สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ลายมือชื่อดิจิตอล

Digital Signature

นางสาวจารุวรรณ อังสนากุล

นายชนินทร์ มหารักษ์

วศ.
จ337ล
2543

เลขหมู่.....
เลขทะเบียน 42784
วัน, เดือน, ปี 10 มิ.ย. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายมือชื่อดิจิตอล

Digital Signature

โดย

นางสาวจารุวรรณ อังสนากุล

นายชนินทร์ มหารักษ์

อาจารย์ที่ปรึกษา

อ.ธนา หงษ์สุวรรณ

อ.อัครเดช วัชระภูพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ลายมือชื่อดิจิตอล

Digital Signature

ผู้จัดทำ

1. นางสาว จารุวรรณ อังสนากุล รหัสประจำตัว 40010123
2. นาย ชนินทร์ มหารักษ์ รหัสประจำตัว 40010161

อาจารย์ที่ปรึกษา

(อ.ธนา หงษ์สุวรรณ)

อาจารย์ที่ปรึกษา

(อ.อัครเดช วัชระภูพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายมือชื่อดิจิตอล

นางสาวจารุวรรณ อังสนากุล
นายชนินทร์ มหารักษ์
อาจารย์ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
อาจารย์อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ลายมือชื่อดิจิตอลเป็นลายมือชื่ออิเล็กทรอนิกส์ไม่ใช่ลายมือชื่อที่เขียนโดยคนคนหนึ่ง ซึ่งสามารถพิสูจน์บ่งชี้ผู้ส่งหรือผู้ลงนามเอกสารได้ ทั้งยังทำให้แน่ใจได้ว่าข้อความหรือเอกสารที่ส่งมาไม่ได้มีการเปลี่ยนแปลง และประโยชน์เพิ่มเติมก็คือง่ายในการส่ง ไม่สามารถบอกปัดหรือปฏิเสธได้ ไม่สามารถปลอมได้ และเป็นการประทับเวลาโดยอัตโนมัติ

ลายมือชื่อดิจิตอลจะใช้พื้นฐานของการเข้ารหัสแบบคีย์ต่าง ซึ่งผู้ใช้ทุกคนจะมีคีย์ส่วนตัว และคีย์สาธารณะที่กำหนดขึ้นโดยองค์กรพิสูจน์สิทธิ์ เมื่อผู้ใช้บุคคลนั้นได้มีการลงนามลายมือชื่อดิจิตอล ลำดับของตัวเลขชุดหนึ่งจะถูกสร้างมาจากคีย์ส่วนตัวของบุคคลนั้น ประกอบกับข้อมูลที่ต้องการจะลงนาม ซึ่งลำดับของตัวเลขนี้จะถูกเรียกว่าลายมือชื่อดิจิตอล ลายมือชื่อดิจิตอลสามารถที่จะบ่งบอกได้ว่าผู้ลงนามเป็นผู้ใดโดยดูจากใบรับรองสิทธิ์

โครงงานนี้ได้นำประโยชน์ของลายมือชื่อดิจิตอลมาใช้กับเอกสารที่จัดทำขึ้นจากโปรแกรมไมโครซอฟท์เวิร์ด เพื่อให้โปรแกรมไมโครซอฟท์เวิร์ดมีความสามารถเพิ่มเติมการลงลายมือชื่อดิจิตอลและตรวจสอบลายมือชื่อได้ โดยโครงงานนี้เป็นโปรแกรมที่เขียนด้วยวิชวลซีพลัสพลัส (Visual C++) ซึ่งสามารถทำงานร่วมกับโปรแกรมไมโครซอฟท์เวิร์ดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Signature

Jaruwan Angsanakul
Chanin Maharuk
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.

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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

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

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

นางสาวจารุวรรณ อังสนากุล

นายชนินทร์ มหารักษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อภาษาไทย		I
บทคัดย่อภาษาอังกฤษ		II
กิตติกรรมประกาศ		III
สารบัญ		IV
สารบัญรูปภาพ		VII
สารบัญตาราง		VIII
บทที่ 1	บทนำ	1
	1.1 ความสำคัญและที่มา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของงานวิจัย	1
	1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
	1.5 วิธีการดำเนินงาน	2
บทที่ 2	ลายมือชื่อดิจิตอล	3
บทที่ 3	การเข้ารหัส	8
	3.1 พื้นฐานการเข้ารหัสและถอดรหัส (Basic Encryption and Decryption)	8
	3.1.1 ศัพท์เทคนิคและพื้นฐานขั้นต้น (Terminalogy and Background)	8
	3.1.2 อัลกอริทึมการเข้ารหัสและถอดรหัส (Encryption Algorithms & Decryption Algorithms)	8
บทที่ 4	การเข้ารหัสลับแบบคีย์เหมือน	11
	4.1 DES (Data Encryption Standard)	12
	4.1.1 ประวัติและที่มาของ DES	12
	4.1.2 รายละเอียดของ DES	12
	4.1.3 อัลกอริทึมของการเข้ารหัสแบบ DES	12
	4.1.4 โหมดต่างๆ ของการใช้ DES	14
	4.2 Double DES	15
	4.3 3DES(Triple Data Encryption Standard)	16
บทที่ 5	การเข้ารหัสลับแบบคีย์ต่าง	18
	5.1. อาร์เอสเอ (RSA)	18
	5.2 การคำนวณการเข้า-ถอดรหัสของ RSA	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

บทที่ 6	แฮชฟังก์ชัน	22
	6.1 วิธีแฮช	22
	6.2 คำบรรยายวิธีแฮช	22
	6.3 วิธีแฮชที่ปลอดภัย (Secure Hash Algorithm)	23
บทที่ 7	เอกสารสิทธิ์ (Certificate)	26
	7.1 ความสำคัญของเอกสารสิทธิ์ดิจิตอล (Digital Certificate)	26
	7.2 การขอเอกสารสิทธิ์	27
บทที่ 8	คริปโตเอพีไอ (CryptoAPI)	29
	8.1 การเรียกใช้คริปโตเอพีไอ	29
	8.2 การติดต่อกับ ซีเอสพี	29
	8.3 คำศัพท์ต่าง ๆ ที่ควรรู้	29
	8.4 การเรียกใช้คริปโตเอพีไอ	32
	8.5 การติดต่อกับซีเอสพี	34
	8.6 การใช้คริปโตเอพีไอในการทำงานกับเอกสารสิทธิ์	34
บทที่ 9	โอแอลอี	36
	9.1 โอแอลอี	36
	9.2 โอแอลอีออโตเมชั่น (OLE Automation)	36
	9.2.1 การสร้างโปรแกรมให้ทำงานกับโอแอลอีออโตเมชั่นโดยใช้เอ็มเอฟซี (MFC) และชนิดของไลบรารี (Type Library)	36
	9.3 โอแอลอีเซิร์ฟเวอร์	37
	9.3.1 ขั้นตอนการสร้างโปรแกรมแบบโอแอลอีเซิร์ฟเวอร์	37
	9.3.2 คลาสต่าง ๆ ที่เกิดจากการสร้างโปรแกรมแบบโอแอลอีเซิร์ฟเวอร์	37
บทที่ 10	การออกแบบโปรแกรม	38
	10.1 คุณลักษณะของโปรแกรม	38
	10.2 การออกแบบโปรแกรมในส่วนติดต่อกับผู้ใช้	38
	10.3 การออกแบบโปรแกรมในส่วนของการสร้างผู้ใช้ใหม่	38
	10.4 การออกแบบโปรแกรมในส่วนของการลบชื่อผู้ใช้ออก	39
	10.5 การออกแบบโปรแกรมในส่วนของการทำงานในไมโครซอฟท์เวิร์ด	39
บทที่ 11	การทดลองและผลการทดลอง	44
	11.1 ความต้องการของระบบ	44
	11.2 ระบบที่ใช้ทดสอบ	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

11.3 การทดสอบการรันโปรแกรมในครั้งแรก 44
11.4 การทดสอบการสร้างลายมือชื่อดิจิตอลใหม่ 45
11.5 การลบลายมือชื่อดิจิตอลที่ไม่ใช้ 49
11.6 การลงลายมือชื่อดิจิตอล 49
11.7 การตรวจสอบลายมือชื่อดิจิตอล 50
บทที่ 12 วิจารณ์และสรุป 52
12.1 บทวิจารณ์ 52
12.2 แนวทางในการพัฒนาโปรแกรม 52
12.3 บทสรุป 52
ภาคผนวก ก. 53
บรรณานุกรม 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 แสดงบล็อกการเข้ารหัส ถอดรหัส 9
รูปที่ 3.2 แสดงบล็อกการเข้ารหัสถอดรหัสโดยใช้วิธีคีย์เหมือน 10
รูปที่ 3.3 แสดงบล็อกการเข้ารหัสถอดรหัสโดยใช้วิธีคีย์ต่าง 10
รูปที่ 4.1 แสดงการแบ่งคีย์ 56 บิตที่เรียงลำดับบิตแล้วออกเป็น 2 ส่วน 13
รูปที่ 4.2 แสดงบล็อกข้อมูล 64 บิต ที่แบ่งออกเป็น 2 ส่วน หลังจากทำการจัดเรียงบิตแล้ว 14
รูปที่ 4.3 แสดงการเข้ารหัสและถอดรหัสในแบบ Double DES 15
รูปที่ 6.1 วิธีแฮชที่ปลอดภัย (Secure Hash Algorithm) 24
รูปที่ 10.1 ส่วนประกอบต่าง ๆ ของโปรแกรม จะประกอบไปด้วยคลาสหลาย ๆ คลาส 41
รูปที่ 11.1 แสดงลายมือชื่อที่มีอยู่แล้ว 44
รูปที่ 11.2 หน้าจอเมื่อทำการเพิ่มลายมือชื่อ 45
รูปที่ 11.3 หน้าจอเมื่อกดยืนยันการสร้าง 46
รูปที่ 11.4 หน้าจอเมื่อกดปุ่ม Set Security Level 46
รูปที่ 11.5 หน้าจอเลือกว่ายืนยันใช้รหัสผ่านอันนี้ 47
รูปที่ 11.6 หน้าจอเมื่อต้องการสร้างรหัสผ่านอันใหม่ 48
รูปที่ 11.7 ผลของการสร้างลายมือชื่ออันใหม่ 48
รูปที่ 11.8 หน้าจอโปรแกรมระหว่างการทำงานในโปรแกรมไมโครซอฟท์เวิร์ด 49
รูปที่ 11.9 ผลที่ได้จากการลงลายมือชื่อ 50
รูปที่ 11.10 แสดงการตรวจสอบลายมือชื่อดิจิตอลว่าถูกต้อง 51
รูปที่ 11.11 แสดงการตรวจสอบลายมือชื่อดิจิตอลและข้อมูลมีการแก้ไข 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 5.1 เปรียบเทียบข้อดีและข้อเสียของการเข้ารหัส 20
ตารางที่ 5.2 ประมาณเวลาและค่าใช้จ่ายสำหรับการหาคีย์ที่ใช้ถอดรหัสที่มีความยาวต่างๆกัน 20
ตารางที่ 8.1 ชื่อที่ใช้บ่งบอกว่าเป็น ซีเอสพี ตัวใด 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันการส่งข้อมูลทางอิเล็กทรอนิกส์ เช่น การส่งจดหมายทางอิเล็คทรอนิกส์ การทำธุรกรรมบนอินเทอร์เน็ตหรือการส่งข้อมูลที่เป็นความลับบนระบบเครือข่าย ผู้รับข้อมูลจำเป็นต้องตรวจสอบที่มาของข้อมูลว่าถูกต้องหรือไม่ ดังนั้นผู้ส่งข้อมูลจึงต้องมีการรับรองข้อมูลดังกล่าว ลายมือชื่อดิจิตอล (Digital Signature) จึงถือกำเนิดขึ้นโดยใช้ในการแจ้งสิทธิ์ของบุคคลหรือสิ่งของ ซึ่งแตกต่างจากลายมือชื่อที่เขียนด้วยมือทั่วไป เพราะลายมือชื่อดิจิตอลไม่คงที่ขึ้นอยู่กับข้อมูล ลายมือชื่อดิจิตอลสร้างขึ้นจากข้อมูลทั้งหมดโดยใช้การเข้ารหัสทางเดียว ดังนั้นหากมีการเปลี่ยนแปลงข้อมูลในเอกสารแม้เพียงบิตเดียวก็จะทำให้ลายมือชื่อดิจิตอลนั้นเปลี่ยนไปด้วย ทำให้สามารถนำไปใช้ในการพิสูน์ที่มาและความถูกต้องของข้อมูลทางอิเล็กทรอนิกส์ได้

