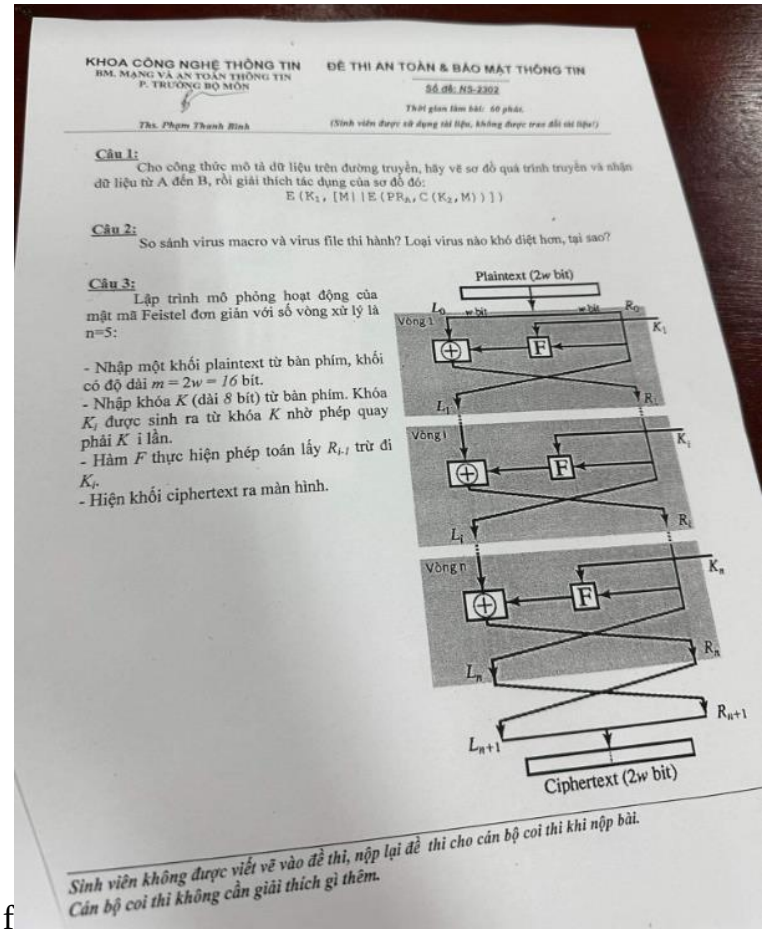
ĐỀ 1



KHOA CÔNG NGHỆ THÔNG TIN
BM. MẠNG VÀ AN TOÀN THÔNG TIN
P. TRƯỞNG BỘ MÔN

Ths. Phạm Thanh Bình

ĐỀ THI AN TOÀN & BẢO MẬT THÔNG TIN

Số đề: NS-2302

Thời gian làm bài: 60 phút.

(Sinh viên được sử dụng tài liệu, không được trao đổi tài liệu!)

Câu 1:
Cho công thức mô tả dữ liệu trên đường truyền, hãy vẽ sơ đồ quá trình truyền và nhận dữ liệu từ A đến B, rồi giải thích tác dụng của sơ đồ đó:

$E(K_1, [M] | E(PR_A, C(K_2, M))]$

Câu 2:
So sánh virus macro và virus file thi hành? Loại virus nào khó diệt hơn, tại sao?

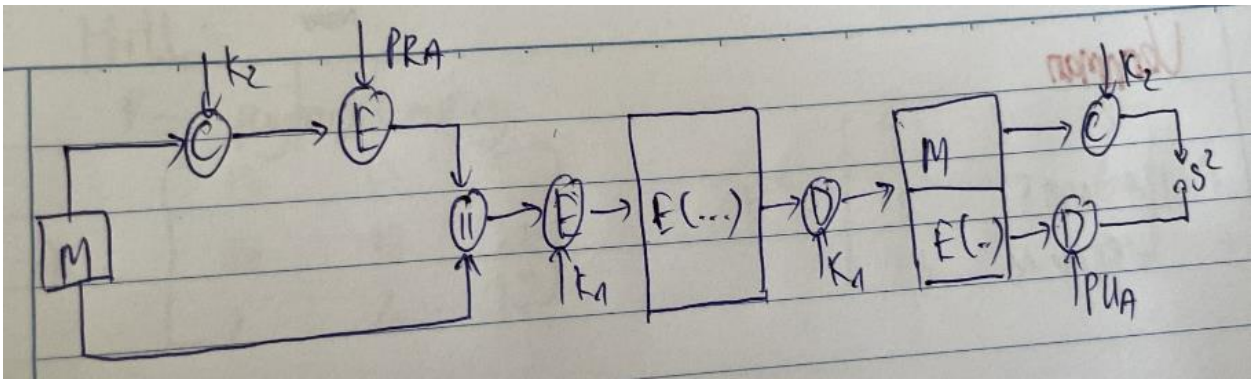
Câu 3:
Lập trình mô phỏng hoạt động của mật mã Feistel đơn giản với số vòng xử lý là n=5:

- Nhập một khối plaintext từ bàn phím, khối có độ dài $m = 2w = 16$ bít.
- Nhập khóa K (dài 8 bít) từ bàn phím. Khóa K_i được sinh ra từ khóa K nhờ phép quay phải K i lần.
- Hàm F thực hiện phép toán lấy R_{i-1} trừ đi K_i.
- Hiện khối ciphertext ra màn hình.

