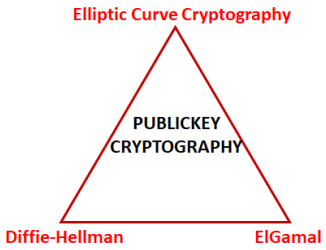


4 HỆ MÃ ĐƯỜNG CONG ELLIPTIC

HỆ MÃ TỰA ELGAMAL

Giao thức Diffie-Hellman

Trước khi bắt đầu, ta nhắc lại giao thức thiết lập khóa Diffie-Hellman (DH key exchange protocol), mà độ bảo mật dựa trên bài toán khó log-rời rạc (**DLP** – *Discrete Logarithm Problem*).

Allice	Bob
. p ← primeGen(λ) #Allice và Bob quy ước dùng chung một trường $\mathbb{Z}_p$ và một phần tử sinh g	
Bước 1: thiết lập khóa chung – **key exchange** . $x \xleftarrow{\$} \mathbb{Z}$ #chọn ngẫu nhiên một số ngẫu nhiên . $k_A \leftarrow g^x \% p$ (giữ bí mật x và chuyển k_A cho Bob) . $k \leftarrow k_B^x \% p$	. $y \xleftarrow{\$} \mathbb{Z}$. $k_B \leftarrow g^y \% p$ (giữ bí mật y và chuyển k_B cho Allice) . $k \leftarrow k_A^y \% p$
Sau bước này, Allice và Bob có chung khóa bí mật $k \equiv g^{xy} \pmod p$	

Như ta thấy, mục tiêu của giao thức-DH là thiết lập một bí mật chung giữa các đối tác. Không phải là hệ mã hóa/giải mã như RSA. ElGamal đã phát triển giao thức-DH thành một hệ mã, mô hình này có thể được dùng để thiết lập các hệ mã khác khi ta chọn hay xây dựng được một bài toán khó và sử dụng trực tiếp nó kiểu Diffie-Helman thay vì hông trực tiếp như kiểu RSA. Ta sẽ gọi chung là hệ mã tựa-ElGamal (ElGamal-like).

Hệ mã ElGamal

Trong mô hình này, giả sử Bob muốn gửi thông điệp mật của thông điệp rõ m cho Allice. Bob và Allice thi hành giao thức sau trên trường $\mathbb{Z}_p$, với p là số nguyên tố lớn:

Allice	Bob
Bước 1: sinh khóa – **Key generating** . p ← primeGen(λ) #tạo số nguyên tố p . g ← generator(p) #chọn phần tử sinh g . $d \xleftarrow{\$} \mathbb{Z}_p$ #chọn ngẫu nhiên 1 khóa cá nhân d . e ← g^d % p #tính và công bố khóa e	Cặp khóa (e, d) với (e, p, g) được công bố và d giữ bí mật.
x ngẫu nhiên làm cho hệ mã có tính an toàn ngữ nghĩa, nghĩa là cùng bản rõ, các lần mã khác nhau sẽ cho bản mã khác nhau.	**Bước 2**: mã hóa – **Encryption** . $x \xleftarrow{\$} \mathbb{Z}_p$ #chọn ngẫu nhiên số nguyên x . $c_1 \leftarrow g^x \pmod p$

	. $c_2 \leftarrow m*e^x \pmod p$. C ← (c_1, c_2) #tính C và gửi C cho Allice
Bước 3: giải mã – **Decryption** . $u \leftarrow c_1^d \pmod p$. $v \leftarrow u^{-1} \pmod p$. m ← c_2*v (mod p)	Ta thấy $c_2 * v \equiv (m * e^x) * u^{-1} \equiv m * (g^d)^x (c_1^d)^{-1}$ $\equiv m * g^{dx}((g^x)^d)^{-1} \equiv m * g^{dx}(g^{dx})^{-1}$ $\equiv m \pmod p$

Độ an toàn của ElGamal

Độ an toàn của hệ mã dựa trên giả thuyết Diffie-Hellman: DLP là bài toán khó.

Tuy nhiên, trong trường hợp x bị sử dụng lại, ElGamal có thể bị phá vỡ. Thực vậy, giả sử $C = (c_1, c_2) = (g^x, m * g^{dx}), và\ C' = (c_1', c_2') = (g^x, m' * g^{dx})$. Nếu biết m, c_1 và c_2, ta có thể suy ra m' như sau:

$$(m^{-1} * c_2)^{-1} * c_2' = \left(m^{-1} * (m * g^{dx})\right)^{-1} * (m' * g^{dx}) = (g^{dx})^{-1} * m' * g^{dx} = m'.$$

Như vậy, để an toàn, hàm sinh số ngẫu nhiên phải thực sự ngẫu nhiên.

