فصل –5 نظرية الاعداد والزمر

5.1 مقدمة

تعتمد مناهج التشفير معلنة المفتاح على وجود مسائل رياضية يفترض ان تكون صعبة. هذه المسائل هي مسائل عددية في طبيعتها. لذا يحسن ان نتعرف على الاساس الرياضي لتلك المسائل ، ومن ثم نعرض بعض الافتراضات الرياضية الصعبة. ختاما سنقدم بعض خوارزميات اثبات اولية الاعداد ، تحليل الاعداد لعواملها الاولية ، وخوارزميات ايجاد اللوغاريتم المتقطع.

5.2 نظرية الاعداد

5.2.1 الاعداد الاولية والقسمة

- يشير الرمز $\mathbb{Z}$ لمجموعة الاعداد الصحيحة.
- بالنسبة للعددين $a, b \in \mathbb{Z}$ فإن a يقسّم b ويكتب بالصورة $a|b$ اذا وجد عدد صحيح c بحيث ان $ac = b$ وفي هذه الحالة يعرف a بكونه "مقسّم" b او بكونه احد "عوامل" b.
- نستخدم الرمز $a \nmid b$ للاشارة الى ان a لا يقسّم b.

مثال(5.1): عندما $b = 24$ فإن احد عوامله $a = 4$ وذلك لان 24=8*4.

- يعرف العدد الصحيح $p > 1$ بانه عدد اولي prime اذا لايوجد اي عدد يقسّمه سوى 1 وp نفسه.
- جميع الاعداد الصحيحة التي هي اكبر من 1 والتي ليست اعداد اوليه تدعى اعداد مركبة composite.

تنص نظرية الاعداد الاساسية على ان اي عدد صحيح $N > 1$ يمكن ان يكتب بصورة فريدة بالشكل $N = \prod_i p_i^{e_i}$ ، بحيث $\{p_i\}$هي اعداد اولية منفردة و$e_i \geq 1$

لجميع قيم i.

5.2.2 باقي القسمة

عندما يكون لدينا العدد صحيح a والعدد الصحيح الموجب b ، فانه يوجد اعداد صحيحة فريدة q و r بحيث $a = qb + r$ و$0 \leq r < b$. يعرف العدد r بكونه باقي القسمة ، والعدد qيعرف بكونه ناتج القسمة.

مثال(5.2): عندما $b = 15$ و$a = 4$ فان $q = 3$و $r = 3$.

تستخدم عملية mod لحساب باقي القسمة ويمكن صياغتها بالشكل:

$$r = b \bmod a.$$

مثال(5.3): لحساب $101 \bmod 7$ ، نكتب $101 = 7 \times 14 + 3$ ، وبالتالي: $101 \bmod 7 = 3$.

اما لحساب $-101 \bmod 7$ ، فنكتب $-101 = 7 \times (-15) + 4$ ، وبالتالي $-101 \bmod 7 = 4$.

5.2.3 القاسم المشترك الاعظم

- عندما يكون لدينا عددين صحيحين غير سالبين a و b فإن **القاسم المشترك الاعظم** لهما والذي يعرف بـ $\gcd(a,b)$ هو اكبرعدد صحيح c بحيث $c|a$ و $c|b$.
- عند احتساب $\gcd(a,b)$ فاننا نفترض عادة ان $a,b \geq 0$. لاحظ ، ان $\gcd(a,0) = \gcd(0,a) = a$. عندما يكون $\gcd(a,b) = 1$ فاننا نقول ان a و b هما "اوليّان نسبيا" relative prime.

خوارزمية اقليدس.تستخدم خوارزمية اقليدس Euclidean algorithm لحساب القاسم المشترك الاعظم بين عددين موجبين a و b. تعتمد هذه الخوارزمية على مبدأ $\gcd(a,b) = \gcd(b, a \bmod b)$. توضّح خوارزمية (5.1) طريقة عمل هذه الخوارزمية.

ليس من الصعب ملاحظة ان

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots = \gcd(r_{m-1}, r_m) = r_m ،$$

Algorithm (5.1): Euclidean algorithm

Input: $a \geq b > 1$

Output: gcd(a,b)

$r_0 = a$
$r_1 = b$
$m = 1$
while $r_m \neq 0$

- $q_m = \lfloor r_{m-1}/r_m \rfloor$
- $r_{m+1} = r_{m-1} - qr_m$
- $m = m + 1$

$m = m - 1$
return r_m

وبذلك فإن $\gcd(r_0, r_1) = r_m$.

مثال (5.4): لحساب القاسم المشترك الاعظم بين العددين 1160718171 و316258250 . نقسّم العددين لحين الحصول على باقي قسمة صفر ،

المقسوم	المقسوم عليه	الناتج	باقي القسمة
1160718174	316258250	3	211943424
316258250	211943424	1	104314826
211943424	104314826	2	3313772
104314826	3313772	31	1587894
3313772	1587894	2	137984
1587894	137984	11	70070
137984	70070	1	67914
70070	67914	1	2156
67914	2156	31	**1078**
2156	1078	2	0

عندها يكون باقي القسم قبل الاخير هو الحل: 1078.

خوارزمية اقليدس الموسعة. ليكن لدينا عددين صحيحين موجبين a و b فانه يوجد عددين صحيحن X و Y بحيث يكون $Xa + Yb = \gcd(a, b)$. للحصول على قيم X و Y بالاضافة الى $\gcd(a, b)$ نستخدم خوارزمية اقليدس الموسّعة extended Euclidean algorithm. خوارزمية (5.2) توضّح كيفية عمل هذه الخوارزمية.

مثال(5.5): عند حساب $Xa + Yb = \gcd(1759,550)$، فاننا نقسّم العددين لحين الحصول على باقي قسمة صفر، عندها نتوقف ونسترجع القيم قبل الاخيرة($\gcd(a, b) = r = 1 ،Y = 355 ،X = -111$).

