สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

การใช้งานลายมือชื่อดิจิตอล

Digital Signature Implementation



นายรัตนชัย วิเศษแสง

นายอุกฤษฏ์ ธีระธนานนท์

เลขหมู่.....
เลขทะเบียน 46193
วัน, เดือน, ปี 20 มี.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ว.ท.ส.ร.ม.ด
ว.ท.ส.ร.ม.ด
ว.ท.ส.ร.ม.ด

การใช้งานลายมือชื่อดิจิตอล

Digital Signature Implementation

โดย

นายรัตนชัย วิเศษแสง

นายอุกฤษฎ์ ธีระธนานนท์

อาจารย์ที่ปรึกษา

อ.ธนา หงษ์สุวรรณ

อ.อัครเดช วัชระภูพงษ์



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การใช้งานลายมือชื่อดิจิตอล

Digital Signature Implementation

ผู้จัดทำ

1. นาย รัตนชัย วิเศษแสง รหัสประจำตัว 41014357
2. นาย อุกฤษฏ์ ธีระธนานนท์ รหัสประจำตัว 41014543



_____ อาจารย์ที่ปรึกษา

(อ.ธนา หงษ์สุวรรณ)

_____ อาจารย์ที่ปรึกษา

(อ.อัครเดช วัชระภูพงษ์)

การใช้งานลายมือชื่อดิจิตอล

นาย รัตนชัย วิเศษแสง 41014357
นาย อุกฤษฎ์ ธีระธนานนท์ 41014543
อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์ อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

ลายมือชื่อดิจิตอลเป็นลายมือชื่ออิเล็กทรอนิกส์ไม่ใช่ลายมือชื่อที่เขียนโดยคนคนหนึ่ง ซึ่งสามารถพิสูจน์บ่งชี้ผู้ส่งหรือผู้ลงนามเอกสารได้ ทั้งยังทำให้แน่ใจได้ว่าข้อความหรือเอกสารที่ส่งมาไม่ได้มีการเปลี่ยนแปลง และประโยชน์เพิ่มเติมก็คือง่ายในการส่ง ไม่สามารถบอกปัดหรือปฏิเสธได้ ไม่สามารถปลอมได้ และเป็นการประทับเวลาโดยอัตโนมัติ

ลายมือชื่อดิจิตอลใช้พื้นฐานของการเข้ารหัสแบบคีย์ต่าง ซึ่งผู้ใช้ทุกคนมีคีย์ส่วนตัว และคีย์สาธารณะที่กำหนดขึ้นโดยองค์กรพิสูจน์สิทธิ์ เมื่อผู้ใช้บุคคลนั้นได้มีการลงนามลายมือชื่อดิจิตอล ลำดับของตัวเลขชุดหนึ่งจะสร้างมาจากคีย์ส่วนตัวของบุคคลนั้น ประกอบกับข้อมูลที่ต้องการลงนาม ซึ่งลำดับของตัวเลขนี้เรียกว่าลายมือชื่อดิจิตอล ลายมือชื่อดิจิตอลสามารถบ่งบอกได้ว่าผู้ลงนามเป็นผู้ใดโดยดูจากใบรับรองสิทธิ์

จากโปรแกรมลายมือชื่อที่มีการเข้ารหัสเอกสารแบบอักษรได้เพียงอย่างเดียว ยังไม่สมบูรณ์พอสำหรับการเข้ารหัสเอกสาร จึงได้ทำการเพิ่มความสามารถของโปรแกรมให้สามารถเข้ารหัสเอกสารแบบรูปภาพได้

โครงงานนี้ได้นำหลักการของลายมือชื่อดิจิตอลมาใช้กับเอกสารที่จัดทำขึ้นจากโปรแกรมไมโครซอฟท์เวิร์ด เพื่อให้โปรแกรมไมโครซอฟท์เวิร์ดที่มีความสามารถในการลงลายมือชื่อดิจิตอล และตรวจสอบลายมือชื่อได้อยู่แล้ว ให้สามารถลงลายมือชื่อและตรวจสอบลายมือชื่อกับเอกสารที่มีรูปภาพอยู่ด้วยได้ โดยโครงงานนี้เป็นโปรแกรมที่เขียนด้วยวิชวลซีพลัสพลัส (Visual C++) ซึ่งสามารถทำงานร่วมกับโปรแกรมไมโครซอฟท์เวิร์ด 97 ได้

Digital Signature Implementation

Ratanachai Visetsaeng
Ukrit Teerathananon
Thana Hongsuwan Advisor
Akkradach Watcharapupong Advisor

Abstract

A digital signature is bit-stream rather than a written signature that can be used by someone to identity sender of a message or signer of a document. It can also be used to ensure that the original content of the message or document that has been conveyed is unchanged. Additional benefits to the use of a digital signature are that it is easily transportable, cannot be repudiated, cannot be imitated by someone else, and can be automatically time-stamped.

A digital signature is based on asymmetric cryptography where every user has a unique pair of private and public keys, duly certified by a trusted Certificate Authority. When he or she signs a transaction, a unique mathematical code is created with the help of his or her private key and the actual content of the transaction. This "signature", which is bound to the transaction, can identify the signer's identity by its relationship to the digital certificate.

Digital Signature with only text document is not enough. So that program add new feature about picture.

This project brings the adventage of digital signature to use with Microsoft word. It brings more functionality to Microsoft word to sign and verify digital signature. Program that created in this project is using Microsoft visual C++. It can cooperate with Microsoft word 97.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลสองท่านแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ ธนา หงษ์สุวรรณ และอาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณเพื่อน ๆทุกคนที่ให้คำแนะนำปรึกษา ช่วยเหลือ ให้กำลังใจ ช่วยแก้ไขปัญหา

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้คณะผู้จัดทำมีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูคณะผู้จัดทำมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ คณะผู้จัดทำขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นาย รัตนชัย วิเศษแสง

นาย อุกฤษฏ์ ธีระธนานนท์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ลายมือชื่อดิจิตอล	3
2.1 การทำงานของลายมือชื่อดิจิตอล	3
บทที่ 3 การเข้ารหัส	7
3.1 พื้นฐานการเข้ารหัสและการถอดรหัส	8
3.2 ความต้องการของระบบการเข้ารหัสแบบสมมาตร	9
3.3 ลักษณะที่สำคัญของการเข้ารหัสแบบสมมาตร	9
3.4 ลักษณะของบล๊อกไซเฟอร์	9
3.5 DES (Data Encryption Standard)	13
3.5.1 ประวัติและที่มาของ DES	13
3.5.2 รายละเอียดของ DES	13
3.5.3 อัลกอริทึมของการเข้ารหัสแบบ DES	13
3.6 ปัญหาและความปลอดภัยในการเข้ารหัสแบบสมมาตร	15
3.7 พื้นฐานการเข้ารหัสแบบไม่สมมาตร	15
3.8 ความสามารถของการเข้ารหัสแบบไม่สมมาตร	16
3.9 อาร์เอสเอ (RSA)	19
3.10 การคำนวณการเข้า – ถอดรหัสของ อาร์เอสเอ	19
3.11 บริการของการเข้ารหัสแบบไม่สมมาตร	20
บทที่ 4 คริปโตเอพีไอ	22
4.1 การติดต่อกับ ซีเอสพี	22

หน้า

4.2 คำศัพท์ต่าง ๆที่ควรรู้ 22
4.3 การเรียกใช้คริปโตเอพีไอ 25
4.4 การใช้คริปโตเอพีไอในการทำงานกับเอกสารสิทธิ์ 27
บทที่ 5 โอแอลอี (OLE) 29
5.1 โอแอลอี 29
5.2 อินเทอร์เฟซ 29
5.3 อินเทอร์เฟซ IUnknow 30
บทที่ 6 คลาส COleDataObject 32
6.1 คลาส ColeDataObject 32
6.2 ฟังก์ชันสำคัญในคลาส ColeDataObject 32
บทที่ 7 คลิปบอร์ด 34
7.1 วินโดวส์คลิบบอร์ด 34
7.2 คลิบบอร์ดฟอร์แมต 34
7.2.1 สแตนดาร์ดคลิบบอร์ดฟอร์แมต 34
7.2.2 รีจิสเตอร์ฟอร์แมต 36
7.2.3 ไพรเวทฟอร์แมต 36
7.3 การทำงานของคลิบบอร์ด 37
7.3.1 การส่งผ่านข้อมูลไปยังคลิบบอร์ด 37
7.3.2 การวางข้อมูลจากคลิบบอร์ด 37
7.3.3 คลิบบอร์ดเมสเสจ 38
7.3.4 คลิบบอร์ดวิวเวอร์ 38
บทที่ 8 การสร้างแถบเครื่องมือบนโปรแกรมไมโครซอฟท์เวิร์ด 39
บทที่ 9 การทดลองและผลการทดลอง 47
9.1 ความต้องการของระบบ 47
9.2 ระบบที่ใช้ทดสอบ 47
9.3 การทดสอบการรันโปรแกรมในครั้งแรก 47
9.4 การทดสอบการสร้างลายมือชื่อดิจิตอลใหม่ 48
9.5 การลบลายมือชื่อดิจิตอลที่ไม่ใช้ 51
9.6 การลงลายมือชื่อดิจิตอล 51
9.7 การตรวจสอบลายมือชื่อดิจิตอล 52
บทที่ 10 วิจารณ์และสรุป 54
บรรณานุกรม 55

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 วิธีการตรวจสอบความถูกต้องของข้อมูลแบบพื้นฐาน 3
รูปที่ 2.2 เมื่อข้อมูลเกิดการผิดพลาด 4
รูปที่ 2.3 เมื่อมีผู้อ้างตัวเป็นผู้ส่ง 4
รูปที่ 2.4 วิธีการทำลายมือชื่อดิจิตอลโดยใช้แฮชฟังก์ชันและคีย์ต่าง 5
รูปที่ 2.5 ในกรณีมีผู้แอบอ้าง 6
รูปที่ 3.1 แสดงการเข้ารหัส ถอดรหัส 7
รูปที่ 3.2 แสดงการเข้ารหัสและถอดรหัสแบบสมมาตร 8
รูปที่ 3.3 แสดงการเข้ารหัสแบบอิเล็กทรอนิกส์โค้ดบุ๊ก 9
รูปที่ 3.4 แสดงการเข้ารหัสแบบไซเฟอร์บล๊อกเชนนิ่ง 10
รูปที่ 3.5 แสดงการเข้ารหัสแบบไซเฟอร์ฟีดแบ๊ก 11
รูปที่ 3.6 แสดงการเข้ารหัสแบบเอาท์พุตฟีดแบ๊ก 12
รูปที่ 3.7 แสดงการแบ่งคีย์ 56 บิต ที่เรียงลำดับบิตแล้วออกเป็น 2 ส่วน 14
รูปที่ 3.8 แสดงบล๊อกข้อมูล 64 บิต ที่แบ่งออกเป็น 2 ส่วนหลังจากการจัดทำเรียงบิตแล้ว 14
รูปที่ 3.9 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพื้นฐาน 15
รูปที่ 3.10 แสดงการเข้ารหัสแบบไม่สมมาตรแบบความลับ 16
รูปที่ 3.11 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคล 17
รูปที่ 3.12 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคลและความลับ 18
รูปที่ 5.1 ตัวอย่างอินเทอร์เฟซของคอมออปเจกต์ 29
รูปที่ 5.2 แสดงการสร้างอินเทอร์เฟซใหม่เมื่อมีเมธอดใหม่ 30
รูปที่ 5.3 แสดงอินเทอร์เฟซ IUnknow 30
รูปที่ 5.4 แสดงอินเทอร์เฟซที่สืบทอดมาจาก Iunknow 31
รูปที่ 5.5 แสดงลักษณะ VTABLE 31
รูปที่ 6.1 แสดงการทำงานของคลาส ColeDataObject 32
รูปที่ 8.1 แทรกวัตถุที่ต้องการลงบนเอกสาร 39
รูปที่ 8.2 เลือกโปรแกรม Isag Signature 40
รูปที่ 8.3 Isag Signature บนเอกสาร 40
รูปที่ 8.4 เปิดแมโครเพื่อบันทึกชื่อ 41
รูปที่ 8.5 กำหนดชื่อให้แมโคร 41
รูปที่ 8.6 ปรากฏหน้าต่างการบันทึกแมโคร 42
รูปที่ 8.7 กำหนดคุณสมบัติของแมโคร 42
รูปที่ 8.8 กำหนดชื่อให้แถบเครื่องมือ 43

หน้า

รูปที่ 8.9 แถบเครื่องมือใหม่บนเอกสาร 43
รูปที่ 8.10 เลือกแมโครที่ทำการบันทึก Isag Signature 44
รูปที่ 8.11 ได้แถบเครื่องมือ Isag Signature 44
รูปที่ 8.12 กำหนดชื่อให้เครื่องมือใหม่ 45
รูปที่ 8.13 เลือกรูปตามต้องการ 45
รูปที่ 8.14 แถบเครื่องมือตามต้องการ 46
รูปที่ 8.15 ทำการบันทึกแบบแมโครใหม่ 46
รูปที่ 9.1 แสดงรายชื่อผู้ใช้ที่มีอยู่แล้ว 47
รูปที่ 9.2 หน้าจอเมื่อทำการเพิ่มลายมือชื่อ 48
รูปที่ 9.3 หน้าจอเมื่อกดยืนยันการสร้าง 49
รูปที่ 9.4 หน้าจอเมื่อกดปุ่ม Set Security Level 49
รูปที่ 9.5 หน้าจอเลือกว่ายืนยันใช้รหัสผ่านอันนี้ 50
รูปที่ 9.6 รูปแสดงการสร้างรหัสผ่านอันใหม่ 50
รูปที่ 9.7 ผลของการสร้างลายมือชื่ออันใหม่ 51
รูปที่ 9.8 หน้าจอโปรแกรมระหว่างทำงานในโปรแกรมไมโครซอฟท์เวิร์ด 51
รูปที่ 9.9 ผลที่ได้จากการลงลายมือชื่อ 52
รูปที่ 9.10 แสดงการตรวจสอบลายมือชื่อดิจิตอลว่าถูกต้อง 53
รูปที่ 9.11 แสดงการตรวจสอบลายมือชื่อดิจิตอลและข้อมูลมีการแก้ไข 53

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงข้อดีและข้อเสียของการเข้ารหัส 21
ตารางที่ 4.1 ชื่อที่ใช้บ่งบอกว่าเป็นซีเอสพีตัวใด 24
ตารางที่ 7.1 แสดงรูปแบบคลิบบอร์ดมาตรฐาน 35

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันการทำธุรกรรมต่าง ๆ สามารถกระทำได้บนเครื่องคอมพิวเตอร์ โดยผ่านทางระบบเครือข่ายได้มีมากมายและแพร่หลายเพิ่มมากขึ้น เช่น จดหมายอิเล็กทรอนิกส์ การยืนยันตัวเป็นเรื่องที่สำคัญเพราะเราไม่สามารถรู้ได้เลยว่า ข้อมูลที่เราได้รับมาเป็นของบุคคลที่เราต้องการหรือไม่ ทำให้ลายมือชื่อดิจิตอล (Digital Signature) มีความสำคัญในการระบุสิทธิ์ของบุคคล

ลายมือชื่อดิจิตอล มีไว้แจ้งสิทธิ์ของบุคคล โดยไม่ขึ้นอยู่กับตัวบุคคลแต่เพียงอย่างเดียว แต่ขึ้นอยู่กับข้อมูลทั้งหมดรวมกับข้อมูลของบุคคลนั้น ๆ ลายมือชื่อดิจิตอลสร้างโดยการเข้ารหัสทางเดียว ดังนั้นถ้าข้อมูลมีการเปลี่ยนแปลงไปเพียงบิตเดียวก็จะทำให้ลายมือชื่อดิจิตอลเปลี่ยนไปด้วย จึงทำให้สามารถพิสูจน์ได้ว่าเป็นข้อมูลจากบุคคลที่เราต้องการหรือไม่ และสามารถตรวจสอบได้ว่าข้อมูลมีความถูกต้องจริง

ลายมือชื่อดิจิตอลเป็นลายมือชื่ออิเล็กทรอนิกส์ไม่ใช่ลายมือที่เขียนโดยคนคนหนึ่ง ซึ่งสามารถพิสูจน์ระบุถึงผู้ส่งหรือผู้ลงนามเอกสารได้ ทั้งยังทำให้แน่ใจได้ว่าข้อความหรือเอกสารที่ส่งมาไม่ได้มีการเปลี่ยนแปลงแก้ไข และไม่สามารถบอกปัดหรือปฏิเสธได้ ไม่สามารถปลอมได้
หลักการทำงานของลายมือชื่อดิจิตอลเริ่มจากการนำข้อมูลที่ต้องการส่งมาสังเคราะห์ข้อมูลขึ้น (Message digest-MD) โดยใช้ฟังก์ชันทางเดียว เมื่อได้ข้อมูลมาแล้วเข้ารหัสโดยใช้คีย์ส่วนตัวของผู้ส่ง ข้อมูลที่ผ่านการเข้ารหัสดังกล่าวนี้เรียกว่าลายมือชื่อดิจิตอล โดยลายมือชื่อดิจิตอลของข้อมูลแต่ละชุดนั้นไม่คงที่ ผู้รับเมื่อรับข้อมูลชุดดังกล่าวจากผู้ส่งมาแล้ว ใช้ฟังก์ชันทางเดียวแบบเดียวกันเพื่อสังเคราะห์ข้อมูลชุดหนึ่งขึ้น จากนั้นจะใช้คีย์สาธารณะของผู้ส่งเพื่อถอดรหัสลายมือชื่อดิจิตอลที่ส่งมา ได้ข้อมูลสังเคราะห์อีกหนึ่งชุด จากนั้นเปรียบเทียบกับข้อมูลสังเคราะห์ดังกล่าว ถ้าตรงกันแสดงว่าข้อมูลนั้นไม่มีการปลอมแปลงแก้ไข และทำให้ทราบว่าเป็นบุคคลที่ส่งข้อมูลนั้นจริงๆ แต่ถ้าไม่ตรงกันก็แสดงว่าข้อมูลที่รับมานั้นไม่ถูกต้อง อาจถูกปลอมแปลงแก้ไขทำให้สามารถตรวจสอบบุคคลอื่นที่ปลอมตัวเข้ามาได้

1.2 วัตถุประสงค์

เพื่อศึกษาถึงการทำลายมือชื่อดิจิตอล และสามารถนำความรู้ที่ศึกษามาได้ นำมาพัฒนาโปรแกรมที่สามารถลงนามลายมือชื่อดิจิตอลได้ เพื่อความถูกต้องของข้อมูล ข้อมูลต้องไม่มีการเปลี่ยนแปลงระหว่างการส่ง และทำให้สามารถยืนยันบุคคลที่ส่งมาได้

1.3 ขอบเขตของโครงงาน

- ศึกษาเรื่องการเข้ารหัสทั้งแบบคีย์เหมือน และแบบคีย์ต่าง
- ศึกษาการทำแฮชกับข้อมูล
- ศึกษาวิธีการทำลายมือชื่อดิจิตอล
- ศึกษาวิธีการตรวจสอบลายมือชื่อดิจิตอล
- ศึกษาการอ่านข้อมูลแบบบิตแมพ
- ศึกษาการทำแถบเครื่องมือบนโปรแกรมไมโครซอฟต์เวิร์ด

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- ความเข้าใจในเรื่องการเข้ารหัสทั้งแบบคีย์เหมือนและคีย์ต่าง
- สามารถเขียนโปรแกรมที่สร้างลายมือชื่อดิจิตอลได้
- สามารถเขียนโปรแกรมที่ใช้ในการตรวจสอบลายมือชื่อดิจิตอลได้

1.5 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย อันได้แก่ การเข้ารหัสด้วยวิธีต่างๆ และการทำแฮชฟังก์ชัน ซึ่งมีรายละเอียดดังในบทที่ 3 จากนั้นก็เป็นการศึกษาเกี่ยวกับการเขียนโปรแกรม โดยทำการศึกษาภาษาวิชวลซีพลัสพลัส (Visual C++) และการศึกษาโอแอลอี ดังรายละเอียดในบทที่ 5 และทำการศึกษาโปรแกรม IsagSign เพื่อนำมาพัฒนาให้ง่ายต่อการใช้งาน

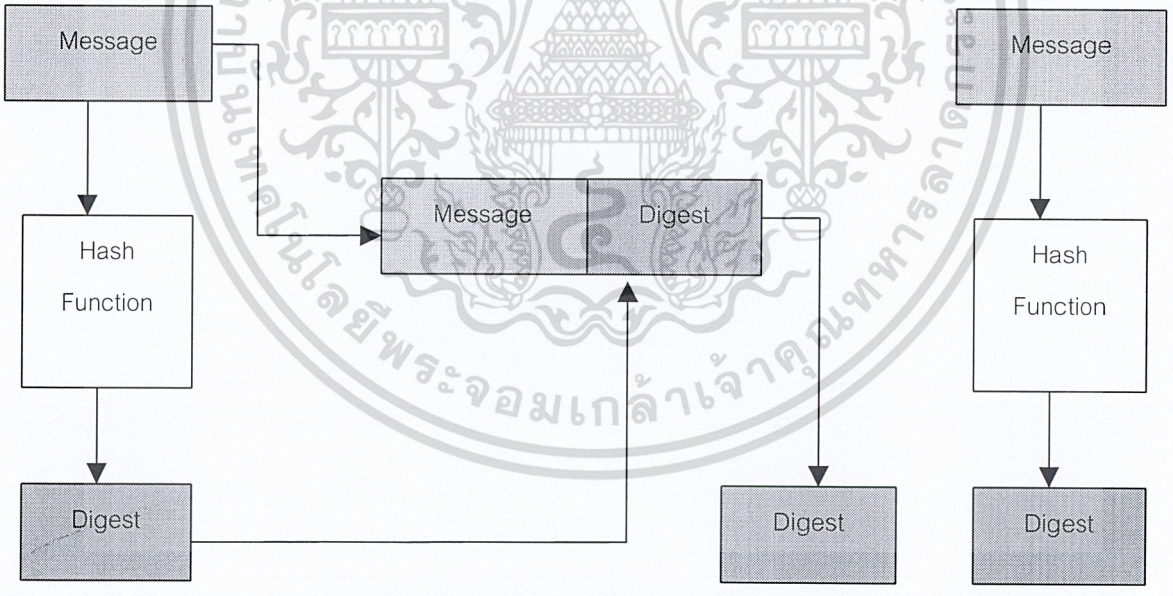
บทที่ 2
ลายมือชื่อดิจิตอล

2.1 การทำงานของลายมือชื่อดิจิตอล

ลายมือชื่อดิจิตอลใช้เมื่อต้องการความมั่นใจในแหล่งที่มา ของเอกสารเปรียบเหมือนลายมือชื่อซึ่งเฉพาะเจ้าของจริงที่สามารถคำนวณขึ้นมาได้ แต่ลายมือชื่อนี้สามารถพิสูจน์ได้คือบุคคลอื่นสามารถตรวจสอบได้ว่าลายมือชื่อนั้นมาจากผู้สร้างจริงๆ วิธีธรรมดาทั่วไปที่จะคำนวณลายมือชื่อดิจิตอลก็คือการเข้ารหัสแบบคีย์สาธารณะ เช่น ผู้ลงนามคำนวณค่าลายมือชื่อโดยใช้คีย์ส่วนตัว (Private key) และคนอื่นสามารถใช้คีย์สาธารณะพิสูจน์ได้ว่าลายมือชื่อมาจากคีย์ส่วนตัวที่ตรงกัน คุณสมบัติที่สำคัญของลายมือชื่อดิจิตอลที่สำคัญนั้นจะต้องประกอบด้วย 2 ประการคือ

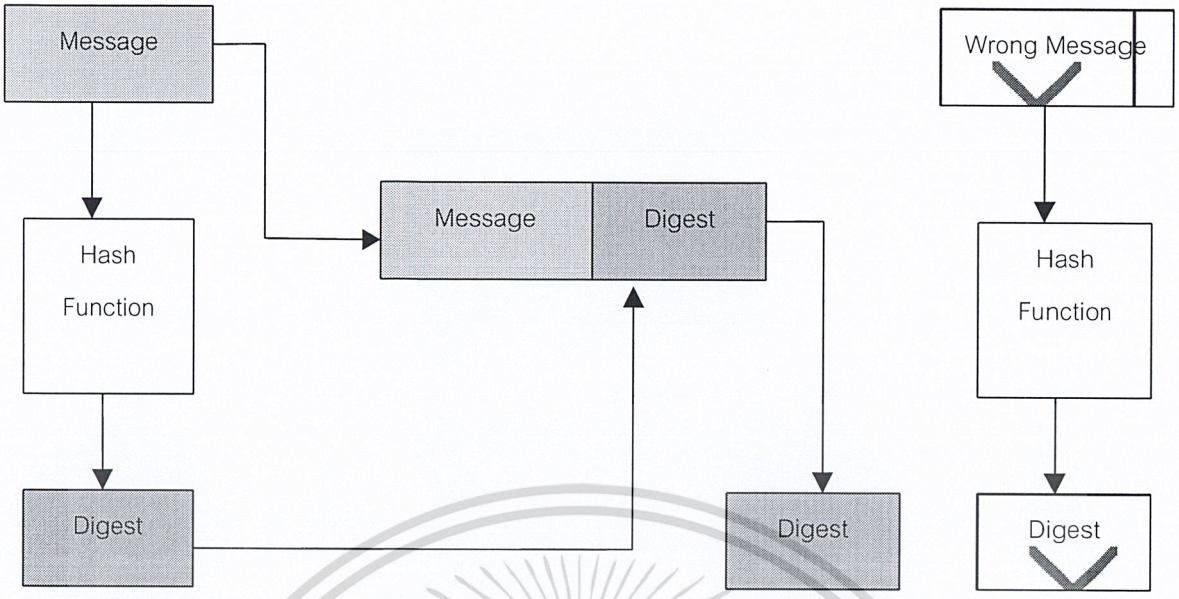
- สามารถยืนยันได้ว่าข้อมูลที่ได้รับมานั้นไม่มีการเปลี่ยนแปลงระหว่างการส่ง
- สามารถยืนยันได้ว่าข้อมูลนั้นได้รับการยืนยันจากผู้ลงลายมือชื่อจริง ๆ

วิธีการยืนยันว่าข้อมูลที่ได้รับมานั้นมีความถูกต้อง และไม่ได้รับการเปลี่ยนแปลงข้อมูลระหว่างการส่งนั้นสามารถทำได้โดยวิธีง่าย ๆ คือ การหาค่าแฮช (Hash Value) ดังรูปที่ 2.1



