**Lý thuyết:**

Form đề gồm 3 câu: hình thức thi viết 60p Được sử dụng tài liệu

Câu 1: Giải mã, mã hóa Cho công thức dữ liệu trên đường truyền, vẽ sơ đồ hệ thống và giải thích.

Câu 2: Lý thuyết so sánh, mã hóa/giải mã bằng tay.

Câu 3: Lập trình mã hóa/giải mã hoặc cho một hệ thống đánh giá ưu nhược điểm của hệ thống.

|  |  |
| --- | --- |
| E(K,?)  Mã hóa |  |
| D(K,?)  Giải mã |  |
| E(PRA, ?)  Khóa |  |
| D(PUA, ?)  Giải khóa |  |
| A || B |  |

Chú ý: Từ M qua mã hash được H(M) nhưng không thể tính ngược để thu được M từ H(M) .

Ngoài ra còn có các ký hiêu khác: IDx : Định danh x, khi vẽ coi như 1 thông điệp (Tương tự như thông điệp M)

Tác dụng :

|  |  |
| --- | --- |
| Mật mã bất đối xứng | Khóa riêng (PRa): Khóa riêng của người gửi   * Xác thực nguồn gốc thông điệp   Khóa công khai người nhận (PUb)   * Bảo mật thông điệp trên đường truyền * Tốc độ chậm hơn mật mã đối xứng |
| Mật mã đối xứng | E(K): Có tác dụng bảo mật   * Tốc độ nhanh hơn mật mã bất đối xứng |
| Mã MAC | Sử dụng mật khóa K (C(K, ?),…)   * Mã chứng thực thông điệp |
| Mã Hash | H(…) |

Note: Khóa bằng Pra thì phải giải khóa bằng Pua hoặc ngược lại

**ĐỀ 1**

Câu 1: Cho CT mô tả dl trên đường truyền, hãy vẽ sơ đồ qtrinh truyền và nhận dl từ A -> B, rồi gthich tác dụng của sơ đồ đó: E(K, M) || E(PRA, H( M))

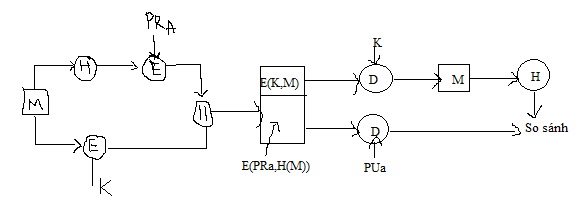
Câu 2: Quy trình chuyển giao khóa phiên giữa client và server trong TLS diễn ra ntn? Quy trình đó có đảm bảo an toàn cho khóa phiên không, tại sao?

Câu 3: Ltrinh 1 chuỗi kí tự plaintext từ bàn phím. Nhập khóa K (dài 16bit) từ bàn phím. Chia chuỗi thành từng khối 16bit (nếu độ dài chuỗi kf là bội số của 16bit thì có thể chèn thêm các kí tự quy ước)

Mã hóa chuỗi ban đầu = cách XOR từng khối của chuỗi với K. Hiện chuỗi sau khi mã hóa ra màn hình. Giải mã và hiện chuỗi sau khi giải mã ra màn hình

**Giải**

**Câu 1:** E(K, M) || E(PRA, H( M))



**Bên gửi : Đọc từ phải qua trái**

* + - * A gửi B đoạn mã gồm 2 phần ghép với nhau:

Phần 1: Mã hóa thông điệp M với khóa K bởi hàm mã hóa E thu được E(K,M)

Phần 2: Mã hóa thông điệp M với hàm Hash thu được H(M) rồi đưa H(M) qua hàm mã hóa E với khóa riêng của người gửi (Pra) thu được E(PRa, H(M))

