LỜI CẢM ƠN

Trước hết em xin gửi lời cảm ơn đến PGS. TS. Trịnh Nhật Tiến, người thầy đã hướng dẫn em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành khóa luận này từ lý thuyết đến ứng dụng. Sự hướng dẫn của thầy đã giúp em có thêm được những hiểu biết về một số vấn đề liên quan đến bảo mật thông tin. Qua đó, những lý thuyết bảo mật cũng lôi cuốn em và sẽ trở thành hướng nghiên cứu tiếp của em sau khi tốt nghiệp.

Đồng thời em cũng xin chân thành cảm ơn các thầy cô trong bộ môn cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành tốt khóa luận này.

Em xin gửi lời cảm ơn đến các thành viên lớp CT1001, những người bạn đã luôn ở bên cạnh động viên, tạo điều kiện thuận lợi và cùng em tìm hiểu, hoàn thành tốt khóa luận.

Sau cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè đã tạo mọi điều kiện để em xây dựng thành công khóa luận này.

Hải Phòng, tháng 7 năm 2010

Sinh viên thực hiện

LÂM THỊ THANH TUYỀN

MỤC LỤC

LỜI NÓI ĐẦU 1
Chương 1. CÁC KHÁI NIỆM CƠ BẢN.....2
1.1. MỘT SỐ KHÁI NIỆM TOÁN HỌC 2
1.1.1. Các khái niệm trong số học 2
1.1.1.1. Ước chung lớn nhất.....2
1.1.1.2. Số nguyên tố.....4
1.1.1.3. Hàm ϕ Euler 4
1.1.1.4. Đồng dư thức.....4
1.1.2. Các khái niệm trong đại số 5
1.1.2.1. Không gian Z_n 5
1.1.2.2. Nhóm nhân Z_n^*.....10
1.1.2.3. Phần tử sinh.....11
1.1.2.4. Thặng dư 11
1.1.3. Khái miệm độ phức tạp của thuật toán 12
1.1.3.1. Khái niệm thuật toán.....12
1.1.3.2. Khái niệm độ phức tạp của thuật toán.....12
1.1.3.3. Lớp bài toán P, NP và NP – complete 14
1.2. VẤN ĐỀ MÃ HÓA.....16
1.2.1. Một số khái niệm 16
1.2.2. Mã hóa khóa đối xứng 17
1.2.3. Mã hóa khóa bất đối xứng 18
1.3. VẤN ĐỀ CHỮ KÝ SỐ (digital signature).....20
1.3.1. Khái niệm.....20
1.3.2. Quá trình tạo ra chữ ký điện tử.....21
1.3.3. Hàm băm sử dụng trong ký điện tử 21
Chương 2. PHƯƠNG PHÁP CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN...22
2.1. KHÁI NIỆM CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN 22
2.1.1. Khái niệm chứng không tiết lộ thông tin (CM KTLTT) 22

2.1.2. Khái niệm về chứng minh tương hỗ 23
2.2. HỆ THỐNG CM KTLTT CHO TÍNH ĐẲNG CẤU CỦA ĐỒ THỊ.....25
2.2.1. Khái niệm đồ thị đẳng cấu 25
2.2.2. Định nghĩa hệ thống CM KTLTT hoàn thiện.....28
2.2.3. Định nghĩa hệ thống CM KTLTT hoàn thiện không điều kiện 31
2.2.4. Định lý về hệ thống chứng minh tương hỗ cho đồ thị đẳng cấu 33
2.3. HỆ THỐNG CM KTLTT CHO BÀI TOÁN THẶNG DƯ BẬC HAI.....35
2.3.1. Sơ đồ chứng minh.....35
2.3.2. Tính chất của sơ đồ.....35
2.3.3. Chứng minh sơ đồ có tính đầy đủ.....36
Chương 3. ỨNG DỤNG CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN 37
3.1. ỨNG DỤNG CM KTLTT TRONG BỎ PHIẾU ĐIỆN TỬ 37
3.1.1. Sơ đồ bỏ phiếu truyền thống.....37
3.1.2. Một số khái niệm 39
3.1.3. Chứng minh tính hợp lệ của lá phiếu (x, y) (Giao thức 1) 41
3.1.4. Chứng minh quyền sở hữu giá trị bí mật β (Giao thức 2) 45
3.1.5. Giai đoạn cử tri chuyển lá phiếu đến ban kiểm phiếu (phương án 2) 47
3.2. ỨNG DỤNG CM KTLTT TRONG SỬ DỤNG TIỀN ĐIỆN TỬ.....49
3.2.1. Khái niệm thanh toán điện tử.....49
3.2.2. Khái niệm tiền điện tử 49
3.2.3. Mô hình giao dịch mua bán bằng tiền điện tử 50
3.2.4. Vấn đề "tiền điện tử" 53
3.2.5. Lược đồ tiền điện tử Brand.....56
Chương 4. THỬ NGHIỆM CHƯƠNG TRÌNH 63
4.1. MÔ TẢ CHƯƠNG TRÌNH.....63
4.1.1. Giới thiệu 63
4.1.2. Các chức năng chính.....64
4.2.1. Cử tri chứng minh tính hợp lệ của lá phiếu 68
4.2.2. Người xác minh trung thực chứng minh có giữ tham số bí mật β 76
TÀI LIỆU THAM KHẢO.....80

DANH MỤC TỪ VIẾT TẮT

Ký hiệu viết tắt	Giải thích
CT	Cử tri
ƯCLN	Ước chung lớn nhất
gcd(m, n)	Ước chung lớn nhất của m và n
KP	Kiểm phiếu
TT	Người trung thực
CM KTLTT	Chứng minh không tiết lộ thông tin
TMĐT	Thương mại điện tử
TTĐT	Thanh toán điện tử
Prover	Người chứng minh
Verifier	Người xác minh

LỜI NÓI ĐẦU

Ngày nay, công nghệ thông tin đang phát triển mạnh mẽ, Internet đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày thì các hoạt động trao đổi thông tin, mua bán,…trên mạng Internet diễn ra thường xuyên và ngày phổ biến hơn. Chính vì vậy mà việc bảo mật, đảm bảo an toàn thông tin đang là nhu cầu cấp thiết. Trước các nhu cầu cấp thiết đó, lý thuyết về mật mã thông tin đã ra đời nhằm đảm bảo tính an toàn dữ liệu tại nơi lưu trữ cũng như khi dữ liệu đang được truyền trên mạng.

Khoá luận này gồm có 4 chương với các nội dung:

Chương 1. CÁC KHÁI NIỆM CƠ BẢN

Chương 2. PHƯƠNG PHÁP CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

Chương 3. ỨNG DỤNG CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

Chương 4. THỬ NGHIỆM CHƯƠNG TRÌNH

"Chứng minh không tiết lộ thông tin", là phương pháp chứng minh không có nghĩa là "không để lộ thông tin" mà là "để lộ thông tin ở mức ít nhất" về sự vật, sự việc cần chứng minh. Với việc "không để lộ" người xác minh sẽ không có nhiều hiểu biết về sự vật sự việc, họ chỉ thu được chút ít thông tin (coi như là không) về đặc điểm tính chất của nó.

Ngành mật mã học luôn phát triển không ngừng, trong phạm vi khóa luận này, chúng tôi chỉ trình bày một vấn đề nhỏ là phương pháp "chứng minh không tiết lộ thông tin" đồng thời tìm hiểu một số ứng dụng thực tế của cơ sở lý thuyết này.

Chương 1. CÁC KHÁI NIỆM CƠ BẢN

1.1. MỘT SỐ KHÁI NIỆM TOÁN HỌC

1.1.1. Các khái niệm trong số học

1.1.1.1. Ước chung lớn nhất

1/. Ước số

Khái niệm:

Cho hai số nguyên a, b $\in$ Z, b $\neq$ 0. Nếu có một số nguyên q sao cho a = b*q, thì ta nói rằng a chia hết cho b, kí hiệu b|a. Ta nói b là ước của a, và a là bội của b.

Ví dụ:

Cho a = 6, b = 2, ta có 6 = 2*3, ký hiệu 2|6. Ở đây 2 là ước của 6 và 6 là bội của 2.

Tính chất:

Cho a, b, c $\in$ Z

+ a|a.

+ a|b, b|c $\Rightarrow$ a|c.

+ a|b, a|c $\Rightarrow$ a|(bx + cy) $\forall$ x, y $\in$ Z.

+ a|b, b|a $\Rightarrow$ a = $\pm$ b.

2/. Ước chung lớn nhất

Khái niệm ước chung lớn nhất:

Số nguyên d được gọi là ước chung của các số nguyên a_1, a_2,…, a_n, nếu nó là ước của tất cả các số đó.

Một ước chung d > 0 của các số nguyên a_1, a_2,…, a_n, trong đó mọi ước chung của a_1, a_2,…, a_n đều là ước của d, thì d được gọi là ước chung lớn nhất (ƯCLN) của a_1, a_2,…, a_n. Ký hiệu d = gcd(a_1,a_2,…, a_n) hay d = ƯCLN(a_1, a_2,…, a_n).

Khái niệm nguyên tố cùng nhau:

Nếu gcd(a_1,a_2,…, a_n) = 1, thì các số a_1, a_2,…, a_n được gọi là nguyên tố cùng nhau.

Ví dụ:

Cho a = 12, b = 15, gcd(12, 15) = 3.

Hai số 8 và 13 là nguyên tố cùng nhau vì gcd(8, 13) = 1.

Tính chất:

+ d = gcd(a_1, a_2,…, a_n) khi và chỉ khi tồn tại các số x_1, x_2,…, x_n sao cho:

  d = $a_1x_1+a_2x_2+\ldots+a_nx_n$.

  Đặc biệt: a_1, a_2,…, a_n nguyên tố cùng nhau $\Leftrightarrow$ tồn tại các số x_1, x_2,…, x_n sao cho: $1 = a_1x_1 + a_2x_2 + \ldots + a_nx_n$.

+ d = gcd(a_1, a_2,…, a_n) $\Leftrightarrow$ gcd(a_1/d, a_2/d,…, a_n/d) = 1.

+ gcd(m.a_1, m.a_2,…, m.a_n) = m * gcd(a_1, a_2,…, a_n) (với m $\neq$ 0).

+ Nếu b > 0, a = b.q +r thì gcd(a, b) = gcd(b, r).

3/. Thuật toán Euclide tìm ước chung lớn nhất

Bài toán:

* Dữ liệu vào: Cho hai số nguyên không âm a, b, a $\geq$ b.

* Kết quả: gcd(a, b).

Thuật toán: (Mô phỏng bằng ngôn ngữ Pascal).

```
Readln(a, b);
While b>0 do
    Begin
        r := a mod b;
        a := b;
        b := r;
    end;
Writeln(a);
```

Ví dụ:

a = 30; b = 18; gcd(30, 18) = gcd(18,12) = gcd(12, 6) = gcd(6, 0) = 6

Bảng 1: Mô tả các bước tính: gcd(30, 18)

a	b	r	a = b.q +r
30	18	12	30 = 18 * 1 + 12
18	12	6	18 = 12 * 1 + 6
12	6	0	12 = 6 * 2 + 0

1.1.1.2. Số nguyên tố

Khái niệm:

Số nguyên tố là số tự nhiên lớn hơn 1 và chỉ có hai ước là 1 và chính nó.

Ví dụ:

Các số 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37 là số nguyên tố. Số 2 là số nguyên tố chẵn duy nhất.

Tính chất:

+ Nếu p là số nguyên tố và p|a.b thì ta có a|p hoặc b|p hoặc cả a và b chia hết cho p.

+ Có vô số số nguyên tố.

1.1.1.3. Hàm ϕ Euler

Định nghĩa:

Cho n ≥ 1, đặt ϕ(n) là số các số nguyên trong khoảng [1, n] và nguyên tố cùng nhau với n. Hàm ϕ như thế được gọi là hàm phi-Euler.

Tính chất:

+ Nếu n là số nguyên tố thì ϕ(n) = n-1.

+ Nếu gcd(n, m) = 1, thì ϕ(n.m) = ϕ(n).ϕ(m).

+ Nếu $n = p_1^{e_1}.p_2^{e_2}...p_k^{e_k}$, là thừa số nguyên tố của n thì:

$$\phi(n) = n(1-\frac{1}{p_1})(1-\frac{1}{p_2})...(1-\frac{1}{p_k}).$$

1.1.1.4. Đồng dư thức

1/. Định nghĩa:

Nếu a và b là các số nguyên, a được gọi là đồng dư với b theo modulo n, được viết a ≡ b (mod n) nếu n chia hết (a - b). Số nguyên n được gọi là modulus của đồng dư.

2/.Ví dụ:

24 ≡ 9 (mod 5) vì 24 − 9 = 3*5.

−11 ≡ 17 (mod 7) vì −11 − 17 = −4*7.

3/. Một số tính chất của đồng dư thức:

Cho a, a_1, b, b_1, c $\in$ Z. Ta có các tính chất sau:

+ a ≡ b (mod n), nếu và chỉ nếu a và b có cùng số dư khi chia cho n. (1)

+ a ≡ a (mod n) *(tính phản xạ).* (2)

+ Nếu a ≡ b (mod n) thì b ≡ a (mod n) *(tính đối xứng).* (3)

+ Nếu a ≡ b (mod n) và b ≡ c (mod n) thì a ≡ c (mod n) *(tính bắc cầu).*(4)

+ Nếu a ≡ a_1 (mod n) và b ≡ b_1 (mod n) thì

a + b ≡ a_1 + b_1 (mod n) (5)

và a.b ≡ a_1b_1 (mod n).

Lớp tương đương của một số nguyên a là tập hợp các số nguyên đồng dư với a theo modulo n. Theo các tính chất (2), (3), (4) ta thấy: cho n cố định đồng dư với n trong không gian Z vào các lớp tương đương (phân hoạch). Nếu $a = qn + r$, trong đó $0 \le r < n$ thì $a \equiv r(\mathrm{mod}\, n)$. Vì vậy, mỗi số nguyên a là đồng dư theo modulo n với duy nhất một số nguyên trong tập hợp Z_n = {0, 1, 2,…, n-1} và được gọi là thặng dư nhỏ nhất theo modulo n. Cũng vì vậy, a và r cùng thuộc một lớp tương đương. Do đó, r có thể đơn giản được sử dụng để thể hiện lớp tương đương.

1.1.2. Các khái niệm trong đại số

1.1.2.1. Không gian Z_n

1/. Định nghĩa:

Các số nguyên theo modulo n, được ký hiệu là Z_n, là tập (lớp tương đương của) các số nguyên {0, 1, 2,..., n-1}. Tập Z_n có thể được coi là tập hợp tất cả các lớp tương đương theo modulo n. Trên tập Z_n xác định các phép cộng, trừ, nhân theo modulo n.

2/. Ví dụ:

Z_{25}= {0, 1, 2, …, 24}. Trong Z_{25}: 13 + 16 = 4, vì 13 + 16 = 29 ≡ 4 (mod 25). Tương tự: 13*16 = 8 trong Z_{25}.

3/. Các phép toán trong không gian modulo:

Cho n là các số nguyên dương. Như trước, các phần tử trong Z_n được thể hiện bởi các số nguyên {0, 1, 2,…, n-1}. Nhận xét rằng: nếu a, b $\in Z_n$ thì:

$$(a+b) \bmod n = \begin{cases} a+b \text{ nêú } a+b<n \\ a+b-n \text{ nêú } a+b \geq n \end{cases}$$

Vì vậy, phép cộng modulo (và phép trừ modulo) có thể được thực hiện mà không cần thực hiện các phép chia dài. Phép nhân modulo của a và b được thực hiện bằng phép nhân thông thường a với b như các số nguyên bình thường, sau đó lấy phần dư của kết quả sau khi chia cho n. Phép tính nghịch đảo trong Z_n có thể được thực hiện nhờ sử dụng thuật toán Euclidean mở rộng.

4/. Định lý phần dư China CRT

Giả sử các số nguyên $n_1, n_2, ..., n_k$ là các số nguyên tố cùng nhau từng cặp một thì hệ phương trình đồng dư:

$$\begin{cases} x_1 = a_1 (\bmod n_1) \\ x_2 = a_2 (\bmod n_2) \\ .... \\ x_k = a_k (\bmod n_k) \end{cases}$$ có nghiệm duy nhất theo modulo n. Với $n = n_1.n_2.....n_k$.

5/. Thuật toán Gausse

Nghiệm duy nhất có trong hệ phương trình đồng dư trong định lý phần dư China được cho bởi biểu thức:

$$x = \sum_{i=1}^{k} a_i N_i M_i \bmod n$$

Trong đó, $N_i = n/n_i$; $M_i = M_i = N_i^{-1} \bmod n$; (có M_i vì N_i và n_i nguyên tố với nhau).

**** Ví dụ:***

Cặp phương trình đồng dư $x \equiv 3 \pmod 7$ và $x \equiv 7 \pmod{13}$ có một nghiệm duy nhất $x \equiv 59 \pmod{91}$.

**** Tính chất:***

Nếu gcd(n_1, n_2) = 1 thì cặp đồng dư $x \equiv a \pmod{n_1}$ và $x \equiv a \pmod{n_2}$ có nghiệm duy nhất $x \equiv a \pmod{n_1 n_2}$.

6/. Phần tử nghịch đảo trong Z_n

* ***Định nghĩa:***

Cho $a \in Z_n$, nếu tồn tại $b \in Z_n$ sao cho $a\,b \equiv 1 \pmod n$, ta nói b là phần tử nghịch đảo của a trong Z_n và ký hiệu a^{-1}.

Một phần tử có phần tử nghịch đảo, gọi là khả nghịch.

**** Tính chất:***

+ Cho a, b $\in Z_n$, a/b (mod n) = $a.b^{-1}$ (mod n) được xác định khi và chỉ khi b là khả nghịch theo modulo n.

+ $a \in Z_n$, a là khả nghịch khi và chỉ khi gcd(a, n) = 1.

  Chứng minh:

  Nếu $a\,a^{-1} \equiv 1 \pmod n$ thì $a\,a^{-1} \equiv 1 + kn \leftrightarrow a\,a^{-1} - kn = 1 \rightarrow (a, n) = 1$.

  Nếu (a, n) = 1, ta có $a\,a^{-1} \equiv 1 + kn \rightarrow a\,a^{-1} + kn$, do đó $a\,a^{-1} \equiv 1 \pmod n$.

  Ví dụ: Các phần tử khả nghịch trong Z_9 là 1, 2, 4, 5, 7, và 8.

  $4^{-1} = 7$ vì $4*7 \equiv 1 \pmod 9$; $2^{-1} = 5$ vì $2*5 \equiv 1 \pmod 9$;

  $8^{-1} = 8$ vì $8*8 \equiv 1 \pmod 9$; $1^{-1} = 1$ vì $1*1 \equiv 1 \pmod 9$.

+ Cho d = gcd(a, n). Khi đó phương trình đồng dư có dạng $a.x \equiv b \pmod n$ sẽ có nghiệm x khi và chỉ khi d chia hết cho b.

* ***Tìm phần tử nghịch đảo bằng Thuật toán Euclid mở rộng***.

Bài toán:

+ Dữ liệu vào: $a \in Z_n$, n

+ Kết quả: Phần tử nghịch đảo của a

Thuật toán:

```
Procedure Invert(a, n);
Begin
    g0:=n; g1:=a; u0:=1; u1:=0; v0:=0;  v1:=1;
    i:=1;
    while gi ≠ 0 do
        begin
            y := gi-1 div gi;  gi+1 := gi+1 - y.gi;
            ui+1 := ui+1 - y.ui;  vi+1 := vi+1 - y.vi;
            i := i+1;
        end;
    t := vi+1;
    if t>0 then a-1 := t else a-1 := t + n;
End;
```

Ví dụ: Tìm phần tử nghịch đảo của 3 trong Z_7

Tức là phải giải phương trình $3x \equiv 1 \pmod 7$, x sẽ là phần tử nghịch đảo của 3.

I	**g_i**	**u_i**	**v_i**	**y**
1	7	1	0	
1	3	0	1	2
2	1	1	-2	3
3	0			

Vì $t = v_2 = -2 < 0$ do đó $x = x = a^{-1} := t + n = -2 + 7 = 5$.

Vậy 5 là phần tử nghịch đảo của 3 trong Z_7.

Chú ý:

Số mũ modulo có thể được tính một cách hiệu quả bằng thuật toán bình phương và nhân liên tiếp, nó được sử dụng chủ yếu trong nhiều giao thức mã hóa. Một phiên bản của thuật toán này như sau: Giả sử biểu diễn nhị phân của k là:

$$\sum_{i=0}^{1} k_i 2^i \quad \text{với} \quad k_i \in \{0,1\}$$

7/. Thuật toán bình phương liên tiếp để tính số mũ modulo trong Z_n

Bài toán :

+ Dữ liệu vào: $a \in Z_n$ và số nguyên dương $0 \le k < n$ trong đó k có biểu diễn nhị phân là:

$$k = \sum_{i=0}^{t} k_i 2^i$$

+ Kết quả: $a^k \bmod n$

Thuật toán:

```
Readln(a, n);
Begin
    b:=1;
    if k = 0 then writeln(b);
    A:=a;
    if k0 = 1 then b:=a;
    for i=1 to n
        begin
            A:=A*A mod n;
            if ki = 0 then b:= A*b mod n;
        end;
    writeln(b);
End;
```

Ví dụ: (Tính số mũ modulo)

Bảng 2: Mô tả các bước tính : $5^{596} \bmod 1234 = 1013$

I	0	1	2	3	4	5	6	7	8	9
k_i	0	0	1	0	1	0	1	0	0	1
A	5	25	625	681	1011	369	421	779	947	925
B	1	1	625	625	67	67	1059	1059	1059	1013

Độ phức tạp theo bit của các phép toán cơ bản trong Z_n được trình bày trong bảng sau:

Bảng 3: Độ phức tạp theo bit của các phép toán cơ bản trong Z

Phép toán	Độ phức tạp về bit
Cộng modulo $(a + b) \bmod n$	$O(\lg n)$
Trừ modulo $(a - b) \bmod n$	$O(\lg n)$
Nhân modulo $(a\ b) \bmod n$	$O((\lg n)^2)$
Nghịch đảo theo modulo $a^{-1} \bmod n$	$O((\lg n)^2)$
Số mũ modulo $a^k \bmod n, k < n$	$O((\lg n)^3)$

1.1.2.2. Nhóm nhân Z_n^*

1/. Định nghĩa:

Nhóm nhân (phép nhân) của tập Z_n kí hiệu là $\mathbf{Z}_n^* = \{a \in Z_n \mid \gcd(a, n) = 1\}$. Đặc biệt, nếu n là một số nguyên tố thì $\mathbf{Z}_n^* = \{a \mid 1 \le a \le n-1\}$.