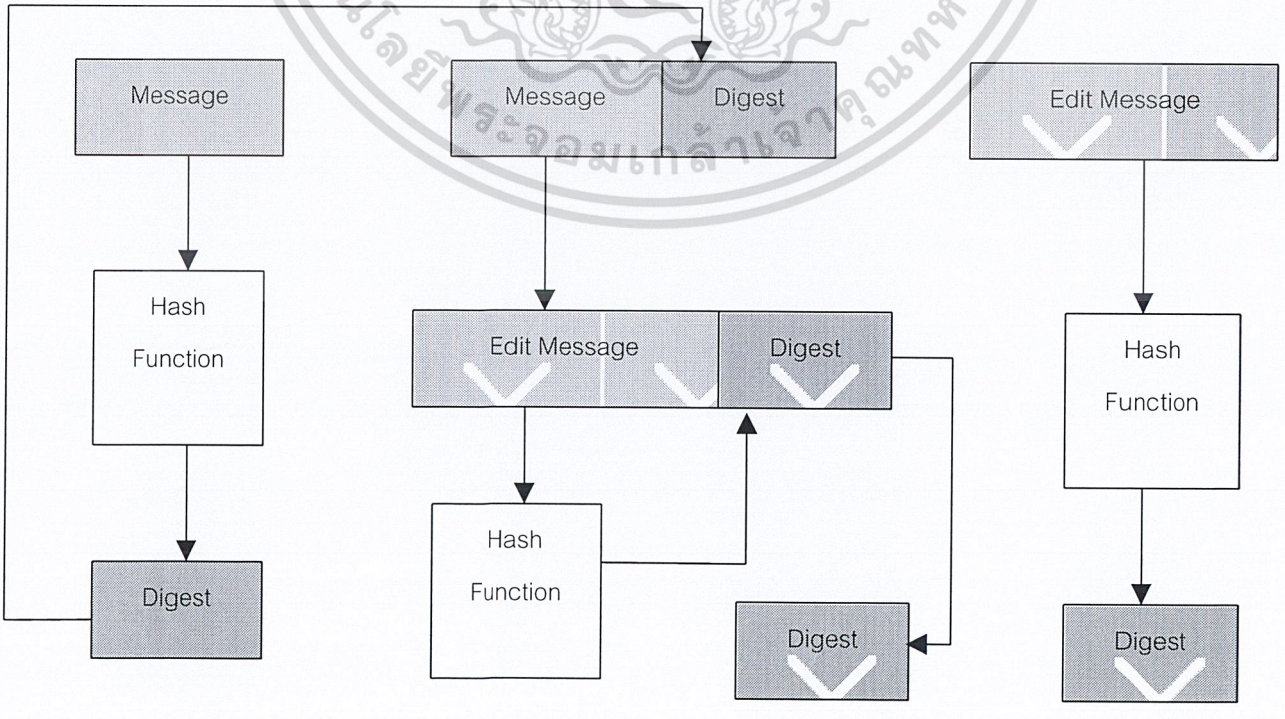
รูปที่ 2.1 วิธีการตรวจสอบความถูกต้องของข้อมูลแบบพื้นฐาน

จากรูปที่ 2.1 จะเป็นวิธีการที่ใช้ในการยืนยันความถูกต้องของข้อมูลอย่างง่าย โดยการหาค่าแฮช หรือจะเรียกว่าไดเจสต์ (Digest หรือ Message Digest) ก็ได้ซึ่งไดเจสต์ จะเปรียบเสมือนตัวแทนของข้อมูล ถ้าข้อมูลที่ได้รับเข้าสู่แฮชฟังก์ชันนั้นต่างกันค่าของไดเจสต์ที่ได้รับออกมานั้นก็จะต่างกัน แต่หากว่าข้อมูลที่ได้รับเข้าสู่แฮชฟังก์ชันนั้นเหมือนกันก็จะทำให้ค่าของไดเจสต์ที่ได้ออกมาเหมือนกัน มีโอกาสน้อยมากที่จะทำให้ข้อมูลเปลี่ยนไปแล้วยังได้ไดเจสต์เหมือนเดิม



รูปที่ 2.2 เมื่อข้อมูลเกิดการผิดพลาด

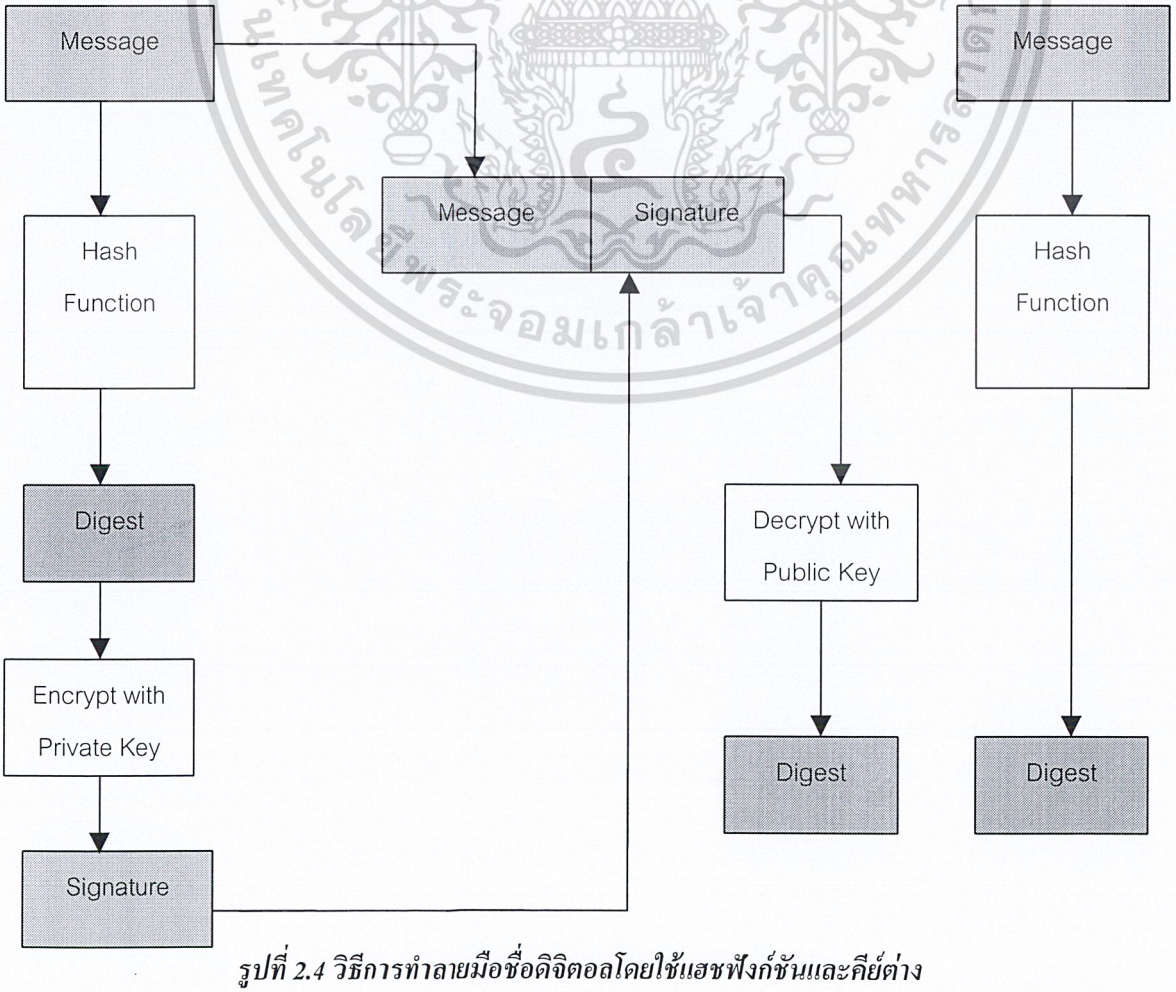
จากรูปที่ 2.2 แสดงให้เห็นทางด้านซ้ายมือเป็นของผู้ส่งซึ่งจะมีทั้งข้อมูลและไดเจสต์ส่งมาให้ ทางด้านผู้รับเกิดได้รับข้อมูลที่มีความผิดพลาดเกิดขึ้น (Wrong) เมื่อด้านผู้รับทำการแฮชชิงได้ไดเจสต์ออกมาเมื่อนำไปเปรียบเทียบกับไดเจสต์ที่ได้รับ จะเห็นว่าไม่เหมือนกัน ทำให้ฝั่งผู้รับทราบได้ทันทีว่าข้อมูลที่ได้รับมานั้นมีความผิดพลาด แต่การทำวิธีนี้อาจไม่ปลอดภัยนักเนื่องจากหากผู้ต้องการปลอมแปลงทราบว่าผู้ส่งใช้แฮชฟังก์ชันอะไร จะทำให้สามารถทำการเปลี่ยนแปลงข้อมูลนั้นได้ดูได้จากรูปที่ 2.3



รูปที่ 2.3 เมื่อมีผู้อ้างตัวเป็นผู้ส่ง

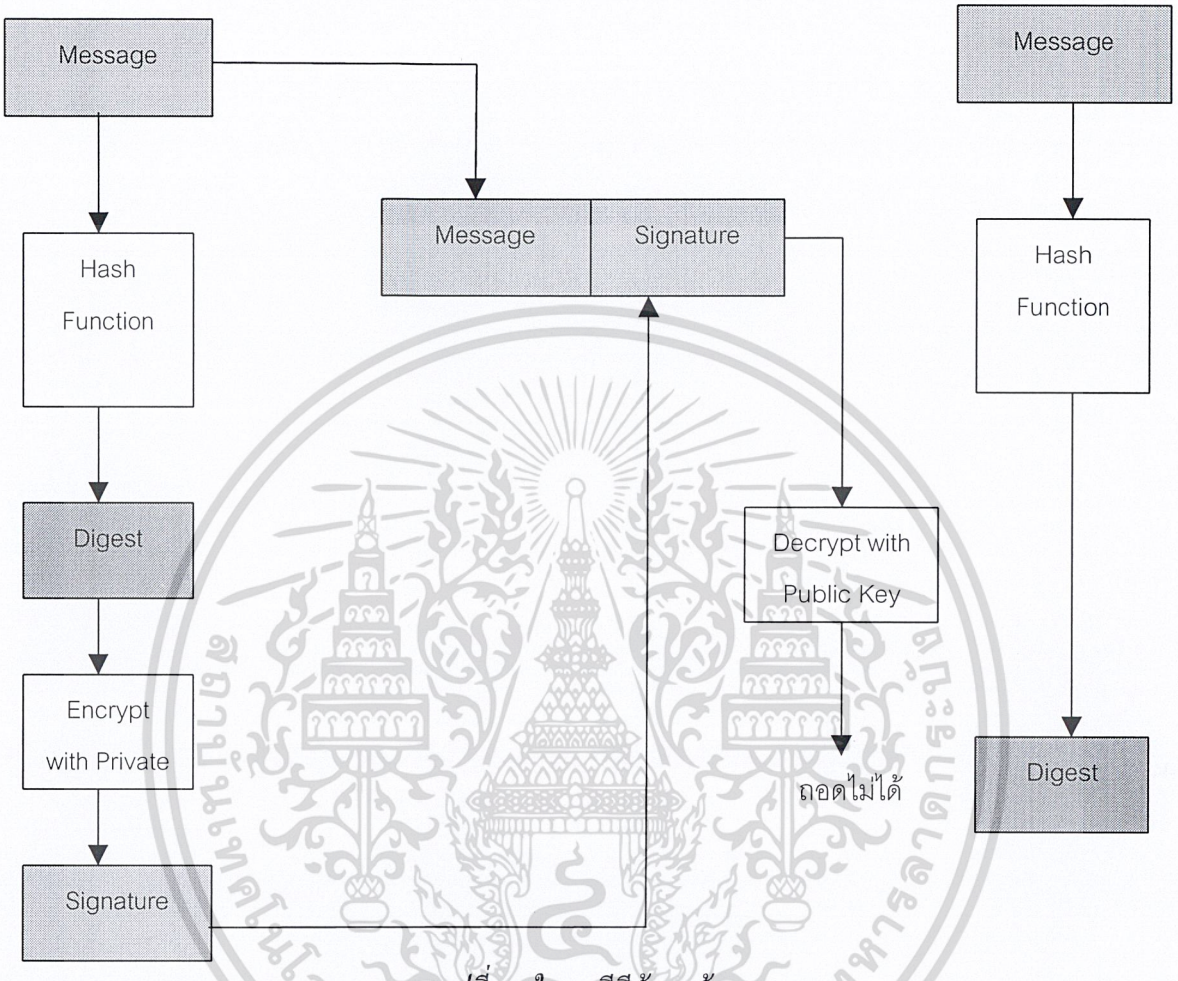
จากรูปที่ 2.3 เมื่อมีผู้อ้างตัวเป็นผู้ส่งนำข้อมูลที่ผู้ส่งส่งมาดัดแปลงแก้ไข แล้วใช้แฮชฟังก์ชันซึ่งเป็นอันเดียวกับผู้ส่งและผู้รับใช้จะทำให้ด้านผู้รับไม่อาจรู้ได้ว่าข้อมูลนั้นมีการเปลี่ยนแปลง ทำให้ไม่ปลอดภัยดังนั้นจึงได้มีการใช้การเข้ารหัสเข้ามาช่วย

ในการยืนยันบุคคลผู้ส่งนั้นเป็นสิ่งที่จะต้องทำตามมาหลังจากการยืนยันความถูกต้องของข้อมูล เทคนิคที่นิยมใช้ในการทำก็คือ ใช้เทคนิคคีย์ต่าง เช่น อาร์เอสเอ (RSA) โดยข้อความแรกเริ่มจะผ่านแฮชฟังก์ชันแล้วได้เป็นแมสเซจไดเจสต์ เพื่อเป็นการยืนยันความถูกต้องของข้อมูลก่อน จากนั้นจะนำไดเจสต์ที่ได้ส่งไปเข้ารหัสกับคีย์ส่วนตัวของผู้ส่งเพื่อเป็นการยืนยันว่าเป็นเจ้าของเอกสารนั้นจริง และจะได้สิ่งที่เรียกว่าลายมือชื่อดิจิตอลออกมาส่งไปพร้อมกับข้อมูลด้วย ทางด้านรับนั้นจะเป็นวิธีการที่คล้ายกับด้านส่ง โดยจะต้องเริ่มจากการหาแมสเซจไดเจสต์ของข้อมูลที่ได้รับมาเก็บไว้ก่อน จากนั้นจะนำลายมือชื่อดิจิตอลที่ได้รับมาด้วย มาทำการถอดรหัสด้วยคีย์สาธารณะของผู้ส่ง ในขั้นตอนนี้เองที่จะสามารถยืนยันบุคคลผู้ส่งได้ เพราะถ้าหากผู้ที่ลงลายมือชื่อไม่ได้เป็นบุคคลที่เราคาดว่าเป็นเจ้าของ ก็จะไม่สามารถถอดลายมือชื่อดิจิตอลได้ ถ้าหากการยืนยันบุคคลผู้ส่งสำเร็จแล้วจะได้รับแมสเซจไดเจสต์ของข้อมูลก่อนจะส่งออกมา ด้านรับจะต้องทำการตรวจสอบความถูกต้องของข้อมูลด้วยโดยการนำแมสเซจไดเจสต์ที่คำนวณได้ในตอนรับข้อมูล กับแมสเซจไดเจสต์ที่ได้หลังจากการถอดรหัสลายมือชื่อดิจิตอลออกมา นำมาเปรียบเทียบกันถ้าหากแมสเซจไดเจสต์ที่ได้ออกมามีค่าเท่ากันนั่นแสดงว่าข้อมูลที่ได้รับนั้นเป็นข้อมูลเดียวกันจากทางด้านส่ง และไม่มีการเปลี่ยนแปลงข้อมูลระหว่างการส่ง ขั้นตอนการส่งและการรับของวิธีการนี้ แสดงไว้ในรูปที่ 2.4



รูปที่ 2.4 วิธีการทำลายมือชื่อดิจิตอลโดยใช้แฮชฟังก์ชันและคีย์ต่าง

จากรูปที่ 2.4 จะเห็นว่ามีความปลอดภัยมากยิ่งขึ้นในกรณีที่มีการปลอมแปลงจะเป็นดังรูปที่ 2.5



รูปที่ 2.5 ในกรณีมีผู้แอบอ้าง

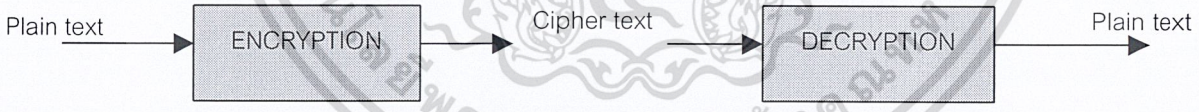
จากรูปที่ 2.5 เมื่อมีผู้แอบอ้าง ผู้แอบอ้างไม่รู้คีย์ส่วนตัวของผู้ส่งจริงๆ จึงใช้คีย์ของตัวเอง ซึ่งไม่ได้เข้าคู่กับคีย์สาธารณะทางฝั่งผู้รับ ทำให้ทางฝั่งผู้รับไม่สามารถถอดรหัสลายมือชื่อดิจิตอลได้ จึงรู้ว่าไม่ใช่ผู้ส่งตัวจริงเป็นคนส่ง

บทที่ 3
การเข้ารหัส

ในการส่งข้อมูลที่เป็นความลับไปยังผู้รับ บางครั้งอาจจะมีการลักลอบอ่านและปลอมแปลงข้อมูลในระหว่างทางที่ส่งไป การเข้ารหัส(Encryption) เป็นวิธีที่จะช่วยไม่ให้มีบุคคลอื่นใดสามารถอ่านข้อมูลที่เราส่งไปได้ จึงทำให้ไม่สามารถที่จะปลอมแปลงข้อมูลของเราได้ การเข้ารหัสเป็นการรักษาความปลอดภัยของข้อมูลอย่างหนึ่ง โดยการเปลี่ยนแปลงข้อมูลโดยที่รู้กันระหว่างผู้รับและผู้ส่ง การเข้ารหัสมีพื้นฐานอยู่ 2 อย่างคือ

1. การแทนที่ (Substitution) เป็นการแทนที่บิตใด ๆด้วยข้อมูลอื่น ทำให้ข้อมูลมีความสับสนยากต่อการถอดรหัส
2. การสับเปลี่ยนตำแหน่ง(Permutation) เป็นการสับเปลี่ยนตำแหน่งใด ๆของข้อมูล เมื่อมีการสับเปลี่ยนตำแหน่งมาก ๆทำให้ข้อมูลมีความซับซ้อนยากต่อการถอดรหัส

ในระบบการเข้ารหัสข้อมูลที่เราสามารถอ่านได้เรียกว่า เพลนเท็กซ์ (Plain text) ข้อมูลที่เข้ารหัสแล้วจะกลายเป็นข้อมูลที่อ่านไม่ออกเรียกว่า ไซเฟอร์เท็กซ์ (Cipher text) วิธีการในการเข้ารหัสเรียก อัลกอริทึมการเข้ารหัส (Encryption Algorithm) ส่วนวิธีการในการถอดรหัสเรียก อัลกอริทึมการถอดรหัส (Decryption Algorithm) เมื่อนำเพลนเท็กซ์มาเข้ารหัสก็จะได้ไซเฟอร์เท็กซ์ ซึ่งถ้าเราได้รับไซเฟอร์เท็กซ์มาต้องทำการถอดรหัสเพื่อให้ได้เพลนเท็กซ์ดังรูป



รูปที่ 3.1 แสดงการเข้ารหัสและถอดรหัส

ซึ่งสามารถอธิบายการทำงานได้ดังนี้

ข้อมูลเพลนเท็กซ์อยู่ในรูปอนุกรมดังนี้ P = [P1,P2,P3,P4,P5,P6,…,Pn] เมื่อเข้ารหัสแล้วข้อมูลไซเฟอร์เท็กซ์อยู่ในรูปอนุกรมดังนี้ C = [C1,C2,C3,C4,C5,C6,…,Cn] สามารถเขียนอยู่ในรูป สมการได้ดังนี้

C = E(P) และ P = D(C)

โดยที่

C = ไซเฟอร์เท็กซ์

P = เพลนเท็กซ์

E = อัลกอริทึมการเข้ารหัส

D = อัลกอริทึมการถอดรหัส

ในระบบรักษาความปลอดภัยได้ดังนี้ P = D(E(P))

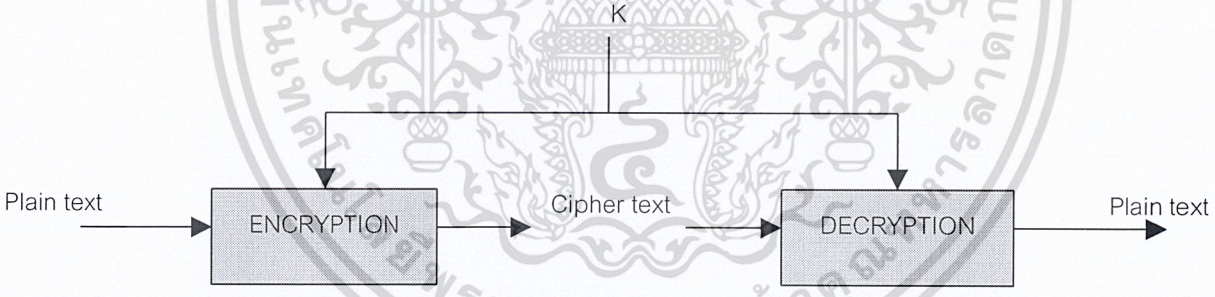
ระบบการเข้ารหัสและถอดรหัส โดยอาศัยคีย์ในการแปลงข้อมูลมีอยู่ 2 ประเภทคือ

1. ระบบการเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem) เป็นระบบที่การเข้ารหัสและการถอดรหัสใช้รูปแบบและคีย์เดียวกัน เช่น DES, 3DES,IDEA
2. ระบบการเข้ารหัสแบบไม่สมมาตร (Asymmetric Cryptosystem) เป็นระบบการเข้ารหัสที่ใช้ 2 คีย์ที่มีความเกี่ยวข้องกันทางคณิตศาสตร์ โดยประกอบด้วย คีย์สาธารณะ (Public Key) และคีย์ส่วนตัว (Private Key) โดยถ้าการเข้ารหัสทำโดยการใช้คีย์สาธารณะการถอดรหัสต้องใช้คีย์ส่วนตัวที่เป็นคู่ของมันในการถอดรหัสเท่านั้น ในทางตรงกันข้ามถ้าใช้คีย์ส่วนตัวในการเข้ารหัสต้องใช้คีย์สาธารณะในการถอดรหัสเท่านั้นเช่นกัน เช่น RSA, DH, DSA

การเข้ารหัสที่ขึ้นอยู่กับคีย์นั้น การเข้ารหัสข้อมูลเปลี่ยนแปลงไปตามคีย์ ถึงผู้อื่นรู้อัลกอริทึมในการเข้ารหัสแต่ไม่รู้คีย์ก็ไม่สามารถถอดรหัสออกมาได้ ซึ่งจะได้กล่าวถัดไป

3.1 พื้นฐานการเข้ารหัสแบบสมมาตร

การเข้ารหัสแบบสมมาตรทำงานโดยทั้งการเข้ารหัสและการถอดรหัสจะทำโดยใช้รูปแบบเดียวกัน หรือจะเรียกว่าการเข้ารหัสแบบคีย์เดียว (Single Key Encryption) หรือการเข้ารหัสแบบซีเครทคีย์ (Secret Key Encryption) เพราะว่าทั้งการเข้ารหัสและการถอดรหัสจะใช้คีย์เดียวกันดังรูป



รูปที่ 3.2 แสดงการเข้ารหัสและการถอดรหัสแบบสมมาตร

ซึ่งสามารถแสดงด้วยสมการได้ดังนี้

C = E(K,P) และ P = D(K,C) หรือได้ว่า P = D(K,(K,C))

โดย P = เพลนเท็กซ์

C = ไซเฟอร์เท็กซ์

E = อัลกอริทึมการเข้ารหัส

D = อัลกอริทึมการถอดรหัส

K = ซีเครทคีย์ที่ใช้ในการเข้ารหัสและการถอดรหัส

3.2 ความต้องการของระบบการเข้ารหัสแบบสมมาตร

ในการเข้ารหัสแบบสมมาตรเพื่อความลับต้องการต้องการอัลกอริทึมการเข้ารหัสที่มีความแข็งแกร่งเพียงพอ ซึ่งคือไม่ว่าผู้ใดรู้ถึงอัลกอริทึมการเข้ารหัสจะต้องไม่สามารถหาเพลนเท็กซ์จากไซเฟอร์เท็กซ์ หรือหาซีเครทคีย์ได้

1. ต้องมีเพียงผู้รับและผู้ส่งเท่านั้นที่รู้ซีเครทคีย์ ถ้าเมื่อใดที่มีผู้รู้ซีเครทคีย์นี้แล้วรู้อัลกอริทึมการถอดรหัสก็จะสามารถหาเพลนเท็กซ์ได้

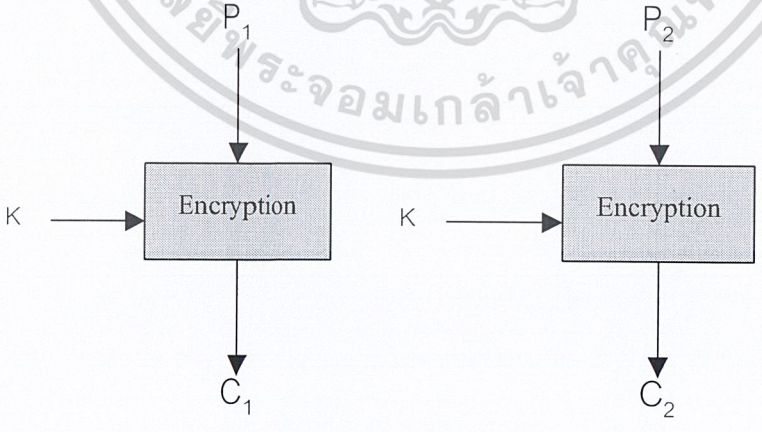
สิ่งสำคัญในการเข้ารหัสแบบสมมาตรคือการรักษาคีย์ให้เป็นความลับเพราะถือว่าการถอดรหัสข้อมูลเป็นวิธีพื้นฐานที่ได้จาก ไซเฟอร์เท็กซ์รวบกับความรู้ในอัลกอริทึมของการเข้ารหัสและถอดรหัส

3.3 ลักษณะที่สำคัญของการเข้ารหัสแบบสมมาตร

1. การแปลงเพลนเท็กซ์เป็นไซเฟอร์เท็กซ์ ใช้การกระทำ 2 แบบ คือ ทั้งการแทนที่ ใช้เพื่อเปลี่ยนข้อมูลตำแหน่งใดๆในเพลนเท็กซ์เป็นข้อมูลอื่น และใช้การเปลี่ยนตำแหน่งเพื่อจัดเรียงข้อมูลใหม่ ที่สำคัญคือข้อมูลต้องไม่หาย
2. ใช้คีย์เพียงตัวเดียวทั้งผู้รับและผู้ส่ง
3. ใช้บล็อกไซเฟอร์ในการเข้ารหัส

3.4 ลักษณะของบล๊อกไซเฟอร์

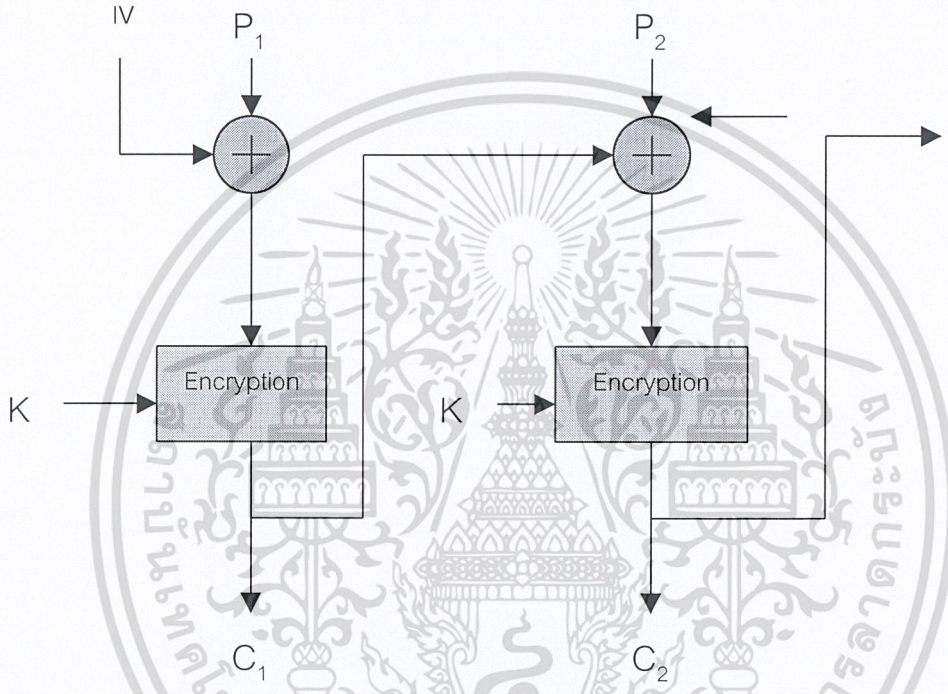
1. อิเล็กทรอนิกส์โค้ดบุ๊ก (Electronic Codebook : ECB)แบ่งเพลนเท็กซ์ออกเป็นบล็อก บล็อกละเท่าๆกันใช้คีย์และอัลกอริทึมเดียวกันดังรูปที่ 3.3



