บทที่ 3

การออกแบบและพัฒนาโปรแกรม

การออกแบบและพัฒนาโปรแกรมที่เพิ่มประสิทธิภาพในการสร้างช่องทางการสื่อสารของข้อมูลที่ปลอดภัยที่ชั้นโปรแกรมประยุกต์ มีข้อดีในแง่ของการใช้งานของผู้ใช้ทั่วๆไป เพราะการใช้งานจะเป็นเหมือนเช่นโปรแกรมเดิม โดยไม่มีขั้นตอนใดๆ เพิ่มขึ้น สิ่งที่ต้องคำนึงถึงในการออกแบบ ก็คือเพื่อให้การทำงานของโปรแกรมที่พัฒนาขึ้นมาใหม่นี้ยังสามารถทำงานร่วมกันกับโปรแกรมที่มีอยู่เดิมได้ในลักษณะเดิมได้ โดยไม่มีข้อจำกัดว่าต้องเป็นการทำงานระหว่างโปรแกรมใหม่ด้วยกันเท่านั้น การพัฒนาเพื่อสร้างช่องทางการสื่อสารที่ปลอดภัยบนโปรแกรมเทลเนต เป็นอีกแนวทางหนึ่งที่เป็นไปได้ และนำมาใช้ประโยชน์ได้เป็นอย่างดี

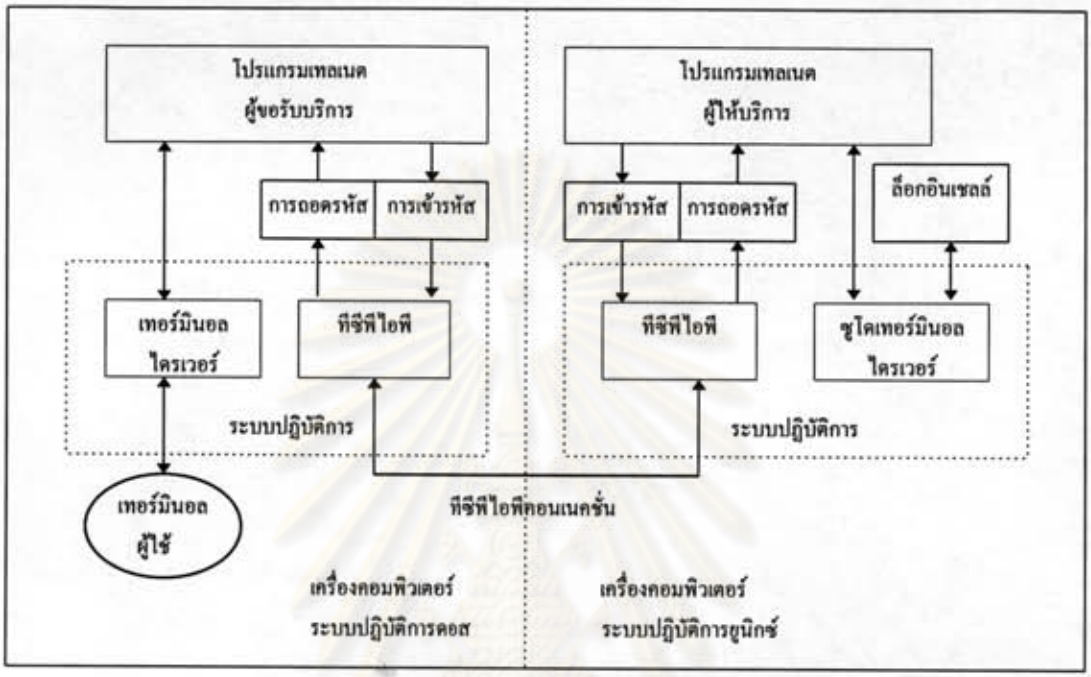
การออกแบบระบบงาน

การพัฒนาระบบความมั่นคงของเทลเนตโดยอาศัยการเข้ารหัส สำหรับการใช้งานจากเครื่องคอมพิวเตอร์ส่วนบุคคลในการเข้าใช้งานระบบจากระยะทางไกลไปยังเครื่องคอมพิวเตอร์ระบบปฏิบัติการยูนิกซ์ แบ่งการพัฒนาออกเป็น 2 ส่วนบน 2 ระบบปฏิบัติการ คือ

1. บนระบบปฏิบัติการยูนิกซ์ เป็นการพัฒนาเพิ่มระบบการเข้ารหัสบนโปรแกรมเทลเนตผู้ให้บริการ (telnet server program) โดยใช้โปรแกรมเทลเนตผู้ให้บริการต้นฉบับที่พัฒนาโดย University of California, Berkeley

2. บนระบบปฏิบัติการดอส เป็นการพัฒนาเพิ่มระบบการเข้ารหัสบนโปรแกรมเทลเนตผู้ขอรับบริการ (telnet client program) โดยใช้โปรแกรมเทลเนตผู้ขอรับบริการต้นฉบับของเอ็นซีเอสเอเทลเนตเวอร์ชั่น 2.3.08 (NCSA Telnet 2.3.08 : National Center for Supercomputing Application Terminal Emulator)

จากรูปที่ 2.1 ที่แสดงการทำงานของโปรแกรมเทลเนต รูปที่ 3.1 แสดงลักษณะการทำงานพื้นฐานของโปรแกรมเทลเนตที่เพิ่มระบบการเข้ารหัสในชั้นของโปรแกรมประยุกต์ได้



รูปที่ 3.1 แสดงการทำงานของโปรแกรมเทลเนตที่มีระบบการเข้ารหัส

จากรูปที่ 3.1 แสดงให้เห็นว่า

1. โปรแกรมเทลเนตผู้ขอรับบริการ รับข้อมูลจากผู้ใช้ที่พิมพ์ผ่านเทอร์มินอล นำมาเข้ารหัสข้อมูลก่อนส่งผ่านโปรโตคอลทีซีพีไอพี และรับข้อมูลจากโปรโตคอลทีซีพีไอพีผ่านการถอดรหัสข้อมูลเพื่อแสดงผลออกทางเทอร์มินอลของผู้ใช้ต่อไป

2. โปรแกรมเทลเนตผู้ให้บริการ รับข้อมูลจากโปรโตคอลทีซีพีไอพี นำข้อมูลไปถอดรหัสได้ข้อมูลที่ถูกต้องเพื่อนำไปประมวลผลต่อไป และเมื่อได้ผลลัพธ์นำมาเข้ารหัสข้อมูลก่อนส่งผ่านโปรโตคอลทีซีพีไอพีไปยังอีกฝ่ายหนึ่ง

การออกแบบระบบการเข้ารหัส ใช้รูปแบบการเข้ารหัสแบบบล็อคไซเฟอร์ โดยใช้อัลกอริทึ่มของ IDEA ในการเข้ารหัสและถอดรหัสข้อมูล

ดังที่ได้กล่าวแล้วว่าความปลอดภัยในการเข้ารหัสข้อมูลอยู่ที่ความปลอดภัยของเซสชั่นคีย์ การแลกเปลี่ยนเซสชั่นคีย์ระหว่างสองฝ่ายใช้วิธีการเข้ารหัสแบบคีย์สาธารณะในการเข้ารหัสเซสชั่นคีย์ก่อนส่งผ่านไปให้อีกฝ่าย โดยใช้อัลกอริทึมของ RSA ซึ่งเป็นการเข้ารหัสแบบคีย์สาธารณะ

โปรแกรมเทลเนต (telnet program)

การติดต่อสื่อสารของโปรแกรมเทลเนตผู้ขอรับบริการ และโปรแกรมเทลเนตผู้ให้บริการ เกิดขึ้นด้วยการติดต่อผ่านโปรโตคอลทีซีพีเพียงครั้งเดียว หลังจากนั้นโปรแกรมเทลเนตเริ่มต้นการทำงานด้วยการเจรจาทางเลือกต่างๆ ระหว่างกัน เพราะการทำงานเป็นลักษณะที่สมมาตรกันทั้งสองฝ่าย ในการเข้ารหัสข้อมูลก็เช่นเดียวกัน โปรแกรมเทลเนตผู้ขอรับบริการ และโปรแกรมเทลเนตผู้ให้บริการ มีการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล เพื่อสามารถส่งผ่านข้อมูลที่มีการเข้ารหัส และนำข้อมูลมาถอดรหัสได้ถูกต้องตรงกัน

