การพัฒนาโปรแกรมเพื่อซ่อนข้อมูลสำหรับตรวจสอบเอกสารแฟกซ์

Application Development for Data Hiding in Fax Document

โดย

นางสาว วิภาศิริ สายวิรุณพร

รหัส 42067117

อาจารย์ที่ปรึกษา

ผศ.ดร. นพพร โชติกกำธร

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมเพื่อซ่อนข้อมูลสำหรับตรวจสอบเอกสารแฟกซ์
นักศึกษา	นางสาว วิภาศิริ สายวิรุณพร
อาจารย์ที่ปรึกษา	ผศ.ดร. นพพร โชติกกำธร
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ปัจจุบันเอกสารแฟกซ์ได้มีส่วนสำคัญกับระบบธุรกิจ ซึ่งวิธีการส่งเอกสารแฟกซ์ผ่านโมเด็มด้วยวิธีการธรรมดาจะสามารถทำการดัดแปลง แก้ไขข้อมูล ทำให้ข้อมูลเหล่านั้นไม่มีความน่าเชื่อถือ หรือทำความเสียหายให้กับธุรกิจได้ วิธีการหนึ่งที่จะสามารถป้องกันการกระทำเช่นนี้ได้ ก็คือวิธีการทำการซ่อนข้อมูลบางอย่างลงไปในไฟล์เอกสารแฟกซ์นี้ เพื่อเป็นประโยชน์ในการพิสูจน์หลักฐาน หรือเป็นหมายเหตุ หรือเพื่อใช้อ้างสิทธิ์ ในแบบที่ยากต่อการสังเกตพบได้ ความจำเป็นดังกล่าวข้างต้นจึงนำมาใช้เป็นแนวทางเพื่อทำการพัฒนาโปรแกรมเพื่อซ่อนข้อมูลสำหรับตรวจสอบไฟล์เอกสารแฟกซ์

โครงการนี้เป็นการศึกษาพัฒนาวิธีการซ่อนข้อมูลลงไปในไฟล์เอกสารภาพที่จะทำการส่งแฟกซ์ผ่านโมเด็มทั้งในแบบที่สามารถมองเห็นได้ด้วยตาเปล่า และไม่สามารถมองเห็นได้ด้วยตาเปล่า โดยวิธี Least Significant Bit และสามารถถอดรหัสข้อมูลที่ซ่อนไปกับไฟล์เอกสารได้

Title	Application Development for Data Hiding in Fax Document
Student	Miss. Wipasiri Saiwiroonporn
Advisor	Assist.Prof.Dr. Nopporn Chotikakamthorn
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

Abstract

At present, fax document is important in business. Fax sending via fax/modem in normal process can edits and transforms the information and also makes the information be unbelievable and destroy to that business. We can protect that problem by data hiding in fax document file. It's useful in proving the evidence or the reference. It's difficult to destroy and observe easily. This necessity can use in guideline for program development to data hiding in fax document.

This project developing the application for data hiding in fax document that sending documents via fax/modem. The method for develop application are use both visible watermark and invisible watermark.

กิตติกรรมประกาศ

โครงการพัฒนาระบบโปรแกรมเพื่อซ่อนข้อมูลลงไปในเอกสารแฟกซ์ นี้ สำเร็จลุล่วงด้วยดี เนื่องจากได้รับคำแนะนำจาก ดร. นพพร โชติกกำธร อาจารย์ที่ปรึกษา ซึ่งกรุณาให้ข้อคิดเห็นต่างๆ เพื่อเป็นแนวทางในการศึกษาและดำเนินการให้เป็นไปอย่างต่อเนื่อง เพื่อให้โครงการนี้สำเร็จตามวัตถุประสงค์ที่ได้ตั้งไว้

นอกจากนี้ต้องขอขอบคุณแรงบันดาลใจจากพ่อ และกำลังใจจากแม่ อีกทั้งเพื่อนๆที่คอยให้กำลังใจ และสุดท้ายนี้ขอขอบคุณเครื่องคอมพิวเตอร์ที่ถึงแม้จะมีแบดก็ยังทำงานต่อได้ ขอบคุณร้านขายซีดีในตะวันนาที่วางขายแผ่น โปรแกรมราคาถูกมาให้ได้ทำการศึกษาในราคาย่อมเยา ซึ่งมีประโยชน์ต่อการพัฒนาโครงการและทำให้โครงการนี้ได้รับความสำเร็จเป็นอย่างดี

วิภาศิริ สายวิรุณพร

สารบัญ

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

หน้า

ภาพที่

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความก้าวหน้าของสื่อดิจิตอลที่ใช้ในการสื่อสารข้อมูลเพื่อความสะดวกและรวดเร็ว เช่น การส่งเอกสารแฟกซ์ ซึ่งในบางครั้งอาจจะมีการดัดแปลงแก้ไขเอกสารที่จะทำการส่งแฟกซ์หรือส่งมาโดยผู้ที่ไม่ใช่เจ้าของเอกสาร โดยในปัจจุบันนี้มีเครื่องสแกนภาพ (Scanner) รวมทั้งเทคโนโลยีในการรู้จำตัวอักษร (Character recognition technology) ทำให้เอกสารเหล่านี้สามารถทำซ้ำและเผยแพร่ต่อออกไปได้โดยง่าย และง่ายต่อการดัดแปลงแก้ไขข้อมูลในเอกสารให้บิดเบือนไปจากความเป็นจริง ดังนั้นจึงเกิดความจำเป็นในการที่จะมีการทำการป้องกันการละเมิดลิขสิทธิ์ วิธีการหนึ่งที่สามารถทำได้ก็คือการทำการซ่อนข้อมูลลงไปในเอกสารที่จะใช้ในการส่งแฟกซ์ โดยข้อมูลที่ซ่อนลงไปนั้นสามารถที่จะชี้เฉพาะความเป็นเจ้าของเอกสารแฟกซ์นี้ได้ และข้อมูลที่ทำการซ่อนลงไปในเอกสารแฟกซ์จะคงทนอยู่ในไฟล์เอกสาร

โดยวิธีที่จะนำมาใช้ในการพัฒนาระบบโปรแกรมประยุกต์ใช้ในการซ่อนข้อมูลลงไปในเอกสารที่จะใช้ในการส่งแฟกซ์นั้น สามารถทำได้โดยการซ่อนข้อมูลลงไปในไฟล์เอกสารภาพ โดยใช้วิธีซ่อนข้อมูลลงไปในพิกเซลของไฟล์เอกสาร

1.2 วัตถุประสงค์ของการพัฒนาระบบโปรแกรมประยุกต์ในการซ่อนข้อมูลในเอกสารแฟกซ์

1. เพื่อพัฒนาระบบโปรแกรมประยุกต์ที่สามารถซ่อนข้อมูลลงไปในไฟล์ที่จะใช้ในการส่งข้อมูลแบบแฟกซ์
2. เพื่อศึกษาหาวิธีที่จะใช้ในการซ่อนข้อมูลลงไปในไฟล์ที่จะใช้ส่งข้อมูลแบบแฟกซ์ที่เหมาะสมและสามารถตรวจสอบความถูกต้องรวมทั้งตรวจสอบข้อมูลที่ซ่อนไปกับไฟล์ข้อมูลของเอกสารได้

1.3 แนวความคิดที่เกี่ยวข้องในการพัฒนาระบบโปรแกรมประยุกต์ในการซ่อนข้อมูลเอกสารแฟกซ์

1.1 เอกสารแฟกซ์

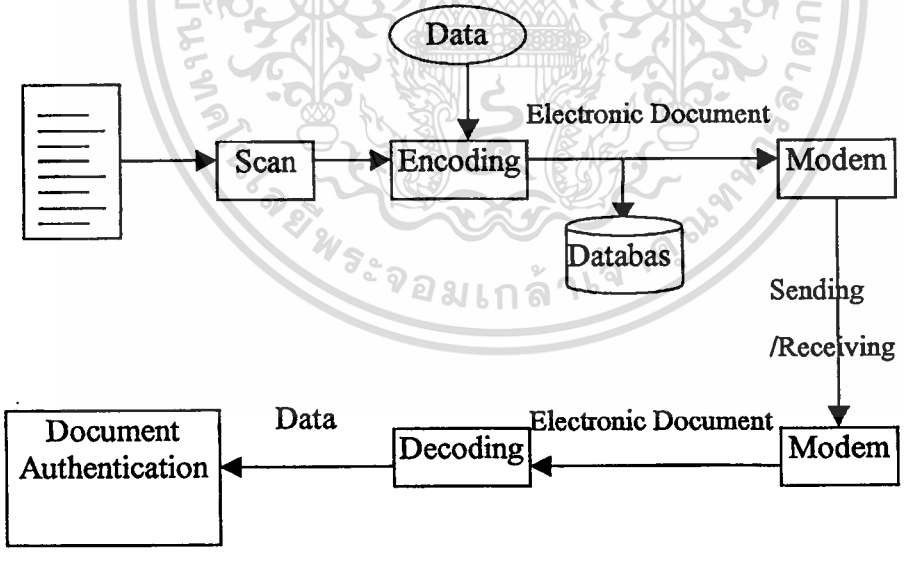
เครื่องส่งแฟกซ์ บางทีจะเรียกว่า "telecopying" เครื่องส่งเอกสารแฟกซ์ คือ อุปกรณ์ที่ใช้ในการส่งเอกสารข้อมูลผ่านโทรศัพท์ โดยเอกสารต้นฉบับจะถูกสแกนด้วยเครื่องแฟกซ์ โดยจัดการให้ข้อมูลเหล่านั้น (ภาพและตัวอักษร) อยู่ในรูปของไฟล์ภาพ แล้ว

แปลงข้อมูลที่ได้อยู่ในรูปของไฟล์ bitmap ในรูปแบบของดิจิตอล จากนั้นข้อมูลจะถูกส่งผ่านไปเป็นสัญญาณไฟฟ้าผ่านระบบโทรศัพท์ไปยังฝั่งผู้รับ เครื่องแฟกซ์ที่ฝั่งผู้รับจะทำการรับข้อมูลเป็นเอกสารภาพแล้วพิมพ์ออกมา

ปัจจุบันนี้บริษัทผู้ผลิตโมเด็มสามารถทำให้โมเด็มรับส่งข้อมูลแฟกซ์ได้แล้ว โดยมีโปรแกรมสำหรับแฟกซ์โมเด็มเพื่อสร้างสัญญาณแฟกซ์โดยตรงจากไฟล์ข้อมูลไปยังผู้รับได้โดยไฟล์ที่ส่งไปยังผู้รับจะอยู่ในรูปของไฟล์บิตแมพ ซึ่งวิธีการรับและส่งเอกสารแฟกซ์ด้วยโมเด็มนี้ปัจจุบันมีใช้กันอย่างแพร่หลายเพราะช่วยประหยัดกระดาษ เนื่องจากการส่งและรับจะอยู่ในลักษณะของไฟล์ข้อมูล

1.4 ขอบเขตของการศึกษา

เป็นการทำการศึกษาและพัฒนาระบบโปรแกรมประยุกต์เพื่อซ่อนข้อมูลลงไปในเอกสารภาพที่จะใช้ในการส่งแฟกซ์เพื่อแสดงความเป็นเจ้าของเอกสารนั้นๆ โดยมีขอบเขตของการศึกษาและพัฒนาดังต่อไปนี้



ภาพที่ 1.1 แสดงขอบเขตของการศึกษา

1. เอกสารที่จะทำการเข้ารหัสเพื่อใช้ในการส่งแฟกซ์นั้น สามารถได้มาจากการสแกน หรืออาจจะนำมาจากไฟล์ภาพเดิมที่มีอยู่แล้วในรูปแบบของไฟล์ภาพที่สามารถเป็นได้ทั้งภาพที่มีนามสกุลเป็น .jpg , .gif, .bmp และ .tiff

2. เอกสารที่จะนำมาถอดรหัสข้อมูลนั้นจะอยู่ในรูปของเอกสารอิเลคทรอนิคในรูปของไฟล์ภาพบิตแมพ
3. มีการจัดเก็บรายละเอียดการส่งเอกสารอิเลคทรอนิคส์ไว้ในฐานข้อมูล เพื่อสามารถตรวจสอบรายละเอียดการส่งเอกสารได้

1.5 แผนการดำเนินงาน

1 ศึกษาวิเคราะห์เปรียบเทียบวิธีการซ่อนข้อมูลในแต่ละประเภท เพื่อนำมาประยุกต์ใช้กับโปรแกรมประยุกต์ที่จะทำการพัฒนา
2. ออกแบบระบบการทำงานของโปรแกรมประยุกต์
3. ออกแบบโปรแกรมในส่วนติดต่อกับผู้ใช้งาน รวมทั้งรายงานการส่งเอกสารอิเลคทรอนิค
4. พัฒนาระบบและทดสอบการทำงานของโปรแกรมประยุกต์

บทที่ 2
ความรู้พื้นฐานเกี่ยวกับการซ่อนข้อมูล

2.1 การซ่อนข้อมูล

2.1.1 วัตถุประสงค์ของการทำการซ่อนข้อมูล

1. เพื่อนำข้อมูลที่ซ่อนไปพร้อมกับสื่อนั้นมาตรวจสอบความเป็นเจ้าของสื่อได้
2. เพื่อส่งข้อมูลที่เป็นความลับไปพร้อมกับสื่อ

2.1.2 คุณสมบัติของการซ่อนข้อมูล

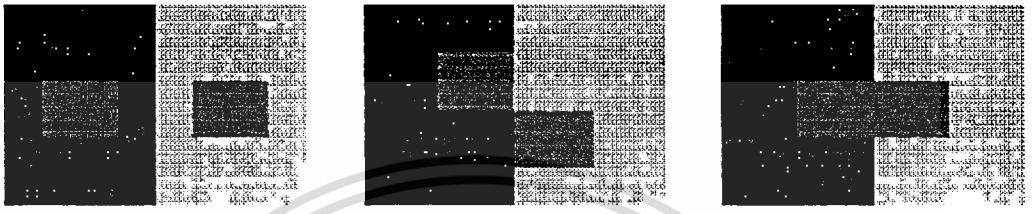
1. Discretion การซ่อนข้อมูลควรจะมองเห็นด้วยตาเปล่าไม่ได้
2. Cryptographic-security การซ่อนข้อมูลควรที่จะไม่สามารถลบทิ้งได้
3. Clipping-resistance การซ่อนข้อมูลจะต้องสามารถกู้ข้อมูลเก่ากลับมาได้
3. Transform-resistance การซ่อนข้อมูลควรที่จะคงอยู่ถึงแม้จะมีการเปลี่ยนแปลงของภาพไป เช่นการ rotation, scaling เป็นต้น

2.1.3 ลักษณะและเทคนิคของการซ่อนข้อมูล

1. สื่อที่ผสมกับข้อมูลที่ซ่อนนั้นไม่ควรลดคุณภาพของสื่อลง (nonobjectionally degraded) โดยข้อมูลที่ซ่อนอยู่นั้นไม่ควรที่จะสังเกตเห็นได้ง่าย
2. ข้อมูลที่ฝังไปนั้นควรที่จะเข้ารหัสไปโดยตรงกับข้อมูลภายในสื่อ (directly encoded into media) มากกว่าที่จะฝังไปกับส่วนหัวหรือที่อื่นของสื่อ เพื่อหลีกเลี่ยงการเปลี่ยนแปลงรูปแบบของแฟ้มข้อมูล
3. ข้อมูลที่ฝังไปนั้นควรที่จะมีความทนทานต่อการดัดแปลงแก้ไข (immune to modification)

เทคนิคการซ่อนข้อมูลได้นำมาพัฒนาประยุกต์เป็นโปรแกรมใช้งาน โดยมีประโยชน์มากมายในแง่ของการป้องกันการละเมิดลิขสิทธิ์ การควบคุมการทำสำเนา การเป็นหมายเหตุ และเพื่อการกำหนดสิทธิ์การใช้งาน โดยส่วนมากการซ่อนข้อมูลจะทำกับภาพที่เป็นสี หรือ ภาพ grayscale โดยที่พิกเซลจะมีค่าอยู่ในช่วงกว้าง ถ้าหากมีความเปลี่ยนแปลงเพียงเล็กน้อยนั้นโดยความสามารถในการมองเห็นด้วยตาของมนุษย์นั้นจะสังเกตเห็นได้ยากว่าภาพนั้นได้มีการเปลี่ยนแปลงเกิดขึ้น ดังตัวอย่างภาพที่ 2.1 แสดงถึงรูปสี่เหลี่ยมที่มีสีที่ต่างกัน โดยเปรียบเทียบมีสี่เหลี่ยมสีเทาเฉดขนาด

กลางเมื่อวางอยู่ในตำแหน่งที่ต่างกัน เมื่อมองด้วยสายตาแล้วจะเห็นได้ว่าสี่เหลี่ยมเล็กสีเทาจะมีความสว่างเพิ่มขึ้นเมื่ออยู่ในสี่เหลี่ยมสีดำ และจะมีความเข้มเพิ่มขึ้นเมื่ออยู่ในสี่เหลี่ยมสีเทาอ่อน ซึ่งความสามารถในการจำแนกความแตกต่างของสีด้วยสายตามนุษย์มีข้อจำกัดทำให้มีประโยชน์ในการทำการซ่อนข้อมูล ดังจะได้กล่าวต่อไป



ภาพที่ 2.1 แสดงความสามารถในการมองเห็นภาพด้วยตาของมนุษย์

2.2 Image Processing Algorithms

อัลกอริทึมที่ใช้ในกระบวนการสร้างภาพนี้ควรที่จะแบ่งออกตามกลุ่มโดยอาศัยพื้นฐานของแต่ละหลักการโดยใช้องค์ประกอบของภาพนั้น ๆ ได้แก่

1. Point Algorithms

การเปลี่ยนแปลงค่าของพิกเซลตามค่าเดิมหรือตำแหน่งเดิมบนภาพนั้นๆ แต่ละจุดในภาพประกอบไปด้วยพิกเซลเล็กๆมากมาย point algorithms จะคำนวณค่าบนพิกเซล โดยผลลัพธ์ที่ได้จะมีค่าอยู่ภายในค่าของภาพต้นฉบับและตำแหน่งของภาพนั้นในภาพต้นฉบับ ตัวอย่างของการทำกระบวนการนี้ได้แก่

- การเพิ่มความสว่างให้กับภาพ
- การเพิ่ม contrast ให้กับภาพ
- การทำ histogram equalisation
- การทำภาพ negative

2. Area Algorithms

เป็นอัลกอริทึมที่คำนวณค่าพิกเซลโดยใช้ค่าของกลุ่มของพิกเซลภายในภาพต้นฉบับ โดยกลุ่มเหล่านี้ควรที่จะกำหนดเป็นอาเรย์ 2 มิติ เช่น 3x3, 5x5 หรือ 7x7 เป็นต้น ตัวอย่างของการทำกระบวนการนี้ได้แก่

- การทำ Noise filtering

- การทำภาพให้มีความสมบูรณ์ขึ้น อันเนื่องมาจากกระบวนการส่งภาพ หรือการจัดเก็บข้อมูลภาพ เป็นต้น

3. Frame Algorithms

ค่าของพิกเซลในภาพขึ้นอยู่กับค่าของพิกเซลจากตำแหน่งทั่วไปของภาพ โดย frame algorithms เป็นอัลกอริทึมซึ่งทำงานร่วมกันกับภาพมากกว่าหนึ่งภาพในเวลาเดียวกัน ตัวอย่างของอัลกอริทึมที่นำมาใช้กับกระบวนการนี้ได้แก่

- การเพิ่มภาพ
- การลบภาพ
- การซ้อนภาพ

4. Geometrical Algorithms

อัลกอริทึมนี้โดยทั่วไปจะเป็นการเปลี่ยนรูปทรงของภาพ โดยการเปลี่ยนแปลงนี้ทำโดยการเปลี่ยนจำนวนของพิกเซล ตำแหน่งของพิกเซล โดยใช้ในกระบวนการ

- การหมุนภาพ (Image rotation)
- การทำภาพสะท้อน (Image mirroring)

5. อัลกอริทึมอื่น ๆซึ่งไม่ได้อยู่ในกลุ่มที่กล่าวไปข้างต้น

2.3 วิธีการซ่อนข้อมูลลงไปในเอกสารภาพที่จะส่งแฟกซ์

การซ่อนข้อมูลลงไปในเอกสารภาพสามารถทำได้หลายวิธีด้วยกัน เช่น วิธี Block-based pixel-domain ทำได้โดยการเปลี่ยนค่าของพิกเซลไป และวิธีการซ่อนข้อมูลโดยการเปลี่ยนแปลงลักษณะของตัวอักษรภายในเอกสารภาพ

