โปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา

Secure Instant Messenger using Java

นายธีรพงศ์ เหมปรัชยกุล

นายวีระ เวียงชัยศรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

โปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา

Secure Instant Messenger using Java

โดย

นายธีรพงศ์ เหมปรัชยกุล

นายวีระ เวียงชัยศรี

อาจารย์ที่ปรึกษา

อ.ธนา หงษ์สุวรรณ

อ.อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา

Secure Instant Messenger using Java

ผู้จัดทำ

1. นายธีรพงศ์ เหมปรัชยกุล รหัสนักศึกษา 40013252
2. นายวีระ เวียงชัยศรี รหัสนักศึกษา 40013270

..... อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

..... อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

โปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา

นายธีรพงศ์ เหมปรัชยกุล
นายวีระ เวียงชัยศรี

อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ในปัจจุบันโปรแกรมที่ใช้รับ-ส่งข้อความประเภท Instant messenger ตัวอย่างเช่น ICQ ได้รับความนิยมและใช้งานกันเป็นอย่างมาก แต่ยังไม่เหมาะสมกับการใช้งานในองค์กรที่ต้องการความปลอดภัยในการรับ-ส่งข้อความระหว่างกันเพราะข้อมูลยังสามารถถูกดักจับได้ระหว่างทาง ผู้จัดทำจึงได้เกิดแนวคิดที่จะสร้างโปรแกรมรับ-ส่งข้อความ ที่มีการเข้ารหัสของข้อมูลก่อนและหลังการรับ-ส่งเพื่อให้เกิดความปลอดภัยในการรับ-ส่งข้อความด้วยความปลอดภัยมากขึ้น โดยใช้ภาษาจาวาในการเขียนโปรแกรมส่งสารด่วนแบบปลอดภัยนี้เพื่อให้โปรแกรมทำงานได้บนหลายแพลตฟอร์ม

Secure Instant Messenger using Java

Mr. Teerapong Hempruchayakul
Mr. Weera Wiangchaisree

Mr. Thana Hongsuwan Advisor
Mr. Akkradach Watcharapupong Advisor

1999

ABSTRACT

Nowadays, the online instant messenger software, such as ICQ, are very popular. But it does not appropriate to use in the organizations who need security. This project offers a secure instant messenger program which encrypts and decrypts message before and after transferring using JAVA programming. Java programs can run on many platforms.

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากในปัจจุบันโปรแกรมที่ใช้รับ-ส่งข้อความและรับ-ส่งไฟล์ ตัวอย่างเช่น ICQ ได้รับความนิยมและใช้งานกันเป็นอย่างมาก แต่ยังไม่เหมาะกับการใช้งานในองค์กรที่ต้องการความปลอดภัยในการรับ-ส่งข้อความและรับ-ส่งไฟล์ระหว่างกัน ทางผู้จัดทำจึงได้เกิดแนวคิดที่จะสร้างโปรแกรมรับ-ส่งข้อความ ที่มีการเข้ารหัสของข้อมูลก่อนการรับ-ส่ง เพื่อให้เกิดความปลอดภัยในการรับ-ส่งข้อความด้วยความปลอดภัยมากขึ้น โดยใช้ภาษาจาวาในการเขียนโปรแกรมส่งสารด่วนแบบปลอดภัยนี้

1.2 วัตถุประสงค์ของการพัฒนา

1. เพื่อสร้างโปรแกรมต้นแบบที่ส่งข้อความและข้อมูลสำหรับไคลเอ็นต์ (Client)
2. ใช้การเข้า-ถอดรหัสข้อมูลเพื่อความปลอดภัยของข้อความและข้อมูลระหว่างการส่ง-รับ
3. เพื่อสร้างโปรแกรมด้วยภาษาจาวาให้สามารถใช้งานได้หลายแพลตฟอร์ม

1.3 เป้าหมายของการที่พัฒนา

1. โปรแกรมรับส่งข้อความและข้อมูลแบบปลอดภัยด้วยภาษาจาวาสำหรับไคลเอ็นต์
2. สามารถทำการรันข้ามแพลตฟอร์มได้ตามความสามารถของจาวา
3. ใช้งานกับโปรแกรม ICQ ของ Mirabilis ได้
4. เอกสารประกอบเรื่องโพรโตคอล ICQ, วิธีการเข้า-ถอดรหัสข้อมูล และคู่มือการใช้งานของโปรแกรมโดยละเอียด

1.4 ขอบเขตของการพัฒนา

: Input/output specification

- รับค่าการพิมพ์ทางคีย์บอร์ดและเมาส์
- แสดงผลทางจอภาพโดยเป็นกราฟิก

: Function specification

- รับส่งข้อความเป็น Plan text ระหว่างผู้ใช้ ICQ และโปรแกรม IsagQ ได้
- ใส่ User login และพาสเวิร์ดใช้งานโปรแกรม IsagQ ได้
- แสดงสถานะออนไลน์ (Online) , ออฟไลน์ (Offline) และ Secure ICQ ที่ติดต่อด้วยได้
- ค้นหาผู้ใช้ทาง UIN, E-mail, Name ได้

- เข้ารหัส (Encryption) และถอดรหัส (Decryption) ข้อความระหว่างผู้ใช้โปรแกรม IsagQ และโปรแกรม ICQ ที่มีใช้งานทั่วไปได้

- ใช้งานภาษาไทยได้

1.5 รายละเอียดของการพัฒนา

เทคนิคหรือเทคโนโลยีที่ใช้:

เทคโนโลยีที่ใช้ในเรื่องของการโปรแกรมเราจะใช้ภาษาจาวาในการสร้างเพราะว่าน่าศึกษาและกำลังได้รับความนิยม เหมาะแก่การใช้งานในลักษณะของโปรแกรมที่ต้องทำงานบนหลายแพลตฟอร์ม

ส่วนในด้านเทคนิคแล้วเราต้องทราบถึงการทำงานของโพรโตคอล ICQ ซึ่งมีหลายเวอร์ชันด้วยกันเช่นเวอร์ชัน 2, 3, 4 และปัจจุบันเวอร์ชัน 5 ซึ่งในการสื่อสารระหว่างผู้ใช้ ICQ แต่ละคนจะใช้การเชื่อมต่อแบบ TCP packet ส่วนการสื่อสารอื่นๆ นั้นทำงานผ่านทาง UDP packet โดยการติดต่อจะมีด้วยกัน 3 รูปแบบคือ การเชื่อมต่อระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์, การเชื่อมต่อระหว่างเซิร์ฟเวอร์กับไคลเอ็นต์ และการเชื่อมต่อระหว่างไคลเอ็นต์กับไคลเอ็นต์

เนื่องจากโปรแกรมที่จะสร้างขึ้นนี้ต้องการความปลอดภัยเป็นสำคัญ จึงต้องทำการเข้ารหัสและถอดรหัสข้อมูลที่ต้องการเพื่อให้เป็นความลับไม่สามารถอ่านได้โดยผู้ไม่หวังดี โดยการเข้ารหัสจะใช้การเข้ารหัสแบบกุญแจสาธารณะ (หรือที่เรียกกันว่าคีย์คู่) ซึ่งต้องใช้กุญแจ 2 ชนิดด้วยกันในการทำงานคือต้องใช้กุญแจสาธารณะที่เรียกว่าพับลิกคีย์ (Public-Key) และกุญแจที่เป็นความลับ (Secret-Key หรือ Private-Key) ในการเข้าและถอดรหัสเพื่อความปลอดภัย รวมถึงมีการนำการเข้ารหัสแบบ DES มาใช้งานด้วยเพื่อให้เกิดความเร็วในการทำงาน

เครื่องมือที่ใช้ในการพัฒนา:

ใช้ภาษาจาวาในการพัฒนาโครงงาน เพราะว่าจาวามีความเป็นอิสระต่อแพลตฟอร์ม (Platform Independent) ช่วยให้เคลื่อนย้ายโปรแกรมจากระบบหนึ่งไปรันยังเครื่องคอมพิวเตอร์ระบบอื่นได้โดยง่าย และสามารถเขียนโปรแกรมแบบ OOP (Object Oriented Programming) ได้ง่ายและขนาดของซอร์สโปรแกรมที่ได้มีขนาดเล็ก

โปรแกรมที่จะได้นี้เป็นจาวาแอพพลิเคชันไม่ใช่จาวาแอพเพล็ต เพราะว่าต้องการให้สามารถรันได้ด้วยตัวเองไม่ต้องใช้บราวเซอร์ในการเปิดขึ้นมาทำงาน

คุณลักษณะของซอฟต์แวร์:

1. มีความปลอดภัยในการรับส่งข้อมูลเมื่อผู้ใช้งานทำการติดตั้งโปรแกรมนี้ทั้งสองฝ่าย
2. ใช้งานได้กับโปรแกรม ICQ ของ Mirabilis
3. โปรแกรมสามารถทำงานได้บนหลายระบบไม่จำกัดอยู่แค่บนวินโดว์
4. ใช้งานภาษาไทยได้

บทที่ 2

ภาษาจาวา

ภาษาจาวา หรือ Java Programming Language ถูกพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ เริ่มแรกซันได้คิดค้นภาษาจาวาขึ้นมาเพื่อใช้กับระบบปฏิบัติการ Set-top box คือโครงการของระบบตลาดอิเล็กทรอนิกส์ ที่ต้องการให้ระบบสามารถทำการควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพล็ตฟอร์ม (Platform) ที่ใช้คือการพัฒนาระบบปฏิบัติการหรือซอฟต์แวร์ให้สามารถทำงานได้โดยไม่สนใจว่าไมโครโพรเซสเซอร์ที่ใช้อยู่จะเป็นรุ่นใด ยี่ห้อไหน ฮาร์ดแวร์เป็นของอะไรทั้งยังได้รวมความสามารถไปถึงการเชื่อมต่อการเปลี่ยนข้อมูลข่าวสารทั่วๆ ไป และทางด้านธุรกิจผ่านทางระบบเครือข่ายของคอมพิวเตอร์อีกด้วย

ทางทีมงานซึ่งพัฒนาโปรแกรมนี้ได้เล็งเห็นถึงการเจริญเติบโตและความสามารถในการแลกเปลี่ยนข่าวสารบนอินเตอร์เน็ตจึงได้วางแผนและพัฒนาระบบอินเทอร์เฟซแบบใหม่เพื่อใช้กับอินเตอร์เน็ตขึ้น ด้วยสาเหตุนี้เองภาษาจาวาจึงได้บังเกิดขึ้น รวมทั้งยังได้มีการเปิดตัวเว็บบราวเซอร์ชื่อว่า Hotjava เพื่อใช้สำหรับแสดงความสามารถของภาษาจาวา

2.1 ความสามารถของภาษาจาวา

ลักษณะของภาษาจาวา เป็นการเขียนโปรแกรมอ้างอิงเชิงวัตถุ OOP (Object Oriented Programming) สามารถสรุปความสามารถการทำงานได้ดังต่อไปนี้

1. การทำงานที่ไม่ขึ้นอยู่กับแพล็ตฟอร์ม โดยใช้วิธีการแปลงข้อมูลแบบไบต์โค้ด
2. การใช้เมธอดจัดการกับข้อมูลแทนการใช้ฟังก์ชัน
3. ความสามารถในการโต้ตอบแบบอินเตอร์แอ็กทีฟ กับงานด้านมัลติมีเดีย
4. ความสามารถในการดึงเอาคลาสไลบรารีจากที่ต่างๆ ผ่านทาง HTTP และ FTP
5. ความสามารถในการเล่นไฟล์เสียงไฟล์วิดีโอและ ไฟล์ภาพกราฟิกเคลื่อนไหวการ
6. โต้ตอบแบบเรียลไทม์
7. เป็นระบบมัลติเธรดคือสามารถทำงานได้หลายงานพร้อมกันในเวลาเดียว
8. ไม่มีตัวแปรประเภทพอยน์เตอร์จึงหมดปัญหาเกี่ยวกับการขอใช้หน่วยความจำโดยตรง
9. มีความปลอดภัยสูง ด้วยความสามารถที่ไม่ให้ผู้ใช้เข้าไปยุ่งเกี่ยวกับหน่วยความจำของระบบ จึงปลอดภัยจากไวรัส
10. การดาวน์โหลดซอฟต์แวร์ใหม่ๆ และอัพเกรดเวอร์ชันใหม่ได้อย่างง่ายดายโดยผ่านทางเว็บเพจ

2.2 การนำจาวาไปประยุกต์สร้างและพัฒนาในงานด้านต่างๆ

จากความสามารถดังที่กล่าวผ่านมาข้างต้นคงพอจะเป็นข้อสรุปได้แล้วว่าจาวาเป็นภาษาโปรแกรมที่มีความสามารถรอบด้าน อีกทั้งยังมีความยืดหยุ่นภายในตัวสูง การที่นำมันมาสร้างและพัฒนางานดีๆ มีประสิทธิภาพคงจะทำได้ไม่ยากนัก ซึ่งเราสามารถใช้จาวาสร้างและพัฒนาแอพพลิเคชันต่างๆ โดยยกตัวอย่างพอสังเขปได้ดังนี้

ไคลเอ็นต์

จาวาสามารถเป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้านไคลเอ็นต์ได้เพียงพอโดยเฉพาะด้านกราฟิกและระบบติดต่อกับผู้ใช้ เพราะปัจจุบันโปรแกรมที่เขียนขึ้นจากภาษาจาวาจะทำงานที่ฝั่งไคลเอ็นต์เป็นส่วนมากจะเห็นได้จากจาวาแอพเพล็ตต่างๆ บนเวบบราวเซอร์ แต่จะให้ดีและสมบูรณ์ยิ่งขึ้นต่อความเป็นไคลเอ็นต์ซอฟต์แวร์ที่ดีก็คือ การติดต่อระบบฐานข้อมูลจากไคลเอ็นต์ไปเป็นเซิร์ฟเวอร์ แม้ว่าในปัจจุบันจาวาจะสามารถทำได้แล้วก็ตามแต่ก็ต้องใช้เครื่องมือชนิดอื่นเป็นตัวช่วย เช่น การเชื่อมต่อกับฐานข้อมูลแบบ ODBC (Object-Oriented Database Connectivity) ซึ่งการทำงานของทางฝั่งเซิร์ฟเวรอ์ต้องใช้ภาษาคอมพิวเตอร์ชนิดอื่นช่วยเพื่อการติดต่อกับฐานข้อมูล

เซิร์ฟเวอร์

จาวาเมื่อใช้สร้างโปรแกรมประยุกต์บนเซิร์ฟเวอร์นั้นจาวาจำเป็นที่จะต้องได้รับการปรับปรุงการทำงานอยู่สามอย่างคือ การเชื่อมต่อกับฐานข้อมูลโดยตรง การทำงานให้สอดคล้องกับอินพุต เอาต์พุต และความเร็วของการรันต้องเพิ่มขึ้นเท่าๆ กับโปรแกรมประยุกต์ทั่วไป

ระบบ JDBC (Java Database Connectivity) ได้เข้ามาแก้ปัญหาในส่วนของการติดต่อกับฐานข้อมูลโดยตรง โดยการร่วมมือกันของบริษัทซอฟต์แวร์ที่มีชื่อเสียงหลายบริษัท ได้สร้างมาตรฐานการเชื่อมต่อกับฐานข้อมูลของจาวาขึ้น แต่ในขณะนี้ก็ยังไม่เป็นที่รู้จักแพร่หลายนักสำหรับ JDBC คาดว่าในอนาคตอันใกล้นี้การเชื่อมต่อฐานข้อมูลจะทำได้ง่ายขึ้นจากไคลเอ็นต์สู่เซิร์ฟเวอร์โดยใช้จาวา

Security

จาวาได้สร้างสภาพแวดล้อมเสมือน "Java Virtual Machine" ซึ่งจะอนุญาตให้เฉพาะจาวาแอพเพล็ต หรือจาวาแอพพลิเคชันซึ่งเป็นไบต์โค้ด เท่านั้นที่สามารถรันได้โดยที่ไบต์โค้ดจะปราศจากไวรัสหรือส่วนที่จะทำอันตรายต่อระบบ อันเนื่องจากสภาพแวดล้อมเสมือนจะจำกัดสิทธิ์การเข้าใช้ทรัพยากรของไบต์โค้ดในอนาคตการรักษาความปลอดภัยอีกระบบหนึ่งก็กำลังถูกนำมาปรับใช้คือ "การเข้ารหัสข้อมูล" เพื่อเพิ่มความปลอดภัยของข้อมูล

การใช้จาวาประยุกต์สร้างและพัฒนางานยังสามารถแบ่งเป็นประเภทตามลักษณะของกลุ่มงานได้ดังนี้

- งานด้านการศึกษา โดยจัดทำเป็นลักษณะสั่งการสอนคล้ายกับ CAI มีความสามารถในเชิงโต้ตอบระหว่างผู้ใช้งานกับคอมพิวเตอร์

- ตอบสนองงานทางด้านเว็บเพจทำให้งานสร้างเว็บเพจดูมีชีวิตชีวาขึ้น โดยการนำเอาความสามารถทางด้านมัลติมีเดียมาใช้อย่างเต็มที่
- สร้างสรรค์เครื่องมือพัฒนา (Software Developer tool kits)
- สร้างแอพพลิเคชันทางธุรกิจ มีความสามารถในการประมวลผลข้อมูลด้วยตัวเอง
- สร้างตัวจัดการบนอินเตอร์เน็ต มีความสามารถในการเรียกหาข้อมูลผ่านทางระบบเครือข่าย
- พัฒนาเกม จะมีความเป็นมัลติ คือสามารถเล่นได้หลายๆ คนพร้อมกันผ่านทางระบบเครือข่ายหลากหลายพื้นที่ไม่ว่าจะเล่นด้วยกัน ณ ตำแหน่งใด ทั่วทุกมุมโลก

2.3 สภาพแวดล้อมเสมือน (Java Virtual Machine)

สภาพแวดล้อมเสมือนเป็นหลักการสร้างคอมพิวเตอร์จำลองของจาวา โดยการสมมติให้มีคอมพิวเตอร์อีกเครื่องหนึ่งขึ้นมาโดยเครื่องนี้จะใช้ในการคอมไพล์โปรแกรมภาษาจาวาทุกโปรแกรม เมื่อต้องการให้โปรแกรมภาษาจาวาไปทำงานบนคอมพิวเตอร์จริงๆ เครื่องใดเราก็เพียงแต่สร้างตัวอินเทอร์พรีเตอร์ของคอมพิวเตอร์จำลองตัวนี้บนเครื่องนั้นๆ ภาษาจาวาทุกโปรแกรมก็จะสามารถทำงานบนระบบคอมพิวเตอร์นั้นๆ ได้ตามต้องการด้วยหลักการนี้ก็เป็นที่มาของคุณสมบัติของจาวาที่ไม่ขึ้นกับแพล็ตฟอร์มและฮาร์ดแวร์ใดๆ หรือพอร์เทเบิล (Portable) เราอาจจะเรียก Java Virtual Machine สั้นๆ ว่า JVM ก็ได้

สำหรับเหตุผลที่ต้องมี JVM นี้จริงๆ แล้วเป็นการกำหนดคำขึ้นมา เพื่อเป็นคำจำกัดความเฉพาะในความคิดเท่านั้น เพื่อให้นักพัฒนาจะได้ไม่ถูกบังคับให้ต้องสร้างตัวอินเทอร์พรีเตอร์ตามแนวทางแบบใดแบบหนึ่งโดยเฉพาะแต่ตัวอินเทอร์พรีเตอร์ที่สร้างขึ้นตามข้อกำหนดดังกล่าวไม่ว่าจะอยู่บนแพล็ตฟอร์มใดจะสามารถรันโปรแกรมที่เขียนขึ้นในภาษาจาวาได้ โดยให้ผลลัพธ์ออกมาเหมือนกัน

แต่ในข้อกำหนดของ JVM ก็มีการให้นิยามที่ชัดเจนมากๆ เกี่ยวกับการออกแบบอินเทอร์พรีเตอร์ในหลายๆ ส่วนโดยเฉพาะอย่างยิ่งส่วนที่เกี่ยวข้องกับการกระจายโค้ดของจาวา ไปใส่ไว้ในรูปแบบที่กำหนด ข้อกำหนดนี้ได้แก่ไวยากรณ์ของออบโค้ดและโอเปอร์แรนด์ พร้อมกับค่าประจำตัว การจัดโครงสร้างของโค้ด การจัดวางรูปแบบของออบเจ็กต์ของจาวา ข้อกำหนดต่างๆ เหล่านี้จะทำให้นักพัฒนาอินเทอร์พรีเตอร์ทั้งหลาย สามารถจะสร้างอินเทอร์พรีเตอร์ขึ้นมาใหม่บนแพล็ตฟอร์มใดๆ ก็ได้ ดังนั้นการพัฒนาภาษาจาวาจึงไม่ได้ถูกปิดกั้นอยู่กับระบบของซันซึ่งเป็นผู้คิดค้นภาษาจาวานี้ขึ้นมาเท่านั้น

จากที่กล่าวมาข้างต้นถึงคุณสมบัติของจาวาที่ไม่ยึดติดอยู่กับแพล็ตฟอร์มใดๆ ทำให้การทำงานกับระบบคอมพิวเตอร์ที่ทำงานแบบกระจาย (Distributed Computing) ได้รับการตอบสนองอย่างเหมาะสม นั่นคือ ทำให้เกิดความพอร์เทเบิลนั่นเอง

อย่างไรก็ตาม JVM ก็ยังมีข้อจำกัดอยู่ ข้อจำกัดของ JVM นั้นจะอยู่ที่ข้อจำกัดของการออกแบบตัวอินเทอร์พรีเตอร์แทน เช่น การจำกัดค่าของโอเปอร์แรนด์ และขนาดของสแต็ก เป็นต้น ข้อกำหนดเหล่านี้หมายถึงว่า JVM สามารถจะอ้างถึงหน่วยความจำได้เฉพาะในห้วงแอดเดรสเท่าที่มีอยู่เท่านั้น

ข้อจำกัดภายในของตัว JVM เองนั้นมีห้วงแอดเดรสให้ใช้อยู่ถึง 4 GB เพราะขนาดของความกว้างของการอ้างแอดเดรสเป็น 32 บิตเมธอดต่างๆ ของจาวามีขนาดได้เพียง 32 KB เพราะมีข้อจำกัดของการใช้คำสั่งกระโดด เป็นแบบ 16 บิต โดยมีบิตแรกเป็นตัวบอกว่ากระโดดไปข้างหน้า หรือข้างหลัง และบิต

ต่อๆ ไป บอกระยะทางการกระโดดจากจุดเหล่านี้มีขนาดเพียง 8 บิต นอกจากนั้นจำนวนของค่าคงที่ที่อยู่ในส่วนกลาง ยังถูกจำกัดด้วยขนาดของดัชนี 16 บิต ทำให้มีจำนวนได้เพียง 32,000 ค่าในแต่ละเมธอด

การที่เราถือว่าการมีค่าเหล่านี้เป็นข้อจำกัด ก็อาจจะเป็นการมองการณ์ไกลไปสักหน่อย เนื่องจากปัจจุบันเครื่องคอมพิวเตอร์ส่วนใหญ่มีหน่วยความจำกัดเพียง 16 หรือ 32 MB กันเท่านั้น หน่วยความจำขนาด 4 GB จึงยังเป็นเรื่องที่ไม่ต้องคิดหนักในตอนนี้ ในขณะที่ขนาดของเมธอดจำกัดที่ 32 KB ก็เป็นเพียงเมธอดเดียวเท่านั้น

2.4 หลักการทำงานของภาษาจาวา

ในที่นี้ขออธิบายการทำงานของภาษาจาวาเริ่มตั้งแต่การคอมไพล์ตัวโปรแกรมจนกระทั่งเราเรียกใช้งานในเว็บเพจ ดังนี้

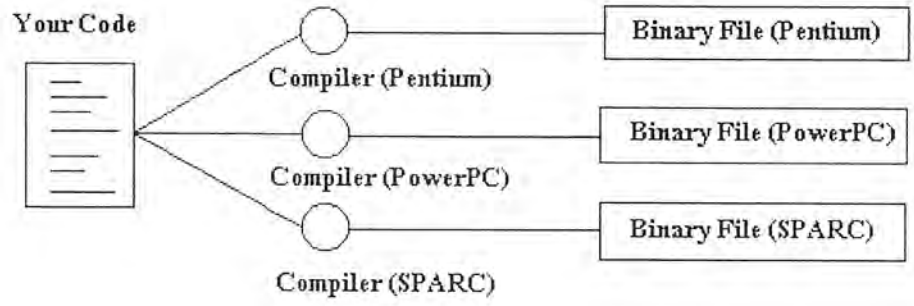
2.4.1 การคอมไพล์

คอมไพเลอร์ของภาษจาวาก็เช่นเดียวกับคอมไพเลอร์ในภาษาอื่นๆ นั่นคือมันจะสร้างรหัสภาษาเครื่อง (Machine Code หรือ Assembler Code) จากภาษาในระดับที่สูงกว่าเพื่อให้ซีพียู (CPU : Central Processing Unit) สามารถนำไปใช้งานได้

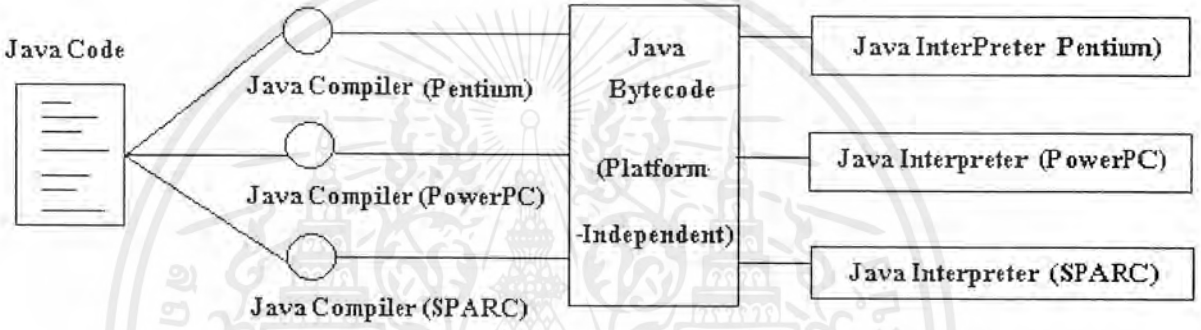
แต่ข้อแตกต่างที่สำคัญระหว่างคอมไพเลอร์ของภาษาจาวากับภาษาอื่นๆ คือซีพียูหรือโปรเซสเซอร์ที่จะทำหน้าที่ในการปฏิบัติตามคำสั่งที่ได้จากการคอมไพล์ภาษาจาวานั้นไม่มีอยู่จริง เป็นเพียงสิ่งที่สมมติขึ้นมาที่เรียกว่า Java Virtual Machine นอกจากนี้การอ้างถึงส่วนต่างๆ ของโปรแกรมที่คอมไพล์ด้วยคอมไพเลอร์ของภาษาจาวาก็จะมีวิธีการที่แตกต่างออกไป

คอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนการอ้างถึงส่วนของโปรแกรมจากการใช้ชื่อแบบในภาษาสูง ไปเป็นตัวเลขเหมือนคอมไพเลอร์ภาษาอื่นๆ ทำกันและคอมไพเลอร์ภาษาจาวาก็จะไม่มีการสร้างแผนที่ของการจัดวางโปรแกรมบนหน่วยความจำขึ้นมาในระหว่างการคอมไพล์ด้วยเหตุผลที่สำคัญคือ เพื่อเป็นการสร้างความพอร์เทเบิลให้กับตัวโปรแกรมเพราะการจัดวางตำแหน่งของโปรแกรมจะต้องขึ้นอยู่กับลักษณะการทำงานของโปรเซสเซอร์ตัวใดตัวหนึ่ง การยังไม่จัดวางตำแหน่งช่วยทำให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลาง สามารถนำไปใช้บนคอมพิวเตอร์แพล็ตฟอร์มอื่นๆ ได้นอกจากนั้นยังทำให้เกิดความปลอดภัยอีกด้วย

สิ่งที่ได้จากการคอมไพล์ในภาษาจาวาเราเรียกว่า ไบต์โค้ด (bytecode)



รูปที่ 2.1 การประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน



รูปที่ 2.2 การประมวลผลของโปรแกรมภาษาจาวา

2.4.2 การวางตำแหน่งในหน่วยความจำ

ในภาษาจาวาจะไม่มีการลดรูปแบบการอ้างถึงส่วนต่างๆ ของโปรแกรมจากการเรียกเป็นชื่อให้เหลือเพียงตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งของโปรแกรมลงในหน่วยความจำ คอมไพเลอร์ภาษาจาวาจะทิ้งชื่อของแต่ละส่วนของโปรแกรม (โดยเฉพาะเมธอด) เอาไว้ในตัวโปรแกรมที่สร้างขึ้นเมื่อโปรแกรมทำงาน จะเป็นหน้าที่ของตัว อินเทอร์พรีเตอร์ที่จะคอยเปิดตารางค้นหาที่อยู่ของเมธอดที่ต้องการเรียกใช้งาน โดยก่อนที่จะเริ่มทำงานจริง อินเทอร์พรีเตอร์จะต้องสร้างแผนที่ในการจัดวางสิ่งต่างๆ ลงในหน่วยความจำขึ้นมาก่อนแล้วจึงสร้างตารางขึ้นมาเพื่อช่วยหาตำแหน่งของเมธอดเมื่อมีการเรียกใช้งานโดยใช้ชื่อของเมธอด

2.4.3 การรันโปรแกรม

การรันโค้ดที่คอมไพล์เอาไว้สำหรับ Java Virtual Machine เป็นหน้าที่ของตัวอินเทอร์พรีเตอร์ การรันโปรแกรมจะแบ่งได้เป็น 3 ขั้นตอนหลักๆ คือ การอ่าน การตรวจสอบความถูกต้อง และการทำตามโค้ด หน้าที่ในการอ่านโค้ดเข้าสู่ระบบจะเป็นของ Class Loader หน้าที่การทำงานในส่วนนี้จะไม่ได้อ่านเข้ามาเฉพาะๆ ไฟล์จาวาที่กำลังจะเรียกใช้เท่านั้น แต่จะอ่านคลาสที่มีการอ้างถึงและคลาสที่มีการสืบทอดต่อมาโดยคลาสที่อ้างถึง เมื่อผ่านขั้นตอนนี้แล้ว โค้ดทั้งหมดก็จะถูกส่งผ่านตัวตรวจสอบไบต์โค้ดเพื่อให้

แน่ใจว่าโค้ดที่ส่งมามีความถูกต้องตามมาตรฐานของจาวา และจะไม่รบกวนเสถียรภาพของระบบ เมื่อผ่านการตรวจสอบแล้วโค้ดก็จะถูกส่งต่อไปยังระบบรันไทม์ (Run-Time System) ซึ่งจะสั่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่งซึ่งหลักการทำงานในแต่ละขั้นตอนที่กล่าวมาข้างต้นเป็นดังนี้

2.4.4 Class Loader

ตัวคลาสโหลดเดอร์ จะทำหน้าที่ดึงโค้ดทั้งหมดที่จำเป็นในการทำงานของแอพพลิเคชัน ไม่ว่าจะเป็นคลาสที่ถูกสืบทอดมาหรือคลาสอื่นๆ ที่มีการเรียกใช้เมื่อโหลดเดอร์ดึงคลาสใดเข้ามาแล้วก็จะจัดคลาสนั้นๆ ใส่เข้าไว้ใน namespace ของมันเองโดยการเก็บจะใช้ชื่อของคลาสเป็นสำคัญ ไม่ได้ใช้การอ้างถึงเป็นตัวเลขหลักการนี้ก็จะเหมือนกันกับการทำงานของ Virtual Machine ที่โอเอส (OS : Operating System) สร้างขึ้นให้แอพพลิเคชันแต่ละตัวทำงานถ้าไม่ได้มีการเจาะจงเรียกใช้คลาสที่อยู่นอก namespace นี้ การเรียกใช้ชื่อต่างๆ ในคลาสก็จะไม่มีการรบกวนกันระหว่างคลาสเลย

คลาสทั้งหมดที่อยู่บนเครื่องโลคอล จะได้รับห้วงแอดเดรส (Address Space) เป็นของตัวเอง ส่วนคลาสต่างๆ ที่ดึงมาจากภายนอกจะได้รับ namespace เป็นของตัวเองการทำงานลักษณะนี้จะช่วยให้คลาสที่อยู่บนโลคอลทำงานได้ประสิทธิภาพดีขึ้นเพราะใช้ namespace ร่วมกันได้ แต่ก็ยังมีการป้องกันความผิดพลาดที่อาจจะเกิดจากคลาสที่ดึงเข้ามาจากภายนอกและ ในทางกลับกันคลาสที่อิมพอร์ตเข้ามาก็ปลอดภัยจากความผิดพลาดที่อาจจะเกิดขึ้นจากคลาสโลคอลด้วย

