**NỘI DUNG THUYẾT TRÌNH**

**BẢO MẬT CƠ SỞ DỮ LIỆU – NHÓM 2 – TUẦN 5**

**MÃ HÓA BẤT ĐỐI XỨNG (RSA, ELGAMAL), CHỮ KÝ ĐIỆN TỬ**

**🙞---🕮---🙜**

***Thông tin học phần:***

*- Mã lớp: 2421COMP131102*

*- Giảng viên hướng dẫn: ThS. Nguyễn Văn Thịnh*

***Thành viên nhóm Thursday:***

*- Cao Võ Tuấn Kiệt – 49.01.104.076*

*- Vũ Nguyễn Việt Phương – 49.01.104.117*

*- Nguyễn Hữu Minh Quân – 40.01.104.120*

*- Nguyễn Ngọc Phú Tỷ – 49.01.104.172 (Nhóm trưởng)*

**NỘI DUNG**

[1. Mã hóa bất đối xứng 1](#_Toc192713855)

[1.1. Khó khăn của mã hóa đối xứng 1](#_Toc192713856)

[1.2. Khái niệm mã hóa bất đối xứng 1](#_Toc192713857)

[1.3. Cấu trúc hệ thống mật mã bất đối xứng 1](#_Toc192713858)

[1.4. Ứng dụng của mã hóa bất đối xứng 2](#_Toc192713859)

[1.5. Thuật toán mã hóa bất đối xứng phổ biến 2](#_Toc192713860)

[2. Thuật toán RSA 2](#_Toc192713861)

[2.1. Tổng quan thuật toán RSA 2](#_Toc192713862)

[2.2. Độ an toàn của thuật toán RSA 4](#_Toc192713863)

[2.3. Ưu điểm của thuật toán RSA 4](#_Toc192713864)

[2.4. Nhược điểm của thuật toán RSA 5](#_Toc192713865)

[3. Thuật toán ElGamal 5](#_Toc192713866)

[3.1. Tổng quan thuật toán ElGamal 5](#_Toc192713867)

[3.2. Độ an toàn của thuật toán ElGamal 7](#_Toc192713868)

[3.3. Ưu điểm của thuật toán ElGamal 7](#_Toc192713869)

[3.4. Nhược điểm của thuật toán ElGamal 7](#_Toc192713870)

[4. Chữ ký điện tử 7](#_Toc192713871)

[4.1. Định nghĩa chữ ký điện tử 7](#_Toc192713872)

[4.2. Phân loại các hình thức của chữ ký điện tử 7](#_Toc192713873)

[4.3. Chữ ký số 7](#_Toc192713874)

[4.4. Chữ ký RSA 10](#_Toc192713875)

[5. Code minh họa 12](#_Toc192713876)

[6. Tài liệu tham khảo 12](#_Toc192713877)

# 1. Mã hóa bất đối xứng

## 1.1. Khó khăn của mã hóa đối xứng

Trong mã hóa đối xứng, mỗi cặp người dùng cần phải có khóa bí mật để trao đổi thông tin. Giả sử có *n* người trong một hệ thống, mỗi cặp người phải có một khóa chung để trao đổi dữ liệu, do đó số lượng khóa cần thiết sẽ là  khóa, do đó rất khó quản lý, phức tạp và không an toàn.

Người dùng bên gửi và bên nhận sử dụng cùng một khóa bí mật để mã hóa và giải mã, tạo nên vấn đề về phân phối khóa, làm thế nào để hai bên có thể chia sẻ khóa đó một cách an toàn. Nếu khóa được trao đổi qua một kênh không bảo mật, kẻ tấn công có thể chặn lấy và sử dụng nó để giải mã thông tin, làm mất đi tính bảo mật của hệ thống.

Mã hóa bất đối xứng có thể giải quyết 2 vấn đề trên.

## 1.2. Khái niệm mã hóa bất đối xứng

Mã hóa bất đối xứng sử dụng một cặp khóa mã hóa gồm khóa công khai (public key) và khóa bí mật (private key). Dữ liệu được mã hóa bằng khóa công khai và chỉ có thể được giải mã bằng khóa bí mật.

Thuật toán mật mã bất đối xứng chủ yếu dựa trên các hàm toán học hơn là dựa vào các thao tác trên chuỗi bit. Mã hóa bất đối xứng hay còn được gọi bằng một tên thông dụng hơn là mã hóa dùng khóa công khai (public key encryption).

## 1.3. Cấu trúc hệ thống mật mã bất đối xứng

Các bước cơ bản của một hệ thống mật mã bất đối xứng gồm:

- Mỗi thực thể thông tin (user) tạo ra một cặp khóa (public, private) để dùng cho việc mã hóa và giải mã.

- Mỗi user thông báo một trong hai khóa của mình cho các user khác biết, khóa này được gọi là khóa công khai (public key *KU*). Khóa còn lại được giữ bí mật và được gọi là khóa bí mật (private key *KR*).

- Nếu một user A muốn gởi thông tin cho user B, user A sẽ thực hiện mã hóa thông tin cần gởi bằng khóa công khai của user B.

- Khi nhận được thông tin đã mã hóa từ user A, user B thực hiện giải mã thông tin đó bằng khóa riêng của mình. Do khóa riêng không phổ biến công khai nên chỉ có một mình user B có khả năng giải mã được.

A diagram of a key to a key

AI-generated content may be incorrect.

## 1.4. Ứng dụng của mã hóa bất đối xứng

Mật mã hóa bất đối xứng được sử dụng trong các ứng dụng: che giấu thông tin, tạo chữ ký số (digital signature) và trao đổi khóa trong các thuật toán mật mã đối xứng (key exchange).

Các kỹ thuật mã hóa bất đối xứng đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

## 1.5. Thuật toán mã hóa bất đối xứng phổ biến

- Thuật toán RSA: Dựa trên vấn đề phân tích ra thừa số nguyên tố các số nguyên lớn và bài toán RSA.

- Thuật toán ElGamal: Dựa trên bài toán logarit rời rạc, là 1 biến thể của sơ đồ phân phối khóa Diffie-Hellmal.