1. โครงสร้างข้อมูลการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล โครงสร้างข้อมูลของคำสั่งและทางเลือกที่ใช้ในการเจรจาทางเลือกของโปรโตคอลเทลเนตสำหรับการเข้ารหัสข้อมูลในปัจจุบันยังไม่มีเอกสารอ้างอิงในอาร์เอฟซี วิทยานิพนธ์นี้ได้กำหนดโครงสร้างของคำสั่งและทางเลือกในการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูลและรหัส (code) ดังนี้

ENCRYPTION	38
IS	0
SEND	1
REPLY	2
ACCEPT	3
REJECT	4

ชนิดของการเข้ารหัสและรหัสที่ใช้ (encryption type pair-list)

CP_PRIVACY	99

2. การเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล ลักษณะการเจรจาการเข้ารหัสข้อมูลระหว่างโปรแกรมเทลเนตผู้ขอรับบริการ และโปรแกรมเทลเนตผู้ให้บริการ เป็นไปในลักษณะที่คล้ายกันกับการเจรจาในทางเลือก authentication ซึ่งมีลักษณะของการเจรจา ดังนี้

IAC WILL ENCRYPTION

ผู้ขอรับบริการส่งคำสั่งนี้ เพื่อแสดงความจุดประสงค์ในการส่งและรับข้อมูลของการเข้ารหัส

IAC DO ENCRYPTION

ผู้ให้บริการส่งคำสั่งนี้ เพื่อแสดงความจุดประสงค์ในการส่งและรับข้อมูลของการเข้ารหัส

IAC WONT ENCRYPTION

ผู้ขอรับบริการส่งคำสั่งนี้ เพื่อปฏิเสธการส่งและรับข้อมูลของการเข้ารหัส ในส่วนของผู้ให้บริการส่งคำสั่งนี้ ถ้าได้รับคำสั่ง DO TELOPT_ENCRYPTION จากผู้ขอรับบริการ

IAC DONT ENCRYPTION

ผู้ให้บริการส่งคำสั่งนี้ เพื่อปฏิเสธการส่งและรับข้อมูลของการเข้ารหัส สำหรับผู้ขอรับบริการส่งคำสั่งนี้ ถ้าได้รับคำสั่ง DO ENCRYPTION จากผู้ขอรับบริการ

IAC SB ENCRYPTION SEND encryption-type-pair-list IAC SE

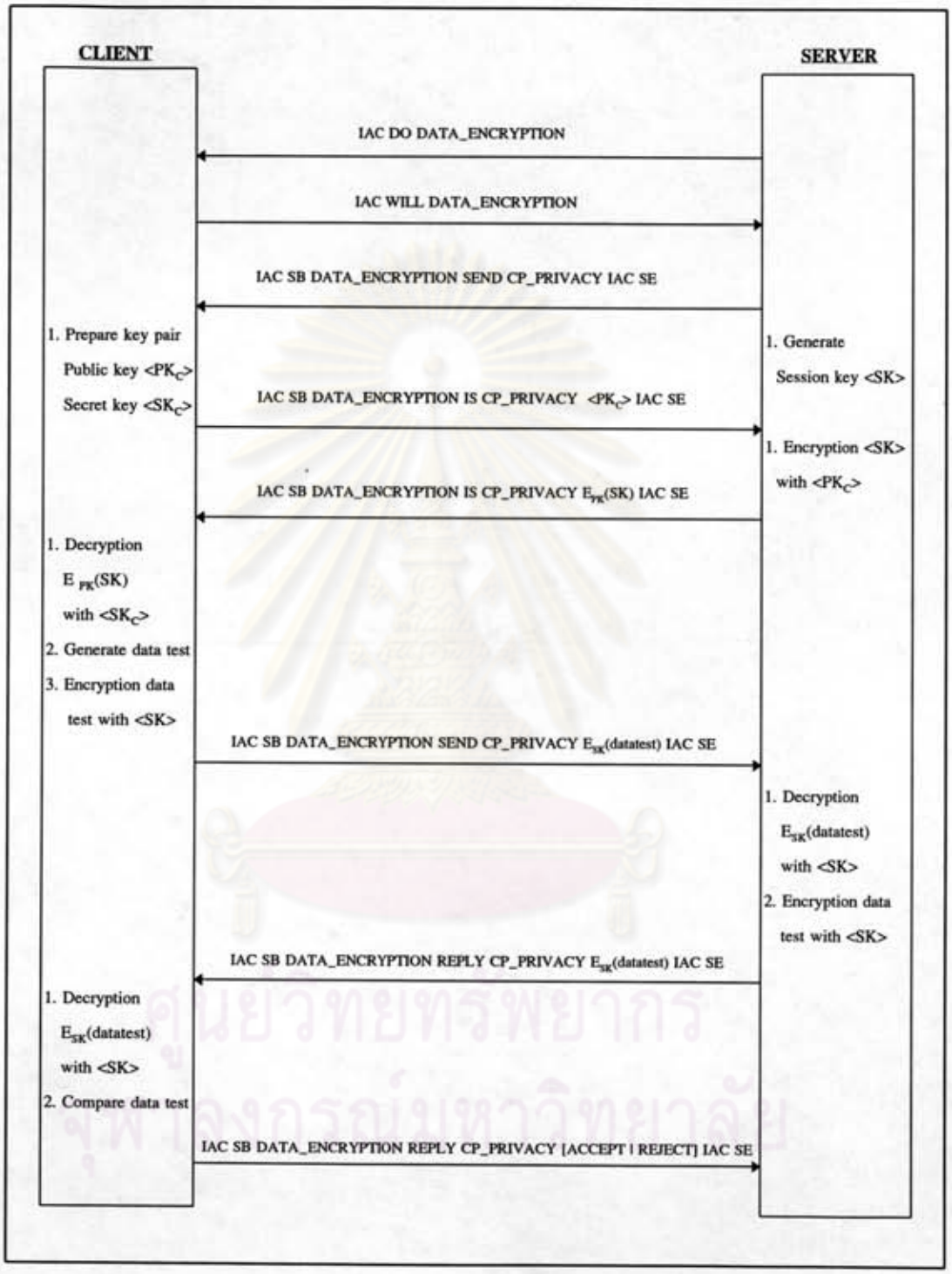
ผู้ให้บริการส่งคำสั่งนี้ เพื่อเสนอให้ผู้ขอรับบริการส่งข้อมูลที่เป็นชนิดของการเข้ารหัสแบบใดแบบหนึ่งที่มีอยู่ใน “encryption-type-pair-list” (รายชื่อชนิดของการเข้ารหัสที่ผู้ให้บริการมีไว้บริการ สำหรับในวิทยานิพนธ์นี้ กำหนดไว้คือ CP_PRIVACY)

IAC SB ENCRYPTION [SEND / IS / REPLY] encryption-type [encryption data] IAC SE

ผู้ขอรับบริการ และผู้ให้บริการ ส่งคำสั่งในกลุ่มนี้ระหว่างกัน เพื่อการส่งข้อมูลที่ใช้สำหรับการเข้ารหัสข้อมูลตามชนิดที่ตกลงร่วมกัน

3. การแลกเปลี่ยนเซสชั่นคีย์ การเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล เป็นจุดเริ่มต้นสำคัญของการเข้าสู่การทำงานในการเข้ารหัสข้อมูล เพื่อการสื่อสารข้อมูลที่ปลอดภัย ข้อมูลที่ใช้สำหรับการเข้ารหัสข้อมูลที่ถูกส่งผ่านระหว่างผู้ให้บริการ และผู้ขอรับบริการในขั้นตอนของการเจรจาในส่วนย่อยของทางเลือกสำหรับการเข้ารหัสก็คือ การแลกเปลี่ยนเซสชั่นคีย์ และข้อมูลทดสอบที่ใช้ในการตรวจสอบเซสชั่นคีย์ เพื่อให้ทั้งสองฝ่ายใช้คีย์ร่วมกันในการเข้ารหัสและถอดรหัสข้อมูลได้ถูกต้องตรงกัน

