การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน

วนารัตน์ จุฬพันธ์ทอง

วิทยานิพนธ์เสนอบัณฑิตวิทยาลัย มหาวิทยาลัยนเรศวร เพื่อเป็นส่วนหนึ่งของการศึกษา
หลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มีนาคม 2563

วิทยานิพนธ์ เรื่อง "การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน"
ของ นางสาววนารัตน์ จุฬพันธ์ทอง
ได้รับการพิจารณาให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการสอบวิทยานิพนธ์
(ศาสตราจารย์ ดร.ชิดชนก เหลือสินทรัพย์)

.....ประธานที่ปรึกษาวิทยานิพนธ์
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.พงศ์พันธ์ กิจสนาโยธิน)

.....กรรมการผู้ทรงคุณวุฒิภายใน
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

อนุมัติ

.....
(ศาสตราจารย์ ดร.ไพศาล มุณีสว่าง)
คณบดีบัณฑิตวิทยาลัย
19 ธค 2563

ประกาศคุณูปการ

ผู้วิจัยขอกราบขอบพระคุณ ดร.สุรเดช จิตประไพกุลศาล ประธานที่ปรึกษาวิทยานิพนธ์เป็นอย่างสูงในความกรุณาที่ได้สละเวลามาเป็นที่ปรึกษา พร้อมทั้งให้คำแนะนำอันมีค่าตลอดระยะเวลาในการทำวิทยานิพนธ์ฉบับนี้ และขอกราบขอบพระคุณคณะกรรมการผู้ทรงคุณวุฒิอันประกอบไปด้วย ศาสตราจารย์ ดร.ชิดชนก เหลือสินทรัพย์ ผู้ช่วยศาสตราจารย์ ดร.พงศ์พันธ์ กิจสนาโยธิน และผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล ที่ได้กรุณาให้คำแนะนำตลอดจนแก้ไขข้อบกพร่องของวิทยานิพนธ์ด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์และทรงคุณค่า

เหนือสิ่งอื่นใดขอกราบขอบพระคุณมารดาและขอบคุณน้องสาวของผู้วิจัยที่ให้กำลังใจและให้การสนับสนุนในทุก ๆ ด้านอย่างดีที่สุดเสมอมา

คุณค่าและคุณประโยชน์อันพึงจะมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบและอุทิศแด่ผู้มีพระคุณทุก ๆ ท่าน

วนารัตน์ จุฬพันธ์ทอง

ชื่อเรื่อง การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน
ผู้วิจัย วนารัตน์ จุฬพันธ์ทอง
ประธานที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล
ประเภทสารนิพนธ์ วิทยานิพนธ์ ปร.ด. สาขาวิชาวิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยนเรศวร, 2562
คำสำคัญ การเข้ารหัสลับแบบอสมมาตร จำนวนเต็มแบบเกาส์เซียน

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน โดยมีพื้นฐานมาจากระบบการเข้ารหัสลับ McEliece เพื่อเป็นอีกระบบหนึ่งที่สามารถทนทานต่อการโจมตีโดยอัลกอริทึมควอนตัม โดยมุ่งเน้นนำเสนอวิธีการใหม่โดยใช้รหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติอย่างรหัส Mannheim ซึ่งเป็นรหัสบนฟิลด์ของจำนวนเต็มแบบเกาส์เซียน ผลการวิจัยพบว่ารหัสแบบสองมิติสามารถช่วยเพิ่มจำนวนกุญแจและและเวกเตอร์ความผิดพลาดของข้อความที่ถูกเข้ารหัส โดยที่กุญแจขนาดใกล้เคียงกันของวิธีการเข้ารหัสลับที่นำเสนอเปรียบเทียบกับวิธีการเข้ารหัสลับแบบเดิม วิธีการที่นำเสนอสามารถเพิ่มประสิทธิภาพด้านความปลอดภัย โดยสามารถเพิ่มค่าความซับซ้อนของการโจมตีเพื่อค้นหากุญแจลับ และมีค่าความซับซ้อนใกล้เคียงกันในการโจมตีเพื่อถอดรหัสข้อความ ซึ่งการเพิ่มความซับซ้อนนี้ทำให้ประสิทธิภาพในการประมวลผลของระบบลดลงได้ และเมื่อเปรียบเทียบกับระบบแบบเดิมพบว่ากระบวนการเข้ารหัสและถอดรหัสอาจจะใช้เวลาในการประมวลผลเพิ่มขึ้น อย่างไรก็ตามกระบวนการสร้างกุญแจมีแนวโน้มที่จะใช้เวลาในการประมวลผลน้อยลง

Title	AN ASYMMETRIC CRYPTOGRAPHY USING GAUSSIAN INTEGER
Author	Wanarat Juraphanthong
Advisor	Suradet Jitprapaikulsarn, Ph.D.
Academic Paper	Thesis Ph.D. in Computer Engineering, Naresuan University, 2019
Keyword	Asymmetric Cryptography, Gaussian Integer

ABSTRACT

This thesis extends the McEliece cryptosystem a candidate of post-quantum cryptography, with Gaussian integer. The scheme aims to introduce a novel approach of asymmetric cryptographic scheme with two-dimensional code called Mannheim error correcting code which is the codes over Gaussian integer field. By substituting the one-dimensional linear code with the two-dimensional code employing the finite Gaussian integer, the new system simultaneously increases the key space and the errors to be correct by the codeword decoding. With the same key length compared to classical cryptosystem, this scheme can improve the security performance by increasing both the work factors of key recovery attack and the work factors of decoding attacks. The tradeoff for the higher security level is the higher cost of computation. The running time of encoding and decoding processes of purposed cryptosystem may take longer compared to the classical McEliece cryptosystem. However, the key generation process tends to take less running time that is the benefit for short-term key asymmetric cryptography.

สารบัญ

บทที่ **หน้า**

สารบัญ (ต่อ)

บทที่ **หน้า**

สารบัญ (ต่อ)

บทที่ **หน้า**

สารบัญตาราง

ตาราง **หน้า**

สารบัญตาราง (ต่อ)

ตาราง **หน้า**

สารบัญภาพ

ภาพ **หน้า**

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการประมวลผลควอนตัมเป็นที่กล่าวถึงเป็นอย่างมากในระบบการเข้ารหัสลับแบบอสมมาตร เนื่องจากอัลกอริทึมควอนตัมสามารถโจมตีบนระบบการเข้ารหัสลับที่นิยมใช้ในปัจจุบันได้สำเร็จ ยกตัวอย่างเช่น อัลกอริทึมควอนตัมของ Shor (1997) สามารถโจมตีระบบ RSA (Rivest et al., 1978) และ ElGamal (1985) ได้สำเร็จภายใน Polynomial time ดังนั้นระบบการเข้ารหัสลับที่ทนทานต่อการโจมตีโดยใช้อัลกอริทึมควอนตัม (Post-quantum cryptography) จึงเป็นตัวเลือกของระบบการเข้ารหัสลับที่จะใช้ในอนาคต

การเข้ารหัสลับแบบอสมมาตร McEliece (1978) เป็นระบบที่เป็นตัวเลือกหนึ่งที่ทนทานต่อการโจมตีโดยอัลกอริทึมควอนตัม McEliece มีสมมติฐานความยากในการถอดรหัสกลุ่มรหัสที่ผิดพลาดบนรหัสบล็อกเชิงเส้น จัดอยู่ในกลุ่มความซับซ้อนของปัญหา NP-complete (Kabatiansky et al., 2005) ข้อดีของระบบคือการเข้ารหัสและถอดรหัสข้อความมีความรวดเร็ว (Baldi et al., 2016) ระบบเดิมของ McEliece ใช้รหัสบล็อกเชิงเส้น Goppa เป็นรหัสสำหรับแก้ไขข้อผิดพลาด (Error Correcting Code) ข้อด้อยของระบบเดิมคือขนาดของกุญแจที่มีขนาดใหญ่ทำให้ไม่เป็นที่นิยม จากนั้นจึงมีแนวคิดการปรับปรุงระบบโดยการการแทนที่รหัส Goppa ด้วยรหัสสำหรับแก้ไขข้อผิดพลาดแบบอื่น เช่น การใช้รหัส Generalized Reed-Solomon (GRS) (Niederreiter, 1986) Reed-Muller (Sidelnikov, 1994) Quasi cyclic low density parity check (QC-LDPC) (M. Baldi et al., 2013) Moderate density parity check (MDPC) (Misoczki et al., 2013) และ Polar (Shrestha & Kim, 2014)(R. Hooshmand et al., 2014) เป็นต้น โดยสามารถช่วยลดขนาดของกุญแจลง ซึ่งช่วยเพิ่มระดับความปลอดภัยเมื่อเปรียบเทียบกับขนาดกุญแจของระบบเดิมได้

อย่างไรก็ตามรหัสข้างต้นเป็นรหัสสำหรับแก้ไขข้อผิดพลาดแบบมิติเดียว ผู้วิจัยจึงได้มีแนวคิดใหม่ในการศึกษาและพัฒนาระบบการเข้ารหัสลับแบบอสมมาตรโดยมีพื้นฐานจากระบบ McEliece โดยใช้รหัสบล็อกที่อยู่บนฟิลด์ของจำนวนเต็มแบบเกาส์เซียนอย่างรหัสสำหรับแก้ไขข้อผิดพลาด Mannheim (Huber, 1994) ซึ่งเป็นรหัสแบบสองมิติที่ยังไม่มีผู้นำเสนอมากนัก แทนที่การใช้รหัสแบบเดิมบนมิติเดียว โดยผู้วิจัยได้ศึกษาถึงประสิทธิภาพด้านความปลอดภัยของระบบที่

พัฒนาจากคุณสมบัติของรหัสที่สามารถเพิ่มโดเมนในการสร้างกุญแจและข้อความ รวมทั้งศึกษาประสิทธิภาพด้านประมวลผลสำหรับนำไปใช้งานจริงอีกด้วย

วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและพัฒนาการใช้รหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติบนจำนวนเต็มแบบเกาส์เซียนในการเข้ารหัสลับแบบอสมมาตรบนพื้นฐานระบบ McEliece
2. เพื่อศึกษาประสิทธิภาพด้านความปลอดภัยในการเข้ารหัสลับแบบอสมมาตรบนพื้นฐานระบบ McEliece โดยใช้รหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติบนจำนวนเต็มแบบเกาส์เซียน
3. เพื่อศึกษาประสิทธิภาพด้านการประมวลผลในการเข้ารหัสลับแบบอสมมาตรบนพื้นฐานระบบ McEliece โดยใช้รหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติบนจำนวนเต็มแบบเกาส์เซียน

ขอบเขตของงานวิจัย

การเข้ารหัสลับแบบอสมมาตรในงานวิจัยนี้ ใช้ระบบการเข้ารหัส McEliece เป็นพื้นฐานในการพัฒนาระบบ โดยใช้รหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน คือรหัสสำหรับแก้ไขข้อผิดพลาด Mannheim

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนนี้ ผู้วิจัยได้ทำการศึกษาแนวคิดและทฤษฎีพื้นฐาน และทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้องดังนี้

1. แนวคิดพื้นฐานของการเข้ารหัสลับ
   1.1 การรักษาความน่าเชื่อถือและความปลอดภัยทางสารสนเทศ
   1.2 การเข้ารหัสลับ
   1.3 การวิเคราะห์เพื่อถอดรหัสลับ
2. การเข้ารหัสลับแบบอสมมาตร
   2.1 การปรับปรุงประสิทธิภาพของการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน
   2.2 การเข้ารหัสลับแบบอสมมาตรที่ทนทานต่อการประมวลผลควอนตัม
3. ระบบการเข้ารหัสลับแบบอสมมาตร McEliece และการโจมตี
   3.1 ระบบการเข้ารหัสลับแบบอสมมาตร McEliece
   3.2 การโจมตีบนระบบการเข้ารหัสลับแบบอสมมาตร McEliece
4. การใช้รหัสแบบอื่นบนระบบการเข้ารหัสลับแบบอสมมาตร McEliece
5. รหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน

แนวคิดพื้นฐานของการเข้ารหัสลับ

1. การรักษาความน่าเชื่อถือและความปลอดภัยทางสารสนเทศ

การรักษาความน่าเชื่อถือและความปลอดภัยทางสารสนเทศนั้น เริ่มต้นจาก CIA-triad ซึ่งเป็นแนวคิดพื้นฐานในการรักษาความปลอดภัยทางคอมพิวเตอร์และสารสนเทศ ประกอบด้วย 1) Confidentiality คือ การรักษาความลับ 2) Integrity คือ การรักษาความถูกต้องและสมบูรณ์ และ 3) Availability คือ การพร้อมใช้งาน

ต่อมานักวิจัยได้นำแนวคิดนี้มาประยุกต์ใช้กันอย่างแพร่หลายในการจัดการเกี่ยวกับความปลอดภัยทางสารสนเทศจนถึงปัจจุบัน ดังเช่น McCumber (2004) พัฒนาโมเดลในการรักษาความปลอดภัยทางสารสนเทศเรียกว่า McCumber's cube โดยนำ CIA-triad ไปใช้เป็น

information characteristic ต่อมา Parker (Bosworth & Kabay, 2002) นำเสนอโมเดลใหม่เพื่อใช้ในการรักษาความปลอดภัยทางสารสนเทศ โดยได้เพิ่มมิติเป็น 6 ด้าน เรียกว่า Parkerian-hexad ประกอบด้วยมิติเดิม 1) Confidentiality 2) Integrity 3) Availability และมิติใหม่คือ 4) Utility 5) Authenticity และ 6) Possession อีกโมเดลหนึ่งซึ่งนำเสนอโดย Maconachy (2001) ได้เพิ่มมิติอีก 2 ด้าน จาก CIA-triad ได้แก่ Authentication และ Non-repudiation ต่อมา Schneier (2006) นำเสนอโมเดล Pentagon of Trust เป็นโมเดลรักษาความปลอดภัยที่ประกอบด้วย 5 ขั้นตอน ได้แก่ 1) Authentication 2) Authorization 3) Availability 4) Authenticity และ 5) Admissibility นอกจากนี้ Cherdantseva และ Hilton (2013) นำเสนอ RMIAS โมเดลสำหรับรักษาความน่าเชื่อถือและความปลอดภัยทางสานสนเทศ โดยได้แบ่งเป้าหมายของการรักษาความปลอดภัยออกเป็น 8 มิติ ซึ่งเป็นการผสมผสานระหว่างโมเดลข้างต้น ประกอบด้วย CIA-triad คือ 1) Confidentiality คือ ระบบสามารถรักษาความลับของข้อมูล โดยอนุญาตเฉพาะผู้ที่มีสิทธิ์สามารถเข้าถึงข้อมูลได้เท่านั้น 2) Integrity คือ ระบบสามารถรักษาความสมบูรณ์ของข้อมูล โดยป้องกันการเปลี่ยนแปลงข้อมูลโดยที่ไม่ได้รับอนุญาต 3) Availability คือ ระบบสามารถใช้งานได้เมื่อผู้ที่มีปฏิสัมพันธ์กับระบบ (ที่ได้รับการอนุญาต) ร้องขอเพื่อเข้าถึงระบบ และมิติอื่นอีก 5 มิติคือ 4) Accountability คือ ความสามารถของระบบในการระบุผู้ใช้ที่กระทำการต่อข้อมูลหรือคอมพิวเตอร์ 5) Auditability คือ ความสามารถในการตรวจสอบการกระทำว่ามีการกระทำใดบ้างเกิดขึ้นบนระบบ 6) Authenticity คือ ความสามารถของระบบในการตรวจสอบตัวตนและความน่าเชื่อถือของผู้ที่มีปฏิสัมพันธ์กับระบบ เช่น ผู้ใช้ ข้อมูล กระบวนการ ซอฟต์แวร์ ฮาร์ดแวร์ และเครือข่าย เป็นต้น 7) Non-repudiation คือ ระบบสามารถป้องกันการปฏิเสธว่าไม่ได้มีกระทำนั้นเกิดขึ้น หรือป้องกันการอ้างที่เป็นเท็จว่ามีกระทำการนั้นเกิดขึ้น 8) Privacy คือ ระบบมีนโยบายความเป็นส่วนตัวสำหรับข้อมูลของผู้ใช้ และสามารถให้ผู้ใช้ควบคุมหรือปรับเปลี่ยนการแสดงข้อมูลส่วนตัวเหล่านั้นได้ ดังสรุปในตาราง 1

มาตรการความปลอดภัย (Security countermeasure) เป็นองค์ประกอบของโมเดลการรักษาความน่าเชื่อถือและความปลอดภัยทางสารสนเทศ เพื่อบรรลุเป้าหมายของการรักษาความปลอดภัยที่กำหนดไว้ โดยสามารถแบ่งออกเป็นหลายประเภท เช่น มาตรการที่ใช้วิธีการทางกฎหมาย มาตรการที่ใช้วิธีการจัดการบุคคล มาตรการที่ใช้วิธีการการจัดการองค์กร และมาตรการที่ใช้วิธีการทางเทคนิค เป็นต้น ซึ่งวิทยาการเข้ารหัสลับ (Cryptology) เป็นวิธีการทางเทคนิควิธีหนึ่งที่นำมาใช้เพื่อรักษาความน่าเชื่อถือและความปลอดภัยของข้อมูลที่แลกเปลี่ยนกันระหว่างระบบหรือเครือข่าย โดยความหมายดั้งเดิมมาจากรากศัพท์ที่แปลว่า "Hidden" และ "Word" หมายถึง

การซ่อนความหมายของคำเพื่อรักษาความลับ แต่ในปัจจุบันหากกล่าวถึงวิทยาการเข้ารหัสลับจะหมายถึงการนำวิทยาศาสตร์ทางคณิตศาสตร์มาศึกษาเกี่ยวกับการเข้ารหัสลับ (Cryptography) และการวิเคราะห์เพื่อถอดรหัสลับ (Cryptanalysis) ซึ่งจะกล่าวถึงในหัวข้อถัดไป

ตาราง 1 เป้าหมายในการรักษาความมั่นคงและปลอดภัยของสารสนเทศของโมเดล

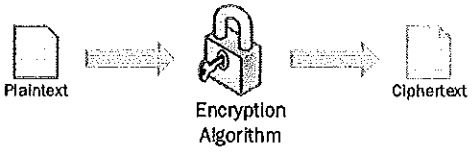
โมเดล	เป้าหมายในการรักษาความมั่นคงและปลอดภัยของสารสนเทศ												
	Confidentiality	Integrity	Availability	Utility	Authenticity	Possession	Authentication	Authorization	Admissibility	Accountability	Auditability	Non-repudiation	Privacy
1. CIA-triad	✓	✓	✓										
2. McCumber [2]	✓	✓	✓										
3. Parkerian-hexad	✓	✓	✓	✓	✓	✓							
4. Maconachy	✓	✓	✓				✓					✓	
5. Pentagon of Trust			✓		✓		✓	✓	✓				
6. RMIAS	✓	✓	✓		✓					✓	✓	✓	✓

2. การเข้ารหัสลับ

การเข้ารหัสลับ (Cryptography) (Oppliger, 2011) แบ่งระบบออกเป็น 3 ประเภท ซึ่งลักษณะสำคัญของแต่ละระบบมีดังนี้

2.1 Unkeyed cryptography

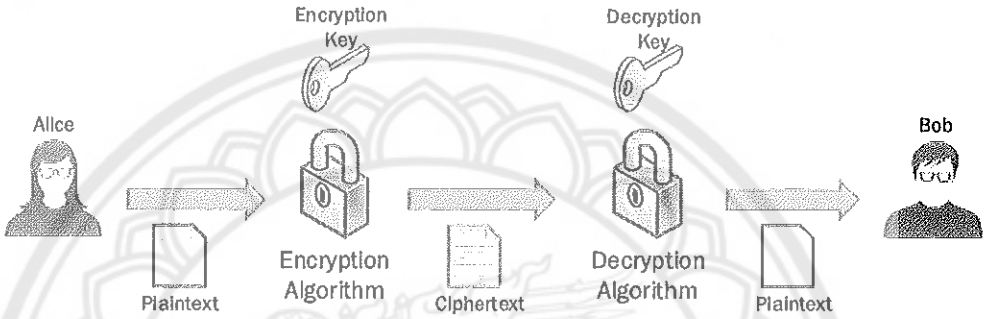
คือระบบการเข้ารหัสที่ไม่ได้ใช้พารามิเตอร์ลับ (Secret key) ดังภาพ 1 ซึ่งเป็นการเข้ารหัสข้อความเป็นข้อความลับ ด้วยวิธีการที่ไม่สามารถหรือยากต่อการถอดรหัสข้อความกลับได้ เช่น การเข้ารหัสแบบทางเดียว (One-way function) การเข้ารหัสแบบแฮช (Hash function) โดยตัวอย่างมาตรฐานที่ใช้ระบบ Unkeyed ได้แก่ MD5, SHA-1 และ CRC32 เป็นต้น



