ปลั๊กอินการเข้ารหัสของ UltraVNC

Data stream Encryption plug-in for UltraVNC

นางสาวปวีณา ดวงแก้วกูล รหัส 45380084
นายศุภพงศ์ เยาวรัตน์ รหัส 45380121
นางสาวสุชาดา อินต๊ะชัย รหัส 45380137

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน ปลั๊กอินการเข้ารหัสของ UltraVNC

ผู้ดำเนินโครงงาน นางสาวปวีณา ดวงแก้วกูล รหัส 45380084
นายศุภพงศ์ เยาวรัตน์ รหัส 45380121
นางสาวสุชาดา อินต๊ะชัย รหัส 45380137

อาจารย์ที่ปรึกษา ดร.สุรเดช จิตประไพกุลศาล

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2548

.....

คณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการตรวจสอบโครงงานวิศวกรรม

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

.....กรรมการ
(ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

หัวข้อโครงงาน	ปลั๊กอินการเข้ารหัสของ UltraVNC		
ผู้ดำเนินโครงงาน	นางสาวปวีณา	ดวงแก้วกูล	รหัส 45380084
	นายศุภพงศ์	เยาวรัตน์	รหัส 45380121
	นางสาวสุชาดา	อินต๊ะชัย	รหัส 45380137
อาจารย์ที่ปรึกษา	ดร.สุรเดช	จิตประไพกุลศาล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2548		

.....

บทคัดย่อ

VNC เป็นซอฟต์แวร์แบบ open source ที่ใช้สำหรับควบคุมคอมพิวเตอร์ระยะไกล เนื่องจาก VNC สามารถทำงานได้บนหลาย platform แต่ VNC ยังขาดระบบรักษาความปลอดภัยด้านการสื่อสาร UltraVNC สามารถแก้ปัญหานี้ได้โดยเราสามารถเพิ่ม plug-in สำหรับการเข้ารหัสลับข้อมูลเพื่อเพิ่มความมั่นคงให้แก่การสื่อสารได้ สำหรับโครงงานนี้เราจะศึกษาความเป็นได้ของการพัฒนา DSM โดยใช้อัลกอริทึมสำหรับการเข้ารหัสลับ 3 อัลกอริทึม คือ DES, AES และ RSA นอกจากนี้เรายังศึกษาผลกระทบต่อประสิทธิภาพการทำงานของเครื่องเมื่อมีการใช้ DSM

Project title	Data stream Encryption plug-in for UltraVNC		
Name	Miss Paweena	Duangkaewkul	ID. 45380084
	Mr. Supapong	yaovarat	ID. 45380121
	Miss Suchada	Intachai	ID. 45380137
Project advisor	Dr. Suradet	Jitprapaikulsarn	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic year	2005		

.....

Abstract

VNC is a very popular open source tool for remotely controlling computers; however, it lacks the security needed for any sensitive operation. UltraVNC resolves this problem by providing a mechanism to add a Data Stream Modulation (DSM) plug-in to improve the security of the communication. In this project we attempt to create three DSM's using three encryption algorithms: DES, AES and RSA. We also study the impact of using DSM on the performance of the computers involve.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้เกิดขึ้นเนื่องจากการทำงานร่วมกันในหลายๆส่วน บุคคลแรกที่ต้องกล่าวถึง คือ ดร.สุรเดช จิตประไพกุลศาล อาจารย์ที่ปรึกษา ดร.พนมขวัญ ริยะมงคล และ ดร.อัครพันธ์ วงศ์กังแห อาจารย์ที่ปรึกษาร่วม ที่ให้ความเอาใจใส่แนะนำ และช่วยเหลือเสมอ รวมถึงอาจารย์ท่านอื่นๆที่มิได้กล่าวถึงที่ได้คอยแนะนำ และให้คำปรึกษาจนคลายความข้องใจ ซึ่งต้องขอขอบพระคุณเป็นอย่างยิ่งที่ให้การสนับสนุนผู้จัดทำโครงงานให้สามารถทำโครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้คณะผู้จัดทำมีวันนี้ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดู พร้อมทั้งให้โอกาสทางการศึกษามาเป็นอย่างดี และยังให้กำลังใจ เอาใจใส่อย่างเต็มที่ในทุกๆด้านอันหาที่เปรียบมิได้ คณะผู้จัดทำขอระลึกในพระคุณอันสุดประมาณค่า และขอกราบขอบคุณมา ณ ที่นี้

ปวีณา ดวงแก้วกูล
ศุภพงศ์ เยาวรัตน์
สุชาดา อินต๊ะชัย

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในยุคของเทคโนโลยีการสื่อสารที่ก้าวหน้าไปทุกวัน การพัฒนาเทคโนโลยีในด้านต่างๆได้เปลี่ยนแปลงไปอย่างรวดเร็ว ทั้งในด้านความสามารถและความสะดวกสบาย เทคโนโลยีในการส่งข้อมูลก็เป็นหนึ่งในนั้น ด้วยความสะดวกสบายที่เพิ่มขึ้นนั่นเองจึงทำให้ผู้คนนิยมใช้เทคโนโลยีเหล่านี้ในชีวิตประจำวันมากขึ้น ข้อมูลมากมายหลายล้านข้อมูลจึงถูกส่งกันในแต่ละวัน ข้อมูลเหล่านี้มีทั้งข้อมูลที่ไม่มีความสำคัญอะไรมากมาย และข้อมูลที่มีความสำคัญที่จำเป็นจะต้องเก็บรักษาไว้เป็นความลับ เช่น ข้อมูลสำคัญของหน่วยงานต่างๆ ข้อมูลทางด้านการเงิน หรือหมายเลขบัตรเครดิตที่ถูกส่งออกไปโดยผ่านธุรกิจ E-commerce เป็นต้น ซึ่งข้อมูลเหล่านี้นี่เองที่เป็นที่หมายปองของบรรดา Hacker ต่างๆซึ่งได้คิดวิธีการมากมายที่คอยดักจับเอาข้อมูลเหล่านั้นมาเพื่อการใดการหนึ่งซึ่งไม่ส่งผลดีต่อเจ้าของข้อมูลอย่างแน่นอน

โปรแกรม UltraVNC เป็นโปรแกรมที่ใช้ควบคุมหรือการเข้าถึงข้อมูลจากทางไกลในรูปแบบของ Remote Desktop ซึ่งช่วยอำนวยความสะดวกให้กับบุคคลที่ใช้งานเครื่องปลายทางที่ห่างไกลกัน ในปัจจุบัน Ultra VNC และโปรแกรมจำพวกเดียวกันนี้ได้รับความนิยมมากขึ้นเนื่องจากความเร่งรีบทางธุรกิจและ ความต้องการที่จะย่นเวลาการทำงานต่างๆ เมื่อมีผู้นำไปใช้มากขึ้นข้อมูลที่ถูกส่งออกไปก็มีจำนวนมากขึ้นข้อมูลส่วนหนึ่งซึ่งจำเป็นจะต้องปกปิดเป็นความลับและการถูกส่งออกไปโดยไม่มีการป้องกันใดๆ จึงเป็นช่องทางให้ผู้ไม่สงค์ประสงค์ดี นั้นมาลักลอบโจรกรรมข้อมูลได้โดยง่าย ด้วยเหตุนี้ ผนวกกับความที่ UltraVNC เป็นโปรแกรมแบบ Open source จึงได้มีการพัฒนา Plug-in ในการเข้ารหัสลับของข้อมูล (Data stream Encryption Plug in) ขึ้น โดยใช้หลักการของวิทยาการเข้ารหัสลับ(Cryptography) ซึ่งหลักการคือการทำให้ข้อมูลที่ส่งผ่านไปทางเครือข่ายอยู่ในรูปแบบที่ไม่สามารถอ่านออกได้ด้วยการเข้ารหัสลับ(Encryption) ทำให้ข้อมูลนั้นเป็นความลับ ซึ่งผู้ที่มีสิทธ์จริงเท่านั้นจะสามารถอ่านข้อมูลได้ด้วยการถอดรหัสลับ(Decryption) นั่นคือสามารถรักษาข้อมูลให้เป็นความลับ(Confidentiality) และการพิสูจน์ตัวจริงการใช้อำนาจ (Authentication & Authorization) สำหรับการเข้ารหัสและถอดรหัสนั้นจะอาศัยสมการทางคณิตศาสตร์ที่ซับซ้อน ซึ่ง plug-in ในรุ่นแรกๆ ยังไม่มีความสมบูรณ์ในการรักษาความปลอดภัยเพียงพอ ซึ่งบรรดา Hacker สามารถ Hack การเข้ารหัสในส่วนนี้ได้ง่าย กลุ่มของเราจึงได้ทำการพัฒนา Plug in ในส่วนนี้ขึ้นมาใหม่โดยพัฒนาจากตัวเดิมที่มีอยู่แล้วให้มีความปลอดภัยและสมบูรณ์ยิ่งขึ้น

1.2 วัตถุประสงค์โครงงาน

1. ศึกษาเกี่ยวกับทฤษฎีและหลักการในเขียนโปรแกรมเพื่อที่จะทำการเข้ารหัสในการส่งข้อมูลของโปรแกรม UltraVNC
2. สามารถออกแบบ Plug-in เพื่อความปลอดภัยในการส่งข้อมูลของโปรแกรม UltraVNC
3. สามารถศึกษาการเข้ารหัสข้อมูลลับแบบ Symmetric หรือ private key และ Asymmetric หรือ public key
4. การพัฒนา Software แบบ Open Source

1.3 ขอบข่ายการทำงาน

1. ศึกษาการเข้ารหัสข้อมูลลับทั้งแบบ Symmetric และแบบ Asymmetric
2. สร้าง Plug-in สำหรับเข้ารหัสลับ ของโปรแกรม UltraVNC
3. สร้างและพัฒนา Plug-in ของการเข้ารหัสลับ โดยใช้ภาษาซี

1.4 ขั้นตอนดำเนินงาน

1. ศึกษาเกี่ยวกับทฤษฎีและหลักการในสิ่งต่างๆเหล่านี้
   - หลักการในเขียนโปรแกรมเพื่อที่จะทำการเข้ารหัสในการส่งข้อมูลของโปรแกรม UltraVNC
   - หลักการทำงานของวิทยาการเข้ารหัสลับ ทั้งแบบ Symmetric และแบบ Asymmetric
   - การเขียนโปรแกรมการเข้ารหัสโดยใช้ภาษาซี
2. ออกแบบและพัฒนาโปรแกรม
3. ทดสอบโปรแกรม
4. ทำการปรับปรุงและแก้ไขโปรแกรม
5. วิเคราะห์การทดสอบพร้อมทั้งสรุปผล
6. จัดทำเป็นรูปเล่ม

1.5 แผนการดำเนินงาน

กิจกรรม	ปี 2548				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
1. ศึกษาหลักการในการเขียนโปรแกรมเพื่อที่จะทำการเข้ารหัสในการส่งข้อมูลของโปรแกรม UltraVNC ,หลักการทำงานของ Cryptography, การเขียนโปรแกรมการเข้ารหัสโดยใช้ภาษาซี	←	→			
2. ออกแบบและพัฒนาโปรแกรม		←	→	→	
3. ทดสอบโปรแกรม			←	→	
4. ทำการปรับปรุงและแก้ไขโปรแกรม				←	→
5. วิเคราะห์การทดสอบพร้อมทั้งสรุปผล				←	→
6. จัดทำรูปเล่มโครงงาน					← →

1.6 ผลที่คาดว่าจะได้รับ

1. เข้าใจวิธีการสร้าง Plug-in สำหรับการเข้ารหัสลับของโปรแกรม UltraVNC
2. เข้าใจหลักการทำงานของ Cryptography
3. สามารถสร้างและพัฒนา Plug-in ของการเข้ารหัสลับของโปรแกรม UltraVNC
4. ได้โปรแกรมเสริมการทำงานของการเข้ารหัสลับของโปรแกรม UltraVNC
5. ได้ Open Source Software

1.7 งบประมาณ

1. ค่าวัสดุสำนักงาน	เป็นเงิน	400	บาท
2. ค่าวัสดุคอมพิวเตอร์	เป็นเงิน	500	บาท
3. ค่าวัสดุไฟฟ้าและวิทยุ	เป็นเงิน	300	บาท
4. ค่าถ่ายเอกสาร	เป็นเงิน	1,000	บาท
5. ค่าวัสดุอื่น ๆ	เป็นเงิน	800	บาท
	รวมเป็นเงินทั้งสิ้น	3,000	บาท (สามพันบาทถ้วน)

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้เราจะกล่าวถึงหลักการวิทยาการรหัสลับ (Cryptography) เป็นส่วนสำคัญส่วนหนึ่งในเทคโนโลยีของการรักษาความปลอดภัยในระบบเครือข่าย Cryptography จะแบ่งเป็น 2 กลุ่มใหญ่ๆ คือ แบบกุญแจสมมาตร (Symmetric cryptography) และแบบกุญแจอสมมาตร (Asymmetric cryptography or Public Key Technology) โดยในแต่ละกลุ่มก็จะมีหลายวิธี ในที่นี้จะกล่าวถึงแบบสมมาตร 2 วิธี คือ DES และ AES ส่วนแบบอสมมาตรจะกล่าวถึง 1 วิธี คือ RSA

รูปที่ 2.1 การเข้ารหัสข้อมูล (Encryption)

2.1 วิทยาการรหัสลับ (Cryptography)

วิทยาการรหัสลับ (Cryptography) เป็นกระบวนในการเปลี่ยนแปลงข้อมูลที่สามารถอ่านได้ให้อยู่ในรูปที่ไม่สามารถอ่านให้เข้าใจได้ ซึ่งโดยทั่วไปแล้วการเข้ารหัสนี้จะทำก่อนที่จะจัดเก็บข้อมูลหรือส่งข้อมูลออกไป โดยนำข้อมูลกับกุญแจ (Key) มาผ่านกระบวนการทางคณิตศาสตร์ทำให้ได้ผลลัพธ์เป็นข้อมูลที่เข้ารหัส และเมื่อผู้รับต้องการอ่านข้อมูลก็นำเอาข้อมูลเข้ารหัสกับกุญแจมาผ่านกระบวนการทางคณิตศาสตร์ ก็จะได้ผลลัพธ์เป็นข้อมูลเดิมก่อนจะทำการเข้ารหัส

2.1.1 การเข้ารหัสลับข้อมูล (Encryption)

หมายถึง วิธีการเปลี่ยนแปลงข้อมูลเพื่อป้องกันไม่ให้ผู้อื่นสามารถเข้าใจข้อมูลของเราได้ โดยการนำข้อความที่สามารถอ่านได้ (Plain text, Clear Text) มาทำการเข้ารหัสเพื่อเปลี่ยนแปลงให้เป็นข้อความที่เข้ารหัส (Cipher Text) แล้วจึงส่งไปให้บุคคลที่เราต้องการที่จะติดต่อด้วย ซึ่งเป็นการป้องกันไม่ให้บุคคลอื่นอ่านข้อความของเราได้ ดังจะเห็นได้ในรูปที่ 2.1

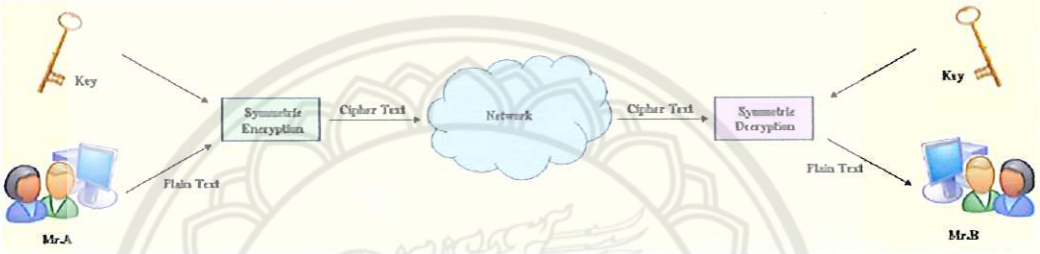
2.1.2 การถอดรหัสลับข้อมูล (Decryption)

หมายถึง วิธีการเปลี่ยนแปลงข้อมูลที่ได้จากการเข้ารหัสข้อมูลกลับเป็นข้อมูลก่อนที่จะถูกทำการเข้ารหัส การที่จะทำให้ข้อมูลเป็นความลับนั้นต้องทำให้ข้อมูลไม่สามารถถูกอ่านจากบุคคลอื่นได้ แต่ถูกอ่านได้จากบุคคลที่เราต้องการให้อ่านเท่านั้น ดังจะเห็นได้ในรูปที่ 2.1

จะเห็นได้ว่าจากระบบการเข้ารหัสข้อมูลและการถอดรหัสข้อมูลจะพบว่ากุญแจเป็นตัวแปรสำคัญ ดังนั้นระบบการเข้ารหัสข้อมูล สามารถแบ่งตามวิธีการใช้กุญแจได้ 2 วิธี คือ

2.1.3 วิทยาการรหัสลับแบบกุญแจสมมาตร (Symmetric cryptography)

หรือเรียกอีกอย่างว่า Single Key Algorithm หรือ Secret Key Algorithm คือ การเข้ารหัสข้อมูลด้วยกุญแจเพียงกุญแจเดียว (Single Key) ทั้งผู้ส่งและผู้รับ โดยผู้รับและผู้ส่งจะทำการตกลงกันก่อนว่าจะใช้กุญแจไหนหรือรูปแบบไหนในการเข้ารหัสข้อมูล เช่น ผู้ส่งและผู้รับตกลงกันว่าจะใช้เทคนิคการแทนที่ตัวอักษรที่อยู่ตำแหน่งถัดไป 1 ตำแหน่ง เช่น ถ้าเห็นตัวอักษร H ก็ให้เปลี่ยนไปเป็น I หรือเห็นตัวอักษร E ก็ให้เปลี่ยนไปเป็น F เป็นต้น ดังรูปที่ 2.2



รูปที่ 2.2 การเข้ารหัสลับแบบกุญแจอสมมาตร

ข้อดีของวิทยาการรหัสลับลับแบบสมมาตร

1. การเข้ารหัสและถอดรหัสข้อมูลใช้เวลาน้อย เนื่องจากอัลกอริทึมที่ใช้ไม่มีความซับซ้อน
2. ขนาดของข้อมูลหลังจากทำการเข้ารหัสแล้วจะมีขนาดเปลี่ยนแปลงไม่ใหญ่ไปกว่าเดิมมากนัก

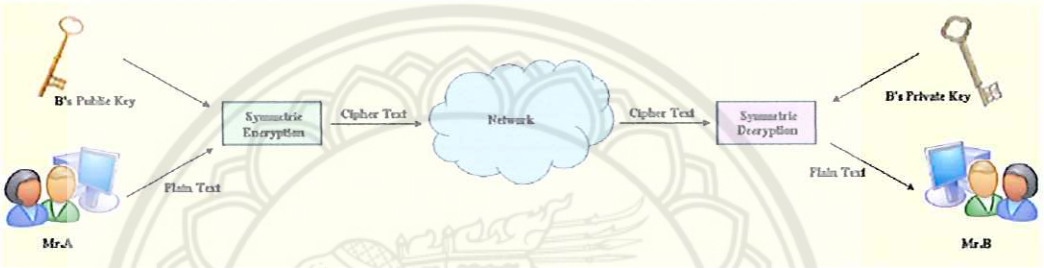
ข้อเสียของวิทยาการรหัสลับลับแบบสมมาตร

1. การจัดการกับกุญแจลับมีความยุ่งยาก เนื่องจากการติดต่อกับแต่ละคนจะใช้กุญแจไม่เหมือนกัน จากตัวอย่างข้างต้น Mr. A ติดต่อกับ Mr. B จะใช้กุญแจแบบหนึ่ง แต่ถ้า Mr. A ต้องการติดต่อกับ Mr. C จะใช้กุญแจอีกแบบหนึ่ง ดังนั้น Mr. A จะต้องจำไว้ด้วยว่ากุญแจแบบนี้ใช้ติดต่อกับใคร หรือถ้าติดต่อกับคนนี้จะใช้กุญแจแบบใด

2. การกระจายกุญแจลับ เนื่องจากการเข้ารหัสแบบนี้จะต้องใช้กุญแจลับ 1 ดอกต่อผู้รับ 1 คน ดังนั้นถ้า Mr. A ต้องการติดต่อกับคนมากๆ จะต้องส่งกุญแจลับที่ใช้ไปให้กับทุกคน

2.1.4 วิทยาการรหัสลับแบบกุญแจอสมมาตร (Asymmetric cryptography or Public Key Technology)

ระบบการเข้ารหัสแบบกุญแจอสมมาตรได้ถูกคิดค้นโดย นายวิทฟิลด์ ดิฟฟี (Whitfield Diffie) นักวิจัยแห่งมหาวิทยาลัยสแตนฟอร์ด สหรัฐอเมริกา ในปี พ.ศ. 2518 โดยใช้หลักการกุญแจคู่ทำการเข้ารหัสและถอดรหัส ซึ่งกุญแจคู่จะประกอบด้วย กุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) หลักการทำงาน คือ ถ้าใช้กุญแจลูกใดทำการเข้ารหัสต้องใช้กุญแจอีกลูกหนึ่งทำการถอดรหัส สำหรับการเข้ารหัสและถอดรหัสด้วยกุญแจคู่จะใช้ฟังก์ชั่นทางคณิตศาสตร์เข้ามาช่วยโดยที่ฟังก์ชั่นทางคณิตศาสตร์ที่นำมาใช้จะได้รับการพิสูจน์แล้วว่ามีเฉพาะกุญแจคู่ของมันเท่านั้นที่จะสามารถถอดรหัสได้ ไม่สามารถนำกุญแจคู่อื่นมาถอดรหัสได้



รูปที่ 2.3 การเข้ารหัสลับแบบกุญแจอสมมาตร

ข้อดีของวิทยาการรหัสลับแบบกุญแจอสมมาตร

1. การจัดการกับกุญแจทำได้ง่าย เพราะว่า Mr. A ไม่ต้องจำว่าได้ใช้กุญแจคู่ไหนกับใคร, Mr. A จะใช้แค่กุญแจส่วนตัวทำการถอดรหัสข้อมูลที่ Mr. B ส่งมาให้หรือเอากุญแจส่วนตัวเข้ารหัสส่งไปให้ Mr. B, Mr. B ก็สามารถที่จะอ่านได้ ซึ่งเป็นวิธีที่ง่ายมากเพราะ Mr. A จะใช้แค่กุญแจส่วนตัวดอกเดียวสามารถที่จะติดต่อกับ Mr. B หรือใครๆก็ได้ตามต้องการ

2. การกระจายกุญแจลับ เนื่องจากกุญแจที่ใช้ในการเข้ารหัสและถอดรหัสเป็นกุญแจคนละอัน ดังนั้นจึงเป็นการง่ายต่อการจัดการกับกุญแจ เนื่องจากคุณ B เพียงแค่เปิดเผยกุญแจสาธารณะให้กับทุกคนที่ต้องการติดต่อกับ A แล้วใช้ Private Key ในการถอดรหัส

ข้อเสียของระบบเข้ารหัสแบบกุญแจอสมมาตร

1. การเข้ารหัสและถอดรหัสข้อมูลใช้เวลามาก เพราะว่าอัลกอริทึมที่ใช้มีความซับซ้อนมาก

2. ขนาดของข้อมูลหลังจากทำการเข้ารหัสแล้วจะมีการเปลี่ยนแปลงมากหรือข้อมูลหลังจากทำการเข้ารหัสแล้วจะมีขนาดใหญ่กว่าเดิมมากขึ้น ซึ่งเป็นปัญหาในการใช้งาน

2.2 มาตรฐานรหัสลับ DES (Data Encryption Standard)

มาตรฐานการเข้ารหัส DES เป็นมาตรฐานการเข้ารหัสลับของประเทศสหรัฐอเมริการ่างขึ้นในปี ค.ศ.1977 โดย NIST (National Institute of Standards and Technology) ซึ่งเป็นหน่วยงานราชการภายใต้การดูแลของกระทรวงพาณิชย์ ประเทศสหรัฐอเมริกา มีจุดประสงค์เพื่อปกป้องข้อมูลทางราชการจากการดักฟังของผู้ที่ไม่ได้รับอนุญาตหรือป้องกันข้อมูลไม่ให้มีการเปลี่ยนแปลงในระหว่างที่ส่งผ่านข้อมูลในช่องสัญญาณ หรือบรรจุในฐานข้อมูล

2.2.1 วิธีการการเข้ารหัสลับข้อมูลและถอดรหัสของ DES

การเข้ารหัสลับข้อมูล DES จะพิจารณาข้อความต้นฉบับ (Plain Text) ครั้งละ 64 บิตแล้วป้อนเข้าสู่กระบวนการเข้ารหัสลับโดยจะมีการทำงานทั้งหมด 16 รอบ ตามรูปที่ 4 และผลลัพธ์ที่ได้คือ ข้อความที่เข้ารหัสแล้ว (Cipher Text) ซึ่งมีขนาดเท่าเดิม คือ 64 บิต ส่วนขั้นตอนในการถอดรหัสลับ (Deciphering) ก็มีวิธีคล้ายคลึงกับการเข้ารหัสลับและใช้กุญแจลับขนาด 64 บิตชุดเดียวกัน แต่รายละเอียดในการนำกุญแจลับมาใช้งานจะต่างไป คือ การใช้งานจะทำตรงข้ามกัน ในแผนภาพโครงสร้างโดยรวมของการเข้ารหัสลับ DES สามารถแบ่งออกได้เป็น 2 ส่วน คือ การจัดเตรียมกุญแจ (Key Schedule) และการเข้ารหัสลับ (Decipherment) โดยการเข้ารหัสจะใช้กล่อง 2 ลักษณะ คือ กล่องสลับลำดับ (Permutation) หรือ P-box และกล่องแทนค่า (Substitution) หรือ S-box

การอธิบายรายละเอียดในแต่ละขั้นตอนจะต้องนิยามตัวแปร ดังนี้

1. ข้อความต้นฉบับ Plaintext $X= (x_1, x_2,\ldots, x_{64})$
2. ข้อมูลที่ได้จากการเข้ารหัสลับ (ข้อความไซเฟอร์) Ciphertext $Y = (y_1, y_2,\ldots, y_{64})$
3. กุญแจลับที่ใช้ Key $K= (k_1, k_2,\ldots, k_{64})$

ในการทำงานจริงจำนวนบิตของกุญแจลับจะมีเพียง 56 บิตเท่านั้น ส่วนอีก 8 บิตที่เหลือจะมีหน้าที่เป็นบิตตรวจสอบพาริตี้ (Parity Bit) ดังนั้นปริภูมิของกุญแจ (Key Space) จะมีทั้งหมด 256 รูปแบบ โดยการอธิบายขั้นตอนการเข้ารหัสลับจะอาศัยตัวอย่างประกอบ และกำหนดให้ข้อความต้นฉบับเป็น

