การเข้ารหัสลับของสัญญาณภาพ โดยใช้รหัสลับอลวนแบบไม่ต่อเนื่อง

Image Encryption using Discrete Chaotic Signals

โดย

นายชานนท์ จันทร์วงศ์

นายธันยะวัฒน์ สวงโท

นายนิตินัย สามงามยา

นายอภินันท์ อยู่สมบูรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเข้ารหัสลับของสัญญาณภาพโดยใช้รหัสลับอลวนแบบไม่ต่อเนื่อง

Image Encryption using Discrete Signals

ผู้จัดทำ

1. นายชานนท์ จันทร์วงศ์ รหัสนักศึกษา 48010205
2. นายธันยะวัฒน์ สวงโท รหัสนักศึกษา 48010379
3. นายนิตินัย สามงามยา รหัสนักศึกษา 48010455
4. นายอภินันท์ อยู่สมบูรณ์ รหัสนักศึกษา 48012041

อาจารย์ที่ปรึกษา

(ผศ.ดร.สุทธิชัย นพนาคีพงษ์)

อาจารย์ที่ปรึกษาร่วม

(รศ.ดร.ปิติเขต สู้รักษา)

อาจารย์ที่ปรึกษาร่วม

(ผศ.กฤดากร กล่อมการ)

โครงงาน 512434

การเข้ารหัสลับของสัญญาณภาพโดยใช้รหัสลับอลวนแบบไม่ต่อเนื่อง

Image Encryption using Discrete Chaotic Signals

โดยนายชานนท์ จันทร์วงศ์ รหัสนักศึกษา 48010205

นายธันยะวัฒน์ สวงโท รหัสนักศึกษา 48010379

นายนิตินัย สามงามยา รหัสนักศึกษา 48010455

นายอภินันท์ อยู่สมบูรณ์ รหัสนักศึกษา 48012041

อาจารย์ที่ปรึกษา ผศ.ดร.สุทธิชัย นพนาคีพงษ์

อาจารย์ที่ปรึกษาร่วม รศ.ดร.ปิติเขต สู้รักษา

อาจารย์ที่ปรึกษาร่วม ผศ.กฤดากร กล่อมการ

บทคัดย่อ

โครงงานนี้นำเสนอเกี่ยวกับการรักษาความปลอดภัยในการติดต่อสื่อสารบนพื้นฐานของพลวัตรแบบไม่ต่อเนื่องและวงจรการเข้ารหัสสัญญาณภาพ ซึ่งใช้เทคนิคสัญญาณอลวน คีย์พารามิเตอร์ ที่ได้ศึกษามา โดยจะแสดงการส่งสัญญาณภาพจริงที่ทำการเข้ารหัสลับและประเมินคุณภาพของภาพที่ได้

Abstract

This project presents secure communication based on discrete dynamics and image encryption. The chaotic signal technique and key parameter are also studied. The findings will be demonstrated in real transmission of the cipher signal. Image quality will also be evaluated.

สารบัญ

สารบัญ (ต่อ)

สารบัญรูป

สารบัญรูป (ต่อ)

หน้า

สารบัญรูป (ต่อ)

หน้า

สารบัญตาราง

บทที่ 1
บทนำ

แนวคิดและที่มาของปัญหา

ในหลายปีมานี้การใช้เทคโนโลยีเป็นที่แพร่หลายมากในการสื่อสารในชีวิตประจำวันซึ่งใช้ในการติดต่อสื่อสาร ส่งข้อมูลในระบบดิจิตอล เช่น การส่งภาพ การประชุมโดยใช้ Conference การส่งข้อมูลเหล่านี้สามารถผ่านทางอินเตอร์เน็ต คัดลอกไฟล์ในรูปแบบ CD DVD USB ทั้งผู้รับและผู้ส่งข้อมูลจึงเน้นถึงความปลอดภัยของข้อมูลเป็นสิ่งที่สำคัญที่สุด

ทางคณะผู้จัดทำจึงมีความคิดที่จะสร้างความปลอดภัยของข้อมูล จึงทำโครงการนี้ขึ้น คือ Image Encryption Using Discrete Chaotic Signal โดยการเข้ารหัสลับ ซึ่งนำมาประยุกต์กับสมการอลวนเพื่อเป็นการเข้ารหัสลับในรูปแบบใหม่ โดยต่างจากการเข้ารหัสลับทั่วไป โดยไม่สามารถคาดเดารหัสหลังการเข้าสมการอลวนได้ จากการกำหนดค่า Seed คือค่าเริ่มต้นของสมการอลวนนำมา XOR กับข้อมูลภาพ โดยทางผู้ส่งและผู้รับจะต้องใช้ อุปกรณ์ ET-ICDX V1.0 นี้เพื่อเข้ารหัสลับและถอดรหัสลับ โดยมีค่า Seed ที่ตรงกันจึงจะสามารถเข้าและถอดรหัสของข้อมูลเหล่านี้ได้

ในปริญญานิพนธ์นี้ แบ่งออกเป็น 5 บท โดยในบทที่ 1 กล่าวถึงแนวคิดและที่มาของปัญหาในการทำปริญญานิพนธ์ ในบทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องของปริญญานิพนธ์ ประกอบด้วย ทฤษฎีการเข้ารหัสลับ ทฤษฎีอลวน ข้อมูลเบื้องต้นเกี่ยวกับภาพสกุล JPEG และโครงสร้างภาษาซี บทที่ 3 กล่าวถึงการคำนวณและการสร้างในส่วนซอร์ฟแวร์ คือ โปรแกรมเข้ารหัสลับ และฮาร์ดแวร์ คือการคำนวณและการออกแบบสมการโลจิสติกแมป ในอุปกรณ์ ET-ICDX V1.0 เพื่อใช้เป็นอัลกอริทึมในการคำนวณ บทที่ 4 กล่าวถึงผลการทดลองในส่วนต่างๆของโปรแกรมเข้ารหัสลับ บทที่ 5 กล่าวถึงสรุปและวิจารณ์ผลการทดลอง

ปริญญานิพนธ์นี้มีขอบเขตการทำงานคือ การเข้าและถอดรหัสลับของโปรแกรมต้องใช้ อุปกรณ์ตัวเดียวกัน ในปริญญานิพนธ์ โดยต้องมีสมการอลวน และ ค่า Seed ที่ตรงกัน และในส่วนสกุลของภาพในการเข้าและถอดรหัสลับ ปริญญานิพนธ์นี้ กล่าวถึงภาพสกุล JPEG เป็นสำคัญ จากการใช้กันอย่างแพร่หลายในปัจจุบัน

สำหรับอุปกรณ์ในปริญญานิพนธ์นี้ยังเป็นเพียงอุปกรณ์ต้นแบบอยู่โดยผู้ที่มีความสามารถนำอุปกรณ์ที่ได้ออกแบบไว้ไปพัฒนาต่อไปให้ได้สามารถใช้งานได้สะดวกและมีประสิทธิภาพมากยิ่งขึ้น โดยการทำโครงงานนี้ เน้นแต่เพียงใช้ในการรักษาความปลอดภัยในส่วนของผู้รับและผู้ส่งเท่านั้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องในการทำปริญญานิพนธ์นี้ ซึ่งในหัวข้อ 2.1 จะกล่าวถึงทฤษฎีความอลวน (chaos theory) ประกอบด้วยสมการต่างๆ หัวข้อ 2.2 คือการเข้ารหัส (encryption) ประกอบด้วยทฤษฎีเบื้องต้นในการเข้ารหัส และอัลกอริทึมต่างๆ หัวข้อ 2.3 คือ ส่วนประกอบของภาษาซี และหัวข้อที่ 2.4 คือ องค์ประกอบของภาพ และความหมายของสกุลต่างๆของภาพ

2.1 ทฤษฎีความอลวน (chaos theory)

ทฤษฎีความอลวน เป็นทฤษฎีที่อธิบายถึง ลักษณะพฤติกรรมของระบบพลวัต (คือ ระบบที่มีการเปลี่ยนแปลง เช่น เปลี่ยนแปลงตามเวลาที่เปลี่ยนไป) โดยลักษณะการเปลี่ยนแปลงของระบบที่เรียกว่าอลวนนี้จะมีลักษณะที่ปั่นป่วนจนดูคล้ายว่า การเปลี่ยนแปลงนั้นเป็นแบบสุ่มหรือไร้ระเบียบ (random/stochastic) แต่จริง ๆ แล้ว ระบบอลวนนี้เป็นระบบแบบไม่สุ่ม หรือระบบที่มีระเบียบ (deterministic)

ในทางคณิตศาสตร์และฟิสิกส์ คำจำกัดความของระบบอลวน คือ ระบบแบบไม่เป็นเชิงเส้น (Nonlinear system) ประเภทหนึ่ง ที่มีความไวต่อสภาวะเริ่มต้น กล่าวอีกนัยหนึ่งคือ ถ้าระบบ 2 ระบบนั้นเริ่มต้นจากสภาวะที่แตกต่างกันเพียงเล็กน้อย คือเกือบจะเหมือนกันทุกประการ เมื่อระบบได้มีการเปลี่ยนไปสักระยะหนึ่ง สภาวะของระบบทั้งสองที่เราสังเกตได้เมื่อเวลาผ่านไปจะแตกต่างกันอย่างสังเกตเห็นได้ชัด

เรามักจะได้ยินคำพูดที่นิยมพูดกันอย่างกว้างขวางที่ว่า "เด็ดดอกไม้สะเทือนถึงดวงดาว" หรือ "ผีเสื้อขยับปีกทำให้เกิดพายุ" (จาก "Butterfly effect") ซึ่งมีคนจำนวนไม่น้อยที่ตีความคำพูดนี้ในลักษณะของขนาดความรุนแรงของผลลัพธ์เท่านั้น ระบบอลวนนั้นไม่จำเป็นจะต้องแตกต่างกันในแง่ของ ขนาด ของผลลัพธ์เสมอไป แต่อาจแตกต่างกันในแง่ของพฤติกรรมการเปลี่ยนแปลงก็ได้ จากตัวอย่างข้างต้น การเปลี่ยนแปลงของระบบทั้งสองนั้นจะมีลักษณะที่คล้ายคลึงกันมากในขณะเริ่มต้น เมื่อเวลาผ่านไปการเปลี่ยนแปลงนั้นแทบจะเรียกได้ว่าไม่มีอะไรที่เหมือนกันเลย

จุดเริ่มต้นของทฤษฎีอลวนนี้ สามารถสืบย้อนกลับไปได้ถึงในช่วงปี พ.ศ. 2443 (ค.ศ. 1900) จากการศึกษาปัญหาวงโคจรของวัตถุสามชิ้นในสนามแรงดึงดูดระหว่างกัน ซึ่งมีชื่อเรียกเป็นทางการว่า ปัญหาสามวัตถุ (three-body problem) โดย Henry Poincare ซึ่งได้ค้นพบว่า วงโคจรที่ศึกษานั้นอาจจะมีลักษณะที่ไม่ได้เป็นวงรอบ (periodic) คือไม่ได้มีทางวิ่งซ้ำเป็นวงรอบ ยิ่งไปกว่านั้น วงโคจรนั้นก็ไม่ได้ขยายวงออกไปเรื่อย ๆ หรือมีลักษณะที่ลู่เข้าหาจุดใด ๆ ต่อมาได้มีการศึกษาถึงปัญหาสมการเชิงอนุพันธ์ไม่เป็นเชิงเส้นที่เกี่ยวข้องโดยที่ Berkoff นั้นศึกษาปัญหาสามวัตถุ Kolmogorov ศึกษาปัญหาความปั่นป่วน (หรือ Tuebulence) และปัญหาเกี่ยวกับดาราศาสตร์. ส่วน Cartwright และ Littlewood นั้นศึกษาปัญหาทางวิศวกรรมการสื่อสารด้วยคลื่นวิทยุ Smell นั้นอาจเป็นนักคณิตศาสตร์คนแรก ที่ทำการศึกษาถึงปัญหาทางด้านพลศาสตร์ของระบบไม่

เป็นเชิงเส้น ถึงแม้ว่าความอลวนของเส้นทางโคจรของดาว นั้นยังไม่ได้มีการทำการสังเกตบันทึกแต่อย่างใด แต่ก็ได้มีการสังเกตพบ พฤติกรรมความอลวนในความปั่นป่วนของการเคลื่อนที่ของของไหล และ ในการแบบไม่เป็นวงรอบของวงจรวิทยุ ซึ่งไม่มีทฤษฎีใดในขณะนั้นสามารถอธิบายพฤติกรรมเหล่านี้ได้

ความตื่นตัวในการพัฒนาทฤษฎีความอลวนนี้ เกิดขึ้นในช่วงกลางของศตวรรษที่ 20 เมื่อเป็นที่ประจักษ์ว่า ทฤษฎีของระบบเชิงเส้นนั้นไม่สามารถใช้อธิบายพฤติกรรมบางอย่าง แม้กระทั่งพฤติกรรมของ

ระบบที่ไม่ซับซ้อนอย่าง Logistic map อีกปัจจัยหนึ่งที่ส่งผลให้พัฒนาการของทฤษฎีความอลวนเป็นไปอย่างรวดเร็วก็คือ คอมพิวเตอร์ การคำนวณในทฤษฎีความอลวนนั้น โดยส่วนใหญ่จะมีลักษณะที่เป็นการคำนวณค่าแบบซ้ำ ๆ จากสูตรคณิตศาสตร์ และสามารถใช้คอมพิวเตอร์ช่วยในการคำนวณได้อย่างมีประสิทธิภาพ

Edward Lorenz เป็นผู้ริเริ่มบุกเบิกทฤษฎีความอลวน เขาได้สังเกตพฤติกรรมความอลวน ในขณะทำการทดลองทางด้านการพยากรณ์อากาศ ในปี ค.ศ. 1961 ลอเรนซ์ใช้คอมพิวเตอร์จำลองสภาพอากาศ ซึ่งในการคำนวณครั้งถัดมาเขาไม่ต้องการเริ่มจำลองจากจุดเริ่มต้นใหม่ เพื่อประหยัดเวลาในการคำนวณ จึงใช้ข้อมูลในการคำนวณก่อนหน้านี้เพื่อเป็นค่าเริ่มต้น ปรากฏว่าค่าที่คำนวณได้มีความแตกต่างไปจากเดิมอย่างสิ้นเชิง เขาพบว่าสาเหตุเกิดจากการปัดเศษ ของค่าที่พิมพ์ออกมา จากค่าที่ใช้ในคอมพิวเตอร์ ซึ่งมีค่าน้อยมาก แต่สามารถนำไปสู่ความแตกต่างอย่างมากมาย เรียกว่า ไวต่อสภาวะเริ่มต้น

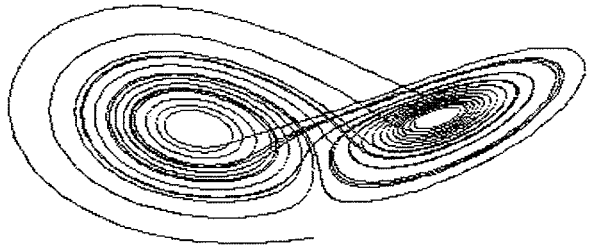
คำ "butterfly effect" ซึ่งเป็นคำที่นิยมใช้เมื่อกล่าวถึงทฤษฎีความอลวน นั้นมีที่มาไม่ชัดเจน เริ่มปรากฏแพร่หลายหลังจากการบรรยายของ Lorenz ในปี ค.ศ. 1972 ภายใต้ชื่อหัวข้อ "Does the Flap of a Butterfly's Wings in Brazil Set Off a Tornado in Texas" นอกจากนี้แล้วยังอาจมีส่วนมาจาก รูปแนวโคจรของตัวดึงดูดLorenz ดังสมการ Lorenz ที่มีรูปร่างคล้ายผีเสื้อ ซึ่งได้ตีพิมพ์ในบทความวิชาการก่อนหน้านี้

$$\frac{dx}{dt} = \sigma (y - x)$$

$$\frac{dy}{dt} = rx - y - xz$$

$$\frac{dz}{dt} = xy - bz$$

สมการLorenz



รูปที่ 2.1 แนวโคจรของตัวดึงดูดLorenz

ในปริญญานิพนธ์นี้ ใช้กระบวนการอลวน (Chaotic pattern) แบบ Logistic map ในการเข้ารหัสลับ โดยมีรายระเอียดดังต่อไปนี้

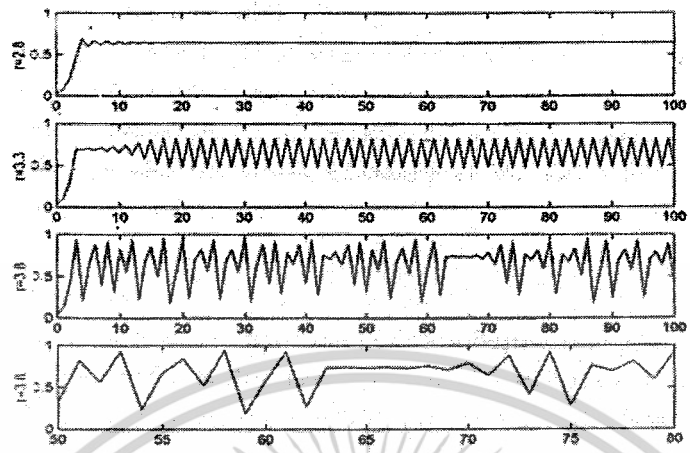
2.1.1 The Logistic map

สมการ Logistic map เป็นสมการแบบไม่เชิงเส้น ซึ่ง output คือ (X_{n+1} ขึ้นกับ X_n) โดยสมการแบบ logistic เป็นลักษณะ recursive ซึ่งหมายความว่าเทอมที่สามเป็นฟังก์ชันของตัวที่สอง, เทอมที่สี่เป็นฟังก์ชันของตัวที่สามไปเรื่อยๆ โดยแสดงสมการได้ดังนี้

$$x_{n+1} = rx_n(1 - x_n) \qquad (2.1)$$

r คือค่าคงที่ เมื่อเปลี่ยนค่า r ไปเรื่อยๆ ผลจากกราฟ เมื่อค่า r = 2.8 กราฟที่ได้จะเป็นแบบ damped harmonic oscillator ตัวอย่างเช่น การแกว่งของลูกตุ้ม ซึ่งลูกตุ้มก็จะแกว่งช้าลงๆ จนหยุด สำหรับ r = 3.3 สัญญาณจะเป็นสัญญาณ oscillator โดยไม่มีการ damping และ oscillator นี้จะดำเนินต่อเนื่องไปไม่มีที่สิ้นสุด

เมื่อค่า r มากกว่า 3.3 logistic map จะเริ่มแสดงปรากฏการณ์ อลวน ดูจากกราฟข้างล่างจะเห็นความแตกต่างของจุด peak และการแตกต่างกันเหลี่ยมคมของกราฟ โดยสมการจะให้ความแตกต่างกันของรูปสำหรับค่า x_n ใดๆ



รูปที่ 2.2 แสดงการเปลี่ยนแปลง Parameter โดยการเพิ่มค่า r จาก 3.3 เป็น 3.8

2.1.2 แผนผัง Bifurcation ของ logistic map

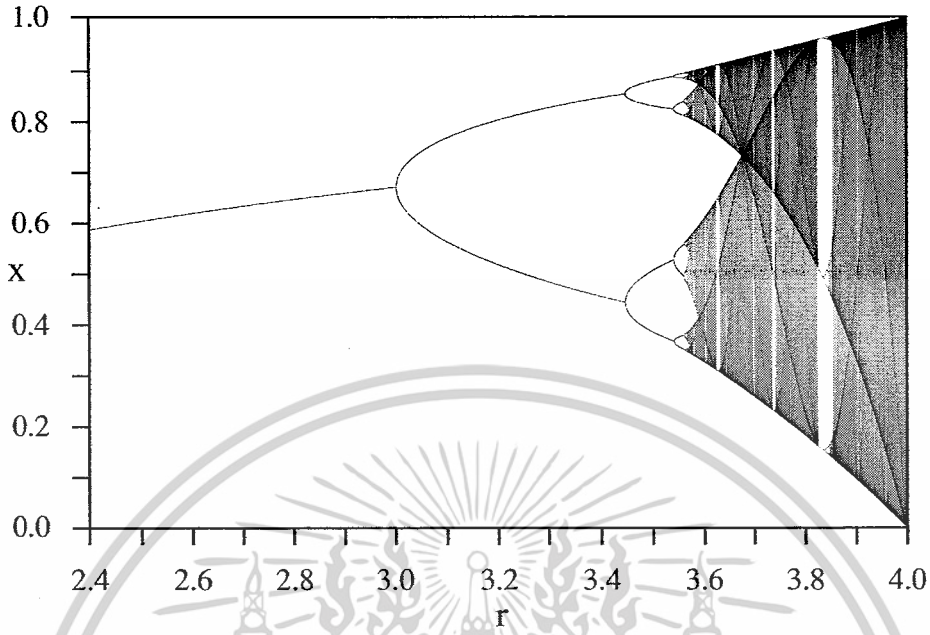
พฤติกรรมของระบบ ที่ค่าพารามิเตอร์ r ต่างๆ

1. ช่วง 1. $0 \leq r \leq 1$ ประชากรจะตายไปจนหมดโดยไม่ขึ้นกับค่าเริ่มต้น โดยระบบมีจุดตายตัว (fixed point) เพียงจุดเดียวที่ $x = 0$ ซึ่งเป็นจุดตายตัวแบบดึงดูด (attracting fixed point) หรือเรียกว่า "จุดดูดซับ" (sink) และดึงดูดค่าเริ่มต้นทุกค่าใน [0,1]
2. ช่วง $1 < r \leq 3$ ระบบมีจุดตายตัว 2 จุดคือที่ $x = 0$ และ $x = (r - 1) / r$ โดยที่ $x = 0$ เป็นจุดตายตัวแบบผลักออก (repelling fixed point) หรือ เรียกว่า "จุดกำเนิด" (source) และ $x = (r - 1) / r$ เป็นจุดตายตัวแบบดึงดูด
   - ที่ค่า r อยู่ระหว่าง 1 ถึง 2 จำนวนประชากรจะลู่เข้าหาค่า $(r - 1) / r$ และคงตัวอย่างรวดเร็ว
   - ที่ค่า r อยู่ระหว่าง 2 และ 3 จำนวนประชากรจะเริ่มแกว่งก่อนลู่เข้าหาจุดดูดซับ โดยมีอัตราการลู่เข้าเป็นเชิงเส้น
   - ที่ค่า r เท่ากับ 3 อัตราการลู่เข้าจะช้ากว่าอัตราที่เป็นเชิงเส้น
3. ช่วง $3 < r \leq 3.45$ จำนวนประชากรจะมีค่าแกว่งสลับระหว่างค่า 2 ค่า ซึ่ง 2 ค่านี้ขึ้นกับค่า r แต่ไม่ขึ้นกับค่าเริ่มต้น ซึ่งก็คือ ระบบมีจุดวงรอบคาบ 2 แบบดึงดูด หรือ จุดดูดซับแบบวงรอบคาบ2
4. ช่วง $3.45 < r \leq 3.54$ จำนวนประชากรจะมีค่าแกว่งสลับระหว่างค่า 4 ค่า ไม่ขึ้นกับค่าเริ่มต้น ซึ่งก็คือ ระบบมีจุดดูดซับแบบวงรอบคาบ 4

5. เมื่อค่า r มีค่าเพิ่มมากกว่า 3.54 จำนวนประชากรจะมีค่าแกว่งสลับ เป็นวงรอบด้วยคาบ 8,16,32 และเพิ่มขึ้นเรื่อยๆ ช่วงการเพิ่มของค่า r ที่ส่งผลให้คาบวงรอบการแกว่งเพิ่มขึ้นจะลดลงอย่างรวดเร็ว สัดส่วนของระยะของค่าพารามิเตอร์ที่ทำให้มีการเพิ่มคาบ (หรือเรียก ช่วงระยะไบเฟอร์เคชัน) ที่อยู่ถัดกัน จะลู่เข้าหา Feigenbaum δ = 4.669 ซึ่งพฤติกรรมดังกล่าวนี้จะไม่ขึ้นกับค่าเริ่มต้นแต่อย่างใด
6. ที่ค่า r = 3.57 (โดยประมาณ) เป็นจุดที่ระบบเริ่มมีพฤติกรรมความอลวน ระบบจะไม่มีพฤติกรรมการแกว่งเป็นวงรอบดังค่า r ที่ผ่านมา แต่ระบบจะมีพฤติกรรมที่ไวต่อค่าเริ่มต้นซึ่งเป็นคุณลักษณะของความอลวน ความแตกต่างเพียงเล็กน้อยของค่าจำนวนประชากรเริ่มต้น จะมีผลต่อการเปลี่ยนแปลงของค่าประชากรในระยะยาว
7. ค่า r ระหว่าง 3.57 และ 4 มีหลายค่าที่ระบบมีพฤติกรรมความอลวน แต่ก็ยังมีค่า r บางค่าที่ระบบไม่แสดงพฤติกรรมความอลวน ตัวอย่างเช่น ที่ r ประมาณ 3.82 จะมีบางช่วงที่ระบบมีพฤติกรรมแกว่งเป็นวงรอบคาบ 3 และที่ค่า r สูงขึ้นเล็กน้อยจะแกว่งเป็นวงรอบคาบ 6, 12 และเพิ่มขึ้นตามลำดับ และจะมีบางช่วงที่มีการแกว่งเป็นคาบ 5 และอื่นๆ ซึ่งพฤติกรรมทั้งหมดนี้จะมีการแกว่งเป็นวงรอบ และไม่ขึ้นกับค่าเริ่มต้น
8. ที่ค่า r = 4 และมากกว่านั้น ค่าของระบบจะลู่ออก สำหรับทุกค่าเริ่มต้นใน [0,1]

แผนผังBifurcation ดังรูปที่ 2.3 แสดงให้เห็นถึงพฤติกรรมดังกล่าวข้างต้นนี้ โดยที่แกนนอนของแผนผังเป็น ค่า r และ แกนตั้งเป็นค่าจำนวนประชากร หรือค่าของระบบในระยะยาว

แผนผังBifurcationนี้เป็น Fractal ถ้าเราพิจารณาที่ค่า r = 3.82 ที่ระบบมีพฤติกรรมแกว่งเป็นวงรอบคาบ 3 เมื่อเราเลือกกิ่งหนึ่งใน 3 และขยายที่กิ่งนั้นเราจะเห็นภาพที่มีลักษณะเหมือนภาพเดิม



รูปที่ 2.3 แผนผังBifurcation ของlogistic map

2.1.3 Hyper Chaos

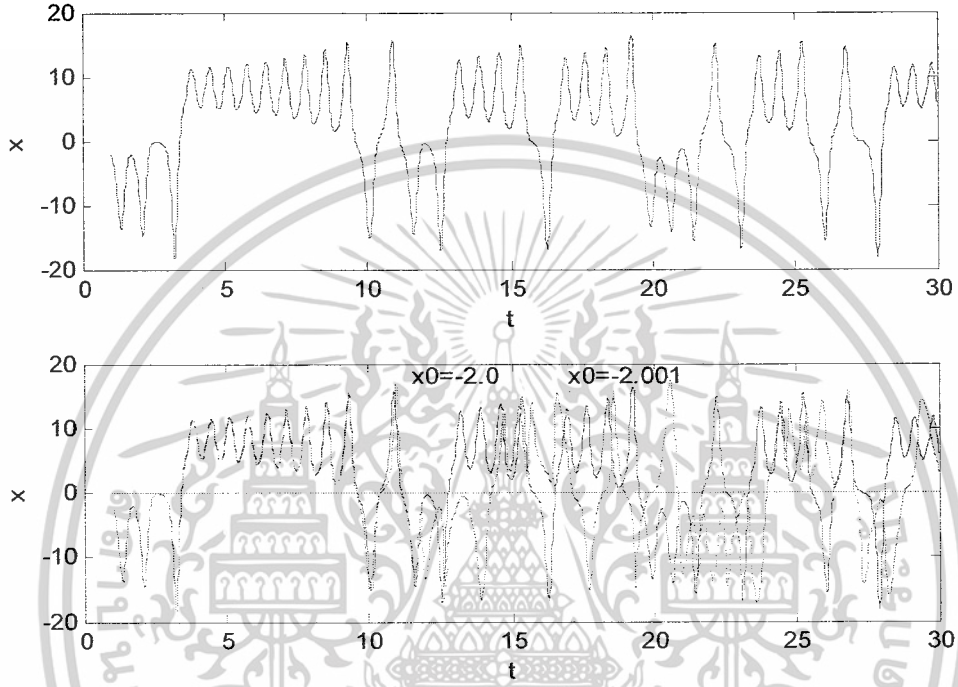
คือสัญญาณอลวน ที่มีความยุ่งเหยิง คาดเดาได้ยากกว่าสัญญาณอลวนธรรมดา ตัวอย่างสมการที่ใช้ในการสร้างสัญญาณนี้ได้แก่ Hyper Henon ผู้ค้นพบสมการ Hyper Henon คือนักวิจัยชื่อ Baier และ Klein ชาวเยอรมัน ซึ่งตีพิมพ์ในวารสาร PHYSICES LATTERS A Volume 151, Number 6, 7 วันที่ 17 December 1990 มีสมการดังนี้

$$\left.\begin{aligned} x_1(k+1) &= r - x_2^2(k) - 0.1x_3(k) \\ x_2(k+1) &= x_1(k) \\ x_3(k+1) &= x_2(k) \end{aligned}\right\} \qquad (2.2)$$

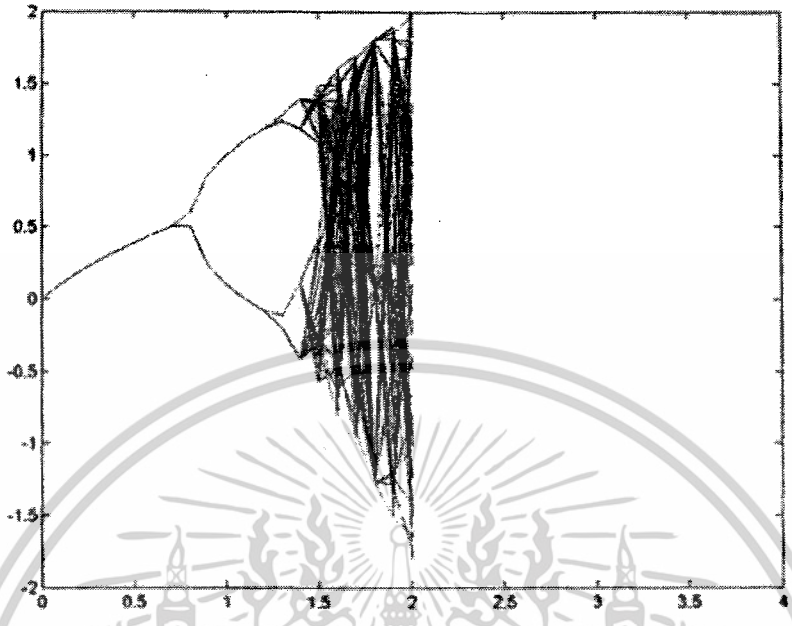
ค่าพารามิเตอร์สำหรับปรับให้เกิดสัญญาณอลวนคือค่า r

r คือค่าคงที่ เมื่อเปลี่ยนค่า r ไปเรื่อยๆ ผลจากกราฟ เมื่อค่า r = 0.5 กราฟที่ได้จะเป็นแบบ damped harmonic oscillator ตัวอย่างเช่น การแกว่งของลูกตุ้ม ซึ่งลูกตุ้มก็จะแกว่งช้าลงๆ จนหยุด สำหรับ r = 1.3 สัญญาณจะเป็นสัญญาณ oscillator โดยไม่มีการ damping และ oscillator นี้จะดำเนินต่อเนื่องไปไม่มีที่สิ้นสุด

เมื่อค่า r มากกว่า 1.3 logistic map จะเริ่มแสดงปรากฏการณ์อลวน ดูจากรูปที่ 2.4 จะเห็นความแตกต่างของจุด peak และการแตกต่างกันเหลี่ยมคมของกราฟ โดยสมการจะให้ความแตกต่างกันดังรูปที่ 2.4 สำหรับค่า x_n ใดๆ



รูปที่ 2.4 การเปลี่ยนแปลง parameter โดยการเปลี่ยนค่า x_n จาก -2.0001 เป็น -2.0



รูปที่ 2.5 แผงผัง Bifurcation ของสมการ Hyper Henon ที่ r ระหว่าง 0 ถึง 4

จากรูปที่ 2.5 แผงผัง Bifurcation Diagram อธิบายได้ว่า เมื่อ

$r < 0$ ค่าสัญญาณที่เกิดจากสมการอลวน จะมีค่า $x = 0$ เมื่อเวลาผ่านไปช่วงหนึ่ง

$r < 0.6$ ค่าสัญญาณที่เกิดจากสมการอลวน จะมีค่า $x =$ ค่าคงที่

$r = 0.6$ ค่าสัญญาณที่เกิดจากสมการอลวน จะมีลักษณะเป็น 2 ค่า หรือสภาวะ Oscillate

$r < 1.3$ ค่าสัญญาณที่เกิดจากสมการอลวน จะมีลักษณะเป็น 4 ค่า หรือสภาวะ Double Period

$r > 1.3$ สัญญาณจะมีลักษณะเป็นอลวน

2.1.4 สมการ Lorenz

สมการ Lorenz ถูกค้นพบในปี 1963 โดย Ed Lorenz โดยพบในรูปแบบของการหมุนในบรรยากาศเหนือชั้น atmosphere ขึ้นไป ที่ถูกลดรูปลงมา หลังจากนั้นก็ได้ค้นพบสมการที่เหมือนกันนี้ปรากฏในการศึกษาของลำแสงเลเซอร์ แบตเตอรี่ และกังหันน้ำธรรมดาแบบ Chaotic ที่สามารถสร้างได้โดยง่าย