รูปที่ 3.3 แสดงการเข้ารหัสแบบอิเล็กทรอนิกส์โค้ดบุ๊ก

ข้อดีสำหรับบล็อกไซเฟอร์แบบอิเล็กทรอนิกส์โค้ดบุ๊ก คือสำหรับข้อมูลสั้นๆจะได้ไซเฟอร์เท็กซ์ที่ดีและสามารถทำที่ละหลายบล็อกทำให้เร็ว แต่ถ้าข้อมูลยาวมากๆอาจมีข้อมูลซ้ำกัน เมื่อทำการเข้ารหัสจะทำให้ได้ไซเฟอร์เท็กซ์ที่มีลักษณะซ้ำกันได้

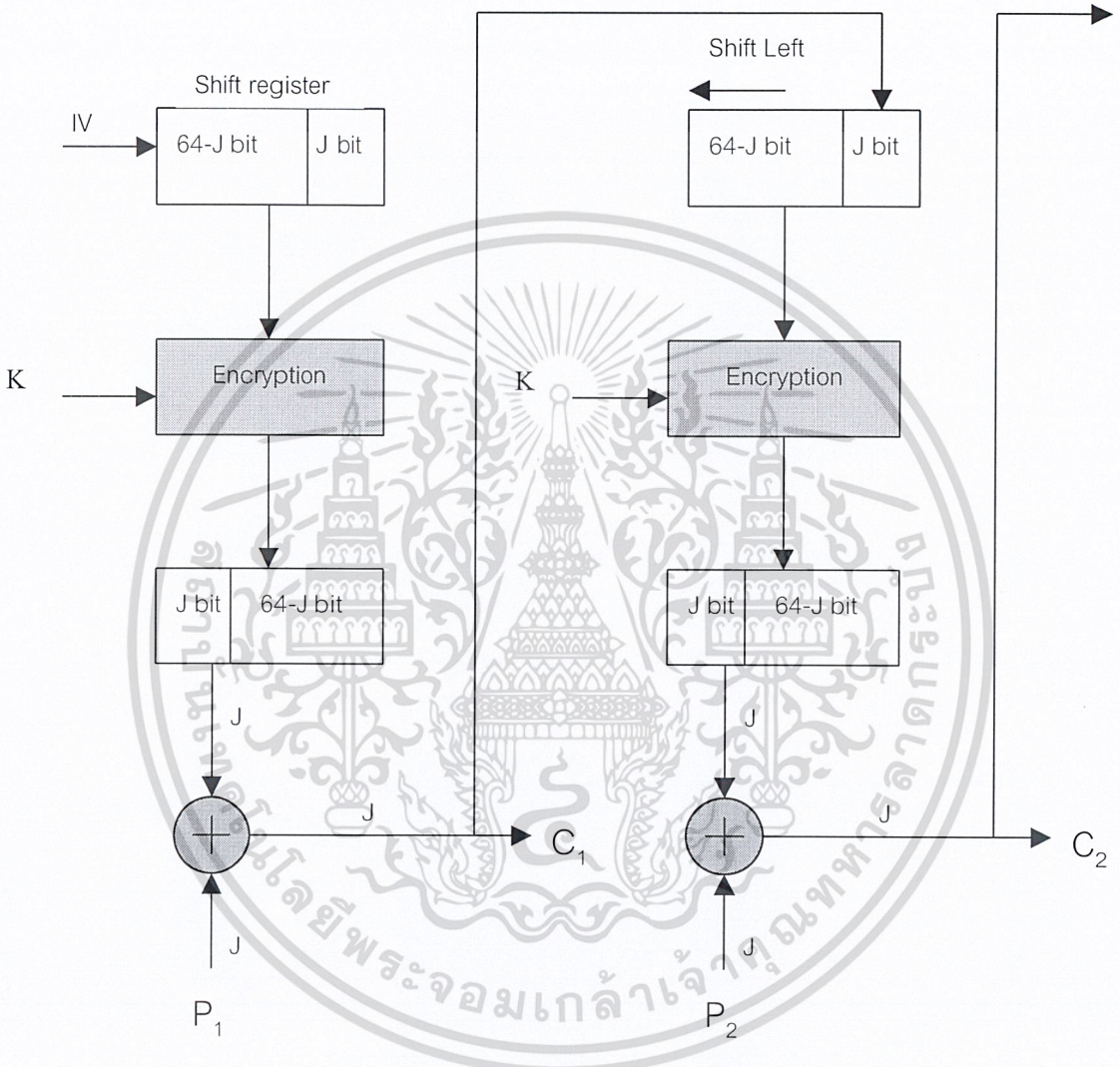
2. ไซเฟอร์บล๊อกเชนนิ่ง (Cipher Block Chaining :CBC) แบ่งเพลนเท็กซ์ ออกเป็น บล็อก ๆละเท่าๆกัน โดยก่อนการเข้ารหัสนำเพลนเท็กซ์มาเอ็กซ์คลูซีฟออร์ (XOR) กับไซเฟอร์เท็กซ์ก่อนหน้ามัน โดยเพลนเท็กซ์แรกจะทำการ XOR กับเวคเตอร์เริ่มต้น (Initial Vector : IV) ดังรูปที่ 3.4



รูปที่ 3.4 แสดงการเข้ารหัสแบบไซเฟอร์บล็อกเชนนิ่ง

ข้อดีของบล็อกไซเฟอร์ลักษณะนี้คือสามารถแก้ปัญหาซ้ำกันของข้อมูลก่อนเข้ารหัสเพราะต้องนำเพลนเท็กซ์มา XOR กับไซเฟอร์เท็กซ์ก่อนหน้ามันก่อนที่จะนำมาเข้ารหัส แต่มีข้อเสียเนื่องจากการเข้ารหัสแต่ละบล๊อกต้องรอบล๊อกข้างหน้ามันก่อนจึงทำการเข้ารหัสได้จึงไม่สามารถทำพร้อมกันได้จึงทำให้ช้า

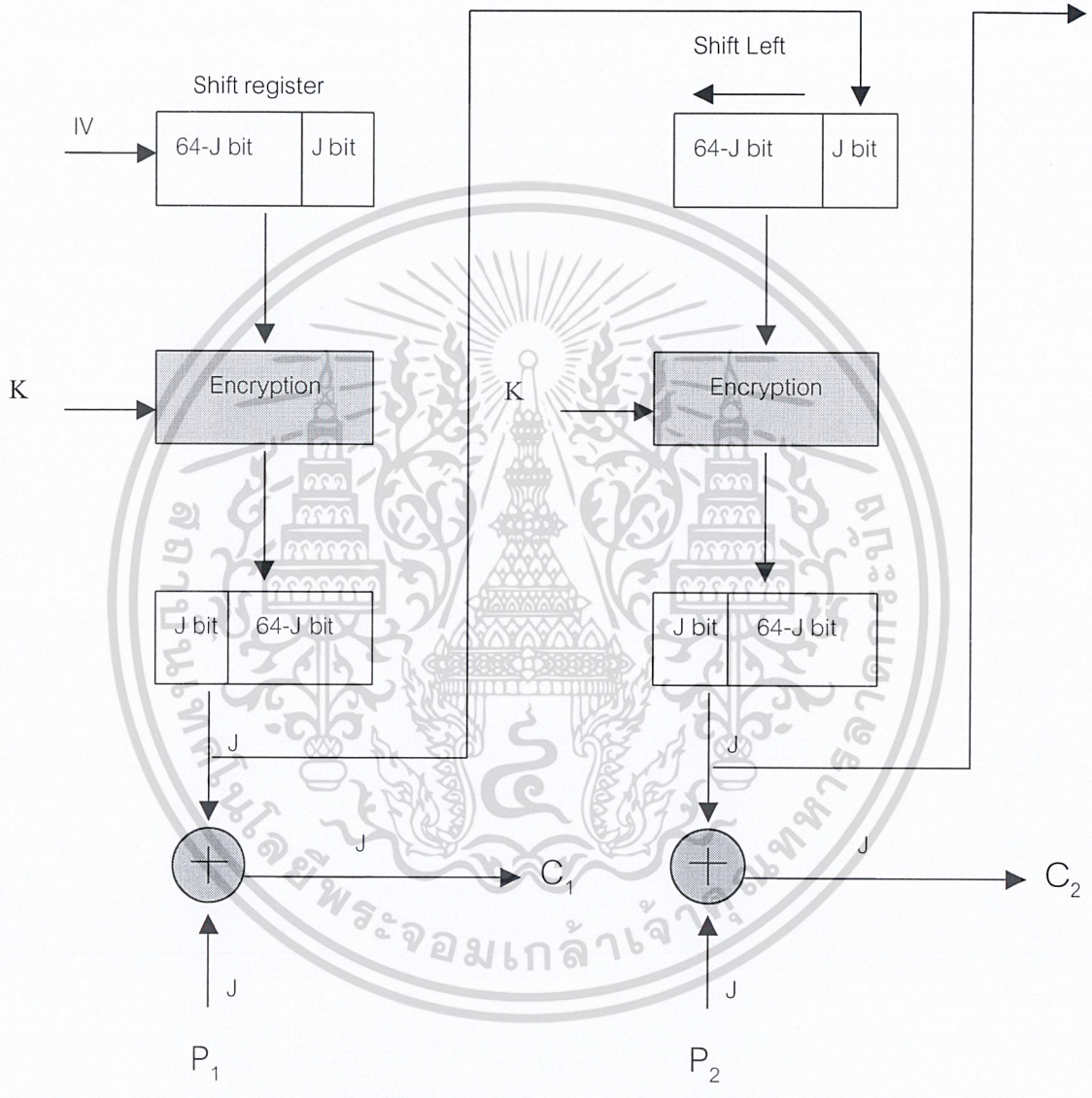
3. ไซเฟอร์ฟีดแบ๊ก (Cipher Feed Back : CFB) ขั้นแรกมี ชิฟริจิสเตอร์ (Shift Register) ขนาด 64 บิตโดยกำหนดค่าเริ่มต้นเป็นเวคเตอร์เริ่มต้น (IV) แล้วนำมาเข้ารหัสกับคีย์ (K) จากนั้นจำ J บิต หน้าสุดนำมา XOR เพลนเท็กซ์แล้วนำไซเฟอร์เท็กซ์ที่ได้ไปใช้ใน ชิฟริจิสเตอร์ J บิตท้าย โดยเลื่อนไปด้านซ้าย ไป J บิต เพื่อนำข้อมูลไซเฟอร์เท็กซ์เข้ามาดังรูปที่ 3.5



รูปที่ 3.5 แสดงการเข้ารหัสแบบไซเฟอร์ฟีดแบ็ก

โดยที่ส่วนใหญ่จะทำทีละ 8 บิต ซึ่งคือ 1 ตัวอักษร (Character) โดยสามารถทำงานแบบ เวลาจริง ได้ (Real Time) ได้

4. เอาท์พุตฟีดแบ็ก (Output Feedback :OFB) เป็นบล็อกไซเฟอร์ที่ทำงานคล้ายกับไซเฟอร์ฟีดแบ็กเพียงแต่นำข้อมูลที่ได้จากการเข้ารหัส J บิตแรกไปใส่ในชิฟริจิสเตอร์ J บิตท้ายแทนดังรูปที่ 3.6



รูปที่ 3.6 แสดงการเข้ารหัสแบบเอาท์พุตฟีดแบ็ก

ข้อดีของเอาท์พุตฟีดแบ็กความผิดพลาดจะไม่แพร่ไปยังบล็อกถัดไป คือถ้าเกิดความผิดพลาดจากไซเฟอร์เท็กซ์บล็อกแรกบล็อกต่อมาจะไม่ได้รับผลกระทบจากบล็อกแรกเพราะนำข้อมูล J บิตก่อนทำการ XOR กับข้อมูลเพลนเท็กซ์มาใส่ในชิฟริจิสเตอร์แทน

3.5 DES (Data Encryption Standard)

3.5.1 ประวัติและที่มาของ DES

ในปลายทศวรรษที่ 1960 บริษัท IBM ได้จัดตั้งโครงงานวิจัยทางด้านการเข้ารหัสด้วยคอมพิวเตอร์ (Computer Cryptography) ซึ่งนำโดยฮอสท์ เฟสเทล (Horst Feistel) ซึ่งโครงงานนี้เสร็จสิ้นในปี 1971 ซึ่งผลงานวิจัยของโครงงานนี้คือลูซิเฟอร์ (LUCIFER[FEIS73]) โดยมีลักษณะเป็นการเข้ารหัสข้อมูลเป็นบล็อคขนาด 64 บิตและใช้คีย์ขนาด 128 บิต ซึ่งต่อมาได้ถูกพัฒนาขนาดของคีย์ให้ลดลงเหลือขนาด 56 บิต

โดยอัลกอริทึมของการเข้ารหัสข้อมูลของลูซิเฟอร์ได้ถูกพัฒนาโดย IBM สำหรับ NBS(National Bureau of Standards) อัลกอริทึมนี้ได้เป็นที่รู้จักในนามของ DES (Data Encryption Standard) ถึงแม้ว่าชื่อจริงของมันคือ DEA (Data Encryption Algorithm) ในสหรัฐและ DEAI (Data Encryption Algorithm-1) ในอีกหลายๆ ประเทศ

3.5.2 รายละเอียดของ DES

เป็นวิธีการเข้ารหัสที่ใช้กันอย่างแพร่หลายที่เป็นพื้นฐานบน Data Encryption Standard(DES) ที่ได้พัฒนาขึ้นในปี 1977 โดย National Bureau of Standards ซึ่งปัจจุบันคือ Federal Information Processing Standard 46(FIPS PUB46) สำหรับ DES ข้อมูลจะถูกเข้ารหัสเป็นบล็อคขนาด 64 บิต ซึ่งใช้คีย์ขนาด 56 บิต โดยวิธีการจัดการกับข้อมูล 64 บิตที่เข้ามาเพื่อแปลงเป็นข้อมูล 64 บิตออกไป และใช้คีย์ตัวเดียวกันนี้ในการถอดรหัส

แม้ว่า DES ถูกนำมาใช้ตั้งแต่ช่วงทศวรรษที่ 70 (ค.ศ.1960-1970) และได้รับการตอบรับอย่างดีจากเหล่านักวิเคราะห์รหัส(Cryptanalysis) อย่างแพร่หลาย แต่ก็เป็นข้อถกเถียงกันเป็นอย่างมากถึงเรื่องDES นั้นจะปลอดภัยหรือไม่และมีความปลอดภัยมากน้อยแค่ไหน แต่จนถึงปัจจุบันเราก็ยังไม่พบช่องโหว่ของ DES ตามเอกสารที่ตีพิมพ์เป็นสาธารณะ แม้ว่าจะใช้คีย์เพียงไม่กี่บิตก็ตาม ในทางตรงกันข้ามแนวความคิดแบบ IDEA กลับใช้คีย์แบบ 128 บิต(ซึ่งมีขนาดกว่า 2 เท่าของ DES) และได้รับการตอบรับจากสาธารณะตั้งแต่ทศวรรษที่ 90 (ค.ศ.1980-1990) (แต่ดีไม่เท่าตอนประกาศใช้ DES) IDEA มีความปลอดภัยมากกว่า DES และสามารถประมวลผลได้เร็วกว่า DES อย่างไรก็ตาม IDEA ยังต้องรอการตรวจสอบจากผู้เชี่ยวชาญอีกมากถึงเรื่องช่องโหว่ของความปลอดภัย

3.5.3 อัลกอริทึมของการเข้ารหัสแบบ DES

โดยพิจารณาออกเป็น 2 ส่วนเพื่อให้ง่ายแก่การทำความเข้าใจคือ ส่วนที่เป็นคีย์ที่จะใช้ในการเข้ารหัสและส่วนที่เป็นข้อมูลที่จะนำมาทำการเข้ารหัส

อัลกอริทึมเป็นผลมาจากทฤษฎีของแชนนอน (Shannon) ซึ่งเกี่ยวกับการปิดบังข่าวสาร ซึ่งแนะนำ 2 วิธีในการปกปิดข่าวสารนั่นคือคอนฟิวชัน (Confusion) และดิฟฟิวชัน (Diffusion)

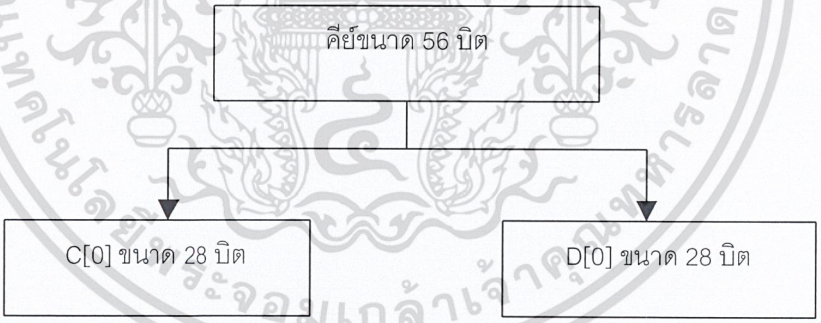
คอนฟิวชัน คือการทำให้ชิ้นส่วนของข่าวสารถูกเปลี่ยนไป ดังนั้นเอาท์พุทบิทจะสังเกตไม่เห็นความสัมพันธ์กับอินพุทบิต

ดิฟฟิวชัน จะทำการกระจายเพลนเท็กซ์บิตไปที่บิตอื่นในไซเฟอร์เท็กซ์ มีผลทำให้ข้อมูลต่าง ๆ มีความซับซ้อนมากขึ้น เพราะข้อมูลจะถูกกระจายไปในตำแหน่งอื่นด้วย

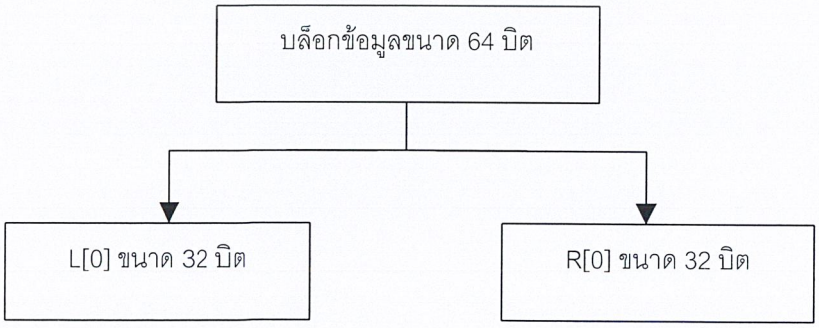
อัลกอริทึม DES ที่กระทำบนบล็อกของข้อมูล บล็อกของข้อมูลจะถูกแยกออกเป็น 2 ส่วน แต่ละส่วนจะแยกออกเป็นอิสระต่อกัน จากนั้นจะทำการรวมคีย์กับส่วนใดส่วนหนึ่งของข้อมูลและก็สลับกัน 2 ส่วนโปรเซส (Process) นี้จะทำซ้ำ 16 ครั้ง อัลกอริทึมในการทำซ้ำจะใช้เทเบิลลุคอัพ (Table lookup) และการคำนวณบิตแบบง่าย ๆ (simple bit) ถึงแม้ว่าการจัดการระดับบิทของอัลกอริทึมจะยุ่งยากซับซ้อน

อินพุทที่เข้ามา DES จะแบ่งอินพุทออกเป็นบล็อกๆ ละ 64 บิต ซึ่งจะถูกเปลี่ยนไปใช้คีย์ขนาด 64 บิต ข้อมูลขนาด 64 บิต จะถูกสับเปลี่ยนตำแหน่งโดยการสับเปลี่ยนตำแหน่งเริ่มต้น (Initial Permutation) และคีย์จะถูกลดลงจาก 64 บิต เหลือ 56 บิต โดยการทิ้งบิตที่ 8, 16, 24, ..., 64 ซึ่งบิตเหล่านี้จะถูกกำหนดเป็นพาริตี้บิต โดยแสดงไว้ในรูปที่ 3.1

ข้อมูล 64 บิตที่ถูกสับเปลี่ยนตำแหน่งแล้วจะถูกแบ่งเป็นครึ่งซ้ายและครึ่งขวา (แต่ละครึ่งมีขนาด 32 บิต) แสดงไว้ในรูปที่ 3.2 คีย์จะถูกชิพไปทางซ้ายโดยการกำหนดที่จำนวนบิตและจะทำการสลับตำแหน่ง ต่อไปคีย์จะถูกรวมกับครึ่งขวา หลังจากนั้นก็จะมารวมกับครึ่งซ้ายใหม่อีกครั้ง ผลลัพธ์ของการรวมนี้จะเปลี่ยนเป็นครึ่งด้านขวาใหม่ ส่วนครึ่งขวาเก่าจะกลายมาเป็นครึ่งซ้ายใหม่ กิจกรรมเหล่านี้จะทำเป็นวัฏจักร (Cycle) วัฏจักรจะถูกทำซ้ำ 16 ครั้ง หลังจากวัฏจักรสุดท้ายซึ่งเป็นการสับเปลี่ยนครั้งสุดท้ายซึ่งจะถูกสับเปลี่ยนตำแหน่งบิตผกผันกับแบบเริ่มต้น (Inverse Initial Permutation (IP)) หรือการสับเปลี่ยนตำแหน่งบิตผกผันกับแบบเริ่มต้น ตามตารางจะได้ผลลัพธ์สุดท้ายออกมา คือข้อมูลที่ถูกทำการเข้ารหัสแล้ว



รูปที่ 3.7 แสดงการแบ่งคีย์ 56 บิตที่เรียงลำดับบิตแล้วออกเป็น 2 ส่วน



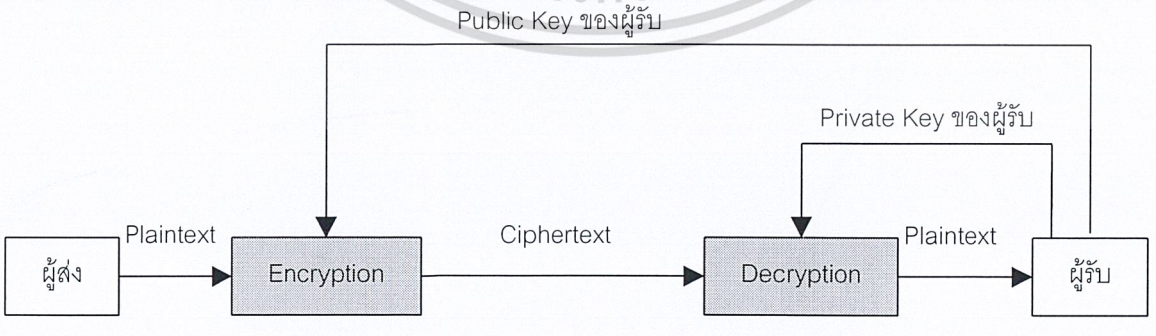
รูปที่ 3.8 แสดงบล็อกข้อมูล 64 บิต ที่แบ่งออกเป็น 2 ส่วน หลังจากทำการจัดเรียงบิตแล้ว

3.6 ปัญหาและความปลอดภัยในการเข้ารหัสแบบสมมาตร

1. ระบบที่ใช้คีย์เพียงดอกเดียวในการเข้ารหัสข้อมูลนั้น ถ้าคีย์ที่เป็นความลับถูกขโมยไป บุคคลที่รู้คีย์นั้นสามารถถอดรหัสข้อมูลที่เข้ารหัสนั้นได้ นอกจากนั้นยังอาจปลอมแปลงข้อมูลเดิมนั้นใหม่ แล้วเข้ารหัสข้อมูลที่ทำการปลอมแปลงนั้นด้วยคีย์เดียวกัน แล้วส่งไปยังผู้รับตัวจริง ดังนั้นเพื่อความปลอดภัยควรเลือกคีย์ที่ยากแก่การคาดเดาและเก็บไว้อย่างปลอดภัย รวมทั้งไม่ควรใช้คีย์เดียวกันนี้ซ้ำกันหลาย ๆครั้ง
2. การเข้ารหัสแบบสมมาตรค่อนข้างอ่อน เสี่ยงต่อการโจรกรรม เนื่องจากความยาวคีย์ที่ใช้ไม่มากพอ หรืออัลกอริทึมที่ใช้นั้นค่อนข้างง่ายไม่ค่อยซับซ้อนมากนัก
3. การส่งคีย์ไปพร้อมกับข้อมูลที่เข้ารหัสนั้น อาจทำให้เกิดปัญหาได้ ซึ่งถ้าทำอย่างนั้นแล้วคีย์ที่ส่งไปด้วยนั้นจะต้องถูกส่งไปด้วยความปลอดภัยสูงมาก ยิ่งเป็นการส่งไปในระยะทางไกล ๆ วิธีที่ง่ายคือการส่งคีย์ให้กับผู้รับด้วยมือของคนส่งเอง แต่อาจทำให้ไม่สะดวกและเสียเวลา ส่วนอีกวิธีคือการส่งคีย์ไปพร้อมกับข้อมูลนั้น แต่ทำการแบ่งคีย์ออกเป็นส่วน ๆก่อน แล้วจึงทำการส่งไปตามเส้นทางที่ต่างกัน ซึ่งแม้ว่าคีย์บางส่วนจะถูกดักได้แต่ก็ไม่สามารถรู้ถึงตัวคีย์ที่สมบูรณ์ได้

3.7 พื้นฐานการเข้ารหัสแบบไม่สมมาตร

แนวความคิดของการเข้ารหัสแบบไม่สมมาตรหรือการเข้ารหัสแบบคีย์สาธารณะ (Public Key Encryption) ได้เกิดขึ้นจากการพยายามแก้ไขปัญหาของการเข้ารหัสแบบสมมาตร 2 ข้อด้วยกันคือ การจัดการคีย์ (Key Management) และการลงนามรับรองข่าวสารทางดิจิตอล (Digital Signature) ใช้เพื่อการพิสูจน์สิทธิ์ ระบบการเข้ารหัสเป็นตัวอย่างหนึ่งของการป้องกันไม่ให้มีการเปิดเผยข้อมูลแก่ผู้ที่ไม่มีสิทธิ์ ลักษณะที่สำคัญของการเข้ารหัสแบบไม่สมมาตรคือการเข้ารหัสและการถอดรหัสใช้คีย์คนละดอกกันโดยคีย์ทั้งสองต้องเป็นคู่คีย์ (Key Pair) กัน คือ คีย์สาธารณะ (Public Key)ซึ่งเป็นคีย์ที่ทำการแจกให้ผู้อื่น และคีย์ส่วนตัว (Private Key)เป็นคีย์ที่เก็บไว้เป็นความลับ ดังรูป

