การใช้งานโปรโตคอล SSL บนอุปกรณ์ Electronics Data Capture สำหรับระบบชำระเงิน

SSL PROTOCOL IMPLEMENTATION ON ELECTRONIC DATA CAPTURE FOR PAYMENT SYSTEM

ยุทธนา ศรลสรรค์

YUTTHNA SROULSRUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

การใช้งานโปรโตคอล SSL บนอุปกรณ์ Electronics Data Capture สำหรับระบบชำระเงิน

SSL PROTOCOL IMPLEMENTATION ON ELECTRONIC DATA CAPTURE FOR PAYMENT SYSTEM

ยุทธนา สรวลสรรค์

YUTTHNA SROULSRUN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2550

SSL PROTOCOL IMPLEMENTATION ON ELECTRONIC DATA CAPTURE FOR PAYMENT SYSTEM

YUTTHNA SROULSRUN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2007

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การใช้งานโปรโตคอล SSL บนอุปกรณ์ Electronics Data Capture สำหรับระบบชำระเงิน

SSL Protocol Implementation on Electronic Data Capture for Payment System

นักศึกษา นายยุทธนา สรวลสรรค์

รหัสประจำตัว 45061213

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.สมศักดิ์ มิตะถา

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ผศ.ดร.ศักดิ์ชัย ทิพย์จักษุรัตน์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ธนา	หงษ์สุวรรณ	
ดร.วรวัฒน์	ลิ้มโภคา	
รศ.ดร.โกสินทร์	จำนงไทย	
ผศ.ดร.ศักดิ์ชัย	ทิพย์จักษุรัตน์	
รศ.สมศักดิ์	มิตะถา	

วัน / เดือน/ ปี ที่สอบ 21 พฤษภาคม 2550 เวลา 11.00-13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.จารุวัตร เจริญสุข)

คณบดีบัณฑิตวิทยาลัย

วันที่ ๒๐ **เดือน** กันยายน **พ.ศ.** ๒๕๕๐

หัวข้อวิทยานิพนธ์	การใช้งานโปรโตคอล SSL บนอุปกรณ์ Electronics Data Capture สำหรับระบบชำระเงิน
นักศึกษา	นายยุทธนา สรวลสรรค์
รหัสประจำตัว	45061213
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2550
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.สมศักดิ์ มิตะถา
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผศ.ดร. ศักดิ์ชัย ทิพย์จักษุรัตน์

บทคัดย่อ

ความปลอดภัยของเครือข่ายจะได้รับความสำคัญในการศึกษาและวิจัยอยู่เสมอ โดยเฉพาะโลกอินเตอร์เน็ตในปัจจุบันก็เห็นว่าเป็นสิ่งที่มีความสำคัญในระดับต้นๆ อีกด้วย หัวข้อการศึกษาและวิจัยนี้จะขาดความสำคัญก็เป็นได้ หากแม้ว่าความปลอดภัยดังกล่าวได้รับการนำไปใช้งานอย่างเต็มประสิทธิภาพในเครือข่ายแล้ว มักจะมีคำถามเสมอว่า ความปลอดภัยนั้นทำงานได้อย่างสมเหตุสมผลแล้วหรือไม่ แม้ว่ามีหลากหลายแนวทางในการจัดการสภาพแวดล้อมของระบบเครือข่ายอย่างปลอดภัย แต่สถาปัตยกรรมความปลอดภัยของ IP (SSL) เป็นกลไกความปลอดภัยของเครือข่ายยอดนิยมและเครือข่ายที่ทันสมัยที่สุด

ในงานวิจัยนี้ เราทำการวิเคราะห์ประสิทธิภาพภายหลังจากที่ได้เพิ่มระบบความปลอดภัย (SSL Protocol) เข้ากับชุดข้อมูลใน TCP/ IP บนอุปกรณ์ EDC (Electronic Data Capture) ต้นทาง ซึ่งสิ่งดำเนินนั้นได้ทำการส่งข้อมูลที่เป็น Plaintext เปรียบเทียบกับ Ciphertext บน SSL ที่แสดงให้เห็นว่าสามารถป้องกันการลักลอบดักข้อมูลได้ เพื่อความเข้าใจถึงผลกระทบที่มีต่อประสิทธิผลของอุปกรณ์ EDC ความปลอดภัยที่ได้ดำเนินการนั้น ได้ทำการตรวจวัดประสิทธิภาพอัตราการส่งผ่านข้อมูลที่มีความปลอดภัยในหลายรูปแบบของการเข้ารหัสข้อมูล

Thesis title	SSL PROTOCOL IMPLEMENTATION ON ELECTRONIC DATA CAPTURE FOR PAYMENT SYSTEM.
Student	Mr. Yutthna Sroulsrun
Student ID.	45061213
Degree	Master of Engineering
Program	Computer Engineering
Year	2007
Thesis Advisor	Ass.Prof. Somsak Mitatha
Thesis Co-advisor	Asst.Prof.Dr. Sakchai Thipchaksurat

ABSTRACT

Network security has always been a significant issue, but a recognized priority today due to the popular of internet. The issue is not if security should be implemented on a network; rather, the question to ask is if security has been implemented properly and the interoperability with today's network architecture. Although there are various ways to perform a secure network environment, but the most popular and the most progressive network security mechanism is Security Architecture for IP (SSL Protocol).

In this research, we have analyzed the performance when combine security system (SSL Protocol) into current TCP/IP module by porting on security shareware into the EDC (Electronic Data Capture). Finally, in order to understand the impact on the EDC's performance, when using various encryption algorithms provided security system, we testing the throughput of the EDC before and after applying security.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดีเนื่องด้วยความรักและความเอื้ออาธรณ์ที่ยิ่งใหญ่จาก คุณแม่พี่ๆทุกคน ข้าพเจ้าขอสำนึกในพระคุณนี้อย่างเป็นที่สุด

วิทยานิพนธ์นี้จะไม่สำเร็จลุล่วงหากปราศจากแรงผลักดัน และคำแนะนำที่มีประโยชน์ของ รศ.สมศักดิ์ มิตะถา ผู้ควบคุมวิทยานิพนธ์ และ ผศ.ดร.ศักดิ์ชัย ทิพย์จักษุรัตน์ ผู้ควบคุมวิทยานิพนธ์ร่วม ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูง

ข้าพเจ้าขอกราบเท้า คุณครูและอาจารย์ทุกท่านตั้งแต่เล็กจนเติบใหญ่ ที่ได้มอบวิชาความรู้ให้แก่ข้าพเจ้า รวมทั้งคำสั่งสอนและอบรมให้ข้าพเจ้าเป็นคนดี ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูง

ข้าพเจ้าขอขอบคุณบริษัท INGENICO THAILAND ที่ได้สนับสนุนเครื่องมือ ตลอดจนข้อมูลต่างๆ ที่ใช้ในการทำวิจัย

สุดท้ายนี้ต้องขอขอบคุณภรรยาและบุตรสาวที่รักของข้าพเจ้า นาง อุมาพร สรวลสรรค์ที่เป็นเสมือนเพื่อนคู่คิด และ ด.ญ.ศศิกานดา สรวลสรรค์ ที่เป็นกำลังใจที่ดีตลอดมา

สำหรับคุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับผู้มีพระคุณทุกท่าน หากวิทยานิพนธ์ฉบับนี้มีข้อผิดพลาดประการใดข้าพเจ้าขอน้อมรับไว้เพียงผู้เดียว

ยุทธนา สรวลสรรค์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในโลกปัจจุบันสิ่งอำนวยความสะดวกอันได้แก่ กระแสไฟฟ้า น้ำประปา รวมไปถึงโทรศัพท์พื้นฐาน เป็นตัวอย่างของสิ่งจำเป็นในการดำเนินชีวิตของมนุษย์ ทั้งนี้ภายหลังจากที่เราได้ใช้บริการของระบบสาธารณูปโภคแล้ว เราก็จำเป็นต้องฝ่าวิกฤตจราจรเพื่อไปชำระค่าบริการดังกล่าว แต่ด้วยภารกิจอันจำเป็นของแต่ละบุคคล ทำให้เหลือเวลาดำเนินการด้านนี้น้อยมาก จึงมีผู้ให้บริการอำนวยความสะดวกด้วยการรับชำระค่าบริการหรือสาธารณูปโภคโดยการหักเงินผ่านบัญชีธนาคาร, ชำระผ่านบัตรเครดิต, ชำระที่เคาน์เตอร์บริการต่างระบบหรือแม้แต่ระบบอิเล็กทรอนิกส์เป็นต้น แต่บางครั้งก็ได้รับความสะดวกยังไม่เพียงพอกับการที่ต้องรอคิวเป็นเวลานานๆ หรือแม้แต่ชำระผ่านตู้ ATM ซึ่งบางครั้งอาจจะไม่มีตู้ ATM ที่เราถือบัตรอยู่ในบริเวณนั้น ก็เกิดเป็นปัญหาในความไม่สะดวกได้

งานวิจัยนี้นำเสนอวิธีการชำระเงินผ่านระบบเครือข่าย โดยใช้อุปกรณ์ EDC (Electronic Data Capture) ที่สามารถรองรับการส่งผ่านข้อมูล โดยการรูดบัตร หรือการสแกนบาร์โค้ด ซึ่งวงจรควบคุมที่อยู่ภายในอุปกรณ์ EDC นั้นเป็นไมโครคอนโทรลเลอร์ และที่นำมาใช้ในงานวิจัยนี้ ใช้ชิปตระกูล ARM7 ซึ่งสามารถทำการโปรแกรม และสามารถแก้ไขโปรแกรมได้ อุปกรณ์ EDC ที่ใช้นั้นสามารถติดต่อสื่อสารกับระบบเครือข่ายด้วย พอร์ต Modem, RS-232 และ Ethernet ในการวิจัยครั้งนี้ได้ทดลองกับพอร์ต Ethernet ติดต่อระบบเครือข่าย ซึ่งในการต่อเข้าระบบเครือข่ายนั้น เรื่องสำคัญอีกประการคือเรื่องความปลอดภัยของข้อมูล ในการรับ-ส่งข้อมูลต้องคำนึงถึงความปลอดภัย ดังนั้นจึงจำเป็นต้องมีการเข้ารหัสข้อมูลเพื่อรักษาความปลอดภัย โดยงานวิจัยนี้ได้ดำเนินการศึกษาและเพิ่มขีดความสามารถในด้านความปลอดภัยให้กับตัวอุปกรณ์ EDC ด้วยโปรโตคอลความปลอดภัย SSL และสุดท้ายได้ทำการตรวจวัดประสิทธิภาพของอุปกรณ์ EDC ในลักษณะการเข้ารหัสข้อมูลแบบต่างๆ เปรียบเทียบกันเพื่อหาความเหมาะสมที่จะนำไปใช้ประโยชน์ต่อไป

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาและเพิ่มขีดความสามารถให้กับตัวอุปกรณ์ EDC ด้วย SSL Protocol สำหรับระบบชำระเงิน
2. เพื่อให้มีความเหมาะสมด้านความรวดเร็วในการสื่อสารข้อมูล

3. เพื่อรักษาความปลอดภัยของแพ็กเก็ตข้อมูล
4. เพื่อสร้างความเชื่อมั่นในการชำระเงิน
5. เพื่อนำไปใช้งานกับระบบเครือข่ายที่แท้จริงได้

1.3 สมมุติฐานของการศึกษา

การเพิ่มขบวนการพิสูจน์ตัวตนและการเข้ารหัส เพื่อความปลอดภัยของข้อมูลกับอุปกรณ์ EDC (Electronic Data Capture) เพื่อใช้ในการชำระเงินผ่านระบบเครือข่ายสามารถใช้งานได้

1.4 แนวความคิดที่ใช้ในการวิจัย

การทำวิจัยนี้จะนำเอาทฤษฎีทางด้านความปลอดภัยบนระบบเครือข่ายมาใช้สนับสนุน และโปรโตคอลที่ใช้ในการวิจัยคือ SSL Protocol ที่สามารถสร้างการเชื่อมต่อเครือข่ายจากเทอร์มินัลที่เป็นอุปกรณ์ EDC (Electronic Data Capture) ไปยังเซิร์ฟเวอร์ปลายทาง ด้วยพอร์ตการสื่อสารแบบ Ethernet ซึ่งจะต้องเชื่อมต่อเข้าสู่ระบบเครือข่ายที่ใช้งาน Internet จริง ดังนั้นจึงมีแนวความคิดที่จะเพิ่มขีดความสามารถของอุปกรณ์ EDC ด้วยการเพิ่มโปรโตคอลรักษาความปลอดภัย SSL Protocol เพื่อเป็นการรักษาความลับของข้อมูลในการติดต่อสื่อสารบนเครือข่าย Internet และสุดท้ายได้ทำการตรวจวัดประสิทธิภาพของอุปกรณ์ EDC ในลักษณะการเข้ารหัสข้อมูลแบบต่างๆ เปรียบเทียบกันเพื่อหาความเหมาะสมที่จะนำไปใช้ประโยชน์ต่อไป

1.5 ขอบเขตการวิจัย

1. ทดสอบการรับ-ส่งข้อมูลที่เป็น Plain text ที่ยังไม่มีระบบความปลอดภัย
2. ตรวจวัดคุณลักษณะของแพ็กเก็ตข้อมูล
3. ศึกษาและใช้งานอัลกอริธึมของการเข้ารหัส (Encryption) หลายรูปแบบ
4. สร้างระบบความปลอดภัยสำหรับอุปกรณ์ EDC ด้วย SSL Protocol
5. ทดลองระบบทั้งหมด
6. วิเคราะห์และสรุปผลระบบความปลอดภัยเครือข่ายสำหรับระบบการชำระเงิน

1.6 ขั้นตอนการศึกษา

1. ศึกษา SSL Protocol
2. ศึกษา Stunnel
3. ศึกษาและทดลองใช้โปรแกรมตรวจวัดคุณลักษณะแพ็กเก็ตข้อมูล (Ethereal)
4. ศึกษาคุณสมบัติและทดลองใช้งานอุปกรณ์ EDC
5. ทดลองส่งข้อมูลผ่านอุปกรณ์ EDC และวัดคุณลักษณะแพ็กเก็ตข้อมูล
6. ศึกษาและใช้งานอัลกอริธึมการเข้ารหัสหลายรูปแบบ
7. เขียนโปรแกรมที่จะใช้กับอุปกรณ์ EDC
8. ทดลองส่งข้อมูลที่มีโปรโตคอลความปลอดภัย SSL ผ่าน EDC ทำการตรวจวัดคุณลักษณะแพ็กเก็ตข้อมูล และทำการตรวจวัดอัตราการส่งผ่านข้อมูลที่มีความปลอดภัย
9. สรุปผลการทดลอง

1.7 เนื้อหาของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บทคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมุติฐาน แนวความคิด ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้อง ภาพรวมของระบบเครือข่าย Protocol มาตรฐานการรับ-ส่งสัญญาณในเครือข่าย การพิสูจน์ตัวตน ความปลอดภัยของข้อมูล

บทที่ 3 การออกแบบระบบ SSL Protocol บนอุปกรณ์ Electronic Data Capture

บทที่ 4 เป็นผลการทดลอง

บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

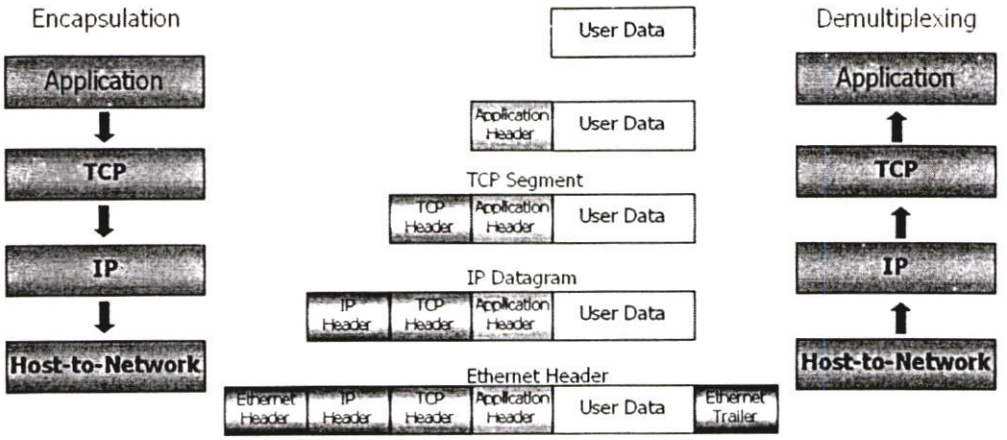
2.1 ความรู้พื้นฐานเกี่ยวกับโปรโตคอล TCP/IP

การใช้งานระบบการสื่อสารข้อมูลในปัจจุบันนั้น TCP/IP (Transmission Control Protocol/ Internet Protocol) [1] ,[2] เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหาโปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

จุดประสงค์ของการสื่อสาร TCP/IP ตามมาตรฐาน คือ เพื่อใช้สื่อสารระหว่างระบบที่มีความแตกต่างกัน, สามารถแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ, มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

2.1.1 Encapsulation/Demultiplexing

การส่งข้อมูลผ่านในแต่ละเลเยอร์ (Layer) ซึ่งแต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือ โปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



รูปที่ 2.1 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data เมื่อแอพพลิเคชั่นได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่น เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP

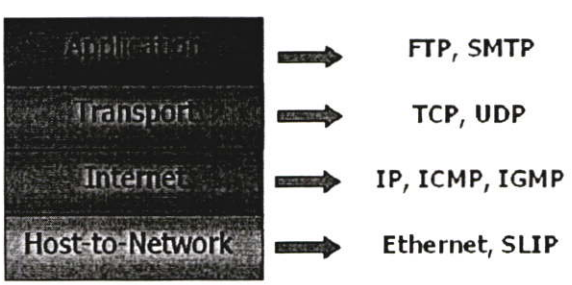
เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP

เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer

ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณต่อไป

2.1.2 การทำงานของชั้นเลเยอร์

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่2.2 โครงสร้าง TCP/IP

2.1.2.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้ เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

2.1.2.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้น แพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้ในที่นี้จะขออธิบายเฉพาะ IP Protocol

- **IP (Internet Protocol)**

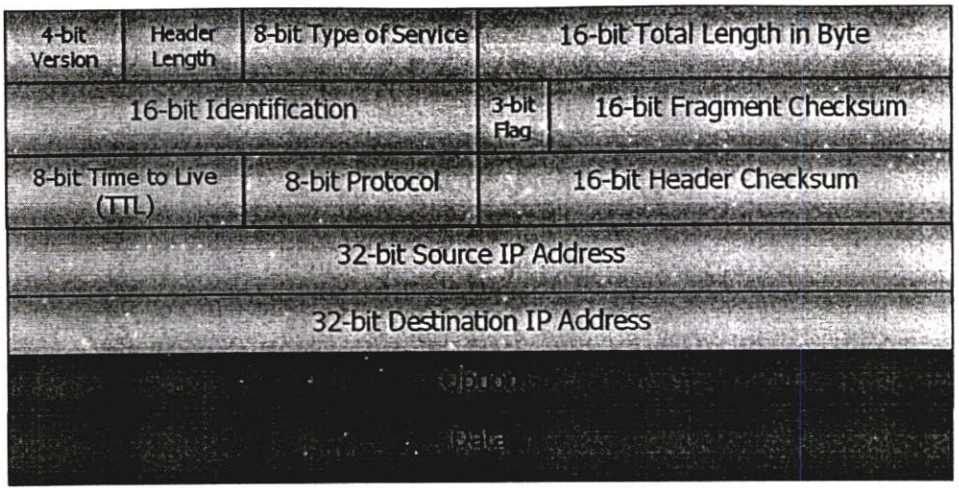
IP เป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูลและควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ต ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบดาต้าแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับ data link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet, Token Ring หรือ Apple Talk

การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูล จะเป็นแบบ connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆ ครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ(fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

Version : หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)

Header Length : ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีส่วน option จะมีค่าเป็น 5 (5*32 bit)



รูปที่ 2.3 IP Header

Type of Service (TOS) : ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกการเราต์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

Length : ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟิลด์จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535 byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอพพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte

Identification : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรม เมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน

Flag : ใช้ในกรณีที่มีการแยกดาต้าแกรม

Fragment offset : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

Time to live (TTL) : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

Protocol : ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP ,UDP หรือ ICMP

Header checksum : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์

Source IP address : หมายเลข IP ของผู้ส่งข้อมูล

Destination IP address : หมายเลข IP ของผู้รับข้อมูล

Data : ข้อมูลจากโปรโตคอลระดับบน

2.1.2.3 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

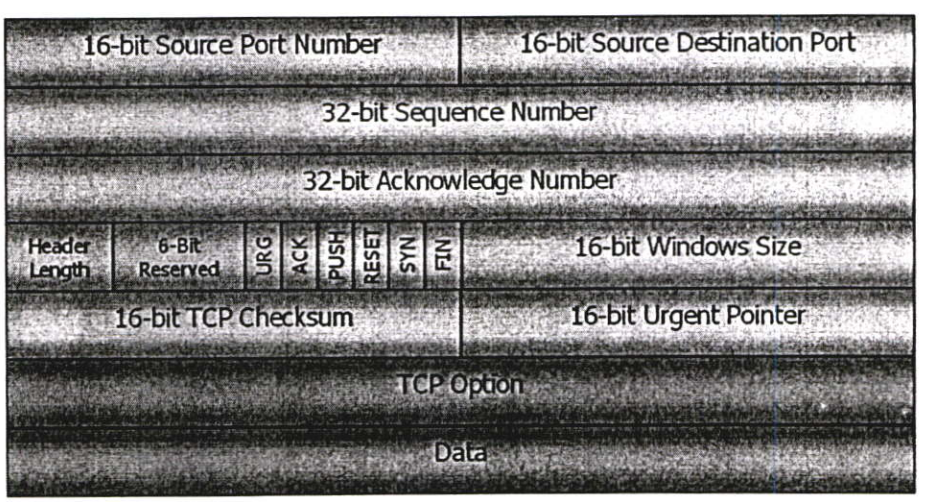
แบ่งเป็นโปรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client / server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพ เคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

ซึ่งจะขออธิบายเฉพาะ TCP Protocol

- **TCP : (Transmission Control Protocol)**

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยคาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (connection-oriented) มีรายละเอียด ดังนี้



รูปที่ 2.4 TCP Header

Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง

Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ

Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์

Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ตามตารางที่ 2.1ดังนี้

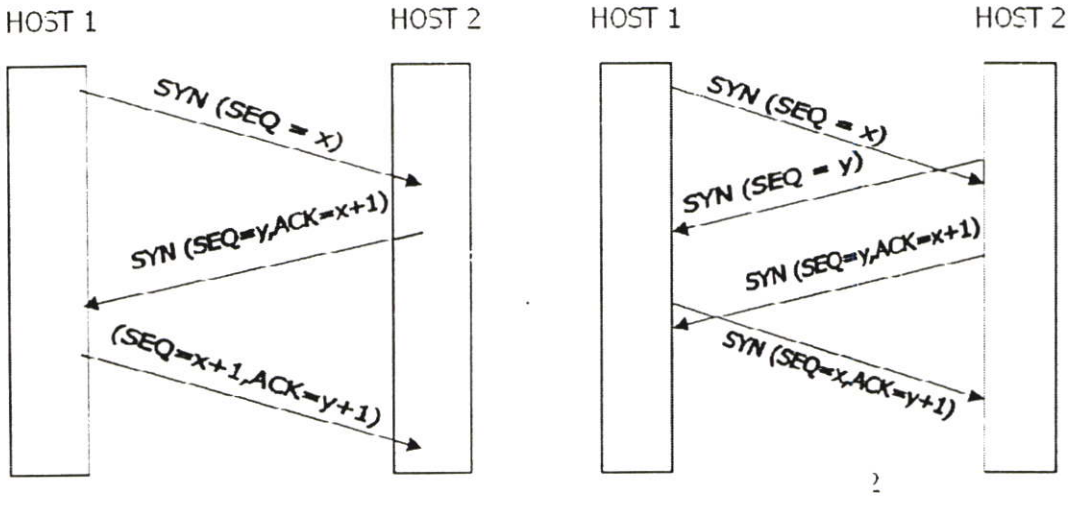
ตารางที่ 2.1 รายละเอียดของ flag

Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

2.1.2.3.1 การสื่อสารของ TCP

เมื่อเซกเมนต์ CONNECT (SYN = "1" และ ACK = "0") เดินทางมาถึง Entity TCP ที่โฮสต์ปลายทางจะค้นหาโพรเซสตามหมายเลขพอร์ตที่กำหนดในเขตข้อมูล Destination port ซึ่งถ้าหากไม่พบก็จะตอบปฏิเสธด้วยเซกเมนต์ที่มี RST = "1" กลับไปยังผู้ส่ง



รูปที่ 2.5 การสื่อสารของ TCP

เซกเมนต์ CONNECT ของผู้ส่งจะถูกส่งต่อไปยังโพรเซส ตามพอร์ตที่ระบุซึ่งอาจจะตอบรับหรือตอบปฏิเสธก็ได้ ถ้าโพรเซสนั้นต้องการสื่อสารด้วยก็จะส่งเซกเมนต์ตอบรับกลับไป รูปที่ 2.5 แสดงลำดับขั้นตอนการส่ง TCP เซกเมนต์ในการสร้างการเชื่อมต่อในสภาวะปกติระหว่างผู้ส่งและผู้รับ

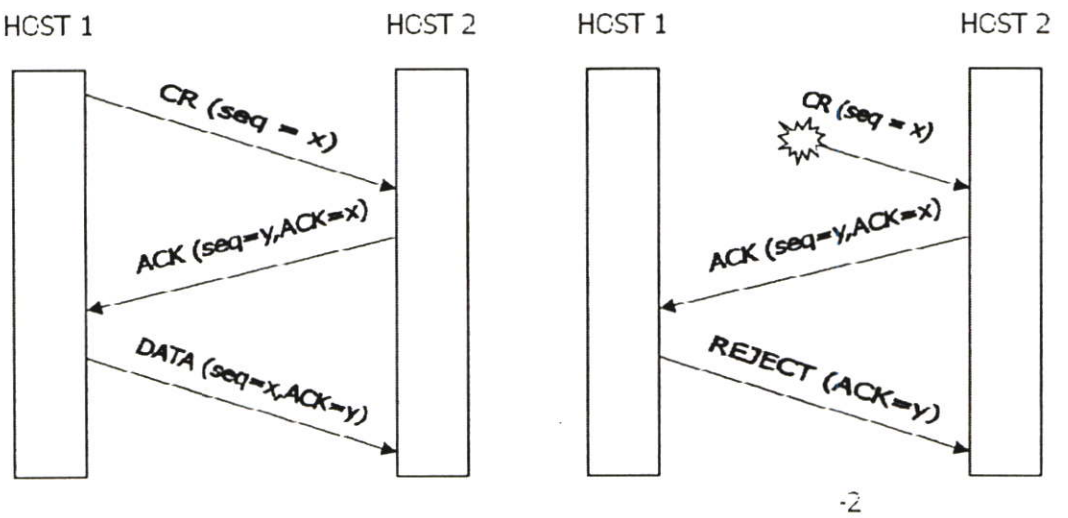
ในกรณีที่โฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อคเก็ตคู่เดียวกัน จะเกิดเป็นลำดับขั้นตอนแสดงในรูปที่ 2.5 ผลสุดท้ายจะมีการเชื่อมต่อเกิดขึ้นเพียงหนึ่งช่องทางเท่านั้น เนื่องจากการเชื่อมต่อในแต่ละช่องทางจะถูกกำหนดขึ้นโดยใช้หมายเลขซ็อคเก็ตผู้ส่งและผู้รับ ถ้าการเชื่อมต่อลำดับแรกสำเร็จก็จะถูกบันทึกไว้ในตารางการสื่อสาร เช่น (x, y) ถ้าการเชื่อมต่อลำดับที่สองสำเร็จในเวลาต่อมา ข้อมูลนี้ก็จะถูกบันทึกไว้ที่เดียวกันคือ (x, y)

การเชื่อมต่อเริ่มต้นจากสถานะ CLOSED เมื่อเรียกใช้บริการ LISTEN หรือ CONNECT ก็จะมีการเปลี่ยนสถานะไปจากเดิม และถ้าอีกฝ่ายต้องการเชื่อมต่อด้วย การเชื่อมต่อก็จะเกิดขึ้นและย้ายไปอยู่ในสถานะ ESTABLISHED คือการเชื่อมต่อสมบูรณ์ และเมื่อยกเลิกการติดต่อก็จะกลับไปสู่สถานะ CLOSED อย่างเดิม

2.1.2.3.2 การเริ่มต้นการสื่อสารของ TCP โดยใช้การบันทึกเวลาแบบ Three-way handshake

