**CÁC MÔ HÌNH CHIA SẺ DỰA TRÊN MÃ HÓA NGƯỠNG ỨNG DỤNG VÀO BÀI TOÁN CHỮ KÝ SỐ**

TRẦN MINH LONG 1 (\*), PHAN TÔN LỘC NGUYÊN1, NGUYỄN MINH PHÚC1, TRẦN NGỌC HUYỀN1, ĐÀO THỊ MỸ LINH1, NGUYỄN THỊ TUYẾT LAN1, NGUYỄN QUANG HUY1,

DƯƠNG HOÀNG HUY1,NGUYỄN HUY HOÀNG1, LÊ PHÚC LỮ1, NGUYỄN HỮU TÌNH1

*1Khoa Công nghệ Thông tin, Đại học Công nghiệp Thành phố Hồ Chí Minh*

*\* minhlong2002@gmail.com*

**Tóm tắt.** Mã hóa, bảo mật luôn đóng vai trò quan trọng trong nhiều lĩnh vực của đời sống, đặc biệt là khi một dữ liệu nhạy cảm cần chia sẻ trong một nhóm người. Bài toán đặt ra là sau khi mã hóa sử dụng khóa công khai, để có thể giải mã, cần có sự tham gia của một số lượng đủ lớn của những người trong nhóm (mã hóa ngưỡng). Nhóm đề tài sẽ tìm hiểu ý nghĩa của bài toán, cách thức hoạt động của các mô hình có liên quan như: Secret Shamir Sharing, Blakley, Định lý Số dư Trung Hoa (CRT) và nền tảng Toán khác. Nhóm sẽ tiến hành cài đặt và đánh giá, so sánh giữa các schemes nêu trên, đồng thời kết hợp thêm mã hóa bất đối xứng, từ đó tìm cách ứng dụng vào bài toán chữ ký số.

**Từ khóa:** Mã hóa ngưỡng, Secret Shamir Sharing, Blakley, Định lý Số dư Trung Hoa, chữ ký số.

**THRESHOLD CRYPTOGRAPHY SYSTEM APPLIED TO**

**DIGITAL SIGNATURE PROBLEMS**

**Abstract.** Encryption and security always play a crucial role in many areas of life, especially when sensitive data needs to be shared among a group of people. The problem is that after encrypting using public-key encryption, to be able to decrypt, the participation of a large enough amount of people in the group is required (threshold encryption). In this research, we will investigate the definition of the problem, the algorithms such as Secret Shamir Sharing, Blakley, Chinese Remainder Theorem (CRT), and other mathematical approaches. We will implement and evaluate, compare between the above schemes, and combine asymmetric encryption and apply them to the digital signature problem.

**Keywords.** Threshold cryptography, Secret Shamir Sharing, Blakley, CRT, Digital Signature.

**Lĩnh vực:** Điện – Điện tử - Công nghệ thông tin.

**1. GIỚI THIỆU**

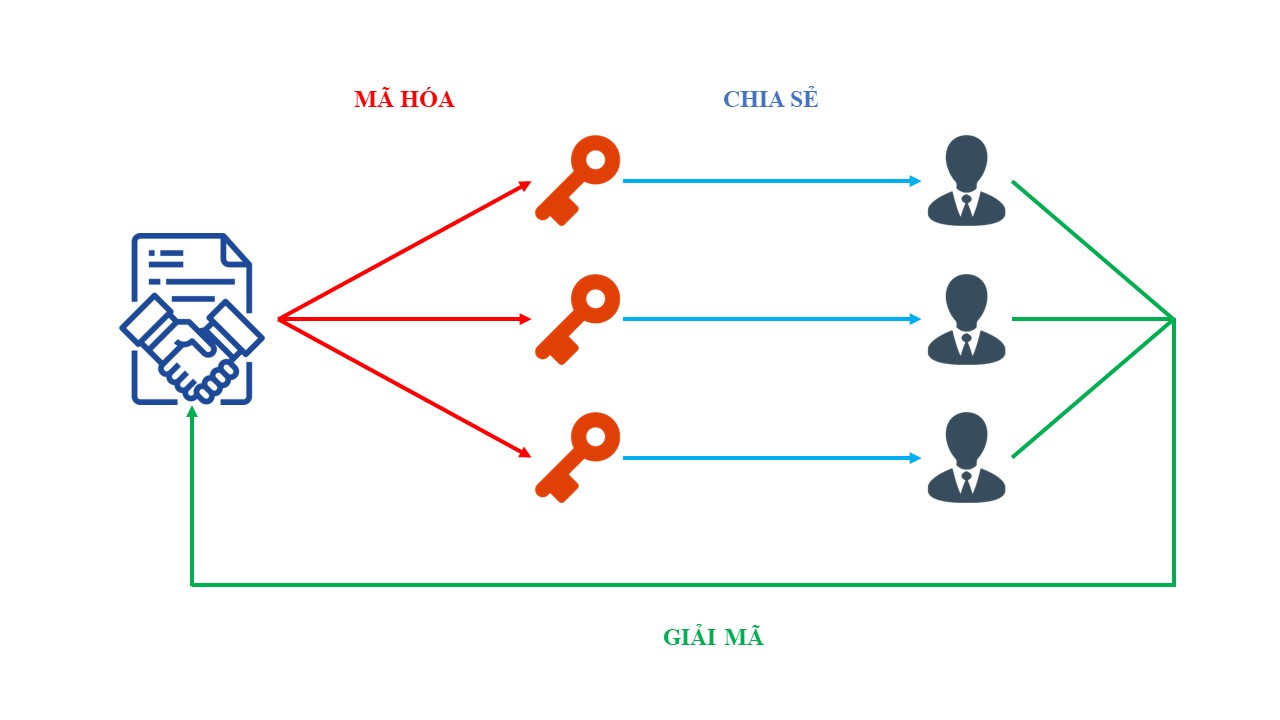
**1.1 Mô hình mã hóa ngưỡng (threshold encryption model)**

**1.1.1 Định nghĩa**

*Mô hình mã hóa ngưỡng* là một trong những mô hình mật mã học mà trong đó thông tin sẽ được mã hóa tạo thành các khóa khác nhau rồi phân phối cho nhiều bên sở hữu. Để giải mã thông tin đã được mã hóa cần sự tham gia của một số bên giữ khóa nhất định (nhiều hơn hoặc bằng ngưỡng định trước) cùng hợp tác.

Một trong những ưu điểm cơ bản của mô hình này là nó có thể ra tạo sự thuận tiện trong quá trình quản lý khóa. Với các hệ thống mật mã truyền thống, nếu người giữ khóa bị thất lạc khóa hoặc bị tấn công, toàn bộ hệ thống có thể cần phải đổi khóa mới để đảm bảo an toàn. Trong mô hình mã hóa ngưỡng, khóa có thể được xây dựng lại bằng cách sử dụng các phần còn lại, cho phép hệ thống hoạt động tiếp, không gián đoạn.