Mô hình hệ mã tựa-ElGamal

Qua xem xét quy trình xây dựng hệ mã ElGamal đựa trên cấu trúc $(\mathbb{Z}_p, *)$, ta thấy, để xây dựng một hệ mã theo kiểu ElGamal, gọi là tựa-ElGamal (ElGamal-like) ta thực hiện các bước sau:

Bước 1: Xây dựng được một cấu trúc đại số (G, ⊗) sao cho có thể định nghĩa được bài toán khó kiểu log-rời rạc (DLP-like). Nghĩa là cho phần tử sinh g ∈ G và y = g ⊗…⊗ g = g^x ∈ G, với x là một số nguyên bí mật, thì tìm lại x từ g và y là bài toán khó.

Bước 2: Nhúng thông điệp, là một số nguyên thành một phần tử m thuộc G.

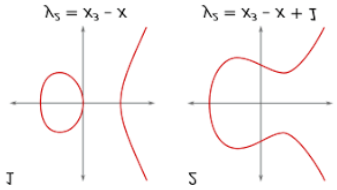
Bước 2: Sử dụng lược đồ ElGamal trên (G, ⊗) ta được hệ mã tựa-ElGamal. Nghĩa là với thông điệp nhúng m, khóa công khai e, phần tử sinh g, thì bản mã của m là cặp C = (g^x, m⊗e^x). Bản mã C = (c_1, c_2) ∈ G×G được giải mã thành lại thông điệp nhúng m = c_2⊗$((c_1)^d)^{-1}$.

Ta sẽ sử dụng quy trình xây dựng ElGamal-like này để khảo sát hệ mã đường cong elliptic.

HỆ MÃ ĐƯỜNG CONG ELLIPTIC

Đường cong elliptic

Đường cong elip E trên trường hữu hạn Fq là tập hợp tất cả các điểm (x, y) nằm trên đường cong (E): $y^2 = x^3 + ax + b$, trong đó a, b thuộc một truongf F_q và thỏa $4a^3 + 27b^2 \neq 0$. Hình dưới minh họa 2 đường cong elliptics đặc biệt.



Phép cộng điểm trên đường cong elliptic

Trên tập E = {(x, y) ∈ $\mathbb{Z}_p \times \mathbb{Z}_p$: $y^2 = x^3 + ax + b$} ∪ {O}, ta định nghĩa phép toán 2 ngôi +: E × E → E như sau.

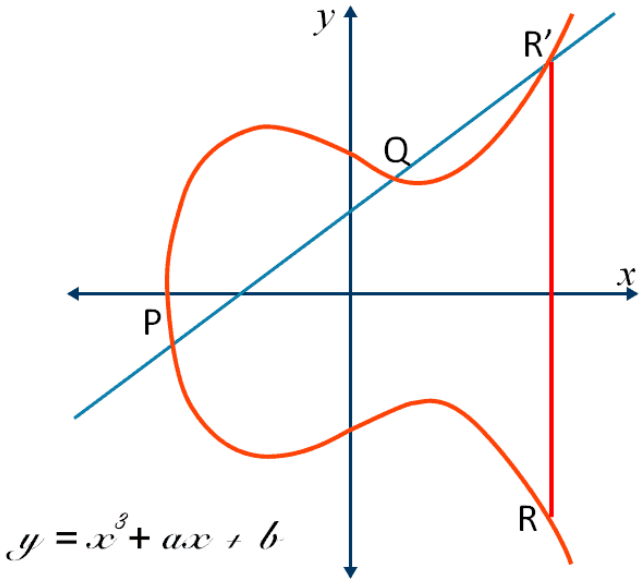
. Nếu P = (x_1, y_1) ∈ E thì –P = (x_1, -y_1).

. Nếu Q = (x_2, y_2) ∈ E, Q = –P thì R = Q + P = (x_3, y_3), trong đó

$x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda*(x_1 - x_3) - y_1$ và

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{nếu } P \neq Q \\ \frac{3x_1^2 + a}{2y_1}, & \text{nếu } P = Q \end{cases}.$$

Hình dưới minh họa quy tắc phép + điểm trên đường cong elliptic.



