MỤC LỤC

LỜI CẢM ƠN

Người xưa có câu: "Uống nước nhớ nguồn, ăn quả nhớ kẻ trồng cây". Với em sinh viên khoá 9 của trường Đại Học Dân Lập Hải Phòng luôn luôn ghi nhớ những công lao to lớn của các thầy giáo, cô giáo. Những người đã dẫn dắt chúng em từ khi mới bước chân vào giảng đường đại học những kiến thức, năng lực và đạo đức chuẩn bị hành trang bước vào cuộc sống để xây dựng đất nước khi ra trường sau 4 năm học. Em xin hứa sẽ lao động hết mình đem những kiến thức học được phục vụ cho Tổ quốc. Em xin chân thành cảm ơn đến:

Cha, mẹ người đã sinh thành và dưỡng dục con, hỗ trợ mọi điều kiện về vật chất và tinh thần cho con trên con đường học tập lòng biết ơn sâu sắc nhất.

Thầy cô của trường và các thầy cô trong Ban giám hiệu, thầy cô trong Bộ môn CNTT của trường Đại học Dân lập Hải Phòng đã tận tình giảng dạy và tạo mọi điều kiện cho chúng em học tập trong suốt thời gian học tập tại trường.

Thầy Trần Ngọc Thái– Giáo viên hướng dẫn tiểu án tốt nghiệp đã tận tình, hết lòng hướng dẫn em trong suốt quá trình nghiên cứu để hoàn thành đồ án tốt nghiệp này. Em mong thầy luôn luôn mạnh khoẻ để nghiên cứu và đào tạo nguồn nhân lực cho đất nước.

Một lần nữa em xin chân thành cảm ơn.

Hải Phòng, ngày ..... tháng ..... năm 2009

Sinh viên thực hiện

Nguyễn Thanh Tùng

MỞ ĐẦU

Hiện nay, sự kết hợp của vật lý lượng tử và cơ sở toán học hiện đại đã tạo nền móng cho việc xây dựng máy tính lượng tử trong tương lai. Theo các dự báo thì máy tính lượng tử sẽ xuất hiện vào khoảng những năm 2010-2020. Isaac L. Chuang, người đứng đầu nhóm nghiên cứu của IBM về máy tính lượng tử cũng đã khẳng định "*Máy tính lượng tử sẽ bắt đầu khi định luật Moore kết thúc – vào khoảng năm 2020, khi mạch được dự báo là đạt đến kích cỡ của nguyên tử và phân tử*").

Với khả năng xử lý song song và tốc độ tính toán nhanh, mô hình máy tính lượng tử đã đặt ra các vấn đề mới trong lĩnh vực CNTT. Vào năm 1994, Peter Shor đã đưa ra thuật toán phân tích số ra thừa số nguyên tố trên máy tính lượng tử với độ phức tạp thời gian đa thức. Như vậy khi máy tính lượng tử xuất hiện sẽ dẫn đến các hệ mã được coi là an toàn hiện nay như RSA sẽ không còn an toàn. Điều này đặt ra vấn đề nghiên cứu các hệ mật mới để đảm bảo an toàn khi máy tính lượng tử xuất hiện. Đồng thời, do máy tính lượng tử hiện nay mới chỉ xuất hiện trong phòng thí nghiệm, nhu cầu mô phỏng các thuật toán lượng tử trên máy tính thông thường là tất yếu.

Ở Việt Nam hiện nay, các nhà toán học cũng bước đầu có những nghiên cứu về tính toán lượng tử và mô phỏng tính toán lượng tử trên máy tính thông thường. Ví dụ như nhóm Quantum của trường Đại học Bách Khoa Hà Nội. Tuy nhiên vẫn còn nhiều vấn đề để mở, và việc này cần có sự đầu tư thích đáng, tìm tòi, thực nghiệm trên cơ sở những thành tựu về lý thuyết và kinh nghiệm sẵn có trên thế giới, đồng thời áp dụng vào thực tế.

Mục đích, đối tượng và nội dung của luận văn

Trong khuôn khổ luận văn này, trên những cơ sở những thành tựu đã có trên thế giới và trong nước em sẽ trình bày tổng quan các nghiên cứu lý thuyết về tính toán lượng tử, đồng thời mô phỏng thuật toán mã hóa lượng tử BB84. Luận văn gồm có phần mở đầu, kết luận và 04 chương đề cập tới các nội dung chính như sau:

Chương 1: Giới thiệu tổng quan về an toàn bảo mật thông tin,các khái niệm toán học, các hệ mã cổ điển,các chữ ký số

Chương 2: Các khái niệm cơ bản về mã hóa lượng tử, đặc trưng và một số vấn đề liên quan

Chương 3: Mã hóa lượng tử và giao thức phân phối khóa BB84

Chương 4: Mô phỏng giao thức BB84

CHƯƠNG 1: CÁC KHÁI NIỆM CƠ BẢN

1.1 Một số khái niệm toán học

1.1.1 Số nguyên tố và nguyên tố cùng nhau

Số nguyên tố là số nguyên dương chỉ chia hết cho 1 và chính nó.

Ví dụ: 2, 3, 5, 7, 17, … là những số nguyên tố.

Hệ mật mã thường sử dụng các số nguyên tố ít nhất là lớn hơn 10^{150}.

Hai số *m* và *n* được gọi là **nguyên tố cùng nhau** nếu ước số chung lớn nhất của chúng bằng 1. Ký hiệu: gcd(*m*, *n*) = 1.

Ví dụ: 9 và 14 là nguyên tố cùng nhau.

1.1.2 Đồng dư thức

Cho a và b là các số nguyên tố, n là số nguyên dương thì a được gọi là đồng dư với b theo modulo n nếu **n|a-b** (tức a - b chia hết cho n, hay khi chia a và b cho n được cùng một số dư như nhau). Số nguyên n được gọi là modulo của đồng dư.

Kí hiệu: **a ≡ b (mod n)**

Ví dụ: 67 ≡ 11 (mod 7), bởi vì 67 (mod 7) = **4** và 11 (mod 7) = **4**.

Tính chất của đồng dư:

Cho a, a_1, b, b_1, c ∈ Z. Ta có các tính chất:

- a≡b mod n nếu và chỉ nếu a và b có cùng số dư khi chia cho n.
- Tính phản xạ: a≡a mod n.
- Tính đối xứng: Nếu a≡b mod n thì b≡a mod n.
- Tính giao hoán: Nếu a≡b mod n và b≡c mod n thì a≡c mod n.
- Nếu a≡a_1 mod n, b≡b_1 mod n

  thì a + b≡(a_1 + b_1) mod n và ab≡a_1b_1 mod n.

1.1.3 Không gian Z_n và Z_n^*

Không gian Z_n (các số nguyên theo modulo n)

Là tập hợp các số nguyên {0, 1, 2, …, n-1}. Các phép toán trong Zn như cộng, trừ, nhân, chia đều được thực hiện theo module n.

Ví dụ: Z_{11} = {0, 1, 2, 3, …, 10}

Trong Z_{11}: 6 + 7 = **2**, bởi vì 6 + 7 = 13≡ **2** (mod 11).

Không gian Z_n^*

Là tập hợp các số nguyên p ∈ Z_n, nguyên tố cùng n.

Tức là: Z_n^* = {p ∈ Zn | gcd (n, p) =1}, Φ(n) là số phần tử của Z_n^*

Nếu n là một số nguyên tố thì: Z_n^* = {p ∈ Zn |1 ≤ p ≤ n-1}

Ví dụ: Z_2 = {0, 1} thì Z_2^* = {1} vì gcd(1, 2) = 1.

1.1.4 Phần tử nghịch đảo

Định nghĩa:

Cho a ∈ Z_n. Nghịch đảo của a theo modulo n là số nguyên $x \in Z_n$ sao cho ax ≡ 1 (mod n). Nếu x tồn tại thì đó là giá trị duy nhất, và *a* được gọi là khả nghịch, nghịch đảo của *a* ký hiệu là a^{-1}.

Tính chất:

- Cho $a, b \in Z_n$. Phép chia của *a* cho *b* theo modulo *n* là tích của *a* và b^{-1} theo modulo *n*, và chỉ được xác định khi *b* có nghịch đảo theo modulo *n*.
- Cho $a \in Z_n$, *a* là khả nghịch khi và chỉ khi gcd(*a*, *n*) = 1.
- Giả sử *d*=*gcd* (*a*, *n*). Phương trình đồng dư $ax \equiv b$ mod n có nghiệm *x* nếu và chỉ nếu *d* chia hết cho *b*, trong trường hợp các nghiệm *d* nằm trong khoảng 0 đến *n* - 1 thì các nghiệm đồng dư theo modulo *n/d*.

Ví dụ: 4^{-1} = 7 (mod 9) vì 4.7≡1 (mod 9)

1.1.5 Khái niệm nhóm, nhóm con, nhóm Cyclic

Nhóm là bộ các phần tử $(G, *)$ thỏa mãn các tính chất:

- Kết hợp: $(x * y) * z = x * (y * z)$
- Tồn tại phần tử trung lập $e \in G: e * x = x * e = x , \forall x \in G$
- Tồn tại phần tử nghịch đảo $x' \in G: x' * x = x * x' = e$

Nhóm con của nhóm $(G,*)$ là bộ các phần tử $(S,*)$ thỏa mãn các tính chất:

- $S \subset G$, phần tử trung lập $e \in S$.
- $x, y \in S => x * y \in S$.

Nhóm Cyclic: Là nhóm mà mọi phần tử của nó được sinh ra từ một phần tử đặc biệt $g \in G$.

Phần tử này được gọi là **phần tử sinh (nguyên thủy)**, tức là:

Với $\forall x \in G: \exists n \in N$ mà $g^n = x$.

Ví dụ: $(Z^+, *)$ là nhóm cyclic có phần tử sinh là 1.

Định nghĩa:

Ta gọi **Cấp** của nhóm là số các phần tử trong nhóm đó.

Như vậy, nhóm Z_n^* có cấp $\Phi(n)$.

Nếu p là số nguyên tố thì nhóm Z_p^* có cấp là p-1

Định nghĩa:

Cho $a \in Z_n^*$, cấp của a ký hiệu là **ord(a)**

được định nghĩa là số nguyên dương nhỏ nhất **t** thoả mãn: $\mathbf{a^t \equiv 1} \pmod n$.

Ví dụ: $Z_{21}^* = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$, $\Phi(21) = 12 = |Z_{21}^*|$

và cấp của từng thành phần trong Z_{21}^* là:

$a \in Z_{21}^*$	1	2	4	5	8	10	11	13	16	17	19	20
Cấp của a	1	6	3	6	2	6	6	2	3	6	6	2

1.1.6 Bộ phần tử sinh (Generator-tuple)

{$\mathbf{g_1}$, ..., $\mathbf{g_k}$} được gọi là bộ phần tử sinh nếu mỗi $\mathbf{g_i}$ là một phần tử sinh và những phần tử này khác nhau ($\mathbf{g_i} \neq \mathbf{g_j}$ nếu $\mathbf{i} \neq \mathbf{j}$).

Ví dụ: **{3, 5}** là bộ phần tử sinh của Z_7^*, bởi vì:

$1 = \mathbf{3}^6 \bmod 7 = \mathbf{5}^6 \bmod 7$

$2 = \mathbf{3}^2 \bmod 7 = \mathbf{5}^4 \bmod 7$

$3 = \mathbf{3}^1 \bmod 7 = \mathbf{5}^5 \bmod 7$

$4 = \mathbf{3}^4 \bmod 7 = \mathbf{5}^2 \bmod 7$

$5 = \mathbf{3}^5 \bmod 7 = \mathbf{5}^1 \bmod 7$

$6 = \mathbf{3}^3 \bmod 7 = \mathbf{5}^3 \bmod 7$.

2 không phải là phần tử sinh của Z_7^*, bởi vì:

$\{2, 2^2, 2^3, 2^4, 2^5, 2^6\} = \{2,4,1,2,4,1\} <=> \mathbf{\{1,2,4\}}$

Tuy nhiên **{1,2,4}** là tập con của {1, 2, 3, 4, 5, 6} = Z_7^*,

do đó số **2** được gọi là "phần tử sinh của nhóm **G(3)"**,

G(3) là nhóm có 3 thành phần {1,2,4}.

1.1.7 Bài toán đại diện (Presentation problem).

Gọi **g** là phần tử sinh của nhóm con G(q) thuộc Z_n^*. Bài toán logarit rời rạc liên quan đến việc tìm số mũ **a**, sao cho:

$\mathbf{a} = \log_{\mathbf{g}}\mathbf{h} \bmod n$ (với $h \in G(q)$).

Cho **k**>= 2, 1<=**ai**<= q, i = 1 …k.

Bài toán đại diện là: cho **h** thuộc G(q), tìm {$\mathbf{a_1}$, ... , $\mathbf{a_k}$}, của bộ phần tử sinh {g1, ... , gk} ,

sao cho:

$$h = g_1^{a_1} * g_2^{a_2} * .. * g_k^{a_k} \bmod n$$

{$\mathbf{a_k}$, ... , $\mathbf{a_k}$} được gọi là ***đại diện (representation)***.

Ví dụ:

Cho tập Z^*_{23}, thì ta có thể tìm được:

nhóm con **G(11)**={1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18} với những phần tử sinh g_i là: **2**, **3**, 4, 6, 8, 9, 12, 13, 16, 18.

{2, 3} là 2 phần tử sinh của nhóm con G(11) trong Z^*_{23}.

Bài toán đại diện là với **h** = 13 ∈ G(11), tìm $\{a_1, a_2\}$ sao cho:

$$13 = 2^{a_1} * 3^{a_2} \bmod 23$$

Logarit hai vế, có a_1*log (2) + a_2*log (3) = log (13) mod 23.

Kết quả là: $\mathbf{a_1}$ = 2 và $\mathbf{a_2}$ = 2, vì $2^2 * 3^2 = 4*9 = 36 = 13 \bmod 23$.

Hay $\mathbf{a_1}$ = 7 và $\mathbf{a_2}$ = 11, vì $2^7 * 3^{11} = 128*177147 = 13 \bmod 23$.

1.1.8 Hàm băm.

Hàm băm ***h*** là hàm một chiều (one-way hash) với các đặc tính sau:

- Với thông điệp đầu vào ***x*** thu được bản băm ***z*** = ***h***(***x***) là duy nhất.
- Nếu dữ liệu trong thông điệp ***x*** thay đổi hay bị xóa để thành thông điệp ***x***' thì ***h***(***x***') ≠ ***h***(***x***). Cho dù chỉ là một sự thay đổi nhỏ hay chỉ là xóa đi 1 bit dữ liệu của thông điệp thì giá trị băm cũng vẫn thay đổi. Điều này có nghĩa là: hai thông điệp hoàn toàn khác nhau thì giá trị hàm băm cũng khác nhau.
- Nội dung của thông điệp gốc "khó" suy ra từ giá trị hàm băm. Nghĩa là: với thông điệp ***x*** thì dễ dàng tính được ***z*** = ***h***(***x***), nhưng lại "khó" suy ngược lại ***x*** nếu chỉ biết giá trị hàm băm ***h***(***x***).

Tính chất:

Hàm băm ***h*** là không va chạm yếu:

Nếu cho trước một bức điện x, thì không thể tiến hành về mặt tính toán để tìm ra một bức điện x' ≠ x mà h(x') = h(x).Hàm băm ***h*** là không va chạm mạnh:

Nếu không có khả năng tính toán để tìm ra hai bức thông điệp x và x' mà x ≠ x' và h(x) = h(x').

1.2 Các khái niệm mã hóa

1.2.1 Khái niệm mã hóa.

Ta biết rằng tin truyền trên mạng rất dễ bị lấy cắp. Để đảm bảo việc truyền tin an toàn người ta thường mã hoá thông tin trước khi truyền đi. Việc mã hoá thường theo quy tắc nhất định gọi là hệ mật mã. Hiện nay có hai loại hệ mật mã mật mã cổ điển và mật mã khoá công khai. Mật mã cổ điển dễ hiểu, dễ thực thi nhưng độ an toàn không cao. Vì giới hạn tính toán chỉ thực hiện trong phạm vi bảng chữ cái sử dụng văn bản cần mã hoá (ví dụ Z_{26} nếu dùng các chữ cái tiếng anh, Z_{256} nếu dùng bảng chữ cái ASCII...). Với các hệ mã cổ điển, nếu biết khoá lập mã hay thuật toán thuật toán lập mã, người ta có thể "dễ" tìm ra được bản rõ. Ngược lại các hệ mật mã khoá công khai cho biết khoá lập mã K và hàm lập mã C_k thì cũng rất "khó" tìm được cách giải mã.

1.2.1.1 Hệ mã hóa.

Hệ mã hóa là hệ bao gồm 5 thành phần (P, C, K, E, D) thỏa mãn các tính chất sau:

P (Plaitext): là tập hợp hữu hạn các bản rõ có thể.

C (Ciphertext): Là tập hữu hạn các bản mã có thể

K (Key): Là tập hợp các bản khoá có thể

E (Encrytion): Là tập hợp các quy tắc mã hoá có thể

D (Decrytion): Là tập hợp các quy tắc giải mã có thể.

Chúng ta đã biết một thông báo thường được xem là bản rõ. Người gửi sẽ làm nhiệm vụ mã hoá bản rõ, kết quả thu được gọi là bản mã. Bản mã được gửi đi trên đường truyền tới người nhận. Người nhận giải mã để tìm hiểu nội dung bản rõ. Dễ dàng thấy được công việc trên khi định nghĩa hàm lập mã và hàm giải mã:

$E_k(P) = C$ và $D_k(C) = P$

1.2.1.2 Những khả năng của hệ mật mã.

- Cung cấp một mức cao về tính bảo mật, tính toàn vẹn, chống chối bỏ và tính xác thực.
- Tính bảo mật: Bảo đảm bí mật cho các thông báo và dữ liệu bằng việc che dấu thông tin nhờ các kỹ thuật mã hoá.
- Tính toàn vẹn: Bảo đảm với các bên rằng bản tin không bị thay đổi trên đường truyền tin.
- Chống chối bỏ: Có thể xác nhận rằng tài liệu đã đến từ ai đó, ngay cả khi họ cố gắng từ chối nó.
- Tính xác thực: Cung cấp hai dịch vụ:
  - Nhận dạng nguồn gốc của một thông báo và cung cấp một vài bảo đảm rằng nó là đúng sự thực.
  - Kiểm tra định danh của người đang đăng nhập một hệ thống, tiếp tục kiểm tra đặc điểm của họ trong trường hợp ai đó cố gắng kết nối và giả danh là người sử dụng hợp pháp.

1.2.2 Các phương pháp mã hóa.

1.2.2.1 Mã hóa đối xứng

Hệ mã hoá đối xứng: là hệ mã hoá tại đó khoá mã hoá có thể "dễ" tính toán ra được từ khoá giải mã và ngược lại. Trong rất nhiều trường hợp, khoá mã hoá và khoá giải mã là giống nhau.