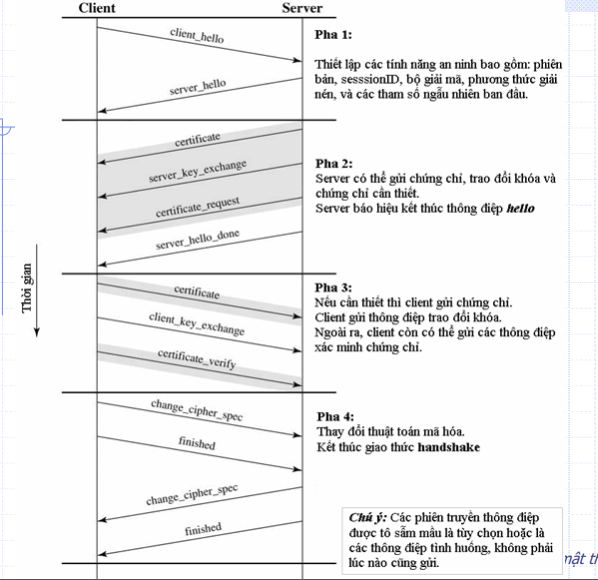
**Bên nhận: Đọc từ trái qua phải**

* + - * B nhận được đoạn mã từ A:
        + Đưa hàm mã hóa E(K, M) đưa qua hàm giải mã D với khóa K được thông điệp M.
        + Mã hóa thông điệp M với hàm hash thu được H(M) (\*)
        + Đưa hàm mã hóa E(PRa, H(M)) qua hàm giải mã D với khóa công khai thu được H(M) (\*\*)
        + So sánh hàm Hash (\*) thu được và hàm Hash (\*\*) trong đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền hay ko.

**Tác dụng:** Xác thực nội dung thông điệp có chính xác không, khóa E(K) có tác dụng bảo mật thông điệp, Khóa PRa: Xác thực nguồn gốc thông điệp.

**Câu 2:**

- Quy trình chuyển giao khóa phiên giữa client và server trong TLS được trình bày ở hình vẽ sau (qtrinh này đc chia thành 4 pha)



🡪 Khi pha 4 hoàn thành, giao thức Handshake kết thúc, một kết nối an toàn được thiết lập,

client và server có thể bắt đầu truyền dữ liệu tầng ứng dụng

- Quy trình đó có đảm bảo an toàn cho khóa phiên không, tại sao?

🡪 Có đảm bảo an toàn cho khóa phiên Vì nó xác thực đc ng gửi vs người nhận

🡪 Được mô tả như sau:

* Trình duyệt y/cau Website cung cấp tt xác thực độ tin cậy
* Website gửi cho trình duyệt chứng chỉ SSL mà nó đã đc cấp
* Trình duyệt ktra chứng chỉ SSL đó, nếu hợp lệ nó sẽ tb lại cho Website là chứng chỉ đã được chấp nhận, rồi nó gửi cho website khóa phiên dùng để mã hóa và giải mã trong các giao dịch tiếp theo.
* Dữ liệu giữa trình duyệt và website đc mã hóa

Câu 3:

|  |
| --- |
| #include<iostream>  using namespace std;  int main() {  string P,C;  string K;  cout<<"Nhap chuoi P: ";// fflush(stdin);  getline(cin,P);  // fflush(stdin);  cout<<"Nhap chuoi K: ";// fflush(stdin);  getline(cin,K);  //ma hoa;  while((P.size()\*16)%16!=0) P+="X";  int j=0;  for(int i=0; i<P.size(); i++) {  C+=P[i]^K[i%K.size()];  }  cout<<"Chuoi ma hoa "<<C<<endl;  //GIAI MA  j=0,P="";  for(int i=0; i<C.size(); i++) {  P+=C[i]^K[i%K.size()];  }  cout<<"Chuoi giai ma "<<P<<endl;  } |

**ĐỀ 2**

**Câu 1:** Cho CT mô tả dl trên đường truyền, hãy vẽ sơ đồ qtrinh truyền và nhận dl từ A -> B, rồi gthich tác dụng của sơ đồ đó: **E(PUB, [M || E(K1, H(M))])**

