**BÀI TẬP MẬT MÃ ỨNG DỤNG**

**I.Feistel**

**B1.** **Lập trình mô phỏng các phương pháp mã hóa/giải mã 3 cấp sử dụng thuật toán Feistel đơn giản:**

#include <iostream>

#include <math.h>

using namespace std;

char F (char R,char ki){

return R + ki; }

//ham MH

string hamMH(char P0,char P1,char k){

char L[3],R[3],K[3];

string C="XX";

L[0]=P1;R[0]=P0;K[0]=k;

for(int i=1 ; i<=2 ; i++){

K[i] = K[0]<<i;

L[i] = R[i-1];

R[i] = L[i-1]^F(R[i-1],K[i]);

}

C[0]=L[2];

C[1]=R[2];

return C;}

//ham GM

string hamGM(char C0,char C1,char k0){

char L[3],R[3],K[3];

string P="XX";

L[2]=C0;R[2]=C1;K[0]=k0;

for(int i=2 ; i>=1 ; i--){

K[i] = K[0]<<i;

L[i-1] = R[i]^F(L[i],K[i]);

R[i-1] = L[i];

}

P[0]=R[0];

P[1]=L[0];

return P;}

int main(){

string P,C;

char k1,k2,k3;

cout<<"Nhap chuoi plaintext: ";getline(cin,P);

//Nhap khoa

cout<<"Nhap khoa K1: ";cin>>k1;

cout<<"Nhap khoa K2: ";cin>>k2;

cout<<"Nhap khoa K3: ";cin>>k3;

//Ma hoa

C=hamMH(P[0],P[1],k1);

C=hamMH(C[0],C[1],k2);

C=hamMH(C[0],C[1],k3);

cout<<"CHUOI MA HOA LA: "<<C<<endl;

//Giai ma

C=hamGM(C[0],C[1],k3);

C=hamGM(C[0],C[1],k2);

P=hamGM(C[0],C[1],k1);

cout<<"CHUOI GIAI MA LA: "<<P<<endl;

return 0;}

**B2: Nhập một chuỗi plaintext từ bàn phím. Lập trình mã hóa chuỗi bằng thuật toán 3 cấp đơn giản ở bài tập 1. Hiện chuỗi mã hóa ra màn hình. (Tách chuỗi thành từng khối 16 bit rồi áp dụng thuật toán). Giải mã ,hiện chuỗi giải mã ra màn hình**

#include <iostream>

#include <math.h>

using namespace std;

char F(char R,char ki){

return R+ki;}

string hamMH(char P0,char P1,char k){

char L[3],R[3],K[3];

string C="XX";

L[0]=P1;R[0]=P0;K[0]=k;

for(int i=1;i<=2;i++){

K[i]=K[0]<<i;

L[i]=R[i-1];

R[i]=L[i-1]^F(R[i-1],K[i]);

}

C[0]=L[2];

C[1]=R[2];

return C;}

string hamGM(char C0,char C1,char k0){

char L[3],R[3],K[3];

string P="XX";

L[2]=C0;R[2]=C1;K[0]=k0;

for(int i=2;i>=1;i--){

K[i]=K[0]<<i;

L[i-1]=R[i]^F(L[i],K[i]);

R[i-1]=L[i];

}

P[0]=R[0];

P[1]=L[0];

return P;}

string MHK(char P0,char P1,char k1,char k2,char k3){

string C="XX";

C=hamMH(P0,P1,k1);

C=hamMH(C[0],C[1],k2);

C=hamMH(C[0],C[1],k3);

return C;}

//GM khoi

string GMK(char C0,char C1,char k1,char k2,char k3){

string C="XX";

string P="XX";

C=hamGM(C0,C1,k3);

C=hamGM(C[0],C[1],k2);

P=hamGM(C[0],C[1],k1);

return P;}

int main(){

string P,C="",D="";

char k1,k2,k3;

cout<<"Nhap plaintext: ";getline(cin,P);

cout<<"Nhap khoa K1: ";cin>>k1;

cout<<"Nhap khoa K2: ";cin>>k3;

cout<<"Nhap khoa K3: ";cin>>k2;