Thuật toán này có nhiều tên gọi khác nhau như thuật toán khoá bí mật, thuật toán khoá đơn giản, thuật toán một khoá. Thuật toán này yêu cầu người gửi và người nhận phải thoả thuận một khoá trước khi thông báo được gửi đi và khoá này phải được cất giữ bí mật. Độ an toàn của thuật toán này phụ thuộc vào khoá, nếu để lộ ra khoá này nghĩa là bất kỳ người nào cũng có thể mã hoá và giải mã thông báo trong hệ thống mã hoá. Sự mã hoá và giải mã của hệ mã hoá đối xứng biểu thị bởi:

$E_k : P \rightarrow C$ Và $D_k : C \rightarrow P$

Nơi ứng dụng: Sử dụng trong môi trường mà khoá đơn dễ dàng được chuyển, như là trong cùng một văn phòng. Cũng dùng để mã hoá thông tin khi lưu trữ trên đĩa nhớ.

Các vấn đề đối với Hệ mã hoá đối xứng:

- Phương pháp mã hoá đối xứng đòi hỏi người mã hoá và người giải mã phải cùng chung một khoá. Khoá phải được giữ bí mật tuyệt đối. "Dễ dàng" xác định một khoá nếu biết khoá kia và ngược lại.
- Hệ mã hoá đối xứng không an toàn nếu khoá bị lộ với xác xuất cao. Hệ này khoá phải được gửi đi trên kênh an toàn.
- Vấn đề quản lý và phân phối khoá là khó khăn, phức tạp khi sử dụng hệ mã hoá đối xứng. Người gửi và người nhận phải luôn thống nhất với nhau về khoá. Việc thay đổi khoá là rất khó và dễ bị lộ.
- Khuynh hướng cung cấp khoá dài mà nó phải được thay đổi thường xuyên cho mọi người, trong khi vẫn duy trì cả tính an toàn lẫn hiệu quả chi phí, sẽ cản trở rất nhiều tới việc phát triển hệ mật mã.

1.2.2.2 Mã hóa phi đối xứng (Mã hóa công khai).

Hệ mã hoá khoá công khai: là Hệ mã hoá trong đó khoá mã hoá là khác với khoá giải mã. Khoá giải mã "khó" tính toán được từ khoá mã hoá và ngược lại. Khoá mã hoá gọi là khoá công khai (Public key). Khoá giải mã được gọi là khoá bí mật (Private key).

Nơi ứng dụng: Sử dụng chủ yếu trong việc trao đổi dữ liệu công khai.

Các điều kiện của một hệ mã hoá công khai:

Việc tính toán ra cặp khoá công khai K_B và bí mật k_B dựa trên cơ sở các điều kiện ban đầu, phải được thực hiện một cách dễ dàng, nghĩa là thực hiện trong thời gian đa thức.

Người gửi A có được khoá công khai của người nhận B và có bản tin P cần gửi B, thì có thể dễ dàng tạo ra được bản mã C.

$C = E_{KB}(P) = E_B(P)$

Người nhận B khi nhận được bản mã C với khoá bí mật k_B, thì có thể giải mã bản tin trong thời gian đa thức.

$P = D_{kB}(C) = D_B[E_B(P)]$

Nếu kẻ địch biết khoá công khai K_B cố gắng tính toán khoá bí mật thì chúng phải đương đầu với trường hợp nan giải, đó là gặp bài toán "khó".

1.2.3 Một số hệ mã hoá cụ thể.

1.2.3.1 Hệ mã hoá RSA.

Cho n=p*q với p, q là số nguyên tố lớn. Đặt $P = C = Z_n$

Chọn b nguyên tố với $\phi(n)$, $\phi(n) = (p-1)(q-1)$

Ta định nghĩa: $K=\{(n,a,b): a*b \equiv 1(\text{mod } \phi(n))\}$

Giá trị n và b là công khai và a là bí mật

Với mỗi K=(n, a, b), mỗi $x \in P$, $y \in C$ định nghĩa

Hàm mã hóa: $y = e_k(x) = x^b \text{ mod } n$

Hàm giải mã: $d_k(x) = y^a \text{ mod } n$

1.2.3.2 Hệ mã hoá ElGamal.

Hệ mã hóa với khoá công khai ElGamal có thể được dựa trên tuỳ ý các nhóm mà với họ đó bài toán lôgarit rời rạc được xem là “khó” giải được.

Thông thường người ta dùng nhóm con G_q (cấp q) của Z_p; ở đó p, q là các số nguyên tố lớn thoả mãn q|(p-1). Ở đây giới thiệu cách xây dựng nhóm Z_p, với p là một số nguyên tố lớn.

Sơ đồ:

Chọn số nguyên tố lớn p sao cho bài toán logarit rời rạc trong Z_p là “khó” (ít nhất $p = 10^{150}$). Chọn g là phần tử sinh trong Z^*_p .

Lấy ngẫu nhiên một số nguyên α thoả mãn $1 \leq \alpha \leq p-2$ và tính toán $h = g^{\alpha} \bmod p$.

Khoá công khai chính là (p, g, h), và khoá bí mật là α.

Mã hoá: khoá công khai là (p, g, h) muốn mã hoá thư tín m ($0 \leq m < p$)

Lấy ngẫu nhiên một số nguyên k, $0 \leq k \leq p-2$.

Tính toán $x = g^k \bmod p$, $y = m * h^k \bmod p$.

Giải mã. Để phục hồi được bản gốc m từ c = (x, y), ta làm như sau:

Sử dụng khoá riêng α, tính toán $r = x^{p-1-\alpha}$.

(Chú ý rằng $r = x^{p-1-\alpha} = x^{-\alpha} = (gk)^{-\alpha} = g^{-k\alpha}$).

Phục hồi m bằng cách tính toán $m = y*r \bmod p$.

1.2.3.3 Mã hoá đồng cấu.

Xét một sơ đồ mã hoá xác suất. Giả sử P là không gian các văn bản chưa mã hoá và C là không gian các văn bản mật mã. Có nghĩa là P là một nhóm với phép toán 2 ngôi $\oplus$ và C là một nhóm với phép toán $\otimes$. Ví dụ E của sơ đồ mã hoá xác suất được hình thành bởi sự tạo ra khoá riêng và khoá công khai của nó. Giả sử $E_r(m)$ là sự mã hoá thư tín m sử dụng tham số (s) r ta nói rằng sơ đồ mã hoá xác suất là $(\oplus,\otimes)$ đồng cấu. Nếu với bất kỳ ví dụ E của sơ đồ này, ta cho $c_1 = E_{r1}(m_1)$ và $c_2 = E_{r2}(m_2)$ thì tồn tại r sao cho:

$c_1 \otimes c_2 = E_r(m_1 \oplus m_2)$

Chẳng hạn, sơ đồ mã hoá Elgamal là đồng cấu. Ở đây, P là tập tất cả các số nguyên modulo p ($P = Z_p$), còn $C = \{(a,b) \mid a,b \in Z_p\}$. Phép toán $\oplus$ là phép nhân modulo p . Đối với phép toán 2 ngôi $\otimes$ được định nghĩa trên các văn bản mật mã, ta dùng phép nhân modulo p trên mỗi thành phần.

Hai văn bản gốc m_0, m_1 được mã hoá:

$E_{ko}(m_o) = (g^{ko}, h^{ko}m_o)$

$E_{k1}(m_1) = (g^{k1}, h^{k1}m_1)$

Ở đó k_o,k_1 là ngẫu nhiên.

Từ đó: $E_{ko}(m_o)\ E_{k1}(m_1) = (g^{ko}, h^{ko}m_o)\ (g^{k1}, h^{k1}m_1) = E_k(m_om_1)$

với $k = k_o + k_1$

Bởi vậy, trong hệ thống bí mật ElGamal từ phép nhân các văn bản mật mã chúng ta sẽ có được phép nhân đã được mã hoá của các văn bản gốc tương ứng.

1.2.3.4 Mã nhị phân.

Giả sử rằng Alice muốn gửi cho Bob 1 chữ số nhị phân b. Cô ta không muốn tiết lộ b cho Bob ngay. Bob yêu cầu Alice không được đổi ý, tức là chữ số mà sau đó Alice tiết lộ phải giống với chữ số mà cô ta nghĩ bây giờ.

Alice mã hoá chữ số b bằng một cách nào đó rồi gửi sự mã hoá cho Bob. Bob không thể phục hồi được b tới tận khi Alice gửi chìa khoá cho anh ta. Sự mã hoá của b được gọi là một blob.

Một cách tổng quát, sơ đồ mã nhị phân là một hàm ξ: {0, 1} x X$\rightarrow$ Y, trong đó X, Y là những tập hữu hạn. Mỗi mã hoá của b là giá trị ξ (b, k), k$\in$X. Sơ đồ mã nhị phân phải thoả mãn những tính chất sau:

- Tính che đậy (Bob không thể tìm ra giá trị b từ ξ(b, k))

- Tính mù (Alice sau đó có thể mở ξ(b, k) bằng cách tiết lộ b, k thì được dùng trong cách xây dựng nó. Cô ta không thể mở blob bởi 0 hay 1).

Nếu Alice muốn mã hoá một xâu những chữ số nhị phân, cô ta mã hoá từng chữ số một cách độc lập.

Sơ đồ mã hoá số nhị phân mà trong đó Alice có thể mở blob bằng 0 hay 1 được gọi là mã hoá nhị phân cửa lật.

Mã hoá số nhị phân có thể được thực hiện như sau:

Giả sử một số nguyên tố lớn p, một phần tử sinh g $\in$ Z_p và G $\in$ Z_p đã biết logarit rời rạc cơ số g của G thì cả Alice và Bob đều không biết (G có thể chọn ngẫu nhiên). Sự mã hoá nhị phân ξ: {0,1} x $Z_p\rightarrow Z_p$ là:

$\xi(b, k) = g^kG^b$

Đặt $\log_g G = a$. Blob có thể được mở bởi b bằng cách tiết lộ k và mở bởi -b bằng cách tiết lộ k-a nếu b=0 hoặc k+a nếu b=1. Nếu Alice không biết a, cô ta không thể mở blob bằng –b.

Tương tự, nếu Bob không biết k, anh ta không thể xác định b với chỉ một dữ kiện $\xi(b, k) = g^k G^b$.

Sơ đồ mã hoá chữ số nhị phân cửa lật đạt được trong trường hợp Alice biết a.

Nếu Bob biết a và Alice mở blob cho Bob thông qua kênh chống đột nhập đường truyền (untappable channel) Bob có thể sẽ nói dối với người thứ ba về sự mã hoá chữ số nhị phân b. Rất đơn giản, anh ta nói rằng anh ta nhận được k-a hoặc k+a (mà thực tế là k). Sơ đồ mã hoá số nhị phân mà cho phép người xác minh (Bob) nói dối về việc mở blob, được gọi là sự mã hoá nhị phân chameleon.

Thay vì mã hoá từng chữ số nhị phân trong sâu s một cách độc lập, Alice có thể mã hoá một cách đơn giản $0 \le s \le p$ bằng $\xi(b, k) = G^s g^k$.

Hơn nữa, những thông tin về số a sẽ cho Alice khả năng mở ξ (s,k) bởi bất kì s', k' thoả mãn as + k = as' + k'.1.3 Khái niệm về chữ ký điện tử

1.3.1 Định nghĩa

Một sơ đồ chữ ký gồm bộ 5 (P, A, K, S, V) thoả mãn các điều kiện dưới đây:

P là tập hữu hạn các bức điện (thông điệp) có thể

A là tập hữu hạn các chữ kí có thể

K không gian khoá là tập hữu hạn các khoá có thể

Sig_k là thuật toán ký $P \rightarrow A$

$x \in P \rightarrow y = Sig_k(x)$

Ver_k là thuật toán kiểm thử: $(P, A) \rightarrow$ (Đúng, sai)

$$Ver_k(x, y) = \begin{cases} \text{Đúng Nếu } y = Sig_k(x) \\ \text{Sai Nếu } y \neq Sig_k(x) \end{cases}$$

1.3.2 Phân loại sơ đồ chữ ký điện tử.

Chữ ký "điện tử" được chia làm 2 lớp, lớp chữ ký kèm thông điệp (message appendix) và lớp chữ ký khôi phục thông điệp (message recovery).

Chữ ký kèm thông điệp: Đòi hỏi thông điệp ban đầu là đầu vào của giải thuật kiểm tra. Ví dụ: chữ ký Elgamal.

Chữ ký khôi phục thông điệp: Thông điệp ban đầu sinh ra từ bản thân chữ ký. Ví dụ: chữ ký RSA.

1.3.3 Một số sơ đồ ký số cơ bản.

1.3.3.1 Sơ đồ chữ ký Elgamal.

Chọn p là số nguyên tố sao cho bài toán log rời rạc trong Z_p là khó.

Chọn g là phần tử sinh $\in Z_p^*$; a $\in Z_p^*$.

Tính $\beta \equiv g^a \bmod p$.

Chọn r ngẫu nhiên $\in Z^*_{p-1}$

Ký trên x: Sig(x) = (γ, δ),

Trong đó $\gamma = g^k \bmod p$, $\delta = (x - a\gamma)\, r^{-1} \bmod (p-1)$.

Kiểm tra chữ ký:

Ver(x, γ, δ)=True $\Leftrightarrow \beta^\gamma \gamma^\delta \equiv g^x \bmod p$

Ví dụ:

Chọn p=463; g=2; a=211;

$\beta \equiv 2^{211} \bmod 463 = 249$;

chọn r =235; $r^{-1}=289$

Ký trên x = 112

Sig(x,r) = Sig (112,235)=(γ,δ)=(16,108)

$\gamma = 2^{235} \bmod 463 = 16$

δ= (112-211*16)*289 mod (463-1)=108

Kiểm tra chữ ký:

Ver(x, γ, δ)=True $\Leftrightarrow \beta^\gamma \gamma^\delta \equiv g^x \bmod p$

$\beta^{\gamma}\ \gamma^{\delta} = 249^{16} * 16^{108} \bmod 463 = 132$

$g^x \bmod p = 2^{112} \bmod 463 = 132$

1.3.3.2 Sơ đồ chữ ký RSA.

Chọn p, q nguyên tố lớn .

Tính n=p.q; $\phi(n)=(p-1)(q-1)$.

Chọn b nguyên tố cùng $\phi(n)$.

Chọn a nghịch đảo với b; $a=b^{-1} \bmod \phi(n)$.

Ký trên x:

Sig (x) = $x^a \bmod n$

Kiểm tra chữ ký:

Ver (x,y)= True $\Leftrightarrow x \equiv y^b \bmod n$

Ví dụ:

p=3; q=5;

n=15; $\phi(n)= 8$;

chọn b=3; a=3

Ký x =2:

Chữ ký :

$y = x^a \bmod n = 2^3 \bmod 15=8$

Kiểm tra:

$x = y^b \bmod n = 8^3 \bmod 15 =2$ (chữ ký đúng)

1.3.3.3 Sơ đồ chữ ký Schnorr.

Chuẩn bị:

Lấy *G* là nhóm con cấp *q* của Z_n^* , với *q* là số nguyên tố.

Chọn phần tử sinh $g \in G$ sao cho bài toán logarit trên *G* là khó giải.

Chọn $x \neq 0$ làm khóa bí mật, $x \in Z_q$. Tính $y = g^x$ làm khóa công khai.

Lấy *H* là hàm băm không va chạm.

Ký trên thông điệp m:

Chọn r ngẫu nhiên thuộc Z_q

Tính $c = H(m, g^r)$

Tính $s = (r - c\,x) \bmod q$

Chữ ký Schnorr là cặp (c, s)

Kiểm tra chữ ký:

Với một văn bản *m* cho trước, một cặp (*c, s*) được gọi là một chữ ký Schnorr hợp lệ nếu thỏa mãn phương trình:

$$\mathbf{c = H(m, g^S * y^C)}$$

Để ý rằng ở đây, c xuất hiện ở cả 2 vế của phương trình

1.4 Phân phối khóa và thỏa thuận khóa

Như chúng ta đã biết, hệ thống mã khóa công khai có ưu điểm hơn hệ thống mã khóa cổ điển ở chỗ có thể công khai thuật toán mã hoá cho nhiều người sử dụng. Tuy nhiên, hầu hết các hệ thống mã khóa công khai đều chậm hơn hệ thống mã khóa cổ điển, chẳng hạn như DES. Vì thế thực tế các hệ thống mã khóa riêng được sử dụng để mã các bức điện dài.

Giả sử, có một mạng không an toàn gồm **n** người sử dụng, có trung tâm được uỷ quyền (**TT**) để đáp ứng những việc như xác minh danh tính của người sử dụng, chọn và gửi khoá đến người sử dụng…Do mạng không an toàn nên cần được bảo vệ trước các đối phương. Đối phương có thể là người bị động, nghĩa là hành động của anh ta chỉ hạn chế ở mức nghe trộm bức điện truyền trên kênh. Song mặt khác, anh ta có thể là người chủ động, tức là anh ta có thể tráo đổi khoá mật của 2 đối tác tham gia truyền tin.

Ta phân biệt giữa phân phối khóa và thỏa thuận khóa. Phân phối khóa là cơ chế một nhóm chọn khóa mật và sau đó truyền nó đến các nhóm khác. Thoả thuận khóa là giao thức để hai nhóm (hoặc nhiều hơn) liên kết với nhau cùng thiết lập một khóa mật bằng cách liên lạc trên kênh công khai.

Mục tiêu của phân phối khoá và giao thức thoả thuận khoá là tại thời điểm kết thúc thủ tục, hai nhóm đều có cùng khoá K song nhóm khác không biết được .

1.4.1 Phân phối khóa

*** Vấn đề.**

Theo phương pháp cơ bản, TT tạo ra $\binom{n}{2}$ khóa và đưa mỗi khoá cho duy nhất một cặp người sử dụng trong mạng n dùng. Như đã nêu ở trên ta cần một kênh an toàn giữa TT và mỗi người sử dụng để truyền đi các khóa này. Nếu n lớn, giải pháp này không thực tế cả về lượng thông tin cần truyền đi an toàn, lẫn thông tin mà mỗi người sử dụng phải cất giữ an toàn (nghĩa là các khoá mật của n – 1 người dùng khác).

Như vậy, điều cần quan tâm là cố gắng giảm được lượng thông tin cần truyền đi và cất giữ trong khi vẫn cho phép mỗi cặp người sử dụng U và V có khả năng tính khoá mật $K_{u,v}$.

1.4.1.1 Sơ đồ phân phối khoá trước Blom.

Ta giả thiết rằng có một mạng gồm n người sử dụng.

Giả sử rằng các khoá được chọn trên trường hữu hạn Z_p, trong đó p là số nguyên tố ($p \geq n$).

