การเปรียบเทียบประสิทธิภาพการแยกตัวประกอบของชุดข้อมูลที่มีเลขประจำหลักเดียวกัน ตั้งแต่ 2 – 20 หลัก ด้วย Pollard's rho Algorithm และ Fermat's Factorization Method

COMPARING THE PERFORMANCE FACTORIZATION OF A DATA SET WITH THE SAME NUMBER FROM 2-20, WITH THE POLLARD'S RHO ALGORITHM AND FERMAT'S FACTORIZATION METHOD

*จีรศักดิ์ พุ่มเจริญ** *ลักษนันท์ พลอยวัฒนาวงศ์*

Jeerasak Phumcharoen*, *Luxsanan Ploywattanawong*

สาขาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

Department of Information Technology, Faculty of Science and Technology, Rajamangala University of Technology Suvarnabhumi.

****Corresponding author, e-mail:** jeerasak.ph@rmutsb.ac.th*

บทคัดย่อ

งานวิจัยนี้ได้นำเสนอการเปรียบเทียบประสิทธิภาพการแยกตัวประกอบ เป็นการวิจัยเชิงทดลองด้วยวิธี Pollard's rho Algorithm และ Fermat's Factorization Method ทั้งสองอัลกอริทึมนั้นเป็นอัลกอริทึมที่ได้รับความนิยมในปัจจุบัน หากเปรียบเทียบประสิทธิภาพในการแยกตัวประกอบของตัวเลขทั่วไปแล้วนั้นทั้ง 2 อัลกอริทึม มีประสิทธิภาพในการทำงานแตกต่างกัน โดยทดลองด้วยชุดข้อมูลที่มีค่าข้อมูลเลขประจำหลักเดียวกันทั้งหมด ดังนั้นจึงนำอัลกอริทึมแยกตัวประกอบทั้ง 2 แบบ โดยใช้ชุดข้อมูลตัวเลขทั้งสิ้น 171 ชุด ซึ่งประกอบไปด้วยตัวเลขตั้งแต่ 2 หลัก จนถึง 20 หลัก โดยมีเลขประจำหลักเดียวกัน เริ่มตั้งแต่ 1-9 และเปรียบเทียบว่าทั้งสองอัลกอริทึมนั้นให้ผลลัพธ์ของเวลาและประสิทธิภาพในการแยกตัวประกอบจากชุดข้อมูลแต่ละชุด โดยการแยกตัวประกอบของชุดข้อมูลที่มีเลขประจำหลักเดียวกันทั้งหมด พบว่า การแยกตัวประกอบวิธี Pollard's rho Algorithm มีประสิทธิภาพด้านความเร็วในการแยกตัวประกอบดีกว่าวิธี Fermat's Factorization Method

คำสำคัญ: การแยกตัวประกอบ อัลกอริทึมพอลลาร์ด โร อัลกอริทึมทฤษฎีแฟร์มาต์

Abstract

This research showed the factorization to compare the results of the algorithm used to factorization. Experimental research using Pollard's rho algorithm and Fermat's factorization method, both algorithms are currently popular algorithms. Comparing the efficiency of factorization of common numbers, the two algorithms are not very different in their efficiency. Therefore, the two factorial algorithms are used, using 171 numerical data sets, consisting of 2 - 20 digits, with the same numerals from 1 to 9, and comparing the two algorithms. Comparing these two algorithms gives the results of the time and efficiency of the factorization from each set. The results show that the Pollard's rho algorithm is more efficient and faster than the Fermat's factorization method.

Keywords: Factorization, Pollard's rho Algorithm, Fermat's Factorization Method

บทนำ

การแยกตัวประกอบ คือ การแยกค่าของจำนวนเต็มนั้นๆ ให้อยู่ในรูปผลคูณของจำนวนอื่น ซึ่งเมื่อคูณตัวประกอบเหล่านั้นเข้าด้วยกันจะได้ผลลัพธ์เป็นค่าจำนวนเต็มดังเดิม จากทฤษฎีและหลักการแยกตัวประกอบที่มีความซับซ้อนนั้น ทำให้มีบทบาทสำคัญต่อการใช้งานด้านการหากุญแจ เพื่อทำการเข้ารหัสและถอดรหัสแบบ RSA ให้มีความปลอดภัยอย่างสมบูรณ์

ปัจจุบันการแยกตัวประกอบที่นิยมใช้เป็นอย่างมาก ได้แก่ การแยกตัวประกอบด้วยวิธี Pollard's rho Algorithm และ Fermat's Factorization Method ซึ่งทั้งสองวิธีมีขั้นตอนและวิธีที่แตกต่างกัน หากลองคำนวณหาค่าการแยกตัวประกอบด้วยวิธีคำนวณด้วยมือหรือแบบ Manual นั้น วิธีการคำนวณแบบ Fermat's Factorization Method ใช้ขั้นตอนคำนวณการแยกตัวประกอบสั้นกว่า Pollard's rho Algorithm [1] แต่หากใช้คอมพิวเตอร์คำนวณเพื่อแยกตัวประกอบของทั้งสองอัลกอริทึมในแต่ละชุดข้อมูลจะใช้เวลาแตกต่างกันหรือไม่ อัลกอริทึมใดเหมาะกับการแยกตัวประกอบเลขคี่หรือเลขคู่ และอัลกอริทึมทั้งสองแบบเหมาะกับกลุ่มตัวเลขแบบใด

