สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรับส่งสารด่วนแบบปลอดภัย

(Secure Instant Messenging System)

นายนิธิพล ธรรมภูษิต

นางสาวนีรชา เทียนพิ่งเวียน

ปพ.
ท 613 ร
2543

เลขหมู่.....
เลขทะเบียน 42838
วัน, เดือน, ปี 10 มิ.ย. 2545

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b11917595

ระบบรับส่งสารด่วนแบบปลอดภัย

(Secure Instant Messenging System)

โดย

นายนิธิพล ธรรมภูษิต

นางสาวนีรชา เทียนพึ่งเวียน

อาจารย์ที่ปรึกษา

อ.ธนา หงษ์สุวรรณ

อ.อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรับส่งสารด่วนแบบปลอดภัย

Secure Instant Messenging System

ผู้จัดทำ

1. นายนิธิพล ธรรมภูษิต รหัสนักศึกษา 41013536
2. นางสาวนีรชา เทียนพึ่งเวียน รหัสนักศึกษา 40010401

.....อาจารย์ที่ปรึกษา

(อาจารย์ธนา หงษ์สุวรรณ)

.....อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรับส่งสารด่วนแบบปลอดภัย

นายนิธิพล ธรรมภูษิต 41013536
นางสาวนีรชา เทียนพึ่งเวียน 40010401
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ในปัจจุบันโปรแกรมที่ใช้สำหรับรับ-ส่งข้อความประเภท Instant Messenger เช่นโปรแกรม ICQ ได้รับความนิยมและใช้งานกันอย่างแพร่หลาย แต่ยังไม่เหมาะสำหรับการใช้งานในองค์กรที่ต้องการความปลอดภัยในการรับ-ส่งข้อความระหว่างกันเพราะข้อมูลสามารถถูกดักจับได้ระหว่างทาง จึงเกิดแนวคิดที่จะสร้างโปรแกรมรับ-ส่งข้อความที่มีความปลอดภัย โดยใช้ ICQ เป็นต้นแบบในการพัฒนาและเพิ่มส่วนของการเข้ารหัสข้อมูลที่รับ-ส่งเพื่อช่วยให้มีความปลอดภัยมากขึ้น และใช้ภาษาจาวาในการพัฒนาเพราะสามารถนำไปใช้ได้หลายแพลตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Secure Instant Messenging System

Mr.Nitipol Thammapusit	41013536
Ms.Neeracha Tianpungwian	40010401
Mr.Thana Hongsuwan	Advisor
Mr.Akkradach Watcharapupong	Advisor

ABSTRACT

Nowadays.the online instant messenger softwares,such as ICQ, are very popular.But it does not appropriate to use in organization who need security. Because message may be entraped during send. This project offers a secure instant messenger which use ICQ for prototype and increase encrypts/ decrypts portion. This project development by java language because it can run on many platforms.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความช่วยเหลือ และได้รับความร่วมมือจากบุคคลหลาย ๆ ท่านด้วยกัน ซึ่งบุคคลแรกที่ขอกล่าวคือต้องขอขอบพระคุณอาจารย์ที่ปรึกษาคืออาจารย์ ธนา หงษ์สุวรรณ และอาจารย์ อัครเดช วัชระภูพงษ์ ที่คอยดูแลเอาใจใส่ให้คำปรึกษา คำแนะนำ และให้การสนับสนุนทางด้านซอฟต์แวร์และฮาร์ดแวร์

ขอขอบพระคุณห้องวิจัยไอแซคที่ช่วยอำนวยความสะดวกต่าง ๆ รวมทั้งเอื้อเฟื้อสถานที่และอุปกรณ์ในการทำโครงงาน

และท้ายที่สุดต้องขอขอบพระคุณบิดามารดาเป็นอย่างสูงที่ให้การเลี้ยงดู อบรม ให้โอกาสทางการศึกษา ให้กำลังใจ และพระคุณอื่น ๆ อีกมากมายซึ่งยากที่จะทดแทนได้หมด จึงขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

นิธิพล ธรรมภูษิต

นีรชา เทียนพึ่งเวียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของการพัฒนา	1
1.3 ขอบเขตของการพัฒนา	1
1.4 รายละเอียดของการพัฒนา	2
บทที่ 2 ภาษาจาวา	3
2.1 ประวัติของภาษาจาวา	3
2.2 คุณสมบัติเด่นของภาษาจาวา	3
2.3 หลักการทำงานของภาษาจาวา	4
2.3.1 จาวาเวอร์ชวลแมชีน	4
2.3.2 การคอมไพล์	5
2.3.3 การรันโปรแกรม	5
2.3.3.1 การอ่านไบต์โค้ด	6
2.3.3.2 การตรวจสอบไบต์โค้ด	6
2.3.3.3 การทำงานตามโค้ด	6
2.4 การสร้างและรันโปรแกรมภาษาจาวา	7
บทที่ 3 การเข้ารหัสข้อมูล	9
3.1 การเข้ารหัสข้อมูล	9
3.1.1 ระบบการเข้ารหัสแบบคีย์เดี่ยว	9
3.1.2 ระบบการเข้ารหัสแบบคีย์คู่	9
3.2 การเข้ารหัสแบบ DES	10
3.2.1 รายละเอียดของ DES	10
3.2.2 การทำ Initial Permutation และ Inverse Permutation	11
3.2.3 รายละเอียดของการทำฟังก์ชัน Permutation และ Substitution ในแต่ละรอบ	12
3.2.4 การสร้างคีย์	18
3.2.5 การถอดรหัสข้อมูล DES	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 โหมด CBC (Cipher Block Chaining) 21
3.3 การเข้ารหัสแบบ RSA 22
3.3.1 รายละเอียดของ RSA 23
3.4 การนำการเข้ารหัสแบบ DES และแบบ RSA ไปใช้งาน 24
บทที่ 4 โพรโตคอล ICQ 25
4.1 เวอร์ชันของโพรโตคอล 25
4.2 เซิร์ฟเวอร์ ICQ 25
4.3 การสื่อสารระหว่างไคลเอนต์กับเซิร์ฟเวอร์ 26
4.3.1 แพ็กเก็ตที่ใช้สื่อสารกับเซิร์ฟเวอร์ 26
4.3.2 ข้อกำหนดในการสื่อสาร 27
4.3.3 การล็อกอิน 27
4.3.4 การส่งข้อความ 29
4.3.5 การค้นหาผู้ใช้คนอื่น 30
4.3.6 การเพิ่มชื่อผู้ใช้อื่นเข้าสู่คอนแทคลิสต์ 31
4.3.7 การขอข้อมูลผู้ใช้จากเซิร์ฟเวอร์ 31
4.3.8 การขออนุญาตเพิ่มชื่อผู้ใช้อื่นเข้าคอนแทคลิสต์ 32
4.3.9 การเพิ่มรายชื่อผู้ใช้งาน 32
4.3.10 การเปลี่ยนผู้ใช้งาน 34
4.3.11 การแก้ไขข้อมูลส่วนตัว 35
4.3.12 การส่ง KeepAlive 35
4.4 การสื่อสารระหว่างไคลเอนต์กับไคลเอนต์ 36
4.4.1 รูปแบบของแพ็กเก็ตในการสื่อสารกันโดยตรง 36
4.4.2 การรับส่งข้อความโดยตรง 42
4.4.3 การรับส่งไฟล์ 43
บทที่ 5 การพัฒนาโครงงาน 46
5.1 การพัฒนาส่วนติดต่อกับผู้ใช้งาน 46
5.1.1 ขั้นตอนในการพัฒนาส่วนติดต่อกับผู้ใช้งาน 47
5.1.1.1 การเลือกคอมโพเนนท์ที่จะนำมาใช้จัดการกับข้อมูลแต่ละส่วนตามความเหมาะสม 48
5.1.1.2 การออกแบบลักษณะการจัดเรียงคอมโพเนนท์และนำมาประกอบกัน 48
5.1.1.3 เขียนโปรแกรมจัดการในส่วนที่ต้องมีการตอบสนองต่อผู้ใช้ 50
5.1.1.4 นำโปรแกรมที่พัฒนาส่วนติดต่อกับผู้ใช้งานแล้วไปเชื่อมต่อกับส่วนติดต่อกับเน็ตเวิร์ก 51
5.2 การพัฒนาส่วนติดต่อกับเน็ตเวิร์ก 51
5.2.1 OwnerUserInfo 52
5.2.2 LoginSeq 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 SendToServer 53
5.2.4 ServerListener 54
5.2.5 ServerHandle 54
5.2.6 ContactUserInfo 55
5.2.7 SendFileSeq 56
5.2.8 RreceiveFileSeq 56
5.2.9 ClientListener 56
5.2.10 ClientHandle 56
5.3 การพัฒนาส่วนของการเข้ารหัส 56
บทที่ 6 การทดสอบโครงงาน 58
6.1 การทดสอบรับส่งข้อความแบบปรกติ 58
6.2 การทดสอบรับส่งข้อความแบบเข้ารหัส 59
6.3 การทดสอบรับส่งไฟล์แบบปรกติ 61
6.4 การทดสอบรับส่งไฟล์แบบเข้ารหัส 65
บทที่ 7 สรุปผลการพัฒนาโครงงาน 66
7.1 สรุปผลการดำเนินงาน 66
7.2 คุณสมบัติของโปรแกรมที่ได้จากการพัฒนา 66
7.3 ข้อจำกัดของโปรแกรม 67
7.4 แนวทางในการพัฒนาต่อ 67
บรรณานุกรม 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้าที่
ตารางที่ 3-1	แสดงการจัดเรียงของข้อมูลก่อนเข้าสู่ฟังก์ชัน Permutation	12
ตารางที่ 3-2	แสดงการจัดเรียงของข้อมูลหลังจากทำตามฟังก์ชัน Permutation	12
ตารางที่ 3-3	แสดง Initial Permutation (IP) ของ DES	14
ตารางที่ 3-4	แสดง Inverse Initial Permutation (IP^{-1}) ของ DES	14
ตารางที่ 3-5	แสดง Expansion Permutation (E) ของ DES	14
ตารางที่ 3-6	แสดง Permutation Function (P) ของ DES	15
ตารางที่ 3-7	แสดง S-Box ของ DES	17
ตารางที่ 3-8	แสดงการคำนวณของ Permuted Choice One (PC-1)	17
ตารางที่ 3-9	แสดงการคำนวณของ Permuted Choice One (PC-2)	19
ตารางที่ 3-10	แสดงจำนวนบิตที่ทำ Circular Left Shift	19
ตารางที่ 4-1	แสดงรูปแบบแพ็กเก็ตที่ส่งจากไคลเอนต์ไปเซิร์ฟเวอร์	26
ตารางที่ 4-2	แสดงรูปแบบแพ็กเก็ตที่ส่งจากเซิร์ฟเวอร์ไปไคลเอนต์	27
ตารางที่ 4-3	แสดงข้อมูลภายในแพ็กเก็ต CHANNEL_INIT	36
ตารางที่ 4-4	แสดงข้อมูลภายในแพ็กเก็ต CHANNEL_MESSAGE	37
ตารางที่ 4-5	แสดงข้อมูลเพิ่มเติมของแพ็กเก็ต CHANNEL_MESSAGE สำหรับการขอส่งไฟล์	37
ตารางที่ 4-6	แสดงข้อมูลภายในแพ็กเก็ต ACK	38
ตารางที่ 4-7	แสดงข้อมูลเพิ่มเติมของแพ็กเก็ต ACKสำหรับการตอบรับการขอส่งไฟล์	38
ตารางที่ 4-8	แสดงตัวแปรสำหรับฟิลด์ COMMAND ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK	39
ตารางที่ 4-9	แสดงตัวแปรสำหรับฟิลด์ TYPE ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK	39
ตารางที่ 4-10	แสดงตัวแปรสำหรับฟิลด์ TCP_STATUS ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK	39
ตารางที่ 4-11	แสดงตัวแปรสำหรับฟิลด์ MSG_COMMAND ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK	40
ตารางที่ 4-12	แสดงแพ็กเก็ต 0x00	40
ตารางที่ 4-13	แสดงแพ็กเก็ต 0x01	40
ตารางที่ 4-14	แสดงแพ็กเก็ต 0x02	41
ตารางที่ 4-15	แสดงแพ็กเก็ต 0x03	41
ตารางที่ 4-16	แสดงแพ็กเก็ต 0x04	41
ตารางที่ 4-17	แสดงแพ็กเก็ต 0x05	41
ตารางที่ 4-18	แสดงแพ็กเก็ต 0x06	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

		หน้าที่
รูปที่ 2-1	แสดงหลักการทำงานของภาษาจาวาโดยรวม	4
รูปที่ 2-2	แสดงการประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน	5
รูปที่ 2-3	แสดงการประมวลผลของโปรแกรมภาษาจาวา	6
รูปที่ 3-1	แสดงการเข้ารหัสและถอดรหัสแบบคีย์เดี่ยว	9
รูปที่ 3-2	แสดงการเข้ารหัสและถอดรหัสแบบคีย์คู่	10
รูปที่ 3-3	แสดงขั้นตอนการทำงานของ DES	11
รูปที่ 3-4	แสดงการเข้ารหัส DES ในแต่ละครั้ง (ทำทั้งหมด 16 ครั้ง)	13
รูปที่ 3-5	แสดงการทำ Permutation สำหรับ row 0 ของ S_1	15
รูปที่ 3-6	แสดงขั้นตอนการทำงานของ DES โดยรวมแบบฟังก์ชัน	17
รูปที่ 3-7	แสดงคีย์และขั้นตอนการเข้ารหัสและถอดรหัส DES	20
รูปที่ 3-8	แสดงการเข้ารหัสของ DES CBC	21
รูปที่ 3-9	แสดงการถอดรหัสของ DES CBC	21
รูปที่ 3-10	แสดงขั้นตอนการเข้ารหัสและถอดรหัสแบบ RSA	22
รูปที่ 3-11	แสดงขั้นตอนการเข้ารหัสแบบ RSA	23
รูปที่ 4-1	แสดงขั้นตอนในการล็อกอิน	28
รูปที่ 4-2	แสดงกรณีที่พาสเวิร์ดผิด	29
รูปที่ 4-3	แสดงขั้นตอนการส่งข้อความผ่านเซิร์ฟเวอร์	29
รูปที่ 4-4	แสดงขั้นตอนการค้นหาผู้ใช้คนอื่น	30
รูปที่ 4-5	แสดงขั้นตอนการเพิ่มผู้ใช้อื่นเข้าคอนแทคลิสต์	31
รูปที่ 4-6	แสดงขั้นตอนการขอข้อมูลเกี่ยวกับผู้ใช้จากเซิร์ฟเวอร์	32
รูปที่ 4-7	แสดงขั้นตอนการขอ Authorize	33
รูปที่ 4-8	แสดงขั้นตอนการเพิ่มผู้ใช้งาน	33
รูปที่ 4-9	แสดงขั้นตอนการเปลี่ยนผู้ใช้งาน	34
รูปที่ 4-10	แสดงขั้นตอนการแก้ไขข้อมูลส่วนตัว	35
รูปที่ 4-11	แสดงขั้นตอนการส่ง Keep Alive	36
รูปที่ 4-12	แสดงขั้นตอนการส่งข้อความโดยตรงระหว่างไคลเอนต์	43
รูปที่ 4-13	แสดงขั้นตอนการรับส่งไฟล์	44
รูปที่ 5-1	แสดงรูปของโปรแกรมที่ออกแบบก่อนการพัฒนา	47
รูปที่ 5-2	แสดงการจัดเลย์เอาต์ในส่วนหน้าต่างหลัก	49
รูปที่ 5-3	แสดงการจัดเลย์เอาต์ในส่วนคอนแทคลิสต์	49
รูปที่ 5-4	แสดงการจัดเลย์เอาต์ในส่วนปุ่มด้านล่างของหน้าต่างหลัก	50
รูปที่ 5-5	แสดงการจัดเลย์เอาต์ในส่วนแสดงรายชื่อผู้ใช้แต่ละกลุ่ม	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5-6	แสดงการจัดเลย์เอาต์ของป๊อพอัพเมนูของปุ่ม IsagQ	50
รูปที่ 5-7	แสดงโครงสร้างหลักของโปรแกรมในส่วนติดต่อกับเน็ตเวิร์ก	51
รูปที่ 5-8	แสดงโครงสร้างของคลาส LoginSeq	52
รูปที่ 5-9	แสดงการติดต่อระหว่างคลาส LoginSeq กับคลาสอื่น	53
รูปที่ 5-10	แสดง Flow Chart ของคลาส SendToServer	53
รูปที่ 5-11	แสดงการติดต่อระหว่างคลาส SendToServer และ TimeCount	54
รูปที่ 5-12	แสดงการใช้คลาส ServerAck ร่วมกับคลาส ServerListener	54
รูปที่ 5-13	แสดงการจัดการคิวของคลาส ServerListener	55
รูปที่ 5-14	แสดงการติดต่อระหว่างคลาส ContactUserInfo กับคลาสอื่นๆ	55
รูปที่ 5-15	แสดงการแลกเปลี่ยนคีย์ระหว่างไคลเอนต์	57
รูปที่ 6-1	แสดงหน้าต่างส่งข้อความแบบปรกติ	58
รูปที่ 6-2	แสดงหน้าต่างที่รับข้อความแบบปรกติ	58
รูปที่ 6-3	แสดงหน้าต่างแสดงการดักจับข้อความที่ส่งแบบปรกติ	59
รูปที่ 6-4	แสดงหน้าต่างส่งข้อความแบบเข้ารหัส	60
รูปที่ 6-5	แสดงหน้าต่างที่รับข้อความแบบเข้ารหัส	60
รูปที่ 6-6	แสดงหน้าต่างแสดงการดักจับข้อความที่เข้ารหัส	60
รูปที่ 6-7	แสดงขนาดของไฟล์ต้นทาง	61
รูปที่ 6-8	แสดงไฟล์ Doc1.doc ต้นทาง	62
รูปที่ 6-9	แสดงไฟล์ Doc2.doc ต้นทาง	62
รูปที่ 6-10	แสดงไฟล์ Pic1.jpg ต้นทาง	62
รูปที่ 6-11	แสดงไฟล์ Pic2.jpg ต้นทาง	63
รูปที่ 6-12	แสดงขนาดของไฟล์ปลายทาง	63
รูปที่ 6-13	แสดงไฟล์ Doc1.doc ปลายทาง	63
รูปที่ 6-14	แสดงไฟล์ Doc2.doc ปลายทาง	64
รูปที่ 6-15	แสดงไฟล์ Pic1.jpg ปลายทาง	64
รูปที่ 6-16	แสดงไฟล์ Pic2.jpg ปลายทาง	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การติดต่อสื่อสารผ่านระบบเครือข่ายช่วยเพิ่มความสะดวกและประหยัดทั้งเวลาและค่าใช้จ่ายในการเดินทาง และเมื่อมีการพัฒนาโปรแกรมรับส่งข้อมูลแบบด่วน (Instant Message) ขึ้นมา เช่น โปรแกรมไอซีคิว (ICQ) ทำให้สามารถรับส่งข้อมูลและโต้ตอบกันได้ทันที การรับส่งข้อมูลแบบนี้ไม่ค่อยมีความปลอดภัยเพราะถ้าหากมีผู้ไม่ประสงค์ดีมาดักจับข้อมูลระหว่างการรับส่งก็อาจทำให้ข้อมูลที่เป็นความลับรั่วไหลได้เพราะข้อมูลเหล่านั้นสามารถอ่านได้โดยตรง จึงได้เกิดแนวความคิดที่จะสร้างโปรแกรมรับส่งข้อความที่มีการเข้ารหัสข้อมูลก่อนการส่งเพื่อเพิ่มความปลอดภัยในการสื่อสาร ซึ่งก็คือโครงงานโปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา (Secure Instant Messenger Using Java) หรือ IsagQ แต่โปรแกรมนี้ยังไม่สมบูรณ์เนื่องจากขาดฟังก์ชันที่จำเป็นหลายอย่างที่โปรแกรมไอซีคิวมีและไม่มีเสถียรภาพในการทำงานเท่าที่ควร จึงทำการพัฒนาต่อให้สมบูรณ์และสามารถใช้งานได้จริงโดยมีชื่อโครงงานว่าโครงงานระบบรับส่งสารด่วนแบบปลอดภัย (Secure Instant Messenging System)

1.2 วัตถุประสงค์ของการพัฒนา

1.2.1 เพื่อสร้างโปรแกรมรับส่งข้อความและไฟล์ที่สามารถใช้สื่อสารกับโปรแกรม ICQ ได้แต่มีความปลอดภัยมากกว่าโดยใช้การเข้าและถอดรหัสข้อมูลระหว่างการสื่อสาร และเพื่อให้โปรแกรมสามารถทำงานได้หลายแพลตฟอร์มจึงเลือกใช้ภาษาจาวาในการพัฒนา รวมถึงต้องมีเสถียรภาพในการใช้งานมากกว่าโครงงานเดิม

1.2.2 เพื่อสร้างโปรแกรมที่มีฟังก์ชันการทำงานที่สำคัญใกล้เคียงกับโปรแกรม ICQ มากขึ้น เช่น สามารถเพิ่มหรือเปลี่ยนผู้ใช้งาน, รับส่งไฟล์ได้ เป็นต้น

1.2.3 เพื่อนำเสนอความรู้พื้นฐานเกี่ยวกับโพรโตคอลของ ICQ และการเข้า-ถอดรหัสข้อมูลสำหรับนำไปใช้พัฒนาต่อไป

1.3 ขอบเขตของการพัฒนา

1.3.1 มีการล็อกอิน (login) ก่อนการใช้งานโปรแกรมโดยสามารถเลือกชื่อผู้ใช้ที่จะล็อกอินได้

1.3.2 สามารถเพิ่มหรือลบรายชื่อผู้ใช้งานได้

1.3.3 ขณะที่ใช้งานสามารถเปลี่ยนเป็นผู้ใช้งานคนอื่นได้

1.3.4 โปรแกรมสามารถใช้สื่อสาร (รับส่งข้อความและไฟล์) ระหว่าง IsagQ ด้วยกันและสามารถสื่อสารกับ ICQ ได้ ในกรณีที่ทั้งสองฝ่ายใช้โปรแกรม IsagQ และเชื่อมต่อการสื่อสารกันแบบ TCP จะมีการเข้ารหัสข้อมูลระหว่างการสื่อสารเพื่อเพิ่มความปลอดภัย

1.3.5 สามารถแสดงสถานะของผู้ใช้งานได้ เช่น ออนไลน์, ออฟไลน์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.6 สามารถค้นหาผู้ใช้คนอื่นจากหมายเลข UIN, อีเมล์ (e-mail) หรือชื่อได้ และสามารถเพิ่มเข้าสู่คอนแทคลิสต์ (contact list) ได้

1.3.7 สามารถใช้งานภาษาไทยได้

1.3.8 สามารถใช้งานได้หลายแพลตฟอร์ม

1.3.9 ส่วนติดต่อกับผู้ใช้งานเป็นแบบกราฟฟิก โดยรับอินพุตทางคีย์บอร์ดและเมาส์ และแสดงเอาต์พุตทางจอมอนิเตอร์

1.4 รายละเอียดของการพัฒนา

การพัฒนาโปรแกรม IsagQ เลือกใช้ภาษาจาวาเนื่องจากภาษาจาวาสามารถพัฒนาโปรแกรมที่สามารถนำไปใช้งานได้บนแพลตฟอร์มใดๆ ที่มีจาวาเวอร์ชวลแมชีน (Java Virtual Machine) อยู่ และสามารถเขียนโปรแกรมแบบเชิงวัตถุ (Object-Oriented Programming) ได้ง่าย ซึ่งจะกล่าวถึงรายละเอียดในบทที่ 2

นอกจากนี้ยังมีทฤษฎีพื้นฐานที่ต้องนำมาประกอบการพัฒนาโปรแกรมอีก 2 ส่วนคือโพรโตคอลของ ICQ และระบบการเข้าและถอดรหัส โพรโตคอลของ ICQ มีหลายเวอร์ชันตั้งแต่เวอร์ชัน 1 ถึง 5 แต่ปัจจุบันเซิร์ฟเวอร์ของ ICQ รองรับเฉพาะเวอร์ชัน 2 ถึง 5 เท่านั้น และในการพัฒนาโครงงานนี้จะเลือกใช้เวอร์ชัน 2 เนื่องจากทำความเข้าใจและนำมาใช้ได้ง่ายกว่า สำหรับส่วนของการเข้า-ถอดรหัสข้อมูลจะใช้แบบ DES (แบบคีย์เดี่ยว) และแบบ RSA (แบบคีย์คู่) ร่วมกัน โดยนำการเข้ารหัสแบบ DES มาใช้กับข้อมูลที่รับส่งเนื่องจากมีความเร็วมากกว่า และนำการเข้ารหัสแบบ RSA มาใช้ในขั้นตอนการแลกเปลี่ยนคีย์ DES ซึ่งทั้งสองส่วนนี้จะอธิบายอยู่ในบทที่ 3 และบทที่ 4 ตามลำดับ และในการพัฒนาโปรแกรมได้มีการออกแบบโครงสร้างของโปรแกรมและส่วนติดต่อกับผู้ใช้งานใหม่ซึ่งมีการนำโปรแกรมบางส่วนของโครงงานเดิมมาใช้ ซึ่งจะขอกล่าวถึงรายละเอียดในบทที่ 5 สำหรับบทที่ 6 จะกล่าวถึงผลสรุปจากการพัฒนาโครงงานและเสนอแนวทางพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ภาษาจาวา

2.1 ประวัติของภาษาจาวา

ภาษาจาวาพัฒนาขึ้นโดยบริษัทซันไมโครซิสเต็มส์ (Sun Microsystems) เพื่อใช้กับระบบปฏิบัติการเซ็ตท็อปบ๊อกซ์ (set-top box) ซึ่งเป็นโครงการของระบบตลาดอิเล็กทรอนิกส์ที่ต้องการให้ระบบสามารถควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้โดยไม่ขึ้นอยู่กับแพลตฟอร์มที่ใช้ก็คือการพัฒนาระบบปฏิบัติการหรือซอฟต์แวร์ให้สามารถทำงานได้โดยไม่ต้องสนใจว่าไมโครโพรเซสเซอร์ที่ใช้อยู่จะเป็นรุ่นใด ยี่ห้อไหน ฮาร์ดแวร์เป็นของอะไร นอกจากนี้ยังรวมไปถึงความสามารถในการเชื่อมต่อการแลกเปลี่ยนข้อมูลข่าวสารผ่านทางระบบเครือข่ายคอมพิวเตอร์ไว้อีกด้วย

แต่ภาษาจาวาเริ่มเป็นที่สนใจอย่างแพร่หลายเมื่อบริษัทซันไมโครซิสเต็มส์ประกาศให้ภาษาจาวาเป็นภาษาสำหรับใช้สร้างโปรแกรมที่จะใช้งานบนอินเทอร์เน็ตในสภาวะแวดล้อมแบบกระจาย (distibuted environment) โดยเน้นเกี่ยวกับการเป็นภาษาที่ไม่ขึ้นกับแพลตฟอร์ม, ใช้หลักการออกแบบภาษาด้วยวิธีเชิงวัตถุ และการสนับสนุนการพัฒนาโปรแกรมด้วยแนวคิดเชิงวัตถุ

2.2 คุณสมบัติเด่นของภาษาจาวา

2.2.1 การทำงานที่ไม่ขึ้นกับแพลตฟอร์ม เนื่องจากใช้การคอมไพล์ (compile) ซอร์สโค้ด (sourcecode) เป็นไบต์โค้ด (bytecode) ซึ่งไบต์โค้ดนี้สามารถนำไปรันที่ใด ๆ ในระบบเครือข่าย (ไม่ว่าจะเป็นไคลเอนต์หรือเซิร์ฟเวอร์) ที่มีจาวาเวอร์ชวลแมชีนอยู่ โดยจาวาเวอร์ชวลแมชีนจะทำหน้าที่แปลไบต์โค้ดเป็นเนทีฟโค้ด (native code) ซึ่งเป็นโค้ดที่สามารถรันบนคอมพิวเตอร์ฮาร์ดแวร์จริงๆ (แต่ละแพลตฟอร์มจะมีความแตกต่างกันในหลายเรื่อง เช่น ความยาวของคำสั่ง, การวางตำแหน่งในหน่วยความจำขณะทำงาน) ดังนั้นจึงสามารถนำไปใช้งานในสภาวะแวดล้อมที่แตกต่างได้โดยไม่ต้องมีการปรับแต่ง (portability)

2.2.2 ความคงทน (robust) คือ สามารถทำงานต่อเนื่องได้เป็นเวลานานโดยไม่พังง่าย ๆ ด้วยความผิดปกติเพียงเล็กน้อย เนื่องจากออบเจ็กต์ของภาษาจาวาสามารถเก็บข้อมูลจากภายนอกได้โดยไม่ต้องอ้างอิงแอดเดรสของตัวแปรและไม่มีการใช้พอยน์เตอร์ในการอ่านเขียนค่าหน่วยความจำโดยตรง จึงหมดปัญหาเกี่ยวกับการขอใช้หน่วยความจำที่โปรแกรมไม่มีสิทธิ์ซึ่งอาจเป็นสาเหตุให้โปรแกรมหรือระบบปฏิบัติการหยุดการทำงาน นอกจากนี้ภาษาจาวายังเน้นการใช้กลไก exception handling ที่ช่วยให้โปรแกรมสามารถจัดการกับความผิดปกติบางอย่างที่อาจเกิดขึ้นในขณะโปรแกรมทำงานได้

2.2.3 สนับสนุนหลักการของแนวคิดเชิงวัตถุในการสร้างโปรแกรม เช่น การสืบทอด (inheritance), การใช้เมธอด (method) จัดการกับข้อมูลแทนฟังก์ชัน เป็นต้น

2.2.4 การรองรับมาตรฐานความปลอดภัยในการใช้งานในรูปแบบต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 เรียนรู้ได้ง่ายเมื่อเทียบกับภาษาซีพลัสพลัส (C++) แม้จะนำไวยากรณ์มาจากภาษาซีพลัสพลัส แต่ก็ตัดกลไกที่ซับซ้อนออกไปหลายอย่าง โดยไปเพิ่มความสามารถให้คอมไพล์เลอร์ (compiler) จัดการแทน และมีการออกแบบให้เป็นภาษาเชิงวัตถุที่รอบคอบกว่า

2.2.6 ความสามารถในการประมวลผลแบบกระจาย (distributed)

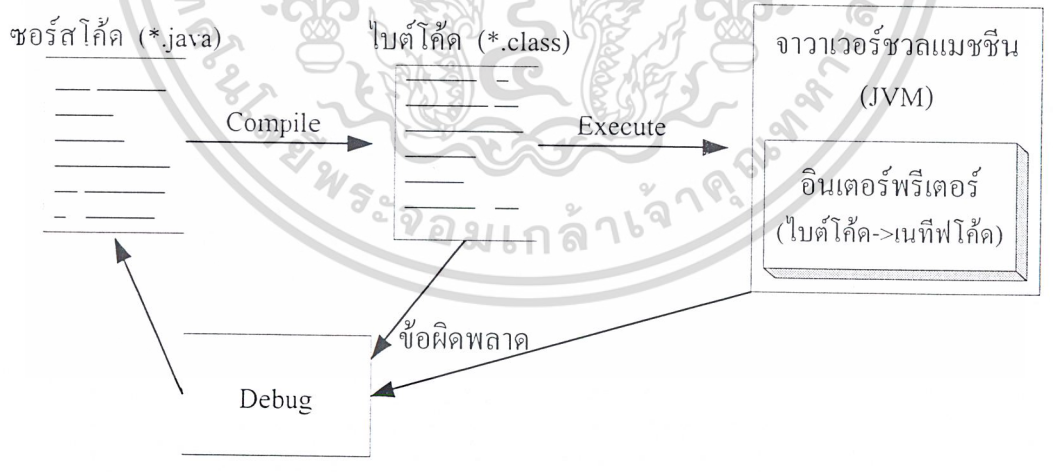
2.2.7 เป็นระบบมัลติเธรด คือสามารถทำงานได้หลายงานพร้อม ๆ กัน

2.2.8 ความสามารถในการดึงเอาคลาสไลบรารี่จากที่ต่าง ๆ ผ่านทางเอชทีทีพี (HTTP) และเอฟทีพี (FTP) โดยผู้ที่รับโปรแกรมนั้นไปใช้งานจะไม่พบปัญหาที่ส่งผลถึงเครื่องหรือระบบข้อมูลของผู้ใช้

2.2.9 ความสามารถในการโต้ตอบแบบอินเตอร์แอกทีฟและแบบเรียลไทม์ รวมถึงสามารถเล่นไฟล์เสียง วิดีโอและกราฟิกต่าง ๆ

2.3 หลักการทำงานของภาษาจาวา

ก่อนที่จะอธิบายถึงรายละเอียดของแต่ละขั้นตอนของการทำงานของภาษาจาวาจะขออธิบายถึงจาวาเวอร์ชวลแมชีนเสียก่อน และสำหรับภาพรวมของหลักการทำงานของภาษาจาวาแสดงดังรูปที่ 2-1 ซึ่งจะเห็นได้ว่ามีองค์ประกอบหลัก 4 ส่วนคือ ขั้นแรกต้องมีซอร์สโค้ดภาษาจาวา (ไฟล์ที่มีนามสกุลเป็น .java) เสียก่อน จากนั้นต้องแปลงซอร์สโค้ดนั้นเป็นไบต์โค้ด (ไฟล์ที่มีนามสกุลเป็น .class) โดยใช้คอมไพล์เลอร์ของภาษาจาวา ซึ่งถ้าหากมีข้อผิดพลาดก็ต้องดีบัก (debug) แก้ไขซอร์สโค้ดใหม่จนไม่มีข้อผิดพลาดจึงสามารถนำไปใช้งานได้ สำหรับการนำโปรแกรมไปใช้งานจะทำโดยตัวอินเทอร์พรีเตอร์ที่อยู่ภายในจาวาเวอร์ชวลแมชีนซึ่งจะประมวลผลไบต์โค้ดนั้นเป็นคำสั่งในรูปแบบของหน่วยประมวลผลในคอมพิวเตอร์เครื่องนั้นเพื่อให้หน่วยประมวลผลทำงานต่อไป



รูปที่ 2-1 แสดงหลักการทำงานของภาษาจาวาโดยรวม

2.3.1 จาวาเวอร์ชวลแมชีน

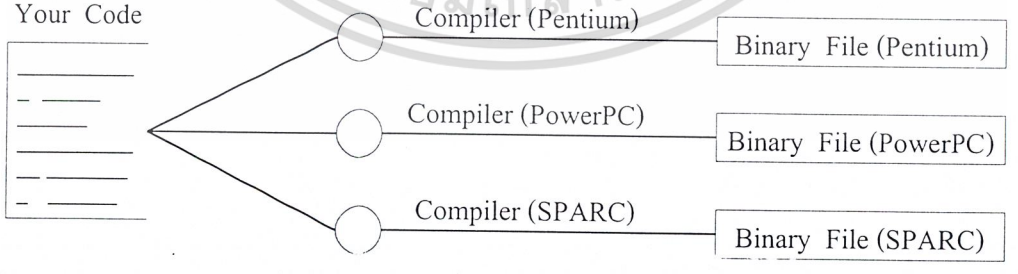
ปัจจุบันบริษัทจาวาซอฟต์ (JavaSoft) ซึ่งเป็นส่วนหนึ่งของบริษัทซันไมโครซิสเต็มส์ได้กำหนดมาตรฐานของภาษาจาวาและเป็นผู้กำหนดชุดคำสั่งของจาวาเวอร์ชวลแมชีน รวมทั้งความหมายของแต่ละคำสั่ง ซึ่งชุดคำสั่งของจาวาเวอร์ชวลแมชีนถูกออกแบบมาเพื่อสนับสนุนการทำงานของโปรแกรมเชิงวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงมีคำสั่งเกี่ยวกับการสร้างอินสแตนซ์ (instance) และการอ้างถึงสมาชิกในอินสแตนซ์ซึ่งไม่มีในหน่วยประมวลผลทั่วไป รวมถึงคำสั่งสำหรับคำนวณชนิดข้อมูลพื้นฐานแต่ละชนิด และจากมาตรฐานของภาษาจาวานี้บุคคลใด ๆ สามารถนำไปสร้างจาวาเวอร์ชวลแมชีนของตนเองขึ้นมาได้ โดยอาจจะเป็นฮาร์ดแวร์หรือซอฟต์แวร์ก็ได้ (ส่วนใหญ่ที่ใช้ในปัจจุบันเป็นซอฟต์แวร์จำลองการทำงานของจาวาเวอร์ชวลแมชีนบนเครื่องคอมพิวเตอร์ทั่วไป) เพียงแต่ต้องสามารถแปลไบต์โค้ดให้ได้เนทีฟโค้ดตามมาตรฐานที่กำหนดไว้ ภายในจาวาเวอร์ชวลแมชีนจะมีหน่วยประมวลผลที่เรียกว่าเวอร์ชวลโพรเซสเซอร์ (virtual processor) ซึ่งก็คืออินเทอร์พรีเตอร์ที่ทำหน้าที่แปลไบต์โค้ดให้เป็นคำสั่งของหน่วยประมวลผลในคอมพิวเตอร์เครื่องนั้นโดยการแมปปิ้ง (mapping) จากคำสั่งไบต์โค้ดไปเป็นเนทีฟโค้ดที่ทำหน้าที่ของคำสั่งนั้น แล้วส่งเนทีฟโค้ดนั้นให้หน่วยประมวลผลทำงาน ซึ่งส่งผลให้การทำงานของโปรแกรมที่ทำงานโดยอินเทอร์พรีเตอร์ย่อมช้ากว่าโปรแกรมที่ทำงานโดยตรงแบบภาษาซีหรือซีพลัสพลัส ในปัจจุบันได้มีการพัฒนาอินเทอร์พรีเตอร์ที่ไม่ใช้การแปลไบต์โค้ดทีละคำสั่งแต่จะเปลี่ยนไบต์โค้ดทั้งโปรแกรมเป็นโปรแกรมเนทีฟโค้ดจนเสร็จแล้วจึงทำงาน วิธีนี้เรียกว่าจัสอินไทม์ (Just In Time หรือ JIT) โดยโปรแกรมจะทำงานเร็วขึ้นมาก เนื่องจากคำสั่งที่ถูกทำซ้ำบ่อย ๆ จะถูกแปลเพียงครั้งเดียว

2.3.2 การคอมไพล์

คอมไพเลอร์ของภาษาจาวาจะสร้างไบต์โค้ดจากภาษาในระดับที่สูงกว่าเพื่อให้หน่วยประมวลผลสามารถนำไปใช้งานได้ แต่ในความเป็นจริงนั้นหน่วยประมวลผลที่ทำหน้าที่นี้ไม่มีอยู่จริงเป็นเพียงสิ่งที่สมมติขึ้นมาซึ่งก็คือ จาวาเวอร์ชวลแมชีน นอกจากนี้สิ่งที่ต่างไปจากภาษาอื่นคือคอมไพเลอร์ของภาษาจาวาจะไม่เปลี่ยนหรือลดรูปแบบการอ้างถึงส่วนต่าง ๆ ของโปรแกรมจากการใช้ชื่อไปเป็นตัวเลขหรือแอดเดรสที่กำหนดขึ้นจากการจัดวางตำแหน่งของโปรแกรมลงในหน่วยความจำในระหว่างการคอมไพล์ ด้วยเหตุผลสำคัญคือ เพื่อเป็นการสร้างความพอร์ทเทเบิลให้กับการตัวโปรแกรมเพราะการจัดวางตำแหน่งของโปรแกรมจะต้องขึ้นอยู่กับลักษณะการทำงานของหน่วยประมวลผลตัวใดตัวหนึ่ง ด้วยเหตุนี้ทำให้โปรแกรมที่ได้จากการคอมไพล์มีความเป็นกลางสามารถนำไปใช้บนแพลตฟอร์มอื่นได้ ดังรูปที่ 2-2



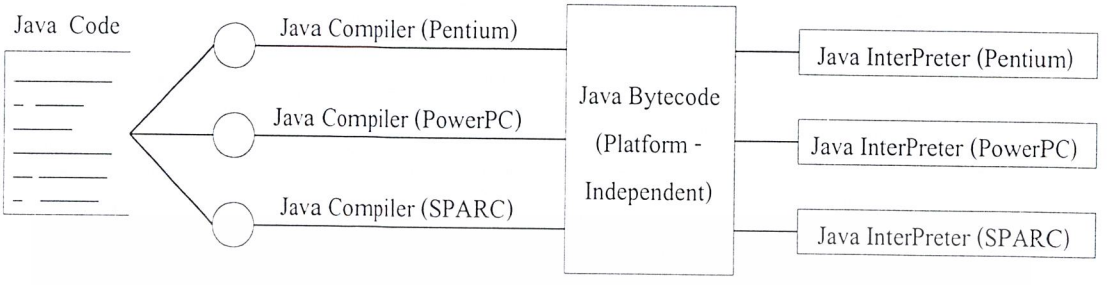
รูปที่ 2-2 แสดงการประมวลผลของโปรแกรมทั่วไปแต่ระบบปฏิบัติการที่ต่างกัน

2.3.3 การรันโปรแกรม

อินเทอร์พรีเตอร์มีหน้าที่ในการรันโปรแกรม (ไบต์โค้ด) ที่ได้จากการคอมไพล์ โดยขั้นตอนในการรันโปรแกรม 3 ขั้นตอน ซึ่งเริ่มต้นจากการอ่านไบต์โค้ดเข้าสู่ระบบซึ่งเป็นหน้าที่ของคลาสโหลดเดอร์ (class loader) ที่จะอ่านคลาสที่เรียกใช้รวมไปถึงคลาสที่อ้างถึงและคลาสที่ถูกสืบทอดมาโดยคลาสที่อ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึง จากนั้นไบต์โค้ดทั้งหมดจะถูกส่งผ่านตัวตรวจสอบไบต์โค้ดเพื่อให้มั่นใจว่าถูกต้องตามมาตรฐานของจาวาและจะไม่มีผลกระทบต่อเสถียรภาพของระบบ สุดท้ายจะส่งไบต์โค้ดที่ผ่านการตรวจสอบแล้วไปยังระบบรันไทม์ (Run-Time System) ซึ่งจะสั่งงานไปยังฮาร์ดแวร์อีกต่อหนึ่ง



รูปที่ 2-3 แสดงการประมวลผลของโปรแกรมภาษาจาวา

2.3.3.1 การอ่านไบต์โค้ด

ตัวคลาสโหลดเดอร์จะทำหน้าที่ดึงไบต์โค้ดทั้งหมดที่จำเป็นในการทำงานของแอพพลิเคชันไม่ว่าจะเป็นคลาสที่มีการเรียกใช้หรือคลาสที่ถูกสืบทอดมา แต่ถ้าหากเป็นคลาสที่อยู่บนเครื่องนั้นก็จะมีห้วงแอดเดรส (address space) เป็นของตนเองซึ่งสามารถใช้เนมสเปซ (namespace) เก็บชื่อคลาสร่วมกันได้ แต่ถ้าเป็นคลาสที่ดึงมาจากภายนอกจะมีการกำหนดเนมสเปซของแต่ละคลาสแยกกันเพื่อป้องกันการรบกวนกันระหว่างคลาสต่างๆ ขณะเรียกใช้ และเมื่อคลาสทั้งหมดที่เกี่ยวข้องถูกดึงเข้ามาเรียบร้อยแล้ว ตัวคลาสโหลดเดอร์จะสร้างตารางสำหรับค้นหาแอดเดรสที่ตรงกันกับชื่อคลาสที่เรียกใช้

2.3.3.2 การตรวจสอบไบต์โค้ด

ตัวตรวจสอบไบต์โค้ดหรือเวอร์ริไฟเออร์ (verifier) จะตรวจสอบความถูกต้องของไบต์โค้ดที่ดึงเข้ามาทีละบรรทัดว่าเป็นไปตามข้อกำหนดของจาวาและสอดคล้องกับการทำงานของตัวโปรแกรมเองหรือไม่ ตัวอย่างเช่น จะไม่มีการสร้างพอยน์เตอร์ที่เกินกว่าหน่วยความจำจริง, ไม่มีคำสั่งใดที่ไปละเมิดสิทธิ์ของโปรแกรมอื่น, ไม่มีการจับคู่ออบเจ็กต์ผิด, จะไม่มีการให้โอเปอแรนด์มากหรือน้อยเกินไป, การกำหนดค่าต่าง ๆ สำหรับไบต์โค้ดถูกต้องครบถ้วน, จะไม่มีการแปลงข้อมูลผิดรูปแบบ เป็นต้น โดยการตรวจสอบไบต์โค้ดเหล่านี้จะทำเพียงครั้งเดียวเท่านั้นแม้ว่าจะมีการเรียกกลับมาทำงานที่ส่วนเดิมของโปรแกรม

จุดประสงค์ของการตรวจสอบคือ เพื่อให้อินเทอร์พรีเตอร์มั่นใจได้ว่าไบต์โค้ดที่ส่งเข้าไปทำงานจะไม่มีขั้นตอนการทำงานที่สร้างปัญหาให้กับตัวระบบปฏิบัติการ และเพื่อให้การทำงานตามไบต์โค้ดเป็นไปอย่างรวดเร็วกว่าเพราะไม่ต้องระวังหรือหยุดแก้ไขปัญหาที่เกิดขึ้นระหว่างการทำงาน

2.3.3.3 การทำงานตามโค้ด

โค้ดที่ผ่านการตรวจสอบแล้วจะถูกส่งไปยังตัวอินเทอร์พรีเตอร์เพื่อให้ทำงานตามคำสั่งของโค้ด โดยการเปลี่ยนโค้ดให้กลายเป็นคำสั่งสำหรับหน่วยประมวลผลซึ่งมี 2 แบบดังนี้ แบบแรกตัวอินเทอร์พรีเตอร์จะคอมไพล์ไบต์โค้ดเหล่านี้ให้เป็นเนทีฟโค้ดที่ตัวเครื่องคอมพิวเตอร์นั้นเข้าใจแล้วจึงทำงานเพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ความเร็วในการทำงานสูงสุด ส่วนแบบที่สองคือตัวอินเทอร์พรีเตอร์จะอ่านโค้ดเข้ามาแล้วแปลทีละคำสั่งจากนั้นจึงทำงานพร้อมทั้งแปลคำสั่งอื่นต่อไปเรื่อย ๆ ตลอดการทำงาน

โดยสรุปแล้วระบบรันไทม์ที่ดีจะต้องถ่วงดุลย์ความสำคัญ 3 ประการให้พอเหมาะนั่นคือความพอร์ทเทเบิล, ความปลอดภัย และสมรรถนะ ในเรื่องของความพอร์ทเทเบิลทำได้โดยการใช้รูปแบบของไบต์โค้ดที่เป็นกลางสามารถนำไปรันบนเครื่องคอมพิวเตอร์แบบต่างๆ ได้โดยง่าย ซึ่งก็คือการจัดวางตำแหน่งในหน่วยความจำในช่วงรันไทม์นั่นเอง เรื่องความปลอดภัยก็ทำโดยตัวตรวจสอบไบต์โค้ด ส่วนเรื่องของสมรรถนะก็จัดการโดยพยายามเอาโอเวอร์เฮดไปใส่ไว้ตอนเริ่มต้นโหลดโปรแกรมเข้าสู่ระบบหรือไม่ก็กำหนดให้ทำงานเป็นแบบแบกกราวนด์ (Background Thread) ซึ่งสามสิ่งเหล่านี้จาวาก็สามารถจัดการได้อย่างน่าพอใจ

2.4 การสร้างและรันโปรแกรมภาษาจาวา

การสร้างโปรแกรมภาษาจาวาเริ่มจากการเขียนซอร์สโค้ดแล้วบันทึกเป็นไฟล์ที่มีนามสกุลเป็น .java จากนั้นนำไปคอมไพล์ด้วยคอมไพล์เลอร์ของภาษาจาวาเพื่อแปลงไปเป็นโปรแกรมที่ประกอบด้วยคำสั่งของจาวาเวอร์ชวลแมชีนซึ่งมีนามสกุลเป็น .class ที่เรียกว่าไฟล์จาวาคลาสหรือไบต์โค้ด ซึ่งไฟล์จาวาคลาสนี้ยังไม่สามารถรันได้ทันทีเหมือนไฟล์นามสกุล .exe หรือ .com ที่รู้จักกัน ถ้าหากจะใช้งานก็ต้องใช้อินเทอร์พรีเตอร์ของภาษาจาวาสำหรับระบบนั้น ๆ จัดการแปลไบต์โค้ดให้เป็นโค้ดที่ระบบปฏิบัติการเข้าใจและสามารถทำงานตามคำสั่งได้

แต่ก่อนที่จะสร้างโปรแกรมภาษาจาวาได้จะต้องมีคอมไพเลอร์และอินเทอร์พรีเตอร์ของภาษาจาวาเสียก่อน ปัจจุบันชุดโปรแกรมเหล่านี้ได้ถูกพัฒนาออกมาจากหลายบริษัท เช่น Java Developer Kit (JDK) และ Software Development Kit (SDK) ของบริษัทจาวาซอฟต์, JBuilder ของบริษัทบอร์แลนด์ (Borland), Visual Café ของบริษัทไซแมนเทค (Symantec), Visual J++ 6.0 ของบริษัทไมโครซอฟท์ (MicroSoft) เป็นต้น JDK เป็นโปรแกรมที่ใช้งานแบบคอมมานด์ไลน์ (Command Line) ซึ่งมีคอมไพเลอร์, อินเทอร์พรีเตอร์ และดีบักเกอร์ให้เรียกใช้ เนื่องจากบริษัทจาวาซอฟต์เป็นผู้กำหนดมาตรฐานของภาษาจาวาทำให้ JDK เป็นโปรแกรมที่มีมาตรฐานมากที่สุด ส่วนของบริษัทอื่นจะเป็นวิชวลทูล (Visual Tool หรือ Integrated Development Environment) ที่ใช้งานได้ง่ายเนื่องจากมีเอดิเตอร์และไฟล์เมเนเจอร์ให้ในตัว ซึ่งต้องเรียนรู้การใช้งานทูลก่อนรวมถึงมาตรฐานของภาษาจาวาก็อาจไม่ตรงตามที่กำหนดและยังมีกลไกของภาษาที่ไม่ใหม่เท่ากับ JDK ของบริษัทจาวาซอฟต์ด้วย

JDK ที่ถูกพัฒนาออกมาแบ่งได้เป็น 4 เวอร์ชันหลักคือ เวอร์ชัน 1.0 ที่ออกมาในช่วงปลายปีค.ศ. 1995 แต่ถือว่าล้าสมัยไปแล้ว และในช่วงปี ค.ศ.1997 ได้ออกเวอร์ชัน 1.1 ซึ่งมีการเปลี่ยนแปลงกลไกของภาษาและคลาสมาตรฐานหลายอย่าง แล้วในปลายปีค.ศ.1998 ได้มี JDK เวอร์ชัน 1.2 ออกมา (เรียกกันว่า Java2) โดยมีสิ่งที่เพิ่มขึ้นมาคือ JFC, 2D Graphics, Collection, Extension Framework, Input Method Framework, Version Identification, JavaIDL, RMI, Serialization, Audio, JAR, JINI, JDBC, Reflection และล่าสุดปีค.ศ.2000 JDK เวอร์ชัน 1.3 ก็ออกมาซึ่งได้มีการพัฒนาในเรื่องของความเร็วและเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกี่ยวกับ JFC, Drag and Drop, Applet Deployment, 2D, Security, Networking, Collection, Serialization, JAR, Reflection, java.math, java.lang และ java.util

ในการพัฒนาโครงงานได้ใช้ JDK1.2.2_th ซึ่งสนับสนุนการใช้งานภาษาไทย (Java2™ Standard Edition, Thai Localization) โดยสามารถแสดงภาษาไทยใน Swing ได้ และเลือกใช้ JBuilder มาช่วยอำนวยความสะดวกในการเขียนโปรแกรมและออกแบบส่วนติดต่อกับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การเข้ารหัสข้อมูล

3.1 ระบบการเข้ารหัสข้อมูล

การเข้ารหัสข้อมูลเป็นส่วนประกอบสำคัญในระบบข้อมูลปัจจุบัน โดยอาศัยหลักการของการเข้ารหัส (Encryption) และการถอดรหัส (Decryption) ซึ่งทั้งสองขั้นตอนจะถูกควบคุมโดยกุญแจหรือคีย์ (Key) การเข้ารหัส เป็นการเปลี่ยนรูปแบบข้อมูลโดยผ่านรูปแบบและกระบวนการแปรรูปของข้อมูลทำให้ข้อมูลที่ส่งมีรูปแบบที่ไม่เหมือนเดิมเพื่อทำให้ข้อมูลเป็นความลับ ส่วนการถอดรหัส เป็นการแปลงข้อมูลที่ได้ผ่านการเข้ารหัสให้กลับมาเป็นข้อมูลเดิม

3.1.1 ระบบการเข้ารหัสแบบคีย์เดี่ยว (Symmetric Cryptosystem หรือ Secret-Key)

การเข้ารหัสข้อมูลระบบนี้ทั้งผู้รับและผู้ส่งจะต้องมีคีย์ที่เหมือนกันและเป็นความลับสำหรับใช้ในการเข้ารหัสและถอดรหัสข้อมูล ซึ่งมีวิธีในการเข้ารหัสและถอดรหัสดังรูปที่ 3-1 หากคีย์ที่ใช้ต่างกันก็จะทำให้ข้อมูลที่สื่อสารกันผิดพลาด ตัวอย่างของการเข้ารหัสระบบนี้ได้แก่ การเข้ารหัสแบบ DES, 3DES, IDEA เป็นต้น

รูปที่ 3-1 แสดงการเข้ารหัสและถอดรหัสแบบคีย์เดี่ยว

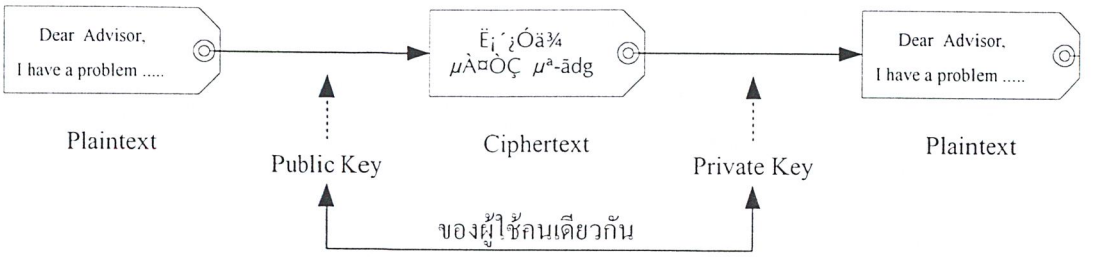
3.1.2 ระบบการเข้ารหัสแบบคีย์คู่ (Asymmetric Cryptosystem หรือ Public-Key)

การเข้ารหัสข้อมูลระบบนี้ประกอบด้วยคีย์ 2 แบบคือ

1. ไพรเวตย์คีย์หรือกุญแจส่วนตัว (Private Key) เป็นคีย์ที่จะต้องเก็บเป็นความลับ
2. พับลิกคีย์หรือกุญแจสาธารณะ (Public Key) เป็นคีย์ที่สามารถเปิดเผยให้ผู้อื่นทราบได้

ซึ่งกุญแจทั้งสองนี้จะเป็นกุญแจที่ต่างกัน โดยมีวิธีในการเข้ารหัสและถอดรหัส ดังรูปที่ 3-2 ตัวอย่างเช่นการเข้ารหัสแบบ RSA หลักการทำงานของการเข้ารหัสแบบคีย์คู่ คือ ข้อมูลที่ถูกเข้ารหัสด้วยพับลิกคีย์ของผู้ใดจะถูกถอดรหัสได้ด้วยไพรเวตคีย์ของเจ้าของเท่านั้น การเข้ารหัสแบบนี้มักนำไปใช้งานกับข้อมูลที่ไม่ยาวมากแต่ต้องการความปลอดภัยสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-2 แสดงการเข้ารหัสและถอดรหัสแบบคีย์คู่

3.2 การเข้ารหัสแบบ DES (Data Encryption Standard)

ในปลายค.ศ.1960 บริษัท IBM ได้จัดตั้งโครงงานวิจัยทางด้าน Computer Cryptography ซึ่งนำโดย Horst Feistel โครงงานนี้ได้เสร็จสิ้นในปีค.ศ.1971 และผลงานวิจัยของโครงงานนี้คือ LUCIFER [FEIS73] โดยมีลักษณะเป็นการเข้ารหัสข้อมูลเป็นบล็อกขนาด 64 บิตและใช้คีย์ขนาด 128 บิต ซึ่งต่อมาได้ถูกพัฒนาขนาดของคีย์ให้ลดลงมาเหลือ 56 บิต

โดยอัลกอริทึมของการเข้ารหัสข้อมูลของ Lucifer ได้ถูกพัฒนาโดย IBM สำหรับ NBS (National Bureau of Standards) ซึ่งเป็นที่รู้จักในชื่อ DES แต่ในสหรัฐอเมริกาเรียกว่า DEA (Data Encryption Algorithm) และสำหรับบางประเทศอาจเรียกว่า DEAI (Data Encryption Algorithm-1)

3.2.1 รายละเอียดของ DES

DES เป็นวิธีการเข้ารหัสที่ใช้กันอย่างแพร่หลาย ซึ่งข้อมูลจะถูกเข้ารหัสเป็นบล็อกโดยแต่ละบล็อกมีขนาด 64 บิต และใช้คีย์ขนาด 56 บิต สำหรับการถอดรหัสก็จะใช้คีย์ตัวเดียวกันนี้ แม้ว่า DES ถูกนำมาใช้ตั้งแต่ช่วงทศวรรษที่ 70 และได้รับการตอบรับอย่างดีจากนักวิเคราะห์รหัส (Cryptanalysis) แต่ก็ยังมีข้อถกเถียงกันอย่างมากในเรื่องของความปลอดภัยว่าน่าเชื่อถือเพียงใด แต่จากเอกสารที่ตีพิมพ์เป็นสาธารณะในปัจจุบันก็ยังไม่พบช่องโหว่ของ DES แม้ว่าจะใช้คีย์เพียงไม่กี่บิต ในทางตรงกันข้ามแนวความคิดแบบ IDEA ที่ใช้คีย์ 128 บิตซึ่งขนาดมากกว่า 2 เท่าของ DES และได้รับการตอบรับจากสาธารณะตั้งแต่ทศวรรษที่ 90 ว่ามีความปลอดภัยมากกว่าแบบ DES และประมวลผลได้เร็วกว่าแบบ DES แต่ก็ยังต้องรอการรับรองจากผู้เชี่ยวชาญอีกมากในเรื่องช่องโหว่ของความปลอดภัย

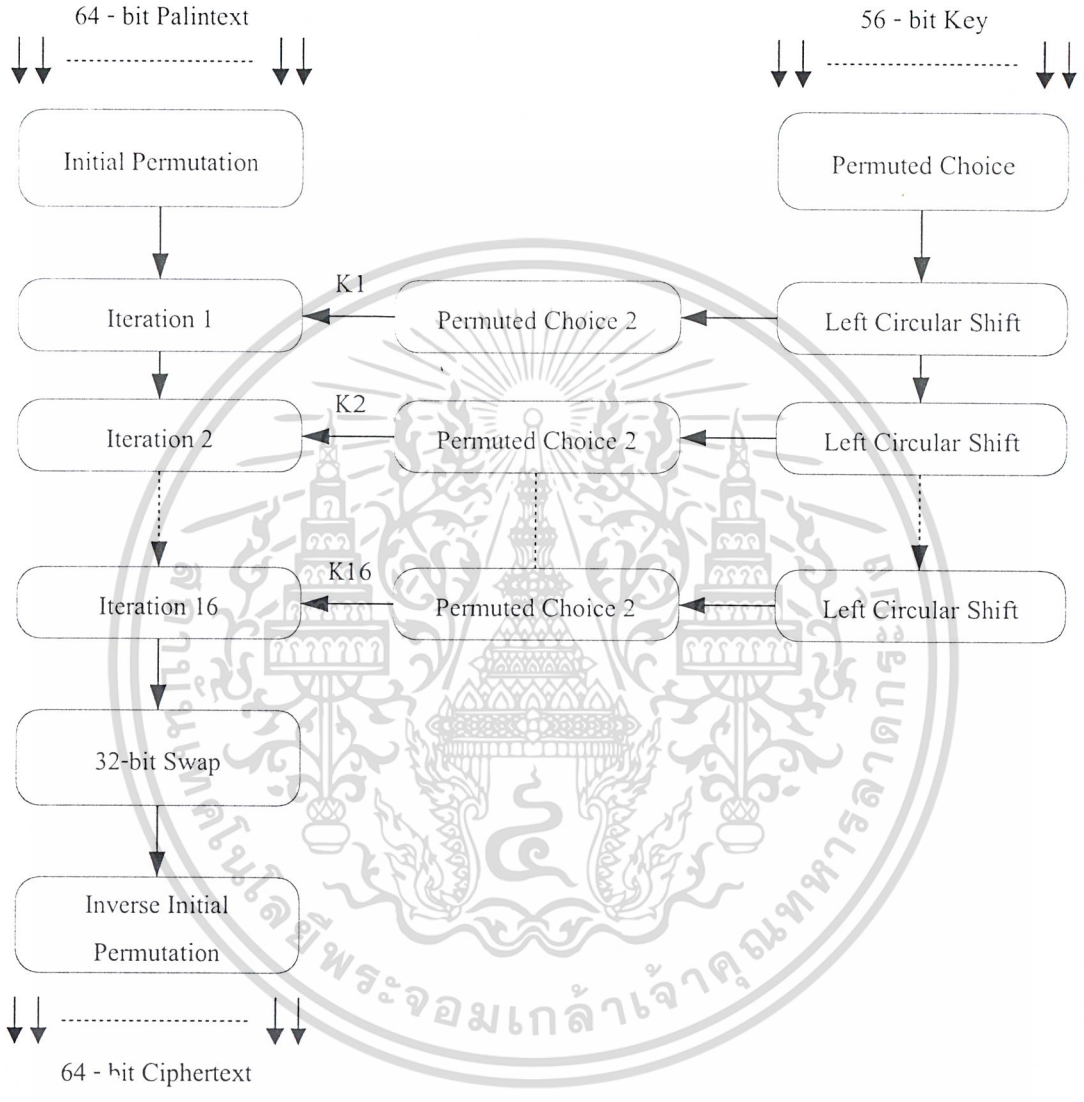
การทำงานของ DES จะมีลักษณะดังรูปที่ 3-3 โดยข้อมูลที่เข้ามาในส่วนฟังก์ชันของการเข้ารหัสจะมี 2 ส่วนด้วยกันคือ ข้อมูลที่ยังไม่ถูกเข้ารหัสขนาด 64 บิตและคีย์ขนาด 56 บิต ซึ่งในด้านซ้ายของรูปจะแสดงขั้นตอนการจัดการกับข้อมูลที่ยังไม่เข้ารหัส (Plaintext) โดยสามารถแบ่งย่อยได้เป็น 3 เฟสดังนี้

1. เฟสที่ 1 จะนำข้อมูลที่ยังไม่เข้ารหัสที่มีขนาด 64 บิตผ่านเข้าไปในส่วนที่เรียกว่า Initial Permutation (IP) ซึ่งจะสลับตำแหน่งบิตเพื่อให้ได้ข้อมูลใหม่คือ Permuted Input

2. เฟสที่ 2 จะนำข้อมูลที่ยังไม่ได้เข้ารหัสและคีย์มาทำ permutation และ substitution เป็นจำนวน 16 รอบ โดยผลลัพธ์ที่ได้จะเป็นข้อมูลขนาด 64 บิต สำหรับผลลัพธ์ 64 บิตที่ได้จะถูกแบ่งเป็นด้านซ้ายและด้านขวาแล้วนำมาสลับด้านกัน (swap) เพื่อให้กลายเป็น Preoutput ขนาด 64 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เฟสที่ 3 จะนำ Preoutput ไปผ่านในส่วนที่เรียกว่า Inversion Initial Permutation (IP^{-1}) ซึ่งทำหน้าที่กลับกันกับ Initial Permutation ซึ่งผลลัพธ์สุดท้ายที่ได้คือข้อมูลที่ผ่านการเข้ารหัสแล้ว (Ciphertext) ที่มีขนาด 64 บิต



รูปที่ 3-3 แสดงขั้นตอนการทำงานของ DES

3.2.2 การทำ Initial Permutation และ Inverse Initial Permutation

การทำ Initial Permutation และ Inverse Initial Permutation จะถูกอธิบายโดยใช้ตารางที่ 3-1 และ 3-2 ตามลำดับ ซึ่งจะเห็นได้ว่าฟังก์ชัน Permutation ทั้งสองเป็นส่วนกลับซึ่งกันและกันโดยพิจารณาได้จาก 64 บิต (M) ที่เข้ามา โดย M_i คือ ตัวเลขฐานสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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}

ตารางที่ 3-1 แสดงการจัดเรียงของข้อมูลก่อนเข้าสู่ฟังก์ชัน Permutation

และเมื่อทำ Initial Permutation : X = IP(M) จะได้ดังนี้

M_{55}	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

ตารางที่ 3-2 แสดงการจัดเรียงของข้อมูลหลังจากทำตามฟังก์ชัน Permutation

ในทางกลับกันเมื่อทำ Inverse Permutation : $Y = IP^{-1}(IP(M))$ ซึ่งจะได้ข้อมูลที่มีลำดับในการเรียงของบิตเหมือนกับก่อนทำ Permutation

3.2.3 รายละเอียดของการทำฟังก์ชัน Permutation และ Substitution ในแต่ละรอบ

ข้อมูลที่ได้จากการทำ Initial Permutation ที่มีขนาด 64 บิตจะถูกแบ่งเป็น 2 ส่วนคือด้านซ้ายและขวาโดยแต่ละส่วนมีขนาด 32 บิต ซึ่งกระบวนการทำในแต่ละรอบสามารถสรุปเป็นสูตรได้ดังนี้

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

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

เมื่อ $\oplus$ หมายถึง การทำฟังก์ชัน XOR,

L_i คือ ข้อมูล 32 บิตทางด้านซ้ายของรอบปัจจุบัน,

R_i คือ ข้อมูล 32 บิตทางด้านขวาของรอบปัจจุบัน,

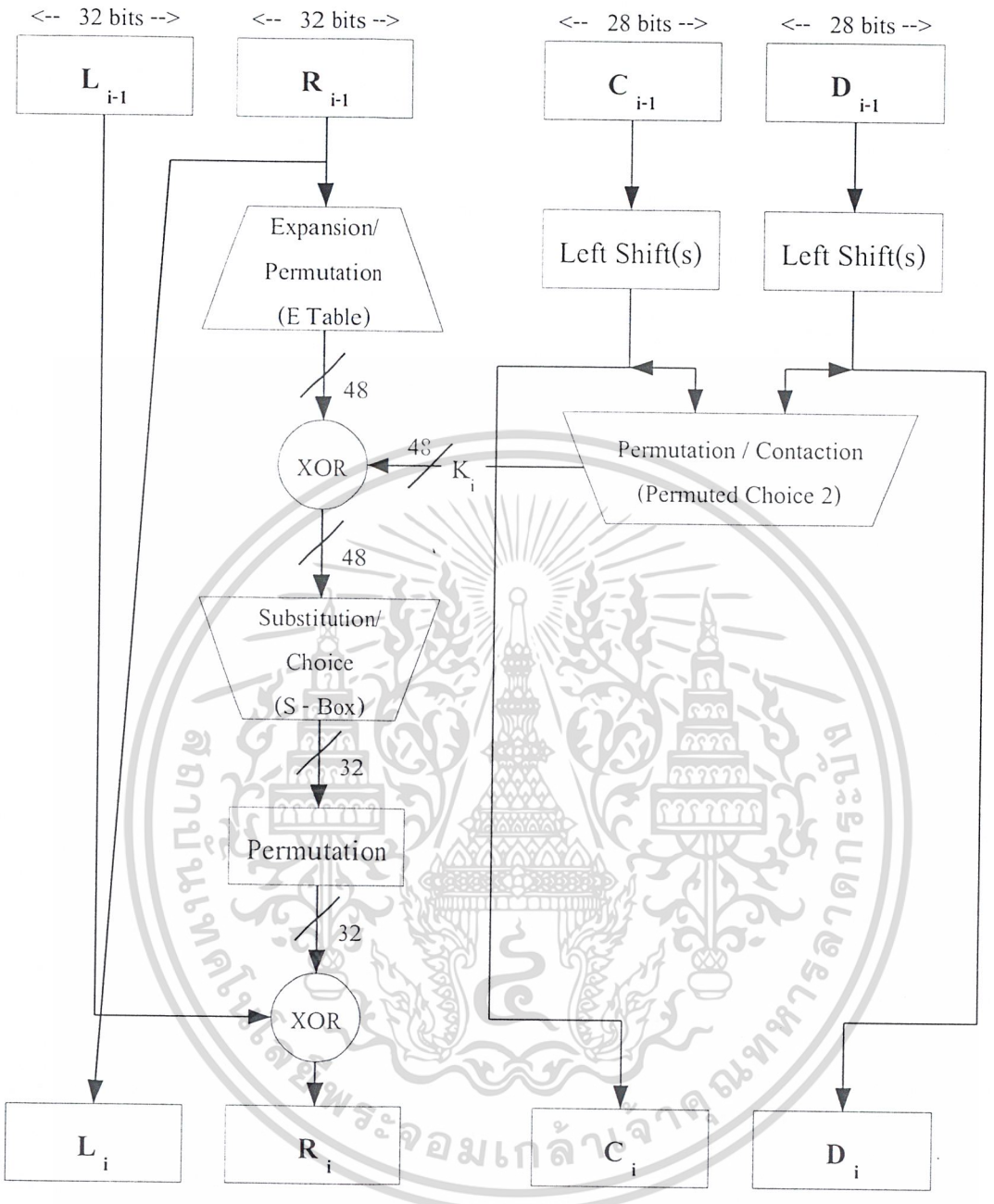
L_{i-1} คือ ข้อมูล 32 บิตทางด้านซ้ายของรอบที่ผ่านมา,

R_{i-1} คือ ข้อมูล 32 บิตทางด้านขวาของรอบที่ผ่านมา,

และ K_i คือ คีย์ที่ใช้ในแต่ละรอบ

จากสูตรจะเห็นได้ว่าข้อมูล 32 บิตด้านซ้ายมือ (L_i) จะเท่ากับด้านขวา (R_{i-1}) ของรอบที่ผ่านมา โดย R_i จะเท่ากับการนำ L_{i-1} มา XOR กับ $f(R_{i-1}, K_i)$ ซึ่งฟังก์ชัน f จะแสดงดังรูปที่ 3-4 โดยคีย์ที่ใช้ในแต่ละรอบจะมีขนาด 48 บิตและข้อมูลอินพุตด้านขวามีขนาด 32 บิต ดังนั้นจึงต้องมีการขยายขนาดจาก 32 บิตให้เป็น 48 บิตโดยการทำ Expansion Permutation ซึ่งใช้ตารางที่ 3-5 ที่ได้กำหนดไว้ ผลลัพธ์ที่ได้ที่มีขนาด 48 บิตจะถูก XOR กับ K_i แล้วนำไปผ่านฟังก์ชัน Substitution และ Permutation จะได้ข้อมูล 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-4 แสดงการเข้ารหัส DES ในแต่ละครั้ง (ทำทั้งหมด 16 ครั้ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
From 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
From 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
From Input Bit	57	49	41	33	25	17	9	1	59	51	46	35	27	19	11	3
Output Bit	49	50	51	52	53	54	55	53	57	58	59	60	61	62	63	64
From Input Bit	61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

ตารางที่ 3-3 แสดง Initial Permutation (IP) ของ DES

Output Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
From 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
From 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
From 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	53	57	58	59	60	61	62	63	64
From Input Bit	34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

ตารางที่ 3-4 แสดง Inverse Initial Permutation (IP^{-1}) ของ DES

Output Bit	1	2	3	4	5	6	7	8	9	10	11	12
From 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
From 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
From 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
From Input Bit	24	25	26	27	28	29	28	29	30	31	32	1

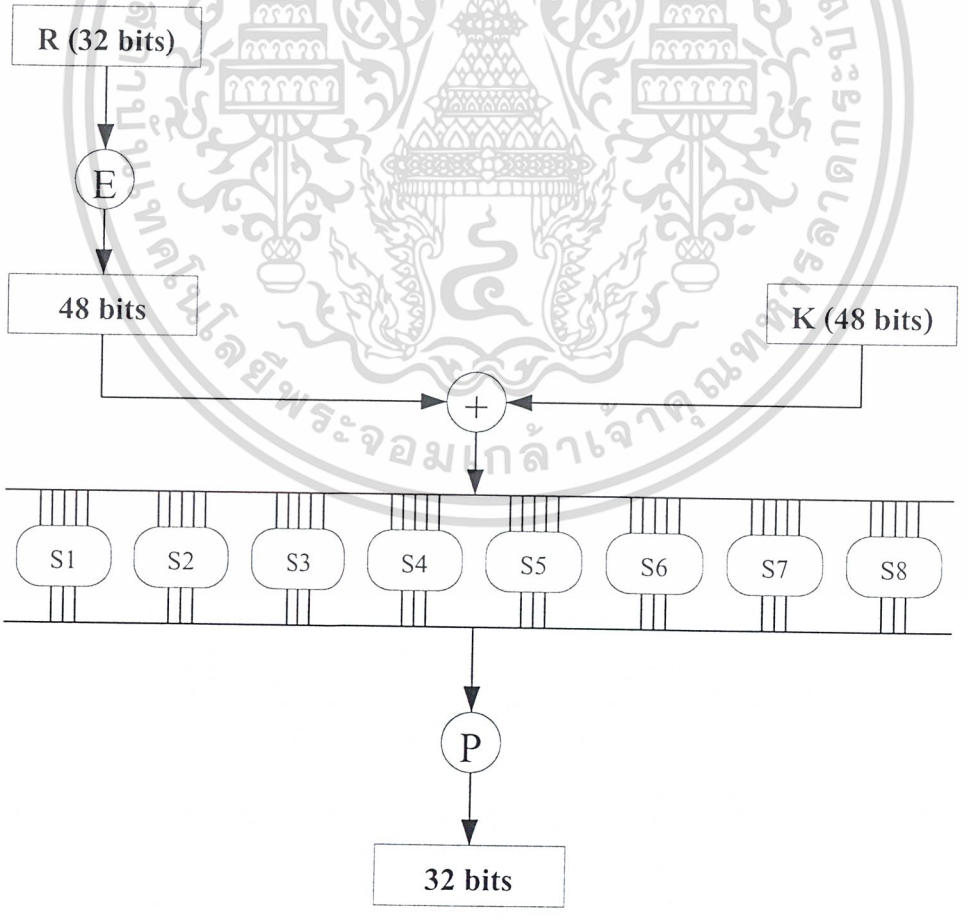
ตารางที่ 3-5 แสดง Expansion Permutation (E) ของ DES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
From 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
From Input Bit	2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

ตารางที่ 3-6 แสดง Permutation Function (P) ของ DES

การทำ Substitution ประกอบด้วยเซตของ S-Box 8 อัน ($S_1 - S_8$) ซึ่ง S-Box จะมีอินพุตขนาด 6 บิตและให้เอาต์พุตขนาด 4 บิตออกมา ซึ่งตารางที่ 3-7 จะแสดง S-Box ของ DES โดยมีวิธีการแปลงอินพุตขนาด 6 บิตให้กลายเป็นเอาต์พุตขนาด 4 บิตดังนี้ เริ่มจากนำบิตแรกและบิตสุดท้ายของอินพุตมาทำเป็นตำแหน่งของแถว แล้วนำ 4 บิตตรงกลางมาเป็นตำแหน่งของคอลัมน์ เช่น S_1 ที่มีค่าเท่ากับ 011011 (6 บิต) จะนำบิตแรก (0) และบิตสุดท้าย (1) มาเป็นตำแหน่งของแถวจะได้แถวที่ 01 ส่วน 4 บิตตรงกลางที่เหลือคือ 1101 ซึ่งก็คือตำแหน่งคอลัมน์ที่ 13 ดังนั้นเมื่อเทียบกับ S-Box ค่าที่ตำแหน่งแถวที่ 1 และคอลัมน์ที่ 13 คือ 0101 (มาจากค่า 5 ใน S_1)



รูปที่ 3-5 แสดงการทำ Permutation สำหรับ row 0 ของ S_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

←----------Column Number----------→ Box

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

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

←Row→																	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	

←Row→																	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	12	15	1	2	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	

←Row→																	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	11	14	2	13	6	15	0	9	10	4	5	3	

←Row→																	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	1	
1	10	15	4	2	7	12	9	5	6	1	12	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

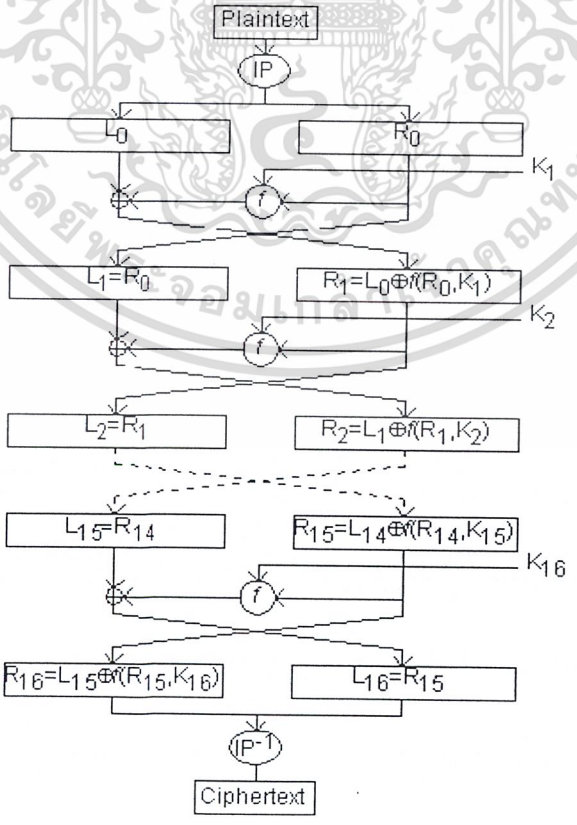
←----------Column Number----------→ Box

←Row→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S_7
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	

←Row→	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S_8
1	1	5	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

ตารางที่ 3-7 แสดง S-Box ของ DES

ในขั้นตอนสุดท้ายจะนำผลลัพธ์นี้ไป XOR กับข้อมูล 32 บิตด้านซ้ายซึ่งจะได้ข้อมูล 32 บิตเพื่อนำไปเป็นอินพุตด้านขวาของรอบต่อไป ส่วนข้อมูล 32 บิตด้านซ้ายที่จะเป็นอินพุตของรอบต่อไปคือข้อมูล 32 บิตด้านขวาของรอบนี้นั่นเอง โดยสามารถแสดงขั้นตอนการทำงานของ DES ในรูปแบบฟังก์ชันได้ดังรูปที่ 3-6



รูปที่ 3-6 แสดงขั้นตอนการทำงานของ DES โดยรวมแบบฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42838

3.2.4 การสร้างคีย์ (Key Generation)

การทำงานในแต่ละรอบของ DES จะใช้คีย์ที่มีขนาด 56 บิตมาผ่านกระบวนการดังนี้ เริ่มจากนำคีย์ 56 บิตนั้นมาทำ Permuted Choice One (PC-1) ตามตารางที่ 3-8 จากนั้นนำคีย์มาแบ่งเป็น 2 ส่วนส่วนละ 28 บิตเรียกว่าส่วน C และ D ซึ่งในแต่ละรอบจะมีการทำ Circular Left Shift หรือ Rotation ในส่วน C และ D โดยในแต่ละรอบจะมีการกำหนดว่าจะให้ชิพ (shift) ไปกี่บิตดังตารางที่ 3-10 ซึ่งค่าที่ถูกชิพจะกลายเป็นอินพุตของการทำในรอบถัดไปและเป็นอินพุตของการทำ Permuted Choice Two (PC-2) ดังตารางที่ 3-9 และหลังจากการทำ PC-2 จะได้เอาต์พุตขนาด 48 บิต ซึ่งจะเป็นอินพุตของ $f(R_{i-1}, K_i)$

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	17	18	19	20	21	22	23	24	25	26	27	28	29	30
From Input Bit	10	2	59	51	43	35	27	19	11	3	60	52	44	36
Output Bit	33	34	35	36	37	38	39	40	41	42	43	44	45	46
From Input Bit	63	55	47	39	31	23	15	7	62	54	46	38	30	22
Output Bit	33	34	35	36	37	38	39	40	41	42	43	44	45	46
From Input Bit	14	6	61	53	45	37	29	21	13	5	28	20	12	4

ตารางที่ 3-8 แสดงการคำนวณของ Permuted Choice One (PC-1)

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

ตารางที่ 3-9 แสดงการคำนวณของ Permuted Choice One (PC-2)

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

ตารางที่ 3-10 แสดงจำนวนบิตที่ทำ Circular Left Shift

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การถอดรหัสข้อมูล DES

กระบวนการถอดรหัสโดยใช้ DES นั้นเหมือนกับขั้นตอนการเข้ารหัส ซึ่งมีขั้นตอนในการทำงานดังนี้คือ นำข้อมูลที่ผ่านการเข้ารหัสแล้วมาเป็นอินพุต แต่จะมีการใช้คีย์ (K_i) ที่มีลำดับย้อนกลับกับคีย์ที่ใช้ในการเข้ารหัส (จาก $K_1, K_2, \ldots, K_{16}$ เป็น $K_{16}, K_{15}, \ldots, K_1$) โดย K_{16} เป็นคีย์แรกที่ใช้ในการเข้ารหัสแทน ดังรูปที่ 3-7 ด้านซ้ายจะเป็นขั้นตอนการเข้ารหัส ส่วนด้านขวาจะเป็นขั้นตอนการถอดรหัส

เราจะแสดงถึงผลลัพธ์ของขั้นตอนแรกในกระบวนการถอดรหัส ซึ่งจะเท่ากับ 32 บิตที่ถูก swap จากอินพุตของการทำทั้งหมด 16 รอบของการเข้ารหัส ดังต่อไปนี้

เมื่อกำหนดให้อินพุตของกระบวนการถอดรหัส คือ L_{16} และ R_{15} จะได้ว่า

$$L_{16} = R_{15}$$

$$R_{16} = L_{15} \oplus f(R_{15}, K_{16})$$

ในส่วนของการถอดรหัสจะเริ่มจาก

$$Ld_0 = R_{16}$$

$$Rd_0 = L_{16}$$

$$Ld_1 = Rd_0 = L_{16} = R_{15}$$

$$Rd_1 = Ld_0 \oplus f(Rd_0, K_{16})$$

$$= R_{16} \oplus f(R_{15}, K_{16})$$

$$= [L_{15} \oplus f(R_{15}, K_{16})] \oplus f(R_{15}, K_{16})$$

เนื่องจากคุณสมบัติของ XOR ที่สำคัญคือ

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

$$D \oplus D = 0$$

$$E \oplus 0 = E$$

ดังนั้นเราจะได้ $Ld_1 = R_{15}$ และ $Rd_1 = L_{15}$ จะได้เอาต์พุตของขั้นตอนแรกในการถอดรหัสคือ $L_{15} \| R_{15}$ ซึ่งเราสามารถเขียนเป็นสมการในการถอดรหัสได้ดังนี้

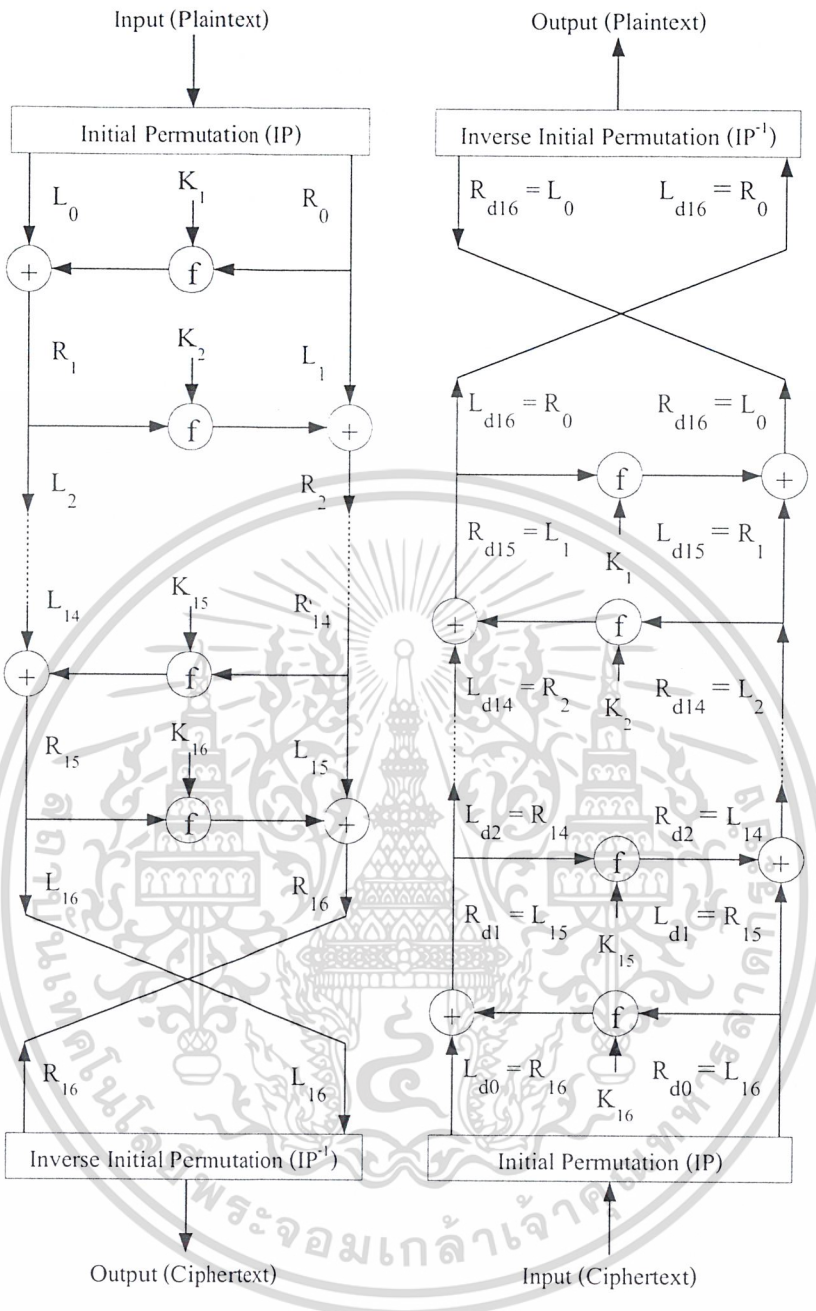
$$R_i^{-1} = L_i$$

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

โดยผลลัพธ์สุดท้ายที่ได้จากขั้นตอนสุดท้ายในการถอดรหัสคือ $R_0 \| L_0$ และนำไปสู่ขั้นตอนในการทำ Inverse Permutation ซึ่งจะได้ข้อมูลก่อนการเข้ารหัสที่ส่งมา ดังสมการต่อไปนี้

$$IP^{-1}(L_0 \| R_0) = IP^{-1}(IP(\text{Plaintext})) = \text{Plaintext}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

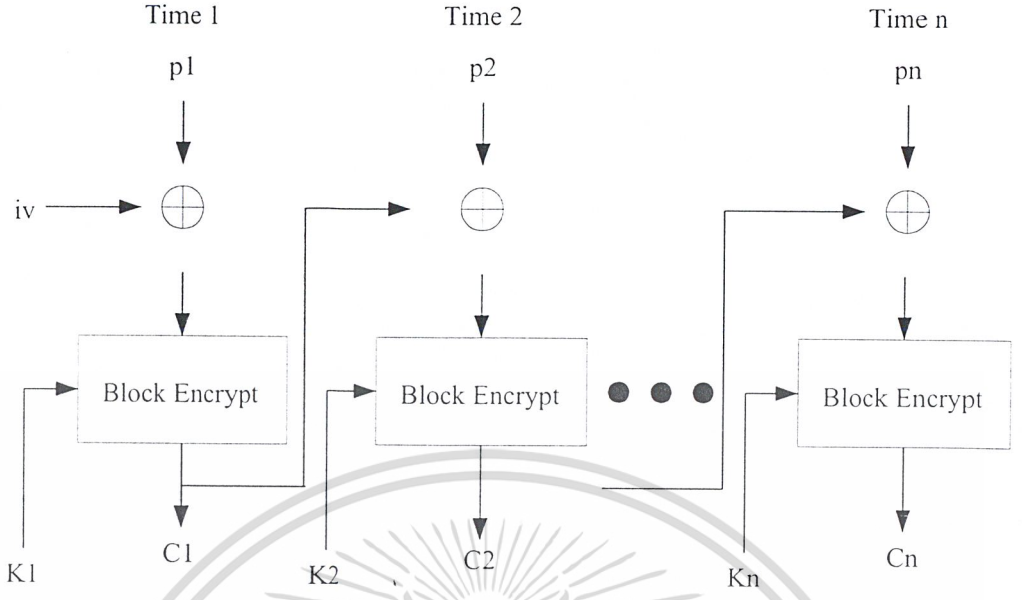


รูปที่ 3-7 แสดงคีย์และขั้นตอนการเข้ารหัสและถอดรหัส DES

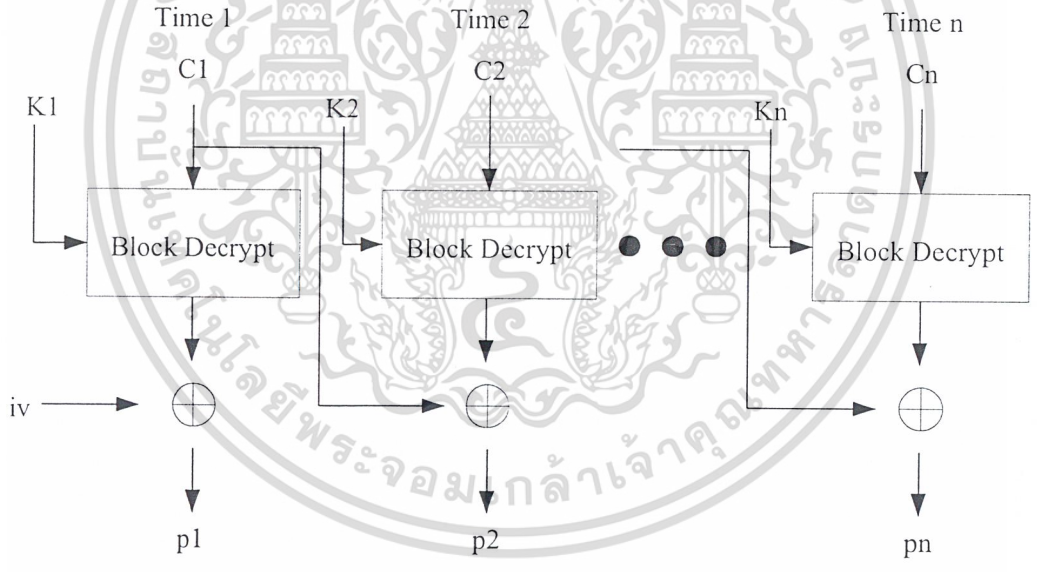
3.2.6 โหมด CBC (Cipher Block Chaining)

เป็นวิธีการเข้ารหัสที่พัฒนามาจาก DES ช่วยทำให้ข้อมูลที่ส่งมีความปลอดภัยมากยิ่งขึ้น โดยจะนำข้อมูลตัวก่อนที่ผ่านการเข้ารหัสแล้วมา XOR กับข้อมูลที่ยังไม่ได้เข้ารหัสของตัวถัดไป จากนั้นจึงนำไปเข้ารหัสแบบ DES ตามปกติ ดังรูปที่ 3-8 และ 3-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงการเข้ารหัสของ DES CBC



รูปที่ 3-9 แสดงการถอดรหัสของ DES CBC

ในการถอดรหัสข้อมูลก็จะทำเช่นเดียวกับการถอดรหัสข้อมูลแบบ DES ปกติ แต่จะนำผลลัพธ์จากการถอดรหัสมา XOR กับข้อมูลที่ผ่านการเข้ารหัสตัวก่อนหน้า เพื่อให้ได้ข้อมูลก่อนการเข้ารหัสจริง ๆ ดังสมการ

$$C_n = E_k [C_n^{-1} \oplus P_n]$$

สมการในการถอดรหัสคือ

$$D_k[C_n] = D_k [E_k (C_n^{-1} \oplus P_n)]$$

$$D_k[C_n] = C_n^{-1} \oplus P_n$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

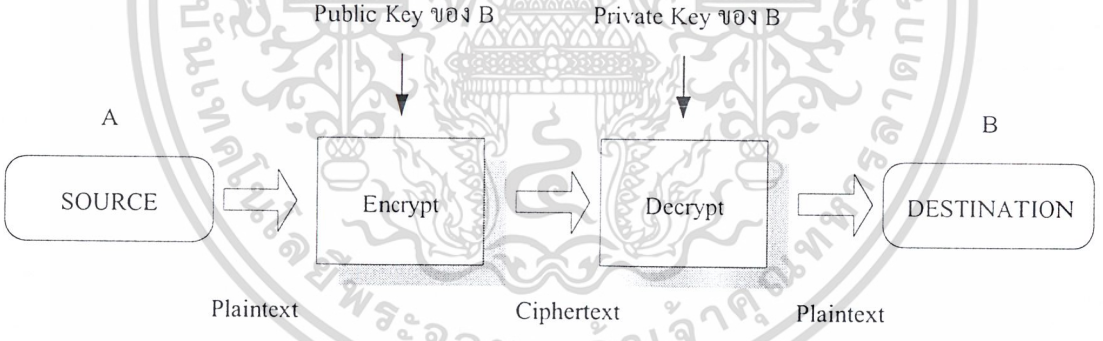
โดยผลลัพธ์สุดท้ายที่ได้ก็คือข้อมูลก่อนการเข้ารหัส ดังสมการนี้

$$C_n^{-1} \oplus D_k[C_n] = C_n^{-1} \oplus C_n^{-1} \oplus P_n = P_n$$

จะสังเกตเห็นได้ว่าบล็อกแรกของข้อมูลที่ผ่านการเข้ารหัสจะมีการนำ iv (Initialization Vector) มาทำ XOR กับข้อมูลที่ไม่ได้เข้ารหัสก่อนจะนำไปเข้ารหัสแบบ DES ตามปกติ และในส่วนของการถอดรหัสข้อมูลก็จะใช้ iv ในการถอดรหัสข้อมูลเช่นเดียวกัน ดังนั้นจึงจำเป็นต้องมีข้อตกลงกันระหว่างผู้ส่งกับผู้รับก่อนว่าจะใช้ iv ค่าใด เพื่อให้มีความปลอดภัยสูงสุด และสำหรับ iv ก็ควรจะมีการป้องกันเช่นเดียวกับคีย์ที่ใช้ในการส่ง ค่า iv อาจจะส่งโดยใช้การเข้ารหัสแบบ ECB

3.3 การเข้ารหัสแบบ RSA

การเข้ารหัสแบบ RSA ถูกคิดขึ้นในปี 1977 โดยชื่อ RSA นี้ได้มาจากอักษรตัวแรกของนามสกุลของผู้ร่วมกันคิดค้นคือ Ron River, Adi Shamir และ Leonard Adleman การเข้ารหัสแบบนี้เป็นวิธีการเข้ารหัสแบบกุญแจสาธารณะที่พัฒนาขึ้นมาเพื่อแก้ปัญหาการทำให้กุญแจเป็นความลับ โดยผู้ที่จะใช้การเข้ารหัสแบบนี้จะต้องมีกุญแจ 2 แบบคือ กุญแจส่วนบุคคลหรือไพรเวตคีย์ซึ่งต้องเก็บไว้เป็นความลับ และกุญแจสาธารณะหรือพับลิกคีย์ที่นำไปให้ใครก็ได้ที่ต้องการส่งเอกสารให้แก่เจ้าของกุญแจ หลักการทำงานคือ ข้อมูลที่ถูกเข้ารหัสด้วยพับลิกคีย์ของผู้ใดจะถูกถอดรหัสได้ด้วยไพรเวตคีย์ของผู้นั้นเท่านั้น ซึ่งสามารถอธิบายการทำงานได้ดังรูปที่ 3-10



รูปที่ 3-10 แสดงขั้นตอนการเข้ารหัสและถอดรหัสแบบ RSA

3.3.1 รายละเอียดของ RSA

ถ้าให้ p และ q เป็นจำนวนเฉพาะที่มีค่ามาก โดยที่ $n = p*q$ ซึ่งเรียก n ว่าโมดูลัส (modulus) จากนั้นจึงเลือก e ที่มีค่าน้อยกว่า n และ e ก็ไม่สามารถหาร (p-1)(q-1) ได้ลงตัว ถ้าให้ d เป็นส่วนกลับของ e ในคณิตศาสตร์ระบบโมดูโลฐาน (p-1)(q-1) นั่นคือ

$$e*d \bmod (p-1)(q-1) = 1$$

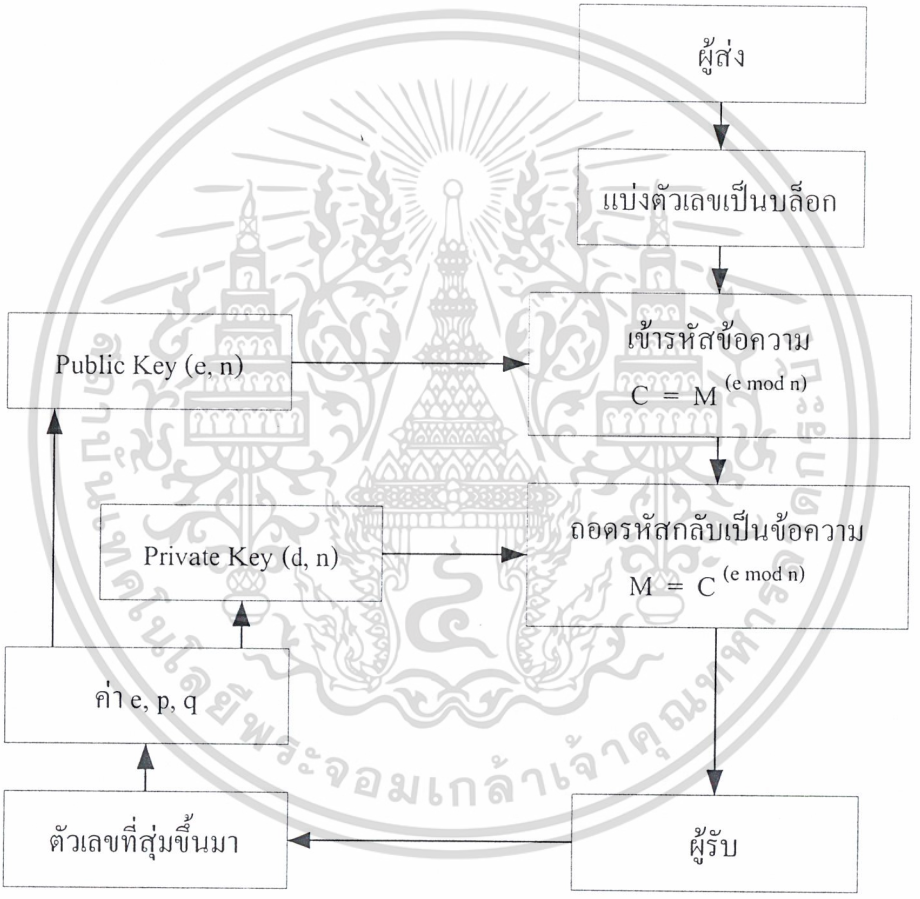
ในการเข้ารหัสแบบ RSA นั้น (n, e) คือ พับลิกคีย์ ส่วน d คือ ไพรเวตคีย์ เมื่อได้ค่าเหล่านี้แล้ว p, q จะต้องเก็บเป็นความลับหรือถูกทำลายในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่างเช่น ถ้าเอต้องการส่งเอกสารส่วนตัว M ไปให้บี เอจะสร้างเอกสารที่เข้ารหัส C จาก M ได้โดยการให้ $C = M^{(e \bmod n)}$ โดย (n, e) คือ พับลิกคีย์ของบี เมื่อบีได้รับเอกสาร C บีจะถอดรหัสด้วย d เพื่ออ่านเอกสาร เพราะความสัมพันธ์ระหว่าง e, d และ n ในสมการที่ผ่านมาจะทำให้

$$C^d = M^{ed \bmod (p-1)(q-1)} \bmod n = m \quad \text{.....(3.2)}$$

เนื่องจากมีเพียงบีเท่านั้นที่รู้ค่า d จึงมีเพียงบีที่สามารถถอดรหัสได้ คุณสมบัติที่สำคัญอีกประการหนึ่งของ RSA ที่เห็นได้จากสมการนี้คือ ถ้าเราเข้ารหัสข้อมูลด้วยไพรเวตคีย์ เราก็สามารถถอดรหัสด้วยพับลิกคีย์ได้ด้วยเช่นกัน ด้วยคุณสมบัติข้อนี้ทำให้ RSA มีประโยชน์มากเพราะใช้ในระบบลายเซ็นดิจิตอลได้ด้วย



รูปที่ 3-11 แสดงขั้นตอนการเข้ารหัสแบบ RSA

ตัวอย่างขั้นตอนการคำนวณหาคีย์ที่ใช้ในการเข้ารหัสแบบ RSA มีดังนี้

1. เลือกจำนวนเฉพาะสองจำนวนที่มีค่ามาก (ในที่นี้ขอเลือกตัวเลขที่มีค่าไม่มากเพื่อให้เห็นเข้าใจง่าย) คือ p = 7, q = 17

2. คำนวณหา n จะได้ n = p*q = 7*17 = 119

3. คำนวณ (p-1)(q-1) = 96

4. เลือก e ที่มีค่าน้อยกว่า n และ ไม่สามารถหาร (p-1)(q-1) ได้ลงตัว คือ e = 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คำนวณค่า d ซึ่งมีความสัมพันธ์กับสมการ e*d = 1 mod 96 โดยจะได้ค่า d = 77 เนื่องจาก 5*77 = 385 = 4*96 + 1

ดังนั้นพับลิกคีย์คือ (119, 5) และไพรเวตคีย์คือ (77,119)

วิธีการทำลายรหัส RSA ที่แพร่หลายที่สุดคือการหาค่า d นั่นเองซึ่งหาได้จากค่า p และ q เนื่องจาก p และ q เป็นจำนวนเฉพาะที่ p*q = n ดังนั้นการทำลายรหัส RSA ขึ้นอยู่กับการแยกตัวประกอบของ n นั่นเอง ซึ่งวิธีการแยกตัวประกอบของ n นั้นจะทำได้ยากหาก n มีค่ามาก ดังนั้นข้อมูลที่มีความสำคัญมากอาจจำเป็นต้องใช้ค่า n ที่มากถึง 700 หรือ 1000

สิ่งที่พบในการเข้ารหัสแบบนี้คือไม่ว่าจะเป็นการเข้ารหัสหรือถอดรหัสจำเป็นต้องใช้การยกกำลังในระบบโมดูโล ซึ่งสามารถทำได้โดยการใช้วงจรคูณระบบโมดูโลมาต่ออนุกรมกัน ดังนั้นความเร็วของการเข้ารหัสและถอดรหัสจึงขึ้นกับความเร็วของวงจรคูณระบบโมดูโลเป็นอย่างมาก ซึ่งถือว่าเป็นจุดอ่อนข้อหนึ่งของ RSA เมื่อเปรียบเทียบกับการเข้ารหัสแบบกุญแจเดียว เช่น DES เพราะปัจจุบันการคูณในระบบโมดูโลยังค่อนข้างยุ่งยากเมื่อเปรียบเทียบกับการแทนค่าหรือสลับตำแหน่งใน DES โดยได้มีการประมาณกันว่าถ้าหากเป็นการเข้ารหัสโดยซอฟต์แวร์สำหรับ n ที่มีความยาว 512 บิต การเข้ารหัสแบบ DES จะเร็วกว่าแบบ RSA ประมาณ 100 เท่า แต่อาจเร็วกว่าถึง 1000 หรือ 10000 เท่าหากทำการเข้ารหัสด้วยฮาร์ดแวร์ โดยทั่วไปแล้วเราต้องการให้การเข้ารหัสเร็วกว่าการถอดรหัส ดังนั้นจึงเลือกใช้ e ที่มีค่าน้อยกว่า d รวมถึงมักให้ค่า e ของทุกคนมีค่าเดียวกันเพื่อให้ฮาร์ดแวร์ของวงจรเข้ารหัสสำหรับแต่ละคนมีลักษณะคล้ายกัน

3.4 การนำการเข้ารหัสแบบ DES และแบบ RSA ไปใช้งาน

เนื่องจาก DES และ RSA มีข้อดีข้อเสียที่แตกต่างกันซึ่งเหมาะกับสถานการณ์ที่ต่างกันไป โดยทั่วไปแล้ว DES มักถูกนำมาใช้ในการเข้ารหัสข้อมูลที่มีขนาดใหญ่เพราะรวดเร็วกว่า ในขณะที่ RSA ถูกใช้ในระบบสื่อสารที่ไม่ยาวนักแต่ต้องการความปลอดภัยสูง ในบางครั้งมีการใช้ RSA ร่วมกับ DES เพื่อเสริมจุดเด่นซึ่งกันและกัน เช่น ตัวเอกสารจริงจะถูกเข้ารหัสด้วย DES โดยที่คีย์ DES จะถูกเข้ารหัสด้วย RSA แล้วส่งไปด้วยกันหรือส่งไปก่อน แต่ในบางครั้งก็ใช้ DES เพียงอย่างเดียวก็พอถ้าหากการแลกเปลี่ยนคีย์สามารถทำได้อย่างปลอดภัย หรือในกรณีที่ผู้ส่งและผู้รับเป็นคนเดียวกัน เช่น ฮาร์ดดิสก์ในคอมพิวเตอร์ส่วนตัว, ข้อมูลส่วนตัวในสมาร์ตการ์ด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โพรโตคอล ICQ

เนื่องจากโปรแกรม ICQ เป็นโปรแกรมที่ใช้สำหรับติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตที่ได้รับความนิยมและมีการใช้งานกันอย่างแพร่หลายแต่ยังขาดความปลอดภัยในการรับส่งข้อมูลเราจึงต้องการพัฒนาโปรแกรมที่มีการทำงานได้เหมือนกับโปรแกรม ICQ และใช้งานร่วมกับโปรแกรม ICQ ได้ แต่มีความปลอดภัยสูงกว่า จึงได้นำโพรโตคอลของ ICQ มาใช้ในการพัฒนาโครงงานนี้

โพรโตคอล ICQ คือรายละเอียดของขั้นตอนการติดต่อสื่อสารของโปรแกรม ICQ ซึ่งโปรแกรม ICQ จะใช้งานอยู่บนโพรโตคอลพื้นฐานของเครือข่ายอินเทอร์เน็ตอีกทีหนึ่งซึ่งก็คือ UDP และ TCP โดยจะใช้โพรโตคอล UDP สำหรับการติดต่อสื่อสารระหว่าง ICQ เซิร์ฟเวอร์กับ ICQ ไคลเอนต์ และใช้โพรโตคอล TCP สำหรับการติดต่อสื่อสารระหว่าง ICQ ไคลเอนต์กับ ICQ ไคลเอนต์

4.1 เวอร์ชันของโพรโตคอล

โพรโตคอลของ ICQ มีหลายเวอร์ชันดังนี้

1. เวอร์ชัน 1 มีความสามารถน้อย ยังไม่เป็นที่นิยม ปัจจุบัน ICQ เซิร์ฟเวอร์ไม่สนับสนุนแล้ว
2. เวอร์ชัน 2 ได้พัฒนาให้มีความสามารถมากขึ้นจนเป็นที่นิยมใช้และ ICQ เซิร์ฟเวอร์ยังสนับสนุนอยู่
3. เวอร์ชัน 3 และ 4 เพิ่มคำสั่งมากขึ้นจากโพรโตคอลเวอร์ชัน 2 และเพิ่มส่วนของการตรวจสอบความถูกต้องของข้อมูลที่รับส่ง ในเวอร์ชัน 4 สามารถเข้ารหัสส่วนหัวของแพ็กเก็ตได้แต่ก็ใช้ได้ไม่นานนัก
4. เวอร์ชัน 5 มีความเสถียรมากกว่าและมีคำสั่งมากขึ้น ซึ่งในปัจจุบัน ICQ ไคลเอนต์ใช้โพรโตคอลเวอร์ชันนี้เป็นส่วนมาก

ในการพัฒนาโครงงานนี้ใช้โพรโตคอล ICQ เวอร์ชัน 2 เพราะมีคำสั่งต่างๆ ที่เพียงพอต่อการใช้งานและไม่ซับซ้อนเกินไปสามารถทำความเข้าใจได้ง่าย ดังนั้นจึงขออธิบายรายละเอียดเฉพาะของเวอร์ชัน 2 เท่านั้น

4.2 เซิร์ฟเวอร์ ICQ

เซิร์ฟเวอร์ ICQ นั้นเป็นตัวกลางที่ให้บริการต่าง ๆ สำหรับ ICQ ไคลเอนต์และก่อนการใช้งาน ICQ ไคลเอนต์ทุกครั้งต้องมีการล็อกอินไปยังเซิร์ฟเวอร์เสียก่อนเพื่อเป็นการรายงานให้เซิร์ฟเวอร์รับรู้และแจ้งต่อไปให้กับ ICQ ไคลเอนต์ตัวอื่น ๆ ได้รับรู้ต่อไป ซึ่งในการล็อกอินไปที่ ICQ เซิร์ฟเวอร์ต้องติดต่อไปที่ icq.mirabilis.com พอร์ต 4000 สำหรับหน้าที่ของเซิร์ฟเวอร์มีดังนี้

1. ให้ข้อมูลสำหรับการสื่อสารของผู้ใช้ ICQ อื่นตามที่ไคลเอนต์ต้องการ เช่น
   - หมายเลข IP
   - หมายเลขพอร์ต
   - ชื่อผู้ใช้งาน
   - สถานะของผู้ใช้ ICQ เช่น Online, Offline, Away เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ให้บริการค้นหาผู้ใช้ ICQ ที่กำหนด ซึ่งสามารถค้นหาได้โดยกำหนดข้อมูลส่วนใดส่วนหนึ่ง ดังนี้

- ค้นหาจากหมายเลขผู้ใช้ ICQ หรือ หมายเลข UIN
- ค้นหาจากชื่อเล่น, ชื่อจริง หรือ นามสกุล
- ค้นหาจากอีเมล์

3. ให้บริการสื่อสารข้อความผ่านทางเซิร์ฟเวอร์หรือเรียกว่า offline message เป็นบริการสำหรับส่งข้อความไปยัง ICQ ไคลเอนต์อื่นที่ไม่ได้ออนไลน์อยู่ในขณะนั้น หรือส่งข้อความไปยัง ICQ ไคลเอนต์ที่ออนไลน์อยู่แต่ไม่สามารถสร้างการเชื่อมต่อโดยตรงเพื่อส่งข้อความได้ ถ้าเป็นกรณีแรกหลังจากที่ ICQ ไคลเอนต์ปลายทางออนไลน์ทางเซิร์ฟเวอร์จะส่งข้อความไปให้ทันที

4. ให้บริการเปลี่ยนแปลงสถานะต่าง ๆ ของผู้ใช้ ICQ เมื่อผู้ใช้ ICQ เปลี่ยนแปลงสถานะไคลเอนต์จะส่งคำสั่งไปเปลี่ยนสถานะให้กับเซิร์ฟเวอร์รับรู้ และจากนั้นเซิร์ฟเวอร์จะส่งคำสั่งเปลี่ยนสถานะของผู้ใช้ ICQ ไปยังผู้ใช้อื่น ๆ ที่อยู่ในคอนแทคลิสต์

5. ให้บริการข้อมูลส่วนตัวต่าง ๆ ของผู้ใช้ที่ต้องการ เช่น ชื่อจริง, ชื่อเล่น, วันเกิด เป็นต้น

6. ให้บริการแก้ไขเพิ่มเติมข้อมูลส่วนตัว

7. ให้บริการลงทะเบียนหมายเลข ICQ ใหม่ สำหรับผู้ที่ยังไม่มีหมายเลข ICQ เป็นของตนเองก็สามารถลงทะเบียนกับเซิร์ฟเวอร์เพื่อขอหมายเลขของตนได้

8. ให้บริการยกเลิกการจดทะเบียนผู้ใช้ ICQ

4.3 การสื่อสารระหว่างไคลเอนต์กับเซิร์ฟเวอร์

ในส่วนนี้กล่าวถึงรายละเอียดในการสื่อสารของโปรแกรม ICQ ระหว่างไคลเอนต์กับเซิร์ฟเวอร์โดยจะแสดงให้เห็นว่าในการใช้บริการต่างๆ จากเซิร์ฟเวอร์ต้องส่งคำสั่งอะไรไปบ้างมีลำดับขั้นตอนการส่งอย่างไรและแสดงให้เห็นโครงสร้างของแพ็กเก็ตที่ใช้ติดต่อกับเซิร์ฟเวอร์ว่าเป็นอย่างไร

4.3.1 แพ็กเก็ตที่ใช้สื่อสารกับเซิร์ฟเวอร์

รูปแบบแพ็กเก็ตที่ไคลเอนต์ใช้สื่อสารกับเซิร์ฟเวอร์มี 2 แบบคือ แพ็กเก็ตที่ส่งจากไคลเอนต์ไปเซิร์ฟเวอร์และแพ็กเก็ตที่ส่งจากเซิร์ฟเวอร์มาไคลเอนต์ซึ่งสามารถแสดงได้ดังตารางที่ 4-1 และ 4-2

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
VERSION	2	เวอร์ชันของโพรโตคอลที่ใช้ เช่น เวอร์ชัน 2 จะมีค่าเป็น 02 00
COMMAND	2	คำสั่งต่างๆ เช่น LOGIN , ACK เป็นต้น
SEQ_NUM	2	ซีเควนนัมเบอร์ (sequence number) สำหรับแพ็กเก็ตนี้
UIN	4	หมายเลข UIN ของผู้ส่ง
PARAMETERS	ไม่แน่นอน	พารามิเตอร์ที่จำเป็น (ขึ้นอยู่กับคำสั่ง)

ตารางที่ 4-1 แสดงรูปแบบแพ็กเก็ตที่ส่งจากไคลเอนต์ไปเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
VERSION	2	เวอร์ชันของโพรโตคอลที่ใช้ เช่น เวอร์ชัน 2 จะมีค่าเป็น 02 00
COMMAND	2	คำสั่งต่างๆ เช่น LOGIN_REPLY เป็นต้น
SEQ_NUM	2	ซีเควนนัมเบอร์สำหรับแพ็กเก็ตนี้
PARAMETERS	ไม่แน่นอน	พารามิเตอร์ที่จำเป็น (ขึ้นอยู่กับคำสั่ง)

ตารางที่ 4-2 แสดงรูปแบบแพ็กเก็ตที่ส่งจากเซิร์ฟเวอร์ไปไคลเอนต์

4.3.2 ข้อกำหนดในการสื่อสาร

- ทุกคำสั่งที่ส่งไม่ว่าจากไคลเอนต์หรือเซิร์ฟเวอร์ต้องมีการตอบรับกลับมาด้วยคำสั่ง ACK (ยกเว้นคำสั่ง ACK เอง) ภายใน 10 วินาที ถ้า ไม่ได้รับต้องส่งคำสั่งนั้นไปใหม่ และจะมีการส่งซ้ำไม่เกิน 6 ครั้งถ้ายังไม่มีการ ACK มาจะถือว่าการเชื่อมต่อมีปัญหา

- แต่ละคำสั่งที่ส่งไปต้องมีซีเควนนัมเบอร์ โดยซีเควนนัมเบอร์จะใช้แยกกันระหว่างฝั่งไคลเอนต์กับฝั่งเซิร์ฟเวอร์ ในการส่งแต่ละคำสั่งต้องเพิ่มซีเควนนัมเบอร์ขึ้นไปทีละ 1 ยกเว้นคำสั่ง ACK จะมีซีเควนนัมเบอร์เท่ากับคำสั่งที่รับมา และคำสั่งที่ส่งซ้ำจะใช้ซีเควนนัมเบอร์เดิมกับที่ส่งครั้งแรก

- ก่อนที่จะสื่อสารกับเซิร์ฟเวอร์ด้วยคำสั่งใดก็ตามต้องล็อกอินก่อนโดยใช้คำสั่งล็อกอินซึ่งขั้นตอนในการล็อกอินจะกล่าวในภายหลัง

- ในขณะที่ออนไลน์หรือล็อกอินเรียบร้อยแล้วต้องส่งคำสั่ง KeepAlive ทุกๆ 4 นาทีเพื่อเป็นการยืนยันกับเซิร์ฟเวอร์ว่ายังออนไลน์อยู่

4.3.3 การล็อกอิน

ก่อนที่การสื่อสารระหว่างไคลเอนต์หรือผู้ใช้ใดๆ จะเกิดขึ้น แต่ละไคลเอนต์ต้องล็อกอินไปยัง ICQ เซิร์ฟเวอร์ก่อนเป็นอันดับแรก เสมือนกับเป็นการรายงานตัวกับเซิร์ฟเวอร์ว่าต้องการใช้บริการของ ICQ และเพื่อให้เซิร์ฟเวอร์แจ้งไปยังไคลเอนต์อื่นที่มีไคลเอนต์นั้นอยู่ในคอนแทคลิสต์ทราบว่าไคลเอนต์นั้นออนไลน์แล้วและสามารถสื่อสารกันได้แล้ว โดยขั้นตอนในการล็อกอินแสดงดังรูปที่ 4-1 และ 4-2 ซึ่งมีรายละเอียดดังนี้

1. ส่งแพ็กเก็ตของคำสั่งล็อกอินไปยังเซิร์ฟเวอร์ ซึ่งเป็นการส่งข้อมูลของไคลเอนต์อันได้แก่ หมายเลข UIN, พาสเวิร์ด, หมายเลข IP, หมายเลขพอร์ต

2. เมื่อเซิร์ฟเวอร์ได้รับแพ็กเก็ตล็อกอินแล้วจะตรวจสอบข้อมูลว่าถูกต้องหรือไม่ เช่น มีหมายเลข UIN นั้นหรือไม่ หรือพาสเวิร์ดถูกต้องหรือไม่ ถ้าถูกต้องเซิร์ฟเวอร์จะส่งสัญญาณตอบกลับมา (ACK) พร้อมทั้งส่งข้อมูลของไคลเอนต์นั้นกลับมา ถ้าหากไม่ถูกต้องเซิร์ฟเวอร์จะส่งข้อความมาแจ้งและไคลเอนต์จะยกเลิกการติดต่อ

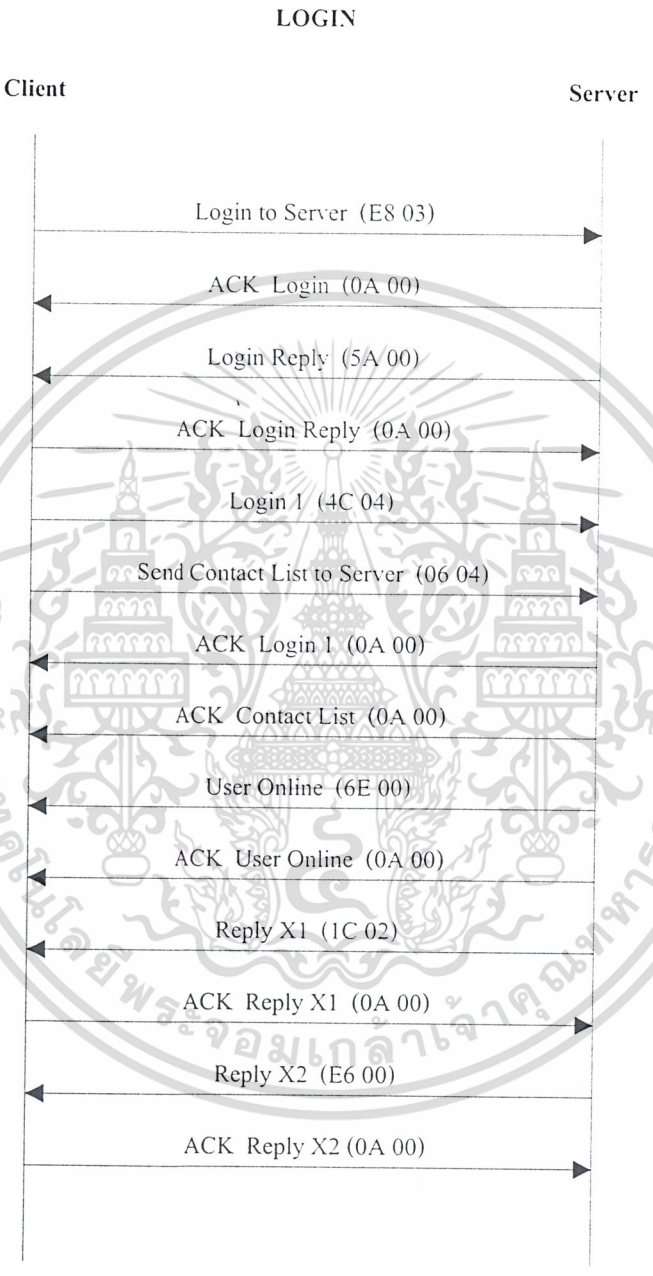
3. เมื่อไคลเอนต์ได้รับการตอบกลับจากเซิร์ฟเวอร์ ไคลเอนต์จะส่งคอนแทคลิสต์ซึ่งเป็นหมายเลข UIN ต่าง ๆ ไปยังเซิร์ฟเวอร์

4. เมื่อเซิร์ฟเวอร์ได้รับคอนแทคลิสต์จากไคลเอนต์ เซิร์ฟเวอร์จะตรวจสอบไคลเอนต์ในคอนแทคลิสต์ถ้าพบว่ามีไคลเอนต์คนใดที่ออนไลน์อยู่จะส่งข้อมูลของไคลเอนต์นั้นกลับมายังไคลเอนต์ที่ทำการล็อกอิน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

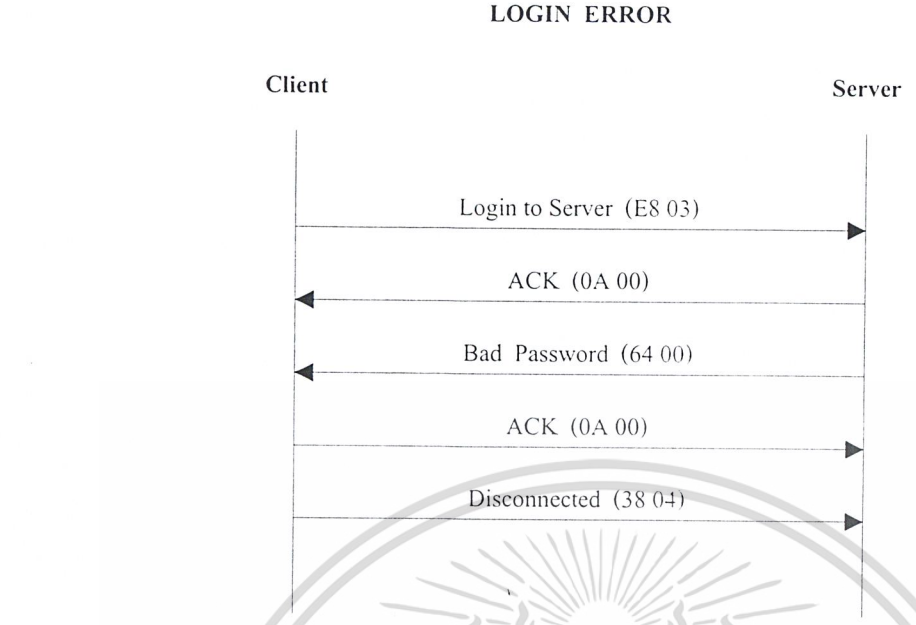
ซึ่งข้อมูลที่เซิร์ฟเวอร์ส่งมานั้นประกอบด้วยหมายเลข UIN, หมายเลข IP, พอร์ตของ TCP, สถานะของผู้ใช้คนอื่นๆ ในคอนแทคลิสต์

5. เมื่อไคลเอนต์ได้รับแพ็กเก็ตที่เซิร์ฟเวอร์ส่งมาให้จะส่งสัญญาณตอบรับกลับไป



รูปที่ 4-1 แสดงขั้นตอนในการล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



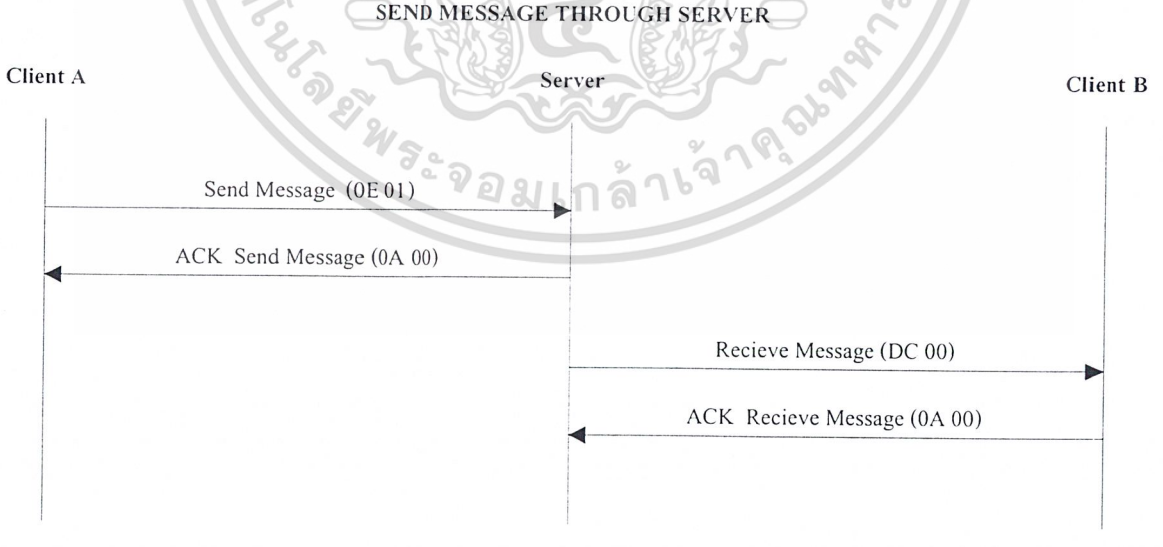
รูปที่ 4-2 แสดงกรณีที่พาสเวิร์ดผิด

4.3.4 การส่งข้อความ

การส่งข้อความไปยังผู้ใช้อื่นๆโดยผ่านเซิร์ฟเวอร์แสดงดังรูปที่ 4-3 และมีรายละเอียดดังนี้

1. ไคลเอนต์ส่งคำสั่ง SEND_MESSAGE ซึ่งประกอบด้วยหมายเลข UIN ของผู้ส่งและผู้รับ, ข้อความ, ชนิดของข้อความไปยังเซิร์ฟเวอร์และเซิร์ฟเวอร์ส่งคำสั่ง ACK กลับมา

2. เมื่อผู้รับปลายทางออนไลน์เซิร์ฟเวอร์ก็จะส่งข้อความนั้นไปยังผู้รับปลายทาง



รูปที่ 4-3 แสดงขั้นตอนการส่งข้อความผ่านเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

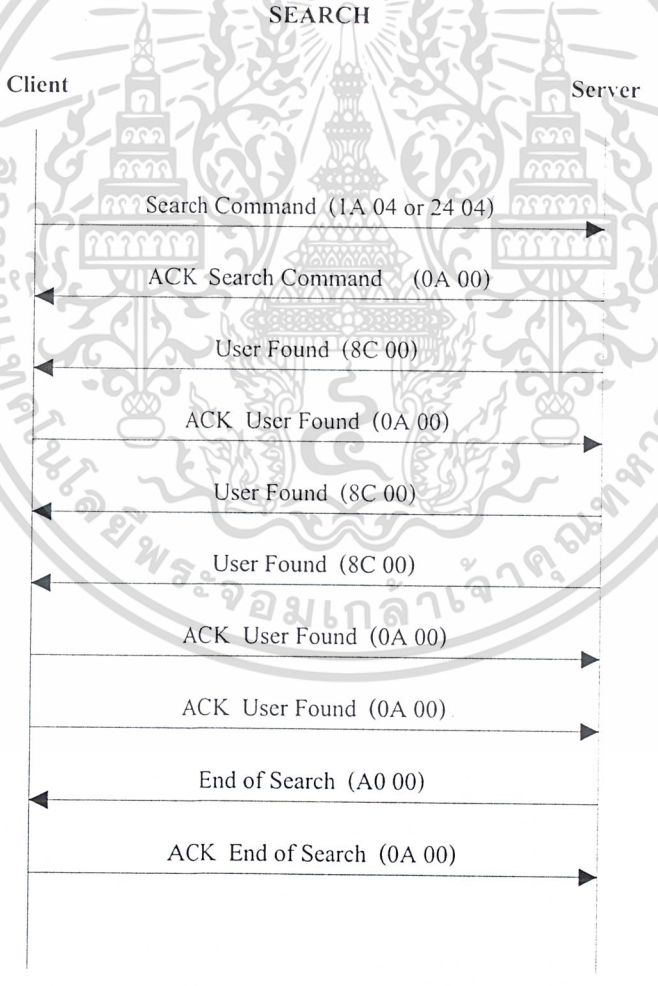
4.3.5 การค้นหาผู้ใช้คนอื่น

ในกรณีที่ไคลเอนต์ต้องการสื่อสารกับไคลเอนต์อื่นมีความจำเป็นที่ไคลเอนต์นั้นต้องทำการค้นหาและเพิ่มผู้ใช้ที่ต้องการสื่อสารด้วยเข้าไปในคอนแทคลิสต์ก่อน ซึ่งขั้นตอนในการค้นหาผู้ใช้แสดงดังรูปที่ 4-4 และมีรายละเอียดดังนี้

1. ไคลเอนต์ที่ต้องการค้นหาผู้ใช้ส่งแพ็กเก็ตของคำสั่ง SEARCH ไปยังเซิร์ฟเวอร์ซึ่งจะมีข้อมูลที่ใช้ประกอบการค้นหาโดยอาจเป็นชื่อเล่น, ชื่อจริง, นามสกุล, อีเมล์ หรือหมายเลข UIN ก็ได้

2. เมื่อเซิร์ฟเวอร์ได้รับแพ็กเก็ตดังกล่าวเซิร์ฟเวอร์จะส่งคำสั่ง ACK กลับมาพร้อมทั้งส่งหมายเลข UIN, ชื่อจริง, นามสกุล, ชื่อเล่น, อีเมล์และ Authorize ของผู้ใช้ที่มีข้อมูลตรงกับที่ไคลเอนต์ต้องการมายังไคลเอนต์นั้นเรื่อย ๆ

3. เมื่อเซิร์ฟเวอร์ไม่มีข้อมูลหรือครบ 40 คนแล้วเซิร์ฟเวอร์จะส่งแพ็กเก็ตของคำสั่ง End_of_Search มายังไคลเอนต์พร้อมทั้งแจ้งแก่ไคลเอนต์ว่าผู้ใช้ที่ค้นหาพบมีมากกว่า 40 คนหรือไม่



รูปที่ 4-4 แสดงขั้นตอนการค้นหาผู้ใช้คนอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

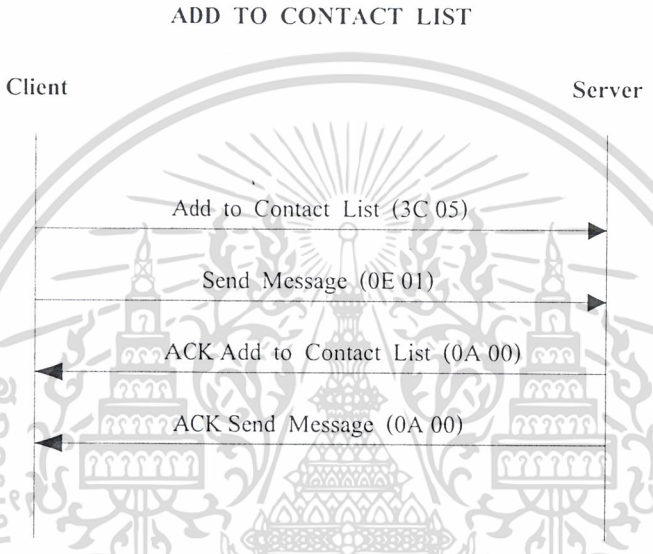
4.3.6 การเพิ่มชื่อผู้ใช้อื่นเข้าสู่คอนแทคลิสต์

