บทที่ 3

ขั้นตอนการดำเนินงาน

หลังจากการศึกษาทฤษฎีและหลักการเกี่ยวกับการเข้ารหัสจากบทที่ 2 แล้ว สามารถนำมาประยุกต์ใช้ได้จริง ซึ่งมีขั้นตอนการดำเนินงานดังนี้ ส่วนที่1 คือการศึกษาทฤษฎีและหลักการของอัลกอริทึมในการเข้ารหัส ส่วนที่ 2 คือการศึกษาและพัฒนาโปรแกรมเพื่อทำให้มีประสิทธิภาพในการเข้ารหัสมากยิ่งขึ้น ส่วนที่ 3 คือการนำโปรแกรมการเข้ารหัสที่ได้มาทดสอบการเข้ารหัสกับไฟล์ข้อมูลหลายๆ ประเภท และหลายๆ อัลกอริทึม เพื่อนำมาเปรียบเทียบประสิทธิภาพ

3.1 การศึกษาทฤษฎีและหลักการของอัลกอริทึมในการเข้ารหัส

การเข้ารหัสลับข้อมูลมีอยู่ 3 ประเภทหลักๆ ด้วยกัน ซึ่งในที่นี้จะนำเสนอการเข้ารหัสแบบสมมาตรซึ่งมีอัลกอริทึม 6 แบบด้วยกัน ได้แก่ AES, Serpent, RC2, CAST-6 (CAST-256), Blowfish และ Twofish ซึ่งหลังจากการศึกษาแล้วจะพบได้ดังนี้

อัลกอริทึมแบบ AES ใช้บล็อกข้อมูลขนาด 128 บิต 196 บิต และ 256 บิต โดยสามารถใช้คีย์ได้ยาวถึง 128 บิต 196 บิต และ 256 บิต โดยจะใช้ Round Function ที่สามารถเลือกได้ว่าจะทำ 10,12 หรือ 14 ครั้ง มีการทำงานอยู่ 4 ส่วนย่อย คือ Byte Sub ก็คือการใช้ S-Boxes ในการสลับข้อมูลระหว่าง 2 บล็อก ShiftRow คือการสลับข้อมูลระหว่างแถว Mix Column คือการ Shift ข้อมูลในแต่ละ Column และสุดท้ายคือ Key Addition คือการนำมาบวกกับคีย์

อัลกอริทึมแบบ Serpent มีแนวคิดมาจากอัลกอริทึมแบบ AES โดยมีขนาดของบล็อกเป็น 128 บิต และขนาดของคีย์เป็น 128, 192 และ 256 บิต รวมทั้งมีการเข้ารหัสทั้งหมด 32 รอบ ในการเข้ารหัสจะทำการแบ่งข้อมูลเป็น 4 ส่วน ส่วนละ 32 บิต ซึ่งจะทำใน S-boxes ไปพร้อมๆ กัน จะทำงานทั้งหมด 32 รอบด้วยกัน

อัลกอริทึมแบบ RC2 ใช้ขนาดของข้อมูล 64 บิต และมีขนาดของคีย์ตั้งแต่ 8 บิต ถึง 128 บิต ส่วนใหญ่จะใช้ที่ 64 บิต มีการทำงาน 18 รอบด้วยกัน โดยทำการ MXING 16 รอบ และทำการ MASHING 2 รอบด้วยกัน

อัลกอริทึมแบบ CAST-6 เป็นการเข้ารหัสแบบบล็อกข้อมูลที่พัฒนามาจากอัลกอริทึมแบบ CAST-128 รวมทั้งการใช้ S-Boxes แต่ใช้ขนาดของบล็อกข้อมูลถึง 128 บิต และสามารถใช้คีย์ได้ทั้ง 128 บิต 192 บิต และ 256 บิต เข้ารหัสทั้งหมด 48 รอบ

อัลกอริทึมแบบ Blowfish การใช้ S-Boxes และ XOR และคีย์ที่มีการเปลี่ยนค่าความยาวได้ตั้งแต่ 32 บิต ถึง 448 บิต แต่ในทางปฏิบัติมักใช้กันที่ 128 บิต มีการนำเอาอัลกอริทึมแบบ Blowfish

