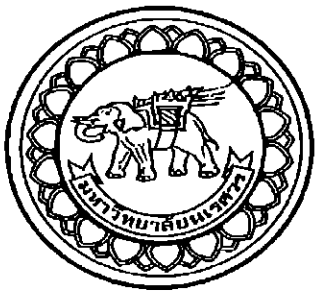


การเข้ารหัสลับโดยใช้การจัดเรียงและเพิ่ม Noise

A Cryptographer Using Permutation with Noise

นายพชร ศรีสุข รหัส 47362041

นางสาวสุธาทิพย์ คล้ายเหล็ง รหัส 47362231

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การเข้ารหัสลับโดยใช้การจัดเรียงและเพิ่ม Noise			
ผู้ดำเนินโครงงาน	นายพชร	ศรีสุข	รหัส	47362041
	นางสาวสุธาทิพย์	คล้ายเหลือง	รหัส	47362231
อาจารย์ที่ปรึกษา	ดร. สุรเดช	จิตประไพกุลศาล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(อาจารย์แสงชัย มังกรทอง)

.....กรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

หัวข้อโครงงาน	การเข้ารหัสลับโดยใช้การจัดเรียงและเพิ่ม Noise			
ผู้ดำเนินโครงงาน	นายพชร	ศรีสุข	รหัส	47362041
	นางสาวสุธาทิพย์	คล้ายเหลือง	รหัส	47362231
อาจารย์ที่ปรึกษา	ดร. สุรเดช	จิตประไพกุลศาล		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2550			

.....

บทคัดย่อ

โครงงานนี้เป็นการออกแบบและคิดค้นวิธีการที่ทำให้การเข้ารหัสลับและถอดรหัสลับที่มีอยู่แล้วปลอดภัยมากขึ้น โดยเพิ่มขั้นตอนวิธีการทำหลักๆอยู่ 3 ขั้นตอน คือ 1. Generate Key เพื่อนำไปใช้ในการเข้ารหัสลับและถอดรหัสลับ 2. เลือกอัลกอริทึมเพื่อใช้ในการเข้ารหัสลับในแต่ละครั้ง 3. เพิ่ม Noise เข้าไปใน Ciphertext โดยที่ค่าของ Noise และตำแหน่งของ Noise ในแต่ละ Ciphertext ไม่แน่นอน ซึ่งทำให้การเข้ารหัสลับมีความซับซ้อนมากยิ่งขึ้น และส่งผลให้ Cracker ต้องใช้เวลาในการคำนวณมากขึ้นเพื่อค้นหา Key ให้พบ จากการทดสอบแล้วพบว่า การเข้ารหัสลับแบบนี้สามารถถอดรหัสลับกลับมาเป็นข้อความตั้งต้นได้เหมือนเดิม

Project Title	A Cryptographer Using Permutation With Noise		
Name	Mr. Potchara	Srisuk	ID. 47362041
	Miss Sutatip	Khlaileng	ID. 47362231
Project Advisor	Dr. Suradet	Jitprapaikulsarn	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2007		

.....

ABSTRACT

In this project we enhance the available cryption algorithms by adding three steps: 1) generating a key; 2) select an algorithm from a set of specific algorithms; 3) adding noise to the ciphertext. These three enhancements make the overall cryptographer system more complex; hence, the attackers have to spend more time to find the key. Based on our experiment our approach can encrypt any testing plaintext and can decrypt the ciphertext back to the original plaintext.

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร.สุรเดช จิตประไพกุลศาล อาจารย์ที่ปรึกษาโครงงานนี้ ที่คอยให้คำปรึกษา ความช่วยเหลือตลอดจนคำแนะนำและแนวทางต่างๆ ในการทำโครงงานนี้ และสุดท้ายขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆ ทุกคนที่ยังไม่ได้เอ่ยนามที่คอยให้การสนับสนุนผู้ดำเนินโครงงาน ให้สามารถทำโครงงานนี้จนสำเร็จลุล่วงไปได้ด้วยดี

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูปภาพ

สารบัญรูปภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันนี้ ข้อมูลข่าวสาร ถือเป็นพลังงานที่สำคัญอย่างยิ่งในการขับเคลื่อนโลกในยุคปัจจุบัน เพราะฉะนั้นการติดต่อสื่อสารและการส่งข้อมูลจึงมีความต้องการในด้านความปลอดภัยในระดับสูง ซึ่งหนึ่งในวิธีที่ใช้เพิ่มความปลอดภัยนั่นก็คือการเข้ารหัสลับ สำหรับวิธีการเข้ารหัสลับในปัจจุบันก็มีอยู่ด้วยกันมากมายหลากหลายวิธี แต่ถึงจะมีวิธีเข้ารหัสลับหลายวิธี แต่ด้วยเทคโนโลยีในปัจจุบันที่มีความก้าวหน้าเป็นอย่างมาก ก็ทำให้ความสามารถในการถอดรหัสลับก็มีมากขึ้นเช่นกัน ดังนั้นจึงได้เกิดแนวความคิดที่จะพัฒนาการเข้ารหัสลับและถอดรหัสลับรูปแบบใหม่ขึ้นมา ซึ่งหนึ่งในแนวคิดที่พัฒนาขึ้นมานั้นก็คือ การเข้ารหัสลับโดยใช้การจัดเรียงและมีการเพิ่ม Noise เข้าไปที่ใน Ciphertext ซึ่งวิธีการเข้ารหัสลับรูปแบบนี้จะเป็นอีกทางเลือกหนึ่ง ที่ช่วยให้การจัดส่งข้อมูลมีความปลอดภัยยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสามารถเข้ารหัสลับและถอดรหัสลับ โดยใช้การจัดเรียงเพื่อในการเข้ารหัสลับ และเพิ่ม Noise ลงไปในข้อมูล
2. เพื่อเพิ่มความซับซ้อนในการเข้ารหัสลับและถอดรหัสลับ
3. เพื่อศึกษาอัลกอริทึมในการเข้ารหัสลับและถอดรหัสลับ
4. เพื่อสามารถนำมาประยุกต์ใช้งานในด้านความปลอดภัยด้านต่างๆ

1.3 ขอบข่ายของโครงงาน

1. ศึกษาทฤษฎีและออกแบบอัลกอริทึมของการเข้ารหัสลับและถอดรหัสลับ
2. ประยุกต์, ออกแบบและพัฒนา ทฤษฎีและอัลกอริทึมของการเข้ารหัสลับและถอดรหัสลับ
3. ศึกษาการใช้งานและการประยุกต์ใช้งานโปรแกรม Java

1.4 ขั้นตอนการดำเนินงาน

1. ค้นหาข้อมูลเกี่ยวกับการเข้ารหัสลับและถอดรหัสลับ

2. หาแนวคิดในการออกแบบการเข้ารหัสลับและถอดรหัสลับโดยที่ใช้การเลือกอัลกอริทึม และเพิ่ม Noise

3. คิดอัลกอริทึมของการเข้ารหัสลับและถอดรหัสลับโดยใช้การเลือกอัลกอริทึม และเพิ่ม Noise

4. ทดสอบอัลกอริทึมการเข้ารหัสลับและถอดรหัสลับ

5. ตรวจสอบและปรับปรุงแก้ไข

6. เขียนโปรแกรมการเข้ารหัสลับและถอดรหัสลับ

7. ตรวจสอบและปรับปรุงแก้ไขตัวโปรแกรม

1.5 แผนการดำเนินงาน

	ระยะเวลาดำเนินงาน											
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ค้นหาข้อมูลเกี่ยวกับการเข้ารหัสลับ	→	→	→	→	→							
2. หาแนวคิดในการออกแบบการเข้ารหัสลับแบบหลายมิติ		→	→	→	→	→	→	→				
3. คิดอัลกอริทึมของการเข้ารหัสลับแบบหลายมิติ						→	→	→	→			
4. เขียนมือโครงสร้างของการเข้ารหัสลับแบบหลายมิติ								→	→	→		
5. ตรวจสอบและปรับปรุงแก้ไข												
6. เขียนโปรแกรมการเข้ารหัสลับแบบหลายมิติ										→	→	→
7. ตรวจสอบและปรับปรุงแก้ไขตัวโปรแกรม												→

1.6 ผลที่คาดว่าจะได้รับ

1. ได้อัลกอริทึมของการเข้ารหัสลับและถอดรหัสลับโดยใช้การเลือกอัลกอริทึม และเพิ่ม Noise
2. สามารถเข้ารหัสลับและถอดรหัสลับได้อย่างถูกต้อง
3. เพิ่มความปลอดภัยในการเข้ารหัสลับและถอดรหัสลับมากขึ้น
4. สามารถนำทฤษฎีและอัลกอริทึมที่ได้นี้ไปประยุกต์ใช้และพัฒนาในงานด้านอื่นๆได้

1.7 งบประมาณที่ใช้

1. ค่าวัสดุสำนักงาน	เป็นเงิน	300	บาท
2. ค่าวัสดุคอมพิวเตอร์	เป็นเงิน	200	บาท
3. ค่าหนังสือ	เป็นเงิน	500	บาท
4. ค่าถ่ายเอกสาร	เป็นเงิน	300	บาท
5. ค่าวัสดุอื่นๆ	เป็นเงิน	700	บาท
	รวมเป็นเงิน	2,000	บาท **(สองพันบาทถ้วน)**

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทที่ 2 นี้ จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการทำโครงงานนี้ โดยนำเสนอเฉพาะเนื้อหาหลักที่ค่อนข้างมีความสัมพันธ์ที่นำมาประยุกต์ใช้เพื่อทำโครงงานนี้ และมีการตัดเนื้อหาบางส่วนของบางหัวข้อที่ไม่ค่อยมีความจำเป็นที่จะต้องใช้กับโครงงานนี้เท่าใดนัก ซึ่งหัวข้อที่จะกล่าวถึงนี้มีหัวข้อหลัก คือ วิทยาการเข้ารหัสลับ

2.1 ประวัติของวิทยาการเข้ารหัสลับ (History of Cryptography)

วิทยาการเข้ารหัสลับ ในความเป็นจริงนั้นมีมานานแล้ว จากหลักฐานการค้นพบครั้งสำคัญที่อาณาจักรอียิปต์ซึ่งเป็นอาณาจักรอันเก่าแก่นานกว่า 4000 ปีมาแล้ว ได้เป็นเครื่องพิสูจน์ให้เห็นชัดได้ว่าวิทยาการรหัสลับได้เริ่มนำมาใช้กันตั้งแต่ในสมัยนั้น ซึ่งในสมัยโบราณวิทยาการเข้ารหัสลับที่นำมาใช้ยังคงเป็นรูปแบบที่ง่ายไม่ซับซ้อน โดยเฉพาะเทคนิคการใช้ตัวอักษรในการนำมาสร้างเป็นรหัสลับ ทั้งนี้เพื่อให้ข้อมูลข่าวสารที่ต้องการติดต่อกัน เป็นความลับและไม่ให้ผู้อื่นรับรู้ข่าวสารได้ และได้พัฒนารูปแบบมาเรื่อยๆจนกระทั่งมีความซับซ้อนมากยิ่งขึ้น โดยเฉพาะในสมัยสงครามโลกครั้งที่ 2 ที่มีการนำหลักการทางวิทยาการเข้ารหัสลับมาใช้เป็นแนวทางในการผลิตเครื่อง Enigma ขึ้นมาสำหรับใช้ในการทำสงครามโลก ซึ่งถือได้ว่าเป็นจุดเปลี่ยนสำคัญของวิทยาการรหัสลับตั้งแต่เริ่มใช้กันมาเพราะมีวิธีการที่ซับซ้อนกว่ามากที่ผ่านมา จนกระทั่งในปัจจุบันซึ่งได้มีการนำวิทยาการเข้ารหัสลับมาประยุกต์ใช้กับคอมพิวเตอร์เนื่องจากถือได้ว่าเป็นนวัตกรรมใหม่ที่มีความทันสมัยและสะดวกในการติดต่อสื่อสารกันเพราะมีความรวดเร็วและมีประสิทธิภาพสูงในการทำงาน ซึ่งทำให้สามารถติดต่อสื่อสารกันได้ทั่วโลกโดยอาศัยการสื่อสารผ่านทางเครือข่ายสาธารณะ ซึ่งทำให้ข้อมูลข่าวสารที่ต้องการติดต่อสื่อสารมีความปลอดภัย จึงได้นำหลักการทางวิทยาการเข้ารหัสลับนั้นเข้ามาใช้ที่เรียกว่า เทคโนโลยีรักษาความปลอดภัยข้อมูล ที่มีอยู่มากมายในปัจจุบัน

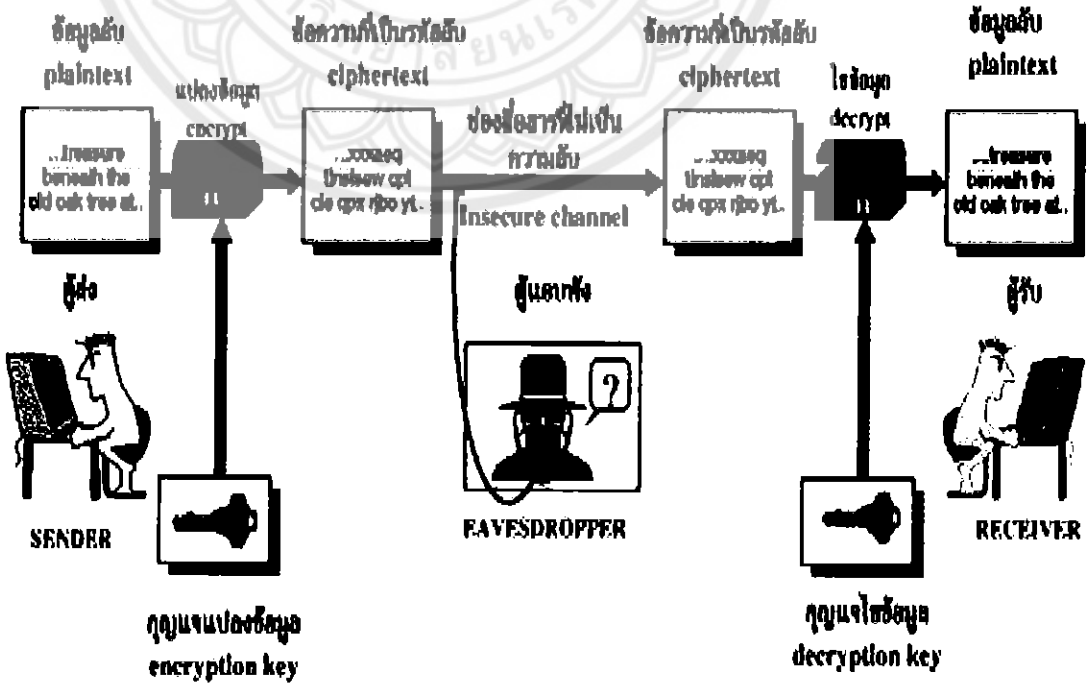
2.2 ความหมายของวิทยาการเข้ารหัสลับ (Definition of Cryptography)