ภาพ 1 Unkeyed cryptography

2.2 **การเข้ารหัสลับแบบสมมาตร** (Symmetric cryptography)

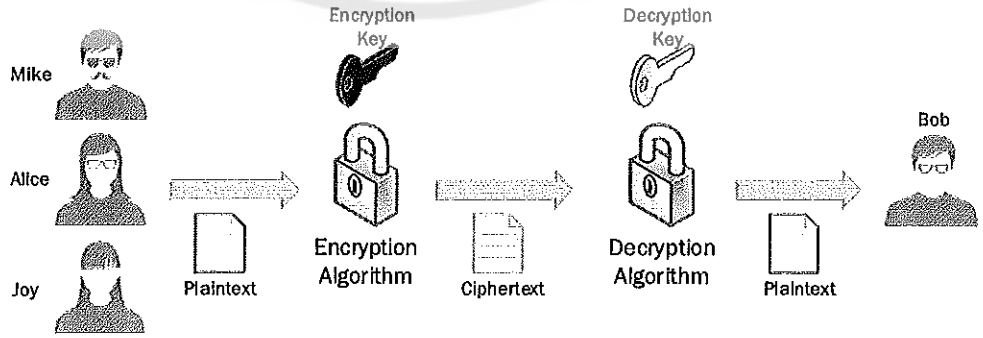
คือระบบการเข้ารหัสที่ใช้และแลกเปลี่ยนพารามิเตอร์ลับระหว่างผู้ใช้ ซึ่งจะใช้กุญแจดอกเดียวกันในการเข้ารหัสข้อความเป็นข้อความลับ และถอดรหัสข้อความลับกลับเป็นข้อความ ทำให้ผู้ที่ไม่มีกุญแจไม่สามารถเข้าใจข้อความลับได้ ดังภาพ 2 โดยตัวอย่างมาตรฐานที่ใช้ระบบ Symmetric ได้แก่ DES, 3DES และ AES เป็นต้น



ภาพ 2 Symmetric cryptography

2.3 **การเข้ารหัสลับแบบอสมมาตร** (Asymmetric cryptography)

คือระบบการเข้ารหัสลับที่ใช้พารามิเตอร์ลับแต่ไม่ได้แลกเปลี่ยนกันระหว่างผู้ใช้ กล่าวคือผู้ใช้ถือกุญแจคนละดอก โดยเผยแพร่กุญแจสาธารณะ (Public key) เพื่อใช้ในการการเข้ารหัสข้อความเป็นข้อความลับ จากนั้นจะใช้ส่วนกุญแจลับ (Private key) ถอดรหัสข้อความลับกลับเป็นข้อความ ซึ่งทำให้มีเพียงผู้ที่มีกุญแจลับเท่านั้นสามารถเข้าใจข้อความลับได้ ดังภาพ 3 โดยตัวอย่างมาตรฐานที่ใช้ระบบ Asymmetric ได้แก่ DSS และ RSA เป็นต้น



ภาพ 3 Asymmetric cryptography

จากระบบการเข้ารหัสข้างต้น ระบบประเภท Unkeyed นำไปใช้สำหรับการเข้ารหัสที่ไม่ต้องการการถอดรหัสกลับ เช่น การเข้ารหัส MD5 บนรหัสผ่านสำหรับตรวจสอบการลงชื่อเข้าใช้ระบบ เป็นต้น ส่วนระบบประเภท Symmetric และ Asymmetric นำไปใช้ในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตที่ต้องมีการเข้ารหัสและถอดรหัสข้อความกลับ เช่น การประยุกต์ใช้ในระบบลายเซ็นดิจิตอลเพื่อตรวจสอบตัวตนที่แท้จริงของผู้ส่งข้อความ เป็นต้น ส่วนปัญหาในเรื่องเวลาในการเข้ารหัสหรือถอดรหัสข้อความระบบประเภท Unkeyed และ Symmetric จะใช้เวลาน้อยกว่าแบบ Asymmetric เนื่องจากใช้อัลกอริทึมที่มีความซับซ้อนน้อย อย่างไรก็ตามปัญหาของระบบ Symmetric คือความยุ่งยากในการแลกเปลี่ยนกุญแจระหว่างกัน รวมถึงปัญหาในการรักษาความปลอดภัยเนื่องจากกุญแจมีจำนวนมากขึ้นเมื่อจำเป็นต้องแลกเปลี่ยนข้อมูลกับผู้ใช้หลายกลุ่ม หากใช้ระบบแบบ Asymmetric จะลดปัญหานี้ได้ เนื่องจากสามารถใช้กุญแจเพียงคู่เดียว ก็สามารถแลกเปลี่ยนข้อมูลกับผู้ใช้หลายกลุ่มได้ ดังสรุปในตาราง 2

ตาราง 2 ข้อดีของการเข้ารหัสลับแบบต่าง ๆ

ข้อดี	การเข้ารหัสลับ (Cryptography)		
	Unkeyed	Symmetric	Asymmetric
1. การเข้ารหัส/ถอดรหัสข้อมูลใช้เวลาน้อย	✓	✓	✗
2. ถอดรหัสข้อความลับกลับเป็นข้อความได้	✗	✓	✓
3. ไม่มีการแลกเปลี่ยนกุญแจระหว่างผู้ใช้	✓	✗	✓

การนำการเข้ารหัสลับมาใช้ในการจัดการข้อมูล สามารถรักษาคุณสมบัติเพื่อบรรลุเป้าหมายในการรักษาความมั่นคงและปลอดภัยของสารสนเทศได้ ดังนี้

1. ความสามารถในการรักษาความลับ (Confidentiality) คือ ระบบสามารถรักษาความลับของข้อมูล โดยอนุญาตเฉพาะผู้ที่มีสิทธิ์ เช่น ใน Symmetric cryptosystem ผู้ที่มีกุญแจเท่านั้นที่สามารถเข้าถึงข้อมูลได้

2. ความสามารถในการรักษาความสมบูรณ์ (Integrity) คือ ระบบสามารถรักษาความสมบูรณ์ของข้อมูล ซึ่งป้องกันการเปลี่ยนแปลงข้อมูลโดยที่ไม่ได้รับอนุญาต เช่น ใน Unkeyed cryptography โดยใช้ Hash function ตรวจสอบความสมบูรณ์ของข้อมูลว่า ข้อมูลนั้นไม่ได้ถูกแก้ไขจากข้อมูลตั้งต้นเมื่อไปถึงผู้รับ

3. ความสามารถในการตรวจสอบความน่าเชื่อถือ (Authenticity) คือ ความสามารถของระบบในการตรวจสอบตัวตนและความน่าเชื่อถือของผู้ที่มีปฏิสัมพันธ์กับระบบ เช่น ใน

Symmetric cryptography จะสามารถตรวจสอบตัวตนและความน่าเชื่อถือจากกุญแจที่ผู้ใช้นั้นถืออยู่

4. ความสามารถในการป้องกันการปฏิเสธ (Non-repudiation) คือ ระบบสามารถป้องกันการปฏิเสธว่าไม่ได้มีกระทำนั้นเกิดขึ้น หรือป้องกันการอ้างที่เป็นเท็จว่ามีกระทำการนั้นเกิดขึ้น เช่น ใน Asymmetric cryptography จะมีเพียงผู้ที่ถือกุญแจลับเท่านั้น ที่สามารถเข้าถึงข้อมูลได้ ทำให้ไม่สามารถปฏิเสธได้ว่าไม่ได้เป็นผู้เข้าถึงข้อมูล

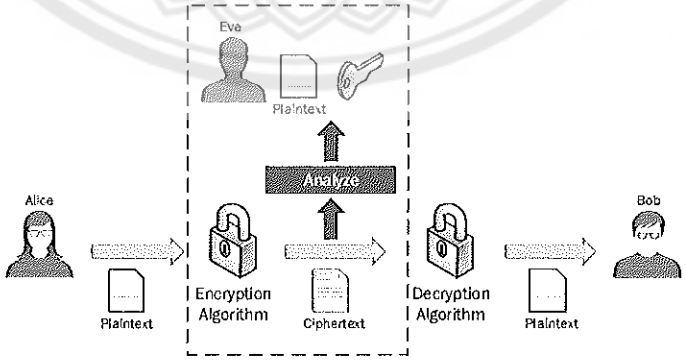
อย่างไรก็ตามการเข้ารหัสลับอาจจะสูญเสียคุณสมบัติข้างต้นไป หากผู้โจมตีสามารถถอดรหัสข้อความลับได้โดยใช้ศาสตร์การวิเคราะห์เพื่อถอดรหัสลับ

3. **การวิเคราะห์เพื่อถอดรหัสลับ**

นอกจากการโจมตีโดยการใช้กุญแจทุก ๆ รูปแบบเพื่อถอดรหัสลับ ที่เรียกว่า Brute-force attack แล้ว ยังมีศาสตร์ที่ใช้การวิเคราะห์เพื่อถอดรหัสลับ (Cryptanalysis) (Wagstaff, 2002) โดยเป็นศึกษาและวิเคราะห์การเข้ารหัสลับเพื่อใช้ในการโจมตีเพื่อถอดรหัสข้อความลับ ซึ่งสามารถแบ่งออกเป็น 3 ประเภท ตามการทราบข้อมูลข้อความของผู้โจมตี ได้แก่

3.1 Ciphertext-only attack

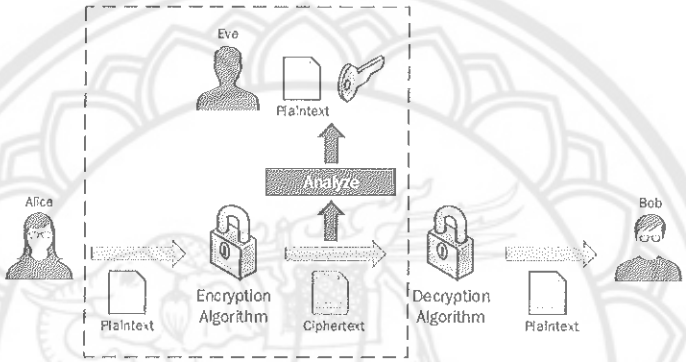
เป็นวิธีการโจมตีที่ผู้โจมตีจะทราบข้อความที่เข้ารหัสไว้เพียงอย่างเดียวเท่านั้น ซึ่งโดยทั่วไปจะถือว่าผู้โจมตีทราบวิธีการเข้ารหัสด้วย ดังภาพ 4 การโจมตีประเภทนี้จะยากที่สุดจากทั้งสามประเภท ตัวอย่างวิธีการคือ ผู้โจมตีจะอาศัยการเข้าถึงข้อความที่ถูกเข้ารหัสโดยใช้อัลกอริทึมเดียวกันหลาย ๆ ข้อความ และพยายามถอดรหัสข้อความให้ได้มากที่สุด เพื่อจะค้นหากุญแจที่ใช้ถอดรหัสข้อความ



ภาพ 4 Ciphertext-only attack

3.2 Known-plaintext attack

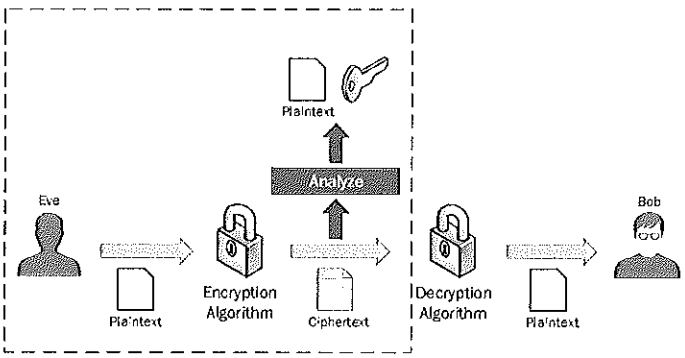
เป็นวิธีการโจมตีที่นอกจากผู้โจมตีจะทราบข้อความที่เข้ารหัสและวิธีการเข้ารหัส ผู้โจมตียังทราบข้อความต้นฉบับบางส่วน ดังภาพ 5 ทำให้การโจมตีประเภทนี้มีประสิทธิภาพ ตัวอย่างวิธีการคือ การวิเคราะห์ส่วนหัวของข้อความที่ถูกเข้ารหัส ซึ่งจะเหมือนเดิมทุกครั้ง (เป็นมาตรฐานของระบบนั้น) ทำให้ผู้โจมตีถอดรหัสข้อความส่วนหัวได้ และพยายามค้นหากุญแจจากข้อความที่ถูกถอดรหัสได้นี้



ภาพ 5 Known-plaintext attack

3.3 Chosen-plaintext attack

เป็นการโจมตีที่ผู้โจมตีสามารถเลือกข้อความที่จะเข้ารหัสได้ ดังภาพ 6 ตัวอย่างวิธีการคือ ใช้เทคนิคหรือวิธีการบางอย่างเพื่อพยายามจะค้นหากุญแจ และตรวจสอบความถูกต้องของกุญแจได้แม้จะใช้ข้อความที่ไม่มีความหมายก็ตาม ซึ่งในระบบการเข้ารหัสลับแบบอสมมาตรส่วนมากจะใช้วิธีการนี้ในการค้นหากุญแจเพื่อถอดรหัสข้อความ



ภาพ 6 Chosen-plaintext attack

การเข้ารหัสลับแบบอสมมาตร

1. การปรับปรุงประสิทธิภาพของการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน

การเข้ารหัสลับแบบอสมมาตรได้ถูกคิดค้นและนำเสนอมาจนถึงปัจจุบัน โดยแบ่งประเภทโดยอาศัยพื้นฐานของปัญหาหรือสมมติฐานความยากที่กำหนดความซับซ้อนของระบบ เช่น Integer-factorization-based Discrete-logarithm-based และ Elliptic-curve-discrete-logarithm-based เป็นต้น

RSA (Rivest et al., 1978) นำเสนอโดย Revist Shamar และ Adleman เป็นระบบการเข้ารหัสหนึ่งที่จัดอยู่ในประเภท Integer factorization based ซึ่งอยู่บนพื้นฐานปัญหาความยากในการแยกตัวประกอบของจำนวนเฉพาะ (IFP) ต่อมามีการปรับปรุงประสิทธิภาพของระบบให้ดียิ่งขึ้นโดยการประยุกต์ใช้จำนวนเต็มแบบเกาส์เซียน ดังเช่น Pradhan (2013) นำเสนออัลกอริทึมโดยใช้จำนวนเต็มแบบเกาส์เซียนแทนที่การใช้จำนวนเต็มบนระบบเข้ารหัสแบบ RSA ซึ่งผลจากการปรับเปลี่ยนตัวแปรนี้ พบว่าช่วยให้ระบบมีความปลอดภัยมากขึ้นเมื่อเปรียบเทียบกับระบบเดิม เนื่องจากมีขอบเขตที่มากขึ้นในการสร้างข้อความเพื่อเข้ารหัส อีกทั้งยังเพิ่มจำนวนกุญแจให้เลือกใช้ได้มากขึ้น ทำให้ผู้โจมตีต้องใช้ความพยายามมากขึ้นในการถอดรหัสข้อความและค้นหากุญแจ ต่อมาระบบการเข้ารหัสประเภท Discrete logarithm based เช่น ElGamal (1985) อยู่บนพื้นฐานความยากในการแก้ปัญหา Discrete logarithm (DLP) มีการนำจำนวนเต็มแบบเกาส์เซียนมาปรับปรุงประสิทธิภาพเช่นกัน ดังเช่น Koval (2016) ได้นำเสนออัลกอริทึมการยกกำลังจำนวนเต็มแบบเกาส์เซียนสำหรับนำไปใช้ในระบบการเข้ารหัสแบบ Discrete logarithm based ซึ่งออกแบบฟังก์ชันการยกกำลังโดยใช้ความสัมพันธ์ระหว่างจำนวนเต็มแบบเกาส์เซียนและ Lucas sequences ผลการทดลองพบว่าสามารถเพิ่มความเร็วในการคำนวณ 34 % เมื่อเปรียบเทียบกับอัลกอริทึมเดิมที่เป็นการยกกำลังจำนวนเต็ม นอกจากนี้ Elkamchouchi et al. (2003) ยังได้นำเสนอเทคนิคใหม่ในการเข้ารหัสแบบอสมมาตรโดยใช้การผสมผสานระหว่าง Integer factorization problem และ Discrete logarithm problem บนโดเมนของจำนวนเต็มแบบเกาส์เซียน ซึ่งเทคนิคนี้ช่วยให้มีความปลอดภัยมากขึ้นและช่วยลดความซับซ้อนในการคำนวณเมื่อเปรียบเทียบกับการเข้ารหัสแบบ RSA และ ElGamal ระบบสุดท้าย ECC (Miller, 1985) ซึ่งเป็นระบบการเข้ารหัสประเภท Elliptic curve discrete logarithm based ซึ่งมีความซับซ้อนของระบบอยู่บนพื้นฐานความยากในการแก้ปัญหาด้วย elliptic curve discrete logarithm (ECDLP) ต่อมา Mohamed & Elkamchouchi (2009) ได้นำเสนอวิธีการเพื่อปรับปรุงระบบ ECC โดยใช้จำนวนเต็มแบบเกาส์เซียนแทนที่จำนวนเต็มแบบเดิม จำนวนของจุดที่สามารถใช้ได้มากขึ้นบน Elliptic curve

ทำให้ระบบมีความปลอดภัยมากขึ้น อย่างไรก็ตามกุญแจที่เป็นจำนวนเต็มแบบเกาส์เซียนต้องใช้พื้นที่มากขึ้นในการจัดเก็บเช่นกัน

2. การเข้ารหัสลับแบบอสมมาตรที่ทนทานต่อการประมวลผลควอนตัม

การเข้ารหัสลับแบบอสมมาตรที่ทนทานต่อการประมวลผลควอนตัม (Post-quantum cryptography) ถูกกล่าวถึงเป็นอย่างมากในปัจจุบัน เนื่องจากการมาถึงของอัลกอริทึมควอนตัมอย่าง Shor ที่สามารถโจมตีระบบการเข้ารหัสลับบนพื้นฐานปัญหา IFP DLP และ ECDLP ข้างต้นได้ การเข้ารหัสแบบอสมมาตรที่ทนทานต่อการประมวลผลควอนตัม สามารถแบ่งประเภทโดยอาศัยพื้นฐานของปัญหาหรือสมมติฐานความยากเช่นเดียวกัน เช่น Multivariate-based Hash-based Supersingular-elliptic-curve-based Lattice-based และ Code-based (Yan, 2013) (Trappe & Washington, 2006) เป็นต้น