เมื่อคลาสทั้งหมดที่เกี่ยวข้องกับการทำงานถูกอิมพอร์ตเข้ามาเรียบร้อยแล้ว การจัดวางหน่วยความจำสำหรับเริ่มต้นการทำงานก็จะเกิดขึ้นได้การเรียกชื่อต่างๆ ก็จะสามารถจับคู่กับแอดเดรสจริงๆ ของหน่วยความจำได้ แล้วตัวโหลดเดอร์จะสร้างตารางสำหรับค้นหาที่อยู่เลี่ยงจากการทำงานผิดปกติของซูเปอร์คลาส (Super class) และการอ้างแอดเดรสที่ไม่ถูกต้องได้

2.4.5 ตรวจสอบไบต์โค้ด

เมื่อโค้ดเดินทางมาจนถึงขั้นตอนการสร้างตารางจับคู่ชื่อกับแอดเดรสแล้ว ก็ยังไม่สามารถแน่ใจได้ว่าโค้ดที่อ่านเข้ามาจะมีความปลอดภัยดังนั้นจึงต้องมีตัว Verifier หรือตัวตรวจสอบไบต์โค้ด ทำหน้าที่ตรวจสอบความถูกต้องทีละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวาและ สอดคล้องกับการทำงานของตัวโปรแกรมเองหรือไม่ การตรวจสอบโค้ดในเชิงทฤษฎีจะสร้างและค้นหาปัญหาต่างๆ ได้หลายอย่าง เช่น จะไม่มีการสร้างพอยต์เตอร์ที่เกินกว่าหน่วยความจำจริง ไม่มีคำสั่งใดสามารถละเมิดสิทธิ์การทำงานของตัวโปรแกรมได้ ไม่มีการจับคู่ออบเจ็คผิด จะไม่มีการให้โอเปอร์แรนด์มากหรือน้อยเกินไป การกำหนดค่าต่างๆ สำหรับไบต์โค้ดจะต้องถูกต้องครบถ้วน และจะไม่มีการแปลงข้อมูลผิดรูปแบบ

การใช้ตัวตรวจสอบตอบสนองจุดประสงค์ 2 ประการที่สำคัญคือสิ่งต่างๆ ดังที่กล่าวมาแล้วจะถูกตรวจสอบก่อน ทำให้ตัวอินเทอร์พรีเตอร์มั่นใจได้ว่า ไบต์โค้ดที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบ และจุดประสงค์ที่สองก็คือตัวอินเทอร์พรีเตอร์จะทำงานตามไบต์โค้ดได้รวดเร็วกว่า เพราะไม่ต้องคอยระวังว่าจะมีปัญหาเกิดขึ้น และไม่ต้องหยุดเป็นช่วงๆ เมื่อพบปัญหาและต้องแก้ไข

ในการทำงานไบต์โค้ดจะถูกตรวจสอบเพียงครั้งเดียวเท่านั้น และจะทำงานไปได้ตลอดไม่ต้องมีการตรวจสอบซ้ำอีกเมื่อมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

2.4.6 การทำงานตามโค้ด

เมื่อตัวโหลดเดอร์ได้รวบรวมโค้ดเข้ามาสู่ระบบทำการจัดวางในหน่วยความจำ และตัวตรวจสอบได้ทำการตรวจสอบความถูกต้องแล้วโค้ดก็จะถูกส่งต่อไปยังตัวอินเทอร์พรีเตอร์เพื่อทำงานตามคำสั่ง การทำงานตามคำสั่งของโค้ดก็คือการเปลี่ยนโค้ดให้กลายเป็นคำสั่ง การทำงานจริงที่ตัวระบบไคลเอ็นต์ที่รันโค้ดนี้สามารถทำงานได้ ซึ่งวิธีการที่ทำได้ก็มีอยู่ 2 วิธีด้วยกันคือ ตัวอินเทอร์พรีเตอร์ทำการคอมไพล์โค้ดหล่านี้ให้กลายเป็นเนทีฟโค้ดที่ตัวเครื่องไคลเอ็นท์เข้าใจแล้วค่อยทำงาน เพื่อให้ได้ความเร็วสูงสุดในการทำงาน กับอีกวิธีหนึ่งก็คือตัวอินเทอร์พรีเตอร์อ่านโค้ดเข้ามาแล้วตีความทำงานไปทีละคำสั่ง และทำการตีความไปเรื่อยๆ ตลอดเวลาที่มีการทำงาน

โดยปกติแล้วผู้สร้างตัวอินเทอร์พรีเตอร์มักจะเลือกใช้วิธีการหลัง รูปแบบของไบต์โค้ดในภาษาจาวามีความยืดหยุ่นเพียงพอที่จะสามารถเปลี่ยนไปทำงานบนเครื่องไคลเอ็นท์แบบต่างๆ ได้โดยไม่มีการก่อให้เกิดโอเวอร์เฮดมากมายเกินความจำเป็นอย่างไรก็ตาม ไคลเอ็นท์ของจาวาบางระบบจะมีความสามารถในการทำงานได้ทั้งสองวิธีคือโปรแกรมเมอร์สามารถจะเลือกใช้วิธีการคอมไพล์กับงานที่เน้นการคำนวนมากๆ เพื่อเป็นการเพิ่มสมรรถนะในการทำงานให้ได้เต็มที่ ซึ่งไคลเอ็นต์แบบนี้จะให้ได้ทั้งความพอร์เทเบิล และสมรรถนะที่ดี

การสร้างระบบรันไทม์ที่ดีจะต้องถ่วงดุลย์ความสำคัญ 3 ประการให้ได้พอเหมาะ นั่นคือความพอร์เทเบิล ความปลอดภัย และสมรรถนะเรื่องของความพอร์เทเบิลทำได้โดยการใช้รูปแบบของไบต์โค้ดที่มีความเป็นกลางเพียงพอ สามารถนำไปใช้รันบนเครื่องคอมพิวเตอร์แบบต่างๆ ได้โดยง่ายนอกจากนั้นการที่ตัวอินเทอร์พรีเตอร์ทำการกำหนดการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์ แทนที่จะเป็นระหว่างการคอมไพล์เหมือนภาษาอื่นๆ ก็เป็นการเพิ่มความแน่นอนว่าคลาสต่างๆ ที่อิมพอร์ตเข้ามาจะยังคงใช้ได้อยู่ตลอดเวลา เรื่องของความปลอดภัยนั้น เป็นสิ่งที่มีการคำนึงถึงอยู่ตลอดการทำงานของระบบรันไทม์ โดยเฉพาะอย่างยิ่งในส่วนของตัวตรวจสอบไบต์โค้ด ที่ทำให้แน่ใจได้ว่าโปรแกรมจะทำงานได้ถูกต้องตามข้อกำหนดของจาวา ส่วนเรื่องของสมรรถนะนั้นก็สามารถจัดการได้ในสองระยะ คือ พยายามเอาโอเวอร์เฮดทั้งหลายไปใส่ไว้ที่ตอนเริ่มต้นโหลดโปรแกรมเข้ามาสู่ระบบ หรือไม่ก็กำหนดให้ทำงานเป็นแบบแบ็กกราวนด์ (Back-Ground Thread)

ด้วยสิ่งต่างๆ เหล่านี้ทำให้จาวาสามารถปล่อยสมรรถนะระดับที่น่าพอใจออกมาได้โดยที่ยังคงไว้ซึ่งความพอร์เทเบิลและสภาพแวดล้อมที่ปลอดภัยนอกจากนั้นยังสามารถจะดึงสมรรถนะระดับสูงสุดออกมาใช้ได้ทันทีเมื่อต้องการ

2.4.7 การสร้างและรันภาษาจาวา

การสร้างโปรแกรมจากจาวา ก็เหมือนกับภาษาโปรแกรมอื่นๆ คือเขียนโค้ดโปรแกรมจากเอดิเตอร์ใดๆ ก็ได้บันทึกอยู่ในนามสกุล .java จากนั้นจึงนำไปคอมไพล์ด้วยคอมไพเลอร์ของจาวาจะได้ไฟล์ใหม่ในนามสกุล .class ที่เก็บรหัสของการคอมไพล์ไว้จาวาเรียกรูปแบบข้อมูลที่อยู่ในไฟล์ใหม่นี้ว่าไบต์โค้ดไฟล์ที่ได้คือแอพเพล็ตนั่นเองแอพเพล็ตยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่เราคุ้นเคยกัน เพราะข้อมูลแบบไบต์โค้ดจะมีรูปแบบข้อมูลที่อยู่กึ่งกลางระหว่างโค้ดโปรแกรม (Source Code) กับโค้ดที่คอมพิวเตอร์อ่านแล้วนำไปปฏิบัติงานได้ทันที (Machine Code) หากจะรันแอพเพล็ตบน

ระบบใดๆ จะต้องใช้อินเทอร์พรีเตอร์จาวาของระบบนั้นๆ เพื่อตรวจสอบความถูกต้องของไบต์โค้ดแล้วแปลให้เป็นรหัสภาษาเครื่องส่งให้ระบบปฏิบัติการนำไปรันต่อไปขั้นตอนการทำงาน

2.5 เว็บบราวเซอร์ที่สนับสนุนจาวา

HTML (Hypertext Markup Language) เป็นภาษาที่มีรูปแบบหนึ่งของภาษา SGML (Standard Generalized Markup Language) นิยมใช้กันทั่วไปบนอินเทอร์เน็ตเป็นภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งของคอมพิวเตอร์มีลักษณะของโปรแกรมเป็นไฟล์ตัวอักษรในมาตรฐานของรหัสแอสกีประกอบด้วย Reserve Markup Directive หรือคำสั่งต่างๆ ซึ่งมีลักษณะเป็นคำสั่งที่ง่ายต่อการทำความเข้าใจ เพื่อให้การสร้างเว็บเพจมีความสะดวกและง่ายมากยิ่งขึ้น ทั้งนี้เพื่อส่งผ่านโครงสร้างของข้อมูลระหว่างผู้ใช้

ในปัจจุบันภาษา HTML ได้ถูกกำหนดมาตรฐานขึ้นมาเป็นรุ่นที่ 3.0 HyperText ที่ถูกสร้างขึ้นมาจะอยู่ในรูปของแฟ้มเอกสาร HTML (Document File) ที่มีการกำหนดคุณสมบัติ (markup) ของเว็บเพจเข้าไป ยังไม่สามารถแสดงผลข้อมูลออกมาให้ใช้งานได้โดยตรง ถ้าต้องการดูผลลัพธ์ที่เกิดจากการสร้างต้องผ่านโปรแกรมที่ทำหน้าที่แปลคำสั่งนั้นก่อน เราเรียกโปรแกรมที่ทำหน้าที่นี้ว่า "โปรแกรมเว็บบราวเซอร์" (Web Browser Program) หน้าที่หลักของโปรแกรมนี้คือเป็นตัวแปลคำสั่งของ HyperText แล้วแสดงผลออกมาเป็นรูปภาพ เสียง ข่าวสารและ ข้อมูล คุณสมบัติอื่นๆ ของโปรแกรมเว็บบราวเซอร์ ไม่ว่าจะเป็นการดาวน์โหลดไฟล์ การดึงรูปภาพมาใช้งาน การพิมพ์เอกสาร HTML ออกทางเครื่องพิมพ์ การส่งจดหมายอิเล็กทรอนิกส์ (e-mail) และความสามารถด้านอื่นๆ อีกมากมาย ทำให้เราสามารถใช้งานอินเทอร์เน็ตหรืออินทราเน็ตได้อย่างสะดวกรวดเร็ว ง่าย และสนุกสนาน ซึ่งทีมพัฒนาภาษาจาวาได้วางแผนและพัฒนาระบบอินเทอร์เฟซโดยเล็งเห็นถึงการเจริญเติบโตและความสามารถของภาษาจาวาจึงได้เปิดตัวเว็บบราวเซอร์มีชื่อว่า Hotjava เพื่อใช้สำหรับแสดงศักยภาพของจาวาที่มีอยู่อย่างเต็มที่ราวปี 2538 เพราะว่าในขณะนั้นยังไม่มีบราวเซอร์ตัวใดที่รู้จักและสามารถทำงานร่วมกับภาษาจาวาได้เลย และแน่นอนว่าบราวเซอร์ที่ได้รับการพัฒนาต่อๆ มาย่อมจะมีการเพิ่มเติมความสามารถที่ดีกว่า HotJava อย่างแน่นอนคือ Netscape Navigator เป็นบราวเซอร์ที่บริษัท Netscape Communications ได้สร้างและเป็นบริษัทแรกที่ได้ลิขสิทธิ์ภาษาจาวาจากบริษัท Sun Web Browser โดย Netscape เวอร์ชันแรกที่สนับสนุนจาวาคือ Netscape 3.0 เพื่อใช้สำหรับการทำงานบน Windows3.X, Windows95, WindowsNT, Solaris และ Apple Macintosh และยังมีเว็บบราวเซอร์ของอีกบริษัทหนึ่งที่มีชื่อเสียงไม่แพ้ Netscape คือบริษัท Microsoft ซึ่งได้ออกเว็บบราวเซอร์ที่สนับสนุนจาวาในชื่อ Internet Explorer 3.0 เพื่อให้เว็บบราวเซอร์ที่บริษัทพัฒนาเป็นเวอร์ชันล่าสุดในขณะนั้น สามารถรองรับการออกแบบโฮมเพจที่ถูกออกแบบด้วยภาษาจาวา

ปัจจุบันนี้มีบริษัทที่สร้างเว็บบราวเซอร์อื่นๆ อีกเช่น AOL (America online), IBM ซึ่งแน่นอนว่าปัจจุบันนี้เว็บบราวเซอร์เหล่านั้นสามารถสนับสนุนกับโฮมเพจในเครือข่ายอินเตอร์เน็ตที่ใช้ภาษาจาวาแน่นอนเพราะปัจจุบันได้มีการนำจาวาไปประยุกต์ใช้งานในงานด้านต่างๆ ทางธุรกิจ การศึกษา บันเทิง ฯลฯ อย่างมากมาย

2.6 จาวาทูล (Java Tools)

การศึกษาและพัฒนาโปรแกรมภาษาจาวานั้นในช่วงเริ่มแรกนั้นมีลักษณะที่เรียกว่า command line driven compiler เครื่องมือแรกที่ใช้ในการประยุกต์เพื่อสร้างแอพเพล็ต คือ Sun's Java Development Kit (JDK) ซึ่งแอพเพล็ตที่ได้มานั้นมีลักษณะที่มีการเคลื่อนไหวไม่มากนักหรือแอพเพล็ตที่มีความยุ่งยากไม่มากนัก และการประยุกต์เพื่อสร้าง standalone Java applications นั้นก็เช่นกัน นักพัฒนาโปรแกรมก็ยังต้องการเครื่องมือที่ใช้ช่วยในการศึกษาและพัฒนาโปรแกรมด้วย ในปัจจุบันมีเครื่องมือที่ที่ช่วยในการพัฒนาโปรแกรมภาษาจาวาซึ่งต่างได้รวมเอาคุณสมบัติของ GUI (Graphic User Interface) ไว้ในการพัฒนาจาวาแอพพลิเคชันโดยจาวาทูลเหล่านั้น Compatible กับ Java Language Standards. เครื่องมือที่ใช้ศึกษาและพัฒนาโปรแกรมภาษาจาวาจึงมีความง่ายและสะดวกกว่าในการที่จะศึกษาและใช้งานภาษาจาวามากขึ้น เครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวานั้นมีหลายบริษัทที่พัฒนาเครื่องมือเหล่านั้นออกมายกตัวอย่างโปรแกรมที่มีชื่อเสียงพอสังเขปดังต่อไปนี้

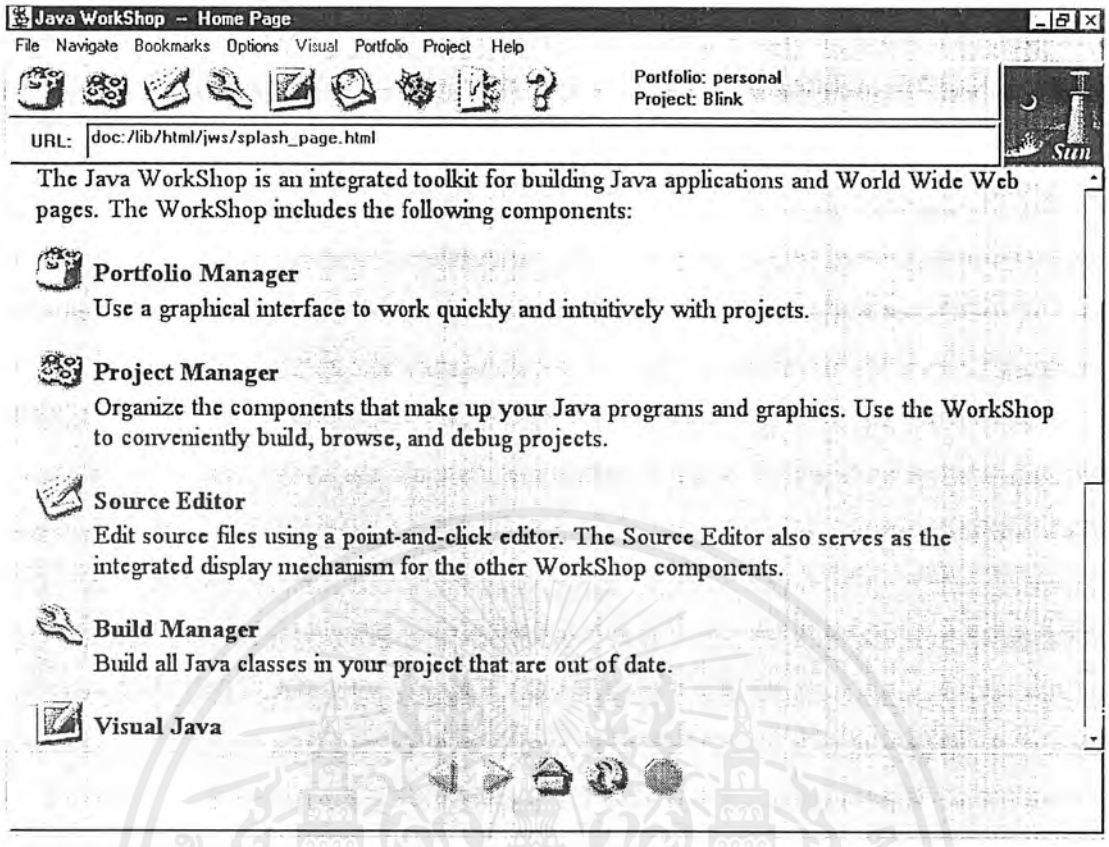
2.6.1 JDK หรือ Java Development Kit

JDK เป็นเครื่องมือสำหรับช่วยในการสร้างและพัฒนาจาวาประกอบไปด้วยไฟล์ต่างๆ ดังต่อไปนี้

1. ไฟล์ appletviewer.exe ใช้สำหรับทดสอบจาวาแอพเพล็ต เพื่อ ดูผลการทำงาน โดยไม่ต้องสั่งรันผ่านทางบราวเซอร์
2. ไฟล์ java.exe เป็นอินเทอร์พลีเตอร์ใช้ทดสอบการทำงานของจาวาแอพพลิเคชัน
3. ไฟล์ javac.exe มาจากคำว่าจาวาคอมไพเลอร์ใช้สำหรับแปลซอสโปรแกรมที่เราเขียนขึ้นแบบแอสกีโค้ด (.java) เป็นไบต์โค้ด (.class)
4. ไฟล์ javadoc.exe ใช้สำหรับสร้างไฟล์เอกสารจากซอสโค้ดของจาวาในรูปแบบของ HTML ภายในจะเป็นการบอกรายละเอียดของคลาสต่างๆ ที่ผู้ใช้เขียนขึ้น กฎการเขียนที่ควรรู้ไว้กรณีที่มีผู้สนใจอยากจะสร้างคือ ผู้ใช้จะต้องใส่คำสั่ง /** เข้าไปด้วยเสมอ ตรงส่วนบนของโปรแกรมก่อนการประกาศคลาส ลักษณะการใช้งานจะเหมือนกับคำสั่ง // และ /* */ (สำหรับใส่ Comment คำอธิบายทั่วๆ ไป)
5. ไฟล์ javap.exe ใช้แปลงคลาสไฟล์ของจาวาที่เป็นภาษาแอสเซมบลีกลับมาเป็นซอสโค้ด

2.6.2 Sun's Java WorkShop

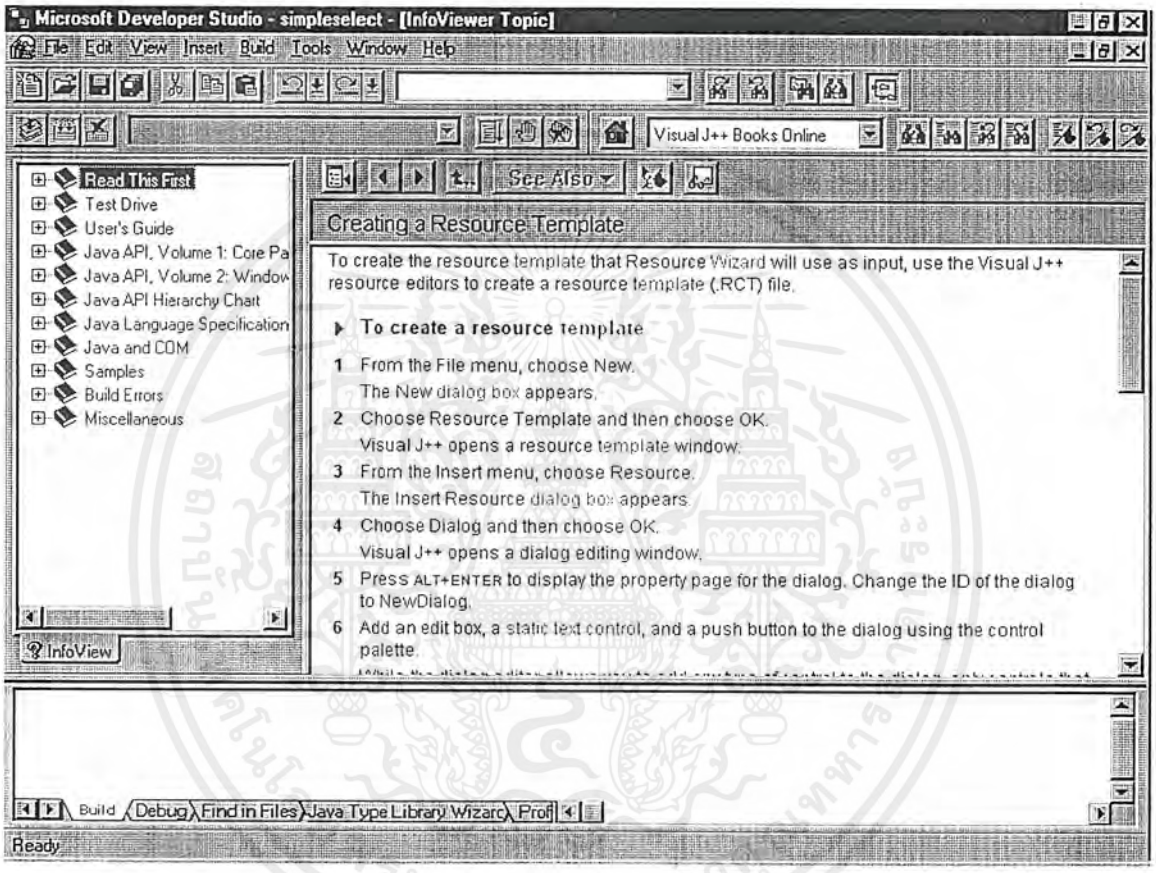
Java WorkShop 1.0 เป็นผลิตภัณฑ์ที่สามารถใช้ในการเขียนโปรแกรมจาวาและ HTML ที่สามารถแสดงผลได้ในรูปแบบหรือลักษณะที่คล้ายกับ Browser ซึ่งมาจาก Sun's HotJava Web browser. ทำให้จาวาคอมโพเน็นต์ที่ใช้รันภายใต้ Microsoft Windows 95, Windows NT, and Sun Solaris 2.4 ซึ่งใช้แพล็ตฟอร์มเดียวกันคือ Java Virtual Machine (JVM) และมีคอมไพเลอร์คอมโพเน็นต์บางตัว ที่มาจาก Sun Java Development Kit (JDK). ซึ่งผลิตภัณฑ์นี้ต้องการใช้ฮาร์ดแวร์ในการพัฒนาที่มีประสิทธิภาพค่อนข้างมากเช่นต้องการ RAM มากถึง 32MB ที่จะทำให้ประสิทธิภาพในการทำงานนั้นเป็นที่ยอมรับได้



รูปที่ 2.3 โปรแกรม Sun Microsystems: Java WorkShop

2.6.3 Microsoft Visual J++, Professional Edition

Visual J++ คือเครื่องมือที่ใช้พัฒนาการเขียนจาวาแอพเพล็ตและแอพพลิเคชันโดยบริษัทไมโครซอฟท์. Visual J++ ยังมีคุณสมบัติอีกอย่างที่ใน Java Language ของซันไม่มีคือในส่วนที่เรียกว่า Component Object Model (COM) และ Type Library Wizard ที่ทำงานโดย type libraries ของ COM objects (รวมถึง ActiveX controls) ที่ผู้ใช้สามารถที่จะอิมพอร์ตมาใช้ได้ด้วยเหตุนี้เองผู้ใช้จึงสามารถใช้เมธอดและกำหนดคุณสมบัติเฉพาะได้อย่างง่ายดาย



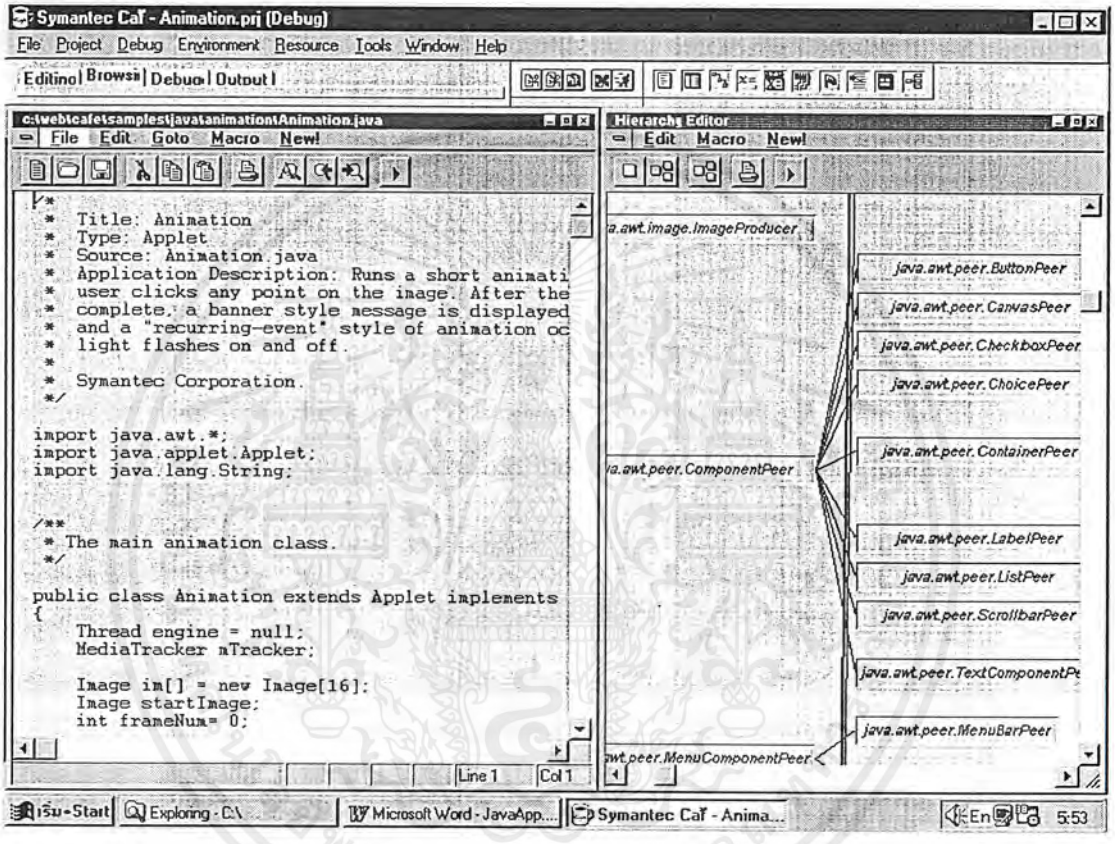
รูปที่ 2.4 โปรแกรม Microsoft Visual J++

Sun JDK 1.1 ได้ถูกรวมอยู่ใน Visual J++ ตั้งแต่เวอร์ชัน 1.0 ซึ่งปัจจุบันมีถึงเวอร์ชัน 1.1 รวมทั้ง International และ UNICODE features, Digital signing ของแอพเพล็ต, JAR packaging capability (our .CAB files), Java Native Method Interface capabilities, จาวาBeans หรือตลอดจน Support ActiveX รวมทั้งการติดต่อกับ Database ด้วยวิธี RDO (Remote Access Object) หรือ DAO (Data Access Object)

Visual J++ ยังมีความสามารถในด้าน visually create java forms โดยซอสเอดิเตอร์ยอมให้ผู้ใช้ visually layout ในการติดต่อกับแอพพลิเคชันของผู้ใช้เอง รีซอสเอดิเตอร์ก็เหมือนกับเอดิเตอร์โดยทั่วไปที่ใช้กันอยู่ใน Microsoft Visual C++ ซึ่งผู้ใช้งานยังสามารถที่จะคอนเวิร์ตเป็นรูปแบบของ Visual Basic และ Visual C++

2.6.4 Symantec Visual Cafe

Visual Cafe ก็เป็นเครื่องมืออีกชนิดหนึ่งที่มีรูปแบบการอินเทอร์เฟซคล้ายกับทูลตัวอื่นๆ แต่จะมี wizard ใหม่สำหรับช่วยทางด้าน component-based programming ซึ่งเรียกว่า RAD (Rapid Application Development) ยกตัวอย่างเช่น Borland Delphi หรือ Microsoft Visual Basic. และ Visual Cafe ยังมี JavaBeans specification. ซึ่งนักพัฒนาโปรแกรมสามารถเขียนโปรแกรมด้วยคอมโพเน็นต์ที่ถูกสร้างขึ้นมาแล้ว dragging มายังรูปแบบที่ต้องการได้ สามารถ setting properties และ events ได้โดยตรง.