ขั้นตอนการเพิ่มผู้ใช้เข้าในคอนแทคลิสต์แสดงดังรูปที่ 4-5 และมีรายละเอียดดังนี้

1. ไคลเอนต์ส่งแพ็กเก็ตของคำสั่ง ADD_TO_CONTACT_LIST ซึ่งจะส่งหมายเลข UIN ของผู้ใช้คนที่ไคลเอนต์ต้องการไปยังเซิร์ฟเวอร์

2. ไคลเอนต์ส่งแพ็กเก็ตที่ประกอบด้วยข้อความที่บอกข้อมูลของตัวไคลเอนต์เองไปยังเซิร์ฟเวอร์เพื่อให้เซิร์ฟเวอร์ส่งไปบอกผู้ที่ถูกเราเพิ่มชื่อ

3. เซิร์ฟเวอร์จะตรวจสอบสถานะของผู้ใช้ที่ถูกเพิ่มชื่อแล้วส่งสถานะกลับไปยังไคลเอนต์นั้น



รูปที่ 4-5 แสดงขั้นตอนการเพิ่มผู้ใช้อื่นเข้าคอนแทคลิสต์

4.3.7 การขอข้อมูลผู้ใช้จากเซิร์ฟเวอร์

กรณีที่ไคลเอนต์ต้องการข้อมูลของตนเองหรือของผู้ใช้อื่น ๆ แสดงดังรูปที่ 4-6 และมีรายละเอียดดังนี้

1. ไคลเอนต์ส่งแพ็กเก็ตของคำสั่ง INFO_REQUEST ไปยังเซิร์ฟเวอร์โดยมีหมายเลข UIN ของผู้ใช้ที่ต้องการข้อมูล

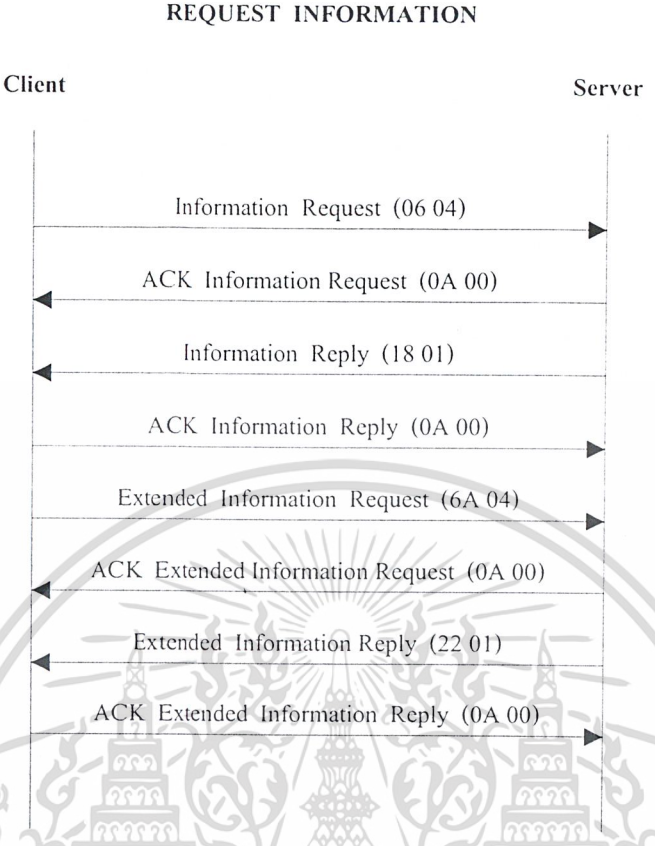
2. เซิร์ฟเวอร์จะส่งคำสั่ง ACK กลับมาพร้อมทั้งส่งข้อมูลพื้นฐานของผู้ใช้นั้นมาให้กับไคลเอนต์ซึ่งได้แก่หมายเลข UIN, ชื่อเล่น, ชื่อจริง, นามสกุล, อีเมล์ และ Authorize

3. ไคลเอนต์จะ ACK กลับไปพร้อมทั้งส่งแพ็กเก็ตของคำสั่ง EXTENDED_INFO_REQUEST เพื่อขอข้อมูลเพิ่มเติม

4. เมื่อเซิร์ฟเวอร์ได้รับแพ็กเก็ตนั้นก็จะส่งคำสั่ง ACK กลับมาพร้อมทั้งส่งหมายเลข UIN, ชื่อเมือง, ชื่อประเทศ, ชื่อรัฐ, อายุ, เพศ, เบอร์โทรศัพท์, โฮมเพจ, คำบรรยายเพิ่มเติมของผู้ใช้ (About) มาให้ไคลเอนต์

5. เมื่อไคลเอนต์ได้รับข้อมูลดังกล่าวจะส่งคำสั่ง ACK กลับไปเพื่อบอกว่าได้รับเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-6 แสดงขั้นตอนการขอข้อมูลเกี่ยวกับผู้ใช้จากเซิร์ฟเวอร์

4.3.8 การขออนุญาตเพิ่มชื่อผู้ใช้อื่นเข้าคอนแทคลิสต์

ในบางกรณีที่ผู้ใช้บางคนมีการกำหนด Authorize ทำให้ไคลเอนต์ที่ต้องการเพิ่มผู้ใช้คนนั้นเข้าในคอนแทคลิสต์ ต้องขออนุญาตจากผู้ใช้นั้นเสียก่อนแสดงดังรูปที่ 4-7 และมีรายละเอียดดังนี้

1. ไคลเอนต์ส่งแพ็กเก็ตของคำสั่ง MESSAGE_TO_NEW_USER ไปยังเซิร์ฟเวอร์เพื่อขออนุญาตเพิ่มชื่อเข้าคอนแทคลิสต์ โดยมีข้อมูลพื้นฐานของไคลเอนต์นั้น, หมายเลข UIN ทั้งของไคลเอนต์และผู้ใช้ที่ต้องการเพิ่มเข้าคอนแทคลิสต์ และข้อความที่ใช้ขอ Authorize

2. เซิร์ฟเวอร์จะส่งคำสั่ง ACK กลับมาพร้อมทั้งบอกให้ไคลเอนต์คอยรับการให้อนุญาตจากผู้ใช้นั้น

4.3.9 การเพิ่มรายชื่อผู้ใช้งาน

ก่อนที่จะใช้โปรแกรม ICQ สื่อสารกับผู้อื่นได้เราต้องมีชื่อของเราอยู่ในรายชื่อผู้ใช้งานของ ICQ ไคลเอนต์นั้นก่อนโดยขั้นตอนการเพิ่มชื่อแสดงดังรูปที่ 4-8 และมีรายละเอียดดังนี้

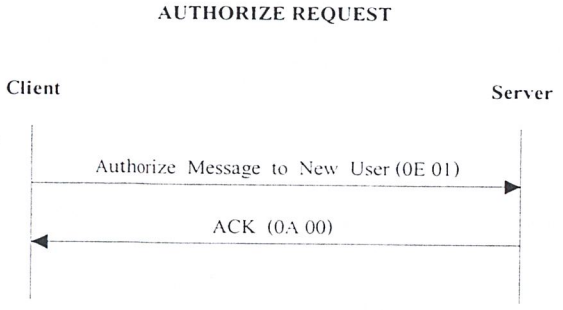
1. หาก ICQ ไคลเคนต์ออนไลน์อยู่ต้องตัดการเชื่อมต่อก่อนโดยส่งคำสั่งดิสคอนเน็ก (disconnect)

2. จากนั้นจึงล็อกอินไปที่เซิร์ฟเวอร์โดยใช้หมายเลข UIN และพาสเวิร์ดของผู้ใช้ที่ต้องการใช้ ICQ ไคลเอนต์นั้น

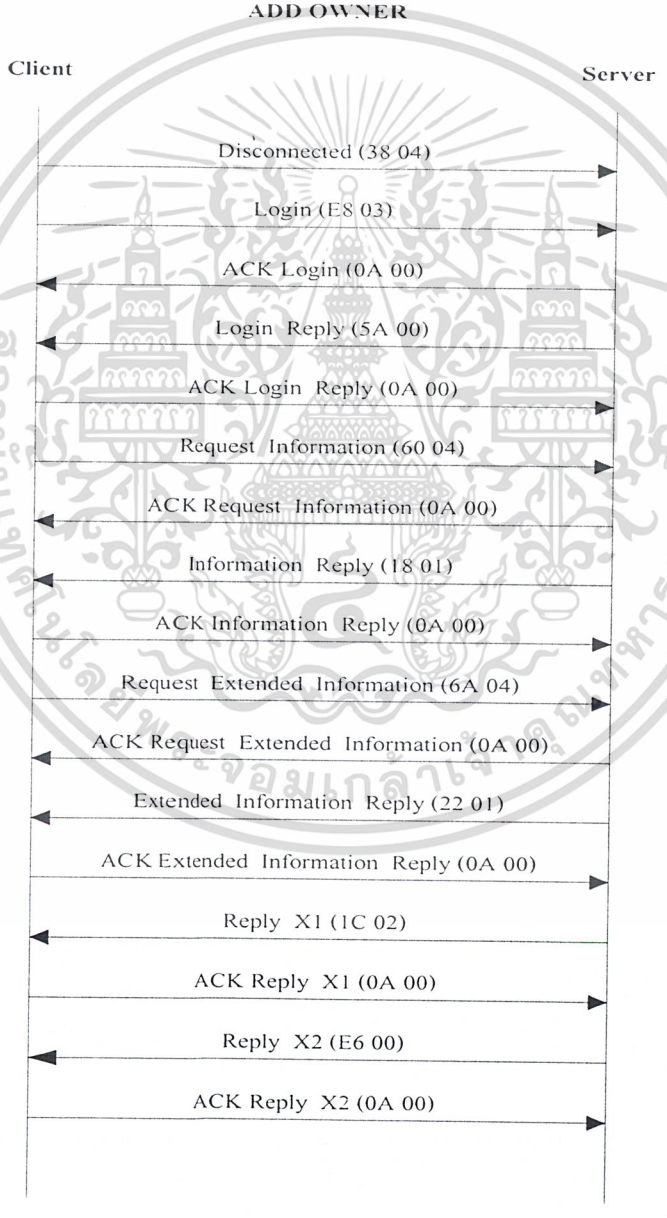
3. ถ้าสามารถล็อกอินได้ก็ส่งคำสั่ง INFO_REQUEST และ EXTENDED_INFO_REQUEST ไปยังเซิร์ฟเวอร์เพื่อขอข้อมูลส่วนตัวของผู้ใช้

4. เมื่อได้ข้อมูลมาก็ถือว่าการเพิ่มชื่อผู้ใช้งานเสร็จสิ้นและเข้าสู่สถานะออนไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 แสดงขั้นตอนการขอ Authorize



รูปที่ 4-8 แสดงขั้นตอนการเพิ่มผู้ใช้งาน

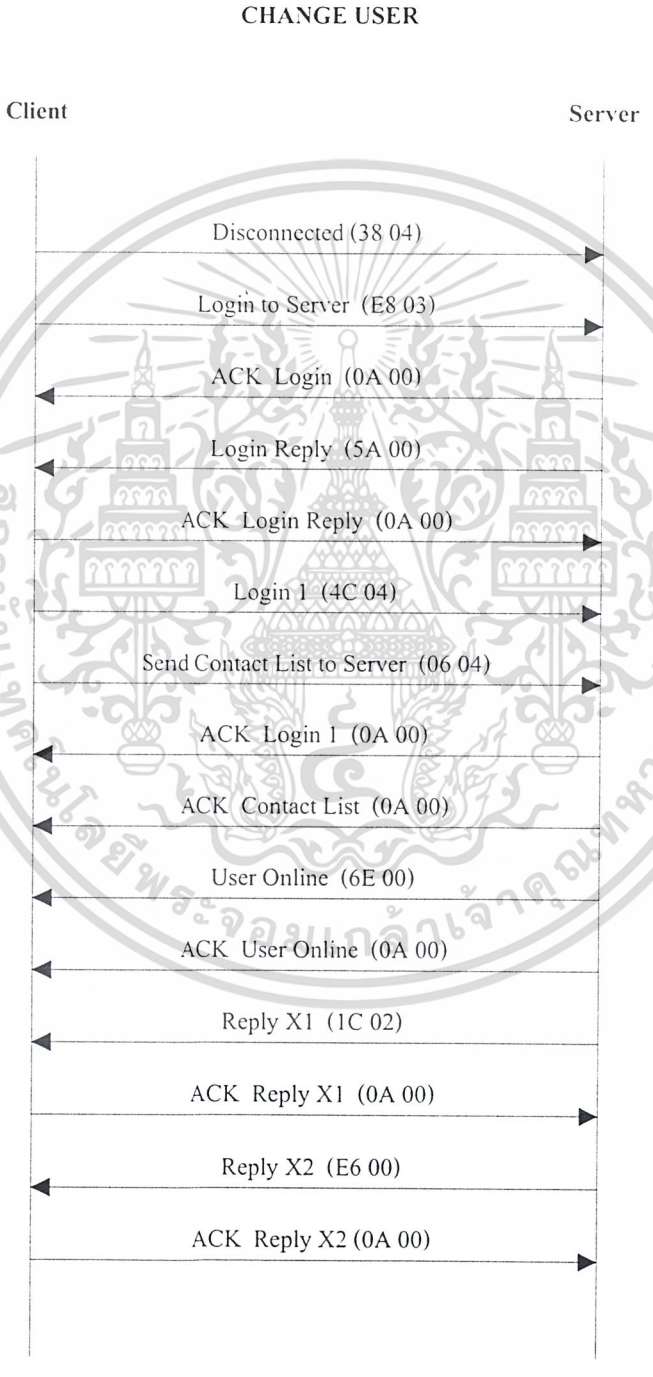
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.10 การเปลี่ยนผู้ใช้งาน

ขั้นตอนในการเปลี่ยนผู้ใช้แสดงดังรูปที่ 4-9 และมีขั้นตอนดังนี้

1. หากออนไลน์อยู่ต้องยกเลิกการติดต่อเสียก่อนจากนั้นจึงล็อกอินไปยังเซิร์ฟเวอร์โดยใช้หมายเลข UIN และพาสเวิร์ดของผู้ใช้ที่ต้องการ

2. ส่วนขั้นตอนอื่นจะเหมือนกับการล็อกอินตามปรกติ



รูปที่ 4-9 แสดงขั้นตอนการเปลี่ยนผู้ใช้งาน

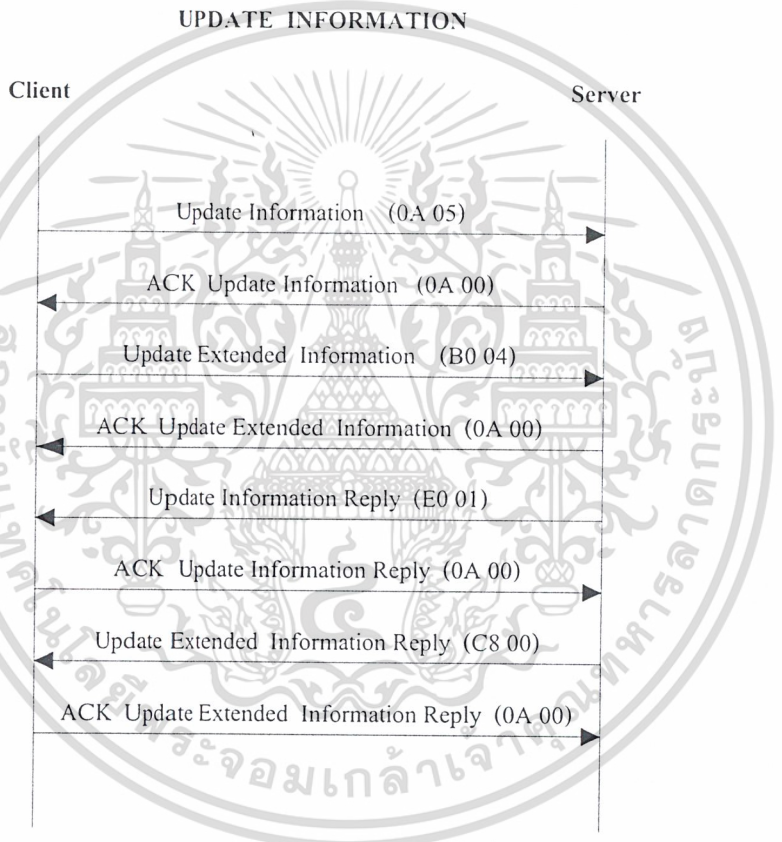
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.11 การแก้ไขข้อมูลส่วนตัว

ผู้ใช้โปรแกรม ICQ ทุกคนสามารถแก้ไขข้อมูลส่วนตัวของตนได้โดยข้อมูลส่วนตัวจะเก็บอยู่ที่เซิร์ฟเวอร์ซึ่งขั้นตอนในการแก้ไขข้อมูลส่วนตัวแสดงดังรูปที่ 4-10 และมีรายละเอียดดังนี้

1. ส่งคำสั่ง Update Information และคำสั่ง Update Extended Information ซึ่งพารามิเตอร์ของสองคำสั่งนี้ก็จะประกอบไปด้วยข้อมูลส่วนตัวต่าง ๆ ของผู้ใช้งาน

2. เซิร์ฟเวอร์จะตอบกลับมาด้วยคำสั่ง Update Information Reply และคำสั่ง Update Extended Information Reply หากการแก้ไขข้อมูลสำเร็จ ถ้าเซิร์ฟเวอร์ตอบกลับมาด้วยคำสั่งอื่นแสดงว่าการแก้ไขข้อมูลมีปัญหา



รูปที่ 4-10 แสดงขั้นตอนการแก้ไขข้อมูลส่วนตัว

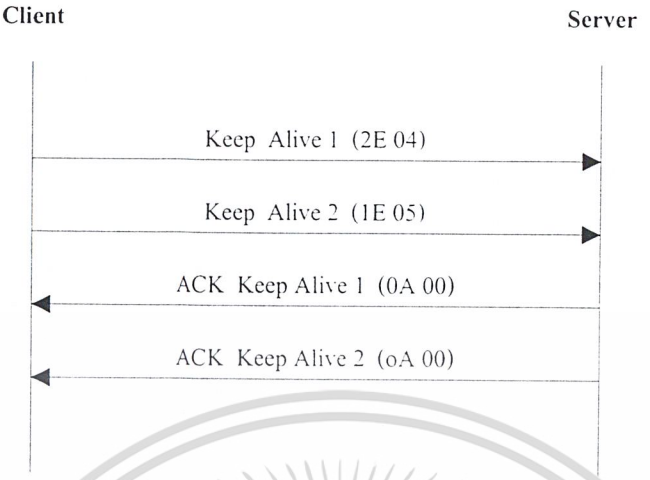
4.3.12 การส่ง KeepAlive

ในระหว่างที่ไคลเอนต์ออนไลน์อยู่จะต้องมีการติดต่อกับเซิร์ฟเวอร์ทุกระยะ (ทุก ๆ 4 นาที) เพื่อให้แน่ใจว่าไคลเอนต์ยังคงติดต่อกับเซิร์ฟเวอร์อยู่ และเพื่อให้เซิร์ฟเวอร์มั่นใจได้ว่าไคลเอนต์ยังออนไลน์อยู่ ขั้นตอนการสื่อสารแสดงดังรูปที่ 4-11 และมีรายละเอียดดังนี้

1. การส่งคำสั่ง Keep_Alive ไปยังเซิร์ฟเวอร์ต้องมีการส่ง Keep Alive 1 และ Keep Alive 2 ไปยังเซิร์ฟเวอร์

2. เมื่อเซิร์ฟเวอร์ได้รับจะส่งคำสั่ง ACK กลับมายังไคลเอนต์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 แสดงขั้นตอนการส่ง Keep Alive

4.4 การสื่อสารระหว่างไคลเอนต์กับไคลเอนต์

การสื่อสารระหว่างไคลเอนต์กับไคลเอนต์ใช้เมื่อมีการส่งข้อความโดยตรง, การส่งไฟล์ หรือการแชต โดยใช้โพรโตคอล TCP ในการสื่อสาร ซึ่งการสื่อสารกันโดยตรงระหว่างไคลเอนต์จะเกิดขึ้นได้ก็ต่อเมื่อมีการเชื่อมต่อกันก่อน โดยต้องมีไคลเอนต์ตัวใดตัวหนึ่งคอนเน็กไปยัง IP และพอร์ตที่ไคลเอนต์อีกตัวหนึ่งเปิดรออยู่ สำหรับข้อมูลเกี่ยวกับ IP และพอร์ตของแต่ละไคลเอนต์นั้นจะได้มาในขณะที่เซิร์ฟเวอร์ส่งคำสั่ง User Online ของไคลเอนต์นั้นมาให้

4.4.1 รูปแบบของแพ็กเก็ตในการสื่อสารกันโดยตรง

การรับส่งข้อความมีรูปแบบของแพ็กเก็ตที่ต้องใช้ในการสื่อสารดังนี้

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
LENGTH	2	ความยาวของแพ็กเก็ตนี้โดยไม่รวมฟิลด์นี้ (ค่าปกติเป็น 26)
IDENT	1	ปกติมีค่าเป็น 0xff
VERSION	2	เวอร์ชันของ TCP โพรโตคอล
REVISION	2	ปกติมีค่าเป็น 0
TCP_MSG_PORT	4	พอร์ตสำหรับรอรับข้อความ
UIN	4	หมายเลข ICQ ผู้ส่งข้อความ
TCP_IP	4	IP แอดเดรสหากอยู่หลังไฟร์วอลล์
TCP_REAL_IP	4	IP แอดเดรสจริง
TCP_FLAG	1	ปกติมีค่าเป็น 0x04
TCP_FILE_PORT	4	พอร์ตสำหรับรอรับแชตและไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-3 แสดงข้อมูลภายในแพ็กเก็ต CHANNEL_INIT

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
LENGTH	2	ความยาวของแพ็กเก็ตนี้โดยไม่รวมฟิลด์นี้
UIN	4	หมายเลข ICQ ผู้ส่งข้อความ
VERSION	2	เวอร์ชันของ TCP โพรโตคอล
COMMAND	2	คำสั่งกำหนดชนิดข้อความ
X1	2	ปกติมีค่าเป็น 0
UIN	4	หมายเลข ICQ ผู้ส่งข้อความ
TYPE	2	ชนิดของข้อความ
MSG_LEN	2	ความยาวของ MESSAGE เป็นไบต์
MESSAGE	ไม่แน่นอน	ข้อความซึ่งมีความยาวเท่ากับ MSG_LEN
TCP_IP	4	IP แอดเดรสหากอยู่หลังไฟร์วอลล์
TCP_REAL_IP	4	IP แอดเดรสจริง
TCP_MSG_PORT	4	หมายเลขพอร์ตสำหรับรอรับข้อความ
TCP_FLAG	1	ปกติมีค่าเท่ากับ 4
TCP_STATUS	2	สถานะของไคลเอนต์
MSG_COMMAND	2	คำสั่งเกี่ยวกับการรับส่งข้อความ
SEQ_NUMBER	2	ซีเควนนัมเบอร์ เริ่มจาก 0xfffe แล้วลดลงทีละ 1

ตารางที่ 4-4 แสดงข้อมูลภายในแพ็กเก็ต CHANNEL_MESSAGE

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
FILE_REV_PORT	2	ปกติมีค่าเป็น 0
FILE_X3	2	ปกติมีค่าเป็น 0
FILE_NAME_LEN	2	ความยาวของ FILE_NAME เป็นไบต์
FILE_NAME	ไม่แน่นอน	ชื่อไฟล์ที่ส่งถ้ามีหลายไฟล์จะแสดงเป็น “n files” เช่น “3 files”
FILE_SIZE	4	ขนาดของไฟล์รวมเป็นไบต์
FILE_PORT	4	ปกติมีค่าเป็น 0

ตารางที่ 4-5 แสดงข้อมูลเพิ่มเติมของแพ็กเก็ต CHANNEL_MESSAGE สำหรับการขอส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
LENGTH	2	ความยาวของแพ็กเก็ตนี้โดยไม่รวมฟิลด์นี้
UIN	4	หมายเลข ICQ ผู้ส่งข้อความ
VERSION	2	เวอร์ชันของ TCP โพรโตคอล
COMMAND	2	ปกติมีค่าเป็น 0xDA07
X1	2	ปกติมีค่าเป็น 0
UIN	4	หมายเลข ICQ ผู้ส่งข้อความ
TYPE	2	ชนิดของข้อความ
MSG_LEN	2	ความยาวของ MESSAGE เป็นไบต์
MESSAGE	ไม่แน่นอน	ข้อความซึ่งมีความยาวเท่ากับ MSG_LEN
TCP_IP	4	IP แอดเดรสหากอยู่หลังไฟร์วอลล์
TCP_REAL_IP	4	IP แอดเดรสจริง
TCP_MSG_PORT	4	หมายเลขพอร์ตสำหรับรอรับข้อความ
TCP_FLAG	1	ปกติมีค่าเท่ากับ 4
TCP_STATUS	2	สถานะของไคลเอนต์
MSG_COMMAND	2	คำสั่งเกี่ยวกับการรับส่งข้อความ
SEQ_NUMBER	2	ซีเควนนัมเบอร์ มีค่าเท่ากับแพ็กเก็ตที่รับมา

ตารางที่ 4-6 แสดงข้อมูลภายในแพ็กเก็ต ACK

ชื่อฟิลด์	ความยาว (ไบต์)	คำจำกัดความ
FILE_REV_PORT	2	พอร์ตที่รอรับการส่งไฟล์ (network order)
FILE_X3	2	ปกติมีค่าเป็น 0
FILE_NAME_LEN	2	ปกติมีค่าเป็น 1
FILE_NAME	ไม่แน่นอน	ปกติมีค่าเป็น 0
FILE_SIZE	4	ปกติมีค่าเป็น 0
FILE_PORT	4	พอร์ตที่รอรับการส่งไฟล์ (intel order)

ตารางที่ 4-7 แสดงข้อมูลเพิ่มเติมของแพ็กเก็ต ACK สำหรับการตอบรับการขอส่งไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปร	ค่าตัวแปร	คำจำกัดความ
TCP_CANCEL	0xD007	ยกเลิกการติดต่อ
TCP_ACK	0xDA07	ตอบกลับ
TCP_MESSAGE	0xEE07	ส่งข้อความ

ตารางที่ 4-8 แสดงตัวแปรสำหรับฟิลด์ COMMAND ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK

ตัวแปร	ค่าตัวแปร	คำจำกัดความ
MSG_MSG	0x0100	ส่งข้อความปกติ
MSG_CHAT	0x0200	เริ่มต้นสื่อสารแบบ Chat
MSG_FILE	0x0300	เริ่มต้นส่งไฟล์
MSG_URL	0x0400	ส่ง URL address
MSG_REQ_AUTH	0x0600	ขอสิทธิในการรับรู้สถานะของผู้อื่น
MSG_DENY_AUTH	0x0700	ไม่ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น
MSG_GIVE_AUTH	0x0800	ให้สิทธิในการรับรู้สถานะตนเองกับผู้อื่น
MSG_ADDED	0x0C00	สัญญาณอัตโนมัติแสดงว่าตนเองถูกผู้อื่นเพิ่มในคอนแทคลิสต์
MSG_READAWAY	0xE803	ได้รับข้อความจากสถานะ Away
MSG_READOCCUPIE	0xE903	ได้รับข้อความจากสถานะ Occupied
MSG_READNA	0xEA03	ได้รับข้อความจากสถานะ Not Available
MSG_READDND	0xEB03	ได้รับข้อความจากสถานะ Do Not Disturb
MSG_READFFC	0xEC03	ได้รับข้อความจากสถานะ Free For Chat
MSGF_MASS	0x0008	ส่งข้อความมากกว่าหนึ่งคนในครั้งเดียว

ตารางที่ 4-9 แสดงตัวแปรสำหรับฟิลด์ TYPE ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK

ตัวแปร	ค่าตัวแปร
TCP_STAT_ONLINE	0x0000
TCP_STAT_REFUSE	0x0100
TCP_STAT_AWAY	0x0400
TCP_STAT_OCCUPIED	0x0900
TCP_STAT_DND	0x0A00
TCP_STAT_NA	0x0E00

ตารางที่ 4-10 แสดงตัวแปรสำหรับฟิลด์ TCP_STATUS ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปร	ค่าตัวแปร	คำจำกัดความ
TCP_MSG_AUTO	0x0000	กำหนดส่งข้อความกลับแบบอัตโนมัติ
TCP_MSG_REAL	0x1000	ส่งข้อความปกติ
TCP_MSG_LIST	0x2000	ส่งรายการคอนแทคลิสต์
TCP_MSG_URGENT	0x4000	ส่งข้อความแบบ Urgently
TCP_MSGF_S_INVISIBLE	0x8000	ผู้ส่งข้อความอยู่ในสถานะ Invisible
TCP_MSGF_S_AWAY	0x0001	ผู้ส่งข้อความอยู่ในสถานะ Away
TCP_MSGF_S_OCCUPIED	0x0002	ผู้ส่งข้อความอยู่ในสถานะ Occupied
TCP_MSGF_S_NA	0x0008	ผู้ส่งข้อความอยู่ในสถานะ N/A
TCP_MSGF_S_DND	0x0010	ผู้ส่งข้อความอยู่ในสถานะ DND

ตารางที่ 4-11 แสดงตัวแปรสำหรับฟิลด์ MSG_COMMAND ในแพ็กเก็ต CHANNEL_MESSAGE และ ACK

สำหรับการรับส่งไฟล์มีแพ็กเก็ตที่ใช้ในการสื่อสารดังนี้

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x00
4	ปกติมีค่าเป็น 0x00000000
4	จำนวนไฟล์ทั้งหมดที่จะส่ง
4	ขนาดของไฟล์ทั้งหมดที่จะส่ง (ไบต์)
4	ความเร็วในการส่ง 0x00 = หยุด 0x61 = ดีเลย์ 0.15 วินาที 0x62 = ดีเลย์ 0.10 วินาที 0x64 = ไม่มีดีเลย์
2	ความยาวชื่อเล่นของผู้ส่ง
ไม่แน่นอน	ชื่อเล่นผู้ส่ง

ตารางที่ 4-12 แสดงแพ็กเก็ต 0x00

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x01
4	ความเร็วในการรับ
2	ความยาวของชื่อเล่นผู้รับ
ไม่แน่นอน	ชื่อเล่นผู้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-13 แสดงแพ็กเก็ต 0x01

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x02
1	ปกติมีค่าเป็น 0
2	ความยาวของชื่อไฟล์ที่จะส่ง
ไม่แน่นอน	ชื่อไฟล์ที่จะส่ง
2	ปกติมีค่าเป็น 1
ไม่แน่นอน	ปกติมีค่าเป็น 0
4	ขนาดของไฟล์ที่จะส่ง
4	ปกติมีค่าเป็น 0
4	ความเร็วในการส่ง

ตารางที่ 4-14 แสดงแพ็กเก็ต 0x02

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x03
4	ปกติมีค่าเป็น 0
4	ปกติมีค่าเป็น 0
4	ความเร็วในการรับ

ตารางที่ 4-15 แสดงแพ็กเก็ต 0x03

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x04
4	ไม่ระบุ

ตารางที่ 4-16 แสดงแพ็กเก็ต 0x04

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x05
4	ความเร็วใหม่ในการส่ง

ตารางที่ 4-17 แสดงแพ็กเก็ต 0x05

ความยาว (ไบต์)	คำจำกัดความ
1	ค่าบ่งบอกแพ็กเก็ต ปกติมีค่าเป็น 0x06
ไม่เกิน 2048	ข้อมูลของไฟล์ที่ส่ง

ตารางที่ 4-18 แสดงแพ็กเก็ต 0x06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การรับส่งข้อความโดยตรง

ในการรับส่งข้อความโดยตรงระหว่างไคลเอนต์นั้นมีแพ็กเก็ตที่ต้องใช้งานดังนี้

1. แพ็กเก็ต CHANNEL_INIT ใช้สำหรับส่งในครั้งแรกของการสื่อสารระหว่างไคลเอนต์ใดๆ ก่อนส่งแพ็กเก็ต CHANNEL_MESSAGE และครั้งต่อไปที่ส่ง CHANNEL_MESSAGE ไม่ต้องส่ง CHANNEL_INIT อีกแล้ว

2. แพ็กเก็ต CHANNEL_MESSAGE ใช้สำหรับส่งข้อความระหว่างไคลเอนต์โดยต้องมีการตั้งค่าฟิลด์ต่างๆ ดังนี้

- ฟิลด์ COMMAND ดูจากตารางที่ 4-8 ตั้งค่าเป็น TCP_MESSAGE
- ฟิลด์ TYPE ดูจากตารางที่ 4-9 ตั้งค่าเป็น MSG_MSG
- ฟิลด์ TCP_STATUS ดูจากตารางที่ 4-10 ตั้งค่าเป็น TCP_STAT_ONLINE หรือสถานะขณะนั้น
- ฟิลด์ MSG_COMMAND ดูจากตารางที่ 4-11 ตั้งค่าเป็น TCP_MSG_REAL

3. แพ็กเก็ต ACK ใช้สำหรับตอบกลับมาหลังจากได้รับข้อความโดยต้องมีการตั้งค่าฟิลด์ต่างๆ ดังนี้

- ฟิลด์ COMMAND ดูจากตารางที่ 4-8 ตั้งค่าเป็น TCP_ACK
- ฟิลด์ TYPE ดูจากตารางที่ 4-9 ตั้งค่าเป็น MSG_MSG
- ฟิลด์ TCP_STATUS ดูจากตารางที่ 4-10 ตั้งค่าเป็น TCP_STAT_ONLINE หรือสถานะขณะนั้น
- ฟิลด์ MSG_COMMAND ดูจากตารางที่ 4-11 ตั้งค่าเป็น TCP_MSG_REAL

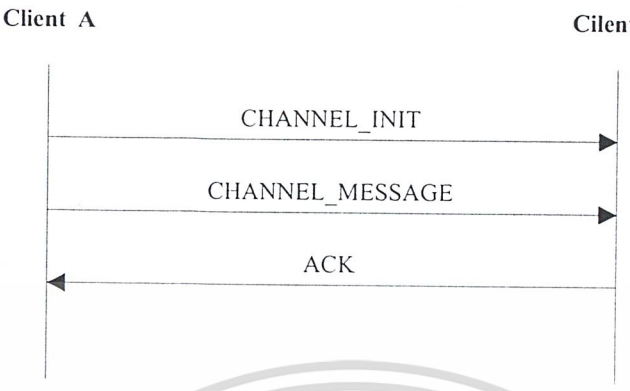
สำหรับขั้นตอนในการสื่อสารโดยตรงระหว่างไคลเอนต์แสดงดังรูปที่ 4-12 และมีรายละเอียดดังนี้ (สมมติให้ไคลเอนต์ A ต้องการส่งข้อความไปยังไคลเอนต์ B)

1. ไคลเอนต์ A ต้องทำการตรวจสอบว่ามีการเชื่อมต่อแบบ TCP กับไคลเอนต์ B หรือยัง ถ้ายังไคลเอนต์จะตรวจสอบหมายเลข IP และพอร์ตของไคลเอนต์ B จากข้อมูลที่ได้รับมาจากเซิร์ฟเวอร์ในขณะที่ไคลเอนต์ B ล็อกอินแล้วสร้างการเชื่อมต่อกับไคลเอนต์ B จากนั้นจึงส่งคำสั่ง CHANNEL_INIT ซึ่งประกอบด้วยหมายเลข UIN, หมายเลข IP, หมายเลขพอร์ตของ TCP ของไคลเอนต์ A ที่ใช้สำหรับการรับข้อความที่เข้ามาไปยังไคลเอนต์ B

2. เมื่อการเชื่อมต่อระหว่างไคลเอนต์ A และ B สำเร็จแล้ว ไคลเอนต์ A จะส่งแพ็กเก็ต CHANNEL_MESSAGE ซึ่งภายในมีหมายเลข UIN ของผู้ส่ง, เวอร์ชันของ ICQ, IP และพอร์ตของไคลเอนต์ A และข้อความพร้อมทั้งชนิดและความยาวของข้อความ

3. ในการส่งข้อความแบบนี้ทุกข้อความต้องมีการตอบรับโดยด้านรับด้วยคำสั่ง CHANNEL_ACK ซึ่งส่วนใหญ่จะไม่มีอะไรใน CHANNEL_ACK ยกเว้นกรณีที่ไคลเอนต์ B มีสถานะ Away/DND และกำหนดข้อความไว้จึงจะมีข้อมูลเพิ่มเติมตอบกลับมายังไคลเอนต์ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-12 แสดงขั้นตอนการส่งข้อความโดยตรงระหว่างไคลเอนต์

4.4.3 การรับส่งไฟล์

ในการรับส่งไฟล์ ICQ มีโพรโตคอลที่เรียกว่า File Direct Protocol สำหรับใช้ควบคุมในการรับส่งไฟล์ระหว่างไคลเอนต์ ซึ่งในการรับส่งไฟล์จะใช้พอร์ตในการรับส่งแยกจากพอร์ตที่ใช้ในการส่งข้อความ และการรับส่งไฟล์ระหว่างไคลเอนต์นั้นมีแพ็กเก็ตที่ต้องใช้งานดังนี้

1. แพ็กเก็ต CHANNEL_INIT หากยังไม่มีการส่งข้อความโดยตรงกันเลยระหว่างไคลเอนต์ต้องส่ง CHANNEL_INIT ไปก่อนเช่นกัน

2. แพ็กเก็ต CHANNEL_MESSAGE และข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ดูจากตารางที่ 4-5 โดยข้อมูลเพิ่มเติมจะแทรกอยู่ระหว่างฟิลด์ MSG_COMMAND และ SEQ_NUMBER แพ็กเก็ตนี้ใช้สำหรับส่งไปยังผู้รับไฟล์เพื่อแจ้งให้รู้ว่าต้องการจะส่งไฟล์พร้อมทั้งรายละเอียดของไฟล์ที่จะส่งโดยต้องมีการตั้งค่าฟิลด์ต่างๆ ดังนี้

- ฟิลด์ COMMAND ดูจากตารางที่ 4-8 ตั้งค่าเป็น TCP_MESSAGE
- ฟิลด์ TYPE ดูจากตารางที่ 4-9 ตั้งค่าเป็น MSG_FILE
- ฟิลด์ TCP_STATUS ดูจากตารางที่ 4-10 ตั้งค่าเป็น TCP_STAT_ONLINE หรือสถานะขณะนั้น
- ฟิลด์ MSG_COMMAND ดูจากตารางที่ 4-11 ตั้งค่าเป็น TCP_MSG_REAL

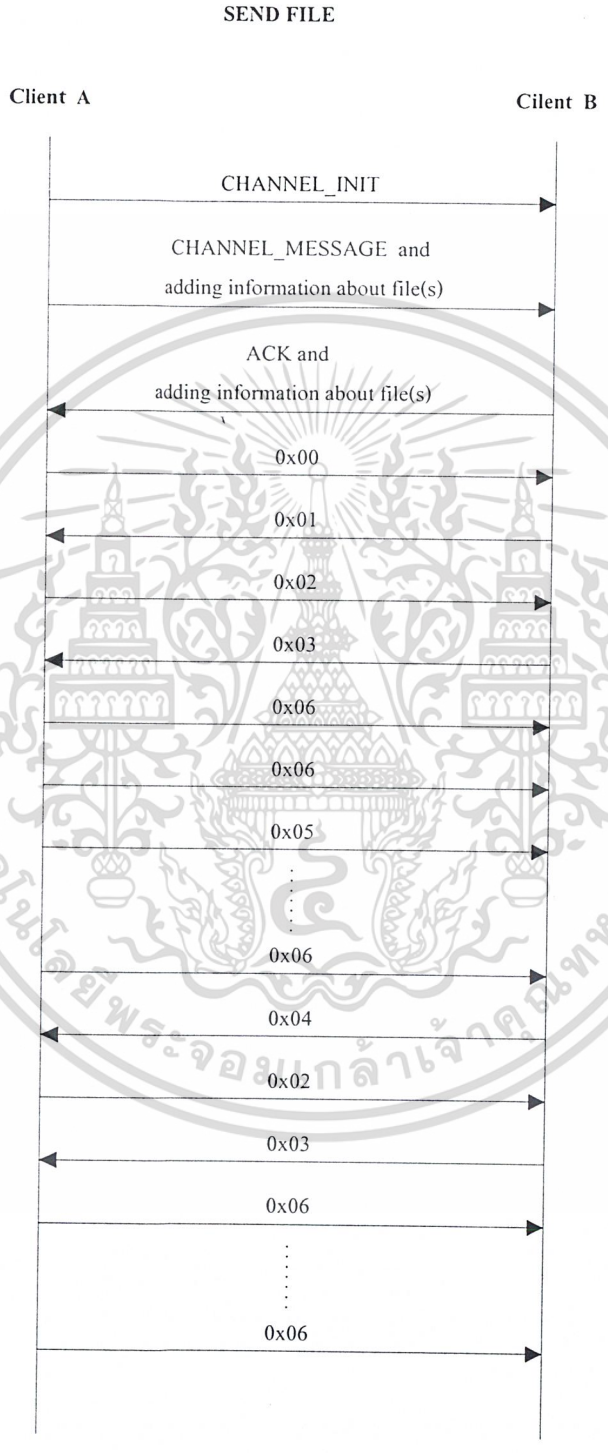
3. แพ็กเก็ต ACK และข้อมูลเพิ่มเติมเกี่ยวกับไฟล์ดูจากตารางที่ 4-7 โดยข้อมูลเพิ่มเติมจะแทรกอยู่ระหว่างฟิลด์ MSG_COMMAND และ SEQ_NUMBER ใช้สำหรับตอบกลับมาหลังจากได้รับแจ้งขอส่งไฟล์ โดยต้องมีการตั้งค่าฟิลด์ต่างๆดังนี้

- ฟิลด์ COMMAND ดูจากตารางที่ 4-8 ตั้งค่าเป็น TCP_ACK
- ฟิลด์ TYPE ดูจากตาราง 4-9 ตั้งค่าเป็น MSG_FILE
- ฟิลด์ TCP_STATUS ดูจากตารางที่ 4-10 ตั้งค่าเป็น TCP_STAT_ONLINE หรือสถานะขณะนั้น
- ฟิลด์ MSG_COMMAND ดูจากตารางที่ 4-11 ตั้งค่าเป็น TCP_MSG_REAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟิลด์ FILE_REV_PORT หมายเลขพอร์ตสำหรับรอรับไฟล์ถ้าไม่ต้องการรับไฟล์ให้ตั้งค่าเป็น 0

- ฟิลด์ FILE_PORT หมายเลขพอร์ตสำหรับรอรับไฟล์ถ้าไม่ต้องการรับไฟล์ให้ตั้งค่าเป็น 0



รูปที่ 4-13 แสดงขั้นตอนการรับส่งไฟล์

สำหรับขั้นตอนในการส่งไฟล์ระหว่างไคลเอนต์แสดงดังรูปที่ 4-13 และมีรายละเอียดดังนี้

1. ทำการสร้างการเชื่อมต่อแบบ TCP ระหว่างไคลเอนต์ A และ B
2. ไคลเอนต์ A ส่งแพ็กเก็ตของข้อความเพื่อบอกว่าต้องการจะส่งไฟล์ไปให้ไคลเอนต์ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อไคลเอนต์ A ได้รับ ACK จาก B ไคลเอนต์ A ก็จะตรวจสอบว่า B ต้องการรับไฟล์จาก A หรือไม่

4. ถ้าไคลเอนต์ B ไม่ต้องการรับไฟล์ก็จะไม่ทำอะไรต่อแต่ถ้าไคลเอนต์ B ต้องการรับไฟล์นั้น ไคลเอนต์ A จะเริ่มกระบวนการส่งไฟล์

5. 0x00 : ไคลเอนต์ A ส่งแพ็กเก็ตไปบอกไคลเอนต์ B ว่าไฟล์ที่จะส่งไปให้มีทั้งหมดกี่ไฟล์และมีขนาดรวมทั้งหมดกี่ไบต์ รวมถึงข้อมูลของผู้ส่ง (A) ได้แก่ ความเร็วในการส่ง, ชื่อเล่น

6. 0x01 : ไคลเอนต์ B จะส่งข้อมูลของผู้รับ (B) ให้กับไคลเอนต์ A เช่นเดียวกัน

7. 0x02 : ไคลเอนต์ A จะส่งชื่อไฟล์, ขนาดไฟล์, ความเร็วในการส่งไปยังไคลเอนต์ A เพื่อบอกว่ากำลังจะเริ่มส่งไฟล์ไหน

8. 0x03 : ไคลเอนต์ B จะส่งขนาดของไฟล์ที่ได้รับเพื่อใช้ในการตรวจสอบว่าได้รับไฟล์สมบูรณ์หรือไม่จะได้ทำการส่งให้ครบต่อจากที่ได้รับ

9. 0x06 : ระหว่างนี้ไคลเอนต์ A จะส่งไฟล์มาเรื่อย ๆ ครั้งละ 2048 ไบต์จนจบไฟล์นั้น

10. 0x05 : ถ้าหากฝ่ายใดมีการเปลี่ยนความเร็วก็จะต้องมีการบอกกับอีกฝ่ายหนึ่ง

11. 0x04 : อาจมีการส่งข้อความเพื่อให้หยุดส่งไฟล์นั้นแล้วส่งไฟล์ต่อไปก็ได้

12. ถ้าในกรณีที่ส่งไฟล์แรกสำเร็จแล้วและยังมีไฟล์ที่ต้องการส่งอีกก็จะไปเริ่มทำในขั้นตอนที่ 7 ต่อคือส่งข้อมูลของไฟล์ถัดไปที่ต้องการส่งทำเช่นนี้เรื่อยไปจนส่งครบหมดทุกไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การพัฒนาโครงงาน

โครงงานนี้เป็นโครงงานที่พัฒนาโปรแกรมต่อจากโครงงานโปรแกรมส่งสารด่วนแบบปลอดภัยด้วยจาวา จึงมีการนำโปรแกรมบางส่วนของโครงงานเดิมมาใช้งานในการพัฒนาต่อและบางส่วนที่เขียนขึ้นมาใหม่โดยขอแบ่งโครงสร้างของโปรแกรมออกเป็น 3 ส่วนใหญ่ ๆ ดังนี้

1. ส่วนติดต่อกับผู้ใช้งาน ในส่วนนี้มีการออกแบบและเขียนโปรแกรมใหม่ทั้งหมดโดยใช้จาวาสวิง (Swing หรือ Java Foundation Class) เพราะเห็นว่ามีความสามารถมากกว่าใช้ awt ที่ใช้ในโครงงานเดิม ทำให้ในการออกแบบไม่สามารถนำของเก่ามาใช้ได้เลย

2. ส่วนติดต่อกับเน็ตเวิร์ก สำหรับส่วนนี้มีการออกแบบโครงสร้างหลัก ๆ และเขียนโปรแกรมใหม่แต่ได้นำบางส่วนของโครงงานเดิมที่สามารถใช้ได้มาประกอบด้วยเพื่อความรวดเร็วในการพัฒนา

3. ส่วนของการเข้ารหัสและถอดรหัส เทคนิคที่ใช้ในการเข้ารหัสยังคงใช้เทคนิคเดิมดังนั้นโปรแกรมในส่วนนี้จึงได้นำของเก่ามาใช้ทั้งหมด

5.1 การพัฒนาส่วนติดต่อกับผู้ใช้งาน

ส่วนติดต่อกับผู้ใช้งานนับว่าเป็นส่วนที่มีความสำคัญเป็นอย่างมาก เนื่องจากผู้ใช้ส่วนใหญ่ต้องการโปรแกรมที่ใช้งานง่าย ไม่ต้องเสียเวลาในการเรียนรู้ ดังนั้นส่วนติดต่อกับผู้ใช้งานที่ออกแบบจึงคล้ายกับโปรแกรม ICQ ซึ่งเป็นโปรแกรมต้นแบบของการพัฒนา โดยส่วนหน้าต่างหลักที่ออกแบบไว้เป็นดังรูปที่ 5-1 ซึ่งสามารถแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

1. ส่วนบนของหน้าต่าง แสดงหมายเลข UIN ของผู้ใช้งาน

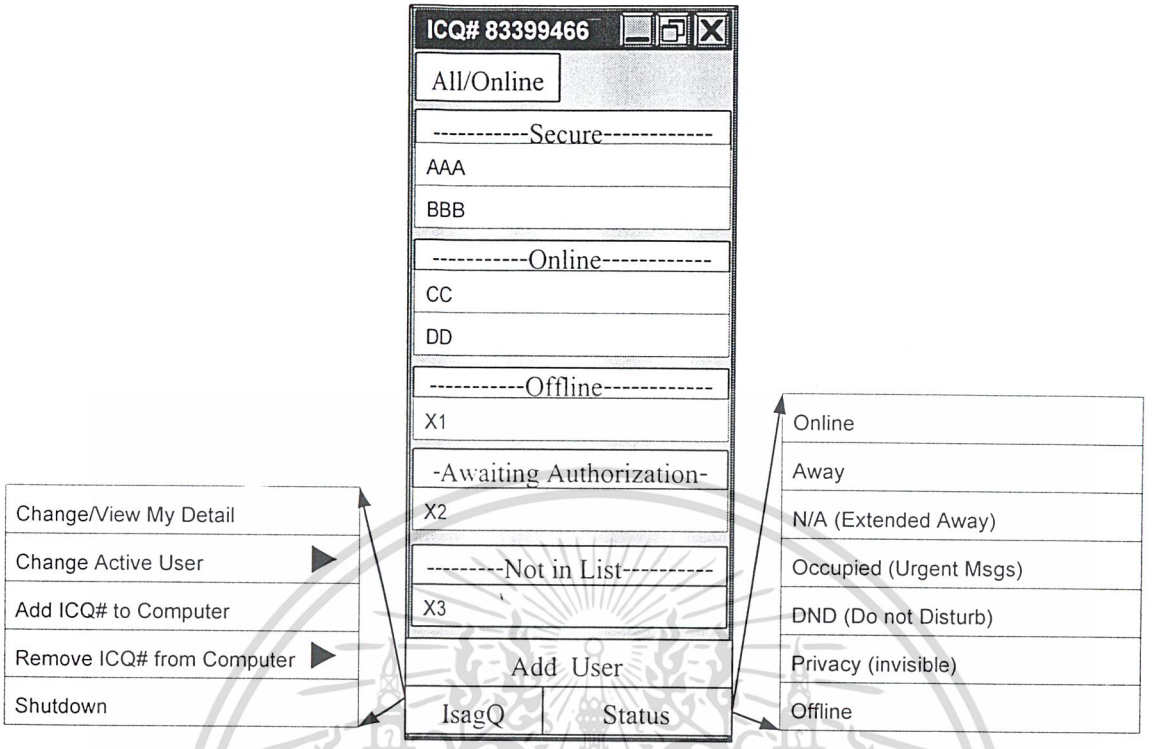
2. ส่วนแสดงรายชื่อผู้ใช้ในคอนแทคลิสต์ โดยแบ่งเป็น 5 กลุ่ม คือ

- ซีเคียว (Secure) แสดงรายชื่อผู้ใช้ที่ออนไลน์ด้วยโปรแกรม IsagQ
- ออนไลน์ (Online) แสดงรายชื่อผู้ใช้ที่ออนไลน์ด้วยโปรแกรม ICQ
- ออฟไลน์ (Offline) แสดงรายชื่อผู้ใช้ที่ออฟไลน์
- รอการให้อนุญาต (Awaiting Authorization) แสดงรายชื่อผู้ใช้ที่ยังไม่อนุญาตให้เพิ่มชื่อเข้าสู่คอนแทคลิสต์
- ไม่อยู่ในคอนแทคลิสต์ (Not in List) แสดงรายชื่อผู้ใช้ที่ผู้ใช้งานยังไม่ได้เพิ่มเข้าสู่คอนแทคลิสต์

3. ส่วนแสดงฟังก์ชันต่าง ๆ ได้แก่ ปุ่ม Add User, ปุ่ม Status ที่แสดงป๊อพอัพเมนู (Popup Menu) สำหรับเปลี่ยนสถานะ, ปุ่ม IsagQ ที่แสดงป๊อพอัพเมนูสำหรับฟังก์ชันอื่น ๆ และปุ่ม All/Online สำหรับเลือกรูปแบบการแสดงคอนแทคลิสต์

นอกจากหน้าต่างหลักที่กล่าวไปยังมีหน้าต่างอื่นอีก เช่น หน้าต่างสำหรับรับส่งข้อความ, หน้าต่างสำหรับรับส่งไฟล์, หน้าต่างสำหรับค้นหาผู้ใช้ เป็นต้น ซึ่งได้ออกแบบมาคล้ายกับของโปรแกรม ICQ อีกเช่นกันจึงไม่ขอกล่าวถึงรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-1 แสดงรูปของโปรแกรมที่ออกแบบก่อนการพัฒนา

5.1.1 ขั้นตอนในการพัฒนาส่วนติดต่อกับผู้ใช้งาน

การสร้างส่วนติดต่อกับผู้ใช้งานในโครงงานนี้เลือกใช้คอมโพเนนท์ (component) ของสวิง เนื่องจากเหตุผลดังต่อไปนี้

1. ลักษณะภายนอก (look and feel) ของโปรแกรมที่พัฒนาโดยใช้สวิงสามารถเลือกได้ว่าจะให้มีลักษณะตามแบบแพลตฟอร์มใด (เช่น วินโดวส์, โซลาริส, แมคอินทอช) รวมถึงสามารถนำไปใช้งานบนแพลตฟอร์มใด ๆ แล้วให้การแสดงผลของโปรแกรมเป็นรูปแบบเดียวกัน หรือจะให้มีลักษณะตามแพลตฟอร์มนั้น ๆ ก็ได้

2. คอมโพเนนท์ของสวิงได้พัฒนามาจาก awt จึงทำให้มีคุณสมบัติหรือเมธอดให้ใช้หลากหลายและยืดหยุ่นต่อการนำไปใช้มากกว่า

3. คอมโพเนนท์ของสวิงได้ถูกออกแบบมาให้มีความสวยงามมากกว่า

แต่การใช้คอมโพเนนท์ของสวิงก็มีปัญหาอยู่บ้าง คือ ไม่สามารถแสดงผลภาษาไทยได้ แต่ทางบริษัทซันประเทศไทยได้มีการพัฒนา Java™ Standard Edition, Thai Locallization (JDK1.2.2_th) ขึ้นมาซึ่งพัฒนามาจาก Java2 Platform Standard Edition v.1.2.2 โดยพัฒนาให้สนับสนุนชุดตัวอักษรภาษาไทยตามมาตรฐานอุตสาหกรรม สมอ.620-2533 (TIS620) ด้วยเหตุนี้จึงเกิดข้อจำกัดในการใช้งานโปรแกรมที่พัฒนาขึ้นมาคือสามารถใช้งานได้เฉพาะบนเครื่องคอมพิวเตอร์ที่มีจาวาเวอร์ชวลแมชีนของ JDK1.2.2_th เท่านั้น (มีการพัฒนาออกมาสำหรับผู้พัฒนาบนวินโดวส์และบนโซลาริสเท่านั้น) นอกจากนี้ปัญหาที่ต่อเนื่องมาจากการใช้ JDK1.2.2_th คือไม่สามารถแสดงผลรหัสแอสกี (ASCII) บางค่าได้เนื่องจากตามมาตรฐาน สมอ. 620-2533 มีการกำหนดให้ตัวอักษรตามรหัสแอสกีบางค่าห้ามนำไปใช้งาน (ได้แก่ ค่า 80-A0, DB, DD-DE, FC-FF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนต่อไปจะกล่าวถึงขั้นตอนในการพัฒนาส่วนติดต่อกับผู้ใช้งานซึ่งมีดังนี้

5.1.1.1 การเลือกคอมโพเนนท์ที่จะนำมาใช้จัดการกับข้อมูลแต่ละส่วนตามความเหมาะสม

ในการเลือกคอมโพเนนท์ที่จะนำมาใช้ต้องทำการศึกษาคุณสมบัติต่าง ๆ ของคอมโพเนนท์ในภาษาจาวาให้เข้าใจเสียก่อนเพื่อให้ได้คอมโพเนนท์ที่มีความเหมาะสมที่สุด และในการพัฒนาได้เลือกคอมโพเนนท์ดังนี้

1. หน้าต่างหลักและหน้าต่างอื่น ๆ เช่น หน้าต่างที่ใช้รับส่งข้อความหรือไฟล์จะใช้ JFrame

2. ปุ่มต่าง ๆ ใช้ JButton

3. ป๊อพอัพเมนูต่าง ๆ ใช้ JPopupMenu, JMenuItem, JMenu ประกอบกัน

4. คอนแทคลิสต์ซึ่งแบ่งเป็น 5 กลุ่มแต่ละกลุ่มจะใช้ JTable ในการแสดงรายชื่อแล้ววางใน JScrollPane อีกที เนื่องจากสามารถจัดการเกี่ยวกับการเพิ่มหรือลบข้อมูล, ค้นหาข้อมูล และสามารถแสดงรูปหน้าข้อความได้ง่าย เพราะข้อมูลจะเก็บอยู่ภายใน TableModel โดยข้อมูลที่ใส่ใน TableModel อาจจะเป็นออบเจ็กต์ที่สร้างขึ้นมาเองก็ได้ และในส่วนการแสดงผลในแต่ละช่องจะใช้ TableCellRenderer จัดการเกี่ยวกับการแสดงไอคอน โดยมันจะจัดการอัพเดทไอคอนหรือชื่อให้อัตโนมัติหากข้อมูลของออบเจ็กต์นั้นที่เก็บใน TableModel มีการเปลี่ยนแปลง

5. สำหรับพื้นที่แสดงผลต่าง ๆ จะแบ่งเป็นส่วน ๆ ตามความเหมาะสมในการใช้งาน แล้วใช้ JPanel ซึ่งเป็นคอนเทนเนอร์ (Container) มาเป็นพื้นที่รองรับคอมโพเนนท์ต่าง ๆ เพื่อความสะดวกในการแก้ไขขนาดหรือตำแหน่งในภายหลัง

5.1.1.2 การออกแบบลักษณะการจัดเรียงคอมโพเนนท์และนำคอมโพเนนท์ที่เลือกไว้มาประกอบกัน

การจัดเรียงคอมโพเนนท์ในคอนเทนเนอร์ของภาษาจาวาจะมีเลย์เอาต์เมเนเจอร์ (layout manager) ทำหน้าที่จัดการเกี่ยวกับการวางตำแหน่งและขนาดแต่ละคอมโพเนนท์ในคอนเทนเนอร์นั้น โดยจะรักษาลำดับของคอมโพเนนท์ที่ถูกนำเข้ามา ถ้าหากไม่มีการกำหนดตำแหน่งคอมโพเนนท์ที่เพิ่มเข้าไปก็จะไปวางต่อจากอันสุดท้ายแทน

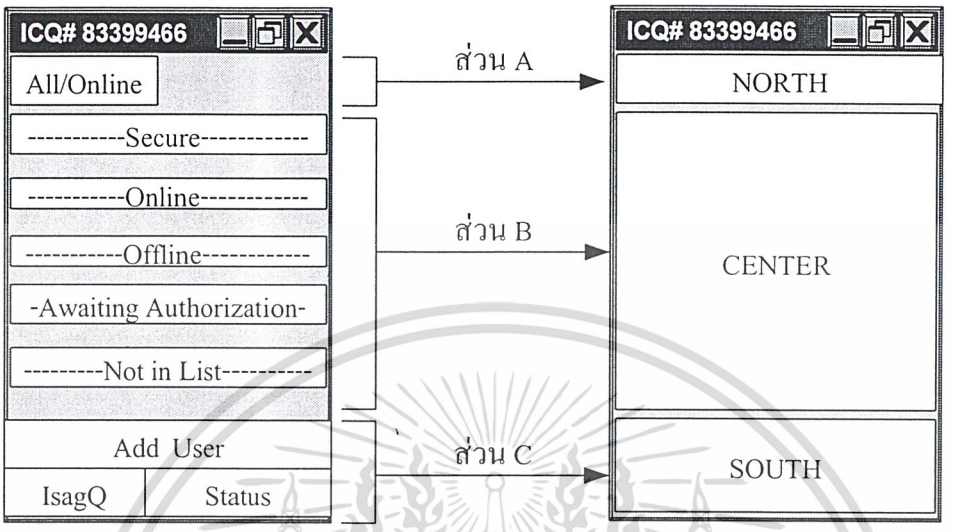
ประโยชน์ของเลย์เอาต์เมเนเจอร์คือ ช่วยให้การเพิ่มคอมโพเนนท์ลงในคอนเทนเนอร์ทำได้ง่าย โดยไม่ต้องระบุตำแหน่ง x, y ของแต่ละคอมโพเนนท์ให้ยุ่งยาก นอกจากนี้โปรแกรมภาษาจาวามักถูกนำไปใช้งานบนเครื่องต่างระบบ ซึ่งอาจมีวิธีการแสดงผลที่แตกต่างกันออกไปทำให้ตำแหน่งและขนาดของภาพต่างกัน หากเราระบุตำแหน่งด้วยค่าจริงก็อาจจะดูดีในเครื่องระบบหนึ่ง แต่เมื่อนำไปใช้งานบนเครื่องอีกระบบหนึ่งก็อาจทำให้มองไม่เห็นบางคอมโพเนนท์หรือการจัดเรียงไม่เป็นตามที่ออกแบบไว้ ด้วยเหตุนี้เราจึงควรหลีกเลี่ยงการระบุตำแหน่งคอมโพเนนท์ด้วยค่าจริง และให้เลย์เอาต์เมเนเจอร์จัดการแทน ซึ่งจะทำให้คอมโพเนนท์ถูกจัดการด้วยกฎเกณฑ์เดียวกัน แม้จะอยู่ในระบบหรือบราวเซอร์ที่แตกต่างกันผลที่ได้จะดูคล้ายกัน

เลย์เอาต์เมเนเจอร์เป็นอินเทอร์เฟซ (interface) ที่กำหนดไว้สำหรับนำไปสร้างเป็นคลาสของเลย์เอาต์เมเนเจอร์ที่มีรูปแบบในการจัดเรียงคอมโพเนนท์ตามที่ต้องการ แต่เพื่อความสะดวกจึงได้มีการสร้างคลาสของเลย์เอาต์เมเนเจอร์ไว้บ้างแล้วเช่น โฟลว์เลย์เอาต์ (FlowLayout), บอร์เดอร์เลย์เอาต์ (BorderLayout), กริดเลย์เอาต์ (GridLayout), กริดแบ็กเลย์เอาต์ (GridBagLayout) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

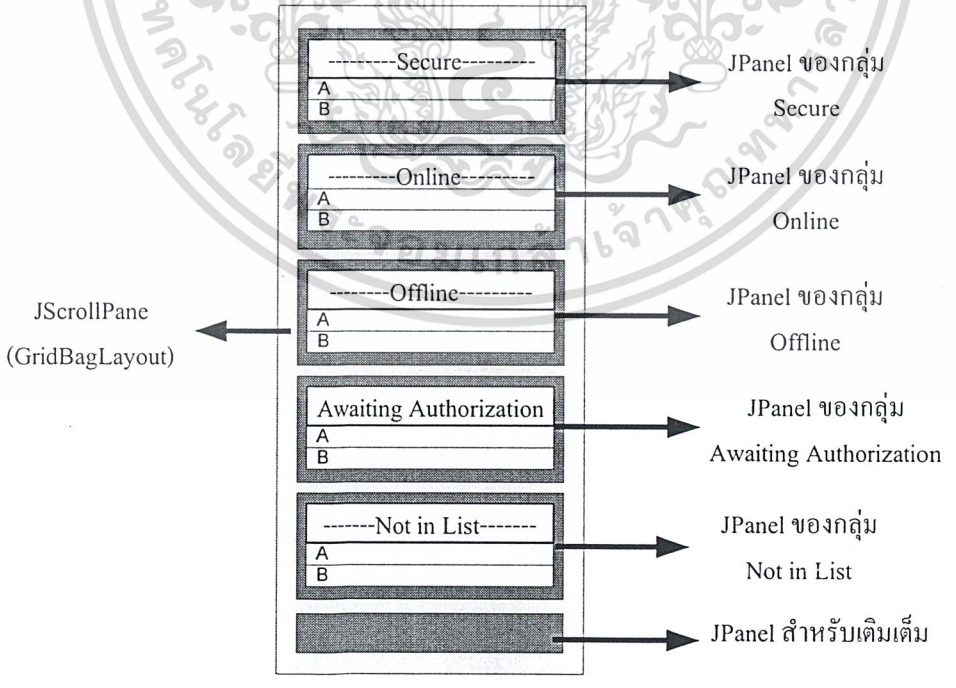
ในการพัฒนาก็ได้ใช้เลย์เอาต์เมเนเจอร์เหล่านี้ด้วยซึ่งมีดังนี้

1. หน้าต่างหลักจะแบ่งเป็น 3 ส่วนดังรูปที่ 5-2 โดยของเรียกแต่ละส่วนว่าส่วน A, B และ C ดังนั้นเลย์เอาต์ที่เหมาะสมคือบอร์เดอร์เลย์เอาต์



รูปที่ 5-2 แสดงการจัดเลย์เอาต์ในส่วนหน้าต่างหลัก

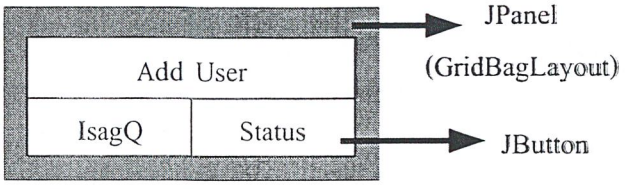
2. ในส่วนของคอนแทคลิสต์ประกอบด้วยส่วนแสดงรายชื่อผู้ใช้ในคอนแทคลิสต์ 5 ส่วนสำหรับ 5 กลุ่มและอีกส่วนเป็นพื้นที่ว่างใช้เป็นส่วนเติมเต็มพื้นที่ระหว่างชื่อผู้ใช้กลุ่มล่างสุดกับส่วน C สำหรับพื้นที่ทั้งหมดที่ใช้แสดงคอนแทคลิสต์ก็คือ JScrollPane ซึ่งจะใช้กริดแบ็กเลย์เอาต์จัดการดังรูปที่ 5-3



รูปที่ 5-3 แสดงการจัดเลย์เอาต์ในส่วนคอนแทคลิสต์

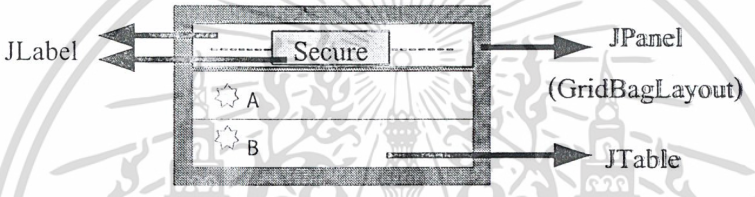
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วน C จะแบ่งเป็น 3 ส่วน ตามรูปที่ 5-4 ซึ่งการใช้กริดแบ็กเลย์เอาต์จะเหมาะสมที่สุด



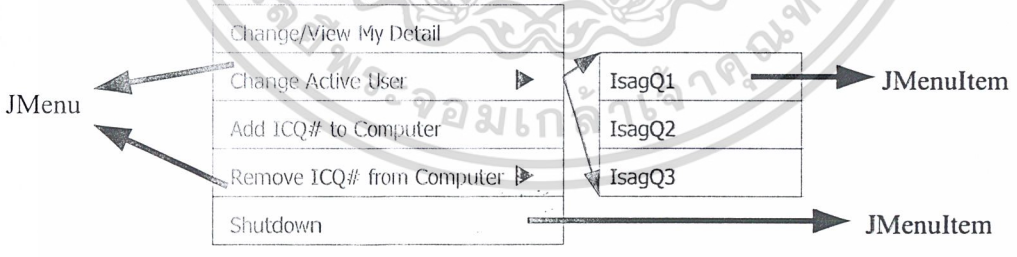
รูปที่ 5-4 แสดงการจัดเลย์เอาต์ในส่วนปุ่มด้านล่างของหน้าต่างหลัก

4. ส่วนแสดงรายชื่อผู้ใช้แต่ละกลุ่มจะแบ่งเป็น 2 ส่วน คือ ส่วนหัวสำหรับบอกว่าเป็นผู้ใช้กลุ่มไหน และส่วนแสดงรายชื่อผู้ใช้ โดยส่วนหัวจะใช้เลเบล (JLabel) 2 อันซ้อนกันแล้วกำหนดคุณสมบัติของเลเบลอันที่มีข้อความให้เป็นแบบไม่โปร่งใส ซึ่งจะเป็นดังรูปที่ 5-5 โดยส่วนนี้จะใช้กริดแบ็กเลย์เอาต์



รูปที่ 5-5 แสดงการจัดเลย์เอาต์ในส่วนแสดงรายชื่อผู้ใช้แต่ละกลุ่ม

5. ป๊อพอัพเมนูของปุ่ม IsagQ จะมีลักษณะดังรูปที่ 5-6 ซึ่งสามารถสร้างได้โดยขั้นตอนดังนี้ สร้าง JPopupMenu ขึ้นมาแล้วเพิ่มเมนูย่อยเข้าไปตามลำดับที่ต้องการ ซึ่งเมนูที่จะเพิ่มเข้าไปในป๊อพอัพเมนูจะมี 2 แบบคือ JMenuItem ใช้สำหรับเมนูที่ไม่มีการเปิดเมนูย่อยอันอื่นอีก และ JMenu สำหรับเมนูที่ต้องการเปิดเมนูย่อยอีก ในการเพิ่มเมนูย่อยทำโดยเพิ่ม (add) JMenuItem ให้กับ JMenu ที่ต้องการ



รูปที่ 5-6 แสดงการจัดเลย์เอาต์ของป๊อพอัพเมนูของปุ่ม IsagQ

6. ป๊อพอัพเมนูจากปุ่ม Status จะมีวิธีสร้างคล้ายกับของปุ่ม IsagQ

7. หน้าต่างอื่น ๆ ก็ใช้หลักการที่กล่าวมาไปประยุกต์ใช้ โดยขึ้นกับองค์ประกอบที่มีในหน้าต่างนั้น ๆ

5.1.1.3 เขียนโปรแกรมจัดการในส่วนที่ต้องมีการโต้ตอบกับผู้ใช้

ในการพัฒนาต้องมีการเขียนโปรแกรมจัดการกับอีเวนต์ต่าง ๆ ที่เข้ามา ซึ่งสามารถทำได้โดยเรียกใช้เมธอดที่มีอยู่แล้วในภาษาจาวา ตัวอย่างเช่น ถ้าต้องการตรวจสอบว่ามีการคลิกเมาส์บนพื้นที่ส่วนหนึ่งเมื่อใดบ้างก็ต้อง add MouseListener() ให้กับอินสแตนซ์ที่เป็นพื้นที่ส่วนนั้น แล้วเขียนเมธอดจัดการว่าต้องการให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำอะไรหากมีการคลิกเมาส์ ซึ่งในเมธอดนี้ก็สามารถตรวจสอบได้ว่าเป็นการคลิกเมาส์ปุ่มไหน หรือคลิกกี่ครั้ง เนื่องจากเมธอดนี้จะรับพารามิเตอร์เป็นอีเวนต์ของเมาส์ เป็นต้น ตัวอย่างส่วนที่ต้องจัดการเรื่องอีเวนต์ได้แก่

1. ถ้ามีการคลิกเมาส์ในบริเวณที่เป็นคอนแทคลิสต์จะมีการตรวจสอบก่อนว่าเป็นการคลิกปุ่มซ้ายหรือขวา ถ้าคลิกปุ่มขวาจะแสดงป๊อพอัพเมนู แต่ถ้าเป็นปุ่มซ้ายแบบดับเบิ้ลคลิกจะแสดงหน้าต่างอื่น เช่น ส่งข้อความ หรือรับข้อความ ซึ่งขึ้นกับข้อมูลที่เข้ามาจากผู้ใช้ชื่อ

2. ถ้ามีการกดปุ่ม IsagQ จะแสดงป๊อพอัพเมนูที่มีฟังก์ชันต่าง ๆ ให้ผู้ใช้เลือก

3. ถ้ามีการกดปุ่ม Status จะแสดงป๊อพอัพเมนูให้ผู้ใช้เปลี่ยนสถานะ

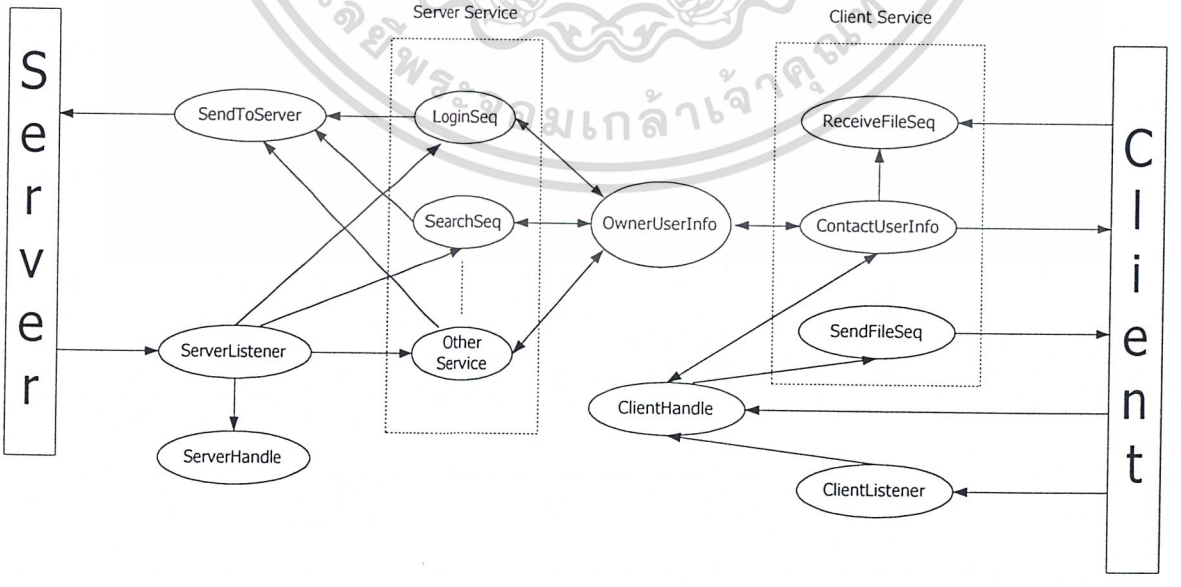
4. ถ้ามีการกดปุ่ม Add User จะแสดงหน้าต่างสำหรับค้นหาผู้ใช้

5.1.1.4 นำโปรแกรมที่พัฒนาส่วนติดต่อกับผู้ใช้งานแล้วไปเชื่อมต่อกับส่วนติดต่อกับเน็ตเวิร์ก

ในส่วนนี้จะเป็นการพัฒนาที่ต้องมีการพัฒนาควบคู่กันไปทั้งในส่วนติดต่อผู้ใช้และส่วนเน็ตเวิร์ก ซึ่งไม่สามารถกล่าวถึงรายละเอียดได้มากนัก เพราะต้องมีการทดลองติดต่อสื่อสารกับเซิร์ฟเวอร์ของ ICQ และดูผลการทำงานว่าต้องมีการแสดงผลอะไรแก่ผู้ใช้บ้างแล้วมาแก้ไขในส่วนติดต่อกับผู้ใช้งานเพิ่มเติม ซึ่งต้องทดลองอย่างต่อเนื่องให้ครอบคลุมทุกฟังก์ชันที่มีจนกว่าการทำงานจะสมบูรณ์

5.2 การพัฒนาส่วนติดต่อกับเน็ตเวิร์ก

โปรแกรมในส่วนนี้มีหน้าที่ในการรับส่งข้อมูลผ่านทางเน็ตเวิร์กไม่ว่าจะเป็นการติดต่อกับเซิร์ฟเวอร์หรือการติดต่อระหว่างไคลเอนต์ ในส่วนนี้จึงต้องพัฒนาโปรแกรมทั้งขั้นตอนในการสื่อสารรวมถึงแพ็กเก็ตที่ใช้ในการสื่อสารให้มีการทำงานที่สอดคล้องตามโพรโตคอลของ ICQ โปรแกรมในส่วนนี้ส่วนมากมีลักษณะเป็นเธรดเพราะการทำงานมีลักษณะควบคู่กันไปเช่น การส่งแพ็กเก็ตไปยังเซิร์ฟเวอร์อาจเกิดขึ้นในเวลาเดียวกับการรับข้อมูลจากเซิร์ฟเวอร์ สำหรับโครงสร้างหลักของโปรแกรมในส่วนนี้สามารถแสดงได้ดังรูปที่ 5-7



รูปที่ 5-7 แสดงโครงสร้างหลักของโปรแกรมในส่วนติดต่อกับเน็ตเวิร์ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

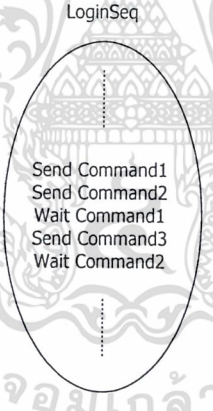
จากรูปเราสามารถอธิบายหน้าที่และรายละเอียดของส่วนต่างๆได้ดังนี้

5.2.1 OwnerUserInfo

คลาสนี้เป็นคลาสที่เปรียบเสมือนเป็นตัวแทนของผู้ใช้งาน โดยในคลาสนี้มีเมธอดต่างๆ ที่เกี่ยวกับการใช้บริการจากเซิร์ฟเวอร์และการติดต่อสื่อสารกับไคลเอนต์อื่น โดยผู้ใช้งานจะสั่งงานผ่านส่วนติดต่อผู้ใช้งานเพื่อเรียกเมธอดต่างๆ ในคลาสนี้และคลาสนี้จะไปเรียกใช้บริการจากคลาสอื่นที่ทำหน้าที่เฉพาะงานที่ต้องการอีกที การออกแบบคลาสนี้พยายามให้ดูง่ายที่สุดคือซ่อนความซับซ้อนในการติดต่อเอาไว้เช่น ในการล็อกอินหรือ การค้นหา เราจะกำหนดไว้เป็น 1 เมธอด เพียงเรียกเมธอดนั้นและส่งค่าที่จำเป็นก็จะได้ผลลัพธ์ออกมาเป็นต้น

5.2.2 LoginSeq

จากรูปจะเห็นว่าคลาสนี้อยู่ในกลุ่มของ Server Service ซึ่งคลาสในกลุ่มนี้จะมีลักษณะเป็นเธรดซึ่งทำหน้าที่ควบคุมขั้นตอนการติดต่อในการใช้บริการจากเซิร์ฟเวอร์ให้ถูกต้องตามขั้นตอนของโพรโตคอล ICQ ที่ต้องออกแบบคลาสในกลุ่มนี้ให้เป็นเธรดทั้งหมดก็เพราะว่าในการใช้บริการจากเซิร์ฟเวอร์อาจมีการใช้บริการมากกว่า 1 อย่างในเวลาเดียวกันเราต้องสามารถควบคุมทุก ๆ ขั้นตอนของทุกบริการได้ในเวลาเดียวกัน สำหรับโครงสร้างอย่างหยาบของโปรแกรมในคลาสนี้สามารถแสดงได้ดังรูปที่ 5-8

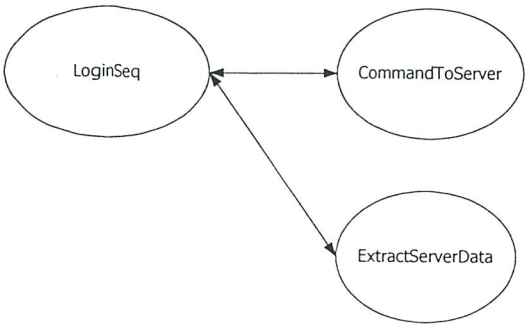


รูปที่ 5-8 แสดงโครงสร้างของคลาส LoginSeq

จากรูปที่ 5-8 จะเห็นว่าในคลาสนี้จะกำหนดขั้นตอนเอาไว้ว่าต้องส่งคำสั่งไหนก่อนคำสั่งไหนหลัง และในการส่งคำสั่งต่อไปต้องรอคำสั่งอะไรจากเซิร์ฟเวอร์บ้างถึงจะส่งได้

การทำงานของคลาสในกลุ่มนี้ต้องส่งและรับแพ็กเก็ตกับเซิร์ฟเวอร์ ซึ่งในการสร้างแพ็กเก็ตสำหรับส่งไปให้เซิร์ฟเวอร์หรือการแกะข้อมูลจากแพ็กเก็ตที่เซิร์ฟเวอร์ส่งมาให้ต้องใช้อีก 2 คลาสช่วยเหลือในการทำหน้าที่สร้างแพ็กเก็ตและแกะแพ็กเก็ตซึ่งก็คือคลาส CommandToServer สำหรับสร้างแพ็กเก็ตของคำสั่งที่ต้องการ และคลาส ExtractServerData สำหรับแกะแพ็กเก็ตที่ส่งมาจากเซิร์ฟเวอร์ ซึ่งการติดต่อระหว่างคลาสในกลุ่มนี้แสดงดังรูปที่ 5-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

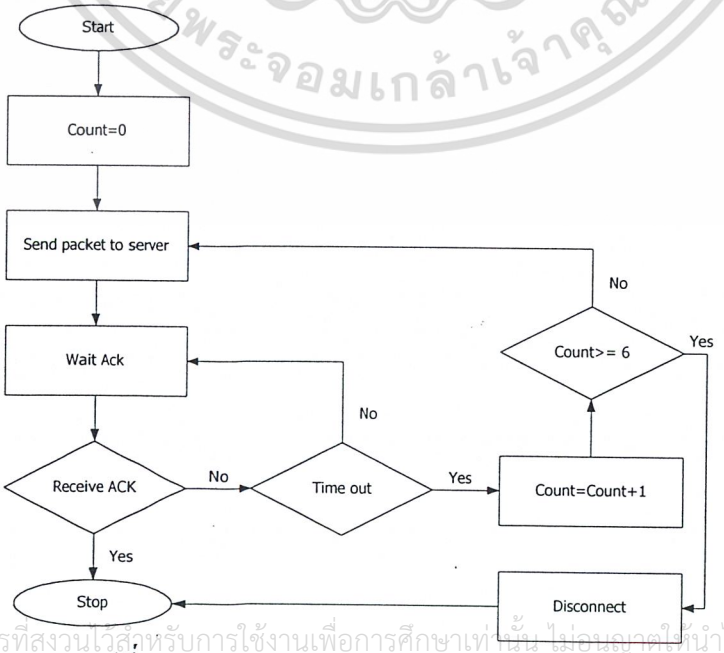


รูปที่ 5-9 แสดงการติดต่อระหว่างคลาส LoginSeq กับคลาสอื่น

จากรูปที่ 5-9 จะเห็นว่าคลาส LoginSeq มีการติดต่อกับคลาส CommandToServer และคลาส ExtractServerData โดยคลาส LoginSeq จะเรียกเมธอดในคลาส CommandToServer ซึ่งมีชื่อตรงกับคำสั่งที่ต้องการส่งพร้อมทั้งพารามิเตอร์ที่จำเป็นสำหรับคำสั่งนั้นจากนั้นเมธอดในคลาส CommandToServer จะสร้างแพ็กเก็ตให้และส่งคืนกลับมาในรูปแบบอาเรย์ของไบต์ และหากคลาส LoginSeq ได้แพ็กเก็ตจากเซิร์ฟเวอร์มาก็ส่งไปให้คลาส ExtractServerData แกะข้อมูลนั้นจากนั้นจึงตรวจสอบดูว่าเป็นแพ็กเก็ตที่รออยู่หรือไม่ สำหรับคลาสอื่น ๆ ที่อยู่ในกลุ่มนี้ก็มีโครงสร้างและการทำงานเช่นเดียวกับคลาส LoginSeq

5.2.3 SendToServer

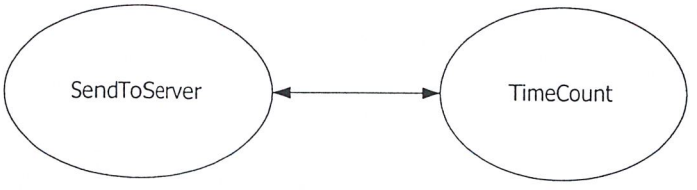
ตามโพรโตคอลของ ICQ ในการส่งแพ็กเก็ตของคำสั่งต่างๆ ไปยังเซิร์ฟเวอร์หากไม่มีการตอบรับมาภายใน 10 วินาทีต้องส่งแพ็กเก็ตนั้นไปใหม่อีก และหากมีการส่งซ้ำเกิน 6 รอบจะถือว่าการเชื่อมต่อมีปัญหาทางฝั่งไคลเอนต์ต้องตัดการเชื่อมต่อ ดังนั้นในการออกแบบการทำงานของคลาสนี้จึงออกแบบให้เป็นเธรดเพราะการส่งแพ็กเก็ตของคำสั่งไปยังเซิร์ฟเวอร์อาจมีการส่งหลายแพ็กเก็ตในขณะเดียวกัน หากไม่ออกแบบให้คลาสนี้เป็นเธรดจะทำให้ยากต่อการควบคุมในการส่งแพ็กเก็ตซ้ำ สำหรับขั้นตอนการทำงานของคลาสนี้สามารถแสดงเป็น FlowChart ดังรูปที่ 5-10



รูปที่ 5-10 แสดง Flow Chart ของคลาส SendToServer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

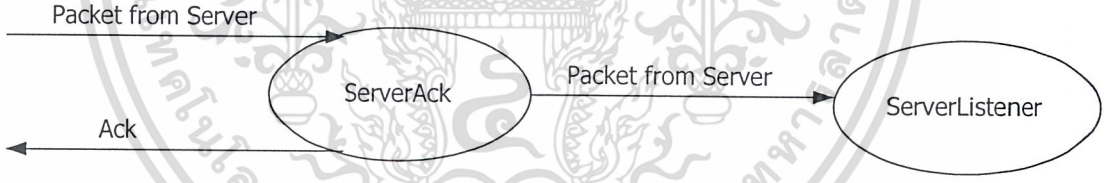
ในการทำงานของคลาสนี้ต้องเกี่ยวข้องกับการจับเวลาจึงต้องมีคลาสที่ช่วยจัดการเกี่ยวกับการจับเวลา ซึ่งก็คือคลาส TimeCount โดยเราสามารถกำหนดเวลาที่ต้องการได้เป็นหน่วยมิลลิวินาที และมีเมธอดสำหรับเช็คว่าหมดเวลาหรือยังโดยส่งค่ากลับมาเป็นบูลีน สำหรับการติดต่อระหว่างคลาสสามารถแสดงได้ดังรูปที่5-11



รูปที่ 5-11 แสดงการติดต่อระหว่างคลาส SendToServer และ TimeCount

5.2.4 ServerListener

การติดต่อกับเซิร์ฟเวอร์ต้องมีการรับแพ็กเก็ตที่เซิร์ฟเวอร์ส่งมาซึ่งคลาส ServerListener ทำหน้าที่ในส่วนนี้แต่ไม่ได้ติดต่อกับเซิร์ฟเวอร์โดยตรงแต่มีคลาส ServerAck มาช่วยในการทำงานส่วนนี้ซึ่งคลาส ServerAck ทำหน้าที่รับข้อมูลที่เซิร์ฟเวอร์ส่งมาและส่งแพ็กเก็ตของคำสั่ง ACK กลับไปยังเซิร์ฟเวอร์และนำแพ็กเก็ตของข้อมูลที่ได้รับมาส่งไปให้กับคลาส ServerListener เพื่อนำไปเข้าคิวรอการอ่านจากคลาสอื่น ๆ ซึ่งรอแพ็กเก็ตจากเซิร์ฟเวอร์อยู่ ในการออกแบบในส่วนนี้ต้องออกแบบให้สามารถส่งแพ็กเก็ต ACK กลับไปยังเซิร์ฟเวอร์ให้ได้เร็วที่สุดเพราะหากมีการทำงานที่ซับซ้อนหลายขั้นตอนจะทำให้เซิร์ฟเวอร์ส่งแพ็กเก็ตมาหลายครั้งอาจทำให้เซิร์ฟเวอร์ตัดการเชื่อมต่อได้ สำหรับการติดต่อระหว่างคลาสสามารถแสดงได้ดังรูปที่ 5-12



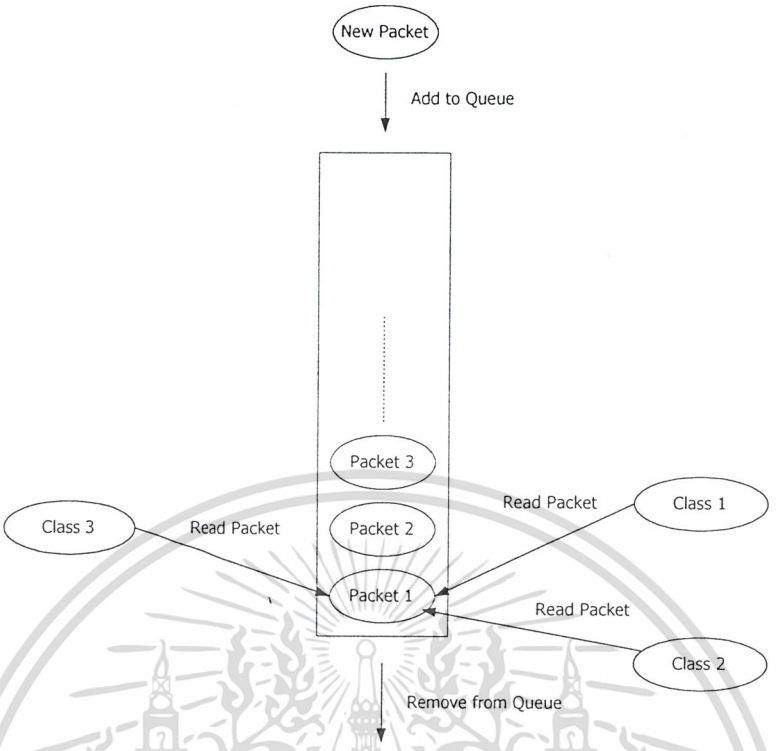
รูปที่ 5-12 แสดงการใช้คลาส ServerAck ร่วมกับคลาส ServerListener

สำหรับคลาสที่รอแพ็กเก็ตจากเซิร์ฟเวอร์อยู่ต้องแจ้งให้คลาส ServerListener รับรู้ เพราะคลาส ServerListener ต้องรู้ว่าขณะนี้มีผู้รอแพ็กเก็ตอยู่เท่าไหร่ และเมื่อคลาสที่รอแพ็กเก็ตไปอ่านแพ็กเก็ตจาก ServerListener ก็ต้องแจ้งให้ ServerListener รู้เช่นกัน โดยการอ่านต้องอ่านข้อมูลในคิวตัวที่เก่าที่สุด เมื่อ ServerListener พบว่าข้อมูลตัวนั้นถูกอ่านโดยทุกคลาสที่รอจนครบก็จะนำข้อมูลตัวนั้นออกจากคิวซึ่งแสดงได้ดังรูปที่ 5-13

5.2.5 ServerHandle

สำหรับบางกรณีแพ็กเก็ตที่เซิร์ฟเวอร์ส่งมาอาจไม่เกี่ยวกับแพ็กเก็ตที่เราส่งไปแต่อาจเกิดจากเหตุการณ์อื่น ๆ เช่น มีผู้ใช้คนอื่นส่งข้อความมาให้หรือมีผู้ใช้อื่นเพิ่มเราเข้าคอนแทคลิสต์ เป็นต้น ในกรณีนี้เราต้อ มีคลาสซึ่งจัดการกับแพ็กเก็ตที่เซิร์ฟเวอร์ส่งมาซึ่งก็คือคลาส ServerHandle นั่นเอง โดยคลาส ServerHandle จะทำงานทุกรอบที่มีแพ็กเก็ตมาจากเซิร์ฟเวอร์

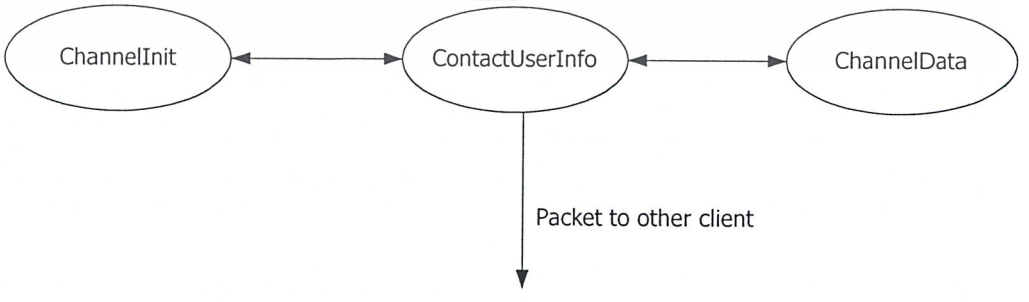
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-13 แสดงการจัดการคิวของคลาส ServerListener

5.2.6 ContactUserInfo

คลาสนี้เป็นคลาสที่แทนผู้ใช้ที่อยู่ในคอนแทคลิสต์โดยในคลาสนี้ประกอบไปด้วยเมธอดต่างๆที่ทำหน้าที่เกี่ยวกับการติดต่อระหว่างไคลเอนต์ การออกแบบคลาสนี้มีลักษณะคล้ายกับคลาส OwnerUserInfo คือพยายามซ่อนความซับซ้อนเอาไว้ สำหรับการติดต่อกับไคลเอนต์ตัวอื่นๆไม่ว่าจะเป็นการส่งข้อความหรือการขอส่งไฟล์ก็ต้องมีการสร้างแพ็กเก็ตที่จำเป็นในการสื่อสารซึ่งหน้าที่ในการสร้างแพ็กเก็ตสำหรับการสื่อสารโดยตรงจะทำโดยคลาส ChannelInit และคลาส ChannelData ซึ่งคลาส ChannelInit ใช้สำหรับสร้างแพ็กเก็ต CHANNEL_INIT และคลาส ChannelData ใช้สำหรับแพ็กเก็ตข้อความ, แพ็กเก็ตขอส่งไฟล์ และแพ็กเก็ตตอบรับ ซึ่งความสัมพันธ์ระหว่างคลาสแสดงดังรูปที่ 5-14



รูปที่ 5-14 แสดงการติดต่อระหว่างคลาส ContactUserInfo กับคลาสอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.7 SendFileSeq

เป็นคลาสที่ใช้สำหรับควบคุมการส่งไฟล์ระหว่างไคลเอนต์ให้ถูกต้องตาม File Direct Protocol ของ ICQ ในคลาสนี้มีการสร้างแพ็กเก็ตของการส่งไฟล์ขึ้นมาเองโดยมีหลักการเหมือนกับ ChannelData และ CommandToServer ส่วนการควบคุมขั้นตอนต่างๆ ก็มีการออกแบบเหมือนกับคลาสในกลุ่ม Server Service โดยคลาสนี้จะเริ่มส่งไฟล์ก็ต่อเมื่อได้รับคำสั่งจากคลาส ClientHandle เพราะคลาส ClientHandle เป็นตัวรับคำสั่ง ACK จากไคลเอนต์อีกฝั่งว่าต้องการรับไฟล์หรือไม่

5.2.8 ReceiveFileSeq

เป็นคลาสที่ใช้สำหรับควบคุมการรับไฟล์ระหว่างไคลเอนต์ให้ถูกต้องตาม File Direct Protocol ของ ICQ โดยคลาสนี้จะเริ่มการรับไฟล์ก็ต่อเมื่อได้รับคำสั่งจากคลาส ContactUserInfo นอกจากนี้คลาส ReceiveFileSeq และ SendFileSeq มีการทำงานร่วมกับ Progress Frame เพื่อแสดงความคืบหน้าในการรับส่งไฟล์ด้วย

5.2.9 ClientListener

เป็นคลาสที่ใช้สำหรับรอรับการติดต่อเข้ามาจากไคลเอนต์อื่นๆและเมื่อมีการติดต่อเข้ามาแล้วจะสร้างคลาส ClientHandle เพื่อทำหน้าที่ในการรับข้อมูลจากไคลเอนต์นั้นต่อไปโดยจะมี ClientHandle 1 ตัวสำหรับไคลเอนต์ที่ติดต่อเข้ามา 1 ตัว ต่อจากนั้นหากไคลเอนต์ตัวเดิมติดต่อเข้ามาอีกหน้าที่ในการรับข้อมูลจะเป็นของ ClientHandle ของไคลเอนต์ตัวนั้น

5.2.10 ClientHandle

ทำหน้าที่รอรับแพ็กเก็ตที่ส่งมาจากไคลเอนต์อื่นโดยคลาสนี้จะถูกสร้างโดยคลาส ClientHandle ถ้าไคลเอนต์อื่นติดต่อเข้ามาก่อนและจะถูกสร้างโดย ContactUserInfo ถ้าเราติดต่อไปยังไคลเอนต์อื่นก่อน

5.3 การพัฒนาส่วนของการเข้ารหัส

ในส่วนของการเข้ารหัสจะใช้การเข้ารหัสทั้งแบบ DES และ RSA ร่วมกันเพราะว่าการเข้ารหัสแบบ DES ใช้เวลาในการเข้ารหัสและถอดรหัสน้อยกว่าแบบ RSA แต่การเข้ารหัสแบบ RSA มีความปลอดภัยต่อการถูกดักจับคีย์มากกว่าแบบ DES เราจึงนำข้อดีของทั้งสองแบบมาใช้ร่วมกันคือใช้ RSA เข้ารหัสคีย์ DES เพื่อแลกคีย์ DES ในตอนแรกต่อจากนั้นในการสื่อสารจะใช้คีย์ DES ที่ได้มาในการเข้ารหัสข้อมูล ซึ่งขั้นตอนในการแลกเปลี่ยนคีย์แสดงดังรูปที่ 5-15 และมีรายละเอียดดังนี้

1. ทั้งไคลเอนต์ A และ B สร้างคีย์ e,n,d และ DES ของตนเองเก็บไว้

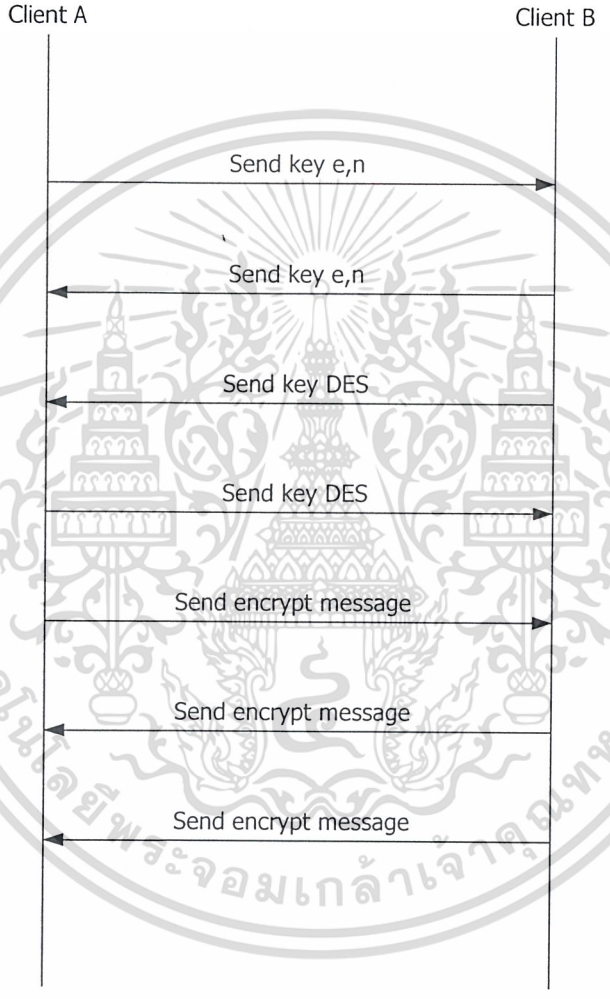
2. เมื่อไคลเอนต์ A ได้รับแจ้งจากเซิร์ฟเวอร์ว่าไคลเอนต์ B ออนไลน์ ไคลเอนต์ A จะส่งคีย์ e,n ของตนไปให้ไคลเอนต์ B

