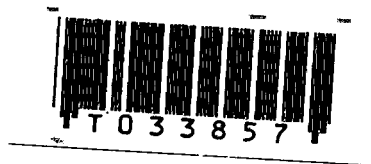


การเปรียบเทียบระบบรหัสลับ  
แบบ DES และแบบ RSA



|              |               |          |
|--------------|---------------|----------|
| นางสาววนิดา  | ดิเรกสุนทร    | 37054136 |
| นายวีระวัฒน์ | ฐานะประสิทธิ์ | 37054141 |
| นายสมชาย     | ฉัฐพรณรัตน์   | 37054145 |

เลขหมู่.....  
เลขทะเบียน...33857...  
วัน, เดือน, ปี 17 ก.ย. 2542

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

# **DES Crytosystem compare with RSA**

## **Cryptosystem Implementation**

|                    |                     |                 |
|--------------------|---------------------|-----------------|
| <b>Miss Wanida</b> | <b>Direksuntorn</b> | <b>37054136</b> |
| <b>Mr. Werawat</b> | <b>Tanaprasit</b>   | <b>37054141</b> |
| <b>Mr. Somchai</b> | <b>Chatpanarat</b>  | <b>37054145</b> |

**A Special Project Submitted in Partial Fulfillment of the  
Requirement for the Degree of Bachelor of Science  
Department of Mathematics and Computer Sciences  
Faculty of Science  
King Mongkut's Institute of Technology Ladkrabang**

**1997**

**หัวข้อปัญหาพิเศษ**

การเปรียบเทียบระบบรหัสลับแบบ DES และแบบ RSA  
DES Crytosystem compare with RSA Cryptosystem  
Implementation

**โดย**

นางสาววนิดา ดิเรกสุนทร 37054136

นายวีระวัฒน์ ฐานะประสิทธิ์ 37054141

นายสมชาย ถัฐพรรณรัตน์ 37054145

**ภาควิชา**

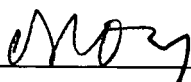
คณิตศาสตร์และวิทยาการคอมพิวเตอร์

**อาจารย์ที่ปรึกษา**

รศ. ภักคินี ชิตสกุล

อ. วิรัตน์ ศิริมังคลานุรักษ์

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นำปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ประยุกต์ ประจำปีการศึกษา 2540



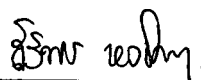
(รศ. ภักคินี ชิตสกุล)

หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

**คณะกรรมการปัญหาพิเศษ**

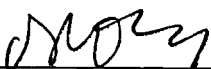

(ผศ. พิชรินทร์ เหมโชติ)

ประธานกรรมการ



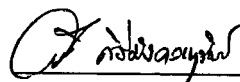
(อ. สิริพร หงษ์ทอง)

กรรมการ



(รศ. ภักคินี ชิตสกุล)

กรรมการและอาจารย์ที่ปรึกษา



(อ. วิรัตน์ ศิริมังคลานุรักษ์)

กรรมการและอาจารย์ที่ปรึกษา

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

|                  |                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------|
| หัวข้อปัญหาพิเศษ | การเปรียบเทียบระบบรหัสลับแบบ DES และแบบ RSA                                                              |
| โดย              | นางสาววนิดา ดิเรกสุนทร 37054136<br>นายวีระวัฒน์ ฐานะประสิทธิ์ 37054141<br>นายสมชาย ธีฐพรรณรัตน์ 37054145 |
| อาจารย์ที่ปรึกษา | รศ. ภักคินี ชิตสกุล<br>อ.วิรัตน์ ศิริมังคลานุรักษ์                                                       |
| ภาควิชา          | คณิตศาสตร์และวิทยาการคอมพิวเตอร์                                                                         |
| ปีการศึกษา       | 2540                                                                                                     |

### บทคัดย่อ

ในปัจจุบัน ได้มีการใช้คอมพิวเตอร์ในการจัดการข้อมูลและข่าวสารกันอย่างแพร่หลาย ในทุกวงการทั้งในด้านการจัดเก็บและการจัดส่ง ในบางครั้ง ข้อมูลนั้นเป็นข้อมูลลับ จึงมีความจำเป็นที่จะต้องใช้ซอฟต์แวร์รหัสลับ ในการรักษาความปลอดภัยของข้อมูล ในการทำปัญหาพิเศษนี้ มีจุดมุ่งหมาย คือ เพื่อจัดสร้างซอฟต์แวร์ที่ใช้ในการเปรียบเทียบระบบรหัสลับแบบ DES และ RSA เพื่อทำการเปรียบเทียบให้ผู้ใช้เห็นความแตกต่างของระบบรหัสลับทั้งสองแบบ และจะได้เลือกนำระบบรหัสลับไปใช้ให้เหมาะสมกับรูปแบบของงานในการจัดเก็บ หรือจัดส่งข้อมูลที่เป็นความลับได้

|                              |                                                                               |
|------------------------------|-------------------------------------------------------------------------------|
| <b>Special Project Title</b> | DES Crytosystem compare with RSA<br>Cryptosystem Implementation               |
| <b>Name</b>                  | Miss Wanida Direksuntorn<br>Mr. Werawat Tanaprasit<br>Mr. Somchai Chatpanarat |
| <b>Advisor</b>               | Mrs.Pakkinee Chitsakul<br>Mr. Wirat Sirimangklanurak                          |
| <b>Department</b>            | Mathematics and Computer Science                                              |
| <b>Year</b>                  | 1997                                                                          |

### **Abstract**

In this time , A computer which is used for manage data and information is widespread in every coterie , both storage and transmission. Sometime ,the data is a secret ,so it necessitates to use cryptosystem software. This study aims at make a software which is a DES crytosystem compare with RSA cryptosystem Implementation for make a user can see differential of two cryptosystems and choose a suitable cryptosystem with type of work , either storage or transmission data.

## กิตติกรรมประกาศ

ปัญหาพิเศษฉบับนี้สำเร็จลงได้ด้วยดีก็เพราะหลายเหตุปัจจัย โดยเฉพาะอย่างยิ่ง

รศ. ภัคคินี ชิตสกุล

อ. วิรัตน์ สิริมงคลานุรักษ์

ที่ได้ให้แนวทางในการดำเนินการ ตลอดจนคำปรึกษาอันก่อให้เกิดแนวความคิดที่สามารถแก้ไขปัญหาดัง ๆ ที่เกิดขึ้นในระหว่างการทำปัญหาพิเศษฉบับนี้ นอกจากนี้ยังช่วยแนะนำแนวทางในการดำเนินงานและตรวจทานแก้ไขด้วยความเอาใจใส่เป็นอย่างดี

ขอขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ที่ให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์ และให้ความสะดวกในการเบิกอุปกรณ์ต่างๆ ที่ใช้ในการจัดทำปัญหาพิเศษ

คณะผู้จัดทำขอกราบขอบพระคุณอาจารย์ทุกท่านที่ประสานวิชาความรู้ทั้งในภาคทฤษฎีและภาคปฏิบัติแก่ผู้จัดทำ จนกระทั่งปัญหาพิเศษฉบับนี้สัมฤทธิ์ผลได้ด้วยดีทุกประการ

คณะผู้จัดทำ

# สารบัญ

|                                                     | หน้า |
|-----------------------------------------------------|------|
| หน้าอนุมัติ                                         | i    |
| บทคัดย่อปัญหาพิเศษภาษาไทย                           | ii   |
| บทคัดย่อปัญหาพิเศษภาษาอังกฤษ                        | iii  |
| กิตติกรรมประกาศ                                     | iv   |
| สารบัญตาราง                                         |      |
| สารบัญรูป                                           |      |
| บทที่ 1 บทนำ                                        |      |
| • ความสำคัญและที่มาของปัญหาพิเศษ                    | 1    |
| • วัตถุประสงค์ของปัญหาพิเศษ                         | 1    |
| • ขอบเขตของปัญหาพิเศษ                               | 1    |
| • ประโยชน์ที่คาดว่าจะได้รับ                         | 1    |
| • อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ                    | 2    |
| • การแบ่งงาน                                        | 2    |
| • งบประมาณที่คาดว่าจะใช้ในการทำปัญหาพิเศษ           | 3    |
| บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง                |      |
| • ทฤษฎีรหัสลับเบื้องต้น                             | 3    |
| • ทฤษฎีรหัสลับแบบ DES                               | 6    |
| - ตัวอย่างการเข้ารหัสแบบ DES                        | 15   |
| • ทฤษฎีรหัสลับแบบ RSA                               | 20   |
| - ตัวอย่างการเข้ารหัสแบบ RSA                        | 22   |
| • ทฤษฎีทางคณิตศาสตร์ที่เกี่ยวข้อง                   | 24   |
| • ความปลอดภัยของระบบรหัสลับ                         | 28   |
| - ความปลอดภัยของการเข้ารหัสโดยใช้ algorithm แบบ DES | 28   |
| - ความปลอดภัยของการเข้ารหัสโดยใช้ algorithm แบบ RSA | 29   |
| บทที่ 3 การออกแบบระบบ                               |      |
| • System Flow Diagram                               | 31   |
| • Flow Chart                                        | 32   |
| • หน้าจอของโปรแกรมที่ติดต่อกับผู้ใช้                | 42   |

|            |                               |    |
|------------|-------------------------------|----|
| บทที่ 4    | ผลการทำงานของระบบ             | 46 |
| บทที่ 5    | บทสรุปและข้อเสนอแนะ           | 52 |
| ภาคผนวก    |                               |    |
| ♦          | ภาคผนวก ก คำศัพท์             |    |
| ♦          | ภาคผนวก ข ตาราง               |    |
| ♦          | ภาคผนวก ค คู่มือการใช้โปรแกรม |    |
| บรรณานุกรม |                               |    |

# สารบัญตาราง

|                                                                                                | หน้า |
|------------------------------------------------------------------------------------------------|------|
| ตารางที่ 2-1 แสดงเวลาโดยประมาณที่ใช้ในการแกะรหัส เมื่อใช้ความยาวของ key และค่าใช้จ่ยที่ต่างกัน | 29   |
| ตารางที่ 2-2 แสดงจำนวนหลักที่สามารถที่จะแยกตัวประกอบออกได้ และเวลาที่ต้องใช้ในการแยกตัวประกอบ  | 30   |
| ตารางแสดงการสลับบิต แบบ Initial Permutation (IP)                                               | ข-1  |
| ตารางแสดงการสลับบิต แบบ PC-1                                                                   | ข-2  |
| ตารางแสดงการสลับบิต แบบ PC-2                                                                   | ข-3  |
| ตารางแสดงการสลับบิต key แบบ $K_1$                                                              | ข-4  |
| ตารางแสดงการสลับบิต key แบบ $K_2$                                                              | ข-5  |
| ตารางแสดงการสลับบิต key แบบ $K_3$                                                              | ข-6  |
| ตารางแสดงการสลับบิต key แบบ $K_4$                                                              | ข-7  |
| ตารางแสดงการสลับบิต key แบบ $K_5$                                                              | ข-8  |
| ตารางแสดงการสลับบิต key แบบ $K_6$                                                              | ข-9  |
| ตารางแสดงการสลับบิต key แบบ $K_7$                                                              | ข-10 |
| ตารางแสดงการสลับบิต key แบบ $K_8$                                                              | ข-11 |
| ตารางแสดงการสลับบิต key แบบ $K_9$                                                              | ข-12 |
| ตารางแสดงการสลับบิต key แบบ $K_{10}$                                                           | ข-13 |
| ตารางแสดงการสลับบิต key แบบ $K_{11}$                                                           | ข-14 |
| ตารางแสดงการสลับบิต key แบบ $K_{12}$                                                           | ข-15 |
| ตารางแสดงการสลับบิต key แบบ $K_{13}$                                                           | ข-16 |
| ตารางแสดงการสลับบิต key แบบ $K_{14}$                                                           | ข-17 |
| ตารางแสดงการสลับบิต key แบบ $K_{15}$                                                           | ข-18 |
| ตารางแสดงการสลับบิต key แบบ $K_{16}$                                                           | ข-19 |
| ตารางแสดงการสลับบิต แบบ Expansion (E)                                                          | ข-20 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 1 (S-box 1)                                         | ข-21 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 2 (S-box 2)                                         | ข-22 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 3 (S-box 3)                                         | ข-23 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 4 (S-box 4)                                         | ข-24 |

|                                                                   |      |
|-------------------------------------------------------------------|------|
| ตารางแสดงการสลับบิต แบบ Substitution boxes 5 (S-box 5)            | ข-25 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 6 (S-box 6)            | ข-26 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 7 (S-box 7)            | ข-27 |
| ตารางแสดงการสลับบิต แบบ Substitution boxes 8 (S-box 8)            | ข-28 |
| ตารางแสดงการสลับบิต แบบ Permutation (P)                           | ข-29 |
| ตารางแสดงการสลับบิต แบบ Inverse Initial Permutation ( $IP^{-1}$ ) | ข-30 |

# สารบัญรูป

|                                                                                                    | หน้า |
|----------------------------------------------------------------------------------------------------|------|
| รูปที่ 2-1 แสดงการเข้ารหัสและถอดรหัสแบบ symmetric-key encryption                                   | 4    |
| รูปที่ 2-2 แสดงการเข้ารหัสและถอดรหัสแบบ Public key encryption                                      | 5    |
| รูปที่ 2-3 แสดงขั้นตอนการเข้ารหัสแบบ DES                                                           | 8    |
| รูปที่ 2-4 แสดง DES Algorithm : Overview                                                           | 9    |
| รูปที่ 2-5 แสดง DES One Round                                                                      | 10   |
| รูปที่ 2-6 แสดง DES Key schedule                                                                   | 12   |
| รูปที่ 3-1 แสดง System Flow Diagram ของระบบ                                                        | 31   |
| รูปที่ 3-2 แสดง Flow Chart ของระบบ                                                                 | 32   |
| รูปที่ 3-2 แสดง Flow Chart ของการเข้ารหัส                                                          | 33   |
| รูปที่ 3-3 แสดง Flow Chart ของการเข้ารหัสด้วย Algorithm แบบ DES                                    | 34   |
| รูปที่ 3-4 แสดง Flow Chart ของการเข้ารหัสด้วย Algorithm แบบ RSA                                    | 35   |
| รูปที่ 3-5 แสดง Flow Chart ของการถอดรหัส                                                           | 36   |
| รูปที่ 3-6 แสดง Flow Chart ของการถอดรหัสด้วย Algorithm แบบ DES                                     | 37   |
| รูปที่ 3-7 แสดง Flow Chart ของการถอดรหัสด้วย Algorithm แบบ RSA                                     | 38   |
| รูปที่ 3-8 แสดง Flow Chart ของการเปรียบเทียบการเข้ารหัสถอดรหัสโดย<br>ใช้ algorithm แบบ DES และ RSA | 39   |
| รูปที่ 3-9 แสดงหน้าจอเริ่มแรกเมื่อเข้าสู่โปรแกรม                                                   | 40   |
| รูปที่ 3-10 หน้าจอแสดงการเลือกไฟล์                                                                 | 40   |
| รูปที่ 3-11 แสดงหน้าจอการรับชื่อไฟล์ , key และ Encryption option<br>ที่ใช้ในการเข้ารหัสแบบ DES     | 41   |
| รูปที่ 3-12 แสดงหน้าจอการรับชื่อไฟล์ , key และ Encryption option<br>ที่ใช้ในการเข้ารหัสแบบ RSA     | 41   |
| รูปที่ 3-13 รูปที่ 3-13 แสดงหน้าจอการแสดง Public key , Private key<br>และค่า n ของ RSA             | 42   |
| รูปที่ 3-14 แสดงหน้าจอการรับชื่อไฟล์ , key และ Decryption option<br>ที่ใช้ในการถอดรหัสแบบ DES      | 42   |
| รูปที่ 3-15 แสดงหน้าจอการรับชื่อไฟล์ , key และ Decryption option<br>ที่ใช้ในการถอดรหัสแบบ RSA      | 43   |

|                                                                                                          |     |
|----------------------------------------------------------------------------------------------------------|-----|
| รูปที่ 3-16 แสดงหน้าจอแสดงกราฟของการเปรียบเทียบขนาดไฟล์ต่อเวลา<br>ในการเข้ารหัสและถอดรหัสแบบ DES และ RSA | 44  |
| รูปที่ 3-17 แสดงหน้าจอแสดงกราฟของการเปรียบเทียบการกระจายตัว<br>อักษรของการเข้ารหัสแบบ DES และ RSA        | 45  |
| รูปที่ 4-1 แสดงข้อความในไฟล์ก่อนที่จะถูกทำการเข้ารหัส                                                    | 47  |
| รูปที่ 4-2 แสดงข้อความในไฟล์หลังจากถูกทำการเข้ารหัสโดยใช้<br>algorithm แบบ RSA                           | 47  |
| รูปที่ 4-3 แสดงข้อความในไฟล์หลังจากที่ทำการถอดรหัสแล้ว                                                   | 48  |
| รูปที่ 4-4 แสดงขนาดต่อเวลาในการเข้ารหัสแบบ RSA                                                           | 49  |
| รูปที่ ก-1 แสดงหน้าจอเริ่มแรกของการเข้าสู่โปรแกรม                                                        | ก-1 |
| รูปที่ ก-2 แสดงการหน้าจอที่ใช้เลือกไฟล์ที่จะทำการเข้ารหัสหรือถอดรหัส                                     | ก-2 |
| รูปที่ ก-3 แสดงหน้าจอการเข้ารหัสแบบ DES                                                                  | ก-3 |
| รูปที่ ก-4 แสดงหน้าจอการเข้ารหัสแบบ RSA                                                                  | ก-4 |
| รูปที่ ก-5 แสดงหน้าจอการแสดง Public Key ,n และ Private Key แก่ผู้ใช้                                     | ก-5 |
| รูปที่ ก-6 แสดงหน้าจอการถอดรหัสแบบ DES                                                                   | ก-6 |
| รูปที่ ก-7 แสดงหน้าจอการถอดรหัสแบบ RSA                                                                   | ก-7 |

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญ/ที่มาของปัญหาพิเศษ

ในปัจจุบัน ได้มีการใช้คอมพิวเตอร์ในการจัดการข้อมูลและข่าวสารกันอย่างแพร่หลาย ในทุกวงการทั้งในด้านการจัดเก็บและการจัดส่ง ในบางครั้ง ข้อมูลนั้นเป็นข้อมูลลับ ซึ่งการจัดการทำให้ข้อมูลให้เป็นความลับนั้นมีทฤษฎีประยุกต์ทางด้านคณิตศาสตร์จำนวนมาก แต่อย่างไรก็ตาม การสร้างระบบรหัสลับตามทฤษฎีในประเทศไทยค่อนข้างจำกัด ปัญหาพิเศษนี้ จึงได้สร้างระบบรหัสลับ ในรูปของซอฟต์แวร์ที่คาดว่าจะสามารถที่จะนำไปปรับปรุงเพื่อใช้กับงานข้อมูลได้อย่างแพร่หลายต่อไป

### 1.2 วัตถุประสงค์ของปัญหาพิเศษ

เพื่อศึกษาทฤษฎีรหัสลับ และจัดสร้างระบบรหัสลับทั้งแบบ DES และ RSA เพื่อนำไปใช้ในการรักษาความปลอดภัยของข้อมูล ด้านการจัดเก็บและการจัดส่ง และทำการเปรียบเทียบระบบรหัสลับแบบ DES และ RSA ในด้านเวลาที่ใช้ในการเข้ารหัสและถอดรหัส

### 1.3 ขอบเขตของปัญหาพิเศษ

เป็นโปรแกรมที่ใช้สำหรับการเข้ารหัสและถอดรหัสข้อมูลที่เป็นตัวอักษรภาษาอังกฤษ (text file) เท่านั้น โดย Algorithm ที่ใช้ในระบบรหัสลับ คือ DES และ RSA

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ในด้านผู้ใช้งานโปรแกรม

สามารถนำระบบรหัสลับไปใช้งานเพื่อรักษาความปลอดภัยของข้อมูลในด้านการจัดเก็บและการจัดส่ง และสามารถเลือกวิธีการเข้ารหัสลับที่เหมาะสมกับงานได้

ประโยชน์ในด้านผู้จัดทำ

1. ศึกษาการใช้ภาษาซี และวิธีการเข้ารหัสและถอดรหัสแบบ DES และแบบ RSA

2. ศึกษาการออกแบบและการพัฒนาระบบงาน
3. ศึกษาการทำ interface ด้วย C++ Builder

## 1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

1. เครื่องคอมพิวเตอร์ Pentium 100 ขึ้นไป
2. หน่วยความจำอย่างน้อย 16 MB
3. Harddisk อย่างน้อย 1 GB
4. Software โปรแกรม Borland C ++
5. Software โปรแกรม C ++ Builder

## 1.6 การแบ่งงาน

แบ่งระบบงานออกเป็น 2 ส่วนใหญ่ ๆ คือ ศึกษาทฤษฎีรหัสลับแบบ DES และศึกษาการใช้ software

1. ศึกษาทฤษฎีรหัสลับ
  - ศึกษาการทฤษฎีรหัสลับแบบ DES
  - ศึกษาการทฤษฎีรหัสลับแบบ RSA
2. ศึกษาการใช้ software
  - ศึกษาการเขียนโปรแกรมด้วย Borland C++
  - ศึกษาการทำ interface ด้วย C ++ Builder

## 1.7 งบประมาณที่ใช้ในการทำปัญหาพิเศษ

1. diskette เพื่อใช้ในการจัดเก็บ Source program 500 บาท
2. หนังสือและเอกสารอ้างอิงสำหรับใช้ค้นคว้า 2,000 บาท

## บทที่ 2

# ทฤษฎีและหลักเกณฑ์ที่เกี่ยวข้อง

## 2.1 ทฤษฎีรหัสลับเบื้องต้น

### Cryptosystem

ระบบรหัสลับ จะประกอบด้วยขั้นตอนสำหรับเข้ารหัส และการถอดรหัส ระบบการเข้ารหัสที่ใช้กันอยู่จะประกอบด้วย ข้อความที่ต้องการเข้ารหัส(plaintext) กุญแจข้อความที่เข้ารหัส(key) และฟังก์ชันที่ใช้ในการเข้ารหัสและถอดรหัส

### Encryption (การเข้ารหัส)

การเข้ารหัส คือ วิธีการรักษาความปลอดภัยสำหรับข่าวสารที่เป็นความลับ มันเป็นกระบวนการทางคณิตศาสตร์ที่ทำการผสมกันระหว่างบิตของข้อมูลกับ Password-key การเข้ารหัสเป็นการแปลงข้อมูลให้ไม่สามารถอ่านได้จนกระทั่งได้ทำการถอดรหัส

- การเข้ารหัส เป็นการทำให้ข้อมูลให้เป็นรูปแบบที่อ่านไม่ได้ จุดประสงค์ของมัน คือทำให้แน่ใจว่าได้รับความลับโดยการปกปิดข้อมูลจากผู้ที่ไม่ควรได้อ่าน ให้เห็นเพียงข่าวสารที่เข้ารหัสแล้ว การถอดรหัส (Decryption) จะเป็นการ reverse ของการเข้ารหัส มันเป็นการเปลี่ยนข้อมูลที่ถูกรหัสให้กลับไปในรูปแบบที่อ่านได้
- การเข้ารหัสและการถอดรหัสจะถูกใช้ในข่าวสารที่เป็นความลับ ปกติแล้วจะอ้างไปถึง Password-key การเข้ารหัสจะทำโดยใช้กระบวนการทางคณิตศาสตร์ในการแปลงข่าวสารและใช้ algorithm

เทคนิคการเข้ารหัสมี 2 ชนิด คือ symmetric-key encryption และ public-key encryption เทคนิคชนิดแรกจะถูกใช้ในการเข้ารหัสข่าวสารที่ถูกเก็บไว้ในดิสก์ ขณะที่ชนิดที่สองจะถูกใช้ในการเข้ารหัสทางการติดต่อและจดหมาย

### Encryption Algorithms

การเข้ารหัส จะทำงานโดยการเข้ารหัสไฟล์ข้อมูลด้วย password-key และแปลงข้อมูลโดยใช้หลักทางคณิตศาสตร์ที่เรียกว่า algorithm วิธีการที่เป็น algorithm ที่ใช้ในการแปลงข้อมูลจะเป็นตัวกำหนดความแข็งแกร่งและความปลอดภัยของข้อมูล

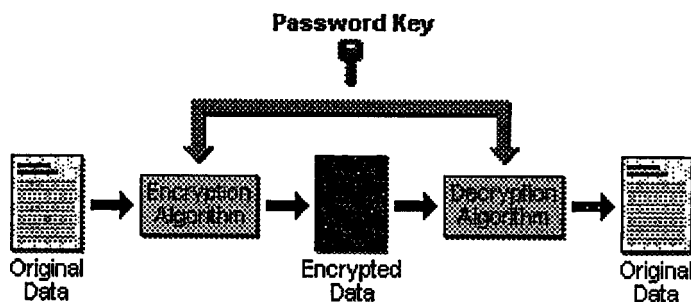
Algorithm เป็นหลักทางคณิตศาสตร์ที่นำ password-key ที่ถูกใช้ในการเข้ารหัสและถอดรหัสไฟล์ข้อมูลมารวมใน algorithm ด้วย

ในที่นี้ เราจะศึกษา Algorithm ของ DES ซึ่งเป็นการเข้ารหัสแบบ symmetric-key encryption และ Algorithm ของ RSA ซึ่งเป็นการเข้ารหัสแบบ public-key encryption

## Symmetric Key Encryption

ชนิดของการเข้ารหัสแบบ symmetric-key encryption ปกติ ส่วนใหญ่จะใช้ password-key เดียวในการเข้ารหัสและถอดรหัสข้อมูล

ในการใช้รูปแบบ symmetric-key encryption นั้น ข้อความปกติ (plaintext) จะถูกเข้ารหัส และมันจะปรากฏเป็นข้อความที่อ่านไม่เป็นภาษา และมันเป็นไปได้ที่จะเปลี่ยนให้กลับไปเป็นข้อความที่เหมือนในตอนแรกโดยปราศจาก password key เมื่อข้อความถูกเข้ารหัสแล้ว มันสามารถที่จะถูกเก็บในอุปกรณ์ media ที่ไม่ปลอดภัยและถูกส่งไปใน network ที่ไม่ปลอดภัยได้ โดยที่มันยังคงปลอดภัยอยู่ ต่อจากนั้น ข้อความนั้นจะถูกถอดรหัสให้อยู่ในรูปแบบเหมือนตอนแรก กระบวนการจะถูกแสดงดังรูป :



รูปที่ 2-1 แสดงการเข้ารหัสและถอดรหัสแบบ symmetric-key encryption

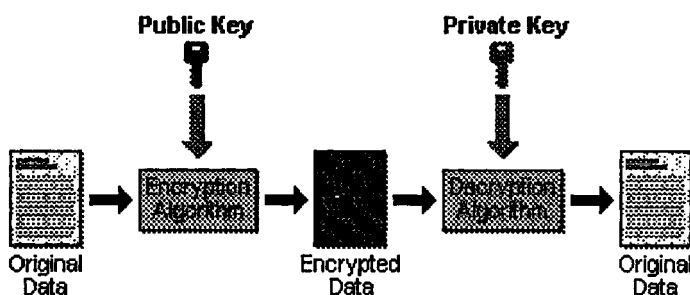
ชนิดการเข้ารหัสแบบ symmetric-key encryption นี้ ผู้ส่งและผู้รับข้อความจะต้อง share password key เดียวกัน และ password key จะต้องถูกเก็บรักษาไว้เป็นความลับเพราะว่าใครก็ตามที่พบข้อความนี้ สามารถถอดรหัสข้อความได้ถ้ามี key ชนิดการเข้ารหัสแบบ symmetric-key encryption จึงต้องการความเชื่อใจในการ share password ด้วย วิธีนี้ อาจจะใช้ในการจัดการทางการทหาร แต่จะไม่ใช้ในระบบการสื่อสารที่ไม่ปลอดภัยสำหรับการแลกเปลี่ยน password key ระหว่างผู้ใช้ที่มีอยู่เป็นล้านคน

Symmetric-key encryption นั้นเหมาะสำหรับกรณีที่ข้อมูลนั้นไม่ต้องถูก share สำหรับการสื่อสารและสำหรับการ share ไฟล์กับผู้อื่นควรจะใช้การเข้ารหัสชนิด public-key encryption

## Public Key Encryption

ในการได้ตอบปัญหาการ share ข้อมูลอย่างปลอดภัย ระบบที่รู้จักกันในนาม Public key encryption ได้ถูกพัฒนา ระบบนี้จะต่างจาก symmetric-key encryption ตรงที่จะไม่ใช่ key เดียว Public key encryption จะใช้ 2 key การเข้าคู่กันของ key คือจะใช้ public key ในการเข้ารหัสข้อมูลและจะใช้ private key ที่ตรงกันในการถอดรหัส

- เมื่อผู้ใช้ต้องการจะส่งไฟล์ที่เข้ารหัสแล้วไปให้ผู้ใช้คนอื่น ผู้ส่งจะต้องเข้ารหัสข้อมูลด้วย public key ซึ่ง private key เพียงอย่างเดียวที่สามารถจะทำการถอดรหัสกับไฟล์ได้ และ public key สามารถที่จะ share ให้กับทุกคนได้
- ในการถอดรหัสกับไฟล์ ผู้รับจะต้องใช้ private key ของเขาเอง เพื่อรักษาความปลอดภัย private key จะต้องถูกเก็บไว้เป็นความลับ



รูปที่ 2-2 แสดงการเข้ารหัสและถอดรหัสแบบ Public key encryption

Public key encryption นั้นจะปลอดภัยมากกว่า symmetric-key encryption แต่การทำงานจะช้ากว่ามาก

ตัวอย่าง : สมมติว่า Alice ต้องการส่ง e-mail ที่ถูกเข้ารหัสให้กับ Bob เธอจะต้องหา public key ของ Bob ใน public key directory แล้วใช้มันในการเข้ารหัสและส่งข้อมูลให้กับ Bob จากนั้น Bob จะต้องใช้ private key ของเขาในการถอดรหัสข้อมูลและอ่านมัน ไม่มีใครอื่นสามารถที่จะถอดรหัสข้อมูลนั้นได้ ทุกคนสามารถจะส่งข้อความที่เข้ารหัสไปให้กับ Bob ได้ แต่จะมี Bob เพียงคนเดียวที่สามารถอ่านมัน

## 2.2 ทฤษฎีรหัสลับแบบ DES

### The Algorithm

DES เป็น block algorithm ที่ใช้ blocks 64-bit และ key 56-bit algorithm ของ DES จะสมมาตรกัน หมายความว่า algorithm และ key จะถูกใช้สำหรับการเข้ารหัสและถอดรหัสเหมือนกัน มันเกิดจากการสลับบิต (permutation) ครั้งแรกและครั้งสุดท้าย และวน 16 รอบ

ปกติแล้ว key จะถูกเก็บเป็นเลข 64-bit ซึ่งทุก ๆ 8 bits จะเป็น parity bit ซึ่ง parity bit ถูกเติมในระหว่าง algorithm และ key 56-bit จะถูกใช้ในการสร้าง subkeys 48-bit ที่แตกต่างกัน 16 ตัวในแต่ละรอบ block 64-bit ที่ถูกเข้ารหัสจะแบ่งออกเป็น 2 ฝั่งเท่า ๆ กัน ฝั่งครึ่งขวาจะทำการ DES round 1 รอบ และผลลัพธ์จะกลายเป็นฝั่งครึ่งซ้ายใหม่ ฝั่งครึ่งซ้ายเก่ากลายเป็นฝั่งครึ่งขวาใหม่ และจะทำ DES round ในรอบถัดไป 1 รอบ ทำเช่นนี้ไปจนครบ 16 รอบ แต่ภายหลังรอบสุดท้าย ฝั่งครึ่งซ้ายและฝั่งครึ่งขวาจะไม่ต้องถูกสลับตำแหน่งกัน ดังนั้นผลลัพธ์ของรอบที่ 16 จะกลายเป็นผลลัพธ์ฝั่งครึ่งขวาสุดท้าย และผลลัพธ์รอบที่ 15 (ซึ่งจะกลายเป็นฝั่งครึ่งซ้ายของรอบที่ 16) จะเป็นฝั่งครึ่งซ้ายสุดท้าย

### DES Permutations

การใช้ DES ในการ permutation ข้อมูล จะโหลด block 64-bit ในแต่ละครั้ง แบบแผนซึ่งใช้โหลด bits ถูกแสดงไว้ใน Table Initial Permutation (Table IP) การ permutation IP ครั้งสุดท้ายเป็น inverse ของครั้งแรก (ตัวอย่างเช่น ในการ permutation ครั้งสุดท้าย bit ที่ 40 จะกลายเป็น bit ที่ 1 ส่วนในการ permutation ครั้งแรก bit ที่ 1 จะกลายเป็น bit ที่ 40)

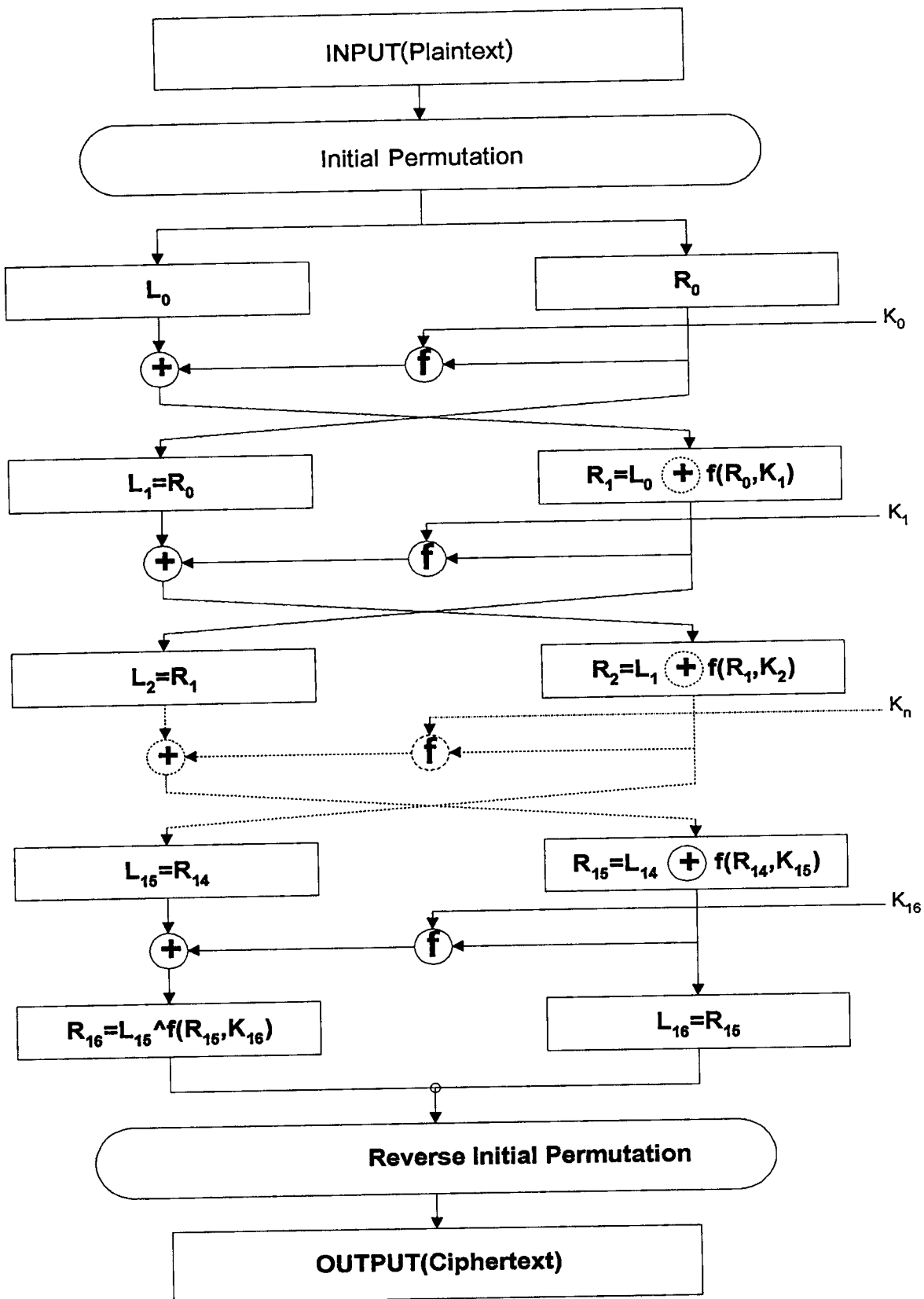
### DES Rounds

DES ใช้เทคนิคพื้นฐานของ cryptography 2 อย่าง คือ

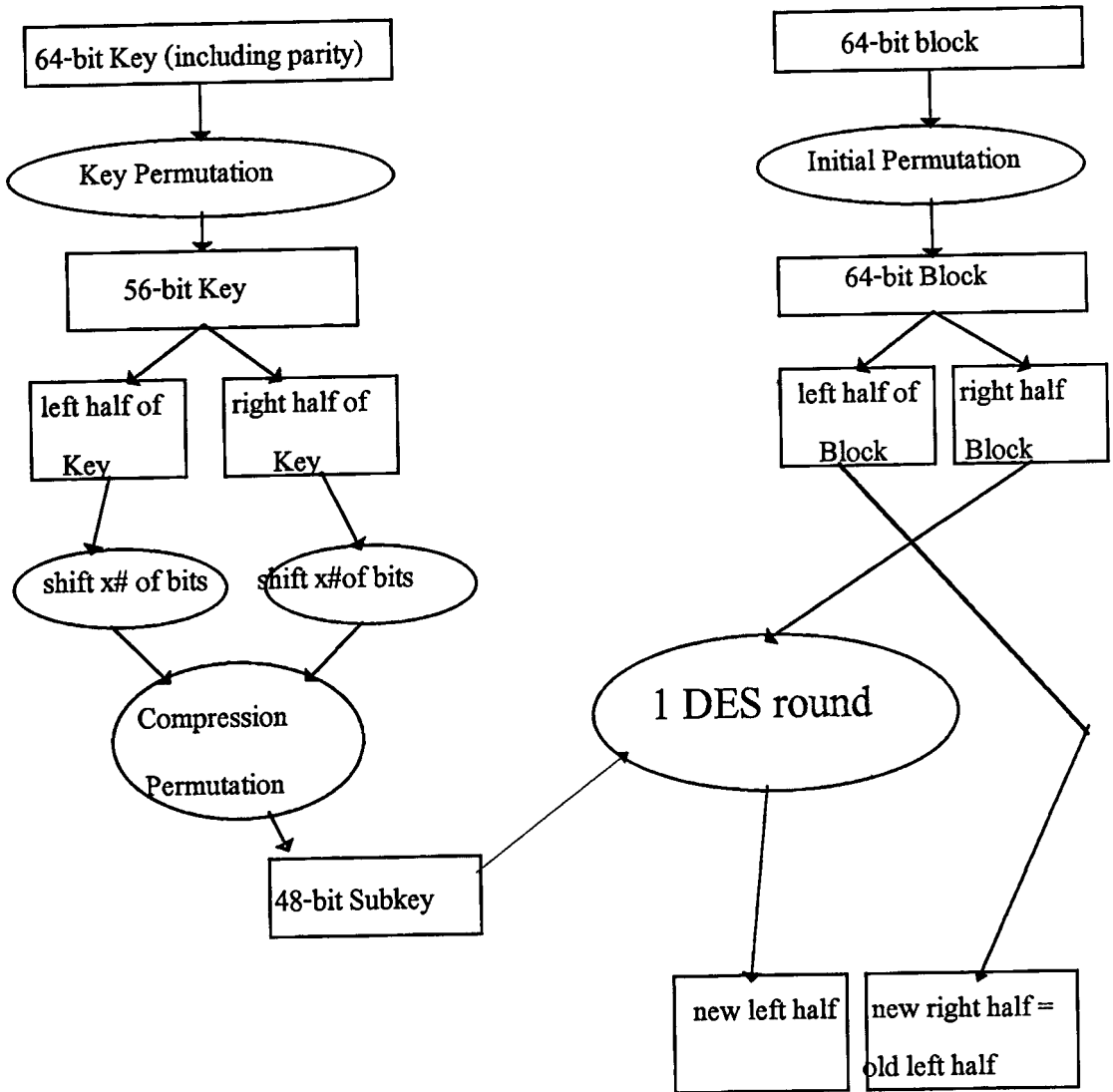
1. confusion (การปนกัน) ได้จากการ permutation หลาย ๆ ครั้ง
2. diffusion (การกระจาย) ได้จากการการทำ XOR operation

แต่ละ DES round จะใช้ subkey ที่แตกต่างกันซึ่งได้จาก key 56-bit ฝั่งครึ่งขวาของ block ข้อความจะถูกขยายจาก 32 bits เป็น 48 bits และถูก XOR ด้วย subkey ผลลัพธ์ของการทำ operation นี้ จะส่งผ่านไปยัง 8 substitution boxes หรือ S-boxes รูปแบบของ S-boxes นี้เป็นขั้นตอนเดียวที่ไม่เป็น linear ใน DES และเป็นการจัดการความปลอดภัยของมัน มันเป็นการบีบอัด

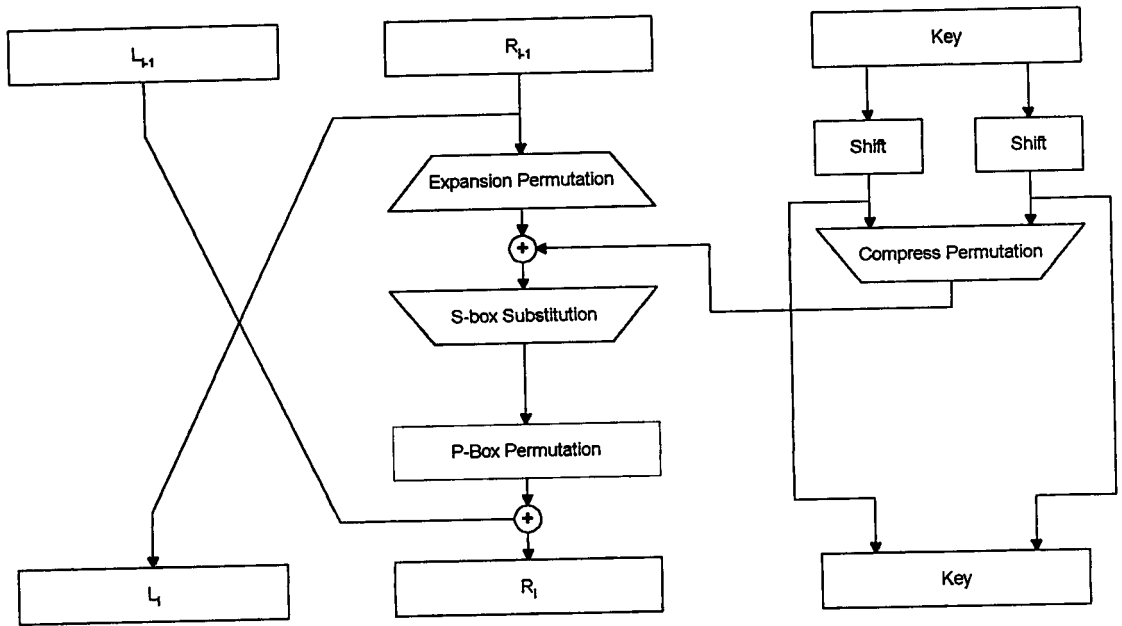
ผลลัพธ์ XOR operation ที่เป็น 48-bit กลับไปเป็น 32 bits ผลลัพธ์นี้เป็นการส่งผ่านไปยัง permutation box หรือ P-box และได้ผลลัพธ์อีกครั้งเป็น 32 bits และเกิดจากทุก ๆ bit ถูกใช้เป็น input สุดท้ายผลลัพธ์นี้จะถูก XOR ด้วยฟังก์ชันซ้ายของ block ข้อความ และฟังก์ชันซ้ายและฟังก์ชันขวาจะถูกสลับตำแหน่ง ดังนั้น ผลลัพธ์ของฟังก์ชันซ้ายและฟังก์ชันซ้ายเก่าจะเป็นฟังก์ชันขวา ปัจจุบัน ตัวเลขนี้จะถูกส่งไปยังรอบอื่น ๆ หลังจากรอบสุดท้าย อย่างไรก็ตามในรอบสุดท้ายฟังก์ชันซ้ายและฟังก์ชันขวาจะไม่ถูกสลับอีก



รูปที่ 2-3 แสดงขั้นตอนการเข้ารหัสแบบ DES



รูปที่ 2-4 แสดง DES Algorithm : Overview



รูปที่ 2-5 แสดง DES One Round

## Description

เป็นการนำแบบแผนของการเกิด 16 subkeys 48-bit จาก key 56 bit มาใช้ ตอนแรก key จะถูกโหลดเข้าไปตาม table แล้วแบ่งครึ่งเป็น 2 ฝั่ง แต่ละฝั่งจะถูก shift ไป 2 บิตในทุก ๆ รอบ ยกเว้นรอบแรก , รอบที่สอง , รอบที่เก้า และรอบสุดท้าย เหตุผลที่ทำเช่นนี้เพื่อสร้างความปลอดภัยจากการวิเคราะห์ความสัมพันธ์ของ key จากนั้นทำการ compression permutation เพื่อเลือก 48 บิตจาก 56 บิต

block ข้อความที่ถูกเข้ารหัสจะเรียกว่า x จะถูกแบ่งครึ่งเป็น 2 ฝั่ง ฝั่งขวาว (R) และ ฝั่งซ้าย (L) ตอนแรก ฝั่งขวาวจะผ่านการ expansion permutation ซึ่งจะมีการขยายจาก 32 บิตเป็น 48 บิต ตาม Table E (แสดงไว้ในภาคผนวกตารางที่ ข-20) ทำเช่นนี้นอกจากจะเพื่อให้มันมีความยาวเท่ากับ subkey เพื่อที่จะทำการ XOR แล้ว แต่มันยังเป็นหลักการสำคัญของวิซาร์หัสลับอีกด้วย ในการขยายขนาดถึง 16 ครั้ง บางบิตจะถูกใช้ซ้ำ (ไม่มีการนำบิตใหม่เข้ามาใช้ ทุก ๆ บิตที่มี จะถูก shift วนกลับ และถูกใช้ซ้ำเป็นครั้งที่ 2) จุดมุ่งหมายคือทำให้ทุก ๆ บิตที่เป็น output ใน DES จะขึ้นกับทุก ๆ บิตที่เป็น input

ผลลัพธ์ของการ expansion permutation เป็นการถูก XOR ด้วย subkey แล้วผ่าน S-boxes ซึ่งมี 8 S-boxes แต่ละ S-boxes จะเอา 6-bit ที่เป็น input ทำเป็น 4 บิตที่เป็น output ขึ้นตอนนี้ไม่เป็น linear สำหรับ input  $i_1, i_2, \dots, i_6$  output จะถูกกำหนดโดยการใช้ความเกี่ยวโยง (concatenation) ของ  $i_1$  และ  $i_6$  และ การความเกี่ยวโยงของ  $i_2 - i_5$  และการทำแบบนี้เป็นเหมือน

กับตัวชี้ table ของ S-box output ของแต่ละ 8 S-boxes เป็นการเกี่ยวโยงไปสู่รูปแบบของเลข 32 bit ซึ่งจะถูก permutate กับ P-box เป็นการ permutation แบบตรง ๆ และเลขผลลัพธ์ที่ได้จะถูก XOR กับฟังก์ชันซ้ายของ input block ซึ่งถูกเรากำหนดไว้ในตอนแรกของรอบนี้ สุดท้าย ถ้ารอบที่ทำไม่ใช่รอบสุดท้าย เราจะสลับฟังก์ชันซ้ายและฟังก์ชันขวา และเริ่มต้นใหม่อีกครั้ง

รอบสุดท้ายที่ทำ จะทำการสลับบิตตาม inverse ของ table IP (แสดงดัง Table IP<sup>-1</sup> ตารางที่ ข-30) ผลลัพธ์ที่ได้ คือ ข้อความที่เข้ารหัสแล้ว(ciphertext) หรือเรียกว่า y

การถอดรหัสจะใช้ algorithm เดียวกับการเข้ารหัส เริ่มด้วย y เป็นเหมือน output แต่จะใช้ key schedule  $K_{16}, \dots, K_1$  เป็นการกลับลำดับ output จะเป็นข้อความปกติ คือ x

## Key Schedule

เมื่อนำ key ไปทำการ permutation ตาม table PC-1 เราเขียนแทนได้ว่า  $PC-1(K) = C_0D_0$  ซึ่ง  $C_0$  แทน 28 บิตแรกของ  $PC-1(K)$  และ  $D_0$  แทน 28 บิตหลังของ  $PC-1$

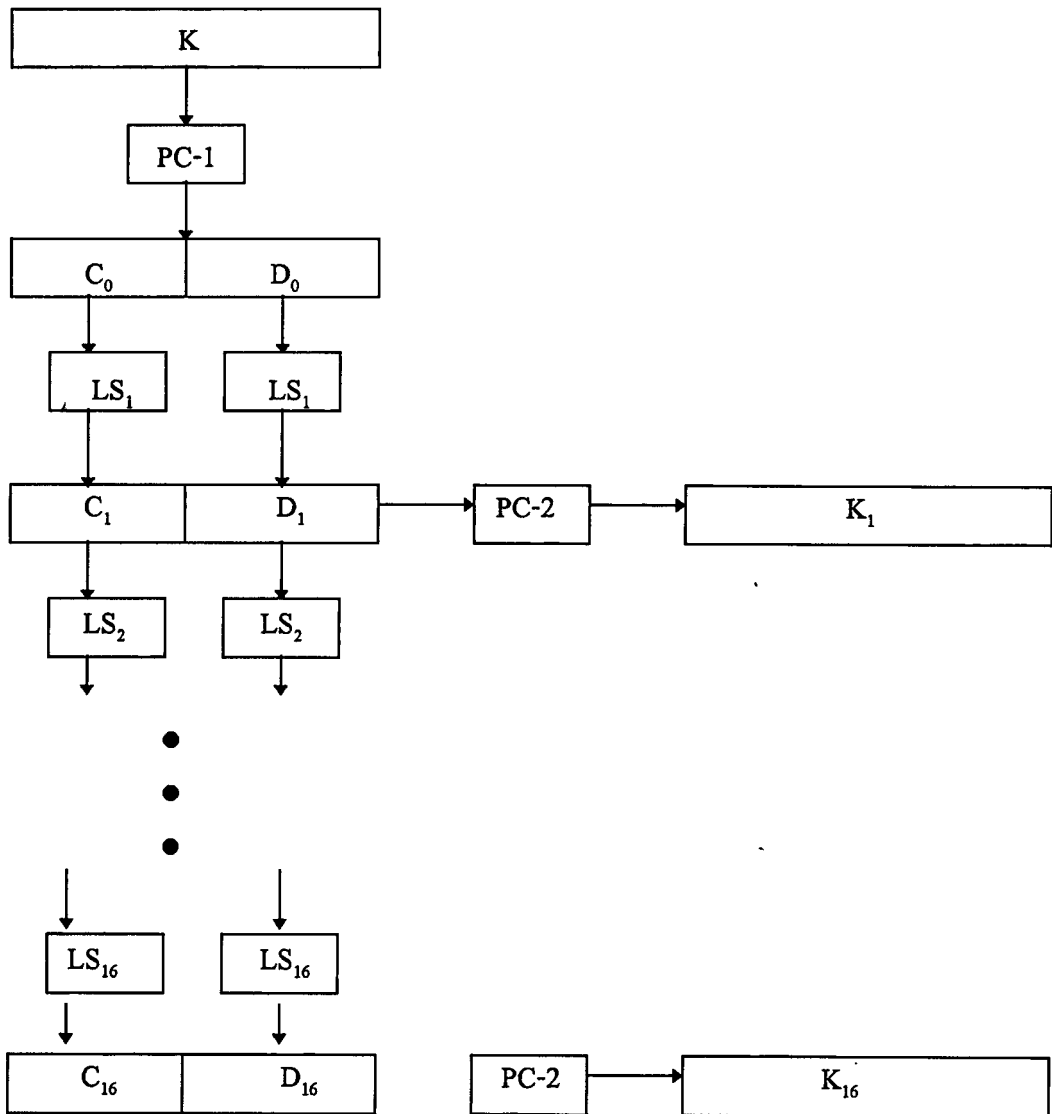
ให้  $I$  มีค่าตั้งแต่ 1 ถึง 16 เราคำนวณ

$$C_i = Ls_i(C_{i-1})$$

$$D_i = Ls_i(D_{i-1})$$

และ  $K_i = PC-2(C_iD_i)$ ,  $Ls_i$  แทนการ shift ซ้ายวนกลับ 1 หรือ 2 บิต ขึ้นกับค่าของ  $I$  คือ จะ shift 1 บิต ถ้า  $I = 1, 2, 9, 16$  นอกจากนั้นจะ shift 2 บิต ส่วน  $PC-2$  จะเป็นการ permutation แบบอื่น

Key schedule แสดงดังรูปที่ 2.6



รูปที่ 2-6 แสดง DES Key schedule

ตอนนี้ เราจะได้ Key schedule ที่เป็นผลลัพธ์แต่ละรอบ โดยใช้ 48 บิตที่เลือกมา ซึ่งเป็นไปตาม Table  $K_1$  ถึง  $K_{16}$  แสดงดังตารางที่ ข-4 ถึง ข-19

### DES : S-box Permutation

S-boxes ค่อนข้างที่จะแตกต่างจากการ permutation แบบอื่น ๆ ขณะที่แบบอื่นจะเกิดขึ้นจาก บิตที่  $x$  ไปเป็นบิตที่  $y$  บิตที่เป็น input สามารถมีจุดหมายแตกต่างสำหรับ S-boxes ถ้า input คือ  $\{d1, d2, d3, d4, d5, d6\}$  แล้วเลข 2 บิต  $\{d1, d6\}$  และเลข 4 บิต  $\{d2, d3, d4, d5\}$  จะถูกใช้เป็นตัวชี้ table S box สำหรับ 48-bit (word  $\{d1, d2, \dots, d48\}$ ) โดย word  $\{d1, \dots, d6\}$  จะส่งไปให้ S-box 1 word  $\{d7, \dots, d12\}$  จะส่งไปให้ S-box 2, ..., word  $\{d43, \dots, d48\}$  จะส่งไปให้ S-box 8

การ permute S-boxes จะเป็นการสลับบิตตาม Table S-boxes 1-8 ที่แสดงไว้ในภาคผนวกตารางที่ ข-21 ถึง ข-28

## DES : P-box Permutation

P-box Permutation เป็นการ Permutation โดยตรง 32 บิต ทุก ๆ บิต จะถูกสลับที่ และทุก ๆ 32 บิตที่ได้จากการ permutation เป็นไปตาม Table P-box ที่แสดงไว้ในภาคผนวกตารางที่ ข-29

## DES : Weak

Key ที่เป็น weak key เมื่อนำไป generate เป็น 16 Subkeys จะทำให้ได้ SubKeys ที่เหมือนกันในทุกรอบ ดังนั้น จึงทำให้การเข้ารหัสมีความปลอดภัยต่ำ

Keys ซึ่งถูกพิจารณาเป็น weak key ได้แก่ (ฐานสิบหก)

- 0000000 0000000
- 0000000 FFFFFFFF
- FFFFFFFF 0000000
- FFFFFFFF FFFFFFFF

Key ที่เป็น Semiweak key เมื่อนำไป generate เป็น 16 Subkeys จะทำให้ได้ 2 Subkeys ที่แตกต่างกันเท่านั้น ซึ่งแต่ละ Subkeys ก็จะถูกใช้ทั้งหมด 8 รอบ ไม่ใช่ 16 subkey ที่แตกต่างกัน ทำให้มีความปลอดภัยต่ำ

บางคู่ของ key ที่เข้ารหัสจาก plaintext เป็น identical ciphertext key ที่เป็น semiweak key ได้แก่

- 01FE01FE01FE01FE และ FE01FE01FE01FE01
- 1FE01FE00EF12EF1 และ E01FE01FF10EF10E
- 01E001E001F101F1 และ E001E001F101F101
- 1FFE1FFE0EFE0EFE และ FE1FFE1FFE0EFE0E
- 011F011F010E010E และ 1F011F010E010E01
- E0FEE0FEF1FEF1FE และ FEE0FEE0FEF1FEF1

ในส่วนนี้ยังรวมถึง 48 key ซึ่งเป็น product ที่ได้จาก 4 subkeys ที่แตกต่างกันด้วย (แทนด้วยฐาน 16) ซึ่งถูกเรียกว่า weak keys ที่เป็นไปได้ (possibly weak)

## ตัวอย่างการเข้ารหัสแบบ DES

สมมติให้ Plaintext (เลขฐานสิบหก) คือ 0123456789ABCDEF

และ Key (เลขฐานสิบหก) คือ 133457799BBCDFF1

ดังนั้น Key ที่เป็นเลขฐานสองที่ยังไม่มี parity bit จะเขียนได้ว่า

0001 0010 0110 1001 0101 1011 1100 1001 1011 0111 1011 0111 1111 1000

Key ที่ตัด parity bit แล้ว คือ

0001001 0110100 0101101 1100100 1011011 1011011 1111100

plaintext แปลงเป็นเลขฐานสอง คือ

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

นำ plaintext ไป apply บน IP table ได้เป็น  $L_0R_0$

$L_0 = 11001100000000001100110011111111$

$L_1 = R_0 = 11110000101010101111000010101010$

ทำการ เข้ารหัสให้ครบ 16 รอบ

|                     |                                                      |
|---------------------|------------------------------------------------------|
| $E(R_0)$            | $= 011110100001010101010101011110100001010101010101$ |
| $K_1$               | $= 00011011000000101110111111111000111000001110010$  |
| $E(R_0) \oplus K_1$ | $= 011000010001011110111010100001100110010100100111$ |
| S-box outputs       | $= 01011100100000101011010110010111$                 |
| $f(R_0, K_1)$       | $= 00100011010010101010100110111011$                 |
| $L_2 = R_1$         | $= 11101111010010100110010101000100$                 |

|                     |                                                      |
|---------------------|------------------------------------------------------|
| $E(R_1)$            | $= 011101011110101001010100001100001010101000001001$ |
| $K_2$               | $= 011110011010111011011001110110111100100111100101$ |
| $E(R_1) \oplus K_2$ | $= 000011000100010010001101111010110110001111101100$ |
| S-box outputs       | $= 11111000110100000011101010101110$                 |
| $f(R_1, K_2)$       | $= 00111100101010111000011110100011$                 |
| $L_3 = R_2$         | $= 11001100000000010111011100001001$                 |

|                     |                                                    |
|---------------------|----------------------------------------------------|
| $E(R_2)$            | = 11100101100000000000010101110101110100001010011  |
| $K_3$               | = 010101011111110010001010010000101100111110011001 |
| $E(R_2) \oplus K_3$ | = 101100000111110010001000111110000010011111001010 |
| S-box outputs       | = 00100111000100001110000101101111                 |
| $f(R_2, K_3)$       | = 01001101000101100110111010110000                 |
| $L_4 = R_3$         | = 10100010010111000000101111110100                 |

|                     |                                                   |
|---------------------|---------------------------------------------------|
| $E(R_3)$            | = 01010000010000101111100000000101011111110101001 |
| $K_4$               | = 011100101010110111010110110110011010100011101   |
| $E(R_3) \oplus K_4$ | = 001000101110111100101110110111100100101010100   |
| S-box outputs       | = 00100001111011011001111100111010                |
| $f(R_3, K_4)$       | = 10111011001000110111011101001100                |
| $L_5 = R_4$         | = 01110111001000100000000001000101                |

|                     |                                                     |
|---------------------|-----------------------------------------------------|
| $E(R_4)$            | = 1011101011101001000001000000000000000001000001010 |
| $K_5$               | = 011111001110110000000111111010110101001110101000  |
| $E(R_4) \oplus K_5$ | = 110001100000010100000011111010110101000110100010  |
| S-box outputs       | = 01010000110010000011000111101011                  |
| $f(R_4, K_5)$       | = 00101000000100111010110111000011                  |
| $L_6 = R_5$         | = 10001010010011111010011000110111                  |

|                     |                                                    |
|---------------------|----------------------------------------------------|
| $E(R_5)$            | = 110001010100001001011111110100001100000110101111 |
| $K_6$               | = 011000111010010100111110010100000111101100101111 |
| $E(R_5) \oplus K_6$ | = 101001101110011101100001100000001011101010000000 |
| S-box outputs       | = 01000001111100110100110000111101                 |
| $f(R_5, K_6)$       | = 10011110010001011100110100101100                 |
| $L_6 = R_5$         | = 11101001011001111100110101101001                 |

|                     |                                                    |
|---------------------|----------------------------------------------------|
| $E(R_6)$            | = 111101010010101100001111111001011010101101010011 |
| $K_7$               | = 111011001000010010110111111101100001100010111100 |
| $E(R_6) \oplus K_7$ | = 000110011010111110111000000100111011001111101111 |
| S-box outputs       | = 00010000011101010100000010101101                 |
| $f(R_6, K_7)$       | = 10001100000001010001110000100111                 |
| $L_8 = R_7$         | = 00000110010010101011101000010000                 |

|                     |                                                     |
|---------------------|-----------------------------------------------------|
| $E(R_7)$            | = 000000001100001001010101010111110100000010100000  |
| $K_8$               | = 11110111100010100011101011000001001110111111011   |
| $E(R_7) \oplus K_8$ | = 1111011110100100001101111100111100111101101011011 |
| S-box outputs       | = 01101100000110000111110010101110                  |
| $f(R_7, K_8)$       | = 00111100000011101000011011111001                  |
| $L_9 = R_8$         | = 11010101011010010100101110010000                  |

|                     |                                                    |
|---------------------|----------------------------------------------------|
| $E(R_8)$            | = 011010101010101101010010101001010111110010100001 |
| $K_9$               | = 111000001101101111101011111011011110011110000001 |
| $E(R_8) \oplus K_9$ | = 100010100111000010111001010010001001101100100000 |
| S-box outputs       | = 00010001000011000101011101110111                 |
| $f(R_8, K_9)$       | = 00100010001101100111110001101010                 |
| $L_{10} = R_9$      | = 00100100011111001100011001111010                 |

|                        |                                                    |
|------------------------|----------------------------------------------------|
| $E(R_9)$               | = 000100001000001111111001011000001100001111110100 |
| $K_{10}$               | = 101100011111001101000111101110100100011001001111 |
| $E(R_9) \oplus K_{10}$ | = 101000010111000010111110110110101000010110111011 |
| S-box outputs          | = 11011010000001000101001001110101                 |
| $f(R_9, K_{10})$       | = 01100010101111001001110000100010                 |
| $L_{11} = R_{10}$      | = 10110111110101011101011110110010                 |

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| $E(R_{10})$               | = 01011010111111101010101111101010111110110100101  |
| $K_{11}$                  | = 001000010101111111010011110111101101001110000110 |
| $E(R_{10}) \oplus K_{11}$ | = 011110111010000101111000001101000010111000100011 |
| S-box outputs             | = 01110011000001011101000100000001                 |
| $f(R_{10}, K_{11})$       | = 11100001000001001111101000000010                 |
| $L_{12} = R_{11}$         | = 11000101011110000011110001111000                 |

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| $E(R_{11})$               | = 011000001010101111110000000111111000001111110001 |
| $K_{12}$                  | = 011101010111000111110101100101000110011111101001 |
| $E(R_{11}) \oplus K_{12}$ | = 000101011101101000000101100010111110010000011000 |
| S-box outputs             | = 01111011100010110010011000110101                 |
| $f(R_{11}, K_{12})$       | = 11000010011010001100111111101010                 |
| $L_{13} = R_{12}$         | = 01110101101111010001100001011000                 |

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| $E(R_{12})$               | = 001110101011110111111010100011110000001011110000 |
| $K_{13}$                  | = 100101111100010111010001111110101011101001000001 |
| $E(R_{12}) \oplus K_{13}$ | = 101011010111100000101011011101011011100010110001 |
| S-box outputs             | = 10011010110100011000101101001111                 |
| $f(R_{12}, K_{13})$       | = 11011101101110110010100100100010                 |
| $L_{14} = R_{13}$         | = 00011000110000110001010101011010                 |

|                           |                                                    |
|---------------------------|----------------------------------------------------|
| $E(R_{13})$               | = 0000111100010110000001101000101010101011110100   |
| $K_{13}$                  | = 010111110100001110110111111100101110011100111010 |
| $E(R_{12}) \oplus K_{13}$ | = 010100000101010110110001011110000100110111001110 |
| S-box outputs             | = 01100100011110011001101011110001                 |
| $f(R_{12}, K_{13})$       | = 10110111001100011000111001010101                 |
| $L_{14} = R_{13}$         | = 11000010100011001001011000001101                 |

$$\begin{aligned}
E(R_{14}) &= 111000000101010001011001010010101100000001011011 \\
K_{15} &= 10111111100100011000110100111101001111100001010 \\
E(R_{14}) \oplus K_{15} &= 0101111110001011101010001110111111111101010001 \\
\text{S-box outputs} &= 10110010111010001000110100111100 \\
f(R_{14}, K_{15}) &= 01011011100000010010011101101110 \\
L_{16} = R_{15} &= 01000011010000100011001000110100
\end{aligned}$$

$$\begin{aligned}
E(R_{15}) &= 001000000110101000000100000100100100000110101000 \\
K_{16} &= 11001011001111011000101100001110000101111110101 \\
E(R_{15}) \oplus K_{16} &= 111010110101011110001111000101000101011001011101 \\
\text{S-box outputs} &= 10100111100000110010010000101001 \\
f(R_{15}, K_{16}) &= 11001000110000000100111110011000 \\
R_{16} &= 00001010010011001101100110010101
\end{aligned}$$

สุดท้าย นำ  $L_{16}R_{16}$  ไป apply บน  $IP^{-1}$  เราจะได้ข้อความที่เข้ารหัสแล้ว ซึ่งเขียนในรูปแบบสิบหก  
ได้ว่า

85E813540F0AB405

## 2.3 ทฤษฎีรหัสลับแบบ RSA

เป็นการเข้ารหัสแบบ Public key encryption ซึ่งจะใช้กุญแจที่แตกต่างกัน 2 ตัวในการเข้ารหัสและถอดรหัส

ในการเข้ารหัสแบบ RSA ข้อความจะถูกแทนด้วยตัวเลขจำนวนเต็มที่อยู่ในช่วง  $[0, n-1]$  ผู้ใช้แต่ละคนจะเลือกค่า  $n$  ของตนเอง และเลขจำนวนเต็มอีกคู่ คือ ค่า  $e$  และค่า  $d$  ในลักษณะที่จะอธิบายต่อไปนี้ ผู้ใช้สามารถจะแสดงกุญแจ  $(n, e)$  ไว้ในที่เปิดเผย เช่น ในสมุดโทรศัพท์ ส่วนกุญแจการถอดรหัสจะประกอบด้วยค่า  $(n, d)$  โดยที่ค่า  $d$  เก็บไว้เป็นความลับ การเข้ารหัสข้อความ  $C$  และข้อความที่ถูกถอดรหัส  $M$  กำหนดได้ดังนี้

$$\text{การเข้ารหัส : } C = E(M) = M^e \bmod n \quad (1)$$

$$\text{การถอดรหัส : } M = D(C) = C^d \bmod n \quad (2)$$

เมื่อ  $e$  และ  $n$  เป็นกุญแจในการเข้ารหัส  $M$  จะถูกหาค่ากลับโดยวิธีเดียวกัน แต่โดยใช้ตัวเลขยกกำลัง  $d$  ที่ต่างกัน ค่า  $n$  ได้จากการเลือกเลขจำนวนเฉพาะ 2 จำนวน  $p, q$  ที่มีค่ามาก ๆ ค่า  $n$  จะเป็น ผลคูณของ  $p$  และ  $q$  ดังนี้

$$n = pq \quad (3)$$

การเข้ารหัสและถอดรหัสสามารถทำได้โดยการคำนวณเลขยกกำลังแบบเร็ว (Fast Exponentiation Algorithm) ซึ่งผลลัพธ์ที่ได้จะเป็นจำนวนเต็มที่อยู่ในช่วง  $[0, n-1]$

$$C = \text{FastExp}(M, e, n) \quad (4)$$

$$M = \text{FastExp}(C, d, n) \quad (5)$$

การเข้ารหัสและการถอดรหัสจะขึ้นกับ Euler's Generalization of Fermat's Theorem ซึ่งกล่าวว่าสำหรับแต่ละ Relatively Prime กับ  $n$

$$M_p^{(n)} \bmod n = 1 \quad (6)$$

เมื่อ  $\phi(n) = (p-1)(q-1)$  เรียกว่า Euler Totient Function คุณสมบัตินี้หมายความว่า ถ้า  $e$  และ  $d$  ที่สอดคล้องกันกับสมการ

$$ed \bmod \phi(n) = 1 \quad (7)$$

ดังนั้น จะได้ว่าสมการ (2) จะเป็นอินเวอร์สของสมการที่ (1) ซึ่งแสดงได้ดังนี้ ให้  $e$  และ  $d$  สอดคล้องตามสมการที่ (6) และให้ข้อความ  $M$  อยู่ในช่วง  $[0, n-1]$  ที่ซึ่ง  $\text{GCD}(M, n) = 1$  ( $\text{GCD}$  หมายถึงตัวหารร่วมมาก) จากสมการ (2) จะได้

$$\begin{aligned} M &= D(C) = D(E(M)) = (M^e \bmod n)^d \bmod n \\ &= M^{ed} \bmod n \end{aligned} \quad (8)$$

จาก  $ed \bmod \phi(n) = 1$  หมายความว่า  $ed = t\phi(n) + 1$

$(n)+1$  เป็นจำนวนเต็ม  $t$  ดังนั้น

$$M^{ed} \bmod n = M^{t\alpha(n)+1} \bmod n \quad (9)$$

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

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

$$\text{เมื่อ } M^{t\alpha(n)} \bmod n = (M^{\alpha(n)} \bmod n)^t \bmod n \quad (10)$$

$$= 1^t \bmod n$$

$$= 1$$

$$\text{ดังนั้น } M^{ed} \bmod n = (M \times 1) \bmod n = M$$

## ตัวอย่างการเข้ารหัสแบบ RSA

ถ้าต้องการเข้ารหัสข้อความว่า KING MONGKUT INSTITUTE OF TECHNOLOGY โดยสมมติให้แทนตัวอักษรด้วยตัวเลขต่อไปนี้ A = 01, B = 02, C = 03, D = 04, E = 05, F = 06, G = 07, H = 08, I = 09, J = 10, K = 11, L = 12, M = 13, N = 14, O = 15, P = 16, Q = 17, R = 18, S = 19, T = 20, U = 21, V = 22, W = 23, X = 24, Y = 25, Z = 26, BLANK = 00 จะแปลงข้อความนี้เป็นตัวเลขได้ดังนี้

110914070013151407112120000914192009202120050015060020050308141512507

25

เลือก  $p = 73$ ,  $q = 151$  จะได้  $n = pq = 73 \cdot 151 = 11023$  และสุ่มเลือก  $e = 11$  แบ่งข้อความที่ได้เป็น block block ละ 4 หลัก จากนั้นหาโมดูลอ  $n$  ดังนี้

$$C_1 = 1109^{11} \bmod 11023 = 5902$$

$$C_{10} = 2021^{11} \bmod 11023 = 9538$$

$$C_2 = 1407^{11} \bmod 11023 = 8015$$

$$C_{11} = 2005^{11} \bmod 11023 = 7420$$

$$C_3 = 0013^{11} \bmod 11023 = 4385$$

$$C_{12} = 0015^{11} \bmod 11023 = 10205$$

$$C_4 = 1514^{11} \bmod 11023 = 8433$$

$$C_{13} = 0600^{11} \bmod 11023 = 4855$$

$$C_5 = 0711^{11} \bmod 11023 = 7922$$

$$C_{14} = 2005^{11} \bmod 11023 = 7420$$

$$C_6 = 2120^{11} \bmod 11023 = 1436$$

$$C_{15} = 0308^{11} \bmod 11023 = 9892$$

$$C_7 = 0009^{11} \bmod 11023 = 576$$

$$C_{16} = 1415^{11} \bmod 11023 = 10833$$

$$C_8 = 1419^{11} \bmod 11023 = 9127$$

$$C_{17} = 1215^{11} \bmod 11023 = 5601$$

$$C_9 = 2009^{11} \bmod 11023 = 3133$$

$$C_{18} = 0725^{11} \bmod 11023 = 3911$$

จะสังเกตได้ว่า ขนาดของการแบ่ง block ของข้อความจะต้องน้อยกว่าค่า  $n$  เพราะเซตทั้งหมดของโมดูลอ  $n$  จะอยู่ในช่วง  $[0, n-1]$  เมื่อจะถอดรหัสทำได้โดยวิธีการเดียวกันโดยใช้กุญแจในการถอดรหัสอีกด้วย คือ 5891 ซึ่งได้มาจาก

$$\rho(n) = (p-1)(q-1) = (73-1)(151-1) = 72 \cdot 150 = 10800$$

$$5891 \cdot 11 \bmod 10800 = 1$$

การถอดรหัสทำได้ดังนี้

$$M_1 = 5902^{5891} \bmod 11023 = 1109$$

$$M_{10} = 9538^{5891} \bmod 11023 = 2021$$

$$M_2 = 8015^{5891} \bmod 11023 = 1407$$

$$M_{11} = 7420^{5891} \bmod 11023 = 2005$$

$$M_3 = 4385^{5891} \bmod 11023 = 1013$$

$$M_{12} = 10206^{5891} \bmod 11023 = 0015$$

$$M_4 = 8433^{5891} \bmod 11023 = 1514$$

$$M_{13} = 4855^{5891} \bmod 11023 = 0600$$

$$M_5 = 7922^{5891} \bmod 11023 = 0711$$

$$M_{14} = 7420^{5891} \bmod 11023 = 2005$$

$$M_6 = 1436^{5891} \bmod 11023 = 2120$$

$$M_{15} = 9892^{5891} \bmod 11023 = 0308$$

$$M_7 = 576^{5891} \bmod 11023 = 0009$$

$$M_{16} = 10833^{5891} \bmod 11023 = 1415$$

$$M_8 = 9127^{5891} \bmod 11023 = 1419$$

$$M_{17} = 5601^{5891} \bmod 11023 = 1215$$

$$M_9 = 3133^{5891} \bmod 11023 = 2009$$

$$M_{18} = 3911^{5891} \bmod 11023 = 0725$$

จากตัวอย่างเดิม ถ้าเลือกค่า  $p$  และ  $q$  ที่มีจำนวนมากขึ้น จะลดจำนวนครั้งของการเข้ารหัสซึ่งทำให้การเข้ารหัสเร็วขึ้น

เลือก  $p = 23203941528510160092833$  (จำนวนเฉพาะ 23 หลัก)

$q = 1735739017547947065597$  (จำนวนเฉพาะ 22 หลัก)

จะได้  $n = pq = 40275986671936234424621110207398375752479$

552 (44 หลัก)

$\phi(n) = (p-1)(q-1) = 40275986671936234424621110207398375752$

479552

จากคุณสมบัติ  $ed = 1 \bmod (p-1)(q-1)$  จะได้

กุญแจเข้ารหัส  $e = 7$

กุญแจถอดรหัส  $d = 5753712381705176346374444315342625107497079$

ซึ่ง  $exd = 40275986671936234424621110207398375752479553$

จากข้อความเดิม KING MONGUT INSTITUTE OF TECHNOLOGY

เมื่อแปลงเป็นตัวเลขโดยแทน A = 01, B = 02, C = 03, ..., Y = 25, Z = 26, BLANK = 00

แล้วจะได้ตัวเลข 72 หลัก

110914070013151407112120000914192009202120050015060020050308141512150725

จากจำนวนหลักของ  $n$  เท่ากับ 44 หลัก เราสามารถแบ่งข้อความที่แปลงเป็นตัวเลขได้ 2 block

โดย block แรกใช้ 40 หลัก ส่วน block ที่สองใช้ 32 หลักแล้วเข้ารหัสได้ข้อความดังนี้

ข้อความ block ที่ 1  $M_1 = 1109140700131514071121200009141920092021$

ทำการเข้ารหัสได้ดังนี้  $C_1 = M_1^e \bmod n$

จะได้ข้อความที่เข้ารหัส  $C_1 = 9380655288089560794796743877454715306$

769400

ข้อความบล็อกที่สอง  $M_2 = 20050015060020050308141512150725$

โดย  $C_2 = M_2^e \bmod n$

จะได้ข้อความที่เข้ารหัส  $C_2 = 36747029698992796936193619396589429$

53373787579167

ซึ่งทำทั้งหมดเพียง 2 block ทำให้เวลาที่ใช้สั้นลง

## 2.4 ทฤษฎีทางคณิตศาสตร์ที่เกี่ยวข้อง

### 1. เลขคณิตมอดูลาร์

กำหนดให้  $a, b$  เป็นจำนวนเต็ม และ  $n$  เป็นจำนวนเต็มบวก เรียก  $a$  เป็นคอนกรูเอนซ์กับ  $b$  มอดุโล  $n$  ( $a$  is congruent to  $b$  modulo  $n$ ) ก็ต่อเมื่อ  $n \mid (a-b)$  หมายถึง  $n$  หาร  $(a-b)$  ลงตัว ที่เขียนแทนด้วย  $a \equiv b \pmod{n}$  หรือ  $a \equiv_n b$

จากนิยามการหารลงตัว เราจะได้ว่า  $a \equiv b \pmod{n}$  ก็ต่อเมื่อ  $a = b + kn$  สำหรับ  $k$  บางค่า

ถ้า  $a \equiv b \pmod{n}$  จะเรียก ว่าเศษเหลือของ  $a \equiv b \pmod{n}$  เช่น  $17 \equiv 5 \pmod{12}$  นั่นคือ 5 เป็น เศษเหลือของ  $17 \pmod{12}$

สังเกตว่า ค่า  $a$  และ  $b$  สามารถที่จะเป็นจำนวนเต็มลบได้ เช่น  $-6 \equiv 10 \pmod{8}$  ซึ่งได้มาจาก  $-6 = 8(-1) + 2$

กำหนดให้  $A = \{r_1, r_2, \dots, r_n\}$  เป็นเศษเหลือทั้งหมด ในมอดุโล  $n$  สำหรับจำนวนเต็ม  $a$  ใด ๆ จะหาค่า  $r_i$  สำหรับบางค่า  $i \leq n$  ในเซต  $A$  เพียงค่าหนึ่งแลค่าเดียวเท่านั้น ซึ่ง

$$a \equiv r_i \pmod{n}$$

ดังนั้น จะได้ว่า เซตจำนวนเต็มที่เรียงติดต่อกัน  $n$  จำนวนเป็น ระบบเศษเหลือสมบูรณ์ มอดุโล  $n$  (Compleat Residues System modulo  $n$  : C.R.S.)

โดยปกติแล้ว ระบบเศษเหลือสมบูรณ์ มอดุโล  $n$  ที่นิยมใช้กันมากได้แก่

เซตของ  $0, 1, 2, \dots, n-1$

เซตของ  $0, 1, 2, \dots, n$

$$\text{และเซตของ } -\frac{(n-1)}{2}, -\frac{(n-1)}{2}+1, \dots, -1, 0, 1, \dots, \frac{(n-1)}{2}-1, \frac{(n-1)}{2}$$

(สำหรับ  $n$  ที่เป็นจำนวนคี่บวก) เช่น เมื่อ  $n = 5$  จะได้  $(-2, 1, 0, 1, 2)$  เป็นเซตของระบบเศษเหลือ มอดุโล 5

จากความจริงที่ว่า จำนวน  $0, 1, 2, \dots, n-1$  เป็นระบบเศษเหลือสมบูรณ์ มอดุโล  $n$  ทำให้สรุปต่อไปได้ว่า จำนวนเหล่านี้ เมื่อนำมาบวกหรือลบ หรือคูณกัน แล้วจะเป็นคอนกรูเอนซ์ มอดุโล  $n$  กับจำนวนเต็มจำนวนเดียว ในระบบเศษเหลือสมบูรณ์ มอดุโล  $n$  นี้ จึงนำไปสู่แนวคิด ในการสร้าง เลขคณิตมอดุโล  $n$  (Arithmetic Modulo  $n$ ) หรือบางครั้งเรียกว่า เลขคณิตมอดูลาร์

ดังนั้น จะได้ว่า กฎทางคณิตศาสตร์ที่เกี่ยวกับ การบวก และการคูณของจำนวนเต็ม ยังจริงอยู่ในเลขคณิตมอดุโล  $n$  เช่น

$$\text{ถ้า } a \equiv b \pmod{n} \text{ และ } c \equiv d \pmod{n} \text{ แล้ว}$$

$$a+c \equiv b+d \pmod{n}$$

$$ac \equiv bd \pmod{n}$$

$$a+c \equiv c+a \pmod{n}$$

$$ac \equiv ca \pmod{n}$$

$$a+(b+c) \equiv (a+b) + c \pmod{n}$$

$$a(bc) \equiv (ab)c \pmod{n}$$

$$a(b+c) \equiv ab + ac \pmod{n}$$

$$(b+c)a \equiv ba + ca \pmod{n}$$

## 2. The Greatest Common Divisor

สำหรับ ค่า  $a$  และ  $b$  จำนวน  $(a,b)$  คือ จำนวนที่ใหญ่ที่สุดที่สามารถหาร  $a$  และ  $b$  ได้ลงตัว เช่น

$$(56,98) = 14 \quad (76,190) = 38$$

**ทฤษฎีบท 1** สำหรับ บางค่า  $a,b$  จะมีจำนวนเต็ม  $x,y$  ซึ่ง  $ax + by = (a,b)$

ทฤษฎีนี้ สามารถใช้แก้ปัญหาโดยการใช้อย่าง

$$30x + 69y = 3$$

$$30x' + 9y = 3 \quad [x' = x+2y]$$

$$3x' + 9y' = 3 \quad [y' = y+3x']$$

$$3x'' + 9y' = 3 \quad [x'' = x' + 3y']$$

มันง่ายที่จะพบว่าผลเฉลยสุดท้ายของสมการ คือ  $x'' = 1, y' = 0$  และเมื่อเราแทนค่าผลเฉลยย้อนกลับไปในสมการเริ่มต้น จะได้ว่า

$$x' = x'' - 3y' = 1 \quad y = y' - 3x' = -3 \quad x = x' - 2y = 7$$

เราสามารถจะแก้ไขสมการที่คล้ายกับสมการนี้ได้ด้วย เช่น  $30x + 69y = 15$  ด้วยการคูณผลลัพธ์ของเรา :  $y = -15, x = 35$  ซึ่งชัดเจนอยู่แล้วว่าสมการจะไม่มีผลเฉลยที่เป็นจำนวนเต็ม ถ้า 15 ถูกเปลี่ยนไปโดยไม่เป็นผลคูณของ  $(30,69) = 3$

ทุก ๆ ผลเฉลยของ  $30x + 69y = 15$  เราสามารถจะเขียนสมการผลเฉลยแรกได้ว่า

$$y = -15 + \frac{30}{3}t \quad x = 35 - \frac{69}{3}t \quad \text{สำหรับจำนวนเต็ม } t$$

ความเป็นจริงแล้ว ขั้นตอนดังกล่าว จะถูกแสดงในเรื่อง Euclid's algorithm สำหรับการหา greatest common divisor

**บทแทรก 1** ถ้า  $p$  เป็นจำนวนเฉพาะ ,  $ar \equiv as \pmod{p}$  และ  $a \not\equiv 0$  แล้ว  $r \equiv s$

**บทแทรก 2** ถ้า  $p$  เป็นจำนวนเฉพาะ และ  $a \not\equiv 0 \pmod{p}$  แล้ว สำหรับ บาง  $b$  แล้ว  $ay \equiv b \pmod{p}$

บทแทรก 3 (Chinese Remainder Theorem - ทฤษฎีเศษเหลือไชนีส) ให้  $p_1, \dots, p_r$  เป็นจำนวนเฉพาะที่สัมพัทธ์แบบคู่ เมื่อ  $n = p_1 p_2 \dots p_r$  แล้วสมการ  $r$  สมการ ที่เขียนได้เป็น  $x \equiv x_i \pmod{p_i}; i = 1, \dots, r$  มีาร่วมเพียงรากเดียวใน มอดุโล  $n$

พิสูจน์

$$\text{สำหรับ } i \text{ แต่ละตัว, } \gcd \left[ p_1, \frac{n}{p_i} \right] = 1$$

จะมี  $y_i$  ซึ่ง

$$\left[ \frac{n}{p_i} \right] y_i \equiv 1 \pmod{p_i}$$

หรือ

$$\left[ \frac{n}{p_i} \right] y_i \equiv 0 \pmod{p_j} \quad j \neq i, p_i \mid \frac{n}{p_i}$$

$$\text{ให้ } x = \left[ \sum \left[ \frac{n}{p_i} \right] y_i x_i \right] \pmod{n}$$

ซึ่งจะมีาร่วมใน มอดุโล  $n$

### 3. Power modulo a prime

ลำดับ  $a, a^2, a^3, \dots \pmod{p}$  จะใช้ประโยชน์อย่างมากในเรื่องรหัสลับ ก่อนที่จะดูทฤษฎีเราจะแสดงตัวอย่างให้ดูเพื่อให้เกิดความชัดเจนในการคำนวณตัวเลขที่มีหลายหลัก

สมมติให้ เราต้องการคำนวณ  $432^{678} \pmod{987}$  จะได้ว่า

$$432^2 \equiv 186624 \equiv 81 \quad 432^4 \equiv 81^2 \equiv 639 \quad 432^8 \equiv 639^2 \equiv 690 \dots 432^{512} \equiv 858$$

ด้วยเหตุที่  $678 = 512 + 128 + 32 + 4 + 2$ ,

$$\text{ดังนั้น } 432^{678} \equiv (81)(639) \dots (858) \equiv 204$$

ลำดับของเลขที่ยกกำลัง 2 แล้ว  $\pmod{11}$  คือ

$$2 \ 4 \ 8 \ 5 \ 10 \ 9 \ 7 \ 3 \ 6 \ 1$$

แต่ละจำนวนจาก 1 ถึง 10 ปรากฏอยู่ในลำดับ

ทฤษฎีบท 2 ให้  $p$  เป็นจำนวนเฉพาะ สำหรับ  $1 \leq b \leq p-1, 1 \leq x \leq p-1$  จะมี  $a^x \equiv b \pmod{p}$

แต่มันจะไม่ถูกต้องในกรณีที่  $a = 2$  ลำดับของเลขที่ยกกำลัง 2 แล้ว  $\pmod{7}$  คือ 2, 4, 1 และเมื่อทำซ้ำลำดับ เราจะได้ 3, 5, หรือ 6

บทแทรกของทฤษฎีบท 2 ได้แก่

บทแทรก 4 ให้  $a$  มีคุณสมบัติดังทฤษฎีบท 5 แล้ว  $a^{p-1} \equiv 1 \pmod{p}$

บทแทรก 5 สำหรับบาง  $b \neq 0, b^{p-1} \equiv 1 \pmod{p}$

**บทแทรก 6** ถ้า  $x \equiv P \pmod{p-1}$  แล้ว  $b^x \equiv b^y \pmod{p}$

**Lemma 1** ให้  $b \neq 0$ ,  $d$  เป็นจำนวนบวกที่น้อยที่สุด และ  $b^d \equiv 1$  แล้ว สำหรับ  $b^e \equiv 1$   $d$  จะหาร  $e$  ได้ลงตัว

#### 4. Primitive roots

จากทฤษฎีบท 2 ถูกเขียนไว้ว่า  $p$  เป็นจำนวนเฉพาะ ดังนั้นเราจะได้สมการ

$$a^x \equiv b \pmod{p}$$

จะมีผลเฉลยสำหรับบาง  $b \neq 0$  ดังนั้น  $a$  จะถูกเรียกว่า primitive root of  $p$  และ  $x$  จะถูกเรียกว่า discrete logarithm of  $b$

ยังไม่มีวิธีการค้นหา primitive roots ที่มีประสิทธิภาพในทุกกรณี อย่างไรก็ตาม มันต้องพบกรณีหนึ่งที่เป็นกรณีพิเศษ เราจะใช้  $p = 1223$  เป็นตัวอย่าง  $p-1 = 2 \cdot 13 \cdot 47$  โดย Lemma 1 ถ้า  $a$  ไม่เป็น primitive root แล้ว เราจะได้ว่า ถ้าไม่  $a^{26}$ ,  $a^{94}$ , หรือไม่เช่นนั้นก็  $a^{611} \equiv 1 \pmod{1223}$  ในกรณี  $a = 2$ , และ 3 จะเป็นเท็จ แต่  $a = 5$  จะถูกต้องทั้ง 3 เงื่อนไข ดังนั้น มันเป็น primitive root เราสามารถบอกได้ว่า  $a = 4$  จะไม่เป็น primitive root โดยไม่ต้องทดสอบ

มันง่ายที่เราจะแสดงว่า ถ้า  $a$  เป็น primitive root,  $a^x$  จะเป็น primitive root ก็ต่อเมื่อ  $(x, p-1) = 1$  ตัวอย่างเช่น

$$1222 \left( \frac{1}{2} \right) \left( \frac{12}{13} \right) \left( \frac{46}{47} \right) = 552$$

ดังนั้น ถ้าเราจะทำการสุ่ม ความน่าจะเป็นที่มันจะเป็น primitive root  $\approx .45$  เมื่อทำการสุ่มและการทดสอบจนกระทั่งพบ primitive root จะใช้เวลาไม่นานเกินไปนัก

## 2.5 ความปลอดภัยของระบบรหัสลับ

ความปลอดภัยของระบบรหัสนั้นขึ้นอยู่กับ Algorithm ของระบบรหัสนั้น ๆ เราจึงต้องแยกพิจารณาความปลอดภัยของระบบรหัสลับแบบ DES และ RSA

### ความปลอดภัยของการเข้ารหัสโดยใช้ Algorithm แบบ DES

ในการที่จะถอดรหัสโดยที่ผู้ทำการถอดรหัสไม่รู้ Key เรียกว่า การแกะรหัส ผู้ที่จะทำการแกะรหัสต้องรู้ ข้อความที่เข้ารหัสแล้ว (ciphertext) และข้อความปกติ (plaintext) ที่สัมพันธ์กันส่วนหนึ่งก่อน เพื่อที่จะนำมา generate เป็น 64 บิต

ถ้า Key ที่ใช้เข้ารหัส มีขนาด 8 บิต ท่านจะมี Key ที่เป็นไปได้ทั้งหมด  $2^8$  key หรือ 256 key ที่เป็นไปได้ อย่างไรก็ตาม ต้องใช้ความพยายามถึง 256 ครั้งที่จะหาค่า key ที่ถูกต้อง และมีโอกาสถึง 50% ของการหา key ที่จะพบ key ที่ถูกต้อง หลังจากที่ได้ลองทดสอบไปครั้งหนึ่งแล้ว แต่ถ้า key มีขนาด 56 บิต ดังนั้น มันจะมี key ที่เป็นไปได้ถึง  $2^{56}$  key สมมติให้เครื่องซูเปอร์คอมพิวเตอร์ สามารถที่จะคำนวณได้ ล้าน key ต่อวินาที ดังนั้น มันต้องใช้เวลาทั้งหมด 2285 ปีที่จะหาค่า key ที่ถูกต้อง และถ้า key มีขนาด 64 ดังนั้น เมื่อใช้เครื่องซูเปอร์คอมพิวเตอร์เครื่องเดิม ต้องใช้เวลาในการคำนวณถึง 585,000 ปี ที่จะหา key ที่ถูกต้องใน  $2^{64}$  key ที่เป็นไปได้ ถ้า key มีขนาด 128 บิต มันต้องใช้เวลาราว  $10^{25}$  ปี และถ้า key ที่ใช้มีขนาด 2048 บิต จะต้องใช้เครื่องคอมพิวเตอร์ทำงานขนานกัน ที่สามารถคำนวณหา key ได้ ล้าน ล้านครั้งต่อวินาที ก็ต้องใช้เวลาราว  $10^{597}$  ปี ที่จะหา key ที่ถูกต้อง

สิ่งที่ใช้เวลากำหนดความเร็วของการแกะรหัส คือ ขนาดของ key และ ความเร็วในการทดสอบของแต่ละ key

ในปี พ.ศ. 2530 Whitfield Diffie และ Martin Hellman ได้อ้างถึงเครื่องที่ใช้ในการแกะรหัสแบบ DES เครื่องที่อ้างถึงนี้ ใช้ชิพถึงหนึ่งล้านชิพ ชิปแต่ละตัวสามารถที่จะคำนวณ key ได้ หนึ่งล้าน key ต่อวินาที เครื่องนั้นสามารถที่จะคำนวณ key ได้ 256 key ภายใน 1 ชั่วโมง และถ้าใช้ key ในการเข้ารหัสทั้งหมด 64 บิต จะต้องใช้เวลารอบคอบถึง 214 วัน

การแกะรหัสโดยใช้โปรเซสเซอร์ที่ต่อขนานกัน แต่ละโปรเซสเซอร์สามารถที่จะคำนวณถึง ช่องว่างได้ (Keyspace) โปรเซสเซอร์แต่ละตัวไม่สามารถที่จะติดต่อกันได้ การติดต่ออย่างเดียวยังต้องการ คือ สัญญาณเมื่อประสบความสำเร็จ และไม่มีคามจำเป็นที่จะต้องทำการแบ่งปันหน่วยความจำกัน

เมื่อไม่นานมานี้ Michael Wiener ได้ออกแบบเครื่องที่ใช้ชิพ , บอร์ด และอุปกรณ์พิเศษ และซึ่งถูกประมาณราคาไว้ 1 ล้านดอลลาร์ ทำให้สามารถที่จะแกะรหัสแบบ DES ได้โดยใช้เวลาเฉลี่ย 3.5 ชั่วโมง (แต่ให้การรับรองว่าสามารถที่จะแกะรหัสได้ในเวลา 7 ชั่วโมง)

เวลาประมาณโดยเฉลี่ยสำหรับการแกะรหัสเมื่อปี พ.ศ. 2538

| ค่าใช้จ่าย | ความยาวของ key (บิต) |               |           |           |              |              |
|------------|----------------------|---------------|-----------|-----------|--------------|--------------|
|            | 40                   | 56            | 64        | 80        | 112          | 128          |
| \$100 K    | 2 วินาที             | 35 ชั่วโมง    | 1 ปี      | 70,000 ปี | $10^{14}$ ปี | $10^{19}$ ปี |
| \$1 M      | .2 วินาที            | 3.5 ชั่วโมง   | 37 วัน    | 7,000 ปี  | $10^{13}$ ปี | $10^{18}$ ปี |
| \$10 M     | .02 วินาที           | 21 นาที       | 4 วัน     | 700 ปี    | $10^{12}$ ปี | $10^{17}$ ปี |
| \$100M     | 2 มิลลิวินาที        | 2 วินาที      | 9 ชั่วโมง | 70 ปี     | $10^{11}$ ปี | $10^{16}$ ปี |
| \$1G       | .2 มิลลิวินาที       | 13 วินาที     | 1 ชั่วโมง | 7 ปี      | $10^{10}$ ปี | $10^{15}$ ปี |
| \$10G      | .02 มิลลิวินาที      | 1 วินาที      | 5.4 นาที  | 245 วัน   | $10^9$ ปี    | $10^{14}$ ปี |
| \$100G     | 2 ไมโครวินาที        | .1 วินาที     | 32 วินาที | 24 วัน    | $10^8$ ปี    | $10^{13}$ ปี |
| \$1T       | .2 ไมโครวินาที       | .01 วินาที    | 3 วินาที  | 2.4 วัน   | $10^7$ ปี    | $10^{12}$ ปี |
| \$10T      | .02 ไมโครวินาที      | 1 มิลลิวินาที | .3 วินาที | 6 ชั่วโมง | $10^6$ ปี    | $10^{11}$ ปี |
|            | วินาที               |               |           |           |              |              |

ตารางที่ 2-1 แสดงเวลาโดยประมาณที่ใช้ในการแกะรหัส เมื่อใช้ความยาวของ key และค่าใช้จ่ายที่ต่างกัน

ดังนั้น สิ่งที่น่าสนใจในการที่จะแกะรหัสอีกอย่าง คือ ค่าใช้จ่ายที่ต้องใช้ในการแกะรหัสอีกด้วย

ความปลอดภัยของการเข้ารหัสโดยใช้ Algorithm แบบ RSA

ระบบรหัสลับแบบ RSA เป็นการเข้ารหัสแบบ Trap-door ซึ่งจะทำการเปลี่ยนข้อความที่คนทั่วไปสามารถเข้าใจได้ เป็นรหัสหรือข้อความที่คนทั่วไปไม่สามารถเข้าใจ การเข้ารหัสแบบ RSA เป็นแบบ Trap-door คือ เป็นการใช้ฟังก์ชันทางคณิตศาสตร์ ซึ่งง่ายต่อการหาคำตอบ แต่ยากเมื่อต้องการทำให้กลับเป็นเหมือนเดิม ตัวอย่างเช่น การคูณเลขที่มีค่ามาก ๆ 2 จำนวน ย่อมง่ายกว่าการหาตัวประกอบของเลขที่มีค่ามาก ๆ เพียงจำนวนเดียว

ถ้าต้องการที่จะถอดรหัส โดยที่ไม่ทราบค่า Private Key จะต้องทำการแยกตัวประกอบ pq ออกเป็น p และ q ซึ่งไม่มีใครที่สามารถที่จะคิดรูปแบบ algorithm การแยกตัวประกอบที่สมบูรณ์ได้

ระยะเวลาที่ต้องใช้ในการคำนวณตัวประกอบ จะขึ้นอยู่กับขนาดของตัวเลข  $n$  เหมือน เช่นที่เรารู้กันอยู่แล้วว่า ตัวเลข  $n$  ถ้าหากมีขนาดเล็กเกินไปก็จะทำให้มีความปลอดภัยต่ำ

ในปี พ.ศ. 2520 Ron Rivest ได้กล่าวไว้ว่า สามารถที่จะหาตัวประกอบของเลข 125 หลักได้ โดยใช้เวลา  $40 \times 10^{15}$  ปี แต่ต่อมาในปี พ.ศ. 2537 ก็สามารถที่จะคำนวณการแยกตัวประกอบได้ถึง 129 หลัก แสดงให้เห็นว่าเราไม่สามารถที่จะทำการประเมินการแยกตัวประกอบในอนาคตได้ ขึ้นอยู่กับการพัฒนาทางฮาร์ดแวร์และ algorithm ด้วย

การแยกตัวประกอบที่สามารถที่จะแยกได้ ในช่วงปีที่ผ่านมา โดยใช้วิธี Number field sieve แสดงดังตารางที่ 1

| ปี        | จำนวนหลักที่สามารถที่จะแยกตัวประกอบได้ | เวลาที่ ใช้ ในการแยกตัวประกอบ(mips-ปี) |
|-----------|----------------------------------------|----------------------------------------|
| พ.ศ. 2526 | 71                                     | $> 20 \times 10^6$                     |
| พ.ศ. 2528 | 80                                     | $> 2 \times 10^6$                      |
| พ.ศ. 2531 | 90                                     | 250,000                                |
| พ.ศ. 2532 | 100                                    | 30,000                                 |
| พ.ศ. 2536 | 120                                    | 500                                    |
| พ.ศ. 2537 | 129                                    | 100                                    |

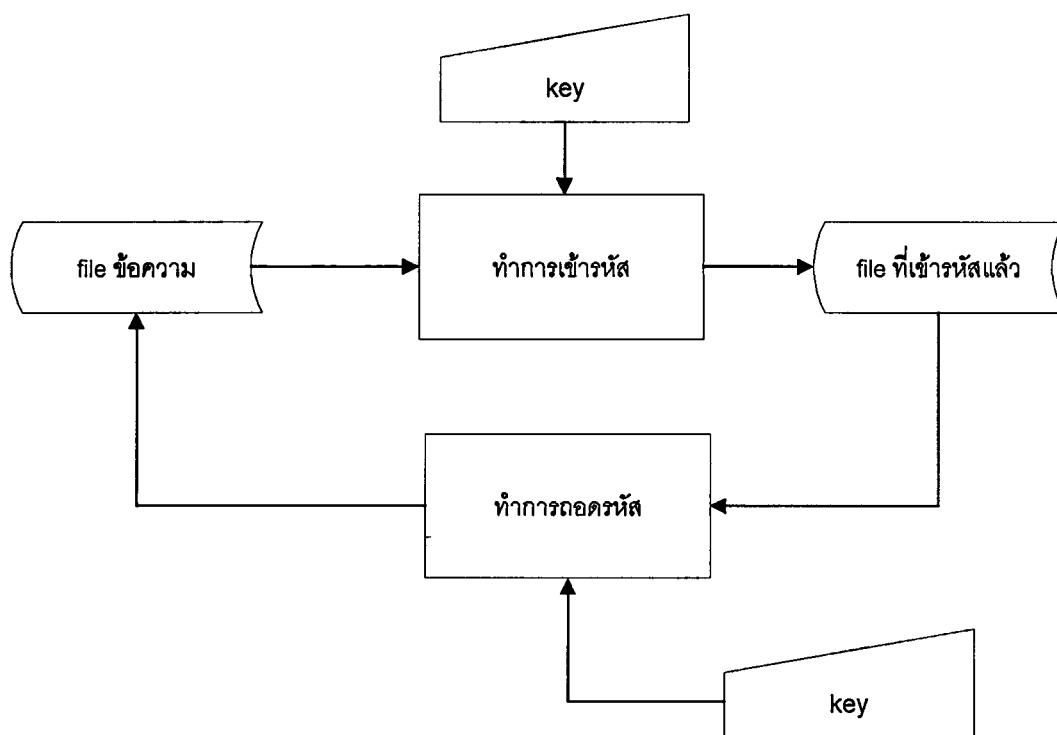
ตารางที่ 2-2 แสดงจำนวนหลักที่สามารถที่จะแยกตัวประกอบออกได้ และเวลาที่ต้องใช้ในการแยกตัวประกอบ

และในอนาคต ก็สามารถที่จะแยกตัวประกอบที่มีจำนวนหลักได้เพิ่มมากขึ้นอีก โดยที่ความปลอดภัยของการเข้ารหัสแบบ RSA ขึ้นอยู่กับการแยกตัวประกอบ และเราไม่สามารถที่จะทำการประเมินได้ว่าในอนาคตเราสามารถที่จะแยกตัวประกอบได้ถึงกี่หลัก

## บทที่ 3

### การออกแบบระบบ

#### 3.1 System Flow Diagram

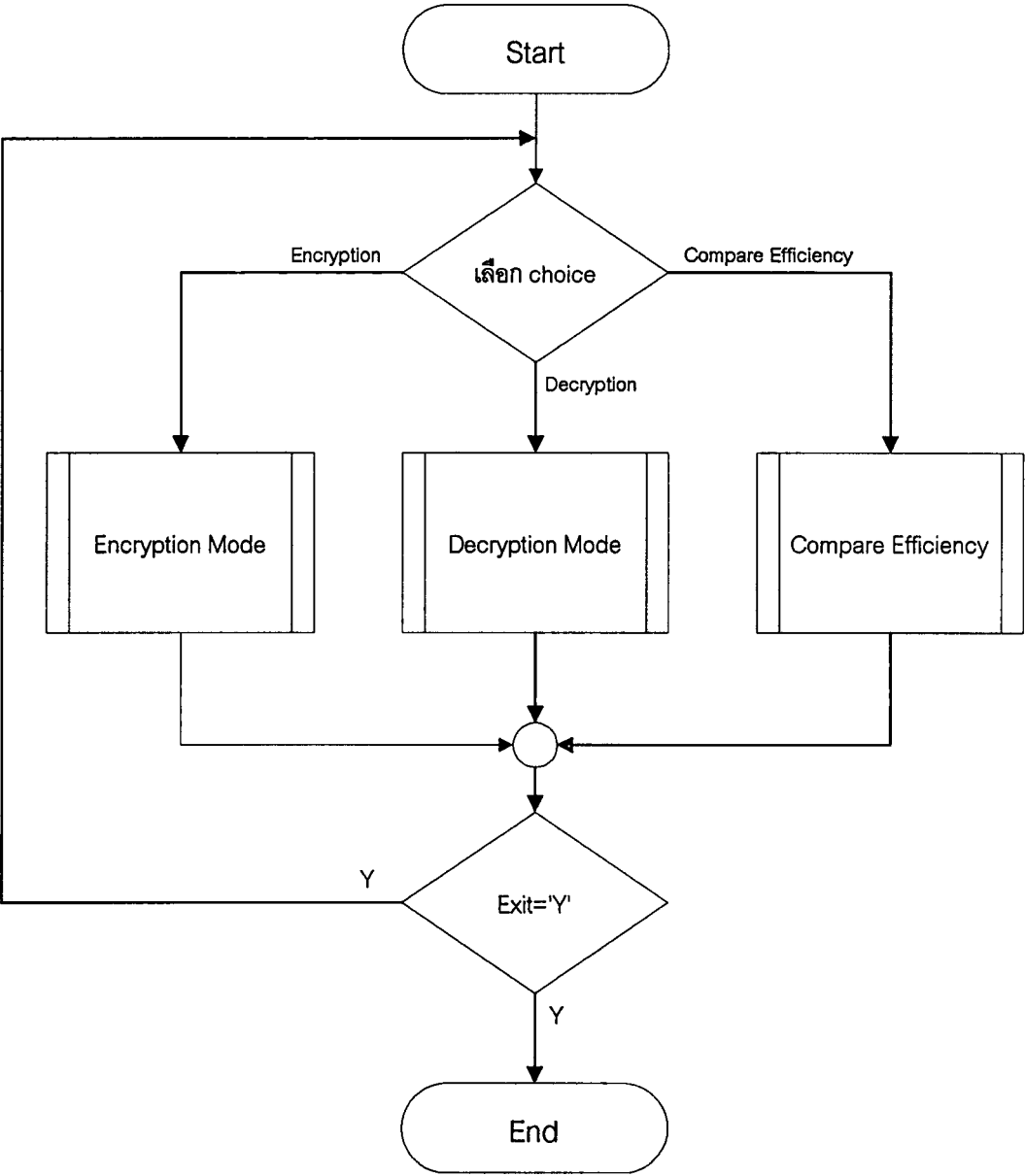


รูปที่ 3-1 แสดง System Flow Diagram ของระบบ

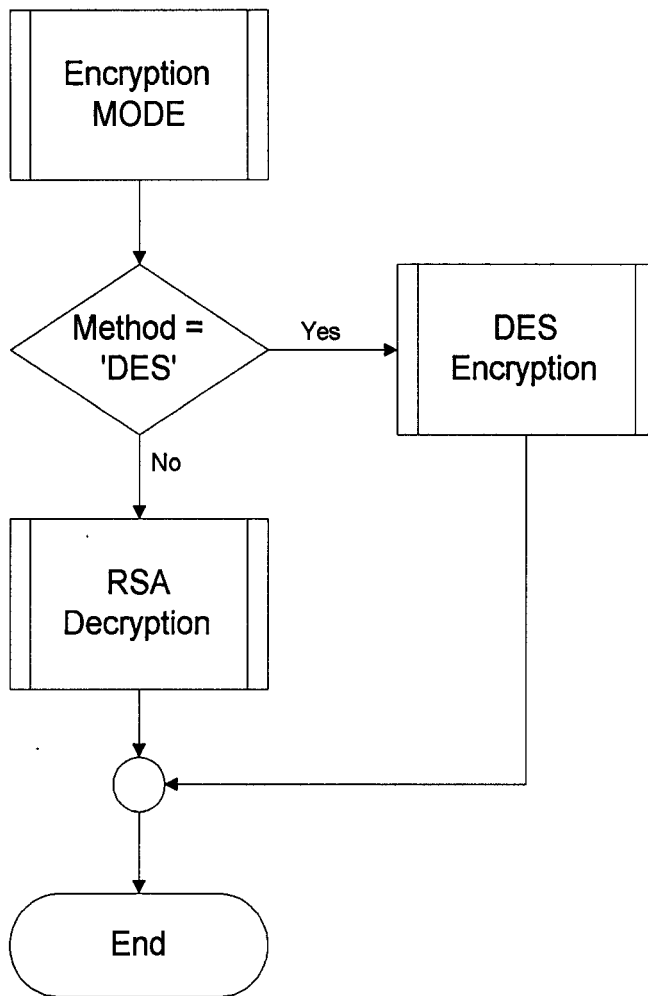
จาก System Flow Diagram สามารถอธิบายการทำงานได้ดังนี้

1. เริ่มต้นที่ผู้ใช้งานกำหนดไฟล์ข้อความ (plain text) จะทำการเข้ารหัสมาให้ และป้อนค่า Key ที่จะใช้ในการเข้ารหัส
2. ทำการเข้ารหัสไฟล์จาก Key ที่กำหนดให้ ผลลัพธ์ที่ได้จะถูกเขียนลงไฟล์
3. เมื่อจะทำการถอดรหัส ก็จะนำไฟล์ที่ทำการเข้ารหัสไว้ และรับ Key จากผู้ใช้งาน ไปทำการถอดรหัส
4. ผลลัพธ์ที่ได้ จะถูกเก็บไว้ในเป็นไฟล์ข้อความปกติ

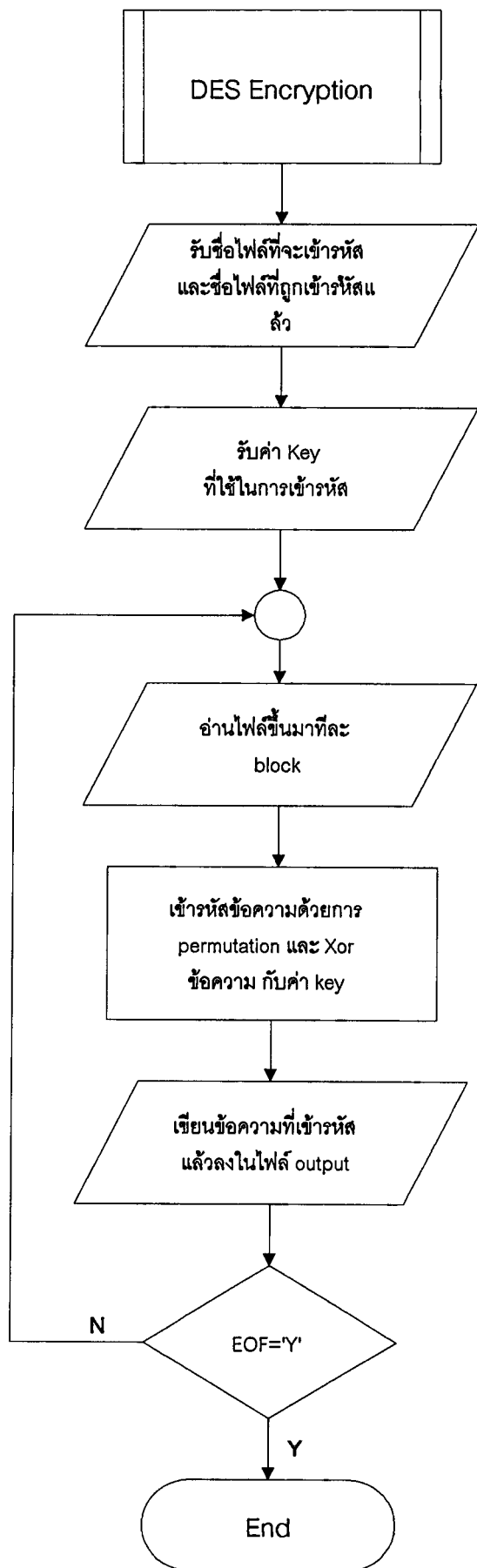
3.2 Flow Chart



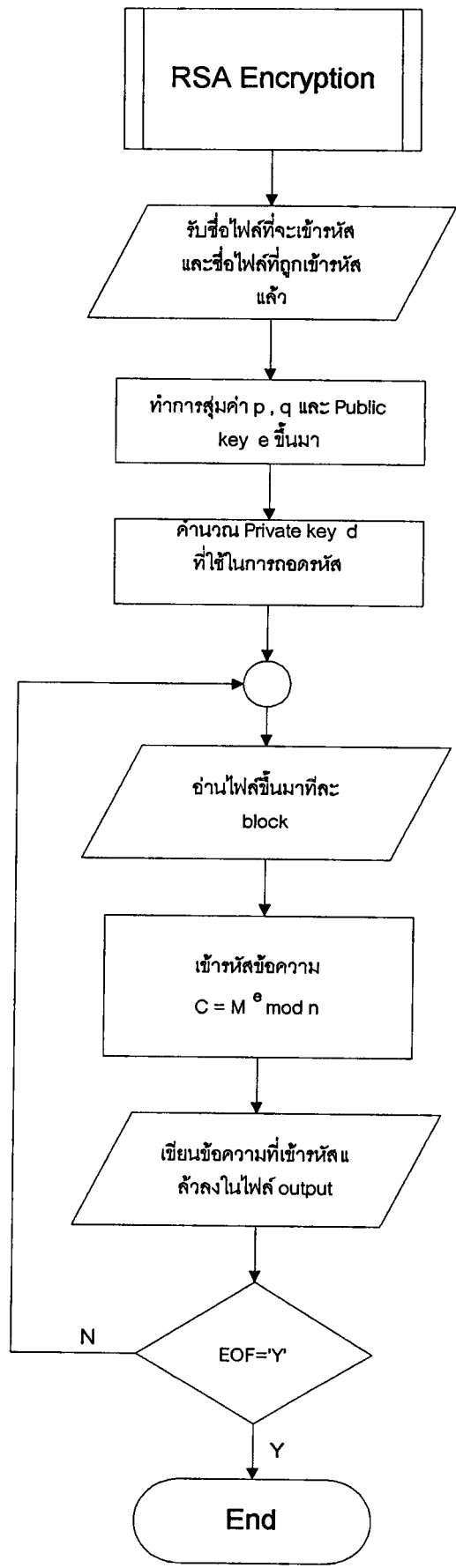
รูปที่ 3-2 แสดง Flow Chart ของระบบ



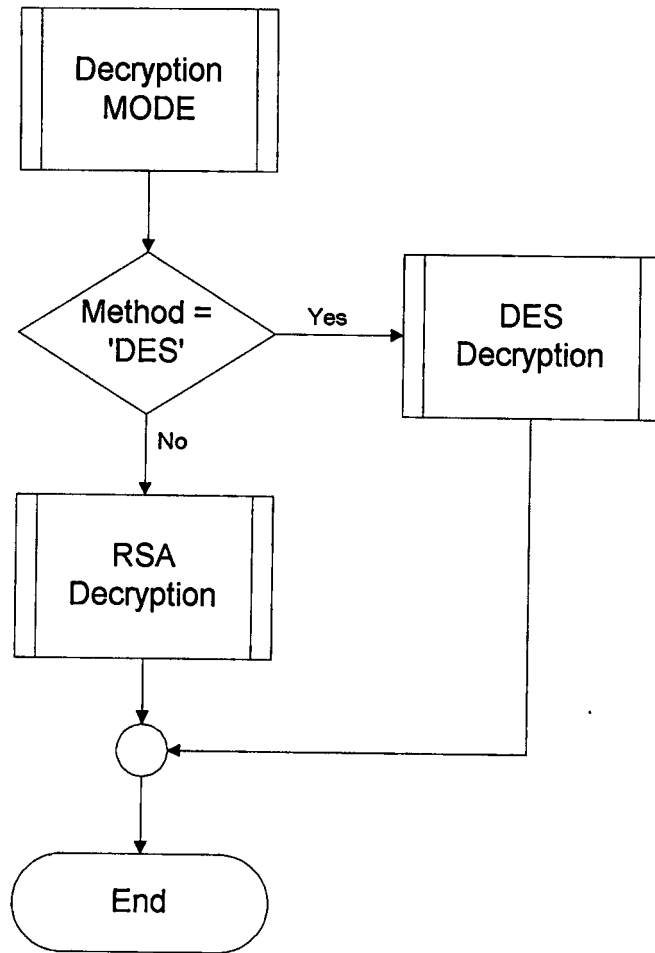
รูปที่ 3-2 แสดง Flow Chart ของการเข้ารหัส



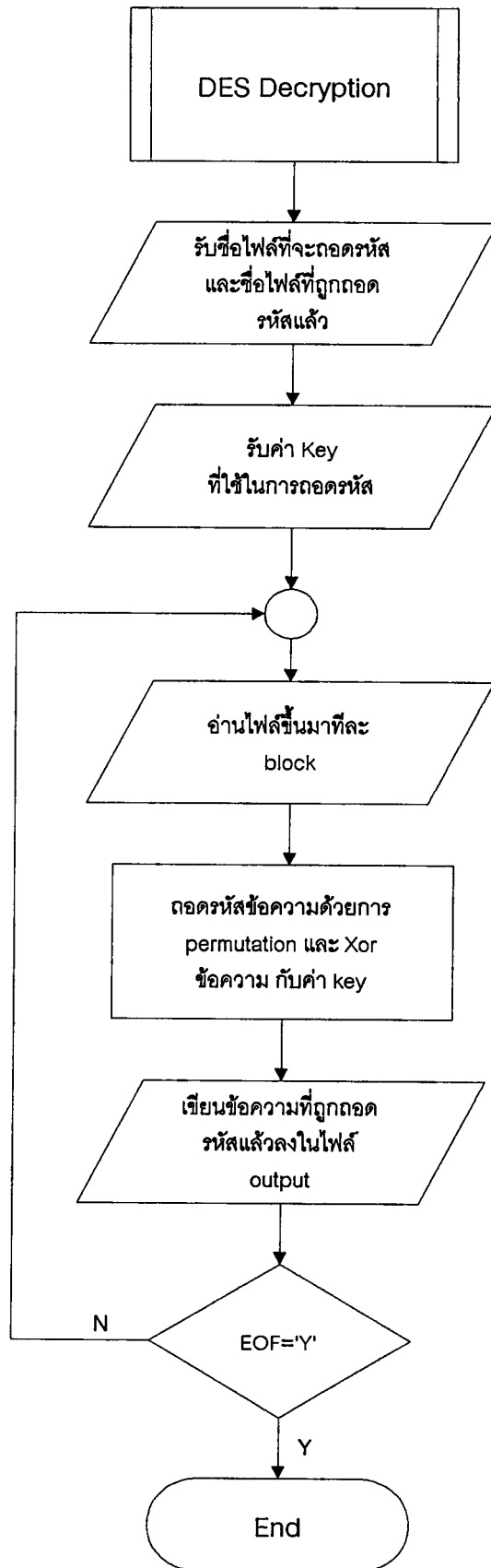
รูปที่ 3-3 แสดง Flow Chart ของการเข้ารหัสด้วย Algorithm แบบ DES



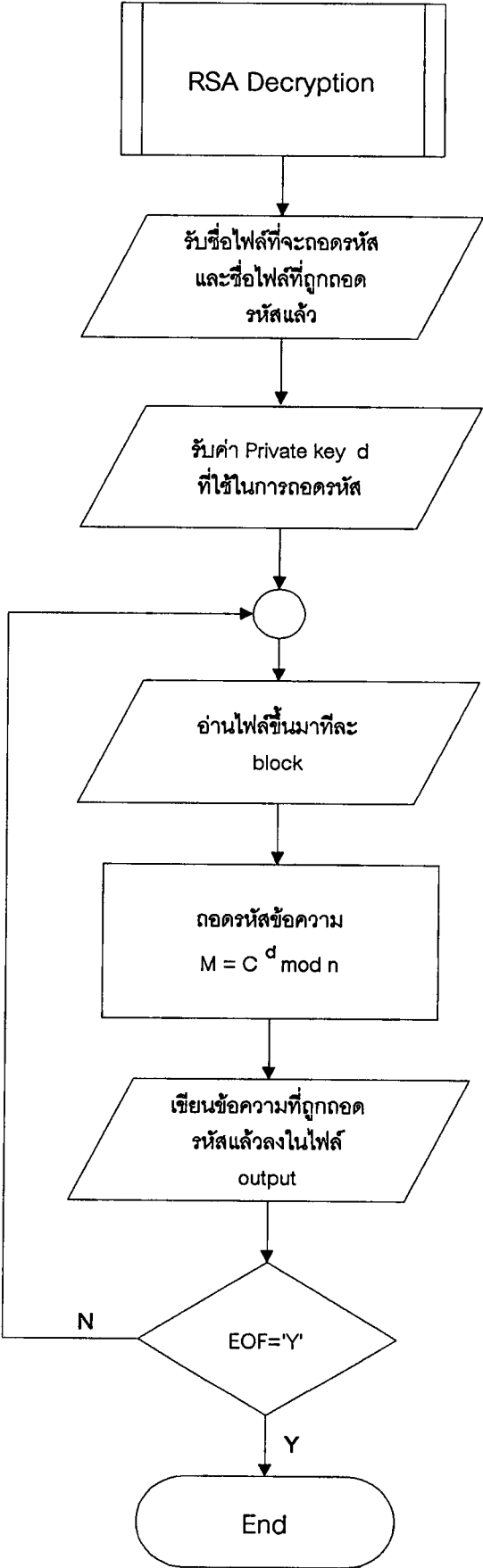
รูปที่ 3-4 แสดง Flow Chart ของการเข้ารหัสด้วย Algorithm แบบ RSA



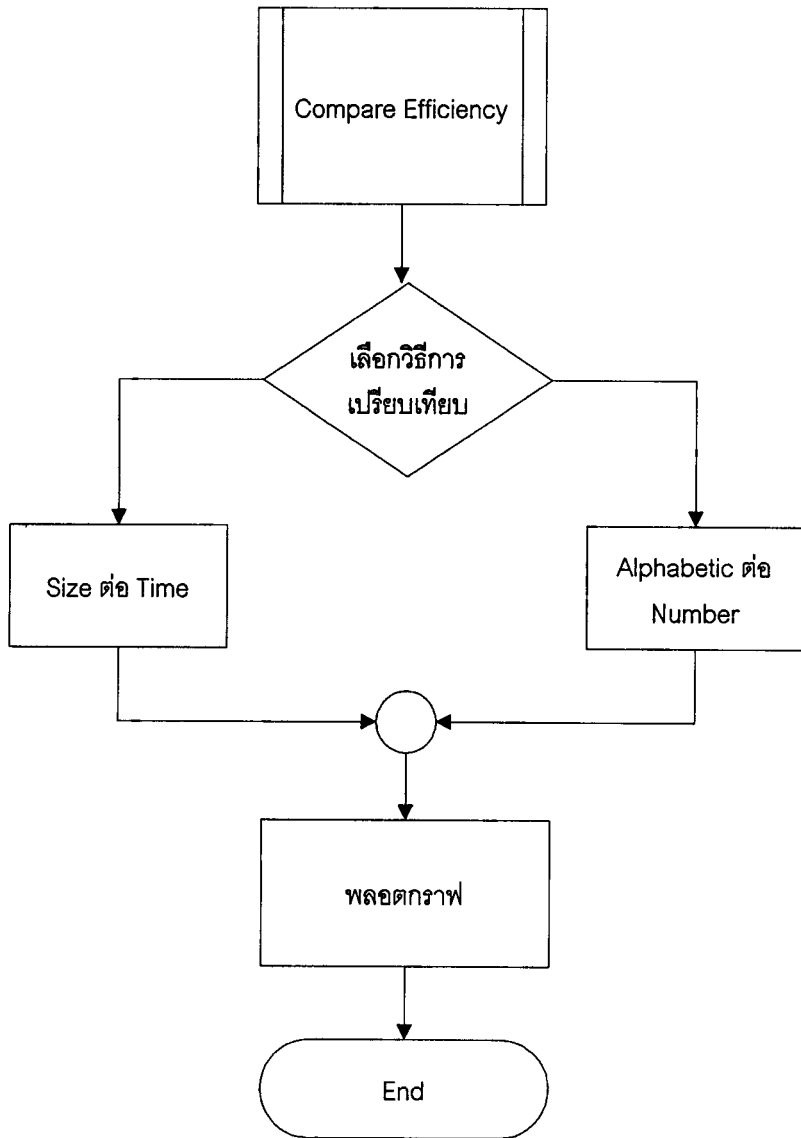
รูปที่ 3-5 แสดง Flow Chart ของการถอดรหัส



รูปที่ 3-6 แสดง Flow Chart ของการถอดรหัสด้วย Algorithm แบบ DES



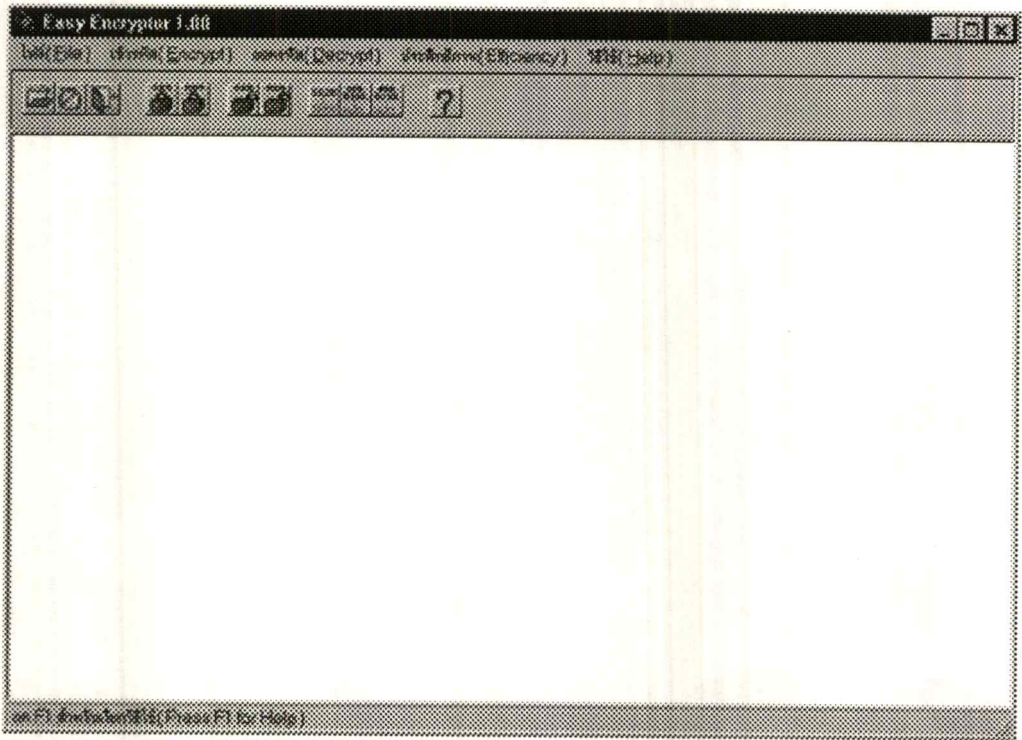
รูปที่ 3-7 แสดง Flow Chart ของการถอดรหัสด้วย Algorithm แบบ RSA



รูปที่ 3-8 แสดง Flow Chart ของการเปรียบเทียบการเข้ารหัสถอดรหัส โดยใช้ algorithm แบบ DES และ RSA

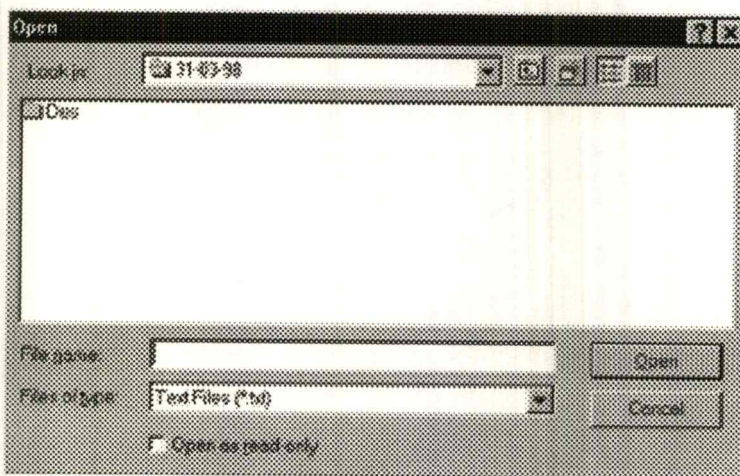
หน้าจอของโปรแกรมที่ติดต่อกับผู้ใช้

1. หน้าจอเริ่มแรกเมื่อเข้าสู่โปรแกรม



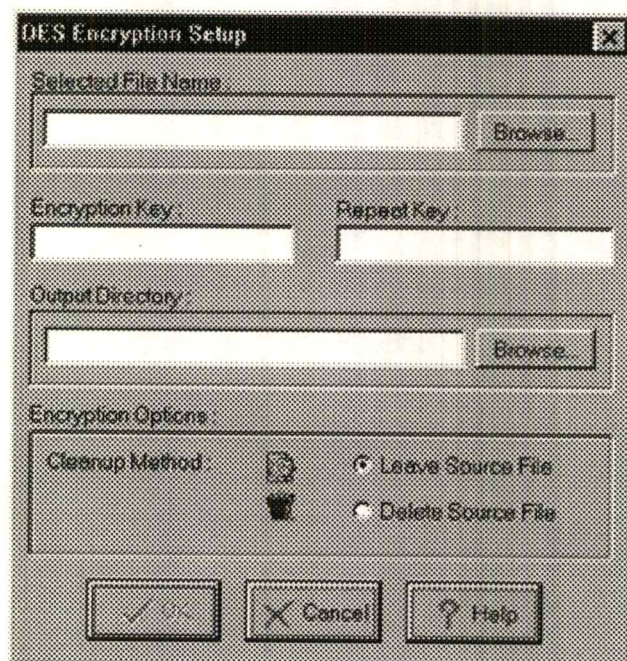
รูปที่ 3-9 แสดงหน้าจอเริ่มแรกเมื่อเข้าสู่โปรแกรม

2. เมนู ไฟล์ (File) เมื่อเลือก Select File หน้าจอจะแสดงดังรูปที่ 3-10



รูปที่ 3-10 หน้าจอแสดงการเลือกไฟล์

3. เมนู Encrypt เมื่อเลือกที่จะทำการ Encrypt แบบ DES จะแสดงหน้าจอการรับชื่อไฟล์, key และ Encryption option ที่ใช้ในการเข้ารหัสแบบ DES ดังรูปที่ 3-11



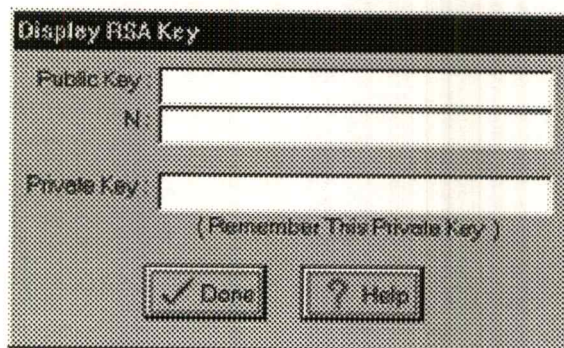
รูปที่ 3-11 แสดงหน้าจอการรับชื่อไฟล์, key และ Encryption option ที่ใช้ในการเข้ารหัสแบบ DES

4. เมนู Encrypt เมื่อเลือกที่จะทำการ Encrypt แบบ RSA จะแสดงหน้าจอการรับชื่อไฟล์ และ Encryption option ที่ใช้ในการเข้ารหัสแบบ RSA ดังรูปที่ 3-12



รูปที่ 3-12 แสดงหน้าจอการรับชื่อไฟล์, key และ Encryption option ที่ใช้ในการเข้ารหัสแบบ RSA

5. เมื่อเลือกที่จะทำการ Encrypt แบบ RSA และทำการเลือกไฟล์แล้ว จะแสดงหน้าจอแสดง Public key , Private key และค่า n ของ RSA ดังรูปที่ 3-13



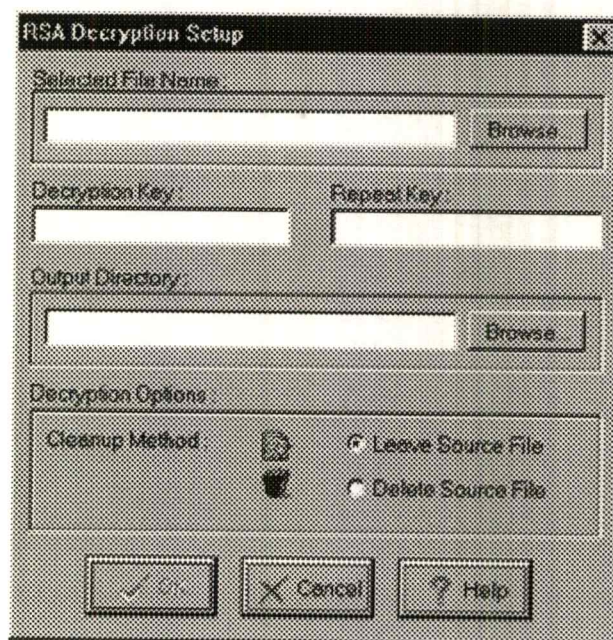
รูปที่ 3-13 แสดงหน้าจอการแสดง Public key , Private key และค่า n ของ RSA

6. เมนู Decrypt เมื่อเลือกที่จะทำการ Decrypt แบบ DES จะแสดงหน้าจอการรับชื่อไฟล์, key และ Decryption option ที่ใช้ในการถอดรหัสแบบ DES ดังรูปที่ 3-14



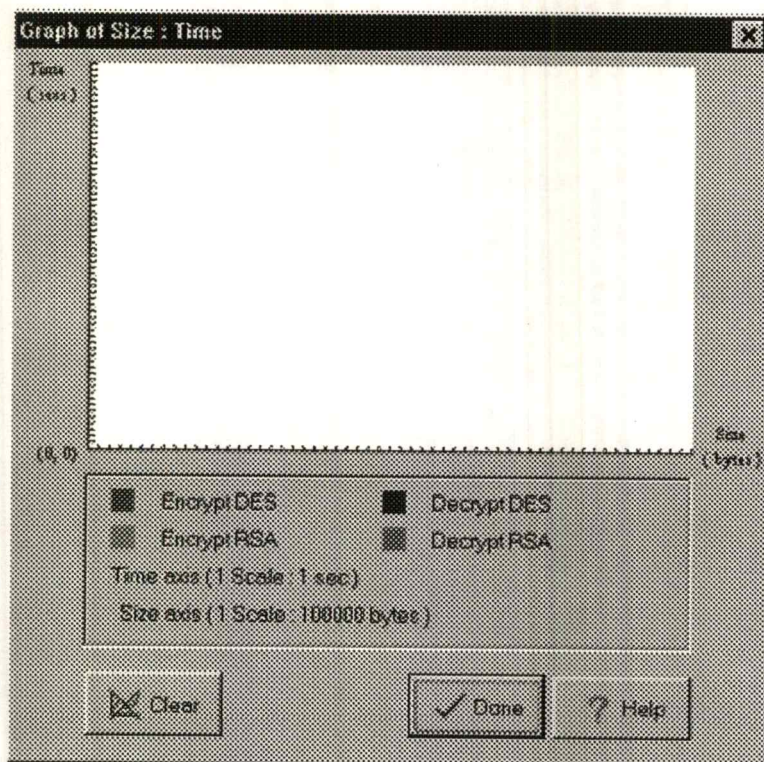
รูปที่ 3-14 แสดงหน้าจอการรับชื่อไฟล์, key และ Decryption option ที่ใช้ในการถอดรหัสแบบ DES

7. เมนู Decrypt เมื่อเลือกที่จะทำการ Decrypt แบบ DES จะแสดงหน้าจอการรับชื่อไฟล์, key และ Decryption option ที่ใช้ในการถอดรหัสแบบ RSA ดังรูปที่ 3-15



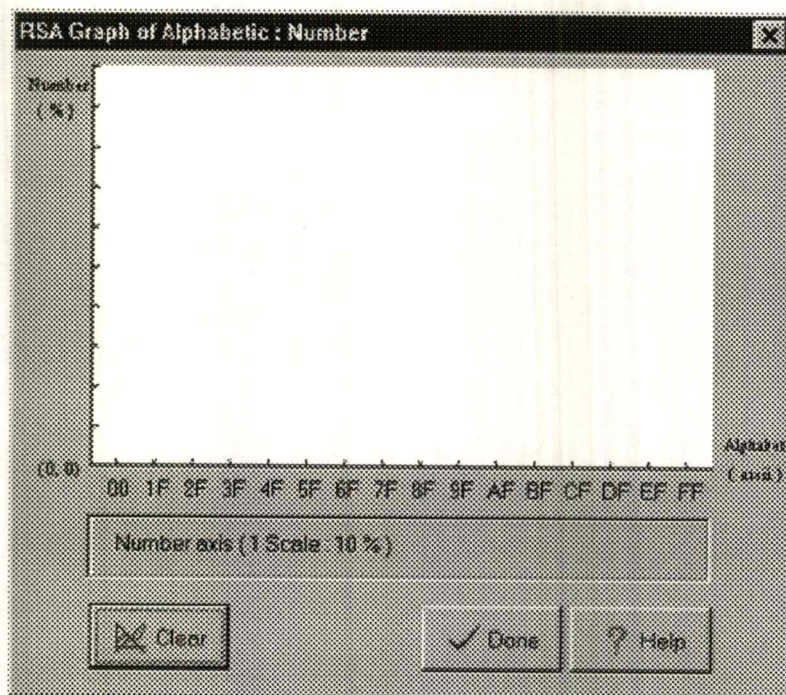
รูปที่ 3-15 แสดงหน้าจอการรับชื่อไฟล์, key และ Decryption option ที่ใช้ในการถอดรหัสแบบ RSA

8. เมนู Efficiency เมื่อเลือกที่จะทำการเปรียบเทียบด้วยขนาดต่อเวลา หน้าจอแสดงกราฟของการเปรียบเทียบขนาดไฟล์ต่อเวลาในการเข้ารหัสและถอดรหัสแบบ DES และ RSA ดังรูปที่ 3-16



รูปที่ 3-16 แสดงหน้าจอแสดงกราฟของการเปรียบเทียบขนาดไฟล์ต่อเวลาในการเข้ารหัสและถอดรหัสแบบ DES และ RSA

9. เมนู Efficiency เมื่อเลือกที่จะทำการเปรียบเทียบด้วยการกระจายตัวอักษร หน้าจอแสดงกราฟของการเปรียบเทียบการกระจายตัวอักษรของการเข้ารหัสแบบ DES และ RSA จะแสดงดังรูปที่ 3-17



รูปที่ 3-17 แสดงหน้าจอแสดงกราฟของการเปรียบเทียบการกระจายตัวอักษรของการเข้ารหัสแบบ DES และ RSA

## บทที่ 4

### ผลการทำงานของระบบ

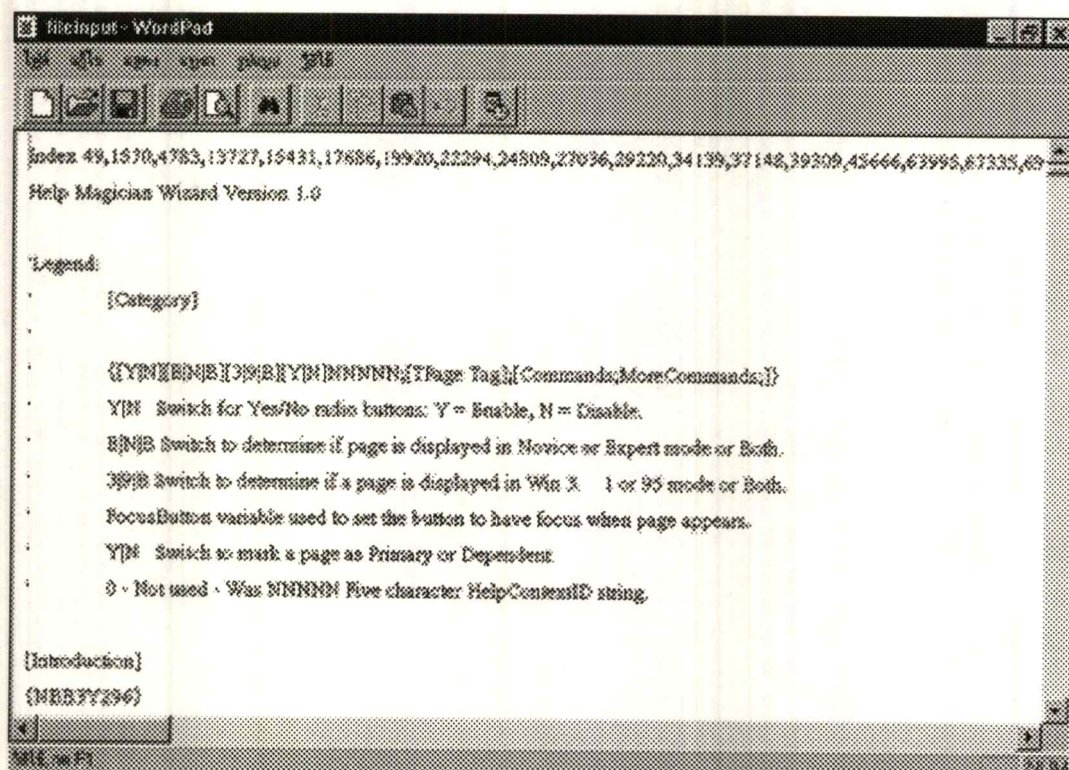
#### ข้อจำกัดของการใช้ระบบ

1. ข้อมูลที่จะใช้ในการเข้ารหัส ต้องเป็นไฟล์ชนิด ไฟล์ข้อความ (text file) เท่านั้น
2. ข้อมูลที่จะเข้ารหัสแบบ RSA จะเป็นภาษาไทย หรือภาษาอังกฤษก็ได้ แต่การเข้ารหัสแบบ DES จะต้องเป็นข้อมูลภาษาอังกฤษเท่านั้น ไม่สามารถที่จะเข้ารหัสข้อมูลภาษาไทยได้
3. การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES นั้น Key ที่ใช้จะต้องมีขนาดอย่างน้อย 8 ตัวอักษร แต่ไม่เกิน 28 ตัวอักษร
4. ข้อมูลที่จะทำการเข้ารหัสและค่า Key ที่ใช้ จะต้องอยู่ในช่วงเลขฐานสิบ ตั้งแต่ 32-122 ของตารางแอสกี สัญลักษณ์อื่น นอกจากนี้จะไม่ถูกคำนึงถึงในการเข้ารหัส-ถอดรหัส
5. ฮาร์ดแวร์ที่ใช้กับระบบนี้ จะต้องมีลักษณะคือ Cpu รุ่น 486 ขึ้นไป มีเนื้อที่ว่างในฮาร์ดดิสก์ตั้งแต่ 10 Mb ขึ้นไป

#### ผลการทำงานของระบบ

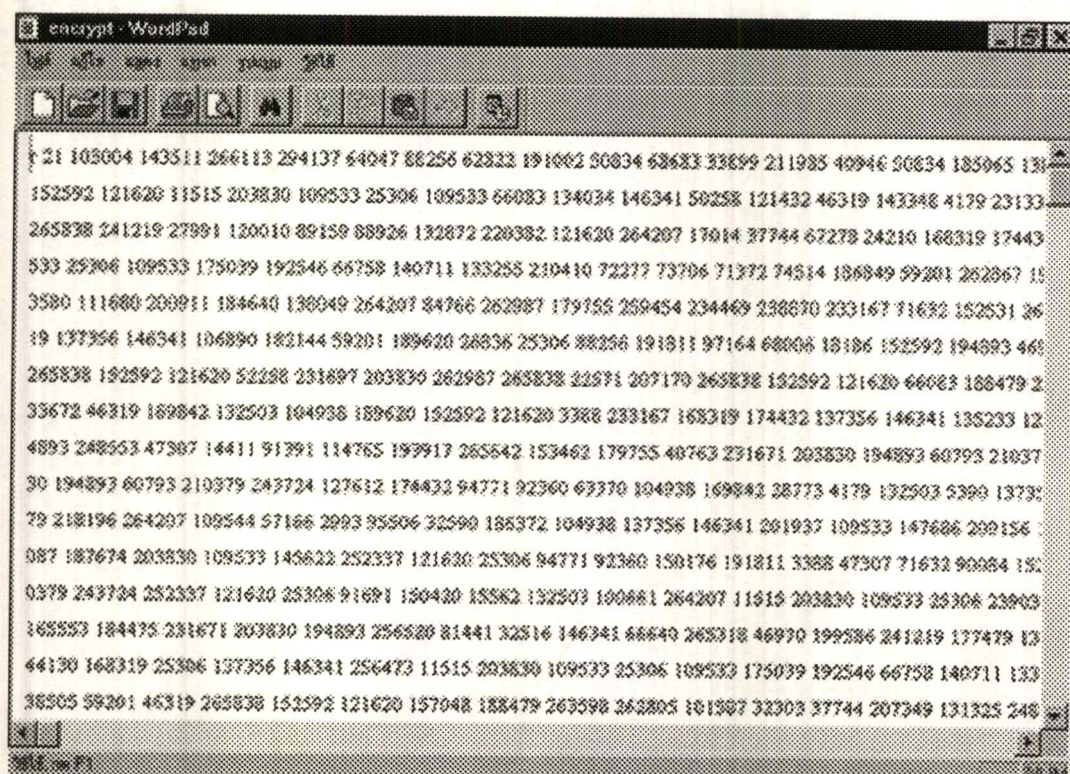
ซอฟต์แวร์ที่จัดทำขึ้นนี้ สามารถที่จะทำการเข้ารหัสและถอดรหัส โดยใช้ Algorithm แบบ DES และ RSA ได้ จุดประสงค์ของการพัฒนาซอฟต์แวร์นี้ คือ เพื่อใช้ในการเปรียบเทียบการเข้ารหัส-ถอดรหัส ระหว่าง Algorithm ทั้ง 2 แบบ

การทำงานของซอฟต์แวร์ เมื่อเลือกไฟล์ต้องการที่จะทำการเข้ารหัส ข้อความที่ทำการเข้ารหัสจะเป็น plaintext ที่สามารถอ่านได้ ซึ่งปรากฏดังรูปที่ 4-1



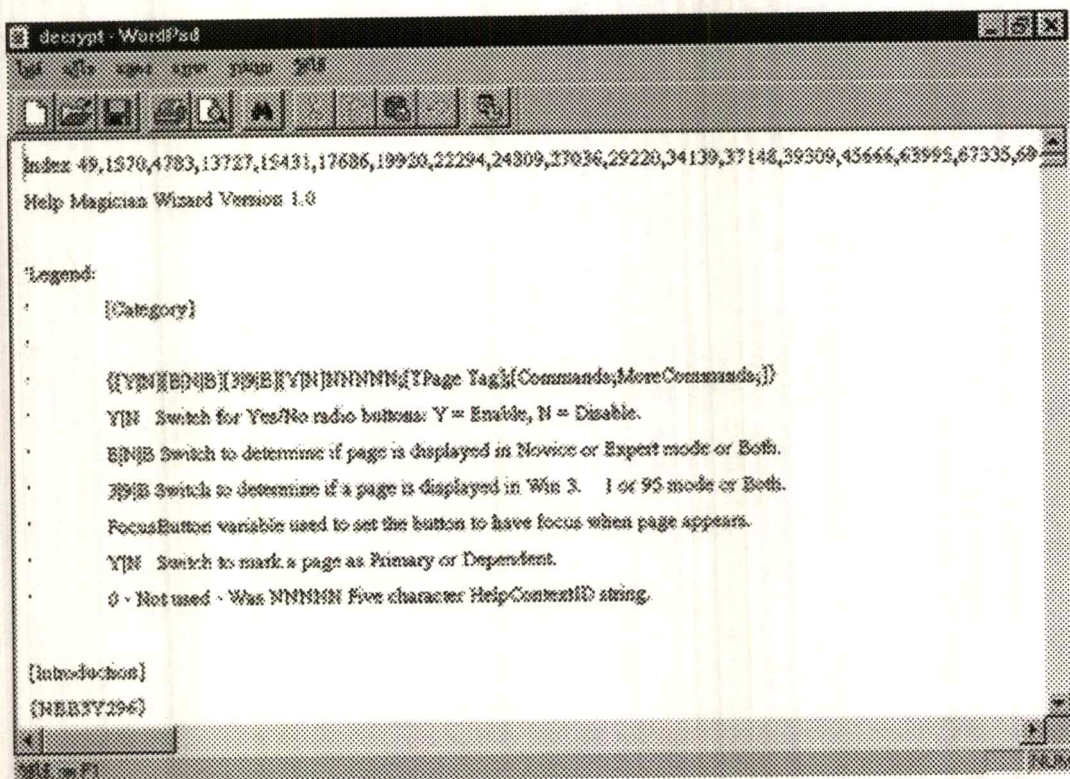
รูปที่ 4-1 แสดงข้อความในไฟล์ก่อนที่จะถูกทำการเข้ารหัส

ถ้าเป็นการเข้ารหัสไฟล์โดยใช้ algorithm แบบ RSA ไฟล์ที่ถูกเข้ารหัสแล้ว แสดงดังรูปที่ 4-2



รูปที่ 4-2 แสดงข้อความในไฟล์หลังจากถูกทำการเข้ารหัสโดยใช้ algorithm แบบ RSA

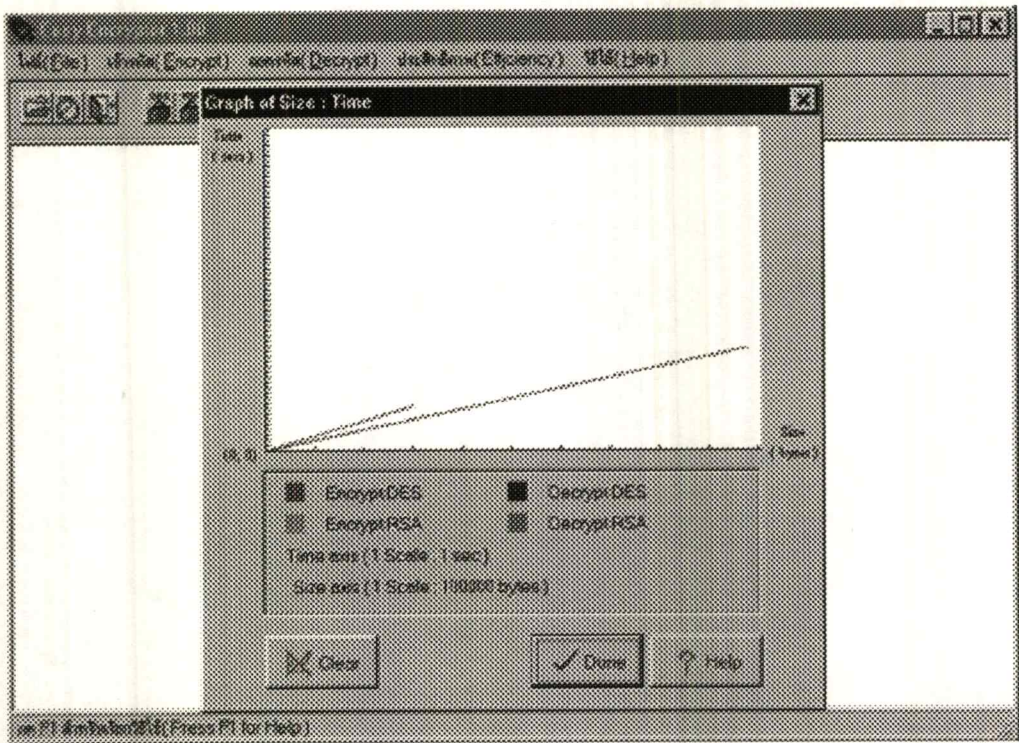
และเมื่อเราทำการใช้ key ที่ถูกต้องในการถอดรหัส เราจะได้ไฟล์ข้อความเดิมกลับมา  
ดังรูปที่ 4-3



รูปที่ 4-3 แสดงข้อความในไฟล์หลังจากที่ทำการถอดรหัสแล้ว

ผลจากการทดลองใช้ซอฟต์แวร์พบว่า

1. การเข้ารหัส จะใช้เวลามากกว่าการถอดรหัส และจะทำให้ขนาดไฟล์ที่ถูกเข้ารหัสแล้วมีขนาดใหญ่ขึ้น แสดงดังรูปที่ 4-4



รูปที่ 4-4 แสดงขนาดต่อเวลาในการเข้ารหัสแบบ RSA

2. การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES นั้น เร็วกว่าการเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA เพราะ
  - การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA ต้องทำการหาค่า  $p, q$  และ  $e$  ที่เหมาะสมในการเข้ารหัส ถ้าค่า  $p, q$  และ  $e$  ที่หามาได้ ไม่เหมาะสม จะต้องทำการหาค่าใหม่ ทำให้การทำงานของระบบช้าลง เนื่องจากต้องทำการหาค่าใหม่จนกว่าจะได้ค่าที่เหมาะสม
  - การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA มีการคำนวณทางคณิตศาสตร์มาก ทั้งการทดสอบจำนวนเฉพาะ การหาโมดูล และการแยกตัวประกอบของจำนวนเต็มที่มีขนาดใหญ่ รวมทั้งยังต้องทำการคำนวณซ้ำ ๆ กันหลาย ๆ รอบ จึงทำให้การทำงานของระบบช้าลง ส่วนการเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES นั้น ใช้ในการสลับบิต และการ Xor เป็นการคำนวณทางคณิตศาสตร์อย่างง่าย จึงทำให้การทำงานนั้นรวดเร็วกว่า
3. คุณสมบัติของฮาร์ดแวร์ที่ต้องการ การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES มีความต้องการทางฮาร์ดแวร์ที่ต่ำกว่า การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA เพราะ การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA ต้องทำการคำนวณทางคณิตศาสตร์มากกว่า

4. การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES นั้นเหมาะสมที่จะนำไปใช้ในการเข้ารหัสข่าวสารที่ถูกเก็บไว้ในดิสก์ การเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA นั้นเหมาะสมที่จะนำไปใช้ในการเข้ารหัสทางการติดต่อและจดหมายมากกว่า

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### สรุปผลการทำปัญหาพิเศษ

การทำปัญหาพิเศษนี้ มีจุดมุ่งหมายคือเพื่อจัดทำซอฟต์แวร์ที่ทำการเปรียบเทียบการเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ DES และการเข้ารหัส-ถอดรหัส โดยใช้ Algorithm แบบ RSA หลังจากที่ได้จัดทำซอฟต์แวร์นี้แล้ว พบว่าสามารถที่จะเข้ารหัสไฟล์ข้อความปกติ ให้เป็นรหัสลับ และสามารถแปลงรหัสลับกลับเป็นข้อความเดิมได้ โดยการใช้ algorithm แบบ RSA และสามารถที่จะทำการเปรียบเทียบการเข้ารหัสโดยใช้ algorithm แบบ DES และแบบ RSA ได้ โดยการเปรียบเทียบ ขนาดไฟล์ ต่อระยะเวลาที่ใช้ในการเข้ารหัส และความถี่ของการกระจายช่วงตัวอักษรเมื่อทำการเข้ารหัสแล้ว ซึ่งได้นำไปพลอตเป็นกราฟเพื่อความสะดวกในการเปรียบเทียบ

#### ปัญหาและข้อเสนอแนะ

ในการจัดทำซอฟต์แวร์ได้พบปัญหาต่าง ๆ ดังนี้

1. ในการเข้ารหัส-ถอดรหัสโดยใช้ algorithm แบบ RSA นั้น ไม่สามารถที่จะทำการคำนวณหาตัวประกอบของค่าที่มีจำนวนมากได้ ทำให้ค่า  $p$  และ  $q$  จะต้องมียกไม่มาก
2. ในการเข้ารหัส-ถอดรหัสโดยใช้ algorithm แบบ RSA นั้น ไม่สามารถที่จะให้ผู้ใช้เลือก Public key ได้ เพราะบางครั้งค่า Public key ที่ถูกกำหนดนั้นไม่สามารถที่จะหาตัวผกผัน (Private key) ได้

และถ้าต้องการให้ซอฟต์แวร์นี้ มีประสิทธิภาพมากขึ้น ควรปรับปรุงแก้ไข ดังนี้

1. เมื่อใช้ซอฟต์แวร์นี้ในการเข้ารหัสแล้ว จะทำให้ไฟล์ที่เข้ารหัสแล้วมีขนาดใหญ่ขึ้นกว่าเดิม ดังนั้น จึงควรปรับปรุงให้ไฟล์ที่เข้ารหัสนี้มีขนาดเท่ากับหรือน้อยกว่าข้อมูลเดิม
2. ซอฟต์แวร์นี้ ไม่สามารถที่จะเข้ารหัสไฟล์ที่ไม่ใช่ไฟล์ข้อความได้ ดังนั้น จึงควรปรับปรุงให้เข้ารหัสไฟล์ที่เป็น binary ไฟล์ได้
3. ซอฟต์แวร์นี้ สามารถที่จะเข้ารหัสแบบ DES ได้เฉพาะข้อความภาษาอังกฤษได้เท่านั้น ดังนั้น จึงควรปรับปรุงให้เข้ารหัสข้อความภาษาไทยได้
4. ในการเข้ารหัส-ถอดรหัสโดยใช้ algorithm แบบ RSA นั้น ควรให้ผู้ใช้ได้เป็นผู้กำหนดค่า Public Key ที่ต้องการ

5. ในการเข้ารหัส-ถอดรหัสโดยใช้ algorithm แบบ DES นั้น เมื่อผู้ใช้ป้อนค่า Key ที่ใช้ในการเข้ารหัสมานั้น ควรจะตรวจสอบดูก่อนว่าเป็น Weak Key หรือ Semiweak Key หรือไม่ ก่อนที่จะทำการเข้ารหัส

# ภาคผนวก ก คำศัพท์

## Cryptography

วิชาที่ว่าด้วยการเข้ารหัสและถอดรหัส

## Ciphertext

ผลลัพธ์ของการเข้ารหัส Ciphertext จะปิดบังข้อความตอนแรกไว้ภายใต้การควบคุมของ Key ถ้าปราศจาก Key ก็จะไม่สามารถทราบข้อความเดิมจาก Ciphertext ได้

## DES (Data Encryption Standard)

การเข้ารหัสซึ่งจะเข้ารหัสข้อมูลทีละ 64 bit ด้วย Key 56 bit ทั้งหมด 16 รอบ

## Decryption

การถอดข้อมูลข่าวสารที่ถูกเข้ารหัสไว้

## Encryption

การเข้ารหัสข้อมูลข่าวสารที่ต้องการให้เป็นความลับ

## Key

เหมือนกับหลักการทั่วไปว่าเราจะรักษาความปลอดภัยของสิ่งต่าง ๆ ด้วยการ lock ดังนั้นสิ่งที่สามารถจะปลดมันได้ก็คือ Key ที่ถูกต้อง

ในเรื่องรหัสลับนี้ Key คือ กุญแจที่ผู้ใช้ป้อนเข้ามาเพื่อใช้ในการเข้ารหัสและถอดรหัส

## Key Schedule

ตารางรวบรวม Table ของ Key ซึ่งจะแตกต่างกันไปทั้ง 16 รอบ

## Permutation

การกระทำทางคณิตศาสตร์ เป็นการสลับที่ของสัญลักษณ์ ถ้ามีสัญลักษณ์อยู่ทั้งหมด  $n$  ตัว มันจะมีการ permutation ที่เป็นไปได้อยู่  $n*(n-1)*(n-2)*...*2*1$

ในเรื่องรหัสลับนี้ เป็นการสลับบิตตามค่าที่ได้จากการสุ่ม

## Plaintext

ข้อความปกติตอนแรกก่อนที่จะทำการเข้ารหัส สามารถอ่านได้

## Symmetric Key Encryption

เป็นการเข้ารหัสแบบสมมาตร คือ key ที่ใช้ในการเข้ารหัสและ key ที่ใช้ในการถอดรหัส เป็น key เดียวกัน

## Public key encryption

เป็นการเข้ารหัสแบบอสมมาตร ต้องใช้ key 2 อันในการเข้ารหัสและถอดรหัส key ที่ใช้ในการเข้ารหัสจะเรียกว่า public key ส่วน key ที่ใช้ในการถอดรหัส เรียกว่า private key

## Public key

key ที่ใช้ในการเข้ารหัสแบบ Public key encryption

## Private key

key ที่ใช้ในการถอดรหัส Public key encryption

## RSA (Rivest ,Sharmir, and Adleman)

การเข้ารหัสข้อมูลแบบ public key โดยคำนวณ Key จากจำนวนเฉพาะ  $p, q$  และจำนวนขนาดใหญ่  $e, d$

ภาคผนวก ข ตาราง

Table Initial Permutation (IP)

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

Table PC-1

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

Table PC-2

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

(8 บิตอื่น ๆ จะถูกละทิ้ง)

Table K<sub>1</sub>

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

Table K<sub>3</sub>

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

Table K<sub>4</sub>

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

Table K<sub>6</sub>

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

Table K<sub>6</sub>

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

Table K<sub>7</sub>

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

Table K<sub>8</sub>

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

Table K<sub>9</sub>

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

Table K<sub>10</sub>

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

Table K<sub>11</sub>

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

Table K<sub>12</sub>

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

Table K<sub>13</sub>

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

Table K<sub>14</sub>

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

Table K<sub>15</sub>

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

Table K<sub>16</sub>

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

Table expasion (E)

| บิตที่ | เปลี่ยนเป็น<br>บิตที่ | บิตที่ | เปลี่ยนเป็น<br>บิตที่ | บิตที่ | เปลี่ยนเป็น<br>บิตที่ | บิตที่ | เปลี่ยนเป็น<br>บิตที่ |
|--------|-----------------------|--------|-----------------------|--------|-----------------------|--------|-----------------------|
| 32     | 1                     | 8      | 13                    | 16     | 25                    | 24     | 37                    |
| 1      | 2                     | 9      | 14                    | 17     | 26                    | 25     | 38                    |
| 2      | 3                     | 10     | 15                    | 18     | 27                    | 26     | 39                    |
| 3      | 4                     | 11     | 16                    | 19     | 28                    | 27     | 40                    |
| 4      | 5                     | 12     | 17                    | 20     | 29                    | 28     | 41                    |
| 5      | 6                     | 13     | 18                    | 21     | 30                    | 29     | 42                    |
| 4      | 7                     | 12     | 19                    | 20     | 31                    | 28     | 43                    |
| 5      | 8                     | 13     | 20                    | 21     | 32                    | 29     | 44                    |
| 6      | 9                     | 14     | 21                    | 22     | 33                    | 30     | 45                    |
| 7      | 10                    | 15     | 22                    | 23     | 34                    | 31     | 46                    |
| 8      | 11                    | 16     | 23                    | 24     | 35                    | 32     | 47                    |
| 9      | 12                    | 17     | 24                    | 25     | 36                    | 1      | 48                    |

DES : S-box 1

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 14 | 0  | 4  | 15 |
| 0001     | 1      | 4  | 15 | 1  | 12 |
| 0010     | 2      | 13 |    | 14 | 8  |
| 0011     | 3      | 1  | 4  | 8  | 2  |
| 0100     | 4      | 2  | 14 | 13 | 4  |
| 0101     | 5      | 15 | 2  | 6  | 9  |
| 0110     | 6      | 11 | 13 | 2  | 1  |
| 0111     | 7      | 8  | 1  | 11 | 7  |
| 1000     | 8      | 3  | 10 | 15 | 5  |
| 1001     | 9      | 10 | 6  | 12 | 11 |
| 1010     | 10     | 6  | 12 | 9  | 3  |
| 1011     | 11     | 12 | 11 | 7  | 14 |
| 1100     | 12     | 5  | 9  | 3  | 10 |
| 1101     | 13     | 9  | 5  | 10 | 0  |
| 1110     | 14     | 0  | 3  | 5  | 6  |
| 1111     | 15     | 7  | 8  | 0  | 13 |

## DES : S-box 2

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 15 | 3  | 0  | 13 |
| 0001     | 1      | 1  | 13 | 14 | 8  |
| 0010     | 2      | 8  | 4  | 7  | 10 |
| 0011     | 3      | 14 | 7  | 11 | 1  |
| 0100     | 4      | 6  | 15 | 10 | 3  |
| 0101     | 5      | 11 | 2  | 4  | 15 |
| 0110     | 6      | 3  | 8  | 13 | 4  |
| 0111     | 7      | 4  | 14 | 1  | 2  |
| 1000     | 8      | 9  | 12 | 5  | 11 |
| 1001     | 9      | 7  | 0  | 8  | 6  |
| 1010     | 10     | 2  | 1  | 12 | 7  |
| 1011     | 11     | 13 | 10 | 6  | 12 |
| 1100     | 12     | 12 | 6  | 9  | 0  |
| 1101     | 13     | 0  | 9  | 3  | 5  |
| 1110     | 14     | 5  | 11 | 2  | 14 |
| 1111     | 15     | 10 | 5  | 15 | 9  |

DES : S-box 3

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 10 | 13 | 13 | 1  |
| 0001     | 1      | 0  | 7  | 6  | 10 |
| 0010     | 2      | 9  | 0  | 4  | 13 |
| 0011     | 3      | 14 | 9  | 9  | 0  |
| 0100     | 4      | 6  | 3  | 8  | 6  |
| 0101     | 5      | 3  | 4  | 15 | 9  |
| 0110     | 6      | 15 | 6  | 3  | 8  |
| 0111     | 7      | 5  | 10 | 0  | 7  |
| 1000     | 8      | 1  | 2  | 11 | 4  |
| 1001     | 9      | 13 | 8  | 1  | 15 |
| 1010     | 10     | 12 | 5  | 2  | 14 |
| 1011     | 11     | 7  | 14 | 12 | 3  |
| 1100     | 12     | 11 | 12 | 5  | 11 |
| 1101     | 13     | 4  | 11 | 10 | 5  |
| 1110     | 14     | 2  | 15 | 14 | 2  |
| 1111     | 15     | 8  | 1  | 7  | 12 |

DES : S-box 4

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 7  | 13 | 10 | 3  |
| 0001     | 1      | 13 | 8  | 6  | 15 |
| 0010     | 2      | 14 | 11 | 9  | 0  |
| 0011     | 3      | 3  | 5  | 0  | 6  |
| 0100     | 4      | 0  | 6  | 12 | 10 |
| 0101     | 5      | 6  | 15 | 11 | 1  |
| 0110     | 6      | 9  | 0  | 7  | 13 |
| 0111     | 7      | 10 | 3  | 13 | 8  |
| 1000     | 8      | 1  | 4  | 15 | 9  |
| 1001     | 9      | 2  | 7  | 1  | 4  |
| 1010     | 10     | 8  | 2  | 3  | 5  |
| 1011     | 11     | 5  | 12 | 14 | 11 |
| 1100     | 12     | 11 | 1  | 5  | 12 |
| 1101     | 13     | 12 | 10 | 2  | 7  |
| 1110     | 14     | 4  | 14 | 8  | 2  |
| 1111     | 15     | 15 | 9  | 4  | 14 |

DES : S-box 5

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 2  | 14 | 4  | 11 |
| 0001     | 1      | 12 | 11 | 2  | 8  |
| 0010     | 2      | 4  | 2  | 1  | 12 |
| 0011     | 3      | 1  | 12 | 11 | 7  |
| 0100     | 4      | 7  | 4  | 10 | 1  |
| 0101     | 5      | 10 | 7  | 13 | 14 |
| 0110     | 6      | 11 | 13 | 7  | 2  |
| 0111     | 7      | 6  | 1  | 8  | 13 |
| 1000     | 8      | 8  | 5  | 15 | 6  |
| 1001     | 9      | 5  | 0  | 9  | 15 |
| 1010     | 10     | 3  | 15 | 12 | 0  |
| 1011     | 11     | 15 | 10 | 5  | 9  |
| 1100     | 12     | 13 | 3  | 6  | 10 |
| 1101     | 13     | 0  | 9  | 3  | 4  |
| 1110     | 14     | 14 | 8  | 0  | 5  |
| 1111     | 15     | 9  | 6  | 14 | 3  |

## DES : S-box 6

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 12 | 10 | 9  | 4  |
| 0001     | 1      | 1  | 15 | 14 | 3  |
| 0010     | 2      | 10 | 4  | 15 | 2  |
| 0011     | 3      | 15 | 2  | 5  | 12 |
| 0100     | 4      | 9  | 7  | 2  | 9  |
| 0101     | 5      | 2  | 12 | 8  | 5  |
| 0110     | 6      | 6  | 9  | 12 | 15 |
| 0111     | 7      | 8  | 5  | 3  | 10 |
| 1000     | 8      | 0  | 6  | 7  | 11 |
| 1001     | 9      | 13 | 1  | 0  | 14 |
| 1010     | 10     | 3  | 13 | 4  | 1  |
| 1011     | 11     | 4  | 14 | 10 | 7  |
| 1100     | 12     | 14 | 0  | 1  | 6  |
| 1101     | 13     | 7  | 11 | 13 | 0  |
| 1110     | 14     | 5  | 3  | 11 | 8  |
| 1111     | 15     | 11 | 8  | 6  | 13 |

## DES : S-box 7

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 4  | 13 | 1  | 6  |
| 0001     | 1      | 11 | 0  | 4  | 11 |
| 0010     | 2      | 2  | 11 | 11 | 13 |
| 0011     | 3      | 14 | 7  | 13 | 8  |
| 0100     | 4      | 15 | 4  | 12 | 1  |
| 0101     | 5      | 0  | 9  | 3  | 4  |
| 0110     | 6      | 8  | 1  | 7  | 10 |
| 0111     | 7      | 13 | 10 | 14 | 7  |
| 1000     | 8      | 3  | 14 | 10 | 9  |
| 1001     | 9      | 12 | 3  | 15 | 5  |
| 1010     | 10     | 9  | 5  | 6  | 0  |
| 1011     | 11     | 7  | 12 | 8  | 15 |
| 1100     | 12     | 5  | 2  | 0  | 14 |
| 1101     | 13     | 10 | 11 | 5  | 2  |
| 1110     | 14     | 6  | 8  | 9  | 3  |
| 1111     | 15     | 1  | 6  | 2  | 12 |

DES : S-box 8

| binary   | d1d6=> | 00 | 01 | 10 | 11 |
|----------|--------|----|----|----|----|
| Vd2..d5V | Dec    | 0  | 1  | 2  | 3  |
| 0000     | 0      | 13 | 1  | 7  | 2  |
| 0001     | 1      | 2  | 15 | 11 | 1  |
| 0010     | 2      | 8  | 13 | 4  | 14 |
| 0011     | 3      | 4  | 8  | 1  | 7  |
| 0100     | 4      | 6  | 10 | 9  | 4  |
| 0101     | 5      | 15 | 3  | 12 | 10 |
| 0110     | 6      | 11 | 7  | 14 | 8  |
| 0111     | 7      | 1  | 4  | 2  | 13 |
| 1000     | 8      | 10 | 12 | 0  | 15 |
| 1001     | 9      | 7  | 5  | 6  | 12 |
| 1010     | 10     | 3  | 6  | 10 | 9  |
| 1011     | 11     | 14 | 11 | 13 | 0  |
| 1100     | 12     | 5  | 0  | 15 | 3  |
| 1101     | 13     | 0  | 14 | 3  | 5  |
| 1110     | 14     | 12 | 9  | 5  | 6  |
| 1111     | 15     | 7  | 2  | 8  | 11 |

Table Permutation (P)

