# การอำพรางข้อความในภาพบนโทรศัพท์มือถือ Android

พงศ์ปณต ทรัพย์วิไล

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตร์มหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม คณะวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2556

DPU

# Embedding Message into Picture on an Android Phone

**PONGPANOD SUBVILAI**

**A Thematic Paper 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**

DPU

หัวข้อสารนิพนธ์ การอำพรางข้อความในภาพบนโทรศัพท์มือถือ Android
ชื่อผู้เขียน พงศ์ปณต ทรัพย์วิไล
อาจารย์ที่ปรึกษา ดร.ชัยพร เขมะภาตะพันธ์
สาขาวิชา วิศวกรรมคอมพิวเตอร์และโทรคมนาคม
ปีการศึกษา 2555

**บทคัดย่อ**

งานวิจัยนี้มีวัตถุประสงค์เพื่อการศึกษาการฝังและอำพรางข้อมูลลงไปในสื่อรูปภาพ ซึ่งเรียกว่าวิธีการสเตกาโนกราฟี โดยผลงานที่ได้เป็นแอพพลิเคชั่นที่ทำงานบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ ซึ่งกำลังได้รับความนิยมเป็นอย่างสูงในปัจจุบัน โดยใช้วิธีการสลับบิตร่วมกับมาตรฐานการเข้ารหัสลับขั้นสูง การสลับบิตข้อมูลจะดำเนินการกับ 2 บิตสุดท้ายที่มีนัยสำคัญต่ำสุดของแต่ละพิกเซลสีในไฟล์ภาพ PNG ด้วยบิตข้อมูลที่ต้องการอำพรางลงไปยังรูปภาพ ทำให้สามารถซ่อนข้อความที่ถูกเข้ารหัสลับได้ 6 บิตต่อ 1 พิกเซล เช่นไฟล์ภาพขนาด 256*256 พิกเซล จะสามารถอำพรางข้อความที่ผ่านกระบวนการเข้ารหัสแล้วได้ประมาณ 49 กิโลไบต์ โดยไม่ทำให้รูปภาพเกิดความเปลี่ยนแปลงไปจนสามารถสังเกตเห็นได้ด้วยสายตาของมนุษย์ นอกจากนี้ได้นำวิธีการวัดการเปลี่ยนแปลงของภาพด้วย PSNR และ MSE เพื่อทดสอบและประเมินประสิทธิภาพในการอำพรางข้อความลงไปยังไฟล์ภาพ โดยทดสอบด้วยวิธีการนำข้อความที่มีขนาดที่แตกต่างกันตั้งแต่ 5 กิโลไบต์จนถึง 50 กิโลไบต์อำพรางลงไปยังรูปภาพเดียวกันที่ไฟล์ภาพ PNG ขนาด 256*256 พิกเซล จากผลการวัดพบว่าค่าเฉลี่ย PSNR ที่ได้จะอยู่ที่ประมาณ 46 เดซิเบล และค่าเฉลี่ยของ MSE จะอยู่ที่ประมาณ 1.69 ซึ่งผลการวัดประสิทธิภาพที่ได้อยู่ในเกณฑ์ที่ดี แต่หากนำข้อความที่มีขนาดยาวมากเกินกว่าขนาดที่จะซ่อนได้ ข้อความที่ถูกอำพรางลงไปจะไม่สมบูรณ์และไม่สามารถถอดข้อความลับดังกล่าวออกมาได้

| | |
|---|---|
| Thematic Paper Title | Embedding Message into Picture on an Android Phone |
| Author | Pongpanod Subvilai |
| Thematic Paper Advisor | Dr. Chaiyaporn Khemapatapan |
| Department | Computer and Communication Engineering |
| Academic Year | 2012 |

## ABSTRACT

This research is to study how to embed and hide data into picture media called steganography technique. The output work is an Android application which is currently popular. Bit swapping and AES are the methods used in this application. 2 last bits having least significance of each color of pixel of PNG picture file will be swapped by 2 information bits wanted to embed. Thus, up to 6 bits can be embedded in each pixel. For example, a picture of 256*256 pixels can embed encryption messages about 49 kB without causing any noticeable change for human's eyes. Moreover, PSNR and MSE are used to test and evaluate the change of picture. The testing process is done by embedding messages having sites from 5 kB to 50 kB to embed into PNG picture files of 256*256 pixels. From the testing results, average PSNR is about 46 dB and average MSE is about 1.69. Thus the performance of the system is good. However, when the size of message embed into the picture file is longer than the acceptable size, the embedded message will be incomplete and cannot retrieve to be the original message.

# กิตติกรรมประกาศ

สารนิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ด้วยดี เพราะด้วยความอนุเคราะห์ความช่วยเหลือ อาจารย์ ดร.ชัยพร เขมะภาตะพันธ์ ที่ปรึกษาสารนิพนธ์เล่มนี้ที่ท่านได้เสียสละเวลาอันมีค่าในการให้คำปรึกษา และคำแนะนำต่างๆ ที่มีประโยชน์ต่อการจัดทำสารนิพนธ์ ทั้งยังคอยสนับสนุน และส่งเสริม พร้อมทั้งให้กำลังใจกับผู้วิจัยด้วยดีเสมอมา และขอขอบพระคุณท่านคณะกรรมการสอบสารนิพนธ์ทุกท่าน ที่ได้ให้คำแนะนำแต่สิ่งที่เป็นประโยชน์ให้กับผู้วิจัย เพื่อได้นำไปปรับปรุงแก้ไข จนทำให้เกิดความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณ เพื่อนๆ พี่ๆ น้องๆ รหัส 53 บัณฑิตวิทยาลัย คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิต ที่ได้ให้การช่วยเหลือ สนับสนุน พร้อมทั้งยังเป็นกำลังใจด้วยดีเสมอมา

สุดท้ายขอขอบพระคุณ คุณแม่ ครอบครัว และเพื่อนๆ ที่คอยสนับสนุน และให้กำลังใจ เป็นแรงผลักดันให้ผู้วิจัย มีกำลังใจที่จะก้าวผ่านอุปสรรคต่างๆ ทำให้การจัดทำสารนิพนธ์เล่มนี้ สำเร็จลุล่วงไปด้วยดี

พงศ์ปณต ทรัพย์วิไล

# สารบัญ

# สารบัญ (ต่อ)



DPU

# สารบัญภาพ

**สารบัญภาพ (ต่อ)**

**ภาพที่** **หน้า**

# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญของปัญหา

การติดต่อสื่อสารข้อมูลรูปแบบดิจิตอลในปัจจุบันได้รับความนิยมอย่างแพร่หลาย เนื่องจากเทคโนโลยีที่มีความก้าวหน้ามากขึ้นทำให้การติดต่อสื่อสารทำได้อย่างสะดวกและรวดเร็ว ยกตัวอย่างเช่น การแลกเปลี่ยนข้อมูลโดยใช้อีเมล์ การพูดคุยผ่านทางโทรศัพท์มือถือ หรือการส่งข้อความสั้น (sms) จะเห็นได้ว่าการติดต่อสื่อสารในแต่ละวิธีนั้นต้องการความเป็นส่วนตัว เนื่องจากข้อมูลคือสิ่งที่สำคัญ ซึ่งอาจเป็นข้อมูลส่วนบุคคลหรือข้อมูลที่มีความสำคัญต่อองค์กรที่ไม่ต้องการให้ผู้ใดล่วงรู้นอกจากตัวผู้รับและผู้ส่งเท่านั้น ในบางกรณีข้อมูลดังกล่าวอาจถูกเผยแพร่ออกไปไม่ว่าด้วยวิธีใดก็ตาม และอาจมีผู้ไม่ประสงค์ดีล่วงรู้ถึงข้อมูลที่มีความสำคัญนั้น

การปกป้องข้อมูลที่อยู่ในรูปแบบดิจิตอลให้มีความปลอดภัยมีอยู่หลายวิธี หนึ่งในนั้นคือ การอำพรางข้อมูล เป็นวิธีการที่จะซ่อนข้อมูลที่ต้องการส่งไปในสื่อพาหะ ทำให้ผู้ที่ได้รับข้อมูลไป จะไม่ทราบว่ามีข้อมูลที่สำคัญอยู่ในสื่อนั้น และการรักษาความปลอดภัยของข้อมูลอีกวิธีที่ได้รับความนิยมคือ การเข้ารหัสข้อมูล ซึ่งจะทำการแปลงข้อมูลหรือข้อความที่ต้องการส่งให้กลายเป็นข้อมูลที่ไม่สามารถทำความเข้าใจได้ ทำให้ผู้ที่ไม่เกี่ยวข้องกับการติดต่อสื่อสารและไม่ทราบวิธีในการถอดรหัสข้อมูล จะไม่สามารถเข้าใจถึงความหมายของข้อมูลนั้นเลย ซึ่งในปัจจุบันมีโปรแกรมคอมพิวเตอร์ที่ใช้ในการอำพรางข้อความให้ลงไปอยู่ในรูปภาพหรือโปรแกรมคอมพิวเตอร์ที่ใช้ในการเข้ารหัสข้อมูล ถ้าหากผู้ใช้ต้องการที่จะส่งข้อมูลลับก็สามารถใช้โปรแกรมดังกล่าวในการอำพรางข้อมูลหรือเข้ารหัสลับข้อมูลได้ผ่านเครื่องคอมพิวเตอร์ แต่ในปัจจุบันการติดต่อสื่อสารในรูปแบบดิจิตอลไม่ได้มีเพียงคอมพิวเตอร์ที่สามารถติดต่อสื่อสารหรือแลกเปลี่ยนข้อมูลได้เพียงอุปกรณ์เดียว แต่ยังมีโทรศัพท์มือถือซึ่งเป็นอุปกรณ์ที่ได้รับความนิยมมาก และมีความสามารถในการแลกเปลี่ยนข้อมูลได้หลายรูปแบบ เช่น เสียง ข้อความ ภาพ หรือภาพเคลื่อนไหว อีกทั้งยังเป็นอุปกรณ์ขนาดเล็กที่สามารถพกพาได้ ทำให้การส่งข้อมูลต่างๆ ผ่านทางโทรศัพท์มือถือนั้นสามารถทำได้ทุกที่ทุกเวลา ซึ่งหากผู้ใช้ต้องการส่งข้อความลับไปหาผู้อื่นจะต้องใช้เครื่องคอมพิวเตอร์ในการรับส่งข้อความดังกล่าว อาจทำให้เกิดความไม่สะดวกในการใช้งาน

จากปัญหาข้างต้น ผู้วิจัยจึงได้นำเสนอแอพพลิเคชั่นที่ใช้ในการอำพรางข้อมูลบนโทรศัพท์มือถือร่วมกับมาตรฐานการเข้ารหัสลับขั้นสูงที่ใช้ระบบปฏิบัติการแอนดรอยด์ที่กำลังได้รับความนิยมในปัจจุบัน เพื่อให้เกิดความสะดวกต่อผู้ใช้งานในการปกป้องข้อมูลที่ต้องการติดต่อสื่อสารกัน

**1.2 วัตถุประสงค์ของการวิจัย**

1. ศึกษาวิธีการปกป้องข้อมูลด้วยวิธีการอำพรางข้อมูล เพื่อเป็นแนวทางในการพัฒนาแอพพลิเคชั่นที่ใช้ในการอำพรางข้อมูล
2. ศึกษาวิธีการเข้ารหัสข้อมูลด้วยมาตรฐานการเข้ารหัสขั้นสูง เพื่อนำไปใช้ในการเข้ารหัสข้อมูลก่อนการอำพรางข้อมูล
3. พัฒนาแอพพลิเคชั่นที่ใช้ในการเข้ารหัสและอำพรางข้อมูลตัวอักษรลงในรูปภาพ ผ่านทางโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์

**1.3 ขอบเขตของการวิจัย**

1. วิเคราะห์และศึกษาวิธีการปกป้องข้อมูลด้วยการอำพรางข้อมูล และวิธีการปกป้องข้อมูลด้วยการเข้ารหัสลับด้วยมาตรฐานการเข้ารหัสลับขั้นสูง เพื่อเป็นแนวทางในการพัฒนาแอพพลิเคชั่นที่ใช้ในการเข้ารหัสและอำพรางข้อมูล
2. พัฒนาแอพพลิเคชั่นที่ใช้ในการเข้ารหัสและอำพรางข้อมูลตัวอักษรลงในรูปภาพ และสามารถถอดข้อมูลตัวอักษรออกจากรูปภาพและถอดรหัสข้อมูลดังกล่าวผ่านทางโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์

**1.4 ประโยชน์ที่คาดว่าจะได้รับ**

1. ความปลอดภัยในการรับส่งข้อมูลระหว่างผู้ใช้มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์ โดยใช้เทคนิคการอำพรางข้อมูลร่วมกับการเข้ารหัสลับข้อมูลก่อนที่จะส่ง

**1.5 แผนการดำเนินงาน**

**ตารางที่ 1.1** แผนการดำเนินงาน

| เดือน | 2554 | | | 2555 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. |
| ศึกษาวิธีการอำพรางข้อมูล | | | | | | | | | | | |
| ศึกษาวิธีการเข้ารหัสลับข้อมูล | | | | | | | | | | | |
| วิเคราะห์และศึกษาแนวทางการพัฒนา | | | | | | | | | | | |
| พัฒนาแอพพลิเคชั่นอำพรางข้อมูล | | | | | | | | | | | |
| เปรียบเทียบวิเคราะห์ผลที่ได้และรูปแบบ | | | | | | | | | | | |
| รวบรวมข้อมูลที่ได้จัดทำสารนิพนธ์ | | | | | | | | | | | |

# บทที่ 2
# แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

## 2.1 ระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์เป็นระบบปฏิบัติการที่ทำงานบนลินุกซ์ เคอร์เนล ซึ่งออกแบบมาเพื่ออุปกรณ์สมาร์ทโฟน และแท็บเล็ตที่ใช้ระบบจอสัมผัส แรกเริ่มระบบปฏิบัติการแอนดรอยด์ถูกพัฒนาขึ้นโดยบริษัทแอนดรอยด์ ซึ่งได้รับการสนับสนุนทางการเงินจากบริษัทกูเกิ้ล และในปี ค.ศ. 2005 ก็ถูกซื้อกิจการโดยบริษัทกูเกิ้ล แอนดรอยด์ถูกเปิดเผยในปี ค.ศ. 2007 พร้อมกับการก่อตั้งของ Open Handset Alliance (OHA) ซึ่งเป็นสมาคมทางด้านฮาร์ดแวร์ ซอฟแวร์ และการติดต่อสื่อสารข้อมูล ที่มุ่งเน้นด้านการพัฒนามาตรฐานสำหรับอุปกรณ์โทรศัพท์เคลื่อนที่ โทรศัพท์ที่ใช้ระบบปฏิบัติการแอนดรอยด์เครื่องแรกวางจำหน่ายเมื่อเดือน ตุลาคม ค.ศ. 2008 ซึ่งก็คือ T-Mobile G1 หรือเรียกอีกชื่อว่า เอชทีซี ดรีม โดยแรกเริ่มใช้ระบบปฏิบัติการแอนดรอย์รุ่น 1.1


**ภาพที่ 2.1** T-Mobile G1 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอย์ที่วางจำหน่ายเครื่องแรก

**ที่มา:** http://www.gsmarena.com/t_mobile_g1-2533.php

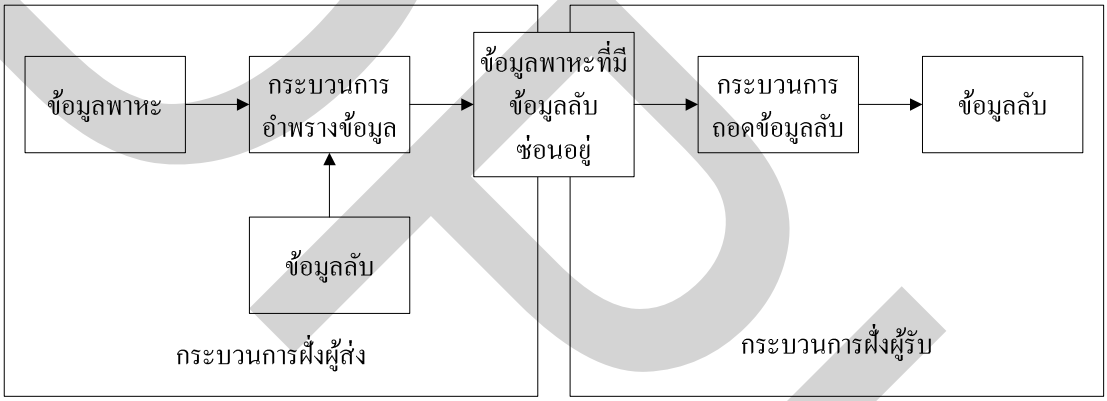
ต่อมาบริษัทผู้ผลิตอุปกรณ์โทรศัพท์เคลื่อนที่หลายๆ รายก็ได้ผลิตอุปกรณ์โทรศัพท์เคลื่อนที่ที่ใช้ระบบปฏิบัติการแอนดรอยด์ออกมามากมายเช่น ซัมซุง โมโตโรล่า โซนี่ หรือเอชทีซี เป็นต้น และในปัจจุบันระบบปฏิบัติการแอนดรอยด์นั้นเป็นที่นิยมอย่างมาก โดยส่วนแบ่งการตลาดของผู้ใช้สมาร์ทโฟนทั่วโลกที่ใช้ระบบปฏิบัติการแอนดรอยด์อยู่ที่ 75 % ในไตรมาสที่ 3 ของปี ค.ศ. 2012 (SARAH PEREZ, 2012, November 2) แอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์นั้นใช้ภาษาจาวาในการพัฒนา และสามารถดาวน์โหลดแอพพลิเคชั่นต่างๆ ได้ผ่านทาง Google Play

**2.2 ทฤษฎีเกี่ยวกับวิทยาการอำพรางข้อมูล (Steganography)**

วิทยาการอำพรางข้อมูล เป็นศาสตร์และศิลป์แห่งการซ่อนข้อเท็จจริงที่ต้องการสื่อสารไว้ในไฟล์ข้อมูลอื่นๆ ซึ่งสามารถใช้ไฟล์ข้อมูลได้หลายรูปแบบในการปกปิดหรืออาจเรียกว่าเป็นไฟล์พาหะ โดยมีวัตถุประสงค์เพื่อทำให้ข้อมูลที่ต้องการติดต่อสื่อสารกันเป็นความลับ ทำให้ข้อมูลปลอดภัยจากบุคคลที่สาม ซึ่งอาจมองว่ามีความคล้ายกับวิทยาการเข้ารหัสลับ (Cryptography) แต่แตกต่างกันตรงที่วิธีการในการปกป้องข้อมูล โดยวิธีการปกป้องข้อมูลของวิทยาการเข้ารหัสลับนั้นมุ่งเน้นที่การรักษาเนื้อหาของข้อความลับโดยใช้เทคนิคการเข้ารหัสและถอดรหัสข้อมูล ทำให้บุคคลที่ไม่พึงประสงค์ที่พบเห็นข้อมูลจะทราบได้ว่าข้อมูลนั้นได้ถูกเข้ารหัสไว้ ทำให้มีความเป็นไปได้ที่บุคคลนั้นจะทำการถอดรหัสเพื่อนำข้อมูลเหล่านั้นไปใช้ประโยชน์ และวิทยาการอำพรางข้อมูลจะใช้วิธีการซ่อนหรือปกปิดข้อมูลที่ต้องการส่งลงไปในไฟล์พาหะ ทำให้การติดต่อสื่อสารข้อมูลนั้นไม่เป็นที่น่าสงสัยทำให้บุคคลอื่นไม่ทันล่วงรู้ว่าเกิดการติดต่อสื่อสารข้อมูลขึ้นแล้ว ซึ่งอาจมองว่าวิธีการที่ใช้ในวิทยาการอำพรางข้อมูลนั้นคล้ายกับการพิมพ์ลายน้ำแบบดิจิตอล (Digital Watermarking) เพียงแต่แตกต่างกันที่จุดประสงค์ในการใช้งาน เพราะการพิมพ์ลายน้ำแบบดิจิตอลโดยมากจะเกี่ยวข้องกับการคุ้มครองทรัพย์สินทางปัญญา โดยในการพิมพ์ลายน้ำแบบดิจิตอลข้อมูลที่ถูกซ่อนอยู่ภายในไฟล์อาจเปิดเผยให้สาธารณะชนรับรู้หรือบางครั้งอาจมองเห็นได้ ในขณะที่วิทยาการอำพรางข้อมูลจะพยายามซ่อนข้อมูลไว้ในไฟล์อื่นเพื่อไม่ให้บุคคลไม่ที่พึงประสงค์รับรู้ถึงการมีอยู่ของข้อมูลนั้น

Steganography มาจากคำในภาษากรีกว่า “steganos” หมายถึง “การปกปิด” และ “graphei” หมายถึงการเขียน จึงสามารถนิยามว่ามันเป็น “การเขียนที่ถูกปกปิด” ในอดีตวิทยาการอำพรางข้อมูลโดยส่วนมากถูกใช้ในการทำสงคราม ตัวอย่างเช่น การจับพลทหารมาโกนศีรษะและทำการสักข้อมูลที่ต้องการอำพรางลงไปบนศีรษะของทหารนายนั้น จากนั้นรอให้ผมของทหารนายนั้นขึ้น แล้วจึงให้ทหารนายนั้นไปส่งข่าวสาร ผู้รับข่าวสารก็จะทำการโกนศีรษะของทหารนายนั้นเพื่อดู

ข้อมูลที่ผู้ส่งต้องการส่งให้ อีกตัวอย่างคือการใช้หมึกล่องหน โดยการใช้น้ำมะนาวหรือน้ำนมเขียนข้อมูลที่ต้องการอำพรางลงไปบนกระดาษ ซึ่งข้อมูลเหล่านี้จะไม่สามารถมองเห็นได้หากไม่นำกระดาษใบนั้นไปอังกับไฟ หรือที่ T.Morkal, J.H.P. Elooff และ M.S.Olivier ศึกษาพบว่าการส่งจดหมายธรรมดาทั่วไป แต่แทรกข้อมูลไมโครดอทซึ่งเป็นเทคนิคที่ถูกพัฒนาโดยชาวเยอรมันเพื่อใช้ในสงครามโลกครั้งที่สอง โดยข้อมูลไม่ว่าจะเป็นตัวอักษรหรือภาพถ่ายจะถูกย่อส่วนจนเล็กมากหรือมีขนาดเท่ากับจุดฟูลสต็อปของเครื่องพิมพ์ดีดทั่วไป ทำให้ยากแก่การตรวจพบมาก จากการอำพรางข้อมูลด้วยวิธีการต่างๆ พอสรุปได้เป็น กระบวนการในการอำพรางและถอดข้อมูล ดังแสดงในภาพที่ 2.2