2.3.1 การซ่อนข้อมูลโดยใช้วิธี Block-based pixel-domain

วิธีการทั่วไปในการซ่อนข้อมูลในภาพขาวดำนั้นทำโดยการเปลี่ยนค่าของพิกเซล และโดยการเปลี่ยนแปลงกลุ่มของพิกเซล โดยในหลักการแรกจะทำการซ่อนข้อมูลที่เป็นพิกเซลสีดำลงไปในพิกเซลสีขาว หลักการต่อมาจะเป็นวิธีการเปลี่ยนแปลงบางคุณลักษณะของภาพไปเช่น ความหนา, ความโค้ง, ตำแหน่ง ฯลฯ

โดยภาพจะถูกแบ่งออกเป็นบล็อก และมีการกำหนดจำนวนบิตที่จะทำการซ่อนข้อมูลลงไป แน่นอนในแต่ละบล็อกโดยการเปลี่ยนแปลงบางพิกเซลในแต่ละบล็อก พิจารณาวิธีการซ่อนข้อมูลดังต่อไปนี้

- จะมีวิธีการเลือกพิกเซลอย่างไรเพื่อนำมาทำการเปลี่ยนแปลงโดยที่ให้สังเกตด้วยตาได้ยากหรือไม่เห็นความเปลี่ยนแปลง
- กำหนดค่าที่จะทำการซ่อนข้อมูลลงไปในแต่ละบล็อกโดยใช้พิกเซลที่มีสีต่างกันกับของเดิม
- ทำอย่างไรจึงจะทำการซ่อนข้อมูลโดยที่มีจำนวนบิตที่ซ่อนนั้นเท่ากันในทุกบล็อก

โดยกระบวนการซ่อนข้อมูลและถอดรหัสข้อมูลมีหลักการที่เกี่ยวข้องดังต่อไปนี้

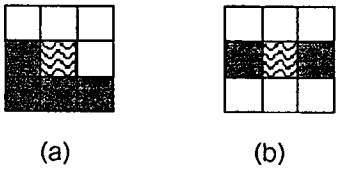
1. Flippable Pixels

เริ่มจากพิจารณาว่าพิกเซลไหนสามารถที่จะทำการสลับสีได้โดยที่สังเกตเห็นความเปลี่ยนแปลงไปได้ยากและมีความกลมกลืนกับภาพ ในตัวอย่างจากภาพที่ 2 ได้ทำการแบ่งบล็อกออกเป็นขนาด 3x3 พิกเซล จะเห็นได้ว่าถ้าหากมีการเปลี่ยนสีของพิกเซลลงในตำแหน่งเดียวกันนั้นจะเห็นความแตกต่างกันคือ ภาพ a จะสังเกตได้ยากเพราะมีความกลมกลืนไปกับภาพ ส่วนภาพ b ถ้าหากเปลี่ยนสีที่ตำแหน่งเดียวกันจะสังเกตเห็นได้ง่าย ดังนั้นการทำการเปลี่ยนสีก็ต้องพิจารณาพิกเซลที่อยู่รอบ ๆ ด้วย

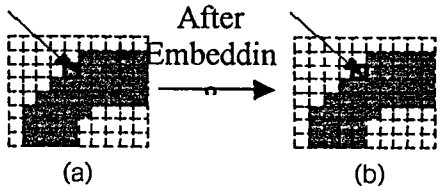
2.ขั้นตอนการซ่อนข้อมูล

ในการที่จะซ่อนข้อมูลจำนวน 1 บิตในทุกบล็อกนั้น พิกเซลที่จะถูกเปลี่ยนสีไปนั้นจะต้องทำให้สังเกตเห็นได้ยาก โดยขั้นตอนซ่อนข้อมูลจะมีส่วนในการพิจารณาการถอดรหัสข้อมูลโดยไม่จำเป็นต้องใช้ภาพต้นฉบับมาทำการเปรียบเทียบด้วย โดยการเข้ารหัสเพื่อทำการซ่อนข้อมูลในพิก

เซลที่สามารถเปลี่ยนสีได้อาจจะใช้ไม่ได้ผลเนื่องมาจากกระบวนการซ่อนข้อมูลอาจจะไปทำให้เกิดการเปลี่ยนแปลงพิกเซลไปในภาพต้นฉบับได้ ซึ่งจะทำให้มองเป็นพิกเซลที่ไม่ได้มีการเปลี่ยนสีหรือเข้ารหัสไว้ ดังตัวอย่าง กำหนดให้มีพิกเซลสีดำและสีขาว โดยพิกเซลสีดำกำหนดให้สามารถเปลี่ยนสีได้ ดังแสดงในภาพที่ 3a พิกเซลที่ถูกเปลี่ยนเป็นสีขาวถูกกำหนดให้เป็น "1" ภาพที่ 2.3b



ภาพที่ 2.2 แสดงบล็อกขนาด 3*3 ที่มีการเปลี่ยนสีที่ตำแหน่งกลางบล็อก



ภาพที่ 2.3 ส่วนที่ลูกศรชี้เป็นขอบของพิกเซลซึ่งจะกลายเป็นส่วนที่ไม่สามารถเปลี่ยนสีได้หลังจากทำการซ่อนข้อมูล

แสดงให้เห็นว่าพิกเซลนี้ไม่สามารถกำหนดให้เป็นพิกเซลที่สามารถเข้ารหัสได้ถ้าหากใช้วิธีการนี้จะตรวจสอบว่าภาพมีการเข้ารหัสไว้ไม่ได้ถ้าหากไม่ได้ใช้ภาพต้นฉบับมาทำการเปรียบเทียบซึ่งวิธีการนี้เป็นการซ่อนข้อมูลโดยการดูจากพิกเซลที่สามารถเปลี่ยนสีได้ ดังนั้นแน่นอนว่าความสัมพันธ์ของกลุ่มของพิกเซลจะเปลี่ยนแปลงไป ในการซ่อนข้อมูล "0" ลงไปในบล็อก โดยอาจจะมีการเปลี่ยนบางพิกเซลไปดังนั้นจำนวนของพิกเซลสีดำทั้งหมดในบล็อกนั้นจะเป็นเลขคู่ ในทางเดียวกันในการซ่อนข้อมูล "1" ดังนั้นจำนวนพิกเซลสีดำทั้งหมดในบล็อกจะเป็นเลขคี่ วิธีการอื่น ๆก็คือการเลือก "quantization" ขนาด Q และกำหนดจำนวนของพิกเซลสีดำในบล็อกให้เป็น 2KQ ในกรณีที่มีการซ่อนข้อมูล "0" และเป็น (2K+1)Q เมื่อมีการซ่อนข้อมูล "1" ค่า Q ยิ่งมากจะให้ความคงทนไม่มีสิ่งรบกวน แต่ภาพที่ได้จะมีคุณภาพลดลงไป

2.3.2 การซ่อนข้อมูลโดยวิธีการเปลี่ยนแปลงลักษณะของตัวอักษรภายในเอกสารภาพ

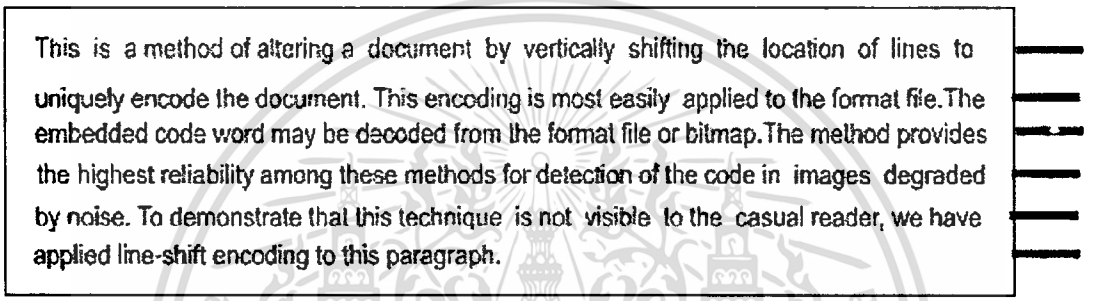
1. วิธี Line Shift Coding

วิธีการนี้เป็นการเปลี่ยนแปลงเอกสารภาพในแนวตั้ง โดยการเลื่อนตำแหน่งบรรทัดของเอกสาร ซึ่งวิธีการนี้เมื่อมีการถอดรหัสของข้อมูลที่ซ่อนไว้นั้น ไม่จำเป็นต้องใช้เอกสารต้นฉบับ โดยในการทำการซ่อนข้อมูลด้วยวิธีการเลื่อนตำแหน่งของบรรทัดขึ้นหรือลงไปเพียงเล็กน้อยดังแสดงในภาพที่ 2.4 โดย 1 บิตจะใช้แทนตำแหน่งบรรทัดที่เลื่อนตำแหน่งไป โดยถ้ามีการเลื่อนตำแหน่งของบรรทัดขึ้นจะแทนด้วย 1 และถ้ามีการเลื่อนตำแหน่งของบรรทัดลงจะแทนด้วย 0 เพื่อจะทำให้การเข้ารหัสทำได้โดยง่ายจะไม่ใช้บรรทัดแรกและบรรทัดสุดท้ายในการเข้ารหัส ส่วนวิธีการถอดรหัสข้อมูลจากเอกสารที่ได้ทำการสำเนานั้น มีขั้นตอนการทำงานดังนี้

- เอกสารถูกสแกนเข้าเครื่องคอมพิวเตอร์ และมีโปรแกรมที่ใช้ในการปรับเอกสารให้ตรง ไม่เอียง
- หาตำแหน่งจุดกึ่งกลางของเอกสาร และกึ่งกลางของแต่ละบรรทัดในเอกสารที่จะทำการถอดรหัส และนำตำแหน่งที่ได้เทียบกับตำแหน่งของเอกสารต้นฉบับ
- กำหนดให้ $\Delta R,+$ เป็นระยะห่างระหว่างจุดกึ่งกลางของบรรทัดที่มีการเลื่อนทำแหน่งขึ้นไป กำหนดให้ $\Delta R,-$ เป็นระยะห่างระหว่างจุดกึ่งกลางของบรรทัดที่มีการเลื่อนตำแหน่งลงไปจากเดิม และกำหนดให้ $\Delta X,+$ และ $\Delta X,-$ เป็นระยะห่างระหว่างจุดกึ่งกลางของสองบรรทัดในเอกสารต้นฉบับ

ถ้า $\frac{\Delta_{R,+} - \Delta_{R,-}}{\Delta_{R,+} - \Delta_{R,-}} > \frac{\Delta_{X,+} - \Delta_{X,-}}{\Delta_{X,+} - \Delta_{X,-}}$ แสดงว่าระยะห่างระหว่างบรรทัดเพิ่มขึ้น

ถ้า $\frac{\Delta_{R,+} - \Delta_{R,-}}{\Delta_{R,+} - \Delta_{R,-}} < \frac{\Delta_{X,+} - \Delta_{X,-}}{\Delta_{X,+} - \Delta_{X,-}}$ แสดงว่าระยะห่างระหว่างบรรทัดลดลง



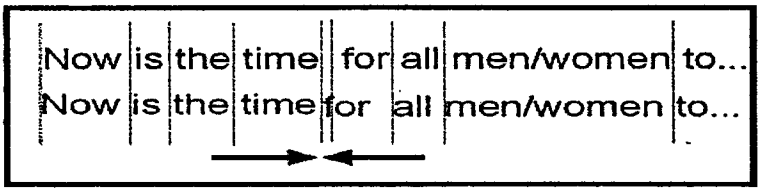
ภาพที่ 2.4 แสดงการเลื่อนตำแหน่งบรรทัดขึ้นไปจากเดิม สังเกตว่าบรรทัดที่ 3 ได้ถูกลื่อนตำแหน่งขึ้นไป

ข้อจำกัดของวิธีการนี้ทางฝั่งผู้รับสามารถที่จะทำการลบข้อมูลที่ถูกซ่อนมาได้ โดยทำการเลื่อนตำแหน่งของบรรทัดให้กลับมาเท่ากันทุกบรรทัดได้ และถ้าหากเอกสารเหล่านั้นประกอบไปด้วย ตัวอักษรประกอบภาพ, ตาราง ฯลฯ จะทำให้การเข้ารหัสและถอดรหัสด้วยวิธีการนี้ทำได้ยากขึ้น

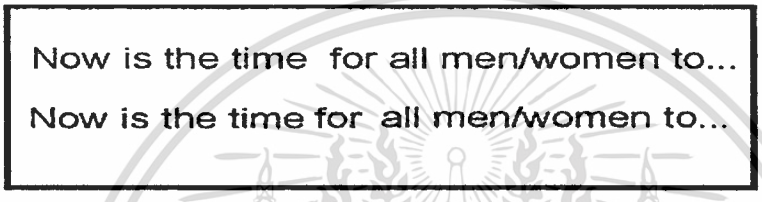
ข้อดีของวิธีการนี้ก็คือ สามารถทนต่อสิ่งรบกวนที่เกิดจากการพิมพ์ หรือการทำสำเนาเอกสารได้ดีกว่าอีกสองวิธีการที่จะกล่าวถึงต่อไป

2. วิธี Word Shift Coding

เป็นวิธีการเปลี่ยนแปลงเอกสารในแนวนอน โดยการเลื่อนตำแหน่งของคำภายในบรรทัดเพื่อเป็นการเข้ารหัสเอกสาร ดังแสดงในภาพที่ 2.5 ซึ่งภายในบรรทัดจะมีช่องว่างระหว่างคำที่ไม่เท่ากัน ซึ่งเป็นสาเหตุให้จำเป็นต้องใช้เอกสารต้นฉบับในการถอดรหัสข้อมูล

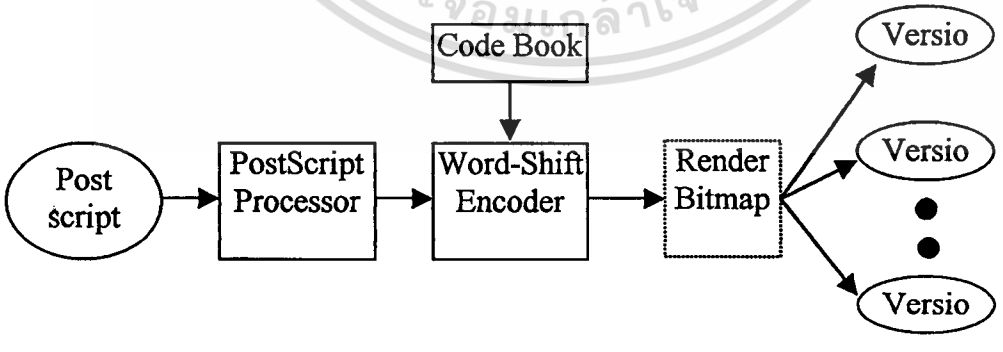


ภาพที่ 2.5 แสดงตัวอย่างของวิธีการเข้ารหัสแบบ word-shift โดยมีการเลื่อนตำแหน่งของคำว่า for ไปทางด้านซ้าย



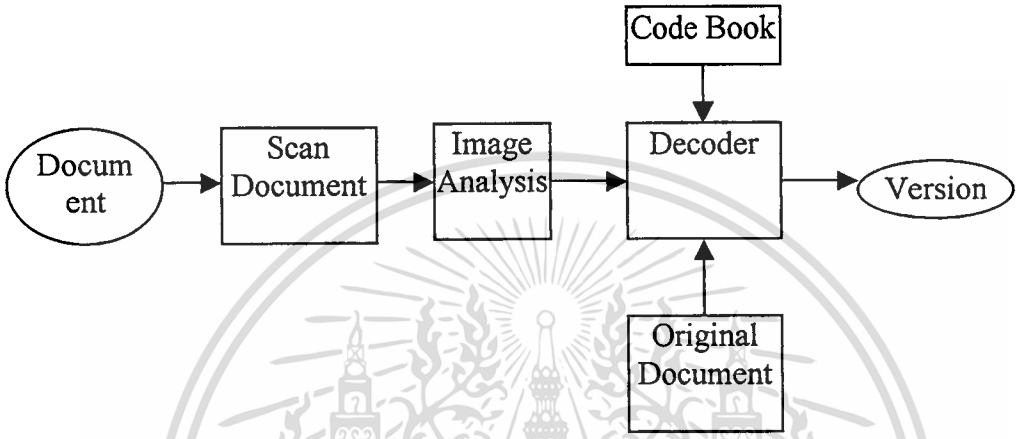
ภาพที่ 2.6 แสดงภาพของเอกสารที่ได้ทำการเลื่อนตำแหน่งของคำไป

ขั้นตอนการเข้ารหัส ตัวอย่างที่ได้ทำการศึกษานี้ใช้เอกสารที่เป็น post script โดย Encoder ประกอบด้วย 2 ส่วนคือ Preprocessor และ word shifter โดย preprocessor จะเป็นการหาตำแหน่งของแต่ละคำที่แน่นอนไว้ ต่อมาก็เป็นหน้าที่ของ word shifter ที่จะทำการเปลี่ยนแปลงตำแหน่งของตัวอักษร โดยการเลื่อนตำแหน่งจากการเพิ่มค่าคงที่ให้กับแต่ละตำแหน่ง การส่งเอกสารให้กับผู้รับแต่ละคนจะกำหนดรหัสคำให้ซึ่งไม่ซ้ำกัน เก็บไว้ใน codebook ดังแสดงในภาพที่ 2.7



ภาพที่ 2.7 แสดงขั้นตอนการเข้ารหัสด้วยวิธี word-shift

ขั้นตอนการถอดรหัส เริ่มจากนำเอกสารที่มีการเข้ารหัสมาทำการวิเคราะห์ โดยการถอดรหัสต้องอาศัย รหัสจาก codebook และเอกสารต้นฉบับมาทำการเปรียบเทียบกัน จึงจะทราบได้ว่าเอกสารที่ได้รับมามีรหัสถูกต้องหรือไม่ ดังแสดงในภาพที่ 2.8



ภาพที่ 2.8 แสดงขั้นตอนการถอดรหัสเอกสารที่ถูกเข้ารหัสไว้ด้วยวิธี word-shift

โดยการถอดรหัสนั้นมีวิธีการทำดังนี้

- หาตำแหน่ง Centroid เพื่อใช้ในการคำนวณสำหรับแต่ละคำทั้งในเอกสารต้นฉบับ และเอกสารที่จะทำการตรวจสอบ
- กำหนดให้ คำที่ I เป็นคำที่ถูกเลื่อนตำแหน่งไปในบรรทัดที่มีการเข้ารหัส
- โดยในเอกสารภาพต้นฉบับกำหนดให้ :

  x_{i-1}, x_i, x_{i+1} เป็นตำแหน่งจุด Centroid ของคำที่ i-1,i, i+1
- ในเอกสารภาพที่จะทำการตรวจสอบ กำหนดให้ :
- x_{i-1}, x_i, x_{i+1} เป็นตำแหน่งจุด Centroid ของคำที่ i-1,i, i+1
- ดังนั้น ค่าที่แตกต่างกันระหว่างคำจากจุดกึ่งกลางบนเอกสารภาพเดียวกันจะเป็นดังนี้

$$d_l \triangleq X_i - X_{i-} \quad , \quad d \triangleq X_{i+1} - X_i$$

$$d' \triangleq X_i' - X_{i-}' \quad , \quad d' \triangleq X_{i+1}' - X_i'$$

โดย dl คือระยะห่างของคำ เมื่อเทียบกับจุดกึ่งกลางคำ ไปทางซ้าย ของเอกสารภาพต้นฉบับ

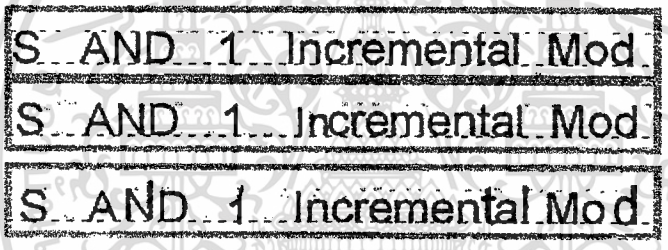
โดย dl คือระยะห่างของคำ เมื่อเทียบกับจุดกึ่งกลางคำ ไปทางซ้าย ของเอกสารภาพที่จะทำการตรวจสอบ

ถ้าหากว่าค่า $d_l - d_l > d_r - d_r$ แสดงว่าคำเลื่อนไปทางขวา

ถ้าหากว่าค่า $d_l - d_l < d_r - d_r$ แสดงว่าคำเลื่อนไปทางซ้าย

3. วิธี Feature Coding

เป็นวิธีการที่อาศัยคุณลักษณะของตัวอักษรเพื่อทำการเข้ารหัสโดยทำการเปลี่ยนแปลงลักษณะของตัวอักษรไป โดยทำการเปลี่ยนแปลงความยาวของพิกเซลของตัวอักษรที่มีหาง โดยอาจจะเพิ่มหรือลดพิกเซล ดังแสดงในภาพที่ 2.9 ซึ่งวิธีการเข้ารหัสโดยอาศัยคุณลักษณะของตัวอักษรนั้นเมื่อทำการถอดรหัสข้อมูลจำเป็นต้องอาศัยเอกสารต้นฉบับมาเปรียบเทียบ