ระบบการเข้ารหัสลับแบบ Multivariate-based อยู่บนพื้นฐานความยากในการแก้ปัญหาสมการพหุนามหลายตัวแปรบนเซตจำกัด จัดอยู่ในกลุ่มความซับซ้อนของปัญหา NP-hard ตัวอย่างคือ Hidden field equation (HFE) นำเสนอโดย Patarin (1996) ซึ่งเป็นการสร้างกุญแจสาธารณะจากสมการพหุนามที่ยากต่อการอินเวิร์ส หากไม่ทราบสมการพหุนามที่เป็นกุญแจลับ ระบบมีข้อจำกัดคือความซับซ้อนในการคำนวณสมการพหุนามหลายตัวแปร จึงมีการปรับปรุงระบบโดยลดความซับซ้อนเพื่อให้การเข้ารหัสและถอดรหัสมีความรวดเร็วขึ้น เช่น การใช้ Groebner basis เป็นต้น ต่อมาระบบแบบ Hash-based ใช้คุณสมบัติของฟังก์ชันแฮช ซึ่งอยู่บนพื้นฐานความยากในการหาค่าข้อความนำเข้าที่แท้จริงจากค่าแฮชที่สร้างโดยระบบ แต่เดิมระบบไม่ค่อยเป็นที่นิยมมากนักเนื่องจากข้อจำกัดของจำนวนกุญแจสาธารณะที่สร้างได้บนระบบลายเซ็นดิจิทัล อย่างไรก็ตามความสามารถที่ทนทานต่อการประมวลผลควอนตัมทำให้ระบบ Hash-based กลับมาเป็นทางเลือกที่น่าสนใจอีกครั้ง ตัวอย่างระบบคือ ระบบลายเซ็นดิจิทัล SPHINCS ถัดมาระบบแบบ Supersingular-elliptic-curve-based มีความซับซ้อนของระบบอยู่บนพื้นฐานความยากในการแก้ปัญหาด้วย Supersingular elliptic curve discrete logarithm จัดอยู่ในกลุ่มความซับซ้อนของปัญหา NP-hard ซึ่งเป็นระบบที่ปรับปรุงมาจาก ECC สำหรับระบบการแลกเปลี่ยนกุญแจ เพื่อให้ทนทานต่อการโจมตีโดยใช้อัลกอริทึมควอนตัม ตัวอย่างระบบ เช่น Supersingular-ECC-Diffie-Hellman

ระบบการเข้ารหัสลับประเภท Lattice-based อยู่บนพื้นฐานความยากในการหา Shortest vector (SVP) บนกุญแจลับของระบบ ซึ่งอยู่ในกลุ่มความซับซ้อนของปัญหา NP-hard ตัวอย่างของระบบคือ NTRU นำเสนอโดย Hoffstein et al. (1998) ระบบ NTRU อยู่บนพื้นฐาน

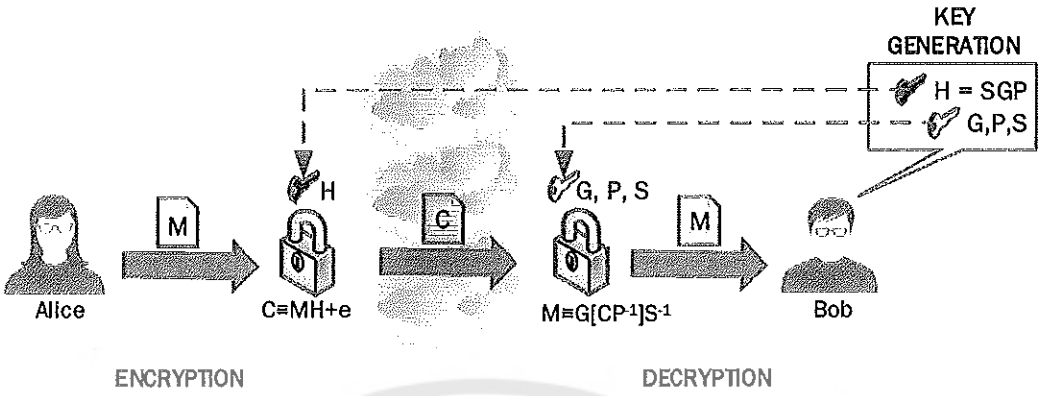
ของ lattice โดยสมการพหุนามดีกรี N-1 ที่มีสัมประสิทธิ์เป็นจำนวนเต็มบนริง $R=Z[x]/(X^N-1)$ การเพิ่มระดับความปลอดภัยบนระบบเป็นหนึ่งในจุดประสงค์ที่สำคัญของเกือบทุกระบบการเข้ารหัส ใน NTRU การเพิ่มดีกรีของสมการพหุนามช่วยเพิ่มระดับความปลอดภัยได้ แต่นั่นทำให้ต้องใช้เวลามากขึ้นในการคำนวณ นักวิจัยพยายามแก้ปัญหาดังกล่าวโดยการปรับปรุงอัลกอริทึมในการหาอินเวิร์สของสมการพหุนาม (Zhao & Su, 2011) (Nyokabi et al., 2017) การใช้เมทริกซ์แทนสมการพหุนาม (Nayak et al., 2012) เป็นต้น ต่อมามีการนำเสนอระบบ NTWO ที่เปลี่ยนไปใช้สมการพหุนามสองตัวแปร (Caboara et al., 2008) ซึ่งทำให้ยากขึ้นในการคาดเดากุญแจลับ การปรับปรุงระบบแบบอื่นที่ช่วยเพิ่มระดับความปลอดภัยคือการใช้สัมประสิทธิ์เป็นจำนวนประเภทอื่นแทนที่จำนวนเต็ม เช่น Nanda et al. (2015) ใช้จำนวนเต็มแบบเกาส์เซียนและใช้การคำนวณบนเมทริกซ์แทนสมการพหุนามบน NTRU วิธีการนี้เพิ่มฟิล์ดในการสร้างกุญแจ ทำให้ผู้โจมตีต้องใช้ความพยายามมากขึ้นในการคาดเดากุญแจ แต่ก็เพิ่มความซับซ้อนในการคำนวณจากการใช้จำนวนเต็มแบบเกาส์เซียน อย่างไรก็ตามการใช้เมทริกซ์ช่วยลดความซับซ้อนบนระบบลงได้ นอกจากจำนวนเต็มแบบเกาส์เซียนแล้ว มีการนำเสนอจำนวนประเภทอื่นเพื่อปรับปรุง NTRU เช่น การใช้จำนวน Kleinian (Thakur et al., 2017) Eisenstein (Jarvis & Nevins, 2015) Quaternion (Bagheri et al., 2017) และ Octonion (Malekian & Zakerolhosseini, 2010) (Bagheri & Sadeghi, 2015) เป็นต้น

ระบบการเข้ารหัสประเภท Code based เป็นระบบการเข้ารหัสลับที่อยู่บนพื้นฐานความยากในการถอดรหัสกลุ่มรหัสบนรหัสบล็อกเชิงเส้น (Syndrome decoding problem : SDP) เช่น ระบบการเข้ารหัส McEliece ซึ่งเป็นระบบการเข้ารหัสที่ผู้วิจัยใช้เป็นพื้นฐานในการพัฒนาระบบการเข้ารหัสลับ โดยจะกล่าวถึงในหัวข้อถัดไป

ระบบการเข้ารหัสลับแบบอสมมาตร McEliece และการโจมตี

1. ระบบการเข้ารหัสลับแบบอสมมาตร McEliece

ระบบการเข้ารหัสลับแบบอสมมาตร McEliece (MECS) นำเสนอโดย McEliece (1978) ซึ่งใช้คุณลักษณะของรหัสแก้ไขความผิดพลาด ในการสร้างความซับซ้อนบนพื้นฐานปัญหาความยากในการถอดหรัสกลุ่มรหัสบนรหัสบล็อกเชิงเส้น สมมติฐานความยากของระบบนี้อยู่ในกลุ่มความซับซ้อนของปัญหา NP-complete และจัดเป็นระบบที่ทนทานต่อการประมวลผลควอนตัม ข้อดีของระบบนี้คือการเข้ารหัสและถอดรหัสมีความรวดเร็ว โดยระบบจะประกอบด้วย 3 ส่วน ดังภาพ 7



ภาพ 7 McEliece cryptography

1.1 **การสร้างกุญแจ** (Key generation) เริ่มแรกสร้างกุญแจลับทั้งสาม โดย G คือเมทริกซ์ขนาด k·n สำหรับ (n, k)-linear error correcting code S คือเมทริกซ์ที่สามารถหาอินเวิร์สได้ขนาด k·k และ P คือเมทริซ์เรียงสับเปลี่ยน (Permutation) ขนาด n·n จากนั้นสร้างกุญแจสาธารณะ H โดย $H = S \cdot G \cdot P$

1.2 **การเข้ารหัส** (Encryption) ทำได้โดยใช้เมทริกซ์ข้อความ M ขนาด k จากนั้นสร้างเมทริกซ์ e ขนาด n ซึ่งเป็นรหัสความผิดพลาด สุดท้ายเข้ารหัสข้อความโดยใช้กุญแจสาธารณะ H และ e ได้เป็นข้อความที่ถูกเข้ารหัส $C = M \cdot H + E$

1.3 **การถอดรหัส** (Decryption) ทำได้โดยใช้กุญแจลับทั้งสามดังนี้ เริ่มแรกคำนวณ $C' = C \cdot P^{-1}$ จากนั้นใช้ Error decoding algorithm ซึ่งคำนวณจาก G แก้ไขข้อผิดพลาด e ที่ถูกเพิ่มในขั้นตอนการเข้ารหัส ได้เป็น M' สุดท้ายจะได้ข้อความที่ถูกถอดรหัส M จาก $M = M' \cdot S^{-1}$

2. **การโจมตีบนระบบการเข้ารหัสลับแบบอสมมาตร** McEliece

ความพยายามในการโจมตี MECS แบ่งการโจมตีเป็น 2 ประเภทหลัก คือ 1.การโจมตีเพื่อค้นหากุญแจลับ และ 2.การโจมตีเพื่อถอดรหัสข้อความ ดังนี้

2.1 การโจมตีเพื่อค้นหากุญแจลับ

ในการเข้ารหัสนี้ผู้โจมตีพยายามจะหากุญแจลับ G P และ S ซึ่งค่า n และ k ที่มากขึ้นจะทำให้ผู้โจมตีต้องใช้ความพยายามมากขึ้นในการคาดเดากุญแจ หากผู้โจมตีค้นหากุญแจได้สำเร็จ จะทำให้สามารถถอดรหัสทุก ๆ ข้อความที่ส่งผ่านระบบนี้ได้

2.1.1 Brute-force attack การโจมตีชนิดนี้คือการคาดเดากุญแจลับที่เป็นไปได้ทุก ๆ ตัว บน MECS จะหมายถึงการพยายามคาดเดาเมทริกซ์ P และ S เนื่องจาก G สามารถใช้วิธีการที่มีประสิทธิภาพอื่นในการโจมตีได้ ซึ่งจะกล่าวถึงต่อไป ความยากในการโจมตีขึ้นอยู่กับ

ขนาดของกุญแจ โดยผู้โจมตีต้องใช้ความพยายามในการคาดเดากุญแจ P เมทริซ์เรียงสับเปลี่ยนขนาด n·n ที่เป็นไปได้จำนวน nl ครั้ง และกุญแจ S เมทริกซ์ที่สามารถหาอินเวิร์สได้ขนาด k·k ที่เป็นไปได้จำนวน $\prod_{i=0}^{k-1}(2^k - 2^i)$ ครั้ง (Jochemsz, 2002)

2.1.2 Structural attack การโจมตีนี้อาศัยคุณสมบัติ Permutation equivalent ระหว่างกุญแจสาธารณะและกุญแจลับ (โดยเมทริซ์ P) ซึ่งผู้โจมตีจะใช้การหา Permutation ระหว่างโคดที่สร้างจากทั้งสองกุญแจ หากโจมตีได้สำเร็จ ผู้โจมตีจะทราบกุญแจลับ G โดยในระบบที่ใช้ n = 1024 และ t = 50 ต้องใช้ความพยายาม 2^{466} ครั้ง (Au et al., 2003) ทั้งนี้การโจมตีสามารถทำให้สำเร็จภายใน Polynomial time ได้โดยใช้อัลกอริทึม Support splitting อย่างไรก็ตามการเพิ่มฟิลด์ของค่าสัมประสิทธิ์บนสมการที่ใช้สร้างรหัสบล็อกเชิงเส้นช่วยให้การโจมตีโดยใช้อัลกอริทึมนี้มีประสิทธิภาพลดลงได้ (Repka & Zajac, 2015)

2.2 การโจมตีเพื่อถอดรหัสข้อความ

เป็นการโจมตีไปที่ข้อความที่ถูกเข้ารหัส โดยพยายามถอดรหัสข้อความ M จากข้อความที่ถูกเข้ารหัส C ขนาดของกุญแจสาธารณะ และ Error code ที่มากขึ้น ทำให้ผู้โจมตีต้องใช้ความพยายามมากขึ้นในการถอดรหัสข้อความ หากโจมตีด้วยวิธีสำเร็จ ผู้โจมตีจะทราบเพียงข้อความที่ถูกถอดรหัสได้เท่านั้น

2.2.1 Exhaustive comparison attack การโจมตีชนิดนี้พยายามถอดรหัสข้อความ M โดยใช้การเปรียบเทียบข้อความที่ถูกเข้ารหัส C ที่ได้รับ กับข้อความที่ถูกเข้ารหัส C' ที่ผู้โจมตีสร้าง ซึ่งผู้โจมตีจะสร้าง C'=M·H จาก M จำนวน 2^k ข้อความ จากนั้นหาความแตกต่างระหว่าง C และ C' โดยข้อความที่เป็นไปได้คือ d(C, C') <= t ดังนั้นการโจมตีนี้ต้องใช้ความพยายาม 2^k ครั้ง ต่อการโจมตีแต่ละข้อความ

2.2.2 Syndrome decoding attack การโจมตีนี้ช่วยให้ผู้โจมตีใช้เวลาน้อยลงกว่า Exhaustive comparison attack ข้างต้น โดยการสร้าง Parity check matrix : CH จากกุญแจสาธารณะ H โดย $H(CH)^T = 0$ จากนั้นหา e ที่ทำให้สมการ $C\cdot(CH)^T = e\cdot(CH)^T$ เป็นจริง โดยสร้าง e ทุกค่า ๆ ที่มีน้ำหนัก <= t เมื่อได้ค่า e ที่ถูกต้อง จะสามารถหาข้อความ M จากการใช้ Gaussian elimination ที่ M·H ความพยายามในการคาดเดาค่า e ทำให้การโจมตีนี้ใช้เวลา $\sum_{i=0}^{t}\binom{n}{i}$ ครั้ง

2.2.3 Information set decoding (ISD) Attack เป็นการโจมตีที่มีประสิทธิภาพกว่า Syndrome decoding attack โดยการโจมตีจะสุ่มเลือก k ตำแหน่งจากรหัสขนาด n ซึ่งคาดหวังว่าจะไม่มีความผิดพลาดบนตำแหน่ง k ที่สุ่มเลือก อย่างไรก็ตามการใช้ขนาดกุญแจที่มาก

เพียงพอจะทำให้การโจมตีทำได้ยากขึ้น การเลือกตำแหน่ง k ที่ถูกต้องบนการโจมตีนี้ต้องใช้ความพยายาม $k^3(1 - \frac{t}{n})^{-k}$ ครั้ง (J. Mceliece, 1978)

2.2.4 Message-Resent Attack การโจมตีนี้ใช้ข้อบกพร่องที่เกิดจากการเข้ารหัสข้อความเดิมมากกว่า 1 ครั้ง เช่น กำหนดข้อความที่เข้ารหัส $C_1 = M \cdot H + e_1$ และ $C_2 = M \cdot H + e_2$ โดย $e_1 \neq e_2$ จากข้อความเดียวกัน จะได้ว่า $C_1 + C_2 = e_1 + e_2$ ซึ่งจะได้คำตอบเป็น 0 และ 1 ดังสมการ

$$L_0 = \{l \in \{1, 2, \ldots, n\}: C_1(l) + C_2(l) = e_1(l) + e_2(l) = 0\}$$
$$L_1 = \{l \in \{1, 2, \ldots, n\}: C_1(l) + C_2(l) = e_1(l) + e_2(l) = 1\}$$

โดยการสุ่มเลือก $l \in L_0$ ที่ $e_1(l) + e_2(l) = 0$ หรือ $e_1(l) + e_2(l) = 1$ และ $l \in L_1$ ที่ $e_1(l) = 0, e_2(l) = 1$ หรือ $e_1(l) = 1, e_2(l) = 0$ ต้องใช้ความพยายาม $\frac{(n-t)^2 + t^2}{n}$ และ $\frac{2t(n-t)}{n}$ ครั้ง ตามลำดับ (Jochemsz, 2002) (Au et al., 2003) อย่างไรก็ตามสามารถป้องกันได้โดยการใช้ CCA-secure schemes ที่นำเสนอโดย (Dowsley et al., 2009) (Dottling et al., 2012)

การใช้รหัสแบบอื่นบนระบบการเข้ารหัสลับแบบอสมมาตร McEliece

MECS นำเสนอครั้งแรกโดยใช้รหัส Goppa เป็นรหัสเชิงเส้นแก้ไขข้อผิดพลาดสำหรับการเข้ารหัสและถอดรหัสบนระบบ ซึ่งมีสัญลักษณ์ [n, k, d] แทน รหัส Goppa ขนาด n มิติ k ที่มีระยะทาง Hamming ขนาด d รหัส Goppa $\Gamma(L, g(x))$ สร้างจากสมการพหุนาม g(x) กำลัง t อยู่บน $GF(p^m)$ โดยรหัส Goppa ฐานสอง (อยู่บน $GF(p^2)$) จากสมการพหุนามที่ลดทอนไม่ได้ (Irreducible polynomial) สามารถสร้างอัลกอริทึมแก้ไขข้อผิดพลาดที่มีประสิทธิภาพ บนรหัสบล็อกเชิงเส้นที่มีพารามิเตอร์ [n, >=n-mt , >=2t+1] บนระบบ McEliece ได้

การปรับปรุงระบบเริ่มต้นโดยการเลือกใช้รหัสแบบอื่น ๆ ยกตัวอย่างเช่น การใช้รหัส Reed-Muller (Sidelnikov, 1994) การใช้รหัส Generalized Reed-Solomon (GRS) (Niederreiter, 1986) บนระบบการเข้ารหัสสำหรับลายเซ็นดิจิทัล (Digital signature) ที่ประยุกต์จาก MECS ซึ่งพบว่าถูกโจมตีโดย Structural Attack และค้นหากุญแจลับได้ใน Polynomial time $O(n^4)$ (Repka & Zajac, 2015) (Sidelnikov & Shestakov, 1992)

เนื่องจาก MECS มีข้อเสียเรื่องของขนาดของกุญแจที่มากกว่าระบบอื่น ดังนั้นการเลือกใช้รหัสแบบอื่น ๆ โดยมากมีจุดประสงค์ในการพยายามลดขนาดของกุญแจของระบบลง ซึ่งหมายถึงการทำให้ระบบมีระดับความปลอดภัยที่มากขึ้นเมื่อเทียบกับขนาดของกุญที่เท่ากัน ดังเช่น

การใช้รหัส LDPC (Low density parity check) แทนที่รหัส Goppa แต่ปัญหาคือการใช้รหัสนี้ทำให้ไม่สามารถใช้เมทริกซ์เรียงสับเปลี่ยน P เพื่อซ่อนกุญแจลับเหมือนระบบดั้งเดิมได้ เนื่องจากผู้โจมตีสามารถหาข้อความแท้จริงซึ่งมี Low-weight จากข้อความที่ถูกเข้ารหัสได้ ทำให้ผู้โจมตีสามารถถอดรหัสข้อความได้โดยง่าย ต่อมาจึงมีการพัฒนาระบบมาใช้รหัส Quasi-cyclic LDPC และใช้เมทริกซ์ Q แทนที่ P เพื่อสร้าง weight ให้กับข้อความ ทำให้สามารถแก้ไขปัญหาที่เกิดขึ้นระบบบนรหัส LDPC ได้ ซึ่งการใช้รหัส QC-LDPC นี้ช่วยลดขนาดกุญแจสาธารณะได้ประมาณ 10 เท่าจาก MECS (อ้างอิงที่ระดับความปลอดภัย 80-bit ของ ISD attack) (Monico et al., 2000) (Baldi et al., 2008)

