e-Business Systems

(Unit – 2)

by

Dr. Sunil Pratap Singh
(Associate Professor, BVICAM, New Delhi)

2023

Computer Security (CIA)

- The protection afforded to an automated information system in order to attain the applicable objectives of preserving the confidentiality, integrity and availability of information system resources (includes hardware, software, firmware, information/data, and telecommunications).
  - **Confidentiality:** It assures that private or confidential information is not made available or disclosed to unauthorized individuals.
  - **Integrity:** It assures consistency, accuracy and trustworthiness of data over its lifecycle.
  - **Availability**: It assures that systems work promptly and service is not denied to authorized users. It involves properly maintaining hardware and technical infrastructure that systems work promptly and service is not denied to authorized users.

Best Practices for implementing Confidentiality

- Data should be handled based on the organization's required privacy.
- Data should be encrypted.
- Keep access control lists and other file permissions up to date.

Best Practices for implementing Integrity

- Ensure employees are knowledgeable about compliance and regulatory requirements to minimize human error.
- Use backup and recovery software.
- To ensure integrity, use version control, access control, security control, data logs and checksums.

 U2.4

Best Practices for implementing Availability

- Use preventive measures such as redundancy, failover and RAID.
- Ensure systems and applications stay updated.
- Use network or server monitoring systems.
- Ensure a data recovery and business continuity (BC) plan is in place in case of data loss.

 U2.5

E-Commerce Security Threats and Issues

- Financial Frauds
- Spam
- Phishing
- Bots
- DDoS Attacks
- Brute Force Attacks
- SQL Injections
- Cross Site Scripting (XSS)

 U2.6

Financial Frauds

- Financial fraud has afflicted online businesses since their inception.
- Hackers make unauthorized transactions and wipe out the trail costing businesses significant amounts of losses.
- Some fraudsters also file requests for fake refunds or returns.
- Refund fraud is a common financial fraud where businesses refund illegally acquired products or damaged goods.

Spam

- Emails are one of the highly used mediums for spamming.
- Nonetheless, comments on your blog or contact forms are also an open invitation for online spammers where they leave infected links in order to harm you.
- Spamming not only affects your website's security, but it also damages your website speed too.

Phishing

- It is one of the common security threats of ecommerce where hackers masquerade as legitimate businesses and send emails to your clients to trick them into revealing their sensitive information by simply presenting them with a fake copy of your legitimate website or anything that allows the customer to believe the request is coming from the business.
- Common phishing techniques include emailing your customers or your team with fake "you must take this action" messages.
- This technique only works your customers follow through with the action and provide them access to their login information or other personal data.

Bots

- Exclusive bots are developed to scrape websites for their pricing and inventory information.
- The hackers use such information to change the pricing of your online store, or to store the best-selling inventory in shopping carts, resulting in a decline in sales and revenue.

DoS Attacks

- Distributed Denial of Service (DDoS) attacks and DoS (Denial of Service) attacks aim to disrupt the e-commerce website and affect overall sales.
- These attacks flood the servers with numerous requests until they succumb to them and the e-commerce website crashes.

Brute Force Attacks

- These attacks target online store's admin panel in an attempt to figure out the password by brute-force.
- It uses programs that establish a connection to e-commerce website and use every possible combination to crack your password.
- We can protect our-self against such attacks by using a strong and complex password.

SQL Injections

- SQL injections are cyber-attacks intended to access your database by targeting your query submission forms.
- They inject malicious code in your database, collect the data and then delete it later on.

Cross Site Scripting (XSS)

- XSS is an attack in which an attacker injects malicious executable scripts into the code of a trusted application or website.
- Attackers often initiate an XSS attack by sending a malicious link to a user and enticing the user to click it.
- Potential consequences of cross site scripting attacks include:
  - Capturing the keystrokes of a user.
  - Redirecting a user to a malicious website.
  - Running web browser-based exploits (e.g., crashing the browser).
  - Obtaining the cookie information of a user who is logged into a website (thus compromising the victim's account).

Important Security Mechanisms

- **Encipherment**
  - The use of mathematical algorithms to transform data into a form that is not readily intelligible.
  - The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.
- **Digital Signature**
  - Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).

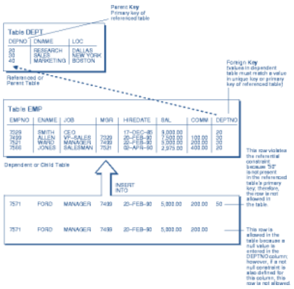
Important Security Mechanisms (contd…)

- **Security Audit Trail**
  - Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
- **Access Control**
  - A security safeguard (i.e., hardware and software features, physical controls, operating procedures, management procedures, and various combinations of these) designed to detect and deny unauthorized access and permit authorized access to an information system or physical facility.