คำว่า วิทยาการเข้ารหัสลับ ในภาษาอังกฤษได้ใช้คำว่า Cryptography ซึ่งเกิดจากรากศัพท์สองคำของภาษากรีก นั่นคือ Krypt หรือ Kryptós ซึ่งแปลว่า หลบ (Secret) หรือซ่อน (Hidden) และคำว่า Gráphein หรือ Graphia ซึ่งหมายถึง รูปวาดหรือข้อความ (Writing) เพราะฉะนั้นคำว่า Cryptography จึงมีความหมายว่า ศาสตร์ในการสร้างข้อมูลให้เป็นรหัสลับ เพื่อใช้ในการส่งข้อมูลอย่างเป็นความลับโดยผ่านทางช่องสื่อสารสาธารณะ

ศาสตร์วิชาที่ว่าด้วยรหัสลับ (Cryptology) ยังมีอีกความหมายหนึ่งของ คำว่า วิทยาการเข้ารหัสลับ ที่ใช้คำว่า Cryptography นั่นก็คือ Cryptanalysis ซึ่งหมายถึง ศาสตร์ในการไขข้อมูลลับ (Plaintext) จากข้อความที่เป็นรหัสลับ (Ciphertext) โดยที่ไม่มีกุญแจที่ใช้ในการถอดรหัสลับข้อมูล (Decryption Key) และคำว่า Cryptanalyst ก็คล้ายๆกับผู้ที่พยายามจะขโมยข้อมูลลับ โดยต้องการตีรหัสลับให้แตก (Attack, Break) แต่ที่จริงแล้ว Cryptanalyst ก็มักจะเป็น Cryptographer ด้วย เพราะว่าเมื่อ Cryptographer ออกแบบระบบรหัสลับ (Cryptosystem) ขึ้นมาแล้วก็จะต้องลองโจมตีระบบรหัสลับที่ได้ออกแบบมาเองด้วย เพื่อให้แน่ใจว่าระบบที่ได้ออกแบบมานั้นปลอดภัยจากพวกที่แอบดักโจรกรรม (Eavesdropper)

2.3 ระบบรหัสลับ (Cryptosystem)

ระบบวิทยาการเข้ารหัสลับ คือ ระบบจะทำการเข้ารหัส (Encrypt) ข้อมูลตั้งต้น (Plaintext) เอาไว้ในรูปแบบที่เป็นรหัสลับ (Ciphertext) โดยจะใช้กุญแจรหัสลับเพื่อใช้ในการเข้ารหัสลับข้อมูล (Encryption Key) ก่อนที่จะถูกส่งออกไป และจะรับได้เฉพาะผู้รับที่ต้องการจะส่งข้อมูลให้เท่านั้น การเปิดอ่านข้อมูลจะทำได้โดยจะต้องทำการถอดรหัสลับของข้อมูล (Decrypt) ที่ถูกเข้ารหัสไว้ก่อน การที่ผู้รับจะอ่านข้อมูลได้นั้นจะต้องมีกุญแจรหัสลับเพื่อใช้ในการถอดข้อมูล (Decryption Key) ที่ถูกต้องด้วย ทั้งนี้วิทยาการเข้ารหัสลับต้องการให้ข้อมูลที่ส่งไปไม่สามารถที่จะถูกแอบขโมยอ่านได้จากบุคคลอื่น หรือถูกดัดแปลงจากบุคคลอื่น ดังรูปที่ 2.1



รูปที่ 2.1 ระบบการทำงานของวิทยาการเข้ารหัสลับ [1]

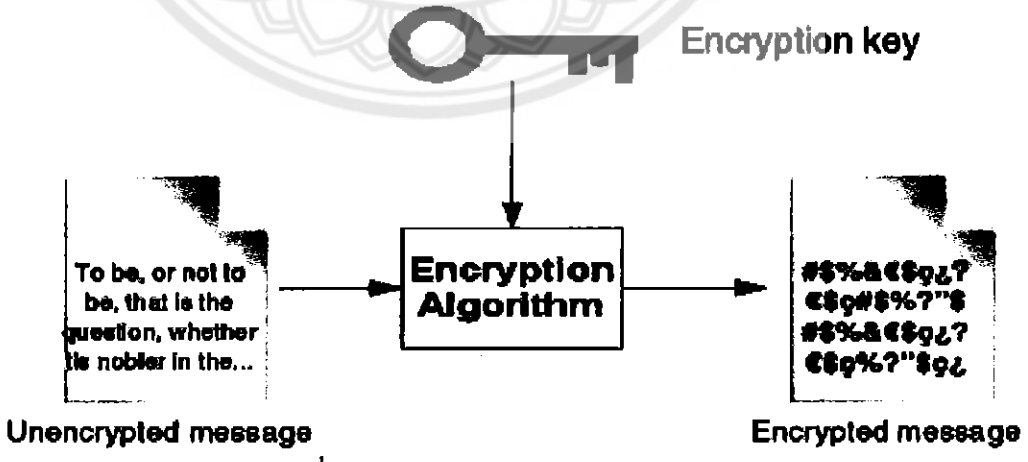
การเข้ารหัสลับข้อมูล (Encryption) หมายถึง วิธีการแปลงข้อมูลตั้งต้น (Plaintext) ที่อ่านเข้าใจได้ให้ไม่สามารถอ่านหรือแปลความหมายเข้าใจได้จากบุคคลทั่วๆไป โดยที่จะอาศัยกระบวนการทางคณิตศาสตร์ที่ซับซ้อนในการแปรรูปข้อมูล ซึ่งข้อมูลที่ไม่สามารถอ่านได้นี้เรียกว่า (Ciphertext)

การถอดรหัสลับข้อมูล (Decryption) หมายถึง วิธีการเปลี่ยนแปลงข้อมูลที่ไม่สามารถอ่านได้ (Ciphertext) ให้สามารถทำความเข้าใจได้ โดยข้อมูลที่ได้กลับคืนมาจากการถูกถอดรหัสลับนี้เรียกว่า (Origin Plaintext)

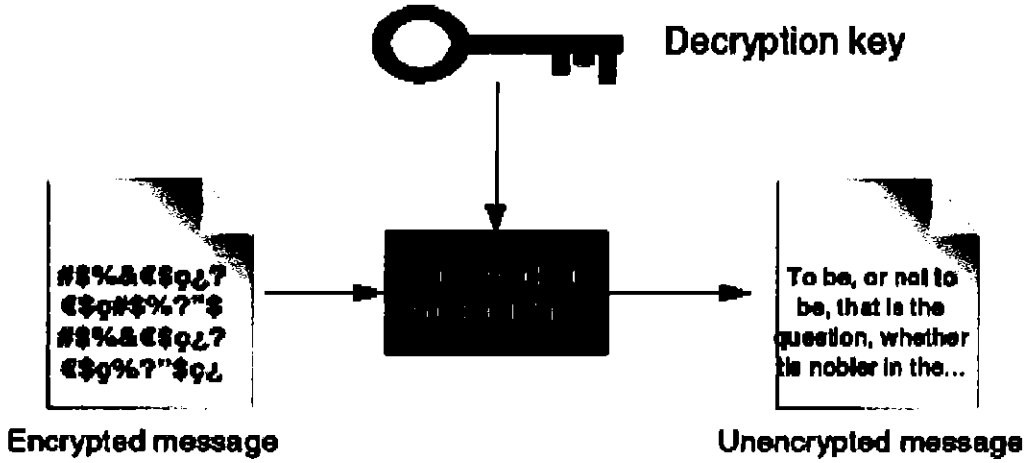
2.4 อัลกอริทึมและกุญแจรหัสลับ (Algorithm and Key)

ปัจจุบันระบบวิทยาการเข้ารหัสลับได้ใช้กระบวนการพื้นฐานทางคณิตศาสตร์ที่ซับซ้อนในการคำนวณเพื่อนำมาใช้แปรรูปข้อมูล โดยการแปรรูปข้อมูลตั้งต้น (Plaintext) ให้กลายมาเป็นรหัสลับ (Ciphertext) เรียกกระบวนการนี้ว่า การเข้ารหัสลับข้อมูล (Encryption) และเมื่อผู้รับได้รับรหัสลับแล้วจะต้องนำรหัสลับที่ได้มาแปรรูปข้อมูลกลับเพื่อจะเป็นข้อความตั้งต้น (Original Plaintext) ที่เหมือนกับข้อความเดิมอีกครั้งหนึ่ง ซึ่งเรียกกระบวนการนี้ว่า การถอดรหัสลับข้อมูล (Decryption)

ดังนั้นกระบวนการแปรรูปข้อมูลตั้งต้นให้เป็นรหัสลับโดยอาศัยพื้นฐานทางคณิตศาสตร์ในการดำเนินการคำนวณนี้จะเรียกว่า อัลกอริทึม (Algorithm) นั่นแสดงว่าอัลกอริทึม ซึ่งเปรียบได้กับว่าเป็นกุญแจสำคัญที่นำมาใช้เป็นเครื่องมือในการแปรรูปข้อมูลร่วมกับกุญแจรหัสลับ เพราะฉะนั้นกระบวนการเข้ารหัสลับข้อมูลจึงใช้กุญแจรหัสลับที่เรียกว่า กุญแจเข้ารหัสลับข้อมูล (Encryption Key) ดังรูปที่ 2.2 และถ้าเป็นกระบวนการถอดรหัสลับข้อมูลจะเรียกกุญแจรหัสลับที่ใช้ว่า กุญแจถอดรหัสลับข้อมูล (Decryption Key) ดังรูปที่ 2.3



รูปที่ 2.2 กระบวนการเข้ารหัสลับข้อมูล [2]



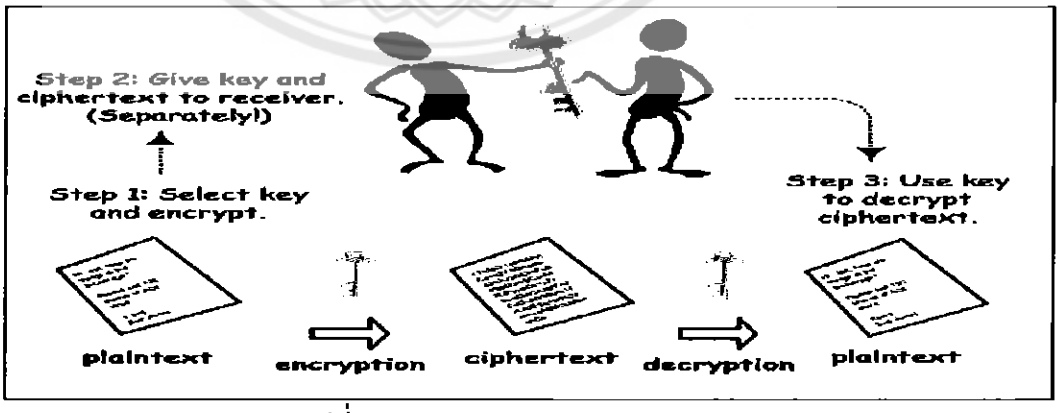
รูปที่ 2.3 กระบวนการถอดรหัสลับข้อมูล [2]

2.4.1 ชนิดของกุญแจรหัสลับ

ชนิดของกุญแจรหัสลับที่เกิดจากการกำหนดหรือสุ่มขึ้นมาเพื่อที่จะนำมาใช้เข้ารหัสลับและถอดรหัสลับข้อมูลนั้น ซึ่งนำมาใช้กับกระบวนการของอัลกอริทึมในการทำการแปรรูปข้อมูล โดยจะแบ่งออกเป็น 2 ประเภท ซึ่งจะแบ่งได้จากการใช้กุญแจรหัสลับในการเข้ารหัสลับและถอดรหัสลับข้อมูล ดังนี้

2.4.1.1 กุญแจรหัสลับแบบสมมาตร (Symmetric Key Cryptography หรือ Secret Key Cryptography)

วิธีนี้เป็นการเข้ารหัสลับและถอดรหัสลับโดยใช้กุญแจดอกเดียวกันเท่านั้น โดยจะต้องเป็นกุญแจที่รู้กันระหว่างผู้ส่งและผู้รับเท่านั้น กุญแจที่ใช้นี้เรียกว่า กุญแจลับ (Secret Key) หรือกุญแจส่วนตัว (Private Key) ดังรูปที่ 2.4



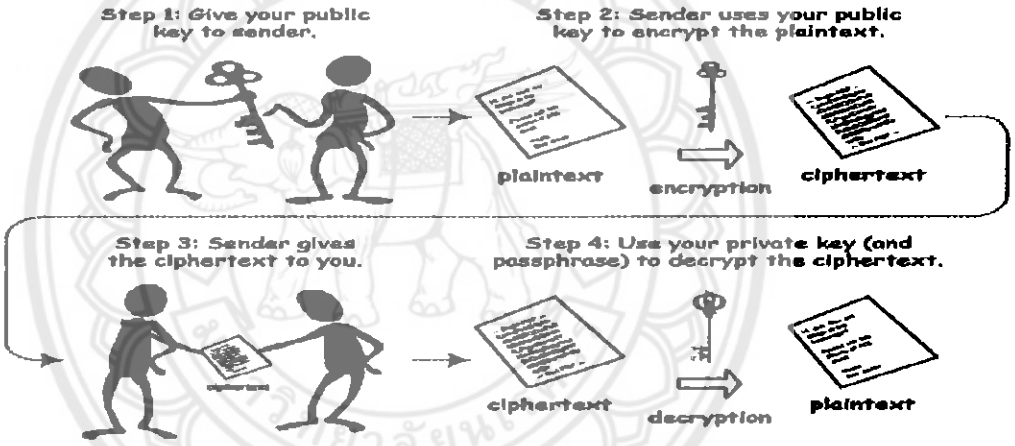
รูปที่ 2.4 กุญแจรหัสลับแบบสมมาตร [3]

2.4.1.2 กุญแจรหัสแบบอสมมาตร (Asymmetric Key Cryptography)

วิธีนี้เป็นการเข้ารหัสลับและถอดรหัสลับโดยใช้กุญแจคนละดอกกัน โดยกุญแจตัวหนึ่งจะใช้ในการเข้ารหัสลับและอีกตัวหนึ่งจะใช้ในการถอดรหัสลับดังรูปที่ 2.5 หรืออาจเรียกได้อีกอย่างหนึ่งว่า การเข้ารหัสลับและถอดรหัสลับด้วยกุญแจ 2 ดอก ที่มีลักษณะแตกต่างกัน ดังนี้