Lorenz พบว่าเส้นโคจรของระบบอันนี้ สำหรับการตั้งค่าที่กำหนด ไม่ตั้งที่ fixed point และไม่เข้าใกล้รอบที่ทำให้คงที่ และกระจายออกไม่สิ้นสุด โดย Lorenz ถูกพบอยู่ที่เวลาที่ไม่มีผู้ที่เกี่ยวข้องในเชิงคณิตศาสตร์สนใจและได้ถูกมองเข้ามาเป็นเวลานานหลายปี แต่ปัจจุบัน attractor นี้มีชื่อเสียงที่สุดในความเป็น attractor ที่แปลก ซึ่งเกี่ยวข้องกับ Chaos

สมการของ Lorenz คือสมการ differential equation 3 สมการ ซึ่งเป็นสมการ first order ของ x, y และ z โดย r, b และซิกมาคือพารามิเตอร์ที่เปลี่ยนพฤติกรรมของระบบดังนี้ คือ

$$\frac{dx}{dt} = \sigma (y - x)$$
$$\frac{dy}{dt} = rx - y - xz \qquad (2.3)$$
$$\frac{dz}{dt} = xy - bz$$

2.1.5 Chen's chaotic

รูปแบบของสมการของ Chen's chaotic คือ

$$dx/dt = a(y - x)$$
$$dy/dt = (c - a)x - xz + cy \qquad (2.4)$$
$$dz/dt = xy - bz$$

โดย ค่า a, b, c เป็นค่าคงที่ (a, b, c ต้องเป็นค่าที่เหมาะสมที่ทำให้เกิดปรากฏการณ์ chaos)

2.1.6 สมการ Rossler

รูปแบบสมการของ Rossler คือ

$$dx/dt = -y - z$$
$$dy/dt = x + a\,y \qquad (2.5)$$
$$dz/dt = b + z(x - c)$$

โดยที่ ค่า a, b, c เป็นค่าคงที่ (a, b, c ต้องเป็นค่าที่เหมาะสมที่ทำให้เกิดปรากฏการณ์ chaos)

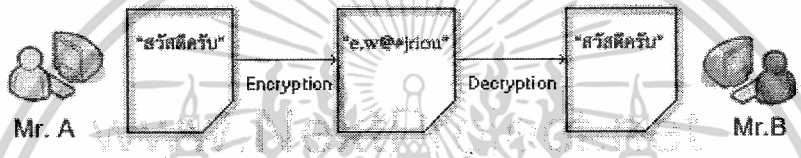
2.2 การเข้ารหัสข้อมูล (Encryption)

การเข้ารหัสข้อมูล หมายถึง วิธีการที่ทำเปลี่ยนแปลงข้อมูลเพื่อไม่ให้สามารถแปลความได้จากบุคคลที่เราไม่ต้องการให้เขาเข้าใจข้อมูล ส่วนการถอดรหัสข้อมูล นั้นจะมีวิธีการที่ตรงกันข้ามกับการเข้ารหัสข้อมูล กล่าวคือการถอดรหัส หมายถึง วิธีการที่ทำการเปลี่ยนแปลงข้อมูลที่ได้จากการเข้ารหัสข้อมูล เป็นข้อมูลก่อนที่จะถูกทำการเข้ารหัส การที่จะทำให้ข้อมูลเป็นความลับ จุดหลักคือ ต้องไม่ให้ข้อมูลความลับนี้ถูกอ่านโดยบุคคลอื่น แต่ให้ถูกอ่านได้โดยบุคคลที่เราต้องการให้อ่านได้เท่านั้น โดยการนำเอาข้อความเดิมที่สามารถอ่านได้ มาทำการเข้ารหัสก่อน เพื่อเปลี่ยนแปลงข้อความเดิมให้ไปเป็นข้อความที่เราเข้ารหัส ก่อนที่จะส่งต่อไปให้

บุคคลที่เราต้องการที่จะติดต่อด้วย เพื่อป้องกันไม่ให้บุคคลอื่นสามารถที่จะแอบอ่านข้อความที่ส่งมาโดยที่ข้อความที่เราเข้ารหัสแล้ว

2.2.1 ระบบการเข้ารหัสข้อมูล (Cryptography)

ระบบการเข้ารหัสข้อมูล เป็นกระบวนการสำหรับการแปรรูปข้อมูลอิเล็กทรอนิกส์ธรรมดาให้อยู่ในรูปที่บุคคลทั่วไปไม่สามารถอ่านเข้าใจได้ ซึ่งโดยทั่วไปแล้วการเข้ารหัสจะกระทำก่อนการจัดเก็บข้อมูลหรือก่อนการส่งข้อมูล โดยการนำข้อมูลอิเล็กทรอนิกส์ธรรมดากับกุญแจ ซึ่งเป็นตัวเลขสุ่มใดๆ มาผ่านกระบวนการทางคณิตศาสตร์ ผลที่ได้ก็คือข้อมูลที่เข้ารหัส ขั้นตอนที่กล่าวมานี้จะเรียกว่า การเข้ารหัส และเมื่อต้องการอ่านข้อมูล ก็นำเอาข้อมูลที่เข้ารหัสกับกุญแจมาผ่านกระบวนการทางคณิตศาสตร์ ผลลัพธ์ที่ได้ก็คือข้อมูลดั้งเดิม ซึ่งขั้นตอนนี้จะเรียกว่า การถอดรหัส



รูปที่ 2.6 การเข้ารหัสข้อมูล

จุดประสงค์ที่สำคัญ 3 ประการของการเข้ารหัสข้อมูลประกอบด้วย

1. การทำให้ข้อมูลเป็นความลับ เพื่อป้องกันไม่ให้ผู้ที่ไม่มีสิทธิ์ในการเข้าถึงข้อมูลสามารถเข้าถึงข้อมูลได้
2. การทำให้ข้อมูลสามารถตรวจสอบความสมบูรณ์ได้ เพื่อป้องกันข้อมูลให้อยู่ในสภาพเดิมอย่างสมบูรณ์ กล่าวคือ ในกระบวนการสื่อสารนั้นผู้รับ ได้รับข้อมูลที่ถูกต้องตามที่ผู้ส่ง ส่งมาให้โดยข้อมูลจะต้องไม่มีการสูญหายหรือถูกเปลี่ยนแปลงแก้ไขใดๆ
3. การทำให้สามารถพิสูจน์ตัวตนของผู้ส่งข้อมูลได้ เพื่อให้สามารถตรวจสอบได้ว่าใครคือผู้ส่งข้อมูล หรือในทางตรงกันข้าม ก็คือเพื่อป้องกันการแอบอ้างได้

ความต้องการของเทคโนโลยีการเข้ารหัสข้อมูล

- การรักษาความลับ (Confidentiality) คือความสามารถในการที่จะรักษาความลับที่ไม่ให้ผู้อื่นที่ไม่มีสิทธิ์เข้าถึงข้อมูลภายในระบบได้
- การรักษาความถูกต้อง (Integrity) คือความสามารถในการรักษาความถูกต้องและสมบูรณ์ของข้อมูล
- การระบุตัวบุคคลได้ (Authenticity) คือการที่เราสามารถที่จะระบุตัวตนของผู้ที่การเข้าถึงข้อมูลภายในระบบได้
- การป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) คือความสามารถในการป้องกันการปฏิเสธความรับผิดชอบของการเข้าถึงข้อมูลภายในระบบ

2.2.2 Block cipher

Block cipher เป็นการเปลี่ยนรูปของข้อมูล plaintext ที่มีขนาดเป็นแบบ fixed-length block ไปเป็นข้อมูล ciphertext ที่มีความยาวของข้อมูลที่เท่ากัน การเปลี่ยนรูปนี้เกิดขึ้นภายใต้กุญแจรหัสลับ (secret key) การถอดรหัสจะทำการเปลี่ยนรูป ciphertext ให้เป็นรูปแบบเดิมคือ plaintext โดยใช้กุญแจรหัสลับอันเดียวกัน ความยาวที่ถูกกำหนดไว้ (fixed length) เรียกว่า block size และแต่ละ block size จะมีขนาด 64 บิท อัลกอริทึมของ block cipher ได้แก่ DES, IDEA, AES (Rijndael), Blowfish , Twofish

เนื่องจากว่า plaintext ที่ต่างกันนั้นจะถูกเปลี่ยนรูปไปเป็น ciphertext ที่ต่างกันโดย block cipher จะใช้เรียงสับเปลี่ยน (permutation) กับชุดของข้อความทั้งหมด ซึ่งเรียงสับเปลี่ยนจะทำให้เกิดการเข้ารหัสขึ้น เพราะว่าการเรียงสับเปลี่ยนคือฟังก์ชันของกุญแจรหัสลับ

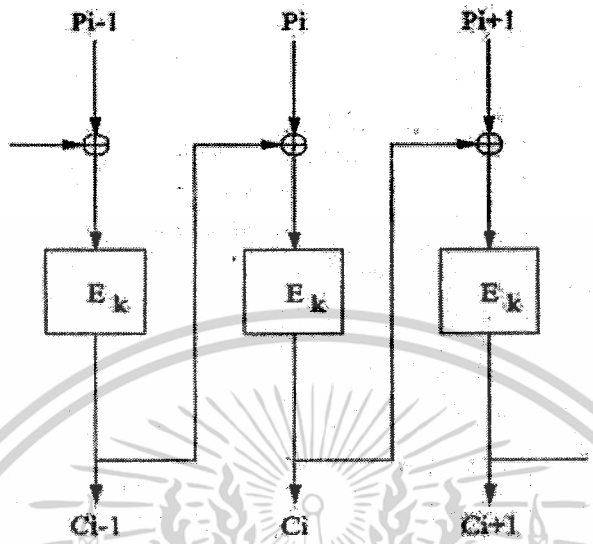
2.2.2.1 Iterated Block Cipher

Iterated Block Cipher คือการเข้ารหัส plaintext โดยการประมวลผลหลายๆ รอบ ในแต่ละรอบจะใช้กุญแจย่อย (subkey) ในการเปลี่ยนรูปแบบที่เหมือนกันคือ round function ชุดของกุญแจย่อยจะมาจากกุญแจรหัสลับ (secret key) โดยดึงมาจากตารางกุญแจ (key schedule) อีกทีหนึ่ง

จำนวนรอบของ iterated cipher ขึ้นกับระดับความปลอดภัยและประสิทธิภาพในกรณีส่วนใหญ่ จำนวนรอบที่เพิ่มจะทำให้ความปลอดภัยของ block cipher เพิ่มขึ้น แต่บาง cipher นั้นจำนวนรอบที่ต้องการเพื่อให้เกิดความปลอดภัยนั้น อาจจะมากเกินไปสำหรับ cipher ที่ต้องการจริง

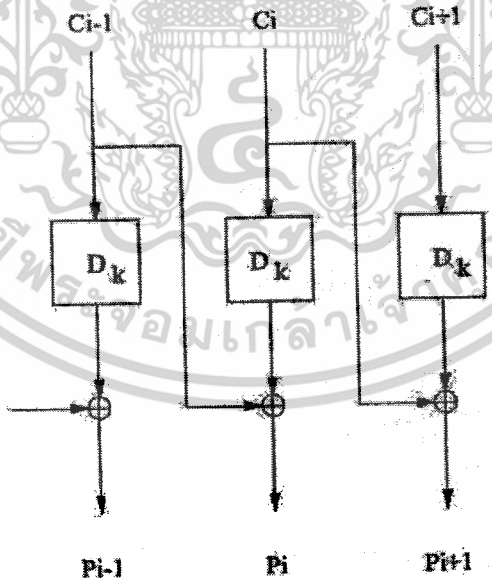
2.2.2.2 Block Encryption Mode

Block cipher มี mode มาตรฐานคือ Electronic Code Book, Cipher Block Chaining Cipher Feedback และ Output Feedback



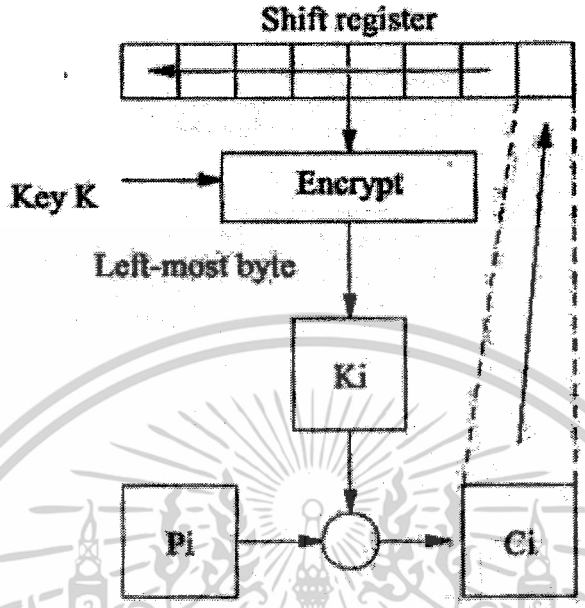
CBC Encryption

รูปที่ 2.7 การเข้ารหัสของ CBC

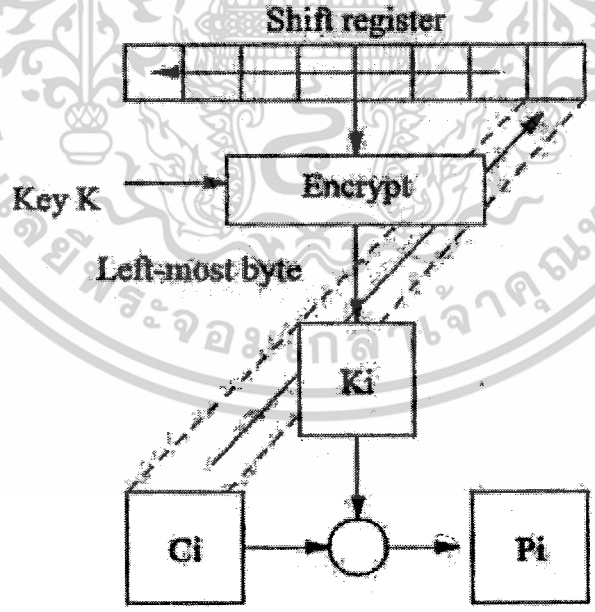


CBC Encryption

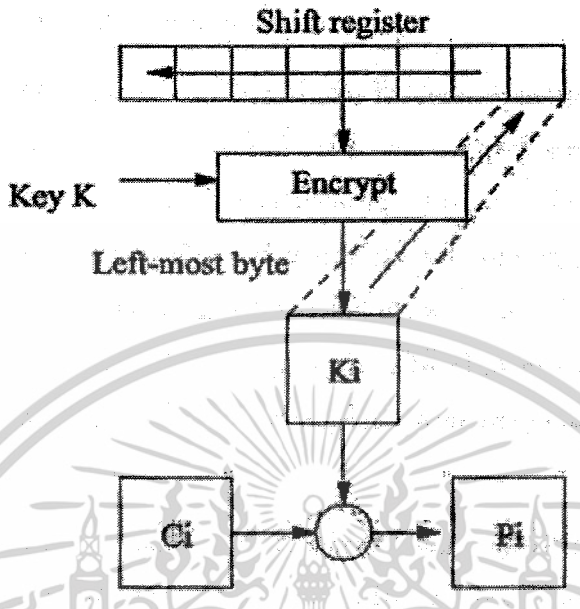
รูปที่ 2.8 การถอดรหัสของ CBC



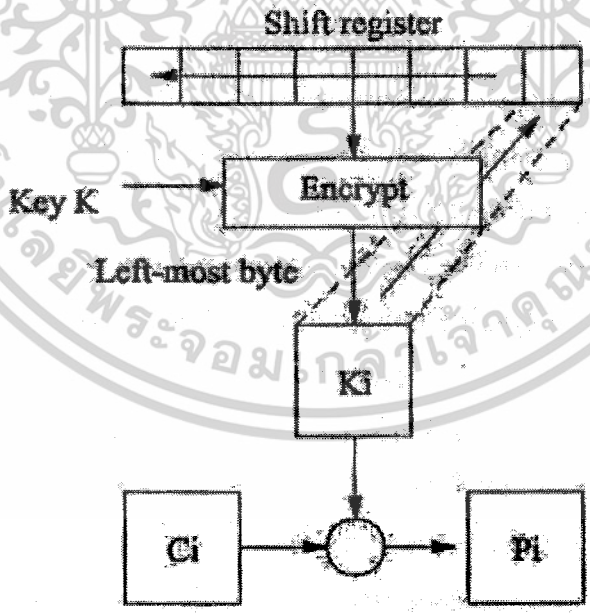
รูปที่ 2.9 การเข้ารหัสของ CFB



รูปที่ 2.10 การถอดรหัสของ CFB



รูปที่ 2.11 การเข้ารหัสของ OFB



รูปที่ 2.12 การถอดรหัสของ OFB

2.2.2.3 Weak Key ของ Block Cipher

Weak Key คือกุญแจรหัสลับที่มีค่าแน่นอน ซึ่ง block cipher ในการเข้ารหัสจะเป็นแบบแผนแน่นอน หรือเป็นระดับของการเข้ารหัสที่เจาะรหัสได้ง่าย ตัวอย่างเช่น DES ซึ่งมี 4 กุญแจที่การเข้ารหัสและถอดรหัสจะเหมือนกันอย่างแน่นอน ซึ่งหมายความว่าถ้าเข้ารหัสสองครั้งด้วย weak key หนึ่งตัว แล้ว plaintext จะสามารถทำให้กลับมาดังเดิมได้ สำหรับ IDEA จะมีคลาสของกุญแจที่เทคนิคถอดรหัสลับถูกทำให้ง่ายเป็นอย่างมาก และกุญแจสามารถกลับมาเหมือนเดิมได้ อย่างไรก็ตามในกรณีทั้งสองนี้จำนวนของ weak key คือส่วนเล็กของกุญแจที่เป็นไปได้ที่โอกาสของการเลือกโดยการสุ่มจะมีค่าเพียงเล็กน้อย ในกรณีเช่นนั้นไม่เป็นส่วนที่สำคัญจะทำลายความปลอดภัยของ block cipher ในการเข้ารหัส

2.2.3 Stream Cipher

Stream Cipher คืออัลกอริทึมการเข้ารหัสแบบสมมาตร โดย stream cipher ถูกออกแบบให้มีความเร็วซึ่งจะมีความเร็วมากกว่า block cipher ในขณะที่ block cipher กระทำบนกลุ่มที่ใหญ่ของข้อมูล (large blocks of data) แต่ stream cipher จะกระทำบนหน่วย (bit) ของ plaintext ที่มีขนาดเล็กกว่า การเข้ารหัสของ plaintext ใดๆ ด้วย block cipher ก็จะเกิด ciphertext ที่เหมือนกันเวลาเมื่อใช้กุญแจเดียวกัน ส่วน stream cipher นั้นการเปลี่ยนรูปของหน่วย plaintext ที่เล็กกว่าเหล่านี้จะมีหลากหลาย และขึ้นอยู่กับการประมวลผลการเข้ารหัส

Stream cipher สร้าง keystream ขึ้น และการเข้ารหัสจะทำโดยการรวม keystream กับ plaintext โดยใช้การ bitwise-XOR การสร้าง keystream จะเป็นอิสระจาก plaintext และ ciphertext (เรียก synchronous stream cipher) หรือขึ้นอยู่กับข้อมูลและการเข้ารหัสนั้น (เรียก self - synchronizing) การออกแบบ stream cipher ส่วนใหญ่นั้นเพื่อสนับสนุน synchronous stream cipher

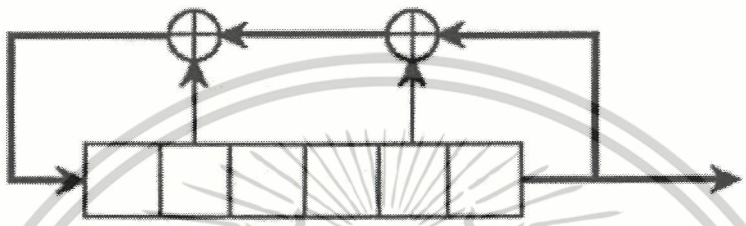
ความสนใจ stream cipher ในปัจจุบันนั้นจะมุ่งไปที่คุณสมบัติของ one-time pad แต่ที่นั่นไม่มีการทำ stream cipher เป็นมาตรฐานเหมือนกับกรณีของ block cipher ที่น่าสนใจก็คือ โหมดของ block cipher นั้นสามารถใช้เป็นตัวสร้าง keystream ได้ และโดยวิธีสามารถใช้ block cipher ใดๆ แทน stream cipher ได้ อย่างไรก็ตาม stream cipher ที่มีการออกแบบโดยเฉพาะนั้นจะเร็วกว่ามากอัลกอริทึมของ stream cipher ได้แก่ RC4, LFSRs, A5, SEAL, WEAK เป็นต้น

2.2.3.1 Linear Feedback Shift Register

LFSR คือกลไกสำหรับการสร้างอันดับของบิทไบนารี่ โดยรีจิสเตอร์ประกอบด้วยชุดของเซลล์ซึ่งกำหนดโดยเวกเตอร์เริ่มต้น ซึ่งก็คือกุญแจรหัสลับ พฤติกรรมของรีจิสเตอร์ถูกปรับโดยสัญญาณนาฬิกาและที่แต่ละสัญญาณนาฬิกานั้น เนื้อหาของเซลล์ในรีจิสเตอร์จะถูกเคลื่อนไปทางขวาหนึ่งตำแหน่ง และการ XOR

ของเซ็ตย่อย (subset) ของเนื้อหาในเซลล์ถูกย้ายไปในเซลล์ซ้ายสุด หนึ่งบิทของเอาท์พุทจะได้มาระหว่างกระบวนการอัพเดท

LFSRs เร็วและง่ายในการสร้างฮาร์ดแวร์และซอฟต์แวร์ โดยตัวเลือกของ feedback taps นั้นลำดับที่เกิดขึ้นสามารถมีลักษณะภายนอกทางสถิติที่ดี อย่างไรก็ตามลำดับที่สร้างขึ้นโดย LFSRs เดียวจะไม่ปลอดภัย เพราะการประมวลผลทางคณิตศาสตร์ถูกพัฒนาเป็นเวลาหลายปีในการที่จะพิจารณาวิเคราะห์โดยตรง



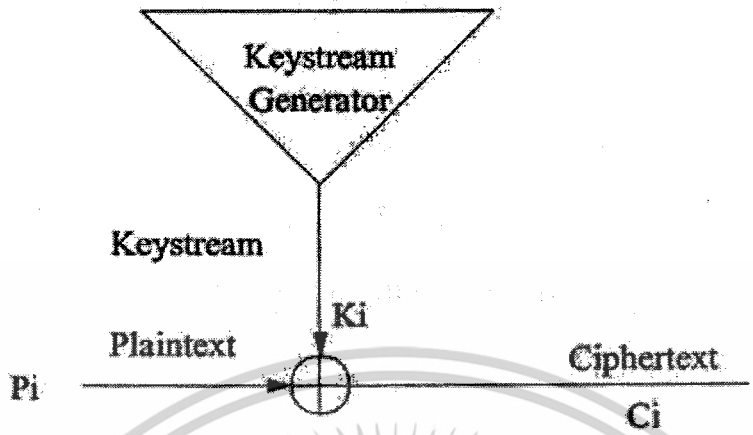
รูปที่ 2.13 การทำงานของ LFSR

2.2.3.2 One – time Pad

One – time Pad บางครั้งจะเรียกว่า Vernam cipher ซึ่งจะใช้ string ของบิตที่ถูกสร้างขึ้นอย่างสมบูรณ์โดยการสุ่ม โดย keystream จะมีความยาวที่เท่ากันกับข้อความ plaintext และค่า random string จะถูกรวมกันโดยการทำ XOR bitwise กับ plaintext เพื่อสร้าง ciphertext เพราะว่า keystream ทั้งหมดคือเป็นการสุ่มขึ้น ดังนั้นฝ่ายตรงข้ามจะมีทรัพยากรเชิงคำนวณที่ไม่จำกัด ซึ่งสามารถคาดเดา plaintext หากทราบ ciphertext ดังนั้นจึงกล่าวได้ว่า cipher จะมีความลับที่สมบูรณ์แบบและการวิเคราะห์ของ One – time Pad ถูกมองว่าเป็นหลักของวิธีการเข้ารหัสสมัยใหม่

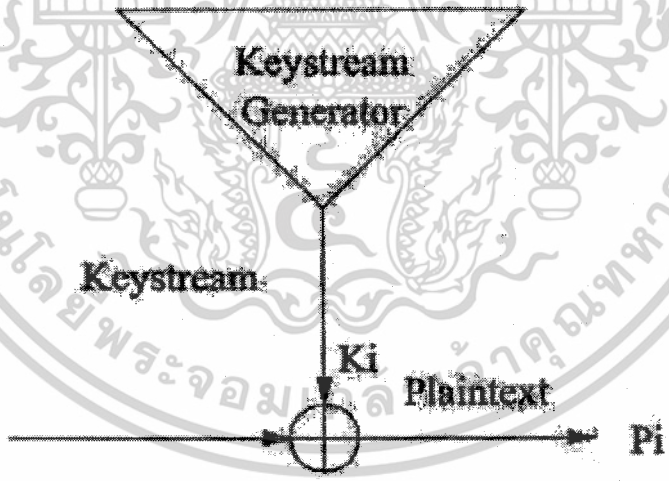
ในขณะที่ one – time pad ถูกใช้ในระหว่างสงครามผ่านช่องทางสื่อสารที่ต้องการความปลอดภัยที่สูงมาก ความจริงที่ว่ากุญแจที่เป็นความลับ (ซึ่งสามารถถูกใช้เพียงครั้งเดียว) ยาวเท่ากับข้อความได้นำเข้าสู่ปัญหาการจัดการกุญแจ (key – management) ที่รุนแรง ในขณะที่ความปลอดภัยสมบูรณ์แบบนั้น one – time pad จะทำได้ไม่จริง

Stream cipher ถูกพัฒนามีค่าใกล้เคียงกับการกระทำของ one – time pad และ stream cipher ที่ไม่สามารถให้ความปลอดภัยในทางทฤษฎีที่พอเพียงของ one – time pad แต่ก็อย่างน้อยได้ผลในทางปฏิบัติ



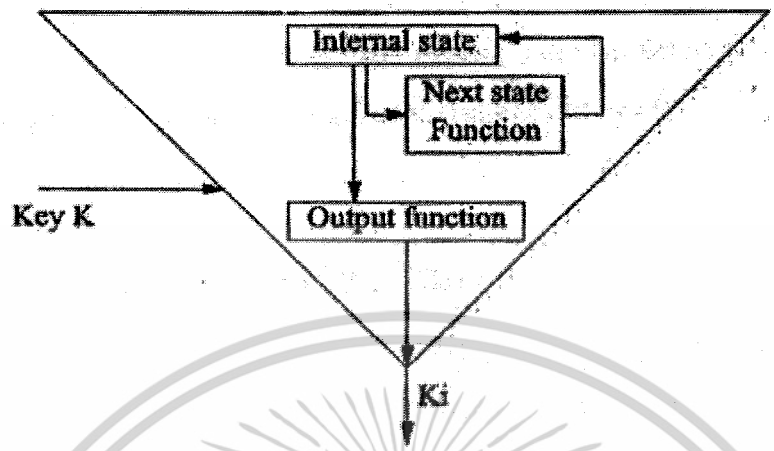
Encrypt

รูปที่ 2.14 การเข้ารหัสของ stream cipher



Decrypt

รูปที่ 2.15 การถอดรหัสของ stream cipher

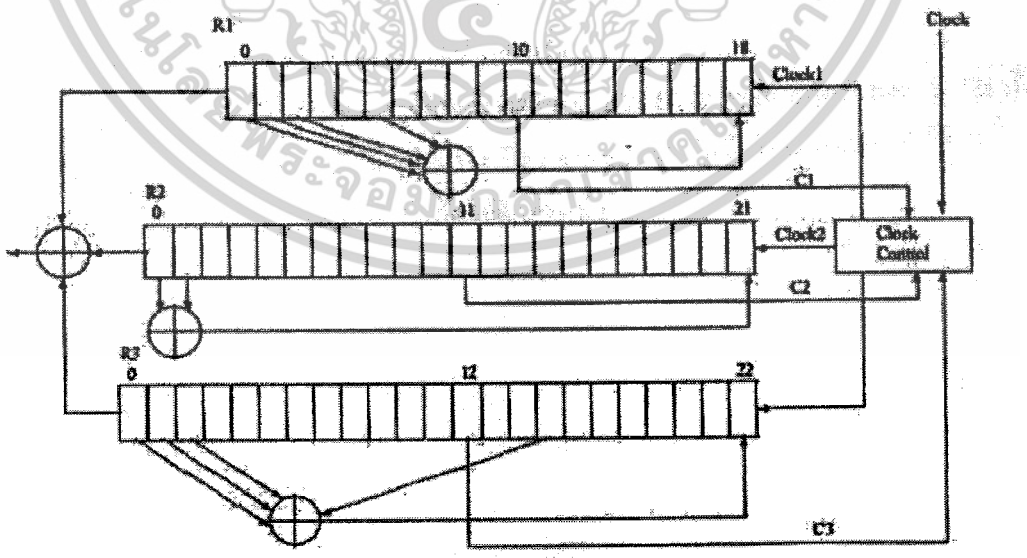


รูปที่ 2.16 การสร้าง keystream

2.2.3.3 A5/1

A5/1 เป็นอัลกอริทึมที่ใช้ในการเข้ารหัสแบบ stream cipher ที่ใช้ในระบบจีเอสเอ็ม โดยข้อมูลในรูปของเสียงจะถูกเข้ารหัสระหว่างสถานีฐานและอุปกรณ์มือถือโดยข้อมูลจะถูกแบ่งเป็น 114 บิตเฟรม

โครงสร้างของ A5/1



รูปที่ 2.17 โครงสร้างของ A5/1

โครงสร้างของ A5/1 ประกอบด้วยรีจิสเตอร์ LFSR จำนวน 3 ตัว คือ R1, R2, R3 ซึ่งมีขนาด 19, 22 , 23 บิต ตามลำดับ การรวมกันของบิตใช้วิธีการ XOR การเลื่อน (shift) ในแต่ละครั้งนั้น จะมีสัญญาณนาฬิกามาควบคุม ซึ่งสัญญาณดังกล่าวได้มาจากบิตกลางในแต่ละรีจิสเตอร์คือ บิตที่ 10 ใน R1, บิตที่ 11 ใน R2 และบิตที่ 12 ใน R3

clock1 = clock ((C1(C2 C3)) - (C1 (C2 C3)))

clock2 = clock ((C2(C1 C3)) - (C2 (C1 C3)))

clock3 = clock ((C3(C1 C2)) - (C3 (C1 C2)))

นอกจากนี้ บิตที่ 18 ของ R1 เท่ากับ R1[0] $\oplus$ R1[1] $\oplus$ R![2] $\oplus$ R1[5]

บิตที่ 21 ของ R1 เท่ากับ R2[0] $\oplus$ R2[1]

บิตที่ 18 ของ R1 เท่ากับ R3[0] $\oplus$ R3[1] $\oplus$ R3[2] $\oplus$ R3[15]

ขั้นตอนการเข้ารหัส A5/1

A5/1 เป็นอัลกอริทึมในการสร้าง pseudo-random ซึ่งใช้คีย์ $K_c \in \{0,1\}^{64}$ และเฟรมบิต $Fn \in \{0,1\}^{22}$ มีขั้นตอนการสร้างคีย์ดังต่อไปนี้

1. กำหนดค่า K_c ขนาด 64 บิต เป็นกุญแจลับ และเฟรมบิต F_n 22 บิต มา setup รีจิสเตอร์ทั้ง 3 ตัว
2. ทำการโหลดค่า K_c ลงรีจิสเตอร์แต่ละตัว โดยทำการshiftไปทีละบิตจนครบ 64 ครั้ง
3. โหลดหาค่า F_n ลงรีจิสเตอร์แต่ละตัว โดยทำการshiftไปทีละบิตจนครบ 22 ครั้ง
4. ทำการมิกซ์ค่า K_c และ F_n ด้วยการshift 100 ครั้ง
5. ผลลัพธ์ของคีย์ที่สร้างขึ้น คือ ค่าที่ออกมาจาก R1 R2 และ R3 บวกกัน จนกระทั่งครบ 228 บิต แล้วจึงนำไปใช้ โดยแบ่งด้านละ 114 บิต (จากอุปกรณ์มือถือไปยังสถานีส่งใช้ค่าหนึ่งและจากสถานีส่งไปยังอุปกรณ์มือถือใช้อีกค่าหนึ่ง

2.3 โครงสร้างของภาษา C

ภาษา C เป็นภาษาคอมพิวเตอร์ชนิดคอมไพล์ (Compiled Language) ซึ่งมีคอมไพลเลอร์ (Compiler) ทำหน้าที่ในการคอมไพล์ (Compile) หรือแปลงคำสั่งทั้งหมดในโปรแกรมให้เป็นภาษาเครื่อง (Machine Language) เพื่อให้เครื่องคอมพิวเตอร์นำคำสั่งเหล่านั้นไปทำงานต่อไป

2.3.1 โครงสร้างของภาษา C

ทุกโปรแกรมของภาษา C มีโครงสร้างเป็นลักษณะ

ส่วนเฮดเดอร์ไฟล์
ส่วนตัวแปรแบบ Global

int main (void)
{
ส่วนตัวแปร Local
ตัวโปรแกรม
ค่าส่งกลับ
}

เฮดเดอร์ไฟล์ (Header Files)

เป็นส่วนที่เก็บไลบรารีมาตรฐานของภาษา C ซึ่งจะถูกดึงเข้ามารวมกับโปรแกรมในขณะที่กำลังทำการคอมไพล์ โดยใช้คำสั่ง

#include<ชื่อเฮดเดอร์ไฟล์> หรือ

#include “ชื่อเฮดเดอร์ไฟล์”

