ระบบไคลเอ็นต์และเซิร์ฟเวอร์สำหรับส่งสารด่วนแบบปลอดภัย

SECURE INSTANT MESSENGER CLIENT / SERVER SYSTEM

นายรชฎ ชะมุนี

นางสาวรัตพร สุทธิมัณฑนกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ระบบไคลเอ็นต์และเซิร์ฟเวอร์สำหรับส่งสารด่วนแบบปลอดภัย

SECURE INSTANT MESSENGER CLIENT / SERVER SYSTEM

โดย

นายรชฎ ชะมุนี เลขประจำตัว 45010623

นางสาวรัตพร สุทธิมัณฑนกุล เลขประจำตัว 45010649

อาจารย์ที่ปรึกษา

ผศ.ธนา พงษ์สุวรรณ

อ.อัครเดช วัชระภูพงษ์

อ.ธนัญชัย ตรีภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบไคลเอ็นต์และเซิร์ฟเวอร์สำหรับส่งสารด่วนแบบปลอดภัย

Secure Instant Messenger Client / Server System

ผู้จัดทำ

1. นายรชฎ ชะมุนี รหัสนักศึกษา 45010623

2. นางสาวรัตพร สุทธิมัณฑนกูล รหัสนักศึกษา 45010649

_____ อาจารย์ที่ปรึกษา

(ผศ. ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อาจารย์ อัครเดช วัชระกูพงษ์)

_____ อาจารย์ที่ปรึกษา

(อาจารย์ ธนัญชัย ตรีภาค)

ระบบไคลเอ็นต์และเซิร์ฟเวอร์สำหรับส่งสารด่วนแบบปลอดภัย

รชฎ ชะมุนี	45010623
รัตพร สุทธิมัณฑนกุล	45010649
ผศ. ธนา หงษ์ สุวรรณ	อาจารย์ที่ปรึกษา
อ. อัครเดช วัชระภูพงษ์	อาจารย์ที่ปรึกษา
อ. ธนัญชัย ตรีภาค	อาจารย์ที่ปรึกษา
ปีการศึกษา 2548	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอกระบวนการพัฒนาโครงงานคือ สร้างส่วนขยายของโปรแกรม GAIM (IsagQ) และโปรแกรมแม่ข่าย(IsagMQ)เพื่อทำการออกใบรับรองสิทธิ์ให้แก่โปรแกรมลูกข่ายซึ่ง พัฒนาด้วยภาษาซี เพื่อตอบสนองด้านความปลอดภัย 2 ประการหลัก คือ การพิสูจน์ตน เพื่อสร้างความมั่นใจให้กับผู้ใช้ว่ากำลังติดต่อกับบุคคลที่ต้องการจริงๆและการเข้าและถอดรหัสเพื่อปิดบังข้อมูลที่ถูกส่งผ่านเครือข่าย

IsagMQ สามารถให้บริการพื้นฐานของ Certificate Authority ได้เช่น การออกใบรับรองสิทธิ์ให้กับผู้ใช้ การจัดทำฐานข้อมูลของผู้ใช้ที่ลงทะเบียนแล้ว การเรียกดูใบรับรองสิทธิ์ได้ และสามารถถอดถอนใบรับรองสิทธิ์ได้ ส่วนIsagQ สามารถลงทะเบียนกับIsagMQอัตโนมัติ

SECURE INSTANT MESSENGER CLIENT / SERVER SYSTEM

Rachod	Chamunee	45010623
Rattaporn	Suttimantanakul	45010649
Asst. Thana	Hongsuwan	Advisor
Mr. Akkradash	Watcharapupong	Advisor
Thananchai	Treepark	Advisor

Academic Year 2004

ABSTRACT

This thesis present a development of project ,GAIM application extension(IsagQ) and server (IsagMQ) that publish a certificate to client This project development in c and has 2 main security purposes including authentication to ensure the user is communicate to the rightperson and data encryption and decryption to protect data that send on network

IsagMQ can serve as certificate authority such as publish certificate to user, creat database of registered user,show certificate IsagQ is automatically register with IsagMQ

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษา ผศ. ธนาหงษ์ สุวรรณ อ. ศักรเดช วัชระภูพงษ์ อ. ธนัญชัย ตรีภาค ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านอาจารย์ทั้งสามท่านและขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณห้องปฏิบัติการ ISAG ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ได้สนับสนุนสถานที่และอุปกรณ์เครือข่ายสำหรับใช้ในการพัฒนาโครงงาน

ขอขอบคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

รชฎ ชะมุนี 45010623

รัตพร สุทธิมัณฑนกุล 45010649

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

โปรแกรมรับส่งสารด่วน เป็นอีกตัวอย่างหนึ่งของความนิยมบนอินเทอร์เน็ตในขณะนี้ ซึ่งโปรแกรมดังกล่าวตอบสนองการให้บริการรับส่งสารได้อย่างทันท่วงทีได้ทุกเวลาและทุกที่ เพื่อการบันเทิงหรือทางธุรกิจโดยเป็นรูปแบบของการสื่อสารที่มีความประหยัดเป็นอย่างมากซึ่งนับได้ว่าเป็นคลื่นลูกใหม่ที่มาแรงแทนที่ความนิยมของ จดหมายอิเล็กทรอนิกส์ อย่างสมบูรณ์แต่เมื่อมองในเรื่องความปลอดภัยเป็นหลักโปรแกรมนี้มิได้คำนึงถึงเรื่องความปลอดภัยไม่ว่าจะเป็นเรื่องการเข้าและถอดรหัสหรือแม้กระทั่งการพิสูจน์ตนของผู้ใช้รวมถึงการขาดความเข้าใจในโปรแกรมของผู้ใช้งานยิ่งพิจารณาถึงปัญหาอาชญากรบนอินเทอร์เน็ตรวมด้วยแล้วจะเห็นได้ว่าปัญหาเรื่องความปลอดภัยเป็นอีกมุมมองหนึ่งที่ควรคำนึงถึงอย่างมากเพื่อสนองความต้องการของผู้ใช้งานได้อย่างครบครัน

ทางผู้พัฒนาจึงได้พัฒนาโครงงาน 2 โครงงานขึ้นมาซึ่งโครงงานแรกเป็นส่วนของการสร้างโปรแกรมแม่ข่ายสำหรับรับส่งสารด่วนแบบปลอดภัย (Secure Instant Messaging Server: IsagMQ) พัฒนาด้วยภาษาซีบนระบบปฏิบัติการตระกูลยูนิกซ์เพื่อทำการออกใบรับรองให้แก่โปรแกรมลูกข่ายและโครงงานที่สองเป็นส่วนของการสร้างโปรแกรมปลั๊กอิน (Secure Instant Messenger: IsagQ) เพื่อความปลอดภัยให้กับโปรแกรมลูกข่ายเกี๊ยม (GAIM) ซึ่งพัฒนาด้วยภาษาซี เพื่อตอบสนองด้านความปลอดภัย 2 ประการหลัก คือ การพิสูจน์ตน เพื่อสร้างความมั่นใจให้กับผู้ใช้ว่ากำลังติดต่อยู่กับบุคคลที่ต้องการจริงๆและการเข้าและถอดรหัสเพื่อปิดบังข้อมูลที่ถูกส่งผ่านเครือข่าย

1.2 วัตถุประสงค์ของโครงงาน

1.1.1. เพื่อศึกษาการทำงานของระบบรับส่งสารด่วน

1.1.2. เพื่อเพื่อปรับปรุงระบบรับส่งสารด่วนให้มีความปลอดภัยยิ่งขึ้น

1.1.3. เพื่อสร้างโปรแกรมที่สามารถเข้าและถอดรหัสลับข้อความเพื่อทำให้ผู้ใช้มั่นใจได้ว่าข้อความที่ส่งถูกปกปิดเป็นความลับ

1.1.4. เพื่อสร้างโปรแกรมที่สามารถใช้ระบบพิสูจน์ตนเพื่อทำให้ผู้ใช้มั่นใจได้ว่ากำลังติดต่อกับบุคคลที่ต้องการติดต่อจริง

1.1.5. เพื่อสร้างการเชื่อมต่อระหว่างโปรแกรมฝั่งเซิร์ฟเวอร์และฝั่งไคลเอ็นต์โดยสร้างเป็นส่วนขยายของโปรแกรมGaim โดยยังคงความสามารถตามข้อ 1.2.3และ1.2.4

1.3 ขอบเขตของโครงงาน

ในปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการเพิ่มความปลอดภัยในด้านการรักษาความลับของข้อมูล และ การพิสูจน์ตัวตน โดยพัฒนา 2 ส่วนคือ ส่วนขยายของโปรแกรม Gaim (IsagQ)และโปรแกรมแม่ข่าย (IsagMQ) ซึ่งระบบสามารถทำงานได้บนทุกระบบปฏิบัติการโดย IsagMQ สามารถให้บริการพื้นฐานของ Certificate Authority ได้เช่น การออกใบรับรองสิทธิ์ให้กับผู้ใช้ การจัดทำฐานข้อมูลของผู้ใช้ที่ลงทะเบียนแล้ว การเรียกดูใบรับรองสิทธิ์ได้ และ สามารถถอดถอนใบรับรองสิทธิ์ได้ ส่วนIsagQ สามารถลงทะเบียนกับIsagMQอัตโนมัติ โดยสามารถกำหนดขนาดกุญแจส่วนตัวได้เมื่อขอใบรับรองสิทธิ์สามารถพิสูจน์ใบรับรองสิทธิ์ของผู้สนทนาที่ติดตั้ง IsagQได้ สามารถเข้ารหัสข้อมูลแบบกุญแจสาธารณะระหว่างไคลเอนต์ด้วยกัน สามารถขอใบรับรองสิทธิ์ได้เมื่อใบรับรองสิทธิ์หมดอายุและสามารถยกเลิกการลงทะเบียนได้

1.4 วิธีการดำเนินการ

1. วิเคราะห์และออกแบบระบบ
2. ศึกษาโครงสร้างโปรแกรมสนทนา
3. ศึกษาโปรโตคอลเอสเอสแอล ซึ่งนำมาใช้เพิ่มความปลอดภัยให้กับระบบ
4. ศึกษาโครงสร้างกุญแจสาธารณะและการออกใบรับรองสิทธิ์
5. ศึกษาไลบรารี่โอเพนเอสเอสแอล
6. ศึกษาโปรแกรมเกี๊ยม
7. ศึกษาการเขียนจีทีเคพลัส
8. ติดตั้งระบบปฏิบัติการและสภาพแวดล้อมที่จำเป็นต้องใช้ในการพัฒนา
9. พัฒนาส่วนขยายของโปรแกรมเกี๊ยม
10. พัฒนาเซิร์ฟเวอร์และสร้างผู้ออกใบรับรองสิทธิ์
11. สร้างการเชื่อมต่อบนโปรโตคอลเอสเอสแอล ติดตั้งการเข้ารหัสและการพิสูจน์ตัวตน
12. วิเคราะห์ผลของระบบที่ได้ทำการพัฒนาขึ้นและแก้ไขส่วนที่ผิดพลาดเพื่อให้สามารถสร้างระบบที่ใช้งานได้อย่างมีประสิทธิภาพมากที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้ความเข้าใจเกี่ยวกับการเข้ารหัสและถอดรหัส
2. ได้รับความรู้ความเข้าใจเกี่ยวกับการระบุตัวตน
3. ได้รับความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมด้วยภาษาซี
4. ได้รับความรู้ความเข้าใจเกี่ยวกับการใช้งานระบบปฏิบัติการตระกูลยูนิกซ์
5. ระบบไคลเอ็นต์และเซิร์ฟเวอร์ที่ติดต่อกันโดยใช้ระบบพิสูจน์ตัวตนรวมถึงการเข้าและถอดรหัส

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการพัฒนาโครงงาน ประกอบด้วย โครงสร้างระบบไคลเอนท์และเซิร์ฟเวอร์สำหรับรับส่งสารด่วน นิยามความมั่นคงปลอดภัยคอมพิวเตอร์ ทฤษฎีสำหรับการส่งข้อมูลบนอินเตอร์เนทให้ปลอดภัยด้วย SSL ทฤษฎีใบรับรองสิทธิ์และทฤษฎีการสร้างผู้ให้บริการใบรับรองสิทธิ์

บทที่ 3 กล่าวถึงการออกแบบซอฟต์แวร์IsagQ และIsagMQ ในปี2547 และ ปี2548 และความแตกต่างระหว่าง IsagMQ และ IsagQ ปีการศึกษา 2547 กับ 2548

บทที่ 4 กล่าวถึงการพัฒนาชิ้นงานของโครงงาน ตั้งแต่การเตรียมสภาพแวดล้อมของทั้งส่วนเซิร์ฟเวอร์ IsagMQ และส่วนไคลเอนต์ IsagQ (ส่วนขยายบน Gaim) การนำโอเพนเอสแอลไลบรารี (OpenSSL Library) มาใช้ในโครงงาน การพัฒนาส่วนขยาย IsagQ กระบวนการรับ-ส่งข้อความของ IsagQ การสร้างผู้ออกใบรับรองสิทธิ์และกระบวนการจัดการใบรับรองสิทธิ์

บทที่ 5 กล่าวถึงการทดลองและผลการทดลองของระบบไคลเอนท์และเซิร์ฟเวอร์ โดยมีการทดสอบการส่งการร้องขอใบรับรองสิทธิ์ระหว่าง IsagQ และ IsagMQ ได้แก่การทดสอบ IsagQ สร้างใบร้องขอใบรับรองสิทธิ์ การทดสอบIsagQ และ IsagMQ สร้างการเชื่อมต่อแบบSSL การทดสอบ IsagMQ สร้างใบรับรองสิทธิ์ และการทดสอบให้ IsagMQ การตรวจใบรับรองสิทธิ์

บทที่ 6 บทนี้กล่าวถึงบทวิจารณ์และบทสรุปของโครงงานนี้ ปัญหาอุปสรรคและแนวทางแก้ไขและแนวทางการพัฒนาต่อ

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

ในบทนี้จะกล่าวถึง โครงสร้างของโปรแกรมแม่ข่ายและลูกข่ายสำหรับรับส่งสารด่วน นิยามความมั่นคงปลอดภัยของคอมพิวเตอร์ และ ทฤษฎีการสร้างผู้ให้บริการใบรับรองสิทธิ์ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา และประเมินประสิทธิภาพของโครงงาน

2. 1 โปรแกรมแม่ข่ายและลูกข่ายสำหรับรับส่งสารด่วน

ในระบบรับส่งสารด่วนจะแบ่งการทำงานเป็น 2 ส่วนหลักคือ

1. โปรแกรมแม่ข่ายสำหรับรับส่งสารด่วน(Instant Messaging Server)
2. โปรแกรมรับส่งสารด่วนฝั่งไคลเอ็นต์ (Instant Messaging Client)

โดยทั่วไปทุกครั้งก่อนที่ไคลเอ็นต์จะใช้บริการ ไคลเอ็นต์จำเป็นต้องทำการล๊อกอินติดต่อไปยังโปรแกรมแม่ข่ายก่อนเสมอเนื่องจากโปรแกรมแม่ข่ายต้องการข้อมูลที่จำเป็นเบื้องต้นหลายๆ อย่างก่อนที่จะเริ่มการให้บริการแก่ไคลเอ็นต์ได้หากการล๊อกอินสำเร็จอย่างสมบูรณ์โปรแกรมแม่ข่ายจะสามารถให้บริการด้านการส่งข้อความแก่ไคลเอ็นต์

หน้าที่ของโปรแกรมแม่ข่ายจะมากหรือน้อยขึ้นอยู่กับการออกแบบรูปแบบการติดสื่อสาร รวมถึงความสามารถต่างๆที่โปรแกรมลูกข่ายต้องการ ซึ่งถ้ายิ่งมากแล้ว โปรแกรมแม่ข่ายก็ยิ่งมีหน้าที่เพิ่มมากขึ้น

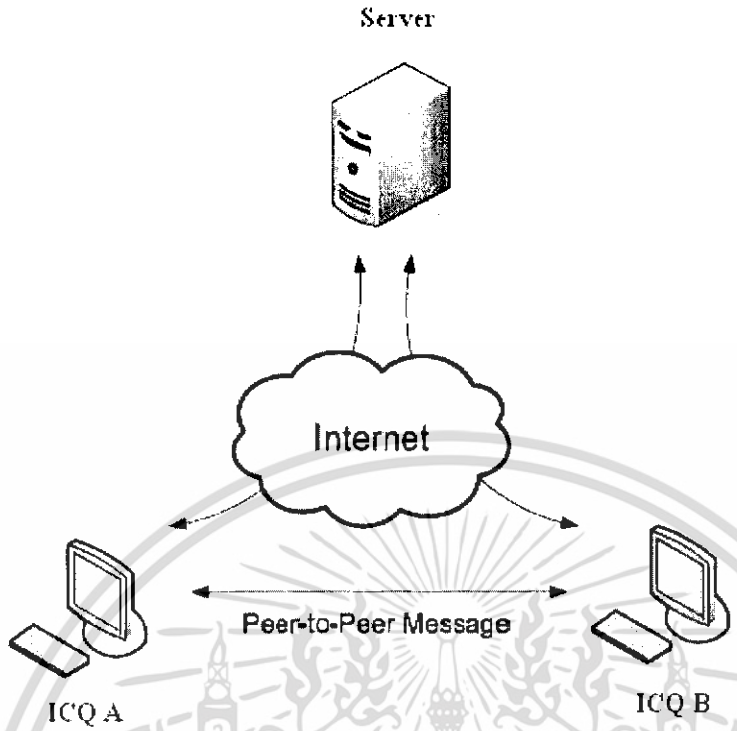
ชนิดของโปรแกรมรับส่งสารด่วนในปัจจุบัน

ในปัจจุบันระบบส่งสารด่วนแบ่งได้เป็น 2 ชนิดดังนี้

1. แบบเพียร์ทูเพียร์ (Peer-to-Peer Message)

การทำงานของระบบส่งสารด่วนแบบนี้ คือ เมื่อผู้ใช้ต้องการใช้บริการก็จะทำการล๊อกอินไปยังเซิร์ฟเวอร์ของโปรแกรมรับส่งสารด่วนเพื่อรับส่งรายละเอียดต่างๆ และเพื่อให้เซิร์ฟเวอร์เซิร์ฟเวอร์ทราบว่าขณะนี้มีผู้ใช้คนใดที่อยู่ในสถานะพร้อมให้บริการเมื่อทำการแลกเปลี่ยนข้อมูลเสร็จสิ้นแล้ว เซิร์ฟเวอร์ก็จะแจ้งไปยังคู่สนทนาของผู้ใช้

หลังจากนั้นผู้ใช้จะสามารถติดต่อไปยังคู่สนทนาได้โดยตรงโดยที่ไม่ผ่านเซิร์ฟเวอร์ซึ่งรูปแบบนี้เป็นลักษณะแบบเพียร์ทูเพียร์นั้นเองตัวอย่างของโปรแกรมรับส่งสารด่วนประเภทนี้ก็คือ ICQ



รูปที่ 2.1 ระบบส่งสารด่วนแบบเพียร์ทูเพียร์

2. แบบเซิร์ฟเวอร์ทำงานเป็นหลัก (Server-based Communication)