ลายมือชื่อดิจิตอลเป็นลายมือชื่ออิเล็กทรอนิกส์ไม่ใช่ลายมือที่เขียนโดยคนคนหนึ่ง ซึ่งสามารถพิสูจน์บ่งชี้ผู้ส่งหรือผู้ลงนามเอกสารได้ ทั้งยังทำให้แน่ใจได้ว่าข้อความหรือเอกสารที่ส่งมาไม่ได้มีการเปลี่ยนแปลง และประโยชน์เพิ่มเติมก็คือมันง่ายในการส่ง ไม่สามารถบอกปัดหรือปฏิเสธได้ ไม่สามารถปลอมได้ และเป็นการประทับเวลาโดยอัตโนมัติ

หลักการทำงานของลายมือชื่อดิจิตอลเริ่มจากการนำข้อมูลที่ต้องการส่งมาสังเคราะห์ข้อมูลขึ้น (Message digest-MD) โดยใช้ฟังก์ชันทางเดียว เมื่อได้ข้อมูลมาแล้วจะทำการเข้ารหัสโดยใช้คีย์ส่วนตัวของผู้ส่ง ข้อมูลที่ผ่านการเข้ารหัสดังกล่าวนี้เรียกว่าลายมือชื่อดิจิตอล โดยลายมือชื่อดิจิตอลของข้อมูลแต่ละชุดนั้นไม่คงที่ ผู้รับเมื่อรับข้อมูลชุดดังกล่าวจากผู้ส่งมาแล้ว ใช้ฟังก์ชันทางเดียวแบบเดียวกันเพื่อสังเคราะห์ข้อมูลชุดหนึ่งขึ้น จากนั้นจะใช้คีย์สาธารณะของผู้ส่งเพื่อถอดรหัสลายมือชื่อดิจิตอลที่ส่งมา ได้ข้อมูลสังเคราะห์อีกหนึ่งชุด จากนั้นเปรียบเทียบกับข้อมูลสังเคราะห์ดังกล่าว ถ้าตรงกันแสดงว่าข้อมูลนั้นไม่มีการปลอมแปลงแก้ไขและทำให้ทราบว่าเป็นบุคคลที่ส่งข้อมูลนั้นจริงๆ แต่ถ้าไม่ตรงกันก็แสดงว่าข้อมูลที่รับมานั้นไม่ถูกต้องอาจถูกปลอมแปลงแก้ไขทำให้สามารถตรวจสอบบุคคลอื่นที่ปลอมตัวเข้ามาได้

1.2 วัตถุประสงค์

เพื่อศึกษาถึงการทำลายมือชื่อดิจิตอล และสามารถนำความรู้ที่ศึกษามาได้ นำมาพัฒนาโปรแกรมที่สามารถลงนามลายมือชื่อดิจิตอลได้ เพื่อความถูกต้องของข้อมูลคือ ข้อมูลจะต้องไม่มีการเปลี่ยนแปลงระหว่างการส่ง และทำให้สามารถยืนยันได้ว่าบุคคลที่ส่งข้อมูลนั้นเป็นบุคคลคนนั้นจริง ๆ

1.3 ขอบเขตของโครงงาน

- ศึกษาเรื่องการเข้ารหัสทั้งแบบคีย์เหมือน และแบบคีย์ต่าง
- ศึกษาการทำแฮชชิงกับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษาวิธีการทำลายมือชื่อดิจิตอล
- ศึกษาวิธีการตรวจสอบลายมือชื่อดิจิตอล
- ศึกษาการเขียนโปรแกรมเพื่อนำไปพัฒนาโปรแกรมสร้างลายมือชื่อดิจิตอล

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- ความเข้าใจในเรื่องการเข้ารหัสทั้งแบบคีย์เหมือนและคีย์ต่าง
- สามารถเขียนโปรแกรมที่สร้างลายมือชื่อดิจิตอลได้
- สามารถเขียนโปรแกรมที่ใช้ในการตรวจสอบลายมือชื่อดิจิตอลได้

1.5 วิธีการดำเนินงาน

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัย อันได้แก่ การเข้ารหัสด้วยวิธีต่างๆ และการทำแฮชฟังก์ชัน ซึ่งมีรายละเอียดดังในบทที่ 3, 4, 5 และ 6 จากนั้นก็เป็นการศึกษาเกี่ยวกับการเขียนโปรแกรม โดยทำการศึกษาภาษาวิชวลซีพลัสพลัส (Visual C++) การศึกษาคริปโตเอพีไอ (CryptoAPI) และการศึกษาโอแอลอี ดังรายละเอียดในบทที่ 7, 8 และ 9 จากนั้นนำเอาความรู้ที่ได้ศึกษาทั้งหมดมาออกแบบและเขียนโปรแกรมขึ้นมา ซึ่งมีรายละเอียดในบทที่ 10 โดยกล่าวถึงองค์ประกอบโดยรวมของโปรแกรมทั้งหมด และยังอธิบายไปถึงรูปแบบการติดต่อกับผู้ใช้

สำหรับบทที่ 11 ก็จะเป็นการทดสอบระบบรวมทั้งหมด และบทที่ 12 ซึ่งเป็นบทสุดท้ายก็จะเป็นการสรุปการทำงาน ผลที่ได้รับจากงานวิจัยชิ้นนี้ และแนวทางในการพัฒนางานวิจัยนี้เพิ่มเติม และแนวทางในการนำไปประยุกต์ใช้

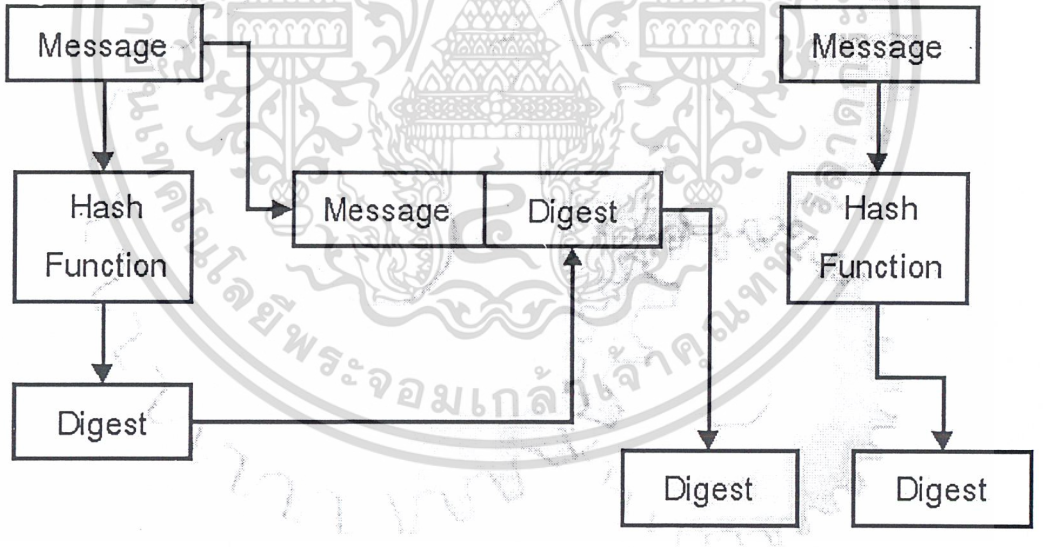
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ลายมือชื่อดิจิตอล

ลายมือชื่อดิจิตอลใช้เมื่อต้องการความมั่นใจในแหล่งที่มาของเอกสารเปรียบเหมือนลายมือชื่อ ซึ่งเฉพาะเจ้าของจริงที่สามารถคำนวณขึ้นมาได้ แต่ลายมือชื่อนี้สามารถพิสูจน์ได้คือบุคคลอื่นสามารถตรวจสอบได้ว่าลายมือชื่อนั้นมาจากผู้สร้างจริงๆ วิธีธรรมดาทั่วไปที่จะคำนวณลายมือชื่อดิจิตอลก็คือการเข้ารหัสแบบคีย์สาธารณะ เช่น ผู้ลงนามคำนวณค่าลายมือชื่อโดยใช้คีย์ส่วนตัว (Private key) และคนอื่นสามารถใช้คีย์สาธารณะพิสูจน์ได้ว่าลายมือชื่อมาจากคีย์ส่วนตัวที่ตรงกัน คุณสมบัติที่สำคัญของลายมือชื่อดิจิตอลที่สำคัญนั้นจะต้องประกอบด้วย 2 ประการคือ

- สามารถยืนยันได้ว่าข้อมูลที่ได้รับมานั้นไม่มีการเปลี่ยนแปลงระหว่างการส่ง
- สามารถยืนยันได้ว่าข้อมูลนั้นได้รับการยืนยันจากผู้ลงลายมือชื่อจริง ๆ