*\*Bài toán RSA:* Cho một số nguyên dương *n = p × q* (*p*, *q* là hai số nguyên tố lẻ), một số nguyên *b* sao cho *gcd(b, (p – 1) × (q – 1)) = 1* và một số *c*. Tìm số nguyên *x* sao cho *xb ≡ c (mod n)*.

*\*Bài toán logarit rời rạc:* Logarit rời rạc được định nghĩa là số nguyên *x* thỏa mãn phương trình *ax ≡ b (mod m)* với *a, b, m* cho trước.

# 2. Thuật toán RSA

## 2.1. Tổng quan thuật toán RSA

RSA là thuật toán mật mã bất đối xứng được xây dựng bởi Ron Rivest, Adi Shamir và Len Adleman tại viện công nghệ Massachusetts (MIT), do đó được đặt tên là Rivest – Shamir – Adleman hay RSA. Thuật toán này ra đời năm 1977 và cho đến nay đã được ứng dụng trong nhiều lĩnh vực. Cũng như các thuật toán mật mã bất đối xứng khác, nguyên lý của RSA chủ yếu dựa trên lý thuyết số chứ không dựa trên các thao tác xử lý bit.

*\*Các bước thực hiện:*

*-* Chọn hai số nguyên tố lớn *p* và *q*, tính *n = p × q*

- Tính *Ф(n) = (p – 1) × (q – 1)*

- Tìm một số *e* sao cho *gcd(e, Ф(n)) = 1* với *0 < e < Ф(n)*

- Tìm một số *d* sao cho *e × d ≡ 1 (mod Ф(n))* (hay *d = e-1 mod Ф(n)*) (*d* là nghịch đảo của *e* trong phép modulo *Ф(n)*

- Chọn khóa công khai *KU*là cặp *(e, n)*, khóa bí mật *KR* là cặp *(d, n)*.

- Mã hóa: *C = Me mod n*

- Giải mã: *M = Cd mod n*

*\*Ví dụ về cách vận hành:*

Tỷ chọn cặp số nguyên tố *p = 13* và *q = 5* để tạo cặp khóa RSA. Khi đó:

- Tính *n = p × q = 13 × 5 = 65*

- Tính *Ф(n) = (p – 1) × (q – 1) = 12 × 4 = 48*

- Chọn *e = 5* thỏa điều kiện *5* và *48* là cặp số nguyên tố cùng nhau

- Với *e = 5* ta xác định được *d = 29* vì *e × d ≡ 1 (mod 48)*

- Khóa công khai *KU = (e, n) = (5, 65)*

- Khóa bí mật *KR = (d, n) = (29, 65)*

Giả sử, Kiệt muốn gửi đoạn thông tin *M = 30* cho Tỷ dựa trên khóa công khai của Tỷ. Khi đó:

- Kiệt dùng khóa công khai của Tỷ *KU = (e, n) = (5, 65)*

- Kiệt mã hóa C = *Me mod n = 305 mod 65 = 10*

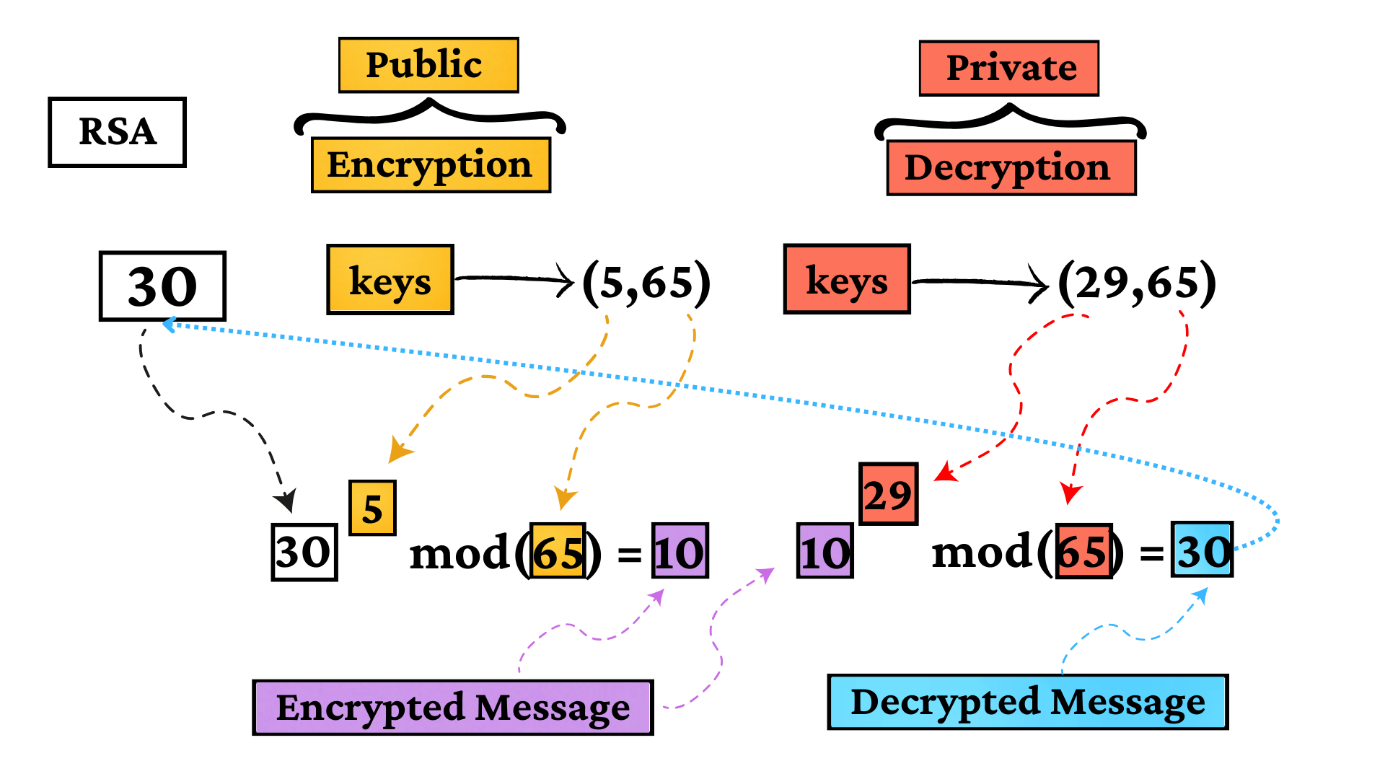
- Thông tin mật Kiệt gửi cho Tỷ là *C = 10*