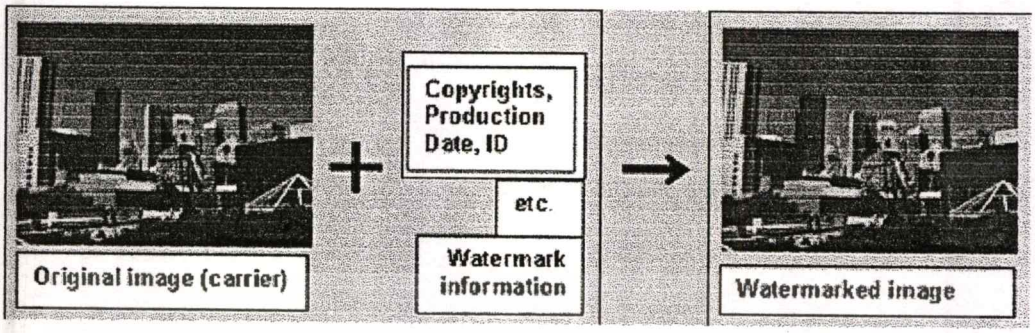
ภาพที่ 2.9 แสดงการเข้ารหัสโดยวิธี Feature Coding คือการอาศัยคุณลักษณะของตัวอักษรเช่นหางของตัวอักษร b,d,h เป็นต้น สามารถเปลี่ยนได้โดยทำให้ความยาวของหางของตัวอักษรนั้นยาวขึ้นหรือว่าสั้นลงไป 1 หรือ หลายพิกเซล ดังภาพ a ยังไม่มีการเข้ารหัส ภาพที่ b มีการเข้ารหัสที่คำว่า In โดยลดความยาว ของตัว I ลงไปเล็กน้อย ส่วนภาพ C เป็นการแสดง การเข้ารหัสที่เกินความจริง

4. ทำลายน้ำดิจิตอล

1. ความหมายของลายน้ำดิจิตอล

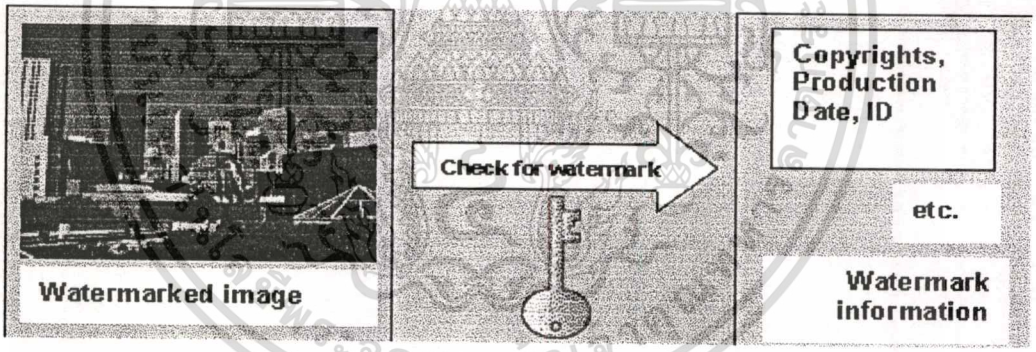
การทำลายน้ำดิจิจอลเป็นเทคนิคสำหรับการฝังข้อมูลหลายรูปแบบลงไปในสื่อดิจิตอล โดยทั่วไปข้อมูลที่ต้องการฝังลงไปเพื่อใช้เป็นลิขสิทธิ์นั้นจะกระทำในลักษณะของลายน้ำ

2. โครงสร้างของลายน้ำดิจิตอล ดังแสดงในภาพที่ 2.10



ภาพที่ **2.10** แสดงโครงสร้างของลายน้ำดิจิตอล

สื่อซึ่งเก็บลายน้ำดิจิตอลไว้จะเรียกว่าพาหะ (carrier) โดยลายน้ำดิจิตอลนี้จะไม่ใช่เป็นการเชื่อมต่อไฟล์เข้าไปด้วยกัน แต่จะเป็นการฝังข้อมูลลงไปในไฟล์พาหะ ซึ่งจำเป็นต้องใช้ซอฟท์แวร์ในการฝังข้อมูลและชี้เฉพาะหาข้อมูลที่ฝังไว้ได้



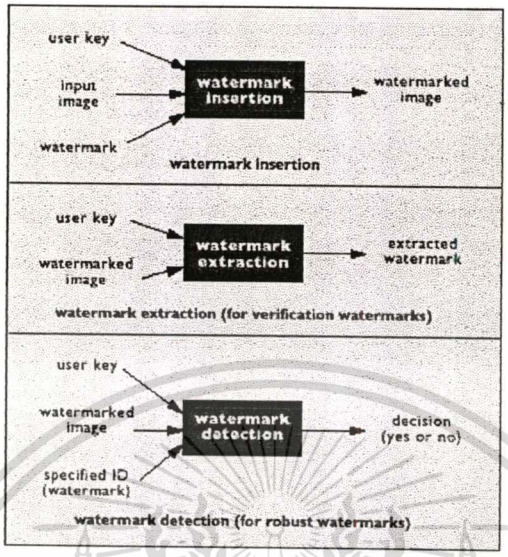
ภาพที่ **2.11** แสดงการถอดรหัสข้อมูลที่ฝังไว้จากภาพที่ได้มีการทำลายน้ำดิจิตอล

เทคนิคในการทำลายน้ำดิจิตอล

3.1 ลายน้ำดิจิตอลที่สามารถมองเห็นได้ (Visible Watermark) โดยทั่วไปอาจจะใช้เป็นสัญลักษณ์ของบริษัทเพื่อแสดงความเป็นเจ้าของเอกสารภาพ

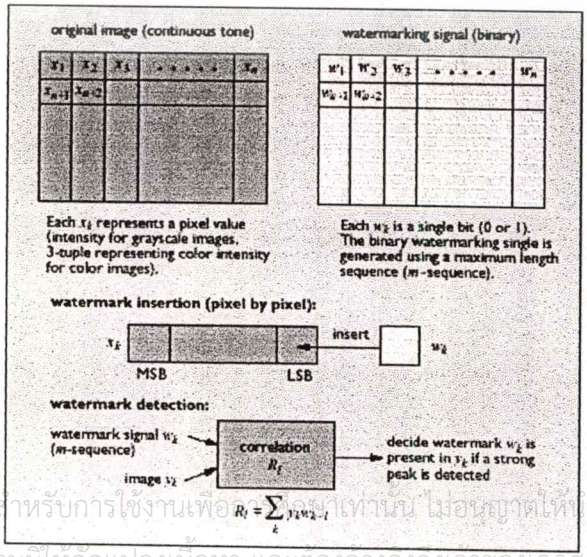
3.2 ลายน้ำดิจิตอลที่ไม่สามารถมองเห็นได้ (Invisible Watermark) ก็คล้ายกับลายน้ำดิจิตอลที่สามารถมองเห็นได้ แต่ไม่มีความจำเป็นในการชี้เฉพาะให้เห็น

หลักการทำงานของลายน้ำดิจิตอลแสดงดังภาพที่ 2.12 แสดงให้เห็นหลักการทำงานโดยทั่วไปสำหรับการใส่ลายน้ำลงไปในภาพ โดยทั่วไปลายน้ำจะประกอบไปด้วยข้อมูลต่าง ๆดังนี้คือ รหัสผู้ใช้, ภาพต้นฉบับ และค่าพารามิเตอร์ของภาพ (ตัวอย่างเช่น ขนาดของภาพ)



ภาพที่ 2.12 แสดงหลักการทำงานโดยทั่วไปของการทำลายน้ำดิจิตอล

ต่อไปเป็นวิธีการใส่ลายน้ำดิจิตอลลงไปในเอกสารภาพว่าสามารถทำได้อย่างไรในที่นี้จะแสดงถึงวิธีการใส่ลายน้ำดิจิตอล ซึ่งมีประโยชน์สำหรับการเก็บรายละเอียดข้อมูลส่วนตัว โดยทำการฝังข้อมูลลงไปในบิตที่มีนัยสำคัญน้อย (least significant bit) ของเอกสารภาพ ดังแสดงในภาพที่ 1 กำหนดให้ $X_1, X_2, \ldots$ คือ พิกเซลที่อยู่ในเอกสารภาพ ในการที่จะใส่ลายน้ำดิจิตอลลงไปในภาพมีขั้นตอนในการสร้างดังนี้ เริ่มจากสร้าง watermark signal โดยใช้คีย์เพื่อสร้างค่าของ m-sequence โดยมีค่าเป็นไบนารี จากนั้นจะจัดเรียงค่าอยู่ในรูปของตารางสองมิติ ดังแสดงในภาพที่ 1 จากนั้นสัญญาณที่ได้จะถูกใส่เข้าไปพิกเซลต่อพิกเซลในตำแหน่งของภาพที่มีนัยสำคัญน้อย (LSB) ของภาพต้นฉบับ X_k



ภาพที่ 2.13 แสดงเทคนิคการทำลายน้ำดิจิตอล

โดยการทำ Least Significant bit insertion นั้นมีหลักการทำงานที่ง่ายสำหรับการซ่อนข้อมูลลงไปในภาพ โดยเทคนิคการทำ LSB ในแต่ละไบท์ของภาพ 24 บิต สามารถใช้ 3 บิตซ่อนลงไปในแต่ละพิกเซล (โดยแต่ละพิกเซลประกอบไปด้วย 3 ไบท์) การเปลี่ยนแปลงในแต่ละพิกเซลนั้นจะไม่สามารถสังเกตเห็นได้ด้วยตามนุษย์ ตัวอย่างเช่น ตัวอักษร A สามารถซ่อนลงไปได้ 3 พิกเซลดังแสดงได้จาก 24 บิต 3 ชุดดังนี้

(00100111 11101001 11001000) (00100111 11001000 11101001)

(11001000 00100111 11101001)

โดยค่าไบนารีของตัวอักษร A เป็นดังนี้คือ (10000011) ทำการใส่ค่าไบนารีของตัวอักษร A ลงไปใน 3 พิกเซล โดยเริ่มจากไบท์สุดท้ายซ้ายมือ ผลที่ได้จะเป็นดังนี้

(00100111 11101000 11001000) (00100110 11001000 11101000)

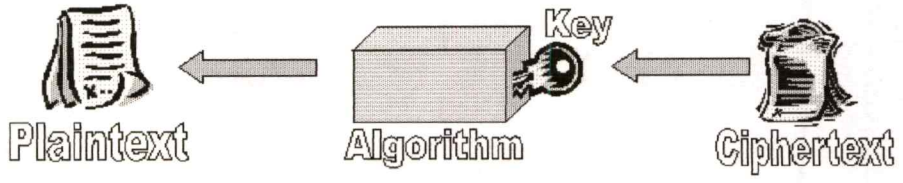
(11001000 00100111 11101001)

จากตัวอย่างข้างต้นตัวอักษรบิตที่เอียงไปเป็นบิตที่ได้มีการเปลี่ยนค่าไป

รูปแบบการทำลายน้ำดิจิตอลซึ่งต้องการรหัสของผู้ใช้สามารถแบ่งได้เป็น 2 ประเภทคือ

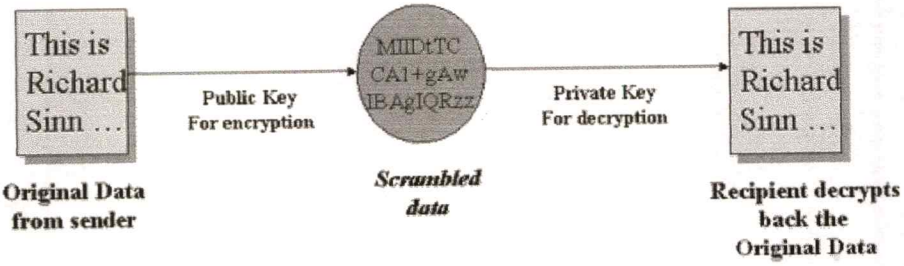
1. **Secret-key** จะใช้รหัสเดียวกันทั้งตอนเข้ารหัสและถอดรหัส ดังนั้นวิธีการเข้ารหัสแบบนี้จำเป็นต้องมีความปลอดภัยในการสื่อสารข้อมูลระหว่างผู้ส่งและผู้รับ ดังแสดงในภาพที่ 2.14

ภาพที่ 2.14 แสดงวิธีการเข้ารหัสแบบ Secret Key



ภาพที่ 2.15 แสดงวิธีการถอดหัสแบบ Secret Key

2. **Public-key** จะมีรหัสของการเข้ารหัสและถอดรหัสคนละตัวกัน โดย private-key จะรู้เฉพาะผู้ที่เป็นเจ้าของเท่านั้นใช้ในการเข้ารหัส ส่วนการถอดรหัสจะมี public-key ซึ่งผู้รับจะรู้รหัสนี้ ดังนั้นผู้ที่สามารถจะทำการเข้ารหัสได้นั้นมีเพียงแต่เจ้าของสื่อเท่านั้น



ภาพที่ 2.16 แสดงวิธีการเข้ารหัสแบบ Public Key

2.4. Faxmodem Overview

มาตรฐานใหม่ของแฟกซ์โมเด็มมีชื่อว่า Class 1 faxmodem และเป็นส่วนขยายมาจากมาตรฐานชุดคำสั่งโมเด็ม AT ซึ่งพัฒนามาจาก ITU แต่ว่า Class 1 faxmodem มีปัญหากับเครื่องคอมพิวเตอร์ที่มีความเร็วซีพียูต่ำ วิธีการแก้ปัญหานี้ก็คือการลดการประมวลผลโดยยกการประมวลผลเหล่านี้ไปไว้ที่ตัวแฟกซ์โมเด็มแทน

มาตรฐานใหม่ต่อมาก็คือ Class 2 faxmodem ซึ่งบริษัทผู้ผลิตแฟกซ์โมเด็มส่วนใหญ่ก็ทำตามมาตรฐานใหม่นี้ ซึ่งในความเป็นจริงแล้ว แฟกซ์โมเด็มที่มีขายกันอยู่ในปัจจุบันนี้สนับสนุนทั้งมาตรฐาน Class 1 และ Class 2

ขั้นตอนของการรับส่งแฟกซ์

มาตรฐานของการรับส่งแฟกซ์ ได้ถูกทำเป็นเอกสารจาก ITU (International telecommunication Union) ซึ่งสามารถแบ่งการทำงานออกเป็นช่วง ๆ ได้ดังต่อไปนี้

Phase A – ในช่วงนี้ เป็นช่วงเริ่มต้นกระบวนการรับส่งแฟกซ์ phase A จะสิ้นสุดเมื่อทั้งสองฝั่งติดต่อกันสื่อสารถึงกันได้

Phase B – ในช่วงนี้เป็นช่วงที่อุปกรณ์แฟกซ์โมเด็มของทั้งสองฝั่งยอมรับในกระบวนการและค่าที่ใช้ในการรับส่งแฟกซ์กัน (ความเร็วในการรับส่ง, ความละเอียดของภาพ เป็นต้น) ในช่วงนี้อุปกรณ์แฟกซ์โมเด็มทั้งสองฝั่งส่งผ่านข้อมูลให้กัน เช่น localID หรือ RemoteID ซึ่งตัวอักษรที่ชี้เฉพาะเจาะจงนี้สามารถมีความยาวได้ถึง 20 ตัวอักษร

Phase C – การส่งผ่านข้อมูลจริงเกิดขึ้นในช่วงนี้

Phase D – ในช่วงนี้อุปกรณ์แฟกซ์โมเด็มที่ทำการรับข้อมูลยืนยันว่าได้รับข้อมูลครบถ้วนแล้ว และทางอุปกรณ์แฟกซ์โมเด็มฝั่งผู้ส่งจะบอกไปยังอุปกรณ์ฝั่งผู้รับว่ามีเอกสารหน้าอื่นที่จะส่งต่อไปอีกหรือไม่ ถ้าหากว่ามีเอกสารหน้าต่อไปที่จะส่งอีกก็จะกลับขึ้นไปยัง Phase B อีก

Phase E – ในช่วงนี้เป็นช่วงยกเลิกการติดต่อ อุปกรณ์แฟกซ์โมเด็มจะหยุดการติดต่อสื่อสารถึงกัน

ขั้นตอนการรับส่งแฟกซ์ สามารถเสนอเป็นตารางการทำงานได้ดังนี้

ตารางที่ 2.1 แสดงขั้นตอนการรับส่งแฟกซ์ผ่านเครื่องแฟกซ์โมเด็ม

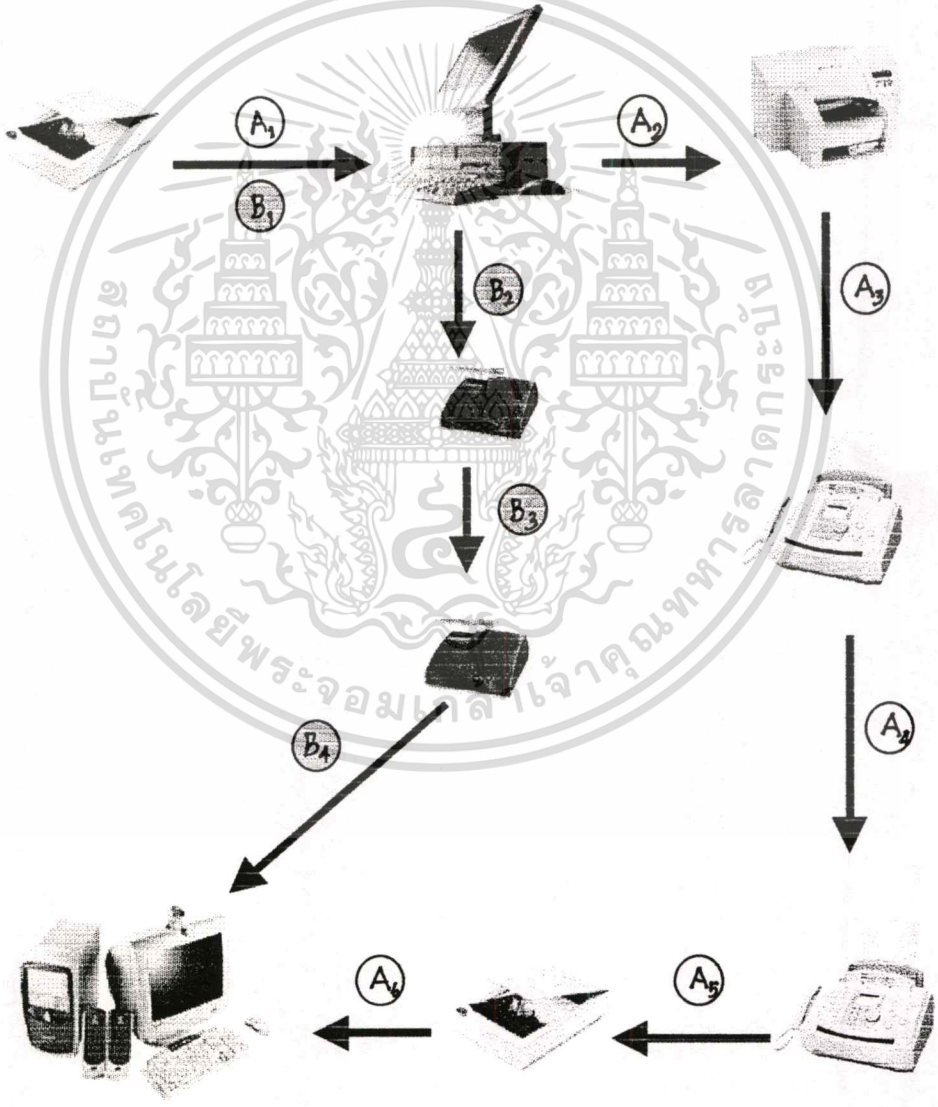
Receiving Events	Sending Events
Initializing Modem	Initializing Modem
Initializing Modem for Receive	Initializing Modem for sending
Waiting for Ring.....	Dialing .....
Answering	Waiting to Connect
Negotiating.....	Connect to remote fax machine.....
Receiving Fax Page Data	Negotiating.....
End of Page	Sending Fax Page.....
Send Complete	Sending Page Data .....
Comm Port Closed	Sending Page, 10% Complete
	Sending Page, 20% Complete
	Sending Page, 30% Complete
	Sending Page, 40% Complete
	Sending Page, 50% Complete
	Sending Page, 60% Complete
	Sending Page, 70% Complete
	Sending Page, 80% Complete
	Sending Page, 90% Complete
	Sending Page, 100% Complete
	Send Complete ....
	Comm Port Closed

บทที่ 3

การพัฒนาโปรแกรมประยุกต์ซ่อนข้อมูลในเอกสารแฟกซ์

ปัจจุบันวิธีการส่งแฟกซ์ที่สามารถกระทำได้อีกวิธีการหนึ่งนั้นก็คือ วิธีการส่งเอกสารแฟกซ์ผ่านเครื่องแฟกซ์โมเด็ม ซึ่งมีรายละเอียดและขั้นตอนการทำงานดังต่อไปนี้