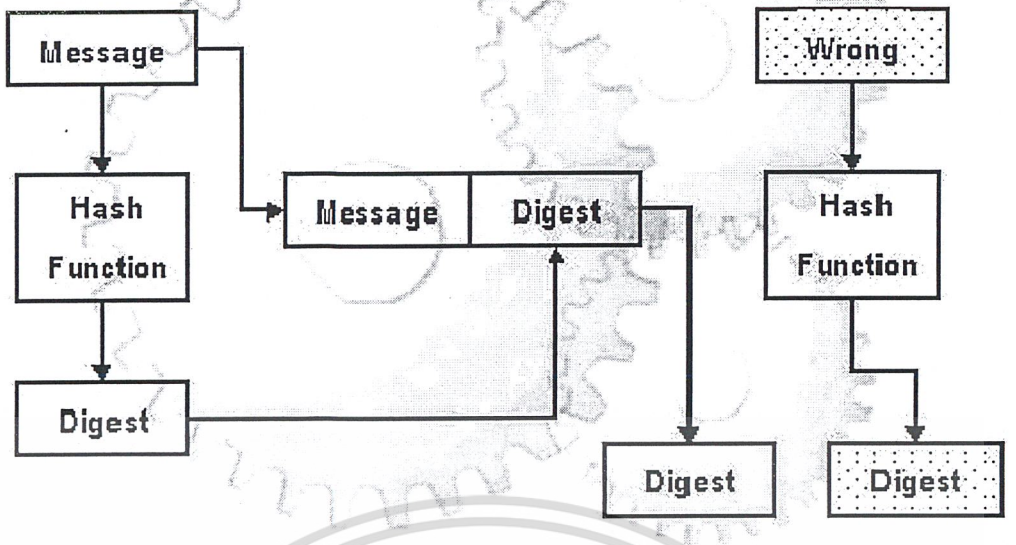
วิธีการยืนยันว่าข้อมูลที่ได้รับมานั้นมีความถูกต้อง และไม่ได้รับการเปลี่ยนแปลงข้อมูลระหว่างการส่งนั้นสามารถทำได้โดยวิธีง่าย ๆ คือ การหาค่าแฮช (Hash Value) ดังรูปที่ 2.1



รูปที่ 2.1 วิธีการตรวจสอบความถูกต้องของข้อมูลแบบพื้นฐาน

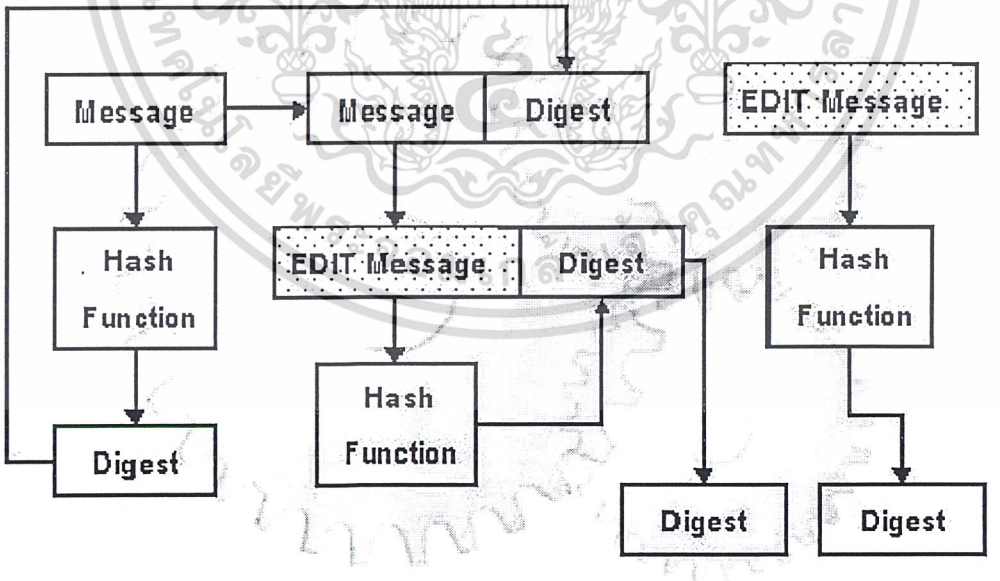
จากรูปจะเป็นวิธีการที่ใช้ในการยืนยันความถูกต้องของข้อมูลอย่างง่าย โดยการหาค่าแฮชหรือจะเรียกว่าไดเจสต์ (Digest หรือ Message Digest) ก็ได้ซึ่งไดเจสต์ จะเปรียบเสมือนตัวแทนของข้อมูล ถ้าข้อมูลที่ได้รับเข้าสู่แฮชฟังก์ชันนั้นต่างกันค่าของไดเจสต์ที่ได้รับออกมานั้นก็จะต่างกัน แต่หากว่าข้อมูลที่ได้รับเข้าสู่แฮชฟังก์ชันนั้นเหมือนกันก็จะทำให้ค่าของไดเจสต์ที่ได้ออกมาเหมือนกัน มีโอกาสน้อยมากที่จะทำให้ข้อมูลเปลี่ยนไปแล้วยังได้ไดเจสต์เหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 เมื่อข้อมูลเกิดการผิดพลาด

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

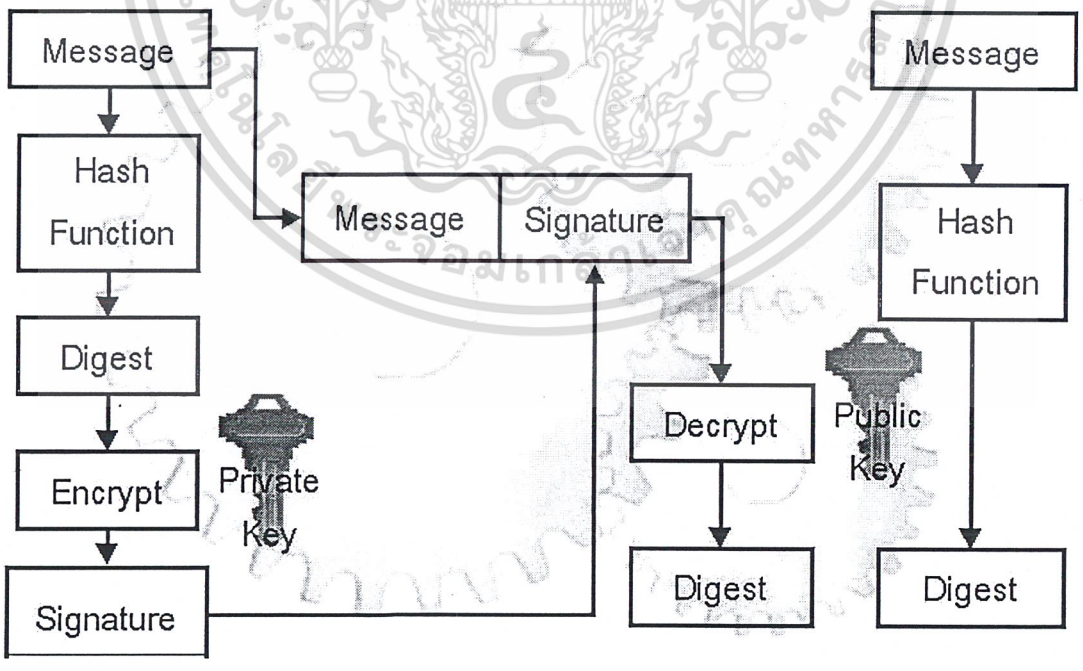


รูปที่ 2.3 เมื่อมีผู้อ้างตัวเป็นผู้ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 เมื่อมีผู้อ้างตัวเป็นผู้ส่งนำข้อมูลที่ผู้ส่งส่งมาดัดแปลงแก้ไข แล้วใช้แฮชฟังก์ชันซึ่งเป็นอันเดียวกับผู้ส่งและผู้รับใช้จะทำให้ด้านผู้รับไม่อาจรู้ได้ว่าข้อมูลนั้นมีการเปลี่ยนแปลง ทำให้ไม่ปลอดภัยดังนั้นจึงได้มีการใช้การเข้ารหัสเข้ามาช่วย

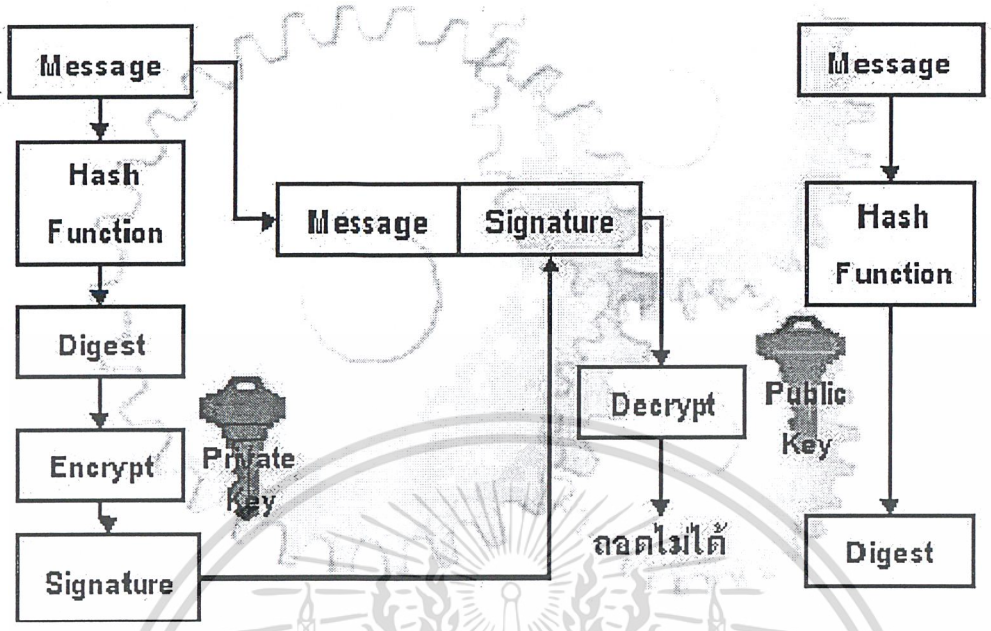
ในการยืนยันบุคคลผู้ส่งนั้นเป็นสิ่งที่จะต้องทำตามมาหลังจากการยืนยันความถูกต้องของข้อมูล เทคนิคที่นิยมใช้ในการทำก็คือ ใช้เทคนิคคีย์ต่าง เช่น อาร์เอสเอ (RSA) โดยข้อความแรกเริ่มจะผ่านแฮชฟังก์ชันแล้วได้เป็นแมสเซจไดเจสต์ เพื่อเป็นการยืนยันความถูกต้องของข้อมูลก่อน จากนั้นจะนำไดเจสต์ที่ได้ส่งไปเข้ารหัสกับคีย์ส่วนตัวของผู้ส่งเพื่อเป็นการยืนยันว่าเป็นเจ้าของเอกสารนั้นจริง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



รูปที่ 2.4วิธีการทำลายมือชื่อดิจิตอลโดยใช้แฮชฟังก์ชันและคีย์ต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 จะเห็นว่ามีความปลอดภัยมากยิ่งขึ้นในกรณีที่มีการปลอมแปลงจะเป็นดังรูปที่ 2.5



รูปที่ 2.5 ในกรณีมีผู้แอบอ้าง

จากรูปที่ 2.5 เมื่อมีผู้แอบอ้าง ผู้แอบอ้างไม่รู้คีย์ส่วนตัวของผู้ส่งจริงๆ จึงใช้คีย์ของตัวเอง ซึ่งไม่ได้เข้าคู่กับคีย์สาธารณะทางฝั่งผู้รับ ทำให้ทางฝั่งผู้รับไม่สามารถถอดรหัสลายมือชื่อดิจิตอลได้ จึงรู้ว่าไม่ใช่ผู้ส่งตัวจริงเป็นคนส่ง

ตัวอย่างของการนำไปใช้ดังเช่น

สมมติว่าคุณต้องการส่งข้อความไปให้ทนายของคุณ โดยประกันว่าจะไม่มีการเปลี่ยนแปลงข้อความ และข้อความที่ส่งมาจากคุณจริงๆ