วัตถุประสงค์ของการวิจัย

ศึกษาออกแบบ วิเคราะห์ และทดลองเพื่อเปรียบเทียบประสิทธิภาพด้านเวลาในการแยกตัวประกอบของวิธี Pollard's rho Algorithm และ Fermat's Factorization Method โดยใช้ชุดข้อมูลตัวเลขทั้งสิ้น 171 ชุด ซึ่งประกอบด้วยตัวเลขตั้งแต่ 2 หลัก จนถึง 20 หลัก โดยมีเลขประจำหลักเดียวกัน เริ่มตั้งแต่ 1-9

วิธีดำเนินการวิจัย

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

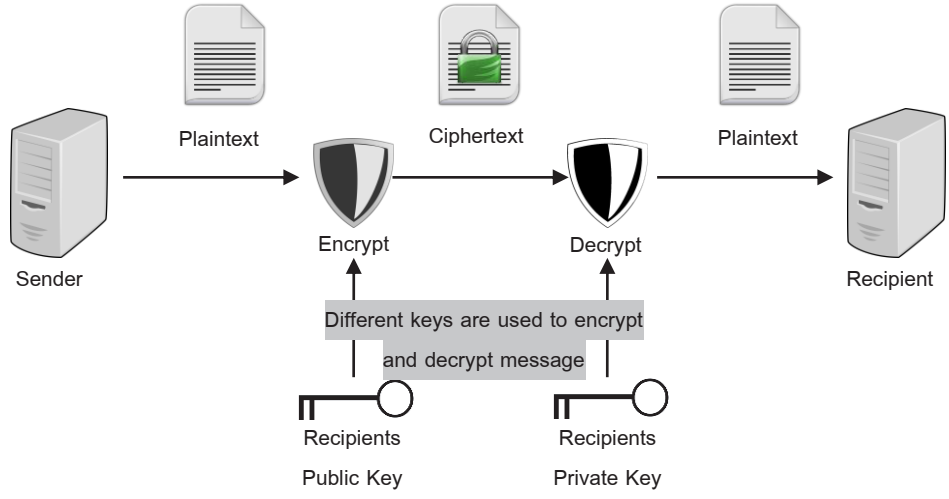
วิทยาการการเข้ารหัสลับ เป็นวิธีการที่จะช่วยแปลงข้อความให้เป็นรหัสที่บุคคลอื่นไม่สามารถอ่านเข้าใจได้ หากบุคคลเหล่านั้นไม่มีกุญแจถอดรหัสข้อความด้วยขั้นตอนวิธี (Algorithm) โดยพื้นฐานจะเกี่ยวข้องกับวิธีการทางคณิตศาสตร์ การเข้ารหัสจึงเป็นกลไกรักษาความมั่นคงปลอดภัย คำว่า "การเข้ารหัส" [2] เกี่ยวข้องกับศาสตร์และศิลป์ของการศึกษาด้านรหัสลับ (Cryptology)

การเข้ารหัสแบบ RSA

RSA (Rivest, Shamir and Adleman) [3] เป็นอัลกอลิธึมที่ถูกนำเสนอเมื่อ พ.ศ. 2520 การเข้ารหัสแบบ RSA ได้จดสิทธิบัตรโดยสถาบัน MIT ในสหรัฐอเมริกาเมื่อปี พ.ศ. 2526

อัลกอริทึม RSA เป็นวิทยาการเข้ารหัสแบบอสมมาตร ใช้กุญแจสองฝั่งในการทำงาน ตัวหนึ่งใช้ในการเข้ารหัสเรียกว่า กุญแจสาธารณะ (Public Keys Algorithms) และอีกตัวหนึ่งใช้ในการถอดรหัสเรียกว่า กุญแจส่วนตัว (Private Keys Algorithms) ในการถอดรหัสข้อมูลนั้น กุญแจสาธารณะนี้สามารถเปิดเผยได้ ส่วนกุญแจส่วนตัวนั้นห้ามเปิดเผยให้ผู้อื่นทราบโดยเด็ดขาด ดังภาพที่ 1

ปัจจุบันนิยมใช้ในการรักษาความปลอดภัยอย่างแพร่หลายบนเครือข่ายอินเตอร์เน็ต เช่น ประยุกต์ใช้กับการลงลายมือชื่ออิเล็กทรอนิกส์ (Digital Signature) การค้าผ่านอินเตอร์เน็ต (e-Commerce) เพื่อพิสูจน์ความเป็นเจ้าของในการทำธุรกรรมต่างๆ อัลกอริทึม RSA มีข้อดีคือ ง่ายในการสร้างคีย์ทั้ง 2 ฝั่ง ง่ายต่อการเข้ารหัสและการถอดรหัสข้อมูล แม้ว่าผู้โจมตีมี Public Key ก็ไม่สามารถคำนวณหา Private Key ได้ง่าย เนื่องจากต้องใช้เวลาและขั้นตอนที่ซับซ้อนมากในการคำนวณหากุญแจที่แท้จริง



ภาพที่ 1 กระบวนการเข้ารหัสลับคีย์สาธารณะ [4]

สำหรับด้านความปลอดภัยนั้นขึ้นอยู่กับความยากง่ายในการแยกตัวประกอบค่า n ออกเป็นค่า p และ q ซึ่งมีส่วนเกี่ยวข้องกับการสร้างกุญแจเข้ารหัสลับแบบ RSA รายละเอียดการสร้างกุญแจสาธารณะของอัลกอริทึม RSA [5] มีขั้นตอนดังนี้

