

Project Report

E-VOTING THROUGH HOMOMORPHIC ENCRYPTION

Submitted by

K.SANJEETH

Under the guidance of

Prof. AMUTHA PRABAKAR M

Bachelor of Technology

in

Computer Science and Engineering



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

NOVEMBER 2020

Introduction

Theoretical Background

A manual voting is one where the votes are recorded manually, but is very time consuming when it has to be carried out. Before declaring the results, it takes huge amount of time to count the number of votes. It also becomes difficult for physically challenged people to cast their vote through paper ballot system as they require someone to cast their vote on behalf. Casting votes using paper ballot is a time consuming task and risky parallelly. There might be a chance that people can easily insert bogus paper votes in the ballot and then it becomes impossible to track the honest votes.

Motivation

In contrast to some of the demerits of Manual voting system, It asks the implementation of E-voting system for the same. E-voting is meant to casting of the votes online over the network. But what if any amendments are done unethically to the count of votes. Then there comes the need to use homomorphic encryption to protect voting privacy. E-voting using homomorphic encryption is an application which will be developed with the prime idea to protect the voting privacy of the people and reveal the number of votes polled for each candidate without generally counting plain data.

Aim of the proposed Work

The votes polled by voters are encrypted and using homomorphic encryption technique without idea of plain data directly computation is performed on encrypted data and results are declared with plain output data by decryption. So, the voting privacy is protected and its an efficient way to eradicate many problems with general voting. We can even reach remote locations with e- voting. People can cast their vote from any place in globe over the internet. People can open the authorized website and can easily cast their vote to their candidate at any place and anytime they want.

Proposed Statement

We proposed an E-Voting mechanism using Paillier Homomorphic Encryption technique which can be used to provide security to the voting system and to help us manipulate and transfer data in encrypted form to other authorized place making it impenetrable by others. Here we use the

Paillier Encryption Homomorphic property that allows us to add the votes in encrypted form and when decrypted will give us original number of votes polled after computation on cipher data.

Paillier cryptosystem has an important feature of homomorphic property. Paillier's has additive homomorphism which makes it an excellent choice to add the results of an election. In Paillier's homomorphic addition the product of two cipher data will decrypt to the sum of their corresponding plain data. So this property of pailliers can be used to design the e-voting.

Literature Survey

- Iram Ahmed and Archana Khandekar in their research work "Homomorphic encryption method applied for cloud computing" of International journal of Information and Computation technology proposed proxy re-encryption algorithm.

Improvements made in their work:

They proposed an algorithm with combination of both paillier and RSA cryptosystem. The paillier encrypted text will be again encrypted with public key.

Limitations:

As the key size for encryption becomes too large it is quite difficult to encrypt it frequently when required.

Link:

- Ramachandran, Sridevi, Srivani in their research work "A survey report on partially homomorphic encryption techniques" proposed all the possible cryptosystems which can be used in homomorphic encryption.

Improvements made in their work:-

They proposed all the possible cryptosystems which can be used while performing homomorphic encryption whenever required.

Limitation:

They did not mention anything about the limitations or efficiency which cryptosystem has over another cryptosystem.

Link: <https://www.ijert.org/phocadownload/V2I12/IJERTV2IS120681.pdf>

- N.sushmetha, S.vairamuthu, B. selvarani in their research work "A Case Study on Partial Homomorphic Encryption for Breast Cancer Diagnosis" of international journal of pure and applied mathematics proposed an idea such that privacy of health records is preserved.

Improvements made in their work:

It performs encrypted image comparison of the mammograms for users who don't have any signs or symptoms of breast cancer and Suitable image processing is carried out on the encrypted data.

Limitations:

The technique used is highly complicated to implement and any mistake in diagnosis can have a life. So it is still under research.

Link: [https:// HYPERLINK "https://acadpubl.eu/jsi/2018-119-7/articles/7a/16.pdf"](https://acadpubl.eu/jsi/2018-119-7/articles/7a/16.pdf)

- Ihsan Jabbar and Saad Najim Alsaad in their research work “Design and Implementation of Secure Remote e-Voting System Using Homomorphic Encryption” proposed an algorithm for voting privacy.

Improvements made in their work:-

It performs the e-voting through use of homomorphic property of ElGamal cryptosystem and it will be faster process than others like RSA cryptosystem.

Limitations:

It has larger keys which makes it complex while use and also not much efficient cryptosystem to choose with.

