เทคนิคการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG

ธมกร บุ้งจันทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2556

A Stegano Technique for hiding text into JPEG file

Thamakorn Boongchan

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Computer and Telecommunication Engineering

Faculty of Engineering, Dhurakij Pundit University

2013

ใบรับรองวิทยานิพนธ์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ เทคนิคการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG

เสนอโดย ธมกร บุ้งจันทร์

สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม

วิชาเอก วิศวกรรมคอมพิวเตอร์สารสนเทศและซอฟต์แวร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(อาจารย์ ดร.ประศาสน์ จันทราทิพย์)

.....กรรมการและอาจารย์ที่ปรึกษาวิทยานิพนธ์

(อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์)

..... กรรมการ

(อาจารย์ ดร.กิตติวงศ์ถาวราวัฒน์)

..... กรรมการ

(อาจารย์ ดร.เนื่องวงศ์ ทวยเจริญ)

คณะวิศวกรรมศาสตร์รับรองแล้ว

..... คณบดีคณะวิศวกรรมศาสตร์

(อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์)

วันที่ 1 เดือน มีนาคม พ.ศ. 2556

กิตติกรรมประกาศ

ผู้จัดทำสารนิพนธ์ฉบับนี้ ขอกราบขอบพระคุณ อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์ อาจารย์ที่ปรึกษา ที่ได้กรุณาให้ความอนุเคราะห์สละเวลา และให้โอกาส พร้อมทั้งคำแนะนำต่างๆ อันเป็นประโยชน์ต่อการทำวิทยานิพนธ์ จนสามารถสำเร็จลุล่วงไปได้ด้วยดี ทางผู้จัดทำสารนิพนธ์จึงขอกราบขอบพระคุณมา ณ โอกาสนี้

สุดท้ายนี้ ขอขอบพระคุณ บิดา มารดา รวมถึงครอบครัวของข้าพเจ้าที่เข้าใจและให้ความช่วยเหลือความช่วยเหลือในด้านต่างๆ จนทำให้สารนิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ด้วยดี

ธมกร บุ้งจันทร์

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

หัวข้อวิทยานิพนธ์ เทคนิคการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG
ชื่อผู้เขียน ธมกร บุ้งจันทร์
อาจารย์ที่ปรึกษา อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์
สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา 2555

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอเทคนิคการอำพรางข่าวสาร ที่เรียกว่า สเตกาโนกราฟี ไว้ในไฟล์รูปภาพแบบบีบอัดได้ชนิด JPEG ซึ่งนิยมใช้กันมากในเครื่องคอมพิวเตอร์ในปัจจุบัน โดยใช้วิธีการนำค่าของข้อมูลไปอำพรางไว้ที่ตำแหน่งต่าง ๆ ของรูปภาพ ซึ่งตำแหน่งของรูปภาพนี้จะทำการแบ่งเป็นคลัสเตอร์ย่อย ๆ ทำให้ได้ข้อมูลที่นำไปอำพรางมีตำแหน่งการอำพรางกระจายตัวไปทั่วไฟล์ ทำให้ยากต่อการตรวจจับมากยิ่งขึ้น นอกจากนี้ยังมีการเข้ารหัสข่าวสารอีกชั้นหนึ่งก่อนการอำพรางข้อมูลไว้ในรูปภาพ จึงทำให้การเจาะรหัสเพื่อนำข่าวสารนี้ออกมายากมากขึ้น

จากผลการทดสอบการอำพรางข้อมูลพบว่า ไฟล์รูปภาพ JPEG ขนาด 60 กิโลไบต์ขึ้นไป เมื่อข้อมูลที่ต้องการอำพรางมีขนาดเฉลี่ยประมาณ 1000 ตัวอักษร จะทำให้การอำพรางข่าวสารได้โดยไม่มีการผิดพลาด แต่ถ้าไฟล์รูปภาพ JPEG ที่นำมาใช้ในการอำพรางมีขนาดเล็กกว่า 60 กิโลไบต์ จะทำให้เมื่อคืนค่าข่าวสารจากการอำพรางจะมีข่าวสารบางตัวอักษรที่ผิดพลาด

Thesis Title	A Stegano Technique for hiding text into JPEG file
Author	Thamakorn Boongchan
Adviser	Dr. Chaiyaporn Khemaphatapan
Department	Computer Engineering and Telecommunication
Academic Year	2012

ABSTRACT

This thesis proposed a technique for hiding information called steganography into a JPEG file. Today, a JPEG file is a compressed picture file that is widely used on computer. The technique is based on hiding information various locations depending on information's value. However, the file location is divided into many clusters. Thus, information will be spreadly located throughout the JPEG file. Moreover, because raw information will be encrypted before hiding process, cracking information from the JPEG file is very difficult.

The testing results found that a JPEG file having about 60 kbytes or more can be used for hiding information up to 1,000 bytes without an error. However, if a JPEG file less than 60 kbytes is used, some error characters may be possible.

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในยุคปัจจุบันนี้ การส่งรูปภาพผ่านทางอินเตอร์เน็ตหรืออินทราเน็ต มีอัตราการเจริญเติบโตและได้รับความนิยมในการใช้งานมากยิ่งขึ้น ซึ่งจะได้เห็นได้จากการส่งภาพผ่านทางอีเมล์ หรือหน้าเว็บไซต์ต่าง ๆ รวมถึงซอฟต์แวร์ที่จัดอยู่ในหมวดหมู่ Instant Messenger เช่น MSN Messenger และ Yahoo Messenger ที่ให้บริการ การส่งภาพไปยังผู้รับ ซึ่งบางครั้งการส่งภาพไปยังปลายทาง ผู้ส่งมีความต้องการที่จะทำการอำพรางข้อมูลที่เป็นข้อความไปยังผู้รับ โดยที่ข้อความบางอย่าง ได้ถูกจัดเป็นให้ข้อความที่เป็นประเภทความลับ อันเป็นข้อความที่ไม่สามารถเปิดเผยให้บุคคลที่สามได้รับรู้ วิธีการในการทำให้ข้อความเป็นความลับที่สามารถอำพรางไว้ในภาพจึงได้เกิดขึ้น โดยวิธีการนั้น สามารถทำให้ข้อความที่ถูกส่งออกไปมีความปลอดภัยและเชื่อมั่นได้ถึง การที่ข้อความไม่สามารถถูกเปิดอ่านได้จากบุคคลที่สาม รายงานการวิจัยที่ผ่านมา ได้ทำการวิเคราะห์เพื่อปรับปรุงวิธีการในการอำพรางข้อมูล โดยการอำพรางข้อความไว้ในที่ LSB ในภาพประเภท BMP[1, 2, 3] เทคนิคการอำพรางไว้ที่ DCT ของภาพ BMP[4] และ เทคนิคการอำพรางข้อความไว้ที่ DCT

---

[1] Nabarun Bagchi, "Secure BMP Image Steganography Using Dual Security Model (I.D.E.A, image intensity and Bit Randomization and Max – Bit Algorithm", International Journal of Computer Applications, 2010 Vol.1 ,No.21, pp. 18 – 22.

[2] Ali Akbar Nikoukar, "An Image Steganography Method with High Hiding Capacity Based on RGB Image", International Journal of Signal and Image Processing, 2010 Vol.1, pp. 238-241.

[3] Jessica Fridrich, Miroslav Gojjan and Rui Du, "Reliable Detection of LSB Steganography in Color and Grayscale Images", MM&Sec '01 Proceedings of the 2001 workshop on Multimedia and Security, 2001, pp. 27-30.

[4] Lala Krikor, Sami Baba, Thawar Arif and Zyad Shaaban, "Image Excryption Using DCT and Stream Cipher", European Journal of Scientific Researchm, 2009 Vol. 32 ,No.1, pp. 47 – 57.

ของภาพ JPEG[5, 6] ซึ่งเทคนิคทั้งสามนี้ยังไม่เหมาะสมกับการนำไปใช้ในการอำพรางข้อความในภาพ เนื่องจาก ภาพ BMP มีขนาดใหญ่และเป็นที่คาดเดาได้ถึง การอำพรางข้อความในภาพ และสำหรับเทคนิคการอำพรางข้อความที่ DCT ของภาพ JPEG พบข้อจำกัดคือ ไม่รองรับปริมาณข้อความที่มีจำนวนมาก และมีขั้นตอนยุ่งยากซับซ้อนจึงไม่เหมาะสมกับการนำไปใช้ในการในหน่วยงานภาคธุรกิจ และส่วนบุคคล

การอำพรางข้อความ (Steganography) [7] เป็นคำที่มาจากภาษากรีก มีความหมายว่า การเขียน (Concealed Writing) ถูกใช้ครั้งแรกเมื่อปี ค.ศ. 1499 โดย Johannes Trithemius หมายถึง การเขียนข้อความ อันเป็นการอาศัยน้ำหมึกที่ไม่สามารถมองเห็นได้จากสายตาปรกติ เนื่องจาก Steganography เป็นการสื่อสารข้อความโดยการอำพรางข้อความ ซึ่งสามารถแบ่งประเภทในการสื่อสารออกเป็นสองรูปแบบ คือ การสื่อสารแบบเปิดเผย และการสื่อสารแบบปกปิด โดยการสื่อสารแบบเปิดเผยนั้น เป็นการสื่อสารที่มีผู้รับและผู้ส่งเพื่อทำการติดต่อแลกเปลี่ยนข้อความโดยไม่สนใจว่าจะมีบุคคลที่สามเข้ามาเพื่อที่จะรับทราบข้อความที่สื่อสารกันอยู่หรือไม่ ส่วนการสื่อสารแบบปกปิดนั้น เป็นการสื่อสารระหว่างผู้ส่งและผู้รับ โดยใช้วิธีการในการอำพรางข้อความเพื่อป้องกันไม่ให้บุคคลที่สามได้รับรู้และไม่สามารถเข้าถึงข้อความนั้น ๆ ได้ ซึ่งวิธีการสื่อสารแบบปกปิดนี้ แบ่งออกเป็นสองประเภท คือ การสื่อสารแบบปกปิดประเภทไม่ซ่อนเร้น และการสื่อสารแบบปกปิดประเภทซ่อนเร้น เช่น การนำข้อความเข้าสู่กระบวนการ Cryptography[8] การสื่อสารแบบซ่อนเร้นนี้ จะถูกเข้ารหัสโดยมีข้อตกลงของทั้งสองฝ่ายในการถอดรหัสเพื่อให้สามารถมองเห็นข้อความได้อย่างแท้จริง โดยที่บุคคลที่สาม ที่ไม่มีความเกี่ยวข้องและไม่ได้รับอนุญาติถึงจะมีการรับรู้ว่ามีการส่งข้อความเกิดขึ้นจริง ก็จะไม่สามารถถอดรหัสข้อความนั้นได้ และด้วยหลักการนี้การ

---

[5] Chiang-Lung Liu and Shiang-Rong Liao, "High-peroformance JPEG steganography using complementary embedding strategy", The journal of the pattern recognition society, 2008 Vol. 41, pp. 2945 – 2955.

[6] Fitri Arnia, Ikue Iizuka, Masaaki Fujiyoshi and Hitoshi Kiya, "DCT Sign – Based Similarity Measure for JPEG Image Retrieval", IEICE Transaction. Fundmentals, 2007 Vol. E90-A, No.9, pp. 1976 – 1985.

[7] Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett, "Steganography and Digital Watermarking", School of Computer Science, The University of Birmingham, 2004.

[8] Ian Curry, "An Introduction to Cryptography and Digital Signatures", Entrust Securing Digital Identities & Information, 2001.

สื่อสารแบบปกปิดประเภทไม่ซ่อนเร้น จะขึ้นอยู่กับวิธีการและกุญแจ (Key/Password) ที่ใช้ในการกระบวนการ Cryptography ดังนั้นความเป็นไปได้ที่บุคคลที่สามจะสามารถถอดรหัสเพื่ออ่านข้อความสามารถทำได้ ด้วยเหตุนี้จึงเกิดวิธีการที่จะใช้การเข้ารหัสข้อความ ซึ่งเป็นการเข้ารหัสแบบปกปิดอีกประเภทหนึ่งคือ การทำ Steganography

ความหมายของคำว่า การอำพรางข้อความ (Steganography) เป็นวิธีการในการอำพรางข้อความในรูปแบบต่าง ๆ โดยมีจุดมุ่งหมายในการปกปิดหรือการทำให้เป็นความลับ เพื่อทำให้ดูเหมือนว่าไม่มีการอำพรางข้อความใด ๆ โดยที่ไม่สามารถสังเกตุได้จากสายตาปรกติ ซึ่งถ้าหากเปรียบเทียบกับวิธีการ Cryptography และวิธีการของ Steganography ทั้งสองวิธีการนี้จะมีความแตกต่างกัน โดยที่ Cryptography มีจุดประสงค์ เพื่อที่จะมุ่งเน้นในการทำให้ข้อความไม่สามารถอ่านและทำความเข้าใจได้ แต่ Steganography เป็นวิธีการที่ทำการมุ่งเน้น เพื่อให้บุคคลที่สามไม่สามารถรับรู้ได้ถึงการมีข้อความลับอำพรางไว้อยู่ โดยอาศัยสื่อสัญญาณข้อมูลของภาพหรือสื่อสัญญาณต่าง ๆ และออกแบบวิธีการอำพรางข้อความ ส่งออกไปพร้อมกับสื่อสัญญาณนั้นได้อย่างแนบเนียน

ดังนั้น จึงได้นำเสนอวิธีการอำพรางข้อความไว้ในไฟล์ภาพชนิด JPEG ที่สามารถรองรับข้อมูลประเภทอักขระภาษาไทยและภาษาอังกฤษ การอำพรางนี้จะไม่ทำให้ภาพดังกล่าวผิดเพี้ยนไปจากเดิม เมื่อสังเกตุด้วยตามนุษย์ โดยใช้วิธีการค้นหาตำแหน่งของข้อมูลภาพที่มีความสัมพันธ์กับข้อมูลที่ต้องการอำพราง ซึ่งข้อมูลดังกล่าวจะถูกเข้ารหัสไว้อีกชั้น โดยวิธีการดังกล่าวนี้ สามารถอำพรางข้อความขนาด 1000 ไบต์ลงในไฟล์ภาพขนาด 60 กิโลไบต์โดยไม่ผิดพลาด

1.2 วัตถุประสงค์ของการวิจัย

1. พัฒนาวิธีการอำพรางข้อความ เพื่อทำการอำพรางข้อความลงไปในรูปภาพบีบอัด ประเภท JPEG โดยให้รองรับกับภาษาไทย
2. พัฒนาวิธีการอำพรางข้อความ ให้รองรับกับภาษาไทยและภาษาอังกฤษ

1.3 ขอบเขตของการวิจัย

1. ออกและพัฒนาวิธีการอำพรางข้อความ ให้สามารถใช้งานได้กับข้อความภาษาไทยและภาษาอังกฤษ

2. สร้างวิธีการที่สามารถทำการอำพรางข้อมูลไว้ในไฟล์ภาพบีบอัด ประเภท JPEG และข้อความที่จะใช้ในการอำพราง ก่อนการทำการอำพรางข้อความ ด้วยภาษา VB.NET โปรแกรม Microsoft Visual Studio 2005 Express

3. ทดสอบประสิทธิภาพการเข้ารหัสและถอดรหัสข้อมูลก่อนที่จะทำการอำพรางข้อความและหลังการอำพรางข้อความ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

สามารถทำการอำพรางข้อความภาษาไทยและภาษาอังกฤษด้วยวิธีการอำพรางแบบใหม่เพื่อเพิ่มประสิทธิภาพในการอำพรางข้อความในภาพที่ถูกบีบอัดประเภท JPEG ทำให้สามารถรองรับการอำพรางข้อความในปริมาณที่เพิ่มขึ้นและลดความเสี่ยงที่จะถูกลักลอบอ่านข้อความที่ทำการอำพรางไว้ที่จะเกิดขึ้น ผลการศึกษายังสามารถใช้เป็นพื้นฐานในการพัฒนาระบบการอำพรางข้อความในรูปแบบที่ถูกบีบอัดประเภท JPEG ที่จะนำไปใช้ในงานเชิงพานิชยน์ สำหรับการให้บริการอำพรางข้อความต่อไป

บทที่ 2
แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

2.1 องค์ประกอบของ Steganography

2.1.1 Steganography

วิธีการของ Steganography เป็นวิธีการที่ใช้ในการส่งข้อความที่เป็นความลับ โดยใช้วิธีการซ่อนข้อความที่แฝงไปกับสื่อ ซึ่งอาจจะเป็นภาพ, วิดีโอและเสียง โดยมีหลักพื้นฐานในการออกแบบวิธีการอำพรางข้อความ อยู่ 3 ประการ[1]

1. ความแนบเนียน

การที่ระบบสามารถทำการอำพรางข้อความโดยไม่ทำให้ข้อมูลเดิมเกิดความเสียหายในระดับที่สายตาปรกติหรือเครื่องจักร อุปกรณ์ ไม่สามารถตรวจจับหรือรับรู้ได้ ในกรณีถ้าหากมีการอำพรางข้อความไว้กับรูปภาพ ภาพที่มีการอำพรางข้อความจะต้องมีลักษณะเช่นเดียวกับภาพเดิมก่อนที่จะมีการอำพรางข้อความ หรือไม่มีการสูญเสียคุณภาพทางสายตา

2. ปริมาณ

ปริมาณของข้อความที่สามารถทำการอำพรางไว้ในข้อมูลหลักได้

3. ความทนทาน

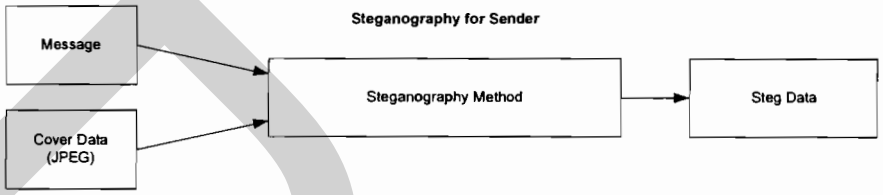
ในขั้นตอนการอำพรางข้อความลงในข้อมูลหลักนั้น ข้อมูลที่จะทำการอำพรางข้อความจะถูกแปรเปลี่ยนไปให้อยู่ในรูปของสัญญาณหรือข้อมูลต่ำ ซึ่งอาจเป็นไปได้ที่สัญญาณข้อมูลที่ทำการอำพรางข้อความจะถูกทำลายหรือกำจัดไปได้โดยง่าย ดังนั้นในการออกแบบระบบอำพรางข้อมูลจำเป็นที่จะต้องคำนึงถึงวิธีการที่จะทำการอำพรางข้อความลับให้ติดแน่นอยู่ในข้อมูลหลักได้

---

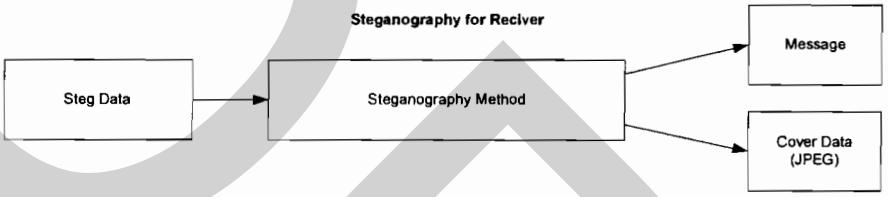
[1] C. Yun – Qing Shi, Sui Song, Zheng Zhang, Zhicheng Ni and Dekun Zou, "Detection of block DCT-based, steganography in gray-scale images ", Manikopoulos, pp. 335 – 358.

2.1.2 Steganography Structure

โครงสร้างพื้นฐานของระบบการอำพรางข้อความ ประกอบไปด้วย ผู้ส่ง วิธีการอำพรางข้อความ และผู้รับ ซึ่งในส่วนของผู้ส่งจะเชื่อมโยงกับวิธีการในการอำพรางข้อความ และส่วนของผู้รับจะเชื่อมโยงกับวิธีการในการอ่านข้อความที่มีการอำพราง ดังแสดงในภาพที่ 2.1



ก) ระบบการอำพรางข้อความสำหรับผู้ส่ง



ข) ระบบการอ่านข้อความที่มีการอำพรางสำหรับผู้รับ

ภาพที่ 2.1 วิธีการส่ง – รับ ข้อความที่มีการอำพรางสำหรับวิธีการ Steganography

จากภาพที่ 2.1 ขั้นตอนในส่ง-รับข้อมูล สำหรับวิธีการ Steganography เริ่มจาก ผู้ส่งเตรียมภาพ และข้อความที่ต้องการส่ง เข้าสู่กระบวนการ การอำพรางข้อความ ผลลัพธ์ที่ได้ จะเป็นภาพที่มีข้อความอำพรางไว้อยู่ในภาพ และเมื่อผู้รับได้ภาพนั้น ๆ ผู้รับจะนำภาพที่ได้ เข้าสู่วิธีการ Steganography เพื่อทำการถอดข้อความที่ทำการอำพรางไว้

2.2 ระบบการซ่อนข้อความไว้ในไฟล์ภาพ

S-Tools

เป็นการซ่อนข้อมูลอยู่ในในแฟ้มข้อมูล รูปภาพ GIF, BMP หรือแฟ้มข้อมูลเสียง WAV และสามารถทำการสร้างรหัสลับกับ IDEA, DES, Triple DES และ MDC ขึ้นมาได้ โดยมีการทำงานเป็นการแยกบิทของข้อมูล มีการทำเครื่องหมายโดยแบ่งแยกสี

Hide and Seek

เป็นการฝังข้อมูลเป็นไฟล์ GIF อยู่ในรูปภาพ โดยใช้พื้นที่เพียงเล็กน้อยของข้อมูลในแต่ละbyte ในการเข้ารหัสตัวอักษร ดังนั้นจึงใช้ในการกระจายข้อมูล (ทำให้รูปภาพเสื่อมคุณภาพ) ผ่านทางไฟล์ GIF

EZ-Stego

เป็นโปรแกรมที่ใช้ซ่อนข้อมูลที่อยู่ในรูปแบบไฟล์ภาพ GIF และเขียนเป็น ภาษา JAVA ได้ ซึ่ง EZ-Stego จะมีการเก็บข้อมูล ที่มีความสำคัญเพียงเล็กน้อย ในไฟล์รูปภาพ GIF โดยจะทำการเก็บข้อมูลลงใน ตารางสี (Color Tables)

F5

เป็นโปรแกรมที่ทำการซ่อนข้อความในรูปภาพที่ประกอบไปด้วย BMP, GIF และรูปภาพกราฟฟิคประเภท JPEG

Steghide

เป็นโปรแกรมที่สามารถซ่อนข้อมูลไว้ในภาพประเภท JPEG, BMP, WAV และ AU ซึ่งความถี่ของสีจะไม่มีมีการเปลี่ยนแปลงทำให้ทนต่อการซ่อนข้อมูล

JpegX

เป็นโปรแกรมที่สามารถซ่อนข้อความไว้ในภาพแบบ Jpeg โดยทำการเข้ารหัสแบบ Substition Key ภาพที่ได้ออกมานั้นยังคงเป็นภาพที่มีลักษณะเหมือนกับภาพต้นฉบับ

2.3 โครงสร้างมาตราฐานของไฟล์ภาพแบบ JPEG

JPEG มาจากคำว่า Joint Photographic Experts Group[2] ทำงานร่วมกับ ISO/IEC เป็นการทำงานเพื่อสร้างมาตราฐานสำหรับการเข้ารหัสภาพ และใช้รูปแบบการบีบอัดไฟล์ภาพดิจิตอลแบบสูญเสีย โดยให้เสียความละน้อยที่สุด และผ่านกระบวนการบีบอัดข้อมูล ทำให้ขนาดของไฟล์ภาพ JPEG หลังจากผ่านการบีบอัดข้อมูลแล้ว จะมีขนาดไฟล์ที่เล็กกว่าภาพต้นฉบับ และสีของภาพหลังจากการบีบอัด ไม่ว่าจะเป็นภาพสี หรือภาพขาว – ดำ สายตาปรกติจะไม่สามารถสังเกตเห็นถึงความเปลี่ยนแปลงของสีที่มีขนาดเล็ก พร้อมยังมีการสนับสนุนของจำนวนสี โดยสนับสนุนจำนวนสีถึง 24 bit ทำให้ภาพมีความคมชัดและมีความละเอียดของภาพสูง

2.3.1 Syntax and structure

โครงสร้างของรูปภาพประเภท JPEG ภายในรูปภาพจะประกอบไปด้วยแต่ละส่วน ซึ่งแต่ละส่วนนั้น จะประกอบไปด้วย Marker (เครื่องหมายหรือสัญลักษณ์ที่ใช้ภายในรูปภาพ) เพื่อระบุถึงความหมายของข้อมูลในภาพ ตัวอย่างเช่น การเริ่มต้นของข้อมูลภาพนั้น จะเริ่มด้วย 0xFF, 0xD8 มีค่าข้อมูลเป็น byte และ สิ้นสุดข้อมูลภาพ จะลงท้ายด้วย 0xFF, 0xD9 ดังแสดงในภาพที่ 2.2

---

[2] Christine Bako. (2004). JPEG 2000 Image Compression. *Analog Dialogue*, 38-09.