Algorithm (5.2): Extended Euclidean algorithm eGCD

Input: Integers a, b with $a \geq b > 0$

Output: (d, X, Y) with $d = \gcd(a, b)$ and $Xa + Yb = d$

if (b|a)

  return (b, 0, 1)

else

  $r = a \bmod b$

  $(d, X, Y) = eGCD(b, r)$

  return $(d, Y, X - Yq)$

i	r_i	q_i	X_i	Y_i
-1	1757		1	0
0	550		0	1
1	109	3	1	-3
2	5	5	-5	16
3	4	21	106	-339
4	**1**	1	**-111**	**355**
5	0	4		

5.2.4 العمليات الرياضية Modular

- ذكرنا ان التعبير $a \bmod N$ يعني باقي قسمة a على N.
- يكون العددان a و b متطابقان congruent بدلالة N ويكتبان بالصيغة $a = b \bmod N$، اذا كان $a \bmod N = b \bmod N$ بمعنى ان باقي قسمة a عند تقسيمه على N هو نفسه عند تقسيم b على N.
- تدعى قيمة N بالـ modulus.

مثال(5.6): $36 = 21 \bmod 15$، وذلك لان باقي قسمة 36 على 15 هو نفسه باقي قسمة 21 على 15 وهو 6.

- يفيد مفهوم التطابق هذا في اختزال العمليات الرياضية. على سبيل المثال لايجاد $[1093028.19301 \bmod 100]$ نقوم اولا باختزال $[1093028 \bmod 100] = 28$ ، واختزال $[190301 \bmod 100] = 1$ ومن ثم حساب $[28.1 \bmod 100] = 28$.
- من المثال اعلاه يتضح كيفية استخدام العملية mod لاختزال وتبسيط عملية الجمع ويمكن تطبيق نفس الفكرة على اختزال عمليات الطرح والضرب كما موضح في القوانين ادناه:

1. $[a + b] \bmod N = [a \bmod N] + [b \bmod N] \bmod N$

2. $[a - b] \bmod N = [a \bmod N] - [b \bmod N] \bmod N$
3. $[a.b] \bmod N = [a \bmod N].[b \bmod N] \bmod N$

5.2.5 المعكوس الضربي

لاتخضع عملية القسمة لطريقة الاختزال التي ذكرناه اعلاه ، بمعنى ادق

$$[a/b] \bmod N \neq [a \bmod N]/[b \bmod N] \bmod N$$

يمكن ان نجد تعريف مناسب لعملية القسمة يتلائم مع رياضيات modular . اذا كان للعدد الصحيح b عدد صحيح b^{-1} بحيث يكون $bb^{-1} = 1 \bmod N$ فأننا نعرّف b^{-1} بكونه المعكوس الضربي multiplicative inverse للعدد $b \bmod N$ ، ويعرف b بانه قابل للعكس $\bmod N$. في ضوء ذلك ، نعرّف عملية القسمة بالشكل:

$$a/b = ab^{-1} \bmod N ،$$

وتكون ممكنة فقط عندما يكون b قابلا للعكس في حالة $\bmod N$.

قد يسأل البعض ما هي الاعداد الصحيحة القابلة للعكس في حالة $\bmod N$؟ والجواب يظهر من الفرضية التالية.

فرضية (5.1)**.** ليكن لدينا عددين صحيحين aو N ، حيث $N > 1$. يكون a قابلا للعكس$\bmod N$ فقط واذا فقط اذا كان $\gcd(a, N) = 1$.

ويمكن برهنة هذه الفرضية بافتراض ان a هو عدد موجب. بما ان $ab = 1 \bmod N$ هذا يعني ان $ab - 1 = cN$ لقيمة $c \in \mathbb{Z}$ ، وهذا يعني ان $\gcd(a, N) = 1$ (حسب خوارزمية اقليدس الموسعة).

مثال(5.7**):** ليكن $a = 11$ و$N = 17$. فإن $(-3).11 + 2.17 = 1$ ، وبذلك يكون $[-3 \bmod 17] = 14$ هي معكوس 11 ، يمكن اثبات ذلك $14.11 = 1 \bmod 17$.

خوارزمية حساب المعكوس الضربي.

- ليكن لدينا N و $a \in \{0, \dots N - 1\}$ عندما يكون $\gcd(a, N) = 1$ فإن خوارزمية اقليدس الموسعة ترجع لنا $Xa + Yb = 1$. يكون $[X \bmod N]$ هو المعكوس الضربي للعدد a.

Algorithm (5.3): **Multiplicative inverse algorithm**
Input: N, a
Output: a^{-1}
$(d, X, Y) = eGCD(a, N)$
if $d \neq 1$ **return** "there is no inverse"
else return [X mod N]

- توضّح خوارزمية (5.3) كيفية عمل حساب المعكوس.

5.3 نظرية الزمر

يمثل تجمع العناصر بالعملية الثنائية ∘ مايعرف بالزمرة $\mathbb{G}$ group بحيث يتحقق:

- **(زمرة مغلقة)** لكل $g \circ h \in \mathbb{G}, h \circ g \in \mathbb{G}$
- **(وجود العنصر المحايد)** يوجد عنصر $e \in \mathbb{G}$ بحيث

لجميع $g \in \mathbb{G}, e \circ g = g = g \circ e$.

- **(وجود المعكوس)** لكل $g \in \mathbb{G}$ يوجد عنصر $h \in \mathbb{G}$ بحيث

$g \circ h = e = h \circ g$. يعرف العنصر h بمعكوس g.

- **(خاصية التشارك)**

لكل $g_1, g_2, g_3 \in \mathbb{G}$ ، $(g_1 \circ g_2) \circ g_3 = g_1 (\circ g_2 \circ g_3)$.

- عندما تكون $\mathbb{G}$ منتهية العناصر ، فأنها تسمى الزمرة المنتهية ويمثل الرمز $|\mathbb{G}|$ **رتبة** order الزمرة وهو يمثل عدد عناصر الزمرة.
- تعرّف الزمرة $\mathbb{G}$ بكونها زمرة **تبادلية** abelian اذا تحقق الشرط: لكل $h, g \in \mathbb{G}$ يكون $g \circ h = h \circ g$.

مثال(5.9): الاعداد الصحيحة $\mathbb{Z}$ هي زمرة تبادلية بالنسبة لعملية الجمع وليست زمرة بالنسبة لعملية الضرب لان الرقم الصحيح 2 ، على سبيل المثال ، لايوجد له معكوس ضربي في مجموعة الاعداد الصحيحة.

مثال(5.10): عندما $N \geq 2$. فإن الزمرة $\{0, \dots, N-1\}$ هي زمرة تبادلية بالنسبة لعملية الجمع $\bmod N$ ورتبة تلك الزمرة هو N. لاحظ ، ان معكوس كل عنصر هو $[(N - a) \bmod N]$. تعرف هذه الزمرة بـ $\mathbb{Z}_N$ وتشمل العناصر $\{0, \dots, N-1\}$.

