การเข้ารหัส-ถอดรหัส เพื่อความปลอดภัยของข้อมูลบนอินเตอร์เน็ต

Encryption-Decryption for Data Security on the Internet

โดย

น.ส. วรรวิภา ท่าพระนา

รหัส 38626044

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2540

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การเข้ารหัส-ถอดรหัส เพื่อความปลอดภัยของข้อมูลบนอินเตอร์เน็ต
นักศึกษา	น.ส. วรรวิภา ท่าพระนา
อาจารย์ที่ปรึกษา	ดร. นพพร โชติกกำธร
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
พ.ศ.	2540

บทคัดย่อ

โครงการนี้ศึกษาถึงการเข้ารหัส-ถอดรหัสข้อมูลบนเครือข่ายอินตอร์เน็ต เพื่อความปลอดภัยของข้อมูล โดยใช้อัลกอริทึม DES(Data Encryption Standard) และ RSA(Rivest,Shamir and Adleman) ซึ่งทำการพัฒนาโปรแกรมการเข้ารหัส-ถอดรหัสด้วยอัลกอริทึมทั้งสองที่กล่าวมาข้างต้นด้วยภาษาจาวา โปรแกรมช่วยพัฒนาคือ Symantec Cafe โปรแกรมที่พัฒนาขึ้นนี้เพื่อวัตถุประสงค์หลักคือ นำไปใช้สร้างเป็นส่วนประกอบในการสร้างวปเพจ แสดงถึงขั้นตอนการทำงานของอัลกอริทึมทั้งสอง นอกจากนี้โดยการใช้งานโปรแกรมดังกล่าวทำให้สามารถทำการเข้ารหัส-ถอดรหัสลงไฟล์ข้อมูลได้ พร้อมทั้งได้ทำการทดสอบความเร็วในการเข้ารหัส- ถอดรหัสของอัลกอริทึมทั้งสองแบบ โดยพิจารณาความสัมพันธ์กับปัจจัยด้าน ขนาดของกุญแจ ขนาดของข้อมูล และประเภทของอัลกอริทึม

Title	**Encryption-Decryption for Data Security on the Internet**
Student	**Ms. Wanwipa Thaprana**
Advisor	**Dr. Nopporn Chotikakamthorn**
level of Study	**Master of Science in Information Technology**
Major	**Information Science**
Year	**1997**

ABSTRACT

This project studied data encryption and decryption method for providing data security on the Internet. The algorithm considered in this report are DES and RSA algorithm. Programs written in JAVA were developed for both algorithms by using the development tool ' Symantec Cafe '. The programs developed have been used to construct a Web page which describes the steps of operations of both encryption-decryption .In addition,speed of both algorithms were tested by considered in relative of key size, data size and type of algorithms.

กิตติกรรมประกาศ

โครงงานนี้ได้รับการส่งเสริมและสนับสนุนให้สำเร็จไปด้วยดีจากบุคคลหลายฝ่ายดังนั้นจึงใคร่ขอขอบพระคุณ บุคคลต่อไปนี้

1. ครอบครัว อันประกอบไปด้วย มารดา เป็นบุคคลที่มีพระคุณสูงสุดเป็นผู้ให้กำเนิดและเลี้ยงดูมอบสิ่งที่ดีงามในการประพฤติปฏิบัติตน เป็นแบบอย่างที่ดีในการดำเนินชีวิต ตลอดจนเป็นกำลังใจสำคัญในการศึกษาเล่าเรียนให้ประสบผลสำเร็จ พี่ชายและพี่สาวเป็นผู้ที่ส่งเสริมและสนับสนุนให้มีโอกาสทางการศึกษา

2. คณาจารย์ เป็นกลุ่มบุคคลที่มีพระคุณมาก ได้ให้วิชาความรู้แก่ศิษย์โดยไม่เห็นแก่ความเหนื่อยยาก

3. อาจารย์นพพร โชติกกำธร เป็นอาจารย์ที่ปรึกษาโครงงานที่กรุณาให้คำปรึกษาแนะนำการทำโครงงาน และให้แนวทางในการแก้ไขปัญหาต่างๆที่เกิดขึ้นในการปฏิบัติโครงงาน

4. เพื่อนๆ ร่วมชั้นเรียนทุกท่านที่ช่วยเหลือสนับสนุนเป็นกำลังในด้านต่างๆ ทำให้การเรียนในหลักสูตรนี้เป็นไปอย่างดี

วรรวิภา ท่าพระนา

สารบัญ

สารบัญ(ต่อ)

หน้า

บทที่

สารบัญภาพ

หน้า

ภาพที่

สารบัญตาราง

หน้า

ตารางที่

บทที่1
บทนำ

1. บทนำ

ในปัจจุบันมีการใช้งานระบบเครือข่ายอย่างแพร่หลาย ซึ่งข้อมูลข่าวสารเหล่านี้มีความสำคัญมากในการดำเนินงานขององค์กรทั้งของรัฐบาล และธุรกิจเอกชน ข้อมูลได้กลายเป็นปัจจัยสำคัญ ในการตัดสินใจเพื่อการบริหาร และคอมพิวเตอร์ถูกนำมาใช้ในการประมวลผล เพื่อให้ได้ข้อมูลที่ถูกต้องรวดเร็ว และทันสมัยอยู่เสมอ แต่ ปัญหา สำคัญก็คือ การป้องกันข้อมูลเหล่านี้ให้ปลอดภัย และถูกต้องอยู่ตลอดเวลา เนื่องจากข้อมูลอาจถูกเข้าถึงจากผู้ที่ไม่มีสิทธิ์ ซึ่งอาจทำให้เกิดผลเสียต่อเจ้าของข้อมูลได้ ถึงแม้ระบบคอมพิวเตอร์ในปัจจุบันจะมีระบบรักษาความปลอดภัยในขั้นพื้นฐานอยู่แล้วแต่ในกรณีที่ข้อมูลมีความสำคัญมาก ๆ ก็ จำเป็น ที่จะต้องมีมาตรการรักษาความปลอดภัย และการป้องกันการเข้าถึงข้อมูลของผู้ไม่มีสิทธิ์ ซึ่งกรณีนี้จะเกี่ยวข้องกับทั้งผู้ส่งและผู้รับจาก เหตุผลนี้ถ้ามีผู้ไม่ประสงค์ดีหรือผู้ที่ไม่เกี่ยวข้องสามารถเข้าถึงข้อมูลได้อย่างง่าย แต่การแก้ปัญหาด้วยวิธีนี้ก็จะมีผลกระทบกับการทำงานปกติ ดังนั้นจึงมีความคิดว่าถ้าให้ การทำงานสามารถรับ-ส่งข้อมูลกันได้อย่างปกติ แต่ให้การรับข้อมูลของผู้ที่ไม่มีสิทธิ์จะไม่สามารถเข้าใจข้อมูลทีรับมาได้แทน ในการใช้งานระบบเครือข่ายข้อมูลที่ส่งไปมาในระบบจากที่หนึ่งไปยังอีกที่หนึ่งถ้าไม่ต้องการให้ผู้อื่นที่ไม่ใช่ผู้รับอ่านได้ ก็ต้องทำการเข้ารหัสข้อมูลนั้นก่อนส่งไป การเข้ารหัสเป็นการแปลงข้อมูลให้อยู่ในรูปแบบที่ไม่สามารถอ่านได้ตามปกติ วิธีที่ใช้แปลงข้อมูลนั้นมีอยู่หลายวิธี มีทั้งแบบง่ายและแบบยากแต่จะแน่ใจได้อย่างไรว่าไม่สามารถถูกถอดรหัสได้ ถ้าไม่มีกุญแจที่ถูกต้อง

เนื่องจากการพัฒนาด้านคอมพิวเตอร์ เริ่มเข้ามามีความสำคัญต่อการใช้งานในสังคมปัจจุบัน การเก็บข้อมูลในคอมพิวเตอร์ หรือการติดต่อสื่อสารในระบบคอมพิวเตอร์ เป็นเรื่องสำคัญที่บางครั้งองค์กรนั้น ๆ จะต้องปกปิดเป็นความลับ วิธีหนึ่งที่ใช้กันมานาน และยังเป็นที่นิยมใช้กันอยู่ทุกวันนี้ก็คือการเข้ารหัสข้อมูล (Cryptography และ Data Encryption) เพื่อให้ข้อมูลเปลี่ยนไปจากเดิมจนไม่สามารถเข้าใจได้ ยกเว้นผู้ที่รู้รหัสหรือ key word เท่านั้นจึงสามารถถอดข้อความกลับมาเป็นรูปแบบเดิมได้

การเข้ารหัส (Encryption) เป็นกรรมวิธีการเข้ารหัสข้อมูลข่าวสารทำให้ความหมายของข่าวสารเดิมแปรเปลี่ยนไป ส่วนการถอดรหัส (Decryption) เป็นกรรมวิธีที่ตรงกันข้ามที่ใช้แปลง

ข่าวสารจากการเข้ารหัสให้กลับไปเป็นข้อมูลเดิม ในระบบที่ประกอบด้วยทั้งส่วนการเข้ารหัสและการถอดรหัสจะเรียกว่า ระบบการสร้างรหัสลับ (Cryptosystem) ซึ่งหลักการเข้ารหัสที่จะกล่าวถึงนี้ คือ RSA และ DES ทั้ง 2 อัลกอริทึมเป็นวิธีการเข้ารหัสที่น่าสนใจและใช้กันอย่างกว้างขวาง

1.1 วัตถุประสงค์ของโครงงานการพัฒนาระบบงาน

1.เพื่อศึกษาวิธีการเข้ารหัส - ถอดรหัส บนระบบเครือข่าย

2.. ทำการเปรียบเทียบเวลาที่ใช้ในการเข้ารหัส ของ อาร์ เอส เอ และ เดส อัลกอริทึม

3. พัฒนาโปรแกรมการเข้ารหัส - ถอดรหัส ด้วยโปรแกรมภาษาจาวา

1.2 ทฤษฎีและหลักการที่เกี่ยวข้อง

1.2.1 Secret-key System หรือ Symmetric Cryptosystem ใช้กุญแจเพียงตัวเดียวในการเข้ารหัสและ ถอดรหัส เช่น DES อัลกอริทึม , DES ได้รับการประกาศให้เป็นมาตรฐานของประเทศอเมริกาเพื่อใช้เป็นสาธารณะทั่วไปและองค์การมาตรฐานระหว่างประเทศ (ISO) ได้ให้การยอมรับเป็นมาตรฐานสากล

ระบบมาตรฐาน DES เป็นวิธีการสลับสับเปลี่ยนและการแทนค่ามาทำงานร่วมกันถึง 16 วงรอบ โดยข้อมูลปกติเดิมจะผ่านการเข้ารหัสเป็นบล็อกของจำนวน 64 บิทและสามารถกำหนดกุญแจยาวเป็นตัวเลข 56 บิทที่ผู้ใช้สามารถเปลี่ยนแปลงตามความต้องการ การเข้ารหัส-ถอดรหัสจะใช้เลขฐานสอง กุญแจประกอบด้วยเลขฐานสอง 64 บิท ใช้ในขบวนการเข้าและถอดรหัส อีก 8 บิท ใช้ในการตรวจสอบข้อผิดพลาดของ 56 บิท (Error Detection)

DES แบ่งการทำงานเป็น 2 ส่วน คือ

1. ส่วนสร้างกุญแจ มีความยาว 56 บิท
2. ส่วนเข้ารหัสและถอดรหัส มีความยาว 64 บิท

การถอดรหัส กระทำได้จะต้องรู้ถึง องค์ประกอบ 2 อย่าง คือ

1. อัลกอริทึมที่นำมา ใช้ในการเข้ารหัส-ถอดรหัส
2. กุญ แจ

ถ้ารู้ อัลกอริทึม แต่ไม่รู้กุญแจต้องใช้เวลานานเพื่อทำการทดลองทุก ๆ กุญแจที่เป็นไปได้ กุญแจใช้ 56 บิท = 2^{56} ประมาณ 7 หมื่นล้านล้าน กุญแจที่เป็นไปได้แต่ถ้าไม่รู้เทคนิคการเข้ารหัสจะไม่สามารถถอดรหัสได้เลย

1.2.2 Public-key System หรือ Asymmetric Cryptosystem ใช้คู่ของกุญแจ คือกุญแจใน

การเข้ารหัสและ กุญแจในการถอดรหัส เช่น **RSA** อัลกอริทึม ในการเข้ารหัสด้วย อาร์ เอส เอ อัลกอริทึม (RSA Algorithm) ข้อความจะถูกแทนด้วยตัวเลขจำนวนเต็ม ซึ่งเปลี่ยนจาก ข้อความ เป็นรูปแบบตัวเลข (อาจใช้ รหัสแอสกี แทนตัวอักษร) ข้อมูลจากผู้ส่งจะถูกเข้ารหัสด้วย กุญ แจสาธารณะ และจะถูกถอดรหัสได้เพียงการใช้ กุญแจส่วนตัว ที่เป็นคู่ของมันเท่านั้น ซึ่งการเข้ารหัสแบบนี้สามารถเปิดเผยกุญแจที่ใช้ในการเข้ารหัสได้ เนื่องจากกุญแจการเข้ารหัสและถอดรหัสจะเป็นคนละตัวกัน กุญแจการเข้ารหัสไม่สามารถนำมาใช้ในการถอดรหัสได้ ดังนั้นถ้ารู้กุญแจสำหรับเข้ารหัสก็ไม่มีความหมาย ผู้ใช้ (user) จึงสามารถที่จะกระจายกุญ แจสาธารณะ ไปได้อย่างอิสระ เป็นที่ไหนก็ได้เพราะถ้าไม่มี กุญแจส่วนตัว ก็ไ ม่ สามารถจะถอดรหัส ได้ แต่เราควรจะเก็บ กุญแจส่วนตัวในที่ ๆ ปลอดภัยที่สุด

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เรียนรู้และเข้าใจการทำงานของ DES และ RSA อย่างละเอียด
2. สามารถทำนายเวลาที่ใช้ในการเข้ารหัสได้ เมื่อทราบขนาดของข้อมูลที่ใช้ในการเข้ารหัส
3. พัฒนาโปรแกรมการเข้ารหัสด้วยภาษา JAVA เพื่อการส่งข้อมูลให้ปลอดภัยผ่านระบบเครือข่าย

1.4 รายละเอียดของแต่บท

บทที่ 2 กล่าวถึงความเป็นมาแนวคิดของการเข้ารหัสรวมทั้งหลักเกณฑ์ที่ได้รับการยอมรับว่าเป็นการเข้ารหัสที่ดี ซึ่งนำไปสู่ระบบมาตราฐานของประเทศอเมริกา ในบทนี้ยังกล่าวถึงรูปแบบการใช้และคุณสมบัติของการเข้ารหัส - ถอดรหัส ยังแสดงการสร้างรหัสลับด้วยคอมพิวเตอร์แบบง่ายและทฤษฎีของ อัลกอริทึม DES และ RSA อย่างละเอียด

บทที่ 3 กล่าวถึงลักษณะของภาษาจาวา ระบบรักษาความปลอดภัยของ ภาษาจาวา รวมทั้งการทำงานของจาวา

บทที่ 4 กล่าวถึงฟังก์ชันการคำนวณหากุญแจสาธารณะ เพื่อใช้ในการเข้ารหัส ฟังก์ชันการคำนวณหากุญแจส่วนตัวเพื่อใช้ในการถอดรหัส และ ฟังก์ชันการคำนวณการถอดรหัส ของอัลกอริทึม RSA แสดงถึงผลการทดลองการเปรียบเทียบเวลาที่ใช้ในการทำงานของการเข้ารหัส ถอดรหัสด้วยอัลกอริทึม DES และ RSA เมื่อเปลี่ยนจำนวนข้อมูลเพิ่มมากขึ้น ตลอดจนการแสดงผลของการทดลองการเข้ารหัส-ถอดรหัสด้วยอัลกอริทึม RSA เมื่อเปลี่ยนขนาดของกุญแจ

บทที่ 5 สรุปถึงโครงการพัฒนาระบบงานที่ใช้ในเข้ารหัส ถอดรหัส ด้วยอัลกอริทึม DES และ RSA และการประยุกต์ใช้อัลกอริทึมทั้งสองภายในหน่วยงานต่าง ๆ กล่าวถึงอุปสรรคที่ได้จาก