Short name	Bytes	Payload	Name	Comments
SOI	0xFF, 0xD8	none	Start Of Image	
SOF0	0xFF, 0xC0	variable size	Start Of Frame (Baseline DCT)	Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
SOF2	0xFF, 0xC2	variable size	Start Of Frame (Progressive DCT)	Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
DHT	0xFF, 0xC4	variable size	Define Huffman Table(s)	Specifies one or more Huffman tables
DQT	0xFF, 0xDB	variable size	Define Quantization Table(s)	Specifies one or more quantization tables.
DRI	0xFF, 0xDD	2 bytes	Define Restart Interval	Specifies the interval between RSTn markers, in macroblocks. This marker is followed by two bytes indicating the fixed size so it can be treated like any other variable size segment.
SOS	0xFF, 0xDA	variable size	Start Of Scan	Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Progressive DCT JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data
RST*n*	0xFF, 0xD0 .. 0xD7	none	Restart	Inserted every *r* macroblocks, where *r* is the restart interval set by a DRI marker. Not used if there was no DRI marker. The low 3 bits of the marker code cycle in value from 0 to 7.
APP*n*	0xFF, 0xE*n*	variable size	Application-specific	For example, an Exif JPEG file uses an APP1 marker to store metadata, laid out in a structure based closely on TIFF
COM	0xFF, 0xFE	variable size	Comment	Contains a text comment
EOI	0xFF, 0xD9	none	End Of Image	

ภาพที่ 2.2 Marker และโครงสร้างภายในของภาพประเภท JPEG

2.3.2 รูปแบบและวิธีการบีบอัดข้อมูลภาพ

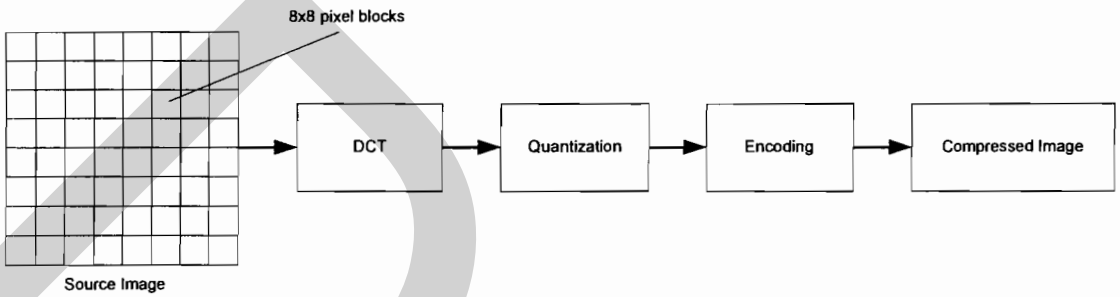
ในการบีบอัดข้อมูลภาพนั้น ได้แบ่งออกเป็น 2 แบบ คือ การบีบอัดข้อมูลแบบมีการสูญเสีย (Lossy data compression) ซึ่งเป็นการสูญเสียข้อมูลบางส่วนในการกระบวนการบีบอัดข้อมูล และเมื่อทำการขยายขนาดของข้อมูลให้มีขนาดเท่าเดิม จะยังคงเหลือข้อมูลบางส่วนที่ยังไม่สูญเสียไป

Original Image Low compression Medium compression High compression

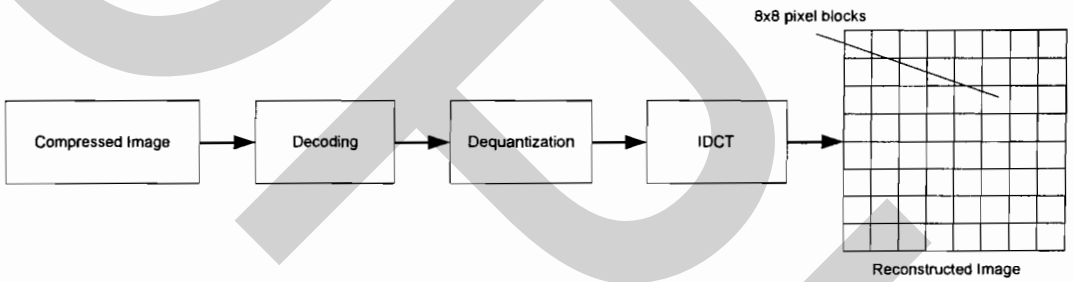
ภาพที่ 2.3 การบีบอัดข้อมูลแบบมีการสูญเสียที่ระดับต่างๆ เมื่อเทียบกับข้อมูลภาพต้นฉบับ

วิธีการบีบอัดแบบไม่มีการสูญเสีย (Lossless data compression) ซึ่งวิธีการนี้ หลังจากขยายข้อมูลให้มีขนาดเท่าเดิมก่อนการบีบอัด ข้อมูลนั้นจะไม่มีการสูญเสียแต่อย่างใด โดยการบีบอัดข้อมูลภาพนั้น อาจเป็นได้ทั้งแบบการสูญเสียและแบบไม่มีการสูญเสีย เช่น ในข้อมูลรูปภาพ จะใช้

วิธีการบีบอัดข้อมูลแบบมีการสูญเสียทำให้ข้อมูลบางส่วนที่ไม่มีความสำคัญ จะเกิดการสูญเสียไปในระหว่างการบีบอัดข้อมูล และเมื่อขยายขนาดข้อมูลออกมาแล้ว ข้อมูลที่มีความสำคัญยังคงอยู่ครบถ้วน ดังแสดงวิธีการบีบอัดข้อมูลและขยายข้อมูลภาพ ดังแสดงในภาพที่ 2.4



ก) ขั้นตอนวิธีการบีบอัดข้อมูลภาพประเภท JPEG



ข) ขั้นตอนวิธีการขยายข้อมูลภาพประเภท JPEG

ภาพที่ 2.4 ลักษณะการแปลงสัญญาณขั้นพื้นฐานของ DCT

2.3.3 Discrete Cosine Transform (DCT)

DCT[3] เป็นการบีบอัดข้อมูลแบบมีการสูญเสีย ในการบีบอัดไฟล์ภาพประเภท JPEG เนื่องมาจากคุณสมบัติของ DCT ที่เรียกว่า energy compaction คือ สามารถอัดพลังงานส่วนใหญ่ของสัญญาณ โดยเฉพาะภาพ ไปไว้ในสัมประสิทธิ์ย่านความถี่ต่ำในโดเมนของการแปลง และ การคำนวณการแปลงในทางปฏิบัติสามารถกระทำได้อย่างมีประสิทธิภาพ เนื่องจากการใช้ DCT เป็นที่นิยมในการบีบอัดข้อมูลสารสนเทศ เพราะว่า เมื่อมีการตัดสัมประสิทธิ์ของการแปลงที่มีค่าใกล้ศูนย์

[3]Andrew B. Watson. (1994). Image Compression Using the Discrete Cosine Transform. *Mathematica Journal*, 4, pp. 81-88.

ออกไปเป็นจำนวนเท่าๆ กัน ผลของการทำ IDCT จะให้ข้อมูลสารสนเทศมีความใกล้เคียงกับข้อมูลต้นแบบ (original sequence)

วิธีการ DCT เป็นการแปลงค่าความสว่างของภาพให้อยู่ในรูปแบบเชิงความถี่ (Frequency Domain) ทำให้สามารถเลือกค่าสัมประสิทธิ์หรือแอมพลิจูดของค่าความถี่ต่าง ๆ ได้โดยอาศัยตัวแปรที่มีนัยสำคัญที่ต่างกัน ตามสมการ

$$D(i,j)=\frac{2}{\sqrt{N}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}p(x,y)\cos\left[\frac{(2x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)i\pi}{2N}\right] \quad (2\text{-}1)$$

$D(i,j)$ คือ การคำนวณค่าของ DCT ของรูปภาพ

i,j คือ ค่าตำแหน่งของ Pixel ข้อมูลภาพ

$p(x,y)$ คือ ค่าข้อมูล ณ ตำแหน่ง x,y ที่อ้างถึงใน *matrix* p

N คือ ขนาด Matrix สำหรับใช้ในการบีบอัดเท่ากับ 64

เมื่อกำหนดให้ $C(i)$ และ $C(j)$ เป็นค่าสัมประสิทธิ์ ซึ่งสามารถหาได้ตามสมการดังต่อไปนี้

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}} & for \quad u=0\\ 1 & for \quad u>0\end{cases} \quad (2\text{-}2)$$

ค่าของ DCT สำหรับภาพ ที่มี block ในการบีบอัดขนาด $8x8$ หาได้จากสมการดังนี้

$$D(i,j)=\frac{1}{4}C(i)C(j)\sum_{x=0}^{7}\sum_{y=0}^{7}p(x,y)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)j\pi}{16}\right] \quad (2\text{-}3)$$

และ เพื่อให้ได้ Matrix ขนาด $8x8$ สำหรับในการบีบอัดภาพ ซึ่งสามารถหาได้จากสมการดังต่อไปนี้

$$T_{i,j}=\begin{cases}\frac{1}{\sqrt{N}} & for \quad i=0\\ \sqrt{\frac{2}{N}}\cos\left[\frac{(2j+1)i\pi}{2N}\right] & for \quad i>0\end{cases} \quad (2\text{-}4)$$

ผลลัพธ์ที่ได้จากสมการที่ (2-4) ดังแสดงในภาพที่ 2.5

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3563 & -.3563 & -.3563 & .3563 & .3563 & -.3563 & -.3563 & .3563 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1919 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

ภาพที่ 2.5 ลักษณะข้อมูล T matrix ที่ใช้สำหรับการบีบอัดข้อมูลภาพ

และสมการที่ (2-5) เป็นการดึงค่าข้อมูลในภาพเพื่อให้ได้ข้อมูลในรูปแบบของเมตริกซ์ ดังแสดงในภาพที่ 2.6

$$I = \left[DN_{(i,j)} \right]_{nxn} \tag{2-5}$$

I คือ ข้อมูลแบบเมตริกซ์

i, j คือ ตำแหน่งของข้อมูลในเมตริกซ์

DN คือ ข้อมูลของเมตริกซ์

$$I = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}_{8x8}$$

ภาพที่ 2.6 ข้อมูล I matrix ที่เป็นข้อมูลภาพ

เนื่องด้วย DCT ได้ถูกออกแบบมาเพื่อใช้งานกับค่าข้อมูลของ pixel ซึ่งมีค่าของข้อมูลอยู่ระหว่าง -128 ถึง 127 ดังนั้นแล้ว ค่าของข้อมูลทุกค่าจะต้องถูกลบด้วย 128 และ i, j เป็นจำนวนแถวและคอลัมน์ของเมตริกซ์ ซึ่งมีค่าตั้งแต่ 0, 1, 2, 3, ... 7 ตามลำดับ

$$I'_{(i,j)} = I_{(i,j)} - DC_{Bias} \qquad (2\text{-}6)$$

$I'_{(i,j)}$ คือ ข้อมูลของเมตริกซ์ที่ถอดค่าออกแล้ว

DC_{Bias} คือ สัมประสิทธิ์ของข้อมูลที่ต้องการถอดออกจากเมตริกซ์ ในที่กำหนดให้มี ค่าเป็น 128

ผลลัพธ์ที่ได้จากสมการที่ (2-6) ดังแสดงในภาพที่ 2.7

$$I' = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}_{8x8}$$

ภาพที่ 2.7 ข้อมูล I'matrix ที่เป็นข้อมูลภาพ

หลังจากนั้นจึงประยุกต์ใช้สมการ DCT ดังนี้

$$I'' = TI'T^{t} \qquad (2\text{-}7)$$

I'' คือ ข้อมูลที่มีการทำทรานสโพส์แล้ว

TI' คือ ข้อมูลจากผลคูณของเมตริกซ์ กับข้อมูลเมตริกซ์ T

T^{t} คือ ข้อมูลทรานสโพสต์ของเมตริกซ์ T

ผลลัพธ์ที่ได้จากสมการที่ (2-7) ดังแสดงในภาพที่ 2.8

$$I''=\begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}_{8x8}$$

ภาพที่ 2.8 ข้อมูล I'' matrix ที่เป็นข้อมูลภาพที่มีการทำทรานสโพส์

2.3.4 Quantization

วิธีการ Quantization เป็นการปรับคุณภาพโดยการลดรายละเอียดของภาพลง เพื่อให้ปริมาณของภาพมีขนาดเล็กลงแต่ยังสูญเสียความละเอียดน้อยที่สุด ตามสมการดังนี้

$$[Q_q]=\begin{cases} \dfrac{50}{q}\ [Q_{50}]for \quad \_q<50 \\ \dfrac{100-q}{50}\ [Q_{50}]for \quad \_q>50 \end{cases} \qquad (2\text{-}8)$$

q คือ ปัจจัยของระดับคุณภาพมีค่าอยู่ระหว่าง 1 ถึง 100

Q คือ การลดรายละเอียดของภาพลง

ผลลัพธ์ที่ได้จากสมการที่ (2-8) ดังแสดงในภาพที่ 2.9

$$Q_{50}=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}_{8x8}$$

ก) ภาพที่ทำการลดรายละเอียดของภาพลง ในอัตราการลดรายละเอียดที่ 50

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 120 & 200 & 255 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}_{8x8}$$

ข) ภาพที่ทำการลดรายละเอียดของภาพลง ในอัตราการลดรายละเอียดที่ 10

$$Q_{90} = \begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix}_{8x8}$$

ค) ภาพที่ทำการลดรายละเอียดของภาพลง ในอัตราการลดรายละเอียดที่ 90

ภาพที่ 2.9 ข้อมูลภาพที่ทำการลดรายละเอียดลง

ข้อมูลภาพหลังจากทำการปรับลดรายละเอียดของภาพลงแล้ว ซึ่งเป็นข้อมูลที่ถูกแปลงอยู่ในรูปของ I'' *matrix* และจัดรูปแบบตัวเลขให้เป็นจำนวนเต็ม โดยใช้ข้อมูลที่ทำการปรับลดรายละเอียดที่ Q_{50} ตามสมการดังนี้

$$I''_{R(i,j)} = ROUND\left[\frac{I''_{(i,j)}}{Q(i,j)}\right] \qquad (2\text{-}9)$$

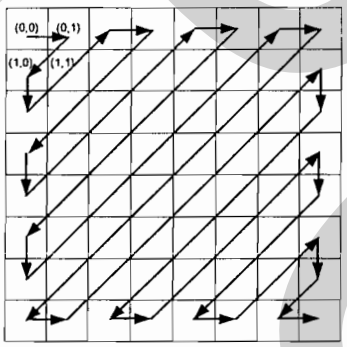
ผลลัพธ์ที่ได้จากสมการที่ (2-9) ดังแสดงในภาพที่ 2.10

$$I''_R = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{8x8}$$

ภาพที่ 2.10 ข้อมูลภาพที่ทำการลดรายละเอียดลง

2.3.5 Encoding

การปรับลดรายละเอียดของภาพนั้น เมื่อได้ข้อมูลสุดท้ายของการปรับลดมา จากสมการที่ (2-9) ซึ่งก่อนที่จะนำข้อมูลมาทำการ encode นั้น จะถูกแปลงโดยการเข้าหรัสให้เป็นลักษณะข้อมูลเลขฐานสอง ซึ่งจะใช้วิธีการ zig-zag ในการ encoding โดยแสดงลำดับในการ encode ดังแสดงในภาพที่ 2.11



ภาพที่ 2.11 การ encoding โดยใช้วิธี zig-zag

2.3.6 รูปแบบและวิธีการขยายข้อมูลภาพ

เมื่อต้องการแปลงข้อมูลย้อนกลับ ใช้สมการดังต่อไปนี้

$$I''_{R(i,j)} = I''_{Q(i,j)} \times Q_{(i,j)} \qquad (2\text{-}10)$$

ผลลัพธ์ที่ได้จากสมการที่ (2-10) ดังแสดงในภาพที่ 2.12

$$I''_R = \begin{bmatrix} 160 & 44 & 20 & 80 & 24 & 0 & 0 & 0 \\ 36 & 108 & 14 & 38 & 26 & 0 & 0 & 0 \\ -98 & -65 & 16 & -48 & -40 & 0 & 0 & 0 \\ -42 & -85 & 0 & -29 & 0 & 0 & 0 & 0 \\ -36 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{8x8}$$

ภาพที่ 2.12 ข้อมูลเมตริกซ์ที่มีการแปลงข้อมูลย้อนกลับ

ดำเนินการเปลี่ยนสมการกลับเป็นสมการเริ่มต้น จะได้ตามสมการดังนี้

$$I'_R = T^r I''_R T \tag{2-11}$$

จากสมการ (11) มาจากการคูณด้วย T^r ทางด้านซ้าย และคูณด้วย T ทางด้านขวา

$$I_{R(i,j)} = I'_{R(i,j)} + DC_{Bias} \tag{2-12}$$

จากสมการ (2-12) เป็นการเพิ่มค่าให้กลับไปเป็นค่าเดิม

2.4 ASCII Code

ASCII Code (รหัสแอสกี)[4] มีใช้ในระบบคอมพิวเตอร์ และเครื่องมือสื่อสารแบบดิจิทัลต่างๆ พัฒนาขึ้นโดยคณะกรรมการX3 ซึ่งอยู่ภายใต้การดูแลของสมาคมมาตรฐานอเมริกา (American Standards Association) ภายหลังกลายเป็น สถาบันมาตรฐานแห่งชาติอเมริกา (American National Standard Institute : ANSI) ในปี ค.ศ. 1969 โดยเริ่มต้นใช้ครั้งแรกในปี ค.ศ. 1967 ซึ่งมีอักขระทั้งหมด 128 ตัว (7 บิต) โดยจะมี 33 ตัวที่ไม่แสดงผล (unprintable/control character) ซึ่งใช้สำหรับควบคุมการทำงานของคอมพิวเตอร์บางประการ เช่น การขึ้นย่อหน้าใหม่สำหรับการพิมพ์ (CR & LF - carriage return and line feed) การสิ้นสุดการประมวลผลข้อมูลตัวอักษร (EOT - end of text)

รหัสแอสกีได้รับการปรับปรุงล่าสุดเมื่อ ค.ศ. 1986 ให้มีอักขระทั้งหมด 256 ตัว (8 บิต) สำหรับแสดงอักขระเพิ่มเติมในภาษาของแต่ละท้องถิ่นที่ใช้ เช่น ภาษาเยอรมัน ภาษารัสเซีย ฯลฯ

---

[4] Tarun Barayan Shankar and G.Sahoo. (2010). Cryptography by Karatsuba Multiplier with ASCII Codes. *International Journal of Computer Applications*, 10(12), pp. 53-60.

โดยจะมีผังอักขระที่แตกต่างกันไปในแต่ละภาษาซึ่งเรียกว่า โคดเพจ (codepage) โดยอักขระ 128 ตัวแรกส่วนใหญ่จะยังคงเหมือนกันแทบทุกโคดเพจ มีส่วนน้อยที่เปลี่ยนแค่บางอักขระ

2.5 การซ่อนข้อความไว้ในไฟล์ภาพ

วิธีการซ่อนข้อความไว้ที่รูปภาพนั้น เป็นการนำข้อความแต่ละตัวอักษรมาแปลงเป็นข้อมูลให้อยู่ในรูปแบบ ASCII Code หรือ Byte (เป็นค่าของตัวเลขที่อยู่ระหว่าง 0-255) ซึ่งเป็นรูปแบบที่สามารถนำไปใช้งานในวิธีการอำพรางข้อความเพื่อที่จะทำการอำพรางข้อความนั้น ๆ ไว้ในรูปภาพ โดยหลักการของการอำพรางจะถูกแบ่งออกเป็น 3 แบบ คือ การอำพรางข้อความไว้ที่ LSB (Least Significant Bit) หรือ MSB (Most Significant Bit) ของภาพ, การอำพรางข้อความไว้ที่ค่าอื่น ๆ ที่ไม่ใช่ค่าพิกเซลของภาพ และการอำพรางไว้ในส่วนของ DCT ของภาพ

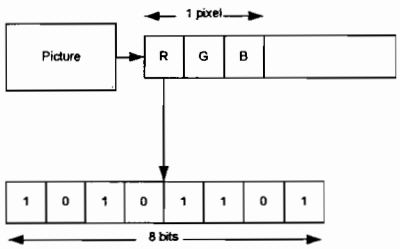
2.5.1 การอำพรางข้อความไว้ที่ค่าพิกเซลของภาพ

ในการอำพรางข้อความที่ค่าพิกเซลของภาพ มีวิธีการอำพรางทั้งหมด 2 แบบ คือ LSB (Least Significant Bit)[5] และ MSB (Most Significant Bit)[6] เป็นวิธีการที่เหมาะสมกับภาพประเภท BMP และ PNG ซึ่งวิธีการในอำพราง ข้อความ โดย LSB จะทำการอำพรางข้อความ ไว้ในบิตที่มีนัยสำคัญน้อยที่สุดของข้อมูลภาพ เนื่องจากผลของการเปลี่ยนแปลงนั้นจะส่งผมให้เฉดสีของภาพในแต่ละ pixel เปลี่ยนไปเล็กน้อยมาก โดยที่สายตาปรกติไม่สามารถสังเกตุเห็นได้ ส่วน MSB จะทำการอำพรางข้อความไว้ที่บิตที่มีนัยความสำคัญสูงที่สุด ทำให้สามารถสังเกตุเห็นได้จากสายตาปรกติอย่างชัดเจน โดยแสดงวิธีการอำพรางข้อความสำหรับ LSB และ MSB ดังแสดงในภาพที่ 2.13

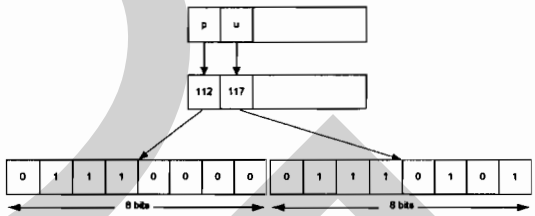
---

[5] Johnson, N. F., and S. Jajodia. (1998). Exploring Steganography : Seeing the Unseen. *IEEE Computer*, 31(12), pp. 26-34.

