Name :- Kanchan Sharma

University Roll No :- 1121070

Subject Name :- Information Security and Cyber Laws

Subject Code :- PBC 601.

MCQs :-

1- Asymmetric key encryption with sender public key.

2- spyware

3- an authentication of an electronic record.

4- cyber laws.

5- only on alphanumeric data.

6- Idea is same, Title is different.

7- Checksum.

8- option (a) and (c) are right.

9- both (b) and (c)

10- none.

1. Problem Statement :-
Analyze the security aspects of your Google Account.

Objective :- A google account is the key to accessing all of Google's product and services. So, main objective is to control the information for Google account holder.

(A) Create a Google Account to access to many Google products.

Step1: Go to offical site of Google account for sign-in.
Step2: Click on create account and create your google account by filling necessary details.
Step3: Create password.
Step4: Account created successfully.
My email id is ks23032001@gmail.com

(B) Change your Google Account Password.

Step 1: Add or update recovery phone number and email address are powerful security tools. Alert you if there's suspicious activity on your account.

Step 2: Verification helps prevent a hacker from getting into your account, even if they steal your password. To avoid common phishing techniques associated with text msg codes, choose a stronger second verification step.

(i) Security keys.
(ii) Security prompts.

Step 3: Update your browser.

(i) Make sure you're using the latest version of browser.
(ii) Update Android devices.
(iii) Update chrome books.

Step 4: Use unique strong password.

It's risky to use the same password on multiple sites. If your password for one site is hacked, it could be used to get into your accounts for multiple sites.

Make sure to create a strong password, unique password for each account.

4- OTP.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main()
{
    char plaintext[100], otp[100];
    printf("enter plain text : ");
    fflush(stdin);
    fgets(plaintext, sizeof(plaintext), stdin);

    printf("enter otp - text of length %d \n",
           strlen(plaintext));
    fflush(stdin);
    fgets(otp, sizeof(otp), stdin);

    for(int i=0; i<strlen(plaintext); i++)
    {
        if(isupper(plaintext[i])
        {
            otp[i] = toupper(otp[i]);
            if(plaintext[i]+(otp[i]-'A')>'Z')
            { plaintext[i] = plaintext[i]+(otp[i]-
              'A')-26; }
```

```
if (plaintext[i] + (otp[i] - 'A') <= 'z') { plaintext[i] =
    plaintext[i] = plaintext[i] + (otp[i] - 'A'); }

}
else if (islower(plaintext[i]))
{
    otp[i] = tolower(otp[i]);
if (plaintext[i] + (otp[i] - 'a') > 'z') { plaintext[i] =
  plaintext[i] + (otp[i] - 'a' - 26; }
if (plaintext[i] + (otp[i] - 'a') <= 'z') { plaintext[i] =
    plaintext[i] + otp[i] - 'a'); }

}
else { plaintext[i] = plaintext[i]; }
printf("cypher-text is \t %s \n", plaintext);

return 0;

}
```

5- Encryption using caesar cipher:-

```
def encrypt (string);
  cipher = " "
for char in string;
    if char == ' ';
      cipher = cipher + char
    elif char.isupper ():
      cipher = cipher + char ((ord (char) + 3-65) %.26
                              +65);
    else:
    cipher = cipher + chr((ord (char) + 3-97) %.26
                          + 97);
    return cipher
text = " Attack from North"
print (" After encryption:" encrypt (text)).
```

Decryption using caesar cipher:-

```
def decrypt (string):
    plain = " "
    for char in string:
        if char == ' ':
            plain = plain + char
```

```
elif char.isupper():
    plain = plain + chr(ord(char) - 3 - 65) % 26 + 65)

else:
    plain = plain + chr((ord(char) - 3 - 97) % 26
                        + 97)

return plain

text = "                    "
print("After decryption:", decrypt(text)).
```