EXAM 1 STUDY GUIDE

NUMBER SYSTEMS

Octal is a base-8 number system, using digits 0 - 7.

- ❖ Three binary digits represent one octal digit.

  0 = 000 1 = 001 2 = 010 3 = 011 4 = 100

  5 = 100 6 = 110 7 = 111

  So 75 in octal — 7 = 111 and 5 = 101, so 75 is 111 101 in binary.

- ❖ Use power series expansions to convert from octal to decimal.

  75 in octal — **7***8^1 + **5***8^0 = 61 in decimal

Hexadecimal is the base-16 number system, using digits 0 - 9 and letters A - F, with A representing 10 and F representing 15.

- ❖ Four binary digits represent one hexadecimal digit.

  1 = 0001 2 = 0010 3 = 0011 4 = 0100 5 = 0101

  6 = 0110 7 = 0111 8= 1000 9 = 1001 A = 1010

  B = 1011 C = 1100 D = 1101 E = 1110 F = 1111

  So A5 in hexadecimal — A = 1010 and 5 is 0101, so 1010 0101 in binary.

- ❖ To convert from binary to hex, take 4 bits at a time from right to left, replacing with hex digits; add 0's to the left if the last section isn't an even 4 bits.

  1011111010101 — 0001 | 0111 | 1101 | 0101 = 17D5

- ❖ Use power series expansions to convert from hexadecimal to decimal.

  A5 in hexadecimal — **A***16^1 (10*16^1) + **5***16^0 = 165 decimal

In base-2 numbering systems (**binary**), with *x* bits, you can only represent 2^x numbers. Since only a fixed amount of positive and numbers can be represented, you might get wrong results in arithmetic operations if they **overflow**.

- ❖ Overflow has only occurred in two's complement if you add two positive numbers and get a negative result, or if you add two negative numbers and get a positive result.
- ❖ Negating two's complement — inverse all the bits, then add 1.
- ❖ To subtract in two's complement, negate the subtracted number, then add the two.

4-bit one's complement number system

0111	0110	0101	0100	0011	0010	0001	0000	1111	1110	1101	1100	1011	1010	1001	1000
+7	+6	+5	+4	+3	+2	+1	+0	-0	-1	-2	-3	-4	-5	-6	-7

- ❖ In **one's complement**, positive numbers are the same, but negative numbers are the bit complement of the corresponding positive value, and always start with 1.

  -7 in one's complement — 7 is 0111, the bit complement is 1000, so -7 is 1000.

MODULAR ARITHMETIC

Modular congruence means that *a* mod *n* = *b* mod *n*, so if *a* and *b* are divided by *n*, they have the same remainder.

Modular reduction:

- **(a + b) mod n** = ((a mod n) + (b mod n)) mod n

  (12 + 8) mod 5 = ((12 mod 5) + (8 mod 5)) mod 5 = (2 + 3) mod 5 = 5 mod 5 = 0
- **(a * b) mod n** = ((a mod n) * (b mod n)) mod n

  (12 * 8) mod 5 = ((12 mod 5) * (8 mod 5)) mod 5 = (2 * 3) mod 5 = 6 mod 5 = 1
- **$a^{(b+c)}$ mod n** = ((a^b mod n) * (a^c mod n)) mod n

  2^6 mod 5 = ((2^2 mod 5) * (2^4 mod 5)) mod 5 = ((4 mod 5) * (16 mod 5)) mod 5 = 4
- If x < 0, we add a sufficiently large multiple of n to x to get a nonnegative number.

  -27 mod 13, add 3 * 13 = 39 to -27 to get 12, and 12 mod 13 is 12.

Modular inverses — y is the modular inverse of x, modulo n, if (x * y) mod n = 1.

- 1 is always the modular inverse of 1 for any modulo n.

SECURITY TERMINOLOGY

Security in computing refers to techniques, methods, and policies for protecting the information in computing systems and the system and hardware that use, store, and transmit that information. Security goals are confidentiality, integrity, and availability, assurance, authenticity, and anonymity.