- กุญแจสาธารณะ (Public Keys) หมายถึง กุญแจที่สามรถเปิดเผยได้ โดยจะส่งมอบให้กับผู้ที่ต้องการมาติดต่อหรือแม้แต่กระทั่งวางไว้บนเว็บไซต์ เพื่อที่จะให้ผู้ซึ่งต้องการติดต่อสามารถดาวน์โหลดไปเก็บไว้ใช้งานได้ ทั้งนี้กุญแจสาธารณะจะมีได้หลายดอกต่อกุญแจส่วนตัวหนึ่งดอก

- กุญแจส่วนตัว (Private Keys) เป็นกุญแจที่ห้ามเปิดเผย ควรจะรู้เฉพาะผู้ที่เป็นเจ้าของกุญแจเท่านั้น โดยจะมีเพียงดอกเดียวซึ่งมีไว้เพื่อใช้ในการติดต่อกับผู้ที่ต้องการติดต่อด้วย ซึ่งผู้ที่มาติดต่อด้วยจะต้องมีกุญแจสาธารณะที่เป็นคู่กันกับกุญแจส่วนตัวของผู้เป็นเจ้าของอยู่ด้วย



รูปที่ 2.5 กุญแจรหัสลับแบบอสมมาตร [4]

2.4.2 ข้อดีและข้อเสียของกุญแจแบบสมมาตรและอสมมาตร

2.4.2.1 ข้อดีและข้อเสียของกุญแจแบบสมมาตร

ข้อดีของกุญแจแบบสมมาตร

- การเข้ารหัสลับและการถอดรหัสลับใช้เวลาน้อยมาก เพราะว่าอัลกอริทึมที่ใช้ไม่ซับซ้อนมาก

- ขนาดของข้อมูลที่เปลี่ยนแปลงไปหลังจากทำการเข้ารหัสลับแล้วไม่มากนัก ดังนั้นขนาดของข้อมูลที่ได้จึงไม่ใหญ่มากไปกว่าเดิม

ข้อเสียของกุญแจแบบสมมาตร

- การจัดการบริหารกุญแจจะยุ่งยาก นั่นคือหากต้องการติดต่อกับใครก็ตามต้องจำให้ได้ว่าแลกเปลี่ยนกุญแจดอกไหน เนื่องจากกุญแจที่ใช้เข้ารหัสลับและถอดรหัสลับเป็นกุญแจแบบเดียวกัน

ถ้าหากต้องการติดต่อกันจำนวน n คน จำนวนกุญแจลับทั้งหมดคือ 2n หรืออีกความหมายหนึ่งก็คือ จำนวนของการรักษาความลับของกุญแจลับเพียงดอกเดียวจะต้องรักษาความลับกันทั้ง 2 ฝ่าย

2.4.2.2 ข้อดีและข้อเสียของกุญแจแบบอสมมาตร

ข้อดีของกุญแจแบบอสมมาตร

- การจัดการบริหารกุญแจทำได้ง่าย เนื่องมาจากผู้ส่งไม่ต้องจำว่าส่งกุญแจสาธารณะไปให้ใครบ้าง โดยจำนวนกุญแจสาธารณะที่ส่งไปนั้นมีจำนวนมากเมื่อเทียบกับจำนวนของกุญแจส่วนตัว นั่นคือกุญแจส่วนตัวหนึ่งดอกสามารถมีคู่ที่เป็นกุญแจสาธารณะได้หลายดอก เพราะฉะนั้นเพียงแค่ใช้กุญแจส่วนตัวเพียงดอกเดียวก็สามารถติดต่อกับใครก็ได้หลายคนหรืออาจจะจำเป็นกลุ่มว่าส่งกุญแจสาธารณะไปให้กลุ่มใดบ้าง

- การจัดการบริหารกุญแจทำได้ง่าย เนื่องจากการเข้ารหัสลับโดยวิธีนี้ใช้แค่กุญแจส่วนตัวของตัวเองเพียงดอกเดียว ซึ่งสามารถที่จะติดต่อกับใครก็ได้ที่มีกุญแจสาธารณะที่เป็นคู่กันกับกุญแจส่วนตัว ดังนั้นหากมีการแลกเปลี่ยนกุญแจ n กลุ่ม จำนวนกุญแจลับทั้งหมดก็คือ n กลุ่ม เพราะจำนวนกุญแจที่ต้องรักษาก็คือ n ดอก หรือรักษาแค่เพียงกุญแจส่วนตัวเท่านั้น ส่วนกุญแจสาธารณะให้ผู้ที่ต้องการติดต่อเป็นผู้ดูแลรักษาเอง

ข้อเสียของกุญแจแบบอสมมาตร

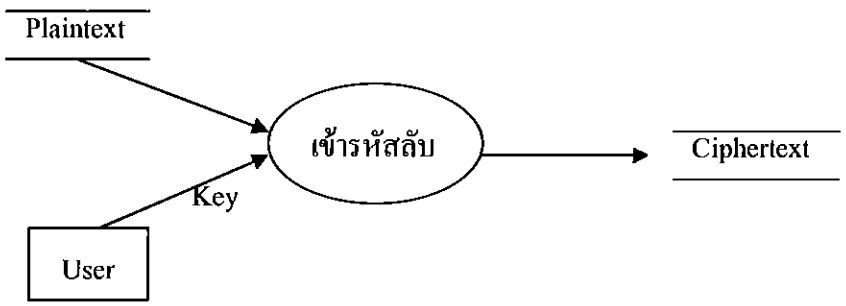
- ใช้เวลานานมากเพราะว่าอัลกอริทึมที่ใช้ซับซ้อนมากกว่าแบบสมมาตร
- ขนาดของข้อมูลหลังจากที่เข้ารหัสลับแล้วมีการเปลี่ยนแปลงไปมากจึงเป็นปัญหาในการรับส่งข้อมูล เพราะถ้าหากข้อมูลยิ่งมากการส่งข้อมูลก็จะยิ่งใช้เวลานานด้วย

บทที่ 3
วิธีการดำเนินงาน

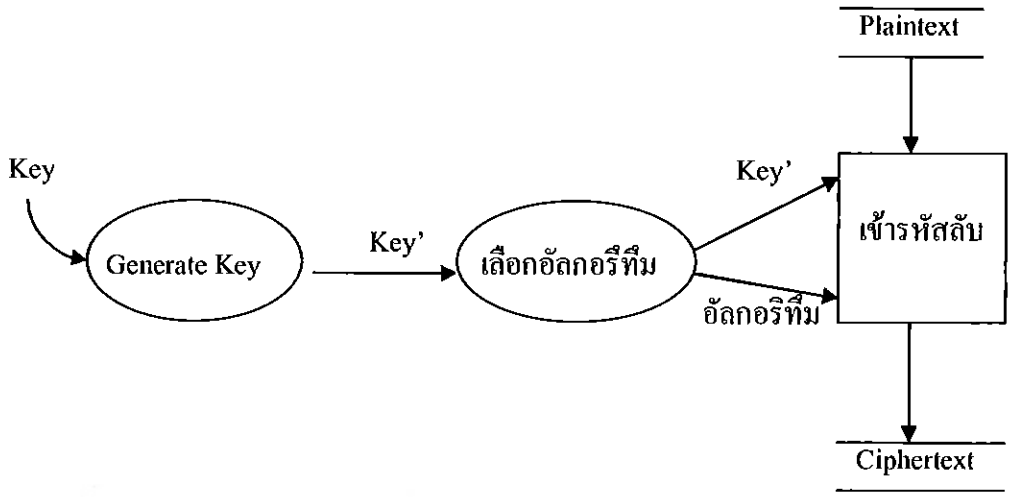
ในบทนี้จะกล่าวถึงขั้นตอนการทำงาน เพื่อให้ได้มาซึ่งอัลกอริทึมในการเข้ารหัสลับ โดยมีการ Generate Key สำหรับใช้ในการเข้ารหัสลับ มีการเลือกอัลกอริทึมเพื่อใช้ในการเข้ารหัสลับ และมีการเพิ่ม Noise เข้าไปในตัว Ciphertext โดยที่สามารถถอดรหัสลับกลับมาเป็นข้อความตั้งต้นได้เหมือนเดิม ซึ่งอัลกอริทึมนี้ทำให้ Cracker ต้องเพิ่มระยะเวลาในการคำนวณเพื่อค้นหา Key ให้พบ โดยมีการศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องเพื่อนำมาออกแบบอัลกอริทึม

3.1 การออกแบบอัลกอริทึมการเข้ารหัสลับและถอดรหัสลับ

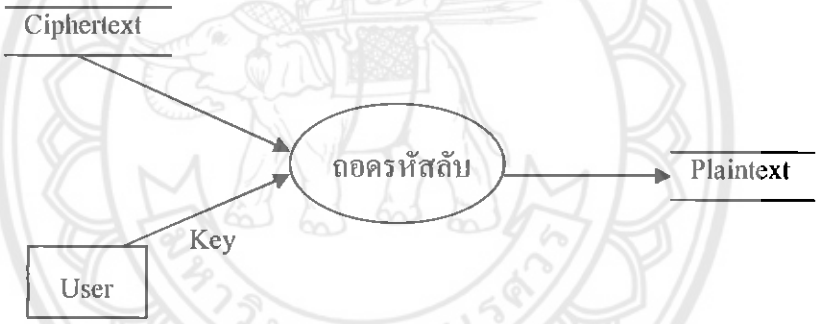
ผู้ดำเนินโครงงานได้เลือกแนวความคิดที่จะใช้การเพิ่ม Noise มาช่วยในการเข้ารหัสลับ ซึ่งจากการที่ได้ศึกษาและทดลองในรูปแบบต่างๆ จึงได้เลือกวิธีการการเข้ารหัสลับข้อความต้นฉบับเป็นไซเฟอร์แบบบล็อก (Block Cipher) โดยที่ไซเฟอร์แบบบล็อก หมายถึง การแบ่งข้อความต้นฉบับออกเป็นส่วนๆ โดยแต่ละส่วนเรียกว่าบล็อก จากนั้นทำการเข้ารหัสลับข้อความต้นฉบับที่ละบล็อก ซึ่งขนาดของบล็อกที่ใช้ในการเข้ารหัสลับนั้นจะขึ้นอยู่กับอัลกอริทึมที่ใช้ในการเข้ารหัสลับ ในส่วนของชนิดกุญแจรหัสลับจะใช้กุญแจรหัสลับแบบสมมาตร (Symmetric Key Cryptography หรือ Secret Key Cryptography) โดยที่จำนวนของกุญแจลับที่ใช้ในการเข้ารหัสลับนั้นขึ้นอยู่กับอัลกอริทึมที่นำมาใช้ ซึ่งในการเข้ารหัสลับจะมีการเข้ารหัสลับข้อความตั้งต้นในอัลกอริทึมที่แตกต่างกันแล้วแต่จะกำหนดว่าใช้อัลกอริทึมใดบ้าง โดยมีแผนภาพในการเข้ารหัสลับและถอดรหัสลับดังรูปที่ 3.5 และรูปที่ 3.6



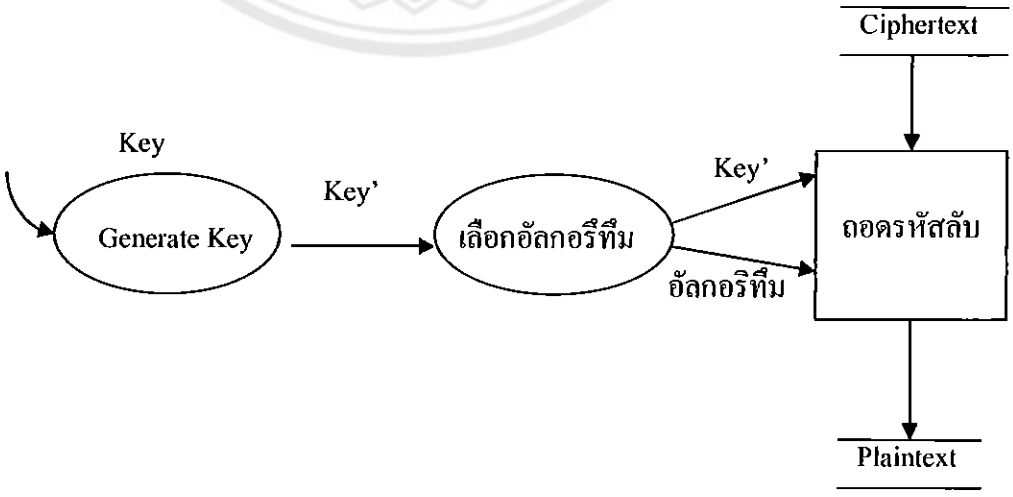
รูปที่ 3.1 แผนภาพ Data flow ของการเข้ารหัสลับ



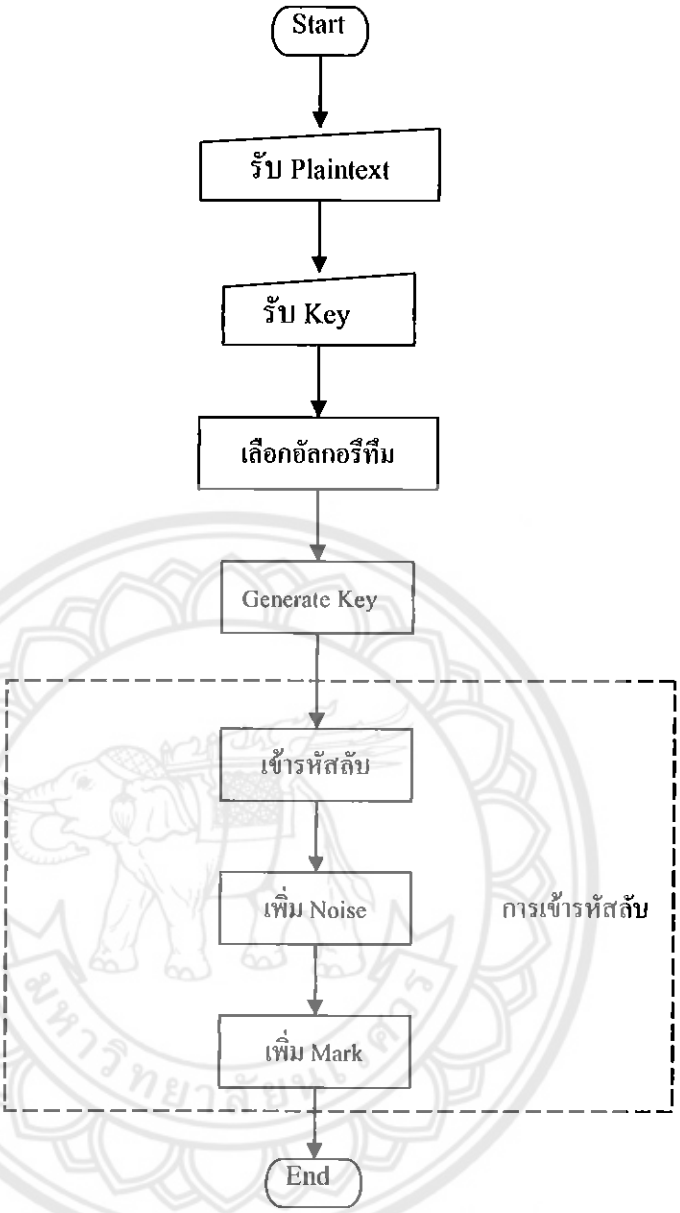
รูปที่ 3.2 แผนภาพ Data flow แสดงขั้นตอนการเข้ารหัสลับ