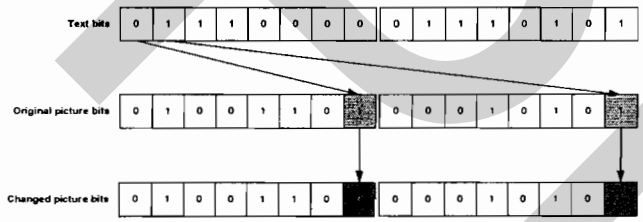
[6] Shrikant S. Khaire and Sanjay L. Nalbalwar. (2010). Review : Steganography – Bit Plane Complexity Segmentation (BPCS) Technique. *International Journal of Engineering Science and Technology*, 2(9), pp. 4860-4868.



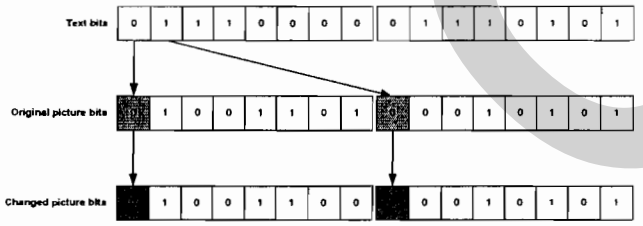
ก) ข้อมูลรูปภาพต่อ 1 pixel



ข) การแทนข้อมูลตัวอักษรด้วย ASCII 1 byte และแทน 1 byte เป็น 8 bits รูปแบบ Binary



ค) การอำพรางข้อความแบบ LSB โดยการแทนข้อมูลของตัวอักษรแต่ละ bit ในข้อมูลภาพ



ง) การอำพรางข้อความแบบ MSB โดยการแทนข้อมูลของตัวอักษรแต่ละ bit ในข้อมูลภาพ

ภาพที่ 2.13 การอำพรางข้อความโดยใช้วิธีการ LSB และ MSB

จากภาพที่ 2.13 ขั้นตอนสำหรับวิธีการ LSB และ MSB เพื่อที่จะทำการอำพรางข้อความไว้ในรูปภาพบนพิกเซลภาพ ซึ่งพิกเซลภาพนั้น จะถูกแยกออกมาเป็น 3 bytes ต่อ 1 pixel โดย 1 pixel จะประกอบไปด้วยค่าสี R(Red), G(Green) และ B(Blue) ซึ่งสามารถแทนค่าของแต่ละสีได้ออกมาเป็น 8 bits และตัวอักษรในข้อความแต่ละตัว จะเก็บเป็นรหัส ASCII ที่มีค่าเท่ากับ 1 byte โดยที่ 1 byte จะแบ่งออกเป็น 8 bits และการเก็บข้อมูลของตัวอักษร เป็นการเก็บแบบเลขฐานสอง โดยหลักการของ LSB และ MSB มีวิธีการดังต่อไปนี้

นำข้อความทั้งหมดที่จะทำการอำพรางข้อความ แปลงให้อยู่ในรูป ASCII byte

นำข้อมูลภาพทั้งหมดแปลงให้อยู่ในรูป ASCII byte

แปลงข้อมูลตัวอักษร จาก byte ให้เป็น bit

แปลงข้อมูลภาพจาก byte ให้เป็น bit

นำแต่ละ bit ของตัวอักษร ไปใส่ไว้แทนที่ bit สุดท้ายของ byte รูปภาพตั้งแต่ byte แรกจนกระทั่งครบตามจำนวน bit ของตัวอักษรทั้งหมด ซึ่งเป็นวิธีการของ LSB ส่วน MSB นั้น จะนำแต่ละบิทของตัวอักษร ไปใส่ไว้แทนที่ bit แรกของ byte รูปภาพ ตั้งแต่แรกจนกระทั่งครบตามจำนวน bit ของตัวอักษร

2.5.2 การอำพรางข้อความไว้ที่ค่าอื่น ๆ ในไฟล์ภาพ

สำหรับภาพที่ผ่านกระบวนการบีบอัดเช่นไฟล์ภาพประเภท JPEG นั้น ไม่สามารถใช้วิธีการซ่อนข้อมูลใน LSB หรือ MSB ของแต่ละ pixel สีได้ เนื่องจากการบีบอัดของภาพ JPEG เป็นรูปแบบการบีบอัดที่เรียกว่า Lossy Compression คือค่าของ pixels ภาพที่แสดงจะมีค่าไม่ตรงกับภาพต้นฉบับบ ซึ่งภาพที่ถูกขยายออกมาจะแตกต่างจากภาพต้นฉบับ ถ้าหากมีการใช้อัตราส่วนของการบีบอัดน้อย ภาพที่จะได้ใกล้เคียงกับภาพต้นฉบับบมากจนไม่สามารถมองเห็นความแตกต่าง แต่จะมีการใช้เนื้อที่ในการจัดเก็บข้อมูลมาก ในขณะที่มีการใช้อัตราการ Compression มาก จะได้ภาพที่ต่างจากต้นฉบับมากจนเห็นได้ชัดจากสายตาปรกติ แต่เนื่องด้วยข้อความ ที่ต้องทำการอำพรางไปกับภาพนั้น ต้องเป็นข้อมูลที่ไม่มีความคลาดเคลื่อน เพราะการอำพรางข้อความไม่สามารถยอมรับความเคลื่อนได้เมื่อได้ทำการอำพรางไปกับรูปภาพ ซึ่งวิธีการอำพรางข้อมูลใน LSB ของ pixels สีนั้นเป็นไปไม่ได้ แต่ทว่า ยังมีบางส่วนของไฟล์ภาพที่สามารถทำการอำพรางข้อความ[7] ลงไปได้เช่นกัน

---

[7] Johnson, N. F., and S. Jajodia. (1998). Exploring Steganography: Seeing the Unseen. *IEEE Computer*, 31(2), pp. 26-34.

เนื่องจากไฟล์รูปภาพประเภท JPEG เป็นภาพที่ผ่านการ Compression ไฟล์ภาพจะถูกแบ่งออกเป็นส่วนๆ นอกจาก Compress data ของรูปภาพจริงๆ เพื่อแสดงค่าชุดข้อมูลที่แตกต่างกัน เช่น Data Marker, Quantization Table, Frame Start, Huffman Table และส่วนอื่นๆ อันเป็นการสนับสนุนการทำ Compression ซึ่งข้อมูลในแต่ละส่วนจะมี Header และ Tag เริ่มต้นเพื่อแสดงให้เห็นถึงความชัดเจน ที่สำคัญนอกเหนือจาก Tag เหล่านี้จะไม่ถูกนำไปประมวลผลในการสร้างรูป (Image Decompression หรือ Image Reconstruction) อันเป็นทางเลือกสำหรับการอำพรางข้อความไว้ที่ Tag ที่ไม่มีการประมวลผล การทำ Steganography จึงเกิดขึ้นได้ โดยเป็นการใช้งานพื้นที่ส่วนอื่นของภาพที่ไม่มีการประมวลผล

2.5.3 การอำพรางข้อความโดยใช้รูปแบบ DCT

วิธีการ Steganography ในรูปแบบการอำพรางข้อความไว้ที่ DCT[8] เป็นการรวบรวม data ของ slice ที่เก็บเป็นรูปตัวเลขไว้ในรูปแบบ DCT ด้วยการแก้ไขค่าแล้วบันทึกข้อมูลลงในบางส่วนของ pixels ซึ่งจะเป็นค่าสีที่สายตาปรกติไม่สามารถมองเห็นได้ ในทุกๆ องค์ประกอบของสีจะใช้วิธีการ DCT สำหรับการเปลี่ยนค่าจาก pixels ให้เป็นตัวเลขนั้น โดยที่ภาพขนาด 8 × 8 pixel จะเท่ากับ 1 block จากสมการดังต่อไปนี้

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)x\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right] \qquad (2\text{-}13)$$

$F(u,v)$ คือ ขนาดของภาพ 1 Block เท่ากับ 8 × 8 pixels

และเมื่อแปลงรูปแล้วจะได้เป็น 64 DCT โดยเป็นไปตามค่าสัมประสิทธิ์ และค่าสัมประสิทธิ์ของ DCT เมื่อกำหนดให้ $C(u)$ และ $C(v)$ เป็นค่าสัมประสิทธิ์ ซึ่งสามารถหาได้ตามสมการดังต่อไปนี้

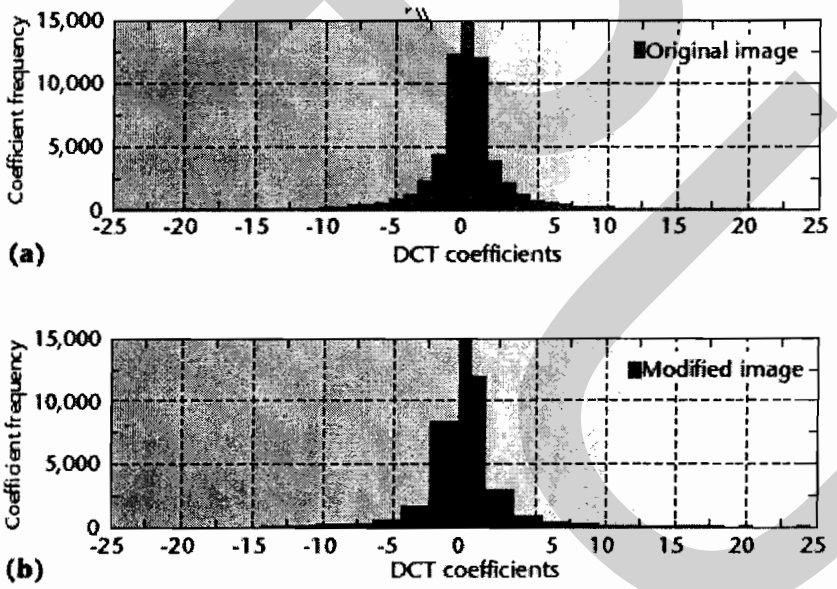
$$C(x)=\begin{cases}\frac{1}{\sqrt{2}} & for \quad x=0\\ 1 & for \quad x>0\end{cases} \qquad (2\text{-}14)$$

[8] Lala Krikor, Sami Baba, Thawar Arif and Zyad Shaaban. (2009). Image Excryption Using DCT and Stream Cipher. *European Journal of Scientific Researchm*, 32(1), pp. 47–57.

โดยที่ การทำงานจะบันทึกแบบ Sequential ลงใน least-significant bit (LSB) ที่อยู่ในรูป discrete cosine transforms (DCT-LSB) ด้วย LSB ที่มี message หรือ data

$$F^{Q}(u,v)=\left[\frac{F(u,v)}{Q(u,v)}\right] \tag{2-15}$$

โดยที่ $Q(u,v)$ คือ 64-DCT ซึ่งสามารถใช้ least-significant bits (LSB) ของ DCT เป็น redundant bits แล้วบันทึกไว้ใน DCT ที่เก็บข้อความหรือข้อมูล เป็นการใช้การบันทึกข้อมูลด้วย algorithm sequential ลงใน least-significant bit ของ DCT ซึ่ง algorithm นี้จะไม่เปิดเผยให้บุคคลอื่นที่รู้วิธี Steganographic หรือ โปรแกรม Steganographic อื่นๆ สามารถถอดข้อความ หรือ ข้อมูลได้ เพราะการจัดเรียง least-significant bits sequential จะสามารถป้องกันการตรวจสอบหรือการ detect ด้วยโปรแกรมประเภท Steganalysis ได้ โดยแสดงลักษณะของ DCT ของภาพก่อนและหลังการทำ Steganographic ดังแสดงในภาพที่ 2.14



ภาพที่ 2.14 ลักษณะของ DCT ของภาพก่อนและหลังการทำ Steganographic

อธิบายได้ว่า ในการเปลี่ยนแปลงแก้ไข bit least-significant bit ของสี ในภาพ image สีทุกสีจะมีการระบุตำแหน่งตัวเองด้วยดรรชนี i ในตารางสี เป็นการแทนค่าซึ่งเป็นความถี่ของสีก่อนและหลังการ แฝงข้อมูลใน bit เป็น ni และ ni^* กำหนดรูปแบบการกระจายข้อมูลเป็น bits

ถ้าหาก $n2i > n2i+1$ แล้ว กำหนดให้ pixels ที่มีสีเป็น color $2i$ เปลี่ยนความถี่สูงขึ้นเป็น color $2i + 1$ หาก pixels ที่มีสีเกิน color $2i + 1$ เปลี่ยนเป็น color $2i$ ตามสมการความสัมพันธ์ดังนี้

$$|n2i - n2i + 1| \geq |n2i^* - n2i^* + 1| \quad (2\text{-}16)$$

2.6 การอำพรางข้อมูลโดยใช้เทคนิค JPEGX

เทคนิคการอำพรางข้อมูลโดยใช้เทคนิค JPEGX เป็นเทคนิคการซ่อนข้อมูลไว้ที่ท้ายไฟล์ของภาพ JPEG โดยข้อมูลที่ทำการซ่อนลงไปนั้น ได้ถูกเข้ารหัสซึ่งใช้หลักในการเข้าหรัสเช่นเดียวกับ Substution Key ตามวิธีการดังต่อไปนี้

ขั้นตอนการอำพรางข้อมูลวิธีการ JPEGX

1. ทำการสร้าง Key ที่ใช้ในการเข้ารหัส โดยใช้ผลรวมของค่า ASCII ทุกตัวอักษรที่เป็นรหัสผ่าน
2. ตัวอักษรที่จะทำการอำพรางข้อความจะถูกเพิ่มไปตามค่าของ Key
3. ทำการเพิ่มค่าของ Key + 1
4. และใช้ Key ที่ทำการเพิ่มค่า เพิ่มเข้าไปที่ตัวอักษร ตัวที่กำลังจะทำการอำพรางข้อมูลถัดไป จนกว่าจะครบทุกตัวอักษร
5. ในกรณีที่ ไม่มี Key จะใช้ค่า ไบต์ของตัวอักษรตัวแรกที่จะทำการอำพรางข้อมูลให้เป็น Key และนำ Key + 1 เพื่อที่เพิ่มเข้าไปที่ตัวอักษรถัดไปจนกว่าจะครบทุกตัวอักษร

บทที่ 3
ระเบียบวิธีวิจัย

งานวิจัยนี้มีวัตถุประสงค์ในการออกแบบวิธีการ การอำพรางข้อความ ที่มีการใช้ภาษาไทย และภาษาอังกฤษ ในการส่งข้อความที่มีการอำพรางในไฟล์ภาพประเภท JPEG ทั้งนี้ผู้วิจัยได้แบ่งขั้นตอนการดำเนินงานออกเป็น 5 ขั้นตอนดังนี้

3.1 ศึกษาปัญหาและความต้องการของระบบ (Feasibility Study)

3.2 การวิเคราะห์และการออกแบบ (Analysis Design)

3.3 การพัฒนาโปรแกรม (Development)

3.4 การทดสอบโปรแกรม (Test)

3.5 การประเมินผลการวิจัย (Conclusion of the Research Evaluation)

3.1 ศึกษาปัญหาและความต้องการของระบบ (Feasibility Study)

การศึกษาปัญหาและความต้องการของการพัฒนาโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG นั้น เป็นการศึกษาถึงความเป็นไปได้ในการพัฒนาโปรแกรมในอนาคตว่าเป็นไปได้หรือไม่ ซึ่งจากปัญหาดังกล่าวเรื่องการอำพรางข้อมูลผ่านโปรแกรมนั้น มีความเป็นไปได้ในการพัฒนาโปรแกรมนี้ จากการศึกษาข้อมูลเบื้องต้น พบว่าในการพัฒนาโปรแกรมนี้สามารถที่จะให้โปรแกรมอำพรางข้อมูลประเภทข้อมูลอักขระที่เป็นภาษาไทย, ภาษาอังกฤษ และภาษาไทยปนกับภาษาอังกฤษได้

3.1.1 ศึกษาขั้นตอนการพัฒนาระบบ

ส่วนของโปรแกรมการอำพรางข้อมูลสำหรับเครื่องที่ใช้งาน ซึ่งส่วนนี้จะทำการอำพรางข้อมูลประเภทอักขระ มีรูปแบบการทำงานคือ เมื่อผู้ใช้งานมีความต้องการที่จะส่งข้อความไปยังปลายทาง โดยที่ผู้รับและผู้ส่งจะต้องมีโปรแกรมที่ผู้วิจัยได้พัฒนาขึ้นรวมถึงรหัสผ่านที่ทั้งสองฝ่ายตกลงที่จะใช้ร่วมกัน เมื่อผู้ส่งทำการเปิดรูปภาพ ที่ต้องการทำการอำพรางข้อมูล และมีการพิมพ์ข้อความที่จะทำการอำพรางรวมถึงรหัสผ่าน หลังจากนั้นผู้ใช้งานกดปุ่ม ประมวลผล ระบบจะทำการอำพรางข้อมูลโดยใช้รหัสผ่านที่ผู้ใช้งานมีการป้อนเข้าไป หลังจากนั้น ผู้ส่งทำการส่งรูปที่มีการอำพรางข้อความโดยผ่าน Web Browser หรือ โปรแกรม Chat Messenger เมื่อผู้รับได้รับรูปภาพ

จากผู้ส่ง ผู้รับจะทำการเปิดรูปภาพที่บันทึกไว้จากผู้ส่งเปิดขึ้นมา และใส่รหัสผ่านที่มีข้อตกลงร่วมกันว่าจะใช้งาน และทำการกดปุ่มประมวลผล โปรแกรมจะแสดงข้อความที่มีการอำพรางไว้ในภาพของผู้รับ

3.1.2 เครื่องมือในการพัฒนาโปรแกรมอำพรางข้อความในไฟล์ภาพชนิด JPEG มีดังต่อไปนี้

1. ด้านฮาร์ดแวร์

1.1 เครื่องคอมพิวเตอร์มีความเร็ว (CPU) ไม่น้อยกว่า 1 GHz

1.2 หน่วยความจำหลักมีความจุ (RAM) ไม่น้อยกว่า 128 MB

1.3 มีพื้นที่ว่าง (Free Hard Disk) ติดตั้งโปรแกรมไม่น้อยกว่า 100 MB

2. ด้านซอฟต์แวร์

2.1 โปรแกรม Microsoft Visual Studio Express 2005 สำหรับพัฒนาโปรแกรม

2.2 โปรแกรม Image Compressor สำหรับตรวจสอบค่า PSNR ในระบบ

3.1.3 เครื่องมือในการทดสอบโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG มีดังต่อไปนี้

1. ด้านฮาร์ดแวร์

1.1 เครื่องคอมพิวเตอร์มีความเร็ว (CPU) เทียบเท่ากับ Pentium 166 GHz หรือเป็นเครื่องที่มีประสิทธิภาพสูงกว่า เพื่อใช้ในการติดตั้งโปรแกรม อีกทั้งยังเป็นการไม่ให้กระทบกระเทือนการทำงานของโปรแกรมอื่น ๆ อีกด้วย หากใช้เครื่องที่มีคุณสมบัติต่ำกว่านี้

1.2 หน่วยความจำหลักมีความจุ (RAM) ไม่น้อยกว่า 128 MB ใช้สำหรับเป็นที่เก็บและประมวลผลของโปรแกรม เพื่อให้เกิดความรวดเร็วและถูกต้องในการทำงาน

1.3 มีพื้นที่ว่างสำหรับติดตั้งโปรแกรม (Free Hard Disk) ไม่น้อยกว่า 20 MB ใช้สำหรับการติดตั้งโปรแกรม เพื่อให้มีพื้นที่เพียงพอกับการทำงานของโปรแกรม

2. ด้านซอฟต์แวร์

2.1 โปรแกรม Microsoft Visual Studio 2005 สำหรับพัฒนาโปรแกรม ซึ่งเป็นโปรแกรมที่รอบรับ Functions ต่าง ๆ รวมถึง Component ที่ต้องใช้ในการพัฒนาโปรแกรมอย่างครบถ้วนและเหมาะสม

2.2 ใช้ระบบปฏิบัติการ Windows 2000 หรือ Windows XP ขึ้นไป

2.3 โปรแกรม Image Compressor สำหรับตรวจสอบค่า PSNR ในระบบ

3.2 การวิเคราะห์และการออกแบบ (Analysis and Design)

การวิเคราะห์และการออกแบบโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG สำหรับเครื่องผู้ใช้งาน เพื่อให้มีความเหมาะสมสอดคล้องกับหลักการและทฤษฎีการออกแบบมีดังต่อไปนี้