รูปที่ 2.5 โปรแกรม SymantecVisual Café

สามารถสรุปเป็นตารางเปรียบเทียบความสามารถของเครื่องมือในการพัฒนาภาษาจาวาได้ดังต่อไปนี้

Summary of features			
Y = YES N = NO			
Products listed in alphabetical order	**Java WorkShop 1.0**	**Microsoft Visual J++ 1.0**	**Visual Café 1.0**
Development Environment			
Context-sensitive help	Y	Y	Y
Wizards or equivalents	Y	Y	Y
Cut-and-paste from online help	N	Y	Y
User can customize and reuse desktop	Y	Y	Y
Third-party program launch from IDE	Y	Y	Y
Include JIT compiler	N	Y	Y
Supports JDK tools	Y	Y	Y
Editor			
Syntax highlighting	N	Y	Y
Remappable keys	N	Y	Y
Macros	N	Y	Y
Brief compatibility	N	Y	Y
Epsilon Compatibility	Y	Y	Y
Programming Tools			
Wizard can generate applets	Y	Y	Y
Wizard can generate applications	Y	Y	Y
Wizard can add control events	Y	N	Y
Wizard adds animation automatically	N	Y	Y
Class browser	Y	Y	Y
Can browse uncompiled projects	Y	Y	Y
Graphical class-designer tool	N	N	Y
GUI resource designer	Y	Y	Y

Attaches code to controls in resource editor	Y	N	Y
Supports .BMP/.GIF/.JPG graphics files	YYY	YYY	YYY
Can import Windows resource script files	N	Y	Y
HTML editor included	N	N	N
Supports ActiveX components	N	Y	N
ActiveX creation in Java	N/A	Y	N/A
Supports JavaBeans	N	N	Y
Can target reusable components	Y	Y	Y
Provides classes for database connectivity	N	Y	N
Provides resolution-independent form design	Y	Y	Y
Debugging			
Integrated debugger	Y	Y	Y
Includes browser with debugging hooks	Y	Y	Y
Debugger catches Java exceptions	Y	Y	Y
Breakpoints	Y	Y	Y
Watch variables	Y	Y	Y
Support multithreaded debugging	Y	Y	Y
Supports remote debugging	N	N	Y
Integrated source control	Y	Y	N
Disassembles Java byte code	N	Y	N
Applet Viewer	Y	Y	Y
JavaScript/HTML debugging	NN	NN	NN

ตารางที่ 2.1 ตารางเปรียบเทียบเครื่องมือในการพัฒนาภาษาจาวา

2.7 ความแตกต่างระหว่างจาวาแอพเพล็ต และจาวาแอพพลิเคชัน

จาวาแอพเพล็ตกับจาวาแอพพลิเคชัน เป็นภาษาคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อใช้งานบนอินเทอร์เน็ตเหมือนกัน ทำงานอยู่บนโฮมเพจร่วมกับภาษา HTML ที่ใช้สร้างโฮมเพจเหมือนกันลักษณะที่แตกต่างกันก็คือ

- จาวาแอพเพล็ต ไม่สามารถทำงานเดี่ยวๆ ด้วยตนเองได้จะทำงานร่วมกันกับไฟล์ .html โดยเราจะเรียกใช้ไฟล์ที่เขียนจากจาวานี้ผ่านทาง Tag คำสั่ง <APPLET>...</APPLET> ภายในไฟล์ HTML จากนั้นก็จะเรียกใช้ไฟล์โปรแกรม javac.exe มาทำการคอมไพล์เพื่อที่จะได้ไฟล์ที่มีนามสกุล .class ไว้สำหรับให้ไฟล์ HTML เรียกผ่านวิธีสั่งแสดงผลการทำงานของจาวาแอพเพล็ต เราจะใช้โปรแกรม appletviewer.exe หรือเว็บบราวเซอร์เช่น Netscape, Hotjava, Internet Explorer ฯลฯ โดยเลือกใช้ตัวใดตัวหนึ่งก็ได้
- จากวาแอพพลิเคชันตัวที่ 2 นี้มันจะสามารถปฏิบัติงานเดี่ยวๆ ได้ด้วยตัวเองคล้ายกับการเขียนโค้ดในภาษาโปรแกรมทั่วไป สิ่งที่ควรจดจำไว้ให้ดีในการเขียนจาวาแอพพลิเคชันก็คือ เราจะต้องมีเมธอดของ Main(string args[]) อย่างน้อยๆ 1 เมธอดเพื่อใช้เป็นจุดเริ่มต้นของโปรแกรม อันนี้ถ้าหากสังเกตให้ดีจะเหมือนกับการเขียนโปรแกรมภาษาซีทั่วๆไป วิธีการคอมไพล์ก็จะใช้โปรแกรม javac.exe เช่นเดียวกับจาวาแอพเพล็ต ส่วนการสั่งแสดงผลเราจะใช้โปรแกรม java.exe แทน ซึ่งจะแตกต่างจากจาวาแอพเพล็ต ตรงจุดนี้

2.8 ความแตกต่างระหว่างจาวาและจาวาสคริปต์

สามารถสรุปเป็นตารางแสดงการเปรียบเทียบระหว่างจาวาสคริปต์และจาวาได้ดังนี้

จาวาสคริปต์	จาวา
1. ไม่มีการถูกคอมไพล์ก่อนรันแต่จะรันได้เลยโดยใช้อินเทอร์พรีเตอร์บนไคลเอ็นต์	1. ถูกคอมไพล์ที่เซิร์ฟเวอร์ก่อนรันบนไคลเอ็นต์
2. Object-based, รหัสใช้ object ภายในซึ่งมีอยู่แล้วและไม่มีการประกาศ class หรือการสืบทอด class	2. Object-oriented, แอพเพล็ต มีการสร้าง Object และสืบทอดจาก class
3. รหัสรวมอยู่ในเนื้อสคริปต์ HTML	3. แอพเพล็ตจะแยกต่างหากจากสคริปต์ HTML
4. ไม่มีการประกาศชนิดของตัวแปร	4. ต้องมีการประกาศชนิดของตัวแปร
5. ตรวจสอบความถูกต้องในขณะที่รันถ้ามีข้อผิดพลาดเกิดขึ้นอินเทอร์พรีเตอร์จะแจ้งให้ทราบ ในขณะที่รันไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้	5. ต้องถูกตรวจสอบความถูกต้องก่อนการรันคือในขณะคอมไพล์ถ้าคอมไพล์ไม่ผ่านไม่สามารถรันได้ ไม่สามารถเขียนข้อมูลลงฮาร์ดดิสก์ได้

ตารางที่ 2.2 ตารางเปรียบเทียบจาวาสคริปต์และจาวา

2.9 ทิศทางของจาวา

จาวาได้เริ่มต้นจากการเป็นภาษาคอมพิวเตอร์ง่ายๆ สำหรับสร้างแอพเพล็ต หรือโปรแกรมเล็กๆ ซึ่งสามารถทำงานบนเครื่องคอมพิวเตอร์ใดๆ ก็ได้โดยไม่จำเป็นต้องสร้างโปรแกรมที่ใหญ่โตมากนักขึ้นมาปัจจุบันนี้กระแสความต้องการและ ความนิยมของจาวากำลังสูงขึ้นอย่างรวดเร็วส่งผลให้เกิดพัฒนาโปรแกรมประยุกต์หรือแอพพลิเคชันใหม่ๆ โดยใช้ภาษาจาวาและ จาวาไม่ได้เป็นเพียงแค่แอพเพล็ตเล็กๆ อีกต่อไปแล้วเพราะเราสามารถใช้จาวาเพื่อเป็นเครื่องมือสำหรับสร้างซอฟต์แวร์ขนาดใหญ่ ซึ่งผู้ที่กำหนดทิศทางหรือมองภาพของจาวาได้ดีในอนาคตอันใกล้ก็คือผู้อยู่เบื้องหลังความสำเร็จของจาวาซึ่งบทความต่อไปนี้เป็นการสรุปบทพูดคุยกันกับบุคคล 2 ท่านซึ่งบุคคลแรกคือผู้ให้กำเนิดจาวาคือ Jame Gosling และบุคคลที่สองคือผู้ที่ทำให้ชาวโลกได้รู้จักกับจาวา โดยเข้ามาพัฒนาระบบการตลาดและสนับสนุนโปรดักต์ชันต่างๆ ของเทคโนโลยีจาวาให้ออกไปสู่ลูกค้าทั่วโลก คือ Dr.Alan Baratz ประธานบริษัท JavaSoft ซึ่งบทความนี้ได้เรียบเรียงสรุปจากวารสารทางอินเทอร์เน็ตที่ชื่อ JavaWorld ดังต่อไปนี้

ในปี 2539 ที่ผ่านมาเป็นปีแห่งการพัฒนาซอฟต์แวร์ประเภทไคลเอ็นต์ หลายต่อหลายฝ่าย ต่างพุ่งเป้าความสนใจไปที่เครื่องคอมพิวเตอร์ประเภทโต้ตอบกับผู้ใช้ในฝั่งไคลเอ็นต์กับแอพพลิเคชันต่างๆ ที่สำหรับใช้ภายในองค์กร ในปี 2540 จะเป็นปีทองที่ซอฟต์แวร์จะก้าวไปไกลมากกว่าไคลเอ็นต์ โดยจะเคลื่อนที่เข้าไปใกล้ชิดกับผู้ใช้งานยิ่งขึ้นในรูปแบบของอุปกรณ์ส่วนบุคคลต่าง ๆ โดยการใช้จาวาควบคุมอุปกรณ์ไฟฟ้าและ เครื่องจักรขนาดเล็กเช่น PDA (Personal Digital Assistant) หรือ เครื่องมือช่วยระบบดิจิตอลส่วนบุคคล, อุปกรณ์ Set-top Boxes (กล่องควบคุมอเนกประสงค์) Smart Phones, ระบบฝังตัวในอุปกรณ์ต่างๆ เช่นในเครื่องวิทยุติดตามตัว, เครื่องพิมพ์, เครื่องส่งแฟกซ์, โทรศัพท์เว็บ, โทรทัศน์เว็บบราวเซอร์, รวมไปถึงเครื่องใช้ไฟฟ้าต่างๆ ซึ่งอุปกรณ์เหล่านี้จะถูกเชื่อมการติดต่อกับระบบ Smart Cards นั่นหมายถึงว่าผู้บริโภคสามารถที่จะตั้งโปรแกรมควบคุมระยะไกลกับสวิทช์ไฟฟ้าผ่านทางอุปกรณ์ช่วย (PDAs) เล็กๆ เหล่านี้ซึ่งขณะนี้มีนักประดิษฐ์จำนวนมากต่างก็พยายามที่จะสร้างสรรค์อุปกรณ์แปลกๆ ใหม่ๆ ขึ้นมา ซึ่งจะทำให้จาวามีศักยภาพสูงสุด ในการเป็นเครื่องมือสนับสนุนอุปกรณ์เหล่านี้

บทที่ 3
ความรู้ทั่วไปเกี่ยวกับการเข้ารหัสข้อมูล

3.1 ระบบการเข้ารหัสข้อมูล

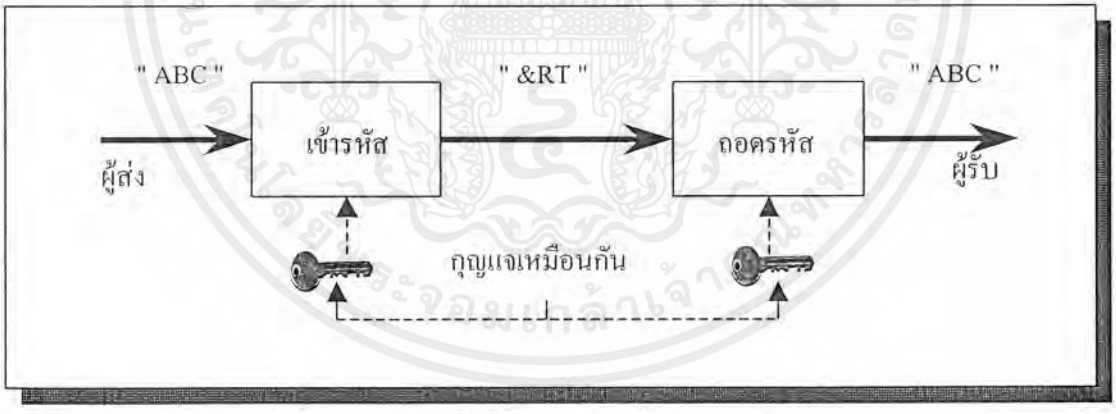
การเข้ารหัสข้อมูลเป็นการทำให้ข้อมูลที่ต้องการเป็นความลับ ซึ่งเป็นส่วนสำคัญในระบบข้อมูลปัจจุบัน โดยอาศัยหลักการของการเข้ารหัส (Encryption) และการถอดรหัส (Decryption)

:- การเข้ารหัส เป็นการเปลี่ยนรูปข้อมูลโดยผ่านรูปแบบและกระบวนการแปรรูปของข้อมูล ทำให้ข้อมูลที่ส่งมีรูปแบบที่ไม่เหมือนเดิม เพื่อทำให้ข้อมูลเป็นความลับ

:- การถอดรหัส เป็นการแปลงข้อมูลที่ได้ผ่านการเข้ารหัสให้กลับมาเป็นข้อมูลเดิม ซึ่งการเข้ารหัสและการถอดรหัส จะถูกควบคุมโดยกุญแจหรือที่เรียกว่า "Key"

3.1.1 ระบบการเข้ารหัสโดยใช้กุญแจเดียว (symmetric key)

การเข้ารหัสข้อมูลระบบนี้จะทั้งผู้รับและผู้ส่งจะต้องมีกุญแจที่เป็นความลับ ที่เหมือนกันในการเข้าและถอดรหัสข้อมูล ซึ่งหากกุญแจที่มีต่างกัน ก็จะทำให้ข้อมูลที่สื่อสารกันผิดพลาด ตัวอย่างการเข้ารหัสระบบนี้ได้แก่ การเข้ารหัสแบบ DES, 3DES, IDEA เป็นต้น



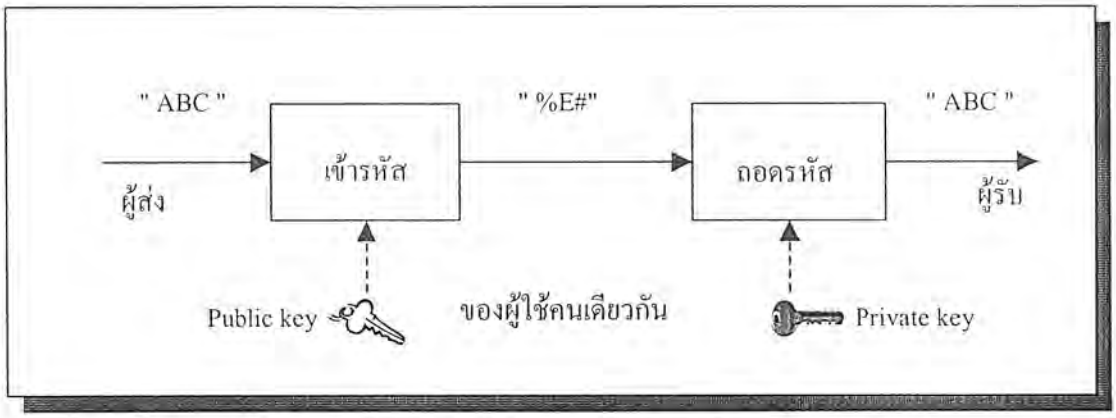
รูปที่ 3.1 แสดงการเข้ารหัสและถอดรหัสโดยใช้กุญแจเดียว

3.1.2 ระบบการเข้ารหัสแบบกุญแจสาธารณะ

การเข้ารหัสข้อมูลระบบนี้จะประกอบด้วยกุญแจ 2 อัน คือ

1. กุญแจส่วนตัว (private key) เป็นกุญแจที่จะต้องเก็บเป็นความลับ
2. กุญแจสาธารณะ (public key) เป็นกุญแจที่สามารถเปิดเผยให้ผู้อื่นทราบได้

ซึ่งกุญแจทั้งสองนี้จะเป็นกุญแจที่ต่างกัน การมีวิธีในการเข้าและถอดรหัส ดังรูป 3.2



รูปที่ 3.2 แสดงการเข้ารหัสและถอดรหัสโดยใช้กุญแจสาธารณะ

3.2 การเข้ารหัสแบบ DES (Data Encryption Standard)

3.2.1 ประวัติและที่มาของ DES

ในปลายทศวรรษที่ 1960 บริษัท IBM ได้จัดตั้งโครงงานวิจัยทางด้าน Computer Cryptography ซึ่งนำโดย Horst Feistel ซึ่งโครงงานนี้เสร็จสิ้นในปี 1971 ซึ่งผลงานวิจัยของโครงงานนี้คือ LUCIFER [FEIS73] โดยมีลักษณะเป็นการเข้ารหัสข้อมูลเป็นบล็อคขนาด 64 บิตและใช้คีย์ขนาด 128 บิต ซึ่งต่อมาได้ถูกพัฒนาขนาดของคีย์ให้ลดลงเหลือขนาด 56 บิต

โดยอัลกอริทึมของการเข้ารหัสข้อมูลของ Lucifer ได้ถูกพัฒนาโดย IBM สำหรับ NBS (National Bureau of Standards) อัลกอริทึมนี้ได้เป็นที่รู้จักในนามของ DES (Data Encryption Standard) ถึงแม้ว่าชื่อจริงของมัน คือ DEA (Data Encryption Algorithm) ในสหรัฐและ DEAI (Data Encryption Algorithm-1) ในอีกหลายๆ ประเทศ

3.2.2 รายละเอียดของ DES

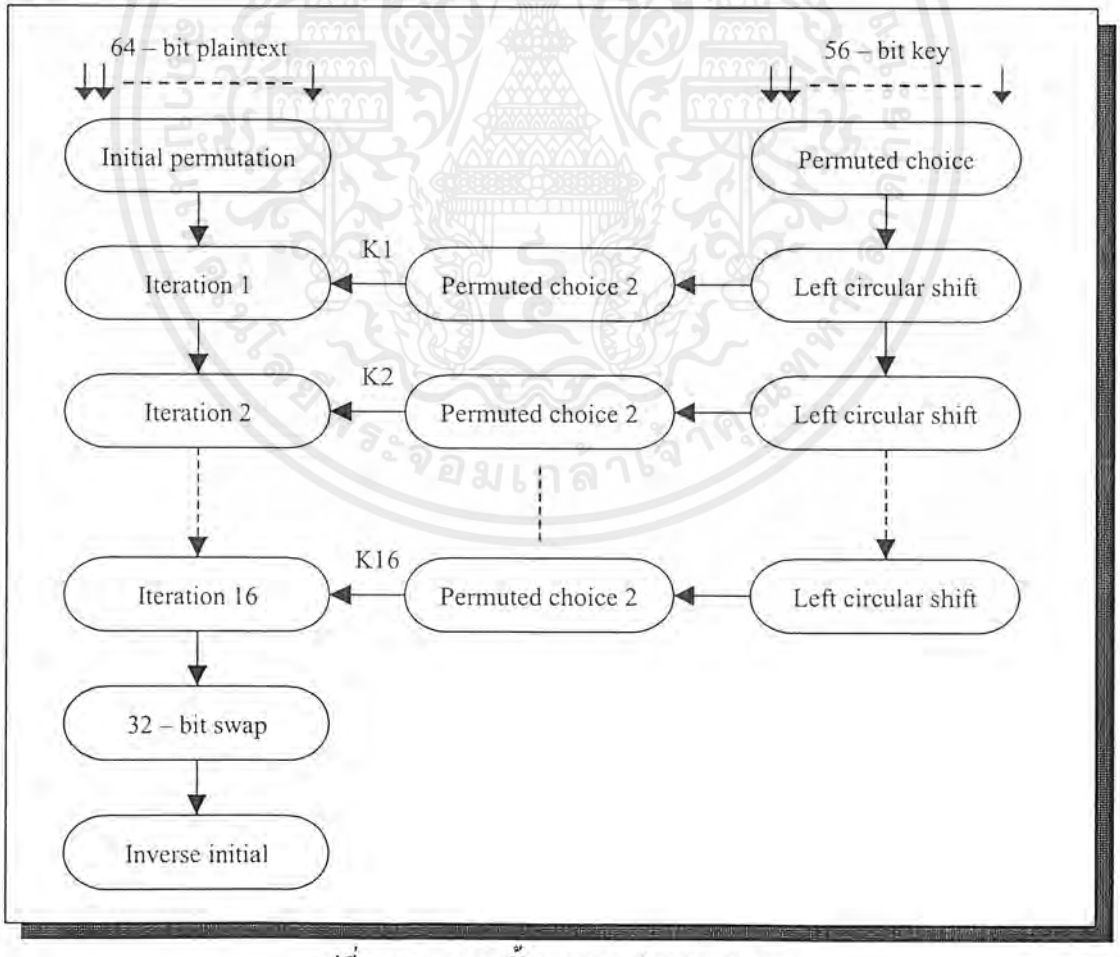
เป็นวิธีการเข้ารหัสที่ใช้กันอย่างแพร่หลายที่เป็นพื้นฐานบน Data Encryption Standard (DES) ที่ได้พัฒนาขึ้นในปี 1977 โดย National Bureau of Standards ซึ่งปัจจุบันคือ Federal Information Processing Standard 46 (FIPS PUB46) สำหรับ DES ข้อมูลจะถูกเข้ารหัสเป็นบล็อกขนาด 64 บิตซึ่งใช้คีย์ขนาด 56 บิต โดยวิธีการจัดการกับข้อมูล 64 บิตที่เข้ามาเพื่อแปลงเป็น 64 บิตเข้ามูลออกไป และใช้คีย์ตัวเดียวกันนี้ในการถอดรหัส

แม้ว่า DES ถูกนำมาใช้ตั้งแต่ช่วงทศวรรษที่ 70 (ค.ศ. 1960 – 1970) และได้รับการตอบรับอย่างดีจาก เหล่านักวิเคราะห์รหัส (Cryptanalysis) อย่างแพร่หลาย แต่ก็เป็นข้อถกเถียงกันเป็นอย่างมากถึงเรื่อง DES นั้นจะปลอดภัยได้หรือไม่และมีความปลอดภัยมากน้อยแค่ไหน แต่จนถึงปัจจุบันเราก็ยังไม่พบช่องโหว่ของ DES ตามเอกสารที่ตีพิมพ์เป็นสาธารณะ แม้ว่าจะใช้คีย์เพียงไม่กี่บิตก็ตาม ในทางตรงกันข้ามแนวความคิดแบบ IDEA กลับใช้คีย์แบบ 128 บิต (ซึ่งขนาดกว่า 2 เท่าของ DES) และได้รับการตอบรับจากสาธารณะตั้งแต่ทศวรรษ 90 (ค.ศ. 1980 – 1990) (แต่ดีไม่เท่าตอนประกาศใช้ DES) IDEA มีความปลอดภัยมากกว่า DES และสามารถประมวลผลได้เร็วกว่า DES อย่างไรก็ตาม IDEA ยังต้องรอการตรวจ

สอบจากผู้เชี่ยวชาญอีกมากถึงเรื่องช่องโหว่ของความปลอดภัย สำหรับบทนี้จะไม่กล่าวถึง IDEA (เพราะมีความใกล้เคียงกับ DES)

การทำงานของ DES จะมีลักษณะดังรูป ข้อมูลที่เข้ามาในส่วนฟังก์ชันของการเข้ารหัสจะมี 2 ส่วนด้วยกันคือ ข้อมูลที่ยังไม่ถูกเข้ารหัสขนาด 64 บิตและคีย์ซึ่งมีขนาด 56 บิต ซึ่งรูปในด้านซ้ายมือจะแสดงขั้นตอนจัดการกับข้อมูลที่ยังไม่เข้ารหัส โดยสามารถแบ่งย่อยๆได้อีก 3 เฟสด้วย

- เฟส 1 จะทำการจัดการกับข้อมูลที่ไม่ได้ผ่านการเข้ารหัสที่มีขนาด 64 บิตผ่านเข้าไปยังส่วนที่เรียกว่า initial Permutation (IP) ซึ่งจะทำการเรียบเรียงบิตใหม่เพื่อผลิต "permuted input" ข้อมูลที่มีการสลับตำแหน่ง
- เฟสที่ 2 จะทำการฟังก์ชันเดียวกัน 16 ครั้งซึ่งฟังก์ชันนี้ร่วมทั้งการทำ permutation และ substitution ซึ่งผลลัพธ์ที่ได้จากการทำทั้งหมด 16 ครั้งนี้จะได้ข้อมูลขนาด 64 บิตโดยใช้ทั้งข้อมูลที่ไม่ได้ผ่านการเข้ารหัสและคีย์ในการทำ ซึ่งข้อมูลที่มีขนาด 64 บิตที่ได้นี้แบ่งเป็น 2 ด้านคือซ้ายและขวา ทั้งหมดจะถูก swap เพื่อผลิต 64 บิตที่เป็น preoutput
- เฟสที่ 3 จะนำ preoutput ผ่านเข้าไปยังส่วนที่เรียกว่า "Inverse initial permutation" :IP^{-1} ซึ่งทำหน้าที่ inverse ตัว initial permutation function ซึ่งทั้งหมดจะผลิต 64 บิตที่เรียกว่า ข้อมูลที่ผ่านการเข้ารหัสแล้ว (Ciphertext)



รูปที่ 3.3 แสดงการขั้นตอนการทำงานของ DES

3.2.3 การทำ Initial Permutation

การทำ initial permutation และการทำ inverse initial permutation จะถูกอธิบายโดยใช้ตารางด้านล่างตามลำดับ ซึ่งจะเห็นได้ว่าฟังก์ชัน permutation ทั้ง 2 เป็นส่วนกลับซึ่งกันและกัน พิจารณาจาก 64 บิต:M ที่เข้ามา

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}	M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}	M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}	M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

M_i คือ เป็นตัวเลขฐานสอง เมื่อทำการ permutation $X = IP(M)$ จะได้ดังนี้

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2	M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6	M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1	M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5	M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

ถ้าเราทำการ inverse permutation $Y = IP^{-1}(X) = IP^{-1}(IP(M))$ เราจะสามารถเห็นลำดับในการเรียงของบิตที่มีรูปแบบดังเดิม

3.2.4 รายละเอียดของการทำฟังก์ชันในแต่ละรอบ

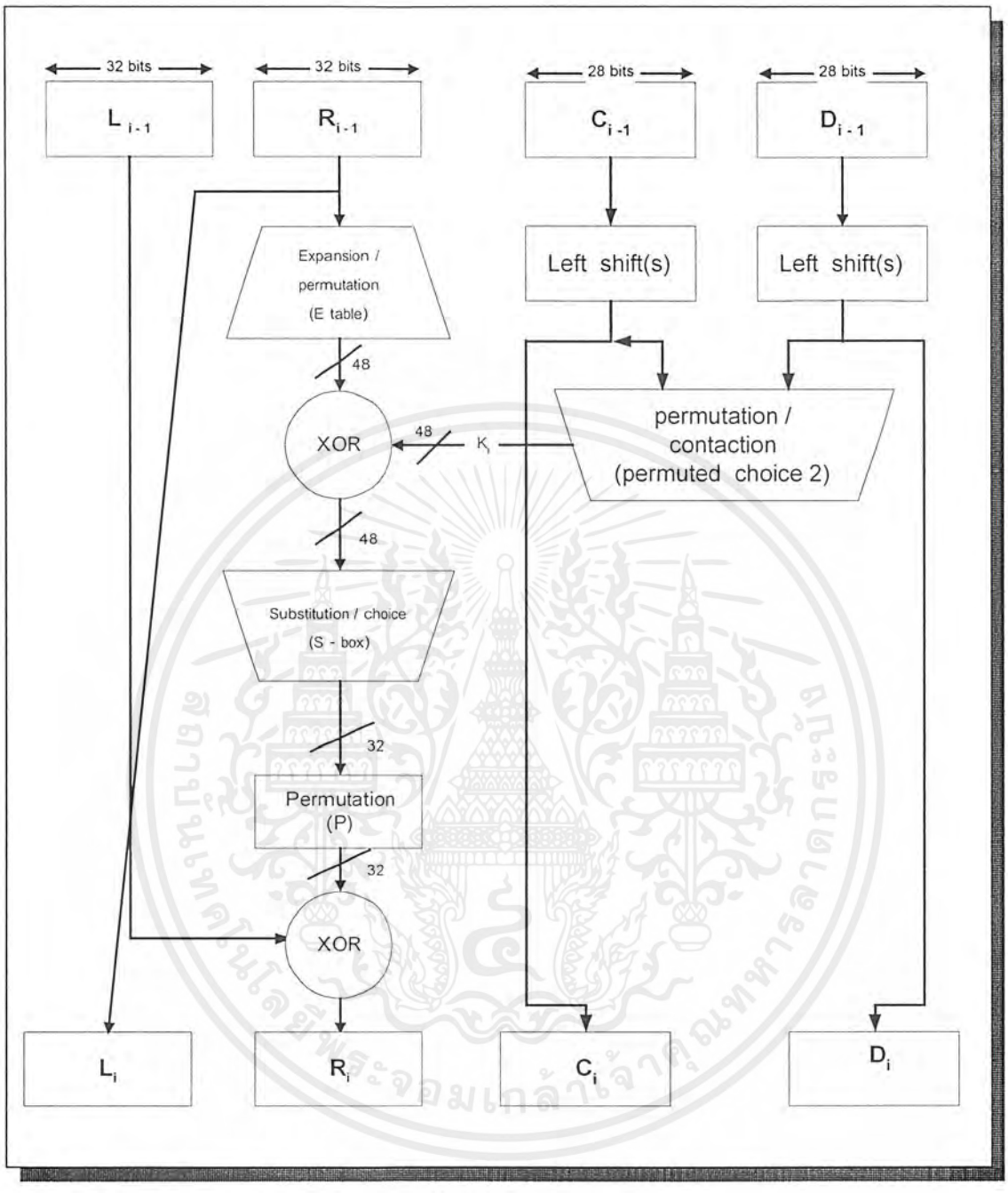
ข้อมูลที่เข้ามาที่มีขนาด 64 บิต โดยจะทำการแบ่งข้อมูลเป็น 2 ส่วนด้วยกันขนาดละ 32 บิตเท่ากัน (แบ่งเป็นซ้ายกับขวา) ซึ่งกระบวนการทำในแต่ละครั้งสามารถสรุปเป็นสูตรได้ดังนี้

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

เมื่อ $\oplus$ หมายถึง การทำ XOR function

จากสูตรจะเห็นได้ว่า 32 บิตด้านซ้ายมือ(L_i) จะเท่ากับด้านขวาของ (R_{i-1}) รอบที่ผ่านมา โดย R_i จะเท่ากับการนำ L_{i-1} มา XOR กับ $f(R_{i-1}, K_i)$ ซึ่งฟังก์ชัน f จะแสดงดังรูปที่ 3.4



รูปที่ 3.4 แสดงการเข้ารหัส DES ในแต่ละครั้ง (ทั้งหมดทำ 16 ครั้ง)

(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (IP^{-1})

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

(c) Expansion Permutation(E)

Output bit	1	2	3	4	5	6	7	8	9	10	11	12
Form input bit	32	1	2	3	4	5	4	5	6	7	8	9
Output bit	13	14	15	16	17	18	19	20	21	22	23	24
Form input bit	8	9	10	11	12	13	12	13	14	15	16	17
Output bit	25	26	27	28	29	30	31	32	33	34	35	36
Form input bit	16	17	18	19	20	21	20	21	22	23	24	25
Output bit	37	38	39	40	41	42	43	44	45	46	47	48
Form input bit	24	25	26	27	28	29	28	29	30	31	32	1

(d) Permutation Function (P)

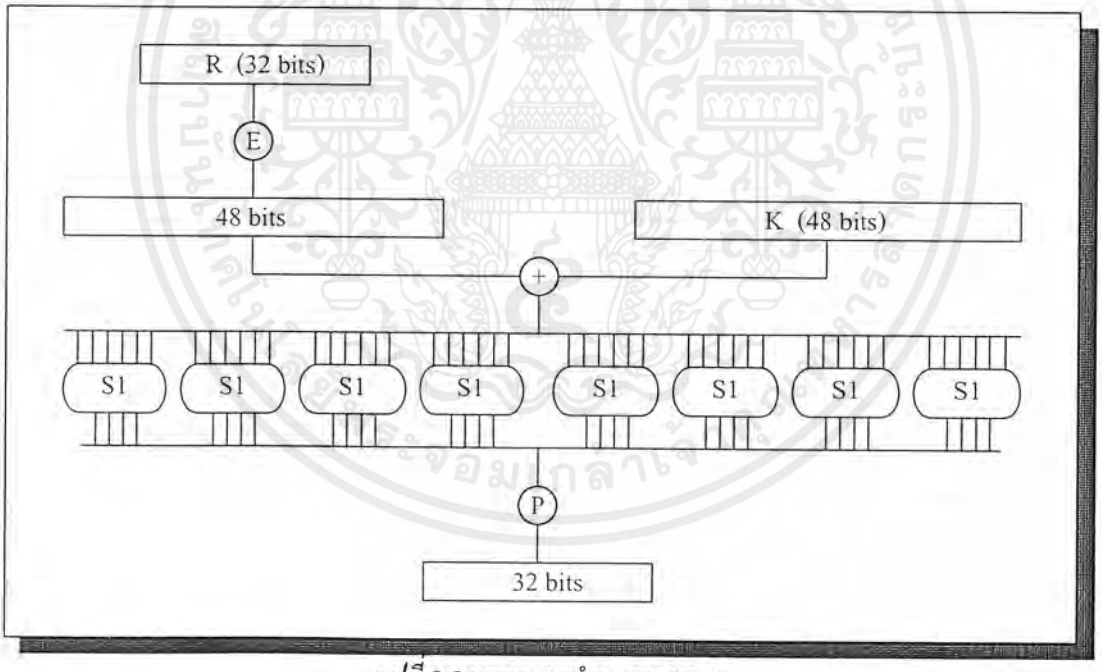
Output bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Form input bit	16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
Output bit	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Form input bit	2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

ตารางที่ 3.1 แสดง permutation ของ DES

โดยคีย์ K_i ที่ใช้ในแต่ละรอบจะมีขนาด 48 บิตและอินพุตด้านขวา(R)มีขนาด 32 บิต ดังนั้นจึงต้องมีการขยายขนาดจาก 32 บิตให้เป็น 48 บิต โดยใช้ตารางที่ได้มีการกำหนด permutation และ expansion ซึ่งรวมทั้งการจำลอง 16 บิตที่เพิ่มขึ้นมาของด้านขวา(R) ซึ่งจะนำผลที่ได้ที่มีขนาด 48 บิตจะถูก XOR กับ K_i โดยจะนำผลที่ได้ผ่านไปยังฟังก์ชันที่เรียกว่า Substitution และ Permutation ที่สามารถผลิตผลลัพธ์ที่มีขนาด 32 บิต

โดย Substitution จะประกอบด้วยเซตของ S – box 8 อัน(S_1 - S_8) ซึ่ง S – box จะมีอินพุตขนาด 6 บิตและผลิตเอาต์พุตขนาด 4 บิต ซึ่งตารางด้านล่างจะแสดง DES S – box โดยมีวิธีการในการแปลงอินพุตขนาด 6 บิตให้กลายเป็นเอาต์พุตขนาด 4 บิตดังนี้คือ การนำบิตแรกและบิตสุดท้ายของอินพุตมาทำเป็นตำแหน่งของแถวและนำ 4 บิตตรงกลางมาเป็นตำแหน่งของคอลัมน์ เช่น S_1 มีค่าเท่ากับ 011011 (ขนาด 6 บิต) เราจะนำบิตแรกและบิตสุดท้ายซึ่งก็คือ 0 และ 1 มาเป็นตำแหน่งของแถวจะได้แถวที่ 01 หรือถือแถว 1และ 4 บิตตรงกลางที่เหลือคือ 1101 จะได้ตำแหน่งของคอลัมน์คือ คอลัมน์ที่ 13 ดังนั้นค่าที่ตำแหน่งแถวที่ 1 และคอลัมน์ที่ 13 ในตารางคือ 0101