5.3.1 الرفع للاس في الزمر

عند تطبيق عملية الضرب على عنصر من عناصر الزمرة m من المرّات فانه يرمز لهذه العملية بالرمز:

$$g^m = \underbrace{g \dots g}_{m \text{ times}}$$

من ابرز قواعد عملية الرفع للاس هي:

$$g^1 = g \text{ ، } (g^m)^{m'} = g^{mm'} \text{ ، } g^m . g^{m'} = g^{m+m'}$$

خوارزمية كفوءة لعملية الرفع للاس. تعتبر عملية الرفع للأس $a^b \bmod N$ ، عندما $a \in \mathbb{Z}_N$ ، و $b > 0$ عملية معقدة. نستطيع ان نعالج هذه العملية وذلك باختزال عمليات الرفع للاس بصورة متكررة ، مما يسمح بأن تكون النتائج الوسطية صغيرة خلال عملية الاحتساب.

Algorithm (5.4): A standard approach for modular exponentiation

Input: $N, a \in \{0, \dots N-1\}, b > 0$

Output: $a^b \bmod N$

```
x = 1
for i = 1 to b:
    x = x. a mod N
return x
```

للأسف ، حتى مع هذا الاسلوب ، فأنه لازالت عملية الرفع للاس عملية غير كفوءة. خورزمية (5.4) توضح كيفية اجراء عملية الرفع للاس بصورة مختزلة.

بما ان الخوارزمية (5.4) تستخدم b من الدورات فانها تحتاج الى وقت تنفيذ خطيا مع $\|b\|$. لاحظ ، ان الخوارزمية (5.4) تتبع التكرار التالي:

$$a^b \bmod N = a.a^{b-1} \bmod N = a.a.a^{b-2} \bmod N = \cdots$$

يمكن ان نحسّن الاداء بصورة افضل عند اتباع التكرار التالي:

- عندما يكون b زوجي: $a^b \bmod N = \left(a^{b/2}\right)^2 \bmod N$
- عندما يكون b فردي: $a^b \bmod N = a.\left(a^{(b-1)/2}\right)^2 \bmod N$

تم صياغة هذا التكرار بشكل خوارزمية تعرف بـ(square-and-multiply) لحساب عملية الرفع للاس التي تتطلب فقط $\mathcal{O}(\log b)$، والتي

Algorithm (5.5): Square–and–multiply for efficient modular exponentiation

Input: N, $a \in \{0, \ldots N-1\}$, $b > 0$

Output: $a^b \bmod N$

$t = 1$
$\ell = \|b\|$
for $i = \ell - 1$ downto 0:
  $t = t^2 \, mod \, N$
  if $b_i = 1$
    $t = (t \times a) mod \, N$
return t

توضحها خوارزمية (5.5).

مثال(5.11): لحساب $9726^{3533} \bmod 11413$ بخوارزمية square-and-multiply نحصل على 5761.

i	b_i	t
11	1	$1^2 \times 9726 = 9726$
10	1	$9726^2 \times 9726 = 2659$
8	1	$5634^2 \times 9726 = 9167$
7	1	$9167^2 \times 9726 = 4958$
6	1	$4958^2 \times 9726 = 7783$
5	0	$7783^2 = 6289$
4	0	$6289^2 = 4629$
3	1	$4629^2 \times 9726 = 10185$
2	1	$10185^2 \times 9726 = 105$
1	0	$105^2 = 11025$
0	1	$11025^2 \times 9726 = 5761$

5.3.2 حقائق مهمة عن الزمر

نظرية (5.1). عندما تكون $\mathbb{G}$ زمرة منتهية العناصر بحيث $m = |\mathbb{G}|$ ، فإن لكل عنصر $g \in \mathbb{G}$ يكون $g^m = 1$.

ويمكن برهنة هذه النظرية بافتراض ان لدينا $g_1, \dots, g_m$ في الزمرة $\mathbb{G}$. الأن ندّعي ان

$$g_1 \cdot g_2 \cdots g_m = (gg_1).(gg_2) \cdots (gg_m)$$

وذلك لكون $gg_i \in \mathbb{G}$ باعتبار ان $\mathbb{G}$ هي زمرة مغلقة. وبالتالي يكون لدينا

$$g_1 \cdot g_2 \cdots g_m = g^m(g_1 \cdot g_2 \cdots g_m)$$

مما يستدعي ان $g^m = 1$.

متلازمة (5.1)، عندما تكون $\mathbb{G}$ زمرة منتهية العناصر بحيث $m = |\mathbb{G}| > 1$ ، فإن لكل عنصر $g \in \mathbb{G}$ ولأي عدد صحيح i يكون $g^i = g^{i \bmod m}$.

ويمكن برهنة هذه المتلازمة بالشكل التالي: ليكن $i = qm + r$، بحيث $r = i \bmod m$. عندها يكون

$$g^i = g^{qm+r} = g^{qm}.g^r = (g^m)^q.g^r = 1.g^r = g^r$$

متلازمة (5.2). لتكن $\mathbb{G}$ زمرة منتهية العناصر بحيث $m = |\mathbb{G}| > 1$. ليكن $e > 0$ هو عدد صحيح ، ولتكن $f_e: \mathbb{G} \to \mathbb{G}$ بالصورة $f_e(g) = g^e$. اذا كان $\gcd(e, m) = 1$ فإن f_e هي بعثرة permutation. علاوة على ذلك ، اذا كان $d = [e^{-1} \bmod m]$ ، فإن f_d هي معكوس f_e.

يمكن برهنة هذه المتلازمة اعتمادا على فرضية (5.1) التي تنص على وجود معكوس ضربي للعددين اذا كانا اوّليان نسبيا ، اما كون f_d معكوسا للدالة f_e فيظهر من:

$$f_d\big(f_e(x)\big) = f_d(x^e) = (x^e)^d = (x)^{ed} = (x)^{[ed \bmod N]} = (x)^1 = x$$

زمرة $\mathbb{Z}_N^*$. تعرف زمرة $\mathbb{Z}_N^*$ بانها مجموعة جميع الاعداد الصحيحة في المجموعة $\{1, \dots, N-1\}$ التي تكون اولية نسبيا مع N. وتعرف بالشكل:

$$\mathbb{Z}_N^* = \{a \in \{1, \dots N-1\} | \gcd(a, N) = 1\}$$

تمثل $\mathbb{Z}_N^*$ زمرة تبادلية فيما يخص عملية الضرب $\bmod N$.

