รายงานโครงการหมายเลข COE2018-04

การเปรียบเทียบประสิทธิภาพของระบบวิทยาการรหัสลับเอ็นทรูด้วยระบบวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

นางสาวณัฐณิชา เตียวสุวรรณ เลขประจำตัว 583040768-8

นายพสิษฐ์ ติวาวงศ์รุจน์ เลขประจำตัว 583040780-8

รายงานนี้เป็นรายงาน งานโครงการของนักศึกษาชั้นปีที่ 4 ซึ่งเสนอเป็นส่วนหนึ่งในหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

พ.ศ. 2561

Project Report No. COE2018-04

The Comparative Performance of the NTRU Cryptosystem with Elliptic Curve ElGamal Cryptosystem

Mrs.	Natnisha Tieosuwan	ID 583040768-8
Mr.	Pasit Tiwawongrut	ID 583040780-8

This is the report of fourth year project assignment submitted in partial fulfillment of the requirement for the Degree of Bachelor of Engineering

Department of Computer Engineering

Faculty of Engineering, Khon Kaen University

2018

ใบประเมินผลงานโครงการ

ชื่อเรื่องภาษาไทย การเปรียบเทียบประสิทธิภาพของระบบวิทยาการรหัสลับเอ็นทรูด้วยระบบวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

ชื่อเรื่องภาษาอังกฤษ The Comparative Performance of the NTRU Cryptosystem with Elliptic Curve ElGamal Cryptosystem

ชื่อผู้ทำโครงการ

นางสาว ณัฐณิชา	เตียวสุวรรณ	เลขประจำตัว	583040768-8
นาย พสิษฐ์	ติวาวงศ์รุจน์	เลขประจำตัว	583040780-8

อาจารย์ที่ปรึกษา

____

(รองศาสตราจารย์พิเชษฐ เชี่ยวธนะกุล)

อาจารย์ผู้ร่วมประเมินผล

____

(รองศาสตราจารย์วนิดา แก่นอากาศ)

____

(อาจารย์ชวิศ ศรีจันทร์)

ประเมินผล ณ วันที่ 21 พฤษภาคม 2562

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยดีและบรรลุไปตามวัตถุประสงค์เพราะได้รับความอนุเคราะห์จากรองศาสตราจารย์พิเชษฐ เชี่ยวธนะกุล อาจารย์ที่ปรึกษาโครงงานในการแนะนำ มอบความรู้ที่เป็นประโยชน์ ชี้แนะแนวทางในการศึกษาค้นคว้าเพิ่มเติม ติดตามความก้าวหน้าในการดำเนินการวิจัย ตลอดจนแก้ไขข้อบกพร่องต่างๆ จนระทั่งโครงงานเล่มนี้สามารถสำเร็จลุล่วงและสมบูรณ์ ผู้จัดศึกษามีความซาบซึ้งในความกรุณาของอาจารย์เป็นอย่างยิ่ง และขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบคุณอาจารย์ชวิศ ศรีจันทร์ และรองศาสตราจารย์วนิดา แก่นอากาศ อาจารย์ผู้ร่วมประเมิน ที่ได้มอบความรู้เพิ่มเติมนอกเหนือจากงานวิจับครั้งนี้ ซึ่งเป็นการเปิดมุมมองใหม่ๆให้กับผู้ศึกษา ให้ข้อเสนอแนะต่างๆและช่วยตรวจทานความถูกต้อง

นอกจากนี้ทางผู้ศึกษาขอขอบคุณคณาจารย์สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ที่ให้ความอนุเคราะห์ และโอกาสในการทำวิจัยครั้งนี้

บทคัดย่อ

ในงานศึกษาครั้งนี้ เป็นงานศึกษาการเปรียบเทียบประสิทธิภาพของวิทยาการรหัสเอ็ลกามอลเส้นโค้งเชิงวงรีเทียบกับวิทยาการรหัสลับเอ็นทรู วิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีที่เราใช้ในการศึกษาครั้งนี้ เป็นวิทยาการรหัสลับที่ได้มีการพัฒนาประสิทธิภาพจากวิทยาการรหัสลับเอ็ลกามอลโดยเพิ่มการคำนวณบนเส้นโค้งคอบลิทซ์ซึ่งจะทำให้ได้ความปลอดภัยที่สูงขึ้น วิทยาการรหัสลับเอ็นทรูเป็นวิทยาการรหัสลับที่มีคุณสมบัติในการป้องกันการโจมตีจากเทคโนโลยีควอนตัมซึ่งเป็นวิทยาการรหัสลับที่พึ่งมีการเผยแพร่ได้ไม่นาน ถึงแม้ว่าวิทยาการรหัสลับเอ็นทรูจะมีประสิทธิภาพในแง่ของเวลาที่สูง แต่ยังต้องมีการศึกษาเพิ่มเติมเนื่องจากขั้นตอนการคำนวณที่ซับซ้อนซึ่งคำนวณบนแลททิซ ในการศึกษาครั้งนี้ เราได้ทำการออกแบบขั้นตอนการคำนวณเหนือริงพหุนามสังวัตนการและนำไปประยุกต์ใช้ในขั้นตอนการคำนวณของวิทยาการรหัสลับเอ็นทรู หลังจากที่ได้ทำการออกแบบขั้นตอนการคำนวณของวิทยาการรหัสลับเอ็นทรูเพื่อใช้ในการทดสอบประสิทธิภาพโดยทำงานบนคอมพิวเตอร์ส่วนบุคคล ในการทดสอบเราจะใช้การทดสอบระดับความปลอดภัยที่ใกล้เคียงกันโดยวัดเวลาจากนาฬิกาของซีพียูซึ่งสามารถใช้คำสั่งของโปรแกรมเซจในการจับเวลา

ทางผู้จัดทำได้ทำการทดลองพบว่าเราสามารถพัฒนาขั้นตอนการคำนวณบนริงสังวัตนาการและนำไปประยุกต์ใช้กับขั้นตอนการคำนวณวิทยาการรหัสลับเอ็นทรู ผลการเปรียบเทียบประสิทธิภาพพบว่า วิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีสามารถทำงานได้เร็วกว่าในขั้นตอนการสร้างกุญแจ ส่วนวิทยาการรหัสลับเอ็นทรูสามารถทำงานได้เร็วกว่าในขั้นตอนการสร้างตัวแปรเสริมสาธารณะ ขั้นตอนการเข้ารหัส และขั้นตอนการเข้ารหัส

Abstract

This project studied the performance comparison between Elliptic Curve ElGamal cryptosystem and NTRU cryptosystem. The Elliptic Curve ElGamal cryptosystem that we studied was an improvement version of the typical ElGamal cryptosystem by doing an arithmetic geometry on Koblitz curve to enhance the information security. The NTRU cryptosystem, which is a post-quantum cryptography technology, has been released to the public recently. Although it has a quick running time, but it still has to do more research due to complicating arithmetic in a lattice. In this study, we developed the algorithms to do arithmetic over convolution rings and integrated them into the NTRU cryptosystem. Then we designed the program for NTRU cryptosystem which was executed at a specific key size of the security level on personal computers to evaluate the speed by using CPU clock counting function in Sage Math.

The experiment shows that the arithmetic over convolution rings able to integrate with NTRU cryptosystem correctly. The comparison between NTRU cryptosystem and ElGamal cryptosystem show that ElGamal cryptosystem can perform faster than NTRU cryptosystem in procedures of key generation. But for the process of public parameter generation, encryption and decryption, the NTRU cryptosystem is faster than the ElGamal cryptosystem.

สารบัญ

สารบัญ(ต่อ)

สารบัญรูป

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

วิทยาการรหัสลับกุญแจสาธารณะ (public key cryptography) มีความสำคัญเพิ่มมากขึ้นในระบบสื่อสารอิเล็กทรอนิกส์และการพาณิชย์ ระบบนี้ไม่เพียงแต่ถูกนำไปใช้งานในคอมพิวเตอร์ตั้งโต๊ะเท่านั้น แต่มีการใช้งานแพร่หลายในบัตรสมาร์ต (smartcards) และอุปกรณ์สื่อสารไร้สาย ที่หน่วยความจำและความสามารถในการประระ- มวลผลจำกัด ความสำคัญของการพิสูจน์ตัวจริงกุญแจสาธารณะ ได้ปรากฏในวรรณกรรมทั้งในด้านเชิงทฤษฎีและ ด้านปฏิบัติ ดังตัวอย่างในงานของ [1, 2, 3]

เอ็นทรู (NTRU) เป็นวิทยาการรหัสลับกุญแจสาธารณะที่เพิ่งค้นพบไม่นาน [4] เอ็นทรู อยู่บนพื้นฐานของปัญหาทางคณิตศาสตร์ที่ยาก คือ ปัญหาการหาเวกเตอร์สั้นสุด (SVP: shortest vector problem) ในแลตทิซ (Lattices) จัดเป็นปัญหาประเภทเอ็นพีฮาร์ด (NP hard) จุดเด่นของเอ็นทรู คือ มีการ คำนวณที่เร็ว มีความปลอดภัยสูงและ เป็นวิทยาการรหัสลับหลังควอนตัม (post-quantum cryptography) ที่ระดับความปลอดภัยเดียวกันกับวิทยาการ รหัสลับอาร์เอ็สเอ และวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี (elliptic curve ElGamal cryptosystem) จึงเหมาะที่จะนำมาใช้กับอุปกรณ์สื่อสารไร้สาย เซ็นเซอร์โหนด (sensor node) และอุปกรณ์ฮาร์ดแวร์ขนาดเล็ก ที่หน่วยความจำ ความสามารถในการประมวลผล และมีพลังงานจำกัด ส่วนจุด ด้อยของเอ็นทรู คือ โครงสร้างการคำนวณมีความยุ่งยากมากกว่าอาร์เอ็สเอ

ประสิทธิภาพของแผนวิธีลายเซ็นเอ็นทรูและแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรียังไม่มีการ ศึกษา ดังนั้น ในงานวิจัยนี้จะทำการศึกษาเปรียบเทียบประสิทธิภาพระหว่างแผนวิธีลายเซ็นเอ็นทรูและแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาเลขคณิตมอดุลาร์ในทฤษฎีจำนวนและโครงสร้างของริง
2. ศึกษาเลขคณิตเส้นโค้งเชิงวงรีเหนือฟีลด์จำกัด
3. ศึกษาโครงสร้างของริงพหุนามผลหารเหนือฟีลด์จำกัด (quotient polynomial ring over field) และกาลัวส์ฟีลด์ (Galois field)
4. ศึกษาเลขคณิตเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง
5. ศึกษาระบบวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี
6. ศึกษาระบบวิทยาการรหัสลับเอ็นทรู
7. ทดสอบระบบวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีและระบบวิทยาการรหัสลับเอ็นทรู

1.3 ขอบข่ายของงาน

ทดสอบแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีและระบบวิทยาการรหัสลับเอ็นทรู

1.4 แนวทางการดำเนินงาน

ขั้นตอนและแผนการดำเนินโครงการทั้งภาคการศึกษาต้น และภาคการศึกษาปลายได้แสดงไว้ในตารางที่ 1.1

ตารางที่ 1.1 ขั้นตอนและแผนการดำเนินโครงการ

ขั้นตอนการดำเนินโครงการ	2561					2562			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย
1. ศึกษาเลขคณิตมอดุลาร์ในทฤษฎีจำนวน และโครงสร้างของริง	*	*							
2. ศึกษาเลขคณิตเส้นโค้งเชิงวงรีเหนือฟีลด์จำกัด	*	*							
3. ศึกษาโครงสร้างของริงพหุนามผลหารเหนือฟีลด์จำกัด และกาลัวส์ฟีลด์		*							
4. ศึกษาเลขคณิตเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง		*							
5. ศึกษาระบบวิธีวิทยาการรหัสลับ เอ็ล กา- มอลเส้นโค้งเชิงวงรี		*	*						
6. ศึกษาระบบวิทยาการรหัสลับเอ็นทรู			*						
7. ศึกษาวรรณกรรมที่เกี่ยวข้อง			*						
8. ทดสอบระบบวิธีวิทยาการรหัสลับเอ็ลกา-มอลเส้นโค้งเชิงวงรีและ ระบบวิทยาการรหัสลับเอ็นทรู			*	*	*				
9. อภิปรายผลและสรุปผล									*
10. จัดทำเอกสารประกอบโครงการ	*	*	*	*	*	*	*	*	*

* แผนงานที่ดำเนินการจริง แผนงานที่วางไว้

1.5 เครื่องมือที่ใช้ในโครงการ

1.5.1 คณิตศาสตร์

1. ทฤษฎีจำนวนและพีชคณิตการคำนวณ
2. เลขคณิตเส้นโค้งเชิงวงรี
3. วิทยาการเข้ารหัสลับ (cryptography)

1.5.2 ซอฟต์แวร์

1. SAGE (system for algebra and geometry experimentation)
2. Python

บทที่ 2
ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 เลขคณิตมอดุลาร์

ในหัวข้อนี้จะกล่าวถึงเลขคณิตมอดุลาร์ (Modular arithmetic) ซึ่งเป็นวิธีการที่สำคัญในทฤษฎีจำนวน โดย ผู้ศึกษาทำการเรียบเรียงจากงานของ Hoffstein J Pipher J and Silverman JH [2] และพิเชษฐ เชี่ยวธนะกุล [7]

บทนิยาม 2.1 [2] ให้ $m \geq 1$ เป็นจำนวนเต็ม แล้วเรียกจำนวนเต็ม a และ b ว่าเป็น*สมภาคมอดุโล (Congruent modulo)* m เขียนแทนด้วย $a \equiv b(mod\ m)$ ถ้า $a - b$ หารลงตัวด้วย m

บทนิยาม 2.2 [2] ให้ $m \geq 1$ เป็นจำนวนเต็ม แล้วเรียกเซต $\mathbb{Z}/m\mathbb{Z} = \{0,\ 1,\ 2, \dots, m-1\}$ ว่า *ริงของจำนวน เต็มมอดุโล (ring of integers modulo)* m

การดำเนินการบวกหรือคูณใน $\mathbb{Z}/m\mathbb{Z}$ จะต้องตามด้วยการหาร m แล้วใช้เศษเหลือ (remainder) เป็นคำ ตอบใน $\mathbb{Z}/m\mathbb{Z}$

บทนิยาม 2.3 [7] ให้ $m \geq 1$ เป็นจำนวนเต็ม แล้วเรียกสมาชิก a ใน $\mathbb{Z}/m\mathbb{Z}$ ว่า *ยูนิต (Unit)* ถ้า $\gcd(a,m) = 1$ และเรียกเซตของทุกยูนิตของ $\mathbb{Z}/m\mathbb{Z}$ เขียนแทนด้วย

$$\begin{aligned}(\mathbb{Z}/m\mathbb{Z})^* &= \{a \in \mathbb{Z}/m\mathbb{Z} : \gcd(a,m) = 1\} \\ &= \{a \in \mathbb{Z}/m\mathbb{Z} : a \text{ มีตัวปกผันการคุณมอดุโล } m\end{aligned}$$

ว่า *กรุปของยูนิตมอดุโล (group of units modulo)* m

ตัวอย่าง 2.1 กำหนดริงของจำนวนเต็มมอดุโล $\mathbb{Z}/9\mathbb{Z} = \{0,\ 1,\ 2,\ 3, \dots ,8\}$ จะได้ว่าสมาชิกของ $\mathbb{Z}/9\mathbb{Z}$ ที่มี $\gcd(a, 9) = 1$ เมื่อ $a \in \mathbb{Z}/9\mathbb{Z}$ คือเซท $\{1,\ 2,\ 4,\ 5,\ 7,\ 8\}$ ดังนั้นจะได้ว่า $(\mathbb{Z}/m\mathbb{Z}\,)^* = \{1,\ 2,\ 4,\ 5,\ 7,\ 8\}$

#

2.2 กรุป

ในหัวข้อนี้ผู้ศึกษาจะกล่าวถึงกรุปที่เรียบเรียงจากงานของ Lidl R and Pilz G [6] มาพอสังเขปสำหรับ รายละเอียดเชิงลึกสามารถดูได้ในตำราพีชคณิตคลาสสิก [8, 9, 10]

ให้ S เป็นเซต แล้วเรียกการส่ง (map) จาก $S \times S$ ไป S ว่า *การดำเนินการทวิภาค (binary operation)*

บทนิยาม 2.4 [6] *กรุป (group)* เขียนแทนด้วย $(G,\circ)$ หมายถึง เซต G พร้อมด้วยการดำเนินการทวิภาค $\circ : G \times G \to G$ บน G ด้วยคุณสมบัติต่อไปนี้

1. $\circ$ เป็นเปลี่ยนหมู่ (associative) สำหรับสมาชิกใดๆ $f, g, h \in G$ มี $f \circ (g \circ h) = (f \circ g) \circ h$
2. มีสมาชิกเอกลักษณ์ (Identity elementor neutralelement) $n \in G$ ที่ซึ่งสำหรับสมาชิกใดๆ $g \in G$ มี $n \circ g = g \circ n = g$
3. สำหรับสมาชิกใดๆ $g \in G$ จะมีสมาชิก $h \in G$ เรียกว่า *ตัวผกผันของ* g ที่ซึ่ง $g \circ h = h \circ g = n$ เมื่อ n แทนสมาชิกเอกลักษณ์

เรียกกรุป $(G,\circ)$ ว่า*สลับที่ (commutative)* หรือ*อาบีเลียนกรุป (Abelian group)* ถ้าสำหรับสมาชิกใดๆ $g, h\ \in G$ มี $g \circ h = h \circ g$ และเรียกจำนวนสมาชิกของ G เขียนแทนด้วย $|G|$ ว่า *อันดับ (order)* ของกรุป G

บทนิยาม 2.5 [2] ให้ $(G,\circ)$ เป็นกรุป และให้ $H \neq \emptyset$ เป็นเซตย่อยของ G แล้วเรียก H ว่า*กรุปย่อย (subgroup)* ของ G ถ้า สำหรับสมาชิกใดๆ $a, b\ \in H$ มี $a \circ b^{-1} \in H$

ให้ G เป็นกรุป และให้สมาชิก $g \in G$ ที่ซึ่งสามารถก่อกำเนิด G เขียนแทนด้วย $\langle g \rangle = G$ แล้วเรียก g ว่า *ตัวก่อกำเนิด (generator)* และเรียก G ว่า *กรุปวัฏจักร (cyclic group)*

ตัวอย่าง 2.2 กำหนดกรุป $G = \mathbb{Z}/7\mathbb{Z} = \{0, 1, 2, 3, 4, 5, 6\}$ จะมีตัวก่อกำเนิด 3 และ 6 ที่ซึ่ง $G = \{3^n | n \in \mathbb{Z}\}$ และ $G = \{5^n | n \in \mathbb{Z}\}$ เราจะเรียก 3 และ 5 ว่าตัวก่อกำเนิด และจะเรียกกรุป G ว่ากรุปวัฏจักร

$$3^0 = 1, 3^1 = 3, 3^2 = 2, 3^3 = 6, 3^4 = 4, 3^5 = 5$$
$$5^0 = 1, 5^1 = 5, 5^2 = 4, 5^3 = 6, 5^4 = 2, 5^5 = 3$$

#

2.3 ริงและพหุนาม

ในหัวข้อนี้ผู้ศึกษาจะกล่าวถึงริงและพหุนามที่เรียบเรียงจากงานของ Lidl R and Pilz G [3] มาพอสังเขป สำหรับรายละเอียดเชิงลึกสามารถดูได้ในตำราพีชคณิตคลาสสิก [4, 5, 6]

ใน *กรุปการบวก (additive group)* เราสามารถที่จะบวก หรือลบได้ ซึ่งในที่ นี้เราจะทำการศึกษาการคูณ

บทนิยาม 2.6 [6] ริง *(ring)* หมายถึงเซต (set) ของ R ร่วมกับ 2 ตัวดำเนินการทวิภาคที่แทนด้วย + และ · เรียกว่า *การบวก (addition)* และ *การคูณ (multiplication)* ถ้าหาก

1. $(R, +)$ เป็นอาบีเลียนกรุป (abelian group)
2. ผลคูณ $r \cdot s$ เมื่อ $r, s \in R$ มีคุณสมบัติการเปลี่ยนหมู่การคูณ (associative)
3. สำหรับทุก $\mathrm{r, s, t} \in \mathrm{R}$: $\mathrm{r} \cdot (\mathrm{s} + \mathrm{t}) = \mathrm{r} \cdot \mathrm{s} + \mathrm{r} \cdot \mathrm{t}$ และ $(\mathrm{r} + \mathrm{s}) \cdot \mathrm{t} = \mathrm{r} \cdot \mathrm{t} + \mathrm{s} \cdot \mathrm{t}$

เรียกว่า *กฎของสมบัติการแจกแจง (distributive laws)*

สามารถเขียนแทนด้วย $(R, +, \cdot)$ หรือ R โดยทั่วไปหากเขียนแทนด้วย $(R, +)$ องค์ประกอบที่ว่างจะแทนด้วย 0 และเรียกว่า *ศูนย์ (Zero)* ตัวผกผันการบวกของ $r \epsilon R$ จะแทนด้วย $-r$ และแทนที่การเขียน $r \times s$ ด้วย rs ให้ $R^* := R\{0\}$ ตามบทนิยามที่ 2.4 ริงจะถูกเรียกว่า *ริงเปลี่ยนหมู่ (Associative rings)* ในทางตรงกันข้ามจะถูกเรียกว่า *ริงไม่เปลี่ยนหมู่ (Non associative rings* เมื่อไม่มีสมบัติการเปลี่ยนกลุ่มของการคูณ ต้นแบบของริงคือ $(\mathbb{Z}, +, \cdot)$

บทนิยาม 2.7 [6] ให้ R เป็นริง เมื่อ R มีการคูณจะเรียกว่า *การสลับที่ (Commutative)* หรือเมื่อมี 1 ใน R เช่น $r \cdot 1 = 1 \cdot r = r$ สำหรับ $r \epsilon R$ ดังนั้นเรียก 1 ว่าเป็นสมาชิก *เอกลักษณ์ (Identity)* หรือ *หน่วย (Unit)* ถ้า $r \neq 0,\ s \neq 0$ แต่ $rs = 0$ แล้ว r คือ *ตัวหารด้านซ้าย (Left divisor)* และ s เป็น *ตัวหารด้านขวาของศูนย์ (Right divisor of zero)* ถ้า R ไม่มีตัวหารที่เป็นศูนย์ เช่น ถ้า $rs = 0$ แล้วให้ $r = 0$ หรือ $s = 0$ สำหรับทุก แล้วเรียกได้ว่า R เป็นอินทิกรัล (integral) และ

อินทิกรัลริงสลับที่ด้วยเอกลักษณ์ 1 ไม่เท่ากับ 0 จะเรียกว่า อินทิกรัลโดเมน (integral domain) ถ้า $(R^*,\cdot)$ เป็นกรุปแล้ว R เป็น *สกิวฟีลด์ (Skew field)* หรือ *ริงการหาร (Division ring)* เมื่อกล่าวถึง *ฟีลด์ (Field)* และ R มีสมบัติการสลับที่ จะกล่าวได้ว่าฟีลด์นั้นคือริง $(R,+,\ \cdot)$ เช่นเดียวกันกับ $(R,+)$ และ $(R^*,\cdot)$ จะเป็น*อาบีเลียนกรุป (Abelian group) ลักษณะเฉพาะ (Characteristic)* ของ R เป็นจำนวนที่เล็กที่สุด k ที่ $kr \coloneqq r + \cdots + r\ (k - times)$ เท่ากับ 0 สำหรับทุก $r \epsilon R$ ดังนั้น เราจะเขียนได้ว่า $k = char\ R$ เมื่อไม่มี k ที่มีจริง เราจะให้ $char\ R = 0$ แล้วเมื่อ $k = char\ R$ ทุกสมาชิกในกรุป $(R,+)$ มีลำดับการหาร k

ตัวอย่าง 2.3 $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C},$ และ $n\mathbb{Z}$ เมื่อทุกตัวเป็น อินทิกรัลริงสลับเนื่องจาก การบวก, $+$, และการคูณ, $\cdot$ โดยทุกตัวมี 1 เป็นเอกลักษณ์ ยกเว้น $n\mathbb{Z}\ (n \geq 2)$ ที่ไม่มีเอกลักษณ์ ทุกสมาชิก x ที่ไม่เป็นศูนย์ ใน $\mathbb{Q}, \mathbb{R}$ และ $\mathbb{C}$ มีการคูณแบบผกผัน $x^{-1} = \frac{1}{x}$ ในเซทเดียวกันตามลำดับ ดังนั้นเมื่อ $\mathbb{Q}, \mathbb{R}$ และ $\mathbb{C}$ เป็นฟีลด์ แล้ว $\mathbb{Z}$ คืออินทิกรัลโดเมน ทุกริงมี 0 เป็นลักษณะเฉพาะ

#

บทนิยาม 2.8 [6] กำหนด R เป็นริงและ SR จะเรียก S ว่า *ซับริง (Subring)* ของ R (เขียนแทนด้วย $S \leq R$) ถ้า S เป็นริงที่มีการดำเนินการซึ่งถูกกำหนดโดย R