3.1 ขั้นตอนในการส่งเอกสารแฟกซ์

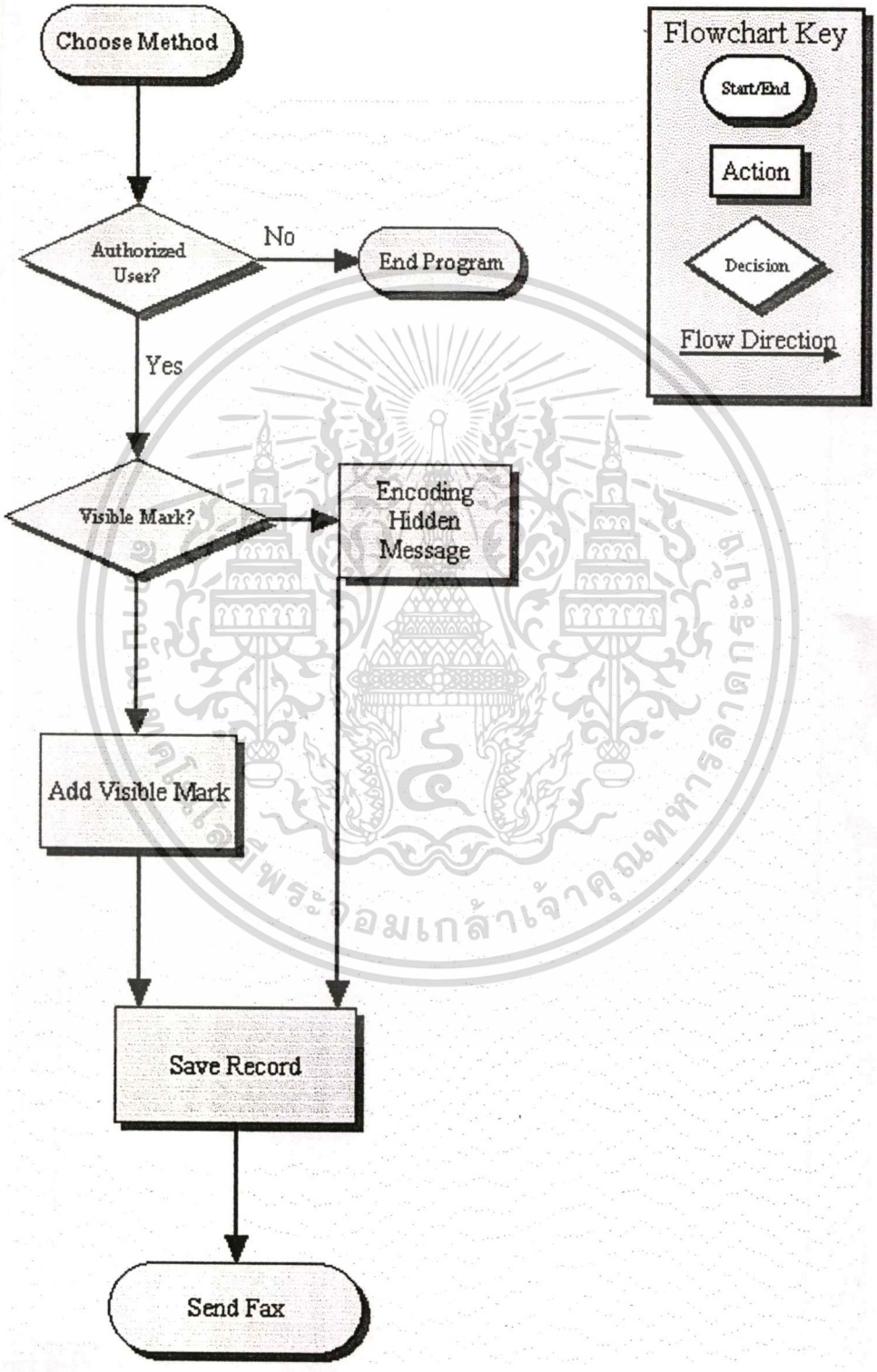


ขั้นตอนในการทำงานของการพัฒนาระบบโปรแกรมประยุกต์ในการซ่อนข้อมูลลงไปในเอกสารแฟกซ์นั้นสามารถทำได้ 2 ขั้นตอนด้วยกัน

1. ขั้นตอนตามตัวอักษร A นั้นเริ่มจาก
   - A1 สแกนเอกสารที่ต้องการซ่อนข้อมูลและส่งแฟกซ์เก็บเป็นไฟล์นามสกุล .bmp ไว้ในเครื่องซึ่งมีโปรแกรมที่พัฒนาขึ้นในการซ่อนข้อมูล (Encode)
   - A2 พิมพ์เอกสารที่ได้มีการซ่อนข้อมูลลงไปแล้วออกทางเครื่องพิมพ์
   - A3 นำเอกสารที่มีการซ่อนข้อมูลแล้วส่งไปทางเครื่องแฟกซ์
   - A4 ทางฝั่งผู้รับรับเอกสารแฟกซ์ด้วยเครื่องแฟกซ์
   - A5 โนทางฝั่งผู้รับถ้าหากต้องการตรวจสอบว่าเอกสารนี้ถูกส่งมาจากผู้ส่งจริงหรือไม่สามารถทำได้โดยสแกนเอกสารที่ได้รับมาโดยเครื่องสแกนเนอร์
   - A6 เซฟไฟล์ที่ได้จากการสแกนเก็บไว้ในไฟล์นามสกุล .bmp จากนั้นนำไฟล์ที่ได้มาตรวจสอบโดยใช้โปรแกรมที่ได้พัฒนาขึ้นเพื่อใช้ตรวจสอบเอกสาร (Decode)
2. ขั้นตอนตามตัวอักษร B นั้นเริ่มจาก
   - B1 สแกนเอกสารที่ต้องการซ่อนข้อมูลและส่งแฟกซ์เก็บเป็นไฟล์นามสกุล .bmp ไว้ในเครื่องซึ่งมีโปรแกรมที่พัฒนาขึ้นในการซ่อนข้อมูล (Encode)
   - B2 นำไฟล์ที่ได้ทำการซ่อนข้อมูลไว้แล้วส่งผ่านเครื่องแฟกซ์โมเด็มไปยังฝั่งผู้รับ
   - B3 ทางฝั่งผู้รับรับไฟล์ที่ส่งมาด้วยเครื่องแฟกซ์โมเด็ม
   - B4 ถ้าหากทางฝั่งผู้รับต้องการตรวจสอบไฟล์เอกสารที่ได้สามารถตรวจสอบโดยใช้โปรแกรมที่พัฒนาขึ้นเพื่อใช้ในการตรวจสอบเอกสาร

จากภาพจะเห็นได้ว่าในการส่งเอกสารแฟกซ์ด้วยการใช้เครื่องแฟกซ์นั้นจะมีขั้นตอนการทำงานที่มากกว่าการใช้เครื่องแฟกซ์โมเด็มในการส่งเอกสาร ในที่นี้จึงจะนำเสนอวิธีการซ่อนข้อมูลลงในเอกสารแฟกซ์โดยใช้วิธีการส่งไฟล์เอกสารด้วยเครื่องแฟกซ์โมเด็มโดยใช้วิธีการทำลายน้ำดิจิตอลในการซ่อนข้อมูลลงไปในเอกสารภาพ โดยสามารถทำได้ทั้งในรูปแบบของ Visible Watermark และ Invisible Watermark

3.2 การออกแบบการทำงานของโปรแกรม



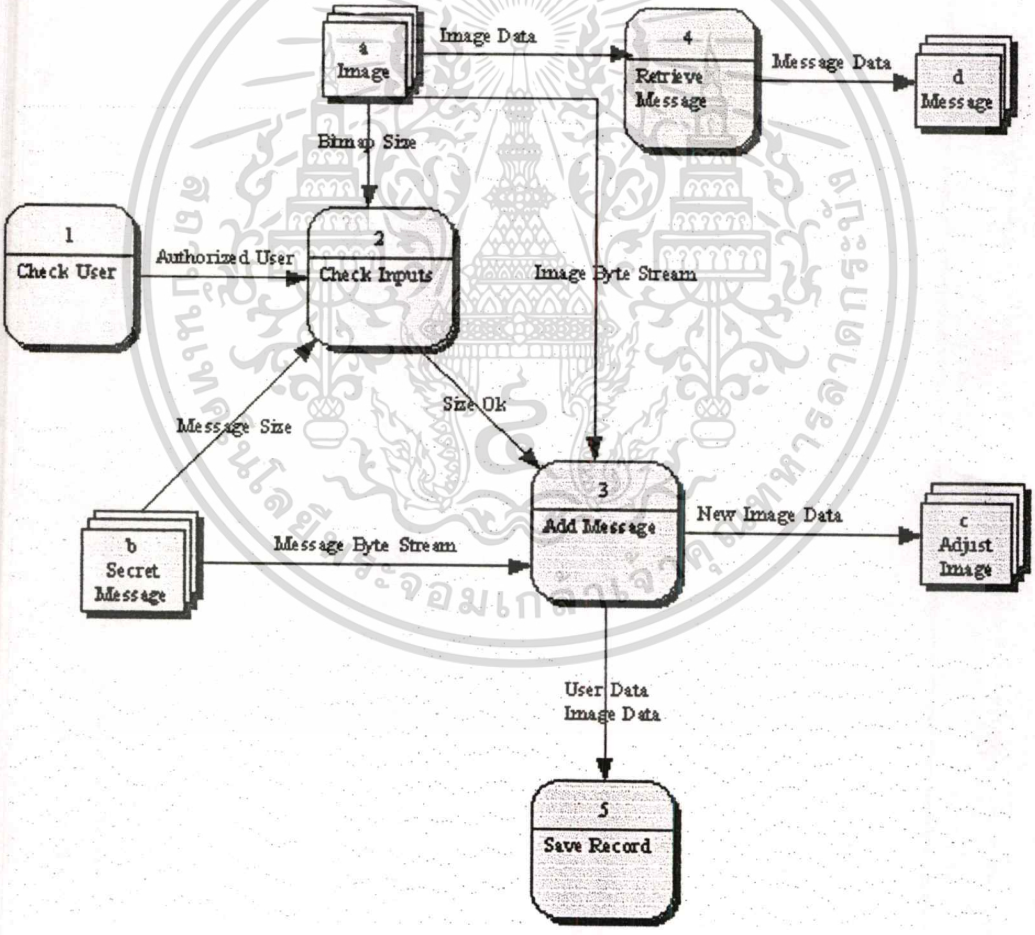
ภาพที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรม

3.2.1 Invisible Watermark

- **Level 1 Data Flow Diagram**

อินพุทของโปรแกรมต้องใช้ข้อมูล 3 อินพุทด้วยกันคือ เอกสารรูปภาพ, รหัสลับที่จะใช้ซ่อนข้อมูล และชื่อของเอกสารภาพ โดยมีขั้นตอนการทำงานดังแสดงในภาพที่ 3.2 ด้วยกัน 5 ขั้นตอนคือ

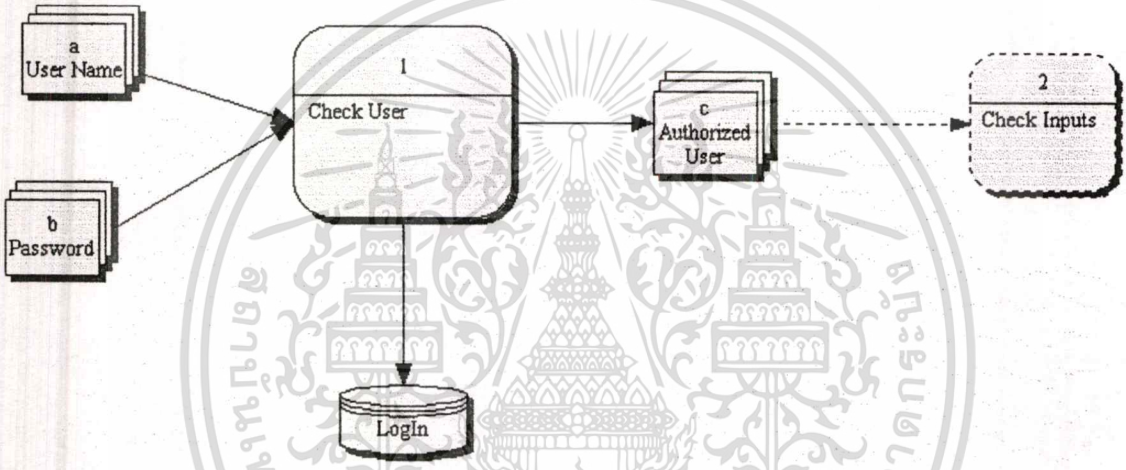
- ตรวจสอบผู้ใช้
- ตรวจสอบอินพุท (Check Input) ว่าเป็นไฟล์ภาพหรือไม่
- ซ่อนรหัสข้อมูล (Add Message)
- การนำเอกสารมาถอดรหัส (Extract Message)



ภาพที่ 3.2 แสดง Data Flow Diagram Level 1 ของการทำ Invisible Watermark

1. ขั้นตอนการตรวจสอบผู้ใช้ (Check User)

ขั้นตอนการตรวจสอบผู้ใช้นี้ จะเป็นการติดต่อกับฐานข้อมูลผู้ใช้ ซึ่งผู้ที่เป็นเจ้าของสามารถให้สิทธิ์การเข้าใช้งานให้กับผู้ใช้ได้ผ่านโปรแกรมประยุกต์ที่จัดทำขึ้น โดยผู้ใช้จะต้องทำการ Log in เพื่อเข้าสู่ระบบก่อน โดยระบบจะทำการตรวจสอบรหัสผู้ใช้และรหัสผ่านที่เก็บไว้ในฐานข้อมูล ถ้าหากรหัสผู้ใช้และรหัสผ่านถูกต้องจึงจะสามารถเข้าใช้โปรแกรมประยุกต์ที่ใช้ในการซ่อนข้อมูลเพื่อส่งเอกสารแฟกซ์นี้ได้ และถ้าหากทำการ Log in เข้าระบบไม่ผ่านเป็นจำนวน 3 ครั้งระบบจะต้องทำการเข้าโปรแกรมใหม่ โดยขั้นตอนการทำงานสามารถแสดงได้ดังภาพที่ 3.3



ภาพที่ 3.3 แสดงขั้นตอนการตรวจสอบผู้ใช้งาน

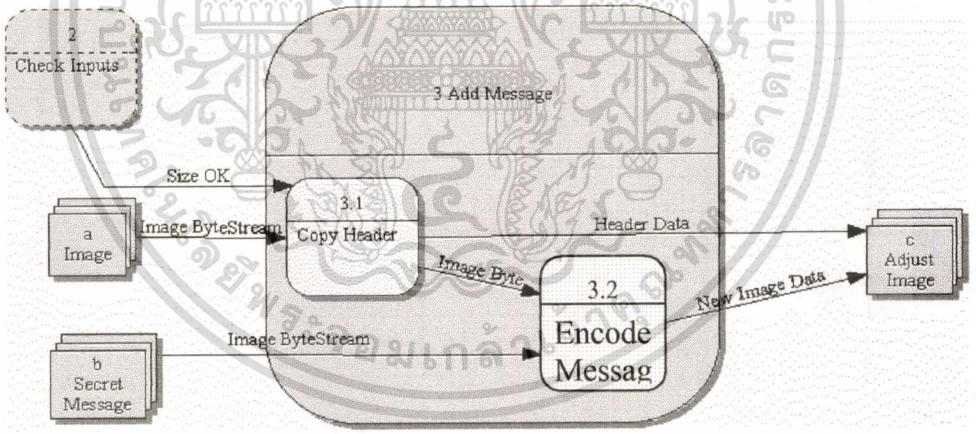
2. ขั้นตอนการ Add Message

ขั้นตอนการเข้ารหัสข้อมูลนี้ ส่วนของ header ของข้อมูลจะต้องไม่ถูกเปลี่ยนแปลงไป ดังนั้นควรที่จะมีการทำสำเนาข้อมูลโดยตรงไปยัง output file การทำการซ่อนข้อมูลลงไปจำเป็นต้องมีการถอดรหัสข้อมูล ดังนั้นจึงมีความสำคัญในการที่จะแบ่งจำนวนไบท์ของข้อมูลเอกสารภาพจากภาพต้นฉบับ ขั้นตอนการเข้ารหัสมีดังต่อไปนี้

- เซ็ตค่ารหัสให้อยู่ในรูปของค่าออฟเซ็ท
- เข้ารหัสความยาวของข้อความที่จะทำการซ่อนลงไปในเอกสารภาพ
- ทำการเข้ารหัสข้อความที่ต้องการซ่อนลงไปในเอกสารภาพ
- แสดงภาพที่ได้ทำการเข้ารหัสเรียบร้อยแล้ว

การเข้ารหัสทำได้โดยการ

ในการเข้ารหัสในโปรแกรมประยุกต์นี้เป็นการใช้วิธีการทำการซ่อนข้อมูลลงไปในภาพ โดยใช้วิธี Least Significant Bit โดยสุ่มหาตำแหน่งที่จะทำการเข้ารหัสโดยใช้ความกว้างและความยาวของภาพเป็นตัวบอกขอบเขตพื้นที่ที่จะสามารถทำการเข้ารหัสได้ โดยเข้ารหัสที่ตำแหน่งใดนั้นจะกำหนดเป็นแถวและคอลัมน์ซึ่งมีการตรวจสอบด้วยว่าตำแหน่งที่เข้ารหัสนั้นจะไม่ซ้ำกัน และจะทำการเปลี่ยนสีที่พิกเซลใดโดยค่าที่จะทำการซ่อนข้อมูลนั้นจะมีค่า 0 หรือ 1 ทำการเข้ารหัสจนครบตามจำนวนความยาวของคำที่จะทำการซ่อนรหัส ตัวอย่างเช่น ในที่นี้กำหนดว่าเลือกพิกเซลสีเขียวขึ้นมา ดังนั้นคอมโพเน้นท์สี แดง,เขียว,น้ำเงิน แสดงในรูปของไบนารีได้เป็น 0000,1111,0000 ถ้าหากต้องการซ่อนข้อมูลลงไปในพิกเซลสีแดง ค่าของพิกเซลใหม่จะเป็นดังนี้คือ 0001,1111,0000 โดยการเปลี่ยนแปลงค่าเพียงเล็กน้อยนี้จะไม่สามารถสังเกตเห็นได้ด้วยตาเปล่า ในโปรแกรมนี้ได้ใช้รหัสผ่าน (Password) เพื่อเป็นตัวกำหนดค่าตัวเลขสุ่ม (Random Number) ซึ่งจะมีประโยชน์ในขั้นตอนของการถอดรหัสโดยรหัสผ่านนี้จะเป็นตัวกำหนดค่าชุดของตัวเลขสุ่มชุดเดียวกันกับในขั้นตอนของการเข้ารหัส โดยการทำงานในส่วนของการเข้ารหัสแสดงได้ในภาพที่ 3.4



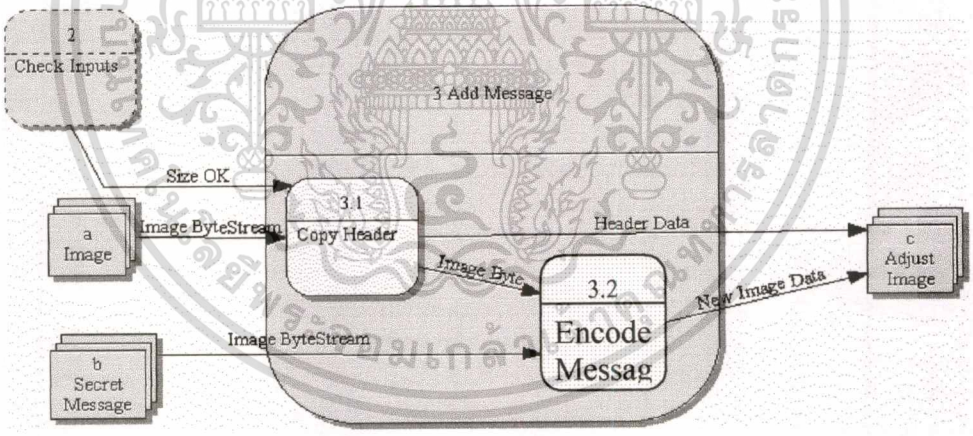
ภาพที่ 3.4 แสดงขั้นตอนการ Add Message ลงไปในไฟล์เอกสารภาพ

โดยขั้นตอนการทำงานสามารถเขียนได้ดังต่อไปนี้

```
For i = 1 to message length
        value = convert text of i message to ascii
        for i = 1 to 8
                byte mask = 1
                pick random position in image and random pixel for encode
```