รูปด้านล่างจะมีรายละเอียดสำหรับ S – box operation ซึ่งในรูปจะแสดงการทำ permutation สำหรับ row 0 ของ S_1



รูปที่ 3.5 แสดงการคำนวณ f(R,K)

Column Number

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Box
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	S_2
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	

Column Number

Box

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	12	15	1	2	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	11	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	1	
1	10	15	4	2	7	12	9	5	6	1	12	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	

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

S_3

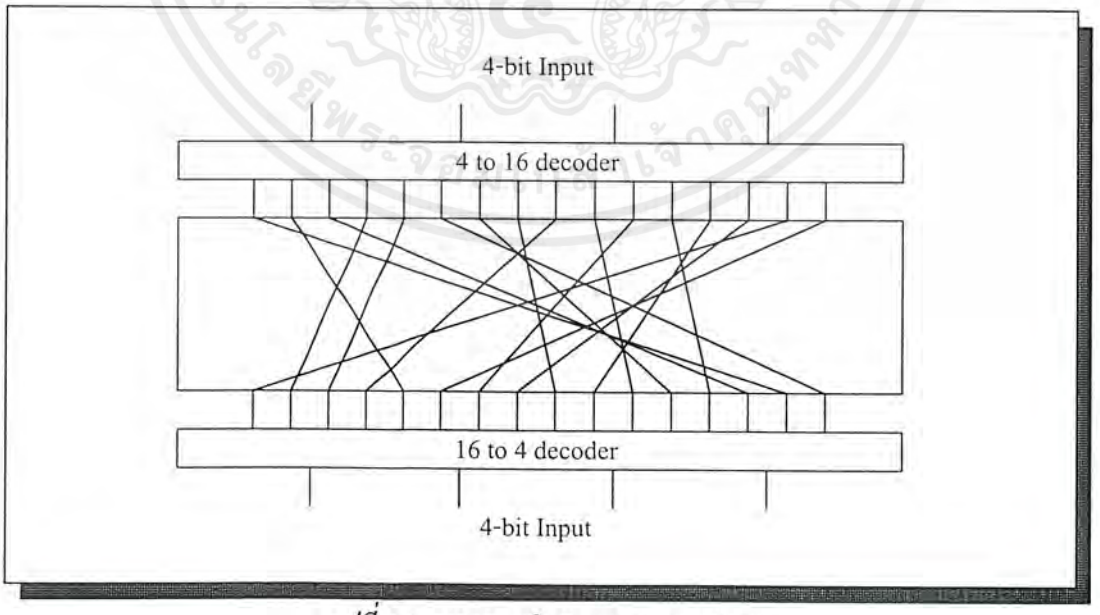
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S_4

ตารางที่ 3.2 แสดง S- box

3.2.5 การสร้างคีย์ (Key Generation)

ในรูปที่แสดงถึงการทำงานในแต่ละรอบของ DES เราจะเห็นว่าคีย์ที่ใช้มีขนาด 56 บิตซึ่งเป็นอินพุตในการทำ permutation โดยตาราง Permuted Choice One ดังรูป โดยเริ่มแรกจะทำการแบ่ง 56 บิตเป็น 2 ส่วนเท่าๆ กันส่วนละ 28 บิตโดยให้ชื่อในแต่ละส่วนว่า C กับ D ซึ่งในแต่ละรอบ (ทั้งหมด 16 รอบ) จะมีการทำ circular left shift ในแต่ละส่วนของ C และ D หรือทำ rotation โดยในแต่ละรอบจะมีการกำหนดว่าจะให้ shift ไปกี่บิตดังตาราง ซึ่งค่าที่ถูก Shift จะกลายเป็นอินพุตของการทำให้รอบถัดไปและเป็นอินพุตของการทำ Permutation Choice Two ดังในตาราง หลังจากการทำ Permutation Choice Two แล้วจะได้เอาต์พุตขนาด 48 บิต ซึ่งเป็นอินพุตของ $f(R_{i-1},K_i)$



รูปที่ 3.6 แสดงการทำ Permutation Choice

(a) Permuted Choice One(PC-1)

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

(b) Permuted Choice Two (PC-2)

Output bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
From input bit	14	17	11	24	1	5	3	28	15	6	21	10	23	19	12	4
Output bit	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
From input bit	26	8	16	7	27	20	13	2	41	52	31	37	47	55	30	40
Output bit	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
From input bit	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32

(c) Schedule of Left Shifts

Iteration number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

ตารางที่ 3.3 แสดงการสร้างคีย์

3.2.6 การถอดรหัสข้อมูล DES

กระบวนการถอดรหัสโดยใช้ DES นั้นเหมือนกับขั้นตอนในการเข้ารหัส ซึ่งมีขั้นตอนในการทำงานดังนี้คือ นำข้อมูลที่ผ่านการเข้ารหัสแล้ว(Ciphertext) มาเป็นอินพุตแต่จะมีการใช้คีย์(K_i) ที่มีลำดับย้อนกลับกับคีย์ที่ใช้ในการเข้ารหัสเช่น K_{16} , K_{15} .....เป็นคีย์แรกและคีย์ถัดไปในการเข้ารหัสแทน ดังรูปด้านซ้ายมือเป็นขั้นตอนการเข้ารหัสและด้านขวามือเป็นขั้นตอนในการถอดรหัส

เราจะแสดงถึงผลลัพธ์ของขั้นตอนแรกในการกระบวนการถอดรหัส ซึ่งจะเท่ากับ 32 บิตที่ถูก Swap จากอินพุตของการทำทั้งหมด 16 รอบของการเข้ารหัส เริ่มจาก

$$L16 = R15$$

$$R16 = L15 \oplus f(R15,K16)$$

ในด้านการถอดรหัส

$$Ld1 = Rd0 = L16 = R15$$

$$Rd1 = Ld0 \oplus f(Rd0,K16)$$

$$= R16 \oplus f(R15,K16)$$

$$= [L15 \oplus f(R15,K16)] \oplus f(R15,K16)$$

ซึ่งคุณสมบัติของ XOR ที่สำคัญคือ

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$

$$D \oplus D = 0$$

$$E \oplus 0 = E$$

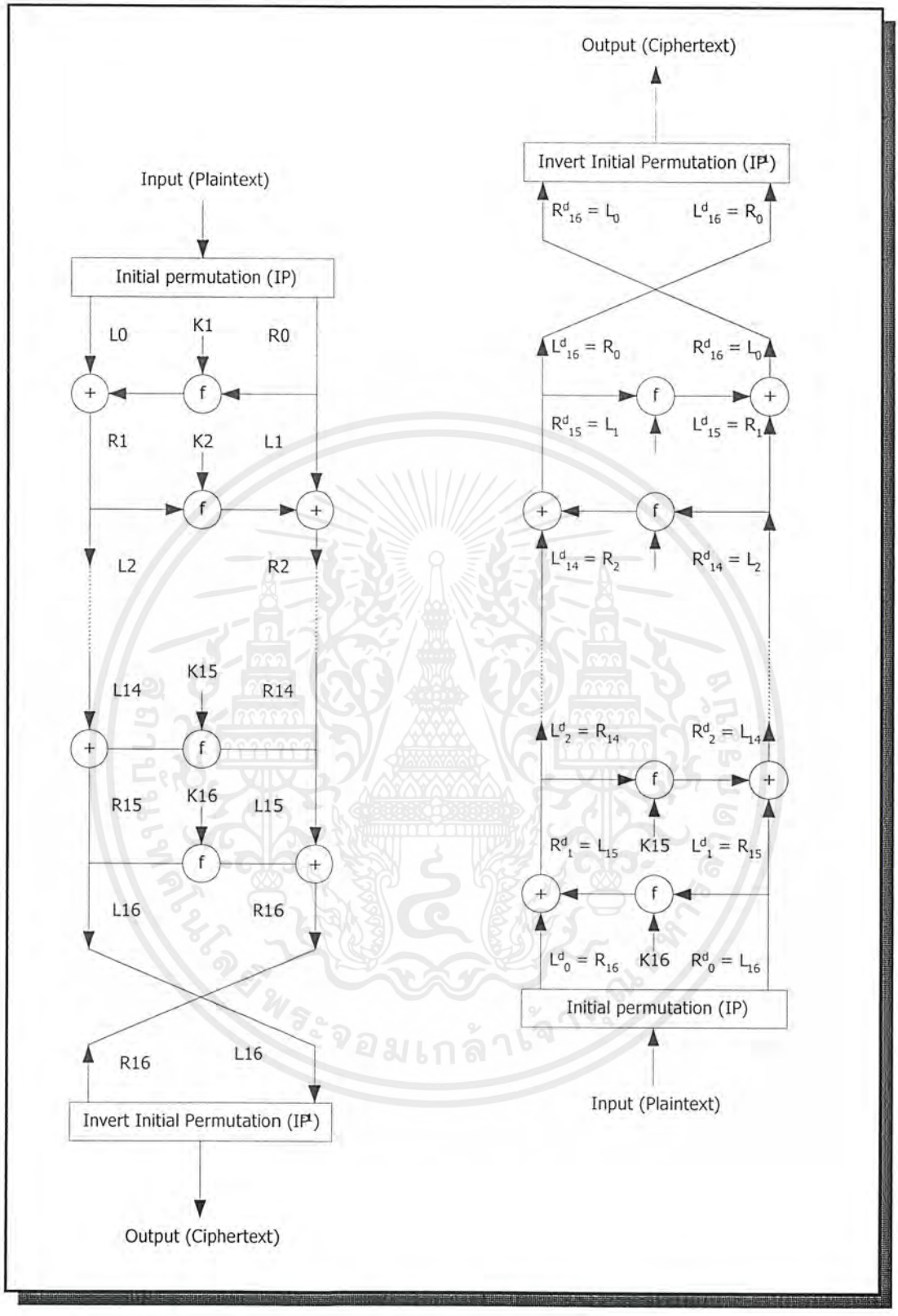
ดังนั้นเรามี Ld1 = R15 และ Rd1 = L15 จะได้เอาต์พุตของขั้นตอนแรกในการถอดรหัสคือ L15||R15 ซึ่งเราสามารถเขียนเป็นสมการในการถอดรหัสได้ดังนี้คือ

$$Ri^{-1} = Li$$

$$Li^{-1} = Ri \oplus f(Ri-1,Ki) = Ri \oplus f(Li,Ki)$$

ซึ่งผลสุดท้ายเอาต์พุตที่ได้จากขั้นตอนสุดท้ายในการถอดรหัสคือ R0||L0 และนำไปสู่ขั้นตอนในการทำ Inverse Permutation เราจะได้ข้อมูลที่ส่งมา(plaintext) ดังสมการ

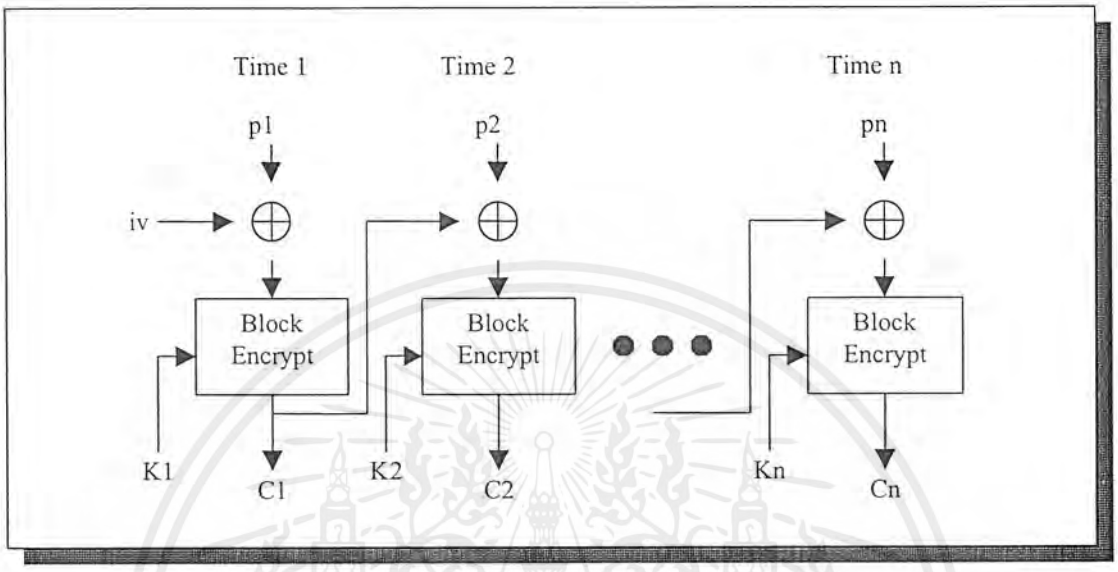
$$IP^{-1}(L0||R0) = IP^{-1}(IP(plaintext)) = plaintext$$



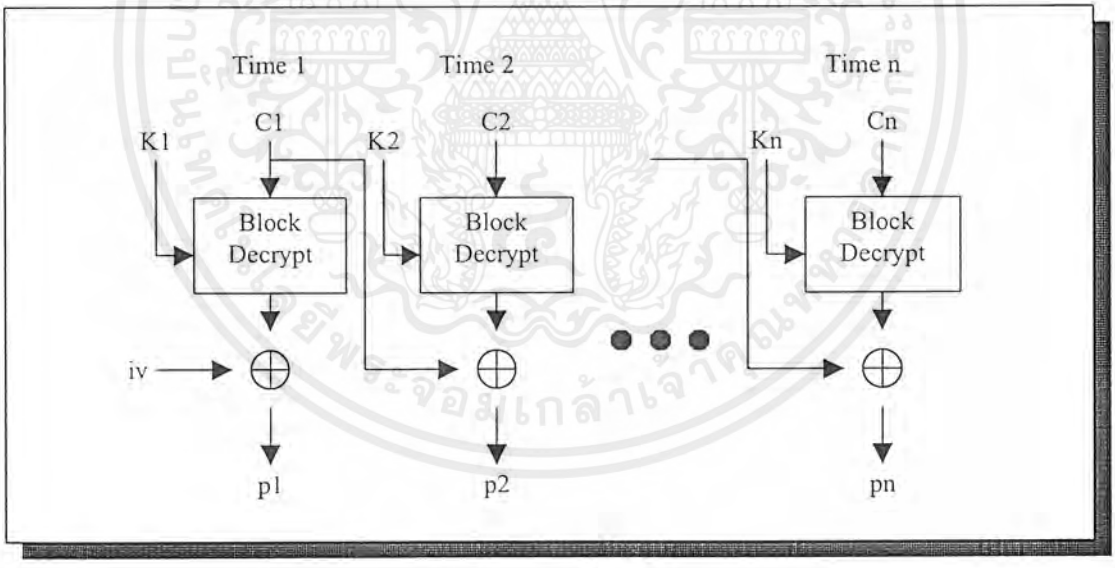
รูปที่ 3.7 แสดงคีย์และขั้นตอนการเข้ารหัสและถอดรหัส DES

3.2.7 โหมด CBC (Cipher Block Chaining)

เป็นวิธีการเข้ารหัสที่พัฒนามาจาก DES ช่วยทำให้ข้อมูลที่ส่งยิ่งมีความปลอดภัยมากยิ่งขึ้น โดยมีวิธีการคือ จะนำข้อมูลที่ผ่านการเข้ารหัสของข้อมูลตัวก่อนมา XOR กับข้อมูลที่ยังไม่ได้เข้ารหัสของตัวถัดไปก่อนจะทำการเข้ารหัสแบบ DES ตามปกติดังรูป



การเข้ารหัส



การถอดรหัส

รูปที่ 3.8 แสดงการเข้ารหัสและถอดรหัสของ DES CBC

ในการถอดรหัสข้อมูล ก็จะทำเช่นเดียวกับการถอดรหัสข้อมูล DES ตามปกติแต่จะนำผลที่ได้จากการทำถอดรหัสมา XOR กับข้อมูลที่ผ่านการเข้ารหัส(Ciphertext)ตัวก่อนหน้านี้ เพื่อจะได้ข้อมูลจริงๆ (plaintext)ดังสมการ

$$Cn = Ek[Cn^{-1} \oplus Pn]$$

สมการในการถอดรหัส

$$Dk[Cn] = Dk[Ek(Cn^{-1} \oplus Pn)]$$

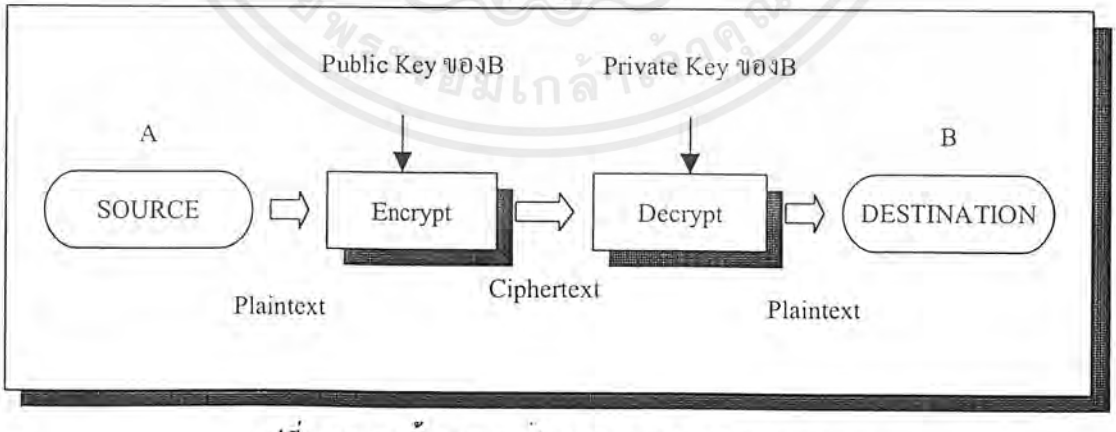
$$Dk[Cn] = Cn^{-1} \oplus Pn$$

$$Cn^{-1} \oplus Dk[Cn] = Cn^{-1} \oplus Cn^{-1} \oplus Pn = Pn$$

จะสังเกตเห็นได้ว่าบล็อกแรกของข้อมูลที่ผ่านการเข้ารหัส(Ciphertext) จะมีการนำ iv (Initialization Vector) มาทำการ XOR กับข้อมูลที่ไม่ได้เข้ารหัส(plaintext) ก่อนจะมีการเข้ารหัส DES ตามปกติ และในส่วนของการถอดรหัสข้อมูลก็จะใช้ iv ในการถอดรหัสข้อมูลเช่นเดียวกัน ดังนั้นจึงจำเป็นต้องมีข้อตกลงกันระหว่างผู้ส่งกับผู้รับก่อนว่าจะใช้ iv เป็นค่าใด เพื่อให้มีความปลอดภัยสูงสุด iv ควรจะมีการป้องกันเช่นเดียวกับ Key ซึ่งในการส่งค่า iv อาจจะทำการส่งโดยใช้การเข้ารหัสแบบ ECB

3.3 การเข้ารหัสแบบ RSA

รหัส RSA ถูกคิดค้นขึ้นในปี 1977 โดยที่ชื่อ RSA ได้มาจากอักษรตัวแรกของนามสกุลของผู้ร่วมกันคิดค้นคือ Ron River, Adi Shamir และ Leonard Adleman เป็นวิธีการเข้ารหัสแบบกุญแจสาธารณะที่เรียกว่าพับลิคคีย์ (Public–Key Cryptosystem) เพื่อแก้ไขปัญหาการทำให้กุญแจเป็นความลับ(Secret–Key Cryptosystem) โดยสมาชิกแต่ละคนจะต้องมีกุญแจ 2 ชนิดคือ กุญแจส่วนตัวหรือไพรเวทคีย์(Private Key) และกุญแจสาธารณะหรือพับลิคคีย์(Public Key) โดยกุญแจส่วนตัวจะถูกเก็บไว้เป็นความลับ ส่วนกุญแจสาธารณะนั้นจะเปิดเผยให้ใครก็ได้ที่ต้องการส่งเอกสารให้แก่ตน การทำงานของรหัสลับมีหลักการว่าข้อมูลที่ถูกเข้ารหัสลับด้วยกุญแจสาธารณะของผู้ใด จะถูกถอดรหัสได้ด้วยกุญแจส่วนตัวของผู้นั้นเท่านั้น การทำงานของระบบพับลิคคีย์สามารถอธิบายได้ดังต่อไปนี้



รูปที่ 3.9 แสดงขั้นตอนการทำ Public – Key Cryptosystem

3.3.1 หลักการทำงานของ RSA

ถ้าให้ p และ q เป็นจำนวนเฉพาะที่มีค่ามาก โดยที่ n = p*q เรียกว่าโมดูลัส(modulus) จากนั้นจึงเลือก e ที่มีค่าน้อยกว่า n และไม่สามารถหาร (p-1)(q-1) ได้ลงตัว ถ้าให้ d เป็นส่วนกลับของ e ในคณิตศาสตร์ระบบโมดูโลฐาน (p-1)(q-1) นั่นคือ

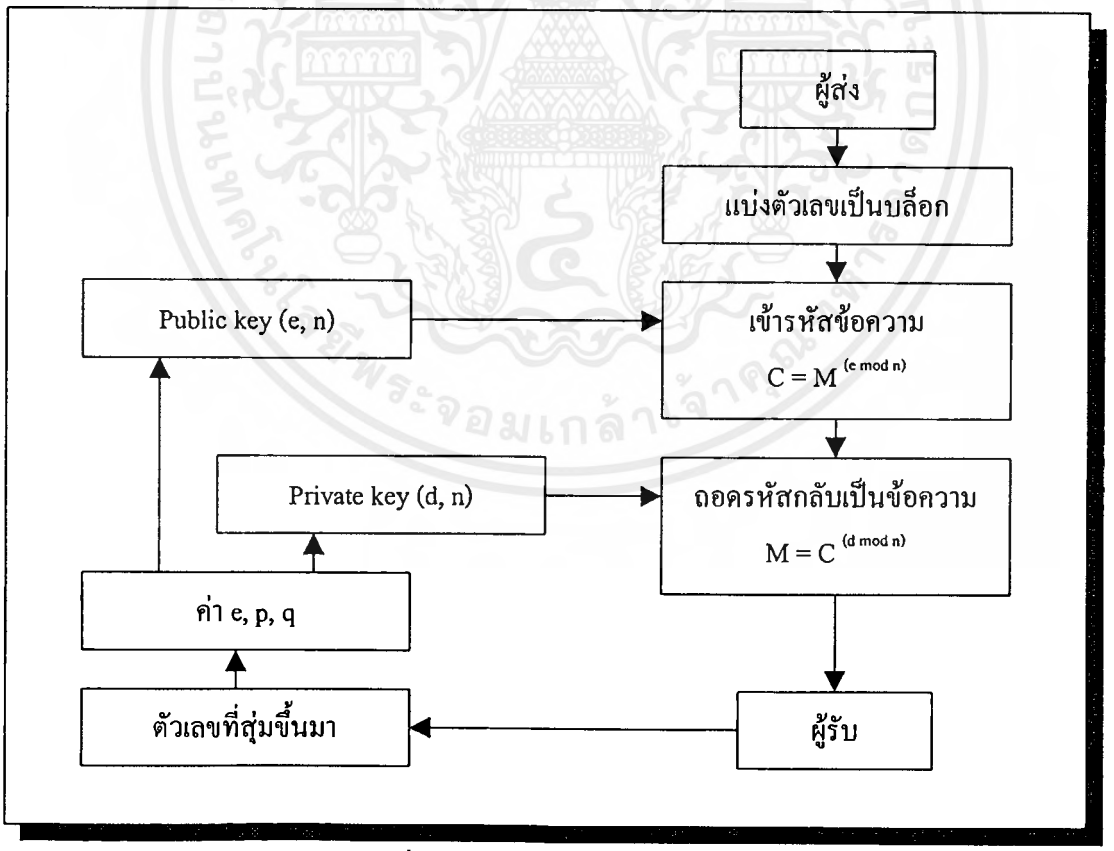
$$e*d \bmod (p-1)(q-1) = 1 \quad \text{.....(1)}$$

ในรหัส RSA นั้น (n,e) คือพับลิคคีย์(public key) ส่วน d คือไพรเวทคีย์(private key) เมื่อได้ค่าเหล่านี้แล้ว p,q จะต้องเก็บเป็นความลับหรือถูกทำลายในทันที

ถ้าอริสาต้องการส่งข้อเอกสารส่วนตัว m ไปให้บุญมี อริสาจะสร้างรหัสลับ c ของ m ได้โดยการให้ $c = m^e \bmod n$ เมื่อ(n,e) เป็นพับลิคคีย์ของบุญมี เมื่อบุญมีได้รับเอกสารรหัสลับ c เขาจะถอดรหัสเพื่ออ่านเอกสาร m ได้ด้วย d เพราะความสัมพันธ์ในสมการ(1) ระหว่าง e,d และ n จะทำให้

$$c^d = m^{ed \bmod (p-1)(q-1)} \bmod n = m \quad \text{.....(2)}$$

และเพราะมีแต่บุญมีเท่านั้นรู้ค่า d บุญมีเท่านั้นจะถอดรหัสได้ คุณสมบัติสำคัญประการหนึ่งของ RSA คือจากสมการ (2) ในทางกลับกันถ้าเราเข้ารหัสด้วยไพรเวทคีย์ เราก็สามารถถอดรหัสด้วยพับลิคคีย์ได้ด้วยเช่นกัน คุณสมบัติข้อนี้เองที่ทำให้ RSA มีประโยชน์มากเพราะในการใช้การเซ็นระบบดิจิตอลได้อีกด้วย



รูปที่ 3.10 แสดงการเข้ารหัสแบบ RSA

ตัวอย่างขั้นตอนการเข้ารหัสแบบ RSA ซึ่งมีขั้นตอนในการหา Key ดังนี้

1. เลือกจำนวนเฉพาะสองจำนวน คือ p = 7 , q = 17

2. คำนวณ n = p*q = 7 * 17 = 119

3. คำนวณ (p-1)(q-1) = 96

4. เลือก e ซึ่งมีความสัมพันธ์กับค่า (p-1)(q-1) ที่ได้กล่าวไว้ข้างต้น ในที่นี้เราจะใช้ 3

5. คำนวณค่า d ซึ่งสัมพันธ์กับสมการ e * d = 1 mod 96 ซึ่งค่าที่ถูกต้องคือ d = 77 เนื่องจาก

77 * 5 = 385 = 4 * 96 +1

ดังนั้น Public Key คือ {5,119} และมีค่า Private Key คือ {77,119}

วิธีทำลายรหัส RSA ที่รู้จักกันดีที่สุดคือการหาค่า d นั่นเองจากสมการที่ (1) เราอาจหาค่า d ได้ หากรู้ค่า p และ q แต่เนื่องจาก p และ q เป็น prime number ที่ p*q = n ดังนั้นการทำลายรหัส RSA ขึ้นอยู่กับการแยกตัวประกอบของ n นั่นเอง แต่วิธีการแยกตัวประกอบของ n นั้นไม่ง่ายเลยสำหรับหาก n มีค่ามาก ดังนั้นสำหรับข้อมูลที่มีความสำคัญมากกว่า อาจจำเป็นต้องใช้ n ที่มีค่ามากถึง 700 หรือ 1000 ก็ได้

จะเห็นได้ว่าไม่ว่าการเข้าหรือถอดรหัส RSA ก็จำเป็นต้องใช้การยกกำลังในระบบโมดูโล ซึ่งการยกกำลังนั้นสามารถทำได้โดยการใช้วงจรคูณระบบโมดูโลมาต่ออนุกรมกัน ดังนั้นความเร็วของทั้งการเข้าและถอดรหัสจึงขึ้นกับความเร็วของวงจรคูณระบบโมดูโลเป็นอย่างมาก นี่เองเป็นจุดอ่อนข้อหนึ่งของ RSA เมื่อเปรียบเทียบกับซีเคทคีย์ เช่น DES เพราะปัจจุบันการคูณในระบบโมดูโลยังค่อนข้างยุ่งยากเมื่อเปรียบเทียบกับการแทนค่าหรือสลับตำแหน่งใน DES ได้มีการประมาณกันว่าสำหรับ n ที่มีความยาว 512 บิต การเข้ารหัสแบบ DES จะเร็วกว่า RSA ประมาณ 100 เท่า ถ้าเราทำการเข้ารหัสด้วยซอฟต์แวร์ และอาจเร็วกว่าถึง 1000 ถึง 10000 แล้วแต่ลักษณะของวงจร หากทำการเข้ารหัสด้วยฮาร์ดแวร์ โดยทั่วไปแล้วเราต้องการให้การเข้ารหัสเร็วกว่าการถอดรหัสดังนั้นเราจึงมักเลือกให้ e มีค่าน้อยกว่า d และยิ่งกว่านั้นเรามักให้ e ของสมาชิกทุกคนมีค่าเดียวกันเพื่อให้ฮาร์ดแวร์ของวงจรเข้าสำหรับสมาชิกแต่ละคนมีลักษณะคล้ายกัน

เนื่องจาก DES และ RSA มีข้อดีข้อเสียที่แตกต่างกัน จึงไม่จำเป็นว่ารหัสชนิดใดชนิดหนึ่งจะเหมาะสมในทุกสถานการณ์ โดยทั่วไปแล้ว DES จะถูกใช้ในการเข้ารหัสข้อมูลที่มีขนาดใหญ่เพราะรวดเร็วกว่าในขณะที่RSA จะถูกใช้ในระบบสื่อสารที่ไม่ยาวนักแต่ต้องการความปลอดภัยสูงในบางครั้ง RSA ยังถูกใช้ร่วมกับ DES เพื่อเสริมจุดเด่นซึ่งกันและกัน เช่นตัวเอกสารจริงจะถูกเข้ารหัสด้วย DES โดยที่คีย์รหัส DES จะถูกเข้ารหัสด้วย RSA แล้วส่งไปด้วยกันหรือส่งไปก่อนแต่ในบางครั้ง DES อย่างเดียวก็พอแล้วหากการแลกเปลี่ยนคีย์สามารถทำได้อย่างปลอดภัยเพียงพอ หรือในกรณีที่ผู้ส่งและผู้รับเป็นบุคคลเดียวกัน เช่น ฮาร์ดดิสก์ในคอมพิวเตอร์ส่วนตัวหรือข้อมูลส่วนตัวในสมาร์ตการ์ด

บทที่ 4
โพรโตคอล ICQ

โพรโตคอล ICQ คือรายละเอียดขั้นตอนการติดต่อสื่อสารของโปรแกรม ICQ ซึ่งขณะนี้ยังไม่ถูกกำหนดให้เป็นมาตรฐาน แต่กำลังจะถูกกำหนดให้เป็นมาตรฐานในภายหน้าต่อไปอีกไม่นานนี้ อันเนื่องมาจากมีผู้นิยมใช้เป็นจำนวนมาก เพราะรูปแบบ และการใช้งานโปรแกรม ICQ ที่ง่ายไม่ยุ่งยากเกินไปนัก การค้นหาข้อมูลโพรโตคอลในการทำโครงงานนี้ ค้นหาจากเว็บไซต์ http://www.student.nada.kth.se/~d95-mih/icq/ ซึ่งเป็นเว็บไซต์ที่รวบรวมข้อมูลโพรโตคอล ICQ และ link อื่นๆที่น่าสนใจ ข้อมูลที่ค้นหาได้มานั้นยังไม่ครบถ้วนสมบูรณ์ทำงานได้ครบทุกฟังก์ชันเต็มความสามารถของ ICQ แต่ก็เพียงพอสำหรับการส่งข้อความระหว่างผู้ใช้ ICQ ได้

โพรโตคอล ICQ ใช้โพรโตคอลพื้นฐานของเครือข่ายอินเทอร์เน็ตคือ UDP และ TCP ใช้โพรโตคอล UDP สำหรับการติดต่อสื่อสารระหว่าง ICQ Server กับ ICQ Client และ ใช้อินเทอร์เน็ตโพรโตคอลแบบ TCP สำหรับการติดต่อสื่อสารระหว่าง ICQ Client กับ ICQ Client โดยสื่อสารกับ Server ในช่วงแรกของการเริ่มโปรแกรม ICQ และต่อจากนั้นจะเป็นครั้งคราว ส่วนการติดต่อระหว่าง ICQ Client เป็นการสื่อสารแบบโดยตรง ยกเว้นมีปัญหาระหว่างการเชื่อมต่อระหว่าง ICQ Client จึงจะส่งผ่าน Server แทน