ตัวอย่าง

#include<stdio.h>

เฮดเดอร์ไฟล์นี้จะมีส่วนขยายเป็น .h เสมอ และเฮดเดอร์ไผล์เป็นส่วนที่จำเป็นต้องมีอย่างน้อย 1 เฮดเดอร์ไฟล์ ก็คือ เฮดเดอร์ไฟล์ stsio.h ซึ่งจะเป็นที่เก็บไลบรารีมาตรฐานทีจัดการเกี่ยวกับ Input และ Output

ส่วนตัวแปรแบบ Global (Global Variables)

เป็นส่วนที่ใช่ประกาศตัวแปรหรือค่าต่าง ๆ ที่ให้ใช้ได้ทั้งโปรแกรม ซึ่งในส่วนไม่จำเป็นต้องมีก็ได้

ฟังก์ชัน (Functions)

เป็นส่วนที่เห็บคำสั่งต่าง ๆ ไว้ ซึ่งในภาา C จะบังคับให้มีอย่างน้อย 1 ฟังก์ชันนั่นก็คือ ฟังก์ชัน Main( ) และในโปรแกรม 1 โปรแกรมสามารถมีฟังก์ชันได้มากกว่า 1 ฟังก์ชัน

ส่วนตัวแปรแบบ Local (Local Variables)

เป็นส่วนที่ใช่สำหรับประกาศตัวแปรที่จะใช้ในเฉพาะฟังก์ชันของตนเอง ฟังก์ชันอื่นไม่สามารถเข้าถึงหรือใช้ได้ ซึ่งจะต้องทำหารประกาศตัวแปรก่อนการใช้งานเสมอ และจะต้องประกาศไว้ในส่วนนี้เท่านั้น

ตัวโปรแกรม (Statements)

เป็นส่วนที่อยู่ถัดลงมาจากส่วนตัวแปรภายใน ซึ่งประกอบด้วยคำสั่งต่าง ๆ ของภาษา C และคำสั่งต่าง ๆ จะใช้เครื่องหมาย ว เพื่อเป็นการบอกให้รู้ว่าจบคำสั่งหนึ่ง ๆ แล้ว ส่วนใหญ่คำสั่งต่าง ๆ ของภาษา C เขียนด้วยตัวพิมพ์เล็ก เนื่องจากภาษา C จะแยกความแตกต่างของตัวพิมพ์เล็กและพิมพ์ใหญ่ หรือ Case Sensitive นั่นเอง ยกตัวอย่างใช้ Files, files หรือ FILES จะถือว่าเป็นตัวแปรคนละตัวกัน นอกจากนี้ภาษา C ยังไม่สนใจกับการขึ้นบรรทัดใหม่ เพราะฉะนั้นผู้ใช้สามารถพิมพ์คำสั่งหลายคำสั่งในบรรทัดเดียวกันได้ โดยมีเครื่องหมาย ; เป็นตัวจบคำสั่ง

ค่าส่งกลับ (Return Value)

เป็นส่วนที่บอกให้รู้ว่า ฟังก์ชันนี้จะส่งค่าอะไรกลับไปให้กับฟังก์ชันที่เรียกฟังก์ชันนี้ ซึ่งเรื่องนี้ผู้เขียนจะยกไปกล่าวในเรื่องฟังก์ชันอย่างละเอียดอีกที

หมายเหตุ (Comment)

เป็นส่วนที่ใช้สำหรับแสดงข้อความเพื่ออธิบายสิ่งที่ต้องการในโปรแกรม ซึ่งจะใช้เครื่องหมาย/* และ*/ ปิดหัวและปิดท้ายของข้อความที่ต้องการ

แสดงการเขียนหมายเหตุหรือ Comment ในลักษณะต่าง ๆ

```
/*นี้คือ Comment*/
/*ถ้ามี 2 บรรทัด
ให้ทำแบบนี้              */
ถ้ามีหลายบรรทัดก็สามารถ
ทำแบบนี้ได้เช่นเดียวกัน
*/
```

2.3.2 การตั้งชื่อ

การตั้งชื่อ (Identifier) ให้กับตัวแปร ฟังก์ชันหรืออื่น ๆ มีกฎเกณฑ์ในการตั้งชื่อ ดังนี้

1. ตัวแรกของชื่อจะต้องขึ้นด้วยตัวอักษาหรือเครื่องหมาย_เท่านั้น
2. ตัวอักษรตั้งแต่ตัวที่ 2 สามารถเป็นตัวเลข หรือเครื่องหมาย_ก็ได้
3. จะต้องไม่มีการเว้นวรรคภายในชื่อ แต่สามารถใช้เครื่องหมาย_คั่นได้
4. สามารถตั้งชื่อได้ยาวไม่จำกัด แต่จะให้ตัวอักษรอ้างอิงแค่ 31 ตัวแรกในการอ้างอิง
5. ชื่อที่ตั้งด้วยอักษรพิมพ์ใหญ่และพิมพ์เล็ก จะถือว่าเป็นคนละตัวกัน
6. ห้ามตั้งชื่อซ้ำกับคำสงวนในภาษา C

ตัวอย่างการตั้งที่ถูกและผิด

แบบที่ถูก	แบบที่ผิด
A	$sum
Student_name	Student Name
_SystemName	2names
A1	int

2.3.3 ชนิดข้อมูล

ในการเขียนโปรแกรมภาษา C นั้น ผู้ใช้จะต้องทำหนดชนิดให้กับตัวแปรนั้นก่อนที่จะนำไปใช้งาน โดยผู้ใช้จะต้องรู้ว่าในภาษา C นั้นจะมี 4 ชนิดข้อมูลมาตรฐาน คือ

ชนิดข้อมูลแบบไม่มีค่า หรือ Void Type (void)

ข้อมูลชนิดนี้ จะไม่มีค่า และจะไม่ใช้ในการกำหนดชนิดของตัวแปร แต่ส่วนใหญ่จะใช้เกี่ยวกับฟังก์ชัน

ชนิดข้อมูลแบบจำนวนเต็ม หรือ Integer Type (int)

เป็นชนิดข้อมูลที่เป็นตัวเลขจำนวนเต็ม ไม่มีทศนิยม ซึ่งภาษา C จะแบ่งข้อมูลชนิดนี้ออกได้เป็น 3 ระดับ คือ short int, int และ long int ซึ่งแต่ละระดับนั้นจะมีขอบเขตการใช้งานที่แตกต่างกัน

แสดงรายละเอียดของชนิดข้อมูลแบบจำนวนเต็ม

ชนิดข้อมูล	คิดเรื่องหมาย	ขนาด(ไบต์)	จำนวนบิต	ค่าน้อยที่สุด	ค่ามากที่สุด
Short int	คิด	2	16	-32,768	32,768
	ไม่คิด			0	65,535
int	คิด	2	16	-32,768	32,768
(16บิต)	ไม่คิด			0	65,535
int	คิด	4	32	-2,147,486,643	2,147,486,643
(32บิต)	ไม่คิด			0	4,294,967,295
long int	คิด	4	32	-2,147,486,643	2,147,486,643
	ไม่คิด			0	4,294,967,295

ชนิดข้อมูลแบบตัวอักษร หรือ Character Type (char)

ข้อมูลชนิดนี้ก็คือ ตัวอักษรตั้งแต่ A-Z เลข 0-9 และสัญลักษณ์ต่าง ๆ ตามมาตรฐาน ASCII (American Standard Code for Information Interchange) ซึ่งเมื่อกำหนอให้กับตัวแปรแล้วตัวแปรตัวนั้นจะรับค่าได้เพียง 1 ตัวอักษรเท่านั้น และสามารถรับข้อมูลจำนวนเต็มตั้งแต่ 128 ถึง 127 จะใช้ขนาดหน่วยความจำ 1 ไบต์หรือ 8 บิต

ชนิดข้อมูลแบบทศนิยม หรือ Floating Point Type (float)

เป็นข้อมูลชนิดตัวเลขที่มีจุดทศนิยม ซึ่งสามารถแบ่งออกเป็น 3 ระดับ คือ float, double c]t long double แต่ละระดับนั้นจะมีขอบเขตที่แตกต่างกันในการใช้งาน

ชนิดข้อมูล	ขนาด (ไบต์)	จำนวนบิต	ค่าน้อยที่สุด
Float	4	32	3.4*10-38ถึง3.4*1038
Double	8	64	1.7*10-308ถึง1.7*10308
Long double	10	80	3.4*10-4932ถึง1.2*104932

2.3.4 ตัวแปร

ตัวแปร คือ ชื่อที่ใช้อ้างถึงตำแหน่งต่าง ๆ ในหน่วยความจำ ซึ่งใช้เก็บข้อมูลต่าง ๆ ด้วยขนาดตามชนิดของข้อมูล

การประกาศตัวแปร

การประกาศตัวแปรในภาษา C นั้นสามารถทำได้ 2 ลักษณะ คือ การประกาศตัวแปรแบบเอกภาพหรือการประกาศตัวแปรแบบ Global คือ ตัวแปรที่จะสามารถเรียกใช้ได้ทั้งโปรแกรม และแบบที่สองการประกาศตัวแปรแบบภายใน หรือการประกาศตัวแปรแบบ Local ซึ่งตัวแปรประเภทนั้นจะใช้ได้ในเฉพาะฟังก์ชันของตัวเองเท่านั้น

การกำหนดค่าให้กับตัวแปร

การกำหนดค่าให้กับตัวแปรนั้น จะสามารถกำหนดได้ตั้งแต่ตอนที่ประกาศตัวแปรเลยหรือจะกำหนดภายในโปรแกรมก็ได้ ซึ่งการกำหนดค่าจะใช้เครื่องหมาย = กั้นตรงกลาง

```
int total = 0;
```

ถ้ามีตัวแปรข้อมูลชนิดเดียวกัน ก็สามารถทำแบบนี้ได้

```
int total = 0,sum;
```

หรือ

```
int total =0,sum = 0;
```

ถ้าเป็นการกำหนดภายในโปรแกรม ซึ่งตัวแปรนั้นได้ประกาศไว้แล้วสามารถทำแบบนี้ได้

```
total = 50;
```

หรือ

```
total = total+sum;
```

หรือกำหรดค่าจากการพิมพ์ข้อมูลเข้าทางคีย์บอร์ด

```
scanf("%d",&total);
```

2.3.5 กำหนดชนิดข้อมูลแบบชั่วคราว

เมื่อผู้ใช้ได้กำหนดชนิดข้อมูลให้กับตัวแปร ใด ๆ ไปแล้ว ตัวแปรตัวนั้นจะมีชนิดข้อมูลเป็นแบบที่กำหนดให้ตลอดไป บางครั้งการเขียนโปรแกรมอาจจะต้องมรความจำเป็นต้องเปลี่ยนชนิดข้อมูลของตัวแปรตัวนั้น ซึ่งภาษา C ก็มีความสามารถที่จะทำเช่นนั้นได้

รูปแบบ

([ชนิดข้อมูล]) [ตัวแปร]

ตัวอย่าง

```
(float)a
(int)a
```

2.3.6 ชนิดข้อมูลแบบค่าคงที่ (Constants)

ชนิดข้อมูลประเภทนี้ เป็นชนิดแบบค่าคงที่ ซึ่งก็คือข้อมูลของตัวแปรประเภทที่เป็น Constants ผู้ใช้จะไม่สามารถเปลี่ยนแปลงค่าของตัวแปรนั้น ในขณะที่โปรแกรมทำงานอยู่

รูปแบบ

Const [ชนิดข้อมูล] [ตัวแปร] = [ค่า หรือ นิพจน์]

ตัวอย่าง

```
const float a = 5.23;
const int b = a%2;
```

Constant นั้นสามารถแบ่งออกได้ ดังนี้

Integer Constants เป็นค่าคงที่ชนิดข้อมูลแบบตัวเลขจำนวนเต็ม ไม่มีจุดทศนิยม

Fioating-Pount Constants เป็นค่าคงที่ชนิดข้อมูลแบบตังเลขที่มีจุดทศนิยม

Character Constants เป็นค่าคงที่ชนิดตัวอักษร ซึ่งจะต้องอยู่ภายในเครื่องหมายเท่านั้น

String Constants เป็นค่าคงที่เป็นข้อความ ซึ่งจะต้องอยู่ภายใต้เครื่องหมาย “”เท่านั้น “”

2.3.7 Statements

Statements ในภาษา C ก็คือ คำสั่งต่าง ๆ ที่ประกอยขึ้นจนเป็นตัวโปรแกรม ซึ่งในภาษา C นั้นแบ่งออกเป็น 6 แบบ แต่จะพูดถึง 2 แบบแรกก่อน คือ Expression Statement และ Compound Statement

Statement แบบต่าง ๆ ของภาษา C

Statement	Expression Statement
	Compound Statement
	Labeled Statement
	Selection Statement
	Iterative Statement
	Jump Statement

Expression Statement หรือเรียกอีกอย่างได้ว่า Single Statement : ซึ่ง Statement แบบนี้จะต้องมีเครื่องหมาย ; หลังจาก Statement เมื่อภาษา พบเครื่องหมาย ; จะทำให้มันรู้ว่าจบชุดคำสั่งแล้ว แล้วจึงข้ามไปทำ Statement ชุดต่อไป

Compound Statement คือ ชุดคำสั่งที่มีคำสั่งต่าง ๆ รวมอยู่ด้านใน Block ซึ่งจะใช้เครื่องหมาย { เป็นการเปิดชุดคำสั่ง และใช้ } เป็นตัวปิดชุดคำสั่ง ตัวอย่างที่เห็นได้ชัดสำหรับ Statement แบบนี้ คือ ตัวฟังก์ชัน Main โดยทั่ว ๆ ไปในภาษา C Compound Statement จะเป็นตัวฟังก์ชัน

ผังงาน

ผังงาน (Flowchart) นั้นมีไว้เพื่อให้ผู้ใช้ออกแบบขั้นตอนการทำงานของโปรแกรมก่อนที่จะลงมือเขียนโปรแกรม ซึ่งจะช่วยให้ผู้ใช้เขียนโปรแกรมได้งานยขึ้นและไม่สับสนซึ่งผังงานที่นิยมใช้มีมาตรฐามมากมายหลายแบบแต่ในที่นี้จะขออธิบายมาตรฐาน ANSI (American National Standard)

2.4 องค์ประกอบของภาพ

2.4.1รูปร่างของภาพ (Image Shape)

วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่นิยมใช้กันมากที่สุด เนื่องจากทำ ให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำ และการแสดงภาพออกทางอุปกรณ์ต่าง ๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงหน่วยความจำ ของคอมพิวเตอร์สามารถทำ ได้โดยการจองหน่วยความจำ ของเครื่องไว้ในรูปของตัวแปรอะเรย์ (array) โดยค่าในแต่ละช่องของอะเรย์แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำ แหน่งของช่องอะเรย์เป็นตัวกำ หนดตำ แหน่งของจุดภาพสมมุติให้ Image เป็นตัวแปรแบบอะเรย์ขนาด M X N (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด M x N จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่เป็นภาพ grey level) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image(5,4) จะเห็นว่าเราใช้ตำ แหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอะเรย์ จากการใช้หน่วยความจำ เพื่อการเก็บภาพในลักษณะที่กล่าวมา เนื้อที่ในการเก็บภาพสามารถคำ นวณได้จาก M x N x g เมื่อ g เป็นจำนวนเต็มที่แทนจำ นวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต

เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำ หรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640x480, 800x600 และ1024x768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ ในงานบางอย่างใช้ความละเอียดแค่ 30 x50 จุด ก็พอแล้วแต่ในงานบางชนิด ใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำ หรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำ ภาพนี้ไปแสดงในจอภาพมาตรฐานจะทำ ให้ภาพที่แสดงนั้นมีขนาดของจุดภาพไม่เป็นสีเหลี่ยมจัตุรัสเช่นในบางระบบอาจจะใช้ความละเอียดในการแสดงเท่ากับ 640 x 512 ซึ่งจะทำ ให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำ หรับการเขียนโปรแกรมทางด้านกราฟฟิก

และการจัดการข้อมูลจำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้เมื่อมีการกำหนดให้ขนาดของบิตต่อจุดมากขึ้นจะทำ ให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 บิต = 21=4 สี

2 บิต = 22=4 สี

4 บิต = 24=16 สี

8 บิต = 28=256 สี

16 บิต = 216=65536 สี เป็นต้น

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำ งานที่จะใกล้เคียงกันเนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยว ๆ ได้ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำ การก็อปปี้ข้อมูลทั้ง 8 บิต (1 byte) ส่งให้กับจอภาพซึ่งในกรณีที่ Pixel มีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำ การก็อปปี้ข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณี Pixel ที่มีขนาด 8 บิต โปรเซสเซอร์จะทำ การก็อปปี้ข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำ งานกับทุกบิตแล้วตัวอย่างสำ หรับระบบที่มีความละเอียดเท่ากับ 800x600 และมีขนาด 16 บิตต่อ Pixel จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 800x600x16 บิต

2.4.2 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำ ไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสี คือ แกนสีแดง เขียว และ น้ำเงินในระบบ HLS จะมีแกนเป็น ค่าสี (hue) ความสว่าง (lightness) และความบริสุทธิ์ของสี (saturation) ตัวอย่างระบบสีที่นิยมใช้กันได้แก่ ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

ระบบสี RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำ ไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสีRGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แต่ RGBCIE และ RGBNTSC ระบบสีแบบ RGB ของ CIE เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm ระบบสีแบบ RGB ของ NTSC เป็นระบบที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำ หรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสำ หรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

2.4.3 ความหมายของสกุล GIF, TIFF, JPEG, PDF

ไฟล์กราฟิกแบ่งเป็นหลายรูปแบบ แต่ที่นิยมใช้กันมากในงานกราฟิกสำหรับเว็บไซต์ เช่น GIF และ JPEG สำหรับงานพิมพ์ เช่น TIFF, EPS และ PDF

2.4.3.1 GIF (Graphic Interchange Format)

รูปแบบไฟล์ GIF ได้รับการออกแบบโดย CompuServe ซึ่งเป็นระบบเครือข่ายข่าวสารแบบออนไลน์ เพื่อให้บริการแลกเปลี่ยนกราฟิกในรูปแบบ bitmap ที่มีการจัดการทางด้านหน่วยความจำที่มีประสิทธิภาพ ข้อจำกัดของภาพแบบ GIF คือ ความสามารถทางด้านสีซึ่งเป็นแผงสีแบบอินเด็กซ์ (ภาพสีแบบ 24 บิตไม่สามารถใช้ได้) แผงสีสามารถบรรจุได้ 2 ถึง 256 สี ซึ่งถูกสร้างจากข้อมูลสี 24 บิต ไฟล์แบบ GIF ถูกบีบขนาดโดยใช้การบีบขนาด LZW แบบประยุกต์ การขยายไฟล์ข้อมูลแบบ GIF กลับคืน จะช้ากว่าการบีบขนาดแบบ RLE แต่จะเปลืองเนื้อที่หน่วยความจำน้อยกว่ารูปแบบไฟล์ GIF เป็นภาพซึ่งใช้สีจำกัด (ไม่เกิน 256 สี ไม่ใช้ทั้งหมดของสเปกตรัมสีที่แสดงได้บนมอนิเตอร์) เหมาะสำหรับภาพที่ต้องการไฟล์ขนาดเล็ก โหลดเร็ว ไฟล์แบบนี้จึงเหมาะกับงานที่ใช้สีแบบ solid color เช่น โลโก้ หรือ ภาพแบบ Illustration Graphic Interchange Format นามสกุลที่ใช้เก็บ GIF ระบบปฏิบัติการ Windows, Windows NT เวอร์ชันที่ได้รับการพัฒนาจนถึงปัจจุบัน 87a และ 89a ซอฟต์แวร์ที่สร้างและเปิดไฟล์ โปรแกรมการแก้ไข bitmap ทุกโปรแกรม, โปรแกรม Desktop Publishing เช่น PhotoShop, CorelDRAW, PaintShop Pro, ACDSee 32 ความสามารถทางด้านสี แผงสีแบบอินเด็กซ์ถึง 256 สี (วาดจากสี RGB แบบ 24 บิต) การบีบขนาดข้อมูล LZW การใส่รหัสแบบ run-length

2.4.3.2 JPEG (Joint Photographic Experts Group)

มาตรฐานการบีบขนาดแบบ JPEG ไม่ได้ถูกออกแบบมาเพื่อฮาร์ดแวร์หรือซอฟต์แวร์ใดๆ โดยเฉพาะ แต่ได้นำเสนอวิธีการบีบขนาดที่สามารถใช้ทั่วๆ ไปหลายวิธี ดังนั้นจึงมีการบีบขนาดหลายวิธีที่เกิดขึ้นมาโดยใช้มาตรฐานการบีบขนาดแบบ JPEG การบีบขนาดด้วยวิธีนี้ช่วยลดขนาดของภาพกราฟิกและประหยัดเวลาในการโหลดได้มาก เหลือเพียงหนึ่งในสิบของภาพเดิม และบางครั้งสามารถลดขนาดลงได้มากถึง 100 ต่อ 1 JPEG เป็นไฟล์ที่เหมาะสำหรับใช้ในภาพประเภทภาพถ่าย (โทนสีต่อเนื่อง) เนื่องจากใช้สีทั้งสเปกตรัมสีที่มีในมอนิเตอร์ และเป็นไฟล์ประเภทที่ถูกบีบอัดให้เล็กลงเพื่อให้โหลดเร็วขึ้นเช่นเดียวกับ GIF โดยการตัดค่าสีในช่วงที่ตามองไม่เห็นทิ้งไป แต่เมื่อบันทึกไฟล์เป็น JPEG แล้ว ข้อมูลสีที่ถูกตัดทิ้งไปจะไม่สามารถเรียกกลับมาได้อีก ถ้าต้องการใช้ค่าสีเหล่านั้นในอนาคต ควรจะบันทึกเป็นไฟล์ชนิดอื่น แล้วเปลี่ยนเป็นไฟล์ JPEG ด้วยการบันทึกเป็นไฟล์ก็อปปี้ Joint Photographic Experts Group นามสกุลที่ใช้เก็บ JPG หรือ JIF (JPG + TIFF) ระบบปฏิบัติการ Windows ซอฟต์แวร์ที่สร้างและเปิดไฟล์ โปรแกรมการแก้ไขภาพ Bitmap และโปรแกรมการแปลงรูปแบบ เช่น PhotoShop, CorelDRAW, PaintShop Pro, ACDSee 32 ความสามารถทางด้านสี 2, 16, 256 สี หรือ 16 ล้านสี และความลึกสีแบบ 32 บิต

2.4.3.3 TIFF (Tagged Image File Format)

TIFF เป็นไฟล์ที่ใช้ได้กับ bitmap เท่านั้น พัฒนาขึ้นโดยความร่วมของ Aldus Corporation และ Microsoft TIFF เก็บบันทึกข้อมูลรูปภาพได้หลากหลายใน Tagged Field จึงกลายเป็นชื่อเรียกของรูปแบบไฟล์ ซึ่งแต่ละ Tagged Field สามารถบันทึกข้อมูลเกี่ยวกับ bitmap หรือชี้ไปยัง Field อื่นได้ ซอฟต์แวร์ที่อ่านไฟล์นี้สามารถข้ามการอ่าน Field ที่ไม่เข้าใจหรือไม่จำเป็นไปได้ TIFF เป็นรูปแบบที่มีความยืดหยุ่น สามารถเปลี่ยนแปลงแก้ไขได้ เนื่องจากมี Tagged Field ให้ใช้ต่างกันหลายร้อยชนิด ไฟล์แบบนี้จึงมีข้อดี คือ ใช้ได้กับโปรแกรมกราฟิกทุกประเภท สามารถใช้ได้ในระบบคอมพิวเตอร์หลายๆ ระบบ และกำหนดขอบเขตที่กว้างขวางของภาพ bitmap ได้ นอกจากนี้ TIFF ยังสามารถทำบางสิ่งที่ bitmap อื่นทำไม่ได้ และเป็นรูปแบบที่สนับสนุนทั้งระบบ PC และ Macintosh Tagged Image File Format นามสกุลที่ใช้เก็บ TIF ระบบปฏิบัติการ Windows, UNIX, Mac Windows เวอร์ชันที่ได้รับการพัฒนาจนถึงปัจจุบัน 5.0 และ 6.0 ซอฟต์แวร์ที่สร้างและเปิดไฟล์ โปรแกรมแก้ไข Bitmap และโปรแกรม Desktop Publishing เช่น PageMaker, QuarkXPress, CorelVentura, PhotoShop, PaintShop Pro ความสามารถทางด้านสี ขาวดำ 1 บิต, Grayscale (4,8, 16 บิต), แผงสี (ได้ถึง 16 บิต), สี RGB (ได้ถึง 48 บิต), สี CMYK (ได้ถึง 32 บิต) การบีบขนาดข้อมูล LZW, PackBits (Macintosh), JPEG (TIFF v 6.0), RLE หลายรูปแบบ

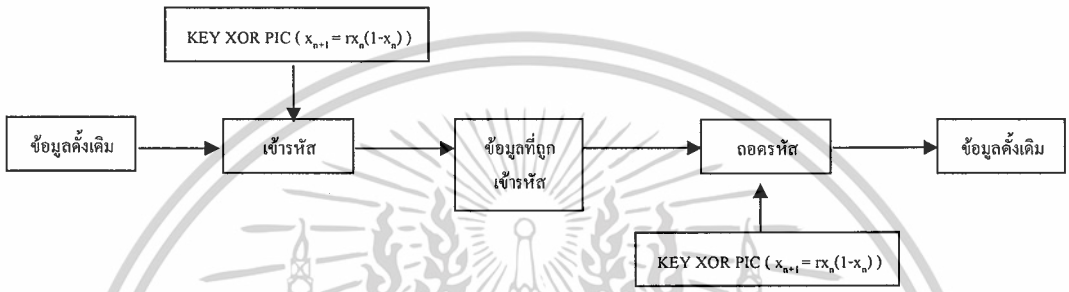
2.4.3.4PDF (Portable Document Format)

PDF เป็นรูปแบบไฟล์ที่ใช้ในโปรแกรม Adobe Acrobat ใช้สำหรับเอกสารบนสื่ออิเล็คทรอนิกส์ เช่น บนอินเทอร์เน็ตหรือบริการออนไลน์ต่างๆ เนื่องจากเป็นไฟล์ขนาดเล็กทำให้สามารถสร้างเอกสาร เช่น โบร์ชัวร์ หรือ แค็ทตาล็อกส่งไปทางอินเทอร์เน็ตได้ ใช้ได้กับทั้งแบบ Bitmap และ Vector และสนับสนุนทั้งระบบ PC และ Macintosh PDF เหมาะสำหรับเอกสารทางเทคนิคที่จะเผยแพร่บนอินเทอร์เน็ต ผู้อ่านสามารถพิมพ์ออกมาได้หรือเรียกดูได้โดยไม่เสียค่าใช้จ่ายเพราะรูปแบบอักษรที่ใช้ประกอบอยู่ในตัวซอฟต์แวร์แล้ว และเนื่องจากใช้ตัวอักษรแบบ PostScript ซึ่งเป็น vector-based จึงสามารถย่อและขยายได้ตามต้องการ โดยคุณภาพของงานไม่เปลี่ยนแปลง ทั้งยังสามารถนำไปสร้างเป็นเอกสาร แบบ Illustration หรือ Bitmap ได้อีกด้วย และเมื่อพิมพ์ออกมาก็จะไม่เสียคุณภาพ ไม่ว่าจะใช้ค่าความละเอียดของภาพเป็นเท่าใด เช่นเดียวกับไฟล์ประเภท Vector อื่นๆ เช่น PS หรือ PRN นอกจากนี้ PDF เป็นไฟล์ที่ประกอบด้วยข้อมูล PostScript จึงสามารถนำไปใช้ในโปรแกรมตกแต่งแก้ไขภาพ หรือ โปรแกรมประเภท Illustration ได้เช่นเดียวกับ EPS Portable Document Format นามสกุลที่ใช้เก็บ PDF ระบบปฏิบัติการ Windows, Mac OS, UNIX และ Dos ซอฟต์แวร์ที่สร้างและเปิดไฟล์ PhotoShop, Acrobat ความสามารถทางด้านสี RGB, Indexed-Color, CMYK, GrayScale, Bitmap และ Lap Color

บทที่ 3
การออกแบบโปรแกรมเข้ารหัส

ในปริญญานิพนธ์นี้กล่าวถึง System Flow Diagram ของโปรแกรมเข้ารหัส ในหัวข้อที่ 3.1 เพื่อแสดงวิธีการเข้ารหัสลับ การออกแบบฮาร์ดแวร์ในหัวข้อที่ 3.2 การออกแบบซอร์ฟแวร์ด้วยภาษาซีในหัวข้อที่ 3.3 และการหาค่า parameter r ในสมการอลวนในหัวข้อที่ 3.4 ดังแสดงรายละเอียดดังหัวข้อต่อไปนี้

3.1 System Flow Diagram ของโปรแกรมเข้ารหัส

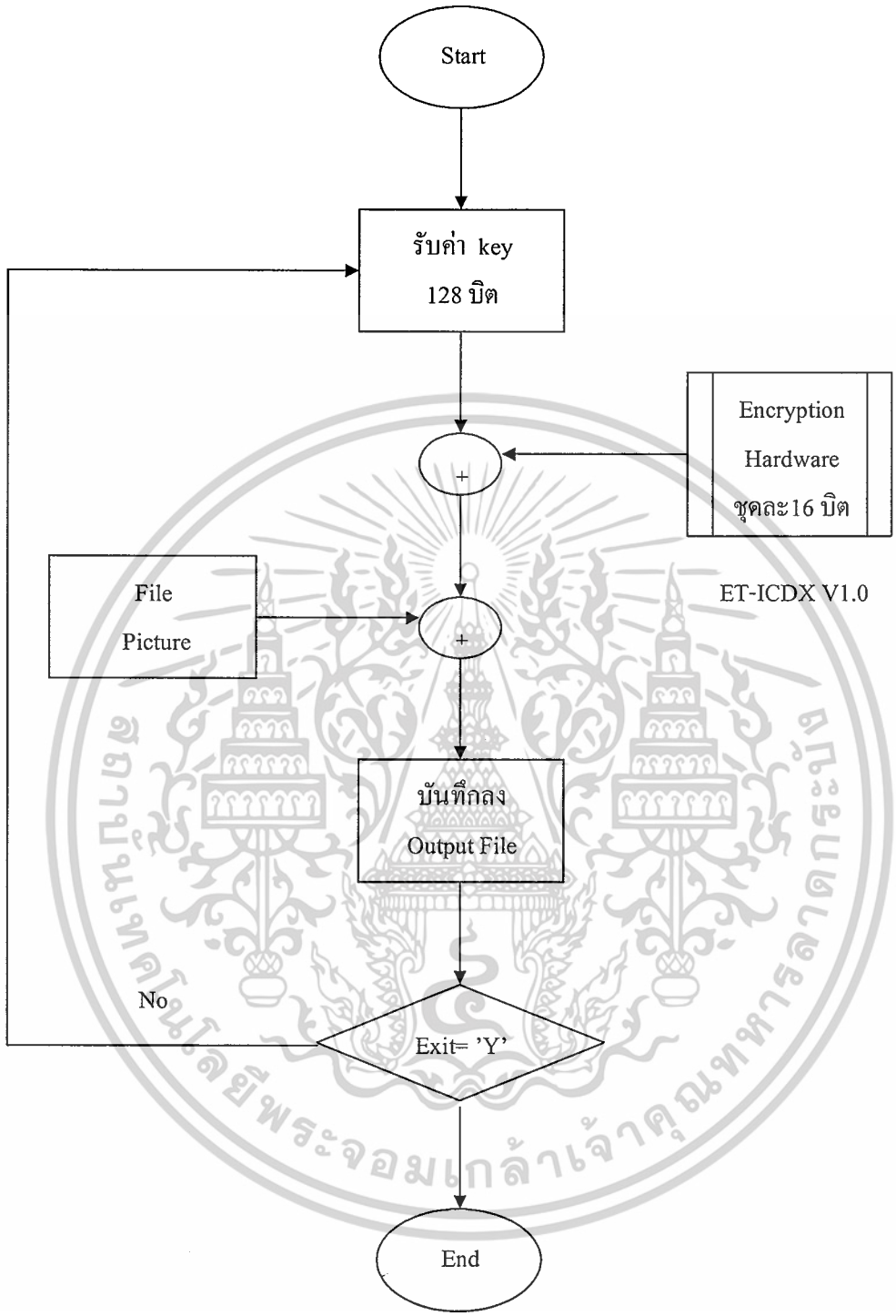


รูปที่ 3.1 System Flow Diagram ของโปรแกรม

จาก System Flow Diagram สามารถอธิบายได้ดังนี้คือ

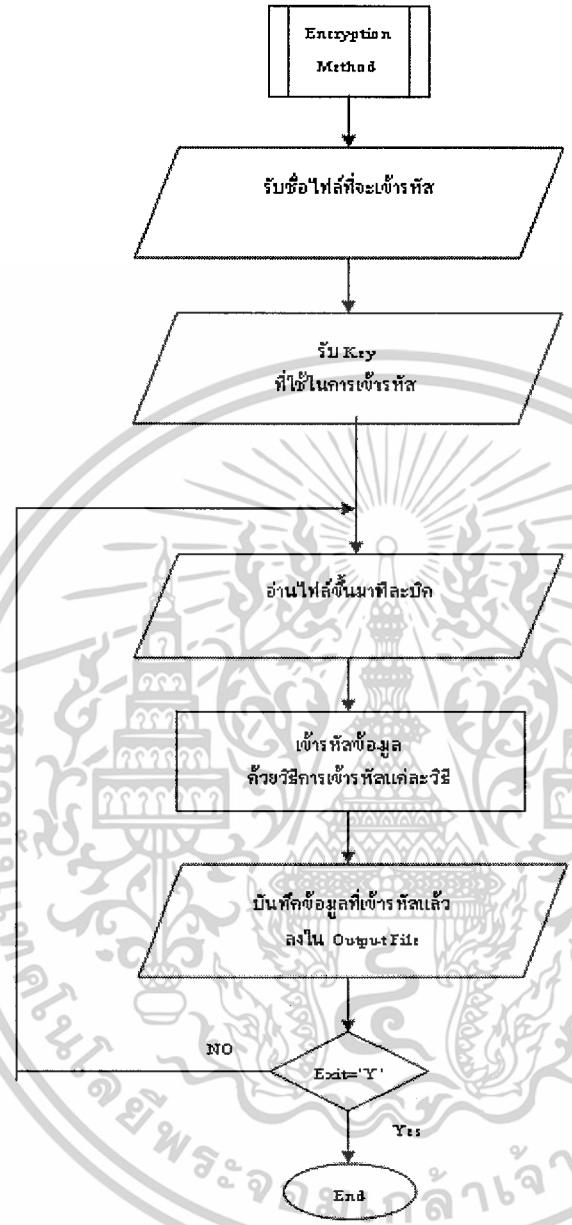
1. ผู้ใช้กำหนดไฟล์ข้อมูล (plaintext) และ key ที่ใช้ในการเข้ารหัส แล้วอ่านข้อมูลไฟล์แบบไบนารี
2. เข้ารหัสไฟล์โดยใช้ key ที่กำหนดให้ จะได้ผลลัพธ์ (ciphertext) ออกมา จากนั้นจึงบันทึกเป็นไฟล์ที่เข้ารหัสแล้ว ซึ่งจะใช้ในการส่งต่อไป
3. ถอดรหัสไฟล์โดยรับ key
4. ผลลัพธ์จากการถอดรหัสจะบันทึกไว้ในไฟล์ข้อมูลเดิม