2/. Định nghĩa cấp của $\mathbf{Z_n^*}$:

Cấp của $\mathbf{Z}_n^*$ được định nghĩa là số phần tử trong $\mathbf{Z}_n^*$, ($|\mathbf{Z}_n^*|$). Theo định nghĩa hàm phi-Euler ta có $|\mathbf{Z}_n^*| = \phi(n)$.

3/. Tính chất:

Cho $n \ge 2$ là số nguyên:

+ (*Định lý Euler*) Nếu $a \in \mathbf{Z}_n^*$ thì $a^{\phi(n)} \equiv 1 \pmod n$.
+ Nếu n là tích của các số nguyên tố phân biệt và nếu $r \equiv s (\bmod \phi(n))$ thì $a^r \equiv a^s \pmod n$ với mọi số nguyên a. Nói cách khác, làm việc với các số theo modulo nguyên tố p thì số mũ có thể giảm theo modulo $\phi(n)$.

Cho p là số nguyên tố:

+ (*Định lý Fermat*) Nếu $\gcd(a, p) = 1$ thì $a^{p-1} \equiv 1 \pmod p$.

+ Nếu $r \equiv s(\text{mod}\, p-1)$ thì $a^r \equiv a^s (\text{mod}\, p)$ với mọi số nguyên a. Nói cách khác, làm việc với các số theo modulo nguyên tố p thì số mũ có thể giảm theo modulo *p*-1.

+ $a^p \equiv a(\text{mod}\, p)$ với mọi số nguyên a.

1.1.2.3. Phần tử sinh

1/. Định nghĩa:

Cho $\alpha \in \mathbf{Z}_n^*$, nếu cấp của α là ϕ(n), khi đó α được gọi là phần tử sinh hay phần tử nguyên thủy của $\mathbf{Z}_n^*$, và nếu $\mathbf{Z}_n^*$ có một phần tử sinh, thì $\mathbf{Z}_n^*$ được gọi là nhóm cyclic. (chú ý nếu n là số nguyên tố thì ϕ(n) = n–1).

2/. Tính chất:

+ Nếu α là phần tử sinh của $\mathbf{Z}_n^*$, thì $\mathbf{Z}_n^* = \left\{\alpha^i \bmod n \mid 0 \le i \le \phi(n)-1\right\}$

+ Giả sử α một là phần tử sinh của $\mathbf{Z}_n^*$. Khi đó, $b = \alpha^i \, mod\, n$ cũng là một phần tử sinh của $\mathbf{Z}_n^*$ khi và chỉ khi gcd(i, $\phi(n)$) = *1*. Và sau đó nếu $\mathbf{Z}_n^*$ là nhóm *cyclic* thì số phần tử sinh sẽ là $\phi(\phi(n))$.

+ $\alpha \in \mathbf{Z}_n^*$ là phần tử sinh của $\mathbf{Z}_n^*$ khi và chỉ khi $\alpha^{\phi(n)/p} \not\equiv 1 \ (mod\ n)$ với mỗi số chia nguyên tố của $\phi(n)$.

+ $\mathbf{Z}_n^*$ có phần tử sinh khi và chỉ khi n = 2, 4, p^k hay $2p^k$ khi p là số nguyên tố lẻ và $k \ge 1$. Còn nếu p là số nguyên tố thì chắc chắn có phần tử sinh.

1.1.2.4. Thặng dư

1/. Định nghĩa:

Cho a $\in \mathbf{Z}_n^*$, a được gọi là thặng dư bậc hai theo modulo n hoặc bình phương theo modulo n, nếu tồn tại một x $\in \mathbf{Z}_n^*$, sao cho $x^2 \equiv a \ (\text{mod n})$, và nếu không tồn tại x như vậy thì a được gọi là bất thặng dư bậc hai theo modulo n. Tập các thăng dư bậc hai ký hiệu là Q_n và tập các bất thặng dư bậc hai ký hiệu là $\overline{Q}_n$.

Chú ý vì định nghĩa $0 \notin \mathbf{Z}_n^*$ nên $0 \notin Q_n$ và $0 \notin \overline{Q}_n$.

2/. Tính chất:

Cho n là tích của 2 số nguyên tố p và q. Khi đó, a $\notin \mathbf{Z}_n^*$ là một thặng dư bậc 2 theo modulo n khi và chỉ khi a$\notin$ Q_n và a $\notin \overline{Q}_n$. Ta có:

$$|Q_n| = |Q_p| \bullet |Q_q| = (p-1)(q-1)/4 \text{ và } |\overline{Q}_n| = 3(p-1)(q-1)/4.$$

3/. Ví dụ: Cho n = 21. Khi đó: $Q_{21} = \{1,4,16\}$ và $\overline{Q}_{21} = \{2,5,8,10,11,13,17,19,20\}$

1.1.3. Khái miệm độ phức tạp của thuật toán

1.1.3.1. Khái niệm thuật toán

"***Thuật toán***" được hiểu đơn giản là cách thức để giải một bài toán. Cũng có thể được hiểu bằng hai quan niệm: Trực giác hay hình thức như sau:

Quan niệm trực giác về "Thuật toán":

Một cách trực giác, thuật toán được hiểu là một dãy hữu hạn các quy tắc (chỉ thị, mệnh lệnh) mô tả một quá trình tính toán, để từ dữ liệu đã cho (Input) ta nhận được kết quả (Output) của bài toán.

Quan niệm toán học về "thuật toán":

Một cách hình thức, người ta quan niệm thuật toán là một máy Turing.

Phân loại: Thuật toán được chia thành hai loại: Đơn định và không đơn định.

Thuật toán đơn định (Deterministic): Là thuật toán mà kết quả của mọi phép toán đều được xác định duy nhất.

Thuật toán không đơn định (NonDeterministic): Là thuật toán có ít nhất một phép toán mà kết quả của nó là không duy nhất.

1.1.3.2. Khái niệm độ phức tạp của thuật toán

Chi phí của thuật toán: (tính theo một bộ dữ liệu vào)

Chi phí phải trả cho một quá trình tính toán gồm chi phí về thời gian và bộ nhớ. Chi phí thời gian của một quá trình tính toán là thời gian cần thiết để thực hiện một quá trình tính toán. Với thuật toán tựa Algol: Chi phí thời gian là số các phép tính cơ bản thực hiện trong quá trình tính toán.

Chi phí bộ nhớ của một quá trình tính toán là số ô nhớ cần thiết để thực hiện một quy trình tính toán.

Gọi A là một thuật toán, e là dữ liệu vào của bài toán đã được mã hóa. Thuật toán A tính trên dữ liệu vào e phải trả một giá trị nhất định. Ta kí hiệu:

$T_A(e)$ là giá thời gian và $I_A(e)$ là giá bộ nhớ.

Độ phức tạp về bộ nhớ: (trong trường hợp xấu nhất)

$L_A(n) = \max \{I_A(e), \text{với } |e| \leq n\}$, n là "kích thước" đầu vào của thuật toán.

Độ phức tạp thời gian: (trong trường hợp xấu nhất)

$T_A(n) = \max \{t_A(e), \text{với } |e| \leq n\}$.

Độ phức tạp tiệm cận:

Độ phức tạp PT(n) được gọi là tiệm cận tới hàm f(n), ký hiệu O(f(n)) nếu $\exists$ các số n_0, c mà $PT(n) \leq c.f(n)$, $\forall n \geq n_0$.

Độ phức tạp đa thức:

Độ phức tạp PT(n) được gọi đa thức , nếu có tiệm cận tới đa thức p(n).

Thuật toán đa thức:

Thuật toán được gọi là đa thức, nếu độ phức tạp về thời gian (trong trường hợp xấu nhất) của nó là đa thức.

Nói cách khác:

+ Thuật toán thời gian đa thức là thuật toán có độ phức tạp thời gian $\mathbf{O(n^t)}$, trong đó t là hằng số.

+ Thuật toán thời gian hàm mũ là thuật toán có độ phức tạp thời gian $\mathbf{O(t^{f(n)})}$, trong đó t là hằng số và f(n) là đa thức của n.

* Thời gian chạy của các lớp thuật toán khác nhau:

Độ phức tạp	Số phép tính ($n = 10^6$)	Thời gian (10^6 ptính/s)
O(1)	1	1 micro giây
O(n)	10^6	1 giây
$O(n^2)$	10^{12}	11,6 giây
$O(n^3)$	10^{18}	32000 năm
$O(2^n)$	10^{301030}	10^{301006} tuổi của vũ trụ

Chú ý: Có người cho rằng ngày nay máy tính với tốc độ rất lớn, không cần quan tâm nhiều tới thuật toán nhanh, chúng tôi xin dẫn một ví dụ đã được kiểm chứng.

1.1.3.3. Lớp bài toán P, NP và NP – complete

1/. Các khái niệm

**** Khái niệm "Dẫn về được"***

Bài toán B được gọi là "dẫn về được" bài toán A một cách đa thức, ký hiệu: $B \propto A$, nếu có thuật toán đơn định đa thức để giải bài toán A, thì cũng có thuật toán đơn định đa thức để giải bài toán B.

Nghĩa là: Bài toán A "khó hơn" bài toán B, hay B "dễ hơn" A, B được diễn đạt bằng ngôn ngữ của bài toán A, hay có thể hiểu B là trường hợp riêng của A. Vậy nếu giải được bài toán A thì cũng sẽ giải được bài toán B. Quan hệ $\propto$ có tính chất bắc cầu: Nếu $C \propto B$ và $B \propto A$ thì $C \propto A$.

**** Khái niệm "Khó tương đương"***

Bài toán A gọi là "khó tương đương" bài toán B, ký hiệu $A \equiv B$, nếu: $A \propto B$ và $B \propto A$

2/. Các lớp bài toán

Lớp bài toán P, NP:

Ký hiệu:

P là lớp bài toán giải được bằng thuật toán đơn định, đa thức (Polynomial).

NP là lớp bài toán giải được bằng thuật toán không đơn định, đa thức.

Theo định nghĩa ta có $P \subset NP$.

Hiện nay người ta chưa biết được $P \neq NP$

Lớp bài toán NP – Hard

Bài toán A được gọi là ***NP – Hard*** (NP - khó) nếu $\forall L \in NP$ đều là $L \propto A$. Lớp bài toán NP – Hard bao gồm tất cả những bài toán NP – Hard. Bài toán NP – Hard có thể nằm trong hoặc ngoài lớp NP.

Lớp bài toán NP – Complete

** Bài toán NP – Complete*

Bài toán A được gọi là NP - Complete (NP đầy đủ) nếu A là NP - Hard và $A \in NP$. Bài toán NP - Complete là bài toán NP - Hard nằm trong lớp NP. Lớp bài toán NP - Complete bao gồm tất cả những bài toán NP - Complete. Lớp NP - Complete là có thực, vì Cook và Karp đã chỉ ra bài toán đầu tiên thuộc lớp này, đó là bài toán "thỏa được": SATISFYABILITY.

** Chứng minh bài toán là NP – Hard*

Cách 1: Theo định nghĩa

Bài toán A được gọi là ***NP - hard*** (NP - khó) nếu $\forall L \in NP$ đều là $L \infty A$.

Chứng minh theo định nghĩa gặp nhiều khó khăn vì phải chứng minh:

Mọi bài toán trong NP đều "dễ hơn" A.

Theo cách 1, năm 1971 Cook và Karp đã chỉ ra bài toán đầu tiên thuộc lớp NP – hard, đó là bài toán "thỏa được " SATISFYABILITY.

Cách 2

Để chứng minh bài toán A là NP – hard, trong thực tế người ta thường dựa vào bài toán B nào đó đã được biết là NP – hard và chứng minh rằng $B \infty A$.

Theo tính chất bắc cầu của quan hệ "dẫn về được", A thỏa mãn định nghĩa NP - hard. Theo cách hiểu trực quan: B đã "khó" thì A càng "khó".

1.2. VẤN ĐỀ MÃ HÓA

Mã hóa đã được sử dụng từ thời xa xưa trong các hoạt động ngoại giao, chính trị và quân sự nhưng chỉ sau khi bài báo "Lý thuyết truyền tin trong các hệ thống bảo mật" của Claude Shannon ra đời thì mới thực sự trở thành một môn khoa học. Trước đó các vấn đề về mã hóa, mật mã gần như là một môn "nghệ thuật".

Mã hóa là phần rất quan trọng trong vấn đề bảo mật. Nhiệm vụ chính của mã hóa là làm cho tài liệu an toàn hơn, nó còn có một lợi ích quan trọng là: thay vì truyền đi tài liệu thô (không được mã hóa) trên một đường truyền đặc biệt, được canh phòng cẩn mật không cho người nào có thể truy nhập vào lấy dữ liệu, người ta có thể truyền một tài liệu đã được mã hóa trên bất cứ đường truyền nào mà không lo dữ liệu bị đánh cắp, vì nếu dữ liệu có bị đánh cắp, kẻ gian cũng không hiểu được.

1.2.1. Một số khái niệm

- Thuật toán mã hóa/giả mã: là thuật toán dùng để chuyển thông tin thành dữ liệu mã hóa hoặc ngược lại.

- Khóa: là thông tin mà thuật toán mã/giải mã sử dụng để mã/giải mã thông tin. Mỗi khi một thông tin được mã hóa thì chỉ có những người có khóa thích hợp mới có thể giải mã. Nếu không thì dùng cùng một thuật toán giải mã nhưng cũng không thể phục hồi lại thông tin ban đầu. Đây là đặc điểm quan trọng của khóa: mã hóa chỉ phụ thuộc vào khóa mà không phụ thuộc vào thuật toán mã/giải mã.

Với hình thức khá phổ biến hiện nay là truyền tin qua thư điện tử và không sử dụng các công cụ mã hóa, bảo mật cũng như chữ ký điện tử thì các tình huống sau có thể xảy ra:

- Không chỉ có người nhận mà người khác có thể đọc được thông tin.
- Thông tin mà ta nhận được có thể không phải của người gửi đúng đắn.
- Bị nghe trộm: thông tin được truyền đi trên đường truyền có thể bị ai đó "xâm nhập" vào lấy ra tuy nhiên vẫn đến được người nhận mà không bị thay đổi.
- Bị lấy cắp: Thông tin bị lấy ra hoàn toàn không đến được người nhận.
- Bị thay đổi: Thông tin bị chặn lại ở một nơi nào đó trên đường truyền và bị thay đổi. Sau đó thông tin đã bị thay đổi này được truyền tới cho người nhận như không có chuyện gì xảy ra.

Để giải quyết các vấn đề này, thông tin trước khi truyền đi sẽ được mã hóa và khi tới người nhận, nó sẽ được giải mã trở lại.

Để đảm bảo rằng chỉ người cần nhận có thể đọc được thông tin mà ta gửi khi biết rằng trên đường đi, nội dung thông tin có thể bị theo dõi và đọc trộm, người ta sử dụng các thuật toán đặc biệt để mã hóa thông tin. Trong trường hợp này, trước khi thông tin được gửi đi, chúng sẽ được mã hóa lại và kết quả là ta nhận được một nội dung thông tin "không có ý nghĩa". Khi thông điệp bị theo dõi hoặc bị bắt giữ trên đường đi, để hiểu được thông tin của thông điệp, kẻ tấn công phải làm một việc là giải mã nó. Thuật toán mã hóa càng tốt thì chi phí giải mã đối với kẻ tấn công càng cao. Khi chi phí giải mã lớn hơn giá trị thông tin thì coi như vấn đề bảo mật đã thành công.

Các thuật toán mã hóa thông tin khá đa dạng nhưng có thể chia ra làm hai loại mã hóa chính: mã hóa khóa đối xứng, và mã hóa khóa bất đối xứng.

1.2.2. Mã hóa khóa đối xứng

Mã hóa khóa đối xứng là hệ mã hóa mà biết được khóa lập mã thì có thể "dễ" tính được khóa giải mã và ngược lại.

1). Ưu điểm

Hệ mã hóa khóa đối xứng mã hóa và giải mã nhanh hơn hệ mã hóa khóa bất đối xứng.

2). Nhược điểm

+ Mã hóa khóa đối xứng chưa thật an toàn với lý do sau:

Người mã hóa và người giải mã phải có "chung" một khóa.

Khóa phải được giữ bí mật tuyệt đối, vì biết khóa này "dễ" xác định được khóa kia và ngược lại.

+ Vấn đề thỏa thuận khóa và quản lý khóa chung là khó khăn và phức tạp. Người gửi và người nhận phải luôn thống nhất với nhau về khóa. Việc thay đổi khóa là rất khó và dễ bị lộ. Khóa chung phải được gửi cho nhau trên kênh an toàn.

Mặt khác khi hai người (lập mã, giải mã) cùng biết "chung" một bí mật, thì càng khó giữ được bí mật.

3) Nơi sử dụng hệ mã hóa khóa đối xứng

Hệ mã hóa khóa đối xứng thường được sử dụng trong môi trường mà khóa chung có thể dễ dàng trao quyền bí mật, chẳng hạn trong cùng một mạng nội bộ. Hệ mã hóa khóa đối xứng thường dùng để mã hóa những bản tin lớn, vì tốc độ mã hóa và giải mã nhanh hơn hệ mã hóa khóa công khai.

Một số hệ mã hóa khóa đối xứng:

+ Hệ mã hóa khóa đối xứng - cổ điển: Hệ mã hóa dịch chuyển, hệ mã hóa thay thế, hệ mã hóa AFFINE, hệ mã hóa VIGENERE, hệ mã hóa hoán vị cục bộ, hệ mã hóa HILL.

+ Hệ mã hóa khóa đối xứng DES:

- DES: 56 bit, không an toàn. Có thể dễ dàng bẻ khóa trong khoảng vài phút.
- Triple DES, DESX, GDES, RDES: Mở rộng độ dài khóa ở mã DES lên 168 bit.

1.2.3. Mã hóa khóa bất đối xứng

Mã hóa bất đối xứng là hệ mã hóa có khóa lập mã và khóa giải mã khác nhau (ke ≠ kd), biết được khóa này cũng "khó" tính được khóa kia. Vì vậy chỉ cần bí mật khóa giải mã, còn công khai khóa lập mã. Do đó hệ mã hóa loại này còn có tên gọi là hệ mã hóa khóa công khai lập mã.

1). Ưu điểm

+ Hệ mã hóa công khai có ưu điểm chủ yếu sau: Thuật toán được viết một lần công khai cho nhiều lần dùng, cho nhiều người dùng, họ chỉ cần giữ bí mật khóa riêng của mình.

+ Khi biết các tham số ban đầu của hệ mã hóa, việc tính ra cặp khóa công khai và khóa bí mật phải là "dễ", tức là trong thời gian đa thức.

Người gửi bản rõ P và khóa công khai, thì "dễ" tạo ra bản mã C.

Người nhận bản mã C và khóa bí mật, thì "dễ" giải được thành bản rõ P.

+ Người mã hóa dùng khóa công khai, người giải mã giữ khóa bí mật. Khả năng lộ khóa bí mật khó hơn vì chỉ có một người giữ gìn.

Nếu thám mã biết khóa công khai, cố gắng tìm khóa bí mật, thì chúng phải đương đầu với bài toán "khó".

+ Nếu thám mã biết khóa công khai và bản mã C, thì việc tìm ra bản rõ P cũng là bài toán "khó", số phép trừ là vô cùng lớn, không khả thi.

2). Nhược điểm

Hệ mã hóa khóa công khai: mã hóa và giải mã chậm hơn hệ mã hóa đối xứng.

3). Nơi sử dụng hệ mã hóa khóa bất đối xứng

Hệ mã hóa khóa bất đối xứng thường được sử dụng chủ yếu trên các mạng công khai như Internet, khi mà việc trao chuyển khóa bí mật tương đối khó khăn.

Đặc trưng nổi bật của hệ mã hóa bất đối xứng là khóa công khai (public key) và bản mã (ciphertext) đều có thể gửi đi trên một kênh truyền tin không an toàn. Có biết cả khóa công khai và bản mã, thì thám mã cũng không dễ khám phá được bản rõ.

Nhưng vì có tốc độ mã hóa và giải mã chậm, nên hệ mã hóa công khai chỉ dùng để mã hóa những bản tin ngắn.

Hệ mã hóa khóa công khai thường được sử dụng cho cặp người dùng thỏa thuận khóa bí mật của hệ mã hóa khóa riêng.

Một số hệ mã hóa khóa bất đối xứng:

- RSA: Loại mã này được dùng nhiều nhất cho web và chương trình email. Độ dài khóa thông thường là từ 512 đến 1024 bit.
- Elgamal: 512 đến 1024 bit

1.3. VẤN ĐỀ CHỮ KÝ SỐ (digital signature)

1.3.1. Khái niệm

Nếu việc sử dụng mật mã trở nên phổ biến, không chỉ trong quân đội mà còn trong thương mại và những mục đích cá nhân thì những đoạn tin và tài liệu điện tử sẽ cần những chữ ký giống như các tài liệu giấy.

Cũng giống như trong thực tế, chữ ký để xác nhận cho người nhận rằng bức thư đó do người này gửi chứ không phải ai khác. Chữ ký điện tử sử dụng thuật toán mã không đối xứng để định danh người gửi. Thông thường, để bảo vệ các văn bản mã hóa người ta dùng chữ ký điện tử. Việc ứng dụng chữ ký điện tử cũng như công nhận giá trị pháp lý của nó là điều kiện tiên quyết trong thương mại điện tử. Nếu như việc giả mạo chữ ký viết tay hoặc dấu là không đơn giản thì việc làm giả một đoạn thông tin nào đó là dễ dàng. Vì lý do đó bạn không thể quét chữ ký của mình cũng như con dấu tròn của công ty để chứng tỏ rằng tài liệu mà bạn truyền đi đúng là của bạn.