X = Generate

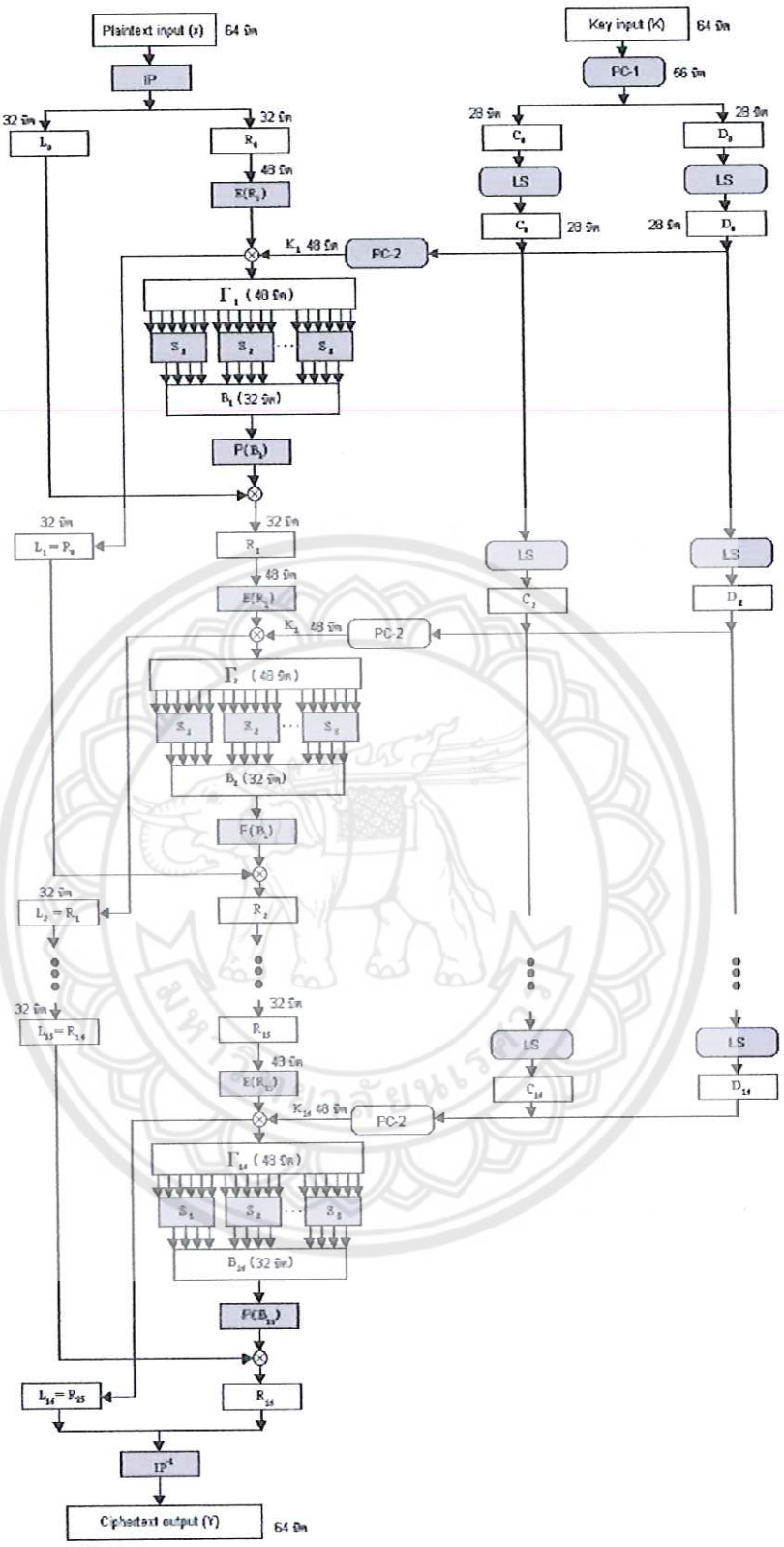
= (47 65 6E 65 72 6174 65)

และกุญแจเป็น

K = ABnormal

= (41 42 6E 6F 72 6D 61 6C)

ซึ่งทั้ง 2 ค่าแสดงในรูปของตัวเลขฐาน 16 เพื่อความกระชับ



รูปที่ 2.4 แผนภาพการเข้ารหัสลับตามมาตรฐาน DES [1]

2.2.2 การจัดเตรียมกุญแจ

เป็นขั้นตอนในการนำกุญแจที่ใช้ในการเข้ารหัสมาทำการสลับเปลี่ยนตำแหน่งบางอย่าง เพื่อใช้ในการเข้ารหัสทั้ง 16 รอบ ดังแสดงในส่วนขวามือของรูปที่ 2.4 การจัดเตรียมกุญแจสามารถสรุปเป็นขั้นตอนดังนี้

1. นำกุญแจมาเข้ากระบวนการเริ่มต้น (Initial Permutation) ผ่านกล่องสลับลำดับ PC-1 ตามตารางที่ 2.1 ซึ่งกุญแจทั้งหมดก่อนที่จะเข้ากล่องสลับลำดับจะมี 64 บิต เมื่อผ่านกล่องสลับลำดับจะเหลือเพียง 56 บิต อีก 8 บิตที่เหลือเป็นพาริตี้ และบิตทั้ง 56 บิต จะถูกแยกออกเป็น 2 ส่วนเท่าๆกัน คือ ส่วนละ 28 บิต โดยเรียกว่า C_0 และ D_0 ตามลำดับ

ตารางที่ 2.1 กล่องสลับลำดับ Permuted Choice 1 (PC-1) [1]

C_0	57	49	41	33	25	17	9	1	58	50	42	34	26	18
	10	2	59	51	43	35	27	19	11	3	60	52	44	36
D_0	63	55	47	39	31	23	15	7	62	54	46	38	30	22
	14	6	61	53	45	37	29	21	13	5	28	20	12	4

2. นำค่า C_0 และ D_0 เปลี่ยนวงจร shift register ซึ่งวงจรนี้จะทำการเลื่อนบิตไปทางซ้ายแบบวนกลับตามการเข้ารหัสในแต่ละรอบ ตามตารางที่ 2.2 โดยผลที่ได้จะเก็บในตัวแปร C_0 และ D_0 ตามลำดับ

ตารางที่ 2.2 จำนวนการเลื่อนบิตไปทางซ้ายมือแบบวนกลับสำหรับการเข้ารหัสแต่ละรอบ [1]

รอบที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
จำนวนบิต	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3. สร้างตัวแปร K_i ที่มีขนาด 48 บิต โดยนำค่า C_i และ D_i ไปผ่านกล่องสลับลำดับ PC-2 ตามตารางที่ 2.3

ตารางที่ 2.3 กล่องสลับลำดับ Permuted Choice 2 (PC-2) [1]

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

4. สร้างตัวแปร $K_2 - K_{16}$ เพื่อใช้ในการเข้ารหัสแต่ละรอบโดยทำซ้ำในข้อ 2 และ3 เรื่อยๆ

ตัวอย่าง การจัดเตรียมกุญแจโดยกำหนดให้ค่ากุญแจลับมีค่าเป็น

K = (41 42 6E 6F 72 6D 61 6C)

สามารถเขียนในรูปตัวเลขฐานสองได้เป็น

K = (0100 0001 0100 0010 0110 1110 0110 1111

0111 0010 0110 1101 0110 0001 0110 1100)

ขั้นตอนแรก คือ หาค่าของ C_0 และ D_0 โดยจะได้จากการเลือกบิตจากกุญแจลับ K สำหรับตำแหน่งที่เลือกจะขึ้นอยู่กับกล่องสลับลำดับ PC-1 ในตารางที่ 2.1 จะได้

C_0 = (0000 0000 1111 1111 1111 1100 0001)

D_0 = (0001 1110 1010 1100 1010 1100 0000)

จากนั้นจะใช้ตารางที่ 2.2 เพื่อพิจารณาหาค่า C_1, D_1 ต่อไปโดยรอบแรกจะเลื่อนตำแหน่งบิตไปทางซ้ายจำนวน 1 ตำแหน่งตามที่ระบุในตารางที่ 2.2 (รอบที่ 1 เลื่อน 1บิต) จะได้

C_1 = (0000 0001 1111 1111 1111 1000 0010)

D_1 = (0011 1101 0101 1001 0101 1000 0000)

หาค่า C_2, D_2 ทำได้โดยนำค่า C_1, D_1 ไปผ่านวงจร LS โดยเลื่อนบิตไปทางซ้ายแบบวนกลับ 1 ตำแหน่ง ตามที่ระบุในตารางที่ 2.2 (รอบที่ 2 เลื่อน 1 บิต) จะได้

C_2 = (0000 0011 1111 1111 1111 0000 0100)

D_2 = (0111 1010 1011 0010 1011 0000 0000)

สำหรับค่า (C_3, D_3), (C_4, D_4), ...และ (C_{16}, D_{16}) ก็มีลักษณะการคำนวณเหมือนกัน โดยเลื่อนบิตเปรียบเทียบกับตารางที่ 2.2 หลังจากที่คำนวณค่าของ C และ D ครบทั้ง 16 ชุดแล้วจะนำไปใช้ในการหาค่า K_1 ถึง K_{16} โดยป้อนค่า C และ D แต่ละชุดเข้าไปในวงจรสลับลำดับ PC-2 ตามตารางที่ 2.3 แล้วจะได้ชุดกุญแจ K_1 - K_{16} ที่ใช้ประกอบในการเข้ารหัสแต่ละรอบ เช่น K_1 หาได้จาก C_1 และ D_1 ซึ่งเมื่อพิจารณาจากตารางที่ 2.3 แล้วจะได้

K_1 = (111000 001011 011001 101110 101000 010011 110010 100101)

และ K_2 หาได้จาก C_2 และ D_2 โดยอาศัยจากตารางที่ 2.3 จะได้

K_2 = (111000 001001 011011 110110 001110 110111 001000 000001)

เมื่อได้ค่า K_1 ถึง K_{16} ครบทุกค่าแล้วจะนำค่าเหล่านี้ไปใช้ประกอบการเข้ารหัสลับข้อมูลต้นฉบับในแต่ละรอบตามแผนภาพทางด้านซ้ายมือของรูปที่ 2.4 ซึ่งผลที่ได้หลังจากทำครบทั้ง 16 รอบแล้วจะได้เป็นข้อความไซเฟอร์ขนาด 64 บิตตามต้องการ

2.2.3 การเข้ารหัสแต่ละรอบ (Encipherment)

เป็นกระบวนการนำข้อความต้นฉบับ (Plaintext) ขนาด 64 บิตไปผ่านการเข้ารหัสโดยใช้ชุดกุญแจ K_1 - K_{16} ที่เตรียมไว้ตอนแรกดังแสดงในส่วนซ้ายมือของรูปที่ 2.4 โดยมีขั้นตอนสรุปได้ดังนี้

1. สมมติว่าข้อความต้นฉบับมีค่าเป็น

X = Generate

= (47 65 6E 65 72 6174 65)

หรือเขียนในรูปตัวเลขฐานสองได้เป็น

X = (0100 0111 0110 0101 0110 1110 0110 0101

0111 0010 0110 0001 0111 0100 0110 0101)

2. ป้อนข้อความต้นฉบับเข้าสู่กล่องสลับลำดับ Initial Permutation ตามตารางที่ 2.4 ข้อความจะถูกแยกบิตออกเป็น 2 บล็อกเท่าๆกัน คือ L_0 และ R_0 โดยแต่ละบล็อกจะมี 32 บิต

L_0 = (1111 1111 0101 0000 1100 1111 1010 1011)

F F 5 0 C F A B

R_0 = (0000 0000 1111 1110 0000 0100 0001 0101)

0 0 F E 0 4 1 5

ตารางที่ 2.4 กล่องสลับลำดับ Initial Permutation (IP) [1]

L_0	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
R_0	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

3. เมื่อพิจารณาตามรูปที่ 2.4 จะพบว่า L_0 ไม่มีการประมวลผล แต่ R_0 จะถูกนำไปผ่านกระบวนการหลายกระบวนการ และนำ K_1 มาใช้ในการเข้ารหัสด้วย ดังนั้น R_0 ที่มีขนาด32 บิตจะถูกเพิ่มเป็น 42 บิต โดยใช้ฟังก์ชั่น $E(R_0)$ ตามตารางที่ 2.5 แล้วจะได้

$E(R_0)$ = (100000 000001 011111 111100 000000 001000 000010 101010)

ตารางที่ 2.5 E Bit-Selection Table [1]

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

4. นำค่า $E(R_0)$ ไปบวกแบบ modulo กับกุญแจ K_1 ซึ่งมีขนาด 48 บิตเท่ากัน ผลลัพธ์ที่ได้จะเก็บในตัวแปร Γ หรือเรียกว่า The key-dependent function ตามสมการ

$$\Gamma_1 = E(R_0) \otimes K_1$$

$$= (011000\ \ 001010\ \ 000010\ \ 010010\ \ 101000\ \ 011011\ \ 110000\ \ 001111)$$

5. ในการคำนวณค่าของ Γ_j ในรอบอื่นๆสามารถหาจาก

$$\Gamma_j = E(R_i) \otimes K_j \qquad , 0 \le i \le 15 \qquad , 1 \le j \le 16$$

6. เมื่อได้ค่า Γ_1 ขนาด 48 บิตแล้ว ให้แบ่งออกเป็น 8 กลุ่ม แต่ละกลุ่มจะมีขนาด 6 บิต เพื่อนำทั้ง 8 กลุ่มไปป้อนให้กับกล่องแทนค่า S-Box ที่มีทั้งหมด 8 ชุด ซึ่งกล่องแทนค่าแต่ละชุดจะให้ผลลัพธ์เป็นบิตที่มีขนาดลดลงเหลือ 4 บิต ดังนั้นจำนวนบิตจะลดลงเหลือแค่ 36 บิต

การทำงานของ S-Box คือ การนำบิตแรกและบิตสุดท้ายมาใช้ระบุหมายเลขแถวของตาราง S-Box ตามตารางที่ 2.6 และใช้ค่าของบิตที่ 2 ถึงบิตที่ 5 ระบุหมายเลขคอลัมน์ของตาราง S-Box ผลลัพธ์ที่ได้ คือ ค่าที่บรรจุอยู่ในตาราง ณ ตำแหน่งที่ระบุ โดยตัวเลขที่ได้จะมีขนาดอยู่ระหว่าง 0 – 15 เมื่อเขียนเป็นตัวเลขฐานสองแล้วจะมีขนาด 4 บิตพอดี

ตารางที่ 2.6 Primitive S-Box Function [1]

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

ตารางที่ 2.6 Primitive S-Box Function (ต่อ) [1]

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

ตารางที่ 2.6 Primitive S-Box Function (ต่อ) [1]

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

นำค่า Γ_1 มาคำนวณหา S_1 ถึง S_8 จากตารางที่ 2.6 ซึ่งค่าแต่ละค่ามาจากกล่องแทนค่า S-Box แต่ละชุด ได้ผลดังนี้

$S_1(00, 1100) = S_1(1, 12) = 5 = 0101$

$S_2(00, 0101) = S_2(0, 5) = 11 = 1011$

$S_3(00, 0001) = S_3(0, 1) = 0 = 0000$

$S_4(00, 1001) = S_4(0, 9) = 2 = 0010$

$S_5(10, 0100) = S_5(2, 4) = 10 = 1010$

$S_6(01, 1101) = S_6(1, 13) = 11 = 1011$

$S_7(10, 1000) = S_7(2, 8) = 11 = 1011$

$S_8(01, 0111) = S_8(1, 7) = 4 = 0100$

7. นำค่าที่ได้จาก S-Box แต่ละชุดมารวมกันจะได้ผลลัพธ์เป็นค่าของ B_1 เท่ากับ

$B_1 = (0101\ \ 1011\ \ 0000\ \ 0010\ \ 1010\ \ 1011\ \ 1010\ \ 0100)$

8. นำค่า B_1 ไปผ่านกระบวนการ $P(B_1)$ ซึ่งเป็นกล่องสลับตำแหน่ง (Permutation Function) โดยจะมีการสลับตำแหน่งแตกต่างจากเดิม ตามในตารางที่ 2.7 จะได้

$P(B_1) = (0101\ \ 0001\ \ 0110\ \ 1000\ \ 1110\ \ 0100\ \ 1010\ \ 0011)$

ตารางที่ 2.7 Permutation Function P [1]

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

9. หาค่า R_1 จาก $P(B_1)$ และ L_0 ตามสมการ

$$R_1 = P(B_1) \otimes L_0$$

$$= (1010\ \ 1110\ \ 0011\ \ 1000\ \ 0010\ \ 1011\ \ 0000\ \ 1000)$$

$$A\quad E\quad 3\quad 8\quad 2\quad B\quad 0\quad 8$$

10. จากแผนภาพในรูปที่ 2.4 จะเห็นว่า R_1 เป็นผลลัพธ์ของกระบวนการเข้ารหัสลับของชุดบิตทางด้านขวามือในการทำงานรอบที่ 1 และ L_1 เป็นผลลัพธ์ของกระบวนการเข้ารหัสลับของชุดบิตทางด้านซ้ายมือกลับมีความซับซ้อนน้อยกว่ามาก ดังนั้นให้นำค่าของ R_0 มาใช้เป็น L_1 ได้เลย จะได้

$$L_1 = R_0 = (0000\ \ 0000\ \ 1111\ \ 1110\ \ 0000\ \ 0100\ \ 0001\ \ 0101)$$

$$0\quad 0\quad F\quad E\quad 0\quad 4\quad 1\quad 5$$

11. รายละเอียดขั้นตอนการทำงานที่ได้อธิบายมาเป็นการเข้ารหัสลับในรอบที่ 1 สำหรับการเข้ารหัสที่เหลืออีก 15 รอบมีรูปแบบการทำงานเหมือนเดิม แต่ชุดกุญแจที่ใช้ในแต่ละรอบจะแตกต่างกัน ผลที่ได้ทั้ง 16 ขั้นตอน คือ L_{16} และ R_{16} โดยจะนำไปป้อนเข้าสู่กล่องสลับลำดับผกผัน IP^{-1} ตามตารางที่ 2.8 เพื่อให้ได้เป็นข้อความไซเฟอร์ Y ตามที่ต้องการ

ตารางที่ 2.8 กล่องสลับลำดับผกผัน (Inverse of Initial Permutation IP^{-1}) [1]

R	40	8	48	16	56	24	64	32
	39	7	47	15	55	23	63	31
	38	6	46	14	54	22	62	30
	37	5	45	13	53	21	61	29
L	36	4	44	12	52	20	60	28
	35	3	43	11	51	19	59	27
	34	2	42	10	50	18	58	26
	33	1	41	9	49	17	57	25

ดังนั้นจะได้

Y = (1110 1001 0101 1100 1010 1111 0101 1100
1101 0010 1111 1100 1011 0111 0010 1111)
= (E9 5C AF 5C D2 FC B7 2F)
= é \ - \ ò ü · /

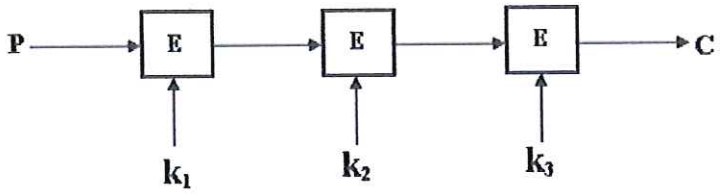
ทั้งนี้เพื่อให้เข้าใจในกระบวนการทำงานของมาตรฐาน DES มากยิ่งขึ้นสามารถศึกษารายละเอียดเพิ่มเติมได้ในภาคผนวก ก ซึ่งจะแสดงตัวอย่างในแต่ละขั้นตอนการทำงานของอัลกอริทึมอย่างละเอียด

2.2.4 อัลกอริทึม Triple DES หรือการใช้ DES 3 ครั้ง

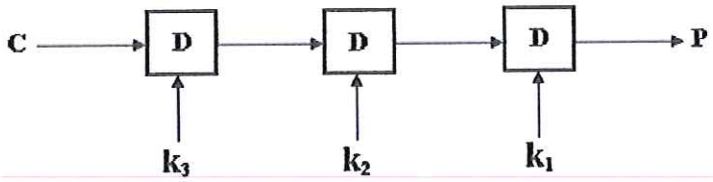
ถึงแม้ว่าการโจมตีแบบพบกันครึ่งทางยังยากที่จะเป็นไปได้ในทางปฏิบัติ แต่ก็มีผู้ที่ต้องการอัลกอริทึมที่ปลอดภัยมากกว่าการใช้ DES 2 ครั้ง ดังนั้น จึงมีการพัฒนาวิธีการเข้ารหัสโดยใช้ DES 3 ครั้ง ซึ่งสามารถแยกได้เป็น 2 วิธี คือ การเข้ารหัสโดยใช้ DES 3 ครั้งด้วยกุญแจ 3 ค่า และการเข้ารหัสโดยใช้ DES 3 ครั้ง ด้วยกุญแจ 2 ค่า

2.2.4.1 การเข้ารหัสลับโดยใช้ DES 3 ครั้ง

วิธีนี้มีวิธีการที่คล้ายคลึงกับการเข้ารหัสโดยใช้ DES 2 ครั้ง กล่าวคือ นำข้อความต้นฉบับ P ผ่านอัลกอริทึมเข้ารหัส DES 3 ครั้ง โดยแต่ละครั้งใช้ค่ากุญแจที่ต่างกันคือ k_1, k_2, k_3 ดังแสดงในรูปที่ 2.5(ก)



(ก) การเข้ารหัสลับ



(ข) การถอดรหัสลับ

รูปที่ 2.5 แสดงการเข้ารหัสโดยใช้ DES 3 ครั้งด้วยกุญแจ 3 ค่า [1]

เราสามารถแสดงนิยามทางคณิตศาสตร์ได้ว่า

$$C = E_{k_3}[E_{k_2}[E_{k_1}[P]]]$$

ส่วนการถอดรหัส ก็กระทำย้อยกลับ ดังแสดงในรูปที่ 1.1(ข) กล่าวคือ **ถอดรหัสด้วยค่ากุญแจ** k_3 ก่อน แล้วตามด้วย k_2 และ k_1 ตามลำดับ ซึ่งเราสามารถแสดงนิยามทางคณิตศาสตร์ได้ว่า

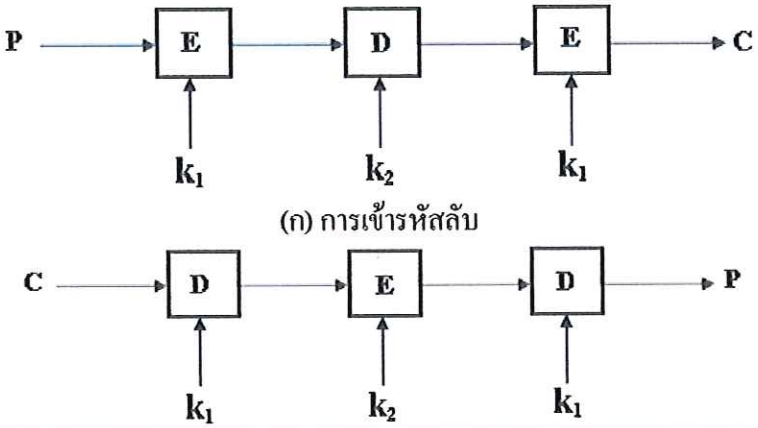
$$P = D_{k_1}[D_{k_2}[D_{k_3}[C]]]$$

2.2.4.2 การเข้ารหัสลับโดยใช้ DES 3 ครั้งด้วยกุญแจ 2 ค่า

การใช้ DES 3 ครั้งด้วยกุญแจ 2 ค่านี้จัดว่าเป็นการเข้ารหัสที่ปลอดภัยมาก**ที่สุดวิธีหนึ่งในปัจจุบัน** เนื่องจากต้องใช้กุญแจ 3 ค่า ดังนั้น ขนาดของกุญแจคือ 56*3 = 168 บิต **ซึ่งทำให้การโจมตีแบบตะลุยโดยการค้นหาค่ากุญแจทั้งหมดไม่สามารถทำได้ในเทคโนโลยีปัจจุบัน แต่การใช้กุญแจถึง 3 ค่าในการเข้ารหัสและการถอดรหัสนี้ ทำให้เกิดความไม่สะดวกในการใช้งาน จึงมีผู้เสนอรูปแบบการเข้ารหัสโดยใช้ ด้วย DES 3 ครั้งแต่ใช้กุญแจเพียง 2 ค่าแทน ซึ่งทำให้ขนาดของกุญแจเท่ากับ** 56*2 = 112 บิต ซึ่งจำนวนบิตเท่ากับการใช้ DES 2 ครั้ง แต่มีความปลอดภัยสูงกว่า

การเข้ารหัส DES 3 ครั้งด้วยกุญแจ 2 ค่านี้ มีขั้นตอนดังนี้

1. นำข้อความต้นฉบับ P มาเข้ารหัส DES 2 ครั้งแรกโดยใช้กุญแจ k_1 คือ ทำ $E_{k1}[P]$
2. นำข้อความเริ่มต้นที่ได้จากขั้นแรก ไปถอดรหัสด้วยกุญแจ k_2 ทำให้ได้ $D_{k2}[E_{k1}[P]]$
3. นำข้อความเริ่มต้นที่ได้จากขั้นที่สอง เข้ารหัสด้วยกุญแจ k_1 เป็นครั้งที่สาม ทำให้ได้ข้อความเริ่มต้น C ดังแสดงในรูปที่ 2.6 (ก)



(ก) การเข้ารหัสลับ

(ข) การถอดรหัสลับ

รูปที่ 2.6 การเข้ารหัสโดยใช้ DES 3 ครั้ง ด้วยกุญแจ 2 ค่า [1]

ดังนั้น นิยามทางคณิตศาสตร์ของการเข้ารหัส DES 3 ครั้งด้วยกุญแจ 2 ค่า

$$C = E_{k_1}[D_{k_2}[E_{k_1}[P]]]$$

การถอดรหัสของ DES 3 ครั้ง ด้วยกุญแจ 2 ค่า ก็ทำย้อนกลับนั่นเอง กล่าวคือ

1. ข้อความเริ่มต้น C มาถอดรหัสครั้งแรกด้วยกุญแจ k_1 กล่าวคือ ทำ $D_{k_1}[C]$
2. นำผลลัพธ์ที่ได้จากขั้นแรก มาเข้ารหัสด้วยกุญแจ k_2 ทำให้ได้ $E_{k_2}[D_{k_1}[C]]$
3. นำผลลัพธ์ที่ได้ในขั้นที่สอง มาผ่านการถอดรหัสลับโดยกุญแจ k_1 อีกครั้ง ทำให้ได้ข้อความต้นฉบับ P ออกมาตามเดิม ดังแสดงในรูปที่ 2.6 (ข)

ดังนั้น นิยามทางคณิตศาสตร์ของการถอดรหัส DES 3 ครั้งด้วยกุญแจ 2 ค่า คือ

$$P = D_{k_1}[E_{k_2}[D_{k_1}[C]]]$$

การถอดรหัสลับโดยใช้ DES 3 ครั้งด้วยกุญแจ 2 ค่า แบบนี้เรียกว่า **การทำงานแบบเข้ารหัสลับ-ถอดรหัสลับ-เข้ารหัสลับ** (Encrypt-decrypt-encrypt) หรือเรียกสั้นๆว่า EDE ส่วนการเข้ารหัสโดยใช้ DES 3 ครั้งด้วยกุญแจ 3 ค่า เรียกว่า การทำงานแบบเข้ารหัสลับ-เข้ารหัสลับ-เข้ารหัสลับ (Encrypt-encrypt-encrypt) หรือเรียกสั้นๆ ว่า EEE จุดประสงค์ของการทำงานแบบ EDE คือ ระบบนี้สามารถนำไปใช้ได้กับ DES แบบดั้งเดิมโดยใช้กุญแจค่าเดียว เพราะถ้าเรากำหนดว่ากุญแจ $k_1 = k_2$ ในนิยามทางคณิตศาสตร์ เราจะได้

$$C = E_{k_1}[D_{k_1}[E_{k_1}[P]]] = E_{k_1}[P]$$

ซึ่ง $C = E_{k1}[P]$ คือการเข้ารหัสแบบปกตินั่นเอง ดังนั้น การทำงานแบบ EDE จึงมีจุดประสงค์เพื่อสามารถนำไปใช้งานกับระบบ DES แบบเดิมที่มีอยู่แล้ว ทำให้การเข้ารหัสโดยวิธีนี้จึงได้รับความนิยมใช้งานอย่างแพร่หลายในปัจจุบัน

2.3 มาตรฐานรหัสลับ AES (Advance Encryption Standard)

วิธีการเข้ารหัสลับ DES ที่ได้กล่าวถึงในหัวข้อ 2.2 นั้นได้ถูกประกาศใช้เป็นมาตรฐานสำหรับการเข้ารหัสลับข้อมูลตั้งแต่ปี ค.ศ.1977 ในระยะเวลาที่ผ่านมามาตรฐาน DES ได้รับความนิยมอย่างแพร่หลาย เพราะสามารถประยุกต์ใช้งานในทางปฏิบัติได้ดีและมีประสิทธิภาพ แต่พื้นฐานในการเข้ารหัสลับ DES ใช้กุญแจในการเข้ารหัสเพียง 56 บิต และบิตพาริตี้อีก 8 บิต รวมเป็น 64 บิต ในขณะที่สมรรถนะของอุปกรณ์คอมพิวเตอร์มีแนวโน้มที่จะเพิ่มประสิทธิภาพให้สูงขึ้นเรื่อยๆ ทั้งในส่วนของหน่วยประมวลผลกลางและขนาดของหน่วยความจำทำให้การคำนวณสมการคณิตศาสตร์ที่มีความซับซ้อนมากๆสามารถทำได้ในเวลาที่รวดเร็ว และความก้าวหน้าในการพัฒนาอุปกรณ์ประมวลผลที่เพิ่มขึ้นอย่างรวดเร็วทำให้เกิดความกังวลถึงประสิทธิภาพในการปกป้องข้อมูลของมาตรฐานการเข้ารหัสลับ DES ซึ่งใช้กุญแจที่มีขนาดไม่ใหญ่ ดังนั้นจึงต้องมีการปรับเปลี่ยนและพัฒนากระบวนการการเข้ารหัสลับข้อมูลขึ้นมาใหม่ เพื่อให้มีประสิทธิภาพในการปกป้องข้อมูลให้สอดคล้องกับประสิทธิภาพของคอมพิวเตอร์ในยุคปัจจุบันและอนาคตมากยิ่งขึ้น