1) เลือกจำนวนเฉพาะซึ่งมีค่ามา 2 ค่า ในที่นี้กำหนดให้เป็นตัวแปร p และ q

2) คำนวณค่าผลคูณระหว่าง p และ q โดยกำหนดให้ $n = p*q$

3) หาค่าผลคูณระหว่าง $p-1$ และ $q-1$ โดยกำหนดให้ $\phi(n) = (p-1) * (q-1)$

4) เลือกค่า e ขึ้นมา โดยค่า e และ $\phi(n)$ จะต้องเป็นจำนวนเฉพาะสัมพันธ์กัน นั่นคือค่า e และ $\phi(n)$ ต้องมีตัวหารร่วมมากเท่ากับ 1

5) คำนวณหาค่า d ซึ่งทำให้ $ed \bmod \phi(n) = 1$ นั่นคือผลคูณระหว่าง e และ d เมื่อถูกหารด้วย $\phi(n)$ จะต้องได้เศษของการหารเท่ากับ 1 โดย mod หมายถึง ตัวดำเนินการมอดูลัส (Modulus) ซึ่งมีหน้าที่หาค่าเศษของการหารออกมา

เมื่อทำครบทั้ง 5 ขั้นตอน จะได้กุญแจสาธารณะเท่ากับ (n, e) และกุญแจส่วนตัวเท่ากับ (n, d) ตัวอย่างการทดสอบเรื่องความปลอดภัยของข้อมูลที่ถูกเข้ารหัสด้วยอัลกอริทึม RSA เกิดขึ้นในปี พ.ศ. 2520 โดยผู้คิดค้นทั้ง 3 คน ได้เสนอโจทย์ปัญหาในหนังสือ Scientific American [6] ในโจทย์ได้บอกค่ากุญแจสาธารณะดังนี้

กำหนดให้

e = 9007, n = 11438162575788886 76692357799761466120102182967212423625625618429 3 5706935245733 89783059712356395870505898907514759299002687954354 1

และข้อมูลที่ถูกเข้ารหัสเป็นดังนี้

c = 96869613754622061477140922254355882905759991124574319874695120930816298225 1 45708356931476622883989628013391990551829945157815154

โจทย์ได้กำหนดให้หาข้อมูลต้นฉบับโดยใช้ข้อมูลที่กำหนดมาให้ข้างต้น สำหรับขั้นตอนที่ยากที่สุดในการถอดรหัสข้อมูลคือ การแยกตัวประกอบ n ออกเป็นจำนวนเฉพาะ p และ q เมื่อได้ค่า p และ q จะสามารถนำไปใช้ในการคำนวณหาค่ากุญแจส่วนตัว d ได้โดยง่ายโจทย์ที่ผู้คิดค้น

ทั้ง 3 คนตั้งขึ้น นักวิชาการต่างๆ ไม่สามารถหาคำตอบได้ ต่อมาในปี พ.ศ. 2537 Pual Leyland, Derek Atkins, Michael Graff และ Arjen Lenstra ได้พยายามแก้ปัญหาโจทย์นี้อีกครั้งพร้อมด้วยอาสาสมัครมากกว่า 600 คน จากประเทศต่างๆ มากกว่า 20 ประเทศ และใช้เวลา 8 เดือน จึงสามารถแยกตัวประกอบ *n* ออกเป็นจำนวนเฉพาะ 2 ตัวคูณกัน โดยจำนวนเฉพาะตัวแรก ได้แก่

34905295108476509491478496199038981334177646384933878439908205 77

และจำนวนเฉพาะ ตัวที่ 2 ได้แก่

32769132993266709549961988190834461413177642967992942539798288533

จากนั้นนำจำนวนเฉพาะทั้ง 2 ตัว มาใช้ในการคำนวณกุญแจส่วนตัวได้ผลลัพธ์ดังนี้

d = 106698614368578024442868771328920154780709906633937862801226224496631063125911774470873340168597462306553968544513277109053606095 เมื่อนำค่ากุญแจส่วนตัวมาใช้ในการถอดรหัส ได้ข้อมูลต้นฉบับดังนี้

m = 2008050013010709030023151804190001180500191721050113091908001519190906180 10705 จากนั้นให้แทนเลขต่างๆ ด้วยตัวอักขระดังนี้ 01 = A, 02 = B, ..., 26 = Z และได้ข้อความออกมาเป็นประโยคว่า “THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE”

จากตัวอย่างข้างต้นมีประเด็นสำคัญที่แสดงให้เห็นความยากในการแยกตัวประกอบ จึงสามารถสรุปถึงความปลอดภัยในการเข้ารหัสด้วยวิธี RSA ที่ได้นำทฤษฎีการแยกตัวประกอบมาใช้เป็นส่วนหนึ่งในการรักษาความลับของข้อมูล โดยการเข้ารหัสด้วยวิธี RSA นั้นขึ้นกับปัจจัยต่างๆ ดังนี้ 1) ความยาวของกุญแจที่ใช้สำหรับปัจจุบันหากกุญแจที่ใช้มีความยาวเท่ากับ 428 บิต จะมีความปลอดภัยไม่เพียงพอในการใช้งานจริง เนื่องจากประสิทธิภาพของเครื่องคอมพิวเตอร์ปัจจุบันมีความรวดเร็วกว่าเครื่องคอมพิวเตอร์ในอดีตมาก ดังนั้นการใช้งานจึงควรกำหนดให้ใช้กุญแจที่มีความยาวอย่างน้อยเท่ากับ 1024 บิต จึงจะสามารถยอมรับได้ว่าการเข้ารหัสนั้นมีความปลอดภัยสูงเพียงพอ 2) ความยากในการแยกตัวประกอบ หากมีผู้ที่สามารถคิดค้นวิธีการแยกตัวประกอบที่ง่ายและรวดเร็วกว่าวิธีที่มีทั้งหมดในปัจจุบัน วิธีการ RSA อาจไม่สามารถนำมาใช้งานได้เลย เนื่องจากอาจมีความปลอดภัยไม่เพียงพอ