3.1.1 Flow Chart



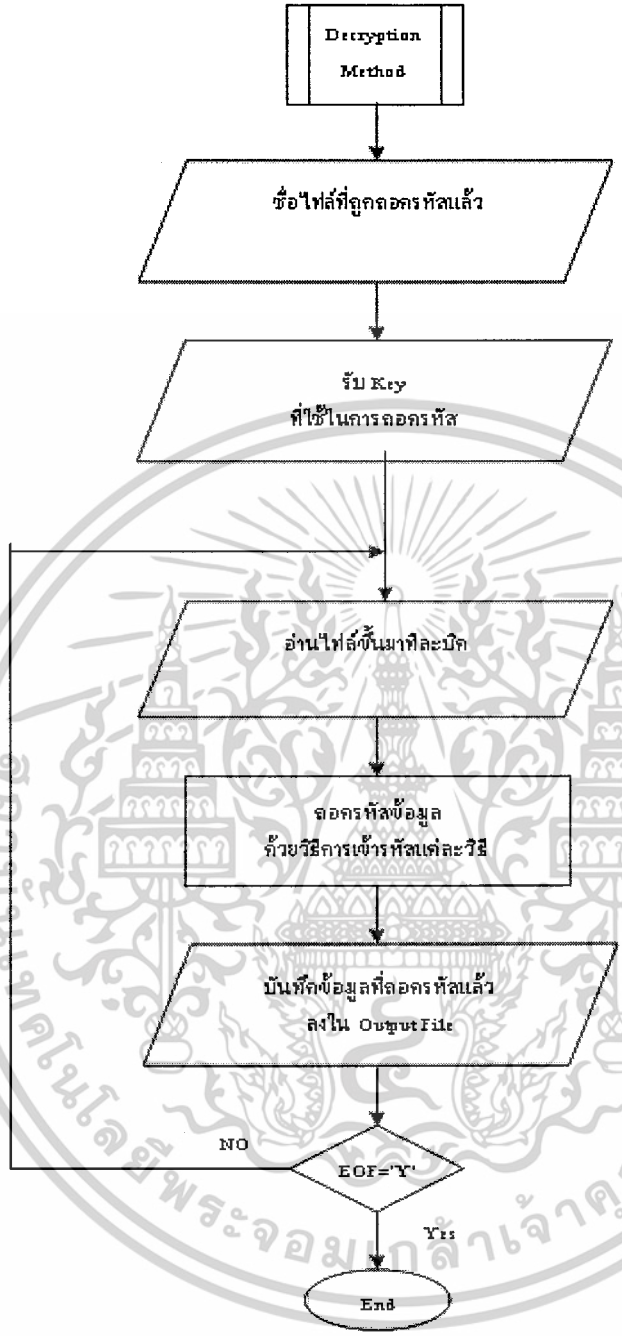
รูปที่ 3.2 Flow Chart ของโปรแกรม

3.1.2 ส่วนการเข้ารหัส



รูปที่ 3.3 Flow Chart ของการเข้ารหัส

3.1.3 ส่วนการถอดรหัส



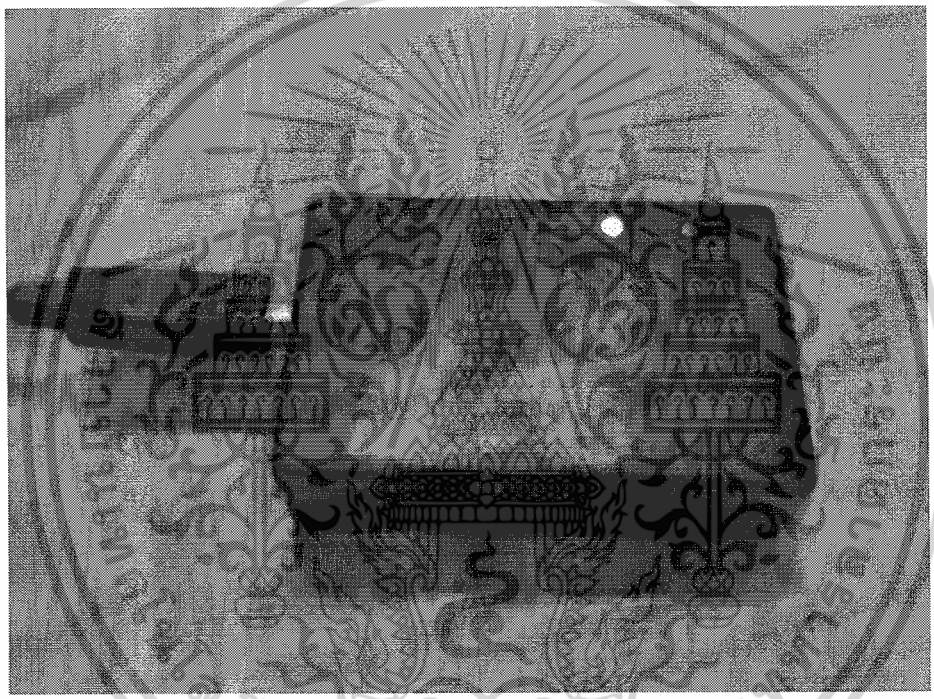
รูปที่ 3.4 Flow Chart ของการถอดรหัส

การออกแบบระบบประกอบด้วย 2 ส่วน คือ

3.2 การออกแบบฮาร์ดแวร์

โครงงานนี้เลือกใช้ไมโครคอนโทรลเลอร์ ET-ICDX V1.0 ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล PIC มาทำหน้าที่เปรียบเสมือนแม่กุญแจโดยการสร้าง Pattern ตามสมการแม็ปลอจิสติก ($x_{n+1} = rx_n(1-x_n)$) ซึ่งตัวแปรต่างๆในสมการจะถูกกำหนดในขั้นตอนต่อไปโดยค่าที่ส่งออกจากไมโครคอนโทรลเลอร์จะไป XOR กับ key ซึ่ง key เปรียบเสมือนลูกกุญแจของกระบวนการเข้ารหัส

ส่วนของฮาร์ดแวร์นี้มีความสำคัญมากต่อกระบวนการเข้ารหัส เพราะนอกจากจะทำหน้าที่เป็นแม่กุญแจแล้ว ยังทำให้การเข้ารหัสมีความแข็งแกร่งมากยิ่งขึ้น



รูปที่ 3.5 ฮาร์ดแวร์ PIC ET-ICDX V1.0

การออกแบบในส่วนคำนวณและประมวลผลในไมโครคอนโทรลเลอร์

ในส่วนของโปรแกรมคำนวณนี้ ทำหน้าที่สร้าง Pattern ตามสมการอลวน แล้วนำผลลัพธ์ที่ได้ส่งไปยังส่วนของโปรแกรมเข้ารหัสลับอลวนผ่านพอร์ท USB โดยในโครงงานนี้ใช้ภาษาซีในการเขียนโปรแกรมลงไมโครคอนโทรลเลอร์ ซึ่งกระบวนการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์ มีดังนี้

1. กำหนดสมการแม็ปลอจิสติก ($x_{n+1} = rx_n(1-x_n)$) โดยกำหนดค่าตัวแปร $x_n = 0.6$ และ $r = 3.9$ โดยเหตุผลที่กำหนดค่าดังกล่าวจะแสดงในส่วนของการหาค่า r ในหัวข้อ 3.4
2. คำนวณสมการแม็ปลอจิสติกจำนวน 100 รอบ
3. แบ่งผลที่ได้จากการคำนวณออกเป็นชุด ชุดละ 16 บิต

4. ส่งผลที่ได้จากข้อ 3 ไปยังโปรแกรมเข้ารหัสลับอลวนผ่านพอร์ต USB โดยส่งชุดค่าที่ 100 ในสมการแม็ปลอจิสติก

การคำนวณค่า x_n ตามสมการแม็ปลอจิสติก ($x_{n+1} = rx_n(1-x_n)$ จำนวน 100 รอบ

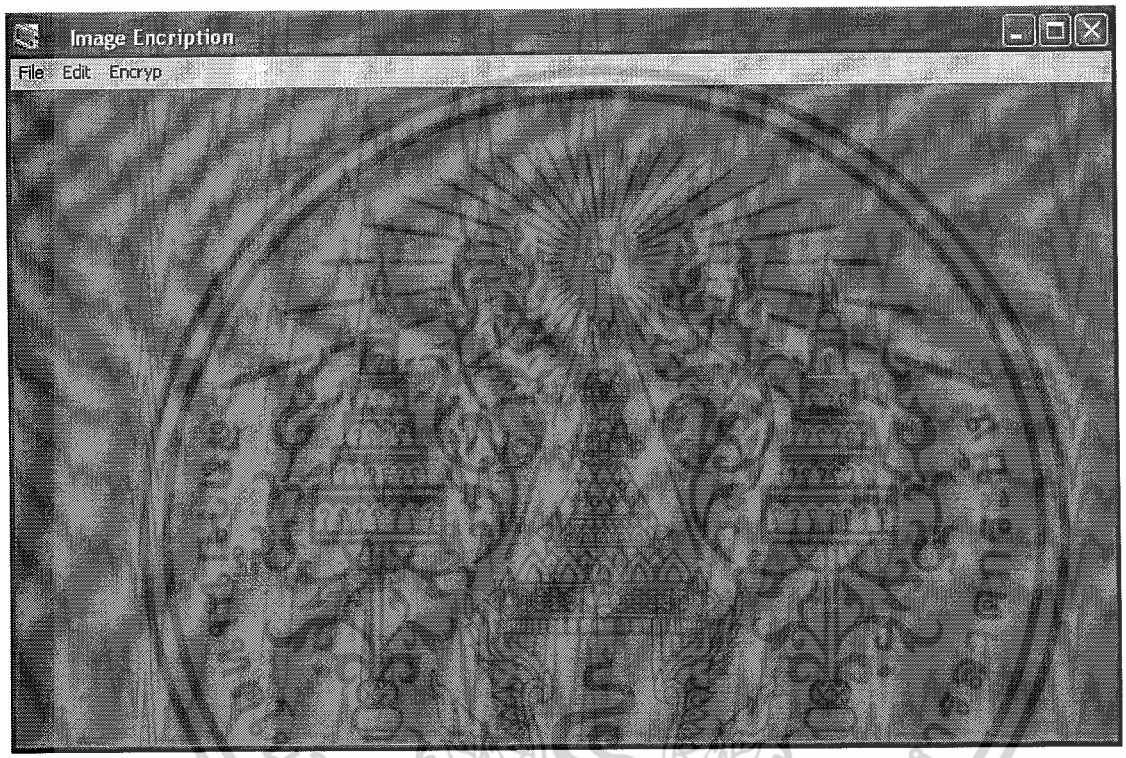
ค่าที่	X_n	ค่าที่	X_n
1	0.6	24	0.291983
2	0.936	25	0.806243
3	0.233626	26	0.60924
4	0.698274	27	0.92846
5	0.821681	28	0.259046
6	0.571434	29	0.748571
7	0.955099	30	0.734029
8	0.167252	31	0.761398
9	0.543186	32	0.708517
10	0.967726	33	0.805431
11	0.121805	34	0.611176
12	0.417178	35	0.926795
13	0.948248	36	0.264598
14	0.191387	37	0.758886
15	0.603556	38	0.713615
16	0.933177	39	0.797038
17	0.243194	40	0.630898
18	0.717797	41	0.908177
19	0.790002	42	0.325228
20	0.647006	43	0.855874
21	0.890718	44	0.48108
22	0.379625	45	0.973604
23	0.918488	46	0.100227

ค่าที่	X_n	ค่าที่	X_n
47	0.351709	74	0.963706
48	0.889238	75	0.13641
49	0.384125	76	0.459429
50	0.922634	77	0.96858
51	0.278383	78	0.118686
52	0.783455	79	0.407939
53	0.661648	80	0.941947
54	0.873093	81	0.213264
55	0.432127	82	0.654351
56	0.957034	83	0.882085
57	0.160369	84	0.405643
58	0.525137	85	0.940277
59	0.972536	86	0.219009
60	0.104169	87	0.667071
61	0.36394	88	0.86614
62	0.902802	89	0.452171
63	0.342228	90	0.966078
64	0.877921	91	0.127807
65	0.417985	92	0.434742
66	0.948767	93	0.958391
67	0.189573	94	0.155522
68	0.599176	95	0.512206
69	0.93664	96	0.974419
70	0.231447	97	0.097214
71	0.69373	98	0.342277
72	0.828628	99	0.877982
73	0.553814	100	0.417806

3.3 การออกแบบซอฟท์แวร์

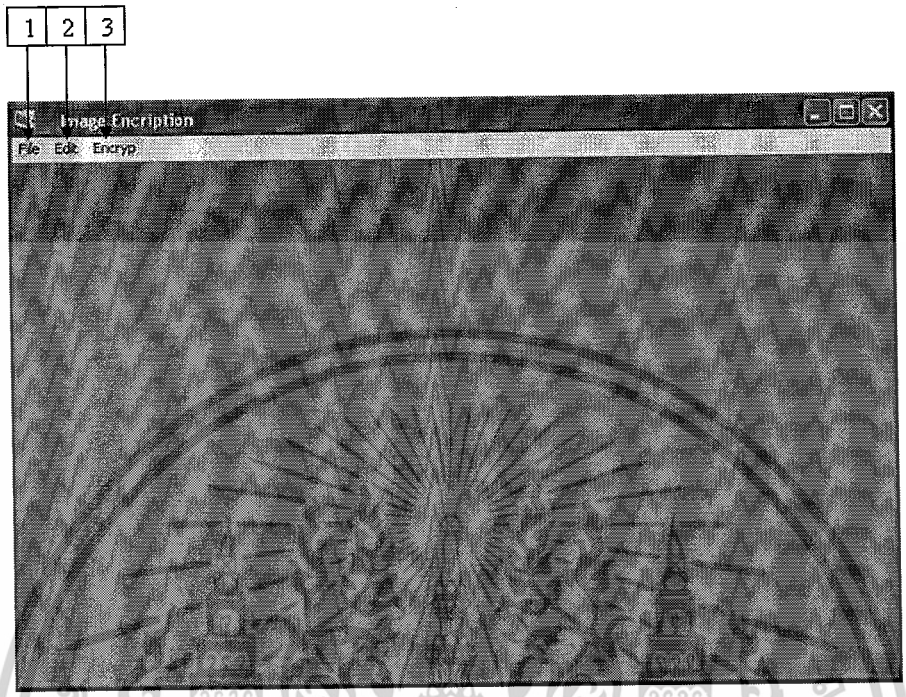
การออกแบบโปรแกรมเข้ารหัสลับอลวน

โปรแกรมการเข้ารหัสลับอลวนจะใช้สมการแม็ปลอจิสติกและ Password ในการเข้ารหัสสัญญาณ โดยโปรแกรมจะรับค่า Password จากผู้ใช้ และ Pattern จากฮาร์ดแวร์ มา XOR ครั้งละ 128 บิต และนำผลลัพธ์ที่ได้ไป XOR กับรูปภาพอีกครั้งหนึ่ง ซึ่งใช้โภาษาซี ในการเขียนและออกแบบโปรแกรมเข้ารหัสสัญญาณ โดยการออกแบบโปรแกรมจะออกแบบให้สามารถทำความเข้าใจและสามารถใช้งานได้อย่างสะดวก ส่วนประกอบต่างๆของโปรแกรมแสดงดังรูป

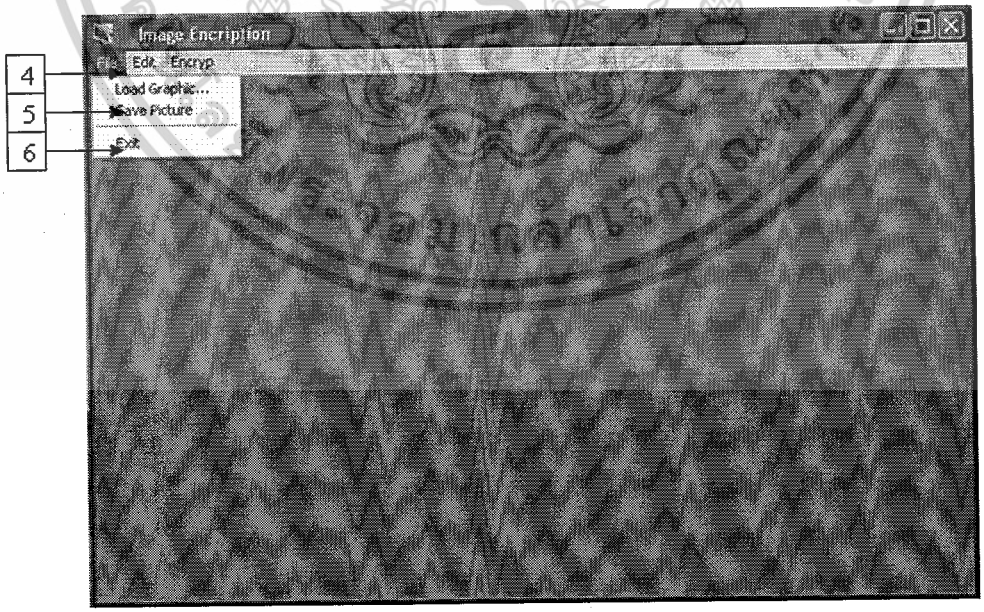


รูปที่ 3.6 โปรแกรมเข้ารหัสลับอลวน

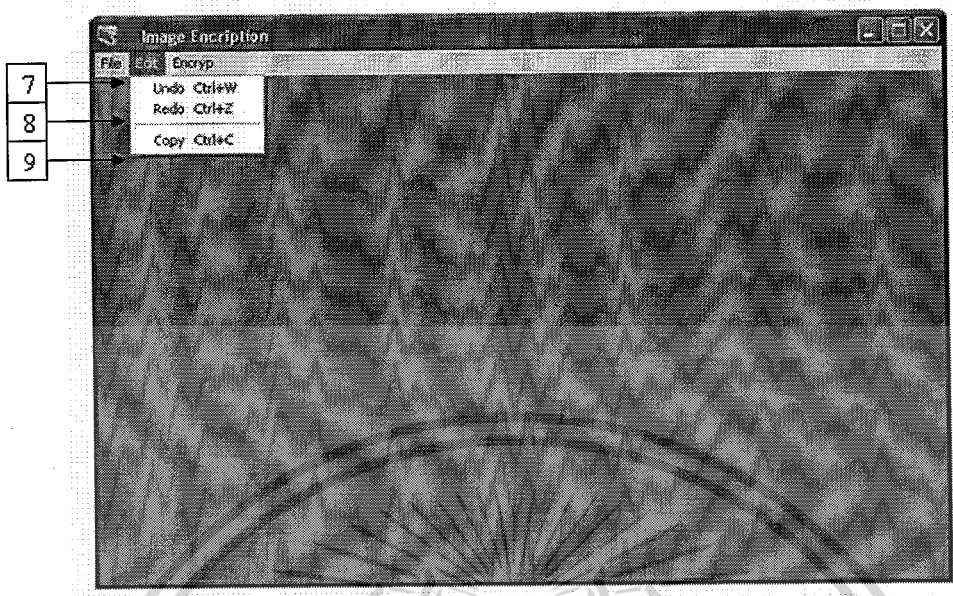
ส่วนประกอบและหน้าที่ของโปรแกรมเข้ารหัสลับอลวน



รูปที่ 3.7 ส่วนประกอบของโปรแกรมเข้ารหัสลับอลวน



รูปที่ 3.8 ส่วนประกอบของคำสั่ง File



รูปที่ 3.9 ส่วนประกอบของคำสั่ง Edit

1 ปุ่มเรียก File

2 ปุ่ม Edit

3 ปุ่มเข้ารหัสลับ และถอดรหัสลับ

4 ใช้สำหรับการเรียกไฟล์รูปภาพที่ต้องการ

5 บันทึกภาพ

6 ออกจากโปรแกรม

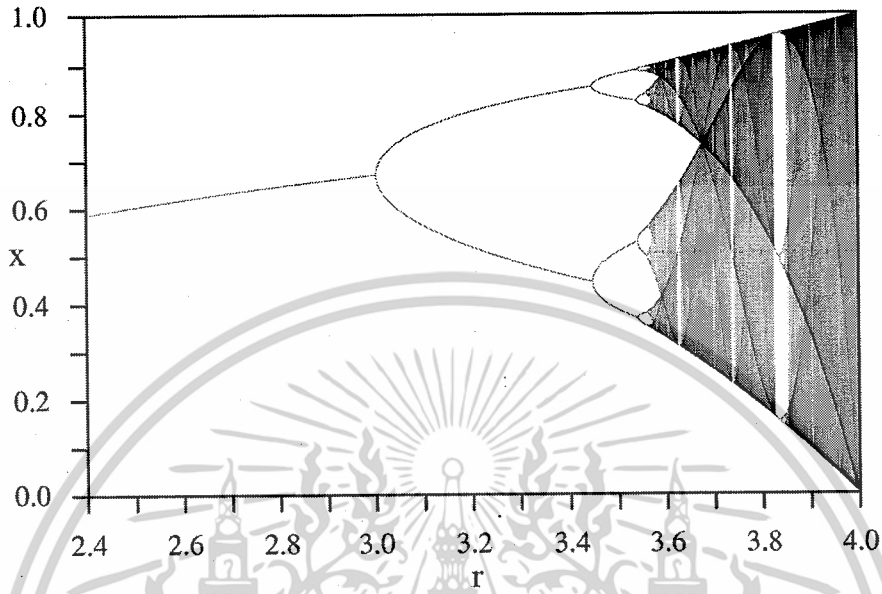
7 คำสั่งกลับไปยังก่อนหน้านี้

8 คำสั่งกลับไปข้างหน้า

9 คัดลอกไฟล์

3.4 การหาค่า r ในสมการอลวน

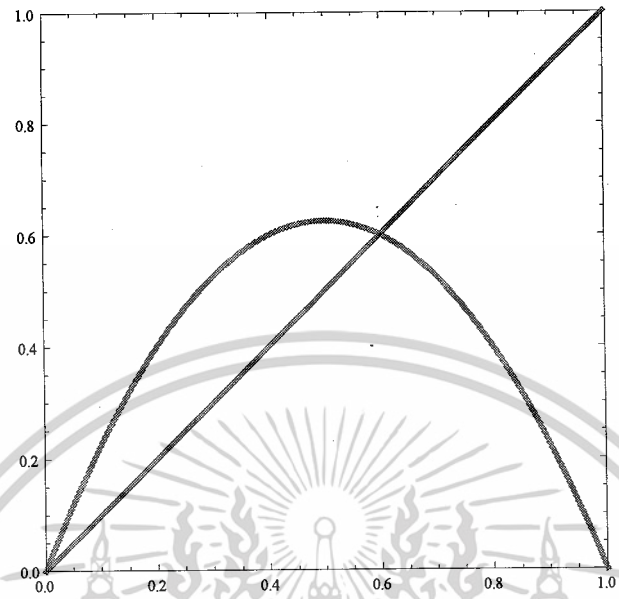
จากสมการ $x_{n+1} = rx_n(1-x_n)$ ต้องหาค่า r ที่เหมาะสม เพื่อจะทำให้เกิดความแปรปรวนของข้อมูล ซึ่งสามารถดูได้จากแผนผังไบเฟอร์เคชัน ของแม็ปลอจิสติก



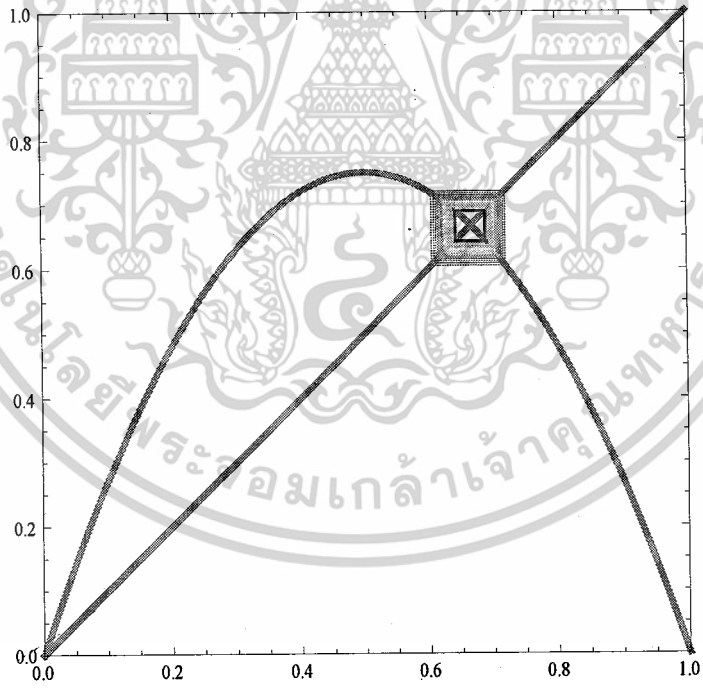
รูปที่ 3.10 แผนผังไบเฟอร์เคชัน ของแม็ปลอจิสติก

ช่วงที่ r >3.5 จะทำให้เกิดความแปรปรวนของข้อมูลมาก ซึ่งจะสามารถหาค่า r จากแบบจำลอง Trajectories Of The Logistic Map ดังนี้ กำหนดให้ n = 100 และ $x_n = 0.6$