Sinh viên không được viết vẽ vào đề thi, nộp lại đề thi cho cán bộ coi thi khi nộp bài.
Cán bộ coi thi không cần giải thích gì thêm.

f

Câu 1:



Bên gửi: A gửi B đoạn mã

- Mã hóa thông điệp M với hàm Mac khóa K2 thu được C(K2,M)

- Mã hóa C(K2,M) với khóa riêng PRa bởi hàm mã hóa E thu được E(PRa,C(K2,M))
- Ghép nối E(PRa,C(K2,M)) với thông điệp M được M || E(PRa,C(K2,M))
- Đưa tất cả mã hóa với hàm mã hóa E với khóa K1 thu được E(K1, (M || E(PRa,C(K2,M))))

Bên nhận: B nhận được đoạn mã gửi từ A

- Đưa hàm mã hóa E(K1, (M || E(PRa,C(K2,M)))) vào hàm giải mã D khóa K1 thu được M || E(PRa,C(K2,M))
- Đưa thông điệp M vào hàm mã hóa Mac - C với khóa K2 thu được C(K2,M) (*)
- Đưa hàm mã hóa E(PRa,C(K2,M)) vào hàm giải mã D với khóa công khai PUa thu được C(K2,M) (**)

-> So sánh hàm C(K2,M) (*) *với hàm C(K2,M)*(**) trong đoạn đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền không.

Tác dụng:

- Bảo mật thông điệp E(K1)
- Xác thực nguồn gốc thông điệp (PRa)
- Chứng thực thông điệp C(K2)

Câu 2:

Virus File thi hành	Virus Macro
Loại virus này lây nhiễm vào các file nhị phân thi hành được (.EXE, .COM, .DLL, .BIN, .SYS…)	Loại virus này lây nhiễm vào các macro trong các file tài liệu của MicroSoft Office (Word, Excel…)
Đoạn mã virus có thể được gắn vào đầu file, cuối file, hoặc giữa file	Một số macro có khả năng tự khi hành khi mở file, cất file… Virus Macro thường nằm trong các macro tự động đó.
Khi file được chạy, virus sẽ được kích hoạt, nó sẽ tìm cách lây vào các file khác trong máy	Khi file được mở, virus sẽ được kích hoạt, sau đó nó tìm cách lây vào các file tài liệu khác.

Virus file thi hành có xu hướng khó diệt hơn so với virus macro.

Kỹ thuật biến đổi và ẩn nấp: Virus file thi hành thường sử dụng kỹ thuật biến đổi mã và ẩn nấp phức tạp, khiến việc phát hiện và diệt trừ khó khăn hơn.

Thiệt hại hệ thống: Virus file thi hành có thể gây ra thiệt hại nghiêm trọng cho hệ thống, làm hỏng các file hệ thống quan trọng, điều này làm cho việc khôi phục hệ thống phức tạp hơn

Câu 3:

```cpp
//feistel

#include <iostream>
#include <string>
using namespace std;

char F(char R, char Ki) {
    return R - Ki;
}

char rotateRight(char ch, int i) {
    return (char)((ch >> i) | (ch << (8 - i)));
}

string MHKhoi(char P0, char P1, char k) {
    char K[6], L[6], R[6];
    string C = "  ";  // Cần có độ dài đủ để chứa 2 ký tự

    // Mã hóa khối
    R[0] = P0;
    L[0] = P1;
    K[0] = k;

    for (int i = 1; i <= 5; i++) {
        K[i] = rotateRight(K[0], i); // Quay phải i lần
        R[i] = L[i-1] ^ F(R[i-1], K[i]);
        L[i] = R[i-1];
    }

    C[0] = L[5];
    C[1] = R[5];
    return C;
}
```

```cpp
int main() {
    string P, C;
    char k;
    cout << "Nhap chuoi plaintext: ";
    getline(cin, P);
    cout << "Nhap khoa K: ";
    cin >> k;

    if (P.size() % 2 == 1) P += 'X';  // Nếu độ dài P lẻ, thêm 'X' vào cuối

    for (int i = 0; i < P.size(); i += 2)
        C += MHKhoi(P[i], P[i + 1], k);

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

    return 0;
}
```

Mã hóa + Giải mã

```cpp
// ------------ feistel mã hóa + giải mã
#include <iostream>
#include <string>
using namespace std;

char F(char R, char Ki) {
   return R - Ki;
}

char rotateRight(char ch, int i) {
    return (char)((ch >> i) | (ch << (8 - i)));
}

string MHKhoi(char P0, char P1, char k) {
   char K[6], L[6], R[6];
   string C = "  ";

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

   for(int i = 1; i <= 5; i++) {
       K[i] = rotateRight(K[0], i); // Quay phải i lần
       R[i] = L[i-1] ^ F(R[i-1], K[i]);
       L[i] = R[i-1];
   }

   C[0] = L[5];
   C[1] = R[5];
   return C;
}
```

```cpp
string GiaiMaKhoi(char C0, char C1, char k) {
    char K[6], L[6], R[6];
    string P = "  ";

    L[5] = C0;
    R[5] = C1;
    K[0] = k;

    for(int i = 1; i <= 5; i++) {
        K[i] = rotateRight(K[0], i); // Quay phải i lần
    }

    for(int i = 5; i >= 1; i--) {
        R[i-1] = L[i];
        L[i-1] = R[i] ^ F(R[i-1], K[i]);
    }

    P[0] = R[0];
    P[1] = L[0];
    return P;
}

int main() {
    string P, C, decryptedP;
    char k;
    cout << "Nhap chuoi plaintext: ";
    getline(cin, P);
    cout << "Nhap khoa K: ";
    cin >> k;

    if (P.size() % 2 == 1) P += 'X';

    for(int i = 0; i < P.size(); i += 2)
        C += MHKhoi(P[i], P[i+1], k);

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

    for(int i = 0; i < C.size(); i += 2)
        decryptedP += GiaiMaKhoi(C[i], C[i+1], k);

    if (!decryptedP.empty() && decryptedP[decryptedP.size() - 1] == 'X') {
        decryptedP.erase(decryptedP.size() - 1);
    }

    cout << "Chuoi giai ma: " << decryptedP << endl;

    return 0;
}
```