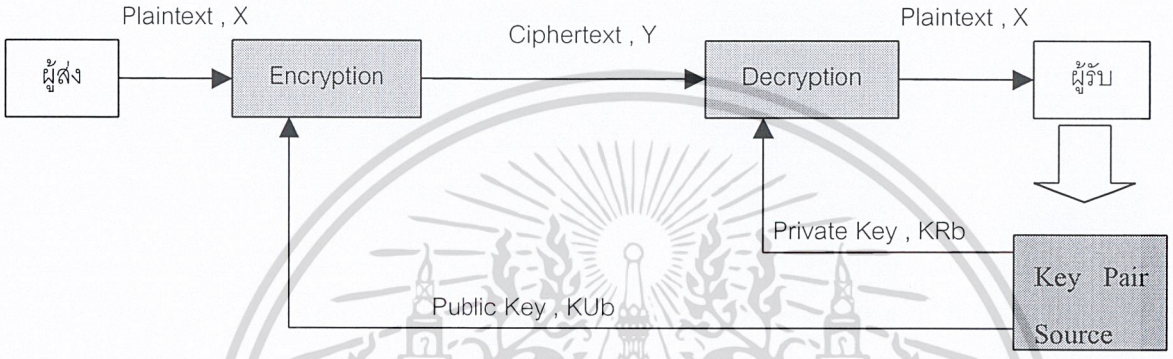


รูปที่ 3.9 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพื้นฐาน

3.8 ความสามารถของการเข้ารหัสแบบไม่สมมาตร

การเข้ารหัสแบบไม่สมมาตรมีความสามารถคือ

1. ความลับ (Secrecy) หมายถึงการไม่ยอมให้มีบุคคลที่ไม่มีสิทธิ์เข้ามาดูข้อมูลได้ ซึ่งสามารถทำได้โดยผู้ส่งเข้ารหัสโดยใช้คีย์สาธารณะของผู้รับ ซึ่งทำให้มีแต่ผู้รับที่มีคีย์ส่วนตัวที่เป็นคู่คีย์ของมันเท่านั้นที่สามารถทำการถอดรหัสได้ ถึงแม้คีย์สาธารณะมีบุคคลอื่นรู้ก็ไม่สามารถทำการถอดได้ ดังรูป



รูปที่ 3.10 แสดงการเข้ารหัสแบบไม่สมมาตรแบบความลับ

ซึ่งสามารถเขียนในรูปแบบสมการได้ดังนี้

ในการเข้ารหัส Y = EKUb(X)

ในการถอดรหัส X = DKRb(Y) หรือ X = DKRb(EKUb(Y))

โดยกำหนดให้

X = ข้อมูลเพลนเท็กซ์

Y = ข้อมูลไซเฟอร์เท็กซ์

EKUb = อัลกอริทึมการเข้ารหัสโดยใช้คีย์สาธารณะ

DKRb = อัลกอริทึมการถอดรหัสโดยใช้คีย์ส่วนตัว

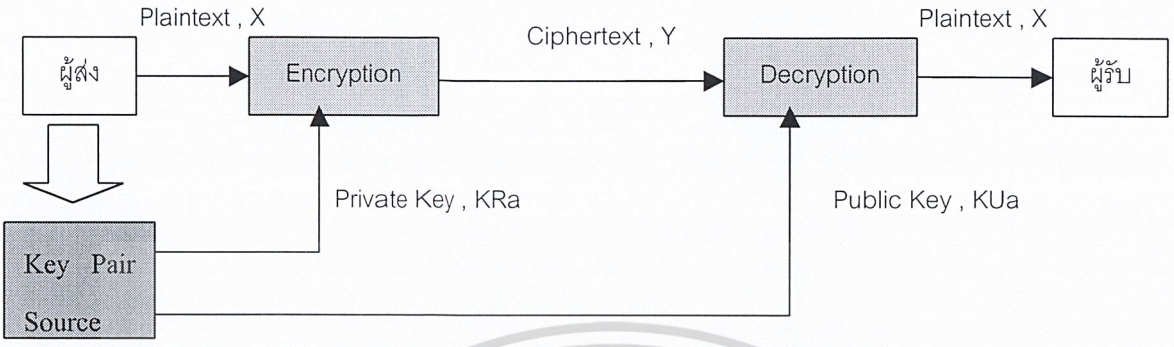
KUb = คีย์สาธารณะของผู้รับ

KRb = คีย์ส่วนตัวของผู้รับ

2. การพิสูจน์บุคคล (Authenticity) หมายถึงการตรวจสอบที่มาของข้อมูล ว่าถูกส่งมาจากผู้ส่งคนนั้นจริงหรือไม่ ซึ่งทำโดยเข้ารหัสข้อมูลโดยใช้คีย์ส่วนตัวของผู้ส่ง การตรวจสอบทำได้โดยใช้คีย์สาธารณะที่เป็นคู่คีย์ทำการถอดรหัส ซึ่งผู้ที่จะสามารถเข้ารหัสได้นั้นต้องเป็นผู้ที่เป็นเจ้าของคีย์ส่วนตัวนั้นเท่านั้น ส่วนมากการพิสูจน์ตัวบุคคลนั้นผู้ส่งจะทำการส่งข้อมูลที่เป็นข้อมูลส่วน

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตัวเข้ารหัสโดยคีย์ส่วนตัวไปให้ผู้รับ โดยผู้รับนำเอาข้อมูลมาทำการถอดรหัสแล้วตรวจดูว่าข้อมูลส่วนตัวของผู้ส่งเป็นจริงหรือไม่ โดยการเข้ารหัสเพื่อพิสูจน์ตัวบุคคลแสดงดังรูป



รูปที่ 3.11 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคล

ซึ่งสามารถเขียนในรูปสมการได้ดังนี้

ในการเข้ารหัส Y = EKRa(X)

ในการถอดรหัส X = DKUa(X) หรือ X = DKUa(EKRa(Y))

โดยกำหนดให้

X = ข้อมูลเพลนเท็กซ์

Y = ข้อมูลไซเฟอร์เท็กซ์

EKRa = อัลกอริทึมการเข้ารหัสโดยใช้คีย์ส่วนตัว

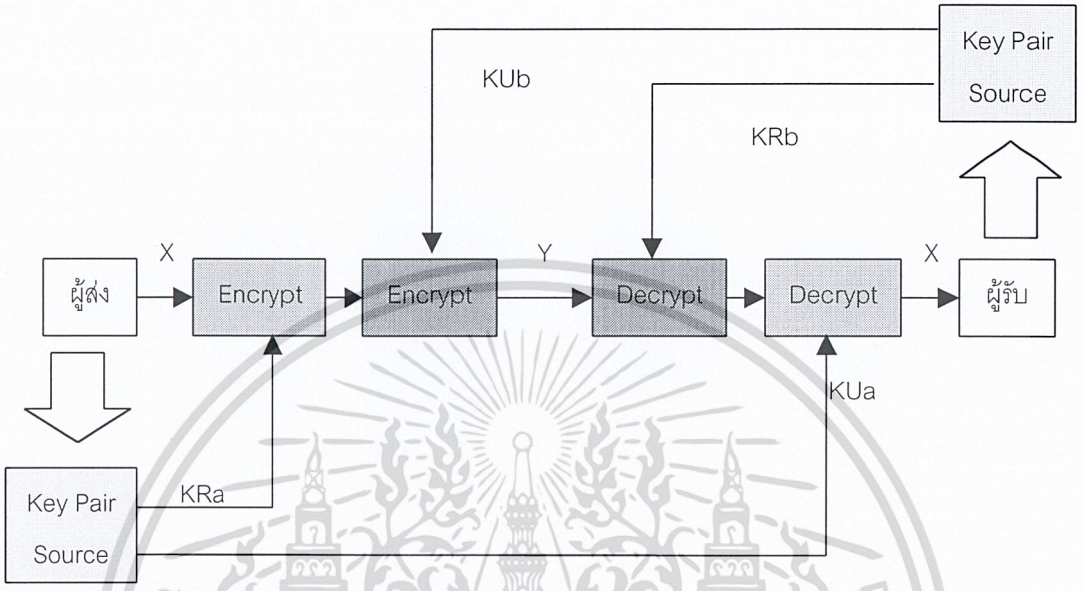
DKUa = อัลกอริทึมการถอดรหัสโดยใช้คีย์สาธารณะ

KUa = คีย์สาธารณะของผู้ส่ง

KRa = คีย์ส่วนตัวของผู้ส่ง

3. การพิสูจน์บุคคลและความลับ (Authenticity and Secrecy) คือการตรวจสอบที่มาของข้อมูลและจำกัดสิทธิ์ข้อมูลให้เพียงแต่ผู้รับเท่านั้นที่สามารถอ่านข้อมูลได้ สามารถทำได้โดยผู้ส่งทำการนำข้อมูลมาเข้ารหัสครั้งแรกด้วยคีย์ส่วนตัวของตนเพื่อเป็นการพิสูจน์ตัวบุคคลจากนั้นนำมาเข้ารหัสโดยคีย์สาธารณะของผู้รับเพื่อเป็นการรักษาความลับให้ผู้รับที่มีคีย์ส่วนตัวที่เป็นคู่คีย์เท่านั้นที่สามารถเปิดอ่านได้ เมื่อผู้รับได้รับไซเฟอร์เท็กซ์มาก็นำมาถอดรหัสโดยใช้คีย์ส่วนตัวของตัวเองเปิด จากนั้นนำมาถอดรหัสโดยใช้คีย์สาธารณะที่เป็นคู่คีย์ของผู้ส่ง ดังแสดงในรูปที่

46193



รูปที่ 3.12 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคลและความลับ

ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

ในการเข้ารหัส Y = EKUb[EKRa(X)]

ในการถอดรหัส X = DKUa[DKRb(Y)]

โดยกำหนดให้

X = ข้อมูลเพลนเท็กซ์

Y = ข้อมูลไซเฟอร์เท็กซ์

EKRa = อัลกอริทึมการเข้ารหัสโดยใช้คีย์ส่วนตัวของผู้ส่ง

EKUb = อัลกอริทึมการเข้ารหัสโดยใช้คีย์สาธารณะของผู้รับ

DKUa = อัลกอริทึมการถอดรหัสโดยใช้คีย์สาธารณะของผู้ส่ง

DKRb = อัลกอริทึมการถอดรหัสโดยใช้คีย์ส่วนตัวของผู้รับ

KRa = คีย์ส่วนตัวของผู้ส่ง

KUa = คีย์สาธารณะของผู้ส่ง

KRb = คีย์ส่วนตัวของผู้รับ

KUb = คีย์สาธารณะของผู้รับ

3.9 อาร์เอสเอ (RSA)

วิธีการของระบบ RSA ใช้ประโยชน์จากรูปแบบสมการเอ็กซ์โปเนนท์เชียล (Exponential) โดยเพลนเท็กซ์จะถูกเข้ารหัสเป็นบล็อก (Ciphertext block) แต่ละบล็อกมีค่าเป็นเลขฐานสองซึ่งมีค่าน้อยกว่าค่า n สำหรับบล็อกเพลนเท็กซ์ M บล็อกไซเฟอร์เท็กซ์ C การเข้ารหัสและถอดรหัสจะอยู่ในรูปแบบดังต่อไปนี้

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

ทั้งผู้ส่งและผู้รับต้องรู้ค่าของ n ผู้ส่งรู้ค่า e และมีผู้รับเพียงคนเดียวที่รู้ค่า d โดยกำหนดให้การเข้ารหัสประกอบด้วย KU = (e, n) และในการถอดรหัสต้องมี KR = (d, n) สิ่งสำคัญที่ต้องการสำหรับระบบนี้มีอยู่ 3 ประการก็คือ

1. ต้องสามารถหาค่า e, d และ n ที่ทำให้ $M^{ed} = M \bmod n$ สำหรับ M ทุกค่าที่ $M<n$
2. ค่า C^d และ M^e ต้องสามารถคำนวณได้โดยง่ายสำหรับค่า M ทุกๆ ค่าที่ $M<n$
3. ระบบต้องใช้ปัญหาทางคณิตศาสตร์ที่ยากพอที่จะไม่ให้สามารถคำนวณค่า d จากค่า e และ n ได้

วิธีในการสร้างคีย์มีดังนี้

1.เลือกเลขจำนวนเฉพาะ p, q (เลือกคีย์ส่วนตัว)

2.คำนวณหาค่า $n = p \times q$ (คำนวณหาคีย์สาธารณะ)

3.เลือกเลขจำนวนเต็ม d เมื่อ ห.ร.ม. $(\Phi^{(n)}, d)$ โดยที่ $1 < d < \Phi^{(n)}$ (คำนวณหาคีย์ส่วนตัว)

4.คำนวณหาค่า e เมื่อ $e = d^{-1} \bmod \Phi^{(n)}$ (เลือกคีย์สาธารณะ)

5.กำหนดให้คีย์สาธารณะเป็น KU = (e, n)

6.กำหนดคีย์ส่วนตัวเป็น KR = (d, n)

คีย์ส่วนตัวประกอบด้วย (d, n) และคีย์สาธารณะประกอบด้วย (e, n) นาย ก.ประกาศคีย์สาธารณะของเขาออกไป เมื่อนาย ข. ต้องการส่งข่าวสาร (สมมติให้เป็น M) ที่เป็นความลับให้แก่นาย ก. นาย ข. ต้องใช้คีย์สาธารณะของนาย ก. เพื่อนำมาใช้ในการคำนวณหรือเข้ารหัสออกมาเป็นไซเฟอร์เท็กซ์ $C = M^e \bmod n$ แล้วส่ง C ไปให้กับนาย ก. และนาย ก. จึงทำการคำนวณหรือถอดรหัสให้กลับเป็นข่าวสาร M เหมือนเดิมโดย $M = C^d \bmod n$

3.10 การคำนวณการเข้า-ถอดรหัสของ RSA

การคำนวณการเข้ารหัสและการถอดรหัสจะเกี่ยวกับการเพิ่มเลขจำนวนเต็มให้เป็นเลขจำนวนเต็มยกกำลังแล้วนำมามอดูลโลด้วย n การลดเวลาของการทำงานกับเลขจำนวนเต็มที่มีค่ามากๆ เราสามารถใช้คุณสมบัติของการมอดูลโลดังนี้ $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

ขั้นตอนการสร้างคีย์นั้นต้องหาเลขจำนวนเฉพาะ p และ q ที่มีขนาดมากๆ และเลือกค่า e และ d ซึ่งเราอาจนำค่า e และ d ที่เลือกมานี้ไปคำนวณอย่างอื่น เนื่องจากค่า n = pq ใช้สำหรับป้องกันการโจมตีโดย วิธี บรูทฟอร์ท (Brute Force) ที่ใช้การลองทุกค่าที่เป็นไปได้ทีละค่าในการแยกค่า p และ q ออกมา จึงต้องเลือกค่าของ p และ q จากเซตขนาดใหญ่และต้องมีค่ามากๆ

วิธีหนึ่งที่จะหาเลขจำนวนเฉพาะขนาดใหญ่จริงๆ ก็คือการเลือกเลขจำนวนคี่ที่ได้จากการสุ่มลำดับของค่าที่ต้องการแล้วทดสอบว่ามันเป็นจำนวนเฉพาะหรือไม่ ถ้าใช่ก็ดำเนินขั้นตอนต่อไป แต่ถ้าไม่ใช่ก็เลือกจำนวนคี่มาอีกตัวหนึ่งแล้วทดสอบ ต่อไปนี้เป็นตัวอย่างง่ายๆ

1.เลือก p = 11 และ q = 13 ดังนั้น n = 143

2.$\Phi(n) = (p-1)\times(q-1)$, $\Phi(n) = 120$ แล้วให้ e = 11

3.ดังนั้น d = 11 จาก $d = e^{-1} \bmod 120$

4.ให้ M = 7 (Plain text)

5.สร้างคีย์สาธารณะ KU = (11, 143)

6.สร้างคีย์ส่วนตัว KR = (11, 143)

7) แปลงเพลนเท็กซ์ให้เป็นไซเฟอร์เท็กซ์โดยคำนวณ $7^{11} \bmod 143 = 106$

8) เราคำนวณหาไซเฟอร์เท็กซ์ C ได้ออกมาเท่ากับ 106

9) นำมาคำนวณกลับแปลงจากไซเฟอร์เท็กซ์เป็นเพลนเท็กซ์ $106^{11} \bmod 143 = 7$

10) 7 เป็นค่าของเพลนเท็กซ์เริ่มต้น = M

3.11 บริการของการเข้ารหัสแบบไม่สมมาตร

1. การเข้ารหัสและการถอดรหัส (Encryption/Decryption) การเข้ารหัสแบบไม่สมมาตรสามารถทำการเข้ารหัสและถอดรหัสได้โดยใช้คีย์สาธารณะของผู้รับในการเข้ารหัส และใช้คีย์ส่วนตัวในการถอดรหัส
2. การลงลายมือชื่อดิจิตอล (Digital Signature) ทำโดยเข้ารหัสโดยใช้คีย์ส่วนตัวของผู้ส่งแล้วผู้รับสามารถใช้คีย์สาธารณะของผู้ส่งที่เป็นคู่คีย์ของมันถอดได้หรือไม่ ถ้าถอดได้คือการยืนยันตัวบุคคลที่ทำการลงลายมือชื่อนั้น
3. การแลกเปลี่ยนคีย์ (Key Exchange) ใช้ในการแลกเปลี่ยนซีเคร็ทคีย์ในการติดต่อในช่วงเวลาหนึ่ง โดยนำซีเคร็ทคีย์มาเข้ารหัสโดยใช้คีย์สาธารณะของผู้รับ ซึ่งผู้ที่สามารถถอดได้ก็คือผู้รับที่มีคีย์ส่วนตัวที่เป็นคู่คีย์ของมันเท่านั้น

การเข้ารหัสแบบสมมาตร	การเข้ารหัสแบบไม่สมมาตร
ข้อดี 1. การเข้ารหัสทำได้อย่างรวดเร็ว 2. สามารถสร้างได้ง่ายโดยฮาร์ดแวร์	**ข้อดี** 1. ใช้คีย์ต่างกันในการเข้ารหัสและการถอดรหัสทำให้การจัดส่งคีย์ทำได้ง่าย 2. สามารถตรวจสอบผู้ใช้ได้โดยใช้ร่วมกับลายมือชื่อดิจิตอล
ข้อเสีย 1. คีย์ของการเข้ารหัสและถอดรหัสต้องเหมือนกันการจัดส่งคีย์ทำได้ยาก	**ข้อเสีย** 1. ค่อนข้างช้าและต้องใช้การคำนวณอย่างมาก

ตารางที่ 3.1 แสดงข้อดีและข้อเสียของการเข้ารหัส

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 4

คริปโตเอพีไอ (CryptoAPI – Cryptographic Application Programming Interface)

คริปโตเอพีไอถือเป็นเอพีไอตัวหนึ่งซึ่งมีความสามารถในการทำงานในด้าน การเข้ารหัส ถอดรหัส และสามารถลงนามลายมือชื่อดิจิตอลได้ด้วย ซึ่งคณะผู้จัดทำได้นำมาใช้ในการพัฒนาตัวโปรแกรม โดยเอพีไอตัวนี้จะสามารถทำงานได้นั้นจะต้องทำการติดต่อกับส่วนของโปรแกรมที่เรียกว่าซีเอสพี (ซีเอสพี - Cryptographic Service Provider) ซึ่งซีเอสพีนั้นก็มีได้หลายตัว แล้วแต่ว่าจะเป็นของผู้ใดสร้าง

ในการเขียนโปรแกรมใด ๆ ให้สามารถที่จะเรียกใช้คริปโตเอพีไอให้ถูกต้องได้นั้นจำเป็นที่จะต้องรู้ถึงคำสั่งต่าง ๆ ในการเรียกใช้เอพีไอให้ถูกต้อง คำสั่งต่าง ๆ จะถูกกำหนดไว้ในไฟล์ที่ชื่อ Wincrypt.h ซึ่งผู้ที่พัฒนาโปรแกรมจำเป็นต้องรู้ถึงคำสั่งพื้นฐานในการใช้งานของเอพีไอ จึงจะสามารถพัฒนาโปรแกรมที่ใช้ความสามารถได้อย่างเต็มที่

4.1 การติดต่อกับซีเอสพี

ในการติดต่อกับซีเอสพีนั้นจะเริ่มจากการที่โปรแกรมใดๆ นั้นได้เรียกใช้คริปโตเอพีไอโดยการผ่านคำสั่ง CryptAcquireContext เพื่อที่จะใช้จัดการกับซีเอสพีใด ๆ โดยเมื่อเรียกใช้คำสั่งนี้แล้วจะเป็นการบ่งถึง ซีเอสพี ที่กำหนดไว้ในพารามิเตอร์ที่ส่งไป ซึ่งสิ่งที่ส่งไปไม่ได้มีเพียงแค่ชื่อของซีเอสพีเท่านั้นแต่จะรวมถึงส่วนที่ใช้เก็บคีย์ที่อยู่ภายในซีเอสพีนั้นด้วย

4.2 คำศัพท์ต่าง ๆ ที่ควรรู้

Key Container: เป็นเสมือนกับฐานข้อมูลที่ใช้เก็บคีย์ต่าง ๆ ในการเข้ารหัส ถอดรหัส และการลงนามลายมือชื่อดิจิตอล โดยคีย์คอนเทนเนอร์ (Key Container) แต่ละตัวนั้นจะมีชื่อเฉพาะ เพื่อไว้สำหรับการเรียกโดยใช้คำสั่ง CryptAcquireContext ซึ่งจะทำให้สามารถจัดการกับคีย์คอนเทนเนอร์ตัวนั้น ๆ ได้ โดยคีย์คอนเทนเนอร์นี้จะถูกใส่เข้าไปในซีเอสพี โดยแต่ละ ซีเอสพี นั้นจะเป็นผู้จัดการเองว่าให้คีย์คอนเทนเนอร์ไปเก็บไว้ที่ใด อาจจะเก็บไว้ในฮาร์ดแวร์เฉพาะ หรือเก็บไว้ในรีจีสตรี (Registry) หรือเก็บไว้ในระบบไฟล์ (File System) ก็ได้ขึ้นอยู่กับซีเอสพี นั้น ๆ

Signature Key Pair: เป็นคู่ของคีย์ที่มีไว้สำหรับการทำการพิสูจน์ตน จะถูกใช้ในการลงนามลายมือชื่อดิจิตอล การสร้างคู่คีย์ (Key Pair) ชนิดนี้สามารถสร้างได้โดยใช้คำสั่ง CryptGenKey ของคริปโตเอพีไอ

Exchange Key Pair: เป็นคู่ของคีย์ที่มีไว้สำหรับการทำการเข้ารหัส/ถอดรหัส ถูกใช้ในการเข้ารหัส เซสชันคีย์ (Session Key) โดยสามารถสร้างคู่คีย์ชนิดนี้จากคำสั่ง CryptGenKey ของคริปโตเอพีไอ

Algorithm Class: เป็นชนิดของอัลกอริทึมซึ่งมีได้ 4 ชนิดคือ

- ALG_CLASS_DATA_ENCRYPT สำหรับการทำการเข้ารหัสข้อมูล
- ALG_CLASS_HASH สำหรับการทำการแฮชชิง
- ALG_CLASS_KEY_EXCHANGE สำหรับการแลกเปลี่ยนคีย์
- ALG_CLASS_SIGNATURE สำหรับลายมือชื่อดิจิตอล

คริปโตเอพีไอ: ประกอบด้วยส่วนประกอบ 4 ส่วน คือ

1. Key Functions
2. Encryption/Decryption Functions
3. Hashing Functions
4. CSP Functions

ก่อนที่จะเรียกใช้คริปโตเอพีไอได้นั้นสิ่งที่จำเป็นที่จะต้องทำการสร้างดีฟอลท์คีย์คอนเทนเนอร์ขึ้นมาเสียก่อน เพราะคีย์คอนเทนเนอร์นั้นถือเป็นส่วนประกอบที่สำคัญที่สุด หลังจากได้สร้างคีย์คอนเทนเนอร์แล้วสิ่งที่ต้องสร้างตามมาอีกก็คือคู่คีย์ซึ่งจะต้องสร้างคู่คีย์ขึ้นมา 2 คู่คือคู่คีย์ลายมือชื่อ (Signature Key Pair) เพื่อใช้สำหรับการลงนามลายมือชื่อดิจิตอล และคู่คีย์แลกเปลี่ยน (Exchange Key Pair) เพื่อใช้สำหรับการเข้ารหัสเพื่อการแลกเปลี่ยนข้อมูลกัน

เอสพีไอ (SPI - Service Provider Interface): เป็นกลุ่มของการอินเตอร์เฟส (Interface) และฟังก์ชัน (Function) ที่มีไว้เพื่อให้เซอร์วิสโพรไวเดอร์ (Service Provider) ต่าง ๆ ได้เรียกใช้ โดยไฟล์ที่ใช้เพื่อให้เซอร์วิสโพรไวเดอร์เรียกใช้นั้นมีชื่อไฟล์ว่า MAPISPI.H

Provider Type: เป็นชนิดของซีเอสพี ซึ่งมีได้หลายชนิด และแต่ละชนิดก็อาจจะมีรูปแบบของข้อมูล (Data Format) ที่ไม่เหมือนกัน หรืออาจจะเป็นฟังก์ชันการทำงานใช้อัลกอริทึมในการทำงานไม่เหมือนกัน ในชนิดโพรไวเดอร์ (Provider Type) หนึ่ง ๆ นั้นสามารถมีซีเอสพี ได้หลายตัว และแต่ละตัวมีการทำงานในลักษณะเดียวกัน แบ่งชนิดของผู้ให้ได้ดังนี้

- ***PROV_DSS Provider Type*** เป็นชนิดโพรไวเดอร์ที่สามารถที่จะทำลายมือชื่อดิจิตอล และทำการแฮชได้ ซึ่งภายในจะประกอบไปด้วย อัลกอริทึมการลงลายมือชื่อ การทำแฮชชิงแบบ MD5 และการทำแฮชชิ่งแบบ SHA-1
- ***PROV_DSS_DH Provider Type*** เป็นชนิดโพรไวเดอร์ที่มีความสามารถในการทำคีย์แลกเปลี่ยน คีย์ลายมือชื่อดิจิตอลและการทำแฮชชิ่งซึ่งจะคล้ายกับ PROV_DSS
- ***PROV_FROTEZZA Provider Type*** เป็นชนิดโพรไวเดอร์ที่มีความสามารถในการทำคีย์แลกเปลี่ยน คีย์ลายมือชื่อดิจิตอล การเข้ารหัสและการทำแฮชชิ่งซึ่งในชนิดนี้อัลกอริทึมต่าง ๆ จะถูกกำหนดโดยสถาบัน National Institute of Standards and Technology (NIST)

- ***PROV_MS_EXCHANGE Provider Type*** จะใช้สำหรับโปรแกรมไมโครซอฟท์เอกเชนจ์ (Microsoft Exchange) หรือว่าโปรแกรม (Application) ที่ทำหน้าที่คล้ายไมโครซอฟท์เมล์ (Microsoft Mail) ซึ่งชนิดโพรไวเดอร์ชนิดนี้จะสามารถทำการแลกเปลี่ยนคีย์ได้ ทำการลงนามลายมือชื่อดิจิตอลได้ เข้ารหัสข้อมูล และการทำแฮชชิ่งก็สามารถทำได้
- ***PROV_RSA_FULL Provider Type*** เป็นชนิดโพรไวเดอร์ที่ทั้งไมโครซอฟท์ (Microsoft) และอาร์เอสเอดาต้าเซอเคียวริตี้ (RSA Data Security) ช่วยกันจัดทำขึ้นมา ถือเป็นชนิดโพรไวเดอร์ที่ทำได้หลายอย่าง เช่น การทำการแลกเปลี่ยนคีย์ การลงนามลายมือชื่อดิจิตอล การเข้ารหัสข้อมูล และการทำแฮชชิ่งโดยการทำการต่าง ๆ ที่เกี่ยวข้องกับการเข้ารหัสนั้นจะใช้อัลกอริทึมที่เป็นของ อาร์เอสเอเป็นหลัก
- ***PROV_RSA_SIG Provider Type*** เป็นชนิดโพรไวเดอร์อีกชนิดหนึ่งที่ถูกจัดทำโดยไมโครซอฟท์และอาร์เอสเอดาต้าเซอเคียวริตี้ ซึ่งชนิดโพรไวเดอร์ชนิดนี้เป็นส่วนที่แตกออกมาจาก PROV_RSA_FULL แต่ว่าในชนิดนี้จะสามารถทำได้แค่การลงนามลายมือชื่อดิจิตอลและการทำแฮชชิ่งเท่านั้น
- ***PROV_SSL Provider Type*** เป็นชนิดโพรไวเดอร์ที่มีความสามารถในการทำตามมาตรฐานเอสเอสแอล (SSL - Secure Sockets Layer) ซึ่งชนิดโพรไวเดอร์ชนิดนี้สามารถที่จะทำคีย์เข้ารหัส คีย์ลายมือชื่อดิจิตอล คีย์การเข้ารหัสข้อมูลและการทำแฮชชิ่ง