دالة Euler phi. يرمز لعدد عناصر $\mathbb{Z}_N^*$ بالرمز $\emptyset(N) = \left|\mathbb{Z}_N^*\right|$ وتعرف بدالة Euler phi.

- عندما تكون $N = p$ هي عدد اولي. عندها يكون $\emptyset(p) = \left|\mathbb{Z}_p^*\right| = p - 1$
- عندما تكون $N = p.q$ هي حاصل ضرب عددين اوليين مختلفين. عندها يكون $\emptyset(N) = (p-1)(q-1)$. وهي العناصر التي لاتقبل القسمة على p او q وبالتي هو عدد العناصر الاولية نسبيا مع N.

مثال(5.12): لتكن $N = 15 = 5.3$، فإن $\mathbb{Z}_{15}^{*} = \{1,2,4,7,8,11,13,14\}$ ، $\left|\mathbb{Z}_{15}^{*}\right| = 8 = 4.2 =$ $\emptyset(15)$.

متلازمة (5.3). ليكن $N > 1$، وليكن $a \in \mathbb{Z}_N^*$، فإن $a^{\emptyset(N)} = 1 \bmod N$.

نظرية Fermat (5.2). عندما يكون $N = p$ عددا اوليا، و $a \in \{1, \dots, p-1\}$ فإن: $a^{p-1} = 1 \bmod p$،

- هناك صيغة اخرى هي $a^p = a \bmod p$

مثال(5.13): عندما $a = 7, p = 19$، فإن $a^{p-1} = 7^{16} = 1 \bmod 17$

متلازمة (5.4). ليكن $e > 0$ هو عدد صحيح، $N > 1$ ، ولتكن $\mathbb{Z}_N^*$ بالصورة $f_e: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ $f_e(x) = x^e$. اذا كان $\gcd\left(e, \emptyset(N)\right) = 1$ فإن f_e هي **بعثرة** permutation. علاوة على ذلك، اذا كان $d = [e^{-1} \bmod \emptyset(N)]$، فإن f_d هي معكوس f_e.

5.4 الافتراضات الصعبة

ذكرنا فيما سبق ان أمنية مناهج التشفير الحديثة تعتمد على وجود مسائل يفترض ان تكون صعبة، بحيث اذا تم حل هذه المسائل فأنه يشكل تهديد لأمنية تلك المناهج. في هذا الجزء نذكر بعض تلك المسائل.

5.4.1 افتراض تحليل العوامل

تعتبر مسأل ايجاد عوامل العدد من مسائل نظرية الاعداد الصعبة. عند توفر العدد المركّب الصحيح N، تعمل مسألة تحليل العوامل على ايجاد العددين الصحيحين الموجبين p وq بحيث $N = pq$. يمكن صياغة هذه المسالة شكليا كما يلي: ليكن $\mathrm{GenModulus}$ خوارزمية تعمل عند ادخال معامل الأمنية n على اخراج (N, p, q) بحيث $N = pq$ وp وq اعداد اولية بطول n. افترض التجربة التالية للخصم $\mathcal{A}$ ولمعامل الأمنية n.

تجربة تحليل العوامل $\mathbf{Factor}_{\mathcal{A},\mathrm{GenModulus}}(n)$

1. تنفيذ $\mathrm{GenModulus}(n)$ للحصول على (N, p, q).
2. يعطى $\mathcal{A}$ العدد N، ويقوم باخراج $p', q' > 1$.
3. يكون اخراج التجربة 1 اذا كان $p'.q' = N$، و0 ماعدا ذلك.

نقول ان Factoring هي مسالة صعبة نسبة الى $\mathrm{GenModulus}$ اذا كان لجميع الخصوم $\mathcal{A}$ توجد دالة ضئيلة جدا negl بحيث:

$$\Pr[\mathrm{Factor}_{\mathcal{A},\mathrm{GenModulus}}(n) = 1] \leq \mathrm{negl}(n)$$

5.4.2 افتراض RSA

على الرغم من صعوبة مسألة تحليل العوامل السابقة الّا انها لا تقدم شيئًا مفيدا في تصميم مناهج التشفير. لذا تم البحث عن مسائل اخرى مرتبطة بمسألة تحليل العوامل. افضل مسألة هي تلك التي قدمها كل من Rivest، Shamir، و Adleman والمعروفة بمسألة RSA.

- كما ذكرنا فإن رتبة الزمرة $\mathbb{Z}_N^*$ هي $\emptyset(N) = (p-1)(q-1)$.
- عندما يكون N قابلا للتحليل فانه من السهل حساب $\emptyset(N)$. اما عندما يكون تحليل عوامل N غير معلوم فانه من الصعب حساب $\emptyset(N)$.
- استثمرت مسألة RSA هذه الميزة ، بحيث تكون مسألة RSA سهلة عندما يكون $\emptyset(N)$ معلوما وتكون صعبة بدون معرفة $\emptyset(N)$.
- تتلخص مسألة RSA بما يلي: ليكن لدينا العدد الصحيح المركب N وليكن $e > 0$ عدد اولي نسبيا مع $\emptyset(N)$، ليكن العدد $y \in \mathbb{Z}_N^*$ ، احسب $y^{1/e} \bmod N$، بحيث عند اعطاء (N, e, y) المطلوب ايجاد قيمة x بحيث $x^e = y \bmod N$.

يمكن صياغة المسألة شكليا كما يلي: ليكن GenRSA خوارزمية تعمل عند ادخال معامل الأمنية n على اخراج $\left(N, e > 0 \mid \gcd(e, \emptyset(N) = 1, d | ed = 1 \bmod \emptyset(N)\right)$. افترض التجربة التالية للخصم $\mathcal{A}$ ولمعامل الأمنية n.

تجربة RSA $\mathbf{RSA - inv_{\mathcal{A},GenRSA}(n)}$

1. تنفيذ $\text{GenRSA}(n)$ للحصول على (N, p, q).
2. اختر $y \leftarrow \mathbb{Z}_N^*$.
3. يعطى $\mathcal{A}$ (N, e, y) ، ويقوم باخراج $x \in \mathbb{Z}_N^*$.
4. يكون اخراج التجربة 1 اذا كان $x^e = y \bmod N$ ، و0 ماعدا ذلك.