- **ĐỀ 2**

KHOA CÔNG NGHỆ THÔNG TIN
BM. MẠNG VÀ AN TOÀN THÔNG TIN
P. TRƯỞNG BỘ MÔN

Ths. Phạm Thanh Bình

ĐỀ THI AN TOÀN & BẢO MẬT THÔNG TIN

Số đề: NS-2304

Thời gian làm bài: 60 phút.
(Sinh viên được sử dụng tài liệu, không được trao đổi tài liệu!)

Câu 1:

Cho công thức mô tả dữ liệu trên đường truyền, hãy vẽ sơ đồ quá trình truyền và nhận dữ liệu từ A đến B, rồi giải thích tác dụng của sơ đồ đó:

$$E(K_1, M) \,||\, E(PR_A, C(K_2, M))$$

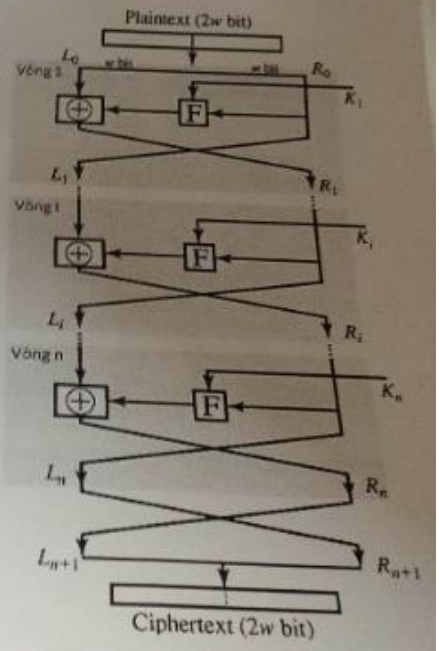
Câu 2:

Worm và virus máy tính có gì giống và khác nhau? Kỹ thuật tìm diệt chúng khác nhau như thế nào?

Câu 3:

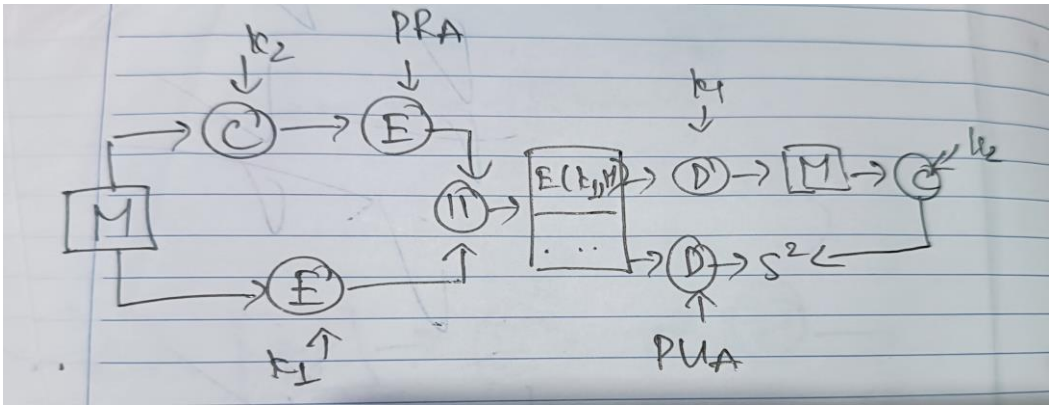
Lập trình mô phỏng hoạt động của mật mã Feistel đơn giản với số vòng xử lý là n=7.

- Nhập một khối plaintext từ bàn phím, khối có độ dài $m = 2w = 16$ bit.
- Nhập khóa K (dài 8 bit) từ bàn phím. Khóa K_i được sinh ra từ khóa K nhờ phép quay phải K i lần.
- Hàm F thực hiện phép XOR giữa R_{i-1} với K_i.
- Hiện khối ciphertext ra màn hình.



n không được viết vẽ vào đề thi, nộp lại đề thi cho cán bộ coi thi khi nộp bài.
oi thi không cần giải thích gì thêm.

Câu1:



Bên gửi: A gửi B đoạn mã

- Phần 1:
- Mã hóa thông điệp M với hàm Mac khóa K2 thu được C(K2,M)
- Mã hóa C(K2,M) với khóa riêng PRa bởi hàm mã hóa E thu được E(PRa,C(K2,M))
- Phần 2:
- Mã hóa thông điệp M với hàm mã hóa E khóa K1 thu được E(K1,M)

Bên nhận: B nhận được đoạn mã gửi từ A

- Đưa hàm mã hóa E(K1,M) vào hàm giải mã D khóa K1 thu được thông điệp M
- Đưa thông điệp M vào hàm mã hóa C với khóa K2 thu được C(K2,M) (*)
- Đưa hàm mã hóa E(PRa,C(K2,M)) vào hàm giải mã D với khóa công khai PUa thu được C(K2,M) (**)

-> So sánh hàm C(K2,M) (*) *với hàm C(K2,M)*(**) trong đoạn đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền không.