Provider Name: เป็นชื่อที่ใช้บ่งบอกว่าเป็นซีเอสพีตัวใด มีตัวอย่างรายชื่อของไมโครซอฟท์ดังต่อไปนี้

Defined Name	*Value*
MS_DEF_PROV	"Microsoft Base Cryptographic Provider v1.0"
MS_ENHANCED_PROV	"Microsoft Enhanced Cryptographic Provider "
MS_DEF_RSA_SIG_PROV	"Microsoft RSA Signature Cryptographic Provider"
MS_DEF_RSA_SCHANNEL_PROV	"Microsoft RSA Schannel Cryptographic Provider"
MS_DEF_DSS_PROV	"Microsoft Base DSS Cryptographic Provider"
MS_DEF_DSS_DH_PROV	"Microsoft Base DSS and Diffie-Hellman Cryptographic Provider"
MS_ENH_DSS_DH_PROV	"Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider"
MS_DEF_DH_SCHANNEL_PROV	"Microsoft DH Schannel Cryptographic Provider"

ตารางที่ 4.1 ชื่อที่ใช้บ่งบอกว่าเป็น ซีเอสพี ตัวใด

4.3 การเรียกใช้คริปโตเอพีไอ

ในการเขียนโปรแกรมใด ๆ ให้สามารถที่จะเรียกใช้คริปโตเอพีไอให้ถูกต้องได้นั้นจำเป็นที่จะต้องรู้ถึงคำสั่งต่าง ๆ ในการเรียกใช้เอพีไอให้ถูกต้อง คำสั่งต่าง ๆ จะถูกกำหนดไว้ในไฟล์เฮดเดอร์ที่ชื่อ Wincrypt.h โดยจะอธิบายคำสั่งพื้นฐาน ดังนี้

1. ***CryptAcquireContext*** เป็นคำสั่งที่ใช้ในการจัดการเรียกใช้ซีเอสพี ใด ๆ โดยมีรูปแบบคำสั่ง คือ

BOOL WINAPI CryptAcquireContext(

HCRYPTPROV **phProv*, // out เป็นแฮนเดิล (Handle) ที่ใช้ควบคุมซีเอสพี ที่จะส่งคืนมาให้เมื่อมีการเรียกใช้

LPCTSTR *pszContainer*, // in เป็นชื่อของคีย์คอนเทนเนอร์ แต่ถ้าถูกกำหนดให้เป็น NULL จะหมายถึง ให้ใช้ส่วนเก็บคีย์ดีฟอลท์ (Default Key Container) แทน

LPCTSTR *pszProvider*, // in เป็นชื่อของโพรไวเดอร์ที่ต้องการจะเรียกใช้ ถ้ากำหนดเป็น NULL ก็จะจัดให้ใช้ผู้ให้บริการหลักแทน

DWORD *dwProvType*, // in เป็นชนิดของโพรไวเดอร์ที่ต้องการจะเรียกใช้

DWORD *dwFlags* // in เป็นแฟลก (Flags) ที่จะใช้ควบคุมการทำงาน

);

ค่าของ dwFlags ที่เป็นไปได้นั้น มีดังนี้คือ

- **CRYPT_VERIFYCONTEXT** จะใช้ต่อเมื่อการทำงานในขณะนั้นไม่ต้องการจะใช้ คีย์ส่วนตัว และการเข้าถึง คีย์ส่วนตัว ก็เป็นไปไม่ได้ด้วย
- **CRYPT_NEWKEYSET** ถ้าแฟลกนี้มีการใช้จะเป็นการสร้างคีย์คอนเทนเนอร์ขึ้นมาใหม่ ซึ่งคีย์คอนเทนเนอร์ที่สร้างขึ้นมาใหม่นั้นจะมีชื่อตาม pszContainer ที่กำหนดไว้ แต่ถ้าไม่กำหนดค่าตัวนี้นั้น จะทำให้เกิดการสร้างส่วนเก็บคีย์ดีฟอลท์ขึ้น
- **CRYPT_MACHINE_KEYSET** โดยปกติแล้วถ้าแฟลกนี้ไม่ได้ถูกกำหนดไว้คีย์ต่าง ๆ จะถูกเก็บไว้ในรีจีสทรีในส่วนของ HKEY_CURRENT_USER แต่ถ้าแฟลกนี้ได้ถูกกำหนดไว้จะทำให้คีย์ต่าง ๆ ถูกเก็บไว้ในรีจีสทรีในส่วนของ HKEY_LOCAL_MACHINE
- **CRYPT_DELETEKEYSET** จะเป็นการลบคีย์คอนเทนเนอร์ที่มีการกำหนดไว้ใน pszContainer แต่ถ้าไม่ได้กำหนดไว้จะเป็นการลบส่วนเก็บคีย์ดีฟอลท์ ซึ่งในการลบนั้นจะทำให้คีย์ต่าง ๆ ที่อยู่ภายในคีย์คอนเทนเนอร์นั้น ๆ ถูกลบไปด้วย และค่า phProv ที่จะต้องให้ออกมาเป็นแอดเดรส (Address) นั้นจะไม่ถูกส่งคืนออกมา เพราะถือว่าได้ลบคีย์คอนเทนเนอร์ไปแล้ว คำสั่งต่อไปที่ควรทำต่อจากการลบก็คือ CryptReleaseContext เพื่อคืนการควบคุม

ค่าที่จะต้องคืนกลับมาหลังจากการเรียกใช้นั้น (Return Value) ถ้าการเรียกใช้ประสบผลสำเร็จจะคืนค่ากลับมาเป็นจริง (True) แต่ถ้าหากเกิดการเรียกใช้ผิดพลาดจะให้ค่ากลับเป็นเท็จ (False) โดยถ้าเกิดข้อผิดพลาดขึ้นจะต้องเรียกใช้คำสั่ง GetLastError เพื่อดูว่าเกิดผิดพลาดประการใด

2.CryptGetProvParam เป็นฟังก์ชันที่ใช้ในการอ่านค่าต่าง ๆ ของผู้ให้บริการ (Provider) ที่กำหนด มีรูปแบบดังนี้

BOOL WINAPI CryptGetProvParam(

HCRYPTPROV *hProv*, // in ค่าแฮนเดิลของซีเอสพี ตัวที่ต้องการอ่านค่า

DWORD *dwParam*, // in เพื่อบอกว่าจะอ่านค่าอะไร

BYTE **pbData*, // out เป็นตัวชี้ (Pointer) ที่ชี้ไปที่แหล่งพักข้อมูล (Buffer) ที่ต้องการให้เก็บข้อมูล

DWORD **pcbData*, // in/out เป็นตัวชี้ที่ชี้ไปยังตัวแปรที่เก็บค่าขนาดของข้อมูลใน
// แหล่งพักข้อมูลซึ่งมีหน่วยเป็นไบต์

DWORD *dwFlags* // in ค่าแฟลกเพื่อใช้กำหนดรูปแบบต่าง ๆ

);

ค่าของ dwParam เป็นได้ดังต่อไปนี้

- **PP_CONTAINER** หมายถึง ชื่อของคีย์คอนเทนเนอร์ค่าที่ได้กลับมาจะเป็นสตริงเก็บไว้ในแหล่งพักข้อมูล
- **PP_ENUMALGS** จะใช้เพื่อต้องการรู้ว่ามีอัลกอริทึมใดที่เก็บไว้ในซีเอสพีนั้น ๆ แต่การเรียกใช้ในครั้งแรกนั้น จำเป็นที่ต้องระบุแฟลก CRYPT_FIRST เพื่อให้ได้อัลกอริทึมตัวแรกสุดมา หลังจากอ่านตัวแรกแล้วก็สามารถวนลูปเพื่ออ่านอัลกอริทึมที่เหลือได้จนกว่าจะเกิด ERROR_NO_MORE_ITEMS ซึ่งแสดงว่าเป็นอัลกอริทึมตัวสุดท้าย ถ้าอ่านได้เป็นผลสำเร็จจะทำให้ได้ข้อมูลเกี่ยวกับอัลกอริทึมมาซึ่งมีรูปแบบดังนี้

ALG_ID aiAlgid; เป็นตัวแปรอัลกอริทึม (Algorithm Identifier) ซึ่งสามารถนำค่านี้ไปกำหนดอัลกอริทึมได้

DWORD dwBits; เป็นจำนวนบิตของคีย์ที่ใช้ในอัลกอริทึมนั้น แต่ถ้าเป็นวิธีแฮช
ค่านี้จะบ่งบอกถึงจำนวนบิทที่แฮชฟังก์ชันได้ทำเป็นผลสำเร็จ

DWORD dwNameLen; เป็นจำนวนตัวอักษรของชื่ออัลกอริทึมซึ่งจะรวมตัวอักษรปิดท้าย (Terminate) ด้วย

CHAR szName[dwNameLen]; เป็นชื่อของอัลกอริทึมที่เป็นสตริง

- **PP_ENUMCONTAINERS** เป็นชื่อของคีย์คอนเทนเนอร์ซึ่งจะใช้การวนลูปเพื่ออ่านคีย์คอนเทนเนอร์ให้ครบทุกตัว
- **PP_IMPTYPE** เป็นชนิดของซีเอสพี ว่าถูกจัดทำขึ้นเป็นแบบใด โดยจะให้ค่าเป็นได้ดังนี้

1. *CRYPT_IMPL_HARDWARE*
2. *CRYPT_IMPL_SOFTWARE*
3. *CRYPT_IMPL_MIXED*
4. *CRYPT_IMPL_UNKNOWN*

- **PP_NAME** หมายถึง ให้อ่านชื่อของซีเอสพี
- **PP_VERSION** ให้อ่านเวอร์ชัน (Version) ของซีเอสพี ซึ่งค่าที่ได้ออกมาจะเป็น DWORD และ ไบต์ที่ต่ำกว่าจะเป็น เวอร์ชันรอง (minor version) ของซีเอสพี ส่วนไบต์ที่สูงกว่าจะเป็น เวอร์ชันหลัก (major version) ของซีเอสพี
- **PP_KEYSET_SEC_DESCR** บ่งบอกว่าจำเป็นจะต้องมีสัญลักษณ์ความปลอดภัย (Security Descriptor) ในการเข้าถึงรีจีสทรีที่ได้มีการเก็บคีย์ไว้ แต่แฟลกนี้ไม่สามารถใช้ได้ใน Windows 95

ในการติดต่อกับซีเอสพี นั้นจะเริ่มจากการที่โปรแกรมใดๆ นั้นได้เรียกใช้คริปโตเอพีไอโดยการผ่านคำสั่ง CryptAcquireContext เพื่อที่จะใช้จัดการกับซีเอสพีใด ๆ โดยเมื่อเรียกใช้คำสั่งนี้แล้วจะเป็นการบ่งถึงซีเอสพีที่กำหนดไว้ในตัวแปรที่ส่งไป ซึ่งสิ่งที่ส่งไปไม่ได้มีเพียงแค่ชื่อของซีเอสพีเท่านั้นแต่จะรวมถึง คีย์คอนเทนเนอร์ที่อยู่ภายในซีเอสพีนั้นด้วย ในการเรียกใช้คำสั่ง CryptAcquireContext นั้นจะต้องบ่งบอกถึง ชนิดผู้ให้บริการ (Provider Type) แต่ชื่อผู้ให้บริการ (Provider Name) นั้นแล้วแต่ว่าจะกำหนดหรือไม่ก็ได้ดังนี้

- กำหนดชื่อผู้ให้บริการด้วยจะทำให้เกิดการหาซีเอสพีที่กำหนดไว้ และทำการโหลดเข้าสู่หน่วยความจำ จากนั้นจะต้องทำการคืนค่าแฮนเดิลให้กับโปรแกรม
- ไม่กำหนดชื่อผู้ให้บริการจะทำให้เกิดการหาซีเอสพีตัวแรกสุดที่อยู่ในรายการของผู้ให้บริการหลักของบุคคล ๆ นั้นที่ล็อกอิน (Log in) เข้าใช้งาน แต่ถ้าการหาจากผู้ให้บริการหลักของบุคคล ๆ นั้นเกิดความผิดพลาดขึ้นเช่น ยังไม่มีผู้ให้บริการหลักก็จะทำการหาจากรายการซีเอสพีของผู้ให้บริการหลักในเครื่องคอมพิวเตอร์เครื่องนั้น เมื่อหาเจอแล้ว ก็จะทำการโหลดซีเอสพีนั้นเข้าสู่เมโมรีและคืนค่าแฮนเดิล

4.4 การใช้คริปโตเอพีไอในการทำงานกับเอกสารสิทธิ์

คำศัพท์ที่เกี่ยวข้อง

CRL – Certificate Revocation List คือ รายชื่อของเอกสารสิทธิ์ที่ถูกนำกลับมาใช้ใหม่ เช่น ในกรณีที่เอกสารสิทธิ์นั้น เจ้าของอาจจะไม่ต้องการใช้อีกต่อไปก็จะสามารถนำกลับไปให้ผู้อื่นใช้ต่อไปได้

CTL – Certificate Trust List คือ รายชื่อของเอกสารสิทธิ์ที่สามารถเชื่อถือได้

คำสั่งต่าง ๆ สามารถแบ่งคำสั่งต่าง ๆ ที่สำคัญออกเป็นกลุ่มได้ดังนี้

Certificate Store Functions

CertOpenStore ใช้เพื่อทำการเปิดที่เก็บเอกสารสิทธิ์ (Certificate Store)โดยจะต้องกำหนดชนิดผู้ให้บริการที่เก็บเอกสารสิทธิ์ด้วยCertRegisterSystemStore ใช้เพื่อทำการลงทะเบียน (Register) ที่เก็บเอกสารสิทธิ์ โดยเราสามารถระบุได้ว่าจะให้เก็บไว้ตรงส่วนใด

CertSetStoreProperty ใช้เพื่อทำการตั้งค่าต่าง ๆ ให้กับที่เก็บเอกสารสิทธิ์

Certificate Functions

CertAddCertificateContextToStore ใช้เพื่อเพิ่มเอกสารสิทธิ์อันใหม่เข้าไปในที่เก็บเอกสารสิทธิ์

CertFindCertificateInStore ใช้สำหรับการหาเอกสารสิทธิ์ที่อยู่ในที่เก็บเอกสารสิทธิ์โดยในการหาครั้งแรกนั้น จะได้เอกสารสิทธิ์ตัวแรกออกมา และในครั้งต่อไปจะได้ตัวถัดไปเรื่อย ๆ ออกมา

CertDuplicateCertificateContext ใช้เมื่อต้องการที่จะทำสำเนาของเอกสารสิทธิ์ไว้

Certificate Revocation List Functions

CertAddCRLContextToStore เป็นการเพิ่ม CRL Context ลงไปในที่เก็บเอกสารสิทธิ์

CertAddCRLLinkToStore ใช้เมื่อต้องการเพิ่มลิงค์ (Link) ให้กับ CRL ในที่เก็บเอกสารสิทธิ์อันหนึ่งเพื่อชี้ไปยัง CRL อื่นในที่เก็บเอกสารสิทธิ์อื่น

CertDuplicateCRLContext เป็นฟังก์ชันที่ถูกเรียกใช้เมื่อต้องการทำสำเนาของ CRL

CertFindCRLInStore ใช้เมื่อต้องการหา CRL ที่มีอยู่ในที่เก็บเอกสารสิทธิ์

Certificate Trust List Functions

CertAddCTLContextToStore ใช้เมื่อต้องการเพิ่ม CTL Context ลงในที่เก็บเอกสารสิทธิ์

CertDuplicateCTLContext ใช้เมื่อต้องการจะทำสำเนา CTL Context

CertFindCTLInStore ใช้เมื่อต้องการหา CTL ที่อยู่ภายในที่เก็บเอกสารสิทธิ์

บทที่ 5
โอแอลอี (OLE)

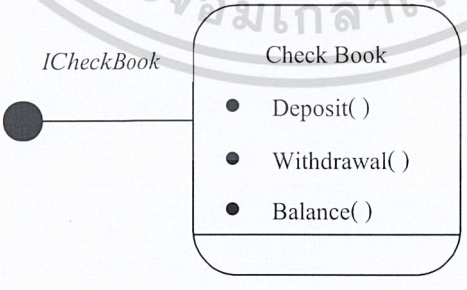
5.1 โอแอลอี

โอแอลอี เป็นลักษณะการเขียนโปรแกรมอย่างหนึ่ง โดยใช้รูปแบบของคอมออปเจ็กต์ (COM Object) โดยจะมีสองส่วนที่สำคัญคือส่วนที่เป็นโอแอลอีเซิร์ฟเวอร์ (OLE Server) และส่วนที่เป็นโอแอลอีไคลเอนต์ (OLE Client) โดยส่วนเซิร์ฟเวอร์ทำหน้าที่ให้บริการแก่ส่วนไคลเอนต์ ซึ่งส่วนไคลเอนต์นี้จะทำการฝังตัวลงไปบนโปรแกรม (Application) แล้วเรียกใช้การทำงานของส่วนเซิร์ฟเวอร์

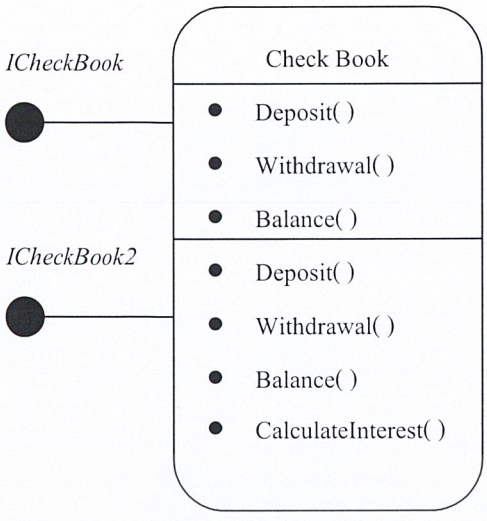
5.2 อินเทอร์เฟซ

รูปแบบการเขียนโปรแกรมแบบคอมออปเจ็กต์นั้นจะประกอบด้วยอินเทอร์เฟซ (Interface)ซึ่งทำหน้าที่ติดต่อระหว่างสองวัตถุ (Object) ใด ๆโดยเป็นกระบวนการที่นำเมธอด (Method) ที่เราต้องการมาใช้จากวัตถุที่เราทำงานอยู่ด้วยออกมา ในการทำงานนั้นจะต้องมีอย่างน้อยหนึ่งอินเทอร์เฟซก็คืออินเทอร์เฟซ *IUnknown* หรืออินเทอร์เฟซใดๆที่สืบทอดมาจากอินเทอร์เฟซ *IUnkown* ซึ่งเมื่อไคลเอนต์ทำการติดต่อกับอินเทอร์เฟซได้แล้ว ก็สามารถเรียกใช้เมธอดของเซิร์ฟเวอร์ได้ โดยผ่านทางอินเทอร์เฟซ เพื่อเข้าไปที่ตารางวีเทเบิล (vtable) ซึ่งทำหน้าที่เก็บ พอยเตอร์ (pointer) ที่ชี้ไปยังเมธอดของเซิร์ฟเวอร์

เมื่อคอมออปเจ็กต์มีการเปลี่ยนแปลงโดยการเพิ่มเมธอดต้องมีการสร้างอินเทอร์เฟซใหม่ซึ่งสืบทอดมาจากอินเทอร์เฟซเดิม เพื่อให้อินเทอร์เฟซใหม่สามารถเห็นเมธอดที่ทำการเพิ่มเข้าไปใหม่ได้ แต่ถ้ามีการปรับปรุงเมธอดที่มีอยู่แล้วต้องทำการเตรียมอินเทอร์เฟซใหม่ที่บรรจุเมธอดที่ปรับปรุงแล้ว ซึ่งไม่สามารถสืบทอดมาจากอินเทอร์เฟซเดิมได้ เพราะชื่อของเมธอดซ้ำกันแต่การทำงานไม่เหมือนกัน



รูปที่ 5.1 ตัวอย่างอินเทอร์เฟซของคอมอออปเจ็กต์



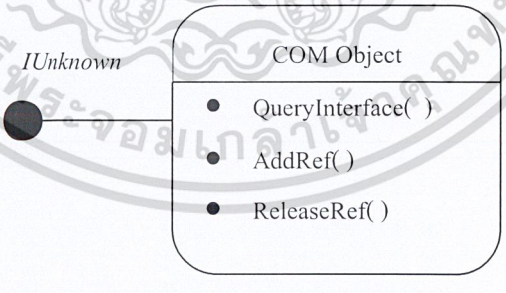
รูปที่ 5.2 แสดงการสร้างอินเทอร์เฟซใหม่เมื่อมีเมธอดใหม่

5.3 อินเทอร์เฟซ *IUnkown*

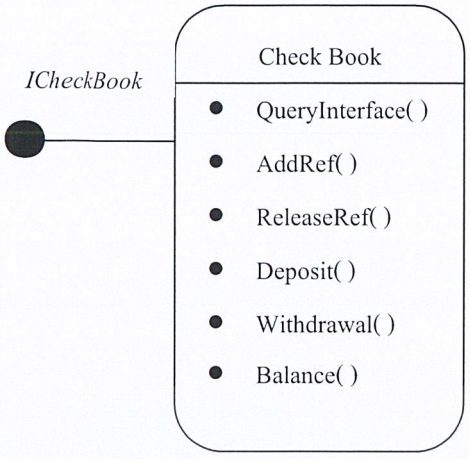
อินเทอร์เฟซ *IUnkown* ประกอบด้วยเมธอดหลักสามเมธอดด้วยกัน

1. *QueryInterface()* ใช้เพื่อดูว่าอินเทอร์เฟซที่ได้นี้เป็นอินเทอร์เฟซที่เราต้องการหรือไม่
2. *AddRef()* ทำการเพิ่มค่าเรเฟอเรนซ์เคาติง (Reference Counting) และสุดท้าย
3. *ReleaseRef()* ทำหน้าที่ลดค่าเรเฟอเรนซ์เคาติง

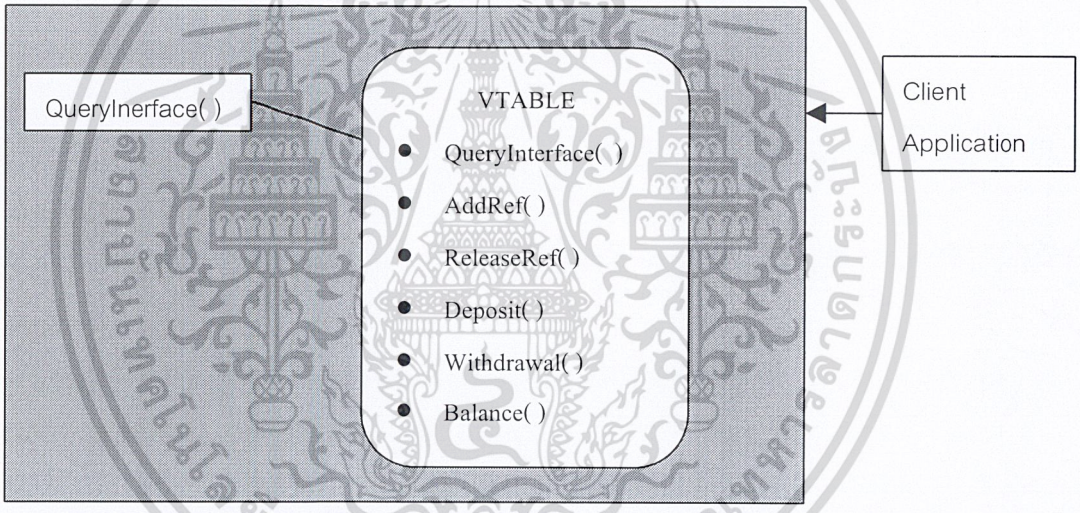
โดยเมื่อเซิร์ฟเวอร์ผ่านพอยเตอร์ออกมาต้องเพิ่มค่าเรเฟอเรนซ์เคาติงโดยเรียกผ่านเมธอด *AddRef()* เมื่อไคลเอนต์ทำงานกับเซิร์ฟเวอร์เสร็จสิ้นก็ลดค่าเรเฟอเรนซ์เคาติงลงโดยเรียกผ่านเมธอด *ReleaseRef()* โดยค่าของเรเฟอเรนต์เคาติงมีไว้เพื่อระบุว่ามีไคลเอนต์จำนวนเท่าใดที่ใช้อินเทอร์เฟซนี้บ้าง เมื่อค่าเรเฟอเรนต์เคาติงเป็นศูนย์ก็ไม่สามารถใช้เซิร์ฟเวอร์นี้ได้อีก



รูปที่ 5.3 แสดงอินเทอร์เฟซ IUnknown



รูปที่ 5.4 แสดงอินเทอร์เฟซที่สืบทอดมาจาก IUnknown



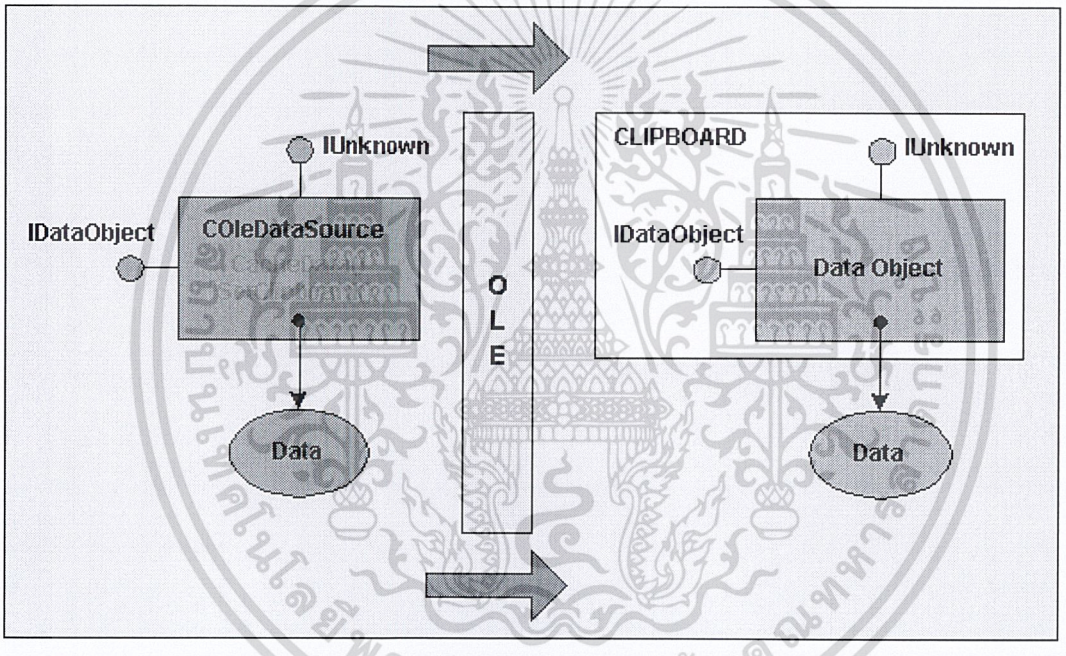
รูปที่ 5.5 แสดงลักษณะ VTABLE

บทที่ 6

คลาส ColeDataObject

6.1 คลาส COleDataObject

คลาส COleDataObject เป็นคลาสที่ MFC เตรียมไว้สำหรับการส่งผ่านข้อมูลในหลาย ๆรูปแบบที่มาจากคลิป บอร์ด ผ่านทางการลากและวาง (Drag & Drop) หรือเอกสารฝังตัวแบบ OLE ข้อมูลที่ได้มามีทั้งที่มา (Source) และ ที่หมาย (Destination) โดยที่มาของข้อมูลได้มาจากการสร้างคลาส ColeDataSource หรือผ่านทางอินเทอร์เฟซ IDataObject ดังรูป