4.1 เวอร์ชันของโพรโตคอลที่มีใช้งาน

โพรโตคอล ของ ICQ ที่มีใช้มีอยู่หลายเวอร์ชัน เช่น

- เวอร์ชันที่ 1 มีความสามารถน้อย ยังไม่เป็นที่นิยม ปัจจุบัน ICQ Server ไม่สนับสนุนแล้ว
- เวอร์ชันที่ 2 ถูกพัฒนาให้มีความสามารถมากขึ้นจนเป็นที่นิยมใช้ ขณะนี้ ICQ Server ยังสนับสนุนอยู่
- เวอร์ชันที่ 3 และ 4 เพิ่มคำสั่งมากขึ้นจากโพรโตคอลเวอร์ชัน 2 และเพิ่มส่วนของการตรวจสอบความถูกต้องของข้อมูลที่รับส่ง ในเวอร์ชัน 4 สามารถ encrypt ส่วน header เข้าไปด้วย แต่ก็ใช้ได้ไม่นานนัก
- เวอร์ชันที่ 5 มีความเสถียรมากกว่าและมีคำสั่งมากขึ้น ปกติ ICQ Client ใช้โพรโตคอลเวอร์ชันนี้เป็นจำนวนมาก

อาจเข้าใจผิดระหว่างเวอร์ชันของโพรโตคอลและเวอร์ชันของโปรแกรม ICQ ซึ่งแตกต่างกัน

- การพัฒนาเวอร์ชันของโปรแกรม ICQ client เป็นการเพิ่มความสามารถและแก้ไขปัญหาโปรแกรมให้รองรับกับเวอร์ชันโพรโตคอล ICQ ใหม่ๆ เช่น ICQ v98a built 1700 ,ICQ 99a และ ICQ 99b เป็นต้น
- ส่วนการพัฒนาเวอร์ชันของโพรโตคอล ICQ เป็นการพัฒนาความสามารถในการสื่อสารระหว่างผู้ใช้ ICQ ไม่ว่าจะเป็นทางด้านการติดต่อด้าน ICQ

โครงการได้นำโพรโตคอลเวอร์ชัน 2 มาประยุกต์ใช้ เนื่องจากนำมาประยุกต์ง่าย

4.2 ลักษณะข้อมูลและขนาดข้อมูลที่ใช้ในการอ้างอิง

ข้อมูลที่อ้างถึงในโครงงานนี้มีขนาดต่างๆ กัน แต่ละขนาดมีชื่อเรียกเฉพาะเพื่อความเข้าใจตรงกัน จึงกำหนดให้ขนาดข้อมูลมีดังนี้

ชนิดข้อมูล	คำอธิบาย
BYTE	ข้อมูลขนาด 1 ไบต์ หรือ 8 บิต
WORD	ข้อมูลขนาด 2 ไบต์ หรือ 16 บิต
DWORD	ข้อมูลขนาด 4 ไบต์ หรือ 32 บิต
STRING	ทุกกลุ่มตัวอักษรมีรหัสจบกลุ่มตัวอักษรขนาด 1 ไบต์ มีค่าเท่ากับ 0
VARIABLE	กลุ่มข้อมูลไบต์ที่มีขนาดไม่คงที่ ซึ่งมักจะกำหนดขนาดข้อมูลมาล่วงหน้า เพื่อสามารถกำหนดขนาดของตัวแปรสำหรับรองรับข้อมูลกลุ่มนี้

ตารางที่ 4.1 แสดงรูปแบบข้อมูลที่อ้างอิง

ลักษณะของมูลที่รับและส่งเป็นแบบ little endian ordering การนำข้อมูลมาใช้ต้องนำมาสลับตำแหน่งไบต์เสียก่อนจึงจะนำมาใช้ได้อย่างปกติ มิฉะนั้นการคำนวณจะผิดพลาดอย่างมาก ดังตัวอย่าง

เลขฐาน 10 ชนิดต่างๆ	เลขฐาน 16	Little endian ordering
WORD w = 10	00 0a	0a 00
DWORD x = 123456	00 01 e2 40	40 e2 01 00

ตารางที่ 4.2 แสดงตัวอย่างข้อมูลแบบ little endian ordering

4.3 การติดต่อระหว่าง ไคลเอ็นต์ และ เซิร์ฟเวอร์ ด้วยอินเตอร์เน็ตโพรโตคอลแบบ UDP

โพรโตคอล ICQ มี 2 รูปแบบคือ

1. การสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์
2. การสื่อสารระหว่างไคลเอ็นต์กับไคลเอ็นต์

หัวข้อนี้กล่าวถึงการสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ส่วนการสื่อสารระหว่างไคลเอ็นต์กับไคลเอ็นต์ จะกล่าวถึงในหัวข้อถัดไป

เริ่มแรกของการทำงานของโปรแกรม ICQ หรือโพรโตคอล ICQ เริ่มจากผู้ใช้ ICQ หรือไคลเอ็นต์ติดต่อไปยังเซิร์ฟเวอร์เพื่อขอข้อมูลของผู้ใช้ ICQ อื่นๆที่มีอยู่ในรายการ(contact list) เช่น หมายเลข IP หมายเลขพอร์ต สถานะผู้ใช้ ICQ นั้นๆ เป็นต้น ข้อมูลเหล่านี้ได้มาจากการบริการทางด้านเซิร์ฟเวอร์

4.3.1 หน้าที่ของเซิร์ฟเวอร์

หน้าที่ของเซิร์ฟเวอร์คือการให้บริการข้อมูลแก่ผู้ใช้ ICQ ซึ่งมีดังต่อไปนี้

1. ให้ข้อมูลสำหรับการสื่อสารของผู้ใช้ ICQ อื่นตามที่ไคลเอ็นต์ต้องการ เช่น
   - หมายเลข IP
   - หมายเลขพอร์ต

- ชื่อจริง ประกอบด้วย first name และ last name
- ชื่อเล่น (nick name)
- สถานะของของผู้ใช้ ICQ เช่น Online ,Offline ,NA , Away DND เป็นต้น

2 ให้บริการสำหรับการค้นหาผู้ใช้ ICQ ที่กำหนด ซึ่งสามารถค้นหาได้โดยกำหนดข้อมูลส่วนใดส่วนหนึ่งดังต่อไปนี้

- ให้บริการการค้นหาโดยค้นหาจากหมายเลขผู้ใช้ ICQ
- ให้บริการการค้นหาโดยค้นหาจากชื่อเล่นหรือ nick name
- ให้บริการการค้นหาโดยค้นหาจากชื่อจริง firstname หรือ last name
- ให้บริการการค้นหาโดยค้นหาจาก email
- สำหรับโพรโตคอลเวอร์ชันใหม่สามารถค้นหาได้จากข้อมูลส่วนอื่นๆได้ เช่น ภาษาสื่อสาร ช่วงอายุ สิ่งที่สนใจ เป็นต้น

3. ให้บริการในการสื่อสารข้อความผ่านทางเซิร์ฟเวอร์ หรือเรียกว่าบริการ offline message เป็นบริการสำหรับส่งข้อความไปยัง ICQ อื่นที่ไม่อยู่ในขณะนั้น ซึ่งอยู่ในสถานะ offline เมื่อใดที่ผู้ใช้ ICQ นั้นใช้งานโปรแกรม ICQ ทางเซิร์ฟเวอร์ก็จะให้บริการส่ง offline message ไปให้ทันที หรือ ถ้าการสื่อสารการส่งข้อความระหว่างผู้ใช้ ICQ มีปัญหาก็จะทำการส่งข้อความผ่านเซิร์ฟเวอร์แทนการส่งโดยตรง
4. ให้บริการการเปลี่ยนแปลงสถานะต่างๆของผู้ใช้ ICQ เมื่อผู้ใช้ ICQ เปลี่ยนแปลงสถานะไคลเอ็นต์จะส่งคำสั่งไปเปลี่ยนสถานะให้กับเซิร์ฟเวอร์รับรู้ และจากนั้นเซิร์ฟเวอร์จะส่งคำสั่งเปลี่ยนสถานะของผู้ใช้ ICQ ไปยังผู้ใช้อื่นๆที่อยู่ในรายการ contact list
5. ให้บริการลงทะเบียนหมายเลข ICQ ใหม่ หมายเลข ICQ เรียกอีกอย่างว่าหมายเลข UIN สำหรับผู้ใช้ใหม่หรือต้องการใช้แต่ละหมายเลข ICQ สำหรับแต่ละงาน
6. ให้บริการยกเลิกการจดทะเบียนผู้ใช้ ICQ ผู้ใช้ ICQ สามารถยกเลิกการลงทะเบียนหมายเลข ICQ ที่ไม่ต้องการใช้ได้ต่อไป

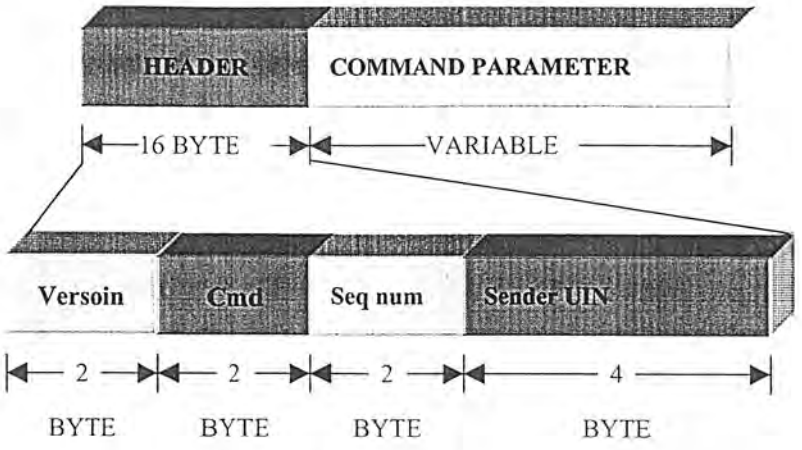
4.3.2 หมายเลข IP และ หมายเลขพอร์ตของเซิร์ฟเวอร์ ICQ

ชื่อเซิร์ฟเวอร์ของ ICQ	หมายเลขพอร์ต
Icq.mirabilis.com	4000

ตารางที่ 4.3 แสดงหมายเลข IP และ พอร์ตของเซิร์ฟเวอร์ ICQ

4.3.3 ลักษณะข้อมูลที่ใช้สื่อสารระหว่าง ICQ Client และ ICQ Server โดยผ่าน UDP

ชุดข้อมูลหรือแพ็กเก็ตโพรโตคอลเวอร์ชัน 2 ที่ใช้อินเตอร์เน็ตโพรโตคอลแบบ UDP จากภาพจะเห็นว่า แพ็กเก็ตแบ่งออกเป็น 2 ส่วน คือ ส่วน header และ ส่วน command parameter



รูปที่ 4.1 แสดงแพ็กเก็ตที่ใช้ในการสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

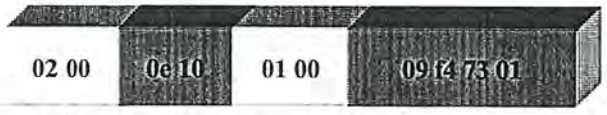
1 **Header** สามารถแยกออกได้อีก 4 ส่วนคือ
- *Version* หมายเลขเวอร์ชันของโพรโตคอล เช่น เวอร์ชัน 2
- *Cmd* หมายเลขคำสั่งที่ใช้สื่อสารระหว่างไคลเอนต์ และเซิร์ฟเวอร์ ซึ่ง ขนาดของ Command parameter ขึ้นอยู่กับคำสั่งนี้ ตัวอย่างคำสั่งเช่น คำสั่ง LOGIN มีเลขรหัส 1000
- *Seqnum* หมายเลขลำดับของคำสั่งที่รับและส่งระหว่างไคลเอ็นต์ และ เซิร์ฟเวอร์ ซึ่งจะเพิ่มขึ้นตลอดเวลาที่ส่งคำสั่งออกไป
- *Sender UIN* หมายเลข ICQ ผู้ส่งแพ็กเก็ต เช่น 24376329

2 **Command parameter** ขนาดขึ้นอยู่กับคำสั่งที่ใช้ในการสื่อสารดังตัวอย่าง ไคลเอ็นต์ส่งข้อความไปยังเซิร์ฟเวอร์มีคำสั่งดังนี้
- คำสั่ง SEND_MESSAGE มีรหัสคำสั่ง 270 (0e 10)
- หมายเลข ICQ ผู้ส่ง 24376329 (09 f4 73 01)
- หมายเลข ICQ ผู้รับ 60710412 (0c 5e 9e 03)

ขนาด	ชื่อพารามิเตอร์	คำอธิบาย
DWORD	RECEIVER_UIN	หมายเลข ICQ ผู้รับ
WORD	MESSAGE_TYPE	ข้อความชนิดปกติมีรหัสเท่ากับ 1
WORD	LENGTH	ขนาดความยาวของข้อความ
VARIABLE	MESSAGE	ข้อความที่ส่งไป เช่น “IsagQ”

ตารางที่ 4.4 แสดง command parameter ของคำสั่งส่งข้อความ

ส่วน HEADER



รูปที่ 4.2 แสดง ส่วนหัวแพ็กเก็ตคำสั่งส่งข้อความ

รูปที่ 4.3 แสดงส่วน command parameter ของคำสั่งส่งข้อความ

4.3.4 การสื่อสารกับเซิร์ฟเวอร์

รูปแบบการติดต่อกับเซิร์ฟเวอร์ เมื่อสั่งคำสั่งใดไปให้ทางเซิร์ฟเวอร์จะตอบกลับมาด้วย คำสั่ง ACK และในทางกลับกัน เมื่อเซิร์ฟเวอร์ส่งคำสั่งใดมาให้ต้องตอบกลับคำสั่งนั้นด้วยคำสั่ง ACK ยกเว้นคำสั่ง ACK ที่เซิร์ฟเวอร์ส่งมา sequence number ทั้งส่งและรับต้องเท่ากัน มิเช่นนั้นจะติดต่อกับเซิร์ฟเวอร์ไม่ได้อีก จนกว่าจะส่ง seqence number ที่ถูกต้อง

รูปที่ 4.4 แสดงการสื่อสารแต่ละคำสั่ง

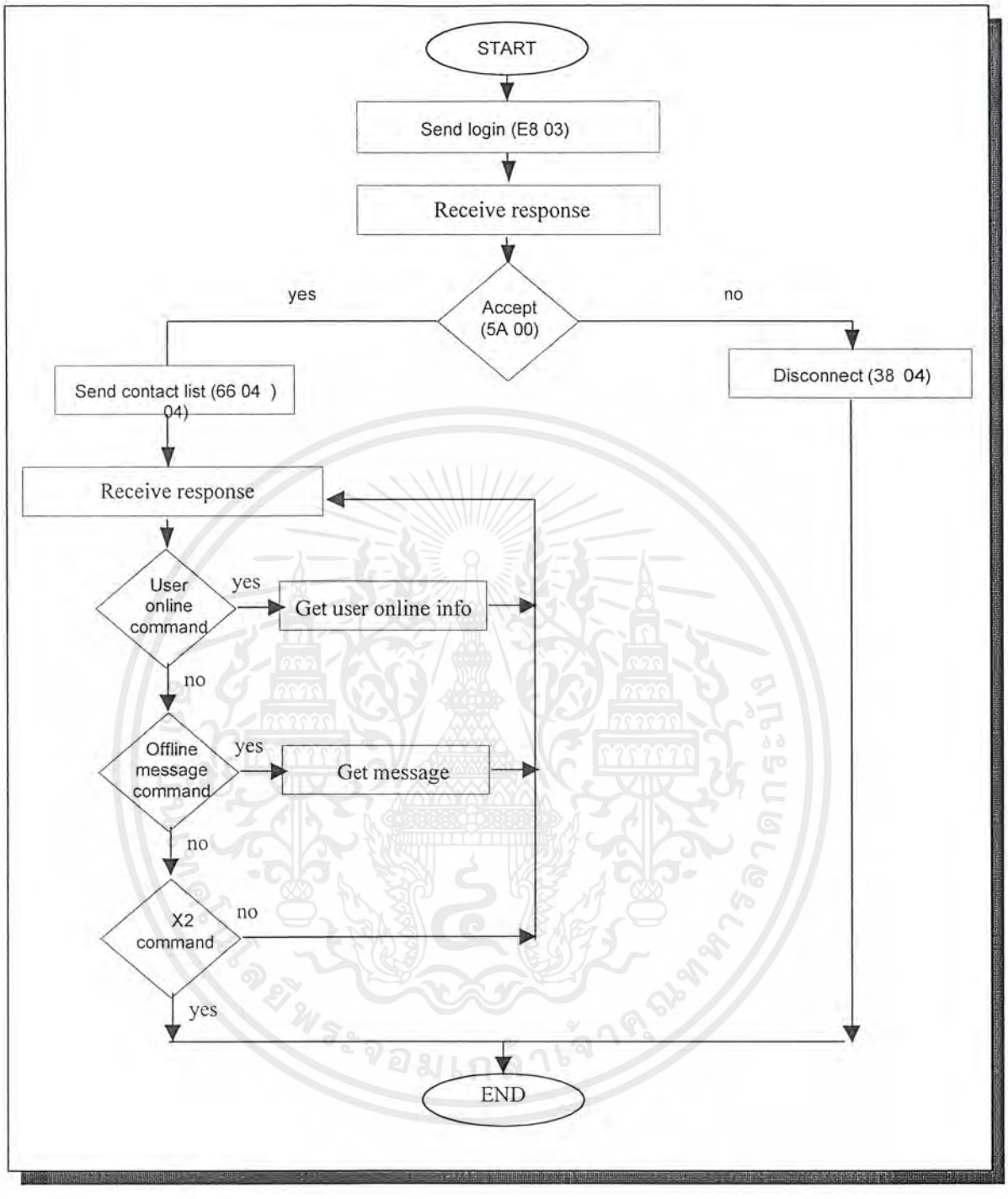
การติดต่อกับเซิร์ฟเวอร์มีหลายชนิด จะกล่าวถึงทั้งหมดคงไม่ได้เพราะบางอย่างก็ไม่ยากเกินไปนักที่จะเข้าใจ จึงเน้นเฉพาะคำสั่งที่ค่อนข้างยาก เช่น

- การ login
- การค้นหาผู้ใช้ ICQ

4.3.3.1 ขั้นตอนการตรวจสอบสิทธิผู้ใช้ (Login)

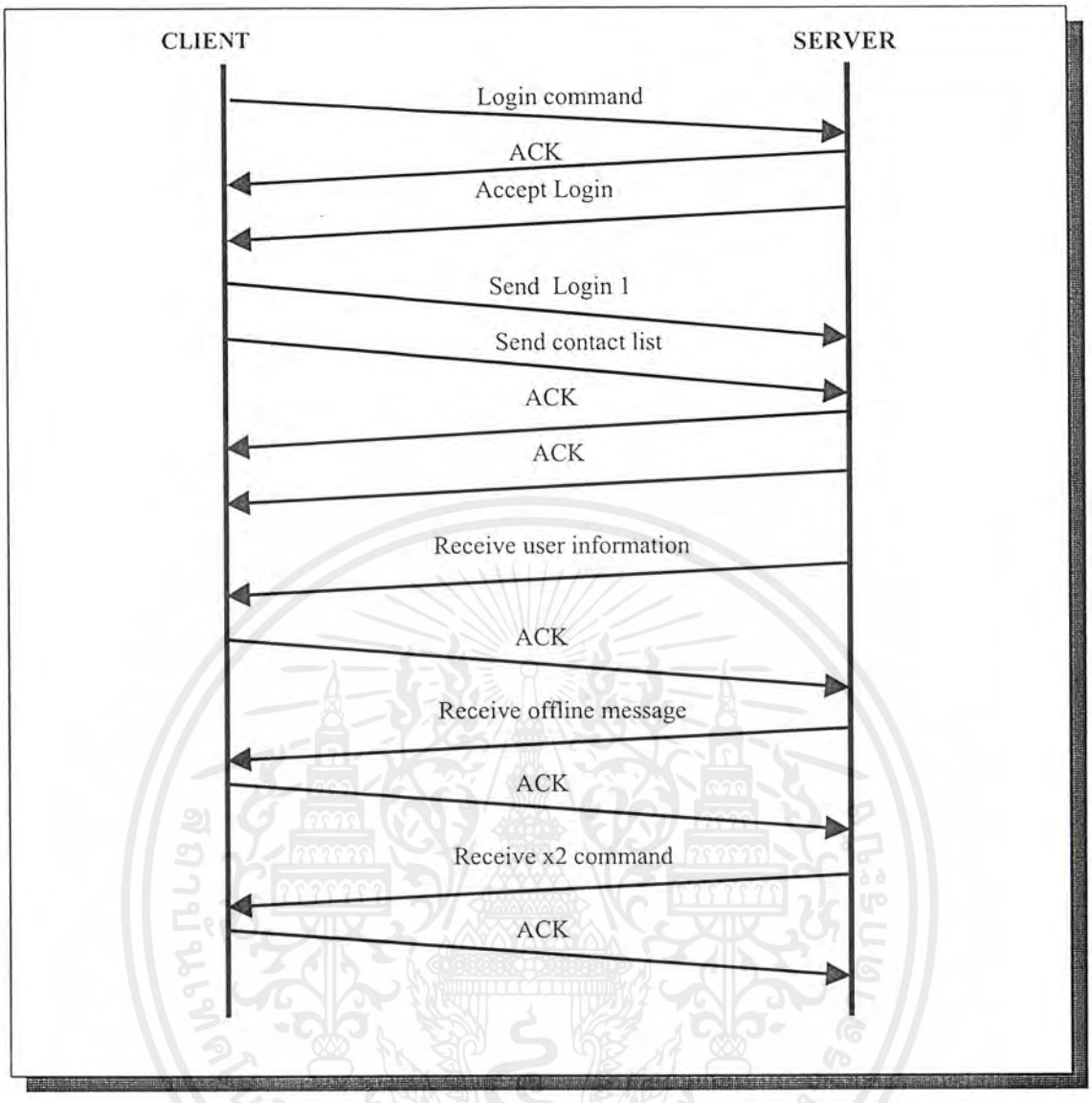
ขั้นตอนแรกในการติดต่อกับเซิร์ฟเวอร์ต้องส่งคำสั่ง LOGIN ให้กับเซิร์ฟเวอร์ก่อน เพื่อทำการตรวจสิทธิ ก่อนจึงจะสื่อสารกันต่อไป หากไม่ผ่านขั้นตอนนี้ไม่ว่าจะส่งคำสั่งใดๆเซอร์ฟเวอร์จะไม่ตอบกลับมาเลย

ขั้นตอนต่างๆทำงานตาม Flowchart ที่แสดงไว้ในรูปที่ 4.5 เมื่อทำงานจนจบขั้นตอนใน flowchart แล้ว เรียกว่าขั้นตอนการ login เสร็จสิ้น ต่อจากนี้สามารถส่งคำสั่งใดๆก็ได้ที่ต้องการ การส่งแต่ละครั้งต้องเพิ่มค่า sequence number ครั้งละ 1 ยกเว้นคำสั่ง ACK ซึ่งค่า sequence number ต้องเหมือนกับคำสั่ง ที่ต้องกันตอบรับกลับไป

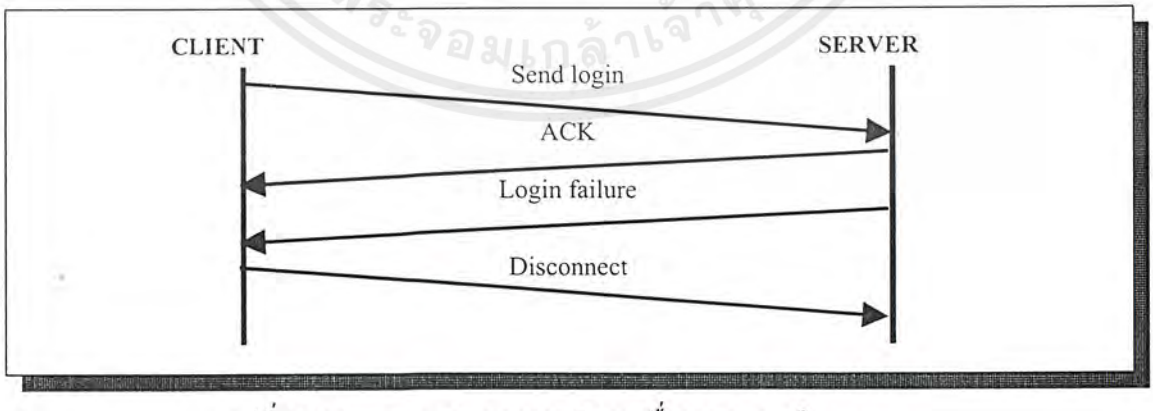


รูปที่ 4.5 แสดง Flowchart ของขั้นตอนการ login

รูปที่ 4.6 แสดงขั้นตอนการส่งคำสั่งในแต่ละคำสั่งของขั้นตอนการ login ซึ่งจะเป็นไปตามขั้นตอนค่อนข้างจะตายตัว ส่วนรูปที่ 4.7 แสดงขั้นตอนการ login ที่ผิดพลาด



รูปที่ 4.6 Flow control แสดงขั้นตอน login



รูปที่ 4.7 Flow control แสดงการทำงานเมื่อการ login เกิด error

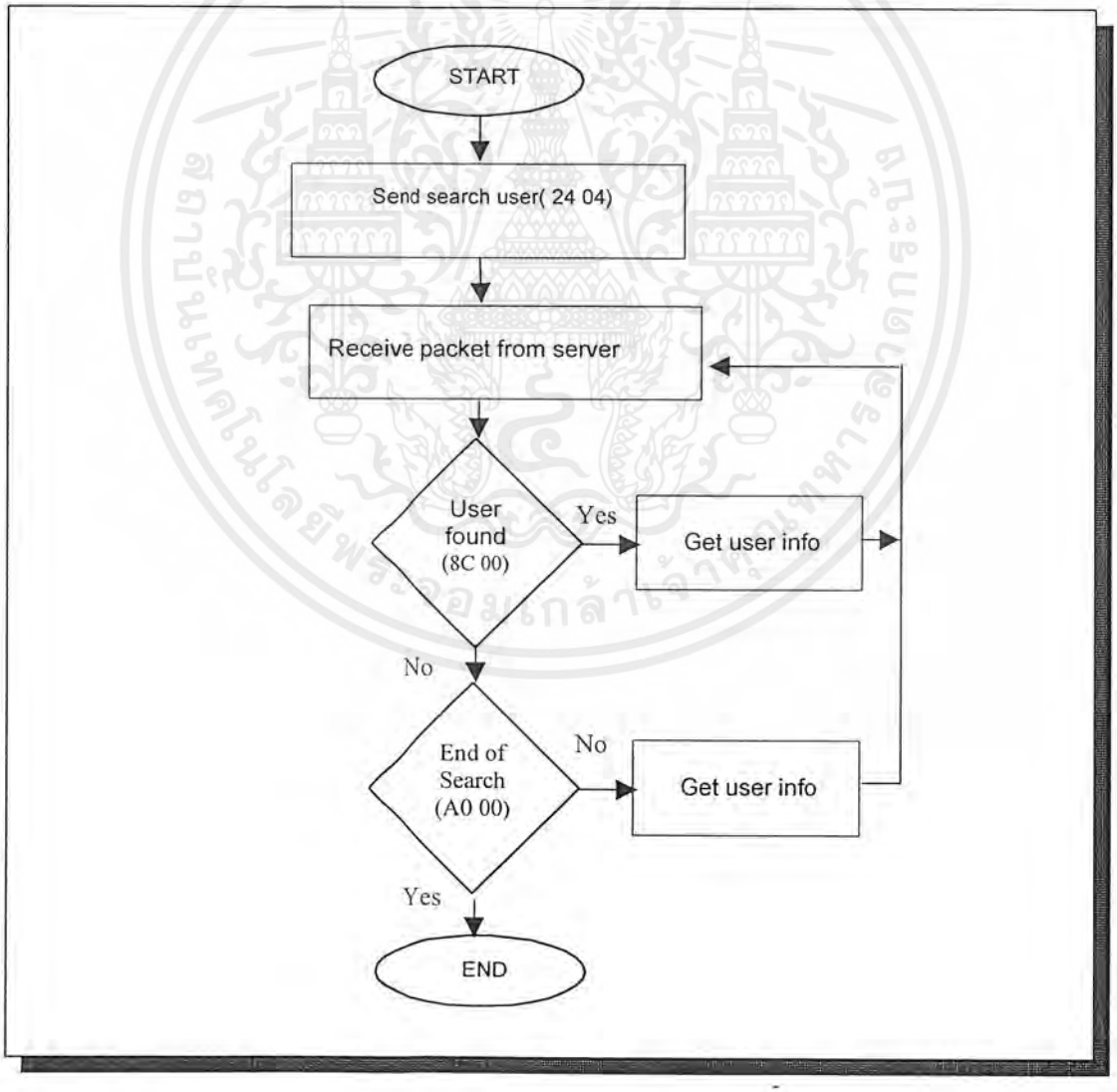
4.3.4.2 การค้นหาผู้ใช้ ICQ

หากเราทราบหมายเลข ICQ ชื่อเล่นที่ใช้ ชื่อจริง หรืออีเมล์ อย่างใดอย่างหนึ่งของผู้ใช้อื่นที่ต้องการเพิ่มในรายการ contact list ซึ่งจะต้องค้นหาจากเซิร์ฟเวอร์ก่อนแล้วจึงเพิ่มในรายการได้ซึ่งมีขั้นตอนดังต่อไปนี้

คำสั่งค้นหาข้อมูลมี คำสั่งคือ คำสั่ง Search UIN และ Search User ขั้นตอนของคำสั่งทั้งสองเหมือนกันดังรูป

- Search UIN เป็นคำสั่งค้นหาข้อมูลผู้ใช้โดยใช้หมายเลข ICQ
- Search User เป็นคำสั่งค้นหาข้อมูลผู้ใช้โดยใช้ชื่อผู้ใช้ หรือ email

เมื่อส่งคำสั่งคนหาไปแล้ว ให้รอรับข้อมูลจากคำสั่งตอบ User Found กลับมาแสดงว่าพบผู้ใช้แล้ว นำเอาแพ็กเก็ตที่รับได้ไปแยกข้อมูลออกเป็นส่วนๆ ซึ่งจะส่งมาเรื่อยๆจนกว่าจะด้านเซิร์ฟเวอร์จะค้นหาไม่เจออีกแล้ว ต่อจากนั้นจะส่งคำสั่ง End of Search ตอบกลับมา แต่ละคำสั่งมีข้อมูลอะไรบ้างนั้น ให้ดูที่ภาคผนวก



รูปที่ 4.8 การค้นหาผู้ใช้งาน

4.3.3.2 การยกเลิกการติดต่อกับเซิร์ฟเวอร์

ใช้คำสั่ง SEND_TEXT_CODE มีรหัสคำสั่ง (38 04) ส่งข้อความ "B_USER_DISCONNECTED "ให้กับเซิร์ฟเวอร์ เราก็จะเข้าสู่สถานะ offline ทันทีและจากนั้นเซิร์ฟเวอร์ทำการยกเลิกการติดต่อ

4.4 การติดต่อสื่อสารระหว่างไคลเอนต์กับไคลเอนต์โดยอินเตอร์เน็ตโพรโตคอลแบบ TCP

การสื่อสารระหว่างไคลเอนต์และไคลเอนต์เรียกอีกอย่างหนึ่งว่าแบบ direct connect ซึ่งไคลเอนต์จะเชื่อมต่อโดยตรงโดยใช้อินเตอร์เน็ตโพรโตคอลแบบ TCP การเชื่อมต่อโดยใช้ TCP นี้สามารถรับประกันความครบถ้วนของข้อมูลส่งถึงผู้ใช้อีกฝั่งอย่างแน่นอน

รูปแบบข้อมูลสำหรับส่งระหว่างไคล์เอ็นต์ต่างจากข้อมูลสำหรับสื่อสารกับเซิร์ฟเวอร์ รูปแบบข้อมูลสำหรับสื่อสารระหว่างไคลเอนต์มีอยู่ด้วยกัน 2 ชุด ซึ่งเรียกว่า CHANNEL_INIT และ CHANNEL_MESSAGE

- **CHANNEL_INIT** ไคลเอนต์ส่งชุดข้อมูลชุดนี้ก่อนที่จะส่งข้อความแรกให้กับแต่ละไคลเอนต์ ประกอบด้วยข้อมูลต่างๆดังตารางที่ 4.5
- **CHANNEL_MESSAGE** ชุดข้อมูลที่บรรจุข้อความ ข้อมูลต่างๆแสดงในตารางที่ 4.6