Ví dụ minh họa

Xét đường cong (E): $y^2 = x^2 + ax + b = x^3 + x + 1$ xác định trên $\mathbb{Z}_{13}$ với phần tử sinh (generator) của $\mathbb{Z}_{13}$ là G = (0,1). Các điểm trên E có thể được biểu diễn the G.

VD1. Ký hiệu G = (0, 1) = (x_1, y_1) thì R = P + P = 2P = (x_3, y_3) được xác định theo

. $\lambda = \frac{3x_1^2 + a}{2y_1}$, $(do\ P = Q)$

$\equiv \frac{3*0^2+1}{2*1} \equiv \frac{1}{2} \equiv 2^{-1} \equiv 7\ (mod\ 13)$.

. $x_3 = \lambda^2 - x_1 - x_2 \equiv 7^2 - 0 - 0 \equiv 10$ (mod 13).

. $y_3 = \lambda(x_1 - x_3) - y_1 \equiv 7(0 - 10) - 1 \equiv -69 \equiv 9$ (mod 13).

Vậy R = 2P = (10, 9).

VD2. Với P = (x_1, y_1) = (10, 9), Q = (x_2, y_2) = (1, 2), thì R = P + Q = (x_3, y_3) được xác đinh như sau:

. $\lambda = \frac{y_2-y_1}{x_2-x_1} \equiv \frac{2-9}{1-10} \equiv \frac{-7}{-9} \equiv \frac{6}{4} \equiv 6 * 4^{-1} \equiv 6 * 10 \equiv 60 \equiv 8.$

. $x_3 = \lambda^2 - x_1 - x_2 \equiv 8^2 - 10 - 1 \equiv 64 - 11 \equiv 1 \pmod{13}$.

. $y_3 = \lambda(x_1 - x_3) - y_1 \equiv 8(10 - 1) \equiv 8*9 \equiv 7$.

Vậy (10, 9) + (1, 2) = (1, 7).

Lưu ý

Khóa ECC 160 bit có độ an toàn tương đương với khóa RSA 1024 bit.

Hệ mã tựa-Elgamal trên đường cong elliptic

Các đường cong elliptic tạo thành các nhóm đơn sinh (cyclic group) và điểm G trên đường sinh ra mọi điểm của nhóm con đơn sinh rất hữu ích trong mật mã. Hệ mã dựa trên đường cong elliptic được gọi là hệ mã đường cong elliptic (ECC – Elliptic Curve Cryptosystem). Ký hiệu E = {(x, y) $\in \mathbb{Z}_p \times \mathbb{Z}_p$: $y^2 = x^3 + ax + b$} $\cup$ {O}, với O là phần tử trung hòa ở vô cực. (E, $\oplus$) được chứng minh là hình thành nhóm giao hoán, được gọi nhóm các đường cong elliptic. Hơn nữa, bài toán log-rời rạc trên nhóm đường cong elliptic (**EDLP** – Elliptic Discrete Logarithm Problem) với mô tả "cho điểm sinh G $\in$ E và Y = G $\oplus$…$\oplus$ G = xG với x là số nguyên bí mật, tìm x" cũng được chứng minh là bài toán khó, đựa trên định lý về số điểm trên đường cong elliptics.

Định lý Hasse. Số điểm trên đường cong elliptic $E(F_{p^k})$ xác định trên trường F_{p^k} là $\#E(F_{p^k}) = p^k + 1 + t_{p^k}$, với t_{p^k} là số thỏa bất đẳng thức $|t_{p^k}| \leq 2p^{\frac{k}{2}}$.

Ví dụ, $\#E(F_2) = 5$. Vì thế $E(F_2)$ không thể dùng cho mật mã.

Như vậy, bằng cách thay ký hiệu a*b, g^x trong $(\mathbb{Z}_p, *)$ lần lượt thành thành P+Q, xG trong (E, +), ta có hệ mã ElGamal trên đường cong elliptic.