การเข้ารหัสทำได้โดยการ

ในการเข้ารหัสในโปรแกรมประยุกต์นี้เป็นการใช้วิธีการทำการซ่อนข้อมูลลงไปในภาพ โดยใช้วิธี Least Significant Bit โดยสุ่มหาตำแหน่งที่จะทำการเข้ารหัสโดยใช้ความกว้างและความยาวของภาพเป็นตัวบอกขอบเขตพื้นที่ที่จะสามารถทำการเข้ารหัสได้ โดยเข้ารหัสที่ตำแหน่งใดนั้นจะกำหนดเป็นแถวและคอลัมน์ซึ่งมีการตรวจสอบด้วยว่าตำแหน่งที่เข้ารหัสนั้นจะไม่ซ้ำกัน และจะทำการเปลี่ยนสีที่พิกเซลใดโดยค่าที่จะทำการซ่อนข้อมูลนั้นจะมีค่า 0 หรือ 1 ทำการเข้ารหัสจนครบตามจำนวนความยาวของคำที่จะทำการซ่อนรหัส ตัวอย่างเช่น ในที่นี้กำหนดว่าเลือกพิกเซลสีเขียวขึ้นมา ดังนั้นคอมโพเน้นท์สี แดง,เขียว,น้ำเงิน แสดงในรูปของไบนารีได้เป็น 0000,1111,0000 ถ้าหากต้องการซ่อนข้อมูลลงไปในพิกเซลสีแดง ค่าของพิกเซลใหม่จะเป็นดังนี้คือ 0001,1111,0000 โดยการเปลี่ยนแปลงค่าเพียงเล็กน้อยนี้จะไม่สามารถสังเกตเห็นได้ด้วยตาเปล่า ในโปรแกรมนี้ได้ใช้รหัสผ่าน (Password) เพื่อเป็นตัวกำหนดค่าตัวเลขสุ่ม (Random Number) ซึ่งจะมีประโยชน์ในขั้นตอนของการถอดรหัสโดยรหัสผ่านนี้จะเป็นตัวกำหนดค่าชุดของตัวเลขสุ่มชุดเดียวกันกับในขั้นตอนของการเข้ารหัส โดยการทำงานในส่วนของการเข้ารหัสแสดงได้ในภาพที่ 3.4



ภาพที่ 3.4 แสดงขั้นตอนการ Add Message ลงไปในไฟล์เอกสารภาพ

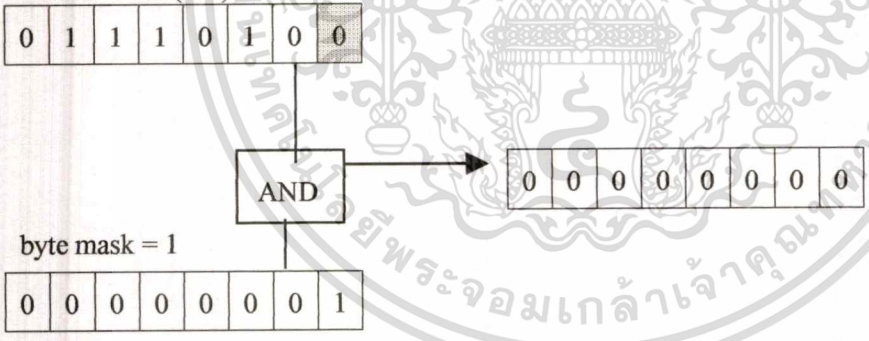
โดยขั้นตอนการทำงานสามารถเขียนได้ดังต่อไปนี้

```
For k = 1 to message length
        value = convert text of i message to ascii
        for i = 1 to 8
                byte mask = 1
                pick random position in image and random pixel for encode
```

```
get the pixel's color component
if value and byte mask then
        color mask = 1
else
        color mask = 0
end if
 update the pixel's color in image
set the pixel's color in image
byte mask = byte mask *2
    next i
next k
```

ตัวอย่างเช่น ต้องการซ่อนค่าตัวอักษร t ลงไปในภาพ เมื่อแปลงค่าตัวอักษร t แล้วจะได้ค่าไบนารี ดังนี้ขั้นตอนการนำข้อมูลตัวอักษร t ซ่อนลงไปในภาพ ในที่นี้จะแสดงขั้นตอนการซ่อนข้อมูลบิต 0 ตัวแรกของตัวอักษร t ลงไปในคอมโพเน้นท์สีที่ได้ทำการสุ่มเลือกมา

Value = asc("t")

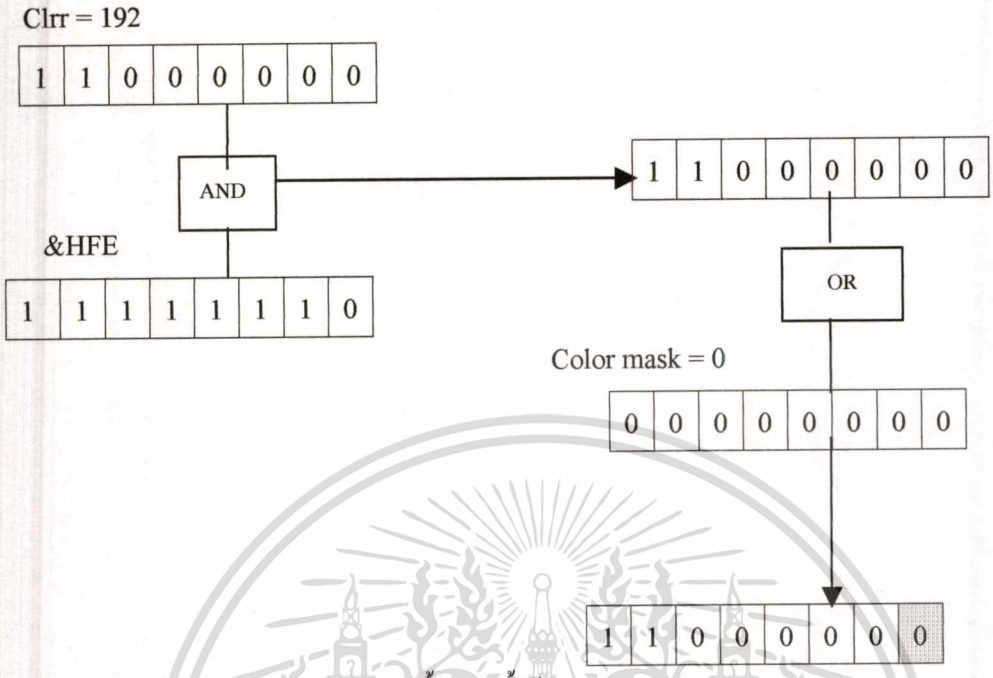


If value and byte mask then

Color mask = 0

0	0	0	0	0	0	0	0

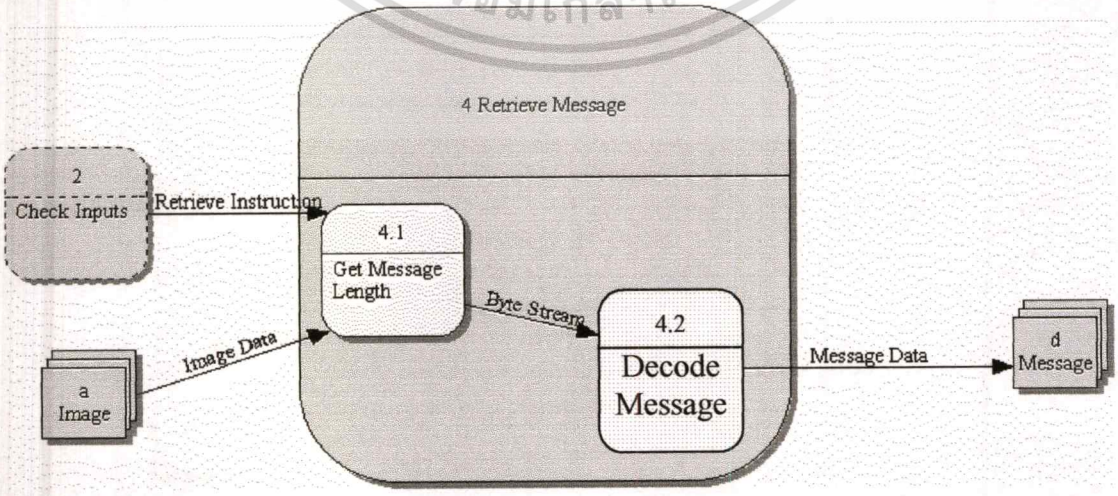
นำบิตแรกมาทำการซ่อนลงไปในพิกเซลที่ได้สุ่มเลือกมาในที่นี้กำหนดให้ทำการซ่อนข้อมูลลงไปในคอมโพเน้นท์สีแดงสามารถแสดงวิธีการซ่อนข้อมูลได้ดังต่อไปนี้ clrr = (clrr And &HFE) Or color_mask กำหนดให้ค่า clrr(ค่าของคอมโพเน้นท์ที่ได้ทำการสุ่มเลือกมา) = 192



(โดยสามารถดู source code ในขั้นตอนนี้ได้ในภาคผนวก)

3. ขั้นตอนการถอดรหัสข้อมูล

การถอดรหัสจะถอดรหัสจากข้อมูลที่ได้ทำการซ่อนไว้ภายในเอกสารภาพ โดยขั้นตอนการถอดรหัสข้อมูลจะรับข้อมูลที่ผู้ใช้ป้อนเป็นรหัสผ่านเข้ามาเพื่อทำการตรวจสอบว่าตรงกับตำแหน่งของข้อมูลที่ได้มีการเข้ารหัสไว้หรือไม่ ถ้าตรงกันจะทำการถอดรหัสข้อมูลข้อความที่ได้มีการซ่อนมากับเอกสารภาพและแสดงข้อความนั้นออกมาได้ โดยขั้นตอนการถอดรหัสข้อมูลภายในเอกสารภาพแสดงได้ใน data flow ภาพที่ 3.5



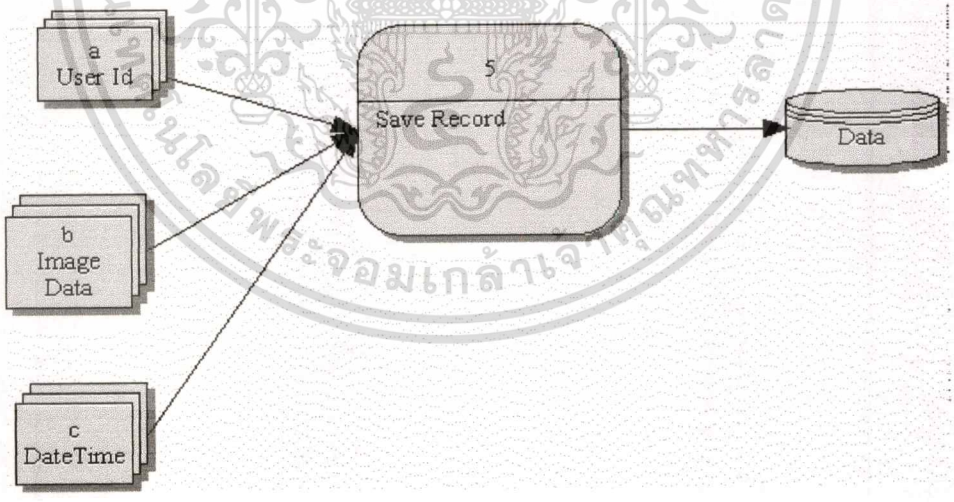
ภาพที่ 3.5 แสดงการถอดรหัสจากไฟล์เอกสารภาพ

โดยขั้นตอนการถอดรหัสข้อมูลสามารถแสดงได้ดังต่อไปนี้

```
For k = 1 to message length
    byte mask = 1
    for i = 1 to 8
        pick random position in image and random pixel for decode
        get the pixel's color components
        get the stored value
        byte mask = byte mask *2
    next i
    message =  message + stored value
next k
```

4. ขั้นตอนการบันทึกข้อมูล

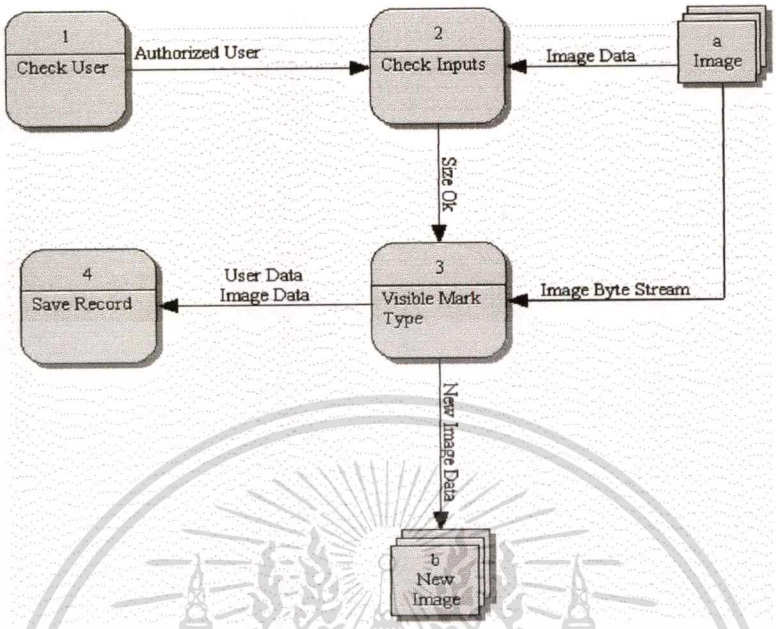
ขั้นตอนนี้จะเป็นการบันทึกข้อมูลการเข้าใช้งานโปรแกรมประยุกต์ของผู้ใช้ โดยจะทำการบันทึก ชื่อผู้ใช้, วันเวลาที่ใช้งาน ดังแสดงขั้นตอนการทำงานในภาพที่ 3.6



ภาพที่ 3.6 แสดงขั้นตอนการบันทึกรายละเอียดการใช้งานโปรแกรมประยุกต์

3.2.2 Visible Watermark

เป็นการทำหมายเหตุให้กับเอกสาร โดยสามารถทำได้ดังนี้



ภาพที่ 3.7 แสดงขั้นตอนการทำ Visible Watermark กับไฟล์เอกสารภาพ

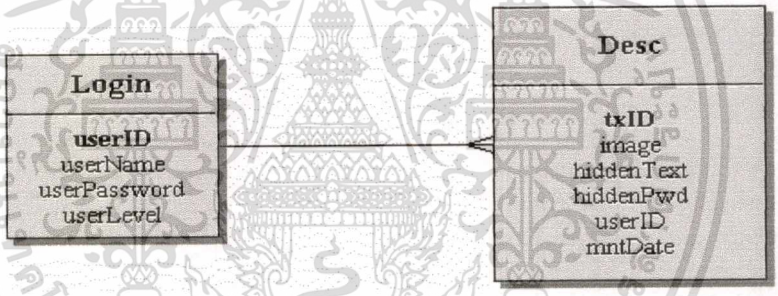
ในที่นี้จะกล่าวถึงเฉพาะขั้นตอนที่ 3 คือ ขั้นตอนการใส่ลายน้ำดิจิตอลลงไปในไฟล์เอกสารภาพ (Add Visible Mark) ส่วนขั้นตอนอื่น ๆ ได้กล่าวไปแล้วในหัวข้อข้างต้น โดยประเภทของการทำลายน้ำดิจิตอลแบบสังเกตเห็นได้นั้นสามารถทำได้ดังนี้

- สามารถวาดเส้น Freehand ลงไปบนเอกสารอิเลคทรอนิคได้ เลือกสีและขนาดของเส้นได้
- สามารถทำการเน้นเฉพาะส่วนบนเอกสารอิเลคทรอนิคได้ (Highlighter) โดยเลือกสีที่จะทำการเน้นได้
- สามารถวาดเส้นตรงเพื่อเป็นการเน้นข้อความบนเอกสารอิเลคทรอนิคได้ เลือกสีและขนาดของเส้นได้
- สามารถวาดกรอบสี่เหลี่ยมลงไปบนเอกสารอิเลคทรอนิคได้ โดยเลือกสีและขนาดของเส้นได้
- สามารถวาดสี่เหลี่ยมทึบ เพื่อใช้เป็นพื้นหลังสำหรับการพิมพ์ข้อความลงไปภายหลังได้ โดยเลือกสีได้
- สามารถพิมพ์ตัวอักษรลงไปบนเอกสารอิเลคทรอนิคได้ โดยเลือกรูปแบบตัวอักษร สี และขนาดได้

- สามารถใส่ข้อความหมายเหตุลงในเอกสารอิเลคทรอนิคได้ โดยจะมีพื้นหลังที่สามารถเลือกสี และตัวอักษรสามารถเลือกรูปแบบ สี และขนาดได้
- สามารถเปิดแฟ้มข้อมูลที่เป็นข้อความลงบนในเอกสารอิเลคทรอนิคได้ ซึ่งสามารถเลือกรูปแบบของข้อความที่จะให้แสดงลงไปบนเอกสารได้ โดยสามารถเลือกคุณสมบัติของตัวอักษรได้ทั้งขนาด รูปแบบ สี ตัวอักษรห้อย ตัวอักษรยก หรือขีดเส้นใต้ได้
- สามารถประทับตราซึ่งจะเป็นตัวอักษรหรือรูปภาพลงไปบนเอกสารอิเลคทรอนิคได้ โดยสามารถเพิ่ม หรือลบคุณสมบัติต่าง ๆ ได้ เลือกประทับตราโดยระบุวันและเวลาได้

3.2.3 การออกแบบฐานข้อมูล

เนื่องจากโปรแกรมประยุกต์ที่จัดทำขึ้นนี้มีเรื่องของการบันทึกรายละเอียดการใช้งานโปรแกรมจึงจำเป็นต้องมีฐานข้อมูลเข้ามาเพื่อบันทึกรายละเอียด และสามารถแสดงรายงานการใช้งานโปรแกรมได้ การออกแบบฐานข้อมูลสามารถแสดงได้ดังภาพที่ 3.8



ภาพที่ 3.8 แสดง ER-Diagram ระหว่างตาราง Login และตาราง Desc

จากภาพสามารถอธิบายได้ดังนี้ คือ ผู้ใช้หนึ่งคนสามารถทำลายน้ำดิจิตอลกับภาพได้หลายภาพด้วยกันความสัมพันธ์ระหว่างตาราง Login กับตาราง Desc จึงเป็นแบบ one-to-many และรายละเอียดของแต่ละตารางแสดงได้ดังตาราง

ตารางที่ 3.1 แสดง Data Dictionary ของตาราง Login

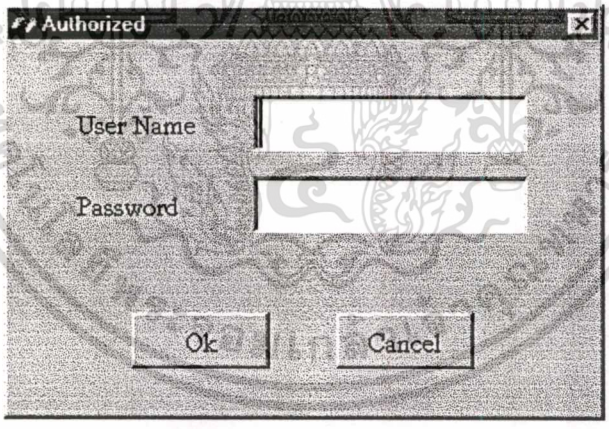
ID	Item (English)	Item (Thai)	Unique	Type	Domain		Remark
					L	Null	
1	**Login**	**ผู้ใช้**					
	UserID	รหัสผู้ใช้	PK	integer		NN	Auto Running
	UserName	ชื่อผู้ใช้		varchar	10	NN	
	UserPassword	รหัสผู้ใช้		varchar	10	NN	

บทที่ 4

การออกแบบโปรแกรมส่วนติดต่อกับผู้ใช้งาน

ในการออกแบบโปรแกรมประยุกต์ในการส่งเอกสารแฟกซ์ที่ได้มีการซ่อนข้อมูลลงไปนั้น การออกแบบส่วนติดต่อกับผู้ใช้งานนั้นได้มีการออกแบบให้สามารถใช้งานได้ง่าย และสะดวก เช่น ในส่วนของปุ่มต่าง ๆ เมื่อนำเม้าส์ไปวางที่ปุ่มจะมีคำอธิบายประกอบว่าปุ่มนี้คืออะไร ในส่วนของการป้อนข้อมูลจะมีชื่อบอกว่าจะต้องกรอกอะไรลงไป โดยโปรแกรมประยุกต์การซ่อนข้อมูลลงไปในไฟล์เอกสารแฟกซ์นี้ประกอบไปด้วยฟอร์มต่าง ๆ ดังต่อไปนี้

1. ฟอร์มตรวจสอบผู้เข้าใช้งานโปรแกรมประยุกต์ เป็นฟอร์มแรกเมื่อมีการเรียกโปรแกรมขึ้นมาใช้งาน ในส่วนติดต่อกับผู้ใช้งานนั้นได้แสดงดังภาพที่ 4.1ส่วนข้อกำหนดการใช้งานโปรแกรมได้มีการออกแบบไว้ดังแสดงในตารางที่ 4.1



ภาพที่ 4.1 แสดงแบบฟอร์มตรวจสอบผู้ใช้งาน (Authorized Form)