CHANNEL_INIT		
ขนาด	**ชื่อข้อมูล**	**คำอธิบาย**
BYTE	Init_ident	ไม่มีข้อมูลอธิบาย ปกติมีค่าเท่ากับ 0xff;
WORD	Version	เวอร์ชั่นของ ICQ
WORD	Revision	ไม่มีข้อมูลอธิบาย ปกติมีค่าเป็น 0
DWORD	Msg_port	พอร์ตสำหรับรอรับข้อความ
DWORD	Uin	หมายเลข ICQ ผู้ส่งข้อความ
DWORD	IP_real	IP address จริง
DWORD	IP	IP address ถ้าหากอยู่หลังไฟร์วอลล์ เป็น IP address ของไฟร์วอลล์
BYTE	Tcp_flag	ค่าแฟล็กเกี่ยวกับ TCP ปกติมีค่าเท่ากับ 4
WORD	Port	หมายเลขพอร์ตสำหรับรอรับ chat และ ไฟล์
WORD	Ext_port	ค่าเพิ่มเติมเกี่ยวกับพอร์ต ปกติมีค่าเท่ากับ 0

ตารางที่ 4.5 แสดงข้อมูลภายใน CHANNEL_INIT

CHANNEL_MESSAGE		
ขนาด	ชื่อข้อมูล	คำอธิบาย
DWORD	UIN	หมายเลข ICQ ผู้ส่งข้อความ
WORD	Version	เวอร์ชันโพรโตคอล TCP ของ ICQ
WORD	Command	คำสั่งกำหนดชนิดข้อความ ดูตารางที่ 4.7
WORD	X1	ไม่รู้ความหมาย ปกติมีค่าเท่ากับ 0
DWORD	UIN	หมายเลข ICQ ผู้ส่งข้อความ
WORD	Type	กำหนดชนิดข้อความ ดูตารางที่ 4.8
WORD	MsgLen	ขนาดของข้อความ (จำนวนตัวอักษร)
VARIABLE	Message	ข้อความซึ่งมีความยาวเท่ากับ MsgLen
DWORD	TCP_MsgIP	IP address ถ้าหากอยู่หลังไฟร์วอลล์ เป็น IP address ของไฟร์วอลล์
DWORD	TCP_RealIP	IP address จริงของ client
DWORD	TCP_MsgPort	หมายเลขพอร์ตสำหรับรอรับ ข้อความ
BYTE	TCP_Flag	ค่าแฟล็กเกี่ยวกับ TCP ปกติมีค่าเท่ากับ 4
WORD	TCP_Status	สถานะของไคลเอนต์ ดูตารางที่ 4.10
WORD	MsgCommand	คำสั่งเกี่ยวกับการรับส่งข้อความ ดูตารางที่ 4.9

ตารางที่ 4.6 แสดงข้อมูลภายใน CHANNEL_MESSAGE

TCP Commands		
ตัวแปร	ค่าตัวแปร	คำอธิบาย
TCP_CANCEL	0x07D0 (2000)	ยกเลิกการติดต่อ
TCP_ACK	0x07DA (2010)	ตอบกลับ
TCP_MESSAGE	0x07EE (2020)	ส่งข้อความ

ตารางที่ 4.7 แสดงชนิดคำสั่งสำหรับส่งข้อความแบบ TCP

TCP Message Types		
ตัวแปร	ค่าตัวแปร	คำอธิบาย
MSG_MSG	0x0001	ส่งข้อความปกติ
MSG_CHAT	0x0002	เริ่มต้นสื่อสารแบบ Chat
MSG_FILE	0x0003	เริ่มต้นส่งไฟล์
MSG_URL	0x0004	ส่ง URL address
MSG_REQ_AUTH	0x0006	ร้องขอสิทธิในการรับรู้สถานะของผู้อื่น
MSG_DENY_AUTH	0x0007	ไม่ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น
MSG_GIVE_AUTH	0x0008	ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น

MSG_ADDED	0x000C	สัญญาณอัตโนมัติแสดงว่าตนเองถูกผู้อื่นเพิ่มใน รายการ contact list
MSG_READAWAY	0x03E8	ได้รับข้อความจากสถานะ Away
MSG_READOCCUPIE	0x03E9	ได้รับข้อความจากสถานะ Occupie
D MSG_READNA	0x03EA	ได้รับข้อความจากสถานะ Not Available
MSG_READDND	0x03EB	ได้รับข้อความจากสถานะ Do Not Disturb
MSG_READFFC	0x03EC	ได้รับข้อความจากสถานะ Free For Chat
MSGF_MASS	0x0800	ส่งข้อความมากกว่าหนึ่งคนในครั้งเดียว

ตารางที่ 4.8 แสดงตัวแปรและค่าตัวแปรสำหรับกำหนดชนิดข้อความ

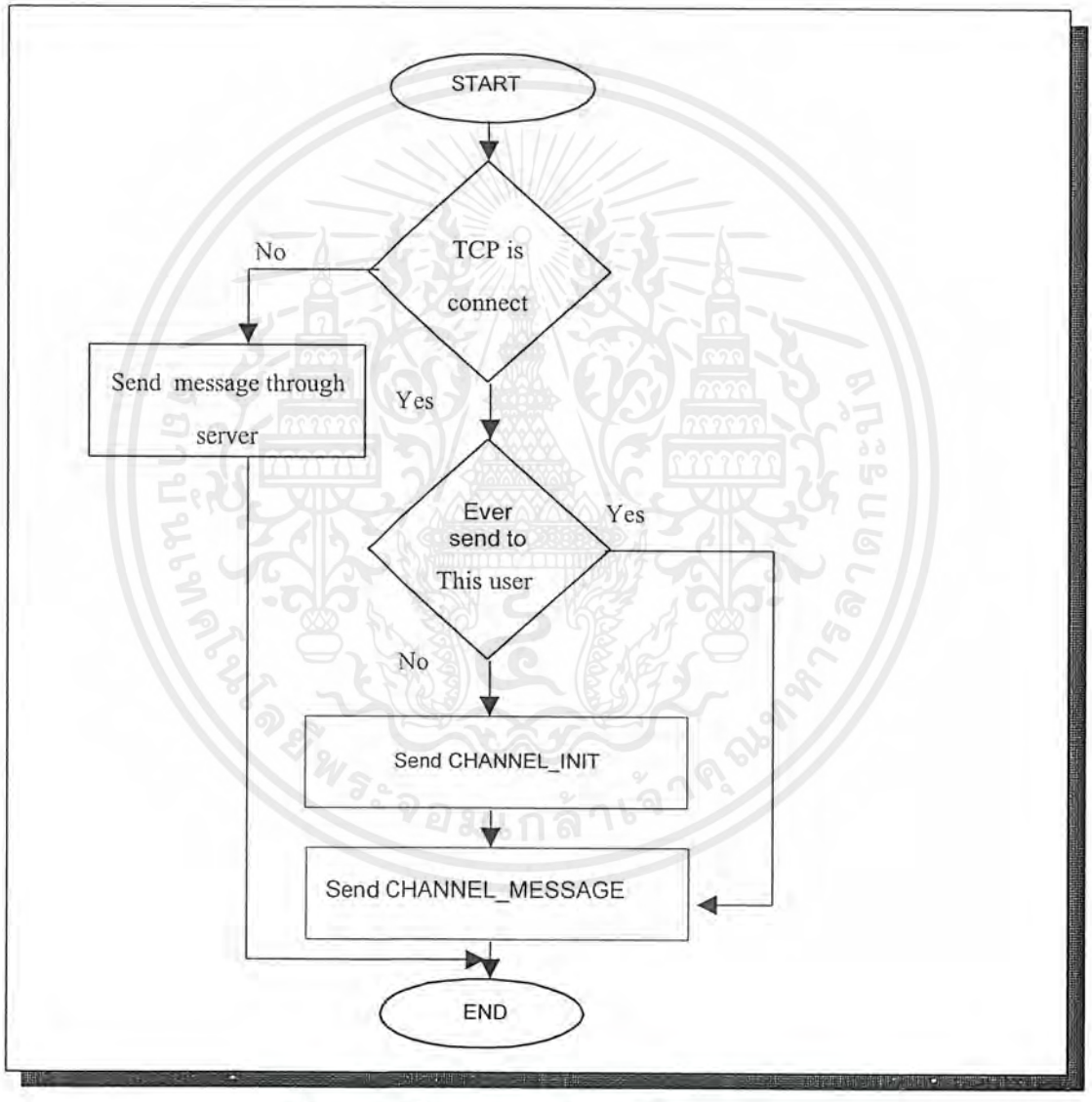
TCP Message Command Types		
ตัวแปร	ค่าตัวแปร	คำอธิบาย
TCP_MSG_AUTO	0x0000	กำหนดส่งข้อความกลับแบบอัตโนมัติ
TCP_MSG_REAL	0x0010	ส่งข้อความปกติ
TCP_MSG_LIST	0x0020	ส่งรายการ contact list
TCP_MSG_URGENT	0x0040	ส่งข้อความแบบ urgently
TCP_MSGF_S_INVISIBLE	0x0080	ผู้ส่งข้อความอยู่ในสถานะ Invisible
TCP_MSGF_S_AWAY	0x0100	ผู้ส่งข้อความอยู่ในสถานะ Away
TCP_MSGF_S_OCCUPIED	0x0200	ผู้ส่งข้อความอยู่ในสถานะ Occupied
TCP_MSGF_S_NA	0x0800	ผู้ส่งข้อความอยู่ในสถานะ N/A
TCP_MSGF_S_DND	0x1000	ผู้ส่งข้อความอยู่ในสถานะ DND

ตารางที่ 4.9 แสดงค่าที่กำหนดคำสั่งของชนิดข้อมูลต่างๆ

TCP Message Status	
ตัวแปร	ค่าตัวแปร
TCP_STAT_ONLINE	0x0000
TCP_STAT_REFUSE	0x0001
TCP_STAT_AWAY	0x0004
TCP_STAT_OCCUPIED	0x0009
TCP_STAT_DND	0x000A
TCP_STAT_NA	0x000E

ตารางที่ 4.10 แสดงสถานะของข้อความ

ขั้นตอนการส่งข้อความผ่านทาง TCP ก่อนอื่นทำการตรวจสอบสถานะการเชื่อมต่อ TCP กับผู้ใช้ปลายทาง ถ้าเชื่อมต่อสำเร็จแล้ว ขั้นต่อไปตรวจสอบเคยส่งให้ผู้ใช้ปลายทางนี้แล้วหรือไม่ ถ้ายังไม่เคยส่งข้อความสื่อสารกันเลย เริ่มส่งชุดข้อมูล CHANNEL_INIT ก่อนแล้วส่ง CHANNEL_MESSAGE ตามกันไป ถ้าส่งชุดข้อมูล CHANNEL_MESSAGE ก่อนเป็นผลทำให้ผู้รับปลายทางไม่สามารถรับข้อความได้ และในอีกกรณีหนึ่ง เมื่อตรวจสอบแล้วพบว่าเคยส่งข้อความให้กับผู้ใช้ปลายทางคนนี้แล้ว ขั้นต่อไปก็ส่งชุดข้อมูล CHANNEL_MESSAGE



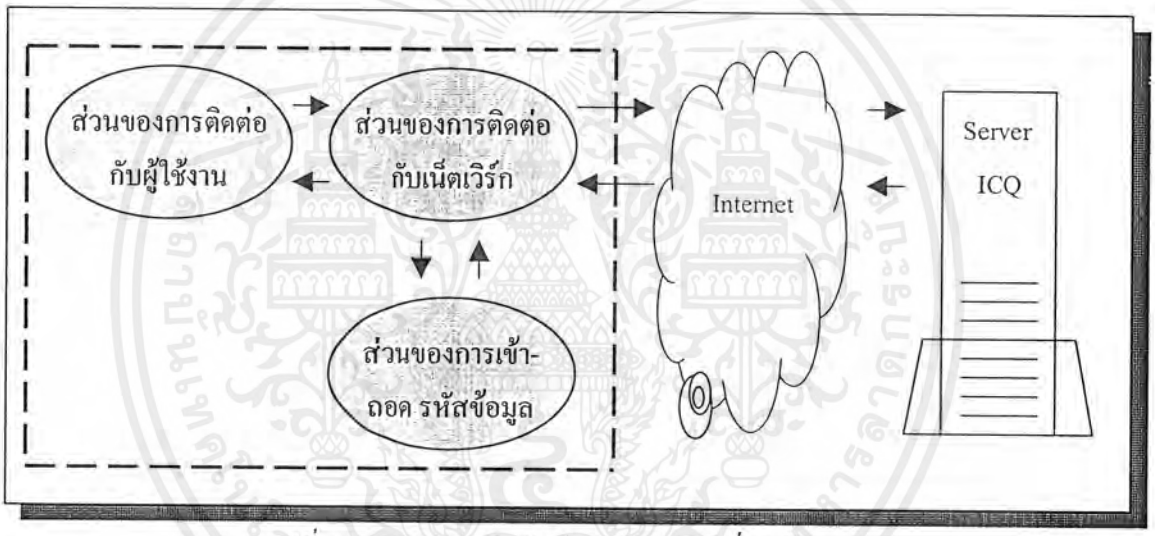
รูปที่ 4.9 แสดงขั้นตอนการส่งข้อความผ่านทาง TCP

บทที่ 5
การพัฒนาโครงงาน

5.1 ส่วนประกอบของโปรแกรม

โครงงานนี้ได้จัดแบ่งการทำงานออกเป็น 3 ส่วนหลักๆ ด้วยกันคือ

1. ส่วนของการติดต่อกับผู้ใช้งาน
2. ส่วนของการติดต่อเน็ตเวิร์ก
3. ส่วนของการเข้า-ถอดรหัสข้อมูล



รูปที่ 5.1 แสดงความสัมพันธ์ของส่วนต่างๆ ที่ออกแบบ

5.1.1. ส่วนของการติดต่อกับผู้ใช้งาน

ส่วนติดต่อกับผู้ใช้ได้ทำการออกแบบโครงสร้างไว้เช่นเดียวกับโปรแกรม ICQ ของเดิมที่เราใช้เป็นต้นแบบ เพื่อให้ผู้งานสามารถใช้งานได้อย่างคุ้นเคยกับ ICQ เดิมแต่ได้ทำการลดฟังก์ชันการทำงานบางอย่างที่ไม่ค่อยได้ใช้งานบางอย่างออก และได้ทำการเพิ่มส่วนของการแสดงสถานะการให้งานที่ปลอดภัยก็คือมีการบอกว่า ผู้ใช้งานคนใดใช้โปรแกรม ICQ ของเดิม และผู้ใดใช้งานโปรแกรม IsagQ ที่เราได้ทำการสร้างขึ้นเพราะการแสดงสถานะการออนไลน์จะต่างกัน โดยจะเพิ่มในส่วนของการแสดงสถานะของ Secure เพื่อบอกกับผู้ใช้ว่าคนใดใช้โปรแกรม IsagQ ของเราซึ่งระหว่างการติดต่อกันนั้นจะใช้การเข้ารหัสข้อมูลแบบ DES เพื่อความปลอดภัยของผู้ใช้เอง

5.1.2. ส่วนของการติดต่อเน็ตเวิร์ก

การติดต่อเน็ตเวิร์กจะใช้โพรโตคอลของ ICQ ซึ่งทำงานบน TCP และบน UDP ซึ่งการเชื่อมต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์จะใช้ UDP ส่วนการติดต่อระหว่างไคลเอนจะใช้ทั้ง TCP ทั้ง UDP ส่วนการติดต่อกับเซิร์ฟเวอร์ของ Mirabilis จะใช้การติดต่อไปที่ icq.mirabilis.com :port 4000 ซึ่งเป็นเซิร์ฟเวอร์ในการขอใช้บริการ

5.1.3. ส่วนของการเข้า-ถอดรหัสข้อมูล

เพื่อความปลอดภัยของข้อมูลระหว่างการใช้งานจึงจำเป็นต้องใช้การเข้ารหัสและถอดรหัสมาช่วย ซึ่งในที่นี้ได้ทำการใช้การเข้ารหัสแบบคีย์เดี่ยวและคีย์คู่ ในการทำงานโดยมีขั้นตอนการทำงานดังนี้คือ ใช้คีย์คู่ในการเข้า-ถอดรหัสคีย์เดี่ยว หลังจากนั้นจะนำคีย์เดี่ยวที่ได้ไปใช้ในการเข้า-ถอดรหัสข้อความที่จะทำการส่ง-รับอีกขั้นตอนหนึ่ง ซึ่งผู้ใช้ที่ไม่ได้ใช้โปรแกรม IsagQ จะไม่มี

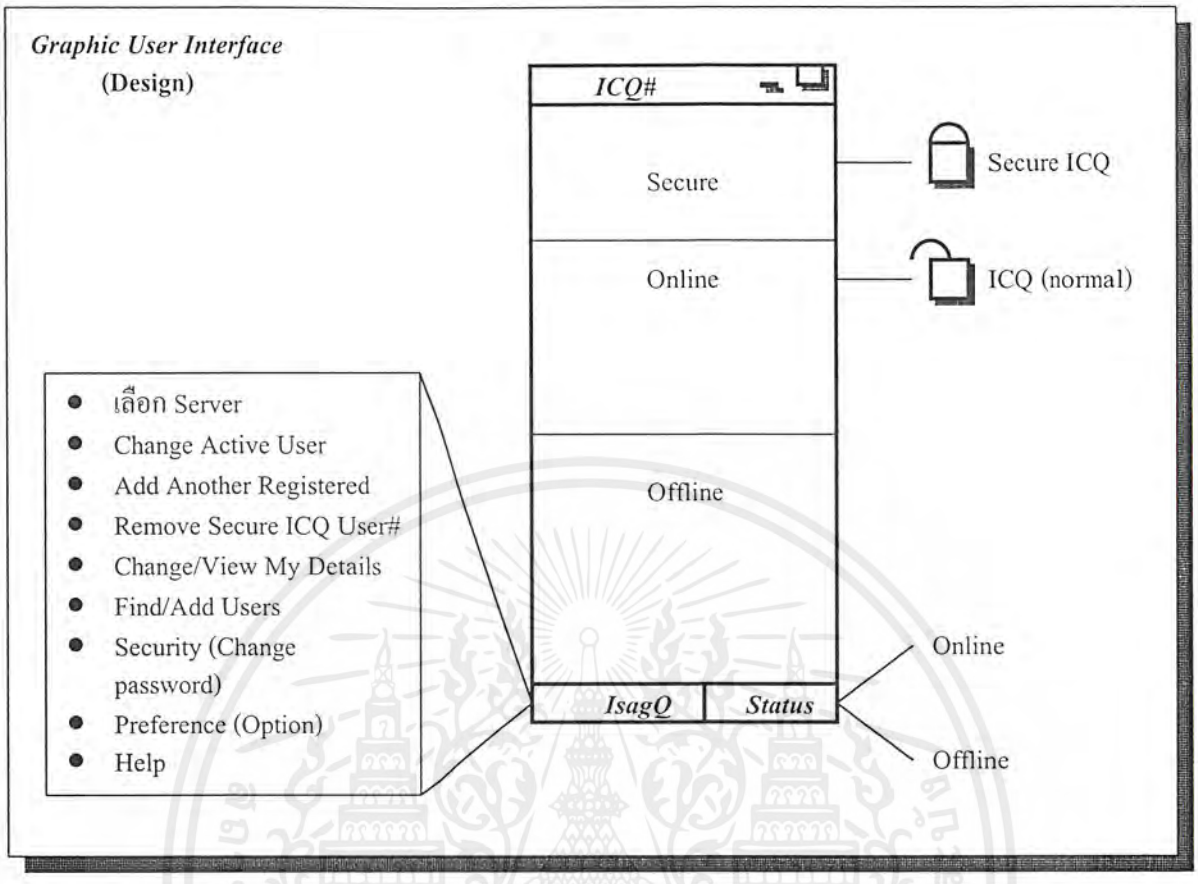
5.2 การ Design GUI และการสร้าง

ในส่วนของการออกแบบรูปที่ 5.2 ได้ทำการออกแบบโครงสร้างของหน้าจอสำหรับการติอต่อกับผู้ใช้ให้มีลักษณะคล้ายกับของ ICQ ที่ใช้งานอยู่ซึ่งทำให้ใช้งานได้ใกล้เคียงกับของเดิม ซึ่งประกอบด้วยส่วนหลักๆ ดังนี้คือส่วนแสดงสถานะของผู้ใช้ ส่วนแสดงหมายเลขผู้ใช้ รวมถึงเมนูต่างๆ ที่สามารถทำงานได้เมื่อมีการกดที่ปุ่ม ส่วนที่เพิ่มขึ้นมาอย่างเด่นชัดก็คือ ส่วนแสดงการใช้งานของผู้ใช้ที่มีการรักษาความปลอดภัยโดยการใช้โปรแกรม IsagQ ซึ่งเมื่อมีการติดต่อกันจะต้องแสดงอีกระดับหนึ่งของการออนไลน์ซึ่งจะแสดงเหนือขึ้นไปจากการออนไลน์แบบธรรมดา

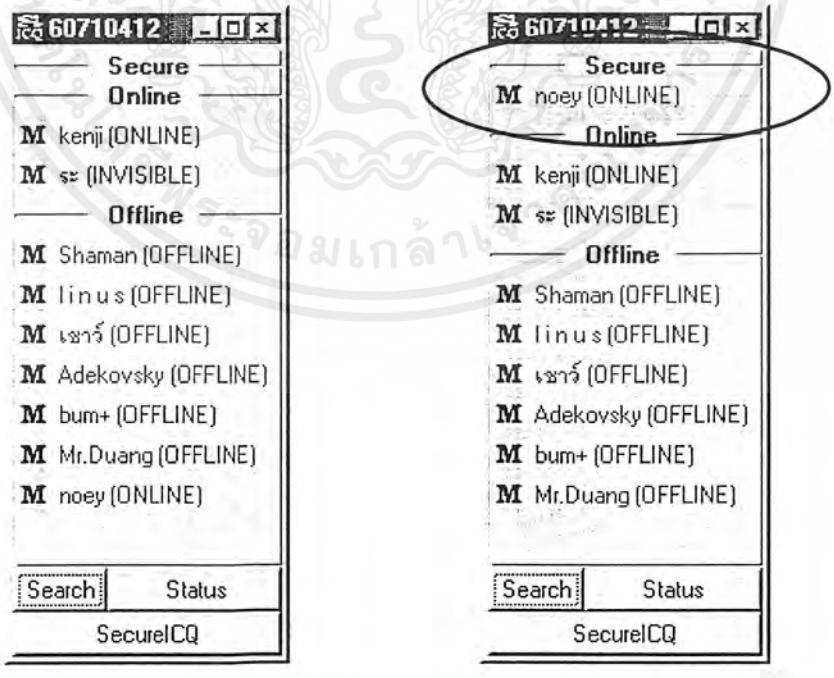
เมื่อได้ทำการออกแบบโครงสร้างส่วนของโปรแกรมที่ใช้ติดต่อกับผู้ใช้แล้ว เราได้ทำการสร้างขึ้นจากภาษาจาวาซึ่งการสร้างโปรแกรมจะใช้ class library ของจาวาที่ให้มาคือ java.awt.*; เป็นส่วนมาก ซึ่งทำให้มีลักษณะคล้ายกับโปรแกรม ICQ ของเดิมอยู่มากเพื่อสะดวกของผู้ใช้งาน รูปที่ 5.3 แสดงหน้าจอการติดต่อกับผู้ใช้ที่ได้สร้างขึ้นจากภาษาจาวา แต่จากรูปแรกจะเห็นว่ายังไม่มีการเข้าใช้งานของผู้ใช้ที่มีความปลอดภัย ซึ่งในส่วนนี้ก็จะยังไม่มีการแสดงชื่อใดๆ นอกจากการออนไลน์แบบธรรมดาและการออฟไลน์ ส่วนรูปต่อมาจะมีการแสดงสถานะที่มีผู้ใช้งานแบบปลอดภัยออนไลน์ ซึ่งจะแสดงรายชื่อออกมาไม่เหมือนเดิมคือจะแสดงชื่อเป็นอีกระดับหนึ่งคืออยู่ในโหมดของการรักษาความปลอดภัย (Secure)

5.3 การติดต่อเน็ตเวิร์ก

การติดต่อเน็ตเวิร์กโดยมากจะเป็นการโปรแกรมด้วยภาษาจาวาซึ่งจะเป็นการเขียนโปรแกรมติดเพื่อทำการติดต่อและรับ-ส่งข้อมูลระหว่างเซิร์ฟเวอร์ซึ่งต้องทำการ แก้ไขและปรับปรุงการทำงานอย่างต่อเนื่องไปโดยเริ่มที่การร้องขอล็อคอินเข้าใช้บริการ, การร้องขอข้อมูลรายการออนไลน์จากเซิร์ฟเวอร์, การส่งการยืนยันการติดต่อไปยังเซิร์ฟเวอร์ และขั้นตอนต่างๆ ดังกล่าวไว้ในบทที่ 4



รูปที่ 5.2 แสดงโครงสร้างการออกแบบโปรแกรม IsagQ



รูปที่ 5.3 แสดงโปรแกรม IsagQ ที่สร้างจากภาษาจาวา

5.4 การเข้ารหัสข้อมูลแบบ DES, RSA และการโปรแกรม

5.4.1 ลำดับขั้นตอนการส่ง-รับคีย์

เมื่อผู้ใช้ A ต้องการติดต่อกับผู้ใช้ B

1. ผู้ใช้ A จะทำการส่งค่าของคีย์ที่ตนเองสร้างขึ้นมาคือ (e, n) ไปให้ผู้ใช้ B (ทำการส่งพับลิคคีย์ไปให้ผู้ใช้ B)

2. จากนั้นผู้ใช้B จะนำคีย์ที่ได้จากผู้ใช้ A มาทำการเข้ารหัสของคีย์ DES ที่ตนเองสร้างขึ้นมา

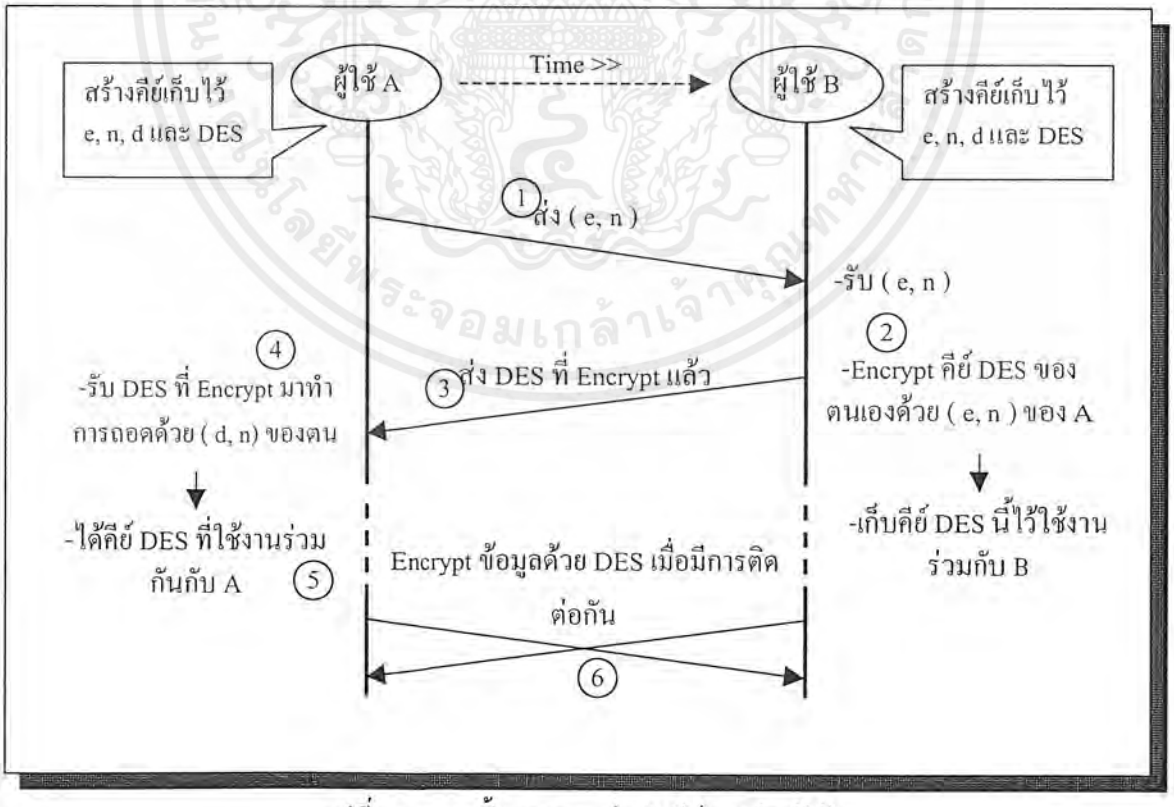
3. ผู้ใช้ B ทำการส่งคีย์ DES ที่ผ่านการเขารหัสด้วยพับลิคคีย์ของผู้ใช้ A ให้กับผู้ใช้ A (ซึ่งแน่ใจได้ว่า A เท่านั้นที่รู้ค่าคีย์ DES ได้คนเดียวเพราะต้องถอดกลับด้วยคีย์ของผู้ใช้ A เท่านั้น)

4. ผู้ใช้ A รับคีย์ DES ที่เข้ารหัสจากผู้ใช้ B ส่งมาให้

5. ผู้ใช้ A ใช้ค่าของคีย์ที่สร้างขึ้นคู่กับพับลิคคีย์ (ก็คือไพรเวทคีย์) ในการถอดรหัสค่าของคีย์ DES ออกมาทำการเก็บไว้เพื่อใช้ในการติดต่อกับผู้ใช้ B ในครั้งต่อไป

6. ทุกครั้งเมื่อผู้ใช้ A จะทำการติดต่อกับผู้ใช้ B จะใช้การเข้ารหัสด้วยคีย์ DES นี้เสมอและผู้ใช้ B ก็จะใช้คีย์ DES นี้ในการติดต่อกับผู้ใช้ A ด้วยเช่นกัน

หมายเหตุ: ขั้นตอนทั้ง 6 นี้จะทำทุกครั้งเมื่อมีการ Online และ Offline ใหม่ซึ่งวิธีการนี้จะใช้เฉพาะกับโปรแกรม IsagQ ของเราเท่านั้น ถ้าผู้ใช้อื่นไม่ได้ใช้ IsagQ จะไม่ต้องทำตามขั้นตอนที่กล่าวมาแล้วนี้เลย แต่ก็สามารถติดต่อและใช้งานได้ตามปกติ



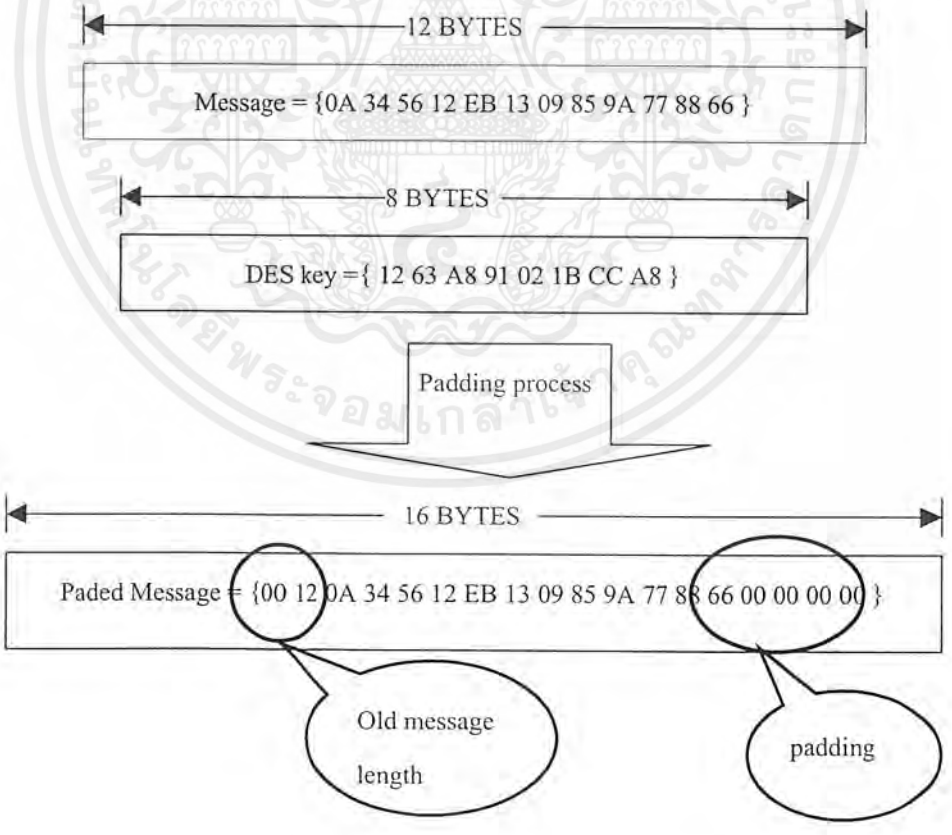
รูปที่ 5.4 แสดงขั้นตอนการส่ง-รับคีย์ระหว่างผู้ใช้

5.4.2 การสร้างคีย์ของการเข้ารหัสแบบคีย์เดี่ยว (DES)