Link: [http:// HYPERLINK "http://ijns.jalaxy.com.tw/contents/ijns-v19-n5/ijns-2017-v19-n-p694-703.pdf"](http://ijns.jalaxy.com.tw/contents/ijns-v19-n5/ijns-2017-v19-n-p694-703.pdf)

Tools and methodologies

Software: Code Blocks

Language used:-

“C” Programming

Overview of the Proposed System

Framework, Architecture for the Proposed System(with explanation)

Paillier's Algorithm

This algorithm has a vast range of different applications like the banking security systems, in the area related to the cloud computing, etc... We used the above stated proposed system for implementation of an E Voting System.

Step 1: As it is assymetric algorithm, the pair of keys are generated.

- Take two prime numbers a and b randomly which should be independent of each other such that $\gcd(a*b, (a-1)*(b-1)) = 1$. GCD is the greatest common divisor of two or more integers which is the largest positive integer that divides the number without a remainder.
- 2. Compute $n = (a*b)$ and also $k(n) = \text{lcm}((p-1), (q-1))$ where, $k(n)$ is the Carmichael function.
- 3. You should select a generator ' g ' such that g belongs to the Z_n .
- 4. Calculate the follow Modular Multiplicative inverse $u = (L(g^k \bmod n^2))^{-1} \bmod n$ where $L(u) = (u-1)/u$.

So, pair of keys generated.

Public key: (n, g)

Private key: (k, u)

Step 2: encryption process

- The message m needs to be encrypted where m belongs to the ' Z '.
- Now, choose a random number r .
- Compute cipher text $c = (g^m * r^n) \bmod n^2$.

Step 3: decryption process

- Cipher text ' c ' should be decrypted to get required message m as follows by using the private key (k, u) such that : $m = L(c^k \bmod n^2) * k \bmod n$.

Implementation:-

Code:-

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>
#include<string.h>
#include<windows.h>
```

```
void decimal_to_binary(int d,int arr[]){
    int result,i=0;
    do{
```

```

        result=d%2;
        d/=2;
        arr[i]=result;
        i++;
    }while(d>0);
}

```

```

int lcm(int n1,int n2){
int minMultiple;
    minMultiple = (n1>n2) ? n1 : n2;
    while(1)
    {
        if( minMultiple%n1==0 && minMultiple%n2==0 )
        {
            break;
        }
        ++minMultiple;
    }
    return minMultiple;
}

```

```

int modular_exponentiation(int a,int b,int n){
    int *bb;
    int count=0,c=0,d=1,i;
    count=(int)(log(b)/log(2))+1;
    bb=(int*)malloc(sizeof(int)*count);

```

```

decimal_to_binary(b,bb);
for (i=count-1;i>=0;i--){
c=2*c;
d=(d*d)%n;
if (bb[i]==1){
c=c+1;
d=(d*a)%n;
}
}
return d;
}

```

```

void cast_vote(int* voter,int* candidate)
{

int darr[6]={0,0,0,0,0,0},i,n,c;
for(int j=0;j<6;j++){
begin:
system("cls");
printf("Welcome To The E-voting service.\n");
printf("Please Enter your VoterID: ");
scanf("%d",&n);
if(darr[n-1]==1){
printf("You have already casted your vote!! if not please check with election
committee");
printf("\nPress 1 to vote or press 0 to exit :");
scanf("%d",&i);
if(i==1)

```

[illegible]


```

else {
    printf("Candidate with this number is not there in list");
    goto begin;}
}

}
}

```

```

int main()
{
    FILE *fp;
    fp=fopen("C:\\Users\\saripella vivekanand\\Desktop\\1.txt","w");
    int candidate[3];
    for(int i=2;i>=0;i--)
    {
        candidate[2-i]=pow(2,2*i);
    }
    int p=5,q=7;
    int total=6;
    int voter[6]={0,0,0,0,0,0};
    cast_vote(voter,candidate);
    system("cls");
    int n1=0,n2=0,n3=0;
    for(int i=0;i<6;i++)
        if(voter[i]==candidate[0])
            n1++;
    int n,n2,lambda;

```

```

n=p*q;
n2=n*n;
lambda=lcm((p-1),(q-1));
int i,g;
g=141;
int r[6]={4,17,26,12,11,32};
int enc[6];
for(i=0;i<6;i++)
{

enc[i]=(((modular_exponentiation(g,voter[i],n2))*(modular_exponentiation(r[i],n,n
2))))%n2);
    fprintf(fp,"%d\n",enc[i]);
}
int y;

y(((((((enc[0]*enc[1])%n2)*((enc[2]*enc[3])%n2))%n2)*((enc[4]*enc[5])%n2))%
n2);
int L1,L2,L11,L22;
L1=modular_exponentiation(y,lambda,n2);
L2=modular_exponentiation(g,lambda,n2);
L11=(L1-1)/n;
L22=(L2-1)/n;

int dec,temp;
for(i=1;i<n;i++)
{
    dec=i;
    temp=((i*L22)%n);
    if(temp==L11)

```