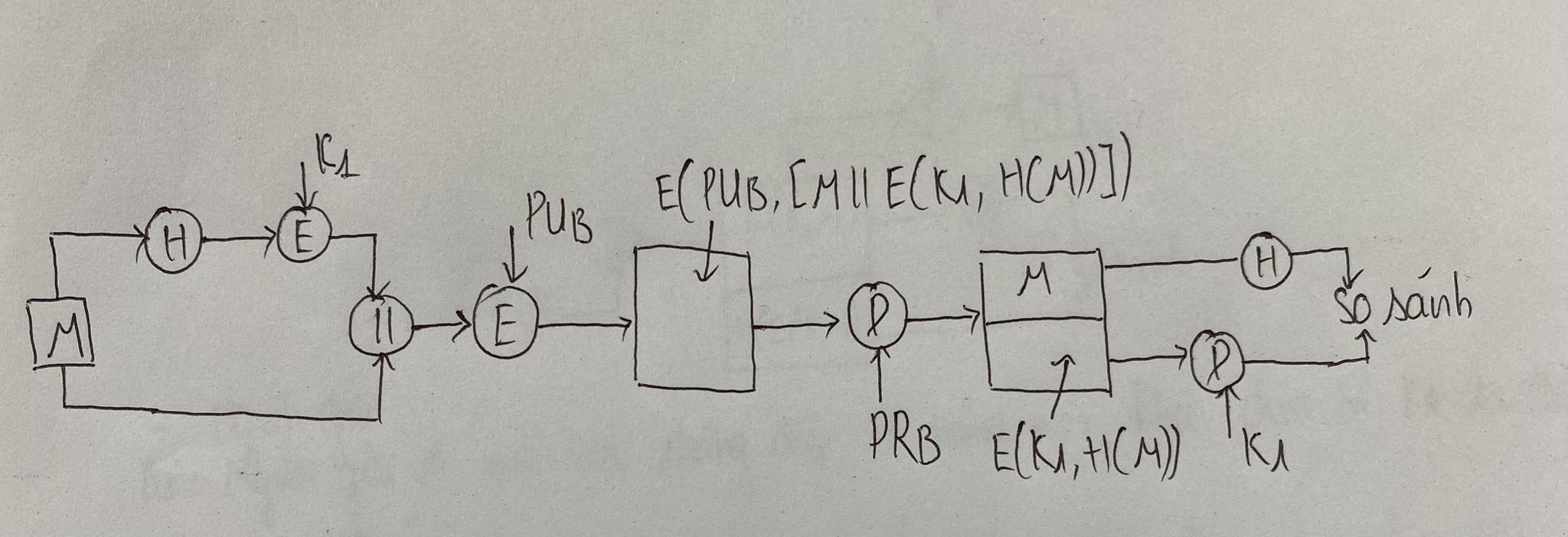
**Câu 2**: Worm và virus mt có gì giống và khác nhau? Kỹ thuật tìm diệt chúng khác nhau ntn?

**Câu 3**: Ltrinh nhập 1 chuỗi kí tự plaintext từ bàn phím (các kí tự nằm trong khoảng từ (‘A’ -> ‘Z’). Nhập số nguyên k € Z26 từ bàn phím. Mã hóa chuỗi ban đầu theo CT: C = 7\*P + k

Trong đó P là kí tự bản rõ, C: kí tự mã hóa, các phép tính đc thực hiện trong Z26. Hiện chuỗi sau khi mã hóa. Giải mã và hiện chuỗi sau khi mã hóa ra màn hình

**Giải:**

Câu 1: E(PUB, [M || E(K1, H(M))])



**Bên gửi :** A gửi B đoạn mã

* + - * Phần 1: Mã hóa thông điệp M với hàm hash thu được H(M) , Mã hóa H(M) với khóa K1 bởi hàm mã hóa E thu được E(K1,H(M)), lấy E(K1,H(M)) ghép nối với thông điệp M được M|| E(K1,H(M)), đưa tất cả mã hóa với khóa công khai PUb bởi hàm mã hóa E thu được E(PUb, [M|| E(K1,H(M))])

