บทที่ 3

การออกแบบและพัฒนา

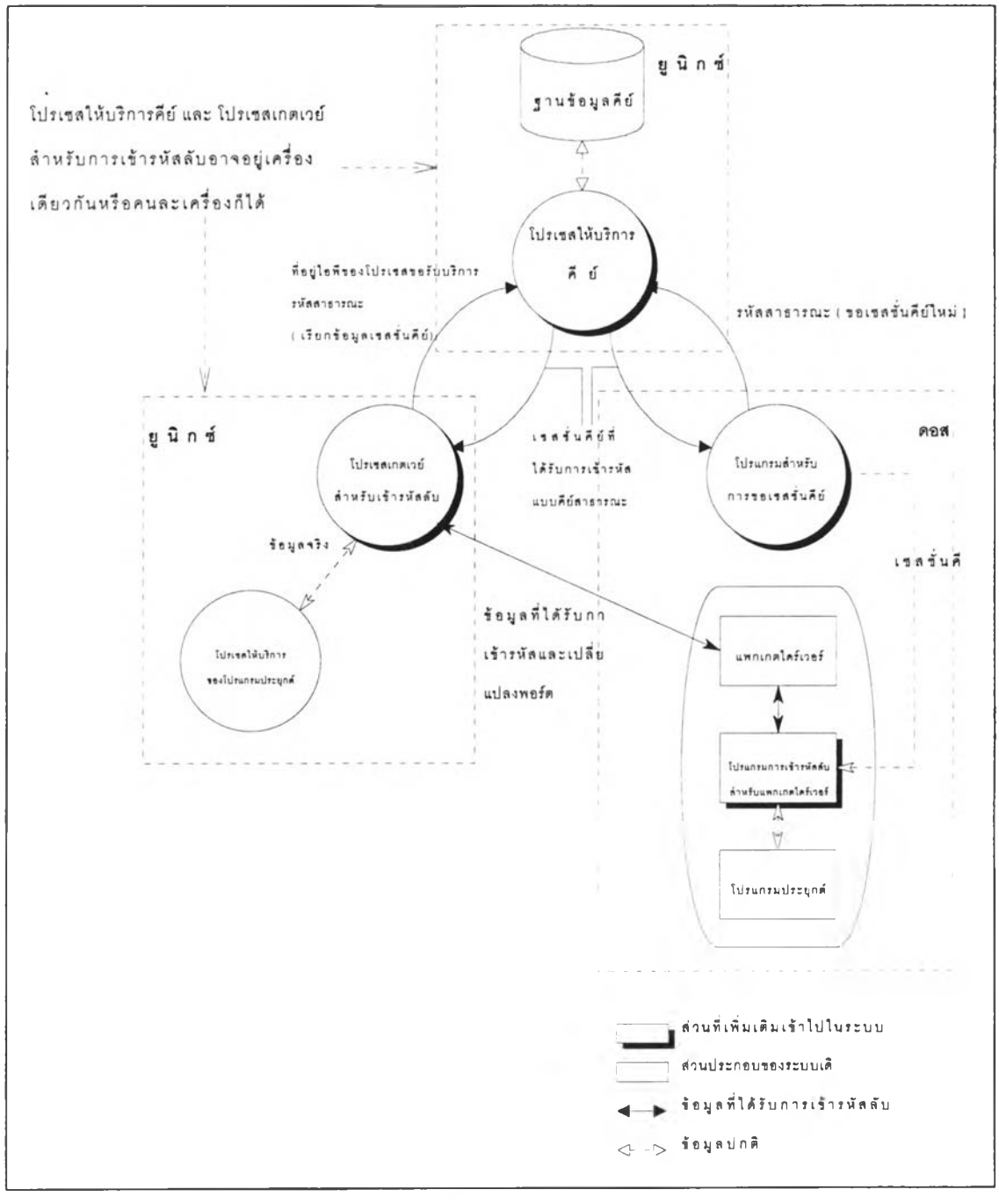
การออกแบบระบบงาน

ในการพัฒนาชั้นการเข้ารหัสลับสำหรับการใช้งานผ่านไปยังระบบยูนิกซ์จากเครื่องคอมพิวเตอร์ส่วนบุคคล แบ่งการพัฒนาออกเป็น 4 ส่วน บน 2 ระบบปฏิบัติการ คือ

บนระบบปฏิบัติการยูนิกซ์ ประกอบด้วย โปรเซสเกตเวย์สำหรับเข้ารหัสลับ (encryption gateway process) และ โปรเซสให้บริการคีย์ (key server process)

บนระบบปฏิบัติการดอส ประกอบด้วย โปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ (encryption program for packet driver) และ โปรแกรมสำหรับการขอเซสชั่นคีย์ (key generation request program)

แต่ละส่วนมีการทำงานที่เป็นอิสระต่อกัน แต่มีการทำงานที่สัมพันธ์กัน โดยความสัมพันธ์ของทั้ง 4 ส่วนสามารถแสดงได้ดังภาพที่ 3.1



รูปที่ 3.1: ความสัมพันธ์ของส่วนประกอบต่างๆ

โปรเซสเกตเวย์สำหรับเข้ารหัสลับ (encryption gateway process)

1) การทำงานของโปรเซสเกตเวย์สำหรับเข้ารหัสลับ ในส่วนนี้เป็นโปรเซสที่ทำงานบนระบบปฏิบัติการยูนิกซ์ ทำหน้าที่ เข้ารหัสและถอดรหัสข้อมูลระหว่างโปรเซสขอรับบริการจากดอสและ โปรเซสให้บริการบนยูนิกซ์

การทำงานของโปรเซสเกตเวย์สำหรับเข้ารหัสลับจะเป็นโปรเซสที่ตั้งรับข้อมูลที่พอร์ตปลอดภัย (secured port) เพื่อให้โปรเซสขอรับบริการจากดอส ที่ได้รับการเปลี่ยนแปลงพอร์ต และเข้ารหัสลับจากโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์มาเรียบร้อย ติดต่อเข้ามายังพอร์ตนี้แทนที่จะติดต่อไปยังพอร์ตของโปรแกรมให้บริการของโปรแกรมประยุกต์จริง

หลังจากที่ได้รับข้อมูลที่เข้ารหัสลับมาโปรเซสเกตเวย์สำหรับเข้ารหัสลับจะเปลี่ยนแปลงพอร์ตของข้อมูลไปยังพอร์ตของโปรแกรมประยุกต์ (application port) และถอดรหัสลับข้อมูลก่อนส่งไปยังโปรแกรมประยุกต์นั้นๆ

ในทางกลับกันเมื่อโปรแกรมประยุกต์จะส่งข้อมูลให้กับโปรเซสขอรับบริการบนดอส โปรแกรมประยุกต์ก็จะส่งข้อมูลที่ยังไม่ได้เข้ารหัสให้โปรเซสเกตเวย์ หลังจากนั้นโปรเซสเกตเวย์ก็จะเปลี่ยนแปลงพอร์ต และ เข้ารหัสลับก่อนส่งกลับไปให้กับโปรเซสขอรับบริการบนดอส

2) รายละเอียดขั้นตอนวิธีสำหรับโปรเซสเกตเวย์สำหรับเข้ารหัสลับ โปรเซสเกตเวย์สำหรับการเข้ารหัสลับเป็นโปรแกรมที่เขียนด้วยภาษาซี โดยใช้ชุดไลบรารีซ็อกเก็ต (socket library) ซึ่งเป็นไลบรารีสำหรับโปรแกรมการสื่อสารข้อมูลในระบบปฏิบัติการยูนิกซ์