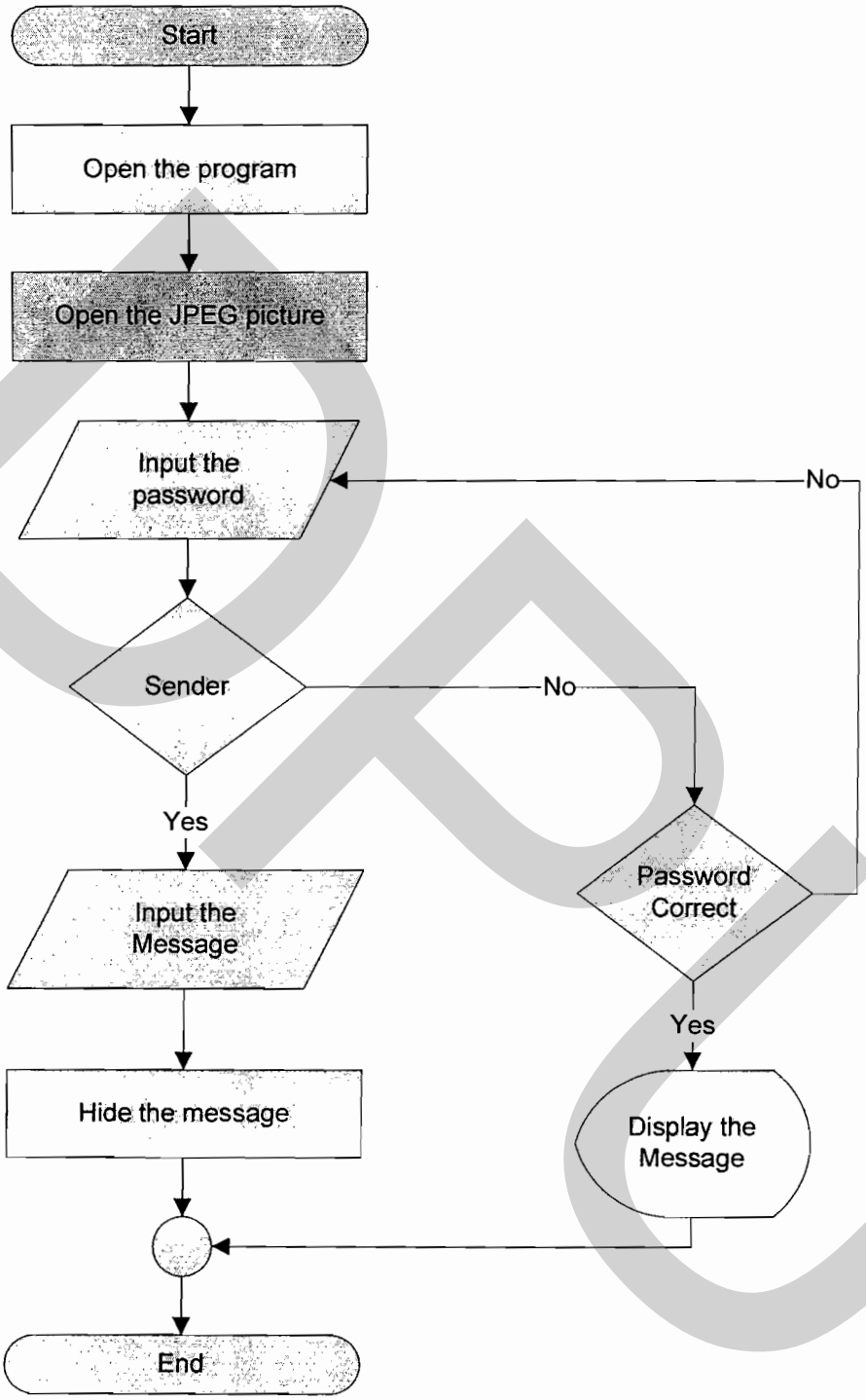
3.2.1 วิเคราะห์และออกแบบโปรแกรม โดยมีขั้นตอนการทำงานออกเป็น 2 ส่วงดังนี้

1. ศึกษาและเปรียบเทียบหาวิธีการอำพรางข้อความที่เหมาะสม การอำพรางข้อความที่มีใช้ในงานต่างประเทศเป็นพื้นฐานอ้างอิงในการวิจัย โดยวิธีการที่ถูกใช้งานอย่างแพร่หลายได้ใช้วิธีการ DCT เพื่อทำการอำพรางข้อความที่เป็นภาพประเภทแบบ JPEG และการใช้วิธี LSB ในภาพ BMP แต่เนื่องจากภาพประเภทที่มีการบีบอัดไม่สามารถทำการอำพรางข้อความได้เท่ากับปริมาณข้อความที่ต้องการส่งและภาพประเภท BMP เป็นภาพที่มีประเภทขนาดใหญ่จึงไม่เหมาะสมในการที่จะส่งผ่านภาพ BMP จึงได้ทำการปรับปรุงวิธีการให้มีความเหมาะสมกับภาพประเภทที่มีการบีบอัด และเปรียบเทียบประสิทธิภาพความถูกต้องและระยะเวลาในการประมวลผล จากการอำพรางข้อความด้วยวิธี ที่มีการปรับปรุง

2. วิเคราะห์ปัญหาการอำพรางข้อความในภาพประเภทบีบอัดแบบ JPEG ในขั้นตอนการวิจัยส่วนที่ 1 จะนำผลการทดสอบมาทำการวิเคราะห์ เพื่อนำไปแก้ไขและปรับปรุงให้มีความสอดคล้องกับข้อความที่จะทำการอำพรางที่จะทำการอำพรางไว้ในรูปภาพ และด้วยการดำเนินงานในส่วนที่ 2 ทำการทดสอบการอำพรางข้อความด้วยภาพและข้อความชุดเดิมอีกครั้ง เพื่อหาส่วนต่างประสิทธิภาพที่เพิ่มขึ้น แล้วสรุปงานวิจัยเพื่อนำไปพัฒนาวิธีการอำพรางข้อความให้สามารถใช้งานกับภาพประเภทบีบอัดแบบ JPEG

3.2.2 ผังงานระบบ (System Flowchart)

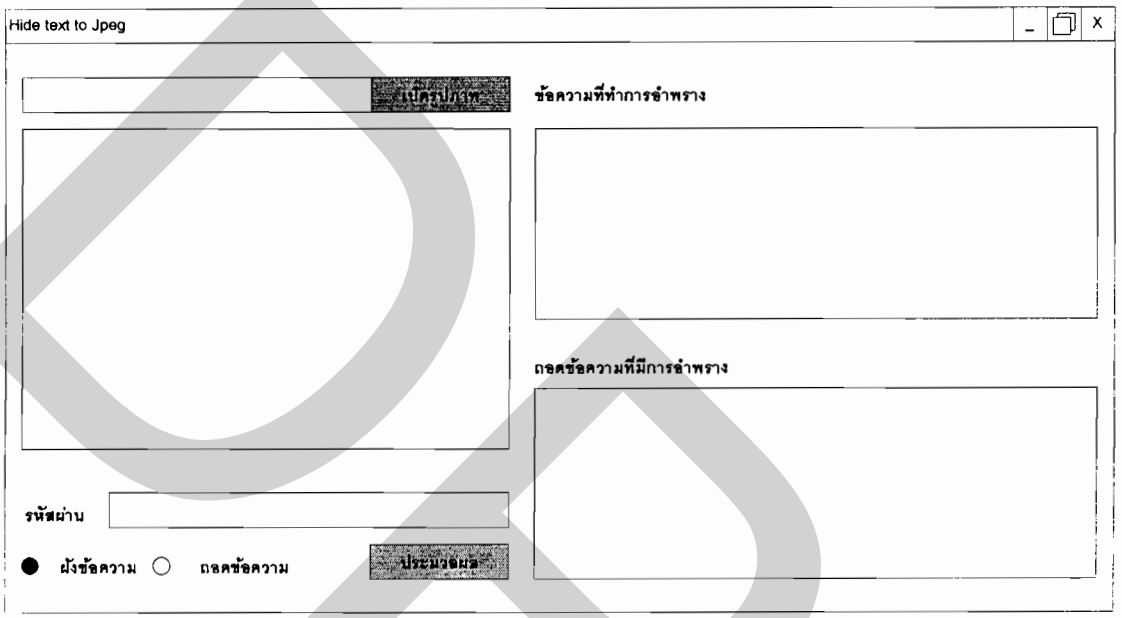
การทำงานในระบบการอำพรางข้อความนั้น ย่อมมีขั้นตอนและวิธีการใช้งานระบบเพื่อให้ผู้ใช้งานได้เข้าใจถึงการทำงานที่ง่ายมากยิ่งขึ้น โดยแสดงดังในภาพที่ 3.1



ภาพที่ 3.1 ผังงานระบบการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG

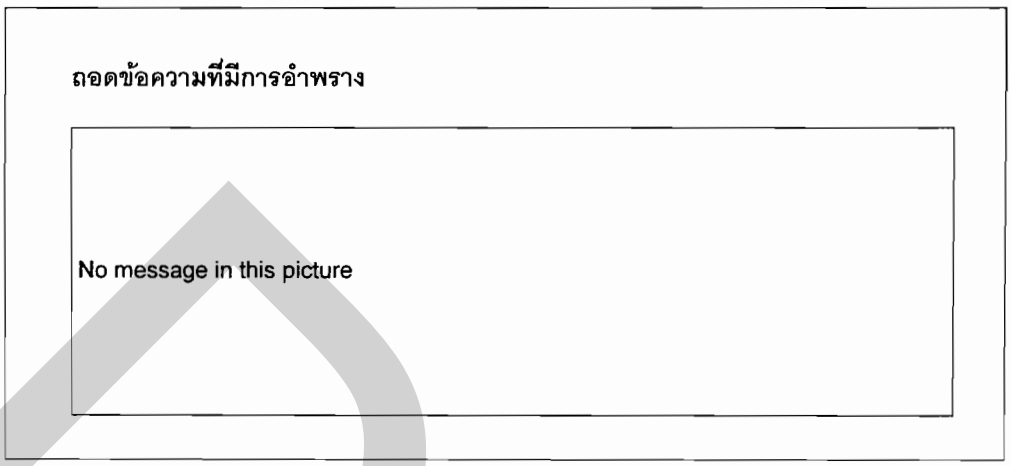
3.2.3 การออกแบบหน้าจอ (Design Interface)

1. หน้าจอโปรแกรมอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG เป็นหน้าจอสำหรับผู้ใช้งานระหว่างผู้รับและผู้ส่ง ดังแสดงในภาพที่ 3.2



ภาพที่ 3.2 หน้าจอโปรแกรมอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG

2. เมื่อผู้ส่งทำการเปิดรูปภาพและทำการกรอกรหัสผ่าน, และข้อความที่ทำการอำพราง พร้อมกับเลือกวิธีการ การฝังข้อความ ระบบจะทำการเข้ารหัสข้อมูลด้วยรหัสผ่านและทำการอำพรางข้อมูลไว้ในไฟล์ภาพ JPEG และทำการส่งไปยังผู้รับ หลังจากนั้นเมื่อผู้รับ เปิดรูปภาพจากผู้ส่งและทำการกรอกรหัสผ่าน พร้อมกับเลือกวิธีการถอดข้อความ ระบบจะตรวจสอบรหัสผ่านที่ได้รับถ้าหากไม่ตรงกันระบบจะไม่แสดงข้อความ “No Message in this picture” ในส่วนของ การถอดข้อความที่มีการอำพราง ดังแสดงในภาพที่ 3.3



ภาพที่ 3.3 หน้าจอแสดงข้อความแสดงข้อผิดพลาด

3.3 การพัฒนาโปรแกรม (Development)

การพัฒนาโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG สำหรับเครื่องคอมพิวเตอร์สำหรับผู้ส่งและผู้รับ โดยใช้โปรแกรม Microsoft Visual Studio โดยมีขั้นตอนการพัฒนาโปรแกรม ดังนี้

3.3.1 ศึกษาทฤษฎีที่เกี่ยวข้อง เพื่อใช้ประกอบการพัฒนาโปรแกรมให้เกิดประสิทธิภาพสูงสุด ได้แก่

1. หลักการทำงานของโครงสร้างข้อมูลภาพบีบอัดประเภท JPEG
2. หลักการทำงานอำพรางข้อความในภาพบีบอัดประเภท JPEG
3. หลักการทำงานในการเข้ารหัสและถอดรหัสข้อความ
4. หลักการทำงานการแปลงตัวอักษรให้เป็นค่า Byte เพื่อรองรับการทำงานกับข้อความภาษาไทย

3.3.2 พัฒนาโปรแกรม

1. ด้านโปรแกรมอำพรางข้อมูลลงในไฟล์ภาพประเภท JPEG พัฒนาด้วยภาษา Visual Basic.NET โดยเขียนโมดูลการทำงาน ดังนี้

1.1 โมดูลในการบีบอัดภาพประเภท JPEG

1.2 โมดูลในการสร้างบล็อกเริ่มต้นของข้อความที่จะทำการอำพรางข้อความด้วยวิธีการดังต่อไปนี้

1) MD5 (Message-Digest algorithm 5)[1]

1.3 โมดูลในการแปลงตัวษรให้เป็นค่า Byte และจากค่า Byte ให้เป็นตัวอักษรด้วยวิธีการดังต่อไปนี้

1) String.Asc (character)

2) Convert.ToChar (byte)

1.4 โมดูลในการแปลงค่าเชิงตัวเลขให้อยู่ในรูปข้อมูลเชิงเลขฐานสอง

1.5 โมดูลในการแปลงข้อมูลเชิงเลขฐานสองให้เป็นข้อมูลเชิงตัวเลข

1.6 โมดูลการเข้ารหัสและถอดรหัสข้อความ

1.7 โมดูลการอำพรางข้อความ

1.8 โมดูลการถอดข้อความที่มีการอำพราง

2 ด้านโปรแกรมการอำพรางข้อมูล

2.1 สำหรับเครื่องคอมพิวเตอร์ที่ใช้งานโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG ในโปรแกรมนี้จะมีส่วนที่ใช้ในการอำพรางและการถอดข้อมูลที่เป็นข้อความประเภทอักขระ โดยลักษณะการใช้งานจะแบ่งการใช้งานออกเป็น 2 ส่วนคือ ส่วนของผู้ส่งและผู้รับ โดยมีหลักการทำงานดังนี้

2.1.1 ส่วนการใช้งานทางฝั่งผู้ส่ง

1) เมื่อผู้ใช้งานเลือกฟังก์ชั่นการทำงานแบบฝังข้อความ ระบบจะนำข้อมูลที่ประกอบไปด้วย ข้อความและรหัสผ่านเข้าสู่กระบวนการการอำพรางข้อมูล

2) กระบวนอำพรางข้อมูลจะทำการเข้ารหัสข้อความที่ผู้ใช้งานต้องการอำพรางข้อมูลด้วยรหัสผ่าน และนำข้อมูลที่ผ่านการเข้ารหัส ทำการอำพรางข้อมูล

3) ระบบจะทำการบันทึกภาพหลังการผ่านกระบวนการอำพรางข้อมูลไว้ที่เครื่องคอมพิวเตอร์ของผู้ใช้งาน

4) ผู้ส่ง ทำการส่งภาพที่มีการอำพรางข้อมูลไว้ ให้กับผู้รับ โดยผ่าน Web Browser หรือ โปรแกรม Messenger Talk

2.1.2 ส่วนการใช้งานทางฝั่งผู้รับ

1) เมื่อผู้ใช้งานเลือกฟังก์ชั่นการทำงานแบบถอดข้อความ ระบบจะนำข้อมูลที่ประกอบไปด้วยรหัสผ่านเข้าสู่กระบวนการถอดข้อความที่มีการอำพราง

---

[1] R. Rivest. (1992). *The MD5 Message-Digest Algorithm*. RFC 1321, MIT LCS & RSA Data Security Inc.

2) กระบวนการถอดข้อความ จะทำการถอดรหัสโดยใช้รหัสผ่านเดียวกันกับทางฝั่งผู้รับ

3) ในกรณีที่รหัสผ่านเหมือนกับทางฝั่งผู้ส่ง ข้อความที่ถูกอำพรางไว้จะแสดงขึ้นมา หรือในกรณีที่รหัสผ่านไม่ตรงกับทางฝั่งผู้ส่ง ระบบจะแจ้งด้วยข้อความ "No message in this picture"

3.3.3 ทดสอบและปรับปรุงแก้ไขโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG

3.4 การทดสอบโปรแกรม (Test)

3.4.1 ทดสอบการทำงานของระบบขั้นต้น โดยหลังจากมีการพัฒนาระบบการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG ผ่านไประยะหนึ่ จะเริ่มทำการทดสอบหาข้อผิดพลาด เพื่อพัฒนาและแก้ไขให้ระบบทำงานได้อย่างสมบูรณ์

หลังจากมีการพัฒนาระบบการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG ไประยะหนึ่ง จะเริ่มทำการทดสอบย่อย เพื่อหาข้อผิดพลาดต่าง ๆ ภายใน Application แล้วทำการแก้ไข

3.4.2 ทำการทดสอบระบบด้วยการ จัดเตรียมชุดข้อมูลสำหรับการทดลอง โดยนำข้อความทั้งหมด 3 ชุด ประกอบไปด้วย

- ข้อความภาษาไทย
- ข้อความภาษาอังกฤษ
- ข้อความภาษาไทยปนกับภาษาอังกฤษ

และภาพบีบอัดประเภท JPEG ซึ่งประกอบไปด้วยภาพทั้งหมด 6 ชุด และมีขนาดดังต่อไปนี้

- 128 x 96 pixels
- 256 x 192 pixels
- 512 x 384 pixels
- 1024 x 768 pixels
- 2048 x 1536 pixels

โดยนำชุดข้อมูลของข้อความ ทำการอำพรางข้อความ กับชุดข้อมูลภาพทั้งหมด เพื่อให้สามารถวัดประสิทธิภาพการอำพรางข้อความ เมื่อการทดสอบเสร็จสิ้น ดำเนินการเก็บข้อมูลการทดสอบเพื่อใช้ในการศึกษาปัญหาการอำพรางข้อความต่อไป

3.4.3 ศึกษาปัญหาและการแก้ไขปัญหาการอำพรางข้อความ ความเป็นไปได้และปัญหาที่จะเกิดขึ้นในการอำพรางข้อความ ได้แก่

1) การตรวจสอบความผิดพลาดในการอำพรางข้อความ

ทั้งข้อความภาษาไทยและภาษาอังกฤษ ที่ถูกป้อนเข้ามาโดยการพิมพ์ข้อความที่จะทำการอำพราง เมื่อทำให้ข้อความที่ถอดออกมาไม่ตรงกับข้อความเดิม และออกแบบวิธีการแก้ปัญหาดังกล่าว

2) การตรวจสอบการเข้ารหัสและถอดหรัสข้อความ

วิธีการเข้ารหัสและถอดหรัส เป็นวิธีการที่ออกแบบมาเพื่อทำการเข้ารหัสและถอดรหัสข้อมูลตามหลัก Cryptography เพื่อให้ข้อความนั้น ได้เป็นความลับ ซึ่งไม่สามารถถอดรหัสออกมาและตรงกับข้อความเดิมได้ จะทำให้เข้าใจถึงความผิดพลาดที่เกิดขึ้น และสามารถนำไปปรับปรุงแก้ไขให้การเข้ารหัสและถอดรหัสในข้อความมีความถูกต้องมากขึ้น

3) การตรวจสอบค่าความเพี้ยนของภาพโดยใช้วิธีการ PSNR

วิธีการตรวจสอบแบบ PSNR[2] เป็นวิธีมาตราฐาน ที่ใช้ในการวัดประสิทธิภาพการแยกส่วนภาพสี ที่บ่งบอกถึงคุณภาพที่เปลี่ยนไประหว่างรูปภาพสองภาพ ที่ผ่านกระบวนการประมวลผลทางสัญญานใดๆ ค่า PSNR ที่สูงจะชี้ให้เห็นถึงคุณภาพของรูปที่ใกล้เคียงกับรูปภาพต้นฉบับ โดยที่ค่า PSNR นี้ได้ถูกนำมาใช้ในการประเมินคุณภาพของรูปที่ผ่านการฝังสัญญานลายน้ำมาแล้ว สามารถคำนวณหาได้จากสมการ (3-1) และ (3-2) โดยการกำหนดให้ $Opixel(i,j)$ คือจุดภาพที่ตำแหน่ง (x, y) ในรูปภาพดิจิตอลต้นฉบับที่มีขนาดเท่ากับ NxN และ $Wpixel(i,j)$ คือจุดภาพที่ถูกฝังสัญญานลายน้ำ

$$MSE = \frac{\sum[OPixel(i,j) - WPixel(i,j)]^2}{NxN} \quad (3\text{-}1)$$

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \quad (3\text{-}2)$$

MSE (Mean Square Error) คือ ค่าที่มาจากการคำนวณในทุก ๆ ตำแหน่งของจุดภาพภายในรูปภาพ

RMSE (Root Mean Square Error) คือ ค่ารากที่สองของ MSE

---

[2] ธำรงรัตน์ อมรรักษา และ วัชร พิชยนันท์. (2546). ภาพพิมพ์ลายน้ำดิจิตอล: วิธีป้องกันการละเมิดสิทธิทางปัญญาสำหรับรูปภาพ. *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, 13(2), 54-63.