Khi nhận được thông tin mật này, Tỷ giải mã bằng khóa bí mật của mình như sau:

- Tỷ dùng khóa bí mật của mình *KR = (d, n) = (29, 65)*

- Tính *M = Cd mod N = 1029 mod 65 = 30*

- Như vậy thông tin giải mã được là *M = 30* đúng với Kiệt đã gửi.



## 2.2. Độ an toàn của thuật toán RSA

Vì khóa là công cộng nên việc tấn công bẻ khóa phương pháp RSA thường dựa vào khóa công khai để xác định được khóa riêng tương ứng. Điều quan trọng là dựa vào *n* để tính *p*, *q* của *n*, từ đó tính được *d*. Như vậy, tính an toàn của phương pháp RSA dựa trên cơ sở các máy tính tại thời điểm hiện tại chưa đủ khả năng giải quyết việc phân tích các số nguyên rất lớn ra thừa số nguyên tố. Tuy nhiên, với sự phát triển ngày càng nhanh chóng của máy tính cũng như những bước đột phá trong lĩnh vực toán học, phương pháp RSA sẽ gặp phải những khó khăn trong việc bảo mật thông tin.

Theo lý thuyết, RSA có thể bị tấn công bằng những phương pháp sau đây:

- Brute-force attack: tìm lần lượt các khóa riêng.

- Mathematical attack: xác định *p* và *q* bằng cách phân tích *n* thành tích của các thừa số nguyên tố rồi từ đó xác định *e* và *d*.

- Timing attack: dựa trên thời gian thực thi của thuật toán giải toán.

- Chosen ciphertext attack: sử dụng các dụng các đoạn thông tin mật (ciphertext) đặc biệt để khôi phục thông tin gốc.

Trang web phân tích số *n* thành các thừa số nguyên tố với *n* cực lớn, truy vấn cực nhanh: <https://factordb.com/index.php>

## 2.3. Ưu điểm của thuật toán RSA

- Tính bảo mật cao, RSA dựa trên bài toán phân tích thừa số nguyên tố của các số rất lớn, khiến cho việc phá khóa nếu không có khóa riêng trở nên rất khó khăn.

- Dễ dàng trao đổi khóa, có thể trao đổi khóa một cách an toàn qua các kênh không bảo mật, vì chỉ khóa riêng mới có thể giải mã dữ liệu được mã hóa.

- Ứng dụng rỗng rãi, RSA được sử dụng không chỉ để mã hóa dữ liệu mà còn để tạo chữ ký số, đảm bảo tính xác thực và toàn vẹn của thông tin trong các giao dịch điện tử, email, và các ứng dụng bảo mật khác.

## 2.4. Nhược điểm của thuật toán RSA

- Tốc độ chậm do tính toán liên quan đến các số nguyên lớn và các phép toán phức tạp, RSA thường xử lý chậm hơn so với các thuật toán mã hóa đối xứng.

- Cần kích thước khóa lớn để đảm bảo mức độ bảo mật cao, làm tăng thời gian tính toán và kích thước của dữ liệu mã hóa, ảnh hưởng đến hiệu suất tổng thể.

- Có thể bị tấn công nếu việc tạo số ngẫu nhiên không đủ mạnh, sẽ dẫn đến khả năng bị tấn công và làm giảm độ bảo mật của hệ thống.

# 3. Thuật toán ElGamal

## 3.1. Tổng quan thuật toán ElGamal

Taher Elgamal – một nhà mật mã học người Ai Cập đề xuất hệ thống mật mã ElGamal vào năm 1984 dựa trên sự phức tạp của bài toán logarit rời rạc. Tính an toàn của hệ thống phụ thuộc vào độ khó của bài toán logarit. ElGamal là một biến thể của sơ đồ phân phối Diffie - Hellman.

*\*Thực hiện sinh khóa:*

- Sinh ngẫu nhiên một số nguyên tố lớn *p*

- Chọn *α* là phần tử sinh của

*\*Một phần tử α ∈ (với p là số nguyên tố) được gọi là phần tử sinh (phần tử nguyên thủy) nếu: Ɐ x ∈ , ∃ k ∈ Z sao cho x ≡ αk (mod p).*

- Chọn ngẫu nhiên một số nguyên *a*, với *1 ≤ a ≤ p – 2*, tính *β = αa mod p*

- Khóa công khai là *(p, α, β)*

- Khóa bí mật là *(p, a)*

*\*Thực hiện mã hóa:*

- Sử dụng khóa công khai *(p, α, β)* theo thuật toán trên

- Chọn bản rõ *M*, trong khoảng *[0, p – 1]*

- Chọn ngẫu nhiên một số nguyên *k* với *1 ≤ k ≤ p – 2*

- Tính *γ = αk mod p* và *δ = M × βk mod p*

- Nhận được bản mã là *C =* *(γ, δ)*

*\*Thực hiện giải mã:*

- Sử dụng khóa bí mật *(p, a)*

- Lấy bản rõ *M = γp – 1 – a × δ mod p*

*\*Theo định lý nhỏ Fermat, nếu p là một số nguyên tố và γ không chia hết cho p ta có: γp – 1 ≡ 1 (mod p) nên γp – 1 – a ≡ γ–a (mod p) vậy M = γ–a × δ mod p*

*\*Ví dụ về cách vận hành:*

Kiệt tiến hành sinh khóa như sau:

- Kiệt chọn số nguyên tố *p = 17* và phần tử sinh *α = 3* trong

- Chọn khóa bí mật *a = 5* và tính *β* = *αa mod p = 35 mod 17 = 5*

- Khóa công khai *KU = (p, α, β) = (17, 3, 5)*

- Khóa bí mật *KR = (p, a) = (17, 5)*

Giả sử, Phương muốn gửi đoạn thông tin *M = 8* cho Kiệt dựa trên khóa công khai của Kiệt. Khi đó:

- Phương dùng khóa công khai của Kiệt *KU = (p, α, β) = (17, 3, 5)*

- Chọn một số ngẫu nhiên *k = 6*

- Tính *γ = αk mod p* = *36 mod 17 = 15*

- Tính *δ = M × βk mod p = 8 × 56 = 16*

- Phương được bản mã *C =* *(γ, δ) = (15, 16)*

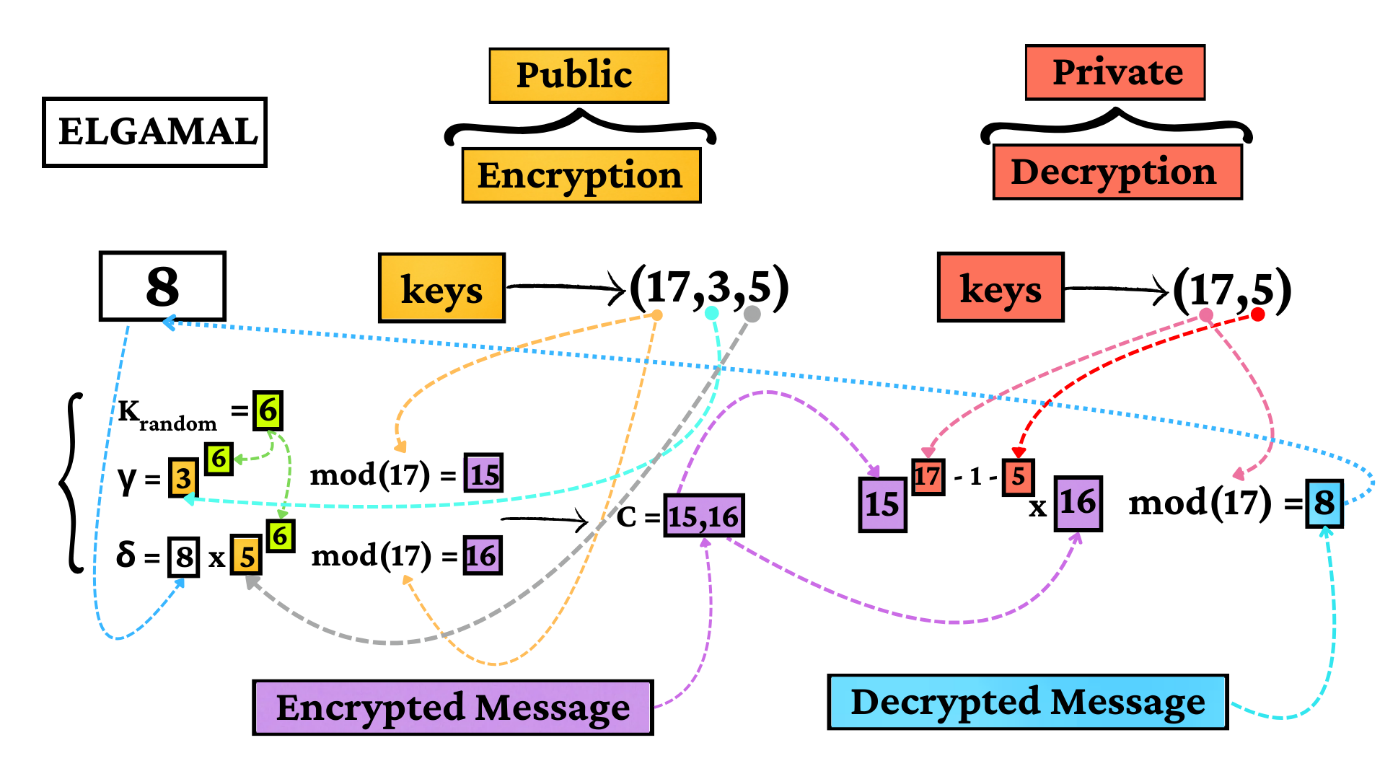
- Thông tin mật Phương gửi cho Kiệt là *C = (15, 16)*

Khi nhận được thông tin mật này, Kiệt giải mã bằng khóa bí mật của mình như sau:

- Kiệt sử dụng khóa bí mật của mình *KR = (p, a) = (17, 5)* để giải mã

- Kiệt tính *M = γp – 1 – a × δ mod p = 1511 × 16 mod 17 = 8*

-Như vậy thông tin giải mã được là *M = 8* đúng với Phương đã gửi



## 3.2. Độ an toàn của thuật toán ElGamal

- Thuật toán ElGamal áp dụng bài toán logarit rời rạc chính vì vậy độ an toàn của hệ mã hóa là rất lớn vì bài toán logarit rời rạc chưa có phương pháp hiệu quả để tính.

- Với 1 số *p* đủ lớn, thuật toán mã hóa ElGamal không có phương pháp thám mã hiệu quả.

## 3.3. Ưu điểm của thuật toán ElGamal

- Độ phức tạp của bài toán logarit lớn nên độ an toàn cao.

- Bản mã phụ thuộc vào bản rõ và giá trị ngẫu nhiên nên từ 1 bản rõ có thể có nhiều bản mã khác nhau.

## 3.4. Nhược điểm của thuật toán ElGamal

- Tốc độ chậm do phải xử lý số nguyên lớn.

- Dung lượng bộ nhớ dành cho việc lưu trữ khóa lớn.

# 4. Chữ ký điện tử

## 4.1. Định nghĩa chữ ký điện tử

Chữ ký điện tử là chữ ký được tạo lập dưới dạng dữ liệu điện tử gắn liền hoặc kết hợp một cách lô gíc với thông điệp dữ liệu để xác nhận chủ thể ký và khẳng định sự chấp thuận của chủ thể đó đối với thông điệp dữ liệu. *(Theo luật số 20/2023/QH15 ngày 22/06/2023)*.