รูปที่ 3.2 แสดงการแลกเปลี่ยนเซสชั่นคีย์ และการตรวจสอบคีย์ ในขั้นตอนของการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล สำหรับชนิดของการเข้ารหัสที่ใช้ในวิทยานิพนธ์นี้



รูปที่ 3.2 แสดงการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล และการแลกเปลี่ยนเซสชั่นคีย์

อธิบายขั้นตอนการทำงานได้ดังนี้

(1) เริ่มต้นการเจรจาทางเลือกการเข้ารหัสข้อมูล โดยโปรแกรมเทลเนตผู้ให้บริการ เป็นผู้เริ่มต้นเสนอการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล คือ

IAC DO TELOPT_ENCRYPTION

(2) เมื่อโปรแกรมเทลเนตผู้ขอรับบริการ ได้รับการเสนอการเจรจาทางเลือกการเข้ารหัสข้อมูล และตอบตกลงในทางเลือกด้วยคำสั่ง คือ

IAC WILL TELOPT_ENCRYPTION

(3) เมื่อโปรแกรมเทลเนตผู้ให้บริการ ได้รับข้อมูลตอบตกลงทางเลือกจากโปรแกรมเทลเนตผู้ขอรับบริการ จะส่งคำสั่งการเจรจาในส่วนย่อยของทางเลือก เพื่อแจ้งชนิดของการเข้ารหัส คือ CP_PRIVACY ไปให้กับผู้ขอรับบริการ พร้อมทั้งสร้างเซสชั่นคีย์สุ่มขึ้นมา

IAC SB TELOPT_ENCRYPTION SEND CP_PRIVACY IAC SE

(4) ถ้าโปรแกรมเทลเนตผู้ขอรับบริการ ตกลงในการเข้ารหัสข้อมูลด้วยชนิดของการเข้ารหัส CP_PRIVACY นี้ จะส่งคีย์สาธารณะของตนไปให้กับโปรแกรมเทลเนตผู้ให้บริการ เพื่อใช้ในการเข้ารหัสเซสชั่นคีย์ โดยที่ $<PK_C>$ คือคีย์สาธารณะของผู้ขอรับบริการ

IAC SB TELOPT_ENCRYPTION IS CP_PRIVACY $<PK_C>$ IAC SE

(5) โปรแกรมเทลเนตผู้ให้บริการ ใช้คีย์สาธารณะของผู้ขอรับบริการ เข้ารหัสเซสชั่นคีย์ที่สร้างขึ้น และส่งคืนเซสชั่นคีย์ที่ถูกเข้ารหัสไปให้ผู้ขอรับบริการ โดยที่ $E_{PK}(SK)$ คือเซสชั่นคีย์ที่ถูกเข้ารหัสด้วยคีย์สาธารณะ

IAC SB TELOPT_ENCRYPTION IS CP_PRIVACY $E_{PK}(SK)$ IAC SE

(6) เมื่อโปรแกรมเทลเนตผู้ขอรับบริการถอดรหัสข้อมูลที่ได้รับ ด้วยคีย์ส่วนตัว ได้เซสชั่นคีย์ ทำการสร้างข้อมูลทดสอบและเข้ารหัสข้อมูลทดสอบนี้ด้วยเซสชั่นคีย์ ส่งข้อมูลทดสอบที่เข้ารหัสด้วยเซสชั่นคีย์นี้ไปให้โปรแกรมเทลเนตผู้ให้บริการ โดยที่ E_{SK}(data test) คือ ข้อมูลทดสอบที่ถูกเข้ารหัสด้วยเซสชั่นคีย์

IAC SB TELOPT_ENCRYPTION SEND CP_PRIVACY E_{SK}(data test) IAC SE

(7) โปรแกรมเทลเนตผู้ให้บริการถอดรหัสข้อมูลที่ได้รับ ได้ข้อมูลทดสอบจะทำการเข้ารหัสข้อมูลทดสอบด้วยเซสชั่นคีย์ของตน และส่งคืนกลับไปให้ผู้ขอรับบริการอีกครั้ง

IAC SB TELOPT_ENCRYPTION REPLY CP_PRIVACY E_{SK}(data test) IAC SE

(8) โปรแกรมเทลเนตผู้ขอรับบริการถอดรหัสข้อมูลที่ได้รับ นำมาเปรียบเทียบข้อมูลทดสอบที่ตนเก็บไว้ เพื่อตรวจสอบการเข้ารหัสและการถอดรหัสว่าทั้งสองฝ่ายสามารถทำงานได้ถูกต้องตรงกัน ถ้าการเปรียบเทียบข้อมูลทดสอบถูกต้องจะส่งคำสั่ง ACCEPT เพื่อเข้าสู่การทำงานในการเข้ารหัสข้อมูล แต่ถ้าการเปรียบเทียบข้อมูลทดสอบผิดพลาด จะส่งคำสั่ง REJECT เพื่อทำงานตามรูปแบบเดิมที่ไม่มีการเข้ารหัสข้อมูล

IAC SB TELOPT_ENCRYPTION REPLY CP_PRIVACY [ACCEPT | REJECT] IAC SE

การพัฒนาโปรแกรมเทลเนต อาศัยเครื่องมือสำคัญที่ช่วยในการพัฒนาคือไฟไนต์สเตทเมชีน โดยการกำหนดการทำงานเป็นลำดับขั้น ตามรายละเอียดของโปรโตคอล

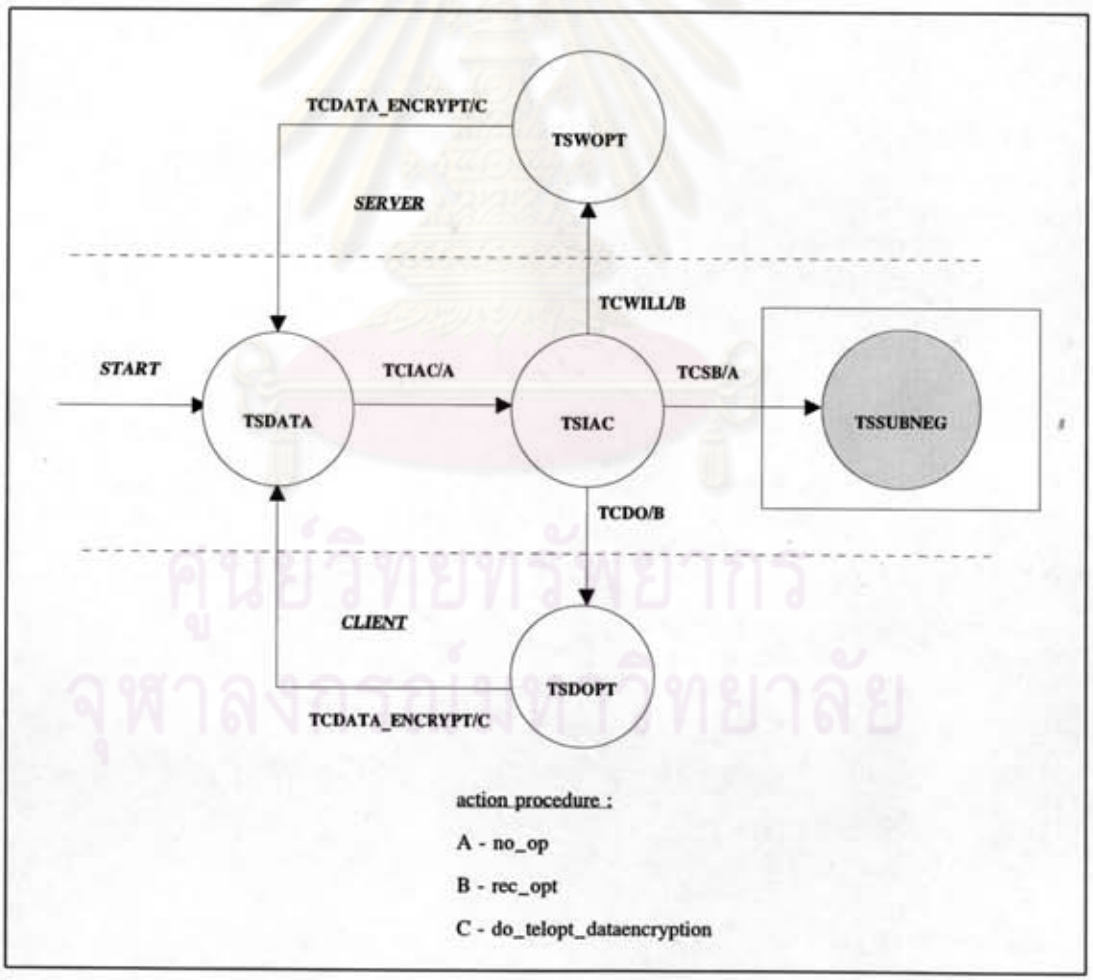
4. ไฟไนต์สเตทเมชีน ขั้นตอนของการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล ถูกกำหนดโดยการใช้ไฟไนต์สเตทเมชีน ดังที่แสดงด้วยรูปที่ 3.3