Cho k là số nguyên, $1 \leq k \leq n-2$. Giá trị k để hạn chế kích thước lớn nhất mà sơ đồ vẫn duy trì được độ mật.

TT sẽ truyền đi k+1 phần tử của Z_p, cho mỗi người sử dụng trên kênh an toàn (so với n-1 trong sơ đồ phân phối trước cơ bản). Mỗi cặp người sử dụng U và V sẽ có khả năng tính khoá $K_{u, v} = K_{v, u}$ như trước đây.

Điều kiện an toàn như sau: tập bất kì gồm nhiều nhất k người sử dụng không liên kết từ {U, V} phải không có khả năng xác định bất kì thông tin nào về $K_{u, v}$.

Xét trường hợp đặc biệt của sơ đồ Blom khi k =1.

a. Sơ đồ phân phối khoá Blom với k=1.

1. Số nguyên tố p công khai, với mỗi người sử dụng U, phần tử $r_u \in Z_p$ là công khai, khác nhau.

2. TT chọn 3 phần tử ngẫu nhiên bí mật a, b, c $\in Z_p$ (không cần khác biệt) và thiết lập đa thức:

$f(x, y) = (a + b*(x + y) + c*x*y) \mod p.$

3. Với mỗi người sử dụng U, TT tính đa thức:

$g_u(x) = f(x, r_u) \mod p$

và truyền $g_u(x)$ đến U trên kênh an toàn.

$g_u(x)$ là đa thức tuyến tính theo x, có thể viết:

$g_u(x) = f(x, r_u) \mod p = (a+b.(x+ r_u) + c.x.\ r_u \mod p) \mod p$

$g_u(x) = a_u + b_u*x$, trong đó:

$a_u = a + b*r_u \mod p$

$b_u = b + c*r_u \mod p$

4. Nếu U và V muốn liên lạc với nhau, họ sẽ dùng khoá chung $K_{u, v} = K_{v, u} = f(r_u, r_v) = (a + b*(r_u+ r_v) + c.r_u.r_v) \mod p.$

U tính $K_{u, v} : f(r_u, r_v) = g_u(r_v).$

V tính $K_{u, v} : f(r_u,r_v) = g_v(r_u).$

Ví dụ 1:

Giả sử có 3 người sử dụng là U,V và W.

Chọn số nguyên tố p =17, các phần tử công khai của họ là $r_u = 12$, $r_v = 7$, $r_w = 1$.

Giả thiết rằng TT chọn ngẫu nhiên, bí mật a = 8, b = 7, c = 2. Khi đó đa thức f như sau:

$f(x, y) = (8 + 7*(x + y) + 2*x*y) \mod 17.$

Các đa thức g của U, V, W tương ứng là:

$g_u(x) = (8 + 7*(x + 12) + 12*2*x) \mod 17 = 7 + 14*x.$

$g_v(x) = (8 + 7*(x + 7) + 7*2*x) \mod 17 = 6 + 4*x.$

$g_w(x) = (8 + 7*(x + 1) + 12*2*x) \mod 17 = 15 + 9*x.$

Như vậy 3 khoá là:

$K_{u, v} = f(r_u,r_v) = (8 + 7*(12 + 7) + 2*12*7) \mod 17 = 3.$

$K_{u, w} = f(r_u,r_w) =(8 + 7*(12 + 1) + 2*12*71) \mod 17 = 4.$

$K_{v, w} = f(r_v,r_w) = (8 + 7*(7 + 1) + 2*7*1) \mod 17 = 10.$

Nếu U và V muốn trao đổi với nhau thì.

U sẽ tính khoá $K_{u, v}$ như sau:

$g_u(r_v) = f(r_u,r_v) = 7 + 14*7 \mod 17 = 3.$

V sẽ tính khoá $K_{u, v}$ như sau:

$g_v(r_u) = f(r_u,r_v) = 6 + 4*12 \mod 17 =3.$

Ví dụ 2:

Giả sử có 3 người sử dụng là U,V và W.

Chọn số nguyên tố p =83, các phần tử công khai của họ là $r_u = 42$, $r_v = 31$, $r_w = 53$.

Giả thiết rằng TT chọn ngẫu nhiên, bí mật a = 10, b = 20, c = 30. Khi đó đa thức f như sau:

$f(x, y) = (10 + 20*(x + y) + 30*x*y) \mod 83.$

Các đa thức g của U, V, W tương ứng là:

$g_u(x) = (10 + 20*(x + 42) + 30*42*x) \mod 83 = 20 + 35*x.$

$g_v(x) = (10 + 20*(x + 31) + 30*31*x) \mod 83 = 49 + 37*x.$

$g_w(x) = (10 + 20*(x + 53) + 30*53*x) \mod 83 =74 + 33*x.$

Như vậy 3 khoá là:

$K_{u, v} = f(r_u,r_v) = (10 + 20*(42 + 31) + 30*42*31) \mod 83 =26.$

$K_{u, w} = f(r_u,r_w) = (10 + 20*(42 + 53) + 30*42*53) \mod 83 =49.$

$K_{v, w} = f(r_v,r_w) = (10 + 20*(31 + 53) + 30*31*53) \mod 83 =18.$

Nếu U và V muốn trao đổi với nhau thì.

U sẽ tính khoá $K_{u, v}$ như sau:

$g_u(r_v) = f(r_u,r_v) = 20 + 35*31 \mod 83 = 26.$

V sẽ tính khoá $K_{u, v}$ như sau:

$g_v(r_u) = f(r_u,r_v) = 49 + 37*42 \mod 17 = 26.$

Sự an toàn của sơ đồ Blom với k=1.

a. Sơ đồ an toàn với 1 đối thủ.

Không một người sử dụng nào có thể xác định được thông tin về khoá của 2 người sử dụng khác.

Định lý:

Theo sơ đồ Blom với k =1, khoá của một cặp đối tác là an toàn không điều kiện trước bất kì người sử dụng thứ ba.

***Chứng minh*:**

Giả sử người sử dụng thứ ba là W muốn thử tính khoá.

$K_{u,v} = (a + b*(r_u + r_v) + c*r_u*r_v) \mod p$

Trong đó các giá trị r_u, r_v là công khai, còn a, b, c không được biết.

W tìm biết được các giá trị:

$a_w = a + b*r_w \mod p.$

$b_w = b + c*r_w \mod p.$

Vì chúng là hệ số của đa thức $g_w(x)$ được TT gửi đến cho W.

Ta sẽ chỉ ra rằng thông tin mà W biết phù hợp với giá trị tùy ý $t \in Z_p$ của khoá $K_{u,v}$.

Xét phương trình ma trận sau:

$$\begin{matrix} 1 & r_u + r_v & r_u r_v \end{matrix} \begin{pmatrix} 1 & r_w & 0 \\ 0 & 1 & r_w \end{pmatrix} \begin{matrix} a \\ \begin{pmatrix} b \\ c \end{pmatrix} \end{matrix} = \begin{matrix} t \\ \begin{pmatrix} a_w \\ b_w \end{pmatrix} \end{matrix}$$

Phương trình đầu tiên thể hiện giả thiết rằng $K_{u,v} = t$, các chương trình thứ hai và ba chứa thông tin cho thấy W biết a, b và c từ $g_w(x)$.

Định thức của ma trận hệ số là:

$r_w^2 + r_u r_v - (r_u + r_v) r_w = (r_w - r_u)(r_w - r_v).$

Các phép số học được thực hiện trong Z_p. Vì $r_w \neq r_u$ và $r_w \neq r_v$ nên định thức ma trận hệ số khác không. Do đó phương trình ma trận có nghiệm duy nhất cho a, b, c. Nói cách khác, bất kì giá trị t nào thuộc Z_p cũng có thể nhận là khoá $K_{u,v}$.

b. Sơ đồ không an toàn với liên minh 2 đối thủ.

Liên minh của 2 người sử dụng {W, X} sẽ có khả năng xác định khoá $K_{u,v}$ bất kì , ở đây {W, X} ∩ {U, V} = 0.

W và X cùng biết rằng:

$a_w = a + b*r_w$.

$b_w = b + c*r_w$.

$a_x = a + b*r_x$.

$b_x = b + c*r_x$.

Như vậy, họ có bốn phương trình trong đó ba ẩn chưa biết và dễ dàng tính ra nghiệm duy nhất cho a, b, c. Một khi đã biết a, b, và c, họ có thể thiết lập đa thức f(x, y) và tính khoá bất kì mà họ muốn .

c. Cách thức khắc phục liên minh k đối thủ.

Để tạo lập sơ đồ có độ an toàn chống lại được liên minh k đối thủ, TT sẽ dùng đa thức f(x, y) có dạng:

$$f(x,y) = \sum_{i=0}^{k} \sum_{j=o}^{k} a_{i,j}\ x^i y^j \quad \text{mod} \quad p.$$

Trong đó:

$a_{i,j} \in Z_p$ $(0 \le i \le k, 0 \le j \le k)$ và $a_{i,j} = a_{j,i}$ với mọi i, j

Phần còn lại của giao thức không thay đổi.

1.4.1.2 Sơ đồ phân phối khoá trước Diffie-Hellman.

a. Sơ đồ.

Ta sẽ mô tả một sơ đồ phân phối khoá trước là cải tiến của giao thức trao đổi khoá Diffie-Hellman và gọi nó là sơ đồ phân phối khoá trước Diffie-hellman. Sơ đồ này an toàn về mặt tính toán vì nó liên quan đến bài toán logarit rời rạc "*khó giải*".

Mô tả sơ đồ trên Z_p (p là số nguyên tố), bài toán logarit rời rạc trong Z_p là khó giải.

Giả sử α là phần tử nguyên thuỷ của Z_p^*, trong đó các giá trị α và p là công khai.

Trong sơ đồ này, ID(U) là thông tin định danh nào đó cho người sử dụng U trên mạng, chẳng hạn như tên, địa chỉ thư điện tử, số điện thoại…Cũng như vậy, mỗi người sử dụng U đều có số mũ mật a_u (trong đó $0 \le a_u \le p - 2$) và một giá trị công khai tương ứng :

$b_u = \alpha^{a_u} \mod p$.

TT có sơ đồ chữ ký với thuật toán xác minh (công khai) ver_{TT} và thuật toán kí mật sig_{TT}. Ta giả thiết rằng tất cả thông tin đều được chia nhỏ ra nhờ dùng hàm hash công khai trước khi nó được kí.

Thông tin về người sử dụng U sẽ được xác thực bằng cách dùng dấu xác nhận của TT, trong đó có chữ ký của TT. Mỗi người sử dụng U sẽ có một dấu xác nhận :

$C(U) = (ID(U), b_u, sig_{TT}(ID(U), b_u))$.

Trong đó b_u được thiết lập theo mô tả ở phần trên. Dấu xác nhận của người sử dụng U sẽ được đóng vào khi U nối mạng. Có thể lưu các dấu xác nhận trong cơ sở dữ liệu công khai hoặc mỗi người sử dụng lưu dấu xác nhận của chính họ. Chữ ký xác nhận của TT cho phép bất kì ai trên mạng đều có thể xác minh được thông tin trên nó.

U và V rất dễ dàng tính ra khoá chung:

$$K_{u,v} = \alpha^{a_u a_v} \mod p$$

Sơ đồ phân phối khoá trước của Diffie- Hellman.

1. Số nguyên tố p, phần tử nguyên thuỷ $\alpha \in Z^*_p$ là công khai.

2. V tớnh:

$$K_{u,v} = \alpha^{a_u a_v} \mod p = b_u^{a_v} \mod p$$

b_u công khai nhận từ dấu xác nhận của U, giá trị a_v mật riêng của V.

3. U tớnh:

$$K_{u,v} = \alpha^{a_u a_v} \mod p = b_v^{a_u} \mod p$$

b_v công khai nhận từ dấu xác nhận của V, giá trị a_u mật riêng của U.

Ví dụ:

∗ Cho số nguyên tố p = 25307, phần tử nguyên thuỷ $\alpha = 2 \in Z^*_p$, chúng là công khai.

Giả sử U chọn $a_u = 3578$. Sau đó U tính:

$b_u = \alpha^{a_u} \bmod p = 2^{3578} \bmod 25307 = 6113.$

Giả sử V chọn $a_v = 19956$. Sau đó V tính:

$b_v = \alpha^{a_v} \bmod p = 2^{19956} \bmod 25307 = 7984.$

∗ U tính khoá:

$K_{u,v} = b_v^{a_u} \bmod p = 7984^{3578} \bmod 25307 = 3694.$

V tính khoá:

$K_{u,v} = b_u^{a_v} \bmod p = 6113^{19956} \bmod 25307 = 3694.$

Hai giá trị khoá bằng nhau.

Sự an toàn của sơ đồ.

Chữ ký của TT trên dấu xác thực của người dùng U ngăn chặn W có thể biến đổi thông tin nào đó trên dấu xác thực của người sử dụng U.

Vì thế ta chỉ cần lo lắng trước những tấn công thụ động. Như vậy câu hỏi là: liệu W có thể tính $K_{u,v}$ nếu W ≠ U,V hay không.

Mặt khác, khi biết $\alpha^{a_u} \bmod p$ và $\alpha^{a_v} \bmod p$ thì có khả năng tính được $\alpha^{a_u a_v} \bmod p$ hay không? Vấn đề này gọi là bài toán Diffie-Hellman, nó được mô tả hình thức như sau:

Bài toán:

I = (p, α, β, γ), trong đó p là số nguyên tố, $\alpha \in Z_p^*$ là phần tử nguyên thuỷ, còn $\beta, \gamma \in Z_p^*$

Mục tiêu:

Tính $\beta^{\log_\alpha \gamma} \bmod p \ (= \gamma^{\log_\alpha \beta} \bmod p)$.

Rõ ràng, sơ đồ phân khối khoá trước Diffie-Hellman là an toàn trước đối phương thụ động khi và chỉ khi bài toán Diffie-Hellman khó giải.

Nếu W có thể xác định được a_u từ b_u hoặc a_v từ b_v thì anh ta có thể tính $K_{u,v}$ một cách chính xác như U hoặc V. Song cả hai tính toán này đều là các trường hợp của bài toán logarit rời rạc. Vì thế chỉ cần bài toán logarit rời rạc trong Z_p là bài toán khó giải, thì sơ đồ phân phối khoá trước Diffie-Hellman sẽ an toàn trước kiểu tấn công này. Tuy nhiên, giả định cho rằng thuật toán bất kì giải được

bài toán Diffie-Hellman thì cũng có thể giải được bài toán logarith rời rạc vẫn chưa được chứng minh. (Điều này cũng tương tự như tình huống với RSA, trong đó giả định cho rằng việc phá RSA tương đương đa thức với bài toán phân tích số cũng không được chứng minh).

Theo nhận xét trên, bài toán Diffie-Hellman không khó hơn bài toán logarit rời rạc. Mặc dù không thể nói chính xác bài toán này khó như thế nào song ta có thể nói rằng độ an toàn của nó tương đương với độ mật của hệ mã hoá Elgamal.

Định lý:

Việc phá hệ mã hoá Elgamal tương đương với việc giải bài toán Diffie-Hellman.

Chứng minh:

Theo cách mã hóa và giải mã của hệ mã hoá Elgamal ta có:

Khoá mã K = (p, α, a, β), $\beta = \alpha^a \bmod p$ trong đó a bí mật còn p, α và β công khai.

Với số ngẫu nhiên bí mật $k \in Z_{p-1}$ ta có : $e_k(x, k) = (y_1, y_2)$.

Trong đó: $y_1 = \alpha^k \bmod p$ và $y_2 = x*\beta^k \bmod p$.

Với $y_1, y_2 \in Z_p^*$ thì $d_k(y_1, y_2) = y_2 * (y_1^a)^{-1} \bmod p$.

1. Giả sử có thuật toán A để giải bài toán Diffie-Hellman và có phép mã Elgamal (y_1, y_2).

Áp dụng thuật toán A với các đầu vào p, α, y_1, và β. Khi đó ta nhận được giá trị:

$A(p, \alpha, y_1, \beta) = A(p, \alpha, \alpha^k, \alpha^a) = \alpha^{ka} \bmod p = \beta^k \bmod p$

Khi đó, phép giả mã (y_1, y_2) có thể dễ dàng tính như sau:

$x = y_2(\beta^k)^{-1} \bmod p$.

2. Đối lập lại, giả thiết rằng ta có thuật toán B để thực hiện giải mã Elgamal, nghĩa là B tính các đầu vào p, α, y_1, β và y_2 và tính lượng:

$x = y_2(y_1^{\log_\alpha \beta})^{-1} \bmod p$.

Áp dụng các đầu vào p, α, β và γ đối với bài toán Diffie-Hellman, dễ dàng nhận thấy:

$B(p, \alpha, \beta, \gamma, 1)^{-1} = 1((\gamma^{\log_\alpha \beta})^{-1})^{-1} \bmod p = \gamma^{\log_\alpha \beta} \bmod p$.

1.4.1.3 Sơ đồ phân phối khoá trực tiếp Kerboros.

Nếu mỗi cặp người sử dụng không muốn tính một khoá cố định như trong phương pháp phân phối trước khoá thì có thể dùng phương pháp trực tiếp, trong đó khoá của phiên làm việc mới chỉ được tạo ra mỗi khi hai người sử dụng muốn liên lạc với nhau (gọi là tính tươi mới của khoá). Dùng phân phối khoá trực tiếp, người sử dụng mạng không cần lưu các khoá khi muốn liên lạc với những người sử dụng khác (Tuy nhiên mỗi người đều được chia sẻ khoá với TT). Khóa của phiên làm việc (khoá session) sẽ được truyền đi theo yêu cầu của TT. Đó là sự đáp ứng của TT để đảm bảo ***khoá tươi.***

Sơ đồ Kerboros.

Kerboros là hệ thống dịch vụ khoá phổ cập dựa trên mã khoá riêng. Mỗi người sử dụng U sẽ chia sẻ khoá DES mật K_u cho TT. Mọi thông báo cần truyền được mã hoá theo chế độ xích khối (CBC). ID(U) chỉ thông tin định danh công khai cho U. Khi có yêu cầu khoá session gửi đến, TT sẽ tạo ra một khoá session mới ngẫu nhiên K. Cũng như vậy TT sẽ ghi lại thời gian khi có yêu cầu T và chỉ ra thời gian tồn tại L để K có hiệu lực. Điều đó có nghĩa là khoá K chỉ có hiệu lực từ T đến T+L. Tất cả thông tin này đều được mã hoá và truyền đến U và V.

1. U yêu cầu TT khoá session để liên lạc với V.

2. TT chọn một khoá session ngẫu nhiên K, thời gian hệ thống T và thời gian tồn tại L.

3. TT tính:

$m_1 = e_{K_u}$ (K, ID(V), T, L)

và $m_2 = e_{K_v}$ (K, ID(U), T, L)

sau đó gửi m_1 và m_2 tới U.