نقول ان RSA هي مسالة صعبة نسبة الى GenRSA اذا كان لجميع الخصوم $\mathcal{A}$ توجد دالة ضئيلة جدا negl بحيث:

$$\Pr[\text{RSA} - \text{inv}_{\mathcal{A},\text{GenRSA}}(n) = 1] \leq \text{negl}(n)$$

Algorithm (5.6): GenRSA

Input: security parameter, n

Output: (N, e, d)

$(N, p, q) = \text{GenModulus}(n)$

$\emptyset(N) = (p-1)(q-1)$

find e such that $\gcd(e, \emptyset(N) = 1$

compute: $d = [e^{-1} \bmod \emptyset(N)]$

return (N, e, d)

- يمكن تصميم خوارزمية GenRSA اعتمادا على خوارزمية GenModulus التي تخرج قيمة N مع عواملها. ويمكن صياغة GenRSA في خوارزمية (5.6).

لاحظ عندما يكون تحليل N معلوما ، تكون مسألة RSA سهلة الحل: نقوم اولا بحساب $\emptyset(N)$ ، بعدها نحسب

$d = [e^{-1} \bmod \emptyset(N)]$، في النهاية نحسب الحل $[y^d \bmod N]$. وبالتالي ، لكي تكون مسألة RSA صعبة الحل ، يجب ان يكون من المتعذر تحليل قيمة N. بمعنى ان مسألة RSA لا يمكن ان تكون اكثر صعوبة من التحليل. ولكن ماذا عن العكس ؟ هل من الممكن تحليل عددين بسهولة اذا امكن حل مسألة RSA. الجواب هو كلا ، اذ لا يمكن استنتاج ان RSA هي صعبة بصعوبة التحليل.

5.5 افتراضات الزمر

المجاميع الجزئية. ليكن لدينا الزمرة المنتهية G والتي رتبتها m. لكل عنصر $g \in G$ نعرّف:

$$\langle g \rangle = \{g^0, g^1, \dots, g^{i-1}\}$$

بحيث $i \leq m$ ، و i هو اصغر عدد صحيح موجب يحقق $g^i = 1$. تدعى $\langle g \rangle$ **زمرة جزئية** متولدة من قبل g.

تعريف (5.2): ليكن لدينا الزمرة المنتهية $\mathbb{G}$ و $g \in \mathbb{G}$. تعرف "**رتبة العنصر**" $\text{ord}(g)$ بكونها اصغر عدد صحيح موجب i بحيث $g^i = 1$.

تعريف(5.3): ليكن mهو رتبة الزمرة $\mathbb{G}$ ($m = |\mathbb{G}|$)، اذا وجد $g \in \mathbb{G}$ بحيث $\text{ord}(g) = m$ ، فإن هذا يعني ان $\langle g \rangle = G$.
في هذه الحالة تسمى $\mathbb{G}$ **زمرة دورية** cyclic group ويعرف g بكونه "مولّد" generator للزمرة $\mathbb{G}$ ، او يعرف بال"**عنصر البدائي**" primitive element.

لاحظ ، ان العنصر g يسمى مولّد $\bmod p$ اذا وفقط اذا

$$\{g^i : 0 \leq i \leq p-2\} = \mathbb{Z}_p^*$$

بمعنى انه يولّد جميع عناصر الزمرة عند رفعه لجميع الاسس.

مثال(5.16): ليكن $p = 13$ ، نرى ان العدد $g = 2$ هو مولّد للزمرة $\mathbb{Z}_{13}^*$ ، بحيث يولّد جميع عناصرها عند رفعه للقيم 0...11.

i	0	1	2	3	4	5	6	7	8	9	10	11
$2^i \bmod 13$	1	2	4	8	3	6	12	11	9	5	10	7

Algorithm (5.7): Generator testing
Input: Group order q; prime factors $\{p_i\}_{i=1}^k$ of q;
element $g \in G$ to be tested
Output: Decision whether g is generator or not
for $i = 1$ to k:
  if $g^{q/p_i} = 1$ **return** false

تعتبر خوارزمية (5.7) طريقة كفوءة لفحص فيما اذا كان العنصر g هو مولّد للزمرة ام لا.

متلازمة (5.4): عندما تكون $\mathbb{G}$ زمرة ذات رتبة اولية p، فإن $\mathbb{G}$ تكون زمرة دورية. كذلك، جميع عناصرها باستثناء العنصر المحايد هي مولدّات للزمرة $\mathbb{G}$.

5.5.1 افتراض DDH

- ليكن $\mathbb{G}$ زمرة دورية رتبتها m، فانه يوجد مولّد $g \in \mathbb{G}$ بحيث $\mathbb{G} = \{g^0, g^1, \dots, g^{m-1}\}$. بصورة اخرى، يمكن القول انه لكل $h \in \mathbb{G}$ يوجد عنصر وحيد $x \in \mathbb{Z}_q$ بحيث $g^x = h$.
- يدعى x **اللوغاريتم المتقطع** لـ h بالنسبة الى g.
- تتمحور **مسالة اللوغاريتم المتقطع** discrete logarithm problem بحساب قيمة x عند اعطاء g وh.
- هناك مسائل اخرى مرتبطة بمسألة اللوغاريتم المتقطع، مثل مسألة Decisional Diffie-Hellman (DDH) ومسالة Computational Diffie-Hellman (CDH). تتلخص مسألة CDH كالاتي: ليكن $\mathbb{G}$ زمرة دورية وليكن $g \in \mathbb{G}$ مولّد تلك الزمرة. عند اعطاء عنصرين من الزمرة $h_1 = g^{x1}$ و$h_2 = g^{x2}$. فأنه من الصعب جدا حساب القيمة:

$$DH_g(h_1, h_2) = g^{x1.x2} = {h_1}^{x2} = {h_2}^{x1}$$

- اما مسألة DDH فتنص على انه من الصعب على اي خصم ان يميّز $DH_g(h_1, h_2)$ عن قيمة عشوائية من الزمرة $\mathbb{G}$.

