ระบบรับส่งสารด่วนแบบปลอดภัย

Secure Instant Messaging System

นายพงศธร มัตยะสุวรรณ

นายพรายพล พุกกะพันธุ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

ระบบรับส่งสารด่วนแบบปลอดภัย

Secure Instant Messaging System

โดย

นายพงศธร มัตยะสุวรรณ

นายพรายพล พุกกะพันธุ์

อาจารย์ที่ปรึกษา

อาจารย์ธนา หงษ์สุวรรณ

อาจารย์อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรับส่งสารด่วนแบบปลอดภัย

Secure Instant Messaging System

ผู้จัดทำ

1. นายพงศธร มัตยะสุวรรณ รหัสนักศึกษา 42010212
2. นายพรายพล พุกกะพันธุ์ รหัสนักศึกษา 42010221

..... อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

..... อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

ระบบรับส่งสารด่วนแบบปลอดภัย

นายพงศธร มัตยะสุวรรณ 42010212
นายพรายพล พุกกะพันธุ์ 42010221
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

โปรแกรมรับส่งสารด่วนตัวอย่างเช่น ICQ และ AIM สามารถตอบสนองความต้องการในการสื่อสารระหว่างบุคคลได้อย่างรวดเร็วจนเป็นที่นิยมทั่วไป แม้การสื่อสารส่วนใหญ่เป็นเพื่อสันทนาการ หากแต่โปรแกรมรับส่งสารด่วนก็มีคุณค่าเชิงธุรกิจไม่น้อย อาจถือได้ว่าเป็นช่องทางการสื่อสารใหม่บนอินเทอร์เน็ตควบคู่กับจดหมายอิเล็กทรอนิกส์

แต่โปรแกรมรับส่งสารด่วนในปัจจุบันยังมิได้คำนึงถึงระบบรักษาความปลอดภัยที่ดีเพียงพอ ไม่ว่าเรื่องข้อความที่ส่งออกไปโดยไม่เข้ารหัส การแอบอ้างส่งข้อความและการรักษาความเป็นส่วนตัวของผู้ใช้งาน ในปีที่ผ่าน ๆ มาทางห้อง ISAG จึงได้ทำโครงงานเพื่อแก้ไขโปรแกรมฝั่งผู้ใช้ (โปรแกรม IsagQ BETA) ให้มีการเข้ารหัสข้อความที่ได้ส่งออกและถอดรหัสปลายทาง แต่โปรแกรมดังกล่าวไม่สามารถใช้งานได้ในปัจจุบัน เนื่องจากมีการใช้งานโพรโตคอลเวอร์ชันเก่า

โครงงานนี้จึงได้จัดทำโปรแกรมฝั่งผู้ใช้ขึ้นใหม่ โดยให้โปรแกรมสามารถใช้งานกับโพรโตคอลเวอร์ชันปัจจุบันได้ พร้อมทั้งสร้างหน้าจอในการใช้งานขึ้นใหม่และปรับปรุงการเข้าและถอดรหัสข้อมูลให้มีความปลอดภัยมากขึ้น

Secure Instant Messaging System

Pongsatorn Matthayasuwan
Praipol Pookkapund
Thana Hongsuwan Advisor
Akkradach Watcharapupong Advisor

ABSTRACT

Instant-messaging programs, such as ICQ and AIM can response requirement of human communication and be popular quickly. Although most of human use it for entertainment, but instant-messaging programs have helpful for business which be a new way of internet communication and work with e-mail.

But nowaday, instant-messaging programs is not considered about security system, example no sending message encryption, no sending message authentication and no authorization. In the past, ISAG offered the project for develop instant-messaging program (IsagQ BETA program) for encrypt sending message and decrypt at destination, but it can not work in present because this program is developed in old ICQ protocol's version.

This project offers a new instant messaging program which can use with the lastest ICQ protocol's version, develop the graphic user interface and more secure encryption.

กิตติกรรมประกาศ

โครงงานนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากบุคคลหลายๆ ฝ่ายด้วยกัน บุคคลสองท่านแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้สำเร็จลงได้ก็คือ อาจารย์ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ก่อให้เกิดโครงงานนี้ พร้อมทั้งให้คำแนะนำและให้คำปรึกษา รวมถึงช่วยแก้ไขปัญหาต่างๆ อย่างต่อเนื่องตลอดโครงงาน ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณห้องปฏิบัติการ ISAG ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ได้สนับสนุนสถานที่และอุปกรณ์เครือข่ายสำหรับใช้ในการพัฒนาโครงงาน

ขอขอบคุณคณาจารย์ในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้กับผู้จัดทำ

ขอขอบคุณเพื่อนห้อง D และห้อง P ที่คอยช่วยเหลือ ให้กำลังใจและสร้างบรรยากาศที่ดีตลอดมา

สุดท้ายนี้ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ผู้จัดทำมีวันนี้ก็คือ บิดา-มารดา อันเป็นที่เคารพรัก ซึ่งได้ให้กำเนิดและเลี้ยงดูมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ คอยให้กำลังใจและเอาใจใส่เสมอมาอันหาที่เปรียบมิได้ ผู้จัดทำขอระลึกในพระคุณและขอกราบขอพระคุณมา ณ ที่นี้

พงศธร มัตยะสุวรรณ

พรายพล พุกกะพันธุ์

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในปัจจุบันการติดต่อสื่อสารได้ก้าวไกลและมีหลากหลายรูปแบบ เพื่อตอบสนองให้สามารถใช้งานได้ตามตรงความต้องการ โปรแกรมรับส่งสารด่วนตัวอย่างเช่น ICQ และ AIM ถือเป็นทางเลือกหนึ่งของการสื่อสารที่สามารถตอบสนองความต้องการของผู้ใช้งานได้หลากหลายประเภท สามารถช่วยให้ผู้ใช้งานทำการสนทนาโต้ตอบกันได้อย่างรวดเร็วและประหยัดค่าใช้จ่าย เนื่องจากสามารถสนทนาได้หลายคนและไม่มีค่าบริการทางไกล ซึ่งทำให้โปรแกรมรับส่งสารด่วนได้รับความนิยมอย่างรวดเร็ว

ถึงแม้ว่าการใช้งานโปรแกรมรับส่งสารด่วนสำหรับสื่อสารทั่วไปจะได้รับความนิยมอย่างมาก แต่การใช้งานโปรแกรมรับส่งสารด่วนในองค์กรธุรกิจกลับไม่เป็นที่นิยม ทั้งๆ ที่หากนำโปรแกรมรับส่งสารด่วนมาใช้งานจะช่วยลดการใช้งานจดหมายอิเล็กทรอนิกส์และวอยส์เมล์ภายในองค์กรได้อย่างมาก ที่เป็นเช่นนี้เนื่องจากโปรแกรมรับส่งสารด่วนยังไม่มีความปลอดภัยในการรับส่งสารระหว่างกัน ซึ่งอาจทำให้บุคคลอื่นสามารถที่จะขโมยข้อความที่เป็นความลับภายในองค์กรได้

ด้วยเหตุนี้ ทางผู้จัดทำจึงได้เกิดแนวคิดที่จะปรับปรุงระบบรับส่งสารด่วนให้มีความปลอดภัยมากขึ้น ซึ่งมีโปรแกรมฝั่งไคลเอ็นต์เพื่อใช้เข้ารหัสข้อมูลสำหรับการส่งสารและถอดรหัสข้อมูลสำหรับการรับสาร เพื่อเพิ่มความปลอดภัยในการรับส่งสารให้มากขึ้น โดยใช้ภาษาจาวาสำหรับเขียนโปรแกรมรับส่งสารด่วนแบบปลอดภัยนี้

1.2 วัตถุประสงค์ของโครงงาน

1. สามารถสร้างโปรแกรมที่ใช้งานได้บนทุกแพลตฟอร์มตามคุณสมบัติของภาษาจาวา
2. สามารถเขียนโปรแกรมเพื่อเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต
3. ศึกษาการเข้า-ถอดรหัสข้อมูลเพื่อความปลอดภัยของข้อมูลที่ทำการการรับส่ง
4. ศึกษาการทำงานของระบบรับส่งสารด่วน และปรับปรุงระบบรับส่งสารด่วนให้มีความปลอดภัยมากขึ้น
5. สร้างโปรแกรมต้นแบบในฝั่งไคลเอ็นต์สำหรับรับส่งสารโดยตรงและรับส่งสารผ่านเซิร์ฟเวอร์

1.3 ขอบเขตของการพัฒนา

Input specification

- รับค่าการพิมพ์และคำสั่งการทำงานผ่านทางคีย์บอร์ดและเมาส์

Output specification

- แสดงผลออกทางจอภาพโดยเป็นกราฟิก

Function specification

- สามารถทำการรับส่งสารโดยเข้า-ถอดรหัสข้อมูลระหว่างผู้ใช้งานโปรแกรม IsagQ และผู้ใช้งานโปรแกรม IsagQ ได้
- สามารถทำการรับส่งสารโดยไม่มีการเข้า-ถอดรหัสข้อมูลระหว่างผู้ใช้งานโปรแกรม IsagQ และผู้ใช้งานโปรแกรม ICQ ได้
- มีการใส่ล็อกอินและรหัสผ่านของผู้ใช้งานสำหรับใช้งานโปรแกรม IsagQ
- แสดงสถานะของคู่สนทนาโดยแบ่งเป็น Secure สำหรับผู้ใช้ IsagQ ซึ่งทำการออนไลน์, Online สำหรับผู้ใช้ ICQ ซึ่งทำการออนไลน์และ Offline สำหรับผู้ใช้ที่ทำการออฟไลน์ได้
- ค้นหาผู้ใช้งานได้จากทางหมายเลขประจำตัวผู้ใช้งาน (UIN), ชื่อและอีเมล์แอดเดรส

1.4 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีต่างๆ ที่เกี่ยวข้องกับงานวิจัย โดยเริ่มจากการศึกษาการทำงานของโปรแกรมรับส่งสารด่วน คุณสมบัติของโปรแกรมรับส่งสารด่วนซึ่งได้รับความนิยมในปัจจุบันและศึกษาระบบการเข้ารหัสข้อมูล จากนั้นจึงทำการศึกษาโพรโตคอลของโปรแกรม ICQ ในเวอร์ชันปัจจุบันพร้อมทั้งทำการดักจับแพ็กเก็ตที่ทำการรับส่ง จากนั้นจึงได้นำความรู้ที่ได้ศึกษามาทั้งหมดทำการออกแบบและเขียนโปรแกรม IsagQ ฝั่งไคลเอ็นต์โดยเริ่มจากการเขียนโปรแกรมสำหรับเรียกใช้บริการต่างๆ กับเซิร์ฟเวอร์เช่น การขอล็อกอิน การเปลี่ยนแปลงสถานะการใช้งาน การค้นหาเพื่อนหรือคู่สนทนาและการขอรายละเอียดของผู้ใช้งานเป็นต้น จากนั้นจึงทำการเขียนโปรแกรมสำหรับรับส่งสารทั้งการรับส่งสารโดยตรงและการรับส่งสารผ่านทางเซิร์ฟเวอร์ให้กับผู้ใช้งานโปรแกรม ICQ จากนั้นทำการเขียนโปรแกรมเข้าและถอดรหัสข้อมูลสำหรับรับส่งสารโดยตรงอย่างปลอดภัยกับผู้ใช้งาน IsagQ หลังจากนั้นจึงทำการทดสอบการทำงานทั้งหมด ตรวจสอบข้อบกพร่องของระบบ เพื่อนำกลับมาแก้ไขให้ดีขึ้น

บทที่ 2
โปรแกรมรับส่งสารด่วน

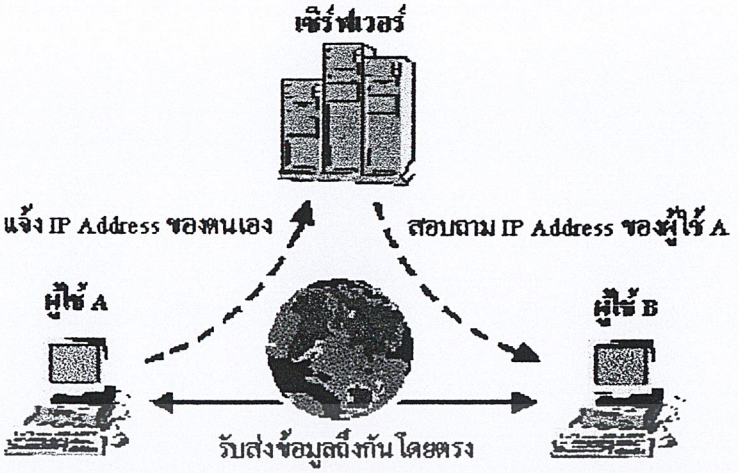
โปรแกรมรับส่งสารด่วนเป็นการรับส่งสารระหว่างผู้ใช้งาน 2 คนที่ทำการออนไลน์ในเวลาเดียวกัน โดยการรับส่งสารมีลักษณะการรับส่งแบบ peer to peer ซึ่งเป็นการรับส่งสารจากผู้ส่งไปยังผู้รับได้โดยตรง ทำให้ผู้ใช้งานสามารถทำการสนทนาระหว่างกันได้อย่างรวดเร็ว โดยในปัจจุบันโปรแกรมรับส่งสารได้รับความนิยมใช้งานเป็นจำนวนมาก และมีผู้ผลิตและพัฒนาโปรแกรมรับส่งสารด่วนมากขึ้น ทำให้โปรแกรมรับส่งสารด่วนมีความสามารถในการใช้งานและคุณสมบัติเสริมที่มากขึ้น สามารถตอบสนองความต้องการของผู้ใช้งานได้อย่างทั่วถึง

2.1 ความสำคัญของโปรแกรมรับส่งสารด่วน

ถึงแม้ว่าการใช้งานโปรแกรมรับส่งสารด่วนส่วนมากจะถูกใช้เพื่อสันทนาการ แต่ก็สามารถที่จะนำมาประยุกต์ใช้งานกับองค์กรทั่วไปได้ โดยสามารถนำโปรแกรมรับส่งสารด่วนมาใช้งานร่วมกับการใช้งานจากจดหมายอิเล็กทรอนิกส์ (E-mail) และวอยส์เมล์ (Voice mail) ซึ่งหากมีการใช้งานโปรแกรมรับส่งสารด่วนเกิดขึ้นในองค์กร ก็จะช่วยลดการใช้งานจดหมายอิเล็กทรอนิกส์และวอยส์เมล์ ซึ่งทำให้ลดเนื้อที่การเก็บข้อมูลจากจดหมายอิเล็กทรอนิกส์และวอยส์เมล์ลงด้วย

2.2 การทำงานในโปรแกรมรับส่งสารด่วน

ในระบบรับส่งสารด่วนจะประกอบด้วยส่วนสำคัญ 2 ส่วน ได้แก่ เซิร์ฟเวอร์ของโปรแกรมรับส่งสารด่วนและโปรแกรมรับส่งสารด่วนฝั่งไคลเอ็นต์สำหรับผู้ใช้งานทั่วไป โดยในการทำงานจะเริ่มจากเมื่อผู้ใช้ทำการออนไลน์ โปรแกรมรับส่งสารด่วนฝั่งไคลเอ็นต์ของผู้ใช้งานจะทำการติดต่อไปยังเซิร์ฟเวอร์ของโปรแกรมรับส่งสารด่วน เพื่อรับส่งค่ารายละเอียดต่าง ๆ เช่น แจ้ง IP Address และสถานะการใช้งานให้กับเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์ก็จะแจ้งรายชื่อผู้ใช้ที่ทำการออนไลน์ซึ่งอยู่ในคอนแทคลิสต์ (Contact list) และ IP Address ของผู้ใช้งานที่อยู่ในคอนแทคลิสต์กลับมาให้กับผู้ใช้งานที่ทำการออนไลน์ และในเวลาเดียวกันนั้นเซิร์ฟเวอร์จะส่งสถานะการใช้งานและ IP Address ของผู้ใช้งานที่ทำการออนไลน์ให้กับผู้ใช้งานที่มีรายชื่อผู้ใช้งานที่ออนไลน์คนนั้นอยู่ในคอนแทคลิสต์ด้วยเช่นกัน จากนั้นผู้ใช้งานทั้งสองฝั่งก็จะสามารถรับส่งสารถึงกันได้โดยตรงไม่ต้องผ่านเซิร์ฟเวอร์ของโปรแกรมรับส่งสารด่วนนั้น



รูปที่ 2.1 การติดต่อกับเซิร์ฟเวอร์เพื่อใช้งาน

อย่างไรก็ตามโปรแกรมรับส่งสารด่วนจะสามารถทำการรับส่งสารโดยตรงระหว่างผู้ใช้งานได้ต่อเมื่อผู้ใช้งานทั้งสองฝั่งนั้นทำการออนไลน์เท่านั้น หากทางฝั่งผู้รับทำการออฟไลน์จะไม่สามารถทำการส่งสารโดยตรงระหว่างผู้ใช้งานได้ ซึ่งในส่วนนี้จะขึ้นอยู่กับแต่ละผลิตภัณฑ์ที่ผู้ผลิตได้ออกแบบมาเช่น ใน ICQ หากผู้รับทำการออฟไลน์ ผู้ส่งสามารถทำการส่งสารได้โดยข้อมูลที่ส่งให้ผู้รับจะถูกเก็บไว้ที่เซิร์ฟเวอร์ของ ICQ เมื่อผู้รับทำการออนไลน์ในภายหลังเซิร์ฟเวอร์ของ ICQ จะส่งสารของผู้ใช้งานคนนั้นที่ได้รับขณะออฟไลน์กลับมาให้ ส่วนใน MSN Messenger และ Yahoo Messenger นั้นเมื่อผู้รับทำการออฟไลน์จะสามารถส่งสารได้โดยผ่านทางการใช้จดหมายอิเล็กทรอนิกส์แทน

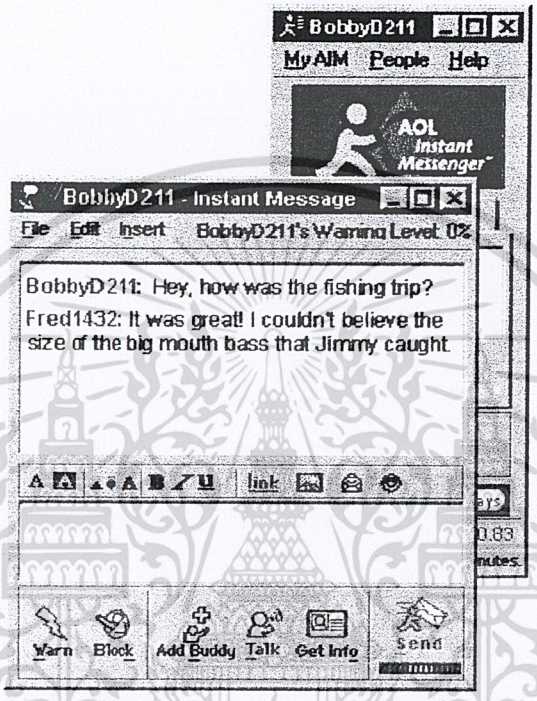
2.3 ตัวอย่างของโปรแกรมรับส่งสารด่วน

ในปัจจุบันมีผู้ผลิตและพัฒนาโปรแกรมรับส่งสารด่วนจำนวนมาก โดยเริ่มจากบริษัทอเมริกันออนไลน์ (America Online Incorporation) ซึ่งเป็นผู้ผลิตโปรแกรมรับส่งสารด่วนรายต้นๆ ได้ผลิตโปรแกรม AIM และ ICQ จนได้รับความนิยมอย่างแพร่หลาย ต่อมาทางบริษัทไมโครซอฟท์ (Microsoft Corporation) ก็ได้ผลิตโปรแกรม MSN Messenger เพื่อตอบสนองความต้องการให้กับผู้ใช้งานจดหมายอิเล็กทรอนิกส์ของทางไมโครซอฟท์ และในเวลาใกล้เคียงกันทาง Yahoo Incorporation ก็ผลิต Yahoo Messenger มาบริการผู้ใช้งานจดหมายอิเล็กทรอนิกส์ของทาง Yahoo เช่นเดียวกัน โปรแกรมรับส่งสารด่วนแต่ละโปรแกรมที่ถูกผลิตออกมานั้นต่างก็มีความสามารถในการใช้งานแตกต่างกันออกไป ตัวอย่างของโปรแกรมรับส่งสารด่วนที่มีชื่อเสียงและเป็นที่นิยมมีดังต่อไปนี้

2.3.1 AOL Instant Messenger

AOL Instant Messenger หรือเรียกสั้นๆ ว่า AIM เป็นโปรแกรมรับส่งสารด่วนที่ถูกผลิตโดยบริษัทอเมริกันออนไลน์ ซึ่งเป็นโปรแกรมรับส่งสารด่วนที่ได้ผลิตออกมาในยุคแรกๆ และเป็นโปรแกรมที่ได้รับความนิยมมากจนถึงปัจจุบันด้วยการมีจำนวนสมาชิกกว่า 150 ล้านราย โปรแกรม AIM มีความ

สามารถในการรับส่งข้อความแบบบุคคลต่อบุคคล, พีซีต่อพีซีและจากพีซีไปยังโทรศัพท์ แต่ยังไม่สนับสนุนการติดต่อผ่านทางเว็บแคม ส่วนการเชื่อมต่อโดยใช้พีดีเอ (PDAs:Personal Digital Assistance) นั้นหากเป็นสมาชิก AOL และใช้งานผ่านทางโปรแกรม AOL for palm จะไม่เสียค่าบริการเพิ่มเติม แต่การที่จะใช้โปรแกรมนี้ได้พีดีเอที่จะใช้งานจะต้องใช้ระบบปฏิบัติการปาล์มโอเอส 3.0 ขึ้นไป และมีโมเด็มอย่างเช่น โมเด็มไร้สาย Minstrel III หรือ OmniSky ติดตั้งอยู่

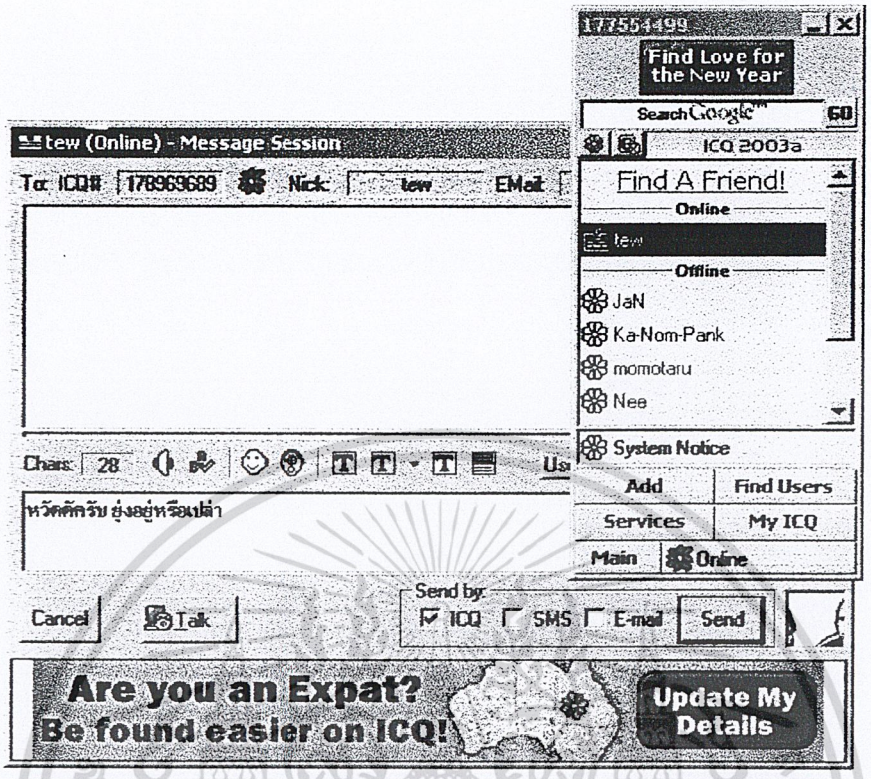


รูปที่ 2.2 โปรแกรม AOL Instant Messenger

จุดที่ทำให้โปรแกรม AIM เป็นที่เสียเปรียบเมื่อเปรียบเทียบกับโปรแกรมรับส่งสารด่วนอื่นๆ ก็คือไม่มีออปชันให้เลือกสำหรับตอบรับหรือปฏิเสธความต้องการของผู้ใช้คนอื่น ในการนำชื่อของผู้ใช้งานคนหนึ่งไปอยู่ในลิสต์ของผู้ใช้งานอีกคนหนึ่ง ทำให้ผู้ใช้สูญเสียความเป็นส่วนตัวไปพอสมควร ซึ่งจุดนี้น่าจะมีส่วนที่ทำให้ผู้ใช้งานหลายคนไม่เลือกใช้โปรแกรม AIM โดยเฉพาะอย่างยิ่งเมื่อมีโปรแกรมรับส่งสารด่วนโปรแกรมอื่นซึ่งสามารถให้ความเป็นส่วนตัวกับผู้ใช้ได้มากกว่านี้

2.3.2 ICQ

โปรแกรม ICQ เป็นอีกโปรแกรมหนึ่งที่ถูกผลิตขึ้นโดยบริษัทอเมริกันออนไลน์และถือเป็นโปรแกรมรับส่งสารด่วนที่สมบูรณ์แบบที่สุดโปรแกรมหนึ่งในปัจจุบัน โดยมีจำนวนผู้ใช้งานซึ่งได้ลงทะเบียนกว่า 300 ล้านราย โปรแกรม ICQ สามารถรองรับการสนทนาได้ทั้งแบบบุคคลต่อบุคคลและสนทนาร่วมกันหลายคน รองรับการติดต่อแบบพีซีกับพีซีและการติดต่อจากพีซีไปยังโทรศัพท์ นอกจากนี้โปรแกรม ICQ ยังสนับสนุนการทำงานกับพีดีเอที่ใช้ระบบปฏิบัติการปาล์มโอเอสและพ็อกเก็ตพีซีด้วย



รูปที่ 2.3 โปรแกรม ICQ

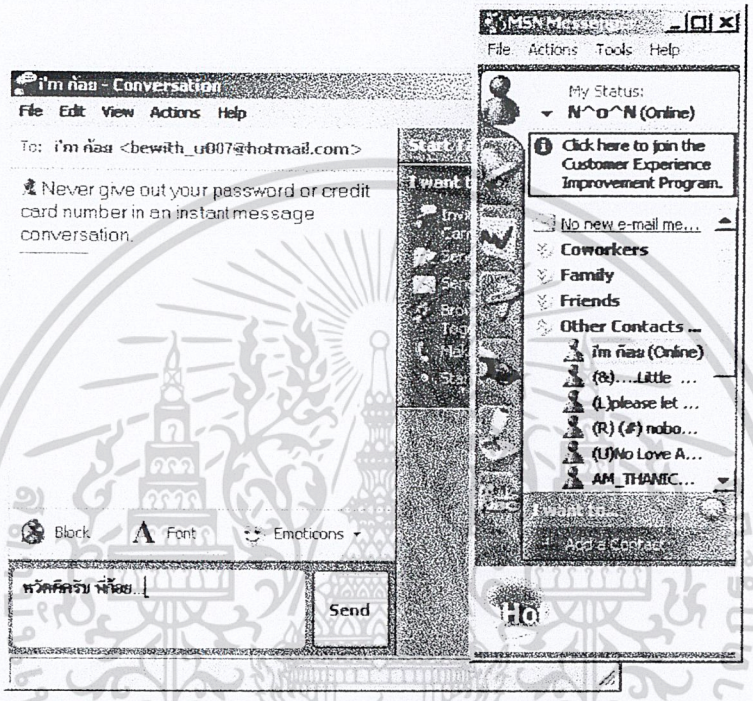
โปรแกรม ICQ แม้จะถูกผลิตขึ้นโดยบริษัทอเมริกันออนไลน์เช่นเดียวกับ AOL Instant Mesenger แต่กลับมีความเป็นส่วนตัวมากกว่า โดยหากผู้ใช้คนหนึ่งต้องการนำชื่อของผู้ใช้อีกคนหนึ่งจะต้องได้รับอนุญาตจากผู้ใช้คนนั้นก่อน หรือหากต้องการความเป็นส่วนตัวมากในเวลาหนึ่งผู้ใช้งานก็สามารถเปลี่ยนสถานะไม่ให้ผู้ใช้งานคนอื่นมองเห็นก็สามารถทำได้เช่นกัน นอกจากนี้โปรแกรม ICQ ยังมีคุณสมบัติสำหรับค้นหาคู่สนทนาได้อย่างยอดเยี่ยมและโดดเด่นที่สุดในโปรแกรมรับส่งสารด่วนทั้งหมด โดยผู้ใช้งานสามารถค้นหาคู่สนทนาได้จากชื่อ, นามสกุล, ชื่อเล่น, อีเมล์แอดเดรส, UIN, ประเทศ, กลุ่มอายุ, กลุ่มอาชีพและอื่นๆ อีกหลายหัวข้อ

สำหรับจุดบกพร่องของโปรแกรม ICQ นั้นเป็นเพียงจุดบกพร่องเพียงเล็กน้อยคือ โปรแกรม ICQ ยังไม่สนับสนุนการสนทนาผ่านทางเว็บแคมนอกเหนือจากนั้นแล้วโปรแกรม ICQ ก็ถือเป็นโปรแกรมรับส่งสารด่วนที่น่าใช้งานที่สุดโปรแกรมหนึ่ง

2.3.3 MSN Messenger

โปรแกรม MSN Messenger เป็นโปรแกรมที่ถูกผลิตขึ้นโดยไมโครซอฟท์ และมีผู้ใช้งานมากกว่า 200 ล้านคน โปรแกรม MSN Messenger จัดเป็นโปรแกรมรับส่งสารด่วนที่มีลูกเล่นมากที่สุดโปรแกรมหนึ่ง สามารถรองรับการสนทนาพื้นฐานได้ค่อนข้างครบถ้วนไม่ว่าจะเป็นการสนทนาแบบบุคคลต่อบุคคล สนทนาแบบเป็นกลุ่มและสนทนาด้วยเสียง แต่ยังไม่สามารถทำการสนทนาผ่านทางเว็บแคมได้เช่นเดียว

กับโปรแกรม ICQ นอกจากนี้โปรแกรม MSN Messenger ยังสามารถทำการส่งข้อความไปยังผู้ใช้งาน MSN Mobile และยังมี Session Initiation Protocol เพื่อใช้สนับสนุนการทำงานกับ Handheld ซึ่งใช้ในวินโดวส์ด้วย อีกคุณสมบัติหนึ่งที่ถือเป็นหัวใจของโปรแกรม MSN Messenger คือการรวมบริการแจ้งเตือนของ Microsoft's .NET ทำให้สามารถแจ้งเตือนผู้ใช้งานถึงเหตุการณ์ต่างๆ ที่เกิดขึ้นเช่น การเปลี่ยนแปลงของราคาหุ้นหรือสถานภาพการประมูลที่ eBay เป็นต้น



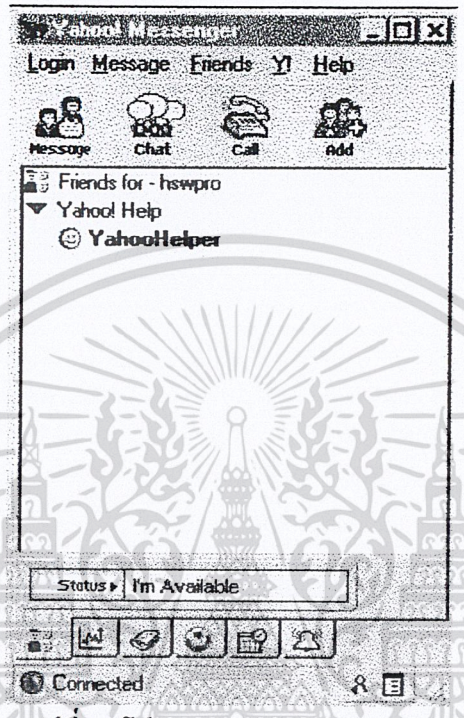
รูปที่ 2.4 โปรแกรม MSN Messenger

ถึงแม้โปรแกรม MSN Messenger จะมีลูกเล่นที่มากแต่ก็มีจุดบกพร่องมากเช่นกันคือ โปรแกรม MSN Messenger ขาดคุณสมบัติบางอย่างที่ควรจะมีเช่น การบันทึกข้อความการสนทนาของผู้ใช้งานเพื่อใช้สำหรับอ่านข้อความที่ได้สนทนาย้อนหลัง การที่ผู้ใช้งานไม่สามารถทำการส่งข้อความไปให้กับเพื่อนหรือคู่สนทนาซึ่งไม่ได้ทำการออนไลน์ โดยในส่วนนี้ผู้ใช้จะต้องทำการส่งจดหมายอิเล็กทรอนิกส์ไปยังคู่สนทนาแทนเป็นต้น

2.3.4 Yahoo Messenger

โปรแกรม Yahoo Messenger จัดเป็นโปรแกรมรับส่งสารด่วนที่มีประสิทธิภาพมากที่สุดตัวหนึ่ง ซึ่งผลิตโดย Yahoo Incorporation โปรแกรม Yahoo Messenger สามารถรองรับคุณสมบัติการสนทนาพื้นฐานไปจนถึงขั้นสูงได้อย่างครบถ้วนไม่ว่าจะเป็นการสนทนาแบบบุคคลต่อบุคคล การสนทนาด้วยเสียง การติดต่อจากพีซีไปยังโทรศัพท์และการสนทนาโดยใช้เว็บแคม สำหรับความเป็นส่วนตัวในโปรแกรม Yahoo Messenger นั้นมีออปชันที่มากพอสมควรเช่น กำหนดให้ผู้ใช้งานคนอื่นที่ต้องการนำชื่อผู้ใช้งาน

คนหนึ่งไปไว้ในลิสต์ต้องทำการขออนุญาตก่อน มีการกำหนดปฏิเสธผู้ใช้งานทุกคนหรือเฉพาะผู้ใช้งานที่ไม่ได้อยู่ในลิสต์เป็นต้น นอกจากนี้โปรแกรม Yahoo Messenger ยังจัดเป็นโปรแกรมรับส่งสารด่วนที่ครอบคลุมการทำงานกับ Handheld ที่มากที่สุดไม่ว่าจะเป็นพีดีเอที่ใช้ระบบปฏิบัติการปาล์มโอเอสหรือวินโดวส์ซีอีเวอร์ชัน 2.11 ขึ้นไป



รูปที่ 2.5 โปรแกรม Yahoo Messenger

สำหรับข้อบกพร่องในโปรแกรม Yahoo Messenger มีอยู่บางจุดเช่น อินเทอร์เฟซไม่เอื้ออำนวยต่อการเข้าถึงรายงายข่าว, หุ้น, พยากรณ์อากาศและอื่นๆ โดยสามารถเข้าถึงได้ค่อนข้างยาก นอกจากนี้การค้นหาผู้ใช้งานคนอื่นทำได้ไม่สะดวกเนื่องจากออปชันที่ใช้ในการค้นหามีค่อนข้างจำกัด หากไม่ทราบอีเมล์แอดเดรสของผู้ใช้งานที่ต้องการค้นหาก็เป็นการยากที่จะสามารถค้นหาผู้ใช้งานคนนั้นได้

2.4 ทิศทางของโปรแกรมรับส่งสารด่วน

ในอดีตผู้ผลิตต้องการให้โปรแกรมรับส่งสารด่วนสามารถรับส่งข้อความได้โดยตรง เพื่อให้ผู้ใช้งานสามารถสนทนากันได้อย่างรวดเร็วและสะดวกกว่าการใช้จดหมายอิเล็กทรอนิกส์ โดยนักวิเคราะห์เปรียบเทียบโปรแกรมรับส่งสารด่วนว่าเป็น "Real-time Email" ซึ่งการใช้โปรแกรมรับส่งสารด่วนทำให้การสนทนาเป็นไปอย่างต่อเนื่องกว่าการใช้จดหมายอิเล็กทรอนิกส์ และประหยัดค่าใช้จ่ายกว่าการสนทนาโดยการใช้โทรศัพท์ เนื่องจากสามารถสนทนาในเวลาเดียวกันได้หลายคนและไม่มีการกำหนดอัตราค่าบริการระหว่างพื้นที่ ทำให้โปรแกรมรับส่งสารด่วนเป็นที่นิยมได้อย่างรวดเร็ว

เมื่อโปรแกรมรับส่งสารด่วนเป็นที่นิยมมากขึ้น ทำให้มีผู้ผลิตเพิ่มมากขึ้นเช่นกันและมีความสามารถที่เพิ่มมากขึ้นรองรับความต้องการของผู้ใช้งานได้อย่างครบถ้วน โดยโปรแกรมรับส่งสารด่วนไม่

ได้เป็นเพียงโปรแกรมที่ใช้สำหรับสนทนาด้วยการรับส่งข้อความเพียงเท่านั้น สามารถสนทนาด้วยเสียง สนทนาโดยผ่านทางเว็บแคมหรือสามารถสนทนารวมกันหลายคนเป็นต้น ทำให้ผู้ใช้งานโปรแกรมรับส่งสารด่วนเพิ่มขึ้นอย่างรวดเร็วและต่อเนื่อง

สำหรับการใช้งานในองค์กรธุรกิจนั้น จากการวิจัยพบว่าบริษัทไอบีเอ็มซึ่งเป็นองค์กรแรกๆ ที่นำโปรแกรมรับส่งสารด่วนไปใช้งานภายในองค์กร ตัวเลขอ้างอิงจากบริษัทประมาณการออกมาว่าพนักงานในบริษัทไอบีเอ็ม 100,000 คนแลกเปลี่ยนข้อความกันไปมา 1 - 2 ล้านข้อความในแต่ละวัน ซึ่งสามารถลดจำนวนการใช้จดหมายอิเล็กทรอนิกส์ภายในบริษัทได้ 30 - 40% และวอยส์เมล์ได้ 10 - 15% แต่การนำโปรแกรมรับส่งสารด่วนไปใช้ในองค์กรธุรกิจกลับไม่เป็นที่นิยมและใช้งานมาก เนื่องจากโปรแกรมรับส่งสารด่วนในปัจจุบันยังไม่มีความปลอดภัยที่ดีพอ ส่งผลให้บุคคลอื่นภายนอกองค์กรสามารถดักจับข้อมูลที่ทำการสนทนาภายในองค์กรได้ หากมีการแก้ปัญหาด้านความไม่ปลอดภัยในโปรแกรมรับส่งสารด่วนแล้ว เชื่อกันว่าจะมีองค์กรธุรกิจจำนวนมากหันมาใช้โปรแกรมรับส่งสารด่วนสำหรับสนทนาภายในองค์กร ซึ่งบริษัท Verisign และอเมริกันออนไลน์กำลังเตรียมทีมพัฒนาโปรแกรมรับส่งสารด่วนแบบเข้ารหัส โดยคาดว่าจะเริ่มนำออกมาใช้งานได้ในปี ค.ศ.2003

บทที่ 3
ภาษาจาวา

3.1 ประวัติของภาษาจาวา

เมื่อหลายปีก่อนคงมีหลายคนเชื่อว่าภาษาคอมพิวเตอร์สำหรับโปรแกรมเป็นสิ่งที่ตายแล้ว ซึ่งอาจมีสาเหตุมาจากความเชื่อว่าไม่มีอะไรที่ภาษาซีทำไม่ได้ และอาจต้องใช้เวลาอีกหลายปีที่คนส่วนใหญ่จะเข้าใจถึงคุณค่าของภาษาซีพลัสพลัส แต่เมื่อปี ค.ศ.1996 ภาษาจาวาก็มีชื่อขึ้นหน้าปกของวารสารคอมพิวเตอร์ชั้นนำเกือบทุกฉบับทั่วโลกและมีการเสนอข่าวออกทาง CNN อย่างต่อเนื่อง ซึ่งไม่เคยมีภาษาคอมพิวเตอร์ใดได้รับความสนใจมากเช่นนี้มาก่อน

ภาษาจาวาเริ่มเป็นที่สนใจมากยิ่งขึ้น เมื่อบริษัท ซัน ไมโครซิสเต็มส์ (Sun Microsystems) ประกาศให้เป็นภาษาสำหรับสร้างโปรแกรมเพื่อใช้งานอินเทอร์เน็ต โปรแกรมที่เขียนโดยภาษาจาวาอาจถูกสร้างขึ้นบนคอมพิวเตอร์เครื่องหนึ่ง แล้วนำไปทำงานบนเครื่องคอมพิวเตอร์ต่างระบบปฏิบัติการได้โดยไม่ต้องคอมไพล์โปรแกรมนั้นใหม่ ความสามารถนี้จะเปลี่ยนแปลงบทบาทของอินเทอร์เน็ตไปอย่างสิ้นเชิง เมื่อเรามีโปรแกรมที่สามารถทำงานบนเครื่องคอมพิวเตอร์ใดก็ได้ อินเทอร์เน็ตจะกลายเป็นหนึ่งเดียว ไม่จำเป็นต้องแบ่งแยกสิ่งที่ถูกส่งออกไปมาบนอินเทอร์เน็ตอีกต่อไป เว็บเพจทั้งหลายจะไม่ใช่เอกสารที่ถูกกระทำแต่จะกลายเป็นเอกสารที่สามารถทำงานได้ นอกไปจากนี้ภาษาจาวาจะส่งผลกระทบต่อทั้งผู้ผลิตและผู้ใช้คือ ถึงเวลาที่บริษัทผู้ผลิตซอฟท์แวร์จะต้องเปลี่ยนแปลงหรือยุบแผนกวินโดวส์, แมคอินทอชหรือยูนิกซ์ให้มารวมกัน เพื่อผลิตโปรแกรมที่ทำงานบนระบบปฏิบัติการใดก็ได้เพียงหนึ่งเดียว สำหรับผู้ใช้งานเมื่อเปิดเครื่องคอมพิวเตอร์เข้าสู่บราวเซอร์ เราสามารถทำการเรียกโปรแกรมที่ต้องการจากเซิร์ฟเวอร์ที่ให้บริการในอินเทอร์เน็ตมาใช้งาน รูปแบบการทำงานของเราก็จะไม่เหมือนเดิมอีกต่อไปคือไม่จำเป็นต้องมีฮาร์ดดิสก์หรือระบบปฏิบัติการอย่างวินโดวส์เลยก็ได้

ผลกระทบเหล่านี้ทำให้ภาษาจาวาเป็นที่น่าสนใจอย่างยิ่ง ในขณะที่บริษัทผู้ผลิตซอฟท์แวร์ทั้งหลายก็ให้การตอบรับภาษาจาวาในทิศทางที่ดีเกินความคาดหมาย ประกอบด้วยภาษาจาวามีประสิทธิภาพในการสร้างโปรแกรมอย่างยิ่ง จึงเป็นเครื่องมือที่ดีสำหรับการสร้างโปรแกรมด้วยคอมโพเน็นต์และวิชวลโปรแกรมมิ่ง เชื่อได้แน่ว่าในอนาคตอันใกล้นี้อุตสาหกรรมซอฟท์แวร์จะเปลี่ยนโฉมหน้าไปจากเดิม และการเรียนรู้ภาษาจาวาจะเป็นเรื่องที่ได้เปรียบและหลีกเลี่ยงไม่ได้