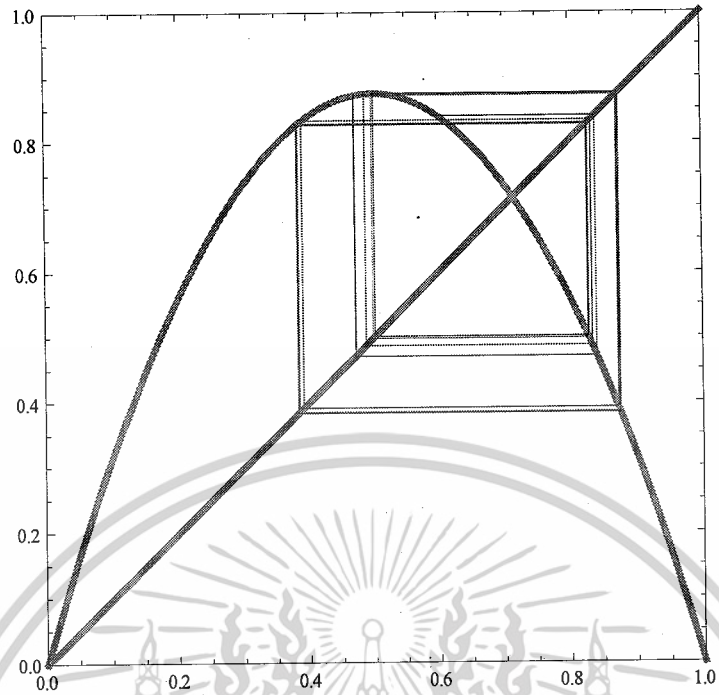
สภาวะอลวนต่างๆเมื่อเปลี่ยนค่า r



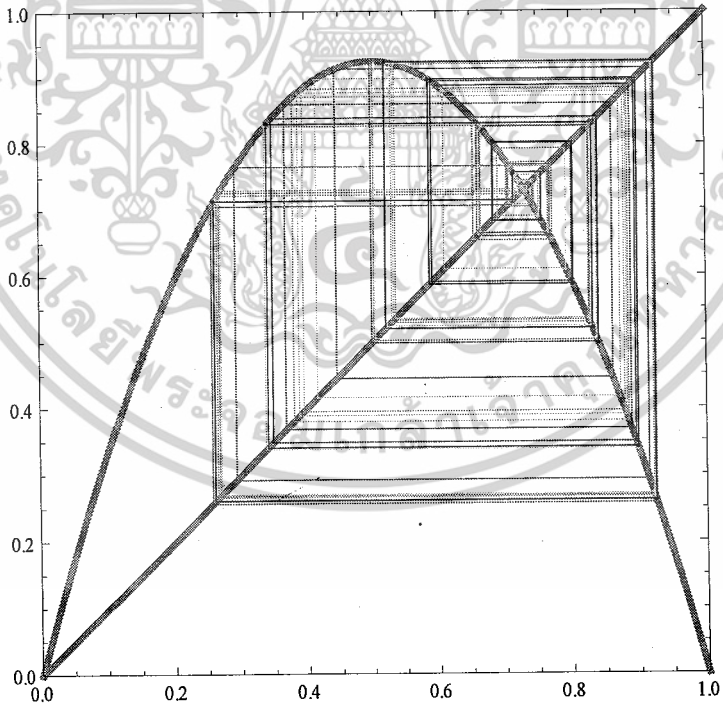
รูปที่3.11 แม็ปลอจิสติก ที่ r=2.5



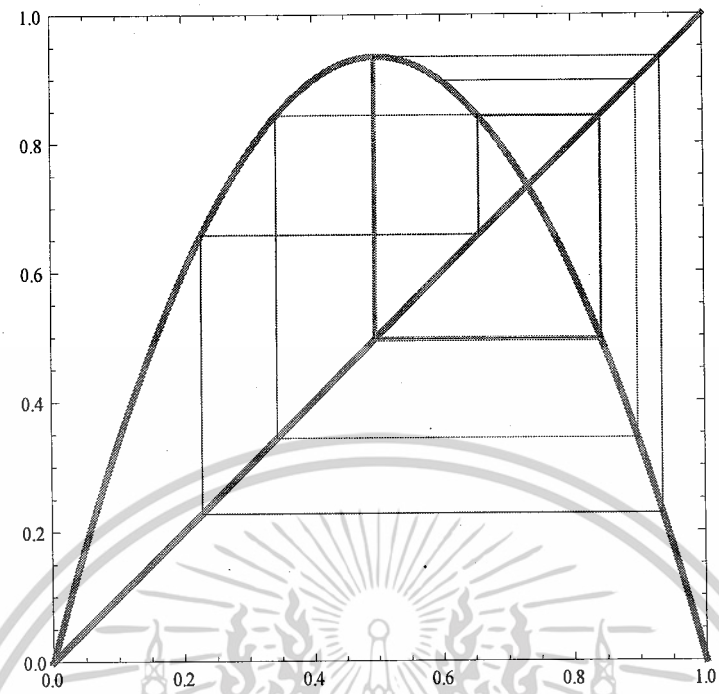
รูปที่3.12 แม็ปลอจิสติก ที่ r=3.0



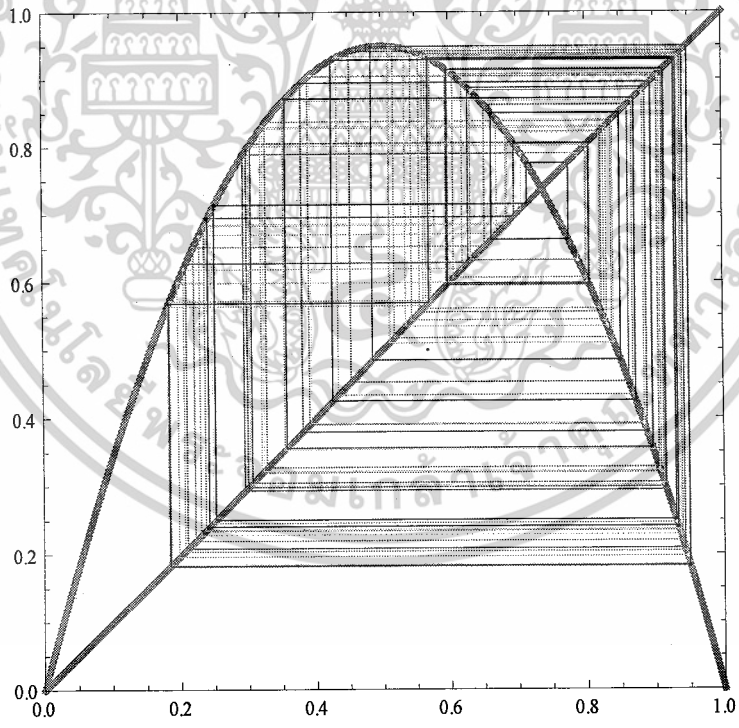
รูปที่3.13 แม็ปลอจิสติก ที่ r=3.5



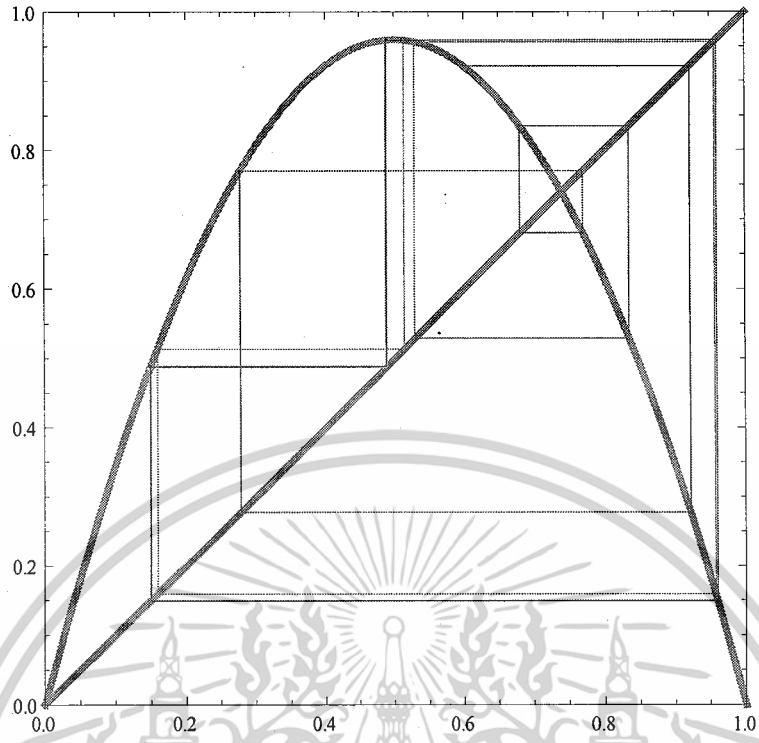
รูปที่ 3.14 แม็ปลอจิสติก ที่ r=3.7



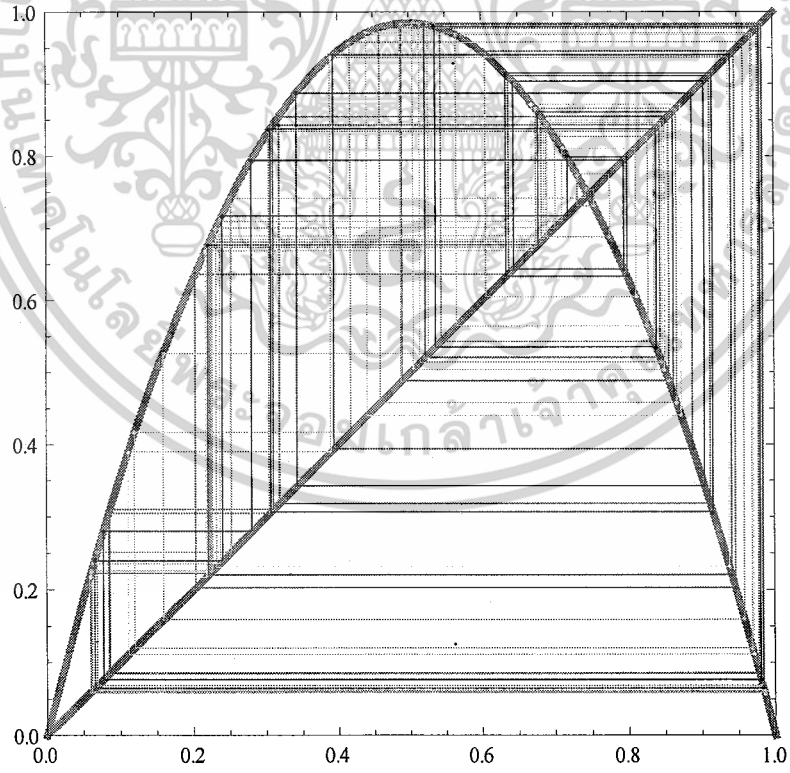
รูปที่ 3.15 แม็ปลอจิสติก ที่ r=3.74



รูปที่ 3.16 แม็ปลอจิสติก ที่ r=3.8

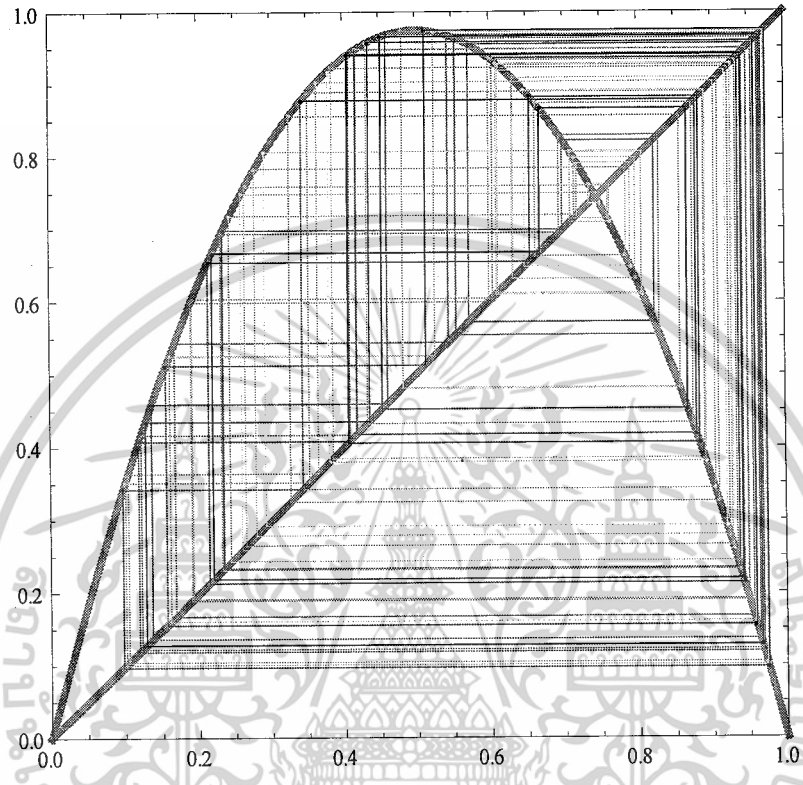


รูปที่ 3.17 แม็ปลอจิสติก ที่ r=3.84

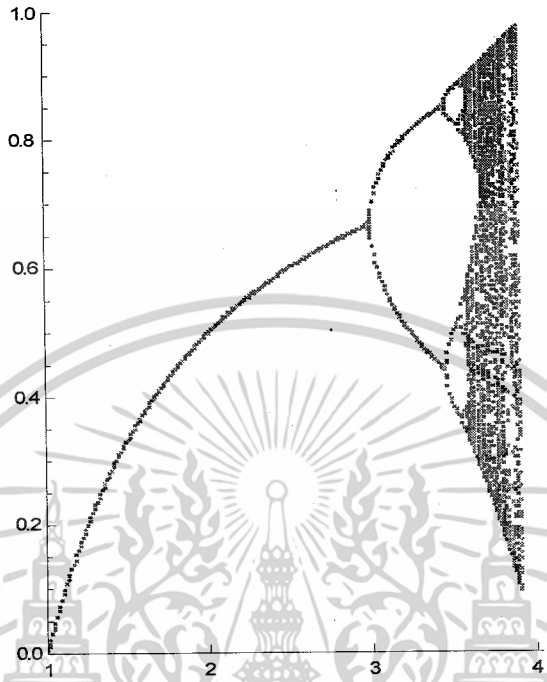


รูปที่ 3.18 แม็ปลอจิสติก ที่ r=3.94

เมื่อค่า r ตั้งแต่ 3.5 จะทำให้ข้อมูลมีความแปรปรวนมากขึ้นเรื่อยๆ แต่จะมีค่า r บางค่าหลังจากนั้นที่จะทำให้ข้อมูลไม่มีความแปรปรวน คือ r=3.74 และ r=3.84 ซึ่งในโครงงานนี้ใช้ค่า r=3.9



รูปที่ 3.19 แม็ปลอจิสติก ที่ x_n=0.6 r=3.9 n=100



รูปที่ 3.20 แผนผังไบเฟอร์เคชัน ที่ x_n=0.6 r=3.9 n=100

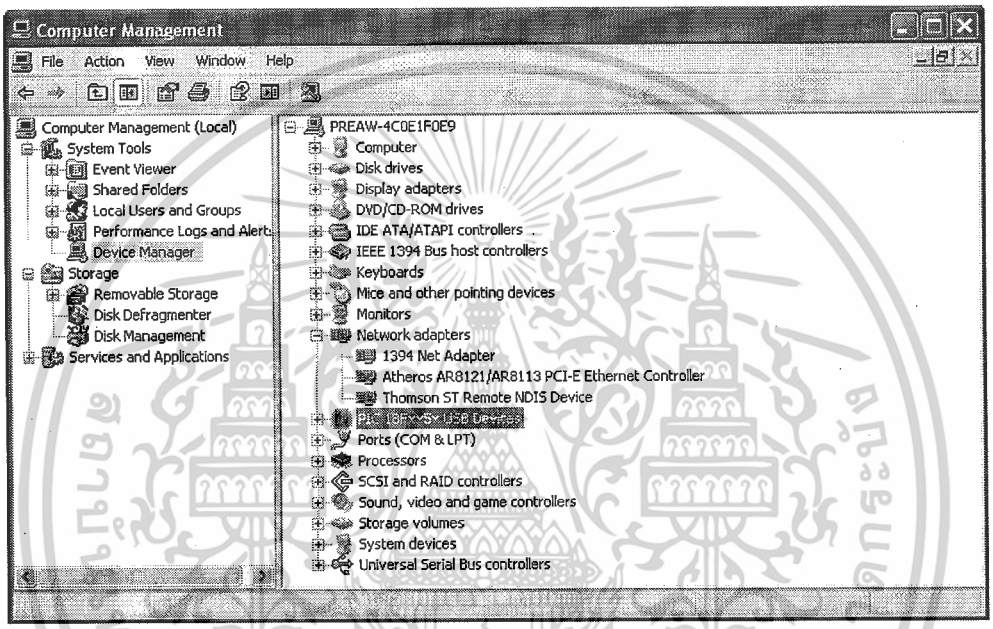
จากการออกแบบโปรแกรมเข้ารหัสลับในบทนี้ สามารถออกแบบฮาร์ดแวร์และซอร์ฟแวร์ได้จริง ในผลการทดลองการเข้ารหัส ในบทต่อไป

บทที่ 4
ผลการทดลอง

ปริญญานิพนธ์นี้ ศึกษาการการทำงานของโปรแกรมเข้ารหัสลับอลวน กับไมโครคอนโทรเลอร์ ซึ่งผลการทดลองแบ่งออกเป็น 2 ส่วนคือ

4.1 การทดลองในส่วนของฮาร์ดแวร์

4.1.1 ทดลองการติดต่อฮาร์ดแวร์ผ่านพอร์ต USB

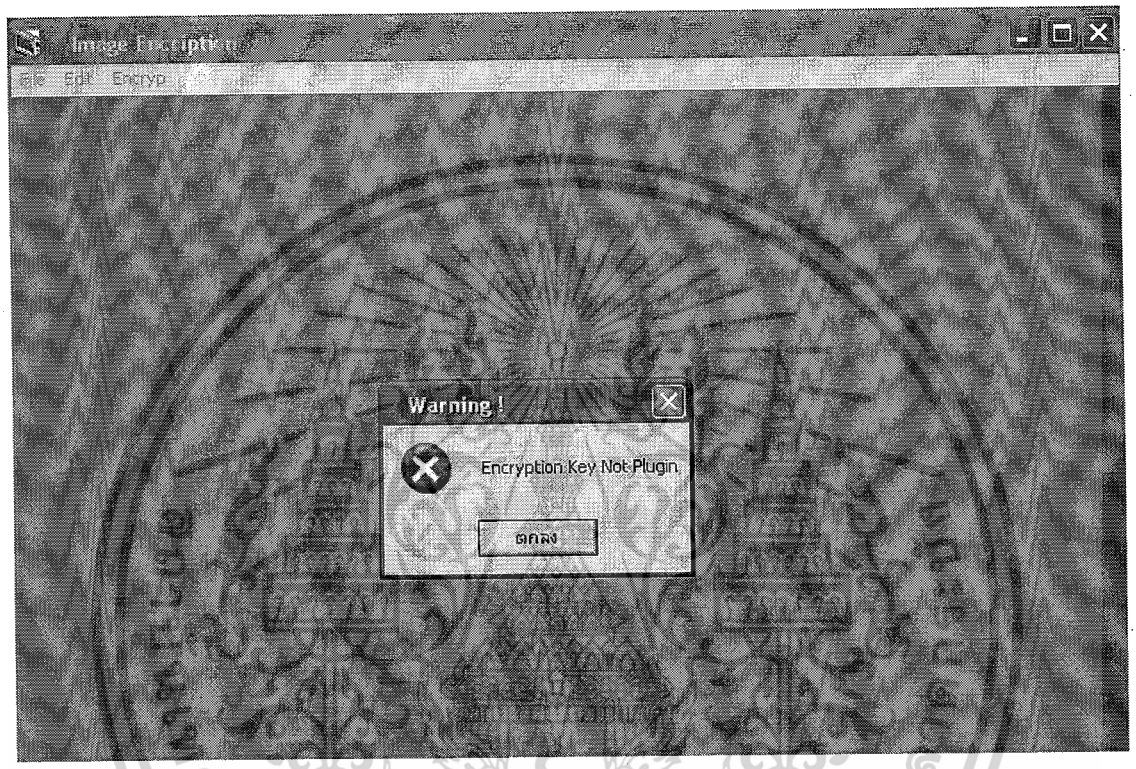


รูปที่ 4.1 การติดต่อระหว่างฮาร์ดแวร์กับคอมพิวเตอร์ผ่านพอร์ต USB

จากรูปที่ 4.1 PIC สามารถทำการติดต่อกับเครื่องคอมพิวเตอร์และโปรแกรมเข้ารหัสลับอลวนได้

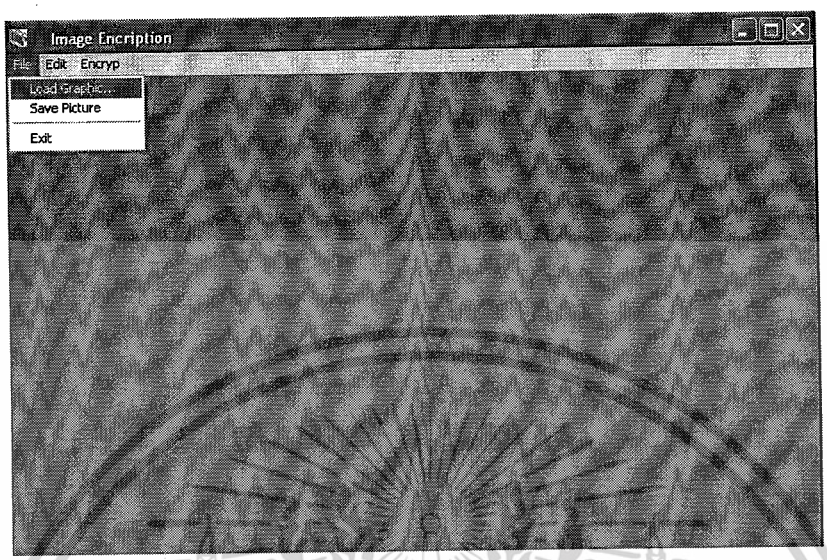
4.2 การทดลองการทำงานของโปรแกรมเข้ารหัสลับอลวนในส่วนต่างๆ

4.2.1 ทดลองป้อนรหัสผ่านให้โปรแกรมเข้ารหัสลับอลวน ซึ่งรหัสผ่านจะต้องเป็นตัวเลขหรือตัวอักษร หากยังไม่ติดตั้ง Driver หรือไม่ติดต่อกับไมโครคอนโทรเลอร์ โปรแกรมจะแจ้งผล และจะไม่ทำงาน ดังรูป

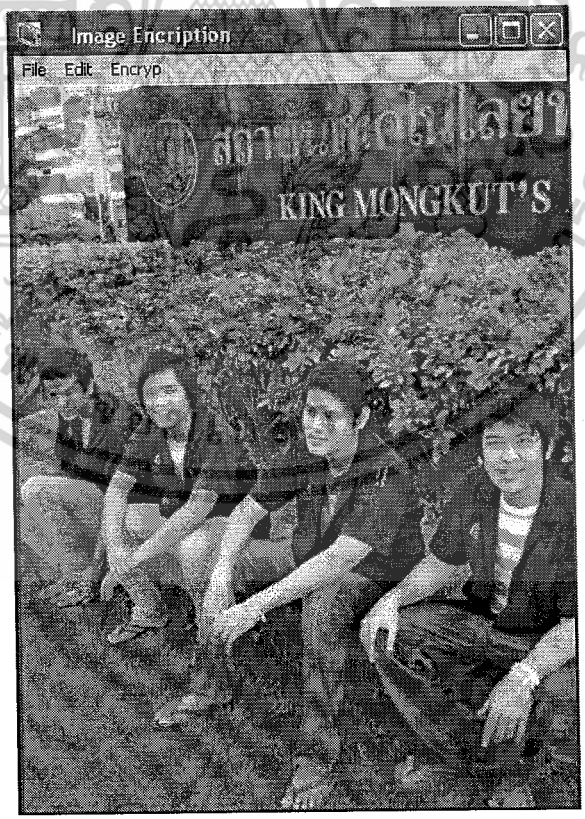


รูปที่ 4.2 ผลการทดลองการใช้โปรแกรมเข้ารหัสลับโดยไม่ได้ติดต่อกับไมโครคอนโทรเลอร์

4.2.2 ทดลองการโหลดไฟล์ภาพโดยคำสั่ง Load Graphic ของโปรแกรมเข้ารหัสลับ ซึ่งจะทำหน้าที่โหลดไฟล์ภาพ เพื่อนำมาเข้ารหัสในโปรแกรม ผลการทดลองแสดงดังรูป



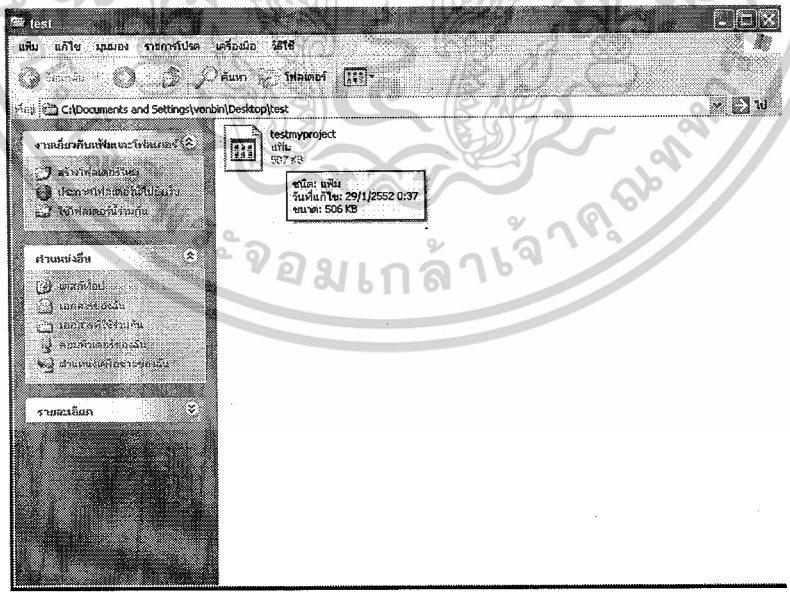
รูปที่ 4.3 คำสั่ง Load Graphic ในโปรแกรมเข้ารหัสลับ



รูปที่ 4.4 การโหลดภาพโดยใช้คำสั่ง Load Graphic

4.2.3 ทดลองเซฟไฟล์ภาพโดยคำสั่ง Save Picture ของโปรแกรมเข้ารหัสลับ

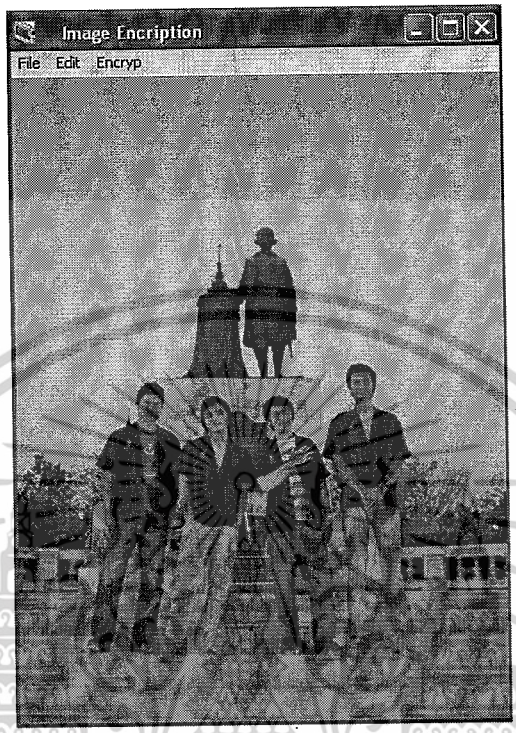
รูปที่ 4.5 ทดลองการบันทึกภาพในโปรแกรมเข้ารหัสลับ



รูปที่ 4.6 ไฟล์ภาพหลังการบันทึก

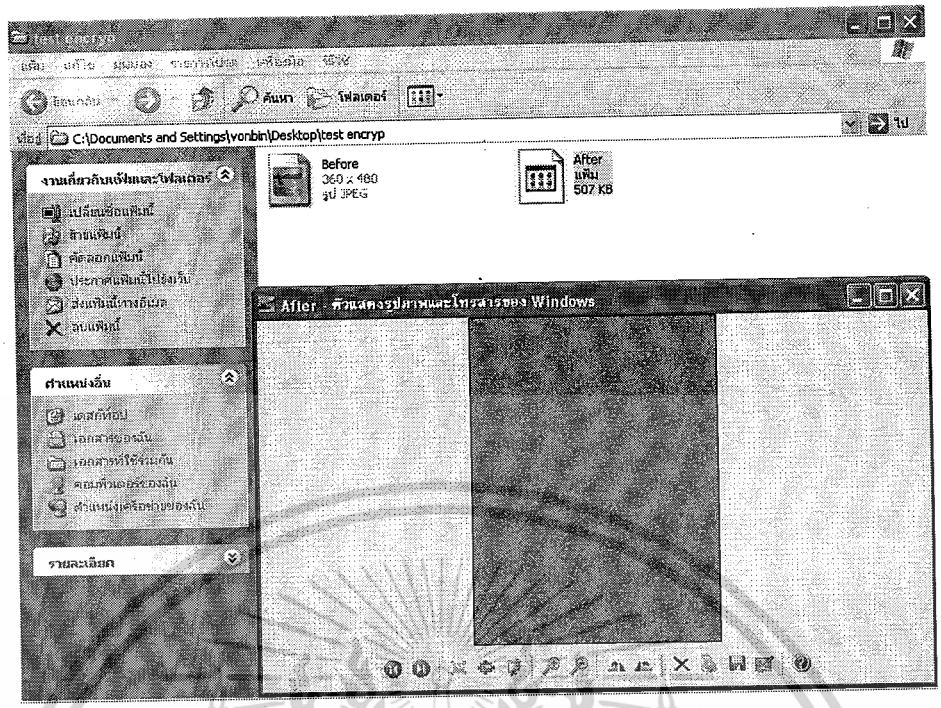
4.3 การทดลองการทำงานในส่วนของการเข้ารหัสลับของโปรแกรมเข้ารหัสลับ

4.3.1 ทดลองการเข้ารหัสลับของภาพ โดยการเรียกไฟล์ภาพเข้าสู่โปรแกรม แล้วทำการใส่รหัสเพื่อเข้ารหัสภาพ โดยไฟล์ภาพจะเป็นไฟล์ชนิด JPEG ซึ่งผลการทดลองเข้ารหัสภาพแสดงดังรูป



รูปที่ 4.7 ภาพก่อนเข้ารหัส

รูปที่ 4.8 ภาพหลังเข้ารหัส

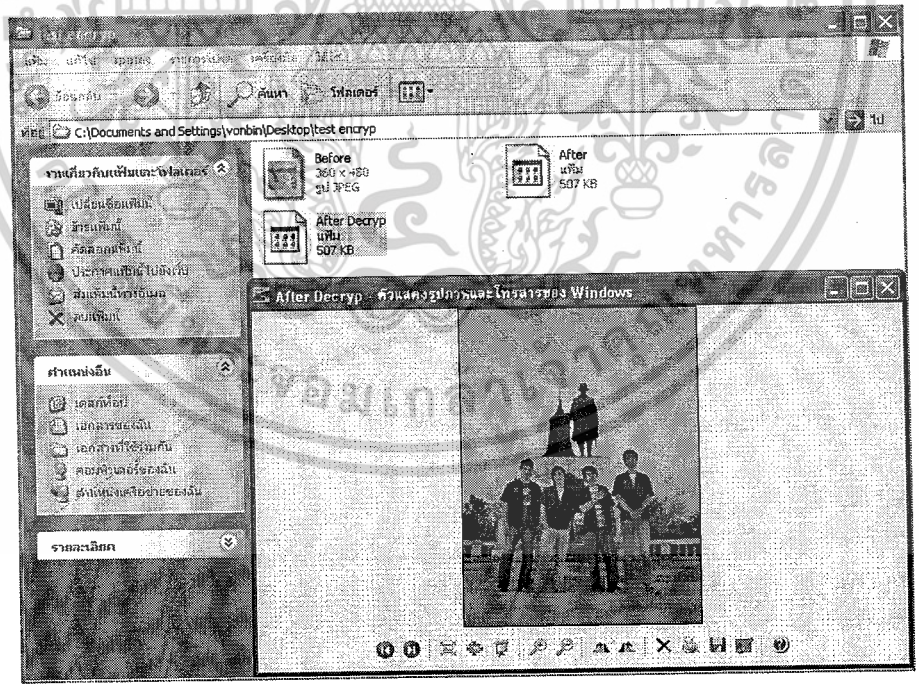


รูปที่ 4.9 ลักษณะไฟล์ภาพก่อนและหลังการเข้ารหัส

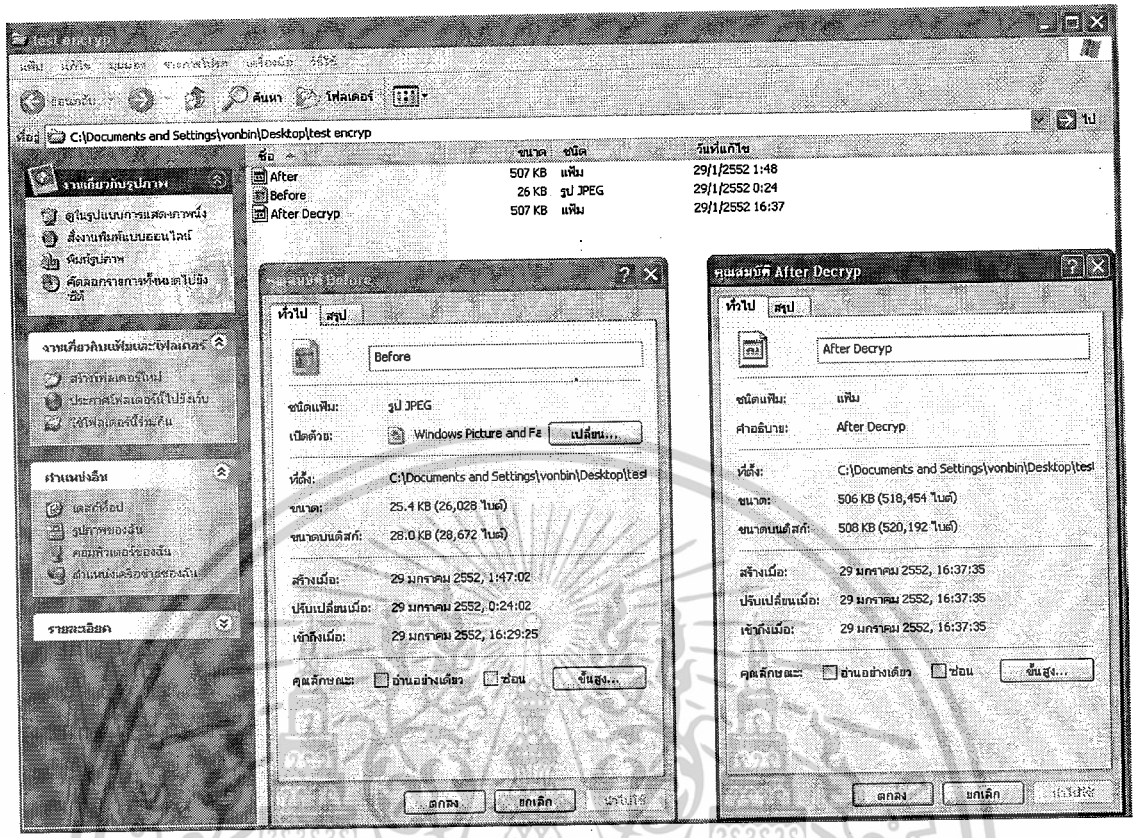
4.3.2 ทดลองการถอดรหัสลับของภาพ โดยการเรียกไฟล์ที่ผ่านการเข้ารหัสจากหัวข้อ 4.2.1 เพื่อนำมาถอดรหัสให้ได้ภาพที่เป็นภาพเดิมก่อนการเข้ารหัส ผลการทดลองแสดงดังรูป

รูปที่ 4.10 ภาพก่อนถอดรหัส

รูปที่ 4.11 ภาพหลังถอดรหัส



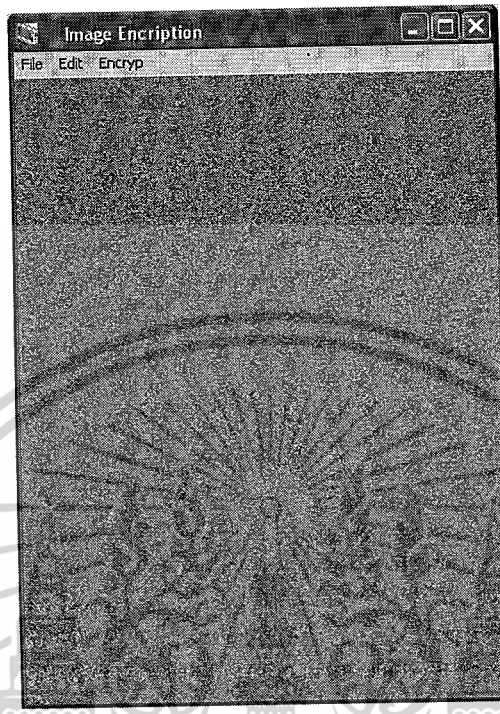
รูปที่ 4.12 ไฟล์ภาพหลังถอดรหัส



รูปที่ 4.13 เปรียบเทียบขนาดของไฟล์ภาพต้นแบบ และหลังการถอดรหัส

รูปที่ 4.14 เปรียบเทียบภาพต้นแบบ และภาพหลังการถอดรหัส

4.3.3 ทดลองการใส่รหัสที่ผิด ในการเข้าถอดรหัสลับ โดยการเรียกไฟล์ที่ผ่านการเข้ารหัสลับเช่นเดียวกับหัวข้อ 4.2.2 เพื่อนำมาทดลองใส่รหัสที่ผิดในการถอดรหัสลับ ผลการทดลองแสดงดังรูป

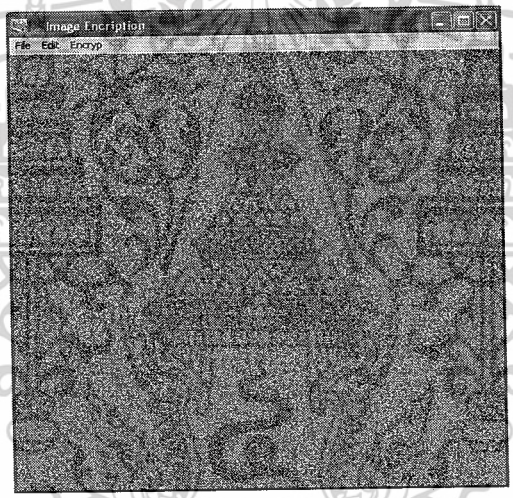


รูปที่ 4.15 ภาพก่อนการถอดรหัส

รูปที่ 4.16 ภาพหลังการใส่รหัสผิดในการถอดรหัส

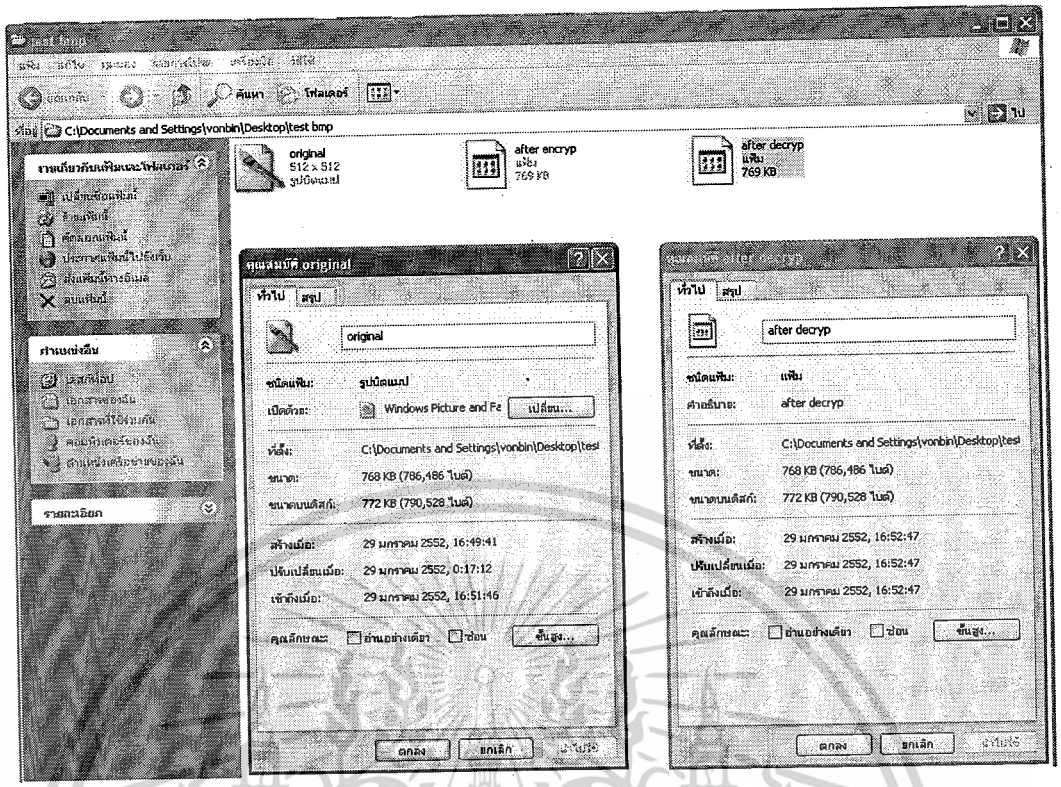
4.3.4 ทดลองการเข้าและถอดรหัสลับ โดยใช้ไฟล์ภาพตระกูล BMP