<u>*ทางด้านของคุณจะส่งข้อมูลผ่านทางอีเมล์*</u>

1. สำเนาข้อความและพิมพ์ลงอีเมล์
2. ใช้โปรแกรมเฉพาะ คุณจะได้ข้อความที่ผ่านการแฮชชิ่ง (Hashing) หรือเรียกว่าแมสเซจไดเจสต์
3. จากนั้นใช้คีย์ส่วนตัวของคุณเองในการเข้ารหัสแมสเซจไดเจสต์ที่ได้จากการทำในขั้นตอนที่แล้ว ซึ่งจะได้ลายมือชื่อดิจิตอลของคุณ
4. ทำการส่งอีเมล์ซึ่งประกอบไปด้วยข้อความในอีเมล์ และลายมือชื่อดิจิตอลของคุณไปให้ทนาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางฝั่งทนายก็จะรับข้อความของคุณ

1. ทนายก็จะทำแมสเซจไดเจสต์จากข้อความที่ได้รับ โดยการผ่านแฮชฟังก์ชันวิธีเดียวกับที่คุณได้ทำมา ซึ่งจะได้เป็นแมสเซจไดเจสต์ทางฝั่งรับ
2. ทนายของคุณก็จะใช้คีย์สาธารณะของคุณถอดรหัสลายมือชื่อดิจิตอลที่อยู่ในอีเมล์ ผลที่ได้รับจะเป็นแมสเซจไดเจสต์ของทางฝั่งที่ส่งมา
3. ถ้าแมสเซจไดเจสต์ทางฝั่งรับ และทางฝั่งส่งเหมือนกันก็แสดงว่าข้อความไม่มีการเปลี่ยนแปลง และข้อความที่ได้รับมานั้นส่งมาจากของคุณจริง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การเข้ารหัส

3.1 พื้นฐานการเข้ารหัสและถอดรหัส (Basic Encryption and Decryption)

ในบางครั้งในการติดต่อส่งข่าวสาร ข่าวสารของเราอาจจะไปไม่ถึงมือผู้รับ โดยอาจมีผู้อื่นมาเอาข่าวสารข้อมูลของเราไป ดังนั้นเราอาจจะใช้การเข้ารหัส (Encryption) เพื่อปลอมแปลงข่าวสารของเราขณะส่งไป การเข้ารหัสเป็นการรักษาความปลอดภัยของข้อมูลที่อยู่ในสภาพแวดล้อมที่ไม่ปลอดภัย

ซึ่งการเข้ารหัสมีพื้นฐานด้วยกันอยู่ 2 อย่างคือการแทนที่ (Substitute) และการสับเปลี่ยนตำแหน่ง

1. การแทนที่ (Substitution) เป็นการแทนที่บิตนั้นๆ ด้วยข้อมูลของบิตอื่น ซึ่งจะทำให้ข้อมูลมีความสับสนมากขึ้น ยากแก่การถอดรหัส
2. การสับเปลี่ยนตำแหน่ง (Permutation) เป็นการสับเปลี่ยนตำแหน่งของบิตต่างๆ ทำให้ข้อมูลมีตำแหน่งที่ผิดเพี้ยนไป ถ้ามีการใช้การสับเปลี่ยนตำแหน่งมาก จะทำให้ข้อมูลยิ่งยากแก่การถอดรหัสเช่นกัน

3.1.1 ศัพท์เทคนิคและพื้นฐานขั้นต้น (Terminalogy and Background)

ถ้านาย ก ต้องการส่งข่าวสารให้ นาย ข เรียกนาย ก ว่าผู้ส่งและเรียกนาย ข ว่าผู้รับ โดยนาย ก จะส่งข่าวสารโดยใช้ T เป็นตัวกลางส่งไปให้ผู้รับ นาย ก เรียก T ว่าตัวกลางการสื่อสาร (Transmission mediam) แต่ถ้ามี นาย ค ซึ่งต้องการข่าวสารนี้เช่นกัน และมาดึงข่าวสารนี้ไป เราเรียกนาย ค ว่าผู้กีดกัน (intercept หรือ intruder) เพราะในการส่งข่าวสารของนาย ก ผ่านตัวกลาง T นั้น เป็นข่าวสารที่เปิดเผย ดังนั้นนาย ค อาจจะพยายามที่จะนำข่าวสารนี้ไปใช้โดยวิธี

- interrupt โดยจะป้องกันไม่ให้นาย ข ติดต่อข่าวสารนี้ไปใช้ประโยชน์
- intercept หาช่องทางในการติดต่อ และทำการปิดบังข่าวสารนั้น
- modify เปลี่ยนแปลงข้อมูลและดึงข้อมูลเข้ามา
- fabricate ทำการแปลงข้อมูลให้เหมือนกับข้อมูลที่ส่งมาจากนาย ก

จากส่วนนี้จะเป็นปัญหาในการติดต่อส่งข้อมูลทำให้ไม่สำเร็จ การเข้ารหัสเป็นเทคนิคในการแก้ปัญหา การเข้ารหัสซึ่งทำให้ผู้อื่นไม่สามารถใช้ได้ การถอดรหัส (Decryption) เพื่อที่จะให้สามารถใช้งานข้อมูลนี้ได้ จะเรียกระบบนี้ว่าคริปโตซิสเต็ม (Cryptosystem)

3.1.2 อัลกอริทึมการเข้ารหัสและถอดรหัส (Encryption Algorithms & Decryption Algorithm)

การเข้ารหัสนั้นถือเป็นพื้นฐานสำคัญในความปลอดภัยของข้อมูล ในระบบการเข้ารหัสนั้นข้อมูลธรรมดาที่เราสามารถอ่านได้เรียกว่า “เพลนเท็กซ์” (Plain text) จะถูกแปลงโดยใช้อัลกอริทึมทางคณิตศาสตร์ ให้เป็นข้อมูลที่เราไม่สามารถอ่านเข้าใจได้เรียกว่า “ไซเฟอร์เท็กซ์” (Cipher text) ซึ่งถ้าได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไซเฟอร์เท็กซ์มาแล้ว ก็จะต้องทำการถอดรหัสให้กลายเป็น เพลนเท็กซ์ ดังแสดงไว้ในรูปที่ 3.1 ซึ่งสามารถอธิบายวิธีการทำได้ดังนี้

ข้อมูล plaintext = P จะแสดงอยู่ในรูปอนุกรมดังนี้

P = [P1, P2, …, Pn] และเมื่อเข้ารหัสแล้วจะเปลี่ยนเป็น C = [C1, C2, …, Cn]

เขียนให้อยู่ในอีกรูปแบบ C = E(P) และ P = D(C)

C = Cipher text

P = Plain text

E = Encryption algorithms

D = Decryption algorithms

แต่ในระบบความปลอดภัยจะได้สมการ P = D(E(P))

รูปที่ 3.1 แสดงบล็อกการเข้ารหัส ถอดรหัส

ระบบการเข้ารหัสหรือถอดรหัส ที่ใช้คีย์ในการแปลงข่าวสาร มีอยู่ 2 ประเภทคือ

1. ระบบเข้ารหัสลับแบบคีย์เหมือน (Symmetric key cryptosystem) ผู้ส่งและผู้รับจะใช้คีย์ลับอันเดียวกันในการเข้า-ถอดรหัสข้อมูล เช่น DES,3DES,DESX,RC4,IDEA,Blowfish ดังแสดงไว้ในรูปที่ 3.2
2. ระบบเข้ารหัสลับแบบคีย์ต่าง (Asymmetric key cryptosystem) จะใช้คีย์ 2 คีย์ซึ่งมีความสัมพันธ์กันทางคณิตศาสตร์ประกอบด้วยคีย์สาธารณะ (Public key) และคีย์ส่วนตัว (Private key) โดยผู้ส่งและผู้รับใช้คีย์ต่างกัน เช่น RSA, DH, DSA ในการเข้ารหัสลับแบบคีย์ต่างนี้ ผู้ส่งจะใช้คีย์สาธารณะของผู้รับในการแปลงเพลนเท็กซ์ให้เป็นไซเฟอร์เท็กซ์ มีแต่ผู้รับเพียงคนเดียวเท่านั้นที่สามารถแปลงไซเฟอร์เท็กซ์ให้กลับเป็นเพลนเท็กซ์โดยใช้คีย์ส่วนตัวของผู้รับ รูปแสดงการเข้ารหัสแบบคีย์ต่างแสดงไว้ในรูปที่ 3.3

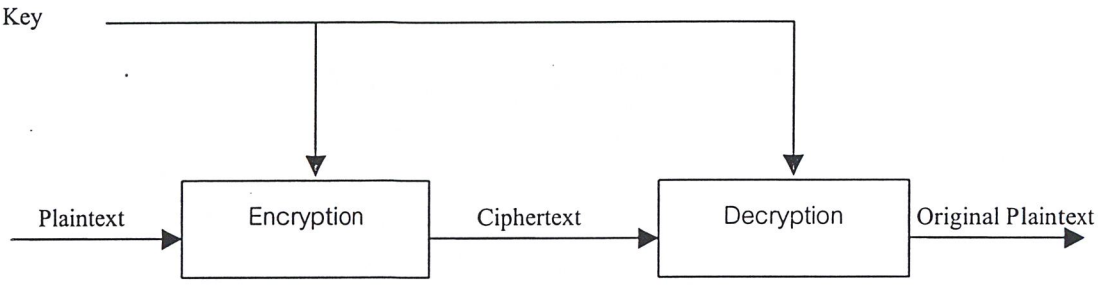
อัลกอริทึมการเข้ารหัส(Encryption algorithms) ใช้คีย์ (Key K) ดังนั้น ข้อมูล C จะมาสัมพันธ์กับข้อมูล P และ K จะได้ C = E(K,P)

E = Set ของ Encryption algorithms

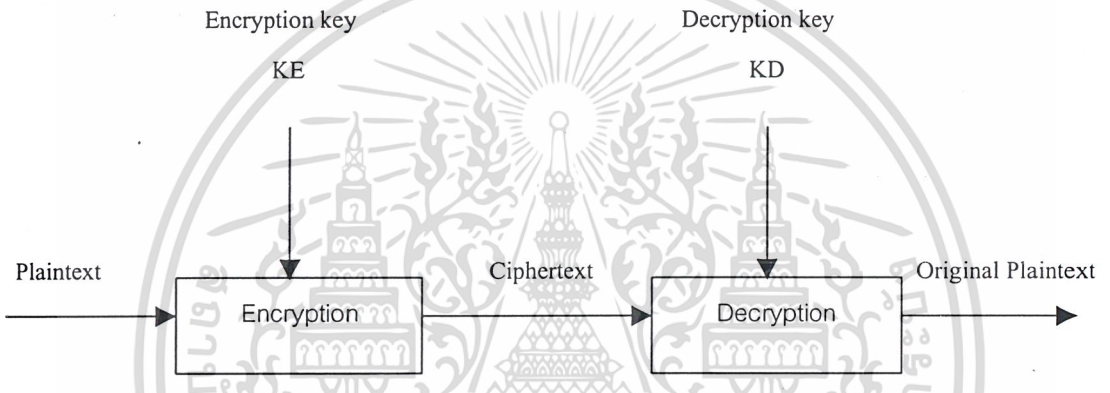
K = สมาชิกหนึ่งของ algorithms ซึ่งก็จะเป็นคีย์ที่ใช้นั่นเอง