Chữ ký điện tử được sử dụng trong các giao dịch điện tử. Xuất phát từ thực tế, chữ ký điện tử cũng cần đảm bảo các chức năng như xác định được người chủ của một dữ liệu nào đó (văn bản, ảnh, video,…) và xác định xem dữ liệu đó có bị thay đổi sau khi ký hay không.

## 4.2. Phân loại các hình thức của chữ ký điện tử

- Chữ ký số: Chữ ký số là một dạng chữ ký điện tử. Được tạo ra thông qua các thuật toán mật mã không đối xứng. Chữ ký số giúp đảm bảo tính toàn vẹn của thông điệp dữ liệu, vì mọi thay đổi sau khi ký đều có thể được phát hiện, từ đó ngăn chặn việc giả mạo.

- Chữ ký scan: Là loại chữ ký được chuyển thành dạng điện tử sau khi thực hiện ký tay trên giấy thông qua máy quét (scan) rồi gửi qua thư điện tử. Đây là dạng chữ ký điện tử phổ biến trong các giao dịch có yêu cầu bảo mật không quá cao.

- Chữ ký hình ảnh: Là loại chữ ký được ký bằng tay sau đó chuyển thành dạng ảnh và chèn vào hợp đồng điện tử.

## 4.3. Chữ ký số

Hiện nay có nhiều định nghĩa về chữ ký số theo khía cạnh và quan điểm nghiên cứu khác nhau. Tuy nhiên, theo Nghị Định 130/2018/NĐ-CP, ngày 27/9/2018 của Chính phủ Việt Nam, chữ ký số được định nghĩa là một loại chữ ký điện tử, được tạo bằng sự chuyển đổi thông điệp dữ liệu sử dụng một hệ thống mật mã không đối xứng, theo đó người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác:

- Việc biến đổi nêu trên được tạo ra bằng đúng khóa bí mật tương ứng với khóa công khai trong cùng một cặp khóa;

- Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên.

*\*Nguyên lý hoạt động:* dựa trên hệ mã hóa bất đối xứng, mỗi người cần 1 cặp khóa công khai và khóa bí mật.

- Khóa bí mật dùng để tạo chữ ký số.

- Khóa công khai dùng để thẩm định, xác thực chữ ký.

Thông thường, thay vì ký số trực tiếp toàn bộ thông điệp (có thể rất lớn), ta ký số giá trị băm có kích thước nhỏ và cố định. Hàm băm giúp chuyển đổi thông điệp có độ dài bất kỳ thành một chuỗi có độ dài có định, Nếu thông điệp bị thay đổi dù chỉ một chút, giá trị băm sẽ hoàn toàn khác, giúp người nhận phát hiện ngay bất kỳ sự thay đổi nào trong nội dung.

Việc ký trực tiếp các tập dữ liệu lớn này bằng khóa riêng sẽ tốn nhiều công sức tính toán và không hiệu quả. Thay vào đó, hàm băm được sử dụng để cô đọng thông báo thành giá trị băm có kích thước cố định, nhỏ hơn nhiều. Sau đó, người ký sẽ ký giá trị băm này thay vì toàn bộ thông báo, làm giảm đáng kể chi phí tính toán và cải thiện hiệu suất của quá trình chữ ký số.

A diagram of a private key signing a document

AI-generated content may be incorrect.

*\*Một sơ đồ chữ ký số* là bộ 5 *(M, A, K, S, V)* thoả mãn các điều kiện dưới đây:

- *M* là tập hữu hạn các bức điện (thông điệp, bản rõ) có thể.

- *A* là tập hữu hạn các chữ ký có thể.

- *K* là tập không gian khoá (tập hữu hạn các khoá có thể). Mỗi khóa *k ∈ K* gồm có hai phần *k = (kR, kU)*, trong đó *kR* là khóa bí mật dùng để ký, còn *kU* là khóa công khai dùng để kiểm thử chữ ký.

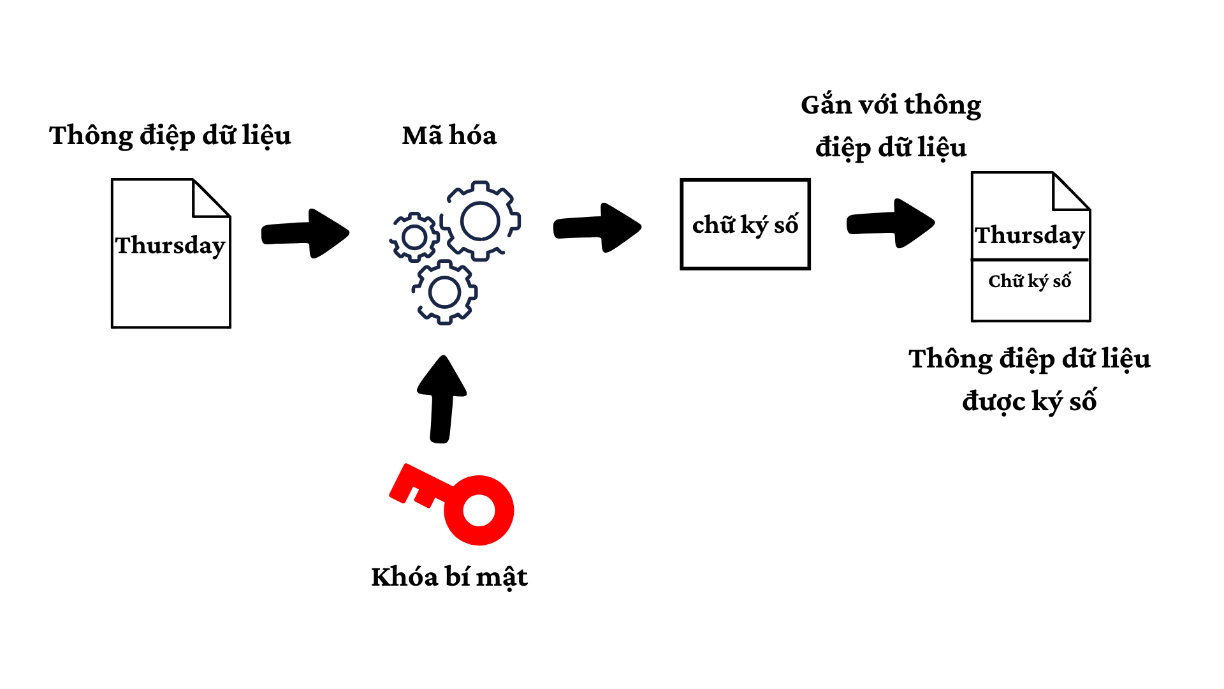
- Với mỗi khóa *k ∈ K* ta có:

+ Thuật toán ký (trong S) : *M → A*

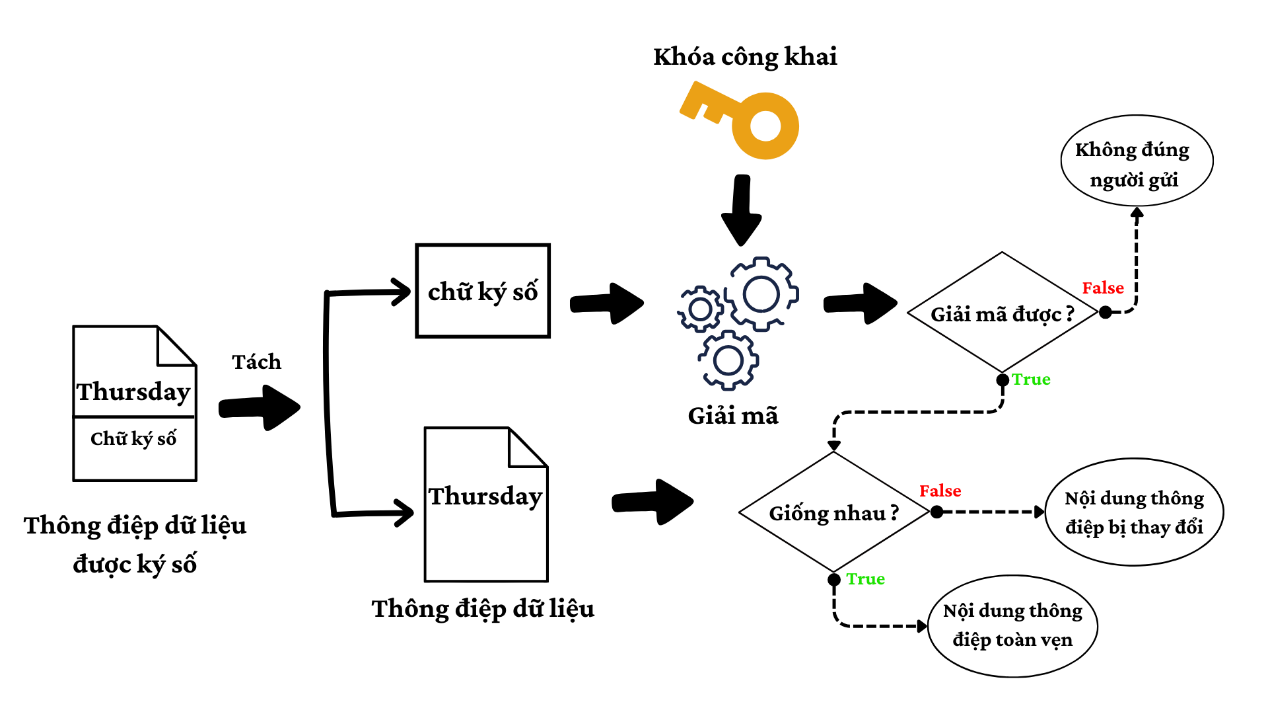
+ Thuật toán kiểm thử (trong V) : *M × A → {True, False}* thỏa mãn điều kiện sau với mọi bức điện *x ∈ M* và mọi chữ ký *y ∈ A:*



*\*Thủ tục ký:* Người A (người thực hiện ký) tạo ra chữ ký cho văn bản *m ∈ M* bằng việc thực hiện tính *s = S(m)*, sao đó truyền cặp *(m, s)* cho người B, trong đó *s* được gọi là chữ ký của văn bản *m*.



*\*Thủ tục kiểm thử:* Việc kiểm thử chữ ký *s* trên văn bản *m* do A tạo ra, người B (người thực hiện kiểm thử) thực hiện bằng cách sử dụng hàm kiểm thử *V*, tính *u = V(m, s)*. Người B sẽ chấp nhận chữ ký được tạo bởi A nếu *u = True* và sẽ từ chối khi *u = False*.



*\*Chức năng của chữ ký số:*

- Xác thực được nguồn gốc thông điệp: Tùy thuộc vào từng thông điệp mà có thể thêm các thông tin nhận dạng như tên tác giả, nhãn thời gian,…

- Tính toàn vẹn của thông điệp: Khi có sự thay đổi bất kỳ vô tình hay cố ý lên thông điệp thì giá trị hàm băm sẽ bị thay đổi và kết quả kiểm tra sẽ cho kết quả không đúng hay nói rằng thông điệp không toàn vẹn.

- Chống từ chối thông điệp: Vì chỉ có chủ thông điệp mới có khóa riêng để ký lên thông điệp nên người ký không thể chối bỏ thông điệp của mình.

## 4.4. Chữ ký RSA

Dựa vào ưu điểm của thuật toán mã hóa bất đối xứng RSA, nếu thiết lập được sơ đồ chữ ký dựa trên bài toán phân tích ra thừa số nguyên tố thì độ an toàn của chữ ký là rất cao.

Việc thiết lập sơ đồ xác thực chữ ký RSA rất đơn giản, ta chỉ cần đảo ngược hàm mã hóa với giải mã.