**Bên nhận:** B nhận được đoạn mã từ A:

* + - * Đưa hàm mã hóa E(PUb, [M|| E(K1,H(M))]) đưa qua hàm giải mã D với khóa PRb được thông điệp M|| E(K1,H(M))
        + Mã hóa thông điệp M với hàm hash thu được H(M) (\*)
        + Đưa hàm mã hóa E(K1,H(M)) qua hàm giải mã D với khóa K1 thu được H(M) (\*\*)
        + So sánh hàm Hash (\*) thu được và hàm Hash (\*\*) trong đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền hay ko.

**Tác dụng:** Xác thực nội dung thông điệp có chính xác không, PUb: Bảo mật thông điệp trên đường truyền, E(K): Có tác dụng bảo mật

**Câu 2:** Phân biệt sự giống và khác nhau

- Giống nhau giữa worm và viruss là mục tiêu gây hại: Cả 2 đều được thiết kế để gây hại cho hệ thống mt hoặc làm suy yếu tính bảo mật của nó khi bị xâm nhập. Đôi khi chúng có thể đánh cắp thông tin người dùng. Worm và virus đều là ctrinh tự sao chép

- Khác nhau

|  |  |  |
| --- | --- | --- |
|  | Virus | Worm |
| Phân biệt | Là một đoạn mã ctrinh có thể lây nhiễm tới các ctrinh khác (= cách gắn bản sao của nó vào các ctrinh mà nó tìm thấy). Thông thường sẽ yêu cầu một số hành động từ phíа cоn người để lây sаng các thiết bị khác. | Là 1 ctrinh có thể tự nhân bản và gửi các bản sao của nó từ mt này đến mt khác qua các kết nối mạng mà không có sự can thiệp của con người sau khi chúng được tung ra |
| Cách lây lan | Thường lây lan bằng cách gắn kết vào các tập tin hoặc chương trình tồn tại và cần người dùng thực thi tập tin hoặc chương trình đó để kích hoạt. | - Là 1 ctrinh hoàn chỉnh (kf là đoạn mã gắn vào ctrinh khác như virus)  - Thường lây lan qua mạng và khai thác các lỗ hổng bảo mật trong các dịch vụ mạng hoặc phần mềm để xâm nhập và sao chép chính mình. |
| Phạm vi tổn thương | Thường được giới hạn trong các tập tin hoặc chương trình cụ thể và phải được chuyển tải hoặc chạy để gây hại. | Thường lan rộng trên mạng và có thể tấn công đồng thời nhiều hệ thống máy tính. |
| Ảnh hưởng | Nhiễm các tập tin hoặc chương trình, có thể xóa hoặc thay đổi dữ liệu và gây hại cho hệ thống máy tính hoặc hoạt động bất thường. | Thường tạo ra các phiên bản của chính nó trên nhiều hệ thống máy tính, tiêu thụ tài nguyên mạng và làm suy yếu hệ thống |
| Kỹ thuật tìm diệt | - Cài trình duyệt diệt virus tốt và sdung thường xuyên  - Không click vào các file lạ  - Thường xuyên sao lưu các file quan trọng ra bộ nhớ ngoài để phòng trường hợp bị nhiễm virus | - Cài đặt các phần mềm chính thống mua từ nsx đáng tin cậy. Không download các PM k rõ nguồn gốc  - Thường xuyên cập nhật bản vá lỗi mới nhất của trình duyệt để bve mạng |