ซึ่งถ้าค่า PSNR อยู่ระหว่าง 20 - 40 dB ถือว่าการแยกส่วนภาพอยู่ในระดับที่ดี แต่ถ้าค่า PSNR ที่ได้มีค่าต่ำหรือเข้าใกล้ศูนย์ แสดงว่าภาพที่นามาเปรียบเทียบกันมีความแตกต่างกันมาก นอกจากนี้ในการเปรียบเทียบถึงคุณภาพที่เพิ่มขึ้นของรูปภาพใด ๆ ที่สายตามนุษย์สามารถสังเกตุเห็นได้ รูปภาพนั้นจะมีการเปลี่ยนแปลงแปลงของค่า PSNR ที่มากกว่า 0.5 dB ขึ้นไป ยกตัวอย่าง เช่น รูปภาพที่ถูกฝังสัญญานลายน้ำที่มีความแรงสูงขึ้น 2 เท่า ส่งผลให้ค่า PSNR ตกลงมาน้อยกว่า 0.5 dB จะทำให้ไม่สามารถสังเกตเห็นถึงความเปลี่ยนแปลงด้วยสายตามนุษย์ ดังแสดงในภาพที่ 3.4

ก) ภาพต้นฉบับก่อนที่จะทำการฝังลายน้ำเพื่อตรวจสอบค่า PSNR

ข) ภาพหลังการทำ การฝังลายน้ำ เพื่อตรวจสอบค่า PSNR

ภาพที่ 3.4 ผลการตรวจสอบภาพโดยใช้วิธี PSNR หลังจากที่มีการฝังลายน้ำ

4) การปรับปรุงในส่วนอื่น ๆ

นอกจากความผิดพลาดของวิธีการอำพรางข้อความและวิธีการเข้ารหัสและถอดรหัสข้อความแล้ว อาจมีองค์ประกอบอื่น ๆ ที่สามารถปรับปรุงและส่งผลกระทบต่อการอำพรางข้อความให้มีประสิทธิภาพสูงขึ้นได้

3.5 การประเมินผลการวิจัย (Conclusions of the Research Evaluation)

การประเมินผลการทำงานของโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG สำหรับเครื่องคอมพิวเตอร์ผู้ใช้งาน งานวิจัยนี้ได้มีการประเมินประสิทธิภาพของการอำพรางข้อมูลที่พัฒนาขึ้นในมิติต่าง ๆ ดังต่อไปนี้

1) การประเมินสิทธิภาพการใช้งาน โดยวัดประสิทธิภาพและเปรียบเทียบผลการอำพรางข้อมูล ด้วยวิธีการของ DCT, JPEGX และวิธีการที่พัฒนาขึ้นมาใหม่

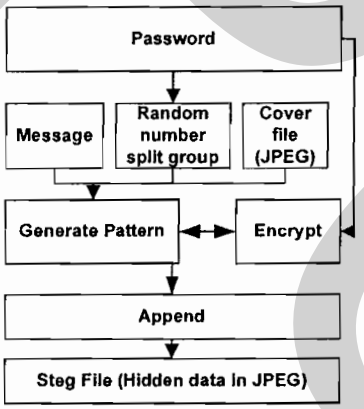
2) วิเคราะห์ผลการทดลอง

บทที่ 4
ผลการวิจัย

เมื่อทำการวิเคราะห์และออกแบบ ตลอดจนพัฒนาโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพชนิด JPEG จึงได้นำโปรแกรมดังกล่าวมาประเมินหาประสิทธิภาพในการทำงานของโปรแกรมอำพรางข้อมูล ผลการดำเนินงานและการประเมินโปรแกรมตรวจสอบ มีดังต่อไปนี้

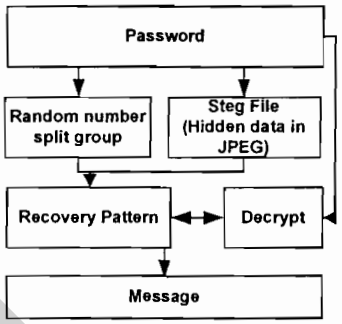
4.1 การอำพรางข้อความในโปรแกรมที่พัฒนา

การอำพรางข้อมูลนั้น เป็นการนำข้อมูลภาพ ซึ่งเป็นไฟล์ภาพบีบอัดชนิด JPEG เข้าสู่กระบวนการ โดยนำภาพบีบอัดที่ได้มา ทำการบีบอัดข้อมูลอีกครั้งโดยผ่านอัตรการบีบอัดที่เหมาะสม เพื่อทำให้ข้อมูลของภาพสามารถรองรับกับข้อมูลอักขระที่จะนำมาทำการอำพรางข้อความลงไปในรูปภาพได้ โดยมีวิธีการ ดังแสดงในภาพที่ 4.1



ก) เทคนิคที่นำเสนอวิธีการอำพรางข้อความในรูปภาพบีบอัดประเภท JPEG

ภาพที่ 4.1 เทคนิคการอำพรางข้อความในรูปภาพบีบอัดประเภท JPEG



ข) เทคนิคที่นำเสนอวิธีการถอดข้อความที่มีการอำพรางในรูปภาพบีบอัดประเภท JPEG

ภาพที่ 4.1 (ต่อ)

4.1.1 หน้าจอโปรแกรม

หน้าจอโปรแกรมสำหรับผู้ช้งาน จะประกอบด้วยส่วนต่าง ๆ ตามรายละเอียดและแสดงในภาพที่ 4.2

1.1) ปุ่ม "เปิดรูปภาพ"

1.2) ช่อง "ข้อความที่ทำการฝังข้อความ"

1.3) ช่อง "ข้อความที่ถอดออกจากภาพ"

1.3) พื้นที่ว่างในการแสดงไฟล์รูปภาพชนิด JPEG

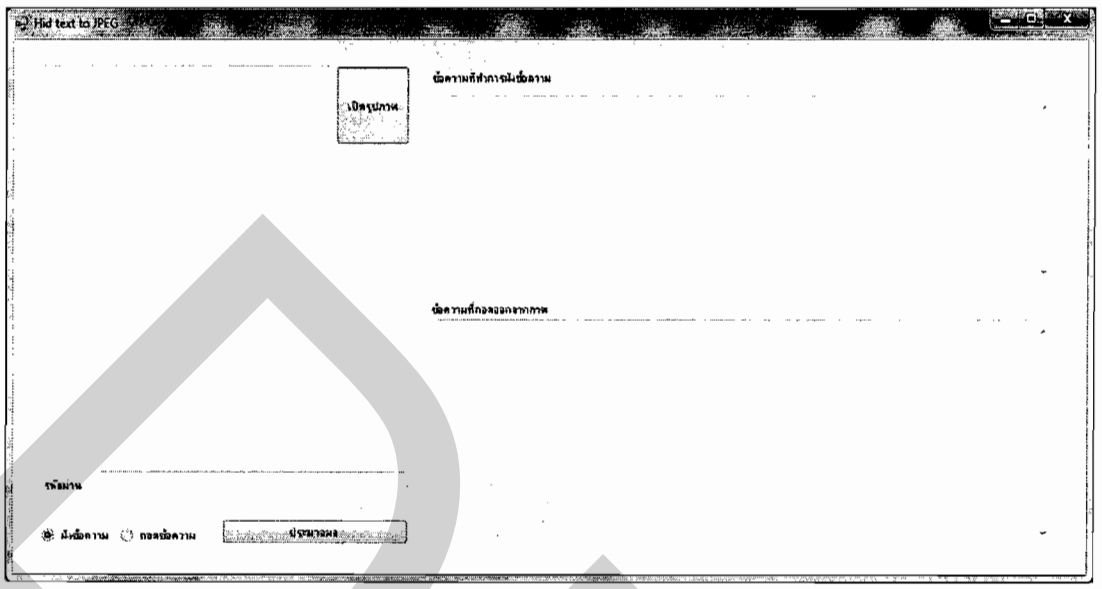
1.4) ช่อง "รหัสผ่าน"

1.5) ฟังก์ชั่นการทำงาน

1.5.1) ฝังข้อความ

1.5.2) ถอดข้อความ

1.6) ปุ่ม (ประมวลผล)

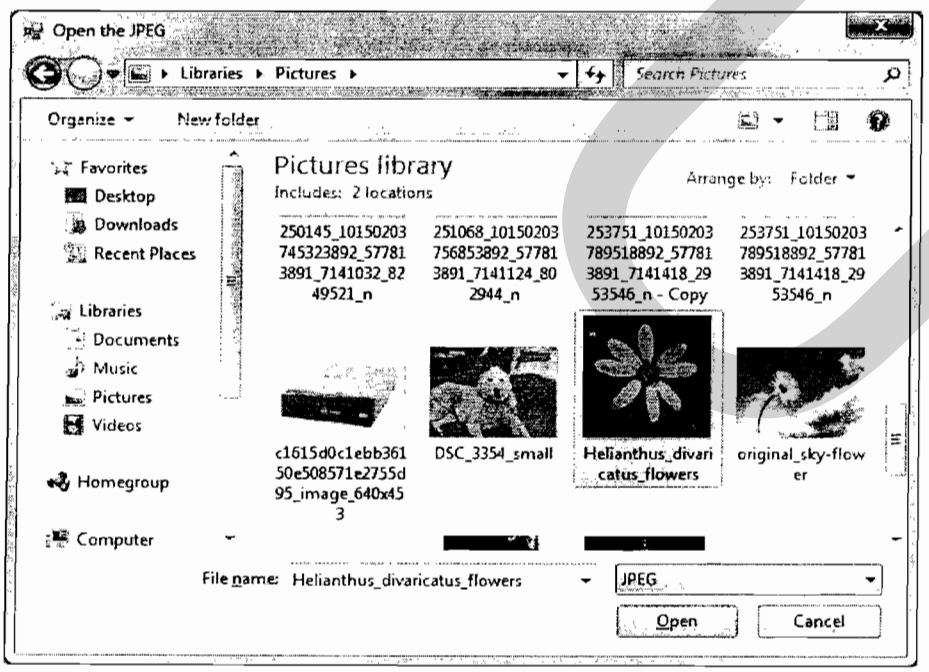


ภาพที่ 4.2 หน้าต่างโปรแกรมที่พัฒนาขึ้นมา

4.1.2 ขั้นตอนการใช้งาน การอำพรางข้อมูล

การอำพรางข้อมูลสำหรับผู้ใช้งานทางด้านผู้ส่ง จะมีขั้นตอนการใช้งานดังนี้

1) ปุ่ม "เปิดรูปภาพ" สำหรับเปิดไฟล์รูปภาพชนิด JPEG เพื่อใช้ในการอำพรางข้อมูล ดังแสดงในภาพที่ 4.3



ภาพที่ 4.3 หน้าต่างสำหรับเลือกไฟล์รูปภาพชนิด JPEG

2) ช่อง “ข้อความที่ทำการฝังข้อความ” สำหรับใส่ข้อความที่ต้องการทำการอำพรางข้อความไว้ในรูปภาพ โดยสามารถใส่ข้อความที่เป็นภาษาไทยหรือภาษาอังกฤษดังแสดงในภาพที่ 4.4

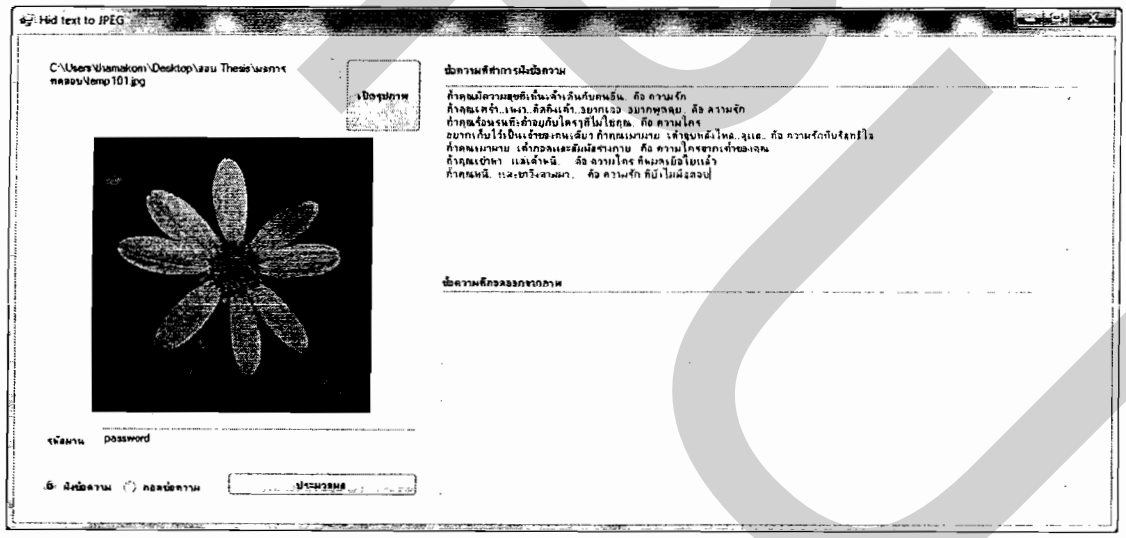
3) พื้นที่ว่างในการแสดงไฟล์รูปภาพชนิด JPEG สำหรับแสดงไฟล์รูปภาพชนิด JPEG แบบย่อส่วน

4) ช่อง “รหัสผ่าน” สำหรับใส่รหัสผ่านเพื่อเป็นการป้องกันไม่ให้บุคคลที่สามได้รับรู้หรืออ่านข้อความที่ทำการอำพรางไว้ในไฟล์ภาพชนิด JPEG

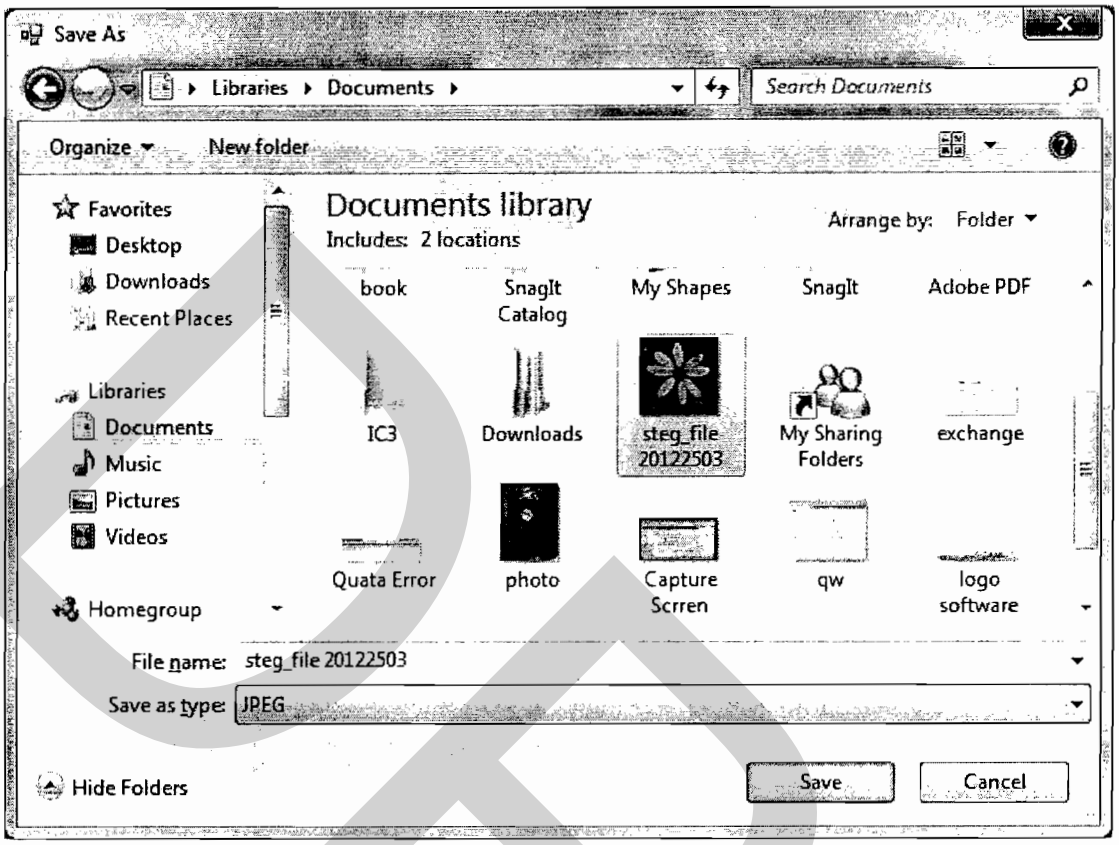
5) ฟังก์ชั่นการทำงาน

5.1) ฝังข้อความ เป็นกระบวนการที่ใช้สำหรับการอำพรางข้อความไว้ในไฟล์ภาพชนิด JPEG

6) ปุ่ม (ประมวลผล) สำหรับทำการประมวลผลโดยเริ่มกระบวนการการเข้ารหัสข้อความจนถึงกระบวนการการอำพรางข้อความ หลังจากนั้นรูปภาพจะถูกบันทึกไว้ที่เครื่องผู้ใช้งานดังแสดงในภาพที่ 4.5



ภาพที่ 4.4 หน้าจอสำหรับการใส่ข้อมูลเพื่อทำการอำพรางข้อความ



ภาพที่ 4.5 หน้าจอบันทึกรูปภาพหลังจากกดปุ่ม "ประมวลผล"

4.1.3 การอำพรางข้อมูล

การอำพรางข้อมูลนั้น จะทำการบีบอัดข้อมูลภาพ เพื่อให้ได้รูปภาพที่มีความเหมาะสมกับขนาดของข้อมูลที่เป็นประเภทอักขระในการอำพราง ซึ่งจะต้องบีบอัดในอัตราการบีบอัดข้อมูลที่เหมาะสมกับข้อมูลที่จะนำไปทำการอำพรางข้อความ ตามสมการดังนี้

$$SN_{Jpeg} = \frac{(SO_{Jpeg} - ((S_{Data} x2) + S_{Split} + S_{EndJpeg}) x100)}{SO_{Jpeg}} \quad (4\text{-}1)$$

SN_{Jpeg} คือ ขนาดของภาพบีบชนิด Jpeg ใหม่

SO_{Jpeg} คือ ขนาดของภาพบีบชนิด Jpeg เดิม

S_{Data} คือ ขนาดของข้อมูล

S_{Spilt} คือ ขนาดของกลุ่มข้อมูลที่ใช้ในการแบ่งข้อมูลเดิมกับข้อมูลข้อความที่จะทำการอำพราง

$S_{EndJpeg}$ คือ ขนาดของข้อมูลสิ้นสุดของข้อมูลภาพ

และนำข้อมูลในลักษณะข้อความ (Messages) เปลี่ยนให้เป็นข้อมูลเชิงตัวเลข และสร้างกลุ่มชุดข้อมูลที่ใช้แบ่งระหว่างข้อมูลภาพและข้อมูลตัวอักษร (Split group data) ซึ่งจะถูกแปลงเป็นข้อมูลเชิงตัวเลขด้วยรหัสผ่าน โดยใช้วิธีการ MD5 (Message-Digest algorithm 5) และนำข้อมูลที่เปลี่ยนเป็นตัวเลขทำการค้นหาตำแหน่งของข้อมูลในรูปภาพ (Cover file) ที่ตรงกับค่าของข้อมูลที่จะทำการอำพราง จะได้ตำแหน่งข้อมูล และกำหนดค่า Prefix ของชุดข้อมูลโดยทำการสุ่มค่าข้อมูลจากค่า Array {00, 01, 10, 11} ทำการหาค่า Segment และ Position ในช่วงของข้อมูลที่ความยาว 0 – 63 bit จากสมการดังนี้

$$D_S = P_p \backslash 64 \tag{4-2}$$

$$D_P = P_p Mod 64 \tag{4-3}$$

D_S คือ ค่าของ Segment

D_P คือ ค่าของตำแหน่ง

จากสมการที่ (4-2) และ (4-3) นำค่าที่หาได้ มาทำการแปลงข้อมูลให้อยู่ในรูปชุดข้อมูลแบบเลขฐานสอง จากสมการดังนี้

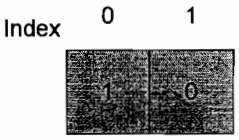
$$B_S = ConvertTobinary(D_S, 8) \tag{4-4}$$

$$B_P = ConvertTobinary(D_P, 6) \tag{4-5}$$

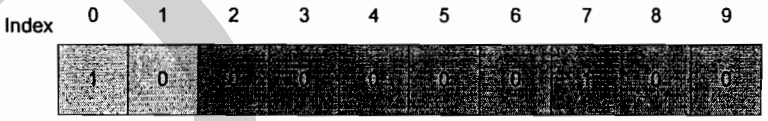
B_S คือ ค่าข้อมูล Segment เชิงเลขฐานสองทั้งหมด 8 bits

B_P คือ ค่าข้อมูล Position เชิงเลขฐานสองทั้งหมด 6 bits

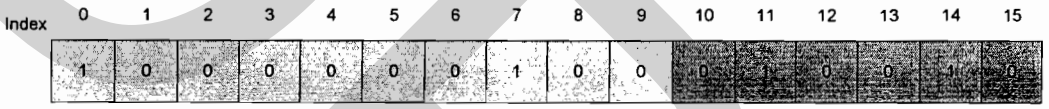
นำค่าที่ได้จากการสุ่มค่าชุดข้อมูลมาทำการต่อด้วยค่าข้อมูลจากสมการที่ (4-4) และ (4-5) เพื่อให้ได้ชุดข้อมูลที่จะนำไปทำการอำพรางข้อความ ดังแสดงในภาพที่ 4.6 หลังจากที่ได้ชุดข้อมูลทั้งหมดมี 16 bits จะทำการแบ่งข้อมูลออกเป็น 2 ส่วน ส่วนแรกจะเป็น bit ที่ 0 – 7 และ ส่วนที่สองเป็น bit ที่ 8 – 15 และนำข้อมูลทั้งสองส่วน เก็บไว้ในรูปแบบ Array พร้อมกับทำการแปลงข้อมูลเชิงเลขฐานสองให้เป็นข้อมูลเชิงตัวเลขและนำข้อมูลที่ได้ไปทำการแปลงเป็นข้อมูลให้อยู่ในรูปของตัวอักษรพร้อมกับทำการเข้ารหัสด้วยรหัสผ่าน โดยใช้วิธี XOR (Exclusive OR) แล้วนำข้อมูลที่ผ่านการเข้ารหัสแปลงข้อมูลให้อยู่ในรูปข้อมูลเชิงตัวเลขพร้อมกับนำข้อมูลและต่อท้ายด้วยชุดข้อมูล {255,217} เชิงตัวเลขที่แสดงถึงการสิ้นสุดข้อมูลภาพ และนำไปทำการอำพรางข้อความไว้ที่ส่วนสุดท้ายของข้อมูลในรูปภาพที่ถูกบีบอัดดังแสดงโครงสร้างการทำงาน ดังแสดงในภาพที่ 4.7 รายละเอียดการทำงานดังแสดงในภาพที่ 4.8



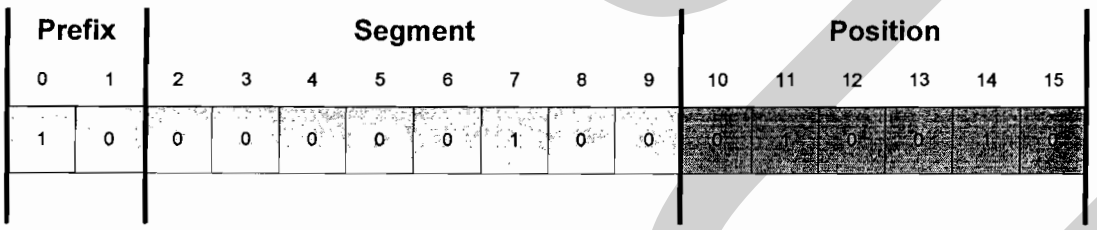
ก) กำหนดค่า Prefix ของชุดข้อมูลโดยทำการสุ่มค่าข้อมูลจากค่า 00, 01, 10 และ 11