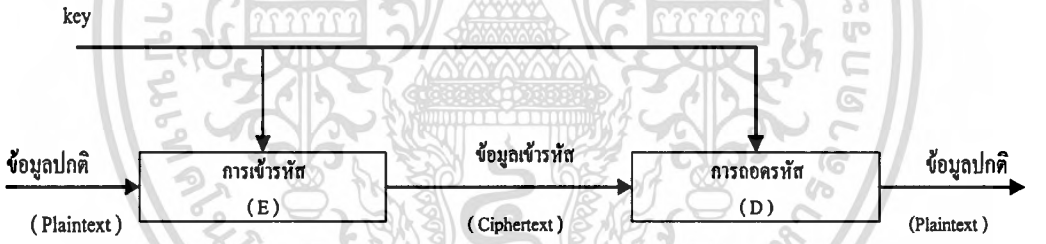
การพัฒนาโปรแกรม และ ข้อเสนอแนะของการนำอัลกอริทึม DES และ RSA มาผสมผสานกัน เพื่อให้การเข้ารหัส-ถอดรหัสข้อมูลมีประสิทธิภาพในการทำงานมากสุด ตลอดจนกล่าวถึงความสามารถในการรักษาความปลอดภัยของจาวาแอปเพล็ตที่ทำให้เกิดข้อจำกัดในการเก็บข้อมูล

บทที่ 2
หลักการของการเข้ารหัส - ถอดรหัส

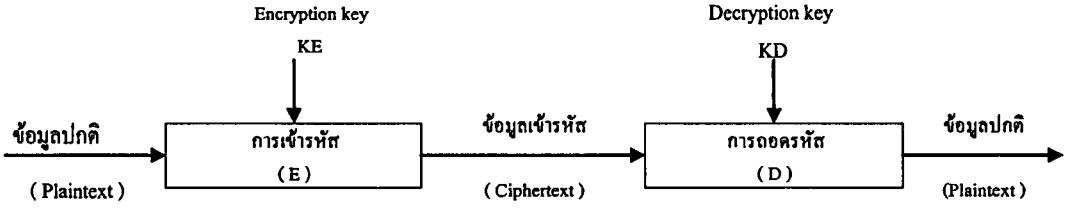
2. หลักการพื้นฐานของการเข้ารหัส - ถอดรหัส

สามารถแสดงวิธีการทำงานตามรูปภาพข้างล่างนี้

รูปที่ 1 แสดงการเข้ารหัสและถอดรหัส



(ก) ระบบการเข้ารหัสลับด้วยคีย์ตัวเดียว



(ข) ระบบการเข้ารหัสลับด้วยคีย์ 2 ตัว

รูปที่ 2 แสดงการเข้ารหัสลับด้วยคีย์

Plaintext คือ ข้อมูลเดิม

Ciphertext คือ ข้อมูลที่ถูกเข้ารหัสแล้ว

- **วิธีการจารกรรมข้อมูลกระทำได้หลายกรณี**

ปัจจุบันการส่งข้อมูลระยะไกลมีความสำคัญมากขึ้น โอกาสที่ข้อมูลจะถูกจารกรรมก็มีมาก ด้วยในการส่งข้อมูลข่าวสารจากผู้ส่ง S ไปตามสายส่ง T เพื่อไปยังผู้รับ R หากมีบุคคลภายนอก มาทำการขัดขวางหรือจารกรรมข้อมูลข่าวสารนั้น เขาสามารถกระทำได้หลายกรณี คือ

1. ป้องกันไม่ให้ข้อมูลนั้นไปถึงผู้รับ R (Interruption)
2. ขโมยดูหรือฟังข้อมูลข่าวสาร (Interception)
3. ยึดและแปรข้อมูลข่าวสาร (Modification)
4. การสอดแทรกข้อมูลข่าวสารเสมือนมาจากผู้ส่ง S (Fabication)

- **ลักษณะของระบบการเข้ารหัสที่ดี**

ระบบการเข้ารหัสลับที่ดีควรเป็นระบบที่มีลักษณะสมบัติดังนี้

1. การกระจายความถี่ของตัวอักษรและกลุ่มตัวอักษรที่ เกิดขึ้นของข้อมูลที่เข้ารหัสแล้ว ควรมีค่าใกล้เคียงกัน
2. ระบบควรป้องกันไม่ให้คำหรือวลีของข้อมูลที่เกิดซ้ำๆ กันเมื่อเข้ารหัสลับแล้วไปเป็นข้อมูลเข้ารหัสตัวเดิมในที่นี้จะหมายถึงว่าระบบควรเข้ารหัสลับเป็นกลุ่มของตัวอักษร แทนที่จะเป็นการเข้ารหัสลับแต่ละตัวอักษร
3. ระบบการเข้ารหัสลับควรจะเป็นระบบที่ให้ผู้ใช้เลือกขนาดของกุญแจเองได้ เพื่อป้องกันการแอบถอดรหัส และต้องใช้เวลายาวนานเพื่อการถอดรหัสลับนั้น
4. ระบบการเข้ารหัสลับที่ดีควรเป็นระบบที่แก้ไขปัญหาอุบัติเหตุไม่ตั้งใจที่เกิดจากการใช้กุญแจที่ผิด การถอดรหัสลับข้อมูลทั้งๆ ที่ยังไม่ได้เข้ารหัสลับไว้เลย จะไม่มีผลเสียหาย ไม่ว่าเราจะเข้ารหัสลับหรือถอดรหัสลับ เช่น ถ้าเราใช้กุญแจรหัสลับที่ผิดในการถอดรหัสลับเราก็เพียงทำการเข้ารหัสลับด้วยกุญแจตัวเดิมจากนั้นก็ถอดรหัสลับด้วยกุญแจตัวใหม่ที่ถูกต้อง
5. วิธีการเข้าและถอดรหัสควรเป็นวิธีที่เข้าใจง่าย เพราะถ้าระบบยากแล้วอาจทำให้เกิดข้อผิดพลาดได้ง่าย หรืออาจทำให้ลืมได้ง่าย
6. ข้อผิดพลาดที่เกิดขึ้น ในการสร้างรหัสลับต้องไม่แผ่กระจายให้ข้อผิดพลาดเพิ่มมากขึ้น เพราะถ้ามีข้อผิดพลาดเกิดขึ้นเล็กน้อย ผู้รับอาจสามารถคาดเดาตัวอักษรที่ขาดหายหรือผิดพลาด
7. ขนาดของข้อมูลเข้ารหัสลับแล้วจะต้องไม่ยาวกว่าข้อมูลปกติเดิม

2.1 ระบบการสร้างรหัสลับ (Cryptographic System)

2.1.1 ระบบกุญแจสาธารณะ (Public - key System)

กุญแจสาธารณะ (Public - key) หรือ asymmetric cryptosystem นี้ใช้คู่ของกุญแจ คือ กุญแจสาธารณะ และ กุญแจส่วนตัว ข้อมูลจากผู้ส่งจะถูกเข้ารหัสด้วย กุญแจสาธารณะ และจะถูกถอดรหัสได้เพียงการใช้ กุญแจส่วนตัว ที่เป็นคู่ของมันเท่านั้น ซึ่งการเข้ารหัสแบบนี้สามารถเปิดเผยกุญแจที่ใช้ในการเข้ารหัสได้ เนื่องจากกุญแจการเข้ารหัสและถอดรหัสจะเป็นคนละตัวกัน กุญแจการเข้ารหัสไม่สามารถนำมาใช้ในการถอดรหัสได้ ดังนั้นถ้ารู้กุญแจสำหรับเข้ารหัสก็ไม่มีความหมาย ผู้ใช้ (user) จึงสามารถที่จะกระจาย กุญ เฉสา ธา รณะ ไปได้อย่างอิสระ เป็นที่ไหนก็ได้เพราะถ้าไม่มี กุญแจส่วนตัว ก็ไม่สามารถจะถอดรหัส ได้ แต่เราควรจะเก็บ กุญแจส่วนตัว ในที่ ๆ ปลอดภัยที่สุด ระบบการสร้างรหัสลับ แบบใช้ กุญแจสาธารณะ มีพื้นฐานของหลักการมาจาก Trapdoor One - way Functions เป็นฟังก์ชันที่ง่ายต่อการคำนวณ แต่การคำนวณค่าฟังก์ชันย้อนกลับทำได้ยาก

2.1.1.1 ลักษณะของการเข้ารหัสแบบใช้ กุญแจสาธารณะ

ระบบการสร้างรหัสลับ แบบใช้ กุญแจสาธารณะ ขั้นตอนการเข้ารหัส หรือ ถอดรหัสต้องมีกุญแจการเข้ารหัส และข้อความที่ต้องการจะส่ง (M) เมื่อเข้ารหัสแล้วจะได้ข้อความเข้ารหัส (C) ความปลอดภัยของระบบจะขึ้นกับการที่กุญแจสำหรับเข้ารหัสสามารถเปิดเผยได้ แต่จะไม่สามารถใช้คำนวณได้

ในตอนต้นของศตวรรษที่ 17 เรารู้จักแต่เพียง classical block ciphersystems แต่ก็พบว่าไม่ได้ใช้งานทางด้านการค้ามากนัก ต่อมา ระบบการสร้างรหัสลับ แบบใช้ กุญแจสาธารณะ ถูกสร้างขึ้นมาใน ปี 1976 โดย Whitfied Diffie และ Martin Hellman (Diffie-Hellman algorithm) และโดย Ralph Merkle เนื่องจากมีการพัฒนาระบบการสร้างรหัสลับอื่น ๆ มากมาย แต่ที่มีชื่อเสียงมากเสียงมากที่สุดคือที่ถูกพัฒนาโดย Rivest ,Shamir, และ Adleman (สร้างขึ้นมาเพื่อหลีกเลี่ยงข้อเสียข้อใหญ่ของ classical cryptography เนื่องจากใช้คณิตศาสตร์ สำหรับทำ cryptoanalysis ในปี 1978 ซึ่งมีชื่อว่า RSA (เป็นชื่อจากผู้ที่เริ่มต้นคิดค้นขึ้นมา) ต่อมาถูกพัฒนา เป็น บริษัท RSA Data Security ใช้ทางด้านการค้า ส่วน RSAREF จะเกี่ยวกับ API สำหรับ RSA ซึ่งไม่เกี่ยวข้องกับด้านการค้า แต่ก็ถูกใช้ในด้านผลิตภัณฑ์ของสินค้าด้วย

ทฤษฎีทางคณิตศาสตร์ ที่ใช้ในระบบการสร้างรหัสลับแบบใช้ กุญแจสาธารณะ ก็จะมีขอบเขตของตัวมันอยู่ ระบบการสร้างรหัสลับ แบบใช้ กุญแจสาธารณะ ส่วนใหญ่แม้ว่าจะเป็น

RSA ก็มีรากฐานทางคณิตศาสตร์เกี่ยวกับ ทฤษฎีของเลขจำนวนเฉพาะ (prime-number) ทั้ง กุญแจสาธารณะ และ กุญแจส่วนตัว

ในปี 1978 R. Rivest, A. Sharmir, และ L. Adleman ได้คิดวิธีการเข้ารหัสที่มีชื่อว่า การเข้ารหัสแบบ อาร์ เอส เอ (RSA) เป็นการเข้ารหัสที่เปิดเผยกุญแจได้ ซึ่งใช้การคำนวณทางคณิตศาสตร์ของเลขจำนวนเต็มที่มีขนาดใหญ่ ความยากในการถอดรหัสวิธีนี้ ขึ้นกับหลักการทางคณิตศาสตร์ของจำนวนเฉพาะ (prime number) และการแยกตัวประกอบของจำนวนที่มีขนาดใหญ่ วิธีการนี้ได้มีการนำไปใช้ในโปรแกรม Netscape Web Browser ซึ่งเป็นโปรแกรมที่กำลังเป็นที่แพร่หลายอยู่ในปัจจุบัน

2.1.1.2 วิธีการเข้ารหัสแบบ อาร์ เอส เอ

ในการเข้ารหัสด้วย อาร์ เอส เอ อัลกอริทึม (RSA algorithm) ข้อความจะถูกแทนด้วยตัวเลขจำนวนเต็ม ซึ่งเปลี่ยนจาก ข้อความ เป็น รูปแบบตัวเลข (อาจใช้ รหัสแอสกี แทน ตัวอักษร) ผู้ใช้แต่ละคนจะเลือก ค่า n ของตนเอง และเลขจำนวนเต็มอีกคู่ คือค่า e และ ค่า d ผู้ใช้สามารถ แสดงกุญแจการเข้ารหัส (n,e) ไว้ในที่เปิดเผยได้ ส่วนกุญแจการถอดรหัสประกอบด้วยค่า (n,d) โดยที่ค่า d เก็บไว้เป็นความลับ การเข้ารหัสข้อความ C และข้อความที่ถูกถอดรหัส M กำหนดได้ดังนี้

กำหนดให้

n	คือ	p*q
public-key	คือ	(n,e)
private-key	คือ	(n,d)
ข้อความ	คือ	M
ข้อความที่ถูกเข้ารหัส	คือ	C
ดังนั้น การเข้ารหัส		$C = E(M) = M^e \bmod n$
การถอดรหัส		$M = D(C) = C^d \bmod n$

เมื่อ e และ n เป็นกุญแจในการเข้ารหัส , d และ n เป็นกุญแจในการถอดรหัส

ค่า n ได้จากการเลือกจำนวนเฉพาะ 2 จำนวน p และ q ที่มีค่ามาก ๆ ค่า n จะเป็นผลคูณของ p กับ q จะได้

$$n = p * q$$

ในการเข้ารหัสและการถอดรหัสขึ้นกับ Euler's Generalization of Fermat's Theorem ซึ่งกล่าวว่าสำหรับแต่ละ M ที่สัมพันธ์เฉพาะ (relatively prime) กับจำนวน n

$$M^{U(n)} \bmod n = 1$$

เมื่อ U(n) = (p - 1)*(q - 1) เรียกว่า Euler Totient Function คุณสมบัตินี้หมายความว่า ถ้า e และ d ที่สอดคล้องกับสมการ

$$ed \bmod U(n) = 1$$

จะได้ว่า การเข้ารหัสและการถอดรหัสเป็นอินเวิร์ส ฟังก์ชันกัน ซึ่งแสดงได้ดังนี้

ให้ e และ d สอดคล้องตามสมการ

$$M\,U(n) \bmod n = 1$$

และให้ข้อความ M เป็นจำนวนที่อยู่ในช่วง [0, n-1] ที่ซึ่ง GCD(M,n) = 1

(GCD = Greatest Common Division) หรือ ตัวหารร่วมมาก จะได้

$$M = D(C) = D(E(M)) = (M^{e} \bmod)^{d} \bmod n$$

$$= M^{ed} \bmod n$$

จาก $ed \bmod U(n) = 1$ หมายความว่า $ed = tU(n) + 1$ สำหรับจำนวนเต็ม t ดังนั้น จะได้ว่า