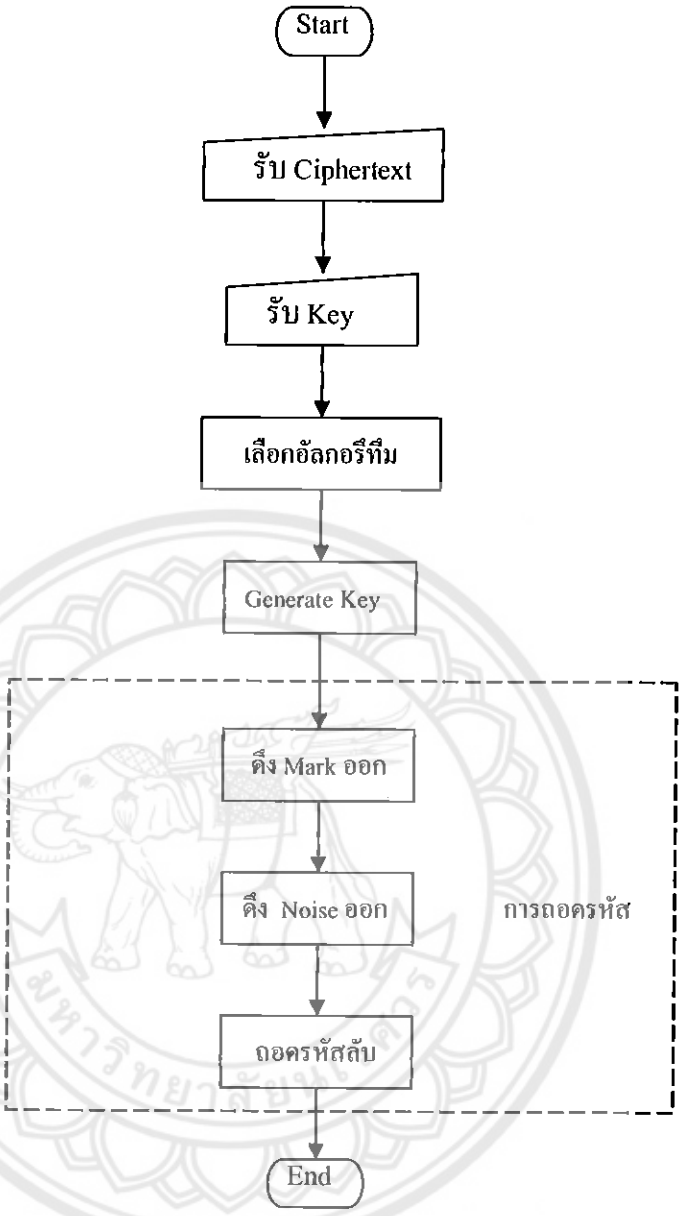
รูปที่ 3.3 แผนภาพ Data flow ของการถอดรหัสลับ



รูปที่ 3.4 แผนภาพ Data flow แสดงขั้นตอนการถอดรหัสลับลับ



รูปที่ 3.5 แผนภาพการเข้ารหัสลับ



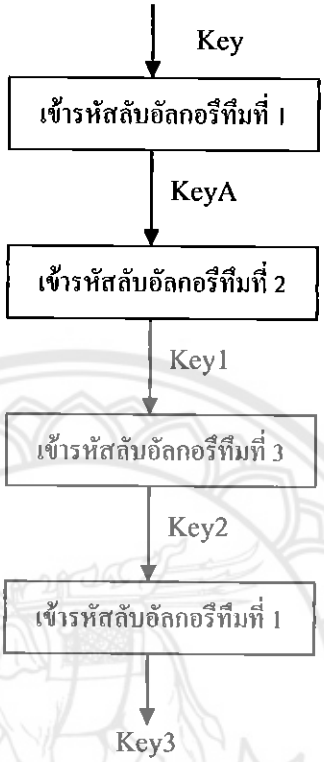
รูปที่ 3.6 แผนภาพการถอดรหัสลับ

3.2 ขั้นตอนการเลือกอัลกอริทึม

ในการเลือกอัลกอริทึมที่ใช้ในการเข้ารหัสลับและถอดรหัสลับนั้นสามารถเลือกได้ว่าจะใช้อัลกอริทึมใด จากนั้นระบุไปว่าอัลกอริทึมที่ 1 คืออัลกอริทึมใด อัลกอริทึมที่ 2 คืออัลกอริทึมใด และอัลกอริทึมที่ 3 คืออัลกอริทึมใด เช่น อัลกอริทึมที่ 1 คือ AES อัลกอริทึมที่ 2 คือ XOR และอัลกอริทึมที่ 2 คือ NXOR เป็นต้น

3.3 ขั้นตอนการ Generate Key

ในขั้นตอนการ Generate Key สามรถเขียนแผนภาพได้ดังรูปที่ 3.7



รูปที่ 3.7 แผนภาพการ Generate Key

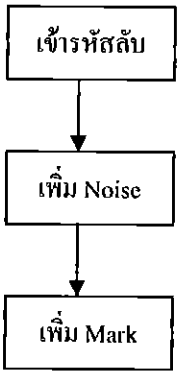
จากรูปที่ 3.2 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1. นำ Key มาเข้ารหัสลับอัลกอริทึมที่ 1 ซึ่ง Ciphertext ที่ได้จะกำหนดให้เป็น KeyA
2. นำ KeyA มาเข้ารหัสลับอัลกอริทึมที่ 2 ซึ่ง Ciphertext ที่ได้จะกำหนดให้เป็น Key1
3. นำ Key1 มาเข้ารหัสลับอัลกอริทึมที่ 3 ซึ่ง Ciphertext ที่ได้จะกำหนดให้เป็น Key2
4. นำ Key2 มาเข้ารหัสลับอัลกอริทึมที่ 1 ซึ่ง Ciphertext ที่ได้จะกำหนดให้เป็น Key3

เมื่อได้ Output มาแล้วจะนำไปใช้ต่อดังนี้ Key1 ใช้เป็น Key ของอัลกอริทึมที่ 1, Key2 ใช้เป็น Key ของอัลกอริทึมที่ 2 และ Key3 ใช้เป็น Key ของอัลกอริทึมที่ 3

3.4 ขั้นตอนวิธีการเข้ารหัสลับ

หลังจากที่ได้เลือกอัลกอริทึมและ Generate Key ซึ่งใช้ในการเข้ารหัสลับและถอดรหัสลับแล้ว การออกแบบวิธีการเข้ารหัสลับมีขั้นตอนดังรูปที่ 3.8



รูปที่ 3.8 แผนภาพวิธีการเข้ารหัสลับ

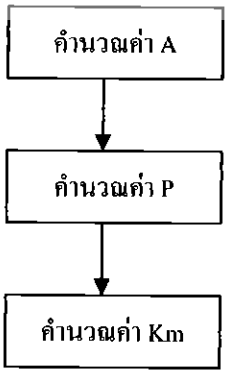
3.4.1 การเข้ารหัสลับ

ในขั้นตอนเข้ารหัสลับสามารถเขียนแผนภาพได้ดังรูปที่ 3.9

รูปที่ 3.9 แผนภาพการเข้ารหัสลับ

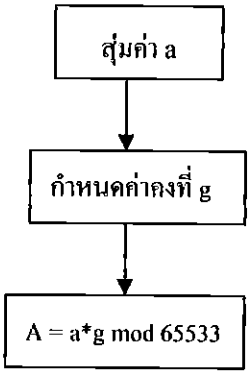
3.4.1.1 การเลือกอัลกอริทึมในการเข้ารหัสลับ

วิธีการเลือกอัลกอริทึมในการเข้ารหัสลับมีลำดับการคำนวณหาได้ดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพลำดับการคำนวณเพื่อเลือกอัลกอริทึมในการเข้ารหัสลับ

- วิธีการคำนวณหาค่า A มีลำดับการทำงานดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพขั้นตอนการคำนวณหาค่า A

ตัวอย่าง การคำนวณหาค่า A

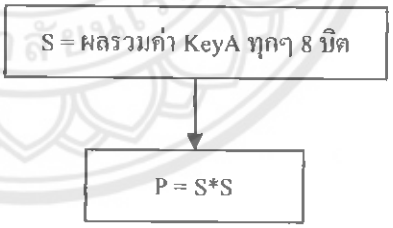
กำหนดให้ a คือค่าตัวเลขที่ได้จากการสุ่ม a = 171337

กำหนดให้ g คือค่าที่ได้กำหนดขึ้นมา g = 5713

A = 171337*5713 mod 65533

A = 47393

- วิธีการคำนวณหาค่า P มีลำดับการทำงานดังรูปที่ 3.12



รูปที่ 3.12 แผนภาพขั้นตอนการคำนวณหาค่า P

ตัวอย่าง การคำนวณหาค่า P

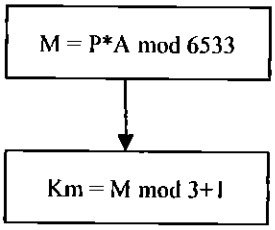
กำหนดให้จำนวน KeyA มีทั้งหมด 128 บิต ซึ่งมีค่าเป็นเลขฐาน 16 ดังนี้ a111f2530495a6b7f889fa1b0cfdce52 แล้วนำค่าทุกๆ 8 บิตมาบวกกันได้เป็น

S = 2220

P = 2220*2220

P = 4928400

- วิธีการคำนวณหาค่า Km มีลำดับการทำงานดังรูปที่ 3.13



รูปที่ 3.13 แผนภาพขั้นตอนการคำนวณหาค่า Km

ตัวอย่าง การคำนวณหาค่า Km

M = P*A mod 65533

M = 4928400*47393 mod 65533

M = 56661

Km = M mod 3 +1

Km = 1

โดยค่า Km ที่ได้นี้เป็นตัวบ่งชี้ว่าได้เลือกอัลกอริทึมที่ Km ในการเข้ารหัสลับ ซึ่งสามารถแจกแจงได้ดังนี้

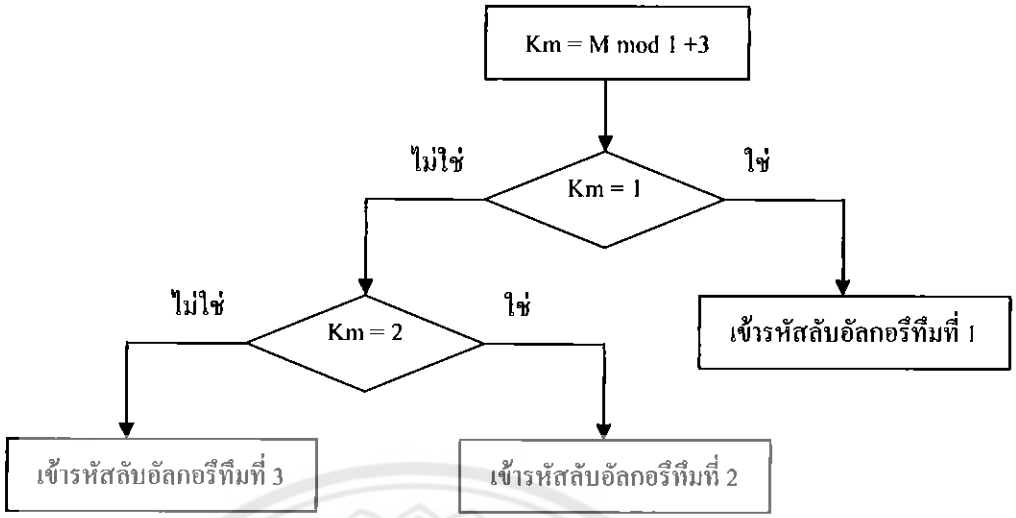
ถ้า Km = 1 ได้เลือกอัลกอริทึมที่ 1 ในการเข้ารหัสลับ

ถ้า Km = 2 ได้เลือกอัลกอริทึมที่ 2 ในการเข้ารหัสลับ

ถ้า Km = 3 ได้เลือกอัลกอริทึมที่ 3 ในการเข้ารหัสลับ

3.4.1.2 การเข้ารหัสลับ

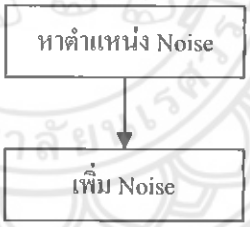
นำ Plaintext ไปเข้ารหัสลับอัลกอริทึมที่ได้ถูกเลือกไว้ซึ่งจะได้ Ciphertext ออกมา



รูปที่ 3.14 แผนภาพสรุปการเข้ารหัสลับ

3.4.2 ขั้นตอนการเพิ่ม Noise

หลังจากที่ได้นำ Plaintext ไปเข้ารหัสลับแล้วจะได้ Ciphertext ออกมา โดยมีการเพิ่ม Noise เข้าไปใน Ciphertext ในขั้นตอนการเพิ่ม Noise สามารถเขียนแผนภาพได้ดังรูปที่ 3.15



รูปที่ 3.15 แผนภาพขั้นตอนการเพิ่ม Noise

3.4.2.1 การหาตำแหน่ง Noise

วิธีการหาตำแหน่ง Noise มีการคำนวณหาได้จาก

$$N = P*A \bmod B + 1$$

กำหนดให้ N = ตำแหน่ง Noise

B = จำนวนไบต์ของ Ciphertext ที่ยังไม่ได้เพิ่ม Noise

โดยค่า N ที่ได้นี้เป็นตัวบ่งชี้ว่าได้เพิ่ม Noise ไปไบต์ที่ N ของ Ciphertext นับจากซ้ายไปขวา