4. U dùng hàm giải mã d_{K_u} để tính K, T, L và ID(U) từ m_1.

Sau đó anh ta tính:

$m_3 = e_K$(ID(U), T) và gửi m_3 đến V cùng với bức điện m_2 mà anh ta nhận được từ TT.

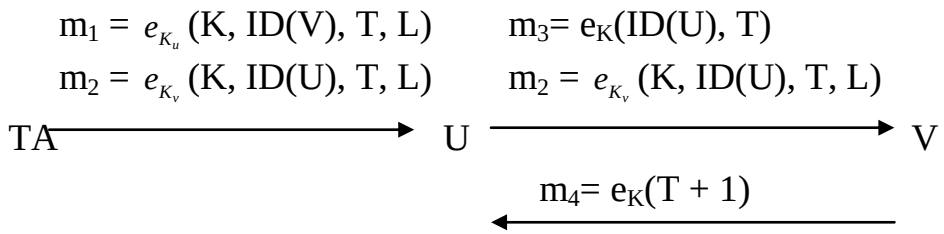
5. V dùng hàm giải mã d_{K_v} để tính K, T, L và ID(U) từ m_2. Sau đó dùng d_K để tính T và ID(U) từ m_3 và kiểm tra xem 2 giá trị của T và 2 giá trị của ID(U) có bằng nhau không. Nếu đúng thì V tính :

$m_4 = e_K(T + 1)$ và gửi nó đến U.

6. U giải mã m_4 bằng d_K và xác minh thấy kết quả bằng T+ 1.

Thực hiện giao thức

1. Thông tin truyền đi trong giao thức được minh hoạ như sau:



2. TT tạo ra K, T và L trong bước 2.

3. Trong bước 3 thông tin này cùng với ID(V) được mã hoá bằng khoá K_u (được U và TT chia sẻ) để tạo lập m_1. Cũng như vậy, K, T, L và ID(U) được mã hoá bằng K_v (được V và TT chia sẻ) để lập m_2. Cả hai bức điện đã mã hoá này được gửi đến U.

4. U có thể dùng khoá của mình để giải mã m_1, để nhận được K, T, L. Anh ta xác minh xem thời gian hiện tại có nằm trong khoảng T đến T + L hay không. Anh ta cũng kiểm tra khoá session K được phát ra cho liên lạc giữa anh ta và V bằng cách xác minh thông tin ID(V) đã giải mã từ m_1.

Tiếp theo, U sẽ làm trễ thời gian m_2 đến V. Cũng như vậy, anh ta sẽ dùng khoá session K mới để mã T và ID(U) và gửi kết quả m_3 tới V.

5. Khi V nhận được m_2 và m_3 từ U thì V sẽ giải mã m_2 thu được T, K, L và ID(U). Khi đó V sẽ dùng khoá session mới K để giải mã m_3 và xác minh xem T và ID(U) nhận được từ m_2 và m_3 có như nhau không. Điều này đảm bảo cho V rằng khoá session được mã từ m_2 cũng là khoá đã dùng để mã m_3. Khi đó V dùng khoá K để mã T + 1 và gửi kết quả m_4 trở về U.

6. Khi U nhận được m_4, anh ta dùng K để giải mã nó và xác minh xem kết quả có bằng T + 1 không. Điều này đảm bảo cho U là khoá session K đã được truyền thành công đến V vì K đã được dùng để tạo ra m_4.

Sự an toàn của sơ đồ.

1. Chức năng khác nhau của các thông báo trong giao thức:

+ m_1 và m_2 dùng để đảm bảo an toàn trong việc truyền khoá session.

+ m_3 và m_4 dùng để khẳng định khoá, nghĩa là cho phép U và V có thể thuyết phục nhau rằng họ sở hữu cùng một khoá session K.

Thời gian hệ thống T và thời hạn L để ngăn đối phương tích cực khỏi "lưu" thông báo cũ nhằm tái truyền lại sau này. Đây là phương pháp hiệu quả vì các khoá không được chấp nhận khi chúng quá hạn.

2. Mọi người sử dụng trong mạng đều phải có đồng hồ đồng bộ với nhau vì cần có thời gian hiện tại để xác định khoá session K cho trước là hợp lệ. Thực tế, rất khó có được sự đồng bộ hoàn hảo, nên phải cho phép có khoảng thay đổi nào đó về thời gian.

1.4.2 Thỏa thuận khóa

1.4.2.1 Sơ đồ trao đổi khoá Diffie-Hellman.

Nếu không muốn dùng dịch vụ khoá trực tiếp thì phải dùng giao thức thoả thuận khoá để trao đổi khoá mật. Giao thức thoả thuận khoá nổi tiếng nhất là giao thức trao đổi khoá Diffie-Hellman.

Sơ đồ.

Giả sử rằng p là số nguyên tố, α là phần tử nguyên thuỷ $\in Z^*_p$ và chúng đều công khai.

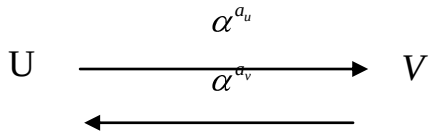
1. U chọn a_u ngẫu nhiên, bí mật ($0 \leq a_u \leq p-2$).
2. U tính :
   $\alpha^{a_u} \bmod p$ và gửi nó đến V.
3. V chọn a_v ngẫu nhiên, bí mật ($0 \leq a_v \leq p-2$).
4. V tính :
   $\alpha^{a_v} \bmod p$ và gửi nó đến U.
5. U tính khoá
   $K = (\alpha^{a_v})^{a_u} \bmod p$.
6. V tính khoá
   $K = (\alpha^{a_u})^{a_v} \bmod p$.

Cuối giao thức, U và V tính ra cùng một khoá:

$$K = \alpha^{a_u a_v} \bmod p.$$

Giao thức này cũng tương tự với sơ đồ phân phối trước khoá của Diffie-Hellman đã được mô tả. Sự khác nhau ở chỗ các số mũ a_u, a_v của U và V đều được chọn lại mỗi lần thực hiện giao thức này thay vì cố định. Như vậy cả U và V đều được đảm bảo ***khoá tươi*** vì khoá session phụ thuộc vào cả hai số ngẫu nhiên bí mật a_u và a_v.

Thông tin trao đổi trong giao thức được mô tả như sau:



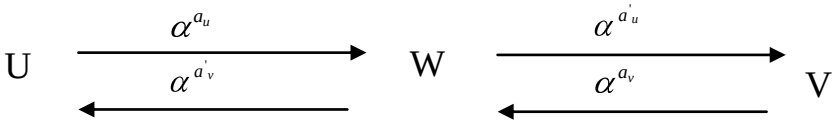
Sự an toàn của sơ đồ.

1. Hạn chế: chưa có xác thực danh tính.

Giao thức này dễ bị tổn thương trước đối phương tích cực – những người sử dụng cách tấn công "kẻ xâm nhập vào giữa cuộc". Đó là tình tiết của vở "The Lucy show", trong đó nhân vật Vivian Vance đang dùng bữa tối với người bạn, còn Lucille Ball đang trốn dưới bàn. Vivian và người bạn của cô đang cầm tay nhau dưới bàn. Lucy cố tránh bị phát hiện đã nắm lấy tay của cả hai người, còn hai người trên bàn vẫn nghĩ rằng họ đang cầm tay nhau.

Cuộc tấn công kiểu "kẻ xâm nhập vào giữa cuộc" trên giao thức trao đổi khoá Diffie-Hellman hoạt động cũng như vậy. W sẽ ngăn chặn các bức điện trao đổi giữa U và V và thay thế bằng các bức điện của anh ta.

Ta có sơ đồ như sau:



Tại thời điểm của cuối giao thức, U thiết lập thực sự khoá mật $\alpha^{a_u a'_v}$ cùng với W, còn V thiết lập khoá mật $\alpha^{a'_u a_v}$ với W.

Khi U cố giải mã bức điện để gửi cho V, W cũng có khả năng giải mã nó song V thì không thể, (tương tự tình huống nắm tay nhau nếu V gửi bức điện cho U).

2. Cải tiến: Bổ sung xác thực danh tính.

Điều cơ bản đối với U và V là bảo đảm rằng, họ đang trao đổi khoá cho nhau mà không có W. Trước khi trao đổi khoá, U và V có thể thực hiện những giao thưc tách bạch để thiết lập danh tính cho nhau. Tuy nhiên, điều này có thể đưa đến việc không bảo vệ được trước tấn công "kẻ xâm nhập giữa cuộc" nếu W vẫn duy trì một cách đơn giản sự tấn công thụ động cho đến khi U và V đã chứng minh danh tính của họ cho nhau. Vì thế giao thức thoả thuận khoá tự nó cần xác thực được các danh tính của những người tham gia cùng lúc khoá được thiết lập. Giao thức như vậy được gọi là giao thức ***thoả thuận khoá đã xác thực***.

1.4.2.2 Giao thức thoả thuận khoá trạm tới trạm.

Phần này sẽ mô tả một giao thức thoả thuận khoá là cải tiến của sơ đồ trao đổi khoá Diffie-Hellman, bổ sung xác thực danh tính C(U).

Sơ đồ.

Giả sử p là số nguyên tố, α là phần tử nguyên thuỷ $\in Z^*_p$. Trong đó p, α công khai và nó dùng với các dấu xác nhận.

Mỗi người sử dụng U sẽ có một sơ đồ chữ ký với thuật toán xác minh ver_u. TT cũng có sơ đồ chữ ký với thuật toán xác minh công khai ver_{TT} .

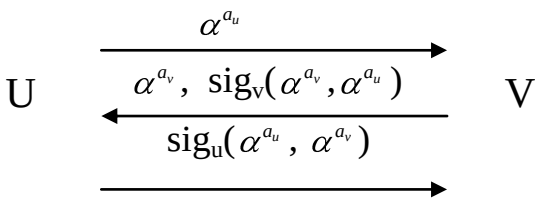
Mỗi người sử dụng U có dấu xác nhận:

$$C(U) = (ID(U), ver_u , sig_{TT}(ID(U), ver_u)).$$

Trong đó ID(U) là thông tin định danh cho U.

1. U chọn số ngẫu nhiên a_u, bí mật ($0 \le a_u \le p - 2$).
2. U tính:
   $\alpha^{a_u} \bmod p$ và gửi nó đến V.
3. V chọn số ngẫu nhiên a_v, bí mật ($0 \le a_v \le p - 2$).
4. V tính:
   $\alpha^{a_v} \bmod p$
   $K = (\alpha^{a_u})^{a_v} \bmod p$
   $y_v = sig_v(\alpha^{a_v}, \alpha^{a_u})$
5. V gửi $(C(V), \alpha^{a_v}, y_v)$ đến U.
6. U tính:
   $K = (\alpha^{a_v})^{a_u} \bmod p$
   U dùng ver_v để xác minh y_v và xác minh C(V) nhờ ver_{TT}.
7. U tính:
   $y_u = sig_u(\alpha^{a_u}, \alpha^{a_v})$
   và gửi $(C(U), y_u)$ đến V.
8. V xác minh y_u bằng ver_u và xác minh C(U) bằng ver_{TT}.

Thông tin trao đổi trong sơ đồ trạm đến trạm (STS) được minh hoạ như sau: Đây là giao thức 3 lần truyền tin.

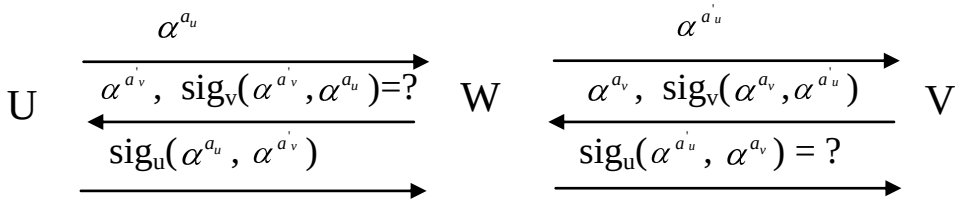


Sự an toàn của sơ đồ.

1. Xét cách bảo vệ trước tấn công kẻ xâm nhập giữa cuộc.

Như trước , W sẽ chặn bắt α^{a_u} và thay nó bằng $\alpha^{a'_u}$. Sau đó W nhận được α^{a_v}, $sig_v(\alpha^{a_v}, \alpha^{a'_u})$ từ V. Anh ta cũng muốn thay α^{a_v} bằng $\alpha^{a'_v}$ như trước đây. Tuy nhiên điều này có nghĩa anh ta cũng phải thay $sig_v(\alpha^{a_v}, \alpha^{a'_u})$ bằng $sig_v(\alpha^{a'_v}, \alpha^{a_u})$. Đáng tiếc là đối với W, anh ta không thể tính chữ ký của V trên $(\alpha^{a'_v}, \alpha^{a_u})$ vì không biết thuật toán ký sig_v của V. Tương tự, W không thể thay $sig_u(\alpha^{a_u}, \alpha^{a'_v})$ bằng $sig_v(\alpha^{a'_u}, \alpha^{a_v})$ do anh ta không biết thuật toán ký của U.

Minh hoạ bằng sơ đồ sau:



Đó là cách sử dụng các chữ ký mà không sợ kiểu tấn công kẻ xâm nhập giữa cuộc.

2. Giao thức, như mô tả không đưa ***ra sự khẳng định khoá***. Tuy nhiên, dễ dàng biến đổi để thực hiện được điều đó bằng cách:

Trong bước 4 mã hoá y_v bằng khoá session K:

$$y_v = e_K(sig_v(\alpha^{a_v}, \alpha^{a_u})) = e_K(y_v).$$

Trong bước 7 mã hoá y_u bằng khoá session K:

$$y_u = e_K(sig_u(\alpha^{a_u}, \alpha^{a_v})) = e_K(y_u).$$

1.4.2.3 Giao thức thoả thuận khoá MTI.

Matsumoto, Takashima và Imai đã xây dựng giao thức thoả thuận khoá đáng chú ý, bằng cách biến đổi giao thức trao đổi khoá của Diffie-Hellman. Giao thức này gọi là MTI.

Giao thức không đòi hỏi U và V phải tính bất kỳ chữ ký nào. Chúng là các giao thức hai lần vì chỉ có hai lần truyền thông tin riêng biệt (một từ U đến V và một từ V đến U). Giao thức STS là giao thức ba lần truyền tin.

Sơ đồ.

Giả thiết p là số nguyên tố, α là phần tử nguyên thuỷ $\in Z^*_p$. Các giá trị này công khai.

Mỗi người sử dụng U đều có định danh ID(U), số mũ bí mật a_u ($0 \le a_u \le p-2$) và giá trị công khai tương ứng:

$$b_u = \alpha^{a_u} \bmod p.$$

TT có sơ đồ chữ ký với thuật toán xác minh (công khai) ver_{TT} và thuật toán ký mật sig_{TT}. Mỗi người sử dụng U sẽ có dấu xác nhận:

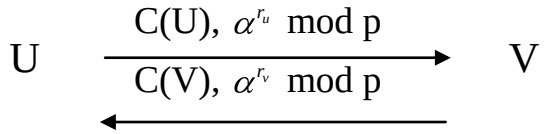
$$C(U) = (ID(U), b_u, sig_{TT}(ID(U), b_u)).$$

1. U chọn ngẫu nhiên r_u, $0 \le r_u \le p-2$
   và tính: $s_u = \alpha^{r_u} \bmod p$.
2. U gửi $(C(U), s_u)$ đến V.
3. V chọn ngẫu nhiên r_v, $0 \le r_v \le p-2$
   và tính: $s_v = \alpha^{r_v} \bmod p$.
4. V gửi $(C(V), s_v)$ đến U.
5. U tính khoá:
   $$K = s_v^{a_u} * b_v^{r_u} \bmod p.$$
   U nhận được giá trị b_v từ C(V).
6. V tính khoá:
   $$K = s_u^{a_v} * b_u^{r_v} \bmod p.$$
   V nhận được giá trị b_u từ C(U).

Cuối giao thức U và V đều tính cùng một khoá :

$$K = \alpha^{r_u * a_v + r_v * a_u} \bmod p.$$

Thông tin được truyền trong giao thức:

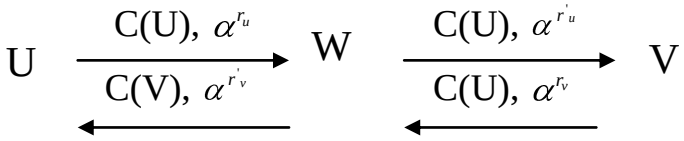


Sự an toàn của sơ đồ.

1. Độ mật của giao thức MTI trước tấn công thụ động đúng bằng bài toán Diffie-Hellman. Cũng như nhiều giao thức, việc chứng minh tính an toàn trước tấn công chủ động không phải đơn giản.

Khi không dùng chữ ký trong suốt quá trình thực hiện giao thức, có thể xuất hiện tình huống không có sự bảo vệ nào trước tấn công xâm nhập vào điểm giữa.

2. Hãy xét giao thức MTI, W có thể tráo đổi các giá trị mà U và V gửi cho nhau. Minh họa bằng sơ đồ sau:



Trong trường hợp này, U và V sẽ tính các khoá khác nhau:

U tính khoá:

$$K = \alpha^{r_u a_v + r'_v a_u} \bmod p.$$

V tính khoá:

$$K = \alpha^{r'_u a_v + r_v a_u} \bmod p.$$

Tuy nhiên, W không thể tính toán ra khoá của U và V vì chúng đòi hỏi phải biết số mũ mật a_u và a_v tương ứng. Thậm chí ngay cả khi U và V tính ra các khoá khác nhau (dĩ nhiên là không dùng chúng) thì W cũng không thể tính được khoá nào trong chúng.

Nói cách khác, cả U và V đều được đảm bảo rằng, người sử dụng khác trên mạng chỉ có thể tính được khoá mà họ tính được (đó là các khoá rởm). Tính chất này còn được gọi là ***xác thực khoá ẩn*** (implicit key authentication).

Ví dụ:

Giao thức thoả thuận khoá MTI:

Giả sử số nguyên tố p = 27803, α = 5 là phần tử nguyên thuỷ $\in Z^*_p$.

* U chọn bí mật a_u = 21131. Sau đó tính:

$$b_u = 5^{21131} \bmod 27803 = 21420.$$

được đặt trên giấy xác nhận của U.

V chọn bí mật a_v = 17555. Sau đó tính:

$$b_v = 5^{17555} \bmod 27803 = 17100.$$

được đặt trên giấy các nhận của V.

* Giả sử rằng U chọn r_u = 169, tính:

$$s_u = 5^{169} \bmod 27803 = 6268.$$

Sau đó U gửi giá trị s_u đến V.

Giả sử rằng V chọn r_v = 23456, tính:

$$s_v = 5^{23456} \bmod 27803 = 26759.$$

Sau đó V gửi giá trị s_v đến U.