นอกจากการแยกตัวประกอบที่ผู้วิจัยสนใจศึกษาแล้วนั้น ยังมีทฤษฎีการแยกตัวประกอบอื่นอีกหลายวิธีโดยตามทฤษฎีทางคณิตศาสตร์มีการแบ่งการแยกตัวประกอบออกเป็น 2 ประเภท ได้แก่ 1) อัลกอริทึมทั่วไป (General-Purpose Algorithms) อัลกอริทึมประเภทนี้ไม่ขึ้นกับปัจจัยด้านขนาดของตัวเลขเป็นสำคัญ วิธีการพื้นฐานทั่วไปคือ “Congruence of Squares” ซึ่งอัลกอริทึมที่อยู่ในกลุ่มนี้ อย่างเช่น Dixon’s, Continued Fraction Factorization, Quadratic Sieve, General Number Field Sieve, Shank’s Square Forms เป็นต้น 2) อัลกอริทึมพิเศษ (Special-Purpose Algorithms) ทฤษฎีการแยกตัวประกอบของกลุ่มนี้ใช้เพื่อการคำนวณการแยกตัวประกอบของ RSA อย่างเช่น Pallard’s p-1, Pallard’s rho, William’s p+1, Fermat’s, Euler’s เวลาในการทำงานจะขึ้นอยู่กับคุณสมบัติของตัวเลขที่เป็นปัจจัย เช่น ขนาด รูปแบบ เป็นต้น [7] สำหรับงานวิจัยนี้ผู้วิจัยสนใจศึกษาทฤษฎีการแยกตัวประกอบในกลุ่มอัลกอริทึมพิเศษ 2 ทฤษฎี ดังนี้

Factorization Method

ทฤษฎีการแยกตัวประกอบ เป็นแนวคิดที่นำมาใช้ในการสร้างความปลอดภัยของข้อมูลที่ถูกเข้ารหัสด้วยอัลกอริทึม RSA [8] ซึ่ง Fermat’s

Factorization Method และ Pollard's rho Algorithm มีวิธีในการแยกตัวประกอบแตกต่างกัน

การแยกตัวประกอบด้วยทฤษฎี Fermat's Factorization Method มีพื้นฐานในการนำเสนอเลขจำนวนเต็มคี่จากสมการผลต่างยกกำลังสอง [7] โดยแนวคิดของ Pierre De Fermat ดังสมการ (1), (2)

$$n = x^2 - y^2 \quad (1)$$

กำหนดให้ $x = (p + q)/2$ และ $y = (p - q)/2$ จะได้

$$n = ((p + q)/2)^2 - ((p - q)/2^2 \quad (2)$$

ซึ่ง n เป็นเลขคี่ เมื่อ p หรือ q เป็นเลขคี่เหมือนกัน โดย n สามารถถูกเขียนใหม่ได้ในรูปของผลต่างยกกำลังสองได้ดังนี้

$$\begin{aligned} n &= x^2 - y^2 \\ &= ((p + q)/2)^2 - ((p - q)/2^2 \\ &= \tfrac{1}{4}(p^2 + 2pq + q^2) - \tfrac{1}{4}(p^2 + 2pq + q^2) \\ &= \tfrac{1}{4}(p^2 + 2pq + q^2 - p^2 + 2pq + q^2) \\ &= \tfrac{1}{4}(4*p*q) \\ &= p*q \end{aligned}$$

จากตัวอย่างสมการ สามารถหาจำนวนเต็ม x และ y ที่ทำให้ $n = x^2 - y^2 = (x + y)*(x - y)$ นั้นจะสามารถคำนวณหาค่า p และ q ได้จาก $p = x + y$ และ $q = x - y$ ดังนั้นจึงนำแนวคิดวิธีพื้นฐานของ Fermat's Factorization Method ทำการคำนวณหาค่าความเร็วในการแยกตัวประกอบด้วยโปรแกรม MATLAB [9] ดังอัลกอริทึมต่อไปนี้

```
clear all;
promptN = 'Input Value N >> ';
n = input(promptN);

t1 = zeros(3,10);
c = 10;

tic;
tStart = tic;
for i=1:c
    if (i==1)
        a = ceil(sqrt(n));
        t1(1,i) = a;
%           disp(a)
```

```
        b2 = (a^2)-n;
        t1(2,i) = b2;
%           disp(b2)

        b = round(sqrt(b2),2);
        t1(3,i) = b;
%           disp(b)
%           disp(t1)

        bb = round(b^2,2);
    elseif (i>1)
        aa = t1(1,i-1)+1;
        t1(1,i) = aa;
%           disp(aa)

        b2 = round((aa^2)-n,2);
        t1(2,i) = b2;
%           disp(b2)

        b = round(sqrt(b2),2);
        t1(3,i) = b;
%           disp(b)
%           disp(t1)

        bb = round(b^2,2);
%           disp(bb);
    end
```