เองมาใช้ในการสร้างคีย์ย่อยและ S-Boxes ด้วย โดยจะต้องมีการวนทั้งหมด 521 รอบในการทำงานเพื่อสร้างคีย์ย่อยและ S-Boxes และมีการวนรอบเข้ารหัสทั้งหมด 16 รอบด้วยกัน

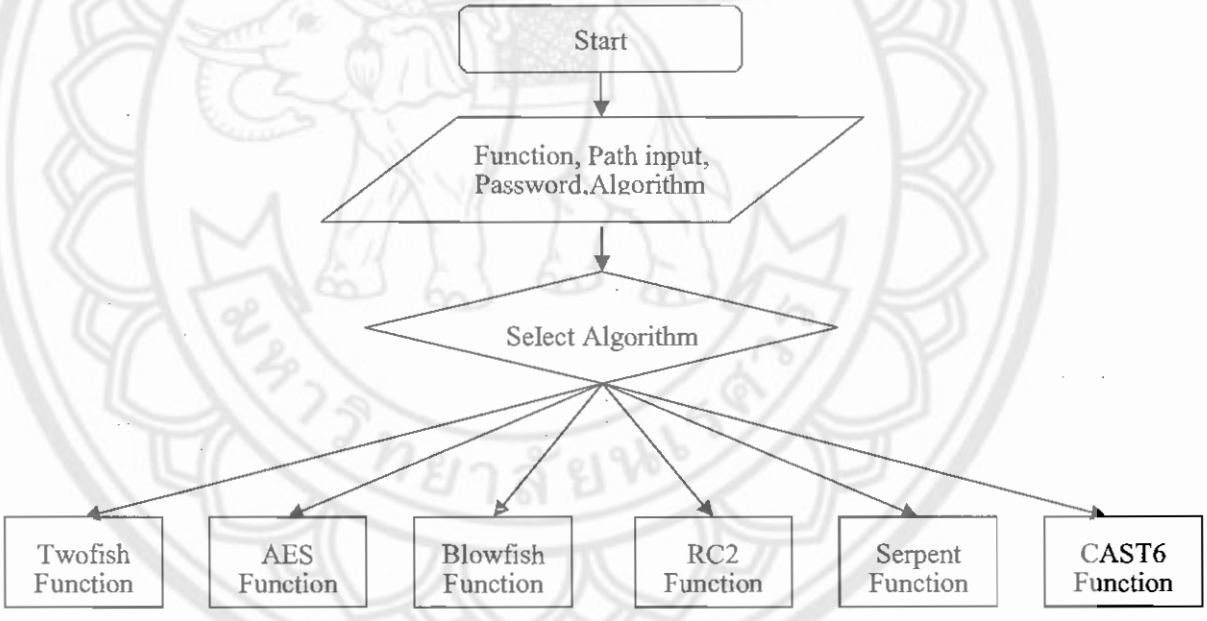
อัลกอริทึมแบบ Twofish เป็นแบบบล็อกข้อมูล บล็อกละ 128 บิต ซึ่งใน 128 บิตนี้จะถูกแบ่งเป็น 4 ส่วนๆ ละ 32 บิต ซึ่งนำแต่ละส่วนมาทำการ XOR กับกุญแจ 4 ตัว แล้วจึงนำมาเข้าฟังก์ชันต่างที่ใช้ในอัลกอริทึมนี้ แล้วทำซ้ำให้ครบ 16 รอบ ซึ่งคีย์ที่สามารถใช้ได้ 3 ขนาด ได้แก่ 128 หรือ 192 หรือ 256 บิต

3.2 การศึกษาและพัฒนาโปรแกรม

ในขั้นตอนการศึกษาและพัฒนาโปรแกรมนั้น ได้ศึกษาเกี่ยวกับวิธีการทำงานของ Library เพื่อนำมาพัฒนาโปรแกรม ซึ่ง Library ที่ใช้คือ Bouncycastle ซึ่งจะมีวิธีการทำงาน ดังนี้