2.1) ขั้นตอนรอรับการติดต่อจากโปรเซสขอรับบริการ

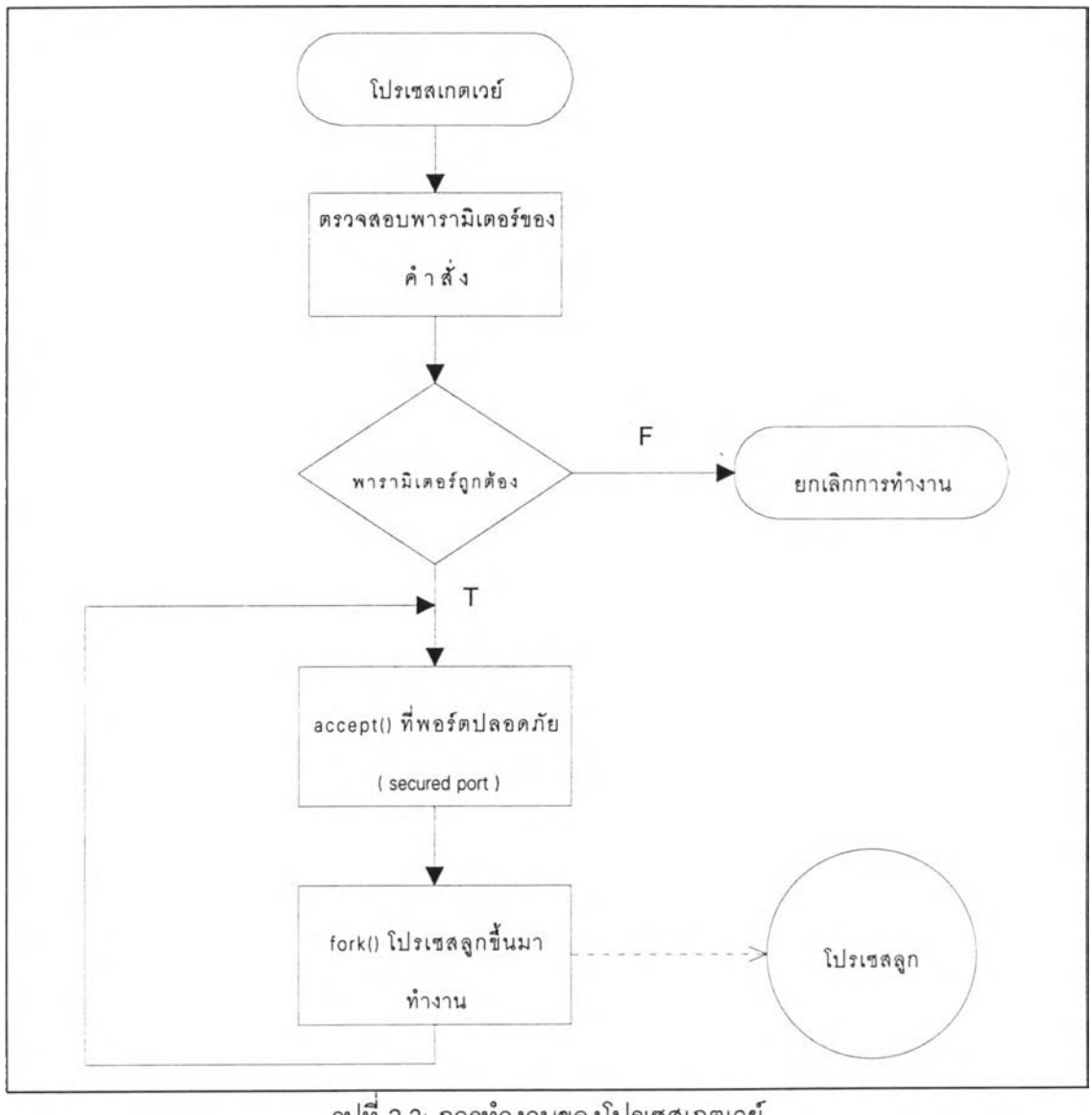
การทำงานของโปรแกรมเริ่มแรก จะอ่านค่าพารามิเตอร์ของคำสั่ง ซึ่งประกอบด้วย

- พอร์ตของโปรแกรมประยุกต์ (application port)
- พอร์ตที่ปลอดภัย (secured port)
- สำหรับเอฟทีพี ต้องใช้พอร์ตของข้อมูลที่ปลอดภัย (secured data port) เนื่องจากเอฟทีพีจะใช้งาน 2 พอร์ต ซึ่งจะได้กล่าวรายละเอียดต่อไป

เมื่อพารามิเตอร์ที่ส่งมาให้ไม่มีปัญหาใดๆ โปรแกรมจะเริ่มงาน โดยการ fork() โปรเซสออกมาเป็นลักษณะของโปรเซสเบื้องหลัง (background process) ทำงาน ในโปรเซสที่ถูก fork() ออกมาจะรอรับการติดต่อ (connect) จากโปรเซสขอรับบริการ โดยใช้ฟังก์ชัน accept()

เมื่อโปรเซสเกตเวย์ได้รับการติดต่อจากโปรเซสขอรับบริการแล้ว จะทำการ fork() โปรเซสขึ้นมารับการทำงานแทน และ โปรเซสแม่จะได้กลับไปรับการติดต่อจากโปรเซสขอรับบริการใหม่ ตามรูปแบบของโปรเซสให้บริการแบบทำงานพร้อมกัน

ลักษณะการทำงานของโปรเซสเกตเวย์สามารถแสดงได้ดังรูป



รูปที่ 3.2: การทำงานของโปรเซสเกตเวย์

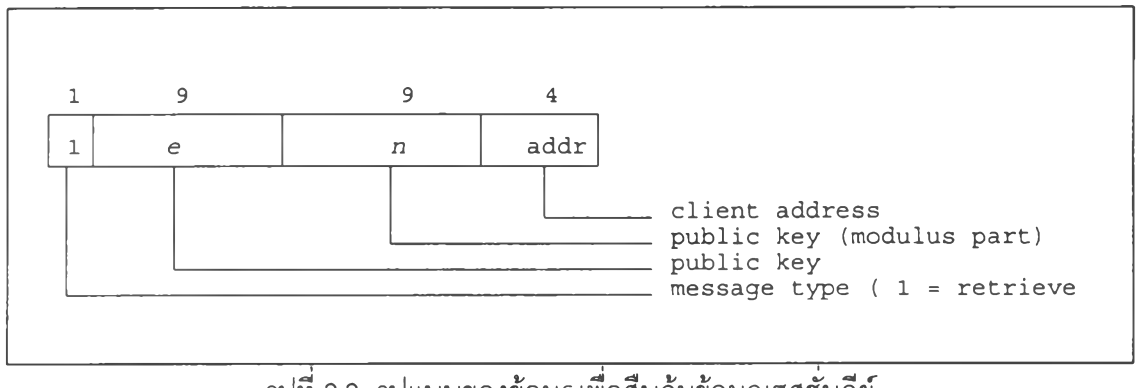
2.2) ขั้นตอนการสืบค้นข้อมูลจากโปรเซสให้บริการคีย์

โปรเซสเกตเวย์ที่ถูก fork() ออกมาจะตรวจสอบที่อยู่ของไอพี ของ โปรเซสขอรับบริการโดยใช้ฟังก์ชัน getpeername() เพื่อใช้เป็นข้อมูลสำหรับสืบค้นเซสชั่นคีย์จาก โปรเซสให้บริการคีย์

การสืบค้นเซสชั่นคีย์จากโปรเซสให้บริการคีย์นั้น โปรเซสเกตเวย์สำหรับการเข้ารหัสลับจะเป็นโปรเซสขอรับบริการของโปรเซสให้บริการคีย์ โดยโปรเซสเกตเวย์สำหรับเข้ารหัสลับจะอ่านพารามิเตอร์จากไฟล์ข้อมูล key.cfg ซึ่งเก็บค่าพารามิเตอร์ดังต่อไปนี้

- คีย์สาธารณะของ RSA *(e,n)*
- คีย์ลับของ RSA *(d,n)*
- ที่อยู่ไอพีของโปรเซสให้บริการคีย์
- พอร์ตของโปรเซสให้บริการคีย์

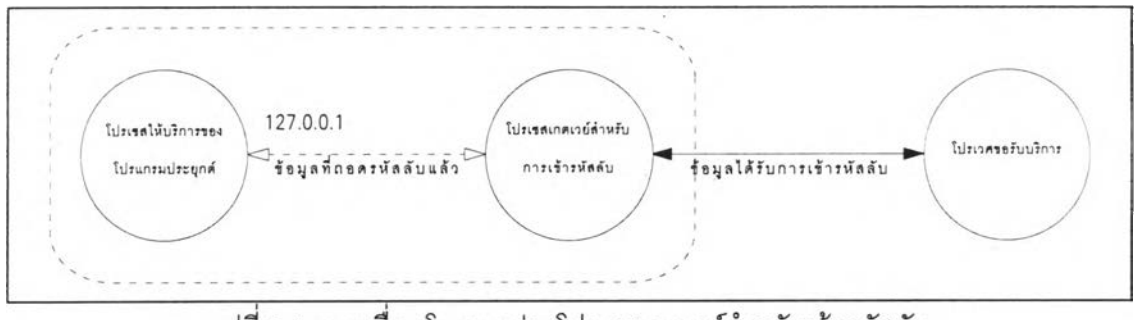
โปรเซสเกตเวย์สำหรับการเข้ารหัสลับจะ connect() ไปยังโปรเซสให้บริการคีย์ก่อน หลังจากนั้นจะส่งข้อมูลไปให้กับโปรเซสให้บริการคีย์มีรูปแบบดังนี้