Tác dụng:

- Bảo mật thông điệp E(K1)
- Xác thực nguồn gốc thông điệp (PRa)
- Chứng thực thông điệp C(K2)

Câu3:

```cpp
/// feistel phép XOR----------------------------------------
#include <iostream>
#include <string>
using namespace std;

char F(char R, char Ki) {
    return R ^ Ki;  // XOR
}

char rotateRight(char k, int n) {
    return (k >> n) | (k << (8 - n));  // Quay phải n lần
}

string MHKhoi(char P0, char P1, char k) {
    char K[8], L[8], R[8];
    string C = "  ";

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

    for(int i = 1; i <= 7; i++) {
        K[i] = rotateRight(K[0], i); //Quay phải n lần
        R[i] = L[i-1] ^ F(R[i-1], K[i]);
        L[i] = R[i-1];
    }

    C[0] = L[7];
    C[1] = R[7];
    return C;
}

string GiaiMaKhoi(char C0, char C1, char k) {
    char K[8], L[8], R[8];
    string P = "  ";

    L[7] = C0;
    R[7] = C1;
    K[0] = k;

    for(int i = 1; i <= 7; i++) {
        K[i] = rotateRight(K[0], i); //Quay phải i lần
    }

    for(int i = 7; i >= 1; i--) {
        R[i-1] = L[i];
        L[i-1] = R[i] ^ F(R[i-1], K[i]);
    }

    P[0] = R[0];
    P[1] = L[0];
    return P;
}
```

```cpp
int main() {
    string P, C, decryptedP;
    char k;
    cout << "Nhap chuoi plaintext: ";
    getline(cin, P);
    cout << "Nhap khoa K: ";
    cin >> k;

    if (P.size() % 2 == 1) P += 'X';

    for(int i = 0; i < P.size(); i += 2)
        C += MHKhoi(P[i], P[i+1], k);

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

    // Giải mã
    for(int i = 0; i < C.size(); i += 2)
        decryptedP += GiaiMaKhoi(C[i], C[i+1], k);

    if (!decryptedP.empty() && decryptedP[decryptedP.size() - 1] == 'X') {
        decryptedP.erase(decryptedP.size() - 1);
    }

    cout << "Chuoi giai ma: " << decryptedP << endl;

    return 0;
}
```

- **ĐỀ 3**

Câu 1: E(PUB, M) || E(PRA, C(K,M))

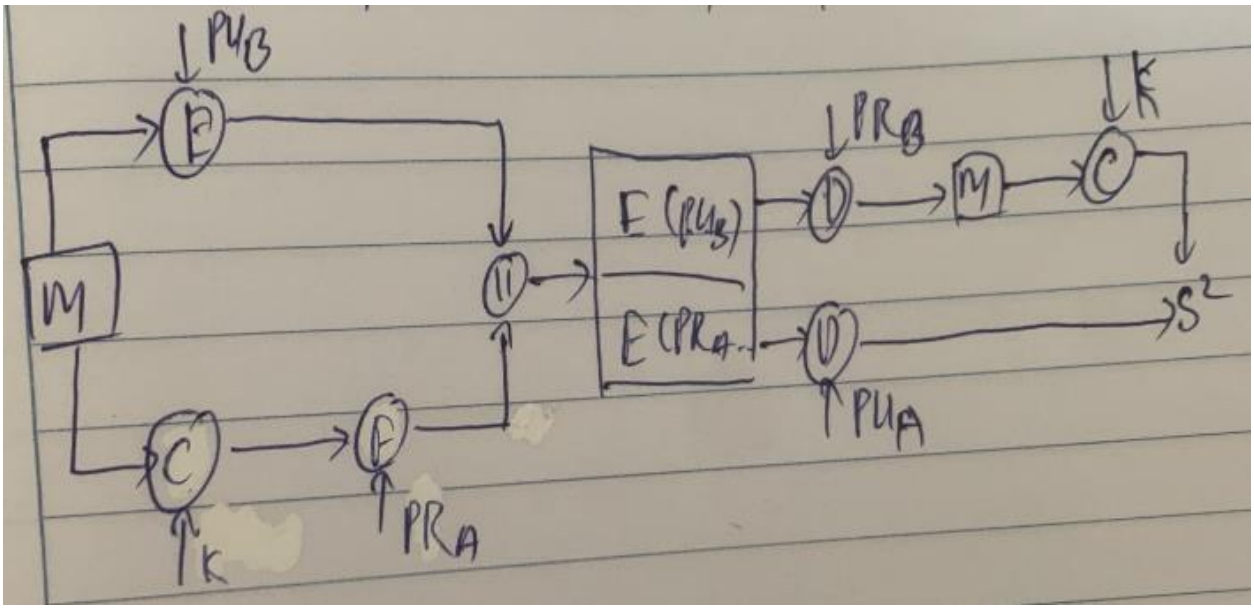
Câu 2: So sánh sự giống và khác nhau của Hash và MAC? Hàm nào sử dụng chữ ký số tốt hơn? Tại sao?