การทำงานของโปรแกรมรับส่งสารด่วนแบบนี้จะเริ่มต้นแบบเพียร์ทูเพียร์ต่างกันที่เซิร์ฟเวอร์จะส่งเพียงชื่อหรือสถานะการออนไลน์หรืออฟไลน์ของคู่สนทนาเท่านั้นของเซิร์ฟเวอร์จะเห็นได้ว่าวิธีนี้จะเป็นลักษณะรวมอำนาจเข้ามาไว้ที่ศูนย์กลาง(centralization)ผู้ใช้นอกจากจะต้องล๊อกอินไปยังเซิร์ฟเวอร์แล้วทุกๆข้อความที่ผู้ใช้ต้องการส่งไปให้ใครก็จำเป็นที่จะต้องส่งไปยังเซิร์ฟเวอร์เพื่อให้เซิร์ฟเวอร์ทำการส่งข้อความต่อไปยังผู้ใช้ปลายทางอีกทีนึง

ด้วยวิธีการนี้เซิร์ฟเวอร์จะต้องมีความสามารถนอกจากการให้บริการพื้นฐานแล้วยังต้องสามารถจัดลำดับการรับส่งข้อความของผู้ใช้ณเวลาจริงโดยข้อความใดมาก่อนก็จะได้รับการส่งไปยังปลายทางก่อนจะเห็นได้ว่าโปรแกรมทางฝั่งแม่ข่ายจะมีความซับซ้อนสูงมากตัวอย่างของระบบรับส่งสารด่วนประเภทนี้ก็คือ MSN Messenger

การเปรียบเทียบข้อดีข้อเสียของทั้งสองระบบ

ระบบส่งสารด่วนทั้งแบบเพียร์ทูเพียร์และแบบเซิร์ฟเวอร์ทำงานเป็นหลักต่างมีข้อดีข้อเสียแตกต่างกันไปซึ่งสามารถสรุปได้ดังนี้

ความสามารถ	เพียร์ทูเพียร์	แบบเซิร์ฟเวอร์
เซิร์ฟเวอร์ทำงานหนัก	มาก	น้อย
ความรวดเร็วในการส่งข้อความ	น้อย	น้อย
ความซับซ้อนของโพรโตคอล	น้อย	มาก
ความปลอดภัย	น้อย	มาก
ปัญหาเรื่องไฟล์วอลล์	มาก	น้อย
ค่าใช้จ่ายในการดูแลรักษา	น้อย	มาก

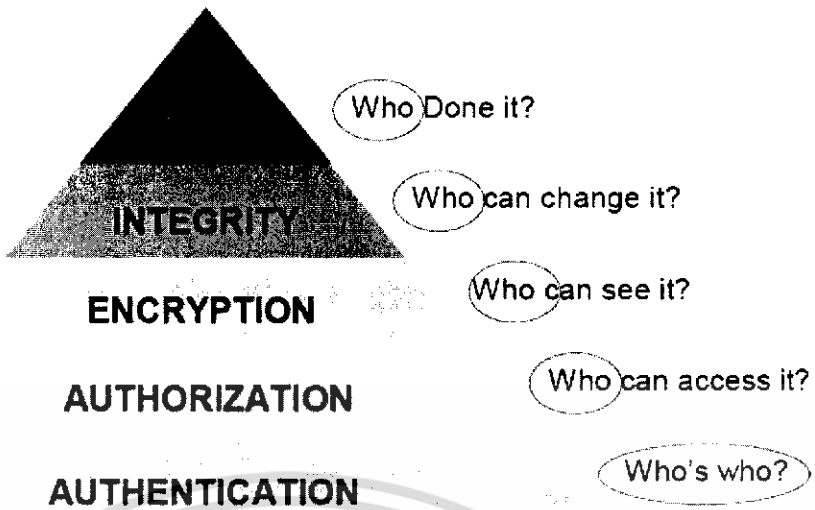
ตารางที่ 2.1 เปรียบเทียบคุณสมบัติระบบส่งสารด่วนแบบเพียร์ทูเพียร์และแบบเซิร์ฟเวอร์ทำงานเป็นหลัก

2.2 นิยามความมั่นคงปลอดภัยคอมพิวเตอร์

ในปัจจุบันระบบคอมพิวเตอร์ได้ถูกคุกคามมากขึ้นทั้งจากไวรัสคอมพิวเตอร์หรือจากผู้ไม่ประสงค์ดีซึ่งความมั่นคงปลอดภัยคอมพิวเตอร์(Computer Security) ช่วยปกป้องเครื่องคอมพิวเตอร์รวมไปถึงอุปกรณ์ต่างๆที่เกี่ยวข้องและที่สำคัญยังสามารถช่วยปกป้องข้อมูลที่ได้จัดเก็บไว้ภายในระบบหรือใช้ในความหมายความปลอดภัยทางข้อมูลสารสนเทศ(Information Security)ก็ได้จุดประสงค์หลักของความปลอดภัยทางข้อมูลคือ ความลับ (Confidentiality) ความสมบูรณ์ (Integrity) ความพร้อมใช้ (Availability) และการห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation) โดยมีรายละเอียดดังนี้

- การรักษาความลับ (Confidentiality) คือการรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับและผู้มีสิทธิเท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้
- การรักษาความสมบูรณ์ (Integrity) คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไม่ว่าจะเป็นโดย อุบัติเหตุหรือโดยเจตนา
- ความพร้อมใช้ (Availability) คือการรับรองว่าข้อมูลและบริการการสื่อสารต่าง ๆ พร้อมที่จะใช้ได้ในเวลาที่ต้องการใช้งาน
- การห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation) คือวิธีการสื่อสารซึ่งผู้ส่งข้อมูลได้รับหลักฐานว่าได้มีการส่งข้อมูลแล้วและผู้รับก็ได้รับการยืนยันว่าผู้ส่งเป็นใคร ดังนั้นทั้งผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่มีความเกี่ยวข้องกับข้อมูลดังกล่าวในภายหลัง

ในทางปฏิบัติสามารถกำหนดลักษณะของการควบคุมความมั่นคงปลอดภัย (Security Control) ได้ 5 ระดับดังรูป



รูปที่ 2.2 ลักษณะของการควบคุมความมั่นคงปลอดภัย (Security Control)

1.การพิสูจน์ตัวตน(Authentication)

การพิสูจน์ตัวตน คือขั้นตอนการยืนยันความถูกต้องของหลักฐาน (Identity) ที่แสดงว่าเป็นบุคคลที่กล่าวอ้างจริง ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

- การระบุตัวตน (Identification) คือขั้นตอนที่ผู้ใช้แสดงหลักฐานว่าตนเองคือใครเช่น ชื่อผู้ใช้ (username)
- การพิสูจน์ตัวตน (Authentication) คือขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลที่กล่าวอ้างจริง

หลักฐานที่ผู้ใช้นำมากล่าวอ้างที่เกี่ยวกับเรื่องของความปลอดภัยนั้นสามารถจำแนกได้ 2 ชนิด

- Actual identity คือหลักฐานที่สามารถบ่งบอกได้ว่าในความเป็นจริงบุคคลที่กล่าวอ้างนั้นเป็นใคร
- Electronic identity คือหลักฐานทางอิเล็กทรอนิกส์ซึ่งสามารถบ่งบอกข้อมูลของบุคคลนั้นได้ แต่ละบุคคลอาจมีหลักฐานทางอิเล็กทรอนิกส์ได้มากกว่า 1 หลักฐาน ตัวอย่างเช่น บัญชีชื่อผู้ใช้

กลไกของการพิสูจน์ตัวตน (Authentication mechanisms) สามารถแบ่งออกได้เป็น 3 คุณลักษณะคือ

- สิ่งที่คุณมี (Possession factor) เช่น กุญแจหรือเครดิตการ์ด เป็นต้น
- สิ่งที่คุณรู้ (Knowledge factor) เช่น รหัสผ่าน (passwords) หรือการใช้พิน (PINs) เป็นต้น

- สิ่งที่คุณเป็น (Biometric factor) เช่น ลายนิ้วมือ รูปแบบเรตินา (retinal patterns) หรือใช้รูปแบบเสียง (voice patterns) เป็นต้น

กระบวนการพิสูจน์ตัวตนนั้นจะนำ 3 ลักษณะข้างต้นมาใช้ในการยืนยันหลักฐานที่นำมากล่าวอ้างทั้งนี้ขึ้นอยู่กับระบบวิธีการที่นำมาใช้เพียงอย่างใดอย่างหนึ่ง (Single-factor authentication) นั้นมีข้อจำกัดในการใช้ ตัวอย่างเช่น สิ่งที่คุณมี (Possession factor) นั้นอาจจะสูญหายหรือถูกขโมยได้ สิ่งที่คุณรู้ (Knowledge factor) อาจจะถูกดักฟัง เดา หรือขโมยจากเครื่องคอมพิวเตอร์ สิ่งที่คุณเป็น (Biometric factor) จัดได้ว่าเป็นวิธีที่มีความปลอดภัยสูงอย่างไรก็ตามการที่จะใช้เทคโนโลยีนี้ได้นั้นจำเป็นต้องมีการลงทุนที่สูง เป็นต้น

ดังนั้นจึงได้มีการนำแต่ละคุณลักษณะมาใช้ร่วมกัน (multi-factor authentication) ตัวอย่างเช่น ใช้สิ่งที่คุณมีกับสิ่งที่คุณรู้มาใช้ร่วมกัน เช่น การใช้ลายมือชื่อร่วมกับการใช้บัตรเครดิตหรือการใช้รหัสผ่านร่วมกับการใช้บัตร ATMเป็นต้น การนำแต่ละลักษณะของการพิสูจน์ตัวตนมาใช้ร่วมกันมากกว่า 1 ลักษณะ จะช่วยเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูล

2.การกำหนดสิทธิ์ (Authorization)

การกำหนดสิทธิ์ คือขั้นตอนในการอนุญาตให้แต่ละบุคคลสามารถเข้าถึงข้อมูลหรือระบบใดได้บ้าง ก่อนอื่นต้องทราบก่อนว่าบุคคลที่กล่าวอ้างนั้นคือใครตามขั้นตอนการพิสูจน์ตัวตนและต้องให้แน่ใจด้วยว่าการพิสูจน์ตัวตนนั้นถูกต้อง

3.การเข้ารหัส (Encryption)

การเข้ารหัสคือการเก็บข้อมูลให้เป็นส่วนบุคคลจากบุคคลอื่นที่ไม่ได้รับอนุญาต ส่วนประกอบ 2 ส่วนที่สำคัญที่จะช่วยทำให้ข้อมูลนั้นเป็นความลับได้ก็คือ การกำหนดสิทธิ์และการพิสูจน์ตัวตนเพราะว่าก่อนการอนุญาตให้บุคคลที่กล่าวอ้างเข้าถึงข้อมูลหรือถอดรหัสข้อมูลนั้นต้องสามารถแน่ใจได้ว่าบุคคลที่กล่าวอ้างนั้นเป็นใครและได้รับอนุญาตให้สามารถเข้ามาดูข้อมูลได้หรือไม่ ในการเข้ารหัสนั้นวิธีการหนึ่งที่ทำได้คือการเข้ารหัสในรูปแบบของกุญแจลับ (Secret key) ซึ่งในการใช้คีย์รูปแบบนี้ต้องเฉพาะผู้ที่มีกุญแจลับนี้เท่านั้นที่สามารถรับข้อมูลที่เข้ารหัสแล้วได้

4. การรักษาความสมบูรณ์ (Integrity)

การรักษาความสมบูรณ์ คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไปจากต้นฉบับ (source) ไม่ว่าจะเป็นโดยบังเอิญหรือดัดแปลงโดยเจตนาที่อาจส่งผลเสียต่อข้อมูล การ

คุกคามความสมบูรณ์ของข้อมูลคือการที่บุคคลที่ไม่ได้รับอนุญาตสามารถที่จะเข้าควบคุมการจัดการของข้อมูลได้

5. การตรวจสอบ (Audit)

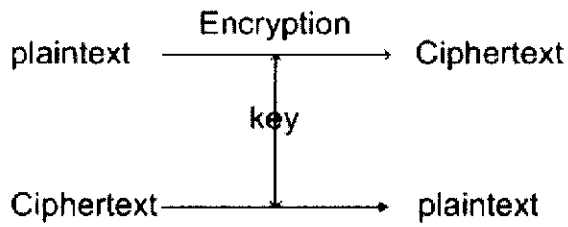
การตรวจสอบ คือการตรวจสอบหลักฐานทางอิเล็กทรอนิกส์ ซึ่งสามารถใช้ในการติดตามการดำเนินการเพื่อตรวจสอบความถูกต้องและแม่นยำ ตัวอย่างเช่นการตรวจสอบบัญชีชื่อผู้ใช้ โดยผู้ตรวจบัญชี ซึ่งการตรวจสอบความถูกต้องของการดำเนินการเพื่อให้แน่ใจว่าหลักฐานทางอิเล็กทรอนิกส์นั้นได้ถูกสร้างและสั่งให้ทำงานโดยบุคคลที่ได้รับอนุญาต และในการเชื่อมต่อเหตุการณ์เข้ากับบุคคลจะต้องทำการตรวจสอบหลักฐานของบุคคลนั้นด้วย ซึ่งถือเป็นหลักการพื้นฐานของขั้นตอนการทำงานของการพิสูจน์ตัวตนด้วย

การพิสูจน์ตัวตนจัดเป็นการตรวจสอบหลักฐานขั้นพื้นฐานที่สำคัญที่สุดใน 5 ระดับชั้นของการควบคุมความปลอดภัย ดังนั้นการพิสูจน์ตัวตนดีจะช่วยเพิ่มความมั่นคงปลอดภัยขั้นพื้นฐานให้กับระบบมากยิ่งขึ้น

2.2.1 ศาสตร์แห่งการเข้ารหัสลับ(Crytography)

การเข้ารหัสลับคือศาสตร์ที่ว่าด้วย การเข้ารหัสลับ(Encryption) และการถอดรหัสลับ (Decryption) ซึ่งจะช่วยในการรักษาความลับของข้อมูล โดยสามารถแบ่งตามลักษณะของกุญแจได้ 2 ชนิดคือ การเข้ารหัสลับด้วยกุจแจลับ (Secret Key Cryptography) และแบบการเข้ารหัสลับด้วยกุญแจสาธารณะ (Public Key Crytography)

การเข้ารหัสลับด้วยกุญแจลับ (Secret Key Cryptography) การเข้ารหัสลับแบบนี้จะใช้กุญแจอันเดียว เมื่อนำข้อความ (Plaintxet) และกุญแจมาเข้ารหัสลับผลที่ได้จะเป็นข้อมูลที่ไม่สามารถเข้าใจความหมายได้ (ciphertext) ซึ่งปกติความยาวของ ciphertext จะเท่ากับความยาวของข้อความด้านการถอดรหัสลับก็คือการทำย้อนกลับของการเข้ารหัสลับและใช้กุญแจเดิมในการถอดรหัส



รูปที่ 2.3 การการเข้ารหัสลับและการถอดรหัสลับ

ในบางทีการเข้ารหัสลับด้วยกุญแจลับจะเรียกว่าการเข้ารหัสแบบสมมาตร(symmetric cryptography)

การเข้ารหัสลับด้วยกุญแจลับกับความปลอดภัย

การใช้การเข้ารหัสลับด้วยกุญแจลับสามารถนำไปใช้ในการสร้างความปลอดภัยในระบบได้หลายอย่าง เช่น การส่งข้อมูลผ่านช่องทางที่ไม่ปลอดภัย

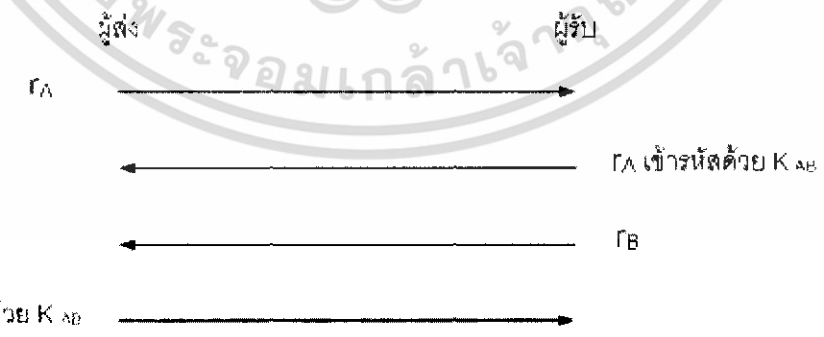
การส่งข้อมูลผ่านตัวกลางนั้นมีโอกาสที่จะโดนดักจับข้อมูลได้ง่าย จึงมีการนำการเข้ารหัสลับด้วยกุญแจลับมาใช้โดยผู้รับกับผู้ส่งจะต้องรู้ความลับร่วมกัน(กุญแจ) เมื่อจะส่งข้อมูลผู้ส่งก็จะเข้ารหัสลับข้อมูลผู้รับก็เพียงแค่นำข้อมูลที่เข้ารหัสมาถอดรหัส วิธีนี้ผู้ประสงค์ร้ายก็ยังสามารถดักจับข้อมูลได้ แต่ข้อมูลที่ได้ไปก็จะไม่สามารถเข้าใจได้ ข้อเสียของวิธีนี้คือผู้ที่รู้ความลับร่วมสามาถถอดรหัสลับได้ถึงแม้จะไม่ใช่ผู้ที่เราต้องการสื่อสารด้วย

การเก็บข้อมูลให้ปลอดภัยบนตัวกลางที่ไม่ปลอดภัย

การเก็บข้อมูลสามารถนำการเข้ารหัสลับด้วยกุญแจลับไปประยุกต์ใช้โดยมีการเข้ารหัสข้อมูลก่อนที่จะเก็บลงบนตัวกลางด้วยวิธีนี้ก็จะไม่มีใครสามารถอ่านข้อมูลได้ถ้าไม่รู้คีย์ แต่ข้อเสียของวิธีนี้คือถ้า ผู้ทีเป็นเจ้าของกูญแจลับทำกุญแจลับหายก็จะไม่สามารถอ่านข้อมูลได้

การระบุตัวตน (Authentication)

เมื่อผู้รับกับผู้ส่งต้องการที่จะระบุตัวตนสามารถทำได้โดยสมมติว่าผู้ใช้ทั้งสองคนรู้ความลับร่วมคือ K_{AB} ทั้งสองคนก็จะสุ่มตัวเลขขึ้นมาเรียกว่า challenge สมมติว่า ผู้ส่งเลือก r_A ผู้รับเลือก r_b ค่า x ก็จะโดนเข้ารหัสลับด้วยกุญแจซึ่งจะเรียกว่า respond