ดังนั้นในบางครั้งการเข้ารหัสกับการถอดรหัสจะใช้คีย์ที่เหมือนกัน จะได้ P = D(K,E(K,P) ในบางครั้งจะใช้คู่กัน จะได้ P = D(Kp,E(Ke,P))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงบล็อกการเข้ารหัสถอดรหัสโดยใช้วิธีคีย์เหมือน



รูปที่ 3.3 แสดงบล็อกการเข้ารหัสถอดรหัสโดยใช้วิธีคีย์ต่าง

การเข้ารหัสของข้อมูลหนึ่งอาจเปลี่ยนแปลงได้โดยการเปลี่ยนคีย์(Key) ถ้าผู้อื่นทราบอัลกอริทึมการเข้ารหัส ก็จะไม่สามารถถอดรหัสนำข้อมูลไปใช้ได้เพราะไม่ทราบคีย์ วิธีการที่ไม่ต้องใช้คีย์ในการเข้ารหัสถอดรหัส จะเรียกไซเฟอร์เท็กซ์ว่าคีย์เลสไซเฟอร์เท็กซ์(Keyless Cipher)

ระบบความปลอดภัยเป็นวิธีการเข้ารหัส การใช้การเข้ารหัสในการซ่อนข้อมูล(พยายามส่งข้อมูลไปให้จากผู้ส่งไปยังผู้รับ) ตามกฎของมันการวิเคราะห์ระบบความปลอดภัย(Cryptanalyst) การหาความหมายของข้อมูลที่ถูกซ้อนวิเคราะห์ (Code) ทั้ง 2 อย่างนี้จะพยายามแปลงโค้ดไปสู่รูปแบบเริ่มต้นเพื่อป้องกันไม่ให้ผู้อื่นเข้ามายุ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การเข้ารหัสลับแบบคีย์เหมือน

การเข้ารหัสแบบคีย์เหมือนคือการเข้ารหัสแบบทั่วไปที่เราๆ เข้าใจกัน โดยมีคีย์ 1 คีย์นำมาเข้ากระบวนการต่างๆกัน เช่น เปลี่ยนลำดับที่, แทนที่, การเอ็กคลูซีฟออร์ (Exclusive-or) เวลาถอดรหัสก็ใช้การทำย้อนกลับ โดยนำคีย์เดียวกับที่ใช้เข้ารหัสมาใช้ในการถอดรหัส ซึ่งไม่ซับซ้อนมากทำให้ง่ายต่อการถอดรหัส ตัวอย่างของวิธีนี้ก็เช่น DES, 3DES เนื่องจากต้องใช้คีย์ดอกเดียวกันดังนั้นระหว่างผู้รับและผู้ส่งจึงจำเป็นต้องเก็บรักษาคีย์ดอกนี้เป็นความลับเพื่อความปลอดภัย

จุดเด่นและข้อด้อยของการเข้ารหัสแบบคีย์เหมือน

การเข้ารหัสแบบคีย์เหมือน ทำให้เกิดการกระทำในสองทิศทางระหว่างผู้ส่งและผู้รับ คือผู้ส่งและผู้รับสามารถที่จะเข้ารหัสข้อมูลแล้วส่งไปหาอีกคนได้ และในขณะเดียวกันนั้นก็สามารถที่จะถอดรหัสข้อมูลนั้นมาดูโดยการใช้คีย์เดียวกันนี้ ทำให้อัลกอริทึมที่ใช้นั้นง่ายไม่ค่อยยุ่งยากซับซ้อนมากนัก ออกแบบได้ง่าย การเข้ารหัสและการถอดรหัสนั้นทำได้อย่างรวดเร็วมากซึ่งถือเป็นข้อได้เปรียบของการเข้ารหัสแบบคีย์เหมือนที่เด่นมากที่สุด

ตราบใดที่คีย์ยังคงเก็บไว้เป็นความลับอยู่ระบบนี้ก็ยังคงปลอดภัย แต่ถ้าเมื่อใดที่ไม่ได้เป็นความลับแล้วก็จะไม่ปลอดภัยอีกต่อไป เพราะอาจถูกบุคคลอื่นปลอมแปลงข้อมูลเดิมเสียใหม่ หรือทำสำเนาข้อมูลอันนั้นไปแล้วจึงส่งไปให้ผู้รับตัวจริง โดยที่ตัวจริงนั้นไม่รู้เลยว่าถูกคนอื่นแอบอ่านข้อมูลนั้น หรือกว่าจะรู้ก็สายไปเสียแล้ว

ปัญหาและความปลอดภัยในการเข้ารหัสแบบคีย์เหมือน

1. ระบบที่ใช้คีย์เพียงดอกเดียวในการเข้ารหัสข้อมูลนั้น ถ้าคีย์ที่เป็นความลับถูกขโมย บุคคลที่รู้คีย์นั้นสามารถที่จะถอดรหัสข้อมูลที่เข้ารหัสนั้นได้ นอกจากนี้ยังอาจปลอมแปลงข้อมูลเดิมนั้นใหม่ แล้วเข้ารหัสข้อมูลที่ปลอมแปลงนั้นด้วยคีย์เดียวกัน แล้วส่งไปที่ผู้รับตัวจริง ดังนั้นเพื่อความปลอดภัยควรเลือกคีย์ที่ยากแก่การเดาและเก็บไว้อย่างปลอดภัย รวมทั้งไม่ควรใช้คีย์เดียวกันนี้ซ้ำกันหลายๆ ครั้ง
2. การเข้ารหัสแบบคีย์เดียวค่อนข้างที่จะอ่อน เสี่ยงต่อการพยายามที่จะถูกโจรกรรม เนื่องจากคีย์ที่ใช้อาจมีความยาวไม่มากพอหรืออัลกอริทึมที่ใช้นั้นค่อนข้างง่ายไม่ค่อยซับซ้อนมากนัก
3. การส่งคีย์ไปพร้อมกับข้อมูลที่เข้ารหัสนั้น อาจทำให้เกิดปัญหาได้ ซึ่งถ้าทำอย่างนั้นแล้วคีย์ที่ส่งไปด้วยนั้นจะต้องถูกส่งไปด้วยความปลอดภัยที่สูงมาก ยิ่งเป็นการส่งไปในระยะทางไกลๆ เช่น คนละเครือข่าย ซึ่งจะทำได้ยากมากเลยทีเดียว วิธีที่ง่ายก็คือให้ส่งคีย์ให้กับผู้รับด้วยมือของคนส่งเอง แต่อาจจะทำได้ไม่สะดวกและเสียเวลาอีกวิธีหนึ่งก็คือการส่งคีย์ไปพร้อมกับข้อมูลนั้น แต่แบ่งคีย์นั้นออกเป็นส่วนๆ ก่อนแล้วจึงส่งไปตามเส้นทางที่ต่างๆ กัน ซึ่งถึงแม้ว่าคีย์บางส่วนจะถูกดักได้แต่ก็ไม่สามารถรู้ถึงตัวคีย์ที่สมบูรณ์จริงๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 DES (Data Encryption Standard)

4.1.1 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

4.1.2 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 แต่ก็เป็นข้อถกเถียงกันเป็นอย่างมากถึงเรื่อง DES 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

4.1.3 อัลกอริทึมของการเข้ารหัสแบบ DES

โดยพิจารณาออกเป็น 2 ส่วนเพื่อให้ง่ายแก่การทำความเข้าใจคือ ส่วนที่เป็นคีย์ที่จะใช้ในการเข้ารหัสและส่วนที่เป็นข้อมูลที่จะนำมาทำการเข้ารหัส

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

คอนฟิวชัน คือการทำให้ชิ้นส่วนของข่าวสารถูกเปลี่ยนไป ดังนั้นเอาท์พุทบิทจะสังเกตไม่เห็นความสัมพันธ์กับอินพุทบิต

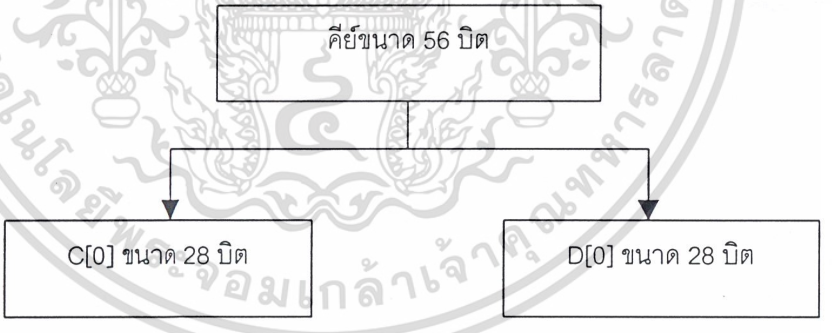
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

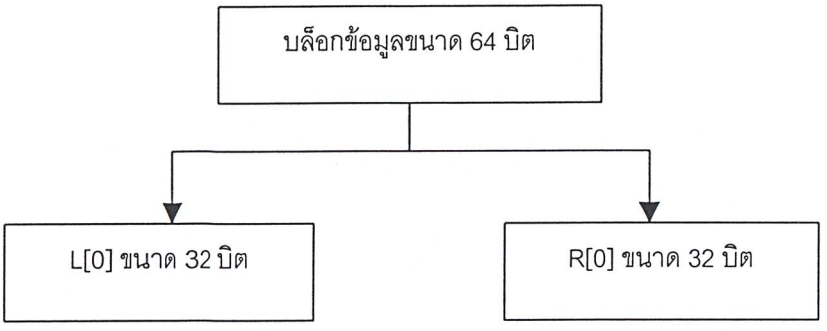
อินพุทที่เข้ามา DES จะแบ่งอินพุทออกเป็นบล็อกๆ ละ 64 บิต ซึ่งจะถูกเปลี่ยนไปใช้คีย์ขนาด 64 บิต ข้อมูลขนาด 64 บิต จะถูกสับเปลี่ยนตำแหน่งโดยการสับเปลี่ยนตำแหน่งเริ่มต้น (Initial Permutation) และคีย์จะถูกลดลงจาก 64 บิต เหลือ 56 บิต โดยการทิ้งบิตที่ 8, 16, 24, ..., 64 ซึ่งบิตเหล่านี้จะถูกกำหนดเป็นพาริตี้บิต โดยแสดงไว้ในรูปที่ 4.1