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

- Số vòng xử lý là n=4

- Hàm F thực hiện phép OR

Câu 1:



Bên gửi: A gửi B đoạn mã

- Phần 1:
- Mã hóa thông điệp M với hàm mã hóa E khóa công khai PUb thu được hàm E(PUb,M)
- Phần 2:
- Mã hóa thông điệp M với hàm mã hóa Mac-C với khóa K thu được C(K,M)
- Đưa hàm mã hóa C(K,M) vào hàm E khóa riêng PRa thu được hàm E(PRa,C(K,M))

Bên nhận: B nhận được đoạn mã gửi từ A

- Đưa hàm mã hóa E(PUb,M) vào hàm giải mã D khóa riêng PRb thu được thông điệp M.
- Đưa thông điệp M vào hàm mã hóa C khóa K1 thu được C(K1,M)(*)

- Đưa hàm mã hóa E(PRa,C(K,M)) vào hàm giải mã D khóa công khai PUa thu được C(K,M) (**)

-> So sánh hàm C(K,M) (*) với hàm C(K,M)*(**) trong đoạn đoạn mã để xác thực nội dung thông điệp có bị thay đổi trên đường truyền không.

Tác dụng:

- Xác thực nguồn gốc thông điệp (PRa)
- Bảo mật thông điệp trên đường truyền (PUb)
- Chứng thực thông điệp C(K)

Câu 2:

- Giống:
- **Tính toán giá trị cố định:** Cả hàm Hash và hàm MAC đều nhận đầu vào có độ dài bất kỳ và trả về một giá trị có độ dài cố định.
- **Đảm bảo tính toàn vẹn:** Cả hai đều dùng để đảm bảo tính toàn vẹn của dữ liệu, nghĩa là một thay đổi nhỏ trong đầu vào sẽ dẫn đến sự thay đổi lớn trong giá trị đầu ra.
- **Ứng dụng trong bảo mật:** Cả hai đều là công cụ mật mã học quan trọng và được sử dụng rộng rãi trong các ứng dụng bảo mật.

	Mac	Hash
Khóa bí mật	Sử dụng khóa bí mật cùng với đầu vào để tính toán giá trị MAC, đảm bảo cả tính toàn vẹn và xác thực nguồn gốc của dữ liệu	Không sử dụng khóa bí mật. Giá trị băm được tính toán chỉ dựa vào đầu vào
Mục đích sử dụng	Dùng để đảm bảo tính toàn vẹn và xác thực nguồn gốc của dữ liệu, thường được sử dụng trong truyền thông mạng và các giao thức bảo mật.	Chủ yếu để kiểm tra tính toàn vẹn của dữ liệu. Dùng trong lưu trữ mật khẩu, chữ ký số, và kiểm tra tính toàn vẹn của tập tin

Hàm Hash sử dụng tốt hơn cho chữ ký số.

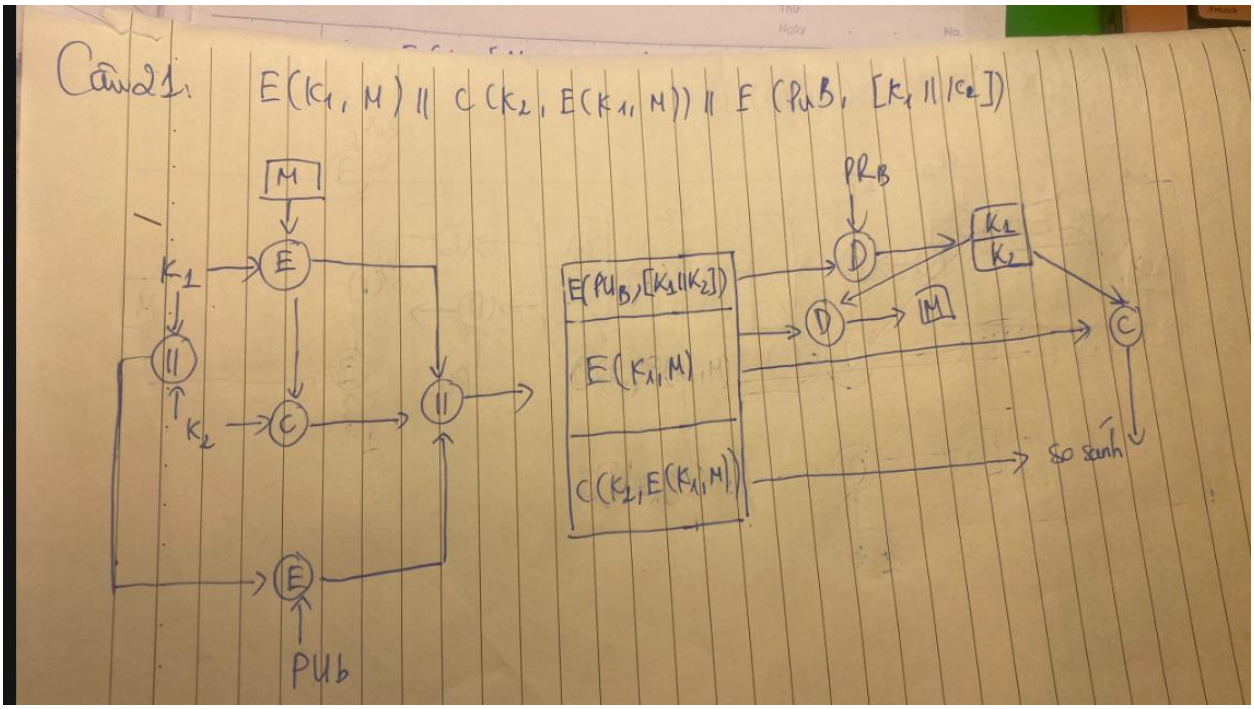
1. **Tính toàn vẹn và hiệu quả:** Hàm hash nhanh chóng tạo ra một giá trị băm nhỏ gọn đại diện cho dữ liệu, giúp phát hiện thay đổi trong dữ liệu một cách hiệu quả.
2. **Hệ thống khóa công khai:** Chữ ký số dựa trên mã hóa giá trị băm bằng khóa riêng và xác minh bằng khóa công khai, phù hợp với cách hoạt động của hàm hash.