ทฤษฎีบท 2.1 ให้ $(R,+,\ \cdot)$ เป็นริงถ้า

1. ถ้า $\sim$ สมภาคกัน ภายใต้ $(R,+,\ \cdot)$ แล้ว $[0]$ จะเป็นซับริง I ของ R ซึ่งมีคุณสมบัติสำหรับทุก $r \in R$ และ $i \in I$ แล้ว

$$ri \in I \text{ และ } ir \in I \tag{2.1}$$

2. ในทางกลับกัน ถ้า $I \leq R$ ที่สอดคล้องกับ (2.1) แล้ว

$$r \sim_I s :\Leftrightarrow r - s \in I$$

บทนิยาม 2.9

1. ซับริง I ของริง R ซึ่งมีคุณสมบัติ (2.1) จะเรียกว่า *ไอดีล (ideal)* ของ R สามารถเขียนแทนด้วยสัญลักษณ์ $I \trianglelefteq R$
2. ถ้าสมภาค $\sim$ นั้นมาจากไอดีล I เช่น $\sim\, = \,\sim_I$ แล้วจะเขียนแทน R/I ด้วย $R/\sim_I$

จากเงื่อนไข (2.1) มักเขียนแทนด้วย "$IR \subseteq I\ and\ RI \subseteq I$" จะได้ว่า $I \lhd R$ ถ้า $I \unlhd R$ แต่ $I \neq R$

ยกตัวอย่าง $n\mathbb{Z} \unlhd \mathbb{Z}$ สำหรับทุก $n \epsilon \mathbb{N}_0$ในทางกลับกัน $\mathbb{Z}$ เป็นซับริงแต่ไม่ใช่ไอดีลของ $\mathbb{Q}$ ตามข้อเท็จจริงแล้ว ถ้าไอดีลของ I ประกอบด้วยเอกลักษณ์ 1 ของ R แล้วจะกล่าวได้ว่า $1 = R$ ยิ่งไปกว่านั้นถ้า F เป็นสกิวฟิลด์ และ $I \neq \{0\}$ เป็นไอดีลของ F ที่เอา $i \epsilon I^*$ แล้ว $i^{-1}i \epsilon I$ ดังนั้น $1 \epsilon I$ และ $I \epsilon F$

อินเตอร์เซกชันของไอดีล R จะมีผลลัพธ์เป็นไอดีล เราสามารถกล่าวได้ว่าแนวคิดของ *ไอดีลก่อกำเนิด (generated ideal)* คือไอดีลที่ถูกก่อกำเนิดมาจากสมาชิก 1 ตัว

กำหนดให้ R เป็นริงและ $a \in R$ ไอดีลที่ก่อกำเนิดโดย a เขียนแทนด้วย (a) และถูกเรียกว่า ไอดีลมุขสำคัญ *(principle ideal)* ถ้าหาก R เป็นริงสลับที่ซึ่งมีเอกลักษณ์แล้วสำหรับทุก $a \in R$ เราจะได้ว่า $(a) = aR = \{ar | r \in R\}$ สำหรับริงที่มีเอกลักษณ์ $\{0\} = (0)$ และ $R = (1)$ จะเป็นไอดีลมุขสำคัญ ใน $\mathbb{Z}$ จะได้ $n\mathbb{Z} = (n)$ เป็น principle ideal สำหรับทุก $n \in \mathbb{N}_0$

สำหรับอินทิกรัลโดเมนที่ทุกไอดีลที่เป็น principle จะเรียกว่า principle ideal domain (PID) เช่น $\mathbb{Z}$ เป็น PID และเช่นเดียวกันกับริงพหุนาม $R[X]$ ถ้าหาก R เป็นฟีลด์

ไอดีล I ใน R จะเรียกว่า *ไอดีลใหญ่สุด (maximal ideal)* ถ้า $I \neq R$ และไม่มีไอดีลอยู่ระหว่าง I และ R ในแบบฝึกหัดข้อที่ 11 เราจะเห็นได้ว่า ไอดีล n เป็นไอดีลใหญ่สุดใน $\mathbb{Z}$ ก็ต่อเมื่อ n เป็นจำนวนเฉพาะ

ทฤษฎีบท 2.2 ให้ $I \lhd R$ และ R เป็นริงสลับที่ซึ่งมีเอกลักษณ์แล้ว I จะใหญ่ที่สุดก็ต่อเมื่อ R/I เป็นฟีลด์

บทนิยาม 2.10 สำหรับส่วนที่เหลือในบทนี้ กำหนดให้ R เป็นริงสลับที่ซึ่งมีเอกลักษณ์ ทุกลำดับของสมาชิก R ที่มีเฉพาะสมาชิกจำกัดที่ไม่เป็นศูนย์หลายตัวจะเรียกว่า *พหุนาม (polynomial)* บน R ทุกลำดับของสมาชิกใน R จะเรียกว่า *อนุกรมกำลังรูปนัย (formal power series)* บน R เซทของพหุนามบน R เขียนแทนด้วย $R[x]$ เซทของอนุกรมกำลังบน R เขียนแทนด้วย $R[[x]]$ ถ้าหาก $p = (a_0, a_1, \dots, a_n, 0, 0, \dots) \in R[x]$ เราจะเขียนด้วย $p = (a_0, a_1, \dots, a_n)$ ถ้า $a_n \neq 0$ แล้ว

เราจะเรียก n ว่า *ระดับขั้น (degree)* ของ p $(n = deg p)$; ถ้า $a_n = 1$ เราเรียก p ว่า *โมนิก (monic)* ใส่ $\deg(0,0,0,\dots) \coloneqq -\infty$ พหุนามของดีกรี ≤ 0 เรียกว่า *ค่าคงตัว (constant)*

ใน $R[x]$ และ $R[[x]]$ เราจะนิยามการบวกในรูปของตัวประกอบว่า $(a_0, a_1, \dots) + (b_0 + b_1, \dots) \coloneqq (a_0 + b_0, a_1 + b_1, \dots)$ และการคูณในรูปของตัวประกอบว่า $(a_0, a_1, \dots) \cdot (b_0 + b_1, \dots) \coloneqq (c_0, c_1, \dots)$ โดยที่ $c_k \coloneqq \sum_{i+j=k} a_i b_j = \sum_{i=0}^{k} a_i b_{k-i}$ และจากแบบฝึกหัดข้อที่ 18 [อ้างอิง] ว่า $\deg pq = \deg p + \deg q$ สำหรับ $p, q \in R[x]$ ถ้า R เป็นอินทิกรัล

จากตัวดำเนินการการบวกและการคูณของเซท $R[x]$ และ $R[[x]]$ จะเป็นริงสลับที่ด้วยเอกลักษณ์ $(1, 0, 0, \dots)$ ถ้า R เป็นอินทิกรัลจะได้ $R[x]$ และ $R[[x]]$ จะเป็นริงสลับที่เช่นกันจากสมการ $\deg pq$ จากการสังเกตกรุปการบวกของ $R[x]$ และ $R[[x]]$ เป็นเพียงการบวกโดยตรง (การคูณโดยตรง, ตามลำดับ) ของกรุป $(R, +)$ ซ้ำกันหลายกรุป

ริง $(R[x], +, \cdot)$ และ $(R[[x]], +, \cdot)$ จะเรียกว่า *ริงพหุนามบน R (ring of polynomials over R)* และ *ริงอนุกรมกำลังรูปนัยบน R (ring of formal power series over R)* ตามลำดับ ใน $R[x]$ และ $R[[x]]$ เรากำหนด $x \coloneqq (0, 1, 0, 0, \dots) = (0,1)$ เราจะได้ $x \cdot x = x^2 = (0, 0, 1), x^3 = (0, 0, 0, 1)$ ไปเรื่อยๆ ด้วย $x^0 = (1, 0, 0, 0, \dots)$ และ $a_i = (a_i, 0, 0, \dots)$ เราจะเห็นได้ว่าใน $R[x]$ และ $R[[x]]$ เราสามารถเขียนได้ว่า

$$p = (a_0, a_1, a_2, \dots) = a_0 + a_1 x + a_2 x^2 + \cdots =: \sum_{i \geq 0} a_i x^i$$

จะได้รูปทั่วไปของพหุนามคือ $\sum_{i=0}^{n} a_i x^i$ และอนุกรมกำลังรูปนัยคือ $\sum_{i=0}^{\infty} a_i x^i$ (เรียกว่ารูปนัยเพราะว่าเราไม่มั่นใจเกี่ยวกับการลู่เข้า) จะเห็นว่า x ไม่ใช่ “ยังไม่กำหนด (indeterminate)” หรือ “สัญลักษณ์ (Symbol)” แต่เป็นเพียงอนุกรมพิเศษ

ถ้า $p, q \in R[x]$ และ $p = a_0 + a_1 x + \cdots + a_n x^n$เราจะนิยามการประกอบ $p \circ q$ ด้วย $a_0 + a_1 q + \cdots + a_n q^n$ เราจะได้ว่า $(p_1 + p_2) \circ q = p_1 \circ q + p_2 \circ q$ และ $(p_1 p_2) \circ q = (p_1 \circ q)(p_2 \circ q)$ ด้วยการคำนวณแบบง่าย

ให้ $p, q \in R[x]$ เราจะกล่าวได้ว่า p หาร q (เขียนแทนด้วย $p|q$) ถ้า $q = p \cdot r$ สำหรับ บางค่า $r \in R[x]$ ถ้า $\deg q > \deg p > 0$ แล้ว p จะเรียกว่า *ตัวหารแท้ (proper devisor)* ของ q พหุนาม q ที่มี $\deg q \geq 1$ที่ไม่มีตัวหารแท้จะเรียกว่า *ลดทอนไม่ได้ (irreducible)*

2.4 ฟีลด์

ในหัวข้อนี้ศึกษาริง $(\mathbb{Z}_n, +, \cdots)$ ที่เป็นฟีลด์ โดยเรียบเรียงจากงานของ พิเชษฐ เชี่ยวธนะกุล [7] และ Lidl R and Pilz G [6]

ทฤษฎีบท 2.3 [6,7] ให้ $\mathbb{N}$ เป็นเซตของจำนวนเต็มบวก และ $n \in \mathbb{N}$ และให้ $\mathbb{P}$ เป็นเซตของจำนวนเฉพาะ แล้วข้อความต่อไปนี้สมมูลกัน

1. $\mathbb{Z}_n$ เป็นอินทิกรัลโดเมน
2. $\mathbb{Z}_n$ เป็นฟีลด์
3. $n \in \mathbb{P}$

สำหรับริง R ที่ซึ่งสามารถฝัง (embedded) ในฟีลด์ F เขียนแทนด้วย $R \hookrightarrow F$ จะเป็นทั้งริงสลับที่ และ เป็นอินทิกรัลโดเมน ในทางกลับกันก็ได้ความสัมพันธ์

ทฤษฎีบท 2.4 [6] ให้ R เป็นอินทิกรัลโดเมน และ $R \neq \{0\}$ แล้วมีฟีลด์ F ที่ซึ่งมีคุณสมบัติต่อไปนี้

1. $R \hookrightarrow F$
2. ถ้า $R \hookrightarrow F'$ และ F' เป็นฟีลด์ แล้ว $F \hookrightarrow F'$

ดังนั้นอินทิกรัลโดเมนใดๆสามารถฝังในฟีลด์ที่เล็กสุด (minimal field) เรียกฟีลด์ในทฤษฎีบท 2.4 ว่า *ฟีลด์ผลหาร (quotient field)* ของ R

บทนิยาม 2.11 [6] เรียกเซตย่อย U ของฟีลด์ F ว่า *ซับฟีลด์ (subfield)* ของ F เขียนแทนด้วย $U \leq F$ ถ้า U เป็นซับริงของ F และ U เป็นฟีลด์ด้วยการดำเนินการใน F ถ้า $U \neq F$ แล้วเรียก $(U, +, \cdot)$ ว่า*ซับฟีลด์แท้ (proper subfield)* ของ $(F, +, \cdot)$ เขียนแทนด้วย $U < F$ และเรียก $(F, +, \cdot)$ ว่า *ฟีลด์ภาคขยาย (extension field)* ของฟีลด์ $(U, +, \cdot)$ ถ้า $(U, +, \cdot)$ เป็นซับฟีลด์ของ $(F, +, \cdot)$ นอกจากนี้ เรียกฟีลด์ P ว่า *ฟีลด์เฉพาะ (prime field)* ถ้า P ไม่มีซับฟีลด์แท้

ถ้า $p = (a_0, \dots, a_n) = a_0 + a_1x + \cdots + a_nx^n \in R[x]$ แล้วเรียก $\bar{p} : R \to R; r \mapsto a_0 + a_1r + \cdots + a_nr^n$ ว่า *ฟังก์ชันพหุนาม (polynomial function)* อินดิวซ์ (induced) ด้วย p

บทนิยาม 2.12 [6] สมาชิก r ในบางฟีลด์ภาคขยายของฟีลด์ F เรียกว่า *ราก (root)* หรือ*ศูนย์ (zero)* ของ $p \in F[x]$ ถ้า $\bar{p}(r) = 0$

สมมุติให้ f เป็นหุนามดีกรี k เหนือฟีลด์ F กำหนดให้ $g + (f)$ เป็น สมาชิกคงที่ (arbitrary element) ใน $F[x]/(f)$ ด้วยวิธีหารแบบยุคลิด (euclidean division) จะได้ $h, r \in F[x]$ โดยที่ $g = hf + r$ เมื่อ $deg\, r < k$ จาก $hf \in (f)$ ทำให้ $g + (f) = r + (f)$ ดังนั้นสมาชิกแต่ละตัวของ $F[x]/(f)$ สามารถเขียนได้ในรูปที่แตกต่างคือ

$$a_0 + a_1x + \cdots + a_{k-1}x + (f), a_i \in F \tag{2.2}$$

ถ้าเรากำหนด F ด้วยซับริง $\{a + (f) | a \in F\}$ ของ $F[x]/(f)$ แล้วสมาชิกใน (2.2) สามารถเขียนแทนด้วย $a_0 + a_1(x + (f)) + \cdots + a_{k-1}(x + (f))^{k-1}$ ถ้า $x + (f) \coloneqq \alpha$ เราสามารถเขียนด้วย

$$a_0 + a_1\alpha + \cdots + a_{k-1}\alpha^{k-1} \tag{2.3}$$

และเราสามารถพิจารณา $F[x]/(f)$ เป็นปริภูมิเวกเตอร์ (vector space) เหนือฟีลด์ F ด้วยฐานหลัก $\{1, \alpha, \alpha^2, \dots, \alpha^{k-1}\}$

จาก $0 + (f)$ เป็นสมาชิกศูนย์ของ $F[x]/(f)$ จะได้ $\bar{f}(\alpha) = f + (f) = 0 + (f)$ นั้นก็คือ α เป็นรากของ f จะเห็นได้ว่า α เป็นสมาชิกใน $F[x]/(f)$ แต่โดยปกติแล้วจะไม่อยู่ใน F เพราะฉะนั้นสมาชิกใน $F[x]/(f)$ ในรูป (2.3) สามารถพิจารณาให้ α เป็นสมาชิกด้วยคุณสมบัติ $\bar{f}(\alpha) = 0$

ทฤษฎีบท 2.5 [6] ให้ F เป็นฟีลด์และ $f \in F[x]/(f)$ ในรูป (2.3) ด้วย $deg\, f = k$ แล้ว $F[x]/(f) = \{a_0 + a_1x + \cdots + a_{k-1}x^{k-1} | a_i \in F\}$ เป็น*ปริภูมิเวกเตอร์ k มิติ (k-dimensional vector space)* เหนือฟีลด์ F ด้วยฐานหลัก $\{1, \alpha, \alpha^2, \dots, \alpha^{k-1}\}$ เมื่อ $\alpha = [x] = x + (f)$ จะมี $\bar{\alpha} = 0$ และ $F[x]/(f)$ เป็นฟีลด์ก็ต่อเมื่อ f ลดทอนไม่ได้

ตัวอย่าง 2.4 [6]

1. ให้ F เป็นฟีลด์ $\mathbb{Z}_2 = \{0,1\}$ แล้ว $f = x^2 + x + 1$ เป็นพหุนามลดทอนไม่ได้ด้วยดีกรี 2 เหนือ $\mathbb{Z}_2$ ดังนั้น $\mathbb{Z}_2[x]/(x^2 + x + 1)$ เป็นฟีลด์ที่ซึ่งสมาชิกสามารถเขียนในรูปของ $a + b\alpha, a, b \in \mathbb{Z}_2$ เมื่อ α เป็นไปตาม $\bar{f}(\alpha) = 0$ นั่นก็คือ $\alpha^2 + \alpha + 1 = 0$ ซึ่งหมายความว่า $\alpha^2 = \alpha + 1$ จาก $-1 = 1$ ใน $\mathbb{Z}_2$ แล้ว ดังนั้น $\mathbb{Z}_2[x]/(x^2 + x + 1)$ เป็นฟีลด์ซึ่งมีสมาชิก 4 ตัว

$$\mathbb{Z}_2[x]/(x^2 + x + 1) = \{0,1,\alpha,1+\alpha\}$$

เช่น $\alpha \cdot (1 + \alpha) = \alpha + \alpha^2 = \alpha + \alpha + 1 = 1$ ตารางของการบวกและการคูณสามารถแสดงได้ดังนี้

$+$	0	1	α	$1+\alpha$
0	0	1	α	$1+\alpha$
1	1	0	$1+\alpha$	α
α	α	$1+\alpha$	0	1
$1+\alpha$	$1+\alpha$	α	1	0

$\cdot$	0	1	α	$1+\alpha$
0	0	0	0	0
1	0	1	α	$1+\alpha$
α	0	α	$1+\alpha$	1
$1+\alpha$	0	$1+\alpha$	1	α

2. ในทางเดียวกัน

$$\mathbb{Z}_2[x]/(x^3 + x + 1) = \{0,1,\alpha,1+\alpha,\alpha^2,1+\alpha^2,\alpha+\alpha^2,1+\alpha+\alpha^2\}$$

เป็นฟีลด์ที่มีสมาชิก 8 ตัว โดย $\alpha^3 = \alpha + 1$

3. $\mathbb{Z}_3[x]/(x^2 + 1) = \{0,1,2,\alpha,1+\alpha,2+\alpha,2\alpha,1+2\alpha,2+2\alpha\}$

เมื่อ $\alpha^2 = -1 = 2$ เป็นฟีลด์ที่มีสมาชิก 9 ตัว

4. ในทฤษฎีบท 2.5. จะพบว่าฟีลด์ในหัวข้อ ก. ข. และ ค. จะเป็นฟีลด์ที่เล็กที่สุดที่ซึ่งไม่ได้เป็นชนิด $\mathbb{Z}_p$

#

ทฤษฎีบท 2.6 [6]

1. ฟีลด์จำกัด F ใดๆด้วยจำนวนสมาชิก p^n เมื่อฟีลด์ $\mathbb{Z}_{p^n}$ นั้นมี p เป็นจำนวนเฉพาะและ n เป็นจำนวนเต็มที่มากกว่า 1

2. สำหรับทุกจำนวนเฉพาะ p และทุก $n \in \mathbb{N}$ จะเป็นฟีลด์ด้วยจำนวนสมาชิก p^n

3. ฟีลด์ใดๆด้วยจำนวนสมาชิก p^n จะเป็นฟีลด์ที่แยกได้ของ $x^{p^n} - x$ และ $x^{p^n-1} - 1 \in \mathbb{Z}_p[x]$ ขึ้นอยู่กับ *สมสัณฐาน (isomorphism)*

บทนิยาม 2.13 [8] ให้ p เป็นจำนวนเฉพาะ และให้ n เป็นจำนวนเต็มที่มากกว่า 1 แล้วเรียกฟีลด์ด้วยจำนวนสมาชิก p^n เขียนแทนด้วย $GF(p^n)$ ว่า *กาลัวส์ฟีลด์ (Galois field)*

2.5 เส้นโค้งเชิงวงรี

ในหัวข้อนี้แนะนำพื้นฐานโครงสร้างเชิงพีชคณิตของเส้นโค้งเชิงวงรีเหนือฟีลด์จำกัด พอสังเขป สำหรับราย- ละเอียดเชิงลึกสามารถดูได้ในงานของ Hankerson D Menezes A and Vanstone SA [10]

บทนิยาม 2.14 [10] *เส้นโค้งเชิงวงรี E เหนือฟีลด์ K (elliptic cirve E over field K)* ถูกกำหนดโดยสมการ

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{2.4}$$

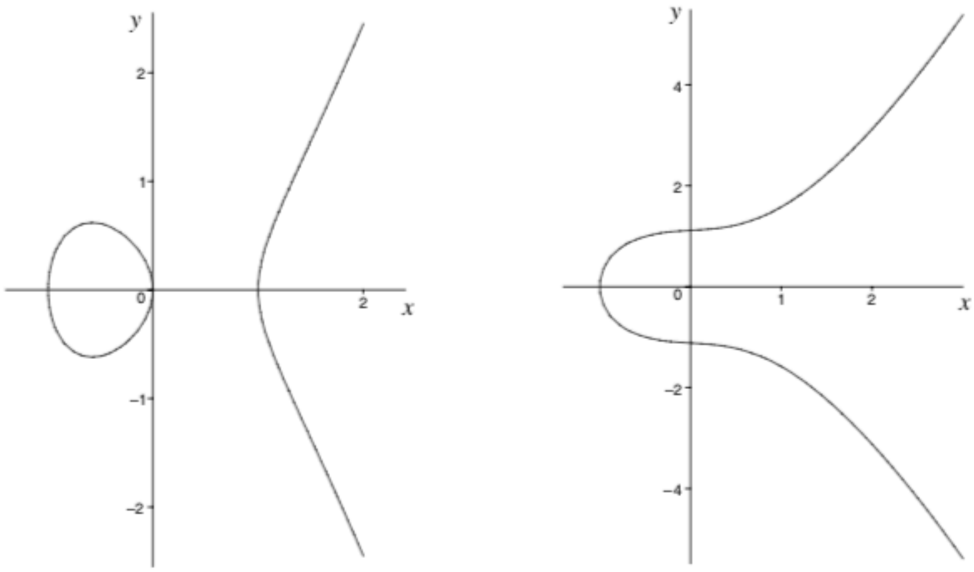
เมื่อ $a_1, a_2, a_3, a_4, a_5, a_6 \in K$ และ $\Delta \neq 0$ โดยที่ Δ คือ *ดิสคริมิแนนต์ (Discriminant)* ของ E และถูกกำหนดโดยสมการดังต่อไปนี้

$$\left.\begin{aligned} \Delta &= -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\ d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \end{aligned}\right\} \tag{2.5}$$

ถ้า L ฟีลด์ภาคขยาย (extension field) ของ K แล้ว เซทของจุด K บนเส้นโค้ง E คือ

$$E(L) = \{(\mathrm{x}, \mathrm{y}) \in \mathrm{L} \times \mathrm{L} : \ y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0\} \cup \{\infty\}$$

เมื่อ ∞ คือจุด ณ อนันต์



(a) $E_1 : y^2 = x^3 - x$ (b) $E_2 : y^2 = x^3 + \frac{1}{4}x + \frac{5}{4}$

รูปที่ 2.1 เส้นโค้งเชิงวงรีเหนือ $\mathbb{R}$

ข้อสังเกต 2.1

1. สมการ (2.4) เรียกว่า สมการไวแยร์สตราสส์ (Weierstrass equation)
2. เรากล่าวว่า E เหนือ K เพราะว่าค่าสัมประสิทธิ์ $\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4, \mathrm{a}_5, \mathrm{a}_6 \in \mathrm{K}$ บางครั้งสามารถเขียนแทนด้วย E/K เพื่อระบุว่า E เหนือ K และ K จะถูกเรียกว่า *ฟิลด์พื้นฐาน (underlying field)* และถ้าหาก E อยู่เหนือ K แล้ว E จะถูกเรียกว่า *ฟิลด์ภาคขยาย (extension field)* ของ K
3. ถ้าหากเงื่อนไข $\Delta \neq 0$ เป็นจริงแล้ว เราจะได้ว่า เส้นโค้งเชิงวงรีนั้นเรียบ หมายความว่าเส้นโค้งเชิงวงรีเส้นนั้นจะไม่มีจุดใดที่มีเส้นสัมผัสมากกว่า 1 เส้น
4. จุด ∞ เป็นจุดเดียวบนเส้น ณ อนันต์ที่สอดคล้องกับภาพฉายของสมการ ไวแยร์สตราสส์ (2.4)
5. จุด L บน E คือจุด (x, y) ที่สอดคล้องกับสมการเส้นโค้งที่พิกัด x และ y เป็นสมาชิกของ L และจุด ณ อนันต์เป็นจุด L ในทุกๆฟิลด์ภาคขยายของ L ใน K

ตัวอย่าง 2.5 (เส้นโคงเชิงวงรีบน $\mathbb{R}$) พิจารณาเส้นโค้งเชิงวงรี

$$\mathrm{E}_1 : \mathrm{y}^2 = \mathrm{x}^3 - \mathrm{x}$$
$$\mathrm{E}_2 : \mathrm{y}^2 = \mathrm{x}^3 + \frac{1}{4}\mathrm{x} + \frac{5}{4}$$

ที่ซึ่งอยู่เหนือฟีลด์ $\mathbb{R}$ ในเซทของจำนวนจริงแล้ว จุดบน $\mathrm{E}_1(\mathbb{R})\backslash\{\infty\}$ และ $\mathrm{E}_2(\mathbb{R})\backslash\{\infty\}$ จะมีลักษณะดังกราฟในรูปที่ 2.1

#

บทนิยาม 2.15 เส้นโค้งเชิงวงรีสองเส้น E_1 และ E_2 เหนือ K และถูกกำหนดโดยสมการไวแยร์สตราสส์

$$E_1 : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$
$$E_2 : y^2 + \bar{a}_1xy + \bar{a}_3y = x^3 + \bar{a}_2x^2 + \bar{a}_4x + \bar{a}_6$$