Important Security Mechanisms (contd…)

- **Data Integrity**
  - A variety of mechanisms used to assure the integrity of a data unit or stream of data units.
  - **Data Integrity for Databases**
    - Domain Integrity: The domain integrity of a database refers to the common ways to input and read this data. For instance, if a database uses monetary values to include dollars and cents, three decimal places will not be allowed.
    - Referential Integrity: Foreign keys in a database is a second table that can refer to a primary key table within the database.
    - User-Defined Integrity: There are sets of data, created by users, outside of entity, referential and domain integrity.

Data Integrity for Databases



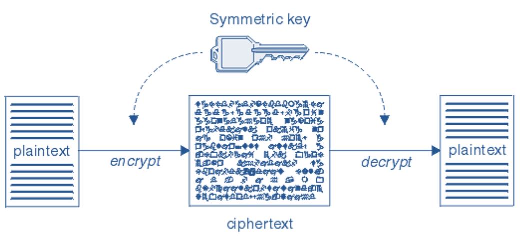
Cryptography

- Cryptography is the process of converting between readable text, called plaintext, and an unreadable form, called ciphertext.
- This occurs as follows:
  - The sender converts the plaintext message to ciphertext. This part of the process is called encryption (sometimes encipherment).
  - The ciphertext is transmitted to the receiver.
  - The receiver converts the ciphertext message back to its plaintext form. This part of the process is called decryption (sometimes decipherment).

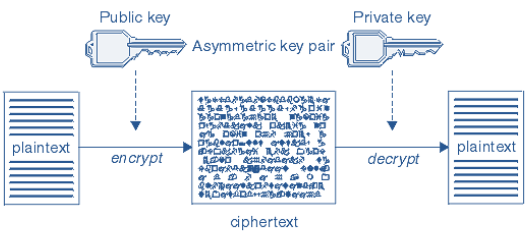
Cryptography, Cryptanalysis and Cryptology

- Schemes used for encryption constitute the area of study known as cryptography.
  - Such a scheme is known as a cryptographic system or a cipher.
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of cryptanalysis.
  - Cryptanalysis is what the layperson calls "breaking the code."
- The areas of cryptography and cryptanalysis together are called cryptology.

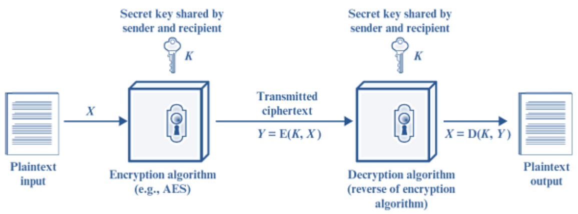
Private (Symmetric) Key Cryptography



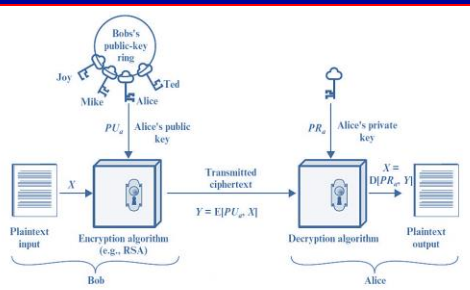
Public (Asymmetric) Key Cryptography



Simplified Model of Symmetric Encryption



Simplified Model of Asymmetric Encryption



Classical Encryption Technique

- Caesar Cipher (substitution of a ciphertext symbol for a plaintext symbol)
  - The earliest known, and the simplest, use of a substitution cipher was by Julius Caesar.
  - The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.
  - Example:

```
plain:  meet me after the toga party
cipher: PHHW PH DIWHU WKH WRJD SDUWB
```


Transposition Encryption Technique

- Rail Fence (plaintext is written down as a sequence of diagonals)
  - The plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.
  - Example:

Plain Text: meet me after the toga party

Rail Fence (Depth 2):