จากทฤษฎีที่ผ่านมาเราทำการสร้างคีย์ของ DES ได้โดยการสุ่มตัวเลขขึ้นมาทำการเก็บไว้เพื่อใช้เป็นคีย์ที่ใช้งานร่วมกันระหว่างผู้ส่งและผู้รับโดยมีขนาดเป็น n เท่าของ 8 เช่น 8, 16 , 24 ,... ค่ายิ่งมากก็ยิ่งทำให้การถอดรหัสข้อมูลกลับเป็นไปได้ยากยิ่งขึ้น ซึ่งในโครงงานนี้จะใช้ 56 ไบต์โดยมีข้อกำหนดของการเข้ารหัสแบบ DES อยู่ว่าขนาดของข้อมูลที่จะนำมาทำการเข้ารหัสนั้นต้องสามารถหารด้วย 8 ได้ลงตัว เพราะการเข้ารหัสแบบ DES จะใช้การแบ่งข้อมูลออกเป็นบล็อกๆ ก่อนจึงค่อยทำการเข้ารหัสด้วยวิธีการของ DES ได้ดังนั้นหากข้อมูลที่จะนำมาเข้ารหัส มีไม่ครบตามขนาดบล็อกก็ต้องทำการเติมในส่วนของข้อมูลนั้นให้มีขนาดเต็มบล็อก ซึ่งเรียกการทำการเติมเต็มนี้ว่าการทำ Padding ซึ่งค่าของ padding นี้จะมีขนาด 1-8 ไบต์ซึ่งจำนวนของการทำ padding นี้สามารถหาได้จาก (8 – (ความยาวแพกเก็จ % 8)) ไบต์ (% คือ modulo) การที่มีการ padding จะช่วยให้การหาข้อมูล (plain text) ทำได้ยากขึ้น

ขั้นตอนการทำงานกระบวนการ padding ข้อมูล

การ Padding เป็นการสร้างให้ข้อความมีความยาวเป็นจำนวนเท่าของ DES key เพื่อการเข้ารหัสแบบ DES มีประสิทธิภาพมากขึ้น



รูปที่ 5.5 แสดงการเข้ารหัสแบบ DES และการ padding

5.4.3 การสร้างคีย์ของการเข้ารหัสแบบคีย์คู่ (RSA)

ระบบกุญแจคู่ดังที่ทราบมาแล้วว่าจะมีค่าคีย์ที่สำคัญด้วยกันอยู่ 3 ค่าคือค่าของ e, n และ d ค่าทั้งสามนี้จะนำไปใช้ในขั้นตอนของการเข้ารหัสโดยใช้ค่า e, n (เรียกว่าพับลิคคีย์) และเมื่อเข้าสู่ขั้นตอนการถอดรหัสจะต้องใช้ค่าของคีย์ d, n (เรียกว่าไพรเวตคีย์) ซึ่งใช้ในการถอดรหัส

ในส่วนของโครงงานนี้เราสามารถทำการสร้างค่าคีย์ได้โดยใช้ภาษาจาวาดังนี้

```
BigInteger one    = new BigInteger("1");
BigInteger p      = new BigInteger(512, 20, new Random());  // กำหนดค่าของ p
BigInteger q      = new BigInteger(512, 20, new Random());  // กำหนดค่าของ q
BigInteger n      = p.multiply(q);   // ค่าของ n ได้จากการนำ p x q
BigInteger phi    = p.subtract(one).multiply(q.subtract(one));
BigInteger e      = new BigInteger(8, 1, new Random());  // กำหนดค่าของ e
BigInteger gcd    = phi.gcd(e);
while (!gcd.equals(one))   {   // วนลูปเพื่อหาค่าคีย์ที่เป็นคู่กันอีก 1 คีย์
        e         = new BigInteger(8, 1, new Random());
        gcd       = phi.gcd(e);
}
BigInteger d      = e.modInverse(phi);
```

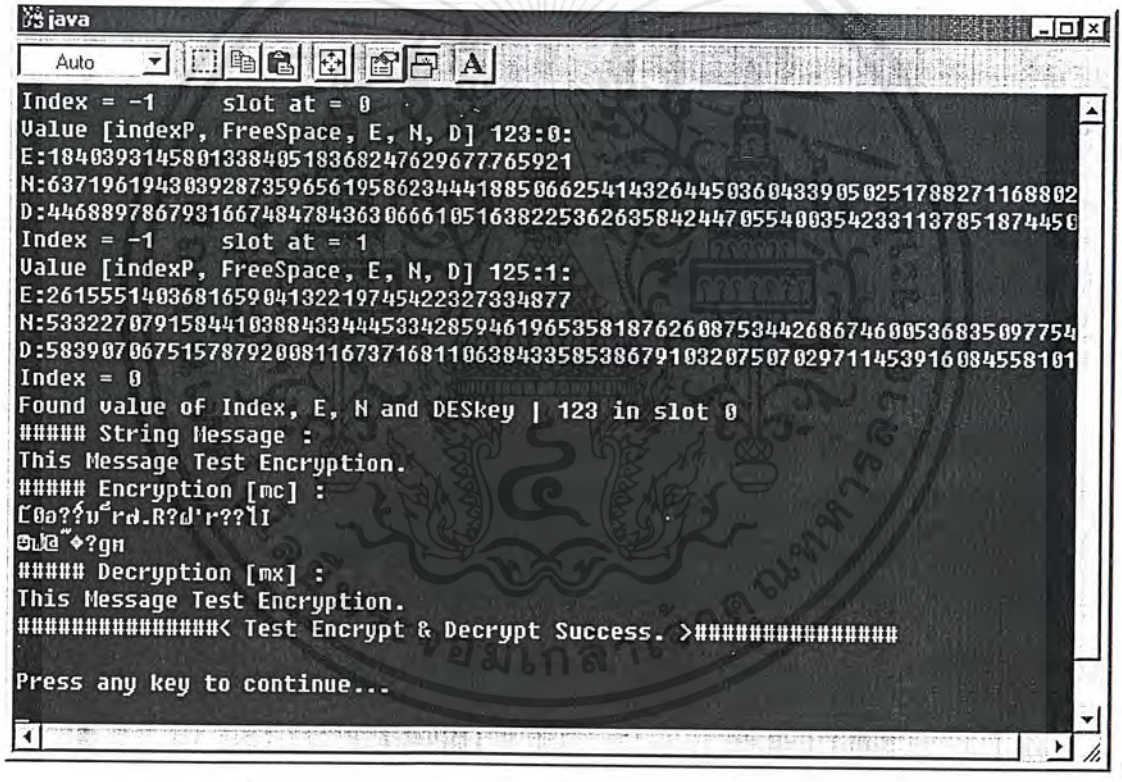
จากโค้ดของโปรแกรมภาษาจาวาจะได้ค่าของ e, n และ d เพื่อใช้ในการเข้ารหัสข้อมูลแบบ RSA ได้แล้วโดยต้องทำการเก็บค่าเหล่านี้ไว้เพื่อใช้ในการติดต่อสื่อสารกับผู้ที่ต้องการความปลอดภัยโดยต้องทำการเก็บค่าของ d ไว้เป็นความลับและใช้ในการถอดรหัสข้อมูลส่วน e, n จะส่งให้ผู้ที่ต้องการติดต่อด้วยเพื่อใช้ในการเข้ารหัสก่อนส่งข้อมูลมาหาเรา

5.5 ขั้นตอนการทดสอบและรวมส่วนประกอบต่างๆ เข้าด้วยกัน

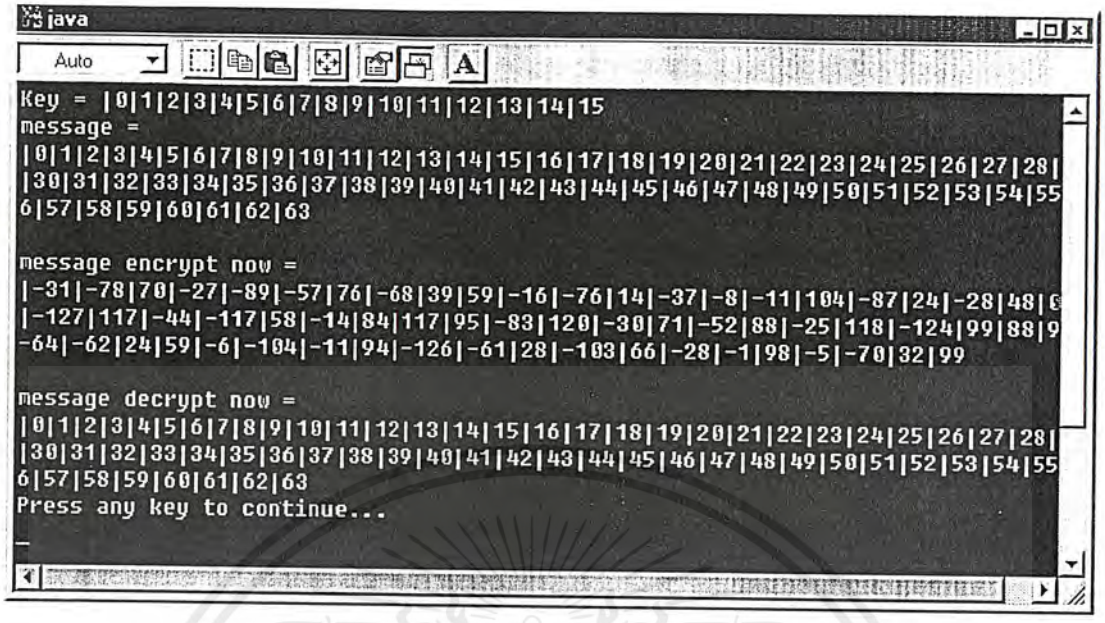
จากส่วนประกอบทั้งสามส่วนที่ได้ทำการออกแบบไว้ ก่อนที่จะนำมารวมเข้าด้วยกันจะต้องทำการทดสอบส่วนประกอบแต่ละส่วนก่อนว่าสามารถทำงานได้ตามความต้องการที่ต้องการหรือไม่เช่น

ส่วนของการติดต่อกับผู้ใช้งาน:- จะต้องสามารถตอบสนองต่อการใช้งานได้อย่างสมบูรณ์โดยจะต้องสามารถตรวจสอบได้ว่าขณะนี้ผู้ใช้ทำการกดคลิกที่ตำแหน่งใด และที่ตำแหน่งนั้นจะต้องมีเหตุการณ์อะไรตอบสนองออกมา โดยทำการทดสอบปิดเปิดไดอะล็อกต่างๆ ที่ได้ทำการออกแบบไว้ให้มีความสัมพันธ์กันด้วย

ส่วนของการเข้า-ถอดรหัสข้อมูล:- สามารถทำการทดสอบได้โดยง่ายโดยการจำลองการทำงานขึ้นมาว่าเราต้องทำการสร้างคีย์ต่างๆ ไว้สำหรับผู้ใช้สองคน จากนั้นทำการเก็บและทำการเรียกออกมาทำการเข้ารหัสและทำการถอดรหัส ออกโดยใช้คีย์ที่เป็นไพรเวตคีย์ (ที่คู่กันกับพับลิคคีย์นั้น) หากสามารถทำการถอดรหัสข้อมูลได้สำเร็จ (ได้ข้อมูลออกมาเหมือนที่ได้สร้างขั้น) ก็แสดงว่าโปรแกรมการเข้า-ถอดรหัสสามารถทำงานได้อย่างถูกต้อง โดยรูปที่ 5.6 แสดงผลของการทดลองการเข้า-ถอดรหัสแบบ RSA เป็นการจำลองการใช้งานจริงขึ้น โดยทำการสร้างคีย์ของผู้ใช้ทั้งสองฝ่ายขึ้นมา จากนั้นจึงนำค่าคีย์ที่ได้มาทำการเข้ารหัสและทำการถอดรหัส แล้วแสดงผลออกมาว่าถูกต้องหรือไม่ ส่วนรูปที่ 5.7 จะแสดงการทดลองเข้า-ถอดรหัสแบบ DES โดยแสดงค่าคีย์ของการเข้ารหัสในที่นี้คือ เลข 0 – 15 จากนั้นทำการสร้างข้อมูลหรือข้อความที่ต้องการเข้ารหัสซึ่งใช้เป็นตัวเลข 0-64 แล้วทำการเข้า-ถอดรหัสข้อมูลออกมาได้ท่าที่เท่าเดิมก็แสดงว่าการทำงานของโปรแกรมทำงานได้อย่างถูกต้องสมบูรณ์



รูปที่ 5.6 แสดงผลการทดลองการเข้า-ถอดรหัสแบบกุญแจคู่ (RSA)



รูปที่ 5.7 แสดงผลการทดลองการเข้า-ถอดรหัสแบบกุญแจเดี่ยว (DES)

ส่วนของการติดต่อเน็ตเวิร์ก:- ในส่วนของการติดต่อเน็ตเวิร์กจะใช้โพรโตคอลของ ICQ เป็นหลักในการเขียนโปรแกรมซึ่งจะใช้การติดต่อ 2 รูปแบบด้วยกันคือ TCP และ UDP โดยที่การติดต่อระหว่างผู้ใช้กับเซิร์ฟเวอร์นั้นจะใช้การติดต่อแบบ UDP ส่วนการติดต่อระหว่างไคลเอนกับไคลเอนจะใช้การติดต่อแบบ UDP และ TCP โดยที่จะใช้การติดต่อแบบ TCP เป็นส่วนมากและจะใช้ UDP ในกรณีที่ไม่สามารถติดต่อผู้ใช้อีกฝ่ายได้โดยตรงก็จะทำการติดต่อไปยังเซิร์ฟเวอร์โดยใช้ UDP การทดสอบก็จะดูจากการคอนเน็กและ การตอบกลับว่าสามารถติดต่อกันได้ตลอดหรือไม่? หรือว่าไม่มีการตอบกลับ? หรือว่ามีการตัดการติดต่อระหว่างการใช้งาน

เมื่อทำการทดสอบทั้งสามส่วนที่ได้สร้างขึ้นแล้วก็มาถึงขั้นตอนการรวมเข้าด้วยกัน ซึ่งในโครงงานนี้ได้ทำการรวมส่วนของการติดต่อกับผู้ใช้กับส่วนของการติดต่อเน็ตเวิร์กก่อน ว่าสามารถทำงานได้ตามที่กำหนดไว้หรือไม่ เมื่อทำงานได้แล้วจึงค่อยนำส่วนของการเข้ารหัสมาทำการรวมอีกขั้นตอนหนึ่ง

บทที่ 6
การติดตั้งและการทำงานของโปรแกรม IsagQ

6.1 การติดตั้งโปรแกรม IsagQ

ขั้นตอนการติดตั้งโปรแกรม IsagQ นั้นไม่ยุ่งยากซับซ้อน เพียงแค่นำไฟล์ที่อยู่ในโฟลเดอร์ IsagQ ไปติดตั้งไว้ที่ไดเรกทอรีเดียวกับ JDK (ในที่นี้ใช้ JDK1.1.8 ในการอธิบาย) หรือที่ไดเรกทอรี JavaIsagQ ซึ่งในที่นี้เราจะติดตั้งไว้ที่ไดเรกทอรี ของไดร์ฟซี (C:\JavaIsagQ) โดยทำการก็อบปี้ลงไปเก็บไว้ดังนี้

```
C:\>xcopy  a:*.* C:\JavaIsagQ /s
```

หลังจากทำการก็อบปี้ไฟล์ไปเก็บไว้แล้วก็จะเข้าสู่ขั้นตอนการเรียกใช้งานโปรแกรม แต่ก่อนจะไปสู่ขั้นตอนการเรียกใช้โปรแกรม IsagQ ต้องมั่นใจก่อนว่าเราได้ทำการตั้งค่าไดเรกทอรีพาร์ต (Directory path)ไปที่ c:\jdk1.1.8\bin และคลาสไฟล์ไปที่ c:\jdk1.1.8\lib\classes.zip โดยทำการเพิ่มประโยคต่อไปนี้ในไฟล์ autoexec.bat (ใช้กับระบบปฏิบัติการวินโดว์ 95/98/2000 เท่านั้น)

```
SET PATH=%PATH%C:\JDK1.1.8\BIN
SET CLASSPATH=%CLASSPATH%.;C:\JDK1.1.8\LIB\CLASSES.ZIP
```

6.2 ขั้นตอนการเรียกโปรแกรม IsagQ ขึ้นมาทำงาน

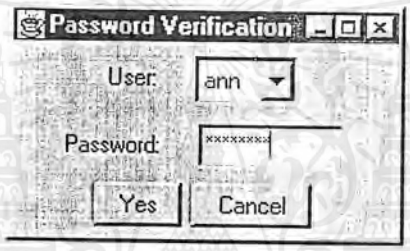
การเรียกโปรแกรมขึ้นมาทำงานให้ทำการเปิดโปรแกรม MS-DOS หรือ Terminal ของระบบ Linux ขึ้นมาทำงานในที่นี้จะแสดงให้ดูเป็นการใช้งาน MS-DOS เป็นตัวอย่างประกอบ โปรแกรมหลักของ IsagQ จะอยู่ในโฟลเดอร์ JavaIsagQ โดยภายในจะประกอบด้วยไฟล์ต่างๆ รวมถึงมีแพ็กเกจต่างๆ อยู่ด้วยดังนั้นการเรียกโปรแกรมขึ้นมาทำงานจึงต้องทำการพิมพ์ดังนี้

```
C:\JavaIsagQ\java IsagQ.Main  [Enter]
```

เมื่อทำการกดปุ่ม Enter โปรแกรม IsagQ จะเริ่มทำงานโดยจะปรากฏไดอะล็อกของโปรแกรมขึ้นมาแสดงว่าโปรแกรมได้เข้าสู่ขั้นตอนการทำงานแล้ว แต่หากมีข้อผิดพลาดเกิดขึ้นให้ทำการตรวจสอบดูว่ามีการเซตพาร์ต ดังที่กล่าวไว้ข้างต้นหรือยัง และอาจตรวจสอบดูด้วยว่าโปรแกรม MainIsagQ.class ที่อยู่ในไดเรกทอรี JavaIsagQ\IsagQ มีอยู่หรือไม่

6.3 การทำงานของโปรแกรม IsagQ

เมื่อเราทำการเรียกโปรแกรมขึ้นมาทำงานจากหัวข้อที่ผ่านมาแล้วโปรแกรมก็จะเริ่มเข้าสู่การทำงานของโปรแกรม IsagQ โดยแรกเริ่มโปรแกรมจะแสดงไดอะล็อกการล็อคอินเข้าใช้งานโปรแกรม โดยให้ผู้ใช้ทำการใส่พาสเวิร์ดซึ่งจะเป็นชนิดเดียวกับที่ใช้ติดต่อกับเซิร์ฟเวอร์ของ Mirabilis หลังจากนั้นโปรแกรมจะทำการติดต่อไปยังเซิร์ฟเวอร์ของ Mirabilis (icq.mirabilis.com : port 4000) หากพาสเวิร์ดผิดพลาดก็จะถูกตัดการติดต่อจากเซิร์ฟเวอร์ แต่ถ้าถูกต้องก็จะเข้าสู่ขั้นตอนการตอบรับและร้องขอข้อมูลของผู้ใช้จากเซิร์ฟเวอร์โดยเซิร์ฟเวอร์จะทำการตรวจสอบว่ามีผู้ใช้งานคนใดที่เราติดต่อด้วยออนไลน์อยู่บ้าง แล้วทำการส่งผลกลับมาให้โปรแกรม IsagQ ซึ่งก็จะแสดงสถานะการใช้งานที่แตกต่างกันออกไปแล้วแต่การตั้งค่าของผู้ใช้งานแต่ละคนที่เราติดต่อด้วย

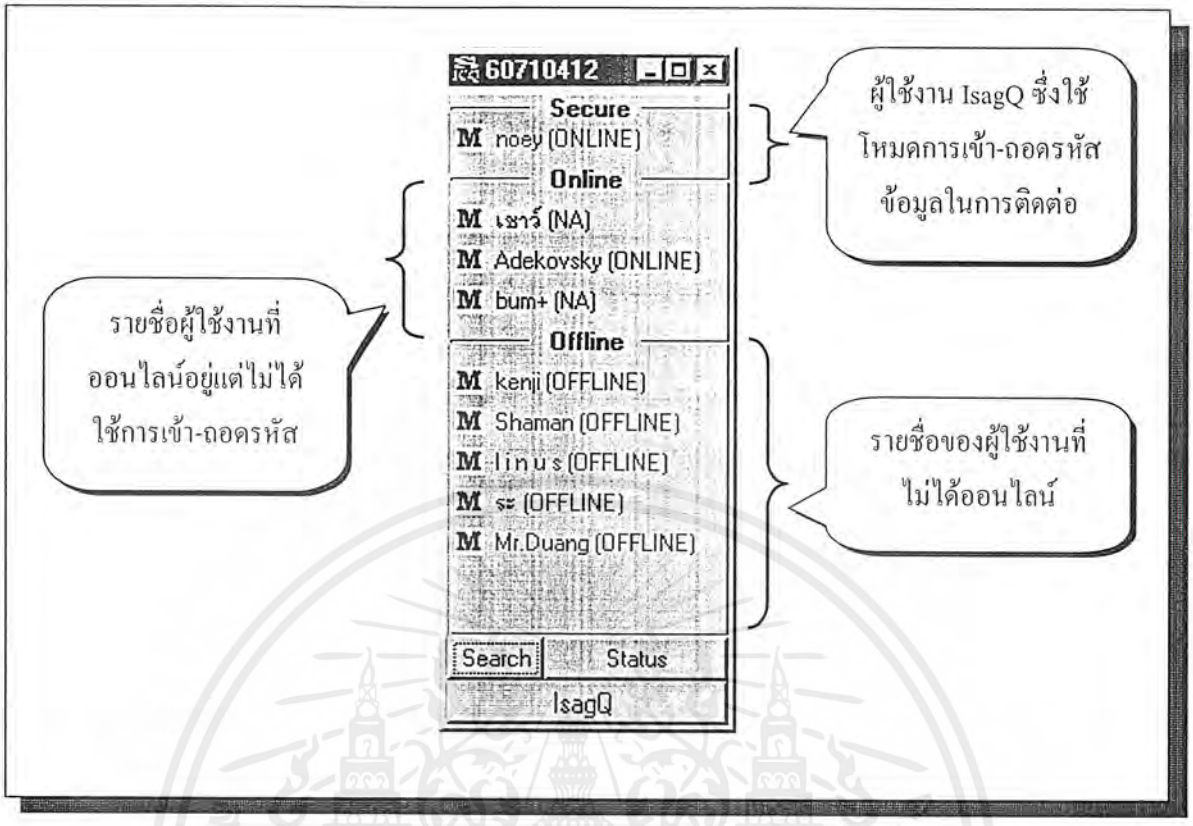


รูปที่ 6.1 แสดงการถามรหัสผ่านก่อนการเข้าใช้งานโปรแกรม IsagQ

การรักษาความปลอดภัยอันดับแรกในโปรแกรม IsagQ คือการตรวจสอบสิทธิผู้ใช้โปรแกรม โดยขึ้นไดอะล็อก Password Verification สามารถเลือกผู้ใช้ได้ จากนั้นใส่รหัสผ่าน หลังจากนั้นโปรแกรมตรวจสอบรหัสผ่าน หากการตรวจสอบรหัสผ่านถูกต้องโปรแกรม IsagQ จะทำงานโดยเข้าสู่ขั้นตอนของการติดต่อเซิร์ฟเวอร์ต่อไป

จากนั้นโปรแกรม IsagQ จะทำการสร้างคีย์ต่างๆ ขึ้นมาโดยจะมีคีย์ e, n, d และ DES เพื่อใช้ในการเข้ารหัสข้อมูลหลังจากนั้นจะทำการส่งค่าคีย์ e, n (พับลิคคีย์) ออกไปให้ผู้ใช้งานรายอื่นๆ ซึ่งอยู่ในรายการที่ออนไลน์ เพื่อตรวจสอบดูว่าผู้ใช้รายใดใช้โปรแกรมที่มีความปลอดภัย IsagQ เช่นเดียวกับเราบ้าง โดยผู้ใช้รายอื่นที่ได้รับคีย์ e, n นี้จะต้องทำการตอบกลับมายังผู้ส่งว่าได้รับคีย์ e, n แล้วพร้อมทั้งทำการส่งคีย์ที่จะใช้ติดต่อกันในครั้งต่อไปว่าจะใช้คีย์ใดในการเข้ารหัสและถอดรหัสระหว่างผู้ใช้ทั้งสองคน (ถ้าไม่มีการตอบกลับแสดงว่าไม่ได้ใช้งานโปรแกรม IsagQ และจะไม่แสดงสถานะที่แถบ Secure)

เมื่อผู้ใช้รายใดทำการตอบกลับก็แสดงว่ายินดีที่จะใช้การติดต่อกันโดยการเข้ารหัสระหว่างกัน จากนั้นรายชื่อของผู้ใช้นั้นจะถูกเลื่อนเข้าไปสู่กลุ่มของผู้ใช้ที่มีความปลอดภัย (ในโครงงานนี้จะมี 3 กลุ่ม คือกลุ่มที่ใช้ IsagQ, กลุ่มผู้ใช้ที่ Online และกลุ่มผู้ใช้ที่ Offline) โปรแกรมจะแสดงสีรายชื่อของผู้ใช้งานที่ต่างกันไป เพื่อแบ่งแยกประเภทของผู้ใช้



รูปที่ 6.2 แสดงหน้าจอของโปรแกรมและกลุ่มรายชื่อผู้ใช้ที่ต่างกัน

เมื่อผู้ใช้ทำการส่งข้อความหาผู้ที่ต้องการติดต่อที่อยู่ในรายการผู้ใช้ที่ Online ก็จะให้การส่งข้อมูลที่ปกติกล่าวคือไม่มีการเข้ารหัสข้อมูล แต่หากมีการติดต่อกับผู้ใช้ที่เป็น IsagQ ซึ่งใช้การเข้ารหัสข้อมูลก่อนทำการส่งโปรแกรมจะทำการนำค่าของคีย์ที่ทำการตกลงกันไว้เมื่อตอนเริ่มต้นโปรแกรมออกมาทำการเข้ารหัสก่อนทำการส่งข้อมูลให้กัน

บทที่ 7
การทดสอบโปรแกรม IsagQ

การทดสอบโปรแกรมสามารถทำการทดสอบได้หลายแบบ แต่ในโครงงานนี้จะทำการทดสอบให้เห็นถึงความปลอดภัยในการใช้งานโดยการรับ-ส่งข้อมูลแบบปกติ และการรับส่งข้อมูลแบบมีการเข้ารหัสของข้อมูลระหว่างการติดต่อกัน เพื่อให้สามารถเห็นผลการทดสอบได้อย่างชัดเจนทางผู้พัฒนาจึงแยกการแสดงผลออกเป็น 2 ชนิดเพื่อให้เข้าใจได้ง่าย

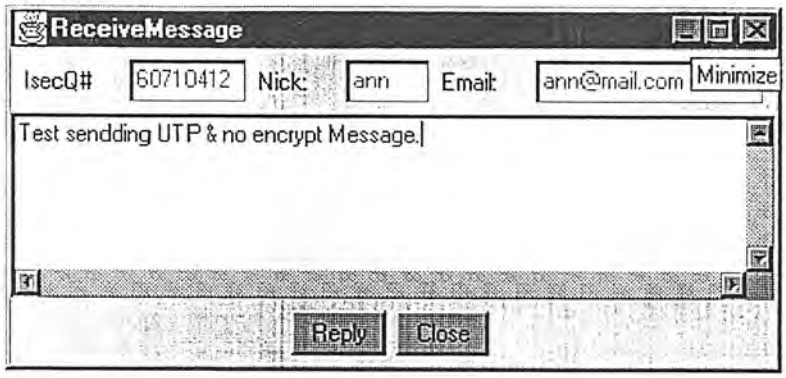
7.1 การทดสอบรับ-ส่งข้อความแบบปกติและผลการตรวจจับข้อความแบบปกติด้วย sniffer

ขั้นตอนในการทดสอบโปรแกรม IsagQ โดยการรับส่งข้อมูลปกตินี้มีการทำงานที่ไม่ยุ่งยาก เพียงแค่ทำการรับส่งข้อมูลตามปกติแล้วใช้โปรแกรมที่ตรวจจับข้อมูล (ในโครงงานนี้ใช้โปรแกรม Sniffer Pro version 2.50.07) มาทำการตรวจจับข้อมูลซึ่งก็จะได้ข้อมูลที่ทำการรับ-ส่งระหว่างกันได้

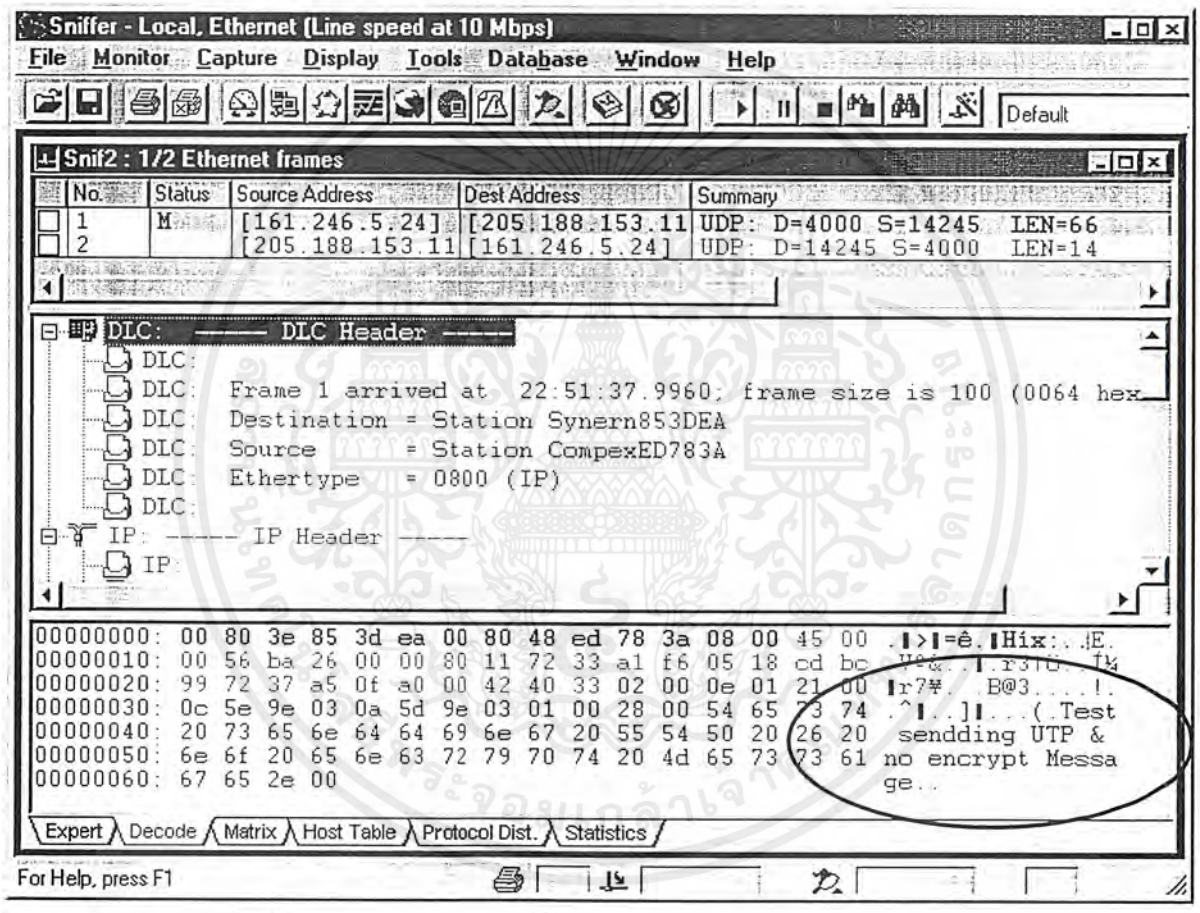
7.1.1 การทดสอบส่ง-รับข้อความแบบปกติ

เมื่อรับส่งข้อมูลระหว่างกันจะได้ผลของการรับส่งข้อมูลดังนี้ รูปที่ 7.1 จะแสดงถึงการส่งข้อมูลแบบปกติและเป็นการใช้การส่งข้อมูลแบบ UDP ซึ่งจะส่งไปผ่านที่เซิร์ฟเวอร์ก่อน (Through Server) ส่วนรูปที่ 7.2 แสดงการรับข้อมูลแบบธรรมดาและทำการนำมาแสดงออกที่ไดอะล็อก การรับข้อมูลแบบธรรมดา

รูปที่ 7.1 ไดอะล็อกแสดงข้อมูลที่ต้องการส่งแบบปกติ (ทางฝ่ายผู้ส่ง)



รูปที่ 7.2 ไดอะล็อกแสดงข้อมูลที่ได้รับแบบปกติ (ทางฝ่ายผู้รับ)