จะกล่าวว่าสมสัณฐานเหนือ K ถ้ามี $\mathrm{u}, \mathrm{r}, \mathrm{s}, \mathrm{t} \in \mathrm{K}, \mathrm{u} \neq 0$ ซึ่งทำให้การเปลี่ยนค่าของตัวแปร

$$(x, y) \rightarrow (u^2x + r, u^3y + u^2sx + t) \tag{2.6}$$

เปลี่ยนรูปของสมการ E_1 ให้อยู่ในรูปของ E_2 การเปลี่ยนแปลง (2.6) จะถูกเรียกว่า *ตัวแปรเปลี่ยนแปลงที่ยอมรับได้ (admissible change of variables)*

สมการไวแยร์สตราสส์

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

กำหนดเหนือ K สามารถทำให้อยู่ในรูปอย่างง่ายโดยการใช้ ตัวแปรเปลี่ยนแปลงที่ยอมรับได้ เราจะพิจารณาออกเป็นหลายกรณีโดย ฟีลด์พื้นฐาน K มีลักษณะเฉพาะไม่เท่ากับจาก 2 และ 3 หรือมีลักษณะเฉพาะเท่ากับ 2 หรือ 3

1. ถ้าลักษณะเฉพาะของ K มีเอกลักษณ์ไม่เท่ากับ 2 หรือ 3 แล้วตัวแปรเปลี่ยนที่ยอมรับได้

$$(x, y) \rightarrow \left(\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} - \frac{a_1^3 + 4a_1a_2 - 12a_3}{24}\right)$$

เปลี่ยน E เป็นเส้นโค้ง

$$y^2 = x^3 + ax + b \qquad (2.7)$$

เมื่อ $\mathrm{a}, \mathrm{b} \in \mathrm{K}$ ดิสคริมิแนนต์ของเส้นโค้งนี้จะมีค่าเท่ากับ $\Delta = -16(4\mathrm{a}^3 + 27\mathrm{b}^2)$

2. ถ้าเอกลักษณ์ของ K เป็น 2 แล้วจะต้องพิจารณา 2 กรณี ถ้าหาก $a_1 \neq 0$ แล้วจะได้ตัวแปรเปลี่ยนแปลงที่ยอมรับได้

$$(x, y) \to \left(a_1^2 x + \frac{a_3}{a_1}, a_1^3 y + \frac{a_1^2 a_4 - a_3^2}{a_1^3}\right)$$

เปลี่ยน E เป็นเส้นโค้ง

$$y^2 + xy = x^3 + ax^2 + b$$

เมื่อ $a, b \in K$ เราจะกล่าวว่าเส้นโค้งนี้ ไม่เป็นสภาวะเอกฐานยิ่งยวด (non-supersingular) และมีดิสคริมิแนนต์ $\Delta = b$

กรณีที่ 2 ถ้า $a_1 = 0$ แล้วจะได้ตัวแปรเปลี่ยนแปลงที่ยอมรับได้

$$(x, y) \to (x + a_2, y)$$

เปลี่ยน E เป็นเส้นโค้ง

$$y^2 + cy = x^3 + ax + b$$

เมื่อ $a, b, c \in K$เราจะกล่าวว่าเส้นโค้งนี้ เป็นสภาวะเอกฐานยิ่งยวด (supersingular) และมีดิสคริมิแนนต์ $\Delta = c^4$

2.5.1 กฎของกรุป

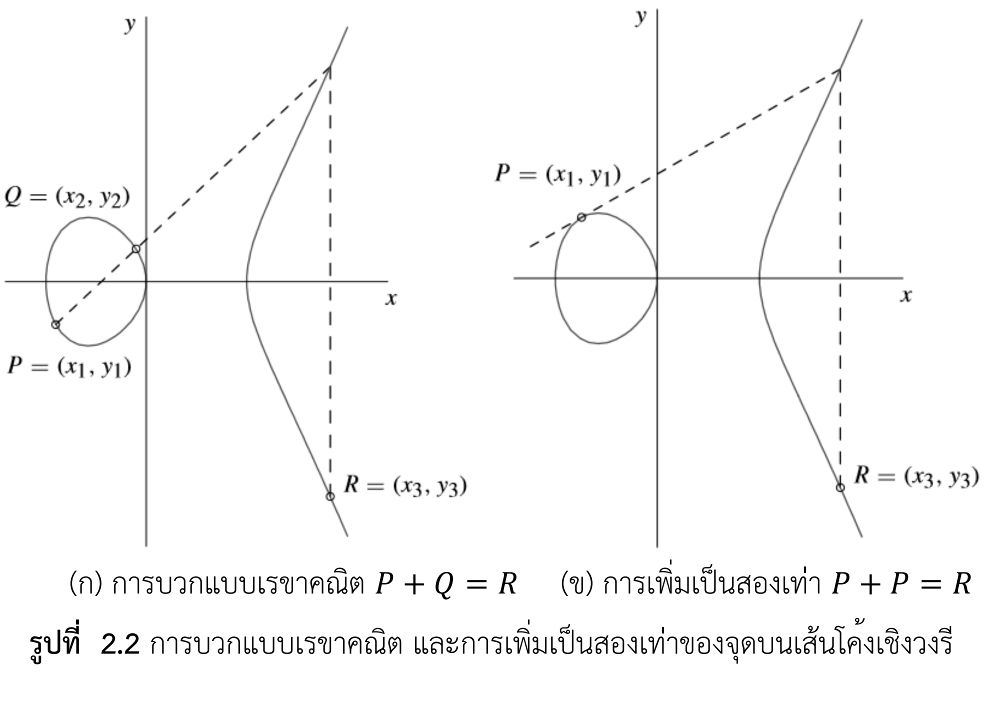
[10] กำหนดให้ E เป็นเส้นโค้งเชิงวงรีเหนือฟิลด์ K ซึ่งมี *กฎของคอร์ดและเส้นสัมผัส (chord-and-tangent rule)* สำหรับการบวกกันของ 2 จุดใน $E(K)$ เพื่อให้ได้จุดที่ 3 ใน $E(K)$

ด้วยการตัวดำเนินการบวก เซตของจุดใน $E(K)$ เป็น *อาบีเลียนกรุป (abelian group)* นั่นคือมีสมบัติการสลับที่ด้วยการมี ∞ เป็นเอกลักษณ์ กรุปนี้ใช้สำหรับในการสร้างเส้นโค้งเชิงวงรีของระบบวิทยาการเข้ารหัสลับ

กฎการบวกสามารถอธิบายด้วยเรขาคณิตโดยกำหนดให้ $P = (x_1, y_1)$ และ $Q = (x_2, y_2)$ เป็น 2 จุดที่เป็นอิสระต่อกันบนเส้นโค้งเชิงวงรี E แล้วผลรวมของ P และ Q คือ R สามารถนิยามได้ดังต่อไปนี้ ขั้นแรกเริ่มวาดเส้นตรงผ่านจุด P และ Q โดยเส้นตรงนี้จะสร้างจุดตัดบนเส้นโค้งเชิงวงรีเป็นจุดที่ 3 จะได้ว่า R เกิดจากภาพฉายบนแกนระนาบ x แสดงดังรูปที่ 2.2 (ก)

สองเท่าของจุด P คือ R สามารถนิยามได้ดังต่อไปนี้ ขั้นแรกลากเส้นสัมผัสบนเส้นโค้งเชิงวงรี ณ จุด P เส้นตรงนี้จะทำให้เกิดจุดตัดบนเส้นโค้งเชิงวงรี ณ จุดที่ 2 จะได้ว่า R เกิดจากภาพฉายของจุดที่ 2 บนระนาบแกน x ดังรูปที่ 2.2 (ข)

สมการพีชคณิต (algebraic formulas) สำหรับกฎของกรุปสามารถอนุพัทธ์จากการอธิบายทางเรขาคณิต สูตรเหล่านี้ จะใช้สำหรับเส้นโค้งเชิงวงรี E ของไวแยร์สตราสส์ในรูปแบบอย่างง่าย (2.5) ใน *โคออร์ดิเนตสัมพรรค (affine coordinates)* ของฟีลด์อ้างอิง K ที่ลักษณะเฉพาะไม่เท่ากับ 2 หรือ 3 เช่น $K = \mathbb{F}_p$ ที่ซึ่ง $P > 3$ เป็นจำนวนเฉพาะ เป็นต้น สำหรับเส้นโค้งเชิงวงรี E ที่รูปแบบไม่เป็นสภาวะเอกฐานยิ่งยวดของรูปแบบ (2.8) เหนือ $K = \mathbb{F}_{2^m}$ และสำหรับเส้นโค้งเชิงวงรี E ที่รูปแบบไม่เป็นสภาวะเอกฐานยิ่งยวดของรูปแบบ (2.9) เหนือ $K = \mathbb{F}_{2^m}$



(ก) การบวกแบบเรขาคณิต $P + Q = R$ (ข) การเพิ่มเป็นสองเท่า $P + P = R$

รูปที่ 2.2 การบวกแบบเรขาคณิต และการเพิ่มเป็นสองเท่าของจุดบนเส้นโค้งเชิงวงรี

2.5.1.1 กฎของกรุปสำหรับ $\boldsymbol{E/K}$: $\boldsymbol{y^2 = x^3 + ax + b, char(K) \neq 2, 3}$

1. เอกลักษณ์, $P + \infty = \infty + P = P$ สำหรับทุก $P \in E(K)$

2. ค่าลบ, ถ้า $P = (x, y) \in E(K)$ แล้ว $(x, y) + (x, -y) = \infty$ จุด $(x, -y)$ เขียนแทนด้วย $-P$ และจะเรียกว่าค่าลบของ P ซึ่ง $-P$ เป็นจุดจริงใน $E(K)$ รวมไปถึง $-\infty = \infty$ ด้วย

3. จุดจากการบวก, ให้ $P = (x_1, y_1) \in E(K)$ และ $Q = (x_2, y_2) \in E(K)$ ที่ซึ่ง $P \neq \pm Q$ แล้ว $P + Q = (x_3, y_3)$ เมื่อ

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \quad \text{และ} \quad y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

4. ค่าสองเท่า, ให้ $P = (x_1, y_1) \in E(K)$ ที่ซึ่ง $P \neq -P$ แล้ว $2P = (x_3, y_3)$ เมื่อ

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \quad \text{และ} \quad y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1$$

ตัวอย่าง 2.6 [10] เส้นโค้งเชิงวงรีเหนือฟีลด์จำนวนเฉพาะ $\mathbb{F}_{29}$ ให้ $p = 29, a = 4$ และ $b = 20$ พิจารณาเส้นโค้งเชิงวงรี

$$E: \; y^2 = \; x^3 + 4x + 20$$

ซึ่งถูกกำหนดเหนือ $\mathbb{F}_{29}$ และ $\Delta = -16\,(4a^3 + 27b^2) = -176896 \not\equiv 0\ (mod\ 29)$ ดังนั้น E เป็นเส้นโค้งเชิงวงรีจริง จุดที่อยู่ใน $E(29)$ มีดังนี้

∞	(2,6)	(4,19)	(8,10)	(13,23)	(16,2)	(19,16)	(27,2)
(0,7)	(2,23)	(5,7)	(8,19)	(14,6)	(16,27)	(20,3)	(27,27)

(0,22)	(3,1)	(5,22)	(10,4)	(14,23)	(17,10)	(20,26)
(1,5)	(3,28)	(6,12)	(10,25)	(15,2)	(17,19)	(24,7)
(1,24)	(4,10)	(6,17)	(13,6)	(15,27)	(19,13)	(24,22)

ตัวอย่างของการบวกกันของเส้นโค้งเชิงวงรี คือ $(5,22)+(16,27)=(13,6)$ และ $2(5,22)=(14,6)$

#

2.5.1.2 กฎของกรุปสำหรับรูปแบบไม่เป็นสภาวะเอกฐานยิ่งยวด $E/\mathbb{F}_{2^m}: y^2+xy=x^3+ax^2+b$

1. เอกลักษณ์, $P+\infty=\infty+P=P$ สำหรับทุก $P\in E(\mathbb{F}_{2^m})$

2. ค่าลบ, ถ้า $P=(x,y)\in E(\mathbb{F}_{2^m})$ แล้ว $(x,y)+(x,x+y)=\infty$ จุด $(x,x+y)$ เขียนแทนด้วย $-P$ และจะเรียกว่าค่าลบของ P ซึ่ง $-P$ เป็นจุดจริงใน $E(\mathbb{F}_{2^m})$ รวมไปถึง $-\infty=\infty$ ด้วย

3. จุดจากการบวก, ให้ $P=(x_1,y_1)\in E(\mathbb{F}_{2^m})$ และ $Q=(x_2,y_2)\in E(\mathbb{F}_{2^m})$ ที่ซึ่ง $P\neq\pm Q$ แล้ว $P+Q=(x_3,y_3)$ เมื่อ

$$x_3=\lambda^2+\lambda+x_1+x_2+a \quad \text{และ} \quad y_3=\lambda(x_1+x_3)+x_3+y_1$$

ด้วย $\lambda=(y_1+y_2)/(x_1+x_2)$

4. จุดจากการเพิ่มเป็นสองเท่า, ให้ $P=(x_1,y_1)\in E(\mathbb{F}_{2^m})$ ที่ซึ่ง $P\neq -P$ แล้ว $2P=(x_3,y_3)$ เมื่อ

$$x_3=\lambda^2+\lambda+a=x_1^2+\frac{b}{x_1^2} \quad \text{และ} \quad y_3=x_1^2+\lambda x_3+x_3$$

ด้วย $\lambda=x_1+y_1/x_1$

ตัวอย่าง 2.7 [10] เส้นโค้งเชิงวงรีเหนือ $\mathbb{F}_{2^4}$ แบบไม่เป็นสภาวะเอกฐานยิ่งยวด พิจารณาฟีลด์จำกัด $\mathbb{F}_{2^4}$ ด้วยฟังชันการลดทอนพหุนาม $f(z)=z^4+z+1$ และสมาชิกของ $a_3z^3+a_2z^2+a_1z+a_0\in\mathbb{F}_{2^4}$ แทนด้วย *บิตสตริง (bit string)* $(a_3a_2a_1a_0)$ ความยาว 4 เช่น (0101) แทนด้วย z^2+1 ให้ $a=z^3, b=z^3+1$ และพิจารณาเส้นโค้งเชิงวงรีแบบไม่เป็นสภาวะเอกฐานยิ่งยวด

$$E:\ y^2+xy=\ x^3+z^3x^2+(z^3+1)$$

ซึ่งถูกกำหนดเหนือ $\mathbb{F}_{2^4}$ และมีจุดใน $E(\mathbb{F}_{2^4})$ ดังนี้

∞	(0011,1100)	(1000,0001)	(1100,0000)
(0000,1011)	(0011,1111)	(1000,1001)	(1100,1100)
(0001,0000)	(0101,0000)	(1001,0110)	(1111,0100)
(0001,0001)	(0101,0101)	(1001,1111)	(1111,1011)
(0010,1101)	(0111,1011)	(1011,0010)	
(0010,1111)	(0111,1100)	(1011,1001)	

ตัวอย่างของการบวกกันของจุดบนเส้นโค้งเชิงวงรีสองจุด คือ $(0010{,}1111) + (1100{,}1100) = (0001{,}0001)$ และ $2(0010{,}1111) = (1011{,}0010)$

#

2.5.1.3 กฎของกรุปสำหรับรูปแบบเป็นสภาวะเอกฐานยิ่งยวด $\boldsymbol{E/\mathbb{F}_{2^m}: y^2 + cy = x^3 + ax + b}$

1. เอกลักษณ์, $P + \infty = \infty + P = P$ สำหรับทุก $P \in E(\mathbb{F}_{2^m})$

2. ค่าลบ, ถ้า $P = (x, y) \in E(\mathbb{F}_{2^m})$ แล้ว $(x, y) + (x, y + c) = \infty$ จุด $(x, y + c)$ เขียนแทนด้วย $-P$ และจะเรียกว่าค่าลบของ P ซึ่ง $-P$ เป็นจุดจริงใน $E(\mathbb{F}_{2^m})$ รวมไปถึง $-\infty = \infty$ ด้วย

3. จุดจากการบวก, ให้ $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$ และ $Q = (x_2, y_2) \in E(\mathbb{F}_{2^m})$ ที่ซึ่ง $P \neq \pm Q$ แล้ว $P + Q = (x_3, y_3)$ เมื่อ

$$x_3 = \left(\frac{y_2+y_1}{x_2+x_1}\right)^2 + x_1 + x_2 \quad \text{และ} \quad y_3 = \left(\frac{y_2+y_1}{x_2+x_1}\right)(x_1 + x_3) + y_1 + c$$

4. จุดจากการเพิ่มเป็นสองเท่า, ให้ $P = (x_1, y_1) \in E(\mathbb{F}_{2^m})$ ที่ซึ่ง $P \neq -P$ แล้ว $2P = (x_3, y_3)$ เมื่อ

$$x_3 = \left(\frac{x_1^2+a}{c}\right)^2 \quad \text{และ} \quad y_3 = \left(\frac{x_1^2+a}{c}\right)(x_1 + x_3) + y_1 + c$$

สำหรับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองที่ใช้ในการศึกษานั้น จะใช้เส้นโค้งคอบลิทซ์ (Koblitz) ดังบทนิยาม

บทนิยาม 2.16 [7] *เส้นโค้งคอบลิทซ์* คือ เส้นโค้งเชิงวงรีนิยามบนฟิลด์ F_2 ด้วยสมการไวแยร์สตราสส์ทั่วไปดังสมการ

$$E_A: Y^2 + XY = X^3 + AX^2 + 1 \tag{2.8}$$

ด้วย $A \in 0,1$ และดิสคริมิแนนต์ของ E_A เขียนแทนด้วย Δ_{E_A} มีค่าเป็น 1 ซึ่ง $\Delta_{E_A} \neq 0$ จึงไม่มีรากซ้ำกัน

2.6 วิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

ในหัวข้อนี้แนะนำเกี่ยวกับวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง โดยใน การศึกษานี้เลือกใช้เส้นโค้งคอบลิทซ์ เพราะมีจุดเด่นคือ สามารถใช้การส่งโฟรเบนิอุส (Frobenius map) เพื่อช่วย ในการเพิ่มความเร็วในการคำนวณ โดยรายละเอียดสามารถดูได้ใน [9]

เริ่มต้นด้วยผู้เข้ารหัสลับคือ อลิซ และผู้ถอดรหัสลับคือ บ็อบ ตกลงเลือกใช้เส้นโค้งคอบลิทซ์เหนือฟิลด์จำ- กัด $\mathbb{F}_{2^k}$ เขียนแทนด้วย E_A และ เลือกจุด $P \in E_A$ ที่ซึ่ง $\langle P \rangle = E_A$ แล้วทำการเปิดเผยไว้ ขั้นตอนการ สร้างกุญแจเริ่มจาก อลิซสุ่มจำนวนเต็มบวก $n_A < \#E_A$ ที่ต้องรักษาเป็นความลับ เรียกว่า กุญแจส่วนบุคคล แล้วได้กุญแจสาธารณะที่จะเผยแพร่เป็น $Q_A = n_A \cdot P \in E_A$ ส่วนขั้นตอนเข้ารหัสลับนั้น เริ่มจากบ็อบเลือกเพลนเท็กซ์ $M_E \in E_A$ ตามด้วยการสุ่มจำนวนเต็มบวก $k < \#E_A$ สำหรับใช้ชั่วคราว จากนั้นใช้กุญแจสาธารณะ ของอลิซคือ Q_A คำนวณหา $C_1 = k \cdot P \in E_A$ และ $C_2 = M_E + k \cdot Q_A \in E_A$ แล้วได้ไซเฟอร์เท็กซ์ (C_1, C_2) สำหรับส่งผ่านช่องสัญญาณให้อลิซ ส่วนขั้นตอนการถอดรหัสลับนั้น อลิซใช้กุญแจส่วนบุคคล n_A มาคำนวณหา ค่า $C_2 - n_A \cdot C_1 \in E_A$ ซึ่งได้ค่าเป็นเพลนเท็กซ์ M_E สำหรับภาพรวมของการทำงานของแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี สามารถแสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 โครงสร้างการออกแบบระบบ การเข้ารหัส และการถอดรหัส

การสร้างตัวแปรเสริมสาธารณะ	
คู่สนทนาสุ่มเลือกเส้นโค้งเชิงวงรี E_A เหนือ $\mathbb{F}_{2^k}$ และจุด P บนเส้นโค้งเชิงวงรีใน E_A หนึ่งจุด	
อลิซ	บ็อบ
การสร้างกุญแจ	
เลือกกุญแจลับ n_A คำนวณ $Q_A = n_A \cdot P$ ใน E_A เผยแพร่กุญแจสาธารณะ Q_A	
การเข้ารหัส	
	เลือกเพลนเท็กซ์ $M_E \in E_A$ เลือก k เป็นกุญแจชั่วคราว ใช้ กุญแจสาธารณะ อลิซ Q_A ไปที่ คำนวณ $C_1 = k \cdot P \in E_A$ และ $C_2 = M_E + k \cdot Q_A \in E_A$ ส่งข้อความที่เข้ารหัส (C_1, C_2) ไปที่อลิซ
การถอดรหัส	
คำนวณ $C_2 - n_A \cdot C_1 \in E_A$ ซึ่งเป็นค่าเดียวกันกับ M_E	

2.7 แลตทิช

ในหัวข้อนี้แนะนำพื้นฐาน*แลตทิช (Lattices)* ใน*ปริภูมิเวกเตอร์ (Vector spaces)* ด้วยบทนิยามที่เรียบเรียง จากงานของ Trappe W and Washington LC [8] เมื่อ $\mathbb{R}$ แทนเซตของจำนวนจริง $\mathbb{Z}$ แทนเซตของจำนวนเต็ม และ n เป็นจำนวนเต็มบวก

บทนิยาม 2.17 ให้ $\mathbb{R}^n$ เป็น $\mathbb{R}-$ ปริภูมิเวกเตอร์ แบบ $n-$ มิติ ให้ $B = \{v_1, \dots, v_n\}$ เป็น $\mathbb{R}-$ ฐานหลัก ($\mathbb{R}-$ basis) สำหรับ $\mathbb{R}^n$ แล้ว *แลตทิซก่อกำเนิดด้วย* B เขียนแทนด้วย $\mathcal{L}$ ที่ซึ่งกำหนดโดย

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} m_i \mathbf{v}_i | m_i \epsilon\ \mathbb{Z}, \mathbf{v}_i\ \epsilon\ B \right\}$$

สังเกตว่า $\mathbb{R}$ เป็นฟีลด์ และ $\mathbb{Z}$ เป็นริง ที่ซึ่ง $\mathbb{Z} \subset \mathbb{R}$ บางครั้งเรียก B ว่า *ฐานหลักของแลตทิซ (Basis of the lattices)*

ตัวอย่าง 2.8 ให้ $\mathbf{v}_1 = (1,0)$ และ $\mathbf{v}_2 = (0,1)$ ให้ $B = \{\mathbf{v}_1 + \mathbf{v}_2\}$ แล้ว $\mathcal{L} = \{m_1\mathbf{v}_1 + m_2\mathbf{v}_2\}$ นอกจากนี้ยังมี $\{(1,0),(3,1)\}$ และ $\{(2,1),(5,3)\}$ ต่างก็เป็นฐานหลักของแลตทิซ เพราะ

$$\det\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} = 1 \text{ และ } \det\begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix} = 1$$

ข้อสังเกต 2.2

1. ถ้า $\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}$ มีค่า*ดีเทอร์มิแนนต์ (Determinant)* เป็น ± 1 แล้ว $\{\mathbf{v}_1, \mathbf{v}_2\}$ เป็นฐานหลักของแลตทิซ [8]
2. แลตทิซจำนวนเต็มเรียกว่า *แลตทิซตัวกำหนดเป็นหนึ่ง (Unimodular lattice)* ถ้ามีค่าดีเทอร์มิแนนต์เป็น ± 1 [9]

ความยาวของเวกเตอร์ $\mathbf{v} = (x_1, \dots, x_\mathrm{n})$ เขียนแทนด้วย $\|\mathbf{v}\|$ นิยามเป็น

$$\|\mathbf{v}\| = (x_1^2 + \cdots + x_n^2)^{1/2}$$

บทนิยาม 2.18 [10] กำหนดฐานหลักของแลตทิซ $\mathrm{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_\mathrm{n}\}$ แล้วเรียกปัญหาการหาเวกเตอร์สั้นสุดใน แลตทิซ $\mathcal{L}$ ว่า *ปัญหาการหาเวกเตอร์สั้นสุด (SVP: Shortest Vector Problem)*

ข้อสังเกต 2.3 ปัญหา SVP จัดเป็นเอ็นพีฮาร์ด (NP-hard) [11]

2.8 ริงพหุนามสังวัตนาการ

ในหัวข้อนี้แนะนำพื้นฐานของ *ริงพหุนามสังวัตนาการ (Convolution polynomial ring)* ซึ่งเป็น *ริงผลหาร พหุนาม (Polynomial quotient rings)* ชนิดหนึ่งที่ถูกนำมาใช้ในระบบวิทยาการเข้ารหัสลับ NTRU