Khi bạn cần "ký"một văn bản hoặc một tài liệu nào đó, thủ tục đầu tiên là tạo ra chữ ký và thêm nó vào trong thông điệp. Có thể hình dung thủ tục này như sau. Phần mềm mã hóa mà bạn sử dụng sẽ đọc nội dung văn bản và tạo ra một chuỗi thông tin đảm bảo chỉ đặc trưng cho văn bản đó mà thôi. Bất kì một thay đổi nào trong văn bản sẽ kéo theo sự thay đổi của chuỗi thông tin này. Sau đó phần mềm sẽ sử dụng khóa mật của bạn để mã hóa chuỗi thông tin này và thêm nó vào cuối văn bản như một động tác ký (bạn có thể thấy là chúng ta hoàn toàn không mã hóa nội dung văn bản, chỉ làm động tác ký mà thôi). Khi nhận được văn bản, người nhận lặp lại động tác tạo ra chuỗi thông tin đặc trưng, sau đó sử dụng mã công khai mà bạn đã gửi để kiểm tra chữ ký điện tử có đúng là của bạn không và nội dung thông điệp có bị thay đổi hay không. Thuật toán mã hóa không đối xứng đầu tiên và nổi tiếng hơn cả có tên gọi là RSA (được ghép từ chữ cái đầu tiên của tên ba tác giả là Rivest, Shamir, Adleman).

1.3.2. Quá trình tạo ra chữ ký điện tử

1) Tạo một câu ngắn gọn để nhận dạng – ví dụ như "Tôi là sinh viên".
2) Mã hóa nó bằng khóa bí mật của mình tạo ra chữ ký điện tử.
3) Gắn chữ ký điện tử vào thông điệp cần gửi rồi mã hóa toàn bộ bằng khóa công khai của người nhận.
4) Gửi thông điệp đi.
5) Người nhận sẽ dùng khóa bí mật của mình để giải mã thông điệp và lấy chữ ký ra sau đó họ sẽ giải mã chữ ký này bằng khóa công khai của người gửi. Chỉ người gửi nào có khóa bí mật phù hợp mới có thể tạo ra chữ ký mà người nhận giải mã thành công. Do đó người nhận có thể định danh người gửi.

1.3.3. Hàm băm sử dụng trong ký điện tử

Một thông điệp được đưa qua hàm băm sẽ tạo ra một giá trị có độ dài cố định và ngắn hơn được gọi là "đại diện" hay "bản tóm tắt". Mỗi một thông điệp đi qua một hàm băm chỉ có duy nhất một đại diện và ngược lại: rất khó để tìm được 2 thông điệp khác nhau nào có cùng một đại diện khi đi qua cùng một hàm băm.

Hàm băm thường kết hợp với chữ ký điện tử ở trên để tạo ra một loại chữ ký điện tử vừa an toàn hơn (không thể cắt/dán) vừa có thể dùng để kiểm tra tính toàn vẹn của thông điệp. Các bước để tạo ra chữ ký điện tử như sau:

1) Đưa thông điệp cần gửi qua hàm băm tạo ra đại diện cho thông điệp đó.
2) Mã hóa đại diện bằng khóa bí mật của người gửi để tạo ra chữ kí điện tử.
3) Mã hóa toàn bộ thông điệp và chữ ký bằng khóa công khai của người nhận và gửi đi.

Người nhận sẽ giải mã thông điệp bằng khóa bí mật của mình, giải mã chữ kí bằng khóa công khai của người gửi để lấy đại diện ra. Sau đó cho thông điệp qua hàm băm để tạo lại đại diện của thông điệp rồi so sánh với đại diện nhận được: nếu giống nhau thì người nhận có thể vừa định danh người gửi vừa kiểm tra tính toàn vẹn của thông điệp.

Một số hàm băm thường gặp:

- MD5 (Message Digest): 128 bit, nhanh, được sử dụng rộng rãi.
- SHA (Secure Hash Algorithm): 160 bit.

Chương 2. PHƯƠNG PHÁP CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

2.1. KHÁI NIỆM CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

2.1.1. Khái niệm chứng không tiết lộ thông tin (CM KTLTT)

1). Khái niệm

Nói một cách đơn giản, hệ thống chứng minh không tiết lộ thông tin cho phép một đối tượng thuyết phục một đối tượng khác tin vào một điều gì đó (chứng minh) mà vẫn không để lộ phương pháp chứng minh (không tiết lộ thông tin).

Xét một ví dụ đơn giản:

Giả sử P và V cùng tham gia trò chơi với các quân bài. P đưa ra 2 quân bài úp và nói đó là "**át**" và "**2**". P yêu cầu V chọn quân "át".

Trước khi chọn quân "át", V muốn kiểm tra chắc chắn rằng 2 quân bài đó đích thực là "**át**" và "**2**". V yêu cầu P chứng minh điều này. Nếu P lật 2 quần bài đó lên coi như là một cách chứng minh, thì trò chơi kết thúc, vì V đã nhìn thấy chúng và dĩ nhiên là anh ta có thể chọn ngay ra được quân bài "**át**".

Có một cách khác để P chứng minh rằng 2 quân bài đó là "**át**" và "**2**", mà không phải lật 2 quân bài đó lên, tức là không làm lộ thông tin về 2 quân bài trên tay P. Rất đơn giản, anh ta đưa 50 quân bài còn lại cho V. Nếu V kiểm tra thấy thiếu một quân bài "**át**" và một quân bài "**2**", thì có thể xem 2 quân bài trên tay P có đúng như anh ta nói.

Qua ví dụ trên có thể tạm hiểu "Chứng minh không tiết lộ thông tin" không có nghĩa là "không để lộ thông tin", mà có nghĩa là "để lộ thông tin ở mức ít nhất" về sự vật, sự việc cần chứng minh. Với những "thông tin để lộ", người xác minh không có đầy đủ hiểu biết (knowledge) về sự vật sự việc, họ chỉ thu được chút ít thông tin (coi như "zero knowledge") về đặc điểm tính chất của nó.

2). Giao thức $\sum$

Giao thức $\sum$ là giao thức "Hỏi - Đáp" 3 bước, để P chứng minh cho V một vấn đề nào đó.

- P gửi cho V: một giá trị ngẫu nhiên.

- V gửi lại P: một giá trị ngẫu nhiên như là giá trị dùng để kiểm thử.
- P gửi đáp lại V: một giá trị.

Kết quả V thừa nhận hoặc bác bỏ vấn đề P chứng minh.

"Chứng minh không tiết lộ thông tin" được phát minh bởi Goldwasser, Micali và Rackoff năm 1981 (được viết tắt là GMR). Chứng minh không tiết lộ thông tin (và chứng minh tương tác) là một trong những lý thuyết hay và có ảnh hưởng lớn trong khoa học máy tính.

3). Các thành phần trong phép chứng minh không tiết lộ thông tin

Có hai nhân vật mà chúng ta thường xuyên nhắc đến trong vấn đề này :

- Peggy Prover (người chứng minh): Peggy có thông tin muốn chứng minh cho Victor thấy, nhưng cô ấy lại không muốn nói thẳng bí mật đó cho Victor.

- Victor Verifier (người xác minh): Victor hỏi Peggy một loạt các câu hỏi, cố gắng tìm ra được là Peggy có thực sự biết được bí mật đó hay không. Victor không thu được điều gì từ bí mật đó, ngay cả khi anh ta gian lận hay không tuân theo chỉ dẫn của giao thức.

2.1.2. Khái niệm về chứng minh tương hỗ

1). Khái niệm

Trước tiên ta thảo luận ý tưởng về hệ thống chứng minh tương hỗ. Trong hệ thống chứng minh tương hỗ có hai thành viên: Lan và Nam. Lan là người chứng minh và Nam là người kiểm tra phép chứng minh. Lan biết một điều bí mật gì đó và cô ta muốn chứng minh cho Nam rằng cô ta biết điều đó. Phép chứng minh tương hỗ là một giao thức hỏi đáp, gồm một số vòng xác định.

Trong mỗi vòng, Lan và Nam luân phiên thực hiện các công việc sau:

- Nhận một thông báo từ nhóm khác.
- Thực hiện một tính toán riêng.
- Gửi một thông báo tới nhóm khác.

Một vòng của giao thức gồm một yêu cầu của Nam và một đáp ứng của Lan. Tới cuối phép chứng minh, Nam sẽ chấp nhận hoặc từ chối phép chứng minh của Lan tùy thuộc vào việc liệu Lan có đáp ứng thành công các yêu cầu của Nam hay không.

2). Tính chất

Phép chứng minh tương hỗ có:

- Tính **đầy đủ** khi và chỉ khi trong trường hợp Lan biết phép chứng minh x cho bài toán π, thì Nam luôn chấp nhận Lan.

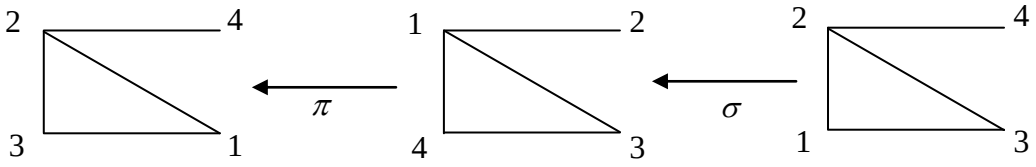
- Tính **đúng đắn** nghĩa là nếu Lan không biết cách chứng minh x cho bài toán π thì xác suất để Nam chấp nhận Lan là rất nhỏ.

Phép chứng minh tương hỗ có thể thực hiện được trong thời gian đa thức gọi là phép chứng minh tương hỗ trong thời gian đa thức.

3). Ví dụ

Minh họa hoạt động của giao thức tương hỗ để chứng minh sự đẳng cấu của hai đồ thị.

Giả sử $G_1 = \{V, E_1\}$ và $G_2 = \{V, E_2\}$ là hai đồ thị với tập đỉnh $V = \{1, 2, 3, 4\}$ và các tập cạnh $E_1 = \{12, 13, 14, 34\}$ và $E_2 = \{12, 13, 23, 24\}$. Giả sử Lan biết G_2 đẳng cấu với G_1 qua hoán vị $\sigma = \{4\ 1\ 3\ 2\}$.



Một vòng của giao thức có thể xảy ra như sau:

- Lan chọn ngẫu nhiên một hoán vị $\pi = \{2\ 4\ 1\ 3\}$ đồ thị H sẽ có tập cạnh $\{12, 13, 23, 24\}$ là ảnh của G_1 qua π, Lan gửi H cho Nam.
- Nam gửi i=2 cho Lan như một câu hỏi.
- Lan thử thấy hoán vị $\rho = \pi.\sigma = \{3\ 2\ 1\ 4\}$ ánh xạ G_2 thành H và gửi ρ cho Nam.
- Nam thử đúng H là ảnh của G_2 qua hoán vị ρ. Ta kết luận vòng hỏi đáp này đã thành công.

Toàn bộ giao thức gồm có $m = \log_2 n$ vòng.

2.2. HỆ THỐNG CM KTLTT CHO TÍNH ĐẲNG CẤU CỦA ĐỒ THỊ

2.2.1. Khái niệm đồ thị đẳng cấu

1). Khái niệm

Bài toán đồ thị đẳng cấu được mô tả dưới đây. Đây là một bài toán mà cho tới nay người ta chưa tìm ra thuật giải nào đó có thời gian đa thức cho bài toán, tuy nhiên nó không nằm trong lớp bài toán NP đầy đủ.

Định nghĩa đồ thị đẳng cấu:

Cho 2 đồ thị n đỉnh $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$, G_1 và G_2 được đẳng cấu nếu có một song ánh $p: V_1 \to V_2$ sao cho $\{u,v\} \in E_1$ khi và chỉ khi $\{p(u), p(v)\} \in E_2$.

2). Một sơ đồ chứng minh tương hỗ cho tính đẳng cấu của đồ thị

Sơ đồ nêu ra dưới đây nhằm thực hiện mục đích: Lan muốn thuyết phục Nam rằng hai đồ thị đã cho là đẳng cấu bằng một giao thức chứng minh tương hỗ, nhưng vào lúc kết thúc giao thức Nam vẫn không có chút thông tin nào về cách chứng minh (cho chính anh ta hoặc chứng minh cho người thứ 3) rằng hai đồ thị đó là đẳng cấu. Đây là một khái niệm rất khó định nghĩa hình thức, vì vậy ta sẽ xét một ví dụ trước khi định nghĩa

Hệ thống CMKTLTT hoàn thiện cho tính đẳng cấu của đồ thị:

Đầu vào:

Thông tin công khai: Hai đồ thị G_1 và G_2, mỗi đồ thị có tập đỉnh {1…n}.

Thông tin bí mật của Lan: Phép hoán vị σ đưa G_2 trở thành G_1.

Thực hiện:

Lặp lại các bước sau n lần:

- Lan chọn một phép hoán vị ngẫu nhiên π của {1…n} cô ta tính H là ảnh của G_1 theo π và gửi H cho Nam.
- Nam chọn một số nguyên ngẫu nhiên i = 1 hoặc 2 và gửi nó cho Lan.
- Lan tính một phép hoán vị ρ đưa H trở thành G_i. Lan sẽ gửi ρ cho Nam (nếu i=1 thì Lan sẽ xác định $\rho = \pi$ nếu i=2 thì Lan sẽ xác định ρ là $\sigma.\pi$ hợp của σ và π).
- Nam sẽ kiểm tra xem H có phải là ảnh của G_i theo ρ hay không.

Kết thúc:

Nam sẽ chỉ chấp nhận chứng minh của Lan, nếu H là ảnh của G_i ở mỗi một trong n vòng.

Ví dụ:

Giả sử $G_1 = (V, E_1)$ và $G_2 = (V, E_2)$ trong đó V = {1, 2, 3, 4}, E_1 = {12, 13, 14, 34} và E_2={12, 13, 23, 24}. Một phép đẳng cấu từ G_2 sang G_1 là hoán vị σ = (4, 1, 3, 2).

Bây giờ giả sử ở trong vòng nào đó của giao thức, Lan chọn hoán vị $\pi = (2,4,1,3)$. Khi đó H có tập cạnh {12, 13, 23, 24}.

Nếu yêu cầu của Nam là i=1 thì Lan sẽ cho Nam phép hoán vị π và Nam sẽ kiểm tra xem ảnh của G_1 theo π có phải là H không.

Nếu yêu cầu của Nam là i=2 thì Lan sẽ cho Nam phép hợp $\rho = \pi.\sigma$ = (3, 2, 1, 4) và Nam sẽ kiểm tra xem ảnh của G_2 theo ρ có phải là H không.

3). Tính chất

Dễ dàng kiểm tra được tính đầy đủ và tính đúng đắn của giao thức. Không khó khăn thấy rằng, xác suất để Nam chấp nhận sẽ bằng 1 nếu Lan biết phép chứng minh G_1 đẳng cấu với G_2. Ngược lại, nếu Lan không biết phép chứng minh thì chỉ có một cách để Lan lừa dối được Nam và cô ta phải giả định giá trị i mà Nam sẽ chọn ở mỗi vòng và truyền cho Nam một đồ thị ngẫu nhiên (đẳng cấu với G_i tương ứng). Xác suất để Lan giả định đúng các yêu cầu của Nam trong cả n vòng là 2^{-n}.

Tất cả các tính toán của Nam có thể thực hiện được trong thời gian đa thức vì tất cả các tính toán phải thực hiện là các phép sinh số ngẫu nhiên và các phép hoán vị. Ta cũng thấy rằng, các tính toán của Lan cũng tương tự như Nam (do đó có thể được thực hiện trong thời gian đa thức) nếu cô ta biết được sự tồn tại của phép hoán vị σ sao cho ảnh của G_2 theo σ là G_1.

Tại sao ta lại coi hệ thống chứng minh là hệ thống chứng minh không tiết lộ thông tin? Lý do là ở chỗ mặc dù Nam đã thuyết phục rằng G_1 là đẳng cấu với G_2 nhưng anh ta vẫn không thu thêm được tý kiến thức nào để giúp tìm được phép hoán vị σ đưa G_2 về G_1. Tất cả những điều mà Nam thấy trong mỗi vòng của phép chứng minh là một đồ thị ngẫu nhiên H đẳng cấu với các đồ thị G_1 và G_2 cùng với một phép hoán vị đưa G_1 thành H hoặc đưa G_2 thành H (nhưng không phải là cả hai). Tuy nhiên Nam có thể tự mình tính các bản sao ngẫu nhiên của các đồ thị này mà không cần tới sự giúp đỡ của Lan. Vì các đồ thị H được chọn một cách độc lập và ngẫu nhiên ở mỗi phần của phép chứng minh nên điều này không giúp đỡ được gì cho Nam trong việc tìm một phép đẳng cấu từ G_1 sang G_2.

Ta xem xét kĩ lưỡng thông tin mà Nam thu được nhờ tham gia vào hệ thống chứng minh tương hỗ:

- Các đồ thị G_1 và G_2.
- Tất cả các thông báo được Lan và Nam gửi đi.
- Các số ngẫu nhiên mà Nam dùng để tạo các yêu cầu của mình.

Bởi vậy, các thông tin T thu được qua sơ đồ chứng minh tương hỗ về phép đẳng cấu đồ thị sẽ có dạng sau:

$$T = ((G_1, G_2); (H_j, i_j, \rho_j)\ldots(H_n, i_n, \rho_n))$$

4). Giả mạo biên bản ghi nhận được sau giao thức chứng minh

Điểm mấu chốt (tạo cơ sở cho định nghĩa hình thức về phép chứng minh không tiết lộ thông tin) là Nam (hay bất kì người nào khác) có thể giả mạo các thông tin T (mà không cần phải tham gia vào hệ thống chứng minh tương hỗ) giống như các thông tin thực tế. Việc giả mạo được thực hiện theo thuật toán được mô tả như sau:

Thuật toán giả mạo chứng minh tương hỗ cho tính đẳng cấu:

Đầu vào:

Hai đồ thị G_1 và G_2, mỗi đồ thị có tập đỉnh {1…n}

Thuật toán:

T = (G_1, G_2)

For j=1 to n do

Chọn ngẫu nhiên i_j =1 hoặc 2

Chọn ρ_j là một hoán vị ngẫu nhiên của {1,…,n}

Tính H_j là ảnh của G_{i_j} theo ρ_j

Ghép (H_j, i_j, ρ_j) vào cuối của T.

Theo ngôn ngữ của phép chứng minh không tiết lộ thông tin, một thuật toán giả mạo thường được gọi là một bộ mô phỏng. Việc một bộ mô phỏng có thể tạo T có một hệ quả rất quan trọng. Bất kì kết quả nào mà Nam (hay bất kì ai khác) có thể tính từ T cũng có thể tính được từ một bản T giả mạo. Bởi vậy, việc tham gia vào hệ thống chứng minh sẽ không làm tăng khả năng tính toán của Nam. Đặc biệt là điều này không cho phép Nam tự chứng minh được rằng G_1 và G_2 là đẳng cấu. Hơn nữa, Nam cũng không thể thuyết phục được ai khác rằng G_1 và G_2 là đẳng cấu bằng cách chỉ cho họ một bản T, bởi vì không có cách nào để phân biệt một bản T hợp lệ với một bản T giả mạo.

2.2.2. Định nghĩa hệ thống CM KTLTT hoàn thiện

Trước hết, ta định nghĩa một cách chính xác về thông tin giả mạo và đưa ra một định nghĩa chặt chẽ theo thuật ngữ về các phân bố xác suất.

Giả sử ta có một phép chứng minh tương hỗ x cho bài toán π và một bộ mô phỏng thời gian đa thức S. Kí hiệu tập tất cả các thông tin T có thể tính từ x là F(x) (tập F này nhận được từ việc thực hiện phép chứng minh tương hỗ của Lan và Nam) và kí hiệu tập τ giả mạo có thể được tạo bởi S là τ(x). Với thông tin bất kì $T \in \tau(x)$, cho $p_\tau(T)$ là xác suất để T là thông tin giả mạo được tạo bởi S. Giả sử rằng τ(x) = F(x) và với bất kì $T \in \tau(x)$ nào, ta có $p_\tau(T) = p_F(T)$ (nói cách khác, tập các thông tin thực đồng nhất với tập các thông tin giả mạo và hai phân bố xác suất là như nhau). Khi đó ta định nghĩa hệ thống chứng minh tương hỗ là hệ thống chứng minh không tiết lộ thông tin hoàn thiện đối với Nam.

Dĩ nhiên là có thể định nghĩa đặc tính không tiết lộ thông tin theo kiểu mà ta thích. Tuy nhiên điều quan trọng là định nghĩa phải giữ nội dung cơ bản của đặc tính này. Ta coi rằng một hệ thống chứng minh tương hỗ là hệ không tiết lộ thông tin cho Nam nếu tồn tại một hệ mô phỏng tạo ra T có phân bố xác suất đồng nhất với phân bố xác suất của các thông tin được tạo ra khi Nam tham gia vào giao thức. Ta đã biết rằng T sẽ chứa tất cả các thông tin mà Nam thu lượm được nhờ tham gia vào giao thức. Bởi vậy, sẽ là hợp lý khi ta xem rằng bất cứ việc gì mà Nam có thể thực hiện được sau khi tham gia vào giao thức cũng chỉ như việc mà anh ta có thể thực hiện được nếu sử dụng hệ mô phỏng để tạo T giả mạo. Mặc dù ta không định nghĩa "thông tin" (hiểu biết) bằng cách tiếp cận này nhưng bất cứ điều gì được coi là thông tin thì Nam không thu lượm được tý nào.

Chứng minh: sơ đồ là hệ thống CMKTLTT hoàn thiện:

Bây giờ ta sẽ chứng tỏ rằng hệ thống chứng minh tương hỗ cho tính đẳng cấu đồ thị là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện đối với Nam.

Giả sử G_1 và G_2 là các đồ thị đẳng cấu có n đỉnh. Một bản T (thực hoặc giả mạo) sẽ gồm n bộ ba dạng (H, i, ρ) trong đó i=1 hoặc i=2, ρ là một phép hoán vị của {1…n} và H là ảnh của G_i theo hoán vị ρ. Ta gọi một bộ ba như vậy là một bộ ba hợp lệ và ký hiệu nó là R. Trước tiên ta sẽ tính |R| là số các bộ ba hợp lệ. Hiển nhiên là |R| = 2.n! vì mỗi phép chọn i và ρ sẽ xác định một đồ thị duy nhất H.

Ở mỗi vòng cho trước j bất kì của thuật toán giả mạo, rõ ràng là mỗi bộ ba hợp lệ (H, i, ρ) sẽ xuất hiện với xác suất như nhau bằng 1/(2.n!). Vậy xác suất để bộ hợp lệ (H, i, ρ) là bộ ba thứ j ở bản sao thực là gì? Trong hệ thống chứng minh tương hỗ, trước tiên Lan sẽ chọn một phép hoán vị ngẫu nhiên ρ nếu i=1, sau đó tính H là ảnh của G_1 theo ρ. Phép hoán vị ρ được xác định là ρ nếu i=1 và nó được xác định là hợp của hai phép hoán vị π và ρ nếu i=2.

Giả sử giá trị của i được chọn ngẫu nhiên bởi Nam. Nếu i=1 thì tất cả n! phép hoán vị ρ là đồng xác suất vì trong trường hợp này $\rho = \pi$ và π đã được chọn là một phép hoán vị ngẫu nhiên. Mặt khác, nếu i=2 thì $\rho = \pi \bullet \sigma$, trong đó π là ngẫu nhiên và σ cố định. Trong trường hợp này mỗi phép hoán vị có thể đều có xác suất bằng nhau. Xét thấy, vì cả hai trường hợp i=1 và i=2 đều có xác suất bằng nhau và mỗi phép hoán vị ρ đồng xác suất (không phụ thuộc vào giá trị của i) và bởi vì i và ρ cùng xác định H nên suy ra mọi bộ ba trong R chắc chắn sẽ đồng xác suất.

Vì thông tin gồm n bộ ba ngẫu nhiên độc lập ghép lại với nhau nên đối với mỗi bản sao có thể có T ta có:

$$p_\tau(T) = p_F(T) = \frac{1}{(2 * n!)^n}$$

Trường hợp có không kẻ trung thực:

Trong chứng minh trên đã giả thiết Nam tuân thủ giao thức khi anh ta tham gia vào hệ thống chứng minh tương hỗ. Tình hình sẽ phức tạp hơn nhiều nếu Nam không tuân theo giao thức. Phải chăng một phép chứng minh tương hỗ vẫn còn giữ được đặc tính không để lộ thông tin ngay cả khi Nam đi chệch khỏi giao thức.

Trong trường hợp ghép đẳng cấu đồ thị, cách duy nhất mà Nam có thể đi chệch khỏi giao thức chọn các yêu cầu i của mình theo cách không ngẫu nhiên. Về mặt trực giác, có vẻ như điều này không cung cấp cho Nam một chút "hiểu biết" nào. Tuy nhiên các bản sao được tạo bởi bộ mô phỏng sẽ không còn giống như các bản sao do Nam tạo ra nếu anh ta đi chệch khỏi giao thức. Ví dụ, giả sử Nam chọn i=1 trong mỗi vòng của phép chứng minh. Khi có một bản sao của phép chứng minh tương hỗ sẽ có $i_j = 1$ với $1 \le j \le n$, trong khi đó một bản sao được tạo bởi bộ mô phỏng sẽ có $i_j = 1$ với xác suất xuất hiện bằng 2^{-n}.

Điều khó khăn ở đây là phải chứng tỏ rằng cho dù Nam "không trung thực" đi chệch khỏi giao thức nhưng vẫn tồn tại một bộ mô phỏng với thời gian đa thức tạo ra các bản sao được tạo bởi Lan và Nam (không trung thực) trong phép chứng minh tương hỗ. Cũng như ở trên, câu "giống như" được hình thức hóa bằng cách nói rằng hai phân bố xác suất này đồng nhất.

2.2.3. Định nghĩa hệ thống CM KTLTT hoàn thiện không điều kiện

Giả sử rằng ta có một hệ thống chứng minh tương hỗ theo thời gian đa thức cho một bài toán quyết định cho trước π. Cho V^* là một thuật toán xác suất theo thời gian đa thức mà Nam (có thể không trung thực) sử dụng để tạo các yêu cầu của mình (tức là V^* biểu thị cho một người kiểm tra trung thực hoặc không trung thực). Ký hiệu tập tất cả các thông tin có thể (được tạo ra do kết quả của phép chứng minh tương hỗ mà Lan và V^* thực hiện với trường hợp Lan biết x của π) là $\tau(V^*, x)$. Giả sử rằng với mỗi V^* như vậy tồn tại một thuật toán xác suất theo thời gian đa thức $S^* = S^*(V^*)$ (bộ mô phỏng) tạo ra một bản sao giả mạo. Kí hiệu tập các bản sao giả mạo có thể bằng $F(V^*, x)$. Với một bản sao bất kỳ $T \in \tau(V^*, x)$ cho $p_F(T)$ là xác suất để T là thông tin do V^* tạo ra khi tham gia vào phép chứng minh tương hỗ. Tương tự, với $T \in F(x)$, cho $p_F(T)$ là xác suất để T là thông tin (giả mạo) được tạo bởi S^*. Giả sử rằng $F(V^*, x) = F(V^*, x)$ và với bất kỳ $T \in F(V^*, x)$, giả sử rằng $p_{F,V^*}(T) = p_{\tau,V^*}(T)$. Khi đó hệ thống chứng minh tương hỗ được gọi là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện không điều kiện.

Để chứng minh rằng hệ thống chứng minh là không tiết lộ thông tin hoàn thiện ta cần một phép biến đổi chung để xây dựng một bộ mô phỏng S^* từ V^* bất kỳ. Ta sẽ tiếp tục thực hiện việc này đối với hệ thống chứng minh cho tính đẳng cấu của đồ thị. Bộ mô phỏng sẽ đóng vai trò của Lan sử dụng V^* như một "chương trình con" có khả năng khởi tạo lại. Nói một cách không hình thức, S^* sẽ cố gắng giả định một yêu cầu i_j mà V^* sẽ đưa ra trong mỗi vòng j. Tức là S^* sẽ tạo ra một bộ ba hợp lệ ngẫu nhiên có dạng (H_j, i_j, ρ_j) và thực hiện thuật toán V^* để thấy được yêu cầu của nó dành cho vòng j. Nếu giả định i_j giống như yêu cầu i'_j (như được tạo bởi V^*) thì bộ ba (H_j, i_j, ρ_j) sẽ được gắn vào bản sao giả mạo. Nếu không thì bộ ba này sẽ bị loại bỏ. S^* sẽ giả định một yêu cầu mới bắt đầu của vòng hiện thời. Thuật ngữ "trạng thái" được hiểu là các giá trị của tất cả các biến dùng trong thuật toán.

Bây giờ ta sẽ đưa ra một mô tả chi tiết hơn về thuật toán mô phỏng S^*. Ở thời điểm bất kì cho trước, trong khi thực hiện chương trình V^*, trạng thái hiện thời của V^* sẽ được ký hiệu là state(V^*).

Thuật toán giả mạo cho V^* đối với các bản sao cho bài toán đồ thị đẳng cấu.

Đầu vào:

Hai đồ thị đẳng cấu G_1 và G_2, mỗi đồ thị có tập đỉnh {1….n}

Thuật toán:

T = (G_1, G_2)

For j = 1 to n do

Xác định trạng thái cũ bằng trạng thái (V^*)

Repeat

Chọn ngẫu nhiên i_j =1 hoặc 2

Chọn p_j là phép hoán vị ngẫu nhiên của {1…n}

Tính H_j là ảnh của G_{i_j} theo ρ_j

Gọi V^* với đầu vào H_j, ta thu được một yêu cầu i'_j

If $i_j = i'_j$ then

Ghép (H_j, i_j, ρ_j) vào cuối của T

Else

Thiết lập lại V^* bằng cách xác định trạng thái (V^*) = trạng thái cũ

Until $i_j = i'_j$

Có khả năng bộ mô phỏng sẽ không dừng lại nếu không xảy ra $i_j = i'_j$. Tuy nhiên có thể chứng tỏ rằng, thời gian chạy trung bình của bộ mô phỏng là thời gian đa thức và hai phân bố xác suất $p_{F,V^*}(T)$ và $p_{\tau,V^*}(T)$ là đồng nhất.

2.2.4. Định lý về hệ thống chứng minh tương hỗ cho đồ thị đẳng cấu

Phát biểu:

Hệ thống chứng minh tương hỗ cho tính đẳng cấu đồ thị là một hệ thống chứng minh không tiết lộ thông tin hoàn thiện.

Chứng minh:

Trước tiên ta thấy rằng bất luận V^* tạo các yêu cầu của nó ra sao, xác suất để giả định i'_j của S^* giống như yêu cầu i_j là bằng 1/2 . Như vậy trung bình S^* phải tạo được hai bộ ba, để tạo ra được một bộ ba gắn vào bản sao giả mạo. Do đó, thời gian chạy trung bình là thời gian đa thức theo n.

Nhiệm vụ khó khăn hơn là phải chứng tỏ rằng hai phân bố xác suất $p_{F,V^*}(T)$ và $p_{\tau,V^*}(T)$ là như nhau. Ở trên, ta đã tính được hai phân bố xác suất và thấy rằng chúng đồng nhất với trường hợp Nam là người kiểm tra trung thực. Ta cũng đã sử dụng một yếu tố là các bộ ba (H, i, ρ) được tạo ở các vòng khác nhau của phép chứng minh độc lập. Tuy nhiên trong bài toán này ta không có cách tính toán tường minh hai phân bố xác suất. Hơn nữa, các bộ ba được tạo ở các vòng khác nhau của phép chứng minh lại không độc lập. Ví dụ, yêu cầu mà V^* đưa ra ở vòng j có thể phụ thuộc theo một kiểu rất phức tạp nào đó vào các yêu cầu đó.

Cách khắc phục các khó khăn này là phải xem xét các phân bố xác suất trên các bản sao bộ phận có thể trong quá trình mô phỏng hoặc chứng minh tương hỗ và sau đó tiếp tục bằng phương pháp quy nạp trên số các vòng. Với $0 \le j \le n$, ta xác định các phân bố xác suất $p_{\tau,V^*,j}(T)$ và $p_{\tau,V^*,n}(T)$ trên tập các bản sao bộ phận T_j xuất hiện ở cuối vòng j. Chú ý rằng $p_{\tau,V^*,j}(T) = p_{\tau,V^*}(T)$ và $p_{\tau,V^*,n}(T) = p_{\tau,V^*}(T)$. Bởi vậy nếu có thể chứng tỏ rằng hai phâp bố $p_{\tau,V^*,j}(T)$ và $p_{\tau,V^*,j}(T)$ là đồng nhất với mọi j thì ta có điều cần chứng minh.

Trường hợp j = 0 ứng với khi bắt đầu thuật toán: lúc này bản sao chỉ gồm hai đồ thị G_1 và G_2. Bởi vậy các phân bố xác suất là đồng nhất khi j = 0. Ta sẽ sử dụng điều này để bắt đầu phép quy nạp.

Trước tiên giả sử hai phân bố xác suất $p_{\tau,V^*,j-1}(T)$ và $p_{\tau,V^*,j-1}(T)$ trên T_{j-1} là đồng nhất với giá trị $j \geq 1$ nào đó. Sau đó ta sẽ chứng tỏ rằng hai phân bố xác suất $p_{\tau,V^*,j}(T)$ và $p_{\tau,V^*,j}(T)$ trên τ_j đồng nhất.

Xét điều xảy ra trong vòng j của phép chứng minh tương hỗ. Xác suất để yêu cầu của V^* là $i_j = 1$ là một số thực p_j nào đó và xác suất để yêu cầu của V^* $i_j = 2$ là $1-p_j$, ở đây p_j phụ thuộc vào trạng thái của thuật toán V^* khi bắt đầu vòng j. Ở trên ta nhận xét rằng, trong phép chứng minh tương hỗ tất cả các đồ thị H có thể đều được Lan chọn với xác suất như nhau. Cũng vậy, một phép hoán vị ρ bất kỳ sẽ xuất hiện với xác suất như nhau (không phụ thuộc vào giá trị p_j), vì mọi phép hoán vị đều đồng khả năng đối với mỗi yêu cầu i_j có thể. Bởi vậy, xác suất để bộ ba thứ j ở trên bản sao (H, i, ρ) bằng p_j/n nếu i = 1 và bằng $(1-p_j)/n$ nếu i=2.

Tiếp theo ta sẽ thực hiện phân tích tương tự cho phép mô phỏng. Trong một bước lặp cho trước bất kỳ của vòng lặp REPEAT, S^* sẽ chọn một đồ thị H bất kỳ với xác suất 1/n!. Xác suất để i=1 và yêu cầu của V^* là 1 bằng $p_j/2$: xác suất để i=2 và yêu cầu của V^* là 2 bằng 1/2 sẽ không có gì được truyền đi trong lần lặp cho bất kì của vòng lặp REPEAT.

Trước hết sẽ xét trường hợp i=1. Như đã nêu ở trên, xác suất để yêu cầu của V^*=1 là p_j. Xác suất để một bộ ba (H, i, ρ) được coi là bộ ba thứ j trong bản sao ((H, i, ρ) được tiếp tục truyền đi) trong bước lặp thứ *l* của vòng lặp REPEAT bằng :

$$\frac{p_1}{2^l.n!}$$

Bởi vậy, xác suất để (H, i, ρ) là bộ ba thứ j trong bản sao là:

$$\frac{p_1}{2^l.n!}\left(1+\frac{1}{2}+\frac{1}{4}+\ldots\right)=\frac{p_1}{n!}$$

Trường hợp i=2 được phân tích theo cách tương tự: xác suất để (H, 2, ρ) được coi là bộ ba thứ j trong bản sao bằng $(1-p_1)/n!$

Như vậy hai phân bố xác suất trên các bản sao bộ phận tại cuối vòng j là đồng nhất. Theo quy nạp, hai phân bố xác suất $p_{\tau,V^*,j-1}(T)$ và $p_{\tau,V^*,j-1}(T)$ là như nhau. Định lý được chứng minh.

2.3. HỆ THỐNG CM KTLTT CHO BÀI TOÁN THẶNG DƯ BẬC HAI

2.3.1. Sơ đồ chứng minh

Bây giờ ta sẽ trình bày một số ví dụ khác về các hệ thống chứng minh không tiết lộ thông tin hoàn thiện. Một phép chứng minh không tiết lộ thông tin hoàn thiện cho các thặng dư bậc hai (modulo n = p.q, trong đó p và q là các yếu tố):

Chứng minh tương hỗ không tiết lộ thông tin hoàn thiện cho thặng dư bậc hai:

Đầu vào: Một số nguyên n có phân tích n = p.q không được biết, trong đó p và q là các số nguyên tố và $x \in QR(n)$

Thuật toán:

Lặp lại các bước sau $\log_2 n$ lần:

- Lan chọn một số ngẫu nhiên $v \in Z_n^*$ và tính $y = v^2 \bmod n$. Lan gửi y cho Nam.
- Nam chọn một số nguyên ngẫu nhiên i=0 hoặc 1 và gửi nó cho Lan.
- Lan tính $z = u^i v \bmod n$. Trong đó u là căn bậc 2 của x và gửi x cho Nam.
- Nam sẽ kiểm tra xem liệu có thỏa mãn $z^2 \equiv x^i y (\bmod n)$.
- Nam sẽ chấp nhận chứng minh của Lan nếu tính toán ở bước 5 được kiểm tra cho mỗi vòng (trong $\log_2 n$ vòng).

Lan đang phải chứng tỏ rằng x là một thặng dư bậc hai. Ở mỗi vòng có ta sẽ tạo một thặng dư bậc hai ngẫu nhiên y và gửi nó cho Nam. Sau đó, tùy thuộc vào yêu cầu của Nam, Lan sẽ đưa cho Nam căn bậc hai của y hoặc căn bậc hai của xy.

2.3.2. Tính chất của sơ đồ

Rõ ràng là giao thức này là đầy đủ. Để chứng minh tính đúng đắn ta thấy rằng, nếu x không phải là một thặng dư bậc 2 thì Lan chỉ có thể trả lời một trong hai yêu cầu có thể vì trong trường hợp này y là một thặng dư bậc hai khi và chỉ khi xy không phải là một thặng dư bậc hai. Bởi vậy Lan sẽ bị tóm ở một vòng cho trước bất kỳ của giao thức với xác suất 1/2 và xác suất để Lan đánh lừa được Nam trong toàn bộ n vòng chỉ bằng $2^{-\log_2 n}$ -1/n (lý do có $\log_2 n$ vòng là do cỡ đặc trưng của bài toán tỉ lệ với số bít trong biểu diễn nhị phân của người là $\log_2 n$). Bởi vậy xác suất đánh lừa của Lan sẽ là một hàm mũ âm của cỡ đặc trưng của bài toán giống như trong phép chứng minh không tiết lộ thông tin cho tính đẳng cấu đồ thị.

2.3.3. Chứng minh sơ đồ có tính đầy đủ

Có thể chỉ ra tính không tiết lộ thông tin hoàn thiện đối với Nam theo cách tương tự như bài toán đẳng cấu đồ thị. Nam có thể tạo ra bộ ba (y, i, z) bằng cách trước tiên chọn i và z và xác định: $y = z^2(x^i)^{-i} \bmod n$.

Các bộ ba được tạo theo cách này có cùng phân bố xác suất như các bộ ba được tạo trong giao thức với giả thiết Nam chọn các yêu cầu của mình một cách ngẫu nhiên. Tính không tiết lộ thông tin hoàn thiện (với V^* tùy ý) có thể được chứng minh theo phương pháp tương tự như đối với bài toán đẳng cấu đồ thị. Nó đòi hỏi phải xây dựng một bộ mô phỏng S^* để giả định các yêu cầu của V^* và chỉ giữ lại các bộ ba ứng với các giải định đúng.

Chương 3. ỨNG DỤNG CHỨNG MINH KHÔNG TIẾT LỘ THÔNG TIN

3.1. ỨNG DỤNG CM KTLTT TRONG BỎ PHIẾU ĐIỆN TỬ

Chúng ta đã biết một số kỹ thuật thăm dò ý kiến từ xa (các kỹ thuật này có trong bỏ phiếu điện tử - Electronic Voting). Cử tri giữ bí mật lá phiếu khi truyền từ xa tới ban kiểm phiếu bằng cách mã hoá nội dung lá phiếu. Theo kỹ thuật "mã hoá đồng cấu", ban kiểm phiếu có thể tính được kết quả thăm dò từ xa mà không cần phải giải mã nội dung lá phiếu. Vấn đề nảy sinh là cử tri phải chứng minh được với ban kiểm phiếu rằng **lá phiếu của mình là hợp lệ** nhưng **nội dung lá phiếu thì không được tiết lộ** với họ. Để thực hiện điều này, hiện nay người ta dùng kỹ thuật "Chứng minh không tiết lộ thông tin" (Zero-knowledge proof). Chúng tôi trình bày ý tưởng trên để thực hiện bỏ phiếu loại "Chọn 1 trong k".

3.1.1. Sơ đồ bỏ phiếu truyền thống

Trong lịch sử thế giới đã có rất nhiều cuộc bầu cử, những cuộc bầu cử giữ một vai trò quan trọng trong việc xác lập các thể chế chính trị của các quốc gia từ lớn đến nhỏ. Trong thế giới hiện đại, việc bỏ phiếu bầu quốc hội là một trong số những sự kiện quan trọng nhất của đất nước. Chính vì vậy, người ta đã bỏ rất nhiều công sức vào việc cải tiến các phương thức bầu cử, làm cho các cuộc bầu cử ngày càng trở nên "tốt" hơn. Phương thức bầu cử được thay đổi theo từng thời kì, theo sự tiến bộ của xã hội, nhưng tính chất của một cuộc bầu cử "tốt" thì không thay đổi đáng kể:

+ Tính chất 1: - Quyền được bỏ phiếu: Chỉ có những người có quyền bỏ phiếu mới được tham gia. Và mỗi người chỉ được bỏ phiếu không quá một lần. Cuộc bỏ phiếu cũng phải được thực hiện làm sao để những người có quyền bầu cử có điều kiện thuận lợi để thực hiện quyền của mình.

+ Tính chất 2: - Tính bí mật: Trong một cuộc bỏ phiếu, người bỏ phiếu có thể yên tâm là không ai có thể tìm ra được mình đã bỏ phiếu cho ai. Điều này để tránh việc trả thù những người bất đồng quan điểm.

+ Tính chất 3: - Kết quả chính xác: Mỗi cá nhân có quyền kiểm tra cuộc bầu cử và có khả năng phát hiện những sai phạm trong quá trình bầu cử so với thể lệ bầu cử đặt ra ban đầu. Thường những đối tượng được quyền kiểm tra bao gồm tất cả các cử tri và ban kiểm phiếu. Nếu có thể thì phải cung cấp phương pháp để giải quyết các sai phạm một cách hiệu quả.

Hiện tại thì đa số các cuộc bầu cử vẫn được thực hiện theo cách truyền thống. Tuy nhiên với tốc độ phát triển nhanh chóng của công nghệ thông tin, và đặc biệt là xu thế thực hiện "chính phủ điện tử" thì việc số hoá cuộc bầu cử để thay thế cho phương thức truyền thống là điều tất yếu sẽ phải diễn ra trong tương lai gần. Trong những năm gần đây, trên thế giới đã có một số nước thử nghiệm việc bỏ phiếu điện tử.

Sơ đồ bỏ phiếu truyền thống:

Khi bỏ phiếu theo phương thức truyền thống, ta mang giấy tờ cá nhân và lá phiếu chưa có nội dung gì đến bàn đóng dấu. Ở đó người ta sẽ kiểm tra giấy tờ để xác minh quyền bỏ phiếu, và đóng dấu xác thực lên lá phiếu. Sau đó ta vào phòng bỏ phiếu, cất giấy tờ đi, như vậy lá phiếu hoàn toàn không còn thông tin định danh. Công việc cuối cùng là điền vào một lá phiếu thông thường và bỏ vào hòm. Quá trình bỏ phiếu truyền thống này được coi là nặc danh nếu những người tham gia quá trình đều tuân thủ quy trình.

Từ sơ đồ bỏ phiếu truyền thống, việc bỏ phiếu có thể chia làm ba giai đoạn: Đăng kí, bỏ phiếu, kiểm phiếu.

3.1.2. Một số khái niệm

1). Vấn đề "bỏ phiếu điện tử" (Electronic Voting)

Nghiên cứu về "Bỏ phiếu thăm dò từ xa" là một chủ đề quan trọng đóng góp cho sự tiến bộ của xã hội dân chủ. Nếu một hệ thống bỏ phiếu thăm dò an toàn và tin cậy, nó sẽ được sử dụng thường xuyên để thu thập ý kiến của mọi người cho nhiều quyết định về chính trị và xã hội thông qua hệ thống tự động hóa. "Bỏ phiếu thăm dò từ xa" cũng phải đạt được các tính chất như "bỏ phiếu truyền thống" [7]. Một qui trình bỏ phiếu gồm một số giai đoạn (công đoạn). Hiện nay có nhiều kỹ thuật mật mã để thực hiện hợp lý trong từng giai đoạn.

Trong khóa luận này tôi xin trao đổi về giai đoạn ***Cử tri (CT) chuyển lá phiếu thăm dò tới Ban kiểm phiếu (Ban KP)*** cho sơ đồ bỏ phiếu loại "**Chọn 1 trong k**". Trong giai đoạn này người ta sử dụng kỹ thuật "Mã hóa đồng cấu - Chia sẻ bí mật" (Homomorphic Encryption – Secret Sharing) [8], kỹ thuật "Chứng minh không tiết lộ thông tin" (Zero-knowledge proof).

2). Giai đoạn cử tri chuyển lá phiếu tới ban kiểm phiếu

Theo suy nghĩ thông thường, khi Cử tri (CT) chuyển lá phiếu tới Ban kiểm phiếu (Ban KP) thì họ chỉ cần mã hóa nội dung lá phiếu là đủ. Vì tiếp theo Ban KP chỉ cần giải mã nội dung lá phiếu là tính được kết quả (kiểm phiếu).

Nhưng trên thực tế có thể xảy ra các tình huống sau:

- Ban KP hay một nhóm thành viên Ban KP không trung thực đã gian lận phiếu thăm dò, ví dụ sửa lại nội dung lá phiếu sau khi giải mã (trước khi kiểm phiếu). Để khắc phục tình hình này, người ta sử dụng kỹ thuật "Mã hóa đồng cấu - Chia sẻ bí mật". Với giải pháp này Ban KP không phải giải mã từng lá phiếu nhưng vẫn tính được kết quả.

- Để bảo đảm công khai kiểm phiếu, lá phiếu đã mã hóa khi tới Ban KP phải được niêm yết công khai. Như vậy nhìn trên bảng niêm yết này, Cử tri sẽ nhận ra lá phiếu của mình và họ có thể "bán" phiếu thăm dò. Để khắc phục tình trạng này, người ta dùng một "Người xác minh trung thực" (TT - honest verifier) làm trung gian giữa Cử tri và Ban KP. Cử tri gửi lá phiếu từ xa tới Ban KP thông qua người xác minh TT. Sau khi xác minh lá phiếu hợp lệ, anh ta làm "mù" lá phiếu (mã hóa lá phiếu lần thứ 2), tiếp đó gửi nó về Ban KP. Trên bảng niêm yết công khai, Cử tri không thể nhận ra lá phiếu của mình để có thể "bán" phiếu thăm dò.

Khi giải quyết 2 tình huống trên lại xuất hiện hai vấn đề khác:

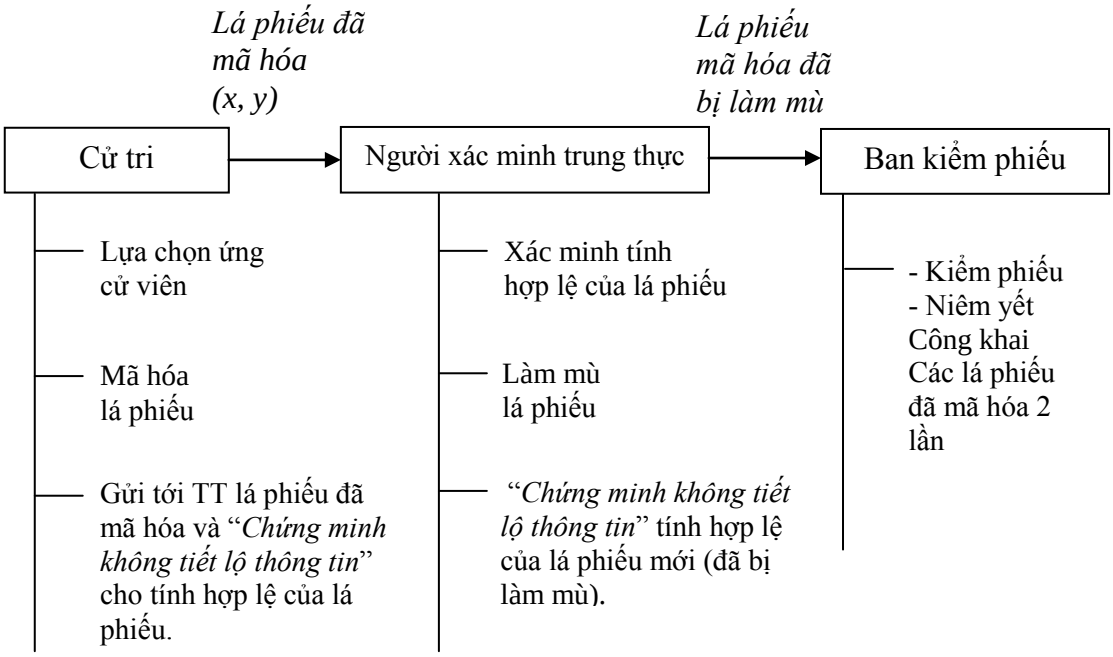
- Một là Cử tri phải chứng minh cho người xác minh TT biết lá phiếu của họ là hợp lệ, tức là nội dung lá phiếu chỉ ghi một trong số **k** lựa chọn (loại lựa chọn "chọn 1 trong **k**"), không cần phải chỉ rõ lá phiếu ghi rõ lựa chọn nào. Cách chứng minh như vậy gọi là "Chứng minh không tiết lộ thông tin". Với cách chứng minh này, nội dung lá phiếu không bị tiết lộ, trong khi mọi người đủ bằng chứng tin được rằng lá phiếu này là hợp lệ.

- Hai là người xác minh TT phải chứng minh cho Cử tri, Ban KP,…biết rằng lá phiếu bị làm "mù" vẫn hợp lệ (theo nghĩa trên) bằng cách chỉ ra rằng anh ta sở hữu giá trị β để là "mù" lá phiếu. Người xác minh TT chứng minh điều này cũng bằng phương pháp "Chứng minh không tiết lộ thông tin", tức là không cần phải tiết lộ chính giá trị β.

Sau đây là sơ đồ giai đoạn Cử tri chuyển lá phiếu tới Ban kiểm phiếu:

Giao thức 1: Cử tri mã hóa lá phiếu bằng hệ mã hóa Elgamal, Cử tri gửi nó tới người xác minh TT kèm theo "Chứng minh không tiết lộ thông tin" cho tính hợp lệ của lá phiếu đó.

Giao thức 2: Sau khi xác minh lá phiếu hợp lệ, người xác minh TT làm "mù" lá phiếu và gửi nó về Ban KP kèm theo "Chứng minh không tiết lộ thông tin" cho tính hợp lệ của lá phiếu đã bị làm "mù". Cụ thể chứng minh quyền sở hữu giá trị bí mật β dùng để làm "mù" lá phiếu.



Sơ đồ Cử chi chuyển lá phiếu đến Ban kiểm phiếu

3.1.3. Chứng minh tính hợp lệ của lá phiếu (x, y) *(Giao thức 1)*

Theo sơ đồ giai đoạn Cử tri (CT) chuyển lá phiếu tới Ban kiểm phiếu (Ban KP), phải thực hiện *giao thức 1*. Tức là Cử tri sẽ mã hóa lá phiếu bằng hệ mã hóa Elgamal, lá phiếu đã mã hóa được gửi tới người xác minh trung thực (TT) kèm theo "Chứng minh không tiết lộ thông tin" cho tính hợp lệ của lá phiếu đó.

Giả sử trong cuộc bầu cử "chọn 1 trong k", nếu cử tri nào đó chọn G_i là ứng cử viên thứ i trong danh sách thì ***lá phiếu hợp lệ*** phải ghi G_i với i = 1, 2,…, k. Bằng mã hóa Elgamal, lựa chọn G_i được mã hóa thành $(x, y) = (g^{\alpha}, h^{\alpha}G_i)$.

Như vậy Cử tri muốn chứng minh với người xác minh trung thực TT rằng lá phiếu (x, y) là hợp lệ, thì anh ta phải chỉ ra một trong k đẳng thức sau là đúng:

$$(\log_g x = \log_h (y/G_1)) \vee \ldots \vee (\log_g x = \log_h (y/G_k)). \qquad (1)$$

Để chứng minh (1) mà không bị lộ G_i, Cử tri và người xác minh TT thống nhất dùng giao thức "Chứng minh không tiết lộ thông tin" như sau:

Giai đoạn 1 cử tri chứng minh lá phiếu hợp lệ

Cử tri (CT)		**Người xác minh TT**
- Mã hóa lá phiếu [(x, y) = (g^{α}, $h^{\alpha}G_i$)] - Chọn ngẫu nhiên $w \in Z_p$ Tính $a_i = g^w$, $b_i = h^w$		
- Với j = 1,…, i - 1, i + 1,…, k Chọn $d_j, r_j \in Z_p$ (chưa chọn d_i, r_i) Tính $a_j = g^{r_i} x^{d_i}$, $b_j = h^{r_i} \left(y / G_j\right)^{d_i}$ - Đặt $(A,B) = (a_1,b_1),...,(a_k,b_k)$ (Sử dụng a_i, b_i đã tính ở trên)	$\xrightarrow{(x, y), (A, B)}$	
	$\xleftarrow{c}$	- TT chọn ngẫu nhiên $c \in Z_p$
- CT tính: (chưa chọn d_i, r_i) $d_i = c - \sum_{j \neq i} d_j$ $r_i = w - \alpha d_j$ $(D,R) = (d_1,r_1),...,(d_k,r_k)$	$\xrightarrow{(D, R)}$	
		- TT kiểm tra: $c = d_1 + ... + d_k$ Cho j = 1,…, k $a_j = g^{r_j} x^{d_j}$ $b_j = h^{r_j} \left(y / G_j\right)^{d_j}$ Nếu đều đúng TT kết luận: Lá phiếu hợp lệ.

Giải thích:

Bầu cử "Chọn 1 trong k"

Với k=1 ta có:

$$\left.\begin{array}{l} x = g^{\alpha} \Rightarrow \alpha = \log_g x \\ y = h^{\alpha}.G_1 \Rightarrow \alpha = \log_h (y / G_1) \end{array}\right\} \Rightarrow \log_g x = \log_h (y / G_1)$$

….

Với k=i

$$\left.\begin{array}{l} x = g^{\alpha} \Rightarrow \alpha = \log_g x \\ y = h^{\alpha}.G_i \Rightarrow \alpha = \log_h (y / G_i) \end{array}\right\} \Rightarrow \log_g x = \log_h (y / G_i)$$

….

Với k=k

$$\left.\begin{array}{l} x = g^{\alpha} \Rightarrow \alpha = \log_g x \\ y = h^{\alpha}.G_k \Rightarrow \alpha = \log_h (y / G_k) \end{array}\right\} \Rightarrow \log_g x = \log_h (y / G_k)$$

Ta có:lá phiếu (x, y)

$$\log_g x = \log_h (y / G_1) \vee ... \vee \log_g x = \log_h (y / G_i) \vee ... \vee \log_g x = \log_h (y / G_k)$$

Vì lá phiếu hợp lệ là lá phiếu đúng quy định (là chỉ chọn 1 ứng cử viên). Giả sử ứng cử viên được chọn là ứng cử viên thứ i$\Rightarrow$ để chứng minh lá phiếu hợp lệ ta chỉ cần chứng minh đẳng thức thứ i trong k đẳng thức trên là đúng:

$$\log_g x = \log_h (y / G_i)$$

Ví dụ: Chứng minh tính hợp lệ của lá phiếu đã mã hóa (x, y) = (g^{α}, $h^{\alpha}G_i$).

Giả sử cuộc bầu cử "chọn 1 trong 3". Các lựa chọn là 1 hoặc 2 hoặc 3. Kí hiệu lựa chọn ứng cử viên thứ i là G_i. Để chứng minh tính hợp lệ của lá phiếu, cử tri phải chứng minh:

$$\left(\log_g x = \log_h (y / G_i)\right) \vee \left(\log_g x = \log_h (y / G_2)\right) \vee \left(\log_g x = \log_h (y / G_3)\right) \qquad (1)$$

Để chứng minh (1), Cử tri và người xác thực TT thống nhất dùng giao thức "Chứng minh không tiết lộ thông tin" như sau:

Chọn phần tử sinh g = 3, α = 5, khóa bí mật s = 7, khóa công khai $h = g^s = 3^7$.

Ký hiệu 3 ứng cử viên $G_1 = 1$, $G_2 = 2$, $G_3 = 3$. Giả sử cử tri chọn $G_i = 2$.

Cử tri (CT)		Người xác minh TT
- Cử tri mã hóa lá phiếu [(x, y) = (3^5, $(3^7)^5$ .2)] - Chọn ngẫu nhiên w = 2 Tính $a_2 = 3^2$, $b_2 = (3^7)^2$		
- Với j = 1, 3: Chọn $d_1 = 8, r_1 = 9$ và tính: $a_1 = 3^9.\left(3^5\right)^8$, $b_1 = \left(3^7\right)^9 (\frac{(3^7)^5.2}{1})^8$ Chọn $d_1 = 10, r_1 = 11$ và tính: $a_3 = 3^{10}.\left(3^5\right)^{11}$, $b_3 = \left(3^7\right)^{11} (\frac{(3^7)^5.2}{3})^{10}$ (A, B)= $(3^9.(3^5)^8, (3^7)^5(\frac{(3^7)^5.2}{1})^8), (3^2, (3^7)^2),$ $(3^{11}.(3^5)^{10}, (3^7)^{11}(\frac{(3^7)^5.2}{3})^{10})$		
	$\xrightarrow{(x, y), (A, B)}$	
	$\xleftarrow{c}$	- TT chọn ngẫu nhiên c = 13
CT tính: $d_2 = c\text{-}\sum_{j \neq i} d_j$ $= c - (d_1 + d_3) = 13 - (8 + 10) = -5$ CT tính: $r_2 = \text{w-}\alpha.d_j$ $= 2\text{-}5.d_2 = 2\text{-}5.(-5) = 2 + 25 = 27$ CT đặt: (D, R) = (8, 9),(-5, 7),(10, 11)	$\xrightarrow{(D, R)}$	
		- TT kiểm tra: $c = d_1 + d_2 + d_3 = 8 + (-5) + 10 = 13$ Cho j = 1, 2, 3 $a_j = g^{r_j} x^{d_j}$; $b_j = h^{r_j} \left(y / G_j\right)^{d_j}$ $\Rightarrow$Kết luận: Lá phiếu hợp lệ.

3.1.4. Chứng minh quyền sở hữu giá trị bí mật β *(Giao thức 2)*

Theo sơ đồ giai đoạn Cử tri (CT) chuyển lá phiếu tới Ban kiểm phiếu (Ban KP) phải thực hiện *Giao thức 2*. Tức là sau khi xác minh lá phiếu của Cử tri là hợp lệ người xác minh TT làm "mù" lá phiếu và gửi nó về Ban KP kèm theo "Chứng minh không tiết lộ thông tin" cho tính hợp lệ của lá phiếu đã bị làm "mù".

Người xác minh TT làm "mù" lá phiếu thông qua cặp (u, v) và dựa trên giá trị ***bí mật*** β. Để chứng minh lá phiếu bị làm "mù" vẫn hợp lệ, người xác minh TT phải chứng minh được là anh ta sở hữu giá trị bí mật β thỏa mãn $u = g^{\beta}$, $v = h^{\beta}$. Nhưng mặt khác người xác minh TT không muốn để lộ β. Có một giao thức hiệu quả để người xác minh TT làm việc này: giao thức ∑. Trong sơ đồ sau đây, người xác minh TT là người chứng minh (P), người kiểm tra (V) là CT, Ban KP,..

Người xác minh TT (P)		**Người kiểm tra (V)**
- P có [(u, v)=$\left(g^{\beta}, h^{\beta}\right)$]		
- P chọn $w \in Z_p$		
-Tính $(a,b) = \left(g^{w}, h^{w}\right)$	$\xrightarrow{(a,\,b)}$ P gửi V giá trị ngẫu nhiên w thông qua (a, b)	
	$\xleftarrow{c}$ V gửi lại P giá trị ngẫu nhiên c	- V chọn $c \in Z_p$
- P tính r := $w + \beta c$	$\xrightarrow{r}$ P đáp lại V bằng r	
		- Kiểm tra: $g^r = au^c$; $h^r = bv^c$ Nếu đều đúng → V thừa nhận P sở hữu giá trị β

Giai đoạn 2 TT chứng minh lá phiếu làm mù là hợp lệ

Giải thích:

Ta có: $g^r = g^{w+\beta c} = g^w . g^{\beta c} = au^c$

$h^r = h^{w+\beta c} = h^w . h^{\beta c} = bv^c$

$\Rightarrow$ Nếu P không biết giá trị β thì P không thể tạo ra r chính xác để cho V kiểm tra.

Ví dụ:

Người chứng minh P chọn g = 3, s = 7, h = g^s = 3^7. Có β = 5 sử dụng (u, v)=$\left(g^\beta, h^\beta\right)$ = (3^5, $(3^7)^5$), cặp số này dùng để làm "mù" lá phiếu đã mã hóa của Cử tri.

P chứng minh với V rằng mình sở hữu β mà không muốn để lộ giá trị β. P thực hiện giao thức $\sum$ với người xác minh V như sau:

Người xác minh TT (P)		**Người kiểm tra (V)**
- P có [(u, v)=$\left(3^5, \left(3^7\right)^5\right)$] - P chọn w = 2. Tính (a, b) = $\left(g^w, h^w\right)$=$\left(3^2, \left(3^7\right)^2\right)$	$\xrightarrow{(a,\ b)}$	
	$\xleftarrow{c}$	- V chọn $c = 11$
- P tính r = $w + \beta c = 2 + 5*11 = 57$	$\xrightarrow{r}$	
		- Kiểm tra các giá trị của đẳng thức: $g^r = 3^{57} = 3^2.\left(3^5\right)^{11} = au^c$ $h^r = \left(3^7\right)^{57} = \left(3^7\right)^5 \left(\left(3^7\right)^2\right)^{11} = bv^c$ $\rightarrow$ V thừa nhận P sở hữu β = 5

Nếu có ai đó giả mạo rằng đã biết β để tạo (u, v) =$\left(g^\beta, h^\beta\right)$ thì "khó" có thể tính được r = w + βc, tức là bước kiểm thử $g^r = au^c$, $h^r = bv^c$ "khó" có thể thực hiện được. Vì vậy a, b, c, r, g, h, u, v đều công khai nên ai cũng có thể xác minh được r = w + βc. Nhờ giao thức trên mọi người tin rằng người xác minh TT đã dùng β để làm "mù" lá phiếu.

3.1.5. Giai đoạn cử tri chuyển lá phiếu đến ban kiểm phiếu *(phương án 2)*

Trong mục 2.3.3 khóa luận đã trình bày giai đoạn Cử tri (CT) chuyển lá phiếu tới Ban kiểm phiếu (Ban KP). Nó được thực hiện bằng ***Giao thức 1*** và ***Giao thức 2***, ta gọi là phương án 1. Có phương án khác (tạm gọi là 2) cũng để thực hiện giai đoạn này bằng 2 giao thức. Giao thức 1 giống trong phương án 1 và giao thức 2 có thay đổi như sau:

Sau khi người xác minh TT xác minh lá phiếu của Cử tri là hợp lệ, sau khi Cử tri xác minh người xác minh TT sở hữu giá trị β thì chính Cử tri làm "mù" lá phiếu và gửi nó về Ban KP (thay vì người xác minh TT làm "mù" lá phiếu và gửi nó về Ban KP như theo giao thức 2 của phương án 1). Trong phương án này chúng tôi đề nghị: mỗi lần xử lý một lá phiếu, tại mỗi bước thử điều kiện nếu không thoả mãn, công việc xử lý dừng lại với lá phiếu này để chuyển sang lá phiếu tiếp theo.

Phương án 1 gồm 2 giai đoạn một và hai

Cử tri (CT)		**Người xác minh TT**
- CT mã hoá lá phiếu $[(x,y)=(g^{\alpha},\ h^{\alpha}G_i)]$ - CT chọn ngẫu nhiên $w_1 \in Z_p$ Tính $a_i = g^{w_1}$ $\quad b_i = h^{w_1}$		
- Tính với $j = 1,..,i-1,i+1,...,k$ Chọn : $d_j,\ r_j \in Z_p$ Tính : $a_j = g^{r_j}x^{d_j}; b_j = h^{r_j}(y/G_j)^{d_j}$ - Đặt $(A,B)=(a_1,b_1),...,(a_k,b_k)$	$\xrightarrow{(x,y),(A,B)}$	
	$\xleftarrow{c_1}$	TT chọn ngẫu nhiên $c_1 \in Z_p$
- CT tính $d_i = c_1 - \sum_{j \neq i} d_j;\ r_i = w_1 - \alpha.d_i$ $(D,R)=(d_i,r_i),...,(d_k,r_k)$	$\xrightarrow{(D,R)}$	

		- TT kiểm tra: $c_1 = d_1 + ... + d_k$ $cho\ j = 1,...,k$ $a_j = g^{r_j} x^{d_j}$ $b_j = h^{r_j} (y / G_j)^{d_j}$ - Nếu điều kiện sai thì dừng giao thức và hủy bỏ lá phiếu. - Nếu đúng thì tiếp tục giao thức.
		- TT chọn β ngẫu nhiên bí mật và tính: $[(u,v) = (g^{\beta}, h^{\beta})]$
	$\xleftarrow{(a,b)}$ TT gửi CT giá trị w_2 thông qua (a, b)	- TT Chọn $w_2 \in Zp$, tính: $(a,b) := (g^{w_2}, h^{w_2})$
- CT chọn: $c_2 \in Z_q$	$\xrightarrow{c_2}$ CT gửi lại TT giá trị c_2	
	$\xleftarrow{r}$ TT đáp lại CT bằng r	- TT tính $r := w_2 + \beta c_2$
- CT kiểm tra: $g^r = a.u^{c_2}$; $h^r = b.v^{c_2}$ - Nếu điều kiện sai thì dừng giao thức và hủy bỏ lá phiếu. - Nếu điều kiện đúng thì CT mã hóa lại (làm "mù") lá phiếu nhờ cặp (u, v) của TT: $(x', y') = (xu, yv)$ Sau đó gửi về Ban KP.		

3.2. ỨNG DỤNG CM KTLTT TRONG SỬ DỤNG TIỀN ĐIỆN TỬ

3.2.1. Khái niệm thanh toán điện tử

Trong thương mại điện tử (TMĐT), khâu quan trọng nhất là việc thanh toán, bởi vì mục tiêu cuối cùng của cuộc trao đổi thương mại là người mua nhận được những cái gì cần mua và người bán nhận được số tiền thanh toán.

Thanh toán là một trong những vấn đề phức tạp nhất của TMĐT. Hoạt động TMĐT chỉ phát huy được tính ưu việt của nó khi áp dụng được hình thức thanh toán điện tử (TTĐT).

TTĐT là việc thanh toán tiền thông qua các thông điệp điện tử (Electronic message) thay cho việc thanh toán bằng tiền Sec hay tiền mặt. Bản chất của mô hình TTĐT cũng là mô phỏng lại mô hình thanh toán truyền thống, nhưng các thủ tục giao dịch, các thao tác xử lý dữ liệu, quá trình chuyển tiền… tất cả đều được thực hiện thông qua mạng máy tính, được nối bằng các giao thức chuyên dụng.

3.2.2. Khái niệm tiền điện tử

Tiền điện tử (E-money, E-currency, Internet money, Digital money, Digital currency, Digital cash) là thuật từ vẫn còn mơ hồ và chưa định nghĩa đầy đủ. Tuy nhiên có thể hiểu Tiền điện tử là loại tiền trao đổi theo phương pháp "điện tử", liên quan đến mạng máy tính và những hệ thống chứa giá trị ở dạng số (Digital stored value Systems).

Tiền điện tử cho phép người dùng có thể thanh toán khi mua hàng, hay vay mượn tiền, nhờ truyền đi các "dãy số" từ máy tính này (hay thiết bị lưu trữ này như Smart Card) tới máy tính khác (hay Smart Card khác).

Cũng như dãy số (Serial) trên tiền giấy, dãy số của tiền điện tử là duy nhất. Mỗi "đồng tiền điện tử" được phát hành bởi một tổ chức (ngân hàng) và biểu diễn một lượng tiền thật nào đó. Tiền điện tử có loại ẩn danh (anonymous identified e-money), định danh (identified e-money).

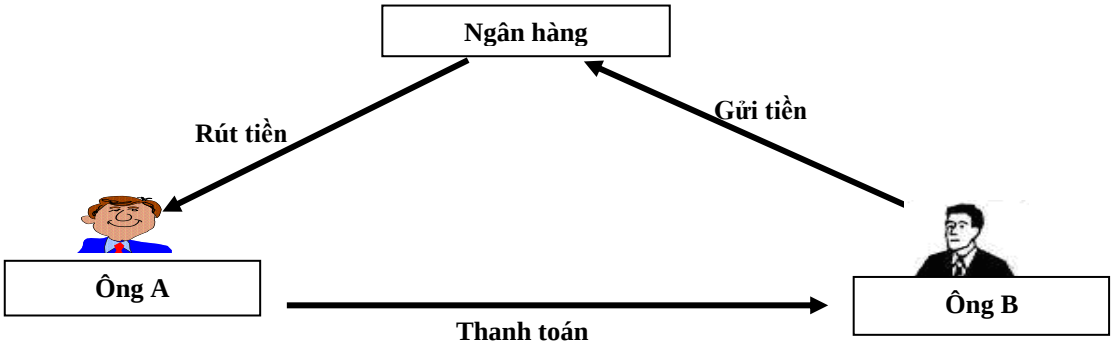
Tiền ẩn danh không tiết lộ thông tin định danh của người dùng. Tính ẩn danh của tiền điện tử tương tự như tiền mặt trên giấy. Tiền điện tử ẩn danh được rút từ một tài khoản, có thể được tiêu xài hay chuyển cho người khác mà không để lại dấu vết. Có nhiều loại tiền ẩn danh, có loại ẩn danh đối với người bán, nhưng không ẩn danh với ngân hàng. Có loại ẩn danh hoàn toàn, ẩn danh với tất cả mọi người.

Tiền điện tử định danh tiết lộ thông tin định danh của người dùng. Nó tương tự như thẻ tín dụng, cho phép ngân hàng lưu dấu vết của tiền khi luân chuyển.

Mỗi loại tiền trên lại chia thành 2 dạng: Trực tuyến (online), không trực tuyến (offline). Trực tuyến: là cần phải tương tác với bên thứ ba để kiểm soát giao dịch. Không trực tuyến: nghĩa là có thể kiểm soát được giao dịch, mà không cần liên quan trực tiếp đến bên thứ ba (ngân hàng).

Hiện nay có 2 hệ thống tiền điện tử chính: thẻ thông minh (Smart Card) hay phần mềm. Tuy nhiên chúng có chung các đặc điểm cơ bản sau: tính an toàn, tính riêng tư, tính độc lập, tính chuyển nhượng, tính phân chia.

3.2.3. Mô hình giao dịch mua bán bằng tiền điện tử



Mô hình giao dịch mua bán bằng tiền điện tử

Mô hình giao dịch mua bán bằng tiền điện tử có 3 giao dịch với 3 đối tượng: Ngân hàng, Người trả tiền A (mua hàng), Người được trả tiền B (bán hàng).

- Rút tiền: Ông A chuyển tiền của mình từ tài khoản ở ngân hàng vào 'Túi' của mình ('Túi' có thể là Smart Card hay máy tính).
- Thanh toán: Ông A chuyển tiền từ 'Túi' của mình đến 'Túi' ông B.
- Gửi tiền: Ông B chuyển tiền nhận được vào tài khoản của mình ở ngân hàng.

Mô hình có thể thực hiện bằng 2 cách:

+ Trực tuyến (online)**:** B liên lạc với ngân hàng để kiểm tra tính hợp lệ của đồng tiền trước khi thanh toán và phân phối hàng. Thanh toán và gửi tiền được tiến hành đồng thời. Thanh toán trực tuyến cần cho giao dịch có giá trị lớn. Hệ thống yêu cầu phải liên lạc với ngân hàng trong suốt mỗi lần giao dịch, vì thế chi phí nhiều hơn (tiền và thời gian).

+ Không trực tuyến (offline)**:** B liên lạc với ngân hàng để kiểm tra tính hợp lệ của đồng tiền được tiến hành sau quá trình thanh toán. Nó phù hợp cho những giao dịch có giá trị thấp.

*Các mô hình thanh toán điện tử***:**

Hệ thống TTĐT thực hiện thanh toán cho khách hàng theo một số cách, mà tiền mặt và séc thông thường không thể làm được. Hệ thống thanh toán cũng cung cấp khả năng thanh toán hàng hóa và dịch vụ qua thời gian, bằng cách cho phép người mua trả tiền ngay, trả tiền sau hay trả tiền trước.

- Mô hình trả tiền sau: Trong mô hình này, thời điểm tiền mặt được rút ra khỏi tài khoản bên mua để chuyển sang bên bán, xảy ra ngay (pay-now) hoặc sau (pay-later) giao dịch mua bán. Hoạt động của hệ thống dựa trên nguyên tắc Tín dụng (Credit crendental). Nó còn được gọi là mô hình mô phỏng Séc (Cheque - like model).

- Mô hình trả tiền trước: Trong mô hình này, khách hàng liên hệ với ngân hàng (hay công ty môi giới - Broker) để có được chứng từ do ngân hàng phát hành. Chứng từ hay Đồng tiền số này mang dấu ấn của ngân hàng, được đảm bảo bởi ngân hàng và do đó có thể dùng ở bất cứ nơi nào đã có xác lập hệ thống thanh toán với ngân hàng này.

Để đổi lấy chứng từ của ngân hàng, tài khoản của khách hàng bị triết khấu đi tương ứng với giá trị của chứng từ đó. Như vậy, khách hàng đã thực sự trả tiền trước khi sử dụng chứng từ này để mua hàng và thanh toán.

Chứng từ ở đây không phải do khách hàng tạo ra, không phải dành cho một cuộc mua bán cụ thể, mà do ngân hàng phát hành và có thể dùng vào mọi mục đích thanh toán. Vì có thể sử dụng giống như tiền mặt, do đó mô hình này còn được gọi là mô hình mô phỏng tiền mặt (Cash-like model).

Khi có người mua hàng tại cửa hàng và thanh toán bằng chứng từ như trên, cửa hàng sẽ kiểm tra tính hợp lệ của chúng, dựa trên những thông tin đặc biệt do ngân hàng tạo ra trên đó.

Cửa hàng có thể chọn một trong hai cách: Hoặc là liên hệ với ngân hàng để chuyển vào tài khoản của mình số tiền trước khi giao hàng (deposit-now), hoặc là chấp nhận và liên hệ chuyển tiền sau vào thời gian thích hợp (deposit-later).

Trường hợp riêng của mô hình mô phỏng tiền mặt là mô hình "tiền điện tử" (Electronic Cash).

Hiện nay hầu hết các dịch vụ mua bán hàng trên mạng đều sử dụng hình thức thanh toán bằng thẻ tín dụng (Credit card). Người dùng chỉ cần nhập các thông tin: tên người dùng, mã số thẻ, ngày hết hạn của thẻ.

Thực tế hiện nay tại châu Âu, các gian lận về thẻ trên Internet chiếm khoảng 7% tổng số các giao dịch thẻ, tỷ lệ này ở châu Á là khoảng 10%. Tại Việt Nam, dịch vụ thẻ tín dụng mới sử dụng cuối năm 1996, nhưng tỷ lệ các giao dịch gian lận trên tổng số các giao dịch là hơn 10%.

Trên thế giới hiện nay, nhu cầu về thương mại điện tử rất phổ biến, nhưng các vấn đề hạ tầng trong thanh toán điện tử vẫn chưa được giải quyết tương xứng và đáp ứng được các đòi hỏi đặt ra. Việc nghiên cứu xây dựng các hệ thống thanh toán điện tử để đảm bảo an toàn thông tin trong các dịch vụ thương mại điện tử là một hướng nghiên cứu cần thiết hiện nay.

Hệ thống thanh toán điện tử về mặt kỹ thuật chính là ứng dụng các thành tựu của lý thuyết mật mã. Mô hình thanh toán sử dụng giao thức mật mã để đảm bảo an toàn cho việc giao dịch giữa các bên tham gia.

3.2.4. Vấn đề "tiền điện tử"

1). Vấn đề ẩn danh người sử dụng đồng tiền

Ẩn danh là đặc tính quan trọng và tiện lợi của phương thức thanh toán bằng tiền nói chung. Tính ẩn danh được hiểu là người tiêu tiền phải được ẩn danh và không để lại dấu vết gì, nghĩa là ngân hàng không thể biết được: tiền giao dịch là của ai.

Đối với tiền điện tử, để giải quyết vấn đề trên, người ta đã sử dụng kỹ thuật "chữ ký mù". Đó là dạng đặc biệt của chữ ký điện tử, nó đòi hỏi người ký thực hiện ký vào thông điệp mà không biết nội dung của nó. Người ký sau này có thể nhìn thấy cặp chữ ký/thông điệp, nhưng không thể biết được là mình đã ký thông điệp đó khi nào và ở đâu, mặc dù anh ta có thể kiểm tra được chữ ký đó là đúng đắn. Nó cũng giống như người "ký" trên giấy khi đang nhắm mắt.

Với "chữ ký mù" của ngân hàng, họ không thể tìm được mối liên hệ nào giữa đồng tiền điện tử và chủ sở hữu của nó.

Lược đồ CHAUM-FIAT-NAOR dùng chữ ký mù RSA. Lược đồ BRAND dùng chữ ký mù Schnorr.

2). Vấn đề gian lận giá trị đồng tiền ("Khai man giá trị" đồng tiền)

Việc Ngân hàng dùng "chữ ký mù" để ký vào đồng tiền làm nảy sinh một vấn đề khác, đó là: Ông A gian lận, xin ngân hàng "ký" vào đồng tiền với giá trị 1$, nhưng thực tế lại gửi tới ngân hàng đồng tiền ghi giá trị 50$. Như vậy ông A đã có đồng tiền 50$ cùng với "chữ ký" của ngân hàng, nhưng tài khoản của ông chỉ bị khấu trừ 1$.

Vì ngân hàng "ký mù" lên đồng tiền, nên họ không thể biết được nội dung của nó là 1$ hay 50$. Để giải quyết trường hợp gian lận này, có hai cách.

Cách 1: Ngân hàng dùng bộ khóa (khoá ký, khóa kiểm tra chữ ký) khác nhau cho mỗi loại tiền. Nếu có n giá trị đồng tiền thì phải có n bộ khoá khác nhau. Ví dụ: với đồng tiền giá trị 1$ thì dùng khoá k_1, đồng tiền 50$ thì dùng khoá k_{50}. Nếu A gian lận tạo ra đồng tiền 50$ với khóa k_1, thì đó là đồng tiền không hợp lệ.

Cách 2: Dùng giao thức "Cắt và chọn" (Cut and choose). Ý tưởng như sau.

Để rút từ ngân hàng một đồng tiền giá trị T, ông A phải tạo k đồng tiền $C_1,C_2,...,C_k$ cùng giá trị T. Chúng đều được gắn định danh, khác nhau duy nhất giữa chúng là số sê-ri. A làm "mù" những đồng tiền này, và gửi chúng đến ngân hàng.

Ngân hàng yêu cầu ông A cung cấp thông tin để khử "mù" k-1 đồng tiền bất kỳ. Ngân hàng khử "mù" và kiểm tra chúng. Nếu tất cả đều hợp lệ, ngân hàng "ký mù" lên đồng tiền còn lại C_i (là đồng tiền mà ngân hàng không khử "mù"), và gửi cho A.

Ngân hàng có sự đảm bảo cao rằng đồng tiền còn lại C_i cũng là hợp lệ, vì nếu ông A gửi kèm đồng tiền không hợp pháp trong số k đồng tiền, thì xác suất bị phát hiện ít nhất là k-1/k. Xác suất này càng cao nếu k càng lớn. Tuy nhiên nếu k quá lớn thì hệ thống xử lý phải trao đổi nhiều dữ liệu và phải tính toán nhiều.

3). Vấn đề tiêu xài một đồng tiền nhiều lần (double - spending)

Tiền điện tử có dạng số hoá, nên dễ dàng tạo bản sao từ bản gốc. Chúng ta không thể phân biệt được giữa đồng tiền "gốc" và đồng tiền "sao". Kẻ gian có thể tiêu xài đồng tiền "sao" này nhiều lần mà không bị phát hiện.

Hệ thống tiền điện tử phải có khả năng ngăn ngừa hay phát hiện được trường hợp "Một đồng tiền tiêu xài nhiều lần" (double spending). Để giải quyết vấn đề này, đã có các giải pháp khác nhau tuỳ theo từng hệ thống tiền điện tử.

* Với hệ thống Tiền điện tử trực tuyến:

Ngân hàng lưu giữ thông tin tất cả những đồng tiền điện tử đã tiêu xài trước đó. Người bán hàng liên lạc tới ngân hàng, và họ có thể cho người bán hàng biết đồng tiền nào còn khả năng tiêu xài được.

Nếu ngân hàng báo rằng đồng tiền nào đó đã tiêu xài rồi, thì người bán hàng lập tức từ chối bán hàng. Điều này giống như cách mà người bán hàng hiện tại kiểm tra thẻ tín dụng tại những điểm bán hàng.

* Với hệ thống Tiền điện tử không trực tuyến: Phát hiện việc "tiêu xài nhiều lần" một đồng tiền, được thực hiện bằng hai cách.

Cách 1: Tạo thẻ thông minh (smart card) có chip "chống trộm cắp", nó còn được gọi là "người theo dõi". Chip lưu giữ lượng nhỏ dữ liệu của tất cả những phần tiền điện tử đã được tiêu xài qua Thẻ. Nếu người sở hữu Thẻ sao chép đồng tiền và tiêu xài nó lần hai, thì chip sẽ phát hiện được hành động này, và không cho phép giao dịch "tiêu xài". Bởi vì chip này dùng để chống sự gian dối, người sở hữu Thẻ không thể xoá được dữ liệu, trừ khi họ phá huỷ Thẻ.

Cách 2: Dựa vào cấu trúc của tiền điện tử và những giao thức mật mã để có thể truy vết tìm ra kẻ gian lận ("tiêu xài" nhiều lần).

Nếu người dùng biết rằng họ sẽ bị xử tội khi cố tính gian lận, về lý thuyết thì hành động gian lận sẽ giảm đi. Điều thuận lợi của phương pháp là không đòi hỏi những con chip đặc biệt. Hệ thống có thể được phát triển trên phần mềm (software) và có thể chạy trên máy tính cá nhân hay Smart card. Cách 2 có hai trường hợp:

Với Tiền điện tử Định danh - Không trực tuyến (Identified offline):

Dựa vào thông tin định danh để truy vết, tìm ra kẻ gian lận. Trong giao dịch, định danh của người dùng tiền được tích lũy đầy đủ trên đường đi của đồng tiền, và thông tin định danh của người dùng sẽ "trưởng thành" ở mỗi lần nó được "tiêu xài". Thông tin chi tiết mỗi lần giao dịch được gắn vào đồng tiền điện tử, và đi với nó, khi nó được chuyển từ người này sang người khác.

Khi đồng tiền chuyển tới ngân hàng, họ kiểm tra dữ liệu của nó, để xem đồng tiền này có bị "tiêu xài" hai lần không? Ngân hàng sử dụng những thông tin này để lần theo vết của những giao dịch, phát hiện ra người nào đã "tiêu xài" hai lần.

Với Tiền điện tử ẩn danh - Không trực tuyến (Anonymous Offline):

Trường hợp này phức tạp nhất vì đồng tiền ẩn danh, hơn thế lại ngoại tuyến. Hệ thống phải vừa đảm bảo tính ẩn danh của người sử dụng tiền, vừa đảm bảo có thể truy vết được định danh người dùng, trong trường hợp xảy ra vi phạm ("tiêu xài" hai lần).

Giải pháp là gắn thông tin "đã tiêu" lên đồng tiền ở mỗi lần giao dịch. Thông tin này sẽ "trưởng thành" ở mỗi lần giao dịch. Khi đồng tiền đến ngân hàng, họ sẽ kiểm tra trong cơ sở dữ liệu (CSDL) xem đồng tiền này đã được tiêu chưa. Nếu ngân hàng phát hiện tiền này đã được "tiêu xài" trước đây, thì họ sẽ dùng thông tin tích lũy để xác định định danh của kẻ gian lận ("tiêu xài" hai lần).

Thông tin tích luỹ trong trường hợp này chỉ có thể dùng để lần theo vết giao dịch nếu như đồng tiền đã "tiêu xài" hai lần, nghĩa là chỉ khi có gian lận thì ngân hàng mới có thể truy tìm được định danh của người tiêu tiền.

Nếu đồng tiền ẩn danh không bị "tiêu xài" hai lần, thì ngân hàng không thể xác định được định danh của người tiêu tiền, và cũng không thể xây dựng lại đường đi của đồng tiền. (Như vậy đồng tiền vẫn là ẩn danh).

3.2.5. Lược đồ tiền điện tử Brand

Lược đồ Brand được xây dựng dựa trên chữ ký số Schnorr và bài toán đại diện trong nhóm cấp nguyên tố. G_q là nhóm con cấp q của Z_p^*, trong đó p, q là số nguyên tố thoả mãn q|(p-1). Ngân hàng khởi tạo 5 thành phần: (g, h, g_1, g_2, d).

Trong đó:

- (g, h) $\in G_q$ (generator – tuple): khoá công khai của ngân hàng được dùng trong sơ đồ ký ở giao thức rút tiền, x là khoá bí mật của ngân hàng.

$$x = \log_g h \quad (h = g^x)$$

- (g_1, g_2): bộ phần tử sinh của G_q.

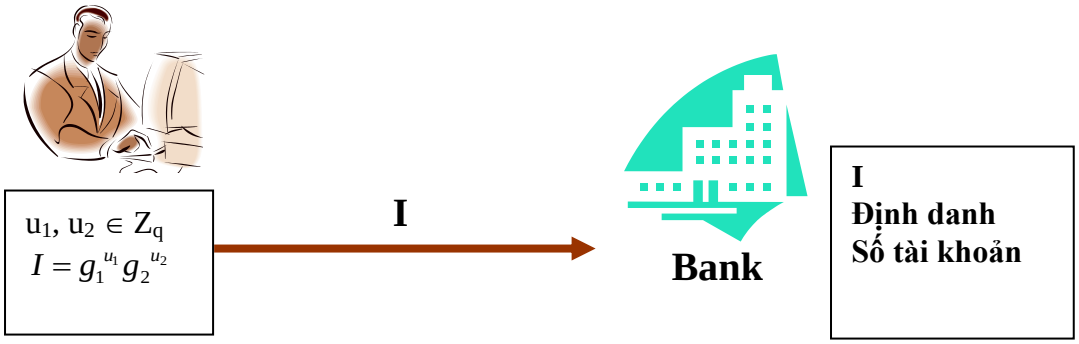
- Phần tử sinh giả d (khác g_1 và g_2), đảm bảo rằng định danh của người dùng sẽ không bị phát hiện trong giao thức thanh toán.

1). Khởi tạo tài khoản

- Alice tạo ngẫu nhiên u_1, $u_2 \in Z_q$, tính $I = g_1^{u_1} g_2^{u_2}$, chuyển I đến Ngân hàng.

- Ngân hàng lưu $I = g_1^{u_1} g_2^{u_2}$ cùng định danh của Alice và số tài khoản, nhưng ngân hàng không biết u_1 và u_2.

Trường hợp Alice tiêu xài đồng tiền hai lần, ngân hàng có thể tìm ra (u_1, u_2) và tính được I, từ I tìm ra định danh kẻ gian lận.



Quá trình khởi tạo tài khoản.

Chứng minh đại diện tài khoản:

Khi Alice rút tiền, đầu tiên phải xưng danh với ngân hàng, bằng cách chứng minh với ngân hàng là sẽ rút tiền trong tài khoản mà Alice sở hữu.

Phương pháp được dùng ở đây là "**chứng minh tri thức của một đại diện**". Alice phải chứng minh cho ngân hàng rằng: Alice biết u_1 và u_2 (vì Alice là chủ sở hữu tài khoản), nhưng không tiết lộ giá trị u_1, u_2 cho ngân hàng.

Quá trình xác thực được tiến hành như sau:

+ Alice chọn ngẫu nhiên $w_1, w_2 \in Z_q$ và gửi $y = g_1^{w_1} g_2^{w_2}$ đến ngân hàng.

+ Ngân hàng thử thách để kiểm tra có đúng Alice sở hữu tài khoản không, bằng cách chọn ngẫu nhiên $C_r \in Z_q$ và gửi đến Alice.

+ Alice tính $r_1 = w_1 + C_r u_1 \bmod q$, $r_2 = w_2 + C_r u_2 \bmod q$, gửi đến ngân hàng.

+ Ngân hàng chấp nhận xác thực là đúng khi và chỉ khi:

$$yI^{C_r} = g_1^{r_1} g_2^{r_2} \text{ trong đó } I = g_1^{u_1} g_2^{u_2}$$

Giải thích:

Ta có:

$$g_1^{r_1} g_2^{r_2} \equiv g_1^{w_1 + C_r u_1} g_2^{w_2 + C_r u_2} \equiv g_1^{w_1} g_2^{w_2} (g_1^{u_1} g_2^{u_2})^{C_r} = yI^{C_r}$$

$\Rightarrow$ Nếu Alice không biết u_1, u_2 (hay đại diện của I), thì Alice không thể tính tạo ra r_1, r_2 để cho ngân hàng kiểm tra.

Alice (người chứng minh)	*Ngân hàng (người kiểm tra)*
Biết u_1, u_2 là đại diện của $I = g_1^{u_1} g_2^{u_2}$	Chỉ biết I, g_1, g_2; không biết u_1, u_2
Tạo 2 số ngẫu nhiên $w_1, w_2 \in Z_q$	
Tính $y = g_1^{w_1} g_2^{w_2}$ gửi đến ngân hàng	Nhận y, chọn ngẫu nhiên $C_r \in Z_q$
	Gửi thử thách C_r đến Alice
Nhận C_r, tính: $r_1 = w_1 + C_r u_1 \bmod q,\ r_2 = w_2 + C_r u_2 \bmod q$ Và gửi chúng đến ngân hàng	Nhận r_1, r_2. Kiểm tra: $yI^{C_r} = g_1^{r_1} g_2^{r_2}$ Nếu thoả mãn, ngân hàng chấp nhận Alice biết đại diện của I (có nghĩa là biết u_1, u_2)

Quá trình chứng minh đại diện tài khoản.

2). Giao thức rút tiền.

Nếu xác thực được chấp nhận, thì quá trình rút tiền được tiến hành như sau:

+ Ngân hàng trừ một lượng tiền tương ứng từ tài khoản Alice. Ngân hàng và Alice cùng tính được m = Id (d là phần tử sinh và công khai). Ngân hàng gửi Alice: **z** = m^x, a = g^w, b = m^w (w được chọn ngẫu nhiên từ Z_q, x là khoá bí mật của ngân hàng).

+ Alice chọn 3 số ngẫu nhiên $s \in Z_q^*$; $u, v \in Z_q$ để làm "mù" m, z, a, b:

$$m' = m^s = (Id)^s = g_1^{u_1 s} g_1^{u_2 s} d^s$$

$$z' = z^s;\ a' = a^u g^v;\ b' = b^{su} m^{sv}$$

Tách ngẫu nhiên:

$$u_1 s = (x_1 + x_2) \bmod q,\ u_2 s = (y_1 + y_2) \bmod q$$

với $s = z_1 + z_2 \bmod q$ tính:

$$A = g_1^{x_1} g_2^{y_1} d^{z_1};\quad B = m' / A = g_1^{x_2} g_2^{y_2} d^{z_2}$$

+ Alice dùng hàm băm H tính c' = H (m', z', a', b', A).

Làm "mù" c' bằng $c = \frac{c'}{u} \bmod q$, gửi c đến ngân hàng.

+ Ngân hàng ký trên c được r = xc + w mod q, gửi r cho Alice, ghi có vào tài khoản của Alice. Alice chấp nhận nếu kiểm tra thấy $g^r = h^c a$ và $m^r = z^c b$ và tính r' = ru + v mod q. Lúc này, Alice có đồng tiền điện tử thật sự được đại diện bởi: A, B, Sign(A, B) với Sign(A, B) = (z', a', b', r') là chữ ký của ngân hàng. Nhưng làm thế nào chúng ta có thể biết được giá trị của từng đồng tiền. Có hai cách khác nhau để giải quyết vấn đề này:

Cách 1: Ngân hàng sử dụng một khoá công khai cho mỗi loại tiền. Nghĩa là, nếu có k đồng tiền khác biệt thì ngân hàng phải công khai khoá công khai k: (g_1, h_1) ... (g_k, h_k).

Cách 2: Chọn k phần tử sinh giả (dummy generator) khác nhau được công khai $d_1, \ldots, d_k$. Mỗi phần tử sinh được dùng để biểu hiện giá trị của mỗi đồng tiền.

Giải thích:

Ta có:

$$g^r \equiv g^{x.c+w} \equiv h^c g^w \equiv h^c a$$

$$m^r \equiv m^{x.c+w} \equiv m^c m^w \equiv z^c b$$

$\Rightarrow$ Nếu Ngân hàng không biết **x,** thì ngân hàng không thể tạo ra **r** để cho Alice kiểm tra.

Alice	Quá trình chứng minh đại diện	**Ngân hàng**
$I = g_1^{u_1} g_2^{u_2}$		x : khoá bí mật (g, h): khoá công khai ($h = g^x$)
	$\xleftarrow{z, a, b}$	$w \in Z_q$ m = Id, **z** = m^x, a = g^w, b = m^w
m = Id = $g_1^{u_1} g_2^{u_2} d$ $s \in Z_q^*$ $m' = m^s = (Id)^s = g_1^{u_1 s} g_1^{u_2 s} d^s$ $z' = z^s$ Tách ngẫu nhiên $u, v \in Z_q$ $a' = a^u g^v$ $b' = b^{su} m^{sv}$ c' = H (m', z', a', b', A). $c = \frac{c'}{u} \bmod q$	$\xrightarrow{c}$	
	$\xleftarrow{r}$	r = x**c** + w mod q
Kiểm tra: $g^r = h^c a$ và $m^r = z^c b$ Tính r' = ru + v mod q Đồng tiền: (A, B, Sign (A, B)) với Sign (A, B) = (z', a', b', r')		

Giao thức rút tiền.

3). Giao thức thanh toán.

Khi Alice muốn mua hàng hay sử dụng dịch vụ của Bob, trước tiên Alice cần phải gửi tiền cho Bob, quá trình thanh toán được thực hiện theo những bước sau:

+ Alice gửi tiền (A, B, Sign (A, B)) đến Bob.

$A = g_1^{x_1} g_2^{y_1} d^{z_1}$; $B = m' / A = g_1^{x_2} g_2^{y_2} d^{z_2}$

Sign (A, B) = (z', a', b', r')

+ Đầu tiên, Bob kiểm tra xem AB ≠ 1 hay không.

Nếu AB = 1, có nghĩa:

$$(g_1^{x_1} g_2^{y_1} d^{z_1})(g_1^{x_2} g_2^{y_2} d^{z_2}) = 1$$

$$\Rightarrow g_1^{x_1+x_2} g_2^{y_1+y_2} d^{z_1+z_2} = g_1^{u_1 s} g_2^{u_2 s} d^{s} = 1$$

$$\Rightarrow s = 0$$

Vậy, ngân hàng không xác định được u_1, u_2 trong trường hợp "double-spending". Sau đó, Bob kiểm tra chữ ký của ngân hàng sign(A,B) có hợp lệ không. Nếu đúng, Bob thử thách Alice bằng cách gửi $c \in Z_q^*$, c không cần thiết là số ngẫu nhiên, nhưng phải đảm bảo là duy nhất trong mỗi lần thanh toán.

Bob tính c như sau:

c = H_0 (A, B, I_b, date/time), với I là định danh của Bob, date/time là nhãn thời gian của giao dịch, H_0 là hàm băm.

+ Alice phản hồi với:

$r_1 = x_1 + cx_2 \mod q$

$r_2 = y_1 + cy_2 \mod q$

$r_3 = z_1 + cz_2 \mod q$

+ Bob kiểm tra, nếu $g_1^{r_1} g_2^{r_2} d^{r_3} = AB^c$ thì chấp nhận thanh toán

Giải thích:

Ta có:

$$g_1^{r_1} g_2^{r_2} d^{r_3} \equiv g_1^{x_1+cx_2} g_2^{y_1+cy_2} d^{z_1+cz_2} \equiv (g_1^{x_1} g_2^{y_1} d^{z_1})(g_1^{x_2} g_2^{y_2} d^{z_2})^c \equiv AB^c$$

$\Rightarrow$ Nếu Alice không biết $x_1, y_1, z_1, x_2, y_2, z_2$, thì Alice không thể tạo ra r_1, r_2, r_3 để cho Bob kiểm tra.

Alice		Bob
	$A,B\ sig(A,B)$ →	
		AB ≠ 1? Kiểm tra sign (A, B) $c \in Z_q^*$
	← c	
$r_1 = x_1 + cx_2 \mod q$ $r_2 = y_1 + cy_2 \mod q$ $r_3 = z_1 + cz_2 \mod q$		
	r_1, r_2, r_3 →	
		$g_1^{r_1} g_2^{r_2} d^{r_3} = AB^c$? Nếu đúng Bob chấp nhận thanh toán.

Giao thức thanh toán.

4). Giao thức Gửi.

+ Bob gửi thông tin thanh toán (A, B, Sign (A, B)), c, r_1, r_2 và r_3 đến ngân hàng.

+ Ngân hàng kiểm tra chữ ký có chính xác không và đồng tiền không được tiêu xài trước đó.

+ Bob thử thách Alice bằng giá trị c = H_0 (A, B, I_b, date/time)

+ Alice trả lời lại giá trị r_1, r_2, r_3.

+ Nếu tất cả đều thoả mãn, Ngân hàng gửi tiền vào tài khoản của Bob.

Chương 4. THỬ NGHIỆM CHƯƠNG TRÌNH

4.1. MÔ TẢ CHƯƠNG TRÌNH

4.1.1. Giới thiệu

Chương trình mô phỏng *giao thức 1*: chứng minh tính hợp lệ của lá phiếu và *giao thức 2*: chứng minh quyền sở hữu giá trị bí mật β, trong ứng dụng CM KTLTT trong bỏ phiếu điện tử, được viết bằng ngôn ngữ lập trình Turbo C.

1). Cấu hình của hệ thống

* Phần cứng (cấu hình tối thiểu):

Bộ nhớ ổ cứng: 20gb

Bộ nhớ ram: 128 mb

Tốc độ máy tối thiểu: 1 GHz

* Phần mềm:

Hệ điều hành:Linux, Window,…

Ngôn ngữ lập trình: Turbo C

2). Các thành phần của chương trình

* Chương trình chứng minh tính hợp lệ của lá phiếu mô phỏng *giao thức 1:*

- Nhập các tham số đầu vào để mã hóa lá phiếu.
- Tính toán các tham số trung gian.
- Kiểm tra tính hợp lệ của lá phiếu.

* Chương trình chứng minh quyền sở hữu giá trị bí mật β mô phỏng *giao thức 2:*

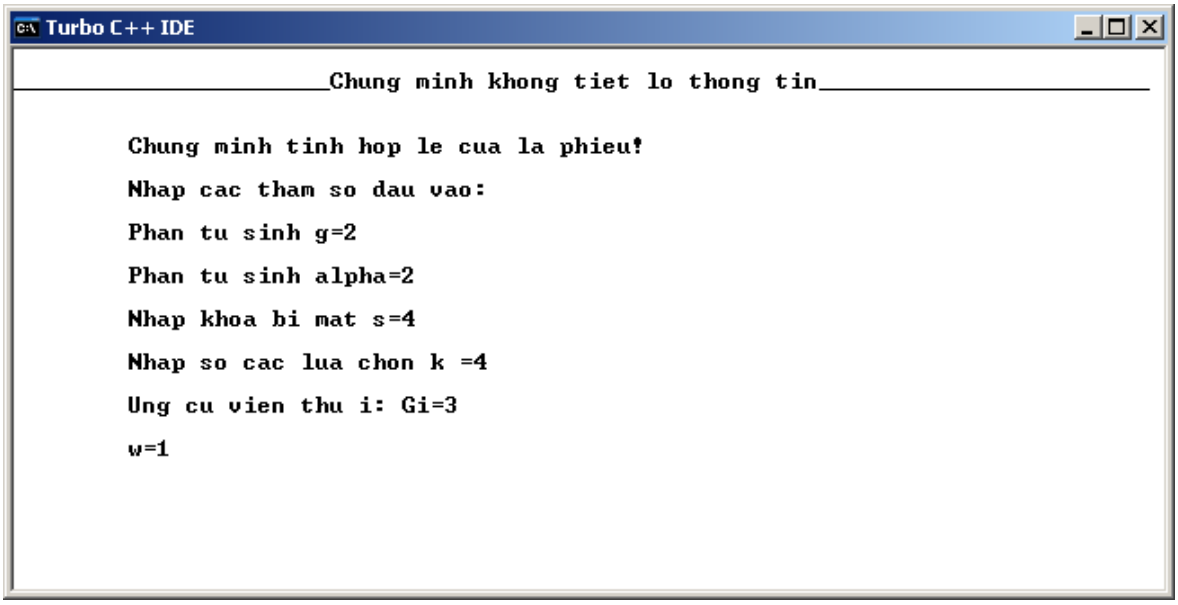
- Nhập các tham số dầu vào.
- Tính toán các tham số trung gian.
- Kiểm tra người xác minh TT có giữ giá trị bí mật β không.

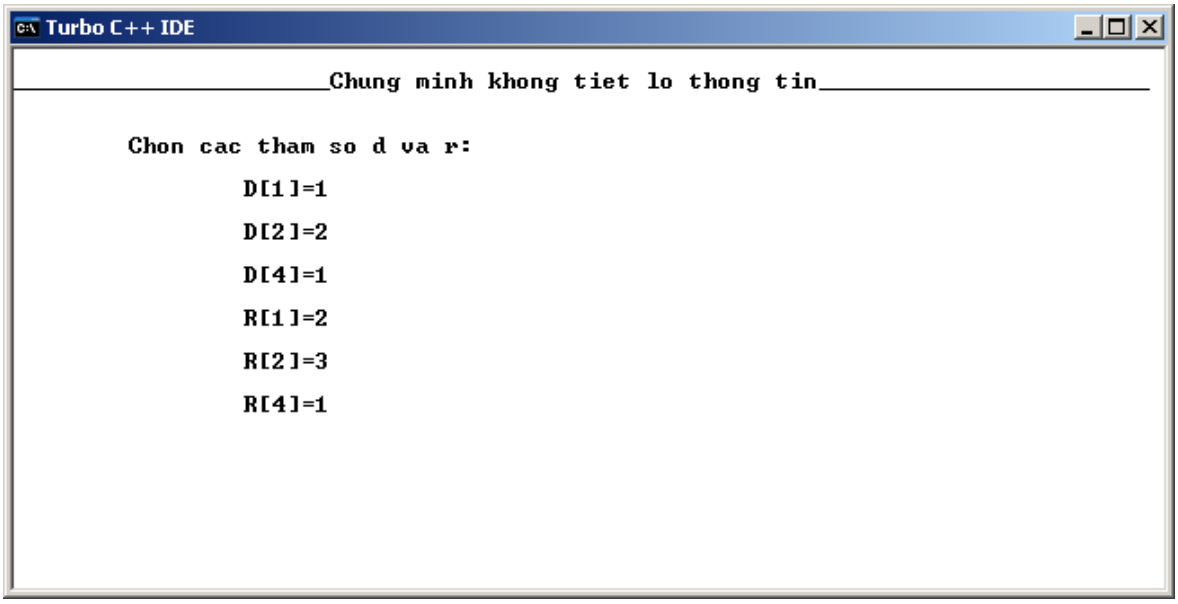
4.1.2. Các chức năng chính

1). Giao thức 1

Cử tri chứng minh tính hợp lệ của lá phiếu sau khi đã được mã hóa và gửi đến người xác minh TT :

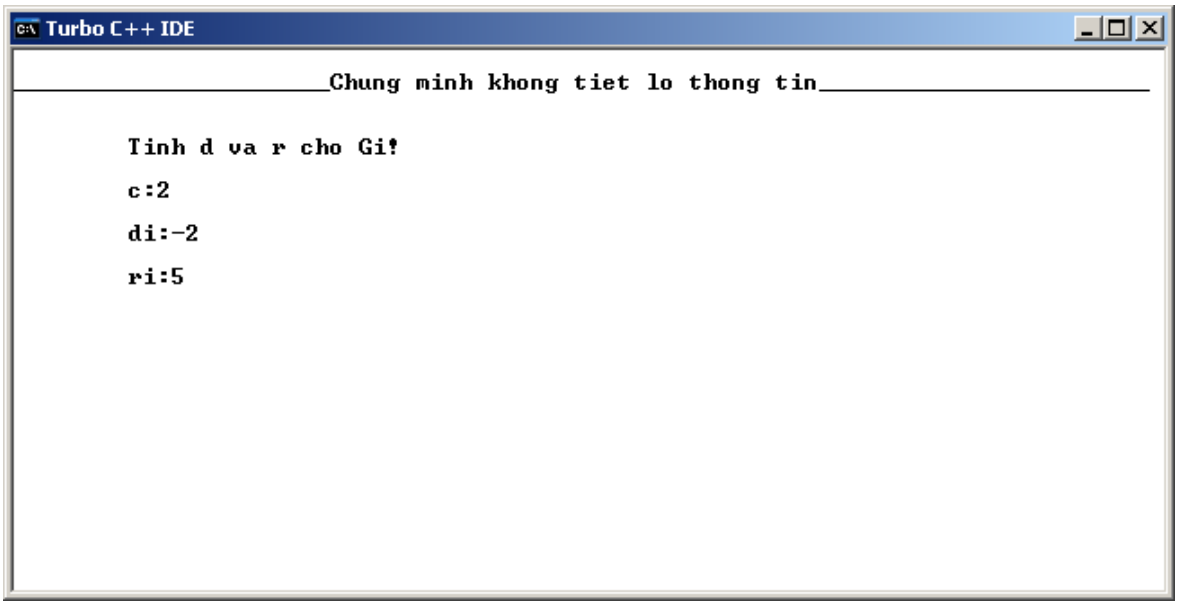
Bước 1: Điền các thông tin cần thiết để có thể mã hóa lá phiếu :





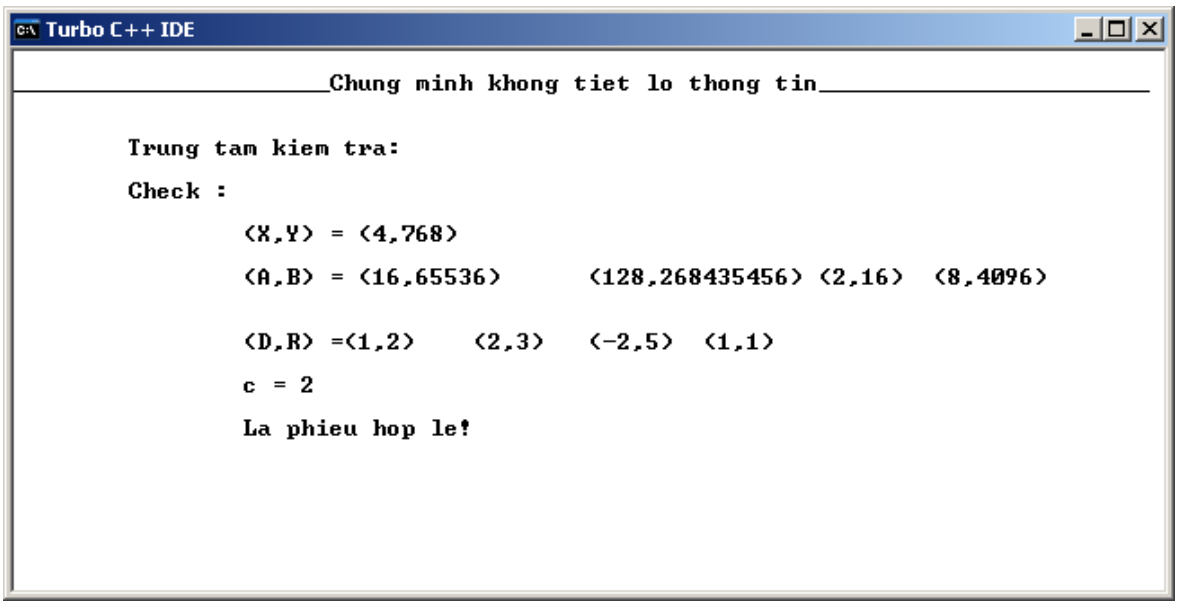
Cử tri điền các thông tin cần thiết để mã hóa lá phiếu thăm dò

Bước 2 : Tính d_i và r_i sau đó gửi (D, R) cho người xác minh TT:



Các thông số trả về từ người xác minh TT và các tính toán của Cử tri

Bước 3: Người xác minh TT sẽ kiểm tra: nếu các tham số không thỏa mãn thì sẽ loại lá phiếu, nếu đúng sẽ chấp nhận và tiếp tục mã hóa lá phiếu lần 2 và gửi cho Ban KP:

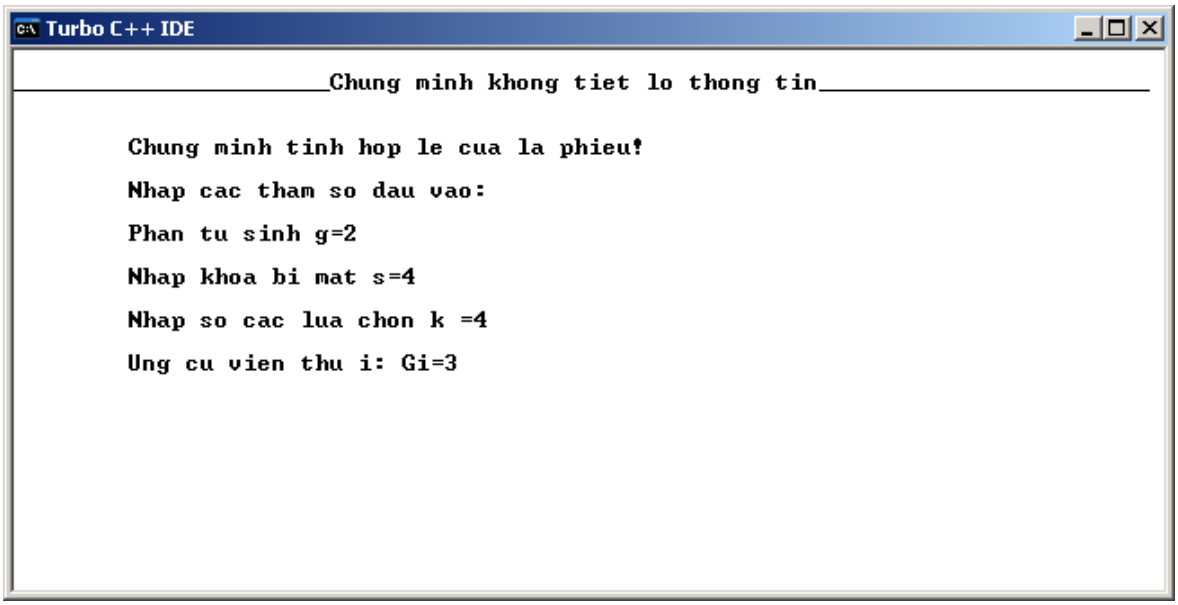


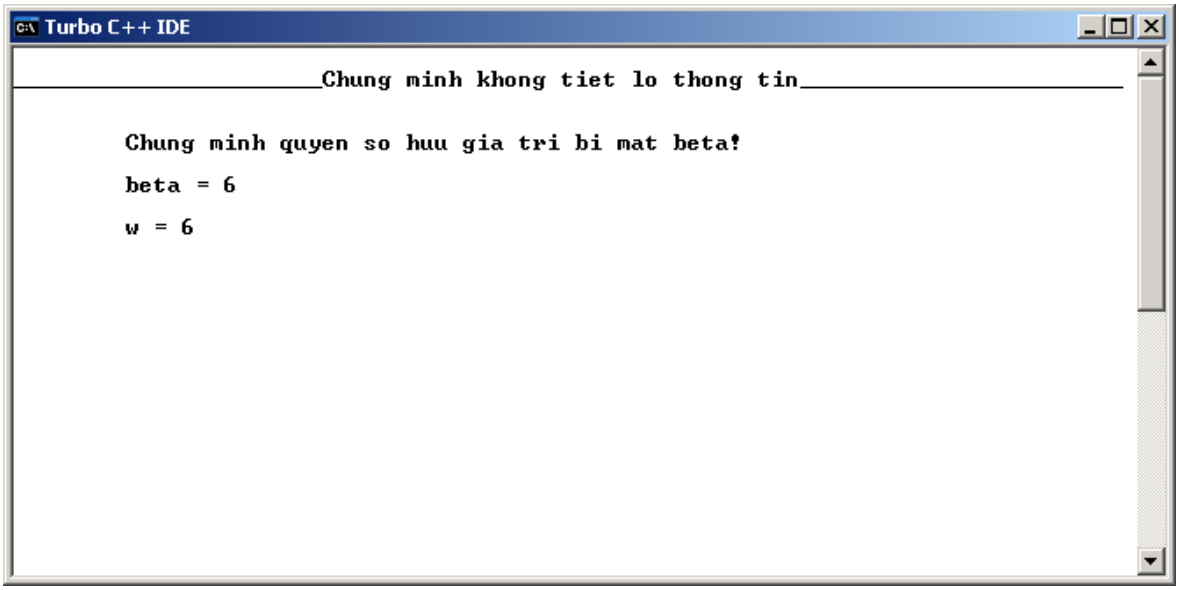
Lá phiếu khi đã được người xác minh TT kiểm tra lại

2). Giao thức 2

Người xác minh trung thực TT chứng minh lá phiếu làm mù gửi tới Ban KP cũng hoàn toàn hợp lệ :

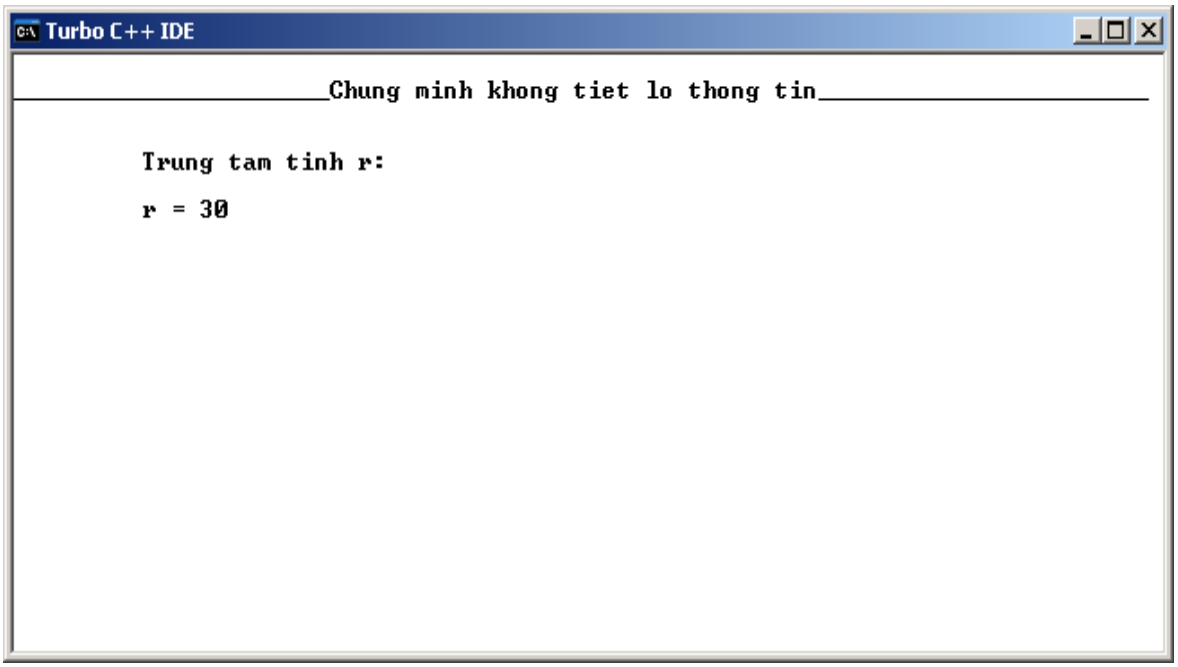
Bước 1: Người xác minh trung thực TT sẽ điền các tham số đầu vào và tính toán, sau đó người xác minh trung thực TT sẽ gửi luôn Beta và w thông qua (a, b) cho Ban KP:





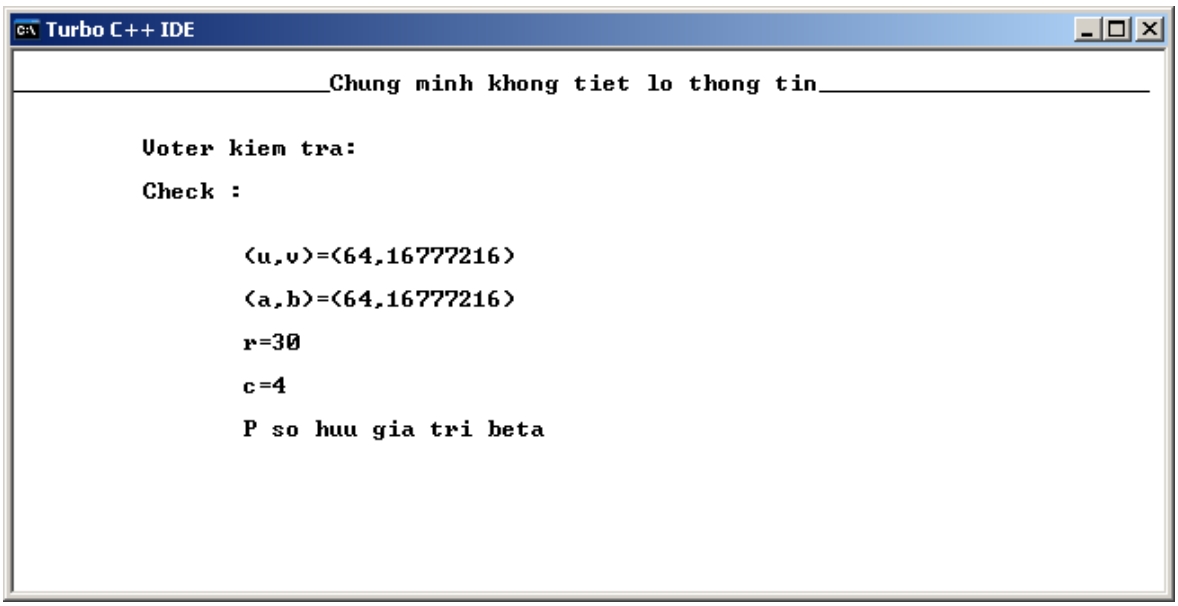
Người xác minh trung thực TT chọn Beta và w

Bước 2 : Ban KP trả lại giá trị c và người xác minh trung thực TT sẽ tính toán r và gửi r cho Ban KP.



Người xác minh trung thực TT tính r

Bước 3 : Ban KP kiểm tra lại kết quả nhận được, nếu đúng thì lá phiếu làm mù lần 2 hoàn toàn hợp lệ, nếu không đúng, Ban KP sẽ không chấp nhận.



Ban KP kiểm tra lại kết quả

4.2. MÃ NGUỒN CỦA CHƯƠNG TRÌNH

4.2.1. Cử tri chứng minh tính hợp lệ của lá phiếu

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#include<iostream.h>
struct ARRAY
{
	long A[10];
	int chiso;
};
ARRAY getArrayAj(int g,ARRAY d,ARRAY r,int Gi,int w,long x,int k);
ARRAY getArrayBj(long h,ARRAY d,ARRAY r,int Gi,int w,long y,int k);
int check(int g,long h,ARRAY a,ARRAY b,ARRAY d,ARRAY r,long x,long y,int
c,int Gi);
long getAi(int g,int w);
long getBi(long h,int w);
int getDi(ARRAY &d,int c,int Gi,int k);
int getRi(int w,int alpha,int di);
void addDi(ARRAY &d,int Gi,int di);
void addRi(ARRAY &r,int Gi,int ri);
void nhapARRAYd(ARRAY &d,int k,int Gi);
void nhapARRAYr(ARRAY &r,int k,int Gi);
```

```
// Lấy mảng a
ARRAY getArrayAj(int g,ARRAY d,ARRAY r,int Gi,int w,long x,int k)
{
	ARRAY a;
	a.chiso=d.chiso=r.chiso=k;
	for(int i=1;i<Gi;i++)  {
		a.A[i]=pow(g,r.A[i])*pow(x,d.A[i]);
	}
	a.A[Gi]=getAi(g,w);
	for(i=Gi+1;i<=a.chiso;i++)  {
		a.A[i]=pow(g,r.A[i])*pow(x,d.A[i]);
	}
	return a;
}
// Lấy mảng b
ARRAY getArrayBj(long h,ARRAY d,ARRAY r,int Gi,int w,long y,int k)
{
	ARRAY b;
	b.chiso=d.chiso=r.chiso=k;
	for(int i=1;i<Gi;i++)  {
		b.A[i]=pow(h,r.A[i])*pow((y/Gi),d.A[i]);
	}
	b.A[Gi]=getBi(h,w);
	for(i=Gi+1;i<=b.chiso;i++)  {
		b.A[i]=pow(h,r.A[i])*pow((y/Gi),d.A[i]);
	}
	return b;
}
```

```
//Kiểm tra tính hợp lệ của lá phiếu
int check(int g,long h,ARRAY a,ARRAY b,ARRAY d,ARRAY r,long x,long y,int
c,int Gi,int k)
{
	long sum=0;
	a.chiso=b.chiso=d.chiso=r.chiso=k;
	for(int i=1;i<=d.chiso;i++)
	{
		sum=sum+d.A[i];
	}
	if(sum!=c) return 0;
	for(i=1;i<=d.chiso;i++)
	{
		if(a.A[i]!=pow(g,r.A[i])*pow(x,d.A[i]))
				return 0;
		if(b.A[i]!=pow(h,r.A[i])*pow((y/Gi),d.A[i]))
				return 0;
	}
	return 1;
}
//Tính giá trị của di
int getDi(ARRAY &d,int c,int Gi,int k)
{
	int di;
	di=c;
	d.chiso=k;
	for(int i=1;i<Gi;i++)
	{
		di=di-d.A[i];
	}
```

```
        for(i=Gi+1;i<=d.chiso;i++)
        {
                di=di-d.A[i];
        }
        return di;
}
//Tính giá trị của ri
int getRi(int w,int alpha,int di)
{
        int ri;
        ri=w-alpha*di;
        return ri;
}
// Tính ai
long getAi(int g,int w)
{
        long ai;
        ai=pow(g,w);
        return ai;
}
// Tính bi
long getBi(long h,int w)
{
        long bi;
        bi=pow(h,w);
        return bi;
}
```

```
// Thêm di vào mảng d
void addDi(ARRAY &d,int Gi,int di)
{

	d.A[Gi]=di;
}
// Thêm ri vào mảng r
void addRi(ARRAY &r,int Gi,int ri)
{
	r.A[Gi]=ri;
}
// Chọn các phần tử d, chưa chọn di
void nhapARRAYd(ARRAY &d,int k,int Gi)
{
	d.chiso=k;
	for(int i=1;i<Gi;i++)
	{
		cout<<"\n\n\t\tD["<<i<<"]=";cin>>d.A[i];
	}
	d.A[Gi]=0;
	for(i=Gi+1;i<=d.chiso;i++)
	{
		cout<<"\n\n\t\tD["<<i<<"]=";cin>>d.A[i];
	}
}
```

```
//Chọn các phần tử r, chưa chọn ri
void nhapARRAYr(ARRAY &r,int k,int Gi)
{
	r.chiso=k;
	for(int i=1;i<Gi;i++)
	{
		cout<<"\n\n\t\tR["<<i<<"]=";cin>>r.A[i];
	}
	r.A[Gi]=0;
	for(i=Gi+1;i<=r.chiso;i++)
	{
		cout<<"\n\n\t\tR["<<i<<"]=";cin>>r.A[i];
	}
}
// Chương trình chính
void main()
{
	clrscr();
	int g,s,alpha,w,k,c,Gi,ri,di;
	long h,x,y;
	ARRAY d,r,a,b;
	cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
	cout<<"\n\n\tChung minh tinh hop le cua la phieu!";
	cout<<"\n\n\tNhap cac tham so dau vao:";
	cout<<"\n\n\tPhan tu sinh g="; cin>>g;
	cout<<"\n\n\tPhan tu sinh alpha="; cin>>alpha;
	cout<<"\n\n\tNhap khoa bi mat s=";cin>>s;
	h=pow(g,s);
	cout<<"\n\n\tNhap so cac lua chon k =";cin>>k;
```

```
        cout<<"\n\n\tUng cu vien thu i: Gi=";cin>>Gi;
        randomize();
        w=random(5);
        cout<<"\n\n\tw="<<w;
        getch();
        clrscr();
        cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
        cout<<"\n\n\tChon cac tham so d va r:\n";
        nhapARRAYd(d,k,Gi);
        nhapARRAYr(r,k,Gi);
        clrscr();
        cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
        cout<<"\n\n\tTinh d va r cho Gi!";
        c=random(5);
        cout<<"\n\n\tc:"<<c;
        di=getDi(d,c,Gi,k);
        cout<<"\n\n\tdi:"<<di;
        addDi(d,Gi,di);
        ri=getRi(w,alpha,di);
        cout<<"\n\n\tri:"<<ri;
        addRi(r,Gi,ri);
        getch();
        clrscr();
```

```
        cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
        cout<<"\n\n\tTrung tam kiem tra:";
        x=pow(g,alpha);
        y=pow(h,alpha)*Gi;
        a=getArrayAj(g,d,r,Gi,w,x,k);
        b=getArrayBj(h,d,r,Gi,w,y,k);
        cout<<"\n\n\tCheck : ";
        if(check(g,h,a,b,d,r,x,y,c,Gi,k)==0)
                cout<<"\n\n\t\tLa phieu khong hop le!";
        if(check(g,h,a,b,d,r,x,y,c,Gi,k)==1)
        {
                cout<<"\n\n\t\t(X,Y) = ("<<x<<","<<y<<")";
                cout<<"\n\n\t\t(A,B) = ";
                for(int i=1;i<=k;i++)
                {
                        cout<<"("<<a.A[i]<<","<<b.A[i]<<")\t";
                }
                cout<<"\n\n\t\t(D,R) =";
                for(int j=1;j<=k;j++)
                {
                        cout<<"("<<d.A[j]<<","<<r.A[j]<<")\t";
                }
                cout<<"\n\n\t\tc = "<<c;
                cout<<"\n\n\t\tLa phieu hop le!";
        }
        getch();
}
```

4.2.2. Người xác minh trung thực chứng minh có giữ tham số bí mật β

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#include<iostream.h>

void check(int r,int c,long a,long b,long u,long v,int g,long h);



//Kiểm tra P có sở hữu giá trị beta không
void check(int r,int c,long a,long b,long u,long v,int g,long h)
{
	if(pow(g,r)==a*pow(u,c)&&pow(h,r)==b*pow(v,c))
		{
			cout<<"\n\n\t\t(u,v)=("<<u<<","<<v<<")";
			cout<<"\n\n\t\t(a,b)=("<<a<<","<<b<<")";
			cout<<"\n\n\t\tr="<<r;
			cout<<"\n\n\t\tc="<<c;
			cout<<"\n\n\t\tP so huu gia tri beta";
		}
		else cout<<"\n\n\t\tP khong so huu gia tri beta";
}
```

```
// Chương trình chính
void main()
{
	clrscr();
	int g,s,beta,w,k,c,Gi;
	long u,v,a,b,h,r;
	cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
	cout<<"\n\n\tChung minh tinh hop le cua la phieu!";
	cout<<"\n\n\tNhap cac tham so dau vao:";
	cout<<"\n\n\tPhan tu sinh g="; cin>>g;
	cout<<"\n\tNhap khoa bi mat s=";cin>>s;
	h=pow(g,s);
	cout<<"\n\tNhap so cac lua chon k =";cin>>k;
	cout<<"\n\tUng cu vien thu i: Gi=";cin>>Gi;
	clrscr();
	cout<<"\n_____Chung minh khong tiet lo thong
tin_____\n";
	cout<<"\n\n\tChung minh quyen so huu gia tri bi mat beta!";
	cout<<"\n\n\tbeta = "; cin>>beta;
	randomize();
	w=random(10);
	cout<<"\n\tw = "<<w;
	u=pow(g,beta);
	v=pow(h,beta);
	a=pow(g,w);
	b=pow(h,w);
	c=random(10);
	getch();
	clrscr();
```

```
        cout<<"\n_____________________Chung minh khong tiet lo thong
tin_____________________\n";
        r=w+beta*c;
        cout<<"\n\n\t Trung tam tinh r:";
        cout<<"\n\n\t r = "<<r;
        getch();
        clrscr();
        cout<<"\n_____________________Chung minh khong tiet lo thong
tin_____________________\n";
        cout<<"\n\n\t Voter kiem tra:";
        cout<<"\n\n\t Check :\n";
        check(r,c,a,b,u,v,g,h);
        getch();
}
```

KẾT LUẬN

Sau thời gian thực hiện khóa luận, em thu được hai kết quả chính:

1/. Tìm hiểu và nghiên cứu qua tài liệu các vấn đề:

- Vấn đề "Phương pháp chứng minh không tiết lộ thông tin"
- Ứng dụng "Chứng minh không tiết lộ thông tin" trong bỏ phiếu từ xa
- Ứng dụng "Chứng minh không tiết lộ thông tin" trong sử dụng tiền điện tử

Về vấn đề chứng minh không tiết lộ thông tin, là nội dung chính của khóa luận này, em đã tìm hiểu các khái niệm cơ bản và một số sơ đồ chứng minh. Từ đó, tìm ra ưu nhược điểm và tính chất của các sơ đồ để có thể áp dụng vào các bài toán một cách thích hợp.

2/. Thử nghiệm chương trình ứng dụng "chứng minh không tiết lộ thông tin" trong bỏ phiếu từ xa.

TÀI LIỆU THAM KHẢO

[1] Andrew Neff, *"Conducting a Universally Verifiable Electronic Election Using Homomorphic Encryption "*, VoteHere.net, November 2000

[2] Berry Schoenmakers, *"A brief Comparision of Cryptographic Schemes for Electronic Voting"*, Tartu, Estonia, May 17, 2004

[3] Byoungcheon Lee, Kwangjo Kim, *"Receipt-free Electronic Voting through Collaboration of Voter and honest Verifier"*

[4] Helger Lipmaa, *"Zero knowledge and some applications"*, Nordic Research Training course, Bergen, June 15, 2004

[5] Information Security Research Centre, Faculty of Information Technology, Queensland University of Technology, *"Electronic Voting and Cryptography"*, May 2002

[6] Ivan Damgard, Jens Groth and Gorm Salomonsen, *"The Theory and Implementation of an Electronic Voting System"*, July 31,2002

[7] Trịnh Nhật Tiến, Nguyễn Đình Nam, Trương Thị Thu Hiền, "*Một số kỹ thuật Bỏ phiếu từ xa",* Hội thảo Một số vấn đề chọn lọc của Công nghệ thông tin, Thái Nguyên, tháng 8 năm 2003

[8] Trịnh Nhật Tiến, Trương Thị Thu Hiền, "*Mã hóa đồng cấu và ứng dụng",* Hội nghị khoa học cơ bản và ứng dụng CNTT toàn quốc lần thứ 1, Đại học Quốc Gia Hà Nội, tháng 10 năm 2003