ข) ต่อท้ายด้วยค่าข้อมูล Segment เชิงเลขฐานสองทั้งหมด 8 bits

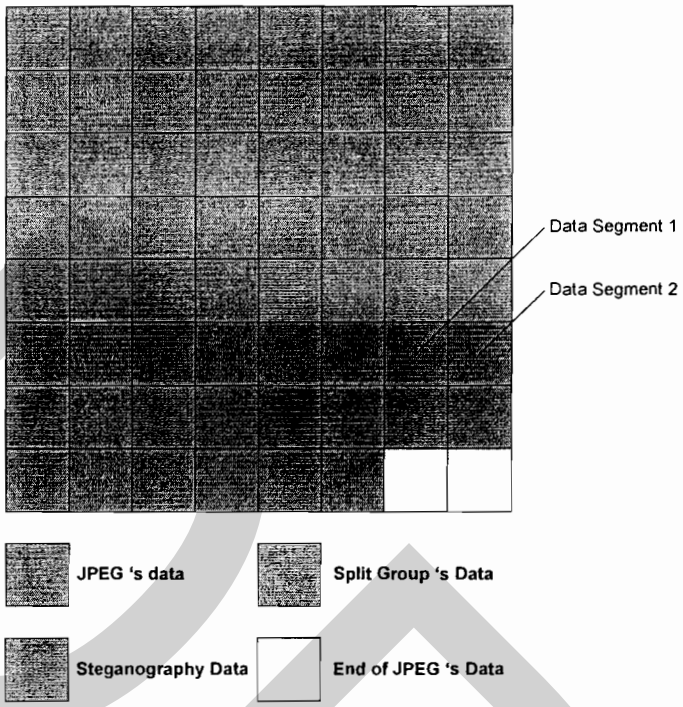


ค) ต่อท้ายด้วยค่าข้อมูล Position เชิงเลขฐานสองทั้งหมด 6 bits

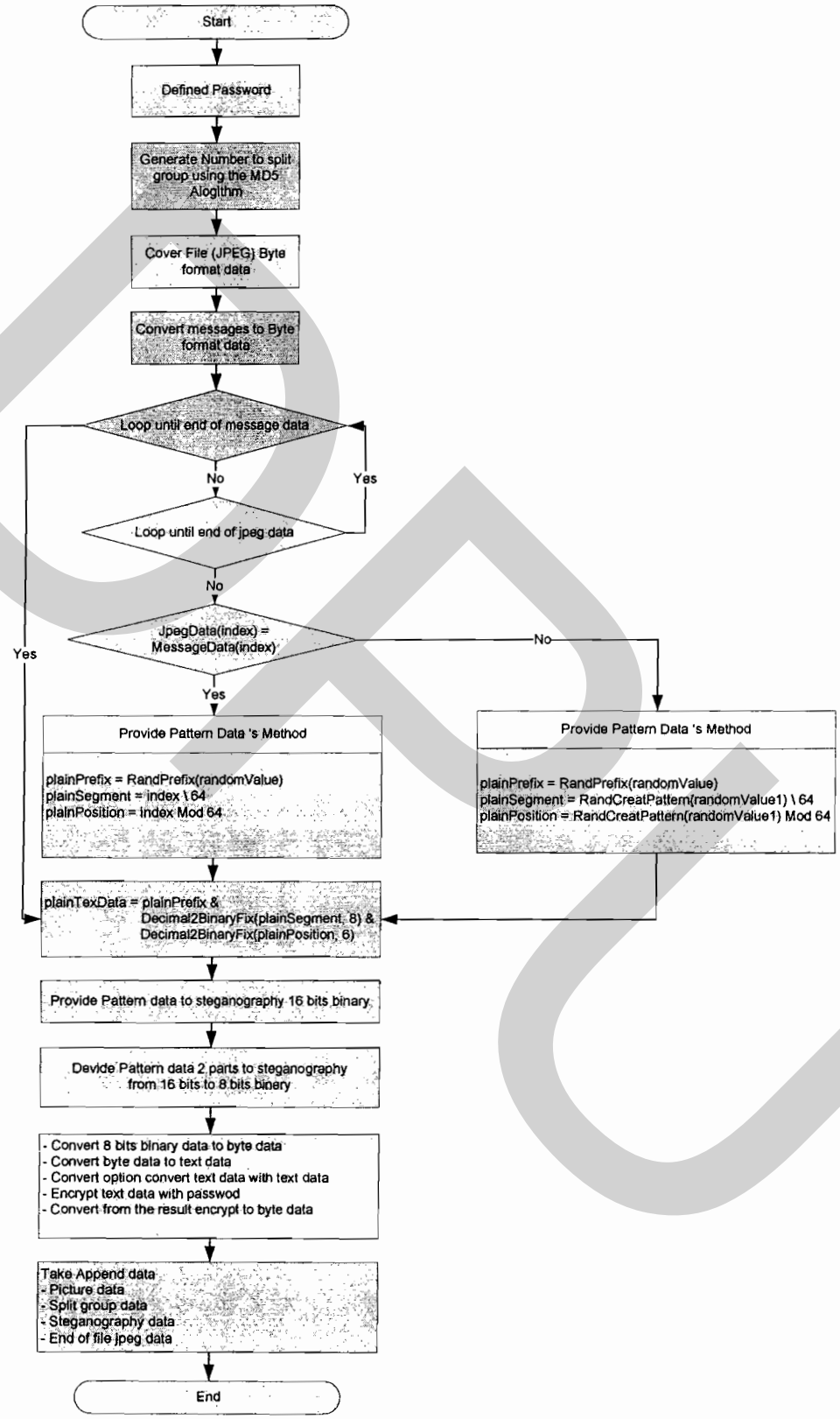


ง) ชุดข้อมูลที่ประกอบไปด้วย Prefix, Segment และ Position

ภาพที่ 4.6 การต่อท้ายข้อมูลจากสมการที่ (4-4) และ (4-5) เพื่อให้ได้ชุดข้อมูลที่จะนำไปทำการอำพรางข้อความ



ภาพที่ 4.7 โครงสร้างเทคนิคการอำพรางข้อความในรูปภาพบีบอัดประเภท JPEG

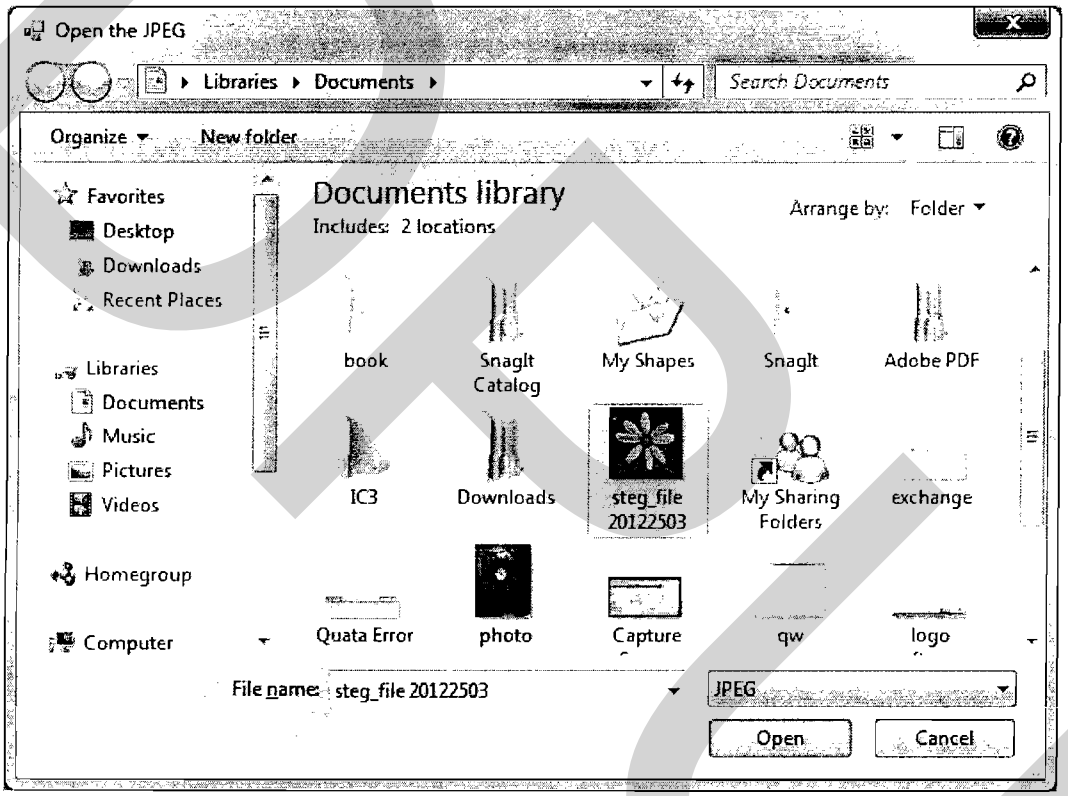


ภาพที่ 4.8 รายละเอียดการทำงานเทคนิคการอำพรางข้อความในรูปภาพบีบอัดประเภท JPEG

4.1.4 ขั้นตอนการใช้งาน การถอดข้อความที่มีการอำพราง

การถอดข้อความที่มีการอำพราง สำหรับผู้ใช้งานทางด้านผู้รับ จะมีขั้นตอนการใช้งานดังนี้

1) ปุ่ม "เปิดรูปภาพ" สำหรับเปิดไฟล์รูปภาพชนิด JPEG หลังการอำพรางข้อมูลโดยได้รับภาพมาจากจากฝั่งผู้ส่งและมีการบันทึกไว้ที่เครื่องคอมพิวเตอร์ของผู้ใช้งาน ดังแสดงในภาพที่ 4.9



ภาพที่ 4.9 หน้าต่างสำหรับเลือกไฟล์ภาพชนิด JPEG ที่มีการอำพรางข้อมูล

2) ช่อง "ข้อความที่ถอดออกจากภาพ" สำหรับแสดงข้อความที่ได้จากการอถอดข้อความในไฟล์รูปภาพชนิด JPEG

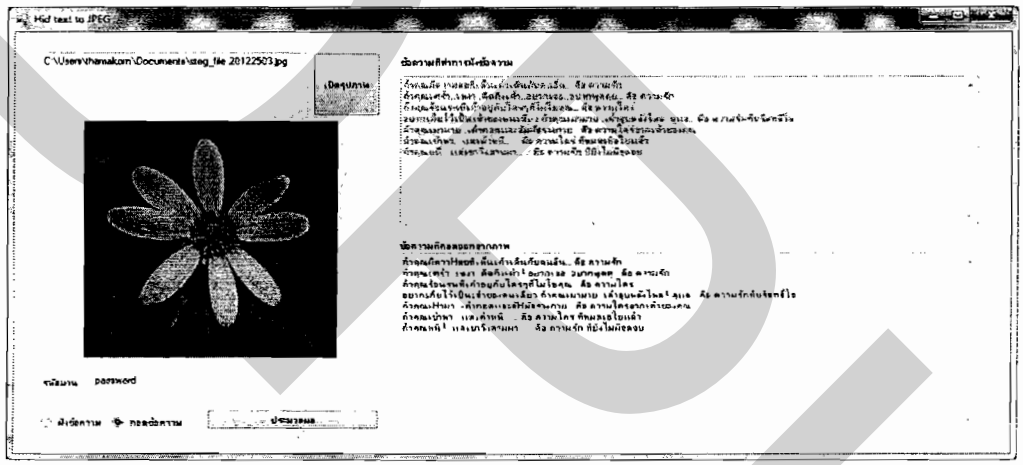
3) พื้นที่ว่างในการแสดงไฟล์รูปภาพชนิด JPEG สำหรับแสดงไฟล์รูปภาพชนิด JPEG แบบย่อส่วน

4) ช่อง "รหัสผ่าน" สำหรับใส่รหัสผ่านเพื่อทำการถอดข้อความจากไฟล์ภาพชนิด JPEG

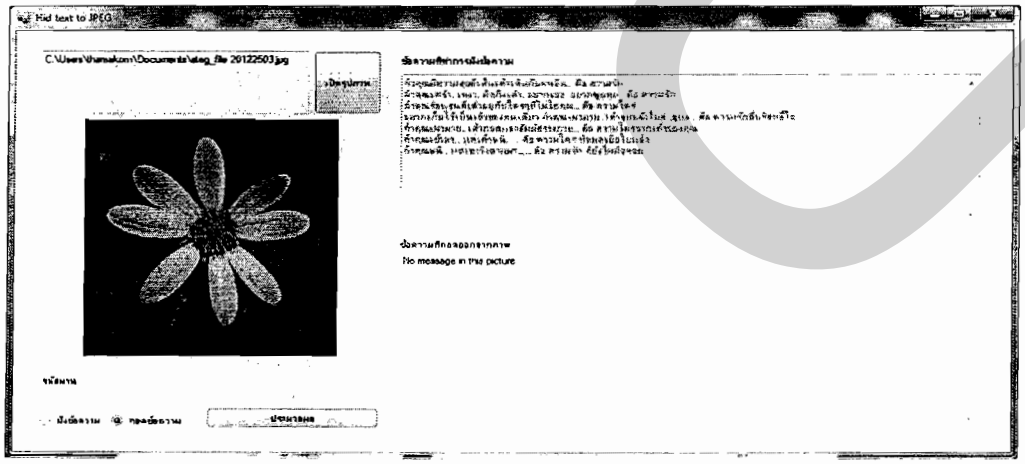
5) ฟังก์ชั่นการทำงาน

5.1) ถอดข้อความ เป็นกระบวนการที่ใช้ในการถอดข้อความออกจากไฟล์ภาพชนิด JPEG

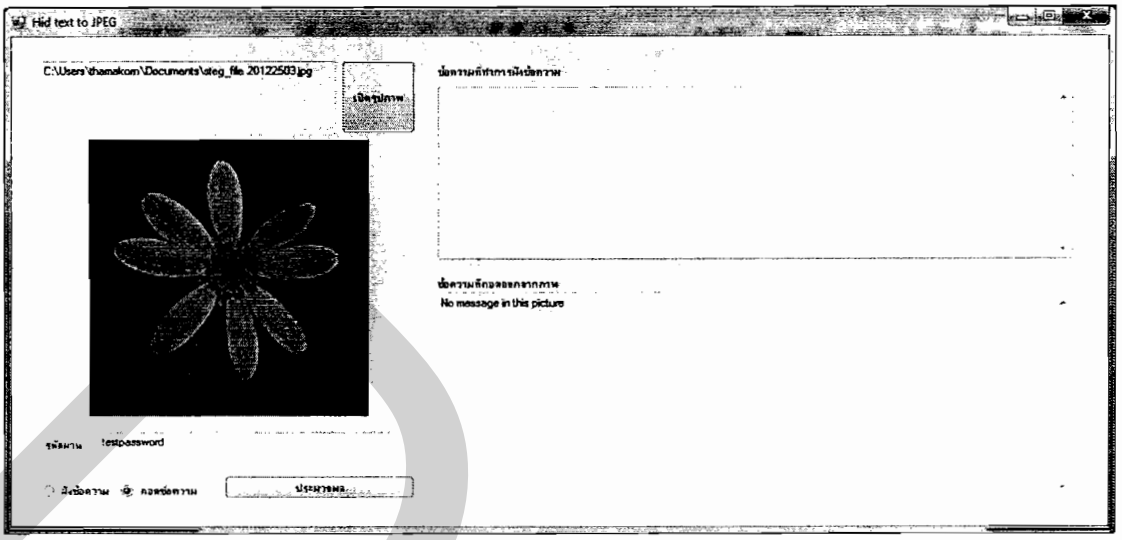
6) ปุ่ม “ประมวลผล” สำหรับทำการประมวลผลโดยเริ่มกระบวนการการถอดรหัสข้อความจนถึงกระบวนการ การถอดข้อความที่มีการอำพราง ถ้าหากรหัสผ่านที่ถูกป้อนเข้าไปในโปรแกรมถูกต้อง โปรแกรมนี้จะแสดงข้อความที่ถูกอำพรางไว้ในภาพ JPEG ดังแสดงในภาพที่ 4.10 ในกรณีที่ไม่มีการป้อนรหัสผ่านหรือป้อนรหัสผ่านผิด โปรแกรมจะแสดงข้อความ “No Message in this picture” ดังแสดงในภาพที่ 4.11 และภาพที่ 4.12 ตามลำดับ



ภาพที่ 4.10 หน้าจอแสดงข้อความ ที่มีการถอดข้อความจากไฟล์ภาพชนิด JPEG



ภาพที่ 4.11 หน้าจอแสดงข้อความในช่อง “ข้อความที่มีการถอดออกจากภาพ” เมื่อไม่มีการใส่รหัสผ่าน



ภาพที่ 4.12 หน้าจอแสดงข้อความในช่อง "ข้อความที่มีการถอดออกจากภาพ" เมื่อมีการใส่รหัสผ่านผิด

4.1.5 การถอดข้อความที่มีการอำพราง

การทำงานในส่วนของการถอดข้อความที่มีการอำพราง เป็นการค้นหาข้อมูลในส่วนของกลุ่มข้อมูลที่ทำการแบ่งระหว่างกลุ่มข้อมูลภาพที่ทำการอำพรางข้อความ (Steg file) โดยกลุ่มของข้อมูลที่ทำการแบ่งนั้นจะถูกสร้างจากรหัสผ่านโดยใช้วิธีการ MD5 แล้วทำการค้นหากลุ่มข้อมูลที่ใช้แบ่งระหว่างข้อมูลภาพและข้อมูล ตัวอักษร และทำการแปลงข้อมูลที่อยู่ในรูปเชิงข้อมูลไบต์ให้กลายเป็น ข้อมูลเชิงตัวอักษร หลังจากนั้น ทำการถอดรหัสด้วยรหัสผ่านโดยใช้ วิธีการ XOR พร้อมกับทำการแปลงข้อมูลที่ถอดรหัสออกมาได้ให้อยู่ใน รูปข้อมูลเชิงไบต์และแปลงข้อมูลเชิงไบต์ให้อยู่ในรูปชุดข้อมูลเชิงฐานสองมีความยาว 16 บิต ดังแสดงในภาพที่ 4.13

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0

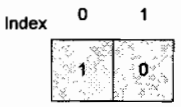
ภาพที่ 4.13 ตัวอย่างชุดข้อมูล 16 bits

โดยข้อมูลที่ได้จะอยู่ในรูปแบบข้อมูลเชิงฐานสองนั้น จะมีข้อมูลต่อท้ายด้วยข้อมูลของตัวถัดไปโดยได้ผลลัพธ์มาจากวิธีการเดียวกันกับวิธีการข้างต้น เท่ากับว่า จะได้ข้อมูล 1 ชุด ที่มี

ความยาวทั้งหมด 16 bits ดังแสดงในภาพที่ 4.13 และนำข้อมูล 16 bits นั้น มาทำการแยกเพื่อหาค่าของ Prefix, Segment และ Position โดยแยกตามวิธีการดังนี้

การหาค่าชุดข้อมูลของ Prefix

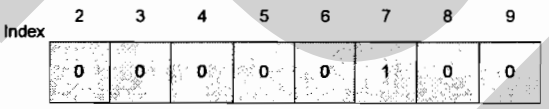
จากชุดข้อมูล 16 bits ทำการแยกข้อมูลโดยนำข้อมูลตั้งแต่ Index ที่ 0 – 1 จากชุดข้อมูลทั้งหมด จะได้ความยาวข้อมูล 2 bits ในรูปข้อมูลเชิงเลขฐานสอง ดังแสดงในภาพที่ 4.14



ภาพที่ 4.14 ตัวอย่างชุดข้อมูล 2 bits

การหาค่าชุดข้อมูลของ Segment

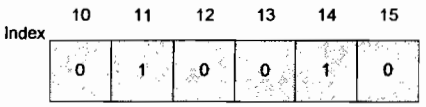
จากชุดข้อมูล 16 bits ทำการแยกข้อมูลโดยนำข้อมูลตั้งแต่ Index ที่ 2 – 9 จากชุดข้อมูลทั้งหมด จะได้ความยาวข้อมูล 8 bits ในรูปข้อมูลเชิงเลขฐานสอง คือค่า B_S ดังแสดงในภาพที่ 4.15



ภาพที่ 4.15 ตัวอย่างชุดข้อมูล 8 bits

การหาค่าชุดข้อมูลของ Position

จากชุดข้อมูล 16 bits ทำการแยกข้อมูลโดยนำข้อมูลตั้งแต่ Index ที่ 10 – 15 จากชุดข้อมูลทั้งหมด จะได้ความยาวข้อมูล 6 bits ในรูปข้อมูลเชิงเลขฐานสอง คือค่า B_P ดังแสดงในภาพที่ 4.16



ภาพที่ 4.16 ตัวอย่างชุดข้อมูล 6 bits

แล้วนำชุดข้อมูลของ Segment และ Position มาทำการแปลงข้อมูลให้อยู่ในรูปเชิง Byte และทำการหาค่าตำแหน่งที่แท้จริง ซึ่งเป็นตำแหน่งที่ระบุถึงข้อมูลตัวอักษรที่ได้อ้างอิงถึง จากสมการดังนี้

$$D_S = ConvertToDecimal(B_S,8) \quad (4\text{-}6)$$

$$D_P = ConvertToDecimal(B_P,8) \quad (4\text{-}7)$$

D_S คือค่าของ Segment ในรูปแบบข้อมูลเชิง byte

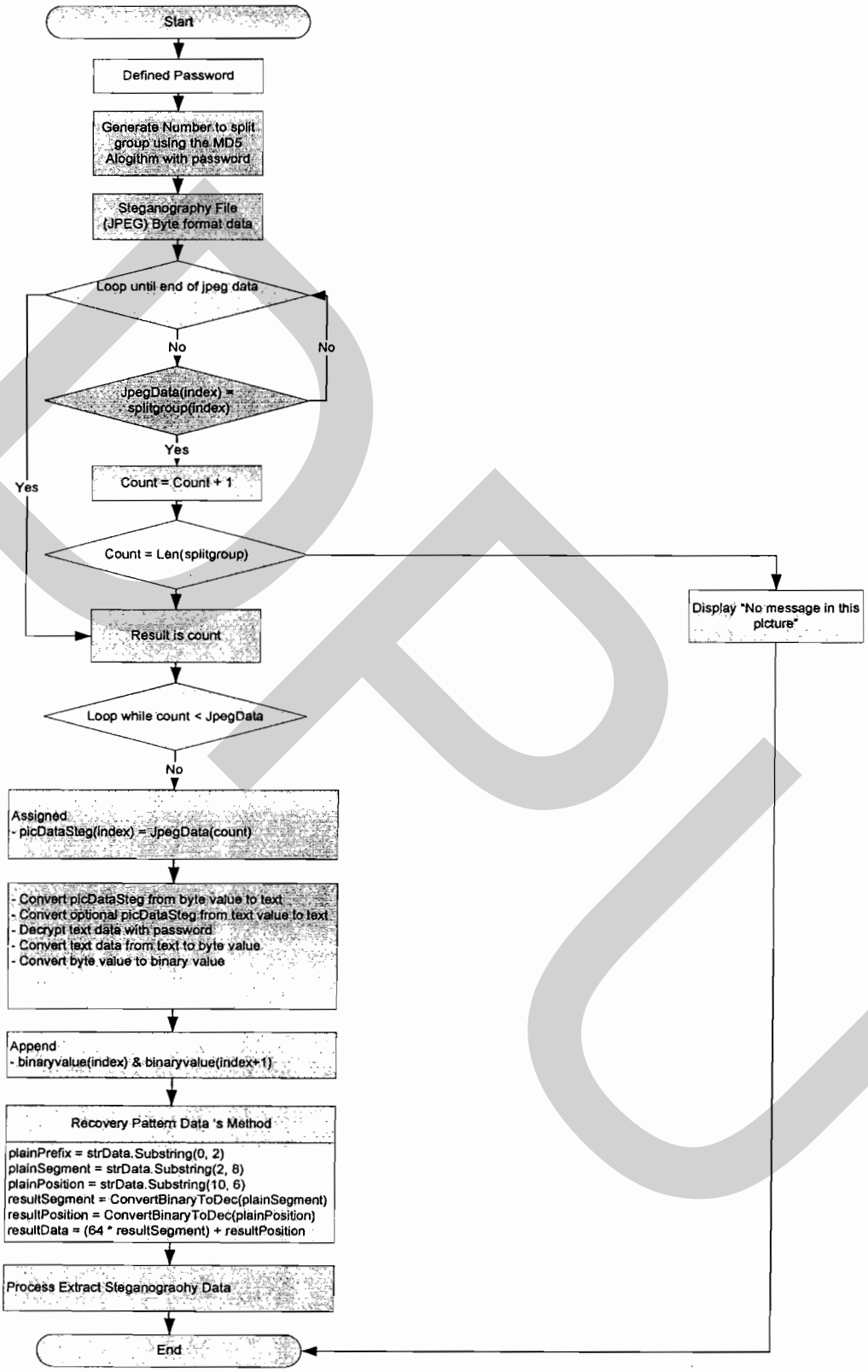
D_P คือค่าของ Position ในรูปแบบข้อมูลเชิง byte