บทนิยาม 2.19. [1] ให้ N แทนจำนวนเต็มบวกที่ถูกตรึงค่าไว้ และ q เป็นจำนวนเต็มบวก แล้ว *ริงพหุนามสังวัต นาการลำดับที่ N (Convolution polynomial ring of rank N)* หรือเรียกแบบย่อว่า *ริงพหุนามสังวัตนาการ* หมาย ถึง *ริงผลหาร (Quotient rings)*

$$R = \mathbb{Z}[x]/(x^N - 1)$$

และ *ริงพหุนามสังวัตนาการมอดุโล q* หมายถึงริงผลหาร

$$R = (\mathbb{Z}/q\mathbb{Z})[x]/(x^N - 1)$$

ข้อสังเกต 2.4 ผลคูณของสองพหุนาม $a(x), b(x) \in R$ สามารถหาได้จากสูตร

$a(x) \star\ b(x) = c(x)$ ด้วย $c_k = \sum_{i+j\equiv k (mod\ N)} a_i b_{k-i}$

ตัวอย่าง 2.9 [1] ให้ $N = 5$ และ $a(x), b(x) \in R$ เป็นพหุนามที่ซึ่ง

$a(x) = 1 - 2x + 4x^3 - x^4$ และ $b(x) = 3 + 4x - 2x^2 + 5x^3 + 2x^4$
แล้ว

$$\begin{aligned} a(x) \star\ b(x) &= 3 - 2x - 10x^2 + 21x^3 + 5x^4 - 16x^5 + 22x^6 + 3x^7 - 2x^8 \\ &= 3 - 2x - 10x^2 + 21x^3 + 5x^4 - 16 + 22x + 3x^2 - 2x^3 \\ &= -13 + 20x - 7x^2 + 19x^3 + 5x^4 \in R = \mathbb{Z}[x]/(x^5 - 1) \end{aligned}$$

ถ้าพิจารณาในริง R_{11} แล้ว

$$a(x) \star\ b(x) = 9 + 9x + 4x^2 + 8x^3 + 5x^4 \in R_{11} = (\mathbb{Z}/11\mathbb{Z})[x]/(x^5 - 1)$$

#

ข้อสังเกต 2.5. [1] การส่ง $R \to R_q$ เป็น *ริงสาทิสสัณฐาน (Homomorphism ring)*

ให้ N เป็นจำนวนเฉพาะที่ตรึงไว้ให้ p เป็นจำนวนเฉพาะและ q เป็นจำนวนเต็มบวก และให้ R, R_p, R_qเป็นริงพหุนามสังวัตนาการที่ซึ่ง

$$R = \mathbb{Z}[x]/(x^N - 1)\,, R_p = \ (\mathbb{Z}/p\mathbb{Z})[x]/(x^N - 1)\,, R_q = (\mathbb{Z}/q\mathbb{Z})[x]/(x^N - 1)$$

บทนิยาม 2.20. ให้ $a(x) \epsilon R_q$ แล้ว *เซ็นเตอร์ลิฟต์ (Centered lift) ของ* $a(x)$ *ไป* R หมายถึง *พหุนามหนึ่งเดียว (Unique polynomial)* $a'(x) \epsilon R$ ที่ซึ่ง

$$a'(x) \bmod q = a(x)$$

เมื่อสัมประสิทธิ์ของ $a'(x)$ เขียนแทนด้วย a'_iอยู่ในช่วง

$$-\frac{q}{2} < a'_i \leq \frac{q}{2}$$

ข้อสังเกต 2.6 [1] ผลบวกหรือผลคูณของเซ็นเตอร์ลิฟต์นั้น ไม่จำเป็นต้องเท่ากับเซ็นเตอร์ลิฟต์ของผลบวกหรือผลคูณ

ตัวอย่าง 2.10 [1] สำหรับ $N = 5$ และ $q = 7$ พิจารณาพหุนาม

$$a(x) = 5 + 3x + 6x^2 + 2x^3 + 4x^4 \in R_7$$

แล้วมีสัมประสิทธิ์ของเซ็นเตอร์ลิฟต์ $a(x)$ ที่เลือกจากเซต $\{-3, -2, \ldots, 2, 3\}$ ดังนั้น เซ็นเตอร์ลิฟต์ของ $a(x) = -2 + 3x + x^2 + 2x^3 - 3x^4 \in R$

ในทำนองเดียวกัน สำหรับพหุนาม

$$b(x) = 3 + 5x^2 - 6x^3 + 3x^4$$

มีเซ็นเตอร์ลิฟต์ของ $b(x) = 3 - 2x^2 + x^3 + 3x^4 \in R$

สังเกตว่า เซ็นเตอร์ลิฟต์ของ $a(x) \star$ เซ็นเตอร์ลิฟต์ของ $b(x) = 20x + 10x^2 - 11x^3 - 14x^4$ แต่เซ็นเตอร์ลิฟต์ของ $a(x) \star b(x) = -x + 3x^2 + 3x^3$

#

ประพจน์ 2.1 [1] ให้ q เป็นจำนวนเฉพาะ แล้ว $a(x) \in R_q$ มีตัวผกผันการคูณก็ต่อเมื่อ

$$\gcd(a(x), x^N - 1) = 1 \in (\mathbb{Z}/q\mathbb{Z})[x] \tag{2.9}$$

ถ้า (2.9) เป็นจริง แล้วการหา $a(x)^{-1} \in R_q$ สามารถทำได้โดยขั้นตอนวิธียุคลิดภาคขยาย (extended Euclidean algorithm) (ดู [2]) เพื่อหาพหุนาม $u(x), v(x) \in (\mathbb{Z}/q\mathbb{Z})[x]$ ที่ซึ่ง

$$a(x)u(x) + (x^N - 1)v(x) = 1$$

แล้ว $a(x)^1 = u(x) \in R_q$

ตัวอย่าง 2.11 [1] ให้ $N = 5$ และ $q = 2$ แล้วสามารถใช้ขั้นตอนวิธียุคลิดภาคขยายคำนวณหา $(1 + x + x^4)^{-1} \in R_2 = (\mathbb{Z}/2\mathbb{Z})[x]/(x^5 - 1)$ ได้ดังนี้

จัดรูปสมการคำนวณภายใต้ $(\mathbb{Z}/2\mathbb{Z})[x]$

$$(1 + x + x^4)u(x) + (x^5 - 1)v(x) = gcd(1 + x + x^4, x^5 - 1) = 1 \tag{2.10}$$

ใช้ขั้นตอนวิธียุคลิดภาคขยายจาก (2.10) ใช้ขั้นตอนวิธียุคลิดภาคขยายได้ $u(x) = x^3 + x^2 + 1 \in (\mathbb{Z}/2\mathbb{Z})[x]$ ดังนั้น

$$(1 + x + x^4)^{-1} = x^3 + x^2 + 1 \in R_2$$

#

2.9 วิทยาการรหัสลับ NTRU

ในหัวข้อนี้แนะนำวิทยาการรหัสลับกุญแจสาธารณะ NTRU ซึ่งทำงาเหนือริงพหุนามสังวัตนาการและความยากของปัญหาขึ้นอยู่กับ SVP ในแลตทิซ [1]

บทนิยาม 2.21 สำหรับ d_1, d_2 เป็นจำนวนเต็มบวกใดๆ ให้

$$\tau(d_1, d_2) = \quad a(x) \in R: \left\{ \begin{array}{l} a(x) \text{ มีสัมประสิทธิ์เป็น 1 จำนวน } d_1 \\ a(x) \text{ มีสัมประสิทธิ์เป็น -1 จำนวน } d_1 \\ a(x) \text{ มีสัมประสิทธิ์ที่เหลือเป็น 0} \end{array} \right\}$$

เรียก พหุนาม $a(x)\,\epsilon\,\tau(d_1, d_2)$ ว่า *พหุนามไตรภาค (Ternary polynomial)*

การทำงานของวิทยาการรหัสลับเอ็นทรูสามารถแสดงได้ดังตารางที่ 2.2 เริ่มต้นด้วยอลิซเลือกตัวแปรเสริมสาธารณะ (N, p, q, d) ที่ซึ่งทั้งสองค่า N และ p เป็นจำนวนเฉพาะ $\gcd(p, q) = \gcd(N, q) = 1$ และ $q > (6d + 1)p$ จากนั้นอลิซสุ่มเลือกพหุนาม $f(x)$ ที่ซึ่งมีตัวผกผันใน R_q และ R_p และสุ่มเลือก$g(x)$ เพื่อใช้ก่อกำเนิดกุญแจส่วนบุคคล

$$f(x) \in \tau(d + 1, d) \text{ และ } g(x) \in \tau(d, d) \tag{2.11}$$

และคำนวณหาตัวผกผัน

$$F_q(x) = f(x)^{-1} \in R_q \text{ และ } F_p(x) = f(x)^{-1} \in R_p \tag{2.12}$$

แล้วได้กุญแจส่วนบุคคลเป็น $(f(x), F_p(x))$ จากนั้นอลิซคำนวณหาพหุนาม $h(x) \in R_p$ เพื่อใช้เป็นกุญแจสาธารณะ

$$h(x) = F_q(x) \star g \in R_p \tag{2.13}$$

บ็อบต้องการส่งเพลนเท็กซ์ $m(x) \in R$ ด้วยสัมประสิทธิ์ระหว่าง $-\frac{1}{2}p$ และ $\frac{1}{2}p$ (หรือกล่าวว่าเพลนเท็กซ์ $m(x) \in R$ เป็นเซ็นเตอร์ลิฟต์ของพหุนามใน R_p) จะทำการสุ่มพหุนามใช้ครั้งเดียว $r(x) \in \tau(d, d)$ และคำนวณไซเฟอร์เท็กซ์

$$e(x) \equiv p \cdot h(x) \star r(x) + m(x) (mod\ q) \tag{2.14}$$

ตารางที่ 2.2 การเข้ารหัสแบบ NTRU ด้วยระบบกุญแจสาธารณะ

การสร้างตัวแปรเสริมสาธารณะ
อลิซเลือกตัวแปรเสริมสาธารณะ (N, p, q, d) ที่ซึ่ง ทั้ง N และ p เป็นจำนวนเฉพาะ $\gcd(p, q) = \gcd(N, q) = 1$ และ $q > (6d + 1)p$

อลิซ	บ็อบ

การสร้างกุญแจ	
เลือกกุญแจส่วนบุคคล $f(x) \in \tau(d + 1, d)$ ที่ซึ่งสามารถหาตัวผกผันใน R_q และ R_p ได้ เลือกกุญแจส่วนบุคคล $g(x) \in \tau(d, d)$ คำนวณ $F_q(x) = f(x)^{-1} \in R_q$ คำนวณ $F_p(x) = f(x)^{-1} \in R_p$ เผยแพร่กุญแจสาธารณะ $h(x) = F_q(x) \star g$	
การเข้ารหัส	
	เลือกเพลนเท็กซ์ $m(x) \in R_p$ สุ่มเลือก $r(x) \in \tau(d, d)$ ใช้กุญแจสาธารณะของอลิซ $h(x)$ เพื่อคำนวณไซเฟอร์เท็กซ์ $e(x) \equiv p \cdot h(x) \star r(x) + m(x)(mod\ q)$ ทำการส่งไซเฟอร์เท็กซ์ $e(x)$ ไปยังอลิซ
การถอดรหัส	
คำนวณ $a(x) \equiv f(x) \star e(x)(mod\ q)$ เซ็นเตอร์ลิฟต์ $a(x)$ จะได้ $b(x) \equiv F_p(x) \star a(x)(mod\ p)$ ได้เพลนเท็กซ์คือ $b(x)$	

หลังจากที่อลิซได้รับไซเฟอร์เท็กซ์ $e(x) \in R_q$ อลิซสามารถถอดรหัสลับโดยเริ่มต้นด้วยการคำนวณ

$$a(x) \equiv f(x) \star e(x)(mod\ q) \tag{2.15}$$

จากนั้นเซ็นเตอร์ลิฟต์ $a(x)$ จะได้สมาชิกของ R แล้วแปลงให้อยู่ใน R_p จากนั้นคำนวณ

$$b(x) \equiv F_p(x) \star a(x)(mod\ p) \tag{2.16}$$

สมมติว่าตัวแปรเสริมถูกต้องแล้วจะได้พหุนาม $b(x)$ เท่ากับเพลนเท็กซ์ $m(x)$

ประพจน์ 2.2 [1] ถ้าตัวแปรเสริมเอ็นทรู (N, p, q, d) ที่เลือกสอดคล้องกับอสมการ (2.17)

$$q > (6d + 1)p \tag{2.17}$$

แล้วพหุนาม $b(x)$ ซึ่งคำนวณโดยอลิซใน (2.16) จะเท่ากับเพลนเท็กซ์ $m(x)$ ของบ็อบ

ข้อสังเกต 2.7 [1]

1. ระบบวิทยาการเข้ารหัสลับแบบแลตทิซมีความเร็วสูงกว่าเมื่อเปรียบเทียบกับวิทยาการเข้ารหัสลับแบบวิยุตลอการิทึมและแบบการแยกตัวประกอบ ซึ่งระยะเวลาที่ใช้มากสุดจะใช้ N^2 การคูณและจะอยู่ในขั้นตอนคำนวณผลคูณสังวัตนาการของพหุนาม ในส่วนของการคำนวณ $r(x) \star h(x), f(x) \star e(x)$ และ $F_p(x) \star a$ เมื่อ $r(x), f(x)$ และ F_pเป็นพหุนามไตรภาค ดังนั้นการเข้ารหัสลับและการถอดรหัสลับในเอ็นทรูใช้ $O(N^2)$ ขั้น และแต่ละขั้นก็มีความเร็วสูงมาก

2. การหาผลเฉลยของกุญแจส่วนบุคคลจากกุญแจสาธารณะในระบบเอ็นทรูมีความสมมูล (equivalence) กับการหาผลเฉลยของ SVP ในบางคลาสของแลตทิซ นั่นคือ กำหนด $h(x)$ การหาผลเฉลยของพหุนามไตรภาค $f(x)$ และ $g(x)$ ที่ซึ่ง $f(x) \star h(x) \equiv g(x)(mod\ q)$ เป็น NP-hard

ตัวอย่าง 2.12 [1] กำหนดพารามิเตอร์สาธารณะ

$$(N, p, q, d) = (7,3,41,2)$$

ซึ่งสามารถถอดรหัสลับได้เพราะว่า $41 = q > (6d + 1)p = 39$

อลิซเลือก

$$f(x) = x^6 - x^4 + x^3 + x^2 - 1 \in \tau(3,2) \text{ และ } g(x) = x^6 + x^4 - x^2 - x \in \tau(2,2)$$

แล้วคำนวณหาฟังก์ชันผกผัน

$$F_q(x) = f(x)^{-1}\ mod\ q = 8x^6 + 26x^5 + 31x^4 + 21x^3 + 40x^2 + 2x + 37 \in R_q,$$
$$F_p(x) = f(x)^{-1}\ mod\ p = x^6 + 2x^5 + x^3 + x^2 + x + 1 \in R_p$$

และเก็บค่ากุญแจส่วนบุคคล $(f(x), F_p(x))$ จากนั้นคำนวณแล้วเผยแพร่กุญแจสาธารณะ

$$h(x) = F_q(x) \star g(x) = 20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30 \in R_q$$

บ็อบต้องการส่งสารถึงอลิซด้วย

$$m(x) = -x^5 + x^3 + x^2 - x + 1$$

และใช้กุญแจครั้งเดียว (ephemeral key)

$$r(x) = x^6 - x^5 + x - 1$$

บ็อบคำนวณและส่งถึงอลิซด้วยไซเฟอร์เท็กซ์

$$\begin{aligned} e(x) &\equiv pr(x) \star h(x) + m(x) \\ &\equiv 31x^6 + 19x^5 + 4x^4 + 2x^3 + 40x^2 + 3x + 25\ (mod\ q) \end{aligned}$$

อลิซถอดรหัสลับของบ็อบได้อย่างราบรื่นโดยเริ่มต้นคำนวณ

$$f(x) \star e(x) \equiv x^6 + 10x^5 + 33x^4 + 40x^3 + 40x^2 + x + 40\ (mod\ q) \quad (2.18)$$

จากนั้นทำการเซ็นเตอร์ลิฟต์ (2.18) มอดุโล q แล้วได้

$$a(x) = x^6 + 10x^5 - 8x^4 - x^3 - x^2 + x - 1 \in R$$

ท้ายสุด อลิซลดรูปเป็น $a(x)$ มอดุโล p และคำนวณหา

$$F_p(x) \star a(x) \equiv 2x^5 + x^3 + x^2 + 2x + 1\ (mod\ p) \quad (2.19)$$

และเซ็นเตอร์ลิฟต์ (2.19) มอดุโล p แล้วได้

$$m(x) = -x^5 + x^3 - x + 1$$

#

2.10โปรแกรมเซจ

ในหัวข้อนี้ขอแนะนำโปรแกรมเซจ (Sage) ซึ่งถูกพัฒนาโดย [18] เป็นโปรแกรมที่พัฒนามาเพื่อใช้ในการศึกษาเกี่ยวกับคณิตศาสตร์ซึ่งเป็นโปรแกรมโอเพนซอร์สซอฟต์แวร์และฟรี ผู้ใช้สามารถใช้โปรแกรมเซจในการพัฒนาโปรแกรมเพื่อศึกษาในด้านต่างๆ เช่น พีชคณิตพื้นฐาน แคลคูลัส ไปจนถึงหัวข้อที่มีความซับซ้อน เช่น ทฤษฏีจำนวน วิทยาการการเข้ารหัสลับ ทฤษฏีกรุป หรือ ทฤษฏีกราฟ คณิตศาสตร์เชิงการจัด (combinatorics) ฯลฯ

โปรแกรมเซจนั้น ถูกพัฒนาด้วยภาษาไพธอนซึ่งเป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมอย่างมากในปัจจุบัณ โปรแกรมเซจประกอบไปด้วยโปรแกรมโอเพนซอร์สสำเร็จรวมกันมากกว่า 100 โปรแกรม ผู้พัฒนาโปรแกรมเซจ ได้รับการระดมเงินทุนจากหลายหน่วยงานเพื่อพัฒนาโปรแกรมเซจ โดยลิขสิทธิ์ของโปรแกรมเซจนั้นเป็นของ GNU Public License (GPL)

จุดเด่นที่น่าสนใจของโปรแกรมเซจนั้นก็คือความเร็วในการประมวลผลที่เร็วกว่าโปรแกรมอื่น เช่น โปรแกรมเซจสามารถคำนวณการแยกตัวประกอบของ $\mathbf{2^{512} - 1}$ ได้ในเวลา 92.29 วินาที ในขณะที่โปรแกรม Mathematica 7 ใช้เวลาทั้งสิ้น 346.494 วินาที ซึ่งเป็นโปรแกรมที่ใช้ในการทำงานด้านคณิตศาสตร์เช่นเดียวกับโปรแกรมเซจ ความน่าเชื่อถือของโปรแกรมเซจนั้นสามารถดูได้จากจำนวนบทความที่ตีพิมพ์และมีการอ้างอิงถึงโปรแกรมเซจเป็นจำนวนมากกว่า 400 บทความ วิทยานิพนธ์มากกว่า 40 เล่ม และหนังสืออีกมากกว่า 40 เล่ม

ในการทำโปรเจคครั้งนี้ เราได้ใช้โปแกรมเซจเพื่อศึกษาวิทยาการรหัสลับแบบเอ็นทรูและวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี โดยจะทำการยกตัวอย่างการวิทยาการการเข้ารหัสลับแบบเอ็ลกามอลดังตารางที่ 2.2 ซึ่งพัฒนาด้วยโปรแกรมเซจโดยใช้กาลัวส์ฟิลด์ และเส้นโค้งเชิงวงรีในการเข้ารหัสลับ ในตัวอย่างนี้ประกอบไปด้วยส่วนขั้นตอนการสร้างตัวแปรเสริมสาธารณะและการสร้างกุญแจสาธารณะ

ขั้นตอนการสร้างตัวแปรเสริมสาธารณะ

```
# Create All Shared Varible Ea, P
l = 3
S.<V> = GF(2^l, "S")
Sgen = S.gen() #Genrator of FiniteField
Sord = S.order() #Order of FiniteField
a = randrange(0, Sord-1)
A = Sgen^a #Random A parameter for EllipticCurve
Ea = EllipticCurve(S, [1,A,0,0,1]) #Create EllipticCurve Over Finite Field 2^l
g = Ea.gen(0) # Random 1 point in EllipticCurve
Eord = len(list(Ea)) # Order of EllipticCurve
n = randrange(1, Eord-1)
P = n*g # Random P point in EllipticCurve
print("Public Varible P : " + str(P))
print("EllipticCurve Order : " + str(Eord))
print("Ea = " + str(Ea))
list(Ea)
```

ขั้นตอนการสร้างกุญแจสาธารณะ

```
# Create Alice public key Qa and private key na
na = randrange(1, Eord - 1)
Qa = na*P
print("Alice's Private Key : %d" % na)
print("Alice's Public Key : " + str(Qa))
```

2.11วรรณกรรมที่เกี่ยวข้อง

2.11.1 แผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง

ในงานศึกษาแผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง [9] ซึ่งเป็นงานที่ศึกษาเกี่ยวกับการออกแบบแผนวิธีการเข้ารหัสลับเพื่อให้ได้ความปลอดภัย มีประสิทธิภาพในการเข้ารหัสที่เร็ว และสามารถนำไปใช้ได้จริงโดยใช้วิธีการแบบอสมาตรในการส่งกุญแจและใช้เส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสองช่วยในการเข้ารหัส

ผลจากงานศึกษาแผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง [9] ได้อธิบายถึงขั้นตอนการเข้ารหัสด้วยอัลกอริทึมที่ได้ทำการออกแบบขึ้นมาโดยใช้วิทยาการการเข้ารหัสลับเอ็ลกามอลด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสองซึ่งมีการทำงานดังตาราง 2.2 รวมทั้งการทดสอบอัลกอริทึมที่ได้ทำการออกแบบ พบว่าอัลกอริทึมสามารถทำงานได้ตามวัตถุประสงค์ของการเข้ารหัสคือสามารถส่งข้อความไปยังผู้รับด้วยวิธีการแบบอสมมาตร และยังมีความปลอดภัยเนื่องผู้ที่ไม่มีกุญแจส่วนบุคคลจะถอดรหัสได้ยากเนื่องจากปัญหาในการค้นหากุญแจซึ่งอยู่บนเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง

2.11.2 วิทยาการเข้ารหัสลับด้วยพื้นฐานของแลตทิซเหนือแลตทิซมาตรฐานบนฮาร์ดแวร์

ในการศึกษางานวิจัยเกี่ยวกับวิทยาการเข้ารหัสลับด้วยพื้นฐานของแลตทิซ [19] เป็นงานวิจัยที่ศึกษาเกี่ยวกับการปรับวิทยาการรหัสลับบนพื้นฐานของแลตทิซเพื่อให้สามารถทำงานบนฮาร์ดแวร์ที่มีพื้นที่และหน่วยความจำที่จำกัดได้ ซึ่งฮาร์ดแวร์ที่ใช้ในงานชิ้นนี้คือ สปาตัน 6 เอฟพีจีเอ (Spartan 6 FPGA) ซึ่งสามารถพัฒนาวิทยาการรหัสลับบนฮาร์ดแวร์ดังกล่าวได้เป็นอย่างดี งานวิจัยชิ้นนี้เป็นงานวิจัยชิ้นแรกที่ได้มีการนำวิทยาการรหัสลับบนพื้นฐานของแลตทิซมาพัฒนาบนในแบบของสถาปัตยกรรมฮาร์ดแวร์ซึ่งจะเป็นตัวชี้วัดได้สำหรับการพัฒนาบนอุปกรณ์ฮาร์ดแวร์ในอนาคต

ในปัจจุบันวิทยาการรหัสลับบนพื้นฐานของแลตทิซกำลังเป็นที่ยอมรับเพื่อใช้แทนวิทยาการรหัสลับกุญแจสาธารณะในปัจจุบัณ เนื่องจากความสามารถในการป้องกันการโจมตีจากควอนตัมได้ ความสามารถที่หลากหลายและมีขนาดกุญแจที่เล็ก โดยใช้ปัญหาการเรียนรู้จากข้อผิดพลาด (learning with error) ซึ่งเป็นปัญหาที่พูดถึงการหากุญแจส่วนบุคคลเมื่อทราบกุญแจ

สาธารณะบนแลททิซ ถึงอย่างไรก็ตามการศึกษาและค้นคว้าอย่างละเอียดเกี่ยวกับจุดยืนของแลตทิซในการทำงานปกติยังจำเป็นต้องมีการพิจารณาอยู่ ถึงแม้ว่าจะมีประสิทธิภาพสูง

จากผลการศึกษาเปรียบเทียบกับฮาร์ดแวร์ที่เทียบเท่ากับพื้นฐานของริงบนการเรียนรู้จากข้อผิดพลาด (learning with error) พบว่า วิทยาการเข้ารหัสลับบนฮาร์ดแวร์มีประสิทธิภาพและความปลอดภัยที่สูง โดยสามารถเข้ารหัสได้ 1272 รหัสต่อวินาที และสามารถถอดรหัส 4395 รหัสต่อวินาที

2.11.3 การพัฒนาความปลอดภัยบนไอโอทีโดยใช้ขั้นตอนวิธีการเข้ารหัส

งานวิจัยเพื่อพัฒนาความปลอดภัยบนไอโอทีโดยใช้ขั้นตอนวิธีการเข้ารหัส [20] กล่าวถึงงานไอโอทีซึ่งในปัจจุบันมีความนิยมอย่างแพร่หลาย ทำให้เกิดการนำไอโอทีไปประยุกต์เข้ากับการอำนวยความสะดวกหรือเครื่องใช้มากมาย ด้วยเหตุนี้ความปลอดภัยจึงเป็นปัจจัยเบื้องต้นของการออกแบบไอโอทีโดยเฉพาะความน่าเชื่อถือในการส่งข้อมูล และกระบวนการที่ฉลาดทำให้ระบบความปลอดภัยมีความสำคัญ ในบทความได้กล่าวถึงการใช้ ขั้นตอนวิธีการเข้ารหัสแบบลูกผสม (hybrid encryption algorithm) ที่จะลดความเสี่ยงในด้านความปลอดภัย เพิ่มประสิทธิภาพด้านความเร็ว, และลดความซับซ้อนของการคำนวณ