//chuoi mh

for(int i=0 ; i< P.size();i+=2){

string block = MHK(P[i],P[i+1],k1,k2,k3);

C+=block; }

cout<<"Chuoi ma hoa la: "<<C<<endl;

//chuoi gm

for(int i=0 ; i< C.size();i+=2){

string block = GMK(C[i],C[i+1],k1,k2,k3);

D+=block; }

cout<<"Chuoi giai ma la: "<<D<<endl;

return 0;}

**RC4**

**BT: Lập trình nhập chuỗi plaintext *P* và chuỗi khóa *K* từ bàn phím.Mã hóa chuỗi *P* bằng thuật toán RC4. Hiện chuỗi mã hóa ra màn hình.Giải mã và hiện kết quả ra màn hình**

#include <iostream>

#include <math.h>

#include <vector>

using namespace std;

void ktaoSbox(vector<int> &S,const string &key){

int keyLen = key.length();

for(int i = 0;i <256 ; i++){

S[i]=i;}

int j=0;

for(int i = 0;i <256 ; i++){

j = (j + S[i] + key[i%keyLen])%256;

swap(S[i],S[j]);}}

string RC4(const string &input,const string &key){

vector<int> S(256);

ktaoSBox(S,key);

int i=0,j=0;

string output=input;

for(size\_t k =0; k<input.length();k++){

i = (i + 1 ) % 256;

j = (j + S[i] ) % 256;

swap(S[i],S[j]);

int rnd = S[(S[i]+S[j])%256];

output[k] ^= rnd;

}

return output;}

int main(){

string P,key,C,D;

cout<<"Nhap plaintext: ";getline(cin,P);

cout<<"Nhap khoa k: ";getline(cin,key);

//MH

C=RC4(P,key);

cout<<"Chuoi ma hoa la: "<<C<<endl;

//GM

D=RC4(C,key);

cout<<"Chuoi giai ma la: "<<D<<endl;

return 0;}

**\*Phần tử nguyên thủy**

**B1:Xét *p=7*, hãy tìm các phần tử nguyên thủy của *p***

-Ta có p=7 nên trong modul p sẽ có 6 giá trị của a thỏa mãn điều kiện a và p nguyên tố cùng nhau,các giá trị đó là:1,2,3,4,5,6

-Ta có bảng sau : (a mod 7)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a | a^1 | a^2 | a^3 | a^4 | a^5 | a^6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 4 | 1 | 2 | 4 | 1 |
| 3 | 3 | 2 | 6 | 4 | 5 | 1 |
| 4 | 4 | 2 | 1 | 4 | 2 | 1 |
| 5 | 5 | 4 | 6 | 2 | 3 | 1 |
| 6 | 6 | 1 | 6 | 1 | 6 | 1 |

=>Với a= 3,5 thỏa mã điều kiện là phần tử nguyên thủy của 7

***B2*:Lập trình nhập một số nguyên *p* từ bàn phím. Hãy hiện ra màn hình tất cả các giá trị của *a Є [1,p-1]* và các lũy thừa tương ứng của *a* (bao gồm *a, a2, a3,…ap-1*) trong modul *p*.**

#include <iostream>

#include <math.h>

using namespace std;

int dmod(int a, int m, int p) {

int kq = 1;

for (int i = 0; i < m; i++) {

kq = (kq \* a) % p; }

return kq;}

int main(){

int p;

int count,S2,S=0;

cout<<"Nhap so nguyen P: ";cin>>p;

for(int a=1 ;a<p;a++){

cout<<"a = "<<a<<": ";

for(int i =1;i<p;i++ ){

cout<<dmod(a,i,p)<<" ";

}cout<<endl;}

return 0;}

**B3: Lập trình nhập một số nguyên *p* từ bàn phím. Hãy hiện ra màn hình tất cả các phần tử nguyên thủy của *p***

#include <iostream>

#include <math.h>

using namespace std;

int dmod(int a, int m, int p) {

int kq = 1;

for (int i = 0; i < m; i++) {

kq = (kq \* a) % p; }

return kq;}

int main(){

int p;

int count,S2,S=0;

cout<<"Nhap so nguyen P: ";cin>>p;

if(p%2==1)

{ S=((p-1)/2)\*p; }

else

{ S=((p-2)/2)\*p+p/2; }

cout<<"Cac phan tu nguyen thuy cua p = "<<p<<" la: "<<endl;

for (int a = 1; a < p; a++) {

S2=0; count=0;

for(int i=1;i<p;i++){

S2=S2+dmod(a,i,p);

if(i>1 && dmod(a,i,p)==a){

count++;}}

} cout<<endl;

return 0;}

**\*Diffie-Hellman**

**B1:** **Chọn số nguyên tố p = 19, hãy thực hiện các bước trao đổi khóa bằng thuật toán Diffie-Hellman**

**Bước 1:**2 người lựa chọn 1 số nguyên tố p và một số a là phần tử nguyên thủy của p

**Bước 2:** Người A chọn ngẫu nhiên một số nguyên X(A) < p và tính Y(A) = (aX(A)) mod p,công bố Y(A), giữ bí mật X(A)

**Bước 3:** Người B chọn ngẫu nhiên một số nguyên X(B) < p và tính Y(B) = (aX(B)) mod p,công bố Y(B), giữ bí mật X(B)

**Bước 4:** A tính K = (YBX(A) ) mod p, B tính K = (Y(A)X(B)) mod p

* Áp dụng bài toán trên:
* Ta có số nguyên tố p=19 ,chọn phần tử nguyên thủy của p là a=3
* Người A chọn khóa bí mật X(A)=14 và tính Y(A)= (aX(A)) mode p= 3^14 mod 19 = 4
* Người B chọn khóa bí mật X(B)=8 và tính Y(B)= (aX(B)) mode p= 3^8 mod 19 = 6
* A tính K = (YBX(A) ) mod p = 6^14 mod 19 = 5
* B tính K = (Y(A)X(B)) mod p = 4^8 mod 19 = 5

**\*Elgamma**

**B1:** **Giả sử người *A* muốn gửi cho người *B* thông điệp *P=10*. Hai người thỏa thuận sẽ sử dụng số nguyên tố *p=19* và phần tử nguyên thủy của *p* là *a=2***

**Sinh viên hãy lựa chọn các số ngẫu nhiên *XA*, *XB* làm khóa bí mật cho *A* và *B*, rồi thực hiện quá trình mã hóa và giải mã thông điệp bằng thuật toán ElGamal**

**Bước 1:**2 người lựa chọn 1 số nguyên tố p và một số a là phần tử nguyên thủy của p

**Bước 2:** Người A chọn ngẫu nhiên một số nguyên X(A) < p và tính Y(A) = (aX(A)) mod p,công bố Y(A), giữ bí mật X(A)

**Bước 3:** Người B chọn ngẫu nhiên một số nguyên X(B) < p và tính Y(B) = (aX(B)) mod p,công bố Y(B), giữ bí mật X(B)

**Bước 4:** A tính K = (YBX(A) ) mod p

A mã hóa P theo công thức : *C = K.P mod p và C được gửi cho B*

**Bước 5:** Ở bên nhận, người B có thể dễ dàng tính được theo công thức: P = C. K-1 mod p (\*) với *K-1* = Y(A)(p-1-X(B)) mod p sau đó thay lại vào (\*)

**->Áp dụng bài tập trên: + T**a có p=19 và a=2

-Người A chọn khóa bí mật X(A)=10 với Y(A)= aX(A) mod p = 210 mod 19 = 17.Công bố YA ,bí mật X(A)

-Người B chọn khóa bí mật X(B)=14 với Y(B)= aX(B) mod p = 214 mod 19 = 6.Công bố Y(B) ,bí mật X(B)

- A tính khóa K = Y(B)X(A) mod p = (6^10) mod 19 = 6

- A mã hóa P: C=K.P mod p = 6.10 mod 19 =3 và bản mã C được gửi cho B

- B giải mã : P = C.K-1 mod p (\*)

+ K-1 = Y(A)(p-1-X(B)) mod p = 17(19-1-14) mod 19=16

+Thay K-1 vào (\*) : P=3.16 mod 19 =10

**\*ECC**

**B1: Lập trình tính số điểm có trên đường cong *Ep(a,b)*, với *a, b, p* nhập từ bàn phím. Liệt kê các điểm của đường cong đó ra màn hình**

#include<iostream>

#include<math.h>

using namespace std;

int main(){

int a,b,p,x,y,y2,yy,d=0;

cout<<"Nhap modul P: ";cin>>p;

cout<<"Nhap so a: ";cin>>a;

cout<<"Nhap so b: ";cin>>b;

for(x=0;x<p;x++){

y2=(x\*x\*x + a\*x + b)%p;

for(y=0;y<p;y++){

yy=(y\*y)%p;

if(y2==yy){

cout<<d<<":("<<x <<","<<y <<")"<<endl;

d++;}}}

cout<<"Tong so diem tren duong cong = "<<d;

return 0;}

**B2: Cho *P = (5,4)* và *Q = (19,5)* trong *E23(1,1)*. - Tính *R = P + Q* .Tính *P = R – Q* rồi so sánh với *P* ban đầu**

**\*R=P+Q**

Bước 1: Với P!=Q : Tính *λ=((yQ − yP)/(xQ − xP))mod p*

Bước 2: x(R) = (*λ2 - xP - xQ*) mod p ; y(R) = (*λ(xP - xR) - yP* ) mod p => R=P+Q=( *xR ,yR* )

**\*P=R-Q**

Bước 1: *P = R-Q = R+(-Q)* =( *xR ,yR) + (xQ,23- yQ) với 23 thuộc vành 23 của* ***E23***

Bước 2: Tính *λ=((yR − yQ)/(xR − xQ))mod p*

Bước 2: x(R) = (*λ2 – xR - xQ*) mod p ; y(R) = (*λ(xR – xP) – yR* ) mod p => P=R-Q=( *xR ,yR* )

**->ÁP dụng bài tập trên:**

**\*R=P+Q**

- *λ=((yQ − yP)/(xQ − xP))mod p = ((5-4)/(19-5)) mod 23 = 1/14 mod 23 = 5*

*-* x(R) = (*λ2 - xP - xQ*) mod p = (25 – 5-19) mod 23 =1

- y(R) = (*λ(xP - xR) - yP* ) mod p = (5(5-1)-4) mod 23 = 16 mod 23 = 16

-> R=P+Q=(1,16)

**\*P=R-Q**

- Ta có: P=R-Q = R + (-Q) = ( *xR ,yR) + (xQ , 23- yQ) =(1,16) + (19,23-5) =(1,16) + (19,18)*

- *λ=((yR − yQ)/(xR − xQ))mod p = ((16-18)/(1-19)) mod 23 = 2/18 mod 23 = 1/9 mod 23 = 18*

*-* x(P) = (*λ2 – xR - xQ*) mod p = (182 – 1-19) mod 23 =5

- y(P) = (*λ(xR – xP) – yR* ) mod p = (18(1-5)-16) mod 23 = -88 mod 23 = 4

-> P=R-Q=(5,4)

**B3: Cho *P = (5,4)* trong *E23(1,1)*, tính *R = 2P***

**\*R=*P+Q = 2P***

Bước 1: Với P!=Q : Tính *λ= mod p*

Bước 2:Tính  *xR = ( λ2 - 2xP ) mod p* và *yR = (λ(xP - xR) - yP ) mod p*

* **Áp dụng bài tập trên:**

-Tính *λ= mod p = mod 23 = mod 23 = 19.(2-1 mod 23) = (19.12) mod 23=21*

- x(R) = (*λ2 – 2xP*) mod p = (212 – 2.5) mod 23 = 17

- y(R) = (*λ(xP – xR) – yP* ) mod p = (21(5-17)-4) mod 23 = -256 mod 23 = 20

-> P=2P=(17,20)

**B3.2:Giả sử plaintext biểu diễn dưới dạng điểm *P = (3,10)* trong *E23(1,1)*. Hãy mã hóa nó bằng công thức: *C=P+K* với khóa *K* là điểm *(9,7)* trong *E23(1,1)* .Giải mã và kiểm tra kết quả**

- Mã hóa: *C = P+K = (3,10) + (9,7)*

+Tính *λ= = mod 23 = mod 23 = -12 mod 23 = 11*

*+xC = (112 – 3 – 9) mod 23 = 109 mod 23 = 17*

+ *yC = (11(3 -17) -10) mod 23 = -164 mod 23 = 20*

*Vậy C = P+K = (17, 20)*

- Giải mã: *P = C - K = C + (-K) = (17, 20) + (9,23-7) = (17, 20) + (9,16)*

+ Tính *λ = mod 23 = mod 23 = mod 23 = 12*

+ *xP = (122-17-9) mod 23 = 118 mod 23 = 3*

+ *yP = (12(17-3) -20) mod 23 = 148 mod 23 = 10*

Vậy *P = C-K = (3,10)*, giống với plaintext

**B4: *P*(*xP*, *yP*) và *Q*(*xQ*, *yQ*) là 2 điểm khác nhau thuộc *Ep(a,b)*. Lập trình nhập tọa độ 2 điểm rồi tính tổng 2 điểm đó, với *p* nhập từ bàn phím**

***B5*: *P*(*xP*, *yP*) là 1 điểm thuộc *Ep(a,b)*. Lập trình nhập tọa độ điểm *P* rồi tính tích *n.P*, với *a* và *p* nhập từ bàn phím**

**B6: *P*(*xP*, *yP*) là 1 điểm thuộc *Ep(a,b)*. Lập trình nhập tọa độ điểm *P* rồi tính bậc *n* của điểm *P* đó, với *a* và *p* nhập từ bàn phím.**

**B7: Giả sử người *A* muốn gửi cho người *B* plaintext *P*, hai người thỏa thuận sử dụng đường cong *E23(1, 1)* *y2 = x3 + x +1* , *p = 23*, điểm *G = (3, 10)* tương ứng bậc của *G* là *n=28*.Plaintext *P* được chuyển đổi thành điểm *(9,7)* trên *E23(1, 1)***

**Sinh viên hãy lựa chọn các số ngẫu nhiên *nA*, *nB* làm khóa bí mật cho *A* và *B*, rồi thực hiện quá trình mã hóa và giải mã thông điệp bằng thuật toán ECC nói trên**

-Bước 1: *A* chọn khóa riêng *nA  (vói nA < n) ,tính PA = nA.G ,công bố PA*

-Bước 2: *B* chọn khóa riêng *nB  (vói nB < n) ,tính PB = nB.G ,công bố PB*

-Bước 3: Quá trình mã hóa:

+*A* tính *K = nA ×PB*

*+A mã hóa P theo công thức: C = P+K -> Bản mã C gửi cho B*

-Bước 4: Quá trình giải mã:

+*B* giải mã thông điệp bằng công thức: *P = C-K = C+(-K) (\*)*  với *K = nB ×PA*

+ (*-K) = (xK,n-yK) -> Thay vào (\*)*

**B8: Giả sử người *A* muốn gửi cho người *B* plaintext *P* bất kỳ, hai người thỏa thuận sử dụng đường cong *E23(1, 1)* *y2 = x3 + x +1* , *p = 23*, điểm *G = (3, 10)***

***A* chọn khóa riêng *nA = 9*, tính khóa công khai *PA = 9.G =9(3,10) = (0,1)***

***B* chọn khóa riêng *nB = 18*, tính khóa công khai *PB = 18.G* *=18(3,10) = (6,19)***

**Hãy thực hiện quá trình mã hóa và giải mã chuỗi nói trên**

**\*Cần chọn *nA, nB* sao cho bậc của *PA<nB* và bậc của *PB<nA ,* vì khi đó mới có thế tính *K* theo công thức *K = nA ×PB* *= nB ×PA***