รูปที่ 4.17 ภาพก่อนเข้ารหัส



รูปที่ 4.18 ภาพหลังเข้ารหัส

รูปที่ 4.19 ภาพหลังการถอดรหัส

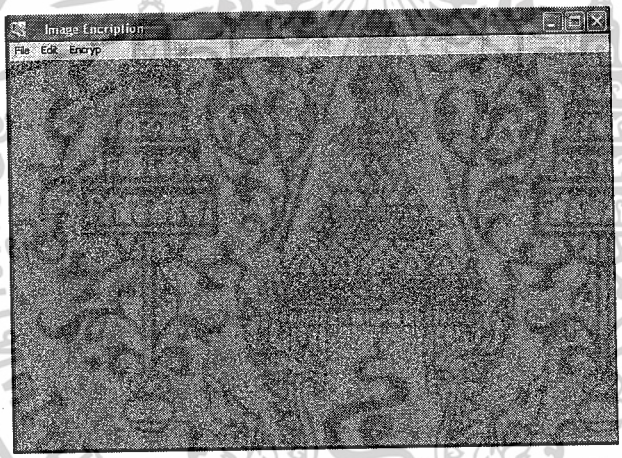


รูปที่ 4.20 เปรียบเทียบขนาดของไฟล์ภาพต้นแบบ และหลังการถอดรหัส

รูปที่ 4.21 เปรียบเทียบภาพต้นแบบ และภาพหลังการถอดรหัส

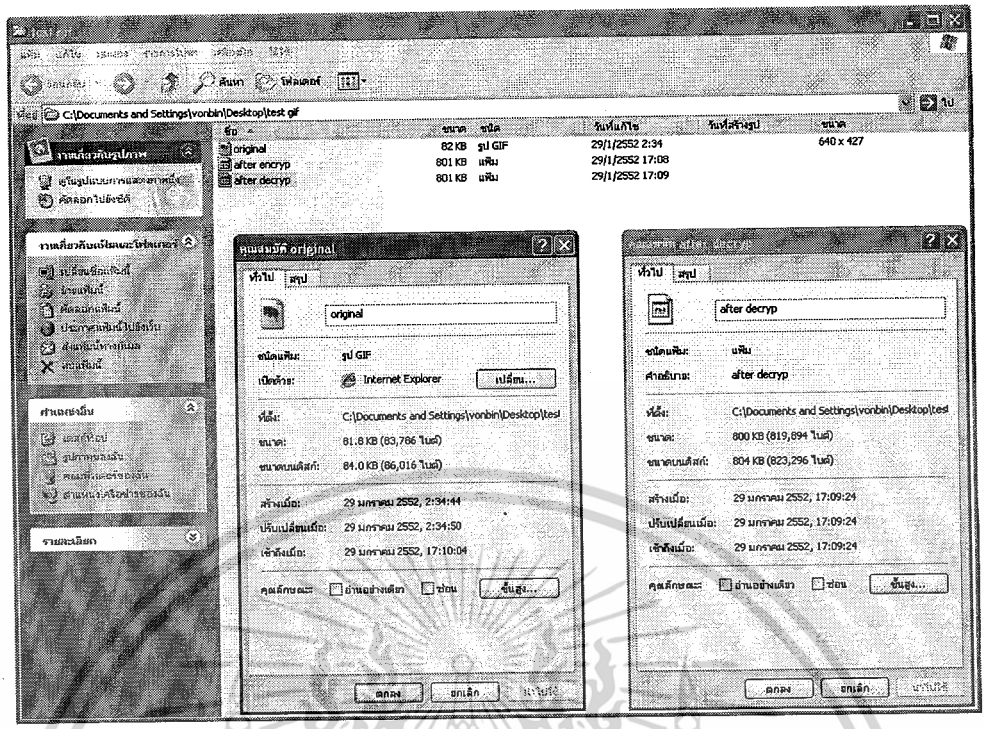
4.3.5 ทดลองการเข้ารหัสลับและการถอดรหัสลับของภาพสกุล GIF

รูปที่ 4.22 ภาพก่อนเข้ารหัส

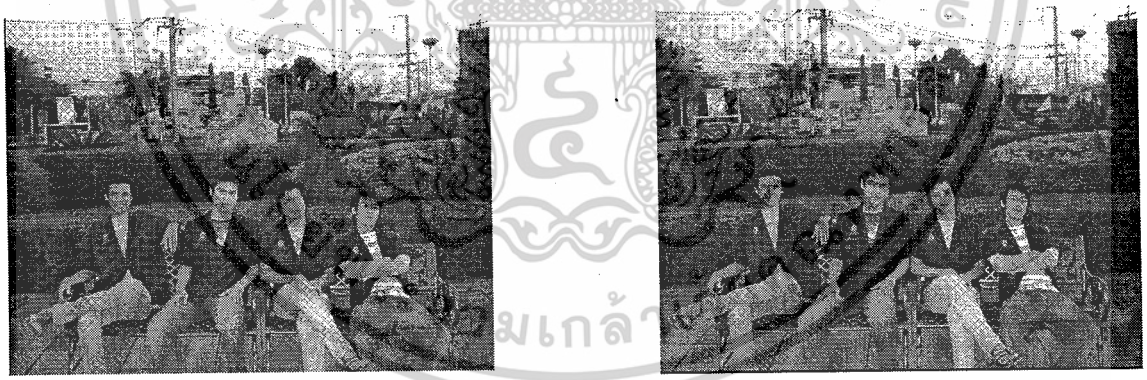


รูปที่ 4.23 ภาพหลังเข้ารหัส

รูปที่ 4.24 ภาพหลังการถอดรหัส

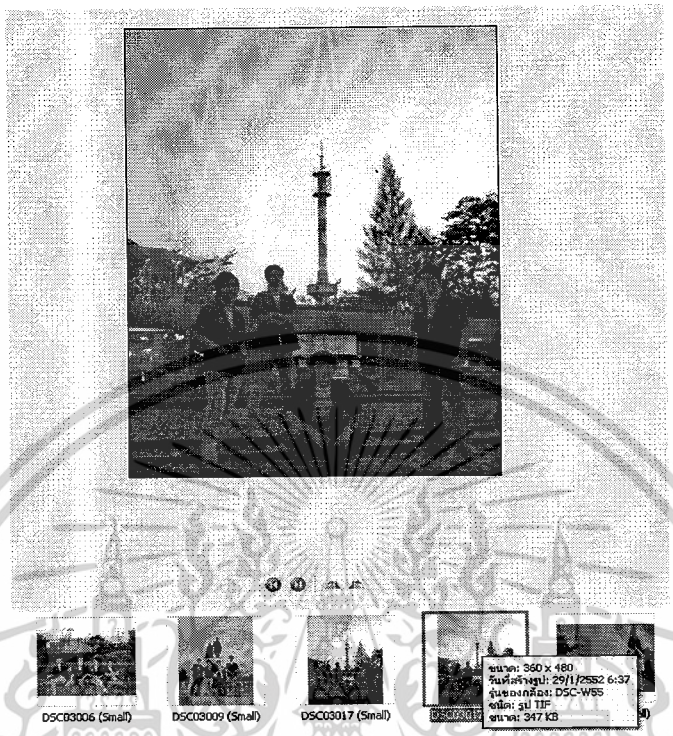


รูปที่ 4.25 เปรียบเทียบขนาดของไฟล์ภาพต้นแบบ และหลังการถอดรหัส



รูปที่ 4.26 เปรียบเทียบภาพต้นแบบ และภาพหลังการถอดรหัส

4.3.6 ทดลองการเข้ารหัสลับและการถอดรหัสลับของภาพสกุล TIFF

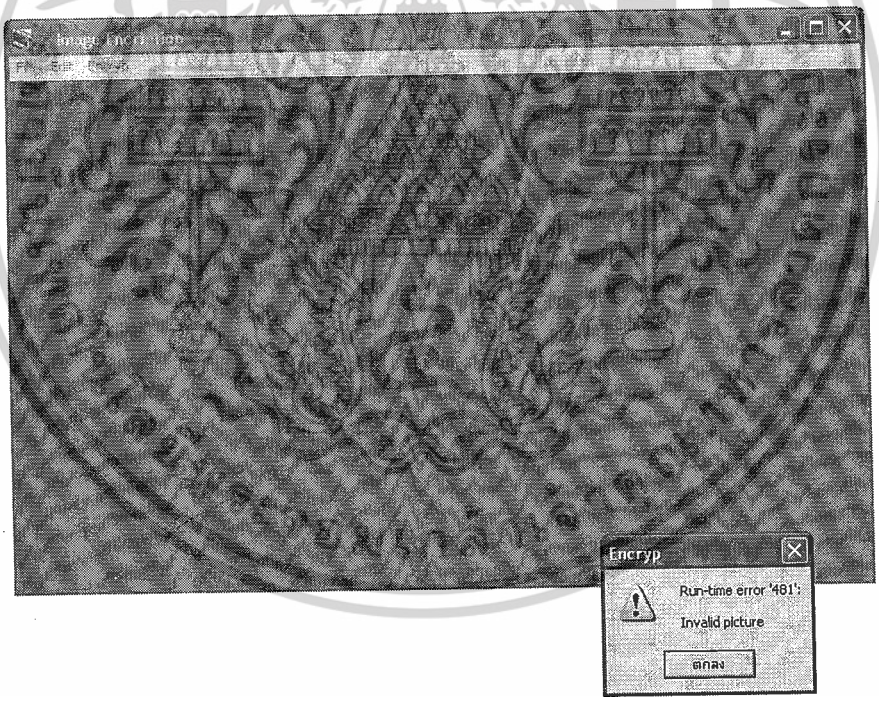


รูปที่ 4.27 ภาพก่อนเข้ารหัส

รูปที่ 4.28 โปรแกรมไม่สามารถเรียกไฟล์มาเข้ารหัสได้

4.3.7 ทดลองการเข้ารหัสลับและการถอดรหัสลับของภาพสกุล PDF

รูปที่ 4.29 ภาพก่อนเข้ารหัส



รูปที่ 4.30 โปรแกรมไม่สามารถเรียกไฟล์มาเข้ารหัสได้

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

ปริญญานิพนธ์นี้ เสนอการเข้ารหัสลับภาพด้วยสัญญาณอลวน โดยสัญญาณอลวนที่ใช้เป็นรูปแบบ Logistic Map โดยการนำเอาพิกเซลของภาพมาทำ XOR กับส่วนที่เป็นสัญญาณอลวน โดยการเข้ารหัสนี้จะประกอบด้วยส่วนของโปรแกรมคอมพิวเตอร์และฮาร์ดแวร์ ซึ่งจะทำให้การเข้ารหัสมีความแข็งแกร่งยิ่งขึ้นกว่าการเข้ารหัสที่ใช้เพียงโปรแกรมคอมพิวเตอร์หรือฮาร์ดแวร์อย่างใดอย่างหนึ่ง

ในส่วนของฮาร์ดแวร์ใช้ไมโครคอนโทรลเลอร์ชนิด PIC ET-ICDX V1.0 เป็นส่วนกำเนิดสัญญาณอลวน ซึ่งจะเปรียบเสมือนแม่กุญแจของกระบวนการเข้ารหัส จากการศึกษาข้อมูลทฤษฎีอลวน ซึ่งสามารถสร้าง Pattern ของผลลัพธ์ที่ได้จากการผ่านสมการมาแล้วนั้น มีความหลากหลายกว่า ทฤษฎี Pseudo random ในกรณีที่มี Parameter เท่ากัน ทางคณะผู้จัดทำจึงใช้ สมการ Logistic Map ในการทำปริญญานิพนธ์นี้การกำเนิดสัญญาณอลวนจะใช้ภาษาซีเขียนโปรแกรมในการคำนวณสมการแม็ปลอจิสติกลงในไมโครคอนโทรลเลอร์ และทำการส่งค่าที่ได้ผ่านช่องสัญญาณ USB เพื่อเข้าสู่กระบวนการเข้ารหัสกับโปรแกรมคอมพิวเตอร์ต่อไป

ในส่วนของโปรแกรมคอมพิวเตอร์จะใช้ภาษาซีในการเขียนโปรแกรมเช่นเดียวกัน โดยโปรแกรมจะรับสัญญาณอลวนจากฮาร์ดแวร์แล้วนำมา XOR กับ key ซึ่งเปรียบเสมือนลูกกุญแจในการเข้ารหัส แล้วจึงนำไป XOR กับพิกเซลของภาพอีกครั้งหนึ่ง

จากการทดสอบภาพที่ได้กับชนิดไฟล์แบบ JPEG, BMP, GIF พบว่า ขนาดของไฟล์ที่เข้ารหัสแบบ BMP ได้ขนาดของไฟล์เท่าเดิม ในขณะที่ไฟล์แบบอื่น ๆ ได้ขนาดของไฟล์ที่เข้าและถอดรหัสเพิ่มขึ้น ทั้งนี้เป็นเพราะการเรียงพิกเซลของไฟล์ผลลัพธ์จากการเข้ารหัสเป็นแบบ BMP จึงทำให้ได้ขนาดของไฟล์เพิ่มขึ้น อย่างไรก็ตาม อุปกรณ์เข้ารหัสลับในโครงงานนี้ไม่ครอบคลุมถึงไฟล์ที่เป็นแบบ TIFF และแบบ PDF จึงเป็นข้อเสนอแนะว่าสมควรขยายผลสำหรับโครงงานในอนาคตที่จะครอบคลุมถึงรูปแบบไฟล์อื่น ๆ โดยอาศัยแนวทางที่นำเสนอในโครงงานนี้

บรรณานุกรม

[1] นิรุธ อำนวยศิลป์, "**โครงสร้างข้อมูล การเขียนโปรแกรมและการประยุกต์,**" บริษัทดวงกมล จำกัด, 2548.

[2] ประภาพร ช่างไม้, "**คู่มือการเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น,**" อินโฟเพรส, 2545.

[3] กฤดากร กล่อมการ, "**การสื่อสารข้อมูล,**" แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2545.

[4] สมเกียรติ อุดมหรรษากุล, "**การประมวลผลภาพเบื้องต้น,**" แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2550.

[5] ดอนสัน ปงผาบ, "**ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน,**" สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2549.

[6] http://en.wikipedia.org/wiki/Bifurcation_diagram

[7] http://hpcmath.kmutt.ac.th/moodle/mod/forum/discuss.php?d=318

[8] http://www.thaicert.org/paper/encryption/intro_crypt.php

[9] http://www.nextproject.net/contents/default.aspx?00044

[10] http://th.wikipedia.org/wiki/Encryption_and_Decryption_TXT

ภาคผนวก ก

Function ในโปรแกรม

Function บางส่วนใน **ET-ICDX V1.0**

```
//Header
#include <p16F877.inc>
#include <stdio.h>


//ประกาศ function
void init_serial0 (void);
char putchar (char ch);
char getchar (void);
void delay(unsigned long int count){


//function หลัก
int main(void)
{
 long int i;
 long float xn    = 0.6;
 long float  r = 3.9;
 char key1[16],key2[16],key3[16];
 char input;


 init_serial0();


  while(1)
{
        for(i=0; i!=100; i++)
        {
         xn = (r*xn*(1-xn));   // สมการอลวน 100 รอบ
        }
```

// ส่ง key 3 ชุด ชุดละ 16 บิต จำนวน ไม่จำกัดรอบรอบ

```
Char Result[] = xn;


For(int i=0,int j=0 ; j<16  ; i++,j++  )
        Key1[i]=result[j];
For(int j=16,int i=0 ; j<32 ;i++, j++ )
        Key[i]=result[j];
For(int j=32,int i=0 ; j<48 ; i++,j++ )
        Key[i]=result[j];



        for(i=0;i!=1;i--)
        {
                putchar(key[1]);
                putchar(key[2]);
                putchar(key[3]);

        }


 }
}
```

//ประกาศ port ติดต่อระหว่าง software กับ hardware

```
void init_serial0 (void)
{
 PINSEL0 &= 0xFFFFFFF0;

 PINSEL0 |= 0x00000001;

 PINSEL0 |= 0x00000004;
```

```
U0LCR &= 0xFC;

U0LCR |= 0x03;

U0LCR &= 0xFB;

U0LCR &= 0xF7;

U0LCR &= 0xBF;

U0LCR |= 0x80;



U0DLM = 0x01;

U0DLL = 0x87;

U0LCR &= 0x7F;



U0FCR |= 0x01;

U0FCR |= 0x02;

U0FCR |= 0x04;
U0FCR &= 0x3F;
}
```

```
// function ส่ง
char putchar (char ch)
{
 if (ch == '\n')
 {
  while (!(U0LSR & 0x20));
  U0THR = 0x0D;
 }
 while (!(U0LSR & 0x20));

 return (U0THR = ch);

}


// function รับค่า
char getchar (void)
{
 while (!(U0LSR & 0x01));

 return (U0RBR);

}
```

ภาคผนวก

ข

ไมโครคอนโทรลเลอร์ PIC ET-ICDX V1.0

ET-ICDX V1.0

ET-ICDX V1.0 เป็นเครื่องมือที่ใช้ในการโปรแกรมและดีบักไมโครคอนโทรลเลอร์ PIC ของบริษัท microchip โดยการใช้งานจะต้องใช้งานกับโปรแกรม MPLAB ซึ่งสามารถดาวน์โหลดมาใช้ได้ฟรีที่ www.microchip.com ซึ่ง ณ ปัจจุบันที่เขียนเอกสารนี้คือเวอร์ชั่น 8.10 โดยเบอร์ของ PIC ที่ ET-ICDX V1.0 สามารถโปรแกรมและดีบักได้มีดังนี้ซึ่งสามารถเพิ่มขึ้นได้ในอนาคตในกรณีที่มี MPLAP เวอร์ชั่นใหม่

Device Support List

Debugger - Full Support

dsPIC30F2010	dsPIC30F4011	dsPIC30F6011A
dsPIC30F2011	dsPIC30F4012	dsPIC30F6012
dsPIC30F2012	dsPIC30F4013	dsPIC30F6012A
dsPIC30F2020	dsPIC30F5011	dsPIC30F6013
dsPIC30F2023	dsPIC30F5013	dsPIC30F6013A
dsPIC30F3010	dsPIC30F5015	dsPIC30F6014
dsPIC30F3011	dsPIC30F5016	dsPIC30F6014A
dsPIC30F3012	dsPIC30F6010	dsPIC30F6015
dsPIC30F3013	dsPIC30F6010A	
dsPIC30F3014	dsPIC30F6011	
dsPIC33FJ128GP202	dsPIC33FJ12GP201	dsPIC33FJ64GP202
dsPIC33FJ128GP204	dsPIC33FJ12GP202	dsPIC33FJ64GP204
dsPIC33FJ128GP206	dsPIC33FJ12MC201	dsPIC33FJ64GP206
dsPIC33FJ128GP306	dsPIC33FJ12MC202	dsPIC33FJ64GP306
dsPIC33FJ128GP310	dsPIC33FJ16GP304	dsPIC33FJ64GP310
dsPIC33FJ128GP706	dsPIC33FJ16MC304	dsPIC33FJ64GP706
dsPIC33FJ128GP708	dsPIC33FJ256GP506	dsPIC33FJ64GP708

คุณสมบัติของ ET-ICDX V1.0

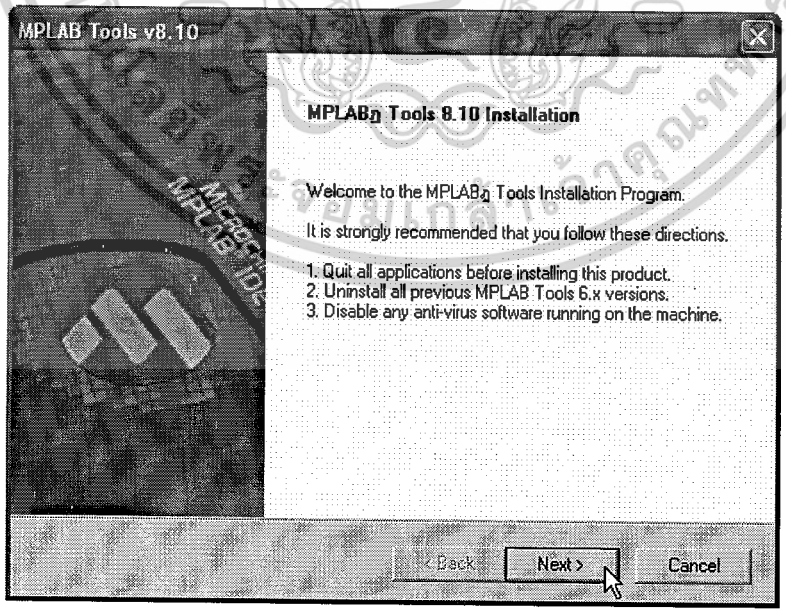
- การเชื่อมต่อเป็น USB (Full Speed 2 M bits/s)
- สามารถโปรแกรมและดีบักไมโครคอนโทรลเลอร์ PIC และ dsPIC ได้
- ใช้งานร่วมกับโปรแกรม MPLAB IDE (ดาวน์โหลดฟรี)
- สามารถอัพเกรดเพิ่มเติมเบอร์ของไมโครคอนโทรลเลอร์ ใหม่ๆ ด้วยตนเองผ่านทางคอมพิวเตอร์
- สามารถใช้งานร่วมกับบอร์ดเป้าหมายที่มีไฟเลี้ยงตั้งแต่ 2.0-6.0 V ได้
- มี LED แสดงผลการทำงาน POWER,BUSY,ERROR
- สามารถอ่านและเขียนพื้นที่ในหน่วยความจำและส่วนของพื้นที่อีอีพรอมได้
- สามารถโปรแกรมค่า configuration bits ได้
- ใช้แจ๊กแบบโมดูล่าร์ 6 ขา และจัดเรียงขาตามมาตรฐานแจ๊ก ICD2 ของ Microchip ทำให้สามารถใช้งานกับบอร์ดของ Microchip หรือบอร์ดที่แจ็ก ICD2 ได้ทันที

1.การติดตั้งโปรแกรม MPLAB IDE

ก่อนที่จะใช้งาน ET-ICDX V1.0 ได้นั้นผู้ใช้จำเป็นที่จะต้องทำการติดตั้งโปรแกรม MPLAB IDE เสียก่อนซึ่งสามารถดาวน์โหลดได้ที่ www.microchip.com หรือใน CD ROM แผ่นนี้

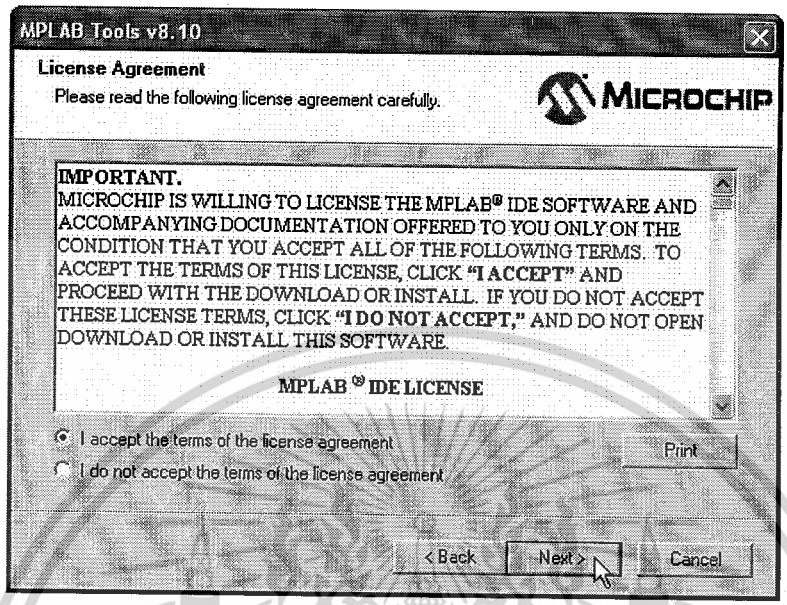
1.1 ทำการติดตั้งโปรแกรม MPLAB IDE โดยการดับเบิลคลิกที่ไฟล์ Install_MPLAB_v810.exe

1.2 จากนั้นปรากฏหน้าต่างดังรูปที่ 1-1 จากนั้นคลิกปุ่ม Next เพื่อสู่ขั้นตอนต่อไป



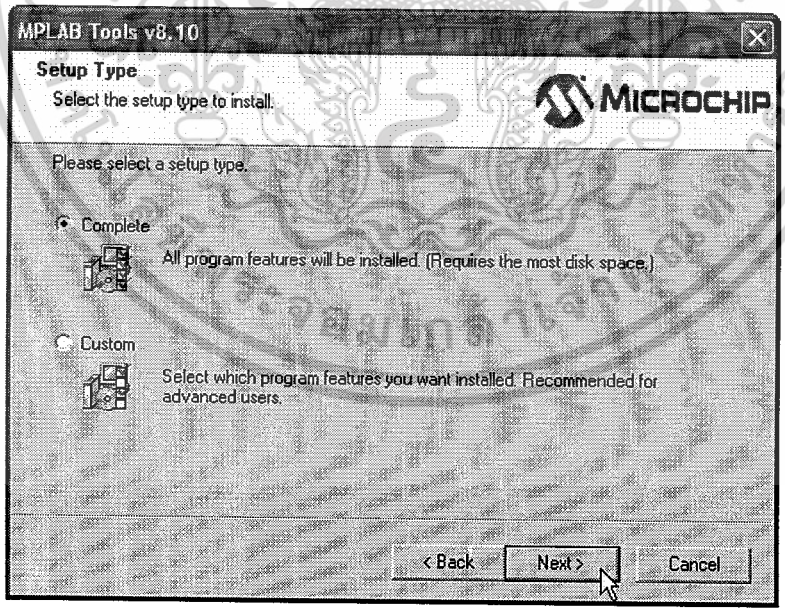
รูปที่ 1-1 แสดงหน้าต่างเริ่มต้นการติดตั้งโปรแกรม

1.3 จากนั้นจะปรากฏหน้าต่างเงื่อนไขการใช้งานโปรแกรมดังรูปที่ 1-2 ให้คลิกยอมรับและคลิกที่ปุ่ม Next



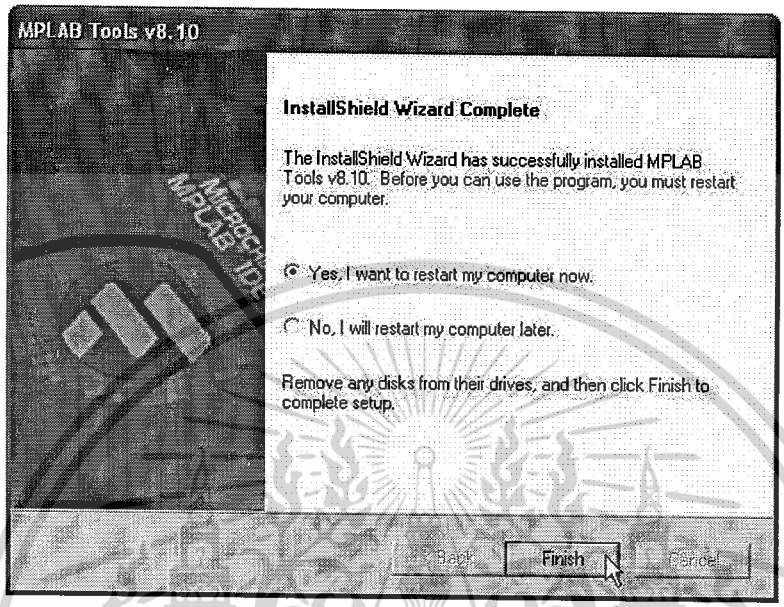
รูปที่ 1-2 หน้าต่างเงื่อนไขการใช้งานโปรแกรม

1.4 เลือกรูปแบบการติดตั้งโปรแกรมตามต้องการในที่นี้จะเลือกแบบ Compleate จากนั้นคลิกปุ่ม Next ดังรูปที่ 1-3



รูปที่ 1-3 แสดงหน้าต่างเลือกรูปแบบการติดตั้งโปรแกรม

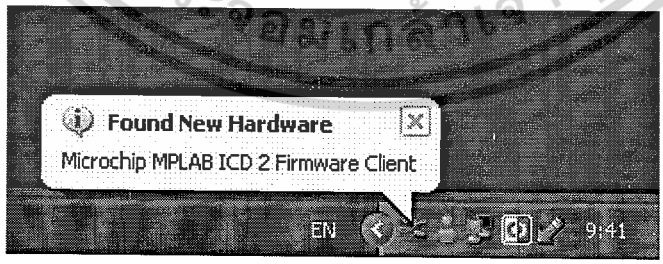
1.5 จากนั้นก็นำการติดตั้งต่อไปเรื่อยๆ ซึ่งก็เหมือนกับการติดตั้งซอฟแวร์บนวินโดว์ทั่วไป เมื่อการติดตั้งโปรแกรมเสร็จสมบูรณ์โปรแกรมจะถามว่าต้องการ Restart คอมพิวเตอร์ใหม่หรือไม่ให้คลิก Yes และคลิกปุ่ม Finish ดังรูปที่ 1-4 เป็นอันเสร็จสิ้นการติดตั้งโปรแกรม MPLAB IDE



รูปที่ 1-4 แสดงหน้าต่างเมื่อการติดตั้งโปรแกรมเสร็จสมบูรณ์

2. การติดตั้งไดร์เวอร์สำหรับ ET-ICDX V1.0

2.1 เมื่อได้ทำการติดตั้งโปรแกรม MPLAB IDE เรียบร้อยแล้วขั้นตอนต่อไปก็เป็นการติดตั้งไดร์เวอร์ของ ET-ICDX V1.0 ซึ่งสามารถทำได้โดย ทำการเชื่อมต่อสาย USB จากคอมพิวเตอร์เข้ากับ ET-ICDX V1.0 จากนั้นวินโดวส์จะตรวจสอบพบฮาร์ดแวร์ใหม่ดังรูปที่ 2-1

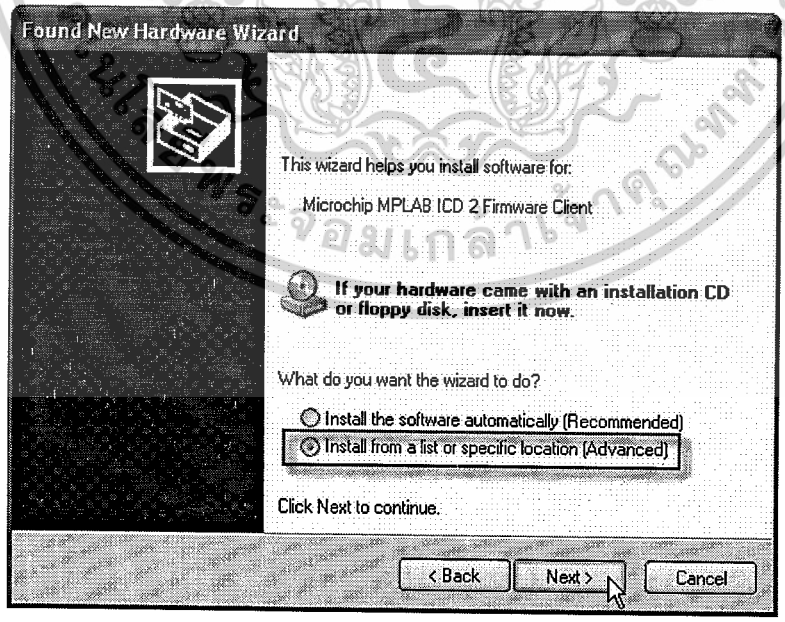


รูปที่ 2-1 แสดงเมื่อวินโดวส์ตรวจพบอาร์ดแวร์ใหม่

2.2 จากนั้นจะปรากฏหน้าต่าง Found New Hardware Wizard ให้เลือกที่ No,not this time และคลิกที่ปุ่ม Next ดังรูปที่ 2-2

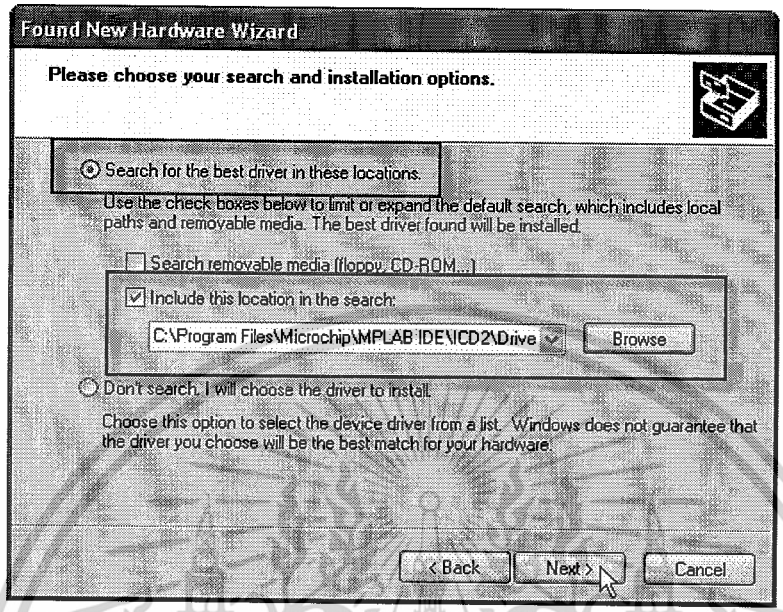
รูปที่ 2-2 แสดงหน้าต่าง Found New Hardware Wizard