รูปที่ 3.3 แสดงขั้นตอนการทำงานของการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูลในส่วนของผู้ให้บริการและผู้ขอรับบริการ เมื่อข้อมูลถูกรับเข้ามา ไฟไนต์สเตทเมชีนทำการแปลคำสั่ง

ที่ได้ เพื่อทำงานในฟังก์ชั่นใดฟังก์ชั่นหนึ่ง พร้อมทั้งมีการเปลี่ยนสถานะจากสถานะหนึ่งไปยังอีกสถานะหนึ่ง ที่เกี่ยวข้องกับคำสั่งที่ได้รับเข้ามา

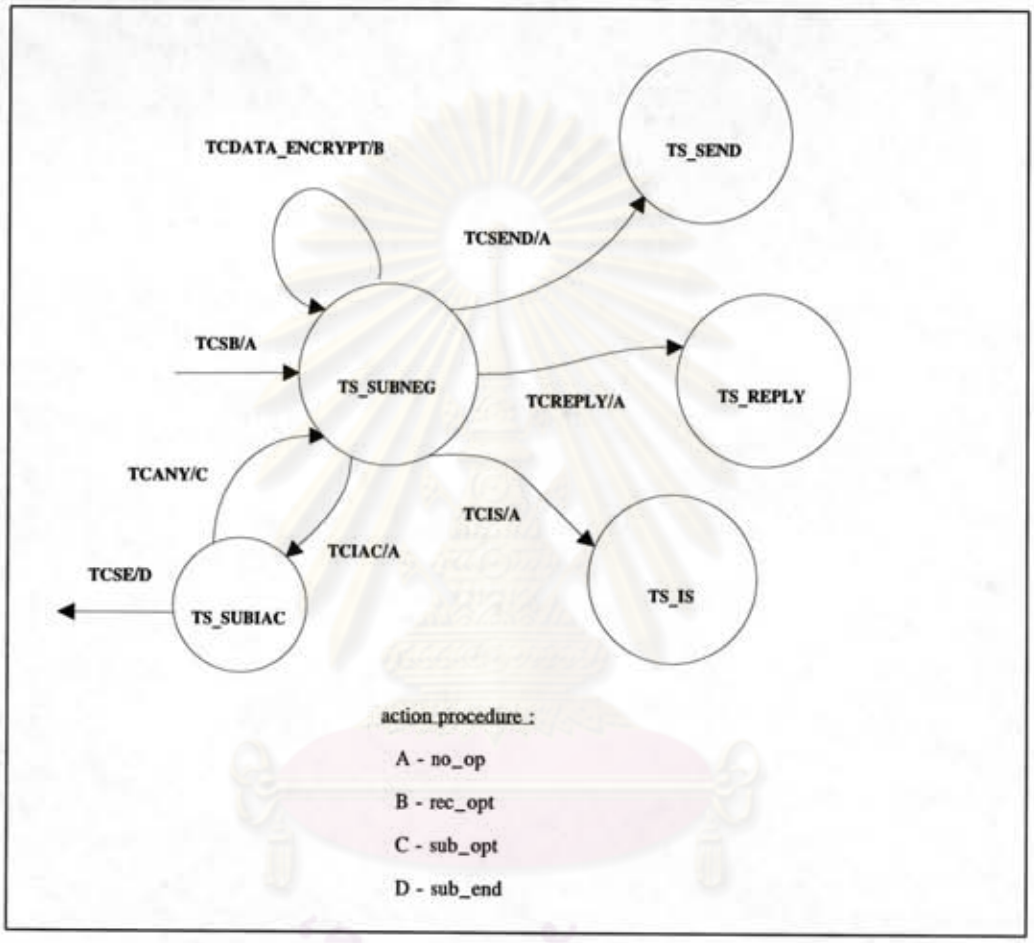
เริ่มต้นการทำงานที่สถานะ TSDATA เมื่อข้อมูลที่ได้รับเป็น TCIAC จะทำฟังก์ชั่น B คือ no operation และเปลี่ยนสถานะไปเป็นสถานะ TSIAC ในสถานะนี้ถ้าข้อมูลที่ได้รับเป็น TCSB ฟังก์ชั่นยังคงเป็น B และเปลี่ยนสถานะไปเป็นสถานะ TSSUBNEG แต่ถ้าข้อมูลที่ได้รับเป็น TCDO หรือ TCWILL จะทำฟังก์ชั่น C เป็นการบันทึกข้อมูลคำสั่ง และเปลี่ยนสถานะไปเป็นสถานะ TSDOPT และ TSWOPT ตามลำดับ

ในสถานะ TSDOPT และ TSWOPT ข้อมูลที่ได้รับเป็นทางเลือกสำหรับการเข้ารหัสข้อมูลคือ TCDATA_ENCRYPT จะทำฟังก์ชั่น do_telopt_dataencryption และเปลี่ยนสถานะไปเป็นสถานะ TSDATA เมื่อรอข้อมูลต่อไป



รูปที่ 3.3 แสดงไฟไนต์สเตทแมชีนของการเจรจาทางเลือกสำหรับการเข้ารหัสข้อมูล

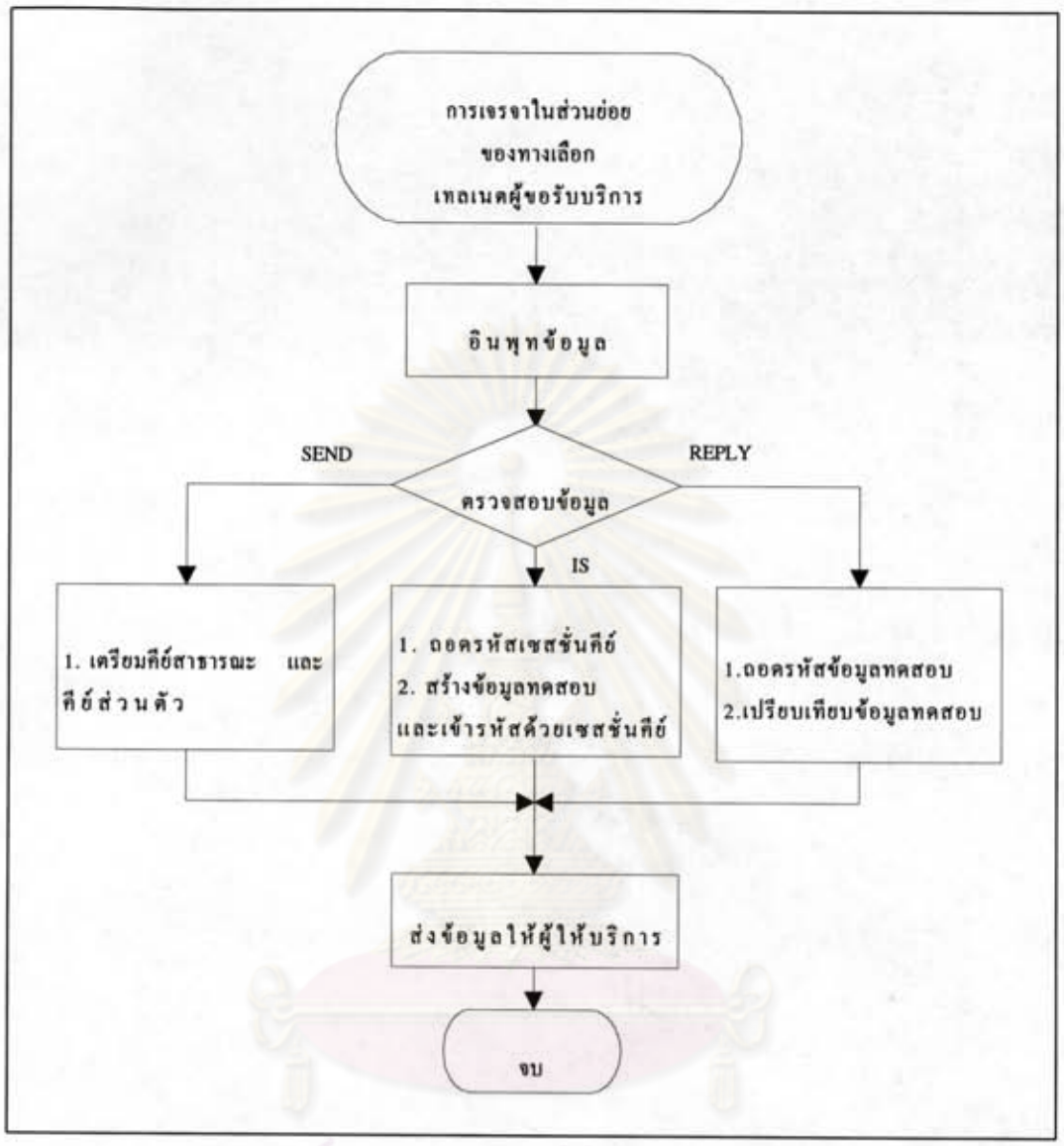
ในสถานะของ TSSUBNEG เป็นขั้นตอนสำคัญที่ใช้ในการส่งและรับข้อมูลที่ใช้ในสำหรับการเข้ารหัสข้อมูล คือเซสชั่นคีย์ รูปที่ 3.4 แสดงไฟไนต์สเตทเมชีนของการเจรจาในส่วนย่อยของทางเลือกสำหรับการเข้ารหัสข้อมูล



รูปที่ 3.4 แสดงไฟไนต์สเตทเมชีนของการเจรจาทางเลือกในส่วนย่อย

เมื่อเริ่มเข้าสู่สถานะของการเจรจาในส่วนย่อย TSSUBNEG ถ้าข้อมูลที่ได้รับเข้ามาเป็น TCDATA_ENCRYPT ทำการบันทึกทางเลือกนี้ไว้และยังคงอยู่ในสถานะเดิม เมื่อข้อมูล TCSEND เข้ามาทำการเปลี่ยนสถานะไปเป็นสถานะ TSSEND แต่ถ้าเป็นข้อมูล TCREPLY และ TCIS สถานะถูกเปลี่ยนไปเป็นสถานะ TSREPLY และ TSIS ตามลำดับ ในทั้ง 3 สถานะมีขั้นตอนการทำงานที่สำคัญในการแลกเปลี่ยนเซสชั่นคีย์

รูปที่ 3.5 แสดงลำดับขั้นตอนการทำงานของโปรแกรมเทลเนตผู้ขอรับบริการ สำหรับการทำงานในสถานะของการเจรจาทางเลือกในส่วนย่อยของสถานะทั้งสาม คือ TSSEND, TSREPLY และ TSIS ตามลำดับ



รูปที่ 3.5 แสดงลำดับขั้นตอนการทำงานของโปรแกรมเทลเนตผู้ขอรับบริการ

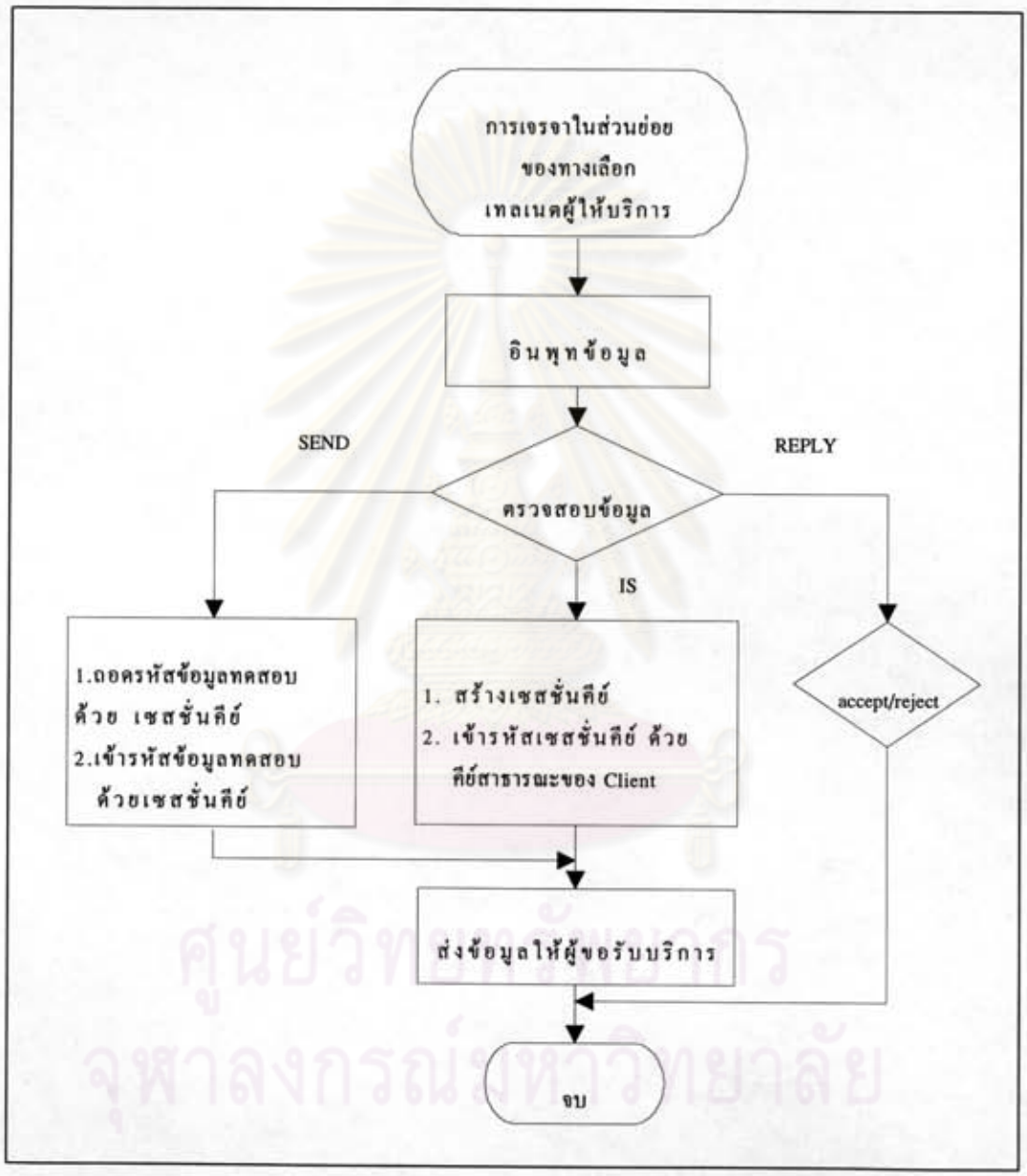
จากรูป แสดงให้เห็นขั้นตอนที่สำคัญ

(1) เริ่มต้นการแลกเปลี่ยนเซสชั่นคีย์ เมื่อโปรแกรมเทลเนตผู้ขอรับบริการ ได้รับคำสั่ง SEND จะส่งคีย์สาธารณะไปให้โปรแกรมเทลเนตผู้ให้บริการ

(2) รับข้อมูลเป็นเซสชั่นคีย์ที่ถูกเข้ารหัส ถอดรหัสเซสชั่นคีย์ นำไปเข้ารหัสข้อมูลทดสอบ เพื่อส่งไปให้โปรแกรมเทลเนตผู้ให้บริการ

(3) รับข้อมูลทดสอบที่ถูกเข้ารหัส นำมาเปรียบเทียบข้อมูลทดสอบที่เก็บไว้ เพื่อตรวจสอบการเข้ารหัสได้ถูกต้องตรงกัน

รูปที่ 3.6 แสดงลำดับขั้นตอนการทำงานของโปรแกรมเทลเนตผู้ให้บริการ สำหรับการทำงานในสถานะของการเจรจาทางเลือกในส่วนย่อยของสถานะทั้งสาม คือ TSSEND, TSREPLY และ TSIS ตามลำดับ

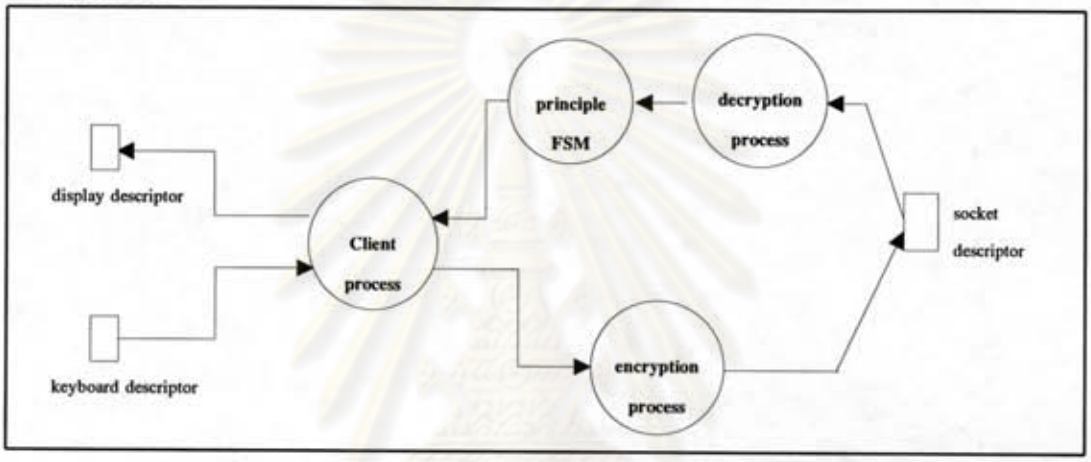


รูปที่ 3.6 แสดงลำดับขั้นตอนการทำงานของโปรแกรมเทลเนตผู้ให้บริการ

จากรูปแสดงให้เห็นขั้นตอนที่สำคัญคือ การสร้างเซสชั่นคีย์และการรอรับข้อมูล accept หรือ reject เพื่อเข้าสู่การทำงานในการเข้ารหัสข้อมูลก่อนส่งผ่านระบบสื่อสาร หรือปฏิเสธการทำงานในการเข้ารหัสข้อมูล และทำงานในรูปแบบปกติ

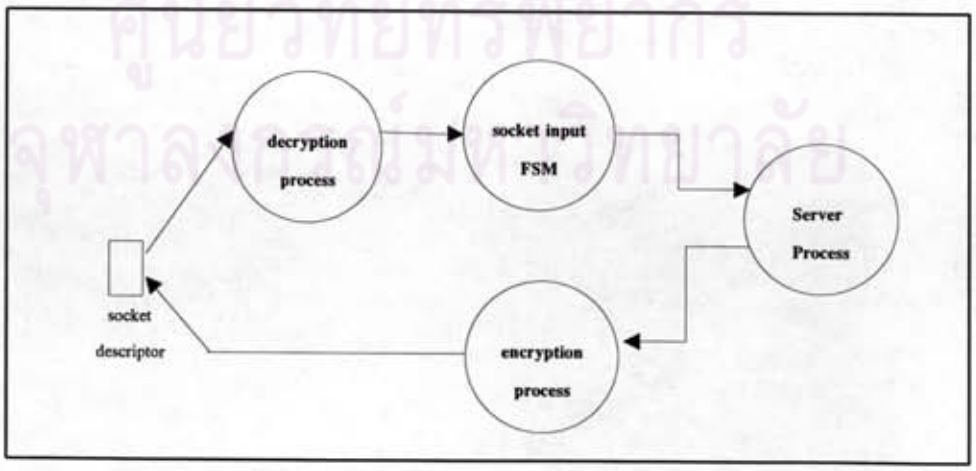
กระบวนการเข้ารหัส (encryption process)

เมื่อผ่านขั้นตอนของการแลกเปลี่ยนเซสชั่นคีย์ และตกลงร่วมกันในการเข้ารหัสข้อมูลก่อนส่งผ่านไปให้อีกฝ่ายหนึ่ง การทำงานของกระบวนการเข้ารหัสที่นำเข้ามาเป็นส่วนหนึ่งของการทำงานของโปรแกรมเทลเนต รูปที่ 3.7 แสดงการทำงานของโปรแกรมเทลเนตผู้ขอรับบริการ ในการนำระบบการเข้ารหัสข้อมูล มาใช้ในการสร้างความปลอดภัยให้กับข้อมูลก่อนส่งผ่านเข้าไปในระบบเครือข่าย



รูปที่ 3.7 แสดงการทำงานของโปรแกรมเทลเนตผู้ขอรับบริการที่มีการเข้ารหัสข้อมูล

รูปที่ 3.8 แสดงการทำงานของโปรแกรมเทลเนตผู้ให้บริการ ในการนำระบบการเข้ารหัสข้อมูล มาใช้ในการสร้างความปลอดภัยให้กับข้อมูลก่อนส่งผ่านเข้าไปในระบบเครือข่าย

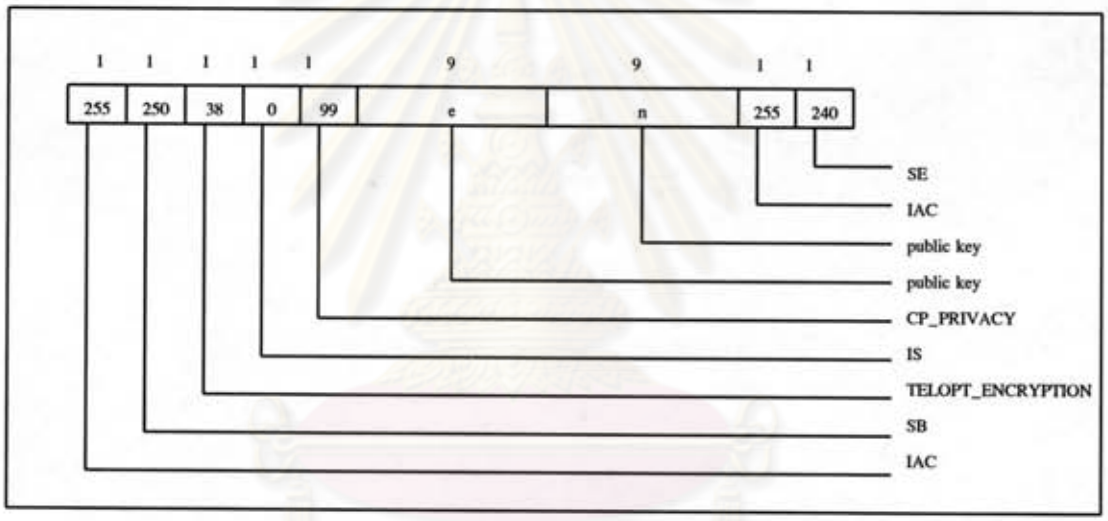


รูปที่ 3.8 แสดงการทำงานของโปรแกรมเทลเนตผู้ให้บริการที่มีการเข้ารหัสข้อมูล

1. กระบวนการเข้ารหัสเซสชั่นคีย์ กระบวนการเข้ารหัสเซสชั่นคีย์โดยการใช้การเข้ารหัสข้อมูลด้วยคีย์สาธารณะของ RSA ประกอบด้วยค่าพารามิเตอร์ดังต่อไปนี้

1) คีย์สาธารณะของ RSA (e,n)
2) คีย์ส่วนตัวของ RSA (d,n)
3) เซสชั่นคีย์ IDEA

เริ่มต้นส่งคีย์สาธารณะจากโปรแกรมเทลเนตผู้ขอรับบริการไปยังโปรแกรมเทลเนตผู้ให้บริการ โดยมีรูปแบบดังรูปที่ 3.9

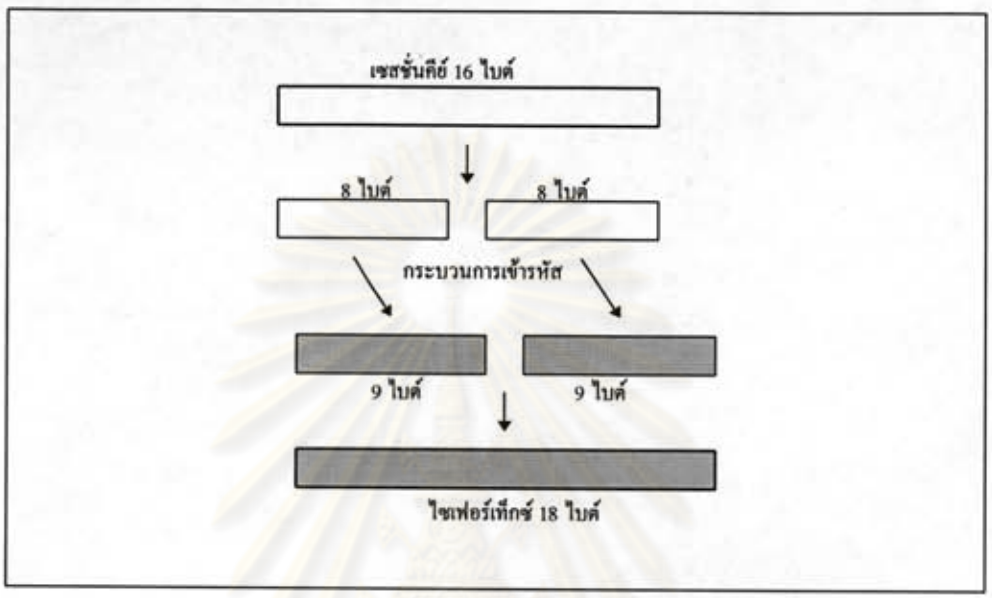


รูปที่ 3.9 แสดงรูปแบบข้อมูลของการส่งคีย์สาธารณะ

เมื่อโปรแกรมเทลเนตผู้ให้บริการได้รับคีย์สาธารณะ ทำการสร้างเซสชั่นคีย์ด้วยการสุ่ม (random) มีขนาดความยาว 16 ไบต์ (128 บิต) และนำมาเข้ารหัสด้วยคีย์สาธารณะ e และ n สำหรับการเข้ารหัสโดยใช้ RSA ได้ใช้ไลบรารี MPILIB (Multi-Precision Integer Library) ซึ่งเป็นไลบรารีในชุดโปรแกรมต้นฉบับของ PGP (Pretty Good Privacy) ของ Zimmerman MPILIB เป็นไลบรารีสำหรับการคำนวณตัวเลขที่มีขนาดใหญ่กว่าตัวแปรชนิด long ที่มีอยู่ในภาษาซี โดยการแทนค่าตัวเลขในรูปของอาร์เรย์ (array)

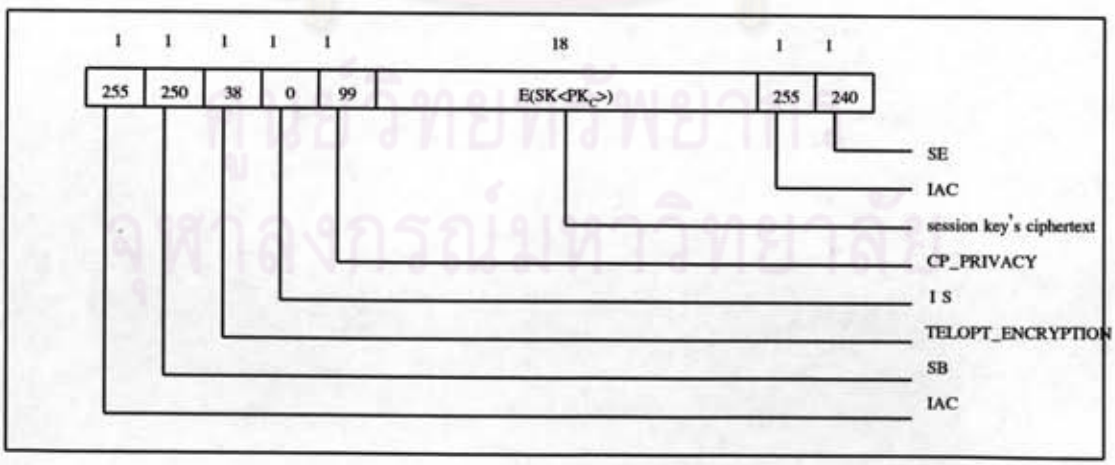
การเข้ารหัสด้วยวิธีของ RSA เป็นลักษณะบล็อคไซเฟอร์ โดยแบ่งข้อมูลออกเป็นชุดๆ ละ 8 ไบต์ (64 บิต) ผ่านการเข้ารหัส ด้วยคีย์ e และ n ที่มีขนาดคีย์ละ 9 ไบต์ (72 บิต) ได้ข้อมูล

ไซเฟอร์เท็กซ์ครั้งละ 9 ไบต์ ดังนั้นการเข้ารหัสเซสชั่นคีย์ 16 ไบต์ ได้ไซเฟอร์เท็กซ์ของเซสชั่นคีย์ขนาด 18 ไบต์



รูปที่ 3.10 แสดงการเข้ารหัสเซสชั่นคีย์

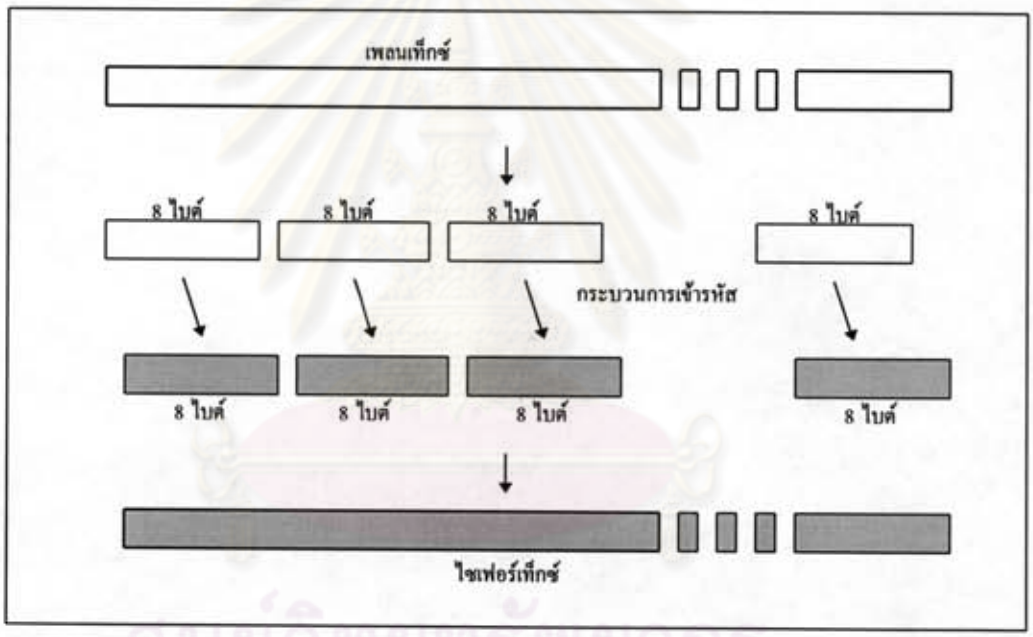
เมื่อเข้ารหัสเซสชั่นคีย์ด้วยคีย์สาธารณะเสร็จเรียบร้อย ทำการส่งเซสชั่นคีย์ที่ถูกเข้ารหัสนี้กลับไปให้ โปรแกรมเทลเนตผู้ขอรับบริการ มีรูปแบบของข้อมูลดังนี้