รูปที่ 2.4 การเข้ารหัสลับด้วยกุญแจลับ

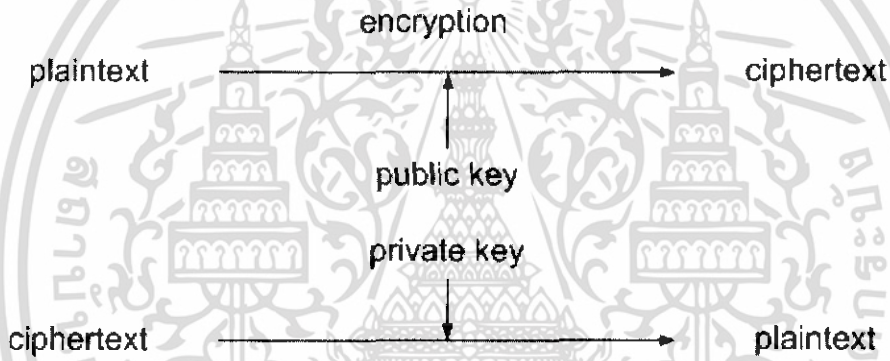
การตรวจสอบความครบถ้วนสมบูรณ์(Integrity check)

การเข้ารหัสลับด้วยกุญแจลับสามารถนำไปใช้ตรวจสอบความครบถ้วนสมบูรณ์ของข้อมูล โดยสามารถใช้ เพื่อสร้าง checksum ของข้อมูลแต่ไม่เป็นที่นิยมเนื่องจากผู้ประสงค์ร้ายสามารถแก้ไขข้อมูลแล้วคิด checksum ใหม่ได้

การเข้ารหัสลับด้วยกุญแจสาธารณะ (Public Key Cryptography)

การเข้ารหัสลับด้วยกุญแจสาธารณะหรือบางทีเรียกว่าการเข้ารหัสลับแบบไม่สมมาตร (Asymmetric cryptography) การใช้กุญแจสาธารณะในการเข้ารหัสลับมีแนวคิดที่ต่างจากการใช้กุญแจลับคือ แทนที่จะต้องมีการรู้ความลับร่วมในกุญแจแต่ละชุดจะประกอบด้วยกุญแจ 2 อัน คือ กุญแจส่วนตัว (Private key) ซึ่งต้องเก็บไว้ไม่ให้ใครรู้และกุญแจสาธารณะ (Public Key) ซึ่งจะแจกจ่ายไป

ในการเข้ารหัสลับด้วยกุญแจสาธารณะจะใช้กุญแจสาธารณะในการเข้ารหัสและใช้กุญแจส่วนตัวในการถอดรหัสซึ่งการเข้ารหัสลับและถอดรหัสลับนั้นจะสร้างจากฟังก์ชันทางคณิตศาสตร์ที่อินเวิร์สกัน



รูปที่ 2.5 การเข้ารหัสลับด้วยกุญแจสาธารณะ

2.2.2 การเข้ารหัสลับด้วยกุญแจสาธารณะกับความปลอดภัย

การเข้ารหัสลับด้วยกุญแจสาธารณะสามารถทำทุกอย่างที่การเข้ารหัสลับด้วยกุญแจลับได้ แต่อัลกอริธึมที่ใช้สร้างกุญแจสาธารณะนั้นช้ากว่าอัลกอริธึมที่ใช้สร้างกุญลับมากซึ่งปกติแล้วการเข้ารหัสลับด้วยกุญแจสาธารณะจะใช้ร่วมกับการเข้ารหัสลับด้วยกุญแจลับ การเข้ารหัสลับด้วยกุญแจสาธารณะใช้กันอย่างกว้างขวาง เพราะระบบความปลอดภัยในเครือข่ายมีรากฐานมาจากเทคโนโลยีของกุญแจสาธารณะ

การส่งข้อมูลผ่านช่องทางที่ไม่ปลอดภัย

สมมติว่าผู้ส่งมีคู่กุญแจสาธารณะ,กุญแจส่วนตัว คือ (e_A , d_A) และของผู้รับคือ (e_B , d_B) สมมติว่าผู้ส่งรู้กุญแจสาธารณะของผู้รับและผู้รับรู้กุญแจสาธารณะของผู้ส่ง

รูปที่ 2.6 การเข้ารหัสลับด้วยกุญแจสาธารณะและถอดด้วยกุญแจลับ

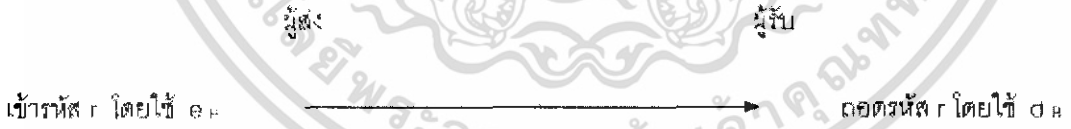
การเก็บข้อมูลให้ปลอดภัยบนตัวกลางที่ไม่ปลอดภัย

วิธีที่ใช้จะเหมือนกับแบบที่ใช้กุญแจลับ โดยเข้ารหัสลับข้อมูลโดยใช้ กุญแจสาธารณะก็จะไม่มีใครสามารถอ่านข้อมูลได้แล้วถอดรหัสโดยใช้กุญแจส่วนตัวแต่ถ้าต้องการความรวดเร็วอาจจะไม่ต้องใช้กุญแจส่วนตัวในการเข้ารหัส แต่สุ่มกุญแจลับขึ้นมาและเข้ารหัสลับข้อมูลด้วยกุญแจลับและเข้ารหัสลับกุญแจลับด้วยกุญแจสาธารณะ

การใช้กุญแจสาธารณะมีข้อดีที่เหนือกว่ากุญแจลับ คือ ผู้ส่งสามารถเข้ารหัสลับข้อมูลได้โดยไม่ต้องรู้กุญแจสำหรับถอดรหัสลับ

การระบุตัวตน (Authentication)

ในการระบุตัวตนโดยใช้กุญแจลับผู้รับและผู้ส่งถ้าต้องการติดต่อสื่อสารกันจำเป็นต้องรู้ความลับร่วมกัน ถ้าผู้ส่งต้องการพิสูจน์ตัวตนของผู้รับหลายๆ คนถ้าใช้วิธีของกุญแจลับผู้ส่งก็ต้องเก็บความลับไว้มากมายซึ่งการใช้กุญแจสาธารณะในการระบุตัวตนจะสะดวกกว่าซึ่งการใช้กุญแจสาธารณะทำให้ผู้ส่งจดจำแค่กุญแจส่วนตัวของตัวเอง ตัวอย่างดังรูป



รูปที่ 2.7 การระบุตัวตนโดยใช้กุญแจลับ

2.3 ทฤษฎีสำหรับการส่งข้อมูลบนอินเตอร์เนทให้ปลอดภัยด้วย SSL

ในระหว่างการพัฒนาเครือข่ายอินเตอร์เนทในระยะเริ่มแรกนั้น ไม่ได้มีการเน้นในการพัฒนาด้านความปลอดภัยในการส่งข้อมูลบนเครือข่าย เนื่องจากในระยะนั้นเครือข่ายอินเตอร์เนทนี้ถูกใช้ในการติดต่อสื่อสารระหว่างกลุ่มนักวิจัยในมหาวิทยาลัยและสถาบันต่างๆไม่กี่กลุ่ม ซึ่งมีความรู้จักคุ้นเคยกันและมีความเชื่อถือต่อกันและกันดังนั้นข้อมูลที่ถูกส่งไปบนเครือข่ายอินเตอร์เนทจึง

เป็นลักษณะของข้อมูลที่ไม่ได้เข้ารหัสลับใดๆจนทุกวันนี้การส่งข้อมูลส่วนใหญ่บนเครือข่ายอินเตอร์เนทก็ยังคงลักษณะนี้อยู่

ในปัจจุบัน ผู้ใช้เครือข่ายอินเตอร์เนทมีจำนวนเพิ่มขึ้นอย่างมากและเครือข่ายนี้ถูกใช้งานในรูปแบบต่างๆมากมายหลายรูปแบบโดยเฉพาะอย่างยิ่งในการพาณิชย์อิเล็กทรอนิกส์(e-commerce) ซึ่งผู้ซื้อและผู้ขายจะต้องส่งข้อมูลที่เป็นความลับถึงกันและกัน เช่นผู้ซื้อส่งหมายเลขบัตรเครดิตหรือที่อยู่และเบอร์โทรศัพท์โดยส่งผ่านไปบนเครือข่ายข่ายอินเตอร์เนทหากข้อมูลเหล่านี้ถูกส่งไปแบบธรรมดาก็จะเป็นการค่อนข้างง่ายที่ผู้ไม่หวังดีจะสามารถดักจับข้อมูลเหล่านี้ (sniffing) แล้วนำไปใช้ได้ เนื่องจากข้อมูลเหล่านี้อยู่ในรูปของข้อความที่ไม่ได้เข้ารหัส ผู้ดักจับข้อมูลก็จะสามารถนำข้อมูลนั้นไปใช้ได้ทันที

ในอีกกรณีหนึ่งบนเครือข่ายอินเตอร์เนทนี้ผู้ใช้สามารถเชื่อมต่อและใช้งานเครื่องคอมพิวเตอร์ใดๆก็ได้หากผู้ใช้นั้นได้รับอนุญาตและสามารถพิสูจน์ตนเองโดยชื่อผู้ใช้และรหัสผ่านที่ถูกต้องบนเครื่องนั้นๆ โดยจะมีโปรแกรมที่ใช้ช่วยในการเชื่อมต่อและใช้งานนั้น เช่น telnet, rsh, rlogin, rcp, และ ftp เป็นต้นโปรแกรมเหล่านี้ส่งข้อมูลตามแบบมาตรฐานดั้งเดิมของเครือข่าย Internetกล่าวคือส่งข้อมูลทุกอย่าง(รวมทั้งชื่อผู้ใช้และรหัสผ่าน)ในรูปของข้อความที่ไม่ได้เข้ารหัส ดังนั้นหากมีผู้ดักจับข้อมูลเกี่ยวกับชื่อผู้ใช้และรหัสผ่านได้ผู้นั้นก็จะสามารถนำเอาชื่อผู้ใช้และรหัสผ่านนี้ไปใช้ในการเชื่อมต่อและใช้งานเครื่องคอมพิวเตอร์เครื่องนั้นได้ต่อไป

เนื่องจากปัญหาที่กล่าวมานี้เป็นปัญหาที่ค่อนข้างใหญ่เพราะการsniffingนั้นสามารถกระทำได้อย่างค่อนข้างง่ายจึงได้มีการคิดแก้ไขปัญหานี้ขึ้นโดย Netscapeได้คิดค้นโปรโตคอลใหม่ขึ้นมาคือ Secure Socket Layer Protocol (SSL) และโปรแกรมเมอร์ชาว Finland ได้เขียนโปรแกรมขึ้นชุดหนึ่ง เรียกว่า Secure Shell (SSH) ซึ่งทั้ง SSL และ SSH จะเข้ารหัสลับข้อมูลใดๆก่อนที่ข้อมูลนั้นจะถูกส่งไปบนเครือข่ายอินเตดร์เนท ดังนั้น หากผู้ไม่หวังดีสามารถดักจับข้อมูลนั้นไปได้ ผู้นั้นก็ไม่สามารถที่จะนำข้อมูลนั้นไปใช้ได้เพราะเขาไม่สามารถตีความข้อมูลนั้นได้

SSL นั้นได้รับการยอมรับอย่างกว้างขวางบน world wide web ในการใช้สำหรับตรวจสอบและเข้ารหัสลับการติดต่อสื่อสารระหว่างไคลเอ็นต์และเซิร์ฟเวอร์หน้าที่ของ SSL จะแบ่งออกเป็น 3 ส่วนใหญ่ๆคือ

1. การตรวจสอบเซิร์ฟเวอร์ว่าเป็นตัวจริง : ตัวโปรแกรมไคลเอ็นต์ที่มีขีดความสามารถในการสื่อสารแบบSSLจะสามารถตรวจสอบเครื่อ เซิร์ฟเวอร์ ที่ตนกำลังจะไปเชื่อมต่อได้ว่า เซิร์ฟเวอร์นั้นเป็นเซิร์ฟเวอร์ตัวจริงหรือไม่ โดยใช้เทคนิคการเข้ารหัสแบบกุญแจสาธารณะ(public key) ในการตรวจสอบใบรับรองสิทธิ์ (certificate) และ public ID ของ เซิร์ฟเวอร์ นั้น (โดยที่มีองค์กรที่ไคลเอ็นต์เชื่อถือเป็นผู้ออกใบรับรองสิทธิ์และ public ID ให้แก่ เซิร์ฟเวอร์นั้น)

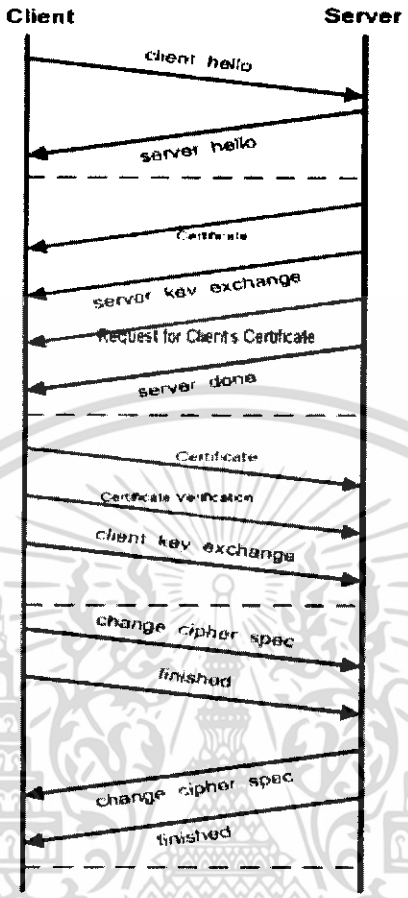
หน้าที่นี้ของ SSL เป็นหน้าที่ที่สำคัญ โดยเฉพาะอย่างยิ่งในกรณีที่ ไคลเอ็นต์ ต้องการที่จะส่งข้อมูลที่เป็นความลับ (เช่น หมายเลข credit card)ให้กับเซิร์ฟเวอร์ซึ่งไคลเอ็นต์จะต้องตรวจสอบก่อนว่าเซิร์ฟเวอร์เป็นตัวจริงหรือไม่

2. การตรวจสอบว่าไคลเอ็นต์เป็นตัวจริง : เซิร์ฟเวอร์ที่มีขีดความสามารถในการสื่อสารแบบSSLจะใช้เทคนิคเช่นเดียวกับในหัวข้อที่แล้วในการตรวจสอบไคลเอ็นต์หรือผู้ใช้ว่าเป็นตัวจริงหรือไม่โดยจะตรวจสอบใบรับรองและpublic ID (ที่มีองค์กรที่ เซิร์ฟเวอร์เชื่อถือเป็นผู้ออกให้) ของไคลเอ็นต์หรือผู้ใช้นั้น

หน้าที่นี้ของ SSL จะมีประโยชน์ในกรณีเช่น ธนาคารต้องการที่จะส่งข้อมูลลับทางการเงินให้แก่ลูกค้าของตนผ่านทางเครือข่ายอินเตอร์เนท(เซิร์ฟเวอร์ ก็จะต้องตรวจสอบ ไคลเอ็นต์ ก่อนว่าเป็น ไคลเอ็นต์ นั้นจริง)

3. การเข้ารหัสลับการเชื่อมต่อ : ในกรณีนี้ ข้อมูลทั้งหมดที่ถูกส่งระหว่างไคลเอ็นต์และเซิร์ฟเวอร์จะถูกเข้ารหัสลับโดยโปรแกรมที่ส่งข้อมูลเป็นผู้เข้ารหัสและโปรแกรมที่รับข้อมูลเป็นผู้ถอดรหัส(โดยใช้วิธีแบบกุญแจสาธารณะ)นอกจากการเข้ารหัสลับในลักษณะนี้แล้ว SSL ยังสามารถปกป้องความถูกต้องสมบูรณ์ของข้อมูลได้อีกด้วย กล่าวคือ ตัวโปรแกรมรับข้อมูลจะทราบได้หากข้อมูลถูกเปลี่ยนแปลงไปในขณะกำลังเดินทางจากผู้ส่งไปยังผู้รับ

ทฤษฎีทางด้านการสร้างการเชื่อมต่อแบบ SSL



รูปที่ 2.8 ขั้นตอนการสร้างการเชื่อมต่อของ SSL

1. ClientHelloจะเป็นข้อความที่ไคลเอ็นต์ใช้สำหรับเริ่มต้นการสื่อสารด้วยSSLโพรโตคอล โดยข้อความนี้จะเป็นการนำเสนอค่าพารามิเตอร์ต่าง ๆ ที่ตัวเองสามารถรองรับได้ส่งไปให้แก่เซิร์ฟเวอร์

2. ServerHello เมื่อเซิร์ฟเวอร์ได้รับข้อความ ClientHello เซิร์ฟเวอร์จะทำการตอบกลับด้วย ServerHello ไคลเอ็นต์จะนำเสนอค่าพารามิเตอร์ต่างๆ ให้แก่เซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการเลือกค่าพารามิเตอร์เหล่านั้นแล้วส่งกลับไปด้วยข้อความ ServerHello

3. Certificate เพื่อพิสูจน์ตัวตนของเซิร์ฟเวอร์ จากรูปไคลเอ็นต์จะมีกุญแจสาธารณะของCAที่เป็นCAที่เซิร์ฟเวอร์ได้นำข้อมูลต่างๆไปให้CAนั้นทำการSignให้โดยข้อความCertificateของฝั่งเซิร์ฟเวอร์จะประกอบไปด้วยกุญแจสาธารณะของเซิร์ฟเวอร์และข้อมูลได้รับการSign มาจาก CA นั้น ส่งไปให้แก่ไคลเอ็นต์

4. ServerKeyExchange เซิร์เวอร์จะทำการส่งข้อความ SeverKeyExchange ไปให้แก่ฝั่งไคลเอ็นต์ ในขณะที่ฟิลด์ CipherSuite จะเป็นตัวที่บอกถึงชนิดของอัลกอริทึมและขนาดของคีย์ข้อความและส่งคีย์สาธารณะไปด้วย

5. Request for Client's Certificate จะเป็นข้อความจากเซิร์ฟเวอร์ที่เป็นการบอกให้แก่ไคลเอ็นต์ทราบว่าเซิร์ฟเวอร์มีความต้องการที่จะทำการพิสูจน์ตัวตนของไคลเอ็นต์

6. Server done จะเป็นข้อความที่บอกให้ไคลเอ็นต์ทราบว่าเซิร์ฟเวอร์ได้ส่งข้อมูลที่ใช้สำหรับการเริ่มต้นการติดต่อสื่อสารเรียบร้อยแล้วซึ่งข้อความนี้จะไม่มีข้อมูลใดๆแต่มันเป็นสิ่งสำคัญต่อไคลเอ็นต์จะต้องได้รับข้อความนี้ก่อนไคลเอ็นต์จึงจะสามารถกระทำเฟสต่อไปของการสร้างการเชื่อมต่อที่ปลอดภัยโดยใช้ SSL โพรโตคอลได้

7. Certificate เป็นข้อความของฝั่งไคลเอ็นต์ที่เป็นข้อมูลที่ไคลเอ็นต์ได้รับจากการส่งข้อมูลของไคลเอ็นต์ เช่น กุญแจสาธารณะของไคลเอ็นต์ไปให้แก่ CA ที่เซิร์ฟเวอร์เชื่อถือทำการ Sign ข้อมูลดังกล่าว