รูปที่ 3.3: รูปแบบของข้อมูลเพื่อสืบค้นข้อมูลเซสชั่นคีย์

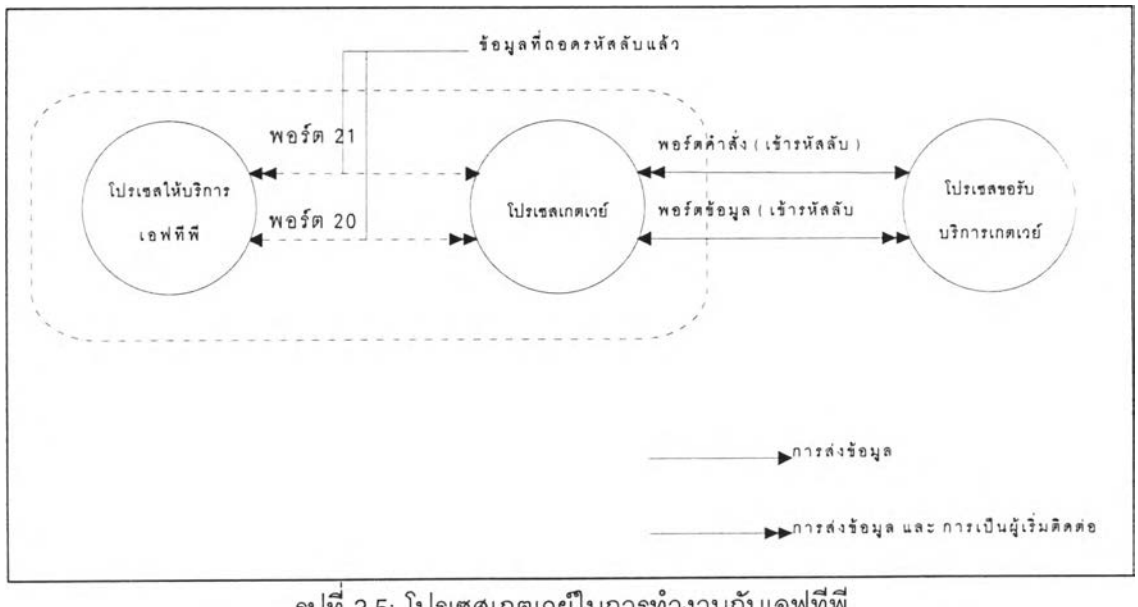
หลังจากนั้นจะได้รับเซสชั่นคีย์กลับมาจากโปรเซสให้บริการคีย์ ในรูปแบบของการเข้ารหัสโดยการใช้คีย์สาธารณะที่โปรเซสเกตเวย์ส่งไปให้ เมื่อโปรเซสเกตเวย์ได้รับเซสชั่นคีย์ที่ได้รับการเข้ารหัสแบบใช้คีย์สาธารณะแล้วจึงถอดรหัสลับของเซสชั่นคีย์โดยการใช้คีย์ลับก่อนเก็บข้อมูลเซสชั่นคีย์ไว้ใช้งาน สำหรับรายละเอียดการเข้ารหัสลับ และ ถอดรหัสลับของเซสชั่นคีย์จะกล่าวในหัวข้อโปรเซสให้บริการคีย์

2.3) ขั้นตอนการเชื่อมต่อกับโปรเซสให้บริการของโปรแกรมประยุกต์ หลังจากที่ได้สืบค้นข้อมูลคีย์จากโปรเซสให้บริการคีย์เป็นที่เรียบร้อยแล้ว โปรเซสเกตเวย์จะ connect() ไปยังโปรแกรมประยุกต์ตามพอร์ตที่ได้ระบุไว้ในพารามิเตอร์ของคำสั่ง และที่สำคัญคือจะต้องใช้ที่อยู่ของไอพีในการติดต่อเป็น 127.0.0.1 คือเป็นที่อยู่ที่เป็นเครื่องเดียวกันกับโปรเซสเกตเวย์ ซึ่งข้อมูลที่รับส่งกับที่อยู่นี้จะเป็นลักษณะของการวนกลับ (loopback) เนื่องจากไม่มีการส่งออกไปตามเครือข่ายจริง จึงทำให้ไม่สามารถดักบันทึกข้อมูลได้ ลักษณะเป็นดังภาพ



รูปที่ 3.4: การเชื่อมโยงระหว่างโปรเซสเกตเวย์สำหรับเข้ารหัสลับ
และ โปรเซสให้บริการโปรแกรมประยุกต์

แผนภาพตามรูปที่ 3.4 สามารถใช้ได้กับโปรแกรมประยุกต์ส่วนใหญ่ แต่สำหรับเอฟทีพี มีการใช้งาน 2 พอร์ต นั่นคือ มีพอร์ตหนึ่งเป็นพอร์ตสำหรับรอรับคำสั่ง และ พอร์ตหนึ่งทำหน้าที่เป็นพอร์ตสำหรับรับส่งข้อมูล การทำงานของโปรเซสเกตเวย์สำหรับเอฟทีพีเป็นดังนี้



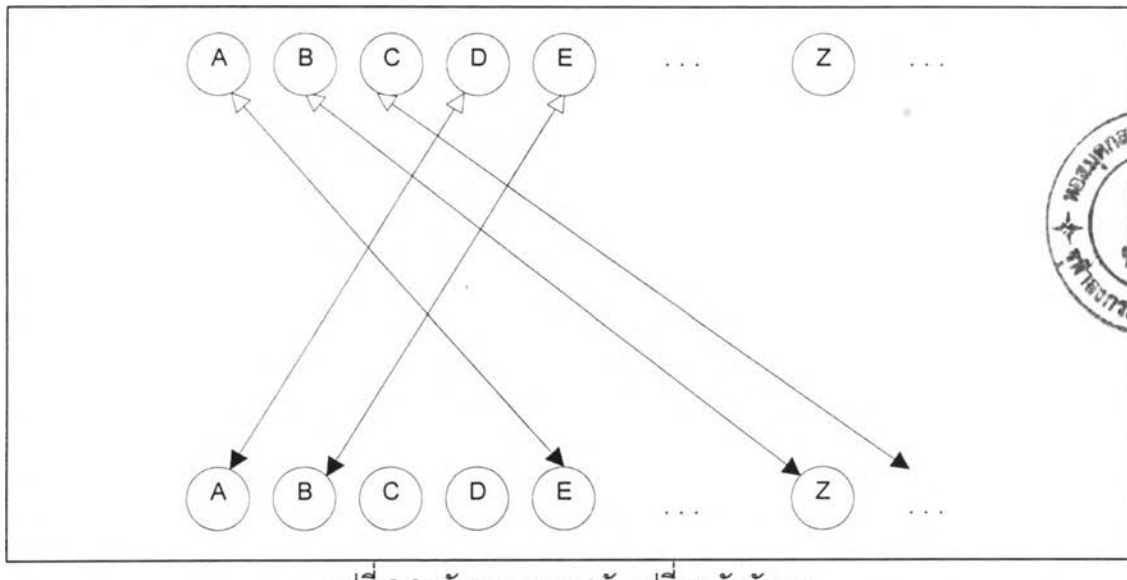
รูปที่ 3.5: โปรเซสเกตเวย์ในการทำงานกับเอฟทีพี

ในการทำงานปกติของโปรเซสให้บริการเอฟทีพีจะติดต่อไปยังโปรเซสขอรับบริการ เมื่อโปรเซสให้บริการต้องการรับหรือส่งข้อมูลกับโปรเซสขอรับบริการ รายละเอียดของการทำงานของเอฟทีพีกล่าวไว้ใน RFC 959 (Postel, 1985) โดยโปรเซสให้บริการจะได้รับคำสั่ง PORT xx,xx,xx,xx,xx,xx จาก โปรเซสขอรับบริการเพื่อบอกเลขพอร์ตที่จะให้โปรเซสให้บริการติดต่อไปเพื่อทำการรับส่งข้อมูล

สำหรับโปรเซสเกตเวย์ที่จะทำงานกับเอฟทีพีจะทำงานลักษณะเดียวกัน โดยโปรเซสเกตเวย์จะต้องกำหนดพอร์ตหนึ่งขึ้นมาเป็นพอร์ตสำหรับรับส่งข้อมูลกับโปรเซสให้บริการ เมื่อโปรเซสให้บริการต้องการรับส่งข้อมูลกับโปรเซสขอรับบริการ โปรเซสขอรับบริการจะส่งคำสั่ง PORT xx,xx,xx,xx,xx,xx (เข้ารหัสลับ) มาให้กับโปรเซสเกตเวย์ หลังจากนั้นโปรเซสเกตเวย์เปลี่ยนคำสั่งเป็น PORT 127,0,0,1,xx,xx ไปยังโปรเซสให้บริการ เพื่อให้โปรเซสให้บริการติดต่อมายังโปรเซสเกตเวย์สำหรับการรับส่งข้อมูล หลังจากที่โปรเซสเกตเวย์รับการติดต่อจากโปรเซสให้บริการแล้ว โปรเซสเกตเวย์จะติดต่อไปยังโปรเซสขอรับบริการตามพอร์ตที่โปรเซสขอรับบริการได้ส่งมาให้กับโปรเซสเกตเวย์ ข้อมูลที่รับส่งระหว่างโปรเซสให้บริการกับโปรเซสเกตเวย์จะอยู่บนเครื่องเดียวกันและเป็นเพลนเท็กซ์ ส่วนข้อมูลที่รับส่งระหว่างโปรเซสเกตเวย์และโปรเซสขอรับบริการจะเป็นไซเฟอร์เท็กซ์ เช่นเดียวกันกับพอร์ตที่ใช้รับส่งคำสั่ง