$$M^{ed} \bmod n = M^{tU(n)+1} \bmod n$$

$$= M\,M^{tU(n)} \bmod n$$

$$= M(M^{tU(n)} \bmod n) \bmod n$$

เมื่อ

$$M^{tU(n)} \bmod n = (M^{tU(n)} \bmod n)^{t} \bmod n$$

$$= 1^{t} \bmod n$$

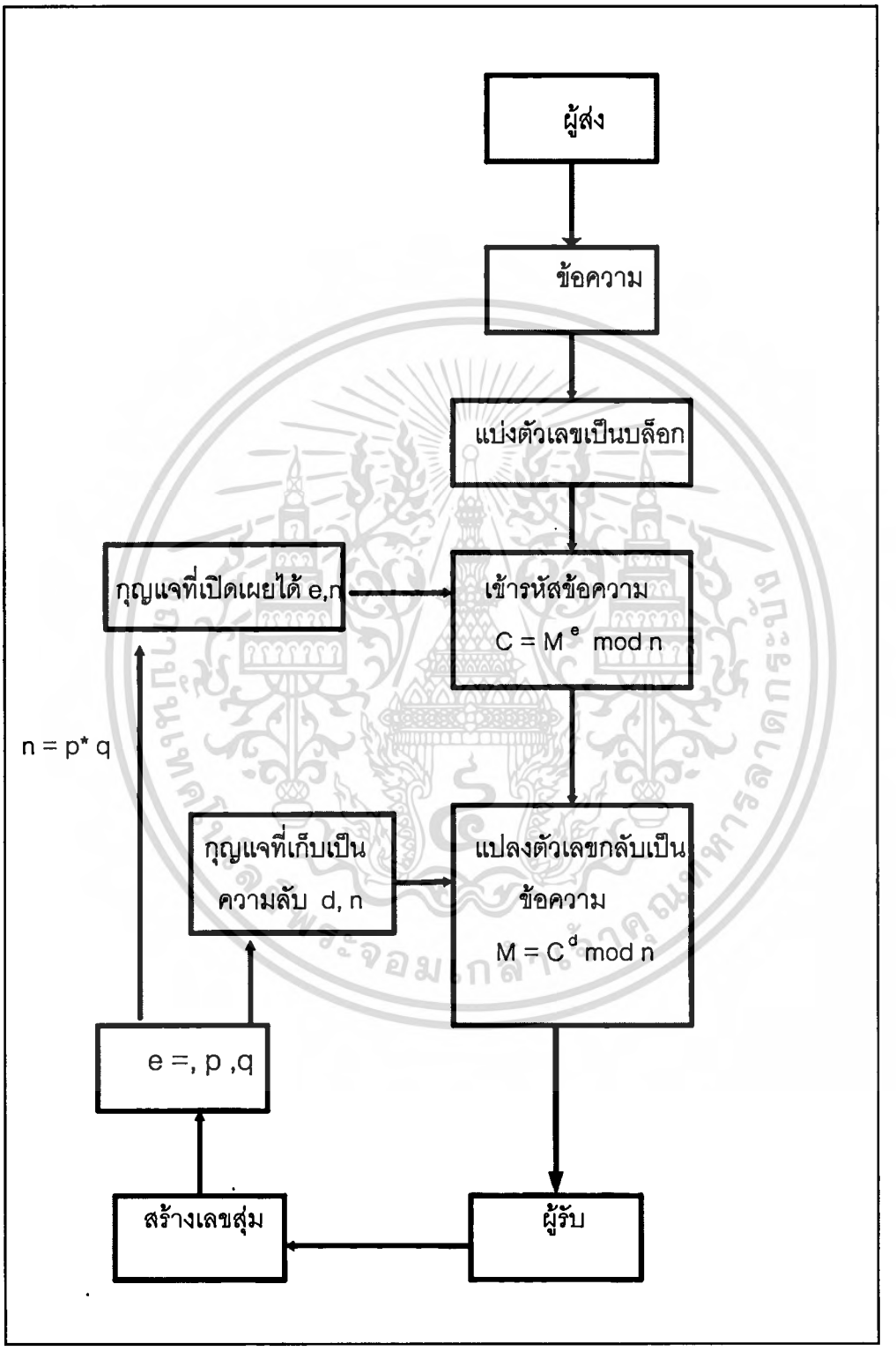
$$= 1$$

ดังนั้น

$$M^{ed} \bmod n = (M * 1) \bmod n = M$$

จึงสามารถสรุปขั้นตอนของการเข้ารหัส - ถอดรหัส ด้วย อาร์ เอส เอ ได้ดังนี้

2.1.1.3 ขั้นตอนของการเข้ารหัสด้วย อาร์ เอส เอ อัลกอริทึม



รูปที่ 3 ขั้นตอนการเข้ารหัสด้วย อาร์ เอส เอ

เพราะข้อมูล ไม่สามารถเปลี่ยนแปลงได้ระหว่างการส่ง หรือจะพิจารณาอีกอย่างก็ได้คือถ้าผู้รับรับข้อมูลได้ถูกต้องแน่นอน public-key ที่มีอยู่ก็เป็นของผู้ส่งที่ถูกต้องเช่นกัน สำหรับเรื่องของลายเซ็นต์ ดิจิตอล (digital signature) ก็สามารถตรวจสอบ public keys ที่แท้จริงได้

2) เพิ่มความปลอดภัย และความสะดวกสบายมากขึ้น คือ ตัว กุญแจสาธารณะ ไม่จำเป็นต้องถูกส่งไปด้วย และไม่จำเป็นจะต้องเปิดเผยให้คนอื่นรู้ ซึ่งมีความปลอดภัยปลอดภัยมากกว่าตัว secret-key , ตัว secret-key ต้องถูกส่งไปด้วย (ไม่ว่าจะเป็นการส่งด้วยกันเองหรือผ่าน communication channel) ดังนั้นจึงมีโอกาสที่อาจจะถูกเจอตัว secret-key ในระหว่างการส่ง

3) เหมาะกับการที่ต้องใช้ข้อมูลที่มีความลับร่วมกัน (shared secret) เพื่อความปลอดภัยจากบุคคลที่3 (ข้อมูลที่เกี่ยวข้องกับฐานข้อมูลส่วนกลางที่จะต้องเก็บกุญแจความลับของ ผู้ใช้ทั้งหมด)

4) ผู้ใช้แต่ละคนสามารถได้รับ การตอบสนองกลับมา ด้วยกุญแจสาธารณะ ของตนเองแต่เพียงผู้เดียว

ข้อเสีย

1) ระบบเข้ารหัสลับแบบกุญแจสาธารณะ จะมีการทำงานช้ากว่า เมื่อเปรียบเทียบกับระบบเข้ารหัสลับ แบบ กุญแจเดียว (secret-key) ดังนั้นในการนำไปใช้เพื่อให้มีความปลอดภัย และประโยชน์มากสุดคือ การผสมผสานระหว่างระบบการเข้ารหัสลับแบบกุญแจสาธารณะ และกุญแจเดียว (secret-key) ในกรณีนี้ก็จะใช้กุญแจสาธารณะ เป็นกุญแจ ที่จำเป็นที่ใช้ในการติดต่อสื่อสาร โดยอยู่ภายใต้ระบบเข้ารหัสลับแบบ กุญแจเดียว ระบบ กุญแจเดียว ก็จะถูกใช้ในการเข้ารหัสข้อมูล ระบบกุญแจสาธารณะ และ กุญแจเดียว นี้ร่วมกันแก้ปัญหาของการเก็บ secret keysและ private keys ผู้ใช้ต้องแน่ใจว่า private keys หรือ secret keys ที่ถูกเก็บนั้นมันปลอดภัย

2) ระบบการเข้ารหัสแบบกุญแจสาธารณะ จะไม่จำเป็นต้องใช้เมื่อมีความปลอดภัยของกุญแจลับ (secret - key) อยู่แล้วด้วย เช่น ผู้ใช้ที่ทำงานส่วนตัวที่ซึ่งรู้เพียงคนเดียว และสามารถจัดการกับ key เองได้

จึงสามารถสรุปได้ว่า ระบบการเข้ารหัสแบบกุญแจสาธารณะ ไม่ได้นำไปใช้แทนวิธีการของ secret-key cryptography แต่ กุญแจสาธารณะ สามารถจะเพิ่มความปลอดภัยของ secret-key มากขึ้น และ ตัวของ secret-key เองจะมีความเร็วมากกว่า และมันก็ยังมีความสำคัญมากสุดอยู่

2.1.1.5 การเจาะหากุญแจ อาร์ เอส เอ

การ "break RSA" ที่เป็นไปได้ก็มีอยู่ 2-3 ทาง ในความเสียหายที่พบมากสุดคือเกิดจากการที่ผู้บุกรุก (attacker) พบ private-key ต่อจากนั้นผู้บุกรุกก็สามารถอ่านข้อความทั้งหมดที่ถูกเข้ารหัสแล้ว วิธีการที่จะทำได้คือ แยกแฟคเตอร์ของ public modulus คือแฟคเตอร์ของ n หาจำนวน 2 จำนวนที่เป็นผลคูณของจำนวนเฉพาะ p และ q ซึ่ง p, q และ e เป็นส่วนประกอบของ public ผู้บุกรุกก็สามารถที่จะหา d ได้อย่างง่าย (ซึ่ง d เป็นส่วนประกอบของ private) ดังนั้นความปลอดภัยของ RSA จึงขึ้นอยู่กับแฟคเตอร์ ถ้ายิ่งเรามีแฟคเตอร์ ที่สามารถหาได้ยากเท่าไร RSA ก็มีความปลอดภัยมากขึ้นเท่านั้น ถ้าในกรณีที่ เราเพิ่มความสามารถของฮาร์ดแวร์ให้มีความสามารถที่จะคำนวณแฟคเตอร์ของจำนวนเฉพาะที่มีขนาดใหญ่ไ ด้เพื่อให้ยากแก่การหาผลคูณของจำนวนเฉพาะแล้วมันก็มีส่วนทำให้เพิ่มความปลอดภัยของ RSA ได้เช่นกัน

จำนวนหลัก	จำนวนครั้งของการกระทำ	เวลา
50	$1.4*10^{10}$	3.9 ชั่วโมง
75	$9.0*10^{12}$	104 วัน
100	$2.3*10^{15}$	74 ปี
200	$1.2*10^{23}$	$3.8*10^{9}$ ปี
300	$1.5*10^{29}$	$4.9*10^{15}$ ปี
500	$1.3*10^{39}$	$4.2*10^{25}$ ปี

ตารางที่ 1 แสดงเวลาที่ใช้สำหรับแยกตัวประกอบ

2.1.1.6 สรุป

ในการเข้ารหัสแบบนี้ สามารถทำได้หลายลักษณะ คือทำเป็นฮาร์ดแวร์โดยทำเป็นวงจรรวมที่สามารถคำนวณกับตัวเลขที่มีค่ามาก ๆ หรือเขียนเป็นซอฟท์แวร์ขึ้นมาโดยมีโปรแกรมย่อยที่จัดการกับจำนวน ที่มีขนาดใหญ่และสามารถหาโมดูลเลขยกกำลังได้เร็ว เราจะพบว่าการเข้ารหัสแบบนี้ ยากต่อการถอดรหัส ก็เนื่องมาจากปัจจุบันยังไม่มีวิธีการที่สามารถแยกตัวประกอบได้อย่างมีประสิทธิภาพ สำหรับวิธีการที่มีอยู่ในปัจจุบันเวลาที่ใช้ในการแยกตัวประกอบสำหรับค่าที่เป็นผลคูณของจำนวนเฉพาะที่ใหญ่มาก จะใช้ เวลานานมากดังแสดงในตาราง นั่นหมายความว่าการเข้ารหัสแบบ RSA นี้จะไม่สามารถรักษาความปลอดภัยของข้อมูลได้ ถ้ามีวิธีการที่มี

ประสิทธิภาพ ในการแยกตัวประกอบเพราะจะทำให้สามารถหากุญแจที่ใช้ในการถอดรหัสจากกุญแจที่ใช้ในการเข้ารหัสได้

ทุกวันนี้วิธีการที่ดีที่สุดก็ไม่สามารถใช้แยกตัวประกอบที่มีจำนวน 200 หลักได้เร็วพอ ถึงแม้จะใช้เครื่องที่มีการทำงานแบบขนาน (Parallel Processing) การเข้ารหัสแบบนี้ จึงยังสามารถใช้ได้อีกนานและยากที่จะถอดรหัสโดยที่ไม่ทราบกุญแจในการถอดรหัส

2.1.2 การเข้ารหัสโดยใช้ Secret-key

2.1.2.1 Caesar cipher

เป็นการ code ที่ง่ายที่สุดในแต่ละ plaintext การเลือกตัวมาแทนที่ plaintext character ถูกเรียกว่า mono-alphabetic cipher หรือ Caesar cipher

Caesar จะทำการ เลื่อนลำดับอักษร (shift) แต่ละตัวอักษรของ message ควรจะมี algorithm มาตรฐานของตัวเอง (เช่น shift ทีละ 3 ไปทางขวา) ตัวอย่าง message คือ "BANANA"

message เดิม "BANANA" จะ encrypt ได้ดังนี้ "EDQDQD"

จากตัวอย่าง "B" ถูก shift เป็น "E" , "A" ถูกshiftเป็น "D" จะเห็นว่าตัวอักษรแต่ละตัวจะถูก shift ไป ข้างหลังอีก 3 ตัวอักษร ส่วนการถอดรหัส ก็จะทำตรงข้ามกับการเข้ารหัสคือการที่ตัวอักษรแต่ละตัวก็ต้องเลื่อนไปข้างหน้า 3 ตัวอักษรเช่นกัน

2.1.2.2 DES

องค์การต่าง ๆ ให้ความสนใจในการป้องกัน และรักษาความปลอดภัยของข้อมูลมากขึ้น และมีการพัฒนาการเข้ารหัสมากมาย แต่ก็ยังพบ ว่ามีปัญหาเรื่องการติดต่อสื่อสารกัน เพราะต่างก็ใช้อัลกอริทึมและอุปกรณ์ที่แตกต่างกัน ดังนั้นจึงมีการพยายามที่จะสร้างมาตรฐานในการเข้ารหัสลับข้อมูลขึ้น โดยมีหน่วยงานที่รับผิดชอบ คือ National Bureau of Standard (NBS) ในปี 1972 ได้พยายามให้เกิดวิธีการเข้ารหัสลับสาธารณะ (public encryption algorithm)ที่มีคุณสมบัติในระบบการเข้ารหัสลับดังนี้

1. อัลกอริทึมที่จะออกแบบ ต้องมีความสมบูรณ์ ชัดเจน

2. ต้องทราบว่าอัลกอริทึม นี้มีความสามารถในการป้องกันข้อมูลได้แค่ไหน ต้องทราบระยะเวลาที่ใช้ในการประมวลผล และ จำนวนขั้นตอนการทำงานในการค้นหากุญแจ

3. ประสิทธิภาพในการป้องกันข้อมูลจะขึ้นอยู่กับคีย์ที่จะต้องเก็บเป็นความลับเท่านั้นไม่ใช่

4. ในการทำงานของอัลกอริทึม เพื่อเข้ารหัสลับ จะต้องไม่กระทบกระเทือนต่อการทำงานของผู้ใช้ (user)

จนกระทั่งในปี 1974 บริษัท ไอบีเอ็ม ได้เสนออัลกอริทึม ที่เรียกว่า " Lucifer" ซึ่งต่อมาได้รับการพัฒนาปรับปรุง และศึกษาอย่างละเอียดโดยกลุ่มผู้เชี่ยวชาญ จนได้อัลกอริทึมใหม่ออกมา ชื่อว่า " อัลกอริทึมเดส " (Data Encryption Standard (DES)) และได้รับการยอมรับให้เป็นมาตรฐาน เมื่อวันที่ 23 พฤศจิกายน ค.ศ. 1976

2.1.2.3 หลักการเข้ารหัสแบบ DES อัลกอริทึม

การเข้ารหัสแบบ DES ใช้เพื่อป้องกันการลักลอบ การดักฟังข้อมูลในระบบคอมพิวเตอร์ หลักการและทฤษฎีของ DES ใช้วิธีทางคณิตศาสตร์เข้ามาสลับสับเปลี่ยนข้อมูล ข้อมูลที่เข้ามาผ่านขบวนการเข้ารหัสจะถูกสลับสับเปลี่ยนเป็นข้อมูลที่ไม่มีรูปแบบเรียกว่า CIPHER การถอดรหัสจะนำข้อมูลที่ไม่มีรูปแบบนี้มาเข้าขบวนการถอดรหัส จะถูกสลับสับเปลี่ยนเป็นข้อมูลเดิมได้ ทฤษฎีของ DES อธิบายถึงวิธีการเข้าและถอดรหัสโดยใช้เลขฐาน 2 กุญแจประกอบด้วยเลขฐาน 2 64 บิท ใช้ขบวนการเข้าและถอดรหัส และอีก 8 บิท ใช้ในการตรวจสอบข้อผิดพลาดของ 56 บิท (Error Detection)

ข้อมูลที่ต้องการเข้ารหัสจะต้องใช้ร่วมกับกุญแจ กุญแจ 56 บิท จะต้องใช้ร่วมกับทฤษฎีการเข้ารหัสและถอดรหัสอีก 8 บิท เป็นการตรวจสอบข้อผิดพลาดซึ่งให้ค่าเป็น 1 (PARITY BIT) ในแต่ละไบต์ (8 บิท) เมื่อมีค่าเป็นเลขคู่ ในแต่ละด้านของการเข้าและถอดรหัสจะมีกุญแจที่เหมือนกัน กุญแจที่เลือกใชจะเป็นกุญแจที่เมื่อใช้ในการเข้ารหัสและถอดรหัสแล้วจะได้ข้อมูลเดิม การใช้กุญแจที่ถอดรหัสที่ไม่ใช่กุญแจที่ใช้ในการเข้ารหัสจะส่งผลให้เมื่อถอดรหัสแล้วจะไม่ได้ข้อมูลเดิม ด้วยเหตุผลนี้การใช้กุญแจที่เหมือนกันจะเก็บค่าของกุญแจเป็นความลับจะป้องกันการดักฟังหรือการโจรกรรมข้อมูลได้

ข้อมูลที่จะถอดรหัสได้นั้นจะต้องใช้กุญแจเดียวกับการเข้ารหัสเท่านั้น ผู้ใดที่รู้หลักการหรือทฤษฎีของการเข้ารหัส แต่ไม่รู้หมายเลขกุญแจที่ใช้ จะไม่สามารถถอดรหัสได้โดยง่าย ผู้ที่รู้ทฤษฎีและหมายเลขกุญแจเท่านั้นที่จะถอดรหัสได้ในเวลาอันสั้น

ระบบมาตรฐาน DES เป็น วิธีการที่ผสมรวมการเข้ารหัสลับพื้นฐานมาใช้คือ วิธีการแทนที่ข้อมูล (Substitution) และวิธีสับเปลี่ยนตำแหน่งของบิต (Transposition หรือ Permutation) และอาศัยวิธีการทางคณิตศาสตร์ คือการเอกว์คลูซีฟออร์ (Exclusive - or) นำวิธีการเหล่านี้มาทำงานร่วมกัน สร้างเป็น อัลกอริทึม ที่มีการทำงานลักษณะเดียวกันซ้ำ ๆ กัน 16 วงรอบ โดยข้อมูลปกติเดิมจะผ่านการเข้ารหัส เป็นบล็อกของข้อมูล จำนวน 64 บิต และสามารถกำหนดกุญแจยาวเป็นตัวเลข 64 บิต ที่ผู้ใช้สามารถเปลี่ยนแปลงตามต้องการ

เนื่องจากวิธีการของ DES ใช้การ กระทำทาง ตรรกศาสตร์ และเลขคณิตแบบมาตรฐานทั่ว ๆ ไป ดังนั้นเราจึงสร้างระบบการเข้ารหัสลับ DES นี้ทาง ซอฟท์แวร์บนเครื่องคอมพิวเตอร์ปัจจุบันได้แทบทั้งหมด และสามารถเป็นระบบทางฮาร์ดแวร์ด้วยไอซีชิพเพียงตัวเดียว

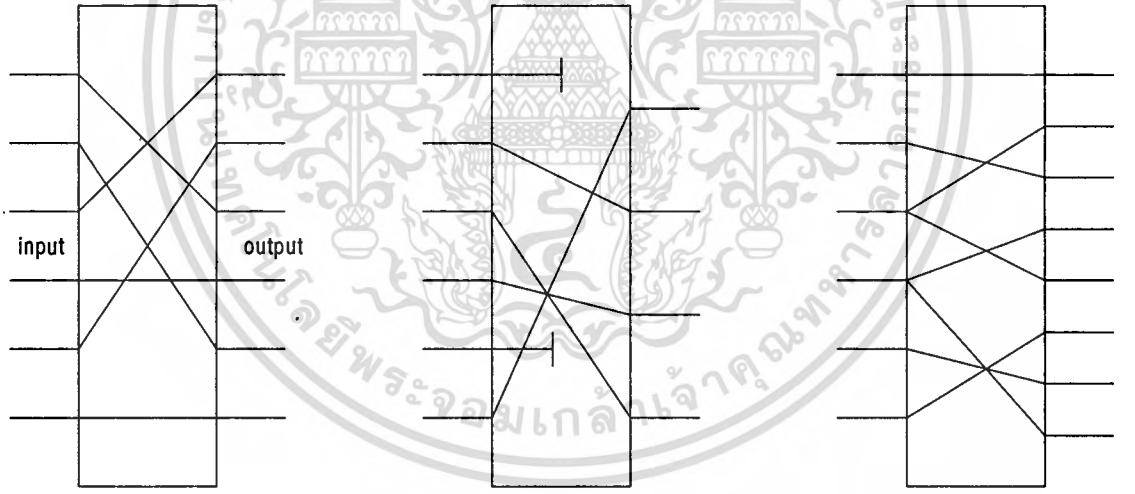
2.1.2.4 การเข้ารหัสลับพื้นฐาน

โดยการจัดลำดับตำแหน่งของบิต ทำได้ 3 ลักษณะคือ

1. PERMUTATION คือการจัดลำดับตำแหน่งของบิตใหม่ เมื่อจัดเสร็จแล้วยังคงมีจำนวนบิตเท่าเดิม

2. PERMUTATION CHOICE คือ การจัดลำดับตำแหน่งของบิตใหม่ โดยการนำบางบิตของข้อมูลเท่านั้นมาจัดลำดับ และบางบิตจะไม่นำมาใช้ เมื่อจัดเสร็จจะมีจำนวนบิตลดลง

3. EXPANDED PERMUTATION คือ การจัดลำดับตำแหน่งของบิตใหม่ โดยมีการใช้บางบิตของข้อมูลซ้ำ เมื่อจัดเสร็จจะมีจำนวนบิตเพิ่มขึ้น



(1) Permutation (2) Permutation Choice (3) Expanded Permutation

รูปที่ 4 แสดงการเข้ารหัสลับพื้นฐาน

DES แบ่งการทำงานเป็น 2 ส่วน คือ

- ส่วนสร้างกุญแจ มีความยาว 56 บิท
- ส่วนเข้ารหัสและถอดรหัส มีความยาว 64 บิท

การถอดรหัส กระทำได้จะต้องรู้ถึง

■ ทฤษฎีที่เข้ารหัส

■ กุญแจ

ถ้ารู้ทฤษฎีแต่ไม่รู้กุญแจต้องใช้เวลานานเพื่อทำการทดลองทุก ๆ กุญแจที่เป็นไปได้ กุญแจใช้ 56 บิท = 2^{56} ประมาณ 7 หมื่นล้านล้าน กุญแจที่เป็นไปได้แต่ถ้าไม่รู้เทคนิคการเข้ารหัสจะไม่สามารถถอดรหัสได้เลย

2.1.2.5 ขบวนการเข้า และ ถอดรหัส

แสดงในรูปที่ 5 ข้อมูลเข้าเป็นบล็อก Ti เข้าสู่ขบวนการสลับสับเปลี่ยนตำแหน่ง IP ดังตารางที่ 2 และข้อมูลออก T0 = IP(T) ข้อมูล T0 เข้าสู่ขบวนการสลับสับเปลี่ยนผนวกกับรหัสกุญแจ 16 ครั้ง ซึ่งเรียกว่า Function_F เมื่อเสร็จขบวนการสลับสับเปลี่ยนผนวกกับรหัสกุญแจ 16 ครั้ง จะเข้าสู่ขบวนการสลับสับเปลี่ยนตำแหน่งย้อนกลับ IP^{-1} ดังตารางที่ 3 ผลลัพธ์ที่ได้ เป็นข้อมูลที่ผ่านขบวนการเข้ารหัสแล้ว

จากรูปที่ 5 ขบวนการสลับสับเปลี่ยนผนวกกับรหัสกุญแจ 16 ครั้ง แต่ละครั้งเรียกว่า Function_F อยู่ระหว่างการทำสลับสับเปลี่ยนตำแหน่ง IP กับการเข้าสลับสับเปลี่ยนตำแหน่งย้อนกลับ

ให้ Ti แสดงถึงผลลัพธ์ของครั้งที่ i แบ่ง Ti ออกเป็น 2 ส่วน ๆ ละบิท 32 บิท แรก ให้เป็นด้านซ้าย L และด้านขวา R ฉะนั้น LiRi จะเป็น

$Li = t1....t32$

$Ri = t33...t64$

ดังนั้น $Li = Ri - 1$

$$Ri = Li - 1 \oplus F(Ri-1, Ki) \qquad (1)$$

เครื่องหมาย $\oplus$ หมายถึง การทำเอ็คคุซีบออร์ (Exclusive - or) และ Ki คือ หมายเลขกุญแจมีความยาว 48 bit

ขบวนการสลับสับเปลี่ยนผนวกกับรหัสกุญแจ มีทั้งหมด 16 รอบ แสดงในรูปที่ 3 รอบที่ 1 ถึง 15 สลับสับเปลี่ยนตาม (1) ในรอบที่ 16 รอบสุดท้าย ด้านซ้าย L และด้านขวา R จะไม่สลับที่กัน ข้อมูล R16 L16 ถูกใช้เป็นเข้าเพื่อทำการสลับสับเปลี่ยนตำแหน่งย้อนกลับ IP^{-1}

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

ตารางที่ 2 ตารางสลับสับเปลี่ยนตำแหน่ง IP

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

ตารางที่ 3 ตารางสลับสับเปลี่ยนตำแหน่ง IP^{-1}

Function_F และ S_box

Function_f(R_{i-1}, K_i) การทำงานเริ่มจาก R_{i-1} ถูกสลับสับเปลี่ยนเป็น 48 บิท โดยใช้ $E(R_{i-1})$

ต่อมา $E(R_{i-1})$ และ Ki มาทำ Exclusive-Or ผลลัพธ์ที่ได้ถูกแบ่งเป็นข้อมูลขนาด 6 บิท 8 ชุด คือ ชุดที่ B1,.....,B8 จะได้

$E(R_{i-1}) \oplus Ki = B1B2.....B8$ (2)

ข้อมูลในแต่ละชุด Bj ซึ่งมีขนาด 6 บิท จะใช้เป็นข้อมูลเข้าใน Functiion(S_box)Sj และข้อมูลออกจะมีขนาด 4 บิท เขียนในรูปของฟังก์ชั่นได้

Sj(Bj) (3)

S_box จะมีทั้งหมด 8 ชุดหลังจากผ่านขบวนการ S_box จะได้ผลลัพธ์ขนาด 32 บิท และเข้าขบวนการสลับสับเปลี่ยน P

ผลลัพธ์ของ S_box หลังการเข้าขบวนการสลับสับเปลี่ยน P จะอยู่ในรูปของ $F(R_{i-1},Ki)$ เขียนได้เป็น

P(S1(B1).....S8(B8)) (4)

วิธีการเข้า S_box อธิบายได้ดังนี้

ให้ Bj เป็นข้อมูลเข้ามีขนาด 6 บิท (b1b2b3b4b5b6) ให้ j เป็นเลขจำนวนเต็มมีค่าระหว่าง 1 ถึง 8 แบ่ง Bj ออกเป็น 2 ส่วนเรียกว่า Row และ Column ให้ในส่วนของ Row = b1b6 และส่วนของ Column = b2b3b4b5 นำค่า Row และ Column มาเทียบในตาราง Sj จะได้ผลลัพธ์เป็นค่า 4 บิท ดังนั้นเขียนได้เป็น

Sj(Bj) ให้ผล ลัพธ์ขนาด 4 บิท

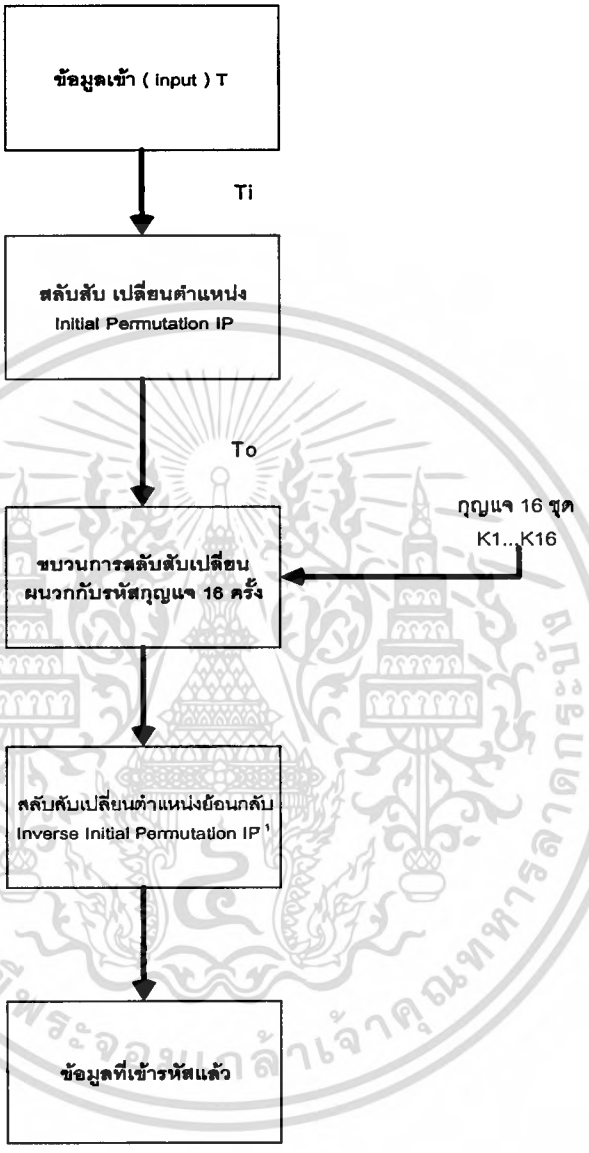
<u>ตัวอย่าง</u>

ถ้า B1 = 101010 S1 จะได้ผลลัพธ์เป็น Row = 10 = 2
Column = 0101 = 5 เทียบในตาราง S1 จะได้ 6 ซึ่งเท่ากับ 0110

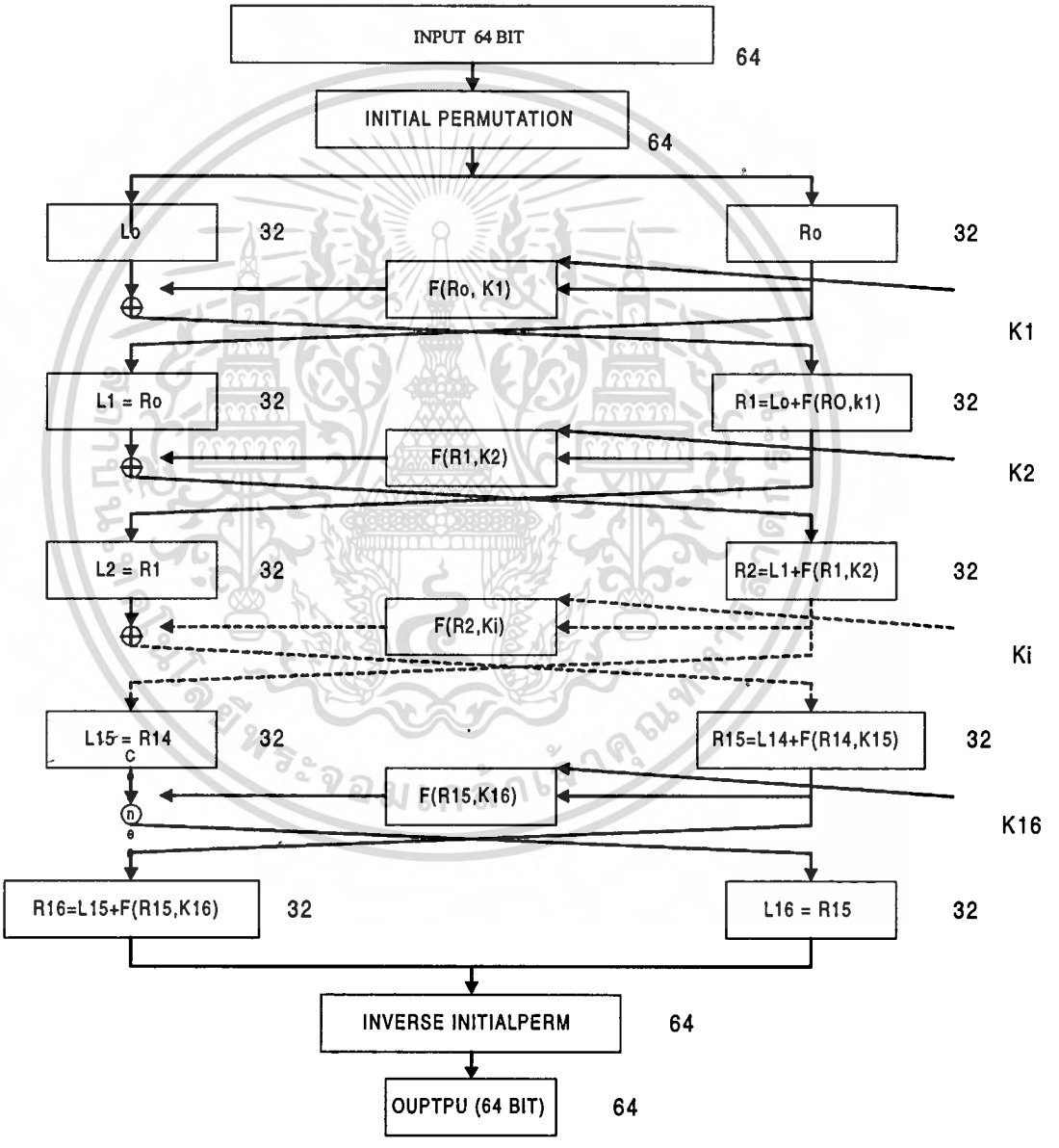
ถ้า B7 = 100101 S1 จะได้ผลลัพธ์เป็น Row = 11 = 3
Column = 0010 = 2 เทียบในตาราง S7 จะได้ 13 ซึ่งเท่ากับ 1101

	Row	Column															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	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
S2	0	15	1	8	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
S3	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
S4	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
S5	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
S6	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
S7	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	12
S8	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

ตารางที่ 4 Choice Permutation To Select 48 Key Bit



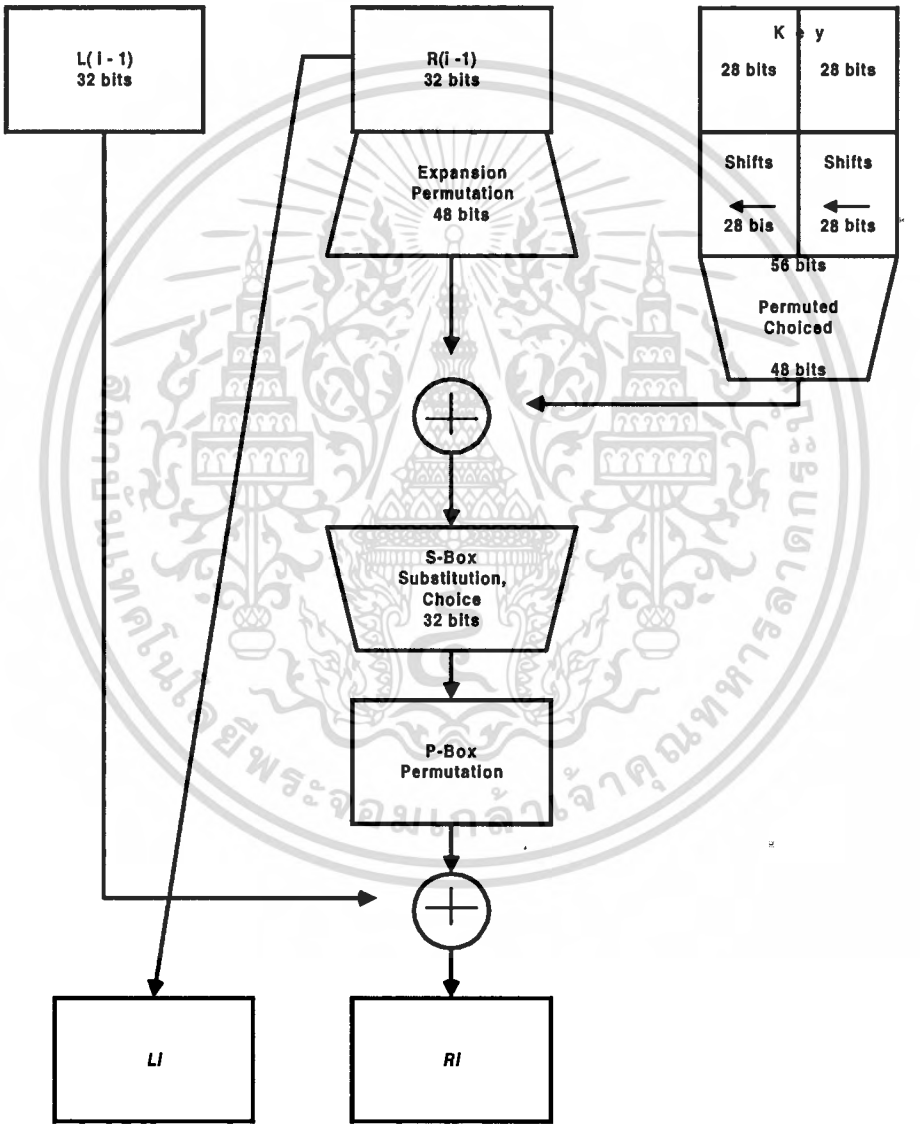
รูปที่ 5 แสดงขบวนการเข้า และถอดรหัส



รูปที่ 6 แสดงวิธีการเข้ารหัสอย่างละเอียด

Data 64 bits สำหรับการเข้ารหัส

ในส่วนของข้อมูลที่จะเข้ารหัส จัดข้อมูลออกเป็นกลุ่ม ๆ กลุ่มละ 8 character หรือขนาด 64 บิท นำข้อมูลมาทำการเข้ารหัสทีละกลุ่ม ก่อนอื่นต้องทำ permutation จากนั้น จะเป็นการเข้ารหัสตามอัลกอริทึม DES ในแต่ละรอบโดยใช้ 64 bit- data เข้ารหัสกับ subkey ตัวที่ n (n = 1, 2, …, 15, 16)



รูปที่ 7 แสดงวิธีการเข้ารหัส 1 วงรอบ

2.1.2.6 การคำนวณหมายเลขกุญแจ (Subkey)

ในแต่ละรอบของการทำสลับสับเปลี่ยนผนวกกับรหัสกุญแจ 16 ชุด จะใช้หมายเลขกุญแจแต่ละชุด ซึ่งมีขนาด 48 บิท หมายเลขกุญแจ 16 ชุด ได้มาจาก Key ตามรูปที่ 5 Key เป็นข้อมูลเข้ามีขนาด 64 บิท ซึ่ง 8 บิท ในตำแหน่งที่ 8,16,.....,64 ใช้เป็นบิทในการตรวจสอบข้อผิดพลาด(Parity bit) การทำขบวนการสลับสับเปลี่ยน Pemuted PC_1 หรือ Permuted choice 1 จะตัดในส่วนของบิทตรวจสอบข้อผิดพลาดออก ใช้ 56 บิทที่เหลือเท่านั้น

ผลลัพธ์หลังจากการทำ PC_1 จะถูกแบ่งเป็น 2 ส่วนเท่า ๆ กันเรียกว่า C และ D และใช้ C และ D หาค่าของหมายเลขกุญแจ Ki

กำหนดให้ Ci และ Di ซึ่งได้มาจาก C , D ใช้หาค่า Ki จะได้

$Ci = Lsi(C_{i-1})$

$Di = Lsi(D_{i-1})$

Lsi คือการทำการหมุนทางซ้ายเท่ากับจำนวนครั้งที่กำหนด ในตารางที่ 3 C0 และ D0 เป็นข้อมูลเริ่มแรกของ C และ D ดังนั้นหมายเลขกุญแจ Ki เขียนได้เป็น

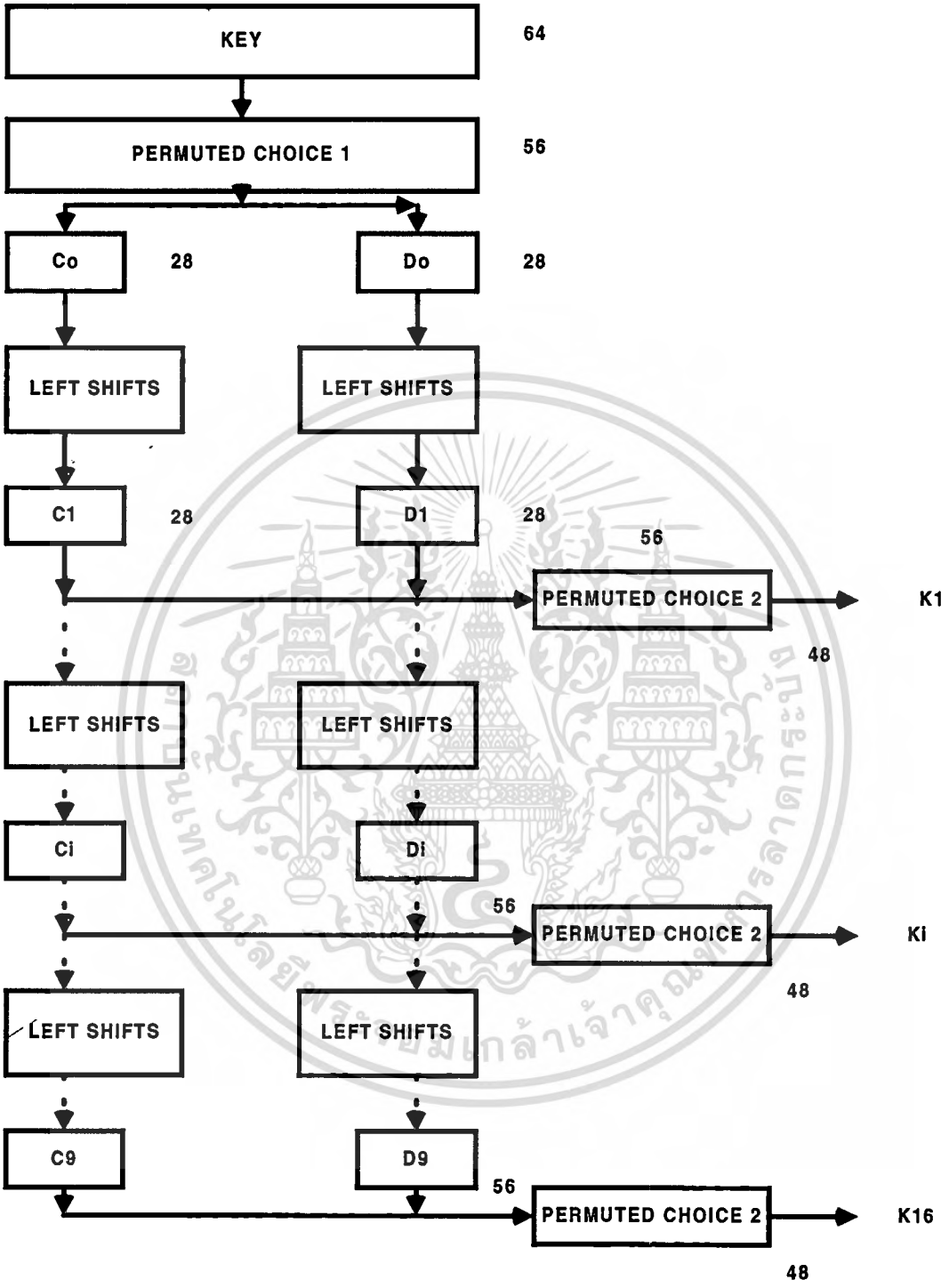
Ki = PC_2 (CiDi)

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

ตาราง 5 แสดง การ Choice Permutation เพื่อเลือกกุญแจให้เป็น 48 บิท

ลำดับ	จำนวนครั้ง
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

ตาราง ที่ 6 แสดงจำนวนครั้งของการ shift กุญแจในแต่ละรอบ



รูปที่ 8 แสดงวิธีการคำนวณหมายเลขกุญแจ 16 ชุด

<u>หมายเหตุ</u> การทำขบวนการย้อนกลับจะย้อนกลับในส่วนของกุญแจ จาก 16 ถึง 1 ส่วนวิธีการจะไม่ย้อนกลับ

2.1.2.7 การ เจาะหากุญแจของ DES

การเจาะหากุญแจจะเข้าถึง DES ได้ มีวิธีการอย่างหนึ่งคือ การค้นหา key space ทั้งหมด ซึ่งจะต้องมีการทำงานเฉลี่ยถึง 2 ยกกำลัง 55 ขั้นตอนในระยะแรกก็มีการคิดกันว่า ถ้าสร้างคอมพิวเตอร์พิเศษขึ้นมาเพื่อเจาะหากุญแจของ DES โดยใช้วิธีการค้นหาทั้งหมด (exhaustive search) ก็อาจจะสามารถทำได้แต่ก็มีในเรื่องของเวลามาเกี่ยวข้อง หลังจากที่ Hellman ได้แสดง time-memory ว่าต้องใช้ memory จำนวนมาก ๆ

1) The RSA Secret-key Challenge

The RSA Secret-key Challenge ถูกประกาศที่ RSA Data Security Conference ในเดือนมกราคม ซึ่งใช้ RC5 ในการทำการเจาะหากุญแจของ DES (RC5 คือ กุญแจที่หลากหลายของ RSA- RSA's variable key. กุญแจที่เป็นสมมาตร ที่ 40 บิท และ 48 บิท RC5 key challenge สามารถเจาะหากุญแจได้แล้ว มีวิธีการทั้งหมด 12 วิธีของ RC5 challenge) RSA Secret-key Challenge จะใช้ความสามารถของการทำการประมวลผลแบบกระจาย บนอินเตอร์เน็ต และ RC5 สามารถใช้กับการหาได้หลายขนาดของกุญแจ และกับ 56 บิท ของ DES ที่มีความยากได้ด้วย

2) วิธีการเจาะหากุญแจของ DES

Rocke Verser กับ ผู้ร่วมงานคือ Matt Curtin และ Justin Dolske เป็นผู้สามารถเจาะหากุญแจของ DES ได้ (ปัจจุบันทางที่จะทำการเจาะหากุญแจของ DES จะใช้วิธี "brute force" ผู้ที่เจาะหากุญแจจะพยายาม หา กุญแจ DES ที่เป็นไปได้ จนกระทั่งพบกุญแจที่ใช้ในการเข้ารหัสตัวที่เหมาะสม) Rocke จะสร้าง "cracking" โปรแกรม ซึ่งจะเก็บกุญแจตัวใหม่มา จนหาสามารถถอดรหัส DES ได้ โปรแกรมสามารถใช้แบบกระจายได้ และ download ได้ โครงการทำงานนี้มีชื่อรหัสว่า DESCHALL สามารถเชื่อมต่อกันได้พร้อม ๆ กันเป็นหมื่นเครื่อง ของคอมพิวเตอร์อาสาสมัคร แต่ละเครื่องของคอมพิวเตอร์อาสาสมัครใหม่ DES key space จะถูกทดสอบตัว กุญแจ DES ที่ผิดจะถูกกำจัดออกไป และกุญแจตัวที่ถูกเท่านั้นที่จะถูกเปิดเผย

กลุ่มของผู้ทำการเจาะหากุญแจของ DES บนอินเตอร์เน็ต รวมถึงอาสาสมัคร และสิ่งที่ใช้ในการคำนวณด้วยนั้นต้องการกระจายสิ่งเหล่านี้ซึ่งต้องใช้การคำนวณมาก ๆ การคำนวณของ DESCHALL บางจุดมีการทดสอบถึง 7 พันล้าน กุญแจ ต่อ วินาที

บทที่ 3

ภาษา จาวา

จาวา คือ ภาษาคอมพิวเตอร์แบบ Object - Oriented มีความคล้ายกับภาษา C/C++ โดยตัดข้อเสียบางอย่างของ C/C++ ออกไปและเพิ่มข้อดีหลาย ๆ อย่างเข้าไป ทั้งนี้เป็นเพราะ Java ถูกพัฒนาขึ้นเพื่อใช้งานกับระบบเครือข่าย Internet ซึ่งต้องปรับปรุงและพัฒนาความสามารถหลาย ๆ อย่างให้เหมาะสมกับการใช้งานบนเครือข่ายคอมพิวเตอร์ แต่ลักษณะการเขียนโปรแกรมเหมือนกับภาษา C/C++ Java ถูกคิดและพัฒนาขึ้นโดยนาย James Gosling และ ทีมงาน Green group ของบริษัท Sun Micrisystems ในปี 1991 มีชื่อเดิมของภาษาว่า Oak ถูกประกาศตัวครั้งแรกในปี 1995 ในงาน Sun World'95 ในชื่อว่า Java และบริษัทซันได้เปิดตัว Web - browser ตัวใหม่ชื่อว่า Hotjava