3. เมื่อไคลเอนต์ B ได้รับแจ้งจากเซิร์ฟเวอร์ว่าไคลเอนต์ A ออนไลน์ ไคลเอนต์ B จะส่งคีย์ e,n ของตนไปให้ไคลเอนต์ A เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อไคลเอนต์ B ได้รับคีย์ e,n ของไคลเอนต์ A จากนั้นไคลเอนต์ B ก็จะเข้ารหัสคีย์ DES ของตนด้วยคีย์ e,n ของไคลเอนต์ A จากนั้นจึงส่งคีย์ DES ที่เข้ารหัสแล้วกลับไปให้ไคลเอนต์ A ทางด้านไคลเอนต์ A ก็ทำเช่นเดียวกัน

5. เมื่อไคลเอนต์ A ได้รับคีย์ DES ของไคลเอนต์ B ก็จะถอดรหัสด้วยคีย์ d,n ของตนและเก็บคีย์ DES ของไคลเอนต์ B ไว้ใช้สำหรับเข้ารหัสข้อความหรือไฟล์เมื่อต้องการส่งไปให้ไคลเอนต์ B ส่วนทางด้านไคลเอนต์ B ก็ทำเช่นเดียวกัน



รูปที่ 5-15 แสดงการแลกเปลี่ยนคีย์ระหว่างไคลเอนต์

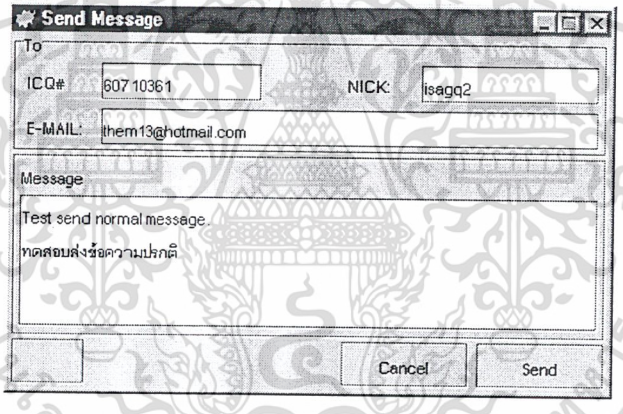
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การทดสอบโครงงาน

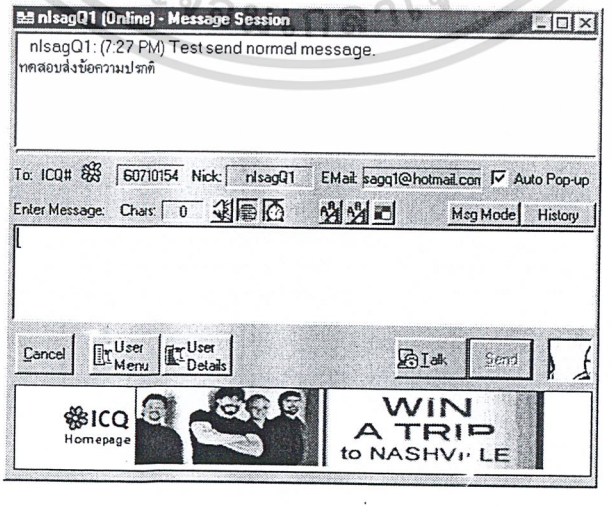
การทดสอบโปรแกรม IsagQ เราจะทดสอบให้เห็นถึงความปลอดภัยในการใช้งาน โดยจะส่งข้อความแบบปรกติและส่งข้อความที่มีการเข้ารหัสและมีการดักจับข้อความที่รับส่งกันเพื่อดูผลเปรียบเทียบที่ได้จากการดักจับว่าเป็นอย่างไร มีความปลอดภัยจริงหรือไม่ นอกจากนี้ก็จะทดสอบการรับส่งไฟล์ทั้งแบบปรกติและเข้ารหัสด้วยเพื่อนำผลมาเปรียบเทียบเช่นกัน

6.1 การทดสอบรับส่งข้อความแบบปรกติ

ในการส่งข้อความระหว่างโปรแกรม IsagQ กับโปรแกรม ICQ จะใช้การส่งข้อความแบบปรกติ และเราจะเห็นชื่อของผู้ใช้งานโปรแกรม ICQ แสดงอยู่ในส่วนออนไลน์ ซึ่งการส่งข้อความหรือไฟล์กับผู้ใช้งานที่อยู่ในส่วนออนไลน์จะไม่มีการเข้ารหัสซึ่งผลการทดสอบแสดงได้ดังรูปที่ 6-1, 6-2 และ 6-3



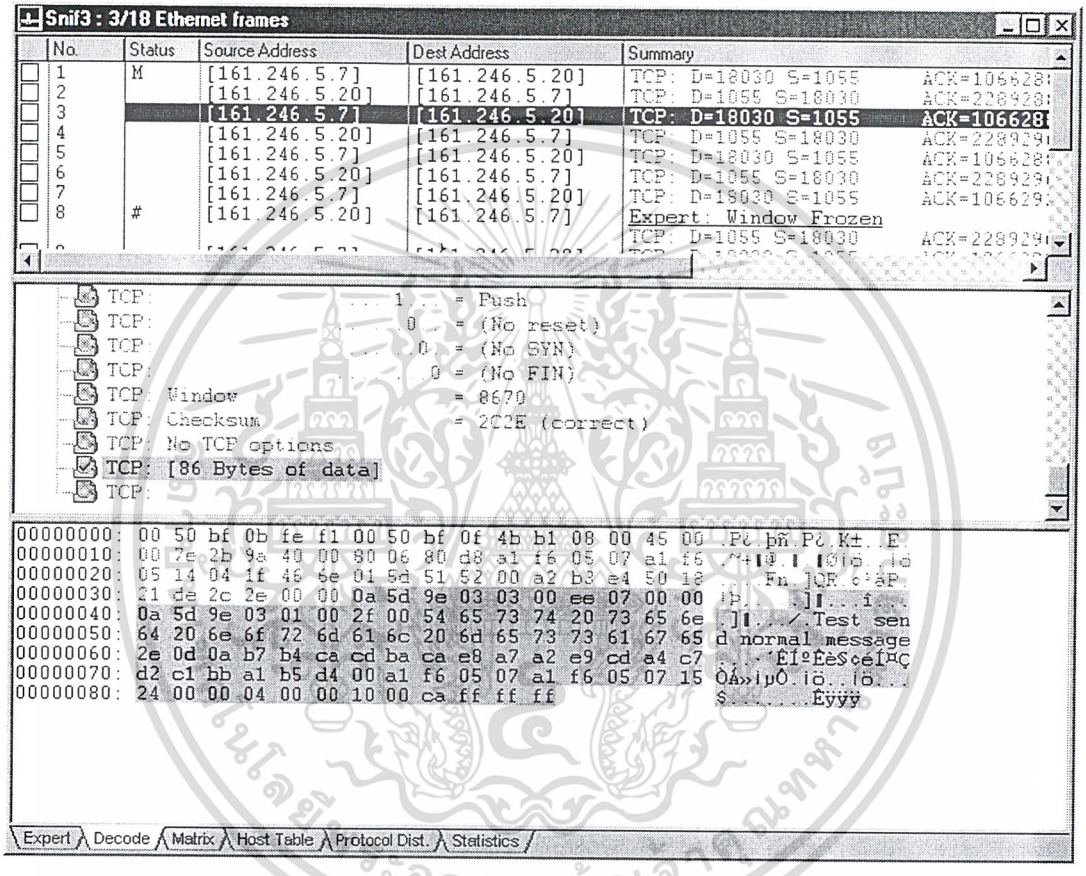
รูปที่ 6-1 แสดงหน้าต่างส่งข้อความแบบปรกติ



รูปที่ 6-2 แสดงหน้าต่างที่รับข้อความแบบปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6-1 แสดงหน้าต่างของการส่งข้อความทางฝ่ายผู้ส่งซึ่งข้อความจะมีทั้งภาษาอังกฤษและภาษาไทยเพื่อแสดงให้เห็นว่าโปรแกรมสามารถรองรับการใช้งานภาษาไทยได้ด้วย ส่วนรูปที่ 6-2 แสดงหน้าต่างของการรับข้อความทางฝ่ายผู้รับซึ่งจะเห็นว่าข้อความที่ได้รับมาตรงกับข้อความทางฝ่ายผู้ส่ง และในรูปที่ 6-3 แสดงการดักจับข้อความระหว่างผู้รับและผู้ส่งซึ่งจะเห็นว่าข้อความที่ดักจับได้สามารถที่จะอ่านได้โดยตรงซึ่งแสดงให้เห็นถึงความไม่ปลอดภัยของการส่งข้อมูลแบบปรกติซึ่งถ้าหากข้อมูลมีความสำคัญอาจถูกดักจับจากผู้ไม่ประสงค์ดีได้

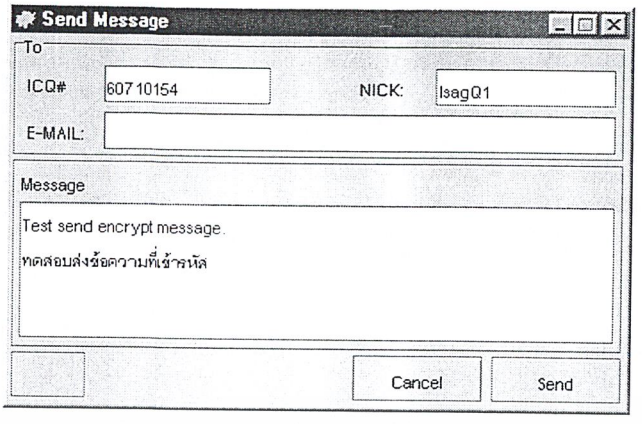


รูปที่ 6-3 แสดงหน้าต่างแสดงการดักจับข้อความที่ส่งแบบปรกติ

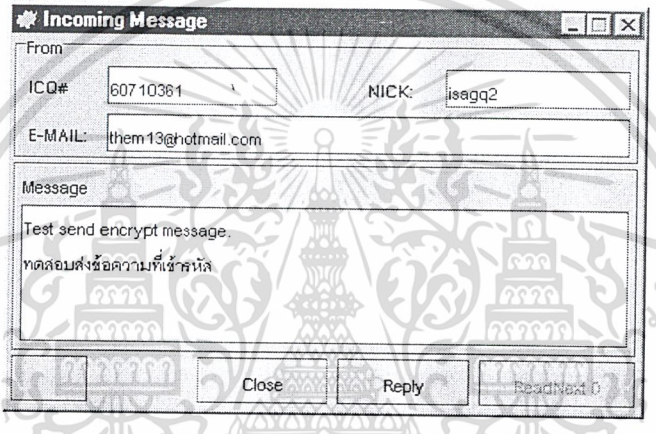
6.2 การทดสอบรับส่งข้อความแบบเข้ารหัส

สำหรับการติดต่อกับผู้ที่ใช้งานโปรแกรม IsagQ ด้วยกันและต้องสามารถติดต่อกันโดยตรงได้ การส่งข้อความหรือไฟล์ระหว่างกันจะมีการเข้ารหัสข้อมูลด้วย ซึ่งทำให้ข้อมูลที่อาจจะถูกดักจับไปนั้นยากต่อการอ่านและการถอดรหัสสำหรับผู้ที่ไม่รู้คีย์ในการถอดรหัส และชื่อของผู้ใช้งานที่ใช้โปรแกรม IsagQ จะแสดงผลอยู่ในส่วนของซีเคียวซึ่งแสดงให้รู้ว่าข้อมูลที่จะส่งไปยังผู้ใช้คนนี้จะมีการเข้ารหัสก่อน ซึ่งผลการทดสอบแสดงดังรูปที่ 6-4, 6-5 และ 6-6

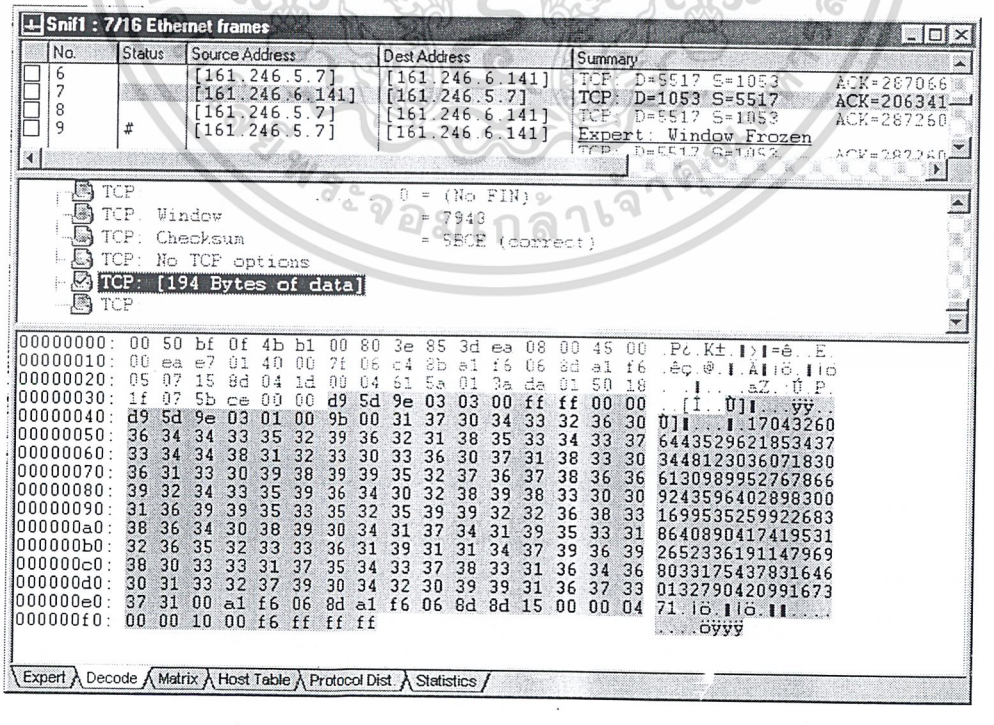
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-4 แสดงหน้าต่างส่งข้อความแบบเข้ารหัส



รูปที่ 6-5 แสดงหน้าต่างที่รับข้อความแบบเข้ารหัส



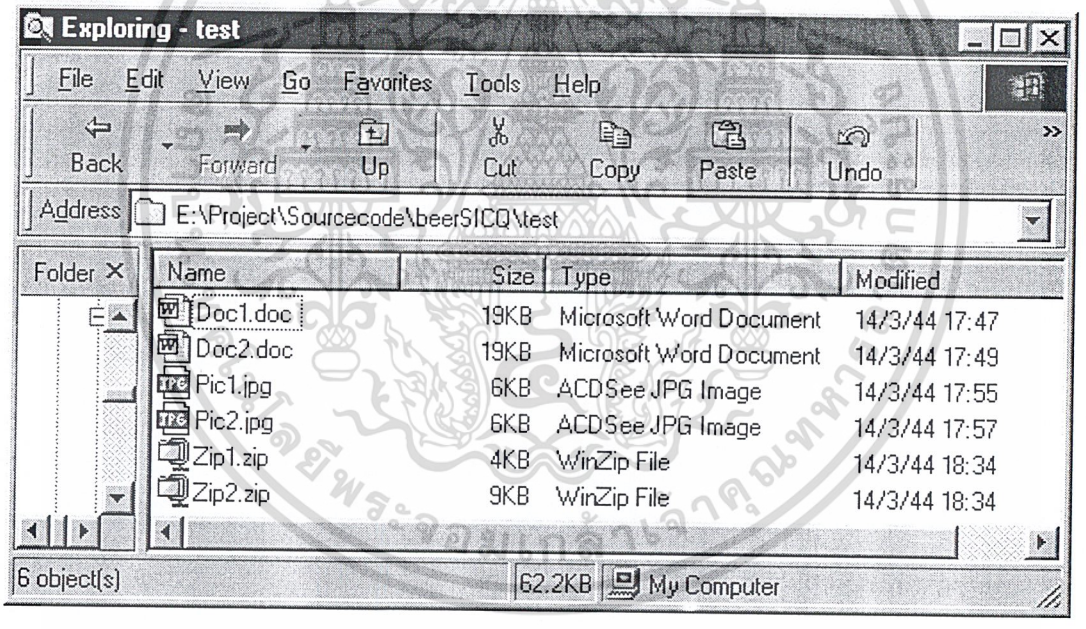
รูปที่ 6-6 แสดงหน้าต่างแสดงการดักจับข้อความที่เข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6-6 จะเห็นว่าข้อความที่ดักจับได้ระหว่างผู้รับและผู้ส่งไม่สามารถอ่านได้เพราะผ่านการเข้ารหัสมาแล้วซึ่งการถอดรหัสข้อความนี้ค่อนข้างทำได้ยากทำให้มีความปลอดภัยในการส่งข้อมูลที่มีความสำคัญหรือเป็นความลับ ส่วนในรูปที่ 6-5 แสดงหน้าต่างของผู้รับซึ่งจะเห็นว่าข้อความที่ได้รับมาเหมือนกับข้อความจากผู้ส่งแสดงให้เห็นว่าผู้ที่รู้คีย์ในการถอดรหัสจะสามารถถอดรหัสข้อความกลับมาได้เหมือนเดิมทุกประการ

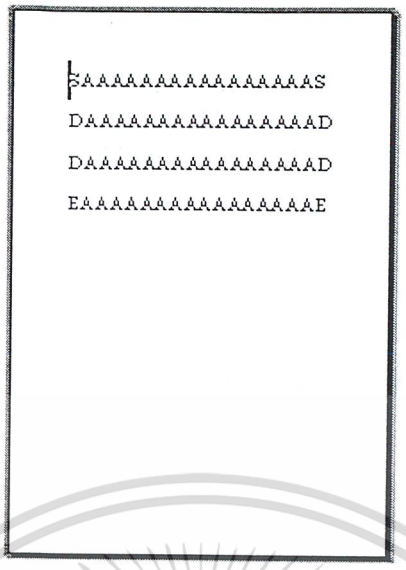
6.3 การทดสอบรับส่งไฟล์แบบปรกติ

ในการรับส่งไฟล์ระหว่างไคลเอนต์จะต้องทำตามขั้นตอนของ File Direct Protocol ของ ICQ ซึ่งการส่งแต่ละไฟล์นั้นจะต้องแบ่งไฟล์ออกเป็นส่วนๆ โดยแต่ละส่วนมีขนาดไม่เกิน 2048 ไบต์ซึ่งถ้าไฟล์มีขนาดเล็กกว่า 2048 ไบต์ก็จะถูกส่งในแพ็กเก็ตเดียว สำหรับการทดสอบการรับส่งไฟล์จะทดสอบโดยการส่งไฟล์บีบอัดสองไฟล์ซึ่งในไฟล์แรกประกอบด้วยรูปสองรูปและไฟล์ที่สองประกอบด้วยไฟล์ข้อความสองไฟล์เมื่อฝ่ายรับได้รับไฟล์จนครบแล้วก็ทดลองคลายการบีบอัดไฟล์ออกมาดูว่าไฟล์ที่รับมาได้มีความสมบูรณ์เหมือนกับไฟล์ต้นทางหรือไม่โดยไฟล์ที่ใช้ในการทดสอบแสดงดังรูปที่ 6-7 ถึง 6-16

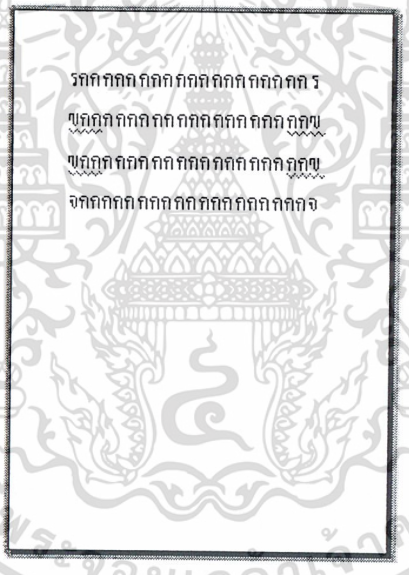


รูปที่ 6-7 แสดงขนาดของไฟล์ต้นทาง

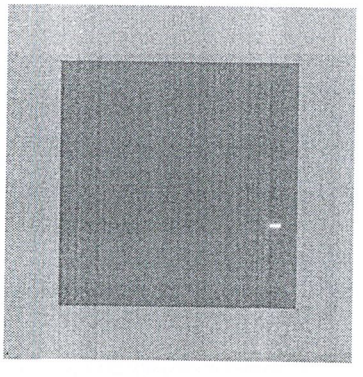
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-8 แสดงไฟล์ Doc1.doc ต้นทาง

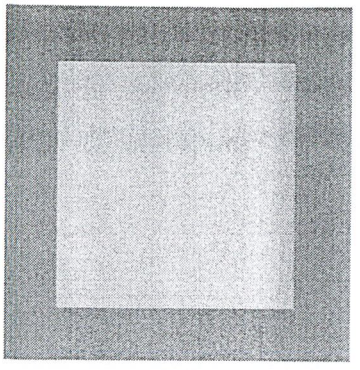


รูปที่ 6-9 แสดงไฟล์ Doc2.doc ต้นทาง

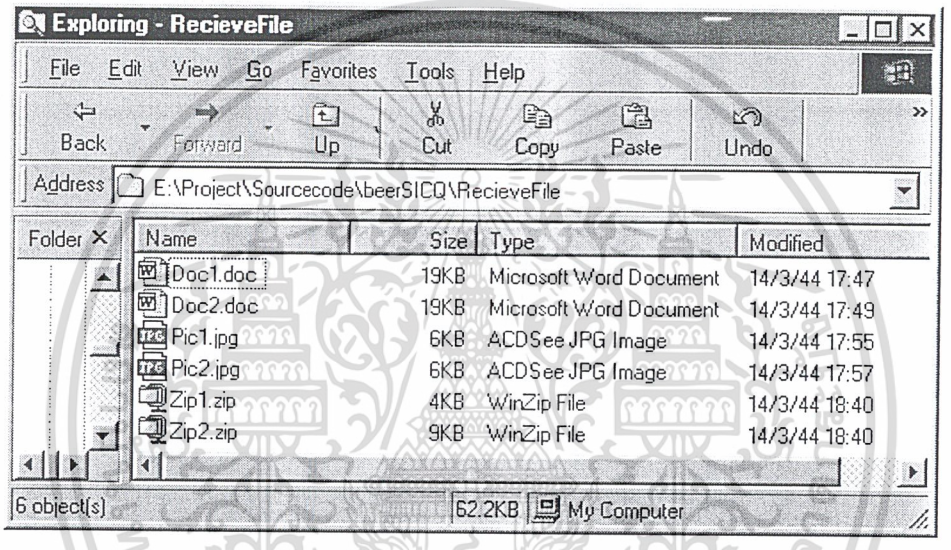


รูปที่ 6-10 แสดงไฟล์ Pic1.jpg ต้นทาง

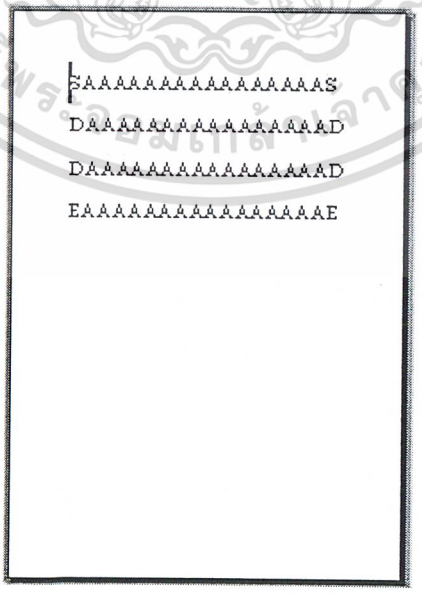
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-11 แสดงไฟล์ Pic2.jpg ต้นทาง

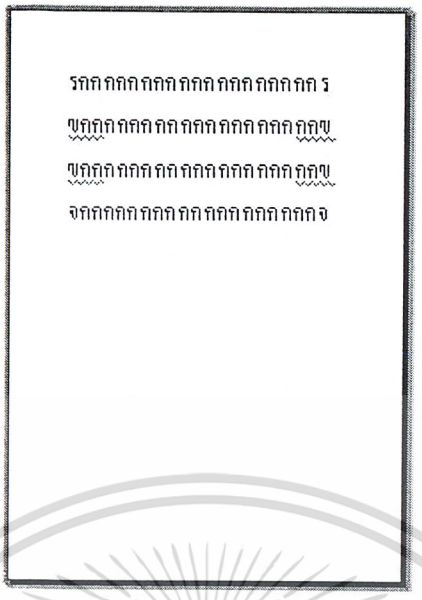


รูปที่ 6-12 แสดงขนาดของไฟล์ปลายทาง



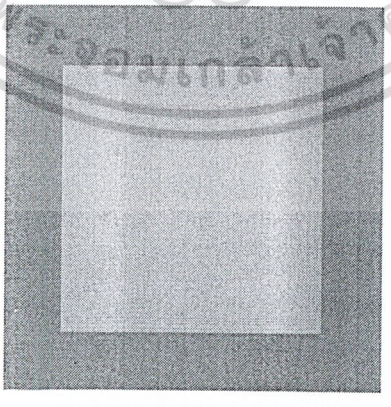
รูปที่ 6-13 แสดงไฟล์ Doc1.doc ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-14 แสดงไฟล์ Doc2.doc ปลายทาง

รูปที่ 6-15 แสดงไฟล์ Pic1.jpg ปลายทาง



รูปที่ 6-16 แสดงไฟล์ Pic2.jpg ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าไฟล์ต้นทางและปลายทางเหมือนกันทุกประการแสดงว่าการรับส่งไฟล์แบบปรกติสามารถส่งได้สมบูรณ์ไม่มีปัญหาในการส่ง

6.4 การทดสอบรับส่งไฟล์แบบเข้ารหัส

เพื่อความปลอดภัยในการรับส่งไฟล์มากขึ้นจึงได้มีการเข้ารหัสไฟล์ก่อนการส่งด้วยสำหรับผู้ที่ใช้โปรแกรม IsagQ ด้วยกันซึ่งในการเข้ารหัสจะทำเป็นบล็อกตามขนาดที่ส่งในแต่ละครั้งผลกระทบจากการเข้ารหัสในการส่งไฟล์ก็คือการส่งจะช้ากว่าการส่งแบบปรกติพอสมควรเพราะต้องเสียเวลาในการเข้ารหัสทุกบล็อกก่อนการส่งซึ่งไฟล์ที่ใช้ในการทดสอบจะใช้ไฟล์เดียวกันกับการส่งแบบปรกติและผลลัพธ์ของการทดสอบปรากฏว่าการรับส่งไฟล์แบบเข้ารหัสสามารถทำได้สมบูรณ์เหมือนการส่งแบบปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการพัฒนาโครงงาน

7.1 สรุปผลการดำเนินงาน

ในการพัฒนาโครงงานนี้มีปัญหาเกี่ยวกับโพรโตคอลที่ใช้ในการสื่อสารของโปรแกรมเพราะต้องการให้ใช้งานได้กับโปรแกรม ICQ และใช้เซิร์ฟเวอร์ของ ICQ ด้วยจึงต้องใช้โพรโตคอลของ ICQ ในการติดต่อสื่อสารของโปรแกรมซึ่งข้อมูลเกี่ยวกับโพรโตคอลของ ICQ นั้นสามารถหาได้ในเว็บไซต์ต่างๆ แต่ว่าข้อมูลที่หามาได้นั้นยังไม่ครบถ้วนชัดเจนเท่าที่ควรทำให้การทำงานค่อนข้างล่าช้าเพราะต้องทำความเข้าใจกับโพรโตคอลของ ICQ ในส่วนที่จำเป็นต้องนำมาใช้งาน

สำหรับภาษาที่นำมาใช้ในการพัฒนาก็คือภาษาจาวาซึ่งข้อดีที่เห็นได้ชัดของภาษานี้ก็คือง่ายต่อการโปรแกรมคือมีคลาสสำหรับจัดการงานที่ยุ่งยากเอาไว้ให้ใช้มากมายทำให้ช่วยลดเวลาในการพัฒนาลงไปพอสมควรและสามารถทำงานได้บนหลายแพลตฟอร์มแต่ภาษาจาวาก็มีปัญหาบางอย่างเช่นในขณะที่โปรแกรมทำงานจะค่อนข้างใช้ทรัพยากรมากกว่าโปรแกรมที่เขียนจากภาษาอื่นและโปรแกรมที่สามารถใช้งานได้บน JDK เวอร์ชันเก่าอาจมีปัญหาในการใช้งานใน JDK เวอร์ชันที่ใหม่กว่าได้

หลังจากได้พัฒนาโครงงานนี้จนสำเร็จและทำการทดสอบโปรแกรมที่ได้สรุปว่าโปรแกรมที่ได้สามารถทำงานได้อยู่ในเกณฑ์ที่น่าพอใจและมีคุณสมบัติต่างๆตรงตามที่ได้ตั้งเป้าหมายเอาไว้คือสามารถส่งข้อความและไฟล์ได้และมีการเข้ารหัสข้อมูลสำหรับการติดต่อระหว่างโปรแกรม IsagQ ซึ่งในการทดสอบการทำงานพบว่าโปรแกรมมีเสถียรภาพในการทำงานที่ค่อนข้างดี

7.2 คุณสมบัติของโปรแกรมที่ได้จากการพัฒนา

7.2.1 ใช้เมาส์และคีย์บอร์ดในการควบคุมการทำงานของโปรแกรม

7.2.2 แสดงผลเป็นกราฟฟิกทางจอมอนิเตอร์

7.2.3 สามารถเพิ่มชื่อผู้ใช้งานสำหรับไคลเอนต์ได้

7.2.4 สามารถเปลี่ยนผู้ใช้งานได้

7.2.5 ต้องใส่พาสเวิร์ดก่อนการใช้งานโปรแกรม

7.2.6 แสดงสถานะของผู้ใช้งานเป็นไอคอน

7.2.7 ค้นหาผู้ใช้อื่นๆได้โดยใช้หมายเลข UIN, ชื่อจริง, ชื่อเล่น, นามสกุล หรือ อีเมล์

7.2.8 เพิ่มชื่อผู้ใช้อื่นเข้าคอนแทคลิสต์ได้

7.2.9 สามารถส่งข้อความและไฟล์ได้

7.2.10 มีการเข้ารหัสและถอดรหัสสำหรับการสื่อสารระหว่างโปรแกรม IsagQ ด้วยกัน

7.2.11 ใช้งานภาษาไทยได้

7.2.12 ใช้งานได้บนหลายแพลตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ข้อจำกัดของโปรแกรม

7.3.1 มีการทำงานที่ช้ากว่าโปรแกรม ICQ

7.3.2 ใช้ทรัพยากรในการทำงานมากเครื่องที่มีสเปกต่ำอาจมีการทำงานของโปรแกรมที่ไม่ดีนัก

7.3.3 การทำงานของโปรแกรมใน WindowsNT ค่อนข้างช้า

7.3.4 มีปัญหาในการแสดงผลสำหรับอักษรบางตัว

7.3.5 ต้องใช้งานโปรแกรมบน JDK 1.2.2_th เท่านั้น

7.4 แนวทางในการพัฒนาต่อ

7.4.1 พัฒนาให้รองรับโพรโตคอลเวอร์ชัน 5 เพราะรองรับการทำงานที่มากกว่า และในการทำงานบางส่วนเช่น การขอข้อมูลสามารถทำได้เร็วกว่าในเวอร์ชัน 2

7.4.2 โปรแกรมควรจะลงทะเบียนกับเซิร์ฟเวอร์ของ ICQ เพื่อขอเบอร์ ICQ ใหม่ได้

7.4.3 เพิ่มเติมฟังก์ชันการทำงานต่างๆในส่วนของไคลเอนต์เพื่อเพิ่มความสะดวกสำหรับผู้ใช้งาน เช่น การส่งข้อความเดียวไปให้หลายๆ คน, การเพิ่ม history สำหรับผู้ใช้ในคอนแทคลิสต์แต่ละคน เป็นต้น

7.4.4 สามารถนำคาต้าเบสของ ICQ มาใช้ได้เพื่อจะได้ไม่ต้องเพิ่มชื่อในคอนแทคลิสต์ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Elliotte Rusty Harold : "Java Network Programming" O'Reilly, 1997.

[2] ดร.วีระศักดิ์ ซึงถาวร : "Fundamental of Java Programming Volume 1 and 2" Sum Publishing, 1998.

[3] Magnus Ihse : "The ICQ Protocol Version 0.91" http://www.student.nada.kth.se/~d95-mih/icq/, 1998.

[4] Henrik Isaksson : "Version 5 of ICQ Protocol" http://www.algonet.se/~henisak/icq/icqv5.html, 2000.

[5] Graham Klyne : "Instant Messaging and Presence Protocol" IETF, 2000.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้