

Name → Kanika Bisht

University R.No → 1121071

Subject Name → Information Security & Cyber laws

Subject Code → TBC 601

MCQ

1 → PGP uses -

Ans ~~Asymmetric~~ symmetric key encryption with receiver public key.

2 → Keyloggers are a form of —?

Ans ~~malware~~ spyware

3 - A digital signature is

Ans encrypted signature of a sender

4 - "NETIQUETTES" deals with

Ans Cyber laws

5 - Encryption can be done

Ans only on alphanumeric

6 - — does not come under the copyright law infringement

Ans Idea is same content is different

7. MD5 is a —?

Ans hash value

8. In Affine Cipher

Ans The identity of the character is changed while its position remains unchanged.

9. The reason behind appending 'x' in playfair cipher is —

Ans to make even no. of letters

10. Module m taken as 26 in substitution cipher because of —

Ans Possibility of replacements

① Security Aspects of Google Account

● we can secure our account by following the security tips provided by Google

② Do a security checkup

• Go to security checkup to get personalized security recommendations for google account, including

- Add or update account recovery options
- Turn on 2-step verification.
- Remove risky access to data.
- Turn on screen locks.

③ Remove apps or unwanted browser extensions.

As more apps are installed on a device, it can become vulnerable. Install only essential apps & browser extensions on devices that have access to sensitive information. Avoid installing unknown apps from unknown sources to protect devices & personal info.

③ Track & delete the information.

Use Activity Control to choose what kinds of activity get saved in our account.

Examples are :-

- searches we do
- Websites we visit.
- Videos we watch.
- Places we go.
- Deleting history & cookies from chrome
- Delete web & app activity.
- Automatically delete youtube history & location history.
- Check third-party access.
- Review ad settings.

8. #include <stdio.h>
#include <string.h>
int main()

{

char msg[] = "CRYPTOGRAPHY";

char key[] = "MONARCHY";

int msglen = strlen(msg), keylen = strlen(key),
i, j;

char newkey[msglen], enmsg[msglen], demsg[msglen];

for (i=0, j=0; i < msglen; ++i, ++j)

{
if (j == keylen)
j = 0;
newkey[i] = key[j];
}

newkey[i] = '\0';

for (i=0; i < msglen; ++i)

{
enmsg[i] = ((msg[i] + newkey[i]) % 26) + 'A';
}

enmsg[i] = '\0';

```
for (i=0; i<msglen; ++i)
```

```
{
```

```
    demsg[i] = ((enmsg[i] - newkey[i] + 26) % 26)
```

```
    + 'A';
```

```
}
```

```
    demsg[i] = '\0';
```

```
printf("Original msg: %s", msg);
```

```
printf("In Key : %s", key);
```

```
printf("New generated key : %s", newkey);
```

```
printf("Encrypted msg : %s", enmsg);
```

```
printf("Decrypted msg : %s", demsg);
```

```
return 0;
```

```
}
```


⑤ Caesar cipher.

```
#include <stdio.h>
```

```
#include <string.h>
```

```
{
```

```
    char msg[100], ch;
```

```
    int i, key;
```

```
    printf("Enter message:");
```

```
    gets(msg);
```

```
    printf("Enter key:");
```

```
    for (i=0; msg[i] != '\0'; ++i)
```

```
    {
```

```
        ch = msg[i];
```

```
        if (ch >= 'a' && ch <= 'z')
```

```
        {
```

```
            ch = ch + key;
```

```
            if (ch > 'z')
```

```
            {
```

```
                ch = ch - 'z' + 'a' - 1;
```

```
            }
```

```
            msg[i] = ch;
```

```
        }
```

```
    }
```

```
    printf("Encrypted msg: %s", msg);
```

```
for (i=0; msg[i] != '\0'; ++i)
```

```
{  
    ch = msg[i];
```

```
    if (ch >= 'a' && ch <= 'z')
```

```
    {  
        ch = ch - key;
```

```
        if (ch < 'a')
```

```
            ch = ch + 'z' - 'a' + 1;
```

```
    }
```

```
    msg[i] = ch;
```

```
}
```

```
}
```

```
printf("Decrypted msg : %s", msg);  
return 0;
```

```
}
```