World Wide Web (WWW) มีการทำงานแบบ Client / Server มีการแลกเปลี่ยนข้อมูลกันบนเครือข่ายคอมพิวเตอร์ จึงนำเอา Java มาเป็นส่วนหนึ่งของ Web เพื่อเพิ่มประสิทธิภาพ ในการทำงาน เมื่อ Client หรือผู้ขอใช้บริการซึ่งอยู่ในที่ต่าง ๆ ของระบบเครือข่าย Internet สามารถทำงานโต้ตอบ (Interactive) กับ Homepage นั้น ๆ ได้ สามารถ download โปรแกรมซึ่งเป็นโปรแกรม Java จาก Server หรือผู้ให้บริการมาปฏิบัติงาน หรือ รัน (Run) บนเครื่อง Client ได้ ซึ่ง Java ได้ถูกออกแบบให้มีความปลอดภัยในการใช้งานกับระบบเครือข่าย Internet โดยเฉพาะ

3.1 จาวา แอพเพล็ต และ จาวา แอพพลิเคชัน

ทั้ง จาวา แอพเพล็ต และ จาวา แอพพลิเคชัน เป็นโปรแกรมซึ่งเขียนขึ้นจากภาษา จาวา เหมือนกันทุกประการและใช้ javac.exe เป็นตัวคอมพายเลอร์เหมือนกัน จะแตกต่างกันเพียงที่ แอพเพล็ต ถูกปฏิบัติงานร่วมกับไฟล์ .html โดยถูกเรียกจาก Tag <APPLET> ภายในไฟล์ .html โดยใช้โปรแกรม appletviewer.exe หรือ Web - browser เช่น Netscape Navigator หรือ Microsoft - Internet Explorer เป็นตัว viewer ดูผลลัพธ์ของการปฏิบัติงาน แอพเพล็ต ไม่สามารถปฏิบัติงานเดี่ยว ๆ ได้ ส่วน แอพพลิเคชัน ใช้ปฏิบัติงานแบบเดี่ยว ๆ (Standalone Java application) เหมือนโปรแกรมทั่วไปทั้ง Text - mode และ Graphics - mode อย่างเช่น โปรแกรม Text Editor ,Paintbrush, Spread Sheets, เป็นต้น แต่ที่สำคัญใน Application จะต้องมี Method main(String

agrs[]) อย่างน้อยหนึ่ง Method เพื่อเป็นจุดเริ่มต้นของโปรแกรมเหมือนภาษา C/C++ ทั่วไป โดยใช้ java.exe เป็นอินเตอร์เพรทเตอร์ในขณะที่กำลังปฏิบัติงาน

3.2 ระบบรักษาความปลอดภัยของจาวา

ภาษา จาวา จะมี tigh restrictions บน memory access ที่แตกต่างจากโมเดลของภาษา C อย่าง มาก ข้อจำกัดนี้รวมไปถึงการตัด pointer arithmetic และ illegal cast operators ออกด้วย

- bytecode verification routine ใน Java interpreter จะตรวจสอบไบต์โค้ดที่ไม่ละเมิดโครงสร้างของภาษาใด ๆ (ซึ่งอาจจะเกิดขึ้นได้ หากมีคอมไพเลอร์ ภาษา Java ให้เลือกมากขึ้นในอนาคต) รูทีนในการตรวจสอบความถูกต้องนี้จะตรวจให้แน่ใจว่าไม่มีพอยเตอร์ หรือการเรียกใช้หน่วยความจำที่ไม่อนุญาต หรือเรียกออบเจ็กต์อื่นเกินกว่าที่มีสิทธิ เพื่อให้แน่ใจว่า method call ได้มีการใส่ค่าอาร์กิวเมนต์ครบถ้วน และถูกต้องเรียบร้อยแล้ว ด้วยรูทีนนี้จะช่วยเลี่ยงปัญหา stack overflows ได้
- การตรวจสอบ class name และ access restrictions ขณะที่โหลด
- ระบบอินเทอร์เฟซซีเคียวริตี (interface security system) ให้ระดับในการรักษาความปลอดภัยหลายระดับ
- ระดับในการเรียกใช้ไฟล์ (file access level) ถ้าไบต์ โค้ดพยายามเรียกใช้ไฟล์ที่ไม่ได้ รับอนุญาตจะมีไดอะล็อกบอกซ์แสดงขึ้นมาผู้ใช้เลือกว่าจะทำงานต่อไปหรือหยุดการเอ็กซิคิวท์
- ในระดับเน็ตเวิร์ก มีการใช้ public key encryption และ cryptographic technique เพื่อตรวจสอบซอร์สโค้ด และข้อกำหนดหลังจากส่งผ่านเน็ตเวิร์กแล้ว เทคโนโลยีการเข้ารหัสนี้จะเป็นหัวใจสำคัญสำหรับทรานเซ็กชันที่เกี่ยวกับการค้าขายผ่านเน็ตเวิร์ก
- ที่รันไทม์ ข้อมูลเกี่ยวกับต้นกำเนิดของไบต์โค้ดอาจถูกใช้เพื่อตัดสินใจว่าโค้ดสามารถทำอะไรได้บ้าง กลไกรักษาความปลอดภัยจะแจ้งให้ทราบว่าไบต์โค้ดสร้างขึ้นภายใน Firewall หรือ ไม่ คุณสามารถเซ็ตระบบรักษาความปลอดภัยที่จำกัดสิทธิของโค้ดที่ไม่น่าไว้วางใจได้ด้วย

3.3 การทำงาน ของ Java

1. ผู้ใช้ส่ง request เพื่อเรียกเอกสาร HTML จากเซิร์ฟเวอร์
2. เอกสาร HTML ถูกส่งไปยังบราวเซอร์ของผู้ใช้ เอกสารที่ส่งไปจะประกอบด้วย APP tag ซึ่งเป็นผู้กำหนดแอพเพล็ต
3. ไบต์โค้ดของแอพเพล็ตที่เกี่ยวเนื่องกันจะถูกส่งผ่านไปยังเครื่องโฮสต์ของผู้ใช้ทั้งหมด โดยไบต์โค้ดเหล่านี้จะถูกสร้างขึ้นล่วงหน้า

4. บราวเซอร์บนโอสต์ของผู้ใช้ จะอินเทอร์พรีตไบต์โค้ด และแสดงผล
5. ผู้ใช้อาจโต้ตอบกับแอพเพล็ตได้โดยไม่ต้องดาวน์โหลดไฟล์จากเซิร์ฟเวอร์เพิ่มเติมเลย เนื่องจากไบต์โค้ดได้เก็บข้อมูลที่แอพเพล็ตไว้ทั้งหมดแล้ว

3.4 การรักษาความปลอดภัยของแอพเพล็ต (Applet Security)

- โดยทั่ว ๆ ไป applet ที่ถูกโหลดมาจากเครือข่าย จะป้องกันจากการอ่าน และเขียนไฟล์ ที่ไคลเอนต์ ไฟล์ ซิสเต็ม (Client file system) ยกเว้นที่ originating host ถ้าใช้ Java - enabled browser ทำการรันแอพเพล็ตจะไม่สามารถทำได้ เพราะ แอพเพล็ต จะไม่สามารถที่จะอ่าน หรือ เขียนไฟล์ได้ Sun'sappletviewer อนุญาต ให้อ่านไฟล์ หรือ เขียนไฟล์ ได้ ใน ที่ ๆ ไฟล์ มี access control list
  ถ้าไฟล์ที่ ไคลเอนต์ ไม่มี access control list แล้ว แอพเพล็ตไม่สามารถทำงานต่าง ๆ ดังนี้กับไฟล์ได้
  1. ตรวจสอบว่ามีไฟล์อยู่
  2. อ่านไฟล์
  3. เขียนไฟล์
  4. เปลี่ยนชื่อไฟล์
  5. สร้าง directory บนไคลเอนต์ ไฟล์ ซิสเต็ม
  6. list ไฟล์ใน directory
  7. ตรวจสอบชนิดของไฟล์
  8. ตรวจสอบ timestamp ที่ไฟล์ที่แก้ไขครั้งสุดท้าย
  9. ตรวจสอบขนาดของไฟล์

- แอพเพล็ตไม่อนุญาตให้ open network connection ที่ เครื่องคอมพิวเตอร์ใด ๆ ยกเว้น host ที่จัดการกับ .class file ซึ่งถ้าไม่ใช่ host ที่มี html ฝังอยู่ ก็เป็น host ที่ถูกกำหนด ใน coldebase เป็น parameter ใน applet tag

  Socket s = new Socket("foo.com",25,trye);

  ถ้า พยายามที่จะทำแบบนี้จากแอพเพล็ต แอพเพล็ตไม่สามารถเริ่มต้นจากเครื่อง foo.com ได้ จะเกิดการ fail กับ security exception
- แอพเพล็ตสามารถ open network connection ที่ originating host ได้อย่างไร เราต้องแน่ใจว่า originating host มีอยู่จริง เมื่อมันถูกกำหนดใน แอพเพล็ตที่ บราว

เซอร์ ทำการโหลด ถ้าโหลด HTML page โดยใช้ URL จะพิมพ์ดังนี้

http://foo.state.edu/~me/appletPage.html

แล้ว แอพเพล็ตจะสามารถที่จะติดต่อกับ เพียง hostที่ใช้ชื่อว่า foo.state.edu การใช้ IP address เพียง foo.state.edu จะไม่ทำงาน

- แอพเพล็ต ไม่สามารถ เริ่มโปรแกรม บนไคลเอนต์ได้ แอพเพล็ตที่โหลดบนเครือข่าย จะไม่อนุญาตให้เริ่มโปรแกรม ซึ่งแอพเพล็ตที่เข้าไปนี้ จะไม่สามารถให้ทำการประมวลผลที่ เครื่อง PC ได้ ส่วนใน UNIX แอพเพล็ตจะไม่อนุญาต ให้ทำการ fork โปรแกรมได้
- มีความแตกต่างระหว่าง แอพเพล็ต ที่โหลดบนเครือข่าย และ แอพเพล็ตที่โหลดจาก file system ถ้า แอพเพล็ต

3.5 สรุปความสามารถของแอพเพล็ต

Key:

NN : Netscape Navigator 2.x, โหลดแอพเพล็ตจากเครือข่าย

NL : Netscape Navigator 2.x, โหลดแอพเพล็ตจาก Local file system

AN : Appletviewer, JDK 1.x, โหลดแอพเพล็ตจากเครือข่าย

AL : Appletviewer, JDK 1.x, โหลดแอพเพล็ตจาก Local file system

JS : Java Standalone applications

Stricter ----------> Less strict

	NN	NL	AN	AL	JS
read file in /home/me, acl.read=null	no	no	no	yes	yes
read file in /home/me, acl.read=/home/me	no	no	yes	yes	yes
write file in /tmp, acl.write=null	no	no	no	yes	yes
write file in /tmp, acl.write=/tmp	no	no	yes	yes	yes
get file info, acl.read=null acl.write=null	no	no	no	yes	yes
get file info, acl.read=/home/me acl.write=/tmp	no	no	yes	yes	yes
delete file, using File.delete()	no	no	no	no	yes
delete file, using exec /usr/bin/rm	no	no	no	yes	yes
read the user.name property	no	yes	no	yes	yes
connect to port on client	no	yes	no	yes	yes
connect to port on 3rd host	no	yes	no	yes	yes
load library	no	yes	no	yes	yes
exit(-1)	no	no	no	yes	yes
create a popup window without a warnin	no	yes	no	yes	yes

ตารางที่ 7 แสดงความสามารถการทำงานของแอพเพล็ต

บทที่ 4
การพัฒนาโปรแกรมการเข้ารหัสด้วยภาษาจาวา

4.1 โปรแกรมการเข้ารหัส-ถอดรหัส บนอินเตอร์เน็ต

1) โปรแกรมการสร้าง กุญแจสาธารณะ(Public - key) และ กุญแจส่วนตัว (Private - key) ตามอัลกอริทึม RSA โดยจะเลือกจำนวนเฉพาะ 2 จำนวน เพื่อนำไปใช้ในการคำนวณหากุญแจสาธารณะและกุญแจส่วนตัว โดยแยกเป็น

-ฟังก์ชันแสดง การคำนวณหากุญแจสาธารณะ

-ฟังก์ชันแสดงการคำนวณหากุญแจส่วนตัว

ฟังก์ชัน(Function) แสดง การคำนวณหากุญแจสาธารณะ

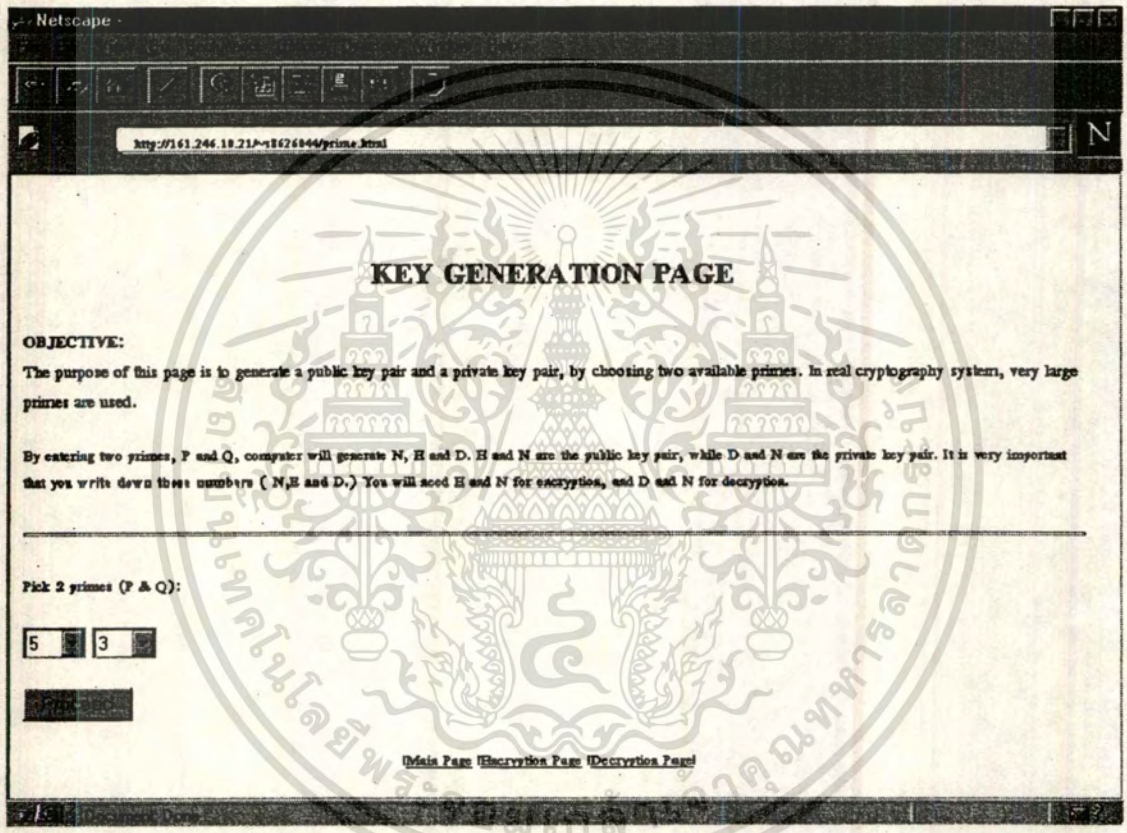
```
        public int GCD(int e,int phi)
                {int great = 0;
                 int a = 0;
                 if(e > phi)
{
  while(e % phi !=0)
    {
      a = e%phi;
      e = phi;
      phi = a;
    }
  great = phi;
}
else {
    while(phi %e != 0 )
      {
```

```
        a = phi %e;
        phi = e;
        e = a;
       }
      great = e;
   }
return great;
}
```

ฟังก์ชัน(Function) แสดงการคำนวณหากุญแจส่วนตัว(private - key)

```
public int inver(int e,int phi)
{
  System.out.println("THIS TEST E = "+e);
  System.out.println("THIS TEST PHI = "+phi);
  int u1 = 1;
  int v2 = 1;
  int u2 =  0;
  int v1 = 0;
  int u3 = phi;
  int v3 = e;
  int x,t1,t2,t3=0;
  int z,uu,vv, inverse;
  int q =0;
 double y ;
  while(v3 != 0)
{
  y = floor(u3/v3);
 q = ((int)y);
```

```
System.out.println("this test q = "+q);
t1 = u1 - q*v1;
t2 = u2 - q*v2;
t3 = u3 - q*v3;
u1 = v1;
u2 = v2;
u3 = v3;
v1 = t1;
v2 = t2;
v3 = t3;
  z = 1;
}
uu = u1;
vv = u2;
if(vv < 0)
{
inverse = vv + phi;
}
else
{
inverse = vv;
}
System.out.println("this is inverse " + inverse);
return inverse;
}
```



ภาพที่ 9 หน้าจอแสดงการสร้างกุญแจการเข้ารหัส RSA อัลกอริทึม

2) โปรแกรมการเข้ารหัสโดยวิธีการ RSA (RSA algorithm) จะเลือกข้อมูลที่ใช้ในการเข้ารหัส และใส่ ค่า N และ E ที่ได้มาจากโปรแกรมการสร้างกุญแจ (ซึ่ง N,E คือ กุญแจสาธารณะ สำหรับใช้เข้ารหัส)

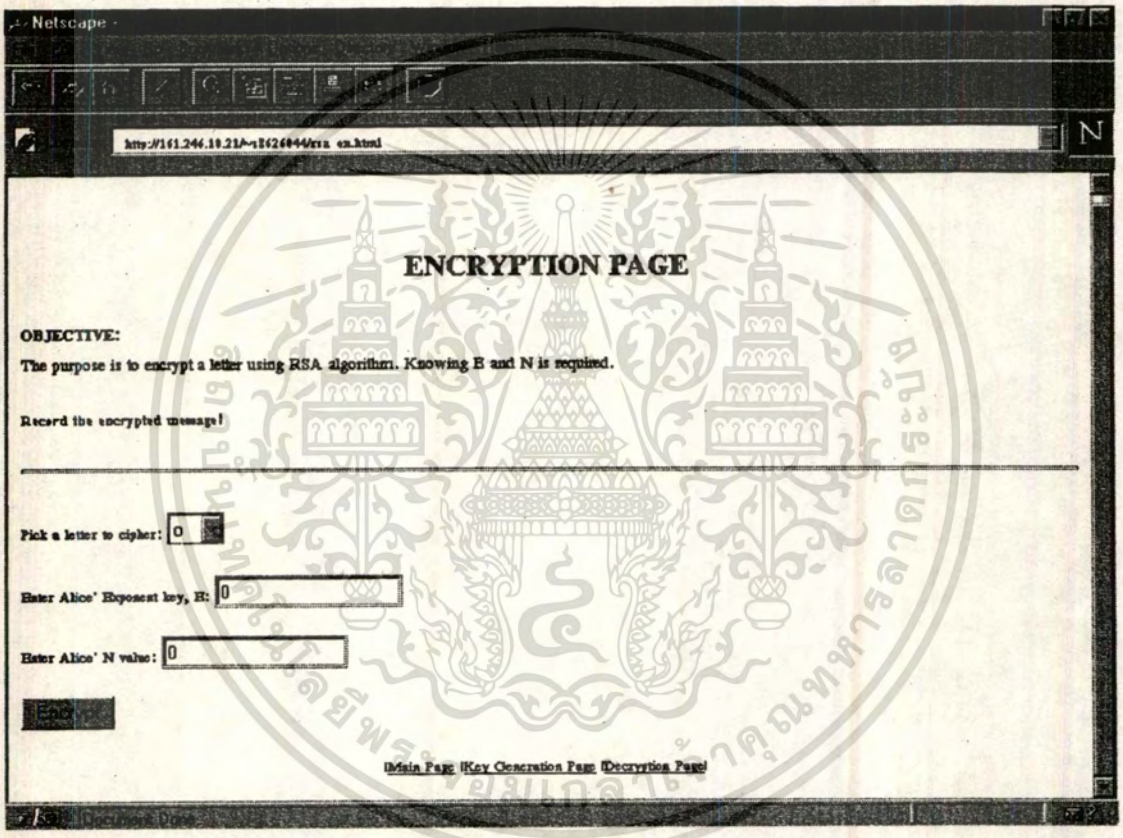
ฟังก์ชัน(Function) แสดงการคำนวณ การเข้ารหัสโดยวิธีการ RSA

```
public double power(int m,int num1)
{
 return Math.pow(m,num1);
}
public boolean action(Event evt, Object o)
{
   int i2 ;
   try
     {
      is = url.openStream();
      dis = new DataInputStream(is);
      while ((txt= dis.readLine()) != null)
         buf.append(txt);
                        dis.close();
     }
   catch (IOException e2)
   {
     System.out.println("File open error: " + e2.getMessage());
   }
         finally
          {
            repaint();
          }
          i2 = buf.length();
   System.out.println("this is i2 = "+i2);
```

```
if(evt.target == b)
 {
   num1 = Integer.parseInt(number1.getText() );
   System.out.println("THIS  prime1 is " +num1);
   num2 = Integer.parseInt(number2.getText() );
   System.out.println("THIS  prime2 is " +num2);


    for (int cc = 0; cc < i2;cc++)
    {
      chr[cc] = buf.charAt(cc);
      a[cc] = new Character(chr[cc]);
      val[cc] = a[cc].hashCode();
//      System.out.println(" Value = "+val[cc]);
      y[cc] = power(val[cc],num1);
//      System.out.println("THIS Y[CC] = " +y[cc]);
      c[cc]=((int)y[cc]%num2);
      System.out.println("C is = "+c[cc]);
    }

}
return true;
}
}
```



ภาพที่ 10 หน้าจอแสดงการเข้ารหัส โดยวิธีการ RSA

3) โปรแกรมการถอดรหัสโดยวิธีการ RSA (RSA algorithm) จะใส่ค่าข้อมูลถูกเข้ารหัสแล้ว และใส่ ค่า D และ N ที่ได้มาจากโปรแกรมการสร้างกุญแจ (ซึ่ง N,D คือกุญแจส่วนตัว สำหรับใช้ถอดรหัส)

ฟังก์ชัน(Function) แสดงการคำนวณ การถอดรหัสโดยวิธีการ RSA

```
public int cal(int c,int d,int n)
 {
 int g ;
 int f = 0;
 if(d%2 ==0)
  {
   g = 1;
   for(int i = 1 ; i<=d/2; i++)
     {
      f = (c*c)%n;
      g = (f*g)%n;
     }
  }
 else
  {
   g = c;
   for(int i = 1;i<=d/2;i++)
     {
      f = (c*c)%n;
      g = (f*g)%n;
     }
  }

 return g;
}
```

ภาพที่ 11 หน้าจอแสดงการถอดรหัส โดยวิธีการ RSA

4) โปรแกรมการเข้ารหัส - ถอดรหัสโดยวิธีการ DES (DES algorithm) จะเลือกข้อมูลที่ใช้ในการเข้ารหัส และใส่ ค่า N และ E ที่ได้มาจากโปรแกรมการสร้างกุญแจ (ซึ่ง N,E คือกุญแจสาธารณะ สำหรับใช้เข้ารหัส)

ภาพที่ 12 หน้าจอแสดงการเข้ารหัส - ถอดรหัส โดยวิธีการ DES

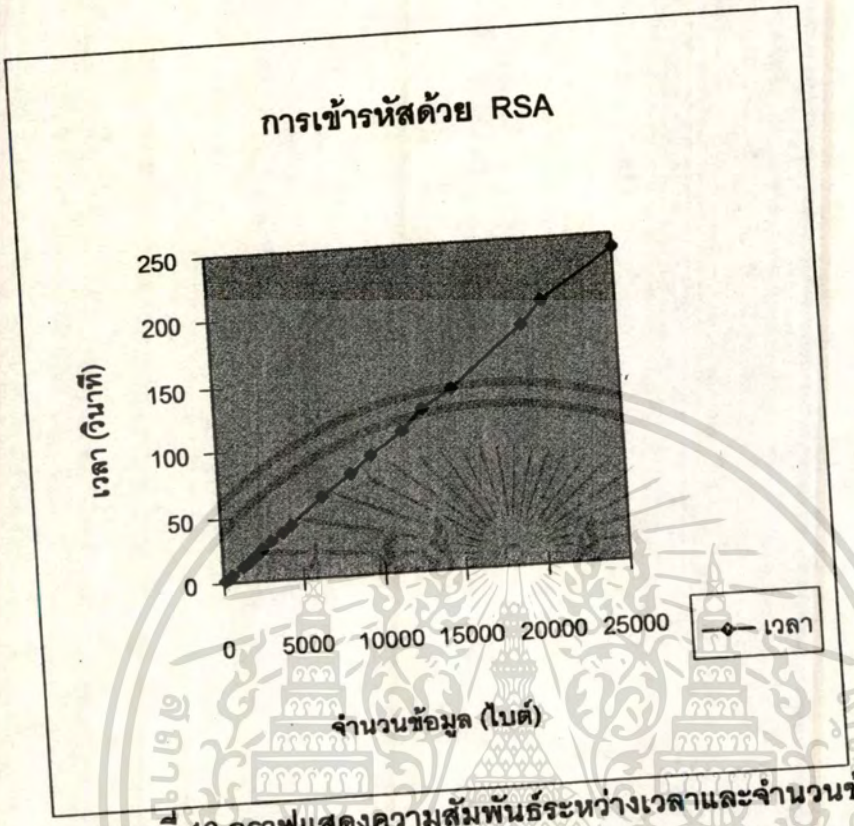
4.2 ผลการทดลองเปรียบเทียบเวลาในการทำงาน อัลกอริทึมของ DES และ RSA

ทดลองการเข้ารหัสด้วย อาร์ เอส เอ อัลกอริทึม และ เดส อัลกอริทึม โดยใช้เครื่อง ไมโครคอมพิวเตอร์ เพนเทียม 166 , เมมโมรี 16 Mb เขียนโปรแกรมด้วยภาษา Java applet ใช้ พบว่า เวลาในการเข้ารหัสข้อมูล 64 บิท = 0.075 วินาที การพยายามจะจารกรรมข้อมูลของ วิธีการ RSA (RSA algorithm) ไม่มีทางอื่นนอกจากลองทุก ๆ แฟกเตอร์ ของจำนวนเฉพาะที่คูณกันอยู่ที่เป็นไปได้

ตารางการทดลอง ทดลองหาความสัมพันธ์ด้านเวลากับขนาดข้อมูลของการเข้ารหัสด้วย RSA algorithm เมื่อใช้จำนวนเฉพาะ 4 หลักทั้ง 2 จำนวน

ข้อมูล	เวลา
160	1.5
320	3
640	6
1280	12
1600	15
1920	18
2240	21
2560	24
3200	30
3840	36
4480	42
6400	62
8320	79
9600	92
11520	109
12800	125
14700	140
19200	185
20480	202
25000	240

ตารางที่ 8 แสด งเวลาที่ ใช้ในการเข้ารหัสด้วยวิธีการ RSA (RSA algorithm) เมื่อเปลี่ยนขนาด ของข้อมูล



ภาพที่ 13 กราฟแสดงความสัมพันธ์ระหว่างเวลาและจำนวนข้อมูล ของวิธีการเข้ารหัสด้วย RSA algorithm

จากผลการทดลองจะเห็นได้ว่า เมื่อใช้ จำ จนเฉพาะ 4 หลัก และเพิ่มขนาดของข้อมูลเพิ่มเป็น 2 เท่าตัว จะได้ว่าเวลาที่ใช้ในการเข้ารหัสจะแปรผันตรงกับขนาดข้อมูล

ให้ข้อมูล 160 ไบต์ = x , เวลาที่ใช้ 1.5 วินาที = y

ข้ อมูลจากผลการทดลอง

ข้อ มูล (160)	x	ใช้เวลา (1.5)	y
ข้อ มูล (320)	2x	ใช้เวลา (3)	2y
ข้อ มูล (640)	4x	ใช้เวลา (6)	4y
ข้อ มูล (1280)	8x	ใช้เวลา (12)	8 y
ข้อ มูล (2560)	16 x	ใช้เวลา (24)	16 y
ข้อ มูล (5120)	2^5x	ใช้เวลา (48)	2^5 y
ข้อ มูล (10240)	2^6 x	ใช้เวลา (98)	2^6 y
ข้อ มูล (20480)	2^7x	ใช้เวลา (202)	2^7 y
ข้อ มูล (40960)	2^8x	ใช้เวลา (403)	2^8y
ข้อ มูล (81920)	2^9x	ใช้เวลา (823)	2^9y
ข้อ มูล (163840)	2^{10} x	ใช้เวลา (1652)	2^{10}y

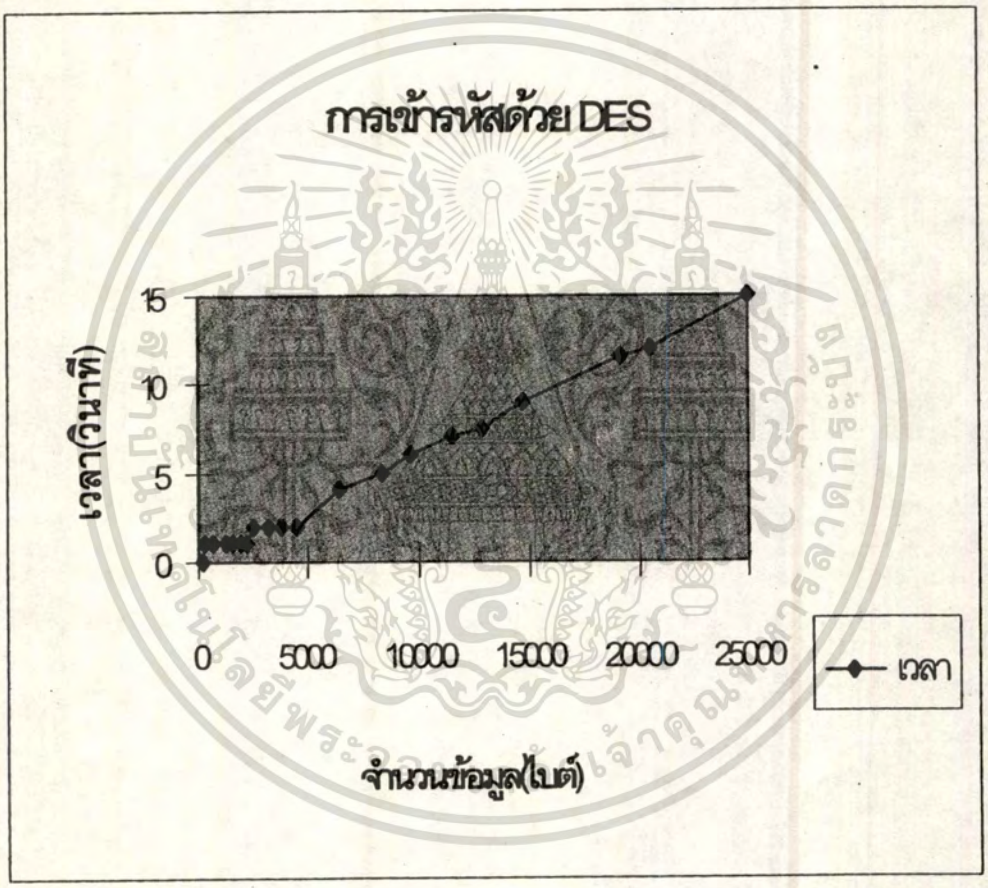
จะเห็นว่า ปริมาณข้อมูลมีความสัมพันธ์กับเวลาคือ เมื่อเวลาเพิ่มขึ้น $2^n x$ จะใช้เวลาในการเข้ารหัสเท่ากับ $2^n y$, เมื่อ x= 160 byte, y = 1.5 second

จึงสามรถสรุปได้ว่า ถ้า เราทราบขนาดของข้อ มูลเราสามารถคำนวณหาเวลาในการเข้ารหัสได้ ตาม สมการ