การแยกตัวประกอบด้วยทฤษฎี Pollard's rho Algorithm เกิดขึ้นในปี 1974 และ 1975 โดยแนวคิดของ J.M Pollard เป็นอัลกอริทึมแบบสุ่มขึ้นเพื่อใช้หาตัวประกอบของจำนวนเลขตัวประกอบที่มีค่ามาก ซึ่งอาศัยคุณสมบัติของการหารในการหาตัวประกอบของเลขจำนวนนั้นๆ อย่างรวดเร็ว โดยทำการแยกตัวประกอบ n โดยทำซ้ำแบบโพลิโนเมียล (Polynomial) มอดูโล (Modulo) กับค่า n [8, 10, 11] ดังนั้นจึงนำแนวคิดอัลกอริทึมของ Pollard's rho Algorithm ทำการคำนวณหาค่าความเร็วในการแยกตัวประกอบด้วยโปรแกรม MATLAB ดังอัลกอริทึมต่อไปนี้

```
%Pollard's rho algorithm%
clear all;
promptN = 'Input Value n >> ';
promptX = 'Input Value x >> ';
promptY = 'Input Value y >> ';
```

```
n = input(promptN);
x = input(promptX);
y = input(promptY);

t1 = zeros(5,4); %Can be change row's number%
r = zeros(10,4); %Can be change row's number%
e=5;    %Can be change e's numeric %
d=10;   %Can be change d's numeric%

tic;
tStart = tic;
for i=1:e
%       disp(i);
    if (i== 1)

        t1(i,1) = i;

        fx = mod((x^2+1),n);
        t1(i,2) = fx;

          fy = mod((y^2+1),n);
          ffy = mod((fy^2+1),n);
          t1(i,3) = ffy;
```

งานวิจัยที่เกี่ยวข้อง

หลายงานวิจัยได้ศึกษาถึงการแยกตัวประกอบ Fermat's Factorization Method ตามจำนวนตัวเลข 7, 49, 99 หลัก [12] และแยกตัวประกอบด้วยทฤษฎีต่างๆ [13] และศึกษาการแยกตัวประกอบ Pollard's rho Algorithm [14] นอกจากนั้นพบว่าได้มีการทดลองเปรียบเทียบความเร็วบนตัวเลขที่มีหลักแตกต่างต่างกันตั้งแต่ 2 หลักจนถึง 50 หลัก เพื่อทดลองหาความเร็วในการแยกตัวประกอบของทฤษฎีที่มีความแตกต่างกัน [7] ซึ่งผลการวิจัยพบว่า อัลกอริทึม Pollard's rho Algorithm มีความเร็วในการแยกตัวประกอบมากกว่าวิธีการต่างๆ ที่ได้เปรียบเทียบในงานวิจัย [8] อัลกอริทึม Pollard's rho Algorithm สามารถแยกตัวประกอบมีความเร็วมากที่สุดใน Special Purpose Factorization Algorithm [7] บทความวิจัยนี้ผู้วิจัยมีความสนใจที่จะศึกษาวิเคราะห์ออกแบบ และทดลองการแยกตัวประกอบด้วยวิธี Pollard's rho Algorithm และ Fermat's Factorization Method และนำมาเปรียบเทียบเรื่องเวลาในการแยกตัวประกอบของทั้ง 2 วิธี โดยเริ่มจากการวางแผน เลือกเครื่องมือ ติดตั้ง จำลอง เปรียบเทียบผล บนเครื่องคอมพิวเตอร์โน้ตบุ๊ก CPU 1.70 GHz, RAM 8 GB, บนระบบปฏิบัติการวินโดว์ 64-bit เพื่อนำผลที่ได้มาประยุกต์ใช้กับงานได้อย่างเหมาะสม

จากการทบทวนวรรณกรรมและดำเนินการศึกษาทดลองการแยกตัวประกอบด้วยวิธี Pollard's rho Algorithm และ Fermat's Factorization Method ถึงความแตกต่างกันด้านความเร็วในการแยกตัวประกอบทั้ง 2 แบบ โดยใช้ชุดข้อมูลตัวเลขทั้งสิ้น 171 ชุด ซึ่งประกอบไปด้วยตัวเลขตั้งแต่ 2 หลัก จนถึง 20 หลัก โดยมีเลขประจำหลักเดียวกัน เริ่มตั้งแต่ 1-9 และเปรียบเทียบทั้งสองอัลกอริทึมนั้น ในด้านผลลัพธ์ของเวลาในการแยกตัวประกอบจากชุดข้อมูลแต่ละชุด โดยการแยกตัวประกอบของชุดข้อมูลที่มีเลขประจำหลักเดียวกันทั้งหมด จากนั้นทำการออกแบบการทดลองการแยกตัวประกอบ โดยมีขั้นตอนวิจัยดังนี้

1) ศึกษาข้อมูลการแยกตัวประกอบด้วยวิธี Pollard's rho Algorithm และ Fermat's Factorization Method

2) วิเคราะห์และออกแบบการทดลอง

2.1) กำหนดกลุ่มตัวอย่างของชุดข้อมูลเพื่อให้อยู่ในขอบเขตที่ทำการศึกษาวิจัย ตั้งแต่ 2-20 หลัก โดยให้เลขประจำหลักเป็นเลขตัวเดียวกันทั้งหมด เริ่มตั้งแต่เลข 1-9 ในหลักต่างๆ จึงมีจำนวนรวม 171 ชุดข้อมูลในการทดลอง ดังภาพที่ 2

2.2) พัฒนาเครื่องมือในการแยกตัวประกอบด้วยวิธีการแยกตัวประกอบของ Pollard's rho Algorithm และ Fermat's Factorization Method โดยพัฒนาโค้ดการทดลองด้วยโปรแกรม MATLAB