ข้อมูล 64 บิตที่ถูกสับเปลี่ยนตำแหน่งแล้วจะถูกแบ่งเป็นครึ่งซ้ายและครึ่งขวา (แต่ละครึ่งมีขนาด 32 บิต) แสดงไว้ในรูปที่ 4.2 คีย์จะถูกชิพไปทางซ้ายโดยการกำหนดที่จำนวนบิตและจะทำการสลับตำแหน่ง ต่อไปคีย์จะถูกรวมกับครึ่งขวา หลังจากนั้นก็จะมารวมกับครึ่งซ้ายใหม่อีกครั้ง ผลลัพธ์ของการรวมนี้จะเปลี่ยนเป็นครึ่งด้านขวาใหม่ ส่วนครึ่งขวาเก่าจะกลายมาเป็นครึ่งซ้ายใหม่ กิจกรรมเหล่านี้จะทำเป็นวัฏจักร (Cycle) วัฏจักรจะถูกทำซ้ำ 16 ครั้ง หลังจากวัฏจักรสุดท้ายซึ่งเป็นการสับเปลี่ยนครั้งสุดท้ายซึ่งจะถูกสับเปลี่ยนตำแหน่งบิตผกผันกับแบบเริ่มต้น (Inverse Initial Permutation (IP)) หรือการสับเปลี่ยนตำแหน่งบิตผกผันกับแบบเริ่มต้น ตามตารางจะได้ผลลัพธ์สุดท้ายออกมา คือข้อมูลที่ถูกทำการเข้ารหัสแล้ว



รูปที่ 4.1 แสดงการแบ่งคีย์ 56 บิตที่เรียงลำดับบิตแล้วออกเป็น 2 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงบล็อกข้อมูล 64 บิต ที่แบ่งออกเป็น 2 ส่วน หลังจากทำการจัดเรียงบิตแล้ว

4.1.4 โหมดต่างๆ ของการใช้ DES

มีโหมดอยู่ด้วยกันทั้งหมด 4 โหมดในการใช้งานเกี่ยวกับ DES ซึ่งได้แก่

1. โหมด ECB (Electronic Codebook)

   Electronic Codebook(ECB) เป็นวิธีที่ไม่ค่อยมีความยุ่งยากซับซ้อนมากนัก ข้อมูลที่จะนำมาเข้ารหัส จะถูกแบ่งออกเป็นบล็อกๆ บล็อกละ 64 บิต แต่ละบล็อกจะถูกนำมาเข้ารหัสโดยใช้คีย์ดอกเดียวกันหมดซึ่งผลลัพธ์ที่ได้ก็คือข้อมูลที่เข้ารหัสแล้วนั่นเอง การทำงานในลักษณะนี้ทำให้การเข้าและถอดรหัสทำได้อย่างรวดเร็ว เพราะสามารถทำการเข้าและถอดรหัสแบบขนานคือที่เดียวหลายๆ บล็อกได้ แต่ไม่ค่อยปลอดภัยนักเพราะถ้าข้อมูลที่จะนำมาเข้ารหัสซ้ำกันมากๆ พอเข้ารหัสแล้วก็จะได้บล็อกที่ซ้ำกันออกมาด้วย

2. โหมด CBC (Cipher Block Chaining)

   Cipher Block Chaining (CBC) โหมดนี้แก้ไขข้อเสียของโหมด ECB ถึงแม้ว่าบล็อกข้อมูลที่จะนำมาเข้ารหัสจะซ้ำกัน แต่ก็จะได้บล็อกข้อมูลที่เข้ารหัสออกมาไม่เหมือนกัน ที่เป็นเช่นนี้เพราะว่าบล็อกข้อมูลที่จะเข้ารหัสนั้นจะทำการเอ็คคลูซีฟออร์กับบล็อกที่เข้ารหัสแล้วที่ได้มาซึ่งอยู่ก่อนบล็อกของตัวมันเองก่อน แล้วจึงนำผลที่ได้ไปเป็นข้อมูลที่ใช้เข้ารหัสอีกทีหนึ่ง โดยบล็อกแรกจะทำการเอ็คคลูซีฟออร์กับตัว Initialization vector (IV) ซึ่งอาจได้มาจากการแรนดอม ซึ่งค่าของ IV นี้ระหว่างผู้ส่งและผู้รับต้องรู้กัน โหมดนี้ทำงานช้ากว่าโหมดแรกเนื่องจากไม่สามารถทำการเข้าหรือถอดรหัสแบบขนานได้ และถ้าบล็อกข้อมูลมีข้อผิดพลาดขึ้นก็จะมีผลต่อบล็อกต่อมาด้วย

3. โหมด CFB (Cipher Feedback)

   Cipher Feedback (CFB) โหมดนี้สามารถทำงานแบบ real time ได้โดยสามารถทำการเข้ารหัสแล้วส่งข้อมูลขนาด 1 ไบต์หรือครั้งละหนึ่งตัวอักษรได้ โดยไม่ต้องรอข้อมูลจนครบบล็อกก่อนแล้วค่อยทำงานโดยปกติตัวอักษรที่ใช้จะแทนด้วยข้อมูล 8 บิต ถ้าเกินกว่านี้ก็จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

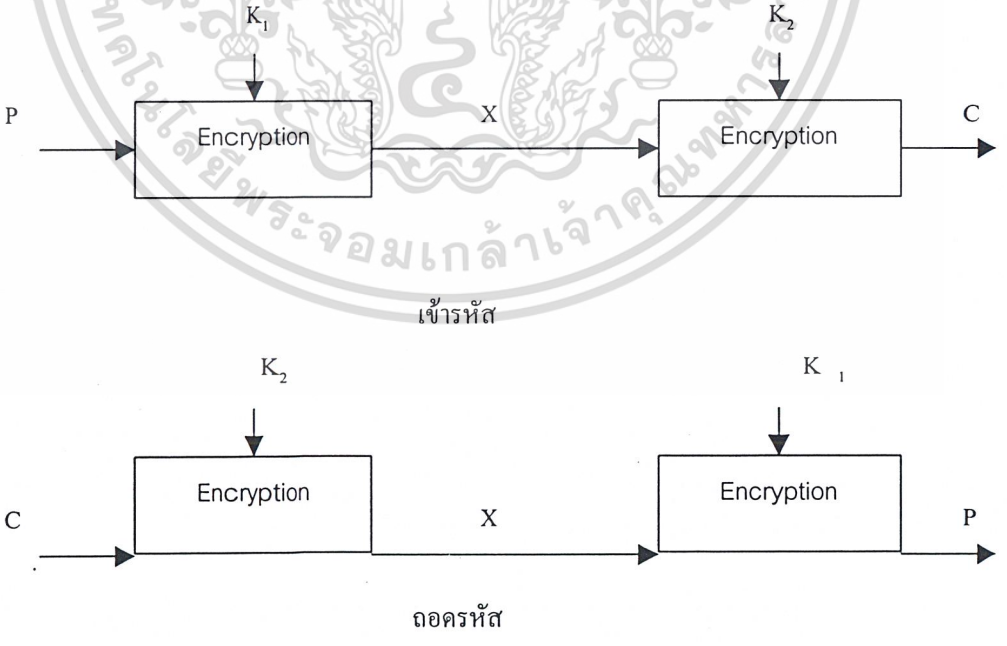
ทำให้ประสิทธิภาพลดลง การทำงานก็คือให้ตัวอักษรแทนด้วยข้อมูลขนาด j บิต ข้อมูลที่จะนำมาเข้ารหัสจะเก็บอยู่ในรีจีสเตอร์ที่เลื่อนค่าได้ ขนาด 64 บิต โดยเริ่มแรกจะเป็นค่า Initialization vector ก่อน แล้วรีจีสเตอร์จะถูกเลื่อนค่าไปทางซ้าย j บิต แล้วนำค่าของรีจีสเตอร์นี้มาเข้ารหัส ได้ผลลัพธ์ขนาด 64 บิตออกมาเลือกเอา j บิตจากทางด้านซ้ายมาทำการเอ็คคลูซีฟออร์กับข้อมูลที่เป็นตัวอักษร ผลลัพธ์ที่ได้จะเป็นข้อมูลที่เข้ารหัสเสร็จแล้วจริงๆ แล้วนำผลลัพธ์นี้ไปเป็นอินพุทสำหรับการเข้ารหัสครั้งต่อไป โดยนำไปใส่ที่ j บิตทางด้านขวาของรีจีสเตอร์ที่เลื่อนค่าได้ภายหลังจากได้เลื่อนค่าในรีจีสเตอร์ไปทางซ้าย j บิตแล้ว

4. โหมด OFB (Output feedback)

Output feedback (OFB) คล้ายกับโหมด CFB เพียงแต่เอาพุทที่ได้หลังจากเข้ารหัสซึ่งมีขนาด 64 บิต จะเอา j บิตจากทางซ้ายมาใส่ในรีจีสเตอร์แทน

4.2 Double DES

เนื่องจาก DES แบบธรรมดานั้นมีข้อด้อยที่ความยาวของคีย์ที่ใช้ในการเข้ารหัสที่มีขนาด 64 บิต ซึ่งค่อนข้างจะเล็กมากความเป็นไปได้ของคีย์ทั้งหมดคือ 2^{56} (ตัดพาริตี้บิตออกเหลือเพียงแค่ 56 บิต) หรือประมาณ 7×10^{16} ซึ่งถ้าอาศัยคอมพิวเตอร์ที่มีประสิทธิภาพสูงในปัจจุบัน หรืออาจใช้หลายๆตัวช่วยกัน ก็สามารถถอดรหัสข้อมูลได้โดยไม่ต้องรู้คีย์ การแก้ไขปัญหาดังกล่าวก็คือการใช้คีย์ที่มีขนาดยาวกว่าเดิม คือใช้คีย์ 2 ดอกซึ่งแตกต่างกันในการเข้ารหัสครั้งแรกและครั้งที่สองตามลำดับดังรูปที่ 4.3



รูปที่ 4.3 แสดงการเข้ารหัสและถอดรหัสในแบบ Double DES

ซึ่งใช้ $C = E_{k2}(E_{k1}(P))$ และ $P = D_{k2}(D_{k1}(C))$ ในการเข้าและถอดรหัสข้อมูลตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้วิธี Double DES ในการเข้ารหัสซึ่งเราหวังไว้ว่าจะเพิ่มโอกาสของคีย์ที่ใช้จากเดิม 2^{56} เป็น 2^{102}อย่างไรก็ตามก็ยังมีวิธีการที่เรียกว่า meet-in-the-middle ซึ่งจะใช้การสุ่มคีย์เพียงแค่ 2^{56+1} ครั้งเท่านั้นเอง โดยวิธีดังกล่าวก็คือ สมมติว่าเรามีข้อมูลต้นฉบับคือ x_1 , x_2 และมีข้อมูลที่เข้ารหัสแล้วที่สอดคล้องกันคือ y_1 , y_2 หรือจะได้ว่า

$y_1 = E_{k2}(E_{k1}(x_1))$ และ $y_2 = E_{k2}(E_{k1}(x_2))$

แล้วใช้หลักการคือ

$$D_{k2}(y_1) = D_{k2}(E_{k2}(E_{k1}(x_1))) = E_{k1}(x_1)$$

วิธี meet-in-the-middle นี้จะทำการเข้ารหัสข้อมูล x_1 โดยใช้คีย์ที่เป็นไปได้ทั้งหมดคือ2^{56} แล้วเอาผลลัพธ์ที่ได้ทั้งหมดไปเก็บไว้ในหน่วยความจำ จากนั้นทำการถอดรหัสข้อมูล y_2 โดยใช้คีย์ทั้งหมดที่เป็นไปได้คือ 2^{56} ดอกเช่นกัน แล้วนำผลลัพธ์ที่ได้ทั้งหมดมาเปรียบเทียบกับผลลัพธ์ที่เก็บไว้ในหน่วยความจำตอนแรกเพื่อหาผลลัพธ์ที่ตรงกัน ซึ่งตัวที่ตรงกันก็จะทำให้เราทราบคีย์ทั้งสองดอกคือ k_1 และ k_2 แล้วใช้ข้อมูล x_2 และ y_2 ตรวจสอบอีกทีหนึ่งว่าคีย์ที่ได้ถูกต้องหรือไม่