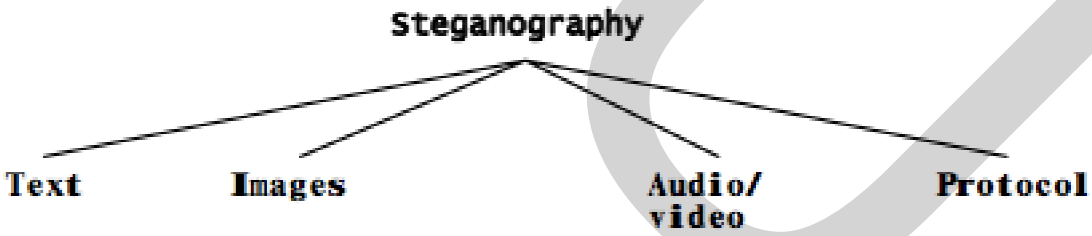

**ภาพที่ 2.2** กระบวนการอำพรางและถอดข้อมูล

ในยุคปัจจุบันการติดต่อสื่อสารข้อมูลส่วนใหญ่จะใช้วิธีการติดต่อสื่อสารข้อมูลในรูปแบบดิจิตอล เนื่องจากความก้าวหน้าทางเทคโนโลยี ทำให้การติดต่อสื่อสารในรูปแบบนี้ ได้รับความนิยมเป็นอย่างมาก และการติดต่อสื่อสารในรูปแบบดิจิตอลนั้นสะดวก รวดเร็ว และประหยัดกว่าการติดต่อสื่อสารด้วยวิธีการในอดีต อย่างเช่นในอดีตจะใช้วิธีการส่งจดหมายเพื่อติดต่อสื่อสารข้อมูลที่เป็นตัวอักษรหรือรูปภาพ การส่งจดหมายนั้นต้องใช้เวลามาก กว่าจะส่งถึงมือผู้รับ อีกทั้งยังมีค่าใช้จ่ายในการรับส่งไม่ว่าจะเป็นค่าอากรแสตมป์ ค่าซองจดหมาย เป็นต้น ซึ่งการส่งจดหมายอิเล็กทรอนิกส์หรืออีเมล์นั้น สามารถส่งข้อความตัวอักษร และไฟล์ต่างๆ ไม่ว่าจะเป็นรูปภาพ เสียง หรือภาพเคลื่อนไหว และสามารถทำได้อย่างสะดวกรวดเร็ว อีกทั้งประหยัดค่าใช้จ่ายในการส่งอีกด้วย ซึ่งทำให้การติดต่อสื่อสารในรูปแบบดิจิตอล ไม่ว่าจะเป็นการส่งจดหมายอิเล็กทรอนิกส์ การพูดคุยผ่านทางโทรศัพท์ การส่งข้อความสั้นทางโทรศัพท์มือถือ หรือการส่งข้อความสื่อประสมผ่านทางโทรศัพท์มือถือ ได้รับความนิยมเป็นอย่างมาก แต่กระนั้นในการติดต่อสื่อสารที่ต้องการความ

เป็นส่วนบุคคลก็อาจมีผู้ไม่ประสงค์ดี ต้องการที่จะนำข้อมูลส่วนบุคคลที่ติดต่อสื่อสารกันนี้ไปใช้ประโยชน์ ไม่ว่าจะเป็นการติดต่อสื่อสารในเรื่องธุรกิจ ข้อมูลทางราชการ หรือเรื่องส่วนตัว ซึ่งอาจส่งผลให้เกิดความเสียหายต่อเจ้าของข้อมูลที่ถูกนำข้อมูลไปไม่ว่าจะนำไปเผยแพร่ ดัดแปลง หรือนำไปใช้ประโยชน์จากข้อมูลที่ได้ไปก็ตาม ทำให้มีความจำเป็นที่จะต้องเพิ่มความปลอดภัยให้กับการรับส่งข้อมูล ซึ่งผู้วิจัยนำเสนอวิทยาการอำพรางข้อมูลเพื่อใช้ในการปกป้องข้อมูลที่มีค่ายิ่งนั้น โดยวิธีการอำพรางข้อมูลที่ต้องการส่งที่อยู่ในรูปแบบดิจิตอลนั้น แทรกเข้าไปในสื่อดิจิตอลอื่นที่ทำหน้าที่เป็นพาหะ เพื่อให้ผู้ไม่ประสงค์ดีไม่สามารถล่วงรู้ถึงการมีอยู่ของข้อมูลลับนั้น ทำให้ผู้ไม่ประสงค์ดีเข้าใจว่าเป็นการติดต่อสื่อสารข้อมูลปกติที่ไม่มีความสำคัญอะไร

## 2.3 ประเภทของวิทยาการอำพรางข้อมูล

รูปแบบไฟล์ดิจิตอลเกือบทุกรูปแบบสามารถใช้สำหรับการอำพรางข้อมูลได้ แต่รูปแบบที่มีความเหมาะสมจะต้องมีระดับความซ้ำซ้อนของข้อมูลสูง ความซ้ำซ้อนของข้อมูลคือบิตของวัตถุที่มีมากเกินความจำเป็นสำหรับการใช้งานและการแสดงผล บิตซ้ำซ้อนคือบิตที่สามารถเปลี่ยนแปลงได้โดยไม่ถูกตรวจพบได้โดยง่าย โดยเฉพาะอย่างยิ่งไฟล์รูปภาพ ซึ่งเป็นรูปแบบไฟล์ที่ได้รับความนิยมอย่างสูง เนื่องจากพบได้มากมายในอินเทอร์เน็ต ในขณะที่การวิจัยยังได้ค้นพบว่ามีรูปแบบไฟล์อื่นๆ ที่สามารถใช้สำหรับการอำพรางข้อมูล ภาพที่ 2.3 แสดงให้เห็นถึงสี่ประเภทหลักของรูปแบบไฟล์ที่สามารถใช้สำหรับการอำพรางข้อมูลได้




**ภาพที่ 2.3** การแบ่งประเภทของการอำพรางข้อมูล จาก “An Overview of Image Steganography.” โดย T.Morkel, J.H.P. Eloff, M.S. Olivier, 2005, Proceedings of the Fifth Annual Information Security South Africa Conference, p. 3.

การอำพรางข้อมูลในข้อความเป็นวิธีการที่ถูกใช้มาอย่างยาวนาน โดยวิธีการที่พบเห็นได้บ่อยๆ คือการซ่อนข้อความลับไว้ในจดหมาย ในทุกๆ n ตัวอักษรของคำตัวอย่างเช่น

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by product, ejecting suets and vegetable oils. (Jonathan Cummins, Patrick Diskin, Samuel Lau and Robert Parlett, 2004, p. 3)

จากข้อความด้านบนหากดูผิวเผินจะเหมือนการสื่อสารข้อมูลปกติทั่วไป แต่แท้ที่จริงแล้วเป็นจดหมายที่สายลับชาวเยอรมันใช้ส่งข่าวความเคลื่อนไหวของสหรัฐอเมริกาในสมัยสงครามโลกครั้งที่สอง โดยหากถอดตัวอักษรตัวที่ 2 ของแต่ละคำมาเรียงกัน จะเกิดเป็นข้อความว่า

"Pershing sails from NY June 1"

(John Pershing เป็นจอมพลผู้นำทัพสหรัฐสมัยสงครามโลกครั้งที่หนึ่ง) ซึ่งมันเป็นเพียงจุดเริ่มต้นของทุกรูปแบบไฟล์ที่ใช้ในการอำพรางข้อมูล แต่การอำพรางข้อมูลลงในข้อความที่อยู่ในรูปแบบไฟล์ดิจิตอลนั้นไม่เป็นที่นิยม เนื่องจากไฟล์ข้อความมีจำนวนข้อมูลซ้ำซ้อนน้อยมาก

การขยายตัวของการใช้ไฟล์รูปภาพแบบดิจิตอล โดยเฉพาะอย่างยิ่งบนอินเทอร์เน็ตทำให้การใช้ไฟล์ภาพดิจิตอลสำหรับนำมาใช้ในการอำพรางข้อมูลได้รับความนิยมเป็นอย่างมาก อีกทั้งไฟล์ภาพดิจิตอลยังมีจำนวนของบิตซ้ำซ้อนที่ใช้ในการแสดงภาพดิจิตอลมาก ซึ่งเป็นผลดีต่อการอำพรางข้อมูล โดยงานวิจัยนี้มุ่งเน้นในการอำพรางข้อมูลในรูปภาพ

การอำพรางข้อมูลในไฟล์เสียง จะใช้เทคนิคที่คล้ายกันกับไฟล์ภาพดิจิตอล แต่สิ่งที่แตกต่างจากเทคนิคของการอำพรางข้อมูลในไฟล์ภาพดิจิตอลคือการกำบังเสียง เพื่อไม่ให้หูของมนุษย์สามารถได้ยินและรับรู้ถึงความผิดปกติได้ ถึงแม้ว่าการอำพรางข้อมูลลงในไฟล์เสียงจะมีศักยภาพเกือบจะเหมือนกับการอำพรางข้อมูลลงในไฟล์ภาพดิจิตอล แต่ด้วยขนาดที่ใหญ่กว่าของไฟล์เสียง ทำให้การอำพรางข้อมูลลงในไฟล์เสียงได้รับความนิยมน้อยกว่าการอำพรางข้อมูลลงในไฟล์ภาพดิจิตอล

การอำพรางข้อมูลลงในโปรโตคอลจะใช้เทคนิคการฝังข้อมูลภายในข้อความและโปรโตคอลควบคุมเครือข่ายที่ใช้ในการส่งผ่านเครือข่าย ในลำดับชั้น OSI รูปแบบเครือข่ายมีช่องทางที่สามารถแอบแฝงข้อมูลลงไปได้ ตัวอย่างเช่นการซ่อนข้อมูลลงไปในส่วนหัวของแพ็คเกจ TCP/IP ในส่วนที่ไม่ถูกใช้งาน เป็นต้น

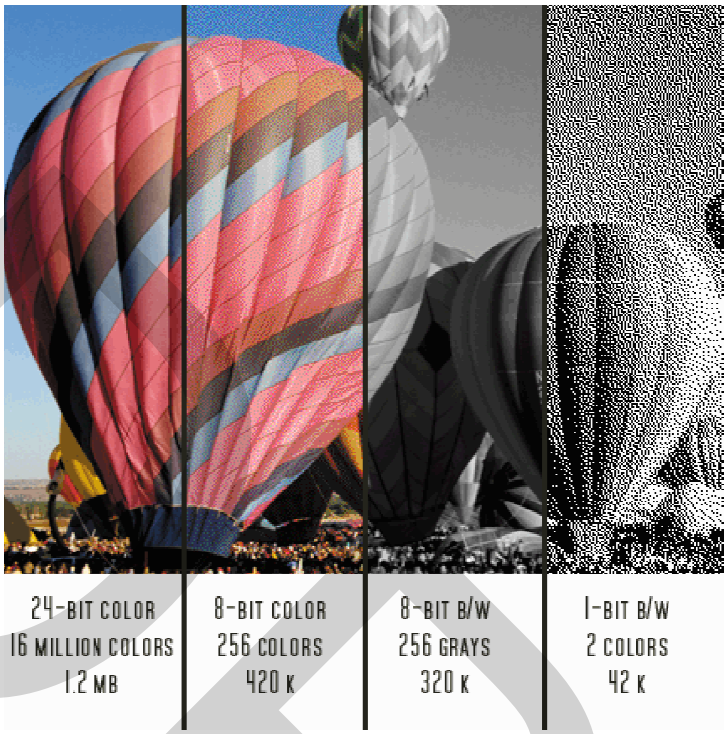
**2.4 การอำพรางข้อมูลลงในไฟล์ภาพ**

จากที่ได้กล่าวไว้ในข้างต้นว่า ไฟล์ภาพดิจิตอลเป็นไฟล์พาหะที่ได้รับความนิยมที่สุดในการใช้สำหรับการอำพรางข้อมูล ซึ่งรูปแบบของไฟล์ภาพดิจิตอลนั้นมีอยู่หลายรูปแบบ โดยแต่ละรูปแบบก็มีขั้นตอนวิธีในการอำพรางข้อมูลที่แตกต่างกันออกไป ในส่วนถัดไปจะขอกล่าวถึงปัจจัยที่มีผลต่อการเลือกขั้นตอนวิธีในการอำพรางข้อมูล

2.4.1 ความคมชัด

ไฟล์รูปภาพดิจิตอลภายในเครื่องคอมพิวเตอร์ มีลักษณะเป็นชุดของตัวเลขที่มีค่าความเข้มของแสงที่แตกต่างกันในแต่ละพื้นที่ของภาพ ซึ่งจะแสดงตัวเลขดังกล่าวออกมาในรูปแบบตารางและแต่ละจุดถูกเรียกว่า “พิกเซล” รูปภาพดิจิตอลที่ใช้บนอินเทอร์เน็ตจำนวนมากมีการระบุพิกเซลของภาพ (แสดงเป็นบิต) ซึ่งในแต่ละพิกเซลจะมีค่าสีของมันอยู่ และจะแสดงพิกเซลออกมาทีละแถวในแนวนอน

จำนวนบิตในรูปแบบสีเรียกว่าความลึกของบิต หมายถึงจำนวนบิตที่ใช้สำหรับแต่ละพิกเซล ความลึกของบิตที่เล็กที่สุดที่ใช้ในรูปแบบสีปัจจุบันคือ 8 ซึ่งหมายถึงมีอยู่ 8 บิตที่ใช้ในการอธิบายสีของแต่ละพิกเซล หรือจะเป็นรูปภาพขาวดำ และภาพเฉดสีเทาก็จะใช้ 8 บิตในแต่ละพิกเซล ซึ่งสามารถแสดงผลออกมาเป็นสีต่างๆ ได้ 256 สีหรือแสดงออกมาเป็นเฉดสีเทา รูปภาพสีดิจิตอลมักจะเก็บไว้ในไฟล์ 24 บิต และใช้รูปแบบสี RGB ที่ให้ความสมจริง




**ภาพที่ 2.4** เปรียบเทียบความแตกต่างระหว่างภาพที่มีระบบภาพแบบต่างๆ

**ที่มา:** http://www.ou.edu/class/digitalmedia/articles/ColorPalettes_Dithering_BitDepth.html

ทุกพิกเซลของภาพ 24 บิต ประกอบด้วยสามสีหลักคือ สีแดง สีเขียว และสีน้ำเงิน ซึ่งแต่ละสีจะมีค่า 8 บิต ดังนั้นในภาพ 24 บิตที่แต่ละพิกเซลประกอบด้วย สีแดง สีเขียว และสีน้ำเงิน จะสามารถแสดงสีต่างๆ ออกมาได้ 16 ล้านสีเลยทีเดียว ซึ่งการที่มีสีที่แสดงผลออกมาได้จำนวนมากนี้ทำให้ไฟล์มีขนาดใหญ่ขึ้นตามไปด้วยจากภาพที่ 2.4 จะสังเกตได้ว่ารูปภาพที่ระบบสียิ่งมากภาพที่ได้จะยิ่งคมชัดสมจริง สังเกตจากภาพที่มีระบบสี 24 บิตจะสามารถแสดงสีต่างๆ ได้ถึง 16,777,216 สี ในขณะที่ภาพที่มีระบบสี 8 บิตจะสามารถแสดงสีต่างๆ ได้เพียง 256 สี แต่สิ่งที่แตกต่างกันอีกอย่างหนึ่งก็คือขนาดของไฟล์ภาพซึ่งเป็นผลมาจากระบบสีของภาพที่มีจำนวนบิตในแต่ละพิกเซลมากขึ้นตามจำนวนของระบบสี

2.4.2 การบีบอัดภาพ

จากการที่ภาพมีความลึกบิตที่มากขึ้น ส่งผลขนาดของภาพใหญ่ขึ้น และมีแนวโน้มว่าขนาดจะใหญ่เกินไปที่จะส่งผ่านอินเทอร์เน็ต เพื่อจะแสดงภาพได้ในเวลาที่เหมาะสม จึงต้องมีการคิดเทคนิคบางอย่างขึ้นเพื่อลดขนาดของไฟล์รูปภาพ ซึ่งเทคนิคเหล่านี้ใช้ประโยชน์จากสูตรทาง

คณิตศาสตร์ในการวิเคราะห์และกลั่นกรองข้อมูลภาพ ทำให้ข้อมูลของภาพมีขนาดไฟล์ที่เล็กลง ซึ่งกระบวนการเหล่านี้เรียกว่าการบีบอัด

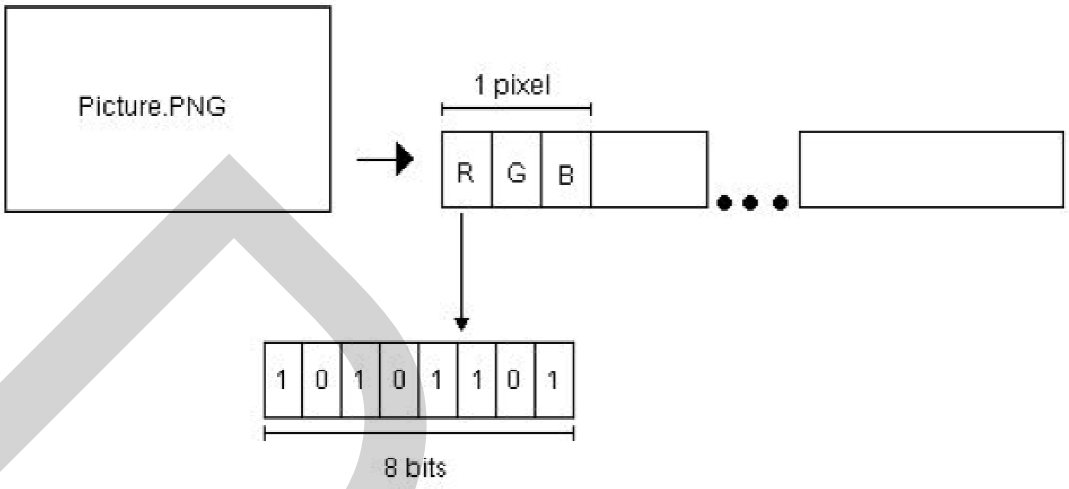
การบีบอัดภาพแบ่งออกเป็นสองประเภทคือ การบีบอัดแบบ lossy และการบีบอัดแบบ lossless โดยทั้งสองวิธีการนี้มีวัตถุประสงค์เพื่อที่จะประหยัดพื้นที่ในการจัดเก็บข้อมูล แต่จะมีวิธีดำเนินการที่แตกต่างกัน โดยการบีบอัดแบบ lossy จะทำให้ไฟล์มีขนาดเล็กลงโดยละทิ้งข้อมูลส่วนเกินออกจากภาพเดิม คือจะเอารายละเอียดที่มีขนาดเล็กเกินกว่าที่มนุษย์จะมองเห็นได้ด้วยตาเปล่าออกไป โดยจะได้ผลลัพธ์ที่ออกมาจะใกล้เคียงกับภาพเดิม ตัวอย่างของรูปแบบภาพที่ใช้เทคนิคการบีบอัดนี้คือ JPEG (Joint Photographic Experts Group)

ส่วนการบีบอัดแบบ lossless จะไม่มีการขจัดข้อมูลใดออกจากภาพเดิมเลย แต่จะใช้การแทนที่ข้อมูลด้วยสูตรทางคณิตศาสตร์ ซึ่งจะทำให้ความสมบูรณ์ของภาพต้นฉบับยังคงอยู่ และได้ผลลัพธ์ของภาพแตกออกมาเป็นแต่ละบิตเหมือนกันกับภาพต้นฉบับ รูปแบบภาพที่นิยมใช้การบีบอัดนี้คือ GIF (Graphic Interchange Format), PNG (Portable Network Graphics) และ 8-bit BMP (a Microsoft Windows bitmap file)

การบีบอัดมีบทบาทสำคัญมากต่อการเลือกใช้ขั้นตอนวิธีในการอำพรางข้อมูล โดยผลลัพธ์ที่ได้จากเทคนิคการบีบอัดแบบ lossy จะให้ไฟล์ภาพที่มีขนาดเล็ก แต่มีความเป็นไปได้ว่าข้อความที่ฝังตัวอยู่อาจจะหายไปส่วนหนึ่งเนื่องจาก เทคนิคการบีบอัดแบบ lossy นี้จะลบข้อมูลส่วนเกินของภาพออกไป ในขณะที่การบีบอัดแบบ lossless จะช่วยเก็บรายละเอียดของภาพดิจิตอลให้เหมือนเดิมโดยไม่มีเกิดการสูญเสียต่อข้อมูลใดๆ แต่การบีบอัดแบบนี้ก็จะให้ผลลัพธ์เป็นไฟล์ภาพที่มีขนาดใหญ่กว่า

**2.5 Least Significant Bit (LSB)**

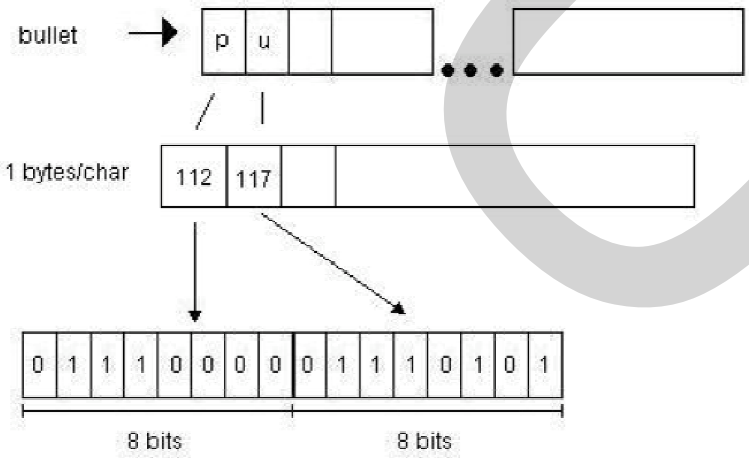
การฝังข้อมูลลงในภาพที่เป็นไฟล์พาหะ โดยแทนที่บิตที่มีนัยสำคัญต่ำสุดของแต่ละไบต์ ของบางส่วนหรือทุกไบต์ของภาพเพื่อเปลี่ยนเป็นข้อความลับที่ต้องการซ่อน เมื่อใช้ระบบภาพแบบ 24 บิต จะสามารถใช้เนื้อที่ในการฝังข้อมูลได้พิกเซลละ 3 บิต เนื่องจากระบบภาพสี 24 บิต ประกอบด้วยสีแดง สีเขียว และสีน้ำเงินในแต่ละพิกเซลจะมี 24 บิต คือสีแดง 8 บิต สีเขียว 8 บิต และสีน้ำเงิน 8 บิต ตัวอย่างเช่นภาพที่มีขนาด 800 x 600 พิกเซล จะสามารถจัดเก็บข้อมูลได้ทั้งหมด 1,440,000 บิต หรือ 180,000 ไบต์




**ภาพที่ 2.5** รายละเอียดของภาพที่มีระบบภาพแบบ 24 บิต

**ที่มา:** http://blog.pupasoft.com/2009/11/09/aeronzsteganography/

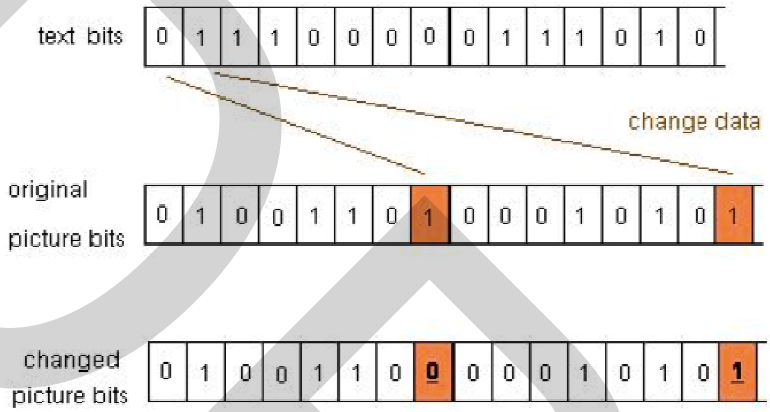
รูปแบบของข้อมูลที่เป็นตัวอักษรจะถูกจัดเก็บเป็นรหัส ASCII ซึ่งในแต่ละตัวอักษรจะมีขนาด 1 ไบต์ (8 บิต) และจะถูกจัดเก็บไว้ในรูปของเลขฐานสอง ดังปรากฏในภาพที่ 2.6




**ภาพที่ 2.6** รูปแบบข้อมูลตัวอักษร ASCII

**ที่มา:** http://blog.pupasoft.com/2009/11/09/aeronzsteganography/

จากนั้นเราจะใช้เทคนิควิธีการอำพรางข้อมูลลงในบิตที่มีนัยสำคัญต่ำสุด หรือก็คือบิตที่มีค่าประจำหลักต่ำสุด นั้นก็คือบิตที่อยู่ทางด้านขวามือสุดของแต่ละพิกเซลสีนั้นเอง โดยจะทำการแตกข้อความที่ต้องการซ่อนออกเป็นไบต์ แล้วทำการแตกรูปภาพที่เราต้องการใช้เป็นไฟล์พาหะออกเป็นไบต์ จากนั้นนำบิตของข้อความไปใส่แทนที่บิตสุดท้ายของไบต์รูปภาพตั้งแต่ไบต์แรก