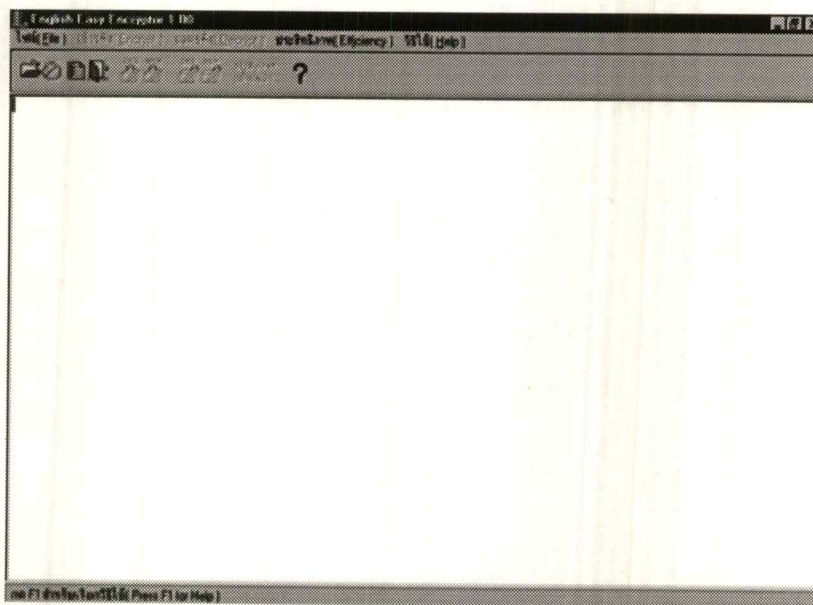
| บิตที่ | เปลี่ยนเป็นบิตที่ | บิตที่ | เปลี่ยนเป็นบิตที่ |
|--------|-------------------|--------|-------------------|
| 16     | 1                 | 2      | 17                |
| 7      | 2                 | 8      | 18                |
| 20     | 3                 | 24     | 19                |
|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

Table IP<sup>1</sup>

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

## ภาคผนวก ค คู่มือการใช้งานซอฟต์แวร์

เมื่อเริ่มเข้าสู่ซอฟต์แวร์นี้ จะปรากฏหน้าจอดังรูปที่ ค-1



รูปที่ ค-1 แสดงหน้าจอเริ่มแรกของการเข้าสู่โปรแกรม

หน้าจอจากภาพที่ ค-1 นี้ ส่วนบน คือ เมนูหลัก (Main Menu) ที่มีลักษณะเป็น Pull down Menu ตัวเมนู ประกอบไปด้วย

### ◆ ไฟล์ (File)

- เลือกไฟล์ (Select File) สำหรับเลือกไฟล์ที่จะทำการเข้ารหัสหรือถอดรหัส

- ลบรายการ (Clear list) ลบรายการที่ได้เลือก (Select file)เอาไว้

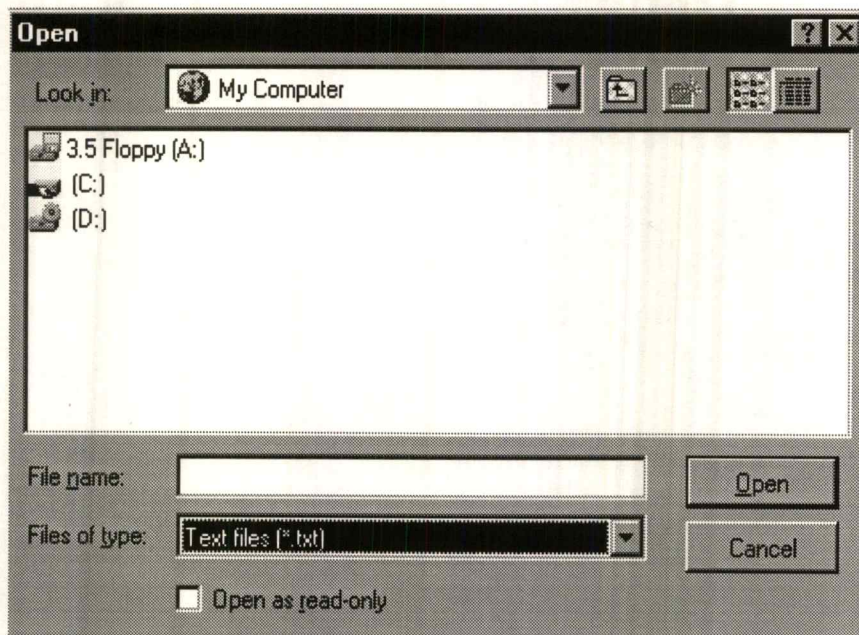
- ออกจากโปรแกรม (Exit) ทำการออกจากโปรแกรม English Easy Encryptor 1.00

### ◆ เข้ารหัส (Encrypt)

- DES ทำการเข้ารหัสด้วยใช้ algorithm แบบ DES

- RSA ทำการเข้ารหัสด้วยใช้ algorithm แบบ RSA

### ◆ ถอดรหัส (Decrypt)



รูปที่ ค-2 แสดงการหน้าจอที่ใช้เลือกไฟล์ที่จะทำการเข้ารหัสหรือถอดรหัส

โดยที่ผู้ใช้สามารถที่จะพิมพ์ชื่อไฟล์ที่ต้องการจะเลือก หรือพิมพ์ type ไฟล์โดยใช้ wildcard (\*) เพื่อเลือกให้ list เฉพาะไฟล์ที่มี type ที่ต้องการได้ ตัวอย่างเช่น ผู้ใช้สามารถพิมพ์ \*.txt เพื่อจะให้ list เฉพาะไฟล์ที่มี type เป็น .txt ได้

บาง type ไฟล์ที่ใช้บ่อย ๆ จะมีอยู่ใน list box อยู่แล้ว ผู้ใช้สามารถจะ click เพื่อเลือก type ที่ต้องการจะ list ได้เลย และถ้าต้องการที่จะ list ไฟล์ทั้งหมดใน directory นั้น ก็สามารถจะเลือก \*.\* ได้

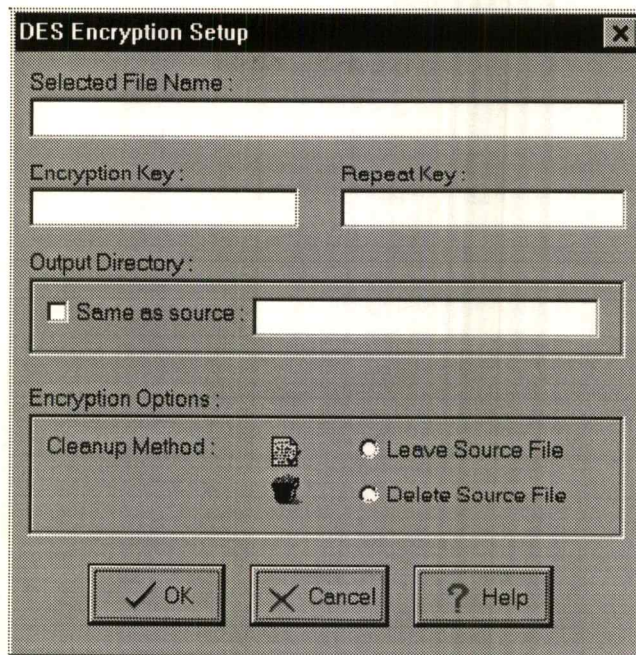
หมายเหตุ ไฟล์ที่จะทำการเข้ารหัส จะต้องเป็นภาษาอังกฤษเท่านั้น ไม่สามารถที่จะเข้ารหัสไฟล์ที่เป็นภาษาไทยได้

เมื่อเลือกไฟล์แล้ว ให้ผู้ใช้ทำการเลือกที่จะทำการเข้ารหัสหรือถอดรหัส

#### ◆ ถ้าต้องการเข้ารหัส

- การเข้ารหัสแบบ DES

เมื่อผู้ใช้เลือก DES จากเมนู Decrypt หรือ เลือก  จาก toolbar จะปรากฏหน้าจอ ดังรูป ค-3



รูปที่ ค-3 แสดงหน้าจอการเข้ารหัสแบบ DES

ส่วน Selected File Name

จะแสดงไฟล์ที่ผู้ใช้ได้เลือกไว้เพื่อทำการเข้ารหัส

ส่วน Encryption Key

ผู้ใช้งานจะต้องกำหนด key ที่ใช้ในการเข้ารหัส โดยที่

Key ที่ใช้ในการเข้ารหัสนี้ต้องมีมากกว่า 8 ตัวอักษร

ส่วน Repeat Encryption Key

ผู้ใช้งานจะต้องพิมพ์ key ที่ใช้ในการเข้ารหัสซ้ำอีกครั้ง

ซึ่งเป็น key เดียวกับ key ที่พิมพ์ลงใน ส่วน Encrypt Key เพื่อป้องกันการพิมพ์ Key ผิดพลาด

ส่วน Output Directory

ผู้ใช้งานจะต้องกำหนดไดเรกทอรีของไฟล์ที่เป็น output ถ้าเป็นไดเรกทอรีเดิมที่เก็บไฟล์ที่จะเข้ารหัสไว้ ก็ให้ทำการเลือกcheckboxล๊อกที่ same as source ถ้าหากต้องการที่จะเปลี่ยนแปลง ให้ทำการพิมพ์เข้าไป

ส่วน Encryption Option

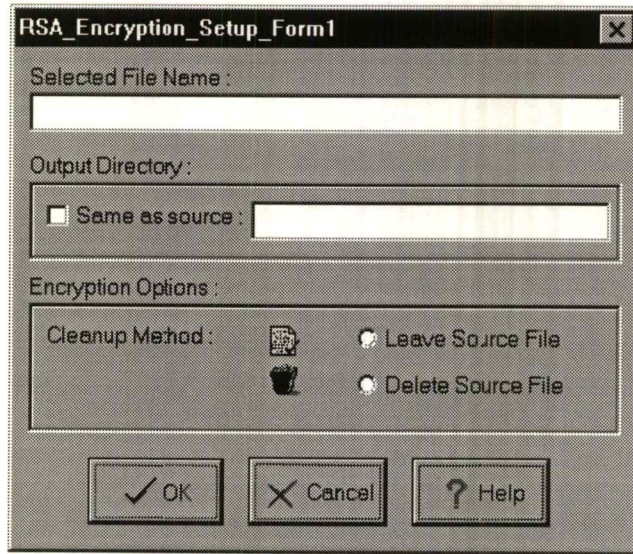
จะประกอบด้วย radio button อยู่ 2 ปุ่ม คือ

- Leave Source File คือ เมื่อทำการ Encrypt แล้ว ไม่ต้องทำการลบไฟล์ที่เป็น Source
- Delete Source File คือ เมื่อทำการ Encrypt แล้ว ให้ทำการลบไฟล์ที่เป็น Source ทิ้งไปด้วย

เมื่อทำการกดปุ่ม Ok โปรแกรมก็จะเริ่มทำการเข้ารหัส

เมื่อผู้ใช้เลือก RSA จาก Encrypt เมนู หรือเลือก  จาก toolbar จะปรากฏ

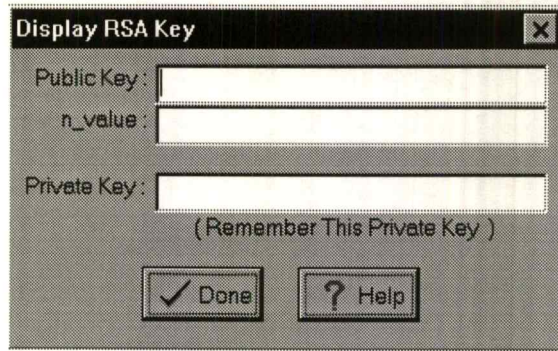
หน้าจอจดังรูป ค-4



รูปที่ ค-4 แสดงหน้าจอการเข้ารหัสแบบ RSA

|                         |                                                                                                                                                                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ส่วน Selected File name | จะแสดงไฟล์ที่ผู้ใช้ได้เลือกไว้เพื่อทำการเข้ารหัส                                                                                                                                                                                                                                 |
| ส่วน Output Directory   | ผู้ใช้จะต้องกำหนดไดเรกทอรีของไฟล์ที่เป็น output ถ้าเป็นไดเรกทอรีเดิมที่เก็บไฟล์ที่จะเข้ารหัสไว้ ก็ให้ทำการเลือกเช็คบลิอกที่ same as source ถ้าหากต้องการที่จะเปลี่ยนแปลง ให้ทำการพิมพ์เข้าไป                                                                                     |
| ส่วน Encryption Option  | จะประกอบด้วย radio botton อยู่ 2 ปุ่ม คือ <ul style="list-style-type: none"><li>● Leave Source File คือ เมื่อทำการ Encrypt แล้ว ไม่ต้องทำการลบไฟล์ที่เป็น Source</li><li>● Delete Source File File คือ เมื่อทำการ Encrypt แล้ว ให้ทำการลบไฟล์ที่เป็น Source ทิ้งไปด้วย</li></ul> |

เมื่อทำการกดปุ่ม Ok โปรแกรมก็จะเริ่มทำการเข้ารหัส เมื่อทำการเข้ารหัสเสร็จ จะปรากฏหน้าจอจดังรูป ค-5



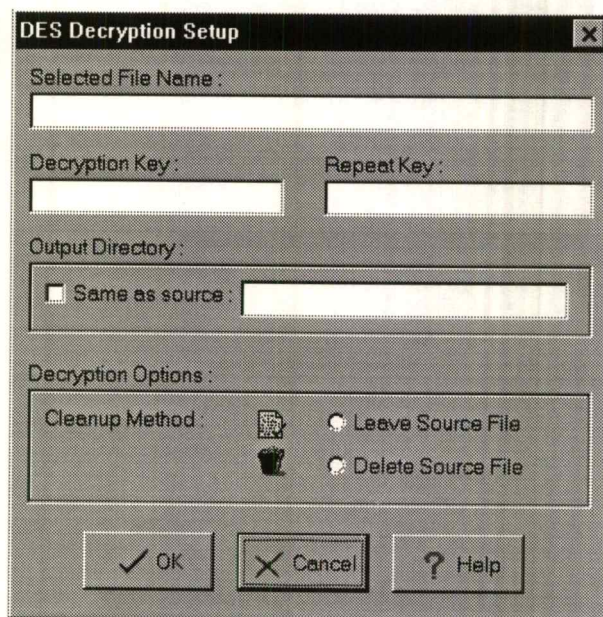
รูปที่ ค-5 แสดงหน้าจอการแสดง Public Key ,n และ Private Key แก่ผู้ใช้

โปรแกรมจะทำการแจ้ง Public key ที่ใช้ในการเข้ารหัสและ ค่า n เพื่อให้ผู้ใช้นำไปเปิดเผยให้ผู้อื่นต่อไป เมื่อผู้อื่นต้องการที่จะส่งไฟล์มาให้ผู้ใช้ ก็ให้ทำการเข้ารหัสด้วย Public key นี้ ผู้ใช้จะต้องจำ key ที่เป็น Private key ไว้ด้วย เพื่อที่จะทำการถอดรหัสไฟล์ที่ผู้อื่นส่งมาให้

#### ◆ ถ้าต้องการถอดรหัส

- การถอดรหัสแบบ DES

เมื่อผู้ใช้เลือก DES จาก Decrypt เมนู หรือเลือก  จาก toolbar จะปรากฏหน้าจอ ดังรูป ค-6



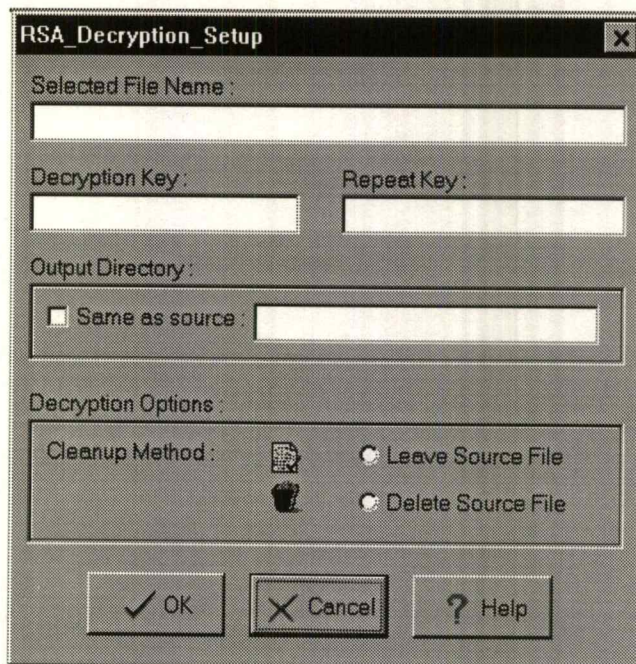
## รูปที่ ค-6 แสดงหน้าจอการถอดรหัสแบบ DES

|                                       |                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ส่วน Selected File Name               | จะแสดงไฟล์ที่ผู้ใช้ได้เลือกไว้เพื่อทำการถอดรหัส                                                                                                                                                                                                                                                                                   |
| ส่วน Decryption Key                   | ผู้ใช้จะต้องกรป<br>Key ที่ใช้ในการถอดรหัสนี้ต้องมีมากกว่า 8 ตัวอักษร<br>และเป็น key เดียวกับที่ใช้ในการเข้ารหัส                                                                                                                                                                                                                   |
| ส่วน Repeat Decryption Key            | ผู้ใช้จะต้องพิมพ์ key ที่ใช้ในการถอดรหัสซ้ำอีกครั้ง<br>ซึ่งเป็น key เดียวกับ key ที่พิมพ์ลงใน ส่วน Decrypt<br>Key เพื่อป้องกันการพิมพ์ Key ผิดพลาด                                                                                                                                                                                |
| ส่วน Output Directory<br>เป็น<br>เช็ค | ผู้ใช้จะต้องกำหนดไดเรกทอรีของไฟล์ที่เป็น output ถ้า<br>ไดเรกทอรีเดิมที่เก็บไฟล์ที่จะเข้ารหัสไว้ ก็ให้ทำการเลือก<br>บล็อกที่ same as source ถ้าหากต้องการที่จะเปลี่ยน<br>แปลง ให้ทำการพิมพ์เข้าไป                                                                                                                                  |
| ส่วน Decryption Option                | จะประกอบด้วย radio button อยู่ 2 ปุ่ม คือ <ul style="list-style-type: none"> <li>• Leave Source File คือ เมื่อทำการ Decrypt แล้ว ไม่ต้องทำการลบไฟล์ที่เป็น Source</li> <li>• Delete Source File คือ เมื่อทำการ Decrypt แล้ว ให้ทำการลบไฟล์ที่เป็น Source ทิ้งไปด้วย</li> </ul> เมื่อกดปุ่ม Ok โปรแกรมก็จะเริ่มทำการถอดรหัสแบบ DES |

## - การถอดรหัสแบบ RSA

เมื่อผู้ใช้เลือก RSA จาก Decrypt เมนู หรือเลือก  จาก toolbar จะปรากฏหน้าจอ

ดังรูป ค-7



รูปที่ ค-7 แสดงหน้าจอการถอดรหัสแบบ RSA

|                            |                                                                                                                                                                                                                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ส่วน Selected File Name    | จะแสดงไฟล์ที่ผู้ใช้ได้เลือกไว้เพื่อทำการถอดรหัส                                                                                                                                                                                                                                     |
| ส่วน Decryption Key        | ผู้ใช้จะต้องกำหนด key ที่ใช้ในการถอดรหัส โดยที่ Key ที่ใช้ในการถอดรหัสนี้ต้องมีมากกว่า 8 ตัวอักษร และเป็น key เดียวกับที่ใช้ในการเข้ารหัส                                                                                                                                           |
| ส่วน Repeat Decryption Key | ผู้ใช้จะต้องพิมพ์ key ที่ใช้ในการถอดรหัสซ้ำอีกครั้ง ซึ่งเป็น key เดียวกับ key ที่พิมพ์ลงใน ส่วน Decrypt Key เพื่อป้องกันการพิมพ์ Key ผิดพลาด                                                                                                                                        |
| ส่วน Output Directory      | ผู้ใช้จะต้องกำหนดไดเรกทอรีของไฟล์ที่เป็น output ถ้าเป็นไดเรกทอรีเดิมที่เก็บไฟล์ที่จะเข้ารหัสไว้ ก็ให้ทำการเลือกเช็คบิล็อกที่ same as source ถ้าหากต้องการที่จะเปลี่ยนแปลง ให้ทำการพิมพ์เข้าไป                                                                                       |
| ส่วน Decryption Option     | จะประกอบด้วย radio button อยู่ 2 ปุ่ม คือ <ul style="list-style-type: none"> <li>• Leave Source File คือ เมื่อทำการ Decrypt แล้ว ไม่ต้องทำการลบไฟล์ที่เป็น Source</li> <li>• Delete Source File File คือ เมื่อทำการ Decrypt แล้ว ให้ทำการลบไฟล์ที่เป็น Source ทิ้งไปด้วย</li> </ul> |

เมื่อกดปุ่ม Ok โปรแกรมก็จะเริ่มทำการถอดรหัสแบบ RSA

หมายเหตุ หากทำการเข้ารหัสโดยใช้ Algorithm แบบ DES ก็ต้องทำการถอดรหัสโดยใช้ Algorithm แบบ DES ด้วย หากทำการเข้ารหัสโดยใช้ Algorithm แบบ RSA ก็ต้องทำการถอดรหัสโดยใช้ Algorithm แบบ RSA ด้วย

เมื่อต้องการที่จะเปรียบเทียบประสิทธิภาพของการเข้ารหัสโดยใช้ algorithm แบบ DES และแบบ RSA ให้ทำการ เลือกที่ ประสิทธิภาพ (Efficiency) ซึ่งจะให้เลือกอยู่ 2 วิธี คือ

- ถ้าต้องการที่จะเลือกเปรียบเทียบเวลาต่อขนาดไฟล์ โดยแกน x จะแทนขนาดไฟล์ ส่วนแกน y จะแทนระยะเวลาที่ใช้ในการเข้ารหัส ให้ทำการเลือกที่ ขนาด : เวลา (Size : time) จาก Efficiency เมนู หรือเลือกที่ปุ่ม  จาก toolbar จากนั้น โปรแกรมก็จะทำการพลอตกราฟ
- ถ้าต้องการที่จะเลือกที่จะเปรียบเทียบการกระจายของตัวอักษร โดยแกน x จะแทนตัวอักษรตามลำดับรหัสแอสกี ส่วนแกน y จะแทนความถี่ในการใช้ตัวอักษร (จำนวนการของซ้ำตัวอักษรที่ปรากฏในไฟล์ที่ถูกเข้ารหัสแล้ว) ให้ทำการเลือกที่ อักษร : จำนวน (Alphabetic : Size) จาก Efficiency เมนู หรือเลือกที่ปุ่ม  จาก toolbar จากนั้น โปรแกรมก็จะทำการพลอตกราฟ

หากต้องการออกจากโปรแกรม ให้เลือก Exit จากเมนู File หรือเลือกที่ปุ่ม  จาก toolbar

## ปัญหาที่เกิดจากการเข้ารหัสแบบ DES

การเข้ารหัสแบบ DES สามารถเข้ารหัสได้ผลออกมาถูกต้อง แต่ในการถอดรหัสให้ผลลัพธ์ออกมาไม่ถูกต้อง เนื่องจากเมื่อเริ่มทำการถอดรหัส เราจะนำ Ciphertext ( ข้อความเข้ารหัสแล้ว) ไป permute เทียบกับเมทริกซ์ IP ซึ่งเป็นอินเวอร์สกับเมทริกซ์ FP แต่ผลลัพธ์ที่ได้ไม่ตรงกับข้อความที่ผ่านการเข้ารหัสในรอบสุดท้าย ( รอบที่ 16 )

## บรรณานุกรม

1. ยืน ภู่วรวรรณ , “ระบบรหัสลับ” , Internet magazine , สำนักพิมพ์ ซีเอ็ด , บริษัทซีเอ็ดยูเคชั่น จำกัด(มหาชน) , พ.ศ. 2540 : หน้า 53-58
  2. ค.ร. กิติ ไพฑูรย์วัฒนกิจ , ยุทธนา สวนสุข , “การสร้างการเข้ารหัสแบบอาร์เอสเอ” , คอมพิวเตอร์ , สำนักพิมพ์ศูนย์การพิมพ์พลชัย , บริษัทเอ.อาร์.อินฟอร์เมชันแอนด์พับลิเคชั่น จำกัด , พ.ศ. 2535 : หน้า 28-37
  3. Douglas R.Stinson , Cryptography and Practice , CRC Press Inc , 1955
  4. Phi Sutherland , Cryptography , John Wiley & Sens , Inc ,1996
-