2.3 จากนั้นจะปรากฏหน้าต่างเพื่อให้เลือกรูปแบบการติดตั้งไดร์เวอร์ให้เลือก Install from a list or Specific location (Advanced) และคลิกที่ปุ่ม Next ดังรูปที่ 2-3



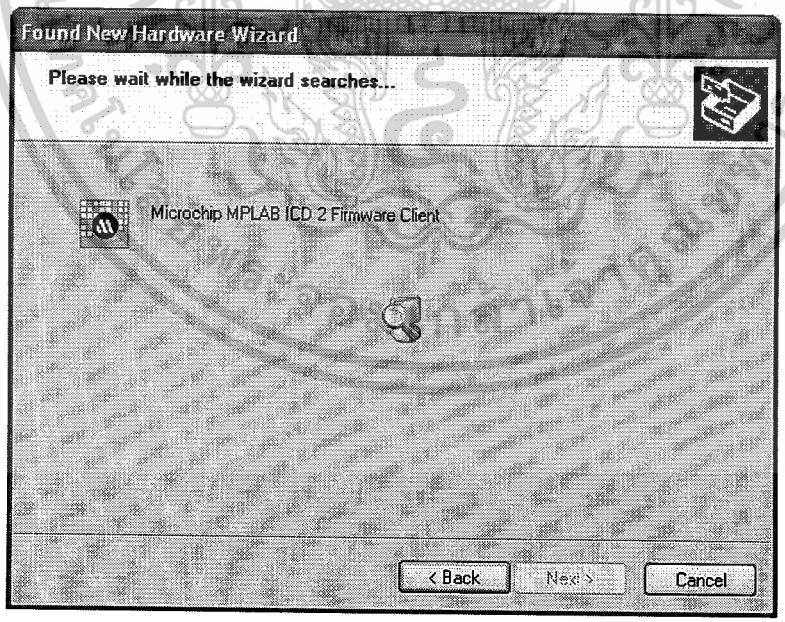
รูปที่ 2-3 แสดงหน้าต่างเพื่อให้เลือกรูปแบบการติดตั้งไดร์เวอร์

2.4 จากนั้นจะปรากฏหน้าต่างเพื่อให้ระบุตำแหน่งที่ตั้งของไดร์เวอร์ ให้ทำการคลิกที่ปุ่ม Browse และเลือกไปที่ *C:\Program Files\Microchip\MPLAB IDE\ICD2\Drivers* และคลิกปุ่ม Next ดังรูปที่ 2-4



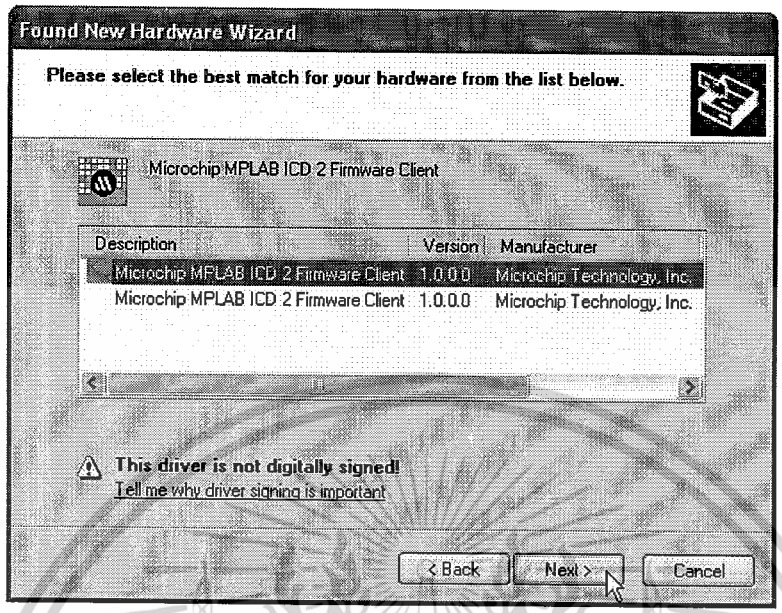
รูปที่ 2-4 แสดงหน้าต่างเพื่อให้ระบุตำแหน่งที่ตั้งของไดร์เวอร์

2.5 จากนั้นวินโดว์จะเริ่มต้นการค้นหาไดร์เวอร์ดังรูปที่ 2-5



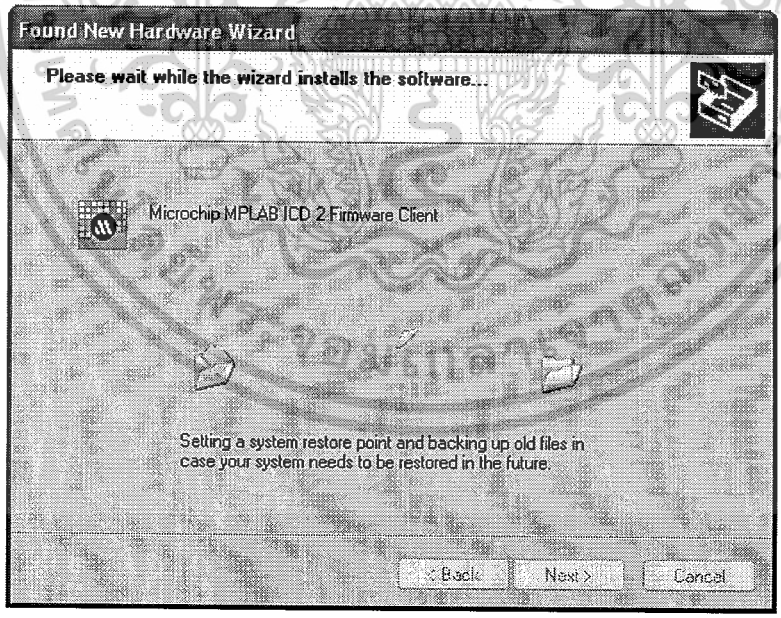
รูปที่ 2-5 แสดงหน้าต่างค้นหาไดร์เวอร์

2.6 เมื่อพบไดร์เวอร์แล้วให้คลิกปุ่ม Next ดังรูปที่ 2-6

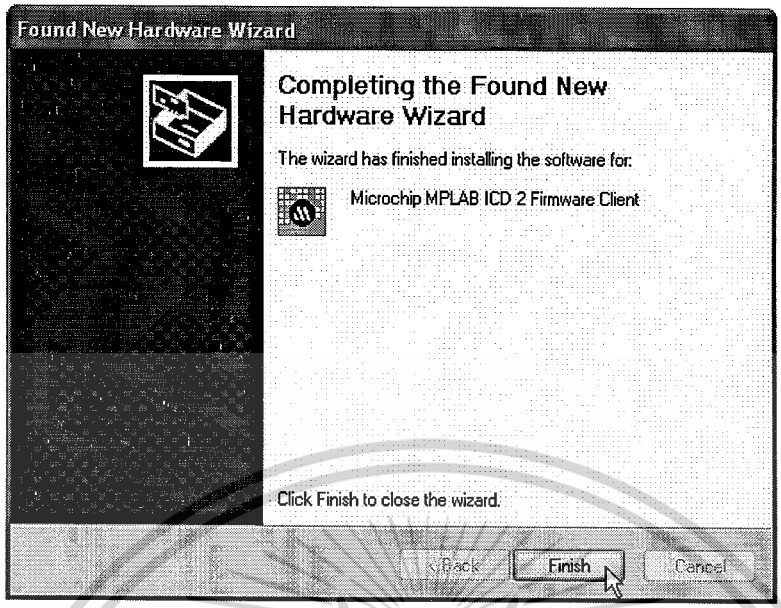


รูปที่ 2-6 หน้าต่างแสดงเมื่อพบไดร์เวอร์

2.7 จากนั้นวินโดว์จะเริ่มทำการติดตั้งไดร์เวอร์ดังรูปที่ 2-7 และเมื่อติดตั้งเสร็จเรียบร้อยจะปรากฏหน้าต่างดังรูปที่ 2-8 ให้คลกที่ปุ่ม Finish เพื่อเสร็จสิ้นการติดตั้งไดร์เวอร์

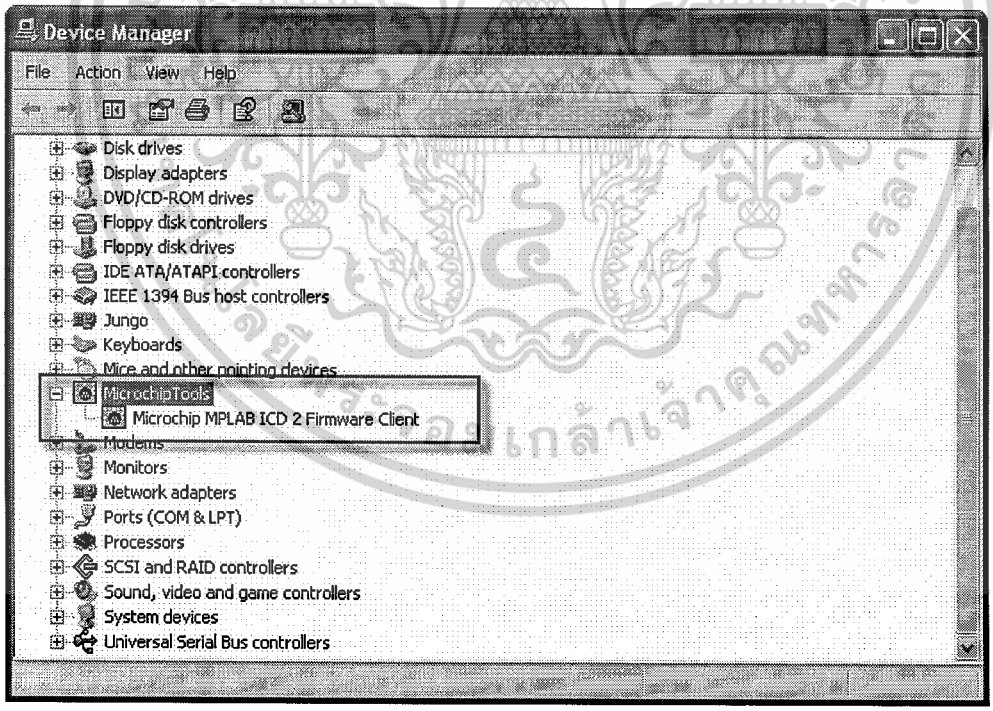


รูปที่ 2-7 แสดงหน้าต่างเมื่อวินโดว์เริ่มทำการติดตั้งไดร์เวอร์



รูปที่ 2-8 หน้าต่างแสดงเมื่อการติดตั้งไดร์เวอร์เสร็จสมบูรณ์

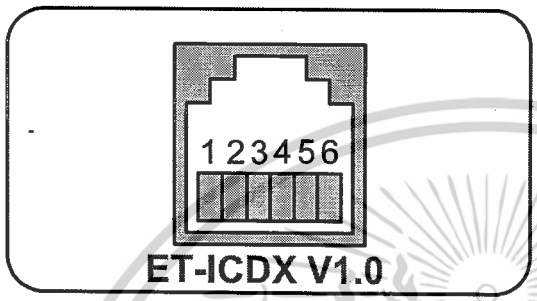
2.8 ถ้าการติดตั้งไดร์เวอร์สมบูรณ์ไม่มีข้อผิดพลาดใดๆ เมื่อเข้าไปดูที่ Device Manager ของวินโดว์จะพบชื่อของอุปกรณ์ *Microchip Tools* ดังรูปที่ 2-9



รูปที่ 2-9 แสดงหน้าต่าง Device Manager

3. การต่อ ET-ICDX V1.0 เข้ากับบอร์ดเป้าหมาย

ET-ICDX V1.0 สามารถจะต่อกับบอร์ดเป้าหมายที่มีแจ๊ก ICD2 หรือ ICSP (ใช้ร่วมกับ สาย ICD2 to ICSP) ที่มีการจัดเรียงขาตามมาตรฐาน Microchip ได้ทันที ซึ่งตำแหน่งและชื่อขาสัญญาณแสดงดังรูปที่ 3-1



ตำแหน่งขา	ชื่อสัญญาณ
1	MCLR/VPP
2	VDD
3	GND
4	PGD
5	PGC
6	NOT USED

รูปที่ 3-1 แสดงตำแหน่งและชื่อขาสัญญาณของ ET-ICDX V 1.0

โดยที่หน้าที่ของขาต่างๆ มีดังนี้

1. MCLR/VPP เป็นขาโปรแกรมแรงดันไฟสูงจะต่อกับ RESET หรือขา MCLR ของ MCU
2. VDD เป็นขาไฟเลี้ยงของ MCU ซึ่งจะต้องต่อขานี้กับไฟเลี้ยงของบอร์ดเป้าหมาย (3.3V,5V)
3. GND เป็นขากราวด์ ซึ่งจะต้องต่อกับกราวด์ของบอร์ดเป้าหมาย
4. PGD เป็นขา PROGRAM DATA ซึ่งจะต้องต่อกับขา PGD ของ MCU
5. PGC เป็นขา PROGRAM CLOCK ซึ่งจะต้องต่อกับขา PGC ของ MCU
6. NOT USED ขานี้จะเป็นขาว่างไม่ได้ใช้งาน

ภาคผนวก

ค

อัลกอริทึมในการเข้ารหัสข้อมูล

อัลกอริทึมในการเข้ารหัสข้อมูล

อัลกอริทึมในการเข้ารหัสข้อมูลมี 2 ประเภทหลัก คือ

- อัลกอริทึมแบบสมมาตร (Symmetric key algorithms)

อัลกอริทึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป อัลกอริทึมยังสามารถแบ่งย่อยออกเป็น 2 ประเภท ได้แก่ แบบบล็อก ซึ่งจะทำการเข้ารหัสทีละบล็อก 1 บล็อกประกอบด้วยหลายไบต์ เช่น 64 ไบต์ เป็นต้น) และแบบสตรีม ซึ่งจะทำการเข้ารหัสทีละไบต์อัลกอริทึมแบบนี้จะใช้กุญแจที่เรียกว่า กุญแจลับ ซึ่งมีเพียงหนึ่งเดียวเพื่อใช้ในการเข้าและถอดรหัสข้อความที่ส่งไป อัลกอริทึมยังสามารถแบ่งย่อยออกเป็น 2 ประเภท ได้แก่ แบบบล็อก ซึ่งจะทำการเข้ารหัสทีละบล็อค (1 บล็อค ประกอบด้วยหลายไบต์ เช่น 64 ไบต์ เป็นต้น) และแบบสตรีมซึ่งจะทำการเข้ารหัสทีละไบต์

- อัลกอริทึมแบบอสมมาตร (Asymmetric key algorithms)

อัลกอริทึมนี้จะใช้กุญแจสองตัวเพื่อทำงาน ตัวหนึ่งใช้ในการเข้ารหัสและอีกตัวหนึ่งใช้ในการถอดรหัสข้อมูลที่เข้ารหัสมาโดยกุญแจตัวแรก อัลกอริทึมกลุ่มสำคัญในแบบอสมมาตรนี้คือ อัลกอริทึมแบบกุญแจสาธารณะ ซึ่งใช้กุญแจที่เรียกกันว่า กุญแจสาธารณะ ในการเข้ารหัสและใช้กุญแจที่เรียกกันว่า กุญแจส่วนตัว ในการถอดรหัสข้อมูลนั้น กุญแจสาธารณะนี้สามารถส่งมอบให้กับผู้อื่นได้ เช่น เพื่อนร่วมงานที่เราต้องการติดต่อด้วย หรือแม้กระทั่งวางไว้บนเว็บไซต์ เพื่อให้ผู้อื่นสามารถดาวน์โหลดไปใช้งานได้ สำหรับกุญแจส่วนตัวนั้นต้องเก็บไว้กับผู้เป็นเจ้าของกุญแจส่วนตัวเท่านั้นและห้ามเปิดเผยให้ผู้อื่นทราบโดยเด็ดขาดอัลกอริทึมแบบกุญแจสาธารณะยังสามารถประยุกต์ใช้ได้กับการลงลายมือชื่ออิเล็กทรอนิกส์ ซึ่งเปรียบเสมือนการลงลายมือชื่อของเราที่ใช้กับเอกสารสำนักงานทั่วไป การลงลายมือชื่อนี้จะเป็นการพิสูจน์ความเป็นเจ้าของและสามารถใช้ได้กับการทำธุรกรรมต่างๆ บนอินเตอร์เน็ต เช่น การซื้อสินค้า เป็นต้น วิธีการใช้งานคือ ผู้เป็นเจ้าของกุญแจส่วนตัวลงลายมือชื่อของตนกับข้อความที่ต้องการส่งไปด้วยกุญแจส่วนตัว แล้วจึงส่งข้อความนั้นไปให้กับผู้รับ เมื่อได้รับข้อความที่ลงลายมือชื่อมา ผู้รับสามารถใช้กุญแจสาธารณะ ที่เป็นคู่ของกุญแจส่วนตัวนั้น เพื่อตรวจสอบว่าเป็นข้อความที่มาจากผู้ส่งนั้นหรือไม่

ปัญหาของอัลกอริทึมแบบสมมาตร

อัลกอริทึมแบบสมมาตรมีความสำคัญไม่ด้อยไปกว่าอัลกอริทึมแบบอสมมาตร ทั้งนี้เนื่องจากอัลกอริทึมแบบแรกทำงานได้รวดเร็วกว่าและง่ายต่อการใช้งานกว่าแบบหลัง อย่างไรก็ตามอัลกอริทึมแบบสมมาตรยังมีปัญหาที่สำคัญ 3 ประการ ซึ่งเป็นข้อจำกัดในการใช้งานอัลกอริทึมนี้

1. ในการใช้งานอัลกอริทึมนี้ สองกลุ่มที่ต้องการแลกเปลี่ยนข้อมูลกัน (เช่น องค์กร ก และ ข) จำเป็นต้องแลกเปลี่ยนกุญแจลับกันก่อน ซึ่งอาจหมายถึงส่งมอบกุญแจลับให้กับอีกกลุ่มหนึ่ง การแลกเปลี่ยนกุญแจนั้นอาจทำได้อย่างยุ่งยากและไม่สะดวก
2. ทั้งสองกลุ่มต้องรักษากุญแจลับนั้นไว้เป็นอย่างดี ห้ามเปิดเผยให้ผู้อื่นล่วงรู้โดยเด็ดขาด การที่กุญแจถูกเปิดเผยออกไปสู่ผู้อื่น จะโดยกลุ่มใดกลุ่มหนึ่งก็ตาม และอีกกลุ่มหนึ่งไม่ได้รับทราบ

ปัญหานี้ อาจก่อให้เกิดปัญหากับกลุ่มที่ไม่ทราบนี้ได้ เช่น กลุ่มนี้อาจส่งข้อความที่เป็นความลับไปให้กับอีกกลุ่มหนึ่ง แต่ข้อความนี้อาจถูกเปิดเผยได้โดยใช้กุญแจลับที่ล่วงรู้โดยผู้อื่น

3. สำหรับสองกลุ่มที่ต้องการติดต่อกัน จำเป็นต้องใช้กุญแจลับเป็นจำนวน 1 กุญแจเพื่อติดต่อกัน สมมติว่ามีผู้ที่ต้องติดต่อกันเป็นจำนวน n กลุ่ม จำนวนกุญแจลับทั้งหมดที่ต้องแลกเปลี่ยนกันคิดเป็นจำนวนทั้งหมด 2n หรือเท่ากับ n (n-1)/2 กุญแจ ซึ่งจะเห็นได้ว่าจำนวนกุญแจมีมากมายเกินไป ซึ่งอาจก่อให้เกิดปัญหาด้านการรักษาความปลอดภัยให้กับกุญแจเหล่านี้

อัลกอริทึมแบบกุญแจสาธารณะ ซึ่งเป็นแบบอสมมาตร ช่วยแก้ปัญหาเหล่านี้ได้ทั้งหมด ผู้ใช้ที่ถือกุญแจส่วนตัวและต้องการให้บุคคลอื่นที่ตนติดต่อด้วยส่งเอกสารหรือข้อความที่เข้ารหัสมาหาตน สามารถเผยแพร่กุญแจสาธารณะของตนไว้บนเว็บไซต์หรือในที่สาธารณะซึ่งผู้อื่นสามารถเข้ามาดาวน์โหลดไปใช้งานได้ วิธีการใช้งานคือให้บุคคลอื่นที่มาดาวน์โหลดกุญแจไปนั้นทำการเข้ารหัสข้อความที่ต้องการส่งด้วยกุญแจสาธารณะ แล้วจึงส่งข้อความที่เข้ารหัสไปให้กับผู้เป็นเจ้าของกุญแจสาธารณะ โดยวิธีนี้จะไม่มีผู้อื่นสามารถเปิดดูข้อความที่เข้ารหัสนั้นได้ยกเว้นผู้ที่ถือกุญแจส่วนตัว ที่เป็นคู่ของกุญแจสาธารณะนั้น จึงจะสามารถเปิดข้อความนี้ดูได้

การเผยแพร่กุญแจสาธารณะในสถานที่ต่างๆ ได้ทำให้ลดความยุ่งยากในการแลกเปลี่ยนกุญแจกันซึ่งเป็นปัญหาข้อแรกของการเข้ารหัสแบบสมมาตร สำหรับปัญหาที่ว่าทั้งสองกลุ่มจะต้องรักษากุญแจลับไว้เป็นอย่างดีนั้น วิธีการของกุญแจสาธารณะจะทำให้ผู้ที่ต้องรับผิดชอบเหลือเพียงผู้เดียว กล่าวคือ ผู้ถือกุญแจส่วนตัวซึ่งห้ามให้ผู้อื่นล่วงรู้โดยเด็ดขาด

สำหรับปัญหาที่สามที่ว่าจำนวนกุญแจลับที่จำเป็นต้องใช้มีมากมายเกินไป วิธีการของกุญแจสาธารณะจะใช้จำนวนกุญแจที่ประหยัดกว่า เนื่องจากกุญแจสาธารณะ 1 กุญแจของกลุ่มๆ หนึ่งจะสามารถเผยแพร่ให้กับกี่กลุ่มก็ได้ที่เราต้องการติดต่อด้วย แทนที่จะเป็น 1 กุญแจลับต่อสองกลุ่มที่ต้องการติดต่อกัน ดังนั้นถ้ามีกลุ่มที่ต้องติดต่อกันจำนวน n กลุ่ม จำนวนกุญแจส่วนตัวที่ต้องระวังรักษาก็คือ n กุญแจ ซึ่งจะเห็นได้ว่าลดลงไปได้เป็นจำนวนมาก

ข้อเสียที่สำคัญของระบบกุญแจสาธารณะที่สำคัญคือ ต้องใช้เวลาในการคำนวณการเข้าและถอดรหัสเมื่อเทียบกับระบบกุญแจสมมาตร และอาจใช้เวลาเป็นพันเท่าของเวลาที่ใช้โดยระบบกุญแจสมมาตร

ความแข็งแกร่งของอัลกอริทึมสำหรับการเข้ารหัส

ความแข็งแกร่งของอัลกอริทึม หมายถึง ความยากในการที่ผู้บุกรุกจะสามารถถอดรหัสข้อมูลได้โดยปราศจากกุญแจที่ใช้ในการเข้ารหัส ซึ่งจะขึ้นอยู่กับปัจจัยดังนี้

- ความยาวของกุญแจเข้ารหัส ปกติกุญแจเข้ารหัสจะมีความยาวเป็นบิต ยิ่งจำนวนบิตของกุญแจยิ่งมาก ยิ่งทำให้การเดาเพื่อสุ่มหากุญแจที่ถูกต้องเป็นไปได้ยากยิ่งขึ้น เช่น กุญแจขนาด 1 บิต จะสามารถแทนตัวเลขได้ 2 ค่าคือ 0 กับ 1 กุญแจขนาด 2 บิต จะเป็นไปได้ 4 ค่าคือ 0, 1, 2, 3 เป็นต้น

- การเก็บกุญแจเข้ารหัสไว้อย่างเป็นความลับ ผู้เป็นเจ้าของกุญแจลับหรือส่วนตัวต้อง ระมัดระวังไม่ให้กุญแจสูญหายหรือล่วงรู้โดยผู้อื่น
- การมีประตูลับในอัลกอริทึม อัลกอริทึมที่ดีต้องไม่แฝงไว้ด้วยประตูลับที่สามารถใช้เป็นทางเข้าไปสู่อัลกอริทึม แล้วอาจใช้เพื่อทำการถอดรหัสข้อมูลได้ ประตูลับนี้ทำให้ไม่จำเป็นต้องใช้กุญแจในการถอดรหัส
- ความไม่เกรงกลัวต่อการศึกษาหรือดูอัลกอริทึมเพื่อหารูปแบบของการเข้ารหัส อัลกอริทึมที่ดีต้องเปิดให้ผู้รู้ทำการศึกษาในรายละเอียดได้โดยไม่เกรงว่าผู้ศึกษาจะสามารถจับรูปแบบของการเข้ารหัสได้
- ความไม่เกรงกลัวต่อปัญหาการหาความสัมพันธ์ในข้อมูลที่ได้รับ กล่าวคือเมื่อผู้บุกรุกทราบข้อมูลบางอย่างที่เป็นข้อมูลตั้งต้นซึ่งยังไม่ได้เข้ารหัส รวมทั้งมีข้อมูลที่เข้ารหัสแล้ว ของข้อมูลตั้งต้นนั้น ผู้บุกรุกอาจจะสามารถหาความสัมพันธ์ระหว่างข้อความทั้งสองนั้นได้ ซึ่งจะเป็นวิธีการในการถอดรหัสข้อมูลได้ ปัญหานี้เรียกกันว่า Known plaintext attack คำว่า plaintext หมายถึงข้อความตั้งต้นที่ยังไม่ได้ผ่านการเข้ารหัส
- คุณสมบัติของข้อความตั้งต้น คุณสมบัตินี้อาจใช้เป็นช่องทางในการถอดรหัสข้อมูลได้ อัลกอริทึมที่ดีต้องไม่ใช้คุณสมบัติของข้อความเป็นกลไกในการเข้ารหัสข้อมูล

คำแนะนำในการเลือกใช้อัลกอริทึมคือให้ใช้อัลกอริทึมที่ได้มีการใช้งานมาเป็นระยะ เวลานานแล้ว ทั้งนี้เนื่องจากหากปัญหาของอัลกอริทึมนี้มีจริง ก็คงเกิดขึ้นมานาน แล้วและก็คงเป็นที่ทราบกันแล้ว นั่นคืออย่างน้อยที่สุดจวบจนกระทั่งถึงปัจจุบัน ก็ยังไม่มีการบุกรุกที่ทำให้อัลกอริทึมนั้นไม่สามารถใช้งานได้อย่างปลอดภัยเป็นที่ประจักษ์ ดังนั้นจึงไม่ควรใช้อัลกอริทึมใหม่ๆ ที่เพิ่งได้มีการนำเสนอกันสู่สาธารณะ เพราะอาจมีช่องโหว่แฝงอยู่และยังไม่เป็นที่ทราบในขณะนี้

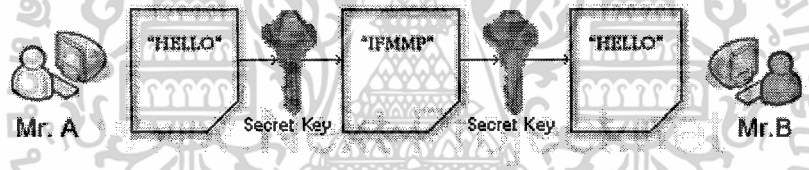
ความยาวของกุญแจที่ใช้ในการเข้ารหัส

ความยาวของกุญแจเข้ารหัสมีหน่วยนับเป็นบิต หนึ่งบิตในคอมพิวเตอร์เป็นตัวเลขฐานสองที่ประกอบด้วยค่า 0 และ 1 กุญแจที่มีความยาว 1 บิต ตัวเลขที่เป็นไปได้เพื่อแทนกุญแจนั้น จึงอาจมีค่าเป็น 0 หรือ 1 กุญแจที่มีความยาว 2 บิต ตัวเลขที่เป็นไปได้จึงเป็น 0, 1, 2 และ 3 ตามลำดับ กุญแจที่มีความยาว 3 บิต ตัวเลขที่เป็นไปได้จะอยู่ระหว่าง 0 ถึง 7 ดังนั้นเมื่อเพิ่มความยาวของกุญแจทุกๆ 1 บิต ค่าที่เป็นไปได้ของกุญแจจะเพิ่มขึ้นเป็นสองเท่าตัว หรือจำนวนกุญแจที่เป็นไปได้จะเพิ่มขึ้นเป็น 2 เท่าตัวนั่นเอง ฉะนั้นจะเห็นได้ว่ากุญแจยิ่งมีความยาวมาก โอกาสที่ผู้บุกรุกจะสามารถคาดเดากุญแจที่ตรงกับหมายเลขที่ถูกต้องของกุญแจจะยิ่งยากมากขึ้นตามลำดับ ในการที่ผู้บุกรุกลองผิดลองถูกกับกุญแจโดยใช้กุญแจที่มีหมายเลขต่างๆ กัน เพื่อหวังที่จะพบกุญแจที่ถูกต้องและสามารถใช้ถอดรหัสข้อมูลได้ การลองผิดลองถูกนี้เราเรียกกันว่า Key search หรือการค้นหากุญแจนั่นเอง ทฤษฎีได้กล่าวไว้ว่าการลองผิดลองถูกนี้โดยเฉลี่ยจะต้องทดลองกับกุญแจเป็นจำนวนครึ่งหนึ่งของกุญแจทั้งหมดก่อนที่จะพบกุญแจที่ถูกต้องความยาวของกุญแจที่มีขนาดเหมาะสมจึงขึ้นอยู่กับความเร็วในการค้นหากุญแจของผู้บุกรุกและระยะเวลาที่ต้องการให้ข้อมูลมีความปลอดภัย ตัวอย่างเช่น ถ้าผู้บุก

รุกสามารถลองผิดลองถูกกับกุญแจเป็นจำนวน 10 กุญแจภายในหนึ่งวินาทีแล้ว กุญแจที่มีความยาว 40 บิต จะสามารถป้องกันข้อมูลไว้ได้ 3,484 ปี ถ้าผู้บุกรุกสามารถลองได้เป็นจำนวน 1 ล้านกุญแจในหนึ่งวินาที เทคโนโลยีปัจจุบันสามารถทำได้ กุญแจที่มีความยาว 40 บิตจะสามารถป้องกันข้อมูลไว้ได้เพียง 13 วันเท่านั้น ซึ่งอาจไม่เพียงพอสำหรับในบางลักษณะงาน ด้วยเทคโนโลยีในปัจจุบันหากผู้บุกรุกสามารถทดลองได้เป็นจำนวน 1,000 ล้านกุญแจในหนึ่งวินาที กุญแจขนาด 128 บิตจะสามารถป้องกันข้อมูลไว้ได้ 1022 ปี ดังนั้นด้วยลักษณะงานทั่วไปกุญแจขนาด 128 บิตจะพอเพียงต่อการรักษาความลับของข้อมูลเอาไว้ได้

อัลกอริทึมสำหรับการเข้ารหัสแบบสมมาตร

ระบบเข้ารหัสแบบกุญแจสมมาตร (Symmetric-key cryptography) คือการเข้ารหัสข้อมูลด้วยกุญแจเดี่ยวทั้งผู้ส่งและผู้รับ โดยวิธีการนี้ผู้รับกับผู้ส่งต้องตกลงกันก่อนว่าจะใช้รูปแบบไหนในการเข้ารหัสข้อมูล ซึ่งรูปแบบไหนในการเข้ารหัสข้อมูลที่ผู้รับกับผู้ส่งตกลงกันแท้ที่จริงก็คือ กุญแจลับ นั่นเอง เช่น ผู้ส่งกับผู้รับตกลงจะใช้เทคนิคการแทนที่ตัวอักษรที่อยู่ถัดไป 1 ตำแหน่ง เช่น ถ้าเห็นตัวอักษร A ก็ให้เปลี่ยนไปเป็น B หรือเห็นตัวอักษร B ก็ให้เปลี่ยนไปเป็น C เป็นต้น นั้นก็คือผู้ส่งกับผู้รับตกลงใช้รูปแบบนี้เป็นกุญแจลับดูตัวอย่างดังรูป



การเข้ารหัสแบบกุญแจสมมาตร (Symmetric-key cryptography)

จากรูป ถ้า Mr. A ได้ตกลงกับ Mr. B ว่ากุญแจลับที่จะใช้เข้ารหัสและถอดรหัสคือ การเปลี่ยนตัวอักษรจากเดิมถัดไป1ตำแหน่ง ถ้า Mr. A ต้องการส่งคำว่า HELLO ไปให้ Mr. B ขั้นตอนจะเป็นดังนี้

1. Mr. A สร้างข้อความว่า "HELLO" ขึ้นมา
2. Mr. A ใช้กุญแจลับมาทำการเข้ารหัสข้อความ โดยการเปลี่ยนตัวอักษรจากเดิมถัดไป 1 ตำแหน่ง ดังนั้น ตัวอักษร H จะเปลี่ยนไปเป็นตัวอักษร E จะเปลี่ยนไปเป็นตัวอักษร F ตัวอักษร L ทั้ง 2 ตัว จะเปลี่ยนไปเป็นตัวอักษร M ทั้งสองตัว และสุดท้ายตัวอักษร O จะเปลี่ยนไปเป็นตัวอักษร P เพราะฉะนั้นหลังจากการทำการเข้ารหัสข้อความที่ Mr. A ต้องการส่งด้วยกุญแจลับแล้ว ข้อความว่า HELLO จะเปลี่ยนไปเป็นข้อความที่เข้ารหัส ว่า IFMMP
3. Mr. A ส่งข้อความที่เข้ารหัสไปให้ Mr. B
4. หลังจากที่ Mr. B ได้รับข้อความที่เข้ารหัสจาก Mr. A แล้ว Mr. B จะต้องทำการถอดรหัสข้อครามนี้ก่อน หรือที่เรียกว่า Decrypt โดยการถอดรหัสข้อความนี้ Mr. B จะต้องใช้กุญแจลับที่ได้ตกลงกันไว้แล้วกับ Mr. A มาทำการถอดรหัส เพราะฉะนั้นกุญแจลับที่ได้ตกลงกันกับ Mr. A ว่า