4.3 3DES(Triple Data Encryption Standard)

บางครั้งการเข้ารหัสแบบ DES ที่ใช้คีย์ที่เข้าขนาด 56 บิต ก็อาจจะถูก crack ข้อความได้ง่าย ดังนั้นหากเราเพิ่มขนาดของคีย์โดยการใช้ การเข้ารหัสแบบมัลติเพิลเข้าช่วย โดยเข้ารหัสข้อความ 3 ครั้ง ขนาดของคีย์ก็จะเพิ่มขึ้นจาก 56 บิตเป็น 56×3 = 168 บิต ซึ่งยากต่อการคาดเดาได้มากขึ้น

โหมดต่างๆของการเข้ารหัสแบบ 3DES

- DES-EEE3 : เป็นการเข้ารหัสแบบ DES 3 ครั้ง โดยใช้คีย์ที่เข้าแตกต่างกัน 3 คีย์ ซึ่งเขียนเป็นสมการได้ดังนี้

$$C = E_{k1}[E_{k2}[E_{k3}(P)]]$$

- DES-EDE3 : เป็นการเข้ารหัสแบบ DES 3 ครั้งโดยลำดับการเข้าเป็น เข้ารหัส ถอดรหัส แล้วก็เข้ารหัสอีกที โดยใช้คีย์ที่เข้ารหัสต่างกัน 3 คีย์ ซึ่งอาจเขียนเป็นสมการในการเข้ารหัสง่ายๆ ดังนี้

$$C = E_{k1}[D_{k2}[E_{k3}(P)]]$$

- DES-EEE2 และ DES-EDE2 : เหมือนโหมดการเข้ารหัสแบบที่กล่าวมา ต่างกันเพียงลำดับในการเข้าครั้งที่ 1 และครั้งที่ 3 ใช้คีย์อันเดียวกัน ซึ่งอาจจะเขียนเป็นสมการได้ดังนี้

$C = E_{k1}[D_{k2}[E_{k1}(P)]]$ ใน EDE 2 โหมด

$C = E_{k1}[E_{k2}[E_{k1}(P)]]$ ใน EEE 2 โหมด

DES มีความระมัดระวังและความซับซ้อนในการรวม 2 บล็อกพื้นฐาน (Fundamental Building Block) ของการเข้ารหัส มีอัลกอริทึมคือจะมีการทำโปรแกรมซ้ำของทั้ง 2 เทคนิค ซึ่งจะมี 16 วัฏจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

(cycle) สูงสุด ในความซับซ้อนของการติดตาม แต่ละบิตทำซ้ำถึง 16 ครั้งของการแทนที่และการสับเปลี่ยนตำแหน่ง

เพลนเท็กซ์จะถูกเข้ารหัสเป็นบล็อกๆ ละ 64 บิต และคีย์ก็จะมีความยาว 64 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42784

บทที่ 5
การเข้ารหัสลับแบบคีย์ต่าง

แนวความคิดของวิชาการเข้ารหัสลับแบบคีย์ต่างได้เกิดขึ้นจากการพยายามแก้ไขปัญหาของการเข้ารหัสลับแบบคีย์เหมือน 2 ข้อด้วยกันคือ การจัดการคีย์ (Key Management) และการลงนามรับรองข่าวสารทางดิจิตอล (Digital Signature) เพื่อใช้ในการพิสูจน์สิทธิ์ ระบบการเข้ารหัสเป็นตัวอย่างหนึ่งของการป้องกันการบุกรุกข้อมูลในคอมพิวเตอร์ จุดประสงค์พื้นฐานของการเข้ารหัสแบบคีย์ต่าง คือเพื่อป้องกันไม่ให้มีการเปิดเผยข้อมูลแก่ผู้ที่ไม่มีสิทธิ์ โดยจะมีคีย์อยู่ 2 คีย์คือคีย์สาธารณะและคีย์ส่วนตัว ซึ่งมีความสัมพันธ์กันทางคณิตศาสตร์ โดยผู้เข้ารหัสและถอดรหัสจะใช้คีย์ต่างกัน

ระบบการเข้ารหัสลับ (Cryptosystems) มีบริการที่สำคัญให้อยู่ 3 ประเภทก็คือ

1. ความลับ (Secrecy) หมายถึงการไม่ยอมให้บุคคลผู้ไม่มีสิทธิ์เข้ามาดูข้อมูล
2. การพิสูจน์บุคคล (Authenticity) หมายถึงการตรวจสอบที่มาของข้อมูล ว่าถูกส่งมาจากผู้ส่งบุคคลคนนั้นจริงหรือไม่
3. ความคงตัว (Integrity) หมายถึงการทำให้มั่นใจว่าข่าวสารไม่ได้ถูกเปลี่ยนแปลงแก้ไขโดยการแทนที่ แทรก หรือลบเนื้อหาข่าวสารเดิม ไม่ว่าโดยเจตนาหรือไม่เจตนาก็ตาม

บางระบบอาจมีบริการเกี่ยวกับนอนเรพิวเตชัน (Nonrepudiation) คือการใช้การเข้ารหัสลับแบบคีย์ต่างลงลายมือชื่อดิจิตอลลงในเอกสาร ผู้รับสามารถตรวจสอบได้ว่าข้อมูลมาจากใคร ทำให้ผู้ส่งไม่สามารถที่จะปฏิเสธเอกสารนี้ได้ เนื่องจากการลงนามรับรองจะรวมเอาลายมือชื่อจากข่าวสารต้นฉบับซึ่งเป็นสิ่งสำคัญในการทำเอกสารสัญญาอิเล็กทรอนิกส์ในสื่อต่างๆ เช่น อินเทอร์เน็ต

อัลกอริทึมสำหรับการเข้ารหัสลับแบบคีย์ต่างจะมีคุณสมบัติที่สำคัญคือ แม้จะรู้อัลกอริทึมและคีย์ในการเข้ารหัส แต่จะไม่สามารถคำนวณหาคีย์ที่ใช้ในการถอดรหัสได้โดยง่าย

แอปพลิเคชันของการเข้ารหัสแบบคีย์ต่างต้องมีโครงสร้างการทำงานด้านการพิสูจน์บุคคล (Authentication Framework) ซึ่งรวมเอารายละเอียดต่างๆ ของผู้ใช้เอาไว้ด้วยกัน การรับรองโดยระบบคีย์ต่างเป็นการพิสูจน์โดยอ้างหลักฐานที่มีการรับรองโดยองค์กรพิสูจน์สิทธิ์ (Certification Authority, CA) ทำให้ผู้ใช้ไม่จำเป็นต้องตรวจสอบที่มาของข่าวสารด้วยตัวเอง

5.1. อาร์เอสเอ (RSA)

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

$$C = M^e \bmod n$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 $M = C^d \bmod n$

5.2 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

รูปแบบในการเข้ารหัส	ข้อดี	ข้อเสีย
การเข้ารหัสแบบคีย์เหมือน	รวดเร็ว สร้างได้ง่ายโดยใช้ฮาร์ดแวร์	คีย์ของการเข้ารหัสและถอดรหัส ต้องเหมือนกัน การจัดส่งคีย์ทำได้ยาก
การเข้ารหัสแบบคีย์ต่าง	ใช้คีย์ต่างกันในการเข้ารหัสและ ถอดรหัส การจัดส่งคีย์ทำง่าย สามารถตรวจสอบผู้ใช้ได้โดยใช้ ร่วมกับลายมือชื่อดิจิตอล	ค่อนข้างช้าและต้องใช้การคำนวณ อย่างมาก

ตารางที่ 5.1 เปรียบเทียบข้อดีและข้อเสียของการเข้ารหัส

ค่าใช้จ่าย	ความยาวของคีย์ (บิต)				
	40	56	64	80	128
$ 100,000	2 วินาที	35 ชั่วโมง	1 ปี	70,000 ปี	10^{19} ปี
$ 1 ล้าน	0.2 วินาที	3.5 ชั่วโมง	37 วัน	7,000 ปี	10^{18} ปี
$ 100 ล้าน	2 มิลลิวินาที	2 นาที	9 ชั่วโมง	70 ปี	10^{16} ปี
$ 1,000 ล้าน	0.2 มิลลิวินาที	13 วินาที	1 ชั่วโมง	7 ปี	10^{15} ปี
$ 100,000 ล้าน	2 ไมโครวินาที	0.1 วินาที	32 วินาที	24 วัน	10^{13} ปี

ตารางที่ 5.2 ประมาณเวลาและค่าใช้จ่ายสำหรับการหาคีย์ที่ใช้ถอดรหัสที่มีความยาวต่างๆกัน

จากตารางที่ 5.1 จะแสดงการเปรียบเทียบระหว่างคีย์ต่าง และคีย์เหมือน ซึ่งจะเห็นได้ว่าการเข้ารหัสโดยใช้คีย์เหมือนนั้นทำได้ง่าย แต่จะต้องแลกกับความปลอดภัยของข้อมูลที่มีน้อยกว่าคีย์ต่าง และใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 จะเป็นการเปรียบเทียบเวลาและ ค่าใช้จ่ายที่ต้องใช้สำหรับการหาคีย์ที่ถูกต้องในการถอดรหัส ซึ่งเมื่อคีย์มีความยาวมากขึ้น จะส่งผลให้ความปลอดภัยของข้อมูลสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
แฮชฟังก์ชัน

6.1 วิธีแฮช

วิธีแฮชเป็นเครื่องหมายตรวจสอบที่ป้องกันข้อมูลจากการแก้ไข ถ้าคุณต้องการป้องกันข้อมูลจากการแก้ไขที่ไม่สามารถตรวจสอบได้ คุณจะต้องทำแฮชและเก็บผลลัพธ์ไว้ ภายหลังคุณหรือคนอื่นสามารถทำแฮชอีกครั้งแล้วนำผลลัพธ์ครั้งใหม่ไปเปรียบเทียบกับผลลัพธ์ครั้งก่อนนั้น ถ้าแตกต่างกันแสดงว่าข้อมูลได้มีการเปลี่ยนแปลงแก้ไข แต่ถ้าผลลัพธ์เหมือนกันก็แสดงว่าข้อมูลไม่ได้ถูกแก้ไข

แฮชฟังก์ชันผลิตรูปแบบที่ลดทอนมาจากข้อมูลทั้งหมดจนกระทั่งถึงการเปลี่ยนแปลงของข้อมูลก็จะทำให้รูปแบบนี้เปลี่ยนแปลงไปด้วย ผลลัพธ์นี้บางครั้งเรียกไดเจสท์ (Digest) หรือค่าตรวจสอบ (Check Value)