จากที่กล่าวมาข้างต้นถึงความแพร่หลายของไอโอทีซึ่งในด้านผู้ประกอบการ และผู้บริโภคยังคงมีความเสี่ยงในเรื่องความปลอดภัยในการใช้ไอโอทีดังนั้นแล้วการเข้ารหัสข้อมูลนั้นเพื่อที่จะลดความเสี่ยงลง การเลือกขั้นตอนวิธีการเข้ารหัสที่เหมาะสมนั้นควรที่จะลดความเสี่ยง ชนิดของการเข้ารหัสสามารถที่จะเป็นการเข้ารหัสแบบกุญแจสาธารณะ มีความรวดเร็วสูงในการเข้ารหัส และใช้หน่วยความจำน้อย

2.11.3.1 ขั้นตอนและวิธีในการส่งข้อมูล

ในบทความจะนำเสนอถึงขั้นตอนวิธีการเข้ารหัสแบบลูกผสมที่มีคุณสมบัติพิเศษในการเข้ารหัส และถอดรหัสด้วยใช้เวลาเพียงเล็กน้อย โดยสามารถที่จะพัฒนาความปลอดภัยอินเทอร์เน็ตระหว่างการใช้ไอโอที และการใช้ลายเซ็นแบบดิจิตอล โดยผสมผสานขั้นตอนวิธีของ เอออีเอส (AES) และ เอ็นทรูจะถูกเรียกว่าขั้นตอนวิธีของเฮชเอเอ็น (HAN) มีขั้นตอนการทำงานระหว่างการส่งข้อมูลดังนี้

1. การสร้างกุญแจ

ในขั้นตอนแรกจะมีสองเมตริกซ์ขนาด 4×4 โดยทั้งสองใช้เพื่อสร้างกุญแจสำหรับเข้ารหัส โดยเราสามารถสุ่มตำแหน่งจากเมตริกซ์สถานะ (state matrix) และสุ่มกุญแจจากเมตริกซ์กุญแจ (key matrix) และสร้างกุญแจ h จากการดำเนินการ XOR

ในขั้นตอนนี้เฮชเอเอ็นได้ทำการดึงกระบวนการการทำงานมาจากขั้นตอนวิธีของเออีเอสในกระบวนการนี้เมื่อได้กุญแจสาธารณะมาแล้ว จะทำการเข้ารหัสข้อความที่ต้องการส่งโดยขั้นตอนวิธีการการเข้ารหัสแบบเอ็นทรูในขั้นตอนต่อไป

2. การเข้ารหัส

กำหนดให้ข้อความส่งจากผู้ส่งถึงผู้รับโดยข้อความเป็นอเนกนาม (multi-nomial) หลังจากนั้นผู้ส่งจะทำการสุ่มเลือกอเนกนามเช่น *r* จาก *Lr* โดยพึงระลึกว่าข้อความนี้จะถูกเปิดเผยโดยผู้ส่ง

$$\mathrm{Encryption} = pr \times h + \mathrm{message}$$

โดยข้อความนี้จะถูกส่งไปถึงผู้รับโดยเป็นข้อความที่มีความปลอดภัยแล้ว

3. การถอดรหัส

ในขั้นตอนนี้ผู้รับจะทำการถอดรหัสเพื่อดูข้อความโดยในขั้นตอนวิธีของเฮชเอเอสนั้นได้ใช้ขั้นตอนวิธีของเอ็นทรูโดยผู้รับจะมีสองกุญแจส่วนบุคคล f และ f_pที่ f_p เป็นส่วนกลับของ f ดังนั้นจะได้ว่า

$$\mathrm{Decryption} = \frac{f_p \times b}{x^2}$$

เมื่อถอดรหัสแล้วเราจึงจะพบข้อความที่ถูกส่งมา

ตารางที่ 2.3 เวลาที่ใช้ในการโจมตีและระดับบิตความปลอดภัยของวิทยาการรหัสลับเอ็นทรู

N	T (มิปส์)	ระดับบิตความปลอดภัย
167	3.21×10^{5}	57
251	5.07×10^{14}	88
400	1.05×10^{31}	142
500	9.33×10^{41}	178
600	8.31×10^{52}	214
800	6.59×10^{74}	287
1000	5.23×10^{96}	360

หลังจากนี้เมื่อทำการเปรียบเทียบประสิทธิภาพการทำงานของขั้นตอนวิธีการของเฮชเอเอส และเอ็นทรูแล้ว เอ็นทรู มีประสิทธิภาพในด้านความเร็ว ในการสร้างกุญแจ การเข้ารหัส และการถอดรหัส อีกทั้งยังเป็นที่ยอมรับในด้านความปลอดภัยสำหรับไอโอที

2.11.4 รายงานการประมาณเวลาในการโจมตีของวิทยาการรหัสลับเอ็นทรูแลททิซ

รายงานการประมาณเวลาในการโจมตีของวิทยาการรหัสลับเอ็นทรูแลททิซ [21] กล่าวถึงการประมาณค่าเวลาที่ใช้ในการถอดรหัสของวิทยาการรหัสลับเอ็นทรูบนแลททิซ ซึ่งเป็นการเข้ารหัสด้วยระบบกุญแจอสมมาตร ในการทดสอบนี้ใช้เครื่องคอมพิวเตอร์ซีพียู 400 MHz เซเลรอน บนระบบปฏิบัติการลินุกส์โดยโปรแกรมที่ใช้คือวิกเตอร์ ชูป

หลังจากที่ได้ทำการทดสอบแล้ว จะได้ค่าออกมาในหน่วยของมิปส์หลังจากนั้นจึงนำมาคำนวณตามความเร็วของซีพียูที่ใช้ จะทำให้เราสามารถรู้ระดับความปลอดภัยได้ดังตารางที่ 2.3

2.11.5 เปรียบเทียบระหว่างวิทยาการรหัสลับอาร์เอสเอและวิทยาการรหัสลับเส้นโค้งเชิงวงรี

ในงานเปรียบเทียบระหว่างวิทยาการรหัสลับอาร์เอสเอและวิทยาการรหัสลับเส้นโค้งเชิงวงรี [22] ได้พูดถึงการเปรียบเทียบประสิทธิภาพของวิทยาการรหัสลับอาร์เอสเอ ในแง่ของระดับความปลอดภัย ความเร็วและปริมาณแบนด์วิธท์ โดยส่วนที่เราจะนำมาใช้ในโครงงานศึกษานี้คือระดับความปลอดภัยของวิทยาการรหัสลับเส้นโค้งเชิงวงรีดังตารางที่ 2.4

ตารางที่ 2.4 ระดับบิตความปลอดภัยของวิทยาการรหัสลับเส้นโค้งเชิงวงรีเทียบกับขนาดกุญแจสาธารณะ

ระดับบิตความปลอดภัย	ขนาดกุญแจสาธารณะของวิทยาการรหัสลับเส้นโค้งเชิงวงรี (บิต)
80	160
112	224
128	256
192	384
256	512

ตารางที่ 2.5 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง

หัวข้อ/สิ่งที่ทำ	ออกแบบแผนวิธีรหัสลับเอ็ลกา-มอลเส้นโค้งเชิงวงรี	ประสิทธิ-ภาพของแผนวิธีรหัสลับเอ็ลกามอล	ออกแบบแผนวิธีเข้ารหัสเอ็นทรู	ประสิทธิ-ภาพแผนวิธีเข้ารหัสเอ็นทรู	ทรัพยากร
1. แผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง	✓	✓	✕	✕	เครื่องคอมพิวเตอร์
2. วิทยาการเข้ารหัสลับด้วยพื้นฐานของแลตทิซเหนือแลตทิซมาตรฐานบนฮาร์ดแวร์	✕	✕	✓	✓	บอร์ดเอฟพีจีเอ สปาร์ตัน 6

ตารางที่ 2.5 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง (ต่อ)

3. การพัฒนาความปลอดภัยบนไอโอทีโดยใช้ขั้นตอนวิธีการเข้ารหัส	✕	✕	✓	✓	อุปกรณ์ไอโอที
4. รายงานการประมาณเวลาในการโจมตีของวิทยาการรหัสลับเอ็นทรูแลททิซ	✕	✕	✕	✓	คอมพิวเตอร์
5. เปรียบเทียบระหว่างวิทยาการรหัสลับอาร์เอสเอและวิทยาการรหัสลับเส้นโค้งเชิงวงรี	✕	✓	✕	✕	ระบบสมองกลฝังตัว

บทที่ 3

การออกแบบ

ในบทนี้พัฒนาขั้นตอนวิธีสำหรับคำนวณเหนือริงสังวัฒนาการเพื่อให้สะดวกในการบูรณาการร่วมกับแผนวิธีวิทยาการเข้ารหัสลับเอ็นทรู ดังปรากฏในตารางที่ 2.2 เพื่อให้สามารถทำงานได้อย่างถูก โดยการคำนวณประกอบด้วยขั้นตอนวิธีต่อไปนี้คือ ขั้นตอนการสร้างตัวแปรเสริมสาธารณะ การสร้างกุญแจ การเข้ารหัสลับและการถอดรหัสลับ

3.1 การสร้างตัวแปรเสริมสาธารณะ

คู่สนทนาเลือกตัวแปรเสริมสาธารณะ (N, p, q, d) ด้วย N และ p เป็นจำนวนเฉพาะ ที่ซึ่ง $\gcd(p, q) = \gcd(N, q) = 1$ และ $q > (6d + 1)p$ (รายละเอียดดูใน [1]) แล้วสามารถออกแบบวิธีการคำนวณได้ดังขั้นตอนวิธีที่ 3.1

ขั้นตอนวิธี 3.1 การสร้างตัวแปรเสริมสาธารณะในวิทยาการเข้ารหัสลับเอ็นทรู

Input: เลขชี้กำลัง k สำหรับกำหนดค่าขอบการสุ่มตัวแปรเสริม

Output: (N, p, q, d) ที่ซึ่ง $\gcd(p, q) = \gcd(N, q) = 1$ และ $q > (6d + 1)p$

$flag = \mathrm{true}$

while $flag$ **do**

    $N = \text{next\_prime}(\text{randrange}(2^k, 2^{k+1}))$

    $p = \text{next\_prime}(\text{randrange}(2^k, 2^{k+1}))$

    $d = \text{randrange}(1, N)$

    $q = \text{next\_prime}(\text{randrange}((6d + 1)p, (6d + 1)p \cdot 2^5))$

    if $\gcd(p, q) == 1$ **then**

        if $\gcd(N, q) == 1$ **then**

9: **if** $q > (6d+1)p$ **then**

10: $flag = \text{false}$

11: **end if**

12: **end if**

13: **end if**

14: **end while**

ในขั้นตอนที่ 3.1 เริ่มต้นกำหนดค่า $flag$ เป็น true จากนั้นทำลูปโดยในบรรทัดที่ 3-4 จะสุ่มจำนวนเฉพาะ N และ p ที่มีค่าอยู่ระหว่าง 2^k ถึง 2^{k+1} ในบรรทัดที่ 5 สุ่มจำนวนเต็ม d ที่มีค่าระหว่าง 1 ถึง $N-1$ และในบรรทัดที่ 6 สุ่มจำนวนเฉพาะ q ที่มีค่ามากกว่า $(6d+1)p$ จากนั้นทำการตรวจตรวจสอบเงื่อนเพื่อให้ได้ตัวแปรสุ่มที่ซึ่ง $\gcd(p,q) = \gcd(N,q) = 1$ และ $q > (6d+1)p$ แล้วได้ค่าตัวแปรเสริม (N,p,q,d)

ตัวอย่างเซจ 3.1 จากขั้นตอนวิธีที่ 3.1 สามารถทำการสร้างตัวแปรเสริมสาธารณะ ด้วยคำสั่งดังต่อไปนี้

```
#---------------------------------------------------------
# Public parameter generator input = degree k
#---------------------------------------------------------
def parameter_gen(k):
    flag = True
    while flag :
        N = next_prime(randrange(2**k, 2**(k+1)))
        p = next_prime(randrange(2**k, 2**(k+1)))
        d = randrange(1, N)
        q = next_prime(randrange((6*d + 1)*p, (6*d +
1)*p*2**5))
        if gcd(p,q) == 1 :
            if gcd(N,q) == 1 :
                if q > (6*d + 1) * p :
                    flag = False
    return (N,p,q,d)
#---------------------------------------------------------
```

ผลรัน:

ให้ขนาดจำนวนเฉพาะเป็น $k = 3$ แล้วได้ผลรัน

ครั้งที่ 1:

```
k = 3
(N,p,q,d) = parameter_gen(k)
print (N,p,q,d)
```

```
(17, 11, 11437, 7)
```

ครั้งที่ 2:

```
k = 3
(N,p,q,d) = parameter_gen(k)
print (N,p,q,d)
```

```
(13, 17, 9817, 3)
```

จะเห็นได้ว่าตัวแปรเสริมสารธารณะทั้ง 4 ตัวแปรนั้นมีค่าเป็นไปตามเงื่อนไขของขั้นตอนวิธีการเข้ารหัสลับนั้นก็คือ $1024 < N < 2048$ และ $1024 < p < 2048$ โดย N และ p เป็นจำนวนเฉพาะและ $1 \leq d < 1237$ รวมทั้ง $q > (6d+1)p = 5372339$

#

3.2 การสร้างกุญแจ

อลิซเป็นผู้ก่อกำเนิดกุญแจจะทำการเลือกกุญแจส่วนบุคคลในรูปฟังก์ชัน $f(x)$ และ $g(x)$ ที่เป็นพหุนามไตรภาคในริง R แล้วคำนวณฟังก์ชันผกผันของ $f(x)$ ในริงสังวัตนาการ R_q ได้เป็น F_q และคำนวณฟังก์ชันผกผันของ $f(x)$ ในริงสังวัตนาการ R_p ได้เป็น F_p จากนั้นคำนวณหาฟังก์ชัน $h = f_q \star g$ ที่จะเผยแพร่เป็นกุญแจสาธารณะในริงสังวัตนาการ R_Q สำหรับกระบวนการคำนวนสามารถแสดงได้ดังขั้นตอนวิธีที่ 3.2

ขั้นตอนวิธีที่ 3.2 ในบรรทัดที่ 1 เราจะทำการสร้างริงพหุนามขึ้นมาและใช้ตัวชั่วคราวคือ x เราจะได้ริง R ขึ้นมา หลังจากนั้นทำการสร้างพหุนามไอดีลขึ้นมาซึ่งมีค่าเท่ากับ $id = x^N - 1$ เพื่อใช้ในก่อกำเนิดริง $\mathbb{Z}[x]/x^N - 1$ ในบรรทัดที่ 3 และ 4 เป็นการสร้างพหุนามไตรภาค f, g โดยที่ $f \in \tau(d+1, d)$ และ $g \in \tau(d, d)$ และ f, g จะมีดีกรีเท่ากับ N ซึ่งใช้ขั้นตอนวิธีที่ 3.3 ในการก่อกำเนิดพหุนามไตรภาค ในบรรทัดที่ 5 และ 6 จะเป็นการหาตัวผกผันของ f ในริง R_q และ f ใน

ริง R_p ซึ่งจะได้ผลลัพธ์คือ F_q และ F_p ในการหาตัวผกผันจะใช้ขั้นตอนวิธีที่ 3.4 ในบรรทัดที่ 7 เป็นส่วนที่ใช้ในการคำนวณหากุญแจสาธารณะ h โดยใช้ขั้นตอนวิธีที่ 3.6 ในการคำนวณ

ขั้นตอนวิธี 3.2 การก่อกำเนิดกุญแจสำหรับวิทยาการเข้ารหัสลับเอ็นทรู

Input: (N, p, q, d)

Output: กุญแจส่วนบุคคล $f(x)$ ในริงสังวัตนาการ $R, F_P(x)$ ในริงสังวัตนาการ R_p และกุญแจสาธารณะ $h(x)$ ในริงสังวัตนาการ R_q

1: $R.\langle x \rangle = \text{PolynomialRing}(ZZ)$

2: $id = x^N - 1$

3: $f = \text{tri\_poly}(d+1, d)$

4: $g = \text{tri\_poly}(d, d)$

5: $F_q = \text{fq\_inv}(f)$

6: $F_p = \text{fp\_inv}(f)$

7: $h = \text{find\_h}(F_q, g)$

ขั้นตอนวิธี 3.3 การก่อกำเนิดพหุนามไตรภาค

Input: (d_1, d_2)

Output: ฟังก์ชันพหุนาไตรภาค

1: $s = [1 \text{ for } j \text{ in range}(d_1 - 1)]$

2: $s = s + [-1 \text{ for } j \text{ in range}(d_2)]$

3: $s = s + [0 \text{ for } j \text{ in range}(N - d_2 - d_1 - 1)]$

4: $\text{random.shuffle}(s)$

5: $s.\text{append}(1)$

6: $\text{return } R(s)$

ในขั้นตอนวิธี 3.3 จะเป็นขั้นตอนในการก่อกำเนิดพหุนามไตรภาคซึ่งถูกเรียกมาจากขั้นตอนวิธีที่ 3.2 โดยจะรับค่าเข้ามา 2 พารามิเตอร์คือ (d_1, d_2) ซึ่งเป็นจำนวนของเลข 1 และ -1 ในพหุนามไตรภาค ในบรรทัดที่ 1 จะทำการสร้างลิสต์ของเลขจำนวนเต็ม 1 จำนวน $d_1 - 1$ แล้วเก็บลิสต์ไว้ในตัวแปล s ในบรรทัดที่ 2 จะทำการสร้างลิสต์ของเลขจำนวนเต็ม -1 จำนวน d_2 ตัวแล้วเพิ่มเข้าไปในลิสต์ s บรรทัดที่ 3 จะเป็นส่วนที่สร้างลิสต์ของเลขจำนวนเต็ม 0 ทั้งหมด $N - d_2 - d_1 - 1$ ตัว

แล้วเพิ่มเข้าไปในลิสต์ s ในบรรทัดที่ 4 จะทำการสลับเลขทุกตัวที่อยู่ในลิสต์ s เพื่อเป็นการสุ่มพหุนามไตรภาคแล้วทำการเพิ่มเลขจำนวนเต็มท้ายสุดของลิสต์ s เพื่อให้พหุนามมีดีกรีสูงสุดเท่ากับ N แล้วทำสร้างพหุนามในริง R แล้วทำการส่งค่ากลับไปในบรรทัดที่ 6

ขั้นตอนวิธี 3.4 การหาฟังก์ชันผกผันของ f แล้วส่งไปยัง R_q

Input: $f(x)$ ในริงสังวัตนาการ R

Output: $F_q(x)$ ในริงสังวัตนาการ R_q

1: $\text{return map\_to\_R}_\text{q}(f)\text{^}(-1)$

ขั้นตอนวิธีที่ 3.4 ใช้ในการหาฟังก์ชันผกผันของ $f(x)$ เราจะใช้ทำการแปลง $f(x)$ ให้อยู่ในริงสังวัตนาการ R_q ด้วยขั้นตอนวิธี 3.8 แล้วจึงทำการหาฟังก์ชันผกผันในริงสังวัตนาการ R_q

ขั้นตอนวิธี 3.5 การหาฟังก์ชันผกผันของ f แล้วส่งไปยัง R_p

Input: $f(x)$ ในริงสังวัตนาการ R

Output: $F_q(x)$ ในริงสังวัตนาการ R_p

1: $\text{return map\_to\_R}_\text{p}(f)\text{^}(-1)$

ขั้นตอนวิธีที่ 3.5 ใช้ในการหาฟังก์ชันผกผันของ $f(x)$ เราจะใช้ทำการแปลง $f(x)$ ให้อยู่ในริงสังวัตนาการ R_p ด้วยขั้นตอนวิธี 3.9 แล้วจึงทำการหาฟังก์ชันผกผันในริงสังวัตนาการ R_p

ขั้นตอนวิธี 3.6 การหากุญแจสาธารณะ h ในริงสังตนาการ R_q

Input: $f(x)$ ในริงสังวัตนาการ R และ g ในริงสังวัตนาการ R

Output: $h(x)$ ในริงสังวัตนาการ R_q

1: $\text{return map\_to\_R}_\text{q}(g) * F\_q$

สำหรับขั้นตอนวิธี 3.6 จะเป็นการคำนวณหากุญแจสาธารณะซึ่งสามารถคำนวณได้ดังสมการ (2.13) ซึ่งในโปรแกรมบรรทัดที่ 1 จะทำการแปลง g ให้อยู่ในริง R_q ก่อนเพื่อทำการคำนวณแล้วจึงมาคูณกับ F_q ซึ่งจะได้ผลลัพธ์อยู่ในริง R_q แล้วส่งผลลัพธ์กลับไป

ขั้นตอนวิธี 3.7 การส่งฟังก์ชัน $f(x)$ ให้อยู่ในริงสังวัตนาการ R

Input: $f(x)$ ในริงสังวัตนาการ $\mathbb{Z}$ และ p จากตัวแปรเสริมสาธารณะ

Output: $f(x)$ ในริงสังวัตนาการ R

1: $\mathrm{R.} < \mathrm{x} > = \mathrm{PolynomialRing}(\mathbb{Z})$

2: $\mathrm{idR} = \mathrm{R.ideal}(x^N - 1)$

3: $\mathrm{QuoR.} < \mathrm{x} >= \mathrm{R.quotient\_ring(idR)}$

4: $\mathrm{return\ QuoR}(f)$

ขั้นตอนวิธี 3.8 การส่งฟังก์ชัน $f(x)$ ให้อยู่ในริงสังวัตนาการ R_p

Input: $f(x)$ ในริงสังวัตนาการ $\mathbb{Z}$ และ p จากตัวแปรเสริมสาธารณะ

Output: $f(x)$ ในริงสังวัตนาการ R_p

1: $\mathrm{Rp} = \mathrm{IntegerModRing}(p)$

2: $\mathrm{Rp.} < \mathrm{x} >= \mathrm{PolynomialRing(Rp)}$

3: $\mathrm{idRp} = \mathrm{Rp.ideal}(\mathrm{x}^{\mathrm{N}} - 1)$

4: $\mathrm{QuoRp.} < \mathrm{x} >= \mathrm{Rp.quotient\_ring(idRp)}$

5: $\mathrm{return\ QuoRp}(f)$

ขั้นตอนวิธี 3.9 การส่งฟังก์ชัน $f(x)$ ให้อยู่ในริงสังวัตนาการ R_q

Input: $f(x)$ ในริงสังวัตนาการ $\mathbb{Z}$ และ q จากตัวแปรเสริมสาธารณะ

Output: $h(x)$ ในริงสังวัตนาการ R_q

1: $\mathrm{Rp} = \mathrm{IntegerModRing}(q)$

2: $\mathrm{Rp.} < \mathrm{x} >= \mathrm{PolynomialRing(Rp)}$

3: $\mathrm{idRp} = \mathrm{Rp.ideal}(\mathrm{x}^{\mathrm{N}} - 1)$

4: $\mathrm{QuoRp.} < \mathrm{x} >= \mathrm{Rp.quotient\_ring(idRp)}$

5: $\mathrm{return\ QuoRp}(f)$

ในขั้นตอนวิธีที่ 3.7 ถึง 3.9 นั้น จะเป็นขั้นตอนวิธีในการสร้างพหุนามในริงสังวัตนาการ $\mathbb{Z}$ ให้อยู่บนริง R, R_p, R_q ตามลำดับโดยแต่ละขั้นตอนนั้น เราจะทำการสร้างริงขึ้นมาก่อน แล้วจึงนำไปใช้ในการสร้างริงพหุนาม หลังจากนั้นเราใช้คำสั่งในการสร้างริงพหุนามผลหารขึ้นมา ขั้นตอนถัดมาจะเป็น

การสร้างริงสังวัตนาการขึ้นมา เมื่อได้ริงสังวัตนาการแล้ว เราจะใช้ในการสร้างคำตอบของแต่ละขั้นตอนวิธี

ตัวอย่างเซจ 3.2 จากขั้นตอนวิธีที่ 3.2 ถึงขั้นตอนวิธีที่ 3.9 สามารถทำการก่อกำเนิดกุญแจสาธารณะและกุญแจส่วนบุคคล ด้วยคำสั่งดังต่อไปนี้

```
#----------------------------------------------------------
# Ring Mapping
#----------------------------------------------------------
# ZZ[x] --> R=ZZ[x]/(x^N-1)
def map_to_R(a):
    R.<x> = PolynomialRing(ZZ)
    idR=R.ideal(x^N-1)
    QuoR.<x>=R.quotient_ring(idR)
    return QuoR(a)
#----------------------------------------------------------
# ZZ[x] --> Rp=ZZp[x]/(x^N-1)

def map_to_Rp(a):
    Rp=IntegerModRing(p)
    Rp.<x>=PolynomialRing(Rp)
    idRp=Rp.ideal(x^N-1)
    QuoRp.<x>=Rp.quotient_ring(idRp)
    return QuoRp(a)
#----------------------------------------------------------
# ZZ[x] --> Rq=ZZq[x]/(x^N-1)

def map_to_Rq(a):
    Rq=IntegerModRing(q)
    Rq.<x>=PolynomialRing(Rq)
    idRq=Rq.ideal(x^N-1)
    QuoRq.<x>=Rq.quotient_ring(idRq)
    return QuoRq(a)
#----------------------------------------------------------
# Tripolynomial
#----------------------------------------------------------
def tri_poly(d_1, d_2) :
    s = [1 for j in range(d_1 - 1)]
    s = s + [-1 for j in range(d_2)]
    s = s + [0 for j in range(N - d_2 - d_1)]
    random.shuffle(s)
    s.append(1)
    return map_to_R(s)
#----------------------------------------------------------
# Fq = f^(-1) in Rq
```

```
#-------------------------------------------------------
def fq_inv(f) :
    return map_to_Rq(f)^(-1)
#-------------------------------------------------------
# Fp = f^(-1) in Rp
#-------------------------------------------------------
def fp_inv(f) :
    return map_to_Rp(f)^(-1)

#-------------------------------------------------------
def find_h(F_q, g) :
    return map_to_Rq(g)*F_q
```