5.6 اختبار الاولية

- تعمل العديد من طرق التشفير معلنة المفتاح على اختيار اعداد اولية، مما يتطلب توليد اعداد صحيحة كبيرة ومن ثم فحص اولية تلك الاعداد.
- قام الباحثون بتطوير خوارزميات محددّة deterministic تعمل على فحص اولية الاعداد, وتعطي نتائج صحيحة دائما، مما يعد تطورا كبيرا في هذا المجال.
- من الناحية العملية، لازال استخدام الخوارزميات الاحتمالية probabilistic مفضلا في فحص اولية الاعداد.
- تمتاز الخوارزميات الاحتمالية بكونها سريعة وتعتمد على مبدأ العشوائية، حيث تفحص العدد الصحيح الذي عدد ثنائياته n بوقت $\log_2 n$،
- ولكن هناك احتمالية للفحص الخاطئ. ونقصد بذلك تحديدا، ان تلك الخوارزميات اذا اعلنت ان العدد المدخل "مركّب" فهو فعلا مركب، واذا اعلنت انه "عدد اولي" فهناك احتمالية ضئيلة جدا انه مركّب.
- يمكن التأكد من صحة الفحص وذلك بتكرار عمل الخوارزمية عدد من المرات لحين تقليل احتمالية الخطأ اقل من عتبة معينة.

استخدام نظرية Fermat. قد يفكّر البعض باستخدام نظرية Fermat (5.2) لفحص اولية العدد وذلك باختبار $a^N \bmod N = a$، حيث N هو العدد المطلوب اختبار اوليته.

- لكن هذا التفكير خاطئ لان نظرية Fermat تعمل بأتجاه واحد فقط، بمعنى انه اذا كان العدد اولي فإن النظرية تكون متحققة، وان كانت النظرية متحققة فلا يعني تحققها بأن العدد اولي. وللتوضيح نلاحظ ان العدد 341 يحقق صحة النظرية $2^{341} \bmod 341 = 2$ مع العلم ان هذ العدد مركّب 341=11.31.
- على الرغم من ذلك فانه يمكن استخدام نظرية Fermat للتحقق من اولية العدد بشكل "غير مؤكد": نقوم بأختيار عدد من الاعداد الصحيحة $a_1, a_2, a_3, \dots$ واختبار ${a_i}^N \bmod N = a_i$ لجميع قيم i. اذا تحقق الاختبار لجميع قيم a_i، فإن هذا يعطي "احتمال كبير وليس مؤكد" بأن N هو عدد اولي، وان لم يتحقق هذا الاختبار لحالة واحدة a_i على الاقل فعندها يكون العدد N هو عدد مركّب.

- خوارزمية (5.8) توضح كيفية استخدام نظرية Fermat لاختبار اولية الاعداد.

تعرف الاعداد a_i التي لايتحقق بها الاختبار $a_i^N \bmod N = a_i$ بكونها **شواهد** على التركيب witness of composite.

- هناك اعداد **شاذة** عن هذه الطريقة التكرارية ، والتي تعرف باعداد Carmichael numbers ، حيث هي اعداد مركبّة ولايوجد شاهد على تركيبها.

Algorithm (5.8): **Fermat primality test**
Input: N: number to be tested, t: test parameter
Output: Decision: N is prime or composite
for $i = 1$ to t:
  $a \leftarrow \{2, \dots, N-2\}$
  if $a^{N-1} \bmod N \neq 1$ then **return** "Composite"
return "may be Prime"

- مثال على ذلك العدد 561=3.11.17 يحقق الاختبار $a^{561} \bmod 561 = a$ لجميع قيم a.
- وقد ثبت ان هذه الاعداد الشاذة هي غير متناهية. لذا نحتاج الى طريقة اقوى من نظرية Fermat لفحص اولية العدد (بصورة احتمالية ايضا).

خوارزمية Miller-Rabin. في هذا الفصل سوف نستعرض خوارزمية Miller-Rabin التي تعد احد اشهر خوارزميات فحص الاولية الاحتمالية.

- تعتمد خوارزمية Miller-Rabin على مبدأ التعبير عن اي عدد فردي موجب $N \geq 3$ بالشكل $N - 1 = 2^k q$ حيث q عدد فردي و $k > 0$. باعتبار ان $N - 1$ هو عدد زوجي وان تقسيم هذا العدد k من المرات سوف يوصلنا الى عدد فردي.
- على سبيل المثال ، عند تقسيم 24 ثلاث مرات نصل للعدد الفردي 3. تعمل هذه الصيغة لخوارزمية Mille-Rabin على ضمان الحصول على شواهد عديدة على كون العدد مركّب.

للاعداد الاولية خواص مميزة ، منها:

- عندما يكون p عدد اولي وa عدد موجب صحيح اصغر من p فأذا كان $a^2 \bmod p = 1$ فإن $(a \bmod p)^2 = 1$ يكون صحيحا ، فقط في حالتي $a \bmod p = 1$ او $a \bmod p = -1$.
- الخاصية الاخرى اي عدد اولي $p \geq 3$ يكتب بالشكل $p - 1 = 2^k q$ حيث q عدد فردي و $k > 0$. عندما يكون لدينا العدد الصحيح $a \in \{1, \dots, p-1\}$ فانه يتحقق احد الشرطين التاليين:
  1. $a^q = 1 \bmod p$.
  2. احد اعداد السلسلة
  $a^q, a^{2q}, a^{4q}, \dots, a^{2^{k-1}q}$ يساوي $-1 \bmod p$.

وهذا يعني وجود عدد $1 \geq j \geq k$ بحيث ان $a^{2^{j-1}q} = -1 \bmod p$ ، ويمكن برهنة هذه الحقيقة استنادا الى نظرية Fermat التي تنص ان $a^{p-1} \bmod p = a^{2^k q} \bmod p = 1$. مما يعني ان اخر عنصر في السلسلة $a^q \bmod p, a^{2q}, a^{4q}, \dots, a^{2^k q}$ له القيمة 1.

- لاحظ ان كل عدد في السلسلة هو تربيع العدد السابق ، مما يعني ان احد الحالتين التالية تكون صحيحة:

1- العدد الاول في السلسلة له قيمة 1.
2- احد اعداد السلسلة ليس له قيمة 1 ولكن مربعّه يساوي 1. وهذه الحالة تكون صحيحة عندما يكون احد عناصر السلسلة يساوي -1.

- نستنتج مما سبق ، ان العدد N اذا كان اوليا ،
  - فاما ان اول عنصر في السلسلة يساوي 1
  - او ان احد العناصر الاخرى في السلسلة يساوي -1.
- لاحظ ان هذا الشرط لايتحقق دائما ، حيث ان احتمالية خطأ الخوارزمية لايتجاوز $1/4$ (الاثبات يتعدى نطاق الدرس)، لذا عند تكرار تنفيذ الخوارزمية t من المرات ولقيم a مختلفة فإن احتمالية الخطأ تكون اقل من $(1/4)^t$.
- توضّح الخوارزمية (5.9) طريقة عمل طريقة Miller-Rabin لفحص الاعداد الاولية.