รูปที่ 7.3 แสดงหน้าจอข้อมูลที่ได้จากการตรวจจับจากการส่ง-รับแบบปกติ

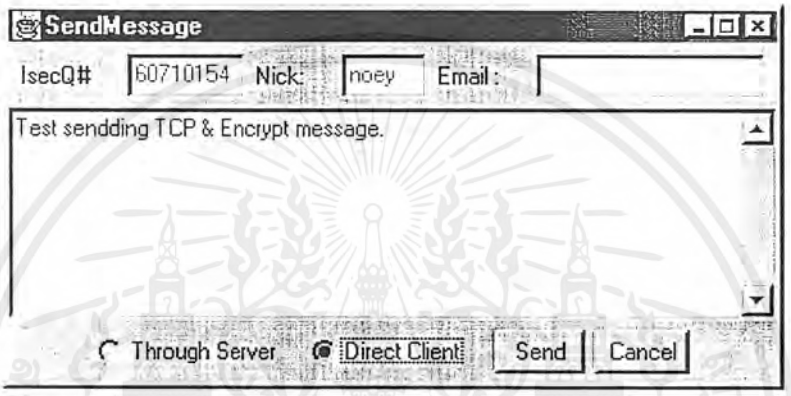
7.1.2 ผลการตรวจจับข้อความแบบปกติ

ข้อมูลที่เราสามารถตรวจจับได้จากการทดสอบนี้เป็นรูปแบบของ UDP ซึ่งใช้โปรแกรม Sniffer Proในการดักจับเมสเซจนี้ และเมื่อเราดูในส่วนของข้อมูลก็จะสามารถอ่านข้อความออกมาได้โดยง่าย ซึ่งแสดงถึงความไม่ปลอดภัยในเรื่องของข้อมูลระหว่างการติดต่อเลย (ข้อมูลที่ดักจับได้สามารถอ่านได้เช่นเดียวกับข้อมูลที่ทำการรับ-ส่งจากผู้ใช้ในรูปที่ 7.1 และ 7.2) โดยรูปที่ 7.3 แสดงหน้าจอการดักจับข้อความของรูปที่ 7.1 และ 7.2

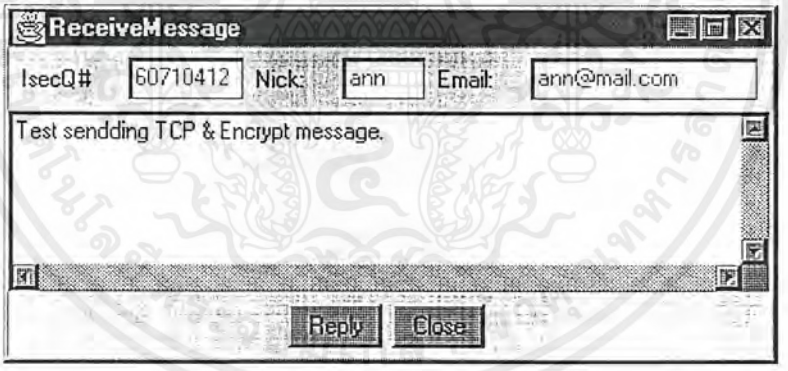
7.2 การทดสอบรับ-ส่งข้อความแบบมีการเข้ารหัสและผลการตรวจจับข้อความที่เข้ารหัสด้วย sniffer

ส่วนของขั้นตอนในการทดสอบโปรแกรม IsagQ โดยการรับ-ส่งข้อมูลแบบเข้ารหัสข้อมูล ก่อนการรับ-ส่งนี้จะมีการทำงานที่ยุ่งยากกว่าแบบการรับส่งแบบปกติ กล่าวคือต้องทำการนำเอาข้อมูลที่ต้องการรับ-ส่งมาทำการเข้ารหัสก่อนจึงค่อยทำการรับ-ส่งได้ตามปกติ ทำให้ข้อมูลที่ส่งออกไปไม่สามารถอ่านได้โดยง่าย เพราะจะต้องมีการถอดรหัสข้อมูลก่อนจึงจะอ่านได้ ดังนั้นหากไม่รู้ค่าคีย์ที่ใช้ในการถอดรหัสนี้ ก็ไม่สามารถอ่านข้อความนี้ได้

7.2.1 การทดสอบส่ง-รับข้อความแบบใช้การเข้ารหัสข้อความ



รูปที่ 7.4 ไดอะล็อกแสดงข้อมูลที่ต้องการส่งแบบมีการเข้ารหัสข้อความก่อนการส่ง

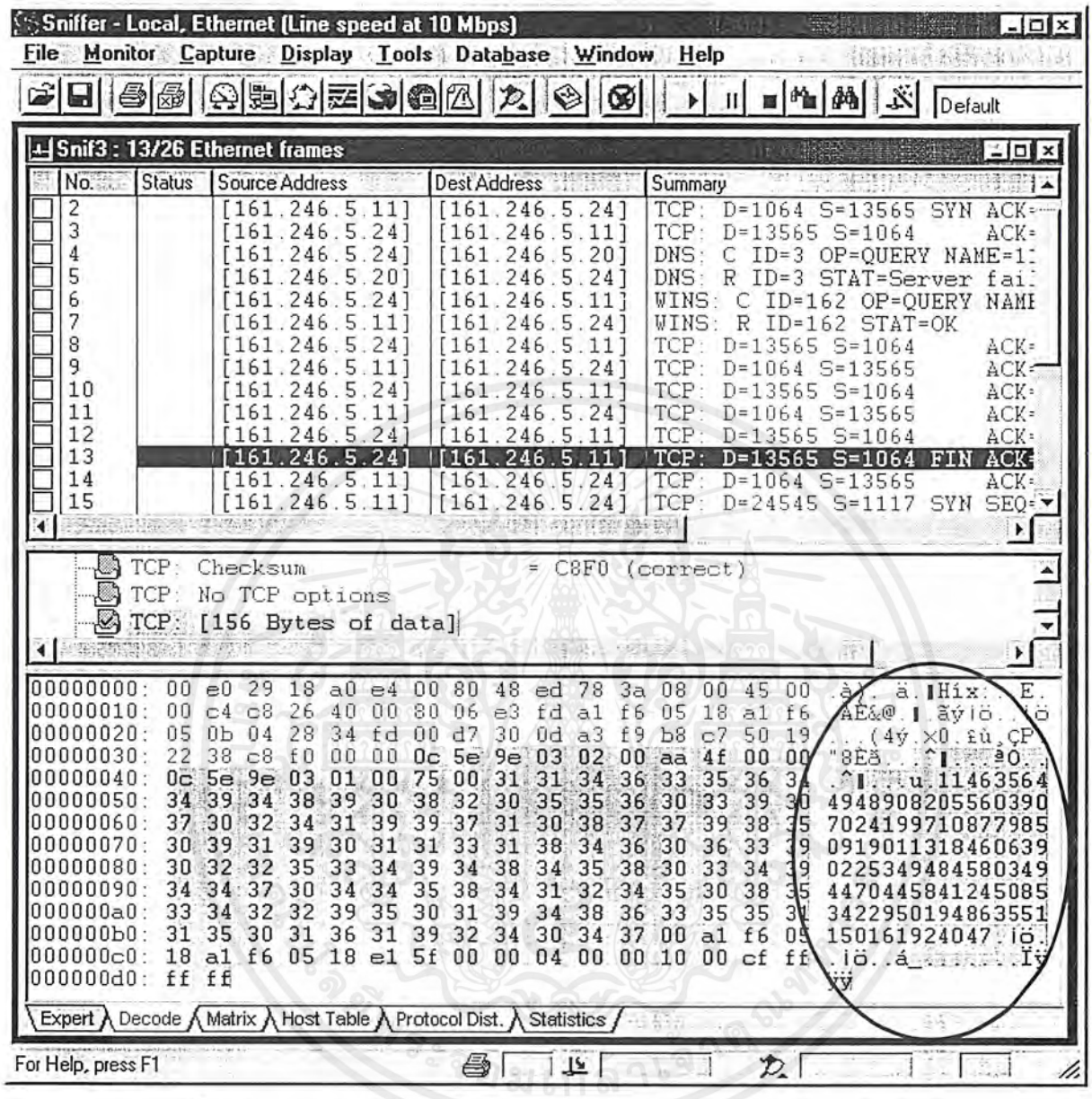


รูปที่ 7.5 ไดอะล็อกแสดง ข้อมูลที่ได้รับแบบมีการเข้ารหัสมาทำการถอดรหัสออก

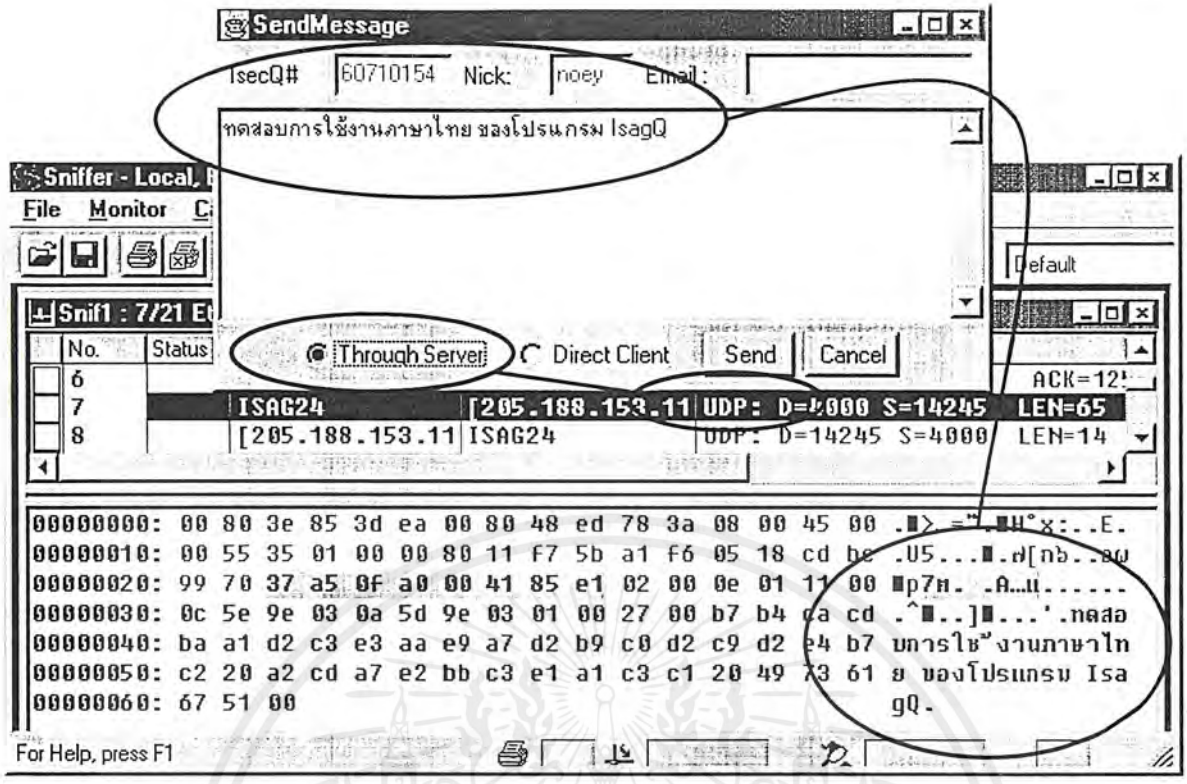
จากรูปที่ 7.6 เป็นผลที่ได้จากการ sniff การส่ง-รับจากรูปที่ 7.4 และ 7.5 แต่ผลที่ได้คือข้อมูลมีการเข้ารหัสแล้วไม่สามารถอ่านออกมาได้ตามปกติ ซึ่งต้องใช้คีย์ที่เข้ารหัสเท่านั้นในการถอดรหัสนี้ได้ จึงมั่นใจได้ว่าโปรแกรม IsagQ ของเราสามารถป้องกันการดักจับข้อมูลได้ตามสเปกที่ได้กำหนดไว้

และรูปที่ 7.7 และ 7.8 ที่จะแสดงในหน้าต่อไปเป็นการทดสอบการใช้งานภาษาไทย ว่าสามารถทำการส่ง-รับภาษาไทยได้ตามสเปกที่ต้องการหรือไม่ โดยได้ใช้การดักจับข้อความดู ผลที่ได้เป็นที่น่าพอใจเพราะว่าการส่งข้อมูลแบบ UDP จะไม่มีการเข้ารหัสทำให้สามารถดักจับข้อความมาอ่านได้โดยง่าย แต่การส่งข้อมูลแบบ TCP จะมีการเข้ารหัสทำให้การดักจับครั้งนั้นได้ข้อมูลที่เป็นขยะ ไม่สามารถอ่านออกได้อย่างง่ายดายแต่ต้องใช้คีย์ในการถอดรหัสข้อมูลนี้ก่อนจึงจะอ่านข้อความได้

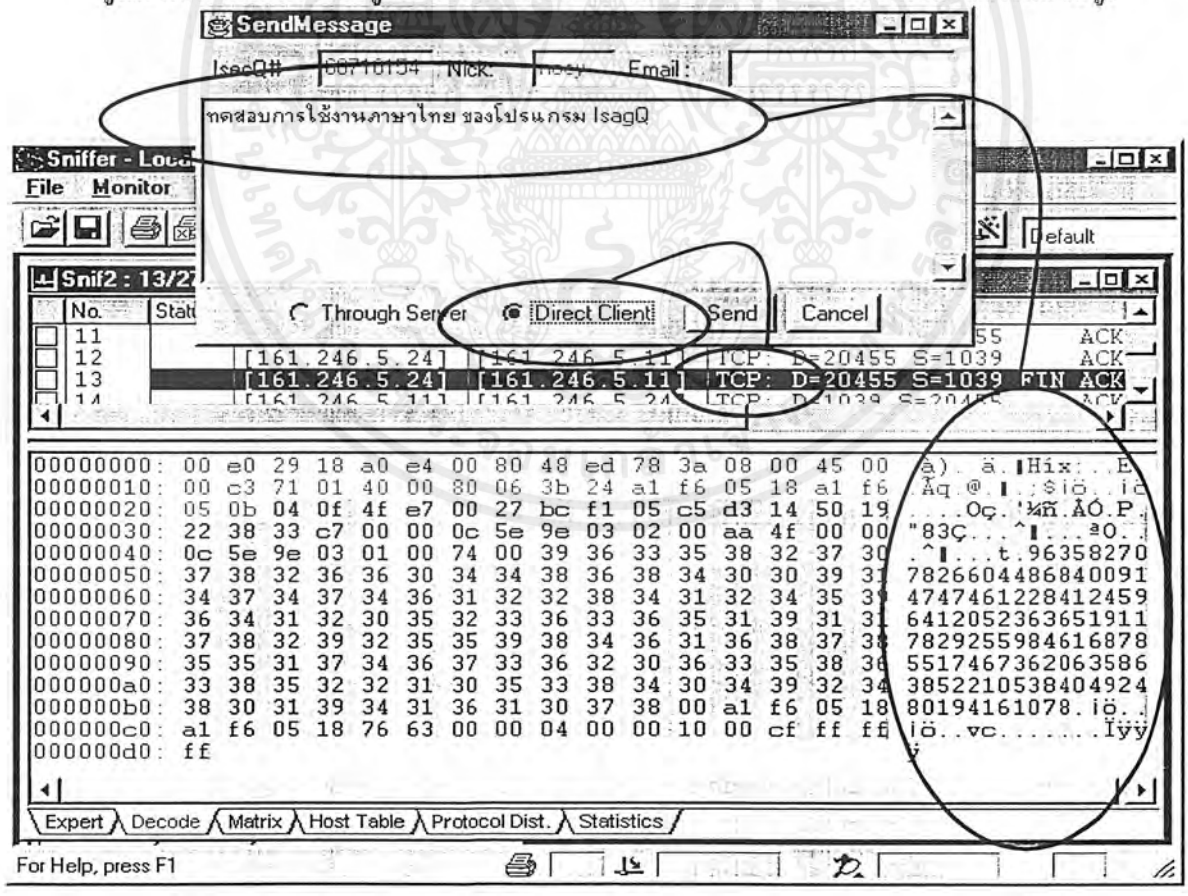
7.2.2 ผลการตรวจจับข้อความแบบที่มีการการเข้ารหัสข้อความ



รูปที่ 7.6 แสดงหน้าจอข้อมูลที่ได้จากการตรวจจับจากการรับส่งแบบมีการเข้ารหัสข้อมูล



รูปที่ 7.7 แสดงหน้าจอข้อมูลที่ได้จากการตรวจจับและไดอะล็อกการส่งแบบไม่มีการเข้ารหัสข้อมูล



รูปที่ 7.8 แสดงหน้าจอข้อมูลที่ได้จากการตรวจจับและไดอะล็อกการส่งแบบมีการเข้ารหัสข้อมูล

บทที่ 8
สรุปผลการพัฒนาโครงงาน

8.1 ผลการทดลองโปรแกรม

ผลการทดลองโปรแกรมดูได้จากข้อ 6 ซึ่งจะเป็นการทดสอบโปรแกรมว่าสามารถทำงานได้ตามสเปกที่กำหนดไว้และสามารถใช้งานได้กับโปรแกรม ICQ ธรรมดาได้ด้วยดี

8.2 คุณสมบัติของโปรแกรม

สามารถพัฒนาโครงงานได้ครบตามความสามารถที่ได้กำหนดไว้ซึ่งมีดังนี้

: Input/output specification

- รับค่าการพิมพ์ทางคีย์บอร์ดและเมาส์ได้
- แสดงผลทางจอภาพโดยเป็นกราฟิกได้

: Function specification

- รับส่งข้อความเป็น Plain text ระหว่างผู้ใช้ ICQ และโปรแกรม IsagQ ได้
- ใส่ User login และพาสเวิร์ดใช้งานโปรแกรม IsagQ ได้
- แสดงสถานะออนไลน์ (Online) , ออฟไลน์ (Offline) และ Secure ICQ ที่ติดต่อด้วยได้
- ค้นหาผู้ใช้ทาง UIN, E-mail, Name ได้
- เข้ารหัส (Encryption) และถอดรหัส (Decryption) ข้อความระหว่างผู้ใช้โปรแกรม IsagQ และโปรแกรม ICQ ที่มีใช้งานทั่วไปได้
- ใช้งานภาษาไทยได้

8.3 ประโยชน์ของโครงการ

- สามารถติดต่อสื่อสารข้อความกับผู้ใช้ ICQ อื่นๆ ได้ง่ายและสะดวกเช่นเดียวกับโปรแกรม ICQ ทั่วไป
- สามารถทำงานข้ามแพลตฟอร์มไปยังแพลตฟอร์มอื่นที่สนับสนุน Java Virtual Machine ได้
- โปรแกรม IsagQ ตรวจสอบสิทธิของผู้ใช้ทุกครั้งที่เริ่มใช้งาน โดยตรวจสอบพาสเวิร์ด ซึ่งปลอดภัยกว่าโปรแกรม ICQ ทั่วไปเพราะเมื่อใช้กับโปรแกรม ICQ ทั่วไปสามารถยกเลิกการทำงานส่วนนี้ได้
- การส่งข้อความระหว่างผู้ใช้โปรแกรม IsagQ จะทำการเข้ารหัสซึ่งปลอดภัยต่อการดักจับข้อความมาอ่านจากเครื่องคอมพิวเตอร์เครื่องอื่นได้

8.4 ข้อจำกัด

- การทำงานของโปรแกรมช้ากว่าโปรแกรม ICQ ทั่วไป
- ยังไม่มีการควบคุมเรื่องการรับรอง ไพรเวต, พับลิกคีย์ ในแบบที่มั่นใจได้ดีพอ

8.5 ข้อเสนอแนะหลังการทำโครงงานสำหรับผู้นำโครงงานไปพัฒนาต่อไป

- ขั้นตอนตรวจสอบผู้ใช้ IsagQ และการแลกคีย์ยังมีข้อเสียสำหรับ โปรแกรม ICQ เวอร์ชันเก่าๆ ซึ่งจะปรากฏ public key บนหน้าจอของผู้อื่นที่มีรายชื่อในรายการผู้ที่ต้องการติดต่อด้วยทำให้เป็นการรบกวน
- ควรเพิ่มเติมหรือแก้ไขข้อจำกัดที่กล่าวไปแล้วเพื่อสะดวกในการใช้สื่อสารกับผู้อื่นได้อย่างเต็มที่
- เรื่องความมีเสถียรภาพของโปรแกรมควรมีมากกว่านี้

ภาคผนวก ก.
ความรู้ทั่วไปเกี่ยวกับโปรแกรม ICQ

ก.1 ความรู้ทั่วไปเกี่ยวกับโปรแกรม ICQ

โปรแกรม ICQ ที่มีใช้อยู่ในปัจจุบันนี้เป็นคำพ้องเสียงมาจากคำว่า I seek you ซึ่งก็หมายถึงเครื่องมือที่ใช้ในการค้นหาเพื่อนหรือคู่สนทนา โดยใช้การติดต่อกันโดยใช้อินเตอร์เน็ตเป็นสื่อ ซึ่งเราสามารถรับทราบได้ว่าคู่สนทนาของเราออนไลน์อยู่หรือไม่ โดยสามารถสนทนาโต้ตอบกันไปมาได้ทันที หรือจะทำการฝากข้อความไว้ให้คู่สนทนาของเราในกรณีที่ไม่ได้ออนไลน์อยู่ในขณะนั้นก็สามารถทำได้ โดยมีลักษณะการทำงานที่เรียกว่า Internet Instant-Messaging โดยที่ผู้สมัครเป็นสมาชิกของ ICQ จะได้รับหมายเลขประจำตัวทุกคนเรียกว่า User Identification Number หรือ UIN เป็นหมายเลขประจำตัวของผู้สมัครเป็นสมาชิกของ ICQ เพื่อใช้ในการติดต่อผ่านทาง ICQ ซึ่งในอนาคตคาดว่านอกจากจะต้องมีอีเมล์แอดเดรสของตนเองแล้วก็อาจต้องมีหมายเลข ICQ ที่เป็นเสมือนหมายเลขอ้างอิงอีกอย่างหนึ่งของบุคคลที่ใช้งานอินเตอร์เน็ตอีกด้วย

ก.1.1 ที่มาของ ICQ

บริการ ICQ เริ่มต้นมาจากแนวคิดของบริการชื่อ Buddy List จากบริษัท America Online หรือ AOL ซึ่งเป็นบริษัท ISP รายใหญ่ที่สุดของอเมริกา โดยพัฒนาขึ้นเพื่อให้บริการแก่สมาชิกของ AOL ได้สนทนาและติดต่อสื่อสารกันได้ ซึ่งต่อมาได้มีบริษัทต่างๆ ที่ทำธุรกิจนอินเตอร์เน็ตเช่น Yahoo, Netscape และอื่นๆ ได้พัฒนาโปรแกรมในลักษณะของ Buddy List ขึ้นมาอีกมากมายแต่ก็ไม่ได้รับความนิยมมากนัก

ข้อจำกัดของ Buddy List ของ AOL (หรือที่ปัจจุบันได้พัฒนาไปเป็นโปรแกรม AOL Instant Messaging หรือ AIM) ก็คือสามารถใช้งานได้เฉพาะกลุ่มของสมาชิก AOL เท่านั้นส่วนผู้ใช้ที่อยู่ในประเทศอื่นๆ เช่นในประเทศไทยจะไม่มีโอกาสใช้บริการนี้ ต่อมาบริษัท Mirabilis อยู่ในประเทศอิสราเอลได้พัฒนาโปรแกรม ICQ ขึ้นมาในปี 1996 และเผยแพร่จนได้รับความนิยมในกลุ่มผู้ใช้อินเตอร์เน็ตทั่วไป บริษัท Mirabilis มีผู้ร่วมก่อตั้งทั้งหมด 5 คนคือ Sefi Vigiser, Yair Golfinger, Arik Vardi, Arik Yossi และ Ammon Amir ซึ่งต่อมาในเดือนมิถุนายนปี 1998 กลุ่มผู้ก่อตั้ง Mirabilis ได้ขายหุ้นให้แก่ AOL ดังนั้นปัจจุบัน Mirabilis จึงได้กลายเป็นส่วนหนึ่งของ AOL ไป (โดยเปลี่ยนชื่อไปเป็น ICQ Inc.) และจัดว่าเป็นโปรแกรมที่มีผู้ใช้มากที่สุดในโลกตัวหนึ่ง โดยมีผู้ลงทะเบียนไว้แล้วถึง 50 ล้านคน (ถึงแม้จะตัดพวกที่ลงทะเบียนซ้ำซ้อนหลายๆ ออกไปบ้างแล้ว ก็ยังถือว่ามากอยู่ดี) และในขณะหนึ่งๆ จะมีผู้ใช้งาน ICQ พร้อมกันทั่วโลกไม่ต่ำกว่าสิบล้านคน

ก.1.2 การติดตั้งและการใช้งานโปรแกรม ICQ

โปรแกรม ICQ ที่ติดตั้งในส่วนไคลเอนต์จะเป็นโปรแกรมที่แจกฟรี ผู้ใช้งานสามารถดาวน์โหลดได้ฟรีที่เว็บไซท์ www.mirabilis.com ไฟล์โปรแกรมปัจจุบันซึ่งมีด้วยกันหลายเวอร์ชัน ปัจจุบันเวอร์ชัน ICQ99b ซึ่งสามารถใช้ร่วมกับระบบปฏิบัติการหลายแบบเช่น Windows95/98, Windows NT หรือ Linux เป็นต้น เมื่อทำการดาวน์โหลดโปรแกรม ICQ เรียบร้อยแล้วก็จะเริ่มเข้าสู่ขั้นตอนการติดตั้งโปรแกรม icq99b.exe (ซึ่งชื่อโปรแกรมอาจมีการเปลี่ยนตามเวอร์ชันของ ICQ) แล้วดำเนินการติดตั้งเช่นเดียวกับโปรแกรมอื่นๆ

ต่อจากขั้นตอนการติดตั้งโปรแกรม ICQ ก็คือผู้ใช้จะต้องทำการลงทะเบียนก่อนเริ่มการใช้งาน โดยคลิกที่โปรแกรม icq.exe หรือคลิกที่ปุ่ม Start > Programs > ICQ > ICQ ซึ่งถ้าเป็นการใช้งานครั้งแรกจะเป็นการติดต่อไปยังเซิร์ฟเวอร์ Mirabilis เพื่อขอลงทะเบียนจากนั้นจึงกรอกรายละเอียดต่างๆ ที่ใช้ในการลงทะเบียน ซึ่งประกอบด้วยรายละเอียดต่างๆ (เช่น ชื่อ-นามสกุล, วันเดือนปีเกิด, เพศ, อาชีพ, ภาษา, และประเทศที่อยู่ เป็นต้น) ส่วนสำคัญที่ต้องระมัดระวังในเรื่องการลงทะเบียนก็คือการกำหนด password ซึ่งจะใช้เมื่อต้องการเปลี่ยนแปลงรายละเอียดต่างๆ ของ ICQ โดยจะต้องจำ password ที่กำหนดไว้ให้ดีเพื่อให้สามารถนำมาใช้ได้เมื่อโปรแกรมต้องการ เมื่อลงทะเบียน ICQ เสร็จเรียบร้อยแล้ว ผู้ใช้จะได้หมายเลข ICQ ประจำตัวเพื่อใช้ในการติดต่อกับคนอื่น ซึ่งจะไม่ซ้ำกับของคนอื่นโดยเรียกว่า UIN เป็นหมายเลขที่ใช้ในการติดต่อกันซึ่งต้องใช้ควบคู่กับ password ตลอดการใช้บริการ ICQ

เมื่อเรียกโปรแกรม ICQ ขึ้นมาทำงานโปรแกรมจะทำการตรวจสอบว่าขณะนี้เครื่องของเราติดต่ออยู่ในเครือข่ายอินเตอร์เน็ตหรือไม่ และเมื่อตรวจสอบเสร็จว่าติดต่ออยู่ก็จะเปลี่ยนสัญลักษณ์ที่ด้านขวาของทาสก์บาร์เป็นรูปดอกทานตะวันของโปรแกรม ICQ จากนั้นเมื่อ user ออนไลน์แต่ละครั้งโปรแกรมจะทำการส่งข้อมูลหมายเลข IP (IP address) ของผู้ใช้นั้นออกไปให้กับเซิร์ฟเวอร์กลางรับทราบ เพื่อเป็นการประกาศว่า "ฉันมาแล้วนะ" เผื่อว่ามีใครต้องการติดต่อกับเรา หากใช้ ICQ อยู่ก็จะสามารถสอบถาม IP address จากเซิร์ฟเวอร์กลางแล้วใช้ IP address นั้นในการส่งข้อมูลติดต่อกับเราโดยตรงทันที ซึ่งจุดนี้เองเป็นข้อดีที่ทำให้ผู้ใช้ติดต่อกันได้โดยตรงทำให้ได้ความเร็วที่ทันใจ การติดต่อกันมี 2 รูปแบบด้วยกันคือการติดต่อกันโดยตรง (ใช้ TCP) และการติดต่อผ่านทางเซิร์ฟเวอร์ (ใช้ UDP)

ข้อเสียของโปรแกรมประเภท Instant Messaging นี้คือเรื่องของความปลอดภัยเพราะโดยปกติโปรแกรมอื่นจะไม่มีทางรู้ถึง IP address ของผู้ใช้อินเตอร์เน็ตรายอื่นๆ ได้ง่ายๆ จะมีแต่โปรแกรมบนเซิร์ฟเวอร์เท่านั้นที่รับรู้ได้และจะไม่มีการนำ IP address ไปแจกจ่ายใครง่ายๆ แบบนี้แต่เนื่องจากวิธีการของ ICQ จะต้องการทำการติดต่อโดยตรงจึงต้องมีกลไกที่จะให้อีกฝ่ายรับรู้ IP address ของผู้ที่จะติดต่อด้วยจุดนี้เองนับเป็นข้อเสียของ ICQ ถึงแม้จะมีความพยายามที่จะปิด IP address ของผู้ติดต่อด้วยก็ตาม ก็ยังมีผู้ไม่ประสงค์ดีทำการดักจับข้อมูลระหว่างการส่ง-รับไปอ่านได้ทำให้ไม่มีความปลอดภัยเท่าที่ควร

ดูข้อมูลเพิ่มเติมที่ http://www.thaiicq.com

กิตติกรรมประกาศ

โครงงานนี้ได้รับการส่งเสริมและสนับสนุนจากหลายฝ่ายจนกระทั่งประสบผลสำเร็จจนได้ ดังนั้นผู้พัฒนาใคร่ขอขอบพระคุณไว้ ณ. โอกาสนี้

บิดา-มารดา เป็นผู้มีพระคุณมากได้ให้กำเนิดและเลี้ยงดู ส่งเสริมให้ได้รับและได้ทำในสิ่งที่ดีมาโดยตลอด

อาจารย์ธนา หงษ์สุวรรณ เป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์และเป็นผู้ก่อให้เกิดปริญญานิพนธ์นี้

อาจารย์อัครเดช วัชระภูพงษ์ เป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์และช่วยให้คำแนะนำและให้คำปรึกษา รวมถึงช่วยแก้ปัญหาต่างๆ อย่างต่อเนื่องตลอดโครงงาน

ห้อง ISAG คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้พื้นที่อำนวยความสะดวกและอุปกรณ์เครือข่ายที่ใช้ในการพัฒนาโครงงาน

เพื่อนๆ รุ่น 6 ทั้งหลายที่ช่วยสร้างบรรยากาศและทัศนคติที่ดีตลอดมา

ตัวผู้พัฒนาเองที่มีพลังและความพยายามในการพัฒนาโปรแกรมจนกระทั่งสำเร็จดังที่หวังไว้

บรรณานุกรม

หนังสืออ้างอิง

[1] David Flanagan : "*JAVA in a Nutshell, Second Edition*", O'Reilly & Associates, Inc. United State of America 1997.

[2] Elliotte Rustly Harold : "*Java Network Programming*", O'Reilly & Associates, Inc. United State of America 1997.

[3] Gary Cornell, Cay S. Horstmann (1996) : "*Core JAVA*", Sun Microsystem, Inc. 1996

[4] Merlin and Conrad Hughes, Michael Shoffner, Maria Winslow :"*Java Network Programming*", Manning Publications Co. 1997

[5] Patrick Chan, Rosanna Lee (1996) : "*The Java Class Libraries An Annotated Reference*", An imprint of Addison Wesley Longman, Inc. 1996.

[6] กิตติ ภักดีวัฒนะกุล : "*Java ฉบับโปรแกรมเมอร์*", บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด 1999.

[7] ดร.วีระศักดิ์ ซึงถาวร : "*Fundamental of JAVA Programming Volume 1*", Sum Publishing Department, Sum System Company Limited. 1998.

[8] ดร.วีระศักดิ์ ซึงถาวร : "*Fundamental of JAVA Programming Volume 2*", Se-Education Public Company Limited. 2000.

[9] Java™ Development Kit "*JDK 1.1.x Documentation*", Copyright © 1996, 1997, 1998 Sun Microsystems, Inc.

เว็บไซต์อ้างอิง

[1] http://www.java.sun.com

[2] http://www.javasoft.com

[3] http://www.student.nada.kth.se/~d95-mih/icq/

[4] http://www.thaiicq.com