รูปที่ 6.1 แสดงการทำงานของคลาส COleDataObject

โดยคลาส COleDataObject เตรียมไว้เพื่อใช้ในการระบุรูปแบบของข้อมูลที่ได้มา ข้อมูลที่ได้รับมาสามารถทำได้ในหลายรูปแบบ เช่น การใช้คลาส CFile การใช้โกลบอลเมมโมรี (HGLOBAL) หรือใช้รูปแบบโครงสร้าง STGMEDIUM

6.2 ฟังก์ชันสำคัญในคลาส COleDataObject

void COleDataObject :: Attach(LPDATAOBJECT lpDataObject , BOOL bAutoRelease = NULL)

lpDataObject เป็นตัวแปรชี้ไปที่ ข้อมูลใน OLE

bAutoRelease ค่าเป็น TRUE เมื่อต้องการให้ปล่อยข้อมูลใน OLE เมื่อ COleDataObject ถูกทำลาย และเป็น FALSE เมื่อต้องการเป็นอย่างอื่น

เป็นฟังก์ชันที่กำหนดจุดเริ่มต้นให้กับตัวแปรของคลาส COleDataObject ว่ารับค่าจากข้อมูลใน OLE ตัวใด โดยผ่านทางตัวแปร lpDataObject

void COleDataObject :: BeginEnumFormats( )

เป็นฟังก์ชันเพื่อเตรียมการใช้งานฟังก์ชัน GetNextFormat เพื่อเก็บค่าตำแหน่งของรูปแบบแรก

BOOL COleDataObject :: GetNextFormat(LPFORMATETC lpFormatEtc)

lpFormatEtc เป็นตัวแปรชี้ไปยังโครงสร้างแบบ FORMATETC เพื่อรับข้อมูลรูปแบบออกมาเมื่อฟังก์ชันทำการคืนค่า

เป็นฟังก์ชันเพื่อเก็บเอารูปแบบที่หาได้ออกมา คืนค่าเป็น ศูนย์ เมื่อไม่มีรูปแบบอื่น ๆอีกแล้ว

BOOL COleDataObject :: IsDataAvailable(CLIPFORMAT cfFormat, LPFORMATETC lpFormatEtc = NULL)

cfFormat ตัวแปรรูปแบบของคลิปบอร์ด

lpFormatEtc เป็นตัวแปรชี้ไปยังโครงสร้างแบบ FORMATETC ปรกติกำหนดเป็น NULL

เป็นฟังก์ชันใช้เพื่อตรวจสอบว่ามีคลิปบอร์ดในข้อมูล OLE หรือไม่ คืนค่าไม่เป็น ศูนย์ เมื่อมีข้อมูลในรูปแบบคลิปบอร์ดที่ต้องการ

HGLOBAL COleDataObject :: GetGlobalData(CLIPFORMAT cfFormat, LPFORMATETC lpFormatEtc = NULL);

cfFormat ตัวแปรรูปแบบของคลิปบอร์ด

lpFormatEtc เป็นตัวแปรชี้ไปยังโครงสร้างแบบ FORMATETC ปรกติกำหนดเป็น NULL

เป็นฟังก์ชันกำหนดโกลบอลเมมโมรีเพื่อรับค่าข้อมูลของรูปแบบคลิปบอร์ดที่กำหนดไว้ในตัวแปร HGLOBAL คืนค่าโดยไม่ใช่ NULL เมื่อเรียกใช้สำเร็จ

บทที่ 7
คลิปบอร์ด

7.1 วินโดว์คลิปบอร์ด

วินโดวส์คลิปบอร์ด เป็นตัวแทนการแลกเปลี่ยนข้อมูลระหว่างแอพพลิเคชั่นต่างๆ ก่อนที่จะมีวิธีลากแล้วปล่อยจากการฝังตัวของโปรแกรมแบบโอแอลอี ผู้ใช้ต้องใช้คลิปบอร์ดเช่น การตัด การคัดลอก การวาง เพื่อส่งข้อมูลจากแอพพลิเคชั่นหนึ่งไปยังอีกแอพพลิเคชั่นหนึ่ง หรือย้ายข้อมูลในแอพพลิเคชั่นเดียวกัน

ปัจจุบันนี้คลิปบอร์ดได้ถูกลืมไปแล้วเพราะถูกแทนที่โดยโอแอลอี ไม่ได้หมายความว่าแอพพลิเคชั่นจะไม่สนับสนุนการทำงานของคลิปบอร์ด แต่แนวคิดเกี่ยวกับคลิปบอร์ดก็ยังมีอยู่ถึงแม้ว่าแอพพลิเคชั่นจะแลกเปลี่ยนข้อมูลด้วยวิธีการขั้นสูงกว่าก็ตาม

จริงๆแล้วคลิปบอร์ดคืออะไร อาจจะเป็นสิ่งอำนวยความสะดวกของ Win32 ซึ่งแอพพลิเคชั่นสามารถวางข้อมูลไว้ได้ เช่น ข้อมูลสามารถเข้าถึงได้โดยทุกแอพพลิเคชั่น ข้อมูลสามารถเป็นได้หลายรูปแบบ บางชนิดสนับสนุนโดยระบบปฏิบัติการ บางชนิดสนับสนุนโดยแอพพลิเคชั่น

7.2 คลิปบอร์ฟอร์แมต

แอพพลิเคชันเก็บข้อมูลลงในคลิปบอร์ดโดยใช้ฟังก์ชั่น SetClipboardData ทั้งยังจัดการควบคุมวัตถุที่เป็นข้อมูล ฟังก์ชั่นนี้สามารถรับพารามิเตอร์ที่เจาะจงรูปแบบของข้อมูลได้ด้วย ทำให้แอพพลิเคชั่นต่างสามารถจัดการข้อมูลในหลายๆรูปแบบได้ ตัวอย่าง เช่น เวิร์ดโปรเซสเซอร์สามารถวางข้อมูลบนคลิปบอร์ดโดยใช้ทั้งรูปแบบส่วนตัวและรูปแบบเพลนเท็กซ์ ซึ่งเพลนเท็กซ์สามารถถูกใช้ได้โดยโปรแกรมอื่นเช่น Notepad

มีรูปแบบของคลิปบอร์ดอยู่สามอย่างที่สามารถรวมเข้าไปในแอพพลิเคชั่นได้คือ

1.รูปแบบมาตรฐาน

2.รูปแบบที่ขึ้นทะเบียนไว้

3.รูปแบบส่วนตัว

7.2.1 สแตนดาร์ดคลิปบอร์ดฟอร์แมต (รูปแบบคลิปบอร์ดมาตรฐาน)

รูปแบบคลิปบอร์ดมาตรฐานมากมายที่มีอยู่ถูกนิยามโดยสัญลักษณ์ที่คงที่ รูปแบบเหล่านี้ถูกรวบรวมไว้ในตาราง ในกรณีที่แอพพลิเคชั่นถูกกำหนดให้หาตัวควบคุมของรูปแบบเฉพาะขณะเรียกฟังก์ชั่น SetCliboardData รูปแบบการควบคุมจะถูกบ่งชี้

Format Type	*Description*
Text Formats	
CF_OEMTEXT	Text containing characters from the OEM character set
CF_TEXT	Text containing characters from the ANSI character set
CF_UNICODETEXT	Text containing Unicode characters
Bitmap formats	
CF_BITMAP	Device-dependent bitmap (HBITMAP)
CF_DIB	Device independent bitmap (HBITMAPINFO)
CF_TIFF	Tagged Image File Format
Metafile formats	
CF_ENHMETAFILE	Enhanced metafile (HENHMETAFILE)
CF_METAFILEPICT	Windows Metafile (METAFILEPICT)
Substitute formats for private formats	
CF_DSPBITMAP	Bitmap representation of private data
CF_DSPENHMETAFILE	Enhanced metafile representation of private data
CF_DSPMETAFILEPICT	Metafile representation of private data
CF_DSPTEXT	Text representation of private data
Sound formats	
CF_RIFF	Resource Interchange File Format
CF_WAVE	Standard wave file format audio data
Special formats	
CF_DIF	Data Interchange Format from Software Arts
CF_OWNERDISPLAY	Data displayed by the owner of the clipboard data
CF_PALETTE	Color palette (HPALETTE)
CF_PENDATA	Microsoft Pen Extensions data
CF_PRIVATEFIRST through CF_PRIVATELAST	Private data
CF_SYLK	Microsoft Symbolic Link format

Windows 95 only formats	
C F _ G D I O B J F I R S T t h r o u g CF_GDIOBJLAST	Application-defined GDI objects
CF_HDROP	List of files (HDROP)
CF_LOCALE	Locale information for CF_TEXT data

ตารางที่ 7.1 แสดงรูปแบบคลิปบอร์ดมาตรฐาน

วินโดวส์สามารถสร้างข้อมูลในรูปที่ไม่เฉพาะเจาะจงสำหรับโปรแกรมได้โปรแกรมหนึ่ง ตัวอย่าง เช่น ถ้าแอพลิเคชั่นเก็บข้อมูลในรูปแบบ CF_TEXT วินโดวส์สามารถแปลงข้อมูลในรูปแบบ CF_OEMTEXT ไปเป็นรูปแบบที่โปรแกรมอื่นต้องการได้ วินโดวส์สามารถแสดงการเปลี่ยนแปลงของรูปแบบระหว่างรูปแบบเท็กซ์ CF_TEXT, CF_OEMTEXT และ (ภายใต้วินโดว์เอ็นที) CF_UNICODETEXT; รูปแบบบิตแมบ CF_BITMAP และ CF_DIB; และรูปแบบเมตาไฟล์ CF_ENHMETAFILE และ CF_METAFILEPICT ท้ายที่สุดวินโดวส์สามารถสร้างรูปแบบ CF_PALETTE จากรูปแบบ CF_DIB ได้

7.2.2 รีจิสเตอร์ฟอร์แมต (รูปแบบที่ถูกขึ้นทะเบียน)

แอพพลิเคชั่นอื่นๆซึ่งต้องการวางข้อมูลลงบนคลิปบอร์ดในรูปแบบอื่นๆนอกเหนือจากรูปแบบมาตรฐานสามารถขึ้นทะเบียนรูปแบบคลิปบอร์ดใหม่โดยใช้ฟังก์ชั่น RegisterClipbioardFormat ตัวอย่าง เช่น แอพพลิเคชั่นต้องการที่จะวางรูปแบบ RTF Text บนคลิปบอร์ดจะต้องทำการขึ้นทะเบียนดังนี้ cfRTF = RegisterClipboardFormat ("Rich Text Format")ถ้าหลายๆแอพพลิเคชั่นรียก RegisterClipboarFormat ด้วยชื่อรูปแบบเดิม รูปแบบจะถูกขึ้นทะเบียนได้ครั้งเดียวเท่านั้น

มีหลายรูปแบบคลิปบอร์ดถูกขึ้นทะเบียนโดนวินโดว์ ตัวอย่างเช่น รูปแบบที่ถูกขึ้นทะเบียนบางอันเกี่ยวข้องกับโอแอลอี บางรูปแบบเกี่ยวข้องกับเชลล์ของวินโดว์95 ชื่อของรูปแบบที่ถูกขึ้นทะเบียนสามารถเรียกดูได้โดยเรียกใช้ฟังก์ชั่น GetClipboardFormatName

7.2.3 ไฟรเวทฟอร์แมต (รูปแบบส่วนตัว)

บางครั้งไม่จำเป็นที่แอพพลิเคชั่นขึ้นทะเบียนรูปแบบคลิปบอร์ดใหม่ กรณีนี้เมื่อคิลบบอร์ดเคยถูกใช้ ตัวอย่าง เช่น การส่งข้อมูลภายในแอพพลิเคชั่นเดียวกันและข้อมูลไม่ได้ถูกใช้โดยแอพพลิเคชั่นอื่น แอพพลิเคชั่นนั้นกำหนดให้เป็นรูปแบบส่วนตัว แอพพลิเคชั่นนั้นสามารถใช้ CF_PRIVATEFIRST จนถึง CF_PRIVATELAST

ตามระเบียบการทำให้ตัวดูคลิปบอร์ดแสดงข้อมูลที่ถูกเก็บไว้ในรูปแบบส่วนตัวได้นั้น เจ้าของคลิปบอร์ดต้องแสดงข้อมูลในรูปแบบที่แสดงออกมาได้เหล่านี้ CF_DSPBITMAP, CF_DSPTEXT, CF_DSPMETAFILEPICT หรือ

CF_DSPENHMETAFILE รูปแบบเหล่านี้เป็นตัวบอกถึงส่วนพื้นฐาน (CF_BITMAP, CF_TEXT, CF_METAFILEPICT, and CF_ENHMETAFILE) นอกจากว่าจะถูกใช้เพื่อแสดงผลและไม่ใช่เพื่อการวาง

7.3 การทำงานของคลิปบอร์ด

ตามระเบียบการใช้งานของคลิปบอร์ด แอพพลิเคชั่นต้องทำงานหลายๆอย่าง ได้แก่ การสร้างข้อมูลคลิปบอร์ด, การขอสิทธ์การใช้ของคลิปบอร์ด, การส่งผ่านข้อมูล, และการตอบสนองเหตุการณ์ที่เกี่ยวเนื่องกับคลิปบอร์ด แอพพลิเคชั่นต้องจัดการกับส่วนติดต่อผู้ใช้ จัดการกับคำส่งของผู้ใช้ที่เจาะจงเกี่ยวกับคลิปบอร์ด (เช่นคำสั่ง ภายใต้เมนูแก้ไข)

7.3.1 การส่งผ่านข้อมูลไปยังคลิปบอร์ด

ก่อนที่ข้อมูลจะถูกส่งไปยังคลิปบอร์ด แอพพลิเคชั่นต้องทำสองสิ่ง 1.วัตถุข้อมูลต้องถูกกำหนดไว้ 2.สิทธ์การใช้คลิปบอร์ดต้องใช้ได้

วัตถุข้อมูลต้องมีตัวควมคุม ตัวควบคุมนี้สามารถอ้างถึงกลุ่มของหน่วยความจำที่ถูกจองโดยการใช้ฟังก์ชัน GlobalAlloc ด้วยแฟลก GMEM_MOVEABLE และ GMEM_DDESHARE (สถานะของแฟลก GMEM_DDESHARE ไม่ได้แสดงถึงกลุ่มของหน่วยความจำที่ใช้ร่วมกันระหว่างแอพพลิเคชั่น) แอพพลิเคชั่นได้รับสิทธิ์การเปิดคลิปบอร์ดโดยการเรียกใช้ฟังก์ชัน OpenClipboard และทำให้คลิปบอร์ดว่างเปล่าโดยเรียกใช้ฟังก์ชัน EmptyClipboard ตัวควบคุมทั้งหมดที่ถูกส่งไปยังคลิปบอร์ดก่อนหน้านี้จะถูกปล่อย ขั้นต่อไปแอพพลิเคชั่นส่งข้อมูลไปยังคลิปบอร์ดโดยใช้ฟังก์ชัน SetClipboardData และ ปิดคลิปบอร์ดโดยเรียกฟังก์ชัน CloseClipboard แอพพลิเคชั่นสามารถเรียกฟังก์ชัน SetClipboardData ได้หลายๆครั้งถ้ามีข้อมูลอยู่หลายๆรูปแบบ ตัวอย่าง เช่นแอพพลิเคชั่นอาจจะเรียก SetClipboardData โดยใช้รูปแบบ CF_DIB และ CF_ENHMETAFILE เพื่อจัดการรูปภาพทั้งในรูปแบบบิตแมบ และรูปแบบเมตาไฟล์

7.3.2 การวางข้อมูลจากคลิปบอร์ด

แอพพลิเคชั่นสามารถใช้ฟังก์ชัน IsClipboardFormatAvailable เพื่อตรวจสอบดูว่าข้อมูลในรูปแบบที่กำหนดสามารถใช้ได้บนคลิปบอร์ดหรือไม่ ถ้ามันต้องการที่จะได้รับสำเนาของข้อมูลบนคลิปบอร์ดแอพพลิเคชั่นสามมารถเรียก OpenClipboard, ตามด้วยการเรียก GetClipboardData แอพพลิเคชั่นควรจะคัดลอกข้อมูลที่ตัวควบคุมนั้นจัดการอยู่ทันที โดยต้องทำก่อนการเรียกฟังก์ชัน CloseClipboard หลังจากการเรียกฟังก์ชัน CloseClipboard แอพพลิเคชั่นอื่นสามารถทำให้คลิปบอร์ดว่างได้ การส่งค่าต่างๆผ่านฟังก์ชัน GetClipboardData จะไม่ผลทันที

ฟังก์ชัน IsClipboardFormatAvailable สามารถใช้เพื่ออับเดทไอเทมที่ใช้ปรับแต่งเมนูได้ด้วย ตัวอย่าง เช่น ถ้าฟังก์ชัน IsClipboardFormatAvailable บอกว่าไม่มีข้อมูลคลิปบอร์ดในรูปแบบที่ใช้งานได้ แล้วแอพพลิเคชั่นจะเข้าใจแอพพลิเคชั่นจะทำการปิดคำสั่งวาง

แอพพลิเคชั่นสามารถรับข้อมูลเกี่ยวกับรูปแบบข้อมูลที่ใช้ได้ในคลิปบอร์ดโดยเรียกฟังก์ชัน CountClipboardFormats หรือ EnumClipboardFormats

7.3.3 คลิปบอร์ดเมสเสจ

มีหลายวินโดวส์เมสเสจที่ถูกรวมเข้าไปไว้ในคลิปบอร์ด เมสเสจ WM_DESTROYCLIPBOARD ถูกส่งไปยังเจ้าของคลิปบอร์ดเมื่อสารบัญของคลิปบอร์ดถูกทำลาย ในการตอบรับเมสเสจนี้ แอพพลิเคชั่นอาจจะปลดปล่อยทรัพยากรที่นำไปใช้ในการจัดการหรือการวาดไอเทมของคลิปบอร์ด

กลุ่มของเมสเสจที่ถูกส่งไปยังแอพพลิเคชั่นที่ข้อมูลอยู่บนคลิปบอร์ดโดยใช้รูปแบบ CF_OWNERDISPLAY โดยประกอบด้วย WM_ASKCBFORMATNAME, WM_DRAWCLIPBOARD, WM_HSCROLLCLIPBOARD, WM_VSCROLLCLIPBOARD, และ WM_PAINTCLIPBOARD ส่วนกลุ่มของเมสเสจอื่นถูกส่งหรือถูกใช้โดยโปรแกรมคลิปบอร์ดวิวเวอร์

7.3.คลิปบอร์ดวิวเวอร์

clipboard viewer คือโปรแกรมที่สามารถแสดงข้อมูลปัจจุบันของคลิปบอร์ดได้ ตัวอย่างของ clipboard viewer ได้แก่โปรแกรม Windows Clipboard Viewer

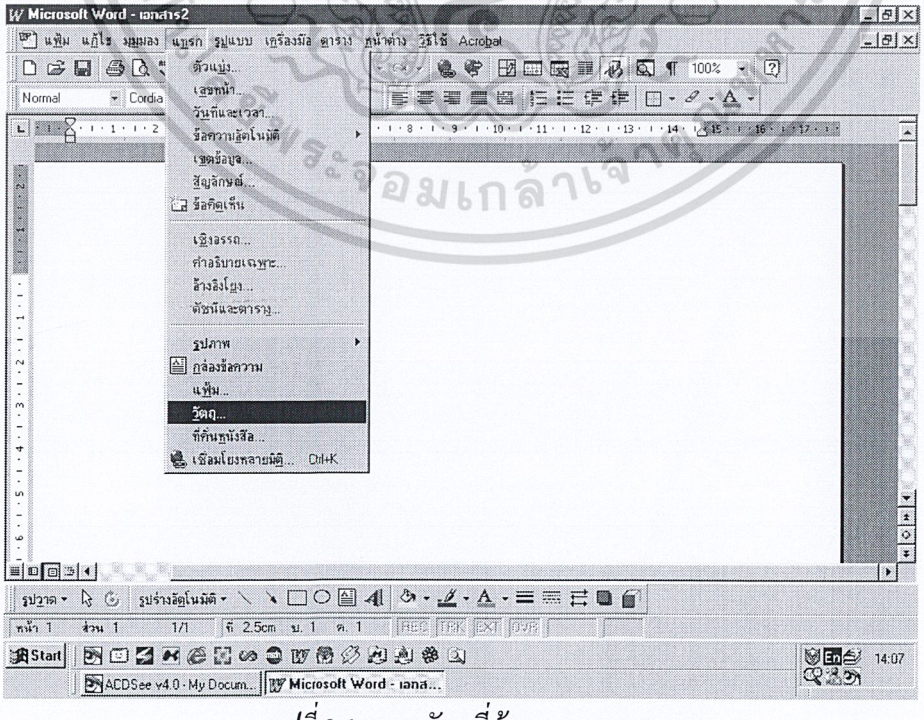
clipboard viewer มีไว้เพียงเพื่อความสะดวกสบายของผู้ใช้เท่านั้น และไม่ทำลายหรือเปลี่ยนแปลงการทำงานของคลิปบอร์ด

สามารถมีหลายๆ clipboard viewers ในการทำงานหนึ่ง แอพพลิเคชั่นหนึ่งใส่วินโดวส์ลงห่วงโซ่ของ clipboard viewers โดยการเรียกฟังก์ชัน SetClipboardViewer ด้วยตัวควบคุมของวินโดวส์ เมื่อเราเติมลงไปครั้งหนึ่ง clipboard viewer จะได้รับข้อความ WM_CHANGECBCHAIN และ WM_DRAWCLIPBOARD clipboard viewer สามารถถอดถอนตัวเองจากห่วงโซ่โดยการเรียกฟังก์ชัน ChangeClipboardChain

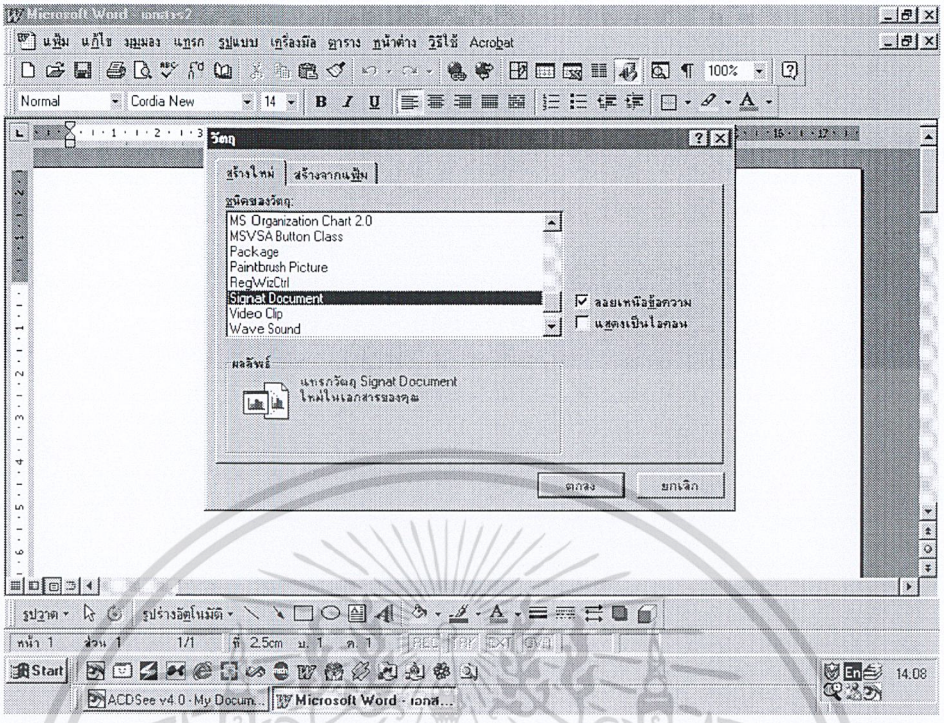
บทที่ 8
การสร้างแถบเครื่องมือบนโปรแกรมไมโครซอฟท์เวิร์ด

การสร้างแถบเครื่องมือบนโปรแกรมไมโครซอฟท์เวิร์ดเพื่อความสะดวกในการเรียกใช้โปรแกรม ISAG Signature โดยสร้างรูปแบบแมโครเป็นตัวเก็บลักษณะการทำงานของโปรแกรม ISAG Signature

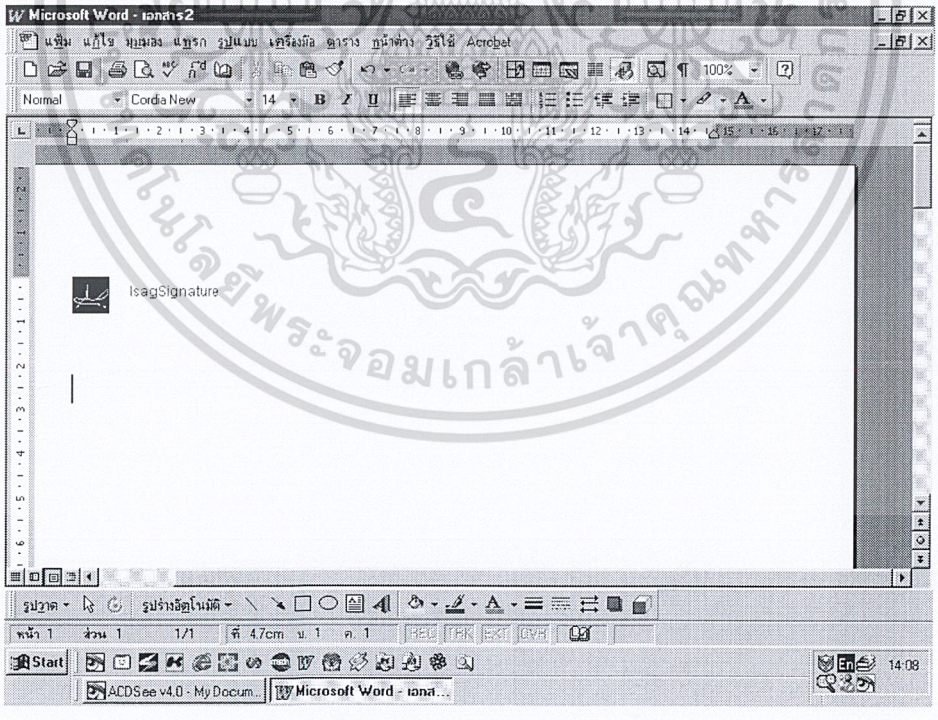
1. เลือก แทรก เลือก วัตถุ
2. หน้าต่าง วัตถุ เลือก Signet Document
3. เลือก เครื่องมือ เลือก แมโคร เลือก บันทึกแมโครใหม่
4. หน้าต่าง บันทึกแมโคร พิมพ์ชื่อ ISAG
5. เลือก เครื่องมือ เลือก กำหนดเอง
6. หน้าต่าง กำหนดเอง ที่แถบเครื่องมือ เลือก สร้าง พิมพ์ชื่อ ISAG ที่หน้าต่าง แถบเครื่องมือใหม่
7. หน้าต่าง กำหนดเอง ที่คำสั่ง เลือก ประเภท แมโคร เลือกคำสั่ง ISAG
8. ลากคำสั่งไปใส่ในแถบเครื่องมือ ISAG
9. เลือก แก้ไขการเลือก ที่ชื่อพิมพ์ SIGNATURE เลือก เปลี่ยนภาพลักษณะของปุ่ม เลือกรูป
10. ลาก แถบเครื่องมือ ISAG ไว้บน แถบเครื่องมือบนโปรแกรมไมโครซอฟท์เวิร์ด
11. เลือก แฟ้ม เลือก บันทึกเป็น หน้าต่างบันทึกเป็น พิมพ์ชื่อแฟ้ม ISAG ที่จัดเก็บเป็นชนิด เลือก แม่แบบเอกสาร ที่บันทึกใน เลือก Microsoft Office\Office\STARTUP ทำการบันทึก