Algorithm (5.9): **Miller–Rabin primality test**
Input: N: number to be tested, t: test parameter
Output: Decision: N is prime or composite
Let $N-1=2^k q, q$ is odd
for $j=1$ to t:
  $a \leftarrow \{1,\dots,N-1\}$
  $a=a^q \bmod N$
  if $a=1 \bmod N$ then **return** "prime"
  for $i=0$ to $k-1$:
    if $a=-1 \bmod N$ then **return** "prime"
    else $a=a^2 \bmod N$
  end
end
return "composite"

مثال(5.17): لاختبار كون العدد 561 اولي ام لا (وهو عدد شاذ كما ذكرنا) باستخدام خوارزمية Miller-Rabin. نستخدم عدد المحاولات $t=1$، وليكن $a=2$ نكتب $(n-1)=560=2^4(35)=2^k q$. نحسب الأن

$$2^{35} \bmod 561 = 263$$

$$2^{2.35} = 263^2 \bmod 561 = 166$$

$$2^{4.35} = 166^2 \bmod 561 = 67$$

$$2^{8.35} = 67^2 \bmod 561 = 1$$

العدد الاول $2^{35} \bmod 561$ هو ليس 1 ولا -1 ، وبقية الاعداد في القائمة هي ليست -1 ، لذا فإن 2 هو شاهد في خوارزمية Miller-Rabin على ان 561 هو عدد مركّب.

مثال(5.18): لنأخذ مثال اخر ، لتكن $N=172947529$ و

$$N-1=172947528=2^3.21618441$$

نطبق Miller-Rabin بقيمة $a = 17$ ونجد ان

$$17^{21618441} \bmod 172947529 = 1$$

لذا لايعد 17 شاهد على التركيب. لنجرب الأن $a = 3$، للاسف

$$3^{21618441} \bmod 172947529 = -1$$

وبذلك فشل 3 ايضا بكونه شاهد. عندها نشك ان العدد N قد يكون اولي، ولكن عند تجربة قيمة اخرى، لتكن $a = 23$، سوف نجد ان

$$23^{21618441} \bmod 172947529 = 40063806$$

$$23^{2.21618441} \bmod 172947529 = 2257065$$

$$23^{4.21618441} \bmod 172947529 = 1$$

وبذلك فإن 23 هو شاهد بكون N هو عدد مركب. في الحقيقة فإن N هو عدد Carmichael، ولكن ليس من السهل تحليله يدويا.

5.7 خوارزميات تحليل العدد

- تم تطوير العديد من الخوارزميات لتحليل العدد الى عوامله.
- من اشهر واكفأ الخوارزميات التي تحلل اعداد كبيرة هي خوارزميات Quadratic sieve، Number field sieve،و Elliptic curve factoring. هناك خوارزميات مشهورة اخرى مثل خوارزمية Pollard's rho، خوارزمية Pollard $p - 1$، وبالتاكيد خوارزمية trial division.
- سوف نفترض ان الصحيح N المطلوب تحليله هو عدد فردي.
- وعندها فإن كان هذا العدد هو عدد مركّب فإن احد عوامله هو $p \leq \lfloor \sqrt{N} \rfloor$.
- وبالتالي فانه يكفي لخوارزمية trial division ان تجّرب جميع الاعداد الفردية التي هي اقل من $\sqrt{N}$ لايجاد احد عوامله.
- عندما تكون $N < 10^{12}$ فإن هذه الخوارزمية تعد طريقة معقولة لتحليل العدد، ولكن عندما يكون العدد كبير فاننا نستخدم خوارزميات اعقد.
- عند تحليل العدد، فاما ان نجد جميع العوامل الاولية له او نجد له عاملين فقط بالشكل $N = n_1 . n_2$ بحيث $1 < n_1, n_2 < N$. في هذا الفصل سوف نرّكز على النوع الثاني.

5.7.1 خوارزمية Pollard $p - 1$

- تعتبر من ابسط الخوارزميات التي تطبق على اعداد كبيرة.
- مدخلات هذه الخوارزمية هي العدد الفردي الصحيح N المطلوب تحليله وعدد صحيح اخر B يدعى Bound.
- تعمل هذه الخوارزمية على ايجاد العامل الاول p، حيث $p|n$، بشرط ان تكون عوامل $p - 1$ اعداد "صغيرة".
- يحتسب العامل الثاني q بسهولة عن طريق $q = N/p$.
- تعتمد هذه الخوارزمية على نظرية Fermat (5.2) التي تنص على ان:

$$a^{p-1} \bmod p = 1$$

وبالتالي فإن $2^{p-1} \bmod p = 1$

- بما ان $p-1$ هو عدد زوجي ، اذن $(p-1)|B!$.
- تعمل الخوارزمية على التطابق $a\ mod\ N = 2^{B!} \bmod N$ ،
- وبما ان $p|N$ فإن هذا يؤدي الى $a = 2^{B!} \bmod p$.
- بما ان $(p-1)|B!$ فإن $2^{B!} \bmod p = (2^{p-1})^t \bmod p = 1 \bmod p$
- اذن $p|a-1$.
- وبما ان $p|N$ هذا يؤدي الى ان $p|d$ ، بحيث

$$d = \gcd(a-1, N)$$

بحيث d هو العامل الاول.

Algorithm (5.10): Pollard p − 1 Factoring

Input: N: number to be factored, B: bound

Output: factor d

```
a = 2
for j = 2 to B:
   a = a^j mod N
   d = gcd(a − 1, N)
   if 1 < d < N:
     return d
return "failure"
```

- توضّح الخوارزمية (5.11) عمل خوارزمية $p-1$ Pollard لتحليل عدد صحيح.

نقدم الأن مثال بسيط.