ในช่วงต้นทศวรรษ 1990 ตลาดเครื่องใช้ไฟฟ้าเช่น เครื่องซักผ้า, หม้อหุงข้าว, โทรทัศน์, ไมโครเวฟและอื่นๆ มีมูลค่าสูงกว่าตลาดคอมพิวเตอร์หลายเท่าตัว อีกทั้งระบบคอมพิวเตอร์ขนาดเล็กสำหรับควบคุมเครื่องใช้ไฟฟ้าเหล่านี้ก็ถูกพัฒนาขึ้นเรื่อยๆ บริษัท ซัน ไมโครซิสเต็มส์ ซึ่งประสบความสำเร็จในตลาดระบบเครือข่ายคอมพิวเตอร์อย่างมากในตอนนั้น เห็นว่าควรใช้ความได้เปรียบของบริษัทเร่งพัฒนาเทคโนโลยีเพื่อยึดครองตลาดเครื่องใช้ไฟฟ้าที่มีคอมพิวเตอร์ควบคุมไว้ก่อนคนอื่น จึงจัดทีม Green group ขึ้นในปี 1991 มี James Gosling เป็นหัวหน้าสำหรับพัฒนาระบบซอฟท์แวร์ควบคุมเครื่องใช้

ไฟฟ้าขนาดเล็ก ทีม Green group ได้สร้างเครื่องต้นแบบที่เรียกว่า Star 7 เป็นระบบรีโมทคอนโทรลขนาดมือถือ สามารถควบคุมการเคลื่อนไหวของวัตถุโดยใช้นิ้วสัมผัสบนแป้นแสดงผล ระบบนี้ถูกทดสอบใช้แสดงการควบคุมตัว The Dunk (ซึ่งต่อมาเป็นตัวนำโชคของภาษาจาวา) ใช้เคลื่อนที่ผ่านกลุ่มของวัตถุในมิติสมมติ

Green group ใช้ภาษาซีพลัสพลัสเขียนโปรแกรมของ Star 7 ผลที่ได้เป็นโปรแกรมที่มักจะทำงานได้ผิดพลาดและล้มเหลวบ่อยๆ จนต้องสรุปว่าภาษาซีพลัสพลัสไม่เหมาะสำหรับงานแบบนี้ เพราะมีข้อจำกัดหลายอย่างนั่นคือ เครื่องใช้ไฟฟ้าขนาดเล็กจะมีหน่วยความจำน้อย ไม่เหมาะกับการเขียนโปรแกรมด้วยภาษาซีพลัสพลัสซึ่งมักจะมีขนาดใหญ่ อีกทั้งในเครื่องใช้ไฟฟ้าเหล่านี้ไม่มีระบบปฏิบัติการ โปรแกรมจึงไม่สามารถเรียกใช้บริการของระบบปฏิบัติการได้เหมือนบนเครื่องคอมพิวเตอร์ซึ่งมีระบบปฏิบัติการ ดังนั้นความสามารถของภาษาซีพลัสพลัสจึงถูกจำกัดไปอย่างมาก เราจึงต้องสร้างโปรแกรมสำหรับทำงานพื้นฐานเอง นอกจากนี้ภาษาซีพลัสพลัสยังเป็นภาษาที่ไม่ปลอดภัย เพราะยอมให้มีการใช้พอยน์เตอร์อย่างไม่จำกัด และละเลยการทำการตรวจสอบชนิดข้อมูลโปรแกรมจึงมักมีความผิดพลาดซ่อนอยู่เป็นจำนวนมาก

ปัญหาที่สำคัญกว่าคือ หน่วยประมวลผลที่ใช้ในงานควบคุมมีมากหลายเบอร์หลายยี่ห้อ และมีชุดคำสั่งที่แตกต่างกัน โปรแกรมที่ทำงานได้บนหน่วยประมวลผลรุ่นหนึ่งจะต้องถูกคอมไพล์ใหม่ จึงจะนำไปใช้งานบนหน่วยประมวลผลอีกรุ่นหนึ่งได้ ด้วยเหตุนี้พวกเขาจึงพัฒนาภาษาใหม่ชื่อโอ๊กให้เป็นภาษาที่ง่ายต่อการเรียนรู้แบะใช้งาน ไม่มีข้อผิดพลาดในตอนทำงานและเหมาะที่จะทำงานในระบบที่มีหน่วยความจำน้อย พวกเขาแน่ใจว่าระบบควบคุมจะมีขนาดใหญ่และซับซ้อนขึ้นเรื่อยๆ จึงให้โอ๊กเป็นภาษาเชิงวัตถุเพื่อให้ง่ายต่อการสร้างและดูแลระบบขนาดใหญ่ ปัญหาใหญ่ของการออกแบบภาษาโอ๊กอยู่ที่ความต้องการให้เป็นภาษาที่ทำงานบนหน่วยประมวลผลใดก็ได้ จึงนำเทคนิคการคอมไพล์โปรแกรมเป็นคำสั่งของหน่วยประมวลผลสมมติตัวหนึ่งแล้วสร้างอินเทอร์พรีเตอร์ของหน่วยประมวลผลสมมติตัวนั้นให้แก่หน่วยประมวลผลที่ทำงานที่จะทำงาน โปรแกรมนั้น ด้วยวิธีนี้โปรแกรมที่สร้างขึ้นจึงสามารถนำไปทำงานบนเครื่องที่มีหน่วยประมวลผลต่างรุ่นได้ ซึ่งเรียกคุณสมบัตินี้ว่า Platform independent

ผลงานของ Green group ให้ความหวังแก่บริษัท ซัน ไมโครซิสเต็มส์ อย่างมาก จึงก่อตั้งบริษัทลูกชื่อว่า FirstPerson Inc. ในปี ค.ศ.1992 เพื่อร่วมมือกับบริษัท Time Warner พัฒนาระบบ video on demand ที่มีอุปกรณ์ควบคุมเครื่องรับโทรทัศน์ที่สามารถติดต่อกับผู้ใช้ในลักษณะของ Interactive TV ภาษาโอ๊กถูกใช้เขียนโปรแกรมของ set-top box ซึ่งเป็นกล่องที่ใช้ต่อพ่วงกับโทรทัศน์เพื่อควบคุมและติดต่อกับผู้ใช้ แม้ระบบนี้จะถูกพัฒนาและสามารถใช้งานได้ แต่บริษัท Time Warner กลับหยุดโครงงานนี้ไป บริษัท ซัน ไมโครซิสเต็มส์ จึงพยายามหาลูกค้ารายใหม่ให้แก่เทคโนโลยีนี้ โดยนำไปทดสอบใช้งานอื่นๆ เช่น ระบบควบคุมบัตรเครดิต ระบบควบคุมเครื่องจักรในโรงงานอุตสาหกรรมและระบบควบคุมเครื่องซีดีรอม แต่ไม่สามารถหาลูกค้าได้

ในปี ค.ศ.1993 ไฮเปอร์เท็กซ์และบราวเซอร์เปลี่ยนแปลงอินเทอร์เน็ตไปอย่างมากและมีผู้ใช้เพิ่มมากขึ้นอย่างรวดเร็ว บริษัท ซัน ไมโครซิสเต็มส์ มองเห็นความจำเป็นที่ต้องมีภาษาสำหรับสำหรับสร้างโปรแกรมซึ่งสามารถทำงานบนคอมพิวเตอร์เครื่องใดก็ได้และเพิ่งนึกถึงคุณสมบัติ Platform independent

ของภาษาโอ๊ก จึงนำภาษาโอ๊กมาปรับปรุงใหม่ และทดลองสร้างเว็บบราวเซอร์ชื่อเว็บรันเนอร์ซึ่งสามารถทำงานโปรแกรมภาษาโอ๊กได้ เมื่อทดลองจนได้ผลพวกเขาทราบว่ามีสิ่งที่สำคัญอย่างยิ่งในมือแล้ว แต่โอ๊กเป็นชื่อทางการค้าที่มีผู้ใช้อยู่ก่อนจึงเปลี่ยนชื่อใหม่เป็นจาวาในช่วงต้นปี ค.ศ.1995 พร้อมกับเปลี่ยนชื่อเว็บรันเนอร์ใหม่เป็นฮอตจาวา (HotJava)

เมื่อเริ่มต้นจาวาทำงานบนระบบโซลาริสเท่านั้น แต่เพียงภายในฤดูร้อนของปี ค.ศ.1995 ก็พัฒนาให้ทำงานได้บนวินโดวส์ เอ็นที, วินโดวส์ 95 และลีนุกซ์พอถึงปลายปี ค.ศ.1995 บริษัทเน็ตสเคปก็สร้างเน็ตสเคป 2.0 ให้สามารถทำงานร่วมกับภาษาจาวาได้ หลังจากนั้นบริษัทไมโครซอฟท์และไอบีเอ็มก็ประกาศสนับสนุนภาษาจาวาด้วย จนกระทั่งช่วงปลายปี ค.ศ.1995 บริษัท ซัน ไมโครซิสเต็มส์ นำโปรแกรมชุดพัฒนาภาษาจาวา JDK (Java Development Kit) รุ่น 1.0 ขึ้นแจกจ่ายในอินเทอร์เน็ต

3.2 คุณสมบัติของภาษาจาวา

ลักษณะของจาวาเป็นการเขียนโปรแกรมอ้างอิงเชิงวัตถุ OOP (Object Oriented Programing) ซึ่งถูกออกแบบให้มีลักษณะดังต่อไปนี้

1. ภาษาจาวาเป็นภาษาที่ง่ายต่อการเรียนรู้และนำไปใช้งาน ซึ่งเราสามารถแยกพิจารณาออกตามได้หลายมุมมองดังต่อไปนี้
   - ภาษาจาวานำไวยากรณ์ภาษาส่วนใหญ่มาจากภาษาซีและซีพลัสพลัส ซึ่งทำให้ผู้ที่คุ้นเคยกับการเขียนโปรแกรมด้วยภาษาซีและซีพลัสพลัสสามารถเข้าใจภาษาจาวาได้ง่ายหรือใช้เวลาศึกษาไม่นาน
   - ภาษาจาวามีกลไกของภาษาไม่มากและไม่ซับซ้อน โดยได้ทำการตัดกลไกของภาษาซีและซีพลัสที่มีความซับซ้อนเช่น pointer, default argument, scope resolution, protected and private inheritance และ operator overloading แต่ในขณะเดียวกันก็เพิ่มความสามารถให้คอมไพเลอร์ของภาษาจาวา ทำให้ไม่มี preprocessor commands ดังนั้นจะไม่มี macros definitions, included files, conditional compilation และ header files และเมื่อจาวาเป็นภาษาเชิงวัตถุแล้วกลไกอย่างเช่น structures, union, bit fields และ enumerated types รวมทั้งการทำ typedef ก็ไม่มีความจำเป็นจึงถูกตัดออกไป ภาษาจาวาถูกออกแบบให้เป็นภาษาเชิงวัตถุได้รอบคอบกว่าภาษาซีพลัสพลัส ดังจะเห็นได้จากกลไกที่ทำให้เกิดปัญหาเช่น mutiple inheritance, copy constructor และกลไกที่อาจทำลายแบบแผนการเขียนโปรแกรมเชิงวัตถุที่ดีอย่างเช่น friend methods ก็ถูกตัดออกไปเช่นกัน
   - ภาษาจาวาประสบความสำเร็จอย่างมากในการใช้เทคนิคของภาษาเชิงวัตถุ ช่วยให้สร้างโปรแกรมที่ยุ่งยากให้ง่ายขึ้น ดังจะเห็นได้ว่าหากเป็นภาษาคอมพิวเตอร์ภาษาอื่นๆ การสร้างโปรแกรมที่เกี่ยวกับการสร้างกราฟริกในส่วนติดต่อกับผู้ใช้งาน (GUIs), multitasking network และ distributed objects ผู้เขียนโปรแกรมจะต้องมีความรู้ในภาษานั้นสูงจึงจะสามารถทำได้ หรือไม่เช่นนั้นก็ต้องใช้โปรแกรมประเภทวิชาร์ดอย่างเช่นวิชวลซีพลัสพลัส (Visual C++) ช่วยในการสร้างโปรแกรม แต่ภาษาจาวามีกลไกการเขียนโปรแกรมเชิงวัตถุที่ส่งเสริมการนำโปรแกรมที่สร้างไว้แล้วมาใช้งานใหม่ ซึ่งสามารถเปลี่ยนแปลงหรือเพิ่มเติมโปรแกรมบางส่วนให้เหมาะกับ

งานที่ต้องการ โดยไม่จำเป็นต้องทราบรายละเอียดของโปรแกรมเดิมทั้งหมด ทำให้เราสามารถสร้างโปรแกรมสำหรับงานที่ยุ่งยากขึ้นได้โดยใช้บางส่วนจากโปรแกรมที่มีผู้สร้างไว้แล้วโดยง่าย

- ในภาษาที่ถูกเรียกว่า typed language จะทำการตรวจสอบเกี่ยวกับการประกาศชนิดของตัวแปร เพื่อช่วยให้โปรแกรมทำงานโดยไม่มีความผิดพลาดเกี่ยวกับชนิดของตัวแปร แต่การเขียนโปรแกรมภาษาเช่นนี้มีข้อยุ่งยากเช่น ตัวอักษรบวกกับเลขจำนวนเต็มไม่ได้ เนื่องจากค่าที่จะนำมาทำการคำนวณหรือกำหนดค่าให้แก่กันจะต้องเป็นตัวแปรชนิดเดียวกัน ภาษาจาวาเองก็เป็น typed language แต่เพื่อให้ใช้งานง่ายจึงสร้างกฎเกณฑ์เกี่ยวกับ automatic type coercion ซึ่งเป็นการเปลี่ยนแปลงค่าระหว่างชนิดของตัวแปรที่แตกต่างกัน

2. โปรแกรมที่สร้างขึ้นด้วยภาษาจาวาจะไม่มีความผิดพลาดจากข้อบกพร่องของภาษา นั่นคือโปรแกรมจะต้องไม่ล้มเหลว (fail) ด้วยความผิดปกติเพียงเล็กน้อยที่ไม่เกี่ยวกับตรรกะของโปรแกรม คุณสมบัติของภาษาลักษณะนี้จะมีความคงทน (robust) ภาษาจาวาถูกออกแบบให้มีความคงทนตามลักษณะดังนี้

- ภาษาจาวาเน้นการใช้กลไก exception handling เพื่อให้โปรแกรมสามารถจัดการกับความผิดปกติบางอย่างที่เกิดขึ้นในขณะโปรแกรมทำงาน ซึ่งโปรแกรมจะทำงานต่อไปได้โดยไม่หยุดลง
- ภาษาจาวาตัดกลไกบางอย่างในภาษาซีและซีพลัสพลัสที่อาจทำให้เกิดความผิดพลาด หากใช้อย่างไม่ระมัดระวังเช่น global variable, variable length arguments และ goto statement นอกจากนี้ภาษาจาวายังได้ตัดการอ้างถึงแอดเดรสของตัวแปร และการใช้พอยน์เตอร์สำหรับอ่านหรือเขียนข้อมูลลงในหน่วยความจำโดยตรง
- ภาษาจาวาไม่มีกลไกคืนหน่วยความจำ (de-allocation) ที่ขอมาในขณะที่โปรแกรมทำงานอย่างที่มีใช้งานในภาษาซีและซีพลัสพลัสคือ free() และ delete() ภาษาจาวาอาศัย automatic garbage collector ทำหน้าที่เก็บหน่วยความจำที่ไม่สามารถอ้างถึงได้แล้วกลับไปใช้งานใหม่
- ภาษาจาวาเป็นภาษาประเภท Strongly typed หมายถึงภาษาที่เน้นความถูกต้องของชนิดข้อมูลที่ใช้ในโปรแกรม คอมไพเลอร์ของภาษาประเภทนี้จะทำการตรวจสอบว่าโปรแกรมการจัดการกับชนิดข้อมูลของตัวแปรถูกต้องหรือไม่เรียกการทำงานนี้ว่า type checking ซึ่งความผิดพลาดเกี่ยวกับชนิดข้อมูลทั้งหมดจะถูกปฏิเสธตั้งแต่ตอนคอมไพล์โปรแกรม จึงไม่มีความผิดพลาดเกี่ยวกับชนิดข้อมูลเกิดขึ้นในระหว่างที่โปรแกรมทำงาน นอกจากนั้นยังมีการตรวจสอบอีกว่าในระหว่างที่โปรแกรมทำงานมีการเปลี่ยนค่าตัวแปรในชนิดข้อมูลหนึ่งไปเป็นค่าในอีกชนิดข้อมูลหนึ่งถูกต้องหรือไม่ รวมทั้งการอ้างถึงสมาชิกในอาร์เรย์ (array) หรือสตริง (String) อยู่ในขอบเขตที่ถูกต้องหรือไม่

3. โปรแกรมภาษาจาวามักจะถูกส่งผ่านระบบเครือข่ายไปทำงานบนเครื่องคอมพิวเตอร์ของผู้อื่น ดังนั้นภาษาจาวาจึงต้องมีหลักประกันให้แก่ผู้รับโปรแกรมนั้นไปทำงาน เพื่อจะไม่ก่อให้เกิดอันตรายและความเสียหายต่อเครื่องหรือระบบของผู้ใช้นั้น ทำให้ภาษาจาวาจึงต้องมีข้อกำหนดหลายอย่างเพื่อให้

โปรแกรมไม่สามารถทำอันตรายหรือสร้างความเสียหายให้กับระบบที่รับโปรแกรมนั้นไปทำงาน คุณสมบัติในลักษณะคือความปลอดภัย (Security) แต่อย่างไรก็ตามไม่มีภาษาคอมพิวเตอร์ใดที่มีความปลอดภัยครบถ้วน ภาษาจาวาถูกจัดว่ามีความปลอดภัยในระดับสูงเท่านั้น เพราะถูกออกแบบมาเพื่อความปลอดภัยมากกว่าภาษาอื่น โดยมีการป้องกันในลักษณะดังนี้

- จากการที่ภาษาจาวาไม่ยอมให้มีการอ้างถึงค่าในหน่วยความจำผ่านทางพอยน์เตอร์ และจะทำการตรวจสอบว่ามีการอ้างถึงสมาชิกในอาร์เรย์อยู่ในขอบเขตหรือไม่ โปรแกรมจึงไม่สามารถเขียนหรืออ่านค่าในหน่วยความจำที่ไม่มีสิทธิ์อ้างถึง การเปลี่ยนแปลงโปรแกรมหรือค่าในหน่วยความจำเพื่อทำการสร้างโปรแกรมที่จะเป็นอันตรายต่อผู้อื่นจึงไม่สามารถทำได้
- อินเทอร์พรีเตอร์ของภาษาจาวามี byte-code verifier สำหรับทำหน้าที่ตรวจสอบโปรแกรมที่จะถูกทำงานว่ามีคำสั่งที่ผิดปกติหรือมีการทำงานที่ไม่สมควรหรือไม่ หากตรวจพบก็จะทำการปฏิเสธการทำงานในโปรแกรมนั้น
- ภาษาจาวาจะมีระบบรักษาความปลอดภัยที่เรียกว่า sandbox model นั่นคือถ้าหากเป็นโปรแกรมที่ถูกนำมาจากเครื่องอื่นผ่านทางระบบเครือข่ายจะถือว่าเป็นโปรแกรมที่ไม่น่าไว้วางใจ (un-trust codes) และจะถูกเก็บอยู่ในภาวะที่เรียกว่า sandbox โปรแกรมที่อยู่ใน sandbox จะมีข้อจำกัดในการทำงานหลายอย่าง ซึ่งถูกควบคุมโดย security manager เช่น ไม่สามารถอ่านหรือเขียนไฟล์ได้ เป็นต้น

4. คุณสมบัติสำคัญซึ่งถือเป็นจุดมุ่งหมายของการออกแบบภาษาจาวาคือ โปรแกรมต้องสามารถทำงานบนเครื่องต่างระบบกันได้ ซึ่งเราเรียกคุณสมบัติของภาษาในลักษณะนี้ว่า architecture neutral หรือ platform independent ทำให้โปรแกรมภาษาจาวาสามารถทำงานบนเครื่องคอมพิวเตอร์หรือระบบใดก็ได้โดยไม่เกิดความผิดพลาดและได้ผลลัพธ์ออกมาเหมือนกัน

3.3 การใช้ภาษาจาวาสำหรับสร้างและพัฒนาในงานด้านต่างๆ

จากความสามารถของภาษาจาวาดังที่กล่าวผ่านมาข้างต้น คงสามารถสรุปได้ว่าภาษาจาวาเป็นภาษาคอมพิวเตอร์สำหรับโปรแกรมที่มีความสามารถรอบด้าน อีกทั้งยังมีความยืดหยุ่นภายในตัวสูง ดังนั้นการที่จะนำเอาภาษาจาวามาสร้างและพัฒนางานด้านต่างๆ ให้มีประสิทธิภาพคงจะทำได้สะดวกขึ้น

ภาษาจาวาสามารถใช้เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้านไคลเอ็นต์ได้อย่างดีภาษาหนึ่ง โดยเฉพาะงานด้านกราฟิกและระบบติดต่อกับผู้ใช้ เพราะปัจจุบันโปรแกรมที่เขียนขึ้นจากภาษาจาวาจะทำงานที่ฝั่งไคลเอ็นต์เป็นส่วนมากจะเห็นได้จากจาวาแอพเพล็ตต่างๆ บนเว็บบราวเซอร์ แต่จะให้ดีและสมบูรณ์ยิ่งขึ้นต่อความเป็นไคลเอ็นต์ซอฟต์แวร์ที่ดีก็คือ การติดต่อระบบฐานข้อมูลจากไคลเอ็นต์ไปเป็นเซิร์ฟเวอร์ แม้ว่าในปัจจุบันจาวาจะสามารถทำได้แล้วก็ตามแต่ก็ต้องใช้เครื่องมือชนิดอื่นเป็นตัวช่วย เช่น การเชื่อมต่อกับฐานข้อมูลแบบ ODBC (Open Database Connectivity) ซึ่งการทำงานของทางฝั่งเซิร์ฟเวอร์ยังคงต้องใช้ภาษาคอมพิวเตอร์ชนิดอื่นช่วยเพื่อการติดต่อกับฐานข้อมูล

หากนำภาษาจาวาไปใช้สร้างโปรแกรมประยุกต์บนเซิร์ฟเวอร์นั้น ภาษาจาวาจำเป็นที่จะต้องได้รับการปรับปรุงการทำงานอยู่สามอย่างคือ การเชื่อมต่อกับฐานข้อมูลโดยตรง, การทำงานให้สอดคล้องกับอินพุทและเอาต์พุท และความเร็วของการรันต้องเพิ่มขึ้นเท่าๆ กับโปรแกรมประยุกต์ทั่วไป โดยระบบ JDBC (Java Database Connectivity) ได้เข้ามาแก้ปัญหาในส่วนของการติดต่อกับฐานข้อมูลโดยตรง และจากการร่วมมือกันของบริษัทซอฟต์แวร์ที่มีชื่อเสียงหลายบริษัทได้สร้างมาตรฐานการเชื่อมต่อกับฐานข้อมูลของจาวาขึ้น ทำให้การเชื่อมต่อฐานข้อมูลจากไคลเอ็นต์มายังเซิร์ฟเวอร์จะทำได้ง่ายขึ้นโดยใช้ภาษาจาวา

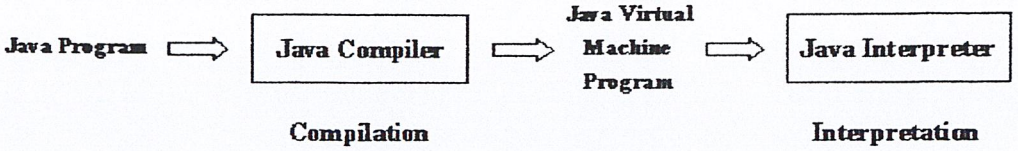
นอกจากนี้ภาษาจาวายังได้สร้างสภาพแวดล้อมเสมือน (Java Virtual Machine) ซึ่งจะอนุญาตให้เฉพาะจาวาแอพเพล็ตและจาวาแอพพลิเคชันซึ่งเป็นไบต์โค้ดเท่านั้นที่สามารถรันได้ โดยไบต์โค้ดจะปราศจากไวรัสหรือส่วนที่จะทำอันตรายต่อระบบ เนื่องจากสภาพแวดล้อมเสมือนจะจำกัดสิทธิการเข้าใช้ทรัพยากรของ ไบต์โค้ด และในปัจจุบันการรักษาความปลอดภัยอีกระบบหนึ่งก็ถูกนำมาปรับใช้คือ การเข้ารหัสข้อมูลเพื่อเพิ่มความปลอดภัยในการรับส่งข้อมูล

การใช้ภาษาจาวาประยุกต์สร้างและพัฒนางานยังสามารถแบ่งออกเป็นประเภทต่างๆ ตามลักษณะของกลุ่มงานได้ดังนี้

- งานด้านการศึกษา โดยจัดทำเป็นลักษณะสั่งการสอนคล้ายกับ CAI มีความสามารถในเชิงโต้ตอบระหว่างผู้ใช้งานกับคอมพิวเตอร์
- สร้างซอฟท์แวร์สำหรับเป็นเครื่องมือใช้พัฒนาโปรแกรม (Software Developer tool kits)
- สร้างแอพพลิเคชันทางธุรกิจให้มีความสามารถในการประมวลผลข้อมูลได้ด้วยตัวเอง
- พัฒนาเกมให้มีความเป็นมัลติ สามารถเล่นได้หลายๆ คนพร้อมกันผ่านทางระบบเครือข่ายหลากหลายพื้นที่ไม่ว่าจะเล่นด้วยกัน ณ ตำแหน่งใดทั่วทุกมุมโลก
- ตอบสนองงานทางด้านเว็บเพจทำให้งานสร้างเว็บเพจดูมีชีวิตชีวาและลูกเล่นมากขึ้น โดยการใช้ความสามารถทางด้านมัลติมีเดีย
- สร้างตัวจัดการบนอินเตอร์เน็ต มีความสามารถในการเรียกหาข้อมูลผ่านทางระบบเครือข่าย

3.4 สภาพแวดล้อมเสมือน (Java Virtual Machine)

สภาพแวดล้อมเสมือนเป็นหลักการสร้างคอมพิวเตอร์จำลองของภาษาจาวา โดยจะสมมติให้มีคอมพิวเตอร์อีกเครื่องหนึ่งขึ้นมา โดยคอมพิวเตอร์สมมติเครื่องนี้จะช่วยในการคอมไพล์โปรแกรมภาษาจาวาทุกโปรแกรม เมื่อต้องการให้โปรแกรมภาษาจาวาไปทำงานบนคอมพิวเตอร์จริงๆ เราก็เพียงแต่สร้างตัวอินเทอร์พรีเตอร์ของคอมพิวเตอร์จำลองตัวนี้บนเครื่องคอมพิวเตอร์เครื่องนั้น ภาษาจาวาทุกโปรแกรมก็จะสามารถทำงานบนระบบคอมพิวเตอร์นั้นได้ตามต้องการ ด้วยหลักการนี้ก็เป็นที่มาของคุณสมบัติของจาวาที่ไม่ขึ้นกับแพลตฟอร์มและฮาร์ดแวร์ใดๆ เราอาจจะเรียก Java Virtual Machine สั้นๆ ว่า JVM ก็ได้



รูปที่ 3.1 การทำงานด้วยคอมพิวเตอร์สมมติที่จำลองโดยจาวาอินเทอร์พรีเตอร์

สำหรับเหตุผลที่ต้องมี JVM นี้จริงๆ แล้วเป็นการกำหนดคำขึ้นมาสำหรับเป็นคำจำกัดความเฉพาะในความคิดเท่านั้น เพื่อให้นักพัฒนาจะได้ไม่ถูกบังคับให้ต้องสร้างตัวอินเทอร์พรีเตอร์ตามแนวทางใดแนวทางหนึ่งโดยเฉพาะ แต่อินเทอร์พรีเตอร์ที่สร้างตามข้อกำหนดดังกล่าวไม่ว่าจะอยู่บนแพลตฟอร์มใดจะสามารถรันโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาได้โดยให้ผลลัพธ์ออกมาเหมือนกัน โดยบริษัทจาวาซอฟท์ซึ่งเป็นบริษัทลูกของบริษัท ซัน ไมโครซิสเต็มส์ เป็นผู้กำหนดชุดคำสั่ง JVM รวมทั้งความหมายของแต่ละคำสั่ง ข้อกำหนดเหล่านี้ถือเป็นมาตรฐานของภาษาจาวาที่เผยแพร่ให้แก่บุคคลทั่วไป เพื่อให้ผู้พัฒนาสามารถสร้าง JVM ของตนขึ้นได้ไม่ว่าจะใช้วิธีทางฮาร์ดแวร์หรือซอฟท์แวร์ ภายใน JVM จะมีหน่วยประมวลผลสมมติที่เรียกว่าเวอร์ชัวล์โพรเซสเซอร์ (virtual processor) ซึ่งทำหน้าที่ประมวลผลคำสั่งของ JVM

ปัจจุบัน JVM ที่เป็นฮาร์ดแวร์ยังอยู่ในระหว่างการพัฒนา ดังนั้น JVM เกือบทั้งหมดที่ใช้งานกันอยู่ในขณะนี้ จึงเป็นโปรแกรมที่จำลองการทำงานของ JVM บนเครื่องคอมพิวเตอร์ทั่วไป โดยปกติแล้วเวอร์ชัวล์โพรเซสเซอร์ของ JVM ที่จำลองขึ้นบนคอมพิวเตอร์เครื่องหนึ่งจะแปลคำสั่งของ JVM เป็นคำสั่งของหน่วยประมวลผลในคอมพิวเตอร์เครื่องนั้นซึ่งเรียกว่าเนทีฟโค้ด (native code) แล้วให้หน่วยประมวลผลทำงานคำสั่งนั้น

ออบโค้ด (opcode) ของ JVM มีขนาด 1 ไบต์ทุกคำสั่ง เราจึงเรียกโปรแกรม JVM ว่าโปรแกรมไบต์โค้ด (byte code) จำนวนคำสั่งของ JVM มีได้สูงสุดเพียง 256 คำสั่ง เมื่อเปรียบเทียบกับหน่วยประมวลผลทั่วไปแล้วดูคล้ายกับว่าจำนวนคำสั่งของ JVM มีมากมาย ซึ่งที่จริงแล้วคำสั่งของ JVM แบ่งออกเป็นไม่กี่ประเภท โดยแต่ละประเภทจะทำหน้าที่คล้ายๆ กันเพียงแต่ทำกับโอเปอร์แรนด์ (operands) ต่างชนิดข้อมูลกันเท่านั้น

ชุดคำสั่ง JVM ถูกออกแบบมาเพื่อสนับสนุนการทำงานของโปรแกรมเชิงวัตถุจึงมีคำสั่งเกี่ยวกับการสร้างอินสแตนซ์ (instances) และการอ้างอิงสมาชิกในอินสแตนซ์ซึ่งไม่มีในหน่วยประมวลผลทั่วไป ภาษาจาวาเป็นภาษาที่เน้นความถูกต้องเกี่ยวกับชนิดของข้อมูล จึงมีคำสั่งสำหรับคำนวณชนิดของข้อมูลพื้นฐานแต่ละชนิด ดังเช่นในคำสั่ง iadd สำหรับบวกเลขจำนวนเต็มชนิด integer และคำสั่ง dadd สำหรับบวกเลขทศนิยมชนิด double เป็นต้น ในบางคำสั่งของ JVM จะเหมือนกับคำสั่งที่มีในหน่วยประมวลผลทั่วไป

JVM ถูกออกแบบให้สามารถจำลองได้บนหน่วยประมวลผลทั่วไป แต่หน่วยประมวลผลทั่วไปนั้นมีจำนวนรีจิสเตอร์ไม่เท่ากัน บางรุ่นมีรีจิสเตอร์ที่ทำหน้าที่พิเศษกว่ารุ่นอื่น ผู้ออกแบบจึงตัดปัญหานี้

โดยให้ JVM ไม่มีรีจิสเตอร์และทำการคำนวณทั้งหมดบนสแต็ก (stack) ชุดคำสั่งของ JVM จึงเป็น stacked operations ซึ่งกล่าวได้ว่า JVM เป็น stack machine

เวอร์ชัวล์โพรเซสเซอร์ใน JVM จะเปลี่ยนคำสั่งไบต์โค้ดไปเป็นเนทีฟโค้ดที่ทำหน้าที่เดียวกันแล้วทำงานเนทีฟโค้ดนั้น สังเกตว่าเนทีฟโค้ดนั้นอาจเป็นชุดคำสั่งของระบบปฏิบัติการที่ใช้หรืออาจเป็นไลบรารีมาตรฐาน (standard library) ที่สร้างขึ้นสำหรับหน่วยประมวลผลนั้น ทำให้ JVM หนึ่งอาจใช้งานได้ในระบบต่างๆ โดยเปลี่ยนแปลงแต่ไลบรารีมาตรฐานเท่านั้น

โปรแกรมภาษาจาวาเป็นโปรแกรมที่มีการทำงานโดยการอินเทอร์พรีเตชันคือ ต้องทำการแปลภาษาเพื่อให้สามารถใช้งานได้บนทุกๆ แพลตฟอร์ม ซึ่งโปรแกรมที่ทำงานโดยการอินเทอร์พรีเตชันย่อมทำงานได้ช้ากว่าโปรแกรมที่ทำงานโดยตรง ทำให้ปัจจุบันโปรแกรมภาษาจาวาจะทำงานช้ากว่าภาษาซีประมาณ 10 เท่า ได้มีการค้นคว้าเพื่อเพิ่มความเร็วของภาษาจาวาเช่น สร้างหน่วยประมวลผลที่สามารถทำงานคำสั่งของ JVM ได้โดยตรงและพัฒนาจาวาอินเทอร์พรีเตอร์ซึ่งไม่ได้ทำงานคำสั่ง JVM ทีละคำสั่งแต่จะเปลี่ยนโปรแกรม JVM ทั้งโปรแกรมให้เป็นเนทีฟโค้ดแล้วทำงานตามโค้ดนั้น วิธีการนี้เรียกว่า Just In Time (JIT) เพราะคล้ายกับว่าทำการคอมไพล์โปรแกรม JVM ก่อนจะเริ่มทำงาน ซึ่งโปรแกรมจะทำงานเร็วขึ้นมากเพราะคำสั่งที่ถูกทำซ้ำบ่อยๆ เช่น คำสั่งที่อยู่ในประโยคทำซ้ำหรือฟังก์ชันที่ถูกเรียกใช้บ่อยๆ จะถูกแปลเพียงครั้งเดียวไม่ใช่ทุกครั้งที่ถูกนำไปใช้งาน

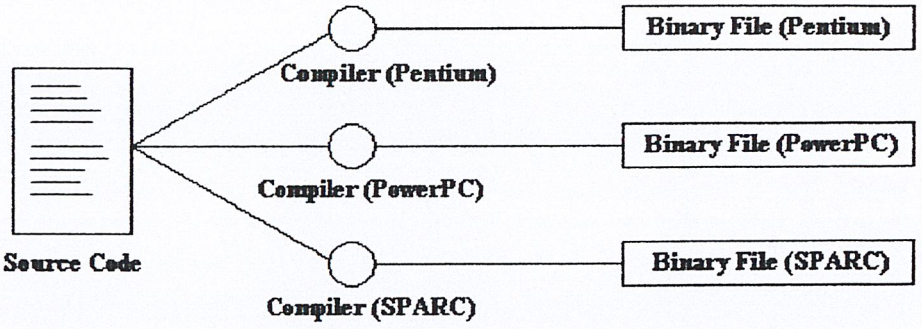
3.5 หลักการทำงานของภาษาจาวา

ในที่นี้ขออธิบายหลักการทำงานของภาษาจาวา โดยเริ่มตั้งแต่ทำการคอมไพล์ตัวโปรแกรมจนกระทั่งผู้ใช้ทำการเรียกใช้งานซึ่งมีลักษณะดังนี้

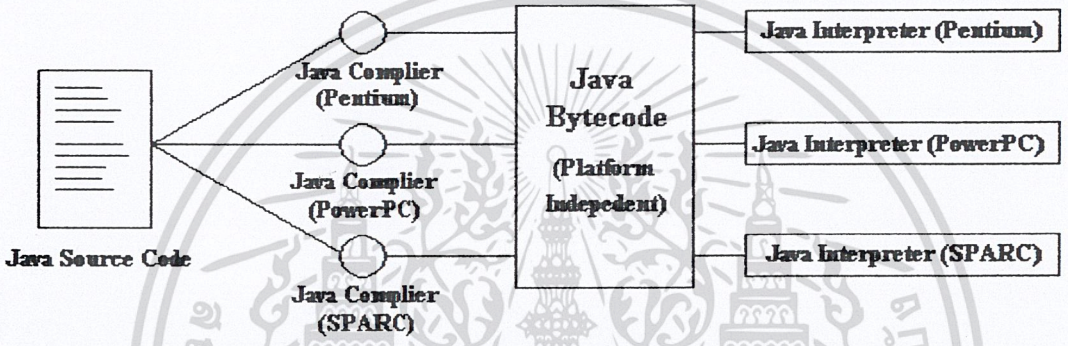
3.5.1 การคอมไพล์

คอมไพเลอร์ของภาษาจาวาก็เป็นเช่นเดียวกับคอมไพเลอร์ในภาษาอื่นๆ นั่นคือจะต้องสร้างรหัสภาษาเครื่อง (Machine Code หรือ Assembler Code) จากภาษาในระดับที่สูงกว่าเพื่อให้ซีพียูสามารถนำไปใช้งานได้แต่ข้อแตกต่างที่สำคัญระหว่างคอมไพเลอร์ของภาษาจาวากับภาษาอื่นๆ คือโพรเซสเซอร์หรือซีพียูที่จะคอยทำหน้าที่ในการปฏิบัติตามคำสั่งที่ได้จากการคอมไพล์ภาษาจาวานั้นไม่มีอยู่จริง เป็นเพียงสิ่งที่สมมติขึ้นมาที่เรียกว่า Java Virtual Machine นอกจากนี้การอ้างถึงส่วนต่างๆ ของโปรแกรมที่คอมไพล์ด้วยคอมไพเลอร์ของภาษาจาวาก็จะมีวิธีการที่แตกต่างออกไป

คอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนการอ้างถึงส่วนของโปรแกรม จากการใช้ชื่อแบบในภาษาสูง ไปเป็นตัวเลขเหมือนที่คอมไพเลอร์ในภาษาอื่นๆ ทำกัน และคอมไพเลอร์ภาษาจาวาก็จะไม่มีการสร้างแผนที่ของการจัดวางโปรแกรมบนหน่วยความจำขึ้นมาในระหว่างการคอมไพล์ ด้วยเหตุผลที่สำคัญคือเพื่อเป็นการสร้างความพอร์เทเบิลให้กับตัวโปรแกรม เพราะการจัดวางตำแหน่งของโปรแกรมจะต้องขึ้นอยู่กับลักษณะการทำงานของโพรเซสเซอร์ตัวใดตัวหนึ่ง การยังไม่จัดวางตำแหน่งช่วยทำให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลาง สามารถนำไปใช้บนคอมพิวเตอร์แพลตฟอร์มอื่นๆ ได้นอกจากนั้นยังทำให้เกิดความปลอดภัยอีกด้วย ซึ่งสิ่งที่ได้จากการคอมไพล์ในภาษาจาวาก็คือไบต์โค้ด



รูปที่ 3.2 การประมวลผลของโปรแกรมทั่วไปในระบบปฏิบัติการที่แตกต่างกัน



รูปที่ 3.3 การประมวลผลของโปรแกรมภาษาจาวา

3.5.2 การวางตำแหน่งในหน่วยความจำ

ในภาษาจาวาจะไม่มีการลดรูปแบบการอ้างถึงส่วนต่างๆ ของโปรแกรมจากการเรียกเป็นชื่อให้เหลือเพียงตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งลงในหน่วยความจำ คอมไพเลอร์ของภาษาจาวาจะทิ้งชื่อของแต่ละส่วนของโปรแกรม (โดยเฉพาะเมธอด) เอาไว้ในตัวโปรแกรมที่สร้างขึ้น เมื่อโปรแกรมทำงานจะเป็นหน้าที่ของตัวอินเทอร์พรีเตอร์ ซึ่งจะคอยเปิดตารางค้นหาที่อยู่ของเมธอดที่ต้องการเรียกใช้งาน โดยก่อนที่จะเริ่มทำงานจริงอินเทอร์พรีเตอร์จะต้องสร้างแผนที่ในการจัดวางสิ่งต่างๆ ลงในหน่วยความจำขึ้นมาก่อน แล้วจึงสร้างตารางขึ้นมาเพื่อช่วยหาตำแหน่งของเมธอดเมื่อมีการเรียกใช้งานโดยใช้ชื่อของเมธอด

3.5.3 การรันโค้ดที่ได้จากการคอมไพล์

การรันโค้ดที่คอมไพล์เอาไว้สำหรับ Java Virtual Machine เป็นหน้าที่ของตัวอินเทอร์พรีเตอร์ การรันโปรแกรมจะแบ่งได้เป็น 3 ขั้นตอนหลักๆ คือ การอ่าน การตรวจสอบความถูกต้อง และการทำตามโค้ด หน้าที่ในการอ่านโค้ดเข้าสู่ระบบจะเป็นของคลาสโหลดเดอร์ (Class Loader) หน้าที่การทำงานในส่วนนี้จะไม่ได้อ่านเข้ามาเฉพาะๆ ไฟล์จาวาที่กำลังจะเรียกใช้เท่านั้น แต่จะอ่านคลาสที่มีการอ้างถึงและคลาสที่มีการสืบทอดต่อมาโดยคลาสที่อ้างถึง เมื่อผ่านขั้นตอนนี้แล้วโค้ดทั้งหมดก็จะถูกส่งผ่านตัวตรวจ

สอบไบต์โค้ดเพื่อให้แน่ใจว่าโค้ดที่ส่งมามีความถูกต้องตามมาตรฐานของจาวาและจะไม่รบกวนเสถียรภาพของระบบ เมื่อผ่านการตรวจสอบแล้วโค้ดก็จะถูกส่งต่อไปยังระบบรันใหม่ (Run-Time System) ซึ่งจะสั่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่ง

3.5.4 คลาสโหลดเดอร์

คลาสโหลดเดอร์จะทำหน้าที่ดึงโค้ดทั้งหมดที่จำเป็นในการทำงานของแอพพลิเคชัน ไม่ว่าจะเป็นคลาสที่ถูกสืบทอดมาหรือคลาสอื่นๆ ที่มีการเรียกใช้เมื่อคลาสโหลดเดอร์ดึงคลาสใดเข้ามาแล้วก็จะจัดคลาสนั้นๆ ใส่เข้าไว้ใน namespace ของมันเองโดยการเก็บจะใช้ชื่อของคลาสเป็นสำคัญ ไม่ได้ใช้การอ้างถึงเป็นตัวเลข หลักการนี้จะเหมือนกับการทำงานของ Virtual Machine ในระบบปฏิบัติการที่สร้างขึ้นให้แอพพลิเคชันแต่ละตัวทำงาน ถ้าไม่ได้มีการเจาะจงเรียกใช้คลาสที่อยู่นอก namespace นี้ การเรียกใช้ชื่อต่างๆ ในคลาสก็จะไม่มีการรบกวนกันระหว่างคลาสเลย