Mr. A จะทำการเข้ารหัสโดยการเปลี่ยนตัวอักษรจากเดิมไปเป็นตัวอักษรที่อยู่ถัดไป 1 ตำแหน่ง ดังนั้น Mr. B จะต้องเอาข้อความเข้ารหัส IFMMP มาถอดรหัส โดยการเปลี่ยนจากตัวอักษร I ไปเป็นตัวอักษร H และตัวอักษร F จะไปเป็นตัวอักษร E และตัวอักษร M ทั้งสองตัวจะเปลี่ยนไปเป็นตัวอักษร O หลังจากนั้น Mr. B ก็จะทราบว่าข้อความที่ Mr. A ส่งมา คือ ข้อความว่า "HELLO"

จากตัวอย่างที่ได้อธิบายมานี้จะเป็นหลักการแบบง่ายๆ ทำให้เห็นการทำงานของการเข้ารหัสแบบสมมาตร หรือกุญแจเดี่ยว เพราะฉะนั้นหลักการเข้ารหัสแบบสมมาตรนี้จะใช้กุญแจลับ (Secret Key) ทำการเข้ารหัสและถอดรหัสข้อความ

ข้อดีของการเข้ารหัสแบบสมมาตร

1. การเข้ารหัสและถอดรหัสข้อมูลใช้เวลาน้อย เพราะว่าอัลกอริทึมที่ใช้ไม่ได้สลับซับซ้อน
2. ขนาดของข้อมูลหลังจากทำการเข้ารหัสแล้ว มีการเปลี่ยนแปลงไม่มาก หรือพูดอีกนัยหนึ่งว่าข้อมูลหลังจากทำการเข้ารหัสแล้ว จะมีขนาดไม่ใหญ่ไปกว่าเดิมมากนัก

ข้อด้อยของการเข้ารหัสแบบสมมาตร

1. การจัดการกับกุญแจลับที่ยุ่งยาก เพราะ Mr. A ต้องจำให้ได้ด้วยว่า ถ้าจะติดต่อกับ Mr. B ต้องใช้กุญแจลับดอกไหน หรือติดต่อกับนายขาวต้องใช้กุญแจลับดอกไหน
2. การกระจายกุญแจลับ เนื่องจากการเข้ารหัสวิธีนี้ต้องใช้กุญแจลับ 1 ดอกต่อผู้รับ 1 คน ดังนั้นถ้า Mr. A ต้องติดต่อกับคนมากๆ Mr. A ก็ต้องส่งกุญแจลับที่ใช้ไปให้กับทุกคน

สำหรับวิธีการเข้ารหัสแบบนี้ ก็จะมีมาตรฐานมารองรับเหมือนกัน มาตรฐานที่ว่าก็คือ มาตรฐาน DES (Digital Encryption Standard) หรือเรียกว่า “เดส” ที่มาของ DES เกิดขึ้นมาจากทีมพัฒนาของบริษัท IBM เมื่อราวๆปลายยุค 1960 ทำการพัฒนาระบบเข้ารหัสและถอดรหัสนี้ โดยหลักการทำงานจะทำการแบ่งข้อมูลที่จะทำการเข้าหรือถอดรหัสออกเป็นบล็อก โดยที่แต่ละบล็อกจะทีขนาด 64 บิต และจำนวนความยาวของกุญแจลับจะมีขนาด 128 บิตในช่วงแรก หลังจากนั้นทางบริษัท IBM ก็ได้เพิ่มทุนให้ทำการพัฒนาและปรับปรุงต่อเรื่อยๆมา โดยในครั้งนี้ได้มีที่ปรึกษาจากสำนักงานความมั่นคงแห่งชาติ ของสหรัฐอเมริกาเข้าร่วมด้วย ผลที่ได้จากการพัฒนานี้ ทำให้ระบบ DES สามารถทนทานต่อผู้ต้องการเจาะรหัสได้ และขณะเดียวกัน ก็ได้ทำการลดความยาวของกุญแจลงเหลือแค่ 56 บิต จากเดิม 128 บิต เหตุผลที่ต้องลดความยาวของกุญแจลับลง ก็เพราะว่าสำนักงานความมั่นคงแห่งชาติของสหรัฐอเมริกา เกรงว่าจะไม่สามารถตรวจสอบข้อมูลที่เข้ารหัสด้วยความยาวของกุญแจลับที่ 128 บิตได้ ซึ่งการลดความยาวของกุญแจลับก็โดนกระแสต่อต้านจากกลุ่มธุรกิจองค์กรต่างๆ มากมาย เพราะพวกกลุ่มธุรกิจองค์กรต่างๆ เหล่านี้ต้องการให้ข้อมูลมีความลับมากๆ เพราะยิ่งกุญแจลับมีความยาวมากเท่าไหร่ ข้อมูลที่เข้ารหัสก็ยิ่งต้องใช้เวลาในการถอดรหัสออกนานมากขึ้น ทำให้ข้อมูลมีความปลอดภัยมากขึ้นอีก แต่รัฐบาลสหรัฐก็ออกมากล่าวว่า ด้วยความยาวกุญแจลับขนาด 56 บิตนี้ ก็ทำให้ต้องใช้เวลาในการถอดรหัสนานมากทีเดียว แต่ในปัจจุบันนี้ มีเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง สามารถที่จะถอดรหัสที่ใช้กุญแจขนาด 56 บิตได้ในเวลาแค่ 56 ชั่วโมง และมีแนวโน้มว่าจะสามารถถอดรหัสโดยใช้

เวลาลดลงกว่านี้ได้อีก แต่สำหรับข้อมูลที่เข้ารหัสด้วยกุญแจขนาด 128 บิต ในปัจจุบันยังถือว่าปลอดภัยอยู่มาก เพราะว่ายังไม่สามารถถอดรหัสได้เร็วเกินที่จะรอคอยได้ เพราะกว่าจะถอดรหัสได้ ข้อมูลเหล่านั้นก็อาจจะไม่มีประโยชน์ต่อการนำกลับไปใช้งานได้อีกแล้ว ซึ่งในปัจจุบันนี้ก็ได้มีมาตรฐานที่เรียกว่า 3DES เกิดขึ้นมาแล้ว โดยมาตรฐานนี้จะใช้กุญแจลับที่มีขนาดความยาว 168 บิต แต่สำหรับธุรกิจองค์กรใดที่จะใช้มาตรฐานนี้จะต้องทำเรื่องขออนุญาตจากรัฐบาลสหรัฐอเมริกาก่อน ถ้าได้รับอนุญาตจากรัฐบาลสหรัฐอเมริกาจึงจะสามารถนำมาใช้งานได้

อัลกอริทึมสำหรับการเข้ารหัสแบบสมมาตรในปัจจุบันมีเป็นจำนวนมาก ข้างล่างนี้จะนำเสนอเพียงจำนวนหนึ่งซึ่งเป็นอัลกอริทึมที่เป็นที่รู้จักกันดีในวงการของการเข้ารหัสข้อมูล

- อัลกอริทึม DES

อัลกอริทึม DES ย่อมาจาก Data Encryption Standard อัลกอริทึมนี้ได้รับการรับรองโดยรัฐบาลสหรัฐอเมริกาในปี ค.ศ. 1977 ให้เป็นมาตรฐานการเข้ารหัสข้อมูลสำหรับหน่วยงานของรัฐทั้งหมด ในปี 1981 อัลกอริทึมยังได้รับการกำหนดให้เป็นมาตรฐานการเข้ารหัสข้อมูลในระดับนานาชาติตามมาตรฐาน ANSI (American National Standards) อีกด้วย DES เป็นอัลกอริทึมแบบบล็อกซึ่งใช้กุญแจที่มีขนาดความยาว 56 บิต และเป็นอัลกอริทึมที่มีความแข็งแกร่ง แต่เนื่องด้วยขนาดความยาวของกุญแจที่มีขนาดเพียง 56 บิต ซึ่งในปัจจุบันถือได้ว่าสั้นเกินไป ผู้บุกรุกอาจใช้วิธีการลองผิดลองถูกเพื่อค้นหากุญแจที่ถูกต้องสำหรับการถอดรหัสได้ ในปี 1998 ได้มีการสร้างเครื่องคอมพิวเตอร์พิเศษขึ้นมาซึ่งมีมูลค่า 250,000 เหรียญสหรัฐ เพื่อใช้ในการค้นหากุญแจที่ถูกต้องของการเข้ารหัสข้อมูลหนึ่งๆ ด้วย DES และพบว่าเครื่องคอมพิวเตอร์นี้สามารถค้นหากุญแจที่ถูกต้องได้ภายในระยะเวลาไม่ถึงหนึ่งวัน

- อัลกอริทึม Triple-DES

อัลกอริทึม Triple-DES เป็นอัลกอริทึมที่เสริมความปลอดภัยของ DES ให้มีความแข็งแกร่งมากขึ้นโดยใช้อัลกอริทึม DES เป็นจำนวนสามครั้งเพื่อทำการเข้ารหัส แต่ละครั้งจะใช้กุญแจในการเข้ารหัสที่แตกต่างกัน ดังนั้นจึงเปรียบเสมือนการใช้กุญแจเข้ารหัสที่มีความยาวเท่ากับ 56*3 = 168 บิต Triple-DES ได้ถูก ใช้งานกับสถาบันทางการเงินอย่างแพร่หลาย รวมทั้งใช้งานกับโปรแกรม Secure Shell (ssh) ด้วยการใช้อัลกอริทึม DES เพื่อเข้ารหัสเป็นจำนวนสองครั้งด้วยกุญแจสองตัว (56*2=112 บิต) ยังถือได้ว่าไม่ปลอดภัยอย่างพอเพียง

- อัลกอริทึม Blowfish

อัลกอริทึม Blowfish เป็นอัลกอริทึมที่มีความรวดเร็วในการทำงาน มีขนาดเล็กกะทัดรัด และใช้การเข้ารหัสแบบบล็อก ผู้พัฒนาคือ Bruce Schneier อัลกอริทึมสามารถใช้กุญแจที่มีขนาดความยาวตั้งแต่ไม่มากนักไปจนถึงขนาด 448 บิต ซึ่งทำให้เกิดความยืดหยุ่นสูงในการเลือกใช้กุญแจ รวมทั้งอัลกอริทึมยังได้รับการออกแบบมาให้ทำงานอย่างเหมาะสมกับหน่วยประมวลผลขนาด 32 หรือ 64 บิต Blowfish ได้เปิดเผยสู่สาธารณะและไม่ได้มีการจดสิทธิบัตรใดๆ นอกจากนั้นยังใช้ในโปรแกรม SSH และอื่นๆ

- อัลกอริทึม IDEA

อัลกอริทึม IDEA ย่อมาจาก International Data Encryption Algorithm อัลกอริทึมนี้ได้รับการพัฒนาในประเทศสวิสเซอร์แลนด์ที่เมือง Zarich โดย James L. Massey และ Xuejia Lai และได้รับการตีพิมพ์เผยแพร่ในปี ค.ศ. 1990 อัลกอริทึมใช้กุญแจที่มีขนาด 128 บิต และได้รับการใช้งานกับโปรแกรมยอดฮิตสำหรับการเข้ารหัสและลงลายมือชื่ออิเล็กทรอนิกส์ในระบบอีเมล์ที่มีชื่อว่า PGP ต่อมา IDEA ได้รับการจดสิทธิบัตรทางด้านซอฟต์แวร์โดยบริษัท Ascom-Tech AG ในประเทศSwitzerland ซึ่งทำให้การนำไปใช้ในงานต่างๆ เริ่มลดลง ทั้งนี้เนื่องจากติดปัญหาเรื่องลิขสิทธิ์นั่นเอง

- อัลกอริทึม RC4

อัลกอริทึมนี้เป็นอัลกอริทึมแบบสตรีม ทำงานกับข้อมูลทีละไบต์ ซึ่งได้รับการพัฒนาขึ้นมาโดย Ronald Riverst และถูกเก็บเป็นความลับทางการค้าโดยบริษัท RSA Data Security ในภายหลังอัลกอริทึมนี้ได้รับการเปิดเผยใน Usenet เมื่อปี ค.ศ. 1994 และเป็นที่ทราบกันว่าเป็นอัลกอริทึมที่มีความแข็งแกร่งโดยสามารถใช้ขนาดความยาวของกุญแจที่มีขนาดตั้งแต่ 1 บิตไปจนกระทั่งถึงขนาด 2048 บิต

- อัลกอริทึม Rijndael หรืออัลกอริทึม AES

อัลกอริทึมนี้ได้รับการพัฒนาโดย Joan Daemen และ Vincent Rijmen ในปี 2000 อัลกอริทึมได้รับการคัดเลือกโดยหน่วยงาน National Institute of Standard and Technology (NIST) ของสหรัฐอเมริกาให้เป็นมาตรฐานในการเข้ารหัสชั้นสูงของประเทศ อัลกอริทึมมีความเร็วสูงและมีขนาดกะทัดรัดโดยสามารถใช้กุญแจที่มีความยาวขนาด 128, 192 และ 256 บิต

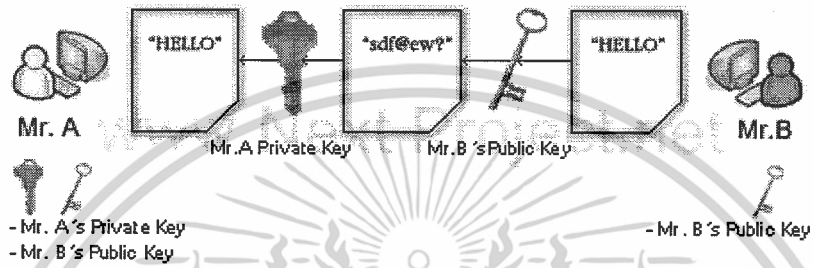
- อัลกอริทึม One-time Pads

อัลกอริทึมนี้ได้รับการยอมรับว่าเป็นอัลกอริทึมที่ไม่มีใครสามารถเจาะความแข็งแกร่งของอัลกอริทึมได้ อัลกอริทึมใช้กุญแจที่มีขนาดความยาวซึ่งอาจจะมากกว่าขนาดความยาวของข้อความที่ต้องการเข้ารหัส กุญแจจะถูกสร้างออกมาแบบสุ่มและโดยปกติจะถูกใช้งานแค่เพียงครั้งเดียวแล้วทิ้งไป แต่ละไบต์ของข้อความที่ต้องการส่งไปจะถูกเข้าและถอดรหัสโดยหนึ่งไบต์ ชนิดไบต์ต่อไบต์ ของกุญแจที่ถูกสร้างขึ้นมาใช้งาน เนื่องจากกุญแจที่ถูกใช้งานแต่ละครั้งจะไม่ซ้ำกันและถูกสร้างขึ้นมาแบบสุ่ม จึงเป็นการยากที่จะค้นหากุญแจที่ถูกต้องได้ข้อจำกัดของอัลกอริทึมนี้ คือขนาดของกุญแจที่อาจมีขนาดยาวกว่าข้อความที่ต้องการส่ง ซึ่งส่งผลให้การส่งมอบกุญแจที่มีขนาดใหญ่ทำได้ไม่สะดวกนัก รวมทั้งการสร้างกุญแจให้มีความสุ่มสูงไม่ใช่เป็นสิ่งที่ทำได้ง่ายนัก อย่างไรก็ตามอัลกอริทึมนี้ก็ยังมีการใช้งานในระบบเครือข่ายที่ต้องการความปลอดภัยสูง

- อัลกอริทึมสำหรับการเข้ารหัสแบบกุญแจสาธารณะ หรือการเข้ารหัสแบบอสมมาตร

ระบบเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric-key cryptography or Public Key Technology) ระบบการเข้ารหัสแบบนี้ได้ถูกคิดค้นโดย Whitfield Diffe ซึ่งเป็นนักวิจัยแห่งมหาวิทยาลัย Stanford สหรัฐอเมริกา ในปี พ.ศ. 2518 โดยการเข้ารหัสแบบนี้จะใช้หลักกุญแจคู่ทำการเข้ารหัสและถอดรหัส

โดยกุญแจคู่ที่ว่านี้จะประกอบไปด้วย กุญแจส่วนตัว และกุญแจสาธารณะ โดยหลักการทำงานจะทำดังนี้ ถ้าใช้กุญแจลูกใดเข้ารหัส ก็ต้องใช้กุญแจอีกลูกหนึ่งถอดรหัส สำหรับการเข้ารหัสและถอดรหัสด้วยกุญแจคู่นี้จะใช้ฟังก์ชั่นทางคณิตศาสตร์เข้ามาช่วยโดยที่ฟังก์ชันทางคณิตศาสตร์ที่นำมาใช้ ได้รับการพิสูจน์แล้วว่าจะมีเฉพาะกุญแจคู่ของมันเท่านั้นที่จะสามารถถอดรหัสได้ ไม่สามารถนำกุญแจคู่อื่นมาถอดรหัสได้อย่างเด็ดขาด



การเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric-key cryptography or Public Key Technology)

จากรูปการเข้ารหัสแบบนี้จะมี Mr. A คนเดียวที่อ่านได้ จะมีขั้นตอนดังนี้

1. Mr. A ต้องมีกุญแจคู่1คู่ขึ้นมาก่อน คือ กุญแจส่วนตัวกับกุญแจสาธารณะ โดยที่กุญแจสาธารณะของ Mr. A นี้ ใครๆก็สามารถที่จัดหามาได้รวมถึง Mr. B ด้วย หรือ Mr. A ส่งกุญแจนี้ไปให้ Mr. B ก่อน
2. หลังจาก Mr. B มีกุญแจสาธารณะของ Mr. A Mr. B จะใช้กุญแจสาธารณะของ Mr. A เข้ารหัสข้อความที่ต้องการจะส่ง
3. Mr. B ส่งข้อความเข้ารหัสไปให้ Mr. A
4. Mr. A ได้รับข้อความเข้ารหัสจาก Mr. B Mr. A จะต้องใช้กุญแจส่วนตัว นำมาใช้ในการถอดข้อความเข้ารหัสของ Mr. B หลังจากนั้น Mr. A จึงสามารถอ่านข้อความเข้ารหัสจาก Mr. B ได้ หรือในทางกลับกันถ้า Mr. B ต้องการส่งข้อความลับให้กับ Mr. A Mr. B ก็แค่ใช้กุญแจสาธารณะของ Mr. A ทำการเข้ารหัสข้อมูลแล้วส่งไปให้ Mr. A พอได้ข้อความเข้ารหัสจาก Mr. B Mr. A ก็จะใช้กุญแจส่วนตัวของตัวเองถอดรหัสข้อความลับจาก Mr. B เพราะฉะนั้นจะมีแต่ Mr. A เท่านั้นที่สามารถอ่านข้อความลับที่ถูกส่งมาจาก Mr. B ได้ แต่จะทำอย่างไรให้แนวคิดของ Whitfield Diffe นำมาประยุกต์ใช้งานได้จริงในโลกของข้อมูลอิเล็กทรอนิกส์ ดังนั้นจึงมีอัศวินสามนายขี่ม้าขาวมาช่วย Whitfield Diffe โดยอัศวินทั้งสามทำการค้นคว้าและวิจัยอยู่ที่Massachusetts Institute of Technology นักวิจัยทั้งสามก็คือ Ronald Rivest , Adi Shamir และ Leonard Adleman ในปี พ.ศ.2520 และตีพิมพ์เผยแพร่เป็นครั้งแรกในปี พ.ศ. 2521 ดังนั้นเราจึงเรียกฟังก์ชั่นที่ทั้งสามค้นพบนี้ตามอักษรแรกของชื่อ

นักวิจัยทั้งสามนี้ว่า ฟังก์ชันRSA แต่โดยทั่วไป มักจะนิยมเรียกว่าอัลกอริทึมRSA สำหรับการทำงานของอัลกอริทึมนี้

ข้อดีของระบบเข้ารหัสแบบกุญแจอสมมาตร

1. การจัดการกับกุญแจทำได้ง่าย เพราะว่า Mr. A ไม่ต้องจำเลยว่าได้ใช้กุญแจคู่ไหนกับใคร Mr. A แค่ใช้กุญแจส่วนตัวของตัวเองทำการถอดรหัสข้อมูลที่ Mr. B ส่งมาให้ หรือเอากุญแจส่วนตัวเข้ารหัสส่งไปให้ Mr. B Mr. B ก็สามารถที่จะอ่านได้ ซึ่งวิธีนี้จะง่ายมากครับ เพราะ Mr. A ใช้แค่กุญแจส่วนตัวของตัวเองดอกเดียวก็สามารถติดต่อกับ Mr. B หรือใครๆก็ได้ตามต้องการ
2. การกระจายกุญแจลับ เนื่องจากการเข้ารหัสโดยวิธีนี้ ใช้แค่กุญแจสาธารณะเพียงดอกเดียวในการเข้ารหัสและถอดรหัส และกุญแจสาธารณะของ Mr. A ก็สามารถที่จะเปิดเผยให้กับใครก็ได้ที่ต้องการจะติดต่อด้วย ไม่ว่าจะเป็น Mr. B นายขาว เหล่านี้เป็นต้น เพราะฉะนั้นการแจกจ่ายกุญแจสาธารณะของ Mr. A ไปให้กับคนสักพันคน หรือหมื่นคน จะไม่เป็นปัญหาอีกต่อไป

ข้อด้อยของระบบเข้ารหัสแบบกุญแจอสมมาตร

1. การเข้ารหัสและถอดรหัสข้อมูลใช้เวลามาก เพราะว่าอัลกอริทึมที่ใช้ค่อนข้างจะสลับซับซ้อนมาก
2. ขนาดของข้อมูลหลังจากทำการเข้ารหัสแล้ว มีการเปลี่ยนแปลงมาก หรือพูดอีกนัยหนึ่งว่า ข้อมูลหลังจากทำการเข้ารหัสแล้ว จะมีขนาดใหญ่กว่าเดิมมากขึ้น เพราะฉะนั้นจะเป็นปัญหาในการใช้งาน

อัลกอริทึมแบบกุญแจสาธารณะ แบ่งตามลักษณะการใช้งานได้เป็น 2 ประเภท คือ

1. ใช้สำหรับการเข้ารหัส
2. ใช้สำหรับการลงลายมือชื่ออิเล็กทรอนิกส์

อัลกอริทึมที่เป็นที่รู้จักกันดีมีดังนี้

- อัลกอริทึม RSA

อัลกอริทึม RSA ได้รับการพัฒนาขึ้นที่มหาวิทยาลัย MIT ในปี 1977 โดยศาสตราจารย์ 3 คน ซึ่งประกอบด้วย Rivest, Shamir และ Adleman ชื่อของอัลกอริทึมได้รับการตั้งชื่อตามตัวอักษรตัวแรกของนามสกุลของศาสตราจารย์ทั้งสามคนอัลกอริทึมนี้ สามารถใช้ในการเข้ารหัสข้อมูลรวมทั้งการลงลายมือชื่ออิเล็กทรอนิกส์ด้วย

- อัลกอริทึม DSS

อัลกอริทึม DSS ย่อมาจาก Digital Signature Standard อัลกอริทึมนี้ได้รับการพัฒนาขึ้นมาโดย National Security Agency ในประเทศสหรัฐอเมริกาและได้รับการรับรองโดย NIST ให้เป็นมาตรฐานกลางสำหรับการลงลายมือชื่ออิเล็กทรอนิกส์ในประเทศสหรัฐอเมริกา

- อัลกอริทึมสำหรับสร้าง Message Digest

Message Digest หรือเรียกสั้นๆ ว่าไดเจสต์ แปลว่าข้อความสรุปจากเนื้อหาข้อความตั้งต้น โดยปกติข้อความสรุปจะมีความยาวน้อยกว่าความยาวของข้อความตั้งต้นมาก จุดประสงค์สำคัญของอัลกอริทึมนี้

คือ การสร้างข้อความสรุปที่สามารถใช้เป็นตัวแทนของข้อความตั้งต้นได้ โดยทั่วไปข้อความสรุปจะมีความยาวอยู่ระหว่าง 128 ถึง 256 บิต และจะไม่ขึ้นกับขนาดความยาวของข้อความตั้งต้น

คุณสมบัติที่สำคัญของอัลกอริทึมสำหรับสร้างไดเจสต์มีดังนี้

1. ทุกๆ บิตของไดเจสต์จะขึ้นอยู่กับทุกบิตของข้อความตั้งต้น

2. ถ้าบิตใดบิตหนึ่งของข้อความตั้งต้นเกิดการเปลี่ยนแปลง เช่น ถูกแก้ไข ทุกๆ บิตของไดเจสต์จะมีโอกาสร้อยละ 50 ที่จะแปรเปลี่ยนค่าไปด้วย ซึ่งหมายถึงว่า 0 เปลี่ยนค่าเป็น 1 และ 1 เปลี่ยนเป็น 0 คุณสมบัติข้อนี้สามารถอธิบายได้ว่าการเปลี่ยนแปลงแก้ไขข้อความตั้งต้นโดยผู้ไม่ประสงค์ดีแม้ว่าอาจแก้ไขเพียงเล็กน้อยก็ตาม เช่น เพียง 1 บิตเท่านั้น ก็จะส่งผลให้ผู้รับข้อความทราบว่าข้อความที่ตนได้รับไม่ใช่ข้อความตั้งต้น (โดยการนำข้อความที่ตนได้รับเข้าอัลกอริทึมเพื่อทำการคำนวณหาไดเจสต์ออกมา แล้วจึงเปรียบเทียบไดเจสต์ที่คำนวณได้กับไดเจสต์ที่ส่งมาให้ด้วย ถ้าต่างกัน แสดงว่าข้อความที่ได้รับนั้นถูกเปลี่ยนแปลงแก้ไข)

3. โอกาสที่ข้อความตั้งต้น 2 ข้อความใดๆ ที่มีความแตกต่างกัน จะสามารถคำนวณได้ค่าไดเจสต์เดียวกันมีโอกาสน้อยมากคุณสมบัติข้อนี้ทำให้แน่ใจได้ว่า เมื่อผู้ไม่ประสงค์ดีทำการแก้ไขข้อความตั้งต้น ผู้รับข้อความที่ถูกแก้ไขไปแล้วนั้นจะสามารถตรวจพบได้ถึงความผิดปกติที่เกิดขึ้นอย่างแน่นอน อย่างไรก็ตามในทางทฤษฎีแล้ว มีโอกาสที่ข้อความ 2 ข้อความที่แตกต่างกันจะสามารถคำนวณแล้วได้ค่าไดเจสต์เดียวกัน ปัญหานี้เรียกกันว่าการชนกันของไดเจสต์(Collision) อัลกอริทึมสำหรับสร้างไดเจสต์ที่ดีควรจะมีโอกาสน้อยมากๆ ที่จะก่อให้เกิดปัญหาการชนกันของไดเจสต์

อัลกอริทึมสำหรับสร้างไดเจสต์ยอดนิยมมีดังนี้

- อัลกอริทึม MD2 ผู้พัฒนาคือ Rivest อัลกอริทึมนี้เชื่อกันว่ามีความแข็งแกร่งที่สุดในบรรดาอัลกอริทึมต่างๆ ที่ Rivest พัฒนาขึ้นมา (ความแข็งแกร่งพิจารณาได้จากคุณสมบัติสามประการข้างต้น) ข้อเสียของอัลกอริทึมนี้คือใช้เวลามากในการคำนวณไดเจสต์หนึ่งๆ MD2 จึงไม่ค่อยได้มีการใช้งานกันมากนัก

  MD2 สร้างไดเจสต์ที่มีความยาว 128 บิต

- อัลกอริทึม MD4 ผู้พัฒนาคือ Rivest เช่นเดียวกับ MD2 อัลกอริทึมนี้พัฒนาขึ้นมาเพื่อแก้ปัญหาความล่าช้าในการคำนวณของ MD2 อย่างไรก็ตามในภายหลังได้พบว่าอัลกอริทึมมีข้อบกพร่องที่เกี่ยวข้องกับคุณสมบัติข้อที่สามโดยตรง กล่าวคือปัญหาการชนกันของไดเจสต์มีโอกาสเกิดขึ้นได้ไม่น้อยซึ่งผู้บุกรุกอาจใช้ประโยชน์จากจุดอ่อนนี้เพื่อทำการแก้ไขข้อความตั้งต้นที่ส่งมาให้ได้ MD4 ผลิตไดเจสต์ที่มีขนาด 128 บิต

- อัลกอริทึม MD5 Rivest เป็นผู้พัฒนาเช่นกันโดยพัฒนาต่อจาก MD4 เพื่อให้มีความปลอดภัยที่สูงขึ้น ถึงแม้จะเป็นที่นิยมใช้งานกันอย่างแพร่หลาย ทว่าในปี 1996 ก็มีผู้พบจุดบกพร่องของ MD5 เช่นเดียวกับ MD4 จึงทำให้ความนิยมเริ่มลดลงMD5 ผลิตไดเจสต์ที่มีขนาด 128 บิต

- อัลกอริทึม SHA ย่อจาก Secure Hash Algorithm อัลกอริทึม SHA ได้รับแนวคิดในการพัฒนามาจาก MD4 และได้รับการพัฒนาขึ้นมาเพื่อใช้งานร่วมกับอัลกอริทึม DSS (ซึ่งใช้ในการลงลายมือชื่ออิเล็กทรอนิกส์) หลังจากที่ได้มีการตีพิมพ์เผยแพร่อัลกอริทึมนี้ได้ไม่นาน NIST ก็ประกาศตามมาว่าอัลกอริทึมจำเป็นต้องได้รับการแก้ไขเพิ่มเติมเล็กน้อยเพื่อให้สามารถใช้งานได้อย่างเหมาะสม SHA สร้างไดเจสต์ที่มีขนาด 160 บิต
- อัลกอริทึม SHA-1 อัลกอริทึม SHA-1 เป็นอัลกอริทึมที่แก้ไขเพิ่มเติมเล็กน้อยจาก SHA การแก้ไขเพิ่มเติมนี้เป็นที่เชื่อกันว่าทำให้อัลกอริทึม SHA-1 มีความปลอดภัยที่สูงขึ้นSHA-1 สร้างไดเจสต์ที่มีขนาด 160 บิต
- อัลกอริทึม SHA-256, SHA-384 และ SHA-512 NIST เป็นผู้นำเสนออัลกอริทึมทั้งสามนี้ในปี 2001 เพื่อใช้งานร่วมกับอัลกอริทึม AES (ซึ่งเป็นอัลกอริทึมในการเข้ารหัสแบบสมมาตร) อัลกอริทึมเหล่านี้สร้างไดเจสต์ที่มีขนาด 256, 384 และ 512 บิต ตามลำดับนอกจากอัลกอริทึมสำหรับการสร้างไดเจสต์ที่กล่าวถึงไปแล้วนั้น อัลกอริทึมสำหรับการเข้ารหัสแบบสมมาตร เช่น DES สามารถใช้ในการสร้างไดเจสต์เช่นกัน วิธีการใช้งานอัลกอริทึมแบบสมมาตรเพื่อสร้างไดเจสต์คือ ให้เลือกกุญแจลับสำหรับการเข้ารหัสขึ้นมา 1 กุญแจโดยวิธีการเลือกแบบสุ่ม และต่อมาใช้กุญแจนี้เพื่อเข้ารหัสข้อความตั้งต้น แล้วใช้เฉพาะบล็อกสุดท้ายที่เข้ารหัสแล้วเพื่อเป็นไดเจสต์ของข้อความทั้งหมด ไม่รวมบล็อกอื่นๆ ที่เข้ารหัสแล้ว อัลกอริทึมแบบสมมาตรสามารถสร้างไดเจสต์ที่มีคุณภาพดี แต่ข้อเสียคือต้องใช้เวลาในการคำนวณไดเจสต์มากไดเจสต์เป็นเครื่องมือที่สำคัญที่สามารถใช้ในการตรวจสอบว่าไฟล์ในระบบที่ใช้งานมีการเปลี่ยนแปลงแก้ไขหรือไม่ ไม่ว่าจะโดยเจตนาหรือไม่ก็ตาม บางครั้งการเปลี่ยนแปลงแก้ไขอาจถูกกระทำโดยผู้ที่ไม่มีสิทธิ์ เช่น ผู้บุกรุก เป็นต้น วิธีการใช้ไดเจสต์เพื่อตรวจสอบไฟล์ในระบบคือให้เลือกใช้อัลกอริทึมหนึ่ง เช่น MD5 เพื่อสร้างไดเจสต์ของไฟล์ในระบบและเก็บไดเจสต์นั้นไว้อีกที่หนึ่งนอกระบบ ภายหลังจากระยะเวลาหนึ่งที่กำหนดไว้ เช่น 1 เดือน ก็มาคำนวณไดเจสต์ของไฟล์เดิมอีกครั้งหนึ่ง แล้วเปรียบเทียบไดเจสต์ใหม่นี้กับไดเจสต์ที่เก็บไว้นอกระบบว่าตรงกันหรือไม่ ถ้าตรงกัน ก็แสดงว่าไฟล์ในระบบยังเป็นปกติเช่นเดิมไดเจสต์ยังเป็นส่วนหนึ่งของการลงลายมือชื่ออิเล็กทรอนิกส์ กล่าวคือการลงลายมือชื่ออิเล็กทรอนิกส์ในปัจจุบันจะใช้การลงลายมือชื่อกับไดเจสต์ของข้อความตั้งต้นแทนการลงลายมือชื่อกับข้อความตั้งต้นทั้งข้อความ