- **Confidentiality** is achieved by encryption, access control, authentication, and physical security, guided by **separation of privilege**, which dictates that multiple conditions should be required to achieve access to restricted resources; it also implies a separation of system components, so that breaches are contained.
- **Integrity** is achieved through backups, checksums (computing a numerical value based on the contents of an entire file to detect alterations), and data-correcting codes (low-level codes for automatically correcting errors, such as retransmitting packets).
- **Availability** is ensured by physical protections and computational redundancies (computers and storage devices that serve as fallbacks in case of failures).
- **Assurance** is how trust is provided and managed in computer systems, guided by the principle of **least privilege**, where each program and user of a computer system should operate with the bare minimum privileges necessary to function.
- **Authenticity** is the ability to determine that statements, policies, and permissions issued are genuine; the primary tool for this is **digital signatures**, which ensure **nonrepudiation** (the property that authentic statements issued by a person or system cannot be denied.
- **Anonymity** is the property that certain records or transactions not to be attributable to any individual.

Attacks can be made to **vulnerabilitie**s in a security system.

- ❖ Types of attacks are access attacks (to confidentiality), modifications (integrity attacks), denials-of-service (availability attacks), repudiations (authenticity attacks), and anonymity attacks.
- ❖ **Access attacks** are attacks where the attacker attempts to gain access to information they aren't authorized to see; this includes eavesdropping, snooping, and interception.
- ❖ **Denial-of-service** attacks attempt to deny the use of resources to legitimate users of a computing system; this includes attacks to system access, communication links, specific applications and information.
- ❖ **Modification** attacks include changes, insertions, and deletions.
- ❖ **Repudiation** attacks are attacks that attempt to give false information or to deny an event or transaction that occurred; this includes masquerading (attempting to act like someone else or another system) and denying an event (such as not transmitting a data receipt acknowledgement).
- ❖ **Anonymity** attacks include correlation (identifying the person or entity associated with a particular piece of information) and traceback (identifying the source system for a data stream).

A **cryptosystem** is the set of ground rules for encrypted communications; it ensures confidentiality, but not integrity or authenticity.

- ❖ A cryptosystem consists of a set of possible plaintexts, a set of possible ciphertexts, a set of encryption keys, a set of decryption keys, the correspondence between encryption keys and decryption keys, the encryption and the decryption algorithms.
- ❖ **Public-key**, or asymmetric cryptosystems, solve the problem of key distribution (use far fewer key pairs), and digital signature issues, but these algorithms run much slower and require a key length one order of magnitude larger than symmetric keys; usually, they are just used to let two parties exchange a shared secret keys that is then used to run a symmetric encryption scheme.
- ❖ **Digital signatures** are obtained by reversing the order of encryption and decryption, which normally goes $C = E_{PB}(M)$, so $S = E_{PR}(M)$, where **the message is encrypted using a private key** instead of the recipients public key, and the recipient uses the public key to reproduce the stored message from the signature; if the messages are the same, then the message is authentic.
- ❖ A **Man-in-the-Middle attack** occurs if the adversary can intercept messages between recipients.
- ❖ A **cryptographic hash function** is an algorithm that produces a fixed-length checksum (called a **hash**); it should be one-way (easy to compute the hash from the message, but not the message from the hash), and collision resistant.
- ❖ Hashes are useful for digital signatures, integrity checks, and **Message Authentication Codes** (MAC), where a sender computes the hash of the shared secret key and the message.

❖ **Digital certificates** vouch for an entity and contain a public key and a digital signature; they're issued by trusted certificate authorities to verify identity.

TYPES OF CIPHERS

Keyword cipher — use a keyword to encrypt a message.

❖ Using the word PROGRAM to encrypt the word "program":

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

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

So "program" becomes LQKBQPI; the number of possible ciphers: 26! / (26 – n)!

Monoalphabetic cipher — an arbitrary letter jumble, so the key is 26 letters long; each plaintext letter maps to a different random cipher text letter.