ตารางที่ 4.1 แสดง Program Specification **ของ** Authorized Form

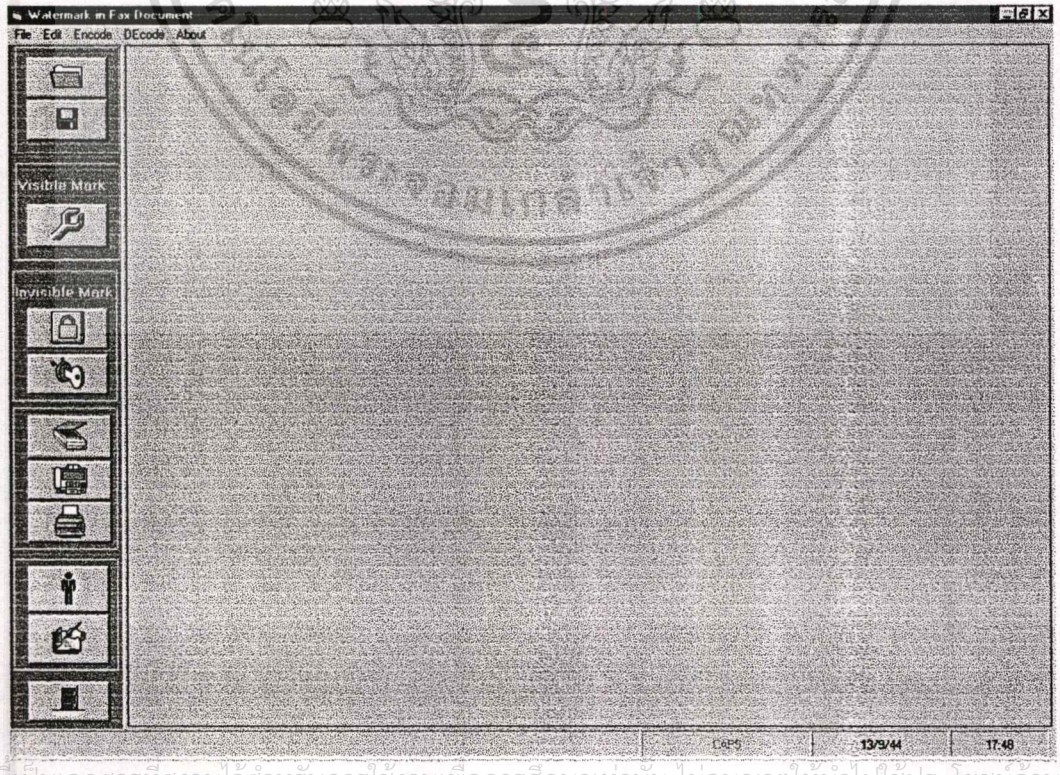
Program Specification	
Program Name : Data Hiding in Fax Document	
Form Name : Authorized Form	
Screen Caption	**Condition**

User Name	-TextBox : MaxLength = 10
Password	-MaskEdit : MaxLength = 10
OK	-Command Button : Check Valid *1
Cancel	-Command Button when click unload form

*1 - เมื่อกดปุ่ม Ok ให้ Check ค่า User Name,Password is not null และ check Valid กับตาราง Login ว่า User Name และ Password ถูกต้องหรือไม่และ userlevel คืออะไร ถ้าเป็น 1 หมายถึง admin ถ้าเป็น 0 จะหมายถึง user

- ถ้า Login ไม่ผ่านจะไม่สามารถเข้าใช้งานฟอร์มหลักได้

2. ฟอร์มหลัก (Main Form) จะประกอบไปด้วยฟอร์มย่อยดังจะได้กล่าวถึงรายละเอียดต่อไป ในฟอร์มหลักนี้จะสามารถทำการเปิดไฟล์ เซฟไฟล์ ซ่อนข้อมูลลงไปภายในภาพ การทำลายน้ำดิจิตอลแบบมองเห็นได้ การสแกนภาพ การพิมพ์ภาพ และการส่งแฟกซ์ได้ อีกทั้งยังได้จัดทำในส่วนของเมนูให้สามารถเลือกได้ และในส่วนของการทำงานในแต่ละเมนูจะเหมือนกันกับการทำงานของแต่ละปุ่ม ในที่นี้ได้เพิ่มในส่วนของเมนูที่ชื่อ About ซึ่งจะเป็นคำอธิบายรายละเอียดของโปรแกรม และผู้จัดทำ โดยภายในฟอร์มหลักจะประกอบไปด้วยส่วนต่าง ๆ ดังแสดงในภาพที่ 4.2 และส่วนของการทำงานของแต่ละปุ่มนั้นแสดงได้ในตารางที่ 4.2



ภาพที่ 4.2 แสดงส่วนประกอบของฟอร์มหลัก

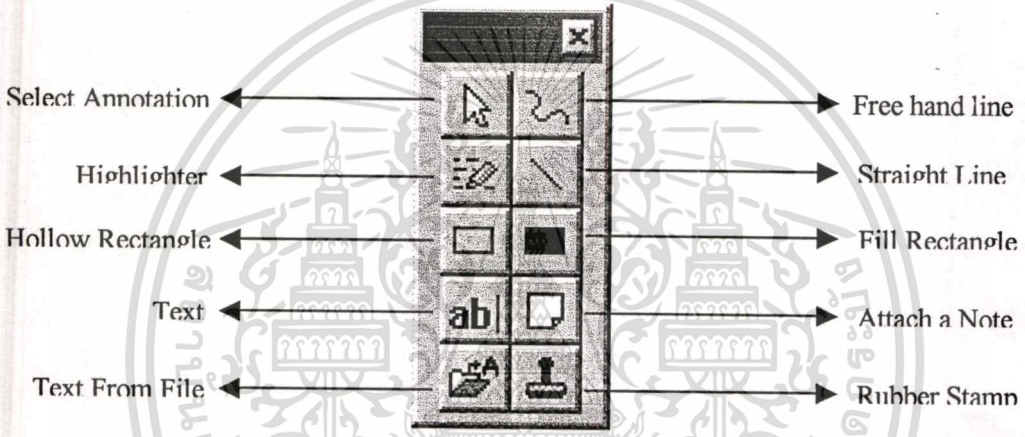
ตารางที่ 4.2 แสดง Program Specification ของฟอร์มหลัก

Program Specification	
Program Name : Data Hiding in Fax Document	
Form Name : Main Form	
Screen Caption	**Condition**
Open File	-Command Button : เป็นการเปิดไฟล์ภาพขึ้นแสดงบน screen ได้
Save File	-Command Button การทำงานสามารถเซฟไฟล์ที่แสดงอยู่บน screen ได้
Visible Mark	-Command Button : การทำงานจะเป็นการเรียกฟอร์มย่อยในการทำ Visible mark ขึ้นมา

Screen Caption	**Condition**
Encode	-Command Button : การทำงานจะทำการซ่อนข้อมูลลงไปในเอกสารที่แสดงอยู่บน Screen ได้โดยจะเป็นการเรียกฟอร์มย่อยในการเข้ารหัสขึ้นมา
Decode	-Command Button : การทำงานจะทำการถอดรหัสข้อมูลไฟล์เอกสารภาพที่แสดงอยู่บน Screen ได้โดยจะเป็นการเรียกฟอร์มย่อยในการเถอดรหัสขึ้นมา
Scan	-Command Button : การทำงานสามารถติดต่อกับเครื่องสแกนเนอร์เพื่อทำการสแกนภาพได้
Print	-Command Button : การทำงานสามารถพิมพ์เอกสารที่แสดงบน Screen ออกเครื่องพิมพ์ได้
Fax	-Command Button : การทำงานสามารถเรียกโปรแกรมที่จะใช้ส่งเอกสารแฟกซ์ได้
Set User	-Command Button : ก่อนที่จะเข้าโปรแกรมหลักนี้จะมีการตรวจสอบ User Level ก่อนในส่วนของฟอร์มตรวจสอบผู้ใช้งาน ถ้าหาก User Level มีค่าเป็น 1 ซึ่งหมายความว่าเป็น admin ปุ่มนี้จะสามารถใช้งานได้ โดยจะเป็นการเรียกฟอร์มย่อยขึ้นมา

Report	-Command Button : จะเป็นรายงานที่แสดงรายละเอียดการใช้งานโปรแกรมประยุกต์นี้
Exit	-Command Button : การทำงานจะเป็นการออกจากโปรแกรม

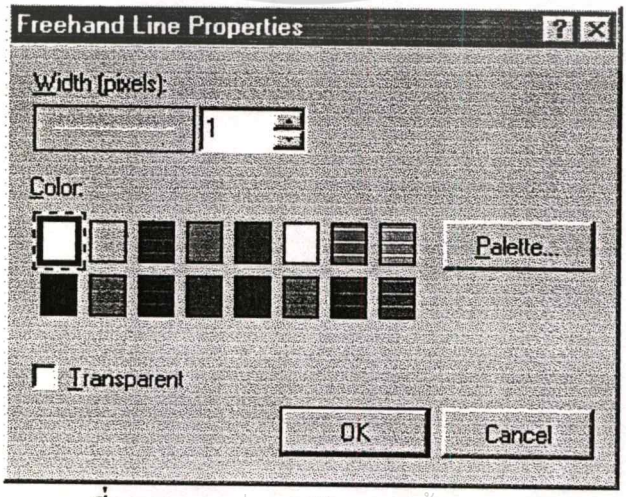
1. **ฟอร์มการทำลายน้ำดิจิตอลที่สามารถมองเห็นได้** (Visible Watermark Form) เมื่อมีการกดปุ่ม Visible Watermark จะปรากฏฟอร์มนี้ขึ้นมา ดังแสดงในภาพที่ 4.3



ภาพที่ 4.3 แสดง ฟอร์มการทำลายน้ำดิจิตอลที่สามารถมองเห็นได้

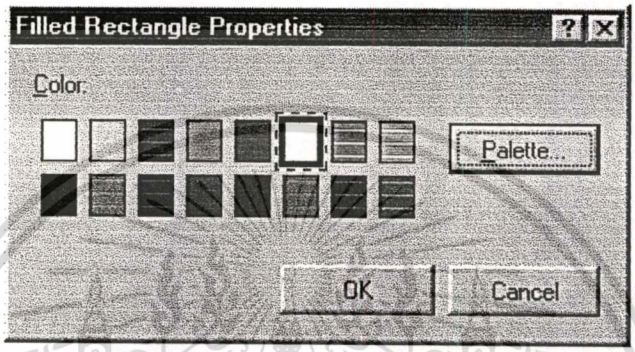
โดยในแต่ละปุ่มจะมีคุณสมบัติต่าง ๆ ที่เราสามารถตั้งค่าได้ดังต่อไปนี้

- Free hand Line เมื่อคลิกเม้าส์ปุ่มขวาจะแสดงคุณสมบัติของเส้นโดย สามารถเลือกความหนาของเส้น หรือสีได้ ดังภาพที่ 4.4



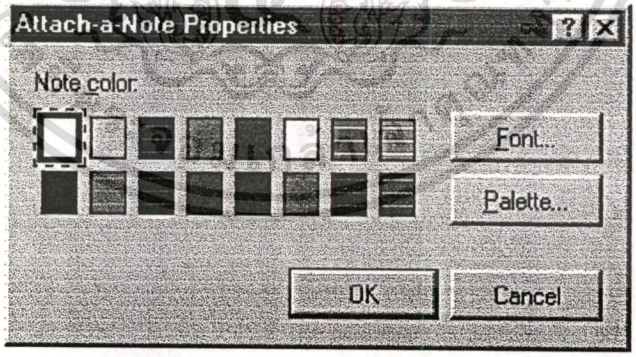
ภาพที่ 4.4 แสดงคุณสมบัติของ Free hand line

- เส้นตรง (Straight Line) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้เหมือนกันกับ Free hand line
- สี่เหลี่ยมทึบ (Fill Rectangle) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกสีได้ ดังแสดงในภาพที่ 4.5



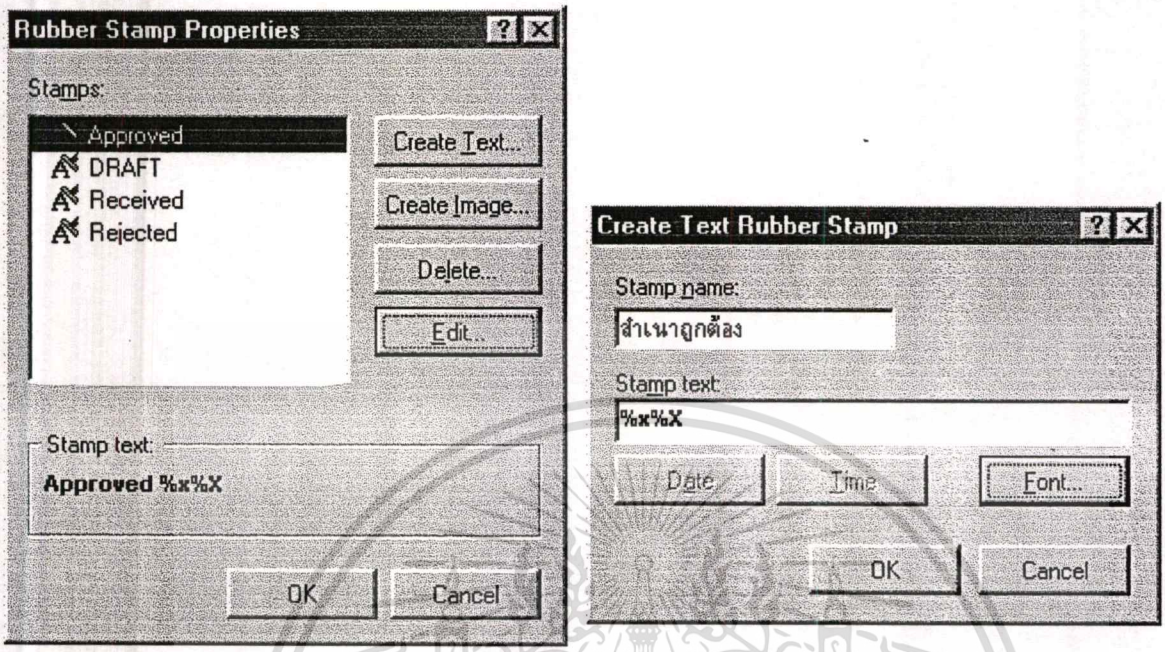
ภาพที่ 4.5 แสดงฟอร์มคุณสมบัติของสี่เหลี่ยมทึบ (Filled Rectangle Properties)

- Attach a Note เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกสีและคุณสมบัติของตัวอักษรได้ดังแสดงในภาพที่ 4.6



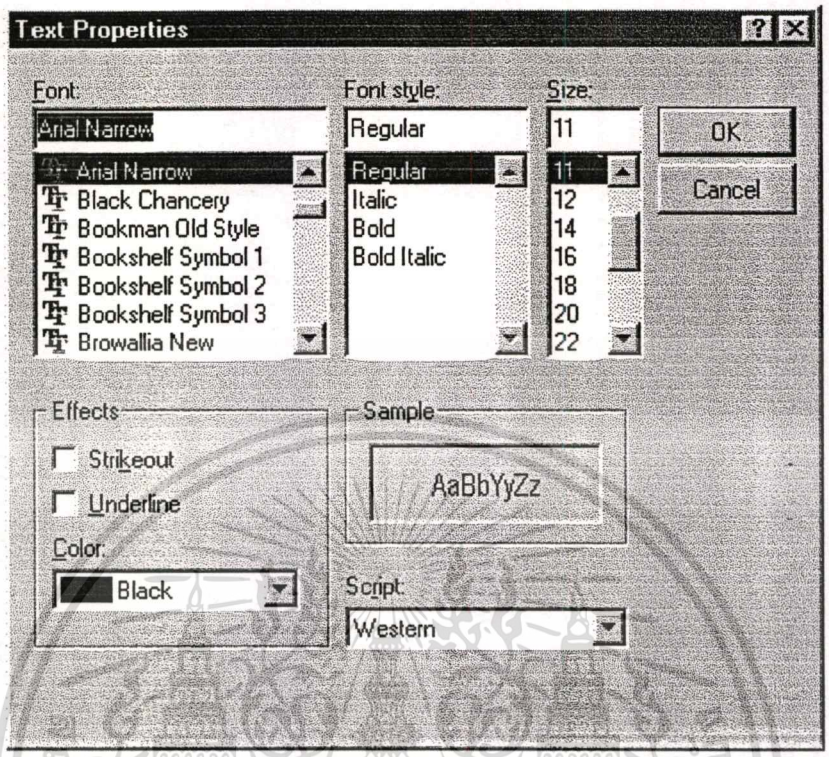
ภาพที่ 4.6 แสดงฟอร์มคุณสมบัติของการเขียนบันทึก (Attach-a-Note Properties)

- การทำตรายาง (Rubber Stamp) จะเหมือนกับการปั๊มตรายางลงบนเอกสาร โดยสามารถสร้างข้อความขึ้นมาเองได้ และเลือกสี เลือกขนาดตัวอักษร ได้ดังแสดงในภาพที่ 4.7



ภาพที่ 4.7 แสดงคุณสมบัติของตรายาง (Rubber Stamp Properties)

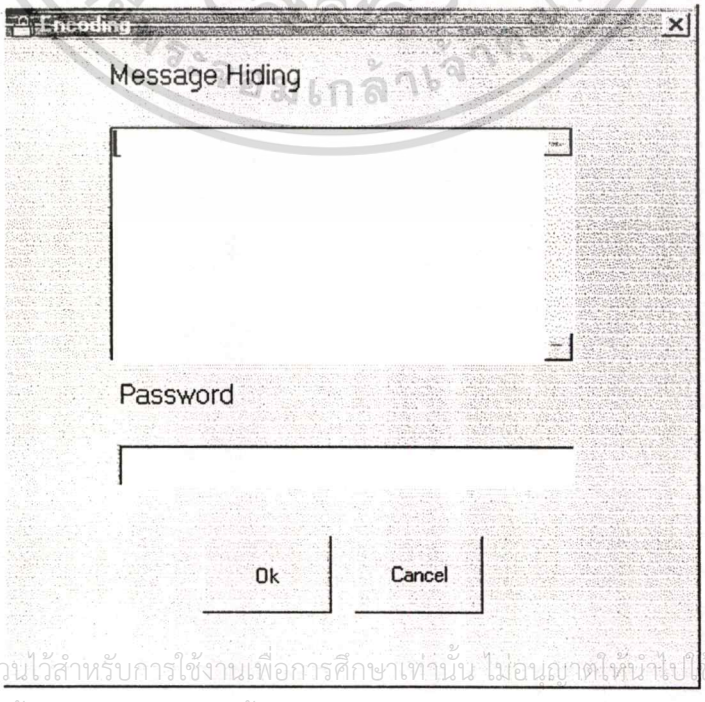
- การเน้นข้อความ (Highlighter) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกสีได้เหมือนกับคุณสมบัติของสี่เหลี่ยมทึบในภาพที่ 4.5
- กรอบสี่เหลี่ยม (Hollow Rectangle)) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกสีและขนาดความหนาของเส้นได้เหมือนกับคุณสมบัติของ Free Hand Lineในภาพที่ 4.4
- การพิมพ์ตัวอักษร (Text)) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกคุณสมบัติของตัวอักษรได้ดังแสดงในภาพที่ 4.8
- การนำแฟ้มข้อมูลตัวอักษรมาใส่ในภาพ (Text From File) เมื่อคลิกเม้าส์ปุ่มขวาก็จะแสดงฟอร์มให้เลือกคุณสมบัติได้โดยสามารถเลือกคุณสมบัติของตัวอักษรได้เหมือนกับคุณสมบัติของตัวอักษรดังแสดงในภาพที่ 4.8



ภาพที่ 4.8 แสดงการนำแฟ้มข้อมูลมาใส่ในภาพ

1. **ฟอร์มเข้ารหัส (Encode Form)**

เป็นฟอร์มย่อยของฟอร์มหลัก เมื่อคลิกที่ปุ่ม Encode Form จะปรากฏฟอร์มนี้ขึ้นมาดังแสดงในภาพที่ 4.9 ส่วน Program specification แสดงดังได้ตารางที่ 4.3