ผลรัน:

กำหนดให้ค่ากุญแจสาธารณะมีค่าดังตัวอย่างที่ 3.1 แล้วได้ผลรันดังต่อไปนี้

ครั้งที่ 1: เมื่อ $(N, p, q, d) = (17, 11, 11437, 7)$

```
#-------------------------------------------------------
# Key generated by Alice
#-------------------------------------------------------
(N,p,q,d) = (17, 11, 11437, 7)
#-------------------------------------------------------
f = tri_poly(d+1, d)
g = tri_poly(d, d)
F_q = fq_inv(f)
F_p = fp_inv(f)
h = find_h(F_q, g)
print "Alice's private key (f,F_p): "
show((f,F_p))
print "Alice's public key (h): "
show(h)
```

จะได้ผลการรัน

Alice's private key (f,F_p):

$$(x^{16} + x^{15} + x^{14} + x^{13} + x^{12} - x^{11} - x^{10} - x^9 - x^7 + x^6 - x^4 - x^3 + x^2 - x + 1, 3x^{16} + 4x^{15} + 8x^{14} + 6x^{13} + 3x^{11} + 8x^{10} + 4x^9 + 7x^8 + 8x^7 + 2x^6 + 9x^5 + x^4 + 4x^2 + x + 8)$$

Alice's public key (h):

$$249x^{16} + 9549x^{15} + 4669x^{14} + 7801x^{13} - 7602x^{12} - 10565x^{11} + 5883x^{10} - 7757x^9 + 5531x^8 + 8444x^7 + 1183x^6 + 4848x^5 - 2021x^4 + 2846x^3 - 3440x^2 + 11044x - 7788$$

ครั้งที่ 2: เมื่อ $(N, p, q, d) = (13, 17, 9817, 3)$

```
#-------------------------------------------------------
```

```
# Key generated by Alice
#----------------------------------------------------------
(N,p,q,d) = (13, 17, 9817, 3)
#----------------------------------------------------------
f = tri_poly(d+1, d)
g = tri_poly(d, d)
F_q = fq_inv(f)
F_p = fp_inv(f)
h = find_h(F_q, g)
print "Alice's private key (f,F_p): "
show((f,F_p))
print "Alice's public key (h): "
show(h)
```

จะได้ผลการรัน

```
Alice's private key (f,F_p):
```

$$(x^{12} - x^{10} - x^7 + x^6 - x^5 + x^4 + x^2, 7x^{12} + x^{10} + 13x^9 + 11x^8 + 10x^7 + 15x^6 + 5x^5 + 12x^3 + 4x^2 + 3x + 9)$$

```
Alice's public key (h):
```

$$1486x^{12} - 3308x^{11} - 5800x^{10} - 2980x^9 + 1934x^8 + 7664x^7 + 5266x^6 - 1799x^5 - 5068x^4 - 5838x^3 - 2191x^2 + 4173x + 6461$$

ในส่วนของผลรันนั้น เราจะเห็นได้ว่าดีกรีของกุญแจสาธารณะและกุญแจส่วนบุคคลนั้นมีค่าน้อยกว่า N โดยกุญแจส่วนบุคคลนั้นจะถูกสร้างจากฟังก์ชัน tri_poly() ซึ่งเป็นฟังก์ชันที่มีขั้นตอนดังขั้นตอนวิธีที่ 3.3 ถัดมาก็จะทำการหาตัวผกผันของ f ด้วยฟังก์ชัน fq_inv() และ fp_inv() ซึ่งจะได้ F_q และ F_p ตามลำดับ fq_inv และ fp_inv มีขั้นตอนวิธีการทำงานดังขั้นตอนวิธีที่ 3.4 และ 3.5 ตามลำดับ หลังจากนั้นก็จะทำการคำนวณ h ซึ่งเป็นกุญแจสาธารณะด้วยฟังก์ชัน find_h ด้วยขั้นตอนวิธีดังขั้นตอนวิธีที่ 3.6 จะทำให้ได้กุญแจสารธารณะ ในบรรทัดที่ 10-14 จะเป็นส่วนของการแสดงผล

#

3.3 การเข้ารหัสลับ

บ็อบเป็นผู้เข้ารหัสลับเพลนเท็กซ์ $m(x)$ ในริงสังวัตนาการ R_p จะทำการสุ่มพหุนามไตรภาค $r(x) \in \tau(d,d)$ จากนั้นใช้กุญแจสาธารณะของอลิซ $h(x)$ ทำการคำนวณ $e(x) \equiv p \cdot (x) \star h(x) + m(x)(mod\ q)$ แล้วส่งไซเฟอร์ $e(x)$ ไปยังอลิซ ดังขั้นตอนวิธีที่ 3.7

ขั้นตอนวิธี 3.10 การเข้ารหัสลับเอ็นทรู

Input: $m(x)$ ในริงสังวัตนาการ R_p, $h(x)$ ในริงสังวัตนาการ R_q และ $r(x)$ ในริงสังวัตราการ R

Output: $e(x)$ ในริงสังวัตนาการ R_q

1: $\mathrm{m} = \mathrm{map\_to\_R}(m)$

2: $\mathrm{r} = \mathrm{map\_to\_R}(r)$

3: $\mathrm{return\ map\_to\_Rq}((p * r * h) + m)$

ในขั้นตอนวิธีที่ 3.10 เราจะทำการคำนวณหาไซเฟอร์เท็กซ์ $e(x)$ เพื่อส่งไปยังผู้รับคือบ็อบโดยคำนวณดังสมการที่ (2.14) โดยในขั้นตอนวิธีที่ 3.10 เราจะทำการแปลง m และ r ให้อยู่ในริงสังวัตนาการ R ด้วยขั้นตอนวิธี 3.7 แล้วจึงทำการคำนวณดังสมการที่ (2.14) หลังจากนั้นทำการแปลงในอยู่ในริงสังวัตนาการ R_q ด้วยขั้นตอนวิธี 3.9

ตัวอย่างเซจ 3.3 จากขั้นตอนวิธีที่ 3.10 และเพื่อเป็นการยืนยันในความถูกต้องของขั้นตอนวิธีนี้ เราจึงใช้พารามิเตอร์ดังตัวอย่างที่ 2.7 นั้นคือ $(N,p,q,d) = (7,3,41,2)$ และพหุนามไตรภาคเป็น $f(x) = \mathrm{x}^6 - \mathrm{x}^4 + \mathrm{x}^3 + \mathrm{x}^2 - 1 \in \tau(3,2)$ และ $g(x) = x^6 + x^4 - x^2 - x \in \tau(2,2)$ แล้วได้กุญแจส่วนบุคคล $\left(f(x), F_p(x)\right)$ และกุญแจสารธารณะที่สัมพันธ์ด้วยเป็น $h(x)$ ให้สารที่บ็อบต้องการส่งเป็น $m(x) = -x^5 + x^3 + x^2 - x + 1$ ด้วยสัมประสิทธิ์มีค่าระหว่าง $-\frac{p}{2}$ และ $\frac{p}{2}$ และให้สุ่มได้กุญแจครั้งเดียวเป็น $r(x) = x^6 - x^5 + x - 1 \in \tau(d,d)$ แล้วสามารถก่อกำเนิดกุญแจ และทำการเข้ารหัสเอ็นทรูด้วยคำสั่งต่อไปนี้

```
#---------------------------------------------------------
# Encryption Function
#---------------------------------------------------------
def encrypt(m, h, r) :
    m=map_to_R(m)
    r=map_to_R(r)
    return map_to_Rq((p*r*h)+m)
#---------------------------------------------------------
# Public parameter
#---------------------------------------------------------
(N,p,q,d) = (7,3,41,2)
#---------------------------------------------------------
# Key generated by Alice
#---------------------------------------------------------
f = x^6-x^4+x^3+x^2-1
g = x^6+x^4-x^2-x
f = x^6-x^4+x^3+x^2-1
g = x^6+x^4-x^2-x
F_q = fq_inv(f)
F_p = fp_inv(f)
h = find_h(F_q, g)
print "Alice's private key (f,F_p): "
show((f,F_p))
print "Alice's public key (h): "
show(h)
#---------------------------------------------------------
# Bob encrypt message
#---------------------------------------------------------
m = -x^5 + x^3 + x^2 - x + 1
#---------------------------------------------------------
# Ephimeral key
#---------------------------------------------------------
r= x^6 - x^5 + x-1
#---------------------------------------------------------
e = encrypt(m, h, r)
print "Plant text : "
show(m)
print "Cypher text : "
show(e)
```

ผลรัน:

```
Alice's private key (f,F_p):
```

$$(x^6 - x^4 + x^3 + x^2 - 1, x^6 + 2x^5 + x^3 - 2x^2 + x + 1)$$

```
Alice's public :
```

$$20x^6 + 40x^5 + 2x^4 + 38x^3 + 8x^2 + 26x + 30$$

```
Plant text :
```

$$-x^5 + x^3 + x^2 - x + 1$$

```
Cypher text :
```

$$31x^6 + 19x^5 + 4x^4 + 2x^3 + 40x^2 + 3x + 25$$

ในส่วนแรกจะเป็นการสร้างฟังก์ชัน encrypt() ซึ่งมีขั้นตอนวิธีดังขั้นตอนวิธีที่ 3.10 โดยจะใช้ในการเข้ารหัส ในส่วนถัดมาเป็นการพารามิเตอร์ต่างๆดังตัวอย่างที่ 2.12 แล้วทำการสร้างกุญสาธารณะและกุญแจส่วนบุคคลดังขั้นตอนวิธีที่ 3.2 ทำการเลือกข้อความและกุญแจใช้ครั้งเดียวดังตัวอย่างที่ 2.12 แล้วทำการเรียกฟังก์ชัน encrypt() เพื่อทำการเข้ารหัสเพลนเทกซ์ที่เลือกไว้

จากผลการรันเราจะเห็นได้ว่ากุญแจสาธารณะที่ได้จากการคำนวณนั้นจะได้ผลลัพธ์เท่ากับตัวอย่างที่ 2.12 เมื่อใช้ f และ g เท่ากับค่าในตัวอย่าง หลังจากนั้นเราจึงนำกุญแจสาธารณะทั้งสองตัวนั้นมาทำการเข้ารหัสซึ่งจะทำให้ได้ไซเฟอร์เทกซ์ที่ตรงกับค่าในตัวอย่าง เมื่อนำกุญแจใช้ครั้งเดียวซึ่งมีค่าเดียวกับในตัวอย่างที่ 2.12 จะมีไซเฟอร์เทกซ์ที่ตรงกับตัวอย่าง

#

3.4 การถอดรหัสลับ

เมื่ออลิซได้รับไซเฟอร์ $e(x)$ ในริงสังวัตนาการ R_q จากนั้นจะนำมาคูณด้วยกุญแจส่วนบุคคล $f(x)$ ในริงสังวัตนาการ R แล้วได้พหุนาม $a(x)$ ในริงสังวัตนาการ R_q ตามด้วยการทำเซ็นเตอร์ลิฟต์ $a(x)$ ไปยังริงสังวัตนาการ R ด้วยขั้นตอนวิธีที่ 3.11 และท้ายสุดนำ $F_p(x)$ ในริงสังวัตนาการ R_p มาคูณกับ $a(x)$ ตามด้วยการทำเซ็นเตอร์ลิฟต์แล้วได้ $\widehat{m}(x)$ ในริงสังวัตนาการ R ดังขั้นตอนวิธีที่ 3.12

ขั้นตอนวิธี 3.11 การทำเซ็นเตอร์ลิฟต์ $a(x)$

Input: $a(x)$ ในริงสังวัตนาการ R_q และจำนวนเต็มบวก t

Output: $a(x)$ ที่ถูกลิฟต์ในริงสังวัตนาการ R

$lst\_a = \text{list}(a)$

for j to len(lst_a) **do**

    if $lst\_a[j] <= -t/2$ **then**

        $lst\_a[j] = lst\_a[j] + t$

    end if

    if $lst\_a[j] > t/2$ **then**

        $lst\_a[j] = lst\_a[j] - t$

    end if

end for

return $R(lst\_a)$

ขั้นตอนวิธี 3.12 การถอดรหัสลับเอ็นทรู

Input: กุญแจส่วนบุคคล $f(x)$ ในริงสังสัตนาการ $R, F_p(x)$ ในริงสังตนาการ R_p และไซเฟอร์ $e(x)$ ในริงสังวัตนาการ R_q

Output: $\widehat{m}(x)$ ในริงสังวัตนาการ R

f = map_to_R(f)

a = map_to_Rq($f * e$)

a = map_to_R(a)

a = center_lift(a, q)

$\widehat{m}(x)$ = map_to_Rp(F_p $* a$)

$\widehat{m}(x)$ = map_to_R(mhat)

7:	$\widehat{m}(x) = \text{center\_lift}(\widehat{m}, p)$
8:	**return** $\widehat{m}(x)$

ในขั้นตอนวิธีที่ 3.11 นั้นเป็นขั้นตอนของการทำเซ็นเตอร์ลิฟต์เพื่อย้ายพหุนามให้อยู่ในริง R โดยการรับอินพุทเป็นพหุนาม $a(x)$ ที่ต้องการทำเซนเตอร์ลิฟต์ และจำนวนเต็มบวก t ซึ่งเป็นอันดับของริงที่ซึ่ง $a(x)$ เป็นสมาชิก ในบรรทัดที่ 1 นั้น เราจะทำการแปลงพหุนามให้อยู่รูปลิสต์ของสัมประสิทธิ์ และในขั้นตอนถัดมาเราจะทำการวนทุกค่าในลิสต์ของสัมประสิทธิ์เพื่อทำการปรับให้ค่าอยู่ในช่วง $-t/2 < a'(x) \leq t/2$ โดยจะทำการแบ่งออกเป็นสองกรณี ถ้าหากสัมประสิทธิ์ตัวที่ j มีค่าน้อยกว่าหรือเท่ากับ $-t/2$ เราจะทำการบวกสัมประสิทธิ์ตัวนั้นด้วย t และในอีกกรณีถ้าหากสัมประสิทธิ์ตัวที่ j มีค่ามากว่า $t/2$ ก็จะทำการลบด้วยค่า t วนไปจนกระทั่งครบทุกตำแหน่งในลิสต์ ในบรรทัดสุดท้ายจึงทำการนำลิสต์นั้นไปสร้างเป็นพหุนามในริง R แล้วจึงส่งค่ากลับไปซึ่งเป็นพหุนาม $a(x)$ ที่ถูกลิฟต์แล้ว

ในขั้นตอนวิธีที่ 3.12 นั้นเป็นขั้นตอนในการถอดรหัสเมื่อได้รับไซเฟอร์เท็กซ์ $e(x)$ มาแล้วทำการคำนวณดังสมการที่ (2.15) เมื่อเราได้รับค่าต่างๆมาแล้ว เราจะทำการจัดค่าต่างๆให้อยู่ในริงที่ถูกต้องโดยใช้ขั้นตอนวิธีที่ 3.7 ถึง 3.9 หลังจากนั้นจะเป็นการคำนวณ $a(x)$ จาก $e(x) \star f(x)$ แล้วแปลงให้อยู่ในริงสังวัตนาการ R_q แปลงกลับมาให้อยู่ในริง R ในบรรทัดถัดมาเป็นการเซ็นเตอร์ลิฟต์ $a(x)$ เพื่อให้อยู่ริงสังวัตนาการ R โดยใช้ขั้นตอนวิธีที่ 3.11 ในการทำเซ็นเตอร์ลิฟต์ ในบรรทัดที่ 5 ทำการคำนวณ $F_p(x)$ คูณกับ $a(x)$ ในริงสังวัตนาการ R แล้วจึงส่งไปยังริงสังวัตนาการ R_pจะได้ผลลัพธ์เก็บอยู่ใน $\widehat{m}(x)$ ทำการแปลงให้อยู่ในริงสังวัตนาการ R แล้วทำการเซ็นเตอร์ลิฟต์ $\widehat{m}(x)$ ด้วยขั้นตอนวิธีที่ 3.11 ซึ่งจะได้ แล้วจึงส่งค่า $\widehat{m}(x)$ กลับไปเป็นผลลัพธ์ของการถอดรหัส

ตัวอย่างเซจ 3.4 จากขั้นตอนวิธีที่ 3.9 การถอดรหัสลับเอ็นทรู และการคำนวณเซ็นเตอร์ลิฟต์ในขั้นตอนวิธีที่ 3.8 สามารถทำการถอดรหัสลับเอ็นทรู ด้วยคำสั่งต่อไปนี้

```
#--------------------------------------------------------
# Center lift function
#--------------------------------------------------------
def center_lift(a, t) :
    lst_a = list(a)
    for i in range(len(lst_a)) :
        if lst_a[i] <= -t/2 :
            lst_a[i] += t
        elif lst_a[i] > t/2 :
            lst_a[i] -= t
    return R(lst_a)
```

```
#-------------------------------------------------------
# Decrypt recieved cypher text e(x)
#-------------------------------------------------------
def decrypt(f, F_p, e) :
    f = map_to_R(f)
    a = map_to_Rq(f*e)
    a = map_to_R(a)
    #a = ((f * e) % id) % q
    a = center_lift(a, q)
    mhat = map_to_Rp(F_p*a)
    mhat = map_to_R(mhat)
    mhat = center_lift(mhat, p)
    return mhat
#-------------------------------------------------------
# Alice decrypt cypher text
#-------------------------------------------------------
mhat_lift = decrypt(f, F_p, e)
print ("Recovery message: ")
show(mhat_lift)
#-------------------------------------------------------
```

ผลรัน:

```
Recovery message
```

$$-x^5 + x^3 + x^2 - x + 1$$

ส่วนแรกของคำสั่งจะเป็นการสร้างฟังก์ชัน center_lift() และ decrypt() โดยฟังก์ชันจะมีขั้นตอนการทำงานดังขั้นตอนวิธีที่ 3.11 โดยจะรับอินพุทคือพหุนาม $a(x)$ และค่ามอดูโล t ในส่วนของ decrypt() ซึ่งจะใช้ในการถอดรหัสไซเฟอร์เทกซ์ที่ได้รับมาโดยรับพารามิเตอร์คือกุญแจส่วนบุคคล f ตัวผกผันการคูณ F_p และไซเฟอร์เทกซ์ตามลำดับ ฟังก์ชัน decrypt() จะมีขั้นตอนการทำงานดังขั้นตอนวิธีที่ 3.12 โดยจะทำการเรียกใช้ฟังก์ชัน center_lift() ซึ่งมีขั้นตอนการทำงานดังขั้นตอนวิธีที่ 3.11 เพื่อใช้ในการหาเซ็นเตอร์ลิฟต์โดยรับพารามิเตอร์คือ พหุนามที่ต้องการจะทำเซ็นเตอร์ลิฟต์และจำนวนเต็ม

ในส่วนถัดมาจะทำการเรียกใช้ฟังก์ชัน decrypt() แล้วนำมาแสดงผล โดยในครั้งแรกเราจะใช้ไซเฟอร์เทกซ์ที่เกิดจากการนำกุญแจสารธารณะที่ถูกสร้างขึ้นมาโดยโปรแกรมและในครั้งที่สองเราจะทำการนำไซเฟอร์เทกซ์ที่ได้มาจากกุญแจสาธารณะที่มีค่าเท่ากับกุญแจสาธารณะในตัวอย่างที่ 2.12 มาทำการถอดรหัส ซึ่งจะเห็นได้ว่าเพลนเทกซ์ที่ได้จากการถอดรหัสนั้นตรงกับเพลนเทกซ์ที่ได้ทำการเลือกไว้ทั้งสองกรณี

#

บทที่ 4
ผลการทดลองและอภิปรายผล

ในบทนี้จะทำการทดลองวัดประสิทธิภาพการคำนวณของแผนวิธีวิทยาการเข้ารหัสลับเอ็นทรูโดยเปรียบเทียบกับแผนวิธีวิทยาการเข้ารหัสลับเอ็ลการมอลเส้นโค้งเชิงวงรี เมื่อขนาดของฟีลด์ใกล้เคียงกัน เริ่มจากการวัดประสิทธิภาพการสร้างตัวแปรเสริมสาธารณะ แล้ววัดประสิทธิภาพการสร้างกุญแจ จากนั้นวัดประสิทธิภาพการเข้ารหัสลับ แล้ววัดประสิทธิภาพการถอดรหัสลับ

4.1 เครื่องมือในการทดลองและข้อกำหนด

4.1.1 เครื่องมือในการทดลอง

ในการทดลองของโครงการนี้ใช้คอมพิวเตอร์แบบพกพา มีรายละเอียดดังนี้

ชื่อรุ่นของอุปกรณ์: MacBook Air

ระบบปฏิบัติการ : macOS Mojave

ซีพียู : 1.6 GHz Intel Core i5

หน่วยความจำแรม : 8GB

โปรแกรมที่ใช้ในการรันขั้นตอนวิธี : Sagemath รุ่น 8.6 ผ่าน Google Chrome บราวเซอร์ร่วมกับ Jupyter Notebook

4.1.2 ข้อกำหนด

ในการทดสอบประสิทธิภาพครั้งนี้ เราได้ใช้ขั้นตอนการคำนวณของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสองโดยอ้างอิงจากงาน แผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะเฉพาะสอง ของ ปานดาว แก้วมณี และ ศิวารุจพรรคชัย [9] ในส่วนของขั้นตอนการคำนวณของวิทยาการรหัสลับเอ็นทรูนั้น เราได้ทำการออกแบบโดยอ้างอิงจากหนังสือ An introduction to mathematical cryptography vol. 1 ซึ่งแต่งโดย Hoffstein J Pipher JC และ Silverman JH

กำหนดริงผลหารของพหุนามเป็นฟีลด์ลักษณะเฉพาะสอง เขียนแทนด้วย $\mathbb{F}_{2^k}$ ประกอบด้วยสมาชิกจำนวน 2^k จำนวน เมื่อ k เป็นจำนวนเต็มบวก ในการวัดประสิทธิภาพของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีเหนือเส้นโค้งคอบลิทซ์จะกำหนดค่า k คือ 160 ซึ่งจะได้ขนาดของกุญแจสาธารณะอยู่ที่ 2^k บิตซึ่งจะได้ระดับบิตความปลอดภัยอยู่ที่ 80 จากตารางที่ 2.4

กำหนดขนาดของตัวแปรเสริมสาธารณะ N ของวิทยาการเข้ารหัสลับเอ็นทรูอยู่ในช่วง $next\_prime(randrange(2^l, 2^{l+1}))$ โดยประมาณ ค่า l ที่ใช้ในการทดสอบประสิทธิภาพนั้นมีค่า $l = 7$ ซึ่งจะทำให้ได้ค่า N อยู่ในช่วง $[131{,}257]$ ซึ่งจากตารางที่ 2.3 จะทำให้ได้ว่าระดับบิตความปลอดภัยอยู่ในช่วง $[57{,}88]$

4.2 การวัดประสิทธิภาพการสร้างตัวแปรเสริมสาธารณะ

ในขั้นตอนนี้เป็นการวัดประสิทธิภาพของขั้นตอนวิธีการสร้างตัวแปรเสริมสาธารณะ โดยจะเปรียบเทียบระยะเวลาที่ใช้ในการสร้างตัวแปรเสริมสาธารณะของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี และวิทยาการรหัสลับเอ็นทรู ในส่วนของการสร้างตัวแปรเสริมสาธารณะของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี เราจะใช้การคำนวณดังตารางที่ 2.1 และในส่วนของวิทยาการรหัสลับเอ็นทรู เราจะใช้ขั้นตอนวิธีที่ 3.1 ในการทดสอบ

ตัวอย่างเซจ 4.1 โปรแกรมวัดประสิทธิภาพขั้นตอนการสร้างตัวแปรเสริมสาธารณะของวิทยาการเข้ารหัสเอ็นทรู

เราสามารถใช้คำสั่งต่อไปนี้ในการวัดประสิทธิภาพของขั้นตอนการสร้างตัวแปรเสริมสาธารณะของวิทยาการรหัสลับเอ็นทรูและคำนวณขนาดของกุญแจสาธารณะ โดยในตัวอย่างนี้จะใช้ค่า $l = 2$

```
#-------------------------------------------------------
# Public parameter generator input = degree k
#-------------------------------------------------------
def parameter_gen(k):
    flag = True
    while flag :
        N = next_prime(randrange(2**k, 2**(k+1)))
        p = next_prime(randrange(2**k, 2**(k+1)))
        d = randrange(1, N)
        q = next_prime(randrange((6*d + 1)*p, (6*d +
1)*(p*2^5)))
        if gcd(p,q) == 1 :
            if gcd(N,q) == 1 :
                if q > (6*d + 1) * p :
                    flag = False
    return (N,p,q,d)
#--------------------------------------------------------
l = 2
```

```
t = cputime()
(N,p,q,d) = parameter_gen(l)
ti = cputime(t)
print ("Cpu time = " + str(ti))
print ("(N,p,q,d) = " + str((N,p,q,d)))
```