* U tính khoá:

$$K_{u,v} = s_v^{a_u} * b_v^{r_u} \bmod p$$
$$= 26759^{21131}\ 17100^{169} \bmod 27803$$
$$= 21600.$$

V tính khoá:

$$K_{u,v} = s_u^{a_v} * b_u^{r_v} \bmod p$$
$$= 6268^{17555}\ 21420^{23456} \bmod 27803$$
$$= 21600.$$

Như vậy U và V đã tính cùng một khoá.

1.4.2.4 Thoả thuận khoá dùng các khoá tự xác nhận.

Phần này mô tả phương pháp thoả thuận khoá do Girault đưa ra không cần dấu xác nhận. Giá trị của khoá công khai và danh tính người sử hữu nó sẽ ngầm xác thực lẫn nhau.

Sơ đồ Girault kết hợp các tính chất của RSA và logarit rời rạc.

Sơ đồ.

Giả sử p, q, p_1, q_1 là các số nguyên tố lớn.

Trong đó n = p*q, p = $2p_1$ + 1, q = $2q_1$ + 1.

Nhóm nhân Z_n^* là đẳng cấu với $Z_p^* \times Z_p^*$. Bậc cực đại của phần tử bất kỳ trong Z_n^* bởi vậy là bội số chung nhỏ nhất của p-1 và q-1 hoặc $2p_1q_1$.

Cho α là phần tử có bậc $2p_1q_1$. Khi đó nhóm con cyclic của Z_n^* do α tạo ra là thiết lập thích hợp của bài toán logarit rời rạc.

Trong sơ đồ Girault, chỉ có TT biết được phân tích nhân tử của n.

Các giá trị n, α công khai, còn p, q, p_1 và q_1 là bí mật.

TT chọn số mũ công khai RSA, ký kiệu là **e**. Số mũ giải mã tương ứng bí mật là **d** (trong đó $d = e^{-1} \bmod \Phi(n)$).

Mỗi người sử dụng U có một định danh ID(U).

U nhận được khoá tự xác nhận công khai p_u từ TT như sau:

1. U chọn một số mũ bí mật a_u và tính:

   $b_u = \alpha^{a_u} \bmod n$.

2. U chuyển a_u và b_u tới TT.
3. TT tính:

   $p_u = (b_u - ID(U))^d \bmod n$.

4. TT chuyển p_u cho U.

Ở đây, U cần TT giúp đỡ để tạo p_u. Chú ý rằng, b_u có thể tính được từ p_u và ID(U) bằng thông tin công khai có sẵn.

$$b_u = p_u^e + ID(U) \bmod n.$$

Giao thức thoả thuận khoá Girault:

1. U chọn ngẫu nhiên, bí mật r_u và tính: $s_u = \alpha^{r_u} \bmod n$.
2. U gửi ID(U), p_u và s_u tới V.
3. V chọn ngẫu nhiên, bí mật r_v và tính: $s_v = \alpha^{r_v} \bmod n$.
4. V gửi ID(V), p_v và s_v tới U.
5. U tớnh khoỏ:

$$K = s_v^{a_u} (p_v^e + ID(V))^{r_u} \bmod n.$$

6. V tính khóa:

$$K = s_u^{a_v} (p_u^e + ID(U))^{r_v} \bmod n.$$

Thông tin được truyền trong giao thức như sau:

$$U \quad \begin{array}{l} \xrightarrow{\text{ID(U), } p_u \text{ , } \alpha^{r_u} \bmod n} \\ \xleftarrow{\text{ID(V), } p_v \text{ , } \alpha^{r_v} \bmod n} \end{array} \quad V$$

Cuối giao thức, U và V tính khoá:

$$K = \alpha^{r_u a_v + r_v a_u} \bmod n.$$

Ví dụ:

* Giả sử p = 839, q = 863. Khi đó n = p*q = 839*863 = 724057.
  Φ(n) = (p - 1)*(q - 1) = 838*862 = 722356.
  Giả sử TT chọn d =125777 làm số mũ giải mã RSA, thì e = 84453.
* Giả sử U có ID(U) = 500021 và a_u = 111899.
  U tính: $b_u = \alpha^{a_u} \bmod n = 5^{111899} \bmod 724057 = 488889$.
  Và $p_u = (b_u - ID(U))^d = 650704$.
  V có ID(V) = 500022 và a_v = 123456.
  V tính: $b_v = \alpha^{a_v} \bmod n = 5^{123456} \bmod 724057 = 111692$.
  Và $p_v = (b_v - ID(V))^d = 683556$.

∗ Bây giờ U và V muốn trao đổi khoá.

Giả sử U chọn $r_u = 5638$, tính:

$$s_u = 5^{5638} \bmod 724057 = 171007.$$

V chọn $r_v = 356935$, tính:

$$s_v = 5^{356935} \bmod 724057 = 320688.$$

∗ Khi đó cả U lẫn V sẽ tính cùng một khoá: K = 42869.

Sự an toàn của sơ đồ.

Xét cách các khóa tự xác thực chống lại một kiểu tấn công:

1. Vì các giá trị b_u , p_u và ID(U) không được TT ký nên không có cách nào để ai đó xác minh trực tiếp tính xác thực của chúng.

Giả thiết thông tin này bị W (người muốn giả danh U, tức là không hợp tác với TT để tạo nó). Nếu W bắt đầu bằng ID(U) và giá trị giả mạo b'_u. Khi đó không có cách nào để W tính được số mũ a'_u tương ứng với b'_u nếu bài toán logarit rời rạc khó giải.

Không có a'_u thì W không thể tính được khoá.

2. Nếu W hoạt động như kẻ xâm nhập giữa cuộc thì W sẽ có thể ngăn được U và V tính ra khoá chung, song W không thể đồng thời thực hiện các tính toán của U và V.

Như vậy, sơ đồ cho khả năng xác thực ngầm như giao thức MTI.

3. Theo sơ đồ trên, TT có thể tính p_u trực tiếp từ b_u mà không cần biết a_u, song điều quan trọng ở đây là TT sẽ được thuyết phục rằng, U biết a_u trước khi TT tính p_u cho U.

4. Sơ đồ có thể bị tấn công nếu TT phát bừa bãi các khoá công khai p_u cho những người sử dụng mà không kiểm tra trước xem họ có sở hữu các a_u tương ứng với các b_u của họ hay không.

Giả sử W chọn một giá trị giả mạo a'_u và tính giá trị tương ứng:

$$b'_u = \alpha^{a'_u} \bmod n.$$

Anh ta có thể xác định khoá công khai tương ứng:

$$p'_u = (b'_u - ID(U))^d \bmod n.$$

W sẽ tính: $b'_w = b'_u - ID(U) + ID(W)$.

và sau đó đưa b'_w và ID(W) tới TT.

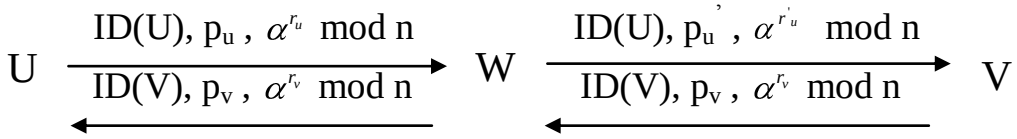
Giả sử TT phát ra khoá công khai cho W là:

$$p'_w = (b'_w - ID(W))^d \bmod n$$

Nhờ dùng yếu tố: $b'_w - ID(W) \equiv b'_u - ID(U) \pmod n$

Có thể suy ra rằng : $p'_w = p'_u$.

5. Giả sử U và V thực hiện giao thức còn W thay thế thông tin như sau:



V sẽ tính khoá:

$$K' = \alpha^{r'_u a_v + r_v a'_u} \bmod n.$$

U sẽ tính khoá:

$$K = \alpha^{r_u a_v + r_v a_u} \bmod n.$$

W có thể tính khoá:

$$K' = s_v^{a'_u} (p_v^e + ID(V))^{r'_u} \bmod n.$$

Như vậy, W và V chia sẻ nhau một khoá, song V nghĩ anh ta đang chia sẻ khoá với U. Như vậy, W sẽ có thể giải mã được các bức điện mà V gửi cho U.

CHƯƠNG 2 CÁC KHÁI NIỆM CƠ BẢN VỀ MÃ HÓA LƯỢNG TỬ

2.1 Ký hiệu Bra-Ket

Ký hiệu Bra-ket, được đưa ra bởi Paul Dirac (do vậy còn được gọi là ký hiệu Dirac). Ký hiệu Bra-ket là ký hiệu chuẩn được sử dụng rộng rãi trong vật lý lượng tử, dùng để mô tả trạng thái lượng tử trong lý thuyết cơ học lượng tử.

Trong cơ học lượng tử, trạng thái của một hệ vật lý được mô tả bởi một vector trong không gian Hilbert phức H (*sẽ nói rõ hơn ở phần sau*), mỗi vector đó được gọi là *ket* và ký hiệu là $|\psi\rangle$ (đọc là psi ket).

Ứng với mỗi *ket* $|\psi\rangle$ có một *bra* và ký hiệu là $\langle\psi|$ (đọc là psi bra) là ánh xạ tuyến tính liên tục từ không gian Hilbert phức H tới không gian số phức H được định nghĩa bởi $\langle\psi|\rho\rangle = (|\psi\rangle, |\rho\rangle)$ với mọi ket $|\rho\rangle$. Trong đó $(|\psi\rangle, |\rho\rangle)$ là tích vô hướng trong không gian Hilbert H. Trong ngôn ngữ ma trận, bra là ma trận chuyển vị liên hợp với ket và ngược lại.

Ký hiệu $\langle\psi|\rho\rangle$ được gọi là tích *bra-ket* (hay *bracket*).

Tính chất của bra-ket:

- Tính tuyến tính của bra và ket: với c_1 và c_2 là các số phức, ta có
  - $\langle\phi|\ c_1|\psi_1\rangle + c_2|\psi_2\rangle\ = c_1\langle\phi|\psi_1\rangle + c_2\langle\phi|\psi_2\rangle$
  - $c_1\langle\phi_1| + c_2\langle\phi_2|\ |\psi\rangle = c_1\langle\phi_1|\psi\rangle + c_2\langle\phi_2|\psi\rangle$
- Cho ket $|\psi_1\rangle$ và $|\psi_2\rangle$ bất kỳ, c_1 và c_2 là các số phức, từ tính chất của tích vô hướng ta có $c_1|\psi_1\rangle + c_2|\psi_2\rangle$ là vector đối ngẫu với $c_1^*\langle\psi_1| + c_2^*\langle\psi_2|$ trong đó c_1^* và c_2^* là các số phức liên hợp với c_1 và c_2
- Cho bra $\langle\phi|$ và ket $|\psi\rangle$ bất kỳ, từ tính chất của tích vô hướng trong không gian Hilbert ta có $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*$

2.2 Nguyên lý cơ bản của cơ học lượng tử

Trong cơ học cổ điển (hay cơ học Newton), trạng thái của một hệ n phần tử tại thời điểm t_0 được xác định bởi vị trí $\{x_1(t_0), x_2(t_0), \ldots, x_n(t_0)\}$ và vận tốc của hệ là đạo hàm bậc nhất của các phần tử $\{x_1'(t_0), x_2'(t_0), \ldots, x_n'(t_0)\}$. Nếu trạng thái khởi đầu được xác định, nhờ các định luật về cơ học cổ điển của Newton, chúng ta có thể hoàn toàn xác định (*ít nhất là về mặt nguyên lý*) trạng thái của hệ tại bất cứ thời điểm t nào.

Tuy nhiên, cơ học lượng tử hoàn toàn dựa trên một nền tảng toán học hoàn toàn khác so với cơ học cổ điển. Dưới đây, tôi sẽ đề cập đến một số tiên đề cơ sở của cơ học lượng tử.

Tiên đề 1. *Trạng thái của một hệ vật lý S được mô tả bằng một vector đơn vị $|\psi\rangle$, được gọi là vector trạng thái hoặc hàm sóng, nằm trong một không gian Hilbert H_S gắn liền với hệ vật lý.*

Sự tiến triển theo thời gian của vector trạng thái $|\psi\rangle$ của hệ tuân theo phương trình Schrödinger

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$$

trong đó H là toán tử tự liên hợp của hệ thống (còn gọi là toán tử Hamiltonian) và ħ là hằng số Planck-Dirac (hay còn gọi là hằng số Planck đơn giản), $\hbar=h/2\pi$, với h là hằng số Planck. Giá trị của $h \approx 6.626\text{x}10^{-34}$ J.s (J là Joule và s là giây) được xác định bằng thực nghiệm.

Như ta thấy ở trên, phương trình Schrödinger là phương trình vi phân tuyến tính bậc nhất. Do đó ta có thể áp dụng nguyên lý siêu trạng thái (*Superposition principle*): Nếu $|\psi_1(t)\rangle$ và $|\psi_2(t)\rangle$ là nghiệm của phương trình Schrödinger, khi đó siêu trạng thái $|\psi(t)\rangle = \alpha|\psi_1(t)\rangle + \beta|\psi_2(t)\rangle$ (với α β là số phức) cũng là nghiệm của phương trình. Do vậy toán tử phát triển theo thời gian của hệ sẽ là:

$|\psi(t)\rangle = U(t, t_0)\,|\psi(t_0)\rangle$

và U là toán tử tuyến tính.

U sẽ là toán tử Unita. Chính vì vậy trong mô hình toán học cho cơ học lượng tử, chúng ta sẽ sử dụng các phép biến đổi Unita.

Tiên đề 2. *Trạng thái của một hệ vật lý S được mô tả bằng một vector đơn vị $|\psi\rangle$, được gọi là vector trạng thái hoặc hàm sóng, nằm trong một không gian Hilbert H_S gắn liền với hệ vật lý.*

Nguyên lý bất định Heisenberg. *Cho A và B là hai toán tử Hermiltian gắn liền với các quan sát, A B là hai toán tử không giao hoán và $|\psi\rangle$ là hàm sóng gắn liền với trạng thái lượng tử. Khi đó bất đẳng thức sau luôn thoả mãn:*

$$\Delta A \Delta B \geq \frac{\left|\left\langle \psi \right|\ A, B\ \left|\psi\right\rangle\right|}{2}$$

trong đó $[A,B] = AB - BA$

Nguyên lý Heisenberg cho ta biết rằng khi đo một trạng thái lượng tử theo hai đại lượng (*như vị trí và vận tốc của hạt cơ bản*), ta không thể đo chính xác được đồng thời cả hai giá trị đó. Nguyên lý Heisenberg được ứng dụng trong các hệ phân phối khoá lượng tử như BB84 mà chúng ta sẽ xem xét ở phần sau.

2.3 Qubit và thanh ghi lượng tử

2.3.1 Khái niệm Qubit

Trước hết ta xét cách quan niệm mới về bit - đơn vị thông tin cơ bản trong mô hình mới này: đó là qubit.

Như ta đã biết: 1 bit cổ điển có thể biểu diễn một trong hai trạng thái: 0 hoặc 1 (ở tại một thời điểm xác định). Do đó, *n* bit có thể biểu diễn 2^n trạng thái khác nhau. Nhưng theo cách quan niệm cổ điển, nếu một thanh ghi được tạo nên từ *n* bit cổ điển, tại một thời điểm, nó chỉ có thể biểu đúng một giá trị nguyên trong khoảng từ $0 \to 2^n - 1$.

Theo quan niệm mới về mô hình tính toán lượng tử dựa trên nền tảng vật lý lượng tử, chúng ta thấy rằng tại một thời điểm một thanh ghi lượng tử có thể chứa được tổ hợp nhiều giá trị.

Xét theo mô hình vật lý, qubit là một vi hạt có hai trạng thái, nó có thể là: spin hạt nhân trong phân tử, ion bị bẫy (trapped ions), …. Chúng ta quan tâm đến hai trạng thái đặc biệt được ký hiệu là $|0\rangle$ & $|1\rangle$, được coi là hai trạng thái cơ sở tính toán.

Theo mô hình toán học, xét không gian Hilbert H^2 (H là trường số phức). Nó có cơ sở trực giao là (1, 0) và (0, 1), ta ký hiệu tương ứng là $|0\rangle$ & $|1\rangle$. Qubit cơ sở bao gồm hai dạng $|0\rangle$ hoặc $|1\rangle$. Khi đó, một 1-qubit tổng quát biểu diễn một vector đơn vị trong không gianH^2, trong đó trạng thái $|0\rangle$ ứng với vector (1, 0), còn trạng thái $|1\rangle$ sẽ ứng với vector (0, 1) đồng thời thoả mãn điều kiện chuẩn hoá về xác suất. Như vậy, dạng tổng quát của một 1-qubit là:

$$\begin{cases} \alpha|0\rangle + \beta|1\rangle \\ \alpha, \beta \in \square \end{cases}$$

Đối với qubit có trạng thái tổng quát là $\alpha|0\rangle + \beta|1\rangle$, chúng ta có thể tiến hành đo (*sẽ nói rõ hơn ở phần sau*) trạng thái của qubit. Khi đó, theo các nguyên lý của cơ học lượng tử thì xác suất để nhận được trạng thái $|0\rangle$ là α^2, xác suất để nhận được trạng thái $|1\rangle$ là β^2, do đó α, β phải thoả mãn điều kiện xác suất $\alpha^2 + \beta^2 = 1$.

Như vậy sử dụng $|0\rangle$ & $|1\rangle$ ta có thể biểu diễn trạng thái của một qubit, cũng giống như 0 & 1 biểu diễn trạng thái của bit cổ điển.

Để đi đến khái niệm thanh ghi lượng tử (*quantum register*), người ta mở rộng không gian trạng thái bằng cách sử dụng *tích tensor* của các không gian H^2.

2.3.2 Khái niệm thanh ghi lượng tử

Định nghĩa: Một thanh ghi lượng tử (*quantum register*) biểu diễn một vector trong không gian Hilbert $H = ((H^2)^{\otimes n})$, đã được chuẩn hoá. $(H^2)^{\otimes n}$ là tích Tensor của *n* không gian H^2)

Do cơ sở của $(H^2)^{\otimes n}$ là:

$$|0\rangle \otimes |0\rangle \otimes ... \otimes |0\rangle \leftrightarrow i_0 = |00...0\rangle \leftrightarrow i_0 = |0\rangle$$
$$|0\rangle \otimes |0\rangle \otimes ... \otimes |1\rangle \leftrightarrow i_1 = |00...1\rangle \leftrightarrow i_1 = |1\rangle$$
$$.....$$
$$|1\rangle \otimes |1\rangle \otimes ... \otimes |1\rangle \leftrightarrow i_n = |11...1\rangle \leftrightarrow i_n = |2^n - 1\rangle$$

nên trạng thái tổng quát của một thanh ghi *n*-qubit $|X\rangle$ có dạng:

$$\begin{cases} |X\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \\ c_i \in \mathbb{C}; \forall i = \overline{0, 2^n - 1} \\ \sum_{i=0}^{2^n-1} |c_i|^2 = 1 \end{cases}$$

$|X\rangle$ có toạ độ trong không gian Hilbert H là (c_0, c_1, c_2, ..., c_n)

Trạng thái lượng tử được biểu diễn một thanh ghi được gọi là một *siêu trạng thái* (Superposition). Ta cũng thấy rằng một quantum register có thể lưu trữ đồng thời 2^n thông tin khác nhau: $0 \to 2^n - 1$.