Three-way Handshake เป็นวิธีการส่งแพ็กเก็ตที่สามารถช่วยแก้ปัญหาในเรื่องแพ็กเก็ตซ้ำซ้อนได้ดี แต่วิธีนี้จำเป็นจะต้องสร้างช่องสื่อสารให้ได้ก่อนที่จะเริ่มรับ-ส่งข้อมูล อย่างไรก็ตาม แพ็กเก็ตควบคุมที่ใช้ในการต่อรองค่าตัวแปรสำหรับการสื่อสารต่างๆ อาจเกิดการตกค้างอยู่ในระบบได้ ทำให้การกำหนดค่าหมายเลขลำดับมีปัญหาไปด้วย เช่นการสร้างช่องสื่อสารระหว่างโฮสต์1 และ โฮสต์2 เริ่มจาก โฮสต์1 ขอเริ่มการเชื่อมต่อด้วยการส่งแพ็กเก็ต CR (Connection Request) ไปยังโฮสต์2 ซึ่งจะมีค่าตัวแปรต่างๆสำหรับการสื่อสารรวมทั้งหมายเลขลำดับและหมายเลขช่องสื่อสารไปด้วย ผู้รับคือโฮสต์2 ก็จะส่ง ACK (Acknowledge) กลับมายังโฮสต์1 แต่ถ้าแพ็กเก็ต จากผู้ส่งเกิดสูญหายระหว่างทาง และสำเนาแพ็กเก็ตที่ยังตกค้างอยู่ระบบเกิดเดินทางไปถึงผู้รับในภายหลังก็จะทำให้การสร้างช่องสื่อสารใช้การไม่ได้เนื่องจากมีค่าตัวแปรต่างๆไม่ตรงกัน

การใช้ Three-way handshake เป็นการไม่บังคับให้ผู้ส่งและผู้รับข้อมูลจะต้องกำหนดค่าเริ่มต้นของหมายเลขลำดับเป็นเลขเดียวกัน ทำให้สามารถนำวิธีนี้มาใช้ร่วมกับวิธีการจัดจังหวะการทำงานให้พร้อมกัน (Synchronization) แบบต่างๆได้ แทนที่จะเป็นการใช้วิธีการบันทึกเวลา ดังรูปที่ 2.6 แสดงขั้นตอนการเริ่มต้นการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2 สมมุติให้โฮสต์ 1 เลือกหมายเลขลำดับเป็น “x” และส่งแพ็กเก็ต CONNECTION REQUEST ไปยังโฮสต์ 2 โฮสต์ 2 ตอบรับด้วยแพ็กเก็ต CONNECTION ACCEPTED ซึ่งจะยอมรับหมายเลขลำดับ “x” พร้อมกับประกาศหมายเลขลำดับ “y” ที่เป็นของตนเอง จากนั้นโฮสต์ 1 ก็จะตอบรับค่าตัวเลือกของโฮสต์ 2 ผ่านทางเขตข้อมูลสำหรับการควบคุมในแพ็กเก็ตข้อมูลแรกที่ส่งมา



รูปที่ 2.6 การสื่อสารของ TCP แบบ Three-way handshake

สมมติว่า ได้เกิดปัญหาการสูญหายของแพ็กเก็ตในขณะที่สำเนาแพ็กเก็ตที่ค้างในระบบเดินทางไปถึงผู้รับแทน รูปที่ 2.6 แสดงเหตุการณ์ที่แพ็กเก็ต TPDU (ตัวแรกในรูป) เป็นสำเนาแพ็กเก็ตเก่าที่พึ่งจะเดินทางไปถึงโฮสต์ 2 โดยที่โฮสต์ 1 ไม่ทราบ โฮสต์ 2 ก็จะทำงานตามปกติคือจะตอบรับด้วยการส่งแพ็กเก็ต CONNECTION ACCEPTED TPDU กลับมา ที่โฮสต์ 1 ซึ่งโฮสต์1 จะสามารถตรวจสอบได้ว่า หมายเลขลำดับโฮสต์2 ตอบกลับมานั้นเป็นหมายเลขลำดับที่ได้เลิกใช้ไปแล้ว จึงมีการส่งแพ็กเก็ต REJECTกลับมายังโฮสต์ 2 เพื่อบอกยกเลิกการทำงาน จะเห็นว่าวิธีการนี้อาศัยการสื่อสารผ่านแพ็กเก็ต 3 ตัวซึ่งเป็นที่มาของคำว่า "การจับมือร่วมสามขั้นตอน" ผลสุดท้ายทั้งโฮสต์ 1 และโฮสต์ 2 ก็จะไม่มีการสร้างช่องสื่อสารขึ้นมาจากข้อมูลในสำเนาแพ็กเก็ตเก่าแต่อย่างใด

2.1.2.4 ชั้นสื่อสารการประยุกต์ (Application Layer)

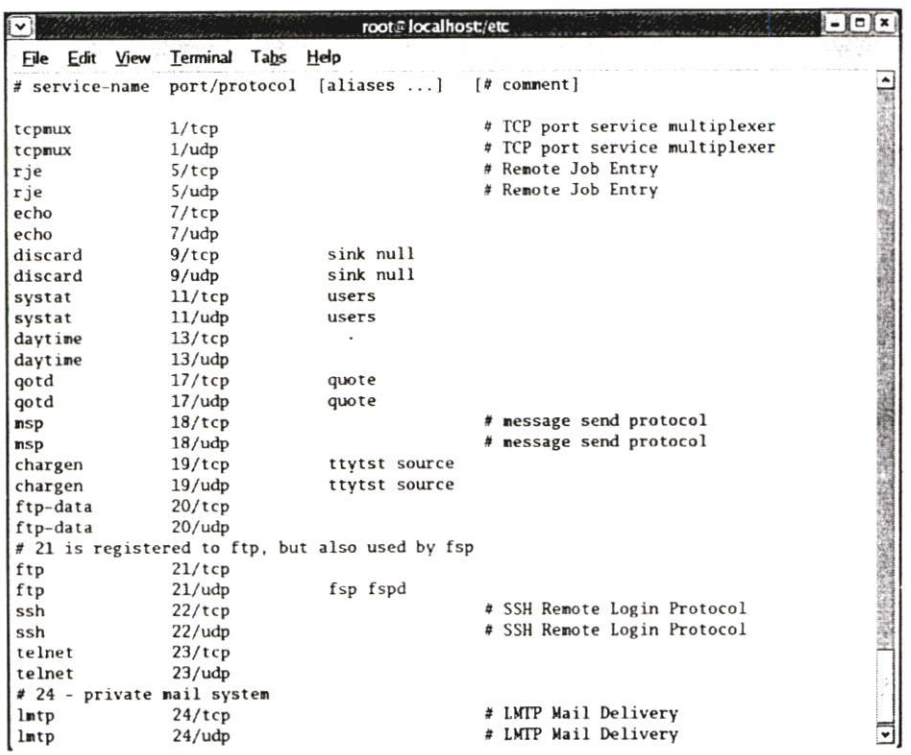
มีโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โปรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP และโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP โดยโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โปรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้ โปรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

2.2 คำอธิบายเกี่ยวกับเรื่อง พอร์ต

สำหรับ Application ในชั้นเลเยอร์สูงๆ ที่ใช้ TCP (Transmission Control Protocol) หรือ UDP (User Datagram Protocol) จะมีหมายเลข พอร์ต [3] หมายเลขของ พอร์ต จะเป็นเลข 16 บิต เริ่มตั้งแต่ 0 ถึง 65535 หมายเลข พอร์ต ใช้สำหรับตัดสินว่า service ใดที่ต้องการเรียกใช้ ในทางทฤษฎี หมายเลข พอร์ต แต่ละหมายเลขถูกเลือกสำหรับ service ใดๆ ขึ้นอยู่กับ OS (operating system) ที่ใช้ ไม่จำเป็นต้องเหมือนกัน แต่ได้มีกำหนดขึ้นให้ใช้ค่อนข้างเป็นมาตรฐานเพื่อให้มีการติดต่อการส่งข้อมูลที่ดีขึ้น ทาง Internet Assigned Numbers Authority (IANA) เป็นหน่วยงานกลางในการประสานการเลือกใช้ พอร์ต ว่า พอร์ต หมายเลขใดควรเหมาะสำหรับ Service ใด และได้กำหนดใน Request For Comments (RFC') 1700 ตัวอย่างเช่น เลือกใช้ TCP Port หมายเลข 23 กับ Service Telnet และเลือกใช้ UDP Port หมายเลข 69 สำหรับ Service Trivial File transfer Protocol (TFTP) ตัวอย่างต่อไปนี้เป็นบางส่วนของ File/etc/services แสดงให้เห็นว่า หมายเลข พอร์ต แต่ละ

หมายเลขได้ถูกจับคู่กับ Transport Protocol หนึ่งหรือสอง Protocol ซึ่งหมายความว่า UDP หรือ TCP อาจจะใช้ หมายเลข พอร์ต เดียวกันก็ได้ เนื่องจากเป็น Protocol ที่ต่างกัน

หมายเลข พอร์ต ถูกจัดแบ่งเป็น 2 ประเภท ตามที่ได้กำหนดใน RFC' 1700 (รายละเอียด Download และศึกษาได้ที่ ftp://ftp.isi.edn/in-notes/rfc'1700.txt) คือ well known Ports และ Registered Ports



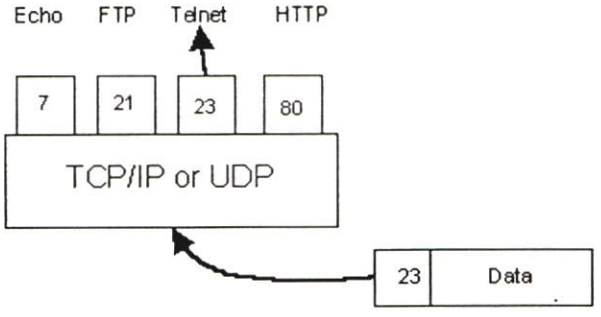
รูปที่ 2.7 แสดงหมายเลขพอร์ต

- Well Known Ports คือจะเป็น พอร์ต ที่ระบบส่วนใหญ่ กำหนดให้ใช้โดย Privileged User (ผู้ใช้ที่มีสิทธิพิเศษ) โดย พอร์ต เหล่านี้ ใช้สำหรับการติดต่อระหว่างเครื่องที่มีระบบเวลาที่ยาวนาน วัตถุประสงค์เพื่อให้ service แก่ผู้ใช้ (ที่ไม่รู้จักหรือคุ้นเคย) แปลกหน้า จึงจำเป็นต้องกำหนด พอร์ต ติดต่อสำหรับ Service นั้นๆ
- Registered Ports จะเป็น พอร์ต หมายเลข 1024 ขึ้นไป ซึ่ง IANA ไม่ได้กำหนดไว้

2.2.1 การใช้ พอร์ต

แต่ละ Transport layer segment จะมีส่วนย่อยที่ประกอบไปด้วยหมายเลข พอร์ต ของเครื่องปลายทาง โดยที่เครื่องปลายทาง (Destination hostt) จะใช้ พอร์ต นี้ในการส่งข้อมูลให้ไหลกับ Application ได้ถูกต้อง หน้าที่ในการส่งหรือแจกจ่าย Segment ของข้อมูลให้ตรงกับ Application เรียกว่าการ "Demultiplexing" ในทางกลับกันเครื่องต้นทาง (Source host) หน้าที่ในการรวบรวม

ข้อมูลจาก Application และเพิ่ม header เพื่อสร้าง segment เรียกว่า "Multiplexing" หรือถ้ายกตัวอย่างเป็นภาษาทั่วๆ ไป คือ ในแต่ละบ้านจะมีคน 1 คนรับผิดชอบเก็บจดหมายจากกล่องจดหมาย ถ้าเป็นการ Demultiplexing คนๆ นั้นจะแจกจ่ายจดหมายที่จ่าหน้าซองให้สอดคล้องกับบุคคลนั้นๆ ในบ้าน ในทางตรงกันข้าม ถ้าเป็นการ Multiplexing คนๆ นั้นก็จะรวบรวมจดหมายจากสมาชิกในบ้านและทำหน้าที่ส่งออกไป Demultiplexing ตามรูปที่ 2.8



รูปที่ 2.8 การใช้งานของหมายเลขพอร์ต

หมายเลข พอร์ต จะอยู่ใน 32 บิต แรกของ TCP และ UDP header โดยที่ 16 บิต แรกเป็นหมายเลข พอร์ต ของเครื่องต้นทาง ขณะที่ 16 บิต ต่อมาเป็นหมายเลข พอร์ต ของ เครื่องปลายทาง ดังแสดงในรูปที่ 2.9

<table>
<tr><td rowspan="2">Ethernet header</td><td rowspan="2">IP header</td><td>Source Port</td><td>Destination Port</td><td rowspan="2">Application data</td><td rowspan="2">Ethernet trailer</td></tr>
<tr><td colspan="2">TCP/UDP header</td></tr>
</table>

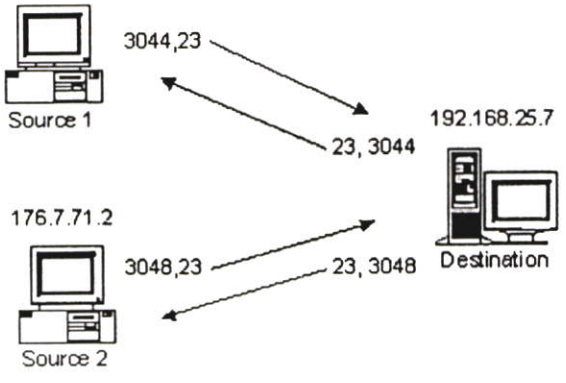
รูปที่ 2.9 เฮดเดอร์ของพอร์ต

TCP หรือ UDP จะดูที่ข้อมูลหมายเลข พอร์ต ใน header เพื่อพิจารณาว่า Application ใดที่ต้องการข้อมูลนั้นๆ หมายเลข พอร์ต ทั้งต้นทางและปลายทางจำเป็นต้องมีเพื่อให้ เครื่องปลายทางมีความสามารถที่จะรัน process มากกว่า 1 process ในเวลาเดียวกัน

ตามที่ได้กล่าวในข้างต้น "Well know Ports" เป็น พอร์ต ที่ค่อนข้างมาตรฐาน ทำให้เครื่องที่อยู่ไกลออกไป (Remote Computer) สามารถรู้ได้ว่าจะติดต่อกับทาง พอร์ต หมายเลขอะไรสำหรับ Service เฉพาะนั้นๆ อย่างไรก็ตามยังมี พอร์ต อีกประเภทที่เรียกว่า Dynamically Allocated Port ซึ่ง พอร์ต ประเภทนี้ไม่ได้ถูก assign ไว้แต่เดิม แต่จะถูก assign เมื่อจำเป็น พอร์ต ประเภทนี้ให้ความสะดวกและความคล่องตัวสำหรับระบบที่มีผู้ใช้หลายๆคนพร้อมๆคน ระบบจะต้องให้ความมั่นใจว่าจะไม่ assign หมายเลข พอร์ต ซ้ำกัน

ยกตัวอย่าง สมมติว่ามีผู้ใช้ต้องการใช้ Service Telnet ทางเครื่องต้นทางจะทำการ assign ให้หมายเลข Dynamic Port (เช่น 3044) โดยที่หมายเลข พอร์ต ปลายทางคือ 23 เครื่องจะ assign หมายเลข พอร์ต ปลายทางเป็น23 เพราะว่า เป็น Well Known Port สำหรับ Service Telnet จากนั้นเครื่องปลายทางจะทำการตอบรับกลับโดยใช้ พอร์ต หมายเลข 23 เป็นหมายเลขต้นทาง และหมายเลข พอร์ต 3044 เป็นหมายเลข ปลายทาง

กลุ่มของหมายเลข พอร์ต และ หมายเลข IP เราเรียกว่า Socket ซึ่งจะเป็นตัวบ่งชี้ที่เฉพาะเจาะจงสำหรับ Network process หนึ่งเดียวที่มีอยู่ในทั้งระบบ Internet คู่ของ Socket ที่ประกอบด้วย Socket หนึ่งตัว สำหรับต้นทาง และอีกตัว สำหรับปลายทาง สามารถใช้บรรยายถึงคุณลักษณะของ Connection oriented protocols เช่น



รูปที่ 2.10 การใช้งานพอร์ต

ถ้าผู้ใช้คนที่ 2 ต้องการใช้ Service Telnet จากเครื่องปลายทางเครื่องเดียวกัน ผู้ใช้นั้นก็จะได้รับการ assign หมายเลข พอร์ต ต้นทางที่แตกต่างกันออกไป โดยมีหมายเลข พอร์ต ปลายทางเหมือนกันกับผู้ใช้คนแรกดังรูปที่ 2.10 จะเห็นได้ว่าการจับคู่ของหมายเลข พอร์ต และหมายเลข IP ทั้งต้นทางและปลายทางสามารถทำให้แยกความแตกต่างของ Internet connection ระหว่างเครื่องต้นทางและเครื่องปลายทางได้

2.2.2 Active และ Passive Ports

สิ่งสุดท้ายที่จะต้องกล่าวถึงเกี่ยวกับ พอร์ต ก็คือ ความแตกต่างระหว่าง Active และ Passive Portในการใช้การติดต่อด้วย TCP สามารถกระทำได้ 2 วิธีคือ Passive และ Active Connection Passive connection คือ การติดต่อที่ Application process สั่งให้ TCP รอหมายเลข พอร์ต สำหรับการร้องขอการติดต่อจาก Source Host เมื่อ TCP ได้รับการร้องขอแล้วจึงทำการเลือกหมายเลข พอร์ต ให้ แต่ถ้าเป็นแบบ Active TCP ก็จะให้ Application process เป็นฝ่ายเลือกหมายเลข พอร์ต ให้เลย

2.3 การพิสูจน์ตัวตน

การดำเนินกิจกรรมทางด้านคอมพิวเตอร์บนระบบเครือข่ายนั้น ไม่ว่าจะเป็นระบบการเงินการธนาคาร ระบบฐานข้อมูล ระบบชำระเงินและอื่นๆ อีกมากมาย ในปัจจุบันจำเป็นต้องคำนึงถึงการปกป้องความมั่นคงปลอดภัยของระบบและข้อมูลถือเป็นเรื่องสำคัญ ทั้งนี้เนื่องจากการถูกคุกคามโดยผู้ไม่ประสงค์ดีหรือจากโปรแกรมบางประเภทได้เพิ่มมากขึ้น และอาจนำมาซึ่งความเสียหายอย่างมากต่อระบบและองค์กร ดังนั้นถ้าภายในระบบมีการดูและระบบควบคุมความปลอดภัยที่ดีจะช่วยลดโอกาสเสี่ยงต่อการถูกคุกคามได้ [4]

การพิสูจน์ตัวตนซึ่งเป็นขั้นตอนพื้นฐานที่สำคัญของการควบคุมความปลอดภัย ในกระบวนการการพิสูจน์ตัวตน จะนำหลักฐานที่ผู้ใช้กล่าวอ้างมาตรวจสอบว่าบุคคลที่กล่าวอ้างนั้นเป็นใครและได้รับอนุญาตให้สามารถเข้ามาภายในระบบได้หรือไม่ การพิสูจน์ตัวตนมีหลายประเภทที่ใช้อยู่ในปัจจุบัน เช่น การพิสูจน์ตัวตนโดยใช้รหัสผ่าน ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล หรือโดยใช้รหัสผ่านที่ใช้เพียงครั้งเดียว เป็นต้น แต่ละชนิดนั้นจะมีข้อดีข้อเสียแตกต่างกันไปขึ้นอยู่กับความจำเป็นในการใช้งาน ในระบบเครือข่ายแบบเปิดหรืออินเตอร์เน็ตนั้นการพิสูจน์ตัวตนถือได้ว่าเป็นกระบวนการเริ่มต้นและมีความสำคัญที่สุดในการปกป้องเครือข่ายให้ปลอดภัย

2.3.1 ความมั่นคงปลอดภัยคอมพิวเตอร์

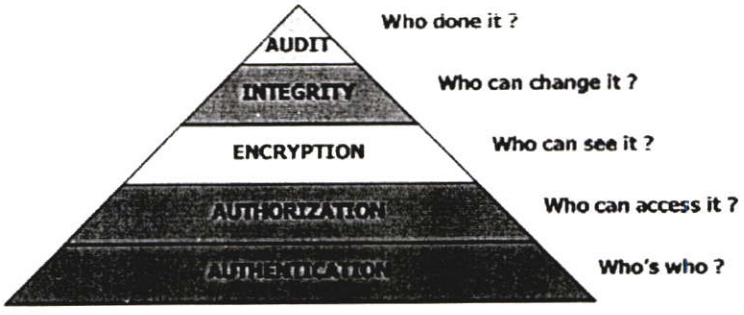
ระบบคอมพิวเตอร์ได้ถูกคุกคามมากขึ้น ทั้งจากผู้ไม่ประสงค์ดีหรือจากไวรัสคอมพิวเตอร์ ซึ่งความมั่นคงปลอดภัยคอมพิวเตอร์ (Computer Security) ช่วยปกป้องระบบคอมพิวเตอร์รวมไปถึงอุปกรณ์ต่างๆที่เกี่ยวข้อง และที่สำคัญยังสามารถช่วยปกป้องข้อมูลที่ได้จัดเก็บไว้ภายในระบบหรือใช้ในความหมายความปลอดภัยทางข้อมูลสารสนเทศ (Information Security) ก็ได้ จุดประสงค์หลักของความปลอดภัยทางข้อมูลคือ ความลับ (Confidentiality) ความสมบูรณ์ (Integrity) ความพร้อมใช้ (Availability) และการห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation) ของข้อมูลต่างๆภายในองค์กร (CIA-N) โดยมีรายละเอียดดังนี้

- **การรักษาความลับ (Confidentiality)** คือการรับรองว่าจะมีการป้องกันข้อมูลไว้เป็นความลับและผู้มีสิทธิเท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้
- **การรักษาความสมบูรณ์ (Integrity)** คือการรับรองว่าข้อมูลจะไม่มีการเปลี่ยนแปลงหรือทำลายไม่ว่าจะเป็นโดย อุบัติเหตุหรือโดยเจตนา
- **ความพร้อมใช้งาน (Availability)** คือการรับรองว่าข้อมูลและบริการการสื่อสารต่าง ๆ พร้อมที่จะใช้ได้ในเวลาที่ต้องการใช้งาน

- **การห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation)** คือวิธีการสื่อสารซึ่งผู้ส่งข้อมูลได้รับหลักฐานว่าได้มีการส่งข้อมูลแล้วและผู้รับก็ได้รับการยืนยันว่าผู้ส่งเป็นใคร ดังนั้นทั้งผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่มีความเกี่ยวข้องกับข้อมูลดังกล่าวในภายหลัง

2.3.2 การควบคุมความมั่นคงปลอดภัย

ในทางปฏิบัติสามารถกำหนดลักษณะของการควบคุมความมั่นคงปลอดภัย (Security Controls) ได้ 5 ระดับตามรูปที่ 2.11

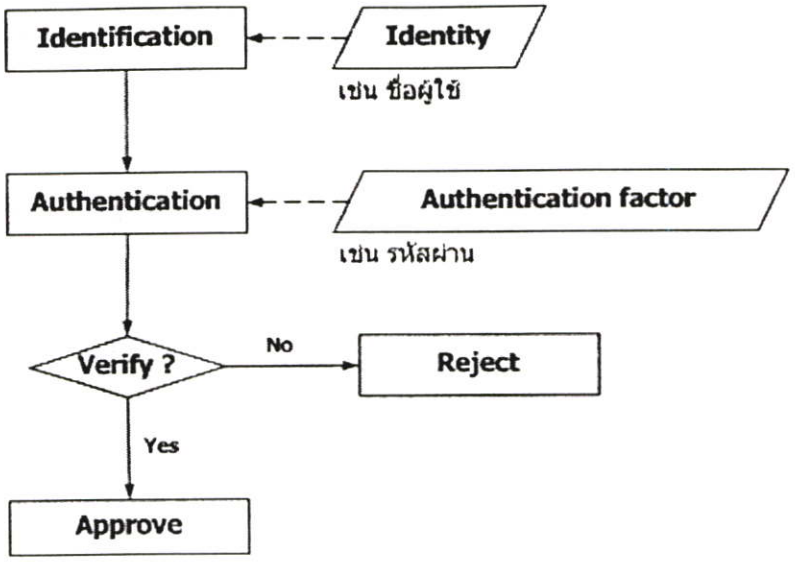


รูปที่ 2.11 แสดง Security Pyramid

2.3.2.1 การพิสูจน์ตัวตน (Authentication)

การพิสูจน์ตัวตน คือขั้นตอนการยืนยันความถูกต้องของหลักฐาน (Identity) ที่แสดงว่าเป็นบุคคลที่กล่าวอ้างจริง ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

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



รูปที่ 2.12 แผนผังแสดงกระบวนการการพิสูจน์ตัวตน

จากแผนผังแสดงกระบวนการพิสูจน์ตัวตน ในขั้นแรกผู้ใช้จะทำการแสดงหลักฐานที่ใช้ในการพิสูจน์ตัวตนต่อระบบ ซึ่งในขั้นนี้คือการระบุตัวตน และในขั้นตอนต่อมาระบบจะทำการตรวจสอบหลักฐานที่ผู้ใช้นำมากล่าวอ้างซึ่งก็คือการพิสูจน์ตัวตน หลังจากระบบได้ทำการตรวจสอบหลักฐานเรียบร้อยแล้วถ้าหลักฐานที่นำมากล่าวอ้างถูกต้องจึงอนุญาตให้เข้าสู่ระบบได้ หากหลักฐานที่นำมากล่าวอ้างไม่ถูกต้องผู้ใช้จะถูกปฏิเสธจากระบบ

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 ลักษณะข้างต้นมาใช้ในการยืนยันหลักฐานที่นำมากล่าวอ้าง ทั้งนี้ขึ้นอยู่กับระบบ วิธีการที่นำมาใช้เพียงลักษณะอย่างใดอย่างหนึ่ง (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 การใช้ลายมือชื่อร่วมกับการใช้บัตรเครดิต หรือการใช้รหัสผ่านร่วมกับการใช้บัตร ATM เป็นต้น การนำแต่ละลักษณะของการพิสูจน์ตัวตนมาใช้ร่วมกันมากกว่า 1 ลักษณะ จะช่วยเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูล

2.3.2.2 การกำหนดสิทธิ์ (Authorization)

การกำหนดสิทธิ์ คือขั้นตอนในการอนุญาตให้แต่ละบุคคลสามารถเข้าถึงข้อมูลหรือระบบใดได้บ้าง ก่อนอื่นต้องทราบก่อนว่าบุคคลที่กล่าวอ้างนั้นคือใครตามขั้นตอนการพิสูจน์ตัวตนและต้องให้แน่ใจด้วยว่าการพิสูจน์ตัวตนนั้นถูกต้อง

2.3.2.3 การเข้ารหัส (Encryption)

การเข้ารหัสคือการเก็บข้อมูลให้เป็นส่วนบุคคลจากบุคคลอื่นที่ไม่ได้รับอนุญาต ส่วนประกอบ 2 ส่วนที่สำคัญที่จะช่วยทำให้ข้อมูลนั้นเป็นความลับได้ก็คือ การกำหนดสิทธิ์และการพิสูจน์ตัวตนเพราะว่าก่อนการอนุญาตให้บุคคลที่กล่าวอ้างเข้าถึงข้อมูล หรือถอดรหัสข้อมูลนั้นต้องสามารถแน่ใจได้ว่าบุคคลที่กล่าวอ้างนั้นเป็นใคร และได้รับอนุญาตให้สามารถเข้ามาดูข้อมูลได้หรือไม่ ในการเข้ารหัสนั้นวิธีการหนึ่งที่ทำได้คือการเข้ารหัสในรูปแบบของกุญแจลับ (Secret key) ซึ่งในการใช้คีย์รูปแบบนี้ต้องเฉพาะผู้ที่มีกุญแจลับนี้เท่านั้นที่สามารถรับข้อมูลที่เข้ารหัสแล้วได้

2.3.2.4 การรักษาความสมบูรณ์ (Integrity)

การรักษาความสมบูรณ์ คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไปจากต้นฉบับ (source) ไม่ว่าจะเป็นโดยบังเอิญหรือดัดแปลงโดยเจตนาที่อาจส่งผลเสียต่อข้อมูล การคุกคามความสมบูรณ์ของข้อมูลคือ การที่บุคคลที่ไม่ได้รับอนุญาตสามารถที่จะเข้าควบคุมการจัดการของข้อมูลได้

2.3.2 5 การตรวจสอบ (Audit)

การตรวจสอบ คือการตรวจสอบหลักฐานทางอิเล็กทรอนิกส์ ซึ่งสามารถใช้ในการติดตามการดำเนินการ เพื่อตรวจสอบความถูกต้องและแม่นยำ ตัวอย่างเช่นการตรวจสอบบัญชีชื่อผู้ใช้โดยผู้ตรวจบัญชี ซึ่งการตรวจสอบความถูกต้องของการดำเนินการ เพื่อให้แน่ใจว่าหลักฐานทางอิเล็กทรอนิกส์นั้นได้ถูกสร้าง และสั่งให้ทำงานโดยบุคคลที่ได้รับอนุญาต โดยในการเชื่อมต่อ