*\*Thực hiện sinh khóa:*

*-* Chọn hai số nguyên tố lớn *p* và *q*, tính *n = p × q*

- Tính *Ф(n) = (p – 1) × (q – 1)*

- Tìm một số *e* sao cho *gcd(e, Ф(n)) = 1* với *0 < e < Ф(n)*

- Tìm một số d sao cho *e × d ≡ 1 mod Ф(n)*

- Chọn khóa ký *KR*là cặp *(d, n)*, khóa kiểm thử *KU* là cặp *(e, n)*

*\*Thực hiện ký:*

- Sử dụng khóa ký *KR*= *(d, n)* theo thuật toán trên

- Chọn một bản thông điệp *m*, trong khoảng *[1, n – 1]*

- Tính *s = md mod N*

- Nhận được chữ ký *(s)*

- Gửi bản rõ kèm chữ ký *(m, s)*

*\*Thực hiện kiểm thử:*

- Lấy khóa kiểm thử *KU* = *(e, n)* được công khai

- Tính *a* = *se mod n*

- Nếu *a ≡ m ↔ True*, ngược lại *↔ False*

*\*Ví dụ về cách vận hành:*

Phương chọn 2 số nguyên tố *p = 3* và *q = 11*. Tạo cặp khóa RSA như sau:

- Tính *n = p × q = 3 × 11 = 33*

- Tính *Ф(n) = (p – 1) × (q – 1) = 2 × 10 = 20*

- Chọn *e = 7* thỏa điều kiện *7* và *20* là cặp số nguyên tố cùng nhau

- Với *e = 7* ta xác định được *d = 3* vì *e × d = 21 ≡ 1 (mod 20)*

- Khóa kiểm thử (công khai) *KU = (e, n) = (7, 33)*

- Khóa ký (bí mật) *KR = (d, n) = (3, 33)*

Trước khi gửi thông điệp *M = 7* cho Quân, Phương ký số lên thông điệp như sau:

- Sử dụng khóa ký *KR = (3, 33)*

- Thực hiện ký *S = Md mod n = 73 mod 33 = 13*

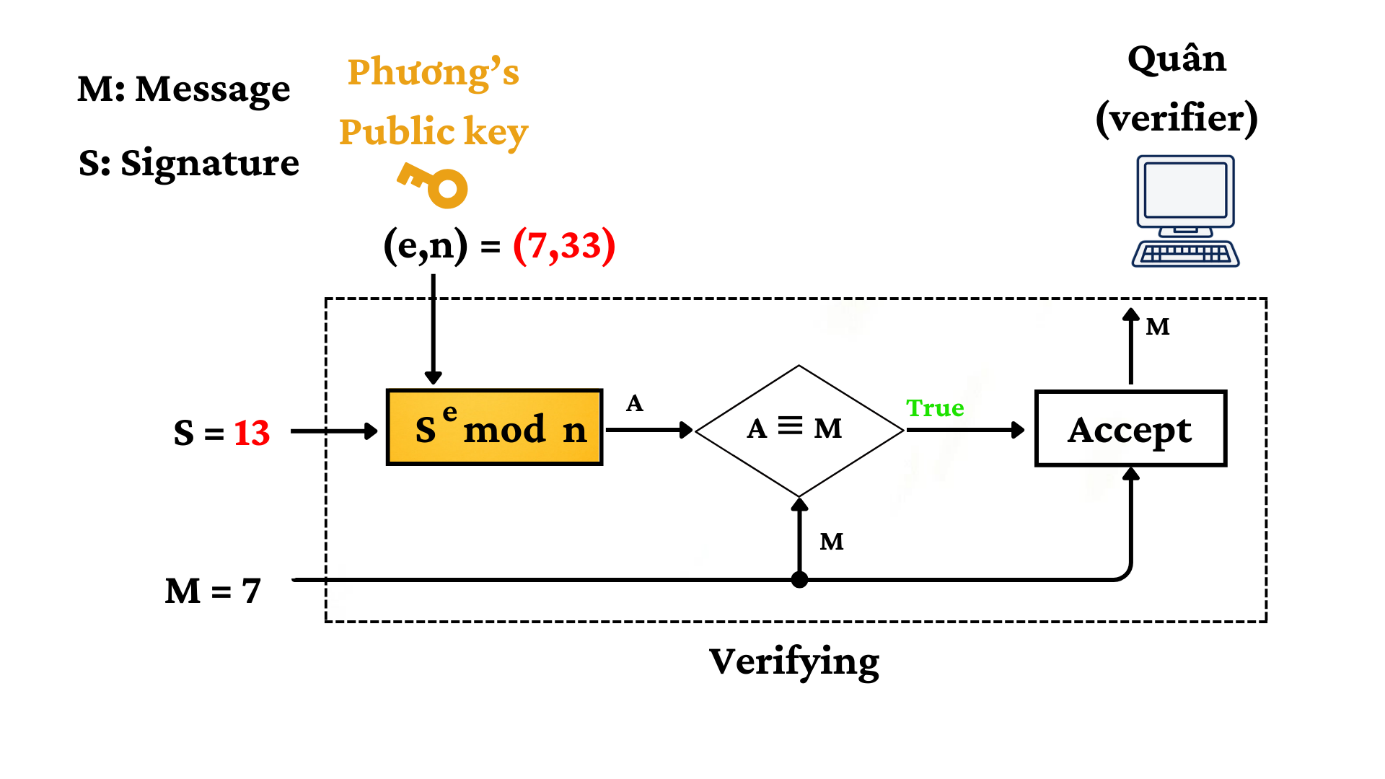
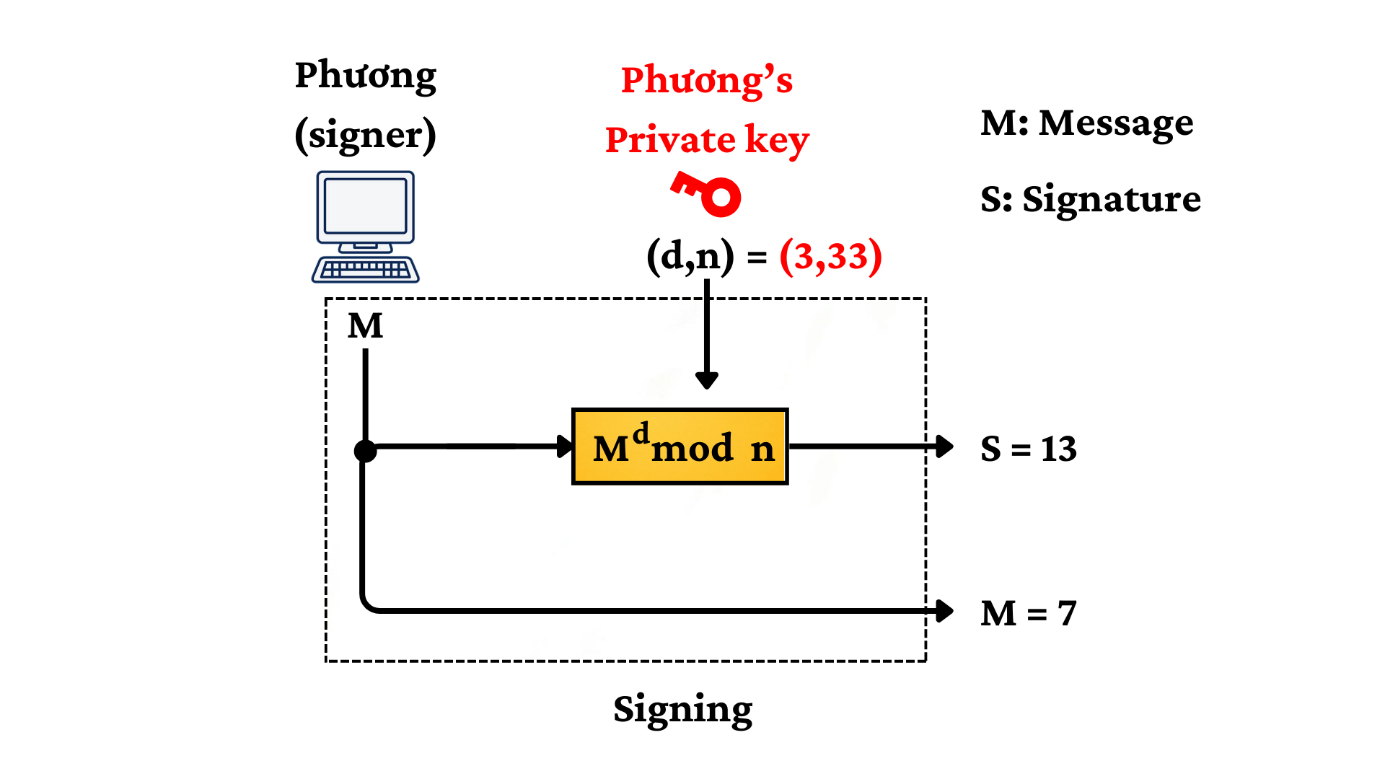
- Gửi đi thông điệp kèm chữ ký *(M, S) = (7, 13)*