Mô hình mã hóa ngưỡng có thể được ứng dụng trong nhiều lĩnh vực, bao gồm giao tiếp an toàn, kiểm soát truy cập và chữ ký số. Nó cũng là một thành phần quan trọng của tính toán đa bên, cho phép nhiều bên cùng tính toán một chức năng mà không tiết lộ đầu vào riêng tư của họ.



*Hình 1. Ví dụ về mô hình mã hóa ngưỡng (ngưỡng là 3)*

**1.1.2 Tính chất**

*Tính bảo mật*: được thiết kế để bảo vệ các thông tin bí mật bằng cách phân chia chúng thành nhiều phần và phân phối chúng đến nhiều người dùng khác nhau. Điều này đảm bảo rằng không có ai có thể khôi phục lại thông tin bí mật nếu không có đủ số lượng người dùng và các phần của thông tin.

*Tính tin cậy*: phân chia thông tin thành nhiều phần khác nhau sẽ đảm bảo rằng thông tin không bị mất hoặc hư hỏng khi một phần bị mất hoặc bị thay đổi. Các phần thông tin được giữ bởi nhiều người dùng khác nhau, nên nguy cơ mất mát hoặc hư hỏng được giảm thiểu.

*Tính linh hoạt*: cho phép sử dụng các thông tin bí mật khác nhau với số lượng người được phân chia và ngưỡng khác nhau tùy theo yêu cầu của người dùng. Điều này làm cho mã hóa ngưỡng trở nên linh hoạt và có thể được áp dụng cho nhiều mục đích khác nhau.

*Tính hiệu suất*: có hiệu suất cao vì nó cho phép nhiều người dùng có thể truy cập vào thông tin bí mật cùng một lúc, giảm thiểu thời gian và tài nguyên cần thiết để quản lý thông tin bí mật.

*Tính an toàn*: có mức độ bảo mật cao hơn so với các hệ thống mật mã đơn người dùng truyền thống. Trong mô hình mã hóa ngưỡng, thông tin được chia thành nhiều khóa khác nhau, mỗi khóa chỉ nắm giữ một phần của thông tin. Điều này làm cho việc tấn công để truy cập đến toàn bộ thông tin trở nên khó khăn.

**1.2 Chữ ký số (digital signature)**

**1.2.1 Định nghĩa**

*Chữ ký số (Digital signature)* là một phần của chữ ký điện tử, đây một dạng dữ liệu văn bản xác thực và thường được đính kèm theo các loại dữ liệu siêu văn bản khác như hình ảnh, âm thanh, video,… Nhằm mục đích xác nhận chủ sở hữu của một vấn đề.

*Chữ ký số khóa công khai* (hay *hạ tầng khóa công khai*) là một mô hình ứng dụng mã hóa bất đối xứng (Asymmetric encryption model) vào chữ ký số để tạo sự tin cậy trong việc xác thực độ uy tín của chữ ký thông qua cặp khóa công khai – bí mật. Qua đó, người dùng có thể ký gián tiếp các văn bản điện tử hay trao đổi các thông tin một cách bí mật, uy tín và an toàn.

**1.2.2 Tính chất**

*Tính toàn vẹn*: do được áp dụng *mô hình mã hóa một chiều (One-way encryption model)* nên thừa hưởng tính vẹn toàn từ mô hình này. Nhờ đó, thông tin đảm bảo được toàn vẹn, vì nếu thông tin được sửa chữa trong quá trình gửi đi thì dẫn đến mã băm sẽ thay đổi và không thể xác thực được thông tin.

*Tính một chiều:* do được áp dụng *mô hình mã hóa một chiều (One-way encryption model)* nên thừa hưởng tính một chiều từ mô hình này. Nhờ đó, nếu quá trình mã hóa và truyền thông tin bị lộ đoạn mã băm thì các tin tặc vẫn không thể giải mã và tìm được về thông tin quan trọng ban đầu.

*Tính xác thực chủ sở hữu, xác định nguồn gốc*: được áp dụng *mô hình mã hóa bất đối xứng.* Nhờ đó, bên gửi đi có thể dùng khóa riêng tư của mình để mã hóa đoạn mã băm (hay mã gốc) tạo thành chữ ký số và gửi đi; bên nhận nhận được chữ ký, dùng khóa công khai của bên gửi giải mã để thu được mã gốc (quá trình xác thực chủ sở hữu), sau đó so sánh mã gốc nhận được với mã gốc ban đầu (xác định nguồn gốc).

*Tính không thể phủ nhận*: được áp dụng *mô hình mã hóa bất đối xứng*. Nhờ đó, thông tin được mã hóa và giải mã ngay từ chính cặp khóa công khai – bí mật của người gửi nên người gửi không thể phủ nhận.

**2. CÁC LÝ THUYẾT CƠ SỞ**

**2.1. Tổng hợp các lý thuyết**

Các mô hình chia sẻ bí mật dựa trên Threshold cryptosystem sử dụng nhiều phép toán rời rạc và số học:

* Phép toán modular (Modulus): Phép toán này được sử dụng để đảm bảo tính bảo mật của thông tin bí mật được chia sẻ. Các phép tính toán trên các phần tử được thực hiện trong không gian modulo để tránh việc thông tin bí mật bị lộ ra bên ngoài.
* Phép toán nghịch đảo modular (Modular Inverse): Đây là một phép toán rất quan trọng trong mô hình chia sẻ bí mật. Phép toán này được sử dụng để tính toán các giá trị khóa riêng tư và khóa công khai cho các thành viên trong tập hợp con.
* Phép toán XOR (Exclusive OR): Đây là một phép toán rời rạc đơn giản được sử dụng để tạo ra các bộ khóa và chia sẻ bí mật. Ví dụ: a XOR b = c, trong đó c được tính bằng cách so sánh từng bit của a và b. Trong C/C++ XOR là một toán tử logic được sử dụng để kiểm tra xem một số điều kiện có đúng hay không.
* Phép toán AND (Logical AND): Đây là một phép toán logic được sử dụng để thực hiện các tính toán trên các bit. Ví dụ: a AND b = c, trong đó c là giá trị được tính bằng cách so sánh từng bit của a và b và trả về giá trị 1 nếu cả hai bit đều là 1.
* Phép toán OR (Logical OR): Đây là một phép toán logic được sử dụng để thực hiện các tính toán trên các bit. Ví dụ: a OR b = c, trong đó c là giá trị được tính bằng cách so sánh từng bit của a và b và trả về giá trị 1 nếu có ít nhất một bit là 1.
* Ma trận: thực hiện các tính toán đại số tuyến tính và kiểm tra tính đúng đắn của quá trình chia sẻ bí mật. Trong Python tính toán trên ma trận thường được thực hiện bằng cách dùng thư viện NumPy.
* Phép toán nhân trên các số nguyên rời rạc: Phép toán này được sử dụng để tạo ra các khóa riêng tư và khóa công khai cho các thành viên trong tập hợp con.
* Phép toán chia trên các số nguyên rời rạc: Phép toán này được sử dụng để tính toán các ngưỡng được thiết lập trước đó, và chỉ khi số thành viên trong tập hợp con đạt đủ ngưỡng thì mới có thể khôi phục lại thông tin bí mật.
* Phép toán cộng và trừ trên các số nguyên rời rạc: Các phép toán này được sử dụng để tính toán các giá trị đại diện cho thông tin bí mật, và đảm bảo tính toàn vẹn của thông tin khi chia sẻ nó đến các thành viên trong tập hợp con.
* Giải hệ phương trình tuyến tính : Xây dựng dựa vào ma trận. Được sử dụng cho mục đích giải mã bí mật Blakley sử dụng phép toán rời rạc được sử dụng để giải hệ phương trình này và tìm ra vector bí mật ban đầu.
* Đa thức Lagrange: Để khôi phục bí mật từ các phần được chia sẻ, các bên sử dụng đa thức Lagrange. Đa thức này được tính bằng cách sử dụng các giá trị rời rạc và các điểm được chọn để chia sẻ bí mật ban đầu.