**ภาพที่ 2.7** การแทนที่บิตสุดท้ายของไฟล์พาหะด้วยบิตของข้อความลับ

**ที่มา:** http://blog.pupasoft.com/2009/11/09/aeronzsteganography/

ซึ่งจากการแทนที่บิตที่มีนัยสำคัญต่ำสุดด้วยบิตของข้อมูลที่ต้องการซ่อน จะทำให้เกิดการเปลี่ยนแปลงของไฟล์พาหะเพียงเล็กน้อย ซึ่งการเปลี่ยนแปลงเหล่านี้ไม่สามารถรับรู้ได้ด้วยตาเปล่า และถึงแม้จะซ่อนข้อความในบิตที่มีนัยสำคัญต่ำสุดสองบิตสุดท้ายของแต่ละพิกเซลสี ก็จะยังคงไม่เห็นความแตกต่างของข้อมูลพาหะ และผลลัพธ์ที่ได้ก็จะไม่ทำให้ขนาดของไฟล์เปลี่ยนไป เนื่องจากเป็นการเข้าไปเปลี่ยนแปลงค่าบิตของพิกเซลสีที่มีอยู่เดิม โดยไม่ได้เพิ่มข้อมูลเกินไปจากส่วนที่มีอยู่

วิธีการอำพรางข้อมูลลงไปในบิตที่มีนัยสำคัญต่ำสุดของแต่ละพิกเซลสี จะสามารถดำเนินการได้ดีเฉพาะกับภาพที่มีการบีบอัดแบบ lossless เท่านั้นเนื่องจากกระบวนการบีบอัดแบบ lossless จะไม่ลบข้อมูลเดิมของภาพออก ในขณะที่การบีบอัดแบบ lossy ในกระบวนการบีบอัดจะทำการตัดข้อมูลที่เกินความจำเป็น หรือส่วนที่ไม่มีผลต่อสายตามนุษย์ออก ซึ่งอาจทำให้เกิดความผิดเพี้ยนของข้อมูลที่ทำการซ่อนลงไปในไฟล์พาหะนั้น ส่วนวิธีการในการซ่อนข้อมูลลงในรูปภาพที่มีการบีบอัดแบบ lossy นั้นมีหลายวิธี ตัวอย่างเช่นการซ่อนข้อมูลไว้ที่ค่าอื่นที่ไม่ใช่ค่าพิกเซลของ

รูปแบบไฟล์ภาพ JPEG ที่ผ่านกระบวนการบีบอัดแบบ lossy ซึ่งสามารถซ่อนข้อมูลไว้ในส่วนหัวของไฟล์ภาพ ซึ่งเป็นส่วนที่จะไม่ถูกนำไปประมวลผลในการสร้างรูป จากกระบวนการดังกล่าว จะได้ไฟล์ที่มีขนาดเปลี่ยนแปลงไปจากเดิม เนื่องจากได้ทำการเพิ่มข้อมูลลงไปในไฟล์นั้นเอง

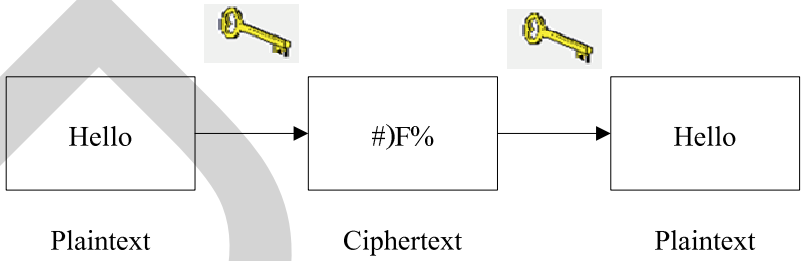
**2.6 ทฤษฎีเกี่ยวกับวิทยาการเข้ารหัสลับ (Cryptography)**

การเข้ารหัสลับข้อมูลเป็นอีกหนึ่งศาสตร์ที่ใช้ในการปกป้องข้อมูลให้เป็นความลับ ซึ่งอาจจะมองว่ามีจุดประสงค์ที่คล้ายคลึงกับวิทยาการอำพรางข้อมูล แต่จะมีเทคนิควิธีการที่ใช้ในการปกป้องข้อมูลที่แตกต่างกัน โดยในวิทยาการเข้ารหัสลับนั้น ผู้ที่ทำการส่งข้อมูลจะทำการแปลงข้อมูลที่ต้องการส่งหรือเรียกว่า เพลนเท็กซ์ (Plaintext) ด้วยขั้นตอนวิธีต่างๆ ผนวกกับกุญแจที่ใช้ในการเข้ารหัสของข้อมูล จนได้ผลลัพธ์ออกมาเป็นข้อมูลที่ไม่สามารถเข้าใจได้หรือข้อมูลที่เข้ารหัสแล้วหรือเรียกว่า ไซเฟอร์เท็กซ์ (Ciphertext) ซึ่งเมื่อผู้รับทำการถอดรหัสข้อมูลด้วยกุญแจที่ใช้ในการถอดรหัสข้อมูลแล้วก็จะได้ข้อมูลที่เป็นเพลนเท็กซ์ออกมาดังเดิม จากวิธีการดังกล่าวหากมีบุคคลที่ไม่พึงประสงค์สามารถดักรับข้อมูลดังกล่าวได้ระหว่างขั้นตอนการสื่อสารข้อมูล บุคคลดังกล่าวก็จะไม่สามารถเข้าใจในข้อมูลที่ผู้รับและผู้ส่งต้องการสื่อสารกันหากไม่มีกุญแจที่ใช้ในการถอดรหัส และไม่ทราบถึงขั้นตอนวิธีที่ใช้ในการถอดรหัส โดยปัจจุบันการเข้ารหัสข้อมูลถูกแบ่งออกเป็น 3 ประเภท คือ การเข้ารหัสลับแบบกุญแจสมมาตร (Secret Key Cryptography) การเข้ารหัสลับแบบกุญแจอสมมาตร (Public Key Cryptography) และแฮชฟังก์ชัน (Hash Function) (จตุชัย แพงจันทร์, 2550, น. 115)

2.6.1 การเข้ารหัสลับแบบกุญแจสมมาตร (Secret Key Cryptography)

การเข้ารหัสลับแบบกุญแจสมมาตรหรือเรียกอีกอย่างหนึ่งว่าการเข้ารหัสแบบกุญแจลับเป็นวิธีการในการเข้ารหัสที่ใช้กุญแจในการเข้ารหัสและถอดรหัสเป็นกุญแจดอกเดียวกัน ดังที่แสดงในภาพที่ 2.8 การเข้ารหัสแบบกุญแจสมมาตรนี้สามารถแบ่งออกได้เป็น 2 ประเภทคือ สตรีมไซเฟอร์ (Stream Ciphers) ซึ่งจะใช้ขั้นตอนวิธีในการเข้ารหัสข้อมูลที่ละไบต์หรือบิต และบล็อกไซเฟอร์ (Block Ciphers) ที่จะเข้ารหัสข้อมูลที่ถูกแบ่งออกเป็นบล็อก ตัวอย่างเช่น ข้อมูลบล็อกละ 56 บิต หรือ 128 บิต เป็นต้น ขั้นตอนวิธีของการเข้ารหัสลับแบบกุญแจสมมาตรนั้นมีอยู่หลายมาตรฐานด้วยกัน ยกตัวอย่างเช่น Data Encryption Standard (DES) ซึ่งเป็นการเข้ารหัสแบบบล็อกไซเฟอร์, Triple-DES หรือ 3DES ซึ่งเป็นตัวที่ถูกพัฒนามาจาก DES โดยจะทำการเข้ารหัสแบบ DES ซ้ำ 3 รอบ ทำให้มีความปลอดภัยมากยิ่งขึ้น หรือ Advance Encryption Standard (AES) ที่ผู้วิจัยจะใช้ในการพัฒนาแอพพลิเคชั่นนี้ ซึ่งจะกล่าวถึงรายละเอียดของขั้นตอนวิธีนี้ในส่วนถัดไป ความปลอดภัยของการเข้ารหัสแบบกุญแจสมมาตรนี้ไม่ได้ขึ้นอยู่กับขั้นตอนวิธีที่ใช้ในการเข้ารหัสและถอดรหัส

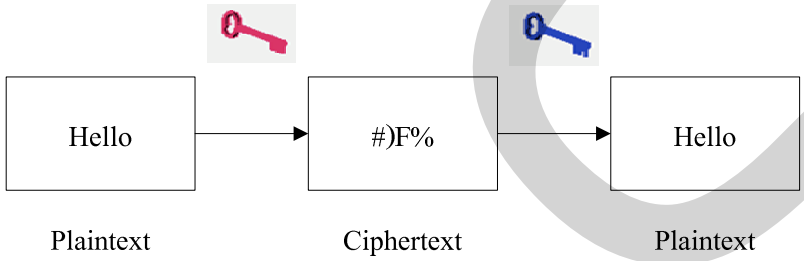
ข้อมูลเพียงอย่างเดียว แต่ส่วนใหญ่จะขึ้นอยู่กับการเก็บรักษาความลับของกุญแจที่ใช้ในการเข้ารหัสและถอดรหัสข้อมูล




**ภาพที่ 2.8** การเข้ารหัสข้อมูลแบบกุญแจสมมาตร

2.6.2 การเข้ารหัสลับแบบกุญแจอสมมาตร (Public Key Cryptography)

การเข้ารหัสลับแบบกุญแจอสมมาตรหรือเรียกอีกอย่างหนึ่งว่าการเข้ารหัสแบบกุญแจสาธารณะ วิธีการเข้ารหัสแบบกุญแจอสมมาตรนี้จะใช้กุญแจ 2 ดอกที่ไม่เหมือนกันในกระบวนการเข้ารหัสและถอดรหัสข้อมูล กุญแจดอกแรกเรียกว่ากุญแจสาธารณะ (Public Key) ส่วนกุญแจอีกดอกเรียกว่ากุญแจส่วนตัว (Private Key) เราจะใช้กุญแจใดก็ได้ในการเข้ารหัสลับข้อมูล แต่จะต้องใช้กุญแจอีกดอกหนึ่งที่ไม่ได้ใช้ในการเข้ารหัสข้อมูลมาเป็นตัวถอดรหัสข้อมูล ดังแสดงในภาพที่ 2.9




**ภาพที่ 2.9** การเข้ารหัสข้อมูลแบบกุญแจอสมมาตร

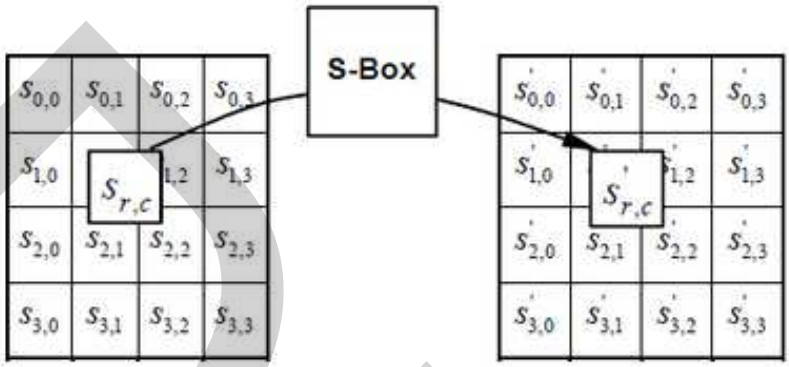
2.6.3 แฮชฟังก์ชัน (Hash Function)

แฮชฟังก์ชันเป็นกระบวนการเข้ารหัสข้อมูลโดยไม่ต้องใช้กุญแจ ซึ่งจะทำการแปลงข้อมูลเดิมให้กลายเป็นข้อมูลที่ผ่านกระบวนการย่อยแล้ว หรือเรียกว่าเมสเสจไดเจสต์ ที่จะใช้ในการตรวจสอบความคงสภาพของข้อมูลนั้น ว่าไม่ได้ถูกเปลี่ยนแปลงไประหว่างการสื่อสารข้อมูล

**2.7 Advance Encryption Standard (AES)**

มาตรฐานการเข้ารหัสลับขั้นสูง หรือ Rijndael (อ่านออกเสียงว่า เรนดอล) เป็นขั้นตอนวิธีในการเข้ารหัสลับข้อมูลแบบกุญแจสมมาตรที่ถูกออกแบบโดยสองนักออกแบบการเข้ารหัสชาวเบลเยี่ยม คือ โจแอน เดเมน (Joan Daemen) และวินเซนต์ ริจเมน (Vincent Rijmen) โดยชื่อของการเข้ารหัสมาจากการนำพยางค์แรกของนามสกุลของผู้คิดค้นมาต่อกัน ซึ่งขั้นตอนวิธีดังกล่าวได้ชนะเลิศในการประกวดการแข่งขันออกแบบขั้นตอนวิธีในการเข้ารหัสลับ และได้รับคัดเลือกโดยสถาบันกำหนดมาตรฐานและเทคโนโลยีแห่งชาติของประเทศสหรัฐอเมริกา (The National Institute of Standards and Technology – NIST) ให้นำมาใช้แทนมาตรฐานการเข้ารหัสลับแบบ DES ซึ่งเป็นมาตรฐานเดิมที่ไม่มีความปลอดภัยเพียงพอ ขั้นตอนวิธีนี้ใช้กระบวนการแบบบล็อกไซเฟอร์ ซึ่งตามมาตรฐานจะใช้การเข้ารหัสข้อมูลที่เป็นกลุ่มๆ ละ 128 บิต (ธำรงรัตน์ อมรรักษา, 2551 : 69) และกุญแจที่ใช้ในการเข้ารหัสและถอดรหัสมีขนาด 128 บิต, 192 บิต และ 256 บิต ให้ได้เลือกใช้ ขึ้นอยู่กับความเหมาะสมในการใช้งาน ซึ่งหากใช้กุญแจที่มีขนาดใหญ่ก็จะทำให้ข้อมูลปลอดภัยและยากต่อการถอดรหัสมากขึ้น แต่ก็ต้องแลกมาด้วยการเข้ารหัสที่นานขึ้นและใช้ทรัพยากรมากขึ้นตามไปด้วย เทคนิคที่ใช้ในการเข้ารหัสและถอดรหัสลับของ AES คือ การดำเนินการผสม (Mixing Operations) ระหว่างกลุ่มข้อมูลในเชิงพีชคณิต โดยกลุ่มบิตย่อยที่ใช้ในการประมวลผลจะถูกจัดให้อยู่ในรูปของแถวลำดับอาเรย์ 2 มิติ ขนาด 4x4 ไบต์ หรือเรียกว่า สเตท (State) ซึ่งขนาดของกุญแจที่เลือกจะเป็นตัวกำหนดจำนวนรอบในการประมวลผล โดยกุญแจขนาด 128 บิต จะทำการประมวลผล 10 รอบ, กุญแจขนาด 192 บิต จะทำการประมวลผล 12 รอบ และกุญแจขนาด 256 บิต จะทำการประมวลผล 14 รอบ ในแต่ละรอบของการประมวลผลจะประกอบไปด้วยกระบวนการย่อย 4 กระบวนการคือ SubBytes, ShiftRows, MixColumns และ AddRoundKey

2.7.1 The SubBytes Transformation




**ภาพที่ 2.10** กระบวนการย่อย SubBytes

**ที่มา:** http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/siaa/se4c03_aes_wiki(7).html

กระบวนการย่อย SubBytes คือกระบวนการที่จะเปลี่ยนแปลงค่าในแต่ละไบต์ของข้อมูลภายในสเตทด้วยการนำข้อมูล 4 บิตแรกในไบต์นั้น มาเป็นตัวกำหนดตำแหน่งเชิงตัวเลขฐานสิบหกของแต่ละแถวใน S-box และ 4 บิตหลังใช้เป็นตัวกำหนดตำแหน่งเชิงเลขฐานสิบหกของแต่ละคอลัมน์ใน S-box

| | | Y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| X | 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| | 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| | 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| | 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| | 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| | 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| | 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| | 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| | 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| | 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| | A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| | B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| | C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| | D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| | E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| | F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

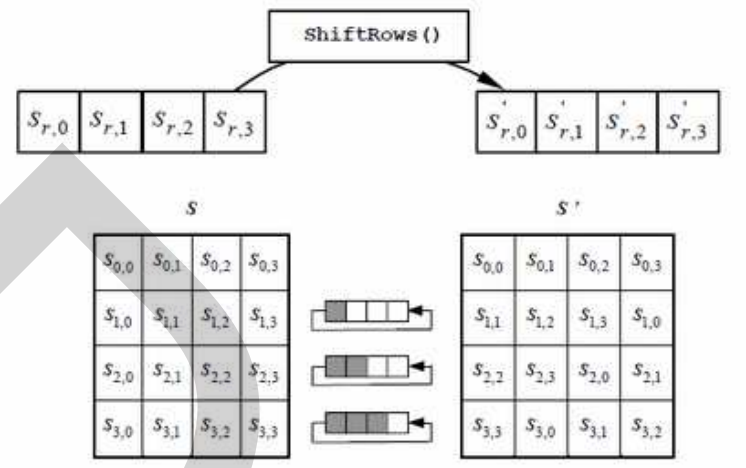
**ภาพที่ 2.11** การแทนที่ S-Box ในขั้นตอนวิธี AES

**ที่มา:** ธำรงรัตน์ อมรรักษา (2551, น. 71)

ตัวอย่างเช่น ถ้ากลุ่มบิตนำเข้า $S_{1,1}$ มีค่าเท่ากับ $EA_{16}$ ตำแหน่งของกลุ่มบิตส่งออกที่ได้จะอยู่ที่แถวที่ $E_{16}$ คอลัมน์ที่ $A_{16}$ ซึ่งก็คือ $87_{16}$ หรือถ้า $S_{3,2}$ มีค่าเท่ากับ $10_{16}$ ผลลัพธ์ที่ได้ผ่านกระบวนการย่อย SubBytes ก็คือ $CA_{16}$ นั่นเอง

2.7.2 ShiftRows Transformation

กระบวนการย่อย ShiftRows จะทำการเลื่อนไบต์แบบวนรอบไปข้างหน้า โดยจะเลื่อนไบต์ในแถวที่ 2, 3 และ 4 ไป 1, 2 และ 3 ตามลำดับ ดังที่แสดงในภาพที่ 2.12




**ภาพที่ 2.12** กระบวนการย่อย Shift Rows

**ที่มา:** http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/siaa/se4c03_aes_wiki(7).html

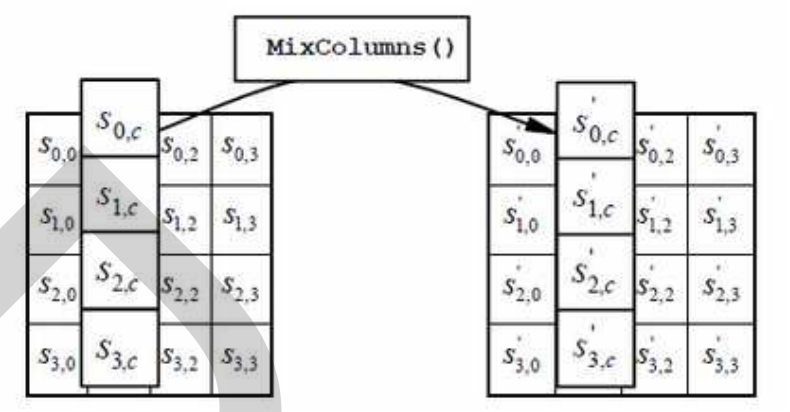
2.7.3 The MixColumns Transformation

กระบวนการย่อย MixColumns จะทำการคูณระหว่างคอลัมน์ภายในสเตทแต่ละคอลัมน์กับพหุนามคงที่ $a(x) = 03_{16}x^3 + 01_{16}x^2 + 01_{16}x + 02_{16}$ เขียนให้อยู่ในรูปของสมการเมทริกซ์ได้ดังนี้

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \quad for 0 \;\; C \;\; Nb \qquad \textbf{(2.1)}$$

และผลลัพธ์ของการคูณสามารถเขียนให้อยู่ในรูปของสมการได้ดังนี้

$$\begin{aligned} S'_{0,C} &= (02_{16} \cdot S_{0,C}) \oplus (03_{16} \cdot S_{1,C}) \oplus S_{2,C} \oplus S_{3,C} \\ S'_{1,C} &= S_{0,C} \oplus (02_{16} \cdot S_{1,C}) \oplus (03_{16} \cdot S_{2,C}) \oplus S_{3,C} \\ S'_{2,C} &= S_{0,C} \oplus S_{1,C} \oplus (02_{16} \cdot S_{2,C}) \oplus (03_{16} \cdot S_{3,C}) \\ S'_{3,C} &= (03_{16} \cdot S_{0,C}) \oplus S_{1,C} \oplus S_{2,C} \oplus (02_{16} \cdot S_{3,C}) \end{aligned} \qquad \textbf{(2.2)}$$

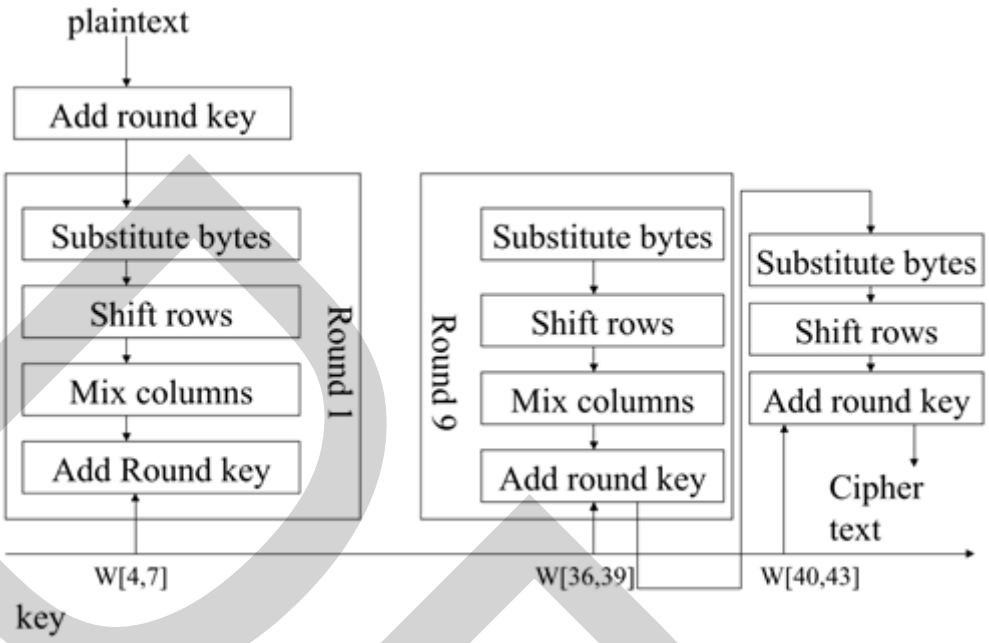



**ภาพที่ 2.13** กระบวนการย่อย MixColumns

**ที่มา:** http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/siaa/se4c03_aes_wiki(7).html

2.7.4 AddRoundKey Transformation

กุญแจของ AES นั้นมีขนาด 128 บิต, 192 บิต และ 256 บิตตามที่กล่าวไปในตอนต้น ซึ่งกุญแจเหล่านี้จะใช้เป็นตัวแปรในการหากุญแจที่จะใช้ในการเข้ารหัสในแต่ละรอบ โดยกุญแจเหล่านี้จะถูกเรียกว่า ราวด์คีย์ (Round Key) ที่มีขนาดเท่ากับขนาดของบล็อก เช่นในสเตทที่มีขนาด 128 บิตจะมีเวิร์ด Nb ขนาด 32 บิต และใช้ตัวอักษรย่อว่า W ตามมาตรฐานของ AES กำหนดให้มีการขยายคีย์เพื่อใช้ในการเข้ารหัสแต่ละรอบซึ่งคีย์ที่ถูกขยายแล้วจะเรียกว่า เอ็กซ์แพนด์คีย์ (Expanded Key) คีย์ต่างๆ สามารถขยายขนาดเพื่อเข้ารหัสในแต่ละรอบ ยกตัวอย่างเช่น คีย์ที่มีขนาด 128 บิต จะทำการเข้ารหัสทั้งหมด 10 รอบ โดยสามารถขยายขนาดคีย์ได้เป็น 1,408 บิต หรือ 176 ไบต์ หรือ 44 เวิร์ด (1 เวิร์ด คือ 1 แถวของอะเรย์ภายในแต่ละสเตท) จากนั้นจะนำคีย์ที่ได้ไปออร์เฉพาะระหว่างสเตทของกลุ่มข้อมูลกับสเตทของกลุ่มกุญแจย่อย