ขั้นตอนการทำงานของโปรแกรม

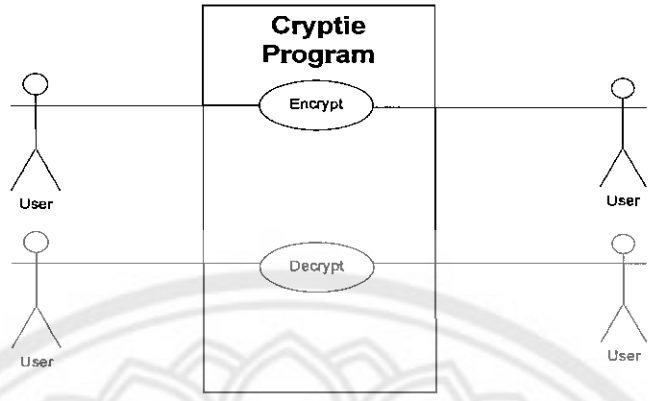
Flow chart



รูปที่ 3.1 Flow chart โปรแกรม Cryptie

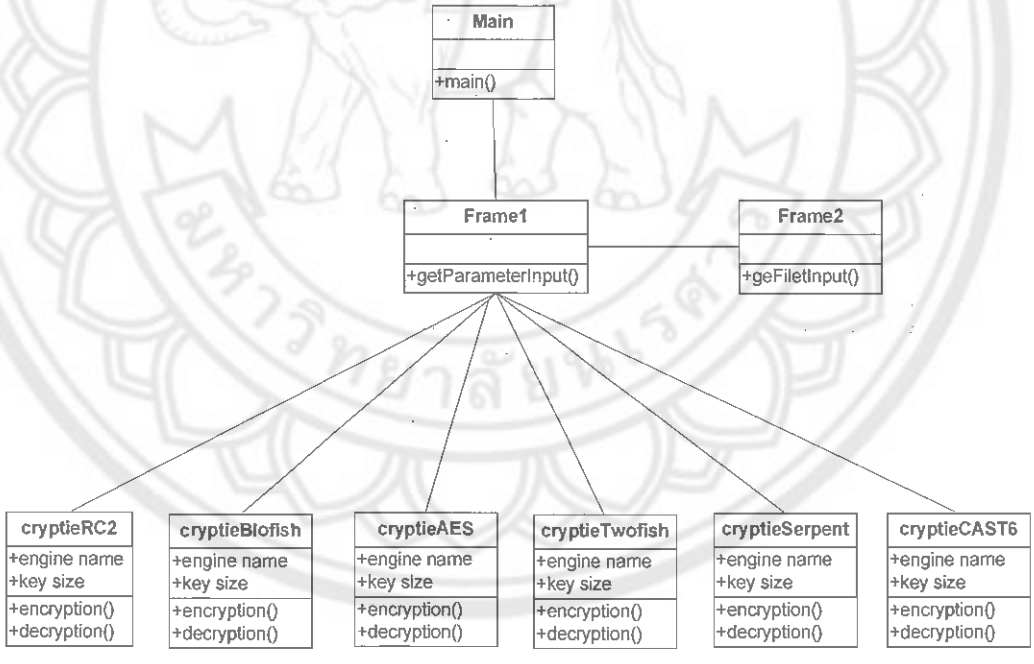
เมื่อทำการเลือก Algorithm, Function (encrypt/decrypt), Input File ที่ต้องการและ password โปรแกรมก็จะทำการส่งค่า ที่ Input ไปยัง function แต่ในละ Algorithm เพื่อทำการเรียกใช้ Libraly เพื่อทำการ Encrypt หรือ Decrypt