คลาสทั้งหมดที่อยู่บนเครื่องโลคอล (local) จะได้รับช่องว่างแอดเดรส (Address Space) เป็นของตนเอง ส่วนคลาสต่างๆ ที่ดึงมาจากภายนอกจะได้รับ namespace เป็นของตนเอง การทำงานลักษณะนี้จะช่วยให้คลาสที่อยู่บนโลคอลทำงานได้ประสิทธิภาพดีขึ้นเพราะใช้ namespace ร่วมกันได้ แต่ก็ยังมีการป้องกันความผิดพลาดที่อาจจะเกิดจากคลาสที่ดึงเข้ามาจากภายนอก และในทางกลับกันคลาสที่นำเข้ามาก็ปลอดภัยจากความผิดพลาดที่อาจจะเกิดขึ้นจากคลาสโลคอลด้วย

เมื่อคลาสทั้งหมดที่เกี่ยวข้องกับการทำงานถูกนำเข้ามาเรียบร้อยแล้ว การจัดวางหน่วยความจำสำหรับเริ่มต้นการทำงานก็จะเกิดขึ้น ได้การเรียกชื่อต่างๆ จะสามารถจับคู่กับแอดเดรสจริงๆ ของหน่วยความจำได้ แล้วตัวโหลดเดอร์จะสร้างตารางสำหรับค้นหาที่อยู่เลี่ยงจากการทำงานผิดปกติของซูเปอร์คลาส (Super class) และการอ้างแอดเดรสที่ไม่ถูกต้องได้

3.5.5 การตรวจสอบไบต์โค้ด

เมื่อโค้ดเดินทางมาจนถึงขั้นตอนการสร้างตารางจับคู่ชื่อกับแอดเดรสแล้ว ก็ยังไม่สามารถแน่ใจได้ว่าโค้ดที่อ่านเข้ามาจะมีความปลอดภัยดังนั้นจึงต้องมีตัว Verifier หรือตัวตรวจสอบไบต์โค้ดทำหน้าที่ตรวจสอบความถูกต้องทีละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวา และ สอดคล้องกับการทำงานของตัวโปรแกรมเองหรือไม่ การตรวจสอบโค้ดในเชิงทฤษฎีจะสร้างและค้นหาปัญหาต่างๆ ได้หลายอย่างเช่น จะไม่มีการสร้างพอยต์เตอร์ที่เกินกว่าหน่วยความจำจริง ไม่มีคำสั่งใดสามารถละเมิดสิทธิ์การทำงานของตัวโปรแกรมได้ ไม่มีการจับคู่วัตถุผิด ไม่มีการให้โอเปอร์แรนด์มากหรือน้อยเกินไป การกำหนดค่าต่างๆ สำหรับไบต์โค้ดจะต้องถูกต้องครบถ้วนและจะไม่มีการแปลงข้อมูลผิดรูปแบบ

การใช้ตัวตรวจสอบตอบสนองจุดประสงค์ 2 ประการที่สำคัญคือ สิ่งต่างๆ ดังที่กล่าวมาแล้วจะถูกตรวจสอบก่อน ทำให้ตัวอินเทอร์พรีเตอร์มั่นใจได้ว่าไบต์โค้ดที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบ และจุดประสงค์ที่สองก็คือตัวอินเทอร์พรีเตอร์จะทำงานตามไบต์โค้ดได้รวดเร็วกว่า เพราะไม่ต้องคอยระวังว่าจะมีปัญหาเกิดขึ้นและไม่ต้องหยุดเป็นช่วงๆ เมื่อพบปัญหาและ

ต้องแก้ไข การทำงานไบต์โค้ดจะถูกตรวจสอบเพียงครั้งเดียวเท่านั้นและจะทำงานไปได้ตลอด ไม่ต้องมีการตรวจสอบซ้ำอีกเมื่อมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

3.5.6 การทำงานตามโค้ด

เมื่อคลาสโหลดเดอร์ได้รวบรวมโค้ดเข้ามาสู่ระบบทำการจัดวางในหน่วยความจำ และตัวตรวจสอบได้ทำการตรวจสอบความถูกต้องแล้ว โค้ดก็จะถูกส่งต่อไปยังตัวอินเทอร์พรีเตอร์เพื่อทำงานตามคำสั่ง การทำงานตามคำสั่งของโค้ดก็คือการเปลี่ยนโค้ดให้กลายเป็นคำสั่ง การทำงานจริงที่ตัวระบบไคลเอ็นต์ที่รันโค้ดนี้สามารถทำงานได้ ซึ่งวิธีการที่ทำได้ก็มีอยู่ 2 วิธีด้วยกันคือ ตัวอินเทอร์พรีเตอร์ทำการคอมไพล์โค้ดเหล่านี้ให้กลายเป็นเนทีฟโค้ดที่ตัวเครื่องไคลเอ็นท์เข้าใจแล้วค่อยทำงาน เพื่อให้ได้ความเร็วสูงสุดในการทำงาน อีกวิธีหนึ่งก็คือตัวอินเทอร์พรีเตอร์อ่านโค้ดเข้ามาแล้วตีความทำงานไปทีละคำสั่ง และทำการตีความไปเรื่อยๆ ตลอดเวลาที่มีการทำงาน

โดยปกติแล้วผู้สร้างตัวอินเทอร์พรีเตอร์มักจะเลือกใช้วิธีการที่สอง รูปแบบของไบต์โค้ดในภาษาจาวามีความยืดหยุ่นเพียงพอที่จะสามารถเปลี่ยนไปทำงานบนเครื่องไคลเอ็นต์แบบต่างๆ ได้โดยไม่มีการก่อให้เกิดโอเวอร์เฮดมากเกินความจำเป็น อย่างไรก็ตามไคลเอ็นต์ของจาวาบางระบบจะมีความสามารถในการทำงานได้ทั้งสองวิธีคือ โปรแกรมเมอร์สามารถจะเลือกใช้วิธีการคอมไพล์กับงานที่เน้นการคำนวนมากๆ เพื่อเป็นการเพิ่มสมรรถนะในการทำงานให้ได้เต็มที่ ซึ่งไคลเอ็นต์แบบนี้จะให้ได้ทั้งความพอร์เทเบิลและสมรรถนะที่ดี

การสร้างระบบรันไทม์ที่ดีจะต้องถ่วงดุลความสำคัญ 3 ประการให้พอเหมาะ นั่นคือความพอร์เทเบิล ความปลอดภัยและสมรรถนะเรื่องของความพอร์เทเบิล ซึ่งทำได้โดยการใช้รูปแบบของไบต์โค้ดที่มีความเป็นกลางเพียงพอ สามารถนำไปรันบนเครื่องคอมพิวเตอร์แบบอื่นๆ ได้ง่าย นอกจากนั้นการที่ตัวอินเทอร์พรีเตอร์ทำการกำหนดการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์ แทนที่จะเป็นระหว่างการคอมไพล์เหมือนภาษาอื่นๆ ก็เป็นการเพิ่มความแน่นอนว่าคลาสต่างๆ ที่นำเข้ามาจะยังคงใช้ได้อยู่ตลอดเวลา ในเรื่องของความปลอดภัยนั้นเป็นสิ่งที่มีการคำนึงถึงอยู่ตลอดการทำงานของระบบรันไทม์ โดยเฉพาะในส่วนของตัวตรวจสอบไบต์โค้ดที่ทำให้แน่ใจได้ว่าโปรแกรมจะทำงานได้ถูกต้อง ส่วนเรื่องของสมรรถนะนั้นก็สามารถจัดการได้ในสองระยะคือ พยายามเอาโอเวอร์เฮดทั้งหลายไปใส่ไว้ที่ตอนเริ่มต้นโหลดโปรแกรมเข้ามาสู่ระบบหรือไม่ก็กำหนดให้ทำงานเป็นแบบแบ็กกราวนด์ (Back-Ground Thread) ด้วยสิ่งต่างๆ เหล่านี้ทำให้จาวาสามารถปล่อยสมรรถนะระดับที่น่าพอใจ โดยยังคงไว้ซึ่งความพอร์เทเบิลและสภาพแวดล้อมที่ปลอดภัย นอกจากนั้นยังสามารถจะดึงสมรรถนะระดับสูงสุดออกมาใช้ได้ทันทีเมื่อต้องการ

3.5.7 การสร้างและรันภาษาจาวา

การสร้างโปรแกรมจากจาวาก็เหมือนกับภาษาโปรแกรมอื่นๆ คือเขียนโค้ดโปรแกรมจากอีดิเตอร์ใดๆ ก็ได้บันทึกอยู่ในนามสกุล .java จากนั้นจึงนำไปคอมไพล์ด้วยคอมไพเลอร์ของจาวาจะได้ไฟล์ใหม่ในนามสกุล .class ที่เก็บรหัสของการคอมไพล์ไว้จาวาเรียกรูปแบบข้อมูลที่อยู่ในไฟล์ใหม่นี้ว่าไบต์โค้ด ซึ่ง

ไฟล์ที่ได้คือแอพเพล็ตนั่นเอง แอพเพล็ตยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่เราคุ้นเคยกัน เพราะข้อมูลแบบไบต์โค้ดจะมีรูปแบบข้อมูลที่อยู่กึ่งกลางระหว่างโค้ดโปรแกรม (Source Code) กับโค้ดที่คอมพิวเตอร์อ่านแล้วนำไปปฏิบัติงานได้ทันที (Machine Code) หากจะรันแอพเพล็ตบนระบบใดๆ จะต้องใช้อินเทอร์พรีเตอร์จาวาของระบบนั้นๆ เพื่อตรวจสอบความถูกต้องของไบต์โค้ดแล้วแปลให้เป็นรหัสภาษาเครื่องส่งให้ระบบปฏิบัติการนำไปรันต่อไปขั้นตอนการทำงาน

3.6 Java Foundation Classes

โดยปกติชุดพัฒนาภาษาจาวาจะมีส่วนสนับสนุนการจัดการและเครื่องมือต่างๆ สำหรับสร้างส่วนติดต่อกับผู้ใช้งาน (User Interface) อยู่ก่อนแล้วคือ AWT (Abstract Window Toolkit) แต่ต่อมาทางบริษัท ซัน ไมโครซิสเต็มส์ ได้ร่วมมือกับหน่วยงานอื่นๆ คือ เน็ตสเคป, ไอบีเอ็มและ Lighthouse Design เพื่อทำการปรับปรุงประสิทธิภาพของส่วนติดต่อกับผู้ใช้งานให้มีความหลากหลายและใช้งานง่าย โดยสิ่งที่ถูกเพิ่มเข้ามาใหม่นี้ถูกเรียกว่าชุด JFC (Java Foundation Classes) ซึ่งถูกรวมอยู่กับ JDK เวอร์ชัน 1.1.5 ขึ้นไป แต่ที่เป็นมาตรฐานในการใช้งานจะอยู่ในรูปของ JDK เวอร์ชัน 1.2 โดยทางบริษัท ซัน ไมโครซิสเต็มส์ ได้เรียกชื่อใหม่ว่า Java 2

JFC เป็นชุดคำสั่งซึ่งแบ่งลักษณะการใช้งานออกเป็น 5 ชุดคำสั่งย่อยๆ ประกอบด้วย AWT API, Swing API, Java 2D API, Accessibility API และ Drag and Drop API ซึ่งทั้ง 5 ชุดคำสั่งย่อยนี้เกี่ยวข้องกับรูปแบบของการสร้างส่วนติดต่อกับผู้ใช้งาน

3.6.1 ชุดคำสั่ง AWT

AWT เป็นชุดคำสั่งที่บรรจุคอมโพเน็นต์ต่างๆ สำหรับติดต่อกับผู้ใช้งานทั้งในส่วนการรับข้อมูลและแสดงผล ซึ่งชุดคำสั่ง AWT ถูกบรรจุอยู่ใน JDK ตั้งแต่เวอร์ชันแรกและขยายคอมโพเน็นต์ให้มีความสามารถและความหลากหลายในการใช้งานมากขึ้น โดยลักษณะการทำงานของชุดคำสั่ง AWT เป็นแบบ Heavyweight ซึ่งจะมีผลในการนำไปใช้งานบนแพลตฟอร์มที่แตกต่างกัน ตัวอย่างเช่นการใช้งานบนวินโดวส์กับยูนิกซ์จะมีการแสดงรูปแบบของวัตถุ (Object) แตกต่างกัน

3.6.2 ชุดคำสั่ง Swing

สวิงซ์เป็นชุดคำสั่งที่บรรจุคอมโพเน็นต์ต่างๆ สำหรับติดต่อกับผู้ใช้งานเช่นเดียวกับ AWT แต่มีรูปแบบการใช้งานที่หลากหลายมากขึ้น โดยจัดเป็นคอมโพเน็นต์แบบ Lightweight ทำให้การแสดงผลบนจอภาพเหมือนกันแม้ว่าจะนำไปใช้งานบนแพลตฟอร์มที่ต่างกัน ทำให้หากมีผู้ใช้นำโปรแกรมไปใช้งานบนวินโดวส์หรือยูนิกซ์จะมีลักษณะการแสดงรูปแบบของวัตถุที่เหมือนกัน

3.6.3 ชุดคำสั่ง Java 2D

ชุดคำสั่งจาวา 2 มิติเป็นชุดคำสั่งที่เพิ่มเข้ามาใหม่ตั้งแต่ JDK เวอร์ชัน 1.2 ขึ้นไป โดยรวบรวมคลาสที่เกี่ยวข้องกับการจัดการภาพและตัวอักษรในรูปแบบ 2 มิติ ซึ่งเป็นผลมาจากการพัฒนาร่วมกัน

ระหว่างบริษัท ซัน ไมโครซิสเต็มส์ และ IBM Taligent โดยตั้งความหวังว่าชุดคำสั่งจาวา 2 มิติจะช่วยให้การจัดการแสดงภาพง่ายขึ้นและสามารถนำไปสร้างโปรแกรมสำหรับการนำเสนอได้อย่างสะดวก

3.6.4 ชุดคำสั่ง Accessibility

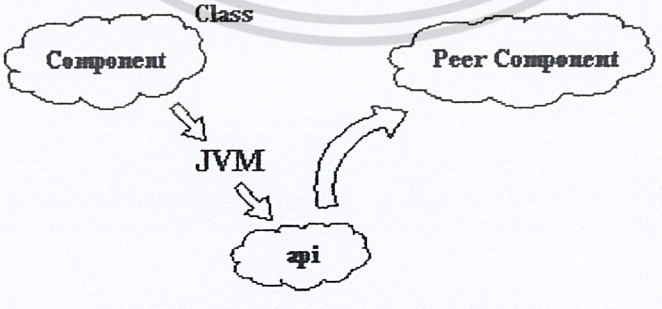
ชุดคำสั่ง Accessibility คือชุดคำสั่งซึ่งใช้สร้างส่วนติดต่อสำหรับผู้ที่ทุพลภาพให้สามารถใช้โปรแกรมได้ ซึ่งสิ่งนี้เป็นสิ่งที่บริษัท ซัน ไมโครซิสเต็มส์ ได้เล็งเห็นว่าจะสามารถช่วยให้โปรแกรมเมอร์สร้างโปรแกรมเพื่อให้ผู้ทุพลภาพไม่ว่าจะเป็นความพิการทางด้านใดสามารถใช้งานคอมพิวเตอร์ได้ ความสามารถของชุดคำสั่ง Accessibility มีมากมาย ตัวอย่างเช่นชุดคำสั่งสำหรับอ่านออกเสียงข้อมูลบนหน้าจอ (Screen Reader) และชุดคำสั่งจดจำเสียงสนทนา (Speech Recognition) เป็นต้น

3.6.5 ชุดคำสั่ง Drag and Drop

ชุดคำสั่ง Drag และ Drop เป็นชุดคำสั่งซึ่งสนับสนุนการทำงานโดยการลากและปล่อยเมาส์ โดยส่วนสำหรับติดต่อกับผู้ใช้งานจะสังเกตเหตุการณ์ที่ผู้ใช้งานทำการเลือกชิ้นส่วนที่ต้องการ โดยวิธีการลากเมาส์ และนำไปวางไว้บนพื้นที่ใดๆ ซึ่งเราพอสังเกตพฤติกรรมการใช้งานเหล่านี้ได้จากโปรแกรมต่างๆ เช่น PowerPoint และ Visio เป็นต้น

3.6.6 ความแตกต่างระหว่างชุดคำสั่ง AWT และชุดคำสั่ง Swing

หลายคนอาจสงสัยทำไมชุดคำสั่ง AWT จึงขึ้นอยู่กับแพลตฟอร์มทั้งๆ ที่โปรแกรมภาษาจาวานั้นไม่ขึ้นกับแพลตฟอร์มใดๆ เหตุผลที่เป็นเช่นนี้เนื่องจากอินเทอร์พรีเตอร์ของภาษาจาวาจะต้องเป็นของแพลตฟอร์มใดแพลตฟอร์มหนึ่งเท่านั้น เพราะต้องเปลี่ยนไบต์โค้ดเป็นคำสั่งของแพลตฟอร์มนั้น เมื่อโปรแกรมของคลาส AWT ถูกคอมไพล์เป็นไบต์โค้ดแล้วโปรแกรมไบต์โค้ดที่ได้จะไม่ขึ้นกับแพลตฟอร์ม แต่ช่วงที่ถูกทำงานโดยอินเทอร์พรีเตอร์จะถูกเปลี่ยนเป็นชุดคำสั่งของแพลตฟอร์มนั้น แล้วทำการวาดและควบคุมการทำงานของตัวคอมโพเน็นต์



รูปที่ 3.4 การทำงานของชุดคำสั่ง AWT

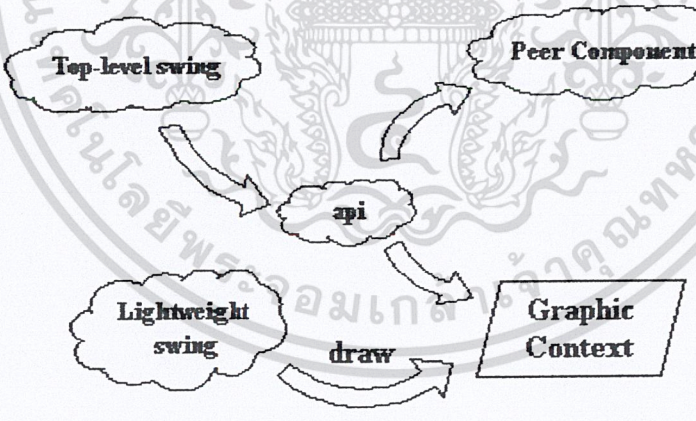
ซึ่งจากวิธีดังกล่าวมีข้อเสียคือโปรแกรมที่ใช้คลาสของ AWT ตัวอย่างเช่น Button หากนำไปทำงานบน win32 จะได้ button แบบของวินโดวส์ แต่ในโปรแกรมเดียวกันหากนำไปทำงานบนโซลาริส

จะได้ button แบบโมทีฟ (Motif) ทำให้คอมโพเน็นต์หนึ่งเมื่อนำไปใช้งานบนแพลตฟอร์มที่ต่างกันจะได้รูปร่างที่แตกต่างกันและที่แย่ไปกว่านั้นคือบางคอมโพเน็นต์อาจมีพฤติกรรมไม่เหมือนกันเมื่ออยู่ต่างแพลตฟอร์ม ด้วยเหตุนี้ทำให้โปรแกรมที่ใช้งานชุดคำสั่ง AWT ต้องขึ้นกับแพลตฟอร์ม

ข้อเสียอีกประการหนึ่งคือ ไบต์โค้ดของคอมโพเน็นต์หนึ่งเมื่อถูกทำงานจะต้องเปลี่ยนไปเป็นคำสั่งจำนวนมากเพื่อวาดและควบคุมตัวคอมโพเน็นต์นั้น เพราะคำสั่งหนึ่งมักจะทำงานที่ค่อนข้างพื้นฐานมาก การติดต่อเรียกทำงานระหว่างสภาพแวดล้อมเสมือน (Java Virtual Machine) กับคำสั่งรวมทั้งการนำส่งเหตุการณ์ (event) ให้แก่กันทำได้ค่อนข้างช้า มีโอเวอร์เฮดสูง เราจึงเรียกคอมโพเน็นต์ประเภทนี้ว่า heavyweight ซึ่งหมายถึงคอมโพเน็นต์ที่ต้องอาศัยคำสั่งของแพลตฟอร์มนั้น ซึ่งกราฟริกของส่วนติดต่อกับผู้ใช้งานในชุดคำสั่ง AWT ทุกตัวเป็นคอมโพเน็นต์ heavyweight

ทางแก้หนึ่งของปัญหาดังกล่าวคือ การใช้คอมโพเน็นต์ lightweight ซึ่งหมายถึงคอมโพเน็นต์ที่วาดตัวเองและจัดการเหตุการณ์ของตัวเองด้วยภาษาจาวาโดยไม่ต้องใช้คำสั่งของแพลตฟอร์มนั้นเลย จึงทำให้รูปร่างและพฤติกรรมของคอมโพเน็นต์ไม่ขึ้นกับแพลตฟอร์ม รวมทั้งปัญหาเกี่ยวกับโอเวอร์เฮดของคำสั่งก็หมดไป

แม้ว่ากราฟริกของส่วนติดต่อกับผู้ใช้งานในสวิงซ์ไม่ใช่คอมโพเน็นต์ lightweight ทั้งหมด เนื่องจากคอมโพเน็นต์ lightweight ไม่สามารถใช้จาวาวาดขึ้นบนจอภาพหรืออุปกรณ์รองรับกราฟริกได้โดยตรง แต่จะสามารถวาดลงบนพื้นที่กราฟริกของคอมโพเน็นต์ตัวหนึ่งที่สร้างไว้ให้ก่อนแล้วเท่านั้น เราเรียกคอมโพเน็นต์ที่มีกราฟริกสำหรับให้คอมโพเน็นต์ตัวอื่นวาดว่าคอมโพเน็นต์ top-level swing



รูปที่ 3.5 การทำงานร่วมกันของคอมโพเน็นต์ top-level swing กับคอมโพเน็นต์ lightweight

จะสังเกตได้ว่าคอมโพเน็นต์ lightweight ของสวิงซ์คือกลุ่มที่ไม่ใช่คอมโพเน็นต์ top-level swing โดยในชุดคำสั่งสวิงซ์มีเพียงบางตัวเป็นคอมโพเน็นต์ heavyweight ทำหน้าที่เป็นคอมโพเน็นต์ top-level swing ให้คอมโพเน็นต์ lightweight ตัวอื่นๆ วาดใส่ตัวมัน ดังนั้นในการใช้งานแล้วเราต้องมีคอมโพเน็นต์ top-level swing อย่างน้อยตัวหนึ่งก่อนจะวาดคอมโพเน็นต์ lightweight ลงไปได้

3.7 เครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวา

การศึกษาและพัฒนาโปรแกรมภาษาจาวาในช่วงเริ่มแรกนั้นมีลักษณะที่เรียกว่า command line driven compiler โดยเครื่องมือแรกที่ใช้ในการประยุกต์เพื่อสร้างโปรแกรมภาษาจาวาคือ Sun's Java Development Kit (JDK) ในปัจจุบันมีเครื่องมือที่ที่ช่วยในการพัฒนาโปรแกรมภาษาจาวาซึ่งได้รวมเอาคุณสมบัติของการสร้างกราฟริกบนส่วนสำหรับติดต่อกับผู้ใช้งาน (Graphic User Interface) ไว้ในการพัฒนาจาวาแอพพลิเคชัน โดยเครื่องมือสำหรับพัฒนาเหล่านี้จะต้องเข้ากับมาตรฐานของภาษาจาวา ซึ่งทำให้เครื่องมือที่ใช้พัฒนาโปรแกรมภาษาจาวามีความง่าย และสะดวกกว่าในการที่จะศึกษาและใช้งานภาษาจาวามากขึ้น เครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวานั้นมีหลายบริษัทที่พัฒนาเครื่องมือเหล่านั้นออกมา โดยตัวอย่างเครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวาที่รู้จักกันมีดังต่อไปนี้

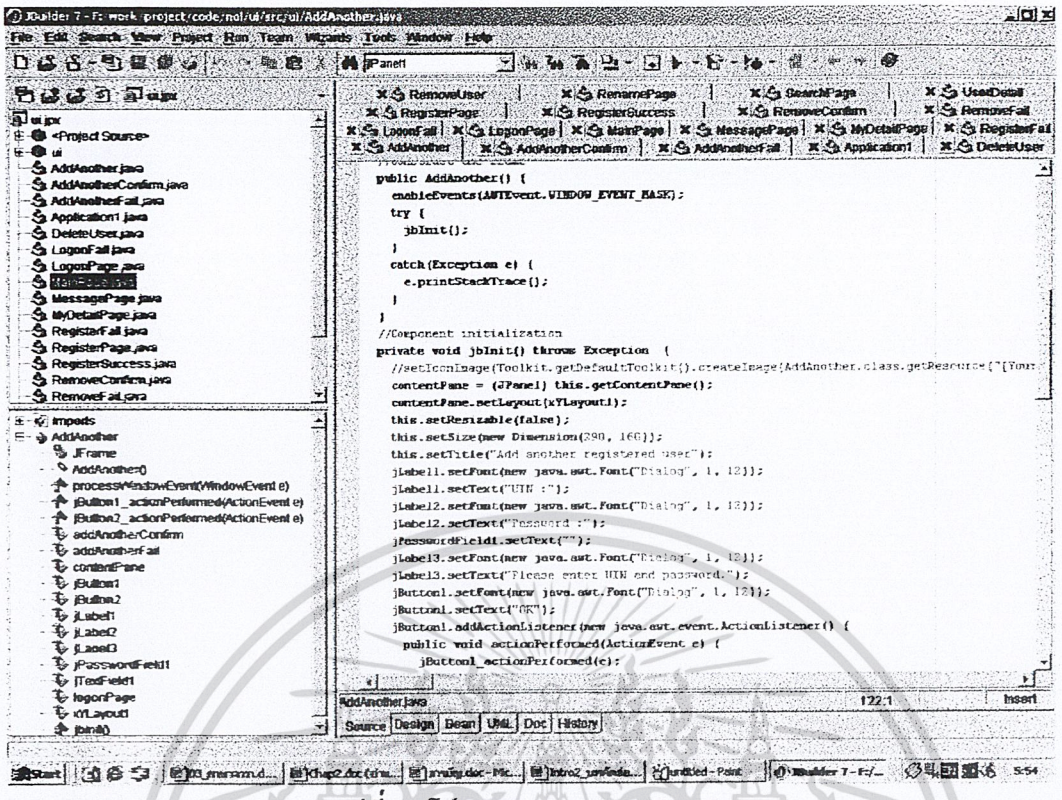
3.7.1 Java Development Kit

Java Development Kit หรือ JDK ถือเป็นเครื่องมือแรกของบริษัท ซัน ไมโครซิสเต็มส์ ที่ผลิตออกมาเป็นเครื่องมือสำหรับใช้ช่วยในการสร้างและพัฒนาโปรแกรมภาษาจาวา ซึ่งประกอบไปด้วยไฟล์ต่างๆ สำหรับใช้ในการทำงานดังต่อไปนี้

1. ไฟล์ appletviewer.exe ใช้สำหรับทดสอบจาวาแอพเพล็ต เพื่อดูผลการทำงาน โดยไม่ต้องสั่งรันผ่านทางเว็บบราวเซอร์
2. ไฟล์ java.exe เป็นอินเทอร์พรีเตอร์สำหรับใช้ทดสอบการทำงานของจาวาแอพพลิเคชัน
3. ไฟล์ javac.exe มาจากคำว่าจาวาคอมไพเลอร์ใช้สำหรับแปลซอสโค้ดของโปรแกรมที่เราเขียนขึ้น (.java) ให้เป็นโปรแกรมไบต์โค้ด (.class)
4. ไฟล์ javap.exe ใช้แปลงคลาสไฟล์ของจาวาที่เป็นภาษาแอสเซมบลีกลับมาเป็นซอสโค้ด
5. ไฟล์ javadoc.exe ใช้สำหรับสร้างไฟล์เอกสารจากซอสโค้ดของจาวาในรูปแบบของ HTML ภายในจะเป็นการบอกรายละเอียดของคลาสต่างๆ ที่ผู้ใช้เขียนขึ้น กฎการเขียนที่ควรรู้ไว้กรณีที่มีผู้สนใจอยากจะสร้างคือ ผู้ใช้จะต้องใส่คำสั่ง /** เข้าไปด้วยเสมอ ตรงส่วนบนของโปรแกรมก่อนการประกาศคลาส ลักษณะการใช้งานจะเหมือนกับคำสั่ง // และ /* */ (สำหรับใส่ Comment คำอธิบายทั่วๆ ไป)

3.7.2 Borland JBuilder

Borland JBuilder เป็นเครื่องมือสำหรับพัฒนาโปรแกรมภาษาจาวาของบอร์แลนด์ ซอฟท์แวร์ คอร์ปอเรชัน (Borland Software Corporation) ซึ่ง JBuilder เป็นเครื่องมือที่สนับสนุนการทำงานบนระบบปฏิบัติการวินโดวส์, โซลาริส, ลีนุกซ์และแมค จุดเด่นของ JBuilder ที่สำคัญคือสามารถใช้งานได้ง่าย สามารถตรวจสอบจุดบกพร่องได้ดี ซึ่งทำให้ JBuilder เป็นเครื่องมือที่เหมาะสำหรับงานสร้างและพัฒนาโปรแกรมขนาดใหญ่ นอกจากนี้ JBuilder ยังมียูทิลิตี้ (Utilities) สำหรับเพิ่มความสามารถให้กับโปรแกรมอีกมากมายเช่น Package Migration tool และ JDBC Explorer เป็นต้น



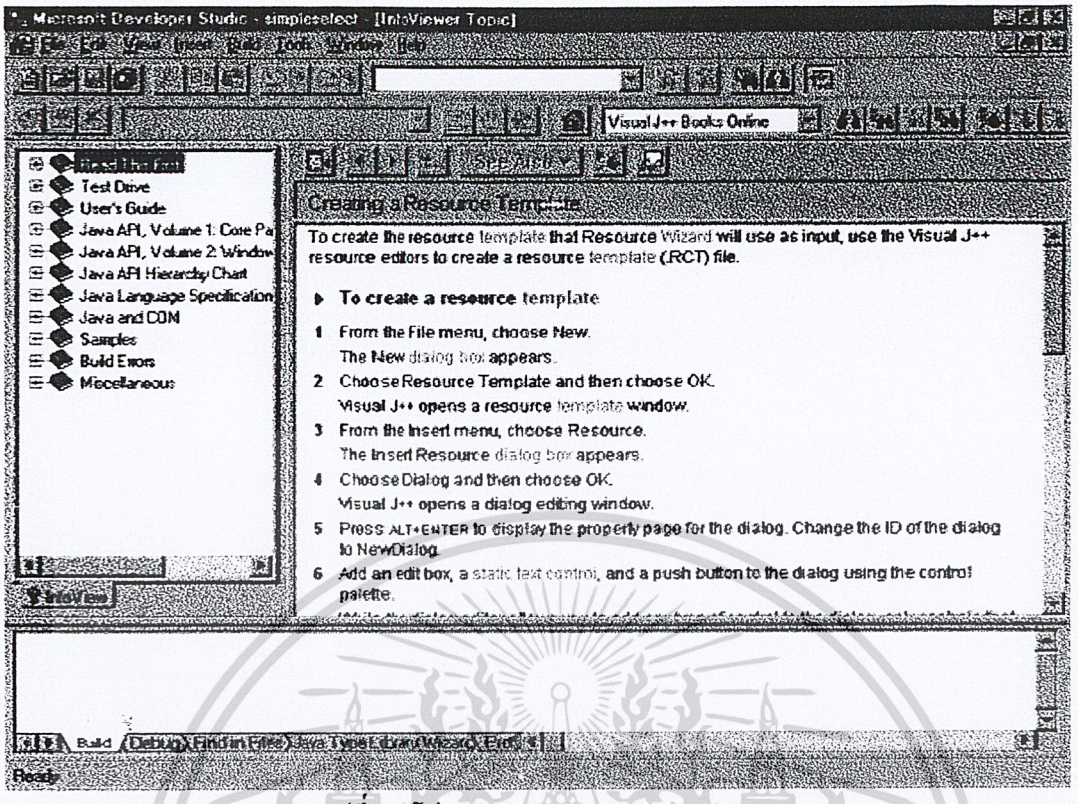
รูปที่ 3.6 โปรแกรม Borland JBuilder

หากดูจากภาพรวมแล้ว JBuilder ก็เป็นเครื่องมือในการพัฒนาโปรแกรมจาวาที่น่าใช้งานตัวหนึ่ง แต่ JBuilder ก็มีจุดอ่อนที่ต้องใช้หน่วยความจำมากและประมวลผลช้า นอกจากนี้โปรแกรม JBuilder สำหรับองค์กรธุรกิจก็มีราคาสูงมาก

3.7.3 Microsoft Visual J++

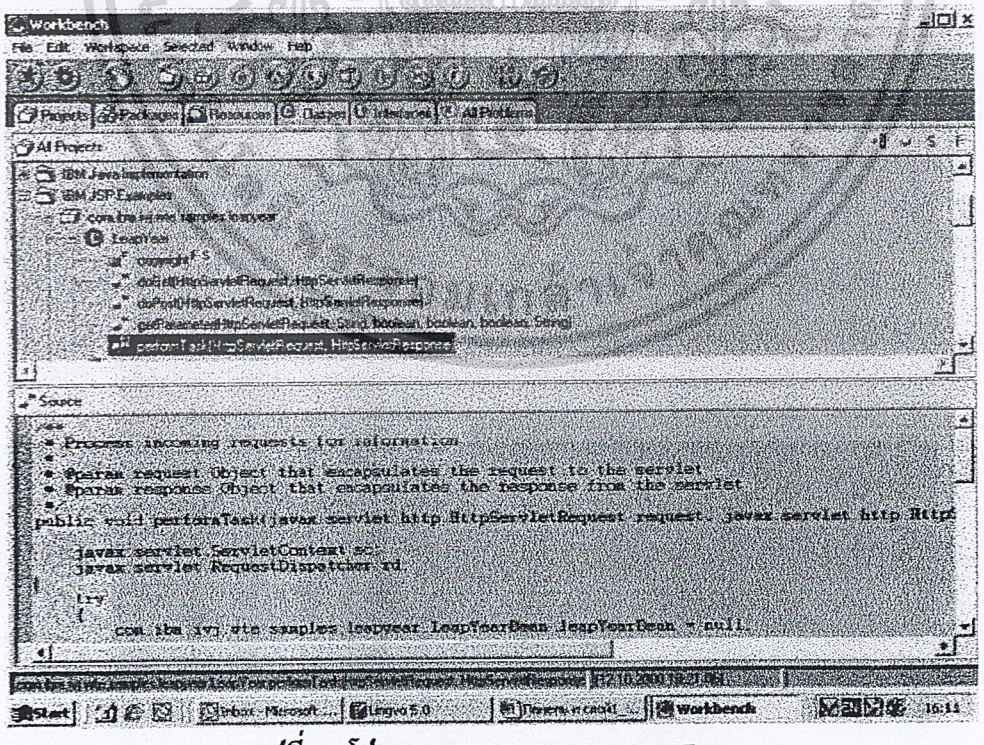
Visual J++ เป็นเครื่องมือที่ใช้พัฒนาการเขียนจาวาแอพเพล็ตและแอพพลิเคชันโดยไมโครซอฟท์คอร์ปอเรชัน ซึ่ง Visual J++ จัดเป็นเครื่องมือที่มีจุดเด่นในเรื่องการตรวจสอบจุดบกพร่องเหมือนกับ JBuilder แต่ไม่ใช้หน่วยความจำมากเหมือนกับ JBuilder นอกจากนี้ Visual J++ ยังมีความสามารถในด้าน visually create java forms โดย resource editor จะยอมให้ผู้ใช้สามารถใช้ visually layout ในการติดต่อกับแอพพลิเคชันของผู้ใช้เอง ซึ่งผู้ใช้งานยังสามารถที่จะคอนเวิร์ตเป็นรูปแบบของ Visual Basic และ Visual C++

จากลักษณะของการคอนเวิร์ตใน Visual J++ จะพบว่า Visual J++ จะสนับสนุนการใช้งานร่วมกับเครื่องมือหรือคอมโพเน็นต์ของไมโครซอฟท์ได้ดี ซึ่งในส่วนนี้เองทำให้ Visual J++ ไม่เข้ากับจาวาได้ดี (Java-compatible) เหมือนกับเครื่องมือพัฒนาโปรแกรมจาวาอื่นๆ



รูปที่ 3.7 โปรแกรม Microsoft Visual J++

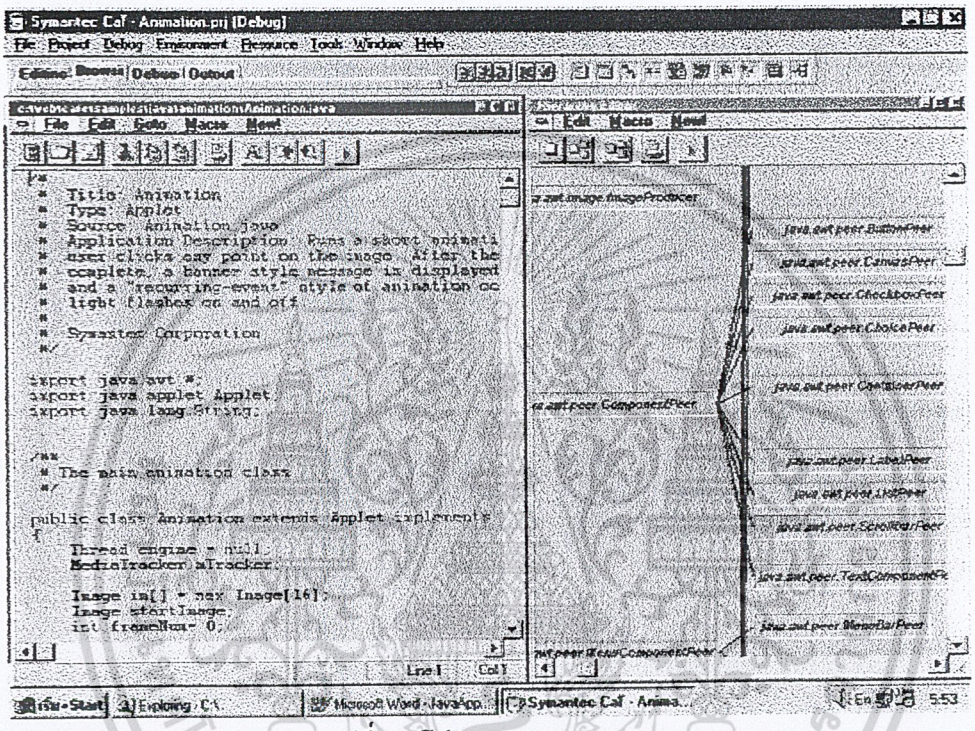
3.7.4 IBM VisualAge for Java



รูปที่ 3.8 โปรแกรม IBM VisualAge for Java

VisualAge เป็นเครื่องมือที่ใช้พัฒนาโปรแกรมจาวาของไอบีเอ็ม VisualAge จัดว่าเป็นเครื่องมือที่มีความน่าเชื่อถือสูง ตรวจสอบจุดบกพร่องได้ดี สามารถใช้งานได้ทั้งบนระบบปฏิบัติการวินโดวส์และลีนุกซ์ แต่ข้อเสียของ VisualAge คือการใช้งานค่อนข้างยุ่งยาก อาจต้องใช้เวลาศึกษานาน เป็นโปรแกรมที่มีขนาดใหญ่ ใช้หน่วยความจำมากและประมวลผลช้า

3.7.5 VisualCafe



รูปที่ 3.9 โปรแกรม VisualCafe

VisualCafe เป็นเครื่องมือที่ถูกคิดค้นและพัฒนาโดย Symantec แต่ปัจจุบันถูกพัฒนาโดย Webgain ซึ่ง VisualCafe จัดเป็นเครื่องมือที่มีขนาดเล็ก ใช้หน่วยความจำไม่มาก สามารถทำงานได้รวดเร็วและสนับสนุนการใช้งาน Java 2 สำหรับจุดบกพร่องของ VisualCafe คือสามารถใช้งานได้กับระบบปฏิบัติการวินโดวส์เท่านั้นและเป็นเครื่องมือที่มีความน่าเชื่อถือต่ำ

3.8 ความแตกต่างระหว่างจาวาแอพพลิเคชันและจาวาแอพเพล็ต

จาวาแอพพลิเคชันและจาวาแอพเพล็ต ต่างเป็นภาษาคอมพิวเตอร์ซึ่งถูกออกแบบมาเพื่อใช้ในงานอินเทอร์เน็ตเหมือนกัน โดยทำงานอยู่บนโฮมเพจร่วมกับภาษา HTML เหมือนกัน แต่ก็มีลักษณะการนำไปใช้งานที่แตกต่างกันคือ

- จาวาแอพพลิเคชันสามารถปฏิบัติงานเดี่ยวๆ ได้ด้วยตัวเองเช่นเดียวกับการเขียนโปรแกรมด้วยภาษาอื่นๆ ทั่วไป แต่สิ่งที่ควรจดจำในการเขียนจาวาแอพพลิเคชันก็คือผู้เขียนโปรแกรมจะต้องมี

เมธอดของ Main(string args[]) อย่างน้อยๆ 1 เมธอดสำหรับใช้เป็นจุดเริ่มต้นของโปรแกรม ซึ่งหากสังเกตให้ดีจะเหมือนกับการเขียนโปรแกรมโดยใช้ภาษาซี ส่วนวิธีการคอมไพล์โปรแกรมก็จะใช้ javac.exe ซึ่งจะทำให้ได้ไฟล์นามสกุล .class เพื่อใช้ในการแสดงผลของโปรแกรม โดยในส่วนการสั่งแสดงผลเราจะใช้โปรแกรม java.exe ซึ่งจะทำการแสดงผลของไฟล์นามสกุล .class ที่ได้มาจากการคอมไพล์ในข้างต้น

- จาวาแอพเพล็ตไม่สามารถทำงานเดี่ยวๆ ด้วยตนเองได้เหมือนกับจาวาแอพพลิเคชัน โดยที่จาวาแอพเพล็ตจะทำงานร่วมกันกับไฟล์ HTML ซึ่งเราจะเรียกใช้ไฟล์ที่เขียนจากจาวาแอพเพล็ตนี้ผ่านทางแท็กคำสั่ง <APPLET>...</APPLET> ภายในไฟล์ HTML จากนั้นก็จะเรียกใช้ไฟล์โปรแกรม javac.exe มาทำการคอมไพล์เช่นเดียวกับการใช้จาวาแอพพลิเคชันซึ่งจะได้ไฟล์ที่มีนามสกุล .class ไว้สำหรับให้ไฟล์ HTML เรียกผ่านวิธีสั่งแสดงผลการทำงานของจาวาแอพเพล็ต ซึ่งในการแสดงผลของจาวาแอพเพล็ตนั้นจะใช้โปรแกรม appletviewer.exe หรือเว็บบราวเซอร์ เช่น เน็ตสเคป, ฮอตจาวา, Internet Explorer และอื่นๆ โดยเลือกใช้ตัวใดตัวหนึ่งก็ได้