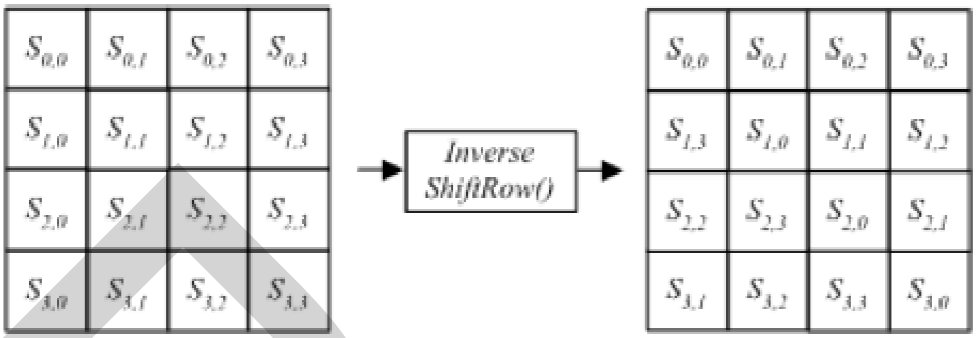

**ภาพที่ 2.14** กระบวนการเข้ารหัสด้วยขั้นตอนวิธี AES แบบใช้กุญแจขนาด 128 บิต

**ที่มา:** ธำรงรัตน์ อมรรักษา (2551, น. 71)

จากภาพที่ 2.14 จะอธิบายถึงขั้นตอนของการเข้ารหัสลับแบบ AES โดยใช้กุญแจที่มีขนาด 128 บิต และจะจัดเรียงกลุ่มข้อมูลขนาด 128 บิตให้อยู่ในรูปของสเตท จากนั้นจะผ่านกระบวนการย่อย AddRoundKey ก่อนในครั้งแรก ผลลัพธ์ที่ได้จะเริ่มเข้าสู่กระบวนการประมวลผลจำนวน 10 รอบ ซึ่งในแต่ละรอบจะประมวลผลทั้ง 4 กระบวนการย่อยที่กล่าวมาตามลำดับ และจะทำการประมวลผลทั้งหมด 10 รอบ แต่ในรอบสุดท้ายของการประมวลผล จะตัดกระบวนการย่อย MixColumns ออก เมื่อผ่านกระบวนการครบแล้วก็จะได้ข้อความรหัส (Ciphertext)

ส่วนในขั้นตอนการถอดรหัสลับของขั้นตอนวิธีแบบ AES จะดำเนินการย้อนกลับกับขั้นตอนการเข้ารหัสลับ กระบวนการย่อยต่างๆ ที่ใช้ในการเจ้ารหัสลับจะเปลี่ยนเป็นกระบวนการย่อยแบบผกผันแทน ซึ่งแต่ละรอบจะประกอบไปด้วยกระบวนการย่อย 4 กระบวนการคือ

2.7.5 กระบวนการย่อย InverseShiftRows ที่จะทำการเลื่อนไบต์แบบวนรอบย้อนหลัง โดยจะเลื่อนไบต์ในแถวที่ 2, 3 และ 4 ไป 1, 2 และ 3 ตามลำดับ ดังที่แสดงในภาพที่ 2.15




**ภาพที่ 2.15** กระบวนการย่อย InverseShiftRows

**ที่มา**: ธำรงรัตน์ อมรรักษา (2551, น. 71)

2.7.6 กระบวนการย่อย InverseSubBytes จะทำการแทนที่ไบต์ข้อมูลในสเตทโดยอ้างอิงตำแหน่งไปยัง S-box ผกผัน (Inverse S-box) ที่กำหนดดังที่แสดงในภาพที่ 2.16

| | | Y | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| X | 0 | 52 | 09 | 6A | D5 | 30 | 36 | A5 | 38 | BF | 40 | A3 | 9E | 81 | F3 | D7 | FB |
| | 1 | 7C | E3 | 39 | 82 | 9B | 2F | FF | 87 | 34 | 8E | 43 | 44 | C4 | DE | E9 | CB |
| | 2 | 54 | 7B | 94 | 32 | A6 | C2 | 23 | 3D | EE | 4C | 95 | 0B | 42 | FA | C3 | 4E |
| | 3 | 08 | 2E | A1 | 66 | 28 | D9 | 24 | B2 | 76 | 5B | A2 | 49 | 6D | 8B | D1 | 25 |
| | 4 | 72 | F8 | F6 | 64 | 86 | 68 | 98 | 16 | D4 | A4 | 5C | CC | 5D | 65 | B6 | 92 |
| | 5 | 6C | 70 | 48 | 50 | FD | ED | B9 | DA | 5E | 15 | 46 | 57 | A7 | 8D | 9D | 84 |
| | 6 | 90 | D8 | AB | 00 | 8C | BC | D3 | 0A | F7 | E4 | 58 | 05 | B8 | B3 | 45 | 06 |
| | 7 | D0 | 2C | 1E | 8F | CA | 3F | 0F | 02 | C1 | AF | BD | 03 | 01 | 13 | 8A | 6B |
| | 8 | 3A | 91 | 11 | 41 | 4F | 67 | DC | EA | 97 | F2 | CF | CE | F0 | B4 | E6 | 73 |
| | 9 | 96 | AC | 74 | 22 | E7 | AD | 35 | 85 | E2 | F9 | 37 | E8 | 1C | 75 | DF | 6E |
| | A | 47 | F1 | 1A | 71 | 1D | 29 | C5 | 89 | 6F | B7 | 62 | 0E | AA | 18 | BE | 1B |
| | B | FC | 56 | 3E | 4B | C6 | D2 | 79 | 20 | 9A | DB | C0 | FE | 78 | CD | 5A | F4 |
| | C | 1F | DD | A8 | 33 | 88 | 07 | C7 | 31 | B1 | 12 | 10 | 59 | 27 | 80 | EC | 5F |
| | D | 60 | 51 | 7F | A9 | 19 | B5 | 4A | 0D | 2D | E5 | 7A | 9F | 93 | C9 | 9C | EF |
| | E | A0 | E0 | 3B | 4D | AE | 2A | F5 | B0 | C8 | EB | BB | 3C | 83 | 53 | 99 | 61 |
| | F | 17 | 2B | 04 | 7E | BA | 77 | D6 | 26 | E1 | 69 | 14 | 63 | 55 | 21 | 0C | 7D |

**ภาพที่ 2.16** การแทนที่ S-box ผกผัน

**ที่มา:** ธำรงรัตน์ อมรรักษา (2551, น. 71)

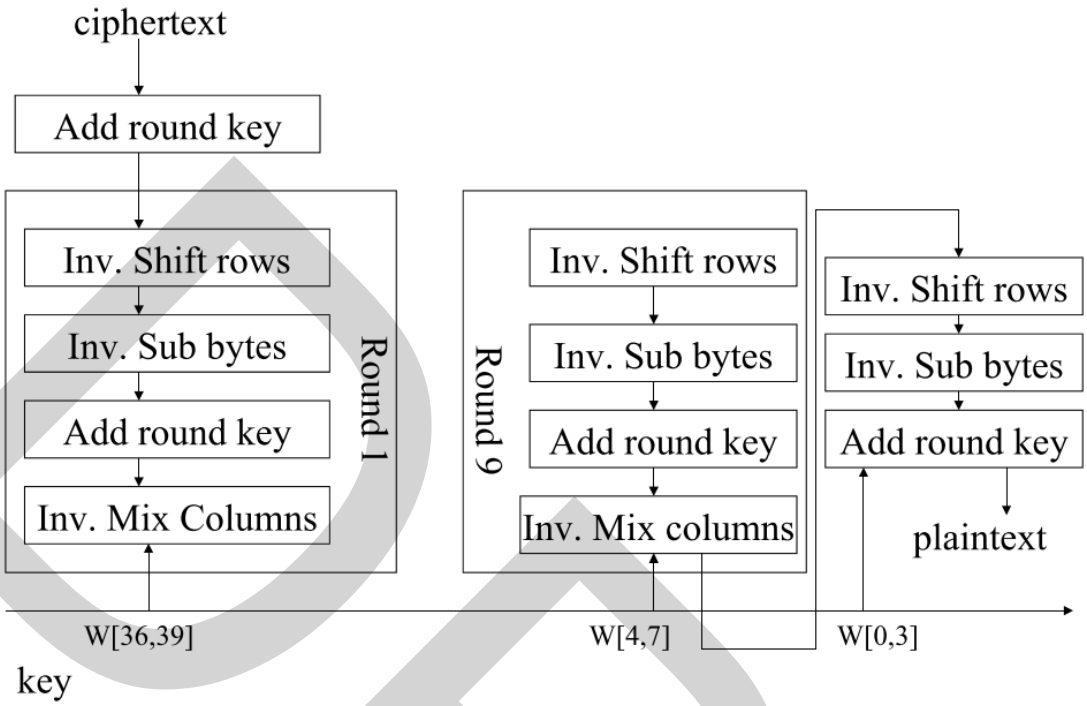
กระบวนการย่อย InverseMixColumns จะทำการคูณระหว่างคอลัมน์ภายในสเตทแต่ละคอลัมน์กับพหุนามคงที่ $a^{-1}(x) = 0B_{16}x^3 + 0D_{16}x^2 + 09_{16}x + 0E_{16}$ สามารถเขียนให้อยู่ในรูปของสมการการคูณเมทริกซ์ได้ดังนี้

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \quad for 0 \;\; C \;\; Nb \qquad \textbf{(2.3)}$$

และผลลัพธ์ของการคูณสามารถเขียนให้อยู่ในรูปของสมการได้ดังนี้

$$\begin{aligned} S'_{0,C} &= (0E_{16} \cdot S_{0,C}) \oplus (0B_{16} \cdot S_{1,C}) \oplus (0D_{16} \cdot S_{2,C}) \oplus (09_{16} \cdot S_{3,C}) \\ S'_{1,C} &= (09_{16} \cdot S_{0,C}) \oplus (0E_{16} \cdot S_{1,C}) \oplus (0B_{16} \cdot S_{2,C}) \oplus (0D_{16} \cdot S_{3,C}) \\ S'_{2,C} &= (0D_{16} \cdot S_{0,C}) \oplus (09_{16} \cdot S_{1,C}) \oplus (0E_{16} \cdot S_{2,C}) \oplus (0B_{16} \cdot S_{3,C}) \\ S'_{3,C} &= (0B_{16} \cdot S_{0,C}) \oplus (0D_{16} \cdot S_{1,C}) \oplus (09_{16} \cdot S_{2,C}) \oplus (0E_{16} \cdot S_{3,C}) \end{aligned} \qquad \textbf{(2.4)}$$

2.7.7 กระบวนการย่อย AddRoundKey ในขั้นตอนของการถอดรหัสข้อมูลจะทำเช่นเดียวกันกระบวนการเข้ารหัสข้อมูลคือการออร์เฉพาะระหว่างคีย์ที่ขยายกับข้อมูลในสเตท แตกต่างกันที่จะทำการออร์เฉพาะแบบย้อนกลับ




**ภาพที่ 2.17** กระบวนการถอดรหัสด้วยขั้นตอนวิธี AES แบบใช้กุญแจขนาด 128 บิต

**ที่มา**: ธำรงรัตน์ อมรรักษา (2551, น. 71)

กระบวนการถอดรหัสจะคล้ายกับกระบวนการเข้ารหัสแต่ทำกระบวนการย้อนกลับ โดยจะจัดเรียงข้อความรหัสให้อยู่ในรูปของสเตทแล้วขยายขนาดของคีย์แล้วทำการ AddRoundKey จากนั้นจะผ่านกระบวนการย่อย 4 กระบวนการตามลำดับ 10 รอบ และตัดกระบวนการ MixColumns ออกในรอบสุดท้ายของการประมวลผล

## 2.8 การวัดประสิทธิภาพของการอำพรางข้อมูล

องค์ประกอบที่สำคัญของการอำพรางข้อความลงไปยังรูปภาพคือ ความแนบเนียน, ปริมาณ และความทนทาน (ธมกร บุ้งจันทร์ 2554 : 1093) ซึ่งการจะวัดประสิทธิภาพด้านความแนบเนียนนั้นสามารถวัดได้ด้วยตาเปล่าว่ารูปภาพที่ได้ถูกอำพรางข้อมูลลงไปกับรูปภาพต้นฉบับนั้นมีความแตกต่างกันมากน้อยเพียงใด และยังมีขั้นตอนวิธีที่สามารถวัดความแตกต่างของภาพซึ่งเรียกว่า PSNR (Peak Signal-to-Noise Ratio) ซึ่งเป็นค่ามาตรฐานที่นักวิจัยทั่วไปนำมาใช้เพื่อเปรียบเทียบคุณภาพของรูปภาพดิจิตอลที่ผ่านกระบวนการประมวลผลทางสัญญาณใดๆ ค่า PSNR

ที่สูงจะชี้ให้เห็นถึงคุณภาพของรูปที่ใกล้เคียงกับรูปภาพต้นฉบับ ค่า PSNR นี้ได้ถูกนำมาใช้ในการประเมินคุณภาพของรูปภาพที่ผ่านการอำพรางข้อมูลมาแล้ว โดยกำหนดให้ OPixel(i,j) คือจุดภาพที่ตำแหน่ง (x, y) ในรูปภาพดิจิตอลที่เป็นภาพต้นฉบับซึ่งมีขนาดเท่ากับ NxN จุดภาพ และ WPixel(i,j) คือจุดภาพที่ถูกฝังข้อความ ค่า PSNR สามารถคำนวณหาได้จากสมการดังนี้

$$MSE = \frac{\sum[OPixel(i,j) - WPixel(i,j)]^2}{N^2} \quad \textbf{(2.5)}$$

$$PSNR = 20\log 10\left(\frac{255}{RMSE}\right) \quad \textbf{(2.6)}$$

โดยที่ค่า Mean Square Error (MSE) หรือค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองจะมาจากการคำนวณในทุกๆ ตำแหน่งของจุดภาพภายในรูปภาพ และค่า RMSE (Root Mean Square Error) คือค่ารากที่สองของ MSE

ซึ่งคุณภาพของรูปภาพ หลังจากผ่านกระบวนการอำพรางข้อความที่ดีนั้นจะมีค่า PSNR ที่สูง โดยทั่วไปแล้วค่า PSNR ที่ยอมรับได้สำหรับการนำรูปภาพไปใช้งานในทางปฏิบัติควรจะอยู่ระหว่าง 20-40 dB

## 2.9 ผลงานวิจัยที่เกี่ยวข้อง

ผู้วิจัยได้ทำการศึกษาเกี่ยวกับรูปแบบของไฟล์ภาพที่จะนำมาใช้ในการอำพรางข้อมูล และให้ความสนใจเกี่ยวกับขั้นตอนวิธีการอำพรางข้อมูลแบบแทนที่ข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุด งานวิจัยที่เกี่ยวข้องกับวิทยาการอำพรางข้อมูล การอำพรางข้อมูลไปยังไฟล์พาหะรูปแบบต่างๆ และขั้นตอนวิธีของการอำพรางข้อมูลที่ใช้กับไฟล์ดิจิตอลรูปแบบต่างๆ ซึ่งสามารถนำมาใช้ในการวิเคราะห์และศึกษาได้เป็นอย่างดี

T.Morkel et al. (2005) ได้นำเสนอภาพรวมของ Steganography หรือการอำพรางข้อมูลโดยใช้รูปเป็นพาหะ เพื่อความปลอดภัยในการรับส่งข้อมูล โดยกล่าวถึงประวัติความเป็นมาของวิทยาการอำพรางข้อมูล วิธีการในการอำพรางข้อมูลในอดีต วัตถุประสงค์ของการใช้วิทยาอำพรางข้อมูลในอดีต ภาพรวมของวิทยาการอำพรางข้อมูลในปัจจุบัน ซึ่งเปลี่ยนไปอยู่ในรูปของไฟล์ข้อมูลดิจิตอลที่ถูกจัดเก็บอยู่บนเครื่องคอมพิวเตอร์ และแลกเปลี่ยนกันอยู่บนอินเทอร์เน็ต ความแตกต่างระหว่างไฟล์ข้อมูลประเภทต่างๆ ที่สามารถใช้ได้ในการอำพรางข้อมูล ซึ่งประกอบไปด้วย ไฟล์ข้อความที่ใช้กันมาตั้งแต่ในอดีต ซึ่งเป็นจุดเริ่มต้นของวิทยาการอำพรางข้อมูลในปัจจุบัน รูปแบบ

ไฟล์รูปภาพ ซึ่งเป็นรูปแบบไฟล์ที่ได้รับความนิยมมากที่สุดสำหรับใช้ในการอำพรางข้อมูล รูปแบบไฟล์เสียงและขั้นตอนวิธีในการอำพรางข้อมูลที่แตกต่างจากขั้นตอนวิธีที่ใช้ในการอำพรางข้อมูลลงในไฟล์รูปภาพ รูปแบบโปรโตคอลที่เป็นเทคนิคในการอำพรางข้อมูลลงไปยังพื้นที่ว่างหรือในส่วนที่ไม่ได้ถูกเรียกใช้ในการรับส่งข้อมูล และมุ่งเน้นในเรื่องของวิทยาการอำพรางข้อมูลลงไปยังรูปภาพดิจิตอล ซึ่งมีหลากหลายรูปแบบ ทำให้ต้องเลือกขั้นตอนวิธีในการอำพรางข้อมูลลงไปยังรูปภาพที่แตกต่างกันแต่ละรูปแบบนั้น การดูรูปแบบของไฟล์รูปภาพดิจิตอล โดยตรวจสอบจากความละเอียดของภาพ ความแตกต่างกันระหว่างระบบสีของภาพที่มีความลึกบิตไม่เท่ากัน การบีบอัดไฟล์ภาพดิจิตอลมีอยู่สองประเภท ซึ่งก็คือการบีบอัดแบบ lossless ที่จะไม่ทำการลบรายละเอียดส่วนใดออกจากภาพเดิมเลย แต่จะใช้การแทนที่ด้วยสูตรทางคณิตศาสตร์ และการบีบอัดแบบ losssy ที่จะทำการลบข้อมูลที่ไม่จำเป็นในภาพนั้น หรือส่วนการแสดงผลที่มนุษย์ไม่สามารถมองเห็นได้ด้วยตาเปล่า และการเลือกขั้นตอนวิธีจากโดเมนของภาพ อีกทั้งยังกล่าวถึงขั้นตอนวิธีที่ใช้ในการอำพรางข้อมูลลงไปยังภาพดิจิตอล ตัวอย่างเช่น การอำพรางข้อมูลไปยังภาพดิจิตอลโดยเทคนิคการสลับบิตของข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดของภาพในแต่ละพิกเซล ซึ่งในรูปภาพดิจิตอลที่มีระบบภาพ 8 บิต กับ ระบบภาพ 24 บิต นั้นจะใช้เทคนิคที่ต่างกันเล็กน้อยโดยในระบบภาพ 8 บิต การซ่อนข้อมูลลงไปยังรูปภาพดิจิตอลสามารถทำได้เพียง 1 บิตต่อพิกเซล แต่หากเป็นระบบภาพแบบ 24 บิต จะสามารถซ่อนข้อมูลไปลงยังภาพดิจิตอลได้ถึง 3 บิตต่อพิกเซลเลยทีเดียว และเทคนิคการอำพรางข้อมูลลงไปยังบิตที่มีนัยสำคัญต่ำสุดยังสามารถทำได้ถึง 2 บิตที่มีนัยสำคัญต่ำสุด ซึ่งมนุษย์ก็ยังไม่สามารถมองเห็นได้ด้วยตาเปล่าเช่นกัน เพราะฉะนั้นหากทำการซ่อนข้อมูลลงไปยังบิตที่มีนัยสำคัญต่ำสุด 2 อันดับสุดท้าย ภาพระบบ 8 บิตก็จะสามารถซ่อนข้อมูลได้ 2 ไบต์ต่อพิกเซล และภาพระบบ 24 บิตก็จะสามารถซ่อนข้อมูลได้ 6 ไบต์ต่อพิกเซลเลยทีเดียว แต่เทคนิคดังกล่าวสามารถใช้ได้กับข้อมูลภาพดิจิตอลที่ถูกบีบอัดแบบ lossless เท่านั้น แต่ในเอกสารก็ได้กล่าวถึงการอำพรางข้อมูลในไฟล์ภาพดิจิตอลที่ถูกบีบอัดแบบ lossy ด้วย และสุดท้ายได้เปรียบเทียบและสรุปข้อดีข้อเสียของขั้นตอนวิธีการอำพรางข้อมูลไปยังไฟล์รูปภาพด้วยวิธีการต่างๆ

Ms. Dhobale Dhanashri D. et al. (2010) ได้นำเสนอการอำพรางข้อมูลในอุปกรณ์สมาร์ทโฟนและส่งผ่านเครือข่ายโดยบริการส่งข้อความสื่อประสม เพื่อความปลอดภัยในการรับส่งข้อมูลลับ อีกทั้งยังเพิ่มความสะดวกสบายให้แก่ผู้ใช้อุปกรณ์สมาร์ทโฟน โดยได้กล่าวถึงความสำคัญของอุปกรณ์สมาร์ทโฟนที่ได้รับความนิยมเพิ่มมากขึ้นกว่าในอดีต และอุปกรณ์สมาร์ทโฟนก็ยังมีคุณสมบัติต่างๆ เพิ่มขึ้นกว่าในสมัยก่อนมากมาย ซึ่งหนึ่งในความสามารถนั้นก็คือการรับส่งข้อความสื่อประสม ที่โทรศัพท์เกือบทุกเครื่องที่ผลิตออกมาในปัจจุบันล้วนแล้วแต่รองรับความสามารถด้านนี้ทั้งสิ้น และผู้เขียนมีความสนใจในศาสตร์ด้านการอำพรางข้อมูล จึงได้พัฒนา

โปรแกรมที่สามารถอำพรางข้อความลงไปยังรูปภาพเพื่อในใช้การส่งเป็นข้อความสื่อประสม และผู้เขียนยังได้กล่าวถึงขั้นตอนวิธีที่ใช้ในการอำพรางข้อมูลในโปรแกรมดังกล่าว ตัวอย่างเช่นการอำพรางข้อความลงไปยังรูปภาพดิจิตอลที่มีรูปแบบเป็น PNG ซึ่งถูกบีบอัดแบบ lossless ผู้เขียนได้ใช้วิธีการสลับข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดเพื่อที่จะแฝงข้อมูลลับลงไปยังภาพ ซึ่งวิธีการนี้มีข้อดีคือไม่ทำให้ขนาดของภาพเปลี่ยนแปลงไป แต่ข้อเสียคือไม่สามารถใช้วิธีการนี้กับรูปภาพดิจิตอลที่ถูกบีบอัดแบบ lossy อย่างเช่นไฟล์ภาพแบบ JPEG ได้ ซึ่งผู้เขียนก็ได้กล่าวถึงการอำพรางข้อมูลไปยังไฟล์ภาพแบบ JPEG ซึ่งจะต้องใช้การแปลงโคไซน์ไม่ต่อเนื่องเข้ามาช่วยในการอำพรางข้อมูล และกล่าวถึงขั้นตอนวิธีต่างๆ ที่สามารถอำพรางข้อมูลลงไปยังภาพที่ถูกบีบอัดแบบ lossy ดังกล่าวได้ เช่นขั้นตอนวิธี Jsteg ที่หลังจากการทำควอนไทซ์ จะแทนที่ข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดของสัมประสิทธิ์ความถี่ เป็นต้น