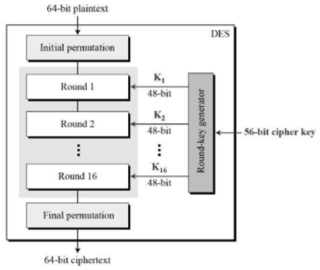
```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

Encrypted Message: MEMATRHTGPRYETEFETEOAAT

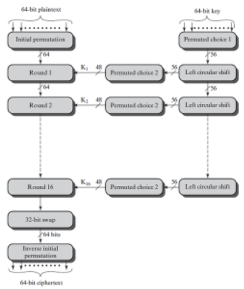
Data Encryption Standard

- Until the introduction of the Advanced Encryption Standard (AES) in 2001, the Data Encryption Standard (DES) was the most widely used encryption scheme.
- DES is a block cipher - Data are encrypted in 64-bit blocks using a 56-bit key.
  - The keys are actually stored as being 64 bits long, but every 8th bit in the key is not used.
  - The eight bits just mentioned get eliminated when we create sub-keys.
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- The same steps, with the same key, are used to reverse the encryption.

Data Encryption Standard (contd…)



Data Encryption Standard (contd…)



Data Encryption Standard Example

Let M be the plain text message M = 0123456789ABCDEF, where M is in hexadecimal (base 16) format.

Rewriting M in binary format, we get the 64-bit block of text:

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
L = 0000 0001 0010 0011 0100 0101 0110 0111
R = 1000 1001 1010 1011 1100 1101 1110 1111

Let K be the hexadecimal key K = 133457799BBCDFF1. This gives us as the binary key:

K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

Data Encryption Standard - Example (contd…)

- Step 1: Create 16 sub-keys, each of which is 48-bits long.
  - The 64-bit key is permuted according to the following table:

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

  - Since the first entry in the table is "57", this means that the 57th bit of the original key K becomes the first bit of the permuted key K+. The 49th bit of the original key becomes the second bit of the permuted key. The 4th bit of the original key is the last bit of the permuted key.
  - Note: Only 56 bits of the original key appear in the permuted key.

Data Encryption Standard - Example (contd…)

- From the original 64-bit key:
  - K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
- We get the 56-bit permutation:
  - K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111
- Next, split this key into left and right halves, C_0 and D_0, where each half has 28 bits.
  - C_0 = 1111000 0110011 0010101 0101111
  - D_0 = 0101010 1011001 1001111 0001111

Data Encryption Standard - Example (contd…)

- Now, create sixteen blocks C_n and D_n from the previous pair C_{n-1} and D_{n-1}, respectively, using the following schedule of "left shifts" of the previous block.

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

For example, $\mathbf{C_3}$ and $\mathbf{D_3}$ are obtained from $\mathbf{C_2}$ and $\mathbf{D_2}$, respectively, by two left shifts, and $\mathbf{C_{16}}$ and $\mathbf{D_{16}}$ are obtained from $\mathbf{C_{15}}$ and $\mathbf{D_{15}}$, respectively, by one left shift.

Data Encryption Standard - Example (contd…)

- From original pair C_0 and D_0
  $C_0 = 1111000011001100101010101111$
  $D_0 = 0101010101100110011110001111$
  , we obtain:

$C_1 = 1110000110011001010101011111$
$D_1 = 1010101011001100111100011110$

$C_2 = 1100001100110010101010111111$
$D_2 = 0101010110011001111000111101$

$C_3 = 0000110011001010101011111111$
$D_3 = 0101011001100111100011110101$

$C_4 = 0011001100101010101111111100$
$D_4 = 0101100110011110001111010101$

$C_5 = 1100110010101010111111110000$
$D_5 = 0110011001111000111101010101$

$C_6 = 0011001010101011111111000011$
$D_6 = 1001100111100011110101010101$

$C_7 = 1100101010101111111100001100$
$D_7 = 0110011110001111010101010110$

$C_8 = 0010101010111111110000110011$
$D_8 = 1001111000111101010101011001$

$C_9 = 0101010101111111100001100110$
$D_9 = 0011110001111010101010110011$

$C_{10} = 0101010111111110000110011001$
$D_{10} = 1111000111101010101011001100$

$C_{11} = 0101011111111000011001100101$
$D_{11} = 1100011110101010101100110011$

$C_{12} = 0101111111100001100110010101$
$D_{12} = 0001111010101010110011001111$

$C_{13} = 0111111110000110011001010101$
$D_{13} = 0111101010101011001100111100$

$C_{14} = 1111111000011001100101010101$
$D_{14} = 1110101010101100110011110001$

$C_{15} = 1111100001100110010101010111$
$D_{15} = 1010101010110011001111000111$

$C_{16} = 1111000011001100101010101111$
$D_{16} = 0101010101100110011110001111$

Data Encryption Standard - Example (contd…)

- We now form the keys K_n, for 1 <= n <= 16, by applying the following permutation table to each of the concatenated pairs C_nD_n.

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

- Each pair has 56 bits, but PC-2 only uses 48 of these.
- Therefore, the first bit of K_n is the 14^{th} bit of C_nD_n, the second bit the 17^{th}, and so on, ending with the 48^{th} bit of K_n being the 32^{th} bit of C_nD_n.

Data Encryption Standard - Example (contd…)

- For the first key, we have

  C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

  which, after we apply the permutation table, becomes:

  K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

  Similarly,

  K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
  K_3 = 010101 011111 110010 001010 010000 101100 111110 011001
  …
  …
  K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101

Data Encryption Standard - Example (contd…)

- Step 2: Encode each 64-bit block of data.
  - There is an initial permutation (IP) of the 64 bits of the message data M.
  - This rearranges the bits according to the following table,

IP

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

where, the entries in the table show the new arrangement of the bits from their initial order.