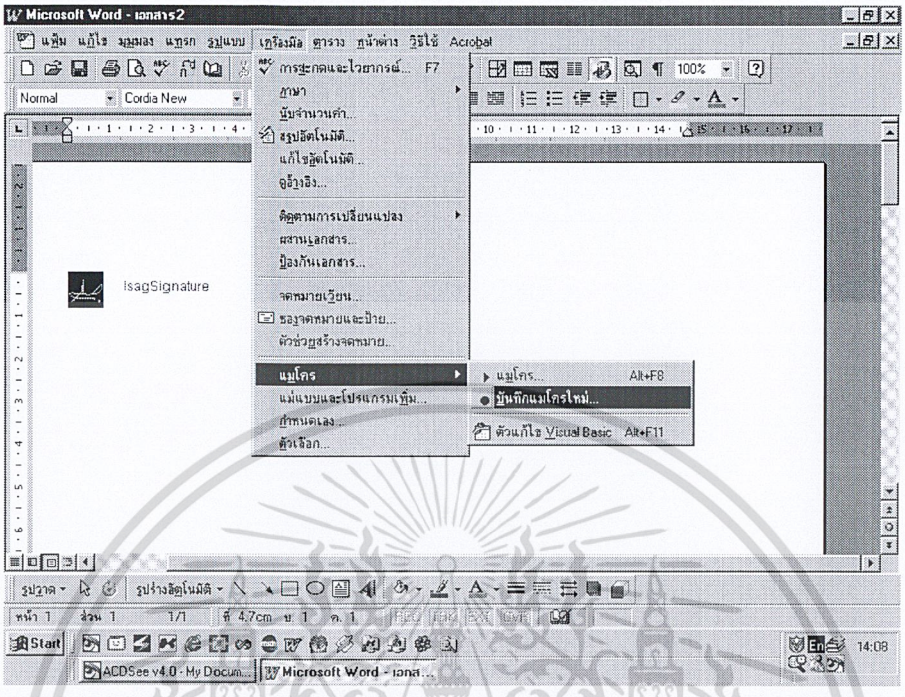
รูปที่ 8.1 แทรกวัตถุที่ต้องการลงบนเอกสาร



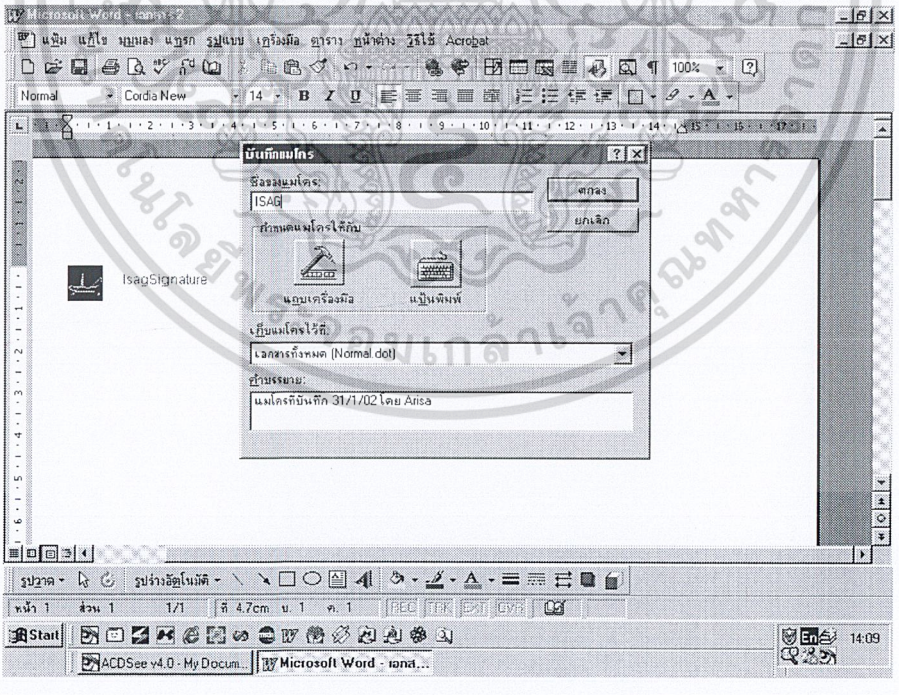
รูปที่ 8.2 เลือกโปรแกรม IsagSignature



รูปที่ 8.3IsagSignature บนเอกสาร



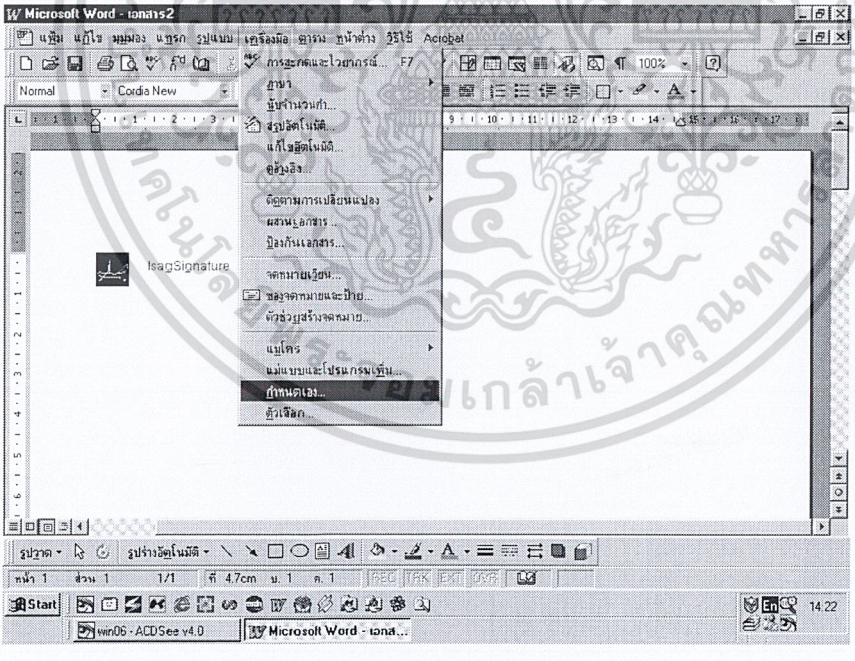
รูปที่ 8.4 เปิดแมโครเพื่อบันทึก



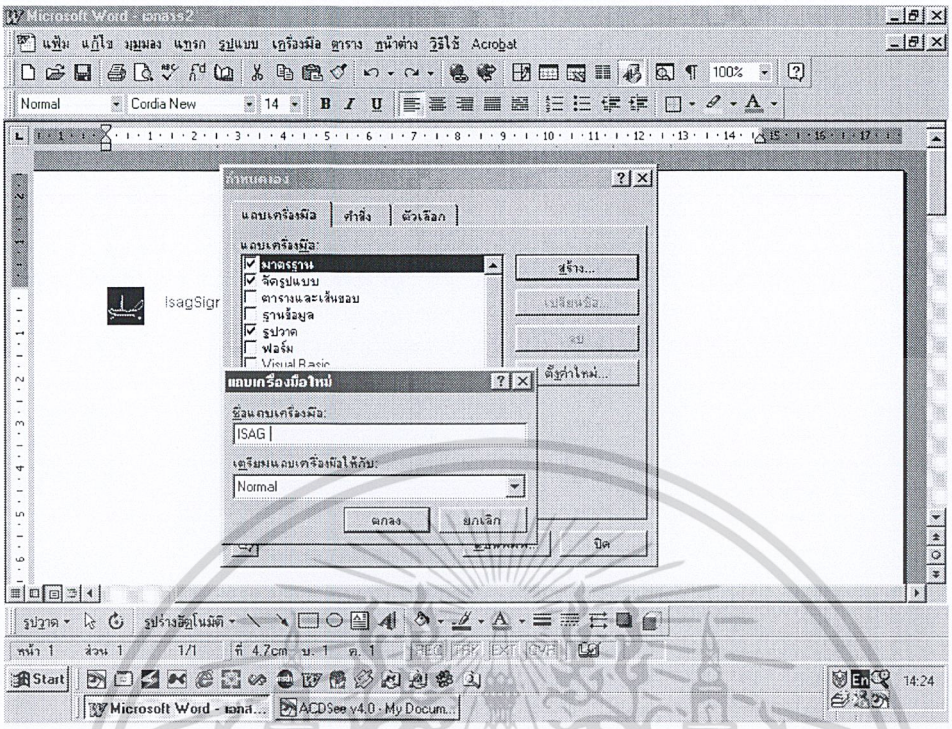
รูปที่ 8.5 กำหนดชื่อให้แมโคร



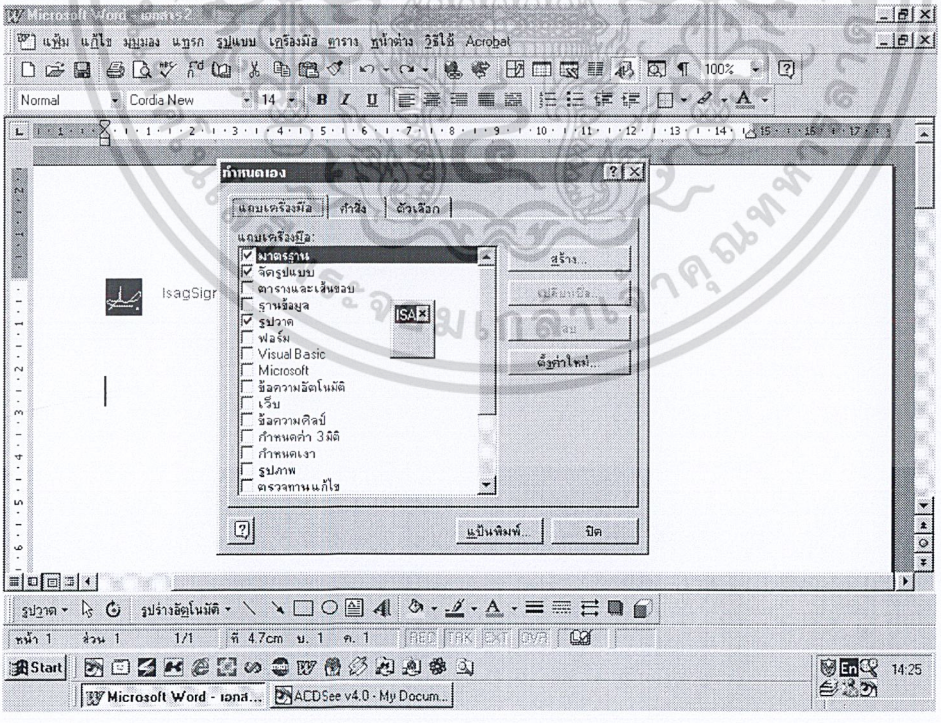
รูปที่ 8.6 ปรากฏหน้าต่างการบันทึกแมโคร



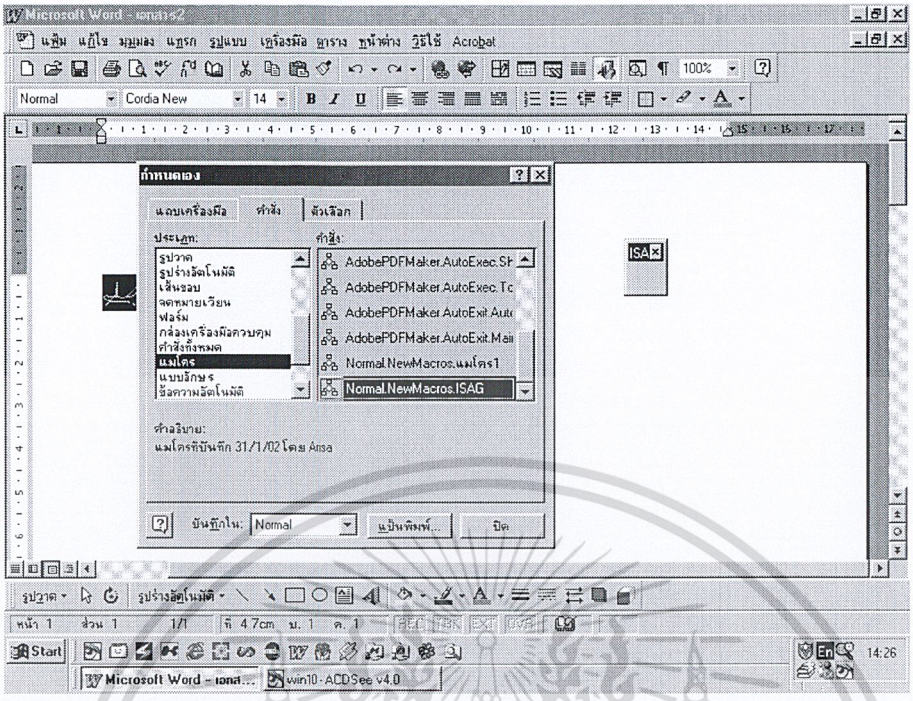
รูปที่ 8.7 กำหนดคุณสมบัติของแมโคร



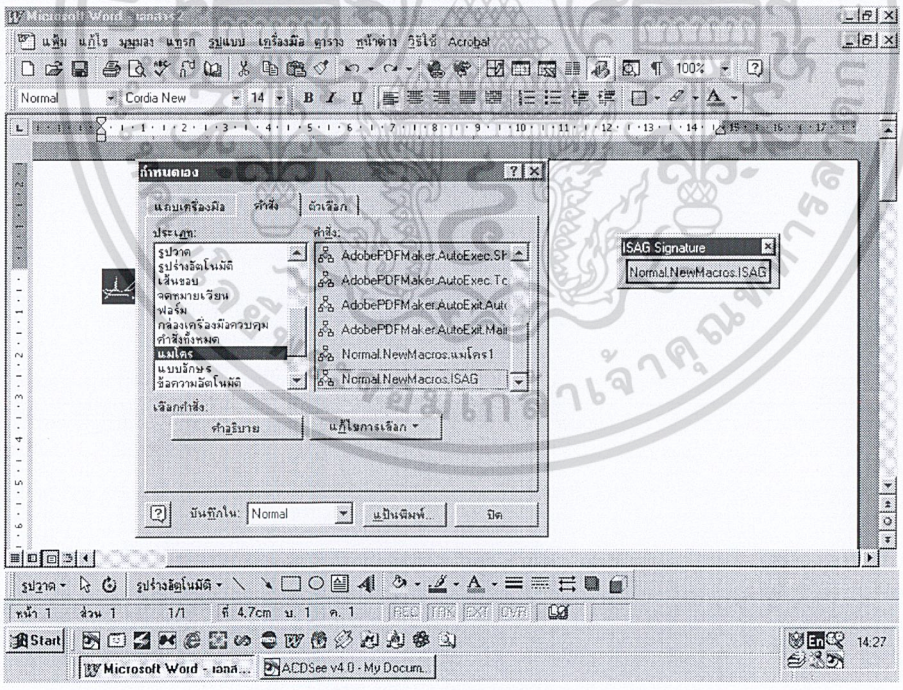
รูปที่ 8.8 กำหนดชื่อให้แถบเครื่องมือ



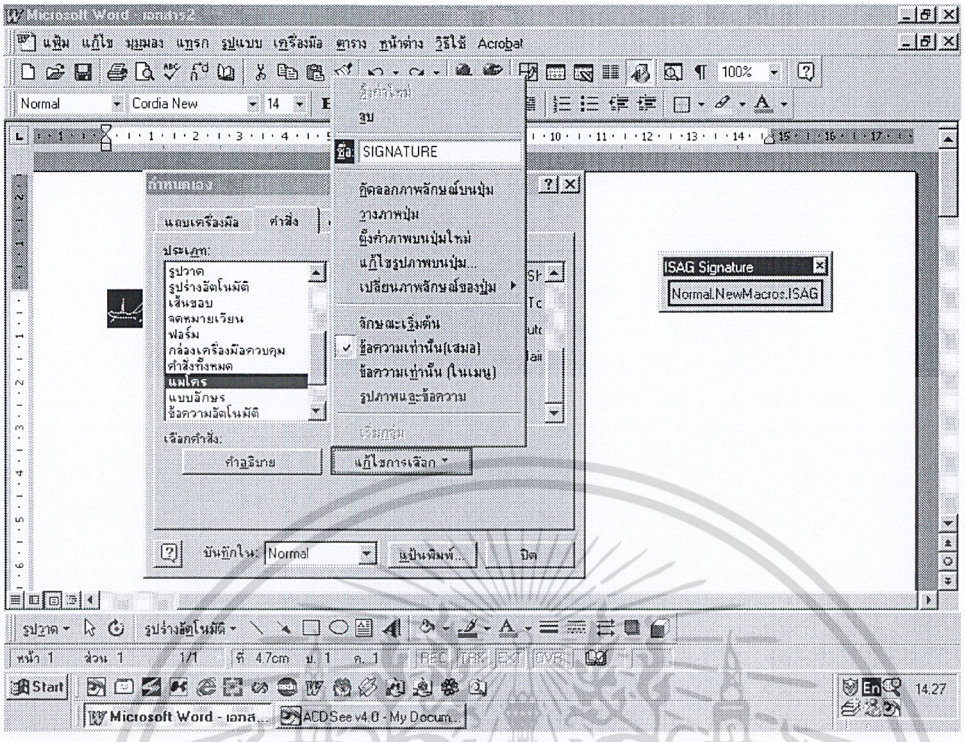
รูปที่ 8.9 แถบเครื่องมือใหม่บนเอกสาร



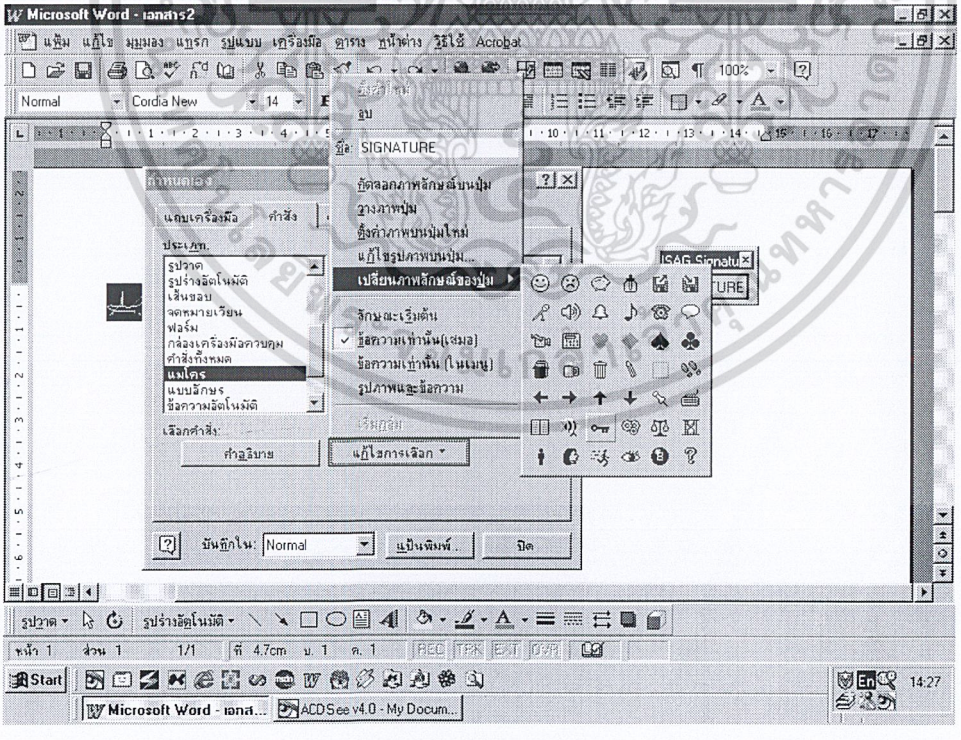
รูปที่ 8.10 เลือกแมโครที่ทำการบันทึก IsagSignature



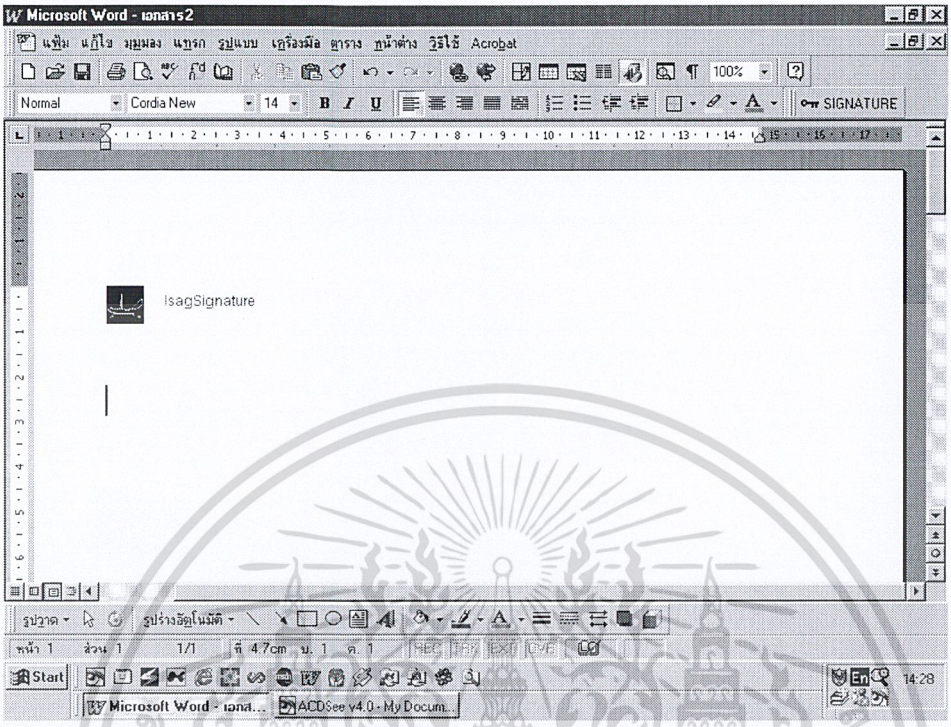
รูปที่ 8.11 ได้แถบเครื่องมือ IsagSignature



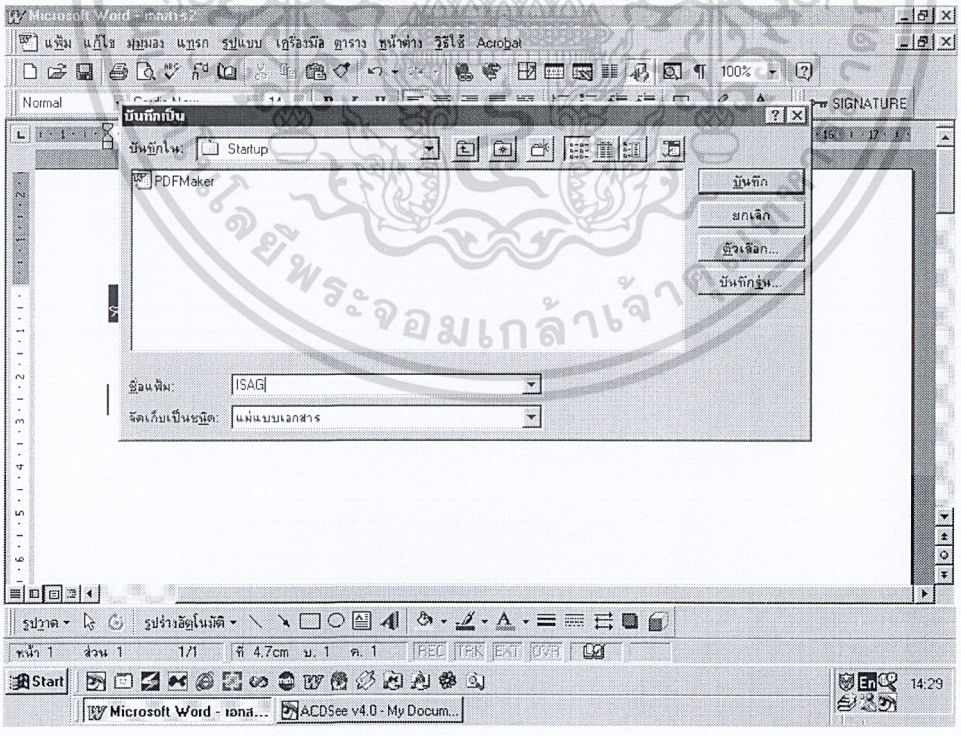
รูปที่ 8.12 กำหนดชื่อให้เครื่องมือใหม่



รูปที่ 8.13 เลือกรูปตามต้องการ



รูปที่ 8.14 แถบเครื่องมือตามต้องการ



รูปที่ 8.15 ทำการบันทึกแม่แบบแมโครใหม่

บทที่ 9

การทดลองและผลการทดลอง

9.1 ความต้องการของระบบ

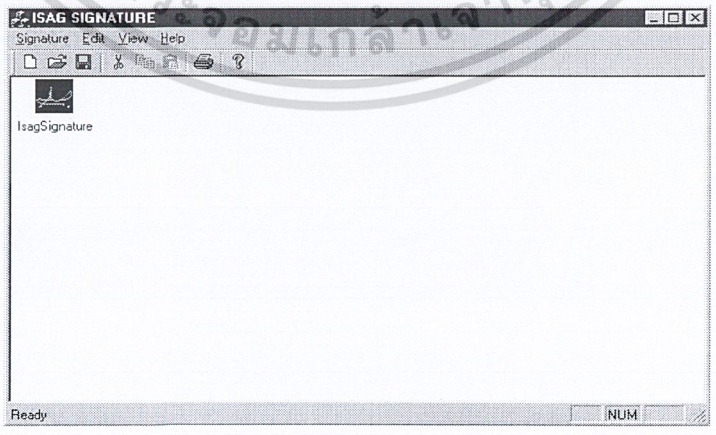
- เครื่องที่ได้ลงระบบปฏิบัติการ Windows 95 OSR 2.0 ขึ้นไป
- ในเครื่องจะต้องรองรับการทำงานของ CryptoAPI 2.0 หรือได้ทำการลง Internet Explorer 4.0 ขึ้นไปเนื่องจากถ้าเป็น Internet Explorer เวอร์ชันที่ต่ำกว่า 4.0 แล้วจะทำให้การสร้างคีย์ไม่สามารถกำหนดรหัสผ่านได้
- จะต้องมีโปรแกรม Microsoft Word 97 อยู่ในเครื่อง

9.2 ระบบที่ใช้ทดสอบ

- ระบบปฏิบัติการ Windows 98
- รองรับการทำงานของ CryptoAPI 2.0
- มีโปรแกรม Microsoft Word 97 ไว้

9.3 การทดสอบการรันโปรแกรมในครั้งแรก

เมื่อมีการรันโปรแกรมในครั้งแรกนั้น โปรแกรมจะทำการสร้างคีย์คอนเทนเนอร์ชื่อ Isag-Signature ซึ่งจะใช้เป็นดีฟอลต์คีย์คอนเทนเนอร์ของโปรแกรม ดังนั้นโปรแกรมครั้งแรกจะสร้างคีย์ส่วนตัว และคีย์สาธารณะเก็บไว้ในคีย์คอนเทนเนอร์ IsagSignature จึงทำให้โปรแกรมทำการคำนวณชั่วขณะหนึ่งซึ่งผู้ใช้จะต้องรอการทำงานในจุดนี้ให้เสร็จ จึงจะเริ่มการทำงานกับโปรแกรมได้ เมื่อการสร้างดีฟอลต์คีย์คอนเทนเนอร์เสร็จเรียบร้อยแล้ว โปรแกรมจะแสดงหน้าต่างดังรูปที่ 9.1

