**Information Security Lab [15B17CI576] Odd 2022**

|  |  |  |
| --- | --- | --- |
| **COURSE OUTCOMES** | | **COGNITIVE LEVELS** |
| C374.1 | Demonstrate and illustrate the different cipher techniques and understand various anti-virus and anti worms | Level-2  (Understanding Level) |
| C374.2 | Develop and make a code to implement various Symmetric key , Asymmetric key cryptographic techniques and steganography techniques | Level-3  (Applying Level) |
| C374.3 | Apply a client server programming for DES and RSA algorithm. | Level-3  (Applying Level) |
| C374.4 | Examine and analyze the packet information for different protocols using Wireshark. | Level-4  (Analyzing Level) |
| C374.5 | Apply security techniques to real world problems | Level-3  (Applying Level) |

**Note: Students are advised to submit their solutions to respective lab faculty. The solution file must be named as “rollno\_first name\_w1.doc” (here w1 represents week1).**

**Week 1(3 August – 9 August)**

1. Explore the following terms with respect to information security:

* Cryptography
* Cryptanalysis
* Attacks
* Encryption/decryption
* Plain text/ cipher text
* Key: symmetric /asymmetric

2. Explore following classical symmetric key encryption techniques:

* Substitution encryption

1. Caesar Cipher - **Shift the ASCII Code**
2. Monoalphabetic Cipher- **Provide permanent letter**
3. Playfair Cipher- **Use matrix to encrypt a message with certain rules**
4. Hill Cipher- **Generate key matrix and multiply them with the message matrix**
5. Polyalphabetic Cipher – **Uses Multiple keys; first time a->d 2nd time a->n ; more security**
6. One Time Pad- **Randomly generated keys, can be used only once**

* Transposition encryption

1. Rail fence cipher-
2. Understand the following codes and examine which encryption techniques is used :

CODE1:

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

intkey,i;

char data[30];

clrscr();

printf("\nEnter the plain text: ");

gets(data);

printf("\nEnter the key value: ");

scanf("%d",&key);

for(i=0;i<strlen(data);i++)

{

if(data[i]!=' ')

{

if(data[i]>= data[strlen(data)-1-key])

{

data[i]=data[i]-26;

}

data[i]=data[i]+key;

}

}

printf("Your cipher text is: %s",data);

getch();

}

CODE 2:

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char s[30],k[27],c[30];

inti, index;

clrscr();

printf("Enter plain text: ");

gets(s);

printf("\nEnter key with 26 character:");

for(i=0;i<26;i++)

{

printf("\n%c",i+97);

k[i]=getch();

printf("%c",k[i]);

}

for(i=0;i<strlen(s);i++)

{

index=s[i]-97;

c[i]=k[index];

}

printf("Your cipher text is:");

for(i=0;i<strlen(s);i++)

{

printf("%c",c[i]);

}

getch();

}

* **Caesar Cipher**

#include <iostream>

using namespace std;

int main()

{

string msg;

cout<<"Enter a string "<<endl;

getline(cin, msg);

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

{

msg[i]=msg[i]+4;

}

cout<<endl;

cout<<"Encrypted string "<<msg<<endl;

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

{

msg[i]=msg[i]-4;

}

cout<<"Decrypted string "<<msg<<endl;

cout<<endl;

return 0;

}

* **Monoalphabetic Cipher**

#include <iostream>

using namespace std;

int binarySearch(string array, int left, int right, char item){

if (right >= left){

// calculation of new mid

int mid = left + (right - left)/2;

// returns position where found

if (array[mid] == item)

return mid+1;

// goes to recursive calls in left half

if (array[mid] > item)

return binarySearch(array, left, mid-1, item);

// goes to recursive calls in right half

else

return binarySearch(array, mid+1, right, item);

}

// if element is not found we return -1

else

return -1;

}

int main()

{

int index;

string ltrs= "abcdefghijklmnopqrstuvwxyz";

string cipher= "qwertyuiopasdfghjklzxcvbnm";

string msg;

cout<<"Enter your message"<<endl;

getline(cin, msg);

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

{

index= binarySearch(ltrs,0,26, msg[i]);

msg[i]=cipher[index-1];

}

cout<<"Encrypted string "<<msg<<endl;

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

{

index= binarySearch(cipher,0,26, msg[i]);

msg[i]=ltrs[index-1];

}

cout<<"Decrypted string "<<msg<<endl;

return 0;

}