User case Diagram



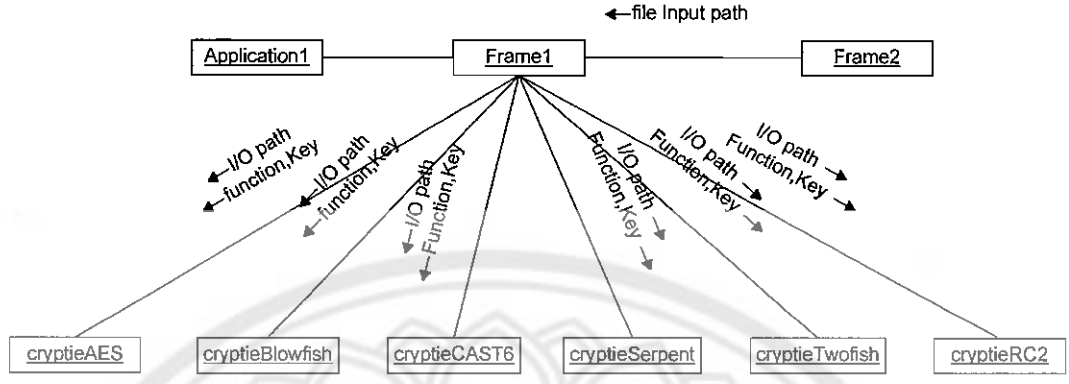
รูปที่ 3.2 User Case Diagram

Class Diagram



รูปที่ 3.3 Class Diagram

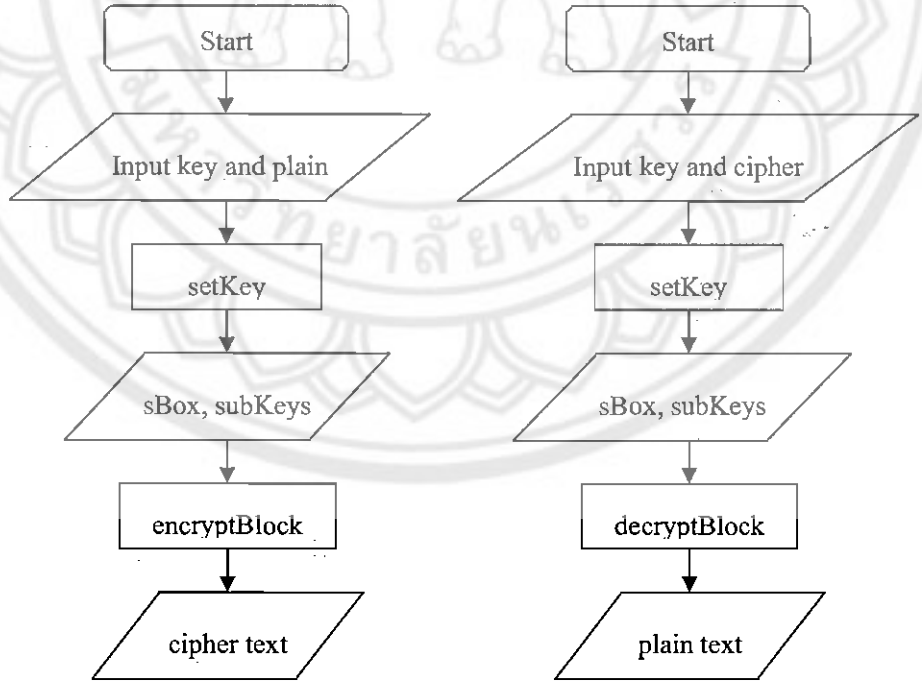
Collaboration Diagram



รูปที่ 3.4 Collaboration Diagram

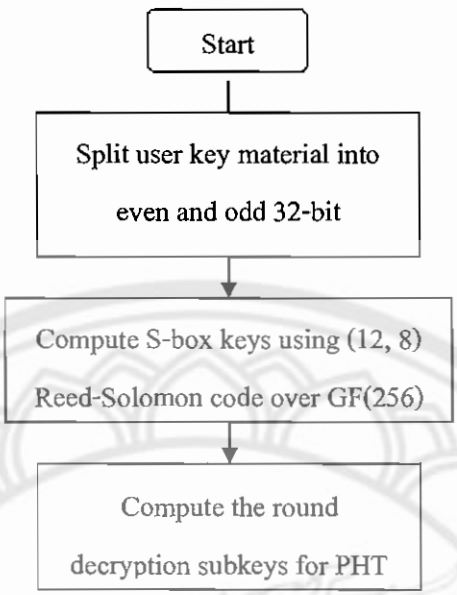
วิธีการทำงานของ Library

เนื่องจากโปรแกรมได้มีการเรียกใช้ Library หลายตัวดังนั้นจะยกตัวอย่างการทำงานของ library twofish ดังนี้