3.9 ทิศทางของจาวา

จาวาเริ่มต้นจากการเป็นภาษาคอมพิวเตอร์ง่ายๆ สำหรับสร้างแอพเพล็ตหรือโปรแกรมขนาดเล็กๆ ซึ่งสามารถทำงานบนเครื่องคอมพิวเตอร์ระบบใดๆ ก็ได้ โดยไม่จำเป็นต้องสร้างโปรแกรมที่ใหญ่โตมากนักขึ้นมาปัจจุบันนี้กระแสความต้องการและความนิยมของจาวายังคงสูงขึ้นอย่างรวดเร็ว ส่งผลให้เกิดพัฒนาโปรแกรมประยุกต์หรือแอพพลิเคชันใหม่ๆ โดยใช้ภาษาจาวา ซึ่งจาวาไม่ได้เป็นเพียงแค่แอพเพล็ตเล็กๆ อีกต่อไปแล้วเพราะเราสามารถใช้จาวาเป็นเครื่องมือสำหรับสร้างซอฟต์แวร์ขนาดใหญ่

ในปี ค.ศ.1996 ถือเป็นปีแห่งการพัฒนาซอฟต์แวร์ประเภทไคลเอ็นต์ หลายต่อหลายฝ่ายต่างพุ่งเป้าความสนใจไปที่เครื่องคอมพิวเตอร์ประเภทโต้ตอบกับผู้ใช้ในฝั่งไคลเอ็นต์กับแอพพลิเคชันต่างๆ ที่สำหรับใช้ภายในองค์กร ต่อมาในปี ค.ศ.1997 เป็นปีทองที่ซอฟต์แวร์จะก้าวไปไกลมากกว่าไคลเอ็นต์ โดยจะเคลื่อนที่เข้าไปใกล้ชิดกับผู้ใช้งานยิ่งขึ้นในรูปแบบของอุปกรณ์ส่วนบุคคลต่างๆ โดยการใช้จาวาควบคุมอุปกรณ์ไฟฟ้าและเครื่องมือช่วยระบบดิจิตอลส่วนบุคคล, อุปกรณ์ Set-top Boxes (กล่องควบคุมอเนกประสงค์) Smart Phones, ระบบฝังตัวในอุปกรณ์ต่างๆ เช่นในเครื่องพิมพ์, เครื่องส่งแฟกซ์, โทรศัพท์เว็บ, โทรทัศน์เว็บบราวเซอร์ รวมไปถึงเครื่องใช้ไฟฟ้าต่างๆ ซึ่งอุปกรณ์เหล่านี้จะถูกเชื่อมการติดต่อกับระบบ Smart Cards นั่นหมายถึงว่าผู้บริโภคสามารถที่จะตั้งโปรแกรมควบคุมระยะไกลกับสวิทช์ไฟฟ้าผ่านทางอุปกรณ์ช่วยขนาดเล็กเหล่านี้

นอกจากนี้ในปัจจุบันจาวายังเข้ามามีบทบาทกับการใช้งานบนโทรศัพท์มือถือ PDAs และอุปกรณ์ไร้สาย โดยที่ผู้พัฒนาสามารถเขียนโปรแกรมเพื่อสร้างแอพพลิเคชันให้ทำงานบนอุปกรณ์เหล่านี้ได้เสมือนเป็นคอมพิวเตอร์ขนาดเล็กอีกเครื่องหนึ่ง ซึ่งเทคโนโลยีการเขียนโปรแกรมด้วยภาษาจาวาบนเครื่องมือสื่อสารเหล่านี้ก็คือ J2ME (Java 2 Micro Editon) ตัวอย่างของแอพพลิเคชันที่พัฒนามาจาก J2ME และนำมาใช้งานบนอุปกรณ์สื่อสารได้เช่น โปรแกรมวิเคราะห์และแสดงราคาหุ้น, โปรแกรมรับส่งไฟล์, โปรแกรมรับส่งสารด่วนและเกมส์ออนไลน์เป็นต้น ทำให้การใช้งานอุปกรณ์สื่อสารกลับมามีผู้ให้

ความสนใจอย่างมากอีกครั้งหนึ่ง และจากการเติบโตของการใช้งานอุปกรณ์สื่อสารนี้ทางบริษัท ซัน ไมโครซิสเต็มส์ ร่วมกับ 3COM จึงพัฒนาอุปกรณ์ปาล์มท็อปที่ใช้เทคโนโลยีของจาวาออกมาด้วย เพื่อเป็นการชดเชยความสามารถของปาล์มท็อป เพราะปาล์มท็อปมีข้อจำกัดคือ พื้นที่ในการเก็บข้อมูลซึ่งโดยส่วนใหญ่ปาล์มท็อปจะมีพื้นที่สำหรับเก็บข้อมูลกัน 2-8 เมกะไบต์ (สามารถขยายได้ถึง 16 เมกะไบต์) ดังนั้นการจะเก็บตัวโปรแกรมและข้อมูลทุกอย่างไว้ในเครื่องนั้นย่อมเป็นไปไม่ได้ ดังนั้นจาวาจึงถือเป็นเครื่องมือหนึ่งที่เหมาะสมสำหรับพัฒนางานนี้

ซึ่งขณะนี้ยังคงมีนักประดิษฐ์จำนวนมากต่างพยายามที่จะสร้างสรรค์อุปกรณ์แปลกๆ ใหม่ๆ ขึ้นมาเพื่อตอบสนองความต้องการของมนุษย์ ซึ่งภาษาจาวายังคงเป็นเครื่องมือหนึ่งที่มีศักยภาพสูงสุดในการสนับสนุนอุปกรณ์ใหม่ๆ เหล่านี้

บทที่ 4
การเข้ารหัสข้อมูล

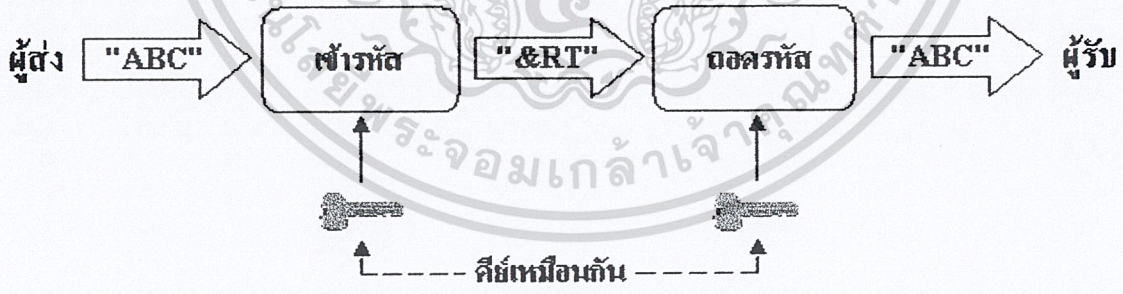
4.1 ระบบของการเข้ารหัสข้อมูล

การเข้ารหัสข้อมูลคือการทำให้ข้อมูลที่ต้องการเป็นความลับ ซึ่งถือเป็นส่วนสำคัญในระบบข้อมูลปัจจุบัน โดยอาศัยหลักการของการเข้ารหัส (Encryption) และการถอดรหัส (Decryption)

- การเข้ารหัสเป็นการเปลี่ยนรูปข้อมูลโดยผ่านรูปแบบและกระบวนการแปรรูปของข้อมูล ทำให้ข้อมูลที่ส่งมีรูปแบบที่ไม่เหมือนเดิมเพื่อทำให้ข้อมูลเป็นความลับ
- การถอดรหัสเป็นการแปลงข้อมูลที่ได้ผ่านการเข้ารหัสให้กลับมาเป็นข้อมูลเดิม ซึ่งการเข้ารหัสและการถอดรหัส จะถูกควบคุมโดยกุญแจหรือที่เรียกว่า "Key"

4.1.1 ระบบการเข้ารหัสแบบสมมาตร

ในระบบการเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem) ซึ่งสามารถเรียกว่าการเข้ารหัสแบบคีย์เดียว (Single Key Encryption) หรือการเข้ารหัสแบบกุญแจความลับ (Secret Key Encryption) ทั้งผู้รับและผู้ส่งจะต้องมีคีย์ซึ่งเป็นความลับที่เหมือนกันในการเข้าและถอดรหัสข้อมูล หากมีคีย์ที่ต่างกันก็จะทำให้ข้อมูลที่สื่อสารกันผิดพลาด ตัวอย่างการเข้ารหัสระบบนี้ได้แก่ การเข้ารหัสแบบ DES, Triple-DES และ IDEA เป็นต้น



รูปที่ 4.1 แสดงการเข้ารหัสและถอดรหัสแบบสมมาตร

ในงานบางประเภทการเข้ารหัสแบบสมมาตรอาจจะยังไม่เพียงพอ เนื่องจากการเข้ารหัสแบบสมมาตรนั้นมีปัญหาในด้านความปลอดภัยดังนี้

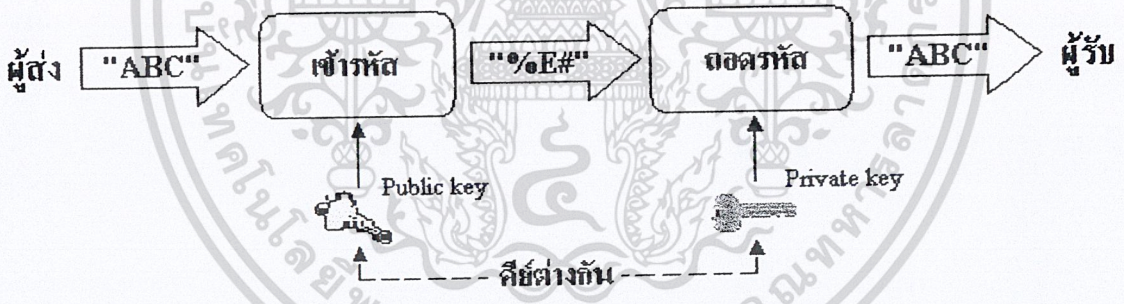
1. ระบบที่ใช้คีย์ที่เป็นความลับเพียงคีย์เดียวในการเข้ารหัสข้อมูล หากคีย์นั้นถูกขโมยไป บุคคลที่ทราบคีย์นั้นสามารถถอดรหัสข้อมูลที่ถูกเข้ารหัสไว้ได้ นอกจากนั้นอาจปลอมแปลงข้อมูลนั้นใหม่แล้วเข้ารหัสข้อมูลที่ทำการแปลงนั้นส่งไปยังผู้รับที่มีคีย์ซึ่งเป็นความลับเช่นเดียวกัน ทำให้ผู้รับนั้นได้รับข้อ

มูลที่ผิดพลาด ดังนั้นเพื่อความปลอดภัยควรเลือกคีย์ที่ยากแก่การคาดเดาและเก็บไว้อย่างปลอดภัย รวมทั้งไม่ควรใช้คีย์เดียวกันนี้ซ้ำกันหลายครั้ง

2. การเข้ารหัสแบบสมมาตรค่อนข้างอ่อนและเสี่ยงต่อการโจรกรรม เนื่องจากความยาวคีย์ที่ใช้ไม่มาก พอหรืออัลกอริทึมที่ใช้นั้นค่อนข้างง่ายไม่ซับซ้อนมากนัก
3. การส่งคีย์ไปพร้อมกับข้อมูลที่เข้ารหัสนั้นอาจทำให้เกิดปัญหาได้ ซึ่งถ้าทำเช่นนั้นแล้วคีย์ที่ส่งไปด้วยนั้นจะต้องถูกส่งไปด้วยความปลอดภัยสูง วิธีที่ง่ายคือการส่งคีย์ให้กับผู้รับด้วยมือของผู้ส่งเองแต่อาจทำให้ไม่สะดวกและเสียเวลา ส่วนอีกวิธีคือการส่งคีย์ไปพร้อมกับข้อมูล แต่จะต้องทำการแบ่งคีย์ออกเป็นส่วนๆ ก่อนแล้วจึงทำการส่งไปตามเส้นทางที่ต่างกัน ซึ่งแม้ว่าคีย์จะถูกดักจับได้ แต่ก็เป็นเพียงคีย์ในบางส่วนไม่สามารถรู้ถึงคีย์ที่สมบูรณ์ได้

4.1.2 ระบบการเข้ารหัสแบบไม่สมมาตร

แนวคิดของการเข้ารหัสแบบไม่สมมาตร (Asymmetric Cryptosystem) หรือการเข้ารหัสแบบคีย์สาธารณะ (Public key Encryption) ได้เกิดขึ้นจากการพยายามแก้ไขปัญหาของการเข้ารหัสแบบสมมาตร 2 ข้อด้วยกันคือ การจัดการคีย์และการพิสูจน์สิทธิ์ ลักษณะที่สำคัญของการเข้ารหัสคือการเข้ารหัสและเข้ารหัสโดยใช้คีย์คนละประเภทกันคือไพรเวตคีย์ (private key) ซึ่งเป็นคีย์ที่ผู้รับจะต้องเก็บเป็นความลับและพับลิกคีย์ (public key) เป็นคีย์ที่สามารถเปิดเผยให้ผู้อื่นทราบได้



รูปที่ 4.2 แสดงการเข้ารหัสและถอดรหัสแบบไม่สมมาตร

คีย์ทั้งสองนี้จะเป็นคีย์ที่ต่างกันดังรูป 4.2 ผู้ส่งจะต้องมีคีย์ของผู้รับซึ่งก็คือพับลิกคีย์ โดยเมื่อต้องการส่งผู้ส่งจะเข้ารหัสด้วยคีย์ของผู้รับ และผู้รับจะทำการถอดรหัสโดยไพรเวตคีย์ของตนเอง

จากระบบของการเข้ารหัสข้อมูลทั้งสองแบบที่ได้กล่าวมานั้น เราสามารถนำมาสรุปข้อดีและข้อเสียในระบบการเข้ารหัสข้อมูลแต่ละระบบได้ตามตารางที่ 4.1 โดยมีรายละเอียดดังนี้

การเข้ารหัสแบบสมมาตร	การเข้ารหัสแบบไม่สมมาตร
ข้อดี 1. การเข้ารหัสทำได้อย่างรวดเร็ว 2. สามารถสร้างได้ง่ายโดยฮาร์ดแวร์	**ข้อดี** 1.ใช้คีย์ต่างกันในการเข้ารหัสและถอดรหัส ทำให้การจัดส่งคีย์ทำได้ง่าย 2. สามารถตรวจสอบผู้ใช้ได้
ข้อเสีย 1. คีย์ของการเข้ารหัสและถอดรหัสต้องเหมือนกัน ทำให้การจัดส่งคีย์ทำได้ยาก	**ข้อเสีย** 1. ค่อนข้างช้าและต้องใช้การคำนวณอย่างมาก

ตารางที่ 4.1 แสดงข้อดีและข้อเสียในระบบของการเข้ารหัสแต่ละระบบ

สำหรับในโครงงานนี้จะใช้วิธีการเข้ารหัสข้อมูลคือ DES (Data Encryption Standard) ซึ่งเป็นระบบการเข้ารหัสแบบสมมาตร โดยทำการสร้างคีย์ DES ด้วยวิธี Diffie-Hellman ซึ่งเป็นแนวคิดในการเข้ารหัสแบบพับลิกคีย์

4.2 รูปแบบการเข้ารหัสสำหรับการสื่อสารข้อมูลในโพรโตคอลทีซีพี/ไอพี

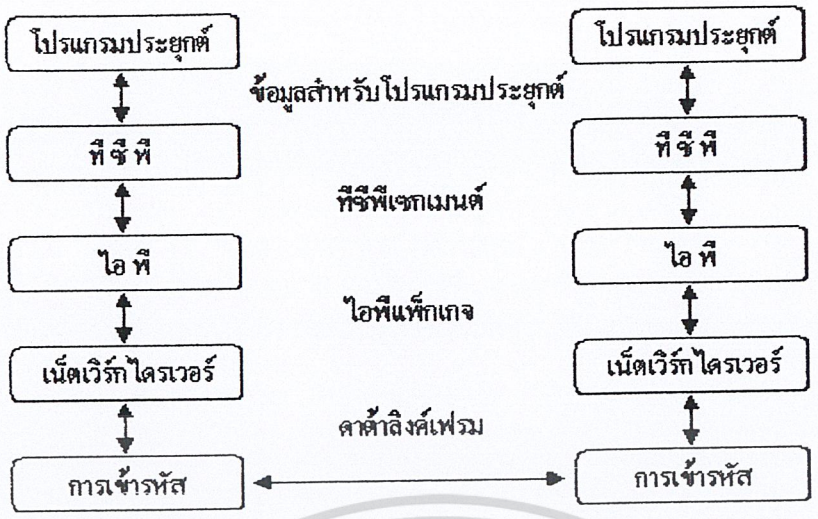
สำหรับรูปแบบการเข้ารหัสสำหรับการสื่อสารข้อมูลในโพรโตคอลทีซีพี/ไอพี แบ่งออกเป็น 4 ระดับดังต่อไปนี้

- การเข้ารหัสระดับการเชื่อมโยงข้อมูล (link level encryption)
- การเข้ารหัสระดับเครือข่าย (network level encryption)
- การเข้ารหัสระดับทรานสปอร์ต (transport level encryption)
- การเข้ารหัสระดับโปรแกรมประยุกต์ (application level encryption)

4.2.1 การเข้ารหัสระดับการเชื่อมโยงข้อมูล

การเข้ารหัสระดับการเชื่อมโยงข้อมูลเป็นการเข้ารหัสในระดับคาต้าลิงค์ โดยทั่วไปมักจะเป็นการใช้อุปกรณ์พิเศษที่เรียกว่ากล่องเข้ารหัส (encryption box) ซึ่งวิธีการนี้ถูกใช้ในระบบไซเฟอร์หรือการใช้คีไวซ์ไดร์เวอร์

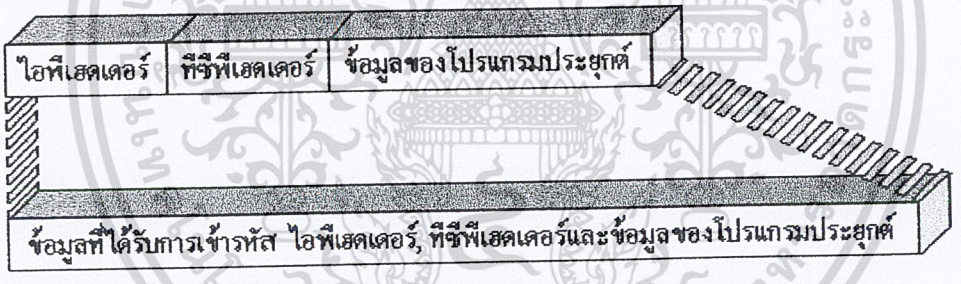
วิธีการนี้จัดเป็นวิธีการที่ปลอดภัยที่สุดเนื่องจากข้อมูลถูกเข้ารหัสทั้งหมด นั่นคือทั้งข้อมูลที่ใช้งานและข้อมูลที่เป็นโครงสร้างของโพรโตคอล แต่วิธีการนี้ก็เป็นวิธีการที่มีค่าใช้จ่ายสูง เนื่องจากอุปกรณ์สำหรับการสื่อสารข้อมูลทั้งหมดจะต้องสนับสนุนการเข้ารหัสนี้ด้วย



รูปที่ 4.3 การเข้ารหัสในระดับการเชื่อมโยงข้อมูล

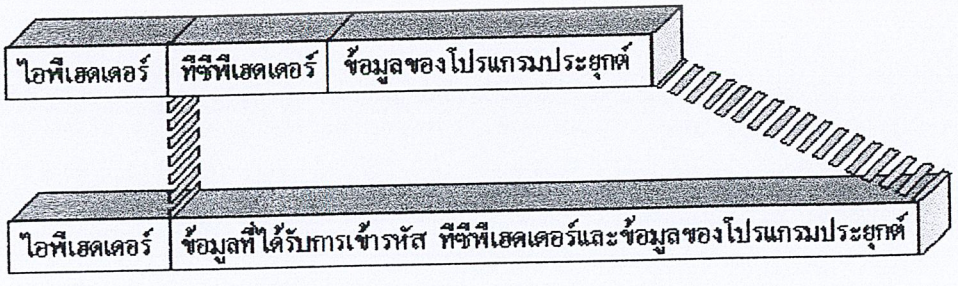
4.2.2 การเข้ารหัสระดับเครือข่าย

การเข้ารหัสข้อมูลในระดับเครือข่ายเป็นการเข้ารหัสในไอพีแพ็กเก็ต (IP Packet) โดยมีลักษณะดังรูปที่ 4.4



รูปที่ 4.4 การเข้ารหัสในระดับเครือข่าย

4.2.3 การเข้ารหัสระดับทรานสปอร์ต

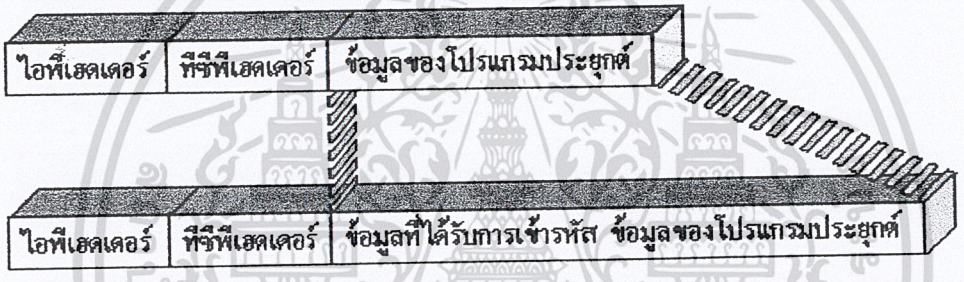


รูปที่ 4.5 การเข้ารหัสในระดับทรานสปอร์ต

การเข้ารหัสข้อมูลในระดับทรานสปอร์ตเป็นการเข้ารหัสข้อมูลในส่วนของทีซีพีเซกเมนต์ (TCP Segment) ได้แก่ทีซีพีเฮดเดอร์ (TCP Header) และข้อมูลของโปรแกรมประยุกต์ ซึ่งการเข้ารหัสในลักษณะนี้จะทำให้การส่งข้อมูลไปตามเครือข่ายสามารถทำได้โดยไม่ต้องเปลี่ยนแปลงอุปกรณ์ใดๆ เนื่องจากโครงสร้างในส่วนของไอพีไม่มีการเปลี่ยนแปลง

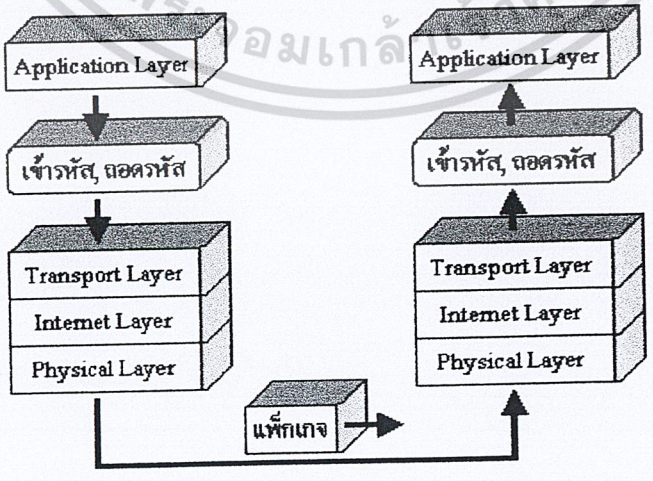
4.2.4 การเข้ารหัสระดับโปรแกรมประยุกต์

การเข้ารหัสข้อมูลในระดับโปรแกรมประยุกต์ เป็นการเข้ารหัสในลักษณะของการเข้ารหัสแบบจุดต่อจุด (end-to-end encryption) นั่นคือโปรแกรมประยุกต์จะเข้ารหัสในส่วนข้อมูลของตนเองก่อนที่จะส่งผ่านไปยังระดับล่าง ทำให้กระบวนการในระดับล่างไม่จำเป็นต้องมีการเปลี่ยนแปลงใดๆ ซึ่งวิธีการนี้ถ้าต้องการนำมาใช้กับโปรแกรมประยุกต์เดิมเช่นเทลเน็ต (telnet) หรือเอฟทีพี (FTP) จะต้องทำการแก้ไขโปรแกรมใหม่ทั้งโปรแกรมขอรับบริการและโปรแกรมสำหรับให้บริการ



รูปที่ 4.6 การเข้ารหัสในระดับโปรแกรมประยุกต์

สำหรับในโครงงานนี้ จะทำการเข้าและถอดรหัสข้อมูลในระหว่างชั้นโปรแกรมประยุกต์และชั้นทรานสปอร์ตในระบบโอเอสไอโมเดล (OSI Model) ของโพรโตคอลที่ซีพี/ไอพี



รูปที่ 4.7 การเข้ารหัสบนระบบโอเอสไอโมเดลของที่ซีพี/ไอพี

4.3 การเข้ารหัสแบบ DES

4.3.1 ประวัติและที่มาของ DES

ในปลายทศวรรษที่ 1960 บริษัทไอบีเอ็มได้จัดตั้งโครงงานวิจัยทางด้านการเข้ารหัสด้วยคอมพิวเตอร์ (Computer Cryptography) ซึ่งนำโดยฮอสท์ เฟสเทล (Horst Feistel) ซึ่งโครงงานนี้เสร็จสิ้นในปี ค.ศ.1971 ซึ่งผลงานวิจัยของโครงงานนี้ก็คือลูซิเฟอร์ (LUCIFER [FEIS73]) โดยมีลักษณะเป็นการเข้ารหัสข้อมูลเป็น บล็อกขนาด 64 บิตและใช้คีย์ขนาด 128 บิต ซึ่งต่อมาได้ถูกพัฒนาขนาดของคีย์ให้ลดลงเหลือขนาด 56 บิต

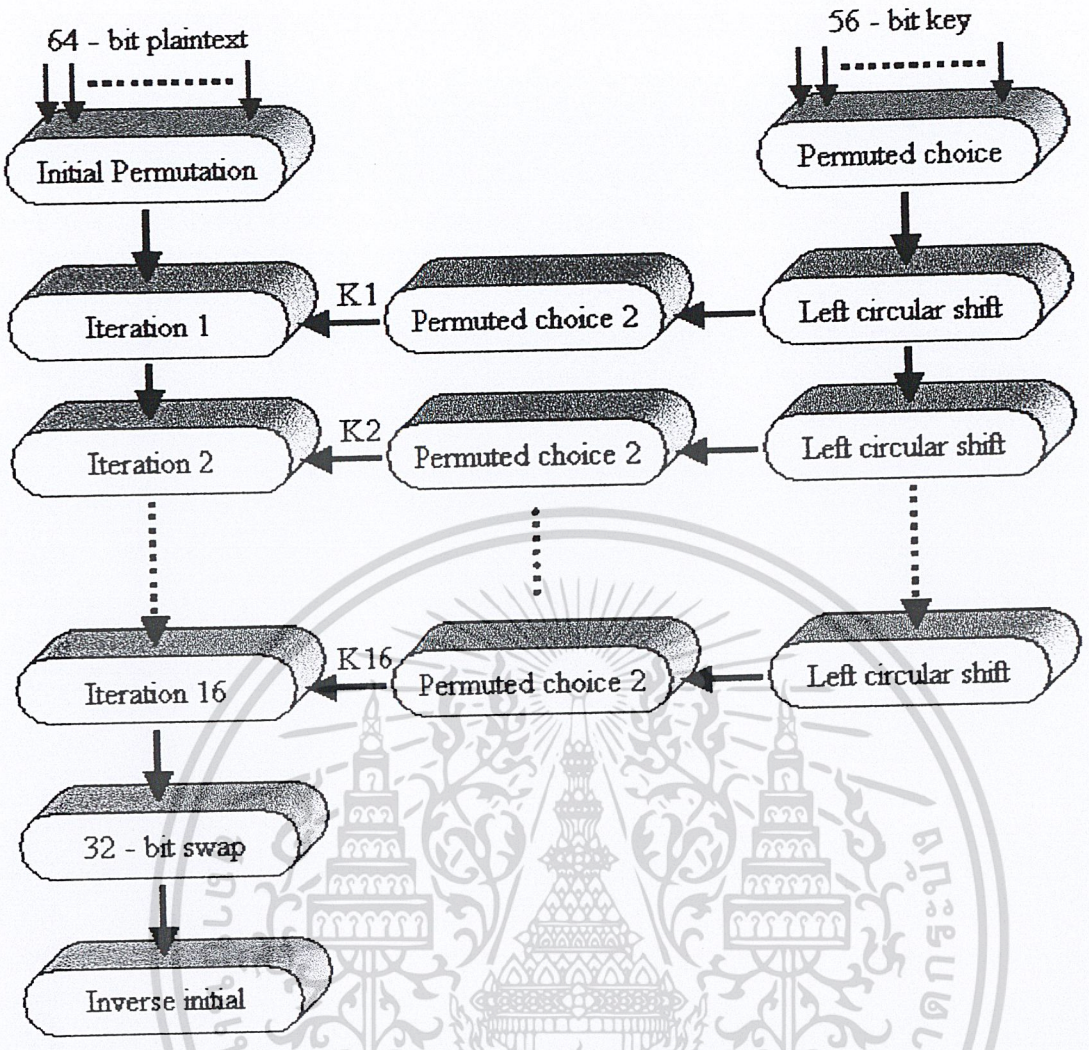
โดยอัลกอริทึมของการเข้ารหัสของลูซิเฟอร์ได้ถูกพัฒนาโดยไอบีเอ็มสำหรับ NBS (National Bureau of Standards) อัลกอริทึมนี้ได้เป็นที่รู้จักในนามของ DES (Data Encryption Standard) ถึงแม้ว่าชื่อจริงของมัน คือ DEA (Data Encryption Algorithm) ในประเทศสหรัฐอเมริกาและ DEAI (Data Encryption Algorithm-1) ในอีกหลายๆ ประเทศ

4.3.2 รายละเอียดของ DES

เป็นวิธีการเข้ารหัสที่ใช้กันอย่างแพร่หลายที่เป็นพื้นฐานบน Data Encryption Standard (DES) ที่ได้พัฒนาขึ้นในปี ค.ศ.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 บิตเข้ามูลออกไปและใช้คีย์ตัวเดียวกันนี้ในการถอดรหัส

แม้ว่า DES ถูกนำมาใช้ตั้งแต่ช่วงทศวรรษที่ 70 (ค.ศ.1960 – 1970) และได้รับการตอบรับอย่างดีจาก เหล่านักวิเคราะห์รหัส (Cryptanalysis) อย่างแพร่หลาย แต่ก็มีข้อถกเถียงกันเป็นอย่างมากถึงเรื่อง DES นั้นจะปลอดภัยได้หรือไม่และมีความปลอดภัยมากน้อยแค่ไหน แต่จนถึงในปัจจุบันเราก็ยังไม่พบจุดบกพร่องของ DES ตามเอกสารที่ตีพิมพ์เป็นสาธารณะ แม้ว่าจะใช้คีย์เพียงไม่กี่บิตก็ตาม ในทางตรงกันข้ามแนวความคิดแบบ IDEA กลับใช้คีย์แบบขนาด 128 บิต (ซึ่งมีขนาดเป็น 2 เท่าของ DES) และได้รับการตอบรับจากสาธารณะตั้งแต่ทศวรรษ 90 (ค.ศ. 1980 – 1990) IDEA มีความปลอดภัยมากกว่า DES และสามารถประมวลผลได้เร็วกว่า DES อย่างไรก็ตาม IDEA ยังต้องรอการตรวจสอบจากผู้เชี่ยวชาญถึงเรื่องช่องโหว่ของความปลอดภัย

การทำงานของ DES จะมีลักษณะคือข้อมูลที่เข้ามาในส่วนฟังก์ชันของการเข้ารหัสจะมี 2 ส่วนด้วยกันคือ ข้อมูลที่ยังไม่ถูกเข้ารหัสขนาด 64 บิตและคีย์ซึ่งมีขนาด 56 บิต ซึ่งรูปในด้านซ้ายมือจะแสดงขั้นตอนจัดการกับข้อมูลที่ยังไม่เข้ารหัส โดยสามารถแบ่งย่อยๆได้อีก 3 เฟส



รูปที่ 4.8 ขั้นตอนการทำงานของ DES

- เฟสที่ 1 จะทำการจัดการกับข้อมูลที่ไม่ได้ผ่านการเข้ารหัสขนาด 64 บิตผ่านเข้าไปยังส่วนที่เรียกว่า Initial Permutation (IP) ซึ่งจะทำการเรียบเรียงบิตใหม่เพื่อผลิตข้อมูลที่มีการสลับตำแหน่ง
- เฟสที่ 2 จะทำฟังก์ชันเดียวกัน 16 ครั้ง ซึ่งฟังก์ชันนี้รวมการทำ permutation และ substitution ซึ่งผลลัพธ์ที่ได้จากการทำทั้งหมด 16 ครั้งนี้จะได้ข้อมูลขนาด 64 บิตโดยใช้ทั้งข้อมูลที่ไม่ได้ผ่านการเข้ารหัสและคีย์ในการทำ ซึ่งข้อมูลที่มีขนาด 64 บิตที่ได้นี้แบ่งเป็น 2 ด้านคือซ้ายและขวา ทั้งหมดจะถูกสับเปลี่ยนเพื่อผลิต 64 บิตที่เป็น preoutput
- เฟสที่ 3 จะนำ preoutput ผ่านเข้าไปยังส่วนที่เรียกว่า Inverse initial permutation หรือ IP^{-1} ซึ่งทำหน้าที่อินเวอร์สตัวฟังก์ชัน initial permutation ซึ่งทั้งหมดจะผลิต 64 บิตที่เรียกว่าข้อมูลที่ผ่านการเข้ารหัสแล้ว (Ciphertext)

4.3.3 การทำ initial permutation

การทำ initial permutation และการทำ inverse initial permutation จะถูกอธิบายโดยใช้ตารางด้านล่างตามลำดับ ซึ่งจะเห็นได้ว่าฟังก์ชัน permutation ทั้งสองต่างเป็นส่วนกลับซึ่งกันและกัน 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 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

4.3.4 รายละเอียดของการทำฟังก์ชันในแต่ละรอบ

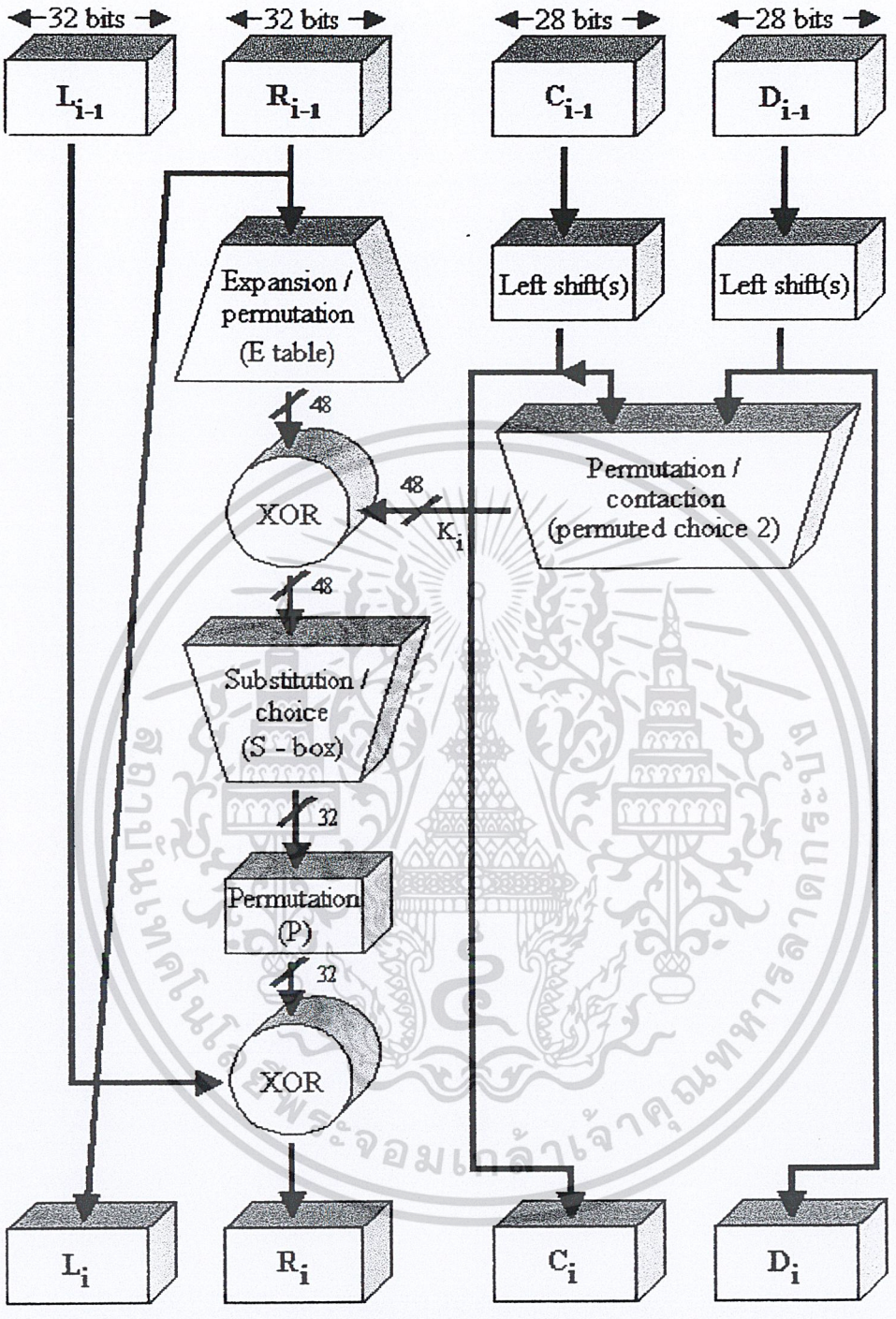
จากข้อมูลที่เข้ามาที่มีขนาด 64 บิตจะทำการแบ่งข้อมูลเป็น 2 ส่วนด้วยกัน ส่วนละขนาด 32 บิต (แบ่งเป็นซ้ายกับขวา) ซึ่งกระบวนการทำในแต่ละครั้งสามารถสรุปเป็นสูตรได้ดังนี้

$$L_I = R_{I-1}$$

$$R_I = L_{I-1} \oplus f(R_{I-1}, K_I)$$

โดย $\oplus$ หมายถึงการทำ XOR function

จากสูตรในข้างต้นจะเห็นได้ว่า 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 4.9



รูปที่ 4.9 แสดงการเข้ารหัส DES ในแต่ละครั้ง (ทำทั้งหมด 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

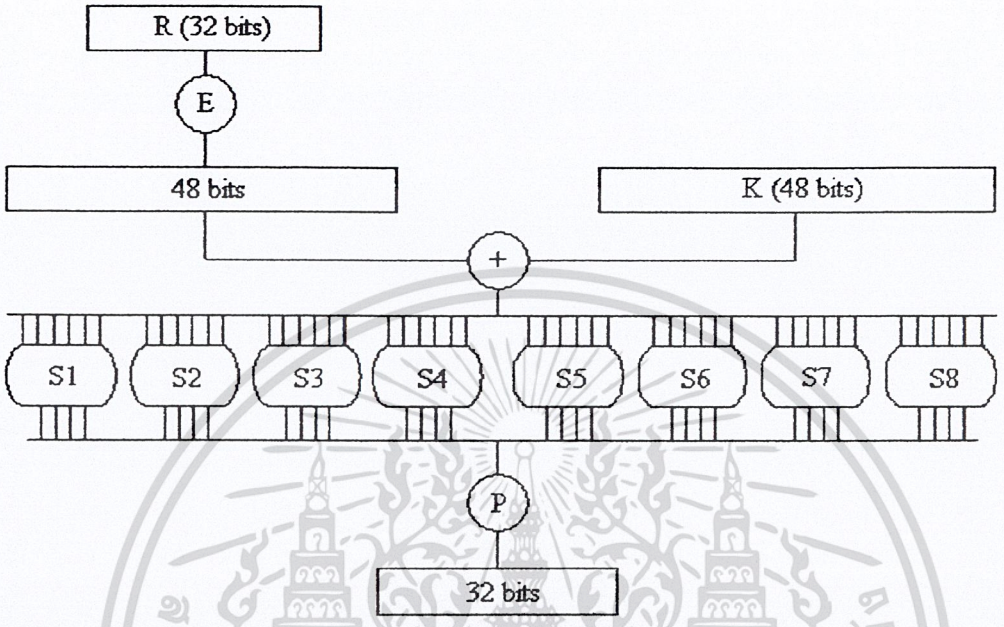
(d) Permutation Function (P)

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 4.2 แสดง permutation ของ DES

โดยคีย์ K_i ที่ใช้ในแต่ละรอบจะมีขนาด 48 บิตและอินพุทด้านขวา (R) มีขนาด 32 บิต ดังนั้น จึงต้องมีการขยายขนาดจาก 32 บิตให้เป็น 48 บิต โดยใช้ตารางที่ได้มีการกำหนด permutation และ

expansion ซึ่งรวมทั้งการจำลอง 16 บิตที่เพิ่มขึ้นมาของด้านขวา ซึ่งจะนำผลที่ได้ที่มีขนาด 48 บิตจะถูก XOR กับ K_i โดยจะนำผลที่ได้ผ่านไปยังฟังก์ชันที่เรียกว่า Substitution และ Permutation ที่สามารถผลิตผลลัพธ์ที่มีขนาด 32 บิต



รูปที่ 4.10 แสดงการคำนวณ f(R,K)

โดย Substitution จะประกอบด้วยเซตของ S – box 8 อัน (S_1 - S_8) ซึ่ง S – box จะมีอินพุทขนาด 6 บิตและผลิตเอาต์พุทขนาด 4 บิต ซึ่งจากตารางที่ 4.3 จะแสดง DES S – box โดยมีวิธีการในการแปลงอินพุทขนาด 6 บิตให้กลายเป็นเอาต์พุทขนาด 4 บิตดังนี้คือ การนำบิตแรกและบิตสุดท้ายของอินพุทมาทำเป็นตำแหน่งของแถวและนำ 4 บิตตรงกลางมาเป็นตำแหน่งของคอลัมน์เช่น S_1 มีค่าเท่ากับ 011011 เราจะนำบิตแรกและบิตสุดท้ายซึ่งก็คือ 0 และ 1 มาเป็นตำแหน่งของแถวจะได้แถวที่ 01 และ 4 บิตตรงกลางที่เหลือคือ 1101 จะได้ตำแหน่งของคอลัมน์คือ คอลัมน์ที่ 13 ดังนั้นค่าที่ตำแหน่งแถวที่ 1 และคอลัมน์ที่ 13 ในตารางคือ 0101

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	0	S1
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	

Column Number

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Box
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	S2
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	

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	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	S3
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	
2	10	6	9	0	12	11	7	12	15	1	2	14	5	2	8	4	S4
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	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	S5
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	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	S6
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	S7
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	

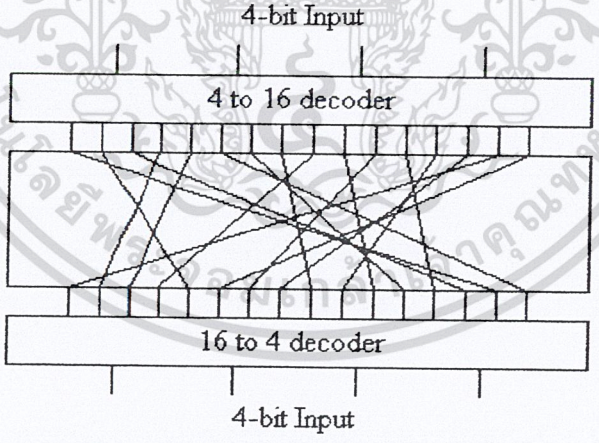
Column Number

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Box
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	S8
3	2	1	14	7	4	10	3	13	15	12	9	0	3	5	6	11	