The 58^{th} bit of M becomes the first bit of IP. The 50^{th} bit of M becomes the second bit of IP. The 7^{th} bit of M is the last bit of IP.

Data Encryption Standard - Example (contd…)

- Applying the initial permutation to the block of text M, given previously, we get:

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

- Next, divide the permuted block IP into a left half L_0 of 32 bits, and a right half R_0 of 32 bits. From IP, we get L_0 and R_0

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

Data Encryption Standard - Example (contd…)

- Now, proceed through 16 iterations, for 1<=n<=16, using a function ***f*** which operates on two blocks: **a data block of 32 bits**, and **a key K_n of 48 bits** to produce a block of 32 bits.
- Let + denote XOR addition, (bit-by-bit addition modulo 2), then, for n (from 1 to 16):

$L_n = R_{n-1}$

$R_n = L_{n-1}$ XOR $f(R_n\text{-}1,K_n)$

- We take the right 32 bits of the previous result and make them the left 32 bits of the current step. For the right 32 bits in the current step, we XOR the left 32 bits of the previous step with the calculation ***f***.

Data Encryption Standard - Example (contd…)

- For n = 1, we have:
  - K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
  - $L_1 = R_0$ = 1111 0000 1010 1010 1111 0000 1010 1010
  - $R_1 = L_0 + f(R_0, K_1)$
- To calculate *f*, we first expand each block R_{n-1} from 32 bits to 48 bits. This is done by using a selection table that repeats some of the bits in R_{n-1} .
  - We call the use of this selection table the function **E**.
  - Thus, $E(R_{n-1})$ has a 32 bit input block, and a 48 bit output block.

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

Data Encryption Standard - Example (contd…)

- Using the Selection Table, we calculate $E(R_0)$ from R_0 as follows:
  - R_0 = 1111 0000 1010 1010 1111 0000 1010 1010
  - $E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
- Next, in the *f* calculation, we XOR the output $E(R_{n-1})$ with the key K_n:
  - K_n XOR $E(R_{n-1})$
- For K_1 and $E(R_0)$, we have:
  - K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
  - $E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
  - K_1 XOR $E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111
- To this point, R_{n-1} is expanded from 32 bits to 48 bits, using the selection table, and XORed the result with the key K_n .

Data Encryption Standard - Example (contd…)

- Now, we have 48 bits, or eight groups of six bits.
- Next, 4 bit number will replace the original 6 bits.
  - The net result is that the eight groups of 6 bits are transformed into eight groups of 4 bits for 32 bits total.
  - Use each group of six bits as addresses in tables called "S boxes".
  - Each group of six bits will give us an address in a different S box.
  - Located at that address will be a 4 bit number.
- Write the previous result, which is 48 bits, in the form:
  - $K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$, where each B_i is a group of six bits.

Data Encryption Standard - Example (contd…)

- Now, we calculate $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ where $S_i(B_i)$ referes to the output of the i^{th} S-box.
- Each of the functions $\mathbf{S_1, S_2, \ldots, S_8}$, takes a 6-bit block as input and yields a 4-bit block as output.
- The table to determine $\mathbf{S_1}$ is shown and explained below:

S1

Column Number

Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

 U2.43

Data Encryption Standard - Example (contd…)

- $S_1(B)$ is determined as follows:
  - The first and last bits of B represent, in base 2, a number in the decimal range 0 to 3 (or binary 00 to 11). Let that number be i.
  - The middle 4 bits of B represent in base 2 a number in the decimal range 0 to 15 (binary 0000 to 1111). Let that number be j.
  - Look up in the table the number in the i^{th} row and j^{th} column. It is a number in the range 0 to 15 and is uniquely represented by a 4 bit block.
  - For example, for input block **B = 011011** the first bit is "0" and the last bit "1" giving 01 as the row. This is row 1. The middle four bits are "1101". This is the binary equivalent of decimal 13, so the column is column number 13. In row 1, column 13 appears 5. This determines the output; 5 is binary 0101, so that the output is 0101. Hence $S_1(011011) = 0101$.

 U2.44

Data Encryption Standard - Example (contd…)

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

 U2.45

Data Encryption Standard - Example (contd…)

- For the first round, we obtain as the output of the eight **S** boxes:
  - $K_1 + E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111.
  - $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111
- The final stage in the calculation of ***f*** is to do a permutation **P** of the **S**-box output to obtain the final value of ***f***:
  - $f = P(S_1(B_1)S_2(B_2)...S_8(B_8))$
- The permutation **P** is defined in the following table:

P

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

 U2.46

Data Encryption Standard - Example (contd…)

- From the output of the eight **S** boxes:
  - $S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111 ,

  we get
  - $f(R_0, K_1)$ = 0010 0011 0100 1010 1010 1001 1011 1011
  - $R_1 = L_0 + f(R_0, K_1)$ = 1100 1100 0000 0000 1100 1100 1111 1111 + 0010 0011 0100 1010 1010 1001 1011 1011 = **1110 1111 0100 1010 0110 0101 0100 0100**
  - **ROUND 1 COMPLETED HERE**

 U2.47

Data Encryption Standard - Example (contd…)

- At the end of the sixteenth round, we have the blocks L_{16} and R_{16}.