Mohammad Shirali-Shahreza (2007) ได้นำเสนอวิทยาการอำพรางข้อมูลในบริการส่งข้อความสื่อประสม ซึ่งเป็นทางเลือกสำหรับการสื่อสารข้อมูลลับ โดยได้กล่าวถึงข้อมูลเบื้องต้นเกี่ยวกับบริการส่งข้อความสื่อประสม ไม่ว่าจะเป็นรูปแบบของไฟล์ที่รองรับ ขนาดของไฟล์ที่สามารถส่งผ่านบริการส่งข้อความสื่อประสมได้ และยังได้กล่าวถึงการอำพรางข้อความในบริการส่งข้อความสั้นด้วยเทคนิคการใช้ตัวอักษรหรือคำที่ออกเสียงเหมือนกับคำๆ นั้น เพื่อเป็นการตัดข้อความให้สั้นลง และยังแสดงตัวอย่างของการอำพรางข้อความในข้อความเบื้องต้น แต่ผู้เขียนมุ่งเน้นที่การอำพรางข้อความลงไปยังไฟล์รูปภาพดิจิตอล และผู้เขียนได้กล่าวถึงวิธีการที่ใช้ในการอำพรางข้อมูลลงไปยังไฟล์ภาพดิจิตอลแบบ PNG ซึ่งเป็นไฟล์ภาพดิจิตอลที่มีการบีบอัดแบบ lossless โดยใช้วิธีการสลับบิตข้อมูลที่ต้องการซ่อนไปยังบิตที่มีนัยสำคัญต่ำสุดของพิกเซลของรูปภาพนั้นๆ ซึ่งโปรแกรมที่พัฒนาได้ใช้ภาษาโปรแกรม J2ME ในการพัฒนา สุดท้ายกล่าวถึงแนวทางการพัฒนาสามารถนำไปประยุกต์ให้สามารถอำพรางข้อมูลลงไปยังไฟล์ภาพแบบ JPEG โดยใช้ขั้นตอนวิธีแบบ F5 ต่อไป

Pasquale Paola and Paolo Manzo (2006) ได้นำเสนอแอพพลิเคชั่นที่ใช้ในการอำพรางข้อมูลตัวอักษรลงบนรูปภาพ ผ่านทางโทรศัพท์เคลื่อนที่ที่ใช้ระบบปฏิบัติการแอนดรอยด์ ซึ่งแอพพลิเคชั่นดังกล่าวถูกพัฒนาขึ้นด้วยภาษาจาวา และใช้ขั้นตอนวิธีแบบ LSB ในการอำพรางข้อความลงบนรูปภาพ ซึ่งแอพพลิเคชั่นดังกล่าวสามารถบันทึกรูปภาพที่อำพรางไว้ในหน่วยความจำของเครื่องสมาร์ทโฟน และสามารถส่งรูปภาพที่ถูกอำพรางข้อมูลแล้วนั้นผ่านทาง MMS ได้อีกด้วย โดยขั้นตอนวิธีที่ใช้นั้นจะเป็นการซ่อนข้อมูลไว้ที่ 2 บิตสุดท้ายของแต่ละแชลแนลสีของแต่ละพิกเซล คือสามารถซ่อนข้อมูลได้พิกเซลละ 6 บิต

มนธวัล พรรณวิเชียรและคณะ (2010) ได้นำเสนอการอำพรางข้อมูลลงภาพโดยใช้วิธีการสลับบิตร่วมกับการเข้ารหัสข้อมูลโดยใช้มาตรฐานการเข้ารหัสข้อมูลระดับสูง เพื่อเพิ่มความปลอดภัยในการรับส่งข้อมูล โดยกล่าวถึงปัจจัยและผลกระทบต่อข้อมูลที่ทำให้ต้องใช้วิทยาการอำพรางข้อมูล เพิ่มไปจากการที่ใช้เทคนิคการเข้ารหัสข้อมูลเพียงอย่างเดียว และกล่าวถึงขั้นตอนวิธีเดิมของการเข้ารหัสข้อมูลที่ขนาดความปลอดภัย โดยในอันดับแรกจะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับการอำพรางข้อมูล ความแตกต่างระหว่างการอำพรางข้อมูลและการเข้ารหัสข้อมูล และยังกล่าวถึงวิธีการสลับบิตสำคัญน้อยสุด ซึ่งเป็นวิธีการที่ใช้ในการอำพรางข้อมูล โดยยกตัวอย่างการสลับบิตที่มีนัยสำคัญน้อยสุดเบื้องต้น และกล่าวถึงมาตรฐานการเข้ารหัสข้อมูลแบบ Advance Encryption Standard (AES) ว่าเป็นขั้นตอนวิธีที่มีความรวดเร็วสามารถใช้คีย์ได้หลายๆ ขนาด และกล่าวถึงมาตรฐานการเข้ารหัสข้อมูลแบบ Rivest-Shamir-Adelman Encryption (RSA) ที่ใช้กุญแจแบบอสมมาตรในการเข้ารหัสและถอดรหัสข้อมูล จากนั้นกล่าวถึงแฮชฟังก์ชัน ซึ่งเป็นการเข้ารหัสที่ไม่สามารถถอดรหัสได้ แต่เป็นการรหัสที่มีไว้ตรวจสอบความถูกต้องของข้อมูลว่าไม่มีการเปลี่ยนแปลงไปจากต้นฉบับ หรือความคงสภาพของต้นฉบับ โดยการแปลงข้อมูลให้อยู่ในรูปของเมสเสจไดเจสต์ ซึ่งเป็นเอกลักษณ์ของข้อมูลนั้นๆ นั่นเอง จากนั้นผู้เขียนได้พูดถึงการพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถอำพรางข้อมูลไปยังไฟล์ภาพดิจิตอลที่ถูกบีบอัดแบบ lossless โดยใช้วิธีการสลับข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดของพิกเซลของภาพ 2 ภาพ และสามารถเข้ารหัสข้อมูลลับด้วยขั้นตอนวิธี AES และใช้ขั้นตอนวิธี RSA ในการเข้ารหัสและถอดรหัส รหัสผ่านของขั้นตอนวิธี AES อีกทีหนึ่ง อีกทั้งยังมีการทำแฮชฟังก์ชันที่ใช้ในการตรวจสอบความคงสภาพของข้อมูลโดยใช้ขั้นตอนวิธีแบบ MD5 และยังได้ทำสรุปประเมินการใช้โปรแกรมคอมพิวเตอร์ดังกล่าว ซึ่งคุณภาพที่ประเมินได้ออกมาอยู่ในระดับดี และได้ให้ข้อเสนอแนะเกี่ยวการโปรแกรมไว้ว่าสามารถเปลี่ยนขั้นตอนวิธีที่ใช้ในการเข้ารหัสและถอดรหัสข้อมูลเพื่อเพิ่มความปลอดภัยให้มากยิ่งขึ้น ผู้เขียนยังแนะนำให้พัฒนาวิธีการอำพรางข้อมูลไปใช้อย่างอื่นนอกจากการสลับบิตที่มีนัยสำคัญต่ำสุด และสุดท้ายผู้เขียนยังได้แนะนำให้ใช้สื่อประเภทอื่นๆ ในการอำพรางข้อมูลเช่น ไฟล์เสียง หรือไฟล์วิดีโอ เป็นต้น

ธนาวัฒน์ เดชกำแหงและคณะ (2010) ได้นำเสนอการอำพรางข้อมูลลงไฟล์เสียงโดยใช้วิธีการสลับบิตร่วมกับการเข้ารหัสข้อมูล เพื่อเพิ่มความปลอดภัยในการรับส่งข้อมูล โดยกล่าวถึงปัจจัยที่ทำให้ต้องใช้มีการนำวิทยาการอำพรางข้อมูลมาใช้ และผลกระทบจากการใช้วิทยาการอำพรางข้อมูลในทางที่ผิด โดยในอันดับแรกจะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับการอำพรางข้อมูล ความแตกต่างระหว่างการอำพรางข้อมูลและการเข้ารหัสข้อมูล และยังกล่าวถึงเทคนิควิธีการสลับบิตสำคัญน้อยสุด ซึ่งเป็นวิธีการที่ใช้ในการอำพรางข้อมูล โดยยกตัวอย่างการสลับบิตที่มีนัยสำคัญ

น้อยสุดเบื้องต้น และกล่าวถึงมาตรฐานการเข้ารหัสข้อมูลแบบทูฟิช และขั้นตอนวิธีของมาตรฐานการเข้ารหัสข้อมูลแบบทูฟิช จากนั้นกล่าวถึงแฮชฟังก์ชัน ซึ่งเป็นการเข้ารหัสที่ไม่สามารถถอดรหัสได้ แต่เป็นการรหัสที่มีไว้ตรวจสอบความถูกต้องของข้อมูลว่าไม่มีการเปลี่ยนแปลงไปจากต้นฉบับหรือความคงสภาพของต้นฉบับ โดยการแปลงข้อมูลให้อยู่ในรูปของเมสเสจไดเจสต์ ซึ่งเป็นเอกลักษณ์ของข้อมูลนั้นๆ นั่นเอง จากนั้นผู้เขียนได้พูดถึงการพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถอำพรางข้อมูลไปยังเสียงโดยใช้วิธีการสลับข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดของพิกเซลของไฟล์เสียง 2 ไฟล์ และสามารถเข้ารหัสข้อมูลลับด้วยขั้นตอนวิธีทูฟิชก่อนที่จะอำพรางข้อมูลดังกล่าวไปยังไฟล์เสียง อีกทั้งยังมีการทำแฮชฟังก์ชันที่ใช้ในการตรวจสอบความคงสภาพของข้อมูลโดยใช้ขั้นตอนวิธีแบบ MD5 และยังได้ทำสรุปประเมินการใช้โปรแกรมคอมพิวเตอร์ดังกล่าว ซึ่งคุณภาพที่ประเมินได้ออกมาอยู่ในระดับดี และได้ให้ข้อเสนอแนะเกี่ยวการโปรแกรมไว้ว่าสามารถเปลี่ยนขั้นตอนวิธีที่ใช้ในการเข้ารหัสและถอดรหัสข้อมูลเพื่อเพิ่มความปลอดภัยให้มากยิ่งขึ้น ผู้เขียนยังแนะนำให้พัฒนาวิธีการอำพรางข้อมูลไปใช้อย่างอื่นนอกจากการสลับบิตที่มีนัยสำคัญต่ำสุด เช่น Parity Coding, Phase Coding และ Spread Specturm เป็นต้น ผู้เขียนยังได้แนะนำให้พัฒนาให้สามารถใช้ได้กับไฟล์เสียงรูปแบบอื่นๆ เช่น .wav .cda .wma .asf เป็นต้น และสุดท้ายผู้เขียนยังได้แนะนำให้ใช้สื่อประเภทอื่นๆ ในการอำพรางข้อมูลเช่น ไฟล์ภาพ หรือไฟล์วิดีโอ เป็นต้น

ภัทรพงษ์ เรียบร้อยเจริญและคณะ (2010) ได้นำเสนอแอพพลิเคชั่นสำหรับอำพรางข้อมูลลงในภาพผ่านบริการรับและส่งสารแบบสื่อประสม ซึ่งมีจุดมุ่งหมายหลักในการปกปิดข้อมูลทำให้บุคคลอื่นไม่รู้ว่ามีการซ่อนข้อมูลลับอยู่ เพื่อเพิ่มความปลอดภัยในการรับส่งข้อมูล โดยกล่าวถึงปัจจัยที่ทำให้ต้องใช้มีการนำวิทยาการอำพรางข้อมูลมาใช้ โดยในอันดับแรกจะกล่าวถึงความรู้เบื้องต้นเกี่ยวกับการอำพรางข้อมูล ความแตกต่างระหว่างการอำพรางข้อมูลและการเข้ารหัสข้อมูล และกล่าวถึงมาตรฐานการเข้ารหัสข้อมูลแบบโบว์ฟิส และขั้นตอนวิธีของมาตรฐานการเข้ารหัสข้อมูลแบบโบว์ฟิส จากนั้นพูดถึงงานวิจัยที่เกี่ยวข้องต่างๆ และผู้เขียนได้พูดถึงการพัฒนาแอพพลิเคชั่นที่สามารถอำพรางข้อความลงไปยังรูปภาพโดยใช้วิธีการสลับข้อมูลไปยังบิตที่มีนัยสำคัญต่ำสุดของพิกเซลของไฟล์เสียง 2 ไฟล์ และสามารถเข้ารหัสข้อมูลลับด้วยขั้นตอนวิธีโบว์ฟิสก่อนที่จะอำพรางข้อมูลดังกล่าวไปยังไฟล์รูปภาพ และยังได้ทำสรุปประเมินการใช้โปรแกรมคอมพิวเตอร์ดังกล่าว ซึ่งคุณภาพที่ประเมินได้ออกมาอยู่ในระดับดี

จักริน สุขเพ็งและคณะ (2010) ได้นำเสนอโปรแกรมฝังตัวบนโทรศัพท์เคลื่อนที่เพื่อช่วยป้องกันการเปิดดูข้อมูลภาพโดยใช้วิธีการเข้ารหัสแบบ 64-Bit Block Cipher (Blowfish) และการพิสูจน์ตัวตนภาพโดยการซ่อนข้อความลับด้วยวิธีการสร้างลายน้ำดิจิตอลแบบไม่สามารถ

มองเห็นได้ เพื่อป้องกันข้อมูลภาพที่จัดเก็บอยู่ในเครื่องโทรศัพท์เคลื่อนที่จากผู้ที่ไม่ได้รับอนุญาต โดยกล่าวถึงทฤษฎีการเข้ารหัสข้อมูลแบบ 64-Bit Block Cipher (โบว์ฟิช) ซึ่งเป็นการเข้ารหัสข้อมูลแบบกุญแจสมมาตร และบอกถึงขั้นตอนวิธีของการเข้ารหัสข้อมูลดังกล่าว และยังกล่าวถึงการสร้างลายน้ำดิจิตอลแบบไม่สามารถมองเห็นได้ รวมถึงขั้นตอนวิธีของมัน ซึ่งจะต้องใช้การแปลงโคไซน์ไม่ต่อเนื่องมาช่วยเพื่อที่จะทำการแก้ไขสัมประสิทธิ์ของการแปลง และกล่าวถึงการพัฒนาโปรแกรมว่าใช้เครื่องมือใดในการพัฒนา อีกทั้งยังกล่าวถึงเหตุผลที่เลือกใช้ขั้นตอนวิธีแบบโบว์ฟิชในการเข้ารหัสข้อมูล เนื่องจากไม่ซับซ้อนทำให้สามารถทำงานได้รวดเร็ว และไม่ใช้ทรัพยากรมาก จึงเหมาะกับอุปกรณ์มือถือ และยังแสดงผังลำดับของโปรแกรม และตัวอย่างโค๊ดในภาษาจาวา และอันดับสุดท้ายกล่าวถึงสรุปผลการดำเนินงาน

ธมกร บุ้งจันทร์และคณะ (2011) ได้นำเสนอเทคนิคการอำพรางข้อความไว้ในไฟล์ภาพเจเพ็ก ซึ่งเป็นไฟล์ภาพที่ได้รับความนิยมสูงที่สุดบนอินเทอร์เน็ตเนื่องจากมีการบีบอัดไฟล์ภาพทำให้ได้ภาพที่มีขนาดเล็ก เหมาะกับการใช้งานมากที่สุด โดยผู้วิจัยได้ให้ความสำคัญของการอำพรางข้อมูลลงไปยังสื่อพาหะไว้ 3 อย่างคือ ความแนบเนียน, ปริมาณ และความทนทาน และได้อธิบายถึงข้อมูลเชิงเทคนิคของไฟล์ภาพแบบเจเพ็ก ซึ่งใช้หลักการพื้นฐานทางคณิตศาสตร์ที่ชื่อว่า DCT (Discrete Cosine Transform) ในการบีบอัดภาพ โดยผู้วิจัยได้ใช้เทคนิคการอำพรางข้อความไว้ที่ DCT ที่เหมาะกับไฟล์ภาพชนิดเจเพ็ก และใช้วิธีการวัดค่าความผิดเพี้ยนของข้อมูลภาพด้วย PSNR (Peak-Signal to Noise Ratio) ซึ่งพบข้อบกพร่องโดยกระบวนการแทนที่ข้อความบางอักขระอาจจะไม่สามารถอำพรางได้ถ้าไม่สามารถหาตำแหน่งของข้อมูลที่มีค่าพิกเซลที่สัมพันธ์กับอักขระที่จะอำพรางในภาพมาใช้ในการอำพรางข้อมูลได้

# บทที่ 3
# ระเบียบวิธีวิจัย

แอพพลิเคชั่นสำหรับการอำพรางข้อมูลในรูปแบบข้อความตัวอักษร ลงไปยังข้อมูลที่เป็นไฟล์ภาพดิจิตอลที่มีการบีบอัดแบบ lossless โดยเข้ารหัสข้อมูลด้วยมาตรฐานการเข้ารหัสลับขั้นสูง เป็นแอพพลิเคชั่นบนอุปกรณ์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์ โดยใช้หลักการสลับข้อมูลที่ต้องการซ่อนไปยังบิตที่มีนัยสำคัญต่ำสุดของไฟล์ภาพดิจิตอล เพื่อสร้างความปลอดภัยให้กับการรับส่งข้อมูล โดยไม่ต้องการให้บุคคลที่ไม่เกี่ยวข้องรับรู้ถึงการมีอยู่ของข้อมูลลับดังกล่าว

## 3.1 เครื่องมือที่ใช้ในการวิจัย

3.1.1 โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ จำนวน 2 เครื่อง ดังนี้

1) HTC One X ระบบสัญญาณ Dual Mode (WCDMA/GSM)
   - ประมวลผลการทำงานด้วย Quad-Core Processor (ชิปเซ็ต Nvidia Tegra 3)
   - ความเร็วในการประมวลผล 1.5 GHz พร้อมระบบปฏิบัติการ Android OS เวอร์ชัน 4.0 (Ice Cream Sandwich)
   - หน่วยความจำภายในสำหรับเก็บบันทึกข้อมูลขนาด 32 GB
   - หน่วยความจำ RAM ขนาด 1 GB
2) Samsung Galaxy Tob P1000T ระบบสัญญาณ Dual Mode (WCDMA/GSM)
   - ประมวลผลการทำงานด้วย ARM Cortex A8 Processor
   - ความเร็วในการประมวลผล 1 GHz พร้อมระบบปฏิบัติการ Android OS เวอร์ชัน 2.2 (Froyo)
   - หน่วยความจำภายในขนาด 32 GB
   - หน่วยความจำ RAM ขนาด 512 MB

3.1.2 เครื่องคอมพิวเตอร์ส่วนบุคคล จำนวน 1 เครื่อง มีคุณสมบัติดังนี้

- CPU Intel Xeon E31230 @ 3.20 GHz
- RAM 4 GB DDR3
- Hard Disk 1 TB 7200 RPM

- Graphic Card NVIDIA Quadro 600
- DVD Writer (Dual Layer Support)
- MS Windows 7 Ultimate

3.1.3 ซอฟต์แวร์สำหรับใช้ในการพัฒนาแอพพลิเคชั่น

- Eclipse Development Tools and Java Development Kit (JDK)
- ADT (Android Development Tools Plugin for eclipse)
- Android SDK
- Android Virtual Device Manager (Emulator)

**3.2 ขั้นตอนการทำงานของระบบเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ**

ระบบการอำพรางข้อมูลลงไปยังไฟล์ภาพ เป็นระบบที่มีวัตถุประสงค์เพื่อการส่งข้อมูลลับไปยังผู้รับ โดยไม่ต้องการให้ผู้ที่ไม่เกี่ยวข้องหรือบุคคลที่ไม่พึงประสงค์ รับรู้ถึงข้อมูลข่าวสารดังกล่าว โดยใช้วิธีการปกปิดข้อมูลที่ต้องการส่งด้วยข้อมูลที่ดูไม่มีความน่าสงสัยหรือผิดปกติใดๆ ซึ่งการใช้งานที่สมบูรณ์คือผู้รับและผู้ส่งเท่านั้นที่สามารถทราบข้อมูลที่ถูกอำพรางอยู่ในไฟล์ภาพนั้นได้ และไม่มีผู้ใดล่วงรู้ข้อมูลดังกล่าว และยังได้เพิ่มการเข้ารหัสลับข้อมูลด้วยมาตรฐานการเข้ารหัสลับขั้นสูงเพื่อความปลอดภัยของข้อมูลมากขึ้น โดยขั้นตอนของการอำพรางข้อมูลในไฟล์ภาพแบ่งออกเป็น 2 ขั้นตอน ดังนี้

3.2.1 ขั้นตอนการอำพรางข้อมูลลงไปยังไฟล์ภาพ ซึ่งเป็นขั้นตอนของทางผู้ส่ง

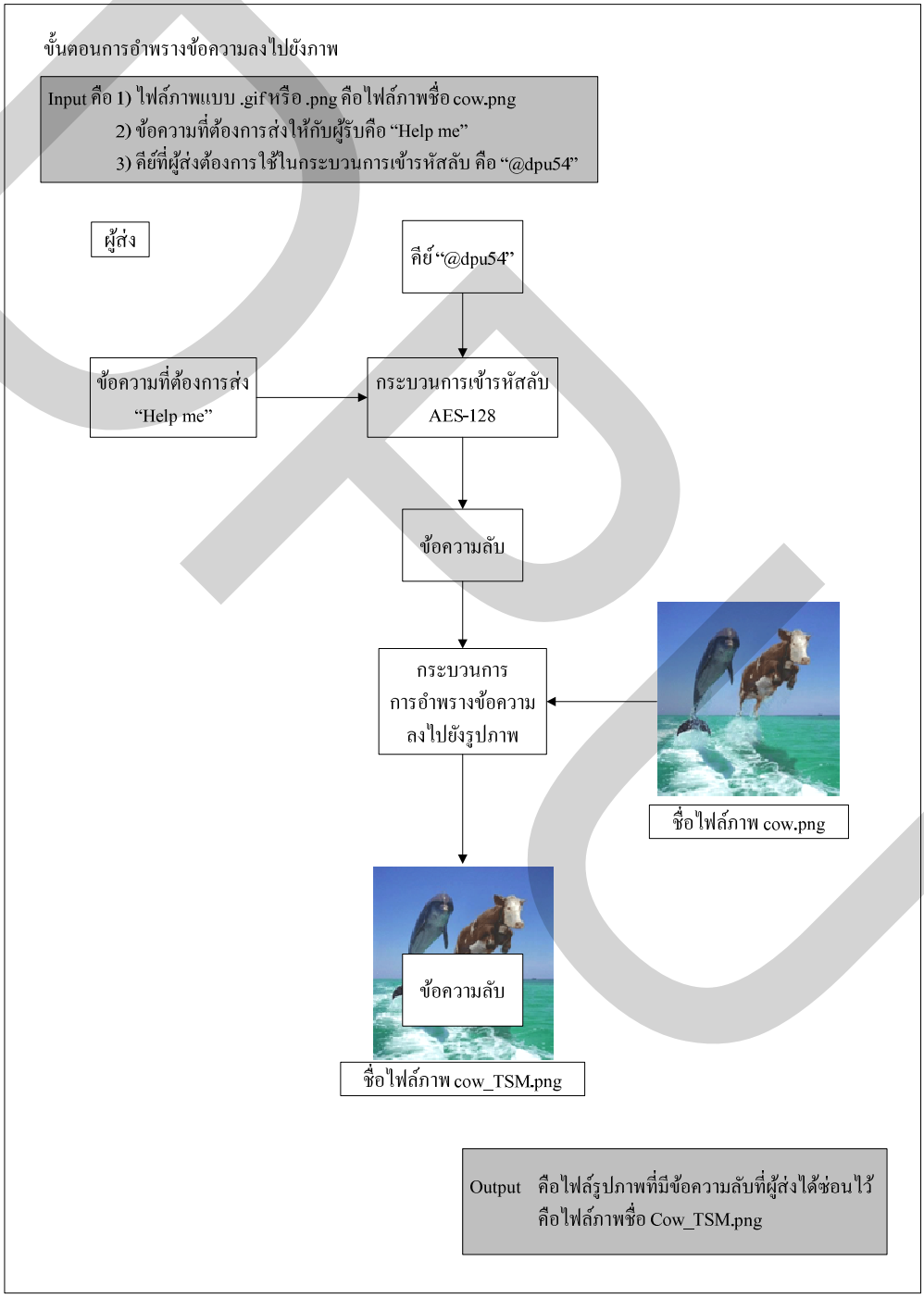
โดยภาพที่ 3.1 แสดงถึงขั้นตอนของการเข้ารหัสและอำพรางข้อความที่ผู้ส่งต้องการส่งให้กับผู้รับไปยังไฟล์ภาพที่อยู่ในรูปแบบ PNG ซึ่งอธิบายได้ดังนี้