รูปที่ 3.5 ขั้นตอนการ Encrypt/Decrypt ของ Library Bouncycastle

SetKey Function



รูปที่ 3.6 ขั้นตอนการสร้าง key ของ Library Bouncycastle

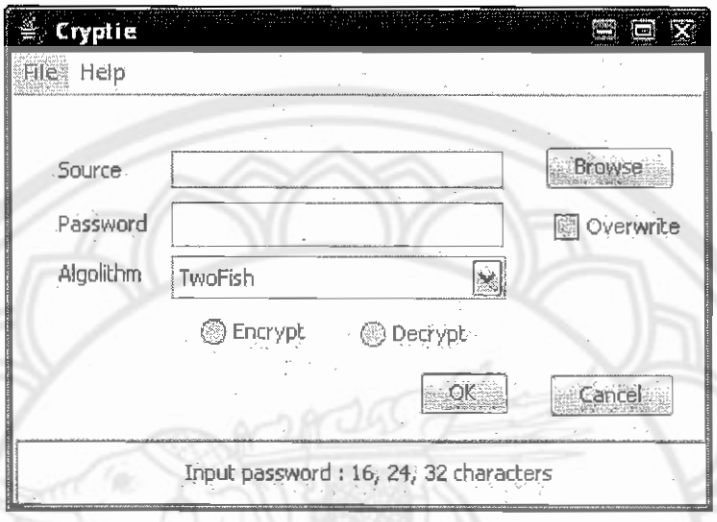
ขั้นแรก setKey function จะเป็นนำค่า key material ที่ผู้ใช้ได้ Input เข้ามามาแบ่งค่าเป็น odd key และ even key และคำนวน S-box key โดยใช้ (12, 8) Reed-Solomon code over GF(256) จากนั้นก็จะคำนวนหา subkeys สำหรับใช้ในขั้นตอนการ PHT แต่ในการคำนวณหา session key นั้นจะใช้วิธีการ map กับ table ที่มีการคำนวนค่าไว้แล้ว เพื่อความเร็วในการเข้ารหัส เมื่อได้ key ครบแล้ว ก็จะทำการเข้ารหัส โดยใช้ encryptBlock function ส่วนการถอดรหัสจะมีวิธีการทำงานแบบเดี่ยวกันแต่เป็นการ ย้อนกลับ key

3.3 การทดสอบประสิทธิภาพในการเข้ารหัสแต่ละอัลกอริทึม

ในการทดสอบการเข้ารหัสของแต่ละอัลกอริทึมจะมีหลักการดังนี้ ทำการแบ่งข้อมูลออกเป็น 3 ประเภทหลักๆ ด้วยกันได้แก่ ข้อมูลประเภท Plan text ซึ่งได้แก่ข้อมูลประเภท text file หรือข้อมูลที่มีนามสกุล .txt ข้อมูลประเภท Semi binary file ได้แก่ ข้อมูลประเภท Word file, Excel เป็นต้น ข้อมูลประเภท pure binary file ได้แก่ ข้อมูลรูปภาพ หรือข้อมูลที่ถูกลดขนาดเอาไว้เช่น ข้อมูลที่มีนามสกุล .bmp, .rar, .zip เป็นต้น โดยการทดสอบจะนำผลที่ได้มาพิจารณาประสิทธิภาพของการเข้ารหัสลับข้อมูล ได้แก่ ประเภทของข้อมูล ขนาดของข้อมูล ระยะเวลาในการทดสอบ รวมทั้งความแตกต่างของข้อมูลที่ได้