3) ทดลองแยกตัวประกอบทั้ง 171 ชุดข้อมูล ดังภาพที่ 2 ด้วยโปรแกรมที่พัฒนาขึ้นตามขั้นตอน Pollard's rho Algorithm และ Fermat's Factorization Method โดยแต่ละชุดข้อมูลจะทำการประมวลผลเพื่อแยกตัวประกอบ 20 ครั้ง และนำมาหาค่าเฉลี่ยของการคำนวณเวลาในการแยกตัวประกอบทั้ง 20 ครั้ง เป็นการตรวจสอบและเปรียบเทียบความแตกต่างของเวลาที่ใช้ในการคำนวณ เพื่อให้ได้ค่าที่เที่ยงตรงมากที่สุด โดยหาค่าเฉลี่ยรวมของเวลาในการประมวลผลการแยกตัวประกอบด้วยโปรแกรม Microsoft Excel ดังสมการ (3)

11	111	1111	11111	111111	1111111	11111111	111111111	1111111111	11111111111
22	222	2222	22222	222222	2222222	22222222	222222222	2222222222	22222222222
33	333	3333	33333	333333	3333333	33333333	333333333	3333333333	33333333333
44	444	4444	44444	444444	4444444	44444444	444444444	4444444444	44444444444
55	555	5555	55555	555555	5555555	55555555	555555555	5555555555	55555555555
66	666	6666	66666	666666	6666666	66666666	666666666	6666666666	66666666666
77	777	7777	77777	777777	7777777	77777777	777777777	7777777777	77777777777
88	888	8888	88888	888888	8888888	88888888	888888888	8888888888	88888888888
99	999	9999	99999	999999	9999999	99999999	999999999	9999999999	99999999999

1111111111111	11111111111111	111111111111111	1111111111111111	11111111111111111
2222222222222	22222222222222	222222222222222	2222222222222222	22222222222222222
3333333333333	33333333333333	333333333333333	3333333333333333	33333333333333333
4444444444444	44444444444444	444444444444444	4444444444444444	44444444444444444
5555555555555	55555555555555	555555555555555	5555555555555555	55555555555555555
6666666666666	66666666666666	666666666666666	6666666666666666	66666666666666666
7777777777777	77777777777777	777777777777777	7777777777777777	77777777777777777
8888888888888	88888888888888	888888888888888	8888888888888888	88888888888888888
9999999999999	99999999999999	999999999999999	9999999999999999	99999999999999999

111111111111111	1111111111111111	11111111111111111	111111111111111111
2222222222222222	22222222222222222	222222222222222222	2222222222222222222
3333333333333333	33333333333333333	333333333333333333	3333333333333333333
4444444444444444	44444444444444444	444444444444444444	4444444444444444444
5555555555555555	55555555555555555	555555555555555555	5555555555555555555
6666666666666666	66666666666666666	666666666666666666	6666666666666666666
7777777777777777	77777777777777777	777777777777777777	7777777777777777777
8888888888888888	88888888888888888	888888888888888888	8888888888888888888
9999999999999999	99999999999999999	999999999999999999	9999999999999999999

ภาพที่ 2 ชุดข้อมูลตั้งแต่ 2 – 20 หลักของชุดข้อมูล

$$X_i = \frac{\sum t_i}{n} \qquad (3)$$

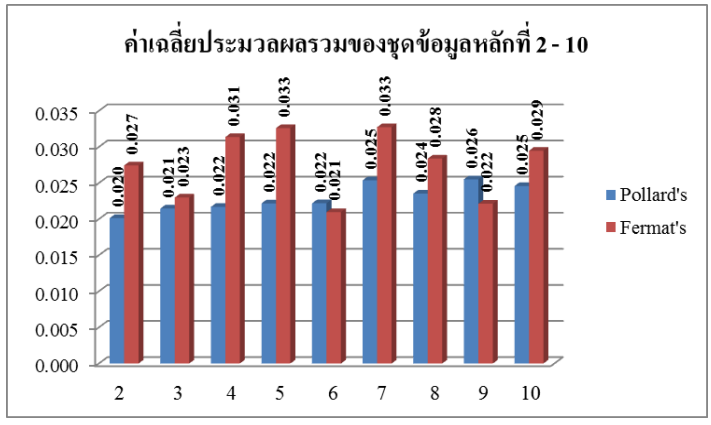
กำหนดให้ t คือ เวลาที่ใช้ในการประมวลการแยกตัวประกอบ
n คือ จำนวนครั้งทั้งหมดของการแยกตัวประกอบ
i คือ ประเภทของวิธีแยกตัวประกอบ

ซึ่งทำการทดลองเช่นนี้กับการแยกตัวประกอบทั้งสองวิธีและนำผลลัพธ์ด้านเวลาของการแยกตัวประกอบแต่ละชุดข้อมูลของทั้งสองอัลกอริทึมมาวิเคราะห์เปรียบเทียบข้อมูล