Allice	Bob
Bước 1: sinh khóa – **Key generating** . p ← primeGen(λ)#tạo số nguyên tố p cho $\mathbb{Z}_p$. G ← generator(E) #chọn điểm sinh G . $d \xleftarrow{\$} \mathbb{Z}_p$ #chọn ngẫu nhiên 1 khóa cá nhân d . e ← dG #tính và công bố khóa e	Cặp khóa (e, d) với (e, $\mathbb{Z}_p$, G) được công bố và d giữ bí mật.
	Bước 2: mã hóa – **Encryption** . $x \xleftarrow{\$} \mathbb{Z}_p$ #chọn ngẫu nhiên số nguyên x . c_1 ← xG

x ngẫu nhiên làm cho hệ mã có tính an toàn ngữ nghĩa, nghĩa là cùng bản rõ, các lần mã khác nhau sẽ cho bản mã khác nhau.	. $c_2 \leftarrow m + xe$. $C \leftarrow (c_1, c_2)$ #tính C và gửi C cho Allice
Bước 3: giải mã – **Decryption** . $m = c_2 + (-dc_1)$	$c_2 + (-dc_1) = m + xe + (-d(xG))$ $= m + xe + (-dxG) = m + xdG + (-xdG) = m.$

Nhúng thông điệp m vào đường cong elliptic

Mục đích là để có thể thực hiện mã hóa thông điệp bằng ECC trong một trường F_q. Cụ thể, muốn nhúng thông điệp rõ thành một điểm trên đường cong elliptic xác định trên trường F_q, giả sử $q = p^r$ và p là số nguyên tố. Xem thông điệp m như một số nguyên $0 \le m \le M$, và chọn k là một số nguyên, lower $\le k \le$ upper. Với đường cong (E): $y^2 = x^3 + ax + b$ xác định trên F_q. Thông điệp m được nhúng như sau:

(1) Tính $x = mk + j$, $j = 0, 1, 2,\ldots$ cho đến khi tìm được $x^3 + ax + b$ là số chính phương (mod p), trả về điểm $(x, \sqrt{x^3 + ax + b})$ trên (E).

(2) Để chuyển điểm (x, y) trên (E) lại thành thông điệp m, tính $m = \lfloor x/k \rfloor$.

Ví dụ. Cho (E): $y^2 = x^3 + 3x$, và m = 2174, p = 4177.

Chọn k = 30 và tính $x = \{30\times2174 + j, j = 0, 1, 2,\ldots\}$ cho đến khi $x^3 + 3x$ là số chính phương. Khi j = 15, ta có

$x = 30 \times 2174 + 15 = 65235.$

$x^3 + 3x = (30 \times 2174 + 15)^3 + 3(30 \times 2174 + 15) = 277614407048580$

$\equiv 1444 \equiv 38^2 \pmod{4177}$.

Vậy thông điệp m = 2174 được nhúng thành điểm P(x, y) = P(65235, 38).

Để chuyển thông điệp nhúng P(65235, 38) về lại thông điệp gốc, ta tính

$m = \lfloor 65235/30 \rfloor = 2174.$

CÀI ĐẶT HỆ MÃ ĐƯỜNG CONG ELLIPTIC

Đường cong hữu ích

Không phải mọi đường cong elipptic đều có thể dùng tốt cho mật mã. Vì vậy, trong thực tiễn cài đặt, cần biết loại đường cong elliptic nào có ích trong mật mã học. Đường cong elliptic

(E): $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$,

với các hệ số thỏa biệt thức $\Delta \neq 0$, trong đó

$\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$,

với

$b_2 = a_1^2 + 4a_2$,

$b_4 = 2a_4 + a_1a_3,$

$b_6 = a_3^2 + 4a_6,$

$b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2.$

Bên cạnh đó, các đường cong sau là các đường cong hữu ích cho mật mã:

- Đường cong Edwards: $x^2 + y^2 = 1 + dx^2y^2$.
- Đường cong xoắn: $–x^2 + y^2 = 1 + dx^2y^2$.
- Đường cong Montgomery: $y^2 = x^3 + ax^2 + x$.

Các ứng dụng trong thực tế thường cài đặt ECC với hai đường cong elliptic "**curve25519**" và "**curve448**"[1] xác định trên các trường nguyên tố. Những đường cong này nhằm mục đích xây dựng các hệ mã hoạt động ở mức bảo mật lần lượt mức 128-bit và 224-bit thỏa các yêu cầu bảo mật chuẩn.

Curve25519 và **curve448** có khả năng chống lại các tấn công kênh phụ (*side channel attacks*), bao gồm tấn công khai thác thời gian (*timing*) và bộ nhớ đệm (*cache*). **Curve25519** và **curve448** là các đường cong Montgomery và do đó có phiên bản Edwards tương đương. Đường cong Edwards hỗ trợ cài đặt hiệu quả các phép tác động nhóm đường cong elliptic. Cụ thể, đường cong Edwards ứng với số nguyên tố p thỏa $p \equiv 3 \pmod 4$, và đường cong Edwards xoắn ứng với số nguyên tố p, $p \equiv 1 \pmod 4$.