ในปี ค.ศ.2001 ได้มีการพัฒนามาตรฐานการเข้ารหัสลับข้อมูลแบบใหม่ขึ้น เรียกว่า Advance Encryption Standard (AES) กระบวนการเข้ารหัสลับ AES นี้เป็นกระบวนการที่ทำการเข้ารหัสข้อมูลโดยใช้ขนาดของกุญแจ 3 แบบ คือ 128, 192 และ 256 บิต โดยทั่วไปจะเรียกอัลกอริทึมเหล่านี้ว่า AES-128, AES-192 และ AES-256 ตามลำดับ ซึ่งมาตรฐานนี้มีความสามารถในการปกป้องและให้ความปลอดภัยแก่ข้อมูลได้อย่างมีประสิทธิภาพและสอดคล้องกับสมรรถนะของคอมพิวเตอร์ที่มีอยู่ในปัจจุบันได้เป็นอย่างดี

การเข้ารหัสลับ AES มีโครงสร้างการทำงานโดยรวมคล้ายกับมาตรฐาน DES คือ ได้แบ่งกระบวนการทำงานออกเป็น 2 ส่วน คือ ส่วนของข้อมูลต้นฉบับที่จะเข้ารหัสลับ (plaintext) และส่วนของกุญแจที่ใช้ในการเข้ารหัสลับ (key) ในการเข้ารหัสลับจะพิจารณาข้อความต้นฉบับทีละ 128 บิต (1 บล็อก) กระบวนการเข้ารหัสลับ (enciphered) จะทำงานทีละรอบเหมือนกับมาตรฐาน DES โดยจำนวนรอบของการเข้ารหัสลับ AES จะขึ้นอยู่กับจำนวนบิตของกุญแจที่ใช้ทำการเข้ารหัสลับ ตามมาตรฐานของ AES ได้กำหนดว่า กุญแจขนาด 128 บิตจะมีการวนรอบเข้ารหัสลับทั้งหมด 10 รอบ, กุญแจขนาด 192 บิต จะมีการวนรอบเข้ารหัสลับทั้งหมด 12 รอบ และกุญแจขนาด 256 บิต จะมีการวนรอบเข้ารหัสลับทั้งหมด 14 รอบ

2.3.1 การเข้ารหัสลับ AES

การเข้ารหัสลับตามมาตรฐาน AES จะพิจารณาชุดของข้อมูลครั้งละ 128 (1 บล็อก) โดยเก็บอยู่ในตัวแปร state ที่มีโครงสร้างการจัดเรียงเป็นอาร์เรย์ของไบต์ 2 มิติ ดังรูป

State array

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

รูปที่ 2.7 การจัดเรียงไบต์ในตัวแปร State [1]

จากรูปจะเห็นว่าตัวแปร state ประกอบด้วยไบต์จำนวน 4 แถวเสมอ ในแต่ละแถวจะมีอยู่ Nb ไบต์ การคำนวณ Nb คำนวณได้จากการนำขนาดของบล็อกไปหารด้วย 32 แต่ในมาตรฐาน AES ได้กำหนดให้ Nb = 4 เท่านั้น สำหรับไบต์แต่ละตัวในอาร์เรย์จะเขียนแทนด้วย s โดยจะมีดรรชนีเป็นตัวห้อยอยู่ 2 ตัว เพื่อใช้ระบุตำแหน่งของไบต์ในอาร์เรย์

ดรรชนีตัวที่ 1 r หมายถึง หมายเลขแถว r ที่มีค่าอยู่ระหว่าง $0 \leq r \leq 4$

ดรรชนีตัวที่ 2 c หมายถึง คอลัมน์ มีค่าอยู่ระหว่าง $0 \leq c \leq Nb$

ดังนั้น ค่าของแต่ละไบต์ในอาร์เรย์จะอ้างอิงในรูปของ $s_{r,c}$ หรือ s[r,c]

อัลกอริทึม AES กำหนดให้การเข้ารหัสและถอดรหัสลับมีการทำงานเป็นรอบๆ โดยในแต่ละรอบจะมีการแปลงข้อมูลในระดับของไบต์ 4 ขั้นตอน ได้แก่

1. SubBytes เป็นกระบวนการแทนที่ไบต์โดยใช้ตารางแทนที่ (S-Box)
2. ShiftRows การเลื่อนไบต์ในแนวแถวของอาร์เรย์ State ด้วยออฟเซตที่แตกต่างกันไปในแต่ละแถว
3. MixColumns ผสมผสานข้อมูลภายในคอลัมน์แต่ละคอลัมน์ของอาร์เรย์ State
4. AddRoundKey บวกค่ากุญแจในแต่ละรอบกับอาร์เรย์ State

ไซเฟอร์สามารถเขียนเป็นภาพรวมในรูปของ Pseudo Code ดังรูปที่ 2.8

```
Cipher (byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)] )
begin
     byte state[4,Nb]

     state = in

     AddRoundKey (State, w[0, Nb-1] )

     for round = 1 step 1 to Nr-1

SubBytes (State)
ShiftRows  (State)
MixColumns  (State)
AddRoundKey  (State, w [round*Nb] , (round+1)*Nb-1 ])

end for

SubBytes (State)
ShiftRows  (State)
AddRoundKey  (State, w[Nr*Nb] , (Nr+1)*Nb-1 ] )
          out = state
end
```

รูปที่ 2.8 ภาพรวมของการเข้ารหัสลับ AES ที่แสดงโดยอาศัย pseudo code [1]

จากรูปจะเห็นว่าข้อมูลที่ต้องการเข้ารหัสลับ (in) ถูกป้อนเข้าสู่ตัวแปร State ในขั้นแรกตัวแปร State จะถูกนำไปผ่านกระบวนการ AddRoundKey ก่อนที่จะเข้าสู่กระบวนการวนลูป โดยการวนลูปแต่ละครั้งประกอบด้วยกระบวนการทั้งหมด 4 ขั้นตอน คือ SubBytes, ShiftRows, MixColumns และ AddRoundKey อีกอย่างละครั้ง โดยการทำงานรอบที่ Nr จะแตกต่างไปจากรอบอื่นๆ คือ ไม่มีการทำ MixColumn

2.3.1.1 การแปลง SubBytes

ฟังก์ชั่นการแปลง SubBytes() เป็นกระบวนการแทนที่ไบต์แต่ละไบต์ในอาร์เรย์ State โดยใช้ตารางการแทนที่ (S-Box) ในตารางที่ 2.9 โดยการแทนที่ในแต่ละไบต์จะทำอย่างอิสระกัน เช่น ไบต์ที่เข้ามาใน S-Box คือ 00000010 (02_{hex}) เมื่อเทียบจากตาราง S-Box จะได้ 01100110 (77_{hex})

ตารางที่ 2.9 ตารางการแทนที่ (S-Box) สำหรับใช้ในฟังก์ชันการแปลง SubBytes() แสดงในรูปแบบตัวเลขฐาน 16 [1]

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

รายละเอียดของขั้นตอน SubBytes ประกอบด้วยขั้นตอนย่อย 2 ขั้นตอน คือ

1. การหาตัวผกผันการคูณ (multiplicative inverse) ในฟิลด์จำกัด GF(2^8)
2. การแปลงแอฟไฟน์ (affine transform)

การหาอินเวิร์สการคูณของไบต์มีวิธีตามเงื่อนไข ดังนี้

(ข้อมูลไบต์ที่แสดงในรูปพหุนาม)(พหุนามที่เป็นอินเวิร์ส) = 1

เช่น ไบต์ 00000010 เขียนในรูปพหุนามได้เป็น X มีค่าตัวผกผันการคูณ คือ 10001101 หรือเขียนในรูปของพหุนามได้เป็น $x^7 + x^3 + x^2 + 1$ จะได้

$$(X)(x^7 + x^3 + x^2 + 1) = x^8 + x^4 + x^3 + x$$

หลังจากนั้นนำไปทำ modulo กับพหุนามไม่ลดรูป $m(x) = x^8 + x^4 + x^3 + x + 1$ เพื่อให้ได้พหุนามที่มีกำลังไม่เกิน 7 จะได้

$$x^8 + x^4 + x^3 + x \text{ modulo } (x^8 + x^4 + x^3 + x + 1) = 1$$

สังเกตไบต์ที่มีค่าเป็น 00000001 จะมีตัวผกผันการคูณเป็นตัวของมันเอง ส่วนไบต์ที่มีค่าเป็น 00000000 เป็นกรณียกเว้น เนื่องจากไม่สามารถหาพหุนามที่มีคุณสมบัติตรงตามเงื่อนไขที่กำหนดในมาตรฐานได้จึงให้มีค่าตัวผกผันการคูณเป็น 0000000 สำหรับไบต์อื่นๆที่เหลืออีก 254 ตัว จะมีคู่ตัวผกผันที่ไม่ซ้ำตัวเอง b0'

การแปลงแอฟไฟน์ สามารถเขียนในรูปสมการได้ดังนี้

$$b_0' = b_i \otimes b_{(i+4) \bmod 8} \otimes b_{(i+5) \bmod 8} \otimes b_{(i+6) \bmod 8} \otimes b_{(i+7) \bmod 8} \otimes C_i \text{ โดย } 0 \leq i < 8$$

ค่า b_i คือ ตำแหน่งของบิตที่ i ของไบต์ที่ต้องการแปลงแอฟไฟน์

ค่า c_i คือ ตำแหน่งของบิตที่ i ของไบต์ c ซึ่งเป็นค่าคงที่เท่ากับ {63} หรือ {01100011} สามารถแยกรายละเอียดการแปลงแอฟไฟน์สำหรับแต่ละบิตได้ดังนี้

$$\begin{aligned}
b_0' &= b_0 \otimes b_4 \otimes b_5 \otimes b_6 \otimes b_7 \otimes 1 \\
b_1' &= b_0 \otimes b_1 \otimes b_5 \otimes b_6 \otimes b_7 \otimes 1 \\
b_2' &= b_0 \otimes b_1 \otimes b_2 \otimes b_6 \otimes b_7 \otimes 0 \\
b_3' &= b_0 \otimes b_1 \otimes b_2 \otimes b_3 \otimes b_7 \otimes 0 \\
b_4' &= b_0 \otimes b_1 \otimes b_2 \otimes b_3 \otimes b_4 \otimes 0 \\
b_5' &= b_1 \otimes b_2 \otimes b_3 \otimes b_4 \otimes b_5 \otimes 1 \\
b_6' &= b_2 \otimes b_3 \otimes b_4 \otimes b_5 \otimes b_6 \otimes 1 \\
b_7' &= b_3 \otimes b_4 \otimes b_5 \otimes b_6 \otimes b_7 \otimes 1
\end{aligned}$$

โดยทั่วไปจะเขียนความสัมพันธ์ข้างต้นในรูปของเมทริกซ์เพื่อความกระชับ

$$\begin{pmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

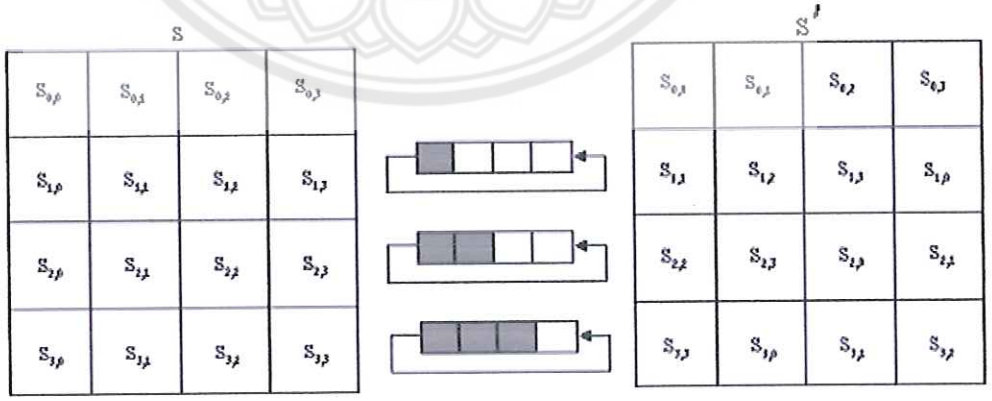
เช่น หากนำพหุนามตัวผกผัน คือ $(x^7 + x^3 + x^2 + 1)$ หรือ 100011011 ไปทำการแปลงจะได้

$$b'_0 = 1 \otimes 0 \otimes 0 \otimes 0 \otimes 1 \otimes 1 = 1$$
$$b'_1 = 1 \otimes 0 \otimes 0 \otimes 0 \otimes 1 \otimes 1 = 1$$
$$b'_2 = 1 \otimes 0 \otimes 1 \otimes 0 \otimes 1 \otimes 0 = 1$$
$$b'_3 = 1 \otimes 0 \otimes 1 \otimes 1 \otimes 1 \otimes 0 = 0$$
$$b'_4 = 1 \otimes 0 \otimes 1 \otimes 1 \otimes 0 \otimes 0 = 1$$
$$b'_5 = 0 \otimes 1 \otimes 1 \otimes 0 \otimes 0 \otimes 1 = 1$$
$$b'_6 = 1 \otimes 1 \otimes 0 \otimes 0 \otimes 0 \otimes 1 = 1$$
$$b'_7 = 1 \otimes 0 \otimes 0 \otimes 0 \otimes 1 \otimes 0 = 0$$

ผลที่ได้จะเป็นไบต์มีค่าเท่ากับ 01110111 หากเทียบผลที่ได้จากการนำไบต์ 00000010 ไปผ่าน 2 ขั้นตอนย่อย คือ การหาตัวผกผันการคูณ และการแปลงแอฟไฟน์ กับการอ่านผลลัพธ์โดยตรงจากตารางฟังก์ชันการแปลง SubBytes() ในตารางที่ 2.2 จะพบว่าผลลัพธ์ที่ได้มีค่าตรงกัน คือ {77}

2.3.1.2 การแปลงด้วยวิธีการเลื่อนแถว

เมื่อนำค่าตัวแปร state ผ่านการแปลง SubBytes() แล้ว จะนำผลที่ได้ไปผ่านการแปลงด้วยกระบวนการต่อไป คือ วิธีการเลื่อนแถว ShiftRows () ซึ่งขั้นตอนนี้เป็นกระบวนการที่ง่าย คือ ให้เลื่อนไบต์แต่ละแถวไปทางซ้ายมือด้วยออฟเซตที่แตกต่างกัน ไบต์ในแถวแรกให้อยู่ในตำแหน่งเดิม ไบต์ในแถวที่ 2 ให้เลื่อนไปหนึ่งตำแหน่ง ไบต์ในแถวที่ 3 ให้เลื่อน 2 ตำแหน่ง และไบต์ในแถวที่ 4 ให้เลื่อน 3 ตำแหน่ง โดยการเลื่อนจะทำในแบบวนกลับ คือ เมื่อเลื่อนตำแหน่งของไบต์ในแต่ละครั้ง ไบต์ที่อยู่ซ้ายมือสุดจะวนกลับไปอยู่ตำแหน่งขวามือสุด ดังรูปที่ 2.10



รูปที่ 2.9 แผนภาพการแปลงด้วยวิธีการเลื่อนแถวของฟังก์ชัน ShiftRows () [1]

สามารถเขียนการแปลง ShiftRows () ในรูปของสมการความสัมพันธ์ได้เป็น

$$S'_{r,C} = S'_{r,(C,shift(r,Nb)) \bmod Nb} \text{ สำหรับ } 0 < r < 4 \text{ และ } 0 \leq c < Nb$$

ค่าของ Shift (r,Nb) จะขึ้นอยู่กับหมายเลขแถว r คือ

shift (1,4) = 1; shift (2,4) = 2; Shift (3,4) = 3

2.3.1.3 การแปลงด้วยวิธีผสมผสานคอลัมน์

กระบวนการแปลงด้วยวิธีผสมผสานหรือ MixColumns() จะพิจารณาการแปลงข้อมูลในตัวแปร State ทีละคอลัมน์ โดยข้อมูลแต่ละคอลัมน์จะประกอบด้วยข้อมูลจำนวน 4 ไบต์ที่สามารถพิจารณาว่าเป็นพหุนามบนฟิลด์ GF(2^8) มีทั้งสิ้น 4 พจน์ คือ มีสัมประสิทธิ์จำนวน 4 ตัว นำพหุนามคูณด้วยพหุนาม a(x) ที่มีค่าคงที่เท่ากับ

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

การคูณพหุนามสามารถเขียนในรูปของเมทริกซ์ได้ดังนี้

$$S'(x) = a(x) \otimes S(x)$$

นั่นคือ

$$\begin{pmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{pmatrix}$$

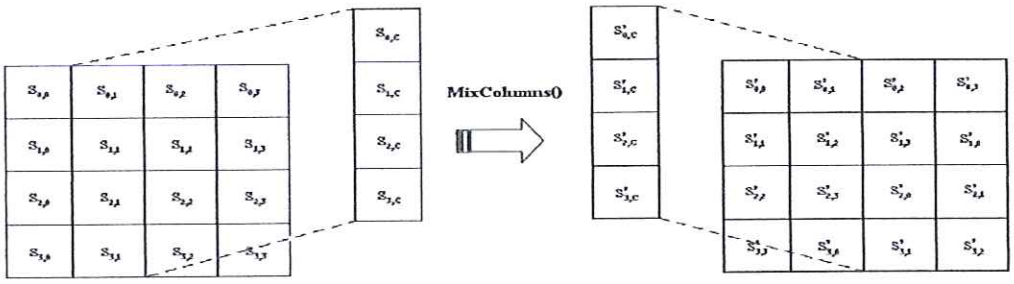
โดยที่ $0 \leq c < Nb$ และ Nb=4 ดังนั้นไบต์ที่ป้อนในขั้นตอนการแปลง MixColumns() ทั้ง 4 ไบต์สามารถเขียนได้เป็น

$$S'_{0,C} = (\{02\} \cdot S_{0,C}) \otimes (\{03\} \cdot S_{1,C}) \otimes S_{0,C} \otimes S_{3,C}$$

$$S'_{1,C} = S_{0,C} \otimes (\{02\} \cdot S_{1,C}) \otimes (\{03\} \cdot S_{2,C}) \otimes S_{3,C}$$

$$S'_{2,C} = S_{0,C} \otimes S_{1,C} \otimes (\{02\} \cdot S_{2,C}) \otimes (\{03\} \cdot S_{3,C})$$

$$S'_{3,C} = (\{03\} \cdot S_{0,C}) \otimes S_{1,C} \otimes S_{2,C} \otimes (\{02\} \cdot S_{3,C})$$



รูปที่ 2.10 การแปลงด้วยวิธีผสมผสานคอลัมน์ด้วยฟังก์ชัน MixColumns() [1]

2.3.1.4 การบวกค่ากุญแจในแต่ละรอบ AddRoundKey ()

มาตรฐานของ AES มีขั้นตอนการบวกค่ากุญแจในแต่ละรอบ AddRoundKey() โดยใช้ตามขนาดของกุญแจ มีอยู่ 3 แบบ คือ 128, 192 หรือ 256 บิต ขนาดของกุญแจจะเขียนในหน่วยของเวิร์ดขนาด 32 บิต คือ Nk = 4, 6 หรือ 8 ตามลำดับ การเลือกหน่วยเป็นเวิร์ดก็เพื่อแสดงถึงจำนวนคอลัมน์ที่ใช้สำหรับกุญแจแต่ละขนาด โดยจำนวนรอบที่ต้องทำการเข้ารหัสลับ Nr จะขึ้นอยู่กับขนาดของกุญแจที่ใช้ ดังตารางที่ 2.10

ตารางที่ 2.10 สรุปรายละเอียดของจำนวนรอบการเข้ารหัสลับสำหรับกุญแจทั้ง 3 ขนาด, [1]

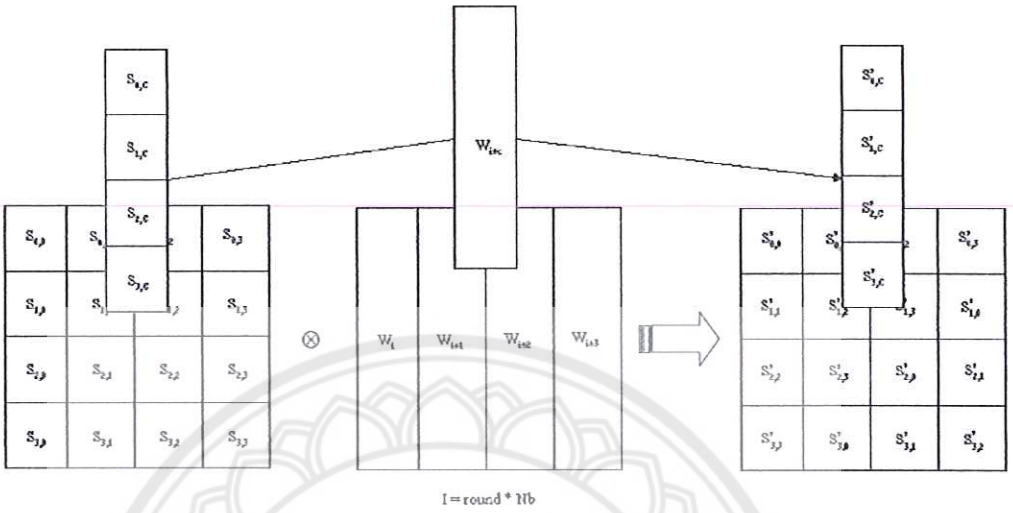
รูปแบบรหัสลับ	ขนาดของกุญแจ (Nk)	ขนาดของบล็อกข้อมูล (Nb)	จำนวนรอบการเข้ารหัสลับ (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

อัลกอริทึม AES จะนำค่าของกุญแจที่กำหนดไปผ่านการขยายขนาดของกุญแจ (key expansion) และทำเป็นตารางกุญแจ (key schedule) เพื่อใช้ในการทำ XOR กับอาร์เรย์ State ในแต่ละรอบของขั้นตอน AddRoundKey โดยตารางกุญแจที่ได้หลังจากนำตารางกุญแจไปผ่านขั้นตอนการขยายขนาดจะมีทั้งหมด Nb(Nr+1) เวิร์ด และเก็บลงในอาร์เรย์ของเวิร์ดที่เขียนแทนด้วย [Wi] สำหรับ $0 \leq i < Nb(Nr+1)$ เมื่อเตรียมตารางกุญแจแล้วจะนำกุญแจไป XOR กับตัวแปร State ดังนี้

$$[S'_{0,C}, S'_{1,C}, S'_{2,C}, S'_{3,C}] = [S_{0,C}, S_{1,C}, S_{2,C}, S_{3,C}] \otimes [W_{round*Nb+c}]$$

สำหรับ $0 \leq c < Nb$ สมการระบุว่าให้ทำ XOR กับตัวแปร State ทีละคอลัมน์ จนครบ Nb คอลัมน์ ดังนั้นการทำงานในแต่ละรอบของฟังก์ชัน AddRoundKey() ต้องใช้กุญแจ [Wi] จำนวน Nb เวิร์ด โดยเวิร์ดของกุญแจที่ใช้จะขึ้นอยู่กับค่าของตัวแปร round ซึ่งขึ้นกับหมายเลขของรอบที่ทำฟังก์ชัน

AddRoundKey () มีค่าอยู่ระหว่าง $0 \leq \text{round} < Nr$ ถ้า round = 0 เป็นการบวกค่ากุญแจเริ่มต้นให้ดูรูปที่ 2.11 ประกอบ หลังจากนั้นจึงเป็นการทำฟังก์ชัน AddRoundKey() ที่แท้จริง Nr รอบภายในลูป ซึ่งเป็นช่วงที่ $1 \leq \text{round} < Nr$



รูปที่ 2.11 ฟังก์ชั่น AddRoundKey() เป็นการทำ XOR ระหว่างตารางกุญแจกับตัวแปรstate [1]

2.3.1.5 การขยายขนาดของกุญแจ

การขยายขนาดของกุญแจ (key expansion) เป็นขั้นตอนที่ใช้เตรียมกุญแจในการเข้ารหัสลับในแต่ละรอบ ในรูปที่ 2.12 แสดงรายละเอียดการทำงานของการขยายขนาดกุญแจในรูปของโปรแกรมเทียม pseudo code โดยภายในโปรแกรมมีการเรียกใช้ฟังก์ชันเฉพาะ 2 ฟังก์ชัน ได้แก่ SubWord() และ RotWord() สำหรับฟังก์ชัน SubWord() จะรับเวิร์ดขนาด 4 ไบต์เพื่อป้อนเข้าสู่ตารางการแทนที่ S-Box ในตารางที่ 2.2 ส่วนฟังก์ชัน RotWord() รับเวิร์ด $[a_1,a_2,a_3,a_0]$ นอกจากฟังก์ชันแล้วยังมีอาร์เรย์ของเวิร์ดที่เป็นค่าคงที่ที่เรียกว่า Rcon[i] อาร์เรย์ดังกล่าวนี้บรรจุค่า $[x^{i-1},\{0,0\},\{0,0\},\{0,0\}]$ โดยที่ x มีค่าเป็น 02hex และ i มีค่าตั้งแต่ 1 ถึง 10 ยกตัวอย่างเช่น

$$\text{Rcon}[1] = 01000000_{hex}$$

$$\text{Rcon}[2] = 02000000_{hex}$$

$$\text{Rcon}[3] = 04000000_{hex}$$

$$\text{Rcon}[4] = 08000000_{hex}$$

จากรูปที่ 2.12 จะเห็นว่าเวิร์ด Nk แรกของตารางกุญแจที่ได้ผ่านการขยายขนาดแล้วเป็นค่าคงที่สำเนามาจากกุญแจเริ่มต้นที่ผู้ใช้กำหนด ส่วนเวิร์ดอื่นๆในลำดับต่อมา w[i] คำนวณได้จากการนำเวิร์ดก่อนหน้า w[i-1] ไปทำ XOR กับเวิร์ด ณ ตำแหน่ง Nk ให้นำ w[i-1] ไปผ่านขั้นตอน

การแปลง RotWord() และ SubWord() ก่อนจะทำกระบวนการ XOR นอกจากนี้ยังให้นำผลที่ได้ไปทำการ XOR กับค่าคงที่ Rcon[i]

การขยายขนาดของกุญแจสำหรับกุญแจขนาด 256 บิต (Nk = 8) มีวิธีการที่แตกต่างไปจากกรณีกุญแจขนาด 128 และ192 บิต คือ Nk = 8 และ i-4 มีค่าเป็นจำนวนเท่าของ Nk ให้ทำเพียงฟังก์ชัน SubWord() ก่อนจะนำไปผ่านกระบวนการ XOR

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)