จากสมการ ที่ (4-6) และ (4-7) สามารถหาค่าตำแหน่งที่ถูกทำการอำพราง จากสมการดังนี้

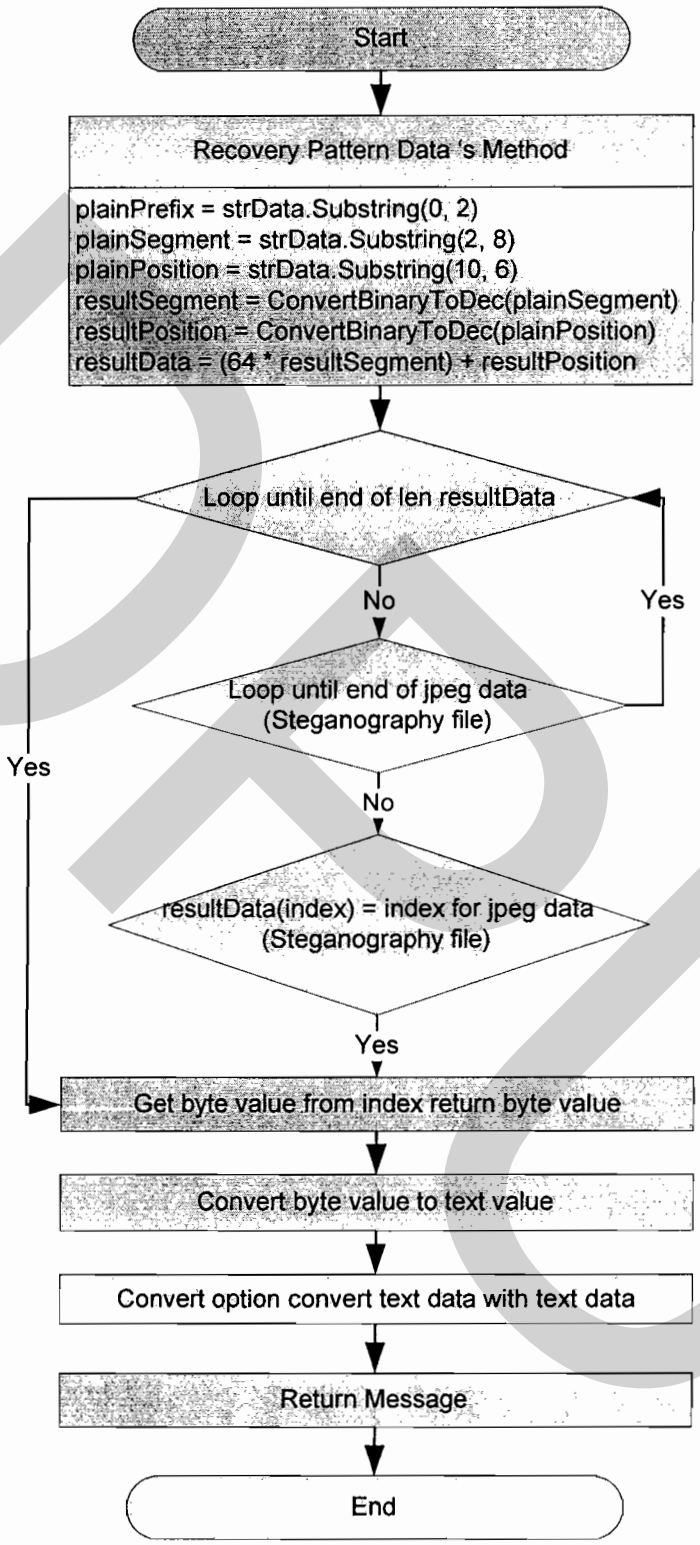
$$P_P = (64xD_S) + D_P \quad (4\text{-}8)$$

P_P คือค่าของตำแหน่งที่ถูกทำการอำพราง

จากสมการ ที่ (4-8) เมื่อได้ค่าตำแหน่งของข้อมูลแล้ว นำไปทำการค้นหาข้อมูลที่อยู่ตำแหน่งในข้อมูลภาพ เพื่อให้ได้ตัวอักษรที่เป็นค่าข้อมูลเชิง byte และแปลงข้อมูลเชิง byte ที่หาพบให้เป็นข้อมูลเชิงตัวอักษร และนำข้อมูลเชิงตัวอักษรนั้นทำการแปลงอยู่ในรูปข้อมูลเชิงตัวอักษรอีกครั้ง เพื่อให้รองรับตัวอักษรภาษาไทย รายละเอียดการทำงานตาม ดังแสดงในภาพที่ 4.17 และ 4.18



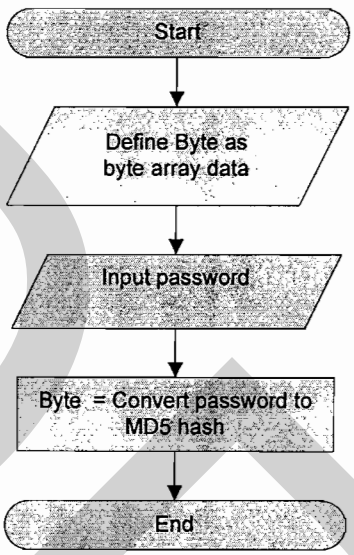
ภาพที่ 4.17 รายละเอียดขั้นตอนการทำงานสำหรับการถอดข้อความจากภาพบีบอัดประเภท JPEG



ภาพที่ 4.18 รายละเอียดขั้นตอนการทำงานถอดข้อความในรูปภาพบีบอัดประเภท JPEG

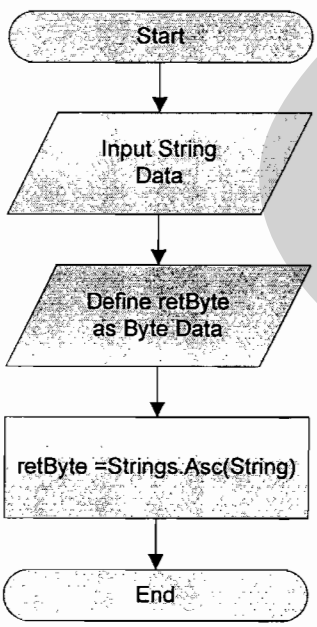
4.2 การเขียนโปรแกรมอำพรางข้อความ

4.2.1 สร้างบล็อกเริ่มต้นของข้อความที่จะทำการอำพรางข้อความ ดังแสดงในภาพที่ 4.19



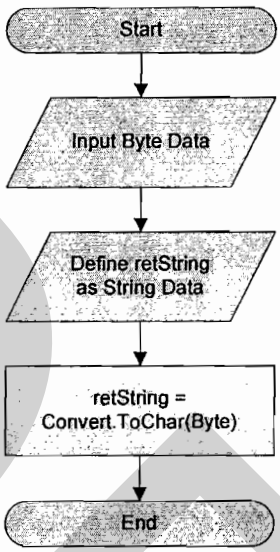
ภาพที่ 4.19 ขั้นตอนการทำงานของการสร้างบล็อกเริ่มต้น

4.2.2 การแปลงตัวษรให้เป็นค่า Byte ดังแสดงในภาพที่ 4.20



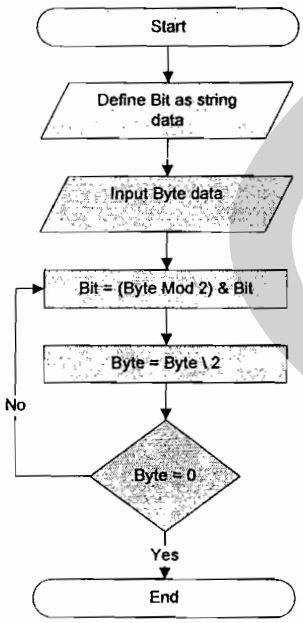
ภาพที่ 4.20 ขั้นตอนการทำงานการแปลงค่าตัวอักษรให้เป็นค่า Byte

4.2.3 การแปลงค่า Byte ให้เป็นตัวอักษร ดังแสดงในภาพที่ 4.21



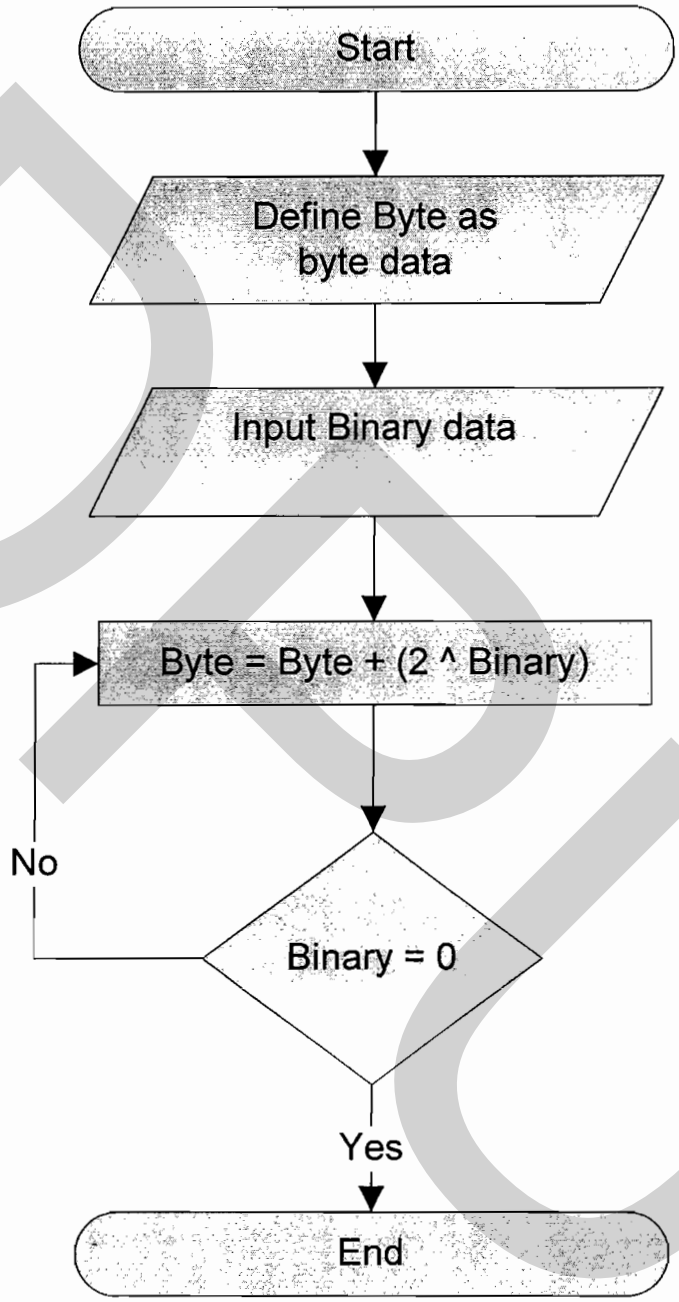
ภาพที่ 4.21 ขั้นตอนการทำงานการแปลงค่า Byte ให้เป็นตัวอักษร

4.2.4 แปลงค่าข้อมูลเชิง Byte ให้อยู่ในรูปชุดข้อมูลเชิงแบบเลขฐานสอง ดังแสดงในภาพที่ 4.22



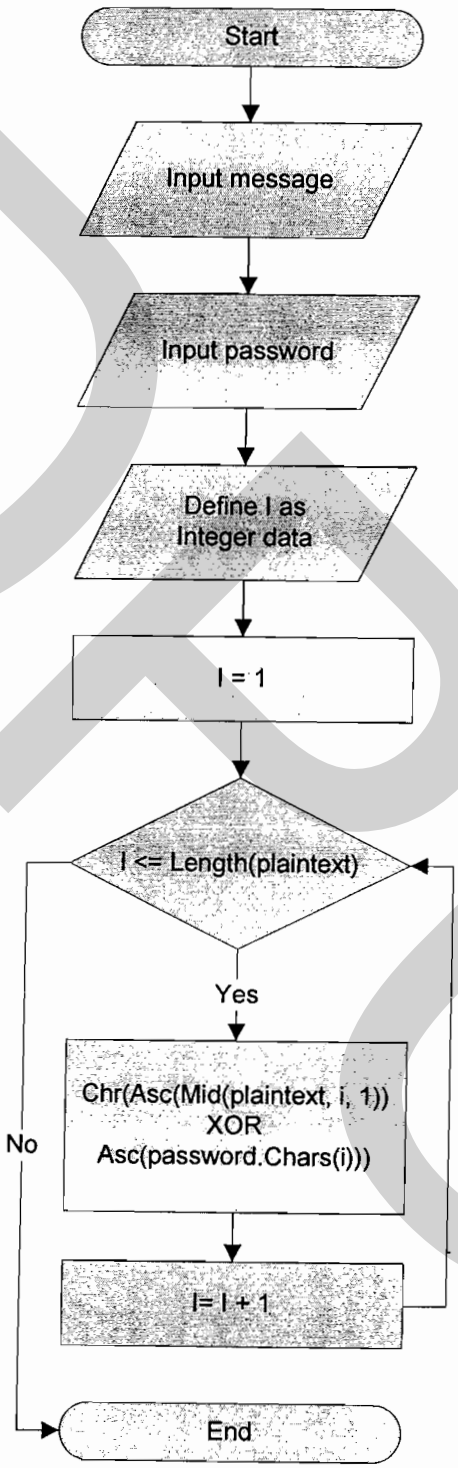
ภาพที่ 4.22 ขั้นตอนการทำงานการแปลงค่า Byte ให้เป็นเลขฐานสอง

4.2.5 แปลงค่าข้อมูลเชิงแบบเลขฐานสองให้เป็นชุดข้อมูลเชิงตัวเลข ดังแสดงในภาพที่ 4.23



ภาพที่ 4.23 ขั้นตอนการทำงานการแปลงค่าตัวเลขฐานสองให้เป็นค่า Byte

4.2.6 การเข้ารหัสและถอดรหัสข้อมูล ดังแสดงในภาพที่ 4.24



ภาพที่ 4.24 ขั้นตอนการทำงานการเข้ารหัสและการถอดรหัสข้อมูล

4.3 ผลทดสอบประสิทธิภาพด้านการทำงานของโปรแกรม

หลังจากได้พัฒนาโปรแกรมการอำพรางข้อมูลไว้ในไฟล์ภาพสำหรับเครื่องคอมพิวเตอร์ผู้ใช้งาน ได้แบ่งแนวทางในการประเมินประสิทธิภาพในมิติต่าง ๆ โดยทดสอบประสิทธิภาพของโปรแกรมด้วยการเปรียบเทียบกับวิธีการ DCT และ JPEGX ดังนี้

1) กำหนดชุดข้อมูลที่ใช้ในการทดสอบ โดยแบ่งข้อมูลออกเป็น 2 ประเภท คือ ภาพที่มีการบีบอัดข้อมูลประเภท JPEG และข้อความที่ใช้ในการทดสอบ

1.1) ข้อความที่ใช้ในการทดสอบ

- ข้อความภาษาไทย
- ข้อความภาษาอังกฤษ
- ข้อความภาษาไทยปนกับภาษาอังกฤษ

โดยแต่ละข้อความมีขนาด100,200,300,1000,2500,3000, 5000, 10000, 15000, 25000, 30000 และ 33800 byte

ถ้าคุณมีความสุขที่เห็นเค้าเดินกับคนอื่น... คือ ความรัก
ถ้าคุณเศร้า..เหงา..คิดถึงเค้า..อยากเจอ..อยากพูดคุย... คือ ความรัก
ถ้าคุณร้อนรนที่เค้าอยู่กับใครๆที่ไม่ใช่คุณ... คือ ความใคร่
อยากเก็บไว้เป็นเจ้าของคนเดียว ถ้าคุณเมามาย..เค้าลูบหลังไหล่..ดูแล... คือ ความรักที่บริสุทธิ์ใจ
ถ้าคุณเมามาย..เค้ากอดและสัมผัสร่างกาย... คือ ความใคร่จากเค้าของคุณ
ถ้าคุณเข้าหา.. แต่เค้าหนี... ... คือ ความใคร่ ที่หมดเยื่อใยแล้ว

If you are happy to see me walking with someone else ... is love.
If you think about it .. I .. I .. I .. I would say ... is love.
If he is impatient with anyone other than you ... is tender.
I kept one owner. If you binge .. He rubbed shoulders back .. take care ... love is pure.
If you binge .. He hugged and touched my body ... the one from your
If you approach .. But he is the one who runs away ..... and tie.
If you are running away .. but he ..... is love no end.

ภาพที่ 4.25 ตัวอย่างข้อมูลประเภทอักขระที่ใช้ในการอำพรางข้อความ

1.2) ภาพบีบอัดประเภท JPEG

ภาพบีบอัดประเภท JPEG ที่ผ่านการบีบอัด ซึ่งประกอบไปด้วยภาพทั้งหมด 6 ชุด ดังแสดงตัวอย่างในภาพที่ 4.26 และมีขนาดดังต่อไปนี้

128 x 96 pixels

256 x 192 pixels

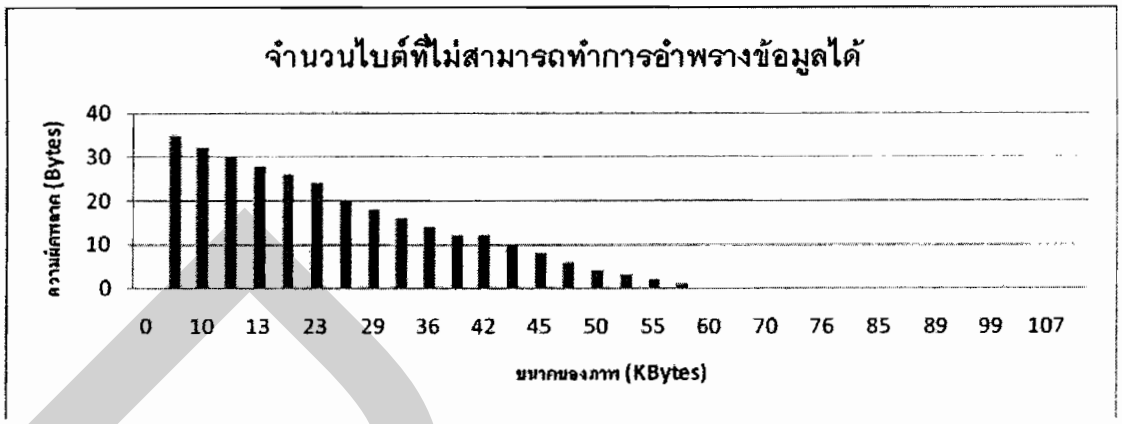
512 x 384 pixels

1024 x 768 pixels

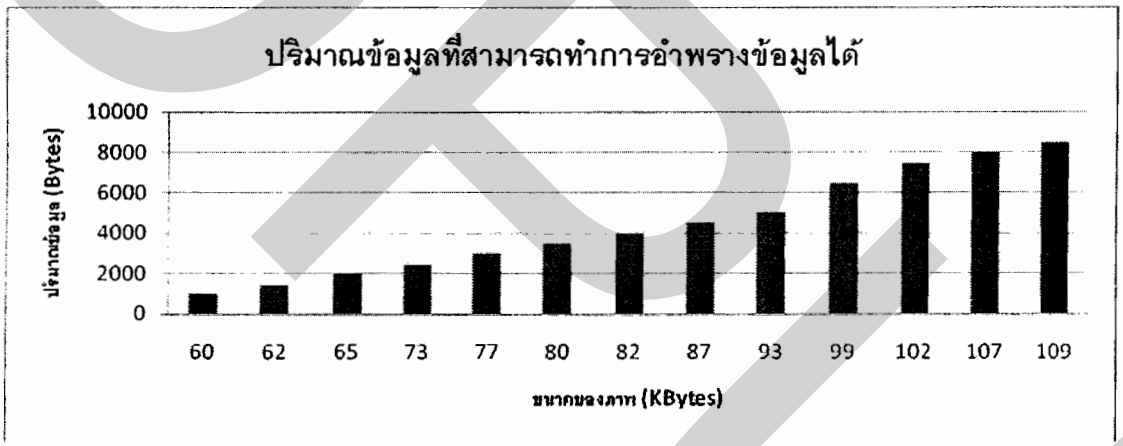
2048 x 1536 pixels

ภาพที่ 4.26 ตัวอย่างชุดไฟล์ภาพชนิด JPEG ที่ใช้สำหรับการอำพรางข้อมูล

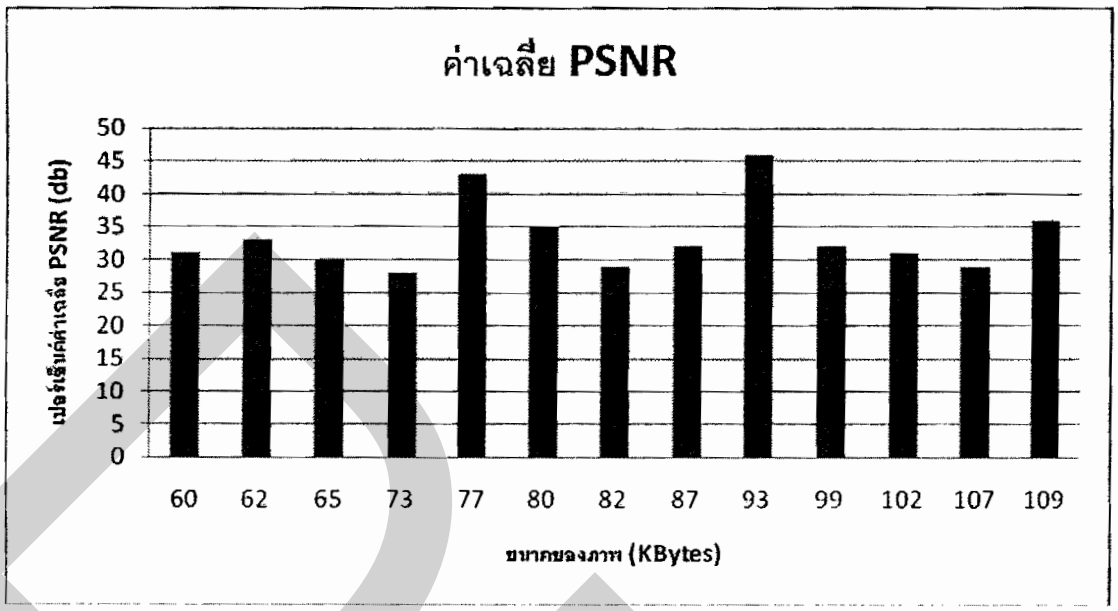
2) ผลการประเมินประสิทธิภาพของโปรแกรม ผู้วิจัยได้ทำการทดลองโดยใช้ข้อความตามชุดการทดสอบ และรูปภาพจำนวน 30 รูปภาพ โดยแบ่งออกเป็น 2 ชุด ประกอบไปด้วย รูปภาพที่ 1-15 เป็นรูปภาพที่มีขนาดแตกต่างกันแต่ภาพชุดเดียวกัน และ รูปภาพที่ 16-30 เป็นรูปภาพที่มีขนาดเท่ากันแต่ภาพคนละชุด ซึ่งสามารถสรุปผลได้ดังนี้ การนำภาพบีบอัดชนิด JPEG โดยมีข้อมูลที่เป็นข้อความประเภทอักขระ ดังตัวอย่างข้อความสำหรับใช้ในการอำพรางข้อมูล ซึ่งการวัดประสิทธิภาพนั้น ได้ใช้วิธีการวัดค่าความเพี้ยนของข้อมูลภาพ และนำข้อมูลภาพที่ใช้ในการทดลองมาทำการตรวจวัดค่าความเพี้ยนของข้อมูลภาพ โดยทำการทดลองและสรุปดังแสดงความสัมพันธ์ต่อการทดลอง ดังต่อไปนี้