2.4)การเข้ารหัสลับสำหรับข้อมูลที่รับส่งระหว่างโปรเซสเกตเวย์กับโปรเซสขอรับบริการ เนื่องจากโปรเซสเกตเวย์ และ โปรเซสขอรับบริการซึ่งเกิดจากการเปลี่ยนแปลงแพกเกตที่รับส่งในแพกเกตไดรเวอร์ ทำงานคนละชั้นของโปรโตคอลแสตก ทำให้มีข้อจำกัดในการเข้ารหัสลับแบบกลุ่ม (block encryption) ซึ่งจะได้กล่าวรายละเอียดต่อไป

ดังนั้นในการเข้ารหัสลับข้อมูลที่รับส่งระหว่างโปรเซสเกตเวย์ และ โปรเซสขอรับบริการจึงทำได้ในลักษณะของการเข้ารหัสทีละตัวอักษร (byte encryption) โดยจะเป็นลักษณะของการสลับเปลี่ยนแบบสุ่ม (random permutation) ลักษณะดังภาพ



รูปที่ 3.6: ลักษณะการสลับเปลี่ยนตัวอักษร

คีย์ของการเข้ารหัสลับสำหรับระบบนี้เป็นคีย์ที่เป็นการจับคู่ระหว่างตัวอักษรเพลนเท็กซ์ และ ตัวอักษรไซเฟอร์เท็กซ์ ซึ่งจะได้มาด้วยการสุ่ม (random) โดยโปรแกรมให้บริการคีย์ซึ่งจะได้กล่าวถึงรายละเอียดในเรื่องของโปรเซสให้บริการคีย์

โปรเซสให้บริการคีย์ (key server process)

1) การทำงานของโปรเซสให้บริการคีย์ โปรเซสให้บริการคีย์ทำหน้าที่เป็นศูนย์กลางของการกระจายคีย์ (key distribution center) โดยโปรเซสให้บริการคีย์ทำหน้าที่ดังต่อไปนี้

- สร้างเซสชันคีย์ (generate session key) ให้กับ โปรเซสขอรับบริการบนดอส
- เก็บเซสชันคีย์ในฐานข้อมูลคีย์ (session key database)
- ส่งข้อมูลเซสชั่นคีย์ของโปรเซสขอรับบริการบนดอสให้กับ โปรเซสเกตเวย์สำหรับเข้ารหัสลับบนยูนิกซ์

ในการทำงานของระบบการเข้ารหัสลับสำหรับวิทยานิพนธ์นี้ จะต้องมีโปรเซสให้บริการคีย์ซึ่งทำงานบนระบบปฏิบัติการยูนิกซ์ โดยในระบบปฏิบัติการดอสจะมี โปรแกรมที่ทำหน้าที่ขอเซสชั่นคีย์จากโปรเซสให้บริการคีย์ เมื่อโปรเซสให้บริการคีย์สร้างคีย์ให้กับโปรแกรมจากดอสจะเก็บเซสชั่นคีย์ในฐานข้อมูลคีย์ โปรเซสเกตเวย์สำหรับการเข้ารหัสลับจะสืบค้นเซสชั่นคีย์จากโปรเซสให้บริการคีย์นี้เช่นกัน

การส่งข้อมูลคีย์ (key transferring) ใช้รูปแบบการเข้ารหัสแบบคีย์สาธารณะ (public key encryption) ในการส่งข้อมูล โดยโปรแกรมที่ต้องการคีย์จะส่งคีย์สาธารณะของตัวเองไปยังโปรแกรมให้บริการคีย์ โปรแกรมให้บริการคีย์จะส่งข้อมูลคีย์ที่เข้ารหัสด้วยคีย์สาธารณะที่ได้รับมา เมื่อโปรแกรมที่ต้องการคีย์ได้รับคีย์ที่เข้ารหัสโดยคีย์สาธารณะแล้ว จะถอดรหัสโดยคีย์ส่วนตัว (private key) ของตนเอง ด้วยวิธีนี้ก็จะทำให้การลักลอบดักข้อมูลคีย์ระหว่างการส่งในสายสื่อสารทำได้ยากขึ้น

2) รายละเอียดสำหรับโปรเซสให้บริการคีย์

2.1) ขั้นตอนวิธีสำหรับโปรเซสให้บริการคีย์ โปรเซสให้บริการคีย์ จะเป็นโปรเซสที่รอรับการติดต่อที่พอร์ตของโปรเซสให้บริการคีย์ ซึ่งเป็นพารามิเตอร์ที่ส่งมาจากคำสั่ง เมื่อรับการติดต่อ (ใช้ฟังก์ชัน accept()) จากโปรเซสขอรับบริการคีย์ แล้ว โปรเซสให้บริการคีย์จะ fork() โปรเซสขึ้นมาทำงานแทน ในลักษณะของโปรเซสให้บริการแบบทำงานพร้อมกัน

เมื่อโปรเซสให้บริการคีย์รับการติดต่อจากโปรเซสขอรับบริการแล้วจะรับข้อมูลจากโปรเซสขอรับบริการตามรูปแบบตามที่ได้กล่าวไว้ในหัวข้อ 2.2 โดยข้อมูลที่ได้รับจะประกอบด้วย

ไบต์ที่ 1 บอก ชนิดของบริการที่ต้องการ

ถ้าเป็น 0 หมายถึง โปรเซสขอรับบริการต้องการให้โปรเซสให้บริการคีย์สร้างคีย์ให้ใหม่

ถ้าเป็น 1 หมายถึง โปรเซสขอรับบริการต้องการถามข้อมูลจากโปรเซสให้บริการคีย์

ไบต์ที่ 2-10 บอก คีย์สาธารณะ *e*

ไบต์ที่ 11-19 บอก คีย์สาธารณะ *n*

ไบต์ที่ 20-23 บอก ที่อยู่ไอพีที่ต้องการถามเซสชั่นคีย์ (สำหรับชนิดบริการ ที่ 1)

สำหรับการทำงานในชนิดบริการ 0 (สร้างเซสชั่นคีย์ใหม่) โปรเซสให้บริการคีย์จะทำการสลับค่าของตัวอักษร 256 ตัวขึ้นมาใหม่แบบสุ่มโดยการใช้ฟังก์ชัน random() ใน 256 ตัวอักษรที่สลับขึ้นมาใหม่จะเป็นเซสชั่นคีย์ตามที่ได้กล่าวในหัวข้อ "การเข้ารหัสลับสำหรับข้อมูลที่รับส่งระหว่างโปรเซสเกตเวย์กับโปรเซสขอรับบริการ" เมื่อสร้างคีย์ใหม่เสร็จ โปรเซสให้บริการคีย์จะบันทึกข้อมูลคีย์ไว้ในไดเรกตอรี ./key โดยไฟล์จะถูกเก็บโดยใช้ที่อยู่ของไอพีเป็นชื่อไฟล์

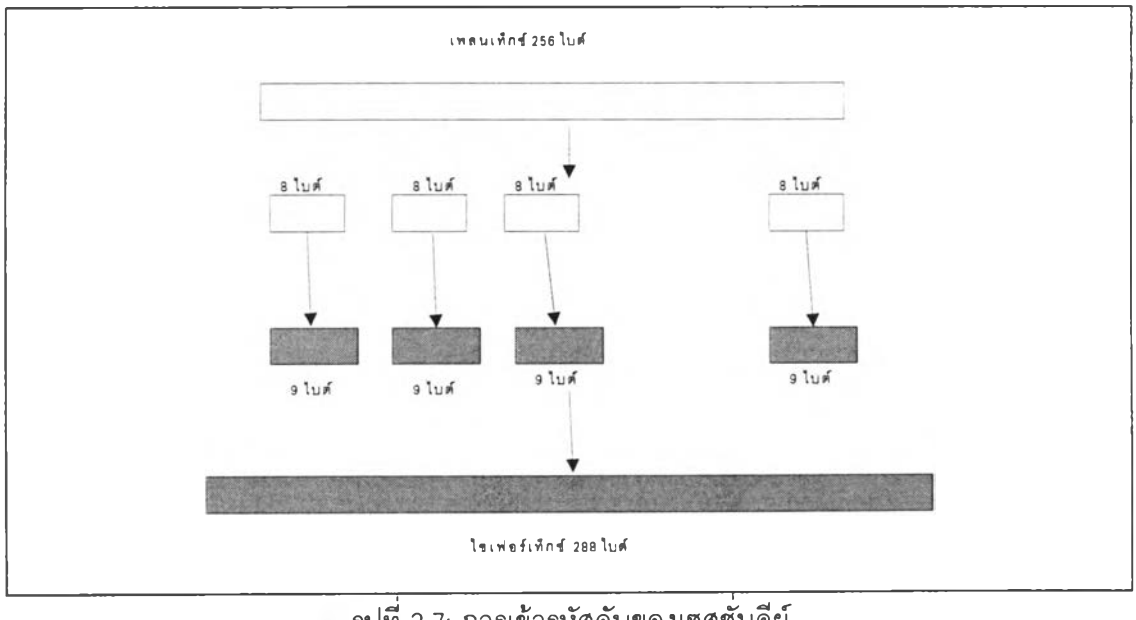
สำหรับการทำงานในชนิดบริการ 1 (เรียกค้นข้อมูลเซสชั่นคีย์) โปรเซสให้บริการคีย์จะใช้ข้อมูลที่ได้รับจากโปรเซสขอรับบริการในฟิลด์ที่อยู่ไอพี เพื่อนำมาค้นหาข้อมูลคีย์

หลังจากที่ทำการสร้างคีย์ หรือ เรียกค้นข้อมูลเซสชั่นคีย์เสร็จเรียบร้อย โปรเซสให้บริการคีย์จะส่งข้อมูลกลับไปให้กับโปรเซสขอรับบริการคีย์โดยการเข้ารหัสลับแบบใช้คีย์สาธารณะ RSA

2.2)การเข้ารหัสลับสำหรับโปรเซสให้บริการคีย์และโปรเซสขอรับบริการคีย์ สำหรับการเข้ารหัสลับสำหรับโปรเซสให้บริการคีย์ และ โปรเซสขอรับบริการคีย์ ได้ใช้ไลบรารี MPILIB (Multi-Precision Integer Library) ซึ่งเป็นไลบรารีในชุดโปรแกรมต้นฉบับของ PGP (Pretty Good Privacy) ของ Zimmermann MPILIB เป็นไลบรารีสำหรับการคำนวณตัวเลขที่มีขนาดใหญ่กว่า long ในภาษาซี โดยการแทนค่าตัวเลขในรูปของอาร์เรย์

การเข้ารหัสลับจะเป็นการใช้ขั้นตอนวิธีของ RSA โดยจะแบ่งข้อมูลเพลนเท็กซ์ออกเป็นทีละ 8 ไบต์ (64 บิต) ผ่านการเข้ารหัสลับ โดยการใช้ *e,n* มีขนาด 9 ไบต์ (72 บิต) ทำให้ได้ข้อมูลไซเฟอร์เท็กซ์แต่ละครั้งมีขนาด 9 ไบต์ ดังนั้น การเข้ารหัสลับสำหรับเซสชั่นคีย์ 256 ไบต์จะทำให้ได้ไซเฟอร์เท็กซ์ขนาด 288 ไบต์

เมื่อโปรเซสขอรับบริการได้รับข้อมูลเซสชั่นคีย์ที่ได้รับการเข้ารหัสลับมาจะถอดรหัสลับ โดยการแบ่งข้อมูลออกเป็นครั้งละ 9 ไบต์ หลังจากถอดรหัสลับแล้วจะตัดออกไปหนึ่งไบต์ (มีค่าเป็น 0) ซึ่งจะเหลือเป็นเพลนเท็กซ์ 8 ไบต์ ดังนั้นเมื่อทำการถอดรหัสลับครบจะได้ข้อมูลเพลนเท็กซ์ขนาด 256 ตามเดิม



รูปที่ 3.7: การเข้ารหัสลับของเซสชั่นคีย์

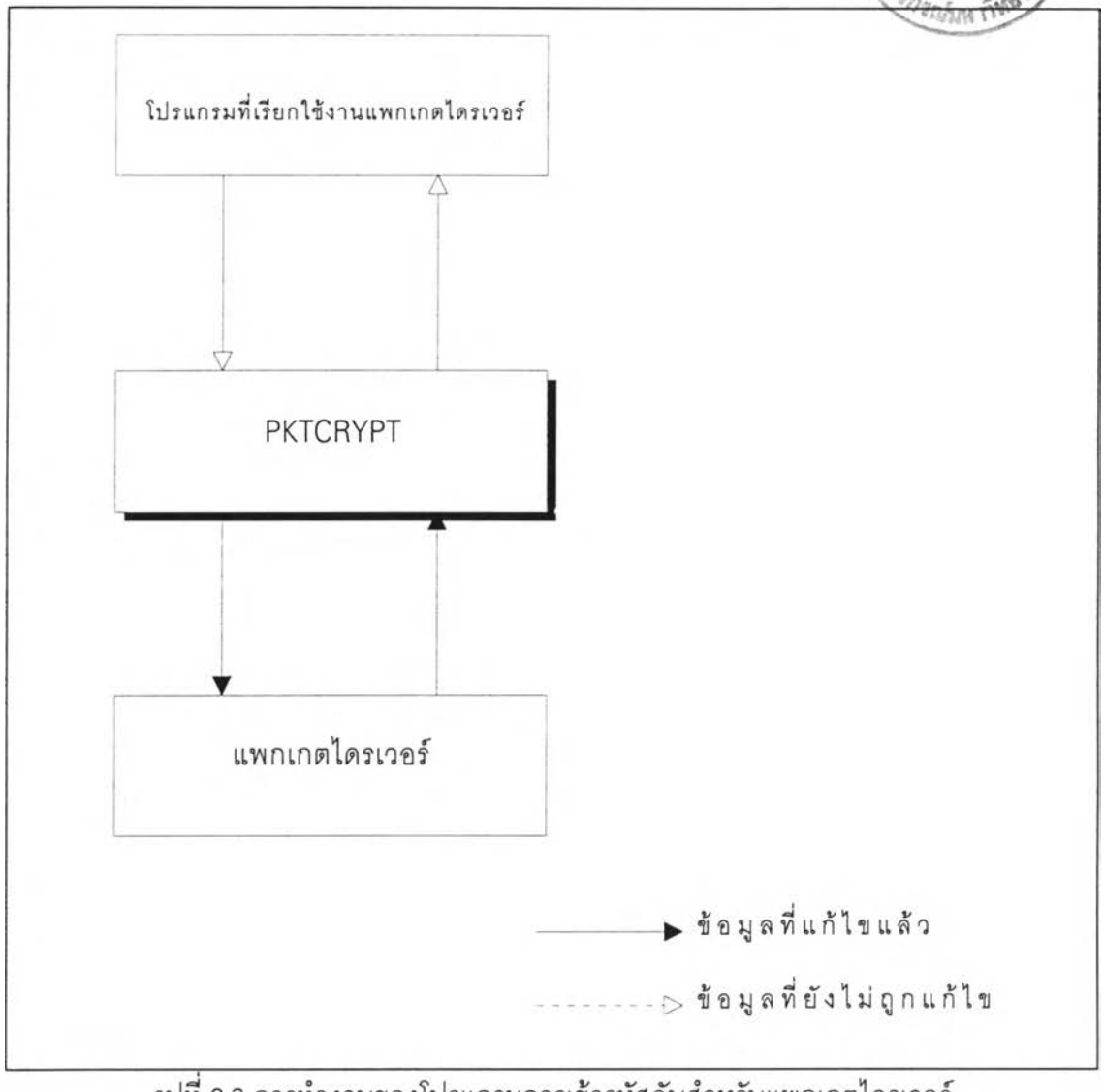
โปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์

1) การทำงานของโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ โปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ (encryption program for packet driver) ซึ่งในวิทยานิพนธ์นี้จะใช้ชื่อว่า "PKTCRYPT" เป็น โปรแกรมฝังตัวในระบบปฏิบัติการดอส (terminate and stay resident program) ทำหน้าที่เปลี่ยนแปลงข้อมูลในแพกเกตจากโปรแกรมที่ใช้งานแพกเกตไดรเวอร์ ก่อนที่จะส่งไปให้แพกเกตไดรเวอร์ และ ทำหน้าที่เปลี่ยนแปลงข้อมูลในแพกเกตที่ได้รับจากแพกเกตไดรเวอร์ก่อนที่จะส่งไปให้กับโปรแกรมที่ใช้งานแพกเกตไดรเวอร์ต่อไป

ข้อมูลที่ถูกเปลี่ยนแปลงในแพกเกต ได้แก่

- พอร์ตปลายทาง (destination port) กรณีที่ส่งข้อมูลไปยังสายสื่อสาร
- พอร์ตต้นทาง (source port) กรณีที่รับข้อมูลจากสายสื่อสาร
- เข้ารหัสลับและถอดรหัสลับข้อมูลในแพกเกตของทีซีพี

สำหรับลักษณะการทำงานของโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์สามารถแสดงได้ด้วยภาพที่ 3.8



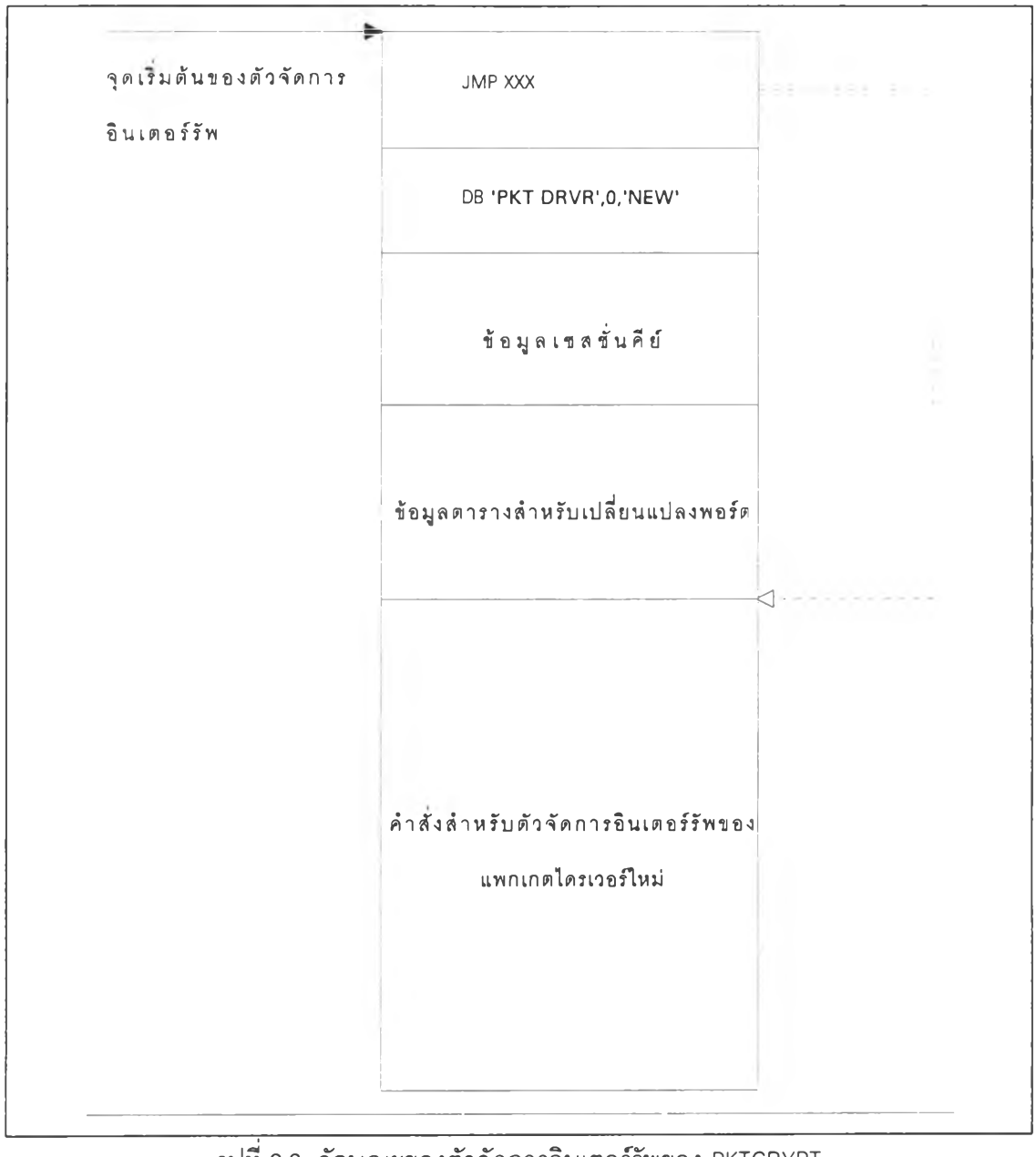
รูปที่ 3.8 การทำงานของโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์

2) รายละเอียดของโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ PKTCRYPT โปรแกรมที่ไปเปลี่ยนแปลงการทำงานของแพกเกตไดรเวอร์เดิม โดยการเปลี่ยนแปลงอินเตอร์รัพเวกเตอร์ (interrupt vector) ซึ่งชี้ไปยังตัวจัดการอินเตอร์รัพ (interrupt handler) ของแพกเกตไดรเวอร์เดิมมาเป็นของ PKTCRYPT ในวิทยานิพนธ์นี้จะใช้ภาษาซี (บอร์แลนด์ซี 3.1) และ ภาษาแอสเซมบลี (แทอร์โบแอสเซมเลอร์) ในการพัฒนา

2.1)การทำงานเริ่มต้นของโปรแกรมเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ PKTCRYPT มีการทำงานเริ่มต้นดังนี้

- ทำการค้นหาเลขที่อินเตอร์รัพ (interrupt number) ของแพกเกตไดรเวอร์เดิม โดยเริ่มจาก 0x60 ไปจนถึง 0x80 ถ้าในตัวจัดการอินเตอร์รัพใดมีข้อความ 'PKT DRVR',0 อยู่ที่ ไบต์ที่ 4 จากจุดเริ่ม แสดงว่าเป็น ตัวจัดการอินเตอร์รัพของแพกเกตไดรเวอร์
- เมื่อพบเลขที่อินเตอร์รัพแพกเกตไดรเวอร์แล้ว เก็บที่อยู่ของตัวจัดการอินเตอร์รัพนั้นไว้ แล้วเปลี่ยนตัวจัดการอินเตอร์รัพใหม่เป็นของ PKTCRYPT
- เปลี่ยนตัวจัดการอินเตอร์รัพของอินเตอร์รัพ 0x65 เป็นโปรแกรมสำหรับถอนโปรแกรมฝังตัวสำหรับโปรแกรมฝังตัว
- เมื่อเปลี่ยนอินเตอร์รัพทุกอย่างแล้ว PKTCRYPT ทำการฝังตัวในหน่วยความจำ และ ยกเลิกการทำงาน โดยการใช้บริการของดอส (INT 21) ที่ฟังก์ชัน 0x31

2.2) ลักษณะของตัวจัดการอินเตอร์รัพของ PKTCRYPT ตัวจัดการอินเตอร์รัพสำหรับ PKTCRYPT มีลักษณะดังภาพ



รูปที่ 3.9: ลักษณะของตัวจัดการอินเตอร์รัพของ PKTCRYPT

ส่วนประกอบสำหรับรูทีนการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ มีดังนี้

– จากจุดเริ่มของตัวจัดการอินเตอร์รัพ 3 ไบต์แรก จะเป็น คำสั่ง JMP ไปยังคำสั่งของตัวจัดการอินเตอร์รัพแพกเกตไดรเวอร์

– ส่วนถัดจากคำสั่ง JMP จะเป็นตัวอักษรว่า 'PKT DRVR',0,'NEW' โดย 'PKT DRVR',0 มีไว้เพื่อให้เป็นไปในลักษณะเดียวกับตัวจัดการอินเตอร์รัพของแพกเกตไดรเวอร์ทั่วไป ส่วน 'NEW' เป็นส่วนที่เพิ่มเข้าไปเพื่อให้ PKTCRYPT ตรวจสอบป้องกันการโหลดซ้ำ

- ข้อมูลของเซสชั่นคีย์สำหรับการเข้ารหัสลับ โดยมีรูปแบบดังนี้
  - ความยาวของคีย์ (2 ไบต์)
  - ตารางสำหรับการเข้ารหัสลับ (256 ไบต์)
  - ตารางสำหรับการถอดรหัสลับ (256 ไบต์)
- ตารางสำหรับการเปลี่ยนแปลงพอร์ต โดยรูปแบบเป็นดังนี้
  - ขนาดของตาราง (2 ไบต์)
  - ตารางสำหรับการเปลี่ยนแปลงพอร์ต มีลักษณะดังนี้

พอร์ตโปรแกรมประยุกต์ (2 ไบต์)	พอร์ตที่ปลอดภัย (2 ไบต์)

- คำสั่งของตัวจัดการอินเตอร์รัพเป็นคำสั่งสำหรับให้โปรแกรมที่ใช้งานแพกเกตไดรเวอร์เรียกใช้งาน

2.3) การทำงานของตัวจัดการอินเตอร์รัพของ PKTCRYPT ตัวจัดการอินเตอร์รัพใหม่ที่เปลี่ยนแปลงให้กับแพกเกตไดรเวอร์ จะเป็นการเปลี่ยนแปลงการทำงานของฟังก์ชันของในแพกเกตไดรเวอร์ 3 ฟังก์ชัน ได้แก่

- access_type() (AH = 2)
- release_type() (AH = 3)
- send_pkt() (AH = 4)

นอกจากนี้ยังเปลี่ยนแปลงการทำงานของ receiver() ซึ่งเป็นฟังก์ชันที่อยู่ในโปรแกรมที่เรียกใช้โดยแพกเกตไดรเวอร์เมื่อได้รับข้อมูลด้วย

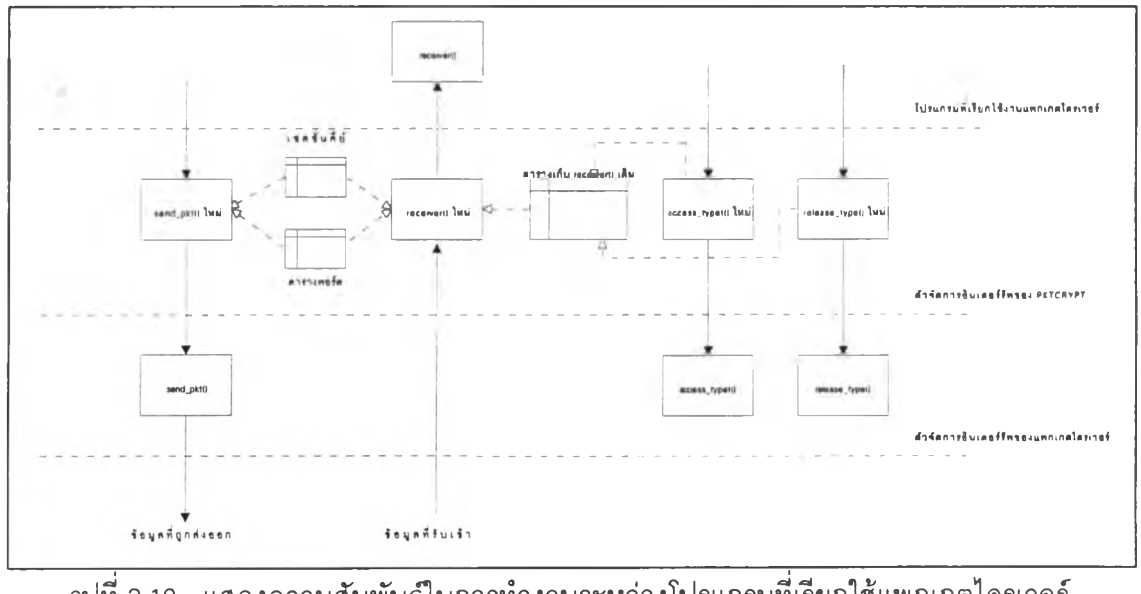
access_type() ของ PKTCRYPT มีการเปลี่ยนแปลงการทำงานก่อนที่จะส่งไปให้กับ access_type() ของแพกเกตไดรเวอร์ ดังนี้

- เก็บที่อยู่ของ receiver() ของโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์ไว้ในตารางเพื่อเรียกใช้
- เปลี่ยนแปลงค่าในตัวชี้ที่อยู่ของ receiver() (ES:DI) เป็นที่อยู่ของ receiver() ของ PKTCRYPT ก่อนที่จะเรียก access_type() ของแพกเกตไดรเวอร์

release_type() ของ PKTCRYPT มีการเปลี่ยนแปลงการทำงานก่อนที่จะส่งไปให้กับ release_type() ของแพกเกตไดรเวอร์ คือ จะคืนเนื้อที่ในตารางที่เก็บ receiver() ของแฮนเดิล (BX) ก่อนแล้วจึงเรียก release_type() ของแพกเกตไดรเวอร์

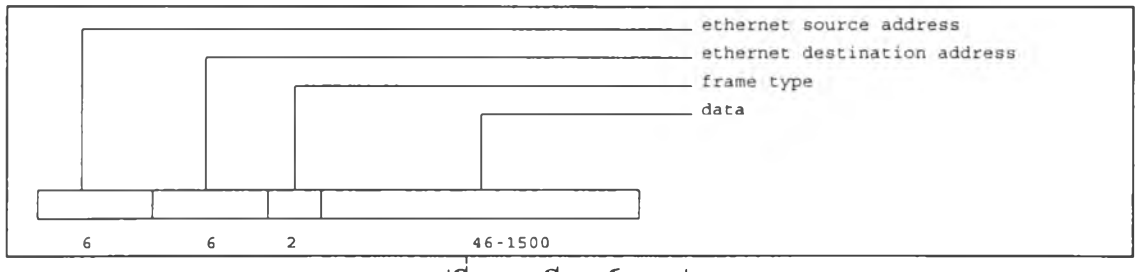
send_pkt() ของ PKTCRYPT จะทำการเปลี่ยนแปลงข้อมูลในแพกเกตก่อนที่จะส่งต่อไปให้กับ send_pkt() ของ แพกเกตไดรเวอร์ ซึ่งการแก้ไขข้อมูลในแพกเกตจะกล่าวถึงในหัวข้อถัดไป

สำหรับ receiver() ของ PKTCRYPT เป็นฟังก์ชันที่ทำงานก่อนที่จะส่งข้อมูลนั้นไปให้กับ receiver() ของ โปรแกรมที่เรียกใช้แพกเกตไดรเวอร์ การทำงานของ receiver() ของ PKTCRYPT จึงทำงานเลียนแบบการรับข้อมูลของ receiver() จริงๆ นั่นคือ receiver() ของ PKTCRYPT จะถูกใช้งาน 2 ครั้งเช่นเดียวกัน คือ ครั้งแรก (AX = 0) receiver() ของ PKTCRYPT จะส่งที่อยู่ของบัฟเฟอร์กลับไปให้กับผู้เรียก และ ครั้งที่สองจะรับข้อมูลมาแล้วจึงทำการเปลี่ยนแปลงข้อมูลก่อนที่จะส่งไปให้กับ receiver() ของโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์



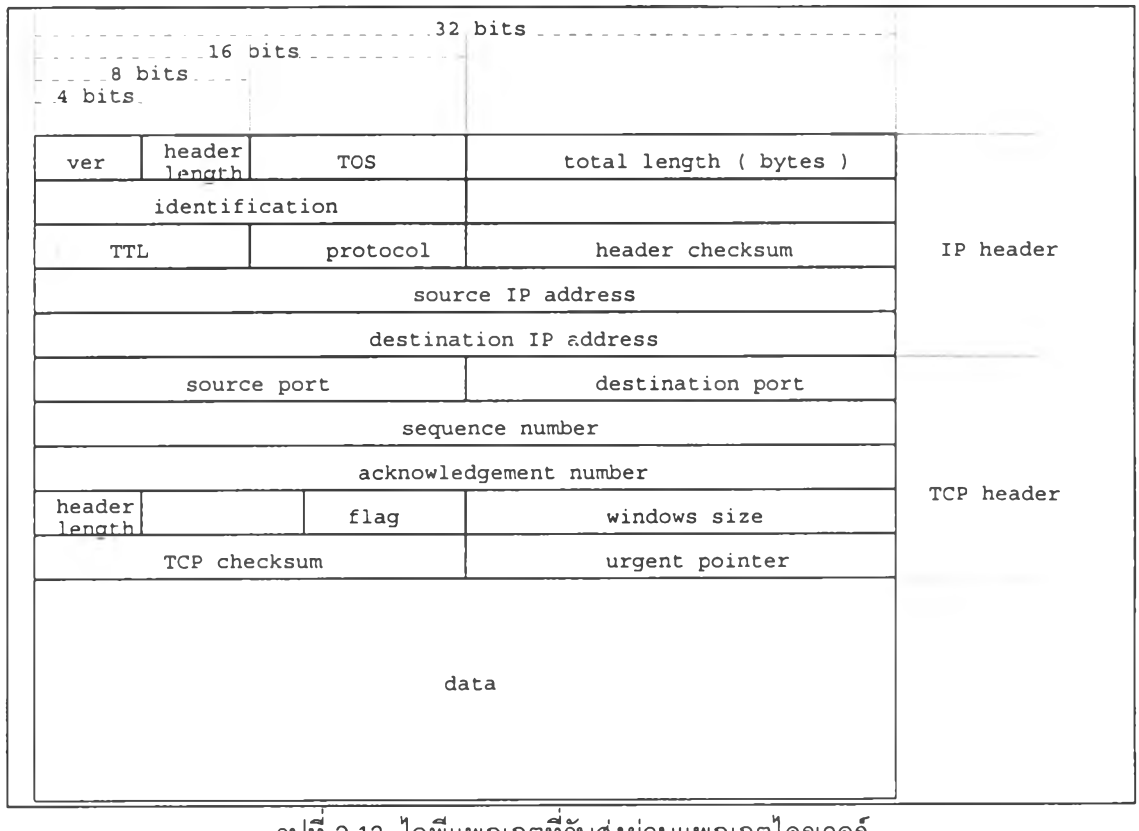
รูปที่ 3.10 : แสดงความสัมพันธ์ในการทำงานระหว่างโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์, PKTCRYPT และ แพกเกตไดรเวอร์