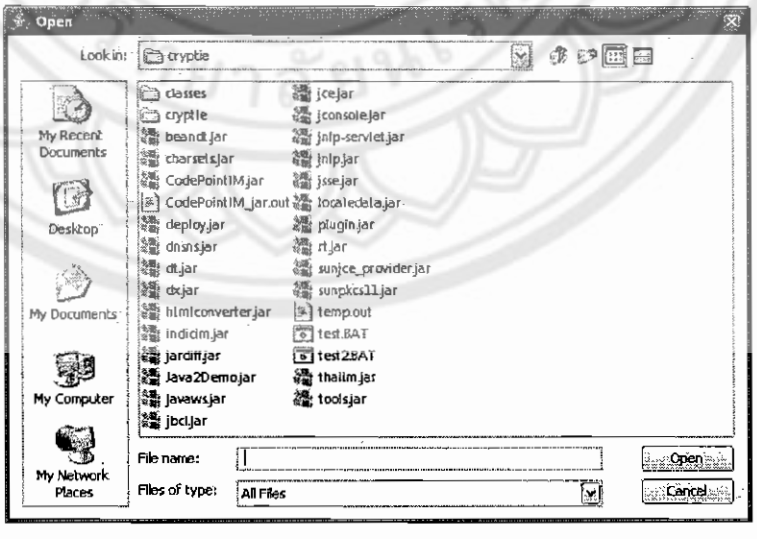
ขั้นตอนการทดสอบการทำงานของโปรแกรม

1. บันทึกคุณสมบัติของเครื่องคอมพิวเตอร์ที่ใช้การทดสอบ
2. ขั้นตอนในการเข้ารหัสข้อมูลมีดังนี้
   - เปิดโปรแกรม Cryptie จะได้ดังรูปที่ 3.7



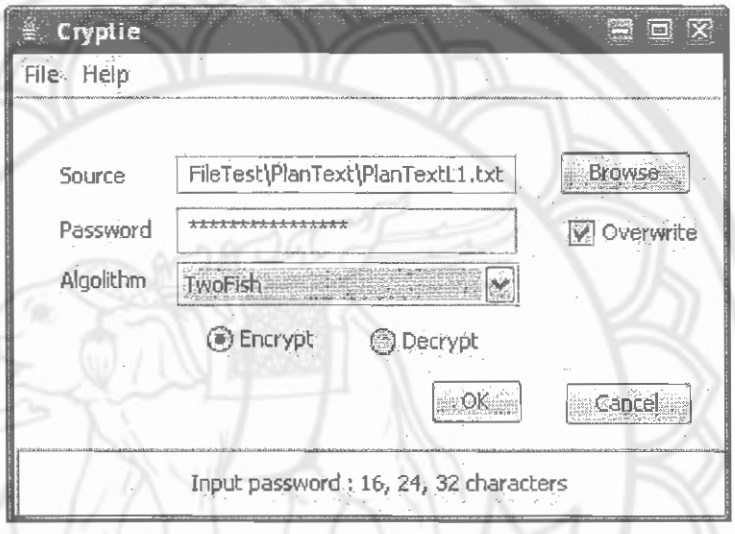
รูปที่ 3.7 หน้าต่างโปรแกรม Cryptie

- แล้วคลิก Browse เพื่อหาตำแหน่งข้อมูล จากนั้นจะมีหน้าต่างขึ้นมา เพื่อทำการเลือกตำแหน่งข้อมูลที่ต้องการ จะได้ดังรูปที่ 3.8



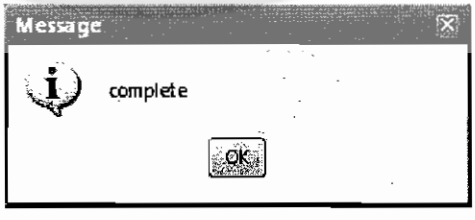
รูปที่ 3.8 หน้าต่างเพื่อให้หาตำแหน่งของข้อมูลที่ต้องการ

- เมื่อเลือกข้อมูลได้แล้วให้ใส่ Password ในการใส่ Password จะต้องใส่ให้ถูกต้องตามขนาดคีย์ของแต่ละอัลกอริทึม ซึ่งสังเกตได้จากส่วนของ Input password ด้านล่างของหน้าต่างซึ่งจะบ่งบอกถึงจำนวนคีย์ที่ใส่ได้
- เลือกอัลกอริทึมที่ใช้ในการเข้ารหัส
- เลือก Solution เป็นแบบ Encryption ที่ ⦿ Encrypt หากต้องการที่จะเข้ารหัสโดยให้ข้อมูลที่ได้ทับไฟล์ต้นฉบับให้เลือกที่ ☑ Overwrite แสดงดังรูปที่ 3.9