Curve25519

Cho mức bảo mật 128-bit, số nguyên tố $p = 2^{255} - 19$ nên được chọn vì cho hiệu suất cao trên nhiều kiến trúc. Số nguyên tố p này thỏa $p \equiv 1 \pmod 4$.

Curve25519 và giao thức thiết lập khóa – **key exchange protocol**

Curve25519 sử dụng **hàm X25519**[1] cho giao thức Elliptic Curve Diffie-Hellman (**ECDH**):

. Alice sinh ngẫu nhiên 32 byte a[0] … a[31] và truyền K_A = X25519(a, 9) tới Bob, trong đó 9 là tọa độ u của điểm sinh và được mã hóa dưới dạng byte có giá trị 9, sau đó là 31 byte 0.

. Tương tự, Bob sinh ngẫu nhiên 32 byte b[0] … b[31], tính K_B = X25519(b, 9) và truyền cho Alice.

. Alice tính X25519(a, K_B) và Bob tính X25519(b, K_A).

Cả Allice và Bob hiện có chung K = X25519(a, X25519(b, 9)) = X25519(b, X25519(a, 9)) như một bí mật được chia sẻ.

Curve448

Cho với mức bảo mật 224-bit, số nguyên tố $p = 2^{448} - 2^{224} - 1$ được đề xuất để thực hiện trên nhiều kiến trúc. Số nguyên tố p có tính chất $p \equiv 3 \pmod 4$.

[1] https://datatracker.ietf.org/doc/html/rfc7748

Curve448 sử dụng **hàm X448**[2] trong giao thức ECDH tương tự hàm X25519 của curve25519. Điểm khác biệt duy nhất là Alice và Bob tạo 56 bytes ngẫu nhiên (thay vì 32) và tính K_A = X448(a, 5) hoặc K_B = X448(b, 5), trong đó 5 là tọa độ u của điểm sinh và được mã hóa dưới dạng byte có giá trị 5, tất cả 55 byte theo sau là 0.

MÃ ĐỒNG CẤU

Khái niệm đồng cấu

Một hàm *f* xác định trên D được gọi là đồng cấu (*homomorpic*) theo phép toán $\oplus$ nếu

$f(x \oplus y) = f(x) \oplus f(y)$.

Khái niệm mã đồng cấu (*homomorphic encryption*) cũng được định nghĩa tương tự.

Hệ mã $E_{k \in K}: \mathcal{M} \to C$ được gọi là mã đồng cấu theo phép toán $\otimes$ nếu $E_k(m_1) \otimes E_k(m_2) = E_k(m_1 \otimes m_2)$.

Mã đồng cấu được áp dụng cho những ứng dụng ở đó cần phải thực hiện một số tính toán trên bản rõ mà không phải giải mã các bản mã nhận được. Chẳng hạn các hệ hợp tác tính toán đảm bảo tính bí mật thông tin, hay các hệ xác minh mà không phải cung cấp thông tin bản rõ có thể sử dụng mã đồng cấu để hiện thực.

Các hệ mã đồng cấu phổ biến

Hệ mã đồng cấu RSA

RSA là hệ mã đồng cấu theo phép nhân modulo.

Thực vậy, giả sử $c_1 = RSA_{e,n}(m_1)$ và $c_2 = RSA_{e,n}(m_2)$ lần lượt là bản mã RSA của các thông điệp rõ m_1 và m_2 với khóa công khai e, n. Ta có

$c_1 \equiv m_1^e \ (mod\ n); c_2 \equiv m_2^e \ (mod\ n)$, thì

$c_1 * c_2 \ \equiv m_1^e * m_2^e \ \equiv (m_1 * m_2)^e \ (mod\ n)$,

Hay $c_1 * c_2$ chính là bản mã của thông điệp $m = m_1 * m_2$.

Hệ mã đồng cấu ElGamal

ElGamal là hệ đồng cấu theo phép nhân modulo.

Thực vậy, giả sử $C_1 = ElGamal_{e,g,p}(m_1)$ và $C_2 = ElGamal_{e,g,p}(m_2)$ lần lượt là bản mã ElGamal của các thông điệp rõ m_1 và m_2 với khóa công khai e, g, p. Ta có

$C_1 = (g^{x_1} \ mod\ p, m * e^{x_1} \ mod\ p); C_2 = (g^{x_2} \ mod\ p, m_2 * e^{x_2} \ mod\ p)$.