**\*Quá trình mã hóa:**

-Bước 1: *A* tính khóa *K = nA ×PB*

-Bước 2: Chuỗi *“XXXXX”* gồm *5* kí tự ứng với *5* điểm trên đường cong, áp dụng công thức *C = P+K*:

VDU: *P1=‘H’ ≈ (5, 4) → C1= (5,4) + (12,19) = (12,4)*

**\*Quá trình giải mã :**

-Bước 1: *B* giải mã thông điệp bằng công thức: *P = C-K = C+(-K) (\*)*

*với K = nB ×PA → -K = (xK,n-yK)*

-Bước 2: Thay -K vào (\*) ,VDU: *C1= (12,4) → P1= (12,4) + (12,4) = (5, 4) ≈ ‘H’*

**B9: Giả sử người *A* muốn gửi cho người *B* plaintext *P* *= 8*, hai người thỏa thuận sử dụng đường cong *E23(1, 1)* *y2 = x3 + x +1* , *p = 23*, điểm *G = (3, 10)***

***A* chọn khóa riêng *nA = 8*, tính khóa công khai *PA = 8.G =8(3,10) = (13,16)***

***B* chọn khóa riêng *nB = 5*, tính khóa công khai *PB = 5.G* *=5(3,10) = (9,16)***

**Hãy thực hiện quá trình mã hóa và giải mã plaintext nói trên**

***\*Quá trình mã hóa:*** *P = 8* ứng với điểm *(a,b)(xem bảng)*

*A* tính khóa *K = nA ×PB*

*A* mã hóa *P* theo công thức: *C = P+K*  -> Bản mã *C* và *PA* được gửi cho *B*

***\*Quá trình giải mã :***

*B* sẽ giải mã thông điệp nhận được bằng công thức: *P = C-K = C+(-K) (\*)*

với *K = nB ×PA=5(13,16) = (5,4)* *→ -K = (xK,n-yK)* -> Thay -K vào *(\*)*

**Chương 2:**

1. **Samsung Pay có thực sự an toàn?**

- Một nhà nghiên cứu cho biết: Có thể lấy trộm được Token của Samsung Pay khi thanh toán bằng các máy POS sử dụng công nghệ MST đời cũ, sau đó tái sử dụng Token này để mua hang.Quá trình đó được thực hiện như sau:

- Anh ta chế tạo một thiết bị đọc từ tính và đeo nó vào tay, nhờ vậy có thể đọc được token + cryptogram khi cầm điện thoại của người khác (có Samsung Pay) hoặc anh ta có thể ngụy trang thiết bị đó giống một máy cà thẻ để lừa người khác đưa điện thoại vào

- Sau khi lấy được token + cryptogram, anh ta nạp nó vào 1 thiết bị phát từ tính tự chế, rồi đưa nó tới gần máy POS ở cửa hàng

- Máy phát tự chế sẽ chuyển đổi token + cryptogram thành các tín hiệu từ tương ứng và gửi tới máy POS

- Kết quả là máy POS đã chấp nhận thanh toán

**Câu hỏi: Tại sao phương pháp trên lại có thể thành công?**

**+ Liệu nó chỉ thành công trong thử nghiệm trên, hay có thể thành công trên thực tế?Bài học rút ra là gì?**

**\*Tại sao lại thành công?**

- Do các lỗ hổng trong cách mà công nghệ MST (Magnetic Secure Transmission) hoạt động, cụ thể như sau:

+ **MST giả lập thẻ từ vật lý:**  
MST cho phép Samsung Pay giả lập tín hiệu từ của thẻ từ vật lý. Khi điện thoại phát ra tín hiệu này, máy POS sẽ hiểu như đang nhận dữ liệu từ một thẻ từ thông thường. Kẻ tấn công có thể tạo ra thiết bị giả lập tương tự, phát lại tín hiệu đã đánh cắp để lừa máy POS.