2.4) การการแก้ไขข้อมูลในแพกเกตสำหรับแพกเกตไดรเวอร์ ลักษณะของแพกเกตที่รับส่งในแพกเกตไดรเวอร์สำหรับอีเทอร์เนตเฟรม เป็นดังนี้



รูปที่ 3.11: อีเทอร์เนตเฟรม

ข้อมูลที่จะถูกแก้ไขจะเป็นข้อมูลที่เป็น ทีซีพีแพกเกต โดยจากรูปที่ 3.11 เป็นรูปของอีเทอร์เนตเฟรม โดยข้อมูลในส่วนของชนิดของเฟรม (frame type) จะมีค่าเป็น 0x800 สำหรับไอพีแพกเกต และในข้อมูลของไอพีแพกเกตจะเป็นดังนี้



รูปที่ 3.12: ไอพีแพกเกตที่รับส่งผ่านแพกเกตไดรเวอร์

การส่งข้อมูลทีซีพีแพกเกตจากโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์จะใช้ผ่านทาง send_pkt() ดังนั้นใน send_pkt() ของ PKTCRYPT จะต้องเปลี่ยนแปลงข้อมูลก่อนส่งต่อไปให้กับ send_pkt() ของแพกเกตไดรเวอร์ โดยการเปลี่ยนแปลงมีขั้นตอนดังนี้

- ตรวจสอบข้อมูลในตารางเปลี่ยนแปลงพอร์ตกับที่อยู่ของพอร์ตปลายทาง (destination port) เพื่อเปลี่ยนแปลงจากพอร์ตของโปรแกรมประยุกต์ไปเป็นพอร์ตที่ปลอดภัย

- เมื่อเปลี่ยนแปลงพอร์ตปลายทางแล้ว ให้เข้ารหัสลับข้อมูลโดยการใช้ตารางเซสชั่นคีย์ที่เก็บอยู่ในตัวจัดการอินเตอร์รัพของ PKTCRYPT โดยจะเข้ารหัสลับเฉพาะส่วนข้อมูล และจะต้องเป็นข้อมูลซึ่งมีแฟลก(flag) ของ ทีซีพีเฮดเดอร์ เป็น ACK หรือ ACK | PSH เท่านั้น

- หลังจากเปลี่ยนแปลงข้อมูลในทีซีพีแพกเกตเรียบร้อยแล้วให้คำนวณ CRC ของทีซีพีเฮดเดอร์ใหม่แล้วเปลี่ยนแปลงค่าใน ทีซีพีเช็กซัม (TCP checksum)

สำหรับการรับข้อมูลเข้ามา ตัวจัดการอินเตอร์รัพของ PKTCRYPT จะมีฟังก์ชัน receiver() ที่ทำงานก่อน receiver() ของโปรแกรมที่เรียกใช้แพกเกตไดรเวอร์ โดยจะมีการเปลี่ยนแปลงข้อมูลในทีซีพีแพกเกตในลักษณะที่กลับกันกับการส่งข้อมูล ดังนี้

- ตรวจสอบทีซีพีเช็กซัม (TCP checksum) ของข้อมูลที่ได้รับเข้ามาก่อนว่าถูกต้องหรือไม่ ถ้าไม่ถูกต้องจะไม่ทำการเปลี่ยนแปลงข้อมูลของแพกเกตนั้น

- ตรวจสอบข้อมูลในตารางเปลี่ยนแปลงพอร์ตกับที่อยู่ของพอร์ตต้นทาง (source port) เพื่อเปลี่ยนจากพอร์ตที่ปลอดภัยเป็นพอร์ตของโปรแกรมประยุกต์

- เมื่อเปลี่ยนแปลงพอร์ตปลายทางแล้ว ให้ถอดรหัสลับข้อมูลโดยการใช้ตารางเซสชั่นคีย์ที่เก็บอยู่ในตัวจัดการอินเตอร์รัพของ PKTCRYPT โดยจะเข้าถอดรหัสลับเฉพาะส่วนข้อมูล และจะต้องเป็นข้อมูลซึ่งมีแฟลก(flag) ของ ทีซีพีเฮดเดอร์ เป็น ACK หรือ ACK | PSH เท่านั้น

- หลังจากเปลี่ยนแปลงข้อมูลในทีซีพีแพกเกตเรียบร้อยแล้วให้คำนวณ CRC ของทีซีพีเฮดเดอร์ใหม่แล้วเปลี่ยนแปลงค่าใน ทีซีพีเช็กซัม (TCP checksum)

โปรแกรมสำหรับขอเซสชั่นคีย์(session key request program)

1) การทำงานของโปรแกรมสำหรับขอเซสชั่นคีย์ โปรแกรมสำหรับขอเซสชั่นคีย์เป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการดอสทำหน้าที่เป็นโปรแกรมขอรับบริการจากโปรเซสให้บริการคีย์บนยูนิกซ์แล้วส่งคีย์ที่ได้รับมาไปยัง โปรแกรมสำหรับเข้ารหัสลับสำหรับแพกเกตไดเวอร์

การทำงานของโปรแกรมสำหรับขอเซสชั่นคีย์ จะส่งคีย์สาธารณะของตัวเองไปให้กับโปรแกรมให้บริการคีย์ เมื่อโปรแกรมสำหรับให้บริการคีย์สร้างเซสชั่นคีย์แบบสุ่ม (random session key generate) แล้ว จะส่งเซสชั่นคีย์กลับมาพร้อมกับเข้ารหัสโดยใช้คีย์สาธารณะที่โปรแกรมสำหรับขอเซสชั่นคีย์ส่งให้ไป

ก่อนที่โปรแกรมนี้จะส่งคีย์ไปให้กับโปรแกรมการเข้ารหัสลับสำหรับแพกเกตไดรเวอร์ โปรแกรมจะถอดรหัสโดยการใช้คีย์ส่วนตัวตามลักษณะของการเข้ารหัสแบบใช้คีย์สาธารณะ

2) รายละเอียดของโปรแกรมสำหรับขอเซสชั่นคีย์ โปรแกรมสำหรับขอเซสชั่นคีย์พัฒนาโดยใช้ภาษาซี (บอร์แลนด์ซี) โดยการใช้ชุดไลบรารีสำหรับซ็อกเก็ตของ Watterloo TCP

การทำงานของโปรแกรมเริ่มต้นด้วยการตรวจสอบความถูกต้องพารามิเตอร์ของคำสั่ง ซึ่งจะต้องประกอบด้วย

- ที่อยู่ไอพีของโปรแกรมให้บริการคีย์
- พอร์ตของโปรแกรมให้บริการคีย์

โปรแกรมขอเซสชั่นคีย์จะตรวจสอบว่ามี PKTCRYPT อยู่ที่อินเตอร์รัพใด ก่อนที่จะทำงานต่อไปโดยการตรวจสอบข้อความ 'PKT DRVR',0,'NEW' ในไบต์ที่ 4 จากจุดเริ่มของตัวจัดการอินเตอร์รัพ

หลังจากนั้นโปรแกรมขอเซสชั่นคีย์จะติดต่อไปยัง โปรเซสให้บริการคีย์ตามพารามิเตอร์ของคำสั่ง เมื่อการติดต่อเรียบร้อย โปรแกรมขอเซสชั่นคีย์จะส่งข้อมูลคีย์สาธารณะซึ่งกำหนดไว้ใน key.cfg ไปให้กับโปรเซสให้บริการคีย์ตามรูปแบบทีของข้อมูลที่ได้กล่าวไว้ในหัวข้อการสืบค้นข้อมูลสำหรับโปรเซสให้บริการคีย์

เมื่อได้รับข้อมูลเซสชั่นคีย์แล้วโปรแกรมขอเซสชั่นคีย์จะถอดรหัสลับข้อมูลโดยใช้คีย์ลับแล้วจึงนำข้อมูลเซสชั่นคีย์ไปใส่ในส่วนที่เก็บข้อมูลคีย์ของตัวจัดการอินเตอร์รัพของ PKTCRYPT

หลังจากที่การทำงานเรียบร้อยโปรแกรมขอเซสชั่นคีย์ก็ปิดการติดต่อกับโปรแกรมให้บริการคีย์