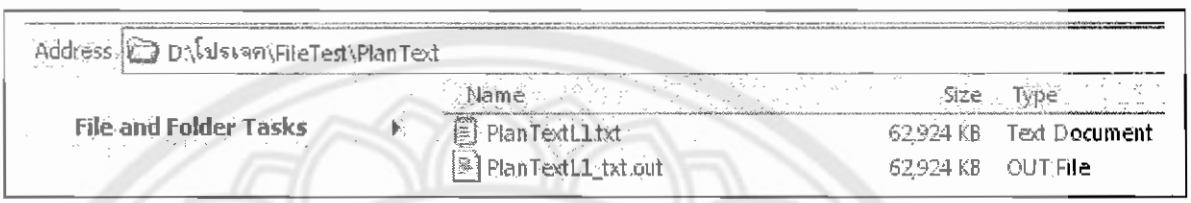
รูปที่ 3.9 การใส่ Password การเลือก Solution ต่างๆ ในการเข้ารหัส

- แล้วคลิกที่ OK เพื่อเริ่มการเข้ารหัสข้อมูล
- จับเวลาที่ใช้ในการเข้ารหัส เมื่อการเข้ารหัสเสร็จแล้วจะมี Dialog Box ขึ้นแสดงดังรูปที่ 3.10



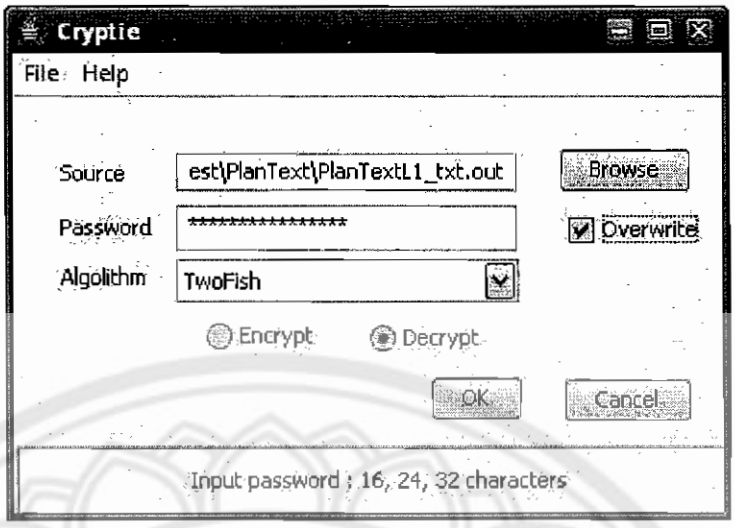
รูปที่ 3.10 ข้อความเมื่อการเข้ารหัสเสร็จแล้ว

- คลิก OK เพื่อปิดหน้าต่าง แล้วบันทึกระยะเวลาที่ใช้ในการเข้ารหัส
- เมื่อได้ไฟล์ที่เข้ารหัสไว้แล้ว ซึ่งจะอยู่ที่เดียวกับไฟล์ต้นฉบับ แต่ชื่อของไฟล์จะเปลี่ยนเป็นชื่อไฟล์ต้นฉบับ ตามด้วยเครื่องหมาย _ (under scroll) ตามด้วยนามสกุลของไฟล์เดิม และมีนามสกุลเปลี่ยนใหม่เป็น .out แสดงดังรูปที่ 3.11