ผลการวิจัย

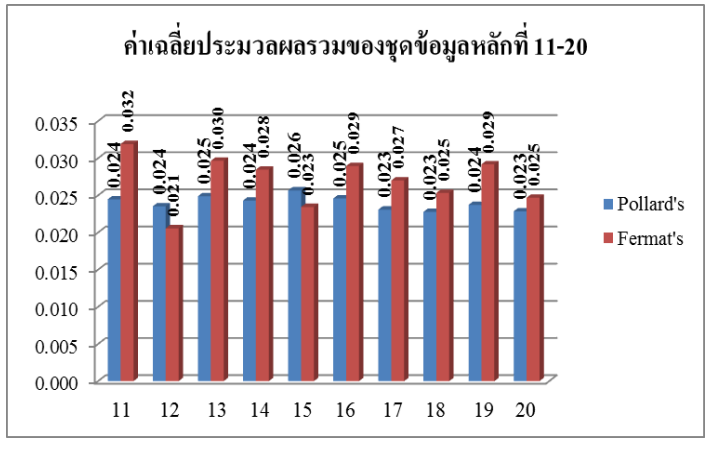
จากผลการดำเนินงานที่เกิดขึ้นจากการใช้ MATLAB แสดงให้เห็นความเร็วของการแยกตัวประกอบของ Pollard's rho Algorithm และ Fermat's Factorization Method ที่แตกต่างกันของทั้ง 171 ชุดข้อมูล โดยดำเนินการวิเคราะห์และแสดงออกมาเป็นค่าเฉลี่ย ดังนี้

1) การประมวลผลรวมของชุดข้อมูลหลักที่ 2 - 10 พบว่าชุดข้อมูลที่ทดสอบด้วยวิธีของ Pollard's rho Algorithm ใช้เวลาในการแยกตัวประกอบน้อยกว่า Fermat's Factorization Method ในชุดข้อมูลชนิด 2 – 5, 7, 8 และ 10 หลัก และ Fermat's Factorization Method ใช้เวลาในการแยกตัวประกอบน้อยกว่า Pollard's rho Algorithm ในชุดข้อมูลชนิด 6 และ 9 หลัก ดังภาพที่ 3



ภาพที่ 3 ค่าเฉลี่ยของชุดข้อมูลหลักที่ 2 - 10

2) การประมวลผลรวมของชุดข้อมูลหลักที่ 11 – 20 พบว่าชุดข้อมูลที่ทดสอบด้วยวิธีของ Pollard's rho Algorithm ใช้เวลาในการแยกตัวประกอบน้อยกว่า Fermat's Factorization Method ในชุดข้อมูลชนิด 11, 13, 14 และ 16 - 20 หลัก และ Fermat's Factorization Method ใช้เวลาในการแยกตัวประกอบน้อยกว่า Pollard's rho Algorithm ในชุดข้อมูลชนิด 12 และ 15 หลัก ดังภาพที่ 4

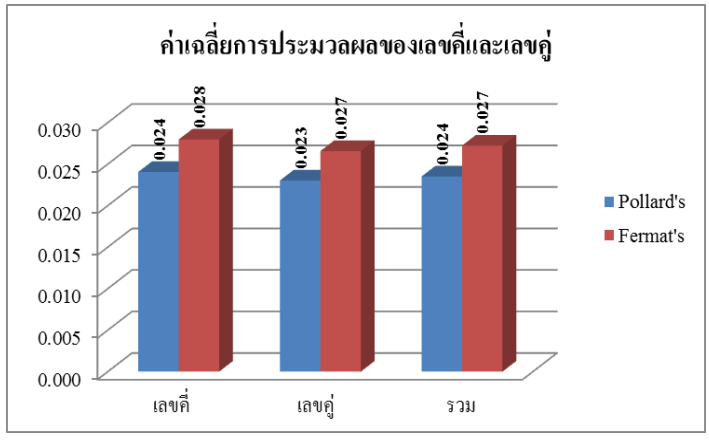


ภาพที่ 4 ค่าเฉลี่ยของชุดข้อมูลหลักที่ 11 – 20

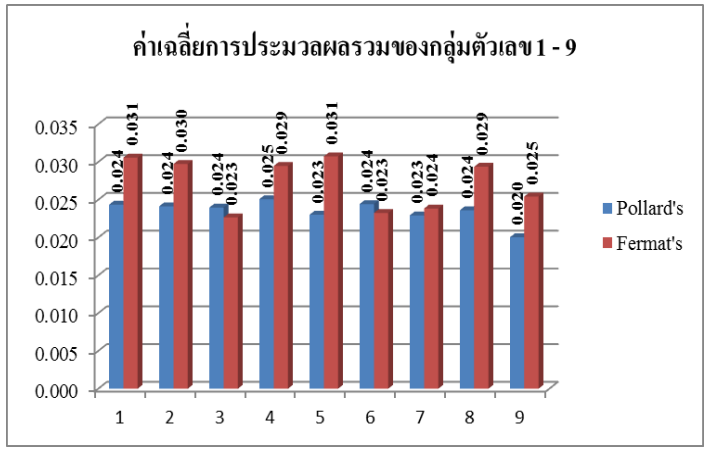
3) การประมวลผลค่าเฉลี่ยของเลขคี่และเลขคู่ พบว่าชุดข้อมูลที่ทดสอบด้วยวิธีของ Pollard's rho Algorithm ใช้เวลาในการแยกตัวประกอบน้อยกว่า Fermat's Factorization Method ทั้งในชุดข้อมูลชนิดเลขคี่และเลขคู่ ดังภาพที่ 5

4) การประมวลผลค่าเฉลี่ยรวมของกลุ่มตัวเลขตั้งแต่ 1 – 9 พบว่า หากแบ่งชุดข้อมูลโดยการประมวลผลค่าเฉลี่ยด้วยการใช้เลข 1 ตั้งแต่ 2 – 20 หลัก ใช้เลข 2...ถึง 9 ตั้งแต่ 2 – 20 หลัก เมื่อทดสอบด้วยวิธีของ Pollard's rho Algorithm ใช้เวลาในการแยกตัวประกอบ