  - We then reverse the order of the two blocks into the 64-bit block $R_{16}L_{16}$ and apply a final permutation IP^{-1} as defined by the following table:

IP^{-1}

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

  - If we process all 16 blocks using the method defined previously, we get, on the 16^{th} round,
  - L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
    R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101
  - $R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100
  - IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101
  - = **85E813540F0AB405** (in hexadecimal form)

 U2.48

Rivest-Shamir-Adleman (RSA) Scheme

- The RSA is a scheme in which the plaintext and ciphertext are integers between 0 and n - 1 for some n.
- A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 2^{1024}.
- Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- The block size must be less than or equal to $\log_2(n) + 1$. In practice, the block size is i bits, where $2^i < n \le 2^{i+1}$.
- Encryption and decryption are of the following form

RSA Scheme (contd…)

- Encryption and decryption are of the following form:
  - $C = M^e \bmod n$
  - $M = C^d \bmod n$
- Both, sender and receiver must know the value of n.
- The sender knows the value of e, and only the receiver knows the value of d.
- Thus, this is a public key encryption algorithm with a public key of PU = {e, n} and a private key of PR = {d, n}.

RSA Scheme (contd…)

Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

RSA - Example

- Select two prime numbers, p = 17 and q = 11.
- Calculate n = pq = 17 × 11 = 187.
- Calculate ϕ(n) = (p - 1)(q - 1) = 16 * 10 = 160.
- Choose e **such that** 1 < e < ϕ(n) and e and ϕ (n) are co-prime; we choose e = 7.
- Determine d **such that** (d * e) % ϕ(n) = 1. The correct value of d is 23, because (23 × 7) % 160 = 1.
- Public key is (e, n) => (7, 187)
- Private key is (d, n) => (23, 187)
- The encryption of m = 88 is c = 88^7 % 187 = 11
- The decryption of c = 11 is m = 11^{23} % 187 = 88

Message Digests and Digital Signatures

- A message digest is a fixed size numeric representation of the contents of a message.
- The message digest is computed by a hash function and can be encrypted, forming a digital signature. The hash function used to compute a message digest must meet two criteria:
  - It must be one way. It must not be possible to reverse the function to find the message corresponding to a particular message digest.
  - It must be computationally infeasible to find two messages that hash to the same digest.
- The message digest is sent with the message itself.
- The receiver can generate a digest for the message and compare it with the digest of the sender.
- The integrity of the message is verified when the two message digests are the same.

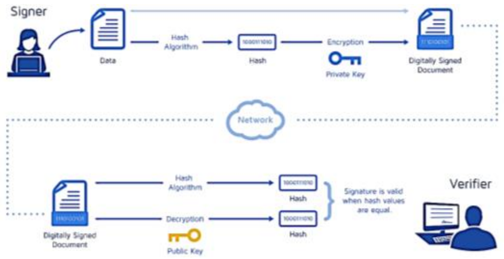
Message Digests and Digital Signatures

- A message digest created using a secret symmetric key is known as a Message Authentication Code (MAC).
- The sender can also generate a message digest and then encrypt the digest using the private key of an asymmetric key pair, forming a digital signature.
  - The signature must then be decrypted by the receiver, before comparing it with a locally generated digest.

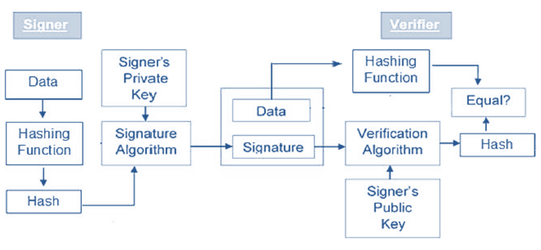
Digital Signature

- A digital signature - a type of electronic signature - is a mathematical technique routinely used to validate the authenticity and integrity of a message (e.g., an email, a credit card transaction, or a digital document).
- Digital signatures use a standard, accepted format, called Public Key Infrastructure (PKI), to provide the highest levels of security and universal acceptance.
  - PKI requires the provider to use a mathematical algorithm to generate two long numbers, called keys. One key is public, and one key is private.
  - Digital signatures are based on public key cryptography, also known as asymmetric cryptography. Using a public key algorithm, such as RSA, two keys are generated, creating a mathematically linked pair of keys, one private and one public.

Digital Signature (contd…)



Digital Signature (contd…)



Common Terms in Digital Signature

- Hash Function
- Public Key Cryptography
- Public Key Infrastructure (PKI)
- Certificate Authority (CA)
- Digital Certificates
- Pretty Good Privacy (PGP)/OpenPGP

Common Terms in Digital Signature (contd…)

- Hash Function
  - A hash function (also called a "hash") is a fixed-length string of numbers and letters generated from a mathematical algorithm and an arbitrarily sized file such as an email, document, picture, or other type of data.
  - This generated string is unique to the file being hashed and is a one-way function — a computed hash cannot be reversed to find other files that may generate the same hash value.
  - Some of the popular hashing algorithms in use today are Secure Hash Algorithm-1 (SHA-1), the Secure Hashing Algorithm-2 family (SHA-2 and SHA-256), and Message Digest 5 (MD5).

Common Terms in Digital Signature (contd…)

- Public Key Cryptography
  - Public key cryptography (also known as asymmetric encryption) is a cryptographic method that uses a key pair system.
  - One key, called the public key, encrypts the data.
  - The other key, called the private key, decrypts the data. Public key cryptography can be used several ways to ensure confidentiality, integrity, and authenticity.

Common Terms in Digital Signature (contd…)

- Public Key Infrastructure (PKI)