เหตุการณ์เข้ากับบุคคลจะต้องทำการตรวจสอบหลักฐานของบุคคลนั้นด้วย ซึ่งถือเป็นหลักการพื้นฐานของขั้นตอนการทำงานของการพิสูจน์ตัวตนด้วย

การพิสูจน์ตัวตนจัดเป็นการตรวจสอบหลักฐานขั้นพื้นฐานที่สำคัญที่สุดใน 5 ระดับชั้นของการควบคุมความปลอดภัย ดังนั้นการพิสูจน์ตัวตนดีจะช่วยเพิ่มความมั่นคงปลอดภัยขั้นพื้นฐานให้กับระบบมากยิ่งขึ้น

2.3.3 ประเภทของการพิสูจน์ตัวตน (Authentication Types)

ส่วนประกอบพื้นฐานของการพิสูจน์ตัวตนสมบูรณ์แบ่งได้เป็น 3 ส่วน คือ

- การพิสูจน์ตัวตน (Authentication) คือส่วนที่สำคัญที่สุดเพราะเป็นขั้นตอนแรกของการเข้าใช้ระบบ ผู้เข้าใช้ระบบต้องถูกยอมรับจากระบบว่าสามารถเข้าสู่ระบบได้ การพิสูจน์ตัวตนเป็นการตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลนั้นจริง
- การกำหนดสิทธิ์ (Authorization) คือข้อจำกัดของบุคคลที่เข้ามาในระบบ ว่าบุคคลคนนั้นสามารถทำอะไรกับระบบได้บ้าง
- การบันทึกการใช้งาน (Accountability) คือการบันทึกรายละเอียดของการใช้ระบบและรวมถึงข้อมูลต่างๆ ที่ผู้ใช้กระทำลงไปในระบบ เพื่อผู้ตรวจสอบจะได้ตรวจสอบได้ว่าผู้ใช้ที่เข้ามาใช้บริการได้เปลี่ยนแปลงหรือแก้ไขข้อมูลในส่วนใดบ้าง

2.3.4 การพิสูจน์ตัวตนโดยการเข้ารหัสโดยใช้กุญแจสาธารณะ (Public-key cryptography)

เป็นการรักษาความปลอดภัยของข้อมูลระหว่างการส่งข้ามเครือข่ายวิธี ที่นิยมใช้กันในปัจจุบัน การเข้ารหัสแบบคู่รหัสกุญแจนี้จะมีความปลอดภัยมากกว่าการเข้ารหัสข้อมูลแบบธรรมดา แต่ก็ไม่ ได้หมายความว่าการเข้ารหัสแบบคู่รหัสกุญแจนี้จะเป็นวิธีที่เหมาะสมที่สุดของวิธีการเข้ารหัส ทั้งนี้ขึ้นอยู่กับประเภทงานของแต่ละองค์กรหรือบุคคล

การเข้ารหัสโดยใช้กุญแจสาธารณะ ประกอบไปด้วยกุญแจ 2 ชนิด ที่ต้องใช้คู่กันเสมอในการเข้ารหัสและถอดรหัสคือ

- กุญแจสาธารณะ (public key) เป็นกุญแจที่ผู้สร้างจะส่งออกไปให้ผู้ใช้อื่นๆ ทราบหรือเปิดเผยได้
- กุญแจส่วนตัว (private key) เป็นกุญแจที่ผู้สร้างจะเก็บไว้ โดยไม่เปิดเผยให้คนอื่นรู้

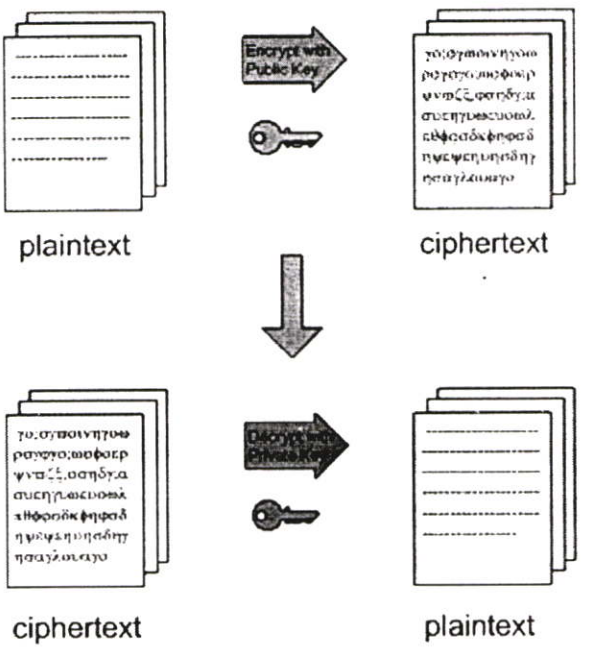
กระบวนการของการเข้ารหัสแบบคู่รหัสกุญแจ มีดังนี้

1. ผู้ใช้แต่ละคนจะสร้างคู่รหัสกุญแจของตัวเองขึ้นมา เพื่อใช้สำหรับการเข้ารหัสและการถอดรหัส
2. กุญแจสาธารณะจะถูกส่งออกไปยังผู้ใช้คนอื่นๆ แต่กุญแจส่วนตัวจะถูกเก็บที่ตนเอง
3. เมื่อจะส่งข้อมูลออกไปหาผู้ใช้คนใด ข้อมูลที่ส่งจะถูกเข้ารหัสด้วยกุญแจสาธารณะ ก่อนถูกส่งออกไป

4. เมื่อผู้รับได้รับข้อความแล้วจะใช้กุญแจส่วนตัวซึ่งเป็นคู่รหัสกันถอดรหัสออกมา

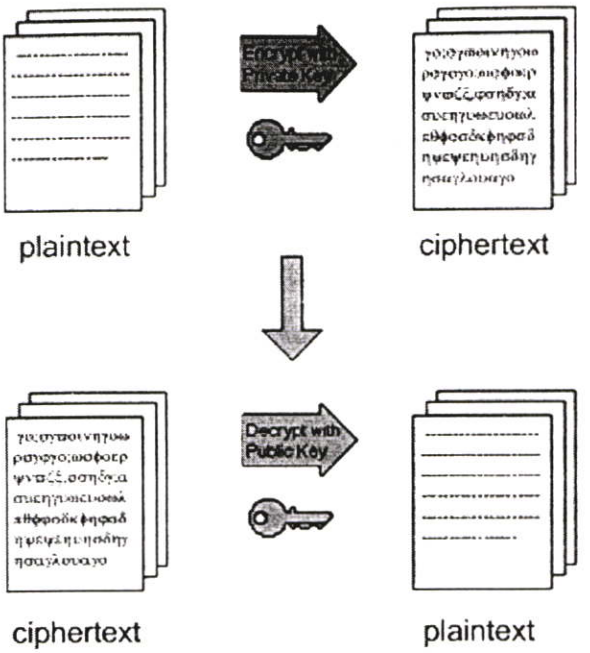
การเข้ารหัสโดยใช้กุญแจสาธารณะสามารถใช้ได้ทั้งในการเข้ารหัส (Encryption) และการพิสูจน์ตัวตน (Authentication)

การประยุกต์ใช้ในการเข้ารหัสข้อมูล (Encryption) เป็นการนำข้อมูลที่จะส่งไปยังผู้รับมาเข้ารหัสด้วยกุญแจสาธารณะของผู้รับ และเมื่อผู้รับได้รับข้อความนั้นแล้วจะถอดรหัสออกมาด้วยกุญแจส่วนตัว จึงจะเห็นได้ว่ามีเพียงผู้รับเท่านั้นที่จะสามารถถอดรหัสออกมาได้



รูปที่ 2.13 ระบบของการเข้ารหัสแบบใช้คู่รหัสกุญแจ

การประยุกต์ใช้ในการพิสูจน์ตัวตน (Authentication) เป็นการนำข้อมูลที่ผู้ส่งต้องการส่งมาเข้ารหัสด้วยกุญแจส่วนตัวของผู้ส่ง แล้วนำข้อมูลนั้นส่งไปยังผู้รับ ซึ่งผู้รับจะใช้กุญแจสาธารณะซึ่งเป็นคู่รหัสกันถอดรหัสออกมา ผู้รับก็สามารถรู้ได้ว่าข้อความนั้นถูกส่งมาจากผู้ส่งคนนั้นจริง ถ้าสามารถถอดรหัสข้อมูลได้อย่างถูกต้อง



รูปที่ 2.14 ระบบของการเข้ารหัสแบบใช้คู่รหัสกุญแจเพื่อการพิสูจน์ตัวตน

2.3.5 การเข้ารหัสข้อมูล (Cryptography)

ในส่วนนี้มีจุดประสงค์เพื่อให้ความรู้เกี่ยวกับการป้องกันสารสนเทศโดยการเข้ารหัสข้อมูล รวมทั้งครอบคลุมถึงอัลกอริธึมที่ใช้ในการเข้ารหัสที่สำคัญๆ และมีความแพร่หลายสูงที่ผู้อ่านมักจะได้อ่านพบในที่ต่างๆ เช่น บทความบนเว็บ หนังสือ หรือสิ่งตีพิมพ์อื่นๆ

จุดประสงค์ที่สำคัญ 3 ประการของการเข้ารหัสข้อมูลประกอบด้วย [5],[6]

1. การทำให้ข้อมูลเป็นความลับ (Confidentiality) เพื่อป้องกันไม่ให้ผู้ที่ไม่มีสิทธิ์ในการเข้าถึงข้อมูลสามารถเข้าถึงข้อมูลได้
2. การทำให้ข้อมูลสามารถตรวจสอบความสมบูรณ์ได้ (Integrity) เพื่อป้องกันข้อมูลให้อยู่ในสภาพเดิมอย่างสมบูรณ์ กล่าวคือ ในกระบวนการสื่อสารนั้นผู้รับ (Receiver) ได้รับข้อมูลที่ถูกต้องตามที่ผู้ส่ง (Sender) ส่งมาให้โดยข้อมูลจะต้องไม่มีการสูญหายหรือถูกเปลี่ยนแปลงแก้ไขใดๆ
3. การทำให้สามารถพิสูจน์ตัวตนของผู้ส่งข้อมูลได้ (Authentication/Nonrepudiation) เพื่อให้สามารถตรวจสอบได้ว่าใครคือผู้ส่งข้อมูล หรือในทางตรงกันข้าม ก็คือเพื่อป้องกันการแอบอ้างได้

การเข้ารหัสข้อมูลโดยพื้นฐานแล้วจะเกี่ยวข้องกับวิธีการทางคณิตศาสตร์เพื่อใช้ในการป้องกันข้อมูลหรือข้อความตั้งต้นที่ต้องการส่งไปถึงผู้รับ ข้อมูลตั้งต้นจะถูกแปรเปลี่ยนไปสู่ข้อมูลหรือข้อความอีกรูปแบบหนึ่งที่ไม่สามารถอ่านเข้าใจได้ โดยใครก็ตามที่ไม่มีกุญแจสำหรับเปิดดูข้อมูลนั้น เราเรียกกระบวนการในการแปรรูปของข้อมูลตั้งต้นว่า "การเข้ารหัสข้อมูล" (Encryption)

และกระบวนการในการแปลงข้อความที่ไม่สามารถอ่าน และทำความเข้าใจให้กลับไปสู่ข้อความดั้งเดิม ว่าการถอดรหัสข้อมูล (Decryption)

2.3.5.1 อัลกอริธึมในการเข้ารหัสข้อมูล

อัลกอริธึมในการเข้ารหัสข้อมูลมี 2 ประเภทหลัก คือ

- อัลกอริธึมแบบสมมาตร (Symmetric key algorithms)
  อัลกอริธึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ (Secret key) ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป อัลกอริธึมยังสามารถแบ่งย่อยออกเป็น 2 ประเภท ได้แก่ แบบบล็อค (Block Algorithms) ซึ่งจะทำการเข้ารหัสทีละบล็อค (1 บล็อคประกอบด้วยหลายไบต์ เช่น 64 ไบต์ เป็นต้น) และแบบสตรีม (Stream Algorithms) ซึ่งจะทำการเข้ารหัสทีละไบต์
- อัลกอริธึมแบบอสมมาตร (Asymmetric key algorithms)
  อัลกอริธึมนี้จะใช้กุญแจสองตัวเพื่อทำงาน ตัวหนึ่งใช้ในการเข้ารหัสและอีกตัวหนึ่งใช้ในการถอดรหัสข้อมูลที่เข้ารหัสมาโดยกุญแจตัวแรก อัลกอริธึมกลุ่มสำคัญในแบบอสมมาตรนี้คือ อัลกอริธึมแบบกุญแจสาธารณะ (Public keys Algorithms) ซึ่งใช้กุญแจที่เรียกกันว่า กุญแจสาธารณะ (Public keys) ในการเข้ารหัสและใช้กุญแจที่เรียกกันว่า กุญแจส่วนตัว (Private keys) ในการถอดรหัสข้อมูลนั้น กุญแจสาธารณะนี้สามารถส่งมอบให้กับผู้อื่นได้ เช่น เพื่อนร่วมงานที่เราต้องการติดต่อด้วย หรือแม้กระทั่งวางไว้บนเว็บไซต์เพื่อให้ผู้อื่นสามารถดาวน์โหลดไปใช้งานได้ สำหรับกุญแจส่วนตัวนั้นต้องเก็บไว้กับผู้เป็นเจ้าของกุญแจส่วนตัวเท่านั้นและห้ามเปิดเผยให้ผู้อื่นทราบโดยเด็ดขาด

อัลกอริธึมแบบกุญแจสาธารณะ ยังสามารถประยุกต์ใช้ได้กับการลงลายมือชื่ออิเล็กทรอนิกส์ (ซึ่งเปรียบเสมือนการลงลายมือชื่อของเราที่ใช้กับเอกสารสำนักงานทั่วไป) การลงลายมือชื่อนี้จะเป็นการพิสูจน์ความเป็นเจ้าของและสามารถใช้ได้กับการทำธุรกรรมต่างๆ บนอินเทอร์เน็ต เช่น การซื้อสินค้า เป็นต้น วิธีการใช้งานคือ ผู้เป็นเจ้าของกุญแจส่วนตัวลงลายมือชื่อของตนกับข้อความที่ต้องการส่งไปด้วยกุญแจส่วนตัว แล้วจึงส่งข้อความนั้นไปให้กับผู้รับ เมื่อได้รับข้อความที่ลงลายมือชื่อมา ผู้รับสามารถใช้กุญแจสาธารณะ (ที่เป็นคู่ของกุญแจส่วนตัวนั้น) เพื่อตรวจสอบว่าเป็นข้อความที่มาจากผู้ส่งนั้นหรือไม่

2.3.5.2 ปัญหาของอัลกอริธึมแบบสมมาตร

อัลกอริธึมแบบสมมาตรมีความสำคัญไม่ด้อยไปกว่าอัลกอริธึมแบบอสมมาตร ทั้งนี้เนื่องจากอัลกอริธึมแบบแรกทำงานได้รวดเร็วกว่า และง่ายต่อการใช้งานกว่าแบบหลัง อย่างไรก็ตามอัลกอริธึมแบบสมมาตรยังมีปัญหาที่สำคัญ 3 ประการ ซึ่งเป็นข้อจำกัดในการใช้งานอัลกอริธึมนี้

1. ในการใช้งานอัลกอริธึมนี้ สองกลุ่มที่ต้องการแลกเปลี่ยนข้อมูลกัน (เช่น องค์กร ก และ ข) จำเป็นต้องแลกเปลี่ยนกุญแจลับกันก่อน (ซึ่งอาจหมายถึงส่งมอบกุญแจลับให้กับอีกกลุ่มหนึ่ง) การแลกเปลี่ยนกุญแจนั้นอาจทำได้อย่างยุ่งยากและไม่สะดวก
2. ทั้งสองกลุ่มต้องรักษากุญแจลับนั้นไว้เป็นอย่างดี ห้ามเปิดเผยให้ผู้อื่นล่วงรู้โดยเด็ดขาด การที่กุญแจถูกเปิดเผยออกไปสู่ผู้อื่น (จะโดยกลุ่มใดกลุ่มหนึ่งก็ตาม) และอีกกลุ่มหนึ่งไม่ได้รับทราบปัญหานี้ อาจก่อให้เกิดปัญหากับกลุ่มที่ไม่ทราบนี้ได้ เช่น กลุ่มนี้อาจส่งข้อความที่เป็นความลับไปให้กับอีกกลุ่มหนึ่ง แต่ข้อความนี้อาจถูกเปิดเผยได้โดยใช้กุญแจลับที่ล่วงรู้โดยผู้อื่น
3. สำหรับสองกลุ่มที่ต้องการติดต่อกัน จำเป็นต้องใช้กุญแจลับเป็นจำนวน 1 กุญแจเพื่อติดต่อกัน สมมติว่ามีผู้ที่ต้องติดต่อกันเป็นจำนวน n กลุ่ม จำนวนกุญแจลับทั้งหมดที่ต้องแลกเปลี่ยนกันคิดเป็นจำนวนทั้งหมด C 2n หรือเท่ากับ n (n-1)/2 กุญแจ ซึ่งจะเห็นได้ว่าจำนวนกุญแจมีมากมายเกินไป ซึ่งอาจก่อให้เกิดปัญหาด้านการรักษาความปลอดภัยให้กับกุญแจเหล่านี้

อัลกอริธึมแบบกุญแจสาธารณะ (ซึ่งเป็นแบบอสมมาตร) ช่วยแก้ปัญหาเหล่านี้ได้ทั้งหมด ผู้ใช้ที่ถือกุญแจส่วนตัว และต้องการให้บุคคลอื่นที่ตนติดต่อด้วยส่งเอกสารหรือข้อความที่เข้ารหัสมาหาตน สามารถเผยแพร่กุญแจสาธารณะของตนไว้บนเว็บไซต์หรือในที่สาธารณะซึ่งผู้อื่นสามารถเข้ามาดาวน์โหลดไปใช้งานได้ วิธีการใช้งานคือให้บุคคลอื่นที่มาดาวน์โหลดกุญแจไปนั้นทำการเข้ารหัสข้อความที่ต้องการส่งด้วยกุญแจสาธารณะ แล้วจึงส่งข้อความที่เข้ารหัสไปให้กับผู้เป็นเจ้าของกุญแจสาธารณะ โดยวิธีนี้จะไม่มีผู้อื่นสามารถเปิดดูข้อความที่เข้ารหัสนั้นได้ยกเว้นผู้ที่ถือกุญแจส่วนตัว (ที่เป็นคู่ของกุญแจสาธารณะนั้น) จึงจะสามารถเปิดข้อความนี้ดูได้

การเผยแพร่กุญแจสาธารณะในสถานที่ต่างๆ ได้ทำให้ลดความยุ่งยากในการแลกเปลี่ยนกุญแจกันซึ่งเป็นปัญหาข้อแรกของการเข้ารหัสแบบสมมาตร สำหรับปัญหาที่ว่าทั้งสองกลุ่มจะต้องรักษากุญแจลับไว้เป็นอย่างดีนั้น วิธีการของกุญแจสาธารณะจะทำให้ผู้ที่ต้องรับผิดชอบเหลือเพียงผู้เดียว กล่าวคือ ผู้ถือกุญแจส่วนตัว ซึ่งห้ามให้ผู้อื่นล่วงรู้โดยเด็ดขาด

สำหรับปัญหาที่สามที่ว่าจำนวนกุญแจลับที่จำเป็นต้องใช้มีมากมายเกินไป วิธีการของกุญแจสาธารณะจะใช้จำนวนกุญแจที่ประหยัดกว่า เนื่องจากกุญแจสาธารณะ 1 กุญแจของกลุ่มๆ หนึ่งจะสามารถเผยแพร่ให้กับกี่กลุ่มก็ได้ที่เราต้องการติดต่อด้วย (แทนที่จะเป็น 1 กุญแจลับต่อสองกลุ่มที่ต้องการติดต่อกัน) ดังนั้นถ้ามีกลุ่มที่ต้องติดต่อกันจำนวน n กลุ่ม จำนวนกุญแจส่วนตัวที่ต้องระวังรักษาก็คือ n กุญแจ ซึ่งจะเห็นได้ว่าลดลงไปได้เป็นจำนวนมาก

ข้อเสียที่สำคัญของระบบกุญแจสาธารณะที่สำคัญคือ ต้องใช้เวลาในการคำนวณการเข้าและถอดรหัส เมื่อเทียบกับระบบกุญแจสมมาตร และอาจใช้เวลาเป็นพันเท่าของเวลาที่ใช้โดยระบบกุญแจสมมาตร

2.3.6 ความแข็งแกร่งของอัลกอริธึมสำหรับการเข้ารหัส

ความแข็งแกร่งของอัลกอริธึม หมายถึงความยากในการที่ผู้บุกรุกจะสามารถถอดรหัสข้อมูลได้โดยปราศจากกุญแจที่ใช้ในการเข้ารหัส ซึ่งจะขึ้นอยู่กับปัจจัยดังนี้

- การเก็บกุญแจเข้ารหัสไว้อย่างเป็นความลับ ผู้เป็นเจ้าของกุญแจลับหรือส่วนตัวต้องระมัดระวังไม่ให้กุญแจสูญหายหรือล่วงรู้โดยผู้อื่น
- ความยาวของกุญแจเข้ารหัส ปกติกุญแจเข้ารหัสจะมีความยาวเป็นบิต ยิ่งจำนวนบิตของกุญแจยิ่งมาก ยิ่งทำให้การเดาเพื่อสุ่มหากุญแจที่ถูกต้องเป็นไปได้ยากยิ่งขึ้น (เช่น กุญแจขนาด 1 บิต จะสามารถแทนตัวเลขได้ 2 ค่าคือ 0 กับ 1 กุญแจขนาด 2 บิต จะเป็นไปได้ 4 ค่าคือ 0, 1, 2, 3 เป็นต้น)
- ความไม่เกรงกลัวต่อการศึกษาหรือดูอัลกอริธึมเพื่อหารูปแบบของการเข้ารหัส อัลกอริธึมที่ดีต้องเปิดให้ผู้รู้ทำการศึกษาในรายละเอียดได้โดยไม่เกรงว่าผู้ศึกษาจะสามารถจับรูปแบบของการเข้ารหัสได้
- การมีประตูลับในอัลกอริธึม อัลกอริธึมที่ดีต้องไม่แฝงไว้ด้วยประตูลับที่สามารถใช้เป็นทางเข้าไปสู่อัลกอริธึม แล้วอาจใช้เพื่อทำการถอดรหัสข้อมูลได้ ประตูลับนี้ทำให้ไม่จำเป็นต้องใช้กุญแจในการถอดรหัส
- ความไม่เกรงกลัวต่อปัญหาการหาความสัมพันธ์ในข้อมูลที่ได้รับ กล่าวคือเมื่อผู้บุกรุกทราบข้อมูลบางอย่างที่เป็นข้อมูลตั้งต้นซึ่งยังไม่ได้เข้ารหัส รวมทั้งมีข้อมูลที่เข้ารหัสแล้ว (ของข้อมูลตั้งต้นนั้น) ผู้บุกรุกอาจจะสามารถหาความสัมพันธ์ระหว่างข้อความทั้งสองนั้นได้ ซึ่งจะเป็นวิธีการในการถอดรหัสข้อมูลได้ ปัญหานี้เรียกกันว่า Known plaintext attack (คำว่า plaintext หมายถึงข้อความตั้งต้นที่ยังไม่ได้ผ่านการเข้ารหัส)
- คุณสมบัติของข้อความตั้งต้น คุณสมบัตินี้อาจใช้เป็นช่องทางในการถอดรหัสข้อมูลได้ อัลกอริธึมที่ดีต้องไม่ใช้คุณสมบัติของข้อความเป็นกลไกในการเข้ารหัสข้อมูล

คำแนะนำในการเลือกใช้อัลกอริธึม คือให้ใช้อัลกอริธึมที่ได้มีการใช้งานมาเป็นระยะเวลานานแล้ว ทั้งนี้เนื่องจากหากปัญหาของอัลกอริธึมนี้มีจริง ก็คงเกิดขึ้นมานานแล้วและก็คงเป็นที่ทราบกันแล้ว นั่นคืออย่างน้อยที่สุดจวบจนกระทั่งถึงปัจจุบัน ก็ยังไม่มีการบุกรุกที่ทำให้อัลกอริธึมนั้นไม่สามารถใช้งานได้อย่างปลอดภัยเป็นที่ประจักษ์ ดังนั้นจึงไม่ควรใช้อัลกอริธึมใหม่ๆ ที่เพิ่งได้มีการนำเสนอกันสู่สาธารณะ เพราะอาจมีช่องโหว่แฝงอยู่และยังไม่เป็นที่ทราบในขณะนี้

2.3.7 ความยาวของกุญแจที่ใช้ในการเข้ารหัส

ความยาวของกุญแจเข้ารหัสมีหน่วยนับเป็นบิต หนึ่งบิตในคอมพิวเตอร์เป็นตัวเลขฐานสองที่ประกอบด้วยค่า 0 และ 1 กุญแจที่มีความยาว 1 บิต ตัวเลขที่เป็นไปได้เพื่อแทนกุญแจนั้น จึงอาจมีค่าเป็น 0 หรือ 1 กุญแจที่มีความยาว 2 บิต ตัวเลขที่เป็นไปได้จึงเป็น 0, 1, 2 และ 3

ตามลำดับ กุญแจที่มีความยาว 3 บิต ตัวเลขที่เป็นไปได้จะอยู่ระหว่าง 0 ถึง 7 ดังนั้นเมื่อเพิ่มความยาวของกุญแจทุกๆ 1 บิต ค่าที่เป็นไปได้ของกุญแจจะเพิ่มขึ้นเป็นสองเท่าตัว หรือจำนวนกุญแจที่เป็นไปได้จะเพิ่มขึ้นเป็น 2 เท่าตัวนั่นเอง

ฉะนั้นจะเห็นได้ว่ากุญแจยิ่งมีความยาวมาก โอกาสที่ผู้บุกรุกจะสามารถคาดเดากุญแจที่ตรงกับหมายเลขที่ถูกต้องของกุญแจจะยิ่งยากมากขึ้นตามลำดับ ในการที่ผู้บุกรุกลองผิดลองถูกกับกุญแจโดยใช้กุญแจที่มีหมายเลขต่างๆ กัน เพื่อหวังที่จะพบกุญแจที่ถูกต้องและสามารถใช้ถอดรหัสข้อมูลได้ การลองผิดลองถูกนี้เราเรียกกันว่า Key search หรือการค้นหากุญแจนั่นเอง ทฤษฎีได้กล่าวไว้ว่าการลองผิดลองถูกนี้โดยเฉลี่ยจะต้องทดลองกับกุญแจเป็นจำนวนครึ่งหนึ่งของกุญแจทั้งหมดก่อนที่จะพบกุญแจที่ถูกต้อง

ความยาวของกุญแจที่มีขนาดเหมาะสม จึงขึ้นอยู่กับความเร็วในการค้นหากุญแจของผู้บุกรุกและระยะเวลาที่ต้องการให้ข้อมูลมีความปลอดภัย ตัวอย่างเช่น ถ้าผู้บุกรุกสามารถลองผิดลองถูกกับกุญแจเป็นจำนวน 10 กุญแจภายในหนึ่งวินาทีแล้ว กุญแจที่มีความยาว 40 บิต จะสามารถป้องกันข้อมูลไว้ได้ 3,484 ปี ถ้าผู้บุกรุกสามารถลองได้เป็นจำนวน 1 ล้านกุญแจในหนึ่งวินาที (เทคโนโลยีปัจจุบันสามารถทำได้) กุญแจที่มีความยาว 40 บิตจะสามารถป้องกันข้อมูลไว้ได้เพียง 13 วันเท่านั้น (ซึ่งอาจไม่เพียงพอสำหรับในบางลักษณะงาน) ด้วยเทคโนโลยีในปัจจุบันหากผู้บุกรุกสามารถทดลองได้เป็นจำนวน 1,000 ล้านกุญแจในหนึ่งวินาที กุญแจขนาด 128 บิตจะสามารถป้องกันข้อมูลไว้ได้ 1022 ปี ดังนั้นด้วยลักษณะงานทั่วไปกุญแจขนาด 128 บิตจะพอเพียงต่อการรักษาความลับของข้อมูลเอาไว้ได้

2.3.8 อัลกอริธึมสำหรับการเข้ารหัสแบบสมมาตร

อัลกอริธึมสำหรับการเข้ารหัสแบบสมมาตรในปัจจุบันมีเป็นจำนวนมาก ข้างล่างนี้จะนำเสนอเพียงจำนวนหนึ่งซึ่งเป็นอัลกอริธึมที่เป็นที่รู้จักกันดีในวงการของการเข้ารหัสข้อมูล