Begin

      word temp
      i = 0

      while (i < Nk)
            w[i] = word(key[4*i],key[4*i+1],key[4*i+2],
            key[4*i+3])
            i = i+1
      end while

      i = Nk

      while (i < Nb * (Nr+1)]
            temp = w[i-1]
            if (i mod Nk = 0
                  temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
            else if (Nk > 6 and i mod Nk = 4)
                  temp = SubWord(temp)
            end if
                  w[i] = w[i-Nk] xor temp
                  i = i + 1
      end while

      end

Note that Nk=4, 6, and 8 do not all have to be implemented;
they are all included in the conditional statement above for

conciseness. Specific implementation requirements for the

Cipher Key are presented in Sec. 6.1.
```

รูปที่ 2.12 โปรแกรมเทียมแสดงการขยายขนาดกุญแจ [1]

กุญแจขนาด 128 บิต (Nk = 4)

กุญแจขนาด 128 บิตจะมีการวนลูปการเข้ารหัสลับ 10 รอบ ดังนั้นตารางกุญแจประกอบด้วยอาร์เรย์ของเวิร์ด w[0], w[1], ... , w[43] จากรูปที่10 ค่าของ w[0], w[1], w[2] และ w[3] มีค่าเท่ากับค่ากุญแจเริ่มต้นตามที่ผู้ใช้กำหนด สำหรับการหาค่าอื่นๆ ที่เหลือให้แบ่งออกเป็น 2 กลุ่ม กลุ่มแรกเป็นกลุ่มของเวิร์ดที่ค่าดรรชนี i เป็นจำนวนเท่าของ Nk = 4 ได้แก่ w[4], w[8],

w[12], ……,w[40] การคำนวณมีขั้นตอนมากกว่ากลุ่มของเวิร์ดที่เหลือที่ประกอบด้วยเวิร์ด เช่น w[5], w[6], w[7], w[9], w[10] หรือ w[11] เป็นต้น

สำหรับกลุ่มแรกให้คำนวณดังนี้

$$W[i] = \{SubWord(RotWord9w[i-1])) \otimes Rcon[i/Nk]\} \otimes w[i-Nk]$$

ส่วนกลุ่มที่สองให้คำนวณดังนี้

$$W[i] = w[i-1] \otimes w[i-Nk]$$

จะเห็นว่าการสร้างตารางกุญแจจะกระทำจากดรรชนีที่มีค่าน้อยไปหามากขึ้นอย่างเป็นลำดับ

กุญแจขนาด 192 บิต (Nk = 6)

กุญแจขนาด 192 บิตจะมีการวนลูปการเข้ารหัสลับ 12 รอบ ดังนั้นตารางกุญแจประกอบด้วยอาร์เรย์ของเวิร์ด w[0], w[1], ., w[51] จากรูปที่ 10 ค่าของ w[0], w[1] ,w[2], w[3], w[4] และ w[5] มีค่าเท่ากับค่ากุญแจเริ่มต้นตามที่ผู้ใช้กำหนด สำหรับการหาค่าอื่นๆ ที่เหลือให้แบ่งออกเป็น 2 กลุ่มเหมือนกับกุญแจขนาด 128 บิต และมีกระบวนการคำนวณตารางกุญแจที่เหลือทั้งหมดในลักษณะเดียวกัน กลุ่มของเวิร์ดที่ค่าดรรชนี i เป็นจำนวนเท่าของ Nk = 6 ได้แก่ w[6], w[12], w[18], …., w[48] ให้คำนวณดังนี้

$$W[i] = \{SubWord(RotWord9w[i-1])) \otimes Rcon[i/Nk]\} \otimes w[i-Nk]$$

ส่วนกลุ่มที่สองให้คำนวณดังนี้

$$W[i] = w[i-1] \otimes w[i-Nk]$$

กุญแจขนาด 256 บิต (Nk = 8)

กุญแจขนาด 256 บิตจะมีการวนลูปการเข้าเข้ารหัสลับ 14 รอบ ดังนั้นตารางกุญแจประกอบด้วยอาร์เรย์ของเวิร์ด w[0], w[1], ., w[59] จากรูปที่ 10 ค่าของ w[0], w[1] , w[2],….,w[7] มีค่าเท่ากับค่ากุญแจเริ่มต้นตามที่ผู้ใช้กำหนด สำหรับการหาค่าอื่นๆ ที่เหลือให้แบ่งออกเป็น 2 กลุ่ม เหมือนกับกรณีกุญแจขนาด 128 บิต และมีกุญแจขนาด 192 บิต แต่กระบวนการคำนวณตารางกุญแจที่เหลือทั้งหมดในลักษณะต่างไป คือ กลุ่มของเวิร์ดที่ค่าดรรชนี i-4 เป็นจำนวนเท่าของ Nk = 8 ได้แก่ w[12], w[20], w[28], …., w[52] ให้คำนวณดังนี้

$$W[i] = SubWord([i-1]) \otimes w[i-Nk]$$

ส่วนกลุ่มที่สองที่เหลือให้คำนวณดังนี้

$$W[i] = w[i-1] \otimes w[i-Nk]$$

2.3.2 การถอดรหัสลับ AES

มาตรฐานอัลกอริทึม AES การถอดรหัสลับมีการทำงานที่คล้ายคลึงกับการเข้ารหัสลับ แต่สลับลำดับการทำงานให้เป็นการทำงานย้อนกลับ การถอดรหัสลับประกอบไปด้วยฟังก์ชันการแปลงข้อมูลในระดับของไบต์ 4 ขั้นตอนหลัก ได้แก่ InvShiftRows(), InvSubBytes(), InvMixColumns() และ AddRoundKey() ดูภาพรวมไซเฟอร์ผกผันสำหรับการถอดรหัสลับ AES ในรูปของโปรแกรมเทียม Pseudo code ได้ในรูปที่ 2.13

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
Begin
byte state[4,Nb]
state = in
      AddRoundKey(state,w[Nr*Nb,(Nr+1)*Nb-1]) //See Sec. 5.1.4
      for round = Nr-1 step -1 downto 1
      InvShiftRows(state) // See Sec. 5.3.1
      InvSubBytes(state) // See Sec. 5.3.2
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(state) // See Sec. 5.3.3
      end for
      InvShiftRows(state)
      InvSubBytes(state)
      AddRoundKey(state, w[0, Nb-1])
out = state
end
```

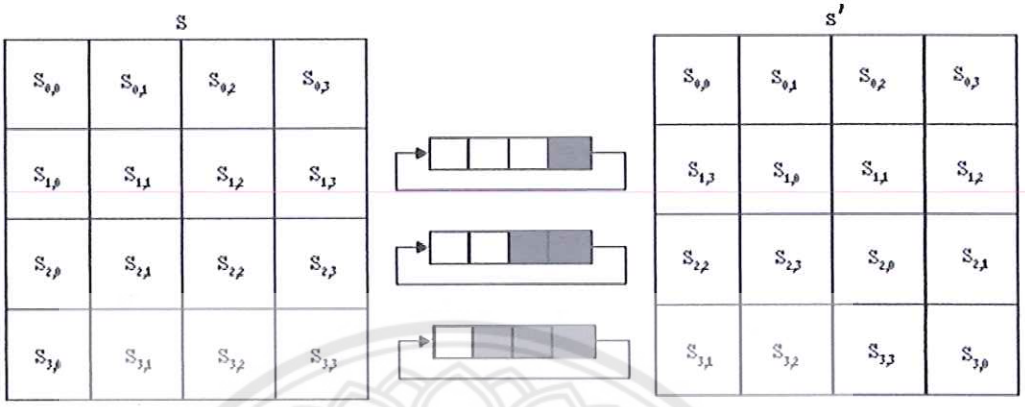
รูปที่ 2.13 การแสดงขั้นตอนการทำงานของไซเฟอร์ผกผันในรูปของโปรแกรมเทียม Pseudo code [1]

2.3.2.1 การแปลง InvshiftRows

ฟังก์ชันการแปลง InvShiftRows() เป็นกระบวนการแปลงผกผันของฟังก์ชัน ShiftRows()ที่ใช้ในกระบวนการเข้ารหัสลับ ชุดของไบต์ที่อยู่ใน 3 แถวล่างของตัวแปร State จะได้รับการเลื่อนตำแหน่งไปด้วยออฟเซตที่แตกต่างกัน ส่วนไบต์ในแถวแรก r = 0 ไม่มีการเลื่อนไบต์แต่อย่างใด

จำนวนตำแหน่งที่เลื่อนของชุดไบต์ในแต่ละแถวมีค่าเท่ากับ Nb – shift(r,Nb) ไบต์ โดยที่ค่าของ shift(r,Nb) ขึ้นกับหมายเลขแถว เราสามารถเขียนกรรมวิธีการแปลง InvShiftRows() ได้ดังนี้

$$S'_{r,(c,shift(r,Nb))mod\ Nb} = S_{r,c} \text{ สำหรับ } 0 < r > 4 \text{ และ } 0 \leq c < Nb$$



รูปที่ 2.14 แผนภาพการแปลงด้วยวิธีการเลื่อนแถวของฟังก์ชัน InvShiftRows () [1]

2.3.2.2 การแปลง InvSubBytes

ฟังก์ชันการแปลง InvSubBytes() เป็นกระบวนการแปลงผกผันของฟังก์ชัน SubBytes() ไบต์ภายในตัวแปรอาร์เรย์ State แต่ละไบต์จะได้รับการแทนที่ด้วยไบต์ค่าใหม่ตามที่ระบุในตารางการแทนที่ (S-Box) ในตารางที่ 2.9 กระบวนการภายใน S-Box ของการถอดรหัสลับนั้นจะทำงานกลับกันกับ S-Box ของกระบวนการเข้ารหัสลับ คือ จะทำการแปลงผกผันกับการแปลงแอฟไฟน์ จากนั้นจึงหาค่าตัวผกผันการคูณภายใต้ฟิลด์ GF(2^8)

ตารางที่ 2.11 รายละเอียดตารางการแทนที่ (S-Box) สำหรับใช้ในฟังก์ชันการแปลง InvSubBytes() ที่แสดงในรูปแบบของตัวเลขฐาน 16 [1]

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	c1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	C3	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

2.3.2.3 การแปลง InvMixColumns

ฟังก์ชันการแปลง InvMixColumns() เป็นกระบวนการแปลงผกผันของฟังก์ชัน MixColumns() ในการทำงานจะพิจารณาอาร์เรย์ State ทีละหนึ่งคอลัมน์ และเขียนแสดงแต่ละคอลัมน์ในรูปของพหุนาม นำพหุนามที่ได้ไปคูณกับ พหุนาม $a^{-1}(x)$ ที่มีค่าตายตัวภายใต้ modulo $x^4 + 1$

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

สามารถเขียนการคูณพหุนามในรูปของเมทริกซ์ได้ดังนี้

$$S'(x) = a^{-1}(x) \otimes s(x)$$

หรือ

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix}$$

โดยที่ $0 \leq c < Nb$ และ $Nb = 4$ ดังนั้นไบต์ที่ป้อนในขั้นตอนการแปลง MixColumns() ทั้ง 4 ไบต์ จะได้รับการแทนที่ดังนี้

$$S'_{0,c} = (\{0e\} \bullet S_{0,c}) \otimes (\{0b\} \bullet S_{1,c}) \otimes (\{0d\} \bullet S_{2,c}) \otimes (\{09\} \bullet S_{3,c})$$
$$S'_{1,c} = (\{09\} \bullet S_{0,c}) \otimes (\{0e\} \bullet S_{1,c}) \otimes (\{0b\} \bullet S_{2,c}) \otimes (\{0d\} \bullet S_{3,c})$$
$$S'_{2,c} = (\{0d\} \bullet S_{1,c}) \otimes (\{09\} \bullet S_{1,c}) \otimes (\{0e\} \bullet S_{2,c}) \otimes (\{0b\} \bullet S_{3,c})$$
$$S'_{3,c} = (\{0b\} \bullet S_{0,c}) \otimes (\{0d\} \bullet S_{1,c}) \otimes (\{09\} \bullet S_{2,c}) \otimes (\{0e\} \bullet S_{3,c})$$

2.3.2.4 การแปลงผกผันของ AddRoundKey

เนื่องจากในขั้นตอนการแปลงของฟังก์ชัน AddRoundKey() ประกอบไปด้วยการทำ XOR เพียงอย่างเดียว ดังนั้น กระบวนการที่สามารถทำกลับกันกับขั้นตอนการบวกค่ากุญแจนี้จึงคือฟังก์ชันเดิม กล่าวคือ ให้ใช้ฟังก์ชัน AddRoundKey() ในการแปลงผกผันเช่นเดียวกับการเข้ารหัสลับ

2.4 การเข้ารหัสลับข้อมูลแบบ RSA (Rivest-Shamir-Adelman Encryption)

อัลกอริทึม RSA ได้รับการพัฒนาขึ้นโดย Rivest Shamir และ Adleman ในปี ค.ศ. 1978 และในปัจจุบันนี้ยังสามารถใช้ รักษาความปลอดภัยของข้อมูลได้เป็นอย่างดี หลักการทำงานของการเข้ารหัสข้อมูลแบบนี้ก็คือ ความยากในการหาส่วนประกอบที่เป็นตัวเลขจำนวนเฉพาะ (Prime Number) ของตัวเลขจำนวนเฉพาะขนาดใหญ่ และข้อความต้นฉบับ ที่จะเข้ารหัสจะอยู่ในรูปของบล็อกตัวเลขไบนารีที่มีค่าจำกัดไม่เกิน n กล่าวคือ มีความยาวไม่เกิน log2(n) บิต

2.4.1 ความปลอดภัยของการเข้ารหัสแบบ RSA

ความปลอดภัยในการเข้ารหัสแบบ RSA ขึ้นกับความยากในการคำนวณหาค่า d จากค่า n และ e ที่มีอยู่ หากเราสามารถทำการถอดรูทหาตัวประกอบ (factorize) ของค่า n แล้วเราก็สามารถที่คำนวณหาค่า p และ q ได้ จากนั้นเราก็สามารถหาค่า d ซึ่งเป็น Private Key ได้ด้วยหากทำการถอดรูทหาตัวประกอบของค่าตัวเลขนั้นๆ ไม่มีความยากแล้วจะทำให้การเข้ารหัสแบบ RSA นั้นไม่มีความปลอดภัย คือผู้ที่ไม่ประสงค์ดีสามารถทำการคำนวณหาค่า d ได้ และนำค่า d ไปใช้ในการคำนวณถอดรหัสหาข้อความเดิมจากข้อความที่ขโมยมาเป็น Cipher Text โดยไม่ได้รับอนุญาต

ด้วยเทคโนโลยีปัจจุบัน เราสามารถทำการหาตัวประกอบตัวเลขที่มีขนาดใหญ่มาก ที่สุดได้ถึง 400 Bits แต่ก็ได้มีการวิจัยอย่างกว้างขวางที่จะพยายามเพิ่มขนาดให้ถึง 512 Bits ซึ่งเป็นขนาดที่ใช้ในการเข้ารหัสแบบ RSA

2.4.2 คุณสมบัติของการเข้ารหัสแบบ RSA

การเข้ารหัสแบบ RSA นั้นเป็นการเข้ารหัสแบบ Block Cipher ปกติการเข้ารหัสแบบ RSA จะช้ากว่าการเข้รหัสแบบอื่นๆ มาก เนื่องจากว่าต้องใช้การคำนวณที่สลับซับซ้อนและขนาดกุญแจที่ใช้มีขนาดใหญ่มาก เมื่อเทียบกับการเข้ารหัสแบบ DES แล้วนั้น การเข้ารหัสแบบ RSA จะช้ากว่าประมาณ 1,000 เท่า เนื่องจากความช้าในการเข้ารหัสข้อมูล จึงไม่นิยมเอา RSA ไปใช้ในการเข้ารหัสข้อความที่มีขนาดใหญ่ แต่จะเอาไปใช้ในการเข้ารหัสข้อมูลขนาดเล็กที่ต้องการความปลอดภัยสูงมากๆ เช่น ใช้ในการเข้ารหัสและแจกจ่าย Secret Key ที่ใช้เป็น Session Key ในการติดต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ในแต่ละครั้ง

2.4.3 การสร้าง Key (Key Generator)

หาจำนวนเฉพาะ 2 ตัว คือ p และ q สำหรับใช้ในการคำนวณค่าดังนี้

- n ได้จาก p*q
- m ได้จาก (p-1)*(q-1) ใช้ในการสร้าง Key
- e (public key) : เป็นตัวเลขตัวหนึ่งที่ไม่เป็นตัวประกอบของ m(e < mและ gcd(e,m) = 1)
- d (private key) = e^{-1} mod (m) คือ inverse ของ e modulo m

หมายเหตุ การเข้ารหัสโดยใช้ RSA Algorithm ต้องเก็บ p, q ไว้เป็นความลับ

ตัวอย่าง กระบวนการสร้างกุญแจ

1. ให้ p = 7 และ q =19
2. ให้ n = pq
   =7*19
   = 133
3. ให้ m = (p-1)(q-1)
   = 6*18
   =108
4. หา e จาก gcd(e, 108) = 1 (ได้ตามเงื่อนไขที่กำหนด) ฉะนั้น e = 5
5. หาค่า d ที่สอดคล้องกับเงื่อนไข d * e mod m = 1 ซึ่งหาค่า d จาก d = (1 + i *m)/ e (ลองแทนค่า i จนทำให้เงื่อนไขที่ระบุไว้เป็นจริง)

เมื่อ i = 3

จะได้ $d = (1 + 3*108)/ 5$

$d = 65$

2.4.4 การเข้ารหัส (Encryption)

การเข้ารหัสทำได้โดย

1.1 แทนค่าข้อมูลที่ต้องการเข้ารหัสในรูปของตัวเลข เช่น แปลง A เป็น 1, B เป็น 2 ฯลฯ

1.2 นำตัวเลขนั้นไปยกกำลังด้วย Public Key (e) และ modulo ด้วยค่า RSA-Modulus

ดังสมการ $C = (T^E) \bmod n$; T คือ ข้อความต้นฉบับ ; C คือ Cipher Text

1.3 ตัวเลขที่ได้คือข้อมูลที่ถูกเข้ารหัสแล้ว ซึ่งจะได้นำไปใช้ต่อไป

ตัวอย่าง กระบวนการเข้ารหัส

$C = T \bmod n$

$= 5^5 \bmod 133$

$= 3125 \bmod 133$

$= 66$

2.4.5 การถอดรหัส (Decryption)

กระบวนการถอดรหัสทำเช่นเดียวกับการเข้ารหัส โดยใช้ Private Key ในการยกกำลังแทน Public Key ดังสมการ $T = (C^d) \bmod n$ หลังจากนั้นจึงนำตัวเลขที่ได้ไปแทนค่าให้อยู่ในรูปแบบของข้อมูลเดิม

ตัวอย่าง กระบวนการถอดรหัส

$T = C^d \bmod n$

$= 66^{65} \bmod 133$

$= 66*66^{64} \bmod 133$

$= 66*(66^2)^{32} \bmod 133$

$= 66*(4356)^{32} \bmod 133$

$= 66*(4356 \bmod 133)^{32} \bmod 133$

$= 66*100^{32} \bmod 133$

$= 66*(100^2)^{16} \bmod 133$

$= 66*(10000 \bmod 133)^{16} \bmod 133$

$= 66*(25)^{16} \bmod 133$

$= 66*(93)^8 \bmod 133$

$= 66*(4)^4 \bmod 133$

= 16896 mod 133

= 5

ในการประยุกต์ใช้งานจริงนั้น ได้มีผู้กำหนดมาตรฐานของการเข้ารหัสโดยใช้ Asymmetric key นี้เพื่อที่จะให้ซอฟท์แวร์ของบริษัทต่างๆ สามารถทำงานเข้ากันได้ เช่น Public Key Cryptography Standard (PKCS) ที่เสนอโดย RSA Laboratories ในส่วนของโครงการฯ ได้นำวิธีการเข้ารหัส RSA นี้มาใช้ในส่วนของการตรวจสอบสิทธิการใช้งานของผู้ใช้ ข้อมูล username, password และสิทธิการใช้งานของผู้ใช้ที่รับส่งระหว่างเครื่องเซิร์ฟเวอร์และเครื่องของผู้ใช้จะถูกเข้ารหัสไว้โดยใช้ RSA ทำให้ผู้อื่นในระบบเครือข่ายไม่สามารถลักลอบนำข้อมูลของผู้ใช้ไปใช้ได้

บทที่ 3
การควบคุมผ่านระบบเครือข่าย

ในบทนี้จะกล่าวถึงโปรแกรม VNC ซึ่งเป็นโปรแกรมจำพวก Remote control และโปรแกรม UltraVNC ซึ่งเป็นหลักการพื้นฐานของ VNC มาใช้พัฒนาให้มีความสามารถมากกว่า VNC เช่น Plug-in ต่างๆ ในที่นี้ได้ศึกษาในเรื่องของ DSM (Data Stream Modification) ซึ่งเป็น Plug-in ตัวหนึ่งของ Ultra VNC โดยทำหน้าที่เข้ารหัสและถอดรหัสในการส่งข้อมูลของโปรแกรม Ultra VNC

3.1. การควบคุมระบบเครือข่าย

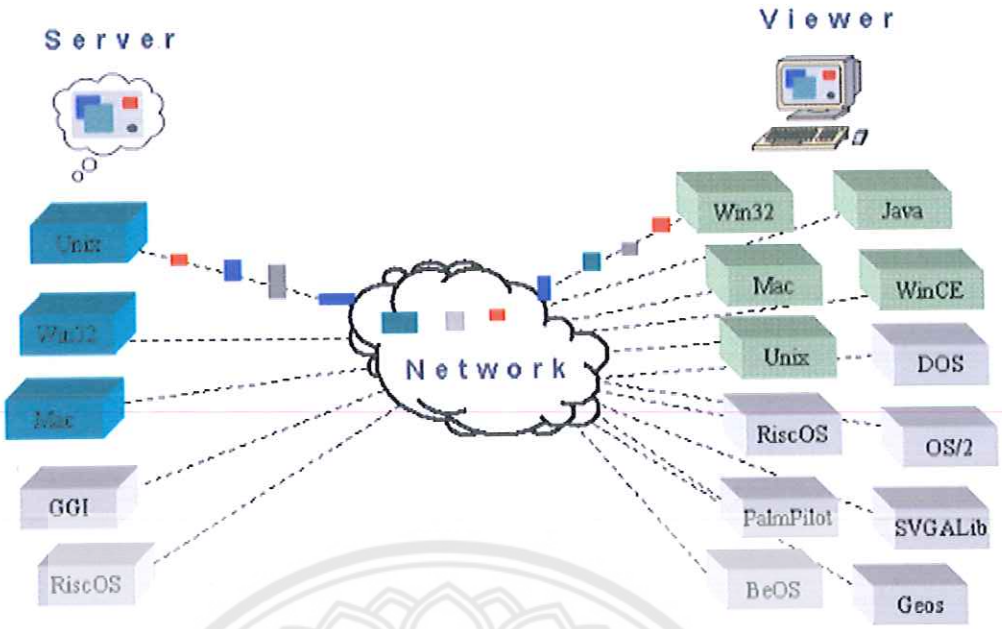
ระบบคอมพิวเตอร์เกิดขึ้นจากการนำคอมพิวเตอร์ตั้งแต่ 2 เครื่องมาเชื่อมต่อกันเพื่อให้เกิดประโยชน์ขึ้น เช่น การใช้ทรัพยากรร่วมกัน หรือการประหยัดค่าใช้จ่าย การควบคุมเครื่องคอมพิวเตอร์ผ่านระบบเครือข่ายก็เป็นประโยชน์อีกประการหนึ่ง เพื่อให้สามารถใช้งานคอมพิวเตอร์อยู่ในระยะไกล หรือควบคุมการทำงานของคอมพิวเตอร์เครื่องนั้นๆ ได้

การควบคุมคอมพิวเตอร์ผ่านเครือข่ายระยะไกลนั้น จำเป็นต้องมีซอฟท์แวร์ที่ใช้ในการติดต่อและควบคุม โดยโปรแกรมควบคุมคอมพิวเตอร์ระยะไกลมีให้เลือกใช้อยู่มากมาย ทั้ง freeware และจำพวกที่ต้องเสียค่าใช้จ่าย ตัวอย่างเช่น pcAnywhere, Vistron SM

ในที่นี้ใช้โปรแกรม UltraVNC ในการศึกษาเนื่องจากเป็น freewere และ open soure ที่สามารถนำมาพัฒนาต่อไปได้

3.2. VNC คืออะไร

VNC ย่อมาจาก Virtual Network Computing เป็น software ที่ควบคุมคอมพิวเตอร์ระยะไกล ซึ่งอนุญาตให้ดูและทำการโต้ตอบกับคอมพิวเตอร์เครื่องหนึ่ง (server) ได้โดยใช้โปรแกรมง่ายๆจากคอมพิวเตอร์อีกเครื่องหนึ่ง (viewer) ในทุกๆที่บนอินเตอร์เน็ต คอมพิวเตอร์ทั้ง 2 เครื่องไม่จำเป็นต้องมีลักษณะเหมือนกัน ตัวอย่างเช่น คุณสามารถใช้ VNC เพื่อดูเครื่องที่ทำงานบนระบบปฏิบัติการ Linux จากเครื่องที่บ้านซึ่งทำงานบนระบบปฏิบัติการ Windows ได้ VNC ใช้ประโยชน์ได้อย่างอิสระและเปิดเผย และใช้ทำงานอย่างแพร่หลายโดยใช้ในธุรกิจจำนวนมาก, ในห้องเรียน และเรื่องส่วนตัว



รูปที่ 3.1 แสดงการเชื่อต่อผ่านระบบเครือข่าย [8]

3.2.1. การทำงานของ VNC

VNC เป็น software ควบคุมคอมพิวเตอร์ระยะไกล ที่มีการทำงานได้หลายรูปแบบ ซึ่งเป็นการอนุญาตให้คอมพิวเตอร์ถูกควบคุมได้จากระยะไกลผ่านระบบเครือข่าย เสมือนนั่งอยู่หน้าจอของคอมพิวเตอร์เครื่องนั้น การใช้งานในด้านชีวิตประจำวัน สามารถใช้ VNC แก้ไขปัญหาเฉพาะด้านต่างๆ ได้ เช่น คุณอยู่ที่กรุงเทพฯสามารถใช้ VNC ควบคุมคอมพิวเตอร์ของแม่ของคุณในลอนดอน และแสดงให้เห็นวิธีติดตั้งและใช้ชุดซอฟแวร์ใหม่ได้อย่างง่ายดาย

การใช้งานในด้านธุรกิจ, VNC สามารถถูกใช้เพื่อความสะดวกสบายในการทำงาน โดยอนุญาตให้พนักงานเข้าถึงคอมพิวเตอร์ หรือเครื่อง Server ของบริษัทได้จากพื้นที่ห่างไกลผ่านระบบเครือข่าย โดยไม่คำนึงถึงระบบปฏิบัติการ VNC ถูกนำไปใช้งานมากที่สุดในเรื่องของการบริหารคอมพิวเตอร์ระยะไกลผ่านระบบเครือข่าย เพื่อควบคุมการทำงานของพนักงาน เพื่อหาสาเหตุและแก้ปัญหา รวมทั้งสิทธิในการเข้าใช้งานและการจัดการเครื่อง Server

นอกจากนี้ VNC สามารถนำมาใช้ในการศึกษา ตัวอย่างเช่น การให้เผยแพร่แก่นักเรียนหลายๆคนในเวลาเดียวกันจากคอมพิวเตอร์ซึ่งจัดการโดยผู้สอน หรือการอนุญาตให้ผู้สอนเข้าไปควบคุมคอมพิวเตอร์ของนักเรียนเพื่อทำการช่วยเหลือ

ข้อดีของ VNC

VNC แตกต่างจากระบบแสดงระยะไกลอื่นๆ 3 ข้อ คือ

1. VNC เป็นระบบที่สามารถทำงานข้ามระบบปฏิบัติการได้ เช่น การทำงานบนระบบปฏิบัติการ Linux สามารถแสดงผลได้ในเครื่องที่ทำงานบนระบบปฏิบัติการ windows, Solaris หรือบนระบบปฏิบัติการอื่นๆ ได้ ถ้าเป็นเครื่อง Java Viewer ทุกๆหน้าจอจะสามารถแสดงได้ด้วย Java-capable browser ถ้าเป็น Windows Server การอนุญาตให้สามารถเข้าไปดูอีกเครื่องหนึ่งได้จากระยะไกลคอมพิวเตอร์ทั้งสองเครื่องจะต้องใช้ระบบปฏิบัติการเดียวกัน ความง่ายของโปรโตคอลทำให้มันง่ายเพื่อทำระบบปฏิบัติการใหม่และคนอื่นต้องมี VNC ในรูปแบบที่หลากหลาย

2. VNC มีขนาดเล็กและง่าย ตัวอย่างเช่นในเครื่อง viewer ใช้เนื้อที่เพียง 150K และสามารถทำงานได้โดยตรงจาก floppy ถ้าเป็นเครื่อง Java Viewer ทั้งหมดจะใช้เนื้อที่อย่างน้อยที่สุด 100K และต้องใช้เวลาในการ download รูปภาพจากหน้า web page

3. VNC ฟรี โดยสามารถ download ได้จาก http://www.realvnc.com/download.html, นำไปใช้ และแจกจ่ายมันอีกครั้งใต้ข้อตกลงของใบอนุญาตมหาชนทั่วไป GNU

3.3. UltraVNC

UltraVNC เป็น free software ที่ใช้งานง่ายและรวดเร็วที่สามารถแสดงหน้าจอคอมพิวเตอร์ของเครื่องอื่น (via internet or network) บนหน้าจอของเครื่องเราเอง โดยผู้ใช้สามารถใช้เมาส์และคีย์บอร์ดเพื่อที่จะควบคุมคอมพิวเตอร์เครื่องอื่นในระยะไกล เสมือนกับการทำงานอยู่กับเครื่องที่ถูกควบคุม

ประโยชน์ที่เห็นได้ชัดจาก UltraVNC เช่นในกรณี การแก้ไขปัญหาให้กับลูกค้าโดยที่ลูกค้าไม่จำเป็นต้องนำเครื่องมาให้กับผู้ให้บริการหรือ เช่นในระบบธุรกิจต่าง ที่เจ้าหน้าที่ technical service ต้องแก้ปัญหาให้กับบริษัทลูกค้าในบางปัญหาก็ไม่จำเป็นที่จะต้องเดินทางไปที่บริษัทของลูกค้า โดยสามารแก้ไขผ่าน VNC ได้เลย