  - PKI consists of the policies, standards, people, and systems that support the distribution of public keys and the identity validation of individuals or entities with digital certificates and a certificate authority.

Common Terms in Digital Signature (contd…)

- Certificate Authority (CA)
  - A CA is a trusted third party that validates a person's identity and either generates a public/private key pair on their behalf or associates an existing public key provided by the person to that person.
  - Once a CA validates someone's identity, they issue a digital certificate that is digitally signed by the CA.
  - The digital certificate can then be used to verify a person associated with a public key when requested.

Common Terms in Digital Signature (contd…)

- Digital Certificates
  - Digital certificates protect against impersonation, certifying that a public key belongs to a specified entity.
  - They are issued by a Certificate Authority. They contain the public key of the individual or organization and are digitally signed by a CA.
  - Digital certificates are also known as public key certificates, because they give you assurances about the ownership of a public key when you use an asymmetric key scheme.
  - Other information about the organization, individual, and CA can be included in the certificate as well.

Common Terms in Digital Signature (contd…)

- Digital Certificates (contd…)
  - If public keys are sent directly by their owner to another entity, there is a risk that the message could be intercepted and the public key substituted by another.
  - The solution to this problem is to exchange public keys through a trusted third party, giving you a strong assurance that the public key really belongs to the entity with which you are communicating.
  - Instead of sending your public key directly, you ask the trusted third party to incorporate it into a digital certificate.
  - The trusted third party that issues digital certificates is called a Certificate Authority (CA).

Common Terms in Digital Signature (contd…)

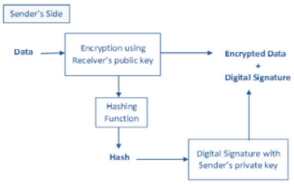
- Pretty Good Privacy (PGP)/OpenPGP
  - PGP/OpenPGP is an alternative to PKI.
  - With PGP/OpenPGP, users "trust" other users by signing certificates of people with verifiable identities.
  - The more interconnected these signatures are, the higher the likelihood of verifying a particular user on the Internet.
  - This concept is called the "Web of Trust."

Importance of Digital Signature

- Message Authentication – When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.
- Data Integrity – In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.
- Non-repudiation – Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

Encryption with Digital Signature

- In many digital communications, it is desirable to exchange an encrypted messages than plaintext to achieve confidentiality.
- In public key encryption scheme, a public key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.
- Combining digital signatures along with encrypted data assures message authentication and non-repudiation.



 U2.67

Use of PKI or PGP with Digital Signatures

- Using digital signatures in conjunction with PKI or PGP strengthens them and reduces the possible security issues connected to transmitting public keys by validating that the key belongs to the sender, and verifying the identity of the sender.
- The security of a digital signature is entirely dependent on how well the key is protected.
- Without PGP or PKI, proving someone's identity or revoking a compromised key is impossible; this could allow malicious actors to impersonate someone without any method of confirmation.
- Through the use of a trusted third party, digital signatures can be used to identify and verify individuals and ensure the integrity of the message.

 U2.68

Public Key Infrastructure

- Public Key Infrastructure (PKI) is a system of facilities, policies, and services that supports the use of public key cryptography for authenticating the parties involved in a transaction.
- Since the public keys are in open domain, they are likely to be abused. It is, thus, necessary to establish and maintain some kind of trusted infrastructure to manage these keys.
- It is observed that cryptographic schemes are rarely compromised through weaknesses in their design. However, they are often compromised through poor key management.
- The most crucial requirement of 'assurance of public key' can be achieved through the PKI, a key management systems for supporting public-key cryptography.

 U2.69

Public Key Infrastructure (contd…)

- PKI provides assurance of public key.
- It provides the identification of public keys and their distribution.
- PKI mainly comprises of the following components:
  - Public Key Certificate, commonly referred to as 'Digital Certificate'
  - Private Key Tokens
  - Certification Authority
  - Registration Authority
  - Certificate Management System

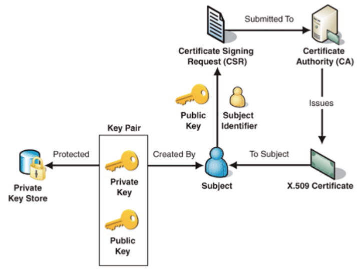
Digital Certificate

- People use ID cards such as a driver's license, passport to prove their identity.
- A digital certificate does the same basic thing in the electronic world, but with one difference.
  - Digital Certificates are not only issued to people but they can be issued to computers, software packages or anything else that need to prove the identity in the electronic world.
- Digital certificates are based on the ITU standard **X.509** which defines a standard certificate format for public key certificates and certification validation.