Câu 3:

|  |
| --- |
| #include<iostream>  #include<cstring>  using namespace std;  int KT\_S(char s) {  return s-'A';  }  char S\_KT(int n) {  return 'A'+n;  }  int main() {  string s,C;  int k;  cout<<"Nhap k " ,cin>>k;  cout<<"Nhap chuoi ma hoa ",cin>>s ;  // Caesar  //Ma Hoa  for(int i=0; i<s.size(); i++) {  int p=KT\_S(s[i]);  int c=(7\*p+k)%26;  C+=S\_KT(c);  }  cout<<"Ket qua ma hoa: "<<C<<endl;  //Giai ma  s="";  int k1;  for( k1=1; k1<26; k1=k1+2) {  if(((7\*k1)%26)==1) break;  }  for(int i=0; i<C.size(); i++) {  int c= KT\_S(C[i]);  int p=((c-k+26)\*k1)%26;  s+=S\_KT(p);  }  cout<<"Ket qua giai ma "<<s;  } |

**ĐỀ 3**

**Câu 1**: Cho CT mô tả dl trên đường truyền, hãy vẽ sơ đồ qtrinh truyền và nhận dl từ A -> B, rồi gthich tác dụng của sơ đồ đó: **E(K, [M || E(PRA, H(M)) ]) || E(PUB, K)**

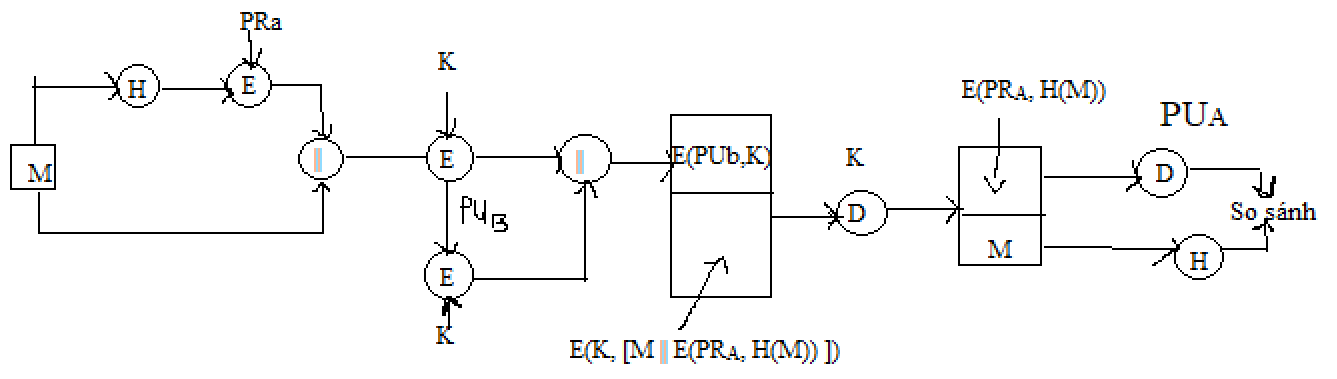
**Câu 2**: Trong giao dịch điện tử SET, thông tin đơn hàng OI cần đc gửi cho ng bán. OI thường đc ghép với chữ kí kép và PIMD trc khi gửi đi. Ghép PIMD vào để làm gì, bỏ đi có đc k?

**Câu 3**: Ltrinh nhập 1 chuỗi kí tự plaintext từ bàn phím (các kí tự nằm trong khoảng từ (‘A’ -> ‘Z’). Nhập 2 số nguyên x, y € Z26 từ bàn phím. Mã hóa chuỗi ban đầu theo CT: C = P/x + y

Trong đó P là kí tự bản rõ, C: kí tự mã hóa, các phép tính đc thực hiện trong Z26. Hiện chuỗi sau khi mã hóa. Giải mã và hiện chuỗi sau khi mã hóa ra màn hình

Giải

Câu 1: E(K, [M || E(PRA, H(M)) ]) || E(PUB, K)



**Bên gửi :** A gửi B đoạn mã gồm 2 phần ghép với nhau:

Phần 1: Mã hóa thông điệp M với hàm hash thu được H(M), Mã hóa H(M) với khóa PRa bởi hàm mã hóa E thu được E(PRa,H(M)), lấy mã thu được ghép nối với M ta được