ภาพที่ 4.9 แสดงฟอร์มเข้ารหัส

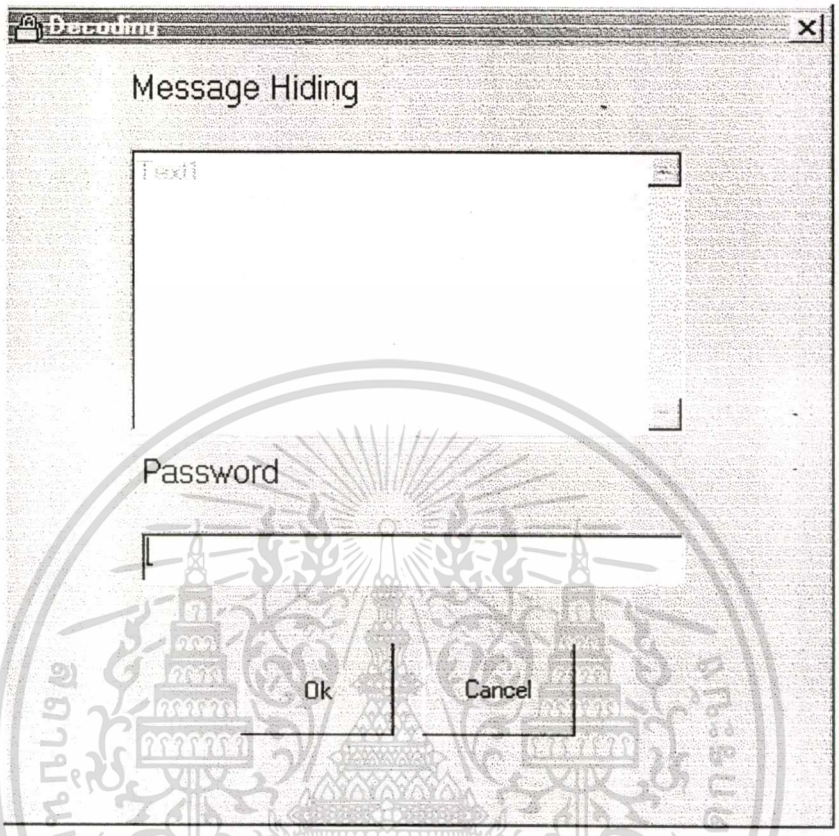
ตารางที่ 4.3 แสดง Program Specification **ของฟอร์มเข้ารหัส**

Program Specification	
Program Name : Data Hiding in Fax Document	
Form Name : Encoding Form	
Screen Caption	**Condition**
Message Hiding	-TextBox : MaxLength = 255
Password	-TextBox : MaxLength = 10
OK	-Command Button : Condition*1
Cancel	-Command Button when click unload form

*1 จะทำการเข้ารหัสข้อมูลโดยทำการซ่อนข้อมูลลงไปในภาพในแบบที่สังเกตด้วยตาไม่เห็น และจะเก็บรหัสผ่านไว้ในภาพด้วย เพื่อใช้ในการถอดรหัส ซึ่งเป็นการเข้ารหัสแบบ Secret Key

2. ฟอร์มถอดรหัสข้อมูล

เป็นฟอร์มย่อยของฟอร์มหลักเมื่อคลิกที่ปุ่ม Decode จะปรากฏฟอร์มถอดรหัสข้อมูลขึ้นมาใช้เพื่อเป็นการตรวจสอบภาพที่ส่งมาว่าเป็นภาพที่มีการเข้ารหัสข้อมูลอะไรไว้ในภาพ โดยใส่รหัสผ่านให้ถูกต้องตรงกับภาพ โปรแกรมจะทำการถอดรหัสและแสดงข้อมูลที่ซ่อนไว้ได้ รูปแบบส่วนติดต่อกับผู้ใช้งานแสดงได้ดังภาพที่ 4.10 และ Program Specification แสดงในตารางที่ 4.4



ภาพที่ 4.10 แสดงฟอร์มถอดรหัสข้อมูล (Decoding Form)

ตารางที่ 4.4 แสดง Program Specification ของฟอร์มถอดรหัส

Program Specification	
Program Name : Data Hiding in Fax Document	
Form Name : Decoding Form	
Screen Caption	**Condition**
Message Hiding Password OK Cancel	-TextBox : MaxLength = 255 Enable = False -TextBox : MaxLength = 10 -Command Button : Condition*1 -Command Button when click unload form

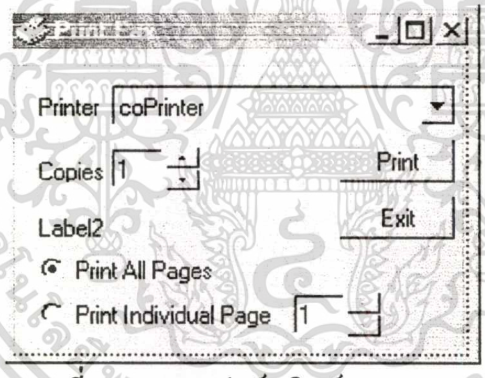
*1 เมื่อกดปุ่ม Ok จะทำการตรวจสอบรหัสผ่านว่าถูกต้องหรือไม่โดยอาศัยข้อมูลที่ซ่อนอยู่ในภาพ จากนั้นจะทำการถอดรหัสข้อมูลที่ซ่อนไว้ออกมาแสดงในส่วนของ Message Hiding

2. ฟอร์มสแกน (Scan Form)

เมื่อกดปุ่มโปรแกรมจะตรวจสอบไฟล์ Twain32.dll โดยจะเป็นการเรียกใช้ API ของสแกนเนอร์ที่ต่อกับเครื่องนั้น ๆ ในที่นี้ใช้เครื่องสแกนของยี่ห้อ Hewlett Packard ก็จะแสดงโปรแกรมสแกนของ HP ขึ้นมา เป็นต้น

3. ฟอร์มพิมพ์ภาพ (Print Form)

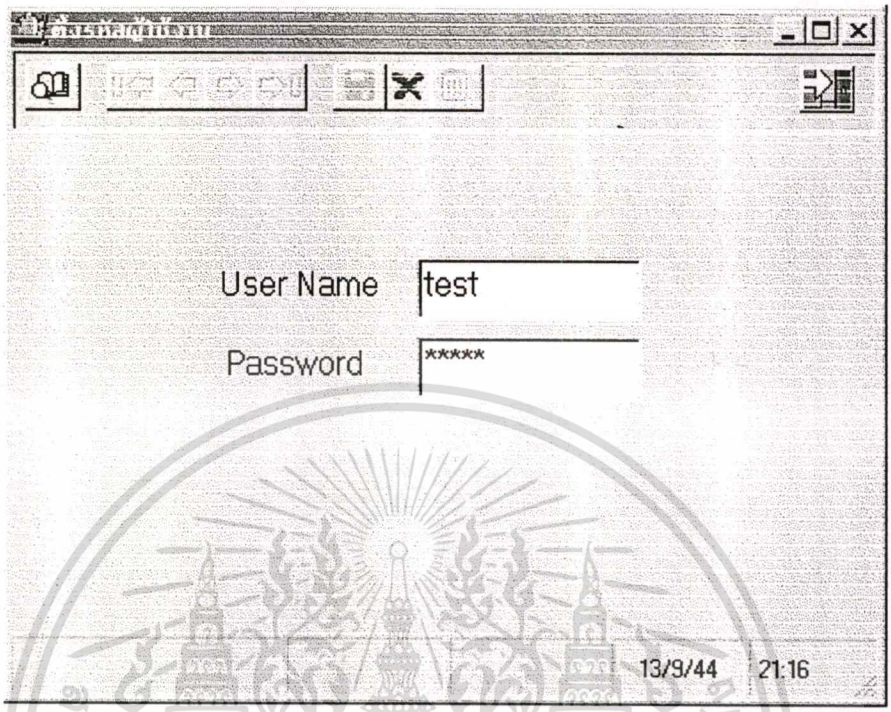
เมื่อกดปุ่ม Print จะปรากฏฟอร์มการพิมพ์ภาพขึ้นมาดังแสดงในภาพที่ 4.11



ภาพที่ 4.11 แสดงฟอร์มพิมพ์ (Print Form)

4. ฟอร์มตั้งค่าผู้ใช้งานโปรแกรม

เมื่อกดปุ่ม Set User จะปรากฏฟอร์มตั้งค่าผู้ใช้งานขึ้นมา ดังแสดงในภาพที่ 4.12 และ Program Specification แสดงได้ดังตารางที่ 4.5



ภาพที่ 4.12 แสดงฟอร์มตั้งรหัสผู้ใช้งาน (Set user form)

ตารางที่ 4.5 แสดง Program Specification ของฟอร์มตั้งรหัสผู้ใช้

Program Specification	
Program Name : Data Hiding in Fax Document	
Form Name : Set User Form	
Screen Caption	**Condition**
Query	-Command Button : สามารถเรียกดูรายชื่อผู้ใช้ได้ทั้งหมด
Next	-Command Button : เป็นการเลื่อนตำแหน่งข้อมูลไป 1 รายการ
Last	-Command Button : เป็นการเลื่อนตำแหน่งข้อมูลไปที่รายการสุดท้าย
Previous	-Command Button : เป็นการเลื่อนตำแหน่งข้อมูลไปที่รายการก่อนหน้านี้ 1 รายการ
First	-Command Button : เป็นการเลื่อนตำแหน่งข้อมูลไปที่รายการแรก

Save	-Command Button : เป็นการบันทึกข้อมูลลงฐานข้อมูลที่ตาราง Login
Delete	-Command Button : เป็นการลบข้อมูล ณ รายการที่แสดงอยู่ออกไป
Exit	-Command Button : ออกจากโปรแกรม
User Name	-TextBox : Max Length = 10 Not Null
Password	-MaskEdit : Max Length = 10 Not Null

5. ฟอร์มรายงานการเข้าใช้โปรแกรม

เมื่อคลิกที่ปุ่ม report จะแสดงรายงานการเข้าใช้โปรแกรมประยุกต์นี้ โดยแสดงใน program specification ในตารางที่ 4.6

ตารางที่ 4.6 แสดงฟอร์มรายงาน

Program Specification
Program Name : Data Hiding in Fax Document
Form Name : Report
Detail
Date : dd/mm/yyyy hh:mm:ss **User Name : X-----10-----X** **File Name : X----------50----------X** **Message Hiding : X----------255----------X** **Password : X----10-----X**

บทที่ 5

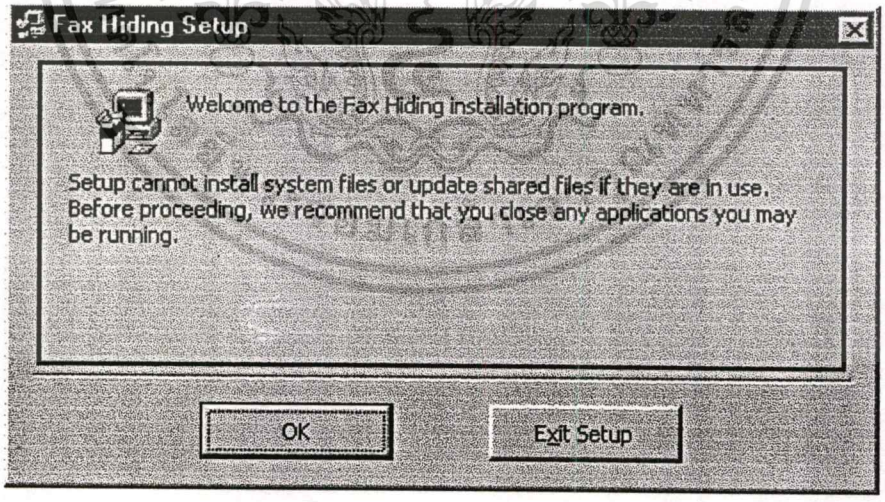
การทดลอง

5.1 สภาพแวดล้อมที่ใช้ในการทดลอง

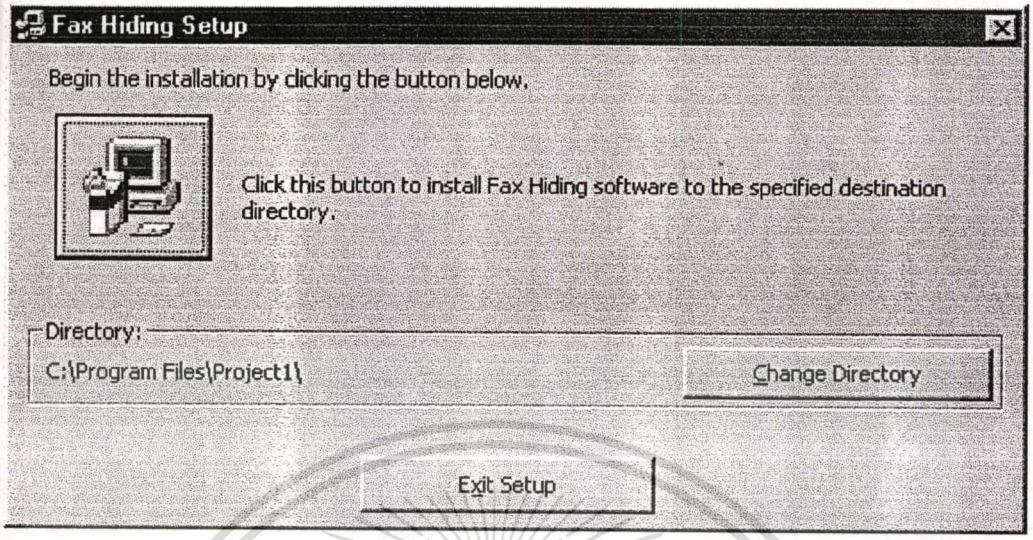
1. คอมไฟล์เลอร์ที่ใช้ในการพัฒนาโปรแกรมประยุกต์เพื่อซ่อนข้อมูลลงไปในเอกสารแฟกซ์ (ในที่นี้จะเรียกว่าโปรแกรม Fax Hiding) ใช้ Microsoft Visual Basic Version 6.0
2. ซอฟแวร์ที่นำมาใช้ทดลองในการส่งเอกสารแฟกซ์ผ่านโมเด็มคือ Impact Color Fax Lite และ Mighty Fax
3. คอมพิวเตอร์ 2 เครื่อง ที่มีโมเด็ม 56 K ทั้งสองเครื่อง
4. Operating System ที่ใช้ทำการทดลองนี้ใช้ Windows98

5.2 ขั้นตอนการติดตั้งโปรแกรม

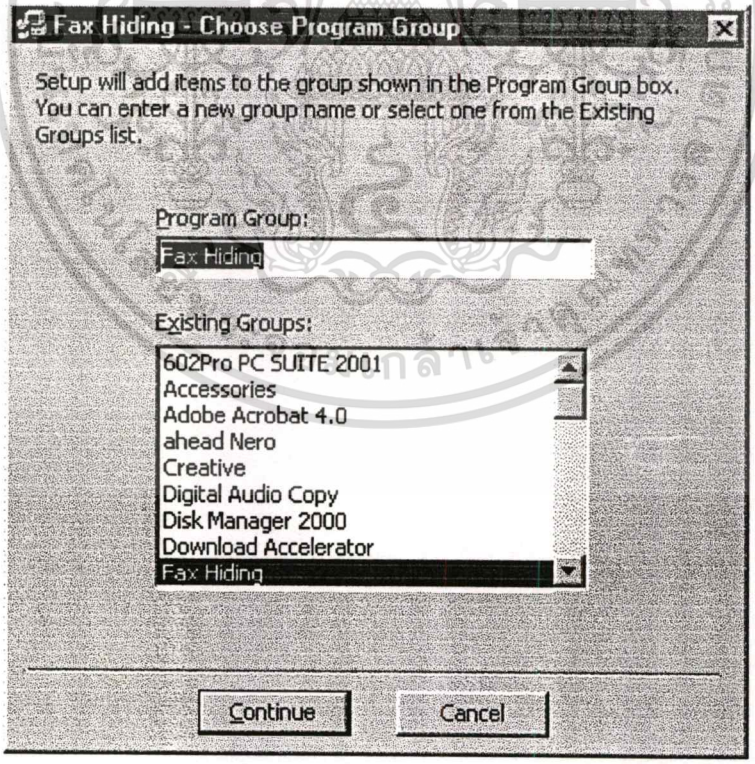
เมื่อดับเบิ้ลคลิกไฟล์ Setup.exe แล้วจะปรากฏขั้นตอนการลงโปรแกรมดังภาพตามลำดับต่อไปนี้



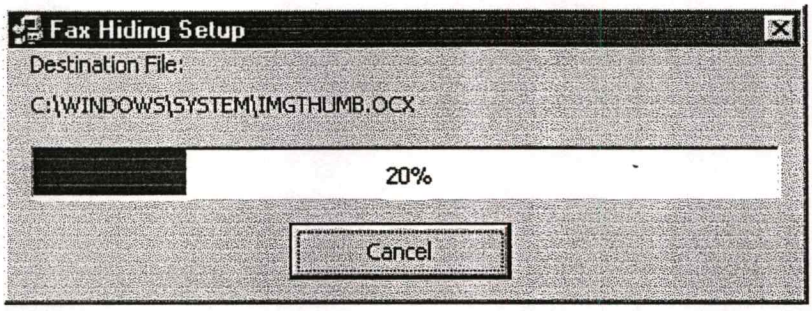
ภาพที่ 5.1 แสดงขั้นตอนการติดตั้งโปรแกรมประยุกต์ขั้นที่1



ภาพที่ 5.2 แสดงขั้นตอนการติดตั้งโปรแกรมประยุกต์ขั้นที่2



ภาพที่ 5.3 แสดงขั้นตอนการติดตั้งโปรแกรมประยุกต์ขั้นที่3



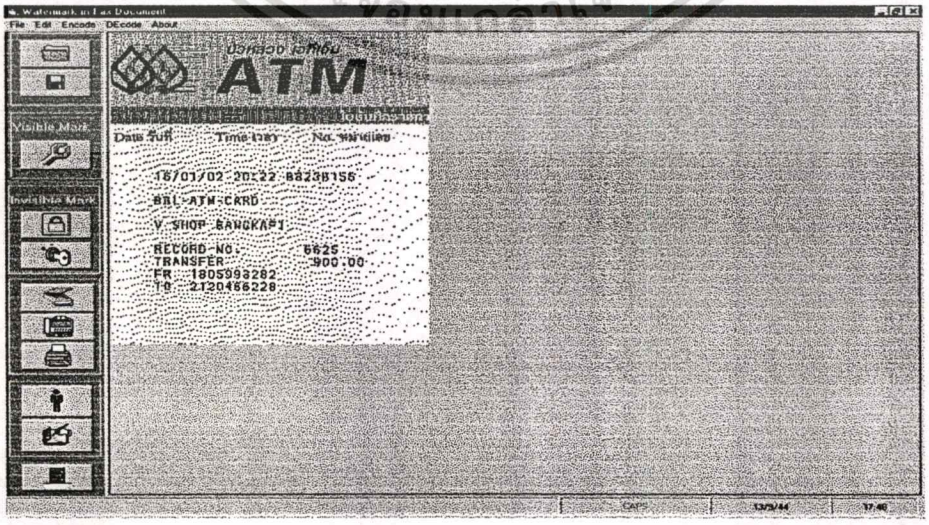
ภาพที่ 5.4 แสดงขั้นตอนการติดตั้งโปรแกรมประยุกต์ขั้นที่ 4

ภาพที่ 5.5 แสดงขั้นตอนการติดตั้งโปรแกรมประยุกต์ขั้นสุดท้าย

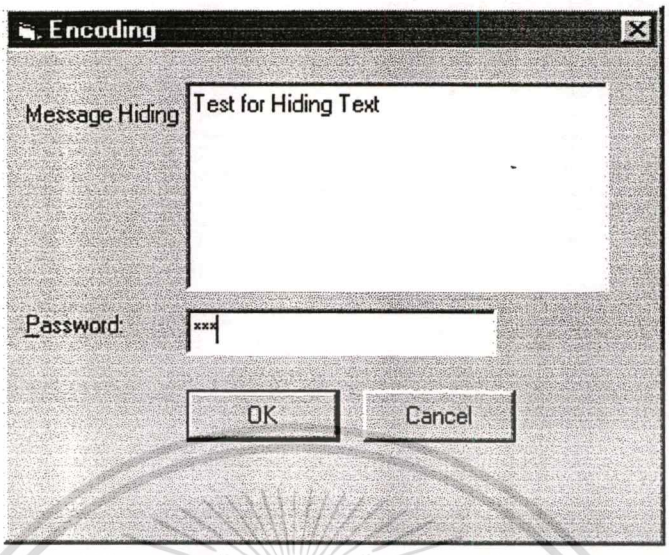
5.3 วิธีทำการทดลอง

ทางฝั่งผู้ส่ง

1. เรียกโปรแกรม Fax Hiding ขึ้นมาเพื่อทำการสแกนเอกสารและทำการบันทึกไฟล์ภาพ
2. เปิดไฟล์ภาพที่ได้ทำการบันทึกไว้ด้วยโปรแกรม Fax Hiding เพื่อนำมาทำการเข้ารหัสข้อมูลลงไปในไฟล์เอกสารภาพ แล้วทำการบันทึกไฟล์เอกสารภาพที่ได้ทำการเข้ารหัสเรียบร้อยแล้วดังแสดงในภาพที่ 5.1



ภาพที่ 5.6แสดงส่วนของโปรแกรมซ่อนข้อมูลลงไปในเอกสารภาพเพื่อทำการเข้ารหัสก่อนทำการส่งผ่านแฟกซ์โมเด็ม