ตารางการทดลอง ทดลองหาความสัมพันธ์ด้านเวลากับขนาดข้อมูลเมื่อ เข้ารหัสด้วยวิธีการ DES (DES algorithm)

ข้อมูล	เวลา
160	0
320	1
640	1
1280	1
1600	1
1920	1
2240	1
2560	2
3200	2
3840	2
4480	2
6400	4
8320	5
9600	6
11520	7
12800	7.5
14700	9
19200	11.5
20480	12
25000	15

ตารางที่ 9 แสด งเวลาที่ ใช้ในการเข้ารหัส ด้วยวิธีการ DES เมื่อเปลี่ยนขนาดของข้อมูล

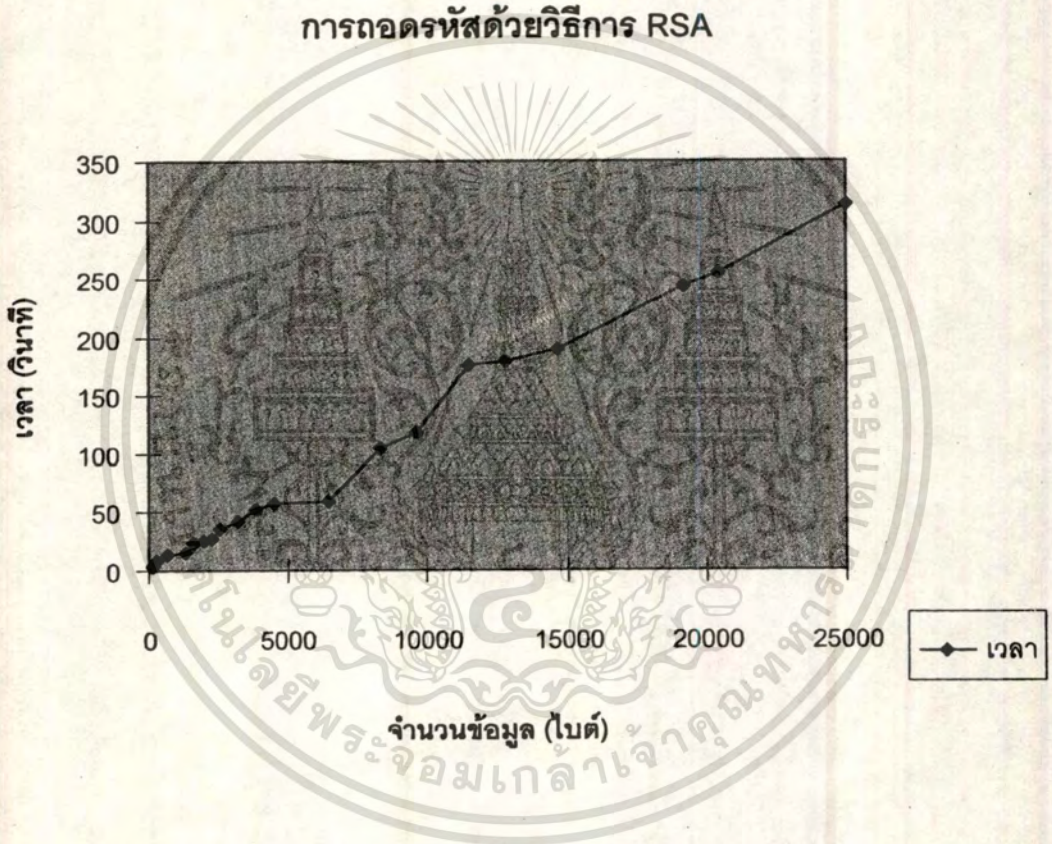


ภาพที่ 14 กราฟแสดงความสัมพันธ์ระหว่างเวลาและจำนวนข้อมูล ของวิธีการถอดรหัสด้วย DES algorithm

ตารางการทดลอง ทดลองหาความสัมพันธ์ด้านเวลากับขนาดข้อมูลของการถอดรหัสด้วย RSA algorithm

ข้อมูล	เวลา
160	0
320	3
640	8
1280	13
1600	15
1920	23
2240	25
2560	27
3200	36
3840	43
4480	52
6400	58
8320	60
9600	105
11520	118
12800	180
14700	190
19200	244
20480	256
25000	313

ตารางที 10 แสดงเวลาที่ ใช้ในการถอดรหัส ด้วยวิธีการ RSA เมื่อเปลี่ยนขนาดของข้อมูล



ภาพที่ 15 กราฟแสดงความสัมพันธ์ระหว่างเวลาและจำนวนข้อมูล ของวิธีการถอดรหัสด้วย RSA algorithm

ข้อมูล	เวลา
160	0
320	1
640	1
1280	1
1600	1
1920	2
2240	2
2560	2
3200	2
3840	2
4480	3
6400	4
8320	5
9600	6
11520	7
12800	8
14700	9
19200	12
20480	12
25000	14

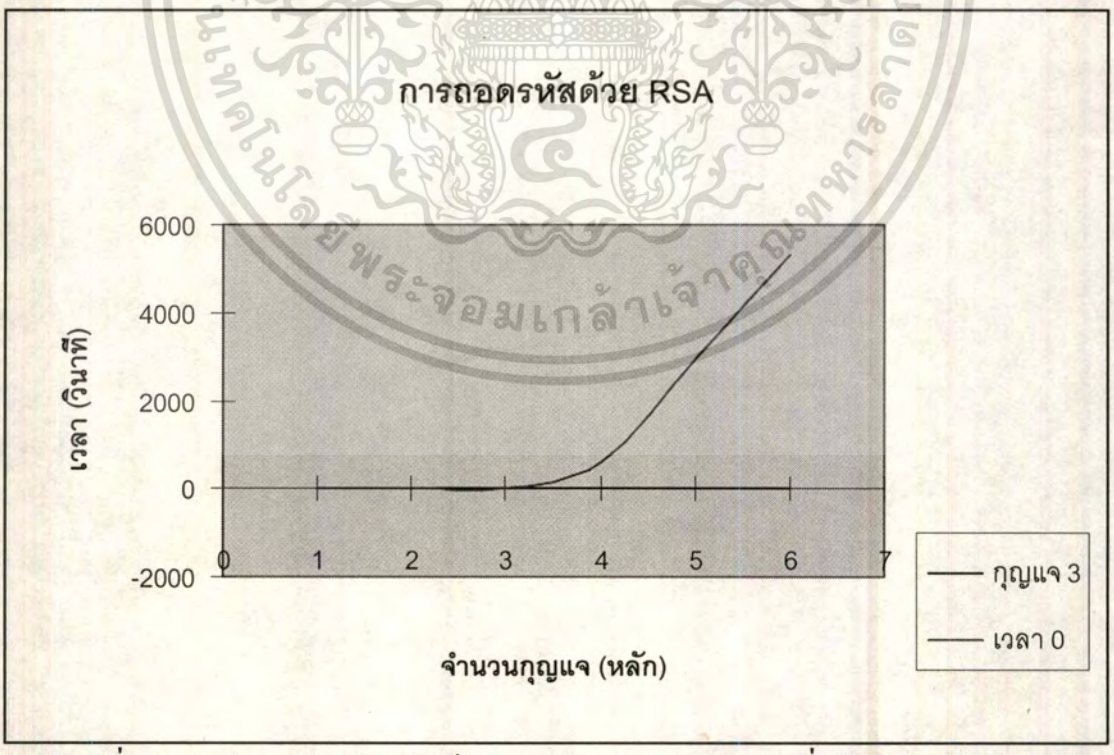
ตารางที่ 11 ผลการทดลอง การถอดรหัสด้วยวิธีการ DES

ภาพที่ 16 กราฟแสดงความสัมพันธ์ระหว่างเวลาและจำนวนข้อมูล ของวิธีการเข้ารหัสด้วย RSA algorithm

ตารางการทดลอง ทดลองหาความสัมพันธ์ด้านเวลากับจำนวนกุญแจที่ใช้ของการเข้ารหัส-ถอดรหัสด้วย RSA

จำนวนกุญแจ (หลัก)	เวลาในการเข้ารหัส (วินาที)	เวลาในการถอดรหัส (วินาที)
3	-	-
4	-	1
5	-	10
6	-	30
7	-	628
8	1	2943
9	1	5317

ตารางที่ 12 แสดงเวลาที่ใช้ในการถอดรหัสด้วยวิธีการ RSA เมื่อเปลี่ยนขนาดกุญแจ



ภาพที่ 17 กราฟแสดงความสัมพันธ์ระหว่างเวลา กับจำนวนกุญแจที่ใช้ในการถอดรหัสด้วย RSA อัลกอริทึม

4.3 สรุปผลการทดลอง

ในการทดลองการเข้ารหัส-ถอดรหัส ด้วย RSA อัลกอริทึม และ DES อัลกอริทึม โดยใช้เครื่องไมโครคอมพิวเตอร์ เพนเทียม 166 , เมมโมรี 16 Mb ทำการพัฒนาโปรแกรมการเข้ารหัส-ถอดรหัสด้วยอัลกอริทึมทั้งสองที่กล่าวข้างต้นด้วยภาษาจาวา โปรแกรมช่วยพัฒนาคือ Symantec Cafe ทำงานบน stand alone และจับเวลาด้วยนาฬิกาภายในเครื่องของคอมพิวเตอร์ จะเห็นได้ว่าการเข้ารหัส ถอดรหัสด้วยวิธีการ DES อัลกอริทึมจะเร็วกว่าวิธีการของ RSA อัลกอริทึม ซึ่งจากการทดลองเป็นเพียงการชี้ให้เห็นถึงเวลาที่ใช้กับจำนวนข้อมูลที่เปลี่ยนไปของแต่ละอัลกอริทึมเท่านั้น ซึ่งทำให้ทราบว่า DES อัลกอริทึม จะสามารถเข้ารหัส ถอดรหัสได้ เร็วกว่า RSA อัลกอริทึม ซึ่งเป็นไปตามทฤษฎีที่ได้กล่าวไว้แล้ว

จากผลการทดลองสามารถสรุปได้ดังนี้

1. ขนาดของข้อมูล มีผลต่อเวลาในการเข้ารหัส-ถอดรหัส

จากตารางการทดลองที่ 8, 9, 10 และ 11 สามารถสรุปได้ว่าเมื่อข้อมูลมีขนาดเพิ่มมากขึ้นเวลาที่ใช้ในการเข้ารหัส-ถอดรหัสก็จะมากขึ้นด้วย ทั้งอัลกอริทึม DES และ RSA

2. อัลกอริทึม RSA ใช้เวลาในการเข้ารหัส-ถอดรหัส มากกว่า อัลกอริทึมDES

จากตารางการทดลองที่ 8, 9, 10 และ 11 สามารถสรุปได้ว่า ในจำนวนข้อมูลที่เท่ากันอัลกอริทึม RSA จะใช้เวลาในการเข้ารหัส-ถอดรหัส มากกว่า อัลกอริทึม DES เสมอ

3. จำนวนกุญแจมีผลต่อเวลาในการเข้ารหัส-ถอดรหัส ด้วย อัลกอริทึม RSA

จากตารางการทดลองที่ 12 สามารถสรุปได้ว่า เมื่อขนาดของกุญแจเพิ่มขึ้น เวลาที่ใช้ในการเข้ารหัส-ถอดรหัส ด้วย RSA จะเพิ่มมากขึ้นด้วย

4. การถอดรหัสของอัลกอริทึม RSA จะใช้เวลามากกว่าการเข้ารหัส

จากตารางการทดลองที่ 12 สามารถสรุปได้ว่า เวลาที่ใช้ในการถอดรหัสของอัลกอริทึม RSA จะใช้เวลามากกว่า การเข้ารหัส และเมื่อเพิ่มขนาดของกุญแจมากขึ้นเวลาในการถอดรหัสก็จะยิ่งเพิ่มมากขึ้น ซึ่งแนวโน้มลักษณะของกราฟเป็น เอกซ์โปเนนเชียล

บทที่ 5
สรุป

5.1 สรุปโครงการพัฒนาระบบงาน

การศึกษาวิจัยที่ผ่านมา พบว่าการเข้ารหัส-ถอดรหัสด้วยอัลกอริทึม RSA และ DES มีความแตกต่างกันในด้านเวลา จากการทดลองทำให้ทราบว่าอัลกอริทึม DES เร็วกว่า RSA ดังนั้น การนำเอาอัลกอริทึมทั้งสองดังกล่าวไปใช้ จึงควรดูตามความเหมาะสม เพราะถึงแม้ว่าอัลกอริทึม RSA จะมีการทำงานที่ช้ากว่าแต่กุญแจที่ใช้ในการเข้ารหัสจะมีความปลอดภัย และเหมาะสมกับการใช้งานที่ต้องมีผู้ใช้ ใช้ร่วมกันเป็นจำนวนมาก ปัจจุบันการนำอัลกอริทึม DES ไปใช้งานมักจะเกี่ยวกับด้านการเงิน เช่น ธนาคาร ส่วนอัลกอริทึม RSA มักใช้ในการเข้ารหัสข้อมูลบนเครือข่ายอินเตอร์เน็ต เช่น LOTUS NOTES และได้มีการนำไปใช้งานในโปรแกรม Netscape Web Browser ที่ใช้งานแพร่หลายในปัจจุบัน

5.2 ข้อเสนอแนะ

- เนื่องจาก อัลกอริทึม RSA มีความปลอดภัยในการรับส่งกุญแจมากกว่า อัลกอริทึม DES แต่ DES มีการทำงานที่เร็วกว่า ดังนั้นในการส่งข้อมูลไปยังผู้รับอย่างสะดวก ปลอดภัยและ รวดเร็ว จึงควรนำข้อดีของอัลกอริทึมทั้งสองที่กล่าวมาข้างต้น มาผสมผสานกัน คือ ทำการเข้ารหัสข้อมูลที่ต้องการส่งไปยังผู้รับด้วย อัลกอริทึม DES หลังจากนั้นนำกุญแจที่ใช้ในการเข้ารหัสของ DES เข้ารหัสด้วย Public-key อีกครั้งหนึ่ง กำหนดให้เป็น กุญแจ x (Public-key และ Private-key สร้างจากอัลกอริทึม RSA) ผู้รับจะได้รับ

1. ข้อมูลที่ถูกเข้ารหัสด้วยอัลกอริทึม DES
2. กุญแจ x

เมื่อถึงผู้รับ ผู้รับจะต้องถอดรหัส กุญแจ x ด้วย Private-key จะได้กุญแจของ DES แล้วนำกุญแจDES นี้ไปถอดรหัส ข้อมูลที่ได้รับ ผู้รับจะได้ข้อมูลที่ส่งมาจากผู้ส่งอย่างถูกต้องและปลอดภัย

- จากความสามารถในการรักษาความปลอดภัยของจาวาแอปเพล็ต ทำให้เกิดข้อจำกัดในการพัฒนาโปรแกรมคือ ต้องใช้ แอปเพล็ตวิวเวอร์เป็นบราวเซอร์เท่านั้นถึงจะทำการเก็บข้อมูลลง local file system และแอปเพล็ตยังมีข้อจำกัดในการเก็บข้อมูลที่เซอร์ฟเวอร์ ดังนั้นโปรแกรมที่พัฒนาขึ้นจึงไม่สามารถส่งข้อมูลโดยใช้ โปรโตคอล FTP ได้ ทำให้การทำงานมีความล่าช้ามากขึ้น

บรรณานุกรม

[1] Deitel,H.M. and P.J. Java How to Program . New Jersey: Prentice-Hall,1997.

[2] Pfleeger,Charles P. Security in computing, Pentice-Hall International,1989.

[3] Man Young, Rhee . Cryptography and secure Communications, McGraw-Hill,1940.

[4]Stinson, Douglas R. Cryptography Theory and Practice,CRC Press,1995.

[5] Warwick, Ford. Computer Communications Security, Prentice-Hall International.1994.

ประวัติผู้เขียน

ชื่อผู้เขียน	น.ส. วรรวิภา ท่าพระนา
วันเดือนปีเกิด	2 มิถุนายน 2518
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษาระดับปริญญาตรี	วท.บ.(คณิตศาสตร์)
สถานที่สำเร็จการศึกษา	คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2535