2.3.8.1 อัลกอริธึม DES

DES ย่อมาจาก Data Encryption Standard อัลกอริธึมนี้ได้รับการรับรองโดยรัฐบาลสหรัฐอเมริกาในปี ค.ศ. 1977 ให้เป็นมาตรฐานการเข้ารหัสข้อมูลสำหรับหน่วยงานของรัฐทั้งหมด ในปี 1981 อัลกอริธึมยังได้รับการกำหนดให้เป็นมาตรฐานการเข้ารหัสข้อมูลในระดับนานาชาติตามมาตรฐาน ANSI (American National Standards) อีกด้วย

DES เป็นอัลกอริธึมแบบบล็อกซึ่งใช้กุญแจที่มีขนาดความยาว 56 บิตและเป็นอัลกอริธึมที่มีความแข็งแกร่ง แต่เนื่องด้วยขนาดความยาวของกุญแจที่มีขนาดเพียง 56 บิต ซึ่งในปัจจุบันถือได้ว่าสั้นเกินไป ผู้บุกรุกอาจใช้วิธีการลองผิดลองถูกเพื่อค้นหากุญแจที่ถูกต้องสำหรับการถอดรหัสได้

2.3.8.2 อัลกอริธึม Triple-DES

Triple-DES เป็นอัลกอริธึมที่เสริมความปลอดภัยของ DES ให้มีความแข็งแกร่งมากขึ้นโดยใช้อัลกอริธึม DES เป็นจำนวนสามครั้งเพื่อทำการเข้ารหัส แต่ละครั้งจะใช้กุญแจในการเข้ารหัสที่แตกต่างกัน ดังนั้นจึงเปรียบเสมือนการใช้กุญแจเข้ารหัสที่มีความยาวเท่ากับ 56*3 = 168 บิต Triple-DES ได้ถูก ใช้งานกับสถาบันทางการเงินอย่างแพร่หลาย รวมทั้งใช้งานกับโปรแกรม Secure Shell (ssh) ด้วย

การใช้อัลกอริธึม DES เพื่อเข้ารหัสเป็นจำนวนสองครั้งด้วยกุญแจสองตัว (56*2=112 บิต) ยังถือได้ว่าไม่ปลอดภัยอย่างพอเพียง

2.3.8.3 อัลกอริธึม RC4

อัลกอริธึมนี้เป็นอัลกอริธึมแบบสตรีม (ทำงานกับข้อมูลทีละไบต์) ซึ่งได้รับการพัฒนาขึ้นมาโดย Ronald Riverst และถูกเก็บเป็นความลับทางการค้าโดยบริษัท RSA Data Security ในภายหลังอัลกอริธึมนี้ได้รับการเปิดเผยใน Usenet เมื่อปี ค.ศ. 1994 และเป็นที่ทราบกันว่าเป็นอัลกอริธึมที่มีความแข็งแกร่งโดยสามารถใช้ขนาดความยาวของกุญแจที่มีขนาดตั้งแต่ 1 บิตไปจนกระทั่งถึงขนาด 2048 บิต

2.3.9 อัลกอริธึมสำหรับการเข้ารหัสแบบอสมมาตร

อัลกอริธึมแบบกุญแจสาธารณะ แบ่งตามลักษณะการใช้งานได้เป็น 2 ประเภท คือ

- ใช้สำหรับการเข้ารหัส
- ใช้สำหรับการลงลายมือชื่ออิเล็กทรอนิกส์

อัลกอริธึมที่เป็นที่รู้จักกันดีมีดังนี้

2.3.9.1 อัลกอริธึม RSA

อัลกอริธึม RSA ได้รับการพัฒนาขึ้นที่มหาวิทยาลัย MIT ในปี 1977 โดยศาสตราจารย์ 3 คน ซึ่งประกอบด้วย Ronald Rivest, Adi Shamir และ Leonard Adleman ชื่อของอัลกอริธึมได้รับการตั้งชื่อตามตัวอักษรตัวแรกของนามสกุลของศาสตราจารย์ทั้งสามคน อัลกอริธึมนี้สามารถใช้ในการเข้ารหัสข้อมูลรวมทั้งการลงลายมือชื่ออิเล็กทรอนิกส์ด้วย

2.3.9.2 อัลกอริธึม DSS

DSS ย่อมาจาก Digital Signature Standard อัลกอริธึมนี้ได้รับการพัฒนาขึ้นมาโดย National Security Agency ในประเทศสหรัฐอเมริกาและได้รับการรับรองโดย NIST ให้เป็นมาตรฐานกลางสำหรับการลงลายมือชื่ออิเล็กทรอนิกส์ในประเทศสหรัฐอเมริกา

2.3.10 อัลกอริธึมสำหรับสร้างเมสเซสไดเจสต์

เมสเซสไดเจสต์ (Message Digest) หรือเรียกสั้นๆ ว่าไดเจสต์ แปลว่าข้อความสรุปจากเนื้อหาข้อความตั้งต้น โดยปกติข้อความสรุปจะมีความยาวน้อยกว่าความยาวของข้อความตั้งต้นมาก จุดประสงค์สำคัญของอัลกอริธึมนี้คือ การสร้างข้อความสรุปที่สามารถใช้เป็นตัวแทนของข้อความตั้งต้นได้ โดยทั่วไปข้อความสรุปจะมีความยาวอยู่ระหว่าง 128 ถึง 256 บิต และจะไม่ขึ้นกับขนาดความยาวของข้อความตั้งต้น

คุณสมบัติที่สำคัญของอัลกอริธึมสำหรับสร้างไดเจสต์มีดังนี้

- ทุกๆ บิตของไดเจสต์จะขึ้นอยู่กับทุกบิตของข้อความตั้งต้น
- ถ้าบิตใดบิตหนึ่งของข้อความตั้งต้นเกิดการเปลี่ยนแปลง เช่น ถูกแก้ไข ทุกๆ บิตของไดเจสต์จะมีโอกาสร้อยละ 50 ที่จะแปรเปลี่ยนค่าไปด้วย ซึ่งหมายถึงว่า 0 เปลี่ยนค่าเป็น 1 และ 1 เปลี่ยนเป็น 0 คุณสมบัติข้อนี้สามารถอธิบายได้ว่าการเปลี่ยนแปลงแก้ไขข้อความตั้งต้นโดยผู้ไม่ประสงค์ดีแม้ว่าอาจแก้ไขเพียงเล็กน้อยก็ตาม เช่น เพียง 1 บิตเท่านั้น ก็จะส่งผลให้ผู้รับข้อความทราบว่าข้อความที่ตนได้รับไม่ใช่ข้อความตั้งต้น (โดยการนำข้อความที่ตนได้รับเข้าอัลกอริธึมเพื่อทำการคำนวณหาไดเจสต์ออกมา แล้วจึงเปรียบเทียบไดเจสต์ที่คำนวณได้กับไดเจสต์ที่ส่งมาให้ด้วย ถ้าต่างกัน แสดงว่าข้อความที่ได้รับนั้นถูกเปลี่ยนแปลงแก้ไข)
- โอกาสที่ข้อความตั้งต้น 2 ข้อความใดๆ ที่มีความแตกต่างกัน จะสามารถคำนวณได้ค่าไดเจสต์เดียวกันมีโอกาสน้อยมากคุณสมบัติข้อนี้ทำให้แน่ใจได้ว่า เมื่อผู้ไม่ประสงค์ดีทำการแก้ไขข้อความตั้งต้น ผู้รับข้อความที่ถูกแก้ไขไปแล้วนั้นจะสามารถตรวจพบได้ถึงความผิดปกติที่เกิดขึ้นอย่างแน่นอนอย่างไรก็ตามในทางทฤษฎีแล้ว มีโอกาสที่ข้อความ 2 ข้อความที่แตกต่างกันจะสามารถคำนวณแล้วได้ค่าไดเจสต์เดียวกัน ปัญหานี้เรียกกันว่าการชนกันของไดเจสต์(Collision) อัลกอริธึมสำหรับสร้างไดเจสต์ที่ดีควรจะมีโอกาสน้อยมากๆ ที่จะก่อให้เกิดปัญหาการชนกันของไดเจสต์

อัลกอริธึมสำหรับสร้างไดเจสต์ยอดนิยมมีดังนี้

2.3.10.1 อัลกอริธึม MD5

Rivest เป็นผู้พัฒนาเช่นกันโดยพัฒนาต่อจาก MD4 เพื่อให้มีความปลอดภัยที่สูงขึ้น ถึงแม้จะเป็นที่นิยมใช้งานกันอย่างแพร่หลาย ทว่าในปี 1996 ก็มีผู้พบจุดบกพร่องของ MD5 (เช่นเดียวกับ MD4) จึงทำให้ความนิยมเริ่มลดลง MD5 ผลิตไดเจสต์ที่มีขนาด 128 บิต

2.3.10.2 อัลกอริธึม SHA

SHA ย่อจาก Secure Hash Algorithm อัลกอริธึม SHA ได้รับแนวคิดในการพัฒนามาจาก MD4 และได้รับการพัฒนาขึ้นมาเพื่อใช้งานร่วมกับอัลกอริธึม DSS (ซึ่งใช้ในการลงลายมือชื่ออิเล็กทรอนิกส์) หลังจากที่ได้มีการตีพิมพ์เผยแพร่อัลกอริธึมนี้ได้ไม่นาน NIST ก็ประกาศตามมาว่า

อัลกอริธึมจำเป็นต้องได้รับการแก้ไขเพิ่มเติมเล็กน้อยเพื่อให้สามารถใช้งานได้อย่างเหมาะสม SHA สร้างไดเจสต์ที่มีขนาด 160 บิต

2.3.10.3 อัลกอริธึม SHA-1

SHA-1 เป็นอัลกอริธึมที่แก้ไขเพิ่มเติมเล็กน้อยจาก SHA การแก้ไขเพิ่มเติมนี้เป็นที่เชื่อกันว่าทำให้อัลกอริธึม SHA-1 มีความปลอดภัยที่สูงขึ้นSHA-1 สร้างไดเจสต์ที่มีขนาด 160 บิต

ไดเจสต์ เป็นเครื่องมือที่สำคัญที่สามารถใช้ในการตรวจสอบว่าไฟล์ในระบบที่ใช้งานมีการเปลี่ยนแปลงแก้ไขหรือไม่ (ไม่ว่าจะโดยเจตนาหรือไม่ก็ตาม) บางครั้งการเปลี่ยนแปลงแก้ไขอาจถูกกระทำโดยผู้ที่ไม่มีสิทธิ์ เช่น ผู้บุกรุก เป็นต้น วิธีการใช้ไดเจสต์เพื่อตรวจสอบไฟล์ในระบบคือให้เลือกใช้อัลกอริธึมหนึ่ง เช่น MD5 เพื่อสร้างไดเจสต์ของไฟล์ในระบบและเก็บไดเจสต์นั้นไว้อีกที่หนึ่งนอกระบบ ภายหลังจากระยะเวลาหนึ่งที่กำหนดไว้ เช่น 1 เดือน ก็มาคำนวณไดเจสต์ของไฟล์เดิมอีกครั้งหนึ่ง แล้วเปรียบเทียบไดเจสต์ใหม่นี้กับไดเจสต์ที่เก็บไว้นอกระบบว่าตรงกันหรือไม่ ถ้าตรงกัน ก็แสดงว่าไฟล์ในระบบยังเป็นปกติเช่นเดิม

ไดเจสต์ยังเป็นส่วนหนึ่งของการลงลายมือชื่ออิเล็กทรอนิกส์ กล่าวคือการลงลายมือชื่ออิเล็กทรอนิกส์ในปัจจุบัน จะใช้การลงลายมือชื่อกับไดเจสต์ของข้อความตั้งต้นแทนการลงลายมือชื่อกับข้อความตั้งต้นทั้งข้อความ

2.4 งานวิจัยที่เกี่ยวข้อง

ในงานสื่อสารข้อมูลบนระบบเครือข่าย Internet นั้น ความปลอดภัยของข้อมูลเป็นสิ่งจำเป็นอย่างยิ่ง ซึ่งระบบรักษาความปลอดภัยของข้อมูลมีด้วยกันหลายรูปแบบวิธี ยกตัวอย่างเช่น IPSec และ SSL [7] ได้ทำการเปรียบเทียบวัดประสิทธิภาพของระบบทั้งสองเพื่อหาความเหมาะสมในการเลือกใช้งาน และในงานวิจัยอีกงานได้ทำการทดลองสร้างเครื่องมือที่มีชื่อเรียกว่า SSLPerf [8] เพื่อใช้ในการวัดประสิทธิภาพระบบความปลอดภัยที่นำไปทดสอบกับอุปกรณ์การสื่อสารแบบไร้สาย หรือแม้แต่ระบบทางการเงินการธนาคาร ที่เรียกว่า SET [9] ที่ได้มีการเปรียบเทียบกับระบบของความปลอดภัย SSL

2.5 บทสรุป

การรักษาความมั่นคงปลอดภัยของระบบคอมพิวเตอร์ หรือ ระบบเครือข่ายคอมพิวเตอร์ เป็นสิ่งที่ควรตระหนักเป็นอย่างยิ่งในปัจจุบัน เพราะโลกในยุคปัจจุบันเป็นโลกแห่งข้อมูลข่าวสาร การเก็บรักษาข้อมูลให้ปลอดภัยจึงเป็นสิ่งสำคัญกับตัวบุคคลและองค์กร เพราะฉะนั้นการที่จะอนุญาตให้บุคคลใดบุคคลหนึ่งสามารถเข้าถึงข้อมูลจึงเป็นสิ่งที่ควรระมัดระวัง เพราะข้อมูลบางอย่างของบุคคลและองค์กรมีความสำคัญและไม่สามารถเปิดเผยต่อบุคคลภายนอกได้

การพิสูจน์ตัวตนจึงมีความสำคัญ เนื่องจากว่าการที่บุคคลใดบุคคลหนึ่งจะเข้าสู่ระบบได้จะต้องได้รับการยอมรับว่าได้รับอนุญาตจริง การตรวจสอบหลักฐานจึงเป็นขั้นตอนแรกก่อนอนุญาตให้เข้าสู่ระบบ การยืนยันตัวตนยิ่งมีความซับซ้อนมาก นั่นก็หมายถึงว่าความปลอดภัยของข้อมูลก็มีมากขึ้นด้วย

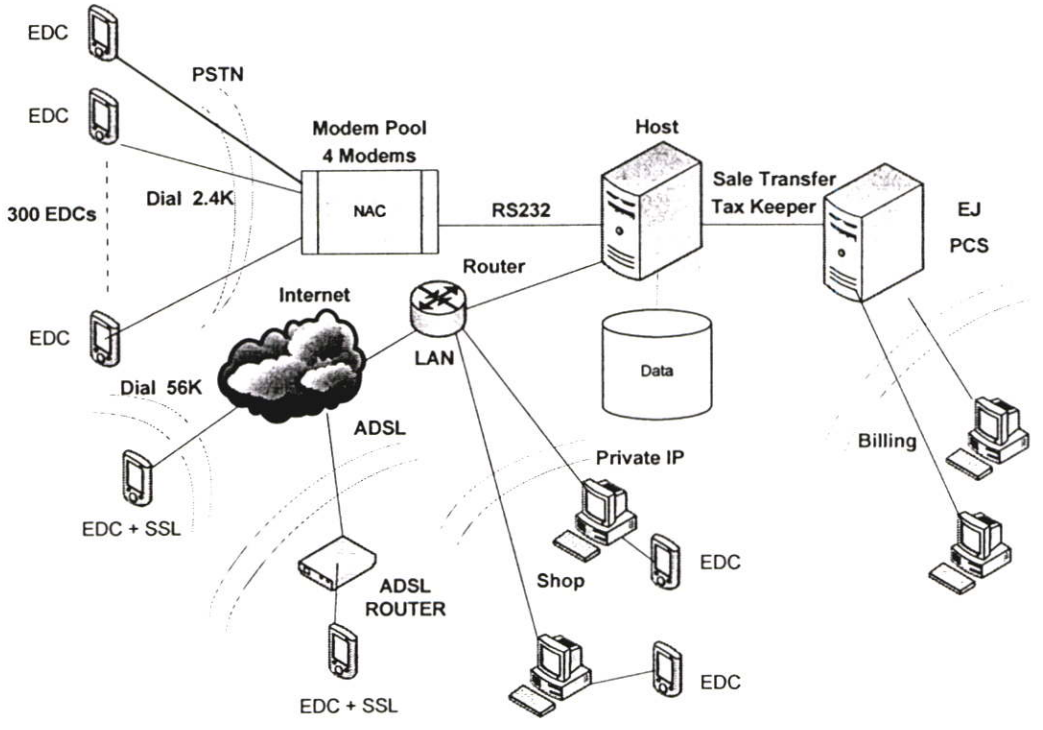
บทที่ 3

การออกแบบระบบ SSL Protocol บนอุปกรณ์ Electronic Data Capture

3.1 ระบบชำระเงิน

ระบบชำระเงินดังรูปที่ 3.1 เป็นระบบที่สามารถดำเนินการได้หลายรูปแบบ เริ่มจาก PSTN (Public Switched Telephone Network) ที่ได้รับการออกแบบให้สามารถรองรับอุปกรณ์ EDC จำนวนมากๆ (ขึ้นอยู่กับจำนวนตัวแทน) EDC แต่ละตัวจะทำการติดต่อกับเซิร์ฟเวอร์ผ่านทางพอร์ตโมเด็ม ความเร็ว 2,400 bps ซึ่งเป็นการทำงานแบบ Off Line กล่าวคือ เมื่อมีข้อมูลเก็บไว้ในตัวอุปกรณ์ EDC ตามจำนวนหรือตามเวลาที่ได้กำหนดไว้ EDC ก็จะทำการติดต่อไปยังเซิร์ฟเวอร์เองโดยอัตโนมัติเพื่อส่งข้อมูลให้กับระบบ

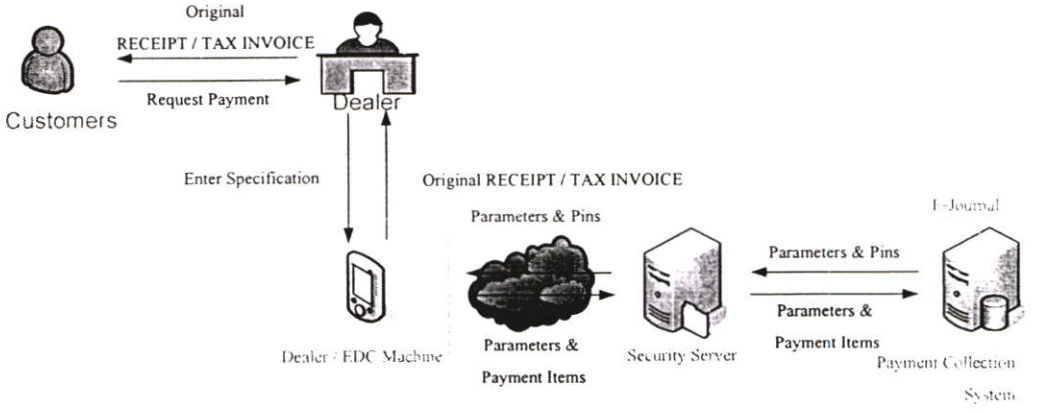
แนวทางถัดมา คือ EDC ทำหน้าที่เป็น Modem ความเร็ว 56k ทำการติดต่อกับระบบชำระเงิน ผ่านระบบเครือข่ายอินเตอร์เน็ต ซึ่งในส่วนนี้ก็ต้องคำนึงถึงความปลอดภัยของข้อมูล



รูปที่ 3.1 ผังระบบชำระเงิน

ส่วนแนวทางที่จะนำมาทำการวิจัย คือการที่ EDC ติดต่อกับระบบชำระเงิน ผ่านระบบเครือข่ายอินเตอร์เน็ต ซึ่งมีหลักการทำงานคล้ายกับแนวทางที่สอง แต่มีความแตกต่างตรงที่พอร์ตสื่อสารนั้นจะใช้พอร์ต Ethernet ที่สามารถติดต่ออินเตอร์เน็ตผ่านทาง ADSL Modem และในส่วนนี้ก็ต้องคำนึงถึงความปลอดภัยของข้อมูลด้วยเช่นกัน ดังนั้นในการวิจัยก็จะทำการศึกษาในเรื่องความปลอดภัยของข้อมูลเป็นหลักสำคัญ

ในการศึกษาขั้นตอนการชำระโดยผ่าน ADSL ดำเนินการโดยลูกค้าที่จะชำระค่าสาธารณูปโภคไปติดต่อตัวแทนเพื่อดำเนินการโดยนำใบแจ้งหนี้ที่ได้รับไปยื่น จากนั้นตัวแทนทำการป้อนข้อมูลของลูกค้าผ่านอุปกรณ์ EDC ที่ต่อกับ ADSL Modem เพื่อทำการพิมพ์ใบเสร็จรับเงินให้กับลูกค้าไว้เป็นหลักฐานการส่งข้อมูลผ่านระบบเครือข่ายอินเตอร์เน็ตในขั้นตอนนี้ข้อมูลจะต้องได้รับการป้องกันจากการที่จะถูกลักลอบดักข้อมูล ข้อมูลที่ผ่านขบวนการป้องกันโดยการเข้ารหัสถูกส่งไปยังเซิร์ฟเวอร์เพื่อดำเนินการตัดยอดคงค้าง ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 ผังการชำระผ่านระบบ ADSL

ในระบบเครือข่ายแบบเปิดหรืออินเตอร์เน็ต การพิสูจน์ตัวตนถือได้ว่าเป็นกระบวนการเริ่มต้นและมีความสำคัญที่สุดในการปกป้องเครือข่ายให้ปลอดภัย โปรโตคอลในการพิสูจน์ตัวตนคือโปรโตคอลการสื่อสารที่มีกระบวนการพิสูจน์ตัวตนรวมอยู่ในชุดโปรโตคอล

3.2 โปรโตคอลการพิสูจน์ตัวตนที่ใช้ในงานวิจัยนี้

งานวิจัยในครั้งนี้ใช้ โปรโตคอลการพิสูจน์ตัวตนแบบ Secure Socket Layer (SSL) Secure Sockets Layer (SSL) [10] , [11] ซึ่งเริ่มพัฒนาโดย Netscape Communications เพื่อใช้ในโปรโตคอลระดับแอพพลิเคชั่นคือ Hypertext Transfer Protocol (HTTP) และเป็นการสื่อสารผ่าน

เว็บให้ปลอดภัย พัฒนาในช่วงต้นของยุคการค้าอิเล็กทรอนิคส์กำลังได้รับความนิยมในโลกอินเตอร์เน็ต

SSL ทำให้เกิดการสื่อสารอย่างปลอดภัยระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ โดยการอนุญาตให้มีกระบวนการพิสูจน์ตัวตนร่วมกับการใช้งานลายเซ็นดิจิตอลสำหรับการรักษาความถูกต้องของข้อมูลและการเข้ารหัสข้อมูลเพื่อป้องกันความเป็นส่วนตัวระหว่างการสื่อสารข้อมูล

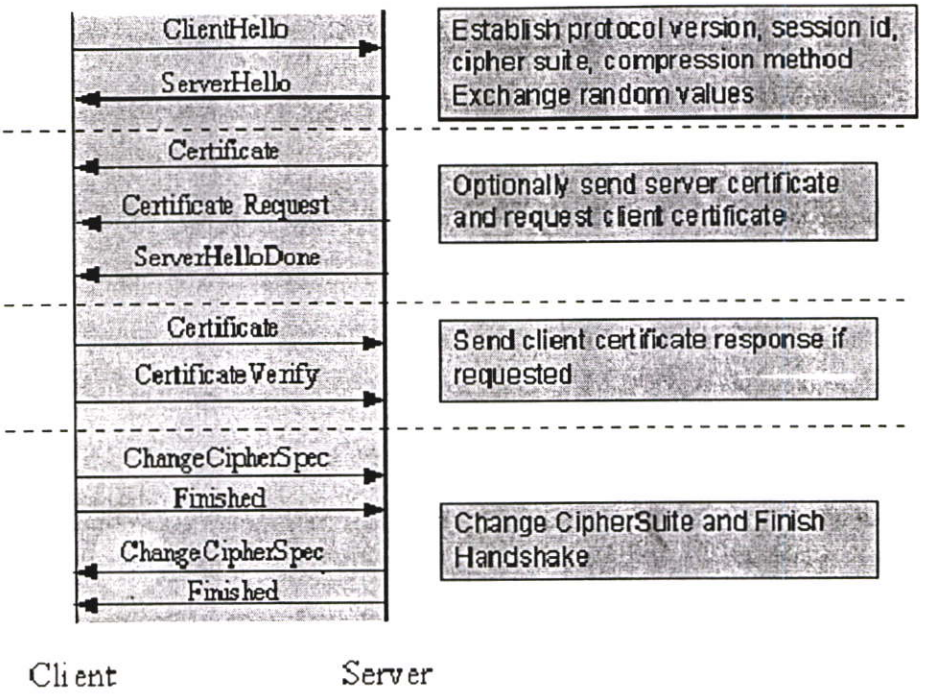
โปรโตคอล SSL อนุญาตให้สามารถเลือกวิธีการในการเข้ารหัส วิธีสร้างไดเจสต์ และลายเซ็นดิจิตอล ได้อย่างอิสระก่อนการสื่อสารจะเริ่มต้นขึ้น ตามความต้องการของทั้งเว็บเซิร์ฟเวอร์และบราวเซอร์ ทั้งนี้เพื่อเพิ่มความยืดหยุ่นในการใช้งาน เปิดโอกาสให้ทดลองใช้วิธีการในการเข้ารหัสวิธีใหม่ รวมถึงลดปัญหาการส่งออกวิธีการเข้ารหัสไปประเทศที่ไม่อนุญาต

Netscape เริ่มพัฒนา SSL เวอร์ชันแรกคือเวอร์ชัน 2.0 และเวอร์ชันถัดมาเป็น 3.0 ซึ่งสนับสนุนความสามารถด้านความปลอดภัยมากขึ้น และเป็นเวอร์ชันสุดท้ายก่อนที่จะเป็นมาตรฐานกลางของโปรโตคอลบนอินเตอร์เน็ต โดยเปลี่ยนชื่อเป็น Transport Layer Security หรือ TLS ซึ่งดูแลมาตรฐานโดย Internet Engineering Task Force (IETF) อธิบายเวอร์ชันของ SSL และผู้พัฒนาได้ตามกระบวนการในการเริ่มต้นการสื่อสารผ่านชั้น SSL แบ่งเป็น 4 ขั้นตอนคือ

- ประกาศชุดวิธีการเข้ารหัส ไดเจสต์ และลายเซ็นดิจิตอลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์
- การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์
- การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์ ถ้าจำเป็น
- ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างไดเจสต์ และการใช้ลายเซ็นดิจิตอลตามรูปที่ 3.3 [4]

ตารางที่ 3.1 ตารางเปรียบเทียบเวอร์ชันของ SSL

เวอร์ชัน	ผู้พัฒนา	จุดเด่น	บราวเซอร์ที่สนับสนุน
SSL v2.0	Netscape Corp. [SSL2]	โปรโตคอล SSL รุ่นแรกที่พัฒนาบนบราวเซอร์	• NS Navigator 1.x/2.x • MS IE 3.x • Lynx/2.8 + OpenSSL
SSL v3.0	Netscape Corp. เป็น Internet Drafted รุ่นก่อนเป็นมาตรฐานกลาง [SSL3]	ปรับปรุงใหม่เพิ่มความปลอดภัยมากขึ้น สนับสนุนการใช้ non-RSA ciphers ในการเข้ารหัส และห่วงโซ่ Certificate	• NS Navigator 2.x/3.x/4.x • MS IE 3.x/4.x • Lynx/2.8 + OpenSSL



รูปที่ 3.3 กระบวนการเริ่มต้นการติดต่อสื่อสารของโปรโตคอล SSL

ขั้นตอนที่ 1 : ประกาศชุดวิธีการเข้ารหัส ไดเจสต์ และลายเซ็นดิจิตอลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์

ไคลเอ็นต์และเซิร์ฟเวอร์ส่งข้อความเริ่มต้นการสื่อสาร (Hello message) ซึ่งประกอบไปด้วยเวอร์ชันของโปรโตคอลที่ใช้ วิธีการเข้ารหัสที่เว็บเซิร์ฟเวอร์และไคลเอ็นต์สนับสนุน หมายเลขระบุการสื่อสาร (Session identifier) รวมถึงวิธีการบีบอัดข้อมูลในการสื่อสารที่สนับสนุน

หมายเลขระบุการสื่อสารที่เกิดขึ้น ใช้สำหรับตรวจสอบการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ถ้ามีการเชื่อมต่อก่อนหน้านี้เกิดขึ้น แสดงว่าได้มีการตกลงวิธีการสื่อสารแล้ว สามารถเริ่มต้นส่งข้อมูลได้ทันที เป็นการลดเวลาติดต่อสื่อสารลง

ขั้นตอนที่ 2 : การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์

ถัดมาเว็บเซิร์ฟเวอร์ทำการส่ง Certificate หรือใบยืนยันความมีตัวตนของเซิร์ฟเวอร์ ไคลเอ็นต์จะทำการตรวจสอบ Certificate กับผู้ให้บริการ Certificate Authority ที่ได้ตั้งค่าไว้ เพื่อยืนยันความถูกต้องของ Certificate ของเซิร์ฟเวอร์

ขั้นตอนที่ 3 : การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์

ถ้าจำเป็นเซิร์ฟเวอร์สามารถร้องขอ Certificate จากไคลเอ็นต์เพื่อตรวจสอบความถูกต้องของ Client ด้วยก็ได้ ใช้ในกรณีที่มีการจำกัดการใช้งานเฉพาะไคลเอ็นต์ที่ต้องการเท่านั้น ซึ่ง SSL

สนับสนุนการตรวจสอบได้จากทั้งเซิร์ฟเวอร์ และไคลเอ็นต์ขึ้นอยู่กับการเลือกใช้งานในขณะติดต่อสื่อสารที่เกิดขึ้นนั้น

ขั้นตอนที่ 4 : ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างไดเจสต์ และการใช้ลายเซ็นดิจิตอล

ขั้นตอนการตรวจสอบ Certificate ที่เซิร์ฟเวอร์ร้องขอจากไคลเอ็นต์จะมีหรือไม่มีก็ได้ขึ้นอยู่กับการตั้งค่าบนเซิร์ฟเวอร์ หลังจากขั้นตอนการตรวจสอบเสร็จสิ้น เซิร์ฟเวอร์และไคลเอ็นต์จะตกลงการใช้งานวิธีการเข้ารหัสระหว่างกันโดยใช้ค่าที่ได้จากการประกาศในขั้นตอนแรก

วิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส (Key exchange method) คือการกำหนดกลไกการแลกเปลี่ยนกุญแจที่ใช้ในการเข้ารหัสระหว่างการสื่อสาร โดยทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะใช้กุญแจนี้ในการเข้ารหัสและถอดรหัสข้อมูล ใน SSL เวอร์ชัน 2.0 จะสนับสนุนวิธีการแลกเปลี่ยนกุญแจแบบ RSA ส่วน SSL เวอร์ชัน 3.0 ขึ้นไปจะสนับสนุนวิธีการอื่นๆ เพิ่มเติมเช่นการใช้ RSA ร่วมกับการใช้ Certificate หรือ Diffie-Hellman เป็นต้น

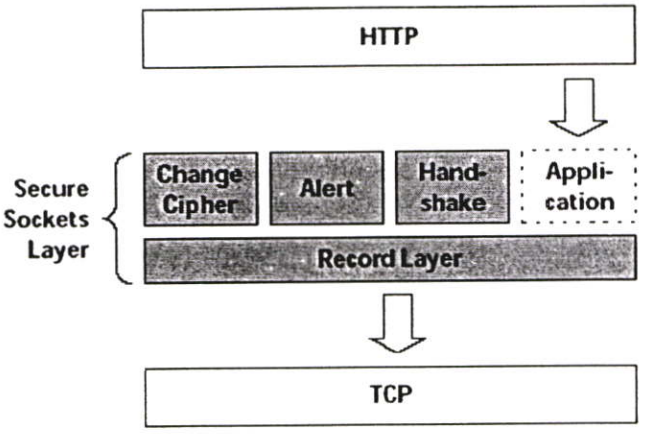
วิธีการเข้ารหัสในปัจจุบันแบ่งเป็นสองวิธีคือ การใช้กุญแจเดียวกันในการเข้ารหัสและถอดรหัส อาจเรียกกุญแจนี้ว่า Public key หรือ Secret key ส่วนอีกวิธีการคือ การใช้กุญแจคนละตัวในการเข้ารหัสและถอดรหัส ประกอบไปด้วยกุญแจสาธารณะและกุญแจส่วนตัวซึ่งเป็นคู่กันเสมอ เข้ารหัสด้วยกุญแจใด จะต้องถอดรหัสด้วยกุญแจที่คู่กันและตรงกันข้ามเท่านั้น มักใช้วิธีการเข้ารหัสด้วยกุญแจคนละตัวมาใช้ในการเข้ารหัส Public key และส่งไปให้ฝั่งตรงข้ามก่อนการสื่อสารจะเกิดขึ้นรวมเรียกว่าวิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส

SSL ใช้วิธีการเข้ารหัสด้วยกุญแจสมมาตร หรือกุญแจเดียวในการเข้ารหัสและถอดรหัสตามที่กล่าวข้างต้น วิธีการเข้ารหัสคือ การเข้ารหัสด้วย DES และ 3DES (Data Encryption Standard), ส่วน RC2 และ RC4 เป็นวิธีการเข้ารหัสของ RSA สำหรับความยาวของการเข้ารหัสที่ใช้คือ 40 บิต, 96 บิต และ 128 บิต

การสร้าง Message Authentication Code (MAC) เพื่อใช้สำหรับการยืนยันความถูกต้องของข้อมูลระหว่างการสื่อสารและป้องกันการปลอมข้อมูล ส่วนฟังก์ชันสร้างไดเจสต์ที่ SSL สนับสนุนและเลือกใช้ได้ในปัจจุบันคือ MD5 ขนาด 128 บิต และ SHA-1 (Secure Hash Algorithm) ขนาด 160 บิต

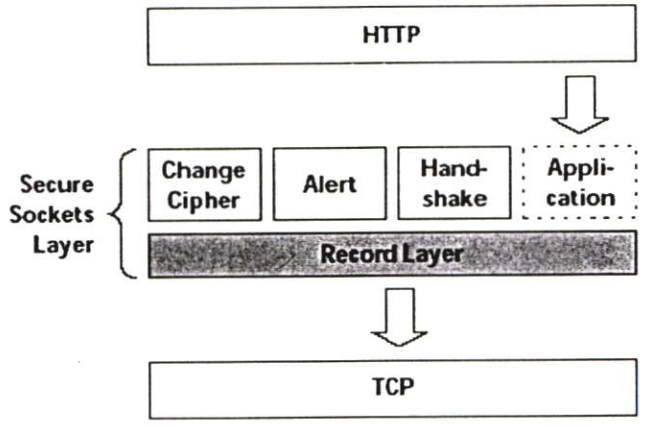
ซึ่งจะได้วิธีการที่ทั้งสองฝ่ายสนับสนุนและเหมาะสมซึ่งเป็นขั้นตอนสุดท้ายก่อนการสื่อสารที่มีการเข้ารหัสจะเริ่มต้นขึ้น

ในส่วนนี้กล่าวถึงการนำระบบความปลอดภัยเพิ่มเข้าไปสู่อุปกรณ์ EDC โดยวิธีการสร้างความปลอดภัยด้วย SSL Protocol ซึ่งประกอบไปด้วยส่วนต่างๆ ดังแสดงในรูปที่ 3.3 ที่มี Protocol ที่แตกต่างกัน 4 Protocol [12] เริ่มตั้งแต่ Change Cipher Spec protocol, Alert protocol, Handshake protocol และ Application Data Protocol



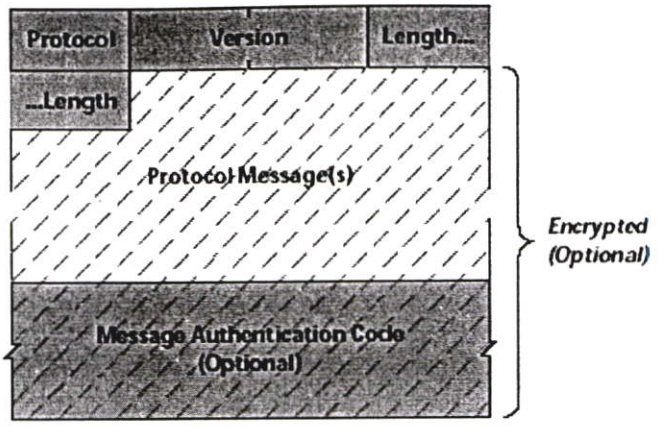
รูปที่ 3.4 ส่วนประกอบของ SSL Protocol

Record Layerใน SSL Protocol จะอาศัย Record Layer Protocol ในการเอนแคปซูเลท (Encapsulated) ข้อมูลทั้งหมด เพื่อใช้ในการแจ้งเตือนข้อผิดพลาด (Alert protocol), ใช้ในการแลกเปลี่ยนคีย์ (Change Cipher Spec protocol), การสร้างการสื่อสาร (Handshake protocol) และแอพพลิเคชัน (Application Data Protocol)



รูปที่ 3.5 แสดงรูปแบบของ Record Layer

การกำหนดโครงสร้างของ Record Layer ได้แสดงไว้ในรูปที่ 3.6 ส่วนในตารางที่ 3.2 นั้นเป็นส่วนอธิบายความหมายของโครงสร้าง



รูปที่ 3.6 การเอนแคปซูเลทข้อมูลของ Record Layer บน SSL

ตารางที่ 3.2 SSL Record Layer Fields

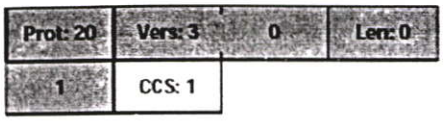
Field	Size	Usage
Protocol	1 byte	แสดงค่าของโปรโตคอลที่อยู่ภายใน Record Layer
Version	2 byte	แสดงเวอร์ชันของ SSL
Length	2 byte	แสดงขนาดของโปรโตคอลที่ที่อยู่ภายใน Record Layer
Protocol Massage	n byte	แสดงข้อมูลของโปรโตคอลที่ที่อยู่ภายใน Record Layer ประกอบไปด้วย Protocol Massage, Massage Authentication Code

จากตารางที่ 3.2 นั้นในส่วนของโปรโตคอลที่ต้องมีการกำหนดค่าของโปรโตคอลที่อยู่ภายใน Record Layer ซึ่งจะได้แสดงความหมายของคาเหน่านั้นในตารางที่ 3.3

ตารางที่ 3.3 แสดงค่าของชนิด Record Layer Protocol

Type Value	Protocol
20	Change Cipher Spec protocol
21	Alert protocol
22	Handshake protocol
23	Application Data protocol

Change Cipher Spec Protocol ในส่วนนี้เป็นการทำงานที่ไม่ค่อยซับซ้อนมากหนักเพราะมีเพียง 1 ข้อความเท่านั้น ที่จะใช้ในการแจ้งเปลี่ยนสถานะของการเข้ารหัสข้อมูล



รูปที่ 3.7 ชนิดของข้อมูล Change Cipher Spec

Alert Protocol ส่วนในหัวข้อนี้กล่าวถึง Alert Protocol ที่ใช้สำหรับการแจ้งเตือนข้อผิดพลาดหรือการแสดงเงื่อนไขของเหตุการณ์อื่นๆ ที่เกิดขึ้นในระหว่างการติดต่อสื่อสารกับอีกด้าน ซึ่งในส่วนนี้เป็นส่วนที่มีความสำคัญ

ตัวอย่างของคำสั่งที่ใช้ในการแจงผลข้อผิดพลาดที่เกิดขึ้น ในการสื่อสารของ SSL

```
if (!(ssl = SSL_new(ctx)))
        int_error("Error creating an SSL context");
```

Cipher Suites ของ SSL เวอร์ชัน 3.0 มีการกำหนดสูตรในการเข้ารหัส (Cipher Suites) ที่แตกต่างกัน ซึ่งมีลักษณะอัลกอริธึมการเข้ารหัสและพารามิเตอร์ประกอบ ในตารางที่ 3.4 แสดงรายชื่อของสูตรการเข้ารหัส แสดงลักษณะของ Key Exchange รูปแบบการเข้ารหัส และอัลกอริธึมแฮช (Hash)

ตารางที่ 3.4 Cipher Suite Algorithm

Key Exchange	Encryption	Hash	Exportable
SSL_NULL_	WITH_NULL_	NULL	•
SSL_RSA_	WITH_NULL_	MD5	•
SSL_RSA_	WITH_NULL_	SHA	•
SSL_RSA_EXPORT_	WITH_RC4_40_	MD5	•
SSL_RSA_	WITH_RC4_128_	MD5	
SSL_RSA_	WITH_RC4_128_	SHA	
SSL_RSA_EXPORT_	WITH_RC2_CBC_40_	MD5	•
SSL_RSA_	WITH_IDEA_CBC_	SHA	
SSL_RSA_EXPORT_	WITH_DES40_CBC_	SHA	•
SSL_RSA_	WITH_DES_CBC_	SHA	
SSL_RSA_	WITH_3DES_EDE_CBC_	SHA	
SSL_DH_DSS_EXPORT_	WITH_DES40_CBC_	SHA	•
SSL_DH_DSS_	WITH_DES_CBC_	SHA	
SSL_DH_RSA_	WITH_DES_CBC_	SHA	
SSL_DH_DSS_	WITH_3DES_EDE_CBC_	SHA	
SSL_DHE_DSS_	WITH_DES_CBC_	SHA	

ตารางที่ 3.4 (ต่อ)

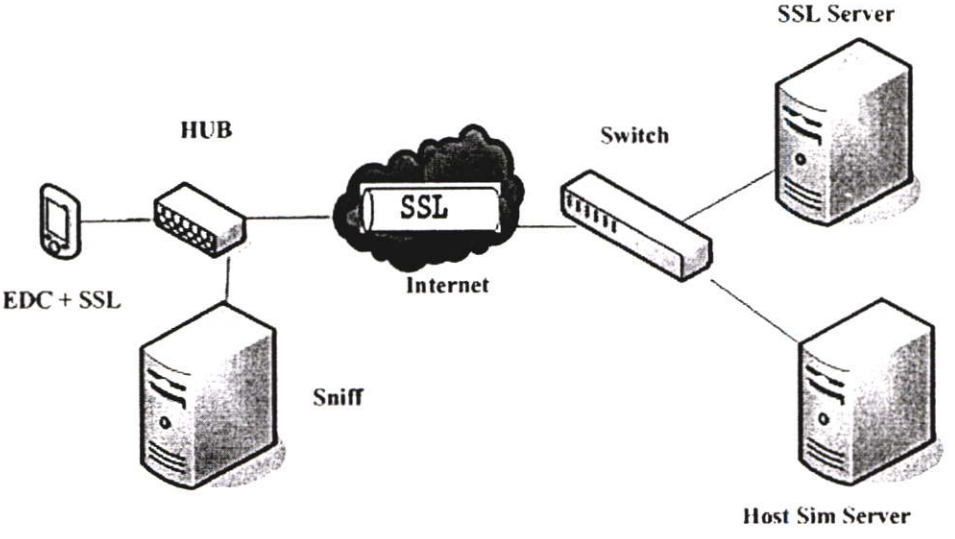
SSL_DHE_DSS_	WITH_3DES_EDE_CBC_	SHA	
SSL_DHE_RSA_EXPORT_	WITH_DES40_CBC_	SHA	•
SSL_DHE_RSA_	WITH_DES_CBC_	SHA	
SSL_DHE_RSA_	WITH_3DES_EDE_CBC_	SHA	
SSL_DH_anon_EXPORT_	WITH_RC4_40_	MD5	•
SSL_DH_anon_	WITH_RC4_128_	MD5	
SSL_DH_anon_EXPORT_	WITH_DES40_CBC_	SHA	
SSL_DH_anon_	WITH_DES_CBC_	SHA	
SSL_DH_anon_	WITH_3DES_EDE_CBC_	SHA	
SSL_FORTEZZA_DMS_	WITH_NULL_	SHA	
SSL_FORTEZZA_DMS_	WITH_FORTEZZA_CBC_	SHA	
SSL_FORTEZZA_DMS_	WITH_RC4_128_	SHA	

การแสดงลักษณะของ Cipher Suite ด้วยคำสั่ง

#openssl ciphers –v –ssl3

3.3 รูปแบบการทดลอง

ในการทดลองได้ดำเนินการทำการเชื่อมต่อระบบเครือข่ายเพื่อใช้ในขั้นการทดลองเก็บข้อมูลที่ทำการส่งข้อมูลสื่อสารระหว่างอุปกรณ์ EDC กับเซิร์ฟเวอร์ แสดงดังรูปที่ 3.8



รูปที่ 3.8 ระบบที่ใช้ในการทดลอง

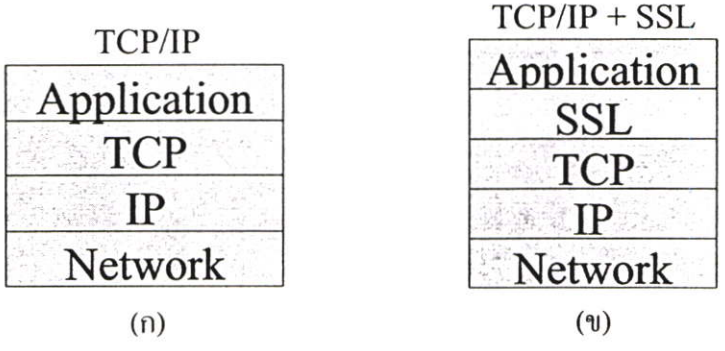
3.3.1 คุณสมบัติเครื่องมือที่ใช้ในการทดลอง

1. PC : SSL Server : ทำหน้าที่เป็น พรอกซี (Proxy) ใช้ในการติดต่อกับ EDC ในกรณีที่มีการติดต่อกับระบบด้วย SSL Protocol จากภายนอก ในรูปแบบที่มีการเข้ารหัสข้อมูล หลังจากนั้นทำการถอดรหัสข้อมูลส่งให้กับเครื่อง Host Sim Server หรือในทางกลับกันคือข้อมูลจากเครื่อง Host Sim Server ที่มีลักษณะเป็นPlaintext ส่งให้กับ SSL Server เพื่อทำการเข้ารหัสแล้วส่งออกไปยัง EDC
   - ติดตั้งซอฟแวร์ Stunnel [13] , [14] เวอร์ชัน 4.20 ที่มีลักษณะในทำงานแบบ SSL
   - Pentium 4, 1.8 GHz, RAM 512MB
   - NIC 100Mbps
2. PC : Host Sim Sever : ทำหน้าที่ในการรับ - ส่งข้อมูลทางการเงินที่มีลักษณะเป็น Plaintext ตามมาตรฐาน ISO-8583 หลังจากมีการถอดรหัสข้อมูลโดย SSL Server
   - Pentium 4, 1.8 GHz, RAM 512MB
   - NIC 100Mbps
3. PC : Sniffer : เป็นเครื่องที่ใช้ในการตรวจจับแพ็กเก็ตที่อยู่ในระบบ และแสดงคุณลักษณะของแพ็กเกต
   - โดยการติดตั้งซอฟแวร์ Ethereal เวอร์ชัน 0.99.0
   - Pentium 4, 1.8 GHz, RAM 512MB
4. EDC
   - Large memory with multiple applications oriented design
   - Graphic LCD and keyboard with backlight ensures clear & easy operation
   - Powerful 32-bit ARM processor
   - 8MB Flash RAM and 1MB SRAM
   - 128x64 dots graphics LCD and keyboard incorporate backlight
   - 18 keys with printing protected by long lasting transparent epoxy
   - 2" fast and silent thermal printer
   - Magnetic stripe card reader and smart card reader
   - TCP/IP interface 10 Mbps and high speed modem (up to 56K bps)
   - 2400bps modem

รูปที่ 3.9 แสดงอุปกรณ์ EDC ที่ใช้ในการทดลอง

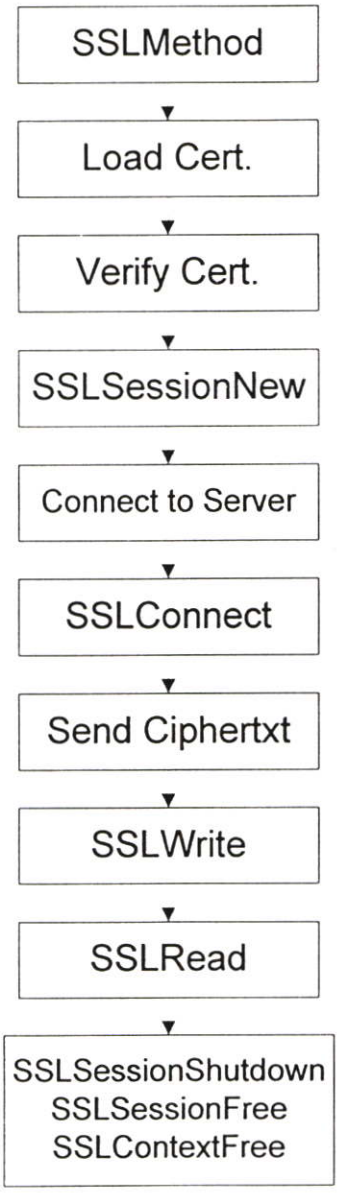
3.4 การดำเนินงานทดลอง

ในขบวนการทดลองได้ดำเนินการส่งข้อมูลที่มีลักษณะเป็น Plaintext ดังแสดงในรูปที่ 3.10 (ก) และ Ciphertext ดังแสดงในรูปที่ 3.10 (ข)

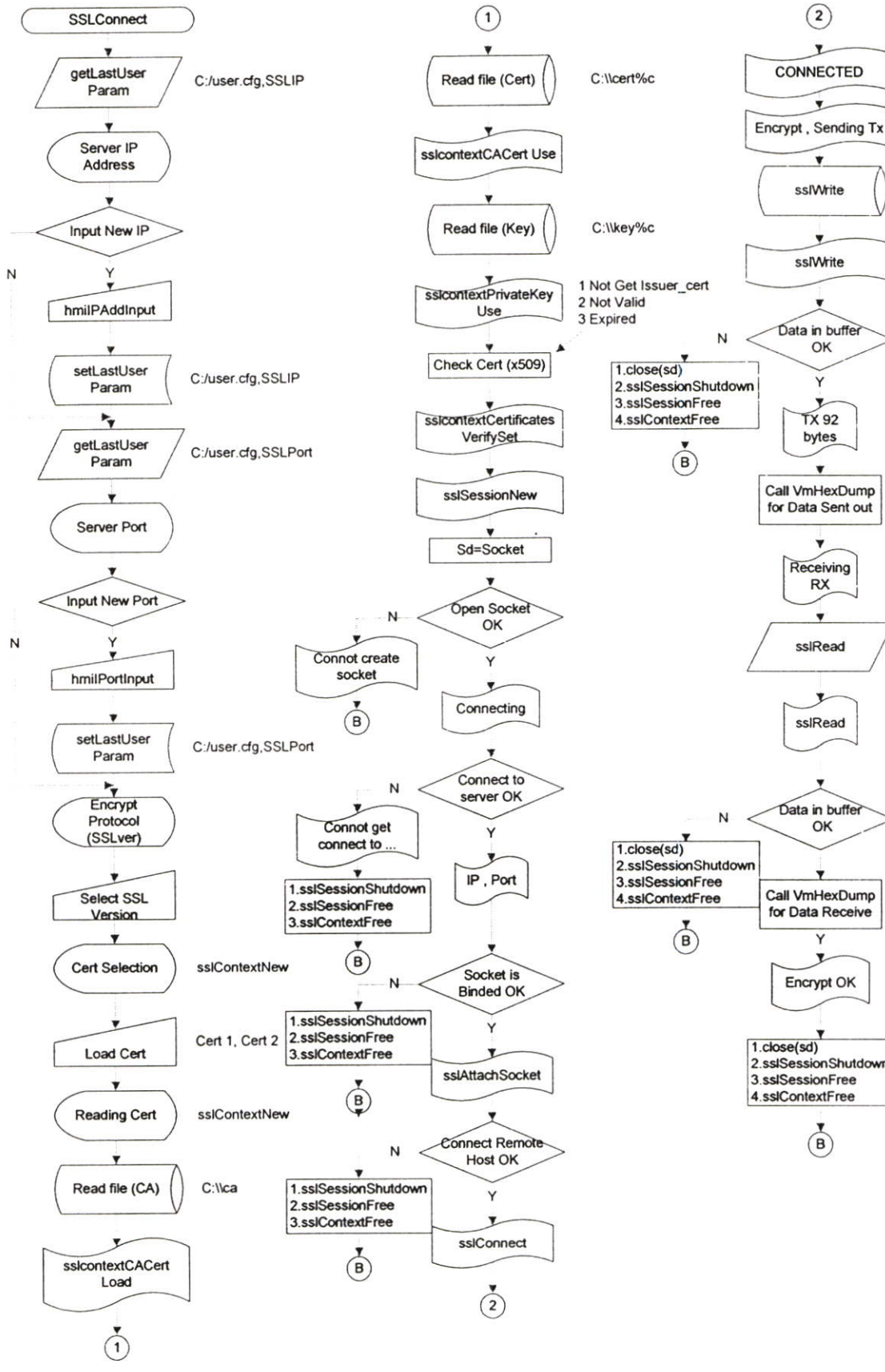


รูปที่ 3.10 แสดงลำดับขั้นในการทดลอง TCP/IP + SSL

อธิบายลักษณะการทำงานของระบบความปลอดภัยบนอุปกรณ์ EDC ที่มีระบบรักษาความปลอดภัยของข้อมูล เริ่มจากระบบทำการเลือกเวอร์ชันของ SSL ด้วย SSLMethod จากนั้นโหลด Certificate เก็บลงในหน่วยความจำ และทำการตรวจสอบความถูกต้องสมบูรณ์ ของ Certificate ที่โหลด หลังจากนั้นทำการสร้างกระบวนการเตรียมข้อมูลบนระบบที่มีการรักษาความปลอดภัยด้วย SSLSessionNew หลังจากนั้นสถาปนาการเชื่อมต่อไปยังเซิร์ฟเวอร์ ด้วย SSLConnect เมื่อสถาปนาการเชื่อมต่อได้แล้วดำเนินการส่งข้อมูลที่มีระบบรักษาความปลอดภัย ด้วย SSLWrite และรับข้อมูลกลับมาจากเซิร์ฟเวอร์ด้วย SSLRead หลังจากนั้นทำการเคลียร์ค่าต่างๆ ให้ระบบว่าง และทำการยุติการสื่อสาร ดังแสดงในรูปที่ 3.11



รูปที่ 3.11 แสดงลำดับขั้นในการทดลอง

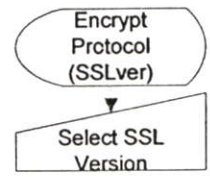


รูปที่ 3.12 แสดง Flowchart ของโปรแกรมการทดลอง

3.4.1 การทำงานของโปรแกรมโมดุลต่างๆ ของระบบ SSL Protocol

ในการทำงานของโปรแกรมโมดุลต่างๆ ของระบบ SSL Protocol บนอุปกรณ์ Electronic Data Capture ซึ่งประกอบด้วยแกรมโมดุลต่างๆ ดังต่อไปนี้

- โปรแกรมในการเลือก Version ของ SSL Protocol

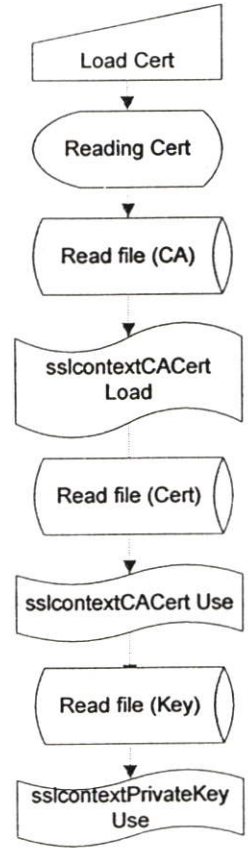


รูปที่ 3.13 แสดง Flowchart ของโปรแกรมในการเลือก Version ของ SSL Protocol

```
SSL_METHOD* method = SSLv3_client_method();
SSL_CTX*    ctx = SSL_CTX_new(method);
```

จากรูปที่ 3.13 แสดง Flowchart ของโปรแกรมที่ใช้เลือก Version ของ SSL Protocol เช่น การใช้ SSLv2 หรือ SSLv3

- โปรแกรมในการเรียกใช้ Certificate

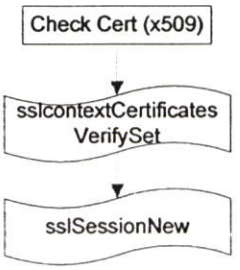


รูปที่ 3.14 แสดง Flowchart ของโปรแกรมในการเรียกใช้ Certificate

```
// client
const char* const cert = "../certs/client-cert.pem";
const char* const key  = "../certs/client-key.pem";
const char* const certSuite = "../../certs/client-cert.pem";
const char* const keySuite  = "../../certs/client-key.pem";
const char* const certDebug = "../../../certs/client-cert.pem";
const char* const keyDebug  = "../../../certs/client-key.pem";
```

จากรูปที่ 3.14 แสดง Flowchart ของโปรแกรมในขณะเรียกใช้ Certificate เพื่อใช้ในการยืนยันการมีตัวตนสำหรับการติดต่อกับผู้อื่น

- โปรแกรม Check Certificate



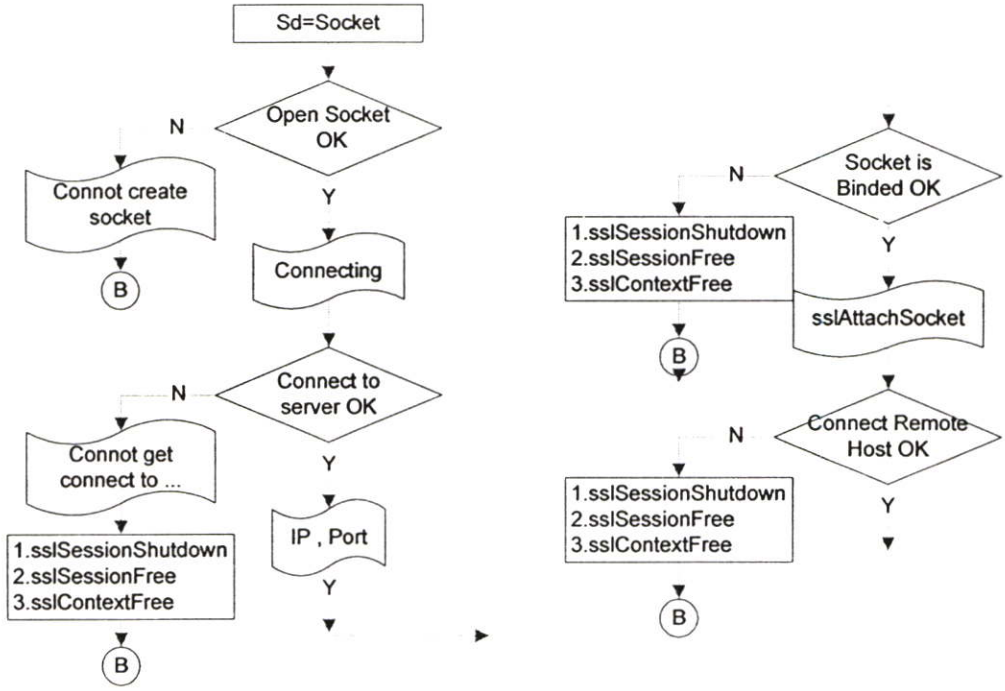
รูปที่ 3.15 แสดง Flowchart ของโปรแกรม Check Certificate

```
// client
inline void set_certs(SSL_CTX* ctx)
{
   store_ca(ctx);
   SSL_CTX_set_default_passwd_cb(ctx, PasswordCallBack);
   // To allow testing from serveral dirs
   if (SSL_CTX_use_certificate_file(ctx, cert, SSL_FILETYPE_PEM)
      != SSL_SUCCESS)
      if (SSL_CTX_use_certificate_file(ctx, certSuite, SSL_FILETYPE_PEM)
         != SSL_SUCCESS)
         if (SSL_CTX_use_certificate_file(ctx, certDebug, SSL_FILETYPE_PEM)
            != SSL_SUCCESS)
            err_sys("failed to use certificate: certs/client-cert.pem");
```

```
// To allow testing from several dirs
if (SSL_CTX_use_PrivateKey_file(ctx, key, SSL_FILETYPE_PEM)
    != SSL_SUCCESS)
    if (SSL_CTX_use_PrivateKey_file(ctx, keySuite, SSL_FILETYPE_PEM)
        != SSL_SUCCESS)
        if (SSL_CTX_use_PrivateKey_file(ctx,keyDebug,SSL_FILETYPE_PEM)
            != SSL_SUCCESS)
            err_sys("failed to use key file: certs/client-key.pem");
}
```

จากรูปที่ 3.15 แสดง Flowchart ของโปรแกรม Check Certificate เพื่อใช้ในการตรวจสอบสถานะของ Certificate ว่ายังสามารถใช้งานได้หรือไม่ หลังจากนั้นจึงทำการเปิดขบวนการของ SSL เพื่อเตรียมความพร้อมที่จะทำการติดต่อกับ Server

- โปรแกรม SSL Connect

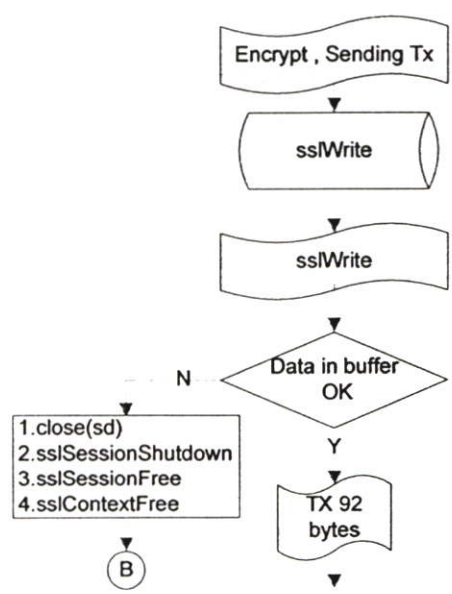


รูปที่ 3.16 แสดง Flowchart ของโปรแกรม SSL Connect

```
#ifdef TEST_RESUME
  tcp_connect(sockfd);
  SSL_set_fd(sslResume, sockfd);
  SSL_set_session(sslResume, session);
  if (SSL_connect(sslResume) != SSL_SUCCESS)
    ClientError(ctx, sslResume, sockfd, "SSL_resume failed");
  showPeer(sslResume);
  if (SSL_write(sslResume, msg, sizeof(msg)) != sizeof(msg))
    ClientError(ctx, sslResume, sockfd, "SSL_write failed");
  input = SSL_read(sslResume, reply, sizeof(reply));
  if (input > 0) {
    reply[input] = 0;
    printf("Server response: %s\n", reply);
  }
```

จากรูปที่ 3.16 แสดง Flowchart ของโปรแกรม SSL Connect เพื่อทำการสถาปนาการเชื่อมต่อ โดยทำการตรวจสอบ IP Port และ Socket

- โปรแกรม SSL Write

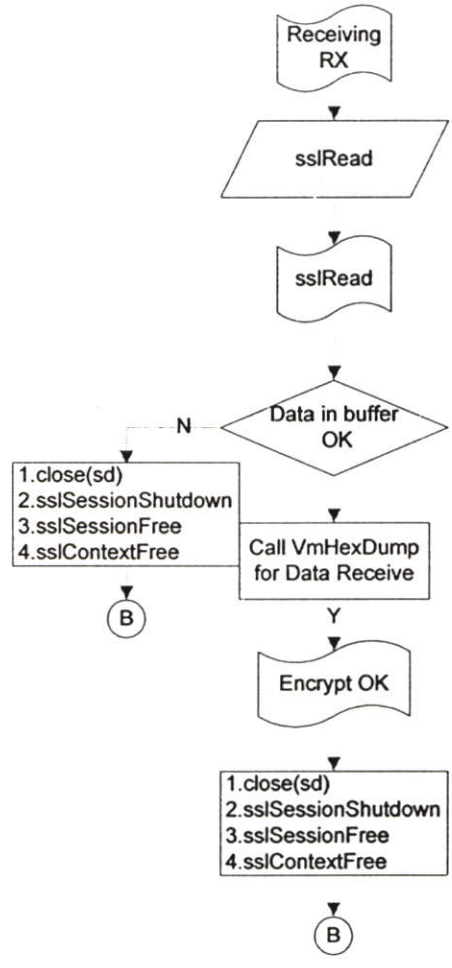


รูปที่ 3.17 แสดง Flowchart ของโปรแกรม SSL Write

```
if (SSL_write(sslResume, msg, sizeof(msg)) != sizeof(msg))
   ClientError(ctx, sslResume, sockfd, "SSL_write failed");
```

จากรูปที่ 3.17 แสดง Flowchart ของโปรแกรม SSL Write เพื่อทำการส่งข้อมูลที่ทำการเข้ารหัสแล้วไปยัง Server

- โปรแกรม SSL Read



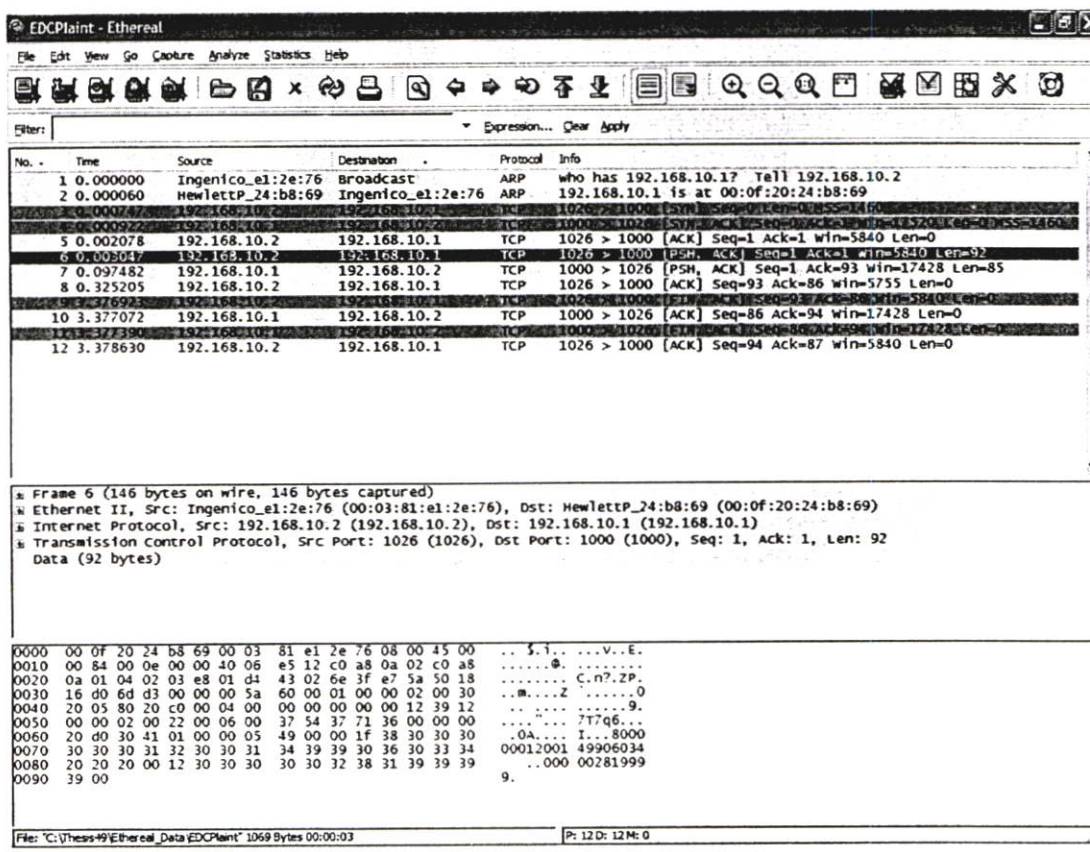
รูปที่ 3.18 แสดง Flowchart ของโปรแกรม SSL Read

```
input = SSL_read(sslResume, reply, sizeof(reply));
if (input > 0) {
   reply[input] = 0;
   printf("Server response: %s\n", reply);
}
```

จากรูปที่ 3.18 แสดง Flowchart ของโปรแกรม SSL Read เพื่อทำการรับข้อมูลที่ทำการเข้ารหัสแล้วจาก Server

3.4.2 ผลของการส่งข้อมูลที่มีลักษณะเป็น Plaintext

ในขั้นตอนการเก็บข้อมูลเริ่มจากทำการส่งข้อมูลที่เป็น Plaintext ที่มีขนาด 92 ไบต์ ซึ่งเป็นข้อมูลที่ใช้ในการรับ-ส่งข้อมูลในระบบทางการเงิน จากอุปกรณ์ EDC ไปยังเครื่องเซิร์ฟเวอร์ และทำการลักลอบดักข้อมูล โดยเครื่องคอมพิวเตอร์ Sniffer และแสดงคุณลักษณะของข้อมูลด้วยโปรแกรม Ethereal ดังแสดงในรูปที่ 3.19 สามารถมองเห็นข้อมูลที่ส่งจาก EDC ไปยังเครื่องเซิร์ฟเวอร์ซึ่งเป็นข้อมูลตามมาตรฐาน ISO-8583

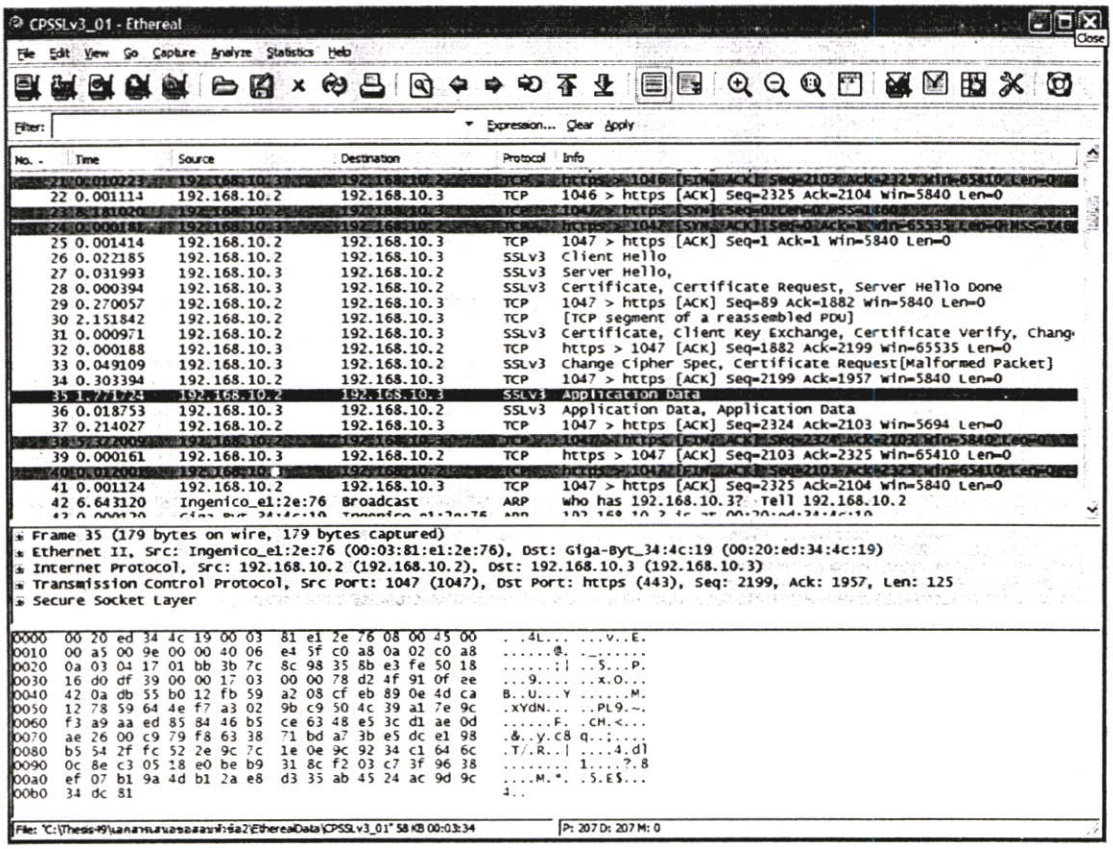


รูปที่ 3.19 แสดงข้อมูล Plaintext จากการลักลอบดัก

3.4.3 ผลของการส่งข้อมูลที่มีลักษณะเป็น Ciphertext

จากขั้นตอนแรกได้ดำเนินการส่งข้อมูล Plaintext ในหัวข้อที่ผ่านมานั้น แสดงให้เห็นว่าเราสามารถมองเห็นลักษณะของข้อมูลได้ ในขั้นต่อมาทำการส่งข้อมูลในลักษณะเดิม แต่ในครั้งนี้ทำการเข้ารหัสข้อมูลบน SSL Protocol ที่มีกระบวนการสร้างการสื่อสารแบบ DES-CBC3-SHA ทำการเข้ารหัสข้อมูลแบบ 3DES และแสดงให้เห็นว่ามีขบวนการสร้างความปลอดภัยของข้อมูลดังแสดงในรูปที่ 3.20 โดยใช้ข้อมูลขนาด 92 ไบต์ ซึ่งเป็นข้อมูลชุดเดียวกันกับการส่งข้อมูลที่เป็น

Plaintext ดังนั้นข้อมูลที่มีขบวนการรักษาความปลอดภัย โดยการเข้ารหัสข้อมูล จะไม่สามารถมองเห็นว่าเป็นข้อมูลอะไรที่ส่งผ่านเข้าไปในระบบเครือข่าย



รูปที่ 3.20 แสดงข้อมูลที่มีการเข้ารหัสบน SSL Protocol

ทำการส่งข้อมูลเข้ารหัสด้วย SSLv2 และ SSLv3 เป็นจำนวน 10, 20, 30, 40 และ 50 ครั้ง เพื่อหาค่าเฉลี่ยเวลาเปรียบเทียบของทั้ง 2 เวอร์ชัน

เมื่อได้ข้อมูลจากการทดลองทั้งหมด ทำการหาค่าเฉลี่ยเวลาที่ใช้ในการสื่อสารข้อมูลโดยการใช้สมการ

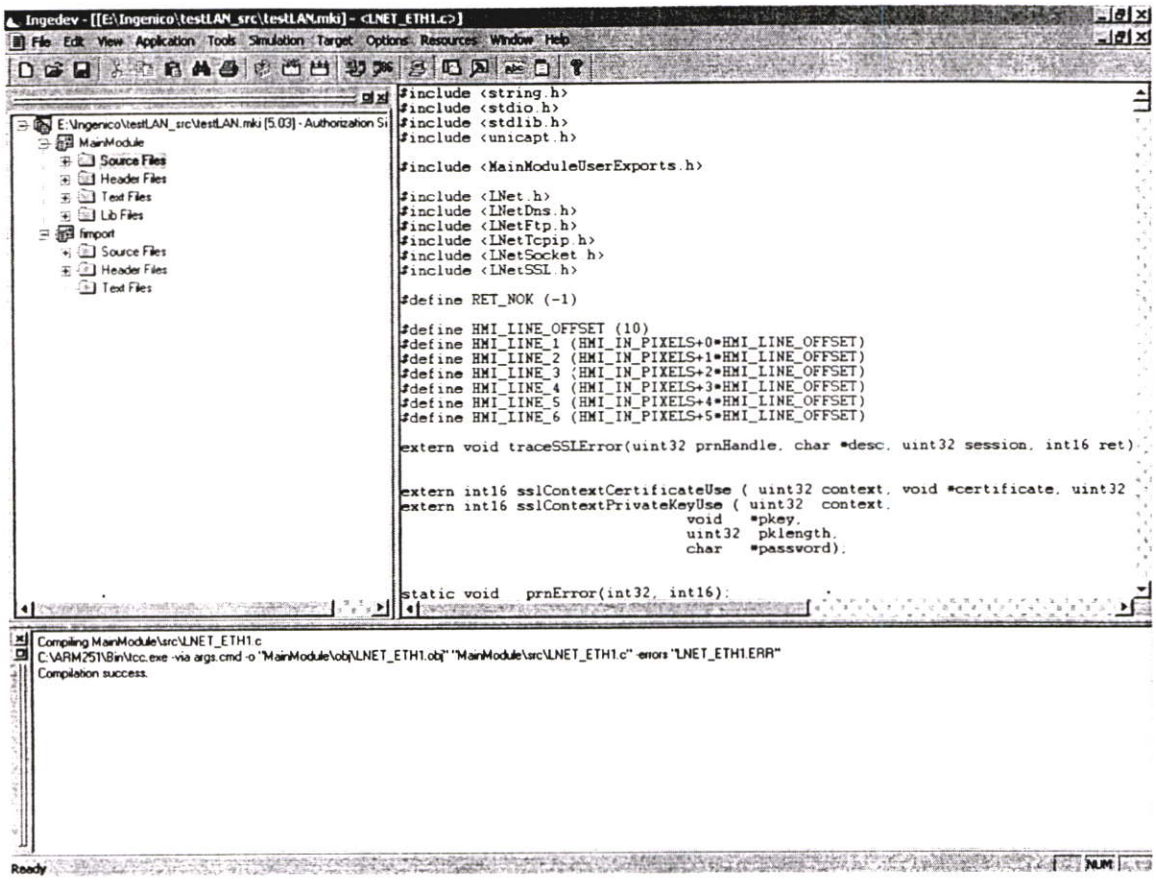
$$T = \frac{1}{N}\sum_{i=1}^{N} t_i \qquad (3.1)$$

โดยที่ T คือค่าเฉลี่ยของเวลาที่ใช้รับ-ส่งข้อมูลทั้งหมด

N คือจำนวนครั้งที่ทดลอง

t_i คือค่าของเวลาที่ใช้รับ-ส่งข้อมูลแต่ละครั้ง

3.4.4 เครื่องมือใช้เขียนโปรแกรม



รูปที่ 3.21 แสดงเครื่องมือใช้เขียนโปรแกรม

ในรูปที่ 3.21 เป็นการแสดงลักษณะของเครื่องมือที่ใช้ในการสร้างชุดโปรแกรมที่ได้ทำการสร้างระบบการรักษาความปลอดภัย SSL เพื่อทำการเขียนโปรแกรมลงไปที่อุปกรณ์ EDC โดยสามารถคอมไพล์ และป้อนข้อมูลได้

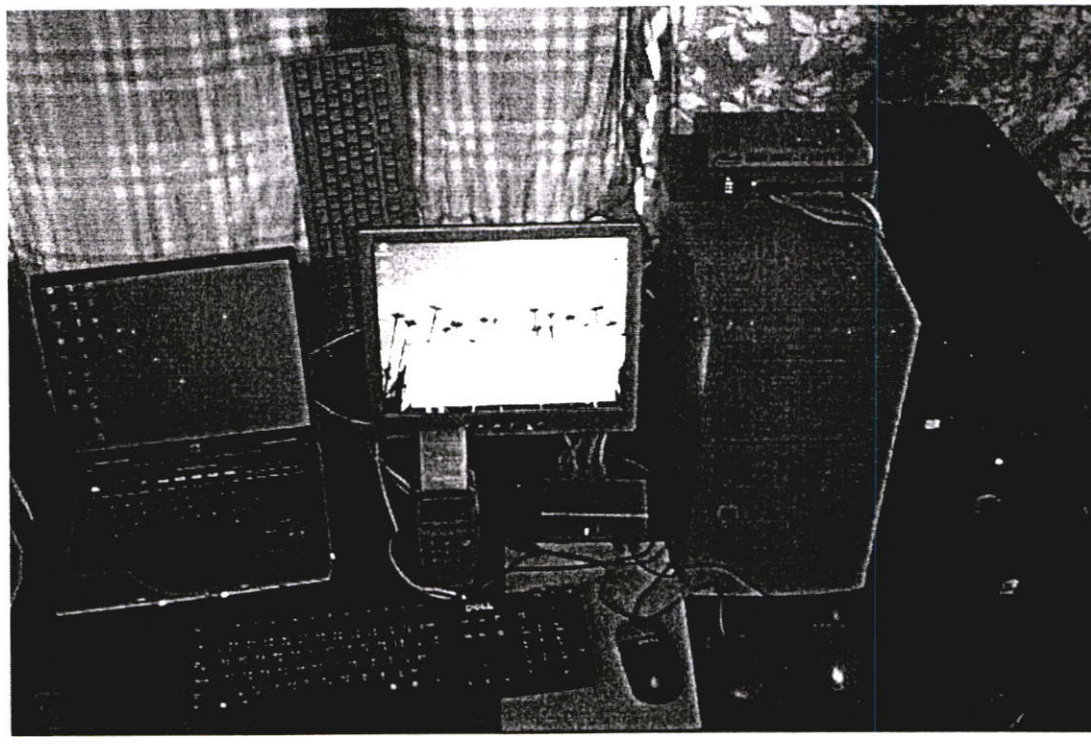
บทที่ 4

ผลการทดลอง

จากการที่ได้นำเสนอรูปแบบของ SSL Protocol และ การเข้ารหัสข้อมูลบน EDC แล้วนั้น ในบทนี้จะได้ทำการวัดประสิทธิภาพการทำงานของ EDC ในรูปแบบการเข้ารหัสแบบต่างๆ

4.1 ผลการทดลอง

ทำการเชื่อมต่อระบบเครือข่ายในขั้นการทดลองเก็บข้อมูลแสดงดังรูปที่ 4.1 ซึ่งเป็นรูปของอุปกรณ์และเครื่องมือที่ใช้ทดลอง



รูปที่ 4.1 ระบบที่ใช้ในการทดลอง

การเก็บข้อมูล

ทำการทดลองส่งข้อมูลจากอุปกรณ์ EDC ไปยัง SSL Server ที่มีระบบปฏิบัติการเป็น Windows XP + Stunnel V. 4.14 , CPU: 1.8 GHz , RAM : 512 MB , NIC : 10/100 และวัดค่าเวลาโดยเฉลี่ยได้จาก

$$T = \frac{1}{N}\sum_{i=1}^{N} t_i \qquad (4.1)$$

สมการที่ 4.1 เป็นการหาค่าเวลาเฉลี่ย

$$\text{Throughput} = \frac{Data}{AvageTime} \qquad (4.2)$$

สมการที่ 4.2 เป็นการหาค่าอัตราการส่งผ่านข้อมูลเฉลี่ย

4.1.1 ผลการทดลองรับส่งข้อมูลโดยใช้รูปแบบการเข้ารหัสแบบ RC4-MD5

ผลการทดลองส่ง-รับข้อมูล ที่เป็น Plaintext, SSLv2 และ SSLv3 ขนาด 92 ไบต์ เฉพาะช่วงเวลาส่งและรับข้อมูล จำนวน 10, 20, 30, 40 และ 50 ครั้ง ได้ค่าเฉลี่ยตามตารางที่ 4.1 ถึง 4.4 และรูปที่ 4.2 และ 4.3

ตารางที่ 4.1 แสดงค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

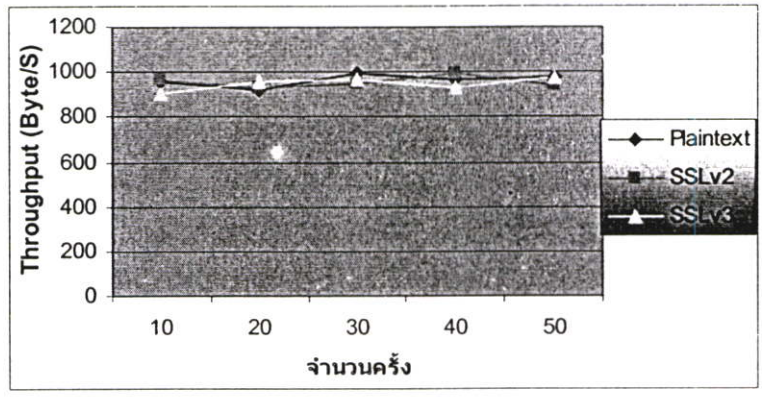
จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.096030	0.095246	0.101498
20	0.100082	0.098872	0.096506
30	0.092624	0.097386	0.094983
40	0.095043	0.092942	0.098876
50	0.094251	0.098226	0.094487

ตารางที่ 4.2 แสดงค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.065120	0.150940	0.130676
20	0.067751	0.153371	0.162724
30	0.069205	0.139709	0.153815
40	0.072084	0.156546	0.141860
50	0.066563	0.129560	0.148475

ตารางที่ 4.3 แสดงค่าเฉลี่ยของ throughput ในการส่งข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

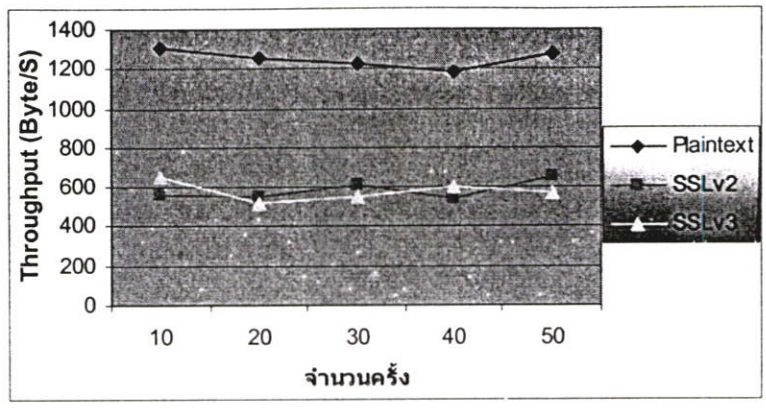
จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	958.03	965.92	906.42
20	919.25	930.50	953.30
30	993.27	944.69	968.59
40	967.98	989.86	930.46
50	976.12	936.62	973.68



รูปที่ 4.2 แสดงกราฟค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

ตารางที่ 4.4 แสดงค่าเฉลี่ยของ throughput ในการรับข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	1305.28	563.14	650.46
20	1254.59	554.21	522.36
30	1228.24	608.41	552.61
40	1179.19	542.97	599.18
50	1276.99	656.07	572.49



กราฟที่ 4.3 แสดงกราฟค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5

4.1.2 ผลการทดลองรับส่งข้อมูลโดยใช้รูปแบบการเข้ารหัสแบบ EXP-RC4-MD5

ผลการทดลองส่ง-รับข้อมูล ที่เป็น Plaintext, SSLv2 และ SSLv3 ขนาด 92 ไบต์ เฉพาะช่วงเวลาส่งและรับข้อมูล จำนวน 10, 20, 30, 40 และ 50 ครั้ง ได้ค่าเฉลี่ยตามตารางที่ 4.5 ถึง 4.8 และรูปที่ 4.4 และ 4.5

ตารางที่ 4.5 แสดงค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

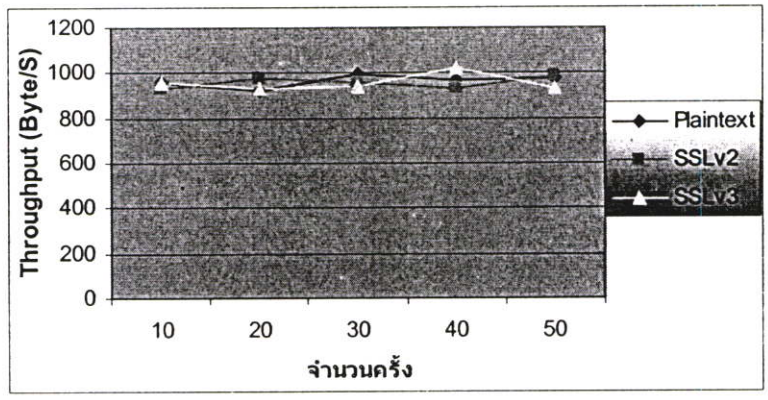
จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.096030	0.098731	0.095621
20	0.100082	0.094543	0.099204
30	0.092624	0.096085	0.097380
40	0.095043	0.098756	0.090243
50	0.094251	0.093127	0.098412

ตารางที่ 4.6 แสดงค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.065120	0.125648	0.139365
20	0.067751	0.133626	0.161148
30	0.069205	0.144311	0.139893
40	0.072084	0.154794	0.141828
50	0.066563	0.156624	0.148033

ตารางที่ 4.7 แสดงค่าเฉลี่ยของ throughput ในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

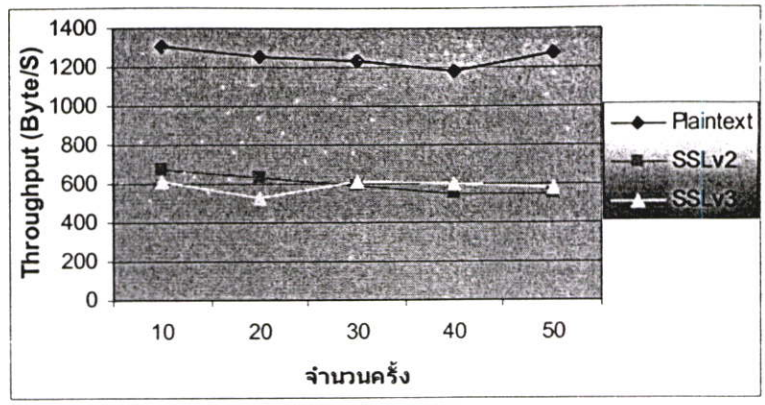
จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	958.03	931.83	962.14
20	919.25	973.10	927.39
30	993.27	957.48	944.75
40	967.98	931.59	1019.47
50	976.12	987.90	934.84



กราฟที่ 4.4 แสดงกราฟค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

ตารางที่ 4.8 แสดงค่าเฉลี่ยของ throughput ในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	1305.28	676.49	609.91
20	1254.59	636.11	527.47
30	1228.24	589.01	607.61
40	1179.19	549.12	599.32
50	1276.99	542.70	574.20



กราฟที่ 4.5 แสดงกราฟค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC4-MD5

4.1.3 ผลการทดลองรับส่งข้อมูลโดยใช้รูปแบบการเข้ารหัสแบบ EXP-RC2-CBC-MD5

ผลการทดลองส่ง-รับข้อมูล ที่เป็น Plaintext, SSLv2 และ SSLv3 ขนาด 92 ไบต์ เฉพาะช่วงเวลาส่งและรับข้อมูล จำนวน 10, 20, 30, 40 และ 50 ครั้ง ได้ค่าเฉลี่ยตามตารางที่ 4.9 ถึง 4.12 และรูปที่ 4.6 และ 4.7

ตารางที่ 4.9 แสดงค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

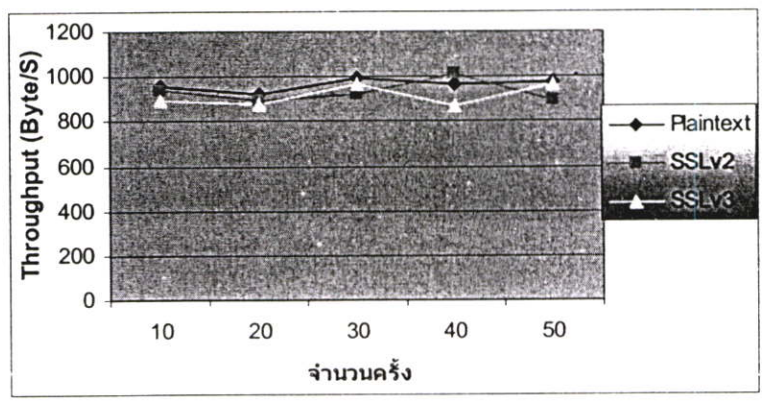
จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.096030	0.098359	0.102981
20	0.100082	0.103307	0.104934
30	0.092624	0.100232	0.095281
40	0.095043	0.091194	0.106098
50	0.094251	0.102920	0.094906

ตารางที่ 4.10 แสดงค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

จำนวนครั้งที่ส่ง	Plaintext (Sec)	SSLv2 (Sec)	SSLv3 (Sec)
10	0.065120	0.142333	0.155662
20	0.067751	0.135482	0.150719
30	0.069205	0.140000	0.144080
40	0.072084	0.142892	0.145775
50	0.066563	0.149036	0.149675

ตารางที่ 4.11 แสดงค่าเฉลี่ยของ throughput ในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

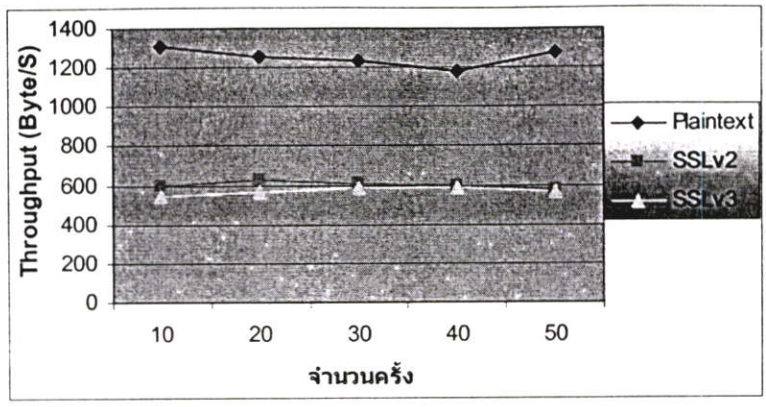
จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	958.03	935.35	893.37
20	919.25	890.55	876.74
30	993.27	917.87	965.56
40	967.98	1008.84	867.12
50	976.12	893.90	969.38



กราฟที่ 4.6 แสดงกราฟค่าเฉลี่ยในการส่งข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

ตารางที่ 4.12 แสดงค่าเฉลี่ยของ throughput ในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

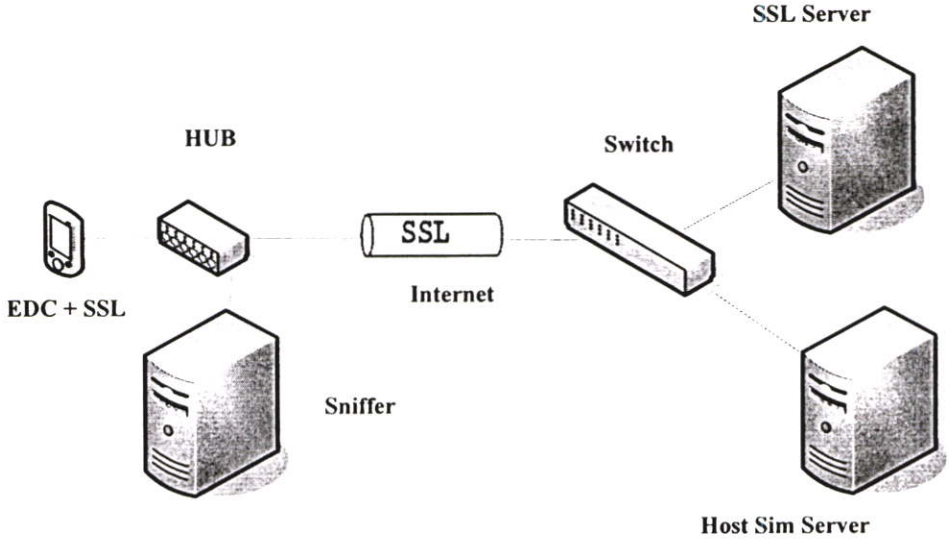
จำนวนครั้งที่ส่ง	Plaintext (Byte/Sec)	SSLv2 (Byte/Sec)	SSLv3 (Byte/Sec)
10	1305.28	597.19	546.05
20	1254.59	627.39	563.96
30	1228.24	607.14	589.95
40	1179.19	594.85	583.09
50	1276.99	570.33	567.90



กราฟที่ 4.7 แสดงกราฟค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ EXP-RC2-CBC-MD5

4.2 เปรียบเทียบผลการทดลอง ส่ง-รับ ข้อมูลแบบ Plaintext กับ SSLv2 และ SSLv3 ที่เข้ารหัสแบบ RC4-MD5, EXP-RC4-MD5 และ EXP-RC2-CBC-MD5

ในรูปที่ 4.8 ในการส่งข้อมูลแบบ Plaintext เป็นการติดต่อกันระหว่างต้นทางคืออุปกรณ์ EDC กับเครื่อง Host Sim Server เท่านั้น ส่วนการส่งข้อมูลโดยใช้ SSLv2 และ SSLv3 ก็จะเป็นการติดต่อกับ SSL Server เท่านั้นเหมือนกัน โดยที่ SSL Server จะเป็นผู้ติดต่อกับ Host Sim Server เอง



รูปที่ 4.8 ระบบที่ใช้ในการทดลอง

ดังนั้นการเปรียบเทียบผลการทดลองส่ง-รับข้อมูลแบบ Plaintext กับ SSLv2 และ SSLv3 ด้วยข้อมูลขนาด 92 ไบต์ โดยใช้ค่าเฉลี่ยในการส่ง-รับข้อมูล จำนวน 10, 20, 30, 40 และ 50 ครั้ง ซึ่งในการส่ง-รับข้อมูลแบบ SSLv2 และ SSLv3 นั้นทำการรหัสสามแบบ คือ RC4-MD5, EXP-RC4-MD5 และ EXP-RC2-CBC-MD5 สามารถสรุปผลการทดลองได้ดังนี้คือ

ตารางที่ 4.13 เปรียบเทียบค่าเฉลี่ยเฉพาะช่วงเวลาในการส่งข้อมูลแบบ Plaintext กับ SSLv2 และ SSLv3 ที่เข้ารหัสแบบ RC4-MD5, EXP-RC4-MD5 และ EXP-RC2-CBC-MD5

ชนิดการเข้ารหัส	ค่าเฉลี่ยการส่ง SSLv2 (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง	ค่าเฉลี่ยการส่ง SSLv3 (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง	ค่าเฉลี่ยการส่ง Plaintext (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง (ไม่มีการเข้ารหัส)
RC4-MD5	953.52	946.49	962.93
EXP-RC4-MD5	956.38	957.72	962.93
EXP-RC2-CBC-MD5	929.30	914.43	962.93

กรณีเปรียบเทียบค่าเฉลี่ยเฉพาะช่วงเวลาในการส่งข้อมูล จากตารางที่ 4.13

1. การส่งข้อมูลแบบ Plaintext เป็นส่งที่ไม่มีการเข้ารหัส ได้ค่าเฉลี่ยเท่ากับ 962.93 Byte/Sec
2. การส่งข้อมูลแบบ SSLv2 ที่เข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 953.52 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 956.38 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 929.30 Byte/Sec
3. การส่งข้อมูลแบบ SSLv3 ที่เข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 946.49 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 957.72 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 914.43 Byte/Sec

จากผลสรุปทั้งสามข้อจะเห็นได้ว่าค่าเฉลี่ยของช่วงเวลาที่ใช้ในการส่งข้อมูลจำนวน 92 Bytes นั้นใช้เวลาใกล้เคียงกันมาก แสดงว่าผลจากการเพิ่มขบวนการในการเข้ารหัสเข้าไปในอุปกรณ์ EDC นั้นไม่ได้ทำให้ประสิทธิภาพในการส่งข้อมูลเปลี่ยนแปลงไปมากนัก แต่มีข้อดีเพิ่มขึ้นคือมีขบวนการในการพิสูจน์ตัวตนและการเข้ารหัสข้อมูลทำให้มีความปลอดภัยเพิ่มขึ้น

จากรูปที่ 4.8 เช่นกันในการรับข้อมูลแบบ Plaintext เป็นการติดต่อกันระหว่างต้นทางคือเครื่อง Host Sim Server กับอุปกรณ์ EDC เท่านั้น ส่วนการรับข้อมูลแบบ SSLv2 และ SSLv3 ก็จะ

เป็นการติดต่อ Host Sim Server กับ SSL Server เพื่อทำการส่งข้อมูลที่เป็น Plaintext ไปให้กับ SSL Server โดยที่ SSL Server จะเป็นผู้ติดต่อกับอุปกรณ์ EDC เองในลักษณะ Ciphertext

ตารางที่ 4.14 เปรียบเทียบค่าเฉลี่ยในการรับข้อมูลด้วยการเข้ารหัสแบบ RC4-MD5, EXP-RC4-MD5 และ EXP-RC2-CBC-MD5

ชนิดการเข้ารหัส	ค่าเฉลี่ยการรับ SSLv2 (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง	ค่าเฉลี่ยการรับ SSLv3 (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง	ค่าเฉลี่ยการรับ Plaintext (Byte/Sec) จำนวน 10,20,30,40,50 ครั้ง (ไม่มีการเข้ารหัส)
RC4-MD5	584.96	579.42	1248.86
EXP-RC4-MD5	598.68	583.70	1248.86
EXP-RC2-CBC-MD5	599.38	570.19	1248.86

กรณีเปรียบเทียบค่าเฉลี่ยเฉพาะช่วงเวลาในการรับข้อมูล จากตารางที่ 4.14

1. การรับข้อมูลแบบ Plaintext เป็นรับที่ไม่มีการเข้ารหัส ได้ค่าเฉลี่ยเท่ากับ 1248.86 Byte/Sec
2. การรับข้อมูลแบบ SSLv2 ที่เข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 584.96 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 598.68 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 599.38 Byte/Sec
3. การรับข้อมูลแบบ SSLv3 ที่เข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 579.42 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 583.70 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 570.19 Byte/Sec

เนื่องจากการรับข้อมูลแบบ Plaintext นั้น เป็นการตอบกลับของข้อมูลจากเครื่องคอมพิวเตอร์ แม่ข่าย Host Sim Server มายังอุปกรณ์ EDC ขนาดของข้อมูลลดลงเหลือ 85 Bytes เท่านั้น ขนาดของข้อมูลลดลงจึงทำให้การรับข้อมูลแบบ Plaintext ได้ค่าเฉลี่ยมากขึ้นเท่ากับ 1248.86 Byte/Sec

ส่วนในกรณีการรับข้อมูลตอบกลับแบบ SSLv2 และ SSLv3 อุปกรณ์ EDC ต้องเพิ่มขบวนการ ในการถอดรหัส ดังนั้นค่าเฉลี่ยในการรับข้อมูลจึงลดลงอยู่ในช่วง 570-599 Byte/Sec แสดงว่าผลจากการเพิ่มขบวนการในการเข้ารหัสเข้าไปในอุปกรณ์ EDC นั้นทำให้ประสิทธิภาพของ

การรับข้อมูลลดลงไปบ้าง แต่มีข้อดีเพิ่มขึ้นคือมีขบวนการในการพิสูจน์ตัวตนและการเข้ารหัสข้อมูล ทำให้มีความปลอดภัยเพิ่มขึ้น

บทที่ 5
สรุปผลและข้อเสนอแนะ

การเพิ่มประสิทธิภาพของอุปกรณ์ Electronic Data Capture ด้วย SSL Protocol สำหรับระบบการชำระเงิน ทำให้อุปกรณ์ EDC สามารถป้องกันการลักลอบดักจับข้อมูล ซึ่งในขั้นการทดลองนั้นไม่สามารถเห็นข้อมูลที่มีการเข้ารหัสแล้ว นั่นหมายความว่าข้อมูลได้รับการป้องกัน แต่ค่าเฉลี่ยเวลาของการสื่อสารข้อมูลต้องใช้เวลาเพิ่มขึ้นบ้างเพื่อความปลอดภัยของข้อมูล

ผลการวิเคราะห์การส่งข้อมูลและวัดประสิทธิภาพของอุปกรณ์ EDC เพื่อเปรียบเทียบเวอร์ชันของ SSL กับการส่งข้อมูลแบบ Plaintext ได้ผลดังต่อไปนี้

1. การส่งข้อมูลแบบ Plaintext ได้ค่าเฉลี่ยเท่ากับ 962.93 Byte/Sec
2. การส่งข้อมูลด้วย SSLv2 โดยทำการเข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 953.52 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 956.38 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 929.30 Byte/Sec
3. การส่งข้อมูลด้วย SSLv3 โดยทำการเข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 946.49 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 957.72 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 914.43 Byte/Sec

ผลที่ได้คือ อัตราเฉลี่ยของการส่งข้อมูลทั้งแบบ Plaintext, SSLv2 ทั้งสามสูตร และ SSLv3 ทั้งสามสูตรได้ค่า Throughput ที่ใกล้เคียงกันมาก สรุปคือการเพิ่ม SSL Protocol ทั้ง SSLv2 และ SSLv3 ให้กับอุปกรณ์ EDC ไม่มีผลทำให้อัตราเฉลี่ยของการส่งข้อมูลเปลี่ยนแปลงเมื่อเปรียบเทียบกับแบบ Plaintext

ผลการวิเคราะห์การรับข้อมูลและวัดประสิทธิภาพของอุปกรณ์ EDC เพื่อเปรียบเทียบเวอร์ชันของ SSL กับการรับข้อมูลแบบ Plaintext ได้ผลดังต่อไปนี้

1. การรับข้อมูลแบบ Plaintext ได้ค่าเฉลี่ยเท่ากับ 1248.86 Byte/Sec
2. การรับข้อมูลด้วย SSLv2 โดยทำการเข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 584.96 Byte/Sec
   - EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 598.68 Byte/Sec
   - EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 599.38 Byte/Sec
3. การรับข้อมูลด้วย SSLv3 โดยทำการเข้ารหัสด้วยสูตร
   - RC4-MD5 (เข้ารหัส 128 บิต) ได้ค่าเฉลี่ยเท่ากับ 579.42 Byte/Sec

- EXP-RC4-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 583.70 Byte/Sec
- EXP-RC2-CBC-MD5 (เข้ารหัส 40 บิต) ได้ค่าเฉลี่ยเท่ากับ 570.19 Byte/Sec

ผลที่ได้คือ อัตราเฉลี่ยของการรับข้อมูลทั้งแบบ SSLv2 ทั้งสามสูตร และ SSLv3 ทั้งสามสูตรได้ค่า Throughput ที่ลดลงครึ่งหนึ่งเมื่อเทียบกับแบบ Plaintext สาเหตุเนื่องจากตัวอุปกรณ์ EDC ที่มีคุณสมบัติ เป็น โพรเซสเซอร์ ARM7 ที่การทำงานช้ากว่าเครื่องคอมพิวเตอร์จึงทำให้การถอดรหัสทำได้ช้ากว่าเครื่องคอมพิวเตอร์ สรุปคือการเพิ่ม SSL Protocol ทั้ง SSLv2 และ SSLv3 ให้กับอุปกรณ์ EDC ทำให้อัตราเฉลี่ยของการรับข้อมูลเปลี่ยนแปลงคือลดลงครึ่งหนึ่ง เมื่อเปรียบเทียบกับแบบ Plaintext

เมื่อรวมผลของการส่งและรับข้อมูลจากการเพิ่ม SSL Protocol เข้ากับอุปกรณ์ EDC จึงทำให้ Throughput ลดลงไปบ้าง แต่ก็มีข้อดีในเรื่องความปลอดภัยของข้อมูลมีเพิ่มขึ้น

ส่วนในเรื่องเข้ารหัสข้อมูลนั้นจะเห็นได้ว่าการเข้ารหัสด้วยสูตร RC4-MD5 เป็นการเข้ารหัสขนาด 128 บิต จึงทำให้มีความปลอดภัยดีกว่าการเข้ารหัสสูตร EXP-RC4-MD5 และ EXP-RC2-CBC-MD5 ซึ่งเป็นการเข้ารหัสเพียง 40 บิต

ดังนั้นจากผลการทดลองเพิ่ม SSL Protocol ทั้ง SSLv2 และ SSLv3 กับอุปกรณ์ Electronic Data Capture สามารถนำไปประยุกต์ใช้กับระบบชำระเงินได้อย่างเหมาะสม

5.1 ปัญหาและข้อเสนอแนะ

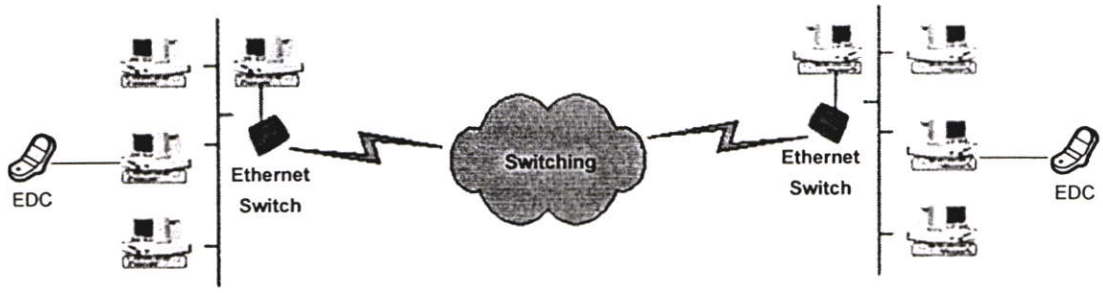
ปัญหาหลักคืออุปกรณ์ EDC เป็นระบบอุปกรณ์ที่ไม่เปิดเผย คือไม่มีขายทั่วไปในท้องตลาด ดังนั้นการที่จะนำอุปกรณ์ EDC มาใช้ในการทดลองจึงไม่สามารถทำได้สะดวกเท่าที่ควร ปัญหาอีกประการคือ ซอฟแวร์ที่ใช้ในการสร้างโปรแกรมก็เป็นซอฟแวร์เฉพาะของระบบ EDC ฉะนั้นจึงเป็นการยากในศึกษาถึงคุณสมบัติของทั้งซอฟแวร์และฮาร์ดแวร์ของอุปกรณ์ EDC

ในส่วนของข้อเสนอแนะของการดำเนินงานขั้นต่อไป จะได้ศึกษาวิธีการตรวจสอบความถูกต้องของ CA Certificates ที่ใช้ในการแลกเปลี่ยนของทั้งสองฝั่ง โดยต้องดำเนินการสร้าง CA Server ขึ้นมาตรวจสอบความถูกต้องนั้น เสมือนว่ามีผู้ดูแลในเรื่องความถูกต้องของ CA Certificates เพื่อใช้ในขั้นตอนการพิสูจน์ตัวตน

5.2 แนวทางในการพัฒนาและการนำไปใช้งาน

การเพิ่มขบวนการพิสูจน์ตัวตนและการเข้ารหัส เพื่อความปลอดภัยของข้อมูลกับอุปกรณ์ EDC (Electronic Data Capture) เพื่อใช้ในการชำระเงินผ่านระบบเครือข่ายที่ทำการทดลองในงานวิจัยนี้ เป็นการสื่อสารผ่านเครือข่ายอีเทอร์เน็ต ซึ่งตรงกับทิศทางของผู้ให้บริการวงจรสื่อสารที่

จะปรับเปลี่ยนมาให้บริการในรูปแบบของอีเทอร์เน็ตที่เรียกว่า เมโทรแลน (MetroLAN) ที่เป็นการสื่อสารความเร็วสูงภายในเมืองที่อาศัยเทคโนโลยีด้านการสื่อสารผ่านใยแก้วนำแสง และให้บริการแล้วในชื่อว่าเมโทรอีเทอร์เน็ต หรือเมโทรแลน โดยพื้นฐานแล้วก็คือการนำสื่อสัญญาณความเร็วสูงระดับ 10 เมกะบิตต่อวินาทีขึ้นไปตรงไปสู่ผู้ใช้ปลายทาง โดยผ่านอุปกรณ์ที่เป็น Ethernet Switch จึงทำให้มีความสามารถในการทำ Virtual Local Area Networks (VLAN's) คือเป็นเครือข่ายเสมือน ทำให้สามารถทำเป็นเครือข่ายส่วนตัว และควบคุมระบบรักษาความปลอดภัยได้อย่างมีประสิทธิภาพ



รูปที่ 5.1 แสดงการเชื่อมต่ออุปกรณ์ EDC ผ่านเน็ตเวิร์กระบบ MetroLAN

ผลจากการประกาศใช้พระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. ๒๕๕๐ ที่ประกาศในราชกิจจานุเบกษา จะมีผลบังคับใช้ในวันที่ 18 กรกฎาคม 2550 และกฎหมายกำหนดหลักเกณฑ์และวิธีการในการทำธุรกรรมทางอิเล็กทรอนิกส์ภาครัฐ ซึ่งการเพิ่มขบวนการพิสูจน์ตัวตนและการเข้ารหัส เพื่อความปลอดภัยของข้อมูลกับอุปกรณ์ EDC เพื่อใช้ในการชำระเงินผ่านระบบเครือข่ายที่ทำการทดลองในงานวิจัยนี้ จึงเป็นการเพิ่มความปลอดภัยในการทำธุรกรรมด้านการเงินผ่านอุปกรณ์ EDC ตามพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ.๒๕๕๐และกฎหมายกำหนดหลักเกณฑ์และวิธีการในการทำธุรกรรมทางอิเล็กทรอนิกส์ภาครัฐ

เอกสารอ้างอิง

[1] ศุภามน วาณิชย์ก่อกุล, สิทธกาณฑ์ ปียมาพรชัย, ชวลิต ทินกรสูติบุตร, เลอศักดิ์ ลิ้มวิวัฒน์กุล **"ความรู้พื้นฐานเกี่ยวกับ โปรโตคอล TCP/IP"** [Online]. Available: http://www.thaicert.nectec.or.th/paper/basic/tcp-ip.php. 2548.

[2] เรืองไกร รังสิพล. **"เจาะระบบ TCP/IP : จุดอ่อนของโปรโตคอลและวิธีป้องกัน"**. บริษัท โปร-วิชั่น จำกัด. 2544.

[3] นพปฎล กุลจรรยาวิวัฒน์. **"คำอธิบายเกี่ยวกับเรื่อง Port"**. [Online]. Available : http://www.thaicert.nectec.or.th/paper/basic/port.php. 2544.

[4] สิริพร จิตต์เจริญธรรม, เสาวภา ปานจันทร์ และ เลอศักดิ์ ลิ้มวิวัฒน์กุล. **"ความรู้เบื้องต้นเกี่ยวกับการพิสูจน์ตัวตน"**. [Online]. Available : http://www.thaicert.nectec.or.th/paper/authen/authentication_guide.php. 2547.

[5] ดร. บรรจง หะรังษี. **"ความรู้เบื้องต้นของการเข้ารหัสข้อมูล (Introduction to Cryptography)"**. [Online]. Available : http://www.thaicert.nectec.or.th/paper/encryption/intro_crypt.php. 2547.

[6] Wolfgang Ranki and Wolfgang Effing ,"**Smart Card Handbook**" 3rd John Wiley and Sons Inc.

[7] A. Alshamsi, and T. Saito , "**A Technical Comparison of IPSec and SSL**" , Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference, vol.2, March 2005, pp. 395-398.

[8] D. Berbecaru, **"On Measuring SSL-based Data Transfer with Handheld Devices"** ISWCS-2005: IEEE International Symposium on Wireless Communication Systems, Siena (Italy), September 5-9, 2005, 5 pages

[9] M.H.Sherif, A.Serhrouchni, A.Y.Gaid, and F.Farazmandnia, **"SET and SSL :Electronic payments on the Internet"**, Computers and Communications, 1998. ISCC '98. Proceedings. Third IEEE Symposium on 30 June-2 July 1998, pp.353 - 358

[10] **"OpenSSL Documents"**. [Online]. Available : http://www.openssl.org/docs. 2549.