ภาพที่ 4.27 จำนวนไบต์ที่ไม่สามารถทำการอำพรางข้อมูลได้ เมื่อข้อความมีขนาด 1,000 ไบต์

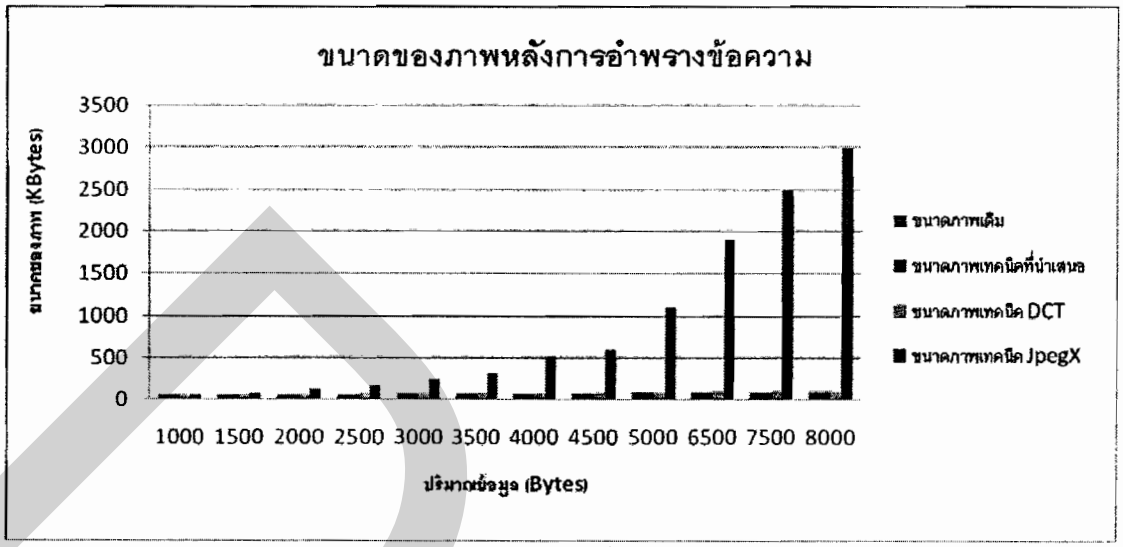


ภาพที่ 4.28 ปริมาณข้อมูลที่สามารถทำการอำพรางข้อมูลได้โดยไม่มีข้อผิดพลาดเกิดขึ้น เมื่อข้อความมีขนาด 1,000 ไบต์



ภาพที่ 4.29 ค่าเฉลี่ยของ PSNR (db) สำหรับข้อมูลที่สามารถทำการอำพรางข้อมูลได้โดยไม่มีข้อผิดพลาดเกิดขึ้น เมื่อข้อความมีขนาด 1,000 ไบต์

จากภาพที่ 4.27 และ ภาพที่ 4.28 เป็นตัวอย่างของสถิติที่ได้จากการทดสอบค่าของภาพที่นำมาใช้ในการทดลอง โดยทำการทดสอบจำนวนทั้งหมด 30 ภาพ ซึ่งแสดงให้เห็นสถิติ จำนวนไบต์ที่ไม่สามารถทำการอำพรางข้อมูลได้และปริมาณข้อมูลที่สามารถทำการอำพรางข้อมูลได้ในแต่ละภาพ และภาพที่ 4.29 เป็นภาพค่าเฉลี่ยของ PSNR (Peak-Signal to Noise Ration) ซึ่งเป็นค่ามาตราฐานที่นำมาใช้ในการเปรียบเทียบคุณภาพของรูปภาพดิจิตอลโดยการเปรียบเทียบกับรูปภาพต้นฉบับ ถ้าค่า PSNR สูงจะชี้ให้เห็นถึงคุณสมบัติของรูปที่ใกล้เคียงกับรูปต้นฉบับของแต่ละภาพเพื่อดูความแตกต่างของข้อมูลภาพที่เกิดขึ้นหลังจากมีการอำพรางข้อมูล และ จากภาพที่ 4.30 เป็นการแสดงการบันทึกค่าปริมาณข้อมูลที่สามารถทำการอำพรางข้อมูลได้โดยไม่มีข้อผิดพลาดเกิดเมื่อเทียบ JPEGX



ภาพที่ 4.30 ขนาดของภาพที่สามารถรองรับปริมาณข้อมูลในการอำพรางข้อความโดยที่ไม่มีข้อผิดพลาดเกิดขึ้น

จากภาพที่ 4.30 ความสัมพันธ์การเปรียบเทียบขนาดของภาพระหว่างเทคนิคที่นำเสนอและเทคนิค JPEGX ซึ่งเทคนิคที่นำเสนอนั้นสามารถทำการอำพรางข้อความได้ดีและขนาดของภาพมีขนาดที่น้อยกว่าเทคนิค JPEGX

บทที่ 5
สรุปผลและข้อเสนอแนะ

งานวิจัยนี้ต้องการศึกษาวิธีการอำพรางข้อมูลที่เป็นข้อมูประเภทอักขระ โดยมีวัตถุประสงค์เพื่อการอำพรางข้อมูลอันเป็นการป้องกันไม่ให้บุคคลที่สามได้รับรู้ และเพื่อออกแบบวิธีการให้มีความเหมาะสมต่อภาพประเภท JPEG กับปริมาณข้อมูลที่ต้องการอำพรางในภาพ ดังมีรายละเอียดต่อไปนี้

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอเทคนิคที่เป็นวิธีการอำพรางข้อมูล ที่เป็นข้อมูลประเภทอักขระ ซึ่งสามารถทำการอำพรางข้อมูลในรูปภาพชนิด JPEG ได้เป็นอย่างดี จากการทดสอบโดยอ้างอิงวิธีการ JPEGX พบว่าเทคนิคที่นำเสนอนี้มีประสิทธิภาพที่ดีกว่าดังข้อมูลตามตามรางดังต่อไปนี้

ตารางที่ 5.1 ประสิทธิภาพการทำงานเมื่อเทียบกับวิธีการที่นำเสนอและวิธีการของ JPEGX

Process	JPEFX	Proposed Technique
Hiding Method	Append	Append
Encrypt	Substitution	Substitution / Picture 's position
Decrypt	1 Step	2 Step
Disadvantage	Bigger file size	Lossy information
Advantage	No distortion picture	Slightly distortion picture / Slightly change in file size

5.2 ข้อเสนอแนะ

งานวิจัยที่นำเสนอนี้พบว่า กระบวนการ การแทนที่ข้อความบางตัวอักษรอาจจะไม่สามารถอำพรางได้ เนื่องจากไม่สามารถหาตำแหน่งของข้อมูลในภาพมาใช้ในการอำพรางข้อมูลได้ ทำให้การอำพรางข้อมูลไม่สามารถอำพรางได้ครบถ้วนตามจำนวนตัวอักษรของข้อความ และ

เนื่องจากไม่สามารถหาตำแหน่งในข้อมูลภาพมาใช้ในการอำพรางข้อมูลได้ ทั้งนี้เพื่อให้เทคนิคที่นำเสนอนี้เกิดประสิทธิภาพในการใช้งานและก่อให้เกิดประโยชน์สูงสุด ควรมีพัฒนาต่อดังนี้

5.1.1 โปรแกรมควรมีความสามารถ สำหรับวิธีการซ่อนข้อมูลที่เป็นข้อความประเภทอักขระลงในทุก ไบต์ ของข้อมูลรูป JPEG

5.1.2 โปรแกรมควรมีความสามารถ ในการอำพรางข้อมูลได้ทุกภาษาโดยใช้หลักการทำงานของ Page Code

บรรณานุกรม

บรรณานุกรม

ภาษาไทย

ธำรงรัตน์ อมรรักษา และ วัชร พิชยนันท์ (2546). ภาพพิมพ์ลายน้ำดิจิตอล : วิธีป้องกันการละเมิดสิทธิทางปัญญาสำหรับรูปภาพ. *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, 13(2), 54-63.

ภาษาต่างประเทศ

Ali Akbar Nikoukar. (2010). An Image Steganography Method with High Hiding Capacity Based on RGB Image. *International Journal of Signal and Image Processing*, 1, 238-241.

Andrew B. Watson. (1994). Image Compression Using the Discrete Cosine Transform. *Mathematica Journal*, 4, 81-88.

C. Yun – Qing Shi, Sui Song, Zheng Zhang, Zhicheng Ni and Dekun Zou. Detection of block DCT-based, steganography in gray-scale images. *Manikopoulos*, 335 – 358.

Chiang-Lung Liu and Shiang-Rong Liao. (2008). High-peroformance JPEG steganography using complementary embedding strategy. *The journal of the pattern recognition society*, 41, 2945 – 2955.

Christine Bako. (2004). JPEG 2000 Image Compression. *Analog Dialogue*, 38-09.

Fitri Arnia, Ikue Iizuka, Masaaki Fujiyoshi and Hitoshi Kiya. (2007). DCT Sign – Based Similarity Measure for JPEG Image Retrieval. *IEICE Transaction. Fundmentals*, E90-A(9), 1976 – 1985.

Ian Curry. (2001). An Introduction to Cryptography and Digital Signatures. *Entrust Securing Digital Identities & Information.*

Jessica Fridrich, Miroslav Gojjan and Rui Du. (2001). Reliable Detection of LSB Steganography in Color and Grayscale Images. *MM&Sec 01 Proceedings of the 2001 workshop on Multimedia and Security*, 27-30.

Johnson, N. F., and S. Jajodia. (1998) Exploring Steganography: Seeing the Unseen. *IEEE Computer*, 31(2), 26-34.

Johnson, N. F., and S. Jajodia. (1998) "Exploring Steganography: Seeing the Unseen". *IEEE Computer*, 31(2), 26-34.

Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett. (2004). "Steganography and Digital Watermarking", School of Computer Science, The University of Birmingham.

Lala Krikor, Sami Baba, Thawar Arif and Zyad Shaaban. (2009). "Image Excryption Using DCT and Stream Cipher". *European Journal of Scientific Researchm,* 32(1), 47 – 57.

Nabarun Bagchi. (2010). "Secure BMP Image Steganography Using Dual Security Model (I.D.E.A, image intensity and Bit Randomization and Max – Bit Algorithm". *International Journal of Computer Applications*, 1.1(21), 18 – 22.

R. Rivest. (April 1992). "The MD5 Message-Digest Algorithm". *RFC 1321, MIT LCS & RSA* Data Security, Inc.

Shrikant S. Khaire and Sanjay L. Nalbalwar. (2010). "Review : Steganography – Bit Plane Complexity Segmentation (BPCS) Technique". *International Journal of Engineering Science and Technology*, 2(9), 4860-4868.

Tarun Barayan Shankar and G.Sahoo. (2010). "Cryptography by Karatsuba Multiplier with ASCII Codes". *International Jpurnal of Computer Applications*, 10(12), 53-60.

ภาคผนวก

1) Microsoft Visual Basic.NET

Visual Basic.NET เป็นภาษาที่พัฒนาต่อจาก Visual Basic 6.0 เป็นภาษาที่สนับสนุนโครงสร้างของภาษาที่เป็น OOP ทำให้โครงสร้างภาษาของ Visual Basic.NET นั้นมีความสมบูรณ์มากขึ้น ซึ่ง Visual Basic.NET เป็นภาษาหนึ่งที่อยู่ในชุดเครื่องมือ Microsoft Studio.NET โดยจะใช้ IDE (Integrated Development Environment) รวมกับภาษาอื่นอีก 3 ภาษา ที่อยู่ในชุดเครื่องมือนี้ ซึ่งได้แก่ Visual C#, Visual C++ และ Visual J#

ลักษณะสำคัญของ Microsoft Visual Basic.NET ในการวิจัยและพัฒนา คือ

- OOP

การเป็น Software ที่สนับสนุนการเขียนโปรแกรมในเชิงวัตถุ ทำให้สามารถสร้างการสืบทอดในการเขียนโปรแกรมได้อย่างมีประสิทธิภาพและลดการเขียนฟังก์ชันในการทำงานได้อย่างรวดเร็ว

- Frame Work

การเป็น Software ที่มี .NET Framework ทำให้มีฟังก์ชั่นในการทำงานที่มีความสมบูรณ์แบบซึ่งสามารถจัดการเรียกใช้การทำงานภายใต้ Common Language Runtime (CLR) ทำให้มีความเสถียนในด้านประสิทธิภาพและความปลอดภัย

- XML

การเป็น Software ที่ถูกพัฒนาขึ้นมา เพื่อให้ใช้งานร่วมกับสถาปัตยกรรมแบบ Distributed interNet Applications (DNA) ในการถ่ายโอนหรือแลกเปลี่ยนข้อมูลในรูปแบบของเอกสาร XML ซึ่งเป็นเพียงตัวอักษรธรรมดาเท่านั้น

- Error handling

ภายใน VB.NET มีการทำงาน ที่ช่วยเหลือในการจัดการข้อผิดพลาดที่เกิดขึ้น โดยมีคำสั่ง Try……..Catch…..Finally เพื่อคอยรับข้อความหรือข้อมูลที่มีข้อผิดพลาดระหว่างการพัฒนาโปรแกรม

- Security

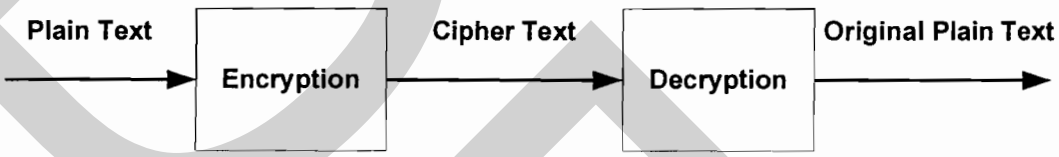
ภายใน VB.NET มีการสนับสนุนในด้านของฟังก์ชั่นทางด้านความปลอดภัยสำหรับระบบเน็ตเวิร์คหรือระบบทางด้าน Cryptography

- Image Processing

ภายใน VB.NET มีฟังก์ชันในการทำงานเกี่ยวกับรูปภาพ เช่น การเข้ารหัส, การบีบอัดข้อมูล, การใช้งานกับค่า pixel ของภาพ เพื่อคอยสนับสนุนงานทางด้านการประมวลผลข้อมูลภาพ

2) วิทยาการเข้ารหัสและถอดรหัส

วิทยาการเข้ารหัสและถอดรหัสนั้น เป็นวิทยาการที่ว่าด้วยการเข้ารหัสลับหรือข้อความที่ถูกเข้ารหัสจากข้อความปรกติให้กลายเป็นข้อความลับ ทำให้บุคคลที่สามไม่สามารถที่จะเข้าใจได้ ซึ่งวิทยาการนี้ถูกใช้มาอย่างช้านาน จนกระทั่งในปัจจุบันได้มีวิทยาการเข้ารหัสลับสมัยใหม่ (Modern Cryptography) โดยอาศัยแนวทางคณิตศาสตร์เพื่อแปลงข้อความปรกติให้เป็นข้อความลับ ซึ่งจะมีแต่เพียงคู่สนทนาเท่านั้นที่สามารถอ่านเข้าใจได้ และวิธีการของการเข้ารหัสลับสมัยใหม่ ได้แก่ Data Encryption Standard, Advance Encryption Standard, One-Time Padding หรือ MD5 (Message-Digest algorithm 5) โดยแสดงดังภาพที่ 1

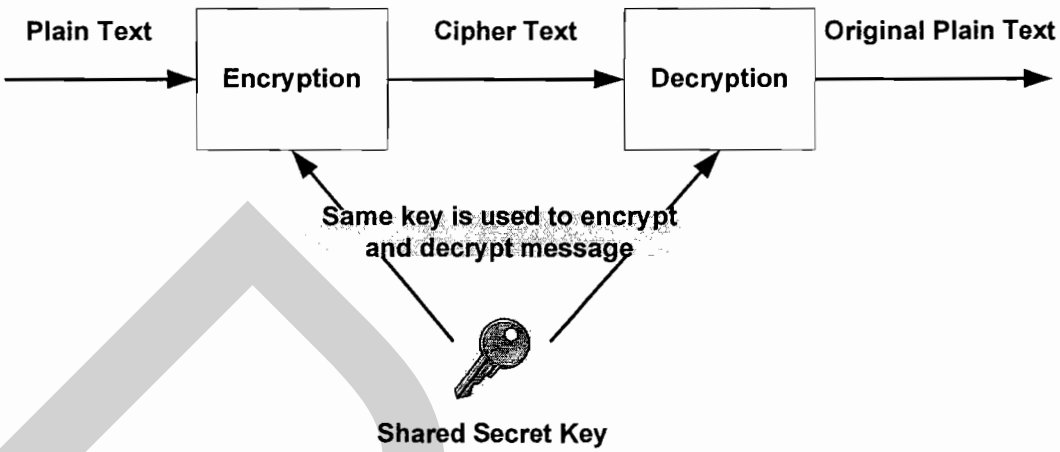


ภาพที่ 1 วิทยาการ การเข้ารหัสข้อมูล

ระบบการเข้ารหัสข้อมูล

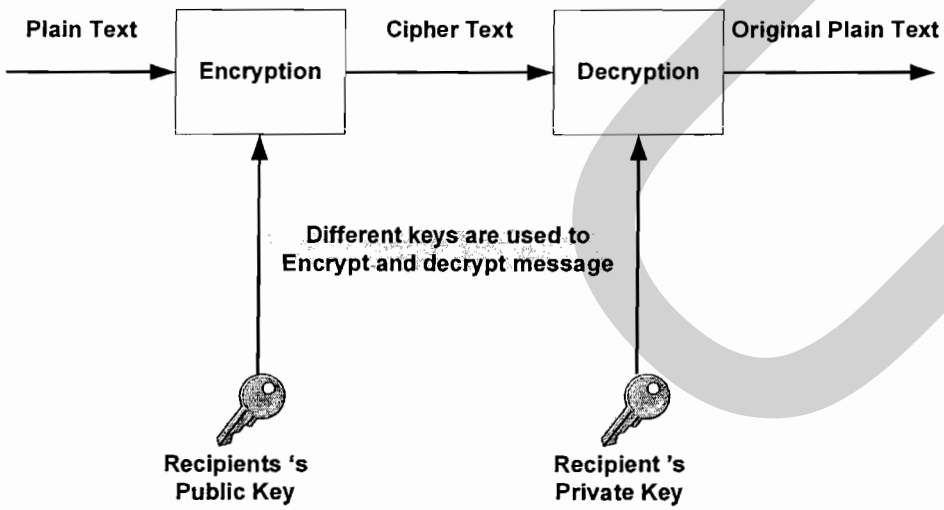
เป็นวิธีการที่ใช้สำหรับการแปรเปลี่ยนรูปแบบของข้อมูลให้อยู่ในรูปแบบที่บุคคลทั่วไปไม่สามารถเข้าใจได้ ซึ่งโดยทั่วไปแล้วการเข้ารหัสข้อมูลนั้นจะทำการจัดเก็บข้อมูลก่อนส่งไปยังปลายทางด้วยรหัสผ่าน (key) ซึ่งอาจจะเป็นตัวเลขที่ได้จากการสุ่มมาผ่านกระบวนการทางคณิตศาสตร์ ทำให้ผลลัพธ์ที่ได้ก็คือข้อมูลที่ถูกเข้ารหัส ซึ่งเป็นขั้นตอนที่เรียกว่า “การเข้ารหัส (Encryption) และเมื่อต้องการอ่านข้อมูลที่ถูกเข้ารหัสนั้น จะต้องนำรหัสผ่านเข้าสู่กระบวนการทางคณิตศาสตร์ ผลลัพธ์ของข้อมูลที่ได้จะเรียกว่า “การถอดรหัส (Decryption)” และระบบการเข้ารหัสนั้น สามารถแบ่งตามวิธีการใช้งานของรหัสผ่านได้ 2 วิธีดังนี้

1. ระบบการเข้ารหัสแบบกุญแจสมมาตร (Symmetric Key Cryptosystem) คือการเข้ารหัสด้วยกุญแจเดี่ยวสำหรับผู้ส่งและผู้รับ โดยวิธีการนี้เป็นวิธีการที่ทั้งสองฝ่ายจะตกลงกันว่าจะใช้รหัสแบบใด โดยแสดงตามภาพที่ 2



ภาพที่ 2 วิธีการเข้ารหัสแบบกุญแจสมมาตร

2. ระบบการเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric Key Cryptosystem) คือการเข้ารหัสโดยใช้หลักการของกุญแจคู่ ประกอบไปด้วย กุญแจส่วนตัว (private key) และกุญแจสาธารณะ (public key) ซึ่งหลักการทำงานจะใช้กุญแบบใดก็ได้ในการเข้ารหัส และการถอดรหัสจะใช้กุญแจอีกชนิดหนึ่ง สำหรับการเข้ารหัสและการถอดรหัสด้วยกุญแจทั้งสองนี้ จะใช้ฟังก์ชั่นทางคณิตศาสตร์เข้ามาช่วยดำเนินการ โดยแสดงตามภาพที่ 3



ภาพที่ 3 วิธีการเข้ารหัสแบบกุญแจอสมมาตร

ประวัติผู้เขียน

ชื่อ-นามสกุล	นาย ธมกร บุ้งจันทร์
ประวัติการศึกษา	สำเร็จการศึกษาระดับปริญญาตรี จากคณะสารสนเทศศาสตร์ มหาวิทยาลัยศรีปทุม ปีการศึกษา 2548
ตำแหน่งและสถานที่ทำงานปัจจุบัน	เจ้าหน้าที่ฝ่ายขาย บริษัท ไอทีสบาย จำกัด