Tồn tại siêu trạng thái của thanh ghi có thể mô tả bởi:

$|X\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$; trong đó $|i_j\rangle$ là qubit thứ j.

Tuy nhiên cũng có những siêu trạng thái không thể biểu diễn được dưới dạng như vậy.

Ta xét hai ví dụ với thanh ghi 2-qubit có thể biểu diễn được bằng tích tensor của hai 1-qubit:

Ví dụ:

$$|X\rangle = \frac{1}{\sqrt{2}}\ |0\rangle + |2\rangle\ = \frac{1}{\sqrt{2}}\ |00\rangle + |10\rangle\ = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle = |q_1\rangle \otimes |q_2\rangle$$

Như vậy, trạng thái của $|X\rangle$ được viết dưới dạng tích của trạng thái hệ thống con: $|q_1\rangle = \frac{1}{\sqrt{2}}\ |0\rangle + |1\rangle$ và $|q_2\rangle = |0\rangle$. Với những trạng thái như thế này, các phép biến đổi Unita, các phép đo chỉ làm thay đổi trạng thái của hệ thống con mà không làm ảnh hưởng đến các hệ thống còn lại. Ví dụ, khi tiến hành đo qubit thứ nhất được giá trị $|0\rangle$ hay $|1\rangle$ thì qubit thứ hai luôn đo được kết quả $|0\rangle$. Có thể so sánh với trạng thái rối lượng tử ở mục sau.

2.3.3 Phép biến đổi Unita và phép đo.

Đối với tính toán lượng tử, có 2 loại phép biến đổi cơ bản là phép biến đổi Unita và phép biến đổi không Unita. Đối với lớp phép biến đổi không Unita chỉ có phép đo.

Các phép biến đổi Unita là các phép biến đổi không mất năng lượng. Do vậy các phép biến đổi Unita là các phép biến đổi khả nghịch. Về mặt toán học có thể coi là các ánh xạ trong các không gian Hilbert đẳng cấu.

U:∑H→∑H'

trong đó H và H’ là hai không gian Hilbert có cùng số chiều (ở đây chúng ta chỉ xét đến không gian Hilbert hữu hạn chiều, với các không gian Hilbert vô hạn chiều, sẽ có cách tiếp cận khác không được đề cập đến trong luận văn này)

Còn phép đo là phép biến đổi mất năng lượng, do đó phép đo là phép biến đổi bất khả nghịch. Về mặt toán học có thể coi là phép đo là phép ánh xạ về không gian Hilbert có số chiều ít hơn.

U:∑H→∑H'

trong đó H và H’ là hai không gian Hilbert, H’ có số chiều nhỏ hơn H.

Đối với hệ lượng tử, khi áp dụng phép đo thì ta sẽ không thể tiên đoán độ xác định của kết quả (nguyên lý bất định Heisenberg). Kết quả thu được phụ thuộc vào xác suất của các trạng thái được biểu diễn bởi hệ lượng tử. Đồng thời theo các nguyên lý của cơ học lượng tử, ngay sau khi đo lập tức hệ lượng tử sẽ sụp đổ về giá trị đo được.

Ví dụ: Trong trường hợp tổng quát, một *n*-qubit $|X\rangle$ đang ở trạng thái lượng tử:

$$\begin{cases} |X\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \\ c_i \in \mathbb{C}; \forall i = \overline{0, 2^n - 1} \\ \sum_{i=0}^{2^n-1} |c_i|^2 = 1 \end{cases}$$

Khi tiến hành phép đo, chúng ta sẽ không biết trước kết quả đo được là bao nhiêu. Theo các nguyên lý của cơ học lượng tử, chúng ta chỉ có thể biết được xác suất đo được giá trị i là c_i^2. Đồng thời ngay sau khi tiến hành đo, $|X\rangle$ sẽ không còn ở siêu trạng thái $\sum_{i=0}^{2^n-1} c_i |i\rangle$ mà sụp đổ về trạng thái $|i\rangle$ đo được.

Ví dụ với hệ lượng tử $|q\rangle = \frac{1}{\sqrt{2}}\ |0\rangle + |1\rangle\ = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, khi tiến hành phép đo, chúng ta sẽ không xác định được kết quả là 0 hay 1 mà chỉ có thể biết được rằng khi đo, chúng ta sẽ thu được kết quả là 0 hay 1 với xác suất bằng nhau (là 50%). Đồng thời, ngay sau khi đo, chẳng hạn ta đo được giá trị 0 thì ngay lập tức $|q\rangle$ sẽ sụp đổ về trạng thái $|0\rangle$.

2.4 Nguyên lý rối lượng tử (Nguyên lý Entanglement)

Nguyên lý rối lượng tử là một trong những nguyên lý quan trọng của tính toán lượng tử. Nguyên lý rối lượng tử cho phép việc tính toán diễn ra một cách đồng thời trên các thành phần của qubit đầu vào khi nó ở trạng thái rối lượng tử.

Ví dụ : Ta xét ví dụ sau đây: $|X\rangle = \frac{1}{\sqrt{2}}\ |0\rangle + |3\rangle\ = \frac{1}{\sqrt{2}}\ |00\rangle + |11\rangle$

Khi tiến hành đo một qubit, tuỳ theo kết quả của phép đo mà ta có ngay trạng thái của qubit còn lại. Tức là phép đo đã ảnh hưởng đến toàn bộ hệ thống:

Nếu kết quả là $|0\rangle$ thì trạng thái qubit còn lại là $|0\rangle$

Nếu kết quả là $|1\rangle$ thì trạng thái qubit còn lại là $|1\rangle$

Suy ra: giữa hai hệ thống con có mối quan hệ nào đó. Người ta gọi những trạng thái như vậy là *rối lượng tử* hay *vướng lượng tử* (*Quantum Entanglement*). Trạng thái này của hệ 2-qubit không thể phân tích thành tích tensor của hai hệ thống con 1-qubit.

2.5 Nguyên lý song song lượng tử

Thanh ghi lượng tử cùng một lúc có thể lưu trữ nhiều trạng thái đơn lẻ khác nhau nhưng có một đặc điểm đáng chú ý là: bất kỳ một phép tác động nào lên một thanh ghi lượng tử thì nó sẽ tác động lên đồng thời toàn bộ các trạng thái mà thanh ghi đó lưu trữ (ta không thể tách rời các trạng thái để thao tác trên chúng một cách riêng lẻ)

Nghĩa là $U\sum_{i=0}^{2^n-1} c_i |i\rangle = \sum_{i=0}^{2^n-1} c_i U|i\rangle$ với U là một phép biến đổi Unita nào đó. Ở đây ta có thể thấy sức mạnh của tính toán lượng tử vì nếu trong tính toán cổ điển để thực hiện được phép biến đổi trên, chúng ta cần nhân ma trận ma trận C = (c_0, c_1, c_2, ..., c_n) với ma trận U cỡ 2^n x 2^n còn trong tính toán lượng tử, chúng ta chỉ cần một phép biến đổi Unita (*có thể biểu diễn bằng một cổng lượng tử, xem 1.7*). Tức là độ phức tạp có thể giảm theo cấp luỹ thừa.

2.6 Nguyên lý không thể sao chép (No-Cloning Theorem)

Trong tính toán cổ điển, có một tính chất của bit cổ điển là chúng ta có thể dễ dàng tạo một bản sao chứa cùng thông tin. Tuy nhiên, đối với tính toán lượng tử, trạng thái của qubit tổng quát nói chung không thể sao chép.

Định lý: *Không thể tạo ra một máy thực hiện các phép biến đổi Unita có khả năng sao chép hoàn hảo trạng thái của một qubit bất kỳ.*

Chứng minh:

Thực vậy, giả sử ta có được một máy sao chép hoàn hảo. Khi đó xét hệ bao gồm hai qubit (qubit được sao chép, qubit sao chép) và máy sao chép trạng thái.

Qubit được sao chép ở trạng thái tổng quát:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Trong đó biên độ α và β là các số phức ràng buộc bởi phương trình

$$\alpha^2 + \beta^2 = 1$$

Trong khi đó qubit thứ hai và máy sao chép đang ở trạng thái φ và A_i (trạng thái khởi đầu trước khi tiến hành sao chép).

Khi đó máy sao chép sẽ tác động phép biến đổi Unita U lên hệ:

$$U\ |\psi\rangle|\varphi\rangle|A_i\rangle = |\psi\rangle|\varphi\rangle|A_{f\psi}\rangle = \alpha|0\rangle+\beta|1\rangle\quad \alpha|0\rangle+\beta|1\rangle\ |A_{f\psi}\rangle \quad (1.6.1)$$

trong đó trạng thái cuối cùng của máy sao chép $|A_{f\psi}\rangle$ sẽ phụ thuộc vào trạng thái của qubit thứ nhất $|\psi\rangle$.

Chúng ta sẽ chứng minh phép biến đổi Unita trên không thể tồn tại.

Thực vậy, nếu trạng thái của qubit thứ nhất là $|0\rangle$, phép biến đổi Unita *U* sẽ tác động là: $U\ |0\rangle|\varphi\rangle|A_i\rangle = |0\rangle|0\rangle|A_{f0}\rangle$ (1.6.2)

Tương tự, nếu trạng thái của qubit thứ nhất là $|1\rangle$, phép biến đổi Unita *U* sẽ tác động là: $U\ |1\rangle|\varphi\rangle|A_i\rangle = |1\rangle|1\rangle|A_{f1}\rangle$ (1.6.3)

Khi đó, với trạng thái tổng quát của qubit thứ nhất và do tính tuyến tính của toán tử Unita *U*, ta có

$$\begin{aligned} U\ |\psi\rangle|\varphi\rangle|A_i\rangle &= U\quad \alpha|0\rangle+\beta|1\rangle\ |\varphi\rangle|A_i\rangle \\ &= \alpha U\ |0\rangle|\varphi\rangle|A_i\rangle + \beta U\ |1\rangle|\varphi\rangle|A_i\rangle \\ &= \alpha|0\rangle|0\rangle|A_{f0}\rangle+\beta|1\rangle|1\rangle|A_{f1}\rangle \end{aligned} \qquad (1.6.4)$$

Ta thấy trạng thái (1.6.4) này khác hoàn toàn so với trạng thái (1.6.1) chúng ta mong muốn, suy ra điều cần phải chứng minh.

Ở đây, định lý này muốn khẳng định không tồn tại một máy sao chép trạng thái bất kỳ, tuy nhiên với một số trạng thái lượng tử đặc biệt như $|0\rangle$ hay $|1\rangle$ thì ta có thể tạo được máy sao chép.

2.7 Mạch và Cổng logic lượng tử

Trong mô hình máy tính cổ điển, các nhà khoa học đã mô hình hoá toán học bằng các mô hình như cổng và mạch logic cổ điển, máy tính Turing. Tương tự vậy, trong tính toán lượng tử, các nhà khoa học cũng được xây dựng các mô hình như mô hình cổng và mạch logic lượng tử, máy tính Turing lượng tử. Yao đã chỉ ra rằng với mỗi máy Turing lượng tử, tồn tại mô hình mạch logic lượng tử mô phỏng máy Turing lượng tử đó với thời gian đa thức. Do đó chúng ta chỉ cần nghiên cứu một mô hình cổng và mạch logic lượng tử, do mô hình này đơn giản

và gần gũi với cách thiết kế máy tính lượng tử. Từ đó dẫn đến kết quả tương tự trong mô hình máy Turing lượng tử.

Một cách tương tự như máy tính cổ điển, được xây dựng dựa trên các cổng logic cơ bản như AND, OR, NOT, … trong mô hình tính toán lượng tử, chúng ta cũng xây dựng các cổng lượng tử.

Do yêu cầu của cơ học lượng tử là các phép biến đổi hệ lượng tử bắt buộc là Unita do đó trong mô hình toán học của tính toán lượng tử, chúng ta sử dụng các toán tử Unita.

Định nghĩa: *Một cổng logic lượng tử n-qubit được sử dụng để biến đổi n-qubit được biểu về mặt toán học bởi một phép biến đổi Unita tác động lên vector siêu trạng thái của n-qubit đó. Ma trận biến đổi Unita tác động lên n-qubit là ma trận* $2^n x 2^n$.

Định nghĩa: *Mạch lôgic lượng tử là một tập các cổng lôgic lượng tử liên kết theo một đồ thị có hướng không chu trình, trong đó đầu ra của cổng này có thể là đầu vào của cổng kia.*

Dưới đây là một số cổng logic lượng tử cơ bản

2.7.1 Cổng 1 qubit

Các cổng logic lượng tử tác động lên 1 qubit đều có thể mô tả bằng ma trận Unita tổng quát sau:

$$U2(\theta,\delta,\sigma,\tau)=\begin{pmatrix} e^{i(\delta+\sigma+\tau)}cos(\frac{\theta}{2}) & e^{-i(\delta+\sigma-\tau)}sin(\frac{\theta}{2}) \\ -e^{i(\delta-\sigma+\tau)}sin(\frac{\theta}{2}) & e^{i(\delta-\sigma-\tau)}cos(\frac{\theta}{2}) \end{pmatrix}$$

với $\theta,\delta,\sigma,\tau \in R$

i/ NOT:

Thực hiện:

$|0\rangle \to |1\rangle$
$|1\rangle \to |0\rangle$

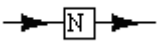
Dạng ma trận:

$$X=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Cổng NOT có thể biểu diễn bằng cổng U2 là

$$X=U2(\pi,0,0,0)=\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Biểu diễn trong mạch:



Hình 1.1. Biểu diễn cổng NOT

ii/ Z-gate:

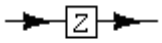
Thực hiện:

$|0\rangle \to |0\rangle$
$|1\rangle \to -|1\rangle$

Dạng ma trận:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Biểu diễn trong mạch:



Hình 1.2. Biểu diễn cổng Z

iii/ Cổng Hadamard

Thực hiện:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}} |0\rangle + |1\rangle$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}} |0\rangle - |1\rangle$$

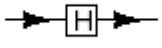
Dạng ma trận

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Cổng Hadamard có thể biểu diễn bằng cổng U2 là

$$H = U2(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}, \pi) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Biểu diễn trong mạch:



Hình 1.3. Biểu diễn cổng Hadamard

2.7.2 Cổng 2 qubit

i/ Controlled NOT (CNOT)

Cổng CNOT này thực hiện tương tự phép toán XOR trong tính toán cổ điển.

Thực hiện:

$a,b \in 0,1$

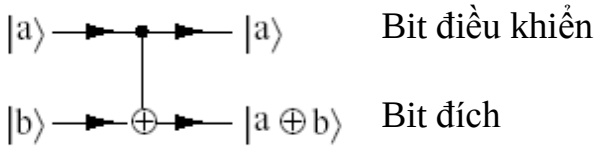
b giữ nguyên nếu a = 0

b đảo bit nếu a = 1

Dạng ma trận

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

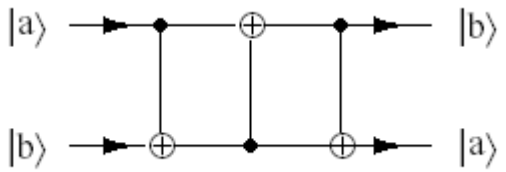
Biểu diễn trong mạch:



Hình 1.4. Biểu diễn cổng CNOT

ii/ Cổng hoán vị hai bit (Cổng Swap)

Ứng dụng cổng CNOT: hoán vị hai bit



Hình 1.5. Biểu diễn cổng Swap

iii/ Cổng dịch pha có điều khiển (Controlled phase shift gate)

Cổng dịch pha có điều khiển là cổng không có cổng tương tự trong tính toán cổ điển.

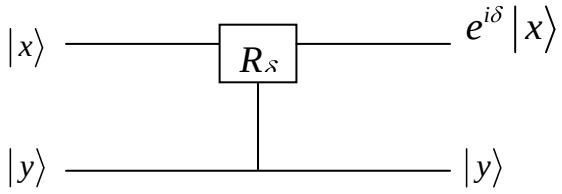
Thực hiện:

Cổng dịch pha sẽ dịch pha của qubit nguồn chỉ khi qubit điều khiển bằng 1

Dạng ma trận:

$$CPHASE(\delta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\delta} \end{pmatrix}$$

Biểu diễn trong mạch:



Hình 1.6. Biểu diễn cổng dịch pha có điều khiển2.7.3. Cổng 3 qubit

i/ Cổng Controlled-Controlled NOT (CCNOT) - Cổng Toffoli

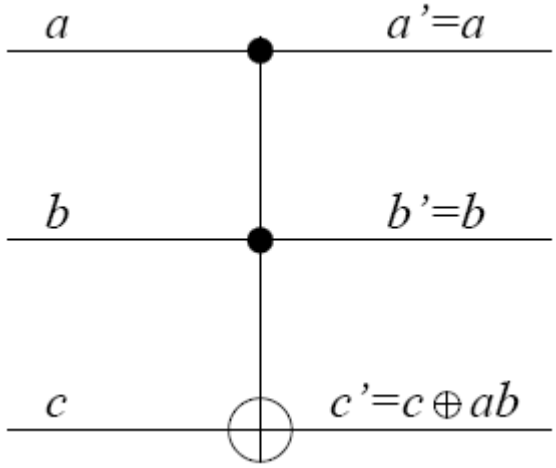
Thực hiện

a	***b***	***c***	***a'***	***b'***	***c'***
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Dạng ma trận:

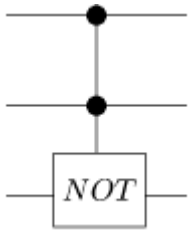
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Biểu diễn trong mạch:



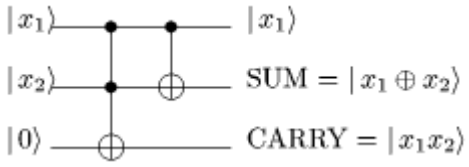
Hình 1.7. Biểu diễn cổng Toffoli

Cổng Toffoli áp dụng cổng Not lên qubit cuối cùng nên ta có thể biểu diễn dưới dạng mạch như sau:



Hình 1.8. Biểu diễn cổng Toffoli

Chúng ta có thể sử dụng cổng Toffoli và cổng CNOT để biểu diễn mạch cộng nhị phân có nhớ như sau:



Hình 1.9. Biểu diễn cổng cộng nhị phân có nhớ2.7.4. Cổng phổ dụng

Trên thực tế, chúng ta muốn xây dựng và sử dụng chỉ một số cổng cơ bản tác động lên một số lượng nhỏ qubit (*độc lập với số lượng n qubit cần xử lý, n có thể rất lớn*) nhưng phải đảm bảo yêu cầu có thể tính được một hàm bất kỳ. Các cổng đó được gọi là *cổng phổ dụng*. Trong tính toán cổ điển, ta đã biết được

cổng NOT và AND là cổng phổ dụng. Tương tự, trong tính toán lượng tử, chúng ta sẽ định nghĩa về cổng phổ dụng.