1) Input ประกอบด้วย
   1. ไฟล์ภาพประเภทไฟล์ .png จากรูปคือไฟล์ภาพชื่อ cow.png
   2. ข้อความที่ผู้ส่งต้องการส่งให้กับผู้รับจากรูปคือข้อความว่า “Help me”
   3. คีย์ที่ผู้ส่งต้องการใช้ในกระบวนการเข้ารหัสลับคือ “@dpu54”
2) Output คือ ไฟล์รูปภาพที่มีข้อความลับที่ผู้ส่งได้ซ่อนไว้คือไฟล์ภาพชื่อ cow_TSM.png
3) Process มีขั้นตอนดังนี้

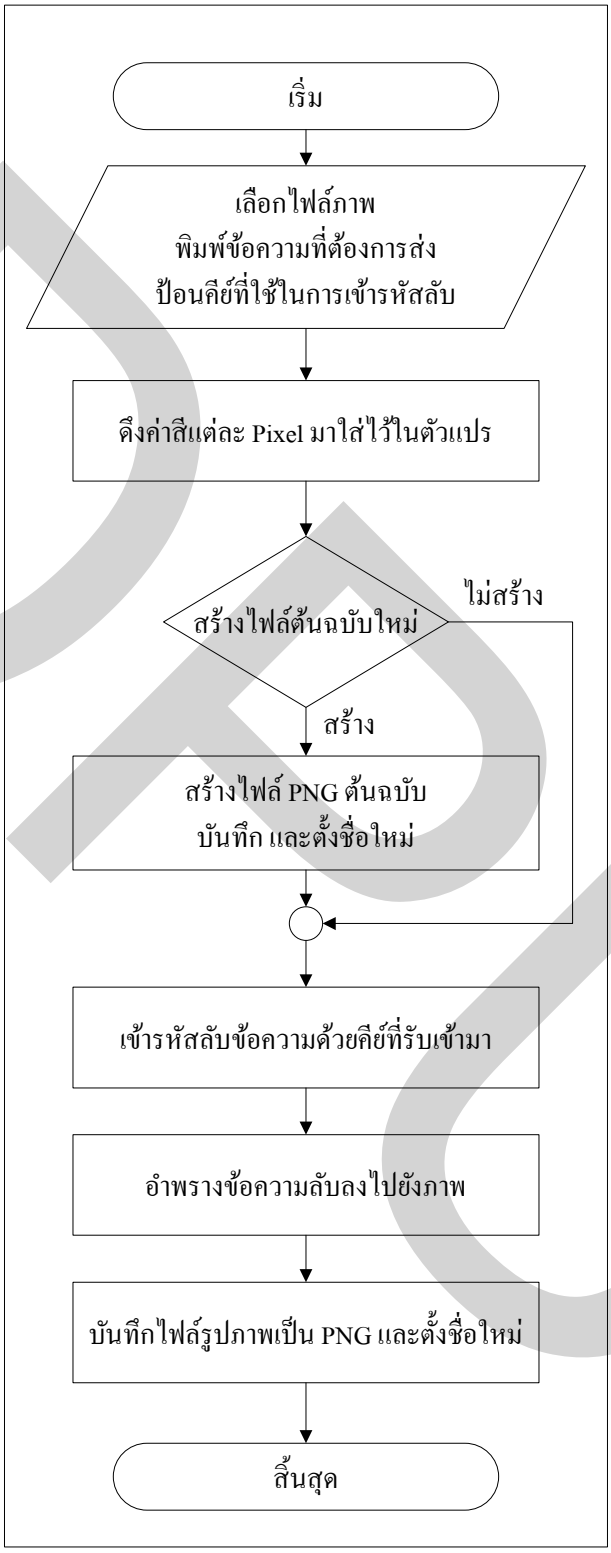
1. ระบบจะรับข้อมูลไฟล์ภาพ คือไฟล์ภาพชื่อ cow.png ข้อความที่ผู้ส่งต้องการส่งให้กับผู้รับ จากรูปคือข้อความว่า “Help me” และคีย์ที่ใช้ในการเข้ารหัสข้อความคือ “@dpu54”
2. นำข้อความว่า “Help me” มาผ่านกระบวนการเข้ารหัสลับด้วยคีย์ “@dpu54”

3. นำข้อความลับที่ได้ มาอำพรางไว้ในไฟล์ภาพ โดยทำการตั้งชื่อไฟล์ภาพที่มีข้อความลับซ่อนอยู่ จากรูปตั้งชื่อเป็น cow_TSM.png

4. ผู้ส่งทำการส่งไฟล์ภาพ cow_TSM.png ไปให้กับผู้รับ




**ภาพที่ 3.1** ขั้นตอนการเข้ารหัสและอำพรางข้อความลงไปยังไฟล์ภาพ




**ภาพที่ 3.2** รายละเอียดการทำงานของการเข้ารหัสและอำพรางข้อความไปยังรูปภาพ

จากภาพที่ 3.2 แสดงรายละเอียดการทำงานของการเข้ารหัสและอำพรางข้อมูลไปยังรูปภาพ โดยขั้นตอนการทำงานสามารถอธิบายได้ดังนี้

1) ผู้ใช้ทำการเลือกไฟล์ภาพต้นฉบับที่ต้องการใช้เป็นไฟล์ภาพพาหะในการอำพรางข้อความ จากนั้นป้อนคีย์เพื่อใช้ในการเข้ารหัสข้อความ และป้อนข้อความที่ต้องการอำพราง

2) แอพพลิเคชั่นจะทำการดึงค่าสีแต่ละพิกเซลของรูปภาพต้นฉบับมาเก็บไว้ที่ตัวแปรอาเรย์

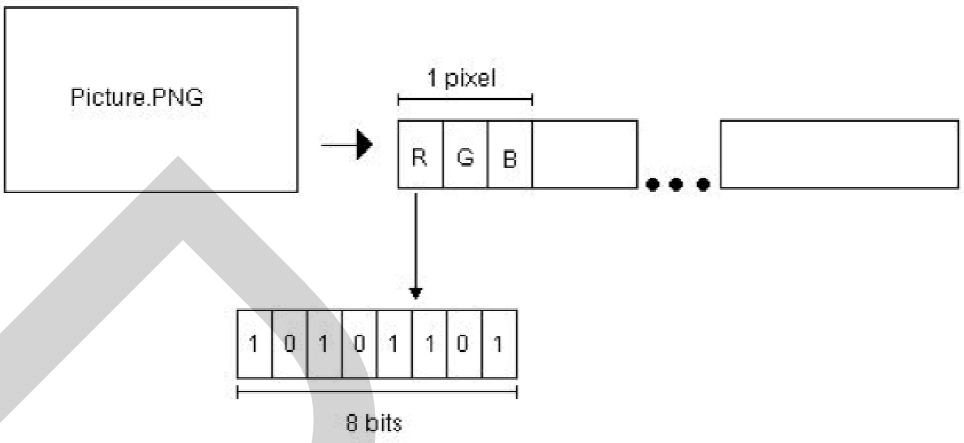
3) แอพพลิเคชั่นจะทำการตรวจสอบว่าได้ผู้ใช้ได้กำหนดคุณสมบัติให้แอพพลิเคชั่นสร้างไฟล์ภาพต้นฉบับในรูปแบบ PNG ขึ้นมาใหม่หรือไม่ หากได้กำหนดไว้แอพพลิเคชั่นจะทำการสร้างไฟล์ต้นฉบับขึ้นมาใหม่ หากไม่ได้กำหนดไว้แอพพลิเคชั่นจะข้ามไปยังขั้นตอนถัดไป

4) แอพพลิเคชั่นจะการเข้ารหัสลับข้อความด้วยคีย์ที่ผู้ใช้ได้กำหนดเข้ามาด้วยวิธีการ AES

5) แอพพลิเคชั่นจะทำการอำพรางข้อความที่ถูกเข้ารหัสแล้วลงไปยังรูปภาพ

6) แอพพลิเคชั่นจะทำการบันทึกไฟล์รูปภาพเป็นไฟล์ใหม่ที่ใช้รูปแบบไฟล์ภาพ PNG และกำหนดชื่อของรูปภาพใหม่

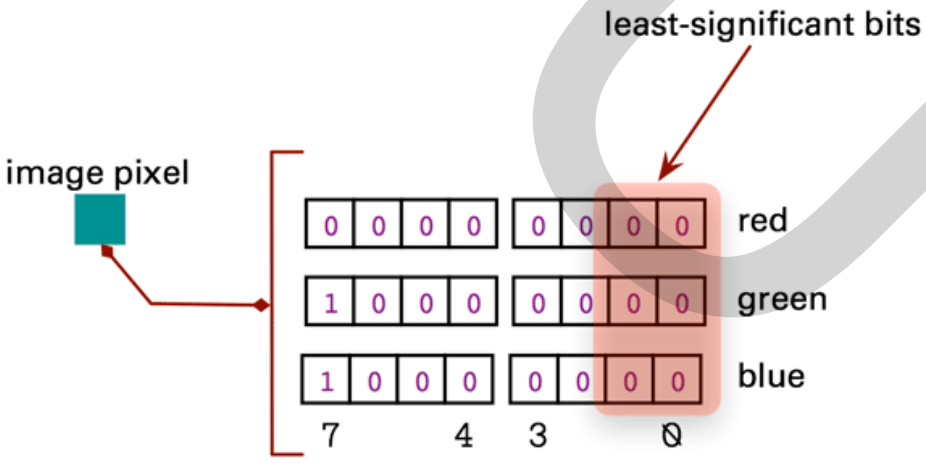
โดยการซ่อนข้อความลงในรูปภาพนั้นจะใช้วิธีการซ่อนลงไปยังบิตที่มีนัยสำคัญต่ำสุด หรือที่เรียกว่า LSB แอพพลิเคชั่นนี้จะใช้การซ่อนบิตข้อความลงไปยัง 2 หลักสุดท้ายของแต่ละค่าสีของแต่ละพิกเซล ซึ่งในแต่ละพิกเซลของรูปภาพสี 24 บิตจะเก็บค่าสี 3 ค่าคือค่าสีแดง (R) ค่าสีเขียว (G) และค่าสีน้ำเงิน (B) ซึ่งแต่ละค่าจะมีขนาดข้อมูล 8 บิต ดังภาพที่ 3.3




**ภาพที่ 3.3** รายละเอียดของภาพที่มีระบบภาพแบบ 24 บิต

**ที่มา:** http://blog.pupasoft.com/2009/11/09/aeronzsteganography/

โดยในขั้นตอนการทำงานของแอพพลิเคชั่น ข้อความที่ผู้ใช้ป้อนเข้ามาจะผ่านกระบวนการเข้ารหัส AES และต่อส่วนหัวและส่วนท้ายของข้อมูลดังกล่าวด้วยอักขระที่ได้จัดเตรียมไว้ จากนั้นจะทำการเปลี่ยนข้อมูลดังกล่าวให้อยู่ในรูปแบบของไบต์อาเรย์ และนำบิตในไบต์ของข้อมูลดังกล่าวไปแทนที่กับ 2 บิตสุดท้ายของแต่ละพิกเซลสี ดังภาพที่ 3.4




**ภาพที่ 3.4** การซ่อนข้อความลงไปยังบิตที่มีนัยสำคัญต่ำสุด 2 หลัก

**ที่มา:** http://www.drdobbs.com/security/how-to-secure-and-authenticate-images-us/229400454

โปรแกรมหลักในส่วนของขั้นตอนวิธีการซ่อนบิตข้อความลงในรูปภาพแสดงดังภาพที่ 3.5 โดยโปรแกรมในส่วนนี้จะทำหน้าที่เปลี่ยนบิตที่มีนัยสำคัญต่ำสุด 2 บิตสุดท้ายของแต่ละพิกเซลสีให้แทนที่ด้วยบิตของข้อความที่ผ่านกระบวนการเข้ารหัสแล้ว ซึ่งในแต่ละบรรทัดมีขั้นตอนการทำงานต่างๆ ดังนี้

1) บรรทัดที่ 6 กำหนดค่าเริ่มต้นให้กับตัวแปรอาเรย์ เพื่อใช้ในการ Shift bit ของรูปภาพ
2) บรรทัดที่ 7 กำหนดค่าเริ่มต้นให้กับตัวแปรอาเรย์เพื่อใช้ในการ Shift bit ของข้อความ
3) บรรทัดที่ 8 เปลี่ยนข้อความที่จะซ่อนให้เป็นไบต์อาเรย์
4) บรรทัดที่ 9 กำหนดค่าเริ่มต้นสำหรับวนลูบ 3 แชลแนลสี
5) บรรทัดที่ 10 กำหนดค่าเริ่มต้นสำหรับเช็คตัวอักษร
6) บรรทัดที่ 11 สร้างตัวแปรอาเรย์มารอรับค่าผลลัพธ์
7) บรรทัดที่ 12 วนลูบตั้งแต่ แถวแรกถึงแถวสุดท้าย
8) บรรทัดที่ 13 วนลูบตั้งแต่ คอลัมน์แรกถึงคอลัมน์สุดท้าย
9) บรรทัดที่ 14 สร้างตัวแปรมาเก็บตำแหน่งของอาเรย์พิกเซล
10) บรรทัดที่ 16 วนลูบแชลแนลสีของแต่ละพิกเซล
11) บรรทัดที่ 17 ตรวจสอบข้อความว่าสิ้นสุดหรือยัง
12) บรรทัดที่ 18 นำ 2 บิตสุดท้ายของข้อความอักษรนั้นๆ มาแทนที่ 2 บิตสุดท้ายของค่าสีของแต่ละพิกเซลสี
13) บรรทัดที่ 21 ตรวจสอบว่าตัวอักษรนั้นสิ้นสุดหรือยัง
14) บรรทัดที่ 24 ตรวจสอบว่าสิ้นสุดข้อความหรือยัง
15) บรรทัดที่ 27 ถ้าสิ้นสุดข้อความแล้วจะให้ใส่ค่าสีของภาพต้นฉบับลงไปเลย
16) บรรทัดที่ 30 เก็บค่าสีใหม่ลงในตัวแปรอาเรย์ผลลัพธ์
17) บรรทัดที่ 31 ตรวจสอบการทำงานเพื่อแสดงแถบการทำงาน

```
1  oneDPix[] คือตัวแปรอาเรย์ที่เก็บ ค่าสี rgb ของแต่ละพิกเซลเป็นชนิด Integer
2  imgCols คือความกว้างของรูปภาพ
3  imgRows คือความสูงของรูปภาพ
4  str[] คือข้อความที่ต้องการซ่อน
5
6  int[] binary = { 16, 8, 0 }; // กำหนดค่าเริ่มต้นสำหรับการ Shift bit
7  int[] toShift = { 6, 4, 2, 0 }; // กำหนดค่าเริ่มต้นสำหรับการ Shift bit
8  byte[] msg = str.getBytes(); // เปลี่ยนข้อความที่จะซ่อนให้เป็นไบต์อาเรย์
9  int channels = 3; // กำหนดแชลแนลเริ่มต้นสำหรับวนลูป 3 สี
10 int shiftIndex = 4; // กำหนดค่าเริ่มต้นสำหรับเช็คตัวอักษร
11 byte[] result = new byte[imgRows * imgCols * channels]; // สร้างตัวแปรอาเรย์มารอรับค่าผลลัพธ์
12 for (int row = 0; row < imgRows; row++) { // วนลูบตั้งแต่ 0 – แถวสุดท้าย
13    for (int col = 0; col < imgCols; col++) { // วนลูบตั้งแต่ 0 – คอลัมน์สุดท้าย
14       int element = row * imgCols + col; // สร้างตัวแปรมาเก็บตำแหน่งของอาเรย์พิกเซล
15       byte tmp = 0;
16       for (int channelIndex = 0; channelIndex < channels; channelIndex++) { // วนลูบแชลแนลสีในแต่ละพิกเซล
17          if (!msgEnded) { // ตรวจสอบว่าข้อความสิ้นสุดหรือยัง
18          tmp = (byte) ((((oneDPix[element] >> binary[channelIndex]) & 0xFF) & 0xFC) |
19             ((msg[msgIndex] >> toShift[(shiftIndex++) % toShift.length]) & 0x3));
20          // นำ 2 บิตสุดท้ายของข้อความอักษรนั้นๆ มาแทนที่ 2 บิตสุดท้ายของค่าสีของแต่ละพิกเซล
21             if (shiftIndex % toShift.length == 0) { // ตรวจสอบว่าสิ้นสุดตัวอักษรหรือยัง
22                msgIndex++;
23             }
24             if (msgIndex == msg.length) { // ตรวจสอบว่าสิ้นสุดข้อความหรือยัง
25                msgEnded = true;
26             }
27          } else { // ถ้าสิ้นสุดข้อความที่จะซ่อนแล้วให้ใส่ค่าสีของแต่ละพิกเซลของรูปภาพต้นฉบับลงไปเลย
28                tmp = (byte) ((((oneDPix[element] >> binary[channelIndex]) & 0xFF)));
29             }
30             result[resultIndex++] = tmp; // เก็บค่าสีใหม่ลงในตัวแปรอาเรย์ผลลัพธ์
31             if (hand != null) // ตรวจสอบการทำงานเพื่อแสดงแถบการทำงาน
32             hand.increment(1);
33          }
34       }
35    }
36
```

**ภาพที่ 3.5** โปรแกรมหลักในส่วนของการอำพรางข้อความลงไปยังรูปภาพ

1.1.2 ขั้นตอนการถอดข้อมูลออกจากไฟล์ภาพ ซึ่งเป็นขั้นตอนของทางผู้รับ

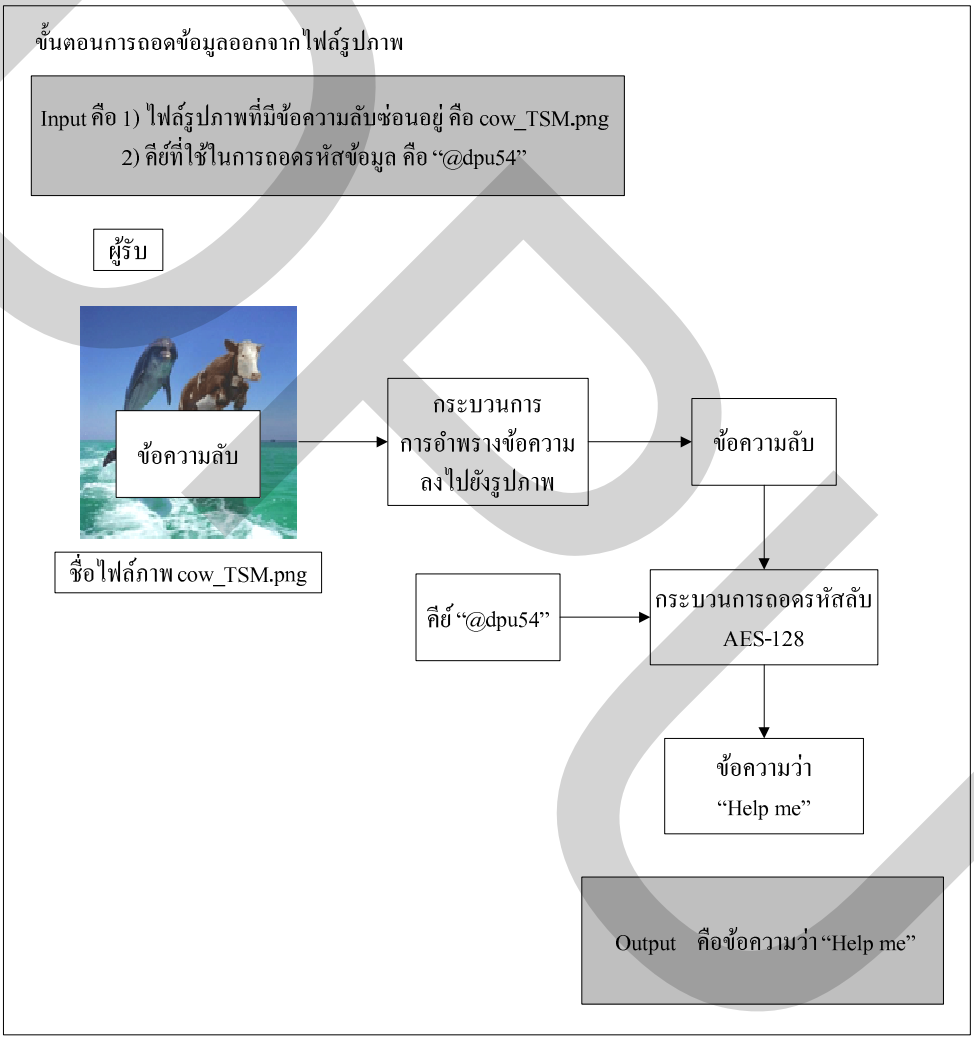
จากภาพที่ 3.6 แสดงถึงขั้นตอนของการถอดข้อความที่ผู้ส่งต้องการส่งให้กับผู้รับออกจากไฟล์ภาพที่อยู่ในรูปแบบ PNG ซึ่งอธิบายได้ดังนี้

1) Input ประกอบด้วย
   1.1) ไฟล์รูปภาพที่มีข้อความลับที่ผู้ส่งได้ซ่อนไว้คือไฟล์ภาพชื่อ cow_TSM.png
   1.2) คีย์ที่ผู้ส่งต้องการใช้ในกระบวนการถอดรหัสลับคือ “@dpu54”
2) Output คือ ข้อความที่ผู้ส่งต้องการส่งให้กับผู้รับ คือ ข้อความว่า “Help me”
3) Process มีขั้นตอนดังนี้