ต่อมารหัส MDPC (Moderate density parity check) นำมาใช้บน MECS เพื่อลดขนาดของกุญแจลงจากระบบเดิมเช่นกัน ซึ่งรหัส MDPC คล้ายกับ LDPC เพียงแต่ใช้ Moderate–weight ในการสร้าง Parity check การใช้ QC-MDPC จะแก้ไขข้อผิดพลาดได้น้อยกว่า QC-LDPC แต่ไม่จำเป็นต้องใช้เมทริกซ์ Q เพื่อสร้าง weight ให้กับข้อความ (Misoczki et al., 2013) อย่างไรก็ตาม การใช้ LDPC และ MDPC นั้นมีปัญหาจากอัลกอริทึม Bit-flipping ซึ่งทำให้บางกรณีมีการถอดรหัสข้อความที่ผิดพลาด จึงมีการนำเสนอระบบที่ใช้รหัส MDPC แต่ได้ปรับปรุงโดยใช้อัลกอริทึม Soft-decision interactive decoding แทนที่อัลกอริทึม Bit-flipping เดิม เพื่อเพิ่มประสิทธิภาพในการถอดรหัสข้อความ (Marco Baldi et al., 2016)

รหัส Polar เป็นรหัสสำหรับแก้ไขข้อผิดพลาดที่มีความซับซ้อนในการเข้ารหัสและถอดรหัสต่ำ รหัส Polar ได้นำมาใช้บน MECS โดย (Shrestha & Kim, 2014) (Hooshmand et al., 2017) ซึ่งนอกจากข้อดีข้างต้นยังสามารถช่วยลดขนาดกุญแจและเพิ่มปริมาณข้อมูลต่อบล็อกได้เมื่อเทียบกับระบบ MECS เดิม อย่างไรก็ตาม Message-resent attack สามารถโจมตีระบบนี้ได้ ทำให้ต้องเพิ่มการใช้ CCA-secure schemes เช่นเดียวกับ Goppa-based MECS

รหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน

รหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียนนำเสนอโดย Huber (1994) เป็นรหัสบล็อกเชิงเส้นที่อยู่บนฟิลด์ของจำนวนเต็มแบบเกาส์เซียน เพื่อนำไปใช้สำหรับแก้ไขข้อผิดพลาดบนสเปซสัญญาณแบบ 2 มิติ โดยมีหัวข้อที่เกี่ยวข้อง ดังนี้

1. **จำนวนเต็มแบบเกาส์เซียน**

จำนวนเต็มแบบเกาส์เซียน (Gaussian integer) เป็นจำนวนเชิงซ้อนซึ่งอยู่ในรูปของ a+bi โดยที่ a และ b เป็นจำนวนเต็มและ $i = \sqrt{-1}$ ดังสมการ

$$\mathbb{Z}[i] = \{ a+bi : a, b \in \mathbb{Z}, i = \sqrt{-1} \}$$

กำหนดให้จำนวนเต็มแบบเกาส์เซียน G = a+bi เขียนแทนด้วยสัญลักษณ์ G = (a, b)

1.1 การบวกและลบจำนวนเต็มแบบเกาส์เซียน

การบวกและลบจำนวนเต็มแบบเกาส์เซียนสองจำนวน $G_1 = (a, b)$ และ $G_2 = (c, d)$ ได้ผลลัพธ์คือ

$$G_1 + G_2 = (a+c, b+d)$$

$$G_1 - G_2 = (a-c, b-d)$$

1.2 การคูณจำนวนเต็มแบบเกาส์เซียน

การคูณจำนวนเต็มแบบเกาส์เซียนสองจำนวน $G_1 = (a, b)$ และ $G_2 = (c, d)$ ได้ผลลัพธ์คือ

$$G_1G_2 = (ac-bd, ad+bc)$$

1.3 การมอดุลาร์จำนวนเต็มแบบเกาส์เซียน

การมอดุลาร์จำนวนเต็มแบบเกาส์เซียนสองจำนวน $G_1 = (a, b)$ และ $G_2 = (c, d)$ ได้ผลลัพธ์คือ

$$G_1 \bmod G_2 = (a, b) - \left[\frac{(a, b)\cdot(c, d)^*}{(c, d)\cdot(c, d)^*}\right](c, d)$$

โดยที่ $(c, d)^*$ คือ สังยุคของ G_2 และ [...] คือ การปัดให้เป็นจำนวนเต็มแบบเกาส์เซียน

1.4 การปัดจำนวนเต็มแบบเกาส์เซียน

การปัดจำนวนเชิงซ้อนไปยังจำนวนเต็มแบบเกาส์เซียนที่ใกล้ที่สุด กำหนดจำนวนเชิงซ้อน $z = (\alpha, \beta)$ ได้ผลลัพธ์คือ

$$[z] = ([\alpha], [\beta]) = (a, b) = G$$

2. **ฟิล์ดจำกัดบนจำนวนเต็มแบบเกาส์เซียน: $\mathcal{G}_\pi$**

จากทฤษฎี Sum of two square ของ Fermat จำนวนเฉพาะที่อยู่ในรูปแบบ $p \equiv 1 \bmod 4$ สามารถเขียนในรูปของ $p = a^2+b^2 = \pi\cdot\pi^*$ โดยที่ π^* คือสังยุคของจำนวนเชิงซ้อน $\pi = a+bi$

การสร้าง $\mathcal{G}_\pi$ (Finite gaussian integer field) ทำได้โดยการหาสมาชิกภายใน $\mathcal{G}_\pi$ คือ μ ซึ่ง $\mu = \varepsilon \bmod \pi$ โดยที่ $\varepsilon \in \mathbb{N}$ และ $p \equiv 1 \bmod 4 = \pi\cdot\pi^*$

ยกตัวอย่าง เช่น $\mathcal{G}_\pi$ เมื่อ $p = 25$ จะได้ $\pi = 3+4i$ โดยให้ $\varepsilon = \mathbb{N}_5$ ดังนั้นได้ผลลัพธ์คือ

$$\mathcal{G}_{4+3i} = \{\varepsilon \bmod (4+3i), \varepsilon = \mathbb{N}_5\}$$
$$= \{1, 2, 3, -3i, -2+i\}$$

3. **ระยะทาง** Mannheim

ระยะทาง Mannheim (Mannheim distance) เป็นระยะทางที่นำเสนอมาเพื่อใช้วัดความแตกต่างระหว่างจำนวนเต็มแบบเกาส์เซียนบน $\mathcal{G}_\pi$ เนื่องจากการคำนวณระยะทางแบบ Hamming (Hamming distance) ที่มักจะใช้บนรหัสบล็อกทั่วไป ไม่รองรับรหัสบล็อกที่มีสองมิติ

กำหนดให้ α และ $\beta \in \mathcal{G}_\pi$ และ $\gamma = \beta - \alpha \bmod \pi$ น้ำหนัก Mannheim W_m ของ γ คำนวณได้ดังนี้

$$W_m(\gamma) = |Re(\gamma)|+|Im(\gamma)|$$

ระยะทาง Mannheim d_m ระหว่าง α และ β คำนวณได้ดังนี้

$$d_m(\alpha, \beta) = W_m(\gamma)$$

น้ำหนัก Mannheim ของเวคเตอร์ $x = \{x_0, x_1, \ldots, x_{n-1}\}$ บน $\mathcal{G}_\pi$ คือ

$$W_m(x) = \sum_{i=0}^{n-1} W_m(x_i)$$

ระยะทาง Mannheim d_m ระหว่างเวคเตอร์ $x = \{x_0, x_1, \ldots, x_{n-1}\}$ และ $y = \{y_0, y_1, \ldots, y_{n-1}\}$ บน $\mathcal{G}_\pi$ คือ

$$d_m(x, y) = \sum_{i=0}^{n-1} d_m(x_i, y_i)$$

4. รหัสสำหรับแก้ไขข้อผิดพลาด Mannheim (Mannheim error correcting code)

4.1 OMEC (One Mannheim error correcting code)

เป็นรหัสบล็อกสำหรับแก้ไขข้อผิดพลาดของรหัสขนาด n = (p-1)/4 โดยที่ $p \equiv 1 \mod 4$ โดยสามารถใส่ค่าความผิดพลาดได้ 4 ค่า คือ {1, -1, i, -i} ลงบนรหัสตำแหน่งที่ l โดย $0 \leq l \leq n-1$

การสร้าง Parity check matrix สำหรับ OMEC ทำได้โดย กำหนดให้ $\alpha \in \mathcal{G}_\pi$ เป็นสมาชิกตามลำดับ p-1 ดังนั้น Parity check matrix คือ

$$CH = \left[\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{((p-1)/4)-1}\right]$$

จากนั้นสร้าง Generator matrix ที่ทำให้ $CH \cdot c^T = 0$ ในทุกๆ รหัสบล็อก $c = \{c_0, c_1, \ldots, c_{n-1}\}$ บน $\mathcal{G}_\pi$ คือ

$$G = \begin{bmatrix} -\alpha^1 & 1 & 0 & \cdots & 0 \\ -\alpha^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\alpha^{((p-1)/4)-1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

4.2 ตัวอย่าง OMEC

กำหนดให้ p = 13 $\pi = 3+2i$ และ $\alpha \in \mathcal{G}_\pi$ ตั้งแต่ 1 ถึง p-1 จะได้ $\alpha_1 = 1$, $\alpha_2 = 1+i$, $\alpha_3 = 2i$, ..., $\alpha_{12} = -2$ จากการกำหนด $\alpha = 1+i$ ข้างต้น จะได้ $CH = \left[\alpha^0 = \alpha_1, \alpha^1 = \alpha_2, \alpha^2 = \alpha_3\right]$ ดังนั้น

$$CH = \left[1, 1+i, 2i\right]$$

$$G = \begin{bmatrix} -(1+i) & 1 & 0 \\ -2i & 0 & 1 \end{bmatrix}$$

สมมติว่าได้รับ $r = [1+i, 0+i, -1+i]$ มีความผิดพลาด 1 ตำแหน่ง นำ $s = H \cdot r^T = -2 = \alpha^{11}$ จะได้ว่ามีความผิดพลาดในตำแหน่งที่ 2 เนื่องจาก $2 \equiv 11 \bmod n$ โดยที่ $n = (p-1)/4$ ค่าความผิดพลาดคำนวณจาก $s \cdot \alpha^{-1} = -2 \cdot (1/2i) = i$ ดังนั้น $e = [0, 0, i]$ จะได้ $c = r - e = [1+i, 0+i, -1+0]$

บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะอธิบายถึงวิธีดำเนินการวิจัยโดยประกอบด้วย 3 ส่วน คือ กรอบแนวคิดงานวิจัย การพัฒนาระบบ และการประเมินผล ซึ่งจะอธิบายดังต่อไปนี้

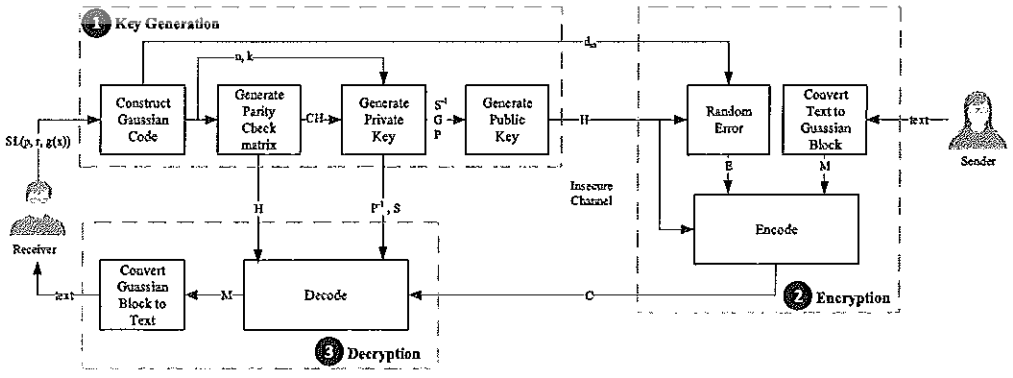
กรอบแนวคิดงานวิจัย

ผู้วิจัยได้ออกแบบและพัฒนาระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนบนพื้นฐานของการเข้ารหัสที่นำเสนอโดย McEliece เนื่องจากเป็นระบบหนึ่งที่ต้านทานต่อการประมวลแบบควอนตัม โดยผู้วิจัยมุ่งเน้นไปที่การนำรหัส Mannheim ซึ่งเป็นรหัสแบบสองมิติที่อยู่บนฟิล์ดจำนวนเต็มแบบเกาส์เซียนมาใช้กับระบบเข้ารหัสลับเดิม โดยกรอบแนวคิดของระบบแสดงดังภาพ 8

ระบบประกอบด้วย 3 กระบวนการหลัก คือ 1) การสร้างกุญแจ (Key generation) 2) การเข้ารหัสลับ (Encryption) และ 3) การถอดรหัสลับ (Decryption) โดยระบบมีปฏิสัมพันธ์กับผู้ใช้สองประเภท คือ ผู้รับข้อความ (Receiver) และผู้ส่งข้อความ (Sender)

ภาพรวมในการทำงานระบบคือ ผู้รับข้อความจะเป็นผู้กำหนดพารามิเตอร์สำคัญในการสร้างกุญแจเพื่อใช้ในกระบวนการเข้ารหัสและถอดรหัสข้อความ และรอรับข้อความที่เข้ารหัสลับเพื่อถอดรหัสลับกลับเป็นข้อความที่แท้จริง ส่วนผู้ส่งข้อความจะนำข้อความที่ต้องการส่งให้ผู้รับข้อความแปลงเป็นรหัสลับผ่านกระบวนการเข้ารหัส

กระบวนการย่อยต่าง ๆ ภายในกระบวนหลักทั้งสาม อธิบายรายละเอียดในหัวข้อ 1-3



ภาพ 8 กรอบแนวคิดการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน

ผู้วิจัยได้พัฒนาระบบอยู่บนพื้นฐานของเมทริกซ์ ดังนั้นกุญแจที่ใช้กระบวนการเข้ารหัสและถอดรหัส รวมถึงตัวแปรลับอื่น ๆ ใช้การแสดงผลแบบเมทริกซ์ โดยจากกรอบแนวคิดของระบบพารามิเตอร์และเมทริกซ์ที่เกี่ยวข้องอธิบายดังตารางที่ 3 และ 4 ตามลำดับ

ตาราง 3 พารามิเตอร์ที่ใช้ในระบบ

พารามิเตอร์	คำอธิบาย	ขอบเขต	ตัวอย่างค่า
SL	ระดับความปลอดภัย (Security level) โดยกำหนดตามความปลอดภัยของการโจมตีข้อความแบบ Information set decoding attack	จำนวนเต็มบวก โดยมากกำหนดขั้นต่ำตามมาตรฐานคือมากกว่า 60	60(1013, 1, g(x))
p	พารามิเตอร์กำหนดระดับความปลอดภัย โดยสร้างสมาชิกที่ใช้ได้บนรหัสเกาส์เซียนจำนวน p ตัว	จำนวนเฉพาะที่ทำให้ $p \equiv 1 \bmod 4$ และสามารถเขียนในรูปของ $p = a^2+b^2$ โดยที่ a และ b เป็นจำนวนเต็มบวก	1013
r	พารามิเตอร์กำหนดระดับความปลอดภัย โดยกำหนดดีกรีสมการพหุนาม	จำนวนเต็มบวก	1
g(x)	สมการพหุนามสำหรับสร้างรหัสเกาส์เซียน (Generator polynomial)	สมการพหุนามดีกรี r ที่มีสัมประสิทธิ์เป็นสมาชิกบน $\mathcal{G}_\pi$	g(x) = x-1
n	ขนาดของกุญแจ และขนาดของรหัสบนจำนวนเต็มแบบเกาส์เซียน	จำนวนเต็มบวก โดยคำนวณจาก $n = (p^r-1)/4$	$n = (1013^1-1)/4$ = 253
k	ขนาดของกุญแจ และขนาดของรหัสบนจำนวนเต็มแบบเกาส์เซียน	จำนวนเต็มบวก โดยที่ $k < n$ และ $k = n-d_m$	k = 253-2 = 251
d_m	จำนวนของความผิดพลาดที่สามารถใส่ได้ในกระบวนการเข้ารหัส	จำนวนเต็มบวก โดย $d_m < n$ และจำนวนสูงสุดที่ใช้ได้คำนวณจากค่าน้ำหนักของรหัสเกาส์เซียน	2
π	จำนวนเต็มแบบเกาส์เซียน	จำนวนเชิงซ้อน $\pi = a+bi$ จาก $p = a^2+b^2$ โดยที่ a และ b เป็นจำนวนเต็มบวก	$\pi = 23+22i$
$\mathcal{G}_\pi$	สมาชิกตั้งต้นของรหัสบนจำนวนเต็มแบบเกาส์เซียนเกาส์เซียน	ฟิลด์จำกัดบนจำนวนเต็มแบบเกาส์เซียนโดยสร้างจาก $\mathcal{G}_\pi = \{\varepsilon \bmod \pi, \varepsilon = \mathbb{N}_\rho\}$	$\mathcal{G}_\pi$ = {1, 13-7i, -11+i, ..., 3+17i}
$\mathcal{G}_{\pi'}$	สมาชิกของรหัสของรหัสบนจำนวนเต็มแบบเกาส์เซียนที่สร้างจาก g(x)	ฟิลด์จำกัดบนจำนวนเต็มแบบเกาส์เซียนโดยสร้างจาก $\mathcal{G}_\pi$ และ g(x)	$\mathcal{G}_\pi^1$ = {1, 13-7i, -11+i, ..., 3+17i}

ตาราง 4 เมทริกซ์ที่ใช้ในระบบ

เมทริกซ์	คำอธิบาย	ขอบเขต
CH	เมทริกซ์ตรวจสอบความผิดพลาด (Parity check matrix) ใช้ในการตรวจสอบความผิดพลาดที่ใส่ในกระบวนการเข้ารหัส	เมทริกซ์ขนาด d_m·n มีสมาชิกอยู่ใน $\mathcal{G}_\pi'$
G	เมทริกซ์กุญแจลับ (Private key) ใช้ในการถอดรหัสข้อความ โดยสร้างจาก null space ของ CH	เมทริกซ์ขนาด k·n มีสมาชิกอยู่ใน $\mathcal{G}_\pi'$
S และ S^{-1}	เมทริกซ์กุญแจลับใช้ในการถอดรหัสข้อความ โดยสร้างเมทริกซ์สุ่ม (Scramble matrix) จากการสุ่มสมาชิกใน $\mathcal{G}_\pi'$ และสร้างเมทริกซ์อินเวิร์สของตนเอง	เมทริกซ์ขนาดา k·k มีสมาชิกอยู่ใน $\mathcal{G}_\pi$
P และ P^{-1}	เมทริกซ์กุญแจลับใช้ในการถอดรหัสข้อความ โดยสร้างเมทริกซ์เรียงสับเปลี่ยน (Permutation matrix) จากการเรียงสับเปลี่ยนเมทริกซ์เอกลักษณ์ของจำนวนเชิงซ้อน และสร้างเมทริกซ์อินเวิร์สของตนเอง	เมทริกซ์ขนาด n·n ค่าในเมทริกซ์เอกลักษณ์สามารถเลือกได้ 4 ค่า คือ {1, -1, i, -i}
H	เมทริกซ์กุญแจสาธารณะ (Public key) ใช้ในการเข้ารหัสข้อความ โดยสร้างจากเมทริกซ์กุญแจลับ	เมทริกซ์ขนาด k·n มีสมาชิกอยู่ใน $\mathcal{G}_\pi$
M	เมทริกซ์ข้อความ	เมทริกซ์ขนาด 1·k มีสมาชิกอยู่ใน $\mathcal{G}_\pi$
C	เมทริกซ์ข้อความเข้ารหัส	เมทริกซ์ขนาด 1·n มีสมาชิกอยู่ใน $\mathcal{G}_\pi$
E	เมทริกซ์ความผิดพลาดที่ใส่ในกระบวนการเข้ารหัส โดยสามารถใส่ค่า {1, -1, i, -i} จำนวน d_m ตำแหน่ง	เมทริกซ์ขนาด 1·n มีสมาชิกอยู่ใน {1, -1, i, -i}

1. การสร้างกุญแจ (Key generation)