Định nghĩa: *Một tập cổng lượng tử G được gọi là phổ dụng nếu* $\forall \varepsilon > 0$ *và mọi ma trận Unita U tác động trên số bít bất kì, U có thể được xấp xỉ với độ chính xác* $\varepsilon > 0$ *bằng một dãy cổng của G. Nói cách khác nhóm con tạo nên bởi G là trù mật trong nhóm nhóm các toán tử Unita,* $\forall n$.

Tức là $\forall U, \forall \varepsilon > 0, \exists U'$ *được tạo nên bằng tích các cổng của G sao cho* $\|U - U'\| \leq \varepsilon$.

Deutsch là người đầu tiên chỉ ra một cổng lượng tử phổ dụng 3 qubit. Đó là cổng Toffoli dạng tổng quát. Sau đó, Di Vincenzo chỉ ra những cổng 2 qubit là phổ dụng. Cho đến nay, có rất nhiều tập cổng phổ dụng đã được đưa ra. Một ví dụ về tập cổng phổ dụng là tập cổng CNOT, Hadamard, các cổng dịch pha.

CHƯƠNG 3. MÃ HÓA LƯỢNG TỬ

Hệ mã hoá công khai lần đầu được đưa ra bởi Diffie và Hellman trong bài báo có tên "*New directions in cryptography*" vào năm 1976. Sau đó 2 năm, vào năm 1978, Rivest, Shamir và Adleman công bố một hệ mã công khai (*do đó được mang tên RSA*) dựa trên bài toán khó là phân tích ra thừa số nguyên tố của số lớn trong bài báo "*A method for obtaining digital signatures and public-key cryptosystems*". Ngày nay, hệ mã công khai RSA và các biến thể được sử dụng rộng rãi trong các ứng dụng thương mại và dân sự.

Tuy nhiên, như chúng ta đã thấy ở trên, với sức mạnh của tính toán lượng tử và thuật toán thích hợp, chúng ta có thể phá vỡ các thuật toán mã hoá được coi là an toàn hiện nay như RSA. Do vậy nhu cầu cấp thiết được đặt ra là thiết kế phương pháp mã hoá và phân phối khoá mới để đảm bảo an toàn dữ liệu khi máy tính lượng tử ra đời, đặc biệt là khi máy tính lượng tử được thương mại hoá.

Như vậy, vấn đề đặt ra là chúng ta phải sử dụng sức mạnh của lượng tử để chống lại các phương pháp tấn công bằng lượng tử. Trong đó, sức mạnh của tính toán lượng tử có thể kết hợp với một hệ mã có độ mật hoàn thiện, hệ mật One-time-pad.

Trong bài báo của mình vào năm 1949, Shannon đã chứng minh hệ mật một lần đệm (*One-time-pad*) là hệ mật có độ mật hoàn thiện dựa trên lý thuyết thông tin. Tuy nhiên trên thực tế, hệ mật One-time-pad không được sử dụng rộng rãi do nhược điểm của hệ mật One-time-pad là yêu cầu không gian khoá lớn (tối thiểu bằng không gian bản rõ) dẫn đến nhiều khó khăn trong quá trình phân phối khoá, bảo mật và lưu trữ khoá. Nhưng khi kết hợp với tính toán lượng tử, hệ mã One-time-pad lại cho thấy tiềm năng to lớn về độ an toàn bảo mật của dữ liệu cũng như tính hiệu quả trong việc phân phối và lưu trữ khoá.

Mục đích của chương này là đề cập đến một ví dụ về một giao thức phân phối khoá lượng tử đơn giản trong bức tranh hết sức sôi động của lĩnh vực mã

hoá lượng tử và thám mã lượng tử. Giao thức phân phối khoá lượng tử này có thể sử dụng để phân phối khoá của hệ mật One-time-pad.

3.1 Giao thức phân phối khoá lượng tử BB84

Giao thức phân phối khoá lượng tử BB84 là giao thức lượng tử đầu tiên được Bennett và Brassard công bố trong bài báo "*Quantum cryptography: Public key distribution and coin tossing*" vào năm 1984 và cài đặt lần đầu vào năm 1992 [21]. Trên thực tế giao thức BB84 đã được cài đặt chuyển thông tin trên khoảng cách 10 km vào năm 1994 qua cáp quang [58] và hi vọng có thể chuyển thông tin ở khoảng cách xa hơn [55].

3.1.1 Giao thức BB84 trường hợp không nhiễu

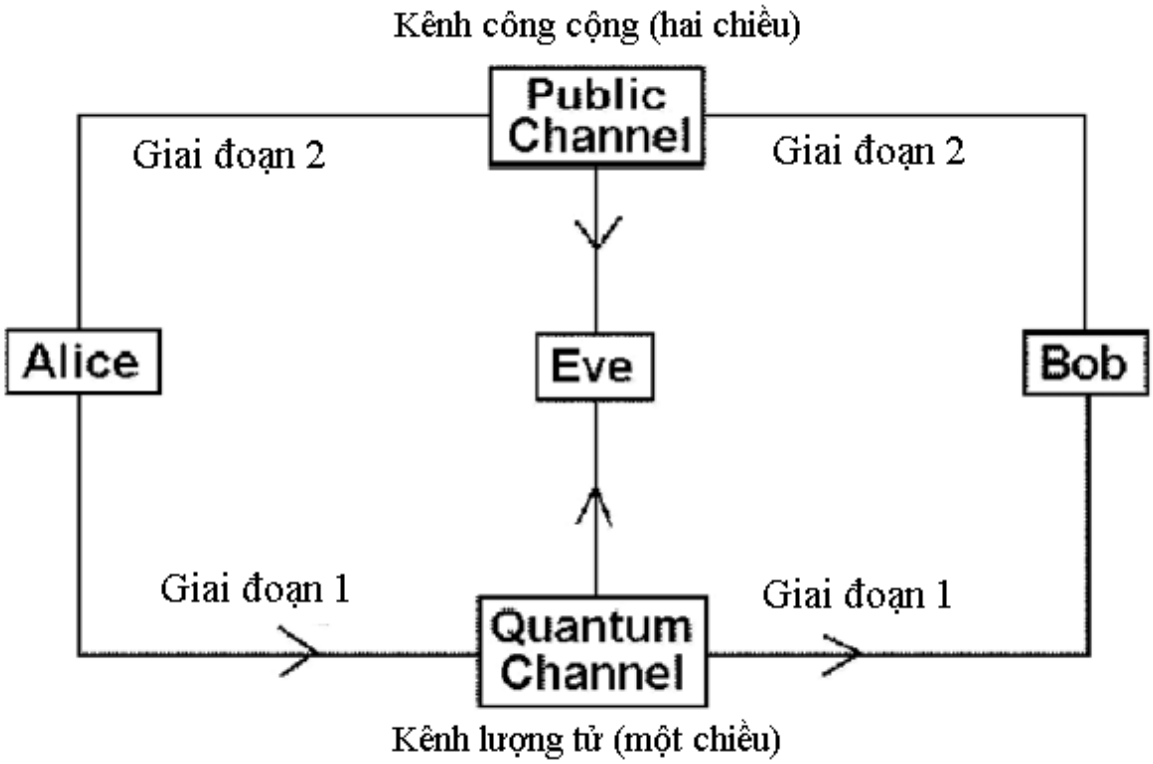
Trong giao thức phân phối khoá lượng tử BB84, chúng ta cần chuẩn bị 4 trạng thái lượng tử $|0\rangle$, $|1\rangle$, $|+\rangle \equiv |0\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ và $|-\rangle \equiv |1\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ chia làm hai nhóm (mà chúng ta sẽ gọi là hai bảng chữ cái):

- Bảng chữ cái z gồm hai trạng thái $|0\rangle$, $|1\rangle$
- Bảng chữ cái x gồm hai trạng thái $|+\rangle \equiv |0\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ và $|-\rangle \equiv |1\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Giao thức BB84 sẽ bao gồm hai giai đoạn:

- Giai đoạn 1: Giao tiếp qua kênh lượng tử\
- Giai đoạn 2: Giao tiếp qua kênh công cộng (gồm 2 pha)

Sơ đồ của giao thức BB84 có thể mô tả bởi hình sau: trong đó Alice và Bob muốn giao tiếp với nhau, còn Eve là kẻ thứ ba muốn đọc trộm thông tin trao đổi giữa Alice và Bob.



Hình 3.1. Sơ đồ của giao thức BB84

Ta sẽ xem xét từng giai đoạn.

3.1.1.1 Giai đoạn 1: Giao tiếp qua kênh lượng tử

Trong giai đoạn này, Alice sẽ truyền trên kênh lượng tử với xác suất ngẫu nhiên bằng nhau các trạng thái của bảng chữ cái z và x. Vì không có toán tử đo trạng thái của bảng chữ cái z hoán vị với toán tử đo trạng thái của bảng chữ x [55] nên theo nguyên lý bất định Heisenberg, không ai, kể cả Bob và Eve có thể nhận được các bit truyền đi từ Alice với xác xuất lớn hơn ¾ (75%).

Điều này có thể chứng minh như sau: vì không ai có thể chọn toán tử đo chính xác cả hai bảng chữ cái z và x đồng thời do Alice chọn ngẫu nhiên nên việc chọn toán tử đo sẽ đúng với xác suất 50% (do có 2 bảng chữ cái). Đồng thời nếu chọn sai bảng chữ cái, xác suất để đo đúng là 50% (do có 2 trạng thái trong 1 bảng chữ cái). Vì vậy xác suất đoán đúng trạng thái Alice truyền đi là

$$P = \frac{1}{2} * 1 + \frac{1}{2} * \frac{1}{2} = \frac{3}{4}$$

Xác suất để Bob đoán sai là 1-P = ¼

Ngoài ra, theo nguyên lý không thể sao chép (*no-clone theorem*), Eve không thể đo thông tin của Alice gửi đi rồi sao chép lại để gửi chuyển tiếp cho Bob.

Trong trường hợp nếu có Eve thực hiện toán tử đo các bit do Alice truyền đi với xác suất λ, $0 \leq \lambda \leq 1$, và không thực hiện các toán tử đo với xác suất 1 - λ Bởi vì Bob và Eve lựa chọn các phép toán đo hoàn toàn ngẫu nhiên độc lập với nhau và độc lập với sự lựa chọn của Alice nên khi Eve thực hiện phép đo ở giữa đường truyền sẽ ảnh hưởng đến bit lượng tử nhận được của Bob. Việc này sẽ làm cho tỷ lệ lỗi của Bob nhận được từ ¼ thành:

$$P_{Error} = \frac{1}{4}(1-\lambda) + \frac{3}{8}\lambda = \frac{1}{4} + \frac{\lambda}{8}$$

Do vậy nếu Eve "*đọc lén*" tất cả các bit do Alice truyền đến cho Bob (nghĩa là λ=1) thì xác suất lỗi của Bob là 3/8 (tăng gần 50%)

3.1.1.2 Giai đoạn 2: Giao tiếp qua kênh công cộng

Giai đoạn này sẽ làm hai pha:

- Pha 1: Tạo khoá thô
- Pha 2: Phát hiện sự do thám của Eve thông qua phát hiện lỗi

Pha 1. Tạo khoá thô.

Pha 1 của giai đoạn 2 là pha loại bỏ vị trí các bit lỗi (và các bit tại vị trí đấy). Các lỗi này bao gồm lỗi xác suất chọn cơ sở đo (bằng ¼) và lỗi do Eve đo trên đường truyền.

Bob truyền trên đường truyền công cộng tên bảng chữ cái mình sử dụng để đo (z và x) cho Alice. Khi đó Alice sẽ thông báo cho Bob biết bảng chữ cái nào đúng. Sau khi hai bên trao đổi thông tin, Alice và Bob sẽ cùng xoá đi các bit tương ứng tại các vị trí không tương ứng giữa bảng chữ mà Alice chọn để truyền thông tin và bảng chữ cái mà Bob dùng để đo. Các bit còn lại của Alice và Bob được gọi là khoá thô.

Nếu không có sự do thám của Eve, khi đó khoá thô của Alice và khoá thô của Bob là giống nhau và có thể sử dụng làm khoá bí mật. Tuy nhiên do có sự do thám của Eve nên xác suất để khoá thô của Alice và Bob không giống nhau là

$$0.(1-\lambda)+\frac{1}{4}.\lambda=\frac{\lambda}{4}$$

Pha 2. Phát hiện sự do thám của Eve thông qua phát hiện lỗi.

Với giả thiết ban đầu là không có nhiễu trên đường truyền do vậy mọi sự khác nhau giữa khoá thô của Alice và Bob đều chứng tỏ là do sự do thám của Eve. Vì vậy để phát hiện sự do thám của Eve, Alice và Bob cùng chọn thoả thuận công khai dựa trên tập con ngẫu nhiên *m* bit của khoá thô, và so sánh công khai các bit tương ứng, đảm bảo rằng không có sự sai khác nào giữa hai tập con ngẫu nhiên đó.

Nếu có bất cứ sự sai khác nào giữa hai tập con ngẫu nhiên, nguyên nhân chắc chắn do sự do thám của Eve. Khi đó Alice và Bob quay trở lại từ giai đoạn 1 để bắt đầu lại. Nếu không có sự sai khác nào, xác suất để Eve thoát khỏi sự kiểm tra trên là

$$P_{false}=\left(1-\frac{\lambda}{4}\right)^{m}$$

trong trường hợp λ=1 và m = 200, khi đó

$$P_{false}=\left(1-\frac{1}{4}\right)^{200}=\left(\frac{3}{4}\right)^{200}\approx 10^{-25}$$

là xác suất rất nhỏ. Do vậy Alice và Bob có thể coi khoá thô là khoá bí mật để sử dụng trong hệ mã One-time-pad.

3.1.1.3 Ví dụ

Dưới đây là ví dụ về giao thức BB84 trong trường hợp không có nhiễu và không có sự do thám của Eve

Bit dữ liệu của Alice	1	0	0	0	1	1	0	1	0	1
Bảng chữ Alice chọn	x	z	x	z	x	x	x	z	z	x
Qubits truyền đi	$\|1\rangle_x$	$\|0\rangle$	$\|0\rangle_x$	$\|0\rangle$	$\|1\rangle_x$	$\|1\rangle_x$	$\|0\rangle_x$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle_x$
Bảng chữ Bob chọn	x	z	x	x	z	x	z	x	z	z
Kết quả đo	1	0	0	0	0	1	0	0	0	1
Bit dữ liệu của Bob	1	0	0	0	0	1	0	0	0	1
Các vị trí không khớp				√	√		√	√		√
Khoá thô	1	0	0			1			0	

3.1.2 Giao thức phân phối khoá lượng tử BB84 trường hợp có nhiễu

Ở đây, giao thức BB84 sẽ được mở rộng trong trường hợp môi trường có nhiễu. Khi đó, Alice và Bob sẽ không phân biệt được lỗi do nhiễu hay lỗi do Eve do thám. Do đó, Alice và Bob sẽ phải giả thiết rằng toàn bộ lỗi của khoá thô là do Eve do thám trên đường truyền.

Trong trường hợp trên đường truyền có nhiễu, chúng ta vẫn có hai giai đoạn của giao thức BB84.

3.1.2.1 Giai đoạn 1: Giao tiếp qua kênh lượng tử.

Trong giai đoạn này, mọi thủ tục của giao thức tương tự như giai đoạn 1 của giao thức BB84 trong trường hợp không có nhiễu.

3.1.2.2 Giai đoạn 2: Giao tiếp qua kênh công cộng.

Trong trường hợp có nhiễu, Alice và Bob sẽ trao đổi với nhau qua kênh công cộng trong 4 pha. Pha 1 là tạo khoá thô, pha 2 là đánh giá lỗi, pha 3 là đồng bộ (*để tạo khoá đồng bộ - reconciled key*), pha 4 là khuyếch đại bí mật (*privacy amplification*).

Pha 1. Tạo khoá thô.

Pha này thực hiện giống pha 1 (giai đoạn 2) của giao thức BB84 trong trường hợp không có nhiễu, ngoại trừ trường hợp Alice và Bob cũng xoá các bit tại vị trí mà Bob không nhận được thông tin. Việc không nhận được thông tin do có sự do thám của Eve hoặc do nhiễu trên đường truyền.

Pha 2. Đánh giá lỗi của khoá thô.

Alice và Bob sử dụng kênh công cộng để đánh giá tỷ lệ lỗi của khoá thô bằng cách công bố các đoạn mẫu chọn ngẫu nhiên của khoá thô đã được thoả thuận giữa hai bên, công khai so sánh các bit này để có được tỷ lệ lỗi R. Những bit công khai này sẽ được loại bỏ ra khỏi khoá thô. Nếu R đạt quá ngưỡng R_{Max}, khi đó Alice và Bob sẽ phải quay lại giai đoạn 1 để bắt đầu lại. Nếu R nhỏ hơn ngưỡng R_{Max}, khi đó Alice và Bob sẽ chuyển qua pha 3.

Pha 3. Tạo khoá đồng bộ.

Mục đích của pha 3 là Alice và Bob sẽ xoá tất cả các bit lỗi từ khoá thô và được thực hiện trong 2 bước.

Bước 1. Alice và Bob công bố một thoả thuận hoán vị ngẫu nhiên và sử dụng nó vào phần khoá tương ứng, Kế tiếp Alice và Bob phân hoạch phần còn lại của khoá thô (*sau khi xử lý ở pha 2*) thành các khối độ dài l, trong đó l được chọn sao cho có không quá 1 lỗi trong khối đó. Sau đó Alice và Bob công bố công khai bit kiểm tra chẵn lẻ. Nếu bit kiểm tra chẵn lẻ không bằng nhau. Alice và Bob sẽ sử dụng thuật toán tìm kiếm nhị phân để tìm bit lỗi bằng cách chia khối làm 2 khối con, sau đó kiểm tra bit chẵn lẻ của hai khối con và tiếp tục tìm kiếm bit lỗi tại khối có bit kiểm tra chẵn lẻ khác nhau cho đến khi tìm và loại bỏ được bit lỗi. Alice và Bob tiếp tục lặp lại với các khối độ dài l khác.

Alice và Bob có thể lặp lại nhiều lần bước 1 với các hoán vị ngẫu nhiên khác nhau, chiều dài block l khác nhau để làm loại bỏ các bit lỗi.