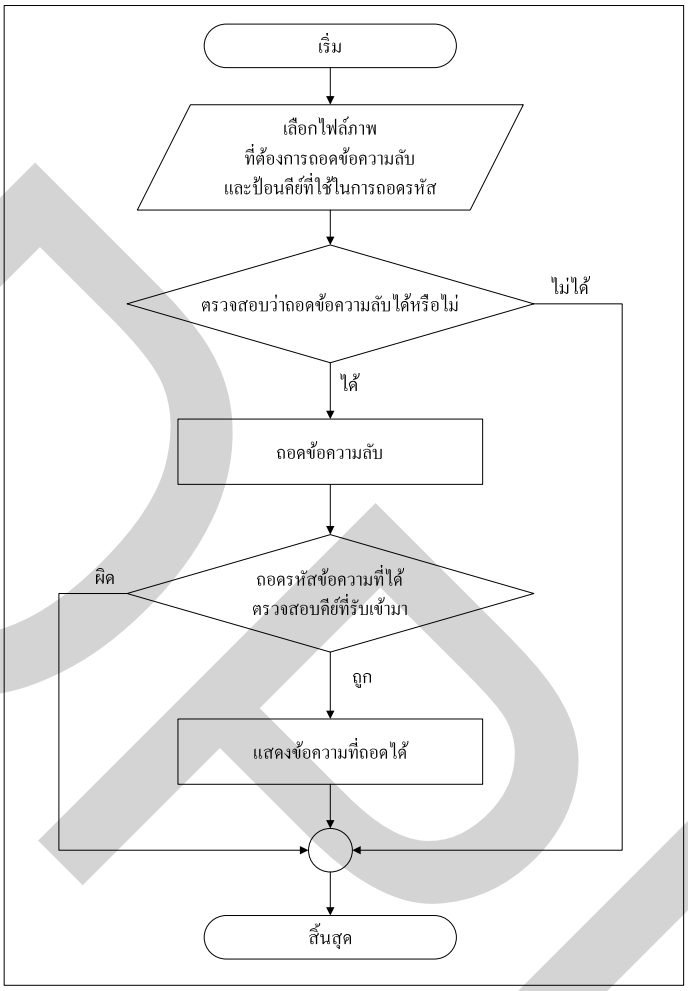
3.1) ระบบจะรับข้อมูลไฟล์ภาพ คือไฟล์ภาพชื่อ stego_cow.png และคีย์ที่ผู้รับต้องใช้ในการถอดรหัสข้อมูล

3.2) ถอดข้อความออกจากไฟล์ภาพ ซึ่งจะได้ข้อความออกมาเป็นข้อความลับ

3.3) ระบบจะนำข้อความลับมาผ่านกระบวนการถอดรหัสโดยใช้คีย์ที่รับเข้ามาซึ่งจะได้ข้อความว่า “Help me”




**ภาพที่ 3.6** ขั้นตอนการถอดข้อความออกจากไฟล์รูปภาพ




**ภาพที่ 3.7** รายละเอียดการทำงานของการถอดข้อความลับออกจากรูปภาพและถอดรหัสข้อความ

จากภาพที่ 3.7 แสดงรายละเอียดการทำงานของการถอดข้อมูลความลับออกจากรูปภาพและถอดรหัสข้อความ โดยขั้นตอนการทำงานสามารถอธิบายได้ดังนี้

1) ผู้ใช้ทำการเลือกไฟล์รูปภาพที่ต้องการถอดข้อความลับออกมา จากนั้นป้อนคีย์เพื่อใช้ในการถอดรหัสข้อความ

2) แอพพลิเคชั่นจะทำการตรวจสอบว่าไฟล์รูปภาพนั้นมีข้อความลับที่ถูกซ่อนไว้ด้วยแอพพลิเคชั่นนี้หรือไม่

3) ถ้ามีข้อความลับซ่อนอยู่ แอพพลิเคชั่นจะทำการถอดข้อความลับนั้นออกมา

4) ทำการถอดรหัสลับข้อความด้วยคีย์ที่ผู้ใช้ป้อนเข้ามา

5) ถ้าคีย์ที่ผู้ใช้ป้อนเข้ามาถูกต้อง จะแสดงข้อความที่ถูกซ่อนไว้ในไฟล์รูปภาพนั้น

โปรแกรมหลักในส่วนของขั้นตอนวิธีการถอดบิตข้อความออกจากรูปภาพแสดงดังภาพที่ 3.8 โดยโปรแกรมในส่วนนี้จะทำหน้าที่ดึงบิตที่มีนัยสำคัญต่ำสุด 2 บิตสุดท้ายของแต่ละพิกเซลสีออกมาประกอบเป็นข้อความที่ถูกซ่อนอยู่ภายใน ซึ่งจะต้องนำไปผ่านกระบวนการถอดรหัสลับด้วยคีย์ที่ผู้ใช้ป้อนเข้ามาอีกครั้ง ถึงจะสามารถแสดงผลข้อความนั้นออกมาได้ ซึ่งในแต่ละบรรทัดมีขั้นตอนการทำงานต่างๆ ดังนี้

1) บรรทัดที่ 5 กำหนดค่าเริ่มต้นให้กับตัวแปรอาเรย์ เพื่อใช้ในการ And bit ของรูปภาพ
2) บรรทัดที่ 6 กำหนดค่าเริ่มต้นให้กับตัวแปรอาเรย์เพื่อใช้ในการ Shift bit ของข้อความ
3) บรรทัดที่ 7 กำหนดค่าเริ่มต้นเพื่อใช้ตรวจสอบจุดเริ่มต้นของข้อความ
4) บรรทัดที่ 8 กำหนดค่าเริ่มต้นเพื่อใช้ตรวจสอบจุดสิ้นสุดของข้อความ
5) บรรทัดที่ 10 สร้างตัวแปรมารอรับค่าข้อความที่ถอดออกมา
6) บรรทัดที่ 11 กำหนดค่าเริ่มต้นสำหรับตรวจสอบตัวอักษร
7) บรรทัดที่ 13 วนลูบอาเรย์ที่เก็บค่าสีตั้งแต่ อาเรย์ตัวแรกถึงอาเรย์ตัวสุดท้าย
8) บรรทัดที่ 14 ดึงบิต 2 หลักสุดท้ายของค่าสีของแต่ละพิกเซลสีออกมาเก็บไว้ในตัวแปรเก็บค่าชั่วคราว
9) บรรทัดที่ 16 ตรวจสอบว่าดึงค่าออกมาครบ 1 ตัวอักษรหรือยัง
10) บรรทัดที่ 20 ตรวจสอบว่าข้อความตรงกับสัญลักษณ์สิ้นสุดข้อความหรือไม่
11) บรรทัดที่ 24 ตรวจสอบขนาดข้อความว่าถูกต้องหรือไม่

```
oneDPix[] คือตัวแปรอาเรย์ที่เก็บ ค่าสี rgb ของแต่ละพิกเซลเป็นชนิดไบต์
imgCols คือความกว้างของรูปภาพ
imgRows คือความสูงของรูปภาพ

byte[] andByte = { (byte) 0xC0, 0x30, 0x0C, 0x03 };
int[] toShift = { 6, 4, 2, 0 };
String END_MESSAGE_COSTANT = "#!@";
String START_MESSAGE_COSTANT = "@!#";
Vector<Byte> v = new Vector<Byte>();
String builder = ""; // สร้างตัวแปรสตริงมารอรับข้อความที่จะถอดออกมา
int shiftIndex = 4; // กำหนดค่าเริ่มต้นสำหรับเช็คตัวอักษร
byte tmp = 0x00; // กำหนดค่าตัวแปรไบต์เก็บค่าชั่วคราวเริ่มต้น
for (int i = 0; i < oneDPix.length; i++) { // วนลูปตั้งแต่ 0 – อาเรย์ตัวสุดท้าย
tmp = (byte) (tmp | ((oneDPix[i] << toShift[shiftIndex % toShift.length]) & andByte[shiftIndex++ % toShift.length]));
// ดึงบิต 2 หลักสุดท้ายของค่าสีของแต่ละพิกเซลออกมาเก็บไว้ในตัวแปรเก็บค่าชั่วคราว
  if (shiftIndex % toShift.length == 0) { // ตรวจสอบว่าดึงค่าออกมาได้ครบ 1 ตัวอักษรหรือยัง
     v.addElement(new Byte(tmp));
     byte[] nonso = { (v.elementAt(v.size() - 1)).byteValue() };
     String str = new String(nonso);
if (builder.endsWith(END_MESSAGE_COSTANT)) { // ตรวจสอบว่าข้อความตรงกับสัญลักษณ์สิ้นสุดข้อความหรือยัง
        break;
     } else {
        builder = builder + str;
if (builder.length() == START_MESSAGE_COSTANT.length() && !START_MESSAGE_COSTANT.equals(builder)) {
  // ตรวจสอบขนาดข้อความว่าถูกต้องหรือไม่
           builder = null;
           break;
        }
     }
     tmp = 0x00; // เคลียร์ค่าตัวแปรเก็บข้อมูลตัวอักษรชั่วคราว
  }
}
if (builder != null)
  builder = builder.substring(START_MESSAGE_COSTANT.length(), builder.length()
  - END_MESSAGE_COSTANT.length());
     return builder;
```

**ภาพที่ 3.8** โปรแกรมหลักในส่วนของการถอดข้อความอักษรออกมาจากรูปภาพ

# บทที่ 4
# ผลการศึกษา

## 4.1 ขั้นตอนการทำงานของระบบเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ

จากการดำเนินการพัฒนาแอพพลิเคชั่นสำหรับการอำพรางข้อมูลในรูปแบบข้อความตัวอักษร ลงไปยังข้อมูลที่เป็นไฟล์ภาพดิจิตอลที่มีการบีบอัดแบบ lossless โดยเข้ารหัสข้อมูลด้วยมาตรฐานการเข้ารหัสลับขั้นสูงบนอุปกรณ์โทรศัพท์เคลื่อนที่ ที่ใช้ระบบปฏิบัติการแอนดรอยด์ เมื่อทำการพัฒนาแอพพลิเคชั่นเสร็จเรียบร้อยแล้ว สามารถแบ่งการทำงานของแอพพลิเคชั่นออกเป็น 2 ส่วนคือ ส่วนของการเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ และส่วนของการถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1.1 ไอคอน

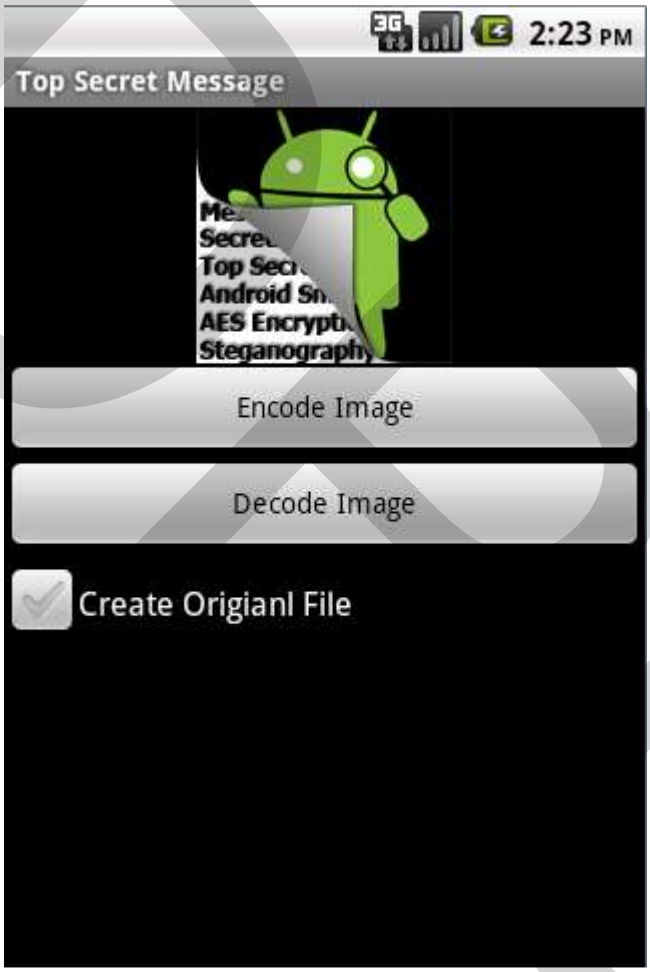
ผู้ใช้สามารถเข้าใช้งานแอพพลิเคชั่นได้จากไอคอนของแอพพลิเคชั่น ดังภาพ


**ภาพที่ 4.1** ไอคอนที่ใช้ในการเข้าสู่แอพพลิเคชั่น

เมื่อผู้ใช้เข้าสู่แอพพลิเคชั่น จะพบกับหน้าจอหลักของแอพพลิเคชั่น

4.1.2 หน้าจอหลักของแอพพลิเคชั่น มีเมนูต่างๆ ดังนี้

1) ปุ่มสำหรับไปยังหน้าจอเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ
2) ปุ่มสำหรับไปยังหน้าจอถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ
3) ช่องสำหรับเลือกว่าจะสร้างไฟล์ต้นฉบับใหม่หรือไม่



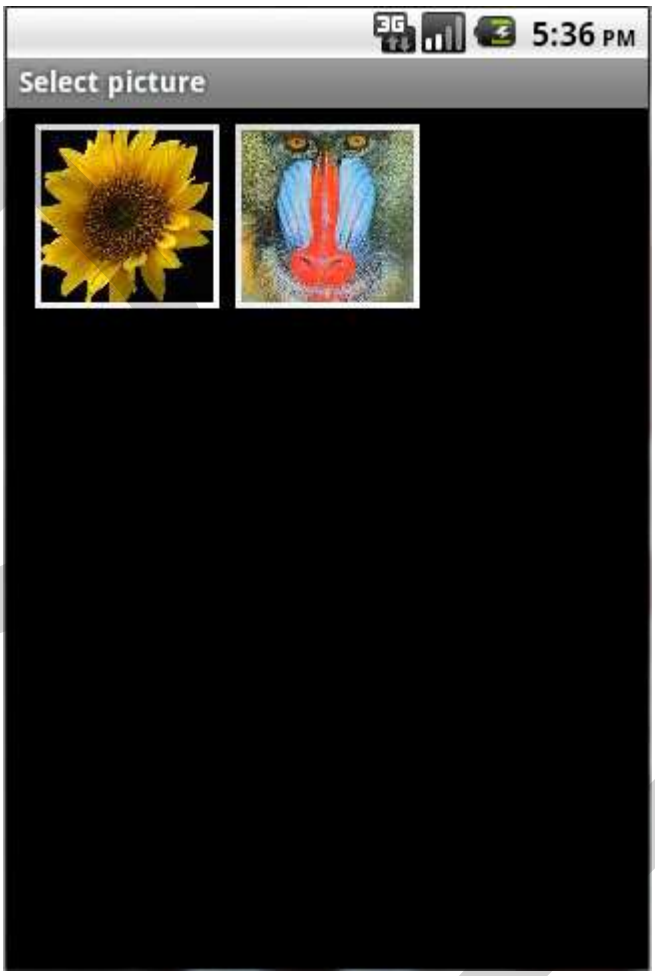

**ภาพที่ 4.2** หน้าจอหลักของแอพพลิเคชั่น

4.1.3 หน้าจอสำหรับเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ ประกอบด้วยส่วนต่างๆ ดังนี้

1) ช่องแสดงที่อยู่ของไฟล์ภาพ
2) ปุ่มสำหรับเลือกภาพที่อยู่ในเครื่อง
3) ช่องสำหรับกรอกค่าคีย์ที่จะใช้ในการเข้ารหัสข้อมูล
4) ช่องสำหรับป้องข้อความที่ต้องการซ่อนลงไปในรูปภาพ


**ภาพที่ 4.3** หน้าจอสำหรับเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ

เมื่อแตะที่ปุ่มเลือกรูปภาพ จะแสดงหน้าจอสำหรับเลือกรูปภาพ เพื่อนำมาใช้ในการซ่อนข้อความลงไป ดังภาพที่ 4.4 โดยให้ผู้ใช้แตะที่ภาพที่ต้องการ เพื่อนำภาพดังกล่าวไปใช้




**ภาพที่ 4.4** หน้าจอสำหรับเลือกรูปภาพที่ต้องการใช้ในการซ่อนข้อความ

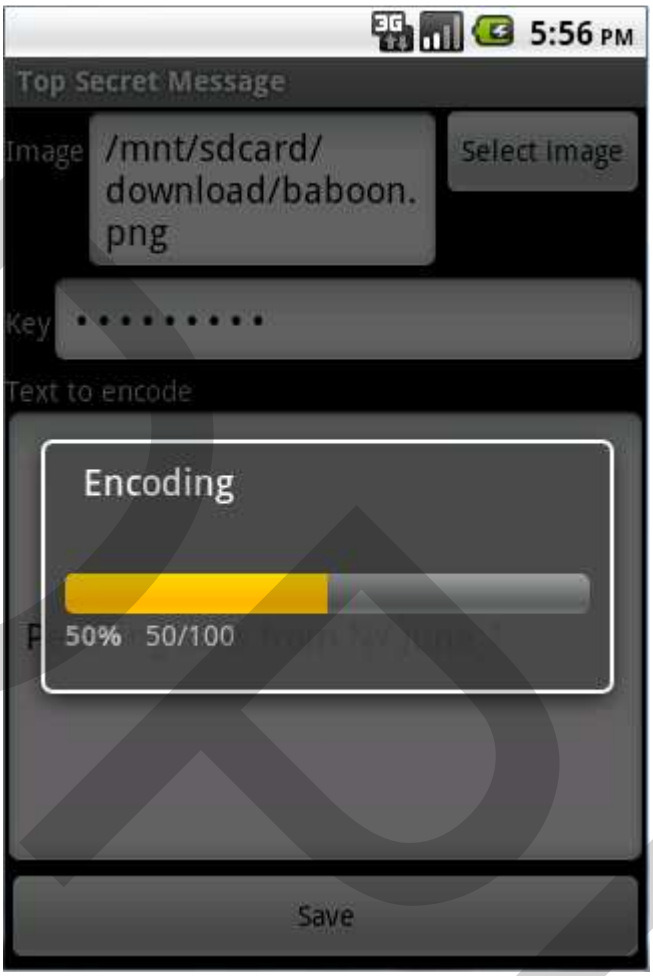
หลังจากที่เลือกรูปภาพแล้วที่ช่อง Image จะแสดงที่อยู่และชื่อของรูปภาพที่ผู้ใช้ได้เลือกมา ดังรูปที่ 4.5


**ภาพที่ 4.5** หน้าจอสำหรับเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ หลังจากเลือกรูปภาพที่ต้องการนำมาใช้เรียบร้อยแล้ว

จากนั้นให้ทำการป้อนคีย์ที่ต้องการ ลงไปยังช่อง Key และป้อนข้อความที่ต้องการจะซ่อนลงไปยังช่อง Text to encode ดังภาพที่ 4.6


**ภาพที่ 4.6** หน้าจอสำหรับเข้ารหัสและอำพรางข้อความลงในไฟล์ภาพ หลังจากป้อนคีย์ที่ต้องการใช้ในการเข้ารหัส และป้อนข้อความที่ต้องการซ่อนแล้ว

จากนั้นให้ทำให้แตะที่ปุ่ม Save แอพพลิเคชั่นจะเริ่มทำงาน โดยการนำข้อความกับคีย์ที่ป้อนเข้ามา ไปทำการเข้ารหัสก่อนด้วย ขั้นตอนวิธี AES แบบคีย์ขนาด 128 บิต เมื่อเข้ารหัสข้อความด้วยคีย์ดังกล่าวเรียบร้อยแล้ว จะนำข้อความที่ผ่านการเข้ารหัสแล้วไปซ่อนลงในภาพที่ได้เลือกไว้ ระหว่างที่แอพพลิเคชั่นกำลังทำงานอยู่นั้น หน้าจอโปรแกรมจะแสดงแถบผลการทำงานบอกเป็นเปอร์เซ็นต์ ดังภาพที่ 4.7



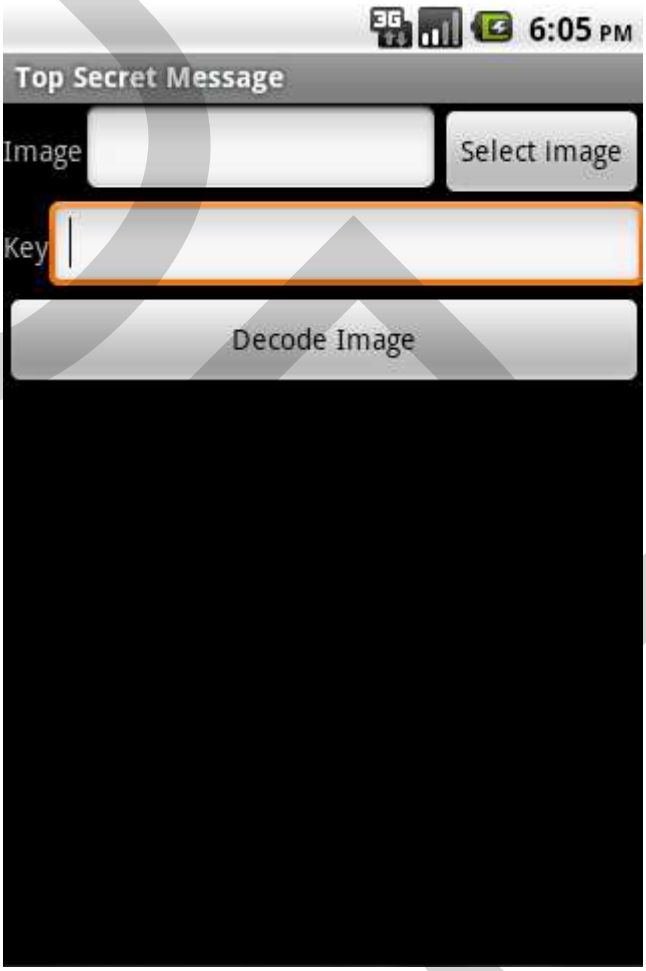

**ภาพที่ 4.7** หน้าจอระหว่างการทำงานของแอพพลิเคชั่น ซึ่งอยู่ในขั้นตอนของการเข้ารหัสข้อความและนำไปซ่อนลงในรูปภาพ

เมื่อแอพพลิเคชั่นทำการเข้ารหัสลับด้วยคีย์ที่ป้อน และซ่อนข้อความลงไปยังภาพเสร็จเรียบร้อยแล้ว จะมีข้อความแสดงขึ้นมาบอกว่ารูปภาพใหม่ได้ถูกบันทึกแล้ว ดังภาพที่ 4.8 โดยเมื่อแตะที่ปุ่ม ตกลง ก็จะเป็นการกลับสู่หน้าจอหลักของแอพพลิเคชั่น


**ภาพที่ 4.8** หน้าจอเมื่อแอพพลิเคชั่นทำการเข้ารหัสลับด้วยคีย์ที่ป้อน และซ่อนข้อความลงไปยังภาพเสร็จเรียบร้อยแล้ว

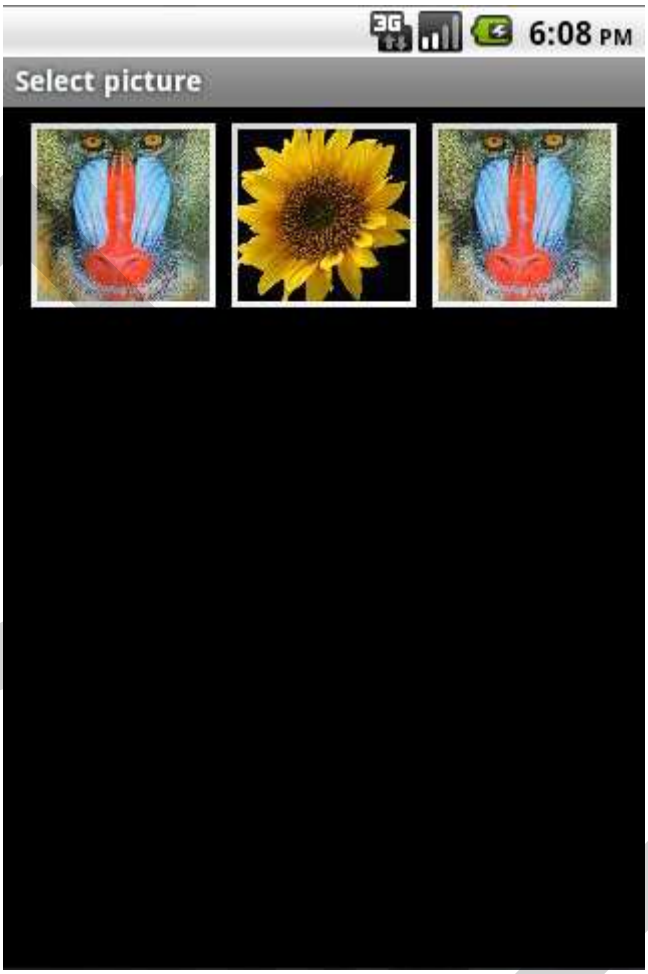
4.1.4 หน้าจอสำหรับถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ

1) ช่องแสดงที่อยู่ของไฟล์ภาพ
2) ปุ่มสำหรับเลือกภาพที่อยู่ในเครื่อง
3) ช่องสำหรับกรอกค่าคีย์ที่จะใช้ในการถอดรหัสข้อมูล



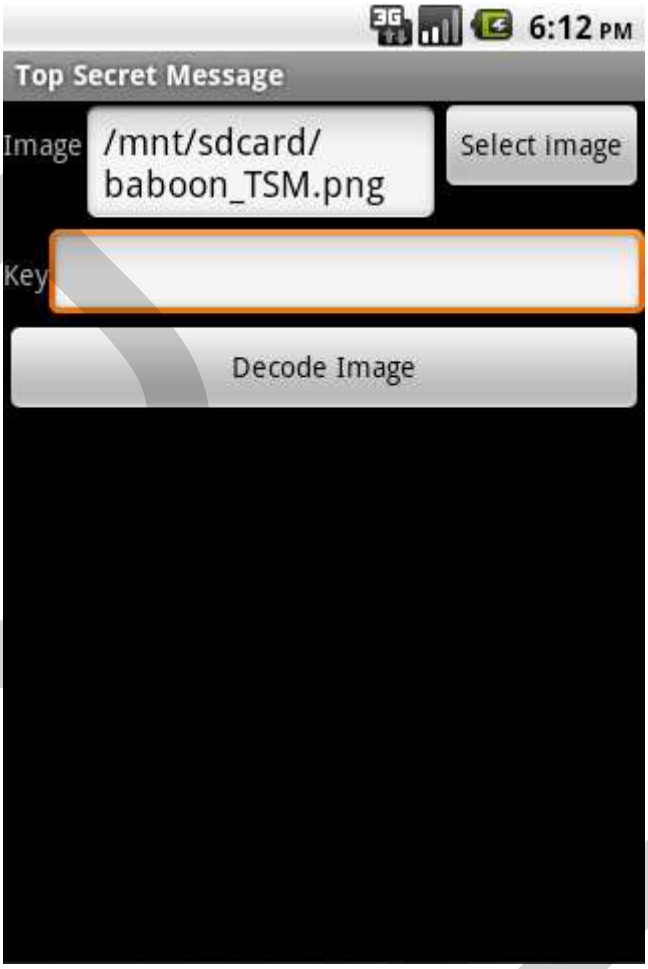

**ภาพที่ 4.9** หน้าจอสำหรับถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ

เมื่อแตะที่ปุ่มเลือกรูปภาพ จะแสดงหน้าจอสำหรับเลือกรูปภาพ เพื่อนำมาใช้ในการถอดข้อความออกจากรูปภาพ ดังภาพที่ 4.10 โดยให้ผู้ใช้แตะที่ภาพที่ต้องการ



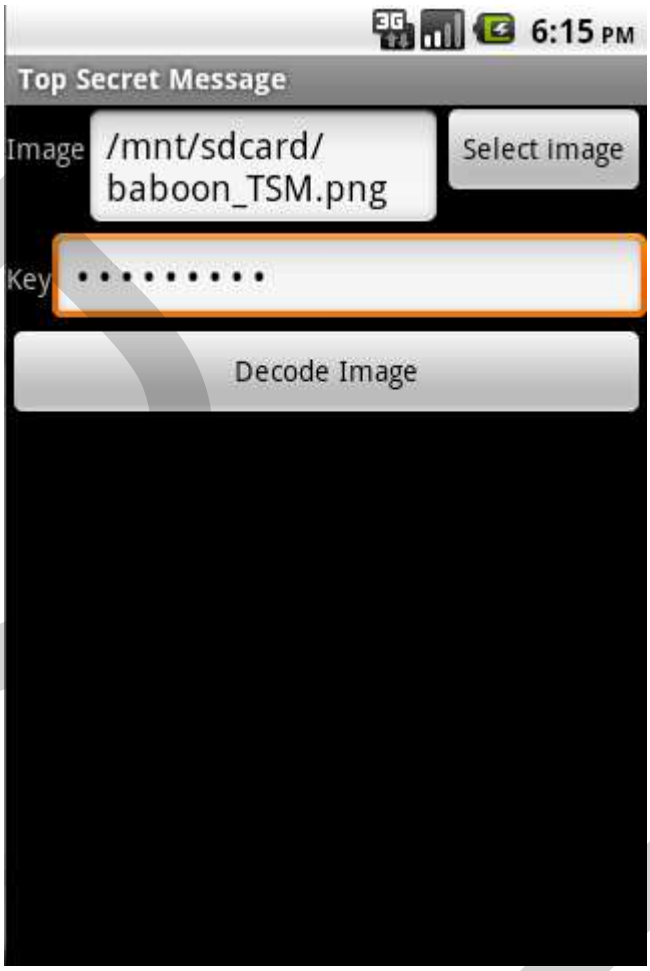

**ภาพที่ 4.10** หน้าจอสำหรับเลือกรูปภาพเพื่อนำมาใช้ในการถอดข้อความออกรูปจากภาพ

หลังจากที่เลือกรูปภาพแล้วที่ช่อง Image จะแสดงที่อยู่และชื่อของรูปภาพที่ผู้ใช้ได้เลือกมา ดังรูปที่ 4.11 ซึ่งวิธีการสังเกตว่ารูปภาพไหนได้ผ่านการเข้ารหัสและซ่อนข้อความด้วยแอพพลิเคชั่นนี้ ชื่อของรูปภาพจะมีข้อความ “_TSM” ต่อท้ายชื่อของรูปภาพนั้นๆ




**ภาพที่ 4.11** หน้าจอสำหรับถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ หลังจากเลือกรูปภาพที่ต้องการนำมาใช้ในการถอดข้อความเรียบร้อยแล้ว

จากนั้นให้ทำการป้อนคีย์ที่ใช้ในการเข้ารหัสข้อความในรูปภาพนั้นๆ ดังภาพที่ 4.12




**ภาพที่ 4.12** หน้าจอสำหรับถอดข้อมูลที่ซ่อนอยู่ออกจากไฟล์ภาพ หลังจากที่ได้ป้อนคีย์ที่ใช้ในการถอดรหัสข้อความในภาพเรียบบร้อยแล้ว

จากนั้นให้ทำให้แตะที่ปุ่ม Decode Image แอพพลิเคชั่นจะเริ่มทำงาน โดยการถอดข้อความที่ถูกเข้ารหัสอยู่นั้นออกมาจากรูปภาพ เมื่อได้ข้อความแล้วแอพพลิเคชั่นจะนำข้อความนั้นไปถอดรหัสด้วยคีย์ที่ผู้ใช้ป้อน เมื่อแอพพลิเคชั่นถอดรหัสข้อความเรียบร้อยแล้วจะแสดงข้อความที่ถูกซ่อนอยู่ภายในรูปภาพออกมา ดังภาพที่ 4.13


**ภาพที่ 4.13** หน้าจอผลลัพธ์ ซึ่งมีช่องแสดงข้อความที่ถูกซ่อนไว้ในรูปภาพ

**4.2 ผลการวัดประสิทธิภาพของแอพพลิเคชั่น**

จากการทดสอบการใช้งานแอพพลิเคชั่นในการอำพรางข้อความลงในรูปภาพนั้น จะเห็นว่าสามารถซ่อนข้อความได้เป็นจำนวนมากโดยไม่ทำให้ภาพเกิดความเสียหายมากนัก โดยปริมาณข้อความที่สามารถซ่อนได้นั้นขึ้นอยู่กับค่าความละเอียดของภาพ และความแตกต่างของภาพต้นฉบับกับภาพที่ผ่านกระบวนการการอำพรางข้อความลงไปแล้วนั้นไม่สามารถสังเกตเห็นความแตกต่างได้ชัดเจน ดังภาพตัวอย่างต่อไปนี้

rabbit_ORI.png rabbit_TSM.png

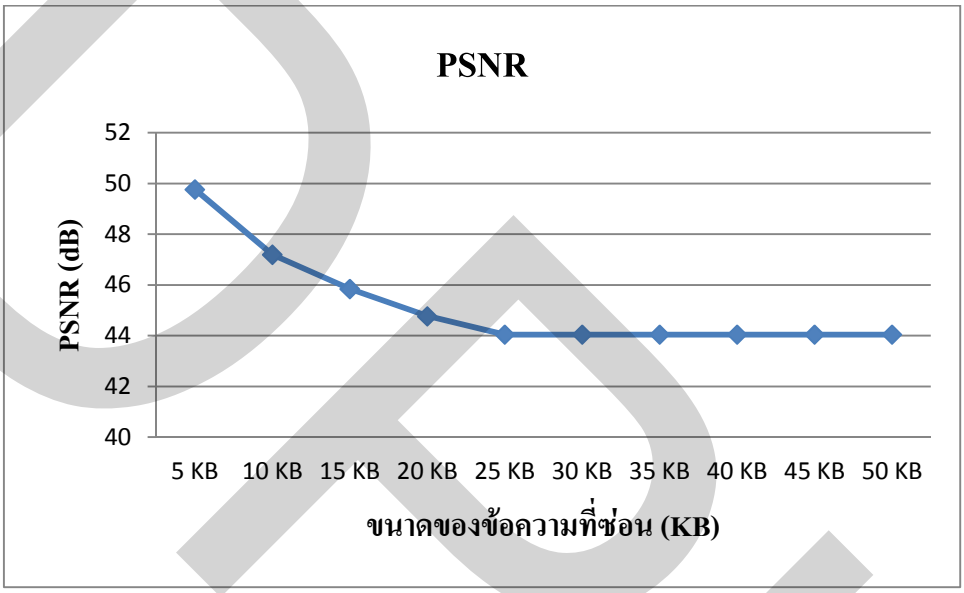
**ภาพที่ 4.14** การเปรียบเทียบผลลัพธ์ภาพชื่อ rabbit_ORI.png ก่อนและหลังอำพรางข้อความ

lena_ORI.png lena_TSM.png

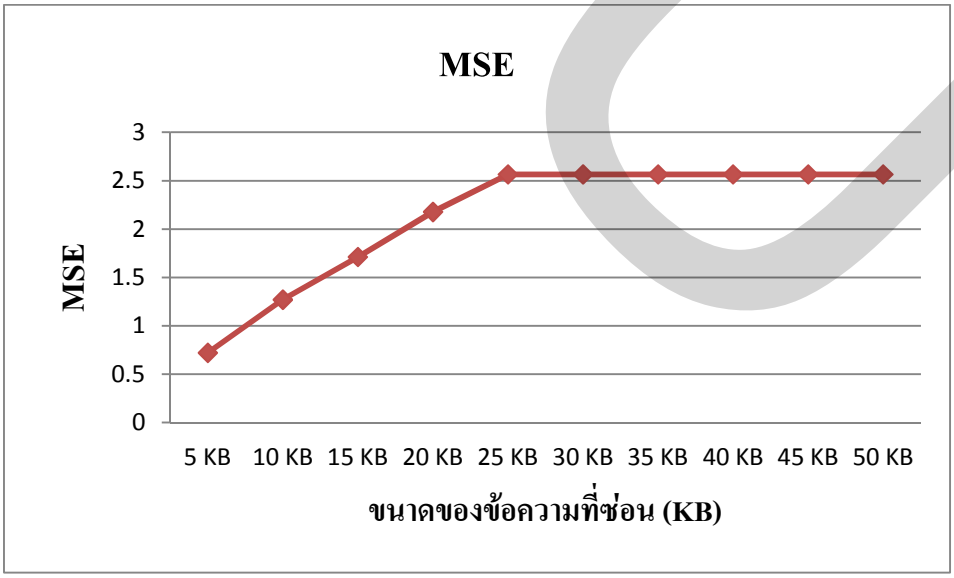
**ภาพที่ 4.15** การเปรียบเทียบผลลัพธ์ภาพชื่อ lena_ORI.png ก่อนและหลังอำพรางข้อความ

จะสังเกตเห็นได้ว่ารูปภาพตัวอย่างทั้ง 2 รูป ที่นำมาใช้ในการอำพรางข้อความนั้น ผลลัพธ์ของรูปภาพที่ได้หลังจากการอำพรางข้อความก็ดูไม่แตกต่างกันมากนักหากสังเกตด้วยสายตาของมนุษย์ นอกจากนี้ผู้วิจัยได้นำวิธีการวัดประสิทธิภาพแบบ PSNR และ MSE มาใช้เพื่อ

ประเมินประสิทธิภาพของแอพพลิเคชั่นที่พัฒนาขึ้น โดยทดสอบด้วยการซ่อนข้อความที่มีขนาดต่างกันลงไปยังไฟล์รูปภาพชื่อ rabbit_ORI.png ขนาดของไฟล์รูปภาพอยู่ที่ 256*256 พิกเซล และขนาดข้อมูลของไฟล์ภาพอยู่ที่ 105 KB โดยผลการเปรียบเทียบไฟล์รูปภาพที่ผ่านกระบวนการอำพรางข้อความกับไฟล์รูปภาพต้นฉบับมีดังต่อไปนี้




**ภาพที่ 4.16** ผลการวัดประสิทธิภาพในการอำพรางข้อความด้วย PSNR




**ภาพที่ 4.17** ผลการวัดประสิทธิภาพในการอำพรางข้อความด้วย MSE

จากภาพที่ 4.16 และ 4.17 แสดงให้เห็นถึงการเปรียบเทียบประสิทธิภาพของการอำพรางข้อความในรูปภาพเดียวกันคือ rabbit_ORI.png ซึ่งมีขนาดของรูปภาพคือ 256*256 พิกเซล โดยการอำพรางข้อความแต่ละครั้งจะใช้ขนาดของข้อความที่แตกต่างกัน จะสังเกตได้ว่ายิ่งข้อความมีขนาดใหญ่มากเท่าไหร่ ความแนบเนียนในการอำพรางข้อความก็จะลดลง โดยสังเกตได้จากคุณภาพของภาพจากการหาอัตราส่วนของสัญญาณรบกวนสูงสุด (PSNR) ซึ่งเป็นค่ามาตรฐานที่บ่งบอกถึงคุณภาพที่เปลี่ยนระหว่างรูปภาพสองภาพ ดังแสดงในภาพที่ 4.16 โดยจะสังเกตได้ว่าการอำพรางข้อความที่มีขนาดใหญ่มากขึ้นทำให้ค่า PSNR ลดลงโดยค่า PSNR ที่สูงจะชี้ให้เห็นถึงคุณภาพของรูปภาพที่ใกล้เคียงกับภาพต้นฉบับ โดยค่าที่สามารถยอมรับได้จะอยู่ที่ 20-40 dB ซึ่งจะไม่เกิดความแตกต่างหากสังเกตด้วยสายตาของมนุษย์ และจะสังเกตได้ว่าค่า PSNR จะหยุดนิ่งเมื่อซ่อนข้อความที่มีขนาด 25KB เป็นต้นไป เนื่องมาจากข้อจำกัดของการอำพรางข้อความนั้นสามารถอำพรางข้อความลงไปยังรูปภาพได้เพียง 6 บิต ต่อ 1 พิกเซล ซึ่งควรจะซ่อนข้อความที่ขนาดความยาวได้เท่ากับ 49 KB แต่เนื่องจากข้อความจะถูกนำไปผ่านกระบวนการเข้ารหัสด้วยคีย์ที่ป้อนเข้าไป ทำให้ข้อความมีขนาดยาวขึ้นทำให้สามารถซ่อนข้อความลงไปยังรูปภาพที่ทดสอบได้เพียง 24 KB เท่านั้น หากความยาวเกินที่กำหนดข้อความที่ซ่อนจะถูกตัดออกทำให้ข้อความไม่สมบูรณ์ไม่สามารถถอดข้อความลับออกมาได้ เช่นเดียวกับค่าความผิดพลาดเฉลี่ยกำลังสอง (MSE) ดังแสดงในภาพที่ 4.17 โดยจะสังเกตได้ว่าการอำพรางข้อความที่มีขนาดใหญ่มากขึ้นทำให้ค่า MSE เพิ่มขึ้นโดยค่า MSE ที่น้อยจะชี้ให้เห็นถึงคุณภาพของรูปภาพที่ใกล้เคียงกับภาพต้นฉบับ ด้วยวิธีการเปรียบเทียบค่าผิดพลาดเฉลี่ยของแต่ละพิกเซลในภาพ โดยค่าผิดพลาดยิ่งน้อยก็แสดงให้เห็นว่าคุณภาพของภาพนั้นใกล้เคียงกับภาพต้นฉบับมาก

# บทที่ 5
# สรุปผลการศึกษา

## 5.1 สรุปผลการศึกษา

จากการทดสอบแอพพลิเคชั่นที่ใช้ในการอำพรางข้อความลงไปยังรูปภาพ ซึ่งใช้เทคนิคการอำพรางข้อมูลแบบ LSB ร่วมกับการเข้ารหัสข้อความด้วยขั้นตอนวิธีแบบ AES นั้น แอพพลิเคชั่นที่ได้พัฒนาขึ้นสามารถใช้งานกับไฟล์รูปภาพได้หลายชนิด และผลจากการวัดประสิทธิภาพด้วยวิธี PSNR และ MSE นั้นอยู่ในเกณฑ์ที่น่าพอใจ โดยแอพพลิเคชั่นสามารถเก็บข้อความที่ต้องการอำพรางได้เป็นจำนวนมากโดยไม่ทำให้รูปภาพเกิดความผิดเพี้ยนจนสามารถสังเกตด้วยสายตาของมนุษย์ได้ ซึ่งปริมาณข้อความที่สามารถจัดเก็บในรูปภาพนั้นขึ้นอยู่กับค่าความละเอียดของรูปภาพ

## 5.2 ปัญหาและอุปสรรค

ด้วยข้อจำกัดของหน่วยความจำบนเครื่องโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ และลักษณะการพัฒนาแอพพลิเคชั่น ทำให้แอพพลิเคชั่นที่พัฒนาขึ้นไม่สามารถใช้งานกับไฟล์รูปภาพที่มีความละเอียดสูงได้ อีกทั้งไฟล์ภาพที่ผ่านกระบวนการอำพรางข้อความจะถูกเปลี่ยนประเภทของไฟล์รูปภาพไปเป็น PNG ซึ่งทำให้สามารถเก็บข้อความได้เป็นจำนวนมาก แต่จะทำให้ขนาดของไฟล์รูปภาพใหญ่ขึ้น จึงทำให้ความแนบเนียนในการอำพรางข้อความลดลงเล็กน้อย

## 5.3 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับแอพพลิเคชั่นที่ใช้ในการอำพรางข้อความลงในรูปภาพดิจิตอลร่วมกับการเข้ารหัสลับด้วยขั้นตอนวิธีแบบ AES มีดังนี้

5.3.1 ควรมีการปรับปรุงให้สามารถใช้งานกับไฟล์รูปภาพที่มีขนาดใหญ่ได้

5.3.2 อาจพัฒนาให้ประเภทของไฟล์ภาพผลลัพธ์ออกมาเป็น JPG

5.3.3 เพิ่มฟังก์ชั่นในการส่งหรือแชร์ภาพไปยังเครือข่ายที่ใช้สำหรับเก็บข้อมูลหรือทางอีเมล์

# บรรณานุกรม

DPU

# บรรณานุกรม

**ภาษาไทย**

## หนังสือ

จตุชัย แพงจันทร์. (2550). *Master in security.* นนทบุรี: ไอดีซีฯ.

ธำรงรัตน์ อมรรักษา. (2551). *ความปลอดภัยของข้อมูลสำหรับการสื่อสารสื่อประสม : จากทฤษฎีสู่ปฏิบัติ.* กรุงเทพฯ: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.

## บทความ

มนธวัล พรรณวิเชียร, และเกียรติศักดิ์ โยชะนัง. (2553). *โปรแกรมประยุกต์การอำพรางข้อมูลลงภาพโดยใช้วิธีการสลับบิตร่วมกับการเข้ารหัสข้อมูลโดยใช้มาตรฐานการเข้ารหัสลับขั้นสูง.* Sixth National Conference on Computing and Information Technology, 692-697.

ธนาวัฒน์ เดชกำแหง, และเกียรติศักดิ์ โยชะนัง. (2553). *โปรแกรมประยุกต์การอำพรางข้อมูลลงไฟล์เสียงโดยใช้วิธีการสลับบิตร่วมกับการเข้ารหัสข้อมูล.* Seventh National Conference on Computing and Information Technology, 229-234.

ภัทรพงษ์ เรียบร้อยเจริญ, และเกียรติศักดิ์ โยชะนัง. (2553). *การพัฒนาแอพพลิเคชั่นโดยการอำพรางข้อมูลลงในภาพผ่านระบบบริการรับและส่งสารแบบสื่อประสม.* Seventh National Conference on Computing and Information Technology, 619-624.

จักริน สุขเพ็ง, และนริศร แสงคะนอง. (2553). *โปรแกรมแบบฝังตัวบนโทรศัพท์เคลื่อนที่เพื่อช่วยป้องกันการเปิดดูข้อมูลภาพโดยใช้วิธีการเข้ารหัสแบบ 64-Bit Block Cipher (Blowfish) และการพิสูจน์ตัวตนภาพโดยการซ่อนข้อความลับด้วยวิธีการสร้างลายน้ำดิจิตอลแบบไม่สามารถมองเห็นได้.* Seventh National Conference on Computing and Information Technology, 600-605.

ธมกร บุ้งจันทร์, และชัยพร เขมะภาตะพันธ์. (2555). *เทคนิคการอำพรางข้อความไว้ในไฟล์ภาพเจเพ็ก.* การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 34, 1093-1096

**ภาษาต่างประเทศ**

**ARTICLES**

T.Morkel, J.H.P. Eloff, & M.S.Olivier. (2005). *An Overview of Image Steganography.* Fifth Annual Information Security South Africa Conference

Dhobale Dhanashri D, Ratil Babaso S, & Patil Shubhangi H.Mms. (2010). *Steganography For Smartphone Devices.* Second International Conference on Computer Engineering and Technolog

Mohammad Shirali-Shahreza. (2007). *Steganography in MMS*. IEEE

**ELECTRONIC SOURCES**

Pasquale Paola, & Paolo Manzo (2006). MobiStego (beta). Retrieved September 2011. from http://mobistego.sourceforge.net/

Wikipedia. Steganography, Retrieved September 2011. from http://en.wikipedia.org/wiki/Steganography

LSB . Retrieved September 2011. from http://blog.pupasoft.com/2009/11/09/aeronzsteganography/

Color Palettes, Bit Depth, and Dithhering, Retrieved September 2011. from http://www.ou.edu/class/digitalmedia/articles/ColorPalettes_Dithering_BitDepth.html

Alan Sia (2007). Advanced Encryption Standard (AES). Retrieved September 2011. from http://imps.mcmaster.ca/courses/SE-4C03-07/wiki/siaa/se4c03_aes_wiki(7).html

## ประวัติผู้เขียน

| | |
|---|---|
| ชื่อ-นามสกุล | พงศ์ปณต ทรัพย์วิไล |
| ประวัติการศึกษา | 2550-2552 สำเร็จการศึกษาระดับปริญญาตรี<br>คณะเทคโนโลยีสารสนเทศ<br>สาขาเทคโนโลยีสารสนเทศ<br>วิทยาลัยนอร์ทกรุงเทพ |
| ตำแหน่งและสถานที่ทำงานปัจจุบัน | นักคอมพิวเตอร์โครงการ 2 ส่วนพัฒนาบริการมัลติมีเดีย<br>บริษัท ทีโอที จำกัด (มหาชน) |

DPU