**2.2 Số học**

**2.2.1 Đồng dư**

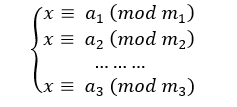
Cho . Nếu a và b khi chia cho m có cùng số dư ta nói: a và b đồng dư theo modulo m.

Kí hiệu: Hệ thức: gọi là đồng dư thức. Các tính chất

**2.2.2 Định lý số dư Trung Hoa (Chinese Remainder Theorem - CRT)**

**2.2.2.1. Nội dung và ứng dụng**

*Định lý số dư Trung Hoa (Chinese Remainder Theorem - CRT)* là một định lý toán học liên quan đến số học modulo và được sử dụng rộng rãi trong lý thuyết số, mã hóa thông tin và các lĩnh vực liên quan đến toán học ứng dụng. Định lýnói về nghiệm của hệ phương trình đồng dư bậc nhất:



*trong đó*: m1,m2,…, mk đôi một nguyên tố cùng nhau. Cụ thể là: hệ luôn có nghiệm duy nhất theo modulo m1.m2….mk . Phương pháp giải:

Bước 1: Đặt với .

Bước 2: Tìm các nghiệm của phương trình (sử dụng thuật toán Euclid mở rộng)

*Định nghĩa modulo nghịch đảo:* Nếu gcd(a,b) = 1 => a\*x (mod b) = 1 thì x là nghịch đảo của a theo mod b. Kí hiệu là: . Tìm x: a.x + b.y = 1

Bước 3: Tìm được một nghiệm của hệ là: .

Bước 4: Kết luận nghiệm: .

*Ví dụ:* (bài toán Hàn Tín Điểm binh) Hàn Tín là một danh tướng thời Hán Sở, từng được phong tước vương thời Hán Cao Tổ Lưu Bang, ông có khả năng tính chính xác số binh lính bằng cách: Cho binh lính lần lượt xếp thành hàng 3, hàng 5, hàng 7, rồi lấy số binh lính dư ra ở mỗi cách xếp hàng để tính ra được số lính. Cụ thể: binh lính xếp hàng 3 thì dư 2 người, xếp hàng 5 thì dư 3 người, xếp hàng 7 thì dư 5 người.

Ta được một hệ phương trình đồng dư là: 3, 5, 7 đôi một nguyên tố cùng nhau, có UCLN là 1.

* Bước 1: m = 3.5.7 = 105; = 5.7 = 35; = 3.7=21; = 3.5 = 15
* Bước 2: Tìm được gcd(35,3) = 1. Tìm modulo nghịch đảo của 35 trong mod 3, được kết quả là 2.
* Tìm gcd(21,5) = 1. Tìm modulo nghịch đảo của 21 trong 5, được 1.
* Tìm gcd(17,5) = 1. Tìm modulo nghịch đảo của 17 trong 5, được 3.
* Bước 3: x0 = 2.2.35 + 3.1.21 + 5.1.15 = 278.
* Bước 4: do 278 % 105 = 68 nên .

Bên dưới là các ứng dụng:

* + - * Để giải quyết những bài toán về sự tồn tại và đếm các số nguyên thỏa mãn một hệ các điều kiện đồng dư, quan hệ chia hết,…
      * Có nhiều ứng dụng trong các bài toán về số nguyên lớn áp dụng vào lí thuyết mật mã, trong đó tiêu biểu là lý thuyết mật mã RSA.
      * Sử dụng định lý để chứng minh công thức Euler, hay giải bài toán mở rộng của định lý Wilson và đếm số nghiệm của phương trình đồng dư.
      * Tùy vào mục đích sử dụng, còn một số ứng dụng khác,....

Ta cũng có thể áp dụng CRT vào mã hóa:

* + - * CRT được sử dụng để giải mã thông điệp được mã hóa bằng thuật toán RSA, giúp tăng tốc độ xử lý và đảm bảo tính toán chính xác, tối ưu quá trình giải mã và mã hóa.
      * CRT giúp tăng tốc độ tính toán trong mã hóa, giảm thiểu áp lực tính toán trên các số lớn và không gian lưu trữ yêu cầu cho khối dữ liệu mã hóa.
      * CRT là một công cụ toán học được sử dụng rộng rãi trong các hệ thống mã hóa hiện đại, trong đó quan trọng là mã hóa bất đối xứng, phổ biến nhất là mã hóa RSA.

**2.2.2.2 Mã nguồn C/C++**

Đường dẫn Github: <https://github.com/TranNgocHuyen19/YSC-2023>.

**2.3 Mô hình mã hóa bất đối xứng (asymmetric encryption model)**

**2.3.1 Định nghĩa**

*Mô hình mã hóa bất đối xứng (Asymmetric encryption model)* là một trong những mô hình mật mã học mà trong đó thông tin sẽ được mã hóa thông qua một cặp khóa gồm *khóa công khai (Public key)* và *khóa bí mật (Private key)*. Khóa công khai có thể được chia sẻ rộng rãi và được sử dụng để mã hóa thông tin, trong khi khóa bí mật chỉ có thể được sở hữu bởi chủ sở hữu của nó và được sử dụng để giải mã thông tin.

Phương pháp mã hóa bất đối xứng thường được sử dụng để tăng tính bảo mật và độ an toàn cho các thông tin quan trọng trong giao thức truyền tin qua mạng Internet. Với mã hóa bất đối xứng, người gửi thông tin mã hóa dữ liệu của họ bằng khóa công khai của người nhận, và chỉ người nhận có thể giải mã thông tin đó bằng khóa bí mật của mình.