❖ Number of possible ciphers is 26!.

Polyalphabetic cipher — a substitution cipher that uses multiple substitution alphabets and switches between them systematically.

❖ One example is the Vigenere cipher, as well as the enigma cipher.

❖ In the **Vigenere cipher**, the key is multiple letters long, and the ith letter of the key specifies which letter of the Vigenere table to use for substitution by using each key letter to index the row; after each letter of the key has been used, the key is started again. This cipher is **symmetric**.

❖ Using the word "deceptive" to encrypt "wearediscoveredsaveyourself"

d e c e p t i v e d e c e p t i v e d e c e p t i v e (3 4 2 4 15 19 8 21 4)

w e a r e d i s c o v e r e d s a v e y o u r s e l f

So the cipher text is "ZICVTWQNGRZGVTWAVZHCQYGLMGJ".

❖ A shortcut is to add the number values together, modulo 26, i.e. (S + E) mod 26, or (18 + 4) mod 26, or 22, which results in W.

One-time pad, aka a Vernam cipher, is a special case of a Vigenere cipher but the keyword length is as long or longer than the entire plaintext, and each shift amount must be completely random.

❖ If properly used, the one-time pad is unbreakable, and a key can never be used more than once.

❖ **Binary one-time pad** operates on the bit-level, and XORs the plaintext with the key.

Playfair cipher — a manual, symmetric, bigram (2 letter pairs) encryption technique

❖ The key matrix is a 5x5 matrix of letters based on a keyword, then the rest of the matrix is filled in using the rest of the alphabet (minus duplicates), pairing "i" and "j".

❖ Each letter is replaced by the letter in the **opposite diagonal corner** (same row, but the column of the other letter).

❖ If there's one character left over, **pad with "Z"**, then random letters as needed; if a pair is a repeated letter, **use "X" as a filler**; if a letter must be selected from the I/J box, always select I.

❖ If both letters are in the same row, replace each letter with the letter to the right; if both letters fall in the same column, replace each letter with the letter below it (wrapping around to the left, or to the top).

Hill cipher — a cipher based on matrix multiplication where letters are encoded as numbers modulo 26.

❖ They key is a random matrix of the same dimension as the vectors (which are the blocks of letters); the key **must have an inverse** modulo 26 to allow for decryption.

❖ First, multiply the bigram vector by the key, and after getting the answer in the form of the matrix, get modulo 26 on each of the numbers to find which letters the vector corresponds to.

$$C = \begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix} \begin{bmatrix} 19 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \cdot 19 + 8 \cdot 4 \\ 17 \cdot 19 + 3 \cdot 4 \end{bmatrix} = \begin{bmatrix} 23 \\ 23 \end{bmatrix} = XX$$

❖ Pad with "Z".

Transposition ciphers — mapping is performed by a permutation (rearrangement) of plaintext letters.

❖ **Rail fences** — writing the messages with alternating letters across rows.

Depth 2:

❑ Example: meet me after the toga party

m e m a t r h t g p r y

e t e f e t e o a a t

cipher: MEMATRHTGPRYETEFETEOAAT

Depth 3:

❑ Example: meet me after the toga party

m m t h g r

e t e f e t e o a a t

e a r t p y

cipher: MMTHGRETEFETEOAATEARTPY

❖ **Columnar transposition cipher** — plaintext written out in fixed-length rows, pad with random characters at the end if necessary; use a numeric key to encrypt by reading characters by column in the order specified by the key.

Key:	4	3	1	2	5	6	7
Text:	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z

Key gives row number to transpose columns

Ciphertext: TTNAAPTMTSUOAODWCOIXK NLYPETZ

Substitution-permutation networks — based on substitution and permutation;

❖ Provides **diffusion** (dissipates statistical structure of plaintext over bulk of cipher text).

❖ Provides **confusion** (makes relationship between cipher text and key as complex as possible).