```

        break;
    }
    if(n11>=2)
        dec=dec+n;
    int binary[100],cha;
    n=dec;
    i=0;
    while(n>0)
    {
        binary[i]=n%2;
        n=n/2;
        i++;
    }
    int v1,v2,v3;
    if(i==5)
    {
        if(binary[0]==1 && binary[1]==1)
            v3=3;
        else if(binary[0]==1 && binary[1]==0)
            v3=1;
        else if(binary[0]==0 && binary[1]==1)
            v3=2;
        else
            v3=0;

        if(binary[3]==1 && binary[2]==1)
            v2=3;
        else if(binary[3]==1 && binary[2]==0)
            v2=2;
    }

```

```
else if(binary[3]==0 && binary[2]==1)
    v2=1;
else
    v2=0;

if(binary[4]==1)
    v1=1;
else if(binary[4]==0)
    v1=0;

}
if(i==6)
{
    if(binary[0]==1 && binary[1]==1)
        v3=3;
    else if(binary[0]==1 && binary[1]==0)
        v3=1;
    else if(binary[0]==0 && binary[1]==1)
        v3=2;
    else
        v3=0;

    if(binary[3]==1 && binary[2]==1)
        v2=3;
    else if(binary[3]==1 && binary[2]==0)
        v2=2;
    else if(binary[3]==0 && binary[2]==1)
        v2=1;
    else
```

```

    v2=0;

    if(binary[5]==1 && binary[4]==1)
        v1=3;
    else if(binary[5]==1 && binary[4]==0)
        v1=2;
    else if(binary[5]==0 && binary[4]==1)
        v1=1;
    else
        v1=0;

}
char pass[10],password[10]="results";
int main_exit;
beep:
printf("\n\n\tEnter the password to calculate and display results :");
scanf("%s",pass);
if (strcmp(pass,password)==0)
    {printf("\n\nPassword Match!\nLOADING");
      system("cls");
      system("color 0");
      printf("Total votes polled: %d\n",total);
      printf("Candidate 'congress' votes: %d\n",v1);
      printf("Candidate 'BJP' votes: %d\n",v2);
      printf("Candidate 'TRS' votes: %d\n",v3);
      if(v1==v2 && v2==v3)
          printf("Clash occurred b/w candidate 1 and candidate 2 and candidate
3");

```

```

else if(v1>v2 && v1>v3)
    printf("And the winner is CANDIDATE 1 (SANJEETH)!!!");
else if(v2>v1 && v2>v3)
    printf("And the winner is CANDIDATE 2(SAMPATH)!!!");
else if(v3>v2 && v3>v1)
    printf("And the winner is CANDIDATE 3 (PHANIDER)!!!");

}
else

{ printf("\n\nWrong password!!\a\a\a");
login_try:
printf("\nEnter 1 to try again and 0 to exit:");
scanf("%d",&main_exit);
if (main_exit==1)
{