รูปที่ 3.11 แสดงรูปแบบข้อมูลของการส่งเซสชั่นคีย์ที่ถูกเข้ารหัส

เมื่อโปรแกรมเทลเนตผู้ขอรับบริการได้รับไซเฟอร์เท็กซ์ของเซสชั่นคีย์ นำมาถอดรหัสด้วยคีย์ส่วนตัวคือ d และ n โดยแบ่งข้อมูลที่ได้รับนำมาถอดรหัสครั้งละ 9 ไบต์ หลังจากถอดรหัสจะตัดออกหนึ่งไบต์ (มีค่าเป็น 0) เหลือเพลนเท็กซ์ 8 ไบต์ ดังนั้นถอดรหัสไซเฟอร์เท็กซ์ของเซสชั่นคีย์สองครั้ง ได้เซสชั่นคีย์ขนาด 16 ไบต์ เหมือนเช่นในโปรแกรมเทลเนตผู้ให้บริการ

2. กระบวนการเข้ารหัสข้อมูล กระบวนการเข้ารหัสข้อมูลที่ส่งผ่านระหว่างโปรแกรมเทลเนตผู้ให้บริการ และโปรแกรมเทลเนตผู้ขอรับบริการใช้ IDEA ในการเข้ารหัสข้อมูล โดยการทำงานเป็นแบบบล็อคไซเฟอร์ เข้ารหัสข้อมูลครั้งละ 8 ไบต์ (64 บิต) ได้ไซเฟอร์เท็กซ์ขนาด 8 ไบต์ (64 บิต)

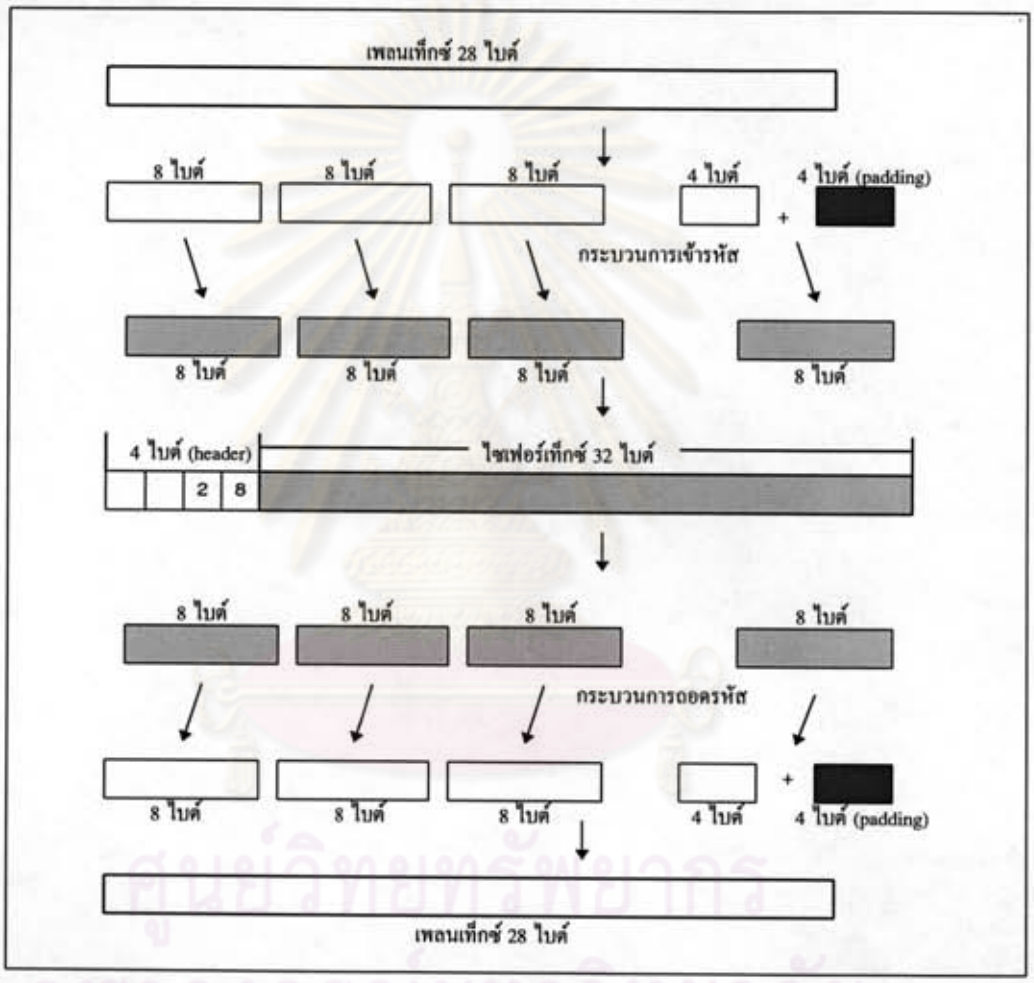


รูปที่ 3.12 แสดงการเข้ารหัสข้อมูลของ IDEA

การทำงานของ IDEA เป็นการผสมฟังก์ชั่นทางพีชคณิต 3 กลุ่มคือ XOR, ADD และ MUL ได้ใช้ฟังก์ชั่นการทำงานของ IDEA โดยใช้โปรแกรมต้นฉบับของ PGP ในแนวคิดและทฤษฎีได้อธิบายขั้นตอนของการเข้ารหัส และถอดรหัสของ IDEA ไว้

ขั้นตอนของการส่งและรับข้อมูลระหว่างโปรแกรมเทลเนตผู้ให้บริการ และโปรแกรมเทลเนตผู้ขอรับบริการ สำหรับข้อมูลที่มีขนาดไม่ครบบล็อคของการเข้ารหัสข้อมูลใน IDEA ต้องสุ่มข้อมูลใส่ให้ครบบล็อค (padding) ในสายของข้อมูลจริง และเข้ารหัสข้อมูลส่งไปยังอีกฝ่ายหนึ่ง เมื่อทำการถอดรหัส ตัดข้อมูลสุ่มที่ถูกแพดดิ้งออก ให้เหลือเฉพาะข้อมูลจริง เพื่อนำไปใช้งานต่อไป

เพราะการส่งและรับข้อมูลของโปรแกรมเทลเนตผู้ให้บริการ และโปรแกรมเทลเนตผู้ขอรับบริการ เป็นลักษณะที่ไม่สมมาตรกัน คือจำนวนข้อมูลที่เป็นผลลัพธ์ของกระบวนการเข้ารหัสเป็นจำนวนที่ไม่เท่ากับข้อมูลที่นำเข้ากระบวนการถอดรหัส วิทยานิพนธ์นี้ได้ออกแบบการเพิ่มส่วนหัว (header) เป็นจำนวนของข้อมูลที่เข้าสู่กระบวนการเข้ารหัสในสายข้อมูลที่ถูกเข้ารหัส และส่งไปให้อีกฝ่ายทำการถอดรหัสต่อไป รูปที่ 3.13 แสดงตัวอย่างของการเข้ารหัสข้อมูล



รูปที่ 3.13 แสดงการเพิ่มข้อมูลส่วนหัว และการ padding ข้อมูลสุ่มให้เต็มบล็อค

การออกแบบและพัฒนาระบบการเข้ารหัสที่ชั้นโปรแกรมประยุกต์ มีข้อดีในแง่การใช้งาน ผู้ใช้ไม่จำเป็นต้องเรียนรู้การใช้งานใหม่ การใช้งานในโปรแกรมใหม่เหมือนการใช้งานในโปรแกรมเดิมทุกประการ แต่เพิ่มประสิทธิภาพในการสร้างความปลอดภัยให้กับข้อมูลที่ส่งผ่านระบบเครือข่าย เป็นการสร้างความมั่นใจในการใช้งานให้กับผู้ใช้ได้เป็นอย่างดี