8. Certificate Verification เป็นข้อมูลของกุญแจสาธารณะที่ไคลเอ็นต์ใช้ในการพิสูจน์ตัวตน

9. Client key exchange เมื่อไคลเอ็นต์ได้รับข้อความ Certificate จากเซิร์ฟเวอร์ จะใช้ฟังชั่น Verify()ในการพิสูจน์ตัวตนของเซิร์ฟเวอร์ถ้าพิสูจน์ตัวตนถูกต้องไคลเอ็นต์จะนำกุญแจสาธารณะของเซิร์ฟเวอร์ที่ได้ส่งมาด้วยนั้นนำมาเข้ารหัสกุญแจลับที่ได้สร้างขึ้นมา

10. Change cipher spec หลังจากที่ไคลเอ็นต์ส่งข้อความ ClientKeyExchange เรียบร้อยแล้วจะถือว่าสิ้นสุดการต่อรองค่าพารามิเตอร์ต่าง ๆ ระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ ณ จุดนี้ทั้งสองระบบสามารถที่จะสื่อสารกันได้อย่างปลอดภัยโดยใช้SSLโพรโตคอลข้อความChangeCipherSpecจะเป็นข้อความที่ชี้ให้เห็นอย่างชัดเจนว่าการรับข้อมูลอย่างปลอดภัยโดยใช้SSLโพรโตคอลสามารถทำงานได้แล้ว

11. Finished ทันทีหลังจากที่ได้ทำการส่งข้อความ ChangeCipherSpec แต่ละระบบจะทำการส่งข้อความ Finished ซึ่งเป็นข้อความที่ใช้ตรวจสอบความถูกต้องของข้อมูลที่ได้ทำการส่งไป เช่น ข้อมูลเกี่ยวกับคีย์ รายละเอียดเกี่ยวกับการต่อรองค่าพารามิเตอร์ต่างๆ ในครั้งก่อน และข้อมูลใดๆ ที่จะเป็นการตัวระบุตัวตนของทั้งเซิร์ฟเวอร์และไคลเอ็นต์ โดยค่าเหล่านี้จะต้องทำให้เป็นแฮชแวลู (hash value) ก่อนทำการส่งออกไป

2.4 ทฤษฎีใบรับรองสิทธิ์(Certificate)

การเข้ารหัสลับให้กับข้อความอาจไม่เพียงพอกับความปลอดภัยเนื่องจากอาจเกิดปัญหา man-in-the-middle ได้ โครงงานนี้จึงได้เพิ่มใบรับรองสิทธิ์ซึ่งใช้ในการพิสูจน์ตัวตนว่าผู้ที่เราติดต่ออยู่ด้วยเป็นบุคคลเดียวกันกับที่เราเข้าใจ

2.4.1 ปัญหา man-in-the-middle

กระบวนการการทำงานของโพรโตคอลSSL ที่ใช้การเข้ารหัสข้อมูลเพียงอย่างเดียวระหว่างสองระบบนั้นมันยังไม่มีความปลอดภัยเพียงพอ เมื่อ Alice ที่มีบทบาทเป็นไคล์เอ็นต์และ Bob ที่บทบาทเป็นเซิร์ฟเวอร์ และมีบุคคลที่อยู่ตรงกลางชื่อว่า Trudy โดยขั้นแรก Trudy จะแสดงบทบาทตัวเองเป็น Alice โดยบอก Bob ว่า "I'm Alice" และ Bob จะส่งข้อความว่า "I'm Bob" พร้อมส่งกุญแจสาธารณะของตนเองให้แก้ Trudy ซึ่งในขณะนี้ได้ปลอมตัวเป็น Alice เมื่อได้กุญแจสาธารณะของ Bob Trudy จะทำการสร้างกุญแจลับขึ้นมาแล้วใช้กุญแจสาธารณะของ Bob มาเข้ารหัสกุญแจลับที่ได้สร้างขึ้นมา เสร็จแล้วก็ทำการส่งข้อมูล (Cipher text)นั้นให้แก่ Bob ณ จุดนี้ Trudy สามารถที่จะติดต่อสื่อสารกับ Bob ได้ จากนั้น Trudy จะทำการปลอมตัวเองให้เป็น Bob ที่มีบทบาทเป็นเซิร์ฟเวอร์ โดย Trudy จะรับข้อความจาก Alice ว่า "I'm Bob" พร้อมส่งกุญแจสาธารณะของตนเองให้แก่ Alice ณ ตอนนี้ Trudy ก็สามารถติดต่อสื่อสารกับ Alice ได้ เมื่อถึงจุดนี้ Trudy จะทราบข้อมูลทุกอย่างที่ Bob กับ Alice ติดต่อกัน ดังนั้น Trudy สามารถสร้างความเสียหายให้เกิดแก่ Bob และ Alice ได้ โพรโตคอลได้ทำการแก้ไขปัญหาดังกล่าวโดยได้เพิ่มวิธีการที่เรียกว่าการพิสูจน์ตัวตนโดยการใช้ใบรับรองสิทธิ์

2.4.2 ใบรับรองสิทธิ์(Certificate)

คำจำกัดความง่ายๆ ของใบรับรองสิทธิ์ก็คือใบรับรองสิทธิ์จะผนึกรวมกุญแจสาธารณะกับชื่อพิเศษชื่อพิเศษคือชื่อของบุคคลหรือสิ่งซึ่งเป็นเจ้าของกุญแจสาธารณะใบรับรองสิทธิ์ก็เปรียบเหมือนหนังสือเดินทางซึ่งผนึกรูปภาพกับชื่อเพื่อเป็นเอกลักษณ์ของแต่ละบุคคลใบรับรองสิทธิ์ตีพิมพ์ โดยบุคคลที่สามซึ่งเชื่อถือได้โดยประกอบด้วยข้อมูลเกี่ยวกับบุคคลที่สามที่ตีพิมพ์ใบรับรองสิทธิ์นั้นในใบรับรองสิทธิ์ยังมีเครื่องป้องกันซึ่งมีจุดมุ่งหมายเพื่อการพิสูจน์ตัวตนและเพื่อป้องกันการปลอมแปลงและ tampering ใบรับรองสิทธิ์ใช้ได้ในช่วงเวลาที่กำหนดเมื่อหมดอายุจะต้องออกใบรับรองสิทธิ์ใหม่และใบรับรองสิทธิ์เก่าจะไม่ได้รับการเชื่อถืออีกต่อไป

ใบรับรองสิทธิ์จะเซ็น(Sign)โดยด้วยกุญแจส่วนตัวของผู้ออกใบรับรองสิทธิ์และประกอบด้วยข้อมูลที่จำเป็นเพื่อพิสูน์ความถูกต้องซึ่งข้อมูลเหล่านี้ประกอบด้วยรายละเอียดทั่วๆไป,ผู้ออกใบรับรองสิทธิ์และช่วงเวลาที่ใบรับรองสิทธิ์นั้นมีผล ส่วนประกอบที่เป็นกุญแจสำคัญคือ ใบรับรอง

สิทธิ์ของผู้ออกใบรับรองสิทธิ์สำหรับพิสูจน์ว่าใบรับรองสิทธิ์นั้นออกโดยผู้ออกใบรับรองสิทธิ์นั้นจริงๆเพราะว่ามันประกอบด้วยกุญแจสาธารณะของผู้ออกใบรับรองสิทธิ์ซึ่งจำเป็นในการพิสูจน์ลายเซ็นของใบรับรองสิทธิ์นั้น

การเซ็นใบรับรองสิทธิ์ด้วยกุญแจส่วนตัวของของผู้ออกใบรับรองสิทธิ์ ใครก็ตามที่มีกุญแจสาธารณะของผู้ออกใบรับรองสิทธิ์สามารถพิสูจน์ตัวตนได้,ลายเซ็นทำหน้าที่เป็นเครื่องป้องกันการ tampering เมื่อผู้ออกใบรับรองสิทธิ์เซ็นใบรับรองสิทธิ์ของผู้ที่ร้องก็สามารถยืนยันว่าได้ตรวจสอบความเป็นจริงของกุญแจสาธารณะซึ่งบรรจุในใบรับรองสิทธิและสถานะความน่าเชื่อถือเมื่อใดก็ตาม ที่ยังเชื่อถือผู้ออกใบรับรองสิทธิ์ใบรับรองสิทธิ์ที่ออกโดยผู้ออกใบรับรองสิทธิ์ก็สามารถเชื่อถือได้

ใบรับรองสิทธิ์สร้างพร้อมกับหมายเลขซีเรียลซึ่งจะบรรจุอยู่ภายใน ซึ่งหมายเลขซีเรียลนี้จะไม่ซ้ำกันในแต่ละใบรับรองสิทธิ์ที่ออกโดยผู้ออกใบรับรองสิทธิ์หมายเลขซีเรียลของใบรับรองสิทธิ์มักจะใช้เพื่อจำแนกใบรับรองสิทธิ์ได้อย่างรวดเร็ว

2.5 ทฤษฎีการสร้างผู้ให้บริการใบรับรองสิทธิ์(Certification Authorities)

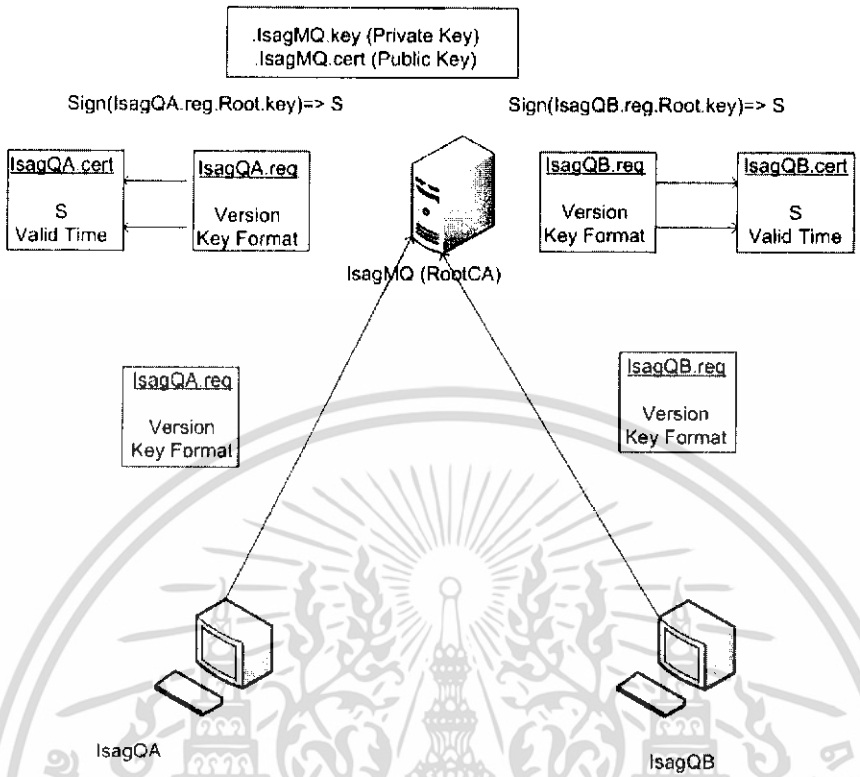
ผู้ให้ใบรับรองสิทธิ์มีอยู่ 2 ชนิดคือ Private CA และ Public CA โดยโครงงานนี้เป็นแบบแรก ซึ่งจะออกใบรับรองให้เฉพาะ IsagQ ซึ่งเป็น Self-sign CA

2.5.1 ผู้ให้บริการใบรับรองสิทธิ์(Certification Authorities)

Certification Authorities(CA) คือ องค์กรหรือบริษัทซึ่งทำหน้าที่ออกใบรับรองสิทธิ์,CA มีสิ่งที่ต้องรับผิดชอบคือใบรับรองสิทธิ์ที่ออกต้องถูกต้องสมบูรณ์นั่นคือCAต้องทำให้แน่ใจว่าใบรับรองสิทธิ์ทุกๆใบที่ออกไปบรรจุกุญแจสาธารณะที่ออกโดยฝ่ายที่ร้องขอจริงๆ

CA แบ่งได้เป็น 2 ชนิดคือ Private CA ซึ่งรับผิดชอบการออกใบรับรองสิทธิ์เฉพาะสมาชิกขององค์กรที่เป็นเจ้าของและเชื่อถือได้เพื่อสมาชิกขององค์กรที่เป็นเจ้าของทางด้าน Public CA เช่น Verisign , Thwate มีหน้าที่รับผิดชอบในการออกใบรับรองสิทธิ์ให้กับใครก็ได้ที่ต้องการและต้องได้รับความเชื่อถือจากสาธารณะ

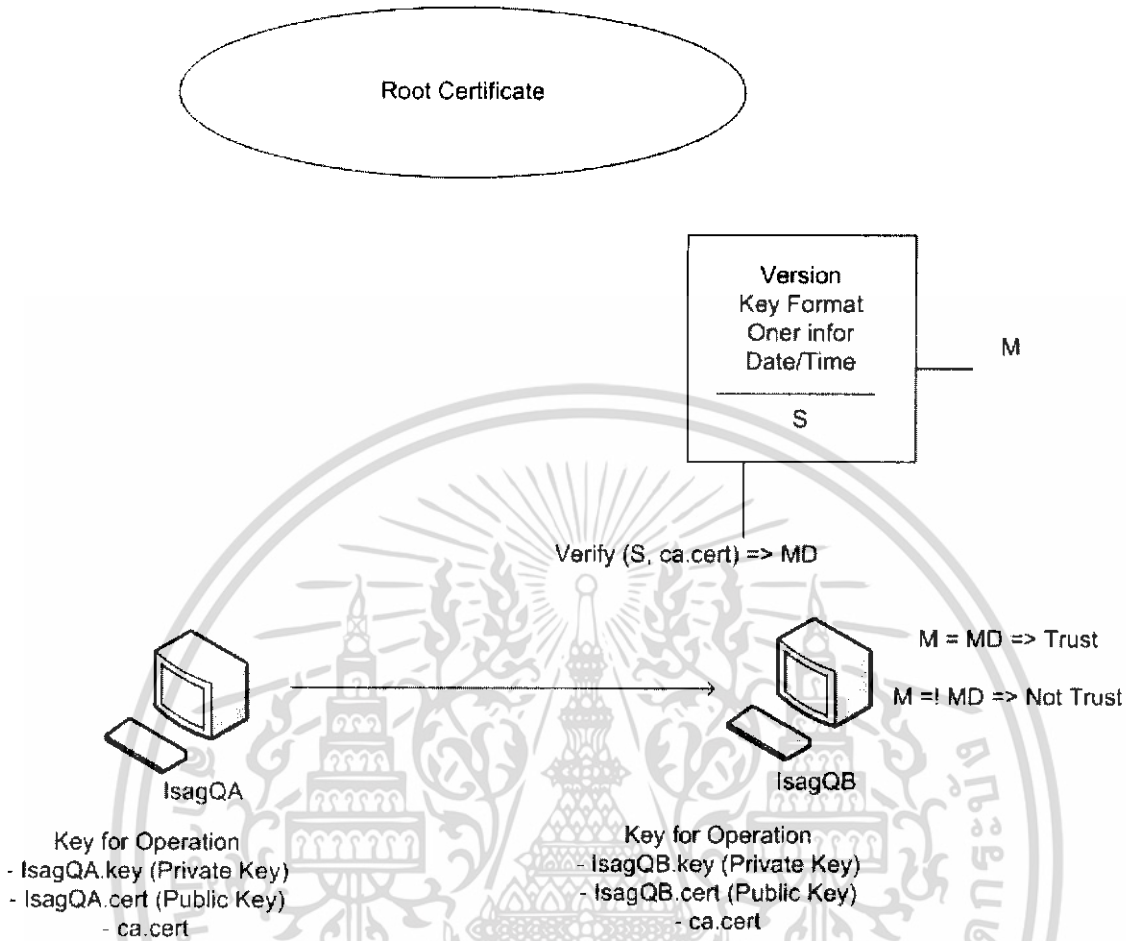
2.5.2 ขั้นตอนในการร้องขอใบรับรองสิทธิ์



รูปที่ 2.9 ขั้นตอนการร้องขอใบรับรองสิทธิ์

1. ผู้ร้องขอซึ่งในที่นี้จะประกอบด้วย IsagQA และ IsagQB จำเป็นต้องมีใบรับรองสิทธิ์ของ Root CAซึ่งก็คือ IsagMQ เพื่อเก็บไว้ใช้ในระบบพิสูจน์ตน
2. เมื่อผู้ร้องขอมีใบรับรองสิทธิ์ของIsagMQ แล้ว ก็ให้ทำการร้องขอไปยัง Root CA ก่อนที่ผู้ร้องขอจะส่ง .req ออกไปผู้ร้องขอจำเป็นต้องมีไฟล์ 2 ชนิดคือ คือ IsagQA.key หรือ IsagQB.key ซึ่งก็คือกุญแจส่วนตัวของผู้ร้องขอและ IsagQA.req หรือ IsagQBซึ่งก็คือไฟล์ที่ใช้ร้องขอไปยังผู้ออกใบรับรองสิทธิ์โดยไฟล์ที่ร้องขอต้องถูกเข้ารหัสด้วยroot.certก่อนเพื่อที่จะมั่นใจได้ว่า Root CA จะสามารถเปิดดูได้เพียงผู้เดียว
3. หลังจากที่ Root CA ได้รับ IsagQA.req หรือ IsagQB.req แล้วก็จะทำการแฮช(hash)แล้วนำค่านั้นมา sign ด้วย Root.key ผลลัพธ์ที่ได้จะได้ค่า S (Digital Signature) แล้วนำค่า S ที่ได้ไปเก็บไว้ใน IsagQA.cert หรือ IsagQB.cert
4. ผู้ออกใบรับรองสิทธิ์จะทำการส่ง IsagQA.cert หรือ IsagQB.cert คืนกลับให้ผู้ร้องขอ ขั้นตอนในการร้องขอใบรับรองสิทธิ์

2.5.3 การพิสูจน์ตนโดยการใช้ใบรับรองสิทธิ์(Certificate Authentication)



รูปที่ 2.10 รูปแสดงการพิสูจน์ตัวตนโดยใช้ใบรับรองสิทธิ์

เพื่อเริ่มทำการพิสูจน์ตัวตน IsagQAก็จะส่ง(IsagQA.cert) ให้แก่IsagQBทำการพิสูจน์ตนของฝั่ง IsagQA เมื่อ IsagQB ได้รับไฟล์ IsagQA.cert ซึ่งไ ฟล์นี้ประกอบไปด้วยสองส่วนคือ ส่วนข้อมูลส่วนตัวของIsagQAและค่า S (Digital Signature) กระบวนการพิสูจน์ตนทางฝั่ง IsagQB คือ จะนำเอาข้อมูลส่วนตัวของIsagQAมาทำ MD5 ได้ค่าแฮช (hash) มาหนึ่งค่าคือค่า M จากนั้นนำค่า S ของIsagQAและกุญแจสาธารณะของ Root Certificate มาทำการตรวจสอบค่าที่ได้ออกมาคือ MD นำค่า M กับ MD มาเปรียบเทียบกัน ถ้าค่านั้นเท่ากันแสดงว่าการพิสูจน์ตัวตนนั้นถูกต้อง ถ้าไม่เท่ากันแสดงว่าการพิสูจน์ตนผิดพลาด