ได้ผลรันดังต่อไปนี้

```
Cpu time = 0.001109
(N,p,q,d) = (7, 11, 1499, 1)
```

จากตัวอย่างเซจที่ 4.1 เราจะได้ตัวแปรเสริมสาธารณะซึ่งจะใช้ในการสร้างกุญแจโดยตัวแปรสาธารณะที่ได้นั้นจะมีค่าเป็นไปตามเงื่อนไขดังตารางที่ 2.2 ซึ่งแสดงถึงขั้นตอนารเข้ารหัสแบบเอ็นทรูและจะทำให้ทราบถึงขนาดกุญแจสาธารณะ

#

ตัวอย่างเซจ 4.2 โปรแกรมวัดประสิทธิภาพขั้นตอนการสร้างตัวแปรเสริมสาธารณะของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

เราสามารถใช้คำสั่งต่อไปนี้ในการจับเวลาขั้นตอนการสร้างตัวแปรสาธารณะวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีได้ โดยในตัวอย่างนี้จะใช้ค่า $k = 8$

```
# Create All Shared Varible Ea, P
k = 8
t = cputime()
# Start Timer
S.<V> = GF(2^k)
Sgen = S.gen() #Genrator of FiniteField
Sord = S.order() #Order of FiniteField
a = randrange(0, Sord-1)
A = Sgen^a #Random A parameter for EllipticCurve
Ea = EllipticCurve(S,[1,A,0,0,1]) #Create EllipticCurve Over
Finite Field 2^l
g = Ea.gen(0) # Random 1 point in EllipticCurve
Eord = Ea.order()
n = randrange(1, Eord-1)
P = n*g # Random P point in EllipticCurve
ti = cputime(t)
print ("Cpu time = " + str(ti))
print("Public Varible P : " + str(P))
print("EllipticCurve Order : " + str(Eord))
print("Ea = " + str(Ea))
```

ได้ผลรันดังต่อไปนี้

```
Cpu time = 0.08904
Public Varible P : (V^5 + V^3 + V + 1 : V^3 + V + 1 : 1)
EllipticCurve Order : 226
Ea = Elliptic Curve defined by y^2 + x*y = x^3 + (V^6+V^5+V^4
+V^3+V^2)*x^2 + 1 over Finite Field in V of size 2^8
```

จากตัวอย่างเซจที่ 4.2 เป็นผลรันที่ได้จากโปรแกรมที่ใช้ในการสร้างตัวแปรเสริมสาธารณะ เมื่อกำหนดค่า $k = 8$ โดยจะได้ผลตัวแปรสาธารณะ P ซึ่งเป็นจุดบนเส้นโค้งเชิงวงรี E_a ซึ่งจะใช้ในขั้นตอนต่างๆของวิทยาการรหัสลับลับเอ็ลกามอลเส้นโค้งเชิงวงรี นอกจากนี้ยังแสดงถึงจำนวนจุดบนเส้นโค้งเชิงวงรีและสมการของเส้นโค้งเชิงวงรี E_a

#

ตารางที่ 4.1 ผลการวัดประสิทธิภาพของขั้นตอนการสร้างตัวแปรเสริมสาธารณะ

ลำดับ	ขนาดพารามิเตอร์ N ของ NTRU	การสร้างตัวแปรเสริมสาธารณะ (วินาที)	
		NTRU	ECC
1	163.000000	0.000114	0.190317
2	257.000000	0.000149	0.090256
3	239.000000	0.000094	0.117454
4	251.000000	0.000104	0.107994
5	211.000000	0.000143	0.107395
6	223.000000	0.000208	0.107191
7	223.000000	0.000164	0.141290
8	233.000000	0.000115	0.096128
9	223.000000	0.000149	0.122634
10	199.000000	0.000176	0.104254
11	149.000000	0.000112	0.134396
12	191.000000	0.000101	0.127720
13	163.000000	0.000486	0.111161
14	211.000000	0.000113	0.119428
15	257.000000	0.000129	0.088716
16	191.000000	0.000130	0.141194
17	211.000000	0.000111	0.095562

ตารางที่ 4.1 ผลการวัดประสิทธิภาพของขั้นตอนการสร้างตัวแปรเสริมสาธารณะ (ต่อ)

18	191.000000	0.000141	0.119609
19	211.000000	0.000098	0.131713
20	179.000000	0.000153	0.132922
21	193.000000	0.000104	0.122860
22	191.000000	0.000097	0.111714
23	251.000000	0.000105	0.104760
24	239.000000	0.000159	0.093535
25	191.000000	0.000099	0.122409
26	173.000000	0.000095	0.182855
27	251.000000	0.000688	0.132502
28	149.000000	0.000097	0.094003
29	163.000000	0.000125	0.122964
30	257.000000	0.000149	0.127715

ค่าเฉลี่ย	207.800000	0.000157	0.120088
ค่ามัธยฐาน	211.000000	0.000120	0.119519
ส่วนเบี่ยงเบนมาตรฐาน	33.662012	0.000123	0.023578

อภิปรายผล

จากการทดลองสร้างตัวแปรเสริมสาธารณะด้วยวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีและทำการบันทึกค่าดังตารางที่ 4.1 เมื่อค่า $k = 160$ บิต และทำการทดสอบทั้งสิ้น 30 ครั้ง จะได้เวลาเฉลี่ยที่ใช้ในการสร้างตัวแปรเสริมสาธารณะคือ 0.120088 วินาที ค่ามัธยฐาน คือ 0.119519 วินาที และส่วนเบี่ยงเบนมาตรฐาน 0.023578 จะเห็นว่า ค่าเฉลี่ยมีค่ามากกว่าค่ามัธยฐานนั่นแสดงถึงเวลาที่ใช้ในการสร้างตัวแปรเสริมส่วนมากนั้นมีค่าน้อยกว่าค่าเฉลี่ย ทำให้ข้อมูลมีการแจกแจงแบบเบ้ซ้าย ส่วนเบี่ยงเบนมาตรฐานแสดงให้เห็นว่าเวลาที่ใช้ในการทดลองนั้นมีค่าต่างจากค่าเฉลี่ยไม่สูงมาก

ในส่วนของการวัดประสิทธิภาพของการสร้างตัวแปรเสริมวิทยาการหัสลับเอ็นทรูได้ทำการทดสอบทั้งหมด 30 ครั้ง จากตารางที่ 4.1 จะเห็นได้ว่าค่า N มีค่าเฉลี่ยอยู่ที่ 207.8 โดยที่ใช้เวลาเฉลี่ย 0.000157 วินาที จากค่ามัธยฐานแสดงให้เห็นว่าค่า N ที่ใช้ในการทดลองส่วนมากมีค่ามากกว่า

ค่าเฉลี่ยซึ่งมีค่ามัธยฐานอยู่ที่ 211 และค่ามัธยฐานของเวลาที่ใช้ในการสร้างตัวแปรเสริมสาธารณะนั้นส่วนมากมีค่าน้อยกว่าค่าเฉลี่ยโดยมีค่ามัธยฐานอยู่ที่ 0.00120 วินาที ค่าเฉลี่ยของเวลาที่ใช้มีค่ามากกว่าค่ามัธยฐานนั่นแสดงถึงเวลาที่ใช้ในการสร้างตัวแปรเสริมส่วนมากนั้นมีค่าน้อยกว่าค่าเฉลี่ย ทำให้ข้อมูลมีการแจกแจงแบบเบ้ซ้าย จากค่าส่วนเบี่ยงเบนมาตรฐาน จะเห็นได้ว่าค่า N มีค่าที่ต่างจากค่าเฉลี่ยค่อนค้างน้อยมีส่วนเบี่ยงเบนมาตรฐานเท่ากับ 33.662012 และจากค่าส่วนเบี่ยงเบนมาตรฐานของเวลาที่ใช้ในการสร้างตัวแปรเสริมสาธารณะในการทดลองแต่ละครั้ง มีค่าแตกต่างกันค่อนข้างสูงโดยมีค่าเท่ากับ 0.000123

4.3 การวัดประสิทธิภาพการสร้างกุญแจ

ตัวอย่างเซจ 4.3 โปรแกรมวัดประสิทธิภาพขั้นตอนการสร้างกุญแจของวิทยาการรหัสเอ็นทรู

เราสามารถใช้คำสั่งต่อไปนี้ในการวัดประสิทธิภาพของขั้นตอนการสร้างกุญแจของวิทยการรหัสลับเอ็นทรู โดยในตัวอย่างนี้จะใช้ค่า $l = 2$ และใช้ตัวแปรเสริมสาธารณะดังค่าที่ได้จากตัวอย่างเซจที่ 4.1

```
#----------------------------------------------------------
# Key generated by Alice
#----------------------------------------------------------
t = cputime()
f = tri_poly(d+1, d)
g = tri_poly(d, d)
F_q = fq_inv(f)
F_p = fp_inv(f)
h = find_h(F_q, g)
ti = cputime(t)
total_time += ti
print ("Cpu time = " + str(ti))
print ("Alice's private key (f,F_p): ")
show((f,F_p))
print ("Alice's public key (h): ")
show(h)
```

ได้ผลรันดังต่อไปนี้

```
Cpu time = 0.037317
Alice's private key (f,F_p):
```

$$(x^6 - x^3 + x, 5x^6 + 7x^5 + 10x^4 + 10x^3 + 4x^2 + 6x + 3)$$

```
Alice's public key (h):
```

$$1034x^6 + 52x^5 + 827x^4 + 1240x^3 + 362x^2 + 1292x + 1189$$

จากตัวอย่างเซจที่ 4.3 เราจะได้กุญแจสาธารณะและกุญแจส่วนบุคคลของวิทยการรหัสลับเอ็นทรูโดยในขั้นตอนการคำนวณกุญแจส่วนบุคคลจะเริ่มจากการสร้างพหุนามไตรภาค $f = \tau(d+1, d)$

แล้วจึงนำไปหาตัวผกผันใน R_p จะทำให้ได้กุญแจสาธารณะ ในส่วนของกุแจสาธารณะนั้น จะทำการสร้างพหุนามไตรภาค $g = \tau(d, d)$ แล้วทำการหาตัวผกผันของ f ในริง R_q แล้วนำมาคูณกับ g จะได้กุญแจสาธารณะ

#

ตัวอย่างเซจ 4.4 โปรแกรมวัดประสิทธิภาพขั้นตอนการสร้างกุญแจของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

เราสามารถใช้คำสั่งต่อไปนี้ในการจับเวลาขั้นตอนการสร้างกุญแจของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีได้ โดยในตัวอย่างนี้จะใช้ค่า $k = 8$

```
na = randrange(1, Eord - 1)
Qa = na*P
ti = cputime(t)
total_time += ti
print ("Cpu time = " + str(ti))
print("Alice's Private Key : %d" % na)
print("Alice's Public Key : " + str(Qa))
```

ได้ผลรันดังต่อไปนี้

```
Cpu time = 0.000951999999998
Alice's Private Key : 83
Alice's Public Key : (V^5 + V^4 + V^3 + V + 1 : V^7 + V^5 + V
^3 + V + 1 : 1)
```

จากตัวอย่างเซจที่ 4.4 เราจะได้กุญแจสาธารณะและกุญแจส่วนบุคคลของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีโดยกุญแจส่วนบุคคลนั้นจะทำการสุ่มขึ้นมาโดยจะอยู่ช่วง 1 ถึงขนาดสมาชิกของจุดบนเส้นโค้งเชิงวงรี หลังจากนั้นนำกุญแจส่วนบุคคลที่สุ่มได้ไปคูณกับจุด P จะทำให้ได้กุญแจสาธารณะของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

#

ตารางที่ 4.2 ผลการวัดประสิทธิภาพของขั้นตอนการสร้างสร้างกุญแจ

ลำดับ	การสร้างกุญแจ (วินาที)	
	NTRU	ECC
1	0.043601	0.018026
2	0.078247	0.010416
3	0.056357	0.017032
4	0.072286	0.016346
5	0.045302	0.016729
6	0.069735	0.017376
7	0.038515	0.016054
8	0.058850	0.020444
9	0.049776	0.014341
10	0.058243	0.010846
11	0.041478	0.018399
12	0.052041	0.010816
13	0.032451	0.018176
14	0.041746	0.013321
15	0.054702	0.010437
16	0.040519	0.012204
17	0.047180	0.017668
18	0.045108	0.014417
19	0.049727	0.017106
20	0.047351	0.011501
21	0.062323	0.016083
22	0.046573	0.015709
23	0.058144	0.015717
24	0.046214	0.010134
25	0.040968	0.015264
26	0.044884	0.015599
27	0.060036	0.016149

ตารางที่ 4.2 ผลการวัดประสิทธิภาพของขั้นตอนการสร้างสร้างกุญแจ (ต่อ)

28	0.034379	0.016578
29	0.033443	0.015556
30	0.050328	0.016853
ค่าเฉลี่ย	0.050017	0.015177
ค่ามัธยฐาน	0.047266	0.015886
ส่วนเบี่ยงเบนมาตรฐาน	0.011204	0.002760

อภิปรายผล

จากการทดลองการสร้างกุญแจด้วยวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี จากตารางที่ 4.3 เมื่อค่า $k = 160$ บิต พบว่าสามารถสร้างกุญแจได้ทั้ง 30 ครั้ง เวลาเฉลี่ยที่ใช้ในการสร้างกุญแจ คือ 0.015177 วินาที ค่ามัธยฐาน คือ 0.015886 วินาที จะเห็นว่า ค่าเฉลี่ยมีค่าน้อยกว่าแต่ยังคงใกล้เคียงกับค่ามัธยฐานแสดงให้เห็นว่าข้อมูลมากว่าส่วนมากมีค่ามากกว่าค่าเฉลี่ย ทำให้ข้อมูลมีการแจกแจงแบบเบ้ขวา ส่วนเบี่ยงเบนมาตรฐานมีค่าอยู่ที่ 0.002760 ซึ่งจะเห็นได้ว่าเวลาที่ได้แต่ละครั้งไม่แตกต่างจากค่าเฉลี่ยมาก

ในส่วนของการวัดประสิทธิภาพการสร้างกุญแจวิทยาการหัสลับเอ็นทรูทั้งหมด 30 ครั้ง จะใช้เวลาเฉลี่ย 0.050017 วินาที จากค่ามัธยฐาน เวลาที่ใช้ในการสร้างกุญแจนั้นส่วนมากมีค่าน้อยกว่าค่าเฉลี่ยโดยมีค่ามัธยฐานอยู่ที่ 0.047266 วินาที ทำให้การแจกแจงของข้อมูลเวลามีลักษณะเบ้ซ้าย จากค่าส่วนเบี่ยงเบนมาตรฐาน มีค่าอยู่ที่ 0.011204 จะเห็นได้ว่าเวลาที่ใช้ในการสร้างกุญแจในการทดลองแต่ละครั้ง มีค่าแตกต่างกันค่อนค้างน้อย

4.4 การวัดประสิทธิภาพการเข้ารหัสลับ

ตัวอย่างเซจ 4.5 โปรแกรมวัดประสิทธิภาพขั้นตอนการเข้ารหัสของวิทยาการเข้ารหัสเอ็นทรู

เราสามารถใช้คำสั่งต่อไปนี้ในการวัดประสิทธิภาพของขั้นตอนการเข้ารหัสของวิทยการรหัสลับเอ็นทรู โดยในตัวอย่างนี้จะใช้ค่า $l = 2$ และใช้ตัวแปรเสริมสาธารณะดังค่าที่ได้จากตัวอย่างเซจที่ 4.1 และกุญแจสาธารณะจากตัวอย่างเซจที่ 4.3

```
m = [randrange(0,p-1) for i in range(30)]
m = map_to_R(m)
r = tri_poly(d,d)
t = cputime()
e = encrypt(m, h, r)
ti = cputime(t)
#total_time += ti
print ("Cpu time = " + str(ti))
print ("Plant text : ")
show(m)
print ("Cipher text: ")
show(e)
```

ได้ผลรันดังต่อไปนี้

```
Cpu time = 0.016205
Plant text :
```
$$8x^6 + 5x^5 + 2x^4 + 3x^3 + 4x^2 + 6x + 10$$
```
Cipher text:
```
$$374x^6 + 1298x^5 + 1192x^4 + 1033x^3 + 50x^2 + 841x + 1246$$

ในขั้นตอนนี้เราจะทำการสุ่มข้อความที่จะใช้ในการส่งโดยการสุ่มเลขที่มีช่วงอยู่ระหว่างค่า 0 ถึง p ซึ่งเป็นค่าตัวแปรเสริมสาธารณะจำนวนทั้งหมด N ตัวเพื่อนำไปสร้างเป็นข้อความที่มีดีกรีสูงสุดเท่ากับ $N - 1$ แล้วจึงแปลงเลขที่สุ่มได้ทั้งหมดไปอยู่ในรูปของริงสังวัตนาการ R หลังจากนั้นจึงมาทำการคำนวณหาไซเฟอร์เทกซ์โดยช้ารคำนวณดังตารางที่ 2.2

#

ตัวอย่างเซจ 4.6 โปรแกรมวัดประสิทธิภาพขั้นตอนการเข้ารหัสของวิทยาการเข้ารหัสลับเอ็ลกามอล เส้นโค้งเชิงวงรี

เราสามารถใช้คำสั่งต่อไปนี้ในการจับเวลาขั้นตอนการเข้ารหัสของวิทยาการเข้ารหัสลับเอ็ลกามอล เส้นโค้งเชิงวงรีได้ โดยจะทดสอบด้วยค่า $k = 8$

```
# Create Bob's cipher text (C1,C2)
t = cputime()
# Random Message
point = Ea.gen(0)
n = randrange(1, Eord-1)
Me = n * point  # Plain text member of EllipticCurve
k = randrange(1, Eord - 1) # Create temporary key
C1 = k * P
C2 = Me+(k * Qa)
ti = cputime(t)
total_time += ti
print ("Cpu time = " + str(ti))
print("Bob's Plain Text : " + str(Me))
print("Bob's Cipher Text : " + str((C1,C2)))
```

ได้ผลการรันดังต่อไปนี้

```
Cpu time = 0.015318
Bob's Plain Text : (V^7 + V^6 + V^4 + V^2 + V + 1 : V^5 + V^4
+ V^2 + 1 : 1)
Bob's Cipher Text : ((0 : 1 : 0), (V^7 + V^6 + V^4 + V^2 + V
+ 1 : V^5 + V^4 + V^2 + 1 : 1))
```

ในขั้นแรกนั้น เราจะต้องทำการสุ่มข้อความขึ้นมาก่อนโดยข้อความที่เลือกนั้นจะต้องเป็นสมาชิกของเส้นโค้งเชิงวงรีด้วย เราจึงใช้วิธีการสุ่มตัวเลขขึ้นมาหนึ่งตัวแล้วนำไปคูณกับจุดเป็นเส้นโค้งเชิงวงรีจะทำให้ได้จุดที่ถูกสุ่มขึ้นมา หลังจากนั้นเราจึงนำข้อความที่สุ่มมาไปทำการคำนวณดังขั้นตอนวิธีในตารางที่ 2.1 จะทำให้เราได้ไซเฟอร์เทกซ์ขึ้นมา

#

ตารางที่ 4.3 ผลการวัดประสิทธิภาพของขั้นตอนการเข้ารหัส

ลำดับ	การเข้ารหัส (วินาที)	
	NTRU	ECC
1	0.013396	0.039138
2	0.018864	0.033226
3	0.013884	0.038665
4	0.010790	0.049342
5	0.012598	0.049771
6	0.013885	0.050189
7	0.015332	0.051891
8	0.015716	0.045569
9	0.016013	0.036849
10	0.010794	0.032642
11	0.009897	0.052848
12	0.013481	0.034916
13	0.012540	0.053756
14	0.012654	0.039876
15	0.016393	0.050991
16	0.010157	0.035478
17	0.013521	0.041815
18	0.017738	0.059933
19	0.017227	0.051925
20	0.013176	0.032855
21	0.017722	0.049001
22	0.013611	0.047208
23	0.013489	0.049004
24	0.013922	0.031815
25	0.011943	0.048060
26	0.012754	0.049539
27	0.013636	0.051973

ตารางที่ 4.3 ผลการวัดประสิทธิภาพของขั้นตอนการเข้ารหัส (ต่อ)

28	0.011652	0.046656
29	0.011505	0.049519
30	0.020559	0.049537
ค่าเฉลี่ย	0.013962	0.045133
ค่ามัธยฐาน	0.013505	0.048531
ส่วนเบี่ยงเบนมาตรฐาน	0.002607	0.007662

อภิปรายผล

จากการทดลองการเข้ารหัสด้วยวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี จากตารางที่ 4.3 เมื่อค่า $k = 160$ บิต พบว่าสามารถสร้างกุญแจได้ทั้ง 30 ครั้ง เวลาเฉลี่ยที่ใช้ในการสร้างกุญแจ คือ 0.045133 วินาที ค่ามัธยฐาน คือ 0.048531 วินาที จะเห็นว่า ค่าเฉลี่ยมีค่าน้อยกว่าค่ามัธยฐาน แสดงให้เห็นว่าข้อมูลมากว่าส่วนมากมีค่ามากกว่าค่าเฉลี่ย ทำให้ข้อมูลมีการแจกแจงแบบเบ้ขวา ส่วนเบี่ยงเบนมาตรฐานมีค่าอยู่ที่ 0.007662 ซึ่งจะเห็นได้ว่าเวลาที่ใช้แต่ละครั้งไม่แตกต่างจากค่าเฉลี่ยมาก

ในส่วนของการวัดประสิทธิภาพการสร้างกุญแจวิทยาการหัสลับเอ็นทรูทั้งหมด 30 ครั้ง จะใช้เวลาเฉลี่ย 0.013962 วินาที จากค่ามัธยฐาน เวลาที่ใช้ในการสร้างกุญแจนั้นส่วนมากมีค่าน้อยกว่าแต่ไม่แตกต่างกันมากจากค่าเฉลี่ยโดยมีค่ามัธยฐานอยู่ที่ 0.013505 วินาที จากค่าส่วนเบี่ยงเบนมาตรฐานมีค่าอยู่ที่ 0.002607 จะเห็นได้ว่าเวลาที่ใช้ในการสร้างกุญแจในการทดลองแต่ละครั้ง มีค่าแตกต่างกันค่อนค้างน้อย

4.5 การวัดประสิทธิภาพการถอดรหัสลับ

ตัวอย่างเซจ 4.7 โปรแกรมวัดประสิทธิภาพขั้นตอนการถอดรหัสของวิทยาการเข้ารหัสเอ็นทรู

เราสามารถใช้คำสั่งต่อไปนี้ในการวัดประสิทธิภาพของขั้นตอนการถอดรหัสของวิทยการรหัสลับเอ็นทรู โดยในตัวอย่างนี้จะใช้ค่า $l = 2$ และใช้ตัวแปรเสริมสาธารณะดังค่าที่ได้จากตัวอย่างเซจที่ 4.1 รวมทั้งกุญแจส่วนบุคคลที่ได้จากตัวอย่างเซจที่ 4.3 และไซเฟอร์เทกซ์จากตัวอย่างเซจที่ 4.5

```
t = cputime()
mhat_lift = decrypt(f, F_p, e)
ti = cputime(t)
#total_time += ti
print ("Cpu time = " + str(ti))
print ("Recovery message: ")
show(mhat_lift)
print ("Recovery message in Rp: ")
show(map_to_Rp(mhat_lift))
mhat_lift = map_to_Rp(mhat_lift)
print ("Plain text: ")
show(m)
```

ได้ผลการรันดังต่อไปนี้

```
Cpu time = 0.011482
Recovery message:
```

$$-3x^6 + 5x^5 + 2x^4 + 3x^3 + 4x^2 - 5x - 1$$

```
Recovery message in Rp:
```

$$8x^6 + 5x^5 + 2x^4 + 3x^3 + 4x^2 + 6x + 10$$

```
Plain text:
```

$$8x^6 + 5x^5 + 2x^4 + 3x^3 + 4x^2 + 6x + 10$$

ในขั้นตอนนี้หลังจากที่เราได้รับไซเฟอร์เทกซ์มาแล้ว เราจะทำการถอดรหัสโดยทำการคำนวณดังตารางที่ 2.2 เมื่อเราคำนวณได้เราจะได้เพลนเทกซ์ ซึ่งเพลนเทกซ์ที่ได้นั้นจะยังไม่ตรงกับเพลนเทกซ์จริงๆ เนื่องจากผลลัพธ์ที่ได้นั้นอยู่คนละโครงสร้าง เราจึงต้องทำการเปลี่ยนโครงสร้างเพื่อเปรียบเทียบว่าค่าเพลนเทกซ์ที่ได้นั้นตรงกับเพลนเทกซ์จริงๆ ซึ่งจากผลลัพธ์ จะเห็นได้ว่าเพลนเทกซ์อันที่สองซึ่งเป็นเพลนเทกซ์ที่ได้มีการย้ายโครงสร้างแล้ว จะมีค่าตรงกับเพลนเทกซ์ตั้งต้นจริงๆ

#

ตัวอย่างเซจ 4.8 โปรแกรมวัดประสิทธิภาพขั้นตอนการถอดรหัสของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

เราสามารถใช้คำสั่งต่อไปนี้ในการจับเวลาขั้นตอนการถอดรหัสของวิทยาการเข้ารหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีได้ โดยจะทดสอบด้วยค่า $k = 8$

```
# Alice Decrypt (C1,C2) that is recieved from Bob
t = cputime()
PlainText = C2 - na * C1 # Decrypt message
ti = cputime(t)
total_time += ti
print ("Cpu time = " + str(ti))
print("Derypted : " + str(PlainText))
```