**2.3.2. Ưu điểm**

* *Bảo mật cao:* Mã hóa bất đối xứng sử dụng một cặp khóa gồm khóa công khai và khóa bí mật. Với việc sử dụng khóa công khai để mã hóa và khóa riêng tư để giải mã, thông tin được bảo vệ tốt hơn so với mã hóa đối xứng.
* *Dễ dàng chia sẻ khóa công khai:* Khóa công khai có thể được chia sẻ rộng rãi mà không ảnh hưởng đến tính bảo mật của thông tin được mã hóa.
* *An toàn trong truyền thông:* Mã hóa bất đối xứng được sử dụng rộng rãi trong truyền thông an toàn qua mạng, xác thực người dùng, chữ ký số,…

**2.3.3 Nhược điểm**

* *Tốc độ chậm:* Tốc độ mã hóa bất đối xứng thường chậm hơn so với mã hóa đối xứng.
* *Khóa riêng tư phải được giữ tốt:* Nếu khóa riêng tư bị rò rỉ, thông tin sẽ dễ dàng bị xâm nhập và đánh cắp. Do đó, việc bảo vệ khóa riêng tư là rất quan trọng trong mã hóa bất đối xứng.
* *Khó thực hiện trên các thiết bị có tài nguyên hạn chế:* Việc thực hiện mã hóa bất đối xứng trên các thiết bị có tài nguyên hạn chế như điện thoại thông minh, máy tính nhúng, IoT,... có thể gặp khó khăn do tính toán phức tạp và tốc độ xử lý chậm.

**2.4 Phương pháp mã hóa RSA**

**2.4.1 Giới thiệu sơ lược**

*Phương pháp mã hóa RSA* là một trong những phương pháp mã hóa bất đối xứng phổ biến nhất được sử dụng hiện nay. Mã hóa RSA là một trong những phương pháp mã hóa công khai được phát triển bởi Ron Rivest, Adi Shamir và Leonard Adleman vào năm 1977. RSA là viết tắt các chữ cái đầu tiên trong tên của ba nhà khoa học này.

Mã hóa RSA dựa trên nguyên tắc rằng việc phân tích một số nguyên thành các thừa số nguyên tố là rất khó khăn, đặc biệt là khi số đó rất lớn. Do đó, phương pháp mã hóa RSA khai thác sử dụng tính chất này của các số nguyên tố để tạo ra cặp khóa công khai và khóa bí mật.

**2.4.2 Giải thuật**

*Thuật toán mã hóa RSA* hoạt động dựa trên nguyên tắc về tính khó giải của việc phân tích một số nguyên lớn thành các *thừa số nguyên tố*. Mã hóa RSA bao gồm các quá trình sau:

Quá trình tạo khóa:

*Bước 1*: Chọn hai số nguyên tố lớn *p* và *q* ngẫu nhiên ().

*Bước 2*: Tính .

*Bước 3*: Tính hàm số mũ số nguyên tố Euler của n: .

*Bước 4*: Chọn số *e* sao cho và *e* là [số nguyên tố cùng nhau](https://vi.wikipedia.org/wiki/S%E1%BB%91_nguy%C3%AAn_t%E1%BB%91_c%C3%B9ng_nhau) với .

*Bước 5*: Tính [số nguyên tố cùng nhau](https://vi.wikipedia.org/wiki/S%E1%BB%91_nguy%C3%AAn_t%E1%BB%91_c%C3%B9ng_nhau) với và thỏa mãn . Số d là *khóa bí mật*.

* *Khóa công khai bao gồm:*
  + - *n* – môđun.
    - *e* - số mũ công khai (số mũ mã hóa).
* *Khóa bí mật bao gồm:*
  + - *n* - môđun, xuất hiện cả trong *khóa công khai* và *khóa bí mật*.
    - *d* - số mũ bí mật (số mũ giải mã).
* *Một dạng khác của khóa bí mật bao gồm:*
  + - *p và q* - hai số nguyên tố chọn ban đầu.
    - *d mod (p - 1), d mod (q - 1)* - thường được gọi là *dmp1, dmq1*.
    - *mod p* - thường được gọi là *iqmp*.

Dạng này cho phép thực hiện giải mã và ký nhanh hơn với việc sử dụng [định lý số dư Trung Quốc](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C3%BD_s%E1%BB%91_d%C6%B0_Trung_Qu%E1%BB%91c). Ở dạng này, tất cả thành phần của khóa bí mật phải được giữ bí mật. A gửi khóa công khai cho B, và giữ bí mật khóa bí mật của mình. Ở đây, *p* và *q* giữ vai trò rất quan trọng. Chúng là các phân tố của *n* và cho phép tính *d* khi biết *e*. Nếu không sử dụng dạng sau của khóa bí mật (dạng CRT) thì *p* và *q* sẽ được xóa ngay sau khi thực hiện xong quá trình tạo khóa.

Quá trình mã hóa: Giả sử A muốn gửi đoạn thông tin *M* cho B. Đầu tiên A chuyển *M* thành số *m* *(m < n)* theo một phương thức được thỏa thuận trước. Lúc này A có *m* và biết *n*, *e* do B gửi. Bình sẽ tính *c* là bản đã mã hóa của *m* theo công thức:

Hàm trên có thể tính dễ dàng bằng cách sử dụng phương pháp tính [hàm mũ](https://vi.wikipedia.org/wiki/H%C3%A0m_m%C5%A9) (theo môđun) thông qua [thuật toán bình phương và nhân](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_b%C3%ACnh_ph%C6%B0%C6%A1ng_v%C3%A0_nh%C3%A2n). Cuối cùng A gửi *c* cho B.

Quá trình giải mã: B nhận *c* từ A và biết khóa bí mật *d*. B có thể tìm được *m* từ *c* theo công thức sau:

Biết *m*, B tìm lại *M* theo phương pháp đã thỏa thuận trước đó. Quá trình giải mã hoạt động vì ta có:

Do *e \* d* ≡ 1 (mod *p* - 1) và *e \* d* ≡ 1 (mod *q* - 1), (theo [Định lý Fermat nhỏ](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C3%BD_nh%E1%BB%8F_Fermat)) nên:

*me\*d = m mod p* và *me \* d = m mod q*

Do *p* và *q* là hai số nguyên tố cùng nhau, áp dụng [định lý số dư Trung Quốc](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_l%C3%BD_s%E1%BB%91_d%C6%B0_Trung_Qu%E1%BB%91c), ta có:

*me \* d = m mod (q \* p)* hay *cd = m mod n*

**2.4.3 Mã C/C++:**

Đường dẫn Github: <https://github.com/phantonlocnguyen/RSA>

**2.5. Mô hình mã hóa một chiều (One-way encryption model)**

**2.5.1. Giới thiệu.**

*Mô hình mã hóa một chiều (One-way encryption model)* là một trong những mô hình mật mã học mà trong đó thông tin sẽ được mã hóa thành một mật mã duy nhất và không thể giải mã tìm lại thông tin đầu. Nó có các đặc điểm sau đây:

* *Tính toàn vẹn*: tạo ra một đoạn giá trị mã hóa cố định duy nhất và không trùng lập cho mỗi thông tin đầu vào khác nhau.
* *Tính một chiều:* thông tin sau khi được mã hóa sẽ không thể giải mã tìm lại thông tin ban đầu.