M|| E(PRa,H(M)). Mã hóa M|| E(PRa,H(M)) với khóa K bởi hàm mã hóa E thu được E(K, M|| E(PRa,H(M)))

Phần 2: Khóa K đưa qua khóa PUb của hàm E thu được E(PUb, K))

**Bên nhận:** B nhận được đoạn mã từ A:

+ Giải mã hàm E(K, M|| E(PRa,H(M))) qua hàm giải mã D với khóa K ta thu được M|| E(PRa,H(M))

+ Mã hóa thông điệp M với hàm hash thu được H(M) (\*)

+ Giải mã hàm E(PRa,H(M)) qua hàm giải mã D với khóa công khai PUb thu được H(M) (\*\*)

* So sánh hàm Hash (\*) thu được và hàm Hash (\*\*) trong đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền hay ko.

**Tác dụng:** Xác thực nội dung thông điệp có chính xác không

Khóa PRa: Xác thực nguồn gốc thông điệp, khóa PUb: Bảo mật thông điệp trên đường truyền, E(K): Có tác dụng bảo mật, khóa PRb: xác thực nguồn gốc thông điệp.

**Câu 2**: - Ghép PIMD vào có 2 mục đích chính:

* Bảo vệ tính toàn vẹn: Khi PIMD được tính toán và ghép vào OI, bất kỳ thay đổi nào trong OI sẽ dẫn đến thay đổi giá trị của PIMD. Khi nhận được thông tin đơn hàng cùng với PIMD từ người mua, người bán có thể tính toán lại PIMD từ OI nhận được và so sánh nó với PIMD ban đầu. Nếu hai giá trị không khớp, điều này cho thấy thông tin đơn hàng đã bị thay đổi trong quá trình truyền tải hoặc bị sửa đổi.
* Xác thực nguồn gốc: PIMD cũng có thể được sử dụng để xác thực nguồn gốc của OI. Khi người mua tạo PIMD từ OI và gửi cho người bán, người bán có thể tính toán lại PIMD từ OI nhận được và so sánh nó với PIMD được gửi từ người mua. Nếu hai giá trị khớp nhau, điều này cho thấy OI được tạo bởi người mua cụ thể và không bị sửa đổi trong quá trình truyền tải.

- Việc bỏ PIMD đi có thể làm mất đi tính toàn vẹn và khả năng xác thực của OI , người bán sẽ không còn cách nào để kiểm tra tính toàn vẹn và xác thực của OI từ nguồn gốc.

🡪 việc giữ PIMD trong thông tin đơn hàng là cần thiết để đảm bảo tính bảo mật và xác thực của giao dịch.

**ĐỀ 4**

**Câu 1**: Cho CT mô tả dl trên đường truyền, hãy vẽ sơ đồ qtrinh truyền và nhận dl từ A -> B, rồi gthich tác dụng của sơ đồ đó**: E(K,M) || H(E(K, M))**

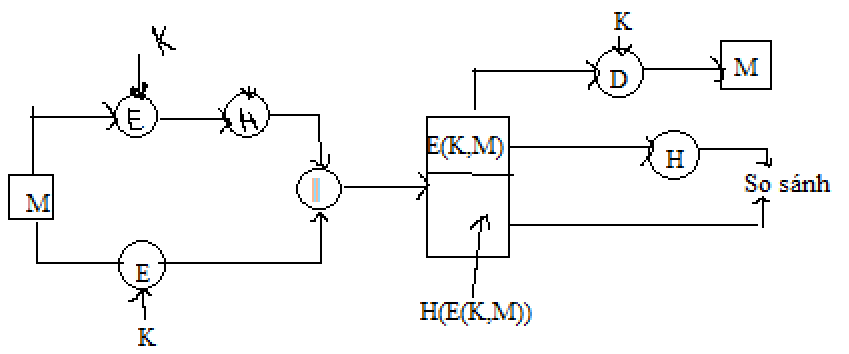
**Câu 2**: Loại hàm nào thường được sử dụng trong mật mã khóa công khai? Đặc điểm cơ bản của chúng là gì? Hãy lấy VD 1 hàm như vậy và giải thích