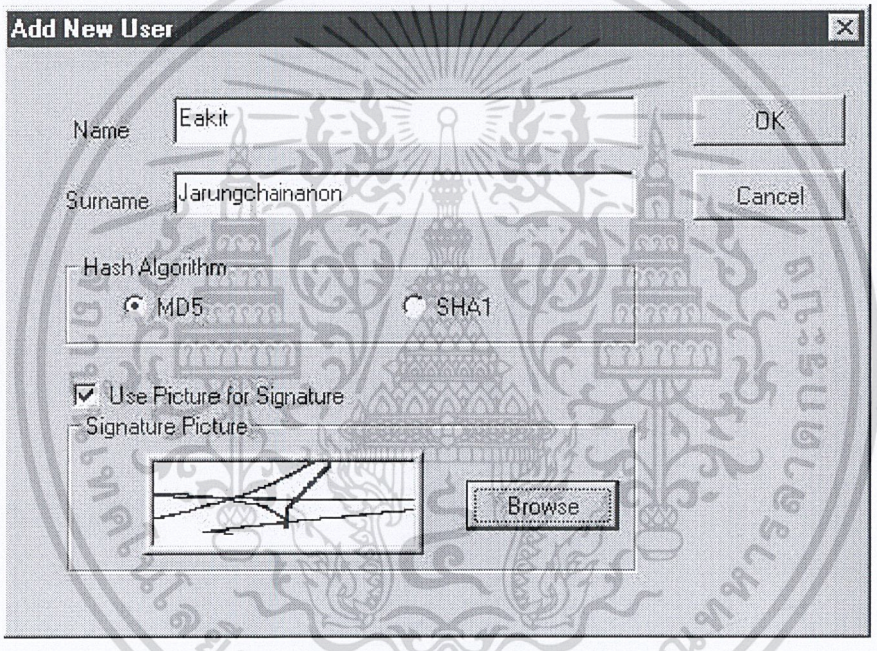


รูปที่ 9.1 แสดงรายชื่อผู้ใช้ที่มีอยู่แล้ว

จากรูปที่ 9.1 จะเห็นว่ามีไอคอนที่ชื่อ IsagSignature ซึ่งเป็นไอคอนหลักของโปรแกรม เพื่อแสดงว่าขณะนี้ได้มีการสร้างคีย์คอนเทนเนอร์ชื่อ IsagSignature ไว้เป็นที่เรียบร้อย ผู้ใช้จะสามารถนำคีย์ดังกล่าวนี้ไปใช้ในการลงลายมือชื่อดิจิตอลได้

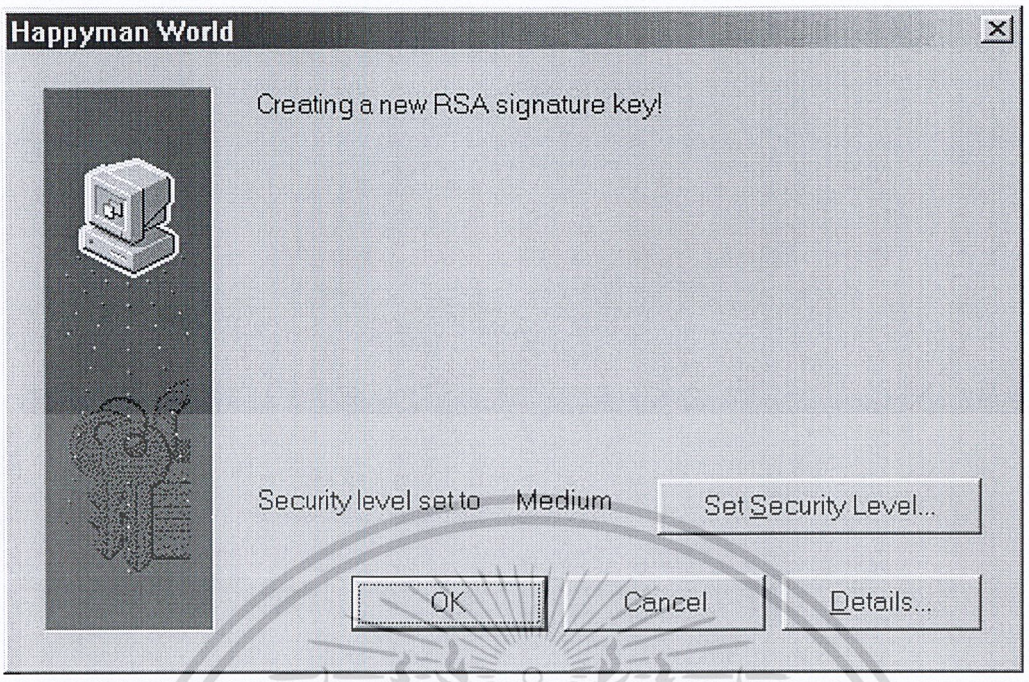
9.4 การทดสอบการสร้างลายมือชื่อดิจิตอลใหม่

- ผู้ใช้จะต้องเลือกเมนู Signature / New Signature
- โปรแกรมจะขึ้นหน้าต่างของการสร้างลายมือชื่อดิจิตอลใหม่ดังรูปที่ 9.2 ให้ผู้ใช้ใส่ชื่อ และนามสกุลของผู้ใช้ลงไป และถ้าผู้ใช้คนใดมีไฟล์รูปภาพที่จะใช้แสดงเมื่อเกิดการลงลายมือชื่อดิจิตอล ก็ให้กดที่ปุ่ม Browse จากนั้นให้เลือกไฟล์รูปภาพ ซึ่งจะมีรูปภาพตัวอย่างแสดงในหน้าต่าง เมื่อผู้ใช้ใส่ข้อมูลครบแล้วให้กดปุ่ม OK

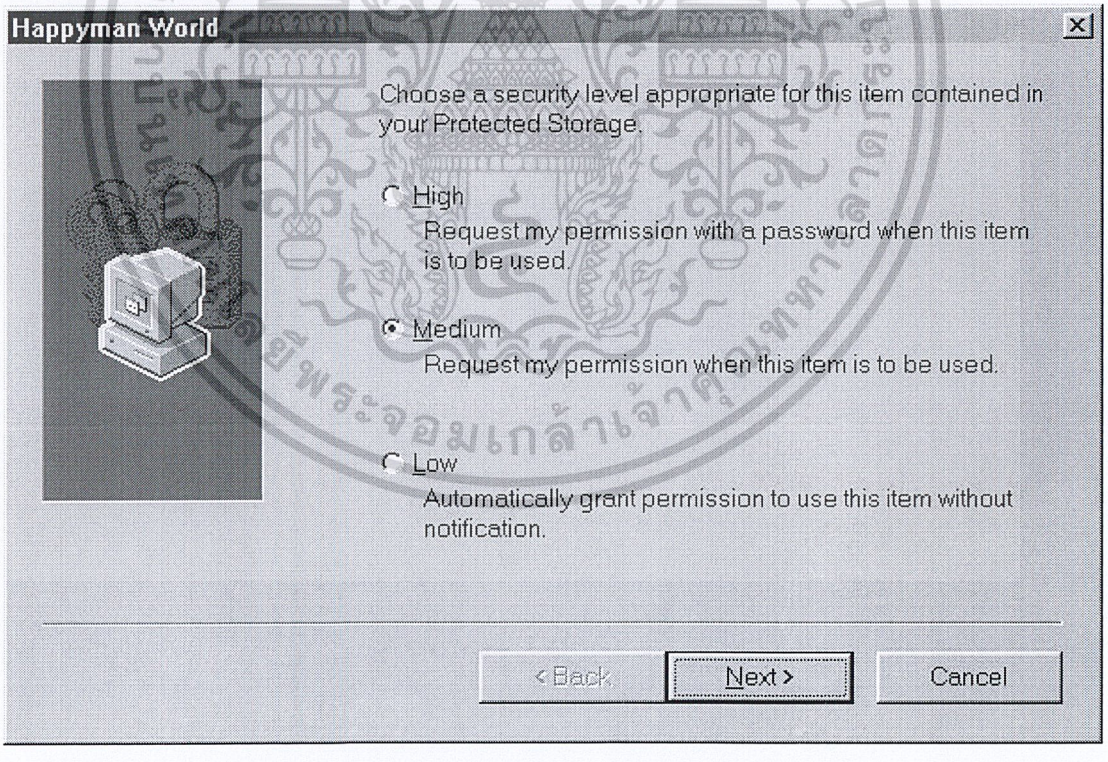


รูปที่ 9.2 หน้าจอเมื่อทำการเพิ่มลายมือชื่อ

- โปรแกรมจะขึ้นหน้าต่างเพื่อถามว่าต้องการความปลอดภัยจากการใช้คีย์ส่วนตัวระดับใด ดังรูปที่ 9.3 โดยการกดที่ปุ่ม Set Security Level จะได้หน้าจอของโปรแกรมดังรูปที่ 9.4 โดยมีให้เลือก 3 ระดับคือ ระดับต่ำ ผู้ใช้จะสามารถใช้คีย์ส่วนตัวได้โดยไม่มีการแจ้งเตือนใด ๆ ทั้งสิ้น, ระดับกลาง ผู้ใช้จะได้รับการแจ้งเตือนเมื่อมีการใช้คีย์ส่วนตัว และสุดท้ายระดับสูง ซึ่งจะต้องมีการใส่รหัสผ่านเพื่อกำหนดค่าไว้สำหรับเป็นความลับ

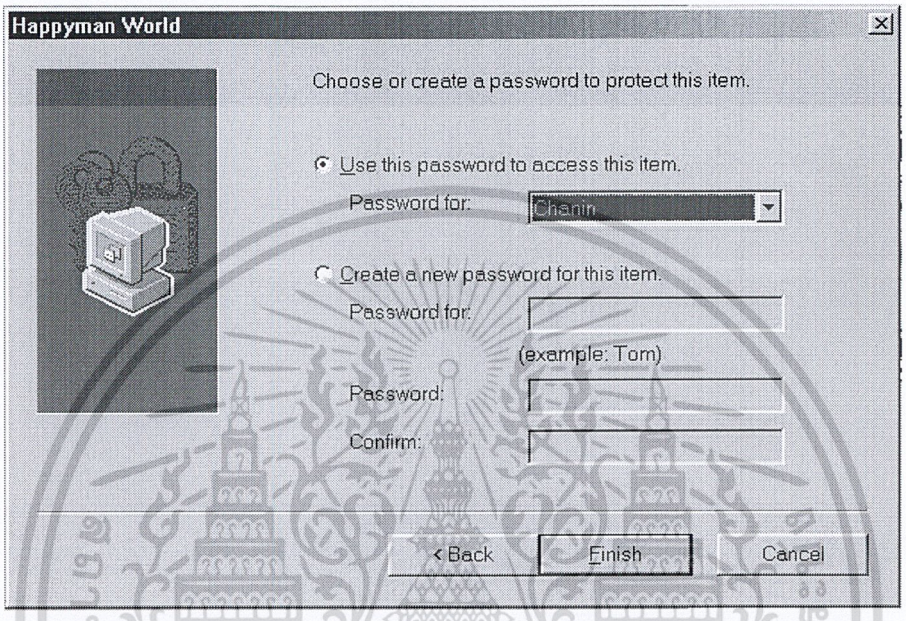


รูปที่ 9.3 หน้าจอเมื่อกดยืนยันการสร้าง

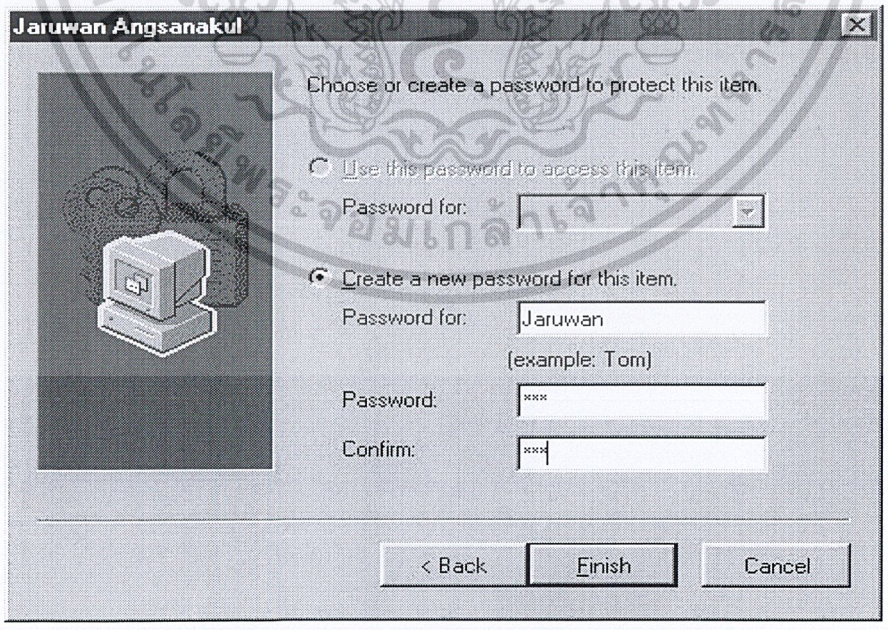


รูปที่ 9.4 หน้าจอเมื่อกดปุ่ม Set Security Level

- ถ้าผู้ใช้เลือกความปลอดภัยระดับต่ำและกลาง จะสามารถจบขั้นตอนของการสร้างลายมือชื่อได้เลย แต่หากผู้ใช้เลือกระดับสูง จะต้องมีการกำหนดรหัสผ่านซึ่งการสร้างรหัสผ่านจะสามารถสร้างขึ้นมาใหม่ หรือจะใช้รหัสผ่านที่เคยสร้างไว้แล้วก็ได้ ซึ่งถ้าต้องการใช้รหัสผ่านที่มีอยู่แล้ว ดังรูปที่ 9.5 โปรแกรมจะให้เลือกชื่อของรหัสผ่านที่เคยสร้างมาแล้ว แต่ถ้าต้องการสร้างใหม่ผู้ใช้จะต้องสร้างชื่อรหัสผ่านขึ้นมาใหม่ และกำหนดรหัสผ่านให้ดังรูปที่ 9.6

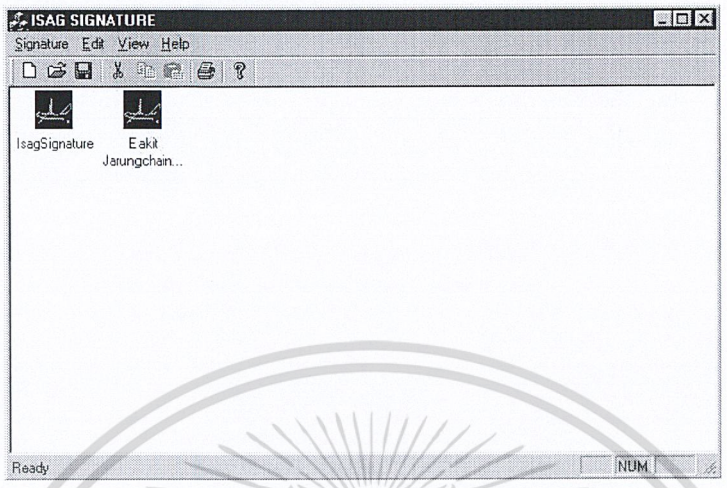


รูปที่ 9.5 หน้าจอเลือกว่ายืนยันใช้รหัสผ่านอันนี้



รูปที่ 9.6 รูปแสดงการสร้างรหัสผ่านอันใหม่

- โปรแกรมจะให้ใส่รหัสผ่านอีกครั้งเผื่อเป็นการสร้างคีย์ส่วนตัว และคีย์สาธารณะ
- จะได้ลายมือชื่อดิจิตอลที่สร้างขึ้นมาตามรูปที่ 9.7



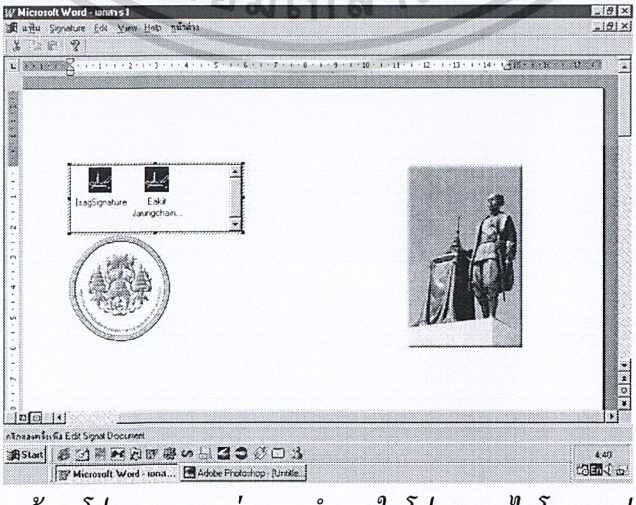
รูปที่ 9.7 ผลของการสร้างลายมือชื่ออันใหม่

9.5 การลบลายมือชื่อดิจิตอลที่ไม่ใช้

- เริ่มจากผู้ใช้จะต้องทำการเลือกแถบสีให้กับไอคอนของลายมือชื่อดิจิตอลที่ต้องการลบทิ้ง
- โปรแกรมจะทำการลบคีย์ส่วนตัว และคีย์สาธารณะของลายมือชื่อดิจิตอลนั้น ถ้าหากในขั้นตอนการสร้างลายมือชื่อดิจิตอล มีการกำหนดรหัสผ่านไว้ ในขั้นตอนนี้จะต้องมีการใส่รหัสผ่านนั้นด้วยเพื่อเป็นการยืนยัน

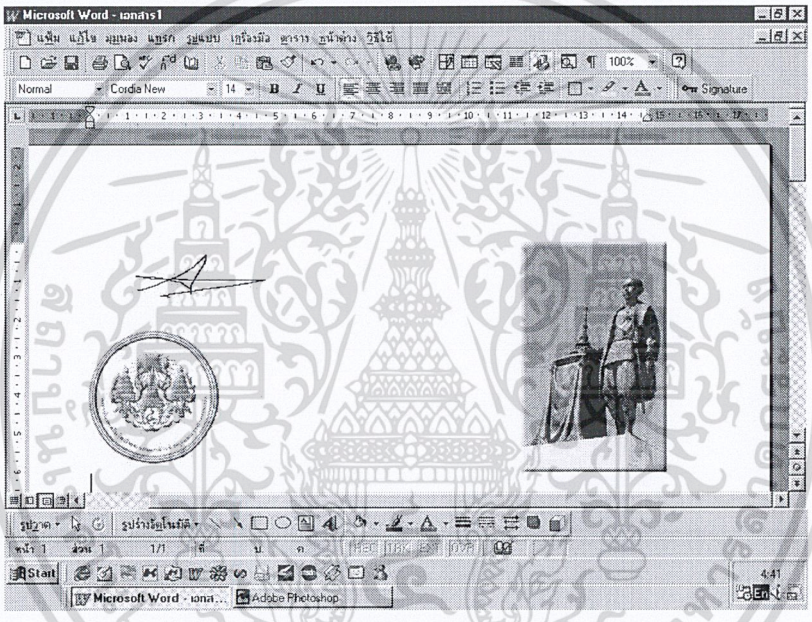
9.6 การลงลายมือชื่อดิจิตอล

- ในโปรแกรมที่ต้องการลงลายมือชื่อดิจิตอล ให้เลือกที่เมนู IsagSign จากนั้นเลือก Sign Signature
- โปรแกรมจะทำการหาว่าในเครื่องนั้น ๆ ได้มีการสร้างลายมือชื่อดิจิตอลใดไว้บ้าง และจะนำมาแสดงในรูปของไอคอนดังแสดงในรูปที่ 9.8



รูปที่ 9.8 หน้าจอโปรแกรมระหว่างการทำงานในโปรแกรมไมโครซอฟท์เวิร์ด

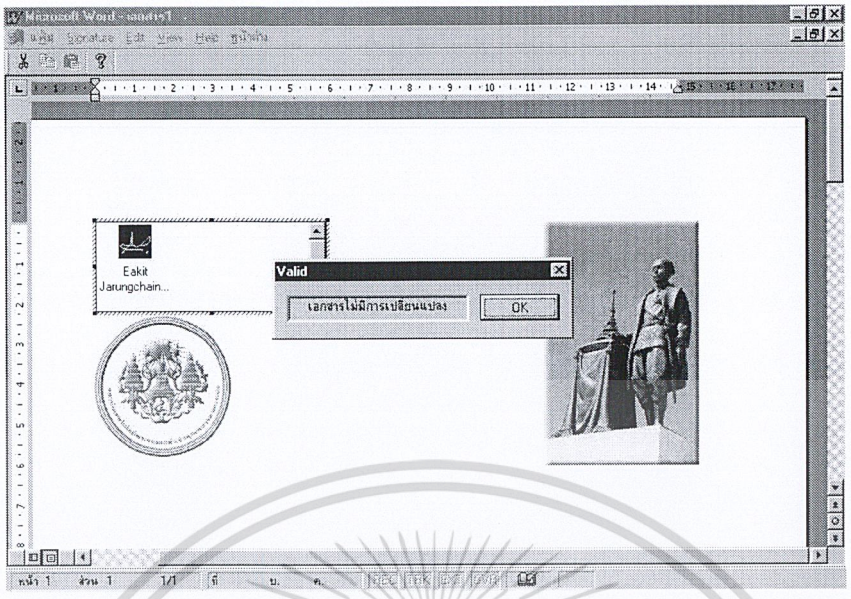
- ผู้ใช้จะต้องทำการเลือกลายมือชื่อดิจิตอลของตนเองเพื่อทำการลงลายมือชื่อดิจิตอล
- โปรแกรมจะนำคีย์ส่วนตัวของผู้ใช้ผู้นั้นออกมาทำการลงลายมือชื่อดิจิตอล ซึ่งในขั้นนี้อาจต้องมีการใส่ รหัสผ่านของการเข้าใช้คีย์ส่วนตัว ขึ้นอยู่กับขั้นตอนของการสร้างลายมือชื่อดิจิตอลนั้น
- โปรแกรมจะตรวจสอบว่าผู้ใช้นั้นมีการใส่รูปลายมือชื่อดิจิตอลไว้หรือไม่ ถ้ามีจะแสดงรูปของลายมือชื่อดิจิตอลนั้นออกมาดังแสดงในรูปที่ 9.9 แต่ถ้าไม่มีรูปจะแสดงไอคอนของโปรแกรมขึ้นมา



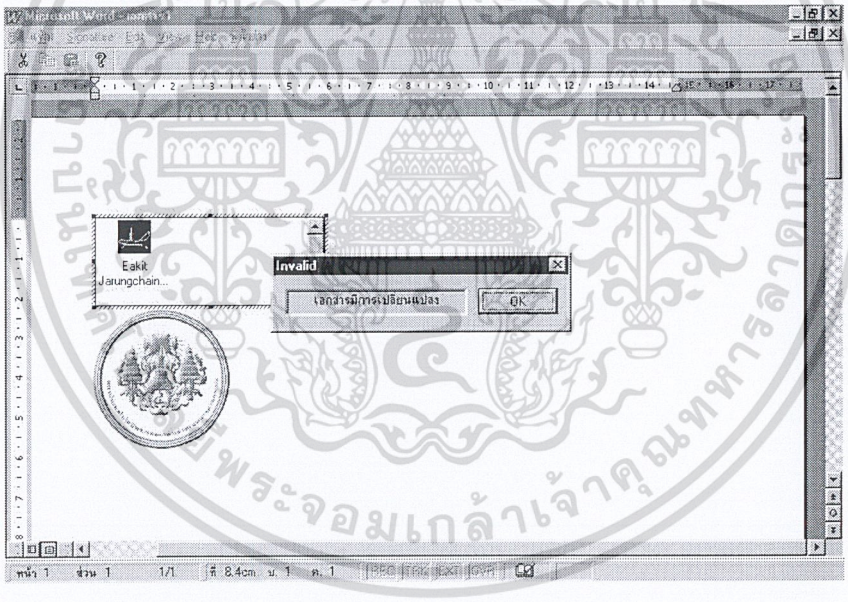
รูปที่ 9.9 ผลที่ได้จากการลงลายมือชื่อ

9.7 การตรวจสอบลายมือชื่อดิจิตอล

- ในเอกสารที่มีการลงลายมือชื่อดิจิตอลไว้แล้ว ให้เลือกที่เมนูของ IsagSign / Verify Signature
- โปรแกรมจะทำการตรวจสอบลายมือชื่อดิจิตอลที่ได้เคยลงไว้ กับข้อมูลปัจจุบันว่าเปลี่ยนแปลงไปหรือไม่ถ้าข้อมูลของเอกสารไม่มีการเปลี่ยนแปลง โปรแกรมจะแสดงข้อความ Signature Valid ดังรูปที่ 9.10



รูปที่ 9.10 แสดงการตรวจสอบลายมือชื่อดิจิตอลว่าถูกต้อง



รูปที่ 9.11 แสดงการตรวจสอบลายมือชื่อดิจิตอลและข้อมูลมีการแก้ไข

บทที่ 10
วิจารณ์และสรุป

10.1 บทวิจารณ์

จากโปรแกรมที่ได้พัฒนามาเพื่อใช้สำหรับการลงลายมือชื่อดิจิตอล ผลที่ได้ออกมาสามารถที่จะลงลายมือชื่อดิจิตอลได้ แต่ตัวโปรแกรมมีข้อจำกัดอยู่ที่การใช้งานได้ในโปรแกรมไมโครซอฟท์เวิร์ดเท่านั้น จากการทดสอบเมื่อลงลายมือชื่อดิจิตอลในเอกสาร หลังจากนั้นตรวจสอบลายมือชื่อดิจิตอล จะได้ผลลัพธ์ว่าการตรวจสอบลายมือชื่อดิจิตอลนั้นถูกต้อง หลังจากได้ทดลองแก้ไขข้อมูลที่อยู่ในเอกสาร จากนั้นตรวจสอบลายมือชื่อดิจิตอล จะได้ว่าการตรวจสอบลายมือชื่อดิจิตอลผิดพลาด ซึ่งเป็นผลลัพธ์ที่ออกมาตรงตามทฤษฎีของลายมือชื่อดิจิตอล

10.2 แนวทางในการพัฒนาโปรแกรม

เนื่องจากโปรแกรมจะสามารถตรวจสอบได้เฉพาะในโปรแกรมไมโครซอฟท์เวิร์ดเท่านั้น ดังนั้นแนวทางในการพัฒนาต่อ จะต้องทำให้โปรแกรมสามารถตรวจสอบข้อมูลได้ในเอกสารที่หลากหลาย โปรแกรมที่จะพัฒนาต่อควรจะมีคุณสมบัติด้านความปลอดภัยสูงกว่าโปรแกรมปัจจุบัน ด้วยการยืนยันบุคคลที่เป็นเจ้าของลายมือชื่อดิจิตอลด้วยเอกสารสิทธิ์ เพราะผู้ที่มีเอกสารสิทธิ์ที่เชื่อถือได้จะต้องได้รับมาจากองค์กรพิสูจน์สิทธิ์

10.3 บทสรุป

การทำลายมือชื่อดิจิตอลนั้นมีความสำคัญกับการยืนยันบุคคล ถ้าเอกสารดิจิตอลใด ๆ ที่มีลายมือชื่อดิจิตอลอยู่ ผู้ที่เป็นเจ้าของลายมือชื่อดิจิตอลจะไม่สามารถปฏิเสธความรับผิดชอบต่อเอกสารนั้น ๆ ได้ และความถูกต้องของข้อมูลก็สามารถยืนยันได้โดยลายมือชื่อดิจิตอลเช่นกัน ว่าจะไม่ถูกเปลี่ยนแปลง หรือแก้ไขหลังจากการลงนามเรียบร้อยแล้ว สามารถนำไปใช้ในการทำธุรกรรมต่าง ๆ ผ่านคอมพิวเตอร์ เพราะเอกสารทางคอมพิวเตอร์นั้นเป็นสิ่งที่ทุกคนสามารถเข้าไปแก้ไขได้ ลายมือชื่อดิจิตอลจึงมีบทบาทสำคัญเป็นอย่างยิ่งต่อการพัฒนาการติดต่อสื่อสาร และการทำธุรกรรมในเครือข่ายคอมพิวเตอร์

บรรณานุกรม

[1] Dorothy E. Denning (1982) : "*Cryptography and Data Security*", Massachusettesa, Addison-Wesley.

[2] Eugene Olafsen ,Kenn Scribner ,K. David White(1999) : "*MFC Programming with Visual C++ 6"*,Sams Publishing