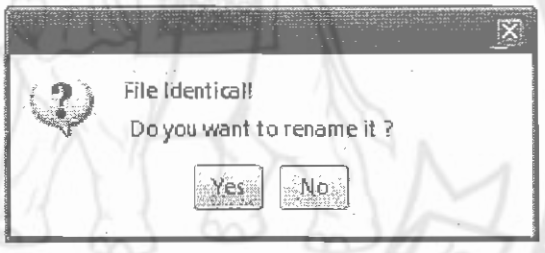


รูปที่ 3.11 ตำแหน่งของข้อมูลหลังการเข้ารหัสแล้ว

3. ขั้นตอนในการถอดรหัสมีดังนี้
   - เปิดโปรแกรม Cryptie แล้วคลิก Browse เพื่อหาตำแหน่งข้อมูล จากนั้นจะมีหน้าต่างขึ้นมาเพื่อทำการเลือกตำแหน่งข้อมูลที่ต้องการซึ่งเป็นไฟล์นามสกุล .out
   - เมื่อเลือกข้อมูลได้แล้วให้ใส่ Password ในการใส่ Password จะต้องใส่ให้ถูกต้องตามขนาดคีย์ของแต่ละอัลกอริทึม ซึ่งสังเกตได้จากส่วนของ Input password ด้านล่างของหน้าต่างซึ่งจะบ่งบอกถึงจำนวนคีย์ที่ใส่ได้
   - เลือกอัลกอริทึมที่ใช้ในการถอดรหัส
   - เลือก Solution เป็นแบบ Decryption ที่ Decrypt หากต้องการที่จะถอดรหัสโดยให้ข้อมูลที่ได้ทับไฟล์ต้นฉบับให้เลือกที่ Overwrite แสดงดังรูปที่ 3.12
   - แล้วคลิกที่ OK เพื่อเริ่มการถอดรหัสข้อมูล
   - เนื่องจากตำแหน่งที่ใช้เก็บไฟล์หลังจากการถอดรหัสจะอยู่ที่ตำแหน่งเดียวกับไฟล์ต้นฉบับ และในการถอดรหัสจะใช้ชื่อไฟล์เดิมเหมือนต้นฉบับ ดังนั้นเมื่อคลิก OK แล้วจึงมีข้อความดังรูปที่ 3.13 เพื่อบอกว่ามีไฟล์นี้อยู่แล้ว ต้องการเปลี่ยนชื่อหรือไม่



รูปที่ 3.12 การใส่ Password การเลือก Solution ต่างๆ ในการถอดรหัส



รูปที่ 3.13 ข้อความแสดงว่าในตำแหน่งนั้นมีไฟล์ชื่อนี้อยู่แล้ว ต้องการเปลี่ยนชื่อหรือไม่

- คลิก Yes หากต้องการเปลี่ยนชื่อ โดยจะมีข้อความขึ้นมาเพื่อให้ใส่ชื่อไฟล์ใหม่ เมื่อใส่ชื่อไฟล์ที่ต้องการ แล้วคลิกที่ OK
- คลิก No หากไม่ต้องการเปลี่ยนชื่อ แล้วไฟล์หลังการถอดรหัสจะแทนที่ข้อมูลในไฟล์เดิม
- จับเวลาที่ใช้ในการถอดรหัส เมื่อการถอดรหัสเสร็จแล้วจะมี Dialog Box แสดงดังรูปที่ 3.10

คลิก OK เพื่อปิดหน้าต่าง แล้วบันทึกระยะเวลาที่ใช้ในการถอดรหัส

4. ขั้นตอนการเปรียบเทียบข้อมูล

   - เปรียบเทียบขนาดของไฟล์ โดยดูจาก Properties ของแต่ละไฟล์แล้วบันทึกผลการทดลอง
   - เปรียบเทียบข้อความก่อนการเข้ารหัส และหลังการถอดรหัสโดยใช้โปรแกรม Compare It! 3.86 ในการเปรียบเทียบ เมื่อเปิดโปรแกรมแล้วเปิดไฟล์ 2 ไฟล์ แล้วทำการเปรียบเทียบ การทำงานของโปรแกรม Compare It! 3.86 เป็นการเปรียบเทียบข้อความของทั้งสองไฟล์ทีละตัว หากมีข้อมูลไม่ตรงกันก็จะแสดงส่วนที่ต่างกันออกมา หรือมีข้อมูลเหมือนกันก็จะแสดงว่าไฟล์ทั้งสองเหมือนกันออกมา แล้วบันทึกผลการทดลอง