กระบวนการสร้างกุญแจประกอบด้วย 4 ส่วนหลัก ดังนี้

1.1 **การสร้างรหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน**

โดยผู้รับข้อความจะกำหนดพารามิเตอร์ 2 ตัว คือ p และ r โดยที่ p เป็นจำนวนเฉพาะ และ $r > 0$ จากนั้นระบบจะสร้างพารามิเตอร์ g(x) ให้อัตโนมัติเพื่อให้สอดคล้องกัน การกำหนดพารามิเตอร์ข้างต้นจะได้รหัสบล็อกสำหรับแก้ไขข้อผิดพลาด $[n, k, d_m]_p$ ซึ่งเป็นรหัสขนาด $n = (p^r-1)/4$ ที่มีมิติ $k = n-d_m$ มีจำนวนของความผิดพลาด d_m โดยมีสมาชิกบนฟิลด์จำกัดบนจำนวนเต็มแบบเกาส์เซียนดั้งเดิม $\mathcal{G}_\pi$ จำนวน p-1 ตัว และ ฟิลด์ $\mathcal{G}_\pi'$ ที่สร้างจากสมการพหุนาม g(x) จำนวน p^r-1 ตัว ดังนั้นการกำหนดค่า p และ r ที่มากขึ้นจะทำให้ระดับความปลอดภัยสูงขึ้น เนื่องจากจำนวนสมาชิกบนฟิลด์และขนาดของรหัสที่มากขึ้น ซึ่งสามารถสร้างกุญแจได้หลากหลายกว่าเดิม ทำให้ผู้โจมตีต้องใช้ความพยายามมากขึ้นในการคาดเดากุญแจลับ อย่างไรก็ตามระบบอาจจะต้องใช้เวลาในการประมวลเพิ่มขึ้นและพื้นที่ที่จัดเก็บมากขึ้นจากการเพิ่มขนาดของกุญแจ

1.2 การสร้างเมทริกซ์ตรวจสอบความผิดพลาด

เป็นการสร้างเมทริกซ์สำหรับให้ระบบสามารถแก้ไขโคดให้ถูกต้อง (Parity check matrix) จากการใส่ความผิดพลาดให้กับรหัสบล็อกในกระบวนการเข้ารหัส ซึ่งช่วยให้ผู้ที่มีกุญแจลับสามารถค้นหาตำแหน่งความผิดพลาดของรหัสและแก้ไขได้รวดเร็วขึ้น เมื่อเปรียบเทียบกับผู้โจมตีที่ต้องพยายามคาดเดาตำแหน่งความผิดพลาดไปเรื่อย ๆ

โดยเมทริกซ์ตรวจสอบความผิดพลาด CH นี้ ทำได้โดยการหา β ซึ่งเป็นสมาชิกของ $\mathcal{G}_{\pi^r}$ ตามลำดับจนถึง p-1 จะได้

$$CH = \left[\beta^0, \beta^1, \beta^2, \ldots, \beta^{((p^r-1)/4)-1}\right]$$

1.3 การสร้างกุญแจลับ

ระบบจะสร้างกุญแจลับ (Private key) จากพารามิเตอร์เริ่มต้นและเมทริกซ์ตรวจสอบความผิดพลาดที่กำหนดในกระบวนการก่อนหน้าคือ

กุญแจลับ G โดยสร้างเมทริกซ์ขนาด k·n ซึ่งได้จาก $(CH)^T \cdot G = 0$ นั่นคือสร้างแต่ละแถวของเมทริกซ์ G ด้วยปริภูมิว่าง (Null space) ของ CH ดังนี้

$$G = \begin{bmatrix} -\beta^r & 1 & 0 & \cdots & 0 \\ -\beta^{r+1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\beta^{((p^r-1)/4)-1} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

กุญแจลับ S โดยสร้างเมทริกซ์ที่สามารถหาอินเวิร์สได้ (Nonsingular matrix) ขนาด k·k

กุญแจลับ P โดยสร้างเมทริกซ์เรียงสับเปลี่ยน (Permutation matrix) ขนาด n·n

โดยกุญแจลับทั้งหมดถูกสร้างอยู่บนโดเมนของจำนวนเต็มแบบเกาส์เซียน $\mathcal{G}_{\pi}$ ทั้งนี้ผู้รับข้อความจะต้องไม่เปิดเผยกุญแจลับ เพื่อป้องกันการโจมตีที่อาจเกิดขึ้นบนระบบ

1.4 การสร้างกุญแจสาธารณะ

โดยระบบจะสร้างกุญแจสาธารณะ H นี้จากกุญแจลับดังสมการ

$$H = S^{-1} \cdot G \cdot P \pmod{\pi}$$

จะได้กุญแจสาธารณะเป็นเมทริกซ์ขนาด k·n จากนั้นระบบจะเผยแพร่กุญแจสาธารณะที่สร้างขึ้น เพื่อให้ผู้ส่งข้อความที่ต้องการจะส่งข้อความผ่านระบบมายังผู้รับข้อความ

อัลกอริทึมของกระบวนการสร้างกุญแจแสดงดังตาราง 5

ตาราง 5 กระบวนการสร้างกุญแจของระบบ

อัลกอริทึมที่ 1: การสร้างกุญแจของระบบ	
1 :	กำหนดตัวแปรเริ่มต้นให้กับระบบ คือ p r และ g(x)
2 :	สร้างรหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน $[n, k, d_m]_p$ บนฟิลด์ $\mathcal{G}_\pi$ และ $\mathcal{G}_\pi'$
3 :	สร้างเมทริกซ์ตรวจสอบความผิดพลาด CH
4 :	สร้างกุญแจลับ G S และ P
5 :	คำนวณเมทริกซ์ G ขนาด k·n
6 :	สร้างเมทริกซ์ที่สามารถหาอินเวิร์สได้ S ขนาด k·k
7 :	สร้างเมทริซ์เรียงสับเปลี่ยน P ขนาด n·n
8 :	สร้างกุญแจสาธารณะ $H = S^{-1} \cdot G \cdot P \pmod{\pi}$
9 :	ส่งค่า H ไปยังกระบวนการเข้ารหัส
10 :	ส่งค่า CH S และ P ไปยังกระบวนการถอดรหัส

2. **การเข้ารหัส** (Encryption)

ส่วนที่สองเป็นกระบวนการเข้ารหัส ประกอบด้วย 3 ส่วนหลัก ดังนี้

2.1 การแปลงข้อความเป็นรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน

กระบวนการนี้จะแปลงข้อความ M จากผู้ส่งเป็นรหัสบล็อก M_π ซึ่งเป็นรหัสภายใต้โดเมนของจำนวนเต็มแบบเกาส์เซียน $\mathcal{G}_\pi$ โดยใช้วิธีการแปลงโดยใช้ระบบตัวเลขฐาน p แสดงในหัวข้อ 4.1

2.2 การสุ่มตำแหน่งความผิดพลาด

เป็นการใส่เวกเตอร์ความผิดพลาดไปยังรหัสบล็อกที่ถูกเข้ารหัส โดยสร้างเมทริกซ์ E ขนาด n มีค่าความผิดพลาดแต่ละตำแหน่งเท่ากับ {1, -1, i, -i} และมีจำนวนตำแหน่งความผิดพลาดที่ระบบสามารถสร้างได้ โดยขึ้นอยู่กับค่า d_m ที่กำหนดไว้เบื้องต้น การกำหนดจำนวน

ตำแหน่งของความผิดพลาดที่มากขึ้น จะทำให้การแก้ไขข้อผิดพลาดทำได้ยากขึ้นเมื่อผู้โจมตีไม่ทราบกุญแจลับ

2.3 การเข้ารหัส

เป็นการนำรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน M_π ขนาด k มาเข้ารหัสโดยใช้กุญแจสาธารณะ และใส่ค่าความผิดพลาดที่สร้างจากขั้นตอนก่อนหน้า ได้เป็นรหัสบล็อกที่ถูกเข้ารหัสแล้ว C ขนาด n ดังสมการ

$$C = M_\pi \cdot H + E \pmod{\pi}$$

จากนั้นรหัสบล็อกที่ถูกเข้ารหัส C จะถูกส่งไปยังผู้รับข้อความ เพื่อทำการถอดรหัสต่อไป

อัลกอริทึมของกระบวนการเข้ารหัสของระบบแสดงดังตาราง 6

ตาราง 6 กระบวนการเข้ารหัสของระบบ

อัลกอริทึมที่ 2: การเข้ารหัสข้อความของระบบ	
1 :	แปลงข้อความ M เป็นบล็อกของรหัสจำนวนเต็มแบบเกาส์เซียน M_π
2 :	สุ่มตำแหน่งความผิดพลาด E
3 :	เข้ารหัส $C = M_\pi \cdot H + E \pmod{\pi}$
4 :	ส่งรหัสบล็อกที่ถูกเข้ารหัส C ไปยังกระบวนการถอดรหัส

3. **การถอดรหัส** (Decryption)

ส่วนสุดท้ายกระบวนการถอดรหัส ประกอบด้วย 2 ส่วนหลัก ดังนี้

3.1 การถอดรหัส

เป็นการนำกุญแจลับมาใช้เพื่อถอดรหัสของรหัสบล็อกที่ถูกเข้ารหัส กลับเป็นรหัสบล็อกตั้งต้นที่ถูกส่งมาจากผู้ส่งข้อความ โดยคำนวณจากสมการตามลำดับ คือ

เริ่มต้นด้วยกำจัดการเรียงสับเปลี่ยนโดยใช้อินเวิร์สของเมทริกซ์เรียงสับเปลี่ยน P

$$C' = C \cdot P^{-1}$$

จากนั้นแก้ไขความผิดพลาดบนรหัสบล็อกจากค่าความผิดพลาดที่ถูกใส่มาในกระบวนการเข้ารหัสโดยใช้เมทริกซ์ตรวจสอบความผิดพลาด CH ทำได้โดยคำนวณหาซินโดรมของรหัส $s = CH \cdot C'^{T}$ จะได้ตำแหน่งที่ผิดพลาด $l = \log_{\alpha} s \pmod n$ และค่าผิดพลาด $v = s \cdot \alpha^{-l}$

$$C'' = C' \text{ where correcting by CH}$$

สุดท้ายกำจัดการสุ่มที่ใส่ลงบนรหัส โดยใช้เมทริกซ์ S ได้เป็นรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน M_{π}

$$M_{\pi} = C'' \cdot S \pmod{\pi}$$

3.2 การแปลงรหัสจำนวนเต็มแบบเกาส์เซียนเป็นข้อความ

กระบวนการนี้จะแปลงรหัสบล็อก M_{π} ของรหัสภายใต้โดเมนของจำนวนเต็มแบบเกาส์เซียนที่ถอดรหัสมาจากกระบวนการก่อนหน้าเรียบร้อยแล้วกลับเป็นข้อความ M ต้นฉบับจากผู้ส่ง โดยใช้วิธีการแปลงโดยใช้ระบบตัวเลขฐาน p แสดงในหัวข้อ 4.1

อัลกอริทึมของกระบวนการถอดรหัสของระบบแสดงดังตาราง 7

ตาราง 7 กระบวนการถอดรหัสของระบบ

อัลกอริทึมที่ 3: การถอดรหัสข้อความของระบบ	
1 :	กำจัดการเรียงสับเปลี่ยนบนรหัสบล็อกที่ถูกเข้ารหัส C ได้เป็น $C' = C \cdot P^{-1}$
2 :	แก้ไขความผิดพลาดของรหัสบล็อก C' จาก CH ได้เป็น C''
3 :	กำจัดการสุ่มที่ใส่ลงบนรหัส C'' ได้เป็น $M_{\pi} = C'' \cdot S \pmod{\pi}$
4 :	แปลงบล็อกของรหัสจำนวนเต็มแบบเกาส์เซียน M_{π} เป็นข้อความ M

4. การส่งรหัสบล็อกจำนวนเต็มแบบเกาส์เซียนไปยังช่องสัญญาณ

เมื่อผู้ส่งข้อความได้ทำการเข้ารหัสข้อความแล้ว ข้อความเข้ารหัสที่เป็นรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน (C_{π} : Gaussian ciphertext) จะถูกส่งไปหาผู้รับข้อความผ่านช่องสัญญาณ เพื่อทำการถอดรหัสข้อความ ซึ่งการส่งผ่านช่องสัญญาณนี้จำเป็นต้องใช้รหัสที่อยู่ใน

ระบบเลขฐานสองหรือรูปแบบบิต (C_2 : Binary ciphertext) ดังนั้นระบบจะแปลงรหัสบล็อกจำนวนเต็มแบบเกาส์เซียนให้เป็นรหัสฐานสองเพื่อส่งผ่านช่องสัญญาณ แล้วแปลงรหัสฐานสองกลับเป็นรหัสบล็อกจำนวนเต็มแบบเกาส์เซียนก่อนที่จะเข้าสู่กระบวนการถอดรหัสต่อไป โดยกระบวนการนี้แสดงดังภาพที่ 9

ภาพ 9 วิธีการส่งรหัสบล็อกจำนวนเต็มแบบเกาส์เซียนไปยังช่องสัญญาณ

ในระบบการเข้ารหัสลับที่ผู้วิจัยพัฒนานี้ นอกจากการส่งรหัสจำนวนเต็มแบบเกาส์เซียนไปยังช่องสัญญาณแล้ว การแปลงข้อความเป็นรหัสจำนวนเต็มแบบเกาส์เซียน และการแปลงรหัสจำนวนเต็มแบบเกาส์เซียนกลับเป็นข้อความ ใช้หลักการแปลงเดียวกันทั้งหมด ซึ่งใช้เป็นระบบตัวเลขฐาน p อ้างอิงจากจำนวนสมาชิกของฟิลด์จำนวนเต็มแบบเกาส์เซียน G_π โดยวิธีการแปลงอธิบายในหัวข้อ 4.1 ดังนี้

4.1 **การแปลงโดยใช้ระบบตัวเลขฐาน p**

วิธีการนี้ใช้ฟิลด์จำนวนเต็มแบบเกาส์เซียน G_π ที่มีจำนวนสมาชิก p ตัว เป็นตัวกลางในการแปลงรหัสกลับไปมา โดยในการแปลงรหัสจำนวนเต็มแบบเกาส์เซียนเป็นรหัสฐานสอง ระบบจะเริ่มต้นแปลงรหัสภายใต้โดเมนของจำนวนเต็มแบบเกาส์เซียน G_π เป็นรหัสฐาน p ก่อน จากนั้นใช้วิธีการของ Horner (Seroul, 2000) แปลงรหัสฐาน p เป็นรหัสฐานสอง ดังอัลกอริทึมที่ 4 แสดงในตาราง 8 ส่วนวิธีการแปลงรหัสฐานสองเป็นรหัสจำนวนเต็มแบบเกาส์เซียนใช้วิธีการย้อนกลับ กล่าวคือระบบจะเริ่มต้นแปลงรหัสฐานสองเป็นรหัสฐาน p โดยใช้วิธีการของ Horner จากนั้นแปลงรหัสฐาน p เป็นรหัสจำนวนเต็มแบบเกาส์เซียน โดยจับคู่สมาชิกบนฟิลด์จำนวนเต็มแบบเกาส์เซียน G_π ดังอัลกอริทึมที่ 5 แสดงในตาราง 9

ตาราง 8 กระบวนการแปลงรหัสจำนวนเต็มแบบเกาส์เซียนเป็นรหัสฐานสอง

อัลกอริทึมที่ 4: การแปลงรหัสจำนวนเต็มแบบเกาส์เซียนเป็นรหัสฐานสอง	
1 :	จับคู่รหัสบนฟิลด์ G_π เป็นรหัสฐาน p
2 :	แปลงรหัสฐาน p เป็นรหัสฐานสอง

ตาราง 9 กระบวนการแปลงรหัสฐานสองเป็นบล็อกของรหัสจำนวนเต็มแบบเกาส์เซียน

อัลกอริทึมที่ 5: การแปลงรหัสฐานสองเป็นบล็อกของรหัสจำนวนเต็มแบบเกาส์เซียน	
1 :	แปลงรหัสฐานสองเป็นรหัสฐาน p
2 :	จับคู่รหัสฐาน p กับรหัสบนฟิลด์ $\mathcal{G}_\pi$

ตัวอย่างของการแปลงโดยใช้ระบบตัวเลขฐาน p จะแสดงในหัวข้อ 5 ต่อไป ซึ่งนอกจากวิธีการนี้แล้ว ยังมีวิธีการแปลงแบบอื่นที่สามารถทำได้ดังนี้

4.2 **การแปลงโดยใช้ระบบตัวเลขฐาน** Quarter-imaginary

ระบบเลขฐาน Quarter-imaginary (Knuth, 1960) ใช้จำนวนจินตภาพ 2i เป็นฐานของระบบตัวเลข โดยเป็นระบบเลขฐานที่สามารถแสดงเป็นเลขจำนวนเต็มแทนเลขที่เป็นจำนวนเชิงซ้อนได้ และสามารถแปลงกลับเป็นจำนวนเชิงซ้อนค่าเดิมได้ นอกจากนี้ยังสามารถใช้กับจำนวนเชิงซ้อนที่เป็นค่าลบได้ เช่น จำนวนเต็มแบบเกาส์เซียนคือ 1+2i สามารถแปลงได้เป็น 11_{2i} ซึ่งอยู่ในรูปแบบบิตเพื่อส่งไปยังช่องสัญญาณ และแปลงกลับเป็น 1+2i ค่าเดิมผ่านระบบเลขฐานนี้

4.3 **การแปลงโดยใช้คู่เลขจำนวนเต็ม**

ภายในรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน จะเป็นเลขจำนวนเต็มแบบเกาส์เซียนซึ่งเป็นจำนวนเชิงซ้อนชนิดหนึ่งที่ส่วนจริงและส่วนจินตภาพเป็นเลขจำนวนเต็ม ดังนั้นเราสามารถใช้ประโยชน์โดยเก็บเป็นคู่เลขจำนวนเต็มซึ่งสามารถแปลงให้อยู่ในรูปแบบบิตได้ เช่น จำนวนเต็มแบบเกาส์เซียนที่อยู่บนรหัสบล็อกคือ 1+2i เราแปลงได้เป็นคู่ตัวเลข (1, 2) ซึ่งคู่เลขจำนวนเต็มนี้สามารถใช้รหัสมาตรฐานต่าง ๆ ทำให้อยู่ในรูปแบบบิตเพื่อส่งไปยังช่องสัญญาณ และแปลงกลับโดยใช้กระบวนการย้อนกลับ

4.4 การแปลงเป็นสายข้อมูลในรูปแบบบิต

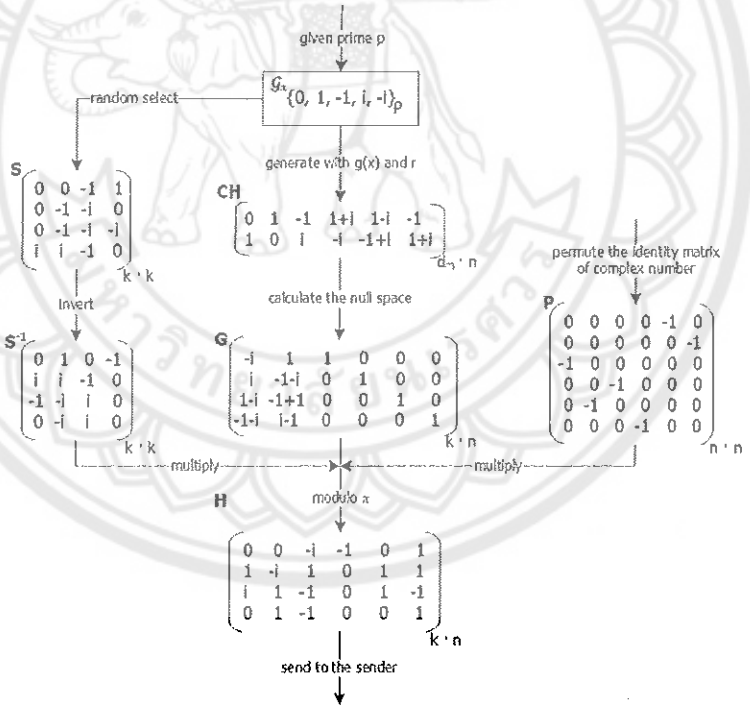
วิธีการนี้เป็นวิธีอย่างง่ายในการแปลงวัตถุที่อยู่บนโครงสร้างของภาษาต่าง ๆ ให้กลายเป็นสายข้อมูลในรูปแบบบิต (Serialization) และสามารถแปลงย้อนกลับมาเป็นวัตถุเดิม (Deserialization) ยกตัวอย่างเช่น โมดูล Pickle บนภาษา Python โดยข้อความเข้ารหัสของระบบเป็นเมทริกซ์ของจำนวนเชิงซ้อนซึ่งสามารถสร้างเป็นวัตถุบนภาษา Python จากนั้นใช้โมดูลนี้ในการแปลงวัตถุเป็นสายข้อมูลในรูปแบบบิต เพื่อส่งไปยังช่องสัญญาณ และแปลงกลับเป็นวัตถุเดิมจากสายข้อมูลนั้น