Về hàm Hash (hàm băm): là hàm thực hiện quá trình biến một dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi đầu ra đặc trưng có độ dài cố định. Các giá trị được trả về bởi hàm băm được gọi là giá trị băm, mã băm, thông điệp băm, hoặc đơn giản là *“Hash”*. Ta mô tả cụ thể như bên dưới:

**2.5.2. Giải thuật MD5**

Dưới đây là các bước thực hiện giải thuật MD5:

Định nghĩa một macro để thực hiện phép xoay trái (rotate left) trên một giá trị 32 - bit.

Định nghĩa hàm *‘md5\_hash’* nhận các tham số đầu vào: một con trỏ trỏ tới một mảng các byte đầu vào (*‘initial\_msg’*), một kích thước của đầu vào (*‘initial\_len’*), và một con trỏ trỏ tới một mảng các byte để lưu giá trị băm (*‘digest’*).

void md5\_hash(const uint8\_t \* initital\_msg, size\_t initial\_len, uint8\_t \*digest)

Khởi tạo các giá trị hằng số cho bước khởi tạo của thuật toán MD5.

Tính toán độ dài đầu vào của đơn vị bit.

Chuẩn bị đầu vào cho thuật toán MD5 bằng cách tạo một mảng *‘msg’* có độ dài phù hợp với quy tắc của thuật toán. Mảng này chứa đầu vào gốc, bit “1” được thêm vào cuối, và các phần tử còn lại được được điền bằng 0. Độ dài bit cuối cùng của đầu vào được ghi vào cuối mảng.

Tạo con trỏ *‘M’* tới mảng *‘msg’* với với kiểu dữ liệu là uint32\_t.

Chia nhỏ phần đầu vào thành các khối 512 - bit (64 - byte). Con trỏ *‘X’* trỏ tới một khối trong mảng *‘M’*.

Khởi tạo các biến đệm ban đầu.

Thực hiện một vòng lặp để tính toán các giá trị mới của biến đệm A, B, C, D dựa trên các phép toán logic và phép xoay trái:

* Cập nhật các giá trị của biến đệm với các giá trị mới tính toán.
* Chuyển đổi giá trị của các biến đệm thành dạng byte và lưu vào mảng *‘digest’* để đưa ra kết quả cuối: Hàm *‘main()’* chứa ví dụ sử dụng hàm MD5. Khởi tạo một chuỗi *‘message’* và một mảng *‘digest’* để lưu giá trị băm. Sau đó, gọi hàm *‘md5\_hash()’* với đầu vào là chuỗi và lưu kết quả vào mảng *‘digest’*. Cuối cùng, in ra giá trị MD5 dưới dạng mã hexa.

**2.5.3 Mã C/C++**

Đường dẫn Github: <https://github.com/phantonlocnguyen/HASH_MD5>

**3. CÁC CÁCH TIẾP CẬN CHO MÃ HÓA NGƯỠNG**

**3.1 Mã hóa Shamir**

**3.1.1 Định nghĩa**

Mã hóa Shamir (*Shamir's Secret Sharing Scheme – SSS*) là một thuật toán được đặt theo tên của nhà mật mã học Adi Shamir người Israel, ông là người đã phát minh ra nó. Thuật toán được sử dụng để phân phối một bí mật thành nhiều phần, được gọi là các chia sẻ bí mật, trong đó mỗi chia sẻ bí mật được phân phối cho một nhóm người tham gia. Mục đích của *Shamir's Secret Sharing Scheme* là bảo vệ tính riêng tư và đảm bảo an toàn trong việc truyền và xác minh thông tin. Các phần của chia sẻ bí mật được tập hợp lại với nhau để xây dựng lại bí mật ban đầu, tổng số các chia sẻ bí mật nhỏ hơn tổng số các chia sẻ bí mật ban đầu được chia sẻ gọi là ngưỡng, là điều bắt buộc để xây dựng lại bí mật. Điều này giúp tránh thất bại trong việc giải mã thông tin được lưu giữ chặt chẽ nếu chỉ một hoặc một vài bên không có mặt. Thuật toán *Shamir's Secret Sharing Scheme – SSS* dựa trên nguyên tắc đa thức và áp dụng nó vào việc phân phối bí mật. Mỗi chia sẻ được tạo ra bằng cách chọn ngẫu nhiên một điểm trên đường cong của một đa thức với hệ số ngẫu nhiên. Để xây dựng lại bí mật, người dùng cần có đủ số lượng chia sẻ và phải biết đường cong và ngưỡng được sử dụng để tạo ra các chia sẻ.

**3.1.2 Công thức**

**3.1.2.1 Mã hóa Shamir**

Công thức Chia sẻ bí mật Shamir gồm các bước sau:

1. Xác định ngưỡng K để tái tạo lại chia sẻ bí mật ban đầu. Sau đó ta lập đa thức độc lập tuyến tính với bậc đa thức là
2. Chọn số S (secret) là bí mật cần chia sẻ, số nguyên a1, a2,... am (1 < a1, a2,...am < S).
3. Tính toán hàm đa thức mod S bằng công thức:

trong đó:

* m là hằng số ngẫu nhiên (là các hệ số tuyến tính a1, a2,... am).
* secret là 1 hằng số bí mật cần được mã hóa.
* K là ngưỡng chia sẻ bí mật.

1. Chọn N (là số các chia sẻ bí mật được quy định ban đầu) điểm ngẫu nhiên trên đường cong bằng cách tính toán các giá trị mà tại đó điểm N phải là nghiệm của hàm (x là giá trị bất kì, x i, đây là điều kiện cần) và điểm này phải có ít nhất tối thiểu N K (là điều kiện đủ) điểm hình thành các chia sẻ bí mật.
2. Các điểm N này sẽ được chia sẻ cho các bên cần truy cập bí mật.
3. Để tái tạo bí mật, tối thiểu K (như đã quy định ban đầu, K là ngưỡng chia sẻ bí mật) phần chia sẻ bí mật được cung cấp cho các bên, sau đó áp dụng công thức nghịch đảo Lagrange để tính toán giá trị của bí mật gốc.

*Ví dụ minh họa:*

1. Ban đầu, để mã hóa mật mã, ta xây dựng đa thức bậc (K - 1).
2. Đặt đa thức là . Ở đây, phần hằng số 'a' = secret là mã bí mật của chúng ta.
3. Gọi b là một số bất kỳ, giả sử b = 2.
4. Do đó, đối với đa thức y = 7 + 2x này, ('a' = secret = 7, b = 2) chúng ta tạo ra được N = 4 điểm từ nó.
5. Gọi 4 điểm đó là (1, 9), (2, 11), (3, 13), (4, 15). Rõ ràng, chúng ta có thể tạo đa thức ban đầu từ bất kỳ từ 2 trong số 4 điểm này và trong đa thức kết quả, hằng số 'a' là mã bí mật bắt buộc.