ตัวอย่าง การคำนวณหาค่า N

N = P*A mod 16 +1

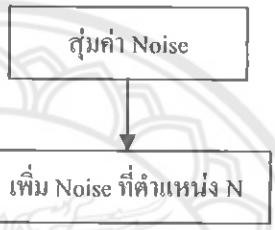
N = 4928400*47393 mod 16 +1

N = 1

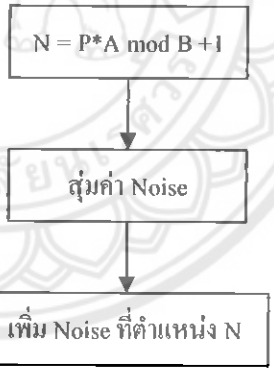
จากตัวอย่าง N = 1 แสดงว่าต้องเพิ่ม Noise ไปที่ไบต์ที่ 1

3.4.2.2 การเพิ่ม Noise

หลังจากคำนวณหาตำแหน่งของ Noise ได้แล้ว ต่อจากนั้นทำการสุ่มค่า Noise ไม่เกิน 16 บิต ซึ่งก็ค่า 65535 ในเลขฐานสิบ และเพิ่ม Noise เข้าไปใน Ciphertext ตำแหน่งที่ N ขั้นตอนการเพิ่ม Noise แสดงดังรูปที่ 3.16



รูปที่ 3.16 แผนภาพการเพิ่ม Noise



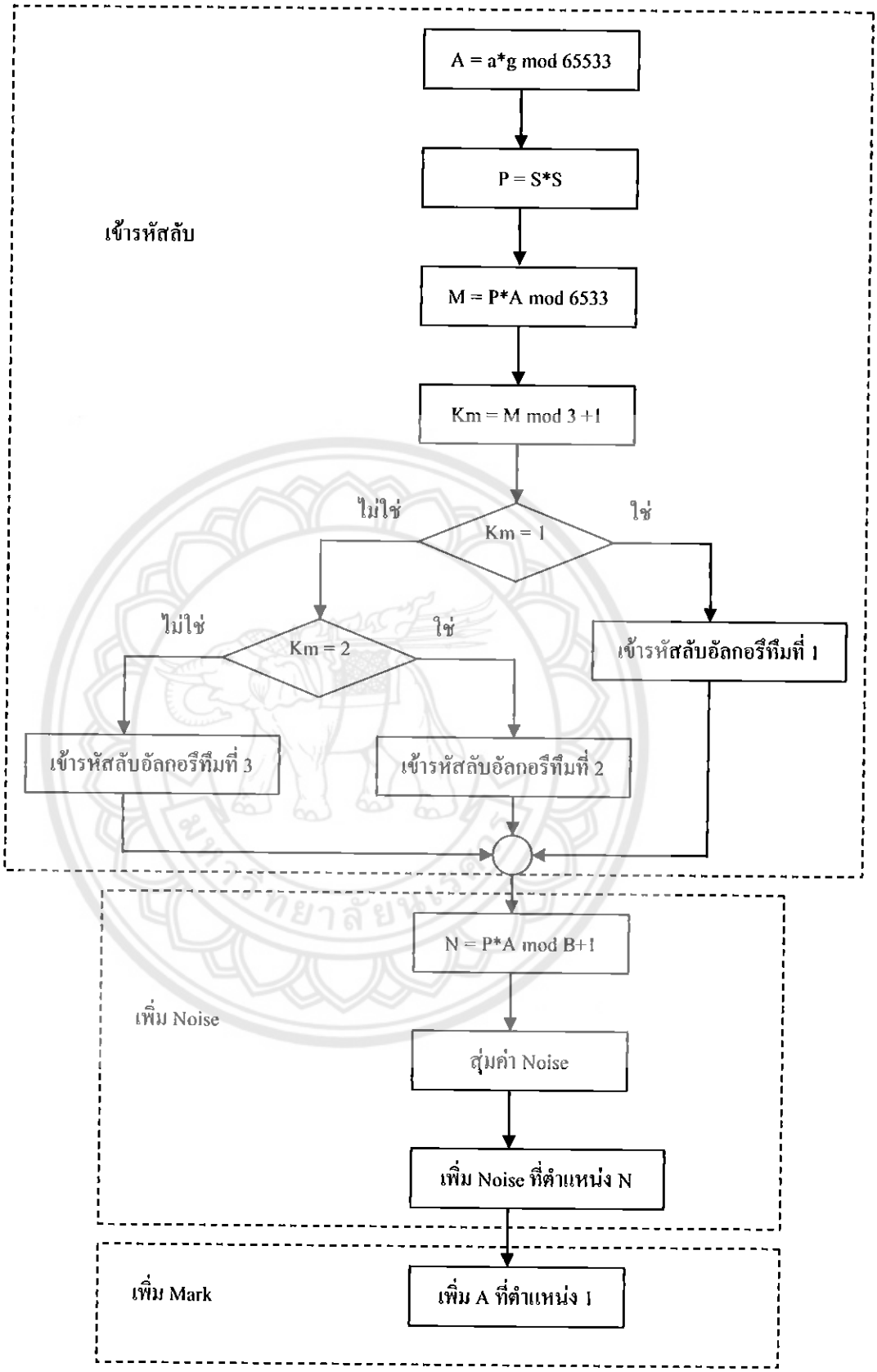
รูปที่ 3.17 แผนภาพสรุปขั้นตอนการเพิ่ม Noise

3.4.3 ขั้นตอนการเพิ่ม Mark

เมื่อเพิ่ม Noise เสร็จแล้ว ต่อจากนั้นทำการเพิ่ม Mark ไว้หน้าสุดของ Ciphertext ที่เพิ่ม Noise แล้ว โดยที่ Mark นั้นคือค่า A

สรุป

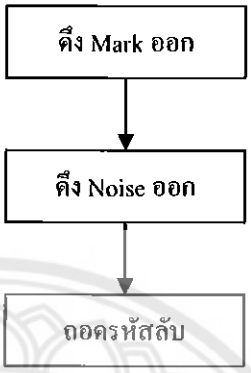
สุดท้ายแล้ว Ciphertext ที่ถูกส่งไปจะมี Noise และ A อยู่ในนั้น



รูปที่ 3.18 แผนภาพสรุปวิธีการเข้ารหัสลับ

3.5 ขั้นตอนวิธีการถอดรหัสลับ

เมื่อได้รับ Ciphertext ซึ่งมาจากการเข้ารหัสลับแล้ว จะมีขั้นตอนการถอดรหัสลับเพื่อให้ได้มาซึ่งข้อความตั้งต้นดังรูปที่ 3.19



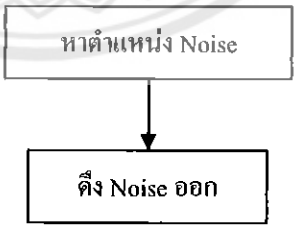
รูปที่ 3.19 แผนภาพวิธีการถอดรหัสลับ

3.5.1 ขั้นตอนการดึง Mark ออก

เมื่อได้รับ Ciphertext มา อันดับแรกที่ต้องทำคือ ดึง Mark ออกจาก Ciphertext ก่อน โดยที่ Mark คือ 16 บิตแรกของ Ciphertext และค่า Mark ก็คือค่า A

3.5.2 ขั้นตอนการดึง Noise ออก

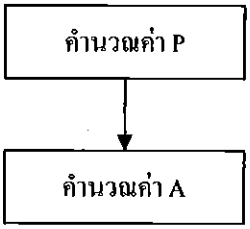
หลังจากดึง Mark ออกแล้ว ต่อมาเป็นการดึง Noise ออกจาก Ciphertext โดยมีขั้นตอนดังรูปที่ 3.20



รูปที่ 3.20 แผนภาพขั้นตอนการดึง Noise ออก

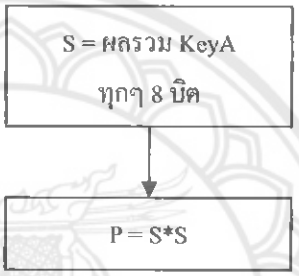
3.5.2.1 การหาตำแหน่ง Noise

วิธีการหาตำแหน่งของ Noise ที่แทรกอยู่ใน Ciphertext มีลำดับขั้นตอนการคำนวณหาได้ดังรูปที่ 3.21



รูปที่ 3.21 แผนภาพลำดับการคำนวณเพื่อหาตำแหน่ง Noise

- วิธีการคำนวณหาค่า P มีลำดับการทำงานดังรูปที่ 3.22



รูปที่ 3.22 แผนภาพขั้นตอนการคำนวณหาค่า P

ตัวอย่าง การคำนวณหาค่า P

กำหนดให้จำนวน KeyA มีทั้งหมด 128 บิต ซึ่งมีค่าเป็นเลขฐาน 16 ดังนี้ a111f2530495a6b7f889fa1b0cfdce52 แล้วนำค่าทุกๆ 8 บิตมาบวกกันได้เป็น

S = 2220

P = 2220*2220

P = 4928400

- วิธีการหาค่า N มีการคำนวณหาได้จาก

$$N = P*A \bmod B + 1$$

กำหนดให้ N = ตำแหน่ง Noise

B = จำนวนไบต์ของ Ciphertext ที่ยังไม่ได้เพิ่ม Noise

ตัวอย่าง การคำนวณหาค่า N

N = P*A mod 16 +1

N = 4928400*47393 mod 16 +1

N = 1

จากตัวอย่าง N = 1 แสดงว่าต้องดึง Noise ไบต์ที่ 1 และ 2 ออกจาก Ciphertext

3.5.2.2 การดึง Noise ออก

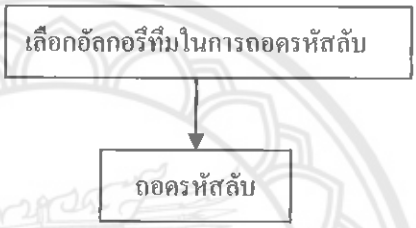
เมื่อคำนวณหาตำแหน่งที่ Noise แทรกอยู่ได้แล้ว ซึ่งก็คือค่า N โดยค่า N ที่ได้เป็นตัวบ่งชี้ว่าต้องดึง Noise ไบต์ที่ N และ N+1 ของ Ciphertext นับจากซ้ายไปขวาออก ที่ต้องดึงออก 2 ไบต์ เพราะว่าได้มีการเพิ่ม Noise ไป 16 บิต

ตัวอย่าง การดึง Noise ออก

สมมติให้ N = 1 แสดงว่าต้องดึง Noise ไบต์ที่ 1 และ 2 ออกจาก Ciphertext

3.5.3 ขั้นตอนการถอดรหัสลับ

ในขั้นตอนถอดรหัสลับสามารถเขียนแผนภาพได้ดังรูปที่ 3.23



รูปที่ 3.23 แผนภาพการถอดรหัสลับ

3.5.3.1 เลือกอัลกอริทึมในการถอดรหัสลับ

วิธีการเลือกอัลกอริทึมในการเข้ารหัสลับมีลำดับการคำนวณหาได้ดังรูปที่ 3.24

รูปที่ 3.24 แผนภาพลำดับการคำนวณเพื่อเลือกอัลกอริทึมในการถอดรหัสลับ

- วิธีการหาค่า M มีการคำนวณหาได้จาก

$$M = P*A \bmod 65533$$

ตัวอย่าง การคำนวณหาค่า M

$$M = P*A \bmod 65533$$

$$M = 4928400*47393 \bmod 65533$$

$$M = 56661$$

- วิธีการหาค่า Km มีการคำนวณหาได้จาก

$$Km = M \bmod 3 + 1$$

ตัวอย่าง การคำนวณหาค่า Km

$$Km = M \bmod 3 + 1$$

$$Km = 56661 \bmod 3 + 1$$

$$Km = 1$$

โดยค่า Km ที่ได้นี้เป็นตัวบ่งชี้ว่าได้เลือกอัลกอริทึมที่ Km ในการถอดรหัสลับ ซึ่งสามารถแจกแจงได้ดังนี้

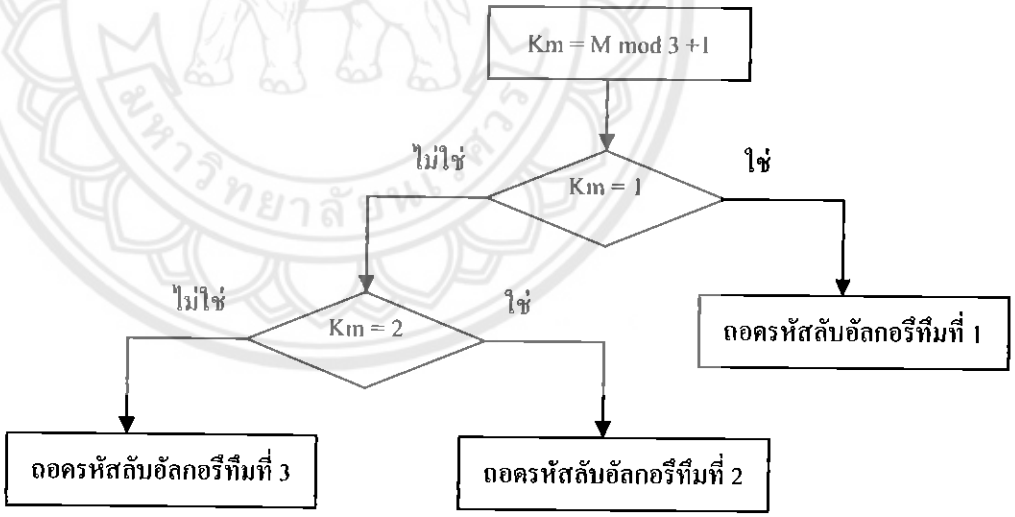
ถ้า Km = 1 ได้เลือกอัลกอริทึมที่ 1 ในการถอดรหัสลับ