3. **Không yêu cầu khóa bí mật:** Hàm hash không cần khóa bí mật để tính toán giá trị, đơn giản hóa quy trình tạo và xác minh chữ ký số.

Câu 3:

```cpp
// ----------------------feistel OR
#include <iostream>
#include <string>
using namespace std;

char F(char R, char Ki) {
    return R | Ki;  // OR
}

char rotateRight(char k, int n) {
    return (k >> n) | (k << (8 - n));  // Quay phải n lần
}

string MHKhoi(char P0, char P1, char k) {
    char K[5], L[5], R[5];
    string C = "  ";

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

    for(int i = 1; i <= 4; i++) {
        K[i] = rotateRight(K[0], i); // Quay phải i lần
        R[i] = L[i-1] ^ F(R[i-1], K[i]);
        L[i] = R[i-1];
    }

    C[0] = L[4];
    C[1] = R[4];
    return C;
}
```

```cpp
string GiaiMaKhoi(char C0, char C1, char k) {
    char K[5], L[5], R[5];
    string P = "  ";

    L[4] = C0;
    R[4] = C1;
    K[0] = k;

    for(int i = 1; i <= 4; i++) {
        K[i] = rotateRight(K[0], i); // Quay phải i lần
    }

    for(int i = 4; i >= 1; i--) {
        R[i-1] = L[i];
        L[i-1] = R[i] ^ F(R[i-1], K[i]);
    }

    P[0] = R[0];
    P[1] = L[0];
    return P;
}

int main() {
    string P, C, decryptedP;
    char k;
    cout << "Nhap chuoi plaintext: ";
    getline(cin, P);
    cout << "Nhap khoa K: ";
    cin >> k;

    if (P.size() % 2 == 1) P += 'X';  /

    for(int i = 0; i < P.size(); i += 2)
        C += MHKhoi(P[i], P[i+1], k);

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

    // Giải mã
    for(int i = 0; i < C.size(); i += 2)
        decryptedP += GiaiMaKhoi(C[i], C[i+1], k);

    if (!decryptedP.empty() && decryptedP[decryptedP.size() - 1] == 'X') {
        decryptedP.erase(decryptedP.size() - 1);
    }

    cout << "Chuoi giai ma: " << decryptedP << endl;

    return 0;
}
```



- Bên gửi: A gửi B đoạn mã gồm 3 phần ghép với nhau:

+ Phần 1: Thông điệp M được mã hóa bởi hàm E với khóa K1

+ Phần 2: Thông điệp M được mã hóa bởi hàm E đưa qua hàm MAC với mật khóa K2 thu được C(K2, E(K1, M)

+ Phần 3: Sau đó ghép toàn bộ với hàm E gồm khóa K2 với khóa K1 được mã hóa bởi khóa công khai Pub

- Bên nhận:

+ Giải mã hàm E bằng khóa riêng PrB thu được 2 khóa K1 và K2

+ Giải mã hàm E biết khóa K1 thu được thông điệp M

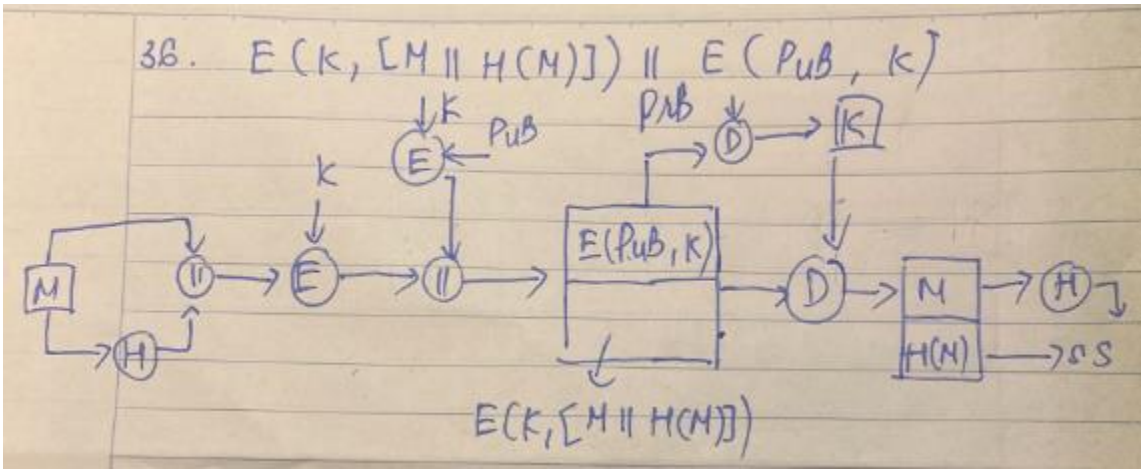
+ Đưa hàm E(K1, M) qua hàm MAC với khóa K2 thu được hàm MAC (C(K2, E(K1, M)) (1)