Đặt

$C = (g^{x_1 + x_2} \ mod\ p, (m_1 * m_2) * e^{x_1 + x_2} \ mod\ p)$.

[2] https://datatracker.ietf.org/doc/html/rfc7748

Thì C chính là bản mã ElGamal của $m = m_1 * m_2$.

Một cách tổng quát,

ElGamal-like là mã đồng cấu.

Hệ mã đồng cấu Paillier

Bên cạnh 2 hệ mã công khai RSA và ElGamal-like, ta học them hệ mã công khai, cũng có tính đồng cấu, cũng được sử dụng phổ biến trong bảo mật thông tin, mã Paillier (*Paillier encryption*).

Hệ mã Paillier, được phát minh và đặt theo tên của Pascal Paillier vào năm 1999, là một thuật toán mã bất đối xứng xác suất mà độ bảo mật dựa trên bài toán tính lượng thứ n được cho là bài toán khó. Giả định về độ dư tổng hợp quyết định là giả thuyết về tính khó sửa mà hệ thống mật mã này dựa vào.

Lược đồ mã hóa Paillier được mô tả như sau.

Sinh khóa – **KeyGen**

(1) Chọn ngẫu nhiên 2 số nguyên tố lớn phân biệt p và q thỏa

gcd(pq, (p – 1)(q – 1)) = 1.

(2) Tính n = pq và λ = lcm(p – 1, q – 1), với lcm (*least common multiple*) là hàm tính bội số chung nhỏ nhất của 2 số nguyên dương.

(3) Chọn ngẫu nhiên số $g \in \mathbb{Z}^*_{n^2}$, với $\mathbb{Z}^*_{n^2}$ là tập các số khả nghịch (mod n^2).

(4) Kiểm tra đẳng thức $\mu = L\left(g^{\lambda} \bmod n^2\right)^{-1}$ với L(x) ký hiệu số nguyên không âm lớn nhất sao cho (x – 1) ≥ vn.

Khóa công khai sẽ là e = (n, g), và kháo cá nhân là (λ, μ).

Lưu ý: nếu p và q có chiều dài bit như nhau, đơn giản chỉ cần chọn g = n + 1, λ = ϕ(n) = (p – 1)(q – 1), và $\mu \equiv \phi(n)^{-1} \pmod p$.

Mã hóa – **Encryption**

Để mã hóa thông điệp m, 0 ≤ m < n.

(1) Chọn ngâu nhiên số nguyên r, 0 < r < n nguyên tố cùng nhau với n, gcd(n, r) = 1.

(2) Tính $c = g^m * r^n \bmod n^2$.

c chính là bản mã của m: $c = \text{Paillier}_{n,g}(m)$.

Giải mã – **Decryption**

Để giải mã bản mã $c \in \mathbb{Z}^*_{n^2}$.

(1) Tính $m = L\left(c^{\lambda} \bmod n^2\right) * \mu \bmod n$.

Thì m chính là thông điệp rõ của bản mã c.

Các tính chất đồng cấu của Paillier – **homomorphic properties**

Mã Paillier là mã đồng cấu với nhiều tính đồng cấu.

Thực vậy, ký hiệu D là hàm giải mã và E là hàm mã hóa.

Đồng cấu theo phép cộng

- Tích của 2 bản mã được giải mã thành tổng của 2 thông điệp.

$D(E(m_1, r_1) * E(m_2, r_2)) \bmod n^2) = m_1 + m_2 \bmod n.$

- Tích của bản mã của thông điệp này với lũy thừa theo g của thông điệp khác được giải mã thành tổng của 2 thông điệp.

$D(E(m_1, r_1) * g^{m_2} \bmod n^2) = m_1 + m_2 \bmod n.$

Đồng cấu theo phép nhân

- Lũy thừa bản mã của thông điệp này với lũy thừa theo g của thông điệp khác được giải mã thành tích của 2 thông điệp.

$D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 * m_2 \bmod n,$

$D(E(m_2, r_2)^{m_1} \bmod n^2) = m_1 * m_2 \bmod n.$

Tổng quát hơn, một bản mã được nâng lũy thừa k sẽ giải mã thành tích của thông điệp và hằng số,

$D(E(m_1, r)^k \bmod n^2) = k * m \bmod n.$

Tuy nhiên, với mã hóa Paillier của hai thông điệp, không có cách nào để tính bản mã của tích 2 thông điệp này mà không biết khóa cá nhân.