- Public key pertaining to the user is stored in digital certificates by the Certification Authority (CA) along with other relevant information such as client information, expiration date, usage, issuer etc.

Digital Certificate (contd…)

- CA digitally signs this entire information and includes digital signature in the certificate.
- Anyone who needs the assurance about the public key and associated information of client, he carries out the signature validation process using CA's public key.
- Successful validation assures that the public key given in the certificate belongs to the person whose details are given in the certificate.

Process of obtaining Digital Certificate



X.509 certificates can be revoked by the issuing CA if the integrity of the certificate has somehow been compromised.

Certifying Authority (CA)

- The CA issues certificate to a client and assist other users to verify the certificate.
- The CA takes responsibility for identifying correctly the identity of the client asking for a certificate to be issued, and ensures that the information contained within the certificate is correct and digitally signs it.
- The key functions of a CA are:
  - Generating Key Pairs
  - Issuing Digital Certificates
  - Publishing Certificates – The CA need to publish certificates so that users can find them.
  - Verifying Certificates – The CA makes its public key available in environment to assist verification of his signature on clients' digital certificate.
  - Revocation of Certificates

Classes of Digital Certificates

- Class 1 – These certificates can be easily acquired by supplying an email address.
- Class 2 – These certificates require additional personal information to be supplied.
- Class 3 – These certificates can only be purchased after checks have been made about the requestor's identity.
- Class 4 – They may be used by governments and financial organizations needing very high levels of trust.

Registration Authority (RA)

- CA may use a third-party Registration Authority (RA) to perform the necessary checks on the person or company requesting the certificate to confirm their identity.
- The RA may appear to the client as a CA, but they do not actually sign the certificate that is issued.

Certificate Management System (CMS)

- It is the management system through which certificates are published, temporarily or permanently suspended, renewed, or revoked.
- Certificate management systems do not normally delete certificates because it may be necessary to prove their status at a point in time, perhaps for legal reasons.
- A CA along with associated RA runs certificate management systems to be able to track their responsibilities and liabilities.

Private Key Tokens

- The public key of a client is stored on the certificate.
- The associated secret private key can be stored on the key owner's computer.
  - This method is generally not adopted.
  - If an attacker gains access to the computer, he can easily gain access to private key.
- For this reason, a private key is stored on secure removable storage token access to which is protected through a password.
- Different vendors often use different storage formats for storing keys.
  - For example, Entrust uses the proprietary .epf format, while Verisign, GlobalSign, and Baltimore use the standard .p12 format.

Secure Sockets Layer

- Secure Sockets Layer **(SSL)**, is an encryption-based Internet security protocol.
- It was first developed by Netscape in 1995 for the purpose of ensuring **privacy**, **authentication**, and **data integrity** in Internet communications.
- SSL is the direct predecessor to the modern **TLS** (Transport Layer Security) encryption used today.
  - In 1999, the Internet Engineering Task Force (IETF) proposed an update to SSL.
  - Since this update was being developed by the IETF and Netscape was no longer involved, the name was changed to TLS.
  - The differences between the final version of SSL (3.0) and the first version of TLS are not drastic; the name change was applied to signify the change in ownership.

Secure Sockets Layer (contd…)

- A website that implements SSL/TLS has "HTTPS" in its URL instead of "HTTP."

Secure Sockets Layer (contd…)

- The SSL/TLS protocol provides communications security over the Internet, and allow client/server applications to communicate in a way that is confidential and reliable.
- SSL/TLS uses asymmetric and symmetric cryptography techniques.
- An SSL or TLS connection is initiated by an application, which becomes the SSL or TLS client.
- The application which receives the connection becomes the SSL/TLS server.
- Every new session begins with a handshake, as defined by the SSL/TLS protocol.

Importance of SSL/TLS

- SSL encrypts data that is transmitted across the web.
  - This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt.
- SSL initiates an authentication process called a handshake between two communicating devices to ensure that both devices are really who they claim to be.
- SSL also digitally signs data in order to provide data integrity, verifying that the data is not tampered with before reaching its intended recipient.

SSL Certificate

- SSL can only be implemented by websites that have an SSL Certificate (technically a "TLS Certificate").
- An SSL certificate is like an ID card or a badge that proves someone is who they say they are.
- SSL certificates are stored and displayed on the Web by a website's or application's server.
- One of the most important pieces of information in an SSL certificate is the website's public key.