[11] John Viega, Matt Messier and Pravir Chandra, "**Network Security with OpenSSL**" O'Reilly

[12] Stephen A. Thomas, "**SSL and TLS Essentials Securing the Web**" John Wiley and Sons Inc.

[13] **"Stunnel-4.20 Man Page".** [Online]. Available :

http://www.stunnel.org/faq/stunnel.html#description. 2548.

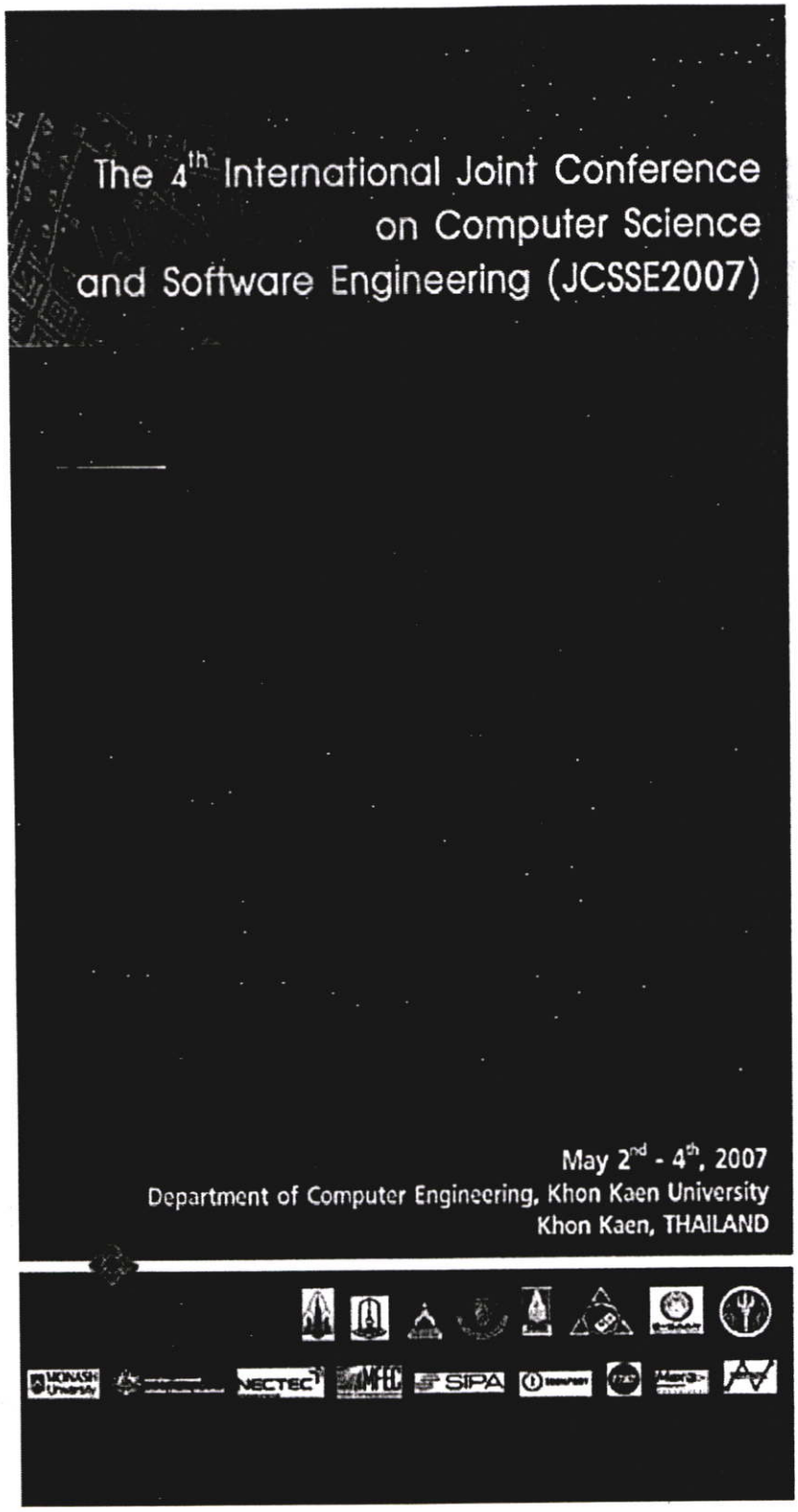
[14] ชวลิต ทินกรสูติบุตร. **"การเสริมสร้างความปลอดภัยด้วยโปรแกรม Stunnel" .** [Online].

Available : http://www.thaicert.nectec.or.th/paper/encryption/stunnel.php. 2546.

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์

1. ยุทธนา สรวลสรรค์, สมศักดิ์ มิตะถา, ศักดิ์ชัย ทิพย์จักษุรัตน์. **"การเพิ่มประสิทธิภาพอุปกรณ์ Electronic Data Capture ด้วย SSL Protocol สำหรับระบบชำระเงิน"**, การประชุมวิชาการนานาชาติร่วมสาขาวิทยาการคอมพิวเตอร์และวิศวกรรมซอฟต์แวร์ ครั้งที่ 4 (JCSSE2007) มหาวิทยาลัยขอนแก่น พฤษภาคม 2550 หน้า 465-470


The 4th International Joint Conference
on Computer Science
and Software Engineering (JCSSE2007)
May 2nd - 4th, 2007
Department of Computer Engineering, Khon Kaen University
Khon Kaen, THAILAND
NECTEC
SIPA


JCSSE
2007

National Joint Conference on Computer Science & Software Engineering (JCSSE2007)
May, 2-4 2007

การเพิ่มประสิทธิภาพอุปกรณ์ Electronic Data Capture ด้วย SSL Protocol สำหรับระบบชำระเงิน

Electronic Data Capture Improvement by SSL Protocol for Payment Systems.

ยุทธนา สรวลสรรค์, สมศักดิ์ มิตะถา, ศักดิ์ชัย ทิพย์จักษุรัตน์
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 10520
E-mail : {s5061213, kmsomsak, ktsakcha}@kmitl.ac.th

บทคัดย่อ

ในบทความนี้ เราได้ทำการวิเคราะห์ประสิทธิภาพภายหลังจากที่ได้เพิ่มระบบความปลอดภัย (SSL Protocol) เข้ากับชุดข้อมูลใน TCP/ IP บนอุปกรณ์ EDC (Electronic Data Capture) ต้นทาง ซึ่งสิ่งดำเนินนั้นได้ทำการส่งข้อมูลที่เป็น Plaintext เปรียบเทียบกับ Ciphertext บน SSL ที่แสดงให้เห็นว่าสามารถป้องกันการลักลอบดักข้อมูลได้ เพื่อความเข้าใจถึงผลกระทบที่มีต่อประสิทธิผลของอุปกรณ์ EDC ความปลอดภัยที่ได้ดำเนินการนั้น ได้ทำการตรวจวัดประสิทธิภาพอัตราการส่งผ่านข้อมูลที่มีความปลอดภัยในหลายรูปแบบของการเข้ารหัส

Abstract

In this paper, we have analyzed the performance when combine the security system (SSL Protocol) into current TCP/IP module by porting the security shareware into an EDC (Electronic Data Capture). Finally, in order to understand the impact on EDC's performance, when using various the encryption algorithms provided the security system, we testing the throughput of the EDC before and after applying security.

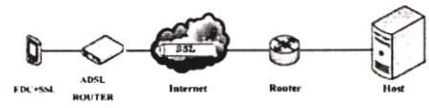
Key words: SSL, EDC, Plaintext, Ciphertext

1. บทนำ

กระแสไฟฟ้า น้ำประปา รวมไปถึง โทรศัพท์พื้นฐาน เป็นตัวอย่างของสิ่งจำเป็นในการดำเนินชีวิตของมนุษย์ในปัจจุบัน ทั้งนี้ภายหลังจากที่เราได้ใช้บริการของระบบสาธารณูปโภคแล้ว เราก็จำเป็นต้องฝ่าวิกฤตจราจรเพื่อไปชำระค่าบริการดังกล่าว แต่ด้วยภารกิจอันจำเป็นของแต่ละบุคคล ทำให้เหลือเวลาดำเนินการค้านนี้น้อยมาก จึงมีผู้ให้บริการอำนวยความสะดวกด้วยการรับชำระค่าบริการหรือสาธารณูปโภค โดยการหักเงินผ่านบัญชีธนาคาร, ชำระผ่านบัตรเครดิต, ชำระที่

เคาน์เตอร์บริการต่างระบบ เป็นต้น แต่บางครั้งก็ได้รับความสะดวกยังไม่เพียงพอ กับการที่ต้องรอคิวเป็นเวลานาน ๆ หรือแม้แต่ชำระผ่านตู้ ATM ซึ่งบางครั้งอาจจะไม่มีตู้ ATM ที่เราถือบัตรอยู่ในบริเวณนั้น ก็เกิดเป็นปัญหาในความไม่สะดวกได้เช่นกัน

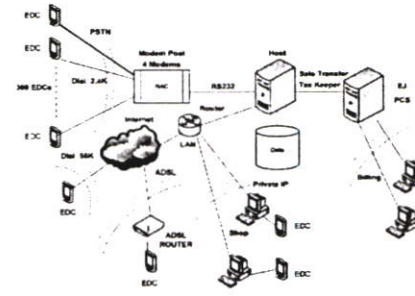
จากปัญหาที่ได้กล่าวมานั้น จึงมีแนวความคิดหาวิธีการใหม่ ซึ่งเป็นทางเลือกอีกทางให้สามารถนำไปใช้ได้ วิธีการที่จะนำเสนอคือการใช้อุปกรณ์ที่เรียกว่า EDC (Electronic Data Capture) เพื่อเป็นสื่อในการชำระผ่านระบบเครือข่าย เป็นการส่งผ่านข้อมูล โดยการรูดบัตร หรือการสแกนบาร์โค้ด ซึ่งวงจรควบคุมที่อยู่ภายในอุปกรณ์ EDC นั้นเป็นไมโครคอนโทรลเลอร์ และที่นำมาใช้ในงานวิจัยนี้ ใช้ชิปตระกูล ARM7 ซึ่งสามารถทำการโปรแกรม และสามารถแก้ไขโปรแกรมได้ อุปกรณ์ EDC ที่ใช้นั้นสามารถพิมพ์ใบเสร็จรับเงินเก็บไว้เป็นหลักฐาน ในการชำระค่าสาธารณูปโภคได้สามารถติดต่อสื่อสารกับระบบเครือข่ายด้วย พอร์ต Modem, RS-232 และ RJ-45 ในการวิจัยครั้งนี้จะทดลองกับพอร์ต RJ-45 ติดต่อระบบเครือข่าย ดังแสดงระบบการเชื่อมต่อในรูปที่ 1 ซึ่งในการต่อเข้าระบบเครือข่ายนั้น เรื่องสำคัญอีกประการคือ เรื่องความปลอดภัยของข้อมูล ในการรับ-ส่งข้อมูลต้องคำนึงถึงความปลอดภัย ดังนั้นจำเป็นต้องมีการเข้ารหัสข้อมูลเพื่อความปลอดภัย โดยงานวิจัยนี้ได้ดำเนินการศึกษาและเพิ่มโปรโตคอลความปลอดภัย SSL ลงไปบนตัวอุปกรณ์ EDC และทำการวิเคราะห์หาอัลกอริธึมที่เหมาะสมกับตัวอุปกรณ์ต่อไป



รูปที่ 1 ผังการเชื่อมต่อ

2. ดำเนินการ

จากการศึกษาระบบชำระ ดังรูปที่ 2 เป็นระบบที่ได้ทำการศึกษารูปแบบ แบ่งเป็นหลายแนวทาง ส่วนแนวทางที่จะนำมาทำการวิจัย คือการที่อุปกรณ์ EDC ติดต่อกับระบบชำระ ผ่านระบบเครือข่ายอินเตอร์เน็ต ซึ่งพอร์ตสื่อสารนั้นจะใช้พอร์ต RJ-45 ที่สามารถติดต่ออินเตอร์เน็ตผ่านทาง ADSL Modem และในส่วนนี้ต้องคำนึงถึงความปลอดภัยของข้อมูลด้วยเช่นกัน ดังนั้นในการวิจัยก็จะทำการศึกษาในเรื่องความปลอดภัยของข้อมูลเป็นหลัก



รูปที่ 2 ผังระบบชำระ

2.1 SSL Protocol

SSL (Secure Socket Layer) [1],[2],[3],[4] เป็นโปรโตคอลที่อยู่ในชั้น Application SSL ส่วนใหญ่แล้วจะป้องกันการแลกเปลี่ยนข้อมูลของ HTTP และเพื่อจุดประสงค์อื่นเช่น IMAP และ POP3 เป็นต้น SSL เป็น

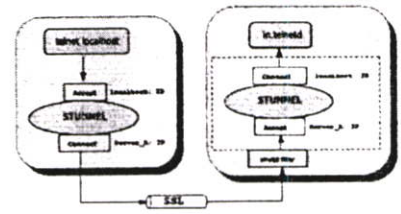
application ที่ทำงานอยู่บน TCP แต่บางอย่างที่มีการปรับเปลี่ยนก็สามารถเป็น application ที่ทำงานอยู่บน SSL เมื่อไม่นานมานี้มีการพัฒนาซอฟต์แวร์ที่เรียกว่า Stunnel [5],[6] สำหรับการใช้งานที่ไม่ค่อยสลับซับซ้อน SSL ประกอบไปด้วยโปรโตคอลดังนี้

1. Handshake protocol
2. Change Cipher Spec protocol
3. Alert protocol
4. Application Data protocol

Handshake protocol ใช้สำหรับการ authentication และ key exchange, Change Cipher Spec protocol ใช้แสดงคีย์ที่ถูกเลือกใช้, Alert protocol ใช้ส่งสัญญาณความผิดพลาด และการปิดงาน และ Application Data protocol ใช้รับ-ส่งข้อมูลที่เข้ารหัส [7]

2.2 หลักการทำงานของ Stunnel

Stunnel ใช้หลักการทำงานของลูกข่าย-แม่ข่าย (Client-Server Model) ดังนั้น Stunnel จะสามารถทำงานได้สองโหมดคือ Client Mode และ Server Mode การทำงานทั้งสองแบบจะอาศัยหลักการของ Port Forwarding เช่นเดียวกับ SSH Tunneling เพื่อเปลี่ยนการเชื่อมต่อของโปรโตคอล หรือโปรแกรมบริการพื้นฐานให้อยู่ภายใต้การทำงานของการเข้ารหัสแบบ SSL ของโปรแกรม Stunnel ศึกษาตัวอย่างต่อไปนี้เพื่อความเข้าใจกระบวนการทำงานของ Stunnel ที่เพิ่มมากขึ้น ถ้าผู้ดูแลระบบที่มีความจำเป็นต้องเปิดให้บริการ telnet กับเครื่องลูกข่าย แต่ผู้ดูแลระบบรู้ดีว่าการให้บริการ telnet นั้นอาจจะไม่ปลอดภัยจากการถูกดักจับรหัสผ่าน ดังนั้นผู้ดูแลระบบจึงป้องกันการดักจับรหัสโดยกำหนดให้การเชื่อมต่อของ telnet อยู่ภายใต้การทำงานของโปรแกรม Stunnel รูปแบบการทำงานของระบบนี้แสดงได้ดังรูปที่ 3



รูปที่ 3 แสดงกระบวนการทำงานของ Stunnel [6]

จากรูปที่ 3 เครื่องเซิร์ฟเวอร์เปิดให้บริการ telnet ผ่านพอร์ต 23 และเข้ารหัสข้อมูลด้วยโปรแกรม Stunnel การขอใช้บริการ telnet จากเครื่องลูกข่ายจะต้องผ่านกระบวนการทำงานของโปรแกรม Stunnel ก่อนมิเช่นนั้นกระบวนการขอใช้บริการจะไม่เกิดขึ้น เนื่องจากข้อมูลที่ติดต่อระหว่างเครื่องลูกข่าย และเครื่องเซิร์ฟเวอร์นั้นถูกเข้ารหัสแบบ SSL ทำให้โปรแกรม telnet ธรรมดาไม่เข้าใจข้อมูลที่ถูกเข้ารหัสเหล่านั้น

2.3 การเข้ารหัสข้อมูล (Cryptography)

การเข้ารหัสข้อมูลโดยพื้นฐานแล้วจะ เกี่ยวข้องกับวิธีการทางคณิตศาสตร์เพื่อใช้ในการป้องกันข้อมูลหรือข้อความดั้งเดิมที่ต้องการส่งไปถึงผู้รับ ข้อมูลดั้งเดิมจะถูกแปรเปลี่ยนไปสู่ข้อมูลหรือข้อความอีกรูปแบบหนึ่งที่ไม่สามารถอ่านเข้าใจได้โดยใครก็ตามที่ไม่มีกุญแจสำหรับเปิดดูข้อมูลนั้น เราเรียกกระบวนการในการแปรรูปของข้อมูลดั้งเดิมว่า "การเข้ารหัสข้อมูล" (Encryption) และกระบวนการในการแปลงข้อความที่ไม่สามารถอ่าน และทำความเข้าใจให้กลับไปสู่ข้อความดั้งเดิม ว่าการถอดรหัสข้อมูล (Decryption) อัลกอริธึม

ในการเข้ารหัสข้อมูล อัลกอริธึมในการเข้ารหัสข้อมูลมี 2 ประเภทหลัก คือ

1. อัลกอริธึมแบบสมมาตร (Symmetric key algorithms) อัลกอริธึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ (Secret key) ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป สามารถแบ่งย่อยออกเป็น 2 ประเภท ได้แก่ แบบบล็อค (Block algorithms) ซึ่งจะทำการเข้ารหัสทีละบล็อค (1 บล็อคประกอบด้วยหลายไบต์ เช่น 64 ไบต์ เป็นต้น) และแบบสตรีม (Stream algorithms) ซึ่งจะทำการเข้ารหัสทีละไบต์

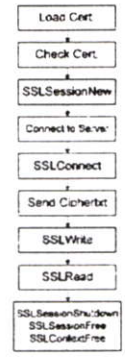
2. อัลกอริธึมแบบอสมมาตร (Asymmetric key algorithms) อัลกอริธึมนี้จะใช้กุญแจสองตัวเพื่อทำงานตัวหนึ่งใช้ในการเข้ารหัสและอีกตัวหนึ่งใช้ในการถอดรหัสข้อมูลที่เข้ารหัสมาโดยกุญแจตัวแรก อัลกอริธึมแบบอสมมาตรนี้คือ อัลกอริธึมแบบกุญแจสาธารณะ (Public keys algorithms) ซึ่งใช้กุญแจที่เรียกกันว่า กุญแจสาธารณะ (Public keys) ในการเข้ารหัสและใช้กุญแจที่เรียกกันว่า กุญแจส่วนตัว (Private keys) ในการถอดรหัสข้อมูลนั้น กุญแจสาธารณะนี้สามารถส่งมอบให้กับผู้อื่นได้ [3], [8], [9]

3. ขั้นเตรียมการ

ในรูปที่ 4 นั้นแสดงโครงสร้างของระบบเครือข่ายที่มี SSL Protocol ชั้นของ TCP/IP

รูปที่ 4 ระบบเครือข่ายที่มี SSL Protocol

สร้างฟังก์ชันการทำงานของ SSL บนอุปกรณ์ EDC จากการที่ได้ศึกษาโดยแสดงหลักการทำงานตามผังการทำงานดังรูปที่ 5



รูปที่ 5 แสดงผังการทำงานของโปรแกรม

คุณสมบัติเครื่องมือ

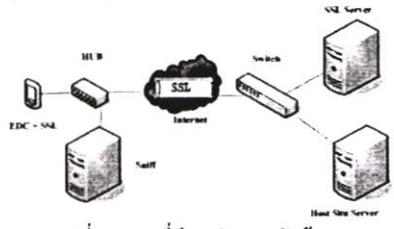
1. PC: SSL Server : Stunnel 4.18
   -Pentium 4, 1.8 GHz, RAM 512MB
   -NIC 100Mbps
2. PC : Host Sever
   -Pentium 4, 1.8 GHz,RAM 512MB
   -NIC 100Mbps
3. PC : Sniffer : Ethereal 0.99.0
   -Pentium 4, 1.8 GHz,RAM 512MB
4. EDC
   -Powerful 32-bit ARM processor
   -8MB Flash RAM and 1MB SRAM
   -128x64 dots graphics LCD and keyboard
   -18 keys with printing protected by long lasting transparent epoxy

-2" fast and silent thermal printer

-TCP/IP interface and high speed modem (up to 56K bps)

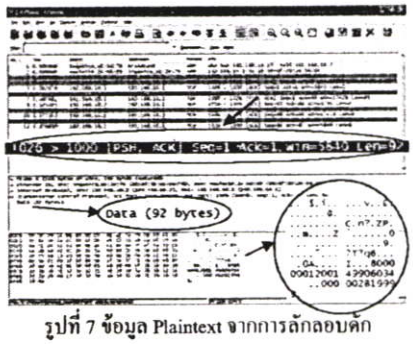
4. การทดลอง

ทำการเชื่อมต่อระบบเครือข่ายในขั้นการทดลองเก็บข้อมูลแสดงดังรูปที่ 6



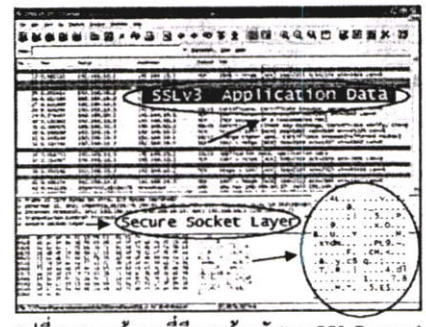
รูปที่ 6 ระบบที่มีการลักลอบดักข้อมูล

ทำการส่งข้อมูลที่เป็น Plaintext จากอุปกรณ์ EDC ไปยังเครื่องเซิร์ฟเวอร์ และทำการลักลอบดักข้อมูลโดยเครื่องคอมพิวเตอร์ PC และแสดงคุณลักษณะของข้อมูลด้วยโปรแกรม Ethereal จากรูปที่ 7 สามารถมองเห็นข้อมูลที่ส่งจาก EDC ไปยังเครื่องเซิร์ฟเวอร์



รูปที่ 7 ข้อมูล Plaintext จากการลักลอบดัก

หลังจากนั้นก็ดำเนินแบบเดิม แต่คราวนี้ทำการเข้ารหัสข้อมูลบน SSL Protocol มีกระบวนการสร้างการสื่อสารแบบ DES-CBC3-SHA [10] ทำการเข้ารหัสข้อมูลแบบ 3DES และแสดงให้เห็นว่ามีขบวนการสร้างความปลอดภัยของข้อมูลดังแสดงในรูปที่ 8



รูปที่ 8 แสดงข้อมูลที่มีการเข้ารหัสบน SSL Protocol

ทำการส่งข้อมูลเข้ารหัสด้วย SSLv2 และ SSLv3 เป็นจำนวน 10, 20, 30, 40 และ 50 ครั้ง เพื่อหาค่าเฉลี่ยเวลาเปรียบเทียบของทั้ง 2 เวอร์ชัน

ทำการหาค่าเฉลี่ยโดยการใช้สมการ

$$T = \frac{1}{N}\sum_{i=1}^{N} t_i$$

โดยที่ T คือค่าเฉลี่ยของเวลาที่ใช้รับ-ส่งข้อมูลทั้งหมด

N คือจำนวนครั้งที่ทดลอง

t_i คือค่าของเวลาที่ใช้รับ-ส่งข้อมูลแต่ละครั้ง

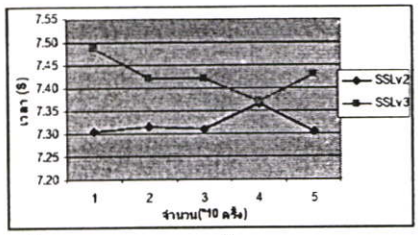
5. ผลการทดลอง

5.1 ผลการทดลองรับ-ส่งข้อมูล ที่เป็น Plaintext ขนาด 92 ไบต์ จากอุปกรณ์ EDC ไป Server เป็นจำนวน 20 ครั้ง ได้ค่าเฉลี่ยของเวลาเท่ากับ 3.39 S

5.2 ผลการทดลองรับ-ส่งข้อมูลเข้ารหัส SSLv2 และ SSLv3 ที่เป็น Ciphertext ขนาด 92 ไบต์ จากอุปกรณ์ EDC ไป Server เป็นจำนวน 10, 20, 30, 40 และ 50 ครั้ง

ตารางที่ 1 แสดงค่าเฉลี่ยเวลาเฉลี่ย (S)

จำนวนครั้ง	SSLv2	SSLv3
10	7.304052	7.486902
20	7.314657	7.420357
30	7.310040	7.420489
40	7.364674	7.371649
50	7.304220	7.429460



รูปที่ 9 กราฟแสดงค่าเวลาเฉลี่ย (S) SSLv2 และ SSLv3

ในตารางที่ 2 แสดงค่าเวลาของการทำ Handshake ด้วย Cipher Suites แบบต่างโดยที่ใช้ RSA Public Key ขนาด 1024 บิต

ตารางที่ 2 เวลาที่ใช้ในการทำ Handshake ด้วย Cipher Suite แบบต่างๆ

SSL Cipher Suite	Time (S)
RC4-SHA	2.966471
EXP-DES-CBC-SHA	2.981825
DES-CBC-SHA	2.877254
DES-CBC3-SHA	2.977532

6. สรุป

การเพิ่มประสิทธิภาพอุปกรณ์ Electronic Data Capture ด้วย SSL Protocol สำหรับระบบการชำระเงิน ได้ดำเนินการเพิ่มโปรโตคอลความปลอดภัย SSL บนอุปกรณ์ EDC ซึ่งสามารถป้องกันการลักลอบดักจับข้อมูล ในขั้นการทดลองไม่สามารถเห็นข้อมูลเมื่อมีการเข้ารหัสแล้ว นั่นหมายความว่า ข้อมูลได้รับการป้องกัน และในการทดลองได้ทำการเปลี่ยนกุญแจเข้ารหัส โดยที่เซิร์ฟเวอร์สามารถร้องขอ Certificate จากไคลเอ็นต์เพื่อตรวจสอบความถูกต้องของไคลเอ็นต์ ในกรณีที่มีการจำกัดเฉพาะไคลเอ็นต์ที่ต้องการเท่านั้น ซึ่ง SSL สนับสนุนการตรวจสอบได้จากทั้งเซิร์ฟเวอร์และไคลเอ็นต์ ขึ้นอยู่กับการเลือกใช้งานขณะติดต่อสื่อสารที่เกิดขึ้นนั้น เพื่อความปลอดภัยของข้อมูล ดังนั้นค่าเฉลี่ยเวลาของการสื่อสารข้อมูลจึงต้องใช้เวลาเพิ่มขึ้น

ทำการวิเคราะห์การส่งข้อมูลและวัดประสิทธิภาพของ EDC เมื่อเปรียบเทียบเวอร์ชันของ SSL จะเห็นว่า SSLv2 นั้นสามารถส่งข้อมูลได้เร็วกว่า SSLv3

ดังนั้นจากผลการทดลองสามารถนำไปประยุกต์ใช้กับระบบชำระเงินได้อย่างเหมาะสม

7. เอกสารอ้างอิง

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

[2] http://www.apache-ssl.org/

[3] Stephen A. Thomas, "SSL and TLS Essentials Securing the Web" John Wiley and Sons Inc.

[4] John Viega, Matt Messier and Pravir Chandra, "Network Security with OpenSSL" O'Reilly

[5] http://www.stunnel.org/faq/stunnel.html#description

[6] http://www.thaicert.nectec.or.th/paper/encryption/stunnel.php

[7] A. Alshamsi, and T. Saito , "A Technical Comparison of IPSec and SSL," Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference, vol.2, March 2005, pp. 395-398.

[8] http://www.thaicert.nectec.or.th/paper/encryption/intro_crypt.php

[9] Wolfgang Ranki and Wolfgang Effing ,"Smart Card Handbook" 3rd John Wiley and Sons Inc.

[10] D. Berbecaru, "On Measuring SSL-based Data Transfer with Handheld Devices" *ISWCS-2005: IEEE International Symposium on Wireless Communication Systems*, Siena (Italy), September 5-9, 2005, 5 pages

ประวัติผู้เขียน

ชื่อ-สกุล	นายยุทธนา สรวลสรรค์
วันเดือนปีเกิด	วันที่ 25 มิถุนายน พ.ศ. 2515 ณ จังหวัดศรีสะเกษ
ที่อยู่	90/123 ซ.วงศ์สว่าง 19 ถ.วงศ์สว่าง บางซื่อ กรุงเทพฯ
ประวัติการศึกษา	
พ.ศ. 2538	สำเร็จการศึกษาครุศาสตร์บัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2544	สำเร็จการศึกษาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ ศูนย์กลางสถาบันเทคโนโลยีราชมงคล
พ.ศ. 2550	สำเร็จการศึกษาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประสบการณ์ทำงาน	
พ.ศ. 2538-ปัจจุบัน	เข้ารับราชการตำแหน่งอาจารย์สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนครเหนือ