การปรับปรุงวิธีการเข้ารหัสข้อมูลแบบมาตรฐาน

THE DEVELOPMENT OF DATA ENCRYPTION STANDARD

ยงยศ รัตะเสรี

YONGYOT RATASEREE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-9708-92-X

การปรับปรุงวิธีการเข้ารหัสข้อมูลแบบมาตรฐาน

THE DEVELOPMENT OF DATA ENCRYPTION STANDARD

ยงยศ รตะเสรี

YONGYOT RATASEREE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2547

ISBN 974-9708-92-X

THE DEVELOPMENT OF DATA ENCRYPTION STANDARD

YONGYOT RATASEREE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQURIEMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

ISBN 974-9708-92-X

หัวข้อวิทยานิพนธ์	การปรับปรุงวิธีการเข้ารหัสข้อมูลแบบมาตรฐาน
นักศึกษา	นายยงยศ รตะเสรี
รหัสนักศึกษา	42061106
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2547
อาจารย์ผู้ควบคุม	รศ. ดร. ปัญญา ฐิติมัชฌิมา

บทคัดย่อ

ปัจจุบันการจัดเก็บข้อมูลในคอมพิวเตอร์หรือการติดต่อสื่อสารในระบบคอมพิวเตอร์เป็นเรื่องที่สำคัญ หลายองค์กรจำเป็นต้องปกปิดข้อมูลที่เป็นความลับโดยใช้วิธีการเข้าถึงข้อมูลด้วยรหัสผ่าน วิธีการที่นิยมคือการเข้ารหัสลับ (cryptography หรือ Data Encryption) แบบ DES เพื่อให้ข้อมูลผิดเพี้ยนไปจากเดิม วิทยานิพนธ์ฉบับนี้ นำเสนอถึงเทคนิคการปรับปรุงอัลกอริทึมการเข้ารหัสและถอดรหัสข้อมูล เพื่อเพิ่มประสิทธิภาพของการเข้ารหัสข้อมูลที่ดีกว่าแบบ Data Encryption Standard (DES) เทคนิคใหม่นี้จะเพิ่มฟังก์ชั่นพิเศษเข้าไปใน DES จุดประสงค์เพื่อต้องการลดจำนวนรอบ (cycle) ของการสลับสับเปลี่ยน (Permutation หรือ Transposition) และการแทนที่ (Substitution) ข้อมูล เพื่อทำให้กระบวนการเข้ารหัสและถอดรหัสข้อมูลสามารถทำได้รวดเร็วขึ้น ฟังก์ชั่นพิเศษนี้จะเพิ่มความปลอดภัยให้กับข้อมูลมากยิ่งขึ้นด้วยจำนวนกุญแจรหัสที่เพิ่มขึ้นเป็น 2 เท่าต่อจำนวนรอบของการเข้ารหัสในแต่ละรอบ และสามารถตรวจสอบการลักลอบถอดรหัสข้อมูลที่ปลายทางได้ เมื่อฟังก์ชันพิเศษตรวจพบจะทำการหน่วงเวลาให้วงจรถอดรหัสทำงานช้าลงด้วยเวลาหน่วงที่คงที่ ทำให้การลักลอบถอดรหัสทำได้ยากยิ่งขึ้น

Thesis Title	The development of Data Encryption Standard
Student	Mr.Yongyot Rataseree
Student ID.	42061106
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2004
Thesis Advisor	Assoc. Prof. Dr. Panya Tithimushima

ABSTRACT

The present, storing data on computers or communication by computer is an important issue. Many organizations keep their data secret by using passwords. The most popular method for ensuring privacy is cryptography or data encryption by DES.

This research will present the techniques for developing Algorithm for Encryption and Decryption to be more efficient than the Data Encryption Standard technique. (DES)

This new technique, including special functions in DES, will reduce the number of cycles of Permutation (Transposition) and Substitution of the data, making the process of encryption and decryption faster. These special functions will provide more security for information by doubling the code keys per cycle of encryption. Add on checking secretly decode from destination. When this special function detect hacking, it will delay circuit decoding with constant delay time that will make more difficulty to hacking encoding.

กิตติกรรมประกาศ

ขอขอบพระคุณ บัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ให้โอกาสทางด้านการศึกษา และ รศ. ดร. ปัญญา ฐิติมัชฌิมา ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่ให้ความกรุณาในคำปรึกษาที่ดีเยี่ยม โดยเฉพาะอย่างยิ่งสำนักงานคณะกรรมการการอาชีวศึกษา (เดิมคือกรมอาชีวศึกษา) ที่เปิดโอกาสให้ใช้เวลาราชการบางส่วนสำหรับการศึกษาในครั้งนี้ อีกทั้งอาจารย์ประจำแผนกวิชาช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคฉะเชิงเทราและวิทยาลัยเทคนิคภูเก็ตที่คอยซักถามและให้กำลังใจด้วยดีตลอด ผศ.วิสุทธ์ อธิพรธรรม คณะครุศาสตร์อุตสาหกรรม และ ผศ.สักรียา ชิตวงศ์ ภาควิชาเทคโนโลยีการวัดคุมที่ให้คำแนะนำและคอยให้กำลังใจเรื่อยมา

ขอขอบคุณน้องสาวที่แสนดีที่คอยกระตุ้น-เตือนและให้กำลังใจเพื่อให้งานเสร็จลุล่วงตามกำหนดเวลา

สุดท้ายขอขอบพระคุณพ่อและแม่ที่ให้โอกาสทางด้านการศึกษา คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ของมอบแด่ผู้ที่มีพระคุณ โดยเฉพาะอย่างยิ่ง คุณพ่อ เฉ่งเตี๋ยน รตะเสรี (ที่ล่วงลับไปแล้วโดยมิทันได้ตอบแทนพระคุณอันยิ่งใหญ่ของท่าน)

ยงยศ รตะเสรี

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ที่มาของปัญหา

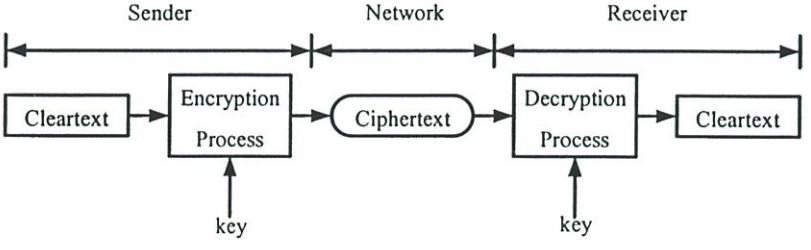
การป้องกันเอกสารที่สำคัญไม่ให้ผู้อื่นล่วงรู้นั้นมีมานานกว่า 2,000ปีมาแล้ว ตั้งแต่สมัยอียิปต์ โดยใช้ในการป้องกันเอกสารสำคัญทางด้านการทหาร ปัจจุบันมีการใช้คอมพิวเตอร์ในการจัดเก็บข้อมูลต่างๆ มีการคิดค้นออกแบบโปรแกรมใหม่ๆ ขึ้น เพื่อให้เหมาะสมกับงานที่ก้าวหน้า ข้อมูลเหล่านี้ถือว่าสำคัญมาก ในบางครั้งเจ้าของไม่ต้องการให้ข้อมูลถูกนำไปเผยแพร่ ความปลอดภัยของระบบคอมพิวเตอร์จึงเป็นหัวข้อสำคัญสำหรับผู้เชี่ยวชาญทางด้านคอมพิวเตอร์ จึงมักจะมีข่าวอยู่เป็นระยะๆ ว่า มีผู้เจาะเข้าไปในระบบคอมพิวเตอร์ที่สำคัญๆ ได้เสมอ

การเข้ารหัสลับ (cryptography) คือการแปลงข้อมูลปกติที่อ่านเข้าใจได้ ให้เปลี่ยนไปอยู่ในรูปแบบที่ยากแก่การอ่าน โดยบุคคลอื่นๆที่ไม่มีรหัสพิเศษหรือคีย์ (key) ที่ใช้ในการถอดรหัสลับ แต่สำหรับผู้ที่มีคีย์ที่ใช้ในการถอดรหัสลับ จะสามารถนำข้อมูลที่ถูกเข้ารหัสลับแล้วนั้นมาเปลี่ยนกลับให้อยู่ในรูปแบบเดิมที่สามารถอ่านได้

คีย์ คือ คำ หรือข้อความ หรือชุดของตัวอักษรที่ใช้สำหรับเข้ารหัสลับหรือถอดรหัสข้อมูล ความปลอดภัยของข้อมูลที่ใช้ระบบรักษาความปลอดภัยที่ดีนั้น จะขึ้นอยู่กับคีย์เพียงอย่างเดียวเท่านั้น

การเข้ารหัสลับแบ่งเป็น2ประเภทใหญ่ๆ คือ แบบสมมาตร (Symmetry) หรือคีย์แบบเดี่ยว (Single-key) และแบบอสมมาตร (Asymmetry) หรือคีย์แบบสาธารณะ (Public-key)

การเข้ารหัสลับแบบคีย์เดี่ยว เป็นกระบวนการเข้ารหัสลับข้อมูลที่ใช้คีย์เดียวกันสำหรับการเข้ารหัสและการถอดรหัส ระบบคีย์เดี่ยวที่ดีและปลอดภัยที่สุดคือ ระบบของ DES (Data Encryption Standard)



รูปที่ 1.1 หลักการทั่วไปของการเข้าและถอดรหัสลับ

หลักการพื้นฐานทั่วไปของการเข้ารหัสลับข้อมูลแสดงในรูปที่1.1 ข้อมูลเข้าเรียกว่า Cleartext หรือ Plaintext เมื่อผ่านกระบวนการเข้ารหัสลับ (Encryption Process) โดยใช้คีย์ จะทำให้ได้ข้อมูลที่เป็นข้อมูลลับเรียกว่า Ciphertext ซึ่งมีขนาดเท่าข้อมูลปกติส่งเข้าไปในเครือข่าย โดยบุคคลอื่นไม่สามารถอ่านเข้าใจได้ ทางด้านรับก็จะผ่านกระบวนการถอดรหัสลับ (Decryption Process) ซึ่งจะต้องใช้คีย์ในการถอดรหัสลับเช่นเดียวกัน จึงจะได้ข้อมูล Cleartext หรือ Plaintext ที่สมบูรณ์และอ่านเข้าใจได้กลับคืนมา

1.2 ปัญหาที่กำลังศึกษาและวิจัย

เนื่องจากการเข้ารหัสแบบมาตรฐานของ Data Encryption Standard (DES) มีความซับซ้อนของจำนวนรอบ (cycle) ในการการสลับสับเปลี่ยน (Permutation หรือ Transposition) ข้อมูลและการแทนที่ข้อมูล (Substitution) สูง จุดประสงค์ก็เพื่อให้ข้อมูลที่ถูกเข้ารหัสมีความปลอดภัยสูงขึ้น ทำให้กระบวนของการการเข้ารหัสและถอดรหัสข้อมูลสามารถทำได้ช้า ดังนั้นอุปกรณ์ที่ใช้ในกระบวนการจะต้องมีความเร็วในการทำงานที่สูงมากๆ เพื่อให้กระบวนการการเข้ารหัสและถอดรหัสข้อมูลสามารถทำได้รวดเร็วที่เวลาจริงโดยไม่จำเป็นที่จะต้องใช้อุปกรณ์ที่มีราคาแพงๆ จึงจำเป็นที่จะต้องปรับปรุงมาตรฐานของ DES ให้มีความซับซ้อนน้อยลงแต่ยังคงมีระดับความปลอดภัยของข้อมูลเท่าเดิมหรือเพิ่มความปลอดภัยสูงขึ้นกว่าเดิม

1.3 เหตุใดปัญหานี้ถึงน่าสนใจ

ในปัจจุบันการติดต่อสื่อสารนับว่ามีความจำเป็นอย่างมาก ข้อมูลที่ใช้ติดต่อสื่อสารบ่อยครั้งที่มีความสำคัญและเป็นความลับสุดยอด หากปล่อยให้ข้อมูลข่าวที่สำคัญถูกลักลอบขโมย อาจจะทำให้เกิดความเสียแก่หน่วยงานหรือองค์กรของผู้ส่งข้อมูลได้ ข้อมูลข่าวสารที่ใช้ในการสื่อสารเหล่านั้นจะเป็นข้อมูลที่อยู่ในรูปเชิงเลข ซึ่งจะถูกส่งผ่านไปยังโครงข่ายสื่อสารสาธารณะ หรือโครงข่ายเฉพาะกิจ (leased line)ไปยังปลายทาง เมื่อข้อมูลที่สำคัญเหล่านั้นมีระดับการรักษาความปลอดภัยที่ต่ำ สามารถที่จะถูกโจรกรรมได้ง่าย การเพิ่มความปลอดภัยให้กับข้อมูลข่าวสารที่ส่งผ่านโครงข่ายสื่อสารสาธารณะหรือโครงข่ายเฉพาะกิจ สามารถทำได้ด้วยการเปลี่ยนแปลงข้อมูลข่าวสารให้แตกต่างไปจากเนื้อหาเดิม เพื่อปกปิดข้อมูลที่แท้จริง โดยวิธีการที่เรียกว่าการเข้ารหัสลับทางด้านภาคส่งสัญญาณ ก่อนที่จะส่งข้อมูลผ่านโครงข่ายสื่อสารสาธารณะหรือโครงข่ายเฉพาะกิจไปยังภาครับสัญญาณปลายทาง โดยภาครับสัญญาณปลายทางจะนำข่าวสารที่ถูกปกปิดมาทำการถอดรหัสลับ เพื่อนำข้อมูลไปใช้งานต่อไป

1.4 การนำเสนอหลักการใหม่

ในงานวิจัยนี้จะนำเสนอถึงเทคนิคการปรับปรุงคุณสมบัติของ DES โดยเพิ่มฟังก์ชั่นพิเศษเข้าไปในขั้นตอนวิธีของ DES ฟังก์ชั่นพิเศษนี้คือฟังก์ชั่น y มีคุณสมบัติในการเพิ่มจำนวนกุญแจรหัสและทำให้จำนวนรอบในการสลับสับเปลี่ยนและการแทนที่ข้อมูลลดลง โดยฟังก์ชั่นที่เพิ่มเข้าไปจะมีคุณสมบัติที่เพิ่มขึ้นดังต่อไปนี้

1. ลดจำนวนรอบของกระบวนการเข้ารหัสและถอดรหัสข้อมูล
2. เพิ่มจำนวนของรหัสกุญแจให้มากขึ้น
3. ตรวจสอบการลักลอบถอดรหัสข้อมูล
4. หน่วงเวลาเมื่อเกิดการลักลอบถอดรหัสข้อมูล (ด้วยเวลาหน่วงที่คงที่)

1.5 โครงร่างของวิทยานิพนธ์

ภายในวิทยานิพนธ์ ประกอบด้วยรายละเอียดการวิจัยซึ่งแยกออกเป็น 5 บทดังต่อไปนี้

บทที่ 1 บทนำกล่าวถึง ที่มาของปัญหาที่เกิดขึ้น ปัญหาที่กำลังศึกษาและวิจัย เหตุใดปัญหานี้ถึงน่าสนใจ การนำเสนอหลักการใหม่ และเนื้อหาภายในวิทยานิพนธ์

บทที่ 2 หลักการและทฤษฎีของ Data Encryption แบบมาตรฐาน

บทที่ 3 นำเสนอหลักการออกแบบอัลกอริทึมที่ได้ออกแบบขึ้นมาใหม่ โดยใช้หลักการจากทฤษฎีในบทที่2 และทำการเขียนโปรแกรมจำลองการทำงานของหลักการที่ได้นำเสนอขึ้นมาใหม่

บทที่ 4 การทดลองหลักการที่ได้นำเสนอและผลที่ได้จากหลักการที่ได้นำเสนอโดยจะทำการเปรียบเทียบผลการทดลองที่ได้จากหลักการเดิมและหลักการที่ได้นำเสนอในวิทยานิพนธ์

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ เป็นการสรุปผลที่ได้จากการทดลอง และวิจารณ์ถึงข้อดีข้อเสีย พร้อมทั้งเสนอแนะแนวทางการวิจัยและแนวทางในการพัฒนาต่อไป

บทที่ 2

หลักการและทฤษฎี

วิธีการทำ Data Encryption มีหลายวิธี วิธีที่ยอมรับจำเป็นต้องมีการรักษาความปลอดภัยที่ดีพอ ในบทนี้จะกล่าวถึงความเป็นมาแนวคิดของการเข้ารหัสทั้งหลักเกณฑ์ที่ได้รับการยอมรับว่าเป็นการเข้ารหัสที่ดี ซึ่งนำไปสู่ระบบมาตรฐานของประเทศอเมริกา ยังกล่าวถึงรูปแบบการใช้และคุณสมบัติของ DES และแสดงการสร้างรหัสลับด้วยคอมพิวเตอร์แบบง่ายและทฤษฎีของ DES อย่างละเอียด

2.1 หลักพื้นฐานของการเข้ารหัส

ขอนำเสนอหลักพื้นฐานของการเข้ารหัสดังนี้

1. วิธีการจารกรรมข้อมูลกระทำได้หลายกรณี
2. ลักษณะของระบบการเข้ารหัสที่ดี
3. มาตรฐานการเข้ารหัสลับของข้อมูล
4. รูปแบบของการใช้ DES
5. คุณสมบัติของ DES
6. การสร้างรหัสลับด้วยคอมพิวเตอร์

2.1.1 วิธีการจารกรรมข้อมูลกระทำได้หลายกรณี

ปัจจุบันการส่งข้อมูลระยะไกลมีความสำคัญมากขึ้น โอกาสที่ข้อมูลจะถูกโจรกรรมก็มีมากด้วย ในการส่งข้อมูลข่าวสารจากผู้ส่ง S ไปตามสายส่ง T เพื่อไปยังผู้รับ R หากมีบุคคลภายนอก O มาทำการขัดขวางหรือจารกรรมข้อมูลข่าวสารนั้น เขาสามารถกระทำได้หลายกรณี คือ

1. ป้องกันไม่ให้ข้อมูลนั้นไปถึงผู้รับ R (Interruption)
2. ขโมยดูหรือฟังข้อมูลข่าวสารนั้น (Interception)
3. ยึดและแปรเปลี่ยนข้อมูลข่าวสาร (Modification)
4. การสอดแทรกข้อมูลข่าวสารเสมือนมาจากผู้ส่ง S (Fabrication)

2.1.2 ลักษณะของระบบการเข้ารหัสที่ดี

ระบบการเข้ารหัสลับที่ดีควรเป็นระบบที่มีลักษณะสมบัติดังนี้

1. การกระจายความถี่ของตัวอักษรและกลุ่มตัวอักษรที่เกิดขึ้นของข้อมูลที่เข้ารหัสแล้วควรมีค่าใกล้เคียงกัน

2. ระบบควรป้องกันไม่ให้คำหรือวลีของข้อมูลที่เกิดซ้ำๆ กัน เมื่อเข้ารหัสลับแล้วไปเป็นข้อมูลเข้ารหัสตัวเดิม ในที่นี้จะหมายถึงว่าระบบควรเข้ารหัสลับเป็นกลุ่มตัวอักษรแทนที่จะเป็นการเข้ารหัสแต่ละตัวอักษร

3.ระบบการเข้ารหัสลับ ควรจะเป็นระบบที่ให้ผู้ใช้เลือกขนาดของกุญแจเองได้ เพื่อป้องกันการแอบถอดรหัส และต้องใช้เวลายาวนานเพื่อการถอดรหัสลับนั้น

4.ระบบการเข้ารหัสลับที่ดี ควรเป็นระบบที่แก้ไขปัญหาอุบัติเหตุไม่ตั้งใจที่เกิดจากการใช้กุญแจที่ผิด การถอดรหัสลับข้อมูลทั้งๆ ที่ยังไม่ได้เข้ารหัสลับไว้เลย หรือการเข้ารหัสลับกับข้อมูลซ้อนกันสองครั้ง ในที่นี้หมายถึงว่าระบบการเข้ารหัสลับจะไม่มีผลเสียหายไม่ว่าเราจะเข้ารหัสลับหรือถอดรหัสลับ เช่น ถ้าเราใช้กุญแจรหัสลับที่ผิด ในการถอดรหัสเราก็เพียงทำการเข้ารหัสลับด้วยกุญแจตัวเดิม จากนั้นก็ถอดรหัสลับด้วยกุญแจตัวใหม่ที่ถูกต้อง

5.วิธีการเข้าและถอดรหัส ควรเป็นวิธีที่เข้าใจง่าย เพราะถ้าระบบยากแล้วอาจทำให้เกิดข้อผิดพลาดได้ง่าย หรืออาจทำให้ลืมได้ง่าย

6.ข้อผิดพลาดที่เกิดขึ้น ในการสร้างรหัสลับต้องไม่แผ่กระจายให้ข้อผิดพลาดเพิ่มมากขึ้น เพราะถ้ามีข้อผิดพลาดเกิดขึ้นเล็กน้อย ผู้รับอาจสามารถคาดเดาตัวอักษรที่ขาดหายหรือผิดพลาด

7.ขนาดของข้อมูลเข้ารหัสลับแล้วจะต้องไม่ยาวกว่าข้อมูลปกติเดิม

2.1.3 มาตรฐานการเข้ารหัสลับของข้อมูล

รัฐบาลสหรัฐอเมริกาโดยสำนักงานมาตรฐานแห่งชาติ (U.S. National Bureau of Standards – NBS) ได้เล็งเห็นความสำคัญของการเข้ารหัสเพื่อใช้งานทั่วไปมาตั้งแต่ปี พ.ศ. 2513 ได้พยายามให้เกิดวิธีการเข้ารหัสสาธารณะ (Public Encryption Algorithm) ที่มีกฎเกณฑ์ดังต่อไปนี้

1. มีระดับความปลอดภัยต่อข้อมูลสูง
2. มีการกำหนดขั้นตอนครบสมบูรณ์และเข้าใจได้ง่าย
3. ผู้ใช้งานทั่วไปสามารถนำไปใช้ได้อย่างอิสระ
4. สามารถปรับเปลี่ยนไปใช้เฉพาะงาน
5. สามารถสร้างเป็นระบบเข้ารหัสทางฮาร์ดแวร์ที่ง่ายและประหยัด

วิธีการที่เรียกว่า “Lucifer” ได้รับการพัฒนาปรับปรุงโดยบริษัท ไอบีเอ็ม จำกัด เพื่อ NBS ได้ใช้งานทั่วไป ต่อมาได้กลายเป็นระบบ DES ที่ย่อมาจาก Data Encryption Standard ซึ่งในวันที่ 23 พฤศจิกายน พ.ศ. 2519 ระบบ DES นี้ก็ได้รับการประกาศตัวเป็นมาตรฐานของประเทศ เพื่อใช้เป็นแบบสาธารณะทั่วไปและองค์การมาตรฐานระหว่างประเทศ (ISO) ได้ยอมรับเป็นมาตรฐานสากลเรียบร้อยแล้ว

ระบบมาตรฐาน DES เป็นวิธีการที่ผสมรวมการเข้ารหัสลับพื้นฐานของแบบแทนที่และแบบสับเปลี่ยนมากทำงานร่วมกันถึง 16 วงรอบ โดยข้อมูลปกติเดิมจะผ่านการเข้ารหัสเป็นบล็อกของจำนวน 64 บิต และสามารถกำหนดกุญแจยาวเป็นตัวเลข 56 บิต ที่ผู้ใช้สามารถเปลี่ยนแปลงตามต้องการ

2.1.4 รูปแบบของการใช้ DES

การใช้ DES ให้เข้ากับลักษณะงานกระทำได้ในหลายรูปแบบ การใช้งานในระบบคอมพิวเตอร์โดยรวมจะใช้ป้องกันการลักลอบดักฟังข้อมูล เมื่อถูกส่งออกไปตามสายส่งระยะไกล เช่น สายโทรศัพท์ หรือสายเฉพาะกิจ (Leased Line) เชื่อมต่อระหว่างสองจุดหรือใช้กับข้อมูลที่เก็บอยู่ในแผ่นเก็บข้อมูล อาจเป็นเทปหรือแผ่นดิสก์ (Disk) การป้องกันการลักลอบดักฟังข้อมูลจะติดตั้งระบบที่จุดส่งข้อมูล โดยทำการเข้ารหัสข้อมูลที่จุดส่งออกและที่จุดรับทำการถอดรหัส ทั้งทางด้านส่ง (เข้ารหัส) และด้านรับ (ถอดรหัส) ต้องใช้กุญแจเดียวกัน การเข้ารหัสข้อมูลในรูปของแฟ้มข้อมูล ข้อมูลในแฟ้มข้อมูลจะถูกอ่านและทำขบวนการเข้ารหัสและเขียนเก็บที่แผ่นเก็บข้อมูลสำรอง เมื่อต้องการใช้ข้อมูลที่ถูกเก็บแบบเข้ารหัส ทำได้โดยการอ่านข้อมูลที่เข้ารหัสในแฟ้มข้อมูลและเข้าขบวนการถอดรหัส โดยใช้กุญแจเดียวกับการเข้ารหัสก็จะได้ข้อมูลเดิม

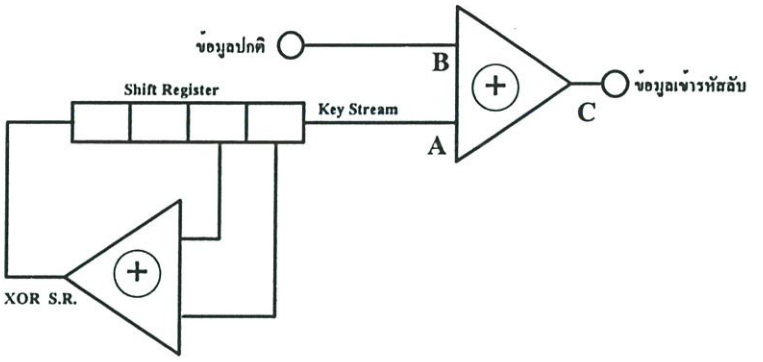
2.1.5 คุณสมบัติของ DES

ขบวนการเข้าและถอดรหัส DES ระบุในมาตรฐานนี้ ทำการเปลี่ยนแปลงในรูปแบบของไบนารี 64 บิต ไปเป็นค่าในรูปแบบของไบนารี 64 บิต ซึ่งการแปลงเข้าหรือถอดรหัสขึ้นอยู่กับตัวแปรกุญแจ 56 บิต การหาค่ากุญแจหรือการพยายามที่จะลักลอบดักฟังข้อมูลไม่มีวิธีอื่นนอกจากการทดลองทุกๆ กุญแจที่เป็นไปได้ กุญแจขนาด 56 บิต เท่ากับ 2^{56} ประมาณเจ็ดหมื่นล้านล้าน กุญแจที่เป็นไปได้ ดังนั้นความพยายามที่จะถอดรหัส โดยวิธีการทดลองทุกๆ เจ็ดหมื่นล้านล้านกุญแจจึงไม่สามารถกระทำได้ในเวลาอันสั้น ยิ่งกว่านั้นถ้ากุญแจเปลี่ยนบ่อยๆ ความเสี่ยงต่อเหตุการณ์นี้จะน้อยลงอย่างมาก

2.1.6 การสร้างรหัสลับด้วยคอมพิวเตอร์แบบง่าย

วิธีการเข้ารหัสที่กล่าวมาแล้วเป็นเพียงทฤษฎี ในทางปฏิบัติเราสามารถเข้ารหัสได้ทั้งทางฮาร์ดแวร์และซอฟต์แวร์ การเข้ารหัสแบบสับเปลี่ยนสามารถใช้การกระทำด้วยการ Exclusive - OR (คำสั่ง XOR) ข้อมูลข่าวสารแต่ละไบต์กับกุญแจที่เลือกไว้แล้ว การเข้ารหัสแบบแทนที่สามารถนำตัวอักษรไปเทียบกับตาราง เพื่อนำตัวอักษรใหม่มาแทนที่

รูปที่ 2.1 เป็นตัวอย่างของระบบการเข้ารหัสลับแบบสายธาร (Character stream) ที่ใช้ Shift register และวงจร Exclusive - OR มาสร้างข้อมูลกุญแจ เพื่อไปกระทำแบบ Exclusive - OR กับข้อมูลข่าวสารได้ข้อมูลเข้ารหัสลับ



รูปที่ 2.1 แสดงระบบการเข้ารหัสแบบสายธาร (Character stream)

ตารางที่ 2.1 แสดงผลของการเข้าและถอดรหัสแบบสายธาร

ลำดับที่	XOR S.R.	KEY Shift Register	จุด A	ข้อมูลเข้า จุด B	เข้ารหัส A ⊕ B = จุด C	ถอดรหัส A ⊕ C = จุด B
1	0	1 0 1 1	1	0	1	0
2	1	0 1 0 1	1	0	1	0
3	1	1 0 1 0	0	1	1	1
4	1	1 1 0 1	1	0	1	0
5	1	1 1 1 0	0	0	0	0
6	0	1 1 1 1	1	1	0	1
7	0	0 1 1 1	1	0	1	0
8	0	0 0 1 1	1	1	0	1

สมมติให้ ข้อมูลที่จุดทำ Shift register $= (1011)_2$

และข้อมูลเข้า $(A4)_{16} = (1010\ 0100)_2$

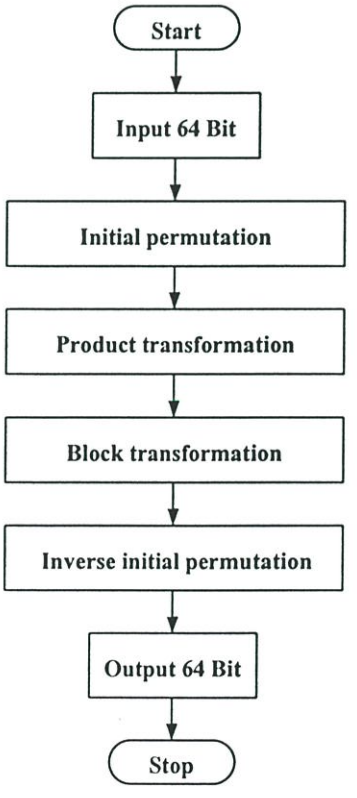
จากตารางที่ 2.1 จุด A คือผลจากการทำ Shift register ดังรูปที่ 2.1 ข้อมูลที่เข้ารหัสแล้ว ณ. จุด C ได้จากการทำ A XOR กับ B จะได้ $(0100\ 1111)_2 = (4F)_{16}$ เมื่อนำข้อมูลที่เข้ารหัสแล้ว ณ. จุด C มาเป็นข้อมูลเข้า โดยมี KEY หรือ Shift register ตัวเดิม (จุด A) จะได้ข้อมูลที่ถอดรหัส

A $\oplus$ C = จุด B ดังในตารางที่ 2.1 เห็นได้ว่า ณ จุด B ซึ่งเป็นข้อมูลเข้าเท่ากับข้อมูลที่ถอดรหัส (A $\oplus$ C) โดยมี A เป็น Key stream ตัวเดียวกัน

2.2 หลักการของการเข้ารหัสลับแบบ DES

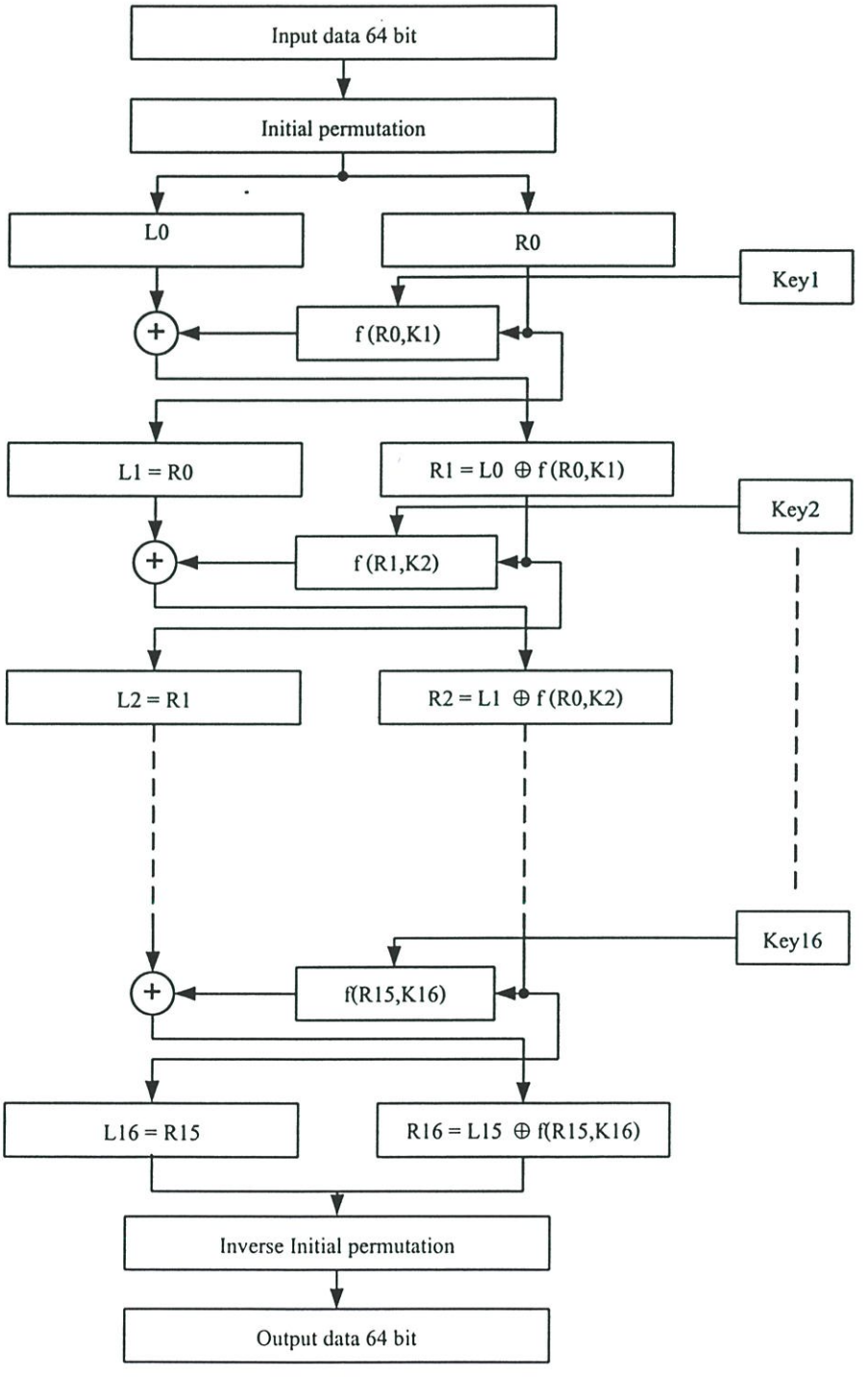
ขั้นตอนวิธีของ DES เป็นวิธีการเชื่อมต่ออย่างประณีตและสลับซับซ้อนของการเข้ารหัสด้วยเทคนิคที่สำคัญ 2 แบบ คือ เทคนิคการแทนที่บิต (Substitution) และเทคนิคการสลับลำดับบิต (Permutation) ตามที่กำหนดในขั้นตอนวิธี กระบวนการของ DES จะได้จากการดัดแปลงการประยุกต์เทคนิคทั้งสองอย่างนี้

จากรูปที่ 2.2 จะแสดงขั้นตอนการทำงานโดยสังเขปของการเข้ารหัสลับแบบ DES ข้อความที่ต้องการเข้ารหัสขนาด 64 บิต จะนำมาสลับบิต (Initial permutation) ขั้นแรกเสียก่อนแล้วจึงทำ Product transformation ซึ่งเป็นส่วนที่ยุ่งยากที่สุด ในส่วน Block transformation เป็นเพียงการสลับที่ด้านซ้ายและด้านขวา 32 บิต สุดท้ายคือ การสลับบิตกลับการสลับบิตขั้นแรก (Inverse initial permutation) ก็จะได้ข้อความที่เข้ารหัสลับออกมาตามต้องการ



รูปที่ 2.2 แสดงขั้นตอนการเข้ารหัสลับแบบ DES [18]

การทำงานที่ละเอียดขึ้นจะแสดงในรูปที่ 2.3 อินพุตในการเข้ารหัสลับมี 2 ชุด คือ ข้อมูลปกติและคีย์ โดยมีขนาด 64 บิตเท่ากัน ข้อมูลปกติที่เป็นอินพุตจะถูกนำไปเข้าบล็อก IP (Initial Permutation) ซึ่งจะทำการจัดเรียงบิตของข้อมูลปกติเสียใหม่และให้เอาท์พุตออกมาที่เรียกว่า Permuted input ในขั้นตอนต่อจากนี้จะแบ่งครึ่ง Permuted input ออกเป็น 2 ส่วนๆ ละ 32 บิต โดยเรียกเป็น L (Left) และ R (Right)



รูปที่ 2.3 แสดงDES algorithm [14]

ขั้นตอนต่อไปจะนำ Permuted input ไปผ่านกระบวนการเข้ารหัสลับ โดยนำส่วนของ L ไปกระทำ Modulo – 2 หรือ XOR (Exclusive – OR) กับคีย์ย่อยที่ถูกบล็อกมาและย้ายตำแหน่งของ L ในขั้นตอนปัจจุบันให้ไปอยู่ในตำแหน่ง R ในขั้นตอนถัดไป ซึ่งเป็นไปตามสมการที่ 2.1

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \qquad \text{..... (2.1)}$$

ส่วนของ R นั้น ก็เพียงแต่ย้ายตำแหน่งในขั้นตอนปัจจุบันให้ไปอยู่ที่ตำแหน่ง L ในขั้นตอนต่อไป ซึ่งเป็นไปตามสมการ $L_i = R_{i-1}$ หลังจากนั้นก็กระทำเช่นนี้ไปเรื่อยๆ จนครบ 16 ครั้ง ในแต่ละครั้งเรียกว่า วงรอบ (Cycle) จนได้เอาท์พุตขนาด 64 บิต ที่ผ่านการเข้ารหัสลับ โดยใช้คีย์ย่อย 16 คีย์ จากนั้นนำเอาท์พุตขนาด 64 บิต ที่ได้มาทำการกลับ L เป็น R และ R เป็น L อีกครั้งหนึ่ง ก็จะได้ค่าเอาท์พุตออกมาที่เรียกว่า Permuted output

สุดท้ายก็จะนำข้อมูล Permuted output ขนาด 64 บิต ไปผ่านเข้า IP^{-1} หรือ FP (Inverse Initial Permutation or Final Permutation) โดยทำการจัดเรียงบิตใหม่คล้ายๆ กับขั้นตอน IP แต่จะกลับกันเท่านั้น เอาท์พุตที่ได้นี้เรียกว่า Ciphertext พร้อมที่จะส่งไปยังปลายทาง

2.2.1 การสลับบิตขั้นแรกและขั้นสุดท้าย

จากขั้นตอนวิธีของ DES ในรูปที่2.3 จะเห็นว่าข้อมูลที่เข้ามาขนาด 64 บิต จะถูกสลับตำแหน่ง

ตารางที่ 2.2 แสดงการสลับบิตขั้นแรก

1	2	3	4	5	6	7	8
58	50	42	34	26	18	10	2
9	10	11	12	13	14	15	16
60	52	44	36	28	20	12	4
17	18	19	20	21	22	23	24
62	54	46	38	30	22	14	6
25	26	27	28	29	30	31	32
64	56	48	40	32	24	16	8
33	34	35	36	37	38	39	40
57	49	41	33	25	17	9	1
41	42	43	44	45	46	47	48
59	51	43	35	27	19	11	3
49	50	51	52	53	54	55	56
61	53	45	37	29	21	13	5
57	58	59	60	61	62	63	64
63	55	47	39	31	23	15	7

บิตแต่ละบิตจนได้เอาท์พุตขนาด 64 บิต เรียกว่า Permuted input การทำงานของการสลับบิตขั้นแรกอธิบายได้ตามตารางที่ 2.2 โดยตารางนี้จะทำงานเหมือนกับตารางอื่นๆ ทั้งหมดที่พบได้ในบทนี้ คือ จะทำงานจากซ้ายไปขวาและจากบนลงล่าง ตัวอย่างเช่น ตาราง Initial permutation จะสลับ Plaintext บิตที่ 58 ไปตำแหน่งที่ 1, บิตที่ 50 ไปตำแหน่งที่ 2 และบิตที่ 42 ไปตำแหน่งที่ 3 เช่นนี้เรื่อยๆ ไปจนครบ 64 บิต

การสลับบิตขั้นสุดท้ายจะมีลักษณะการทำงานตามตารางที่ 2.3 ซึ่งกลับกันกับตารางที่ 2.2 ดังนั้น ถ้านำ Plaintext ผ่านการสลับบิตตามตารางที่ 2.2 ครั้งหนึ่ง แล้วนำผลที่ได้ไปผ่านการสลับบิตตามตารางที่ 2.3 ก็จะได้ Plaintext ของเดิมกลับคืนมา ซึ่งจะสามารถพิสูจน์การทำงานของการสลับบิตได้ โดยดูจากตารางทั้งสอง นั่นคือ จากตารางที่ 2.2 จะเห็นว่าบิตที่ 1 ของ Plaintext จะถูกสลับไปเป็นบิตที่ 40 จากนั้นเมื่อมาผ่านตารางที่ 2.3 บิตที่ 40 จะถูกสลับมาเป็นบิตที่ 1 เมื่อเป็นเช่นนี้ทุกๆ บิตของ Plaintext เมื่อผ่านทั้งสองตารางก็จะได้ข้อความที่เป็น Plaintext ดังเดิม

ตารางที่ 2.3 แสดงการสลับบิตขั้นสุดท้าย[18]

1	2	3	4	5	6	7	8
40	8	48	16	56	24	64	32
9	10	11	12	13	14	15	16
39	7	47	15	55	23	63	31
17	18	19	20	21	22	23	24
38	6	46	14	54	22	62	30
25	26	27	28	29	30	31	32
37	5	45	13	53	21	61	29
33	34	35	36	37	38	39	40
36	4	44	12	52	20	60	28
41	42	43	44	45	46	47	48
35	3	43	11	51	19	59	27
49	50	51	52	53	54	55	56
34	2	42	10	50	18	58	26
57	58	59	60	61	62	63	64
33	1	41	9	49	17	57	25

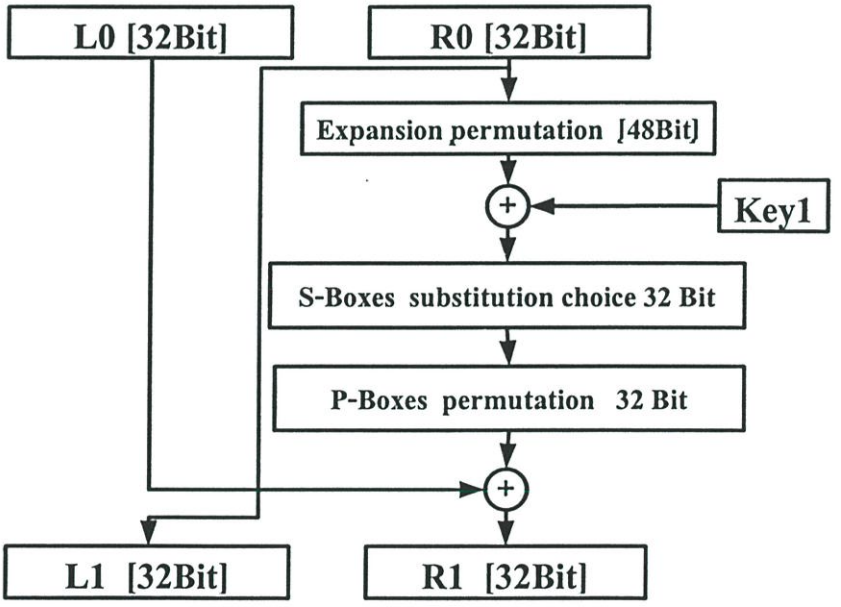
2.2.2 การทำงานของวงรอบ

ขั้นตอนวิธีของแต่ละวงรอบแบ่งการทำงานเป็น 4 กระบวนการ อันดับแรกเป็นข้อมูลด้านขวา R จะถูกขยายจาก 32 บิต เป็น 48 บิต เพื่อนำไป XOR กับคีย์ ต่อมาจะถูกย่อยเป็น 32 บิต

ด้วย S-boxes (Substitution boxes) แล้วทำการสลับบิตด้วย P-box (Permutation box) โดยเอาท์พุตที่ได้จะทำ XOR กับข้อมูลด้านซ้าย L เพื่อจะได้เป็นข้อมูลด้านขวาในวงรอบต่อไป กระบวนการทั้งหมดแสดงในรูปที่ 2.4 แต่ละส่วนของขั้นตอนวิธีในหนึ่งวงรอบของ DES จะมีรายละเอียดและหลักการทำงานดังต่อไปนี้

2.2.2.1 การสลับบิตแบบขยาย

ข้อมูลด้านครึ่งขวาจะถูกขยายจาก 32 บิต เป็น 48 บิต ด้วยการสลับบิตแบบขยาย ซึ่งจะมีบางบิตที่ซ้ำกันเสมอ การขยายบิตมีจุดประสงค์ 2 อย่าง คือ หลังจากการแบ่งครึ่งข้อมูลเป็นสองด้านๆ ละ 32 บิตแล้ว ข้อมูลด้านขวาจะต้องขยายขนาดให้เท่ากับขนาดของคีย์ เพื่อจะได้ทำการ XOR กับคีย์ย่อยขนาด 48 บิตได้ และอีกประการหนึ่ง เพื่อขยายขนาดของข้อมูลให้ยาวขึ้นสำหรับทำการบีบอัดได้ในเวลาต่อมา ด้วยการทำงานของ S-boxes ที่จะต้องใช้ข้อมูลอินพุตขนาด 48 บิต มาทำการบีบอัดให้ได้ข้อมูลเอาท์พุตขนาด 32 บิต



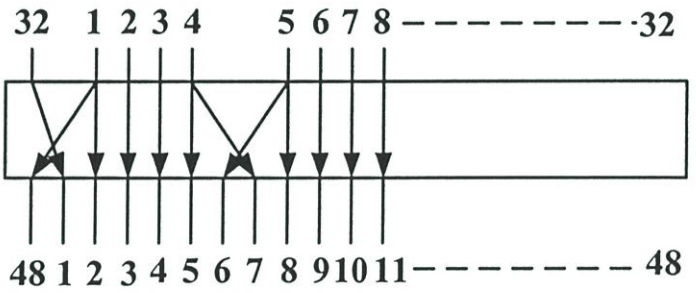
รูปที่ 2.4 แสดงCycle of the DES [3]

การสลับบิตแบบขยาย กำหนดโดยตารางที่ 2.4 จะสังเกตได้ว่าแต่ละกลุ่ม 4 บิตของข้อมูลจะมีการซ้ำบิตที่ 1 และ 4 ขณะที่บิตที่ 2 และ 3 จะถูกใช้เพียงครั้งเดียว ตารางนี้แสดงตำแหน่งของบิตเอาท์พุต ซึ่งได้มาจากอินพุต ด้วยเหตุนี้ Expansion permutation จึงมีการเคลื่อนบิตบางตัวมากกว่า 1 ตำแหน่ง แต่ละแถวของตารางแสดงการเคลื่อนของบิต 8 บิต จะมีบิตที่ซ้ำกัน

ตารางที่ 2.4 แสดงการขยายบิต [17]

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

ของตารางนี้ คือ อินพุตบิตที่ 1 จะเคลื่อนไปที่ตำแหน่งเอาท์พุตบิตที่ 2 และบิตที่ 48 ขณะที่บิตอินพุตที่ 10 จะเคลื่อนไปที่ตำแหน่งเอาท์พุตที่ 15



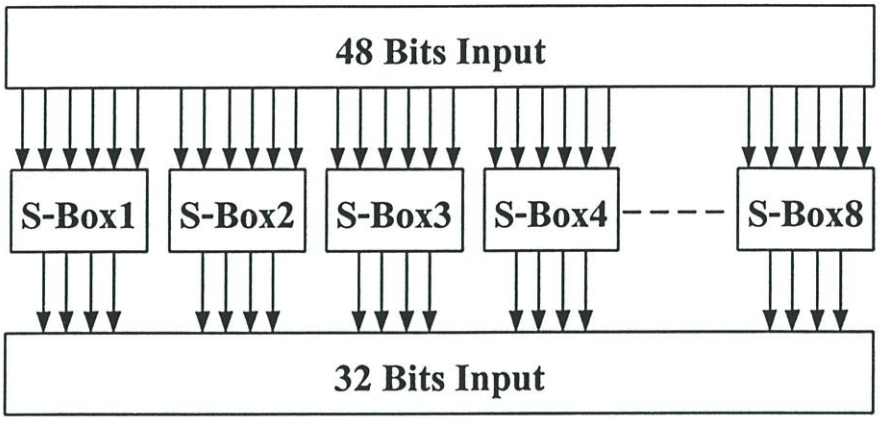
รูปที่ 2.5 แสดงExpansion permutation [14]

2.2.2.2 การแทนบิตด้วย S-boxes

หลังจากข้อมูลด้านซีกขวาถูกขยายบิตจาก 32 บิต เป็น 48 บิต แล้วจะนำมา XOR กับคีย์ย่อยขนาด 48 บิต โดยกระทำแบบบิตต่อบิต ข้อมูลที่ทำการ XOR แล้วจะต้องลดจำนวนบิตลงให้เหลือขนาด 32 บิต เพื่อให้มีขนาดเท่ากับข้อมูลด้านซ้ายซึ่งมีขนาด 32 บิต เพราะขั้นตอนต่อไปจะมีการ XOR กับข้อมูลด้านซ้าย

การแทนจำนวนบิตข้อมูลขนาด 48 บิต ด้วยบิตข้อมูลขนาด 32 บิต ตามขั้นตอนวิธีของ DES จะใช้หลักการทำงานของ S-boxes ซึ่งเป็นการแทนบิตชนิดหนึ่ง โดยข้อมูลเข้าขนาด 48 บิตจะถูกแบ่งออกเป็น 8 ชุดๆ ละ 6 บิต ดังนั้นข้อมูล 48 บิต ก็จะถูกแทนด้วย S-box ย่อย 8 ชุด ซึ่ง S-box ย่อยแต่ละชุดจะมีตารางเป็นตัวกำหนดข้อมูลเอาท์พุต 4 บิต ที่ได้รับการอ้างอิงข้อมูลจาก

อินพุต 6 บิต ดังนั้น S-box ย่อย 8 ชุดๆ ละ 4 บิต ก็จะเป็นข้อมูลเอาท์พุตขนาด 32 บิต ตามต้องการ แผนผังของ S-boxes ทั้งระบบแสดงไว้ในรูปที่ 2.6



รูปที่ 2.6 แสดงS-boxes substitution[14]

การทำงานของ S-box ย่อยแต่ละชุด อธิบายได้ดังนี้ ขั้นแรกจะกำหนดให้แยกข้อมูลอินพุตออกเป็น 8 กลุ่มๆ ละ 6 บิต ซึ่งจะแทนด้วย $B_1\ B_2\ B_3 \ldots B_8$ เรียกว่า B_i เป็นกลุ่มที่ถูกทำงานด้วย S-box ย่อย S_i S-box ย่อยเป็นการแทนบิตบนตารางของข้อมูล 4 แถวและ 16 คอลัมน์ (ดูตารางที่ 2.4 ประกอบ) สมมติว่าบล็อก B_i มีข้อมูลอินพุต 6 บิต เป็น $b_1\ b_2\ b_3\ b_4\ b_5\ b_6$ บิต b_1 และ b_6 จะถูกนำมารวมกันเป็น 2 บิต แบบเลขฐานสอง ซึ่งมีค่าเท่ากับเลขฐานสิบจาก 0 ถึง 3 และจะเป็นตัวกำหนดค่าทางด้าน Row (r) ส่วนบิต $b_2\ b_3\ b_4$ และ b_5 ก็จะถูกนำมารวมกันเป็น 4 บิต แบบเลขฐานสอง ซึ่งมีค่าเท่ากับเลขฐานสิบจาก 0 ถึง 15 และจะเป็นตัวกำหนดค่าทางด้าน Column (c)

ตารางที่ 2.5 แสดง Table of S_boxes1[3]

Column / Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1																
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

ตารางที่ 2.6 แสดง Table of S_boxes2[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_2																
0	15	1	5	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

ตารางที่ 2.7 แสดง Table of S_boxes3[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_3																
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

ตารางที่ 2.8 แสดง Table of S_boxes4[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_4																
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

ตารางที่ 2.9 แสดง Table of S_boxes5[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_5																
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

ตารางที่ 2.10 แสดง Table of S_boxes6[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_6																
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

ตารางที่ 2.11 แสดงTable of S_boxes7[3]

Column / Row	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S_7																
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	1

ตารางที่ 2.12 แสดง Table of S_boxes8[3]

Column / Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_8																
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

การแทนบิตข้อมูลของ S-boxes จะแปลง S-box ย่อยแต่ละชุด B_i 6 บิต เป็น 4 บิต ตามส่วนของ S_i โดยแสดงเป็นแถว r และคอลัมน์ c ของข้อมูลตามตารางที่ 2.5 ถึงรูปที่ 2.12 ตัวอย่างเช่น สมมติให้ B_7 มีข้อมูลในเลขฐานสองเป็นค่า 010011 ดังนั้นจากบิต b_1 และ b_6 จะได้ค่า $r = 01 = 1$ และจากบิต $b_2\, b_3\, b_4$ และ b_5 จะได้ค่า $c = 0011 = 3$ การแปลง B_7 จะพบในแถวที่ 1 คอลัมน์ที่ 9 ของตารางในส่วนของ S_7 ตามตารางที่ 2.4 ค่าที่ได้ออกมาคือ 3 ซึ่งมีค่าเท่ากับ 0011 และจะเป็นข้อมูลเอาท์พุต 4 บิตที่ใช้แทนข้อมูลอินพุตค่า 010011

2.2.2.3 การสลับบิตด้วย P-box

หลังจากผ่านกระบวนการของ S-boxes เอาท์พุตของ S-boxes ที่มีขนาด 32 บิต จะถูกสลับบิตตามตารางที่ 2.13 เรียกว่า ตาราง P (Permutation) ซึ่งจะแสดงตำแหน่งของบิตอินพุตที่เคลื่อนไปเป็นบิตเอาท์พุต โดยไม่มีบิตใดที่ใช้สองครั้งหรือตัดบิตใดออกไปเลย กระบวนการสลับบิตแบบนี้เรียกว่า Straight permutation ตัวอย่างเช่น อินพุตบิตที่ 1 เคลื่อนไปเป็นเอาท์พุตบิตที่ 9 ขณะที่อินพุตบิตที่ 10 เคลื่อนไปตำแหน่งเอาท์พุตบิตที่ 16 เอาท์พุตที่ได้ของ P-box ทั้งหมด 32 บิต

ตารางที่ 2.13 แสดง Table of P-box [14]

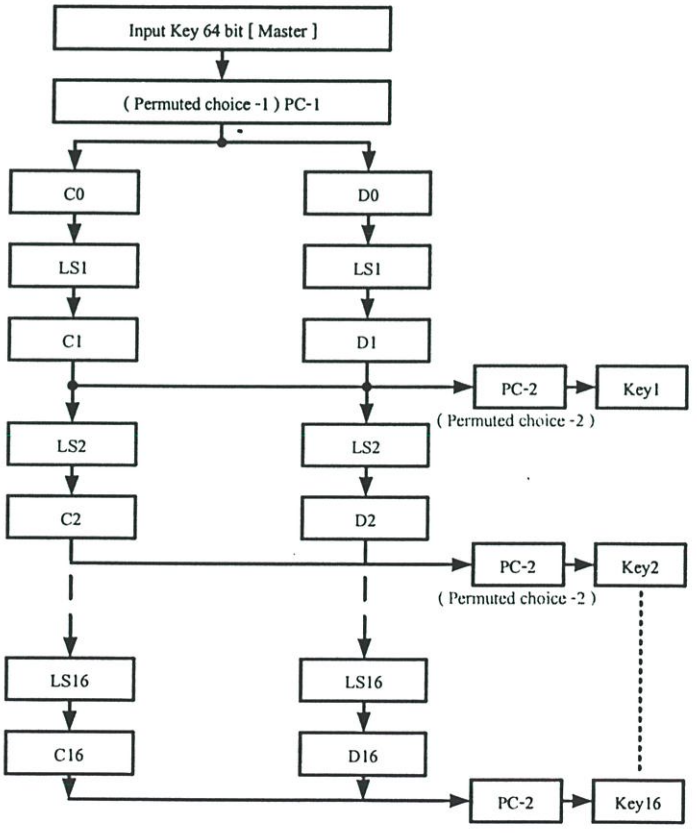
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	12	11	4	25

จะถูกนำมาทำ XOR กับข้อมูลด้านซ้ายแบบบิตต่อบิต เอาท์พุตสุดท้ายที่ได้ก็จะเป็นข้อมูลด้านขวาในวงรอบต่อไป ส่วนข้อมูลด้านซ้ายของวงรอบต่อไปก็จะนำมาจากข้อมูลด้านขวาของวงรอบที่ผ่านมา ตามขั้นตอนวิธีที่แสดงในรูปที่ 2.3 และ 2.4 เมื่อครบ 16 วงรอบก็จะได้ข้อความที่เข้ารหัสแล้วเป็น Ciphertext

2.2.3 การแปลงคีย์ย่อย

ระยะแรก DES ใช้คีย์ยาว 64 บิต แต่เมื่อประกาศให้เป็นมาตรฐาน ปรากฏว่า ถูกรัฐบาลสหรัฐตัดความยาวของคีย์ลงเหลือเพียง 56 บิต ด้วยเหตุผลว่า คีย์ยาวแค่นี้ก็ต้องใช้คอมพิวเตอร์เมนเฟรมถอดรหัสกันหลายพันปีกว่าจะถอดออก ในขณะที่นักคอมพิวเตอร์บางคนวิจารณ์ว่ารัฐบาลสหรัฐเองอาจมีขีดความสามารถที่จะถอดรหัส DES ที่มีคีย์ยาว 56 บิตได้ จึงลดความยาวมาตรฐานของคีย์ลงมาเป็น 56 บิต เพื่อที่รัฐบาลจะสามารถดักฟังความลับของภาคเอกชนได้ [17]

ขั้นตอนวิธีในการแปลงคีย์ย่อยของ DES แสดงไว้ในรูปที่ 2.7 และสามารถอธิบายการทำงานได้เป็นขั้นตอนดังนี้



รูปที่ 2.7 แสดงขั้นตอนการสร้างคีย์ย่อย[16]

อินพุตคีย์ที่ป้อนเข้ามาขนาด 64 บิต จะถูกลดบิตลงเหลือ 56 บิต เมื่อผ่านการลดบิตและสลับบิตตามตาราง PC - 1 (Permutation Choice –1) เพื่อทำการจัดเรียงตำแหน่งของบิตใหม่ โดยตัดบิตทุกๆ ตำแหน่งที่ 8 (Parity bit) ออก ต่อมาก็จะทำการแบ่งคีย์ขนาด 56 บิตออกเป็นด้านซ้ายและขวา ด้านละ 28 บิต กำหนดให้เป็น C_0 และ D_0 ตามลำดับ จากนั้นก็เป็นการเลื่อนบิตไปทางซ้ายแบบวนรอบ (Circular left shift) จะได้เอาท์พุตเป็นค่า C_1 และ D_1 ออกมา แล้วจึงนำมาต่อกันเป็น 56 บิต หลังจากนั้นนำไปผ่านตาราง PC-2 เพื่อทำการลดบิตให้เหลือ 48 บิต และสลับบิตใหม่ด้วยก็จะได้เอาท์พุตออกมาเป็นคีย์ย่อยที่ 1 (K_1) ซึ่งครบการทำงาน 1 วงรอบตามขั้นตอนวิธีการสร้างคีย์ย่อย ส่วนวงรอบต่อไปก็จะนำ C_1 และ D_1 ในวงรอบที่ 1 มาเลื่อนบิต แล้วผ่านตาราง PC-2 ก็จะได้คีย์ย่อยที่ 2 (K_2) ออกมา จะทำซ้ำเช่นนี้จนครบ 16 วงรอบ ก็จะได้คีย์ย่อยจำนวน 16 คีย์ สำหรับใช้ในการทำ XOR เพื่อเข้ารหัสกับด้านข้อมูลตามขั้นตอนวิธีของ DES

2.2.3.1 การสลับบิตขั้นแรกของคีย์

ขั้นตอนแรกของการสร้างคีย์ย่อย จะต้องสลับบิตและลดบิตของคีย์ขนาด 64 บิต ลงเหลือ 56 บิต Parity bit คือ บิตที่ 8, 16, 24, 32, 40, 48, 56 และ 64 จะถูกตัดออก กระบวนการทั้งหมดจะเป็นไปตามตารางที่ 2.14 ซึ่งเรียกว่าตาราง PC-1 (Permutation Choice-1)

ตารางที่ 2.14 แสดงตาราง PC-1[14]

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

ตัวอย่างเช่น บิตที่ 57 จะถูกเคลื่อนไปที่ตำแหน่งบิตที่ 1 และบิตที่ 49 ก็จะเคลื่อนไปที่ตำแหน่งบิตที่ 2 เรื่อยมาจนถึงบิตที่ 4 จะเคลื่อนไปที่ตำแหน่งบิตที่ 33

2.2.3.2 การเลื่อนบิตทางซ้าย

หลังจากผ่านขั้นตอนการสลับบิตตามตาราง PC-1 แล้ว คีย์ขนาด 56 บิต จะถูกแบ่งเป็นสองด้านๆ ละ 28 บิต ในแต่ละด้านจะต้องทำการเลื่อนบิตไปทางซ้ายจำนวน 1 หรือ 2 บิตในแต่ละวงรอบ ทั้งนี้ขึ้นอยู่กับว่าจะกระทำในวงรอบที่เท่าไร โดยตารางที่ 2.15 จะกำหนดจำนวนบิตดังกล่าว

สังเกตจากตารางที่ 2.15 จะเห็นว่าวงรอบที่ 1, 2, 9 และ 16 มีการเลื่อนบิตไปทางซ้ายวงรอบละ 1 บิต นอกนั้นในวงรอบอื่นๆ จะมีการเลื่อนบิตไปทางซ้ายวงรอบละ 2 บิตทั้งหมด

2.2.3.3 การสลับบิตตามตาราง PC-2

ในแต่ละวงรอบของการสร้างคีย์ย่อย เมื่อได้ค่า C_i และ D_i มาแล้ว ก็จะนำค่าทั้งสองมารวมกันเป็น 56 บิต จากนั้นต้องสลับบิตแบบลดบิตให้เหลือ 48 บิต ก็จะได้คีย์ย่อยสำหรับทำ XOR กับข้อมูลด้านขวา ซึ่งขยายจาก 32 บิตเป็น 48 บิต การสลับบิตในขั้นตอนนี้เป็นไปตามตารางที่ 2.16

ตารางที่ 2.15 แสดงตารางการเลื่อนบิต[3]

Cycle No.	Bits shift
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

ตารางที่ 2.16 แสดงตาราง PC-2 [17]

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

จากตารางที่ 2.16 จะเห็นว่าบิตที่ 1 จะเคลื่อนไปเป็นบิตที่ 5 บิตที่ 2 จะเคลื่อนไปเป็นบิตที่ 24 และบิตที่ถูกตัดออกไปจำนวน 8 บิตคือ บิตที่ 9, 18, 22,25, 35, 38, 43 และ 54

2.3 การถอดรหัสของ DES

ขั้นตอนวิธีของ DES จะเหมือนกันทั้งการเข้ารหัสและการถอดรหัส เพราะว่าวงรอบในการทำงาน i จะได้มาจากวงรอบ (i -1) ตามความหมายดังนี้

$$L_i = R_{i-1} \tag{2.2}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \tag{2.3}$$

ซึ่งสัญลักษณ์ $\oplus$ คือ การทำ XOR และ f เป็นฟังก์ชันการทำงานในแต่ละวงรอบ ซึ่งประกอบด้วย การขยายบิต การเลื่อนบิตของคีย์ การแทนบิต และการสลับบิต สองสมการนี้แสดงว่า ผลลัพธ์ของแต่ละวงรอบ (i) จะขึ้นอยู่กับวงรอบก่อนหน้านั้น (i -1) เท่านั้น ถ้าเขียนสมการใหม่ในเทอมของ R_{i-1} และ L_{i-1} จะได้

$$R_{i-1} = L_i \tag{2.4}$$

และ

$$L_{i-1} = R_i \oplus f(R_{i-1}, K_i) \tag{2.5}$$

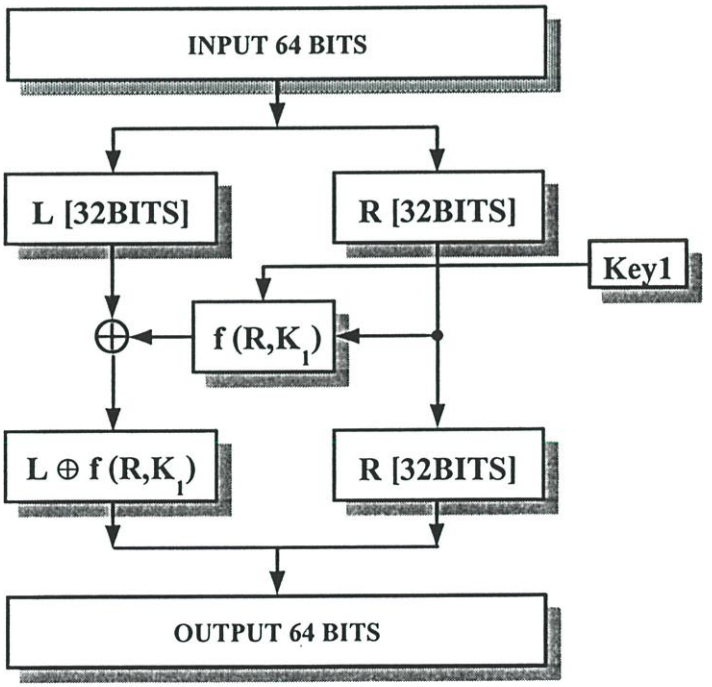
แทนสมการ 2.4 ลงในสมการ 2.5 จะได้

$$L_{i-1} = R_i \oplus f(L_i, K_i) \qquad (2.6)$$

สมการที่ 2.4 และ 2.6 แสดงค่าที่เหมือนกันนี้ จะได้รับผลจากวงรอบก่อนหน้านั้น เป็นคุณสมบัติซึ่งทำให้พฤติกรรมของ DES กลับกัน ดังนั้นด้วย DES เราสามารถจะเข้าและถอดรหัสลับโดยใช้ฟังก์ชัน f ที่เหมือนกัน โดยเปลี่ยนแปลงค่าคีย์ที่ใช้ และค่าคีย์ย่อยต้องกลับลำดับกันเท่านั้น $(K_{16}, K_{15}, \ldots, K_1)$ สำหรับการถอดรหัสลับ ซึ่งขั้นตอนวิธีหนึ่งชุดสามารถใช้ได้ทั้งการเข้ารหัสและถอดรหัส เป็นสิ่งสะดวกมากสำหรับงานทางด้านฮาร์ดแวร์และซอฟต์แวร์ของ DES เพื่อให้เข้าใจเกี่ยวกับการถอดรหัสลับของ DES ได้ดีขึ้น จะอธิบายการทำงานโดยละเอียดดังนี้

2.3.1 การถอดรหัสโดยทำฟังก์ชัน f หนึ่งครั้ง

เพื่อให้เข้าใจได้ง่ายขึ้น จะตัดส่วนของการสลับบิตขั้นแรกและขั้นสุดท้ายออก และพิจารณาทำฟังก์ชัน f เพียงครั้งเดียว (ตามขั้นตอนวิธีของ DES จะต้องทำถึง 16 ครั้ง) ข้อมูลที่เข้าสู่



รูปที่ 2.8 แสดงการเข้ารหัสโดยทำฟังก์ชัน f หนึ่งครั้ง [13]

กระบวนการนี้ คือ ข้อมูล 32 บิต ด้านซ้ายและคีย์ 48 บิต ในรูปที่ 2.8 แสดงขั้นตอนวิธีการทำฟังก์ชัน f หนึ่งครั้ง ซึ่งจะเหมือนกันทั้งการเข้าและการถอดรหัสลับ เพราะใช้คีย์เพียงคีย์เดียว

จากรูปข้างบนนี้ ในการเข้ารหัส ถ้ากำหนดให้

INPUT = LR = 3A4A E604 2711 BE7D (64 บิต)

KEY = K_1 = 3710 2520 0405 340E (48 บิต)

จะได้

OUTPUT = $[L \oplus f(R, K_1)]R$ = 5764 61CC 2711 BE7D (48 บิต)

และการถอดรหัส โดยการนำข้อมูลจากการเข้ารหัสมาเป็นข้อมูลอินพุตของการถอดรหัส จะได้

INPUT = $[L \oplus f(R, K_1)]R$ = 5764 61CC 2711 BE7D (64 บิต)

KEY = K_1 = 3710 2520 0405 340E (64 บิต)

จะได้

OUTPUT = $\{[L \oplus f(R, K_1)] \oplus f(R, K_1)\}R$ = LR

= 3A4A E604 2711 BE7D (64 บิต)

ข้อมูลเข้าของการเข้ารหัสจะเท่ากับข้อมูลออกของการถอดรหัส แนวคิดในการทำมี 3 ส่วน คือ XOR, ฟังก์ชัน f และเอาท์พุต แบ่งเป็น 2 ส่วน คือ $[L \oplus f(R, K_1)]$ กับ [32 บิต]

ในหลักการของ DES จะใช้เทคนิคของ XOR เข้าช่วยในการเข้ารหัสและถอดรหัส ด้วยคุณสมบัติของ XOR ที่ว่า เมื่อทำการ XOR กับข้อมูลใดๆ เช่น L กับ X แล้วนำผลลัพธ์ที่ได้ คือ $L \oplus X$ มาทำการ XOR กับค่าเดิม X จะได้ข้อมูลเดิม ดังสมการที่ 2.7

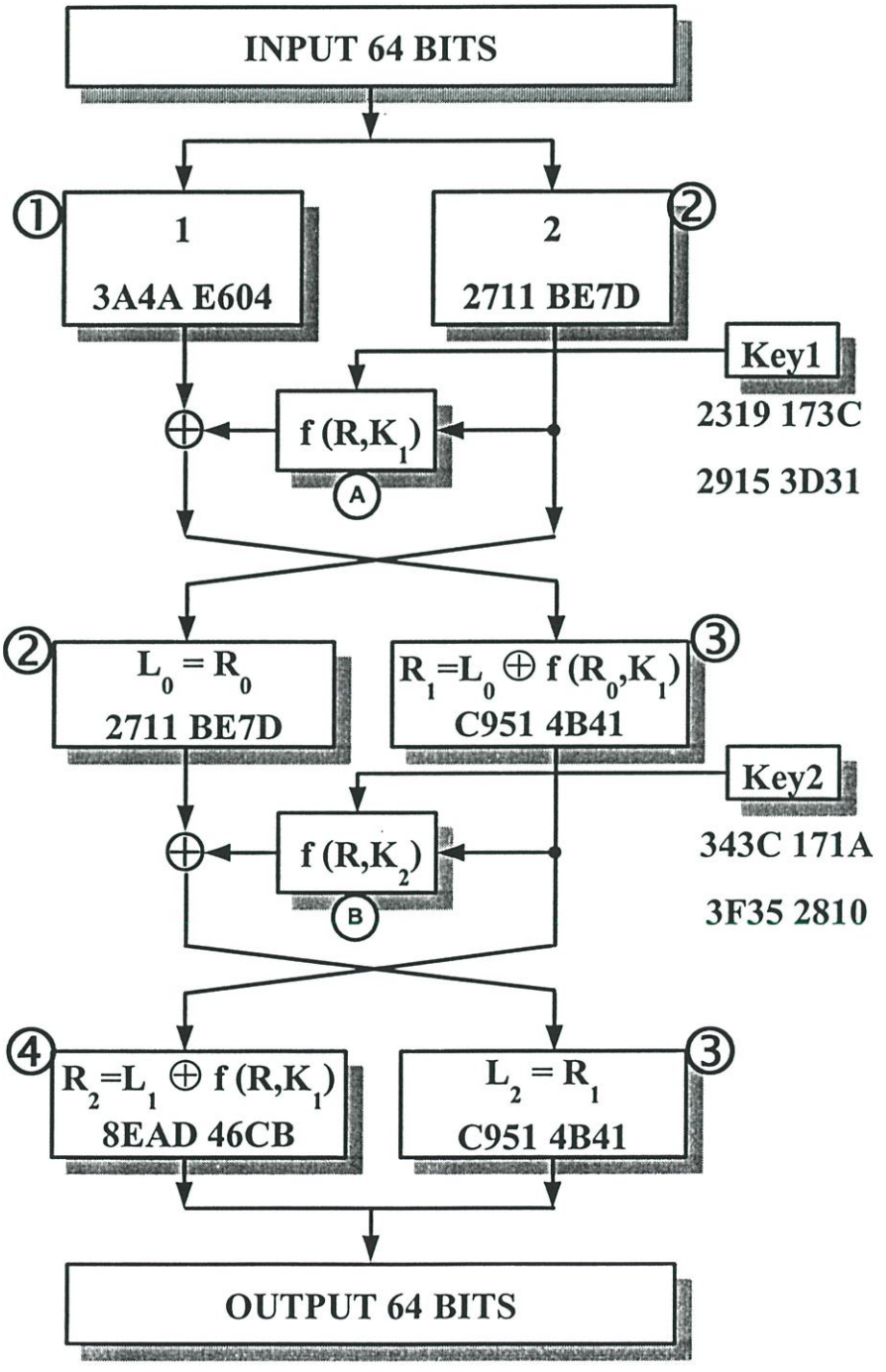
$$[L \oplus X] \oplus X = L \tag{2.7}$$

สำหรับฟังก์ชัน f จะนำอินพุตซึ่งเป็น R 32 บิต และคีย์ K_1 มาทำขบวรการสลับบิต โดยใช้ S-boxes ผลลัพธ์ที่ได้มาทำการ XOR กับ L เขียนได้เป็น $L \oplus f(R, K_1)$ ในขณะนี้ให้คิดว่าฟังก์ชัน f เป็นฟังก์ชันอื่น โดยไม่ต้องคำนึงถึงความสลับซับซ้อนภายใน และกำหนดให้

$$f(R, K_1) = X \tag{2.8}$$

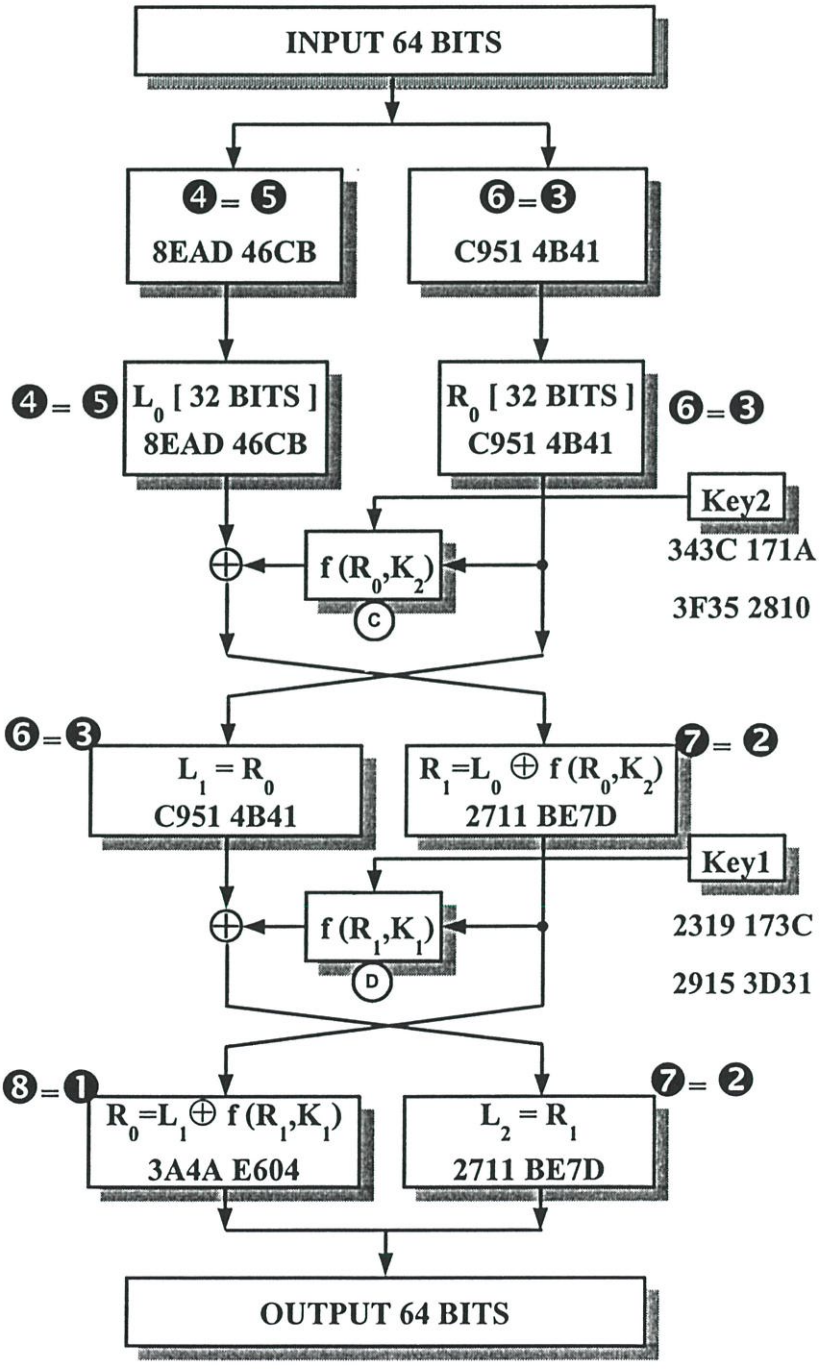
จากรูปที่ 2.9 จะเห็นว่าผลลัพธ์จาการเข้ารหัสแบ่งเป็น 2 ส่วน คือ

รูปที่ 2.9 แสดงเอาท์พุตของการทำฟังก์ชัน f หนึ่งครั้ง [13]



รูปที่ 2.10 แสดงการเข้ารหัสโดยทำฟังก์ชัน f สองครั้ง [13]

ข้อมูล R ไม่ถูกเปลี่ยนแปลงและใช้เป็นอินพุตของฟังก์ชัน f เพื่อทำการถอดรหัส ขอให้สังเกตว่า การเข้าและถอดรหัสฟังก์ชัน f ยังคงใช้อินพุต R และ K_1 เหมือนกัน ดังนั้นค่า $f(R, K_1)$ ในการเข้าและถอดรหัสจะเท่ากัน จะเห็นว่า ข้อมูลเข้าของการเข้ารหัสและข้อมูลออกของการถอดรหัสเหมือนกัน เมื่อใช้คีย์ชุดเดียวและทำงานรอบเดียว ก็จะยังคงรักษาคุณลักษณะของการเข้าและถอดรหัสได้ แต่ไม่ซับซ้อนมาก



รูปที่ 2.11 แสดงการถอดรหัสโดยใช้ฟังก์ชัน f สองครั้ง [13]

2.3.2 การเข้าและถอดรหัสโดยทำฟังก์ชัน f สองครั้ง

การใช้ฟังก์ชัน f สองครั้ง ใช้หลักการเดียวกับการเข้าและถอดรหัส โดยทำฟังก์ชัน f หนึ่งครั้งที่กล่าวมาแล้ว แต่มีข้อแตกต่างคือ หลังจากทำฟังก์ชัน f หนึ่งครั้ง ข้อมูล L และ R จะถูกสลับที่กัน โดยให้ $L_1 = R_0$ และ $R_1 = L_0 \oplus f(R_0 \oplus K_1)$ หลังจากนั้นก็เข้าสู่ขบวนการทำฟังก์ชัน f ครั้งที่สอง ดังรูปที่ 2.10

เมื่อนำข้อมูลที่ผ่านการเข้ารหัสที่จุด 4 และจุด 3 โดยทำฟังก์ชัน f สองครั้งแล้ว มาเข้ากระบวนการถอดรหัสตามรูปที่ 2.11 โดยในข้อมูลที่จุด 5 = จุด 4 และ จุด 6 = จุด 3 กระบวนการถอดรหัสจะทำย้อนกลับกับกระบวนการในการเข้ารหัส ให้สังเกตที่การเข้ารหัส คีย์จะเริ่มจาก K_1 และ K_2 ส่วนการถอดรหัส คีย์จะเริ่มจาก K_2 และ K_1

พิจารณารูปที่ 2.10 และ 2.11 ในรอบแรกอินพุตที่จุด C จะเท่ากับอินพุตที่จุด B คือ 3 = 6 และ K_2 ได้ผลลัพธ์ฟังก์ชันที่เท่ากัน เมื่อผลลัพธ์ของ $5 \oplus C$ ได้ 7 ข้อมูลที่จุด 7 จะเท่ากับ 2 ซึ่งเป็นการทำกระบวนการย้อนกลับในรอบแรกของการถอดรหัส โดยใช้หลักของการทำ XOR 2 ครั้ง จะได้ค่าเดิมดังเสนอไปแล้วนั้น

ในรอบที่สองอินพุตที่จุด D จะเท่ากับจุด A คือ 2 = 7 และ K_1 ซึ่งผลลัพธ์ของฟังก์ชันที่เท่ากัน เนื่องจาก 7 = 2 และใช้ K_1 ชุดเดียวกัน ผลลัพธ์ของฟังก์ชันที่จุด $D \oplus 6$ ได้ผลลัพธ์เท่ากับ 8 ซึ่ง 8 = 1 โดยใช้หลักของการทำ XOR กับข้อมูลเดิมสองครั้ง

การแสดงการเข้ารหัสและถอดรหัสลับโดยทำฟังก์ชัน f สองครั้ง ที่ได้นำเสนอแล้วนั้น เป็นการแสดงให้เห็นวิธีการเข้ารหัสและถอดรหัส เพื่อให้เข้าใจได้ง่ายและไม่ยุ่งยากซับซ้อนมากนัก

2.3.3 การเข้าและถอดรหัสโดยทำฟังก์ชัน f หลายครั้ง

ในระบบของ DES จะใช้หลักการเข้าและถอดรหัสถึง 16 รอบ โดยใช้คีย์ 16 ชุด การเข้ารหัสจะใช้คีย์ตั้งแต่ K_1 ถึง K_{16} แต่การถอดรหัสจะใช้คีย์ K_{16} ถึง K_1 ซึ่งเป็นกระบวนการย้อนกลับการเข้ารหัส ความสลับซับซ้อนจะมากขึ้น แต่ยังใช้หลักการเดียวกับการเข้าและถอดรหัสโดยทำฟังก์ชัน f สองครั้ง

2.4 ความปลอดภัยของ DES

เมื่อเริ่มประกาศใช้ครั้งแรก มีข้อโต้เถียงเกี่ยวกับความปลอดภัยของ DES แต่ถึงแม้ว่าจะมีการโต้เถียงกันมาก ก็ยังไม่ปรากฏข้อวิจารณ์จากผู้ออกแบบ และการวิเคราะห์เกี่ยวกับคุณสมบัติของ DES จากภายนอกเลย

มีคำถามและข้อสงสัยมากมายเกี่ยวกับความปลอดภัยของ DES เช่น การกำหนดความยาวของคีย์ (Key length) มากๆ, จำนวนการทำซ้ำของวงรอบ (Number of iterations) และการออกแบบ S-boxes โดยเฉพาะความลึกลับของ S-boxes ทุกสิ่งทุกอย่างเหล่านี้ปราศจากเหตุผลว่าทำไมหรือมันเป็นสำหรับอะไร ประชาชนบางคนกลัวว่า NBS จะฝังกับดักไว้ในขั้นตอนวิธี เพื่อที่ NBS จะถอดรหัสได้ง่าย [14] การวิเคราะห์ความปลอดภัยของ DES จึงต้องพิจารณาสิ่งต่อไปนี้

2.4.1 จำนวนรอบการทำซ้ำของวงรอบ

มีการวิเคราะห์กันมากว่าจำนวนการทำซ้ำ 16 วงรอบนั้นเพียงพอหรือไม่ เนื่องจากการทำซ้ำในแต่ละวงรอบจะกระจายข้อมูล Plaintext ไปเป็น Ciphertext ซึ่งไม่แน่ชัดว่า 16 วงรอบ จะกระจายข้อมูลอย่างเพียงพอ ตัวอย่างเช่น ด้วยวงรอบเพียง 1 วงรอบ Plaintext 1 บิต ก็จะมีผลต่อ Chiphertext เพียง 1 บิตเท่านั้น ถ้าวงรอบมากกว่า การกระจายก็มากกว่า ดังนั้นในทางเป็นจริง Ciphertext ทุกๆ บิตจะไม่ขึ้นอยู่กับ Plaintext ทุกบิต และหลังจาก 8 รอบ Ciphertext จะเป็นฟังก์ชันแรมดอมของ Plaintext และคีย์ทุกบิต ดังนั้นจึงมีคำถามว่า ทำไมจึงไม่หยุดทำงานหลังจาก 8 วงรอบ [3]

หลายปีที่ผ่านมา การเปลี่ยนแปลงของ DES ด้วยการลดจำนวนรอบลงถูกเจาะเข้าได้เป็นผลสำเร็จ ในปี ค.ศ. 1982 มีการเจาะเข้าระบบ DES ที่มีจำนวนวงรอบ 3 หรือ 4 วงรอบ และจำนวนวงรอบ 6 วงรอบ ก็ถูกเจาะเข้าได้เช่นเดียวกันในปีต่อมา การวิเคราะห์รหัสโดย Differential ของ Biham และ Shamir ได้อธิบายสิ่งเหล่านี้ว่า DES ทุกๆ แบบที่จำนวนวงรอบน้อยกว่า 16 วงรอบ สามารถที่จะถูกทำให้เสียหายได้ ด้วยการเจาะเข้าแบบ Known – planintext attack มากกว่าแบบ Brute – force attack [14] NBS และ IBM ทำการทดสอบ DES และสรุปผลว่า การทำซ้ำ 8 วงรอบก็เพียงพอที่จะกำจัดข้อสังเกตที่เกี่ยวข้อง ดังนั้นจึงแน่ใจได้ว่าการทำซ้ำ 16 วงรอบของ DES พอเพียงอย่างแน่นอน[3]

2.4.2 ความยาวของคีย์

ความยาวของคีย์เป็นจุดที่ถูกหยิบยกขึ้นมามากที่สุด โดยในระยะแรก IBM เสนอคีย์ของ Lucifer ขนาด 128 บิต ขณะที่คีย์ของ DES ยาวเพียง 56 บิตเท่านั้น การเจาะเข้าแบบ Brute-force กระทำการถอดรหัสได้โดยใช้ลำดับอนุกรมของคีย์ ซึ่งใช้คีย์อันใหม่ทดลองไปเรื่อยๆ จนกระทั่งถอดรหัสได้ ดังนั้นจะมีคีย์ที่เป็นไปได้ 2^{56} คีย์ ถ้าใช้เวลาในการทดสอบครั้งละ 100 mS เวลาที่ใช้ทั้งหมดประมาณ 7.2×10^{15} วินาที หรือประมาณ 228 ล้านปี ถ้าใช้เวลาทดสอบ 1 μS เท่านั้น เวลารวมจะประมาณ 2,280 ปี แม้เวลาทดสอบจะเป็น 1nS ซึ่งเป็นไปไม่ได้สำหรับเทคโนโลยีเวลาในการค้นหาก็ยังมากกว่า 2 ปี โดยที่ยังไม่คำนึงถึงการเสียของฮาร์ดแวร์หรือซอฟต์แวร์ คือสามารถทำงานได้ตลอดเวลา [3]

Diffie และ Hellman แนะนำให้ออกแบบโปรเซสเซอร์ขนานกันหลายๆ ตัว ถ้าชิป 1 ตัว ทำงานที่อัตรา 1 คีย์ต่อ 1 μS จะสามารถเช็คคีย์ได้ประมาณ 8.6×10^{10} คีย์ ใช้เวลา 1 วัน ดังนั้น ถ้าคีย์ทั้งหมด $2^{56} \approx 7 \times 10^{16}$ คีย์ จะใช้เวลา 10 วัน อย่างไรก็ตาม ถ้าใช้ชิป 10^6 ตัว ทำงานแบบขนานที่อัตรานี้จะเช็คคีย์ทั้งหมดได้ใน 1 วัน [3]

Hellman ได้แนะนำแนวทางที่จะลดการคำนวณ และการพักข้อมูลให้เหลือ 2^{37} หรือประมาณ 6.4×10^{11} รวมทั้งกำหนดให้อุปกรณ์ DES จำนวนมากทำงานแบบขนาน ซึ่งเป็นไปได้ที่ทำการคำนวณก่อนและเก็บผลลัพธ์ข้อมูลทั้งหมด ทำให้ราคาของฮาร์ดแวร์ลดลงและความเร็วเพิ่มขึ้น แม้ว่าจะมีความเชื่อที่สำคัญเกิดขึ้นว่า การวิเคราะห์เกี่ยวกับคีย์มีความยาวเพียงพอแล้วนั้น แต่บางคนยังคงไม่แน่ใจเกี่ยวกับคีย์ขนาด 56 บิต มีวิธีที่จะเพิ่มความยาวของคีย์ให้มีประสิทธิภาพโดยวิธีที่ไม่เปลี่ยนแปลงขั้นตอนวิธีของตัวเอง ซึ่งจะสะดวกในกรณีที่ขั้นตอนวิธีนั้นถูกสร้างขึ้นด้วยฮาร์ดแวร์ หรือกรณีที่ไม่สามารถแก้ไขในส่วนของซอฟต์แวร์ได้ [3]

2.4.3 Weak keys

เนื่องจากการสร้างคีย์ย่อยแต่ละวงรอบของขั้นตอนวิธีจะสร้างจากคีย์ที่เริ่มต้น บางครั้งคีย์ที่เริ่มต้นเป็นคีย์ที่เรียกว่า Weak keys ซึ่งถ้าดูจากขั้นตอนวิธีจะเห็นว่า คีย์เริ่มต้นจะถูกแบ่งครึ่ง 2 ด้าน และแต่ละครึ่งจะถูกเลื่อนบิตอย่างอิสระต่อกัน ถ้าบิตทั้งหมดในแต่ละครึ่งเป็น 0 หรือ 1 คีย์ที่ใช้สำหรับแต่ละวงรอบของขั้นตอนวิธีจะเหมือนกันหมดทุกวงรอบ ซึ่งจะเกิดขึ้นได้ถ้าคีย์ที่เข้ามาเป็น 1S, 0S หรือเป็น 1S ครึ่งหนึ่ง ส่วนอีกครึ่งหนึ่งเป็น 0S คีย์ลักษณะดังกล่าวแสดงไว้ในรูปของ Hexadecimal ตามตารางที่ 2.17

ตารางที่ 2.17 แสดงDES Weak keys [3]

Left	Right	Weak Key Value			
zeros	zeros	0101	0101	0101	0101
ones	ones	FEFE	FEFE	FEFE	FEFE
zeros	ones	1F1F	1F1F	0E0E	0E0E
ones	zeros	E0E0	E0E0	F1F1	F1F1

มีคีย์อีกประเภทหนึ่งคือ ที่เป็นคีย์คู่กันสองคีย์ ซึ่งสามารถถอดรหัส Plaintext ไปเป็น Ciphertext ที่เหมือนกัน หมายถึง คีย์ชุดหนึ่งจะเป็นคีย์สำหรับการเข้ารหัส ส่วนการถอดรหัสสามารถใช้คีย์อีกชุดหนึ่งถอดรหัสได้ สิ่งเหล่านี้เกิดจากวิธีการของ DES ซึ่งสร้างคีย์ย่อยเอง (แทน

ที่จะสร้างคีย์ย่อยที่มีความแตกต่างกัน 16 คีย์) คีย์เหล่านี้จะสร้างคีย์ย่อยที่แตกต่างกัน 2 คีย์ แต่ละคีย์ย่อยจะถูกใช้ 8 ครั้งในขั้นตอนวิธี เรียกคีย์นี้ว่า Semi-weak keys ซึ่งได้แสดงไว้ในรูปแบบ Hexadecimal ตามตารางที่ 2.18

ตารางที่ 2.18 แสดงDES Semi – weak key pairs [3]

01FE	01FE	01FE	01FE	FE01	FE01	FE01	FE01
1FE0	1FE0	0EF1	0EF1	E01F	E01F	F10E	F10E
01E0	01E0	01F1	01F1	E001	E001	F101	F101
1FFE	1FFE	0EFE	0EFE	FE1F	FE1F	FE0E	FE0E
011F	011F	010E	010E	1F01	1F01	0E01	0E01
E0FE	E0FE	F1FE	F1FE	FEE0	FEE0	FEF1	FEF1

2.5 สรุป

ขั้นตอนวิธีที่เสนอในบทนี้ได้เสนอเป็นรูปแบบทฤษฎี เปรียบเสมือนวิธีการวิจัยเกี่ยวกับ DES ส่วนหนึ่ง เนื้อหาจะเริ่มจากหลักการเบื้องต้น บล็อกการทำงานพื้นฐาน ตารางการสลับบิตต่างๆ ครบทุกตาราง พร้อมทั้งอธิบายการทำงานทุกขั้นตอน การรับข้อมูลและการสร้างคีย์ย่อย ตอนท้ายจะเสนอรูปแบบของการถอดรหัส (Decrypting of DES) ที่แตกต่างจากการเข้ารหัสลับ โดยอธิบายวิธีการและขั้นตอนการถอดรหัส พร้อมด้วยสิ่งที่ควรรู้เกี่ยวกับการเข้ารหัสลับแบบ DES เช่น ความปลอดภัยของ DES จำนวนวงรอบในการทำซ้ำของขั้นตอนวิธี ความยาวของคีย์ รูปแบบของ Weak keys

บทที่ 3
การสร้างและออกแบบ

3.1 บทนำ

จุดมุ่งหมายของงานวิจัยนี้ ต้องการที่จะนำเสนอวิธีการปรับปรุง Data Encryption Standard (DES) ให้สามารถใช้เวลาในการเข้ารหัสและถอดรหัสน้อยลง และเพิ่มความปลอดภัยให้กับข้อมูลมากขึ้นซึ่งในบทนี้จะกล่าวถึงการปรับปรุงอัลกอริธึม DES เพื่อให้กระบวนการเข้ารหัสและถอดรหัสสามารถทำได้รวดเร็วขึ้น หน้าที่และหลักการของฟังก์ชั่นที่พิเศษเข้าไปเพื่อให้อัลกอริธึม DES แบบใหม่สามารถที่ป้องกันกันลักลอบถอดรหัสข้อมูลจากผู้ที่ไม่ประสงค์ดี โดยจะออกแบบให้อัลกอริธึม DES แบบใหม่นี้จะทำการหน่วงเวลาเมื่อฟังก์ชั่นพิเศษตรวจพบว่ามีการลักลอบถอดรหัส ในส่วนสุดท้ายจะนำเสนอการออกแบบโปรแกรมสำหรับใช้จำลองการทำงานและทดสอบการทำงานของการเข้ารหัส แบบ DES ทั้งแบบเก่าและแบบใหม่

3.2 การปรับปรุงอัลกอริธึม DES

3.2.1 การปรับปรุงในเรื่องความเร็วในการเข้ารหัสและถอดรหัส

สิ่งสำคัญที่ใช้ในการกำหนดความเร็วของการถอดรหัสก็คือ จำนวนรอบการเข้ารหัส ,ขนาดของ key และความเร็วในการทดสอบของแต่ละ key จากเอกสารอ้างของ C.P. Pfleeger [3] ได้เขียนว่าจำนวนรอบในการเข้ารหัสของ DES อย่างน้อยเพียง 8 รอบก็เพียงพอการกระจายข้อมูลสำหรับเพื่อการปกปิดข้อมูลได้ ให้ยากต่อการสังเกตว่าเป็นข้อความใดๆแล้ว ในงานวิจัยจึงเลือกใช้จำนวนรอบของ DES เพียง 8 รอบ เพื่อต้องการที่จะให้ DES สามารถทำขั้นตอนของกระบวนการเข้ารหัสได้ไวขึ้นกว่าเดิม

3.2.2 การปรับปรุงในเรื่องของความปลอดภัยของข้อมูลที่เข้ารหัสไปแล้ว

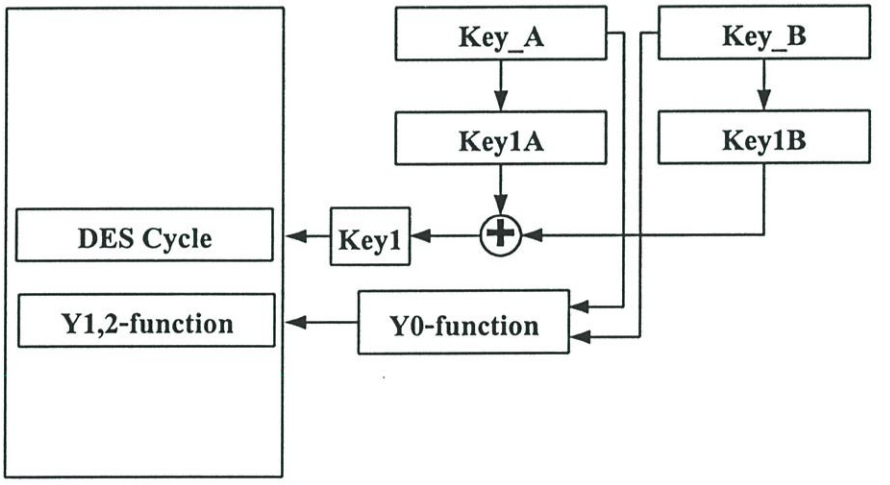
สิ่งสำคัญที่ใช้ในการกำหนดความปลอดภัยของข้อมูลที่ถูกเข้ารหัสคือ ขนาดของ key ถ้าหากจำนวน key มีจำนวนของบิตสูงก็จะทำให้การเข้ารหัสแบบ DES มีความปลอดภัยสูงขึ้น เพราะจำนวน key ยิ่งมากจะทำให้ผู้ที่ลักลอบถอดรหัสต้องใช้เวลานานในการค้นหา key ที่ถูกต้อง ดังนั้นผู้ลักลอบถอดรหัสจะมีวิธีการเดียวที่จะสามารถถอดรหัส key ได้ก็คือการทดลองใช้ค่าของ key ทุกๆค่าสำหรับการถอดรหัส นั้นถ้าเกิดการลักลอบถอดรหัสจะต้องใช้เวลาสำหรับค้นหา Key โดยประมาณ $\underline{2^{64} X \text{ ค่าเวลาที่ในการเข้ารหัสของ DES}}$

กรณี1.ถ้าสมมุติ ให้เวลาที่ใช้ในการเข้ารหัสและถอดรหัสเท่ากับ 20 mSec และใช้ key ขนาด 64 key ซึ่งจะต้องใช้เวลาถอดรหัสเท่ากับ (2^{64} X (20 m Sec)) = 368934881474191032 Sec หรือเท่ากับ **11.698 X 10^3 ปี**

แต่ถ้าหากมีการพัฒนา hardware ให้สามารถเข้ารหัสและถอดรหัสโดยใช้เวลาที่น้อยลงก็จะทำให้เวลาที่ใช้ในการถอดรหัสลดลงดังเช่น

กรณี2.ถ้าสมมุติ ให้เวลาที่ใช้ในการเข้ารหัสและถอดรหัสเท่ากับ 2 mSec และใช้ key ขนาด 64 key ซึ่งจะต้องใช้เวลาถอดรหัสเท่ากับ (2^{64} X (2mSec)) = 36893488147419103 Sec หรือเท่ากับ **1.169 X 10^3 ปี**

กรณี3.ถ้าสมมุติ ให้เวลาที่ใช้ในการเข้ารหัสและถอดรหัสเท่ากับ 0.2 mSec และใช้ key ขนาด 64 key ซึ่งจะต้องใช้เวลาถอดรหัสเท่ากับ (2^{64} X (0.2mSec)) = 3689348814741910.3232 Sec หรือเท่ากับ **0.1169 X 10^3 ปี**



รูปที่ 3.1 แสดงบล็อกไดอะแกรม DES Algorithm ที่ได้พัฒนาขึ้นมาใหม่ จาก DES แบบเก่า

จากทั้ง 3 กรณี ถ้าหากเวลาที่ใช้ในการเข้ารหัสและถอดรหัสน้อยลง ก็จะทำให้ DES มีความปลอดภัยน้อยลง จำเป็นที่จะต้องเพิ่มจำนวน key ให้มากขึ้น ซึ่งในอนาคตอาจจะต้องมีการเพิ่มจำนวนขึ้นเรื่อยๆ ถ้าหากมีการพัฒนา Hardware ให้สามารถเข้ารหัสและถอดรหัสโดยใช้เวลาที่น้อยลงได้ ดังนั้นในงานวิจัยจึงได้นำเสนอให้เพิ่ม Function พิเศษเข้าไปใน DES เพื่อทำหน้าที่คอย

ตรวจสอบการป้อน key ที่ผิดปกติ DES แบบใหม่นี้จะเพิ่ม y-Function เข้าไปเพื่อช่วยในการตรวจสอบการลักลอบถอดรหัสของ key ในรูปที่ 3.1 จะเป็นบล็อกไดอะแกรม DES Algorithm ที่ได้พัฒนาขึ้นมาใหม่

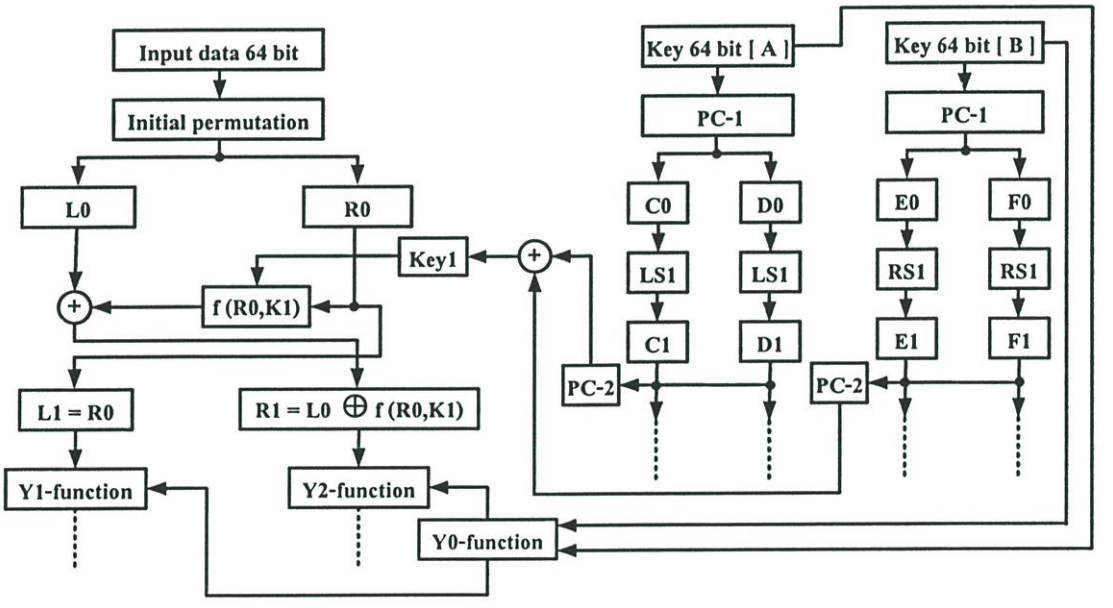
อัลกอริทึมใหม่ที่นำเสนอสำหรับการปรับปรุงการเข้ารหัสของ DES คือ การเพิ่มฟังก์ชั่นพิเศษเข้าไปในขั้นตอนของ DES จุดประสงค์ก็เพื่อป้องกันผู้ที่พยายามลักลอบถอดรหัส บล็อกไดอะแกรมแสดง อัลกอริทึมแบบใหม่ที่นำเสนอแสดงในรูปที่3.1 และรูปที่3.2 (Algorithm ที่พัฒนาขึ้นมาใหม่) โดยจะเพิ่ม Y1-function และ Y2-function เข้าไป ซึ่ง Y1,Y2-function จะมีคุณสมบัติดังนี้

1.หน่วงเวลาข้อมูล

2.ส่งผ่านข้อมูล(บัฟเฟอร์)

3.เลื่อนข้อมูล

คุณสมบัติทั้ง 3 จะเกิดได้จากการสั่งงานของ Y0-function เท่านั้น



รูปที่ 3.2 แสดง DES Algorithm ที่ได้พัฒนาขึ้นมาใหม่

Y0-function ทำหน้าที่ตรวจสอบกุญแจรหัสขนาด 128 บิตที่เกิดจากการนำ $KeyA_i$ และ $KeyB_i$ มารวมบิตกัน เมื่อ Y0-function มีการตรวจพบว่าเกิดการลักลอบถอดรหัสกุญแจ Y1 และ Y2-function จะถูกสั่งงานให้มีคุณสมบัติเป็นตัวหน่วงเวลาและเลื่อนข้อมูลเพื่อทำให้ข้อมูลเกิดข้อ

ผิดพลาดมากว่าเดิม (ข้อมูลเกิดความเสียหาย จนไม่สามารถล่วงรู้ความหมายของคำได้) ในสภาวะปกตินั้น Y1, Y2-function จะถูกสั่งงานให้เป็นบัฟเฟอร์เพื่อส่งผ่านข้อมูลไปให้กับวงรอบถัดไปของ DES

ในการตรวจสอบข้อผิดพลาดของ Y0-function นั้นจะตรวจสอบจาก

- ค่าจาก $KeyA_i$ และ $KeyB_i$ ที่มีการป้อน key แบบเรียงลำดับที่มีความต่อเนื่องกันหลายๆค่า
- ค่าข้อความที่ผ่านการเข้ารหัส (cipher text) หรือ ข้อความปกติ (plain text) ที่ป้อนเข้ามาซ้ำๆกัน เกิน 3 ครั้ง

ในอัลกอริทึมแบบใหม่จะเพิ่มจำนวนของรหัสกุญแจให้มากขึ้นเป็น 128 บิต โดยจะเพิ่มในส่วนของการสร้างกุญแจรหัสอีก 1 ชุด ดังแสดงในรูปที่3.5และ รูปที่3.6 พร้อมทั้งได้ปรับปรุงตารางการเลื่อนบิตของรหัสกุญแจเสียใหม่ (ตารางที่3.1)ให้แตกต่างไปจากเดิม เมื่อนำกุญแจรหัสจากหลักการสร้างกุญแจแบบเดิมและแบบใหม่มารวมกันจะได้จำนวนบิตของรหัสกุญแจเท่ากับ 128 บิต ซึ่งวิธีการรวม $KeyA_i$ และ $KeyB_i$ เข้าด้วยกันโดนใช้วิธีการนำkey 2 ชุดมา XOR กัน สำหรับรหัสกุญแจ (K_i) แบบใหม่ที่สร้างได้ในแต่ละรอบก็สามารถเขียนได้ดังสมการที่ (3.1) - (3.7)

$$Ki = KeyA_i \oplus KeyB_i \qquad (3.1)$$

$$KeyA_i = PC\_2\,(C_i\, D_i) \qquad (3.2)$$

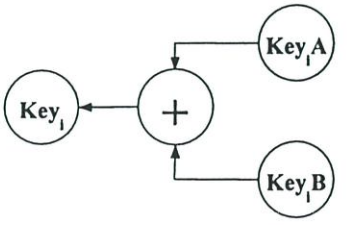
$$KeyB_i = PC\_2\,(E_i\, F_i) \qquad (3.3)$$

$$C_i = LS_i\,(C_{i-1}) \qquad (3.4)$$

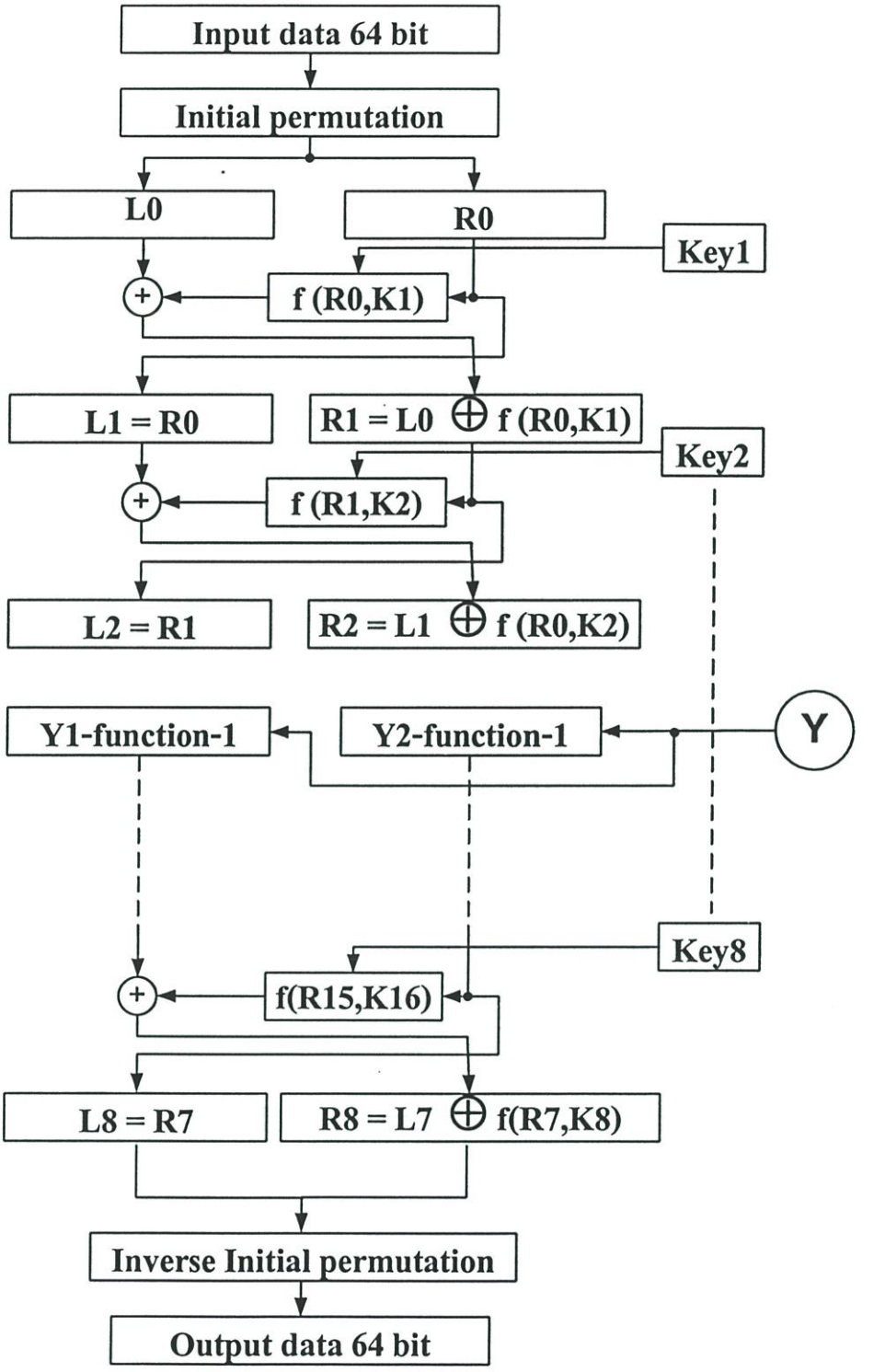
$$D_i = LS_i\,(D_{i-1}) \qquad (3.5)$$

$$E_i = RS_i\,(E_{i-1}) \qquad (3.6)$$

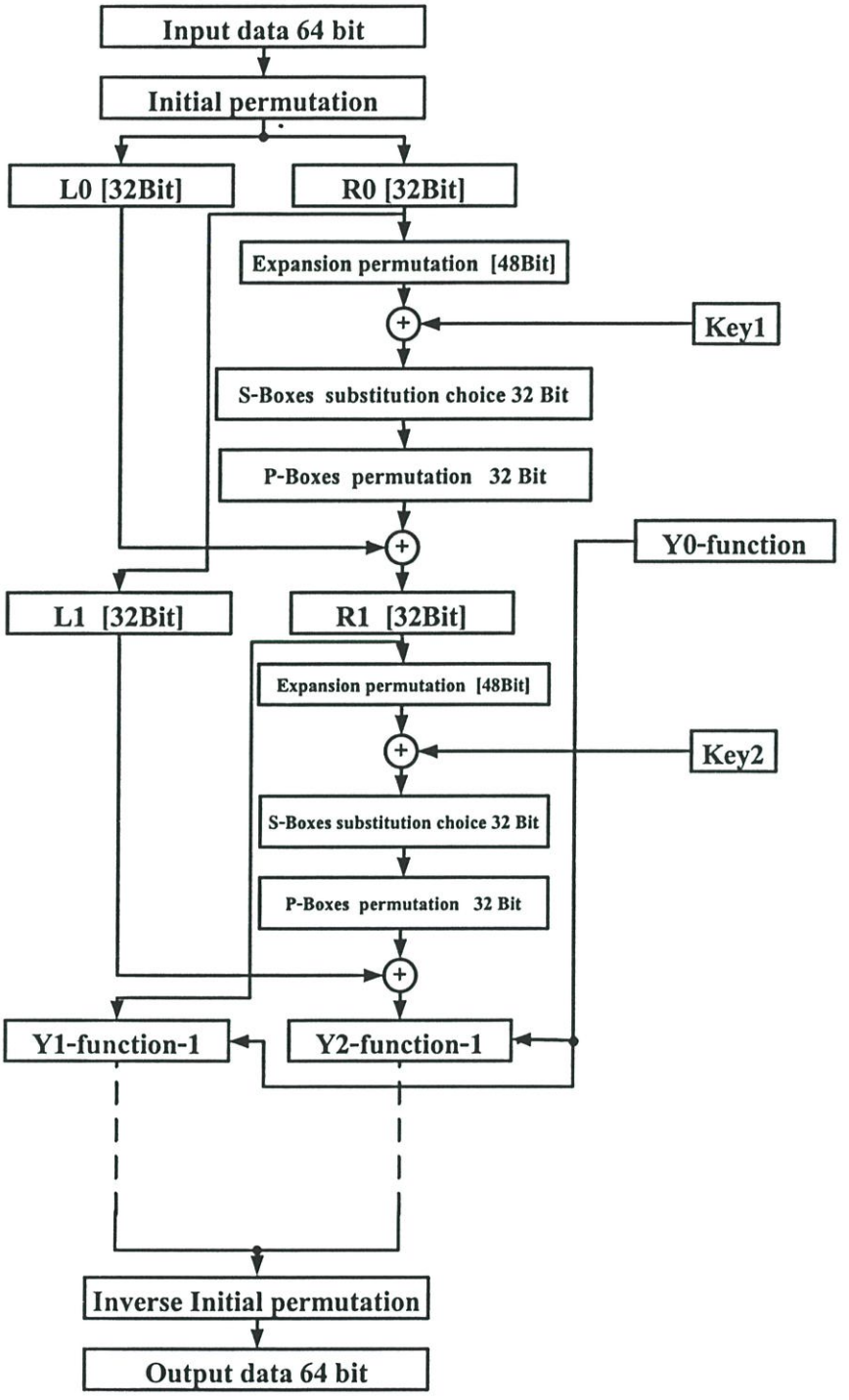
$$F_i = RS_i\,(F_{i-1}) \qquad (3.7)$$



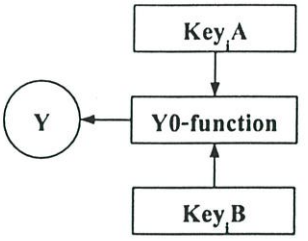
รูปที่ 3.2ก แสดงการนำค่าของ key_iA และ key_iB มา ทำการ XOR เพื่อสร้างเป็น key ใหม่



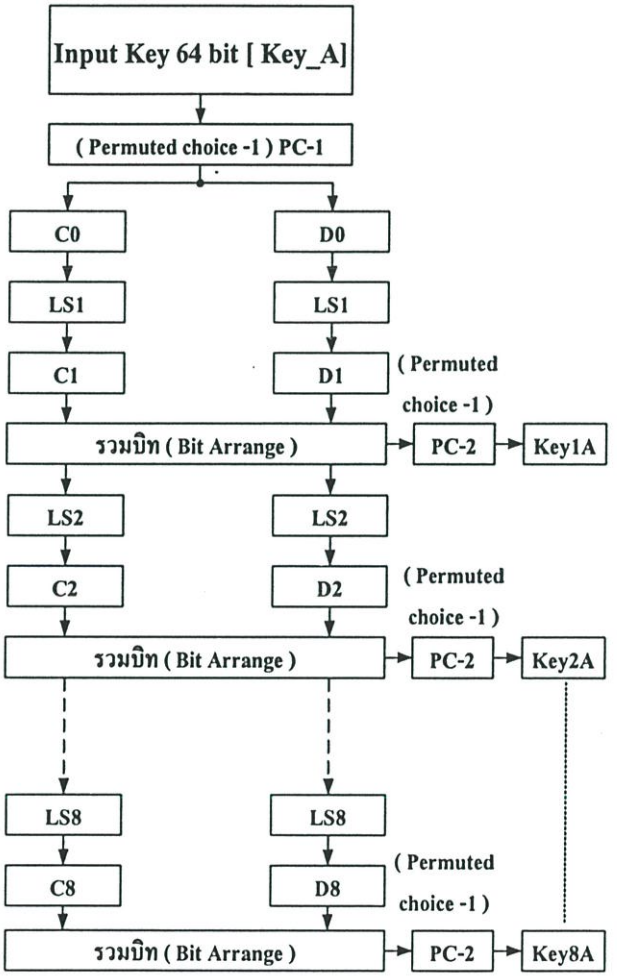
รูปที่ 3.3 แสดง DES Algorithum ที่เพิ่ม Function พิเศษเข้าไป



รูปที่ 3.4 แสดง Cycle of DES ที่เพิ่ม function พิเศษเข้าไป



รูปที่ 3.2ข แสดงการนำค่า key_iA และ key_iB ป้อนให้กับ Y0 Function เพื่อตรวจสอบ key ที่ผิดปกติ



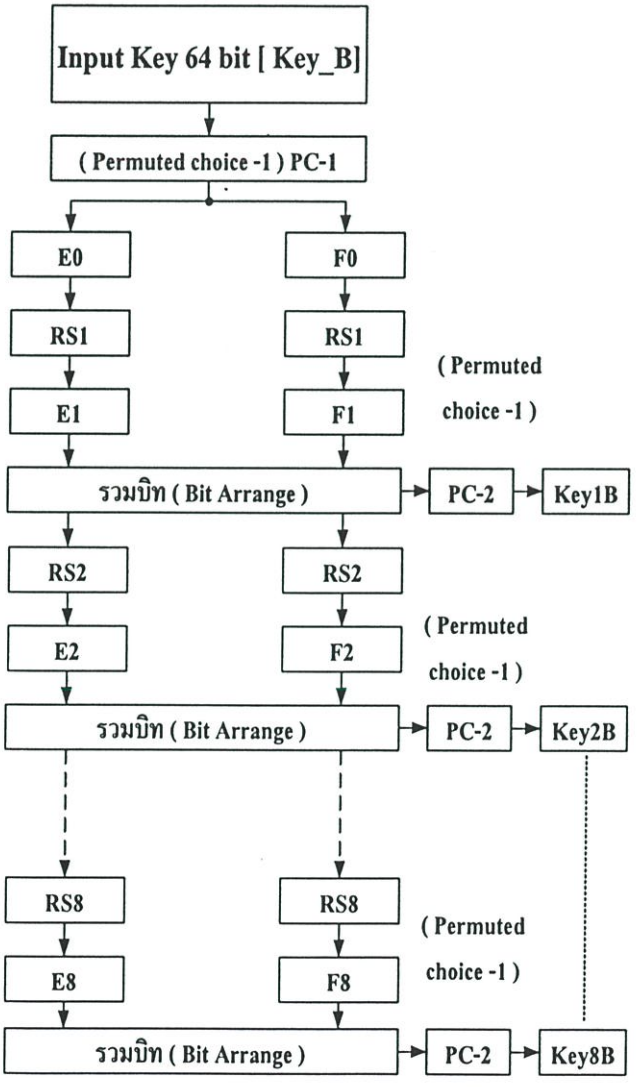
รูปที่ 3.5 แสดงขั้นการสร้างคีย์รหัสย่อย ชุด A

รูปที่3.2ข แสดงถึงนำค่าของ key_iA และ key_iB ส่งไปยัง Y0 Functionเพื่อให้ Y0 Function ตรวจสอบผลที่ได้จากการตรวจสอบ ผลที่ได้จากการตรวจสอบจะนำไปสั่งให้ Y0 Function และ Y0 Function ทำงานต่อไป รูปที่3.3 แสดงถึง DES Algorithum ที่เพิ่ม Function พิเศษเข้าไปเพื่อให้ Algorithum สามารถตรวจสอบ การป้อน key ที่ผิดปกติได้ รูปที่3.4 แสดงถึง Cycle of DES ที่เพิ่ม

function พิเศษเข้าไป รูปที่ 3.4 จะแสดงรายละเอียดของรอบของการเข้ารหัสในจำนวน 1 รอบ โดยเพิ่ม function พิเศษเข้าไปในรอบแรกเพียงรอบเดียวเท่านั้น

3.2.3 การปรับปรุงการสร้าง key รหัส

การปรับปรุงการสร้าง key สำหรับ DES แบบใหม่นั้นจะใช้ key ทั้งหมด 128 key เพื่อใช้สำหรับถอดรหัสข้อความโดยที่ key ขนาด 128 key ใช้เพื่อเพิ่มความปลอดภัยให้กับ DES แบบใหม่ ดังนั้นถ้าเกิดการลักลอบถอดรหัสจะต้องใช้เวลาสำหรับค้นหา Key โดยใช้เวลาประมาณ 2^{128} *X ค่าเวลาที่ในการเข้ารหัสของ DES*



รูปที่ 3.6 แสดงขั้นการสร้างคีย์รหัสย่อย ชุด B

ตารางที่ 3.1 แสดงการเลื่อนบิตของรหัสกุญแจแบบใหม่

	Key_A		Key_B	
Cycle NO.	C0	D0	E0	F0
1	1	1	1	1
2	1	1	1	1
3	2	2	2	2
4	2	2	2	2
5	2	2	2	2
6	2	2	2	2
7	2	2	2	2
8	2	2	2	2
9	1	1	1	1
10	2	2	2	2
11	2	2	2	2
12	2	2	2	2
13	2	2	2	2
14	2	2	2	2
15	2	2	2	2
16	1	1	1	1
	LS	LS	RS	RS

ถ้าสมมุติให้เวลาที่ใช้ในการเข้ารหัสและถอดรหัสเท่ากับ 20 m Sec และใช้ key ขนาด 128 key ซึ่งจะต้องใช้เวลาถอดรหัสเท่ากับ (2^{128} X (20 m Sec)) = 6.8056473384 X 10^{36} m Sec หรือเท่ากับ 5165184683074354333080974611 8 ปี หรือเท่ากับ 5165 X 10^{25} ปี

key ขนาด 128 บิต ที่ใช้สำหรับ DES แบบใหม่นั้นจะปรับปรุงให้วิธีการกำเนิด key DES แบบเก่าจำนวน 2 ชุด แต่ใช้จำนวนรอบในการสร้าง key ย่อยเพียง 8 รอบ แสดงดังในรูปที่ 3.5 และรูปที่ 3.6โดยในรูปที่ 3.5 จะเป็นการสร้าง key ย่อยชุด A โดยจะใช้วิธีการเลื่อนค่า key ย่อยไปทางซ้ายมือ และส่วนของการสร้าง key ย่อยชุด B โดยจะใช้วิธีการเลื่อนค่า key ย่อยไปทางขวามือแทน และนำ key ย่อย ชุด A และ ชุด B ที่ได้มาในแต่ละรอบของการกำเนิด key มาทำกระบวนการ XOR เพื่อทำให้ key ย่อยที่สำหรับการเข้ารหัสที่ได้เกิดการกระจายมากขึ้น และจำนวนครั้งในการเลื่อนข้อมูลของ key ในแต่ละรอบของ key ย่อย ชุด A และ ชุด B นั้นจะอ้างอิงจากตารางที่ 3.1

ตารางที่ 3.1 แสดงจำนวนครั้งในการเลื่อนบิตของ key ย่อยชุด A และชุด B โดยที่ LS ในที่นี้จะมีความหมายถึงการเลื่อนบิตของ key ไปทางซ้าย ในทำนองเดียวกันโดย RS ในที่นี้จะมีความหมายถึงการเลื่อนบิตของ keyไปทางขวา

3.3 การออกแบบโปรแกรมเพื่อทดสอบอัลกอริธึม

ในส่วนนี้จะนำเสนอถึงการออกแบบโปรแกรมจำลองการทำงานของอัลกอริธึมใหม่และเก่า อัลกอริธึมทั้ง 2 แบบ สามารถเขียน System Flow Diagram ได้ดังรูปที่ 3.7 ทั้ง2 อัลกอริธึม ของ DES แบบเก่าและแบบใหม่จะใช้โปรแกรม visual basic เขียนขึ้นเพื่อจำลองการทำงาน โดยการผลการจำลองการทำงานที่ได้จะอยู่ในบทที่4

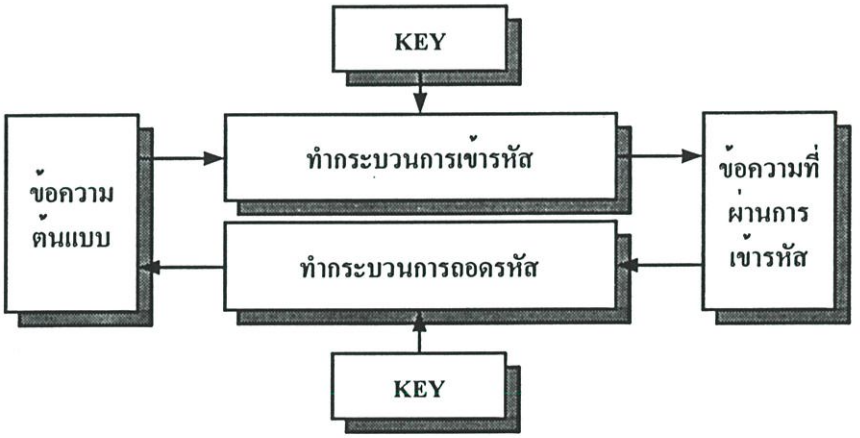
จากรูปที่ 3.7 แสดงถึง System Flow Diagram สำหรับการเข้ารหัสแบบ DES สามารถอธิบายการทำงานได้ดังนี้

1. เริ่มต้นที่ผู้ใช้กำหนดข้อความ (plain text) ที่จะทำการเข้ารหัสและ ป้อนค่า key ที่จะใช้ในการเข้ารหัส

2. ทำการเข้ารหัสจาก key ที่กำหนดให้ ผลลัพธ์ที่ได้จะถูกแสดงให้เห็นเป็น cipher text

3. เมื่อจะทำการถอดรหัสก็จะนำข้อความที่ได้เข้ารหัส (cipher text)ไว้แล้ว และรับค่า key ที่ใช้สำหรับถอดรหัสข้อความนั้น ไปผ่านขั้นตอนการถอดรหัส

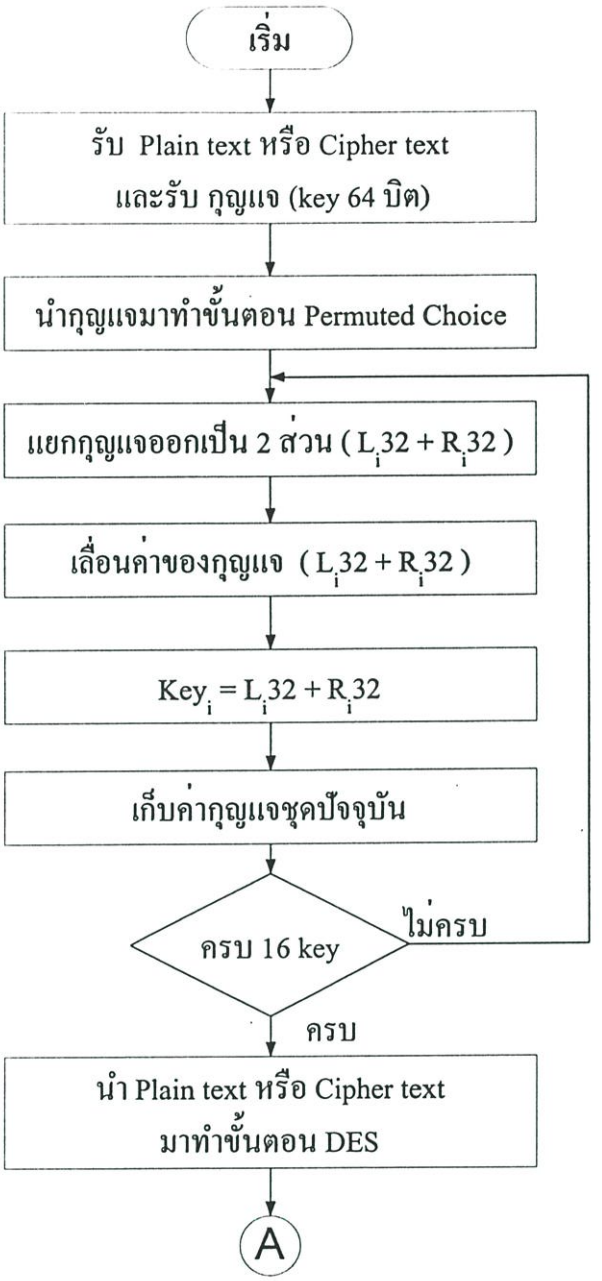
4. แสดงผลลัพธ์ที่ได้ซึ่งจะเป็นข้อความปกติ(plain text)



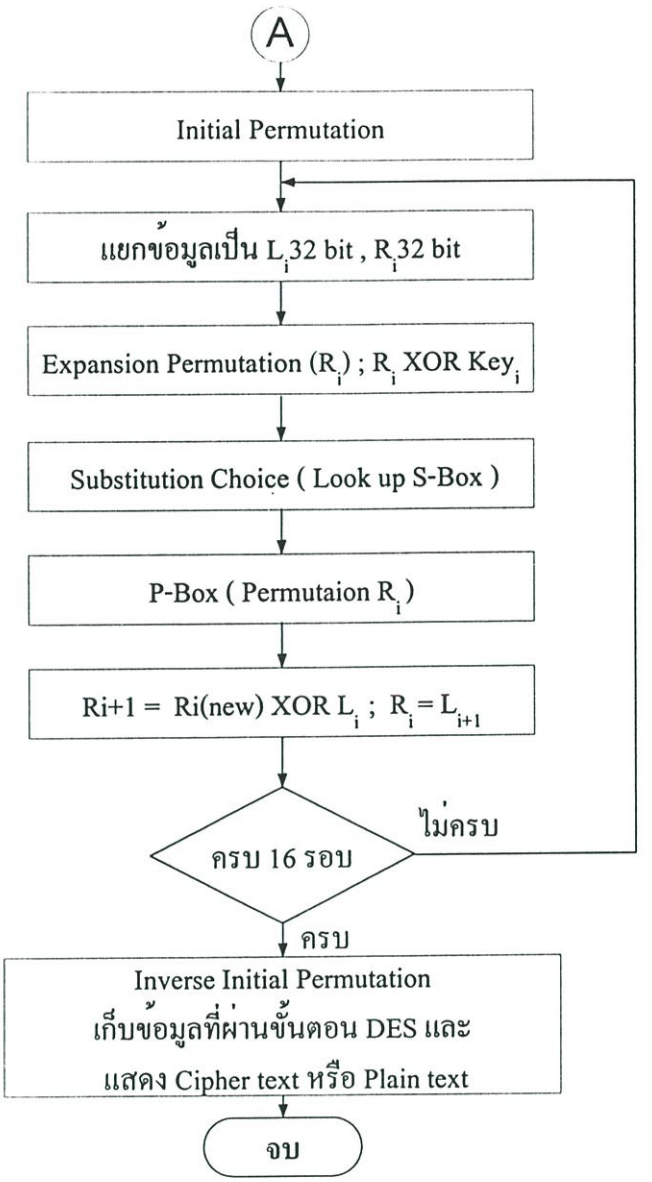
รูปที่ 3.7 แสดง System Flow Diagram

3.3.1 การออกแบบโปรแกรมเพื่อจำลองการทำงานของอัลกอริธึมแบบเก่า

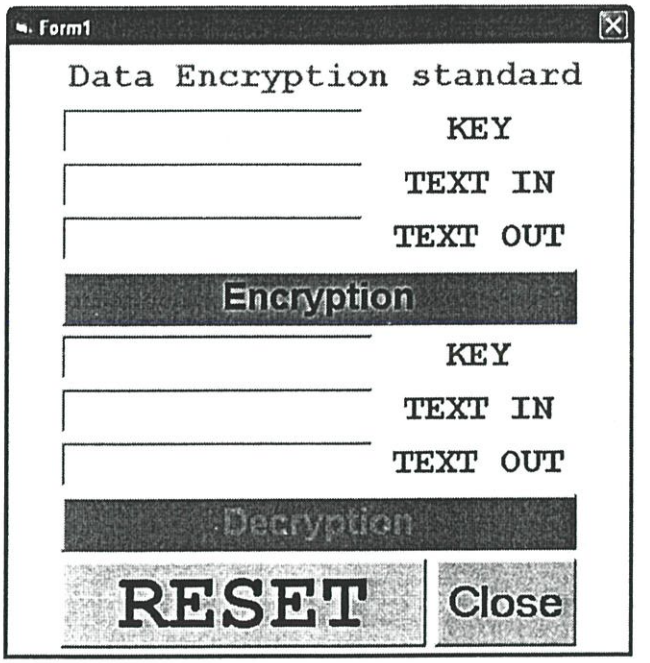
การออกแบบโปรแกรมจำลองการทำงาน แสดง Flow chart ได้ดังรูปที่ 3.8 – 3.9 ซึ่งลำดับขั้นตอนของการเขียนโปรแกรมที่ใช้สำหรับจำลองการทำงานของอัลกอริธึมเข้ารหัสแบบ DES เก่าจาก Flow chart สามารถเขียนโปรแกรมที่ได้ใช้สำหรับงานวิจัยได้ดังแสดงในรูปที่ 3.10 โดยขั้นตอนทุกขั้นตอนจะอ้างอิงอัลกอริธึม DES แบบเก่า ดังนั้นจำนวนรอบที่ใช้ในกระบวนการเข้ารหัสจะใช้ทั้งหมด 16 รอบ



รูปที่ 3.8 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบเก่า (ส่วนที่1)



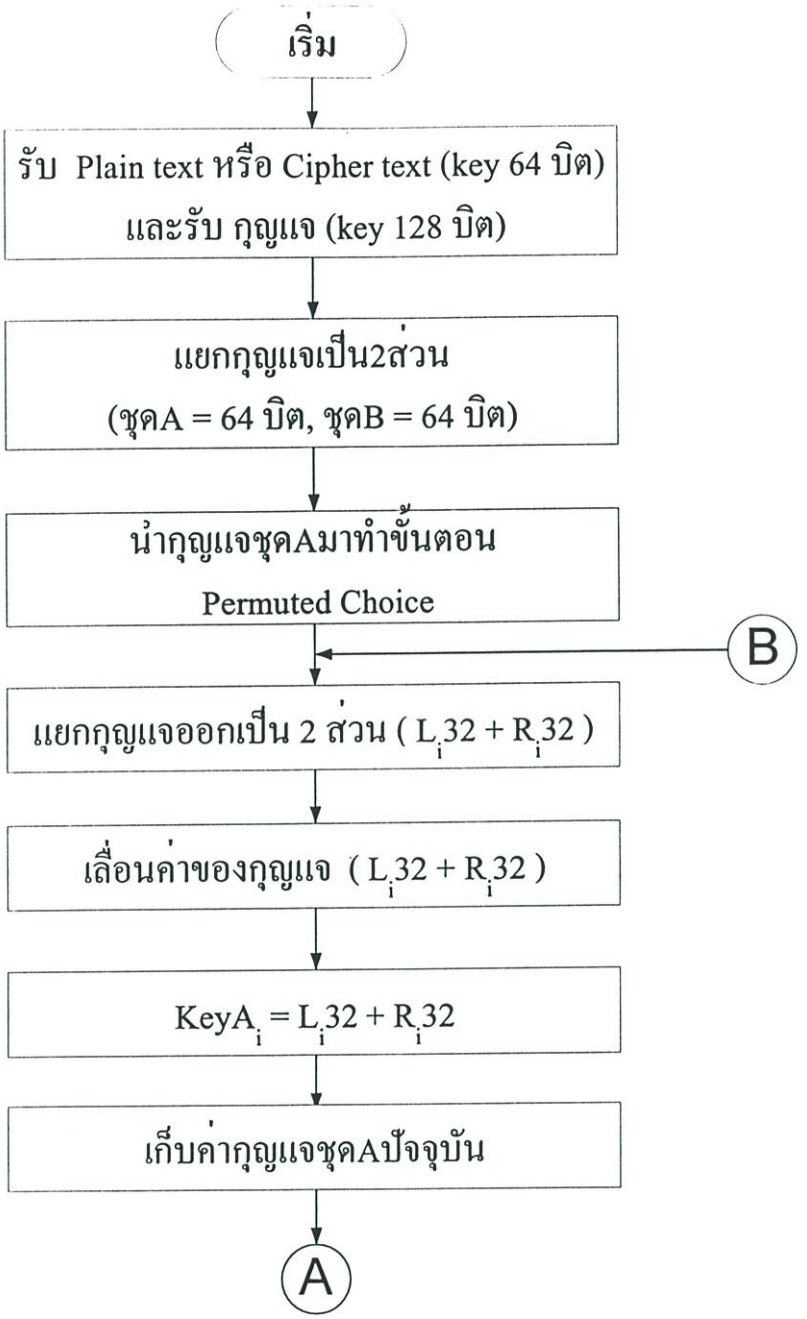
รูปที่ 3.9 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบเก่า (ส่วนที่2)



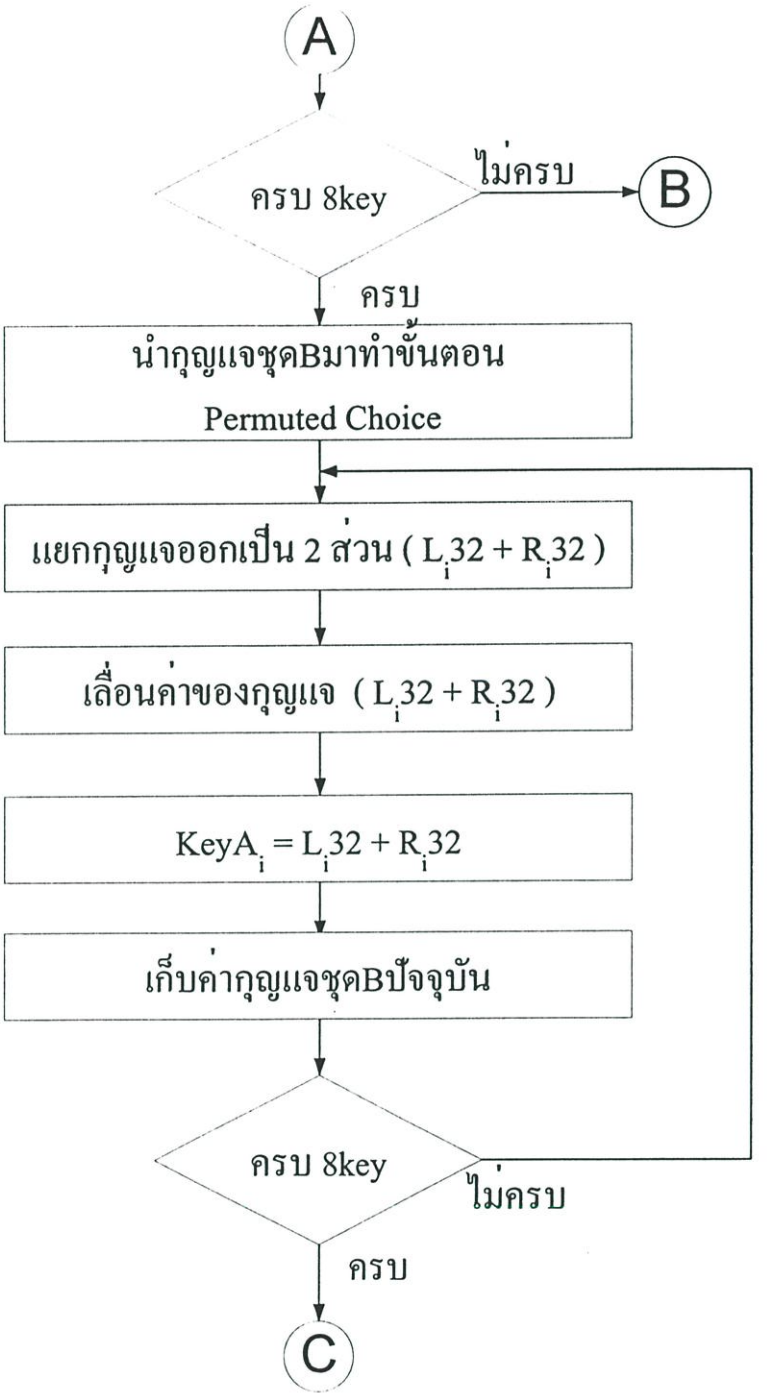
รูปที่ 3.10 แสดงโปรแกรมที่เขียนขึ้นเพื่อใช้ในการทดสอบการเข้ารหัสและรหัส (แบบเก่า)