ถ้า Km = 2 ได้เลือกอัลกอริทึมที่ 2 ในการถอดรหัสลับ

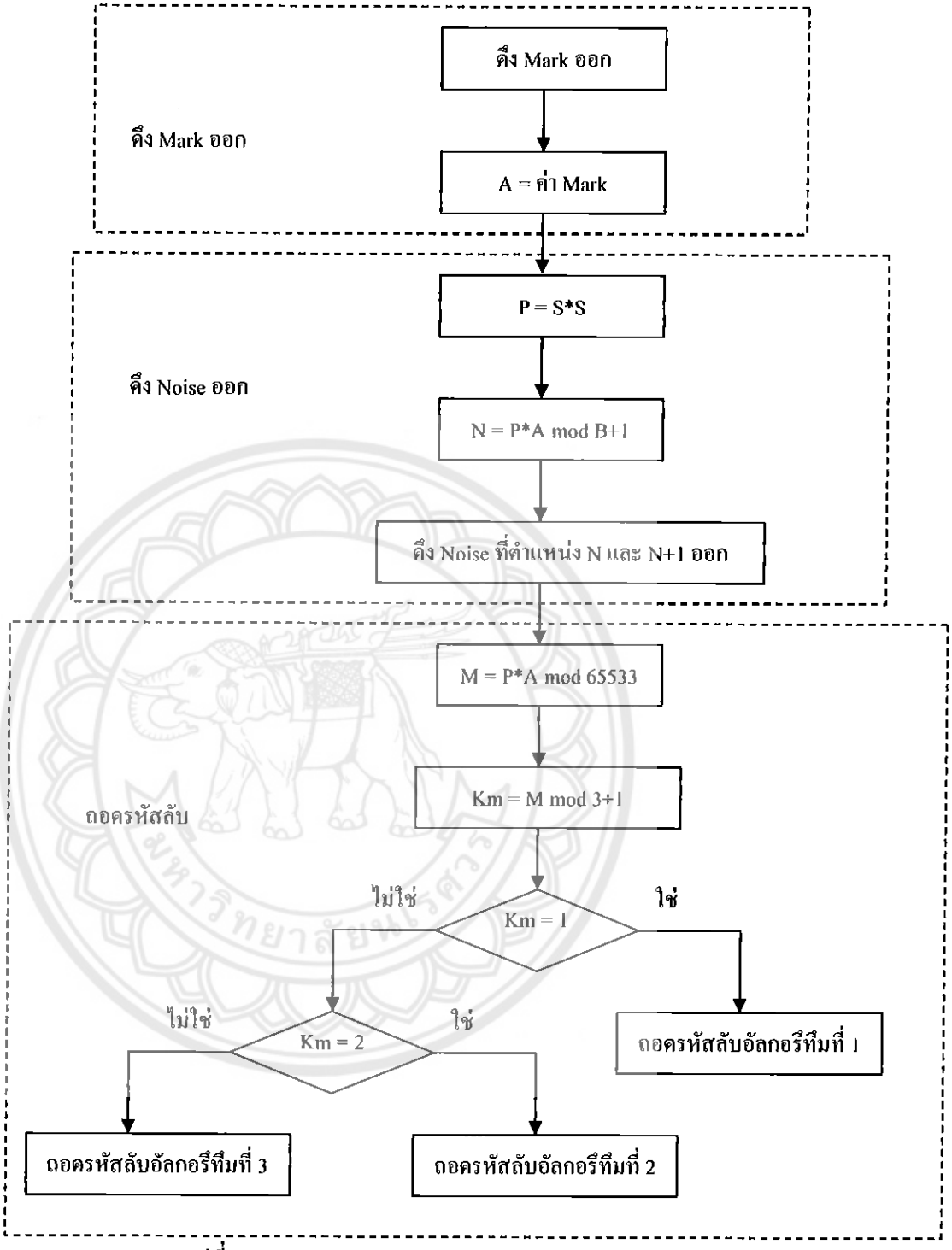
ถ้า Km = 3 ได้เลือกอัลกอริทึมที่ 3 ในการถอดรหัสลับ

3.5.3.2 การถอดรหัสลับ

นำ Ciphertext ที่ได้รับการดึง Mark และ Noise ออกแล้ว ไปถอดรหัสลับกับอัลกอริทึมที่ได้ถูกเลือกไว้ซึ่งจะได้ Plaintext ออกมา



รูปที่ 3.25 แผนภาพสรุปการถอดรหัสลับ



รูปที่ 3.26 แผนภาพสรุปวิธีการถอดรหัสลับ

3.6 ตัวอย่างการเข้ารหัสลับและถอดรหัสลับ

ในการแสดงตัวอย่างนี้ จะมีเป้าหมายเพื่อแสดงให้เห็นถึงการเข้ารหัสลับโดยใช้การเลือกอัลกอริทึมในการเข้ารหัสลับจาก 3 อัลกอริทึม จากนั้นมีการเพิ่ม Noise ไปที่ Block ของ Ciphertext แล้วสามารถที่จะถอดรหัสลับกลับมาเป็นข้อความตั้งต้นได้อย่างถูกต้อง โดยจะยกตัวอย่างของการเข้ารหัสลับ โดยมีขนาดของ Block ในการเข้ารหัสลับเป็น 128 บิต และขนาดของ Key ที่ใช้เป็น 128 บิต เพื่อง่ายต่อการเขียนจะแสดงเป็นเลขฐาน 16 โดยที่อัลกอริทึมที่ 1 คือ AES อัลกอริทึมที่ 2 คือ XOR และอัลกอริทึมที่ 3 คือ NXOR โดยมีตัวอย่างการเข้ารหัสลับและถอดรหัสลับดังนี้

Key คือ 31 32 33 34 35 36 37 38 39 31 32 33 34 35 36 37

Plaintext คือ ComputerEngineer

เปลี่ยน Plaintext จากตัวอักษรเป็นเลขฐาน 16 โดยใช้ตารางที่ 3.1 ได้เป็นดังนี้

Plaintext คือ 43 6f 6d 70 75 74 65 72 45 6e 67 69 6e 65 65 72

3.6.1 Generate key

นำ Key มา Generate ได้ดังนี้

KeyA = fb 15 a8 cc b9 84 b9 e4 5f 67 e1 de 38 e3 d7 e9

Key1 = ca 27 9b f8 8c b2 8e dc 66 56 d3 ed 0c d6 e1 de

Key2 = 04 6a 57 33 46 7b 46 1b 20 18 1e 21 47 1c 28 16

Key3 = bf 50 5f 7c 9e 1b 9f 13 2c de 15 30 81 22 23 45

โดย Key1 ใช้เป็น Key ในการเข้ารหัสลับและถอดรหัสลับอัลกอริทึมที่ 1

Key2 ใช้เป็น Key ในการเข้ารหัสลับและถอดรหัสลับอัลกอริทึมที่ 2

Key3 ใช้เป็น Key ในการเข้ารหัสลับและถอดรหัสลับอัลกอริทึมที่ 3

3.6.2 การเข้ารหัสลับ

1. บวกค่า KeyA ทุกๆ 8 บิต โดยจะมีค่าเท่ากับ 2750 จากนั้นนำมายกกำลังสอง

$$P = 2750*2750$$

$$P = 7562500$$

2. สุ่มค่า a ขึ้นมา โดยที่ค่า a = 38021

3. นำค่า a มาหาค่า A โดยจะกำหนดค่า g = 5713

$$A = 38021*5713 \bmod 65333$$

$$A = 47081 \text{ ฐาน } 10$$

$$A = b7e9 \text{ ฐาน } 16$$

4. นำค่า A และ P ที่ได้มาคำนวณเพื่อเลือกอัลกอริทึมที่จะใช้ในการเข้ารหัสลับ นั้นก็คือหาค่า Km

$$M = P*A \bmod 65333$$

M = 7562500*47081 mod 65333

M = 43091

Km = M mod 3 +1

Km = 3

5. นำ Block ของ Plaintext มาเข้ารหัสลับในอัลกอริทึมที่ 3 ตามค่าของ Km ซึ่งก็คือ NXOR โดยจะได้ Ciphertext ดังนี้

Ciphertext จาก อัลกอริทึม NXOR คือ 03 40 4d 73 14 10 05 1e 16 4f 0d 26 10 38 39 48

7. นำค่า A และ P ที่ได้มาคำนวณหาว่าจะแทรก Noise ไปที่ตำแหน่งใด ซึ่งในที่นี้จะกำหนดตำแหน่งเป็นไบต์ ดังนั้นจะมีตำแหน่งที่ Noise สามารถแทรกได้ 16 ตำแหน่งนับจากซ้ายไปขวา

N = P*A mod 16 +1

N = 7562500*47081 mod 16 +1

N = 5

8. สุ่มค่า Noise ที่จะแทรกเข้าที่ตำแหน่งที่ 5 โดยค่าที่สุ่มจะต้องไม่เกิน 16 บิต ซึ่งก็ค่า 65535 ในเลขฐานสิบ

Noise = 3074 ฐาน 10

Noise = 0c02 ฐาน 16

9. แทรกค่า Noise = 0c02 ไปที่ตำแหน่งที่ 5 ของ Ciphertext ของอัลกริทึม XNOR

03 40 4d 73 0c02 14 10 05 1e 16 4f 0d 26 10 38 39 48

10. นำค่า A ฐาน 16 ใส่ไว้ข้างหน้าสุดของ Ciphertext ที่ได้แทรก Noise แล้ว ซึ่งจะได้ Ciphertext สุดท้าย

Ciphertext = *b7e9* 03 40 4d 73 0c02 14 10 05 1e 16 4f 0d 26 10 38 39 48

3.6.3 การถอดรหัสลับ

1. เมื่อได้รับ Ciphertext เรียบร้อยแล้วต่อมาต้องแยก 16 บิตแรกออกมาก่อนซึ่งก็คือค่า A โดย Ciphertext ที่ได้รับมาคือ

b7e9 03 40 4d 73 0c 02 14 10 05 1e 16 4f 0d 26 10 38 39 48

Ciphertext ที่ถูกแยก 16 บิตแรกออกมาแล้วคือ

03 40 4d 73 0c 02 14 10 05 1e 16 4f 0d 26 10 38 39 48

A = b7c9 ฐาน 16

A = 47081 ฐาน 10

2. บวกค่า KeyA ทุกๆ 8 บิต โดยมีค่าเท่ากับ 2750 จากนั้นนำมายกกำลังสอง

P = 2750*2750

P = 7562500

3. นำค่า A และ P ที่ได้มาคำนวณหาว่า Ciphertext ที่ได้รับมานั้นถูกแทรก Noise ไว้ที่ตำแหน่งใด ซึ่งถูกกำหนดตำแหน่งเป็นไบต์ ดังนั้นจะมีตำแหน่งที่ Noise สามารถแทรกได้ 16 ตำแหน่งนับจากซ้ายไปขวา

N = P*A mod16 +1

N = 7562500*47081 mod 16 +1

N = 5

4. ดึง Noise ตำแหน่งที่ 5 โดยจะดึงออกมา 16 บิต ซึ่งก็คือดึงไบต์ที่ 5 และ 6 ออกจาก Ciphertext ที่ยังมี Noise อยู่

03 40 4d 73 0c 02 14 10 05 1e 16 4f 0d 26 10 38 39 48

Ciphertext ที่ดึง Noise ออกแล้ว

03 40 4d 73 14 10 05 1e 16 4f 0d 26 10 38 39 48

5. นำค่า A และ P ที่ได้ มาคำนวณหา Km โดยค่า Km นี้จะเป็นตัวบ่งชี้ว่าจะใช้อัลกอริทึมใดในการถอดรหัสลับ

M = P*A mod 65333

M = 7562500*47081 mod 65333

M = 43091

Km = M mod 3 +1

Km = 3

6. เนื่องจากค่า Km = 3 ดังนั้นเลือกนำ Ciphertext ไปเข้าการถอดรหัสลับจากอัลกอริทึมที่ 3 นั่นคือ อัลกิริทึม NXOR โดยที่เมื่อถอดรหัสลับเรียบร้อยแล้วจะได้ Plaintext ออกมาอย่างถูกต้อง

Plaintext คือ 43 6f 6d 70 75 74 65 72 45 6e 67 69 6e 65 65 72

7. นำ Plaintext ที่อยู่ในเลขฐาน 16 แต่ละไบต์มาเทียบตามตารางที่ 3.1

Plaintext คือ ComputerEngineer

ตารางที่ 3.1 ตารางในการเปลี่ยนตัวอักษรเป็นเลขฐาน 16 หรือเปลี่ยนเลขฐาน 16 เป็นตัวอักษร [5]

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152		Ž 017D	
90		' 2018	' 2019	" 201C	" 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153		ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

บทที่ 4

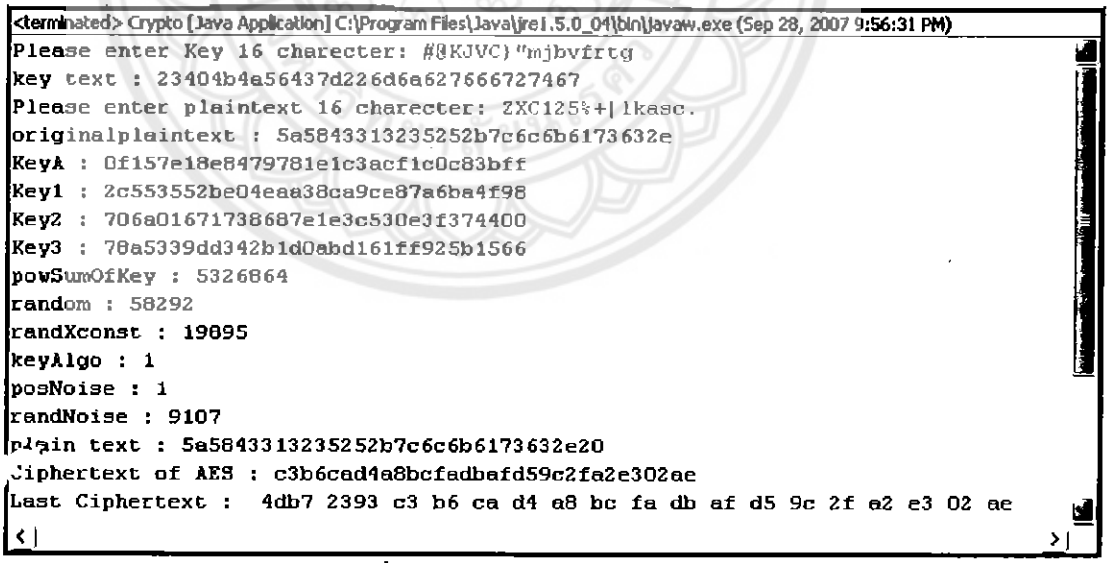
ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบอัลกอริทึมในการเข้ารหัสลับและถอดรหัสลับ ซึ่งจะทดสอบว่าเมื่อเข้ารหัสลับแล้วสามารถถอดรหัสลับกลับมาเป็นข้อความตั้งต้นได้อย่างถูกต้อง โดยการทดสอบจะนำหลายๆข้อความตั้งต้นที่แตกต่างกันมาทดสอบ ซึ่งการทดสอบนั้นจะใช้โปรแกรมภาษา Java ในการทดสอบ

4.1 ทดสอบการเข้ารหัสลับและถอดรหัสลับ

ในการทดสอบการเข้ารหัสลับนั้น จะใช้ขนาดของ Block ในการเข้ารหัสลับเป็น 128 บิต ขนาดของ Key ที่ใช้การเข้ารหัสลับและถอดรหัสลับเป็น 128 บิต และอัลกอริทึมที่ใช้ในการเข้ารหัสลับและถอดรหัสลับมี 3 อัลกอริทึมโดยที่ อัลกอริทึมที่ 1 คือ AES อัลกอริทึมที่ 2 คือ XOR และอัลกอริทึมที่ 3 คือ NXOR

ตัวอย่างที่ 1 ทดสอบการเข้ารหัสลับและถอดรหัสลับ



รูปที่ 4.1 แสดงการเข้ารหัสลับของตัวอย่างที่ 1

จากรูปที่ 4.1 แสดงการเข้ารหัสลับโดยมีขั้นตอนดังนี้

1. รับอินพุตของ Key เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

#@KJVC}"mjbvfrtg

2. นำ Key ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

23404b4a56437d226d6a627666727467

3. นำ Key มา Generate ได้ผลดังนี้

KeyA = 0f157e18e8479781e1c3acf1c0c83bff

Key1 = 2c553552be04eaa38ca9ce87a6ba4f98

Key2 = 706a01671738687e1e3c530e3f374400

Key3 = 78a5339dd342b1d0abd161ff925b1566

4. รับอินพุตของ Plaintext เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

ZXC125%+|lkasc.

5. นำ Plaintext ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

5a5843313235252b7c6c6b6173632e

6. คำนวณค่าผลรวมทุกๆ 8 บิต ของ KeyA แล้วนำมายกกำลังสอง ซึ่งก็คือ

5326864

7. สุ่มค่าตัวเลขไม่เกิน 16 บิตขึ้นมา ซึ่งก็คือ

58292

8. นำค่าจากข้อที่ 6 ซึ่งก็คือค่า P และข้อที่ 7 ซึ่งก็คือค่า a มาคำนวณหาค่า A ซึ่งค่าที่ได้จากการคำนวณคือ

19895

12. นำค่า A ที่ได้มาคำนวณหาว่าต้องเลือกอัลกอริทึมใดในการเข้ารหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 1

13. นำ Plaintext ไปเข้ารหัสลับอัลกอริทึมที่ 1 ซึ่งก็คือ AES โดยจะได้ Ciphertext คือ

c3b6cad4a8bcfadbafd59c2fa2e302ae

14. นำค่า A ที่ได้มาคำนวณหาว่าจะแทรก Noise ไปที่ตำแหน่งไบต์ใดใน 16 ตำแหน่ง ซึ่งก็คือ

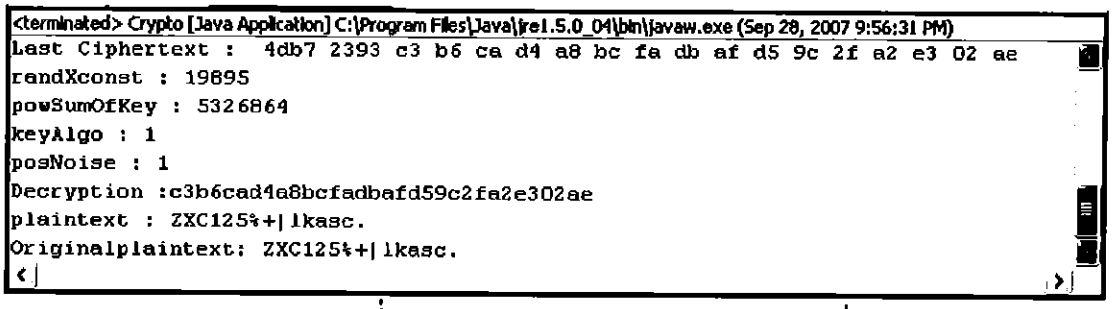
ตำแหน่งที่ 1

15. สุ่มค่า Noise ขึ้นมาไม่เกิน 16 บิต ซึ่งก็คือ

9107

16. นำ Ciphertext จากอัลกอริทึมที่ 1 มาแทรก Noise = 9107 ไปยังตำแหน่งที่ 1 หลังจากนั้นนำ ค่า A มาใส่ไว้ข้างหน้าสุด ในที่สุดจะได้ Ciphertext ที่อยู่ในรูปฐาน 16 ดังนี้

4d b7 23 93 c3 b6 ca d4 a8 bc fa db af d5 9c 2f a2 e3 02 ae



รูปที่ 4.2 แสดงการถอดรหัสลับของตัวอย่างที่ 1

จากรูปที่ 4.2 แสดงการถอดรหัสลับโดยมีขั้นตอนดังนี้

1. นำ Ciphertext ที่ได้มาดึง 16 บิตแรกออกก่อน ซึ่งก็คือ

19895

2. คำนวณค่าผลรวมทุกๆ 8 บิต ของ KeyA ซึ่งก็คือ

5326864

3. นำค่าที่ได้จากข้อที่ 1 และ 2 ซึ่งก็คือค่า A และ P มาคำนวณหาว่าต้องใช้อัลกอริทึมใดในการถอดรหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 1

4. นำค่า A และ P มาคำนวณหาว่าได้มีการแทรก Noise ไปที่ตำแหน่งใด เพื่อที่จะดึง Noise ออก โดยค่าที่ได้คือ

ตำแหน่งที่ 1

5. ดึงค่า Noise ตำแหน่งไบต์ที่ 1 และ 2 ออก ก็จะเหลือ Ciphertext คือ

c3b6cad4a8bcfadbafd59c2fa2e302ae

6. นำ Ciphertext ที่ได้ไปถอดรหัสลับอัลกอริทึม AES แล้วจะได้ Plaintext ดังนี้

ZXC125%+|lkasc.

ตัวอย่างที่ 2 ทดสอบการเข้ารหัสลับและถอดรหัสลับ

```
<terminated> Crypto [Java Application] C:\Program Files\Java\jre1.5.0_04\bin\javaw.exe (Sep 28, 2007 10:40:42 PM)
Please enter Key 16 charecter: aa][a+&nv321'}yH
key text : 61615d5b612b266e76333231277d7948
Please enter plaintext 16 charecter: loVEC@mPu!";62N
originalplaintext : 6c6f564543406d507521223b36324e
KeyA : 61a1d0ca161bc80fdb7522206126a5fb
Key1 : 00c08d917730ee61ad461011465bdcb3
Key2 : 1e5e2f3569643770240a5d5f1e595a04
Key3 : d6ad1cfce10ba510e6e48d77560dbd85
powSumOfKey : 3553225
random : 38938
randXconst : 59262
keyAlgo : 2
posNoise : 15
randNoise : 39203
Ciphertext of xor : 723179702a245a20512b7f64286b1424
Last Ciphertext : e77e 72 31 79 70 2a 24 5a 20 51 2b 7f 64 28 6b 9923 14 24
```

รูปที่ 4.3 แสดงการเข้ารหัสลับของตัวอย่างที่ 2

จากรูปที่ 4.3 แสดงการเข้ารหัสลับโดยมีขั้นตอนดังนี้

1. รับอินพุตของ Key เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

   aa][a+&nv321'}yH

2. นำ Key ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

   61615d5b612b266e76333231277d7948

3. นำ Key มา Generate ได้ผลดังนี้

   KeyA = 61a1d0ca161bc80fdb7522206126a5fb

   Key1 = 00c08d917730ee61ad461011465bdcb3

   Key2 = 1e5e2f3569643770240a5d5f1e595a04

   Key3 = d6ad1cfce10ba510e6e48d77560dbd85

4. รับอินพุตของ Plaintext เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

   loVEC@mPu!";62N

5. นำ Plaintext ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

   6c6f564543406d507521223b36324e

6. คำนวณค่าผลรวมทุกๆ 8 บิต ของ KeyA แล้วนำมายกกำลังสอง ซึ่งก็คือ

   3553225

7. สุ่มค่าตัวเลขไม่เกิน 16 บิตขึ้นมา ซึ่งก็คือ

   38938

8. นำค่าจากข้อที่ 6 ซึ่งก็คือค่า P และข้อที่ 7 ซึ่งก็คือค่า a มาคำนวณหาค่า A ซึ่งค่าที่ได้จากการคำนวณคือ

59262

12. นำค่า A ที่ได้มาคำนวณหาว่าต้องเลือกอัลกอริทึมใดในการเข้ารหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 2

13. นำ Plaintext ไปเข้ารหัสลับอัลกอริทึมที่ 2 ซึ่งก็คือ XOR โดยจะได้ Ciphertext คือ

723179702a245a20512b7f64286b1424

14. นำค่า A ที่ได้มาคำนวณหาว่าจะแทรก Noise ไปที่ตำแหน่งไบต์ใดใน 16 ตำแหน่ง ซึ่งก็คือ

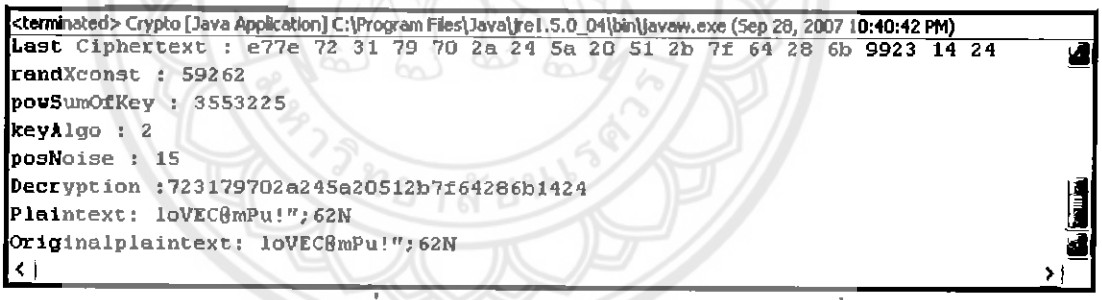
ตำแหน่งที่ 15

15. สุ่มค่า Noise ขึ้นมาไม่เกิน 16 บิต ซึ่งก็คือ

39203

16. นำ Ciphertext จากอัลกอริทึมที่ 2 มาแทรก Noise = 39203 ไปยังตำแหน่งที่ 15 หลังจากนั้นนำ ค่า A มาใส่ไว้ข้างหน้าสุด ในที่สุดจะได้ Ciphertext ที่อยู่ในรูปฐาน 16 ดังนี้

e7 7e 72 31 79 70 2a 24 5a 20 51 2b 7f 64 28 6b 99 23 14 24



รูปที่ 4.4 แสดงการถอดรหัสลับของตัวอย่างที่ 2

จากรูปที่ 4.4 แสดงการถอดรหัสลับโดยมีขั้นตอนดังนี้

1. นำ Ciphertext ที่ได้มาดึง 16 บิตแรกออกก่อน ซึ่งก็คือ

59262

3. คำนวณณค่าผลรวมทุกๆ 8 บิต ของ KeyA ซึ่งก็คือ

3553225

3. นำค่าที่ได้จากข้อที่ 1 และ 2 ซึ่งก็คือค่า A และ P มาคำนวณหาว่าต้องใช้อัลกอริทึมใดในการถอดรหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 2

4. นำค่า A และ P มาคำนวณหาว่าได้มีการแทรก Noise ไปที่ตำแหน่งใด เพื่อที่จะดึง Noise ออก โดยค่าที่ได้คือ

ตำแหน่งที่ 15

5. ดึงค่า Noise ตำแหน่งไบต์ที่ 15 และ 16 ออก ก็จะเหลือ Ciphertext คือ

723179702a245a20512b7f64286b1424

6. นำ Ciphertext ที่ได้ไปถอดรหัสลับอัลกอริทึม AES แล้วจะได้ Plaintext ดังนี้

loVEC@mPu!";62N

ตัวอย่างที่ 3 ทดสอบการเข้ารหัสลับและถอดรหัสลับ

รูปที่ 4.5 แสดงการเข้ารหัสลับของตัวอย่างที่ 3

จากรูปที่ 4.5 แสดงการเข้ารหัสลับโดยมีขั้นตอนดังนี้

1. รับอินพุตของ Key เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

123456789abcdefg

2. นำ Key ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

31323334353637383961626364656667

3. นำ Key มา Generate ได้ผลดังนี้

KeyA = c0017678e799000f02b6a88fcbe27f3a

Key1 = f133454cd2af37373bd7caecaf87195d

Key2 = 3f7e090718667f707d495770341d0045

Key3 = da26576855a6393c28a804cf6b415a33

4. รับอินพุตของ Plaintext เป็นตัวอักขระ 16 ตัว ซึ่งก็คือ

yhdnckol,1234567

5. นำ Plaintext ที่ได้มาเปลี่ยนเป็นฐาน 16 ซึ่งก็คือ

7968646e636b6f6c2c31323334353637

6. คำนวณค่าผลรวมทุกๆ 8 บิต ของ KeyA แล้วนำมายกกำลังสอง ซึ่งก็คือ

3759721

7. สุ่มค่าตัวเลขไม่เกิน 16 บิตขึ้นมา ซึ่งก็คือ

83439

8. นำค่าจากข้อที่ 6 ซึ่งก็คือค่า P และข้อที่ 7 ซึ่งก็คือค่า a มาคำนวณหาค่า A ซึ่งค่าที่ได้จากการคำนวณคือ

17439

12. นำค่า A ที่ได้มาคำนวณหาว่าต้องเลือกอัลกอริทึมใดในการเข้ารหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 3

13. นำ Plaintext ไปเข้ารหัสลับอัลกอริทึมที่ 3 ซึ่งก็คือ NXOR โดยจะได้ Ciphertext คือ

5c314c794932292f7b664903200b137b

14. นำค่า A ที่ได้มาคำนวณหาว่าจะแทรก Noise ไปที่ตำแหน่งไบต์ใดใน 16 ตำแหน่ง ซึ่งก็คือ

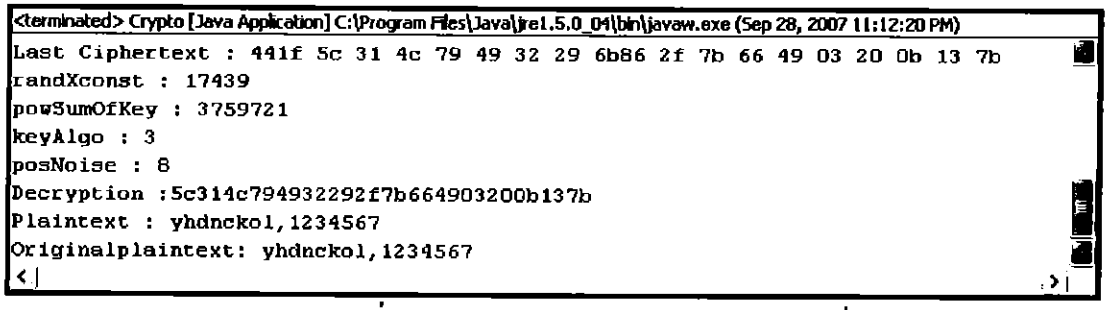
ตำแหน่งที่ 8

15. สุ่มค่า Noise ขึ้นมาไม่เกิน 16 บิต ซึ่งก็คือ

27526

16. นำ Ciphertext จากอัลกอริทึมที่ 3 มาแทรก Noise = 27526 ไปยังตำแหน่งที่ 8 หลังจากนั้นนำ ค่า A มาใส่ไว้ข้างหน้าสุด ในที่สุดจะได้ Ciphertext ที่อยู่ในรูปฐาน 16 ดังนี้

44 1f 5c 31 4c 79 49 32 29 6b 86 2f 7b 66 49 03 20 0b 13 7b



รูปที่ 4.6 แสดงการถอดรหัสลับของตัวอย่างที่ 3

จากรูปที่ 4.6 แสดงการถอดรหัสลับโดยมีขั้นตอนดังนี้

1. นำ Ciphertext ที่ได้มาดึง 16 บิตแรกออกก่อน ซึ่งก็คือ

17439

4. คำนวณณค่าผลรวมทุกๆ 8 บิต ของ KeyA ซึ่งก็คือ

3759721

3. นำค่าที่ได้จากข้อที่ 1 และ 2 ซึ่งก็คือค่า A และ P มาคำนวณหาว่าต้องใช้อัลกอริทึมใดในการถอดรหัสลับ ซึ่งก็คือ

อัลกอริทึมที่ 3

4. นำค่า A และ P มาคำนวณหาว่าได้มีการแทรก Noise ไปที่ตำแหน่งใด เพื่อที่จะดึง Noise ออก โดยค่าที่ได้คือ

ตำแหน่งที่ 8

5. ดึงค่า Noise ตำแหน่งไบต์ที่ 8 และ 9 ออก ก็จะเหลือ Ciphertext คือ

5c314c794932292f7b664903200b137b

6. นำ Ciphertext ที่ได้ไปถอดรหัสลับอัลกอริทึม NXOR แล้วจะได้ Plaintext ดังนี้

yhdnckol,1234567

4.2 สรุปผลการทดสอบ

จากการทดสอบด้วยมือ และทดสอบด้วยโปรแกรม สรุปได้ว่า วิธีการเข้ารหัสลับโดยมีการเลือกอัลกอริทึมในการเข้ารหัสลับและเพิ่ม Noise เข้าไปใน Ciphertext สามารถทำการเข้ารหัสลับและถอดรหัสลับได้จริง โดยข้อความตั้งต้น และข้อความที่ได้จากการถอดรหัสลับ เป็นข้อความเดียวกัน และในทุกๆ ครั้งที่มีการเข้ารหัสด้วยข้อความเดียวกัน Ciphertext ที่ได้ จะมีค่าที่แตกต่างกันเสมอ โดยค่าและตำแหน่งของ Noise ที่แทรกอยู่ภายใน Ciphertext ก็จะมีค่าที่ไม่แน่นอน รวมทั้งค่าของ A ก็ยังต่างกันไปด้วย และนอกจากนั้น ในกรณีที่ใช้ Key ในการถอดรหัสลับเป็นคนละตัวกับ Key ที่ใช้ในการเข้ารหัสลับ ข้อความตั้งต้นที่ได้ออกมา จะแตกต่างกับข้อความตั้งต้นเดิมที่ถูกต้องอย่างสิ้นเชิงนั่นก็คือต้องมี Key ที่ถูกต้องเท่านั้นถึงจะสามารถถอดรหัสลับได้ข้อความตั้งต้นดังเดิม

4.3 การวิเคราะห์รหัสลับ

การวิเคราะห์รหัสลับ (Cryptanalysis) เป็นการกำหนดสมมติฐานพื้นฐาน โดยให้ผู้ไม่ประสงค์ดีทราบถึงโครงสร้างของระบบรหัสลับเป็นอย่าง ดี และจะวิเคราะห์ตามระดับขีดความสามารถของฝ่ายตรงข้ามที่เข้ามาโจมตี ซึ่งแบ่งได้ออกเป็นระดับดังนี้

1. กรณีรู้ข้อความไซเฟอร์อย่างเดียว ในกรณีนี้จะไม่สามารถวิเคราะห์หาค่า Key ได้ เพราะค่า A ซึ่งเป็นค่าที่เกิดจากความสัมพันธ์ของ ข้อความไซเฟอร์และค่า Key นั้น เกิดจากการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อน และมีการสุ่มค่าตัวเลขขึ้นมาเป็นส่วนประกอบในการคำนวณนั้น

ด้วย จึงทำให้ไม่สามารถวิเคราะห์หาค่า Key ได้ และไม่สามารถวิเคราะห์หาข้อความต้นฉบับได้ เพราะ ภายในข้อความไซเฟอร์นั้น จะมีการแทรก Noise เอาไว้ ซึ่งตำแหน่งที่แทรก Noise นั้น จะมีตำแหน่งที่ไม่แน่นอน จึงทำให้ไม่สามารถวิเคราะห์หาข้อความต้นฉบับได้

2. กรณีรู้ข้อความไซเฟอร์และข้อความต้นฉบับ ในกรณีนี้จะไม่สามารถวิเคราะห์หาค่า Key ได้ เพราะ ค่า A ซึ่งเป็นค่าที่เกิดจากความสัมพันธ์ของ ข้อความไซเฟอร์ ข้อความต้นฉบับ และค่า Key นั้น เกิดจากการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อนและมีการสุ่มค่าตัวเลขขึ้นมาเป็นส่วนประกอบในการคำนวณนั้นด้วย จึงทำให้ไม่สามารถวิเคราะห์หาค่า Key ได้

3. กรณีเลือกข้อความต้นฉบับได้ และทำการเข้ารหัสจนได้ข้อความไซเฟอร์ ในกรณีนี้จะไม่สามารถวิเคราะห์หาค่า Key ได้ เพราะ ค่า A ซึ่งเป็นค่าที่เกิดจากความสัมพันธ์ของ ข้อความไซเฟอร์ ข้อความต้นฉบับ และค่า Key นั้น เกิดจากการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อนและมีการสุ่มค่าตัวเลขขึ้นมาเป็นส่วนประกอบในการคำนวณนั้นด้วย จึงทำให้ไม่สามารถวิเคราะห์หาค่า Key ได้

4. กรณีเลือกข้อความไซเฟอร์ได้ และทำการถอดรหัสจนได้ข้อความต้นฉบับ ในกรณีนี้จะไม่สามารถวิเคราะห์หาค่า Key ได้ เพราะ ค่า A ซึ่งเป็นค่าที่เกิดจากความสัมพันธ์ของ ข้อความไซเฟอร์ ข้อความต้นฉบับ และค่า Key นั้น เกิดจากการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อนและมีการสุ่มค่าตัวเลขขึ้นมาเป็นส่วนประกอบในการคำนวณนั้นด้วย จึงทำให้ไม่สามารถวิเคราะห์หาค่า Key ได้

5. กรณีการวิเคราะห์ส่วนต่าง ซึ่งเป็นวิธีที่ลองเข้ารหัสด้วยข้อความต้นฉบับหลายๆ ฉบับที่ต่างกัน แล้วเปรียบเทียบขอความไซเฟอร์ที่ได้ ในกรณีนี้จะไม่สามารถวิเคราะห์หาค่า Key ได้ เพราะ ค่า A ซึ่งเป็นค่าที่เกิดจากความสัมพันธ์ของ ข้อความไซเฟอร์ ข้อความต้นฉบับ และค่า Key นั้น เกิดจากการคำนวณทางคณิตศาสตร์ที่มีความซับซ้อนและมีการสุ่มค่าตัวเลขขึ้นมาเป็นส่วนประกอบในการคำนวณนั้นด้วย จึงทำให้ไม่สามารถวิเคราะห์หาค่า Key ได้

บทที่ 5
บทสรุป

โครงงานนี้ได้ทำการออกแบบอัลกอรึทึมสำหรับการเข้ารหัสลับและถอดรหัสลับที่มีอยู่แล้วให้ปลอดภัยมากยิ่งขึ้น โดยมีการเลือกว่าจะใช้อัลกอริทึมใดในการเข้ารหัสลับและนำ Ciphertext ที่ได้ผ่านการเข้ารหัสลับแล้วมาเพิ่ม Noise เพื่อเพิ่มความซับซ้อนส่งผลให้ Cracker ต้องใช้เวลาในการคำนวณมากขึ้นในการค้นหา Key

5.1 วิเคราะห์ผลการทดลอง

จากผลการทดสอบเห็นได้ว่า หลังจากที่ข้อความตั้งต้นผ่านการเข้ารหัสลับแล้ว Ciphertext ที่ได้จะมีขนาดที่มากขึ้น เนื่องจากจะมีการเพิ่มในส่วนของ Noise และค่า A ที่เป็นค่าที่จะนำไปคำนวณเพื่อการถอดรหัสลับ แต่ก็ทำให้เวลาในการคำนวณมีมากขึ้นตามไปด้วยซึ่งทำให้ Cracker ต้องใช้เวลาในการคำนวณมากขึ้นเพื่อค้นหา Key และนอกจากนี้ จากการที่มีการสุ่มค่าเพื่อนำมาใช้ในการคำนวณการเลือกอัลกอรึทึมในการเข้ารหัสลับ หาตำแหน่งของ Noise และการสุ่มค่า Noise ทำให้ในแต่ละครั้งที่มีการเข้ารหัสลับ แม้จะเป็นข้อความตั้งต้นเดียวกัน แต่ Ciphertext ที่ได้ ก็จะมีค่าที่ต่างกันทำให้ยากต่อการวิเคราะห์หา Key

5.2 ปัญหาและแนวทางแก้ไข

ในระหว่างการทำโครงงานนี้ได้ประสบปัญหาดังนี้

1. อัลกอริทึมของตัววิธีการเข้ารหัสลับในตอนแรก ไม่สามารถถอดรหัสลับกลับมาเป็นข้อความตั้งได้เหมือนเดิม แนวทางแก้ไข คือ ทำการศึกษาหาข้อมูลเกี่ยวกับการเข้ารหัสลับใหม่แล้วนำไปปรับปรุงวิธีการเข้ารหัสลับอันเก่า เพื่อให้สามารถถอดรหัสลับกลับมาเป็นข้อ ความตั้งได้เหมือนเดิม

2. อัลกอริทึมของวิธีการเข้ารหัสลับ ตรงส่วนที่ทำการเลือกอัลกอรึทึมที่ใช้เข้ารหัสลับ มีการกำหนดค่าที่ตายตัว ทำให้ง่ายต่อการ Crack แนวทางแก้ไข คือ ทำการใช้การสุ่มค่าตัวเลขขึ้นมา แล้วมาคำนวณร่วมกับค่า Key เพื่อเลือกอัลกอรึทึมที่ใช้ในการเข้ารหัสลับ

5.3 สรุปผลการทดลอง

จากการทดสอบสรุปได้ว่า ในการเข้ารหัสลับแต่ละครั้งนั้นสามารถถอดรหัสลับกลับมาเป็นข้อความตั้งต้นได้เหมือนเดิมทุกครั้ง ซึ่งในแต่ละครั้งของการเข้ารหัสลับทั้งอัลกอริทึมที่ใช้ในการเข้ารหัสลับ, ค่า A, ตำแหน่ง ของ Noise และค่า Noise จะมีค่าที่ไม่แน่นอน ถึงแม้ว่าเป็นข้อความตั้ง

ต้นเดียวกันแต่ Ciphertext นั้นมีความแตกต่างกัน ทำให้ยากต่อการวิเคราะห์หา Key และ Cracker ต้องใช้เวลาในการคำนวณมากขึ้นเพื่อหา Key ให้พบ

5.4 ข้อเสนอแนะ

1. สำหรับอัลกอรึทึมต่างๆ ที่นำมาใช้ สามารถเปลี่ยนแปลง โดยเลือกใช้วิธีการเข้ารหัสลับแบบต่างๆ มาประยุกต์ใช้ได้

2. รูปแบบของการเข้ารหัสลับ สามารถทำการประยุกต์ หรือปรับปรุงในบางขั้นตอน เพื่อทำให้เกิดความซับซ้อนยิ่งขึ้นได้ ขึ้นอยู่กับผู้ใช้งาน

เอกสารอ้างอิง

[1] บุญฤทธิ์ อินทิยศ. **"วิทยาการรหัสลับ (ตอนที่ 1)".** [Online]. Available: http://www.vcharkarn.com/include/article/showarticle.php?Aid=436. 2006.

[2] Borja Sotomayor. **"Introduction to cryptography".** [Online]. Available: http://gdp.globus.org/gt4-tutorial/singlehtml/progtutorial_0.2.1.html#id2566560. 2005.

[3] Lux Scientiae. **"Section 4: Symmetric and Asymmetric Encryption in a Nutshell".** [Online]. Available: http://www.uic.edu/depts/accc/newsletter/adn26/asymmetric.jpg. 2000.

[4] connect@uic.edu. **"Asymmetric or Public Key Encryption".** [Online]. Available: http://www.uic.edu/depts/accc/newsletter/adn26/asymmetric.jpg. 2000.

[5] W. Polmann. **"CP1256".** [Online]. Available: http://www.ecm-engineering.de/espan/Serv_TXTKONV.htm. 2004.

[6] ลัญฉกร วุฒิสิทธิกุลกิจ. ธงชัย โรจน์กังสดาล. วราภร ศรีเชวงทรัพย์. นพดล พรมภักษร. สุวิทย์ นาคพีระยุทร (ที่ปรึกษา). วิทยาการรหัสลับเบื้องต้น. ครั้งที่ 1. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2548.

[7] derkeiler.com. **"AES java implementation".** [Online]. Available: http://www.derkeiler.com/Newsgroups/sci.crypt/2004-03/1353.html. 2004.

[8] derkeiler.com. **"AES java implementation".** [Online]. Available: http://www.derkeiler.com/Newsgroups/sci.crypt/2004-03/1353.html. 2004.

[9] University of Limerick. **"CEncrypt.java".** [Online]. Available: http://www.ecestudents.ul.ie/Course_Pages/Btech_ITT/Modules/ET4263/More%20Samples/CEncrypt.java.html. 2004.

ประวัติผู้เขียนโครงงาน

ชื่อ นายพชร ศรีสุข

ภูมิลำเนา 138/3 ถนนรจนา ตำบลตาคลี อำเภอตาคลี
จังหวัดนครสวรรค์ 60140

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจาก
  โรงเรียนตาคลีประชาสรรค์
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail potchara926@hotmail.com

ชื่อ นางสาวสุธาทิพย์ คล้ายเหล็ง

ภูมิลำเนา 180/2 หมู่ที่4 ตำบลอินทร์บุรี อำเภออินทร์บุรี
จังหวัดสิงห์บุรี 16110

ประวัติการศึกษา

- จบการศึกษาระดับมัธยมศึกษาจาก
  โรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail me_mee19@hotmail.com