น้อยกว่า Fermat's Factorization Method ในชุดข้อมูลเลข 1, 2, 4, 5, 7, 8 และ 9 และ Fermat's Factorization Method ใช้เวลาในการแยกตัวประกอบน้อยกว่า Pollard's rho Algorithm ในชุดข้อมูลเลข 3 และ 6 ดังภาพที่ 6



ภาพที่ 5 ค่าเฉลี่ยของชุดข้อมูลเลขคี่และเลขคู่



ภาพที่ 6 ค่าเฉลี่ยของชุดข้อมูลกลุ่มตัวเลขตั้งแต่ 1 – 9

การแยกตัวประกอบทั้ง 171 ชุดข้อมูล แสดงให้เห็นความเร็วในการแยกตัวประกอบของ Pollard's rho Algorithm และ Fermat's Factorization Method ขนาดของชุดข้อมูลของอัลกอริทึมแต่ละวิธีใช้ความเร็วในการแยกตัวประกอบต่างกัน ขนาดชุดข้อมูลที่มีจำนวนตัวเลขหลักมากกว่าจะใช้เวลาในการแยกตัวประกอบมากกว่า และนอกจากนั้นยังเห็นได้ถึงความแตกต่างระหว่าง Pollard's rho Algorithm และ Fermat's Factorization Method จะสังเกตว่าผลการแยกตัวประกอบวิธี Pollard's rho Algorithm จะใช้เวลาในการแยกตัวประกอบน้อยกว่าวิธี Fermat's Factorization Method

สรุปและอภิปรายผล

ผลการแยกตัวประกอบทั้ง 171 ชุดข้อมูล ด้านความเร็วในการแยกตัวประกอบของ Pollard's rho Algorithm และ Fermat's Factorization Method แสดงให้เห็นว่า การแยกตัวประกอบวิธี Pollard's rho Algorithm มีประสิทธิภาพด้านความเร็วในการแยกตัวประกอบได้ดีกว่าวิธี Fermat's Factorization Method ซึ่งสอดคล้องกับผลการทดลอง [6, 7] ซึ่งได้สรุปผลว่าวิธี Pollard's rho Algorithm มีความเร็วที่สุดในการแยกตัวประกอบ

สำหรับงานวิจัยนี้สามารถนำไปใช้ในการทดลองเปรียบเทียบกับการแยกตัวประกอบวิธีอื่นอย่างเช่น Dixon's, Continued Fraction Factorization, Quadratic Sieve, General Number Field Sieve, William's p+1, Euler's เป็นต้น อาจโดยการเปรียบเทียบบนเครื่องประมวลผลที่แตกต่างกัน ชุดข้อมูลที่มีจำนวนหลักของตัวเลขมากขึ้น การใช้ภาษาและโปรแกรมในการคำนวณประเภทอื่นๆ ในการหาผลลัพธ์ของเวลาในการแยกตัวประกอบ และทดสอบการแยกตัวประกอบด้วยอัลกอริทึมวิธีการใดมีความปลอดภัยสูงสุด เพื่อให้สามารถนำไปประยุกต์ใช้งานกับการเข้ารหัสลับได้อย่างเหมาะสม

เอกสารอ้างอิง

[1] Riesel, H. (2012). *Prime numbers and computer methods for factorization.* 2nd ed. The Royal Institute of Technology.

[2] Stallings, W. (2006). *Cryptography and network security: principles and practices*. Pearson Education India.

[3] Rivest, R. L., Shamir, A.; & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 21(2): 120-126.

[4] Woodward, Alan. (2012). *An emerging threat to public-key encryption.* Retrieved June, 2017, from https://www.profwoodward.org/2012/01/emerging-threat-to-public-key.html

[5] DI Management Services Pty Limited. (2017). *RSA Algorithm.* Retrieved March 20, 2017, from http://www.di-gt.com.au/rsa_alg.html

[6] Janeba, M. (1994). *Factoring Challenge Conquered - With a Little Help from Willamette.* Retrieved January 15, 2011, from http://www.willamette.edu/~mjaneba/rsa129.html

[7] Duta, C. L., Gheorghe, L.; & Tapus, N. (2016). Framework for evaluation and comparison of integer factorization algorithms. In *SAI Computing Conference (SAI), 2016.* pp. 1047-1053.

[8] Ambedkar, B. R.; & Bedi, S. S. (2011). A new factorization method to factorize rsa public key encryption. *IJCSI International Journal of Computer Science Issues*. 8(6).

[9] Phillips, C. L.; & Nagle, H. T. (2007). *Digital control system analysis and design.* Prentice Hall Press.

[10] Pollard, J. M. (1978). Monte Carlo methods for index computation (modp). *Mathematics of computation*. 32(143): 918-924.

[11] Teske, E. (2001). On random walks for Pollard's rho method. *Mathematics of computation*. 70(234): 809-825.

[12] Lenstra, A. K., Lenstra, H. W., Manasse, M. S.; & Pollard, J. M. (1993). The factorization of the ninth Fermat number. *Mathematics of Computation*. 61(203): 319-349.

[13] Brent, R. (1999). Factorization of the tenth Fermat number. *Mathematics of Computation of the American Mathematical Society*. 68(225): 429-451.

[14] Bach, E. (1991). Toward a theory of Pollard's rho method. *Information and Computation*. 90(2): 139-155.