مثال(5.19): نستخدم خوارزمية Pollard p-1 لتحليل العدد $N = 13927189$. نبدأ من $\gcd(2^{9!}-1, N)$ و نحسب الاس لقيم اكبر للمفكوك ، نجد ان:

$$2^{9!} - 1 \bmod 13927189 = 13867883, \gcd(2^{9!} - 1, 13927189) = 1$$

$$2^{10!} - 1 \bmod 13927189 = 5129508, \gcd(2^{10!} - 1, 13927189) = 1$$

$$2^{11!} - 1 \bmod 13927189 = 4405233, \gcd(2^{11!} - 1, 13927189) = 1$$

$$2^{12!} - 1 \bmod 13927189 = 6680550, \gcd(2^{12!} - 1, 13927189) = 1$$

$$2^{13!} - 1 \bmod 13927189 = 6161077, \gcd(2^{13!} - 1, 13927189) = 1$$

$$2^{14!} - 1 \bmod 13927189 = 879290, \gcd(2^{14!} - 1, 13927189)$$

$$= 3823$$

السطر الاخير يعطي المعامل الاول $p = 3823$ للعدد N.

- هذا العامل هو عدد اولي ،
- والعامل الاخر هو $q = N/p = 13927189/3823 = 3643$ هو ايضا عدد اولي.
- السبب الذي جعل 14! تعمل في هذا المثال هو ان $p-1$ يحلل الى عوامل صغيرة $p - 1 = 3822 = 2.3.7^2.13$.

نقدم مثال اخر بأعداد اكبر.

مثال(5.20): ليكن $N = 168441398857$. عندها

$$2^{52!} - 1 \bmod N = 67210629098, \gcd(2^{14!} - 1, N) = 1$$

$$2^{53!} - 1 \bmod N = 8182353513, \gcd(2^{53!} - 1, N) = 350437$$

اذن عند استخدام $2^{53!} - 1$ نحصل على العامل الاولي $p = 350437$ للعدد N ، والعامل الاخر هو 480661. لاحظ ان

$$p - 1 = 350436 = 2^2.3.19.29.53$$

- يجب ان نذكر انه لتجنب خطر تحليل العدد N بواسطة خوارزمية PollardP-1 فان الاعداد الاولية p ، و q يجب ان يتم احتيارها بحيث ان كل من $p-1$ و $q-1$ لاتحلل الى عوامل اولية.
- عندما تكون قيمة B قريبة من $\sqrt{N}$ فانها لاتكون اسرع من خوارزمية trial division.

5.8 خوارزميات اللوغاريتم المتقطع

يعّبر عن مسألة ايجاد اللوغاريتم المتقطع بالشكل التالي: ليكن لدينا الزمرة G ، وليكن لدينا المولّد $g \in G$ و $y \in \langle g \rangle$ ، رتبة (عدد عناصر) $\langle g \rangle$ هو q ، اوجد x بحيث $g^x = y \bmod q$.

- يمكن استعمال مهاجمة brute-force لمعرفة قيمة x وذلك بحساب جميع القيم g^1 ، g^2 ، g^3 ، .... لحين الحصول على $y = g^x$.
  - من الواضح ان هذه الطريقة تحتاج الى $\mathcal{O}(q)$ وقت و $\mathcal{O}(1)$ خزن.
- هناك اسلوب اخر وهو توليد جميع القيم g^i ابتداءا وتشكيل قائمة مرتبة من الازواج (i, g^i) اعتماد على القيمة الثانية.
- عند تقديم y يتم البحث بهذه القائمة لغرض ايجاد تطابق $y = g^x$ واسترجاع قيمة x.
  - يتطلب هذا الاسلوب $\mathcal{O}(q)$ خزن. لذا نحتاج الى تصميم خوارزميات تجد قيمة x بوقت و وخزن اقل.

5.8.1 خوارزمية Shanks

- طورّت هذه الخوارزمية من قبل Shanks وتدعى احيانا بخوارزمية Baby-step/Giant-step.
- تعتمد هذه الخوارزميه على مبدأ مهاجمة "الالتقاء في المنتصف" لحساب اللوغاريتم المتقطع.
- في هذه الخوارزمية يتم اعادة كتابة x بالشكل:
- $x = im + j$ ، بحيث $m = \lceil \sqrt{q} \rceil$ ، $0 \le i, j < m$ ، وبذلك فإن $g^x = y$ يصبح بالشكل $g^{im+j} = y$ ، مما يعني $g^{im} = y(g^{-1})^j$.

تعمل الخوارزمية على حساب g^{im} لتشكل القائمة $L_1 = (i, g^{im})$ لجميع قيم i. ثم ترفع العدد $g^{-1}y$ لجميع قيم j للحصول على القائمة $L_2 = (j, yg^{-j})$، ومن ثم تقوم الخوارزمية بالبحث عن العنصر المشترك z، $(i, z) \in L_1$ و $(j, z) \in L_2$ لتخرج الناتج بالشكل $im + j$.تعمل الخوارزمية بوقت $\mathcal{O}(m)$ وبخزن $\mathcal{O}(m)$.

توضح الخوارزمية (5.11) طريقة عمل خوارزمية Shanks لايجاد اللوغاريتم المتقطع.

Algorithm (5.11): Shanks algorithm

Input: Group G, generater g, group order q, integer y

Output: x: $g^x = y \bmod q$

$m = \lceil \sqrt{q} \rceil$

for $i = 0$ to $m - 1$:

compute g^i and store (i, g^{mi}) in L_1

Sort L_1 according to the second coordinate

for $j = 0$ to $m - 1$:

compute g^{-j} and store (j, yg^{-j}) in L_2

Sort L_2 according to the second coordinate.

Find $(i, z) \in L_1$ and $(j, z) \in L_2$

return $im + j \bmod q$

مثال(5.22): افترض اننا نرغب بحساب $\log_3 525$ في الزمرة $\mathbb{Z}_{809}^*$.

- لاحظ ، ان 809 هو عدد اولي وبذلك فإن $q = 808$، $g = 3$، $y = 525$، و $m = \lceil \sqrt{808} \rceil = 29$.
- عندها $g^{29} \bmod 809 = 99$.
- نحسب اولا القائمة المرتبة $(i, 99^i : i = 0 \dots 28)$ لنحصل على القائمة L_1.
- ثم نحسب القائمة المرتبة $\left(j, 525 \times \left(3^j\right)^{-1} \bmod 809\right) : j = 0 \dots 28,$ للحصول على القائمة L_2.
- من القائمتين نجد العنصر المشترك $(10,644)$ في L_1 و $(19,644)$ في L_2،
- هنا ، يكون الاخراج هو $(im + j) = (10 \times 29 + 19 \bmod 808 = 309$.
- يمكن التأكد من صحة الحل بفحص $3^{209} \bmod 808 = 525$.