3.3.2 การออกแบบโปรแกรมเพื่อจำลองการทำงานของอัลกอริธึมแบบใหม่

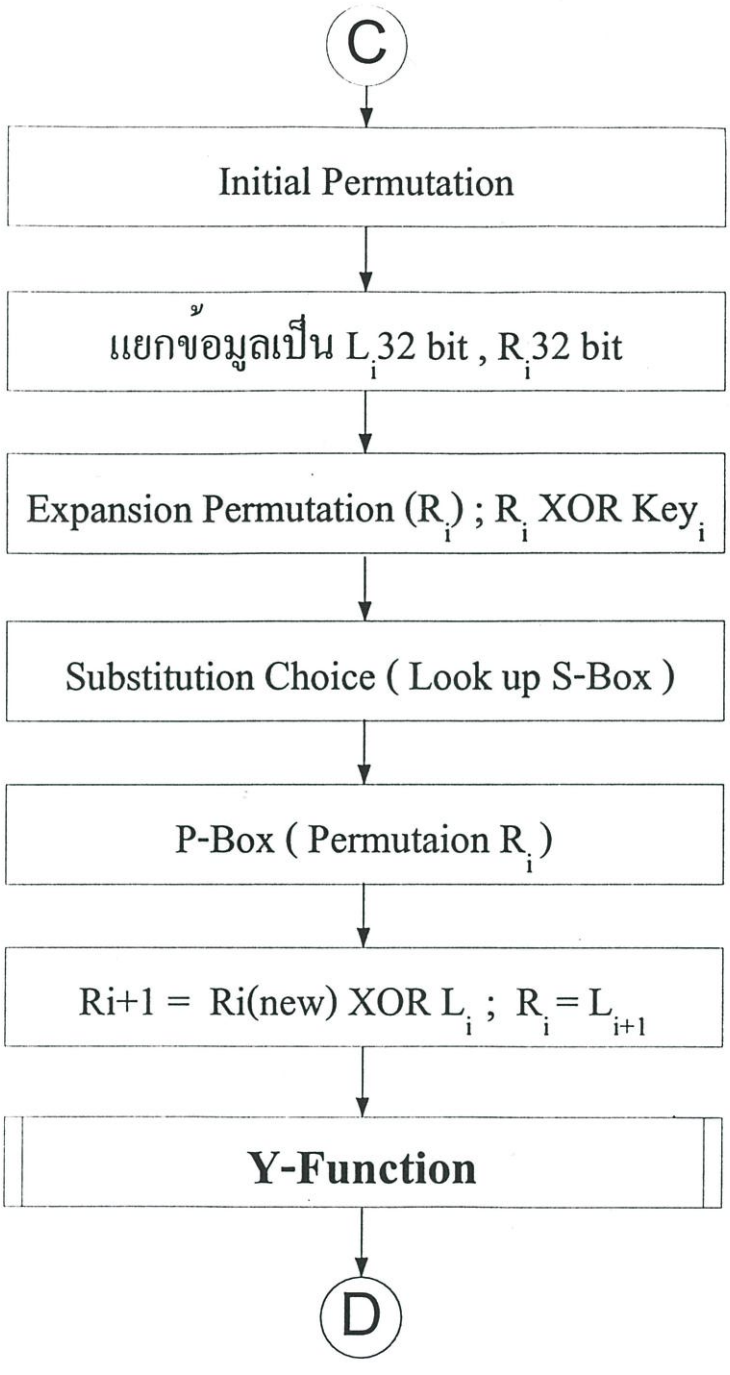
จากรูปที่ 3.11 – 3.14 คือผัง Flow chart ที่แสดงให้เห็นถึงลำดับขั้นตอนของการเขียนโปรแกรมที่ใช้สำหรับจำลองการทำงานของอัลกอริธึมเข้ารหัส DES แบบใหม่ จาก Flow chart สามารถเขียนโปรแกรมที่ใช้สำหรับงานวิจัยได้ดังแสดงในรูปที่ 3.15 โดยจำนวนรอบของการเข้ารหัสที่ใช้ในอัลกอริทึมแบบใหม่ที่นำเสนอนี้จะใช้จำนวนรอบของการสลับสับเปลี่ยนและการแทนที่เพียง 8 รอบเท่านั้น และในส่วนของการรับคีย์รหัสกุญแจจะรับขนาดของคีย์มากกว่าเดิมเป็น 128 บิต ดังนั้นกระบวนการสร้างรหัสกุญแจก็จะแยกเป็นชุด A และชุด B แตะละชุดจะมีการเลื่อนบิตข้อมูลที่แตกต่างกันคือ คีย์ชุด A จะเลื่อนบิตไปทางซ้ายเหมือนเดิมแต่คีย์ชุด B นั้นจะเลื่อนบิตไปทางขวา



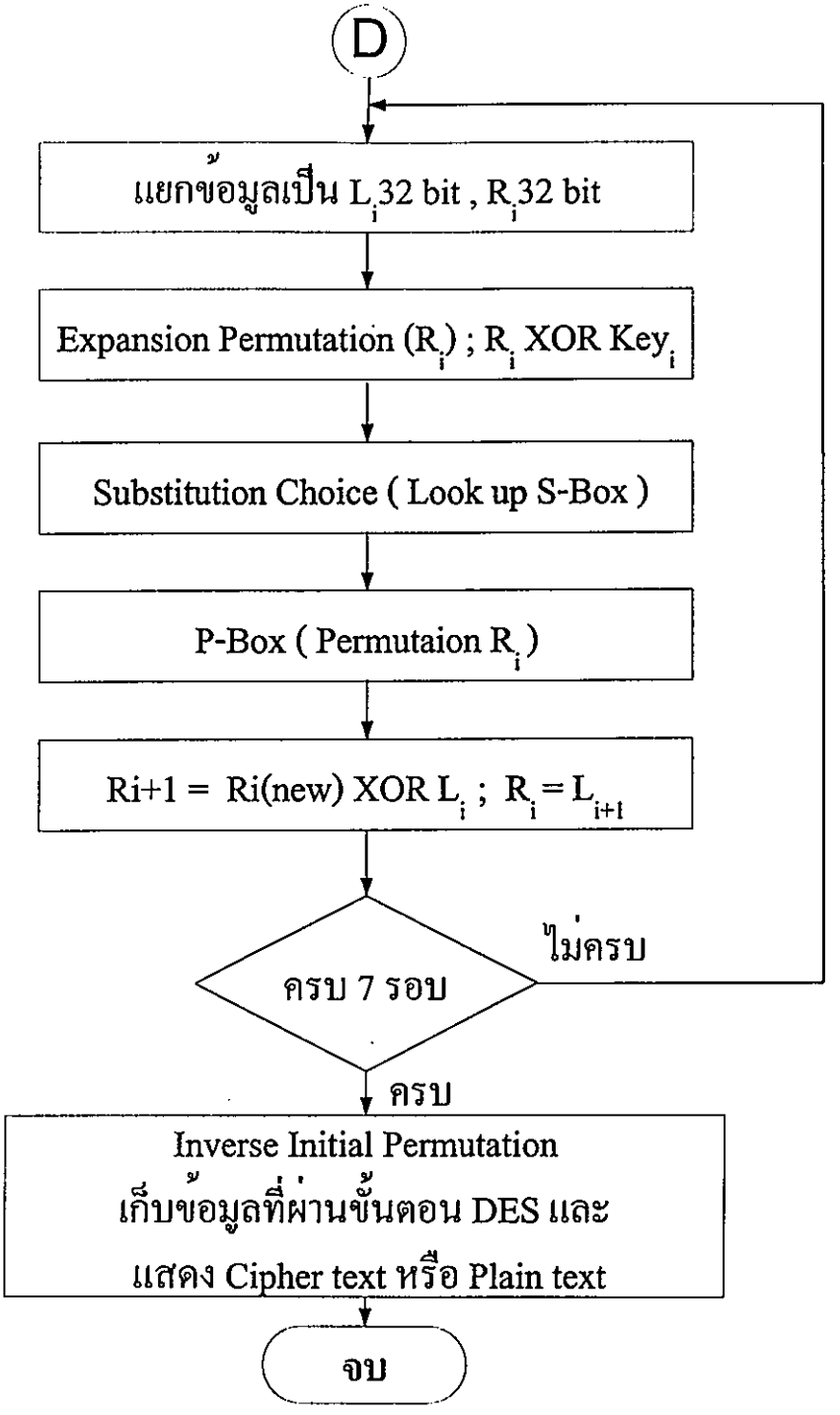
รูปที่ 3.11 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบใหม่ (ส่วนที่1)



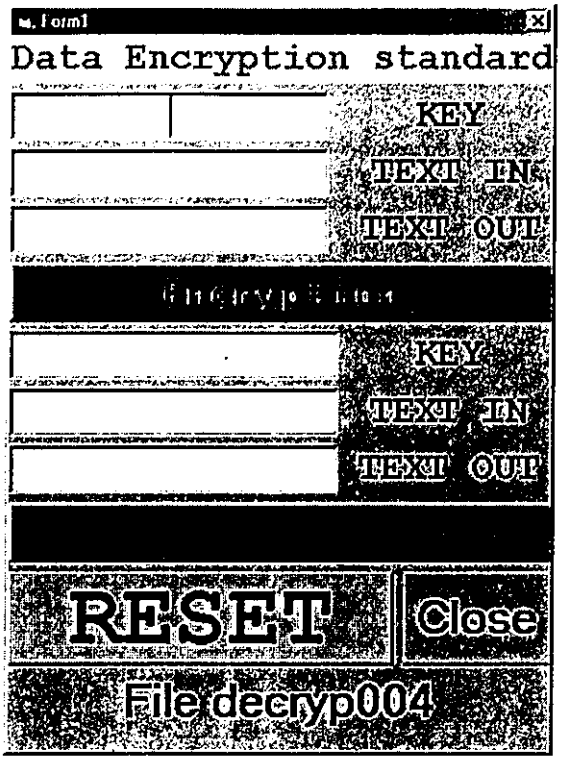
รูปที่ 3.12 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบใหม่ (ส่วนที่2)



รูปที่ 3.13 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบใหม่ (ส่วนที่3)



รูปที่ 3.14 แสดงผังไดอะแกรมขั้นตอนการทำกระบวนการเข้ารหัสและถอดรหัส DES แบบใหม่ (ส่วนที่4)



รูปที่ 3.15 แสดงโปรแกรมที่เขียนขึ้นเพื่อใช้ในการทดสอบการเข้ารหัสและรหัส (แบบใหม่)

บทที่ 4
การทดลองและผลการทดลองที่ได้

4.1 บทนำ

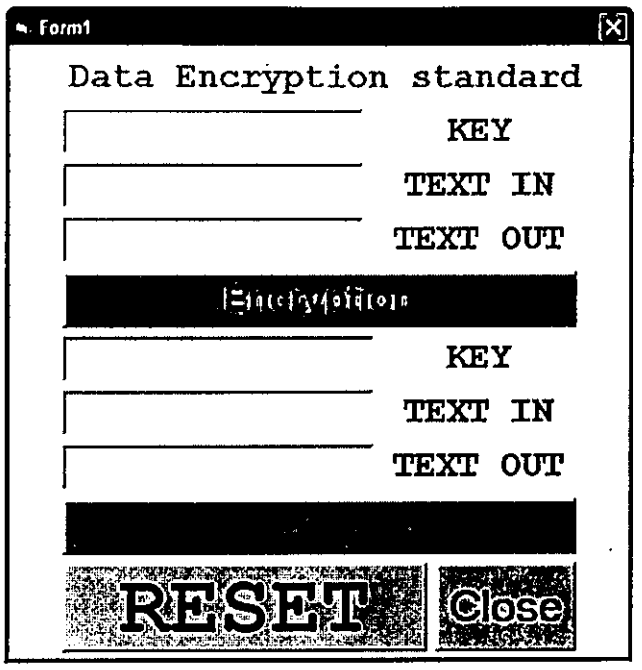
ในบทนี้จะกล่าวถึงผลการทดลองที่ได้จากการทดสอบ อัลกอริธึม DES แบบใหม่ เมื่อนำมาเปรียบเทียบกับ อัลกอริธึม DES แบบเก่า ซึ่งทำโดยการเขียนโปรแกรมที่เป็นประเภท Visual ในที่นี้เลือกใช้ภาษา Visual Basic Version 6 เพื่อใช้เขียนโปรแกรมจำลองการทำงานของอัลกอริธึมทั้ง 2 จากนั้นนำผลที่จากการทดสอบที่ได้ทั้ง 2 มาเปรียบเทียบกัน โปรแกรมที่เขียนขึ้นเพื่อจำลองการทำงานของอัลกอริธึม DES แสดงดังรูปที่ 4.1 และ 4.2 โดยโปรแกรมในรูปที่ 4.1 สร้างขึ้นโดยอาศัยอัลกอริธึม DES แบบใหม่ และ โปรแกรมในรูปที่ 4.2 สร้างขึ้นโดยอาศัยอัลกอริธึม DES แบบเก่า ตามลำดับ สำหรับในขั้นตอนการทดลองจะแบ่งส่วนของการทดลองเป็น 3 ส่วน ดังต่อไปนี้

1. ทำการทดลองเข้ารหัสและถอดรหัสข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่
   1.1 กำหนดให้ DES แบบเก่าและใหม่ ใช้จำนวนรอบของการเข้ารหัส 16 รอบ
   1.2 กำหนดให้ DES แบบเก่า ใช้จำนวนรอบของการเข้ารหัส 16 รอบและใหม่ใช้จำนวนรอบของการเข้ารหัส 8 รอบ
2. ทำการทดสอบเวลาที่ใช้ในการเข้ารหัสและถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่ แล้วสังเกตผลที่ได้ (แสดงในดังตารางที่ 4.1)
3. ทำการทดสอบเวลาที่ใช้ในการลักลอบถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่ แล้วสังเกตผลที่ได้ (แสดงในดังตารางที่ 4.2)

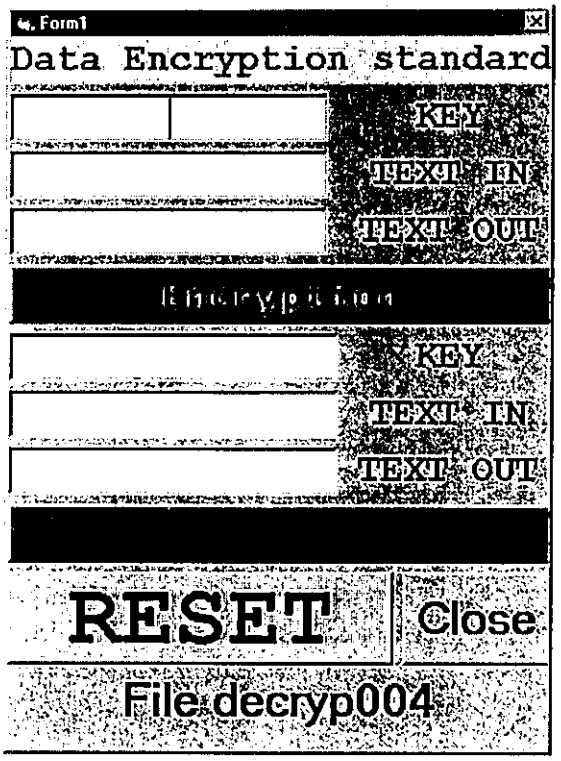
4.2. การทดสอบเข้ารหัสและถอดรหัสข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่

การทดลองและผลการทดลองในนี้ จะทำการทดลองเข้ารหัสและถอดรหัสข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่ โดยใช้โปแกรมในรูปที่ 4.1 และ 4.2 แล้วสังเกตผลที่ได้จากการทดลองเปรียบเทียบการเข้ารหัสข้อความของ DES แบบเก่า และ DES แบบใหม่ ซึ่งในส่วนที่ 1 นั้นจะทำการทดสอบโดยป้อนชนิดของข้อความที่มีความแตกต่างกัน 4 แบบ คือ

1. ข้อความ ที่มีเฉพาะตัวอักษรอย่างเดียวและเป็นชนิดตัวอักษรพิมพ์ใหญ่
2. ข้อความ ที่มีเฉพาะตัวอักษรอย่างเดียวและเป็นชนิดตัวอักษรพิมพ์เล็ก



รูปที่ 4.1 แสดงโปรแกรมที่เขียนขึ้นเพื่อใช้ในการทดสอบการเข้ารหัสและรหัส (แบบเก่า)



รูปที่ 4.2 แสดงโปรแกรมที่เขียนขึ้นเพื่อใช้ในการทดสอบการเข้ารหัสและรหัส (แบบใหม่)

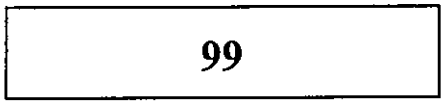
3. ข้อความ ที่มีเฉพาะตัวเลขอย่างเดียว

4. ข้อความ ที่มีทั้งตัวเลขและตัวอักษร

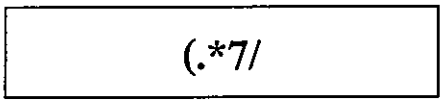
4.2.1. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 99, คำที่ใช้ทดสอบ คือ "KMITL" ทำการทดสอบเฉพาะตัวอักษร (ตัวอักษรพิมพ์ใหญ่)

ผลการทดลองที่ได้

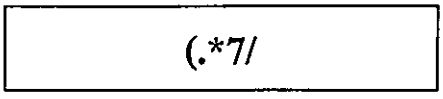
รูปที่ 4.3 แสดงข้อความต้นแบบ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ1



รูปที่ 4.4 แสดงค่าของKEYที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ1



รูปที่4.5 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ1



รูปที่4.6 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ1

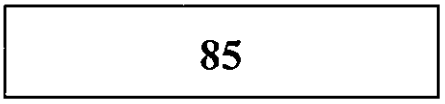
รูปที่4.7 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ1

รูปที่4.8 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ1

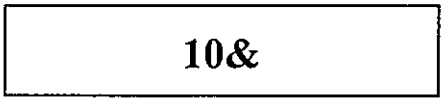
4.2.2. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 85, คำที่ใช้ทดสอบคือ "des " ทำการทดสอบเฉพาะตัวอักษร (ตัวอักษรพิมพ์เล็ก)

ผลการทดลองที่ได้

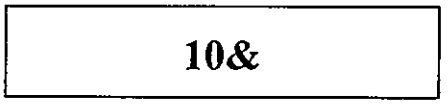
รูปที่ 4.9 แสดงข้อความต้นแบบ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ2



รูปที่ 4.10 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ2



รูปที่4.11 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ2



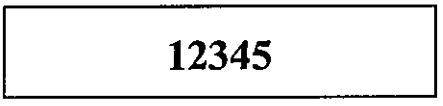
รูปที่4.12 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ2

รูปที่4.13 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ2

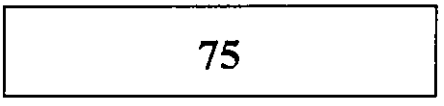
รูปที่4.14 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ2

4.2.3. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 75 ทำการทดสอบเฉพาะตัวเลข

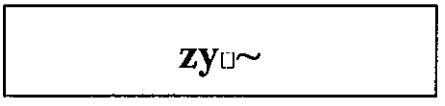
ผลการทดลองที่ได้



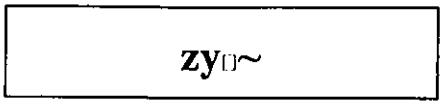
รูปที่ 4.15 แสดงข้อความต้นแบบ ส่วนที่1 ข้อ3



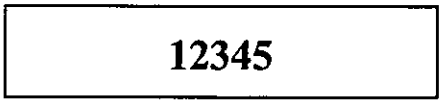
รูปที่ 4.16 แสดงค่าของKEYที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ3



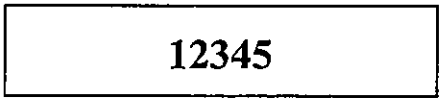
รูปที่4.17 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ3



รูปที่4.18 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DESแบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ3



รูปที่4.19 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ3

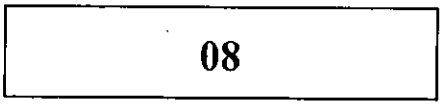


รูปที่4.20 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ3

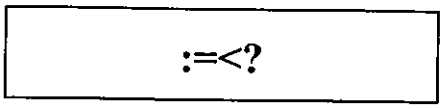
4.2.4. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 08 ทำการทดสอบเฉพาะตัวเลข

ผลการทดลองที่ได้

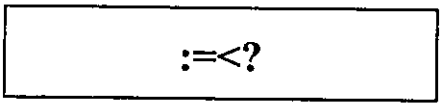
รูปที่ 4.21 แสดงข้อความต้นแบบ ส่วนที่1 ข้อ4



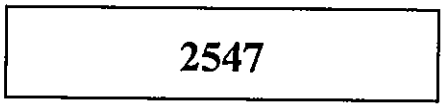
รูปที่ 4.22 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ4



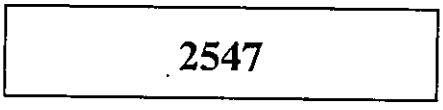
รูปที่4.23 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ4



รูปที่4.24 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ4



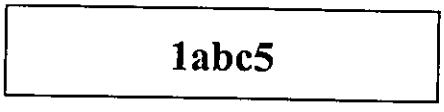
รูปที่4.25 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ4



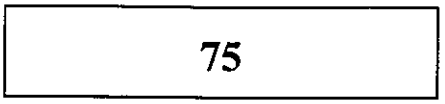
รูปที่4.26 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ4

4.2.5. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 75 ทำการทดสอบตัวอักษรและตัวเลข

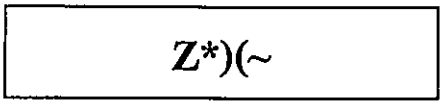
ผลการทดลองที่ได้



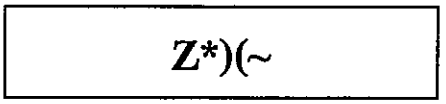
รูปที่ 4.27 แสดงข้อความต้นแบบ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ5



รูปที่ 4.28 แสดงค่าของKEYที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ5



รูปที่4.29 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ5



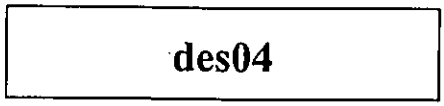
รูปที่4.30 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DESแบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ5

รูปที่4.31 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ5

รูปที่4.32 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ5

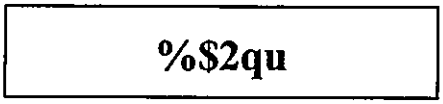
4.2.6. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบ, ใช้คีย์รหัส 65 ทำการทดสอบตัวอักษรและตัวเลข

ผลการทดลองที่ได้

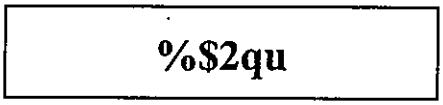


รูปที่ 4.33 แสดงข้อความต้นแบบ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ6

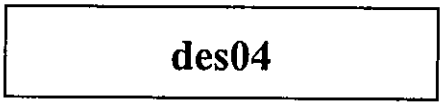
รูปที่ 4.34 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ6



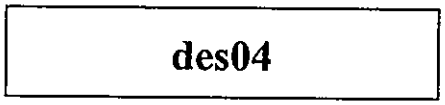
รูปที่4.35 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ6



รูปที่4.36 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ6



รูปที่4.37 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ6



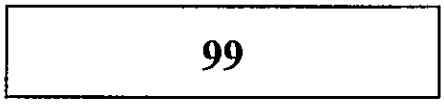
รูปที่4.38 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ6

ดังนั้นจึงสรุปได้ว่าการทดลองที่ 4.2.1 – 4.2.6 DES แบบเก่า และ DES แบบใหม่ที่สร้างขึ้นมา สามารถที่จะปกปิดข้อมูลได้เหมือนกันในกรณีของการเข้ารหัส และในกรณีของการถอดรหัสก็สามารถถอดรหัสได้ข้อความเหมือนเดิมทุกประการ โดยไม่เกิดข้อผิดเพี้ยนแต่อย่างใด

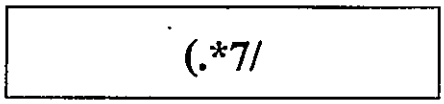
4.2.7. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและ DES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับDES เก่า, 8 รอบสำหรับ DES ใหม่ และ ใช้คีย์รหัส 99 ทำการทดสอบเฉพาะตัวอักษรพิมพ์ใหญ่

ผลการทดลองที่ได้

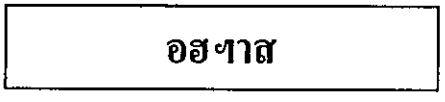
รูปที่ 4.39 แสดงข้อความต้นแบบ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ7



รูปที่ 4.40 แสดงค่าของKEYที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ7



รูปที่4.41 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ7



รูปที่4.42 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DESแบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ7

รูปที่4.43 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ7

KMITL

รูปที่4.44 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ7

4.2.8. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับDES เก่า, 8 รอบสำหรับDES ใหม่ และ ใช้คีย์รหัส 25 ทำการทดสอบเฉพาะตัวอักษรพิมพ์เล็ก

ผลการทดลองที่ได้

kmitl

รูปที่ 4.45 แสดงข้อความต้นแบบ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ8

25

รูปที่ 4.46 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ8

rtpmu

รูปที่4.47 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ8

WQUHP

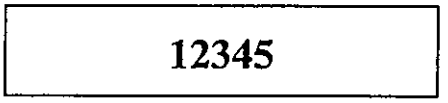
รูปที่4.48 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ8

รูปที่4.49 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวอักษร) ส่วนที่1 ข้อ8

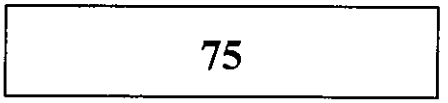
รูปที่4.50 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวอักษร) ส่วนที่1 ข้อ8

4.2.9. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและDES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับDES เก่า, 8 รอบสำหรับDES ใหม่ และ ใช้คีย์รหัส 75 ทำการทดสอบเฉพาะตัวเลข

ผลการทดลองที่ได้

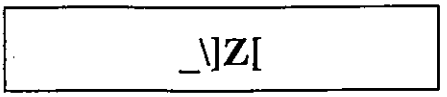


รูปที่ 4.51 แสดงข้อความต้นแบบ ส่วนที่1 ข้อ9

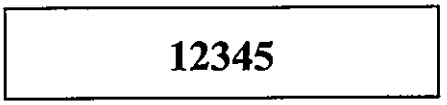


รูปที่ 4.52 แสดงค่าของKEYที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ9

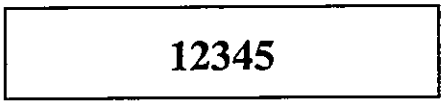
รูปที่4.53 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ9



รูปที่4.54 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ9



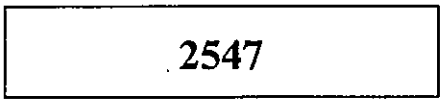
รูปที่4.55 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ9



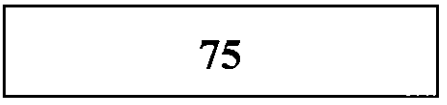
รูปที่4.56 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่(เฉพาะตัวเลข) ส่วนที่1 ข้อ9

4.2.10. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและ DES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับ DES เก่า, 8 รอบสำหรับ DES ใหม่และ ใช้คีย์รหัส 75 ทำการทดสอบเฉพาะตัวเลข

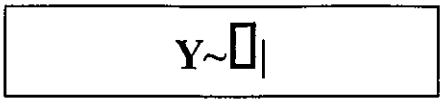
ผลการทดลองที่ได้



รูปที่ 4.57 แสดงข้อความต้นแบบ ส่วนที่1 ข้อ10

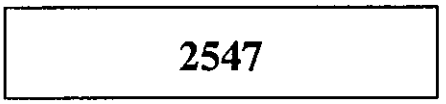


รูปที่ 4.58 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ10

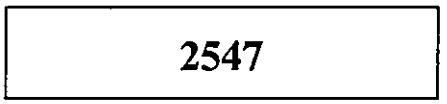


รูปที่4.59 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ10

รูปที่4.60 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ10



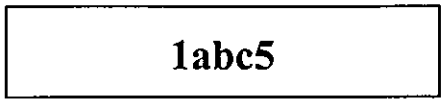
รูปที่4.61 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (เฉพาะตัวเลข) ส่วนที่1 ข้อ10



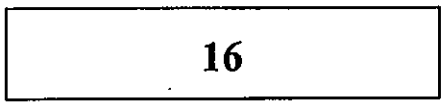
รูปที่4.62 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (เฉพาะตัวเลข) ส่วนที่1 ข้อ10

4.2.11. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและ DES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับDES เก่า, 8 รอบสำหรับDES ใหม่และ ใช้คีย์รหัส 16 ทำการทดสอบเฉพาะตัวอักษรและตัวเลข

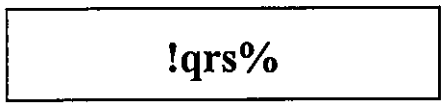
ผลการทดลองที่ได้



รูปที่ 4.63 แสดงข้อความต้นแบบ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ11

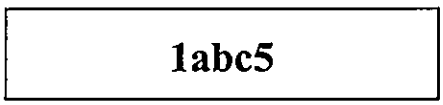


รูปที่ 4.64 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ11

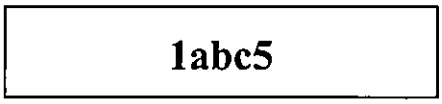


รูปที่4.65 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ11

รูปที่4.66 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ11



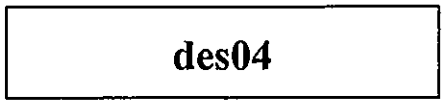
รูปที่4.67 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ11



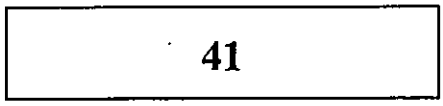
รูปที่4.68 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ11

4.2.12. ทดสอบโปรแกรมที่เขียนขึ้นโดยใช้หลักการของ DES เก่าและ DES ใหม่ ทำการเข้าและถอดรหัส โดยใช้จำนวนรอบในการเข้ารหัส 16 รอบสำหรับDES เก่า, 8 รอบสำหรับDES ใหม่และ ใช้คีย์รหัส 41 ทำการทดสอบเฉพาะตัวอักษรและตัวเลข

ผลการทดลองที่ได้

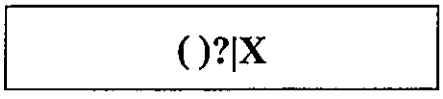


รูปที่ 4.69 แสดงข้อความต้นแบบ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ12

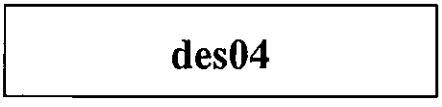


รูปที่ 4.70 แสดงค่าของ KEY ที่ใช้ในการเข้ารหัส ส่วนที่1 ข้อ12

รูปที่4.71 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ12



รูปที่4.72 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการเข้ารหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ12



รูปที่4.73 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบเก่า (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ12

des04

รูปที่4.74 แสดงข้อความต้นแบบที่ผ่านขั้นตอนการถอดรหัส DES แบบใหม่ (ตัวเลขและตัวอักษร) ส่วนที่1 ข้อ12

ดังนั้นจึงสรุปผลการทดลองได้ว่าการทดลองที่ 4.2.7-1.2.12 DES แบบเก่า และ DES แบบใหม่ที่สร้างขึ้นมา สามารถที่จะปกปิดข้อมูลได้เหมือนกันในกรณีของการเข้ารหัส และในกรณีของการถอดรหัสก็สามารถถอดรหัสได้ข้อความเหมือนเดิมทุกประการ โดยไม่เกิดข้อผิดเพี้ยนแต่อย่างใด แม้ว่าจำนวนรอบแบบใหม่จะน้อยกว่าก็ยังสามารถปกปิดข้อมูลได้เช่นเดิม (จำนวนรอบในการเข้ารหัสอย่างน้อยเพียง 8 รอบก็เพียงพอสำหรับการปกปิดข้อมูลได้ [3]) โดยเฉพาะ function พิเศษที่เพิ่มเข้าไปก็ไม่มีผลต่อกระบวนการเข้ารหัสและถอดรหัส เพราะผลที่ได้จากงานวิจัย ยังสามารถเข้ารหัสและถอดรหัสได้เหมือน DES แบบเก่า

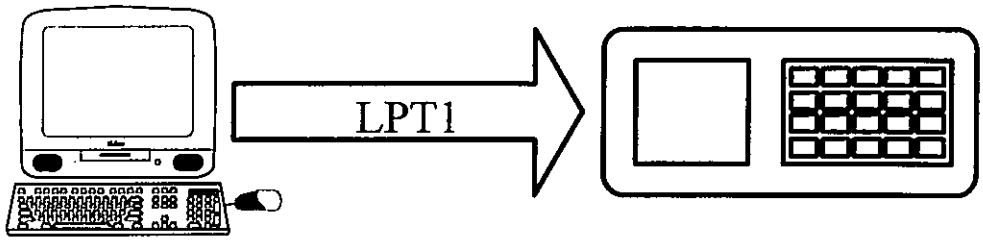
4.3 การทดสอบเวลาที่ใช้ในการเข้ารหัสและถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่

ทำทดสอบเวลาที่ใช้ในการเข้ารหัสและถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่ แล้วสังเกตผลที่ได้ดังตารางที่ 4.1 โดยใช้เงื่อนไขดังต่อไปนี้

1. จำนวนของ CYCLE ของการทำขั้นตอนสลับสับเปลี่ยนใช้ทั้งหมด 8 รอบ สำหรับ DES แบบใหม่

2. จำนวนของ CYCLE ของการทำขั้นตอนสลับสับเปลี่ยนใช้ทั้งหมด 16 รอบ สำหรับ DES แบบเก่า

การทดสอบในหัวข้อ4.3 นี้จะใช้อัลกอริทึมของ DES แบบใหม่และ DES แบบเก่า ทดสอบเวลาในการเข้ารหัสและถอดรหัสบนเครื่องคอมพิวเตอร์ที่ใช้ CPU รุ่นPentium 4 ความเร็ว 2 GHz และในการทดสอบเวลาในการเข้ารหัสและถอดรหัสนั้นในวิทยานิพนธ์นี้ใช้วิธีการทดสอบโดยทำการเขียนโปรแกรมเพิ่มเข้าไปใน DES แบบเก่าและ DES แบบใหม่ให้ส่งสัญญาณออกมาทาง Port Printer แล้วใช้ Digital Storage Osciloscope100 MHz ของTektronix รุ่น2221 ทำการวัดคาบเวลาที่ได้จาการเข้ารหัสและถอดรหัสของ DES แบบเก่าและ DES แบบใหม่ แสดงดังรูปที่ 4.75



รูปที่4.75 แสดงการต่อ Digital Storage Osciloscope วัดคาบเวลาของการเข้ารหัสและถอดของ DES แบบเก่าและ DES แบบใหม่

ตารางที่ 4.1 แสดงเวลาที่ใช้ในการเข้ารหัสและถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่ CPU Pentium4 2GHz

	DES(OLD)	DES(NEW)
เวลาที่ใช้ในการเข้ารหัส	**45.6 mSec**	**21.4 mSec**
เวลาที่ใช้ในการถอดรหัส	**44.8 mSec**	**20.6 mSec**

ดังนั้นจึงสรุปผลการทดลองได้ว่า DES แบบใหม่สามารถที่จะถอดรหัสและเข้ารหัสได้รวดเร็วกว่า DES แบบเก่า ด้วยจำนวนของรอบ (Cycle) ที่ลดลง และ Function พิเศษที่เพิ่มเข้าไปก็มิได้ทำให้ความเร็วในการเข้ารหัสและถอดรหัสของ DES ลดลงแต่อย่างใด

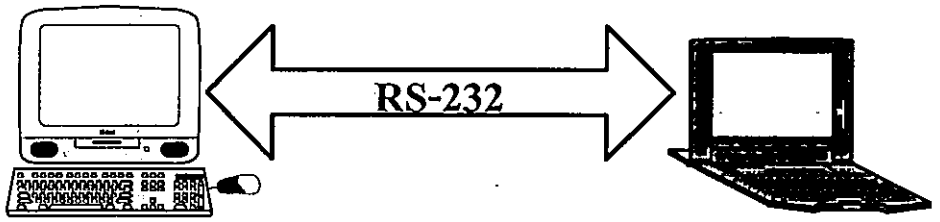
4.4 การทดสอบเวลาที่ใช้ในการลักลอบถอดรหัส ข้อความโดยใช้ DES แบบเก่า และ DES แบบใหม่

ทำทดสอบเวลาที่ใช้ในการลักลอบถอดรหัสโดยเปรียบเทียบระหว่าง DES แบบเก่า และ DES แบบใหม่ (ใช้ CPU Pentium 120MHz, Pentium 4 2GHz โดยทำการป้อนค่ารหัส ตั้งแต่ 0 - 2^{16}) สังเกตผลที่ได้ดังตารางที่ 4.2 และ ตารางที่ 4.3 ตามลำดับ โดยใช้เงื่อนไขดังต่อไปนี้

1. จำนวนของ CYCLE ของการทำขั้นตอนสลับสับเปลี่ยนใช้ทั้งหมด 8 รอบ สำหรับ DES แบบใหม่

2. จำนวนของ CYCLE ของการทำขั้นตอนสลับสับเปลี่ยนใช้ทั้งหมด 16 รอบ สำหรับ DES แบบเก่า

การทดสอบในหัวข้อนี้จะใช้โปรแกรมที่เขียนขึ้นโดยใช้อัลกอริทึมของ DES แบบใหม่และ DES แบบเก่า ทดสอบเวลาในการลักลอบถอดรหัสบนเครื่องคอมพิวเตอร์ที่ใช้ CPU รุ่นPentium ความเร็ว 120 MHz และเครื่องคอมพิวเตอร์ที่ใช้ CPU รุ่น Pentium 4 ความเร็ว 2 GHz เพื่อเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัส (**DES แบบใหม่ สามารถที่หน่วงเวลาได้คงที่หรือไม่ถ้าอุปกรณ์ที่ใช้การเข้ารหัสมีความเร็วในการถอดรหัสสูงขึ้น**) ในการทดสอบเวลาในการลักลอบถอดรหัสนั้นในวิทยานิพนธ์นี้ใช้วิธีการทดสอบโดยทำการเขียนโปรแกรมเพิ่มเข้าไปใน DES แบบเก่าและ DES แบบใหม่ให้รับ-ส่ง ข้อมูล Text (ข้อความที่ต้องการเข้ารหัส) และ Code (รหัสผ่าน) ผ่าน Port Com1 (RS-232) ของเครื่องคอมพิวเตอร์ แล้วใช้บอร์ด Microcontroller (ใช้ CPU เบอร์ PIC18F458) ที่สร้างขึ้นเอง ทำการวัดคาบเวลาที่ได้จากการลักลอบถอดรหัสของ DES แบบเก่าและ DES แบบใหม่ ซึ่งผังแสดงแสดงการทำงานของโปรแกรมที่เขียนขึ้นสำหรับใช้ในการวัดคาบเวลาที่ได้จากการลักลอบถอดรหัสและทำการลักลอบถอดรหัสแสดงดังรูปที่ 4.77 และในรูปที่ 4.76 แสดงถึงการเชื่อมต่อ PC Computer ผ่านPort RS-232 ไปยัง Microcontroller Board เพื่อให้ Microcontroller Board ทำการลักลอบถอดรหัส พร้อมทั้งตรวจสอบเวลาที่ใช้ในการลักลอบถอดรหัส และรูปที่ 4.77 แสดง Microcontroller Board ต้นแบบที่สร้างขึ้นเพื่อใช้สำหรับทดสอบในวิทยานิพนธ์ฉบับนี้



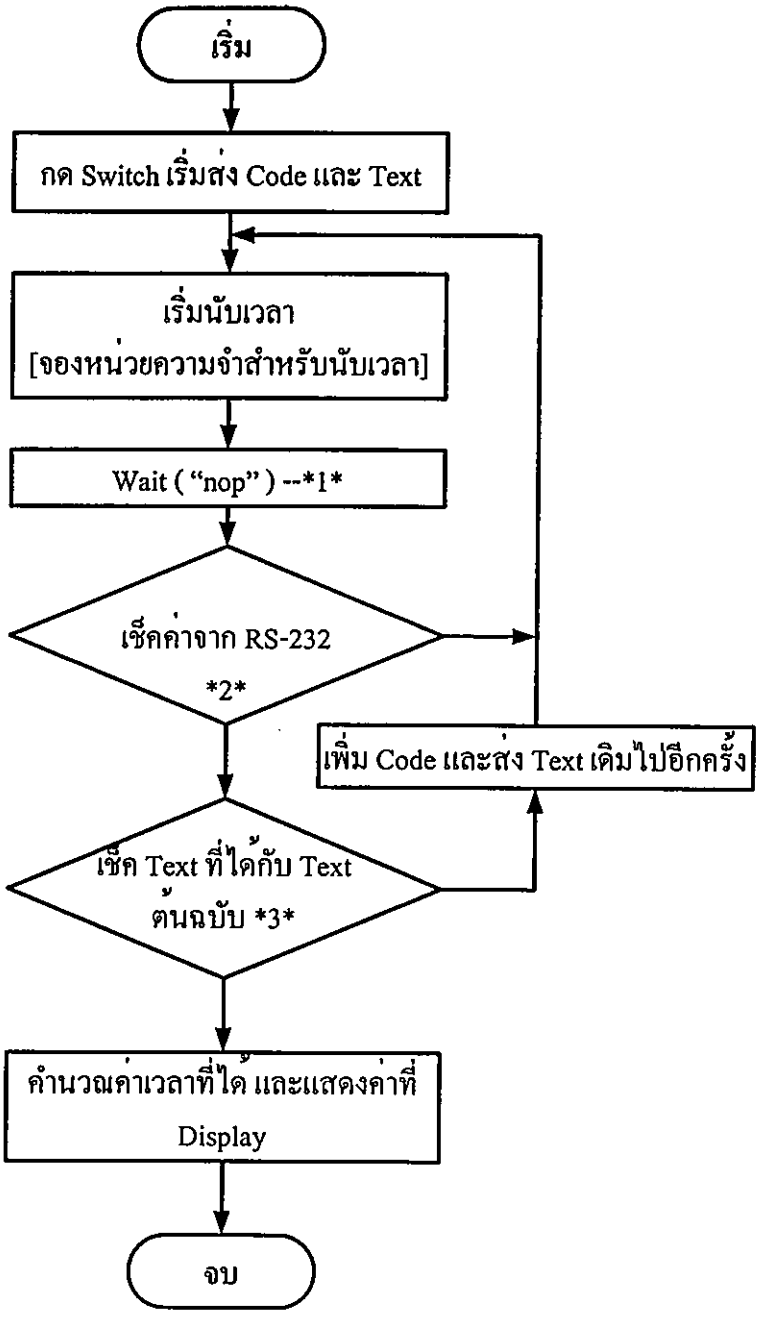
รูปที่4.76 แสดงการต่อ PC Computer ผ่าน RS-232 ไปยัง Microcontroller Board เพื่อให้ Microcontroller Board ทำการลักลอบถอดรหัส พร้อมทั้งตรวจสอบเวลาที่ใช้ในการลักลอบถอดรหัส

ตารางที่ 4.2 แสดงการเปรียบเทียบ เวลาที่ใช้ในการลักลอบถอดรหัส ของ DES แบบเก่า และ DES แบบใหม่ โดยใช้ CPU Pentium 120MHz

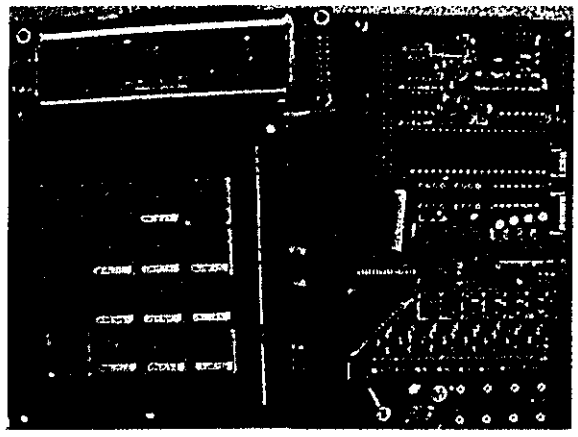
CPU Pentium 120MHz		
	DES (old)	DES (new)
เวลาที่ใช้ในการถอดรหัส	**2936 Sec**	**65531 Sec**

ตารางที่ 4.3 แสดงการเปรียบเทียบ เวลาที่ใช้ในการลักลอบถอดรหัส ของ DES แบบเก่า และ DES แบบใหม่ โดยใช้ CPU Pentium4 2GHz

CPU Pentium 4 2GHz		
	DES (old)	DES (new)
เวลาที่ใช้ในการถอดรหัส	**195 Sec**	**65522 Sec**



รูปที่4.77 แสดงผังการออกแบบโปรแกรมสำหรับลักลอบถอดรหัสและวัดคาบเวลาที่ใช้ในการลักลอบถอดรหัส



รูปที่4.78 แสดง Microcontroller Board สำหรับใช้ทำการลักลอบถอดรหัส พร้อมทั้งตรวจสอบเวลาที่ใช้ในการลักลอบถอดรหัส

การทดสอบในหัวข้อ4.4นี้ ไม่สามารถที่จะทำการทดสอบการลักลอบถอดรหัส โดยใช้รหัสที่มีความยาว 64 บิตเพื่อเปรียบเทียบกับ DES แบบมาตรฐานได้ เนื่องจากจะต้องเวลาในการทดสอบประมาณ **0.583344214028965278283748229103 42** ล้านล้านปี (ค่านี้ได้จากการคำนวณ) จึงทำการทดลองโดยใช้รหัสที่มีความยาวเพียง 16 บิตเท่านั้น เพื่อพิสูจน์ให้เห็นว่า DES ที่นำเสนอนั้นสามารถที่จะหน่วงเวลาได้จริงเมื่อเกิดการลักลอบถอดรหัส โดยเวลาที่หน่วงเมื่อเกิดการลักลอบถอดรหัสนั้นจะจะคงที่ ถึงแม้อุปกรณ์ที่ใช้ในการลักลอบถอดรหัสจะมีความเร็วสูงสักเพียงใดก็ตาม ในที่นี้จะทำการทดสอบโดยใช้เครื่องคอมพิวเตอร์ 2 เครื่องที่ใช้ CPU รุ่นPentium ความเร็ว 120 MHz และเครื่องคอมพิวเตอร์ที่ใช้ CPU รุ่น Pentium 4 ความเร็ว 2 GHz เพื่อทำการเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัส (สังเกตค่าเวลาหน่วงจากตารางที่ 4.2 และตารางที่ 4.3) และถ้าหากจะคำนวณเวลาที่ใช้ในการลักลอบถอดรหัส โดยใช้รหัสที่มีความยาว 128 บิต (ที่นำเสนอในวิทยานิพนธ์ฉบับนี้) จะต้องเวลาในการทดสอบการลักลอบถอดรหัส ประมาณ **2.582592341537177166540487305948 5 X10^{32}** ปี

4.5 วิเคราะห์ผลการทดลอง

ในการทดลอง DES แบบใหม่ที่ได้นำเสนอในวิทยานิพนธ์เล่มนี้เพื่อนำผลที่ได้ไปเปรียบเทียบกับ DES แบบเก่านั้นสามารถทำการทดลองให้เห็นค่าได้ โดยใช้ความยาวของขนาดรหัสคีย์เพียง 16 บิตเท่านั้น ส่วนในการทดลองคีย์รหัสขนาด 64 บิตนั้นจะใช้วิธีการคำนวณเนื่องจากจะต้องใช้เวลาในการทดสอบนานมาก (สังเกตจากตารางที่ 4.4 และ ตารางที่ 4.5) ตารางที่ 4.4 และ ตารางที่ 4.5 เป็นตารางที่แสดงการคำนวณเวลาที่ใช้ในการลักลอบถอดรหัสเมื่อใช้คีย์รหัสเป็น 4, 8, 16, 32, 64 ตามลำดับ สังเกตว่าเมื่อใช้ขนาดของคีย์รหัสยิ่งมากเวลาที่ใช้ในการลักลอบถอดรหัสนาน และในตารางที่ 4.6 จะแสดงให้เห็นถึงเวลาเฉลี่ยที่ได้จากการทดสอบจริงจากการถอดรหัสของ DES แบบเก่า และ DES แบบใหม่ (ต่อการถอดรหัส 1 ครั้ง) เพื่อจะนำค่าที่ได้ไปใช้คำนวณหาค่าในตารางที่4.7,4.8,4.9, 4.10 ตามลำดับ

ตารางที่ 4.4 แสดงค่าที่ได้จากการคำนวณเวลาหน่วง (ในทางอุดมคติ) เมื่อเกิดการลักลอบถอดรหัสของ DES แบบใหม่เมื่อใช้จำนวนบิตของรหัสเป็น 4,8,16,32,64 (หน่วยวัดเป็น วินาที และนาที)

จำนวนบิตที่ใช้	วินาที	นาที
4	16	
8	256	4.26
16	65,536	1,092.26
32	4,294,967,296	71,582,788.26
64	18,446,744,073,709,551,616	307,445,734,561,825,860.26

ตารางที่ 4.5 แสดงค่าที่ได้จากการคำนวณเวลาหน่วง (ในทางอุดมคติ) เมื่อเกิดการลักลอบถอดรหัสของ DES แบบใหม่เมื่อใช้จำนวนบิตของรหัสเป็น 4,8,16,32,64 (หน่วยวัดเป็น ชั่วโมงและปี)

จำนวนบิตที่ใช้	ชั่วโมง	ปี
4		
8		
16	18.20	
32	1,193,046.47	136.19
64	5,124,095,576,030,431	584,942,417,355.07

ตารางที่ 4.6 แสดงค่าเฉลี่ยที่ได้จากผลการทดสอบเวลาหน่วงใน 1 รอบของการลักลอบถอดรหัส ของ DES แบบใหม่และแบบเก่า

	120 MHz	2GHz
แบบเก่า	0.04479 sec	0.00297 sec
แบบใหม่	0.9999237 sec	0.9997863 sec

ตารางที่ 4.7 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของ DES <u>แบบใหม่</u> (คำนวณจากค่าเวลาต่อรอบที่ได้จากการทดสอบคูณกับเวลาจริง) ใช้ **Pentium4 2GHz**

จำนวนบิตที่ใช้	วินาที	นาที	ชั่วโมง	ปี
4	15.99			
8	255.94	4.27		
16	65,521.99	1,092.033	18.20	
32	4,294,049,461.48	71,567,491.02	1,192,791.52	136.16
64	18,442,802,004,500,999,884.81	307,380,033,408,349,998.08	5,123,000,556,805,833.3	584,817,415,160.48

ตารางที่ 4.8 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของ DES <u>แบบใหม่</u> (คำนวณค่าเวลาต่อรอบที่ได้จากการทดสอบคูณกับเวลาจริง) ใช้ **Pentium 120 MHz**

จำนวนบิตที่ใช้	วินาที	นาที	ชั่วโมง	ปี
4	15.99			
8	255.98	4.26		
16	65,530.99	1,092.18	18.203	
32	4,294,639,589.99	71,577,326.49	1,192,955.44	136.18
64	18,445,336,587,136,727,577.21	307,422,276,452,278,792.95	5,123,704,607,537,979.88	584,897,786,248.62

ตารางที่ 4.9 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของ DES <u>แบบเก่า</u> (คำนวณค่าเวลาต่อรอบที่ได้จากการทดสอบคูณกับเวลาจริง) ใช้ **Pentium4 2GHz**

จำนวนบิตที่ใช้	วินาที	นาที	ชั่วโมง	ปี
4	0.047			
8	0.76	0.0126		
16	194.64	3.24		
32	12,756,052.86	212,600.88	3,543.35	0.4044
64	54,786,829,898,917,368.29	913,113,831,648,622.81	15,218,563,860,810.38	1,737,278,979.55

ตารางที่ 4.10 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของ DES <u>แบบเก่า</u> (คำนวณค่าเวลาต่อรอบที่ได้จากการทดสอบคูณกับเวลาจริง) ใช้ **Pentium 120 MHz**

จำนวนบิตที่ใช้	วินาที	นาที	ชั่วโมง	ปี
4	0.71			
8	11.46	0.19		
16	2,935.35	48.92	0.81	
32	192,371,585.19	3,206,193.09	53,436.55	6.1
64	826,229,667,061,450,816.88	13,770,494,451,024,180.28	229,508,240,850,403	26,199,570,873.33

ตารางที่4.7 จะเป็นการแสดงค่าที่ได้จากการนำค่าในตารางที่4.6 ไปคำนวณหาค่าเวลาเมื่อเกิดการลักลอบถอดรหัส ของ DES แบบใหม่โดยใช้ Pentium4 2GHz (ตารางที่4.7) และ Pentium 120 MHz (ตารางที่4.8) ซึ่งขนาดของรหัสคีย์ที่ใช้มีดังนี้ 4, 8, 16, 32, 64 บิต และแสดงให้เห็นถึงเวลาหน่วง ในหน่วยของการวัดเวลาเป็น วินาที, นาที, ชั่วโมงและปี เวลาที่ได้สามารถคำนวณได้ดังสูตร

เวลารวมของการถอดรหัส = (เวลาที่ใช้ถอดรหัส X ขนาดของคีย์รหัส)/(60 X 60 X 24 X 365)

ตารางที่4.9 จะเป็นการแสดงค่าที่ได้จากการนำค่าในตารางที่4.6 ไปคำนวณหาค่าเวลาเมื่อเกิดการลักลอบถอดรหัส ของ DES แบบเก่าโดยใช้ Pentium4 2GHz (ตารางที่4.9) และ Pentium

120 MHz (ตารางที่4.10) ซึ่งขนาดของรหัสคีย์ที่ใช้มีดังนี้ 4, 8, 16, 32, 64บิต และแสดงให้เห็นถึงเวลาหน่วง ในหน่วยของการวัดเวลาเป็น วินาที, นาที, ชั่วโมงและปี

***จากตารางที่ได้นำเสนอ สังเกตว่า เวลาหน่วงของ DES แบบใหม่ไม่ว่าจะใช้ขนาดของรหัสคีย์เท่าใดก็ตามและไม่ว่าจะใช้ CPU รุ่นใดก็ตาม ก็สามารถที่จะหน่วงเวลาในการลักลอบถอดรหัสได้คงที่ และขนาดของคีย์ยิ่งมากเท่าใดเวลาในการหน่วงก็จะนานขึ้นตามไปด้วย แต่ถ้าDES แบบเก่านั้นเวลาในการลักลอบถอดรหัสจะสั้นลงเมื่อใช้ CPU ที่ความเร็วสูงขึ้น (สังเกตจากตาราง4.9 และตาราง 4.10)

<u>เพื่อให้เห็นภาพที่ชัดเจนของผลการเปรียบเทียบ จึงนำค่าจากตารางที่ 4.7 - 4.10 มาทำการเขียนกราฟ</u>

<u>เมื่อกำหนดให้</u>

TEST1 คือ ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบใหม่ด้วย Pentium4 2GHz (ข้อมูลจากตารางที่ 4.7)

TEST2 คือ ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบใหม่ด้วย Pentium 120 MHz (ข้อมูลจากตารางที่ 4.8)

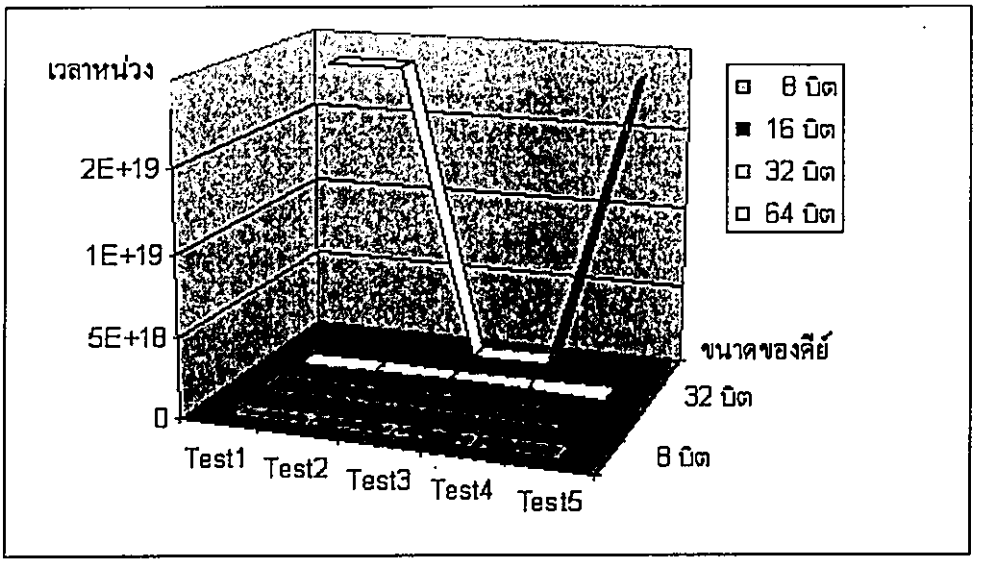
TEST3 คือ ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบเก่าด้วย Pentium4 2GHz (ข้อมูลจากตารางที่ 4.9)

TEST4 คือ ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบเก่าด้วย Pentium 120 MHz (ข้อมูลจากตารางที่ 4.10)

TEST5 คือ ค่าที่ได้จากการคำนวณในทางอุดมคติ (ข้อมูลจากตารางที่ 4.4 และ ตารางที่ 4.4)

ตารางที่ 4.11 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของ DES แบบเก่า ,แบบใหม่, และค่าคำนวณ (<u>ค่าในอุดมคติ</u>)

	Test1	Test2	Test3	Test4	Test5
8 บิต	255.94	255.98	0.047	0.71	256
16 บิต	65,521.99	65,530.99	0.76	11.46	65,536
32 บิต	4,294,049,461	4,294,639,590	194.64	2,935.35	4,294,967,296
64 บิต	1.84428E+19	1.84453E+19	12,756,052.86	192,371,585.2	18,446,744,073,709,500,000

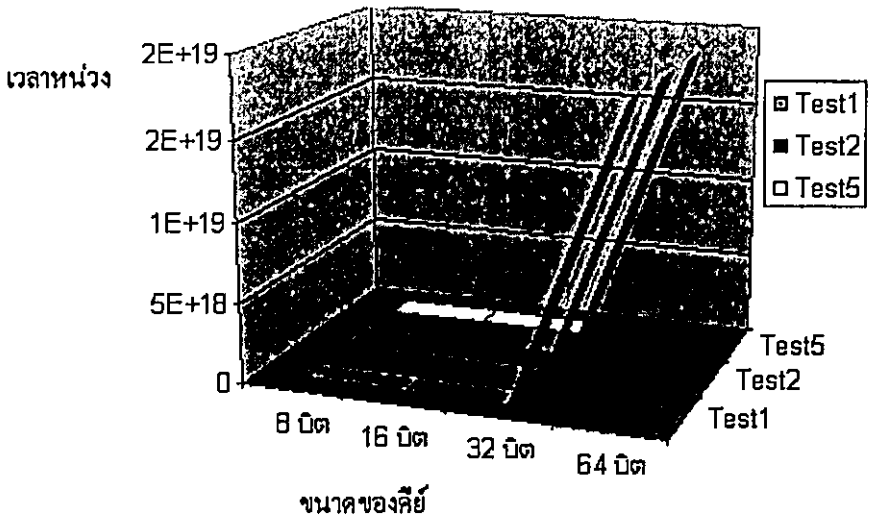


รูปที่4.79 กราฟเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัสของ DES แบบเก่า, แบบใหม่และค่าเวลาที่คำนวณขึ้น ซึ่งจะใช้จำนวนบิตของรหัส 8, 16, 32,64

ตารางที่ 4.12 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของDES แบบใหม่เปรียบเทียบกับ Pentium 120 MHz , Pentium4 2 GHz, และค่าคำนวณ (ค่าในอุดมคติ)

	Test1	Test2	Test5
8 บิต	255.94	255.98	256
16 บิต	65,521.99	65,530.99	65,536
32 บิต	4,294,049,461	4,294,639,590	4,294,967,296
64 บิต	1.84428E+19	1.84453E+19	18,446,744,073,709,500,000

จากตารางที่4.11 เป็นการเปรียบเทียบค่าหน่วงเวลาลักลอบถอดรหัสของ DES ทุกแบบ โดยค่าจากตารางจะนำมาสร้างกราฟได้ดังรูปที่ 4.79 จะสามารถสังเกตได้ว่า TEST1(ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบใหม่ด้วย Pentium4 2GHz), TEST2(ค่าที่ได้จากการคำนวณต่อจากผลการทดสอบ DES แบบใหม่ด้วย Pentium 120 MHz), TEST5(ค่าที่ได้จากการคำนวณในทางอุดมคติ) จะให้เวลาหน่วงที่คงที่ซึ่งจะให้ความปลอดภัยของข้อมูลสูงสุด

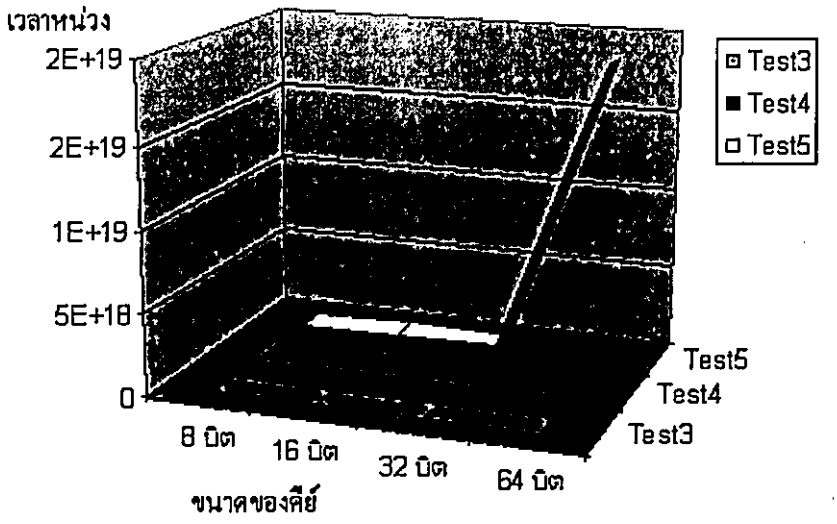


รูปที่4.80 กราฟเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัสของ DES แบบใหม่ ระหว่าง CPU Pentium 120 MHz, CPU Pentium4 2 GHz ค่าที่ได้จากการคำนวณ

ตารางที่ 4.13 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของDES แบบเก่า เปรียบเทียบกับ Pentium 120 MHz, Pentium4 2 GHz, และค่าคำนวณ (ค่าในอุดมคติ)

	Test3	Test4	Test5
8 บิต	0.047	0.71	256
16 บิต	0.76	11.46	65,536
32 บิต	194.64	2,935.35	4,294,967,296
64 บิต	12,756,052.86	192,371,585.2	18,446,744,073,709,500,000

จากตารางที่4.12 เป็นการเปรียบเทียบค่าหน่วงเวลาลักลอบถอดรหัสของ DES แบบใหม่ที่ใช้ Pentium 120 MHz, CPU Pentium4 2 GHz เปรียบเทียบกับ ค่าที่ได้จากการคำนวณในทางอุดมคติ ซึ่งค่าจากตาราง4.12 สามารถสร้างกราฟได้ดังรูปที่ 4.80 จากกราฟสังเกตได้ว่า DES แบบใหม่จะให้ค่าเวลาหน่วงที่ใกล้เคียงกับค่าหน่วงในอุดมคติมากที่สุด

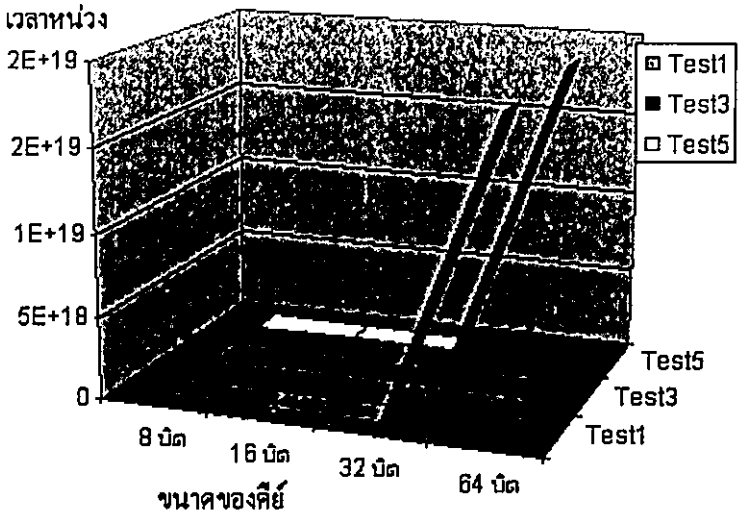


รูปที่4.81 กราฟเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัสของ DES แบบเก่า ระหว่าง CPU Pentium 120 MHz, CPU Pentium4 2 GHz ค่าที่ได้จากการคำนวณ

ตารางที่ 4.14 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของDES แบบเก่า, แบบใหม่เปรียบเทียบกับ ค่าคำนวณ(ค่าในอุดมคติ) โดยใช้ CPU Pentium4 2 GHz

	Test1	Test3	Test5
8 บิต	255.94	0.047	256
16 บิต	65,521.99	0.76	65,536
32 บิต	4,294,049,461	194.64	4,294,967,296
64 บิต	1.84428E+19	12,756,052.86	18,446,744,073,709,500,000

จากตารางที่4.13 เป็นการเปรียบเทียบค่าหน่วงเวลาลักลอบถอดรหัสของ DES แบบเก่าที่ใช้ Pentium 120 MHz, CPU Pentium4 2 GHz เปรียบเทียบกับ ค่าที่ได้จากการคำนวณในทางอุดมคติ ซึ่งค่าจากตาราง4.13 สามารถสร้างกราฟได้ดังรูปที่ 4.81 จากกราฟสังเกตได้ว่า DES แบบเก่าจะให้ค่าเวลาหน่วงที่สั้นลงเมื่อใช้ CPU ที่มีความเร็วสูงขึ้น

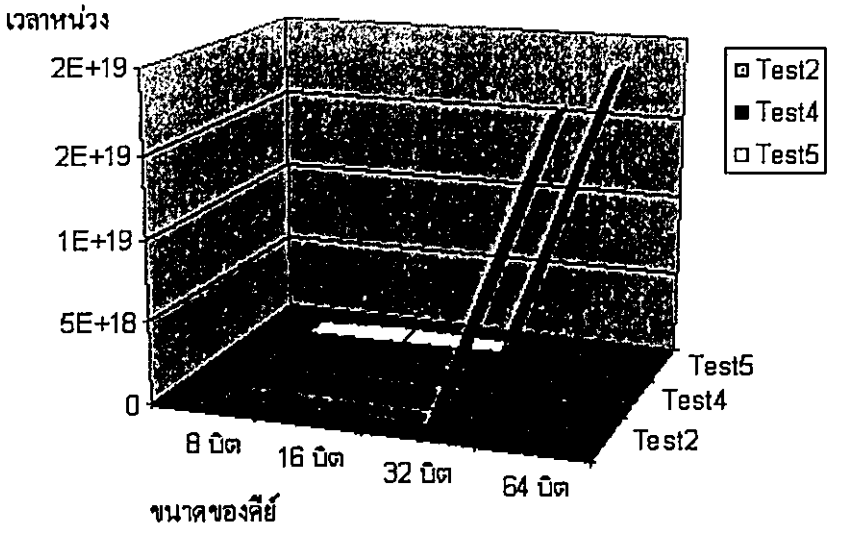


รูปที่4.82 กราฟเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัส ของDES แบบเก่า, แบบใหม่ และค่าที่ได้จากการคำนวณ โดยใช้ CPU Pentium4 2 GHz

ตารางที่ 4.15 แสดงการคำนวณเวลาหน่วงเมื่อเกิดการลักลอบถอดรหัส ของDES แบบเก่า,แบบใหม่ เปรียบเทียบกับ ค่าคำนวณ(ค่าในอุดมคติ) โดยใช้ CPU Pentium 120 MHz

	Test2	Test4	Test5
8 บิต	255.98	0.71	256
16 บิต	65,530.99	11.46	65,536
32 บิต	4,294,639,590	2,935.35	4,294,967,296
64 บิต	1.84453E+19	192,371,585.2	18,446,744,073,709,500,000

จากตารางที่4.14 เป็นการเปรียบเทียบค่าหน่วงเวลาลักลอบถอดรหัสของ DES แบบเก่า ใช้ CPU Pentium4 2 GHz , ของ DES แบบใหม่ ใช้ CPU Pentium4 2 GHz เปรียบเทียบกับ ค่าที่ได้จากการคำนวณในทางอุดมคติ ซึ่งค่าจากตาราง4.14 สามารถสร้างกราฟได้ดังรูปที่ 4.82 จากกราฟสังเกตได้ว่า DES แบบใหม่จะให้ค่าหน่วงในการลักลอบที่ใกล้เคียงกับค่าในทางอุดมคติ(มีความปลอดภัยมากสุด) แต่ DES แบบเก่าให้ค่าหน่วงที่ต่ำสุด(มีความปลอดภัยต่ำสุด) เมื่อใช้ที่ CPU มีความเร็วสูงขึ้น



รูปที่4.83 กราฟเปรียบเทียบเวลาที่ใช้ในการลักลอบถอดรหัส ของDES แบบเก่า, แบบใหม่ และค่าที่ได้จากการคำนวณ โดยใช้ CPU Pentium 120 MHz

จากตารางที่4.15 เป็นการเปรียบเทียบค่าหน่วงเวลาลักลอบถอดรหัสของ DES แบบเก่า ใช้ CPU Pentium120 MHz , ของ DES แบบใหม่ ใช้ CPU Pentium120 MHz เปรียบเทียบกับ ค่าที่ได้จากการคำนวณในทางอุดมคติ ซึ่งค่าจากตาราง4.15 สามารถสร้างกราฟได้ดังรูปที่ 4.82 จากกราฟสังเกตได้ว่า DES แบบใหม่จะให้ค่าหน่วงในการลักลอบที่ใกล้เคียงกับค่าในทางอุดมคติ(มีความปลอดภัยมากสุด) แต่ DES แบบเก่าให้ค่าหน่วงที่ต่ำสุด(มีความปลอดภัยต่ำสุด) ถึงแม้ CPU มีความเร็วต่ำลงมาเหลือเพียง Pentium120 MHz

ดังนั้นข้อมูลที่นำเสนอมาของค่าที่ได้จากการทดสอบ, ผลการคำนวณค่าในทางอุดมคติและค่าที่คำนวณต่อจากผลที่ได้จากการทดสอบ จึงสามารถที่จะสรุปผลการทดลองได้ว่า CPU ที่ใช้ในการลักลอบถอดรหัสจะมีความเร็วสักเพียงใดก็ตามก็ไม่สามารถถอดรหัสได้ไวขึ้น เมื่อใช้ DES แบบใหม่ที่ได้นำเสนอในงานวิจัยนี้ โดยสังเกตได้จากเวลาที่ใช้ในการเข้ารหัสและถอดรหัสของ DES แบบใหม่จะใช้เวลาที่

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 กล่าวนำ

วิทยานิพนธ์ฉบับนี้ทำการศึกษาและวิจัยเกี่ยวกับการปรับปรุงวิธีการเข้ารหัสข้อมูลชนิด DES โดยวิทยานิพนธ์ฉบับนี้ ได้นำเสนอถึงความน่าจะเป็นไปได้ของการปรับปรุงวิธีการเข้ารหัสข้อมูลชนิด DES ให้สามารถเข้ารหัสได้ไวขึ้นจากเดิม โดยลดขั้นของ DES บางขั้นตอนลง และเพิ่มฟังก์ชั่นพิเศษลงไปแทนที่เพื่อให้ DES ยังคงสามารถที่จะรักษาความปลอดภัยได้ดีขึ้น จากผลการทดลองปรับปรุงวิธีการเข้ารหัสและถอดรหัสแล้วทำการทดสอบ อัลกอริธึมให้ผลที่ได้นั้น อยู่ในระดับที่น่าพอใจ เพราะผลที่ได้นั้นสามารถปรับปรุงให้สามารถทำงานได้รวดเร็วขึ้นจริงและสามารถที่จะปกปิดข้อมูลได้เช่นเดียวกันกับ DES แบบเก่าซึ่งจากผลการทดสอบที่ได้จากการเข้ารหัสข้อความแบบต่างๆก็สามารถปกปิดข้อความได้จริงจนไม่สามารถที่จะเดาคำที่แท้จริงได้ และในตอนสุดท้ายได้ทำการทดสอบความสามารถในการหน่วงเวลาเพื่อป้องกันการลักลอบถอดรหัสข้อความ ปรากฏผลที่ได้ว่า วิธีการเข้ารหัสแบบใหม่ที่ได้นำเสนอสามารถที่จะหน่วงเวลาได้จริง โดยทำการทดสอบการลักลอบถอดรหัสจะใช้ เครื่องคอมพิวเตอร์ 2 ชนิดที่ใช้หน่วยประมวลผลกลาง (CPU) ที่มีความเร็วในการประมวลผลที่แตกต่างกันคือ รุ่นที่ใช้ หน่วยประมวลผลกลาง (CPU) ชนิดPentium ใช้ความถี่ของสัญญาณนาฬิกา 120 MHz และรุ่นที่ใช้หน่วยประมวลผลกลาง(CPU) ชนิด Pentium 4 ใช้ความถี่ของสัญญาณนาฬิกา 2 GHz ผลที่ได้จากการทดลองปรากฏว่า DES แบบใหม่สามารถที่จะหน่วงเวลาได้คงที่ เมื่อเกิดการลักลอบถอดรหัส ซึ่งสามารถที่สรุปได้ว่า ถึงแม้ว่าตัว Hardware ที่ใช้สำหรับเพื่อการลักลอบถอดรหัสจะมีความเร็วในการทำงานหรือประมวลผลเร็วสักเพียงใด ก็ไม่สามารถถอดรหัสข้อมูลของ DES ที่ใช้อัลกอริธึมแบบใหม่ได้ไวขึ้น (สามารถหน่วงเวลาในการลักลอบถอดรหัสได้คงที่) และ Function พิเศษที่เพิ่มเขาไปในอัลกอริธึม DES แบบใหม่เพื่อใช้ปรับปรุงปรุงนั้นก็ไม่ทำให้ผลการเข้ารหัสและถอดรหัสผิดเพี้ยนไปจากเดิม และ Function พิเศษ ที่เพิ่มเข้าไปนั้นก็ไม่ทำให้ความเร็วในการเข้ารหัสและถอดรหัสของ DES แบบใหม่นั้นลดลงด้วยเช่นกัน

5.2 สรุปผลการทดลอง

จากผลการทดลองที่ได้จากงานวิจัยทำให้ทราบว่าการเข้ารหัสข้อมูลชนิด DES แบบเก่าและแบบใหม่สามารถปกปิดข้อมูลได้เช่นเดียวกัน แต่เข้ารหัสข้อมูลชนิด DES แบบเก่าสามารถถูกลักลอบถอดรหัสได้ง่ายกว่า (ใช้เวลาในการถอดรหัสน้อยลง) สังเกตได้จากตารางที่ 4.2 และ ตาราง

ที่ 4.3 เมื่ออุปกรณ์ที่ใช้ในการลักลอบถอดรหัสมีความเร็วสูงขึ้นในขณะที่การเข้ารหัสข้อมูลชนิด DES แบบใหม่จะต้องใช้เวลาในการลักลอบถอดรหัสที่คงที่ ถึงแม้ว่าอุปกรณ์ที่ใช้ในการลักลอบถอดรหัสจะมีความเร็วสูงขึ้นเพียงใดก็ตามสังเกตได้จากตารางที่ 4.2 และ ตารางที่ 4.3 ผลจากการทดลองจะได้ ตารางที่ 5.1 สรุปผลการเปรียบเทียบอัลกอริทึมการเข้ารหัสและถอดรหัส ของ DES แบบเก่าและแบบใหม่ ผลจากการทดลองจะสามารถสรุปผลการเปรียบเทียบอัลกอริธึมการเข้ารหัสและถอดรหัสแบบเก่าและแบบใหม่ดังตารางที่ 5.1

ตารางที่ 5.1 แสดงการเปรียบเทียบคุณสมบัติของการเข้ารหัสแบบ DES และ The Development of Data Encryption Standard

	DES (Data Encryption Standard)	The development of Data Encryption Standard
1.ความเร็วในการเข้ารหัส	ขึ้นอยู่กับ Hardware	เร็วขึ้น
2.ความเร็วในการถอดรหัส	ขึ้นอยู่กับ Hardware	เร็วขึ้น
3.การตรวจสอบการลักลอบถอดรหัส	ไม่มี	มี
4.หน่วงเวลาระบบเมื่อเกิดการลักลอบถอดรหัส	ไม่มี	มี
5.จำนวนรอบในการเข้ารหัส	16 ครั้ง	8 ครั้ง
6.สามารถนำไปสร้างบนฮาร์ด-แวร์ได้ง่าย	ยาก	ง่าย
7.เวลาที่ใช้ในการลักลอบถอดรหัส	ขึ้นอยู่กับ Hardware	Cycle Time of DES X 2^{112} (คงที่)
8.จำนวนบิตกุญแจที่ใช้ถอดรหัส	56 บิต	112 บิต
9.ความปลอดภัยของข้อมูล	ขึ้นอยู่กับ Hardware	สูงกว่าเดิม
10.การเลื่อนบิตของกุญแจ	Shift Left	Shift Left , Shift Right

5.3 ปัญหาที่พบในงานวิจัยและการแก้ปัญหา

ในการทดลองวิจัยนี้มีสิ่งที่เป็นข้อจำกัดในงานวิจัยหลายอย่างทำให้เกิดปัญหาในทดลองซึ่งสามารถสรุปได้ดังต่อไปนี้

1. ด้านเครื่องมือที่ใช้สำหรับทดสอบความเร็วในการเข้ารหัสและถอดรหัสอัลกอริธึมจำเป็นที่จะต้องเขียนโปรแกรมเพิ่มเข้าไปในอัลกอริธึมเพื่อให้ส่งสัญญาณอินเตอร์เฟสกับอุปกรณ์ภายนอกที่ทำหน้าที่สำหรับตรวจจับเวลาที่ใช้ในการเข้ารหัสและถอดรหัส โปรแกรมที่เขียนเพิ่มเข้าไปอาจจะทำให้คาบเวลาที่ได้จากการตรวจจับโดยอุปกรณ์ภายนอก เกิดข้อผิดพลาดของคาบเวลาที่วัดได้ จำเป็นที่ต้องใช้เครื่องมือที่ออกแบบมาสำหรับทดสอบช่วงเวลาในการทำงานของหน่วยประมวลผลกลาง (CPU)โดยตรง

2. ความแม่นยำของการเข้ารหัสและถอดรหัสข้อมูลของอัลกอริธึมใหม่ ในวิทยานิพนธ์ได้ทำการทดสอบการเข้ารหัสและถอดรหัสข้อมูลเพียงบางค่าเท่านั้น เพื่อแสดงให้เห็นว่าอัลกอริธึมที่นำเสนอสามารถเข้ารหัสข้อมูล และสามารถถอดรหัสได้ข้อมูลเหมือนเดิมทุกประการ เพื่อให้อัลกอริธึมมีความน่าเชื่อถือได้จำเป็นที่จะต้องทดสอบให้หลายๆค่าเพื่อจะได้ทราบจุดที่เกิดความบกพร่อง(Bug) ของอัลกอริธึมใหม่

3. ในการทดสอบ ความสามารถในการหน่วงเวลา เมื่อเกิดการลักลอบถอดรหัสไม่สามารถที่ทดสอบได้กับทุกๆ key เนื่องจาก key ที่มีระยะห่างมากๆ ต้องใช้เวลาในการทดสอบการลักลอบถอดรหัสนานมาก ในงานวิจัยจะนำเสนอโดยใช้ key ที่มีระยะห่างสั้นๆเท่านั้น เพื่อแสดงให้เห็นว่าอัลกอริธึมที่นำเสนอสามารถป้องกันการลักลอบการถอดรหัสได้จริง *ในวิทยานิพนธ์จึงนำเสนอขนาดของรหัสที่ใช้ในการทดสอบเพียง 16 บิตเท่านั้น ดังนั้นส่วนการทดลองขนาดของรหัส 64 บิต ซึ่งจะเปรียบเทียบกับ DES แบบมาตรฐานนั้นจะใช้วิธีการคำนวณแทนเพื่อให้เห็นถึงความสามารถในการหน่วงเวลา <u>เนื่องจากเวลาที่ใช้ในการทดสอบจะใช้เวลาถึง 584,942,417,355.07 ปี ซึ่งเป็นไปไม่ได้ที่จะทดลองให้เห็นค่าตัวเลขหน่วงเวลา ที่แท้จริงได้</u>*

5.4 ข้อเสนอแนะในการพัฒนา

ผลการทดสอบที่ได้นั้นเป็นเพียงผลการทดลองจากการจำลองการทำงานของ อัลกอริธึมบนเครื่องคอมพิวเตอร์ และใช้ค่าบางส่วนที่ได้มาคำนวณค่าเพื่อนำไปเปรียบเทียบกับหลักการที่มีอยู่เดิม การที่จะนำอัลกอริธึมไปพัฒนาต่อ หรือการที่จะนำนำอัลกอริธึมมาประยุกต์ใช้งานบน Hardware จำเป็นที่จะต้องอาศัยการทดสอบที่มีความละเอียดมากกว่านี้ เพื่อให้อัลกอริธึมมีความน่าเชื่อถือได้มากขึ้นและเพื่อหาจุดที่เกิดความบกพร่อง (Bug) ที่อาจจะเกิดขึ้นได้จากตัวโปรแกรมที่สำหรับเขียนโปรแกรมทดสอบหรือเกิดเนื่องจากตัวอักลริธึมที่ปรับปรุงขึ้นมาใหม่

เอกสารอ้างอิง

[1] Douglas R.Stinson. " Cryptography Theory and Practice. " CRC Press, 1995. pp.70-113.

[2] Man Young Rhee. " Cryptography and Secure Communications. " McGraw-Hill, 1994. pp.45-101.

[3] C.P. Pfleeger. "Security in Computing. " Englewood cliffs, NJ: Prentice-Hall, 1989. pp. 22-74.

[4] Brian W.Kernighan and Dennis M.Ritchie. **The C Programming Language.** Prentice-Hall, Inc. 1983.

[5] Herbert schildt. **Born to Code in C.** Osborne McGraw-Hill. 1989.

[6] Herbert schildt. **C : Power User's Guide.** Osborne McGraw-Hill. 1989.

[7] Robert C.Hutchison and Steven B. Just. **Program Using The C Language.** McGraw-Hill, Inc. 1989.

[8] Wallace Wang. **Visual BASIC programming for dummies.** IDG Book. 1995.

[9] Zane Thomas, Robert Arnson, Mitchell Waite. **Visual Basic how-to.** Waite Group Press. 1993.

[10] Richard Mansfied and Evangelos Petroutsos. **Visual Basic power toolkit : cutting-edge tools and techniques for programmers.** Ventana Press. 1995.

[11] Waiyawut Sanayha. "Data Encryption and Decryption by cipher Feedback Mode of the DES Standard.", Songkla University. 1997.

[12] Yongyot Rataseree and Punya Thitimajshima. "The Development Of Data Encryption Standard." International Conferenceon on Computers And Devices For Communication (CODEC-2004), Kolkata (India), January 2004.

[13] Chaiyong Chaisingthong. " Data Encryption Algorithm for Military Purposes." King Mongkut's Institute of Technology Ladkrabang. 1993.

[14] Schneier, Bruce. **Applied Cryptography:Protocals, Algorithms, and Source Code in C.** John Willey & Sons. 1996 .

เอกสารอ้างอิง (ต่อ)

[15] ฉัททวุฒิ พืชพล และ พิชิต สันติกุลานนท์. **คู่มือเรียน Visual Basic 6.** กรุงเทพมหานคร : Provision. 2542.

[16] เขมะฑัต วิภาตะวนิช. "ลายเซ็นดิจิตอล." กรุงเทพมหานคร : ไมโครคอมพิวเตอร์. ฉบับที่ 118, พฤษภาคม 2538. หน้า 209-219.

[17] เลอสรร ธนสุกาญจน์. "เรียนรหัสด้วยคอมพิวเตอร์ ตอน เครื่องมือของนักแกะรหัส: การแจกแจงความถี่ของอักษรเดี่ยว." กรุงเทพมหานคร : คอมพิวเตอร์, ฉบับที่ 70, 2530. หน้า 38-49.

[18] เลอสรร ธนสุกาญจน์. "เรียนรหัสด้วยคอมพิวเตอร์ ตอน รหัสมาตรฐาน." กรุงเทพมหานคร : คอมพิวเตอร์, ฉบับที่ 73, 2530. หน้า 34-41.

ภาคผนวก

ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] Yongyot Rataseree and Punya Thitimajshima. "The Development Of Data Encryption Standard." International Conferenceon on Computers And Devices For Communication (CODEC-2004), Kolkata (India). 1-3 January 2004.

[2] Yongyot Rataseree, Punya Thitimajshima and Yuttapong Rangsanseri, "The Improvement of Data Encryption Standard." International Conference on Advanced Communication Technology (ICACT 2004), Korea. 11-13 February 2004.

[3] Yongyot Rataseree and Punya Thitimajshima. " Implementation of RISC Microcontroller for Speech Scramble Using Block Code." International Conference on Advanced Communication Technology (ICACT 2004), Korea. 11-13 February 2004.

ประวัติผู้เขียน

ผู้เสนอผลงาน นายยงยศ รตะเสรี เกิดเมื่อวันที่ 4 มิถุนายน 2516 ที่จังหวัดภูเก็ต

ที่อยู่ปัจจุบัน 128/4 ถ.ระนอง ต.ตลาดเหนือ อ.เมือง จ.ภูเก็ต 83000 เบอร์โทรศัพท์ 01-6934788

ประวัติการทำงาน

พ.ศ. 2537 รับราชการในตำแหน่งครู2 ระดับ2 ประจำแผนกช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคฉะเชิงเทรา (วท.ฉช.)

พ.ศ. 2539 รับราชการในตำแหน่งอาจารย์1 ระดับ3 ประจำแผนกช่างอิเล็กทรอนิกส์ วท.ฉช.

พ.ศ. 2540 รับราชการในตำแหน่งอาจารย์1 ระดับ3 และเจ้าหน้าที่งานวางแผนการศึกษาและพัฒนา วท.ฉช.

พ.ศ. 2541 รับราชการในตำแหน่งอาจารย์1 ระดับ4 ประจำแผนกช่างอิเล็กทรอนิกส์ วท.ฉช.

พ.ศ. 2542 รับราชการในตำแหน่งอาจารย์1 ระดับ4 ประจำแผนกช่างอิเล็กทรอนิกส์ วิทยาลัย-เทคนิคภูเก็ต (วท.ภก.)

พ.ศ. 2544 รับราชการในตำแหน่งอาจารย์1 ระดับ4 และผู้ช่วยหัวหน้างานพัสดุกลาง วท.ภก.

พ.ศ. 2544 วิทยากรบรรยายพิเศษ สถาบันราชภัฏภูเก็ต คณะวิทยาศาสตร์ โปรแกรมวิชาวิทยาการคอมพิวเตอร์

พ.ศ. 2545 รับราชการในตำแหน่งอาจารย์1 ระดับ5 ประจำแผนกช่างอิเล็กทรอนิกส์ วท.ภก.

พ.ศ. 2547 รับราชการในตำแหน่งอาจารย์2 ระดับ6 วิทยาลัยเทคนิคภูเก็ต,หัวหน้าศูนย์ฝึกอบรมสารสนเทศ (กระทรวงศึกษาธิการ), หัวหน้าศูนย์อินเตอร์เน็ต (อบจ.) และหัวหน้าศูนย์อินเตอร์เน็ตสมาคมผู้ปกครองและครูวิทยาลัยเทคนิคภูเก็ต

พ.ศ. 2547 วิทยากรบรรยายพิเศษ สถาบันราชภัฏภูเก็ต คณะวิทยาศาสตร์ โปรแกรมฟิสิกส์

ประวัติการศึกษา

พ.ศ. 2537 ประกาศนียบัตรวิชาชีพชั้นสูง วิทยาลัยเทคนิคภูเก็ต

พ.ศ. 2539 ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ (คอบ.), สจล.

พ.ศ. 2547 วิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมไฟฟ้า (วศ.ม.), สจล.

ประสบการณ์หรือความเชี่ยวชาญพิเศษ

1.วิธีการเข้ารหัสข้อมูลแบบมาตรฐาน(Data Encryption)

2.การประยุกต์ใช้ไมโครคอนโทรลเลอร์,ไมโครโปรเซสเซอร์

3.การออกแบบวงจรดิจิตอลชั้นสูง