Bước 2. Alice và Bob sử dụng một thủ tục đồng bộ khác để làm mịn kết quả. Đầu tiên, Alice và Bob công khai lựa chọn một tập con ngẫu nhiên của khoá thô, so sánh bit kiểm tra chẵn lẻ. Nếu bit chẵn lẻ không giống nhau, Alice và Bob sẽ áp dụng chiến lược tìm kiếm nhị phân như ở bước 1 để tìm và loại bỏ bit lỗi. Với xác suất cao, khoá thô cuối cùng không chứa các bit lỗi. Lúc này, khoá thô được gọi là *khoá đồng bộ* và tiếp tục chuyển sang pha 4.

Pha 4. Khuyếch đại bí mật (tạo khoá bí mật cuối cùng).

Lúc này, Alice và Bob đã có khoá đồng bộ nhưng chỉ một phần là bí mật với Eve. Vì vậy Alice và Bob sẽ bắt đầu tiến trình *khuyếch đại bí mật* để tách phần bí mật của khoá đồng bộ.

Dựa trên tỷ lệ lỗi R, Alice và Bob dự đoán cận trên k của số lượng các bit biết bởi Eve trong số n bit của khoá đồng bộ. Gọi s là tham số an toàn mà Alice và Bob mong muốn, khi đó Alice và Bob công bố $n - k - s$ tập con ngẫu nhiên từ khoá đồng bộ (không cần quan tâm đến nội dung). Sau đó Alice và Bob xoá khỏi khoá đồng bộ các phần mà cả hai cùng công khai, phần còn lại của khoá đồng bộ sẽ là khoá bí mật cuối cùng. Thông tin trung bình mà Eve có được về khoá bí mật cuối cùng này sẽ nhỏ hơn $2^{-s}/ln\ 2$ bit [55].

3.1.3 Một số nhược điểm của giao thức BB84.

- Eve có thể phá hoại kênh lượng tử (bằng cách đo các thông tin chuyển đi từ Alice dẫn đến xác suất lỗi tăng), khi đó Alice và Bob bắt buộc phải giao tiếp qua kênh cổ điển.
- Chi phí lớn cho không gian khoá: cần không gian khoá có độ lớn bằng độ lớn của không gian bản rõ mới đảm bảo an toàn tuyệt đối.
- Alice có thể gian lận mà Bob không phát hiện được:
    - Alice tạo một cặp qubit ở trạng thái rối lượng tử
    - Gửi đi một qubit cho Bob và giữ qubit còn lại

- Sau khi Bob tiến hành đo, Alice sẽ đo qubit còn lại và đoán được sự lựa chọn của Bob.

- Không có sự định danh trước khi tiến hành trao đổi khoá thông qua kênh lượng tử nên Alice có thể bị giả mạo.

3.1.4 Về độ an toàn của giao thức phân phối khoá BB84.

Các hệ mã hoá lượng tử, sử dụng các hiện tượng lượng tử như nguyên lý bất định Heisenberg, nguyên lý không thể sao chép, nguyên lý rối lượng tử để bảo vệ và phân phối khoá mã hoá. Các hệ mã hoá lượng tử cho phép hai người (hoặc hai tổ chức), không chia sẻ thông tin bí mật trước đó, có thể giao tiếp trên các kênh công cộng một cách an toàn. Do dựa trên các nguyên lý như trên (đặc biệt là nguyên lý không thể sao chép hoàn hảo và nguyên lý bất định Heisenberg) nên các hệ mã lượng tử được coi là an toàn chống lại các phương pháp tấn công của bên thứ ba.

Tuy nhiên một số sơ đồ tấn công các hệ mã lượng tử đã được phát triển như sơ đồ chặn/chuyển tiếp (*intercept/resend scheme*), sơ đồ phân tia sáng (*beamsplitting scheme*) tuy nhiên các sơ đồ tấn công trên vẫn làm nhiễu thông tin Bob thu được và sẽ bị phát hiện trong giao thức phân phối khoá BB84.

Trong phần này, tôi sẽ trình bày một sơ đồ tấn công có tên là *Sao chép gián tiếp* (Indirect Coping) mà Eve có thể sử dụng để thu được thông tin chuyển đi từ Alice mà không bị phát hiện bởi Alice và Bob trong giao thức phân phối khoá lượng tử BB84 (cũng như trong các giao thức tương tự như B92). Đây là phương pháp tấn công thuộc loại man-in-middle.

Sơ đồ này có thể mô tả như sau:

- Eve xây dựng một hàm quy tắc. Hàm này là hàm đơn trị với mọi trạng thái lượng tử khác nhau được sử dụng bởi Alice và Bob trong giao thức, nghĩa là mọi giá trị của hàm sẽ tương ứng với các trạng thái lượng tử khác nhau.

- Khi Alice gửi các bit lượng tử ngẫu nhiên đến cho Bob, Eve sẽ nhận mọi trạng thái này và tính giá trị tương ứng bởi hàm trên sau đó huỷ qubit nhận được.
- Sau đó Eve sẽ gửi một trạng thái lượng tử mới tới Bob dựa trên bảng tham chiếu trong đó mọi giá trị của hàm sẽ tương ứng với một trạng thái lượng tử.

Theo sơ đồ này, Eve sẽ có được chính xác thông tin trao đổi giữa Alice và Bob mà không bị phát hiện. Sơ đồ này có tên là *Sao chép gián tiếp* (Indirect Coping) do Eve không thực sự sao chép giá trị mà Alice gửi mà tạo một trạng thái lượng tử mới có trạng thái giống hệt trạng thái lượng tử Alice tạo.

3.1.4.1 Tạo bảng tham chiếu.

Trong giao thức phân phối khoá BB84, để thực hiện việc trao đổi thông tin, Alice và Bob sẽ phải công khai các trạng thái lượng tử sử dụng trong giao thức (mà Eve dễ dàng biết). Ở trên ta đã biết đó là các trạng thái thuộc:

- Bảng chữ cái z gồm hai trạng thái $|0\rangle$, $|1\rangle$
- Bảng chữ cái x gồm hai trạng thái $|+\rangle \equiv |0\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ và $|-\rangle \equiv |1\rangle_x = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Xét một trạng thái lượng tử phụ thích hợp, ví dụ $|\alpha\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$

Khi đó

$$\langle \alpha | 0 \rangle = \frac{\sqrt{3}}{2} \quad \rightarrow m_1 = \frac{3}{4} = 0.75$$

$$\langle \alpha | 1 \rangle = \frac{1}{2} \quad \rightarrow m_2 = \frac{1}{4} = 0.25$$

$$\langle \alpha | + \rangle = \frac{\sqrt{6}+\sqrt{2}}{4} \quad \rightarrow m_3 = \left(\frac{\sqrt{6}+\sqrt{2}}{4} \right)^2 = 0.933$$

$$\langle \alpha | - \rangle = \frac{\sqrt{6}-\sqrt{2}}{4} \quad \rightarrow m_4 = \left(\frac{\sqrt{6}-\sqrt{2}}{4} \right)^2 = 0.067$$

trong đó m_j (j = 1, 2, 3, 4) theo cơ học lượng tử là xác suất để trạng thái lượng tử $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ sụp đổ vào trạng thái $|\alpha\rangle$ khi tiến hành phép đo (theo cơ sở $|\alpha\rangle$).

Ta xây dựng bảng tham chiếu sau:

Trạng thái lượng tử	m_j
$\|0\rangle$	0.75
$\|1\rangle$	0.25
$\|+\rangle$	0.933
$\|-\rangle$	0.067

Dựa vào bảng tham chiếu trên, ta xây dựng hàm đơn trị $S_k = f(|j_k\rangle)$, k = 1, 2, 3, 4, $|j_k\rangle$ là một trong bốn trạng thái sử dụng trong BB84.

3.1.4.2 Sơ đồ tấn công.

- Eve xây dựng bảng tham chiếu và hàm đơn trị như trên
- Eve nhận tất cả các trạng thái lượng tử ngẫu nhiên mà Alice gửi cho Bob và tiến hành đo các trạng thái lượng tử nhận được.
- Theo kết quả đo được, Eve sẽ biết được trạng thái lượng tử mà Alice gửi đi là gì dựa trên bảng tham chiếu. Ví dụ, nếu kết quả đo được là 0.25, Eve sẽ biết được trạng thái lượng tử Alice gửi đi là $|1\rangle$.
- Eve sẽ huỷ trạng thái lượng tử nhận được từ Alice, do theo nguyên lý bất định, sau khi tiến hành phép đo, trạng thái lượng tử sẽ bị nhiễu loạn và thay đổi ngẫu nhiên so với trước khi đo.
- Eve tạo trạng thái lượng tử mới có giá trị tương ứng và gửi cho Bob

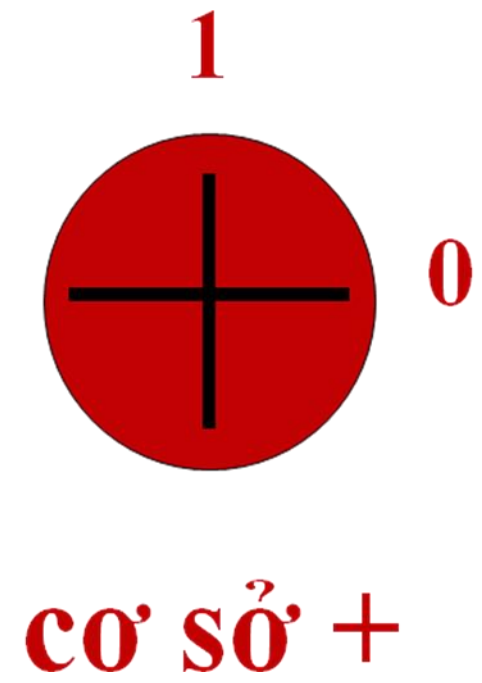
3.1.4.3 Kết luận về độ an toàn của giao thức BB84.

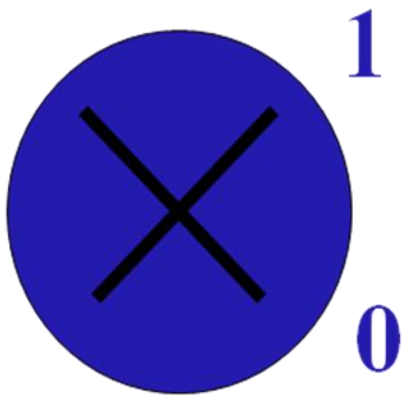
Như trên ta thấy, giao thức phân phối khoá lượng tử BB84 hoàn toàn không an toàn mặc dù theo các nguyên lý của cơ học lượng tử thì giao thức BB84 là giao thức an toàn. Tuy nhiên với sự ra đời của giao thức BB84 cũng cho thấy một tiềm năng hết sức lớn lao của các hệ mã lượng tử trong tương lai.

3.2. Kết luận về mã hoá lượng tử và thám mã lượng tử.

Hiện nay, mã hoá lượng tử và thám mã lượng tử đang là một lĩnh vực nghiên cứu hết sức sôi động trên thế giới. Chương này đã đưa ra một ví dụ đơn giản về mã hoá lượng tử và thám mã lượng tử. Giao thức phân phối khoá BB84 chỉ là một giao thức đơn giản và hoàn toàn không an toàn. Nhưng cũng đã cho ta thấy bức tranh về một lĩnh vực hoàn toàn mới mẻ, còn nhiều vấn đề mở cần đầu tư nghiên cứu phát triển.

CHƯƠNG 4. MÔ PHỎNG GIAO THỨC BB84





cơ sở + **cơ sở x**

Cơ sở x + + x + + x + x x + + + + x x

Dữ liệu 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0

Trạng thái / | - / -- / | \ / - - | - \\

Alice gửi đơn phôtôn với 1 trong 4 trạng thái phân cực ngẫu nhiên được chọn ("0"hoặc "1" trong cơ sở + hoặc x)

Cơ sở x + + x + + x + x x + + + + x x

Dữ liệu 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0

Trạng thái / | - / -- / | \ / - - | - \\

Bob đo ngẫu nhiên trong cở + hoặc x

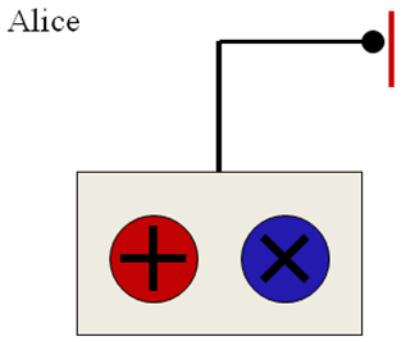
Cơ sở x + + x + + x + x x + + + + x x

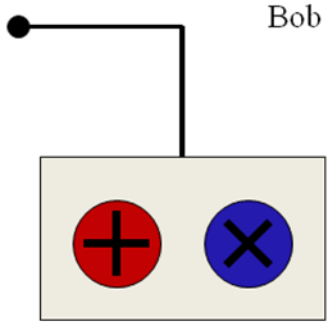
Dữ liệu 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0

Trạng thái / | - / -- / | \ / - - | - \\

Cơ sở x x + + + x + + + + x + + x x x

Nếu Bob sử dụng cơ sở như của Alice, anh ấy sẽ đo cùng trạng thái phân cực





Cơ sở x + + x + + x + x x + + + + x x

Dữ liệu 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0

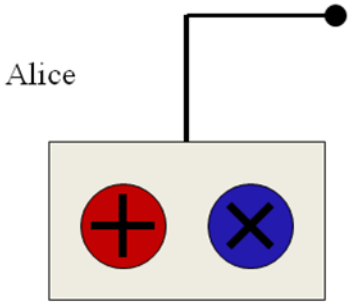
Trạng thái / | - / - - / | \ / - - | - \ \

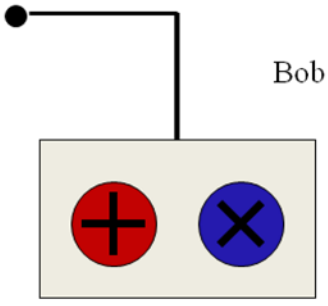
Cơ sở x x + + + x + + + + x + + x x x

Trạng thái /

Dữ liệu 1

Alice và Bob trao đổi qua kênh công cộng giữ lại những bit có cùng cơ sở đo,loại bỏ những bit đo không cùng cơ sở





Cơ sở x + + x + + x + x x + + + + x x

Dữ liệu 1 1 0 1 0 0 1 1 0 1 0 0 1 0 0 0

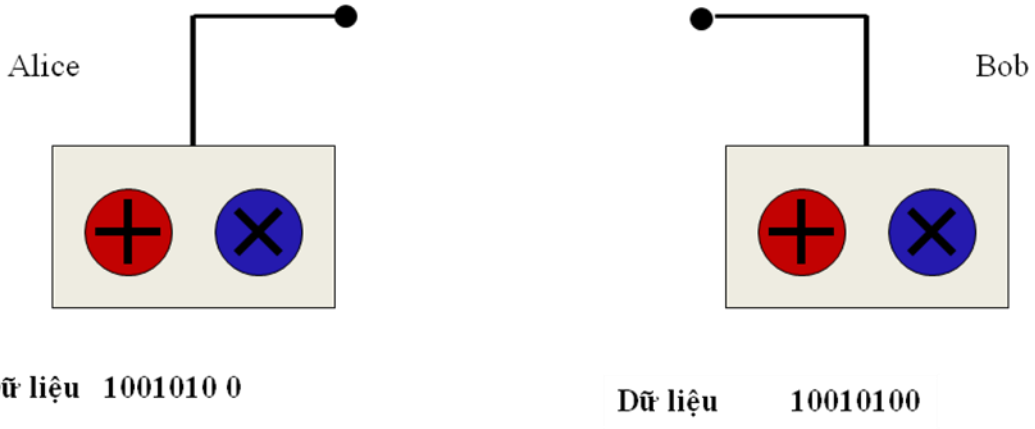
Trạng thái / | - / - - / | \ / - - | - \ \

Cơ sở x x + + + x + + + + x + + x x x

Trạng thái / \ - - - / | | | - \ - |/ \ \

Dữ liệu 1 0 0 0 0 1 1 1 1 0 0 0 1 1 0 0

Dữ liệu được giữ lại là chìa khoá bí mật dùng chung



KẾT LUẬN

Ngày nay, cùng với sự phát triển của khoa học hiện đại và Công nghệ thông tin, ngành mật mã đã có những bước phát triển mạnh mẽ, đạt được nhiều kết quả lý thuyết sâu sắc và tạo cơ sở cho việc phát triển các giải pháp bảo mật, an toàn thông tin trong mọi lĩnh vực hoạt động của con người:

Bước đầu tìm hiểu về tính toán lượng tử và thuật toán lượng tử trong đó chú trọng đi sâu vào ba chuyên đề chính là: Cơ sở tính toán lượng tử, Cổng và mạch lượng tử, Thuật toán lượng tử. Đồ án cũng bước đầu tìm hiểu về mã hóa lượng tử và thám mã lượng tử, đại diện là giao thức phân phối khóa lượng tử BB84.

Đồ án tập trung vào nghiên cứu cơ sở lý thuyết.Tuy còn nhiều điểm cần phải nghiên cứu và hoàn thiện nhưng do thời gian và trình độ còn hạn chế nên không thể tránh khỏi những nhược điểm, rất mong được sự góp ý của các Thầy, Cô và các bạn.

Cuối cùng em xin cảm ơn thầy giáo Thạc Sĩ Trần Ngọc Thái đã tận tình chỉ bảo giúp đỡ em hoàn thành đồ án này

TÀI LIỆU THAM KHẢO

[1] Kim Cương – *Toán học cao cấp, Tập 1, Phần 1: Đại số*, Nhà xuất bản Giáo dục, 1993

[2] Lê Quang Minh – *Tenxơ và Toocxơ*, Nhà xuất bản Giáo dục, 1998

Giải tích hàm, Nhà xuất bản Khoa học và Kỹ thuật, 1998

[3] Phạm Quý Tư, Đỗ Đình Thanh – *Cơ học lượng tử*, Nhà xuất bản Đại học Quốc Gia Hà Nội, Tái bản lần thứ nhất, 2003

[4] Nguyễn Quốc Khánh, Nguyễn Hữu Mạc – *Cơ học lượng tử 2*, Nhà xuất bản Đại học Quốc Gia Thành phố Hồ Chí Minh, 2000

[5] Vũ Văn Hùng – *Cơ học lượng tử*, Nhà xuất bản Đại học Sư phạm, 2004

[6] Nguyễn Hoàng Phương – *Lý thuyết Nhóm và ứng dụng vào Vật lý học lượng tử*, Nhà xuất bản Khoa học và Kỹ thuật, 2002

[7] Phạm Việt Hùng – *Mã hóa lượng tử và và mô phỏng trên máy tính,* Đồ án tốt nghiệp cao học Trường Đại Học Quốc Gia Hà Nội, 2006