5. ตัวอย่างการเข้ารหัสลับแบบอสมมาตรตามกรอบแนวคิดงานวิจัย

ตัวอย่างการเข้ารหัสลับแบบอสมมาตรตามกรอบแนวคิดที่นำเสนอ บนรหัสบล็อคจำนวนเต็มแบบเกาส์เซียน $[6, 4, 3]_5$

5.1 การสร้างกุญแจ

การสร้างกุญแจลับและกุญแจสาธารณะของรหัส $[6, 4, 3]_5$ แสดงดังภาพ 10 เริ่มต้นด้วยการสร้างรหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน โดยผู้รับข้อความกำหนดค่าเริ่มต้น p = 5 r = 2 จากนั้นระบบกำหนดค่า $g(x) = x^2+x-i$ ซึ่งเป็นสมการพหุนามดีกรี r ทำให้ได้ค่า $\pi = 2+i$ n = 6 k = 4 และ $d_m = 3$ ซึ่งจะได้รหัสบล็อกสำหรับแก้ไขข้อผิดพลาดบนจำนวนเต็มแบบเกาส์เซียน $[6, 4, 3]_5$ โดยมีสมาชิก α บนฟิลด์ $\mathcal{G}_{2+i}$ จำนวน 4 ตัว ดังตารางที่ 10 จากนั้นสร้างฟิลด์ $\mathcal{G}_{(2+i)^2}$ โดยใช้ $g(x) = x^2+x-i$ ได้สมาชิก β จำนวน 24 ตัว ดังตารางที่ 11



ภาพ 10 ตัวอย่างการสร้างกุญแจลับและกุญแจสาธารณะของรหัส $[6, 4, 3]_5$

จากนั้นระบบจะสร้างกุญแจลับ โดยเริ่มจากสร้างเมทริกซ์ตรวจสอบความผิดพลาด CH จาก β ซึ่งเป็นสมาชิกของ $\mathcal{G}_{(2+i)^2}$ และเมทริกซ์ G จากปริภูมิว่างของ CH ต่อมาสร้างเมทริกซ์ S จากการสุ่มเลือกสมาชิกของ $\mathcal{G}_{2+i}$ และหาอินเวิร์ส S^{-1} สร้างเมทริกซ์ P โดยการเรียง

สับเปลี่ยนเมทริกซ์เอกลักษณ์ของจำนวนเต็มแบบเกาส์เซียน สุดท้ายระบบจะสร้างกุญแจสาธารณะ H โดยใช้กุญแจลับ G ร่วมกับคุณสมบัติการสุ่มและเรียงสับเปลี่ยนของกุญแจลับ S^{-1} และ P

ผู้รับข้อความจะต้องไม่เปิดเผยกุญแจลับทั้งหมดเพื่อป้องกันการโจมตี ส่วนกุญแจสาธารณะสามารถเผยแพร่ได้แบบสาธารณะ ผู้ที่ต้องการส่งข้อความลับมายังผู้รับข้อความจะใช้กุญแจสาธารณะนี้ในการเข้ารหัสข้อความ

ตาราง 10 สมาชิกของฟิลด์ $\mathcal{G}_{2+i}$

s	α^s	s	α^s	s	α^s	s	α^s
0	1	1	-i	2	i	3	-1

ตาราง 11 สมาชิกของฟิลด์ $\mathcal{G}_{(2+i)^2}$ โดยใช้ $g(x) = x^2+x-i$

s	β^s	s	β^s	s	β^s	s	β^s
0	[0, 1]	6	[0, -i]	12	[0, -1]	18	[0, i]
1	[1, 0]	7	[-i, 0]	13	[-1, 0]	19	[i, 0]
2	[-1, i]	8	[1, 1]	14	[1, -i]	20	-i, -1]
3	[1+i, -i]	9	[1-i, -i]	15	[-1+i, i]	21	[-1+i, 1]
4	[1-i, -1+i]	10	[-1-i, 1+i]	16	[-1+i, 1-i]	22	[1+i, -1-i]
5	[-1, 1+i]	11	[i, 1-i]	17	[1, -1-i]	23	[-i, -1+i]

5.2 การเข้ารหัส

รหัสบล็อกบนจำนวนเต็มแบบเกาส์เซียน $[6, 4, 3]_5$ สามารถสร้างข้อความรหัสที่แตกต่างกันจำนวน $p^k = 5^4$ รหัสต่อหนึ่งบล็อก ซึ่งสามารถจับคู่ไปยังรหัสฐานสองจำนวน $\log_2 p^k =$ 9 บิตต่อหนึ่งบล็อก มีค่า Code rate คือ $\log_2 p^k / n = 1.5$ บิตต่อรหัสหนึ่งตัว (Symbol) บนบล็อก นั่นคือการเข้ารหัสลับจำนวนเต็มแบบเกาส์เซียนนี้ ผู้ส่งจะสามารถส่งข้อความได้มากขึ้นต่อหนึ่งบล็อก

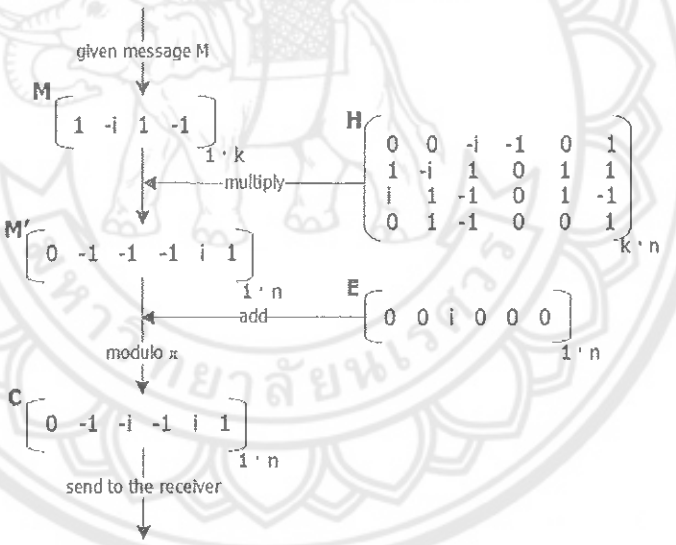
การเข้ารหัสของรหัส $[6, 4, 3]_5$ เริ่มต้นจากการแปลงรหัสฐานสองของข้อความ M_2 เป็นรหัสจำนวนเต็มแบบเกาส์เซียน M_{2+i} โดยใช้ตัวเลขระบบฐาน p เช่น รหัสฐานสองของ

ข้อความ M_2 = [0, 1, 0, 1, 1, 1, 0, 0, 0] จำนวน 9 บิต ใช้วิธีการของ Horner แปลงเป็นรหัสฐาน p ได้ M_5 = [1, 2, 1, 4] จากตารางที่ 12 ซึ่งเป็นตารางจับคู่เลขฐาน p กับสมาชิกบนฟิลด์ $\mathcal{G}_{2+i}$ ที่อ้างอิงมาจากตารางที่ 10 โดยจับคู่ได้เป็น M_{2+i} = [1, -i, 1, -1]

ตาราง 12 การจับคู่เลขฐาน p = 5 กับ $\mathcal{G}_{2+i}$

p	$\mathcal{G}_{2+i}$	p	$\mathcal{G}_{2+i}$	p	$\mathcal{G}_{2+i}$	p	$\mathcal{G}_{2+i}$	p	$\mathcal{G}_{2+i}$
0	0	1	1	2	-i	3	i	4	-1

จากนั้นเข้ารหัสข้อความโดยใช้เมทริกซ์กุญแจสาธารณะ H และเมทริกซ์ความผิดพลาด E แสดงดังภาพ 11



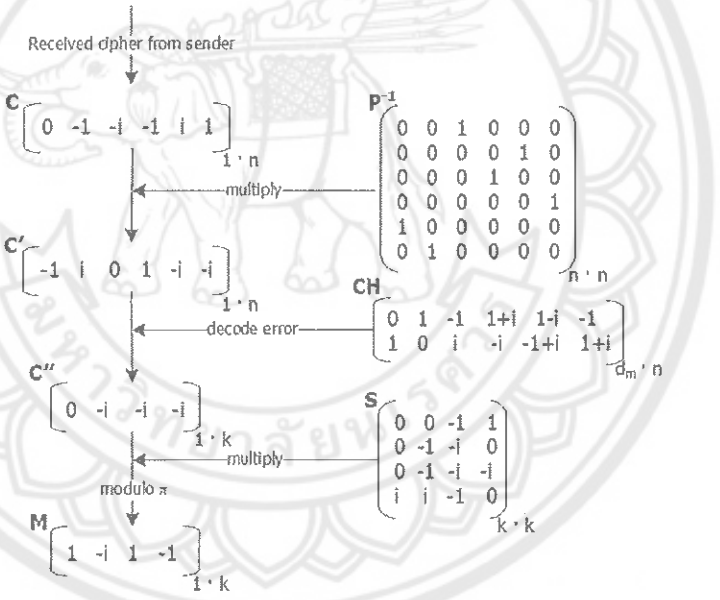
ภาพ 11 ตัวอย่างการเข้ารหัสข้อความของรหัส $[6, 4, 3]_5$

เมื่อเข้ารหัสเรียบร้อยแล้ว ข้อความเข้ารหัสนี้จะต้องถูกส่งผ่านช่องสัญญาณ โดยระบบจะแปลงรหัสจำนวนเต็มแบบเกาส์เซียน C_{2+i} กลับเป็นรหัสฐานสอง C_2 โดยวิธีการเดียวกัน คือ C_{2+i} =[0, -1, -i, -1, i, 1] จากตารางที่ 12 จับคู่สมาชิกบนฟิลด์ $\mathcal{G}_{2+i}$ กับตัวเลขฐาน p ได้เป็น C_5 = [0, 4, 2, 4, 3, 1] ใช้วิธีการของ Horner แปลงเป็นรหัสฐานสอง C_2 = [0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0] จำนวน $\log_2 p^n$ = 13 บิต

5.3 การถอดรหัส

ข้อความเข้ารหัสที่ส่งผ่านช่องสัญญาณมายังกระบวนการถอดรหัส จะอยู่ในรูปแบบรหัสฐานสอง ดังนั้นระบบจะทำการแปลงเป็นรหัสจำนวนเต็มแบบเกาส์เซียนก่อน โดยเริ่มต้น C_2 = [0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0] จะถูกแปลงเป็นรหัสฐาน p ได้เป็น C_5 = [0, 4, 2, 4, 3, 1] จากนั้นจับคู่เลขฐาน p กับสมาชิกบนฟิลด์ $\mathcal{G}_{2+i}$ โดยจับคู่ได้เป็น C_{2+i} =[0, -1, -i, -1, i, 1]

หลังจากแปลงเป็นรหัสจำนวนเต็มแบบเกาส์เซียนแล้ว การถอดรหัสของรหัส [6, 4, 3]$_5$ แสดงดังภาพ 12 โดยเริ่มต้นจากถอดรหัสการเรียงสับเปลี่ยนโดยใช้อินเวิร์สของเมทริกซ์ P ถอดรหัสความผิดพลาดโดยใช้เมทริกซ์ CH โดยได้ s = [-1+i, i] = β^{15} ได้ตำแหน่งที่ผิดพลาด l = 15 mod 6 = 3 และค่าผิดพลาด β^{15-3} = -1 เมื่อแก้ไขความผิดพลาดเรียบร้อยแล้ว สุดท้ายถอดรหัสการสุ่มโดยใช้เมทริกซ์ S ได้เป็นรหัสบล็อกจำนวนเต็มแบบเกาส์เซียน M_π



ภาพ 12 ตัวอย่างการถอดรหัสข้อความของรหัส [6, 4, 3]$_5$

ผู้รับข้อความแปลง M_π กลับเป็นข้อความโดยใช้วิธีการย้อนกลับจากแปลงข้อความในกระบวนการเข้ารหัสคือจับคู่ M_{2+i} = [1, -i, 1, -1] เป็นข้อความบนรหัสฐานห้า คือ M_5 = [1, 2, 1, 4] จากนั้นใช้วิธีการของ Horner แปลงกลับเป็นรหัสฐานสอง M_2 = [0, 1, 0, 1, 1, 1, 0, 0, 0]

การพัฒนาระบบ

ระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน หลังจากผู้วิจัยได้ทำการออกแบบกรอบแนวคิดและอัลกอริทึม ผู้วิจัยได้พัฒนาระบบบนภาษา Python โดยสร้างฟังก์ชันขึ้นใหม่จากการออกแบบอัลกอริทึม ร่วมกับการใช้ฟังก์ชันบนไลบรารี่เฉพาะ อย่าง Numpy และ Sympy สำหรับการคำนวณที่ใช้สมการพหุนามและเมทริกซ์

ผู้วิจัยพัฒนาระบบโดยการสร้างคลาส McElieceExtendMannheimCode ซึ่งฟังก์ชันหลักของระบบประกอบด้วย 5 ส่วนหลักคือ 1) init 2) createPtivatekey 3) createPublicKey 4) encryptM และ 5) decryptM แสดงดังภาพ 13 และฟังก์ชันอื่น ๆ ที่ทำงานภายใต้ฟังก์ชันหลักและคลาสแสดงดังภาพที่ 14 ถึง 16 โดยรายละเอียดของแต่ละฟังก์ชันแสดงในภาคผนวก ก

```
1   #McEliece extend Mannheim code
2   import numpy as np
3   import sympy as sp
4   from sympy.abc import x
5   import math
6   import secrets
7   import random
8   from operator import add
9   #-------------------------------------------------
10  __author__ = "Wanarat J"
11  __email__ = "wanaratj150gmail.com"
12  __date__ = "2018-08-5"
13  __updated__ = "2018-10-15"
14  #-------------------------------------------------
15  class McElieceExtendMannheimCode:
16      #initial
17      def __init__(self,p,pi,r):
31
32      #1. Create Key
33      #1.1 Create pivate key
34      def createPrivateKey(self,G,S,P,CH):
44
45      #1.2 Create public key
46      def createPublicKey(self,G,S,P):
61
62      #2. Encryption
63      def encryptM(self,m,pKey,e):
90
91      #3. Decryption
92      def decryptM(self,enM,P,S,CH):
```

ภาพ 13 ฟังก์ชันหลักของระบบบนภาษา Python

```
120
129      #Create Key: get primitive element of field
130      def getPrimitiveElementOfGF(self):
155
156      #Create Key: get g(x)
157      def getPrimitivePoly(self):
168
169      #Create Key: get primitive element of g(x)
170      def getPriEleWithRootOfPoly(self):
191
192      #Create Key: calculate each primitive element of g(x)
193      def calPriGFOfPoly(self, prev,alpha 1,alpha r):
219
220      #Create Key: create S matrix
221      def createScramblerMatrix(self):
243
244      #Create Key: create P matrix
245      def createPermutationMatrix(self):
262
263      #Create Key: translate element to primitive element
264      def tranToPriEleForCal(self,aList):
271
272      #Create Key: create CH
273      def createParityCheckMatrix(self):
276
277      #Create Key: create G
278      def createGeneratorMatrix(self):
287
```

ภาพ 14 ฟังก์ชันที่ทำงานภายใต้ฟังก์ชัน createPrivateKey และ createPublicKey

```
287
288      #Decryption: calculate syndrome s of code by CH
289      def decodeSyndrom(self,enM,CH):
303
304      #Decryption: check the syndrome is primitive element
305      def checkIsInPriEle(self,aList):
311
312      #Decryption: find error position and value of code
313      def checkSyndromPositionAndValue(self,check pos):
```

ภาพ 15 ฟังก์ชันที่ทำงานภายใต้ฟังก์ชัน decryptM

```
340      #Modulo operation of code over Gaussian integer
341      def divModMannheim(self,c1,c2):
349
350      #Inverse operation of Gaussian integer matrix
351      def calInverse(self,M):
353
354      #Round operation of Gaussian integer
355      def round(self,c):
```

ภาพ 16 ฟังก์ชันอื่น ๆ ที่ทำงานภายในคลาส

การประเมินผล

1. การทดสอบประสิทธิภาพด้านความถูกต้อง

การทดสอบในส่วนนี้ เป็นการทดสอบว่ารหัสบนจำนวนเต็มเกาส์เซียนสามารถทำงานบนอัลกอริทึมที่ออกแบบได้หรือไม่ ซึ่งผู้วิจัยใช้ Pytest สร้างกรณีทดสอบ (Test case) เพื่อทดสอบ

อัลกอริทึมของระบบ โดยประเมินจากความถูกต้อง (Correctness) ในการสร้างกุญแจ การเข้ารหัสและถอดรหัสของกรณีทดสอบ ความถูกต้องคำนวณได้จากสมการ

$$\text{Correctness} = \frac{\text{Number of trials fulfilled}}{\text{Total number of trials}} \times 100$$

2. **การทดสอบประสิทธิภาพด้านความปลอดภัย**

การทดสอบประสิทธิภาพด้านความปลอดภัยของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน การประเมินผลทำได้โดยใช้ค่าความซับซ้อน (Complexity cost) ในการพยายามคาดเดากุญแจลับและความพยายามในการถอดรหัสข้อความที่ระบบสร้างขึ้น

2.1 **การโจมตีเพื่อหากุญแจลับ**

โดยการโจมตีมุ่งเน้นไปที่การค้นหากุญแจลับของระบบ ซึ่งผู้โจมตีพยายามค้นหากุญแจที่เป็นไปได้จนกระทั่งพบกุญแจที่ถูกต้อง ค่าความซับซ้อนของการโจมตี สามารถวัดได้จากการพยายามคาดเดากุญแจลับ S และ P ดังนี้

2.1.1 N_p คือ จำนวนความพยายามในการคาดเดากุญแจ P ซึ่งการคาดเดาเมทริกซ์เรียงสับเปลี่ยนขนาด $n \cdot n$ ที่สร้างขึ้นโดยใช้สมาชิก $\{1, -1, i, -i\}$ เนื่องจากเป็นเมทริกซ์ของจำนวนเชิงซ้อน ใช้ความพยายาม $4n!$ และจากวิธีการประมาณของ Stirling (Scheinerman, 2012) ในการคำนวณค่าแฟกทอเรียล จะได้การคำนวณดังสมการ

$$N_p \approx 4\sqrt{2\pi n}\, n^n e^{-n}$$

2.1.2 N_s คือ จำนวนความพยายามในการคาดเดากุญแจ S ซึ่งการคาดเดาเมทริกซ์ขนาด $k \cdot k$ สร้างขึ้นโดยการใช้สมาชิกบน $\mathcal{G}_\pi$ ที่มีจำนวนสมาชิก p ตัว โดยใน 1 แถว จะใช้ความพยายาม p^k ครั้ง และแต่ละแถวของเมทริกซ์อิสระต่อกัน จึงต้องใช้ความพยายาม $\prod_{i=0}^{k-1}(p^k)$ และเนื่องจาก S ต้องเป็นเมทริกซ์ที่อินเวิร์สได้ ดังนั้นใช้ความพยายาม $\prod_{i=0}^{k-1}(p^k - p^i)$ สุดท้ายใช้วิธีการประมาณค่า (Hooshmand et al., 2014) เพื่อให้ง่ายต่อการคำนวณดังสมการ

$$N_s \leq p^{k(k+1)-1}$$

2.2 การโจมตีเพื่อถอดรหัสข้อความ

การโจมตีมุ่งเน้นไปที่การถอดรหัสข้อความ ซึ่งผู้โจมตีพยายามถอดรหัสข้อความที่เข้ารหัสที่เพื่อหาข้อความจริงที่ส่งผ่านระบบโดยที่ไม่ทราบกุญแจลับ ค่าความซับซ้อนของการโจมตี สามารถวัดได้จากจำนวนความพยายามในการถอดรหัสข้อความ โดยแบ่งการโจมตีเป็น 3 ประเภท คือ 1. Exhaustive comparison 2.Syndrome decoding และ 3. Information set decoding (รายละเอียดในบทที่ 2) ดังนี้

2.2.1 WF_{EC} คือจำนวนความพยายามในการคาดเดาข้อความโดยวิธี Exhaustive comparison ซึ่งการคาดเดาข้อความที่เป็นไปได้ทุก ๆ ข้อความ จำนวนความพยายามคำนวณดังสมการ

$$WF_{EC} = 2^{n \log_2 p}$$

2.2.2 WF_{SD} คือจำนวนความพยายามในการคาดเดาข้อความโดยวิธี Syndrome decoding ซึ่งพยายามคาดเดาเมทริกซ์ความผิดพลาด E จำนวนความพยายามคำนวณดังสมการ

$$WF_{ED} = \sum_{i=0}^{t} \binom{n \log_2 p}{i \log_2 p}$$

2.2.3 WF_{ISD} คือจำนวนความพยายามในการคาดเดาข้อความโดยวิธี Information set decoding ซึ่งพยายามสุ่มเลือกตำแหน่งที่คาดว่าจะใส่ความผิดพลาดลงในข้อความ จำนวนความพยายามคำนวณดังสมการ

$$WF_{ISD} \approx k^3 \left(1 - \frac{t}{n}\right)^{-k}$$

3. การทดสอบประสิทธิภาพด้านการประมวลผล

การทดสอบประสิทธิภาพด้านการประมวลผลของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน จะใช้วิธีการสองแบบคือ 1.การทดสอบเชิงทฤษฎี โดย

การคำนวณความซับซ้อนของระบบและแสดงเป็นค่า Big O และ 2.การทดสอบเชิงประจักษ์ โดยวัดเวลาจริงที่ระบบใช้ในการทำงาน

บทที่ 4

ผลการวิจัย

ในบทนี้จะอธิบายถึงผลการดำเนินการวิจัยโดยประกอบด้วย 3 ส่วน คือ ผลการทดสอบประสิทธิภาพด้านความถูกต้อง ผลการทดสอบประสิทธิภาพด้านความปลอดภัย และผลการทดสอบประสิทธิภาพด้านการประมวลผล ซึ่งจะอธิบายดังต่อไปนี้

ผลการทดสอบประสิทธิภาพด้านความถูกต้อง

ในการการทดสอบประสิทธิภาพด้านความถูกต้องเป็นการหาค่าเฉลี่ยของความถูกต้องของการทำงานบนอัลกอริทึมซึ่งคือความสามารถในการเข้ารหัสและถอดรหัสข้อความได้อย่างถูกต้องโดยใช้กุญแจที่สร้างขึ้น ผู้วิจัยสร้างกรณีทดสอบ(Test case) โดยใช้การสุ่มกุญแจที่ไม่ซ้ำกัน 100 คู่ และกุญแจแต่ละคู่จะใช้ในการเข้ารหัสและถอดรหัสบนข้อความและค่าความผิดพลาดที่สุ่มขึ้นจำนวน 1000 กรณี จากนั้นทดสอบบนค่า p และการเลือกช่วงสมาชิกเพื่อสร้างกุญแจลับ S ที่มีค่าต่ำในช่วงลำดับที่แตกต่างกันเพื่อให้ระบบรองรับการทำงานบนฮาร์ดแวร์ที่มีประสิทธิภาพไม่สูงนัก

ตาราง 13 ร้อยละค่าเฉลี่ยของความถูกต้องของระบบการเข้ารหัสลับแบบอสมมาตร โดยใช้จำนวนเต็มแบบเกาส์เซียน

Member Range	p = 421	p = 1013	p = 1861	p = 2381
First 10%	100	100	99.9970	100
First 20%	100	99.9990	100	100
First 50%	99.9990	99.9980	100	100
All	100	99.9990	99.9990	99.9990

ประสิทธิภาพด้านความถูกต้องของระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียน ที่ค่าพารามิเตอร์ p เท่ากับ 421 1013 1861 และ 2381 ที่มีการเลือกช่วงสมาชิกเพื่อสร้างกุญแจลับ S ที่มีค่าต่ำในช่วง 10 20 และ 50 เปอร์เซ็นแรกของสมาชิกและใช้สมาชิกทั้งหมด แสดงดังตารางที่

13 ซึ่งพบว่ามีค่าเฉลี่ยของความถูกต้องเป็นที่น่าพอใจ โดยมากกว่าร้อยละ 99.9970 สามารถเข้ารหัสและถอดรหัสได้ถูกต้องในทุกกรณีทดสอบ และพบว่าความผิดพลาดที่เกิดขึ้นมาจากการปัดค่าของการมอดุลาร์ในกระบวนการถอดรหัสมีความผิดพลาดในบางกรณี

ผลการทดสอบประสิทธิภาพด้านความปลอดภัย

ในการการทดสอบประสิทธิภาพด้านความปลอดภัย ผู้วิจัยใช้ขนาดของกุญแจสาธารณะเป็นตัวกลางในการเปรียบเทียบระหว่างระบบการเข้ารหัสเดิมของ McEliece กับระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียน โดยกุญแจสาธารณะเป็นเมทริกซ์ขนาด k·n ที่มีสมาชิกในเมทริกซ์เป็นฟิลด์จำนวนเต็มแบบเกาส์เซียนที่มีจำนวน p ตัว สมาชิกในเมทริกซ์จึงถูกมองว่าเป็นสัญลักษณ์ (Symbol) ในขณะที่ระบบการเข้ารหัสเดิมของ McEliece สมาชิกในเมทริกซ์ถูกเก็บเป็นรหัสแบบฐานสอง ผู้วิจัยจึงได้คำนวณขนาดกุญแจของระบบที่นำเสนอไปเป็นรหัสแบบฐานสอง ดังสมการ

$$\text{Size of } G_{pub} = k\cdot(n\cdot\log_2 p) \text{ bits}$$

ผลการคำนวณขนาดของกุญแจสาธารณะ แสดงดังตารางที่ 14 โดยคำนวณขนาดของกุญแจที่ค่า p เท่ากับ 421 1013 1861 และ 2381 ซึ่งแสดงให้เห็นว่าค่า p ที่สูงขึ้น ทำให้กุญแจมีขนาดเพิ่มขึ้น

ตาราง 14 ขนาดกุญแจของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียน

p	421	1013	1861	2381
Public key	12 kB	79 kB	296 kB	529 kB

1. **การโจมตีเพื่อหากุญแจลับ**

ประสิทธิภาพด้านความปลอดภัยของการโจมตีเพื่อค้นหากุญแจลับระหว่างระบบการเข้ารหัสเดิมของ McEliece เปรียบเทียบกับระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียน แสดงดังตารางที่ 15 โดยผู้วิจัยได้เลือกเฉพาะสมาชิกที่มีค่าต่ำในช่วงลำดับต้นของฟิลด์ $\mathcal{G}_\pi$ ในการสร้างกุญแจลับ S เพื่อให้ระบบรองรับการทำงานบนฮาร์ดแวร์ที่มีประสิทธิภาพไม่สูงนัก จะเห็นว่าที่กุญแจขนาดใกล้เคียงกัน ระบบแบบเดิมมีค่าความซับซ้อนของการค้นหากุญแจลับ S คือ $\approx 2^{275099}$

ในขณะที่ระบบที่พัฒนาขึ้นโดยใช้บางส่วนของฟิลด์เท่านั้น ที่ค่าพารามิเตอร์ p =1013 มีค่าความซับซ้อน คือ $\approx 2^{421139}$ ซึ่งมากกว่าระบบเดิมอย่างมาก ส่วนค่าความซับซ้อนของการค้นหากุญแจลับ P แม้ว่าระบบบนเลขจำนวนเต็มเกาส์เซียนจะสามารถเลือกสมาชิกเพื่อสร้างเมทริกซ์เรียงสับเปลี่ยนได้มากกว่าระบบบนจำนวนเต็มแบบเดิม แต่เนื่องจากขนาดและมิติของเมทริกซ์ที่ใช้สร้างรหัสมีขนาดน้อยกว่า ค่าความซับซ้อนจึงน้อยลงไปด้วย อย่างไรก็ตามเมื่อคำนวณผลรวมของค่าความซับซ้อนของกุญแจลับทั้งสอง ระบบที่พัฒนาขึ้นยังคงมีค่าความซับซ้อนของการโจมตีที่มากกว่าระบบเดิม

2. การโจมตีเพื่อถอดรหัสข้อความ

ประสิทธิภาพด้านความปลอดภัยของการโจมตีเพื่อถอดรหัสข้อความระหว่างระบบการเข้ารหัสเดิมของ McEliece เปรียบเทียบกับระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียนแสดงดังตารางที่ 16 จะเห็นว่าการโจมตีแบบ Exhaustive decoding บนระบบที่นำเสนอที่ p = 421 ซึ่งมีขนาดของกุญแจน้อยกว่าระบบเดิมมาก มีค่าความซับซ้อนของการโจมตีเพิ่มจากเดิมได้ถึง 2^{945} อย่างไรก็ตามการใช้คุณลักษณะของจำนวนเต็มแบบเกาส์เซียนบนการโจมตีแบบ Syndrome decoding และ Information set decoding นั้นยังมีประสิทธิภาพน้อยกว่าการโจมตีแบบแรก ยกตัวอย่างเช่น ต้องใช้ p = 1861 บนการโจมตีแบบ Information set decoding จึงจะได้ค่าความซับซ้อนของการโจมตีสูงกว่า ซึ่งต้องเพิ่มขนาดกุญแจกว่า 4 เท่าจากระบบเดิม

ตาราง 15 จำนวนความพยายามในการคาดเดากุญแจลับ P และ S ของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนเปรียบเทียบกับระบบเดิม

Cryptosystem	Original McEliece	Gaussian McEliece		
$(n, k)_p$	(1024, 524)	$(105,103)_{421}$	$(253, 251)_{1013}$	$(465,463)_{1861}$
Public key size	67 kB	12kB	79kB	296kB
N_S	$\approx 2^{275099}$	$\approx 2^{57757}$	$\approx 2^{421139}$	$\approx 2^{1427306}$
N_P	$\approx 2^{8770}$	$\approx 2^{547}$	$\approx 2^{1646}$	$\approx 2^{3440}$
N_{Total}	$\approx 2^{283869}$	$\approx 2^{58304}$	$\approx 2^{422785}$	$\approx 2^{1430746}$

ตาราง 16 จำนวนความพยายามในการถอดรหัสข้อความของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนเปรียบเทียบกับระบบเดิม

Cryptosystem	Original McEliece	Gaussian McEliece		
(n, k)	(1024, 524)	$(105,103)_{421}$	$(253, 251)_{1013}$	$(465,463)_{1861}$
Public key size	67 kB	12kB	79kB	296kB
WF_{EC}	2^{524}	2^{945}	2^{2530}	2^{5115}
WF_{SD}	$\approx 2^{284}$	$\approx 2^{125}$	$\approx 2^{164}$	$\approx 2^{201}$
WF_{ISD}	$\approx 2^{64.2}$	$\approx 2^{55.6}$	$\approx 2^{62.6}$	$\approx 2^{68.6}$

ผลการทดสอบประสิทธิภาพด้านการประมวลผล

1 **การทดสอบเชิงทฤษฎี**

การประมวลผลในเชิงทฤษฎี ผู้วิจัยใช้การคำนวณค่าความซับซ้อนไปยังกระบวนการหลัก 3 กระบวนการของระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียน ได้ค่าความซับซ้อนดังนี้

1.1 **ค่าความซับซ้อนของกระบวนการสร้างกุญแจ** (Com_K) คำนวณจากอัลกอริทึมที่ 1 ในตารางที่ 5 ได้ดังนี้

$$Com_K = Com_{con} + Com_{pri} + Com_{pub}$$

โดยที่ $Com_{con} = O(p(n-k))$ คือค่าความซับซ้อนในการสร้างรหัสบนฟิลด์จำนวนเต็มแบบเกาส์เซียน $Com_{pri} = O(n) + O(n-k) + O(n) + O(k^2) + O(n^3) + O(k^3)$ คือค่าความซับซ้อนในการพารามิเตอร์ลับ ประกอบด้วยเมทริกซ์ CH เมทริกซ์ G เมทริกซ์ P เมทริกซ์ S เมทริกซ์ P^{-1} และเมทริกซ์ S^{-1} ตามลำดับ และ $Com_{pub} = O(k^2n) + O(n^2k)$ คือค่าความซับซ้อนในการสร้างกุญแจสาธารณะ

1.2 **ค่าความซับซ้อนของกระบวนการเข้ารหัส** (Com_E) คำนวณจากอัลกอริทึมที่ 2 ในตารางที่ 6 ได้ดังนี้

$$Com_E = Com_c + Com_m(M_\pi G_{pub}) + Com_a(E)$$

โดยที่ $Com_c = O(k)$ คือค่าความซับซ้อนในการแปลงรหัสเป็นจำนวนเต็มแบบเกาส์เซียน $Com_m(M_\pi G_{pub}) = O(kn)$ คือค่าความซับซ้อนในการคูณข้อความรหัสจำนวนเต็มแบบเกาส์เซียนกับกุญแจสาธารณะ และ $Com_a(E) = O(n\text{-}k)$ คือค่าความซับซ้อนในการบวกเมทริกซ์ความผิดพลาด E ไปยังข้อความรหัสจำนวนเต็มแบบเกาส์เซียน

1.3 ค่าความซับซ้อนของกระบวนการถอดรหัส (Com_D) คำนวณจากอัลกอริทึมที่ 3 ในตารางที่ 7 ได้ดังนี้

$$Com_D = Com_m(C_\pi P^{-1}) + Com_{de}(Y) + Com_m(Y'S) + Com_c$$

โดยที่ $Com_m(C_\pi P^{-1}) = O(n^2)$ คือค่าความซับซ้อนในการคูณข้อความรหัสจำนวนเต็มแบบเกาส์เซียนกับเมทริกซ์ P^{-1} $Com_{de}(Y) = O(p)$ คือค่าความซับซ้อนในการถอดรหัสความผิดพลาด $Com_m(Y'S) = O(k^2)$ คือค่าความซับซ้อนในการคูณข้อความรหัสจำนวนเต็มแบบเกาส์เซียนกับเมทริกซ์ S และ $Com_c = O(k)$ คือค่าความซับซ้อนในการแปลงรหัสจำนวนเต็มแบบเกาส์เซียนกลับเป็นรหัสฐานสอง

ตาราง 17 ค่าความซับซ้อนของกระบวนการหลักของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนเปรียบเทียบกับระบบเดิม

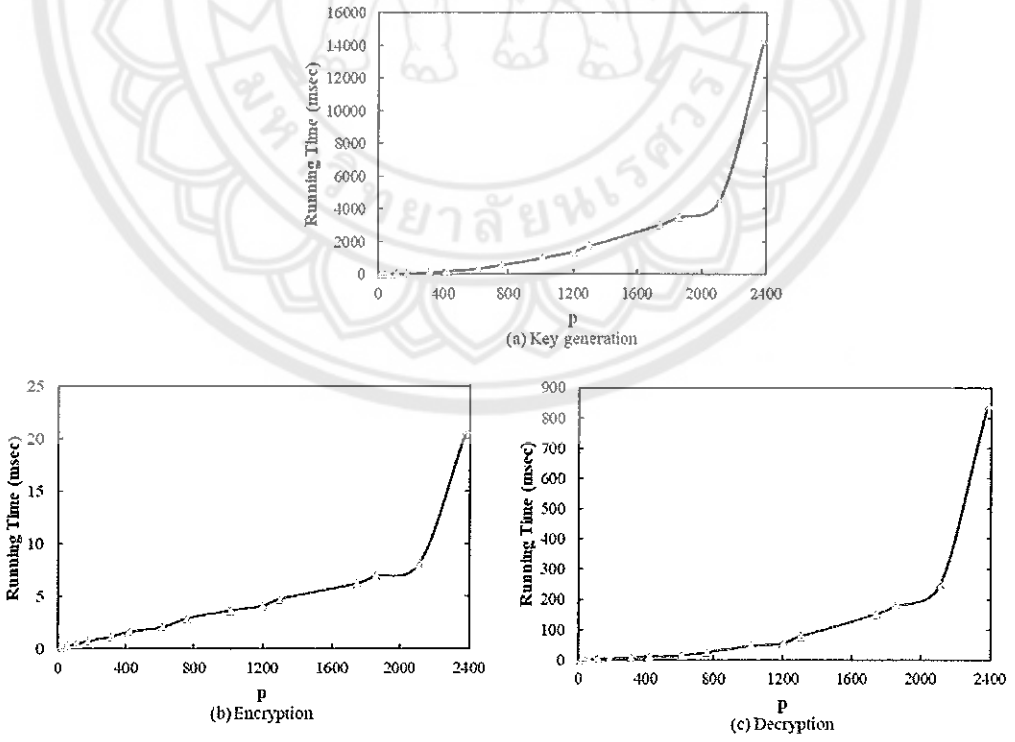
Cryptosystem	Original McEliece (Engelbert et al., 2007)	Gaussian McEliece
Key generation	$O((n\text{-}k)^3)$	$O(n^3)$
Encryption	$O(kn)$	$O(kn)$
Decryption	$O(k^2)$	$O(n^2)$

ค่าความซับซ้อนของกระบวนการสร้างกุญแจ เข้ารหัส และถอดรหัสของระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มแบบเกาส์เซียนโดยประมาณค่าเปรียบเทียบกับค่าความซับซ้อนของระบบการเข้ารหัสเดิมของ (Engelbert et al., 2007) แสดงดังตารางที่ 17 โดยพบว่ากระบวนการสร้างกุญแจมีค่าความซับซ้อนมากที่สุดในกระบวนการหลัก 3 กระบวนการ ซึ่งการสร้างเมทริกซ์ P เมทริกซ์ S และเมทริกซ์อินเวิร์สของทั้งสองเป็นกระบวนการที่ใช้เวลาค่อนข้างมาก เช่น มีค่าความซับซ้อน $O((n\text{-}k)^3)$ ในระบบการเข้ารหัสเดิม และ $O(n^3)$ ในระบบการ

เข้ารหัสแบบจำนวนเต็มเกาส์เซียน เป็นต้น รวมถึงกระบวนการถอดรหัสแบบจำนวนเต็มเกาส์เซียนมีค่าความซับซ้อน $O(n^2)$ ซึ่งมากกว่ากระบวนการแบบเดิมที่มีความซับซ้อน $O(k^2)$ อีกด้วย ดังนั้นการปรับปรุงอัลกอริทึมเพื่อให้มีประสิทธิภาพขึ้นจึงเป็นเรื่องที่น่าสนใจในการศึกษาในอนาคต

2. การทดสอบเชิงประจักษ์

การประมวลผลในเชิงประจักษ์ ผู้วิจัยใช้การวัดเวลาที่ระบบใช้ในการทำงาน (Running time) โดยใช้การรันบนคอมพิวเตอร์ส่วนบุคคลที่มีสภาพแวดล้อมของฮาร์ดแวร์และซอฟแวร์คือ CPU Intel Core i5 3210M 2.5 GHz. RAM 4 GB Windows10 64 bit และ Python 3.5 complier โดยมี Performance counter ซึ่งเป็นไลบรารี่ที่ช่วยวัดเวลาในการทำงานในระดับเศษส่วนวินาที (Fractional seconds) โดยแบ่งเป็น 3 กระบวนการของระบบการเข้ารหัสบนจำนวนเต็มแบบเกาส์เซียน คือ 1) ค่าเฉลี่ยเวลาในการทำงานของกระบวนการสร้างกุญแจ โดยคำนวณจากการสร้างรหัสจำนวนเต็มแบบเกาส์เซียนและกุญแจทั้งสองจำนวน 100 ครั้ง ที่พารามิเตอร์เดียวกัน 2) ค่าเฉลี่ยเวลาในการทำงานของกระบวนการเข้ารหัส และ 3) ค่าเฉลี่ยเวลาในการทำงานของกระบวนการถอดรหัส ของข้อความแบบสุ่มจำนวน 1000 ข้อความโดยใช้กุญแจเดียวกัน