2.5.4 การแยกระหว่างพิสูจน์ตัวตนและการเข้ารหัส

การใช้ใบรับรองสิทธิ์เพียงอย่างเดียวในการทำทั้งการพิสูจน์ตัวตนและการเข้ารหัสข้อมูลนั้น เป็นวิธีการที่ไม่ดีนัก ยกตัวอย่างเช่นมีหลาย ๆ อัลกอริทึมที่ใช้ในการสร้างกุญแจสาธารณะ ที่มี

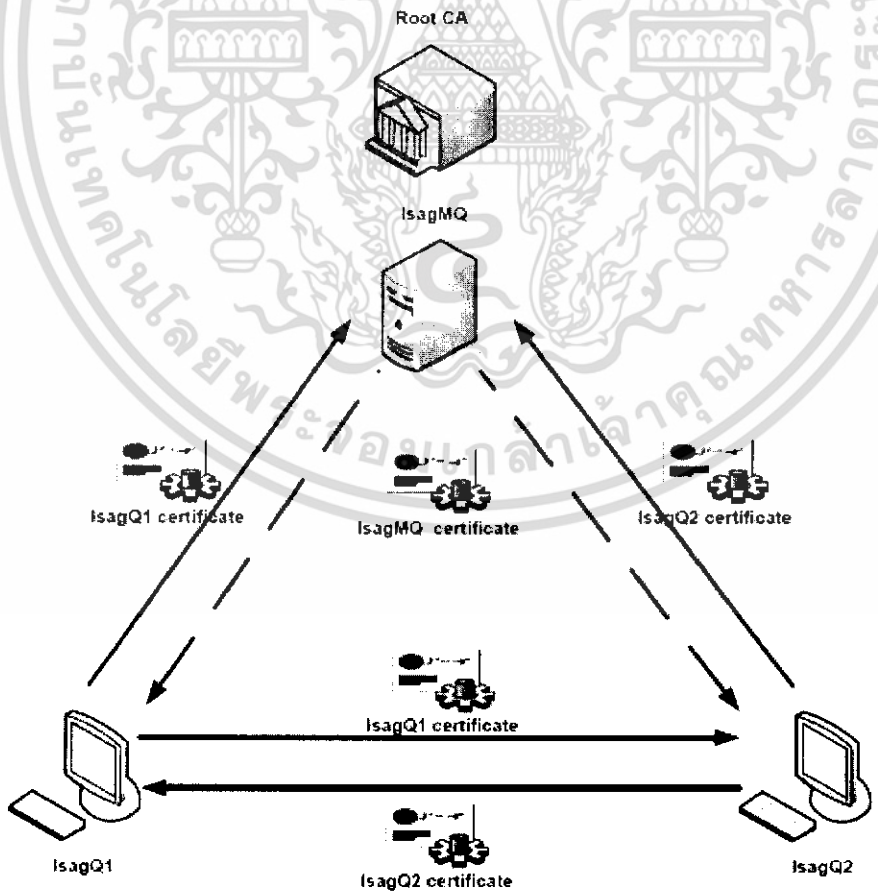
ไว้ใช้สำหรับการ Sign ข้อมูลเท่านั้น ไม่สามารถนำมาใช้ในการเข้ารหัสข้อมูลได้ เช่น DSA (Digital Signature Algorithm) ดังนั้นจึงมีการแยกข้อความระหว่างพิสูจน์ตนกับข้อความสำหรับการเข้ารหัส ซึ่งก็คือ เมื่อผู้ส่งข้อความ ส่งใบรับรองสิทธิ์ไปให้ผู้รับผู้รับก็จะตรวจสอบถ้าถูกต้องก็จะใช้กุญแจสาธารณะของผู้ส่งทำการเข้ารหัสกุญแจลับเพื่อทำกระบวนการ SSL ในขั้นต่อไป

บทที่ 3

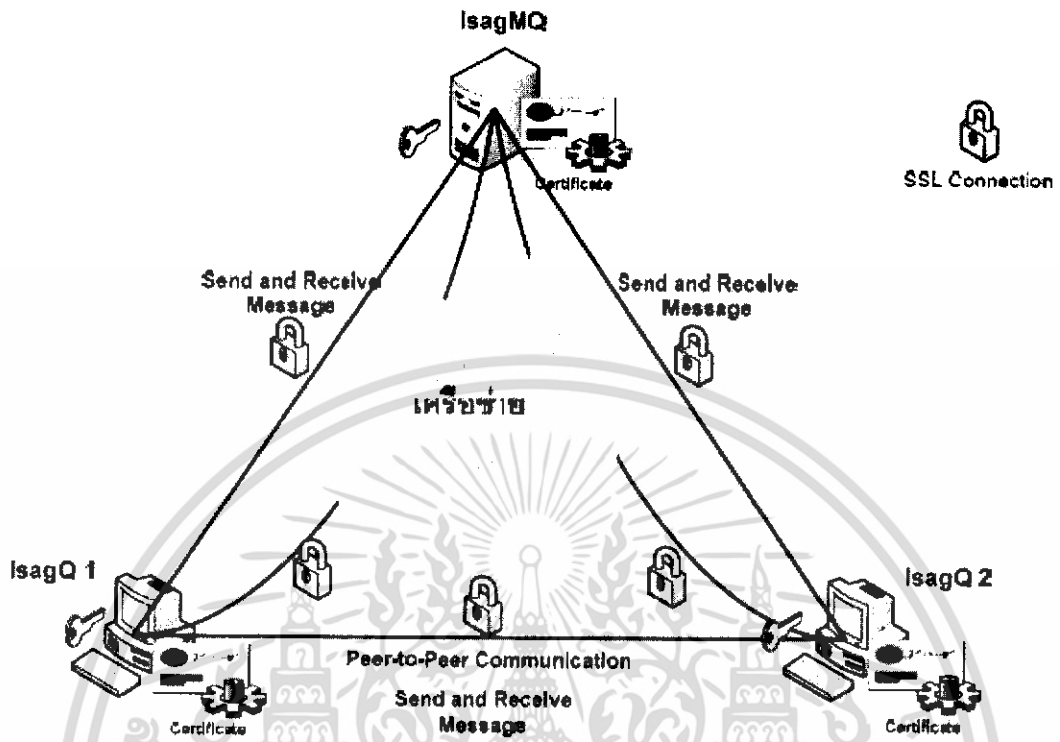
การออกแบบ IsagQ และIsagMQ

3.1 การออกแบบซอฟต์แวร์IsagQและIsagMQ ในปี2547

1. สามารถให้บริการโดยใช้ IsagQ ติดต่อมายัง IsagMQ ผ่านโพรโตคอล SSL ได้โดยไม่ขึ้นกับความแตกต่างของระบบปฏิบัติการที่ใช้ และ ภาษาที่ใช้เขียนโปรแกรม
2. การติดต่อระหว่าง IsagQ ด้วยกัน จะต้องผ่านการพิสูจน์ตัวตน เพื่อ
   2.1 ไม่ให้เกิดปัญหา Man in the middle
   2.2 สามารถสร้างความมั่นใจให้กับผู้ใช้ทั้งสองฝั่งได้ว่ากำลังติดต่ออยู่กับบุคคลที่ต้องการจริงๆ ดังรูป
3. เพิ่มการให้บริการ อาทิ การปฏิเสธ หรือ ยอมรับคู่สนทนา การแสดงรายชื่อผู้รอคำยินยอม การให้คำยินยอมแก่ผู้รอคำยินยอม และ การส่งไฟล์อย่างปลอดภัย ระหว่าง IsagQ ด้วยกันเอง



รูปที่ 3.1 IsagQ และ IsagMQ ในปี2547



รูปที่ 3.2 การติดต่อระหว่างIsagQ และ IsagQ โดยใช้ SSL Protocol

ความสามารถ	IsagQ 47
IsagQ ติดต่อไปยัง IsagMQ ด้วย SSL Protocol	ได้
สื่อสารโดยมีการเข้ารหัสข้อมูล	ได้
ใบรับรองสิทธิ์ของ IsagQ	มี
การพิสูจน์ตนระหว่าง IsagQ ด้วยกัน	มี
บริการยอมรับหรือปฏิเสธคู่สนทนา	มี
บริการแสดงรายชื่อผู้รอคำยินยอม และ การตอบรับคำขอคำยินยอม	มี
บริการลงทะเบียนและยกเลิกการลงทะเบียน	ไม่มี
บริการติดต่อกับ ICQ	มี
การส่งไฟล์ระหว่าง IsagQ แบบปลอดภัย	มี

ตารางที่ 3.1 แสดงความสามารถของ IsagQปี 2547

3.2 ความแตกต่างระหว่าง IsagMQและ IsagQ ปีการศึกษา 2547 กับ 2548

IsagQ ของปี 2547 พัฒนาบนระบบระบบปฏิบัติการ Windows โดยใช้ภาษา Java ในการเขียนโปรแกรมส่วน IsagQ ในปีนี้จะพัฒนาบน GAIM ซึ่งเป็นเป็นโปรแกรมรับส่งสารด่วนแบบโอเพนซอร์สซึ่งรองรับการทำงานหลายโพรโตคอลซึ่งพัฒนาด้วยภาษาซีซึ่งในปีนี้IsagQจะลดบทบาทในส่วนการให้บริการการรับส่งข้อความโดยจะคงไว้แต่บทบาทด้านความปลอดภัยโดยสามารถเปรียบเทียบ GAIM กับIsagQ ได้ดังตาราง

	GAIM	IsagQ47
1.แพลทฟอร์ม	Linux, BSD , MacOS X , Windows	Windows
2.โพรโตคอลที่รองรับ	AIM, ICQ, MSN Messenger, Yahoo, IRC, Jabber, Gadu-Gadu, SILC, GroupWise Messenger, Zephyr	ICQ,IsagQ
3.การเชื่อมต่อ	plaintext	SSL
4.การเข้ารหัสข้อความ	ไม่มีการเข้ารหัสข้อความ	มีการเข้ารหัสข้อความ
5.การพิสูจน์ตัวตน	ไม่มีการพิสูจน์ตัวตน	มีการพิสูจน์ตัวตน
4.ใบรับรองสิทธิ์	ไม่มีการใช้ใบรับรองสิทธิ์	มีการใช้ใบรับรองสิทธิ์

รูปที่ 3.3 แสดงความสามารถของ IsagQปี 2547เทียบกับ GAIM

จะเห็นได้ชัดว่า GAIM จะมีความสามารถในด้านการบริการข้อความด่วนเหนือกว่า IsagQ แต่ IsagQ รองรับการบริการด้านความปลอดภัยที่เหนือกว่า GAIM ทำให้ IsagQ ในปีนี้จะพัฒนา IsagQเป็นส่วนขยายบนโปรแกรม GAIMซึ่งก็จะได้ใช้ข้อดีของ GAIM และยังคงความสามารถของ IsagQ ได้อย่างครบถ้วน

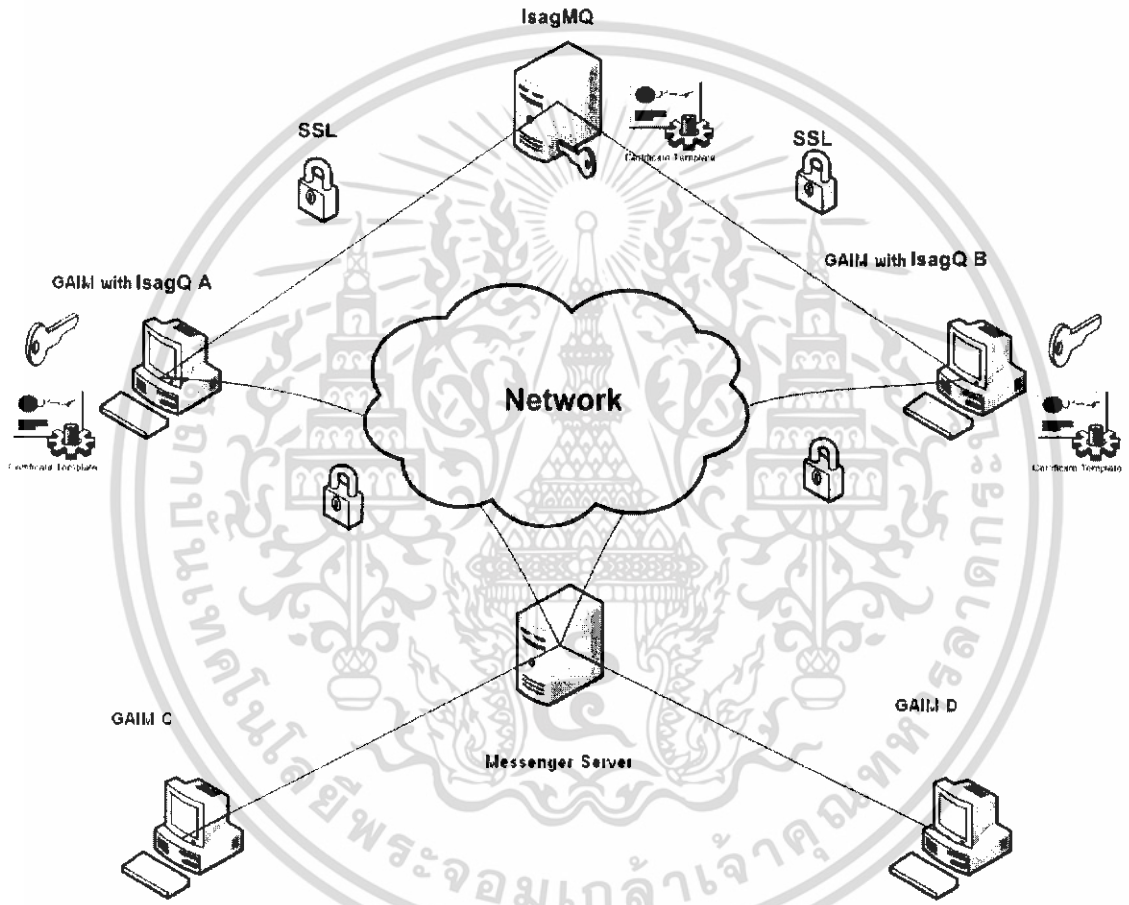
แต่ใน GAIM ก็มีส่วนขยายที่ให้บริการด้านความปลอดภัยชื่อว่า GAIM Encryption ซึ่งใช้ NSS ไลบรารีในการเข้ารหัสแบบ RSAโดยมีความสามารถต่างๆดังนี้

1. สร้างคู่กุญแจสาธารณะและกุญแจส่วนตัวเมื่อเริ่ม plug in โดยอัตโนมัติ
2. ส่งกุญแจสาธารณะให้คู่สนทนาโดยอัตโนมัติ
3. รองรับกุญแจขนาด 512-4096 บิท

4. บันทึกกุญแจของผู้สนทนาที่เคยคุยด้วยและเตือนเมื่อกุญแจสาธารณะเปลี่ยน
5. เก็บกุญแจไว้ไฟล์ที่อ่านได้ในไดเรคทอรีของ GAIM ทำให้เมื่อต้องการใช้สามารถคัดลอกไปได้

เห็นได้ชัดว่า GAIM encryption มีความสามารถในการเข้ารหัสข้อความแต่ก็ไม่มีการพิสูจน์ตัวตน ซึ่งยังไม่สามารถแก้ปัญหา man-in-the-middle ได้

3.3 การออกแบบIsagQและIsagMQ ในปี2548



รูปที่ 3.4 รูปแสดงการโครงสร้างการทำงานระหว่าง IsagQ และ IsagMQ

เมื่อพัฒนา IsagQ 2548 ลงเป็นปลั๊กอินในGAIMซึ่งจะทำให้ทำลายข้อจำกัดของการบริการด้านข้อความของ IsagQ ปี2547 ซึ่ง IsagQ และ IsagMQ ก็จะลดบทบาทลงเป็นเพียงผู้ให้บริการด้านความปลอดภัยแต่ก็จะสามารถนำไปใช้ได้กว้างขวางมากขึ้นโดยผู้ที่จะใช้IsagQนั้นเพียงแค่ลงGAIMแล้วลงปลั๊กอิน IsagQ เพื่อที่จะสามารถส่งข้อความได้แบบปลอดภัย โดยมีฟังก์ชันการทำงานดังนี้ โปรแกรมที่ได้ออกแบบไว้มีการทำงานดังนี้

IsagMQ (โปรแกรมฝั่งแม่ข่าย)

1. ออกใบรับรองสิทธิ์ ให้กับ ผู้ใช้

2. จัดทำฐานข้อมูลของผู้ใช้ที่ลงทะเบียนแล้ว

3. สามารถให้บริการลงทะเบียนและยกเลิกการลงทะเบียนได้

4.สามารถเรียกดูใบรับรองสิทธิ์ได้

5.สามารถถอดถอนใบรับรองสิทธิ์ได้

IsagQ (โปรแกรมฝั่งลูกข่าย)

1. สามารถให้บริการลงทะเบียนและยกเลิกการลงทะเบียนได้

2. ผู้ใช้ติดต่อสื่อสารระหว่างกันอย่างปลอดภัยโดยมีการเข้ารหัสข้อมูลสำหรับข้อมูลระหว่างIsagQ ด้วยกัน

3. ผู้ใช้สามารถติดตั้งบนระบบปฏิบัติการ Windows หรือ Linux ได้

บทที่ 4
การพัฒนาชิ้นงานของโครงงาน

เนื้อหาในบทนี้จะกล่าวถึงการพัฒนาชิ้นงาน ซึ่งประกอบด้วย ส่วนขยายของโปรแกรม Gaim (IsagQ) และส่วนของเซิร์ฟเวอร์ (IsagMQ) ซึ่งนำ OpenSSL Libraly มาช่วยในการเขียนโปรแกรม โดยกล่าวถึงการเตรียมสภาพแวดล้อมก่อนการพัฒนา และกระบวนการพัฒนาจนสำเร็จ

4.1 การนำโอเพนเอสแอลไลบรารี (OpenSSL Library) มาใช้ในโครงงาน

โอเพนเอสแอลไลบรารีเป็น opensource ไลบรารีมีความสามารถในการทำงานในด้านการเข้ารหัสลับและถอดรหัสการสร้างใบรับรองสิทธิ์รวมไปถึงPKI,การสร้างแอพพลิเคชันที่มีการใช้โปรโตคอลSSLหรือTLS และมีความสามาถที่จะใช้แบบ crsoss-platform สามรถใช้ได้ทั้งวินโดว์สและยูนิกซ์ โอเพนเอสเอสแอลไลรารีนี้ใช้ได้ทั้ง c และ c++ แต่ก็สามารถใช้จาก command line ได้ด้วย รวมถึง Python ,Perl ,PHP

ในการสร้าง IsagQ และ IsagMQ นั้นมีการใช้ไลบรารีหลักๆ คือ การเข้ารหัสด้วย RSA การสร้างใบร้องขอใบรับรองสิทธิ์,ใบรับรองสิทธิ์,การจัดการใบรับรองสิทธิ์และการเชื่อมต่อระหว่าง IsagQ และ IsagMQ ด้วย SSL

อาร์เอสเอไลบรารี(RSA Library)