 **Token + Cryptogram không bị ràng buộc với thiết bị gốc:** không có biện pháp xác thực bổ sung (như xác minh thiết bị gốc), chúng có thể được phát lại trên máy POS khác. Điều này cho phép kẻ tấn công tái sử dụng thông tin đã đánh cắp để thực hiện thanh toán gian lận.

 **Máy POS đời cũ không đủ bảo mật:**Các máy POS cũ chỉ xác nhận tín hiệu từ mà không có khả năng phân biệt tín hiệu đó đến từ thiết bị hợp lệ hay từ một thiết bị giả mạo. Điều này giúp kẻ tấn công dễ dàng qua mặt hệ thống.

**Phương pháp này có thể thành công trên thực tế không?**

-Có thể thành công trong thực tế, nhưng chỉ trong những trường hợp sau:

+ Máy POS sử dụng công nghệ MST đời cũ: POS hiện đại hỗ trợ **NFC** và các giao thức bảo mật cao hơn

+ Người dùng thiếu cảnh giác: lợi dụng sự thiếu hiểu biết và mất cảnh giác sử dụng các thiết bị giả dạng như máy POS hoặc thiết bị thanh toán.

**Bài học rút ra là gì?**

+ Cần nâng cấp hạ tầng thanh toán: cần cập nhật các máy POS để hỗ trợ **NFC** và các giao thức bảo mật tiên tiến như **EMV**

+ Người dùng cần cảnh giác khi thanh toán:

+ Các nhà cung cấp dịch vụ thanh toán cần tăng cường bảo mật: Samsung và các công ty cung cấp ví điện tử cần đảm bảo rằng token và cryptogram được ràng buộc với thiết bị gốc ngăn chặn việc tái sử dụng

**B2: Sinh viên thử phán đoán giải pháp lưu trữ thẻ tín dụng của các trang bán hàng online như Shopee, Lazada?**

-Shopee ,Lazada sử dụng các giải pháp lưu trữ và xử lý thẻ tín dụng an toàn dựa trên các tiêu chuẩn bảo mật quốc tế như PCI DSS (Payment Card Industry Data Security Standard).! Số phán đoán giải pháp:

+ **Tokenization (Mã hóa bằng token):**Thay vì lưu trực tiếp số thẻ tín dụng = 1 mã token ngẫu nhiên. Token là **mã hóa một chiều**, nghĩa là không thể khôi phục lại số thẻ gốc từ token.

+ Sử dụng cổng thanh toán của bên thứ ba (Payment Gateway):Họ thường liên kết với bên thứ 3 như **Adyen**,**CyberSource (Visa)**,**Paypal** các cổng trêntuân thủ nghiêm ngặt các tiêu chuẩn bảo mật PCI DSS và có trách nhiệm xử lý, lưu trữ dữ liệu thẻ tín dụng.

+ **Kết hợp các biện pháp bảo mật bổ sung:**Mã hóa dữ liệu (Encryption),Xác thực 2 yếu tố (2FA),Phát hiện gian lận (Fraud Detection)

**B3:** **Tại sao không cần mã hóa trường *Salt*, liệu có sợ bị lộ *Salt* không?**

- Salt là một đoạn dữ liệu ngẫu nhiên được thêm vào trước khi thực hiện băm (hash) mật khẩu nên dù có bị lộ cũng không gây rủi ro bảo mật nghiêm trọng. Salt không cần phải được giữ bí mật vì bản thân nó không giúp hacker đoán được mật khẩu, hacker vẫn không thể dùng Rainbow Table để tấn công. Việc mã hóa Salt sẽ làm phức tạp hệ thống mà không tăng thêm bảo mật.