    system("cls");
    goto beep;
}

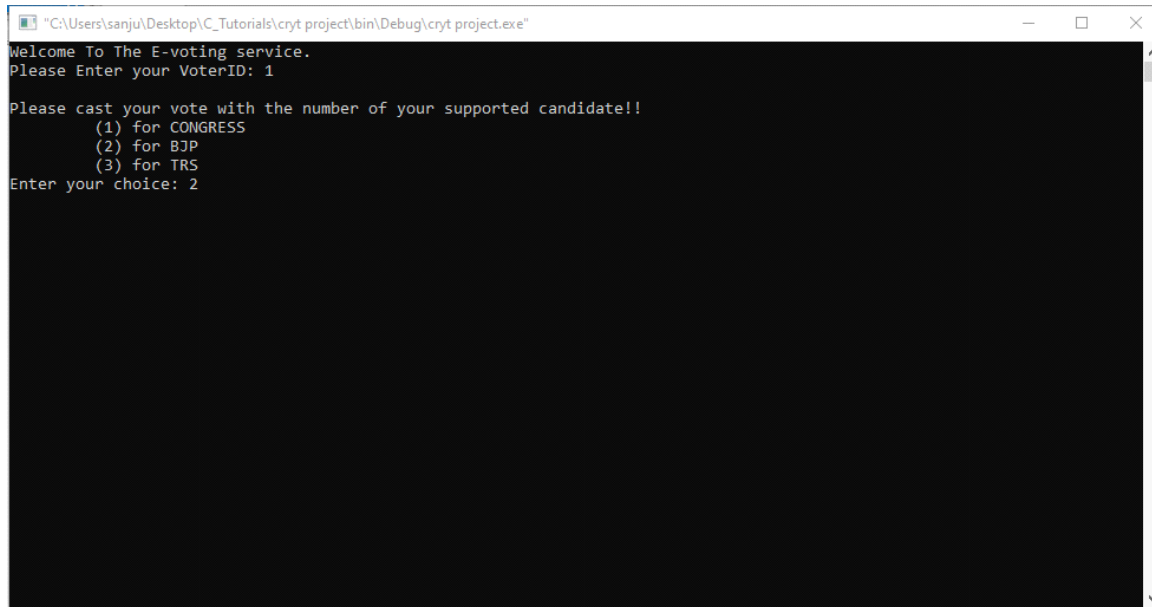
else if (main_exit==0)
{
    system("cls");
    exit(1);}
else
{printf("\nInvalid!");