ฟังก์ชันการทำงานรวมถึงโครงสร้างของ RSA จะถูกใช้โดย include ไฟล์ openssl/rsa.h ซึ่งในโครงสร้างของ RSA นั้นจะประกอบด้วยค่าต่างๆดังนี้

```
typedef   struct     rsa_st
{
        BIGNUM *p;
        BIGNUM *q;
        BIGNUM *n;
        BIGNUM *e;
        BIGNUM *d;
}RSA;
```

ค่า p และ q เป็นเลขจำนวนเฉพาะขนาดใหญ่ทั้งคู่แล้วนำมาคูณกันได้ค่า n ซึ่งจะเรียกว่า public modulus ค่า e (public exponent) จะเลือกอยู่ในค่าแบบสุ่มในช่วง (p-1)(q-1) ซึ่งเป็น relatively prime การสร้างกุญแจของอาร์เอสเอ (RSA Key) สร้างโดยใช้ฟังก์ชันต่อไปนี้

```
RSA *RSA_generate_key ( int num unsigmed long e,void(*callback)(int,int,void *),void *cb_arg)
```

- num คือ ค่าความแข็งแกร่งของกุญแจ อย่างต่ำคือ 1024 ควรจะใช้ 2048
- e คือ ค่า public exponent แต่แระนำให้ใช้ RSA_3,RSA_F4
- callback คือ ฟังก์ชันที่จะเรียกเพื่อดูสถานะขณะสร้าง p,q
- cb_arg คือค่าอาร์กิวเมนต์ของ callback

การเข้ารหัสด้วยอาร์เอสเอ (RSA Encryption)

Int RSA_public_encrypt(int flen,unsigned char *from,unsigned char *to,RSA *rsa,int padding)

- flen คือ ค่าความยาวเป็นไบท์ของข้อมูลในบัฟเฟอร์ที่จะถูกเข้ารหัส
- from คือ บัฟเฟอร์ข้อมูลที่จะนำไปเข้ารหัสลับ
- to คือ บัฟเฟอร์ที่จะรับข้อมูลที่เข้ารหัสแล้วโดยจะต้องมีขนาดที่ใหญ่พอ ซึ่งสามารถคำนวนโดยรียกใช้ RSA_size และส่งออบเจกต์ RSA เป็นอาร์กิวเมนต์
- rsa คือ ออบเจกต์ของRSA
- paddingชนิดของแพดดิ้งซึ่งมี4ชนิด คือ RSA_PKCS1_PADDING,RSA_PKCS1_OAEP_PADDING,RSA_SSLV23_PADDING,RSA_NO_PADDING

การเข้าถอดรหัสด้วยอาร์เอสเอ (RSA Decryption)

Int RSA_private_decrypt(int flen,unsigned char *from,unsigned char *to,RSA *rsa,int padding)

- flen คือ ค่าความยาวเป็นไบท์ของข้อมูลในบัฟเฟอร์ที่จะถูกเข้ารหัส
- from คือ บัฟเฟอร์ข้อมูลที่จะนำไปถอดรหัสลับ
- to คือ บัฟเฟอร์ที่จะรับข้อมูลที่ถอดรหัสแล้วโดยจะต้องมีขนาดที่ใหญ่พอ ซึ่งสามารถคำนวนโดยรียกใช้ RSA_size และส่งออบเจกต์ RSA เป็นอาร์กิวเมนต์
- rsa คือ ออบเจกต์ของRSA
- paddingชนิดของแพดดิ้งซึ่งมี4ชนิด คือ RSA_PKCS1_PADDING,RSA_PKCS1_OAEP_PADDING,RSA_SSLV23_PADDING,RSA_NO_PADDING

การสร้าง SSL แอพลิเคชัน โดย Openssl

การสร้าง SSL แอพลิเคชัน โดย Openssl มีขั้นตอนในการสร้างหลักๆ 3 ขั้นตอนคือ

1. การเลือกเอสเอสแอลเวอร์ชัน(SSL version selection)
2. การระบุตัวตนเพียร์ (Peer Authentication)
3. ตัวเลือกของเอสเอสแอลและชุดของไซเฟอร์(SSL Options and Cipher Suites)

ขั้นตอนที่ 1 การเลือกเอสเอสแอลเวอร์ชัน(SSL version selection)

ในขั้นตอนนี้จะเกี่ยวกับออบเจกต์หลักๆอยู่3ชนิกคือ SSL_METHOD,SSL_CTX,และSSL ซึ่ง SSL_METHOD คือ ตัวแทนของการสร้างเอสเอสแอลฟังก์ชัน ซึ่งมีดังนี้

SSLv2_method	Returns a pointer to SSL_METHOD for generic SSL Version 2
SSLv2_client_method	Returns a pointer to SSL_METHOD for an SSL Version 2 client
SSLv2_server_method	Returns a pointer to SSL_METHOD for an SSL Version 2 server
SSLv3_method	Returns a pointer to SSL_METHOD for generic SSL Version 3
SSLv3_client_method	Returns a pointer to SSL_METHOD for an SSL Version 3 client
SSLv3_server_method	Returns a pointer to SSL_METHOD for an SSL Version 3 server
TLSv1_method	Returns a pointer to SSL_METHOD for generic TLS Version 1
TLSv1_client_method	Returns a pointer to SSL_METHOD for a TLS Version 1 client
TLSv1_server_method	Returns a pointer to SSL_METHOD for a TLS Version 1 server
SSLv23_method	Returns a pointer to SSL_METHOD for generic SSL/TLS
SSLv23_client_method	Returns a pointer to SSL_METHOD for an SSL/TLS client
SSLv23_server_method	Returns a pointer to SSL_METHOD for an SSL/TLS server

ตารางที่ 4.1 เวอร์ชันเอสเอสแอล (SSL)

SSL_CTX คือออบเจกต์สำหรับสร้างการเชื่อมต่อแบบเอสเอสแอล ซึ่งคอนเท็กซ์(context) จะอนุญาติให้เราปรับแต่งค่าการเชื่อมต่อก่อนที่จะสร้างการเชื่อมต่อ เช่น เวอร์ชันขิงโปรโตคอล, รายละเอียดของใบรับรองสิทธิ์ ซึ่งการสร้างออบเจกต์ชนิดนี้ใช้ SSL_CTX_new

ขั้นตอนที่ 2 การระบุตัวตนเพียร์ (Peer Authentication)

การพิสูจน์ตัวตนของใบรับรองสิทธิ์ผู้ที่ตรวจสอบจำเป็นที่จะต้องมีรายการของ CA ดังนั้นเราจึงต้องบอกแอพลิเคชันว่ารายการนั้นอยู่ที่ไหนเพื่อที่จะพิสูจน์ตัวตนของเพียร์ การที่จะใช้ที่เก็บรายการของ CA จำเป็นที่จะเพิ่มส่วนการตระเตรียมออบเจกต์ SSL_CTX ด้วย

int SSL_CTX_load_verify_locations(SSL_CTX *ctx, const char *CAfile, const char *CApath),

ctx คือ เอสเอสแอลคอนเท็กซ์ออบเจกต์ที่จะใช้ใบรับรองสิทธิ์

CAfile คือ ชื่อไฟล์ใบรับรองสิทธิ์ในรูปแบบของ PEM

CApath คือ ชื่อไดเรกทอรีที่เก็บไฟล์ใบรับรองสิทธิ์

ขั้นตอนที่ 3 ตัวเลือกของเอสเอสแอลและชุดของไซเฟอร์(SSL Options and Cipher Suites)

ฟังก์ชัน SSL_ctx_set_options ทำให้ผู้ใช้สามารถใช้งาน bug workaround ซึ่งสามารถทำให้เอสแอลแอพลิเคชันสามารถเชื่อมต่อกับเพียร์ที่มีปัญหาได้หรือสามารถจำกัดระดับความปลอดภัยของการเชื่อมต่อเอสเอสแอล เช่น เลือกใช้ SSL_OP_NO_SSLv2 ซึ่งการใช้ทางเลือกนี้จะทำให้การเชื่อมต่อนี้จะไม่รับโพรโตคอลเอสเอสแอลเวอร์ชัน 2

การเลือกชุดของไซเฟอร์

ชุดของไซเฟอร์คือชุดของอัลกอริทึมที่เอสเอสแอลใช้ในการเชื่อมต่อแบบปลอดภัย ซึ่งจะใช้ฟังก์ชัน SSL_CTX_set_cipher_list ทำให้เราสามารถใช้รายชื่อของชุดของไซเฟอร์ได้ โดยรายชื่อของชุดของไซเฟอร์จะอยู่ในรูปแบบพิเศษคือ คั่นด้วย : ตัวอย่างเช่น

"ALL:!ADH:!LOW:!EXP:!MD5:@strength"

- ALL คือ การใช้ตัวแทนของทุกเอาอัลกอริธึมมาคอมบิเนชันกัน
- ! คือ การยกเว้นอัลกอริธึม
- ADH คือ การใช้อัลกอริธึม Diffie Helman
- Low คือ การอ้างถึงอัลกอริธึมที่ใช้คีย์ 64 บิทหรือ 56 บิท
- EXP คือ การอ้างถึงอัลกอริธึมที่ใช้คีย์ 56 บิทหรือ 40 บิทแบบคลิปเปิล
- @strength คือ การเรียงไซเฟอร์ตามขนาดคีย์

4.2 การพัฒนาส่วนขยายบน Gaim (IsagQ)

ในการพัฒนาปลั๊กอินของGAIMแบ่งได้เป็น2ภาษาคือภาษาซีและภาษาเพิร์ลซึ่งในโครงงานจะใช้ภาษาซีในการพัฒนาส่วนขยายต้องใช้ซอร์สโค้ดของ Gaim ในการคอมไพล์โดยจะมี Gaim apiรองรับการทำงานต่างๆ และสามารถนำไลบรารีจากที่อื่นมาใช้ได้

4.2.1 การเตรียมสภาพแวดล้อมในการพัฒนาส่วนขยายของ Gaim

การเตรียมสภาพแวดล้อมในการพัฒนาส่วนขยายของ Gaim มีขั้นตอนดังนี้

1) ดาวน์โหลดซอร์สโค้ดของ Gaim เวอร์ชัน 1.5
2) เข้าไปที่โฟลเดอร์ Gaim-1.5.0 แล้วสั่ง ./configure
3) ในขั้นนี้เราจะได้ Make file ขึ้นมา
4) ถ้าต้องการincludeไลบรารีใดๆที่ต้องใช้ในส่วนขยายให้เข้าไปที่Gaim-1.5.0/plugins/ Makefile เข้าไปให้ระบุในไฟล์นี้ เช่น เราต้องการใช้ไลบรารีของ Openssl ซึ่งจะเก็บไว้ที่ /usr/local/include ก็เพิ่ม –I/usr/local/include
5) เมื่อต้องการคอมไพล์

   5.1 Window สั่ง make –f makefile.mingw **name**.dll

   5.2 Unix/linux สั่ง make **name**.so

   ซึ่ง **name** คือชื่อไฟล์ .c ที่ต้องการคอมไพล์
6) เมื่อเสร็จสิ้นคำสั่ง make ให้คัดลอกไฟล์ไปยังโฟลเดอร์ที่เก็บไลบรารีของ Gaim ไว้เช่น /usr/X11R6/lib/gaim

4.2.2 การพัฒนาส่วนขยาย IsagQ

โครงสร้างไฟล์ส่วนขยายหลักของ Gaim

ในไฟล์หลักของส่วนขยายนั้นจะมีรูปแบบที่ตายตัวและจำเป็นที่จะต้องใช้ตามรูปแบบเพื่อที่จะสามารถใช้ส่วนขยายได้โดยแบ่งออกได้ดังนี้

1 ส่วนที่หนึ่งเป็นการ include file และ #define GAIM_PLUGINS ซึ่งต้องมีไฟล์อย่างน้อยดังนี้

```
#define GAIM_PLUGINS                    #ต้องประกาศในทุกปลั๊กอิน
#include <glib.h>
#include "notify.h"
#include "plugin.h"
#include "version.h"
static gboolean
plugin_load(GaimPlugin *plugin) {       #ฟังก์ชันนี้จะทำงานตอนที่โหลดปลั๊กอิน
        return TRUE;
}
```

2 ส่วนที่สองคือการตั้งค่าต่างๆ ของส่วยขยาย

```
static GaimPluginInfo info = {    #ฟังก์ชันนี้จะเป็นรายละเอียดที่แสดงในconfig window
                                  (จำเป็นต้องมี)
        GAIM_PLUGIN_MAGIC,              #เพื่อหลีกเลี่ยง crash เวลาโหลด
        GAIM_MAJOR_VERSION,             #บอกเวอร์ชัน
        GAIM_MINOR_VERSION,             #บอกเวอร์ชัน
        GAIM_PLUGIN_STANDARD,           #ชนิดของปลั๊กอิน
        NULL,                           #ui requirement
        0,                              #plugin flags
        NULL,                           #plugin dependencie
        GAIM_PRIORITY_DEFAULT,          #ไพรออริตี้ของปลั๊กอิน
        "core-hello_world",             #ไอดี
        "Hello World!",                 #ชื่อ
        VERSION,                        #เวอร์ชันของปลั๊กอิน
        "Hello World Plugin",           #รายละเอียดโดยย่อของปลั๊กอิน
        "Hello World Plugin",           #รายละเอียดของปลั๊กอิน
        NULL,                           #ชื่อและอีเมลล์
        GAIM_WEBSITE,                   #เวบ
        plugin_load,                    #พอย์นเตอร์ชี้ไปยังฟังก์ชันที่จะโหลด
                                        ตอนเริ่มปลั๊กอิน
        NULL,                           #พอย์นเตอร์ชี้ไปยังฟังก์ชันที่จะโหลด
                                        ตอนปิดปลั๊กอิน
```

```
    NULL,                   #พอย์นเตอร์ชี้ไปยังฟังก์ชันที่จะโหลด
                            ตอนทำลายปลั๊กอิน
    &ui_info,                   #พอย์นเตอร์ชี้ไปยัง UI struct

    NULL,                   #พอย์นเตอร์ชี้ไปยัง GaimPluginLoaderInfo
                            และGaimPluginProtocolInfo
    NULL,                   #พอย์นเตอร์ชี้ไปยัง GaimPluginUiInfo
    NULL                    #พอย์นเตอร์ชี้ไปยังฟังก์ชันที่จะโหลดเวลาเรียกเมนู
};
```

3 . ส่วนที่ 3 จำเป็นต้องประกาศไว้เสมอ ถ้าไม่ประกาศจะไม่สามารถเรียกใช้ส่วนขยายได้

```
static void
init_plugin(GaimPlugin *plugin) {                   #ฟังก์ชันที่เรียกเวลาตรวจสอบปลั๊กอิน
}
GAIM_INIT_PLUGIN(hello_world, init_plugin, info);   #เป็นมาโครที่ทุกปลั๊กอินต้องมี
```

การเขียนหน้าจอการตั้งค่าส่วนขยาย

เมื่อปลั๊กอินโหลดขึ้นมาเราสามารถที่จะสร้างหน้าจอการตั้งค่าได้โดยจะระบุไว้ในฟังก์ชัน GaimPluginInfo info ในอาร์กิวเมนต์ตัวที่ 19 โดยระบุเป็น pointer เช่น &ui_info ซึ่งเรียกฟังก์ชัน ui_info มีลักษณะดังนี้

```
static GaimGtkPluginUiInfo ui_info =
{
    get_config_frame
};
```

ซึ่งจะไปเรียกฟังก์ชัน get_config_frame ซึ่ง ui_info จะเป็นเพียงฟังก์ชันที่บอก Gaim ว่ามีการสร้างหน้าจอการตั้งค่าส่วนขยายซึ่งฟังก์ชันที่สร้างหน้าจอการตั้งค่าจริง ๆ คือ get_config_frame

การตั้งค่าการทำงานเริ่มต้นของปลั๊กอิน

ก่อนที่ตัวปลั๊กอินจะเริ่มทำงานเราสามารถสั่งให้ตัวปลั๊กอินทำงานบางอย่างก่อนได้ โดยใช้ฟังก์ชัน plugin_load (GaimPlugin *handle) ซึ่ง IsagQ จะมีการทำงาน 2 อย่างในช่วงนี้คือ Init_pref() = สร้าง preference ที่สามารถบันทึกค่าลักษณะต่างๆของปลั๊กอินเช่น

- gaim_prefs_add_none สร้าง preference
- gaim_prefs_add_int บันทึกค่าลงไปใน preference ที่สร้างแล้ว
- gaim_prefs_get_int รับค่าจาก preference ที่บันทึกไว้

Init_cert() = สร้างกุญแจสาธารณะ,กุญแจส่วนตัว,ใบร้องขอใบรับรองสิทธิ์,ใบรับรองสิทธิ์

สัญญานบน Gaim (Gaim signal)

เมื่อผู้ใช้หรือคู่สนทนามีการตอบโต้กับโปรแกรม(event)ตัว Gaim ก็จะส่งสัญญาณ (Signal) ออกมาโดยเราสามาถรถจับสัญญาณเหล่านี้ได้โดยใช้ gaim_signal_connect แต่ก่อนที่จะจับสัญญาณได้จะต้องสร้าง handle สำหรับเชื่อมต่อก่อน ในตัว IsagQ จะมีการจับสัญญาณการทำงาน 2 อย่างคือ สัญญานการส่งข้อความ "sending-im-msg" และสัญญาณการรับข้อความ "receiving-im-msg" ซึ่งมีรูปแบบการทำงานดังนี้

```
gaim_signal_connect(conv_handle, "sending-im-msg", isagq_plugin_handle,
        GAIM_CALLBACK(send_msg_cb), NULL);
```

ฟังก์ชันนี้จะมีอาร์กิวเมนต์ 5 ตัวคือ

- conv_handle คือ แฮนเดิลของการสทนาเนื่องจากฟังก์ชันที่จะทำงานนี้เกี่ยวข้องกับการสนทนา ซึ่ง แฮนเดิลนี้สามารถเรียกใช้ได้จาก gaim_conversations_get_handle
- "sending-im-msg" คือ ชื่อของสัญญาณที่เราต้องการจะจับ
- isagq_plugin_handle คือ แฮนเดิลของปลั๊กอินIsagQ ซึ่งจะถูกสร้างในฟังก์ชัน plugin_load
- GAIM_CALLBACK(send_msg_cb) คือ การเรียกใช้ฟังก์ชัน send_msg_cb เมื่อมีสัญญาณการส่งข้อความ
- NULL คือ อาร์กิวเมนต์ของฟังก์ชัน send_msg_cb

การจัดการกับการส่งข้อความและการรับข้อความ

เมื่อเราสามารถจับสัญญาณของการส่งข้อความและการรับข้อความได้แล้วก็จะสามารถจัดการกับข้อความเหล่านั้นก่อนที่จะแสดงผลได้ โดยฟังก์ชันที่จะจัดการกับข้อความจะต้องมีอาร์กิวเมนต์ต่อไปนี้

- GaimAccount *account อาร์กิวเมนต์นี้จะมีรายละเอียดของผู้ใช้งานที่ได้รับหรือส่งข้อความนั้นๆ เช่น ชื่อผู้ใช้,โปรโตคอลที่ใช้
- char *who อาร์กิวเมนต์นี้คือ ชื่อของคู่สนทนา
- char **message คือ ข้อความที่ได้รับหรือส่ง