ตารางที่ 4.3 แสดงขั้นตอนใน S- box ทั้ง 8 ชุด

4.3.5 การสร้างคีย์

ในรูปที่ 4.9 ซึ่งแสดงถึงการทำงานในแต่ละรอบของ DES จะเห็นได้ว่าคีย์ที่ใช้มีขนาด 56 บิตเป็นอินพุทในการทำ permutation โดยในตาราง Permuted Choice One เริ่มแรกจะทำการแบ่ง 56 บิตเป็น 2 ส่วนเท่าๆ กันส่วนละ 28 บิตโดยให้ชื่อในแต่ละส่วนว่า C กับ D ซึ่งในแต่ละรอบ (ทั้งหมด 16 รอบ) จะมีการทำ circular left shift ในแต่ละส่วนของ C และ D หรือทำ rotation โดยในแต่ละรอบจะมีการกำหนดว่าจะให้เลื่อนไปกี่บิตดังตาราง ซึ่งค่าที่ถูกเลื่อนจะกลายเป็นอินพุทของการทำให้รอบถัดไปและเป็นอินพุทของการทำ Permutation Choice Two ดังในตาราง หลังจากการทำ Permutation Choice Two แล้วจะได้เอาต์พุทขนาด 48 บิต ซึ่งเป็นอินพุทของ $f(R_{I-1},K_I)$



รูปที่ 4.11 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 4.4 แสดงการสร้างคีย์

4.3.6 การถอดรหัสข้อมูล DES

กระบวนการถอดรหัสโดยใช้ DES นั้นเหมือนกับขั้นตอนในการเข้ารหัส ซึ่งมีขั้นตอนในการทำงานดังนี้คือ นำข้อมูลที่ผ่านการเข้ารหัสแล้ว (Ciphertext) มาเป็นอินพุทแต่จะมีการใช้คีย์ (K_I) ที่มีลำดับย้อนกลับกับคีย์ที่ใช้ในการเข้ารหัสเช่น K_{16} , K_{15} ... เป็นคีย์แรกและคีย์ถัดไปในการเข้ารหัสแทนดังรูปที่ 4.12 ด้านซ้ายมือจะเป็นขั้นตอนการเข้ารหัสและด้านขวามือเป็นขั้นตอนในการถอดรหัส

เราจะแสดงถึงผลลัพธ์ของขั้นตอนแรกในการกระบวนการถอดรหัส ซึ่งจะเท่ากับ 32 บิตที่ถูกสับเปลี่ยนจากอินพุทของการทำทั้งหมด 16 รอบของการเข้ารหัส เริ่มจาก

$$L16 = R15$$

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

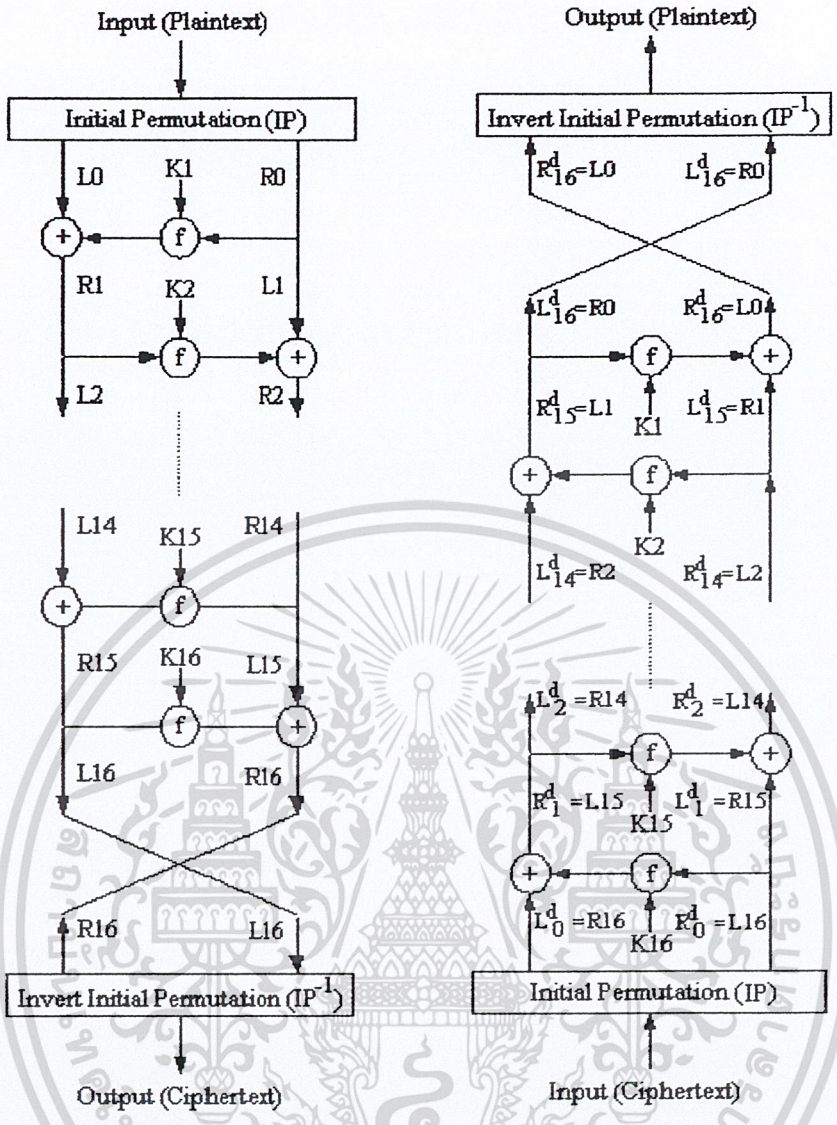
ในด้านการถอดรหัส

$$Ld1 = Rd0 = L16 = R15$$

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

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

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



รูปที่ 4.12 แสดงคีย์และขั้นตอนการเข้าและออดรหัส DES

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 จะได้เอาต์พุทของขั้นตอนแรกในการถอดรหัสคือ L15||R15 ซึ่งเราสามารถเขียนเป็นสมการในการถอดรหัสได้ดังนี้คือ

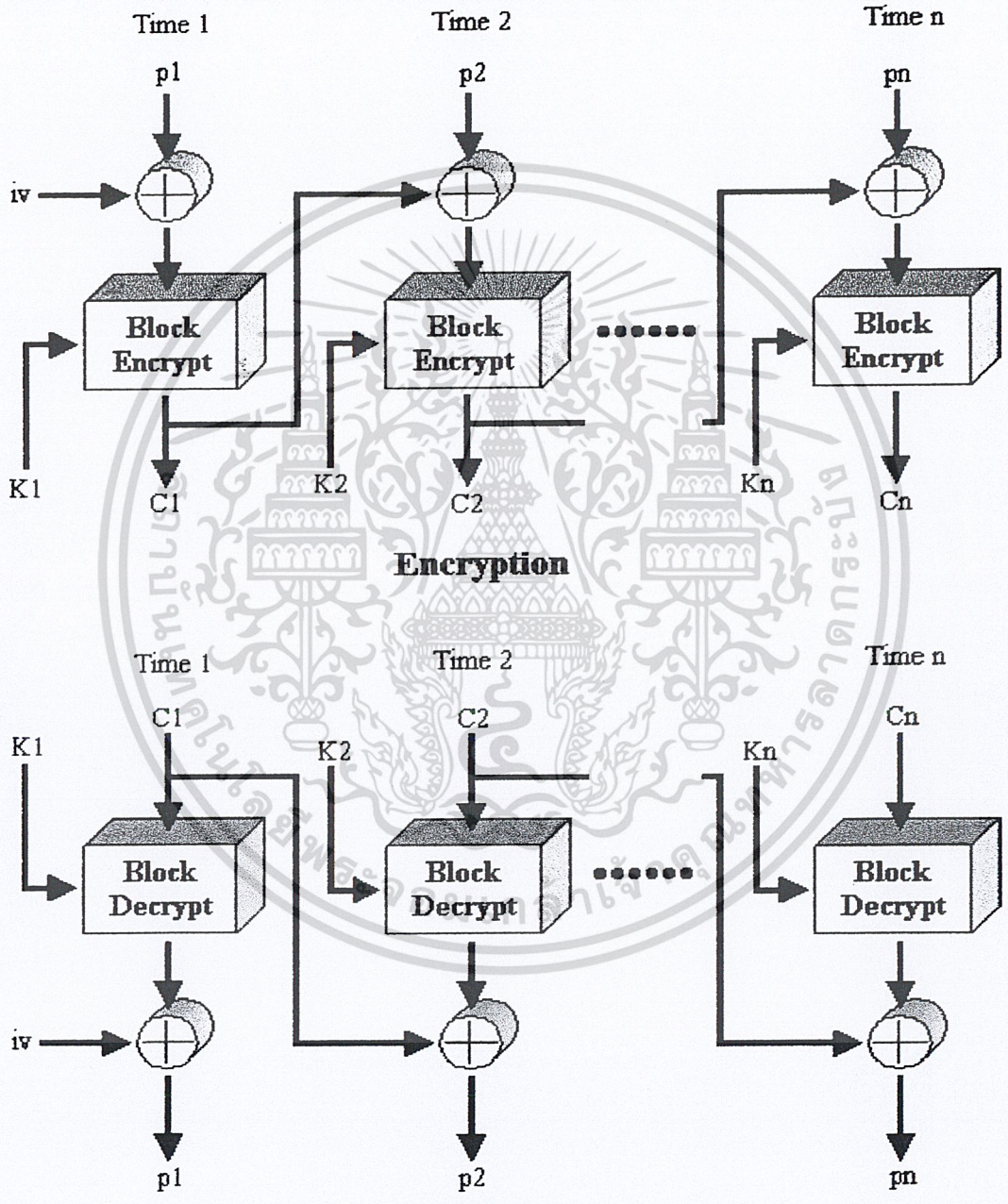
$$Ri^{-1} = Li$$

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

ซึ่งผลสุดท้ายเอาต์พุทที่ได้จากขั้นตอนสุดท้ายในการถอดรหัสคือ R0||L0 และนำไปสู่ขั้นตอนการทำ Inverse Permutation เราจะได้ข้อมูลที่ส่งมา (Plaintext) ดังสมการ

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

4.3.7 โหมด CBC



รูปที่ 4.13 แสดงการเข้าและถอดรหัส DES ในโหมด CBC

โหมด CBC (Cipher Block Chaining) เป็นวิธีการเข้ารหัสที่พัฒนามาจาก DES ช่วยทำให้ข้อมูลที่ส่งมีความปลอดภัยมากขึ้น โดยมีวิธีการคือจะนำข้อมูลที่ผ่านการเข้ารหัสของข้อมูลตัวก่อนมา XOR กับข้อมูลที่ยังไม่ได้เข้ารหัสของตัวถัดไปก่อนจะทำการเข้ารหัสแบบ DES ตามปกติ

ในการถอดรหัสข้อมูลจะทำเช่นเดียวกับการถอดรหัสข้อมูล DES ตามปกติแต่จะนำผลที่ได้จากการถอดรหัสมา XOR กับข้อมูลที่ผ่านการเข้ารหัส (Ciphertext) ตัวก่อนหน้านี้เพื่อจะได้ข้อมูลจริง (Plaintext) ดังสมการ

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

สมการในการถอดรหัส

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

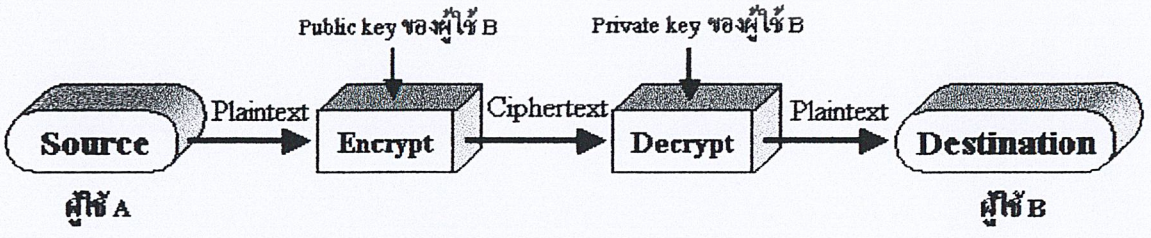
$$Dk[Cn] = Cn^{-1} \oplus Pn$$

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

สังเกตเห็นว่าบล็อกแรกของข้อมูลที่ผ่านการเข้ารหัส (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

4.4 การเข้ารหัสแบบ RSA

การเข้ารหัสข้อมูลแบบ RSA ถูกคิดค้นขึ้นในปี ค.ศ.1977 โดยที่ชื่อ RSA ได้มาจากอักษรตัวแรกของนามสกุลของผู้ร่วมกันคิดค้นคือ Ron River, Adi Shamir และ Leonard Adleman ซึ่งเป็นวิธีการเข้ารหัสแบบคีย์สาธารณะที่เรียกว่าพับลิกคีย์ (Public-key Cryptosystem) เพื่อแก้ไขปัญหาการทำให้คีย์เป็นความลับ (Secret–key Cryptosystem) โดยสมาชิกแต่ละคนจะต้องมีคีย์ 2 ชนิดคือคีย์ส่วนตัวหรือไพรเวตคีย์ (Private key) และคีย์สาธารณะหรือพับลิกคีย์ (Public key) โดยไพรเวตคีย์จะถูกเก็บไว้เป็นความลับ ส่วนพับลิกคีย์นั้นจะเปิดเผยให้บุคคลใดก็ได้ที่ต้องการส่งสารให้แก่ตน ซึ่งมีหลักการการทำงานว่าข้อมูลที่ถูกเข้ารหัสด้วยพับลิกคีย์ของบุคคลใดจะสามารถถูกถอดรหัสด้วยไพรเวตคีย์ของบุคคลนั้นได้เท่านั้น



รูปที่ 4.14 แสดงขั้นตอนการทำ Public–key Cryptosystem

4.4.1 หลักการทำงานของ RSA

ถ้าให้ p และ q เป็นจำนวนเฉพาะที่มีค่ามากๆ 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

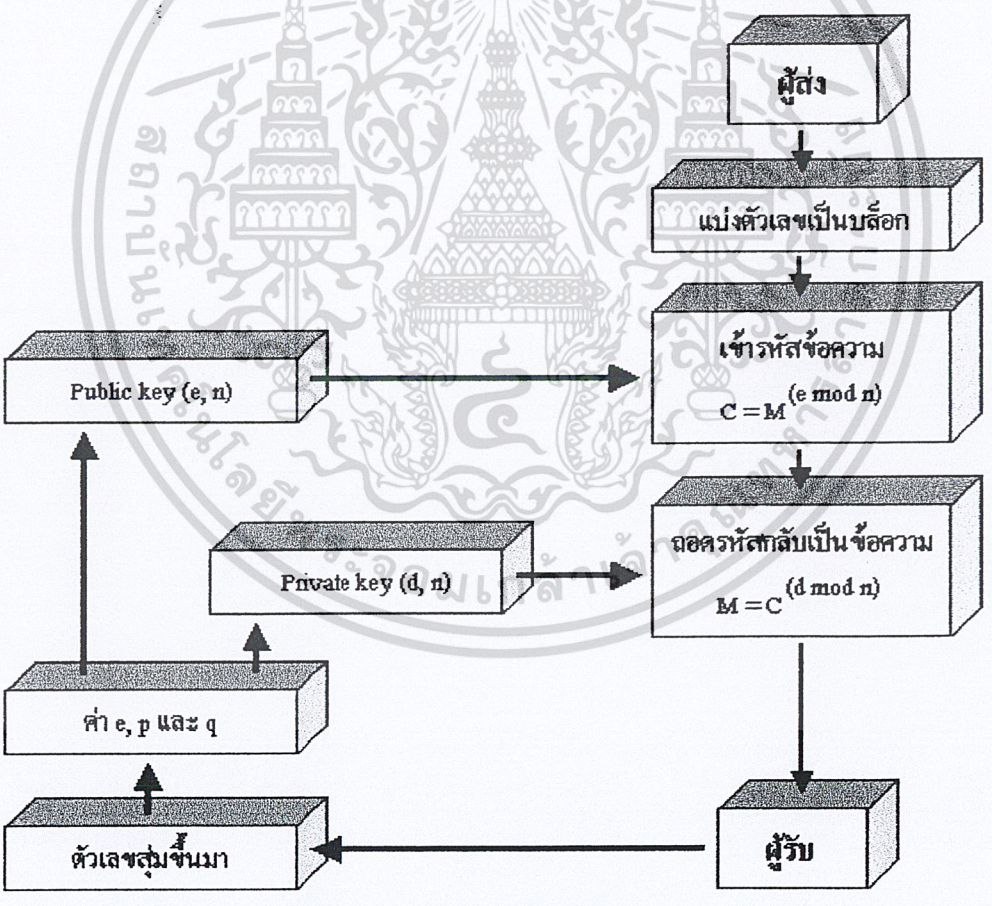
$$e * d \bmod (p-1)(q-1) = 1 \qquad \text{.....(1)}$$

ในรหัส RSA นั้น (n, e) คือพับลิกคีย์ ส่วน d คือไพรเวตคีย์ เมื่อได้ค่าเหล่านี้แล้ว p, q จะต้องเก็บเป็นความลับหรือถูกทำลายในทันที

ถ้าผู้ใช้ A ต้องการส่งข้อเอกสารส่วนตัว m ไปให้ผู้ใช้ B แล้วผู้ใช้ A จะสร้างรหัสลับ c ของ m ได้โดยการให้ $c = m^e \bmod n$ เมื่อ (n, e) เป็นพับลิกคีย์ของผู้ใช้ B เมื่อผู้ใช้ B ได้รับเอกสารรหัสลับ c ผู้ใช้ B จะถอดรหัสเพื่ออ่านเอกสาร m ได้ด้วย d เพราะความสัมพันธ์ในสมการ (1) ระหว่าง e, d และ n จะทำให้

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

และเพราะมีแต่ผู้ใช้ B เท่านั้นรู้ค่า d ทำให้ผู้ใช้ B เท่านั้นที่จะสามารถถอดรหัสได้



รูปที่ 4.15 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}

4.4.2 การทำลายรหัส RSA

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 นั้นไม่ง่ายเลยสำหรับ n มีค่ามาก ดังนั้นสำหรับข้อมูลที่มีความสำคัญมากอาจจำเป็นต้องใช้ n ที่มีค่ามากถึง 700 หรือ 1000 ก็ได้

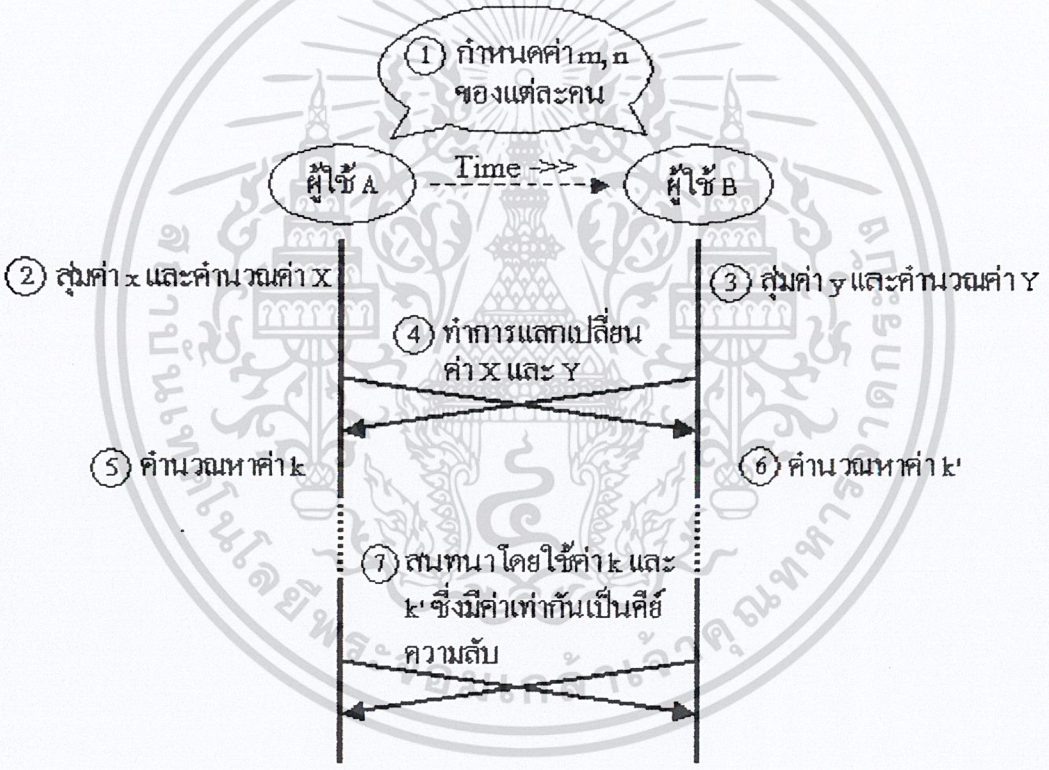
จะเห็นได้ว่าไม่ว่าการเข้าหรือถอดรหัส RSA ก็จำเป็นต้องใช้การยกกำลังในระบบโมดูโล ซึ่งการยกกำลังนั้นสามารถทำได้โดยการใช้วงจรคูณระบบโมดูโลมาต่ออนุกรมกัน ดังนั้นความเร็วของทั้งการเข้าและถอดรหัสจึงขึ้นกับความเร็วของวงจรคูณระบบโมดูโลเป็นอย่างมาก ซึ่งเป็นจุดอ่อนข้อหนึ่งของ RSA เมื่อเปรียบเทียบกับคีย์ความลับเช่น DES เพราะปัจจุบันการคูณในระบบโมดูโลยังค่อนข้างยุ่งยากเมื่อเปรียบเทียบกับการแทนค่าหรือสลับตำแหน่งใน DES ได้มีการประมาณกันว่าสำหรับ n ที่มีความยาว 512 บิต การเข้ารหัสแบบ DES จะเร็วกว่า RSA ประมาณ 100 เท่าถ้าเราทำการเข้ารหัสด้วยซอฟท์แวร์ และอาจเร็วกว่าถึง 1000 ถึง 10000 เท่าหากทำการเข้ารหัสด้วยฮาร์ดแวร์ โดยทั่วไปแล้วเราต้องการให้การเข้ารหัสเร็วกว่าการถอดรหัสดังนั้นเราจึงมักเลือกให้ e มีค่าน้อยกว่า d และยิ่งกว่านั้นเรามักให้ e ของสมาชิกทุกคนมีค่าเดียวกันเพื่อให้ฮาร์ดแวร์ของวงจรเข้าสำหรับสมาชิกแต่ละคนมีลักษณะคล้ายกัน

เนื่องจาก DES และ RSA มีข้อดีข้อเสียที่แตกต่างกันจึงไม่จำเป็นว่ารหัสชนิดใดชนิดหนึ่งจะเหมาะสมในทุกสถานการณ์ โดยทั่วไปแล้ว DES จะถูกใช้ในการเข้ารหัสข้อมูลที่มีขนาดใหญ่เพราะรวดเร็วกว่าในขณะที่ RSA จะถูกใช้ในระบบสื่อสารที่ไม่ยาวนักแต่ต้องการความปลอดภัยสูง ในบางครั้ง RSA ยังถูกใช้ร่วมกับ DES เพื่อเสริมจุดเด่นซึ่งกันและกัน เช่นตัวเอกสารจริงจะถูกเข้ารหัสด้วย DES โดยที่คีย์รหัส DES จะถูกเข้ารหัสด้วย RSA แล้วส่งไปด้วยกันหรือส่งไปก่อน แต่ในบางครั้ง DES อย่างเดียวก็เพียงพอแล้วหากการแลกเปลี่ยนคีย์ทำได้อย่างปลอดภัยพอหรือในกรณีเมื่อผู้ส่งและผู้รับเป็นบุคคลเดียวกัน เช่น ฮาร์ดดิสก์ในคอมพิวเตอร์ส่วนตัวหรือข้อมูลส่วนตัวในสมาร์ตการ์ด

4.5 การสร้างคีย์ความลับด้วยวิธี Deffie-Hellman

Deffie-Hellman ซึ่งถือกำเนิดขึ้นในปี ค.ศ.1976 โดย Whitfield Diffie และ Martin Hellman เป็นวิธีที่ใช้ในการแลกเปลี่ยนค่าระหว่างผู้ใช้งาน 2 คนแล้วนำค่าทั้งสองมาสร้างเป็นคีย์ความลับสำหรับใช้ในการเข้ารหัสข้อมูลในการสื่อสารระหว่างกัน โดยลำดับขั้นตอนในการสร้างคีย์ความลับด้วยวิธี Deffie-Hellman ต่อไปนี้จะแสดงถึงการสร้างคีย์ความลับระหว่างผู้ใช้ A กับผู้ใช้ B ซึ่งมีขั้นตอนดังนี้

1. ผู้ใช้ A และผู้ใช้ B กำหนดค่า m และ n โดยที่ $1 < n < m$ เลขทั้งสองไม่จำเป็นต้องปกปิด
2. ผู้ใช้ A สุ่มตัวเลขที่มีค่ามากๆ มาตัวหนึ่งกำหนดให้เป็นค่า x แล้วหาค่า $X = n^x \bmod m$ และให้ผู้ใช้ A เก็บค่า x เอาไว้เป็นความลับ
3. ให้ผู้ใช้ B ทำเหมือนกับผู้ใช้ A ในข้อ 2 สุ่มตัวเลขที่มีค่ามากๆ มาตัวหนึ่งกำหนดให้เป็นค่า y แล้วหาค่า $Y = n^y \bmod m$ และให้ผู้ใช้ B เก็บค่า y เป็นความลับ
4. ให้ผู้ใช้ A และผู้ใช้ B ทำการแลกค่า X และ Y กัน
5. ผู้ใช้ A คำนวณหาค่า $k = Y^x \bmod m$
6. ผู้ใช้ B คำนวณหาค่า $k' = X^y \bmod m$
7. ผู้ใช้ A และผู้ใช้ B ใช้ค่า k และ k' ซึ่งมีค่าเท่ากันคือ $n^{xy} \bmod m$ เป็นคีย์ความลับในการใช้งานร่วมกัน



รูปที่ 4.16 ขั้นตอนการแลกเปลี่ยนคีย์ด้วยวิธี Deffie-Hellman

ในการสร้างคีย์ด้วยวิธี Deffie-Hellman นี้ก็เหมือนกับการใช้งานพับลิกคีย์และไพรเวตคีย์นั่นเอง ซึ่งเราสังเกตได้ว่าค่า x และ y จะเป็นค่าที่ถูกเก็บไว้กับผู้ใช้เท่านั้นซึ่งก็คือไพรเวตคีย์ ส่วนค่า X และ Y เป็นค่าที่ส่งไปให้กับผู้ใช้ที่ต้องการทำการติดต่อด้วยซึ่งก็คือพับลิกคีย์นั่นเอง โดยค่าคีย์ความลับที่จะใช้ในการนำมาใช้ในการเข้ารหัสระหว่างผู้ใช้งานทั้งสองฝั่งนั้น จะเป็นการนำค่าไพรเวตคีย์ของตนและพับลิกคีย์ของผู้ใช้ที่ต้องการติดต่อด้วยมาสร้างเป็นค่า k และ k' ซึ่งมีค่าเท่ากันมาใช้เป็นคีย์ความลับในการติดต่อกัน

ค่า k และ k' ที่คำนวณได้นี้นอกจากผู้ใช้ A และผู้ใช้ B แล้วบุคคลอื่นจะไม่สามารถหาค่าได้ เพราะค่าที่บุคคลอื่นสามารถทราบได้มีเพียงค่า m, n, X และ Y โอกาสที่จะหาค่า x จากค่า X หรือค่า y จากค่า Y นั้นทำได้ด้วยการหาอินเวอร์สของ X ซึ่งเรียกว่า discrete logarithm ซึ่ง discrete logarithm นี้ไม่ได้คำนวณหาได้โดยง่าย และในการคำนวณ discrete logarithm ในบางค่า X หรือ Y อาจจะไม่มีคำตอบเลยก็เป็นได้

บทที่ 5
โพรโตคอล ICQ

โพรโตคอล ICQ คือการแสดงรายละเอียดขั้นตอนการติดต่อสื่อสารของโปรแกรม ICQ ซึ่งขณะนี้ยังไม่ถูกกำหนดให้เป็นมาตรฐาน แต่กำลังจะถูกกำหนดให้เป็นมาตรฐานในภายหน้า เนื่องมาจากมีผู้นิยมใช้เป็นจำนวนมากเพราะรูปแบบและการใช้งานโปรแกรม ICQ ที่ง่ายไม่ยุ่งยากเกินไปนัก สำหรับการค้นหาข้อมูลโพรโตคอลในการจัดทำโครงงานนี้ค้นหาจากเว็บไซต์ http://www.stricq.com/icqv8 และ http://www.stud.uni-karlsruhe.de/~uck4/ICQ เป็นต้น แต่ข้อมูลที่ค้นหาได้มานั้นยังไม่ครบถ้วนสมบูรณ์ทำงานได้ครบทุกฟังก์ชันเต็มความสามารถของ ICQ แต่ก็เพียงพอสำหรับการรับส่งข้อความระหว่างผู้ใช้ ICQ ได้

ในปัจจุบันโพรโตคอล ICQ จะใช้โพรโตคอลพื้นฐานของเครือข่ายอินเทอร์เน็ตคือ TCP ทั้งในส่วนการติดต่อสื่อสารระหว่างเซิร์ฟเวอร์ของ ICQ กับโปรแกรม ICQ ฝั่งไคลเอ็นต์และการติดต่อสื่อสารระหว่างโปรแกรม ICQ ฝั่งไคลเอ็นต์กับโปรแกรม ICQ ฝั่งไคลเอ็นต์ โดยการสื่อสารกับเซิร์ฟเวอร์ในช่วงแรกของการเริ่มโปรแกรม ICQ และต่อจากนั้นจะเป็นครั้งคราวหรือเมื่อผู้ใช้ทำการขอใช้บริการกับเซิร์ฟเวอร์ ส่วนการติดต่อระหว่างโปรแกรม ICQ ฝั่งไคลเอ็นต์กับโปรแกรม ICQ ฝั่งไคลเอ็นต์ในจะเป็นการสื่อสารโดยตรง ยกเว้นหากมีปัญหาในการเชื่อมต่อระหว่างโปรแกรม ICQ ฝั่งไคลเอ็นต์จึงจะทำการส่งผ่านเซิร์ฟเวอร์ของ ICQ แทน

5.1 เวอร์ชันของโพรโตคอล ICQ

โพรโตคอลที่ใช้ในโปรแกรม ICQ ที่ใช้งานอยู่ในปัจจุบันนี้ยังไม่มีมาตรฐานกำหนดขึ้น แต่ก็ได้มีการพัฒนาโพรโตคอล ICQ อยู่เรื่อยๆ เพื่อให้มีความสามารถที่เหมาะสมและกำหนดเป็นมาตรฐานได้ โดยรายละเอียดของโพรโตคอลแต่ละเวอร์ชันในอดีตจนถึงปัจจุบันมีดังต่อไปนี้

- เวอร์ชัน 1 มีความสามารถน้อยและไม่เป็นที่นิยม ซึ่งเซิร์ฟเวอร์ของ ICQ ไม่สนับสนุนมานานแล้ว
- เวอร์ชัน 2 ถูกพัฒนาให้มีความสามารถมากขึ้น เป็นที่นิยมใช้และถูกใช้มาเป็นเวลานาน แต่ในขณะนี้ เซิร์ฟเวอร์ไม่สนับสนุนแล้ว
- เวอร์ชัน 3 และ 4 ออกมาใกล้เคียงกันมากซึ่งอาจเรียกว่าโพรโตคอลเวอร์ชัน 3/4 โดยมีการเพิ่มคำสั่งมากขึ้นจากโพรโตคอลเวอร์ชัน 2 และเพิ่มส่วนของการตรวจสอบความถูกต้องของข้อมูลที่รับส่ง โดยในโปรแกรม ICQ99a จะรองรับการใช้งานของโพรโตคอลเวอร์ชัน 2 และ 3/4
- เวอร์ชัน 5 โพรโตคอลจะมีความเสถียรมากกว่าและมีคำสั่งมากขึ้น โดยในโปรแกรม ICQ99b จะรองรับการทำงานของโพรโตคอลเวอร์ชัน 2, 3/4 และ 5

- เวอร์ชัน 6 จะเพิ่มคำสั่งให้มากขึ้นและสามารถเก็บและเรียกคอนแทคลิสต์ผู้ใช้งานได้ ทำให้เมื่อเราทำการเรียกโปรแกรมจากเครื่องใดก็ตามคอนแทคลิสต์ก็จะเหมือนกัน เพราะสามารถเรียกจากเซิร์ฟเวอร์ได้เป็นต้น แต่โพรโตคอลเวอร์ชัน 6 ก็ออกมาใช้ได้ไม่นาน โดยโพรโตคอลเวอร์ชันนี้ถูกใช้ในช่วงสั้นๆ ใน ICQ2000b
- เวอร์ชัน 7 และ 8 โพรโตคอลเวอร์ชัน 7 นี้เริ่มมีความสมบูรณ์มากขึ้น แต่ก็ใช้อยู่ไม่นาน โพรโตคอลเวอร์ชัน 8 ก็ออกมาซึ่งมีคุณสมบัติใกล้เคียงกัน บางคนจึงเรียกว่าเป็นเวอร์ชัน 7/8 ซึ่งปัจจุบันโพรโตคอลเวอร์ชัน 7 และ 8 ถูกใช้งานมากที่สุดในเวลานี้ โดยโพรโตคอลเวอร์ชันนี้เริ่มใช้ใน ICQ2001b เป็นต้นไป โดยปัจจุบันก็ยังเป็นโพรโตคอลที่ใช้ในโปรแกรม ICQ2003a

บางคนอาจเข้าใจผิดระหว่างเวอร์ชันของโพรโตคอล ICQ และเวอร์ชันของโปรแกรม ICQ ซึ่งมีความแตกต่างกันดังนี้

- การพัฒนาเวอร์ชันของโปรแกรม ICQ ฝั่งไคลเอ็นต์เป็นการเพิ่มความสามารถและแก้ไขปัญหาของโปรแกรมให้รองรับกับเวอร์ชันโพรโตคอล ICQ ใหม่ๆ เช่น ICQ 2000b v4.60 Build #3278, ICQ 2001b v.5.18 Build 3659 และ ICQ 2003a Build 3777 เป็นต้น
- ส่วนการพัฒนาเวอร์ชันของโพรโตคอล ICQ เป็นการพัฒนาความสามารถในการสื่อสารระหว่างผู้ใช้ ICQ ให้มีเสถียรภาพความหลากหลายมากขึ้น

5.2 ลักษณะของข้อมูลสำหรับการอ้างอิง

ลักษณะของข้อมูลที่อ้างถึงในโครงงานนี้มีรูปแบบที่แตกต่างๆ กัน แต่ละรูปแบบนั้นมีชื่อเรียกเฉพาะและเพื่อความเข้าใจตรงกันจึงกำหนดให้ลักษณะของข้อมูลมีดังนี้

ชนิดของข้อมูล	คำอธิบาย
BYTE	ข้อมูลขนาด 1 ไบต์ หรือ 8 บิต
WORD	ข้อมูลขนาด 2 ไบต์ หรือ 16 บิต
DWORD	ข้อมูลขนาด 4 ไบต์ หรือ 32 บิต
VARIES	ข้อมูลที่มีขนาดไม่คงที่ ซึ่งมักจะกำหนดขนาดของข้อมูลไว้ที่บิตแรก เพื่อสามารถกำหนดขนาดของตัวแปรสำหรับรองรับข้อมูลได้

ตารางที่ 5.1 แสดงรูปแบบข้อมูลสำหรับการอ้างอิง

ลักษณะของข้อมูลที่รับและส่งเป็นแบบ little endian ordering เป็นการนำข้อมูลมาสลับตำแหน่งไบต์เสียก่อนจึงจะนำมาใช้ได้อย่างปกติมิฉะนั้นการคำนวณจะผิดพลาดอย่างมาก โดยส่วนมากการทำ little endian ordering จะพบในการส่งแพ็กเก็ตที่มีการระบุหมายเลขประจำตัวผู้ใช้งาน (UIN) ติดตามไปด้วย

ค่าในเลขฐานสิบตามชนิดของข้อมูล	ค่าในการรับส่งปกติในระบบเลขฐานสิบหก	little endian ordering
WORD w = 10	00 0A	0A 00
DWORD x = 12345678	00 BC 61 4E	4E 61 BC 00

ตารางที่ 5.2 แสดงตัวอย่างการแปลงข้อมูลแบบ little endian ordering

5.3 การติดต่อระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์

จากที่ได้กล่าวมาแล้วนั้นว่าการทำงานของโปรแกรม ICQ และโปรแกรมรับส่งสารด่วนอื่นๆ จะประกอบด้วยโปรแกรมไคลเอ็นต์ซึ่งผู้ใช้งานจะต้องทำการติดตั้งไว้ที่เครื่องของผู้ใช้เอง และเซิร์ฟเวอร์กลางสำหรับให้บริการผู้ใช้งานหรือโปรแกรมไคลเอ็นต์ ซึ่งในส่วนนี้จะนำเสนอการสื่อสารกับระหว่างเซิร์ฟเวอร์กลางและโปรแกรมไคลเอ็นต์

5.3.1 เซิร์ฟเวอร์ของ ICQ

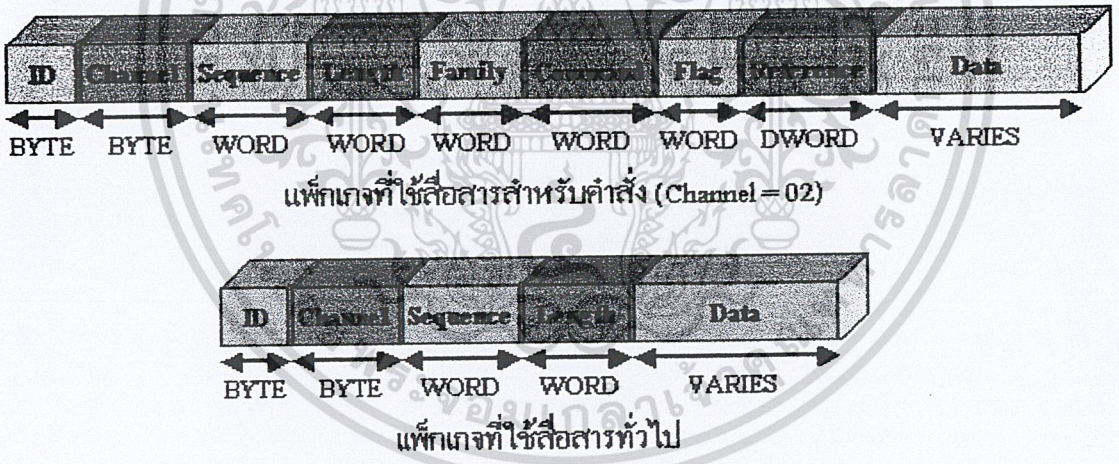
เซิร์ฟเวอร์ของ ICQ นั้นได้ถูกพัฒนาขึ้นตามเวอร์ชันของโปรแกรม ICQ ไคลเอ็นต์และเวอร์ชันของโพรโตคอล ทำให้ความสามารถของเซิร์ฟเวอร์ ICQ ปัจจุบันมีความสามารถมาก และสามารถรองรับบริการต่างๆ ได้ดังนี้

1. ให้ข้อมูลสำหรับการสื่อสารของผู้ใช้งาน ICQ อื่นๆ ตามที่ต้องการ เช่น
   - หมายเลข IP และหมายเลขพอร์ต
   - ข้อมูลส่วนตัวของผู้ใช้งาน (User Details)
   - การแสดงสถานะในการทำงานของผู้ใช้ เช่น Online ,Offline , Away, N/A เป็นต้น
2. ให้การบริการสำหรับการค้นหาเพื่อนหรือคู่สนทนาซึ่งใช้โปรแกรม ICQ โดยสามารถค้นหาได้จากการกำหนดข้อมูลดังต่อไปนี้
   - ค้นหาจากหมายเลขประจำตัวผู้ใช้งาน (UIN หรือ ICQ#)
   - ค้นหาจากชื่อ นามสกุลหรือชื่อเล่นของผู้ใช้งาน
   - ค้นหาจากอีเมล์แอดเดรสของผู้ใช้งาน
   - ค้นหาจากการใช้ ICQ Meet People directory ซึ่งสามารถทำการค้นหาได้หลากหลายขึ้นเช่น ค้นหาจากกลุ่มอายุผู้ใช้งาน กลุ่มอาชีพ กลุ่มเรื่องที่สนใจ ประเทศที่อาศัยอยู่และภาษาที่ใช้เป็นต้น
3. ให้บริการรับฝากข้อความข้อความผ่านทางเซิร์ฟเวอร์ หรือเรียกว่าบริการ offline message ซึ่งเป็นบริการสำหรับส่งข้อความไปยังผู้ใช้งาน ICQ อื่นที่ทำการออฟไลน์ในขณะนั้น โดยเมื่อใดที่ผู้ใช้งานผู้นั้นทำการออนไลน์ ทางเซิร์ฟเวอร์ก็จะให้บริการส่งข้อความขณะที่ผู้ใช้งานออฟไลน์ทั้งหมดไปให้
4. ให้บริการการเปลี่ยนแปลงสถานะในการใช้งานต่างๆ ของผู้ใช้งาน ICQ โดยเมื่อผู้ใช้งาน ICQ ทำการเปลี่ยนแปลงสถานะ โปรแกรม ICQ ไคลเอ็นต์จะส่งคำสั่งการเปลี่ยนสถานะไปแจ้งให้กับเซิร์ฟเวอร์รับรู้ จากนั้นเซิร์ฟเวอร์จะส่งคำสั่งเปลี่ยนสถานะแจ้งไปยังผู้ใช้งานทั้งหมดที่ทำการออนไลน์และมีรายชื่อของผู้ใช้งานที่ทำการเปลี่ยนสถานะอยู่ในคอนแทคลิสต์ทราบถึงสถานะของผู้ใช้งานในปัจจุบัน