❖ **Feistel cipher** — implements S-PN, where each round involves a substitution step then a permutation step; the plaintext block is split into two halves, (L)eft and (R)ight, and R passes through unchanged, and L goes through a substitution known as a the Feistel function (depends on key and R), then they permute by swapping halves.

$L_1 = R_0 \qquad R_1 = L_0 \oplus F(R_0, K_1)$

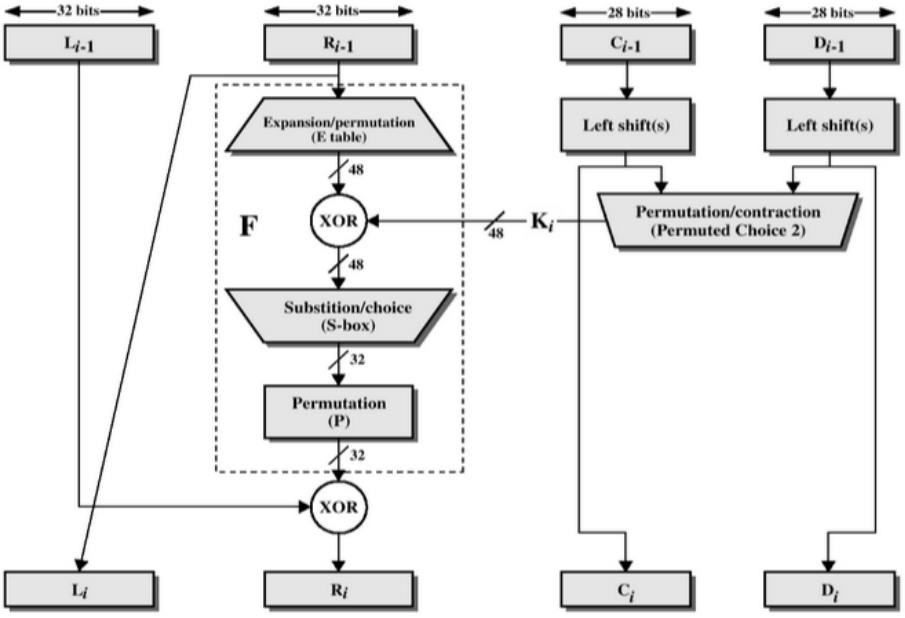

Figure 2.4 Single Round of DES Algorithm

DES uses a 16-round Feistal structure, and encrypts data using a 56-bit key; keys for each round are generated from the single input encryption key.

❖ It encrypts 64-bits at a time, splits the data into two 32-bit halves (L & R), expands R to 42 bits, adds R to the subkey using XOR, and passes through 8 S-boxes to return back to 32-bits, then permutes using a 32-bit P-box.

❖ **Expansion** is done by dividing the 32-bit block into 8 4-bit word, adding a bit to the left of each 4-bit word that's the last bit of the previous word, then adding a bit to the right of each 4-bit word that's the first bit of the next 4-bit word.

❖ **Substitution** divides the 48-bit block into 8 6-bit words, each word is fed into a separate S-box and produces 4-bit output — the first and last bit index the row, the middle 4 bits index the column, and the result is a 4-bit unsigned binary number.

- ❖ S-box input for FOE_{16}: 1111 0000 1110 ➔ 111100 and 001110, so for S_1, 111100, which is row 10, or 2 in decimal, and the column selector 111100, so 1110 or 14 in decimal; for S2, 001110, row 0, and 001110, or 7 in decimal; convert back to base-16 and combine.
- ❖ **Permutation** takes 32-bits and uses a look-up table of fixed permutation for the input bits, and then outputs 32 bits.
- ❖ **New subkeys are generated for each round** based off of the initial 56-bit key; before any round keys are generated, the original key is permuted (using Permutation Choice 1 table); this is called a **key schedule**.
- ❖ To create each subkey, the 56-bit key is split into two halves, and each block is circularly shifted left by 1 or 2 bits, and the halves are joined together and passed to the next round, and shortened to 48-bits (using Permutation Choice 2 which selects a subset the bits) for use in the current round.