เมื่อเราจัดการกับข้อความเสร็จแล้วก็จะสามารถนำส่งหรือแสดงผลได้ โดยการส่งข้อความจะใช้ serv_send_im(server-send-im)หรือถ้าต้องการส่งอย่างอื่นก็ได้เช่นserv_send_file serv_send_typing โดย serv_send_im มีอาร์กิวเมนต์ต่างๆดังนี้

- GaimConnection * คือ ตัวชี้ไปยังการเชื่อมต่อกับผู้สนทนาซึ่งจะเป็น attribute อยู่ในตัวแปรชนิด GaimAccount ชื่อ gc
- Const char* คือ ชื่อของผู้สนทนา

- Const char* คือ ข้อความที่จะต้องการส่ง
- GaimMessageFlags คือชนิดข้อความ เช่นGAIM_MESSAGE_SEND, GAIM_MESSAGE _SYSTEM

เช่น (acct->gc, name,cipher_msg,GAIM_MESSAGE_AUTO_RESP)

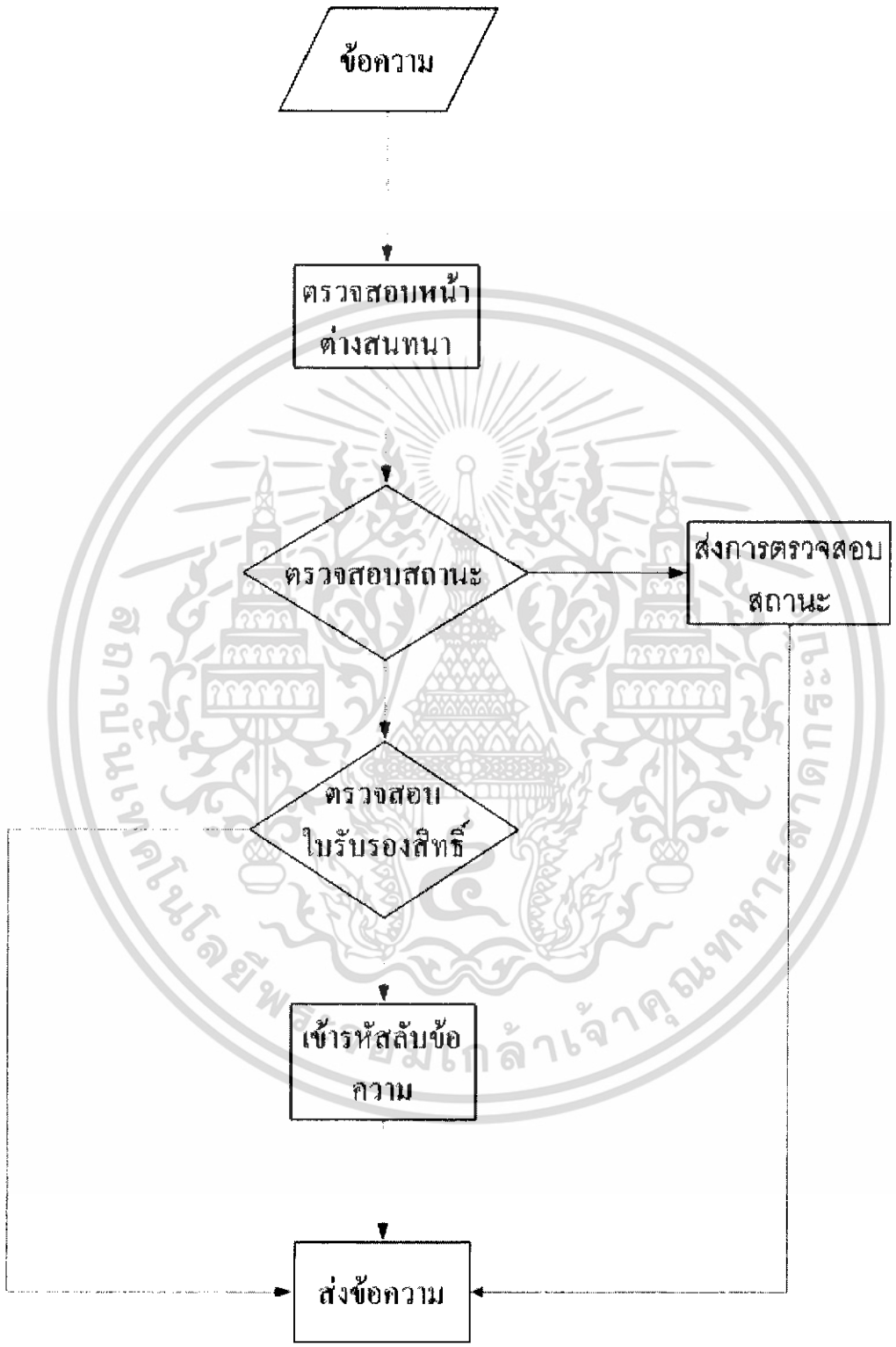
ทางด้านการแสดงผลข้อความก็จะใช้ gaim_conv_im_write โดยมีอาร์กิวเมนต์ดังนี้

- GaimConvIm คือ ตัวชี้ไปยังหน้าต่างสนทนานั้นๆ
- Const char* คือ ชื่อของผู้สนทนา
- Const char* คือ ข้อความที่จะแสดงผล
- GaimMessageFlags คือ ชนิดของข้อความ
- time_t คือ การหน่วงเวลาก่อนที่จะแสดงผล

เช่น gaim_conv_im_write(GAIM_CONV_IM(conv), *who, decrypt_msg, GAIM_MESSAGE_RECV, time((time_t)NULL))

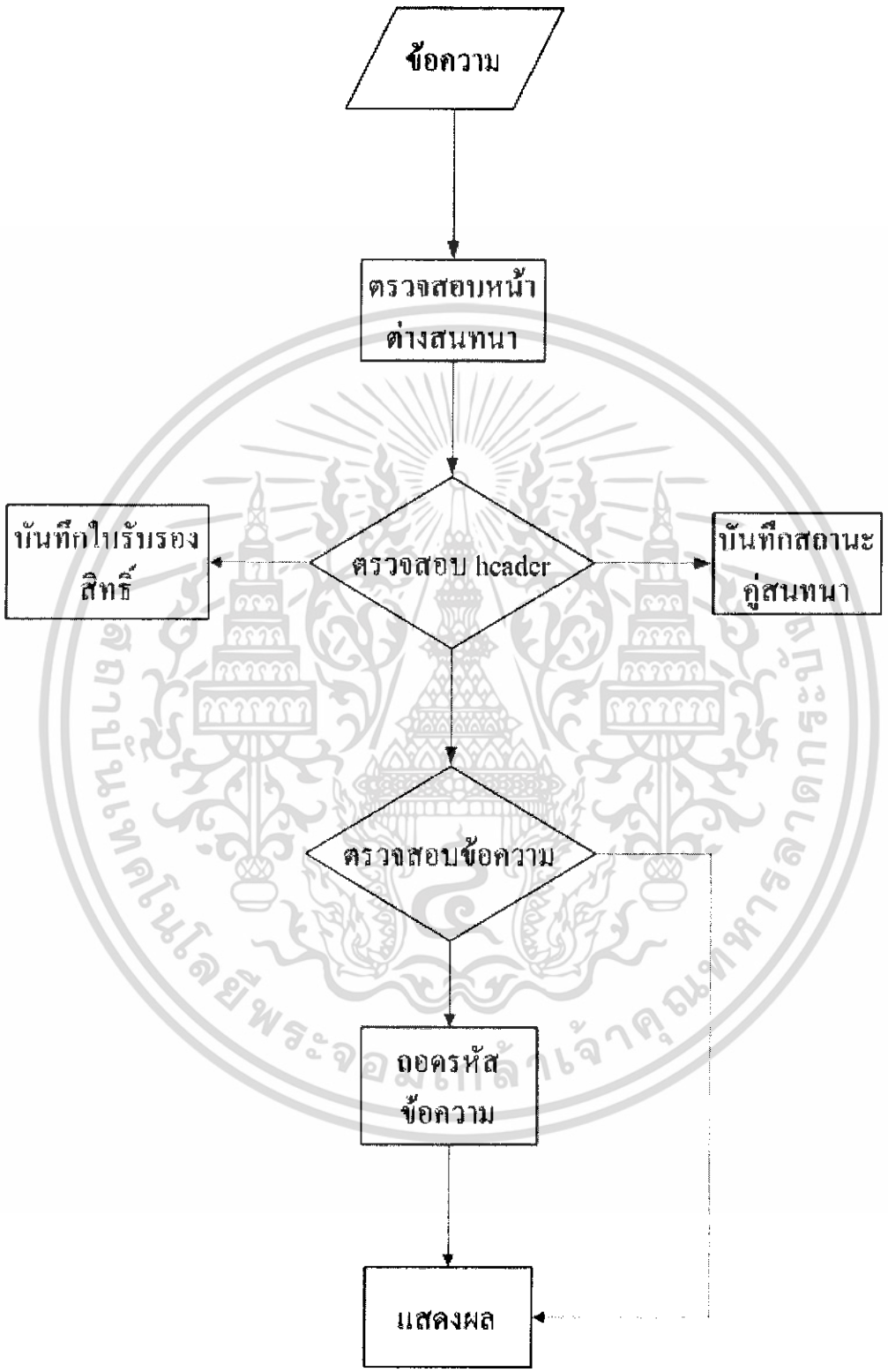
4.2.3 กระบวนการรับ-ส่งข้อความของ IsagQ

ขั้นตอนการส่งข้อความของ IsagQ



รูปที่ 4.1 ขั้นตอนการส่งการข้อความของ IsagQ

ขั้นตอนการรับข้อความของ IsagQ



รูปที่ 4.2 ขั้นตอนการรับข้อความของ IsagQ

4.3 การพัฒนาส่วนเซิร์ฟเวอร์ (IsagMQ)

Certificate Authority เป็นบทบาทของส่วนเซิฟเวอร์ ซึ่งจะสร้าง Root CA และ Server CA ซึ่ง Server CA จะเป็นผู้ sign ให้กับ Client หน้าที่ของ CA คือ การออกใบรับรองสิทธิ์ การเรียกคืนใบรับรองสิทธิ์ การเก็บข้อมูลของผู้ขอใบรับรองสิทธิ์ และการเก็บข้อมูลของใบรับรองสิทธิ์ที่ถูกเรียกคืนหรือหมดอายุแล้ว

4.3.1 การสร้างผู้ออกใบรับรองสิทธิ์ (Certification Authority)

เซิฟเวอร์IsagMQพัฒนาบนระบบปฏิบัติการลีนุกซ์ FreeBsd โดยต้องติดตั้งไลบรารี่ Openssl มีหน้าที่หลักคือ Certification Authority ซึ่งจัดการเรื่องใบรับรองสิทธิ์ การสร้าง CA มีดังนี้

โดยอย่างแรกเราสร้างโฟลเดอร์เพื่อเก็บข้อมูลทั้งหมดของCAนั่นคือใบรองสิทธิ์และ CRL จากนั้นเราจะสร้างไฟล์ 3 ไฟล์ ไฟล์แรกคือไฟล์ serial ซึ่งเก็บ serial number สุดท้ายที่ถูกใช้ในการออกใบรับรองสิทธิ์เพื่อไม่ให้ serial numberซ้ำกัน ไฟล์ที่สองคือ ไฟล์ index.txt ซึ่งเก็บฐานข้อมูลของใบรับรองสิทธิ์ซึ่งออกโดยCA และไฟล์สุดท้ายคือไฟล์ configuration ซึ่งเก็บข้อมูลเกี่ยวกับวิธีการออกใบรับรองสิทธิ์

- ไฟล์ configuration มีดีฟอลท์อยู่แล้วเมื่อลง Openssl Libraly แต่เนื่องจากโครงงานนี้ต้องกำหนดค่าต่างๆเองจึงสร้างไฟล์ configuration ขึ้นมาใหม่ดังนี้

```
[ ca ]
default_ca              = exampleca

[ exampleca ]
dir                     = /root/careCA
certificate             = $dir/rootcert.pem
database                = $dir/index.txt
new_certs_dir           = /root/careClient
private_key             = $dir/rootkey.pem
serial                  = $dir/serial
crl                     = $dir/crl.pem #the current CRL
crl_dir                 = $dir/crl
default_crl_days        = 7
default_days            = 365
```

```
default_md              = md5
policy                  = examplecа_policy
x509_extensions         = certificate_extensions

[ examplecа_policy ]
commonName              = supplied
stateOrProvinceName     = supplied
countryName             = supplied
emailAddress            = supplied
organizationName        = supplied
organizationalUnitName  = optional

[ certificate_extensions ]
basicConstraints        = CA:false

[ req ]
default_bits            = 2048
default_md              = md5

prompt = no


x509_extensions         = root_ca_extensions


[ root_ca_extensions ]
        basicConstraints       = CA:true
```

- สร้าง ROOT CA โดย sign ด้วยตัวมันเอง

```
# openssl req -newkey rsa:1024 -sha1 -keyout rootkey.pem -out rootreq.pem
```

openssl x509 -req -in rootreq.pem -sha1 -extfile careopenssl.cnf -signkey rootkey.pem -out rootcert.pem

#cat rootcert.pem rootkey.pem >root.pem

#openssl x509 -subject -issuer -noout -in root.pem

subject= /CN=care/emailAddress=pare_g7@hotmail.com issuer= /CN=care/emailAddress=pare_g7@hotmail.com

- สร้าง SERVER CA และ sign ด้วย ROOT CA

#openssl req -newkey rsa:1024 -sha1 -keyout serverCAkey.pem -out serverCAreq.pem

#openssl x509 -req -in serverCAreq.pem -sha1 -extfile careopenssl.cnf -CA root.pem -CAkey root.pem -CAcreateserial -out serverCAcert.pem

cat serverCAcert.pem serverCAkey.pem rootcert.pem > serverCA.pem

openssl x509 -subject -issuer -noout -in serverCA.pem

subject= /CN=pare_g7/emailAddress=pare_g72@hotmail.com

issuer= /CN=care/emailAddress=pare_g7@hotmail.com

- สร้าง SERVER'S CERTIFICATE และ sign ด้วยตัวมันเอง

openssl req -newkey rsa:1024 -sha1 -keyout serverkey.pem -out serverreq.pem

openssl x509 -req -in serverreq.pem -sha1 -extfile careopenssl.cnf -CA serverCA.pem -CAkey serverCA.pem -CAcreateserial -out servercert.pem

cat servercert.pem serverkey.pem serverCAcert.pem rootcert.pem >server.pem

openssl x509 -subject -issuer -noout -in server.pem

subject= /CN=pareserver/emailAddress=pare_g73@hotmail.com\x1B[D

issuer= /CN=pare_g7/emailAddress=pare_g72@hotmail.com

- การสร้าง Certificate Revocation List (CRL)

openssl ca –gencrl –config /root/careCA/careopanssl.conf -out /root/careCA/crl.crl

สร้าง CRL โดยกำหนด option ไว้ในไฟล์ careopanssl.cnf ดังนี้

Default crl days = 7

Default days = 365

Default md = md5

4.3.2 กระบวนการออกใบรับรองสิทธิ์

- CLIENT หรือIsagQ สร้าง CERTIFICATE REQUEST

```
# gcc -o certReq certReq.c -lssl -lpthread
# ./certReq priKey.pem certReq.pem
```

- CA สร้าง CLIENT CERTIFICATE

```
# gcc -o createCert createCert.c -lssl -lpthread
# ./createCert certReq.pem clientCert.pem
```

- CA พิสูจน์ CLIENT CERTIFICATE

```
#gcc -o verifycert verifycert.c -lssl lpthread
# ./verifycert.c clientCert.pem
```

Client Certificate จะถูก sign โดย Server CA ซึ่งจะเก็บรายละเอียดของไคลเอนต์ หมายเลขซีเรียล กุญแจสาธารณะของไคลเอนต์ และ ลายเซ็นของ Server CA

```
isagmq# openssl x509 -in client.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            fb:54:de:11:9e:8e:35:26
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=care/emailAddress=pare_g7@hotmail.com
        Validity
            Not Before: Dec 23 12:15:44 2005 GMT
            Not After : Jan 22 12:15:44 2006 GMT
        Subject: CN=rach/emailAddress=rach@hotmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:b8:56:50:ca:1f:10:48:c2:5e:a3:2b:f0:30:06
:
                    c3:2e:47:26:58:b5:4d:d9:30:2c:24:c8:86:84:3d
:
                    cd:48:4b:2a:b0:96:f7:c7:55:ba:dc:cc:53:80:54
:
                    ef:ba:61:74:14:bd:6f:84:c2:cd:f0:4b:46:ec:64
:
                    30:17:66:56:2f:93:25:ec:d3:49:49:73:b3:94:0b
:
                    c6:d2:f6:5e:f1:1a:58:4b:f7:eb:19:ed:0f:36:e1
:
                    d2:28:b6:e8:b9:3b:65:16:16:ac:15:87:f0:58:79
:
                    c8:4b:b4:ff:40:46:57:fb:00:c6:3e:9c:20:c6:96
:
                    cb:be:31:38:3d:fe:1d:07:db
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
        64:a3:9a:5d:3d:fb:8f:6b:d5:86:53:88:14:70:0f:37:dc:37:
        58:d5:f6:ad:3f:86:21:a5:d6:7d:11:fe:88:bc:23:3d:7a:39:
        cf:ee:dc:93:3c:89:2e:5d:6b:8d:0d:10:e0:37:2f:98:17:d6:
        35:f9:5d:31:96:d7:2c:23:e9:0f:3c:b4:ec:7d:bb:51:4d:1e:
        a6:fc:02:7a:b8:9e:d9:00:89:fc:7c:6b:33:f7:52:5e:07:f2:
        8e:ac:0a:5e:f0:54:8b:ff:fe:e6:d4:21:cd:5e:fa:bf:d5:12:
        b1:62:f0:b9:ae:f3:1f:e5:ac:a9:4b:6a:e0:b2:1a:93:76:ee:
        23:fc
```

รูปที่ 4.3 client certificate

Certificate Revocation List (CRL) มีเพื่อเก็บ Serial Number ที่ถูกเรียกคืนโดย CA CRLมีอายุการใช้งาน มีการเก็บค่าเวอร์ชั่น รายละเอียดของผู้สร้าง CRL วันที่ออกและหมดอายุของ CRL และ ลายเซ็นของผู้สร้างนั่นคือ Server CA

```
isagmq# openssl crl -in crl2.crl -noout -text
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: /CN=care/emailAddress=pare_g7@hotmail.com
        Last Update: Dec 27 07:25:50 2005 GMT
        Next Update: Dec 27 08:25:50 2005 GMT
Revoked Certificates:
    Serial Number: FB54DE119E8E3523
        Revocation Date: Dec 25 17:05:41 2005 GMT
    Serial Number: FB54DE119E8E3524
        Revocation Date: Dec 25 17:03:04 2005 GMT
    Serial Number: FB54DE119E8E3525
        Revocation Date: Dec 25 16:29:35 2005 GMT
    Serial Number: FB54DE119E8E3526
        Revocation Date: Dec 25 17:19:24 2005 GMT
    Signature Algorithm: md5WithRSAEncryption
        48:16:eb:0c:0d:0a:e4:0a:a3:dc:25:f3:98:2b:ef:06:65:6b:
        db:ff:23:ed:93:f2:74:7f:c2:52:e3:1a:cb:fe:14:93:3f:a5:
        8b:5c:e9:d2:5c:c3:0f:99:3f:f1:44:fe:fe:61:85:b0:ed:c2:
        83:51:26:d7:26:c7:89:c4:05:d3:76:b8:08:a0:b8:9a:cf:d0:
        02:a0:23:2b:ac:1c:1a:48:11:c3:d1:9e:28:75:05:cc:c3:55:
        9d:16:bb:a1:a6:be:25:68:cf:45:71:e8:3c:5d:3b:19:f8:5b:
        0f:d5:d0:ee:b5:26:4b:63:3a:3e:50:2d:74:b4:67:03:5d:0a:
        e4:92
```