5. ให้บริการในการเก็บคอนแทคลิสต์ของผู้ใช้งาน ซึ่งทำให้เมื่อผู้ใช้งานไปเรียกใช้โปรแกรม ICQ จากเครื่องอื่นหรือลงโปรแกรมใหม่ในภายหลัง รายชื่อเพื่อนหรือคู่สนทนาในคอนแทคลิสต์ก็จะเหมือนเดิม ซึ่งหมายความว่าทุกครั้งที่ผู้ใช้งานทำการเพิ่มรายชื่อคู่สนทนาลงในคอนแทคลิสต์ โปรแกรม ICQ ไคลเอ็นต์จะส่งคำขอไปยังเซิร์ฟเวอร์เพื่อทำการขอเพิ่มรายการในคอนแทคลิสต์ จากนั้นเซิร์ฟเวอร์ก็จะทำการเก็บข้อมูลที่ผู้ใช้ทำการเพิ่มเพื่อนหรือคู่สนทนาไว้ และแจ้งไปยังผู้ใช้งานที่ถูกเพิ่มลงในคอนแทคลิสต์ทราบ
6. ให้บริการลงทะเบียนผู้ใช้งาน ICQ สำหรับผู้ใช้งานใหม่ ซึ่งเซิร์ฟเวอร์จะทำการออกหมายเลขประจำตัวผู้ใช้งานใหม่ให้กับผู้ลงทะเบียนสำหรับนำไปใช้งาน
7. ให้บริการยกเลิกการจดทะเบียนผู้ใช้งาน ICQ โดยผู้ใช้ ICQ สามารถยกเลิกการจดทะเบียนหมายเลข ICQ ที่ไม่ต้องการใช้ได้
8. ให้บริการเสริมอื่นๆ เช่นการส่งข้อความ SMS ไปยังโทรศัพท์มือถือเป็นต้น

5.3.2 ลักษณะแพ็กเก็ตที่ใช้สื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

แพ็กเก็ตของ ICQ เวอร์ชัน 8 ซึ่งติดต่อเครือข่ายอินเตอร์เน็ตผ่านทางโพรโตคอล TCP จะมีรูปแบบในการใช้งานอยู่ 2 รูปแบบคือ แพ็กเก็ตที่ใช้สื่อสารสำหรับคำสั่งและแพ็กเก็ตที่ใช้สื่อสารทั่วไป



รูปที่ 5.1 รูปแบบแพ็กเก็ต ICQ เวอร์ชัน 8

ส่วนประกอบต่างๆ ของแพ็กเก็ต ICQ เวอร์ชัน 8 ซึ่งมีรายละเอียดดังต่อไปนี้

1. ID ใช้สำหรับแสดงว่าการเป็นโพรโตคอลของ ICQ โดยหากเป็นโพรโตคอลของ ICQ จะมีค่าเป็น 02
2. Channel เป็นการกำหนดช่องทางสำหรับการสื่อสาร ซึ่งสามารถกำหนดช่องทางได้ดังนี้
   - 01 - ช่องทางสื่อสารสำหรับล็อกอิน (Log in)
   - 02 - ช่องทางสื่อสารสำหรับคำสั่ง (Command)
   - 03 - ช่องทางสื่อสารสำหรับข้อผิดพลาด (Errors)

- 04 - ช่องทางสื่อสารสำหรับล็อกเอาท์ (Log out)
- 05 - ช่องทางสื่อสารสำหรับ Keep alive

3. Sequence คือหมายเลขลำดับของแพ็กเก็ตที่ทำการรับและส่งระหว่างไคลเอ็นต์และเซิร์ฟเวอร์
4. Length คือขนาดความยาวของแพ็กเก็ต
5. Family เป็นกลุ่มของคำสั่งที่จะใช้งาน
6. Command เป็นคำสั่งสำหรับขอใช้บริการ
7. Flag จะเป็นค่าที่ขึ้นอยู่กับ Command ซึ่งใช้ร่วมกับ Command
8. Reference ใช้สำหรับแยกความแตกต่างของ Data ที่สัมพันธ์กันในแพ็กเก็ตที่ต่างกันเช่น ถ้ามีการใช้คำสั่งค้นหาคู่สนทนาที่แยกจากกัน 2 ครั้ง Reference จะใช้แยกความแตกต่างของผลลัพธ์ของการค้นหาคู่สนทนาที่ตอบกลับมาจากเซิร์ฟเวอร์
9. Data คือข้อมูลที่ทำการรับส่ง

5.3.3 คำสั่งที่ใช้ในการสื่อสารจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์

คำสั่งที่ใช้ในการสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์สามารถแบ่งออกได้เป็น 2 รูปแบบคือ คำสั่งในการสื่อสารจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์และคำสั่งในการสื่อสารจากเซิร์ฟเวอร์ไปยังไคลเอ็นต์ โดยในส่วนนี้จะกล่าวถึงคำสั่งในการสื่อสารจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์เพื่อขอใช้บริการต่างๆ ดังนี้

รหัส		ชื่อคำสั่ง	รายละเอียด
Family	Command		
00 01	00 02	Ready	ส่ง family และ version กลับไปยังเซิร์ฟเวอร์
00 01	00 06	Raterequest	ส่ง request เพื่อถามความเร็วของแพ็กเก็ตที่ต้องการส่งไปยังเซิร์ฟเวอร์และขนาดของแพ็กเก็ตที่ใหญ่ที่สุดที่เป็นไปได้
00 01	00 08	Ackrates	ตอบรับคำสั่ง (srv_rates) ของเซิร์ฟเวอร์ที่ส่งมา
00 01	00 0E	Reqinfo	ขอข้อมูลของไคลเอ็นต์ซึ่งถูกเก็บไว้ยังเซิร์ฟเวอร์
00 01	00 17	Families	ตอบรับคำสั่ง (srv_families) ของเซิร์ฟเวอร์ที่ส่งมา
00 01	00 1E	Setstatus	กำหนดสถานะของไคลเอ็นต์
00 02	00 02	Reqlocation	ขอข้อมูลสำหรับพื้นที่บริการ
00 02	00 04	Setuserinfo	ส่งข้อมูลของผู้ใช้งานไคลเอ็นต์ไปให้เซิร์ฟเวอร์
00 03	00 02	Reqbuddy	ขอข้อมูลสำหรับบริการ buddy
00 03	00 04	Addcontact	ทำการส่งเมื่อทำการล็อกอินและเมื่อทำการเพิ่มคู่สนทนาลงในคอนแทคลิสต์
00 03	00 05	Removecontact	ส่งเมื่อทำการลบคู่สนทนาออกไปจากคอนแทคลิสต์

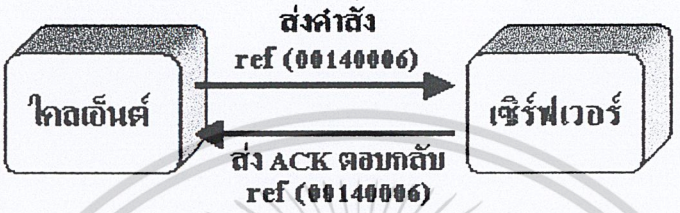
รหัส		ชื่อคำสั่ง	รายละเอียด
Family	Command		
00 04	00 02	Seticbm	ส่งค่าที่เปลี่ยนแปลงจากค่าที่ได้รับมาจากคำสั่ง Replyicbm ที่ส่งมาจากเซิร์ฟเวอร์
00 04	00 04	Reqicbm	ขอข้อมูลสำหรับ icbm
00 04	00 06	Sendmsg	ส่งข้อความผ่านทางเซิร์ฟเวอร์
00 09	00 02	Reqbos	ขอข้อมูลสำหรับ bos
00 09	00 05	Addvisible	เพิ่มรายชื่อคู่สนทนาลงใน visible list
00 09	00 06	Remvisible	ลบรายชื่อคู่สนทนาออกจาก visible list
00 09	00 07	Addinvisible	เพิ่มรายชื่อคู่สนทนาลงใน invisible list
00 09	00 08	Reminvisible	ลบรายชื่อคู่สนทนาออกจาก invisible list
00 13	00 02	Reqlists	ขอลิสต์จากเซิร์ฟเวอร์
00 13	00 04	Reqroster	คำสั่งนี้จะเหมือนกับ checkroster ซึ่งจะขอเซิร์ฟเวอร์ side contact list แต่จะแตกต่างตรงที่ reqroster จะ ไม่มี parameter และส่วนมากเซิร์ฟเวอร์จะตอบกลับมาด้วย replyroster มากกว่าที่จะตอบกลับด้วย replyrosterok
00 13	00 05	Checkroster	จะส่งเป็นจังหวะในการขอเซิร์ฟเวอร์ side contact list ซึ่งจะถูกตอบกลับมาด้วย replyroster หรือ replyrosterok
00 13	00 07	Rosterack	ตอบรับคำสั่ง (srv_replyroster) ของเซิร์ฟเวอร์ที่ส่งมา
00 13	00 08	Addbuddy	ประกาศให้เซิร์ฟเวอร์ทราบว่าไคลเอ็นต์ได้เพิ่มคอนแทคใหม่ลงในคอนแทคลิสต์
00 13	00 09	Updategroup	ส่งเมื่อผู้ใช้เพิ่มกลุ่มในคอนแทคลิสต์และ update เซิร์ฟเวอร์ side contact lists
00 13	00 0A	Deletebuddy	ลบ buddy ออกจากคอนแทคลิสต์
00 13	00 11	Addstart	ส่งก่อน addbuddy เมื่อทำการเพิ่มคอนแทคใหม่ลงในคอนแทคลิสต์
00 13	00 12	Addend	ส่งไปแจ้งให้เซิร์ฟเวอร์ทราบว่าการเปลี่ยนแปลงเซิร์ฟเวอร์ side contact lists เสร็จสิ้นแล้ว
00 13	00 18	Reqauth	ขอ authorize คู่สนทนาซึ่งเพิ่มไว้ในคอนแทคลิสต์
00 13	00 1A	Authorize	ให้อนุญาตผู้ใช้งานที่ขอ authorize
00 15	00 02	Toicqserver	ส่งคำสั่งไปให้เซิร์ฟเวอร์ของ ICQ

รหัส		ชื่อคำสั่ง	รายละเอียด
Family	Command		
00 15	00 02 (60)	Reqofflinemsgs	ขอใช้บริการเซิร์ฟเวอร์สำหรับส่งข้อความออฟไลน์
00 15	00 02 (62)	Ackofflinemsgs	ทำการแจ้งว่าได้รับข้อความออฟไลน์แล้ว
00 15	00 02 (2000)	Meta	กำหนดข้อมูลหรือดึงข้อมูลออกมา
00 15	00 02 (2000/1002)	Metasetgeneral	กำหนดข้อมูลทั่วไปที่เกี่ยวกับผู้ใช้
00 15	00 02 (2000/1021)	Metasetmore	กำหนดข้อมูลที่มากขึ้นที่เกี่ยวกับผู้ใช้
00 15	00 02 (2000/1030)	Metasetabout	กำหนด about String
00 15	00 02 (2000/1232)	Metareqinfo	ขอข้อมูลเกี่ยวกับผู้ใช้อื่น
00 15	00 02 (2000/1331)	Searchwp	ค้นหาผู้ใช้งานโดยใช้ข้อมูลของผู้ใช้ทั้งหมด
00 15	00 02 (2000/1375)	Searchbypersinf	ค้นหาผู้ใช้งานโดยใช้ชื่อ
00 15	00 02 (2000/1385)	Searchbyuin	ค้นหาผู้ใช้งานโดยใช้หมายเลขประจำตัวผู้ใช้งาน
00 15	00 02 (2000/1880)	Metasetrandom	กำหนดกลุ่มสนทนาด้วยการสุ่มค่า
00 15	00 02 (2000/2200)	Reqxml	ขอ URL จากเซิร์ฟเวอร์
00 15	00 02 (2000/2750)	Snac	Unknown
00 15	00 02 (2000/5250)	Sendsms	ส่ง SMS ไปยังโทรศัพท์มือถือ
00 17	00 04	Registeruser	ขอทำการลงทะเบียนผู้ใช้งาน ICQ ใหม่

ตารางที่ 5.3 คำสั่งในการสื่อสารจากไคลเอ็นต์ไปยังเซิร์ฟเวอร์

5.3.4 การสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์

ในการสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ เมื่อไคลเอ็นต์ทำการส่งคำสั่งใดไปให้เซิร์ฟเวอร์ ทางเซิร์ฟเวอร์จะต้องตอบกลับมาด้วยคำสั่ง ACK และในทางกลับกันเมื่อเซิร์ฟเวอร์ส่งคำสั่งใดมาให้ไคลเอ็นต์ ทางไคลเอ็นต์ก็ต้องตอบกลับคำสั่งนั้นด้วยคำสั่ง ACK โดยที่การใช้คำสั่ง ACK ตอบกลับคำสั่งใดนั้นจะต้องมีค่า reference ในการส่งคำสั่งและ ACK ตอบกลับเท่ากัน มิเช่นนั้นจะติดต่อกับเซิร์ฟเวอร์ไม่ได้อีกจนกว่าจะส่งค่า reference ที่ถูกต้อง



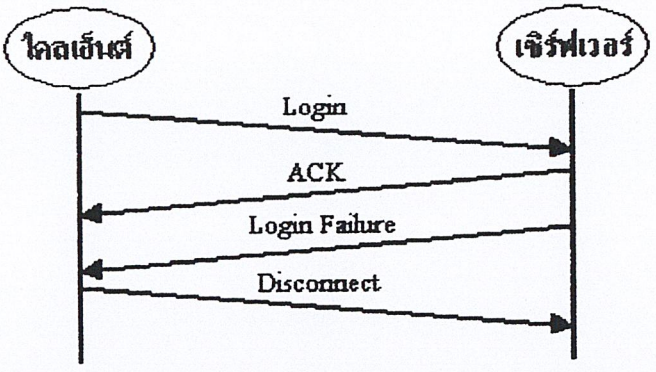
รูปที่ 5.2 การส่งคำสั่งและตอบกลับระหว่างไคลเอ็นต์และเซิร์ฟเวอร์

การใช้ค่า reference ในโปรแกรม ICQ เวอร์ชันปัจจุบันจะมีข้อดีกว่าการใช้ค่า sequence ในโปรแกรม ICQ เวอร์ชันเก่า เนื่องจากการใช้ค่า sequence จะเรียงลำดับแพ็กเก็ตที่ทำการส่งและตอบกลับ หากส่งคำสั่งใดไปก่อนแล้วคำสั่งนั้นก็ต้องกลับมาก่อนซึ่งทำให้การทำงานช้า แต่เมื่อใช้ค่า reference แล้ว ค่า sequence จะเป็นการเรียงลำดับของแพ็กเก็ตที่ทำการส่งเท่านั้น หากแพ็กเก็ตที่ส่งไปทีหลังไม่ต้องทำงานร่วมกับแพ็กเก็ตที่ส่งไปก่อน แพ็กเก็ตที่ส่งไปทีหลังไม่จำเป็นต้องรอให้แพ็กเก็ตที่ส่งไปก่อนทำงานเสร็จ หากแพ็กเก็ตที่ส่งไปทีหลังทำงานเสร็จก่อนก็สามารถตอบกลับมาได้ ทำให้มีการทำงานที่เร็วขึ้น

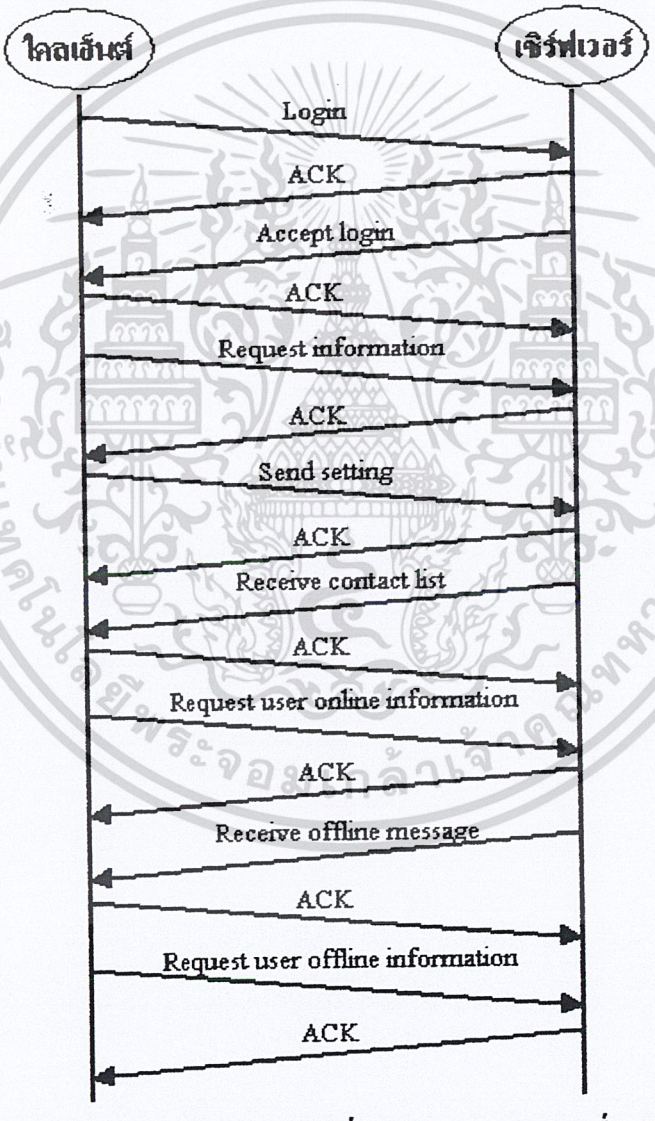
การสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ที่มีการใช้งานในโปรแกรม ICQ เช่น การล็อกอินเพื่อขอเข้าใช้บริการ การเรียกขอคอนแทคลิสต์ของผู้ใช้ การร้องขอข้อมูลรายการผู้ใช้งานที่ทำการออนไลน์จากเซิร์ฟเวอร์ การเปลี่ยนแปลงสถานะ การส่งข้อความผ่านทางเซิร์ฟเวอร์ การค้นหาคู่สนทนา การเรียกขอดูข้อมูลของคู่สนทนาเป็นต้น โดยส่วนนี้ขอแสดงขั้นตอนการล็อกอินเพื่อเข้าใช้บริการและการค้นหาคู่สนทนาซึ่งมีความซับซ้อนค่อนข้างมาก ส่วนในการสื่อสารอื่นๆ มีขั้นตอนเพียงการส่งแพ็กเก็ตโต้ตอบกันเพียงชุดเดียวเท่านั้น

5.3.4.1 ขั้นตอนการล็อกอินเพื่อใช้บริการ

ในการขอใช้บริการจากเซิร์ฟเวอร์ ICQ ขั้นตอนแรกจะต้องทำการเชื่อมต่อกับเซิร์ฟเวอร์เพื่อทำการตรวจสอบสิทธิก่อนจึงจะทำการสื่อสารด้วยคำสั่งอื่นๆ ต่อไปได้ ซึ่งหากไม่ผ่านขั้นตอนนี้จะไม่สามารถเข้าใช้งานได้ไม่ว่าจะส่งคำสั่งใดๆ ไปยังเซิร์ฟเวอร์ก็จะไม่การตอบกลับมาเลย ซึ่งสามารถแสดงเป็นไดอาแกรมได้ดังรูปที่ 5.3



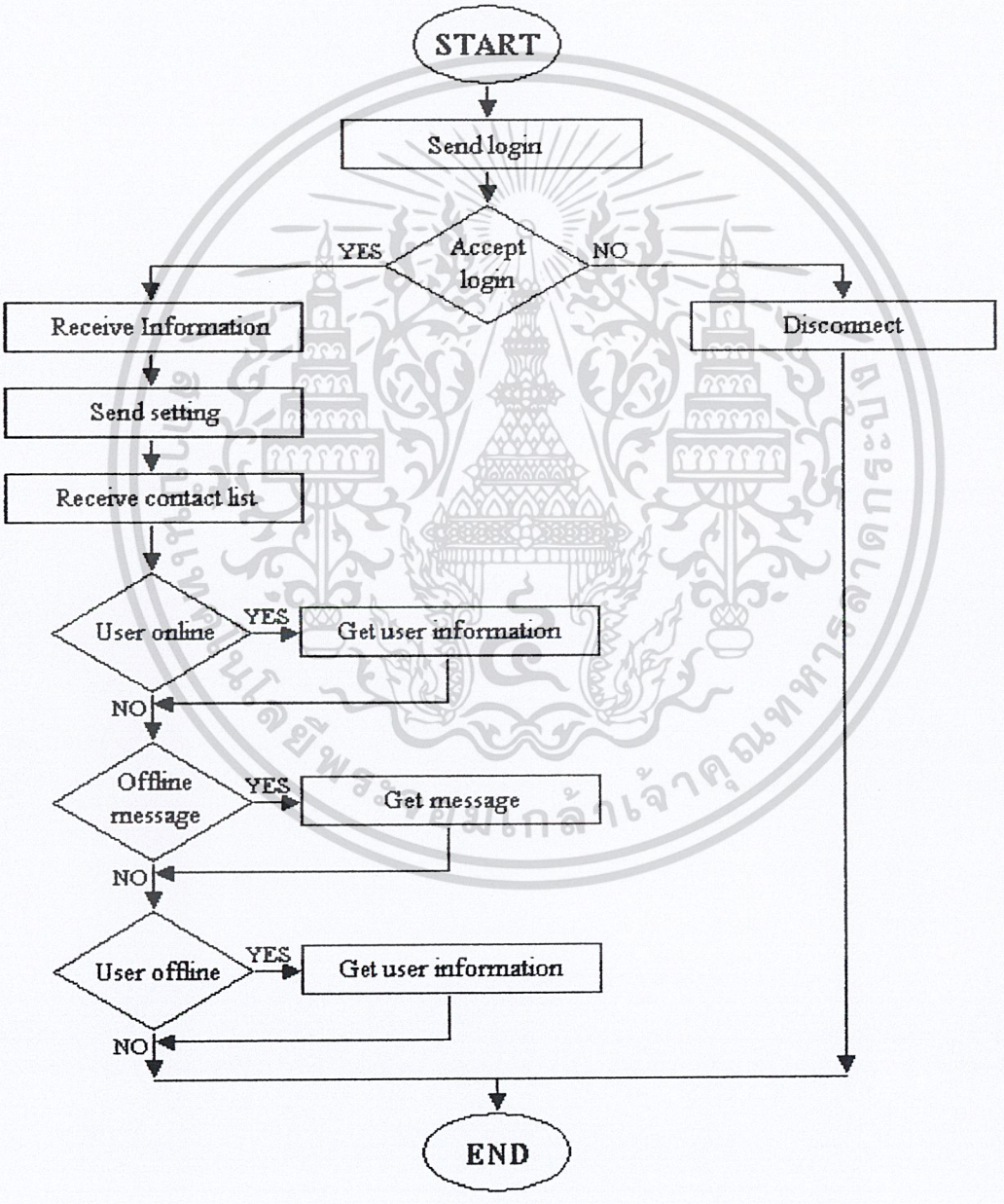
รูปที่ 5.3 ไดอาแกรมแสดงความผิดพลาดจากการตรวจสอบสิทธิในการล็อกอินเพื่อขอใช้บริการ



รูปที่ 5.4 ไดอาแกรมแสดงการรับส่งคำสั่งต่างๆ ในการล็อกอินเพื่อขอใช้บริการ

หากสามารถผ่านการตรวจสอบสิทธิ์ ซึ่งแสดงว่าเป็นผู้ใช้งานนั้นแล้ว จะต้องทำการส่งคำสั่งต่างๆ ดังไดอาแกรมในรูปที่ 5.4 ก่อน เมื่อทำการรับส่งคำสั่งทั้งหมดเสร็จแล้ว จะถือว่าผู้ใช้งานสามารถทำการล็อกอินเสร็จสิ้นแล้วสามารถใช้งานโปรแกรม ICQ และรับส่งคำสั่งอื่นๆ ได้ตามที่ผู้ใช้งานต้องการ

จากการสื่อสารในไดอาแกรม เราสามารถสรุปการทำงานได้ตาม flowchart ที่แสดงไว้ในรูปที่ 5.5 เมื่อทำงานจนจบขั้นตอนใน flowchart แล้ว เรียกว่าขั้นตอนการ login เสร็จสิ้น ต่อจากนี้สามารถส่งคำสั่งใดๆก็ได้ที่ต้องการ โดยการส่งแต่ละครั้งต้องเพิ่มค่า sequence ครั้งละ 1 และการใช้คำสั่ง ACK ตอบกลับต้องให้ค่า reference เหมือนกับคำสั่งที่เซิร์ฟเวอร์ส่งมาจึงสามารถทำงานได้อย่างถูกต้อง



รูปที่ 5.5 Flow chart แสดงการทำงานในการล็อกอินเพื่อขอใช้บริการ

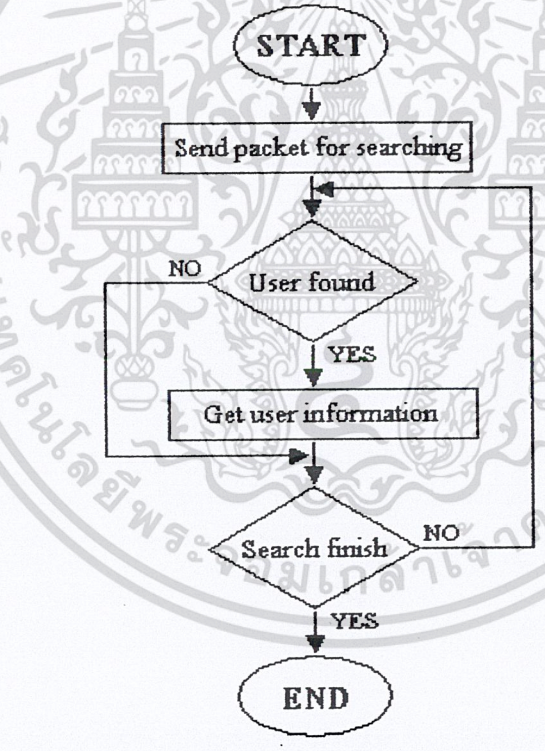
5.3.4.2 ขั้นตอนการค้นหาคู่สนทนา

หากเราทราบหมายเลขประจำตัวผู้ใช้งาน ชื่อ-นามสกุล ชื่อเล่นหรืออีเมล์แอดเดรสอย่างใดอย่างหนึ่งของผู้ใช้งานที่ต้องการเพิ่มลงในคอนแทคลิสต์จะต้องทำการค้นหาผ่านทางเซิร์ฟเวอร์ก่อนแล้วจึงเพิ่มคู่สนทนาลงในคอนแทคลิสต์ได้

คำสั่งในการค้นหาข้อมูลของผู้ใช้งานได้แก่ คำสั่ง SEARCHBYPERSINF, SEARCHBYUIN และ SEARCHBYMAIL โดยแต่ละคำสั่งใช้งานดังนี้

- SEARCHBYPERSINF เป็นคำสั่งค้นหาคู่สนทนาโดยใช้ชื่อ-นามสกุลหรือชื่อเล่นของผู้ใช้
- SEARCHBYUIN เป็นคำสั่งค้นหาคู่สนทนาโดยใช้หมายเลขประจำตัวผู้ใช้งาน
- SEARCHBYMAIL เป็นคำสั่งค้นหาคู่สนทนาโดยใช้อีเมล์แอดเดรส

จากรูปที่ 5.6 จะแสดงการทำงานในการค้นหาคู่สนทนา โดยเมื่อส่งคำสั่งค้นหาแล้วให้รอรับข้อมูลจากทางเซิร์ฟเวอร์ หากพบว่ามีผู้ใช้งานที่ตรงกับข้อมูลที่เราต้องการค้นหา เซิร์ฟเวอร์จะส่งแพ็กเก็ตข้อมูลผู้ใช้งานนั้นกลับมา ซึ่งเซิร์ฟเวอร์จะส่งแพ็กเก็ตมาเรื่อยๆ จนกระทั่งเซิร์ฟเวอร์ไม่สามารถค้นหาได้



รูปที่ 5.6 Flow chart แสดงการทำงานในการค้นหาคู่สนทนา

5.4 การติดต่อระหว่างไคลเอ็นต์กับไคลเอ็นต์

การสื่อสารระหว่างไคลเอ็นต์กับไคลเอ็นต์เรียกอีกอย่างหนึ่งว่าแบบ direct connect ไคลเอ็นต์ทั้งสองฝั่งจะทำการสื่อสารเชื่อมต่อโดยตรงโดยใช้โพรโตคอลแบบ TCP เช่นเดียวกับการสื่อสารระหว่างไคล

เอ็นต์กับเซิร์ฟเวอร์ ซึ่งการเชื่อมต่อโดยใช้ TCP นี้สามารถรับประกันความครบถ้วนของข้อมูลส่งถึงผู้ใช้งานอีกฝั่งหนึ่งได้อย่างแน่นอน

รูปแบบข้อมูลสำหรับส่งระหว่างไคล์เอ็นต์ต่างจากข้อมูลสำหรับสื่อสารกับเซิร์ฟเวอร์ รูปแบบข้อมูลสำหรับสื่อสารระหว่างไคลเอ็นต์มีอยู่ด้วยกัน 2 รูปแบบได้แก่ PEER_INIT และ PEER_MSG

- PEER_INIT ไคลเอ็นต์ซึ่งเป็นผู้ส่งจะส่งชุดข้อมูลชุดนี้ก่อนจะเริ่มทำการส่งข้อความแรกให้กับไคลเอ็นต์ซึ่งเป็นผู้รับ โดยชุดข้อมูลชุดนี้จะประกอบด้วยข้อมูลต่างๆ ดังตารางที่ 5.4

PEER_INIT		
ชื่อข้อมูล	**ขนาด**	**คำอธิบาย**
CMD	BYTE	คำสั่งในการติดต่อ โดยกำหนดเป็น FF
TCPVER	WORD	เวอร์ชันแพ็กเก็ตของ ICQ โดยในที่นี้ใช้เวอร์ชัน 8 จึงกำหนดเป็น 08 00
LENGTH	WORD	ขนาดของแพ็กเก็ต PEER_INIT ที่ทำการส่ง ซึ่งมีขนาด 43 ไบต์ จึงกำหนดเป็น 2B 00
DESTUIN	DWORD	หมายเลขประจำตัวผู้ใช้งาน (UIN) ของผู้รับ
UNKNOWN	WORD	กำหนดเป็น 00 00
OURPORT	DWORD	พอร์ตสำหรับรอรับข้อความ
OURUIN	DWORD	หมายเลขประจำตัวผู้ใช้งาน (UIN) ของผู้ส่ง
OURREMIP	DWORD	IP address จริงของผู้ส่งที่ยอมให้เซิร์ฟเวอร์ทราบ
OURINTIP	DWORD	IP address ถ้าหากอยู่หลังไฟร์วอลล์จะเป็น IP address ของไฟร์วอลล์
TCPFLAGS	BYTE	ค่าแฟล็กเกี่ยวกับ TCP กำหนดค่าเป็น 04
OURPORT2	DWORD	พอร์ตสำหรับรอรับ chat และไฟล์
COOKIE	DWORD	คุ้กกี้ที่เซิร์ฟเวอร์ให้ในการใช้งาน
UNKNOWN	DWORD	กำหนดเป็น 50 00 00 00
UNKNOWN	DWORD	กำหนดเป็น 03 00 00 00
UNKNOWN	DWORD	กำหนดเป็น 00 00 00 00

ตารางที่ 5.4 ข้อมูลต่างๆ ใน PEER_INIT

- PEER_MSG ชุดข้อมูลซึ่งใช้บรรจุข้อความสำหรับทำการสนทนาระหว่างไคลเอ็นต์ โดยชุดข้อมูลชุดนี้จะประกอบด้วยข้อมูลต่างๆ ดังตารางที่ 5.5

PEER_MSG		
ชื่อข้อมูล	**ขนาด**	**คำอธิบาย**
CMD	BYTE	คำสั่งในการส่งข้อความ โดยกำหนดเป็น 02

ชื่อข้อมูล	ขนาด	คำอธิบาย
CHECKSUM	DWORD	ค่า check sum ของแพ็กเก็ต
COMMAND	WORD	คำสั่งกำหนดชนิดของข้อความ โดยแสดงรายละเอียดในตารางที่ 5.6
UNKNOWN	WORD	กำหนดเป็น 0E 00
SEQUENCE	WORD	หมายเลขลำดับของแพ็กเก็ต
UNKNOWN	DWORD	กำหนดเป็น 00 00 00 00
UNKNOWN	DWORD	กำหนดเป็น 00 00 00 00
UNKNOWN	DWORD	กำหนดเป็น 00 00 00 00
MSGTYPE	WORD	ชนิดของข้อความ โดยแสดงรายละเอียดในตารางที่ 5.7
UNKNOWN	WORD	กำหนดเป็น 00 00
STATUS	WORD	สถานะการรับข้อความของไคลเอ็นต์ โดยแสดงรายละเอียดในตารางที่ 5.8
FLAGS	WORD	ค่าแฟล็กของการส่งข้อความ
MSGLEN	WORD	ขนาดของข้อความที่ทำการรับส่ง
MESSAGE	VARIES	ข้อความที่ทำการรับส่ง
EXTRA	VARIES	การกำหนดรูปแบบพิเศษให้กับข้อความเช่น สีตัวอักษร, สีพื้นหลัง, รูปแบบตัวอักษรและขนาดตัวอักษรเป็นต้น

ตารางที่ 5.5 ข้อมูลต่างๆ ใน PEER_MSG

PEER_MSG COMMAND	
ค่าตัวแปร	**คำอธิบาย**
07 D0 (2000)	ยกเลิกการติดต่อ
07 DA (2010)	ติดต่อกลับ
07 EE (2030)	ส่งข้อความ

ตารางที่ 5.6 ข้อมูล COMMAND ใน PEER_MSG

PEER_MSG MSGTYPE	
ค่าตัวแปร	**คำอธิบาย**
00 00	ส่งข้อความตอบกลับอัตโนมัติ
00 01	ส่งข้อความปกติ
00 02	เริ่มต้นสื่อสารแบบ Chat
00 03	เริ่มต้นส่งไฟล์

ค่าตัวแปร	คำอธิบาย
00 04	ส่ง URL address
00 06	ร้องขอสิทธิในการรับรู้สถานะของผู้อื่น
00 07	ไม่ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น
00 08	ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น
00 0C	สัญญาณอัตโนมัติแสดงว่าตนเองถูกผู้อื่นเพิ่มในรายการคอนแทคลิสต์
00 0D	ส่งข้อความ web pager
00 0E	ส่งข้อความ e-mail pager
00 1A	ส่งข้อความหรือการ์ดอวยพร
03 E8	ได้รับข้อความจากสถานะ Away
03 E9	ได้รับข้อความจากสถานะ Occupied
03 EA	ได้รับข้อความจากสถานะ Extended Away
03 EB	ได้รับข้อความจากสถานะ Do Not Disturb
03 EC	ได้รับข้อความจากสถานะ Free For Chat

ตารางที่ 5.7 ข้อมูล MSGTYPE ใน PEER_MSG

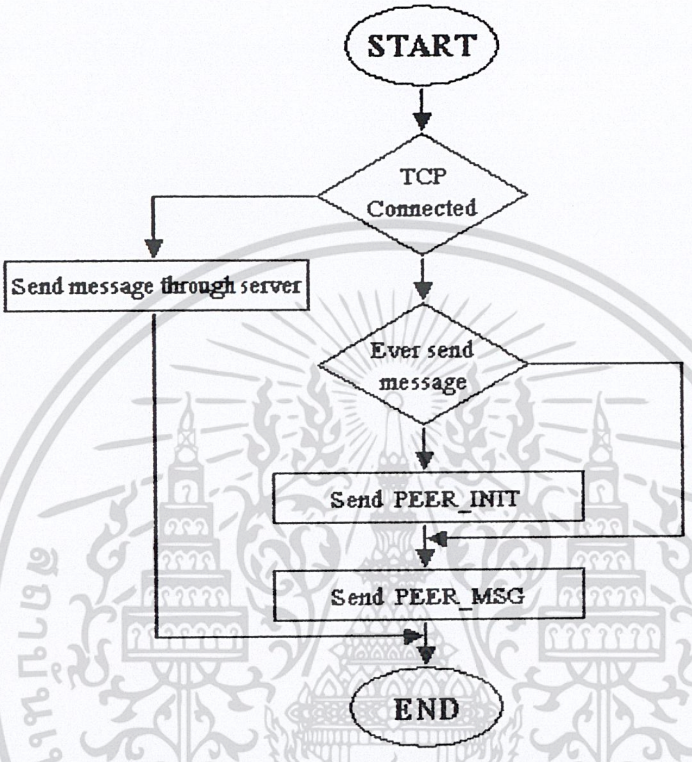
PEER_MSG STATUS	
ค่าตัวแปร	คำอธิบาย
00 00	ผู้รับทำการออนไลน์สามารถรับข้อความได้
00 01	ปฏิเสธการรับข้อความ
00 04	ปฏิเสธการรับข้อความอัตโนมัติจากการที่ผู้รับอยู่ในสถานะ Away
00 09	ปฏิเสธการรับข้อความอัตโนมัติจากการที่ผู้รับอยู่ในสถานะ Occupied
00 0A	ปฏิเสธการรับข้อความอัตโนมัติจากการที่ผู้รับอยู่ในสถานะ Do not disturb
00 0E	ปฏิเสธการรับข้อความอัตโนมัติจากการที่ผู้รับอยู่ในสถานะ Extended Away

ตารางที่ 5.8 ข้อมูล STATUS ใน PEER_MSG

ในการรับส่งข้อความระหว่างไคลเอ็นต์กับไคลเอ็นต์ผ่านทาง TCP ก่อนอื่นจะต้องทำการตรวจสอบสถานะในการเชื่อมต่อ TCP กับผู้ใช้ปลายทาง เมื่อเชื่อมต่อสำเร็จแล้วจะต้องตรวจสอบว่าเคยส่งข้อความไปยังผู้ใช้ปลายทางนี้แล้วหรือไม่ ถ้ายังไม่เคยส่งข้อความสื่อสารกันเลยให้เริ่มโดยส่งชุดข้อมูล PEER_INIT ก่อนแล้วจึงส่ง PEER_MSG ตามไป ซึ่งถ้าหากทำการส่งชุดข้อมูล PEER_MSG ไปโดยไม่ส่ง

PEER_INIT ไปก่อนจะทำให้ผู้ใช้ปลายทางไม่สามารถรับข้อความได้ และในอีกกรณีหนึ่งเมื่อตรวจสอบแล้วพบว่าเคยส่งข้อความให้กับผู้ใช้ปลายทางคนนี้แล้วก็สามารถส่งชุดข้อมูล PEER_MSG ไปได้

ในการสื่อสารระหว่างไคลเอ็นต์กับไคลเอ็นต์เพื่อทำการรับส่งสารระหว่างกันโดยตรง สามารถสรุปได้ดัง flow chart ในรูปที่ 5.7



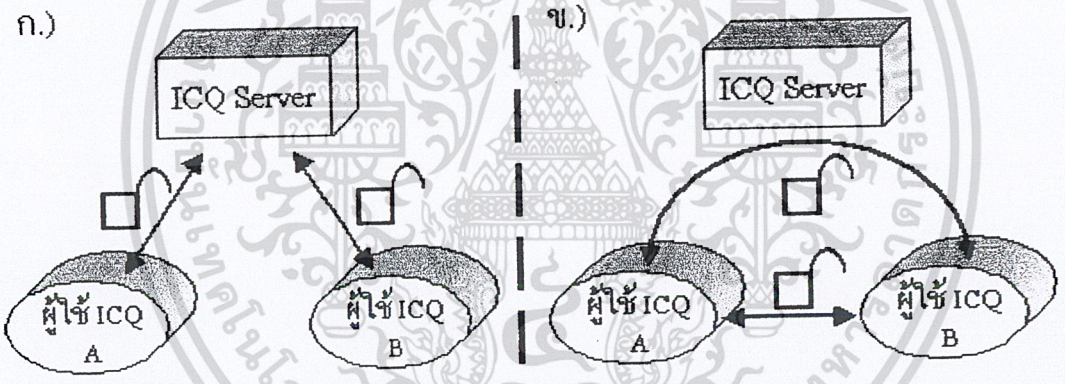
รูปที่ 5.7 แสดงขั้นตอนการส่งข้อความโดยตรงระหว่างไคลเอ็นต์ผ่านทาง TCP

บทที่ 6
การออกแบบและพัฒนาโครงงาน

6.1 แนวคิดและการออกแบบความสามารถของโปรแกรม

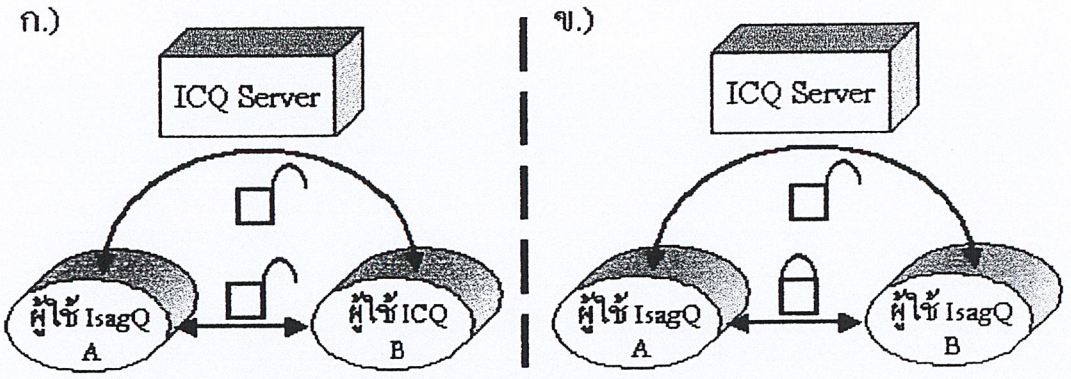
โครงงานนี้จะมุ่งเน้นถึงการปรับปรุงโปรแกรมรับส่งสารด่วนให้มีความปลอดภัยมากขึ้น โดยใช้การเข้าและถอดรหัสข้อมูลสารที่ต้องการรับส่ง

ในโปรแกรม ICQ จะมีขั้นตอนในการใช้งานคือ โปรแกรมฝั่งไคลเอ็นต์จะทำการติดต่อไปยังเซิร์ฟเวอร์ของ ICQ เพื่อขอใช้บริการตามรูปที่ 6.1 ก.) ซึ่งเซิร์ฟเวอร์ของ ICQ จะให้บริการโดยส่งค่ารายการของผู้ใช้ที่มีการติดต่อระหว่างกันกลับมาให้ผู้ใช้ที่ร้องขอไป จากนั้นจะเข้าสู่ขั้นตอนในรูป 6.1 ข.) ซึ่งเป็นการทำงานของผู้ใช้ที่มีการติดต่อกัน ซึ่งสามารถทำได้ 2 รูปแบบคือ การทำงานผ่านทางเซิร์ฟเวอร์ของ ICQ และการทำงานระหว่างกันโดยตรง



รูปที่ 6.1 รูปแบบการติดต่อในโปรแกรม ICQ

จากการทำงานในโปรแกรมไคลเอ็นต์ของ ICQ จะพบว่าการสื่อสารระหว่างกันโดยตรงในรูปที่ 6.1 ข.) จะไม่มีการเข้า-ถอดรหัสข้อมูลสารที่ทำการรับส่ง ทำให้การรับส่งสารระหว่างผู้ใช้งานโดยตรงไม่ปลอดภัย ดังนั้นในการรับส่งสารระหว่างผู้ใช้งานโดยตรงในโปรแกรม IsagQ จะมี 2 รูปแบบคือการรับส่งสารโดยมีการเข้าและถอดรหัสข้อมูลซึ่งจะใช้งานเมื่อผู้รับใช้งานโปรแกรม IsagQ ดังรูปที่ 6.2 ก.) และการรับส่งสารโดยไม่มีการเข้าและถอดรหัสข้อมูลซึ่งใช้เมื่อผู้รับใช้งานโปรแกรม ICQ ดังรูปที่ 6.2 ข.)