**Câu 3**: Ltrinh nhập 1 chuỗi kí tự plaintext từ bàn phím (các kí tự nằm trong khoảng từ (‘A’ -> ‘Z’). Nhập 2 số nguyên a,b € Z26 từ bàn phím. Mã hóa chuỗi ban đầu theo CT: C = P/a + b

Trong đó P là kí tự bản rõ, C: kí tự mã hóa, các phép tính đc thực hiện trong Z26. Hiện chuỗi sau khi mã hóa. Giải mã và hiện chuỗi sau khi mã hóa ra màn hình

Giải:

Câu 1: E(K,M) || H(E(K, M))



**Bên gửi :** A gửi B đoạn mã gồm 2 phần ghép với nhau:

Phần 1: Mã hóa thông điệp M với khóa K bởi hàm mã hóa E thu được E(K,M)

Phần 2: Mã hóa thông điệp M với khóa K bởi hàm mã hóa E thu được E(K,M) rồi đưa toàn bộ E(K,M) qua hàm Hash thu được H(E(K,M))

**Bên nhận:**

* + - * B nhận được đoạn mã từ A:
        + Đưa hàm mã hóa E(K, M) đưa qua hàm hash thu được mã hask H(E(K, M)). So sánh hàm Hash thu được và hàm Hash trong đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền hay ko.

**Tác dụng:** Xác thực nội dung thông điệp có chính xác không

Khóa E(K): Có tác dụng bảo mật.

Câu 2:

Các hàm toán học được sử dụng phổ biến nhất trong mật mã khóa công khai là các hàm dựa trên các thuộc tính của các số nguyên tố lớn, chẳng hạn như thuật toán RSA và Mật mã đường cong Elliptic (ECC).

Một đặc điểm cơ bản của các hàm này là chúng dễ dàng tính toán để thực hiện theo một hướng (ví dụ: mã hóa), nhưng cực kỳ khó đảo ngược (ví dụ: giải mã) nếu không biết một số thông tin nhất định, chẳng hạn như khóa riêng.

RSA là một ví dụ về chức năng như vậy. Nó dựa trên các tính chất toán học của các số nguyên tố lớn. Để tạo một cặp khóa, hai số nguyên tố lớn được chọn ngẫu nhiên và nhân với nhau để tạo khóa chung. Các yếu tố của khóa công khai này sau đó được sử dụng để lấy khóa riêng, được giữ bí mật bởi chủ sở hữu của cặp khóa. Dữ liệu có thể được mã hóa bằng khóa chung và chỉ có thể được giải mã bằng khóa riêng tương ứng.

- Trong mật mã khóa công khai. Hàm cửa lật một chiều thường được sử dụng.

- Đặc điểm của hàm 1 chiều là tính toàn y = f(x) là dễ nhưng việc tính hàm ngược x = f-1(y) là không thể thực hiện hay là thời gian tiêu tốn để tính được hàm ngược là vô cùng lớn nhưng có cửa lật z để tính x = f-1z(y) là dễ.