**B4: Có thể thay đổi được khóa K *(pepper)* hay không?**

**B5: Tại sao lại cần tới các trung gian thanh toán?**

-Đơn giản hóa quy trình cho doanh nghiệp

- Đa dạng hóa hình thức thanh toán

- Giải quyết **vấn đề tin cậy** giữa người mua và người bán, đồng thời **tăng cường bảo mật** và **tối ưu hóa quy trình giao dịch**.

**B6:*****Ưu nhược điểm của Blockchain so với các mô hình quản trị dữ liệu truyền thống?***

**-Ưu điểm:**

+ Tự do: Không phụ thuộc vào bất cứ trung gian hay nhà quản lý nào

+ Minh bạch: Mọi người đều có thể giám sát dữ liệu

+ Đáng tin cậy: Không thể làm giả dữ liệu

**-Nhược điểm:**

+ Tốc độ chậm (do phải chi phí nhiều thời gian cho việc đồng thuận của đám đông)

+ Dư thừa dữ liệu gây lãng phí không gian lưu trữ (có quá nhiều bản sao giống nhau của dữ liệu trên hệ thống)

+ Chi phí lớn (sử dụng nhiều phần cứng máy tính đắt tiền và tiêu tốn nhiều năng lượng điện! )

+ Chưa tôn trọng sự riêng tư và Không thể sửa đổi dữ liệu

**B7:*****Blockchain có thực sự là một công nghệ tiên tiến, hay chỉ là một “trào lưu” công nghệ?Để trả lời câu hỏi trên, ta hãy suy nghĩ nghiêm túc về các vấn đề sau:***

* ***Công nghệ này có giúp cải tiến năng suất lao động không?***
* Có. Blockchain giúp loại bỏ **bên trung gian** trong nhiều lĩnh vực như tài chính, chuỗi cung ứng, và bảo hiểm -> quy trình giao dịch trở nên **nhanh hơn** và **chi phí thấp hơn**
* - Giảm thiểu sai sót và gian lận trong quy trình kiểm toán, kế toán.
* VD: **De Beers** sử dụng Blockchain để theo dõi kim cương từ mỏ khai thác đến tay người tiêu dùng, đảm bảo rằng kim cương không có nguồn gốc xung đột.
* ***Công nghệ này có giúp nâng cao chất lượng cuộc sống không?***

- Blockchain mang lại lợi ích cho nhiều lĩnh vực quan trọng:

+ Giúp **cá nhân hóa tài chính** thông qua các nền tảng **DeFi**, nơi người dùng có thể vay, cho vay, và giao dịch mà không cần ngân hàng trung gian.

+ Blockchain cho phép tạo ra **danh tính số** an toàn và phi tập trung, giúp người dùng **kiểm soát dữ liệu cá nhân** của minh

+ Blockchain giúp **minh bạch hóa chuỗi cung ứng**, từ đó đảm bảo **an toàn thực phẩm** và **đạo đức kinh doanh**. **UNICEF** sử dụng Blockchain để theo dõi việc **phân phối quỹ từ thiện**, đảm bảo rằng tiền đến đúng nơi cần thiết.

***B8: Trong Mempool thường chứa nhiều giao dịch, giao dịch nào sẽ được xử lí trước?***

- Xử lý theo thứ tự ưu tiên **dựa vào phí giao dịch**. Giao dịch có phí cao hơn sẽ được xử lý trước, vì các thợ đào luôn tìm cách tối ưu hóa lợi nhuận của mình.

***B9: Tại sao lại phải tốn nhiều công sức như vậy để tính Nonce?***

- Việc tính Nonce tiêu tốn nhiều năng lượng điện và đòi hỏi nhiều phần cứng đắt tiền.Nhưng việc tính toán phức tạp đó cũng giúp tăng độ an toàn của hệ thống.

-Kẻ tấn công nếu muốn sửa khối dữ liệu thứ *k* sẽ phải sửa lại mã hash và Nonce của tất cả các khối từ khối *k* tới khối hiện tại, đó là một khối lượng tính toán khổng lồ!

***B10:******Nếu một ngày kỹ thuật mã hóa của Bitcoin trở nên lạc hậu, điều gì sẽ diễn ra?***