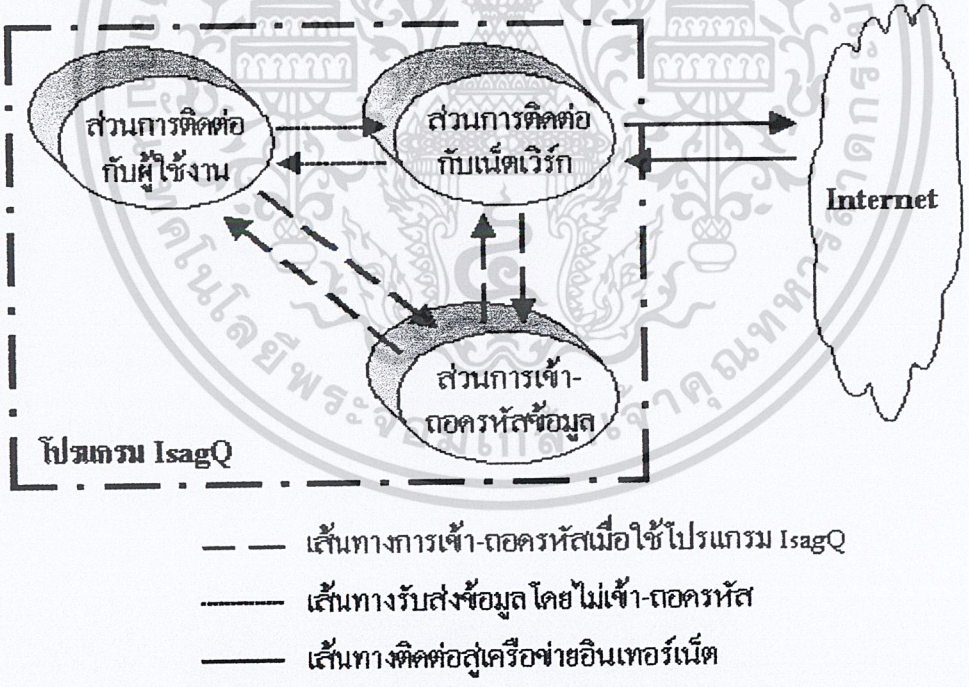


รูปที่ 6.2 รูปแบบการสื่อสารโดยตรงในโปรแกรม IsagQ

6.2 ส่วนประกอบของโปรแกรม

ในโครงงานนี้ได้แบ่งส่วนประกอบของโปรแกรม IsagQ ออกเป็น 3 ส่วนหลักๆ คือ

1. ส่วนการติดต่อกับผู้ใช้งาน
2. ส่วนการติดต่อกับเน็ตเวิร์ก
3. ส่วนการเข้า-ถอดรหัสข้อมูล



รูปที่ 6.3 ส่วนประกอบในการทำงานของโปรแกรม IsagQ

6.2.1 ส่วนการติดต่อกับผู้ใช้งาน

ส่วนการติดต่อกับผู้ใช้งานเป็นส่วนที่ใช้รับคำสั่งและข้อมูลจากผู้ใช้งาน ซึ่งหากเราทำการส่งสาร โดยการเข้ารหัสข้อมูลไปยังผู้ใช้งาน โปรแกรม IsagQ ส่วนการติดต่อกับผู้ใช้งานจะต้องส่งสารไปยังส่วน

การเข้า-ถอดรหัสข้อมูลก่อนเพื่อทำการเข้ารหัสข้อมูลสารที่ต้องการส่ง แต่หากต้องการส่งข้อมูลไปยังผู้ใช้งานโปรแกรม ICQ จะไม่ต้องทำการเข้ารหัสข้อมูลสารที่ต้องการส่ง ส่วนการติดต่อกับผู้ใช้งานจะทำการส่งสารไปยังส่วนการติดต่อกับเน็ตเวิร์กได้ทันที

6.2.2 ส่วนการติดต่อกับเน็ตเวิร์ก

ส่วนการติดต่อกับเน็ตเวิร์กเป็นส่วนที่ทำการติดต่อระหว่างโปรแกรม IsagQ กับเครือข่ายอินเทอร์เน็ต โดยจะทำการส่งข้อมูลออกจากโปรแกรม IsagQ ไปยังเครือข่ายอินเตอร์เน็ตเมื่อผู้ใช้งานทำหน้าที่เป็นผู้ส่ง ในส่วนของผู้ใช้งานที่เป็นหน้าที่เป็นผู้รับเมื่อทำการรับข้อมูลจากเครือข่ายอินเทอร์เน็ตจะต้องตรวจสอบข้อมูลที่ได้รับว่ามีการเข้ารหัสข้อมูลหรือไม่ โดยสามารถตรวจสอบได้จากผู้ที่ทำการส่งข้อมูลมาให้หากมีการเข้ารหัสข้อมูลต้องทำการส่งไปยังส่วนการเข้ารหัส-ถอดรหัสข้อมูลเพื่อทำการถอดรหัส แต่ถ้าข้อมูลไม่มีการเข้ารหัส ข้อมูลจะถูกส่งไปยังส่วนการติดต่อกับผู้ใช้งานเพื่อแสดงข้อมูลที่ได้รับ

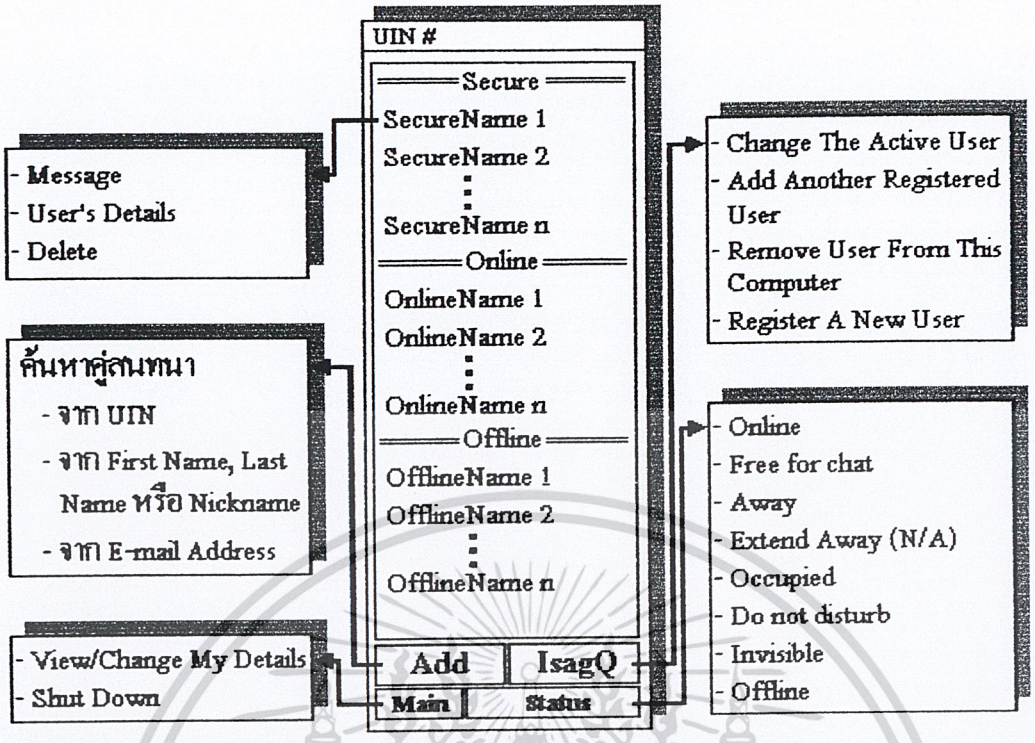
6.2.3 ส่วนการเข้า-ถอดรหัสข้อมูล

ในส่วนการเข้า-ถอดรหัสข้อมูลจะใช้งานเฉพาะการสื่อสารโดยตรงกับผู้ใช้งานโปรแกรม IsagQ เท่านั้น โดยเมื่อผู้ส่งต้องการส่งสารที่ต้องการความปลอดภัยไปยังผู้รับซึ่งใช้โปรแกรม IsagQ สารของผู้ส่งจากส่วนการติดต่อกับผู้ใช้งานจะถูกส่งมายังส่วนการเข้า-ถอดรหัสข้อมูลเพื่อทำการเข้ารหัสข้อมูลสาร จากนั้นสารที่ถูกเข้ารหัสจะถูกส่งไปยังส่วนการติดต่อกับเน็ตเวิร์กเพื่อส่งสารไปให้ผู้รับ

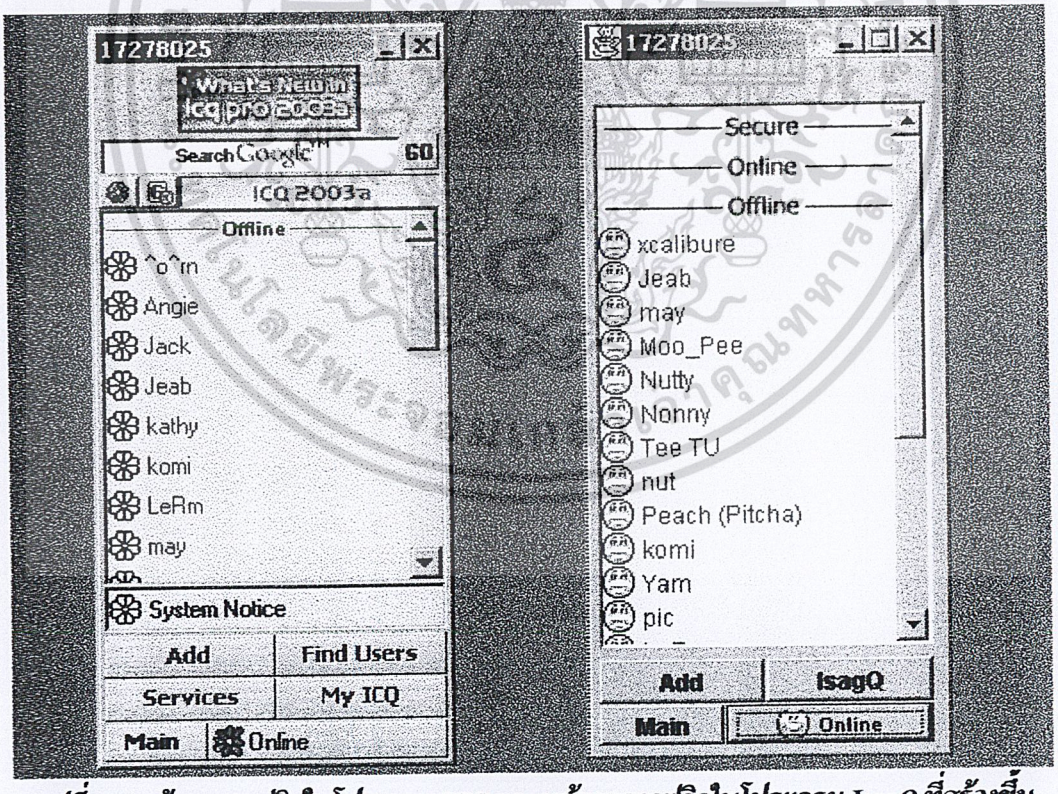
เมื่อผู้รับซึ่งใช้โปรแกรม IsagQ ได้รับสารที่มีการเข้ารหัส ส่วนการติดต่อกับเน็ตเวิร์กจะส่งสารนั้นมายังส่วนการเข้า-ถอดรหัสข้อมูลเพื่อทำการถอดรหัส จากนั้นสารจะถูกส่งไปยังส่วนการติดต่อกับผู้ใช้งานของผู้รับ ซึ่งทำให้สารในการสนทนาสามารถอ่านได้เฉพาะผู้รับและผู้ส่งเท่านั้น

6.3 การออกแบบและสร้างส่วนการติดต่อกับผู้ใช้งาน

ในการออกแบบส่วนการติดต่อกับผู้ใช้งานจะใช้โครงสร้างโปรแกรม ICQ เป็นต้นแบบ เพื่อให้ผู้งานสามารถใช้งานได้ง่ายเหมือนกับโปรแกรม ICQ แต่โปรแกรม IsagQ ได้ทำการลดฟังก์ชันการทำงานบางอย่างที่ไม่ค่อยได้ใช้งานออก และทำการเพิ่มส่วนของการแสดงสถานะสำหรับแสดงว่ามีผู้ใช้งานคนใดใช้โปรแกรม ICQ และผู้ใช้งานคนใดใช้งานโปรแกรม IsagQ โดยจะเพิ่มในส่วนของการแสดงสถานะของ Secure เพื่อแสดงว่าเป็นผู้ใช้งานโปรแกรม IsagQ ซึ่งในการติดต่อกันโดยตรงนั้นจะใช้การเข้ารหัสข้อมูลแบบ DES เพื่อความปลอดภัยของผู้ใช้งาน



รูปที่ 6.4 การออกแบบหน้าจอสำหรับติดต่อกับผู้ใช้งานในโปรแกรม IsagQ



รูปที่ 6.5 หน้าจอกราฟริกในโปรแกรม ICQ และหน้าจอกราฟริกในโปรแกรม IsagQ ที่สร้างขึ้น

เมื่อได้ทำการออกแบบโครงสร้างส่วนของการติดต่อกับผู้ใช้งานแล้ว เราได้ทำการสร้างโดยใช้ภาษาจาวาซึ่งการสร้างโปรแกรมจะใช้ class library ของจาวาที่ให้มาคือ java.swing.*; เป็นส่วนมาก ซึ่งทำ

ให้มีลักษณะคล้ายกับโปรแกรม ICQ เพื่อสะดวกของผู้ใช้งาน และสามารถแสดงผลได้เหมือนกันในทุกแพลตฟอร์มที่สนับสนุน JVM ในรูปที่ 6.5 จะแสดงหน้าจอกราฟริกในส่วนการติดต่อกับผู้ใช้งานที่ได้สร้างขึ้นจากภาษาจาวา แต่จากรูปแรกจะเห็นว่ายังไม่มีการเข้าใช้งานของผู้ใช้ที่มีความปลอดภัย ซึ่งในส่วนนี้ก็จะยังไม่มีการแสดงชื่อใดๆ นอกจากการออนไลน์แบบธรรมดาและการออฟไลน์ ส่วนรูปต่อมาจะมีการแสดงสถานะที่มีผู้ใช้งานแบบปลอดภัยออนไลน์ ซึ่งจะแสดงรายชื่อออกมาไม่เหมือนเดิมคือจะแสดงชื่อเป็นอีกระดับหนึ่งคืออยู่ในโหมดของการรักษาความปลอดภัย (Secure)

6.4 การติดต่อเครือข่ายอินเทอร์เน็ต

ในการเขียนโปรแกรมสำหรับติดต่อเครือข่ายอินเทอร์เน็ต จะเริ่มจากการเขียนโปรแกรมเพื่อทำการติดต่อขอใช้บริการจากเซิร์ฟเวอร์ ซึ่งจะต้องแก้ไขและปรับปรุงการทำงานอย่างต่อเนื่องไปโดยเริ่มที่การล็อกอินเพื่อขอเข้าใช้บริการ การเรียกขอรายการคอนแทคลิสต์ การร้องขอข้อมูลรายการผู้ใช้งานที่ทำการออนไลน์จากเซิร์ฟเวอร์ การเปลี่ยนแปลงสถานะ การส่งข้อความผ่านทางเซิร์ฟเวอร์ การค้นหาคู่สนทนา การเรียกขอดูข้อมูลของคู่สนทนาเป็นต้น ซึ่งสามารถดูรายละเอียดคำสั่งได้จากบทที่ 5

6.5 การสร้างคีย์สำหรับทำการเข้ารหัสข้อมูล

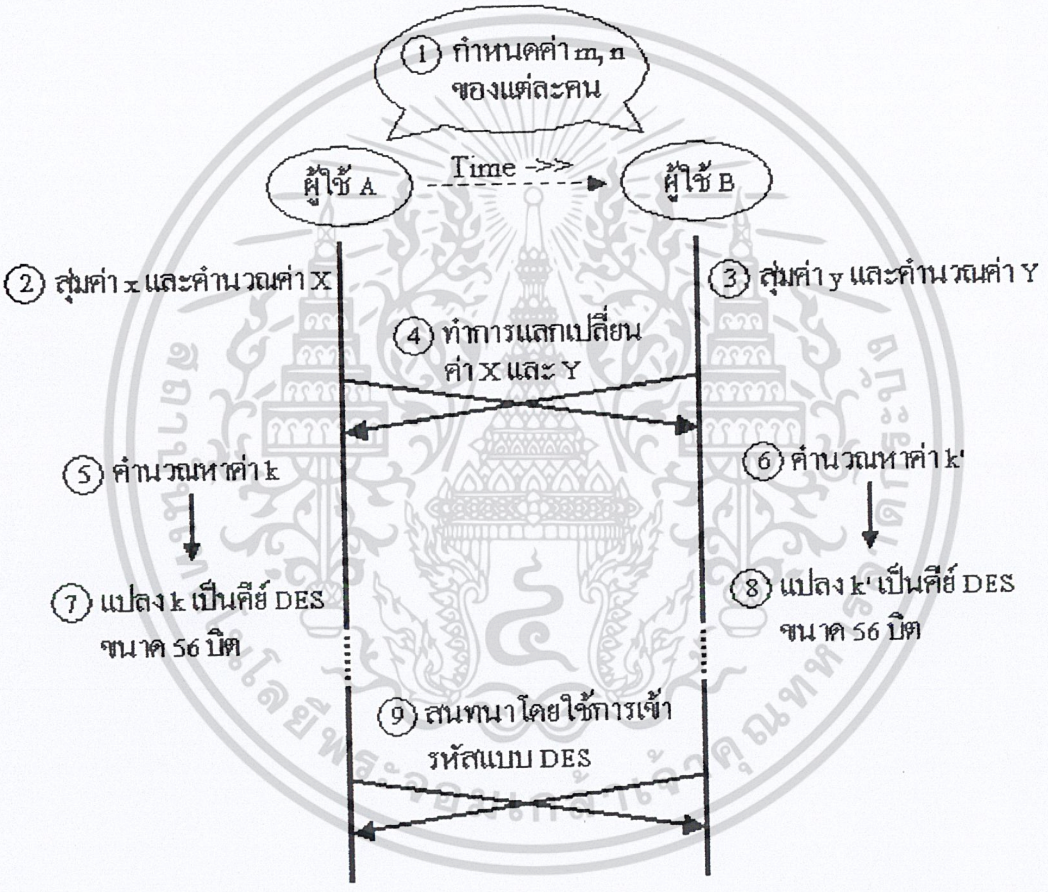
เมื่อผู้ใช้งานโปรแกรม IsagQ ทำการออนไลน์จะต้องทำการสร้างคีย์ความลับระหว่างกัน โดยในโครงงานนี้จะใช้วิธี Deffie-Hellman ในการสร้างคีย์ความลับแล้วนำค่าคีย์ที่ได้มาแปลงเป็นคีย์ DES ขนาด 64 บิต และใช้คีย์ที่ได้นี้ทำการเข้ารหัสข้อความที่ต้องการรับส่งระหว่างกัน ทำให้ผู้ใช้งานโปรแกรม IsagQ ทั้งสองฝั่งสามารถทำการรับส่งสารได้อย่างปลอดภัย

ลำดับขั้นตอนในการรับส่งคีย์ต่อไปนี้จะแสดงถึงเหตุการณ์ที่ผู้ใช้ A และผู้ใช้ B ซึ่งเป็นผู้ใช้งานโปรแกรม IsagQ ทำการสร้างคีย์ DES ด้วยวิธี Deffie-Hellman เพื่อทำการรับส่งสารได้อย่างปลอดภัย ซึ่งมีขั้นตอนต่อไปนี้

1. ผู้ใช้ A และผู้ใช้ B ทำการกำหนดค่า m และ n ของแต่ละผู้ใช้โดยที่ $1 < n < m$ โดยเลขทั้งสองนี้ไม่จำเป็นต้องปกปิด
2. ผู้ใช้ A สุ่มตัวเลขที่มีค่ามากๆ มาตัวหนึ่งกำหนดให้เป็นค่า x แล้วหาค่า $X = n^x \bmod m$ และให้ผู้ใช้ A เก็บค่า x เอาไว้เป็นความลับ
3. ให้ผู้ใช้ B ทำเหมือนกับผู้ใช้ A ในข้อ 2 สุ่มตัวเลขที่มีค่ามากๆ มาตัวหนึ่งกำหนดให้เป็นค่า y แล้วหาค่า $Y = n^y \bmod m$ และให้ผู้ใช้ B เก็บค่า y เป็นความลับ
4. ให้ผู้ใช้ A และผู้ใช้ B ทำการแลกค่า X และ Y กัน
5. ผู้ใช้ A คำนวณหาค่า $k = Y^x \bmod m$
6. ผู้ใช้ B คำนวณหาค่า $k' = X^y \bmod m$
7. ผู้ใช้ A ทำการแปลงค่า k ที่คำนวณได้ให้เป็นค่าขนาด 56 บิต

8. ผู้ใช้ B ทำการแปลงค่า k' ที่คำนวณได้ให้เป็นค่าขนาด 56 บิต ซึ่งค่าคีย์ซึ่งแปลงจากค่า k ของผู้ใช้ A และค่าคีย์ซึ่งแปลงจากค่า k' ของผู้ใช้ B จะมีค่าเท่ากัน ซึ่งสามารถนำค่าที่ได้มาเป็นคีย์ในการเข้ารหัสแบบ DES ในการสนทนา
9. ผู้ใช้ A และผู้ใช้ B ทำการสนทนาโดยการใช้คีย์ DES ที่แปลงมาจากค่า k และ k' ซึ่งมีค่าเท่ากันคือ $n^{xy} \bmod m$ ใช้เป็นคีย์ความลับในการสนทนาระหว่างกัน

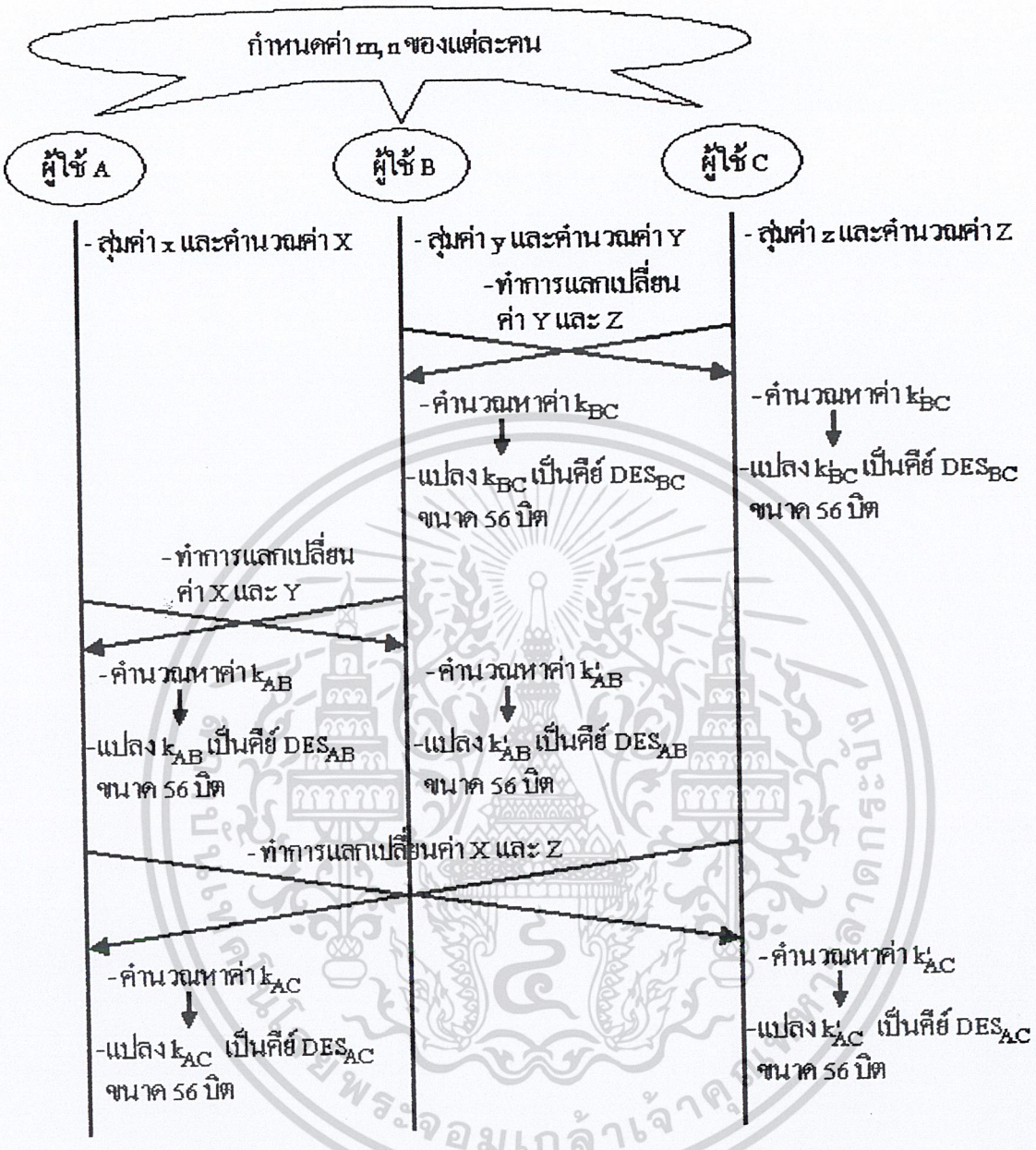
จากการทำงานดังกล่าวนี้จะเกิดขึ้นทุกครั้งเมื่อผู้ใช้งานโปรแกรม IsagQ ทำการออนไลน์และหากผู้ใช้งานผู้อื่นซึ่งใช้โปรแกรม ICQ ก็จะไม่ต้องทำการรับส่งคีย์นี้ สามารถทำการรับส่งสารโดยตรงได้ทันที แต่การรับส่งสารจะไม่ปลอดภัย



รูปที่ 6.6 ขั้นตอนในการสร้างคีย์ความลับสำหรับสนทนาระหว่างผู้ใช้โปรแกรม IsagQ

จากการรับส่งคีย์ดังกล่าวที่ได้ออกแบบนี้ โปรแกรม IsagQ จะต้องทำการสร้างคีย์สำหรับสนทนาระหว่างผู้ใช้งานใหม่ทุกครั้งเมื่อผู้ใช้งานโปรแกรม IsagQ ทำการออนไลน์ ซึ่งการสร้างคีย์ความลับขึ้นมาใหม่นี้จะทำให้การสนทนาเป็นความลับระหว่างผู้ใช้งานเพียง 2 คนเท่านั้น

การรับส่งคีย์ในรูปที่ 6.7 จะเป็นการแสดงการสร้างคีย์เมื่อผู้ใช้ A, B และ C ต้องการสนทนาระหว่างกัน ซึ่งในการใช้งานจริงนั้นจะต้องมีคีย์ความลับ 3 คู่คือ คีย์ระหว่าง A กับ B, คีย์ระหว่าง A กับ C และคีย์ระหว่าง B กับ C เพื่อให้การสนทนานั้นเป็นความลับระหว่างผู้ใช้งาน 2 คน



รูปที่ 6.7 การสร้างคีย์ความลับเมื่อมีผู้ใช้งานโปรแกรม IsagQ หลายคน

6.6 การรวมส่วนประกอบของโปรแกรมเข้าด้วยกัน

เมื่อสามารถเขียนโปรแกรมพื้นฐานได้ครบทั้งสามส่วนแล้ว ก็จะนำมาทดลองใช้งาน โดยจะเริ่มจากการนำส่วนการติดต่อกับผู้ใช้งานและส่วนการติดต่อกับเน็ตเวิร์กมารวมกันก่อน ซึ่งจะต้องปรับปรุงให้ส่วนการติดต่อกับผู้ใช้งานติดต่อกับเครือข่ายอินเทอร์เน็ตผ่านทางส่วนการติดต่อกับเน็ตเวิร์กได้ จากนั้นจึงทำหารทดลองใช้คำสั่งต่างๆ ตามที่ได้ออกแบบกับเซิร์ฟเวอร์เช่น การล็อกอินเพื่อขอเข้าใช้บริการ การเรียกขอรายการคอนแทคลิสต์ การร้องขอข้อมูลรายการผู้ใช้งานที่ทำการออนไลน์จากเซิร์ฟเวอร์ การเปลี่ยนแปลงสถานะ การค้นหาคู่สนทนา การเรียกขอดูข้อมูลของคู่สนทนาเป็นต้น

จากนั้นจึงทำการเขียนโปรแกรมการรับส่งสารผ่านทางเซิร์ฟเวอร์และการรับส่งสารโดยตรง ซึ่งในส่วนนี้จะทำให้ได้โปรแกรมที่มีลักษณะการทำงานเหมือนกับโปรแกรม ICQ สามารถสนทนากับผู้ใช้งานโปรแกรม ICQ ได้

เมื่อรวมการทำงานในส่วนการติดต่อกับผู้ใช้งานและส่วนการติดต่อกับเน็ตเวิร์กเข้ากันได้แล้ว จะนำส่วนการเข้า-ถอดรหัสข้อมูลเข้ามารวมกับโปรแกรมในข้างต้น โดยในขั้นตอนนี้จะต้องทดลองโดยมีผู้ใช้งานโปรแกรม IsagQ อย่างน้อย 2 คนทำการออนไลน์และต่างมีรายชื่อของอีกคนหนึ่งอยู่ใน contact list จากนั้นจึงทำการทดลองการสร้างคีย์ของโปรแกรมตามลักษณะที่ได้กล่าวมาแล้ว เมื่อสามารถทำการสร้างคีย์ได้แล้ว จึงทดลองนำสารที่ต้องการส่งระหว่างผู้ใช้งานโปรแกรม IsagQ มาทำการเข้ารหัสและทดลองส่งสารให้กับผู้ใช้งานโปรแกรม IsagQ เมื่อผู้ใช้งานโปรแกรม IsagQ สามารถสนทนากันได้อย่างถูกต้องแล้วจึงทำการแก้ไขและตกแต่งโปรแกรมให้น่าใช้งานมากขึ้น

บทที่ 7
การทดสอบโปรแกรม IsagQ

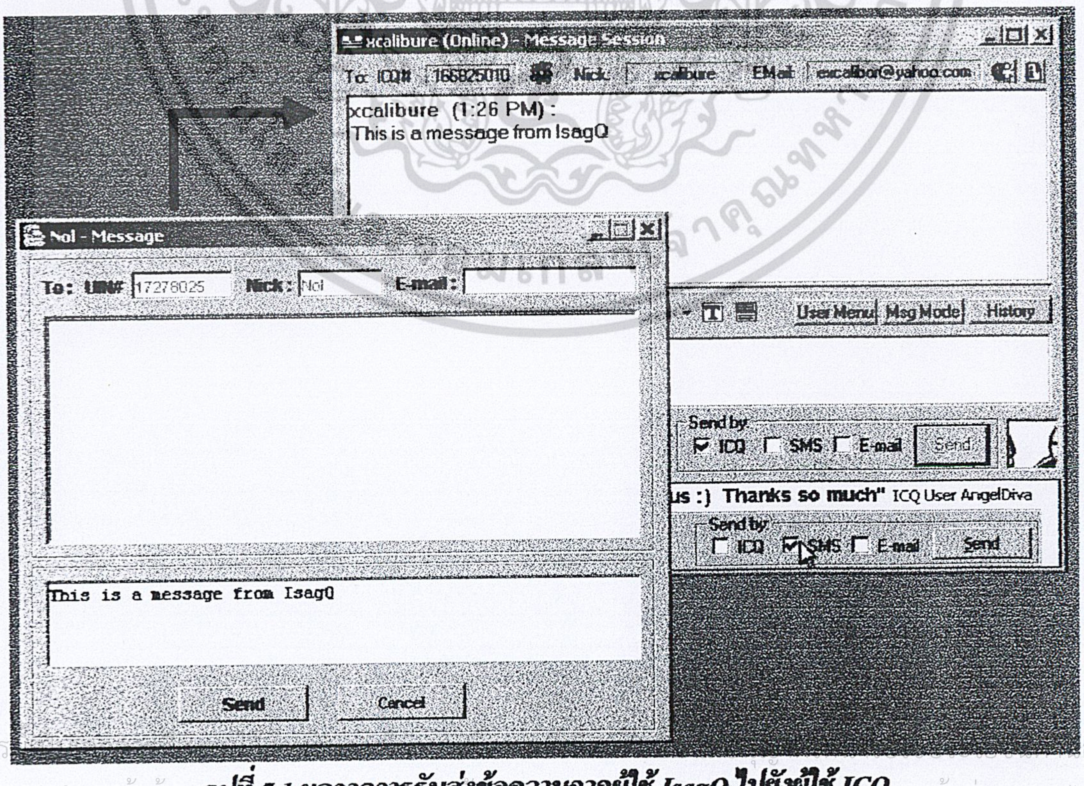
การทดสอบการรับส่งข้อความในโปรแกรม IsagQ สามารถทำการทดสอบได้หลายวิธี แต่สำหรับในโครงงานนี้จะทำการทดสอบเพื่อแสดงให้เห็นถึงความปลอดภัยในการรับข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ ซึ่งเป็นการรับส่งข้อความซึ่งไม่มีการเข้ารหัสข้อมูล และการรับส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ IsagQ ซึ่งเป็นการรับส่งข้อความซึ่งไม่มีการเข้ารหัสข้อมูล

7.1 การทดสอบการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ ICQ

ขั้นตอนการทดสอบโปรแกรม IsagQ ในการรับส่งข้อความกับผู้ใช้โปรแกรม ICQ นี้มีการทำงานที่ไม่ยุ่งยาก เพียงแค่ทำการรับส่งข้อความตามปกติแล้วใช้โปรแกรมดักจับข้อมูล (โดยในโครงงานนี้ใช้โปรแกรม Ethereal version 0.9.7) มาทำการตรวจจับข้อมูลในแพ็กเก็ตต่างๆ ซึ่งจะได้ข้อมูลที่ทำการรับส่งระหว่างกันได้

7.1.1 การทดสอบการส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ

7.1.1.1 ผลจากการรับส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ

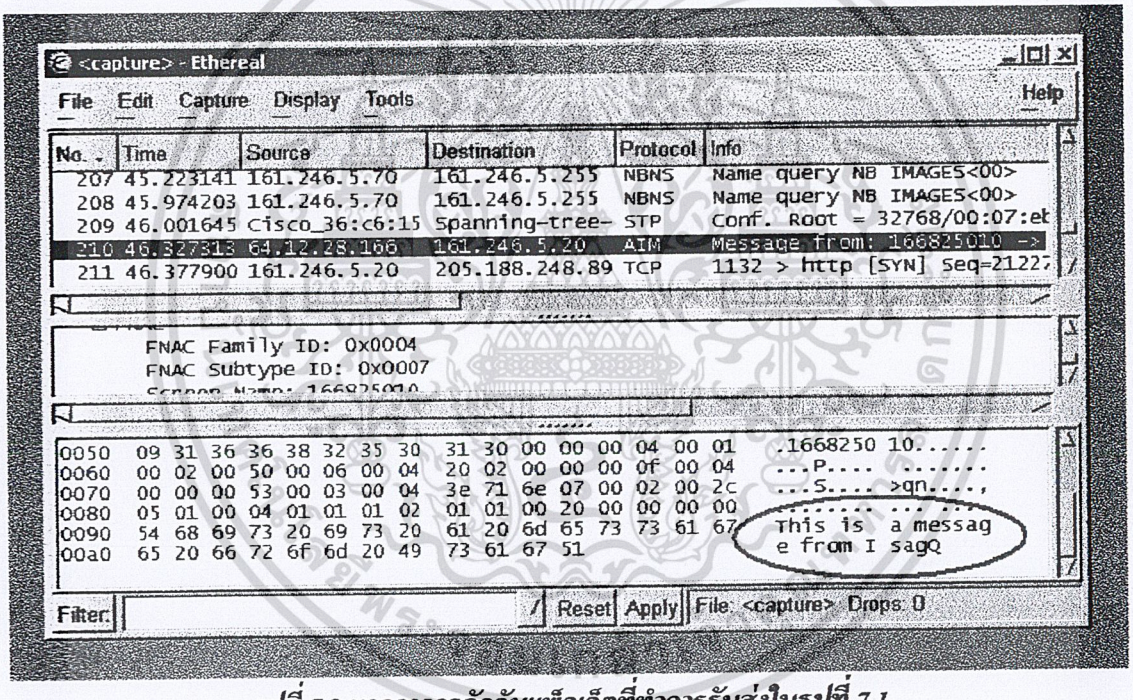


รูปที่ 7.1 ผลจากการรับส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ

เมื่อทำการรับส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ จะได้ผลของการรับส่งข้อความดังรูปที่ 7.1 ซึ่งแสดงถึงการส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ สามารถทำได้อย่างถูกต้อง ทำให้ข้อความที่ผู้ใช้ ICQ ได้รับเหมือนกับข้อความที่ผู้ใช้ IsagQ ทำการส่ง

7.1.1.2 ผลจากการดักจับแพ็กเก็ตในการส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ

ในการใช้โปรแกรม Ethereal ในการดักจับแพ็กเก็ตของข้อความที่ทำการรับส่งนี้ เมื่อเราตรวจสอบรายละเอียดในแพ็กเก็ตจะสามารถอ่านข้อความที่ทำการรับส่งได้ออกมาโดยง่าย เนื่องจากการรับส่งข้อความไม่มีการเข้ารหัสข้อมูลซึ่งแสดงถึงความไม่ปลอดภัยในเรื่องของข้อมูลที่ทำการสื่อสารกัน (ข้อมูลที่ดักจับได้สามารถอ่านได้เช่นเดียวกับข้อมูลที่ทำการรับส่งระหว่างผู้ใช้ในรูปที่ 7.1) โดยในรูปที่ 7.2 จะแสดงการดักจับข้อความที่ทำการรับส่งได้ในรูปที่ 7.1

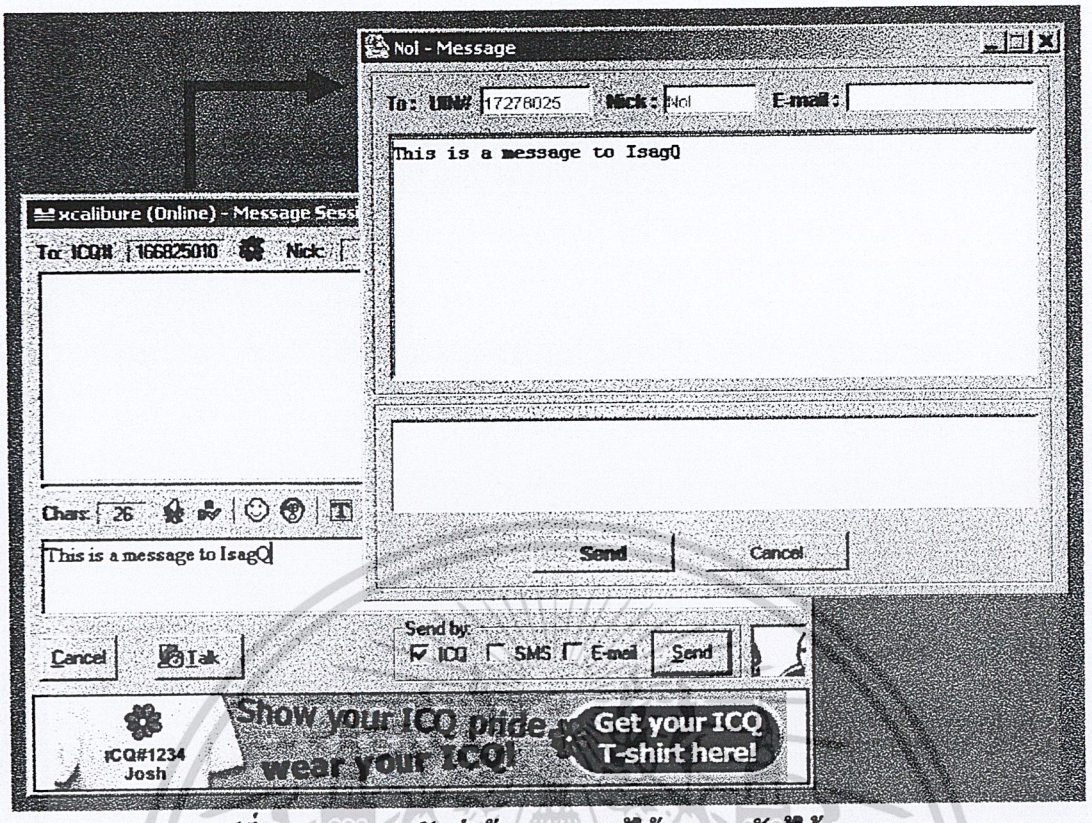


รูปที่ 7.2 ผลจากการดักจับแพ็กเก็ตที่ทำการรับส่งในรูปที่ 7.1

7.1.2 การทดสอบการส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ

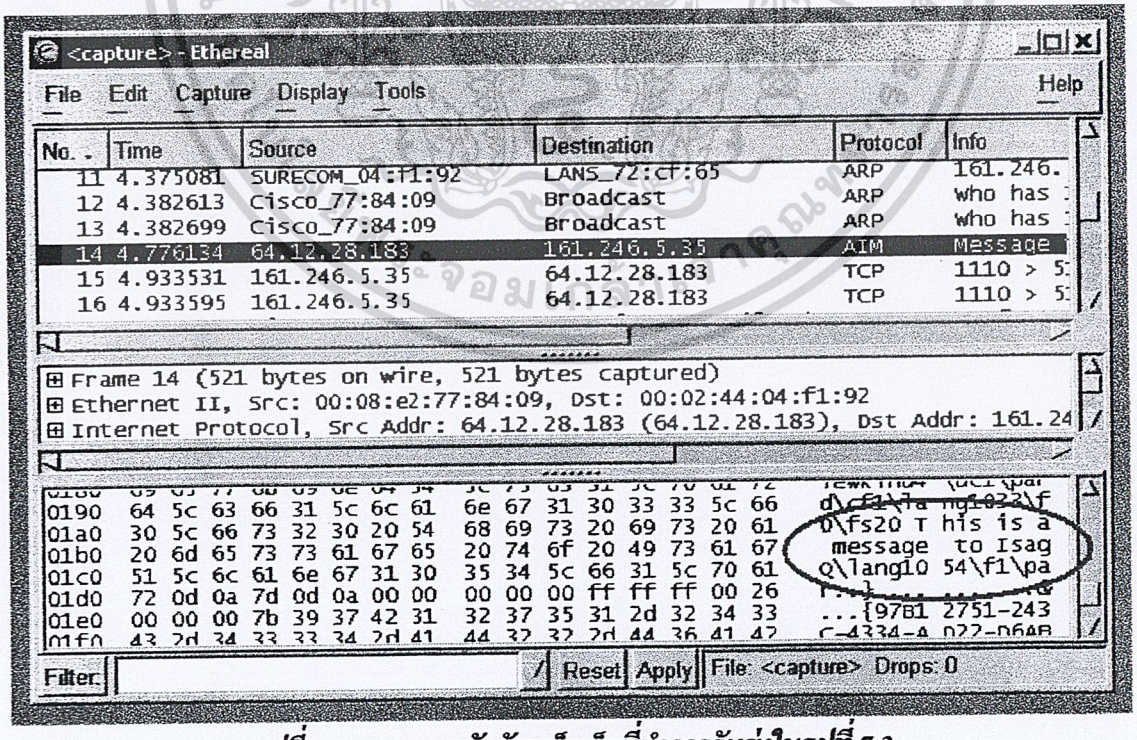
7.1.2.1 ผลจากการรับส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ

เมื่อทำการรับส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ จะได้ผลของการรับส่งข้อความดังรูปที่ 7.3 ซึ่งแสดงถึงการรับข้อความของผู้ใช้ IsagQ จากผู้ใช้ ICQ สามารถทำได้อย่างถูกต้อง ทำให้ข้อความที่ผู้ใช้ ICQ ได้ส่งเหมือนกับข้อความที่ผู้ใช้ IsagQ ทำการรับ



รูปที่ 7.3 ผลจากการรับส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ

7.1.2.2 ผลจากการดักจับแพ็กเก็ตในการส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ



รูปที่ 7.4 ผลจากการดักจับแพ็กเก็ตที่ทำการรับส่งในรูปที่ 7.3

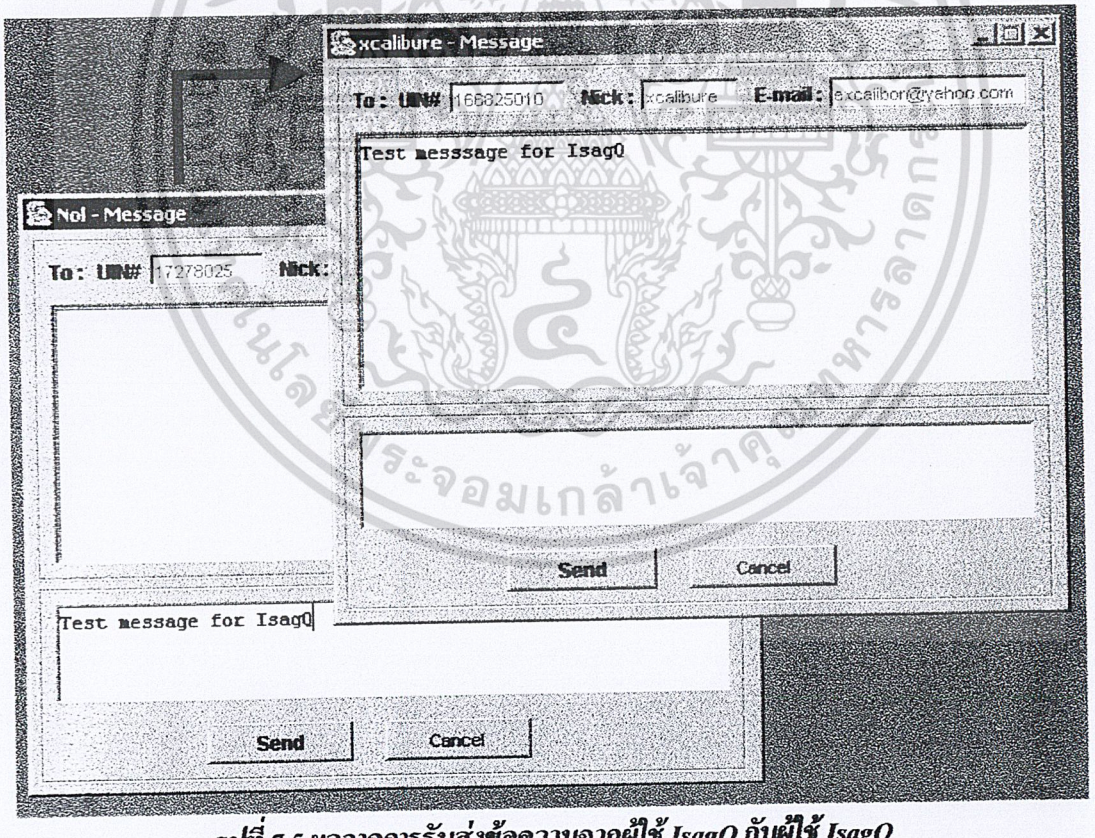
เมื่อเราตรวจสอบรายละเอียดในแพ็กเก็ตโดยใช้โปรแกรม Ethereal จะสามารถอ่านข้อความที่ทำการรับส่งได้โดยง่ายเหมือนกับการดักจับแพ็กเก็ตในการส่งข้อความจากผู้ใช้ IsagQ ไปยังผู้ใช้ ICQ เนื่องจากการรับส่งข้อความไม่มีการเข้ารหัสข้อมูล โดยในรูปที่ 7.4 จะแสดงการดักจับข้อความที่ทำการรับส่งในรูปที่ 7.3 ซึ่งสามารถอ่านข้อความได้โดยง่ายเช่นเดียวกับในรูปที่ 7.2

ดังนั้นในการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ ICQ จะยังคงไม่มีการใช้การเข้ารหัสข้อมูล เพื่อให้ผู้ใช้ IsagQ ยังคงสามารถทำการสนทนากับผู้ใช้ ICQ ได้ดังเดิม

7.2 การทดสอบการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ

ในการทดสอบการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ นั้น ก่อนที่ผู้ใช้จะทำการรับส่งข้อความ จะต้องนำข้อความที่ต้องการรับส่งมาทำการเข้า-ถอดรหัสข้อมูลก่อนจึงทำการรับส่งได้ ทำให้ข้อความที่ส่งออกไปไม่สามารถอ่านได้ เพราะจะต้องมีการถอดรหัสข้อมูลก่อนจึงจะสามารถอ่านได้ ดังนั้นหากไม่รู้ค่าคีย์ที่ใช้ในการถอดรหัสนี้ ก็ไม่สามารถอ่านข้อความนี้ได้

7.2.1 ผลจากการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ

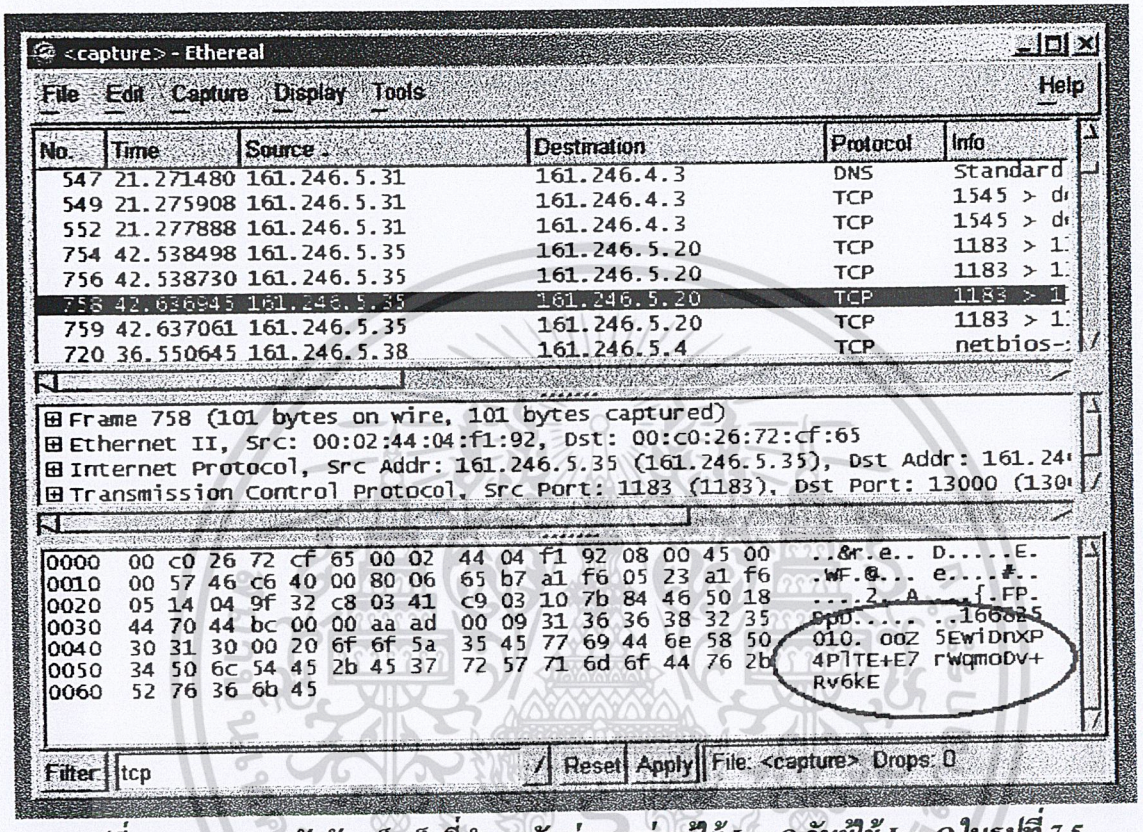


รูปที่ 7.5 ผลจากการรับส่งข้อความจากผู้ใช้ IsagQ กับผู้ใช้ IsagQ

เมื่อทำการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ จะได้ผลของการรับส่งข้อความดังรูปที่ 7.5 ซึ่งแสดงถึงการส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ สามารถทำได้อย่างถูกต้องเช่นเดียวกับ

การรับส่งข้อความกับผู้ใช้ ICQ ซึ่งข้อความที่ผู้ใช้ได้รับมีผลลัพธ์เช่นเดียวกับข้อความที่ผู้ส่งทำการส่งออกไปให้

7.2.2 ผลจากการดักจับแพ็กเก็ตในการส่งข้อความจากผู้ใช้ ICQ มายังผู้ใช้ IsagQ



รูปที่ 7.6 ผลจากการดักจับแพ็กเก็ตที่ทำการรับส่งระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ ในรูปที่ 7.5

ในการใช้โปรแกรม Ethereal ดักจับแพ็กเก็ตของข้อความที่ทำการรับส่งระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ นี้เมื่อเราตรวจสอบรายละเอียดในแพ็กเก็ตจะไม่สามารถอ่านข้อความที่ทำการรับส่งได้ออกมาโดยง่ายเหมือนกับการรับส่งข้อความกับผู้ใช้โปรแกรม ICQ เนื่องจากในการรับส่งข้อความระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ จะมีการเข้ารหัสข้อมูลเพื่อให้เกิดความปลอดภัยของข้อความที่ทำการรับส่งระหว่างกัน โดยในรูปที่ 7.6 จะแสดงการดักจับข้อความที่ทำการรับส่งระหว่างผู้ใช้ IsagQ กับผู้ใช้ IsagQ ในรูปที่ 7.5

บทที่ 8
สรุปผลการพัฒนาโครงงาน

8.1 คุณสมบัติของโปรแกรม

จากการพัฒนาโครงงานสามารถทำการพัฒนาโปรแกรมให้มีความสามารถดังนี้

Input specification

- รับค่าการพิมพ์และคำสั่งการทำงานผ่านทางคีย์บอร์ดและเมาส์

Output specification

- แสดงผลออกทางจอภาพโดยเป็นกราฟิก

Function specification

- สามารถทำการเรียก contact list จากผู้ใช้งานที่ทำการลงทะเบียนจากโปรแกรม ICQ เดิมได้
- สามารถทำการรับส่งสาร โดยเข้า-ถอดรหัสข้อมูลระหว่างผู้ใช้งานโปรแกรม IsagQ และผู้ใช้งานโปรแกรม IsagQ ได้
- สามารถทำการรับส่งสาร โดยไม่มีการเข้า-ถอดรหัสข้อมูลระหว่างผู้ใช้งานโปรแกรม IsagQ และผู้ใช้งานโปรแกรม ICQ ได้
- มีการใส่ล็อกอินและรหัสผ่านของผู้ใช้งานสำหรับใช้งานโปรแกรม IsagQ
- สามารถแสดงสถานะการใช้งานของผู้ใช้โปรแกรม IsagQ ได้ทั้ง Online, Free for chat, Away, Extend Away (N/A), Occupied, Do not disturb, Invisible และ Offline
- สามารถดูรายละเอียดข้อมูลประจำตัวของผู้ใช้งานได้
- แสดงสถานะของคู่สนทนาโดยแบ่งเป็น Secure สำหรับผู้ใช้ IsagQ ซึ่งทำการออนไลน์, Online สำหรับผู้ใช้ ICQ ซึ่งทำการออนไลน์และ Offline สำหรับผู้ใช้ที่ทำการออฟไลน์ได้
- ค้นหาผู้ใช้งานได้จากทางหมายเลขประจำตัวผู้ใช้งาน (UIN), ชื่อและอีเมล์แอดเดรส

8.2 ประโยชน์ของการพัฒนาโครงงาน

- โปรแกรม IsagQ ยังคงสามารถทำการสื่อสารข้อความกับผู้ใช้โปรแกรม ICQ ได้ง่ายและสะดวกเช่นเดียวกับการใช้งานโปรแกรม ICQ
- โปรแกรม IsagQ สามารถทำงานข้ามแพลตฟอร์มไปยังแพลตฟอร์มอื่นที่สนับสนุน JVM ได้ตามคุณสมบัติของภาษาจาวา
- การรับส่งสารระหว่างผู้ใช้งานโปรแกรม IsagQ จะทำการเข้ารหัสซึ่งปลอดภัยต่อการดักจับข้อความมาอ่านจากบุคคลอื่นได้

8.3 ข้อจำกัดของโครงงาน

- โปรแกรมที่ได้จัดทำขึ้นทำงานช้ากว่าโปรแกรม ICQ เนื่องจากจะต้องมีการสร้างคีย์และรับส่งข้อความที่มีการเข้ารหัสในการสนทนากับผู้ใช้ IsagQ
- โปรแกรมที่จัดทำยังไม่มีเสถียรภาพที่ดีพอ

8.4 ข้อเสนอแนะสำหรับผู้นำโครงงานไปพัฒนา

- ควรปรับปรุงให้มีการรับส่งคีย์และการเข้ารหัสข้อมูลที่รวดเร็วและมีความปลอดภัย
- ควรปรับปรุงการทำงานของโปรแกรมให้มีเสถียรภาพมากกว่านี้

ภาคผนวก ก.
ความรู้เกี่ยวกับโปรแกรม ICQ

โปรแกรม ICQ ที่ใช้งานกันอยู่ในปัจจุบันเป็นชื่อที่พ้องเสียงมาจากคำว่า I seek you ซึ่งหมายถึงเครื่องมือที่ใช้ในการค้นหาเพื่อนหรือคู่สนทนา โดยใช้การติดต่อกันผ่านทางอินเตอร์เน็ต ซึ่งเราสามารถรับทราบได้ว่าคู่สนทนาของเราออนไลน์อยู่หรือไม่ หากคู่สนทนาออนไลน์ก็สามารถสนทนาโต้ตอบกันไปมาได้ทันที หรืออาจทำการฝากข้อความไว้ให้กับคู่สนทนาของเราในกรณีที่คู่สนทนาไม่ได้ออนไลน์อยู่ในเวลานั้น

การทำงานของโปรแกรม ICQ มีลักษณะการทำงานที่เรียกว่า Internet Instant-Messaging โดยผู้ที่สมัครเป็นสมาชิกของ ICQ จะได้รับหมายเลขประจำตัวของแต่ละคนเรียกว่า ICQ Number (ICQ#) หรือ User Identification Number (UIN) เพื่อใช้ในการติดต่อสำหรับใช้งานโปรแกรม ICQ ซึ่งในอนาคตคาดว่านอกจากจะต้องมีจดหมายอิเล็กทรอนิกส์ของตนเองแล้ว ก็อาจต้องมีหมายเลข ICQ ที่เป็นเสมือนหมายเลขอ้างอิงอีกอย่างหนึ่งของบุคคลที่ใช้งานอินเตอร์เน็ตอีกด้วย

ก.1 ความเป็นมาของโปรแกรม ICQ

บริการ ICQ เริ่มต้นมาจากแนวคิดของบริการชื่อ Buddy List จากบริษัท America Online หรือ AOL ซึ่งเป็นบริษัท ISP รายใหญ่ที่สุดของอเมริกา โดยพัฒนาขึ้นเพื่อให้บริการแก่สมาชิกของ AOL ได้สนทนาและติดต่อสื่อสารกัน ซึ่งต่อมาได้มีบริษัทที่ทำธุรกิจบนอินเทอร์เน็ตเช่น Yahoo, เน็ตสเคปและอื่นๆ ได้พัฒนาโปรแกรมในลักษณะของ Buddy List ขึ้นมาอีกมากมาย แต่ก็ไม่ได้รับความนิยมมาก

แต่ข้อจำกัดของ Buddy List ของ AOL (ปัจจุบันได้พัฒนาไปเป็นโปรแกรม AOL Instant Messaging หรือ AIM) ก็คือสามารถใช้งานได้เฉพาะกลุ่มของสมาชิก AOL เท่านั้น ส่วนผู้ใช้ที่อยู่ในประเทศอื่นๆ จะไม่มีโอกาสได้ใช้บริการนี้ ซึ่งต่อมาบริษัท Mirabilis ได้พัฒนาโปรแกรม 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 ดังนั้นในปัจจุบัน Mirabilis จึงได้กลายเป็นส่วนหนึ่งของ AOL ไป (โดยเปลี่ยนชื่อไปเป็น ICQ Inc.) และจัดว่าเป็นโปรแกรมที่มีผู้ใช้งานมากที่สุดในโลกตัวหนึ่ง โดยมีผู้ลงทะเบียนไว้แล้วถึง 300 ล้านคน (ถึงแม้จะตัดผู้ใช้งานที่ลงทะเบียนซ้ำซ้อนหลายๆ ครั้งออกไปบ้างแต่ก็ยังถือว่ามาก) และในเวลาขณะหนึ่งๆ จะมีผู้ใช้งาน ICQ พร้อมกันทั่วโลกไม่ต่ำกว่าสิบล้านคน

ก.2 การติดตั้งและการใช้งานโปรแกรม ICQ

โปรแกรม ICQ ที่ติดตั้งในส่วนไคลเอนต์จะเป็นโปรแกรมที่แจกฟรี โดยผู้ใช้งานสามารถดาวน์โหลดได้ฟรีที่เว็บไซต์ http://www.mirabilis.com ซึ่งไฟล์โปรแกรมในปัจจุบันมีด้วยกันหลายเวอร์ชัน ปัจจุบันเวอร์ชันล่าสุดคือ ICQ2003a เมื่อทำการดาวน์โหลดโปรแกรม ICQ เรียบร้อยแล้วก็จะเริ่มเข้าสู่ขั้นตอนการติดตั้งโปรแกรม icqpro2003a.exe (ซึ่งชื่อโปรแกรมอาจมีการเปลี่ยนตามเวอร์ชันของ ICQ) แล้วดำเนินการติดตั้งเช่นเดียวกับโปรแกรมอื่นๆ

จากขั้นตอนการติดตั้งโปรแกรม ICQ ผู้ใช้จะต้องทำการลงทะเบียนก่อนเริ่มการใช้งาน โดยคลิกที่โปรแกรม icq.exe หรือคลิกที่ปุ่ม Start -> Programs -> ICQ -> ICQ ซึ่งถ้าเป็นการใช้งานครั้งแรกจะเป็นการติดต่อไปยังเซิร์ฟเวอร์ Mirabilis เพื่อขอลงทะเบียนจากนั้นจึงกรอกรายละเอียดต่างๆ ที่ใช้ในการลงทะเบียน ซึ่งประกอบด้วยรายละเอียดต่างๆ (เช่น ชื่อ-นามสกุล, วันเดือนปีเกิด, เพศ, อาชีพ, ภาษา, และประเทศที่อยู่ เป็นต้น) ส่วนสำคัญที่ต้องระมัดระวังในเรื่องการลงทะเบียนก็คือการกำหนดรหัสผ่าน ซึ่งจะใช้เมื่อต้องการเปลี่ยนแปลงรายละเอียดต่างๆ ของ ICQ โดยจะต้องจำรหัสผ่านที่กำหนดไว้ให้ดีเพื่อให้สามารถนำมาใช้ได้เมื่อโปรแกรมต้องการ เมื่อลงทะเบียน ICQ เสร็จเรียบร้อยแล้ว ผู้ใช้จะได้หมายเลขประจำตัวผู้ใช้งาน เพื่อใช้ในการติดต่อกับคนอื่นซึ่งจะไม่ซ้ำกับของคนอื่นโดยเรียกว่า UIN เป็นหมายเลขที่ใช้ในการติดต่อกันซึ่งต้องใช้ควบคู่กับรหัสผ่านตลอดการใช้บริการ ICQ

ในกรณีที่ผู้ใช้งานทำการติดตั้งโปรแกรม ICQ แต่ได้เคยลงทะเบียน ICQ และได้หมายเลขประจำตัวผู้ใช้งาน (UIN) มาแล้ว ก็สามารถขอใช้งานผ่านทาง UIN เดิมได้โดยไม่จำเป็นต้องทำการลงทะเบียนใหม่ แต่ผู้ใช้งานก็จะต้องสามารถจดจำหมายเลข UIN และรหัสผ่านเดิมไว้ได้ด้วย ซึ่งการใช้งานโดยใช้หมายเลข UIN เดิมจะช่วยให้ผู้ใช้งานสามารถติดต่อกับเพื่อนหรือคู่สนทนาเดิมได้โดยทันที ไม่ต้องทำการเพิ่มลงในคอนแทคลิสต์ใหม่

เมื่อเรียกโปรแกรม ICQ ขึ้นมาทำงาน โปรแกรมจะทำการตรวจสอบว่าขณะนี้เครื่องของเราติดต่ออยู่ในเครือข่ายอินเตอร์เน็ตหรือไม่ เมื่อตรวจสอบเสร็จว่าติดต่อกับเครือข่ายอินเทอร์เน็ตอยู่ก็จะเปลี่ยนสัญลักษณ์ที่ด้านขวาของทาสก์บาร์เป็นสัญลักษณ์การออนไลน์ของโปรแกรม ICQ จากนั้นเมื่อผู้ใช้งานทำการออนไลน์แต่ละครั้งโปรแกรมจะทำการส่งข้อมูลหมายเลข IP (IP address) ของผู้ใช้งานออกไปให้กับเซิร์ฟเวอร์กลางรับทราบ เพื่อเป็นการแจ้งให้ทราบว่ามีผู้ใช้งานหมายเลข UIN นี้กำลังเข้าใช้งาน เผื่อว่ามีใครต้องการติดต่อกับเรา หากใช้ ICQ อยู่ก็จะสามารถสอบถาม IP address จากเซิร์ฟเวอร์กลางแล้วใช้ IP address นั้นในการส่งข้อมูลติดต่อกับเราโดยตรงทันที ซึ่งจุดนี้เองเป็นข้อดีที่ทำให้ผู้ใช้ติดต่อกันได้โดยตรงทำให้สามารถสนทนาได้รวดเร็วทันใจ โดยการติดต่อสื่อสารในโปรแกรม ICQ มี 2 รูปแบบด้วยกันคือการติดต่อกันโดยตรง (peer to peer) และการติดต่อผ่านทางเซิร์ฟเวอร์กลางของ ICQ

แต่จากข้อดีของโปรแกรม ICQ ในเรื่องของการรับส่งข้อมูลได้โดยตรงกับคู่สนทนา ก็กลับเป็นข้อเสียของโปรแกรม ICQ และโปรแกรมรับส่งสารด่วนคือในเรื่องของความปลอดภัยเพราะโดยปกติโปรแกรมอื่นจะไม่มีทางให้รู้ถึง IP address ของผู้ใช้อินเตอร์เน็ตรายอื่นได้โดยง่ายๆ จะมีแต่โปรแกรมบนเซิร์ฟเวอร์เท่านั้นที่รับรู้ได้และจะไม่มีการนำ IP address ไปแจกจ่ายผู้ใช้งานคนอื่นง่ายๆ แบบนี้แต่เนื่องจากวิธีการของ ICQ จะต้องการทำการติดต่อโดยตรงจึงต้องมีกลไกที่จะให้อีกฝ่ายรับรู้ IP address ของผู้ที่

จะติดต่อด้วยจุดนี้เองนับเป็นข้อเสียของ ICQ ถึงแม้จะมีความพยายามที่จะปิด IP address ของผู้ติดต่อด้วยก็ตาม ก็ยังมีผู้ไม่ประสงค์ดีทำการดักจับข้อมูลระหว่างการส่ง-รับไปอ่านได้ทำให้ไม่มีความปลอดภัยเท่าที่ควร

ก.3 การพัฒนาและอนาคตของโปรแกรม ICQ

ในช่วงเริ่มแรกโปรแกรม ICQ มาเพื่อให้ใช้สำหรับสนทนาเท่านั้น แต่เมื่อโปรแกรม ICQ ได้รับความนิยมก็ส่งผลให้ทางผู้ผลิตได้พัฒนาโปรแกรม ICQ ให้มีความสามารถพื้นฐานเพิ่มมากขึ้น เพื่อตอบสนองการใช้งานของผู้ใช้บริการเช่น มีการจัดเก็บข้อความที่ได้สนทนาแล้ว การรับส่งไฟล์ และการเก็บรายการคอนแทคลิสต์ของผู้ใช้งานไว้ที่เซิร์ฟเวอร์เพื่อหากผู้ใช้ทำการออนไลน์ยังเครื่องอื่นโดยใช้UIN เดิมที่เคยใช้ไว้แล้ว ก็จะได้ไม่ต้องทำการค้นหาและเพิ่มคู่สนทนาที่เคยอยู่ในคอนแทคลิสต์อีก

ต่อมาโปรแกรม ICQ ก็เพิ่มบริการเสริมเพื่อที่จะให้โปรแกรม ICQ เป็นโปรแกรมที่สามารถใช้งานได้หลากหลายมากยิ่งขึ้นเช่น การสร้างห้องสนทนาสำหรับสนทนาร่วมกันหลายคน การตรวจสอบการสะกดตัวอักษร (spell-checker) การส่งข้อความไปยังโทรศัพท์มือถือโดยผ่านทาง SMS รวมถึงใช้ในการติดต่อแบบพีซีไปยังโทรศัพท์เป็นต้น

ในอนาคตโปรแกรม ICQ น่าจะมีการเพิ่มความสามารถในการสนทนาประเภท Video chat สามารถที่จะสนทนาผ่านทางเว็บแคมได้ และในระยะหลังนี้โปรแกรม ICQ และโปรแกรมรับส่งสารด่วนอื่นๆ ได้เริ่มมีการพัฒนาเพื่อแก้ปัญหาด้านความปลอดภัย ทำให้เชื่อกันว่าในอนาคตจะมีองค์กรทางธุรกิจหันมาใช้โปรแกรมรับส่งสารด่วนติดต่อกันภายในมากขึ้น และแนวโน้มการใช้จดหมายอิเล็กทรอนิกส์และวอยส์เมล์ภายในองค์กรจะลดลง

ภาคผนวก ข.

การดักจับแพ็กเก็ตโดยใช้โปรแกรม Damping ICQ

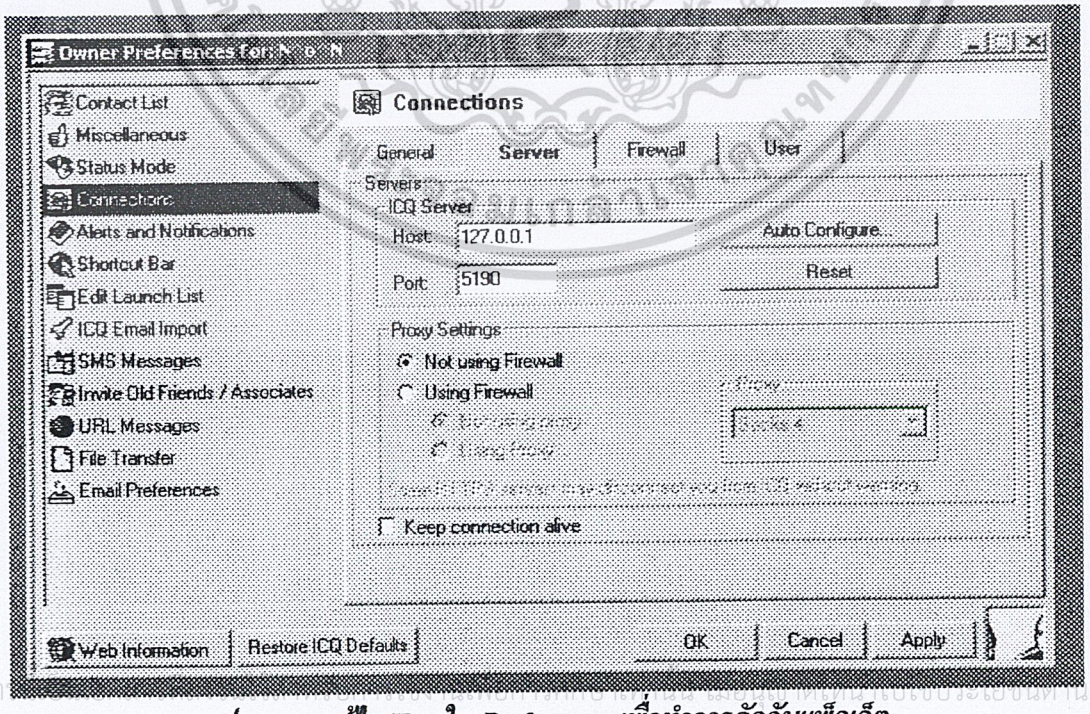
โปรแกรม Damping ICQ เป็นโปรแกรมที่ใช้สำหรับดักจับแพ็กเก็ตที่ใช้งานในโปรแกรม ICQ ซึ่งเป็นโปรแกรมที่สามารถใช้งานได้ง่าย โดยผู้ที่สนใจหรือต้องการนำมาใช้งานโปรแกรมนี้สามารถขอดาวน์โหลดได้จาก http://www.stricq.com/icqv8

ข.1 การติดตั้งโปรแกรม Damping ICQ

เมื่อทำการดาวน์โหลดโปรแกรมมาจากเว็บไซต์ในข้างต้นแล้ว โปรแกรม Damping ICQ จะอยู่ในรูปไฟล์ dampingicq.zip ให้ทำการ extract ไฟล์ออกมาไว้ในไดเรกทอรีที่ต้องการ ซึ่งจะทำให้ได้ไฟล์ dampingicq.exe สำหรับนำมาใช้งาน

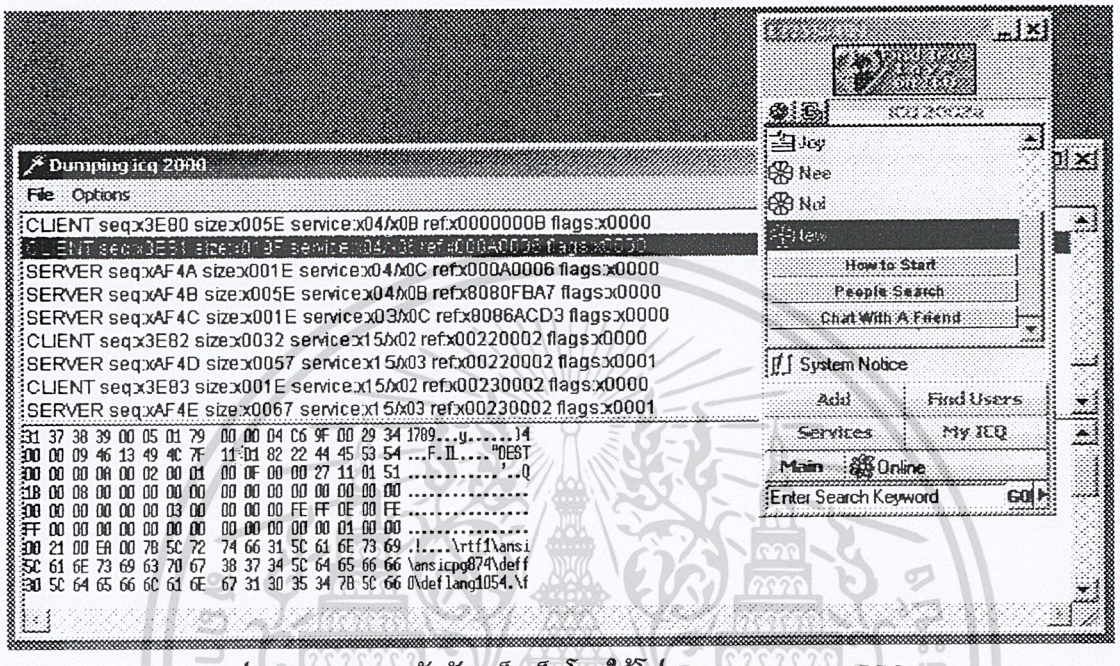
ข.2 การใช้งานโปรแกรม Damping ICQ

ก่อนจะเริ่มทำการดักจับแพ็กเก็ตของโปรแกรม Damping ICQ ผู้ใช้งานจะต้องทำการ จะต้องทำการแก้ไข Server สำหรับทำการติดต่อก่อนโดยเข้าไปยัง Main -> Preferences ซึ่งจะแสดงไดอะล็อกใหม่ขึ้นมา ให้ทำการเลือก Connections และเลือกแท็บ Server ดังรูปที่ ข.1 ให้ทำการแก้ไข Host ของ ICQ Server ใหม่เป็น localhost หรือ 127.0.0.1



รูป ข.1 การแก้ไข Host ใน Preferences เพื่อทำการดักจับแพ็กเก็ต

เมื่อทำการแก้ไข Host เรียบร้อยแล้วจึงทำการเปิดโปรแกรม Damping ICQ และทำการออนไลน์โปรแกรม ICQ ซึ่งโปรแกรม Damping ICQ จะทำการดักจับแพ็กเก็ตที่มีการรับส่งในโปรแกรม ICQ ของผู้ใช้งาน แล้วนำแพ็กเก็ตที่ได้นี้ไปตรวจสอบกับข้อมูลอ้างอิงในการทำงานของโปรแกรม ICQ



รูป ข.2 ผลจากการดักจับแพ็กเก็ตโดยใช้โปรแกรม Damping ICQ

ภาคผนวก ค.

การดักจับแพ็กเก็ตโดยใช้โปรแกรม Ethereal

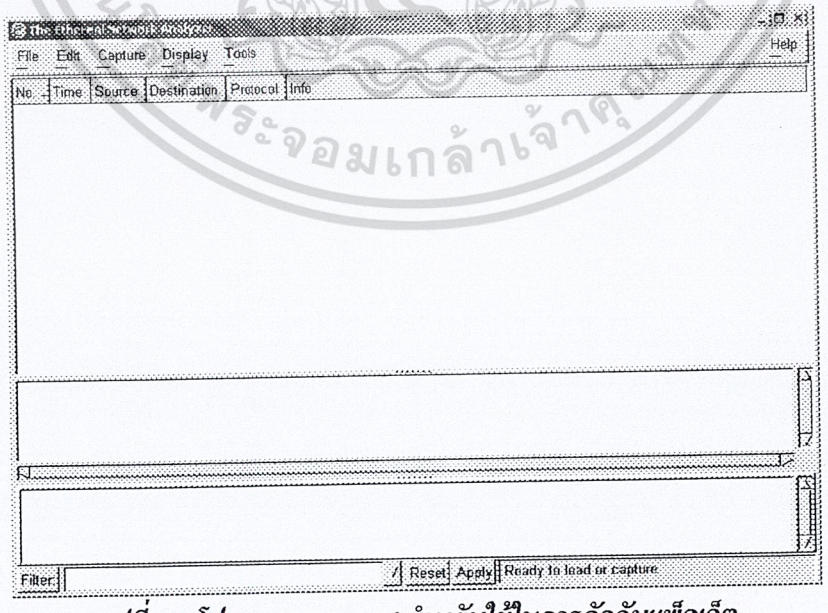
โปรแกรม Ethereal นั้นเป็นโปรแกรมที่ใช้สำหรับดักจับแพ็กเก็ตที่ใช้งานในโปรแกรม ICQ เช่นเดียวกับโปรแกรม Damping ICQ โดยผู้ที่สนใจหรือต้องการนำโปรแกรมนี้มาใช้งานสามารถดาวน์โหลดได้จาก http://www.ethereal.com/distribution/win32/ โดยให้เลือกดาวน์โหลดโปรแกรม Ethereal และโปรแกรม WinPcap

ค.1 การติดตั้งโปรแกรม Ethereal

เมื่อทำการดาวน์โหลดโปรแกรมมาจากเว็บไซต์ในข้างต้นแล้ว จะได้โปรแกรม Ethereal 0.9.7 ซึ่งอยู่ในรูปไฟล์ ethereal-setup-0.9.7.exe และโปรแกรม WinPcap 2.3 ซึ่งอยู่ในรูปไฟล์ WinPcap_2_3.exe ซึ่งในทั้ง 2 โปรแกรมนี้อาจมีการเปลี่ยนแปลงชื่อไฟล์ไปตามเวอร์ชัน จากนั้นให้ทำการติดตั้งโปรแกรมทั้ง 2 โปรแกรมไว้ในไดเรกทอรีเดียวกันโดยจะทำการติดตั้งโปรแกรมใดก่อนก็ได้

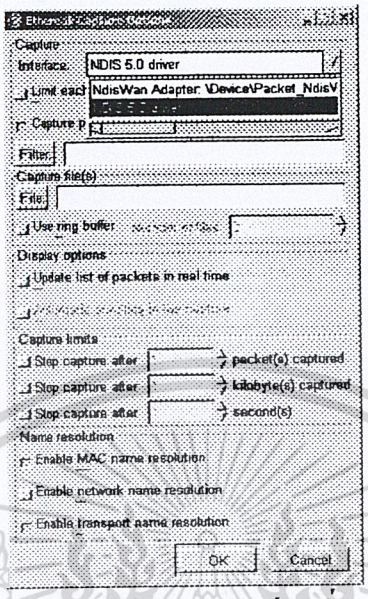
ค.2 การใช้งานโปรแกรม Ethereal

ในการดักจับแพ็กเก็ตโดยใช้โปรแกรม Ethereal ผู้ใช้งานไม่จำเป็นต้องทำการแก้ไขเซิร์ฟเวอร์เหมือนกับการใช้โปรแกรม Damping ICQ ดังนั้นเมื่อเราต้องการจะดักจับแพ็กเก็ตในส่วนใดให้เปิดโปรแกรม Ethereal ขึ้นมา



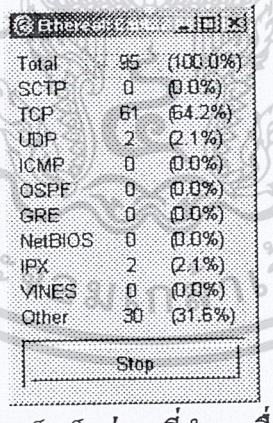
รูปที่ ค.1 โปรแกรม Ethereal สำหรับใช้ในการดักจับแพ็กเก็ต

จากนั้นเมื่อต้องการเริ่มดักจับให้ทำการเลือก Start โดยคลิกที่รายการ Capture -> Start.. ซึ่งจะแสดงหน้าจอดังรูปที่ ค.2 แล้วให้ทำการเลือก Interface ใน Capture ให้เป็นของ Lan Card



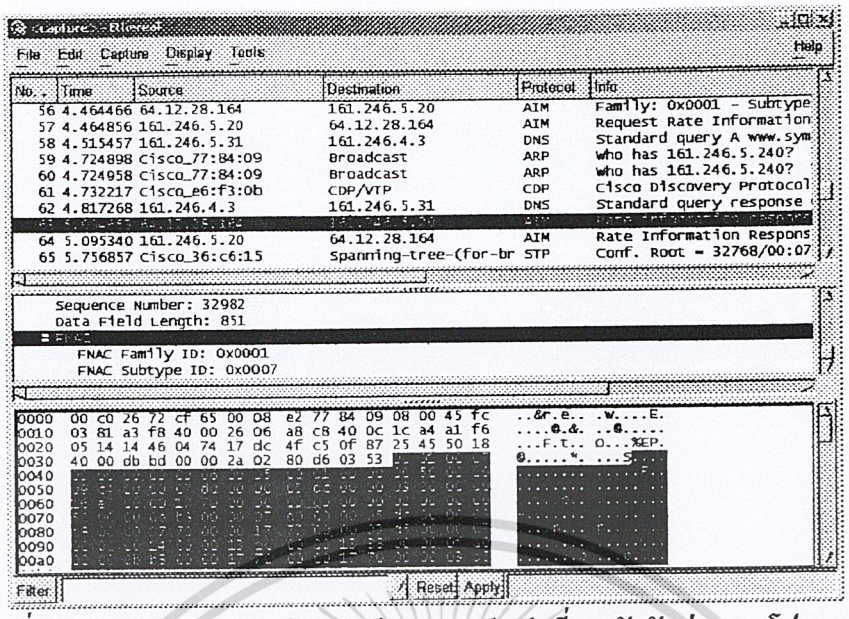
รูปที่ ค.2 การกำหนด Interface ใน Capture ก่อนเริ่มทำการดักจับแพ็กเก็ต

เมื่อกำหนด Interface เรียบร้อยแล้วให้ทำการคลิด OK เพื่อเริ่มทำการดักจับแพ็กเก็ตต่างๆ ที่ทำการรับส่งระหว่างเครื่องของผู้ใช้และสถานที่อื่น



รูป ค.3 แสดงการดักจับแพ็กเก็ตต่างๆ ที่ทำการสื่อสารกับเครื่องของผู้ใช้

เมื่อผู้ใช้ต้องการจะทำการหยุดหรือตรวจสอบรายละเอียดของแต่ละแพ็กเก็ตโดยละเอียด ให้ทำการคลิก Stop เพื่อหยุดการดักจับแพ็กเก็ต เมื่อหยุดการดักจับแพ็กเก็ตแล้ว โปรแกรมจะแสดงแพ็กเก็ตที่สื่อสารกับเครื่องผู้ใช้ ซึ่งผู้ใช้สามารถทำการคลิกเลือกที่แต่ละแพ็กเก็ตเพื่อดูรายละเอียดของแต่ละแพ็กเก็ตได้ โดยที่แพ็กเก็ตในการสื่อสารระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ผ่านทางโพรโตคอล AIM และแพ็กเก็ตในการสื่อสารระหว่างไคลเอ็นต์กับไคลเอ็นต์ผ่านทางโพรโตคอล TCP



รูปที่ ค.4 แสดงการตรวจสอบรายละเอียดของแพ็กเก็ตที่การดักจับผ่านทางโปรแกรม

บรรณานุกรม

หนังสืออ้างอิง

[1] Elliotte Rustly Harold : ***"Java Network Programming"***, O'Reilly & Associates, Inc. United State of America 2000.

[2] Justin Couch : ***"Java 2 Networking"***, McGraw-Hill Company. United State of America 1999.

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

[4] ***"Java™ Unleashed"***, Sams net. United State of America 1996.

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

[6] รุ่งโรจน์ โพนคำ : ***"Advanced Java Programming"***, บริษัท ซัคเซส มีเดีย จำกัด, 2000.

[7] ดร.วีระศักดิ์ ซึงถาวร : ***"JAVA Programming Volume 1"***, Se-Education Public Company Limited. 2000.

[8] ดร.วีระศักดิ์ ซึงถาวร : ***"JAVA Programming Volume 2"***, Se-Education Public Company Limited. 2002.

[9] สุวัฒน์ ปุณณชัยยะ : ***"เปิดโลก TCP/IP และโปรโตคอลของอินเตอร์เน็ต (Second Edition)"***, บริษัท โปรวิชั่น จำกัด, 2002.

เว็บไซต์อ้างอิง

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

[2] http://www.stricq.com/icqv8

[3] http://www.stud.uni-karlsruhe.de/~uck4/ICQ