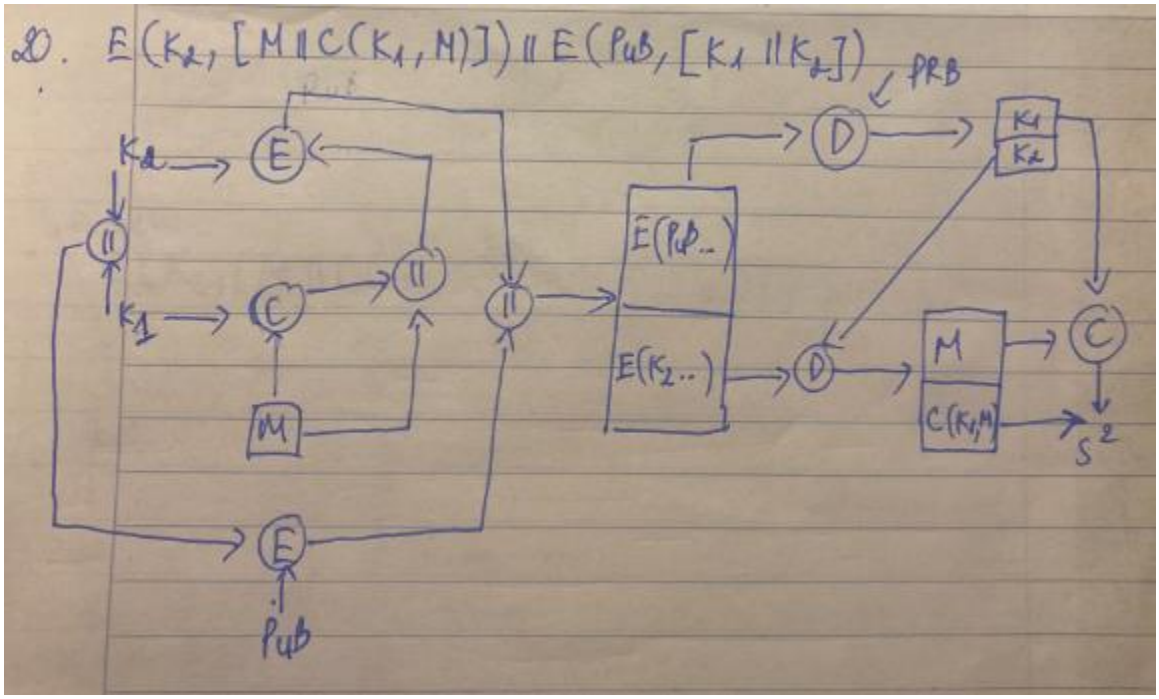
So sánh 2 hàm MAC (1) và (2) để xác thực nội dung thông điệp có bị thay đổi trên đường truyền hay ko

- Tác dụng:

+ Hàm MAC: Xác thực nội dung thông điệp có chính xác không

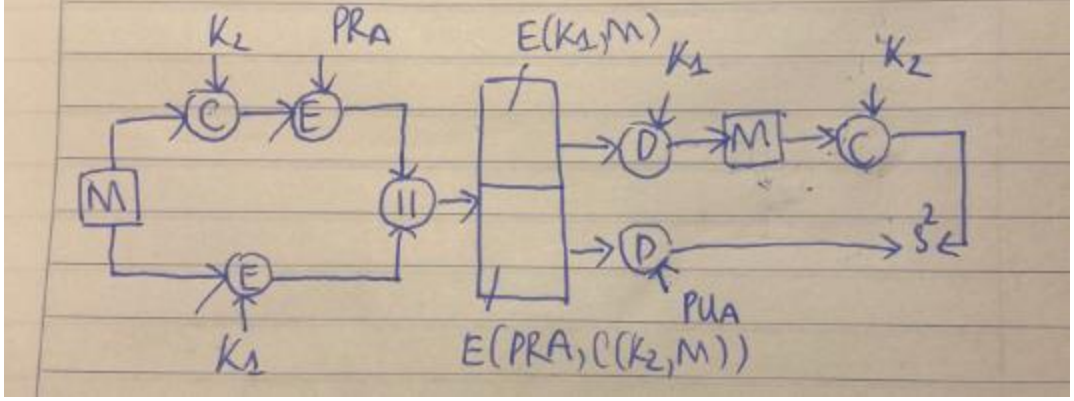
+ Có tác dụng giữ bí mật K1, Pu



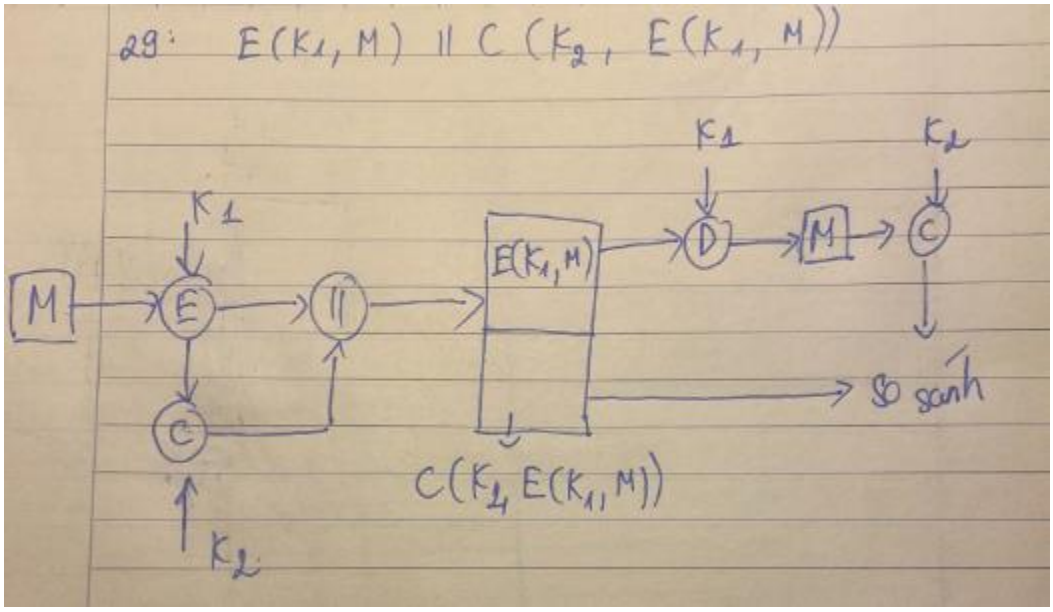


Câu 35

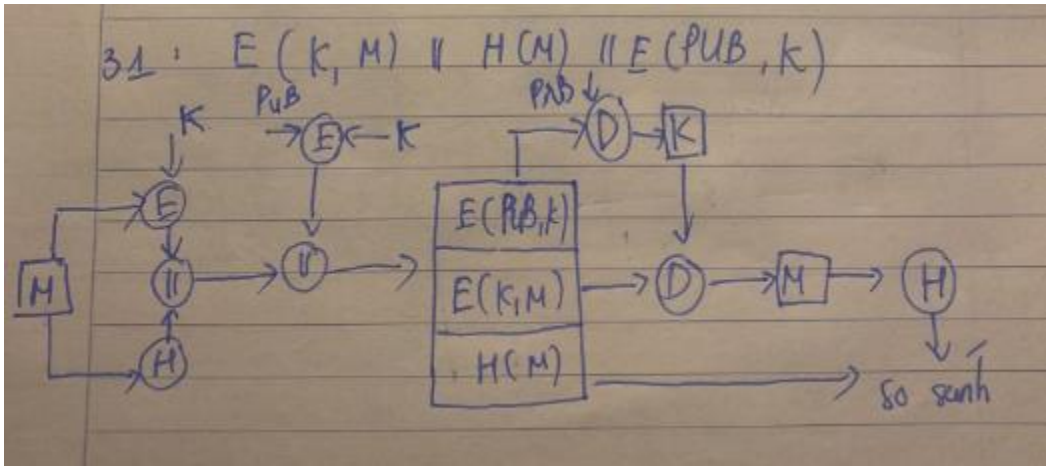
$E(K1, M) \| E(PR_A, C(K2, M))$



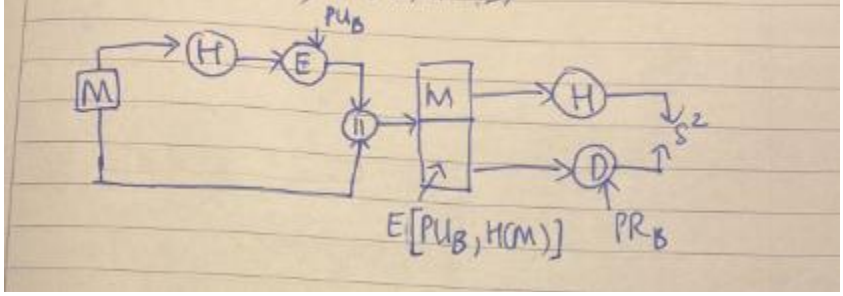
29: $E(K_1, M) \| C(K_2, E(K_1, M))$



31: $E(K, M) \| H(M) \| E(PUB, K)$

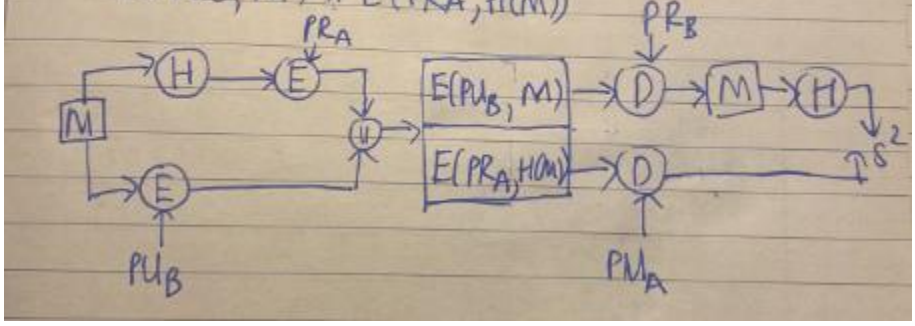


Câu 30. $E(PU_B, [H(M) \| M])$



Câu 32

$E(PU_B, M) \| E(PR_A, H(M))$



Câu 33

$E(PU_B, [M \| E(PR_A, C(K, M))])$