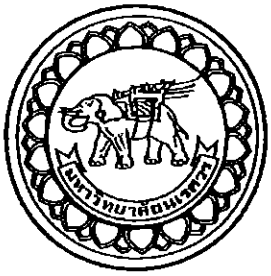


การออกแบบระบบป้องกันสแปม

SPAM PREVENTION SYSTEM DESIGN

นายสุทธิพงศ์ คงชุม รหัส 49362277

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การออกแบบระบบป้องกันสแปม

ผู้ดำเนินโครงงาน นายสุทธิพงศ์ คงชุม รหัส 49362277

ที่ปรึกษาโครงงาน อ. ภาณุพงศ์ สอนคม

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

..... ที่ปรึกษาโครงงาน
(อ. ภาณุพงศ์ สอนคม)

..... กรรมการ
(ดร. สุรเดช จิตประไพกุลศาล)

..... กรรมการ
(อ.เศรษฐา ตั้งค้าวานิช)

ชื่อหัวข้อโครงงาน	การออกแบบระบบป้องกันสแปม
ผู้ดำเนินโครงงาน	นายสุทธิพงศ์ คงชุม รหัส 49362277
ที่ปรึกษาโครงงาน	อ. ภาณุพงศ์ สอนคม
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2553

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอโครงงานเกี่ยวกับการออกแบบระบบป้องกันสแปม เนื่องจากในปัจจุบัน การเพิ่มขึ้นของสแปมเมล์เป็นปัญหาที่สร้างความรำคาญให้แก่ผู้ใช้ ทำให้สิ้นเปลืองทรัพยากร อีกทั้งยังทำให้เสียเวลาในการกำจัดอีเมล์ดังกล่าว ซึ่งวิธีการป้องกันสแปมเมล์ที่มีอยู่ถูกบุกรุกได้ง่ายมากขึ้นจากโปรแกรมอัตโนมัติที่มีความเร็วมากขึ้น ผู้จัดทำได้เล็งเห็นถึงความสำคัญดังกล่าว จึงได้จัดทำโครงงานนี้ขึ้นเพื่อแก้ปัญหาดังกล่าวด้วยการประยุกต์หลักการของการเข้ารหัสลับและเทคนิคการป้องกัน สแปมแบบ Cost-based Spam Control ในลักษณะการทำงานบางอย่างเพื่อยืนยันตัวตนก่อนส่งอีเมล์ หรือวิธีการ Proof Of Work เป็นการป้องกันตั้งแต่ผู้ส่งเพื่อให้ไม่ให้มีการส่งได้โดยง่าย ซึ่งระบบที่ได้ยังสามารถป้องกันและทำให้ผู้บุกรุกไม่สามารถหาคำตอบล่วงหน้าด้วยการใช้โปรแกรมอัตโนมัติได้ง่ายอีกด้วย

Project title	Spam prevention system design
Name	Mr. Sutthipong Kongchum ID. 49362277
Project advisor	Mr. Panupong Sornkhom
Major	Computer Engineering
Department	Electrical and Computer Engineering
Academic year	2010

.....

Abstract

In this senior project, a spam prevention system design. Because of today, increasing of spam mail is problem which make user feel annoyed, waste user's resource and use a lot of time to remove them. Old methods of spam prevention are easier to attack by automatic programs which are faster, in this project. I designed a system by using cryptography and Cost-based Spam Control by Proof of Work technique. Senders must affirm them by do something before send e-mail. This system makes senders unable to send e-mail easily and protect and cannot pre-compute answer by using automatic program speedily.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาอย่างยิ่งจาก อาจารย์ภาณุพงศ์ สอนคม ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่ได้ชี้แนะแนวทางการทำโครงงานจนสำเร็จลุล่วง ผู้ดำเนินโครงงานขอกราบขอบพระคุณท่านเป็นอย่างสูง และขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณ ดร. สุรเดช จิตประไพกุลศาล และ อาจารย์เศรษฐา ตั้งค้าวานิช ที่กรุณาสละเวลามาเป็นกรรมการสอบโครงงาน พร้อมให้คำแนะนำที่เป็นประโยชน์

ขอขอบคุณคณาจารย์ทุกท่านที่เป็นผู้ประสิทธิ์ประสาทวิชาความรู้จนประสบความสำเร็จ

ขอขอบคุณคณะวิศวกรรมศาสตร์ที่ได้สนับสนุนค่าใช้จ่ายส่วนหนึ่งในการทำโครงงานนี้

นอกจากนี้ยังต้องขอขอบคุณการประปานครหลวง สำนักงานใหญ่ ที่ได้จุดประกายความคิดให้แก่ผู้จัดทำในการทำโครงงานนี้ รวมทั้งยังเป็นที่ปรึกษาที่ดีตลอดการทำโครงงานนี้อีกด้วย

เหนือสิ่งอื่นใด ผู้ดำเนินโครงงานขอกราบขอบคุณบุพการีของผู้จัดทำ ที่มอบความรัก ความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจให้ฟันฝ่าอุปสรรคมาจนถึงทุกวันนี้

นายสุทธิพงศ์ ดงชุม

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบัน การสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ตนั้น เป็นที่นิยมอย่างแพร่หลาย โดยเฉพาะการสื่อสารผ่านจดหมายอิเล็กทรอนิคส์ (E-mail) ซึ่งเป็นที่ทราบกันดีว่า การติดต่อสื่อสารโดยจดหมายอิเล็กทรอนิคส์นั้นมีความสะดวกสบาย ประหยัดเวลา ค่าใช้จ่าย แต่มีจดหมายอิเล็กทรอนิคส์อยู่ประเภทหนึ่ง ซึ่งได้รับมาโดยที่ผู้ใช้ไม่ได้ต้องการที่จะรับ โดยถูกส่งมาจากบุคคลหรือโปรแกรมบางประเภทที่ทำการส่งจดหมายโดยอัตโนมัติ ซึ่งเรียกว่า Spam Mail (สแปมเมล์) ซึ่งจดหมายประเภทนี้ได้สร้างความเสียหายแก่ผู้ใช้ ไม่ว่าจะเป็นทำให้เสียเวลาสิ้นเปลืองทรัพยากรของผู้ใช้โดยไม่ก่อให้เกิดประโยชน์ใดๆ

ผู้จัดทำได้เล็งเห็นถึงความสำคัญของปัญหาดังกล่าว จึงได้จัดทำโครงงานการออกแบบระบบป้องกันสแปม (Spam Prevention System Design) ขึ้น เพื่อเป็นการป้องกันและลดจำนวนของสแปมที่อาจสร้างความเสียหายดังที่กล่าวไว้ข้างต้น โดยใช้หลักการของการเข้ารหัสลับ (Cryptography) มาประยุกต์ใช้ในการจัดทำโครงงานนี้ ซึ่งจะทำให้ระบบป้องกันสแปมนั้นมีประสิทธิภาพมากขึ้น

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาและประยุกต์เทคนิคที่ดีของการเข้ารหัสลับ ใช้ร่วมกับระบบจดหมายอิเล็กทรอนิคส์เพื่อป้องกันการส่งสแปมของผู้ที่ไม่หวังดี

1.2.2 เพื่อออกแบบระบบที่สามารถทำการร่วมกับระบบจดหมายอิเล็กทรอนิคส์ที่มีอยู่ได้

1.3 ขอบเขตของโครงงาน

1.3.1 ออกแบบระบบป้องกันสแปมโดยใช้หลักการของการเข้ารหัสลับ

1.3.2 สร้างระบบเซิร์ฟเวอร์ – ไคลเอนท์จำลอง และส่วนติดต่อผู้ใช้โดยใช้ภาษา C# ในการพัฒนาโปรแกรม

1.3.3 ศึกษาหลักการ Proof Of Work และออกแบบส่วนติดต่อผู้ใช้โดยประยุกต์จากหลักการดังกล่าวโดยใช้ภาษา C# ในการพัฒนา

1.4 ขั้นตอนการดำเนินโครงงาน

ลำดับที่	ขั้นตอนการดำเนินงาน	ระยะเวลาดำเนินงาน					
		ปี 2553	ปี 2554				
		ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.	ศึกษาเกี่ยวกับระบบการรับส่งข้อมูลผ่านระบบจดหมายอิเล็กทรอนิคส์ (E-mail)	←	→				
2.	ศึกษาเกี่ยวกับการทำงานของสแปมเมล์ ช่องทางที่สแปมเมอร์ (Spammer) ใช้ในการส่งสแปมเมล์ ความเสียหายและผลกระทบในด้านต่างๆที่เกิดจากสแปมเมล์		↔				
3.	ศึกษาเกี่ยวกับการเข้ารหัสลับ วิธีการที่นำมาใช้ในการเข้ารหัสลับ ข้อดีและข้อเสียของแต่ละวิธี รวมไปถึงการเลือกวิธีการใดวิธีการหนึ่งที่ได้ศึกษามา สร้างเป็นระบบป้องกันสแปม		↔				
4.	ทำการวางแผน ออกแบบ และสร้างระบบตามที่ได้วางแผนไว้			↔	↔	↔	
5.	ทดสอบการทำงานของระบบว่าสามารถใช้ได้จริงหรือไม่ และทำการแก้ไขข้อผิดพลาดที่เกิดจากระบบ					↔	↔
6.	จัดทำคู่มือสำหรับการใช้งานระบบป้องกันสแปม						↔

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

- ได้ระบบป้องกันสแปมเมล์ที่สามารถป้องกันและลดจำนวนสแปมเมล์ได้

1.6 งบประมาณที่ใช้

1.6.1	ค่าถ่ายเอกสาร	150	บาท
1.6.2	ค่าหนังสืออ้างอิง	650	บาท
1.6.3	ค่าจัดทำรูปเล่ม	100	บาท
1.6.4	ค่าอุปกรณ์เครื่องเขียน	100	บาท
	รวมทั้งสิ้น	1,000	บาท

หมายเหตุ เฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีเบื้องต้นและหลักการควบคุม

2.1 วิทยาการเข้ารหัสลับ

วิทยาการเข้ารหัสลับ (Cryptography/Cryptology) เป็นวิชาเกี่ยวกับการเข้ารหัสลับ คือ การแปลงข้อความปกติให้กลายเป็นข้อความลับ โดยข้อความลับคือข้อความที่ผู้อื่นนอกเหนือจากคู่สนทนาที่ต้องการไม่สามารถเข้าใจได้

มนุษย์ได้คิดค้นวิธีการรักษาความลับมาตั้งแต่สมัยโบราณ นับตั้งแต่สมัยจูเลียส ซีซาร์ จนกระทั่งถึงปัจจุบันที่ใช่คอมพิวเตอร์มาช่วยเข้ารหัสลับและถอดรหัสลับ การเข้ารหัสลับแบบซีซาร์ทำโดยการนำตัวอักษาที่อยู่ถัดไปอีกสองตำแหน่งมาแทนที่ ยกตัวอย่างเช่น ถ้าต้องการเข้ารหัสคำว่า HELLO เราก็นำตัวอักษรที่ถัดจากตัว H ไปอีกสองตัวนั่นคือ J มาแทนที่ เช่นเดียวกับตัวอักษรอื่นๆ ตัว E แทนด้วย G ตัว L แทนด้วยตัว N ตัว O แทนด้วย Q ดังนั้น ข้อความ HELLO จึงถูกแปลงให้เป็นคำว่า JGNNQ

การเข้ารหัสลับมีความแตกต่างกับวิทยาการอำพรางข้อมูล ข้อมูลที่ถูกอำพรางนั้นจะไม่ถูกเปลี่ยนแปลง ในขณะที่การเข้ารหัสลับนั้นจะมีการเปลี่ยนแปลงข้อมูลในขั้นตอนการเข้ารหัสและถอดรหัส

วิทยาการเข้ารหัสลับสมัยใหม่ (Modern Cryptography) เป็นวิชาการที่ใช้แนวทางคณิตศาสตร์เพื่อแปลงข้อความปกติให้กลายเป็นข้อความลับ โดยให้เฉพาะคู่สนทนาที่ต้องการสามารถอ่านเข้าใจได้เท่านั้น ขั้นตอนวิธีของการเข้ารหัสลับสมัยใหม่ ได้แก่ Data Encryption Standard, Advanced Encryption Standard หรือ One-Time Padding ฯลฯ โดยหลักการเบื้องต้นของการเข้ารหัสลับมีอยู่สองประการ ประการแรกคือ ขั้นตอนวิธีต้องเป็นที่รู้โดยทั่วไป ประการต่อมาก็คือ รหัสจะต้องใหม่เสมอ

ระบบการเข้ารหัสข้อมูล

ระบบการเข้ารหัสข้อมูล เป็นกระบวนการสำหรับการแปรรูปข้อมูลอิเล็กทรอนิกส์ธรรมดาให้อยู่ในรูปที่บุคคลทั่วไปไม่สามารถอ่านเข้าใจได้ ซึ่งโดยทั่วไปแล้วการเข้ารหัสจะกระทำก่อนการจัดเก็บข้อมูลหรือก่อนส่งข้อมูล โดยการนำข้อมูลอิเล็กทรอนิกส์ธรรมดากับกุญแจ (Key) ซึ่งเป็นตัวเลขสุ่มใดๆมาผ่านกระบวนการทางคณิตศาสตร์ ผลที่ได้คือข้อมูลที่ถูกเข้ารหัส

ขั้นตอนที่กล่าวมานี้จะเรียกว่า "การเข้ารหัส" (Encryption) และเมื่อต้องการอ่านข้อมูล ก็นำเอาข้อมูลที่เข้ารหัสกับกุญแจมาผ่านกระบวนการทางคณิตศาสตร์ ผลลัพธ์ที่ได้ก็คือข้อมูลดั้งเดิม ซึ่งขั้นตอนนี้เรียกว่า "การถอดรหัส" (Decryption) ระบบเข้ารหัสสามารถแบ่งตามวิธีการใช้กุญแจได้เป็น 2 วิธีดังนี้

1. ระบบเข้ารหัสแบบกุญแจสมมาตร (Symmetric-key Cryptography) คือการเข้ารหัสด้วยกุญแจเดี่ยว (Secret Key) ทั้งผู้ส่งและผู้รับ โดยวิธีการนี้ผู้รับกับผู้ส่งต้องตกลงกันก่อนว่าจะใช้รูปแบบไหนในการเข้ารหัสข้อมูล ซึ่งรูปแบบไหนในการเข้ารหัสข้อมูลที่ผู้รับกับผู้ส่งตกลงกันแท้ที่จริงก็คือกุญแจลับ (Secret Key) นั่นเอง เช่น ผู้ส่งกับผู้รับตกลงจะใช้เทคนิคการแทนที่ตัวอักษรที่อยู่ถัดไป 1 ตำแหน่ง เช่น ถ้าเห็นอักษร A ก็ให้เปลี่ยนไปเป็น B หรือเห็นตัวอักษร B ก็ให้เปลี่ยนไปเป็น C เป็นต้น นั่นก็คือผู้ส่งกับผู้รับตกลงใช้รูปแบบนี้เป็นกุญแจลับ

2. ระบบการเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric-key Cryptography or Public Key Technology) ระบบการเข้ารหัสแบบนี้ได้ถูกคิดค้นโดยนายวิทฟิลด์ ดิฟฟี่ (Whitfield Diffie) ซึ่งเป็นนักวิจัยแห่งมหาวิทยาลัยสแตนฟอร์ด สหรัฐอเมริกา ในปี พ.ศ. 2518 โดยการเข้ารหัสแบบนี้จะใช้หลักกุญแจคู่ทำการเข้ารหัสและถอดรหัส โดยกุญแจคู่ที่กล่าวถึงนั้นประกอบไปด้วย กุญแจ-ส่วนตัว (Private Key) และสุญแจสาธารณะ (Public Key) โดยหลักการทำงานจะเป็นดังนี้ ถ้าใช้กุญแจลูกใดเข้ารหัส ก็ต้องใช้อีกลูกหนึ่งถอดรหัส สำหรับการเข้ารหัสและถอดรหัสด้วยกุญแจคู่นี้จะใช้ฟังก์ชันทางคณิตศาสตร์เข้ามาช่วยโดยที่ฟังก์ชันทางคณิตศาสตร์ที่นำมาใช้ ได้รับการพิสูจน์แล้วว่าจะมีเฉพาะคู่ของมันเท่านั้นที่จะสามารถถอดรหัสได้ ไม่สามารถนำกุญแจคู่อื่นมาถอดรหัสได้อย่างเด็ดขาด

2.2 ลายมือชื่อดิจิตอล (Digital Signature)

ในการส่งข้อมูลผ่านเครือข่ายนั้น นอกจากจะทำให้ข้อมูลที่ส่งนั้นเป็นความลับสำหรับผู้ไม่มีสิทธิ์โดยการใช้เทคโนโลยีการรหัสแล้ว สำหรับการทำนิติกรรมสัญญาโดยทั่วไป ลายมือชื่อจะเป็นสิ่งที่ใช้ในการระบุตัวบุคคล (Authentication) และ ยังมีแสดงถึงเจตนาในการยอมรับเนื้อหาในสัญญานั้นๆซึ่งเชื่อมโยงถึง การป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) สำหรับในการทำธุรกรรมทางอิเล็กทรอนิคส์นั้นจะใช้ลายมือชื่ออิเล็กทรอนิคส์ (Electronic Signature) ซึ่งมีรูปแบบต่างๆเช่น สิ่งที่ระบุตัวบุคคลทางชีวภาพ (ลายพิมพ์นิ้วมือ เสียง ม่านตา เป็นต้น) หรือ จะเป็นสิ่งที่มอบให้แก่บุคคลนั้นๆในรูปแบบของ รหัสประจำตัว ตัวอย่างที่สำคัญของลายมือชื่อ

อิเล็กทรอนิคส์ที่ได้รับการยอมรับกันมากที่สุดอันหนึ่งคือ ลายมือชื่อดิจิตอล (Digital Signature) ซึ่งจะเป็นองค์ประกอบหนึ่งใน โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure, PKI)

รูปที่ 2.1 แผนผังแสดงขอบเขตลายมือชื่อ ลายมือชื่ออิเล็กทรอนิคส์ และลายมือชื่อดิจิตอล

ลายมือชื่อดิจิตอล คือ ข้อมูลอิเล็กทรอนิคส์ที่ได้จากการเข้ารหัสข้อมูลด้วยกุญแจส่วนตัวของผู้ส่งซึ่งเปรียบเสมือนเป็นลายมือชื่อของผู้ส่ง คุณสมบัติของลายมือชื่อดิจิตอล นอกจากจะสามารถ ระบุตัวบุคคล และ เป็นกลไกการป้องกันการปฏิเสธความรับผิดชอบแล้ว ยังสามารถป้องกันข้อมูลที่ส่งไปไม่ให้ถูกแก้ไข หรือ หากถูกแก้ไขไปจากเดิมก็สามารถล่วงรู้ได้ กระบวนการสร้างและ ลงลายมือชื่อดิจิตอลมีขั้นตอนดังนี้

2.2.1 เริ่มจากการนำเอาข้อมูลอิเล็กทรอนิคส์ต้นฉบับที่จะส่งไปนั้นมาผ่านกระบวนการทางคณิตศาสตร์ที่เรียกว่า ฟังก์ชันย่อยข้อมูล (Hash Function) เพื่อให้ได้ข้อมูลที่สั้นๆ ที่เรียกว่าข้อมูลที่ย่อยแล้ว (Digest) ก่อนที่จะทำการเข้ารหัส เนื่องจากข้อมูลต้นฉบับมักจะมีความยาวมากซึ่งจะทำให้กระบวนการเข้ารหัสใช้เวลานานมาก

2.2.2 ทำการเข้ารหัสด้วยกุญแจส่วนตัวของผู้ส่งเอง ซึ่งจุดนี้เปรียบเสมือนการลงลายมือชื่อของผู้ส่งเพราะผู้ส่งเท่านั้นที่มีกุญแจส่วนตัวของผู้ส่งเอง และ จะได้ข้อมูลที่เข้ารหัสแล้ว เรียกว่าลายมือชื่อดิจิตอล

2.2.3 ทำการส่ง ลายมือชื่อไปพร้อมกับข้อมูลต้นฉบับ ไปยังผู้รับ ผู้รับก็จะทำการตรวจสอบว่าข้อมูลที่ได้รับถูกแก้ไขระหว่างทางหรือไม่ โดยการนำข้อมูลต้นฉบับที่ได้รับ มาผ่านกระบวนการย่อยด้วย ฟังก์ชันย่อยข้อมูล จะได้ข้อมูลที่ย่อยแล้วอันหนึ่ง

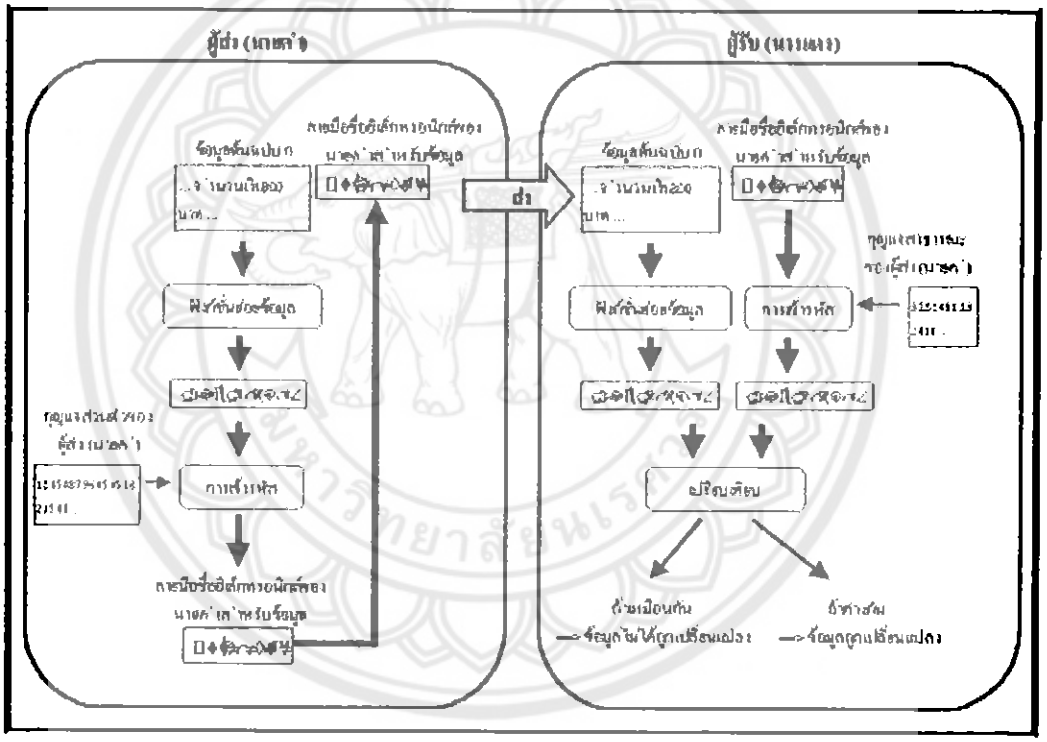
2.2.4 นำลายมือชื่อดิจิตอล มาทำการถอดรหัสด้วย กุญแจสาธารณะของผู้ส่ง ก็จะได้ข้อมูลที่ย่อยแล้วอีกอันหนึ่ง แล้วทำการเปรียบเทียบ ข้อมูลที่ย่อยแล้วทั้งสองอัน ถ้าหากว่า

เหมือนกัน ก็แสดงว่าข้อมูลที่ได้รับนั้นไม่ได้ถูกแก้ไข แต่ถ้าข้อมูลที่ย่อยแล้ว แตกต่างกัน ก็แสดงว่า ข้อมูลที่ได้รับถูกเปลี่ยนแปลงระหว่างทาง

จากกระบวนการลงลายมือชื่อดิจิตอลข้างต้นมีข้อสังเกตดังนี้

1. ลายมือชื่อดิจิตอลจะแตกต่างกันไปตามข้อมูลต้นฉบับและบุคคลที่จะลงลายมือชื่อ ไม่เหมือนกับลายมือชื่อทั่วไปที่จะต้องเหมือนกันสำหรับบุคคลนั้นๆ

2. กระบวนการที่ใช้จะมีลักษณะคล้ายคลึงกับการเข้ารหัสแบบอสมมาตร แต่การ เข้ารหัสจะใช้ กุญแจส่วนตัวของผู้ส่ง และ การถอดรหัสจะใช้ กุญแจสาธารณะของผู้ส่ง ซึ่งสลับกันกับ การเข้าและถอดรหัสแบบกุญแจอสมมาตร ในการรักษาข้อมูลให้เป็นความลับ



รูปที่ 2.2 แสดงการทำงานของ Digital Signature

2.3 อีเมล์ (E-mail)

อีเมล หรือ อีเมล์ (mail, e-mail) ย่อมาจาก "จดหมายอิเล็กทรอนิกส์" หรือ "ไปรษณีย์อิเล็กทรอนิคส์" (electronic mail) คือวิธีการหนึ่งของการเปลี่ยนแปลงข้อความแบบดิจิตอล ซึ่งออกแบบขึ้นเพื่อให้มนุษย์ใช้เป็นหลัก ข้อความนั้นจะต้องประกอบด้วยเนื้อหา ที่อยู่ของผู้ส่งและที่อยู่ของผู้รับ (ซึ่งอาจมีมากกว่าหนึ่ง) เป็นอย่างน้อย บริการอีเมล์บนอินเทอร์เน็ตในทุกวันนี้

เริ่มมีการจัดตั้งมาจาก "อาร์พาเน็ต" (ARPANET) และมีการดัดแปลงโค้ดจนนำไปสู่มาตรฐานของการเข้ารหัสข้อความ RFC 733 อีเมลที่ส่งกันในยุคคริสต์ทศวรรษ 1970 นั้นมีความคล้ายคลึงกับอีเมลในปัจจุบัน การเปลี่ยนจากอาร์พาเน็ตไปเป็นอินเทอร์เน็ตในคริสต์ทศวรรษ 1980 ทำให้เกิดรายละเอียดแบบสมัยใหม่ของการบริการ โดยส่งข้อมูลผ่านเกณฑ์วิธีถ่ายโอนไปรษณีย์อย่างง่าย (SMTP: Simple Mail Transfer Protocol) ซึ่งได้เผยแพร่เป็นมาตรฐานอินเทอร์เน็ต 10 (RFC 821) เมื่อ พ.ศ.2525 (ค.ศ. 1982) และเปลี่ยน RFC 733 ไปเป็นมาตรฐานอินเทอร์เน็ต 11 (RFC 822) การแนบไฟล์มัลติมีเดียเริ่มมีการทำให้เป็นมาตรฐานใน พ.ศ. 2539 (ค.ศ. 1996) ด้วย RFC 2045 ไปจนถึง RFC 2049 และภายหลังก็เรียกกันว่าส่วนขยายสื่อผสมในระบบอินเทอร์เน็ตแบบเอกประสงค์ (MIME)

ระบบอีเมลที่ดำเนินงานบนเครือข่าย มากกว่าที่จะจำกัดอยู่บนเครื่องที่ใช้ร่วมกันเครื่องเดียว มีพื้นฐานอยู่บนแบบจำลองบันทึกและส่งต่อ (store-and-forward model) เครื่องให้บริการอีเมลนั้นจะตอบรับ ส่งต่อ หรือเก็บบันทึกข้อความขึ้นอยู่กับพฤติกรรมของผู้ใช้ โดยที่ผู้ใช้คนนั้นจำเป็นจะต้องเชื่อมต่อกับระบบอีเมลภายในด้วยคอมพิวเตอร์ส่วนบุคคลหรืออุปกรณ์สื่อสารอื่นๆบนเครือข่าย ในการรับส่งข้อความจากเซิร์ฟเวอร์ที่กำหนด ส่วนการส่งอีเมลโดยตรงจากอุปกรณ์สู่อุปกรณ์นั้นพบได้ยากกว่า

รูปแบบของอินเทอร์เน็ตอีเมล กำหนดตามมาตรฐาน RFC 2822 โดยทั่วไปส่วนหัวประกอบด้วยข้อความและตามห้วยเครื่องหมาย ":" และตามด้วยข้อมูล ในแต่ละข้อมูลจะประกอบไปด้วยอย่างน้อย 3 หัวข้อ ได้แก่

2.3.1 จาก: ที่อยู่อีเมลผู้รับ และอาจจะประกอบด้วย ชื่อและนามสกุล

2.3.2 ถึง: ที่อยู่อีเมลผู้รับ และอาจจะประกอบด้วย ชื่อและนามสกุล และสามารถมีได้มากกว่า 1 คน แยกกันด้วยเครื่องหมาย ","

2.3.3 หัวข้อเรื่อง: สรุปเนื้อความของข้อมูล เพื่อให้ผู้รับสามารถเข้าใจเนื้อหาของข้อความคร่าวๆ

2.3.4 เนื้อความ: เนื้อหาของจดหมายที่ผู้ส่งต้องการส่ง

2.4 สแปม (Spam)

สแปม (spam) คือชื่อเรียกของการส่งข้อความที่ผู้รับไม่ได้ร้องขอ ผ่านทางระบบอิเล็กทรอนิกส์ โดยส่วนมากจะทำให้เกิดความไม่พอใจต่อผู้รับข้อความ สแปมที่พบเห็นได้บ่อย

ได้แก่ การส่งสแปมผ่านทางอีเมล์ ในการโฆษณาชวนเชื่อ หรือโฆษณาขายของ โดยการส่ง อีเมลประเภทหนึ่งที่เราไม่ต้องการ ซึ่งจะมาจากทั่วโลก โดยที่เราไม่รู้เลยว่า ผู้ที่ส่งมาให้นั้นเป็นใคร จุดประสงค์คือ ผู้ส่งส่วนใหญ่ต้องการที่จะโฆษณา สินค้าหรือบริการต่าง ๆ ของบริษัทของตนเอง ซึ่งเป็นประเภทหนึ่งของเมล์ขยะ ซึ่งนอกจากจะทำให้ผู้รับรำคาญใจและเสียเวลาในการกำจัดข้อความเหล่านี้แล้ว สแปมยังทำให้ประสิทธิภาพการขนส่งข้อมูลบนอินเทอร์เน็ตลดลงด้วย สแปมในรูปแบบอื่นนอกจาก อีเมล์สแปม ได้แก่ เมสเซนเจอร์สแปม นิวส์กรุ๊ปสแปม บล็อกสแปม และเอสเอ็มเอสสแปม

การส่งสแปมเริ่มแพร่หลายเนื่องจาก ค่าใช้จ่ายในการส่งข้อความผ่านทางระบบอิเล็กทรอนิกส์ มีค่าใช้จ่ายน้อยมากเมื่อเทียบการการส่งข้อความชักชวนทางอื่น เช่นทางจดหมายหรือการโฆษณาทางสื่อต่างๆ ทำให้ผู้ส่งประหยัดค่าใช้จ่ายในการส่งข้อความเชิญชวน และในขณะเดียวกันกฎหมายเกี่ยวกับระบบอิเล็กทรอนิกส์ที่เกี่ยวข้องกับสแปมยังไม่ครอบคลุมจนกระทั่งเริ่มมีใช้ครั้งแรกปี พ.ศ. 2546 (ค.ศ. 2003) ในประเทศสหรัฐอเมริกา

2.5 การป้องกันสแปมเมล์

การกำจัดสแปมเมล์ที่ไม่ต้องการให้หมดแบบถาวรจากอีเมล์ของผู้ใช้ไปเลยนั้นทำได้ยาก แต่มีคำแนะนำในการทำให้สแปมเมล์ในกล่องเก็บอีเมล์ลดลงได้ดังต่อไปนี้

1. ควรใช้โดเมนอินเทอร์เน็ตหรือผู้ให้บริการอีเมล์ที่มีการป้องกันสแปมเมล์ด้วย ซึ่งในความเป็นจริงไม่มีโดเมนใดที่สามารถป้องกันสแปมเมล์ได้ทั้งหมด แต่ก็ยังดีกว่าไม่มีการป้องกันสแปมเมล์

2. ไม่ควรตอบจดหมายอีเมล์ขยะ รวมทั้งไม่ซื้อสินค้าจากโฆษณาที่ผู้ส่งสแปมเมล์ส่งมา หรืออาจจะลบอีเมล์ขยะโดยไม่เปิดอ่านเลย เพราะบางอีเมล์อาจจะบรรจุคุกกี้ (cookies) ไว้ทำให้ผู้ส่งรู้ข้อมูลของผู้รับได้

3. ควรระมัดระวังในการให้ที่อยู่อีเมล์แก่ผู้อื่น

4. เปลี่ยนแปลงที่อยู่อีเมล์เมื่อเข้าไปในเว็บไซต์หรือกลุ่มข่าว เช่น เพิ่มข้อความหน้าที่อยู่อีเมล์ของเรา เพื่อไม่ให้ส่งอีเมล์มาถึงผู้รับได้ "HAPPYyourname@domain.com" เพียงเท่านี้ก็จะทำให้ชื่อของเราไม่อยู่ในบัญชีรายชื่อของผู้ส่งอีเมล์ขยะ

หลักการในการป้องกันสแปมเมล์ในปัจจุบัน

1. **E-mail Client Plug-in** เป็นการลงโปรแกรมที่ทำการตรวจจับสแปมเมล์ลงในเครื่องของผู้ใช้งานทุกเครื่องที่มีความต้องการในการป้องกันสแปมเมล์ วิธีนี้ไม่เหมาะกับองค์กรที่มีขนาดใหญ่มาก เพราะไม่สามารถป้องกันการใช้ Web-Access และ Mobile-Access เพราะ Plug-in จะเริ่มทำงานบนระบบปฏิบัติการคอมพิวเตอร์เท่านั้น

2. **Centralize Filtering Server** เป็นการป้องกันในลักษณะเป็น Single anti-spam Filter เนื่องจากในการตรวจจับสแปม เมล์จะใช้เซิร์ฟเวอร์ของระบบเป็นศูนย์กลางเพียงเครื่องเดียวเท่านั้น การป้องกันด้วยวิธีจึงไม่มีความยืดหยุ่น เนื่องจากต้องเป็นไปตามที่เครื่องหลักขององค์กรตั้งไว้เท่านั้น ไม่สามารถที่จะตั้งค่าเฉพาะเจาะจงได้

3. **Gateway Filtering** ในวิธีนี้จะทำการรับจดหมายอิเล็กทรอนิกส์ทุกฉบับผ่านเข้าสู่ Router และส่งให้ Gateway เพื่อทำการกรองและตรวจจับสแปมก่อนที่จะส่งต่อให้กับเมล์เซิร์ฟเวอร์เพื่อทำการส่งต่อให้กับผู้ใช้งานต่อไป โดยผลของการกรองนี้จะขึ้นอยู่กับการออกแบบโครงสร้างพื้นฐานของระบบเครือข่าย โดยโครงสร้างแบบ P2P (Peer to Peer) เมื่อใช้การตรวจกรองในลักษณะนี้จะทำให้มีความยืดหยุ่นสูง

4. **List-Based Filtering** เป็นวิธีการที่ระบบจะทำการเก็บชื่อของผู้ส่งเมล์ (Sender)ในลักษณะที่เป็น Black List ซึ่งจะอ้างไปกับ Real Time Black List บนระบบ Internet และนำมาเก็บไว้บนฐานข้อมูล เมื่อได้รับจดหมายจากผู้ส่งที่ตรงกับข้อมูลที่มีอยู่ใน Black Listก็จะทำการป้องกันไม่ให้จดหมายฉบับนั้นส่งถึงผู้รับ แต่ในปัจจุบันวิธีนี้มีประสิทธิภาพน้อยลงเนื่องจากสแปมเมอร์มีวิธีการในการหลีกเลี่ยง โดยการเปลี่ยน Address ไปเรื่อยๆ

5. **Bayesian word distribution filters** การตรวจจับแบบ Bayesian ทำงานโดยหาความน่าจะเป็นที่จดหมายฉบับนั้นน่าจะเป็นสแปมเมล์ โดยดูจากคำที่อยู่ในจดหมาย และสร้างรายการของทุกคำที่ปรากฏอยู่ในข้อความอีเมล์ จากนั้นผู้ใช้งานจะเป็นผู้ที่บอกโปรแกรมว่าข้อความนี้ผ่านการพิจารณาหรือไม่ ตัวกรองนี้ จะเพิ่มรายการคำนั้นเข้าไว้ในรายการแยกประเภทจัดไว้เป็นคำว่า "Good" หรือ คำ "Bad" วิธีการเรียนรู้แยกแยะคำไหนดี คำไหนไม่ดี ทำให้มันสามารถปรับตัวเข้ากับสแปมแบบใหม่ๆ และอีเมลใหม่ๆ ที่มีความถูกต้องยอมรับได้ เมื่อใดที่เกิดการทำงานผิดพลาด เราสามารถแก้ไขได้ ข้อเสียคือ เมื่อระบบเจอข้อความที่ไม่เคยถูกอ่านมาก่อนหรือคำเหล่านั้นไม่เคยถูกบันทึกมาก่อนจะทำให้ไม่สามารถตรวจจับได้ และวิธีนี้ยังสิ้นเปลืองหน่วยความจำมากกว่าและใช้เวลาในการจัดเก็บข้อมูลนาน

6. **Cost Based Spam Control** การทำงานของวิธีนี้คือ การทำให้ผู้ส่งนั้นต้องมีค่าใช้จ่ายในการส่งอีเมลแต่ละครั้ง ซึ่งค่าใช้จ่ายที่กล่าวนี้ไม่ได้หมายถึงเงิน แต่เป็นการกระทำการอย่าง เช่น การคำนวณทางคณิตศาสตร์ และการตอบปัญหาอย่างใดอย่างหนึ่ง ฯลฯ ซึ่งวิธีนี้เป็นวิธีการที่ระบบนั้นจะไม่เข้าอ่านข้อความภายใน ทำให้ยังข้อความนั้นยังมีความเป็นส่วนตัวอยู่

2.6 แนวทางในการป้องกันสแปมเมล์

การควบคุมสแปมเมล์มีหลายวิธี

1. การพัฒนานโยบายเกี่ยวกับอีเมล์ และการให้ความรู้แก่ผู้ใช้อินเทอร์เน็ตในองค์กร เพราะถ้าผู้ใช้อินเทอร์เน็ตไม่เอาใจใส่ในนโยบายของหน่วยงานในการกำจัดอีเมล์ขยะแล้ว ก็จะไม่มีทางป้องกันและกำจัดอีเมล์ขยะเหล่านั้นได้เลย

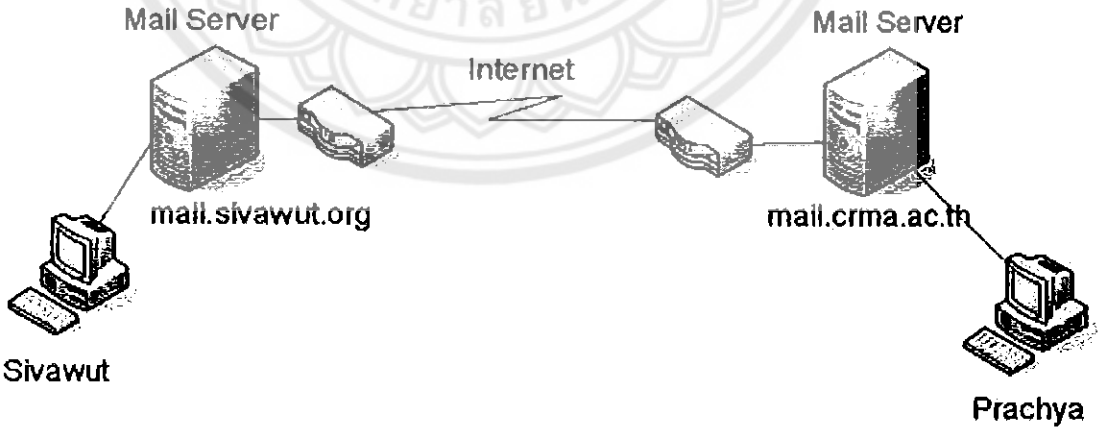
2. ป้องกันอีเมล์ขยะด้วยการติดตั้งเครื่องบริการกรองข้อมูล (Server Filters) ซึ่ง สำนักคอมพิวเตอร์ได้ดำเนินการอยู่เช่นกัน

3. สนับสนุนการออกกฎหมายต่อต้านอีเมล์ขยะโดยตรง เช่น ในสหรัฐอเมริกาก็มี แล้วเป็นต้น

2.7 E-mail Server

E-mail Server คือ เซิร์ฟเวอร์ซึ่งให้บริการรับส่งอีเมล ตัวอย่างโปรแกรมบริการอีเมล เช่น Sendmail, qmail, Microsoft Exchange E-mail Server ซึ่งประกอบด้วย 2 องค์ประกอบหลัก คือ

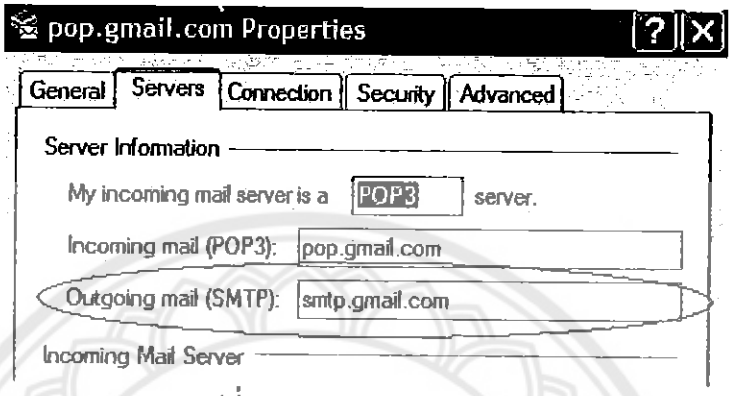
2.7.1 **MTA (Mail Transfer Agent)** ในการส่ง E-mail นั้นจะต้องอาศัย Protocol ที่ชื่อ SMTP (Simple Mail Transfer Protocol) เป็นตัวกลางในการรับและส่ง E-mail แต่ละฉบับ โปรแกรมที่ทำหน้าที่คอยรับการสื่อสารทาง SMTP นั้นมีหลายตัวด้วยกัน เช่น Sendmail, Postfix, qmail, smail, Microsoft Exchange, Lotus Note เป็นต้น โดยปกติ SMTP จะรอรับการติดต่ออยู่ที่ Port 25 อยู่แล้ว



รูปที่ 2.3 การทำงานของ E-mail

เมื่อผู้ใช้งานส่ง E-mail เมื่อผู้ใช้งาน Sivawut บน Domain sivawut.org ต้องการส่ง E-mail ถึง Prachya บนDomain crma.ac.th โดยปกติ Sivawut จะต้องใช้โปรแกรมส่ง E-mail เช่น Outlook Express ทำการส่ง E-mail Address เมื่อกดส่ง โปรแกรมจะเข้าไปดูว่า Outgoing นั้นใช้ช่องทาง

ไหน (Outgoing คือ ข้อมูลที่บอกว่าจะใช้ mail server ที่ไหนเป็นตัวส่งอีเมล) ซึ่งปกติแล้ว Outgoing จะเป็นแอดเดรสของ mail server ที่เรามีบัญชีใช้งานอยู่ ซึ่งในที่นี้ก็คือ mail.sivawut.org นั่นเอง เมื่อเจอ outgoing Mail โปรแกรม Outlook Express ก็จะส่ง mail



รูปที่ 2.4 แสดง Outgoing Mail

เมื่อ mail ถูกส่งถึง Outgoing mail แล้ว Outgoing ก็ดูว่า จดหมายเราจ่าหน้าไปที่ใดอีเมลเซิร์ฟเวอร์ก็จะไปถาม DNS ว่าโดเมนที่จะส่งไป มีอีเมลเซิร์ฟเวอร์ชื่ออะไร จากนั้นก็จะส่งไปที่อีเมล์เซิร์ฟเวอร์ที่จ่าหน้าไปถึง กรณีที่อีเมล์เซิร์ฟเวอร์ที่เราจ่าหน้าไป ไม่มีตัวตนอยู่ หรือเครื่องปิด หรืออาจจะไม่รับอีเมล์จากผู้ส่งที่ไม่รู้จักอีเมล์ฉบับนั้นจะถูกตีกลับมายังผู้ส่ง เมื่ออีเมล์ไปถึงอีเมล์เซิร์ฟเวอร์ปลายทางแล้ว ก็จะเข้าไปอยู่ใน Mailbox ของผู้ใช้แต่ละคน ตามที่จ่าหน้าไปถึง ด้วยหลักการนี้ ทำให้เราสามารถส่งอีเมล์กันถึงได้ทั่วโลกเพียงแค่เชื่อมต่อกับเครือข่ายอินเตอร์เน็ต

2.7.2 MUA (Mail User Agent) เป็นโปรแกรมที่ติดตั้งอยู่ฝั่ง ไคลเอนท์ ที่ผู้ใช้ใช้ทำการเขียนหรืออ่านอีเมล์โดย MUA จะทำการติดต่อกับ MTA ผ่านทางโปรโตคอล POP (Post Office Protocol) หรือ IMAP (Internet Message Access Protocol) เช่นโปรแกรม Microsoft Outlook Express, Mozilla Thunderbird

2.8 Transmission Control Protocol (TCP)

ทีซีพี หรือ TCP มาจากคำว่า Transmission Control Protocol ทีซีพี เป็นหนึ่งในโปรโตคอลหลักในเครือข่ายอินเทอร์เน็ต หน้าที่หลักของทีซีพี คือ ควบคุมการรับส่งข้อมูลระหว่างโฮสถึงโฮสในเครือข่าย เพื่อใช้แลกเปลี่ยนข้อมูลระหว่างกัน โดยตัวโปรโตคอลจะรับประกันความถูกต้อง และลำดับของข้อมูลที่ส่งผ่านระบบเครือข่าย นอกจากนั้นทีซีพียังช่วยจำแนกข้อมูลให้ส่งผ่านไปยังแอปพลิเคชัน ที่ทำงานอยู่บนโฮสเดียวกันให้ถูกต้องด้วย

งานหลักที่สำคัญของทีซีพีอีกงานหนึ่งคือ เป็นโปรโตคอลที่ขั้นกลางระหว่างแอปพลิเคชั่นและเครือข่ายไอพี ทำให้แอปพลิเคชันจากโฮสหนึ่ง สามารถส่งข้อมูลออกยังอีกโฮสหนึ่งผ่านเครือข่ายเปรียบเสมือนมีท่อส่งข้อมูลระหว่างกัน

ทีซีพี เป็นโปรโตคอลที่ได้รับความนิยมที่สุดในโลกของอินเทอร์เน็ต มีแอปพลิเคชั่นจำนวนมากที่ใช้โปรโตคอลทีซีพีเป็นสื่อกลางในการเชื่อมต่อ เช่น เวิลด์ไวด์เว็บ (WWW) เป็นต้น

2.9 MD5 (Message-Digest algorithm 5)

ทั่วไปแล้ว MD5 (Message-Digest algorithm 5) ถูกนำมาใช้เป็นแฮชฟังก์ชัน (hash function) ในการเข้ารหัสลับ (Cryptography) อย่างแพร่หลาย เช่นการเก็บรหัสพาสเวิร์ด และนอกจากนี้ยังมีการนำมาใช้ในการตรวจสอบความสมบูรณ์ของไฟล์ (Md5sum) แต่ก็มีการพบว่า MD5 นั้นไม่เป็นแฮชฟังก์ชันที่ ป้องกันการทับซ้อน (collision resistant) จึงไม่เหมาะสมที่จะนำมาใช้ในแอปพลิเคชันบางอย่างเช่น SSL หรือ Digital Signature ซึ่ง MD5 เป็นการเข้ารหัสทางเดียวโดยใช้คีย์ในการเข้ารหัสขนาด 128 บิต (16 ไบต์) และหลังจากการเข้ารหัสแล้วจะได้เป็นตัวอักษร ACSII ขนาด 32 ตัวอักษร

2.9.1 อัลกอริทึม

เราจะเริ่มจากการสมมติให้มีอินพุทเป็นข้อความ M ขนาด b-bit ซึ่ง b เป็นจำนวนเต็มที่ไม่ติดลบ ไม่จำเป็นต้องหาร 8 ลงตัว และมีความยาวได้ไม่จำกัด ซึ่งจะเขียนใหม่ได้เป็น m_0 m_1 m_2 ... m_b-1 จากนั้นจะทำการหา MD5 โดยการผ่านขั้นตอนต่อไปนี้

- เติมบิทท้าย เติม " 10 " ท้ายข้อความแล้ว เติม " 0 " ไปเรื่อยๆ จนกว่าข้อความจะมีขนาดที่คอนกลูเอนกับ 448 (mod 512)

- เติมขนาดข้อความ เติมขนาดของข้อความความยาว 64 บิต ท้ายข้อความหากขนาดของข้อความใหญ่เกินที่ 64 บิตจะเก็บได้ก็ให้ใช้ 64 บิตหลังของขนาดเท่านั้น สุดท้ายจะได้ข้อความที่แต่งเติมแล้วมีขนาดที่สามารถหาร 512 ลงตัวพอดี นั่นคือจะสามารถแบ่งข้อความได้เป็นชุด ชุดละ 512 บิต หรือ 32 ไบต์ หรือ 16-word block

- กำหนดค่าเริ่มต้นของ MD Buffer ตั้งค่าเริ่มต้นของ buffer ขนาด 32 บิต 4 ตัวดังนี้ A = 0x67452301 , B = 0xEFCDAB89 , C =0x98BADCFE , D = 0x76543210

- คำนวณข้อความใน 10-word block ลำดับแรกเราจะกำหนดฟังก์ชันรับอินพุท 32 บิต และเอาต์พุต 32 บิต ดังนี้

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ แทนการคำเนินการ XOR, AND, OR และ NOT ในขั้นตอนนี้ยังต้องใช้ตารางขนาด 64 ช่อง T[1...64] ซึ่ง T[i] สามารถหาค่าได้จาก ⌊ 2^32 x |sin [illegible] | ⌋ โดย i มีค่าเป็นเรเดียน จากนั้นจะทำการเข้าอัลกอริทึมต่างๆจนได้เป็นค่ารหัสเป็นเลขฐาน 16 จำนวน

2.10 SHA1 (Secure Hash Algorithm 1)

SHA1 เป็นแฮชฟังก์ชันประเภทนึ่งที่ได้รับความนิยมในปัจจุบันมาก มีลักษณะการเข้ารหัสคล้ายคลึงกับ MD5 แต่มีการออกแบบให้มีการป้องกันที่ดีกว่า MD5 โดยมีการใช้คีย์ในการเข้ารหัสขนาด 160 บิต ซึ่งทำให้รหัสที่ได้มีความซับซ้อนมากขึ้นและยากต่อการคาดเดามากขึ้น และความเป็นไปได้ในการชนกัน (Collision) ของ Hash data ที่มาจากวิธีการเข้ารหัสแบบ SHA1 จะไม่มีความเป็นได้เลยทั้งในทางทฤษฎีและทางปฏิบัติ การเข้ารหัสแบบ SHA1 นี้ นิยมนำมาใช้ในแอพพลิเคชั่นต่างๆ รวมไปถึงการรักษาความปลอดภัยบน Protocol อีกด้วย จุดประสงค์ของ SHA1 คือทำการเตรียมกุญแจพื้นฐานเพื่อทำการแฮชในการเข้ารหัสกุญแจสาธารณะ โดยเฉพาะรูปแบบของ RSA ซึ่ง SHA1 คือแฮชฟังก์ชันทางเดียวโดยขั้นแรกของ SHA1 จะทำการทำลักษณะเฉพาะ คือเตรียมกระบวนการให้ง่ายในการคำนวณแฮชฟังก์ชันจากข้อมูลบางส่วนที่ยากในการกำหนดค่าข้อมูลจากคำนวณในการแฮชซึ่งข้อดีของการแฮชแบบ SHA1 นั้นมี 3 ข้อที่เด่นชัดดังนี้

- SHA-1 คือ หนึ่งวิธีที่ให้ค่า SHA-1(x) ซึ่งค่า x เป็นค่าที่การยากในการค้นหา x
- SHA-1 เป็นการป้องกันความขัดแย้งของข้อมูล ซึ่งยากในการหาค่า x,y โดยที่ x != y แต่ในขณะเดียวกัน SHA-1 (x) = SHA-1 (y)
- SHA-1 มีลักษณะการเข้ารหัสแบบทางเดียว ซึ่งทำให้การคิดย้อนกลับ

SHA-1 เป็นประโยชน์ในการจัดเตรียมคุณลักษณะเฉพาะหรือค่าเฉพาะของข้อความที่มีการแฮชโดยที่ทำให้ไม่จำเป็นในการเข้ารหัสเป็นเวลานาน ซึ่งตัวอย่างของการเข้ารหัสแบบ SHA1 มีดังนี้

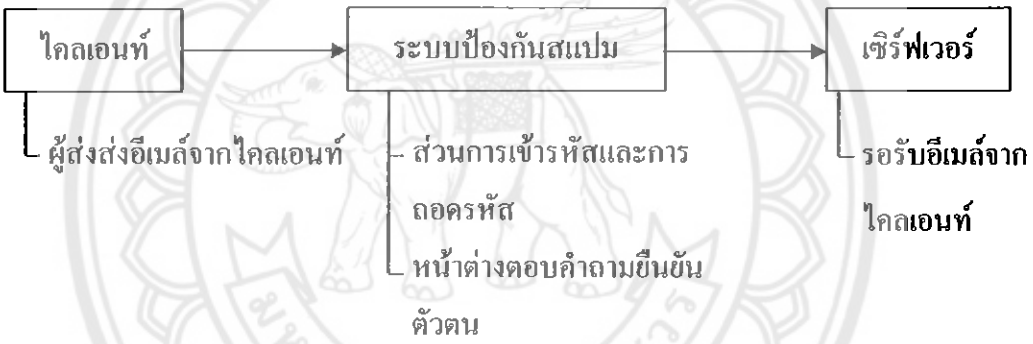
ถ้าให้ข้อความต้นฉบับคือ "The quick brown fox jumps over the lazy dog" เมื่อผ่านกระบวนเข้ารหัสแบบ SHA1 แล้ว ผลลัพธ์ที่ได้จะเป็นเลขฐาน 16 จำนวน 40 ตัวอักษรก็คือ "a75e1658c2f4c9eba530c0fc4b0a60c65e7c22fb" แต่ถ้าเปลี่ยนจากคำว่า dog เป็นคำว่า cog แล้ว

ก็จะให้ผลดังนี้ "6197aa89aebcf4de2730f3c8cd51096b3ea35a6f" ซึ่งแสดงให้เห็นว่าการเปลี่ยนตัวอักษรเพียงตัวอักษรเดียวในข้อความ ผลลัพธ์ที่ได้ก็จะมีค่าต่างกันมากถึง 81 บิต จาก 160 บิต ซึ่งเป็นข้อดีของการเข้ารหัสแบบ SHA1

บทที่ 3
ขั้นตอนและวิธีการดำเนินงาน

หลังจากที่ได้มีการศึกษาเกี่ยวกับทฤษฎีที่เกี่ยวข้องแล้ว เพื่อให้ตรงตามจุดประสงค์ของการจัดทำโครงงาน "การออกแบบระบบป้องกันสแปม" จึงมีการออกแบบส่วนต่างๆ ออกได้ดังนี้

- การจำลองการรับส่งข้อมูลระหว่างผู้ส่งกับเซิร์ฟเวอร์
- การสร้างระบบเพื่อป้องกันสแปมเมล์
- การออกแบบส่วนเข้ารหัสและการถอดรหัส
- ออกแบบส่วนติดต่อผู้ใช้ของระบบ



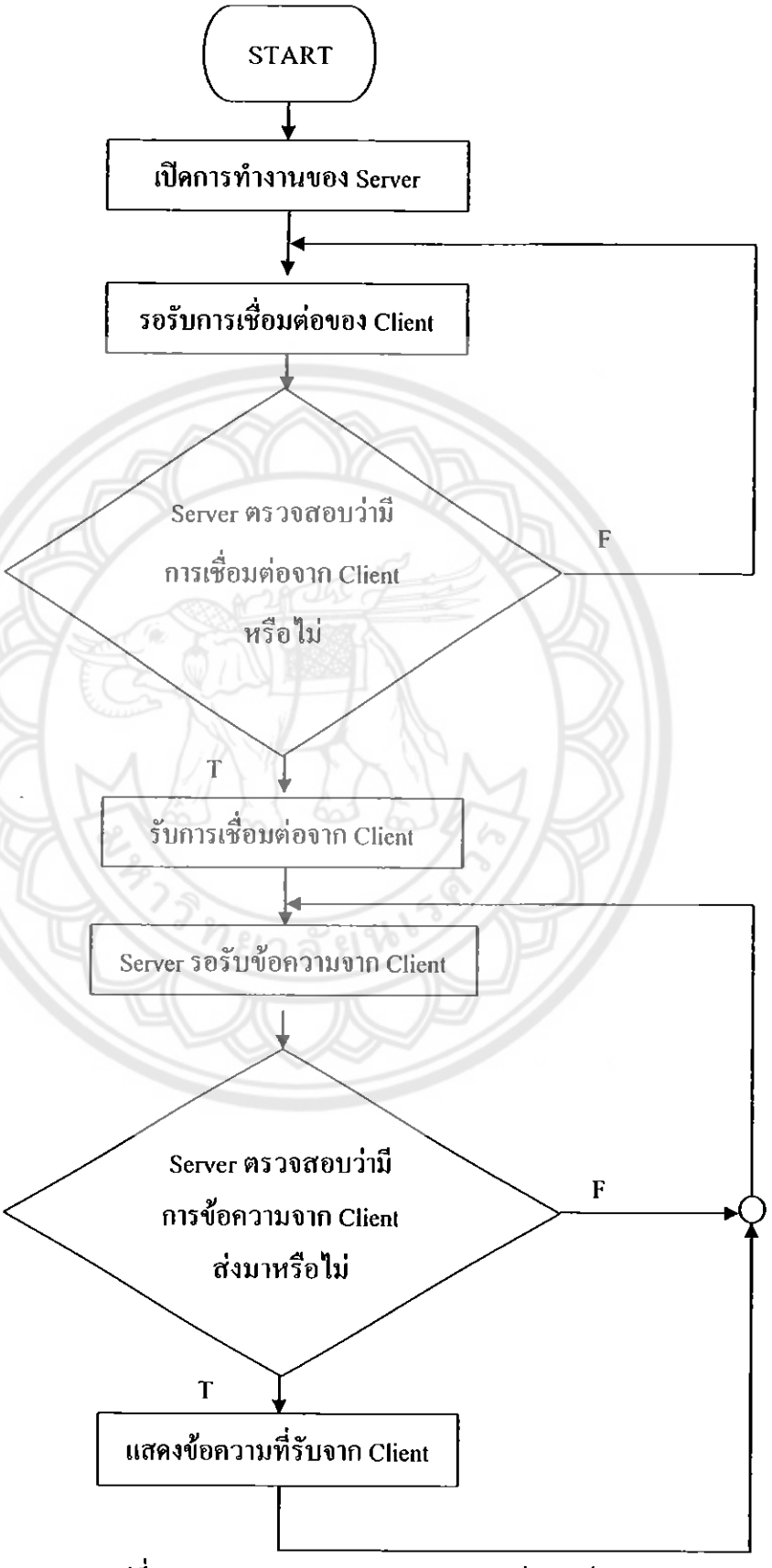
รูปที่ 3.1 แผนภาพแสดงองค์ประกอบโดยรวมของระบบ

จากการทำงานสามารถแสดงรายละเอียดได้ดังนี้

3.1 การจำลองการส่งและการรับข้อมูลระหว่างผู้ส่งและเซิร์ฟเวอร์

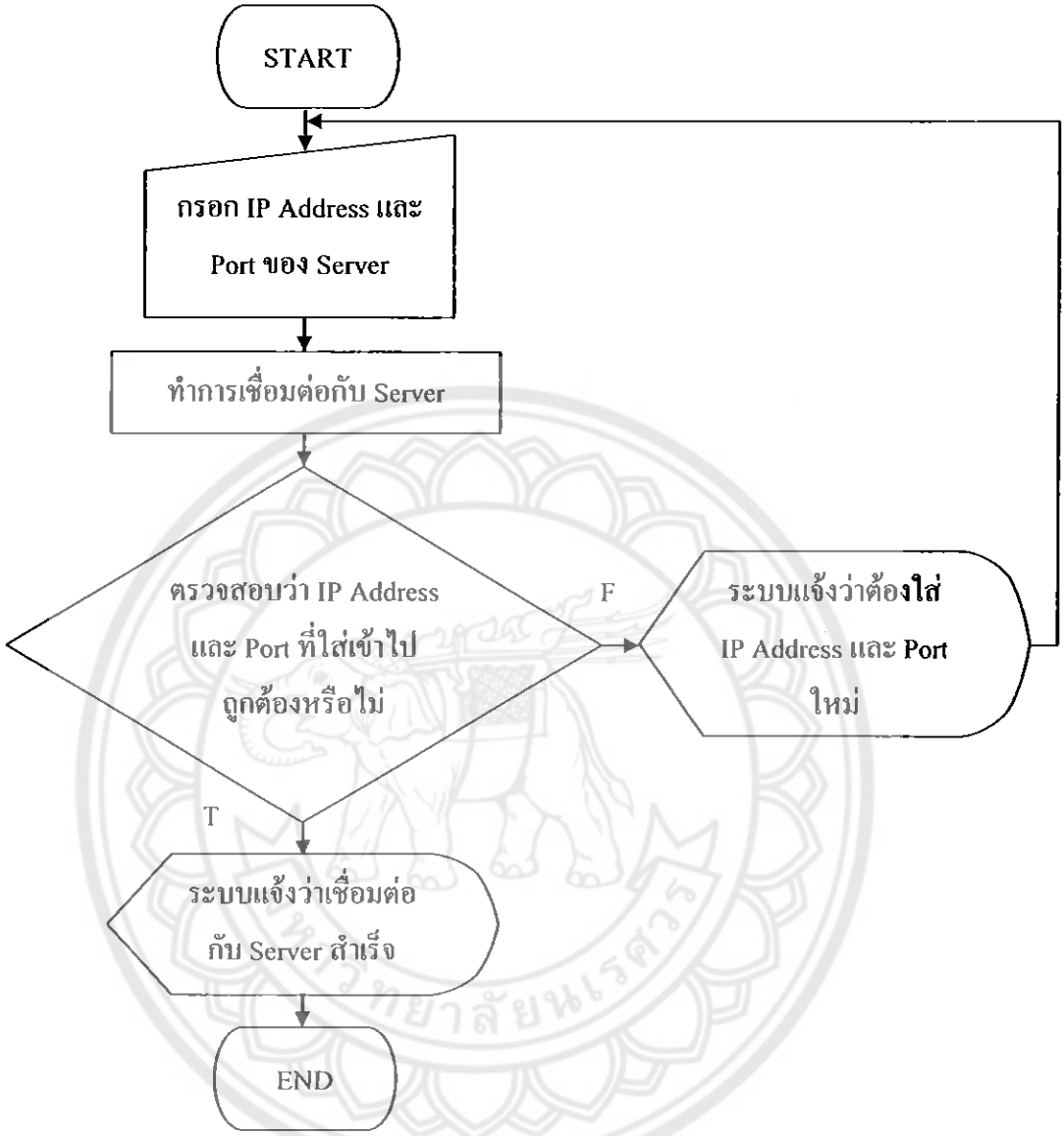
การในจำลองส่วนของการรับส่งข้อมูลระหว่างผู้ส่งและเซิร์ฟเวอร์นั้น ผู้จัดทำได้เขียนโปรแกรมเพื่อทำการติดต่อระหว่างไคลเอนท์และเซิร์ฟเวอร์ โดยที่ทั้งไคลเอนท์และเซิร์ฟเวอร์สามารถเชื่อมต่อกัน และสามารถส่งข้อความถึงกันได้

3.1.1 การทำงานของเซิร์ฟเวอร์



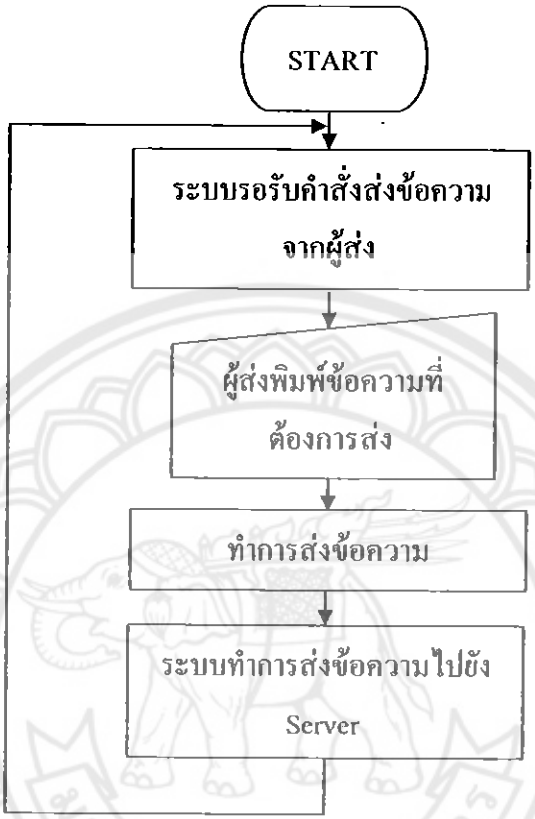
รูปที่ 3.2 แผนผังแสดงการทำงานของเซิร์ฟเวอร์

3.1.2 การทำงานของไคลเอนท์เมื่อทำการเชื่อมต่อกับเซิร์ฟเวอร์



รูปที่ 3.3 แผนผังแสดงการทำงานของไคลเอนท์ขณะที่ทำการเชื่อมต่อกับเซิร์ฟเวอร์

3.1.3 การทำงานของไคลเอนท์ที่ทำการส่งข้อความไปยังเซิร์ฟเวอร์ เมื่อมีการเชื่อมต่อและยังไม่ได้ใส่ระบบป้องกันสแปมเข้าไป



รูปที่ 3.4 แผนผังแสดงการทำงานของไคลเอนท์เมื่อส่งข้อความไปยังเซิร์ฟเวอร์

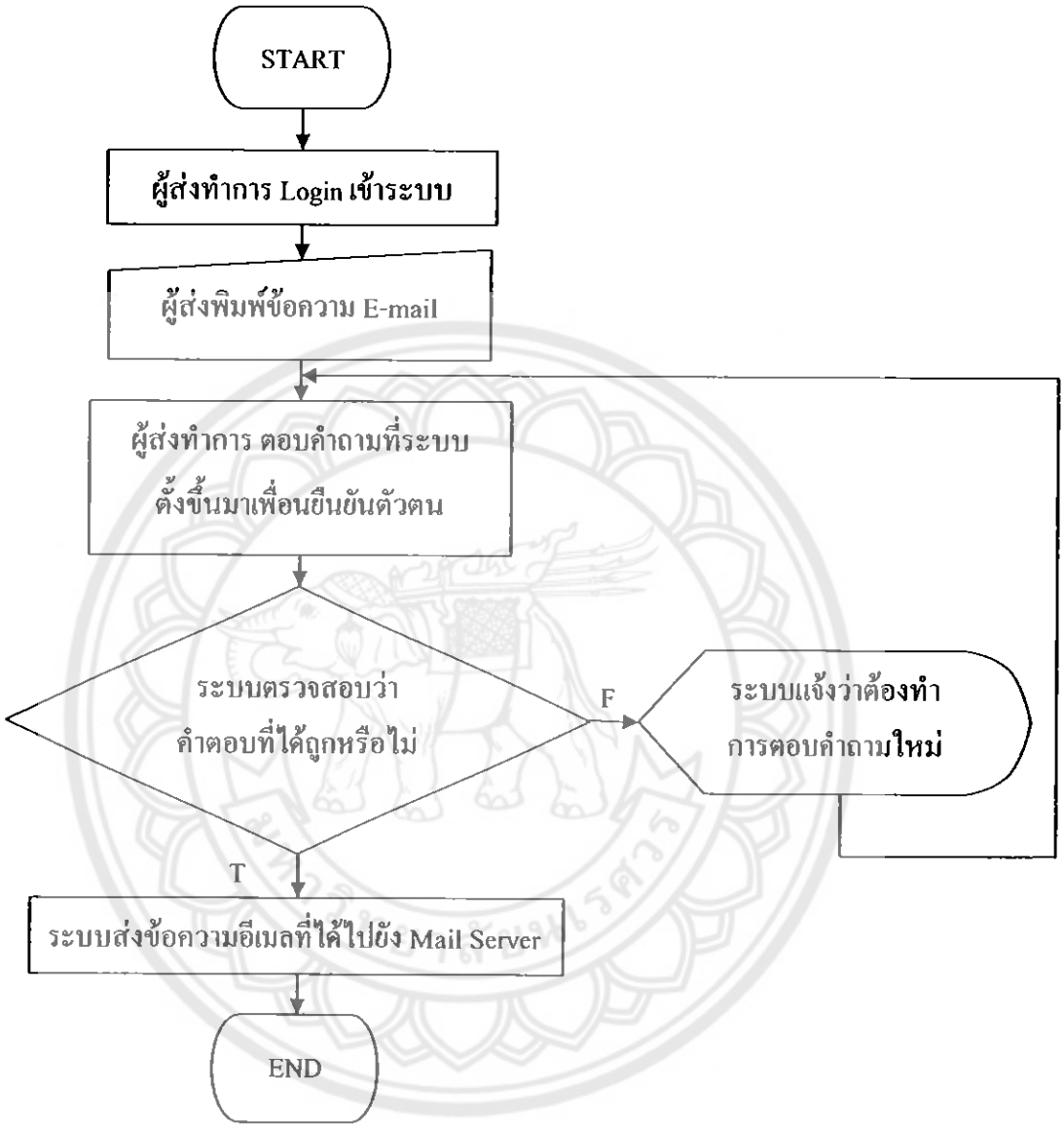
3.2 การสร้างระบบเพื่อป้องกันสแปมเมล์

3.2.1 การวิเคราะห์การทำงานของโปรแกรม

จากการศึกษาเทคนิคในการป้องกันสแปมรูปแบบต่างๆแล้ว ผู้จัดทำได้เลือกเทคนิค Cost Based Spam Control ในลักษณะของ Proof Of Work ซึ่งเป็นวิธีการที่กระทำในส่วนของผู้ส่ง คือ การให้ผู้ส่งนั้นต้องมีการทำงานอย่างใดอย่างหนึ่งก่อนที่จะส่งอีเมล์ไปยังผู้รับ โดยโปรแกรมจะนี้จะไปอยู่ระหว่างเครื่องของผู้ส่งและอีเมล์เซิร์ฟเวอร์เพื่อให้อีเมลทุกฉบับที่ทำการส่งนั้นผ่านการตรวจสอบว่าได้กระทำการเพื่อเป็นค่าใช้จ่ายในการส่งอีเมลแล้วหรือยัง โดยกำหนดเป็นเงื่อนไขว่าหากสามารถผ่านเงื่อนไขที่กำหนดไว้ได้ ก็สามารถส่งข้อความอีเมล์ได้ แต่ถ้าไม่สามารถผ่านเงื่อนที่กำหนดไว้ได้ ก็จะไม่สามารถส่งข้อความอีเมล์ได้

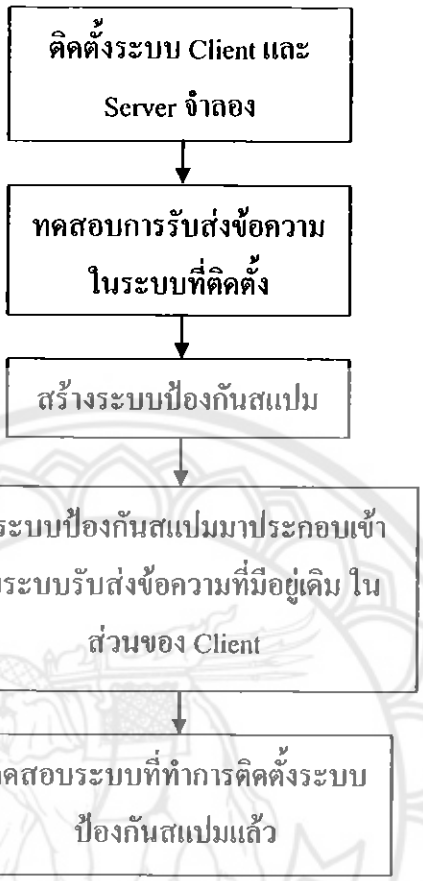
3.2.2 การออกแบบโปรแกรม

- การทำงานของระบบป้องกันและลดจำนวนสแปมเมล์



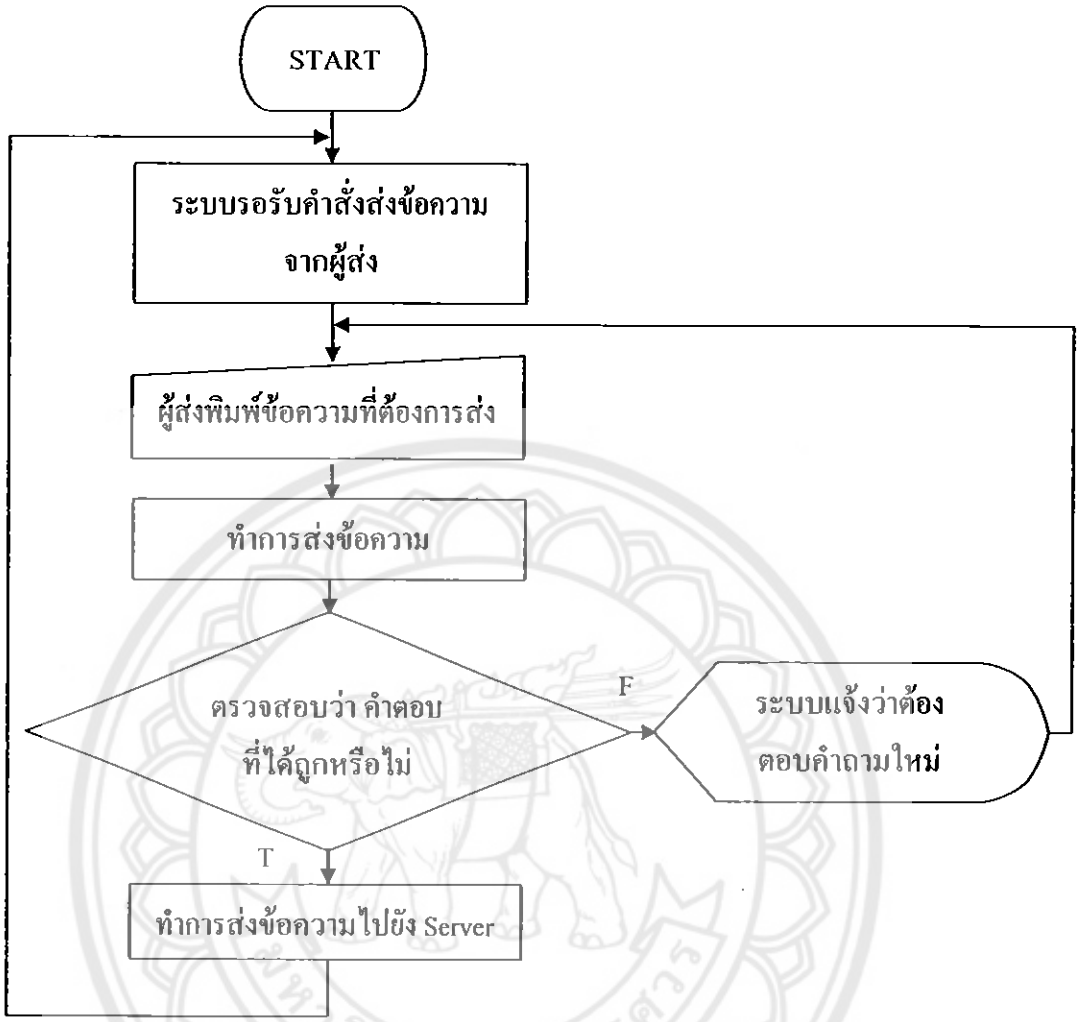
รูปที่ 3.5 แผนผังแสดงการทำงานของระบบป้องกันและลดจำนวนสแปมเมล์

3.2.3 การสร้างระบบป้องกันสแปม



รูปที่ 3.6 แผนผังแสดงขั้นตอนการสร้างโปรแกรม

3.2.4 การทำงานของระบบป้องกันสแปมที่ถูกติดตั้งลงบนฝั่งไคลเอนท์ของระบบแล้ว



รูปที่ 3.7 แผนผังแสดงการทำงานของระบบป้องกันสแปมเมื่อติดตั้งบนฝั่งไคลเอนท์แล้ว

3.2.5 การทดสอบระบบ

การทดสอบระบบนั้นจะทำการทดสอบโดยการส่งข้อความให้ผ่านไปยังเซิร์ฟเวอร์และทดสอบการตอบคำถามที่ระบบนั้นสร้างขึ้นเพื่อเป็นเงื่อนไขในการส่งข้อความ โดยกำหนดว่าถ้าสามารถใส่คำตอบที่ถูกต้องลงไปในระบบได้ ข้อความก็จะถูกส่งไปยังเซิร์ฟเวอร์ แต่ถ้าไม่สามารถใส่คำตอบที่ถูกต้องลงไปในระบบได้ ข้อความก็จะไม่ถูกส่งไปยังเซิร์ฟเวอร์และจะต้องทำการตอบคำถามใหม่อีกครั้งหนึ่ง จึงจะถือว่าโปรแกรมสามารถใช้งานได้

3.3 การออกแบบส่วนการเข้ารหัส

ในส่วนของการออกแบบการเข้ารหัสเพื่อสร้างคำถามให้ผู้ส่งนั้น แบ่งออกเป็นสองส่วน คือ

1. ส่วนที่ใช้การเข้ารหัสแบบ MD5
2. ส่วนที่ใช้การเข้ารหัสแบบ SHA1

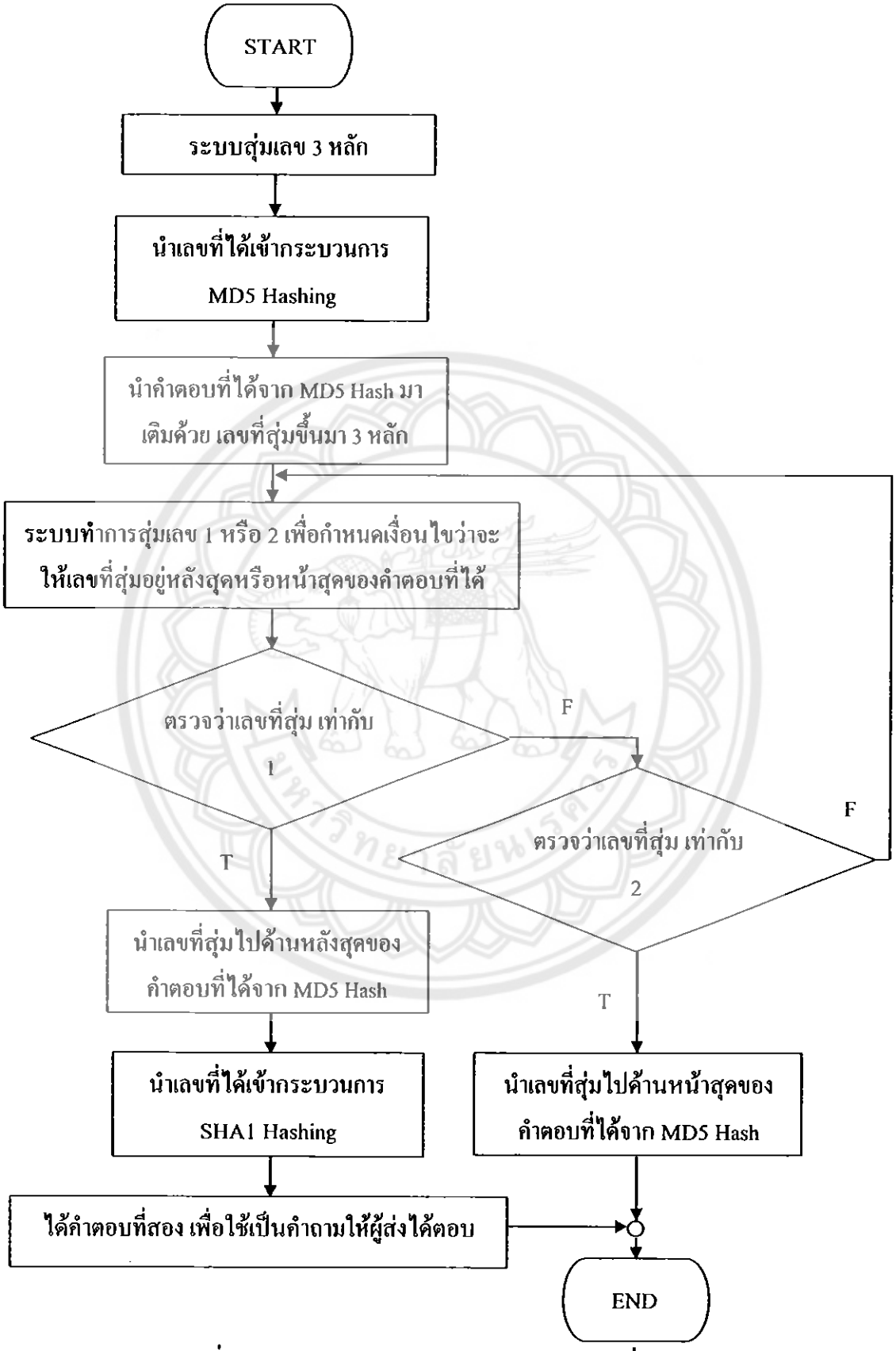
และในส่วนของขั้นตอนการทำงาน สามารถอธิบายรายละเอียดได้เป็นขั้นตอนดังต่อไปนี้

ขั้นตอนแรกของการเข้ารหัส จะใช้วิธีการให้โปรแกรมสุ่มตัวเลข 3 หลักขึ้นมา 1 ชุด เพื่อใช้เป็นข้อความตั้งต้นในการเข้ารหัส หลังจากได้ตัวเลขที่สุ่มมาเรียบร้อยแล้ว ก็จะนำเลขดังกล่าวมาทำการเข้ารหัสด้วยวิธี MD5 Hashing และแสดงคำตอบที่ได้ ซึ่งคำตอบที่ได้นั้นจะเป็นอักขระ (ตัวอักษรภาษาอังกฤษ หรือ ตัวเลข 0-9) จำนวน 32 อักขระ โดยถือว่าผลลัพธ์ที่ได้จากขั้นตอนนี้เป็นคำตอบที่ 1 ซึ่งจะใช้เป็นข้อความตั้งต้นของการคำนวณต่อไป ตัวอย่างเช่น ระบบสุ่มเลข 3 หลักได้เลข "381" เมื่อทำการเข้ารหัสด้วยวิธีการ MD5 Hashing แล้ว คำตอบที่ได้ก็คือ "00ec53c4682d36f5c4359f4ae7bd7ba1"

ขั้นตอนที่สอง คือการนำคำตอบที่ได้นั้นมาเพิ่มตัวเลข 3 หลักอีกหนึ่งชุด ซึ่งตัวเลข 3 หลักดังกล่าวก็จะได้มาจากการที่ระบบสุ่มตัวเลขขึ้นมาอีกครั้งหนึ่งเพื่อเพิ่มเข้าไปในคำตอบที่ 1 สำหรับใช้เป็นข้อความตั้งต้นในการเข้ารหัสแบบ SHA1 ในขั้นตอนต่อไป โดยที่โปรแกรมก็จะทำการสุ่มอีกเช่นเดียวกันว่า เลข 3 หลักที่กำหนดขึ้นมานั้น จะไปอยู่ที่ตำแหน่งหน้าสุดของคำตอบ หรือไปอยู่ที่ตำแหน่งสุดท้ายของคำตอบ โดยในที่นี้กำหนดว่า ถ้าเลขที่สุ่มได้เป็นเลข 1 ตัวเลข 3 หลักที่สุ่มได้นั้นจะต้องไปอยู่ที่ตำแหน่งท้ายสุดของคำตอบที่ 1 และถ้าเลขที่สุ่มได้คือเลข 2 ตัวเลข 3 หลักที่สุ่มได้ในครั้งที่ 2 นั้นจะต้องไปอยู่ที่ตำแหน่งหน้าสุดของคำตอบที่ 1 ตัวอย่างเช่น จากคำตอบที่ได้จากกระบวนการเข้ารหัสแบบ MD5 คือ "00ec53c4682d36f5c4359f4ae7bd7ba1" และตัวเลขที่สุ่มได้ในครั้งที่สองคือเลข "121" โดยที่ตำแหน่งที่สุ่มได้นั้นคือ "2" หมายความว่า เลข "121" นี้จะต้องไปอยู่ที่ตำแหน่งหน้าสุดของคำตอบที่ 1 ข้างต้น ซึ่งผลลัพธ์ที่ได้จะเป็นดังนี้ คือ "12100ec53c4682d36f5c4359f4ae7bd7ba1" โดยส่วนที่ขีดเส้นใต้นั้นคือ ตัวเลข 2 หลักที่สุ่มขึ้นมาในครั้งที่สอง และถูกสุ่มให้ไปอยู่ที่ตำแหน่งหน้าสุดของคำตอบที่ 1 นั่นเอง

ขั้นตอนที่สาม คือการนำคำตอบที่เพิ่มตัวเลข 3 หลักไปแล้วมาทำการเข้ารหัสโดยวิธี SHA1 Hashing และแสดงคำตอบที่ได้ซึ่งคำตอบนั้นจะเป็นอักขระ (ตัวอักษรภาษาอังกฤษ หรือ ตัวเลข 0-9) จำนวน 40 อักขระ และถือว่าคำตอบที่ได้นี้จะเป็นคำตอบที่ 2 และใช้เป็นโจทย์ให้กับผู้ส่งต่อไป ตัวอย่างเช่น จากผลลัพธ์ในขั้นตอนที่สองคือ "12100ec53c4682d36f5c4359f4ae7bd7ba1" เมื่อนำมาเข้ารหัสด้วยวิธี SHA1 Hashing เรียบร้อยแล้ว เราจะได้คำตอบที่ 2 ซึ่งจะใช้เป็นโจทย์ คือ "42d5b37fa4aef07a824bcaee23483a692bce8eec"

3.3.1 การทำงานในส่วนของการเข้ารหัสเพื่อสร้างโจทย์



รูปที่ 3.8 แผนผังแสดงการทำงานของการเข้ารหัสเพื่อสร้างเป็นโจทย์

3.4 การออกแบบส่วนติดต่อผู้ใช้ของระบบ

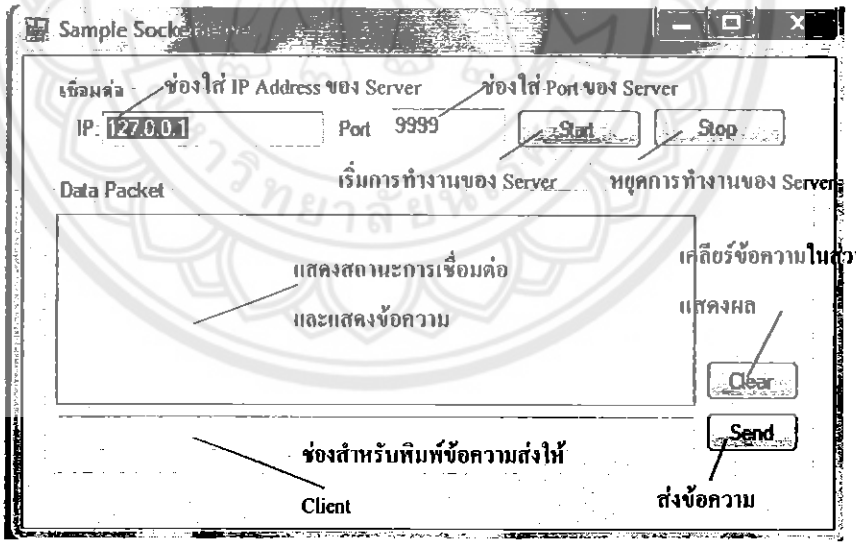
ในส่วนการติดต่อกับผู้ใช้นั้น ได้ออกแบบและพัฒนาในภาษา Visual C# ซึ่งมีคำสั่งที่ใช้ในการเขียนโปรแกรมติดต่อกับ Socket (Socket Programming) ได้และมีคำสั่งที่สามารถใช้งานการเข้ารหัสแบบ MD5 และ SHA1 ได้ ซึ่งได้กำหนดฟังก์ชันการใช้งานไว้ดังนี้

- สามารถทำการเชื่อมต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้
- สามารถส่งข้อความระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้
- การส่งข้อความจากไคลเอนท์ ไปยัง เซิร์ฟเวอรเซิร์ฟเวอร์ จะต้องมีการตอบคำถามที่ระบบสร้างขึ้น เพื่อที่จะสามารถส่งข้อความได้
- ถ้าไม่สามารถตอบคำถามได้ ก็จะไม่สามารถส่งข้อความได้จนกว่าจะสามารถตอบคำถามได้ถูกต้อง โดยคำถามจะต้องถูกสร้างขึ้นใหม่ทุกครั้งที่มีการส่งข้อความ

ซึ่งในการออกแบบส่วนติดต่อกับผู้ใช้ ผู้จัดทำได้ออกแบบให้โปรแกรมนี้มีทั้งหมด 3 หน้าต่าง คือ

3.4.1 หน้าต่างของเซิร์ฟเวอร์

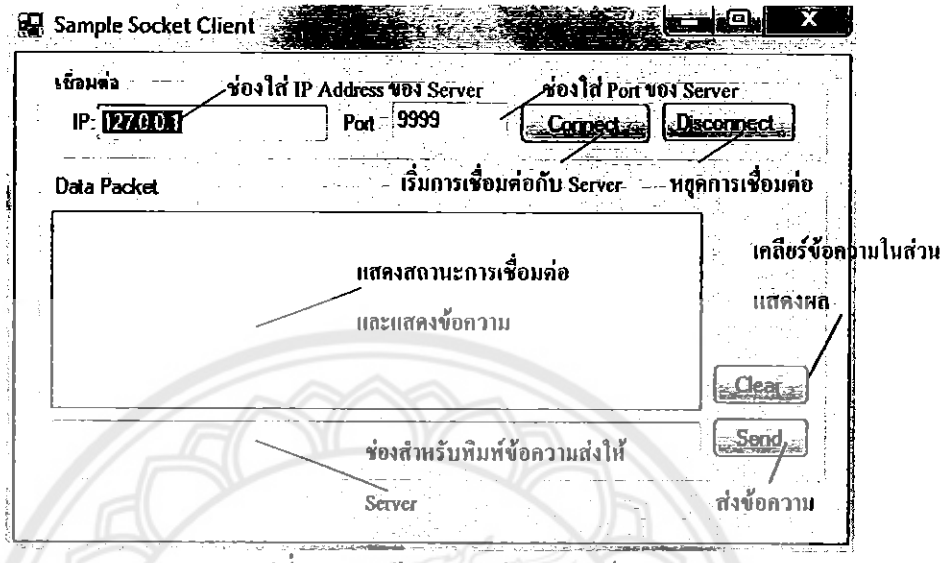
ในหน้าต่างของเซิร์ฟเวอร์มีรูปแบบดังรูป



รูปที่ 3.9 หน้าต่างของเซิร์ฟเวอร์

3.4.2 หน้าต่างของไคลเอนท์

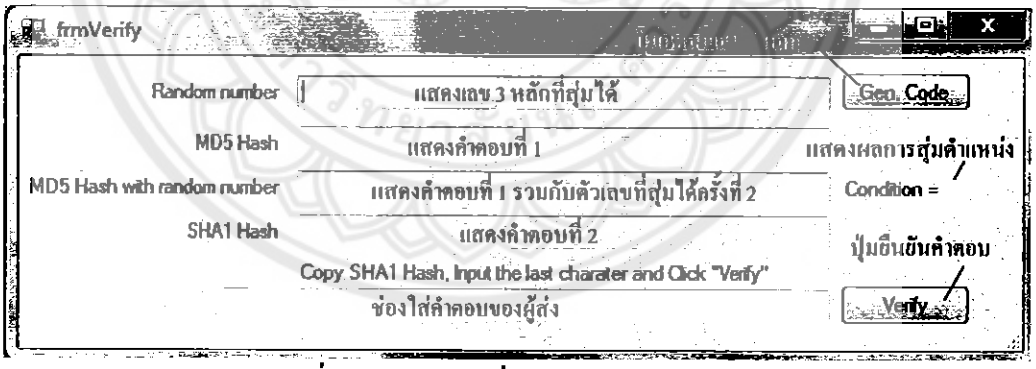
ในหน้าต่างของไคลเอนท์มีรูปแบบดังรูป



รูปที่ 3.10 หน้าต่างของไคลเอนท์

3.4.3 หน้าต่างที่ผู้ส่งใช้ตอบคำถามของระบบ

หน้าต่างของการตอบคำถามเพื่อยืนยันตัวตนเป็นดังรูป



รูปที่ 3.11 หน้าต่างที่ใช้ตอบคำถามของผู้ส่ง

หลังจากที่ได้ออกแบบในส่วนประกอบต่างๆของโปรแกรมเป็นที่เรียบร้อยแล้ว ในขั้นตอนต่อไปจึงเป็นการทดสอบการทำงานของระบบป้องกันสแปม เพื่อเป็นการศึกษาข้อดีและข้อเสียของระบบที่ทำขึ้นว่าระบบที่สร้างขึ้นสามารถบรรลุตามวัตถุประสงค์หรือไม่ ซึ่งการทดสอบระบบดังกล่าวจะได้กล่าวถึงต่อไปในบทที่ 4

บทที่ 4

ผลการทดลองและการวิเคราะห์ผล

การทำทดสอบการทำงานของโปรแกรมที่ได้จัดทำขึ้นนั้นเป็นสิ่งที่สำคัญ เพราะเป็นตัวบ่งชี้ว่าโปรแกรมที่ได้พัฒนาขึ้นมานั้น จะสามารถนำมาประยุกต์ใช้ได้จริงหรือไม่ ซึ่งการทดสอบในแต่ละส่วนนั้นมีดังนี้

ส่วนที่ 1 การทดลองการเชื่อมต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์

ส่วนที่ 2 การทดลองการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยที่ยังไม่มี ส่วนที่ให้ผู้ส่งตอบคำถามยืนยันตัวตน

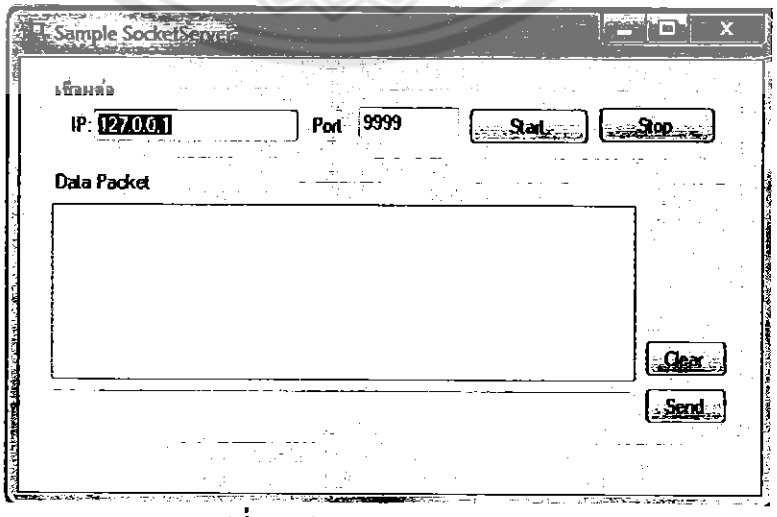
ส่วนที่ 3 การทดลองการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยที่มีส่วนที่ ให้ผู้ส่งตอบคำถามยืนยันตัวตน ในกรณีที่จะสามารถตอบคำถามได้ถูกต้อง

ส่วนที่ 4 การทดลองการเชื่อมต่อและการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยที่มีส่วนที่ให้ผู้ส่งตอบคำถามยืนยันตัวตน ในกรณีที่ไม่สามารถตอบคำถามได้ถูกต้อง

4.1 การทดลองการเชื่อมต่อเซิร์ฟเวอร์ระหว่างและไคลเอนท์

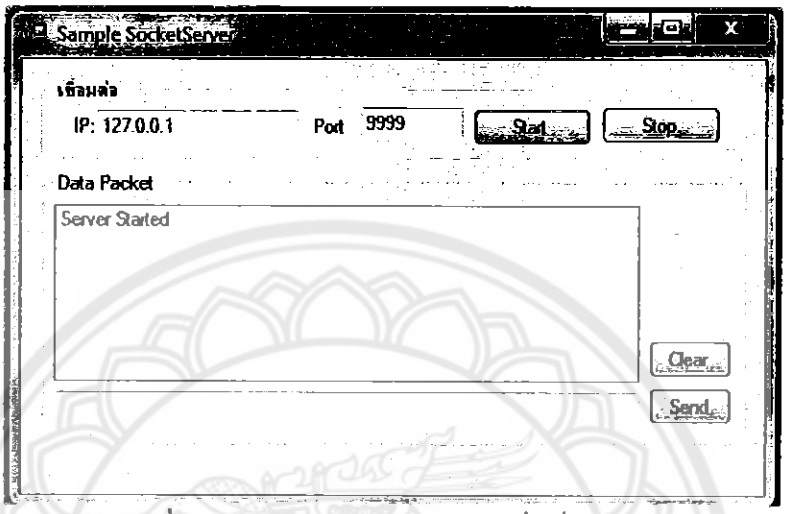
เป็นการทดลองการทำงานของโปรแกรมว่าสามารถเชื่อมต่อและส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์ได้หรือไม่โดยยังไม่มีการตอบคำถามยืนยันตัวตน ซึ่งเป็นการทดลองขั้นแรกก่อนที่จะเพิ่มส่วนที่ให้ผู้ส่งนั้นตอบคำถาม กล่าวคือ เป็นการทดลองการส่งข้อความโดยที่ยังไม่มีเงื่อนไขใดๆนั่นเอง ซึ่งขั้นตอนการทดลองดังนี้

เปิดโปรแกรม SocketServer ขึ้นมาดังรูป



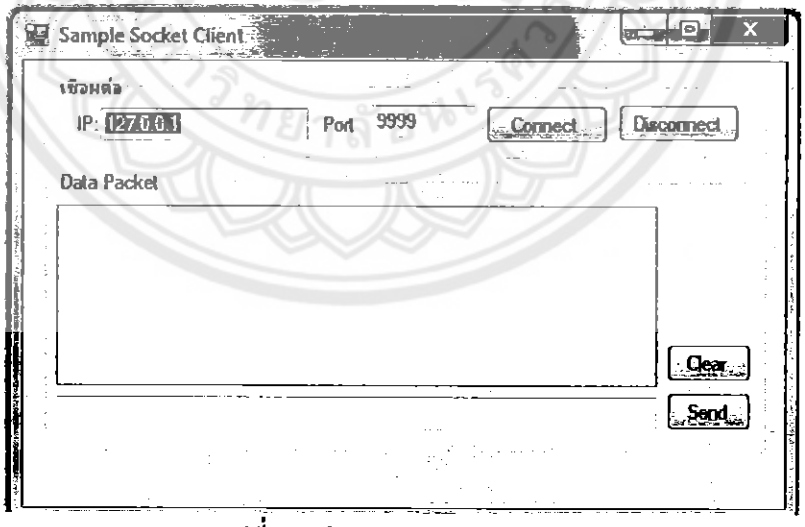
รูปที่ 4.1 โปรแกรม SockerServer

หลังจากเปิดโปรแกรม SocketServer แล้ว ให้กำหนด IP Address และ Port ขึ้นมา โดยกำหนด IP Address คือ 127.0.0.1 และกำหนด Port เป็น 9999 หลังจากนั้นคลิกที่ Start เพื่อเป็นการเริ่มต้นการทำงานของเซิร์ฟเวอร์เมื่อคลิก Start แล้ว จะปรากฏข้อความในช่อง Data Packet ว่า Server Started ดังรูป



รูปที่ 4.2 โปรแกรม SockerServer เมื่อเริ่มทำงาน

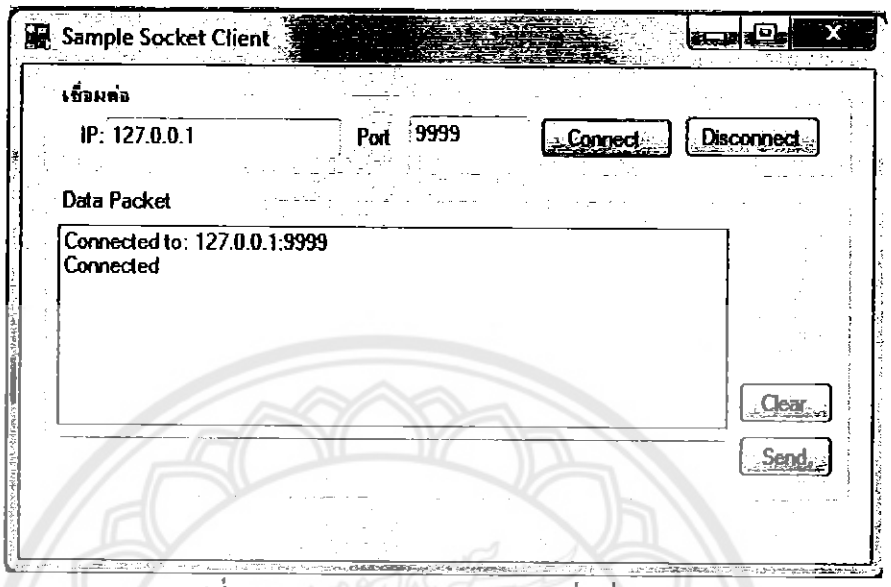
เมื่อโปรแกรม SocketServer เริ่มทำงาน ให้เปิดโปรแกรม SocketClient ขึ้นมาดังรูป



รูปที่ 4.3 โปรแกรม SockerCilent

หลังจากนั้นให้กำหนด IP Address และ Port ของเซิร์ฟเวอร์ที่เราต้องการเชื่อมต่อ ในที่นี้กำหนดให้ IP Address เป็น 127.0.0.1 และ Port เป็น 9999 เมื่อกำหนดค่าแล้ว จึงคลิก Connect

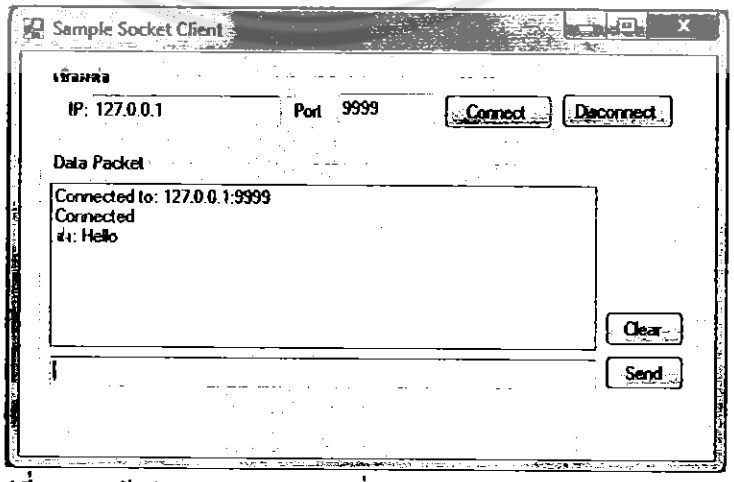
และเมื่อคลิก Connect แล้ว จะขึ้นสถานการณ์เชื่อมต่อว่าทำการเชื่อมต่อกับ IP Address ใด และ Port ใด ดังรูป



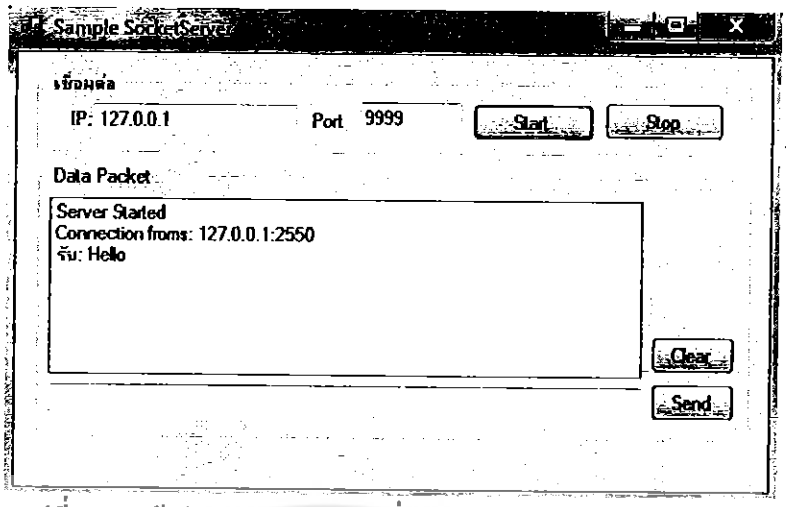
รูปที่ 4.4 โปรแกรม SocketClient เมื่อเริ่มทำงาน

4.2 การทดลองการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยที่ยังไม่มี ส่วนที่ให้ผู้ส่งตอบคำถามยืนยันตัวตน

เมื่อโปรแกรมทั้งสองได้เชื่อมต่อกันเรียบร้อยแล้ว จะทดลองการส่งโดยที่พิมพ์ข้อความในช่องพิมพ์ข้อความของโปรแกรม SocketClient ซึ่งผู้จัดทำทดลองด้วยการพิมพ์คำว่า Hello แล้วคลิกปุ่ม Send หลังจากคลิกแล้วข้อความก็จะถูกส่งไปที่เซิร์ฟเวอร์ และเซิร์ฟเวอร์จะแสดงผลว่าได้รับข้อความจากไคลเอนท์แล้ว ดังรูป



รูปที่ 4.5 หน้าต่าง SocketClient เมื่อมีการส่งข้อความไปยังเซิร์ฟเวอร์



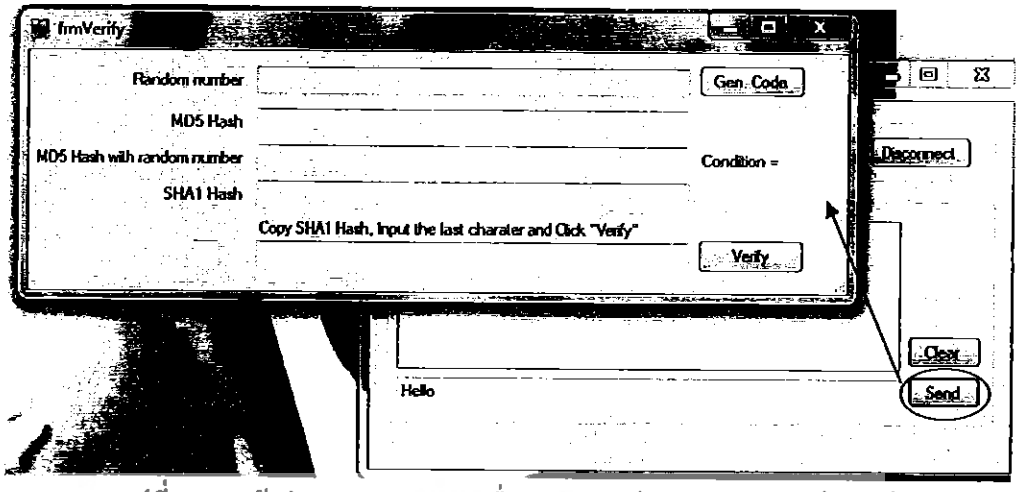
รูปที่ 4.6 หน้าต่าง SocketServer เมื่อมีการรับข้อความจากไคลเอนท์

จากการทดลองเชื่อมต่อและส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์ ผลปรากฏว่าสามารถเชื่อมต่อ และสามารถส่งข้อความได้

4.3 การทดลองการเชื่อมต่อ และการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยมีส่วนที่ให้ผู้ส่งตอบคำถามยืนยันตัวตน ในกรณีที่สามารถตอบคำถามได้ถูกต้อง

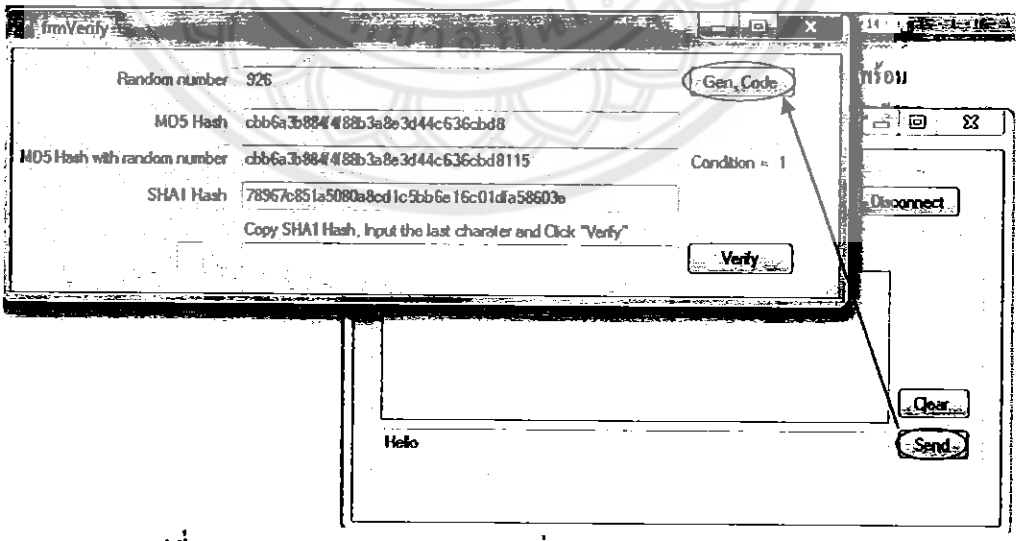
การทดลองในส่วนที่สามนี้ จะเป็นการทดลองส่งข้อความโดยที่ผู้ส่งจะต้องมีการตอบคำถามที่ระบบสร้างขึ้น ในที่นี้ระบบจะให้ข้อความโจทย์มาบางส่วน ซึ่งโจทย์นั้นก็คือคำตอบที่ 2 ที่ได้จากการสุ่มตัวเลขและทำการเข้ารหัสแบบ MD5 และมีการเติมตัวเลขที่สุ่มขึ้นมาอีก จากนั้นทำการเข้ารหัสแบบ SHA1 ทำให้ได้คำตอบที่ 2 และนำคำตอบที่ 2 นี้มาเป็นโจทย์ ซึ่งได้อธิบายวิธีการไปในหัวข้อที่ 3.3 โดยที่จะแสดงทั้งหมด 39 จาก 40 อักขระ โดยที่ 1 ตัวอักขระที่เหลือนั้น ผู้ส่งจะต้องเป็นผู้เติมลงไป โดยที่วิธีการทดลองมีดังนี้

หลังจากที่ได้เชื่อมต่อเซิร์ฟเวอร์และไคลเอนท์เรียบร้อยแล้ว จึงทำการพิมพ์ข้อความลงในช่องพิมพ์ข้อความ หลังจากนั้นคลิกที่ปุ่ม Send และเมื่อคลิกปุ่ม Send แล้ว จะปรากฏหน้าต่างเพื่อให้ผู้ส่งตอบคำถามดังรูป

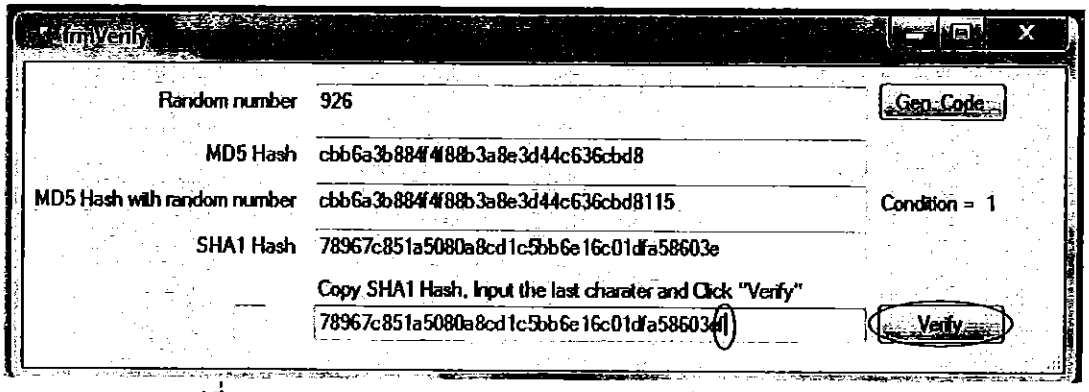


รูปที่ 4.7 หน้าต่าง SocketClient เมื่อทำการส่งข้อความไปยังเซิร์ฟเวอร์
โดยที่มีหน้าต่างให้ตอบคำถามเพื่อยืนยันตัวตน

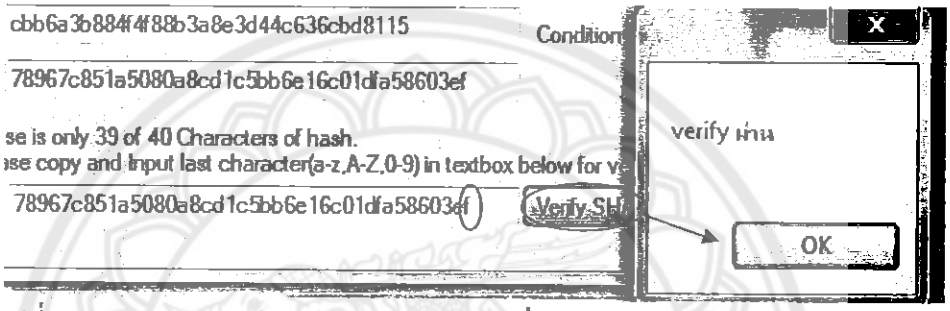
หลังจากที่หน้าต่างที่ให้ผู้ส่งตอบคำถามยืนยันตัวตนถูกเปิดขึ้นมาแล้ว ต่อไปผู้จัดทำจะทดลองตอบคำถามด้วยการคลิกที่ปุ่ม Gen.Code เพื่อสุ่มคำถามขึ้นมา โดยที่โจทย์ของผู้ส่งที่จะต้องทำก็คือ คัดลอกข้อความในช่อง SHA1 Hash และวางในช่องสีขาวด้านล่างสุดของหน้าต่างพร้อมทั้งเติมตัวอักขระ (ตัวอักษรภาษาอังกฤษหรือตัวเลข 0-9) สุดท้ายให้ถูกต้อง ซึ่งอักขระดังกล่าว ผู้ส่งนั้นจะต้องคาดเดาเอง เมื่อคัดลอกและเติมตัวอักขระสุดท้ายแล้ว คลิกที่ปุ่ม Verify เพื่อทำการยืนยันคำตอบ และถ้าคำตอบที่ใส่ลงไปนั้นถูกต้อง ก็จะมีกล่องข้อความแสดงข้อความว่า "Verify ผ่าน" และจะสามารถส่งข้อความไปยังเซิร์ฟเวอร์ได้ ดังรูปที่ 4.8, 4.9, 4.10 และ 4.11 ตามลำดับ



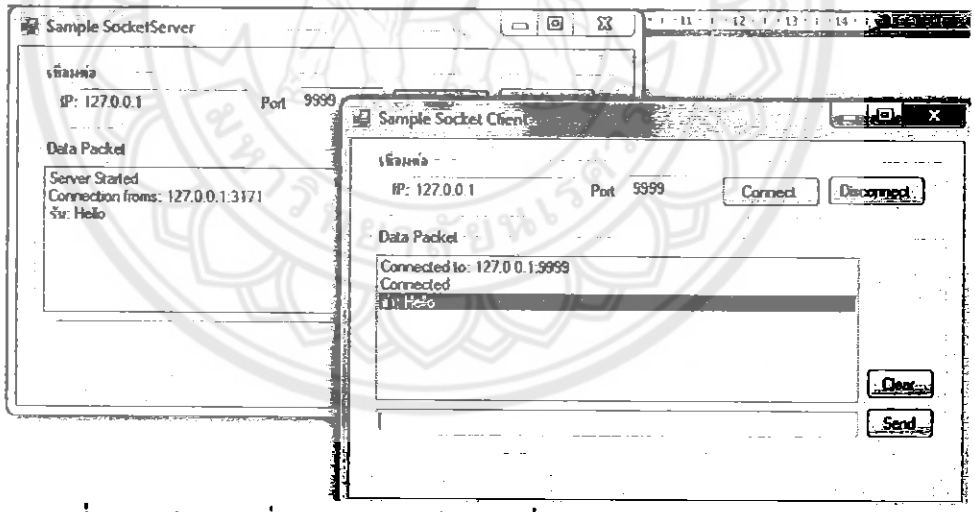
รูปที่ 4.8 แสดงการกด Gen. Code เพื่อทำการสุ่มโจทย์ยืนยันตัวตน



รูปที่ 4.9 ทำการคัดลอกและเติมตัวอักขระสุดท้ายในช่องใส่คำตอบ



รูปที่ 4.10 แสดงกล่องข้อความว่า verify ผ่าน เมื่อมีการเติมตัวอักขระสุดท้ายได้ถูกต้อง



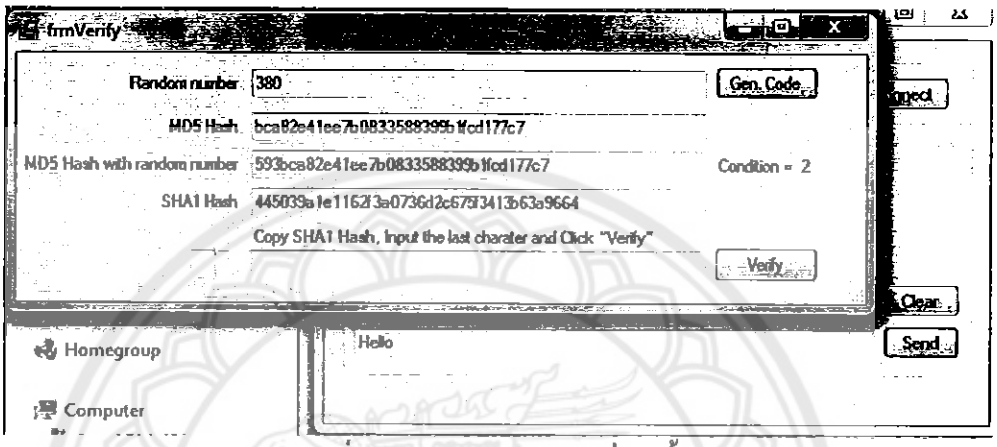
รูปที่ 4.11 ข้อความที่ถูกส่งไปยังเซิร์ฟเวอร์เมื่อผู้ส่งสามารถตอบคำถามได้ถูกต้อง

4.4 การทดลองการเชื่อมต่อและการส่งข้อความของผู้ส่งไปยังเซิร์ฟเวอร์โดยที่มีส่วนที่ให้ผู้ส่งตอบคำถามยืนยันตัวตน ในกรณีที่ไม่สามารถตอบคำถามได้ถูกต้อง

จากหัวข้อ 4.3 ถ้าผู้ส่งสามารถตอบคำถามยืนยันตัวตนได้ถูกต้อง ข้อความก็จะสามารถส่งไปยังเซิร์ฟเวอร์ได้ แต่ในส่วนนี้จะทำการทดลองว่า ถ้าไม่สามารถตอบคำถามยืนยันตัวตนที่

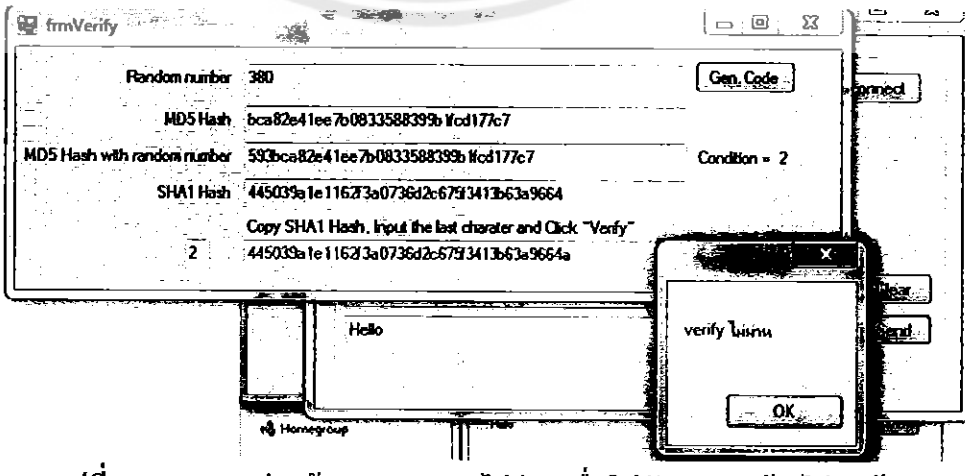
ระบบสร้างขึ้นได้ถูกต้องแล้ว ข้อความที่ผู้ส่งพิมพ์นั้น จะสามารถส่งไปยังเซิร์ฟเวอร์ได้หรือไม่ ซึ่งวิธีการทดลองมีดังนี้

หลังจากที่ทำการเชื่อมต่อเซิร์ฟเวอร์และไคลเอนท์ตามหัวข้อ 4.1 แล้ว เมื่อผู้ส่งพิมพ์ข้อความแล้ว คลิกปุ่ม Send เพื่อทำการส่งข้อความ จะปรากฏหน้าต่างที่ให้ผู้ส่งต้องสุ่มชุดอักขระขึ้นมาดังรูปที่ 4.7 หลังจากนั้น ผู้ส่งจะต้องคลิกที่ Gen. Code เพื่อทำการสุ่มชุดอักขระขึ้นมาดังรูป

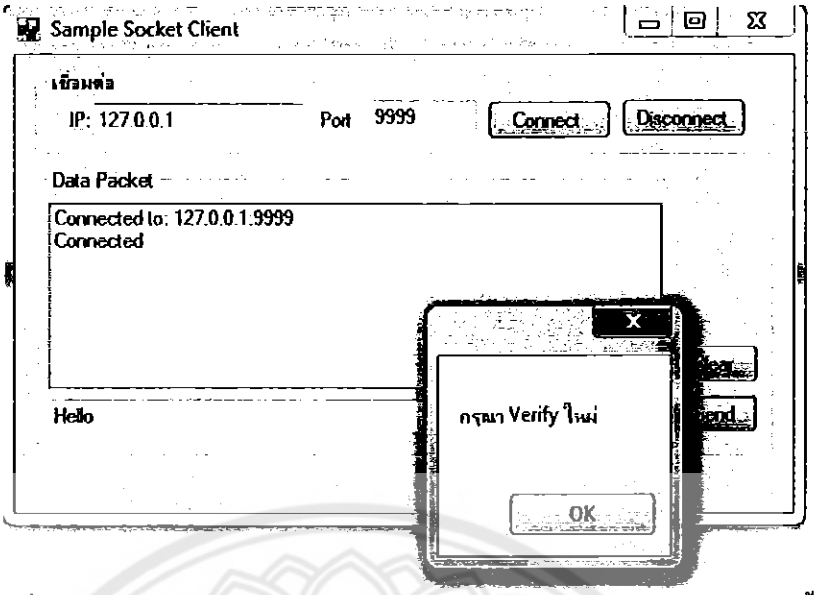


รูปที่ 4.12 แสดงชุดอักขระที่สุ่มขึ้น

เมื่อได้สุ่มชุดอักขระแล้ว ผู้ส่งก็จะทำการคัดลอกชุดอักขระจากช่อง SHA1 Hash และวางในช่องใส่ข้อความด้านล่างของหน้าต่างเดียวกัน จากนั้นผู้ส่งจะต้องเดาตัวอักขระสุดท้ายจากนั้นจึงคลิก Verify แต่ในการทดลองส่วนนี้ กำหนดให้ใส่อักขระสุดท้ายที่ไม่ถูกต้องลงไป ผลลัพธ์ที่ได้ก็คือ จะมีกล่องข้อความแสดงขึ้นว่า "Verify ไม่ผ่าน" และขึ้นกล่องข้อความต่อไปว่า "กรุณา Verify ใหม่" โดยที่โจทย์ที่ผู้ส่งจะต้องทำการตอบนั้น จะถูกสุ่มขึ้นใหม่ทุกครั้งที่มีการส่งข้อความ ดังรูปที่ 4.13 และ 4.14 ตามลำดับ



รูปที่ 4.13 แสดงกล่องข้อความ verify ไม่ผ่าน เมื่อใส่อักขระสุดท้ายไม่ถูกต้อง



รูปที่ 4.14 จากรูป 4.13 จะแสดงกล่องข้อความให้ทำการ Verify ใหม่อีกครั้ง

การผลการทดลองทั้งหมดที่ผ่านมาสามารถสรุปผลโดยรวมได้ว่าระบบสามารถทำตามวัตถุประสงค์ คือ

- สามารถทำการเชื่อมต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้
- สามารถทำการส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์โดยไม่มีเงื่อนไขใดๆ ได้
- สามารถทำการส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์โดยที่สามารถตอบคำถามยืนยันตัวตนที่ระบบสุ่มขึ้นมาได้อย่างถูกต้อง
- ไม่สามารถทำการส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์ได้เมื่อไม่สามารถตอบคำถามยืนยันตัวตนที่ระบบสุ่มขึ้นมาได้อย่างถูกต้อง

และจากวิธีการสร้างโปรแกรมในส่วนของการเข้ารหัสซึ่งได้อธิบายไว้ในบทที่ 3 ว่ามีการสุ่มตัวเลข 3 หลักถึงสองครั้งโดยในครั้งแรกสุ่มเพื่อเป็นค่าเริ่มต้นในการเข้ารหัสแบบ MD5 และครั้งที่สองสุ่มเพื่อเติมเข้าไปในคำตอบแรกที่ได้ก่อนที่จะมีการเข้ารหัสแบบ SHA1 อีกครั้งหนึ่งเพื่อให้เป็นโจทย์ ซึ่งการสุ่มสองครั้งดังกล่าวนี้ จะเป็นการป้องกันผู้ที่ทำการเดาคำตอบมาก่อนโดยวิธีการสร้างตารางความเป็นไปได้ของคำตอบ ซึ่งการสุ่มตัวเลขขึ้นมานั้นจะทำผู้ไม่หวังดีที่จะพยายามเดาคำตอบโดยการสร้างตารางความเป็นไปได้มาล่วงหน้าทำได้ยากขึ้น และเสียเวลามากขึ้น เนื่องจากการสุ่มตัวเลขทั้งสองครั้ง เป็นการสุ่มขึ้นมาของระบบ ซึ่งผู้ไม่หวังดีดังกล่าวไม่สามารถคาดเดาได้ว่า ตัวเลขที่จะสุ่มได้นั้น สามารถเป็นอะไรได้บ้าง ซึ่งทำให้ผู้ที่พยายามคาดเดาคำตอบมาก่อนด้วยการใช้วิธีดังกล่าว ต้องเสียเวลามากขึ้นเพื่อนคาดเดาคำตอบ โดยการสุ่มตัวเลขสองครั้งนั้นมีข้อแตกต่างจากการสุ่มตัวเลขเพียงครั้งเดียว 6 หลักคือ แม้การตัวเลขที่สุ่มขึ้นมาจะมี 6 หลักเหมือนกัน และความเป็นไปได้ของการคาดเดาจะมี 100,000 คำตอบ แต่การที่ระบบสุ่มตัวเลข

ขึ้นมา 6 หลักในครั้งเดียว ผู้ไม่หวังดีก็จะสามารถคาดเดาคำตอบที่จะเกิดขึ้นได้ด้วยการสร้างตารางความเป็นไปได้ในครั้งเดียว แต่ถ้าเราทำการแบ่งการสุ่มเป็นสองครั้ง และใช้วิธีการเข้ารหัสถึง 2 แบบ ทำให้ผู้ไม่หวังดีนั้นต้องทำการคาดเดาคำตอบโดยวิธีการสร้างตารางความเป็นไปได้ 2 ครั้ง ซึ่งจะทำให้ผู้บุกรุกนั้นต้องเสียเวลาในการคาดเดาคำตอบมากขึ้นด้วย แม้การตัวเลขที่สุ่มขึ้นมาจะมี 6 หลักเหมือนกัน แต่อย่างไรก็ตาม ระบบดังกล่าวยังต้องมีการพัฒนาและปรับปรุงต่อไป ซึ่งผลการทดลองที่สรุปได้ ปัญหาและอุปสรรคที่พบ แนวทางแก้ไข ข้อเสนอแนะ และแนวทางในการพัฒนาต่อไปนั้น จะสรุปไว้ในบทถัดไป

บทที่ 5
ผลสรุปและข้อเสนอแนะ

ในบทนี้เป็นการสรุปผลการดำเนินโครงงาน ปัญหาและอุปสรรคพร้อมแนวทางแก้ไข รวมไปถึงข้อเสนอแนะ และแนวทางในการพัฒนาต่อยอดโครงงานนี้ต่อไป

5.1 สรุปผลการดำเนินโครงงาน

โครงงานการสร้างระบบป้องกันสแปมนี้ จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อสร้างระบบที่สามารถป้องกันการส่งสแปมของผู้ไม่หวังดีหรือผู้บุกรุกอื่นๆ ในลักษณะ Cost Base Spam Control ซึ่งผู้ส่งจะต้องทำการบางอย่างเพื่อเป็นค่าใช้จ่ายในการส่งข้อความไปยังเซิร์ฟเวอร์ซึ่งในที่นี้จะให้ผู้ส่งนั้นตอบโจทย์เพื่อเป็นการยืนยันตัวตนดังที่ได้กล่าวไว้ในบทที่ 3 ซึ่งได้ทำการทดลองระบบจำลองการรับส่งข้อความระหว่างไคลเอนท์และเซิร์ฟเวอร์เพื่อทดสอบเงื่อนไขที่ผู้จัดทำได้พัฒนาขึ้น

สำหรับการออกแบบระบบนั้นแบ่ง 4 ส่วน ส่วนแรกคือส่วนของการเชื่อมต่อเซิร์ฟเวอร์ เซิร์ฟเวอร์และ ไคลเอนท์ซึ่งจะทำการเชื่อมต่อการก่อนที่จะมีการส่งข้อความ ส่วนที่สองคือ ส่วนที่ทำการส่งข้อความที่ผู้ส่งพิมพ์ขึ้นโดยส่งจากไคลเอนท์ไปยังเซิร์ฟเวอร์ ส่วนที่สามคือ ส่วนที่เป็นเงื่อนไขในการส่ง ซึ่งได้ทำเป็นหน้าต่างเพื่อให้ผู้ส่งได้สุ่มเลือกโจทย์ โดยที่ระบบจะเป็นผู้สุ่มโจทย์นั้นขึ้นมา และให้ผู้ส่งทำตอบโจทย์นั้นให้ถูกต้อง จึงจะมีการส่งข้อความจากไคลเอนท์ไปยังเซิร์ฟเวอร์ แต่ถ้าหากไม่สามารถตอบคำถามได้ถูกต้อง ข้อความนั้นก็จะไม่ถูกส่งและจะต้องทำการตอบคำถามใหม่อีกครั้งโดยที่โจทย์จะถูกสุ่มใหม่ทุกครั้งที่มีการส่งข้อความ และส่วนที่สี่คือ การออกแบบส่วนติดต่อกับผู้ใช้ ซึ่งผู้จัดทำได้พัฒนาโปรแกรมโดยใช้ภาษา C# โดยให้โปรแกรมสามารถทำการเชื่อมต่อ รับส่งข้อความระหว่างเซิร์ฟเวอร์และไคลเอนท์ และสามารถใช้คำสั่งการเข้ารหัสเพื่อใช้สร้างคำถามยืนยันตัวตนได้

จากการทดลองในบทที่ 4 ในละส่วนนั้น จะเห็นว่าสามารถทำให้ผู้ส่งนั้นต้องเสียเวลาในการส่งข้อความมากขึ้นในแต่ละครั้ง เพราะต้องมีการตอบคำถามที่ระบบสุ่มขึ้นมาซึ่งจะมีผลในเรื่องของลดจำนวนการส่งสแปม เนื่องจากการส่งสแปมผ่านผู้บริการฟรีอีเมล์มีความสะดวก และสามารถส่งได้หลายๆผู้รับในเวลาเดียวกัน แต่ถ้าต้องมีการทำการตอบคำถามที่ระบบสุ่มขึ้นมาในทุกๆครั้งที่มีการส่งข้อความ จะทำให้ผู้ส่งจำเป็นต้องตอบคำถามและเสียเวลามากขึ้น และไม่สามารถส่งได้ถ้าหากว่าไม่สามารถตอบคำถามได้ถูกต้อง และวิธีการตั้งคำถามของระบบนั้น จะมาจากการสุ่มตัวเลข 3 หลักโดยระบบ และทำการเข้ารหัสถึงสองครั้งแบบ MD5 และ SHA1 ตามลำดับ และนอกจากนี้ยังเพิ่มตัวเลขที่สุ่มอีก 3 หลักและใส่เข้าไปในคำตอบแรกก่อนที่จะทำการ

เข้ารหัสแบบ SHA1 อีกด้วย ซึ่งจะช่วยป้องกันผู้ไม่หวังดีที่พยายามจะตอบคำถามด้วยการคำนวณและสร้างตารางความเป็นไปได้ของคำตอบมาก่อน ทำให้การคาดเดาคำตอบด้วยวิธีการดังกล่าวทำได้ยากขึ้น กล่าวคือ เสียเวลามากขึ้นนั่นเอง เนื่องจากไม่สามารถคาดเดาได้ว่าโจทย์จะเป็นแบบใด ดังที่แสดงในตาราง 5.1.1

ตาราง 5.1 เปรียบเทียบข้อดีของสร้างโจทย์ด้วยการเข้ารหัสแบบครั้งเดียวและแบบสองครั้งพร้อมเพิ่มตัวเลขสุ่ม

การสร้างโจทย์แบบเข้ารหัสครั้งเดียว	การสร้างโจทย์แบบเข้ารหัสสองครั้งพร้อมเพิ่มตัวเลขสุ่ม
1. ผู้บุกรุกสามารถสร้างตารางความเป็นไปได้ของคำตอบมาล่วงหน้าได้โดยง่าย เพราะสร้างเพียงตารางเดียวก็สามารถคำนวณคำตอบได้ 2. ผู้บุกรุกให้เวลาที่ไม่มากในการคาดเดาคำตอบโดยวิธีการสร้างตารางความเป็นไปได้ของคำตอบ	1. ผู้บุกรุกต้องสร้างตารางความเป็นไปได้ถึงสองตารางเนื่องจากใช้วิธีการเข้ารหัสสองแบบ ทำให้การคำนวณตารางความเป็นไปได้เพื่อหาคำตอบล่วงหน้าเป็นไปได้ยากขึ้น 2. การเพิ่มเลขสุ่มทำให้การคาดเดาคำตอบมาก่อนโดยการสร้างตารางความเป็นไปได้ของคำตอบทำได้ยากมากขึ้น 3. ผู้บุกรุกต้องเสียเวลามากขึ้นจากการสร้างตารางความเป็นไปได้ของคำตอบขนาดใหญ่ 2 ตาราง และจากการสร้างตารางเพื่อคาดเดาเลขสุ่ม

แต่อย่างไรก็ตาม เนื่องจากคอมพิวเตอร์ในปัจจุบันมีความเร็วมาก ทำให้สามารถคำนวณตารางความเป็นไปได้ด้วยความรวดเร็วมากขึ้นการที่ใช้ตัวเลขสุ่มเพียง 3 หลัก 2 ครั้ง อาจจะยังไม่สามารถทำให้ผู้ไม่หวังดีนั้นเสียเวลาเพิ่มขึ้นมากนัก ซึ่งจะต้องมีการพัฒนาต่อไป

การทำโครงงานนี้ ถือว่าประสบความสำเร็จมากในระดับหนึ่ง เนื่องจากสามารถบรรลุวัตถุประสงค์ได้ตามที่คาดหวังไว้ ได้รับความรู้ใหม่ แม้จะมีอุปสรรคหลายอย่างระหว่างการทำโครงงานนี้ แต่โครงงานนี้ต้องมีการพัฒนาต่อ ซึ่งจะต้องมีการปรับปรุงในส่วนของสร้างโจทย์ให้ผู้ส่งตอบคำถามเพื่อให้มีความปลอดภัยและสามารถป้องกันได้อย่างมีประสิทธิภาพมากขึ้น

5.2 ปัญหา อุปสรรค และแนวทางแก้ไข ข้อเสนอแนะ

5.2.1 ปัญหาด้านซอฟต์แวร์

ตารางที่ 5.2 ปัญหา อุปสรรค และแนวทางแก้ไข ข้อเสนอแนะ ด้านซอฟต์แวร์

ปัญหา อุปสรรค	แนวทางแก้ไข ข้อเสนอแนะ
1. โครงงานที่ทำ ให้ภาษา C# ในการพัฒนาระบบ ซึ่งเป็นภาษาที่ผู้จัดทำไม่เคยใช้มาก่อน ทำให้เสียเวลาพอสมควรในการเรียนรู้การใช้งาน	1. ศึกษาและเรียนรู้การใช้งานภาษา C# จากหนังสือคู่มือการเขียนโปรแกรมและเว็บไซต์ที่แนะนำการใช้โปรแกรมให้มีความเข้าใจ และหมั่นฝึกฝนการเขียนโปรแกรมด้วยภาษา C# บ่อยๆ เพื่อเป็นการเพิ่มทักษะและความคล่องตัวในการใช้งาน
2. ไม่มีความรู้เรื่องการเขียน Socket Programming มาก่อน ทำให้ต้องเสียเวลาในการหาข้อมูลเพื่อประกอบการทำโครงงาน	2. ศึกษาค้นคว้าเกี่ยวกับการเขียน Socket Programming ในส่วนของการเขียนโปรแกรมติดต่อระหว่างเซิร์ฟเวอร์และไคลเอนท์จากหนังสือ C# Network Programming โดยหมั่นฝึกเขียนโปรแกรมให้มีความชำนาญ
3. ไม่มีความชำนาญในการเขียนโปรแกรมในลักษณะที่ต้องใช้ฟังก์ชันเข้ารหัสข้อความมาก่อน ทำให้ต้องใช้เวลาในการศึกษาการเขียนโปรแกรมลักษณะดังกล่าวพอสมควร	3. ฝึกฝนและศึกษาการเขียนโปรแกรมที่ต้องใช้ฟังก์ชันเข้ารหัสอย่างสม่ำเสมอจากจากเว็บไซต์แนะการเขียนโปรแกรมการเข้ารหัสและถอดรหัส เช่น http://www.c-sharpcorner.com เพื่อเป็นการฝึกทักษะและเพิ่มความชำนาญในการเขียนโปรแกรมลักษณะดังกล่าว

5.2.2 ปัญหาด้านอื่นๆ

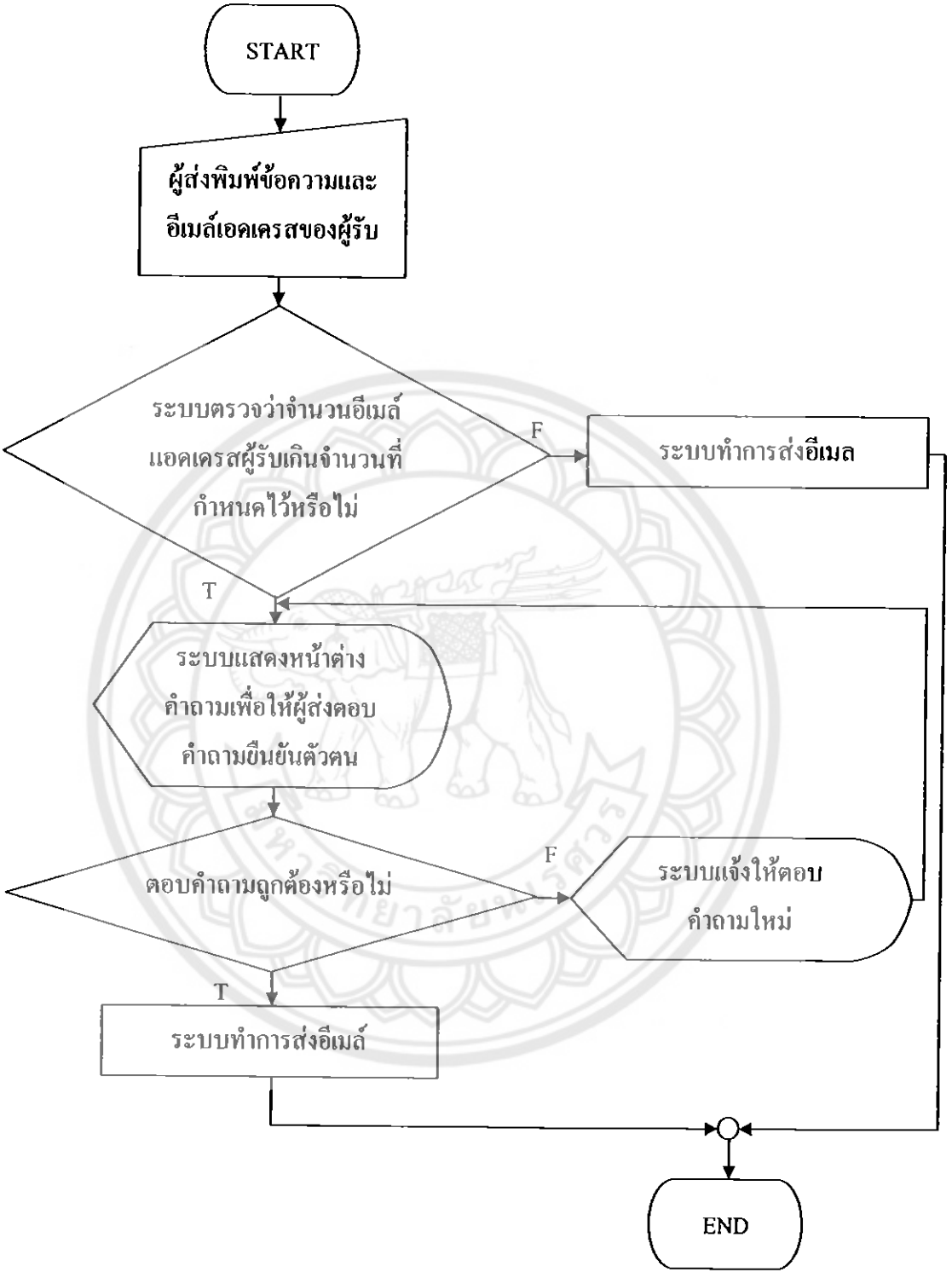
ตารางที่ 5.3 ปัญหา อุปสรรค และแนวทางแก้ไข ข้อเสนอแนะ ด้านซอฟต์แวร์

ปัญหา อุปสรรค	แนวทางแก้ไข ข้อเสนอแนะ
1. ข้อมูลที่เป็นภาษาไทยนั้นหายาก ทำให้ต้องหาข้อมูลที่เป็นภาษาอังกฤษ ซึ่งให้เวลานานในการทำความเข้าใจ เนื่องจากไม่ชำนาญการใช้ภาษาอังกฤษ	1. เพิ่มทักษะการอ่านและการเขียนภาษาอังกฤษโดยการฝึกอ่านและแปลบทความภาษาอังกฤษที่เกี่ยวข้องกับการเข้ารหัสและถอดรหัสจากเว็บไซต์ภาษาอังกฤษ เช่น http://en.wikipedia.org เพื่อเพิ่มความชำนาญในการใช้ภาษา

5.3 แนวทางในการนำไปประยุกต์ใช้

จากการทดสอบข้างต้น เป็นการทดสอบในระบบที่ได้จำลองขึ้น ซึ่งในกรณีที่จะนำระบบดังกว่าไปใช้งานจริงนั้น จะต้องมีการติดตั้งบนเว็บเมล์ โดยอาจทำในลักษณะของหน้าเว็บและมีการกำหนดว่าถ้าส่งอีเมล์เกินจำนวนฉบับที่ได้กำหนดไว้ ก็ให้มีหน้าเว็บใหม่ที่จะใช้ทำการตอบคำถามยืนยันตัวตน ทำงานคล้ายคลึงกับ CAPTCHA คือมีการใส่คำตอบเพื่อยืนยันตัวตน หากตอบคำถามไม่ผ่านก็จะไม่สามารถส่งข้อความอีเมล์ได้ดังแผนผังตัวอย่างการนำไปใช้ในดังรูป 5.1

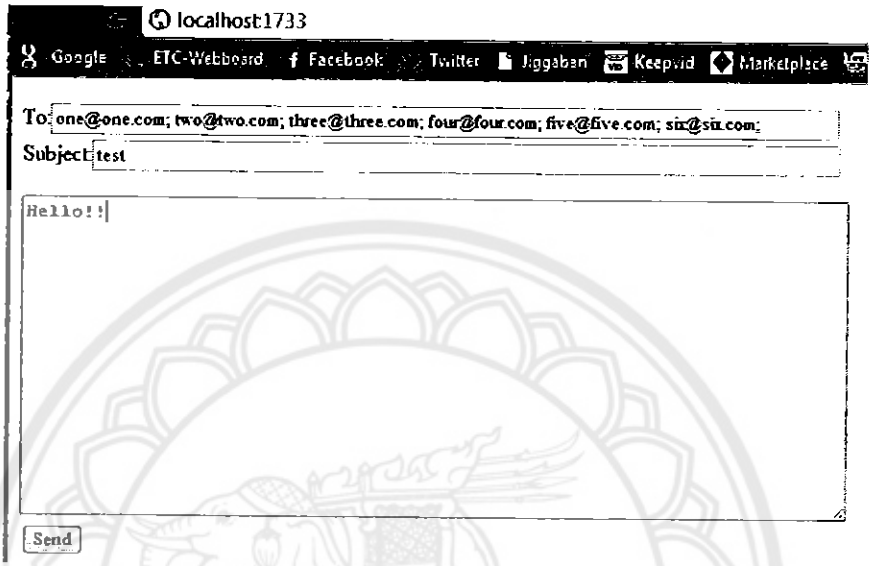
5.3.1 แนวทางการประยุกต์ใช้โปรแกรมกับระบบจริง



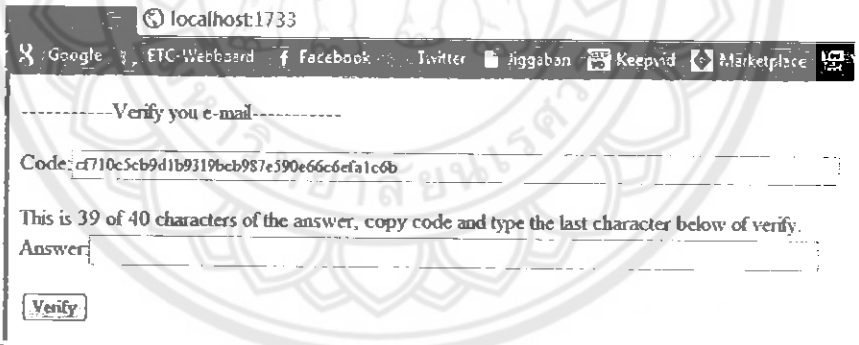
รูปที่ 5.1 แผนผังแสดงแนวทางตัวอย่างการนำระบบไปประยุกต์ใช้บนเว็บเมล์

5.1.2 ตัวอย่างการประยุกต์ใช้ระบบป้องกันสแปม

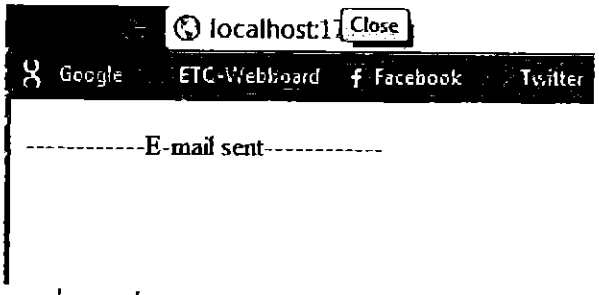
ในส่วนนี้เป็นการแสดงตัวอย่างการนำระบบป้องกันสแปมมาใช้กับเว็บเมล ซึ่งในตัวอย่างนั้นมีการกำหนดค่าไว้ว่า ถ้ามีการส่งอีเมล์ไปยังผู้รับมากกว่า 5 อีเมล์ ผู้ส่งจะต้องมีการตอบคำถามยืนยันตัวตน อีเมล์จึงจะสามารถส่งได้ดังรูป 5.2, 5.3 และ 5.4 ตามลำดับ



รูปที่ 5.2 ตัวอย่างการส่งอีเมล์

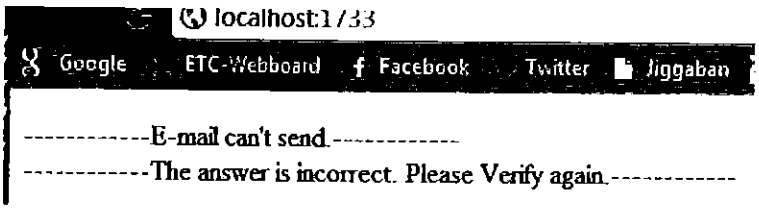


รูปที่ 5.3 หลังจากระบบตรวจสอบว่าอีเมล์แอดเดรสปลายทางเกินจำนวนที่กำหนดไว้ จะต้องมีการตอบคำถามยืนยันตัวตน



รูปที่ 5.4 เมื่อผู้ส่งตอบคำถามยืนยันตัวตนได้ถูกต้อง

แต่ถ้าผู้ส่งไม่สามารถตอบคำถามยืนยันตัวตนได้ถูกต้อง ข้อความจะไม่สามารถส่งได้ และจะต้องทำการตอบคำถามยืนยันตัวตนอีกครั้งดังรูป 5.5



รูปที่ 5.5 เมื่อผู้ส่งตอบคำถามยืนยันตัวตนไม่ถูกต้อง

5.4 แนวทางในการพัฒนาต่อไป

5.4.1 อาจจะใช้วิธีการเข้ารหัสที่มีความซับซ้อนมากขึ้นในการเข้ารหัสเพื่อให้การหาคำตอบนั้นยากมากขึ้น

5.4.2 อาจจะเปลี่ยนตัวเลขที่สุ่มขึ้นมาจาก 3 หลักเป็นตัวเลขที่มีจำนวนหลักมากขึ้น หรือเพิ่มวิธีการเข้ารหัสจาก 2 วิธีให้มากขึ้น และอาจจะเพิ่มจำนวนครั้งการสุ่มตัวเลขให้มากครั้งขึ้นเพื่อให้เสียเวลาในการคำนวณหาคำตอบมากขึ้น ในกรณีที่ผู้ไม่หวังดีทำการคาดเดาคำตอบด้วยวิธีการสร้างตารางความเป็นไปได้

5.4.3 อาจมีการกำหนดระดับความยากของการตอบคำถามไว้หลายระดับ เช่น ถ้ามีการส่งข้อความไปส่งปลายทางเป็นจำนวนน้อย ก็ให้เติมเพียงตัวอักขระสุดท้าย แต่ถ้ามีการส่งข้อความไปยังปลายทางมากขึ้น อาจจะเพิ่มตัวอักขระที่ผู้ส่งจะต้องทำการตอบคำถาม เป็นการเพิ่มค่าใช้จ่ายในการส่งมากขึ้นสำหรับผู้ที่ต้องการส่งข้อความครั้งละมากๆ ซึ่งในที่นี้ค่าใช้จ่ายที่ได้กล่าวถึง คือ การเสียเวลาในการตอบคำถาม

5.4.4 เนื่องจากในโครงงานนี้ การทดลองโปรแกรมนั้น กระทำในระบบที่จำลองขึ้นอาจจะทำให้ไม่ทราบประสิทธิภาพที่แท้จริง ดังนั้น ควรมีการทดลองในระบบอีเมล์จริงเพื่อให้การทดลองมีความแม่นยำและสามารถพัฒนาให้มีประสิทธิภาพมากขึ้น

เอกสารอ้างอิง

[1] ไม่ระบุผู้แต่ง. (4 เมษายน 2554). **วิทยาการอำพรางข้อมูล.** สืบค้นเมื่อ 3 กรกฎาคม 2554, จาก http://th.wikipedia.org/wiki/วิทยาการอำพรางข้อมูล

[2] ไม่ระบุผู้แต่ง. (6 พฤษภาคม 2554). **อีเมล.** สืบค้นเมื่อ 3 กรกฎาคม 2554, จาก http://th.wikipedia.org/wiki/อีเมล

[3] ไม่ระบุผู้แต่ง. (4 เมษายน 2554). **สแปม.** สืบค้นเมื่อ 7 กรกฎาคม 2554, จาก http://th.wikipedia.org/wiki/สแปม

[4] ไม่ระบุผู้แต่ง. (4 พฤษภาคม 2554). **E-mail filtering.** สืบค้นเมื่อ 12 กรกฎาคม 2554, จาก http://en.wikipedia.org/wiki/E-mail_filtering

[5] ไม่ระบุผู้แต่ง. (17 เมษายน 2554). **Anti-spam techniques.** สืบค้นเมื่อ 15 กรกฎาคม 2554, จาก http://en.wikipedia.org/wiki/Anti-spam_techniques

[6] ไม่ระบุผู้แต่ง. (25 ธันวาคม 2553). **Proof-of-work systems.** สืบค้นเมื่อ 20 กรกฎาคม 2554, จาก http://en.wikipedia.org/wiki/Cost-based_anti-spam_systems#Proof-of-work_systems

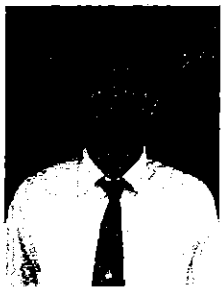
[7] ไม่ระบุผู้แต่ง. (17 เมษายน 2554). **MD5.** สืบค้นเมื่อ 21 กรกฎาคม 2554, จาก http://th.wikipedia.org/wiki/MD5

[8] ไม่ระบุผู้แต่ง. (17 เมษายน 2554). **SHA1.** สืบค้นเมื่อ 21 กรกฎาคม 2554, จาก http://en.wikipedia.org/wiki/SHA1

[9] สัจจะ จรัสรุ่งรวีวร. **เริ่มต้น Visual C# 2008 ฉบับสมบูรณ์.** นนทบุรี : ไอดีซีฯ. 2552

[10] Richard Blum. **C# Network Programming.** United State of America : SYBEX. 2003

ประวัติผู้ดำเนินโครงงาน



ชื่อ นายสุทธิพงศ์ คงชุม

ภูมิลำเนา 1/1 ถนนตั้งใจพัฒนา ตำบลในเมือง อำเภอเมือง จังหวัดเพชรบูรณ์

ประวัติการศึกษา

– จบระดับมัธยมศึกษาจากโรงเรียนเพชรพิทยาคม จังหวัดเพชรบูรณ์

– ปัจจุบันกำลังศึกษาระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

Email: 40-degree@live.com