นอกจากนี้ UltraVNC ยังมีตัวเสริมที่มีประสิทธิภาพสูงอยู่อีกด้วย อธิเช่น Repeater, SingleClick packager และ Nat to Nat connector นั่นจะทำให้ง่ายต่อการจัดการสถานการณ์ของการเชื่อมต่อที่ซับซ้อนต่างๆ

UltraVNC ทำงานอยู่บนระบบปฏิบัติการ Windows™ (95, 98, Me, NT4, 2000, XP, 2003 ...) แต่ก็ได้รวม JavaViewer เพื่ออนุญาตให้ติดต่อ (และทำการส่งข้อมูล) จาก web browser ทั่วๆไปของระบบปฏิบัติการที่รองรับ Java™ (Linux, Mac OS ...) เพื่อที่จะเป็น UltraVNC Server

3.3.1. Data Stream Encryption Plug-in

Data Stream Modification (DSM) Plug-in ระบบทำการอนุญาตสำหรับการแลกเปลี่ยนข้อมูลชนิดต่างๆระหว่าง client และ server จาก DLL ภายนอกตัวหนึ่ง ประกอบด้วย การตรวจสอบความถูกต้อง, ตัวตั้งเวลาการติดต่อ, การบันทึกข้อมูล/persistence, การเข้ารหัส ทั้งนี้ขึ้นอยู่กับการพัฒนา DLL ปัจจุบันมี Encryption Plug-inออกมาหลาย version

DSM โดยสังเขป

UltraVNC ได้รวมเอา Data Stream Modification plug-in เอาไว้ ที่ได้ถูกออกแบบโดยหวังว่าจะเร็ว, มีประสิทธิภาพ, และลดขนาดข้อมูลและการใช้ CPU ให้น้อยที่สุดระหว่างการส่งข้อมูล DSM ยอมให้ใครก็ได้เขียน DLL ภายนอกที่สามารถ load ได้ทั้ง UltraVNC viewer และ UltraVNC server หลังจากนั้น DLL นี้จะสามารถเข้าถึงทุกๆการส่ง data packets และสามารถเปลี่ยนแปลง, ปรับปรุง, บันทึก, หรือ เข้ารหัส VNC data stream ระหว่างการเชื่อมต่อได้ในที่สุด

ระบบ DSM plug-in เป็น "tunnel" ของการเชื่อมต่อ VNC

รูปที่ 3.2 DSM model

ในตัวอย่างข้างต้นการส่งข้อมูลจาก viewer จะผ่านมายังส่วนของ Encoder/Compressor และ ข้อมูลทั้งหมดจะถูกส่งออกไปยัง internet โดยมี DSM ทำหน้าที่เข้ารหัสลับก่อนที่จะส่งออกไป เมื่อข้อมูลไปถึงปลายทาง ที่ปลายทางก็ จะมีส่วนของ DSM ในการถอดรหัสลับเช่นเดียวกันและเมื่อทำการถอดรหัสเรียบร้อยแล้วจึงส่งให้กับส่วนที่ทำหน้าที่ Decoder/Uncompressor แล้วจึงส่งข้อมูลมาแสดงที่ส่วนของ server

บทที่ 4
ขั้นตอนการดำเนินงาน

หลังจากการศึกษาทฤษฎีและหลักการที่นำมาใช้กับโครงงานจากบทที่ผ่านๆมาสามารถนำหลักการมาประยุกต์ใช้กับการดำเนินโครงงานที่สามารถนำมาใช้งานได้จริง

ขั้นตอนการดำเนินงานของโครงการนี้ได้วางแผนในการพัฒนาโดยรวมออกเป็น 3 ส่วน ส่วนแรก คือ การศึกษาและเขียนโปรแกรมแสดงการทำงานของอัลกอริทึม 3 ตัว คือ DES, AES และ RSA เพื่อให้ทราบถึงวิธีการทำงานของอัลกอริทึมนั้นจริงๆ, ส่วนที่ 2 นำ Plug-in จาก UltraVNC มาพัฒนาโดยใช้ API ของ Windows และส่วนสุดท้ายคือ การทดสอบ Performance จากเครื่องคอมพิวเตอร์ 3 เครื่อง ทำการทดสอบ 10 ครั้งเพื่อหาค่าความแตกต่างของการใช้ Plug-in แต่ละตัว ที่เชื่อถือได้มากที่สุด

4.1 การศึกษาและเขียนโปรแกรมแสดงการทำงานของอัลกอริทึม

เนื่องจากอัลกอริทึมที่ทำการศึกษามีอยู่ทั้งหมด 3 ตัว คือ DES, AES และ RSA เพื่อให้ทราบและเข้าใจถึงขั้นตอนการทำงานของอัลกอริทึมแต่ละตัวจึงต้องทำการศึกษาอย่างละเอียดตามขั้นตอนต่างๆที่ได้แสดงในบทที่ 2 แล้วนำมาเขียนเป็นโปรแกรมแสดงการเข้ารหัสและถอดรหัส จากการศึกษาอัลกอริทึมทั้ง 3 ตัว พบว่า

อัลกอริทึม DES พิจารณาข้อความต้นฉบับครั้งละ 64 บิต มีการทำงานเป็นรอบทั้งหมด 16 รอบโดยแบ่งการทำงานในการเข้ารหัสและถอดรหัสออกเป็น 2 ส่วน คือ การจัดเตรียมกุญแจที่ใช้ในการเข้ารหัสแต่ละรอบ และการเข้ารหัสแต่ละรอบ

อัลกอริทึม AES พิจารณาข้อความต้นฉบับได้ครั้งละ 128 บิต มีการทำงานเป็นรอบทั้งหมด 10 รอบโดยแบ่งการทำงานในการเข้ารหัสและถอดรหัสออกเป็น 2 ส่วนเช่นเดียวกับอัลกอริทึม DES คือ การจัดเตรียมกุญแจที่ใช้ในการเข้ารหัสแต่ละรอบ และการเข้ารหัสแต่ละรอบ แต่จะต่างกันที่การเก็บค่าตัวแปร โดยอัลกอริทึม AES จะเก็บแบบ State

อัลกอริทึม RSA เป็นการเข้ารหัสแบบ Block Cipher การเข้ารหัสช้ากว่าการเข้ารหัสแบบอื่นๆ มาก เนื่องจากต้องใช้การคำนวณที่สลับซับซ้อนและขนาดกุญแจที่ใช้มีขนาดใหญ่มาก

การเขียนโปรแกรมจากอัลกอริทึมทั้ง 3 ตัว พบว่า อัลกอริทึม RSA อัลกอริทึมไม่มีความซับซ้อนมากนัก จึงสามารถเขียนโปรแกรมได้ง่าย ส่วนอัลกอริทึม DES อัลกอริทึมมีการทำงานหลายขั้นตอนแต่ไม่ซับซ้อนมากนัก ส่วนอัลกอริทึม AES อัลกอริทึมมีความซับซ้อนมากที่สุด มีการทำงานหลายรอบ อัลกอริทึม AES จึงเขียนได้ยากมากกว่าอัลกอริทึมอื่นๆ

4.2 การนำ Plug-in จาก UltraVNC มาพัฒนาโดยใช้ API ของ Microsoft Visual Studio.NET 2003

การพัฒนา Plug-in ของ UltraVNC ในที่นี้ได้พัฒนาโดยใช้ Microsoft Visual C++ ซึ่งจำเป็นต้องใช้ Cryptographic Library ในการเข้ารหัส ซึ่ง Microsoft Visual Studio.NET 2003 มี API ที่ใช้ในการเขียนโปรแกรมที่เกี่ยวข้องกับการเข้ารหัส คือ Cryptographic API ที่ Microsoft ได้จัดเตรียมไว้ให้อยู่แล้วจึงได้ใช้ Cryptographic API ในการเขียน Plug-in ดังกล่าว โดยพัฒนาจาก Code ที่เป็น Open Source ของ plug-in เดิมที่ใช้การเข้ารหัสแบบ RC4 นั่นคือ MSRC4Plugin118 ซึ่ง provider ของ AES ที่ใช้ คือ PROV_RSA_AES และ Key generator CALG_AES_128 โดยนำเอา provider เหล่านี้ไปแทนใน Code เดิมซึ่งใช้อัลกอริทึมของ RC4 คือ PROV_RSA_FULL และ CALG_RC4

4.3 ทดสอบการทำงานของโปรแกรม

การทดสอบประสิทธิภาพการทำงานของโปรแกรมที่พัฒนาจาก Plug-in ของ UltraVNC จะเปรียบเทียบจาก 2 กรณีที่แตกต่างกัน คือ

1. ไม่มีการใช้ Plug-in เพื่อใช้เป็นมาตรฐานในการเปรียบเทียบกับกรณีที่ต้องการศึกษาต่อไป
2. ใช้ MSRC4Plugin.dsm เป็น Plug-in ที่มีอยู่แล้วในโปรแกรม UltraVNC

ทำการทดสอบค่า CPU Performance ระหว่างที่ใช้โปรแกรม UltraVNC โดยทำงานทดสอบอยู่ 4 ช่วงคือ

1. ทำการเชื่อมต่อคอมพิวเตอร์สองเครื่องด้วย UltraVNC
2. เปิดโปรแกรม Notepad ขึ้นมาใช้งาน
3. ใช้งานโปรแกรม Notepad
4. ใช้งานโปรแกรม Paint

ใช้เครื่องคอมพิวเตอร์ในการทดสอบ 3 เครื่อง คือ

1. IP Address: 192.168.1.2 Microsoft Window XP Professional Version 2002 Service Pack2 Intel® Celeron® M Processor 1300 MHz 1.30 GHz, 512 MB of RAM
2. IP Address: 192.168.1.3 Microsoft Window XP Professional Version 2002 Service Pack2 Intel Pentium III Processor 866 MHz, 256 MB of RAM
3. IP Address: 192.168.1.44 Microsoft Window XP Professional Version 2002 Service Pack2 Intel® Pentium® M Centrino™ 725 Processor 1.60 GHz, 768 MB of RAM

ในการทดสอบจะทดสอบคอมพิวเตอร์ทีละ 2 เครื่องสับเปลี่ยนกันไปจนครบจะมีอยู่ทั้งหมด 18 กรณี โดยเครื่องหนึ่งตั้งค่าให้เป็นเครื่อง Server และอีกเครื่องหนึ่งตั้งค่าให้เป็นเครื่อง Viewer ทำการทดสอบทั้งหมด 10 ครั้งเพื่อหาค่าเฉลี่ยที่เชื่อถือได้

บทที่ 5
ผลการทดลองและการวิเคราะห์ผล

5.1 จุดประสงค์ของการทดสอบโปรแกรม

1. เพื่อทำการทดสอบว่า plug-in ที่พัฒนาขึ้นมาสามารถเข้ารหัสและถอดรหัสได้จริง
2. เพื่อทดสอบประสิทธิภาพการทำงานของ plug-in โดยเปรียบเทียบการทำงานของ CPU ระหว่างไม่มี Plug-in และ MSRC4Plugin

5.2 ขั้นตอนการทดสอบการทำงานของโปรแกรม

1. ติดตั้งโปรแกรม UltraVNC ในส่วนของ Server และ Viewer ไว้ที่คอมพิวเตอร์ทั้ง 3 เครื่องที่จะนำมาทดสอบ
2. เชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกัน โดยระบุ IP Address ของคอมพิวเตอร์แต่ละเครื่องให้ชัดเจน และคอมพิวเตอร์แต่ละเครื่องมีลักษณะของเครื่อง ดังตารางที่ 5.1

ตารางที่ 5.1 แสดงลักษณะของคอมพิวเตอร์ที่ใช้ในการทดสอบ

เครื่อง	ลักษณะของเครื่อง
A	Microsoft Window XP Professional Version 2002 Service Pack 2 Intel® Celeron® M Processor 1300 MHz 1.30 GHz, 512 MB of RAM
B	Microsoft Window XP Professional Version 2002 Service Pack 2 Intel® Celeron® M Processor 1.6 GHz 1.60 GHz, 768 MB of RAM
C	Microsoft Window XP Professional Version 2002 Service Pack 2 Intel Pentium III Processor 866 MHz, 256 MB of RAM

3. เปิดโปรแกรม UltraVNC เพื่อทดสอบการทำงานระหว่างเครื่อง Server และเครื่อง Viewer โดยให้เครื่องหนึ่งเปิด UltraVNC Server ขึ้นมา อีกเครื่องหนึ่งเปิด UltraVNC Viewer (ในกรณีที่ใช้ plug-in ให้ทำการตั้งค่าของ plug-in ของทั้งสองเครื่องให้ตรงกัน)
4. ทดสอบการทำงานโดยใช้เครื่อง Viewer ติดต่อเข้าไปยังเครื่อง Server และทำการใช้โปรแกรมต่างๆในเครื่อง Server ซึ่งในตารางผลการทดลองได้กำหนดหมายเลข คือ

ตารางที่ 5.2 แสดงความหมายของหมายเลขที่แสดงในตารางผลการทดสอบ

หมายเลขแสดงในตารางผลการทดลอง	การใช้งานโปรแกรมเมื่อเปิดใช้งาน UltraVNC
1	เปิดโปรแกรม UltraVNC
2	เปิดโปรแกรม Notepad
3	ใช้งานโปรแกรม Notepad
4	ใช้งานโปรแกรม Paint

5. สังเกตการทำงานของ CPU ในการใช้โปรแกรมต่างๆ โดยดูจาก CPU Performance แล้วจดบันทึก
6. ทำการทดสอบไปเรื่อยๆจนครบทุกกรณี

5.3 ผลการทดสอบโปรแกรม

กรณีที่ 1: ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer C เมื่อไม่มีการใช้ Plug-in

รูปที่ 5.1 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server Bและเครื่อง Viewer C

ตารางที่ 5.3 ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer C เมื่อไม่มีการใช้ Plug-in

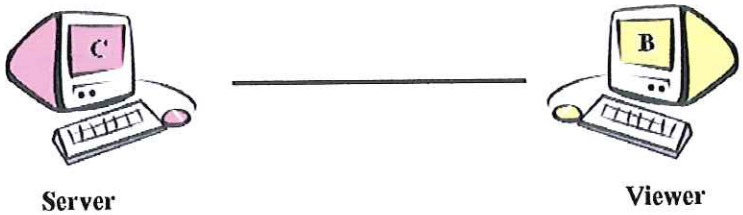
เครื่อง Server B

	1	2	3	4
1	31	34	11	44
2	28	31	11	37
3	23	32	11	39
4	31	34	13	43
5	32	30	12	49
6	30	27	10	49
7	23	28	9	45
8	20	27	12	43
9	27	22	8	38
10	22	28	11	43

เครื่อง Viewer C

	1	2	3	4
1	25	4	2	6
2	31	1	2	6
3	37	6	2	7
4	32	3	1	6
5	32	4	2	7
6	36	3	1	6
7	21	3	2	9
8	23	4	2	9
9	32	2	5	10
10	34	3	3	6

กรณีที่ 2: ผลการทดลองระหว่างเครื่อง Server C และเครื่อง Viewer B เมื่อไม่มีการใช้ Plug-in



รูปที่ 5.2 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server C และเครื่อง Viewer B

ตารางที่ 5.4 ผลการทดลองระหว่างเครื่อง Server C และเครื่อง Viewer B เมื่อไม่มีการใช้ Plug-in

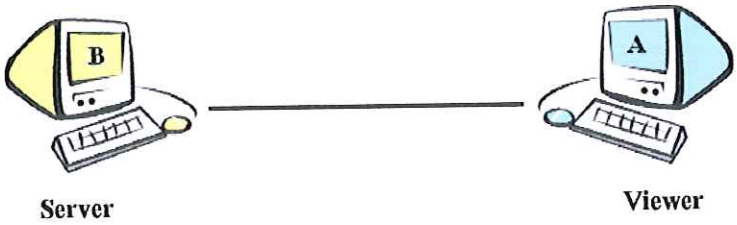
เครื่อง Server C

	1	2	3	4
1	28	53	25	79
2	27	60	16	69
3	23	53	17	81
4	28	54	21	71
5	29	55	19	80
6	17	40	17	64
7	24	42	15	78
8	30	37	12	81
9	28	40	18	75
10	26	43	16	67

เครื่อง Viewer B

	1	2	3	4
1	14	6	1	11
2	21	8	2	5
3	16	5	4	7
4	19	8	4	8
5	18	5	2	10
6	14	5	2	11
7	18	8	4	10
8	12	7	4	8
9	18	7	3	8
10	21	8	2	8

กรณีที่ 3: ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer A เมื่อไม่มีการใช้ Plug-in



รูปที่ 5.3 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server B และเครื่อง Viewer A

ตารางที่ 5.5 ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer A เมื่อไม่มีการใช้ Plug-in

เครื่อง Server B

	1	2	3	4
1	20	22	8	50
2	31	29	10	48
3	22	26	11	40
4	19	21	9	49
5	25	30	11	44
6	30	34	11	47
7	29	22	10	53
8	21	25	11	50
9	29	18	11	41
10	26	29	9	52

เครื่อง Viewer A

	1	2	3	4
1	60	18	6	28
2	60	16	6	28
3	54	14	8	30
4	50	10	6	26
5	43	18	8	32
6	41	10	8	22
7	56	15	8	24
8	57	17	6	22
9	74	10	7	22
10	47	12	7	24

กรณีที่ 4: ผลการทดลองระหว่างเครื่อง Server A และเครื่อง Viewer B เมื่อไม่มีการใช้ Plug-in

รูปที่ 5.4 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server A และเครื่อง Viewer B

ตารางที่ 5.6 ผลการทดลองระหว่างเครื่อง Server A และเครื่อง Viewer B เมื่อไม่มีการใช้ Plug-in

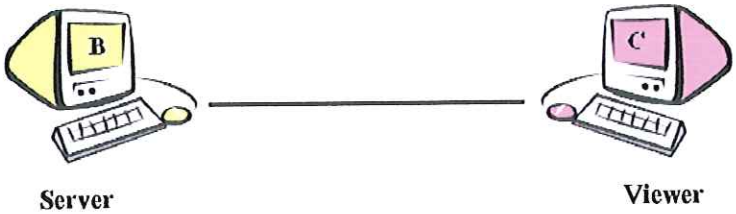
เครื่อง Server A

	1	2	3	4
1	72	74	38	97
2	70	69	42	92
3	60	77	44	88
4	54	82	42	94
5	43	86	34	99
6	68	67	50	99
7	47	71	36	97
8	48	70	36	96
9	50	72	42	99
10	41	89	38	88

เครื่อง Viewer B

	1	2	3	4
1	21	8	4	8
2	18	9	4	12
3	22	8	3	9
4	20	4	4	7
5	19	3	3	11
6	18	9	3	11
7	18	8	3	9
8	18	9	4	7
9	24	7	4	8
10	15	10	4	9

กรณีที่ 5: ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer C เมื่อใช้ MSRC4 Plug-in



รูปที่ 5.5 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server B และเครื่อง Viewer C

ตารางที่ 5.7 ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer C เมื่อใช้ MSRC4 Plug-in

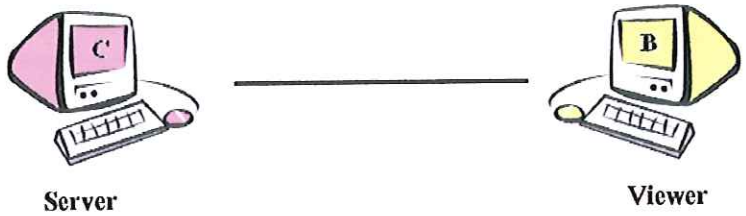
เครื่อง Server B

	1	2	3	4
1	20	37	10	44
2	31	31	8	50
3	27	28	12	49
4	33	28	10	50
5	25	33	12	45
6	33	19	9	43
7	25	29	8	43
8	25	32	9	44
9	30	31	9	50
10	25	36	10	51

เครื่อง Viewer C

	1	2	3	4
1	34	2	2	8
2	34	4	3	7
3	39	3	2	8
4	30	6	2	9
5	39	2	4	12
6	33	3	3	16
7	31	5	2	9
8	34	2	4	6
9	31	3	2	8
10	33	5	3	6

กรณีที่ 6: ผลการทดลองระหว่างเครื่อง Server C และเครื่อง Viewer B เมื่อใช้ MSRC4 Plug-in



รูปที่ 5.6 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server C และเครื่อง Viewer B

ตารางที่ 5.8 ผลการทดลองระหว่างเครื่อง Server C และเครื่อง Viewer B เมื่อใช้ MSRC4 Plug-in

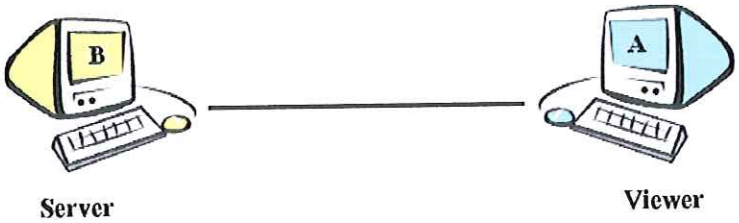
เครื่อง Server C

	1	2	3	4
1	28	44	18	68
2	30	40	19	86
3	27	39	19	66
4	29	43	19	81
5	31	44	20	71
6	30	41	15	76
7	29	38	21	73
8	28	46	17	83
9	28	47	16	73
10	28	36	16	80

เครื่อง Viewer B

	1	2	3	4
1	21	6	3	9
2	15	7	2	11
3	11	7	3	9
4	19	8	4	10
5	18	10	4	8
6	19	10	4	8
7	18	9	3	11
8	17	5	3	8
9	20	11	3	9
10	15	7	3	8

กรณีที่ 7: ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer A เมื่อใช้ MSRC4 Plug-in



รูปที่ 5.7 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server B และเครื่อง Viewer A

ตารางที่ 5.9 ผลการทดลองระหว่างเครื่อง Server B และเครื่อง Viewer A เมื่อใช้ MSRC4 Plug-in

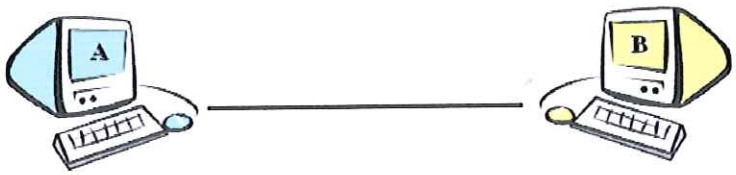
เครื่อง Server B

	1	2	3	4
1	28	33	11	51
2	29	25	13	32
3	29	23	11	43
4	23	22	11	38
5	28	32	13	25
6	21	23	11	42
7	22	23	11	42
8	25	33	12	45
9	26	30	11	28
10	31	27	14	26

เครื่อง Viewer C

	1	2	3	4
1	71	18	12	43
2	64	16	10	43
3	66	14	8	30
4	52	12	8	38
5	67	10	5	58
6	66	12	6	44
7	59	12	8	54
8	48	17	12	56
9	64	16	6	32
10	72	12	7	34

กรณีที่ 8: ผลการทดลองระหว่างเครื่อง Server A และเครื่อง Viewer B เมื่อใช้ MSRC4 Plug-in



รูปที่ 5.8 แสดงลักษณะการเชื่อมต่อระหว่างเครื่อง Server A และเครื่อง Viewer B

ตารางที่ 5.10 ผลการทดลองระหว่างเครื่อง Server A และเครื่อง Viewer B เมื่อใช้ MSRC4 Plug-in

เครื่อง Server A

	1	2	3	4
1	19	8	4	10
2	13	8	4	11
3	17	9	4	11
4	19	8	3	13
5	12	10	4	13
6	13	11	3	10
7	12	8	4	10
8	19	8	4	13
9	15	9	4	10
10	13	9	4	12

เครื่อง Viewer B

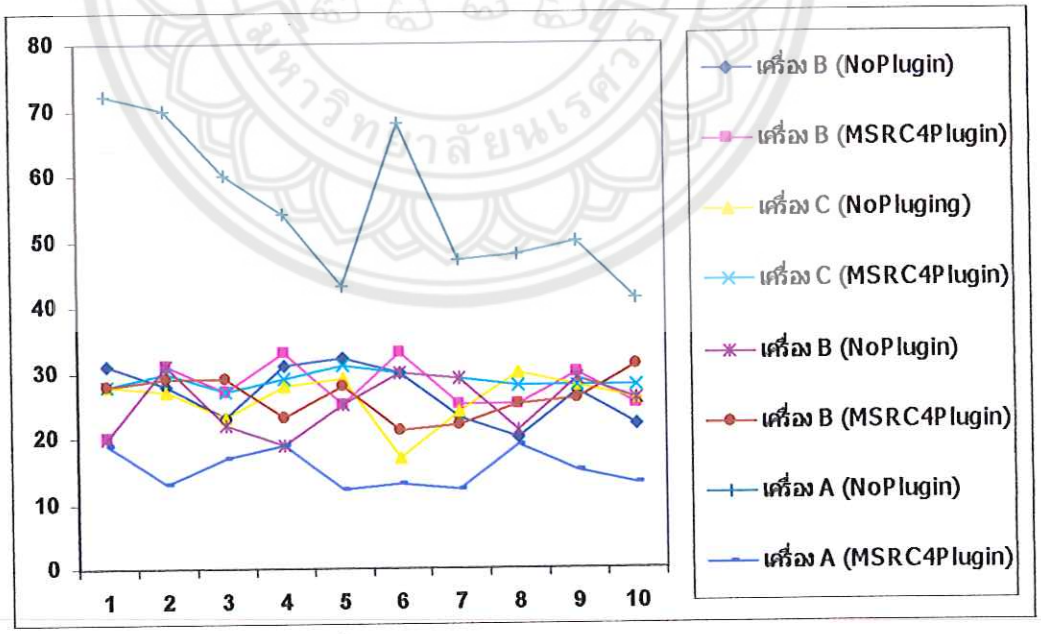
	1	2	3	4
1	52	70	40	95
2	46	72	54	99
3	36	60	48	82
4	50	76	38	88
5	50	58	34	83
6	46	54	36	91
7	46	80	42	83
8	38	64	35	90
9	43	71	44	96
10	48	67	47	87

5.4 วิเคราะห์ผล

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้โปรแกรม UltraVNC (Server) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.5 แล้วนำมาแสดงผลในรูปของกราฟ รูปที่ 5.1

ตารางที่ 5.11 เปิดโปรแกรม UltraVNC (Server)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	31	20	28	28	20	28	72	19
	28	31	27	30	31	29	70	13
	23	27	23	27	22	29	60	17
	31	33	28	29	19	23	54	19
	32	25	29	31	25	28	43	12
	30	33	17	30	30	21	68	13
	23	25	24	29	29	22	47	12
	20	25	30	28	21	25	48	19
	27	30	28	28	29	26	50	15
	22	25	26	28	26	31	41	13
Average	26.7	27.4	26	28.8	25.2	26.2	55.3	15.2
stdev	4.37289632	4.221637380	3.829708431	1.22927259	4.467164151	3.359894178	11.47993418	3.011090611
var	19.12222222	17.82222222	14.66666667	1.511111111	19.95555556	11.28888889	131.7888889	9.066666667
n	10	10	10	10	10	10	10	10
varh	1.912222222	1.782222222	1.466666667	0.151111111	1.995555556	1.128888889	13.17888889	0.906666667
v	39.34831153		11.38551021		25.08148874		10.6403262	
t(table)	1.684		1.796		1.708		1.812	
sqrt	1.922093766		1.271918935		1.767609811		3.753072815	
t'	3.236805902		2.284366408		3.019077557		6.800567941	
AvgX-AvgY	-0.7		-2.8		-1		40.1	

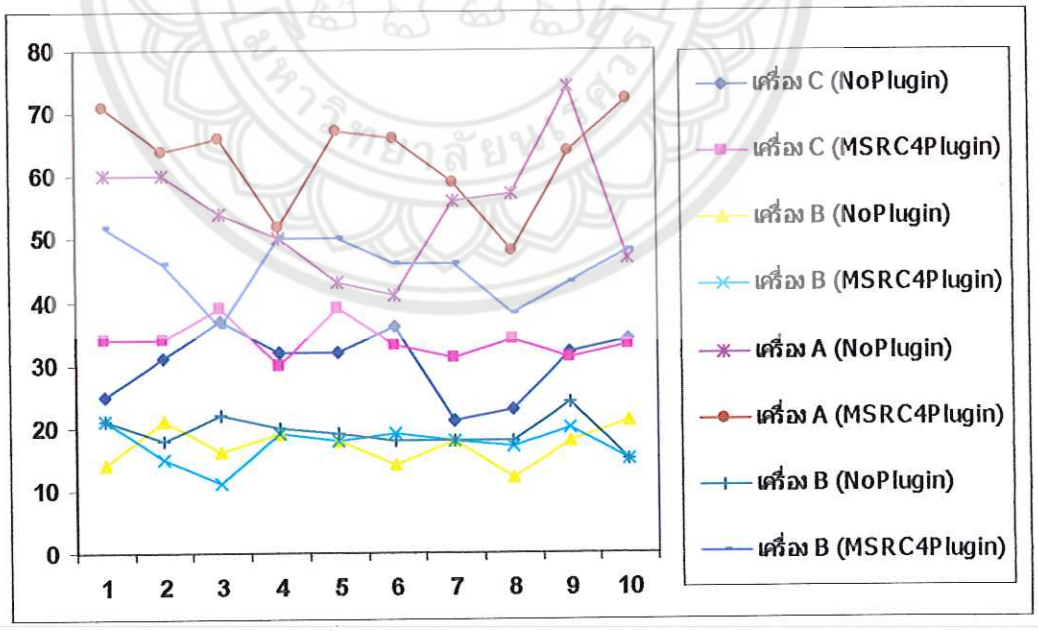


รูปที่ 5.9 เปิดโปรแกรม UltraVNC (Server)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้โปรแกรม UltraVNC (Viewer) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.6 แล้วนำมาแสดงผลในรูปของกราฟรูปที่ 5.2

ตารางที่ 5.12 เปิดโปรแกรม UltraVNC (Viewer)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	25	34	14	21	60	71	21	52
	31	34	21	15	60	64	18	46
	37	39	16	11	54	66	22	36
	32	30	19	19	50	52	20	50
	32	39	18	18	43	67	19	50
	36	33	14	19	41	66	18	46
	21	31	18	18	56	59	18	46
	23	34	12	17	57	48	18	38
	32	31	18	20	74	64	24	43
	34	33	21	15	47	72	15	48
Average	30.3	33.8	17.1	17.3	54.2	62.9	19.3	45.5
stdev	5.457919832	3.084008936	3.034981237	2.945806813	9.658617338	7.766738197	2.540778533	5.190803834
var	29.78888889	9.511111111	9.211111111	8.677777778	93.28888889	60.32222222	6.455555556	26.94444444
n	10	10	10	10	10	10	10	10
varh	2.978888889	0.951111111	0.921111111	0.867777778	9.328888889	6.032222222	0.645555556	2.694444444
v	17.22784611		39.55768819		31.13281199		293.1146229	
t(table)	1.74		1.684		1.697		1.645	
sqrt	1.98242276		1.33749351		3.919325339		1.827566688	
t'	3.449415603		2.252339071		6.651095		3.006347202	
AvgX-AvgY	-3.5		-0.2		-8.7		-26.2	

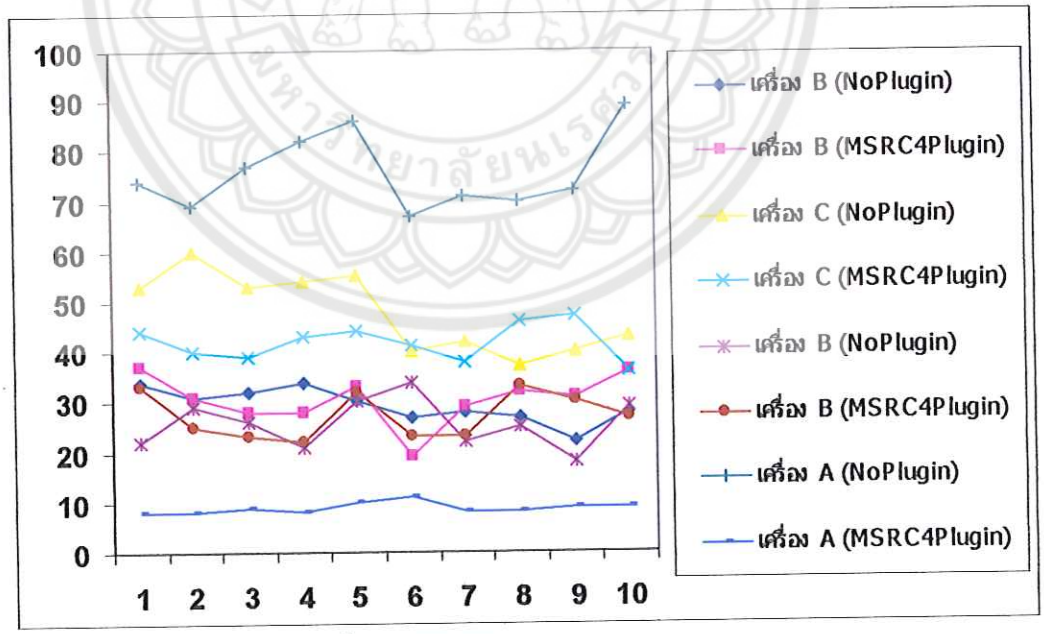


รูปที่ 5.10 เปิดโปรแกรม UltraVNC (Viewer)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้โปรแกรม Notepad (Server) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.7 แล้วนำมาแสดงผลในรูปของกราฟ รูปที่ 5.3

ตารางที่ 5.13 เปิดโปรแกรม Notepad (Server)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	34	37	53	44	22	33	74	8
	31	31	60	40	29	25	69	8
	32	28	53	39	26	23	77	9
	34	28	54	43	21	22	82	8
	30	33	55	44	30	32	86	10
	27	19	40	41	34	23	67	11
	28	29	42	38	22	23	71	8
	27	32	37	46	25	33	70	8
	22	31	40	47	18	30	72	9
	28	36	43	36	29	27	89	9
Average	29.3	30.4	47.7	41.8	25.6	27.1	75.7	8.8
stdev	3.683295626	5.03763613	8.083591061	3.583914682	4.926120851	4.508017549	7.572611468	1.032795559
var	13.56666667	25.37777778	65.34444444	12.84444444	24.26666667	20.32222222	57.34444444	1.066666667
n	10	10	10	10	10	10	10	10
varh	1.356666667	2.537777778	6.534444441	1.284444444	2.426666667	2.032222222	5.734444444	0.106666667
v	89.22913676		13.89943291		35.51406574		9.41406335	
t(table)	1.671		1.771		1.684		1.833	
sqrt	1.973434682		2.796227618		2.111608129		2.416839074	
t'	3.297609354		3.952119111		3.555943089		4.430066023	
AvgX-AvgY	-1.1		5.9		-1.5		66.9	

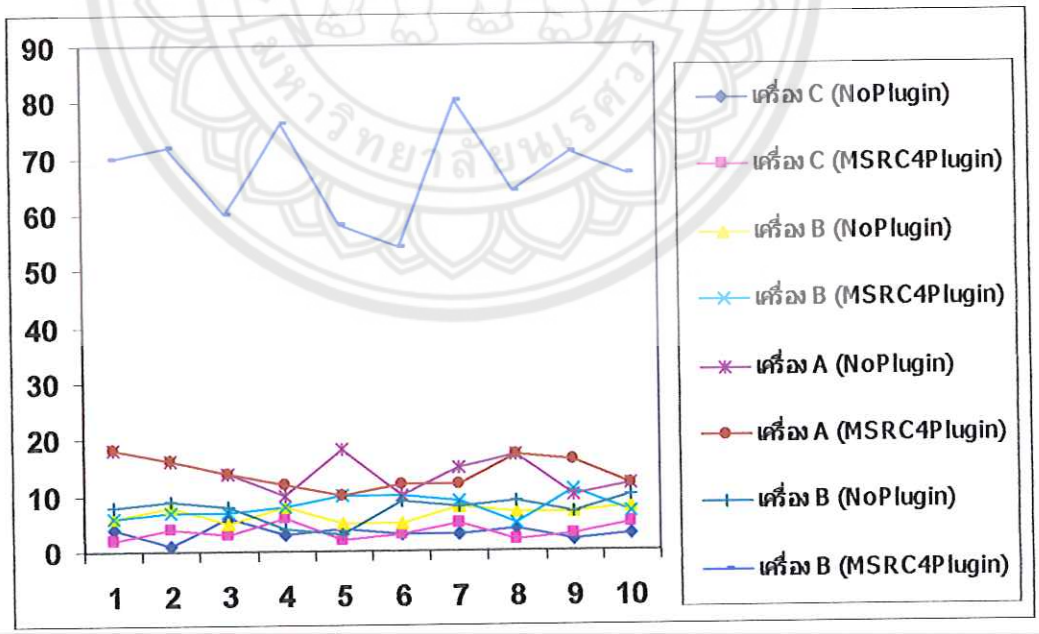


รูปที่ 5.11 เปิดโปรแกรม Notepad (Server)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้โปรแกรม Notepad (Viewer) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.8 แล้วนำมาแสดงผลในรูปของกราฟรูปที่ 5.4

ตารางที่ 5.14 เปิดโปรแกรม Notepad (Viewer)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	4	2	6	6	18	18	8	70
	1	4	8	7	16	16	9	72
	6	3	5	7	14	14	8	60
	3	6	8	8	10	12	4	76
	4	2	5	10	18	10	3	58
	3	3	5	10	10	12	9	54
	3	5	8	9	15	12	8	80
	4	2	7	5	17	17	9	64
	2	3	7	11	10	16	7	71
	3	5	8	7	12	12	10	67
Average	3.3	3.5	6.7	8	14	13.9	7.5	67.2
stdev	1.33749351	1.433720878	1.33749351	1.943650632	3.299831646	2.685351208	2.273030283	8.216514535
var	1.788888889	2.055555556	1.788888889	3.777777778	10.88888889	7.211111111	5.166666667	67.51111111
n	10	10	10	10	10	10	10	10
varh	0.178888889	0.205555556	0.178888889	0.377777778	1.088888889	0.721111111	0.516666667	6.751111111
v	48.80728241		104.5293509		28.44091195		2178.725631	
t(table)	1.684		1.658		1.701		1.645	
sqrt	0.620035841		0.746100976		1.345362409		2.695881633	
t'	1.044140357		-1.237035415		2.28846145		-4.434725286	
AvgX-AvgY	-0.2		-1.3		0.1		-59.7	

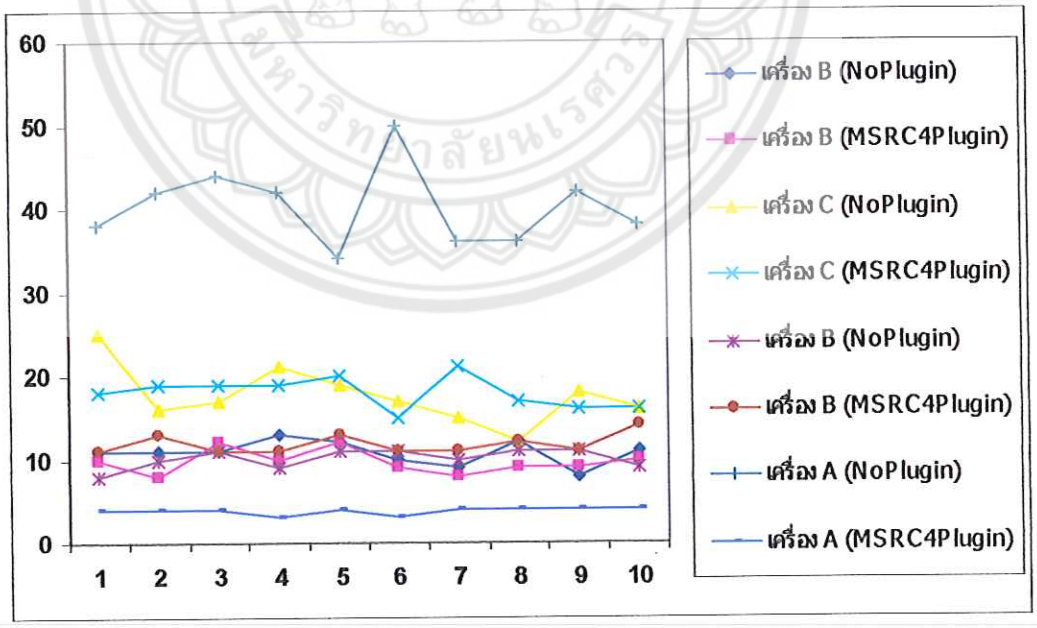


รูปที่ 5.12 เปิดโปรแกรม Notepad (Viewer)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้งานโปรแกรม Notepad (Server) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.9 แล้วนำมาแสดงผลในรูปของกราฟ รูปที่ 5.5

ตารางที่ 5.15 ใช้งานโปรแกรม Notepad (Server)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	11	10	25	18	8	11	38	4
	11	8	16	19	10	13	42	4
	11	12	17	19	11	11	44	4
	13	10	21	19	9	11	42	3
	12	12	19	20	11	13	34	4
	10	9	17	15	11	11	50	3
	9	8	15	21	10	11	36	4
	12	9	12	17	11	12	38	4
	8	9	18	16	11	11	42	4
	11	10	16	16	9	14	38	4
Average	10.8	9.7	17.6	18	10.1	11.8	40.2	3.8
stdev	1.475729578	1.418136492	3.533962208	1.943650632	1.100504939	1.135292424	4.756282395	0.421637021
var	2.177777778	2.011111111	12.48888889	3.777777778	1.211111111	1.288888889	22.62222222	0.177777778
n	10	10	10	10	10	10	10	10
varn	0.217777778	0.201111111	1.248888889	0.377777778	0.121111111	0.128888889	2.262222222	0.017777778
v	38.70075622		16.674288		44.87256484		9.173596072	
t(table)	1.684		1.746		1.684		1.833	
sqrt	0.647216261		1.275408431		0.5		1.509966887	
t'	1.089912185		2.226863121		0.842		2.767769304	
AvgX-AvgY	1.1		-0.4		-1.7		36.4	

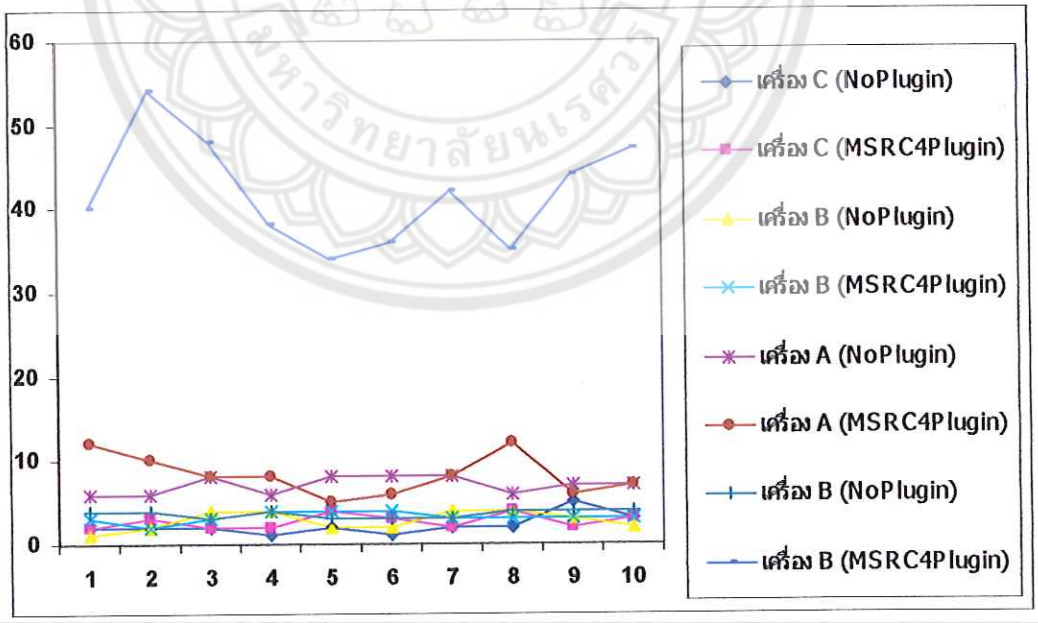


รูปที่ 5.13 ใช้งานโปรแกรม Notepad (Server)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้งานโปรแกรม Notepad (Viewer) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.10 แล้วนำมาแสดงผลในรูปของกราฟรูปที่ 5.6

ตารางที่ 5.16 ใช้งานโปรแกรม Notepad (Viewer)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	2	2	1	3	6	12	4	40
	2	3	2	2	6	10	4	54
	2	2	4	3	8	8	3	48
	1	2	4	4	6	8	4	38
	2	4	2	4	8	5	3	34
	1	3	2	4	8	6	3	36
	2	2	4	3	8	8	3	42
	2	4	4	3	6	12	4	35
	5	2	3	3	7	6	4	44
	3	3	2	3	7	7	4	47
Average	2.2	2.7	2.8	3.2	7	8.2	3.6	41.8
stdev	1.135292424	0.823272602	1.135292424	0.632455532	0.942809042	2.440400690	0.516397779	6.477310827
var	1.288888889	0.677777778	1.288888889	0.4	0.888888889	5.955555556	0.266666667	41.95555556
n	10	10	10	10	10	10	10	10
varh	0.128888889	0.067777778	0.128888889	0.04	0.088888889	0.595555556	0.026666667	4.195555556
v	23.61122321		16.88718469		650.2222442		275763.4891	
t(table)	1.714		1.746		1.645		1.645	
sqrt	0.443471157		0.410960934		0.827311576		2.054804668	
t'	0.760109562		-0.71753779		1.360927543		3.380153678	
AvgX-AvgY	-0.5		-0.4		-1.2		-38.2	

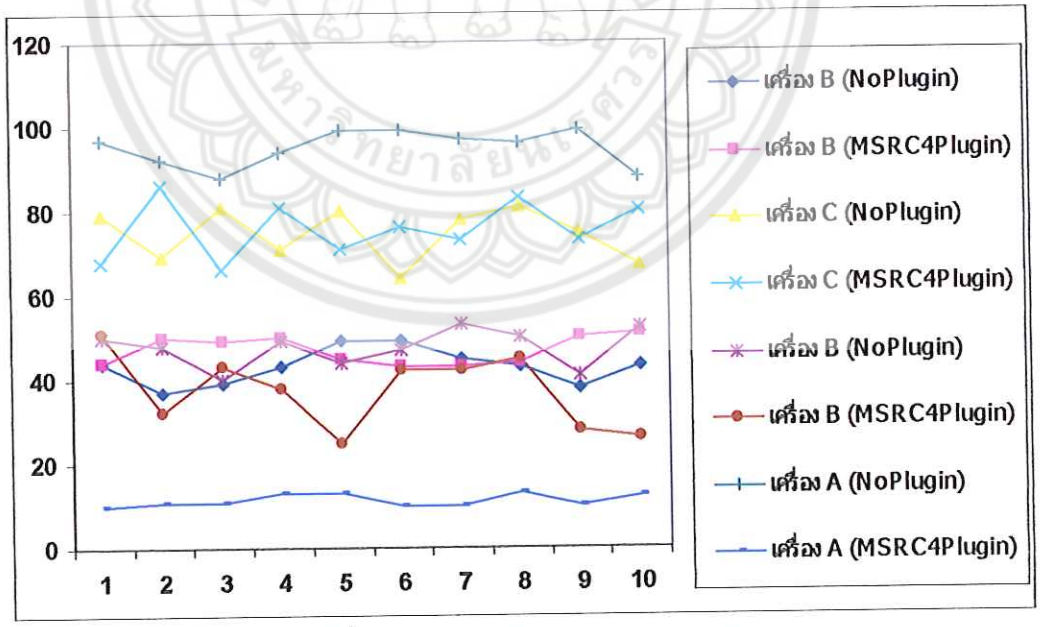


รูปที่ 5.14 ใช้งานโปรแกรม Notepad (Viewer)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้งานโปรแกรม Paint (Server) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.11 แล้วนำมาแสดงผลในรูปของกราฟรูปที่ 5.7

ตารางที่ 5.17 ใช้งานโปรแกรม Paint (Server)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	44	44	79	68	50	51	97	10
	37	50	69	86	48	32	92	11
	39	49	81	66	40	43	88	11
	43	50	71	81	49	38	94	13
	49	45	80	71	44	25	99	13
	49	43	64	76	47	42	99	10
	45	43	78	73	53	42	97	10
	43	44	81	83	50	45	96	13
	38	50	75	73	41	28	99	10
	43	51	67	80	52	26	88	12
Average	43	46.9	74.5	76.7	47.4	37.2	94.9	11.3
stdev	4.136557882	3.348299734	6.293735863	6.634087059	4.427188724	8.929352347	4.280446498	1.337493511
var	17.11111111	11.21111111	39.61111111	44.01111111	19.6	79.73333333	18.32222222	1.788888889
n	10	10	10	10	10	10	10	10
varh	1.711111111	1.121111111	3.961111111	4.401111111	1.96	7.973333333	1.832222222	0.178888889
v	28.25063101		48.78413283		286.3133566		11.25573604	
t(table)	1.701		1.684		1.645		1.796	
sqrt	1.682920742		2.891750719		3.151719108		1.418136492	
t'	2.862648181		4.869708211		5.184577932		2.54697314	
AvgX-AvgY	-3.9		-1.2		10.2		83.6	

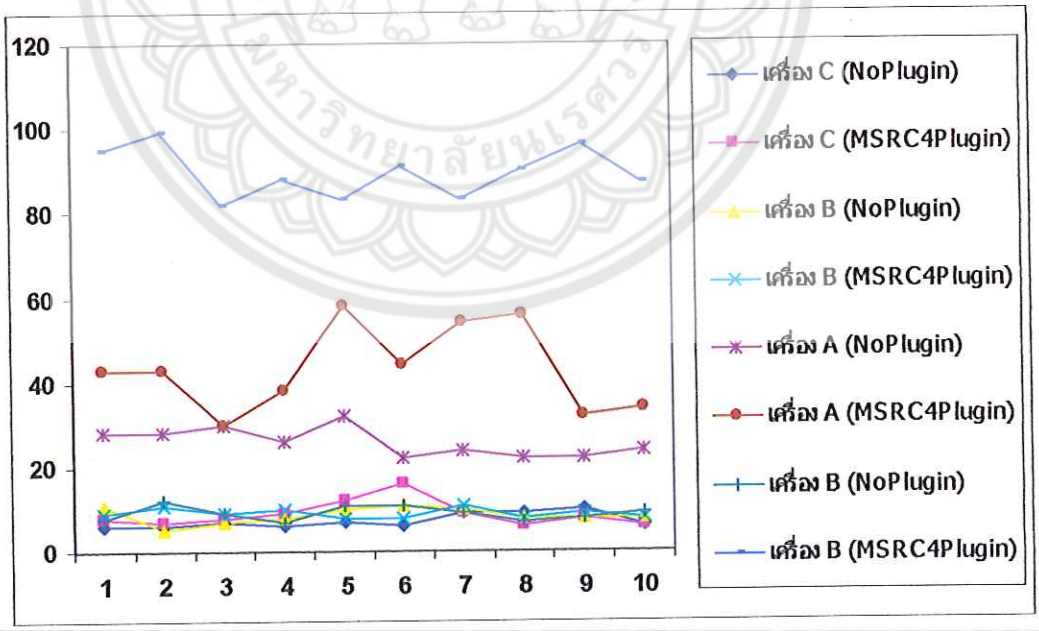


รูปที่ 5.15 ใช้งานโปรแกรม Paint (Server)

นำผลการทดลองมาวิเคราะห์ผลทางสถิติเมื่อเปิดใช้งานโปรแกรม Paint (Viewer) เปรียบเทียบระหว่างคอมพิวเตอร์เครื่องเดียวกันตามตารางที่ 5.12 แล้วนำมาแสดงผลในรูปของกราฟรูปที่ 5.8

ตารางที่ 5.18 ใช้งานโปรแกรม Paint (Viewer)

	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin	Noplugin	MSRC4Plugin
	6	8	11	9	28	43	8	95
	6	7	5	11	28	43	12	99
	7	8	7	9	30	30	9	82
	6	9	8	10	26	38	7	88
	7	12	10	8	32	58	11	83
	6	16	11	8	22	44	11	91
	9	9	10	11	24	54	9	83
	9	6	8	8	22	56	7	90
	10	8	8	9	22	32	8	96
	6	6	8	8	24	34	9	87
Average	7.2	8.9	8.6	9.1	25.8	43.2	9.1	89.4
stdev	1.549193338	3.034981237	1.897366596	1.197219	3.583914682	10.06423812	1.728840331	5.910442736
var	2.4	9.211111111	3.6	1.433333333	12.84444444	101.2888889	2.988888889	34.93333333
n	10	10	10	10	10	10	10	10
varh	0.24	0.921111111	0.36	0.143333333	1.284444444	10.12888889	0.298888889	3.493333333
v	255.5418734		19.50486836		875.8611912		1769.87347	
t(table)	1.645		1.729		1.645		1.645	
sqrt	1.077548656		0.709459888		3.378362522		1.947362889	
t'	1.772567543		1.226656147		5.557406349		3.203411953	
AvgX-AvgY	-1.7		-0.5		-17.4		-80.3	



รูปที่ 5.16 ใช้งานโปรแกรม Paint (Viewer)

จากผลการทดลองที่ได้เพื่อให้ข้อสรุปเป็นที่น่าเชื่อถือจึงนำผลการทดลองไปวิเคราะห์ผลทางสถิติ โดยใช้ทดสอบในหัวข้อของการทดสอบความมีนัยสำคัญค่าเฉลี่ยของตัวแปรสุ่มปกติ (กรณีประชากรสองชุด) เนื่องจากเป็นการทดลองเปรียบเทียบระหว่างการไม่ใช้ plug-in และใช้ plug-in ซึ่งเป็นการทดลองที่มีความแตกต่างกันทำให้ไม่ทราบค่าความแปรปรวน จึงศึกษาในกรณีไม่ทราบค่าที่แน่นอนของ σ_X^2 และ σ_Y^2 และไม่ทราบว่าแตกต่างกันหรือไม่ มีสูตรการคำนวณ ดังนี้

$$t' = \frac{\left(\overline{X} - \overline{Y}\right) - \left(\mu_X - \mu_Y\right)}{\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}}}$$

จะประมาณเป็นตัวแปรสถิติแบบ t ที่มีองศาความอิสระ υ โดยที่

$$\upsilon = \frac{\left(\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}\right)^2}{\frac{\left(S_X^2/n_X\right)^2}{n_X + 1} + \frac{\left(S_Y^2/n_Y\right)^2}{n_Y + 1}} - 2$$

ทั้งนี้การประมาณค่ามีความใกล้เคียงกันต่อเมื่อสมมติฐานเป็นจริง คือ $\mu_X = \mu_Y$ หรือกำหนดตัวสถิติ t เป็น

$$t' = \frac{\left(\overline{X} - \overline{Y}\right)}{\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}}} \sim t_\upsilon$$

ในกรณีการตัดสินใจด้วยการทดสอบความมีนัยสำคัญแบบด้านเดียว และกำหนดให้ α มีค่าเท่ากับ 0.05 จะได้วิธีการตัดสินใจ $d\left[\overline{X} - \overline{Y}\right]$ ว่า

$$d\left[\overline{X} - \overline{Y}\right] = \begin{cases} H_0 : \mu_X = \mu_Y ; \overline{X} - \overline{Y} \geq t_{\alpha;\upsilon}\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}} \\ H_1 : \mu_X < \mu_Y ; \overline{X} - \overline{Y} < t_{\alpha;\upsilon}\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}} \end{cases}$$

ผลการทดลองได้ว่าค่า $\overline{X} - \overline{Y}$ ซึ่งเป็นค่าความแตกต่างอันเนื่องจากการไม่ใช้ plug-in และใช้ plug-in มีอยู่ 2 กรณี คือ

1. มีค่าต่ำกว่าช่วงรีโปรดิวซิบิลิตี้ แสดงว่าประสิทธิภาพของ CPU มีนัยสำคัญเมื่อมีการใช้ plug-in หรือการใช้ plug-in มีผลกระทบต่อประสิทธิภาพของ CPU
2. มีค่าสูงกว่าช่วงรีโปรดิวซิบิลิตี้ แสดงว่าประสิทธิภาพของ CPU ไม่มีนัยสำคัญเมื่อมีการใช้ plug-in หรือการใช้ plug-in ไม่มีผลกระทบต่อประสิทธิภาพของ CPU

ตารางที่ 5.13 แสดงการเปรียบเทียบค่าระหว่าง t' และ $\overline{X}-\overline{Y}$

เปิดโปรแกรม UltraVNC (Server)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.684	1.796	1.708	1.812
t'	3.2368059	2.2843664	3.0190776	6.8005679
AvgX-AvgY	-0.7	-2.8	-1	40.1

เปิดโปรแกรม UltraVNC (Viewer)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.74	1.684	1.697	1.645
t'	3.4494156	2.2523391	6.6510951	3.0063472
AvgX-AvgY	-3.5	-0.2	-8.7	-26.2

เปิดโปรแกรม Notepad (Server)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.671	1.771	1.684	1.833
t'	3.2976094	4.9521191	3.5559481	4.430066
AvgX-AvgY	-1.1	5.9	-1.5	66.9

เปิดโปรแกรม Notepad (Viewer)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.684	1.658	1.701	1.645
t'	1.0441404	1.2370354	2.2884615	4.4347253
AvgX-AvgY	-0.2	-1.3	0.1	-59.7

ใช้งานโปรแกรม Notepad (Server)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.684	1.746	1.684	1.833
t'	1.0899122	2.2268631	0.842	2.7677693
AvgX-AvgY	1.1	-0.4	-1.7	36.4

ใช้งานโปรแกรม Notepad (Viewer)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.714	1.746	1.645	1.645
t'	0.7601096	0.7175378	1.3609275	3.3801537
AvgX-AvgY	-0.5	-0.4	-1.2	-38.2

ตารางที่ 5.13 แสดงการเปรียบเทียบค่าระหว่าง t' และ $\overline{X}-\overline{Y}$ (ต่อ)

ใช้งานโปรแกรม Paint (Server)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.701	1.684	1.645	1.796
t'	2.8626482	4.8697082	5.1845779	2.5469731
AvgX-AvgY	-3.9	-1.2	10.2	83.6

ใช้งานโปรแกรม Paint (Viewer)

	เครื่อง B	เครื่อง C	เครื่อง B	เครื่อง A
t(table)	1.645	1.729	1.645	1.645
t'	1.7725675	1.2266561	5.5574063	3.203412
AvgX-AvgY	-1.7	-0.5	-17.4	-80.3

ซึ่งผลการวิเคราะห์ทางสถิติส่วนใหญ่จะมีค่าต่ำกว่าช่วงรีโปรดิวซิบิลิตี้จึงแสดงว่าการใช้ plug-in น่าจะมีผลกระทบต่อประสิทธิภาพของ CPU

เนื่องจากคอมพิวเตอร์ที่นำมาใช้ในการทดลองมีทรัพยากรของเครื่องที่ต่างกัน รวมไปถึงโปรแกรมต่างๆด้วย จึงทำให้ค่าของ CPU Performance มีค่าที่ต่างออกไปจากกลุ่ม ซึ่งเป็นปัจจัยที่ส่งผลกระทบต่อการทดลองที่ทำให้มีค่าแตกต่างกัน

บทที่ 6
สรุปผล

โครงงานนี้แสดงให้เห็นถึงประสิทธิผลของการนำหลักการวิทยาการเข้ารหัสลับร่วมกับการควบคุมผ่านระบบเครือข่ายเพื่อเขียนโปรแกรมในการเข้ารหัสในการส่งข้อมูลของโปรแกรม UltraVNC และนำมาตรฐานในการเข้ารหัสลับและถอดรหัสลับมาศึกษา 3 มาตรฐาน คือ มาตรฐานรหัสลับ DES, AES และ RSA เพื่อให้เข้าใจถึงหลักการและวิธีในการเข้ารหัสและถอดรหัสลับอย่างแท้จริง และได้ทำการศึกษาและทดลองเขียนโปรแกรม UltraVNC โดยผ่าน Encryption plug-in MSAESPlugin และได้ทำการทดสอบประสิทธิภาพของ CPU ระหว่างเครื่องคอมพิวเตอร์ที่ไม่มี plug-in และเครื่องคอมพิวเตอร์ที่ใช้ plug-in ในการส่งผ่านข้อมูลระหว่างเครื่อง server และเครื่อง viewer แล้วนำผลการทดลองมาวิเคราะห์ผลทางสถิติเพื่อให้ได้คำตอบที่มีความน่าเชื่อถือ

ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรมนั้น กลุ่มผู้ศึกษาได้พัฒนาโดยใช้ Microsoft Visual C++ และ Microsoft Visual Studio.NET 2003 ในการพัฒนา เนื่องจากMicrosoft Visual C++ ซึ่งจำเป็นต้องใช้ Cryptographic Library ในการเข้ารหัส ซึ่ง Microsoft Visual Studio.NET 2003 มี API ที่ใช้ในการเขียนโปรแกรมที่เกี่ยวข้องกับการเข้ารหัส คือ Cryptographic API ที่ Microsoft ได้จัดเตรียมไว้ให้อยู่แล้วจึงได้ใช้ Cryptographic API ในการเขียน Plug-in ดังกล่าว

6.1 ปัญหาและแนวทางแก้ไข

เอกสารอ้างอิงที่ใช้ในการค้นคว้าในเรื่องของมาตรฐานที่ใช้ในการเข้ารหัสและถอดรหัสจากหนังสือภาษาไทยมีข้อผิดพลาดในส่วนของรายละเอียดของอัลกอริทึมมาก ทำให้เสียเวลาในการศึกษาและต้องค้นคว้าเพิ่มมากขึ้น จึงควรค้นคว้าจากหลายๆแหล่งแล้วนำผลจากการค้นคว้ามาเปรียบเทียบความถูกต้องของเนื้อหาก่อนที่จะทำการศึกษา และควรศึกษาจากแหล่งอ้างอิงที่มีความน่าเชื่อถือ เช่น โครงงานนี้เป็นวิทยาการที่พัฒนามาจากต่างประเทศจึงควรศึกษาบทความหรือเอกสารต่างๆที่มาจากต่างประเทศ เพราะหนังสือหรือเอกสารภาษาไทยเป็นการแปลหรือการศึกษาจากต้นฉบับ อาจจะมีข้อบกพร่องในส่วนของเนื้อหาที่ผิดเพี้ยนไป

ในการพัฒนาโปรแกรม UltraVNC โดยผ่าน Encryption plug-in คือ MSAESPlugin ขึ้นมาต้องศึกษาจากโปรแกรมเดิมที่มีอยู่แล้ว ซึ่งมี document ที่ไม่สามารถอธิบายโปรแกรมทั้งหมดให้สามารถเข้าใจได้ง่าย ทำให้ต้องเสียเวลานานพอสมควรในการศึกษาโปรแกรม

อัลกอริทึมที่ใช้ในการศึกษา คือ อัลกอริทึม DES, AES และ RSA เป็นการส่งข้อมูลแบบ block cipher แต่การเข้ารหัสแบบ RC4 ที่ต้องการนำมาพัฒนาเป็นการส่งข้อมูลแบบ stream Cipher

จึงทำให้การที่จะนำเอา Code ของการเข้ารหัสแบบ RC4 มาแก้ไขให้เป็นการเข้ารหัสแบบ AES นั้น จะต้องเสีย Overhead ในการที่จะเปลี่ยนการส่งข้อมูลแบบ Stream Cipher มาเป็น Block Cipher มากเกินไป

6.2 การอภิปรายผล

จากการศึกษาและเขียนโปรแกรมแสดงการทำงานของอัลกอริทึม 3 อัลกอริทึม คือ อัลกอริทึม DES, AES และ RSA สามารถสรุปได้ว่า ในด้านของความปลอดภัยในการส่งข้อมูล อัลกอริทึม RSA ปลอดภัยที่สุด รองลงมาก็คือ อัลกอริทึม AES และท้ายสุด คือ อัลกอริทึม DES เนื่องจากอัลกอริทึม RSA มีความซับซ้อนในการแยกตัวประกอบที่มีกำลังมาก ในด้านของความเร็วในการส่งข้อมูล พบว่าอัลกอริทึม DES สามารถส่งข้อมูลได้เร็วที่สุด เนื่องจากขนาดกุญแจที่ใช้มีขนาดเพียง 64 บิต และอัลกอริทึมไม่มีความซับซ้อนเมื่อเปรียบเทียบกับอีก 2 อัลกอริทึม รองลงมา คือ อัลกอริทึม AES และท้ายสุด คือ อัลกอริทึม RSA

จากการศึกษาและพัฒนา plug-in ของโปรแกรม UltraVNC ทำการพัฒนาโดยใช้ Microsoft Visual Studio.NET 2003 โดยพัฒนา plug-in ในส่วนของการเข้ารหัสแบบ RC4 หรือตัว MSRC4plugin.dsm ให้เป็นการเข้ารหัสข้อมูลแบบ AES แต่เนื่องจากการศึกษาในภายหลังพบว่า การเข้ารหัสข้อมูลแบบ RC4 มีการส่งข้อมูลแบ stream cipher แต่ตัวที่ต้องการพัฒนาต่อมา คือ การเข้ารหัสข้อมูลแบบ AES มีการส่งข้อมูลแบบ block cipher ซึ่งไม่สามารถเปลี่ยนรูปแบการส่งข้อมูลจาก stream cipher เป็น block cipher ได้ เนื่องจากทำให้เกิดปัญหา overhead ขึ้น ประกอบกับเวลาที่ใช้ในการดเนินการโครงงานเหลือน้อยจึงไม่สามารถพัฒนา plug-in ให้เสร็จสมบูรณ์ได้

6.3 สรุปผลการทดลอง

จากการทดลองเพื่อทดสอบประสิทธิภาพของ CPU เมื่อไม่มีการใช้ plug-in มีการใช้ plug-in (MSRC4Plugin.dsm) โดยบันทึกค่า CPU Performance ที่มีค่าสูงสุดจากการใช้งาน 4 กรณี คือ เปิดใช้งานโปรแกรม UltraVNC, เปิดใช้งานโปรแกรม Notepad, ใช้งานโปรแกรม Notepad และใช้งานโปรแกรม Paint แล้วนำผลการทดลองมาวิเคราะห์ทางสถิติ เพื่อให้ได้ข้อสรุปที่น่าเชื่อถือ โดยทำการวิเคราะห์ในเรื่อง กรทดสอบความมีนัยสำคัญค่าเฉลี่ยของตัวแปรสุ่มปกติ(กรณีประชากรสองชุด) และเนื่องจากเป็นการทดลองระหว่างไม่ใช้ plug-in และใช้ plug-in ทำให้ไม่ทราบค่าความแปรปรวน จึงศึกษาในกรณีไม่ทราบค่าที่แน่นอนของ σ_X^2 และ σ_Y^2 และไม่ทราบว่าแตกต่างกันหรือไม่ และได้กำหนดให้ค่า $\alpha = 0.05$ ซึ่งจากการวิเคราะห์ผลการทดลองนี้ด้วยวิธีทางสถิติพบว่า plug-in น่าจะมีผลกระทบต่อประสิทธิภาพของ CPU

6.4 แนวทางในการพัฒนา

อาจเปลี่ยนอัลกอริทึมที่จะนำมาพัฒนาในการเข้ารหัสและถอดรหัส เช่น FEAL, REDOC, LOKI, KHUFU, KHAFRE, RC2, RABIN, ElGamal เป็นต้น

เอกสารอ้างอิง

[1] ลัญฉกร วุฒิสิทธิกุลกิจ.**วิทยาการรหัสลับเบื้องต้น**.สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2548.

[2] "เทคโนโลยีในการรักษาความปลอดภัยมีอะไรบ้าง." [Online]. Available: http://www.ecommerce.or.th/project/e-guide/encryption.html. 2548.

[3] "การเข้ารหัสข้อมูล." [Online]. Available: http://www.ku.ac.th/e-magazine/august44/it/encryp.html. 2544.

[4] จักรกฤษณ์ แร่ทอง. "เทคโนโลยีการเข้ารหัสข้อมูล." [Online]. Available: http://www.nextproject.net/ArticleDetail.aspx?ProID=48. 2547.

[5] "การสื่อสารข้อมูลคอมพิวเตอร์และระบบเครือข่าย." [Online]. Available: http://dusithost.dusit.ac.th/~phitsanulok/e-learning/Ch96.html. 2548.

[6] Wikipedia. "Finite field arithmetic." [Online]. Available: http://en.wikipedia.org/wiki/Finite_field_arithmetic#Notation. 2006.

[7] บรรจง ทะรังษี. [Online]. Available: http://thaicert.nectec.or.th/paper/authen/authentication_guide.php. 2547.

[8] "About RealVNC."[Online] Available: http://www.realvnc.com 2002.

[9] "Announcing the ADVANCED ENCRYPTION STANDARD (AES)." 2001.

[10] "มาช่วยกันสรุปเนื้อหาวิชา Computer Security กันหน่อยนะ." [Online]. Available: http://www.buscience15.com/webboard/index.php?PHPSESSID=3e927eefd4f96c50c3a02e65c503b46a&topic=492.msg5247. 2549.

[11] นางสาวกาญจนา รัตนตรัยรักษ์ และ นางสาวปวีณา วันชัย. "Data Encryption Standard."[Online]. Available:http://freehp.kku.ac.th/seminar/2543/sec02/group08/des.htm. 2006.

[12] กิติศักดิ์ พลอยพานิชเจริญ.**สถิติสำหรับงานวิศวกรรม เล่ม 1**.สมาคมส่งเสริมเทคโนโลยีไทย-ญี่ปุ่น, 2540.

[13] กิติศักดิ์ พลอยพานิชเจริญ.**สถิติสำหรับงานวิศวกรรม เล่ม 2**.สมาคมส่งเสริมเทคโนโลยีไทย-ญี่ปุ่น, 2540.

ภาคผนวก

ภาคผนวก ก

มาตรฐานรหัสลับ DES (Data Encryption Standard)

กำหนดให้ Key = Abnormal และ Input = Generate

1. การจัดเตรียมกุญแจ

1.1 การจัดเตรียมกุญแจโดยกำหนดให้ค่ากุญแจลับมีค่าเป็น

Key = Abnormal

= 41 42 6E 6F 72 6D 61 6C

สามารถเขียนในรูปตัวเลขฐานสองได้เป็น

Key = (0100 0001 0100 0010 0110 1110 0110 1111

0111 0010 0110 1101 0110 0001 0110 1100)

1.2 หาค่าของ C_0 และ D_0 โดยจะได้จากการเลือกบิตจากกุญแจลับ K สำหรับตำแหน่งที่เลือกจะขึ้นอยู่กับกล่องสลับลำดับ PC-1 ในตารางที่ 2.1

จะได้ C_0 = (0000 0000 1111 1111 1111 1100 0001)

D_0= (0001 1110 1010 1100 1010 1100 0000)

1.3 ใช้ตารางที่ 2.2 เพื่อพิจารณาหาค่า C_1, D_1 ต่อไปโดยรอบแรกจะเลื่อนตำแหน่งบิตไปทางซ้ายจำนวน 1 ตำแหน่งตามที่ระบุในตารางที่ 2.2 (รอบที่ 1 เลื่อน 1บิต)

จะได้ C_1 = (0000 0001 1111 1111 1111 1000 0010)

D_1= (0011 1101 0101 1001 0101 1000 0000)

1.4 หาค่า C_2, D_2 ทำได้โดยนำค่า C_1, D_1 ไปผ่านวงจร LS โดยเลื่อนบิตไปทางซ้ายแบบวนกลับ 1 ตำแหน่ง ตามที่ระบุในตารางที่ 2.2 (รอบที่ 2 เลื่อน 1 บิต)

จะได้ C_2 = (0000 0011 1111 1111 1111 0000 0100)

D_2= (0111 1010 1011 0010 1011 0000 0000)

สำหรับค่า (C_3, D_3), (C_4, D_4), ...และ (C_{16}, D_{16}) ก็มีลักษณะการคำนวณเหมือนกัน โดยเลื่อนบิตเปรียบเทียบกับตารางที่ 2.2 จะได้

C_3 = (0000 1111 1111 1111 1100 0001 0000)

D_3= (1110 1010 1100 1010 1100 0000 0001)

C_4 = (0011 1111 1111 1111 0000 0100 0000)

D_4= (1010 1011 0010 1011 0000 0000 0111)

C_5 = (1111 1111 1111 1100 0001 0000 0000)

D_5= (1010 1100 1010 1100 0000 0001 1110)

C_6 = (1111 1111 1111 0000 0100 0000 0011)

D_6 = (1011 0010 1011 0000 0000 0111 1010)

C_7 = (1111 1111 1100 0001 0000 0000 1111)

D_7 = (1100 1010 1100 0000 0001 1110 1010)

C_8 = (1111 1111 0000 0100 0000 0011 1111)

D_8 = (0010 1011 0000 0000 0111 1010 1011)

C_9 = (1111 1110 0000 1000 0000 0111 1111)

D_9 = (0101 0110 0000 0000 1111 0101 0110)

C_{10} = (1111 1000 0010 0000 0001 1111 1111)

D_{10} = (0101 1000 0000 0011 1101 0101 1001)

C_{11} = (1110 0000 1000 0000 0111 1111 1111)

D_{11} = (0110 0000 0000 1111 0101 0110 0101)

C_{12} = (1000 0010 0000 0001 1111 1111 1111)

D_{12} = (1000 0000 0011 1101 0101 1001 0101)

C_{13} = (0000 1000 0000 0111 1111 1111 1110)

D_{13} = (0000 0000 1111 0101 0110 0101 0110)

C_{14} = (0010 0000 0001 1111 1111 1111 1000)

D_{14} = (0000 0011 1101 0101 1001 0101 1000)

C_{15} = (1000 0000 0111 1111 1111 1110 0000)

D_{15} = (0000 1111 0101 0110 0101 0110 0000)

C_{16} = (0000 0000 1111 1111 1111 1100 0001)

D_{16} = (0001 1110 1010 1100 1010 1100 0000)

1.5 นำค่าของ C และ D ทั้ง 16 ชุดใช้ในการหาค่า K_1 ถึง K_{16} โดยป้อนค่า C และ D แต่ละชุดเข้าไปในวงจรสลับลำดับ PC-2 ตามตารางที่ 2.3 จะได้ชุดกุญแจ K_1 - K_{16} ที่ใช้ประกอบในการเข้ารหัสแต่ละรอบ เช่น K_1 หาได้จาก C_1 และ D_1 ซึ่งเมื่อพิจารณาจากตารางที่ 2.3 แล้วจะได้

K_1= (1110 0000 1011 0110 0110 1110 1010 0001 0011 1100 1010 0101)

และ K_2 หาได้จาก C_2 และ D_2 โดยอาศัยจากตารางที่ 2.3 จะได้

K_2= (1110 0000 1001 0110 1111 0110 0011 1011 0111 0010 0000 0001)

สำหรับค่า (K_3), (K_4), ...และ (K_{16}) ก็คำนวณเหมือนกัน โดยเปรียบเทียบกับตารางที่ 2.3

K_3 = (1111 0100 1101 0010 0111 0010 1011 0010 0110 0001 0010 0010)

K_4= (1010 0110 1101 0011 0111 0010 1010 0100 0010 1011 0000 0110)

K_5 = (1010 1110 0101 0011 0101 0111 1111 0100 0010 0010 1101 0010)

K_6 = (0010 1111 0101 0011 0101 1001 0111 0101 1000 0010 0100 1011)

K_7 = (0000 1111 0101 0001 1111 1001 0001 0110 1011 0100 0100 1010)

K_8 = (1001 1111 0100 1001 1101 1001 0010 1100 1011 0101 0110 0100)

K_9 = (0001 1111 0100 1001 1001 1011 0100 1110 0101 0000 1010 1001)

K_{10} = (0011 1111 0010 1001 1000 1101 0100 0010 0111 1001 0110 1001)

K_{11} = (0001 1011 0010 1100 1000 1101 1010 0010 1001 1001 0011 1000)

K_{12} = (0101 1001 0010 1100 1011 1100 1100 0001 0001 1111 0011 0010)

K_{13} = (1101 0100 1010 1100 1010 1100 0101 1101 0000 1010 0011 1000)

K_{14} = (1101 0010 1010 1110 0010 0110 0101 0001 0101 1000 0101 1100)

K_{15} = (1110 1000 1011 1110 0010 0110 0000 0001 1011 0000 1011 1100)

K_{16} = (1110 0001 1011 0110 0010 0110 1001 1000 0110 0110 1001 1001)

2. การเข้ารหัสแต่ละรอบ (Encipherment)

2.1 ข้อความต้นฉบับมีค่าเป็น

Input = Generate

= 47 65 6E 65 72 61 74 65

หรือเขียนในรูปตัวเลขฐานสองได้เป็น

Input = (0100 0111 0110 0101 0110 1110 0100 0101
0111 0010 0110 0001 0111 0100 0110 0101)

2.2 ป้อนข้อความต้นฉบับเข้าสู่กล่องสลับลำดับ Initial Permutation ตามตารางที่ 2.4 ข้อความจะถูกแยกบิตออกเป็น 2 บล็อกเท่าๆกัน คือ L_0 และ R_0 โดยแต่ละบล็อกจะมี 32 บิต

L_0 = (1111 1111 0101 0000 1100 1111 1010 1011)

R_0 = (0000 0000 1111 1110 0000 0100 0001 0101)

2.3 การเข้ารหัสลับแบบ DES ค่า L_0 ไม่มีการประมวลผล แต่ R_0 จะถูกนำไปผ่านกระบวนการหลายกระบวนการ และนำ K_1 มาใช้ในการเข้ารหัสด้วย ดังนั้น R_0 ที่มีขนาด32 บิตจะถูกเพิ่มเป็น 42 บิต โดยใช้ฟังก์ชั่น $E(R_0)$ ตามตารางที่ 2.5 จะได้

$E(R_0)$ = (100000 000001 011111 111100 000000 001000 000010 101010)

2.4. นำค่า $E(R_0)$ ไปบวกแบบ modulo กับกุญแจ K_1 ซึ่งมีขนาด 48 บิตเท่ากัน ผลลัพธ์ที่ได้จะเก็บในตัวแปร Γ หรือเรียกว่า The key-dependent function ตามสมการ

$$\Gamma_1 = E(R_0) \otimes K_1$$

= (100000 000001 011111 111100 000000 001000 000010 101010) $\otimes$

(111000 001011 011001 101110 101000 010011 110010 100101)

= (011000 001010 000110 010010 101000 011011 110000 001111)

2.5 เมื่อได้ค่า Γ_1 ขนาด 48 บิตแล้ว ให้แบ่งออกเป็น 8 กลุ่ม แต่ละกลุ่มจะมีขนาด 6 บิต เพื่อนำทั้ง 8 กลุ่มไปป้อนให้กับกล่องแทนค่า S-Box ที่มีทั้งหมด 8 ชุด ซึ่งกล่องแทนค่าแต่ละชุดจะให้ผลลัพธ์เป็นบิตที่มีขนาดลดลงเหลือ 4 บิต ดังนั้นจำนวนบิตจะลดลงเหลือแค่ 36 บิต

การทำงานของ S-Box คือ การนำบิตแรกและบิตสุดท้ายมาใช้ระบุหมายเลขแถวของตาราง S-Box ตามตารางที่ 2.6 และใช้ค่าของบิตที่ 2 ถึงบิตที่ 5 ระบุหมายเลขคอลัมน์ของตาราง S-Box ผลลัพธ์ที่ได้ คือ ค่าที่บรรจุอยู่ในตาราง ณ ตำแหน่งที่ระบุ โดยตัวเลขที่ได้จะมีขนาดอยู่ระหว่าง 0 – 15 เมื่อเขียนเป็นตัวเลขฐานสองแล้วจะมีขนาด 4 บิตพอดี

นำค่า Γ_1 มาคำนวณหา S_1 ถึง S_8 จากตารางที่ 2.6 ซึ่งค่าแต่ละค่ามาจากกล่องแทนค่า S-Box แต่ละชุด ได้ผลดังนี้

$S_1(00, 1100) = S_1(0, 12) = 5 = 0101$

$S_2(00, 0101) = S_2(0, 5) = 11 = 1011$

$S_3(00, 0011) = S_3(0, 3) = 14 = 1110$

$S_4(00, 1001) = S_4(0, 9) = 2 = 0010$

$S_5(10, 0100) = S_5(2, 4) = 10 = 1100$

$S_6(01, 1101) = S_6(1, 13) = 11 = 1101$

$S_7(10, 1000) = S_7(2, 8) = 10 = 1100$

$S_8(01, 0111) = S_8(1, 7) = 4 = 0100$

2.6 นำค่าที่ได้จาก S-Box แต่ละชุดมารวมกันจะได้ผลลัพธ์เป็นค่าของ B_1 เท่ากับ

B_1 = (0101 1011 1110 0010 1010 1011 1010 0100)

2.7 นำค่า B_1 ไปผ่านกระบวนการ $P(B_1)$ ซึ่งเป็นกล่องสลับตำแหน่ง (Permutation Function) โดยจะมีการสลับตำแหน่งแตกต่างจากเดิม ตามในตารางที่ 2.7 จะได้

$P(B_1)$ = (0101 0001 0110 1001 1110 0101 1010 0111)

2.8 หาค่า R_1 จาก $P(B_1)$ และ L_0 ตามสมการ

$$R_1 = P(B_1) \otimes L_0$$

= (0101 0001 0110 1001 1110 0101 1010 0111) $\otimes$

(1111 1111 0101 0000 1100 1111 1010 1011)

= (1010 1110 0011 1001 0010 1010 0000 1100)

2.9 R_1 เป็นผลลัพธ์ของกระบวนการเข้ารหัสลับของชุดบิตทางด้านขวามือในการทำงานรอบที่ 1 และ L_1 เป็นผลลัพธ์ของกระบวนการเข้ารหัสลับของชุดบิตทางด้านซ้ายมือกลับมีความซับซ้อนน้อยกว่ามาก ดังนั้นให้นำค่าของ R_0 มาใช้เป็น L_1 ได้เลย จะได้

$L_1 = R_0 = (0000\ \ 0000\ \ 1111\ \ 1110\ \ 0000\ \ 0100\ \ 0001\ \ 0101)$

2.10 รายละเอียดขั้นตอนการทำงานที่ได้อธิบายมาเป็นการเข้ารหัสลับในรอบที่ 1 สำหรับการเข้ารหัสที่เหลืออีก 15 รอบมีรูปแบบการทำงานเหมือนเดิม แต่ชุดกุญแจที่ใช้ในแต่ละรอบจะแตกต่างกัน ผลที่ได้ทั้ง 16 ขั้นตอน คือ L_{16} และ R_{16}

2.11 การเข้ารหัสลับในรอบที่ 2

$E(R_1) = (010101\ \ 011100\ \ 000111\ \ 110010\ \ 100101\ \ 010100\ \ 000001\ \ 011001)$

$K_2 = (111000\ \ 001001\ \ 011011\ \ 110110\ \ 001110\ \ 110111\ \ 001000\ \ 000001)$

$\Gamma_2 = (101101\ \ 010101\ \ 011100\ \ 000100\ \ 101011\ \ 100011\ \ 001001\ \ 011000)$

$S_1(11, 0110) = S_1(3, 6) = 1 = 0001$

$S_2(01, 1010) = S_2(1, 10) = 1 = 0001$

$S_3(00, 1110) = S_3(0, 14) = 2 = 0010$

$S_4(00, 0010) = S_4(0, 2) = 14 = 1110$

$S_5(11, 0101) = S_5(3, 5) = 14 = 1110$

$S_6(11, 0001) = S_6(3, 1) = 3 = 0011$

$S_7(01, 0100) = S_7(1, 4) = 4 = 0100$

$S_8(00, 1100) = S_8(0, 12) = 5 = 0101$

$B_2 = (0001\ \ 0001\ \ 0010\ \ 1110\ \ 1110\ \ 0011\ \ 0100\ \ 0101)$

$P(B_2) = (0000\ \ 0001\ \ 0111\ \ 0100\ \ 0111\ \ 1000\ \ 1110\ \ 0110)$

$L_1 = (0000\ \ 0000\ \ 1111\ \ 1110\ \ 0000\ \ 0100\ \ 0001\ \ 0101)$

$R_2 = (0000\ \ 0001\ \ 1000\ \ 1010\ \ 0111\ \ 1100\ \ 1111\ \ 0011)$

$L_2 = R_1 = (1010\ \ 1110\ \ 0011\ \ 1001\ \ 0010\ \ 1010\ \ 0000\ \ 1100)$

2.12 การเข้ารหัสลับในรอบที่ 3

$E(R_2) = (100000\ \ 000011\ \ 110001\ \ 010100\ \ 001111\ \ 111001\ \ 011110\ \ 100110)$

$K_3 = (111101\ \ 001101\ \ 001001\ \ 110010\ \ 101100\ \ 100110\ \ 000100\ \ 100010)$

$\Gamma_3 = (011101\ \ 001110\ \ 111000\ \ 100110\ \ 100011\ \ 011111\ \ 011010\ \ 000100)$

$S_1(01, 1110) = S_1(1, 14) = 3 = 0011$

$S_2(00, 0111) = S_2(0, 7) = 4 = 0100$

$S_3(10, 1100) = S_3(2, 12) = 5 = 0101$

$S_4(10, 0011) = S_4(2, 3) = 0 = 0000$

$S_5(11, 0001) = S_5(3, 1) = 8 = 1000$

$S_6(01, 1111) = S_6(1, 15) = 8 = 1000$

$S_7(00, 1101) = S_7(0, 13) = 10 = 1100$

$S_8(00, 0010) = S_8(0, 2) = 8 = 1000$

$B_3 = (0011\ 0100\ 0101\ 0000\ 1000\ 1000\ 1100\ 1000)$

$P(B_3) = (0001\ 1101\ 0000\ 0001\ 0000\ 0110\ 0001\ 0011)$

$L_2 = (1010\ 1110\ 0011\ 1001\ 0010\ 1010\ 0000\ 1100)$

$R_3 = (1011\ 0011\ 0011\ 1000\ 0010\ 1100\ 0001\ 1111)$

$L_3 = R_2 = (0000\ 0001\ 1000\ 1010\ 0111\ 1100\ 1111\ 0011)$

2.13 การเข้ารหัสลับในรอบที่ 4

$E(R_3) = (110110\ 100110\ 100111\ 110000\ 000101\ 011000\ 000011\ 111111)$

$K_4 = (101001\ 101101\ 001101\ 110010\ 101001\ 000010\ 101100\ 000110)$

$\Gamma_4 = (011111\ 001011\ 101010\ 000010\ 101100\ 011010\ 101111\ 111001)$

$S_1(01, 1111) = S_1(1, 15) = 8 = 1000$

$S_2(01, 0101) = S_2(1, 5) = 2 = 0010$

$S_3(10, 0101) = S_3(2, 5) = 15 = 1111$

$S_4(00, 0001) = S_4(0, 1) = 13 = 1101$

$S_5(10, 0110) = S_5(2, 6) = 7 = 0111$

$S_6(00, 1101) = S_6(0, 13) = 7 = 0111$

$S_7(11, 0111) = S_7(3, 7) = 7 = 0111$

$S_8(11, 1100) = S_8(3, 12) = 3 = 0011$

$B_4 = (1000\ 0010\ 1111\ 1101\ 0111\ 0111\ 0111\ 0011)$

$P(B_4) = (1110\ 0110\ 1011\ 0111\ 0011\ 1101\ 1100\ 1100)$

$L_3 = (0000\ 0001\ 1000\ 1010\ 0111\ 1100\ 1111\ 0011)$

$R_4 = (1110\ 0111\ 0011\ 1101\ 0100\ 0001\ 0011\ 1111)$

$L_4 = R_3 = (1011\ 0011\ 0011\ 1000\ 0010\ 1100\ 0001\ 1111)$

2.14 การเข้ารหัสลับในรอบที่ 5

$E(R_4) = (111100\ 001110\ 100111\ 111010\ 101000\ 000010\ 100111\ 111111)$

$K_5 = (101011\ 100101\ 001101\ 010111\ 111101\ 000010\ 001011\ 010010)$

$\Gamma_5 = (010111\ 101011\ 101010\ 101101\ 010101\ 000000\ 101100\ 101101)$

$S_1(01, 1011) = S_1(1, 11) = 11 = 1011$

$S_2(11, 0101) = S_2(3, 5) = 15 = 1111$

$S_3(10, 0101) = S_3(2, 5) = 15 = 1111$

$S_4(11, 0110) = S_4(3, 6) = 13 = 1101$

$S_5(01, 1010) = S_5(1, 10) = 15 = 1111$

$S_6(00, 0000) = S_6(0, 0) = 12 = 1100$

$S_7(10, 0110) = S_7(2, 6) = 7 = 0111$

$S_8(11, 0110) = S_8(3, 6) = 8 = 1000$

$B_5 = (1011\ 1111\ 1111\ 1101\ 1111\ 1100\ 0111\ 1000)$

$P(B_5) = (1111\ 1111\ 1001\ 1101\ 0101\ 0111\ 1001\ 1110)$

$L_4 = (1011\ 0011\ 0011\ 1000\ 0010\ 1100\ 0001\ 1111)$

$R_5 = (0100\ 1100\ 1010\ 0101\ 0111\ 1011\ 1100\ 0001)$

$L_5 = R_4 = (1110\ 0111\ 0011\ 1101\ 0100\ 0001\ 0011\ 1111)$

2.15 การเข้ารหัสลับในรอบที่ 6

$E(R_5) = (101001\ 011001\ 010100\ 001010\ 101111\ 110111\ 111000\ 000010)$

$K_6 = (001011\ 110101\ 001101\ 011001\ 011101\ 011000\ 001001\ 001011)$

$\Gamma_6 = (100010\ 101100\ 011001\ 010011\ 110010\ 101111\ 110001\ 001001)$

$S_1(10, 0001) = S_1(2, 1) = 1 = 0001$

$S_2(10, 0110) = S_2(2, 6) = 13 = 1101$

$S_3(01, 1100) = S_3(1, 12) = 12 = 1100$

$S_4(01, 1001) = S_4(1, 9) = 7 = 0111$

$S_5(10, 1001) = S_5(2, 9) = 9 = 1001$

$S_6(11, 0111) = S_6(3, 7) = 10 = 1010$

$S_7(11, 1000) = S_7(3, 8) = 9 = 1001$

$S_8(01, 0100) = S_8(1, 4) = 10 = 1010$

$B_6 = (0001\ 1101\ 1100\ 0111\ 1001\ 1010\ 1001\ 1010)$

$P(B_6) = (1011\ 1011\ 0110\ 1011\ 0101\ 0001\ 0001\ 0011)$

$L_5 = (1110\ 0111\ 0011\ 1101\ 0100\ 0001\ 0011\ 1111)$

$R_6 = (0101\ 1100\ 0101\ 0110\ 0001\ 0000\ 0010\ 1100)$

$L_6 = R_5 = (0100\ 1100\ 1010\ 0101\ 0111\ 1011\ 1100\ 0001)$

2.16 การเข้ารหัสลับในรอบที่ 7

$E(R_6) = (001011\ 111000\ 001010\ 101100\ 000010\ 100000\ 000101\ 011000)$

$K_7 = (000011\ 110101\ 000111\ 111001\ 000101\ 101011\ 010001\ 001010)$

$\Gamma_7 = (001000\ 001101\ 001101\ 010101\ 000111\ 001011\ 010100\ 010010)$

$S_1(00, 0100) = S_1(0, 4) = 2 = 0010$

$S_2(01, 0110) = S_2(1, 6) = 8 = 1000$

$S_3(01, 0110) = S_3(1, 6) = 6 = 0110$

$S_4(01, 1010) = S_4(1, 10) = 2 = 0010$

$S_5(01, 0011) = S_5(1, 3) = 12 = 1100$

$S_6(01, 0101) = S_6(1, 5) = 12 = 1100$

$S_7(00, 1010) = S_7(0, 10) = 9 = 1001$

$S_8(00, 1001) = S_8(0, 9) = 9 = 1001$

$B_7 = (0010\ 1000\ 0110\ 0010\ 1100\ 1100\ 1001\ 1001)$

$P(B_7) = (0001\ 1011\ 0100\ 1101\ 0000\ 1010\ 0000\ 1101)$

$L_6 = (0100\ 1100\ 1010\ 0101\ 0111\ 1011\ 1100\ 0001)$

$R_7 = (0101\ 0111\ 1110\ 1000\ 0111\ 0001\ 1100\ 1100)$

$L_7 = R_6 = (0101\ 1100\ 0101\ 0110\ 0001\ 0000\ 0010\ 1100)$

2.17 การเข้ารหัสลับในรอบที่ 8

$E(R_7) = (001010\ 101111\ 111101\ 010000\ 001110\ 100011\ 111001\ 011000)$

$K_8 = (100111\ 110100\ 100111\ 011001\ 001011\ 001011\ 010101\ 100100)$

$\Gamma_8 = (101101\ 011011\ 011010\ 001001\ 000101\ 101000\ 101100\ 111100)$

$S_1(11, 0110) = S_1(3, 6) = 1 = 0001$

$S_2(01, 1101) = S_2(1, 13) = 9 = 1001$

$S_3(00, 1101) = S_3(0, 13) = 4 = 0100$

$S_4(01, 0100) = S_4(1, 4) = 6 = 0110$

$S_5(01, 0010) = S_5(1, 2) = 2 = 0010$

$S_6(10, 0100) = S_6(2, 4) = 2 = 0010$

$S_7(10, 0110) = S_7(2, 6) = 7 = 0111$

$S_8(10, 1110) = S_8(2, 14) = 5 = 0101$

$B_8 = (0001\ 1001\ 0100\ 0110\ 0010\ 0010\ 0111\ 0101)$

$P(B_8) = (0000\ 0010\ 0111\ 1001\ 0101\ 1100\ 1010\ 0010)$

$L_7 = (0101\ 1100\ 0101\ 0110\ 0001\ 0000\ 0010\ 1100)$

$R_8 = (0101\ 1110\ 0010\ 1111\ 0100\ 1100\ 1000\ 1110)$

$L_8 = R_7 = (0101\ 0111\ 1110\ 1000\ 0111\ 0001\ 1100\ 1100)$

2.18 การเข้ารหัสลับในรอบที่ 9

$E(R_8) = (001011\ 111100\ 000101\ 011110\ 101001\ 011001\ 010001\ 011100)$

$K_9 = (000111\ 110100\ 100110\ 011011\ 010011\ 100101\ 000010\ 101001)$

$\Gamma_9 = (001100\ 001000\ 100011\ 000101\ 111010\ 111100\ 010011\ 110101)$

$S_1(00, 0110) = S_1(0, 6) = 11 = 1011$

$S_2(00, 0100) = S_2(0, 4) = 6 = 0110$

$S_3(11, 0001) = S_3(3, 1) = 10 = 1010$

$S_4(01, 0010) = S_4(1, 2) = 11 = 1011$

$S_5(10, 1101) = S_5(2, 13) = 3 = 0011$

$S_6(10, 1110) = S_6(2, 14) = 11 = 1011$

$S_7(01, 1001) = S_7(1, 9) = 3 = 0011$

$S_8(11, 1010) = S_8(3, 10) = 9 = 1001$

$B_9 = (1011\ 0110\ 1010\ 1011\ 0011\ 1011\ 0011\ 1001)$

$P(B_9) = (1111\ 1010\ 1110\ 0000\ 0010\ 1111\ 1101\ 0110)$

$L_8 = (0101\ 0111\ 1110\ 1000\ 0111\ 0001\ 1100\ 1100)$

$R_9 = (1010\ 1101\ 0000\ 1000\ 0101\ 1110\ 0001\ 1010)$

$L_9 = R_8 = (0101\ 1110\ 0010\ 1111\ 0100\ 1100\ 1000\ 1110)$

2.20 การเข้ารหัสลับในรอบที่ 10

$E(R_9) = (010101\ 011010\ 100001\ 010000\ 001011\ 111100\ 000011\ 110101)$

$K_{10} = (001111\ 110010\ 100110\ 001101\ 010000\ 100111\ 100101\ 101001)$

$\Gamma_{10} = (011010\ 101000\ 000111\ 011101\ 011011\ 011011\ 100110\ 011100)$

$S_1(00, 1101) = S_1(0, 13) = 9 = 1001$

$S_2(10, 0100) = S_2(2, 4) = 10 = 1010$

$S_3(01, 0011) = S_3(1, 3) = 9 = 1001$

$S_4(01, 1110) = S_4(1, 14) = 14 = 1110$

$S_5(01, 1101) = S_5(1, 13) = 9 = 1001$

$S_6(01, 1101) = S_6(1, 13) = 11 = 1011$

$S_7(10, 0011) = S_7(2, 3) = 13 = 1101$

$S_8(00, 1110) = S_8(0, 14) = 12 = 1100$

$B_{10} = (1001\ 1010\ 1001\ 1110\ 1001\ 1011\ 1101\ 1100)$

$P(B_{10}) = (0111\ 1111\ 1111\ 1000\ 0011\ 0001\ 0110\ 0011)$

$L_9 = (0101\ 1110\ 0010\ 1111\ 0100\ 1100\ 1000\ 1110)$

$R_{10} = (0010\ 0001\ 1101\ 0111\ 0111\ 1101\ 1110\ 1101)$

$L_{10} = R_9 = (1010\ 1101\ 0000\ 1000\ 0101\ 1110\ 0001\ 1010)$

2.21 การเข้ารหัสลับในรอบที่ 11

$E(R_{10}) = (100100\ 000011\ 111010\ 101110\ 101111\ 111011\ 111101\ 011010)$

$K_{11} = (000110\ 110010\ 110010\ 001101\ 101000\ 101001\ 100100\ 111000)$

Γ_{11} = (100010 110001 001000 100011 000111 010010 011001 100010)

$S_1(10, 0001) = S_1(2, 1) = 1 = 0001$

$S_2(11, 1000) = S_2(3, 8) = 11 = 1011$

$S_3(00, 0100) = S_3(0, 4) = 6 = 0110$

$S_4(11, 0001) = S_4(3, 1) = 15 = 1111$

$S_5(01, 0011) = S_5(1, 3) = 12 = 1100$

$S_6(00, 1001) = S_6(0, 9) = 13 = 1101$

$S_7(01, 1100) = S_7(1, 12) = 2 = 0010$

$S_8(10, 0001) = S_8(2, 1) = 13 = 1011$

B_{11} = (0001 1011 0110 1111 1100 1101 0010 1011)

$P(B_{11})$ = (1101 1001 0100 1111 0111 1100 0100 1110)

L_{10} = (1010 1101 0000 1000 0101 1110 0001 1010)

R_{11} = (0111 0100 0100 0111 0010 0010 0101 0100)

$L_{11} = R_{10}$ = (0010 0001 1101 0111 0111 1101 1110 1101)

2.22 การเข้ารหัสลับในรอบที่ 12

$E(R_{11})$ = (001110 101000 001000 001110 100100 000100 001010 101000)

K_{12} = (010110 010010 110010 111100 110000 010001 111100 110010)

Γ_{12} = (011000 111010 111010 110010 010100 010101 110110 011010)

$S_1(00, 1100) = S_1(0, 12) = 5 = 0101$

$S_2(10, 1101) = S_2(2, 13) = 3 = 0011$

$S_3(10, 1101) = S_3(2, 13) = 10 = 1010$

$S_4(10, 1001) = S_4(2, 9) = 1 = 0001$

$S_5(00, 1010) = S_5(0, 10) = 3 = 0011$

$S_6(01, 1010) = S_6(1, 10) = 13 = 1101$

$S_7(10, 1011) = S_7(2, 11) = 8 = 1000$

$S_8(00, 1101) = S_8(0, 13) = 0 = 0000$

B_{12} = (0101 0011 1010 0001 0011 1101 1000 0000)

$P(B_{12})$ = (1111 0000 0000 0000 1110 0001 1000 1111)

L_{11} = (0010 0001 1101 0111 0111 1101 1110 1101)

R_{12} = (1101 0001 1101 0111 1001 1100 0110 0010)

$L_{12} = R_{11}$ = (0111 0100 0100 0111 0010 0010 0101 0100)

2.23 การเข้ารหัสลับในรอบที่ 13

$E(R_{12}) = (011010\ 100011\ 111010\ 101111\ 110011\ 111000\ 001100\ 000101)$

$K_{13} = (110101\ 001010\ 110010\ 101100\ 010111\ 010000\ 101000\ 111000)$

$\Gamma_{13} = (101111\ 101001\ 001000\ 000011\ 100100\ 101000\ 100100\ 111101)$

$S_1(11, 0111) = S_1(3, 7) = 7 = 0111$

$S_2(11, 0100) = S_2(3, 4) = 3 = 0011$

$S_3(00, 0100) = S_3(0, 4) = 6 = 0110$

$S_4(01, 0001) = S_4(1, 1) = 8 = 1000$

$S_5(10, 0010) = S_5(2, 2) = 1 = 0001$

$S_6(10, 0100) = S_6(2, 4) = 2 = 0010$

$S_7(10, 0010) = S_7(2, 2) = 11 = 1011$

$S_8(11, 1110) = S_8(3, 14) = 6 = 0110$

$B_{13} = (0111\ 0011\ 0110\ 1000\ 0001\ 0010\ 1011\ 0110)$

$P(B_{13}) = (0110\ 0010\ 0010\ 0011\ 1100\ 0110\ 0110\ 0111)$

$L_{12} = (0111\ 0100\ 0100\ 0111\ 0010\ 0010\ 0101\ 0100)$

$R_{13} = (0001\ 0110\ 0110\ 0100\ 1110\ 0100\ 0011\ 0011)$

$L_{13} = R_{12} = (1101\ 0001\ 1101\ 0111\ 1001\ 1100\ 0110\ 0010)$

2.24 การเข้ารหัสลับในรอบที่ 14

$E(R_{13}) = (100010\ 101100\ 001100\ 001001\ 011100\ 001000\ 000110\ 100110)$

$K_{14} = (110100\ 101010\ 111000\ 100110\ 010100\ 010101\ 100001\ 011100)$

$\Gamma_{14} = (010110\ 000110\ 110100\ 101111\ 001000\ 011101\ 100111\ 111010)$

$S_1(00, 1011) = S_1(0, 11) = 12 = 1100$

$S_2(00, 0011) = S_2(0, 3) = 14 = 1110$

$S_3(10, 1010) = S_3(2, 10) = 2 = 0010$

$S_4(11, 0111) = S_4(3, 7) = 8 = 1000$

$S_5(00, 0100) = S_5(0, 4) = 7 = 0111$

$S_6(01, 1110) = S_6(1, 14) = 3 = 0011$

$S_7(11, 0011) = S_7(3, 3) = 8 = 1000$

$S_8(10, 1101) = S_8(2, 13) = 3 = 0011$

$B_{14} = (1100\ 1110\ 0010\ 1000\ 0111\ 0011\ 1000\ 0011)$

$P(B_{14}) = (0110\ 0000\ 1010\ 1110\ 1010\ 1000\ 1101\ 0101)$

$L_{13} = (1101\ 0001\ 1101\ 0111\ 1001\ 1100\ 0110\ 0010)$

$R_{14} = (1011\ 0001\ 0111\ 1001\ 0011\ 0100\ 1011\ 0111)$

$L_{14} = R_{13} = (0001\ 0110\ 0110\ 0100\ 1110\ 0100\ 0011\ 0011)$

2.25 การเข้ารหัสลับในรอบที่ 15

$E(R_{14}) = (110110\ 100010\ 101111\ 110010\ 100110\ 101001\ 101110\ 101111)$

$K_{15} = (111010\ 001011\ 111000\ 100110\ 000000\ 011011\ 000010\ 111100)$

$\Gamma_{15} = (001100\ 101001\ 010111\ 010100\ 100110\ 110010\ 010100\ 010011)$

$S_1(00, 0110) = S_1(0, 6) = 11 = 1011$

$S_2(11, 0100) = S_2(3, 4) = 3 = 0011$

$S_3(01, 1011) = S_3(1, 11) = 14 = 1110$

$S_4(00, 1010) = S_4(0, 10) = 8 = 1000$

$S_5(10, 0011) = S_5(2, 3) = 11 = 1011$

$S_6(10, 1001) = S_6(2, 9) = 0 = 0000$

$S_7(00, 1010) = S_7(0, 10) = 9 = 1001$

$S_8(01, 1001) = S_8(1, 9) = 5 = 0101$

$B_{15} = (1011\ 0011\ 1110\ 1000\ 1011\ 0000\ 1001\ 0101)$

$P(B_{15}) = (0110\ 0011\ 1000\ 0001\ 0100\ 1011\ 1110\ 0111)$

$L_{14} = (0001\ 0110\ 0110\ 0100\ 1110\ 0100\ 0011\ 0011)$

$R_{15} = (0111\ 0101\ 1110\ 0101\ 1010\ 1111\ 1101\ 0100)$

$L_{15} = R_{14} = (1011\ 0001\ 0111\ 1001\ 0011\ 0100\ 1011\ 0111)$

2.26 การเข้ารหัสลับในรอบที่ 16

$E(R_{15}) = (001110\ 101011\ 111100\ 001011\ 110101\ 011111\ 111010\ 101000)$

$K_{16} = (111000\ 011011\ 011000\ 100110\ 100110\ 000110\ 011010\ 011001)$

$\Gamma_{16} = (110110\ 110000\ 100100\ 101101\ 010011\ 011001\ 100000\ 110001)$

$S_1(10, 1011) = S_1(2, 11) = 7 = 0111$

$S_2(10, 1000) = S_2(2, 8) = 5 = 0101$

$S_3(10, 0010) = S_3(2, 2) = 4 = 0100$

$S_4(11, 0110) = S_4(3, 6) = 13 = 1101$

$S_5(01, 1001) = S_5(1, 9) = 0 = 0000$

$S_6(01, 1100) = S_6(1, 12) = 0 = 0000$

$S_7(10, 0000) = S_7(2, 0) = 1 = 0001$

$S_8(11, 1000) = S_8(3, 8) = 15 = 1111$

$B_{16} = (0111\ 0101\ 0100\ 1101\ 0000\ 0000\ 0001\ 1111)$

$P(B_{16}) = (1000\ 1010\ 0000\ 0011\ 1101\ 1010\ 0111\ 0010)$

L_{15} = (1011 0001 0111 1001 0011 0100 1011 0111)

R_{16} = (0011 1011 0111 1010 1110 1110 1100 0101)

L_{16} = R_{15} = (0111 0101 1110 0101 1010 1111 1101 0100)

2.27 นำค่า L_{16} และ R_{16} ป้อนในกล่องสลับลำดับผกผัน IP^{-1} ตามตารางที่ 2.8 เพื่อให้ได้เป็นข้อความไซเฟอร์ Y ตามที่ต้องการ จะได้

Y = (1110 1001 0101 1100 1010 1111 0101 1100
1101 0010 1111 1100 1011 0111 0010 1111)
= è \ ¯ \ o Ò ü · /

ภาคผนวก ข

การเข้ารหัสแบบ AES

กำหนดให้ Input = GenerateGenerate

= 47 65 6E 65 72 61 74 65 47 65 6E 65 72 61 74 65

Key = ABnormalABnormal

= 41 42 6E 6F 72 6D 61 6C 41 42 6E 6F 72 6D 61 6C

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value w[0] – w[43]
Input	47 72 47 72 65 61 65 61 6E 74 6E 74 65 65 65 65			⊗	41 72 41 72 42 6D 42 6D 6E 61 6E 61 6F 6C 6F 6C
1	06 00 06 00 27 0C 27 0C 00 15 00 15 0A 09 0A 09	6F 63 6F 63 CC FE CC FE 63 59 63 59 67 01 67 01	6F 63 6F 63 FE CC FE CC 63 59 63 59 01 67 01 67	A5 B7 A5 B7 2C 6C 2C 6C 54 B4 54 B4 2E FE 2E FE	⊗ 7C 0E 4F 3D AD E0 82 EF 3E 5F 31 50 2F 43 2C 40
2	D9 B9 EA 8A 81 AC AE 83 6A EB 65 E4 01 BD 02 BE	35 56 87 7E 0C 91 E4 EC 02 E9 4D 69 7C 7A 77 AE	35 56 87 7E 91 E4 EC 0C 4D 69 C2 E9 AE 7C 7A 77	21 8E 42 76 75 42 38 31 D7 E4 E1 22 C4 8F 88 89	⊗ A1 AF E0 DD FE 3E BC 53 37 68 59 09 08 4B 67 27
3	80 21 A2 AB 8B 7C 84 62 E0 8C B8 2B CC C4 EF AE	CD FD 3A 62 3D 10 5F AA E1 64 6C F1 4B 1C DF E4	CD FD 3A 62 10 5F AA 3D 6C F1 E1 64 E4 4B 1C DF	39 BA 6C 38 BD 00 51 6B 32 86 6D ED E3 24 3D 5A	⊗ 48 E7 07 DA FF C1 7D 2E FB 93 CA C3 C9 82 E5 C2
4	71 5D 6B E2 42 C1 2C 45 C9 15 A7 2E 2A A5 D8 98	A3 4C 7F 98 2C 78 71 6E DD 59 5C 31 E5 06 61 46	A3 4C 7F 98 78 71 6E 2C 5C 31 DD 59 46 E5 06 61	CF DF 97 67 F1 18 D8 4A A9 6B BA A5 56 45 3E 04	⊗ 71 96 91 4B D1 10 6D 43 DE 4D 87 44 9E 1C F9 3B
5	BE 49 06 2C 20 08 B5 09 77 26 3D E1 C8 59 C7 3F	AE 3B 6F 71 B7 30 D5 01 F5 F7 27 F8 E8 CB C6 75	AE 3B 6F 71 30 D5 01 B7 27 F8 F5 F7 75 E8 CB C6	45 02 E3 11 D2 71 A2 C0 4F 26 D9 62 14 AB C8 44	⊗ 7B ED 7C 37 CA DA B7 F4 3C 71 F6 B2 2D 31 C8 F3
6	3E EF 9F 26 18 AB 15 34 33 57 2F D0 39 9A 00 B7	B2 DF DB F7 AD 62 59 18 C3 5B 15 70 12 B8 63 A9	B2 DF DB F7 62 59 18 AD 15 70 C3 5B A9 12 B8 63	65 2C FE 22 E0 EF 0D 38 1A 50 8D 49 F3 77 C6 32	⊗ E4 09 75 42 FD 27 90 64 31 40 B6 04 B7 86 4E BD
7	81 25 8B 60 1D C8 9D 5C 2B 10 3B 4D 44 F1 88 8F	0C 3F 3D D0 A4 E8 5E 4A F1 CA E2 E3 1B A1 C4 73	0C 3F 3D D0 E8 5E 4A A4 E2 E3 F1 CA 73 1B A1 C4	AA 64 F4 42 89 A6 00 02 AE 91 76 AC F8 CA A5 9C	⊗ E7 EE 9B D9 0F 28 B8 DC 4B 0B BD B9 9B 1D 53 EE
8	4D 8A 6F 9A 86 8E B8 DE E5 9A CB 15 63 D7 F6 72	E3 7E A8 B8 44 19 6C 1D D9 B8 1F 59 FB 0E 42 40	E3 7E A8 B8 19 6C 1D 44 1F 59 D9 B8 40 FB 0E 42	A9 EA 13 AB B0 B6 EC 51 04 B6 0E A1 B8 5A 93 5D	⊗ E1 0F 94 4D 59 71 C9 15 63 68 D5 6C AE B3 E0 0E
9	48 E5 87 10 E9 C7 25 B4 67 DE DB 3D 16 E9 73 A5	52 D9 17 CA 1E C6 3F 8D 85 1D B9 27 47 1E 8F 06	52 D9 17 CA C6 3F 8D 1E B9 27 85 1D 06 47 1E 8F	4A 88 39 3F 13 89 9C 5E F7 61 A9 64 85 E6 0D 43	⊗ A3 AC 34 75 09 78 B1 A4 C8 A0 75 19 4D FE 1E 10
10	E9 24 01 4A 1A F1 2D FA 3F C1 D3 7D C8 18 13 53	1E 36 7C D6 A2 A1 D8 2D 75 78 66 FF E8 AD 7D ED	1E 36 7C D6 A1 D8 2D A2 66 FF 75 78 ED E8 AD 7D	4F 08 57 4F 06 6C 14 7C 5F 28 54 03 24 B6 9D 41	⊗ DC 70 48 3D DD A5 14 B0 02 A2 D7 CE D0 2E 30 20
Output	93 78 1F 72 DB C9 00 CC 5D 8A 13 CD F4 98 AD 61				

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวปวีณา ดวงแก้วกูล

ภูมิลำเนา 54/1 ม.7 ต.บ้านเป้า อ.เมือง จ.ลำปาง

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเขลางค์นคร
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: o_o_no_999@hotmail.com

ชื่อ นายศุภพงศ์ เยาวรัตน์

ภูมิลำเนา 29 ม.5 ต.เหมืองหม้อ อ.เมือง จ.แพร่

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิริยาลัยจังหวัดแพร่
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: tummedia@hotmail.com

ชื่อ นางสาวสุชาดา อินต๊ะชัย

ภูมิลำเนา 94/6 ม.4 ต.ท่าผา อ.เกาะคา จ.ลำปาง

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนเขลางค์นคร
- ปัจจุบันกำลังศึกษาอยู่ในระดับปริญญาตรีชั้นที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: tau_si@hotmail.com