    system("cls");
    goto login_try;}

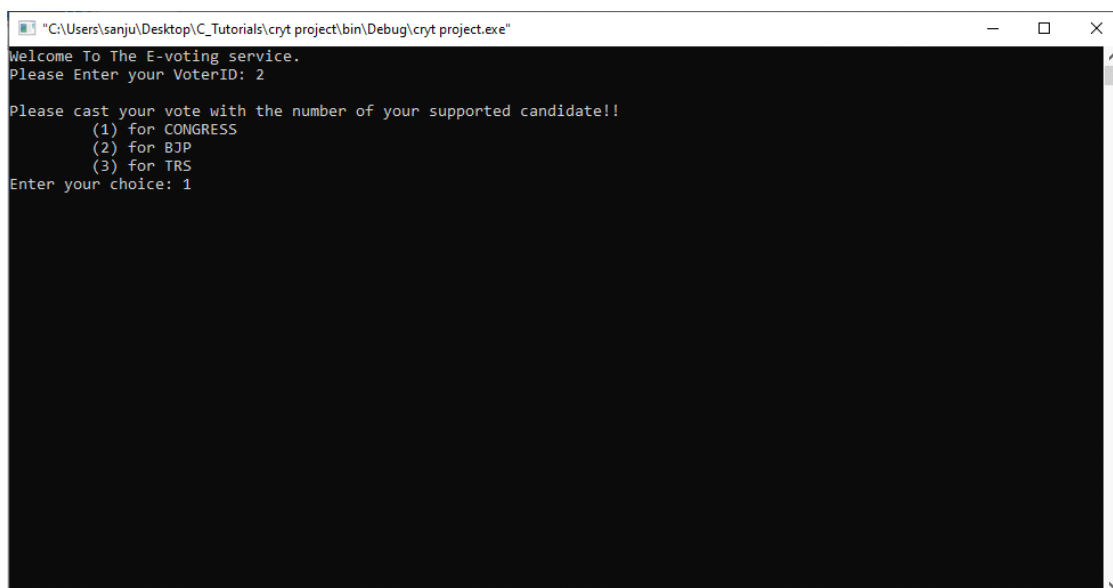
```

```
}  
return 0;  
}
```

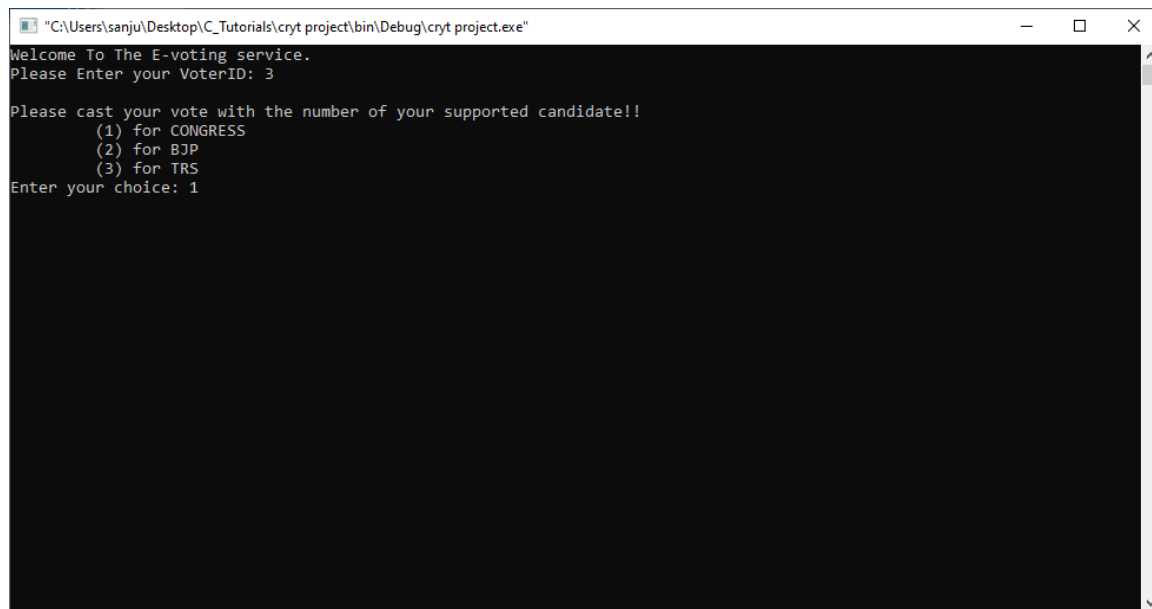
Input:-



```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"  
Welcome To The E-voting service.  
Please Enter your VoterID: 1  
  
Please cast your vote with the number of your supported candidate!!  
    (1) for CONGRESS  
    (2) for BJP  
    (3) for TRS  
Enter your choice: 2
```



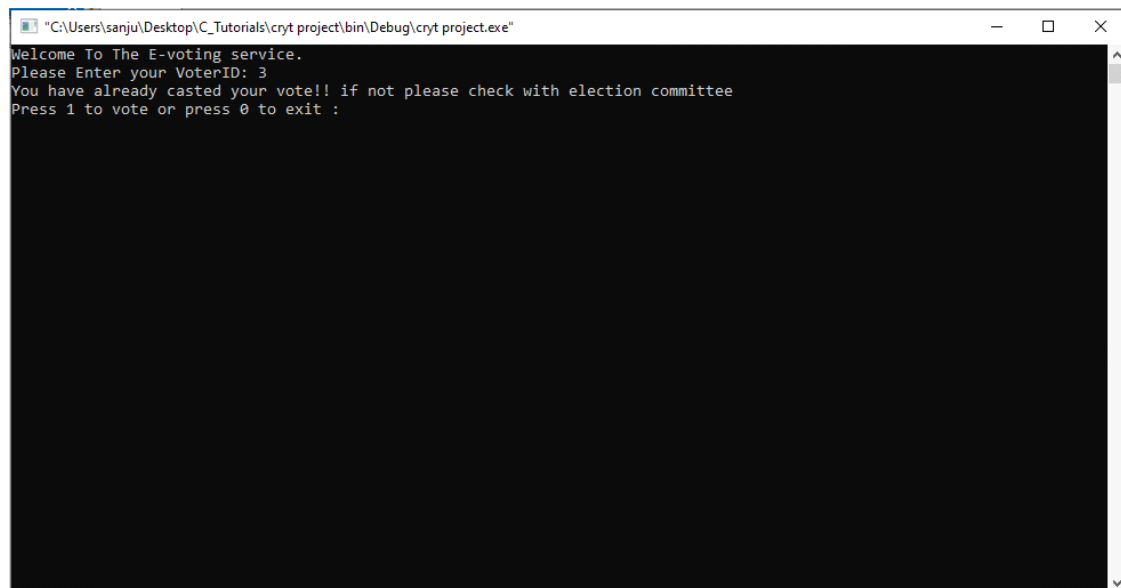
```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"  
Welcome To The E-voting service.  
Please Enter your VoterID: 2  
  
Please cast your vote with the number of your supported candidate!!  
    (1) for CONGRESS  
    (2) for BJP  
    (3) for TRS  
Enter your choice: 1
```



```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"
Welcome To The E-voting service.
Please Enter your VoterID: 3

Please cast your vote with the number of your supported candidate!!
    (1) for CONGRESS
    (2) for BJP
    (3) for TRS
Enter your choice: 1
```

If same voter again votes then vote already casted message is given.