รูปที่ 4.4 Certificate Revocation List (CRL)

บทที่ 5

ผลการทดสอบ IsagQ และ IsagMQ

5.1 ผลการทดสอบการส่งการร้องขอใบรับรองสิทธิ์ระหว่าง IsagQ และ IsagMQ

เมื่อไคลเอนต์ลงโปรแกรม Gaim และลงส่วนขยาย IsagQ ผู้ใช้จะต้องเลือกขนาดกุญแจส่วนตัวและกุญแจสาธารณะ ซึ่งมีให้เลือกขนาด 1024 ไบท์ 2048 ไบท์ และ 4096 ไบท์ จากนั้นไคลเอนต์จะทำกระบวนการขอใบรับรองสิทธิ์

5.1.1 IsagQ สร้างใบร้องขอใบรับรองสิทธิ์

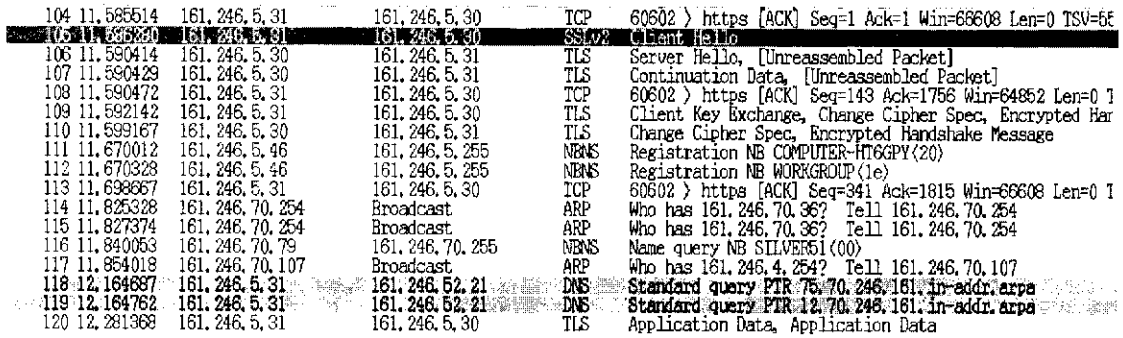
ในการร้องขอใบรับรองสิทธิ์นั้นในขั้นตอนที่ 1 IsagQ จะสร้างใบร้องขอใบรับรองสิทธิ์จากกุญแจสาธารณะของผู้ใช้ที่ต้องการร้องขอใบรับรองสิทธิ์โดยขนาดของใบรับรองสิทธิ์ก็จะขึ้นกับขนาดของกุญแจสาธารณะโดยใบร้องขอใบรับรองสิทธิ์ดังรูป 5.1

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCADB1MQwwCgYDVQQIEwNCS0sxEjAQBgNVBAcTCVNVQU5MVUFORzEM
MAoGAQATBUtNSVRMMQswCQYBABMESXNhZzEjMCEGCgmSJomT8ixkAQMUE3BhcmVf
ZzdAaG90bWFpbC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgYOAMIGJAoGBAMjtYD99
oK2afE/2raUc+XAgnmDoXCI7v169mBPZkdK6+HwFfk62MZ2xkshikrTMBrYLXM/c
4VqzpCpKSdNNr3693xxLyx3tZFgJXDPnZLJ1MYUeemubd6sIQ6ont2za7hRCDEFs
HQXIyQPfAZnOp9vpmPzXsAGSqNa+nTjHFDZ1AgMBAAGgADANBgkqhkiG9w0BAQQF
AAOBgQAOB3+xditCP4HHhFGD7FA/SDwibqOSh8mlpf/IqrkspxcSSkMcbQ1Pa7pG
Zs1PXWGT4NlF0OjQOCzvfqb2+kmvnVNRnKnXArDd+n9/yYtB1mGxQlqHMCruQpGI
rjWzC2j7ApzgO9Mr2e1rFIpNajWwnxwEm6a2nllWm2Pmw02dnw==
-----END CERTIFICATE REQUEST-----
```

รูปที่ 5.1 แสดงใบร้องขอใบรับรองสิทธิ์

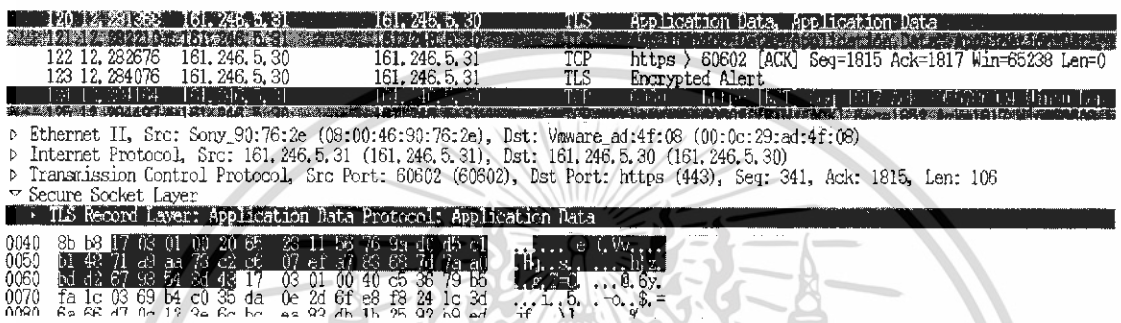
5.1.2 IsagQและIsagMQสร้างการเชื่อมต่อแบบ SSL

เมื่อ IsagQ สร้างใบรับรองสิทธิ์เสร็จแล้วก็จะสถาปนาการเชื่อมต่อแบบ SSL กับ IsagMQ และส่งข้อมูลแบบปลอดภัย เมื่อใช้โปรแกรม Etehreal จับดูแพ็กเกตที่ส่งระหว่าง IsagQ และ IsagMQ จะเห็นได้ว่าทั้งสองฝั่งมีการแลกเปลี่ยนกุญแจสาธารณะและใบรับรองสิทธิ์ โดยแสดงดังรูป 5.2



104	11.585514	161.246.5.31	161.246.5.30	TCP	60602 > https [ACK] Seq=1 Ack=1 Win=66608 Len=0 TSV=65
105	11.586820	161.246.5.31	161.246.5.30	SSLv2	Client Hello
106	11.590414	161.246.5.30	161.246.5.31	TLS	Server Hello, [Unreassembled Packet]
107	11.590429	161.246.5.30	161.246.5.31	TLS	Continuation Data, [Unreassembled Packet]
108	11.590472	161.246.5.31	161.246.5.30	TCP	60602 > https [ACK] Seq=143 Ack=1756 Win=64852 Len=0 T
109	11.592142	161.246.5.31	161.246.5.30	TLS	Client Key Exchange, Change Cipher Spec, Encrypted Har
110	11.599167	161.246.5.30	161.246.5.31	TLS	Change Cipher Spec, Encrypted Handshake Message
111	11.670012	161.246.5.46	161.246.5.255	NBNS	Registration NB COMPUTER-HT6GPY<20>
112	11.670328	161.246.5.46	161.246.5.255	NBNS	Registration NB WORKGROUP<1e>
113	11.698667	161.246.5.31	161.246.5.30	TCP	60602 > https [ACK] Seq=341 Ack=1815 Win=66608 Len=0 T
114	11.825328	161.246.70.254	Broadcast	ARP	Who has 161.246.70.36? Tell 161.246.70.254
115	11.827374	161.246.70.254	Broadcast	ARP	Who has 161.246.70.36? Tell 161.246.70.254
116	11.840053	161.246.70.79	161.246.70.255	NBNS	Name query NB SILVER51<00>
117	11.854018	161.246.70.107	Broadcast	ARP	Who has 161.246.4.254? Tell 161.246.70.107
118	12.164687	161.246.5.31	161.246.52.21	DNS	Standard query PTR 75.70.246.161.in-addr.arpa
119	12.164762	161.246.5.31	161.246.52.21	DNS	Standard query PTR 12.70.246.161.in-addr.arpa
120	12.281368	161.246.5.31	161.246.5.30	TLS	Application Data, Application Data

รูปที่ 5.2 แสดงการเชื่อมต่อแบบ SSL



120	12.281368	161.246.5.31	161.246.5.30	TLS	Application Data, Application Data
122	12.282676	161.246.5.30	161.246.5.31	TCP	https > 60602 [ACK] Seq=1815 Ack=1817 Win=65238 Len=0
123	12.284076	161.246.5.30	161.246.5.31	TLS	Encrypted Alert

▷ Ethernet II, Src: Sony_90:76:2e (08:00:46:90:76:2e), Dst: Vmware_ad:4f:08 (00:0c:29:ad:4f:08)
▷ Internet Protocol, Src: 161.246.5.31 (161.246.5.31), Dst: 161.246.5.30 (161.246.5.30)
▷ Transmission Control Protocol, Src Port: 60602 (60602), Dst Port: https (443), Seq: 341, Ack: 1815, Len: 106
▽ Secure Socket Layer
▷ TLS Record Layer: Application Data Protocol: Application Data

```
0040  8b b8 17 03 01 00 20 65  28 11 f5 76 9a dc f5 c1   ..... e (..v....
0050  bf 48 71 ed aa 78 c2 cc  07 ef a7 e5 68 7d 7a a0   .Hq..x.. ....h}z.
0060  bd d2 67 98 24 81 43 17  03 01 00 40 c5 36 79 b5   ..g.$.C. ...@.6y.
0070  fa 1c 03 69 b4 c0 35 da  0e 2d 6f e8 f8 24 1c 3d   ...i..5. .-o..$.=
0080  6a 65 d7 0a 18 3e 6c bc  ee 92 db 1b 25 92 b0 ed   je...>l. ....%...
```

รูปที่ 5.3 แสดงการดักจับข้อมูลที่ส่งระหว่าง IsagQ และ IsagMQ

จากรูป 5.3 เมื่อสถาปนาการเชื่อมต่อด้วย SSL ข้อมูลที่ส่งผ่านจะมีการเข้ารหัสทำให้มั่นใจได้ว่าข้อมูลที่ส่งผ่านนั้นไม่ถูกเปิดเผยต่อบุคคลอื่น

5.1.3 IsagMQสร้างใบรับรองสิทธิ์

เมื่อ IsagMQ ได้รับใบร้องขอใบรับรองสิทธ์แล้วก็จะนำไปสร้างใบรับรองสิทธิ์โดยจะมีการใส่ข้อมูลที่เกี่ยวข้องของ CA ลงไป

```
-----BEGIN CERTIFICATE-----
MIICNzCCAaCgAwIBAgIJAPBrNjrAkP3RMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDVQQIEwpTb21lLVN0YXRlMSEwHwYDVQQKExhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMDUxMTI2MDk1ODU2WhcNMDUxMjI2MDk1ODU2WjBn
MQswCQYDVQQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50
ZXJuZXQgV2lkZ2lOcyBQdHkgTHRkMSAwHgYJKoZIhvcNAQkBFhFpc2FncUBob3Rt
YWlsLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA4EF8ZaVpkKi1z0kp
V0u5/daPPOJI5/7fdbPRpw1/DhRodpnQH9DVw/HK7RT4gXg687h/75iZYuey/CTB
pl/sSfqlCq5DszlQc6UsVBYuaW0nUT+A9D9dli4ILAT9rSn3BXFkLG58VbSo/CLh
icVdsDj58srYf+4X1naQhKoF7T8CAwEAAaMNMAswCQYDVR0TBAIwADANBgkqhkiG
9w0BAQUFAAOBgQCqlG5vgl9ejgOtJX59+Prq0hDS6b4ucqrlbuEhwVyOABdiPaEB
rCAyV76SSuHfCF9PMwoMMGp1bAIgAt2Q9owZKf0Q4drPomWN7bmJ5seE00ieEJrG
OirFSp962pp38B1F/gmIRSavHU8305lYFQwZzxNYVIMUH0YImoPAnlyXkw==
-----END CERTIFICATE-----
```

รูปที่ 5.4 แสดงใบรับรองสิทธิ์

5.2 ผลการทดสอบการตรวจใบรับรองสิทธิ์

เมื่อออกใบรับรองสิทธิ์ จะมีการกำหนดอายุของใบรับรองสิทธิ์ดังในรูป 5.5 ซึ่งจะเห็นได้ว่ายังไม่หมดอายุ เมื่อทำการตรวจใบรับรองสิทธิ์ จะแสดงข้อความว่า ใบรับรองสิทธิ์ถูกต้องดังรูป 5.6

```
isagmq# openssl x509 -in clientcert3.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: fb:54:de:11:9e:8e:35:26
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=care/emailAddress=pare_g7@hotmail.com
        Validity
            Not Before: Jan 31 14:14:38 2006 GMT
            Not After : Jan 31 14:14:38 2007 GMT
        Subject: ST=BKK, L=SUANLUANG/CCITT=KMITL/CCITT=Isag/mail=pare_g72@hotmail.com
```

รูปที่ 5.5 ใบรับรองสิทธิ์ที่ยังไม่หมดอายุ

```
isagmq# ./verifycert clientcert3.pem
Certificate verified correctly!
```

รูปที่ 5.6 การตรวจใบรับรองสิทธิ์ที่ยังไม่หมดอายุ

ในรูป 5.7 สังเกตว่าส่วนของ Validity ซึ่งใบนี้หมดอายุหลังจาก Jan 24 16.42.14 2006 GMT ซึ่งวันที่ตรวจคือวันที่ Jan 31 2006 แสดงว่าใบรับรองสิทธิ์นี้หมดอายุแล้ว

```
isagmq# openssl x509 -in clientcert2.pem -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            c0:fc:e8:4e:7e:c6:c9:aa
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=care/emailAddress=pare_g7@hotmail.com
        Validity
            Not Before: Dec 25 16:42:14 2005 GMT
            Not After : Jan 24 16:42:14 2006 GMT
        Subject: CN=client2/emailAddress=client2@hotmail.com
```

รูปที่ 5.7 ใบรับรองสิทธิ์ที่หมดอายุแล้ว

เมื่อทำการตรวจใบรับรองสิทธิ์นี้ ระบบจะแจ้งว่าใบรับรองสิทธิ์นี้หมดอายุแล้วดังรูป 5.8

```
isagmq# ./verifycert clientcert2.pem
Error: certificate has expired
** verifycert.c:98 Error verifying the certificate
```

รูปที่ 5.8 การตรวจใบรับรองสิทธิ์ที่หมดอายุแล้ว

บทที่ 6
บทวิจารณ์และสรุป

6.1 บทสรุป

ระบบสามารถทำงานได้ดังนี้

IsagMQ (โปรแกรมฝั่งแม่ข่าย)

1. ออกใบรับรองสิทธิ์ ให้กับ ผู้ใช้
2. จัดทำฐานข้อมูลของผู้ใช้ที่ลงทะเบียนแล้ว
3. สามารถให้บริการลงทะเบียนและยกเลิกการลงทะเบียนได้
4.สามารถเรียกดูใบรับรองสิทธิ์ได้
5.สามารถถอดถอนใบรับรองสิทธิ์ได้

IsagQ (โปรแกรมฝั่งลูกข่าย)

1. สามารถให้บริการลงทะเบียนและยกเลิกการลงทะเบียนได้
2. ผู้ใช้ติดต่อสื่อสารระหว่างกันอย่างปลอดภัยโดยมีการเข้ารหัสข้อมูลสำหรับข้อมูลระหว่างIsagQ ด้วยกัน
3. ผู้ใช้สามารถติดตั้งบนระบบปฏิบัติการ Windows หรือ Linux ได้

6.2 วิจารณ์สิ่งที่ได้จากโครงงาน

1.โครงสร้างโปรแกรมสนทนา

2.โปรโตคอลเอสเอสแอล ซึ่งนำมาใช้เพิ่มความปลอดภัยให้กับระบบ

3.โครงสร้างกุญแจสาธารณะและการออกใบรับรองสิทธิ์

4. การใช้งานไลบรารี่โอเพนเอสเอสแอล

5.โครงสร้างโปรแกรมเกี๊ยม

6. การเขียนจีทีเคพลัส

7. ติดตั้งระบบปฏิบัติการและสภาพแวดล้อมที่จำเป็นต้องใช้ในการพัฒนา

8. การเขียนส่วนขยายของโปรแกรมเกี๊ยม

9. การพัฒนาเซิร์ฟเวอร์ และ สร้างผู้ออกใบรับรองสิทธิ์

10. การสร้างการเชื่อมต่อบนโปรโตคอลเอสเอสแอล และ ติดตั้งการเข้ารหัส และ การพิสูจน์ตัวตน

6.3 ปัญหาอุปสรรคและแนวทางแก้ไข

- ปัญหาการใช้งาน ไลบรารี OpenSSL ก็เป็นอีกหนึ่งในปัญหาหลักที่ทางผู้พัฒนายังขาดความเข้าใจบางส่วน
- ปัญหาในการพัฒนา Gaim plugin ซึ่งไม่ค่อยมีหนังสือหรือเอกสารการพัฒนามีแค่เอกสาร API และต้องศึกษาซอร์สโค้ดจำนวนมากเพื่อที่จะเข้าใจ

6.4 แนวทางการพัฒนาต่อ

- ผู้พัฒนาควรศึกษาทฤษฎี และ ไลบรารี ของ OpenSSL ให้เข้าใจเพื่อให้ง่ายต่อการนำไปใช้และพัฒนาการส่งข้อความให้ปลอดภัยยิ่งขึ้น
- ควรมีการศึกษาและวางแผนการเขียนโปรแกรมให้เกิดความปลอดภัย (Secure Programming)
- ควรจะผนวกความสามารถของการส่งไฟล์แบบปลอดภัยด้วย
- ควรจะสามารถนำใบรับรองสิทธิ์จากผู้ให้บริการใบรับรองสิทธิ์รายอื่นๆมาใช้ได้
- ควรจะมีตัวโปรแกรมช่วยจัดการใบรับรองสิทธิ์บน IsagMQ

บรรณานุกรม

[1] หนังสือ The Definitive Guide to Linux Network Programming (Paperback) ผู้แต่ง Keir Davis, John Turner, Nathan Yocom

[2] ระบบไคลเอ็นต์และเซิรฟเวอร์สำหรับส่งสารด่วนแบบปลอดภัย(Secure Instant Messenger Client /Server

[3] System Network Security with OpenSSL โดย John Viega,Matt Messier, Pravir ,Chandra

[4] Beginning Linux Programming by Neil Matthew,Richard Stones

[5] http://gaim.sourceforge.net/

[6] http://www.mozilla.org/projects/security/pki/nss/

[7] http://www.openssl.org