- A user's device views the public key and uses it to establish secure encryption keys with the web server.
- Meanwhile the web server also has a private key that is kept secret; the private key decrypts data encrypted with the public key.
- Certificate authorities (CA) are responsible for issuing SSL certificates.

Working of SSL

- SSL can only be implemented by websites that have an SSL Certificate (technically a "TLS Certificate").

Working of SSL (contd…)

- When a browser attempts to access a website that is secured by an SSL certificate, it recognizes the SSL, the web server, and the browser, thereby establishing a secure connection.
- This process is called the "SSL Handshake", which happens instantaneously and remains invisible to the users.

The process of encryption and decryption requires a lot of processing techniques. Therefore, it is used only during the "SSL handshake" to create a symmetric session key. After establishing a secure communication, the session key is used to encrypt all the transmitted data.

Types of SSL Certificates

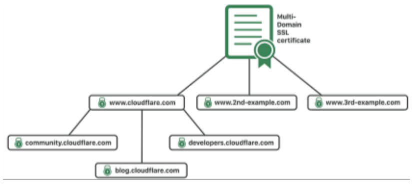
- Single Domain SSL Certificates
  - A single-domain SSL certificate applies to one domain and one domain only. It cannot be used to authenticate any other domain, not even subdomains of the domain it is issued for.

Types of SSL Certificates (contd…)

- Wildcard SSL Certificates
  - Wildcard SSL certificates are for a single domain and all its subdomains. A subdomain is under the umbrella of the main domain.

Types of SSL Certificates (contd…)

- Multi-Domain SSL Certificates (MDC)
  - A multi-domain SSL certificate, or MDC, lists multiple distinct domains on one certificate.
- Multi-Domain Wildcard SSL Certificates



Validation Levels of SSL Certificate

- There are different levels of validation, ranging from bare minimum validation to thorough background investigations.
  - An SSL certificate from any of these validation levels provides the same degree of TLS encryption; the only difference is how thoroughly the CA has authenticated the organization's identity.
  - Domain Validation SSL Certificates - This is the least-stringent level of validation, and the cheapest. All a business has to do is prove they control the domain.
  - Organization Validation - This is a more hands-on process: The CA directly contacts the person or business requesting the certificate. These certificates are more trustworthy for users.
  - Extended Validation - This requires a full background check of an organization before the SSL certificate can be issued.

HTTPS

- HTTPS refers to HTTP with the TLS encryption protocol.
- HTTPS uses both types of encryption (symmetric or asymmetric).
  - All communications over TLS start with a TLS handshake. Asymmetric encryption is crucial for making the TLS handshake work.
  - During the course of a TLS handshake, the two communicating devices establish four session keys, and these will be used for symmetric encryption for the rest of the session.

Firewall

- A firewall is a network security device (software programs or hardware devices) that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.
- When our computer has firewall protection, everything that goes in and out of - it is monitored.
- The firewall monitors all this information traffic to allow 'good data' in, but block 'bad data' from entering our computer.

Firewall (contd…)

- Hardware Firewalls
  - A hardware firewall is a system that works independently from the computer it is protecting as it filters information coming from the internet into the system.
  - If you have a broadband internet router, it likely has its own firewall.
- Software Firewalls
  - A software firewall is a program used by a computer to inspect data that goes in and out of the device.
  - It can be customized by the user to meet their needs.

Types of Firewalls

- Packet-Filtering Firewalls
  - The most basic form of firewall software uses pre-determined security rules to create filters – if an incoming packet of information (small chunk of data) is flagged by the filters, it is not allowed through.
  - Packets that make it through the filters are sent to the requesting system and all others are discarded.
  - Because all web traffic is allowed, a packet-filtering firewall does not block web-based attacks.
  - Therefore, we need additional protection to distinguish between friendly and malicious web traffic.

Types of Firewalls (contd…)

- Proxy Service Firewalls
  - The proxy service firewall is a system that can help protect our network security by filtering messages at the application layer.
  - A proxy firewall is like a mirror of our computer and detects malicious actors attempting to get through to our device.
  - It essentially serves as a gateway or middle man between our internal network and outside servers on the web.
  - Proxy firewalls are a secure solution because of the separation they provide between our computer and the Internet.
    - Attackers often need to connect directly to our computer to attack it. Because a proxy is between our computer and the Internet, hackers cannot form a direct connection to it, rendering their attack useless.

Types of Firewalls (contd…)

- Stateful Inspection
  - A stateful inspection firewall inspects every data packet and compares it against a threat database.
  - During the inspection process, the firewall checks where the data is coming from, the ports it uses, and the applications it is associated with.
  - If the data packet checks out, it is allowed to pass. Otherwise, it is discarded.