ภาพ 17 ค่าเฉลี่ยเวลาที่ระบบใช้ในการทำงานของกระบวนการสร้างกุญแจ(a) การเข้ารหัส(b) และการถอดรหัส(c)

ผลการวิจัยแสดงดังภาพที่ 17 โดยพบว่าที่ค่า p เท่ากับ 421 1013 1861 และ 2381 ในกระบวนการสร้างกุญแจมีค่าเฉลี่ยเวลาในการทำงานเท่ากับ 0.19 1.01 3.51 และ 14.34 วินาที ตามลำดับ กระบวนการเข้ารหัสมีค่าเฉลี่ยเวลาในการทำงานเท่ากับ 1.63 3.66 6.94 และ 20.76 มิลลิวินาทีตามลำดับ สุดท้ายกระบวนการถอดรหัสมีค่าเฉลี่ยเวลาในการทำงานเท่ากับ 9.32, 46.26, 177.44 และ 842.32 มิลลิวินาทีตามลำดับ ผลการวิจัยแสดงให้เห็นว่าเมื่อใช้ค่า p ที่สูงขึ้น เพื่อเพิ่มประสิทธิภาพด้านความปลอดภัย จะทำให้ระบบต้องใช้เวลาในการประมวลผลเพิ่มขึ้น เช่นเดียวกัน

บทที่ 5

บทสรุป

สรุปผลการวิจัย

การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียนที่ศึกษาวิจัยนี้ เป็นแนวคิดใหม่ในการนำรหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติที่อยู่บนฟิลด์ของจำนวนเต็มแบบเกาส์เซียนมาใช้กับการเข้ารหัสลับแบบอสมมาตร โดยผู้วิจัยออกแบบระบบโดยมีพื้นฐานจากระบบ McEliece ดังนั้นระบบที่พัฒนาขึ้นจึงเป็นอีกทางเลือกหนึ่งในการเข้ารหัสลับแบบอสมมาตรที่จะป้องกันการถูกโจมตีด้วยอัลกอริทึมควอนตัม ซึ่งจากผลการทดสอบประสิทธิภาพด้านความถูกต้อง ด้านความปลอดภัย และด้านการประมวลผล สรุปได้ว่า ผลการดำเนินงานเป็นไปตามวัตถุประสงค์ของการวิจัย โดยพบว่ารหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิตินี้ สามารถใช้ได้บนระบบการเข้ารหัสลับแบบอสมมาตร มีประสิทธิภาพด้านความปลอดภัยมากขึ้นในการโจมตีเพื่อหากุญแจลับ มีประสิทธิภาพด้านความปลอดภัยใกล้เคียงกันในการโจมตีเพื่อถอดรหัสข้อความเมื่อเปรียบเทียบกับระบบแบบเดิม ส่วนประสิทธิภาพด้านการประมวลผลพบว่าการพยายามเพิ่มประสิทธิภาพด้านความปลอดภัยของระบบให้สูงขึ้น ทำให้มีประสิทธิภาพด้านการประมวลผลที่ลดลง

อภิปรายผลการวิจัย

จากการทดสอบประสิทธิภาพด้านความถูกต้อง พบว่ารหัสสำหรับแก้ไขข้อผิดพลาดแบบสองมิติโดยใช้จำนวนเต็มแบบเกาส์เซียน สามารถใช้ได้บนระบบการเข้ารหัสลับแบบอสมมาตรตามกรอบแนวคิดที่ได้ออกแบบไว้ โดยมีค่าเฉลี่ยความถูกต้องในทุกกรณีทดสอบมากกว่าร้อยละ 99.997

จากการทดสอบประสิทธิภาพด้านความปลอดภัย การเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียนเมื่อเปรียบเทียบกับการเข้ารหัสลับแบบเดิมที่ขนาดกุญแจใกล้เคียงกัน พบว่าระบบที่นำเสนอมีผลรวมของค่าความซับซ้อนที่สูงกว่าในส่วนของการโจมตีเพื่อหากุญแจลับ โดยสามารถเพิ่มค่าความซับซ้อนจาก $\approx 2^{283869}$ เป็น $\approx 2^{422785}$ ซึ่งสามารถช่วยเพิ่มประสิทธิภาพด้านความปลอดภัยได้อย่างมาก ในขณะที่การโจมตีเพื่อถอดรหัสข้อความพบว่าการโจมตีที่แตกต่างกัน 3 แบบ ให้ผลลัพธ์ที่ต่างกัน โดยระบบที่นำเสนอสามารถเพิ่มค่าความซับซ้อนของการโจมตีแบบ

Exhaustive decoding จาก 2^{524} เป็น 2^{2530} แต่กลับมีค่าความซับซ้อนที่น้อยกว่าในการโจมตีแบบ Syndrome decoding โดยลดจาก $\approx 2^{284}$ เป็น $\approx 2^{164}$ อย่างไรก็ตามการโจมตีที่นิยมใช้และมีประสิทธิภาพมากที่สุดอย่างการโจมตีแบบ Information set decoding มีค่าความซับซ้อนที่ใกล้เคียงกัน โดยระบบที่นำเสนอมีค่าความซับซ้อนที่ $\approx 2^{62.6}$ ส่วนระบบเดิมมีค่าความซับซ้อนที่ $\approx 2^{64.2}$ ดังนั้นการปรับปรุงโครงสร้างของรหัสบนจำนวนเต็มเกาส์เซียน จึงเป็นงานที่จะพัฒนาต่อไปในอนาคตเพื่อเพิ่มประสิทธิภาพด้านความปลอดภัยสำหรับการโจมตีเพื่อถอดรหัสข้อความให้ดียิ่งขึ้น

จากการทดสอบประสิทธิภาพด้านการประมวลผลของการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียนทั้งในเชิงทฤษฎีและเชิงประจักษ์พบว่ากระบวนการสร้างกุญแจใช้เวลาในการประมวลผลมากที่สุด ตามมาด้วยกระบวนการถอดรหัส โดยกระบวนการเข้ารหัสใช้เวลาในการประมวลผลน้อยที่สุด ผลการทดสอบยังพบว่าการเพิ่มประสิทธิภาพด้านความปลอดภัยโดยใช้ค่า p ที่สูงขึ้น จะทำให้ระบบต้องใช้เวลาในการประมวลผลเพิ่มขึ้นในกระบวนการหลักทั้งสามกระบวนการ

ข้อเสนอแนะ

1. การปรับปรุงโครงสร้างของรหัสบนจำนวนเต็มเกาส์เซียน เพื่อเพิ่มประสิทธิภาพด้านความปลอดภัยสำหรับการโจมตีเพื่อถอดรหัสข้อความ

2. การปรับปรุงอัลกอริทึมในการเข้ารหัสและถอดรหัส เพื่อเพิ่มประสิทธิภาพด้านการประมวลผลให้ดียิ่งขึ้น

บรรณานุกรม

บรรณานุกรม

Au, S., Eubanks-Turner, C., & Everson, J. (September 17, 2003). The McEliece Cryptosystem. Retrieved January 16, 2018, from http://www.math.unl.edu/~s-jeverso2/McElieceProject.pdf

Bagheri, K., & Sadeghi, M. R. (2015). A New Non-associative Cryptosystem Based on NTOW Public Key Cryptosystem and Octonions Algebra. *ACM Commun. Comput. Algebra, 49*(1), 13.

Bagheri, K., Sadeghi, M. R., & Panario, D. (2017). A Non-commutative Cryptosystem Based on Quaternion Algebras. *Designs, Codes and Cryptography, 86*(10), 2345-2377.

Baldi, M., Bianchi, M., & Chiaraluce, F. (2013). Security and complexity of the McEliece cryptosystem based on quasi-cyclic low-density parity-check codes. *IET Information Security, 7*(3), 212–220.

Baldi, M., Bodrato, M., & Chiaraluce, F. (2008). A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes. *Security and Cryptography for Networks 2018*, 246–262.

Baldi, M., Santini, P., & Chiaraluce, F. (2016). Soft McEliece: MDPC code-based McEliece cryptosystems with very compact keys through real-valued intentional errors. *2016 IEEE International Symposium on Information Theory*, 795-799.

Bosworth, S., & Kabay, M. E. (2002). *Computer Security Handbook*. Hoboken, New Jersey: John Wiley and Sons.

Caboara, M., Caruso, F., & Traverso, C. (2008). GröBner Bases for Public Key Cryptography. *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation*, 315–324.

Cherdantseva, Y., & Hilton, J. (2013). A Reference Model of Information Assurance Security. *2013 International Conference on Availability, Reliability and Security*, 546–555.

Dottling, N., Dowsley, R., Muller-Quade, J., & Nascimento, A. C. A. (2012). A CCA2 Secure Variant of the McEliece Cryptosystem. *IEEE Transactions on Information Theory, 58*(10), 6672–6680.

Dowsley, R., Müller-Quade, J., & Nascimento, A. C. A. (2009). A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model. *Topics in Cryptology CT-RSA 2009*, 240–251.

Elgamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory, 31*(4), 469–472.

Elkamchouchi, H., Elshenawy, K., & Shaban, H. A. (2003). Two new public key techniques in the domain of Gaussian integers. *Proceedings of the Twentieth National Radio Science Conference*, C17-1.

Engelbert, D., Overbeck, R., & Schmidt, A. (2007). A Summary of McEliece-Type Cryptosystems and their Security. *Journal of Mathematical Cryptology JMC, 1*(2), 151–199.

Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). NTRU: A Ring-Based Public Key Cryptosystem. *Proceedings of the Third International Symposium on Algorithmic Number Theory*, 267–288.

Hooshmand, R., Shooshtari, M. K., Eghlidos, T., & Aref, M. R. (2014). Reducing the key length of mceliece cryptosystem using polar codes. *2014 11th International ISC Conference on Information Security and Cryptology*, 104–108.

Hooshmand, R., Shooshtari, M. K., & Aref, M. R. (2017). PKC-PC: A Variant of the McEliece Public Key Cryptosystem based on Polar Codes. *Computing Research Repository (CoRR)*, 1712.07672, 1-11.

Huber, K. (1994). Codes over Gaussian integers. *IEEE Transactions on Information Theory, 40*(1), 207–216.

J. Mceliece, R. (1978). A Public-Key Cryptosystem Based on Algebraic Coding Theory. *JPL DSN Progress Report*, 44, 114–116.

Jarvis, K., & Nevins, M. (2015). ETRU: NTRU over the Eisenstein integers. *Designs, Codes and Cryptography, 74*(1), 219–242.

Jochemsz, E. (2002). *Goppa Codes & the McEliece Cryptosystem*. Amsterdam: Vrije Universiteit Amsterdam.

Kabatiansky, G., Krouk, E., & Semenov, S. (2005). *Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept*. Hoboken, New Jersey: Wiley.

Knuth, D. E. (1960). A imaginary number system. *Communications of the ACM, 3*(4), 245–247.

Koval, A. (2016). Algorithm for Gaussian Integer Exponentiation. In *Information Technology: New Generations* (pp. 1075–1085). New York: Springer.

Maconachy, W., Schou, C., Ragsdale, D., & Welch, D. (2001). A Model for Information Assurance: An Integrated Approach. *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, 306-310.

Malekian, E., & Zakerolhosseini, A. (2010). OTRU: A non-associative and high speed public key cryptosystem. *2010 15th CSI International Symposium on Computer Architecture and Digital Systems*, 83–90.

McCumber, J. (2004). *Assessing and Managing Security Risk in IT Systems: A Structured Methodology*. Boca Raton, Florida: CRC Press.

Miller, V. S. (1985). Use of Elliptic Curves in Cryptography. *Advances in Cryptology CRYPTO '85 Proceedings*, 417–426.

Misoczki, R., Tillich, J. P., Sendrier, N., & Barreto, P. S. L. M. (2013). MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes. *2013 IEEE International Symposium on Information Theory*, 2069–2073.

Mohamed, E., & Elkamchouchi, H. (2009). Elliptic Curve Cryptography over Gaussian Integers. *IJCSNS International Journal of Computer Science and Network Security, 9*, 413–416.

Monico, C., Rosenthal, J., & Shokrollahi, A. (2000). Using low density parity check codes in the McEliece cryptosystem. *2000 IEEE International Symposium on Information Theory*, 215.

Nanda, A. K., Nayak, R., & Awasthi, L. K. (2015). NTRU with Gaussian Integer Matrix. *International Journal of Advanced Research in Computer Science and Software Engineering, 5*(2), 359–365.

Nayak, R., Pradha, J., & Sastry, C. V. (2012). Evaluation of Performance Characteristics of Polynomial based and Lattice based NRTU Cryptosystem. *ACEEE International Journal on Network Security, 3*, 1–4.

Niederreiter, H. (1986). Knapsack Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory, 15*(2), 157-166.

Nyokabi, G. J., Salleh, M., & Mohamad, I. (2017). NTRU inverse polynomial algorithm based on circulant matrices using gauss-jordan elimination. *2017 6th ICT International Student Project Conference (ICT-ISPC)*, 1–5.

Oppliger, R. (2011). *Contemporary Cryptography*. Norwood, Massachusetts: Artech House.

Patarin, J. (1996). Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. *Advances in Cryptology — EUROCRYPT '96*, 33–48.

Pradhan, S. (2013). A Modified Variant of RSA Algorithm for Gaussian Integers. *Advances in Intelligent Systems and Computing, 236*, 183-187.

Repka, M., & Zajac, P. (2015). Overview of the Mceliece Cryptosystem and its Security. *Tatra Mountains Mathematical Publications, 60*(1), 57–83.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM, 21*(2), 120–126.

Scheinerman, E. A. (2012). *Mathematics: A Discrete Introduction*. Boston, Massachusetts: Cengage Learning.

Schneier, B. (August 1, 2006). Updating the Traditional Security Model. Retrieved June 14, 2017, from https://www.schneier.com/blog/archives/2006/08/updating_the_tr.html

Seroul, R. (2000). *Programming for Mathematicians*. New York: Springer-Verlag.

Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing, 26*(5), 1484–1509.

Shrestha, S. R., & Kim, Y. S. (2014). New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography. *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, 368–372.

Sidelnikov, V. M., & Shestakov, S. O. (1992). On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications, 2*, 439–444.

Sidelnikov, V. M. (1994). A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications, 4*(3), 191–208.

Thakur, K., Tripathi, B. P., & Yadav, M. R. (2017). KTRU: NTRU over the Kleinian Integers. *Journal of International Academy Of Physical Sciences, 20*(3), 177-183.

Trappe, W., & Washington, L. C. (2006). *Introduction to Cryptography: With Coding Theory.* Upper Saddle River, New Jersey: Pearson Prentice Hall.

Wagstaff, S. S. (2002). *Cryptanalysis of Number Theoretic Ciphers.* Boca Raton, Florida: CRC Press.

Yan, S. Y. (2013). *Computational Number Theory and Modern Cryptography.* Hoboken, New Jersey: John Wiley and Sons.

Zhao, N., & Su, S. (2011). An Improvement and a New Design of Algorithms for Seeking the Inverse of an NTRU Polynomial. *2011 Seventh International Conference on Computational Intelligence and Security*, 891–895.

ภาคผนวก

ภาคผนวก ก รายละเอียดของฟังก์ชันในระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียน

ตาราง 18 รายละเอียดของฟังก์ชันในระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียน

ชื่อฟังก์ชัน	คำอธิบาย
init	กำหนดพารามิเตอร์ตั้งต้น
createPrivateKey	สร้างกุญแจลับ
createPublicKey	สร้างกุญแจสาธารณะ
encryptM	เข้ารหัสข้อความ
decryptM	ถอดรหัสข้อความ
getPrimitiveElementOfGF	หาสมาชิกทั้งหมดของฟิลด์ดั้งเดิมบนจำนวนเต็มเกาส์เซียนจากพารามิเตอร์ที่กำหนดเบื้องต้น
getPrimitivePoly	หาสมการดั้งเดิมสำหรับสร้างฟิลด์บนจำนวนเต็มเกาส์เซียนจากพารามิเตอร์ที่กำหนดเบื้องต้น
getPriEleWithRootOfPoly	หาสมาชิกทั้งหมดของฟิลด์บนจำนวนเต็มเกาส์เซียนจากสมการดั้งเดิม
calPriGFOfPoly	คำนวณเพื่อหาสมาชิกของฟิลด์บนจำนวนเต็มเกาส์เซียนจากสมการดั้งเดิม
createScramblerMatrix	สร้างเมทริกซ์สุ่ม (พารามิเตอร์ลับ)
createPermutationMatrix	สร้างเมทริกซ์เรียงสับเปลี่ยน (พารามิเตอร์ลับ)
tranToPriEleForCal	แปลงสมาชิกให้อยู่ในฟิลด์บนจำนวนเต็มเกาส์เซียน
createParityCheckMatrix	สร้างเมทริกซ์ตรวจสอบความผิดพลาด (พารามิเตอร์ลับ)
createGeneratorMatrix	สร้างเมทริกซ์กุญแจลับ
decodeSyndrom	หาระยะความผิดพลาดบนข้อความที่ถูกเข้ารหัส
checkIsInPriEle	ตรวจสอบว่าเป็นสมาชิกที่อยู่ในฟิลด์บนจำนวนเต็มเกาส์เซียนหรือไม่
checkSyndromPositionAndValue	ตรวจสอบตำแหน่งและค่าความผิดพลาดบนข้อความที่ถูกเข้ารหัส

ตาราง 18 (ต่อ)

ชื่อฟังก์ชัน	คำอธิบาย
divModMannheim	มอดุลาร์จำนวนเต็มแบบเกาส์เซียน
callnverse	คำนวนอินเวิร์สของเมทริกซ์บนจำนวนเต็มแบบเกาส์เซียน
round	ปัดค่าจำนวนเต็มแบบเกาส์เซียน

ภาคผนวก ข ความหมายของสัญลักษณ์ในระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียน

ตาราง 19 ความหมายของสัญลักษณ์ในระบบการเข้ารหัสลับแบบอสมมาตรโดยใช้จำนวนเต็มเกาส์เซียน

สัญลักษณ์	คำอธิบาย
$\cdot$	คูณเมเทริกซ์
$+$	บวกเมเทริกซ์
$\binom{n}{i}$	จำนวนวิธีที่เลือก i ตำแหน่ง จากทั้งหมด n ตำแหน่ง
$\{\ldots\}$	เซต
$\lvert\ldots\rvert$	ค่าสัมบูรณ์
$\Gamma(L, g(x))$	รหัส Goppa ที่มีเซต L ประกอบด้วยสมาชิกที่ไม่ซ้ำกัน โดยสร้างจากสมการพหุนาม g(x)
$\alpha \in \mathcal{G}_\pi$	α เป็นสมาชิกของ $\mathcal{G}_\pi$
A^{-1}	อินเวิร์สเมทริกซ์ของ A
A^T	เมทริกซ์สลับเปลี่ยนของ A
$GF(p^m)$	ฟิลด์ที่มีสมาชิกจำนวน p^m ตัว
$Im(\ldots)$	ส่วนจินตภาพของจำนวนเชิงซ้อน
mod	มอดุลาร์จำนวนเต็มแบบเกาส์เซียน
$\mathbb{N}$	จำนวนนับ
[n, k, d]	รหัสที่มีข้อความเข้ารหัสขนาด n ข้อความนำเข้าขนาด k และระยะทาง Hamming ขนาด d
$[n, k, d_m]_p$	รหัสที่มีข้อความเข้ารหัสขนาด n ข้อความนำเข้าขนาด k ระยะทาง Mannheim ขนาด d_m และฟิลด์สมาชิกขนาด p
$Re(\ldots)$	ส่วนจริงของจำนวนเชิงซ้อน
$\mathbb{Z}$	จำนวนเต็ม
$\mathbb{Z}[i]$	จำนวนเต็มแบบเกาส์เซียน