Sau khi Quân nhận được thông điệp và khóa kiểm thử, Quân tiến hành kiểm thử:

- Sử dụng khóa công khai *KU = (7, 33)*

- Quân giải mã *A = Se mod n = 137 mod 33 = 7*

- Quân nhận thấy *M = A = 7* do đó chữ ký hợp lệ, thông điệp Phương gửi cho Quân là toàn vẹn.



# 5. Code minh họa

- Minh họa mật mã RSA: <https://github.com/tynnp/rsa-crypto>

- Minh họa mật mã ElGamal: <https://github.com/tynnp/elgamal-crypto>

- Minh họa chữ ký số RSA: <https://github.com/tynnp/rsa-signature>

# 6. Tài liệu tham khảo

[1] Tùng, T. T. T. (2011). Giáo trình mật mã học và an toàn thông tin.

[2] Dũng, L. H., Dũng, T. T., & Thắng, V. T. Phát triển thuật toán mật mã ElGamal.

[3] Bùi, V. B. (2017). *Nghiên cứu hệ mật mã RSA và ứng dụng trong chữ kí điện tử* (Doctoral dissertation, Học viện công nghệ Bưu chính viễn thông).

[4] Dũng, L. H. (2014). Phát triển thuật toán mật mã khóa công khai dựa trên hệ mật Elgamal. *Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng CNTT&TT*

[5] Trung, T. P. (2017). *Xây dựng và đánh giá hệ mật Affine–Elgamal* (Doctoral dissertation, Học viện công nghệ bưu chính viễn thông).

[6] Trường, N. V. (2013). Nghiên cứu phát triển chữ ký điện tử dựa trên hệ mật mã hóa khóa công khai RSA. *TẠP CHÍ KHOA HỌC ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH-KỸ THUẬT VÀ CÔNG NGHỆ*.

[7] Phạm, H. V., & Lưu, D. H. (2020). Phát Triển Một Dạng Lược Đồ Chữ Ký Số Mới Dựa Trên Bài Toán RSA: Array. *Journal of Science and Technology on Information and Communications*.

[8] Phúc, L. (2007). Bài giảng Bảo mật hệ thống thông tin. *Học viện công nghệ bưu chính viễn thông cơ sở Tp. HCM.*

[9] Ms IT. “AN TOÀN BẢO MẬT THÔNG TIN”. YouTube playlist. n.d. [youtube.com/playlist?list=PL\_jAnFPw7i2GQoIpMSXcitacj\_3qDcD16](https://www.youtube.com/playlist?list=PL_jAnFPw7i2GQoIpMSXcitacj_3qDcD16)

[10] Đỗ Văn Đức – ĐH GTVT. “An toàn bảo mật thông tin”. YouTube playlist. n.d. [youtube.com/playlist?list=PL1MYBVskDVJVKQKMx-9rtj1KXQgDW93W3](https://www.youtube.com/playlist?list=PL1MYBVskDVJVKQKMx-9rtj1KXQgDW93W3)