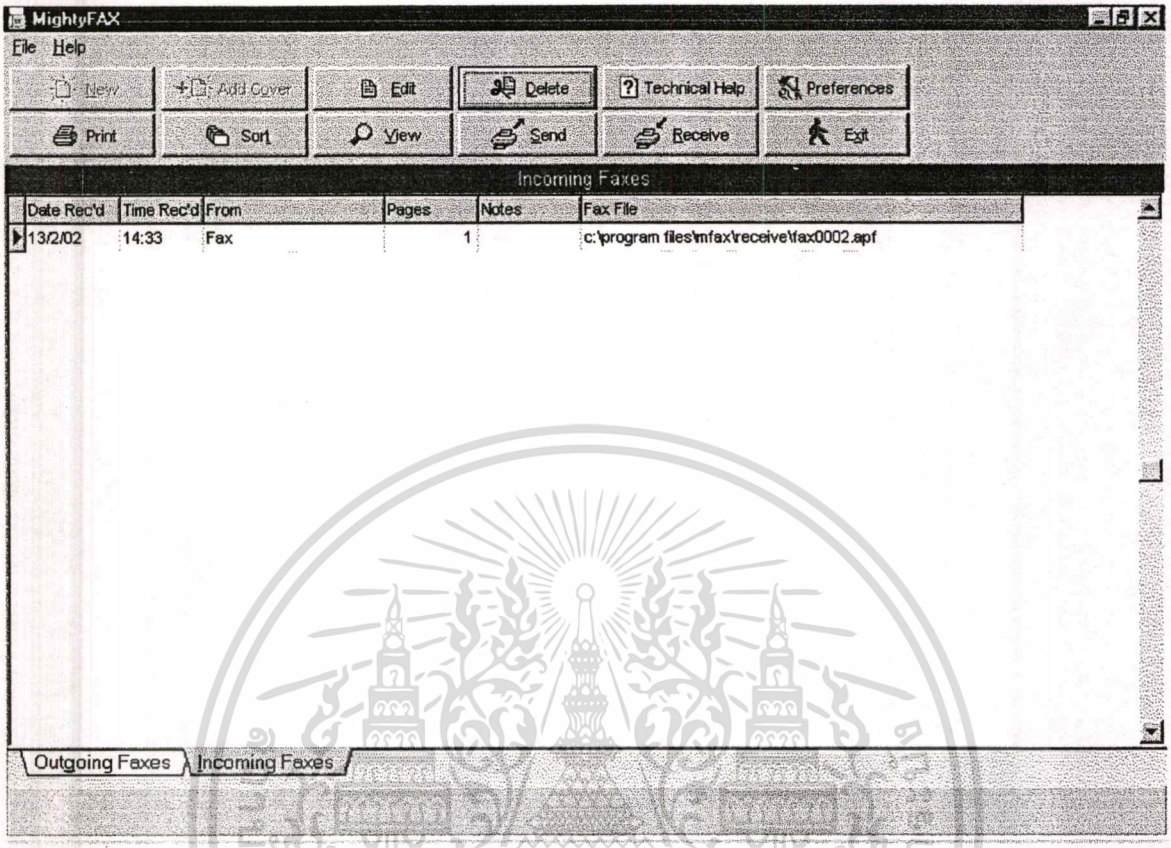


ภาพที่ 5.7แสดงฟอร์มการเข้ารหัสข้อความที่ต้องการซ่อนลงไปในเอกสารภาพ

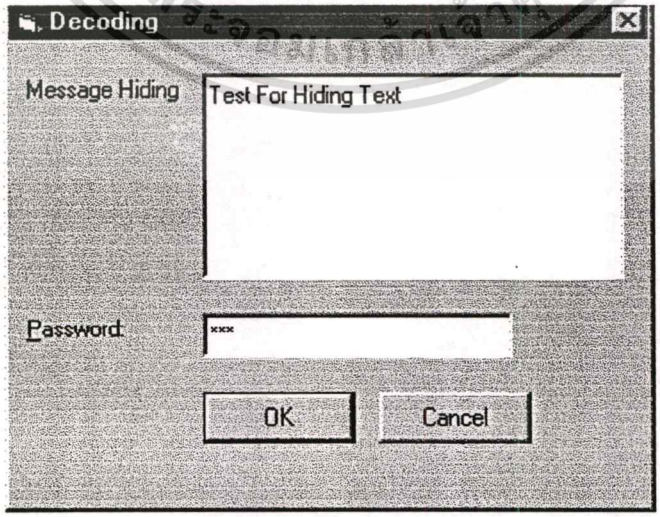
1. ทำการส่งไฟล์เอกสารภาพนี้โดยโปรแกรม Fax Hiding จะทำการเรียกโปรแกรม Impact Color Fax Lite ขึ้นมาเพื่อส่งไฟล์เอกสารภาพทางโมเด็มดังแสดงในภาพที่ 5.2

ภาพที่ 5.8 แสดงส่วนของโปรแกรมที่ใช้ในการส่งเอกสารแฟกซ์ผ่านทางแฟกซ์โมเด็มในที่นี้คือโปรแกรม Inpact ColorFax Lite

3. ทำการบันทึกเอกสารที่ได้รับมาให้อยู่ในรูปของไฟล์บิตแมพ
4. ทำการเปิดไฟล์เอกสารภาพที่ได้รับมาจากฝั่งผู้ส่งด้วยโปรแกรม Fax Hiding
5. ใส่รหัสลงไปในช่องรหัส เพื่อทำการถอดข้อมูลที่ถูกซ่อนมากับไฟล์เอกสารภาพ



ภาพที่ 5.9 แสดงส่วนของโปรแกรมที่ใช้ในการรับกสารแฟกซ์ผ่านทางแฟกซ์โมเด็มในที่นี้คือโปรแกรม MightyFax



ภาพที่ 5.10 แสดงฟอร์มการถอดรหัสข้อมูลซึ่งถ้าหากใส่รหัสผ่านได้ตรงกับที่ซ่อนไว้ในภาพก็จะแสดงข้อความที่ซ่อนไว้ออกมาได้อย่างถูกต้อง

บทที่ 6
สรุป

ในการศึกษาการพัฒนาระบบโปรแกรมใช้งานเพื่อส่งเอกสารแฟกซ์นั้น เนื่องจากว่าวิธีการซ่อนข้อมูลในเอกสารภาพที่ได้ศึกษามา เช่น การซ่อนข้อมูลโดยวิธี Line-Shift coding, Word-Shift coding, Feature coding ยังมีข้อจำกัดสำหรับเอกสารภาพที่มีรูปภาพปะปนไปกับตัวอักษร แต่จะมีข้อดีคือสามารถทนต่อการทำสำเนาซ้ำได้ ส่วนวิธีการทำลายน้ำดิจิตอลสามารถกระทำได้กับเอกสารภาพที่มีภาพปะปนอยู่ได้ และเหมาะสำหรับเอกสารที่ส่งไปโดยอยู่ในรูปของแฟ้มข้อมูล และเปิดอ่านโดยไม่ผ่านการพิมพ์ และยังสามารถนำมาประยุกต์ใช้ในการซ่อนข้อมูลลงไปในเอกสารภาพที่เก็บอยู่ในรูปของข้อมูลดิจิตอลได้ ประโยชน์ก็คือ สามารถส่งไฟล์ข้อมูลดิจิตอลนี้ไปยังจดหมายอีเลคทรอนิคส์ของผู้รับได้ด้วย ดังนั้นโปรแกรมประยุกต์ที่พัฒนาสามารถตรวจสอบเอกสารที่มีการซ่อนข้อมูลได้โดยตรง ซึ่งวิธีการเข้ารหัสทำได้โดยซ่อนข้อมูลลงไปใน Least significant bit ภายในเอกสารภาพ โดยทำการแปลงค่าข้อความและรหัสผ่านที่จะทำการซ่อนลงไปในค่าของคอมโพเน้นของสีภายในเอกสารภาพ โดยทำการซ่อนแต่ละบิตข้อความลงไปในพิกเซลของคอมโพเน้นของสีซึ่งไม่ซ้ำกัน

ประโยชน์ที่จะได้รับ

สามารถซ่อนข้อมูลลงไปในไฟล์เอกสารที่จะส่งแฟกซ์ได้ สามารถใช้เพื่ออ้างสิทธิ์ความเป็นเจ้าของเอกสารได้ อีกทั้งยังสามารถประยุกต์ใช้ในการส่งไฟล์ข้อมูลดิจิตอลนี้ไปยังจดหมายอีเลคทรอนิคส์ของผู้รับได้ด้วย ดังนั้นโปรแกรมประยุกต์ที่พัฒนาสามารถตรวจสอบเอกสารที่มีการซ่อนข้อมูลได้โดยตรง โดยสามารถแสดงความเป็นเจ้าของเอกสารได้ทั้งในแบบที่สามารถมองเห็นได้ด้วยตาเปล่า และไม่สามารถมองเห็นได้ซึ่งต้องมีการเข้ารหัสและถอดรหัสข้อความที่ทำการซ่อนนี้ก็จะสามารถทราบได้ว่าข้อความที่ซ่อนมากับเอกสารจากฝั่งผู้ส่งมีข้อความว่าอะไร ส่วนข้อจำกัดของวิธีการนี้คือจะไม่ทนทานต่อการพิมพ์และการทำสำเนาเอกสารซ้ำ

บรรณานุกรม

An Introduction to Steganography [Online]. Available :
http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.html#SECTION00080000000000000000

Bit Manipulation [Online]. Available :
http://realbasic.zapkerpow.com/nug/2001/11/msg00573.html

Fax a What is Definition [Online]. Available :
http://whatis.techtarget.com/definition/0,,sid9_gci212098,00.html

Min Wu & other "Data Hiding in Digital Binary Image" New Jersey State R&D Excellence Award,NSF grant MIP-9408432,NSF REU grant for Princeton Summer Institute, and Intel Technology fot Education 2000 Grant

Min Wu, Edward Tang, Bede Liu "Data Hiding in Digital Binary Image" Electrical Engineering Dept.,Princeton Univ.,Princton,NJ 08544, Electrical & Computer Engineering Dept.,Johns Hopkins Univ.,Baltimore,MD 21218

Ping Wah Wong "Public-key Watermark for Image Verification and Authentication" Proceedings of International Conference on ICIP'98, vol 1,1998,99 455-459

S.H. Low,N.F. Maxemchuk, A.M. Lapone "Document Identification for Copyright Protection Using Centroid Detection" IEEE Transactions on Communication, to appear.

S.H. Low, N.F. Maxemchuk,J.T. Brassil, and L. O'Gorman "Document Marking and Identification Using Both Line and Word Shifting" Proceedings of Infocom'95, April 1995.

Simple Steganography Visual Basic Source Code [Online]. Available :
http://www.planetsourcecode.com/vb/scripts/ShowCode.asp?txtCodeId=24939&lngWId=1

W.Bender & other "Techniques for data hiding" IBM SYSTEM JOURNAL, VOL 35 NOS 3&4 (1996) pp.313-336

ภาคผนวก

- **Encoding Source Code in Visual Basic**

```
Private Sub EncodeByte(ByVal Value As Byte, ByVal used_positions As Collection, ByVal
wid As Integer, ByVal hgt As Integer)
Dim i As Integer
Dim byte_mask As Integer
Dim r As Integer
Dim c As Integer
Dim pixel As Integer
Dim clrr As Byte
Dim clrg As Byte
Dim clrb As Byte
Dim color_mask As Integer
  byte_mask = 1
  For i = 1 To 8
    PickPosition used_positions, wid, hgt, r, c, pixel
    UnRGB frmWatermark.picImage.Point(r, c), clrr, clrg, clrb
    If Value And byte_mask Then
      color_mask = 1
    Else
      color_mask = 0
    End If
    Select Case pixel
      Case 0
        clrr = (clrr And &HFE) Or color_mask
      Case 1
        clrg = (clrg And &HFE) Or color_mask
      Case 2
```

```
            clrb = (clrb And &HFE) Or color_mask
        End Select
        frmWatermark.picImage.PSet (r, c), RGB(clrr, clrg, clrb)
        byte_mask = byte_mask * 2
    Next i
End Sub
Private Function NumericPassword(ByVal password As String) As Long
Dim Value As Long
Dim ch As Long
Dim shift1 As Long
Dim shift2 As Long
Dim i As Integer
Dim str_len As Integer
    shift1 = 3
    shift2 = 17
    ' Process the message.
    str_len = Len(password)
    For i = 1 To str_len
        ' Add the next letter.
        ch = Asc(Mid$(password, i, 1))
        Value = Value Xor (ch * 2 ^ shift1)
        Value = Value Xor (ch * 2 ^ shift2)
        shift1 = (shift1 + 7) Mod 19
        shift2 = (shift2 + 13) Mod 23
    Next i
    NumericPassword = Value
End Function
Private Sub PickPosition(ByVal used_positions As Collection, ByVal wid As Integer, ByVal
hgt As Integer, ByRef r As Integer, ByRef c As Integer, ByRef pixel As Integer)
Dim position_code As String
```

```
    On Error Resume Next
    Do
        ' Pick a position.
        r = Int(Rnd * wid)
        c = Int(Rnd * hgt)
        pixel = Int(Rnd * 3)
        position_code = "(" & r & "," & c & "," & pixel & ")"
        used_positions.Add position_code, position_code
        If Err.Number = 0 Then Exit Do
        Err.Clear
    Loop
End Sub
Private Sub UnRGB(ByVal color As OLE_COLOR, ByRef r As Byte, ByRef g As Byte,
ByRef b As Byte)
    r = color And &HFF&
    g = (color And &HFF00&) \ &H100&
    b = (color And &HFF0000) \ &H10000
End Sub
Private Sub cmdEncode_Click()
Dim msg As String
Dim i As Integer
Dim used_positions As Collection
Dim wid As Integer
Dim hgt As Integer
Dim show_pixels As Boolean
    Screen.MousePointer = vbHourglass
    DoEvents
    Rnd -1
    Randomize NumericPassword(txtPassword.Text)
```

```
    wid = frmWatermark.picImage.ScaleWidth
    hgt = frmWatermark.picImage.ScaleHeight
    msg = Left$(txtMessage.Text, 255)
    Set used_positions = New Collection
    EncodeByte CByte(Len(msg)), _
        used_positions, wid, hgt
    For i = 1 To Len(msg)
        EncodeByte Asc(Mid$(msg, i, 1)), _
            used_positions, wid, hgt
    Next i
    frmWatermark.picImage.Picture = frmWatermark.picImage.Image
    MsgBox "Encoding Successfull"
    Screen.MousePointer = vbDefault
    Unload Me
End Sub
```

- **Decoding Source Code in Visual Basic**

```
Private Sub cmdDeocde_Click()
Dim msg_length As Byte
Dim msg As String
Dim ch As Byte
Dim i As Integer
Dim used_positions As Collection
Dim wid As Integer
Dim hgt As Integer
Dim show_pixels As Boolean
    Screen.MousePointer = vbHourglass
    DoEvents
    Rnd -1
    Randomize NumericPassword(txtPassword.Text)
```

```
    wid = frmWatermark.picImage.ScaleWidth
    hgt = frmWatermark.picImage.ScaleHeight
    Set used_positions = New Collection
    msg_length = DecodeByte(used_positions, wid, hgt)
    For i = 1 To msg_length
        ch = DecodeByte(used_positions, wid, hgt)
        msg = msg & Chr$(ch)
    Next i
    frmWatermark.picImage.Picture = frmWatermark.picImage.Image
    txtMessage.Text = msg
    Screen.MousePointer = vbDefault
End Sub
Private Function DecodeByte(ByVal used_positions As Collection, ByVal wid As Integer,
ByVal hgt As Integer) As Byte
Dim Value As Integer
Dim i As Integer
Dim byte_mask As Integer
Dim r As Integer
Dim c As Integer
Dim pixel As Integer
Dim clrr As Byte
Dim clrg As Byte
Dim clrb As Byte
Dim color_mask As Integer
    byte_mask = 1
    For i = 1 To 8
        PickPosition used_positions, wid, hgt, r, c, pixel
        UnRGB frmWatermark.picImage.Point(r, c), clrr, clrg, clrb
```

```
    Select Case pixel
      Case 0
        color_mask = (clrr And &H1)
      Case 1
        color_mask = (clrg And &H1)
      Case 2
        color_mask = (clrb And &H1)
    End Select
    If color_mask Then
      Value = Value Or byte_mask
    End If
    byte_mask = byte_mask * 2
  Next i
  DecodeByte = CByte(Value)
End Function
Private Function NumericPassword(ByVal password As String) As Long
Dim Value As Long
Dim ch As Long
Dim shift1 As Long
Dim shift2 As Long
Dim i As Integer
Dim str_len As Integer
  shift1 = 3
  shift2 = 17
  str_len = Len(password)
  For i = 1 To str_len
    ch = Asc(Mid$(password, i, 1))
    'Value = Value Xor (ch * 2 ^ shift1)
    Value = Value Xor (ch * 2 ^ shift2)
    shift1 = (shift1 + 7) Mod 19
```

```
    shift2 = (shift2 + 13) Mod 23
  Next i
  NumericPassword = Value
End Function
Private Sub PickPosition(ByVal used_positions As Collection, ByVal wid As Integer, ByVal
hgt As Integer, ByRef r As Integer, ByRef c As Integer, ByRef pixel As Integer)
Dim position_code As String
  On Error Resume Next
  Do
    r = Int(Rnd * wid)
    c = Int(Rnd * hgt)
    pixel = Int(Rnd * 3)
    position_code = "(" & r & "," & c & "," & pixel & ")"
    used_positions.Add position_code, position_code
    If Err.Number = 0 Then Exit Do
    Err.Clear
  Loop
End Sub
Private Sub UnRGB(ByVal color As OLE_COLOR, ByRef r As Byte, ByRef g As Byte,
ByRef b As Byte)
  r = color And &HFF&
  g = (color And &HFF00&) \ &H100&
  b = (color And &HFF0000) \ &H10000
End Sub
```

Randomize Statement

Description

Initializes the random-number generator.

Syntax

Randomize [*number*]

The *number* argument can be any valid numeric expression.

Remarks

Randomize uses *number* to initialize the **Rnd** function's random-number generator, giving it a new seed value. If you omit *number*, the value returned by the system timer is used as the new seed value.

If **Randomize** is not used, the **Rnd** function (with no arguments) uses the same number as a seed the first time it is called, and thereafter uses the last generated number as a seed value.

Note To repeat sequences of random numbers, call Rnd with a negative argument immediately before using Randomize with a numeric argument. Using Randomize with the same value for *number* does not repeat the previous sequence.

The following example illustrates use of the **Randomize** statement:

```
Dim MyValue, Response
Randomize    ' Initialize random-
number generator.

Do Until Response = vbNo
   MyValue = Int((6 * Rnd) + 1)    '
Generate random value between 1 and
6.
   MsgBox MyValue
   Response = MsgBox ("Roll again? ",
vbYesNo)
Loop
```

Rnd Function

Description

Returns a random number.

Syntax

Rnd[(*number*)]

The *number* argument can be any valid numeric expression.

Remarks

The **Rnd** function returns a value less than 1 but greater than or equal to 0. The value of *number* determines how **Rnd** generates a random number:

If *number* is	Rnd generates
Less than zero	The same number every time, using *number* as the seed.
Greater than zero	The next random number in the sequence.
Equal to zero	The most recently generated number.
Not supplied	The next random number in the sequence.

For any given initial seed, the same number sequence is generated because each successive call to the **Rnd** function uses the previous number as a seed for the next number in the sequence.

Before calling **Rnd**, use the **Randomize** statement without an argument to initialize the random-number generator with a seed based on the system timer.

To produce random integers in a given range, use this formula:

```
Int((upperbound - lowerbound + 1) *
Rnd + lowerbound)
```

Here, *upperbound* is the highest number in the range, and *lowerbound* is the lowest number in the range.

Note To repeat sequences of random numbers, call **Rnd** with a negative argument immediately before using **Randomize** with a numeric argument. Using **Randomize** with the same value for *number* does not repeat the previous sequence.

ประวัติผู้เขียน

ชื่อ	**นางสาว วิภาศิริ สายวิรุณพร**
วันเดือนปีเกิด	12 สิงหาคม พ.ศ. 2516
สถานที่เกิด	กรุงเทพมหานคร
ประวัติการศึกษา	
ระดับมัธยมศึกษา	โรงเรียนสตรีศรีสุริโยทัย
ระดับอุดมศึกษา	คณะเทคโนโลยีการเกษตร ภาควิชา เทคโนโลยีการผลิตพืช สาขาวิชา พืชสวน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ระดับประกาศนียบัตรบัณฑิต	สาขาวิชาการพัฒนาระบบสารสนเทศ สำนักการศึกษาและพัฒนาระบบสารสนเทศ สถาบันบัณฑิตพัฒนบริหารศาสตร์
ประวัติการทำงาน	พนักงานตำแหน่ง โปรแกรมเมอร์ บริษัท วี-สมาร์ท จำกัด 7/19 รามคำแหง 11 หัวหมาก บางกะปิ กรุงเทพฯ