6.2 คำบรรยายวิธีแฮช

ตัวอย่างวิธีแฮชแบบง่ายๆ คือการเอ็กซ์คลูซีฟออร์ (exclusive or) ของบิตทั้งหมดของตัวข้อมูล ฟังก์ชันนี้ได้ลดข้อมูลไปที่บิตเดียว แต่ด้วยผลของแค่บิตเดียวคุณคาดเดาได้ว่าครึ่งหนึ่งของข้อมูลจะได้ค่าแฮช 0 และอีกครึ่งได้ 1 ดังนั้นถ้ามีการเปลี่ยนแปลงที่ข้อมูลน่าจะมีเพียงการเปลี่ยนแปลงแค่ 0.5 ของค่าแฮช

โดยการเปรียบเทียบความแตกต่าง ถ้าข้อมูลมี 8-บิต ไบต์ (8-bit bytes) และแฮชฟังก์ชันจะคำนวณเอ็กซ์คลูซีฟออร์ของข้อมูลทั้งหมด มีค่าแฮชที่เป็นไปได้แตกต่างกัน 2^8 หรือ 256 ค่า ดังนั้นถ้ามีการเปลี่ยนแปลงข้อมูลจะมีเพียงการเปลี่ยนแปลงแค่ 1/256 = 0.003906 ซึ่งไม่มีผลต่อการเปลี่ยนแปลงค่าแฮช แฮชฟังก์ชันลดรูปของข้อมูลใหญ่ๆไปเป็นผลลัพธ์เล็กๆ เรียกว่าไดเจสท์ โดยทั่วไปไดเจสท์ที่เล็กกว่าก็มีค่าข้อมูลที่ต้องแมป (map) ไปมากกว่า และมีแนวโน้มที่จะข้อมูลเปลี่ยนแล้วไดเจสท์จะไม่เปลี่ยนมากกว่า ด้วยเหตุนี้ไดเจสท์มีแนวโน้มที่จะเล็กแต่ไม่เล็กมากจะอยู่ระหว่าง 100 ถึง 1,000 บิต

ทั้งหมดง่ายต่อการกลับ ถ้ามีผู้โจมตี (attacker) สามารถคาดเดาผลลัพธ์ได้โดยง่าย ถ้าวิธีนี้ถูกใช้ในเป็นวิธีแฮช ผู้โจมตีจะง่ายต่อการเปลี่ยนแปลงแก้ไขข้อมูลโดยไม่ทำให้ค่าแฮชเปลี่ยนไปแต่อย่างใด

แฮชฟังก์ชันแบบคริปโตกราฟิก (cryptographic hash function) ใช้ฟังก์ชันแบบคริปโตกราฟิกเป็นส่วนหนึ่งของแฮชฟังก์ชัน ด้วยแฮชฟังก์ชันแบบคริปโตกราฟิก ผู้ส่งสามารถคำนวณค่าแฮชของบล็อคข้อมูลที่จะส่งและส่งทั้งข้อมูลและค่าแฮชไป ผู้รับที่ถูกต้องจะมีฟังก์ชันแบบคริปโตกราฟิกและคำนวณค่าแฮชของข้อมูลที่ได้รับ ผู้บุกรุกจะไม่สามารถมีฟังก์ชันแบบคริปโตกราฟิก ผู้บุกรุกสามารถทำการเปลี่ยนแปลงข้อมูลหรือค่าแฮชหรือทำทั้งคู่ แต่ถ้าไม่รู้ความสัมพันธ์ระหว่างข้อมูลกับแฮชฟังก์ชัน ผู้บุกรุกไม่สามารถจะแก้ไขทั้งสองอันในวิธีที่เข้ากันได้ ดังนั้นการแก้ไขเปลี่ยนแปลงสามารถตรวจพบโดยผู้รับ ด้วยการพึ่งพาความน่าจะเป็นบนความแข็งแกร่งของวิธีคริปโตกราฟิกและระดับที่ข้อมูลถูกลดรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 วิธีแฮชที่ปลอดภัย (Secure Hash Algorithm)

เพื่อที่จะจัดให้มีมาตราฐานของลายมือชื่อดิจิตอล NIST จัดมาตรฐานวิธีแฮช วิธีนี้เรียกวิธีแฮชที่ปลอดภัย (Secure Hash Algorithm) (SHA) ถูกออกแบบโดยยูเอสเนชันแนลเซอเคียวริตี(U.S. National Security Agency (NSA))เพื่อที่จะทำงานกับวิธีลายมือชื่อดิจิตอล วิธีนี้ให้อินพุทมีความยาวน้อยกว่า 2^{64} บิต ซึ่งจะถูกลดรูปให้เหลือไดเจสท์เพียง 160 บิต

สัญลักษณ์นี้ $\oplus$ หมายถึงเอ็กซ์คลูซีฟออร์ $\vee$ หมายถึงออร์ (or) $\wedge$ หมายถึงแอนด์ (and) และ $\neg$ หมายถึงนอต (not) โดยให้ฟังก์ชัน S(n,v) หมายถึง v เลื่อนไปทางซ้าย n ตำแหน่งเป็นวงกลม ทั้งหมดหารเศษด้วย 2^{32} (mod 2^{32})

ขั้นแรกข้อความจะถูกต่อด้วยเลข 1 หนึ่งตัว, 0 หลายๆตัว ให้เต็ม และ ค่า 64 บิตสำหรับจำนวนบิตในข้อความเดิม เพื่อว่าความยาวรวมของข้อความ ความยาวเลข 1 และ 0 หลายๆตัวและค่า 64 บิตเป็นเท่าของ 512 บิต

ข้อความจะถูกทำในบล็อกของเวิร์ด 16 ตัว(32-bits words จำนวน 16 ตัว) แสดงได้ดังนี้ W(0), W(1), W(2),...W(15) โดยที่ W(0) เป็นซ้ายสุด แต่ละบล็อกถูกขยายให้เป็นเวิร์ด 80 ตัว โดย W(t) := W(t-3) $\oplus$ W(t-8) $\oplus$ W(t-14) $\oplus$ W(t-16) สำหรับ t :=16 ถึง 79 จะได้ค่าคงที่ 5 ตัวคือ H0 := 67452301, H1 := EFCDAB89, H2 := 98BADCFE, H3 := 10325476 และ H4 := C3D2E1F0 (แสดงด้วยเลขฐาน 16) โดยมีวิธีแสดงดังรูป 6.1

หลังจากทำเสร็จสมบูรณ์แล้ว จะไดเจสท์ขนาด 160 บิตจะมี เวิร์ด 5 ตัวคือ H0 H1 H2 H3 H4

วิธีนี้แสดงความกระจาย กระบวนการของการกระจายผลกระทบของบิตของเพลนเท็กซ์ตลอดทั้งไซเฟอร์เท็กซ์ ตัวอย่างเช่นในการขยายเริ่มแรกของเวิร์ด 16 ตัวไปเป็น 80 ตัว การเปลี่ยนแปลงในเวิร์ดที่ 14 มีผลโดยตรงกับเวิร์ดที่ 17, 22, 28 และ 30 แต่การเปลี่ยนในเวิร์ดที่ 17 จะมีผลกับเวิร์ดที่ 20, 25, 31 และ 33 การเปลี่ยนแปลงในเวิร์ดที่ 22 มีผลกับเวิร์ดที่ 25, 30, 36 และ 41 และไปเรื่อยๆ บิตบิตเดียวมีการเปลี่ยนแปลงในเวิร์ด 14 มีผลต่อเวิร์ด 17, 20, 22, 25, 28, 30, 31, 33, 34, 36, 37, 39, 41, 42, 44, 45, 47, 47, 49 และต่อไปเรื่อยๆ หลังจาก 80 ขั้นตอนแต่ละ W(I) จะมีผลต่อ TEMP ซึ่งจะมีผลต่อ A, B, C, D และ E ซึ่งจะส่งผลต่อ H0 ถึง H4 แต่ละตัว และจะออกมาเป็นผลลัพธ์ขั้นสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for each 16-word block begin
        A := H0; B := H1; C := H2; D := H3; E := H4
        for I := 0 to 19 begin
                TEMP := S(5,A) + ((B∧C) ∨ (¬B∧D)) + E + W(I) + 5A827999;
                E := D; D := C; C := S(30,B); B := A; A := TEMP
        end
        for I := 20 to 39 begin
                TEMP := S(5,A) + ((B⊕ C⊕D) + E + W(I) + 6ED9EBA1;
                E := D; D := C; C := S(30,B); B := A; A := TEMP
        end
        for I := 40 to 59 begin
                TEMP := S(5,A) + ((B∧C) ∨ (B∧D) ∨ (C∧D)) + E + W(I) + 8F1BBCDC;
                E := D; D := C; C := S(30,B); B := A; A := TEMP
        end
        for I := 60 to 79 begin
                TEMP := S(5,A) + ((B⊕ C⊕D) + E + W(I) + CA62C1D6;
                E := D; D := C; C := S(30,B); B := A; A := TEMP
        end
        H0 := H0+A; H1 := H1+B; H2 := H2+C; H3 := H3+ D; H4 := H4+E
end
```

รูปที่ 6.1 วิธีแฮชที่ปลอดภัย (Secure Hash Algorithm)

สรุปแล้วแฮชฟังก์ชันเป็นฟังก์ชันทางเดียวสามารถคำนวณหาผลลัพธ์ได้ง่ายแต่การคำนวณย้อนกลับทำได้ยาก เราสามารถนำประโยชน์ของแฮชฟังก์ชันมาใช้ในวิธีการทำลายมือชื่อดิจิตอล โดยใช้ในขั้นตอนของการทำแมสเซจไดเจส ถ้า f เป็นแฮชฟังก์ชันทางเดียว (one-way hash function) แล้ว f จะมีลักษณะดังต่อไปนี้

1) สามารถคำนวณหาค่า f(x) โดยค่าของ x มีขนาดเท่าใดก็ได้แต่จะได้ค่า f(x) ที่มีขนาดคงที่ ค่าของ f(x) ที่ได้ออกมา จะถูกเรียกว่าเป็นแฮชแวลลู ซึ่งผลที่ได้จากการทำแฮชฟังก์ชันที่ต่างกันจะได้ผลลัพธ์ออกมามีขนาดที่ไม่เท่ากัน แต่ถ้าเลือกใช้ฟังก์ชันเดียวกันผลที่ได้ออกมาจะมีขนาดเท่ากัน
2) เมื่อกำหนดค่า x จะสามารถหาค่า f(x) ได้ง่าย
3) เมื่อกำหนดค่า y ซึ่ง y = f(x) การคำนวณหาค่า x จะทำได้ยากมาก
4) สำหรับค่า x_1 และ x_2 ใดๆ ที่ x_1 ไม่เท่ากับ x_2 จะไม่สามารถหาค่าที่ทำให้ $f(x_1) = f(x_2)$ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการใช้แฮชฟังก์ชันทางเดียว คือ MD4, MD5 และ Secure Hash Algorithm (SHA) โดยถ้าเลือกใช้ MD4 หรือ MD5 แฮชแวลลูที่ได้จะมีขนาดเท่ากับ 128 bit แต่ถ้าเลือกใช้ SHA จะได้แฮชแวลลูที่มีขนาด 160 bit แนวคิดