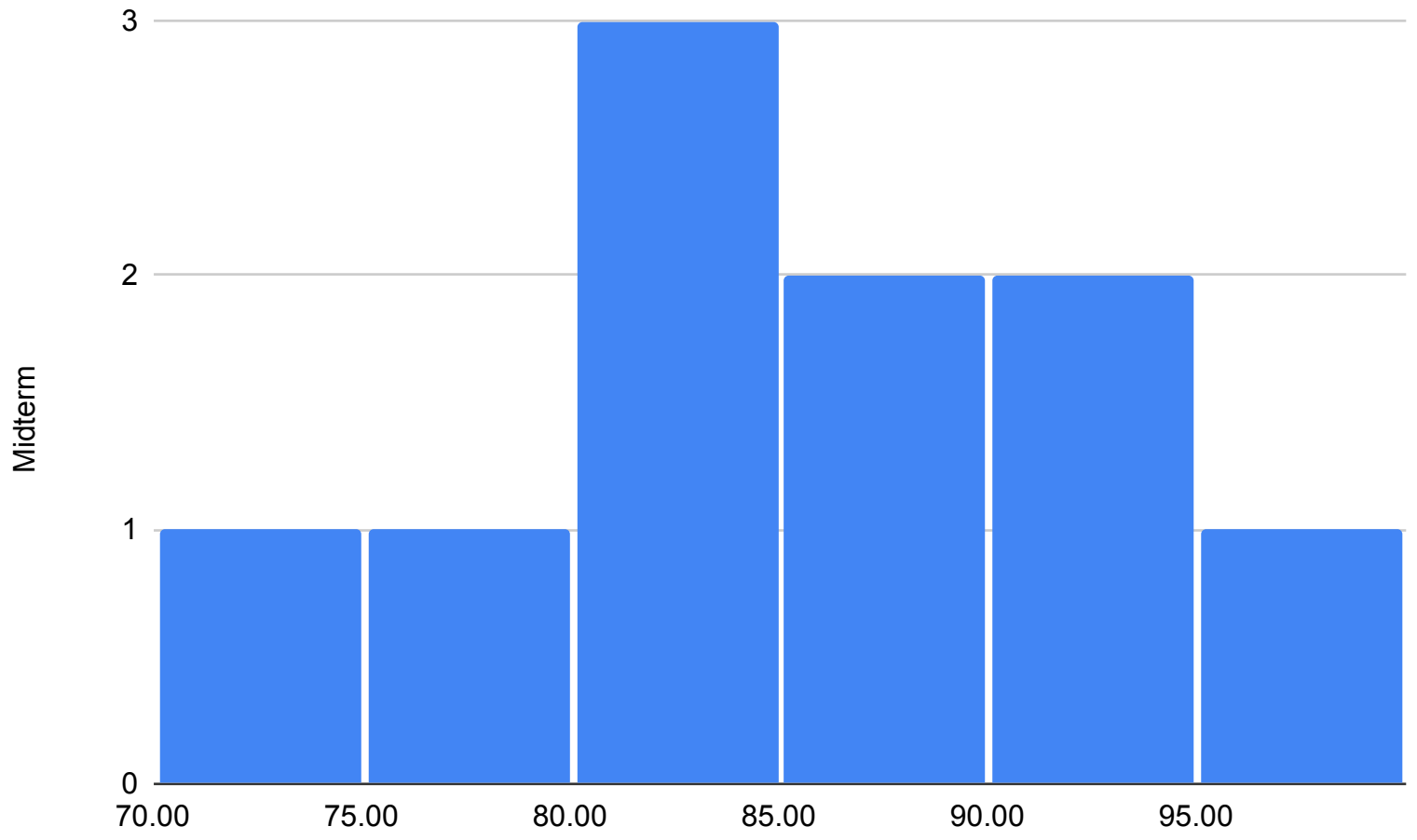
CSCI 3330 Cybersecurity

Word of the day

TO A ZEROTH APPROXIMATION [from to a first approximation] A really sloppy approximation; a wild guess


Midterm
3
2
1
0
70.00
75.00
80.00
85.00
90.00
95.00

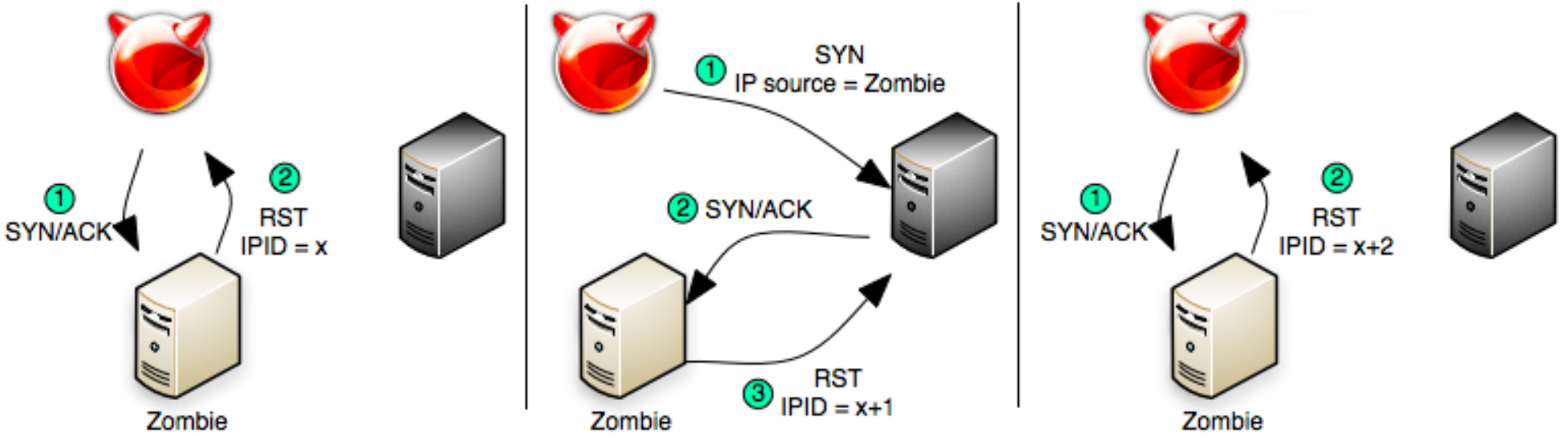
Idle Scan

- Off-path port scanning
- IP identifier
- *Spoofing*

Idle Scan

- IPID counters
- Global IPID
- Per-destination counter
- Hybrid approach (e.g., 2048 counters)

Idle Scan


1
SYN/ACK
2
RST
IPID = x
Zombie
1
SYN
IP source = Zombie
2 SYN/ACK
3
RST
IPID = x+1
Zombie
1
SYN/ACK
2
RST
IPID = x+2
Zombie

- Side channel: ???

- Side channel: IP IDs

Domain Name System (DNS)

- Converts name to IP address
- Record types:
  - NS: nameserver
  - A: IPv4 address
  - AAAA: IPv6 address

Security in DNS

- DNSSEC: DNS Security
- DoT: DNS over TLS
- DoH: DNS over HTTPS

Why not just DNSSEC?

- DNSSEC only validates, does not encrypt
- Most end user clients don’t verify
  - WHY? Because breaking DNS is already too common…
- DNS over HTTPS: piggyback on protocol already considered secure

Putting it all together

- DNSSEC
  - Between servers and recursive resolvers
- DoH/DoT
  - Between you and your DNS server

Cryptography

- Encrypt: convert data into a code
- Decrypt: convert code back into data
- Key: piece of information required to encrypt or decrypt
- Plaintext: the original data
- Ciphertext: the encoded data

Cryptography

- Kerckhoff’s Principle: the security of a cryptosystem must lie in the choice of its keys only; everything else (including the algorithm itself) should be considered public knowledge.

Cryptography

- Opposite: security through obscurity

Cryptography

- Given the ciphertext, an adversary should not be able to recover the original message
  - Enumerating all possible keys must be infeasible
  - There should be no way to produce plaintext from ciphertext without the key
- The ciphertext must be indistinguishable from true random values
  - Given a ciphertext, the probability of any possible plaintext being encrypted should be the same

Two properties of cryptosystem

- (As identified by Claude Shannon)
- Confusion
  - There should be no identifiable relationship between the ciphertext and the key
- Diffusion
  - If we change a bit of the plaintext, we would expect half of the bits of the ciphertext to change

Some attack models

- Chosen plaintext
  - Adversary can choose what to encrypt and see the encrypted output
- Chosen ciphertext
  - Adversary can choose what to decrypt and see the decrypted input

Caesar cipher

- Shift all letters by *i* letters (mod 26)

Caesar cipher

- Key: *i* (the shift amount) = 13

Plaintext	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
Ciphertext	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	j	l	m

- Plaintext: `i have often seen a cat without a grin`
- Ciphertext: `v unir bsgra frra n png jvgubhg n teva`

Caesar cipher

- Ciphertext: `v unir bsgra frra n png jvgubhg n teva`
- How do we crack this?
- How large is key space?

Substitution cipher

- Key:

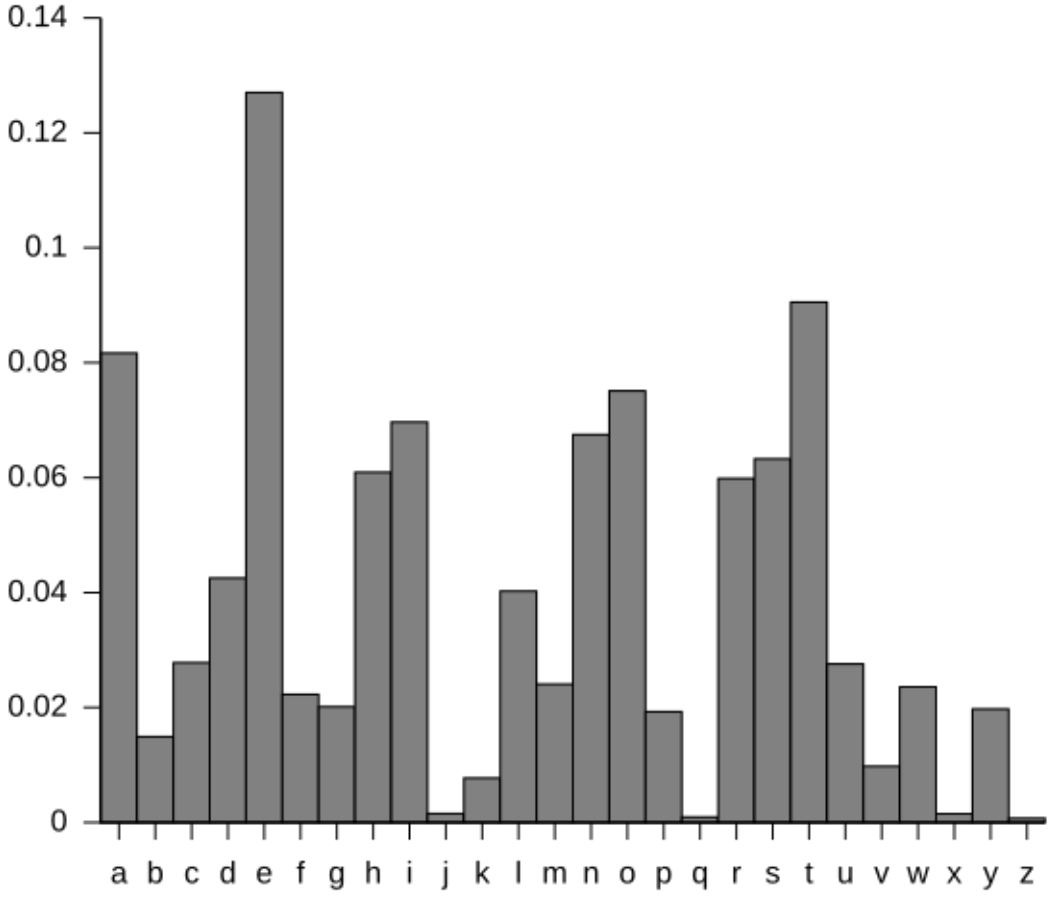
Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	u	h	p	t	g	l	k	m	j	a	w	e	v	b	c	x	z	s	d	f	y	o	q	i	r	n

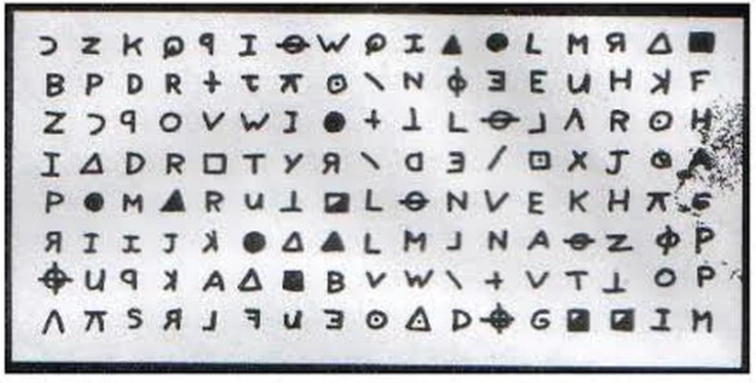
- Plaintext: `i have often seen a cat without a grin`
- Ciphertext: `j muog clfgb dggb u puf qjfmcyf u ksjb`

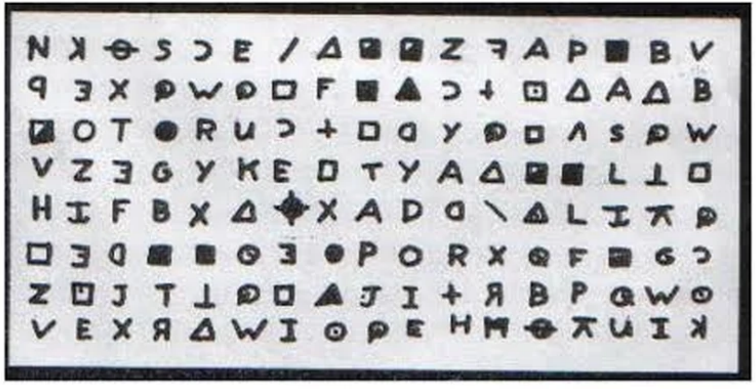
Substitution cipher

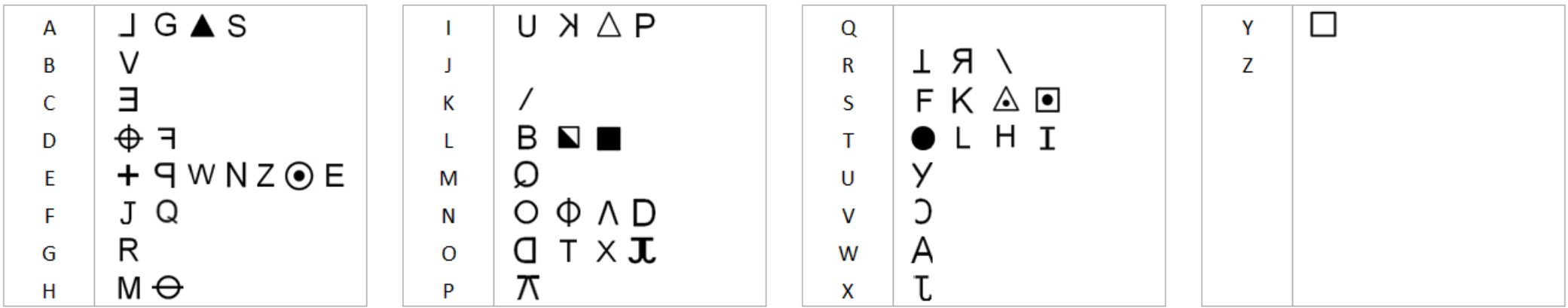
- Ciphertext: `j muog clfgb dggb u puf qjfmcyf u ksjb`
- How do we crack this?
- How large is key space?

Substitution cipher








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
G
S
W
N
Z
E
Q
M

How do we encrypt bytes?

- Stream ciphers
  - Combine input bytes with *keystream*
  - (Typically combined via XOR)
- Block ciphers
  - Encrypts fixed size blocks (typically 64 or 128 bits)
  - Often building block for stream cipher

XOR mask

- E.g., block size == 32 bits, key == 0xdeadbeef
- XOR every 4 bytes with 0xdeadbeef
- What are problems with the XOR mask?

One time pad

- Provably secure
- Requires key to be length of input
- Used by intelligence agencies
- Not very relevant to modern digital communications

Block ciphers problem

- So I have 64-bit block
- What if my message doesn’t exactly fit?

Padding

- Needs to be unambiguous
- What can we do?

PKCS#7 padding

- Pad the remaining *n* bytes with the byte *n*
- What if there are no remaining bytes?

Symmetric encryption

- Same key used to encrypt and decrypt

Common symmetric block ciphers

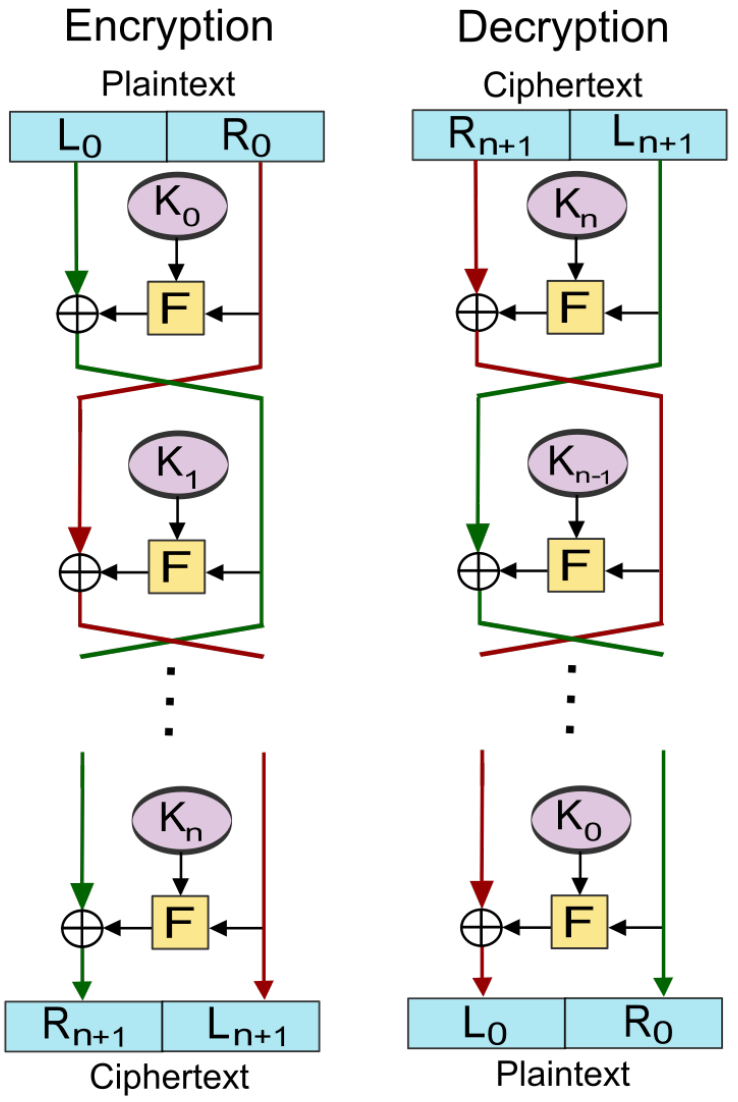
- DES
  - Data Encryption Standard
- AES
  - Advanced Encryption Standard

DES

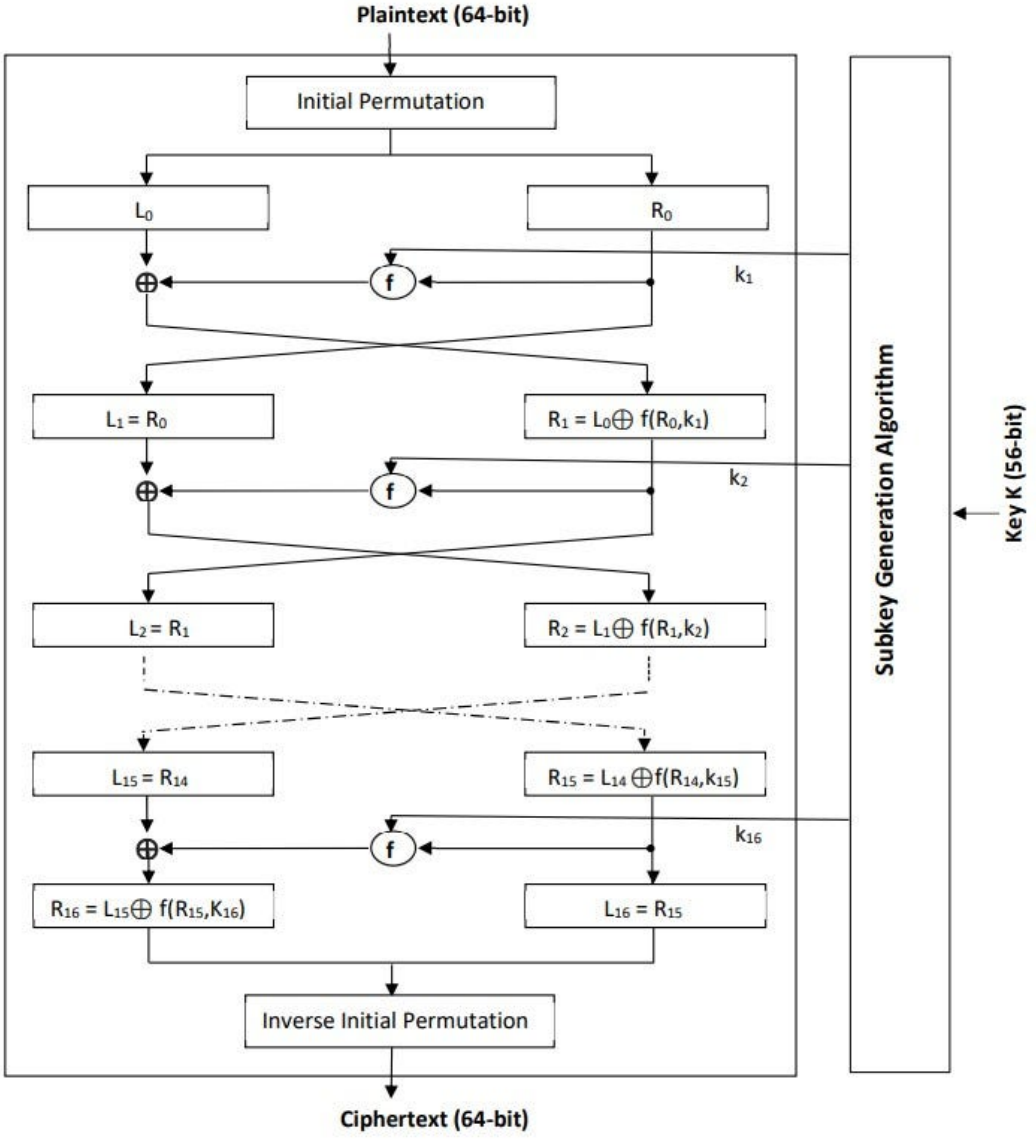
- Developed in early 70s by IBM
- 64-bit block cipher with a 56-bit key
- Feistel structure

Feistel cipher

- Rounds
- Key schedule
- Decryption is encryption with key schedule in reverse
- F: Feistel function

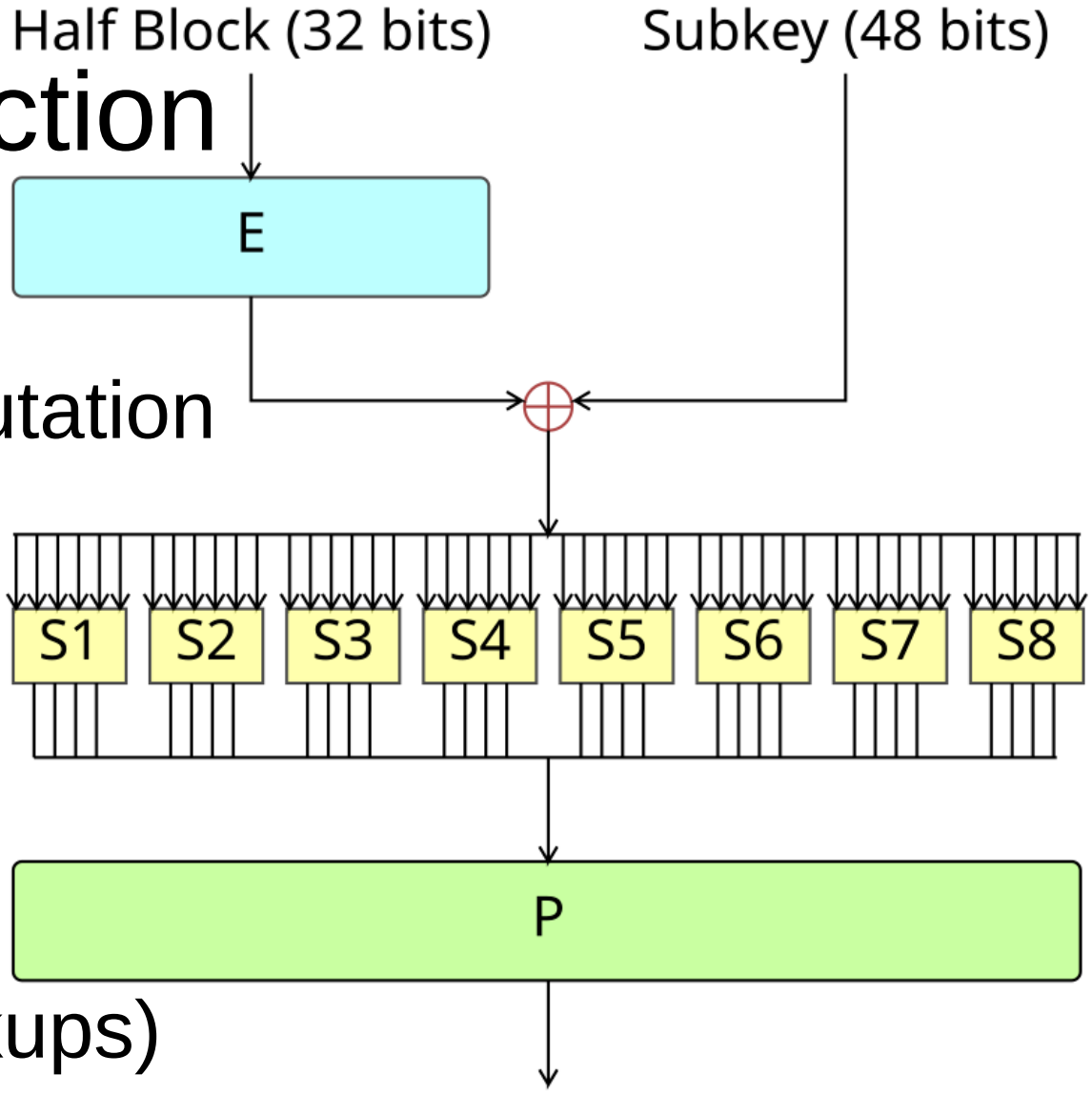


16 rounds


Plaintext (64-bit)
Initial Permutation
L_0
R_0
k_1
$L_1 = R_0$
$R_1 = L_0 \oplus f(R_0,k_1)$
k_2
$L_2 = R_1$
$R_2 = L_1 \oplus f(R_1,k_2)$
$L_{15} = R_{14}$
$R_{15} = L_{14} \oplus f(R_{14},k_{15})$
k_{16}
$R_{16} = L_{15} \oplus f(R_{15},K_{16})$
$L_{16} = R_{15}$
Inverse Initial Permutation
Ciphertext (64-bit)
Subkey Generation Algorithm
Key K (56-bit)

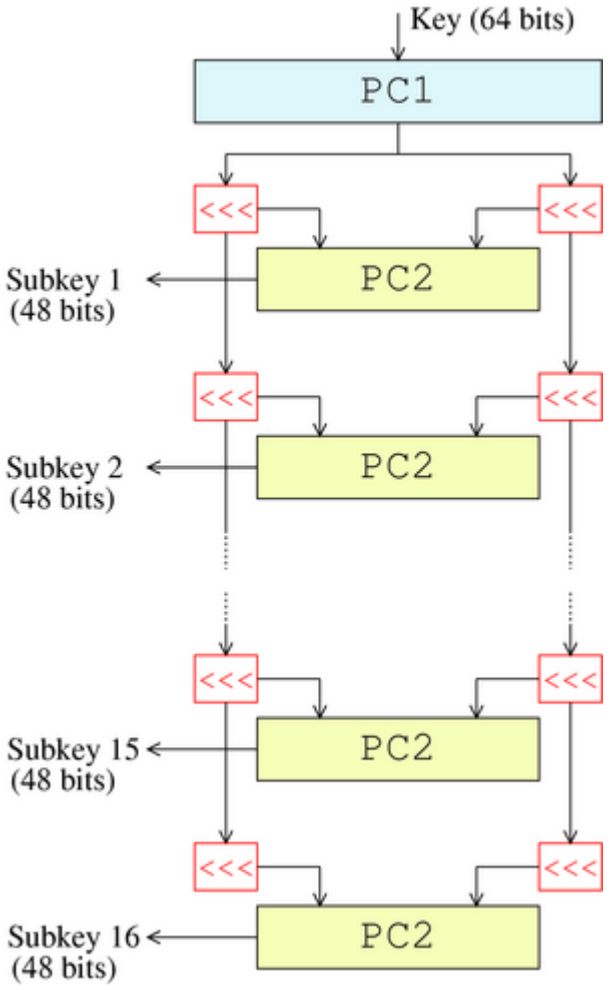
DES Feistel function

- E: expansion permutation (randomly choose bits with some chosen twice in a random order)
- P: permutation
- S: S-box (table lookups)



DES key schedule

- PC: permuted choice
  - (choosing subset of bits in a random order)



DES

- Is DES secure?
- There are generally no proofs of a cryptosytem being secure outside of trivial examples like a one-time pad

DES

- NSA nudged IBM to change S-boxes
- Did they introduce a backdoor?

DES

- NSA nudged IBM to change S-boxes
- Did they introduce a backdoor?
- Added resistance to differential cryptanalysis
- A kind of attack not discovered until 1990!

DES

- No longer considered secure
- Multiple attacks against it
- Can be brute forced using specialized hardware

3DES (Triple DES)

- Key = 3 DES keys k_1 k_2 k_3
- 56 * 3 == 168 bit key

  DES_encrypt(DES_decrypt(DES_encrypt(*plaintext*, k_1), k_2) k_3)

- Can be implemented using DES hardware with three DES operations
- 3DES hardware can implement DES
  - When k_1 == k_2 == k_3, implements DES with key k_1

Assignment 5

- Deadline: October 28, 2025, 10:00pm
- Real-world XSS attacks
  https://tildesites.bowdoin.edu/~j.knockel/csci3330f25/assignment5.pdf