```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"
Welcome To The E-voting service.
Please Enter your VoterID: 3
You have already casted your vote!! if not please check with election committee
Press 1 to vote or press 0 to exit :
```



```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"
Welcome To The E-voting service.
Please Enter your VoterID: 4

Please cast your vote with the number of your supported candidate!!
    (1) for CONGRESS
    (2) for BJP
    (3) for TRS
Enter your choice: 3
```

```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"
Welcome To The E-voting service.
Please Enter your VoterID: 5

Please cast your vote with the number of your supported candidate!!
    (1) for CONGRESS
    (2) for BJP
    (3) for TRS
Enter your choice: 1
```

```
"C:\Users\sanju\Desktop\C_Tutorials\crypt project\bin\Debug\crypt project.exe"
Welcome To The E-voting service.
Please Enter your VoterID: 6

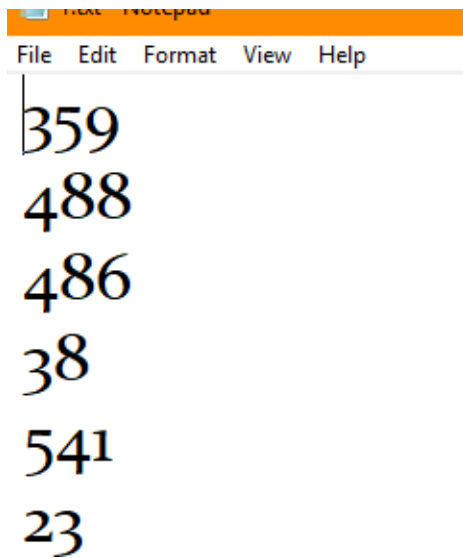
Please cast your vote with the number of your supported candidate!!
    (1) for CONGRESS
    (2) for BJP
    (3) for TRS
Enter your choice: 2
```

Output:-

```
"C:\Users\sanje\OneDrive\Desktop\crypto project\bin\Debug\crypto project.exe"
Total votes polled: 6
Candidate 'congress' votes: 3
Candidate 'BJP' votes: 2
Candidate 'TRS' votes: 1
And the winner is CANDIDATE 1 (Sanjeeth)!!!
Process returned 0 (0x0)   execution time : 15.503 s
Press any key to continue.
```

As candidate 'Sanjeeth' from congress got more no. of votes he is declared as winner and same with others. If any clash comes b/w candidates then clash occurred message will be displayed.

Encrypted file with votes:-



This file will be stored and sent for calculation and this is used to get the winner by decrypting the values.

Conclusion

This Project work can be referenced and can be extended to help the voters in casting their votes over the internet with security. This is the cheapest, secure and quickest possible way to vote when compared to the traditional voting scheme. One don't need to be physically present at the voting booth to cast their respective vote. One can cast his vote from any location in the globe. So by this method the voting percentage of public can also be increased and makes sure maximum every one participates in voting.

References

- chionyambu,s.(2018).[online]Security.hsr.ch.Availableat:_
[http://security.hsr.ch/msevote/seminar-papers/HS09 Homomorphic Tallying with Paillier.pdf](http://security.hsr.ch/msevote/seminar-papers/HS09_Homomorphic_Tallying_with_Paillier.pdf) [Accessed 22 Oct. 2018].
- gentry, c. (2018). [online] Ijcaonline.org. Available at:_
<https://www.ijcaonline.org/archives/volume141/number13/sharma-2016-ijca-909652.pdf> [Accessed 22 Oct. 2018].
- En.wikipedia.org. (2018). *Homomorphic encryption*. [online] Available at:_
[https://en.wikipedia.org/wiki/Homomorphic encryption](https://en.wikipedia.org/wiki/Homomorphic_encryption) [Accessed 22 Oct. 2018].
- En.wikipedia.org. (2018). *Paillier cryptosystem*. [online] Available at:_
[https://en.wikipedia.org/wiki/Paillier cryptosystem](https://en.wikipedia.org/wiki/Paillier_cryptosystem) [Accessed 22 Oct. 2018].
- Microsoft Research. (2018). *Homomorphic Encryption - Microsoft Research*. [online] Available at: <https://www.microsoft.com/en-us/research/project/homomorphic-encryption/> [Accessed 22 Oct. 2018].
- stallings, w. (2013). *Stallings, Cryptography and Network Security: Principles and Practice, 5th Edition / Pearson*. [online] Pearson.com. Available at:_
<https://www.pearson.com/us/higher-education/program/Stallings-Cryptography-and-Network-Security-Principles-and-Practice-5th-Edition/PGM334401.html>
- Fontaine, C. and Galand, F. (2007). A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal on Information Security*, 2007(1), p.013801.

The End