ได้ผลลัพธ์ดังต่อไปนี้

```
Cpu time = 0.000207000000003
Derypted : (V^7 + V^6 + V^4 + V^2 + V + 1 : V^5 + V^4 + V^2 +
1 : 1)
```

เมื่อเราได้รับไซเฟอร์เทกซ์มาแล้ว เราจะทำการคำนวณหาไซเฟอร์เทกซ์โดยใช้วิธีการคำนวณดังตารางที่ 2.1 จะทำให้ได้รับเพลนเทกซ์กลับมาซึ่งจะสังเกตุได้ว่า เพลนเทกซ์นั้นตรงกับเพลนเทกซ์ที่เราได้ทำการสุ่มไว้ในตัวอย่างเซจที่ 4.6

#

ตารางที่ 4.4 ผลการวัดประสิทธิภาพของขั้นตอนการถอดรหัส

ลำดับ	การถอดรหัส (วินาที)	
	NTRU	ECC
1	0.014464	0.011170
2	0.016240	0.010258
3	0.014229	0.011964
4	0.012245	0.016865
5	0.015592	0.015940
6	0.016478	0.016457
7	0.016190	0.017822
8	0.016818	0.014553
9	0.016916	0.011062
10	0.011368	0.011921
11	0.009235	0.012688
12	0.010543	0.012517
13	0.012268	0.016221
14	0.016088	0.013154
15	0.017345	0.019890
16	0.012742	0.013890
17	0.015667	0.012073
18	0.006251	0.019582
19	0.015812	0.016440
20	0.014632	0.017207
21	0.014470	0.016175
22	0.014836	0.015516
23	0.014638	0.015334
24	0.014280	0.010160
25	0.013873	0.015435
26	0.012759	0.015740
27	0.015069	0.011603

ตารางที่ 4.4 ผลการวัดประสิทธิภาพของขั้นตอนการถอดรหัส (ต่อ)

28	0.012191	0.012915
29	0.005379	0.015591
30	0.023236	0.016267
ค่าเฉลี่ย	0.014062	0.014547
ค่ามัธยฐาน	0.014551	0.015385
ส่วนเบี่ยงเบนมาตรฐาน	0.003391	0.002637

อภิปรายผล

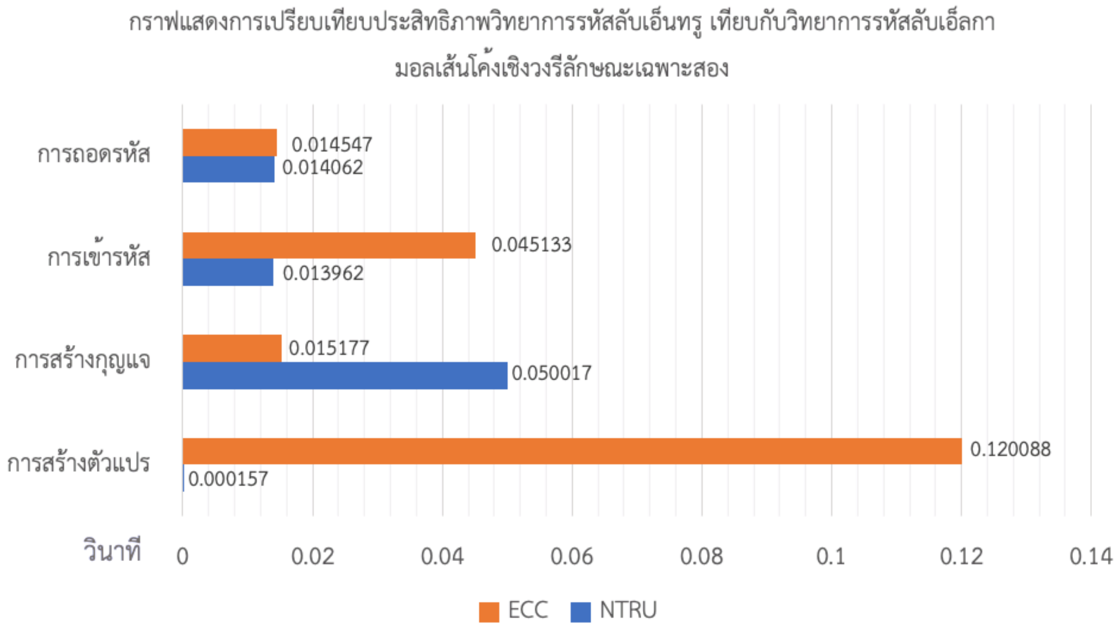
จากการทดลองการถอดรหัสด้วยวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี จากตารางที่ 4.3 เมื่อค่า $k = 160$ บิต พบว่าสามารถถอดรหัสได้ทั้ง 30 ครั้ง เวลาเฉลี่ยที่ใช้ในการถอดรหัสคือ 0.014547 วินาที ค่ามัธยฐาน คือ 0.015385 วินาที จะเห็นว่า ค่าเฉลี่ยมีค่าน้อยกว่า แสดงให้เห็นว่าข้อมูลมากว่าส่วนมากมีค่ามากกว่าค่าเฉลี่ย ทำให้ข้อมูลมีการแจกแจงแบบเบ้ขวา ส่วนเบี่ยงเบนมาตรฐานมีค่าอยู่ที่ 0.002637 ซึ่งจะเห็นได้ว่าเวลาที่ใช้แต่ละครั้งไม่แตกต่างจากค่าเฉลี่ยมาก

ในส่วนของการวัดประสิทธิภาพการถอดรหัสวิทยาการหัสลับเอ็นทรูทั้งหมด 30 ครั้ง จะใช้เวลาเฉลี่ย 0.014062 วินาที จากค่ามัธยฐาน เวลาที่ใช้ในการถอดรหัสนั้นส่วนมากมีค่ามากกว่าค่าเฉลี่ยแต่มีความใกล้เคียงกันมากโดยมีค่ามัธยฐานอยู่ที่ 0.014551 วินาที จากค่าส่วนเบี่ยงเบนมาตรฐาน มีค่าอยู่ที่ 0.003391 จะเห็นได้ว่าเวลาที่ใช้ในการถอดรหัสในการทดลองแต่ละครั้ง มีค่าแตกต่างกันน้อย

จากข้อมูลในตารางที่ 4.1 ถึง 4.4 สรุปข้อมูลการทดลองวัดประสิทธิภาพของวิทยารหัสลับเอ็นทรูและวิทยาการรหัสลับเอ็ลกามอลดังหัวข้อต่อไปนี้ การวัดประสิทธิภาพการสร้างตัวแปรเสริมสาธารณะ แล้ววัดประสิทธิภาพการสร้างกุญแจ จากนั้นวัดประสิทธิภาพการเข้ารหัสลับ แล้ววัดประสิทธิภาพการถอดรหัสลับ ได้ผลของการเปรียบเทียบดังตารางต่อไปนี้

ตารางที่ 4.5 การเปรียบเทียบประสิทธิภาพด้วยค่าเฉลี่ยของวิทยาการรหัสลับเอ็lกามอล และวิทยาการรหัสลับเอ็นทรู

การวัดประสิทธิภาพ	ขนาดพารามิเตอร์ N ของ NTRU	NTRU (วินาที)	ECC (วินาที)
1. การสร้างตัวแปรเสริมสาธารณะ	207.800000	0.000157	0.120088
2. การสร้างกุญแจ	207.800000	0.050017	0.015177
3. การเข้ารหัสลับ	207.800000	0.013962	0.045133
4. ถอดรหัสลับ	207.800000	0.014062	0.014547



รูปที่ 4.1 กราฟแสดงการเปรียบเทียบประสิทธิภาพวิทยาการรหัสลับเอ็นทรู เทียบกับวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีลักษณะเฉพาะสอง

อภิปรายผล

จากตารางที่ 4.5 เป็นตารางที่เปรียบเทียบประสิทธิภาพเวลาที่ใช้ในวิทยาการรหัสลับเอ็ลกา-มอลเส้นโค้งเชิงวงรีและเอ็นทรู โดยค่าในตารางนั้นเป็นค่าเวลาเฉลี่ยของแต่ละขั้นตอน เราจะเห็นได้ว่าเวลาในขั้นตอนการสร้างตัวแปรเสริมสาธารณะนั้น วิทยาการรหัสลับเอ็นทรูจะใช้เวลาที่น้อยกว่ามาก ซึ่งวิทยาการรหัสลับเอ็นทรูใช้เวลาเฉลี่ยอยู่ที่ 0.000157 วินาที ส่วนวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีใช้เวลาเฉลี่ยที่ 0.120088 ในส่วนของขั้นตอนการสร้างกุญแจนั้น วิทยาการรหัสลับเอ็ลกา

มอลเส้นโค้งเชิงวงรีจะใช้เวลาน้อยกว่ามากโดยใช้เวลาเฉลี่ยที่ 0.015177 ส่วนวิทยาการรหัสลับเอ็นทรูนั้นใช้เวลาเฉลี่ย 0.050017 วินาที ในส่วนของการเข้ารหัสลับนั้นเวลาที่ใช้ในการเข้ารหัสของวิทยาการรหัสลับเอ็นทรูจะใช้เวลาน้อยกว่ามากโดยมีค่าเฉลี่ยอยู่ที่ 0.013962 ส่วนเวลาเฉลี่ยของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีใช้คือ 0.045133 ในขั้นตอนการถอดรหัสลับ วิทยาการรหัสลับเอ็นทรูใช้เวลาน้อยกว่าโดยใช้เวลาเฉลี่ย 0.014062 วินาที ส่วนวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีใช้เวลาเฉลี่ย 0.014547 วินาที

บทที่ 5
สรุปผลโครงการและข้อเสนอแนะ

5.1 สรุปผลโครงการ

ในปัจจุบันวิทยาการรหัสลับกุญแจสาธารณะ มีความสำคัญเพิ่มมากขึ้นในระบบสื่อสารอิเล็กทรอนิกส์และการพาณิชย์ ระบบนี้ไม่เพียงแต่ถูกนำไปใช้งานในคอมพิวเตอร์ตั้งโต๊ะเท่านั้น แต่มีการใช้งานแพร่หลายในบัตรสมาร์ตและอุปกรณ์สื่อสารไร้สาย ที่หน่วยความจำและความสามารถในการประมวลผลจำกัด วิทยาการรหัสลับเอ็นทรูเป็นวิทยาการรหัสลับกุญแจสาธารณะที่เพิ่งค้นพบไม่นาน [4] เอ็นทรู อยู่บนพื้นฐานของปัญหาทางคณิตศาสตร์ที่ยาก คือ ปัญหาการหาเวกเตอร์สั้นสุด ในแลตทิซ จัดเป็นปัญหาประเภทเอ็นพีฮาร์ด จุดเด่นของเอ็นทรู คือ มีการคำนวณที่เร็ว มีความปลอดภัยสูง และเป็นวิทยาการรหัสลับหลังควอนตัม ในส่วนของประสิทธิภาพของแผนวิธีลายเซ็นเอ็นทรูและแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรียังไม่มีการศึกษาดังนั้น ในงานวิจัยนี้จะทำการศึกษาเปรียบเทียบประสิทธิภาพระหว่างแผนวิธีลายเซ็นเอ็นทรูและแผนวิธีวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีเหนือเส้นโค้งคอปลิทซ์

โดยในขั้นตอนการออกแบบนั้น ทางผู้จัดทำได้ออกแบบขั้นตอนวิธีการคำนวณเหนือริงสังวัฒนาการ แล้วค่อยบูรณาร่วมกับแผนวิธีรหัสลับเอ็นทรูเพื่อให้การคำนวณในระบบเอ็นทรูมีการใช้งานที่ง่ายและถูกต้องอีกทั้งได้ทำการออกแบบวิธีการทดสอบประสิทธิภาพโดยการพัฒนาโปรแกรมทดสอบการเข้ารหัสลับวิทยาการรหัสลับเอ็นทรูด้วยโปรแกรมเซจ ในส่วนของวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี ทางผู้จัดทำได้นำโปรแกรมการเข้ารหัสและถอดรหัสด้วยวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีที่พัฒนาด้วยโปรแกรมเซจ [9] มาทำการเพิ่มคำสั่งเพื่อใช้ในการจับเวลา

ในส่วนของขั้นตอนการทดลอง ทางผู้จัดทำได้ทำการทดสอบด้วยการเลือกขนาดของกุญแจสาธารณะสำหรับวิทยารหัสลับเอ็ลามอลเส้นโค้งเชิงวงรีอยู่ที่ 160 บิต ทำให้มีบิตระดับความปลอดภัยอยู่ที่ 80 และได้ทำการขนาดของตัวแปรเสริมสาธารณะ N ของวิทยาการรหัสลับเอ็นทรูโดย

การปรับค่า l ให้มีค่าเท่ากับ 7 ซึ่งจะทำให้ได้ค่า N อยู่ในช่วง $[131,257]$ จะทำให้ได้ว่าระดับบิตความปลอดภัยอยู่ในช่วง $[57,88]$ ในการทดสอบทางผู้จัดทำได้ทำการจับเวลาในแต่ละขั้นตอนทั้งหมด 30 ครั้ง เมื่อทำการวัดประสิทธิภาพแล้วพบว่าในขั้นตอนการสร้างตัวแปรเสริมสาธารณะและขั้นตอนการเข้ารหัสนั้น วิทยาการรหัสลับเอ็นทรูมีความเร็วมากกว่าวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีมาก แต่ในส่วนของขั้นตอนการสร้างกุญแจ วิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรีใช้เวลาเฉลี่ยน้อยกว่าเวลาเฉลี่ยของวิทยาการรหัสลับเอ็นทรูมาก ขั้นตอนการถอดรหัสเวลาที่ใช้มีค่าใกล้เคียงกันมาก ในการทดลองของวิทยาการรหัสลับเอ็นทรู จะมีค่า N เฉลี่ยอยู่ที่ 207.8 ในการทดสอบนี้ค่าส่วนเบี่ยงเบนมาตรฐานมีค่าไม่สูงมากในทุกขั้นตอนที่ได้ทำการทดสอบ และข้อมูลเวลาที่ใช้นั้นส่วนมากมีค่าเฉลี่ยใกล้เคียงกับค่ามัธยฐาน

5.2 ข้อเสนอแนะ

1. ในขั้นตอนการเลือกเกณฑ์เพื่อใช้ในการเปรียบเทียบ เนื่องจากวิทยาการรหัสลับเอ็นทรูเป็นวิทยาการรหัสลับที่ค่อนข้างใหม่ ดังนั้นข้อมูลที่ใช้ในการเลือกเกณฑ์ในการเปรียบเทียบประสิทธิภาพโดยให้มีระดับความปลอดภัยใกล้เคียงกันจึงทำได้ยาก
2. ในการทดสอบประสิทธิภาพควรมีการพัฒนาอุปกรณ์เฉพาะขึ้นมาเพื่อทดสอบประสิทธิภาพให้ได้ค่าที่มีความแม่นยำมากที่สุด

เอกสารอ้างอิง

[1] Hoffstein J, Pipher JC, Silverman JH. An introduction to mathematical cryptography. vol. 1. Springer; 2008.

[2] พิเชษฐ เชี่ยวธนะกุล. วิทยาการรหัสลับเชิงคณิตศาสตร์. มหาวิทยาลัยขอนแก่น; 2556.

[3] Lidl R, Pilz G. Applied abstract algebra. Springer Science & Business Media; 2012.

[4] Herstein IN. Abstract algebra. Prentice Hall; 1996.

[5] Lang S. Algebra, volume 211 of Graduate texts in mathematics. Springer-Verlag, New York,; 2002.

[6] Birkhoff G, Mac Lane S. A survey of modern algebra. AK Peters/CRC Press; 1998.

[7] Bronshtein IN, Semendyayev KA, Musiol G, Muehlig H. Tables. In: Handbook of Mathematics. Springer; 2004. p. 1007–1091.

[8] Hankerson D, Menezes AJ, Vanstone S. Guide to elliptic curve cryptography. Springer Science & Business Media; 2006.

[9] ปานดาว แก้วมณี และ ศิวารุจ พรรคชัย. แผนวิธีแบบอสมมาตรด้วยเส้นโค้งเชิงวงรีเหนือฟีลด์ลักษณะ- เฉพาะสอง. รายงานโครงงาน คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น; 2556.

[10] Trappe W, Washington LC. Introduction to cryptography with coding theory. Pearson Education India; 2006.

[11] Nebe G, Profile GA. Boris Venkov’s theory of lattices and spherical designs. Diophantine methods, lattices, and arithmetic theory of quadratic forms. 2013;587:1–19.

[12] Khot S. Hardness of approximating the shortest vector problem in lattices. Journal of the ACM (JACM). 2005;52(5):789–808.

[13] Ajtai M. The shortest vector problem in L2 is NP-hard for randomized reductions. In: Proceedings of the thirtieth annual ACM symposium on Theory of computing. ACM; 1998. p. 10–19.

[14] Stein W. Elementary number theory: primes, congruences, and secrets, Undergraduate Textsin Mathematics. Springer, New York; 2009.

[15] Judson T. Abstract algebra: theory and applications. Stephen F. Austin State University; 2014.

[16] Fraleigh JB. A first course in abstract algebra. Pearson Education India; 2003.

[17] Menezes AJ, Van Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC press; 1996.

[18] Developers T. SageMath. the Sage Mathematics Software System (Version 7.1); 2016.

[19] Howe J, Moore C, O'Neill M, Regazzoni F, G¨uneysu T, Beeden K. Lattice-based encryption over standard lattices in hardware. In: Proceedings of the 53rd Annual Design Automation Conference. ACM; 2016. p. 162.

[20] Yousefi A, Jameii SM. Improving the security of internet of things using encryption algorithms. In: IoT and Application (ICIOT), 2017 International Conference on. IEEE; 2017. p. 1–5.

[21] J. Hoffstein, J. H. Silverman, W. Whyte. NTRU Cryptosystems Technical Report #012, Version 2: Estimated Breaking Times for NTRU Lattices, www.ntru.com

[22] Kerry maletsky. RSA vs ECC Comparison for Embedded Systems. White Paper. 2015

ภาคผนวก

ภาคผนวก ก
โปรแกรมที่ใช้ในการทดลองวิทยาการรหัสลับเอ็นทรู

โปรแกรม ก.1 การสร้างตัวแปรสาธารณะ

```
#--------------------------------------------------------------
# Public parameter generator input = degree k
#--------------------------------------------------------------
def parameter_gen(k):
    flag = True
    while flag :
        N = next_prime(randrange(2**k, 2**(k+1)))
        p = next_prime(randrange(2**k, 2**(k+1)))
        d = randrange(1, N)
        q  =  next_prime(randrange((6 * d  +  1 ) * p,  (6 * d  +
1)*(p*2^5)))
        if gcd(p,q) == 1 :
            if gcd(N,q) == 1 :
                if q > (6*d + 1) * p :
                    flag = False
    return (N,p,q,d)
#--------------------------------------------------------------
l = 7
t = cputime()
(N,p,q,d) = parameter_gen(l)
t1 = cputime(t)
print ("Public parameter : " + str((N,p,q,d)))
print ("Cpu time : " + str(t1))
```

โปรแกรม ก.2 ชุดคำสั่งในการแปลงพหุนามให้อยู่ในริงสังวัตนาการต่างๆ

```
#-------------------------------------------------------------
# Ring Mapping
#-------------------------------------------------------------
# ZZ[x] --> R=ZZ[x]/(x^N-1)
#-------------------------------------------------------------
def map_to_R(a):
    R.<x> = PolynomialRing(ZZ)
    idR=R.ideal(x^N-1)
    QuoR.<x>=R.quotient_ring(idR)
    return QuoR(a)
#-------------------------------------------------------------
# ZZ[x] --> Rp=ZZp[x]/(x^N-1)
#-------------------------------------------------------------
def map_to_Rp(a):
    Rp=IntegerModRing(p)
    Rp.<x>=PolynomialRing(Rp)
    idRp=Rp.ideal(x^N-1)
    QuoRp.<x>=Rp.quotient_ring(idRp)
    return QuoRp(a)
#-------------------------------------------------------------
# ZZ[x] --> Rq=ZZq[x]/(x^N-1)
#-------------------------------------------------------------
def map_to_Rq(a):
    Rq=IntegerModRing(q)
    Rq.<x>=PolynomialRing(Rq)
    idRq=Rq.ideal(x^N-1)
    QuoRq.<x>=Rq.quotient_ring(idRq)
    return QuoRq(a)
```

โปรแกรม ก.3 ชุดคำสั่งในการสร้างพหุนามไตรภาคและการหาพหุนามผกผัน

```
#-------------------------------------------------------------
# Tripolynomial
#-------------------------------------------------------------
def tri_poly(d_1, d_2) :
    s = [1 for j in range(d_1 - 1)]
    s = s + [-1 for j in range(d_2)]
    s = s + [0 for j in range(N - d_2 - d_1)]
    random.shuffle(s)
    s.append(1)
    return map_to_R(s)
#-------------------------------------------------------------
# Fq = f^(-1) in Rq
#-------------------------------------------------------------
def fq_inv(f) :
    return map_to_Rq(f)^(-1)
```

```
#-------------------------------------------------------------
# Fp = f^(-1) in Rp
#-------------------------------------------------------------
def fp_inv(f) :
    return map_to_Rp(f)^(-1)
#-------------------------------------------------------------
def find_h(F_q, g) :
    return map_to_Rq(g)*F_q
```

โปรแกรม ก.4 ชุดคำสั่งในการคำนวณกุญแจสาธารณะ

```
#-------------------------------------------------------------
# find public key
#-------------------------------------------------------------
def find_h(F_q, g) :
    return map_to_Rq(g)*F_q
```

โปรแกรม ก.5 การสร้างกุญแจ

```
#-------------------------------------------------------------
# Key generated by Alice
#-------------------------------------------------------------
t = cputime()
f = tri_poly(d+1, d)
g = tri_poly(d, d)
F_q = fq_inv(f)
F_p = fp_inv(f)
h = find_h(F_q, g)
t2 = cputime(t)
print ("Cpu time : " + str(t2))
print ("Alice's private key (f,F_p): ")
show((f,F_p))
print ("Alice's public key (h): ")
show(h)
```

โปรแกรม ก.6 การเข้ารหัส

```
#---------------------------------------------------------------
# Encryption Function
#---------------------------------------------------------------
def encrypt(m, h, r):
    m=map_to_R(m)
    r=map_to_R(r)
    return map_to_Rq((p*r*h)+m)
#---------------------------------------------------------------
# Bob encrypt message
#---------------------------------------------------------------
m = [randint(0, p-1) for i in range(N)]
m = map_to_R(m)
#---------------------------------------------------------------
# Ephimeral key
#---------------------------------------------------------------
r = tri_poly(d,d)
#---------------------------------------------------------------
t = cputime()
e = encrypt(m, h, r)
t3 = cputime(t)
print ("Cpu time : " + str(t3))
print ("Plant text : ")
show(m)
print ("Cipher text : ")
show(e)
```

โปรแกรม ก.7 การถอดรหัส

```
#--------------------------------------------------------
# Alice decrypt cypher text
#--------------------------------------------------------
t = cputime()
mhat_lift = decrypt(f, F_p, e)
t4 = cputime(t)
print ("Cpu time : " + str(t4))
print ("Recovery message : ")
show(mhat_lift)
print ("Recovery message in Rp : ")
show(map_to_Rp(mhat_lift))
mhat_lift = map_to_Rp(mhat_lift)
print ("Plain text : ")
show(m)
```

ภาคผนวก ข
โปรแกรมที่ใช้ในการทดลองวิทยาการรหัสลับเอ็ลกามอลเส้นโค้งเชิงวงรี

โปรแกรม ข.1 การสร้างตัวแปรสาธารณะ

```
t = cputime()
# Start Timer
S.<V> = GF(2^k)
Sgen = S.gen()
Sord = S.order()
a = randrange(0, Sord-1)
A = Sgen^a
Ea = EllipticCurve(S,[1,A,0,0,1])
Finite Field 2^l
g = Ea.gen(0)
Eord = Ea.order()
n = randrange(1, Eord-1)
P = n*g # Random P point in EllipticCurve
ti = cputime(t)
print ("Cpu time = " + str(ti))
print("Public Varible P : " + str(P))
print("EllipticCurve Order : " + str(Eord))
print("Ea = " + str(Ea))
list(Ea)
```

โปรแกรม ข.2 ชุดคำสั่งในการคำนวณกุญแจสาธารณะ

```
t = cputime()
na = randrange(1, Eord - 1)
Qa = na*P
ti = cputime(t)
print ("Cpu time = " + str(ti))
print("Alice's Private Key : %d" % na)
print("Alice's Public Key : " + str(Qa))
```

โปรแกรม ข.3 การเข้ารหัส

```
point = Ea.gen(0)
n = randrange(1, Eord-1)
Me = n * point  # Plain text member of EllipticCurve
k = randrange(1, Eord - 1) # Create temporary key
C1 = k * P
C2 = Me+(k * Qa)
ti = cputime(t)
print ("Cpu time = " + str(ti))
print C2.parent()
print ("Bob's Plain Text : " + str(Me))
print ("Bob's Cipher Text : " + str((C1,C2)))
```

โปรแกรม ข.4 การถอดรหัส

```
t = cputime()
PlainText = C2 - na * C1 # Decrypt message
ti = cputime(t)
print ("Derypted : " + str(PlainText))
print ("Cpu time = " + str(ti))
```