**3.1.2.2 Giải mã Shamir**

Chia sẻ bí mật Shamir gồm các bước sau:

1. Lấy K (là ngưỡng chia sẻ bí mật) phân đoạn đã được chia sẻ, mỗi phân đoạn bao gồm một cặp giá trị (x, y).
2. Để xây dựng lại đa thức đã cho hay tìm lại đa thức ban đầu, chúng ta hình thành các đẳng thức của Lagrange.

Công thức tổng quát của đa thức Lagrange như sau:

1. Dựa vào công thức trên, giải các đẳng thức cơ sở của đa thức Lagrange tìm ra phương trình từ đó tìm ra bí mật ban đầu.

*Ví dụ cụ thể:* Trong 1 trò chơi đi tìm kho báu, để có thể mở được kho báu đó cần nhập đúng mã bí mật là 2023. Ban tổ chức đã mã hóa mã bí mật gốc này thành 4 chia sẻ bí mật khác nhau, để mở được chiếc ổ khóa đó cần có ít nhất 3 mã chia sẻ bí mật trong 4 mã chia sẻ bí mật đó. Nhưng có một điều ban tổ chức không biết cách nào để tạo được các chia sẻ bí mật, bạn hãy giúp ban tổ chức mã hóa bí mật đó và xác định chia sẻ bí mật đó là trùng khớp với mã bí mật ban đầu.

*Phân tích.* Theo đề bài ra, ta có: K = 3, S = 2023, N = 4.Áp dụng công thức tổng quát: để tạo chia sẻ bí mật. Ta chọn đa thức  Là hàm đa thức bậc 2 để mã hóa bí mật.

* Với  (1)
* Với  (2)
* Với  (3)
* Với  (4)

Từ (1), (2), (3), (4) ta được tập nghiệm của phương trình là S = (x, y) = {(1, 2028); (2, 2039); (3, 2056); (3, 2108)}. Như vậy bạn đã giúp ban tổ chức tạo ra 4 chia sẻ bí mật.

Tiếp đến, chọn 3 điểm trong 4 điểm trên để tìm ra đa thức ban đầu lấy lại mã bí mật gốc mở khóa kho báu trên. Ta có K = 3, gọi lần lượt các điểm (1, 2028); (2, 2039); (3, 2056) là , , .

Áp dụng công thức đa thức Lagrange ta được:

++

Từ kết quả trên ta thấy con số 2023 trong phương trình chính là mã bí mật để mở khóa kho báu.

**3.1.3 Mã C/C++**

*Đường dẫn Github:* [*https://github.com/JosephPhuc/YSC-2023*](https://github.com/JosephPhuc/YSC-2023)

**3.2 Blakley**

**3.2.1 Định nghĩa**

Giả sử có 10 nhà khoa học phát minh ra một phát minh vĩ đại nhưng lại mang tiềm năng sức mạnh quân sự khủng khiếp. Bên cạnh đó họ lại là các nhà khoa học từ khắp thế giới, mỗi nhà khoa học lại thuộc một quốc gia khác nhau. Sau khi biểu quyết họ quyết định cất giấu nghiên cứu này. Nhưng có cách nào để tất cả mọi người đều nắm bí mật nhưng lại không sử dụng được nó, chỉ khi đủ số người đồng ý mới sử dụng. Đây là nơi mà mã hóa ngưỡng nói chung tỏa sáng và thực hiện được mục đích của mình.

Mã hóa ngưỡng là mô hình mã hóa mà ở đó: (với n không bé hơn t )

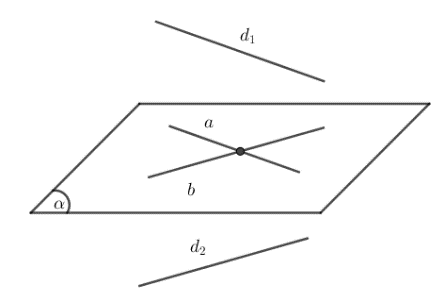
* Bí mật được chia làm n phần,
* Chỉ khi có t phần hoặc hơn trong n phần đó thì mới có thể xác định chính xác bí mật đó. Khi này số “t” được gọi là ngưỡng.

Quay lại bài toán của các nhà khoa học, nếu sử dụng mã hóa ngưỡng, bí mật ở đây sẽ được chia thành 10 phần và ngưỡng có thể là 6,7,...,10 và vấn đề đã được giải quyết. Ai trong số 10 nhà khoa học cũng nắm giữ 1 phần thông tin về bí mật đó cũng như nếu không có t người họp lại và chia sẻ bí mật của họ thì bí mật sẽ không thể được khôi phục và sử dụng cho mục đích xấu. Tất nhiên đó là nếu họ không muốn.

Mô hình chia sẻ bí mật Blakley (*hay mô hình mã hóa ngưỡng tuyến tính Blakley*) là một mô hình chia sẻ bí mật giữa có bên không có sự tin cậy và cần chia sẻ bí mật cũng như số lượng thông tin về bí mật đều nhau. Mô hình mã hóa được đặt tên theo chính tác giả của nó George Robert Blakley (1). Điểm khác biệt của mô hình Blakley so với mô hình Shamir đó chính là ở việc mô hình này sử dụng giao điểm của các siêu phẳng thay vì sử dụng tập hợp các điểm như mô hình Shamir. Khác với mô hình Shamir, mô hình Blakley sử dụng các đa thức tuyến tính để tính toán ra điểm bí mật. Điểm bí mật của Blakley là giao điểm của các siêu phẳng trong một không gian t chiều. Chính vì cách tiếp cận này mà mô hình Blakley không được phổ biến bằng mô hình Shamir do độ phức tạp của hướng tiếp cận toán học.

**3.2.2 Toán học trong mô hình Blakley :**

Thay vì sử dụng tập hợp các điểm tạo thành 1 đường thẳng hay đa thức như mô hình Shamir, trong mô hình Blakley ta lại sử dụng nhiều siêu phẳng để tìm ra một giao điểm của các siêu phẳng đó. Để hiểu rõ hơn về vấn đề này, ta sẽ đi qua những không gian 2 và 3 chiều :

Trong không gian 2 chiều :

* Một đường thẳng chính là một siêu phẳng
* Ta có thể thấy được dễ dàng rằng ta sẽ có 1 giao điểm cho 2 đường thẳng.

A picture containing screenshot, design

Description automatically generatedTrong không gian 3 chiều :

* Phần giao nhau của 2 siêu phẳng là một đường thẳng:
* Sẽ luôn vô số siêu phẳng khác khi giao nhau với 2 siêu phẳng mà

A diagram of a solution

Description automatically generated with low confidencephần giao nhau của chúng là một điểm :

Minh họa cho phần giao của hai siêu phẳng

Minh họa cho giao điểm của 3 siêu phẳng

Như vậy ta có thể thấy được rằng, trong không gian t chiều thì chắc chắn sẽ có t siêu phẳng giao nhau tại 1 điểm. Tiếp theo, ta sẽ nói về mã hóa Blakley :

Mã hóa Blakley đưa chúng ta về bài toán tạo ra n siêu phẳng từ điểm bí mật ban đầu. Hay có thể hiểu cách khác rằng tạo ra ma trận C sao cho ma trận C thỏa điều kiện sau :

Cx mod P = y, với

* C : ma trận chứa hệ số của các siêu phẳng khuyết hệ số tự do,
* y : vectơ chứa hệ số tự do tương ứng với các siêu phẳng đó,
* P : số nguyên tố bất kì mà ta chọn.

Để mã hóa Blakley như đã nói ta sẽ chia bí mật thành các siêu phẳng. Các siêu phẳng này sau đó sẽ được chia sẻ cho những cổ phần hay những người nắm giữ bí mật. Giả sử ta có bí mật ban đầu là , ngưỡng t và cần chia bí mật ra thành n phần :

* + Chọn một số nguyên tố P ( P ≥ )
  + Trong trường số P, chọn giá trị ngẫu nhiên hoặc tùy ý cho các giá trị còn lại của điểm bí mật . Từ đây ta sẽ có điểm bí mật trong không gian t chiều.
  + Trong trường số P, ta sẽ chọn giá trị cho các hệ số của siêu phẳng sao cho thỏa hệ đồng dư :
* F(x) ≡ n (mod P) với

F(x) : phương trình siêu phẳng khuyết hệ số tự do

n : hệ số tự do của siêu phẳng đó

Tương tự như vậy ta sẽ tạo ra n siêu phẳng cho n cổ phần hay người giữ bí mật.

Sau khi mã hóa ta sẽ có được n siêu phẳng trong không gian t chiều tương ứng cho mỗi người giữ bí mật hay cổ phần. Chỉ khi có t hoặc nhiều hơn người giữ bí mật đồng ý chia sẻ bí mật với nhau thì bí mật ban đầu mới có thể khôi phục lại. Ta sẽ đi vào một ví dụ cụ thể để dễ dàng hình dung hơn về cách mã hóa của mô hình Blakley. Giả sử ta có:

* Bí mật ban đầu = 12,
* Số nguyên tố P = 23,
* Ngưỡng t = 3,
* Số phần bí mật n = 5.

Đầu tiên, ta sẽ chọn và cho điểm bí mật của ta : và Ta sẽ có điểm bí mật ban đầu là A ( 12, 20, 15 ). Tiếp theo, ta sẽ bắt đầu sử dụng điểm bí mật để tạo ra các phương trình đồng dư thỏa điểm bí mật ban đầu :

* x + y + z ≡ 1 ( mod 23 )
* x + 2y + 3z ≡ 5 ( mod 23 )
* x +3 y + 6z ≡ 1 ( mod 23 )
* x + 4y + 10z ≡ 12 ( mod 23 )
* x +5 y + 15z ≡ 15 ( mod 23 )

Ta cũng có thể hiểu thuật toán Blakley bằng theo ma trận :

× ≡ (mod 23)

Và ta đã có hoàn thành xong việc mã hóa theo thuật của Blakley. Giờ việc ta cần làm là giao những phần bí mật cho những người giữ bí mật.

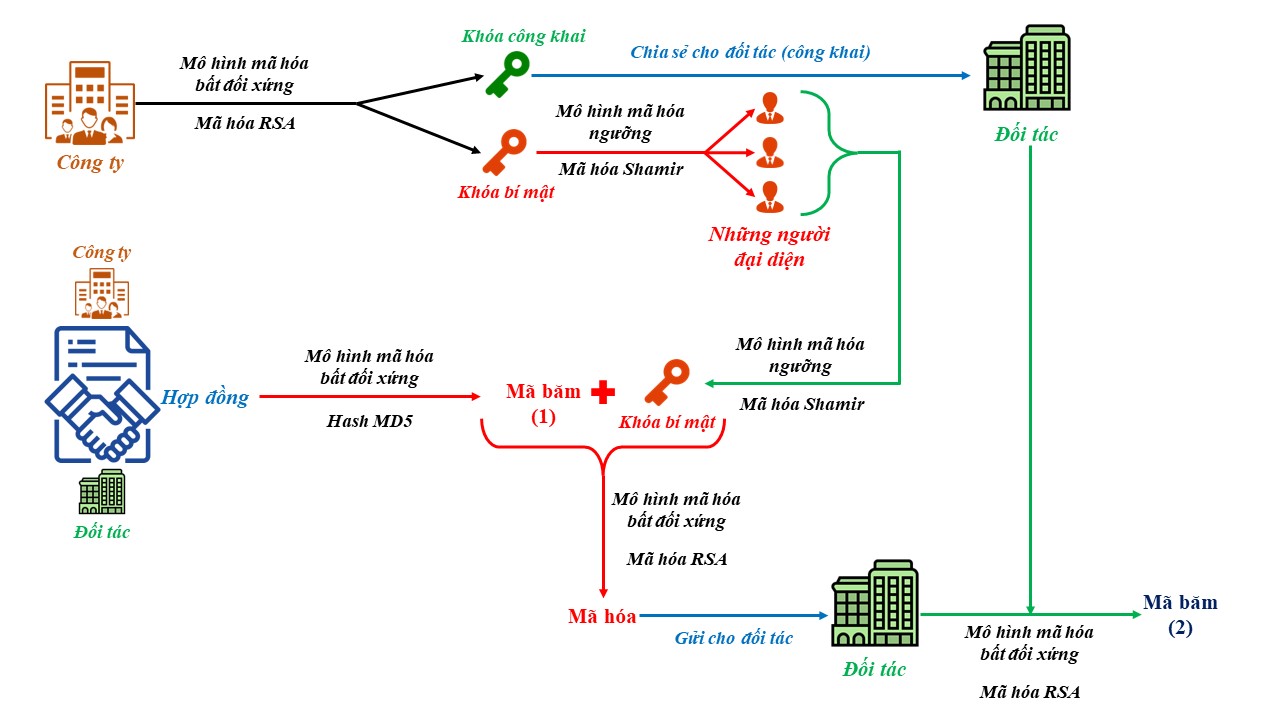
Lưu ý. Trong khi tạo ra các siêu phẳng, ta có thể chọn những hệ số ngẫu nhiên trong trường P. Nhưng để dễ dàng trong việc nghiên cứu cũng như việc thời gian trong lập trình minh họa, chúng ta sẽ sử dụng ma trận Pascal cho thuận tiện.

**3.2.3 Mã python**

Đường dẫn Github: https://github.com/HH214129/BlakleyYSC2023

**3.3. Ứng dụng vào mô hình chữ ký số**

**3.3.1 Mô hình và tính chất riêng**



*Hình 2. Sơ đồ mô hình chữ ký số có áp dụng mã hóa ngưỡng trong thực hiện hợp đồng*

Việc áp dụng *mô hình mã hóa ngưỡng (Threshold encryption model)* vào chữ ký số sẽ gia tăng thêm các tính chất bảo mật cho chữ ký số (thừa hưởng các tính chất từ mô hình này). Từ đó, tạo nên tính chất *đồng thuận tập trung* cho chữ ký số. Chỉ khi được đồng thuận bởi một ngưỡng người đại diện nhất định thì mới tìm được khóa bí mật để thực hiện ký số lên đoạn mã băm.

**3.3.2 Mã C/C++**

Đường dẫn Github: <https://github.com/phantonlocnguyen/CKS>

**3.3.3 Ứng dụng**

Việc mã hóa thông tin theo dạng “đầu – cuối” của mô hình trên có thể đảm bảo được an toàn thông tin trong việc gửi – nhận dữ liệu trên các giao thức truyền tin trực tuyến.

Ngày nay, có nhiều dịch vụ trực tuyến được cung cấp thông qua Internet, chẳng hạn như dịch vụ tài chính - ngân hàng điện tử, dịch vụ ký kết hợp đồng thông minh, dịch vụ thương mại điện tử… Đặc biệt, dịch vụ mạng xã hội là một dịch vụ ngày càng được phát triển và phổ biến trong đời sống, tuy nhiên, chúng có nhược điểm là thường chứa một lượng lớn các thông tin quan trọng của người dùng và những thông tin này truyền qua lại trên các giao thức mạng Internet giữa máy khách và máy chủ (Client – Server).

Do đó, áp dụng mô hình trên vào các hoạt động truyền gửi dữ liệu có thể đảm bảo tính an toàn của dữ liệu: thông tin được vẹn toàn, không bị đánh cắp hay sửa đổi trong quá trình chuyển đi, xác thực lại các thông tin sau khi nhận được, thông tin được mã hóa phân chia ra nhiều máy chủ khác nhau giúp việc tấn công đánh cắp thông tin từ một máy chủ thì không thể giải mã,…

Trong ký kết hợp đồng thông minh bằng chữ ký số, nhờ có tính chất đồng thuận tập trung mà từ đó việc ký kết một hợp đồng quan trọng phải cần có sự thống nhất của toàn bộ người trong hội đồng cấp cao của công ty, tránh một bên cá nhân tự ý quyết định. Việc tấn công lấy thông tin chữ ký số cũng trở nên phức tạp hơn do cần phải tấn công nhiều bên liên quan có nắm giữ chìa khóa.

Trong lĩnh vực sản xuất để bảo mật mã sản phẩm từ quá trình sản xuất cho đến khi nó đến tay người dùng ngăn ngừa được hàng giả, hàng kém chất lượng. Ngoài ra, còn có thể được ứng dụng trong việc bảo mật thư từ, văn bản pháp lý, xác thực thông tin người dùng,…

**4. Kết luận và hướng phát triển.**

Trong bài báo này, nhóm tác giả đã tìm hiểu định nghĩa và các tính chất của các mô hình chia sẻ dựa trên mã hóa ngưỡng và tìm hiểu ứng dụng của chúng vào bài toán chữ ký số. Bài báo đã tập trung vào việc nghiên cứu một số mô hình chia sẻ phổ biến như mô hình chia sẻ Shamir, mô hình chia sẻ Blakley và Định lý số dư Trung Hoa.

Mô hình chia sẻ Shamir là một phương pháp mã hóa ngưỡng được đề xuất bởi Adi Shamir vào năm 1979. Mô hình này cho phép chia sẻ một bí mật thành nhiều phần và chỉ cần một số lượng nhất định (gọi là ngưỡng) của các phần tử để tái tạo lại bí mật ban đầu. Điều này giúp bảo mật thông tin, vì người dùng cần sự hợp tác của nhiều bên để khôi phục bí mật.

Mô hình chia sẻ Blakley, được đề xuất bởi Robert C. Blakley vào năm 1979, là một phương pháp chia sẻ bí mật dựa trên nguyên tắc của đại số tuyến tính. Thay vì sử dụng đa thức như trong mô hình chia sẻ Shamir, mô hình chia sẻ Blakeley sử dụng ma trận và các phép tính trên ma trận để thực hiện việc chia sẻ và tái tạo bí mật.

Định lý số dư Trung Hoa là một định lý toán học quan trọng trong lĩnh vực mã hóa và hệ mật. Định lý này cho phép xác định một số lượng nhất định các phần tử trong một tập hợp con của các phần tử trong tập hợp lớn hơn, dựa trên các mệnh đề về phép chia và số dư. Định lý số dư Trung Hoa đã được áp dụng trong các mô hình chia sẻ mã hoá ngưỡng để đảm bảo tính toàn vẹn và bảo mật cho chữ ký số.

Bên cạnh việc nghiên cứu và phân tích các tính chất của các mô hình chia sẻ dựa trên mã hóa ngưỡng, nhóm tác giả cũng đã tiến hành mô phỏng lại các giải thuật được sử dụng trong các mô hình này bằng ngôn ngữ C/C++ và Python. Họ đã sử dụng các thư viện và chức năng phù hợp để hiện thực hóa các phép toán số học trên ma trận như phép cộng, phép nhân, và phép chia.Việc áp dụng mô hình mã hóa ngưỡng vào bài toán chữ ký số cung cấp thêm lợi ích bảo mật. Thay vì chỉ có một người duy nhất tạo ra chữ ký số, mô hình chia sẻ dựa trên mã hóa ngưỡng cho phép nhiều bên tham gia vào quá trình tạo ra và xác nhận chữ ký. Điều này tăng cường tính phân tán và khó khăn trong việc tấn công và giả mạo chữ ký số.

**TÀI LIỆU THAM KHẢO**

[1] A. Alpher, Frobnication, *Journal of Foo*, vol. 12, no. 1, pp. 234-778, 2002.

[2] A. Alpher and J. P. N. Fotheringham-Smythe, Frobnication revisited, *Journal of Foo*, vol. 13, pp. 234-778, 2003.

[3] A. Alpher, J. P. N. Fotheringham-Smythe, and G. Gamow, Can a machine frobnicate?, *Journal of Foo*, vol. 14, no. 1, pp. 234-778, 2004.

[4] V. Arnold, K. Vogtmann, and A. Weinstein, *Mathematical Methods of Classical Mechanics*, ser. Graduate Texts in Mathematics. Springer, 1989.

[5] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[6] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, and B. Stenger. (2012) Toshiba CAD model point clouds dataset. [Online].

Available: www.toshiba.eu/eu/Cambridge-Research-Laboratory/Computer-Vision-Group/Stereo-Points/

[7] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla, A new distance for scale-invariant 3D shape recognition and registration, in *Proc. Int. Conf. on Computer Vision*, 2011, pp. 145-152.

[8] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla, An image processing method and system, US Patent 20130016913 (A1), 2013.

[9] L. A. Santalo, Integral geometry and geometric probability, in *Encyclopedia of Mathematics and its Applications*, G. C. Rota, Ed. Addison-Wesley, 1976, vol. 1.