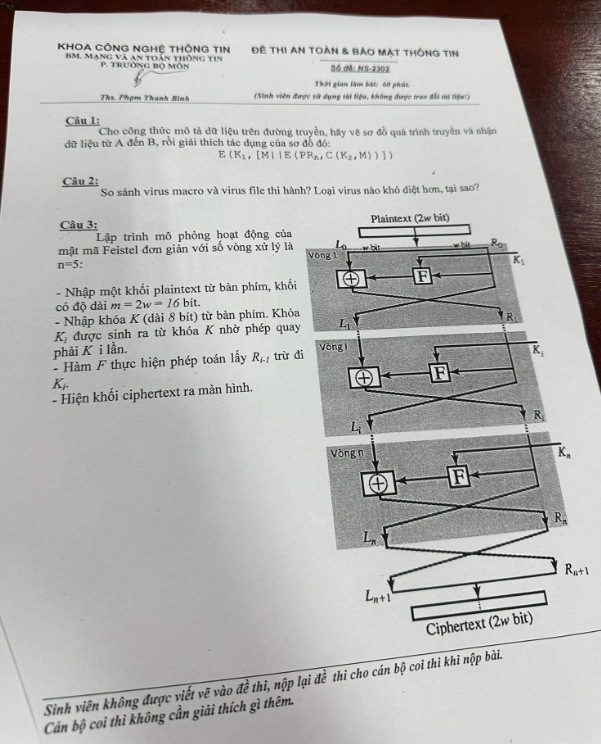
- Ví dụ: Hàm f(x) = xa mod n với n = p.q là tích của 2 số nguyên tố lớn là một hàm một chiều vì tính f(x) là khá dễ dàng nhưng tính hàm ngược x = f-1(y) là rất khó và tốn rất nhiều thời gian tuy nhiên nếu biết cửa lật là hai thừa số p.q của n thì sẽ tính được x = f-1(y) là rất dễ dàng.

Câu 3:

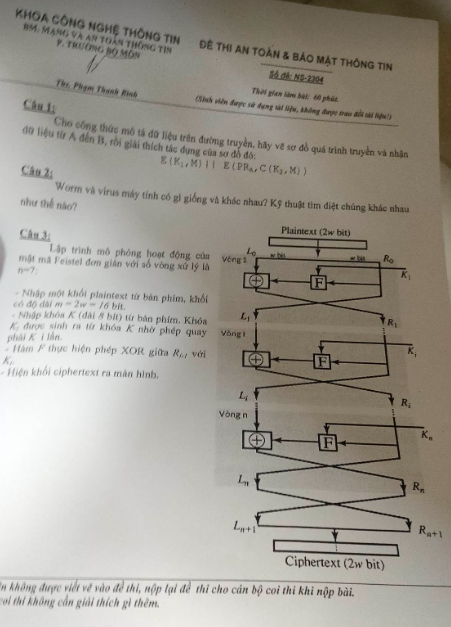
|  |
| --- |
| #include<iostream>  #include<cstring>  using namespace std;  int KT\_S(char s) {  return s-'A';  }  char S\_KT(int n) {  return 'A'+n;  }  int main() {  //Affine  string s,o;  int a,b;  cout<<"Nhap S ",cin>>s;  do {  cout<<"A = ",cin>>a;  } while(a<1||a>25||a%2==0||a==13);  // cout<<"A = ",cin>>a;  cout<<"B = ",cin>>b;  //ma hoa  int a1;  for( a1=1; a1<26; a1=a1+2) {  if(((a\*a1)%26)==1) break;  }  for(int i=0; i<s.size(); i++) {  int c=KT\_S(s[i]);  cout<<c<<endl;  int p=((c\*a1+b))%26;  cout<<p<<endl;  o+=S\_KT(p);  }  cout<<o<<endl;  ///giai ma  string o\_out;  // int a1;  // for( a1=1; a1<26; a1=a1+2) {  // if(((a\*a1)%26)==1) break;  // }  for(int i=0; i<o.length(); i++) {  int c1=KT\_S(o[i]) ;//nghich dao cua a  int p1=((a\*(c1-b+26))%26);  o\_out+=S\_KT(p1);  }  cout<<o\_out;  } |

**ĐỀ AN TOÀN BẢO MẬT K62**

**ĐỀ 1**



ĐỀ 2



ĐỀ 3:

Câu 1: E(PUB, M) || E(PRA, C(K,M)

Câu 2: So sánh sự giống và khác nhau của Hash và MAC? Hàm nào sử dụng chữ kí số tốt hơn? Tại sao?

Câu 3: Giống đề 2 ở trên, chỉ khác ở chỗ:

- Số vòng xử lý là n=4

- Hàm F thực hiện phép OR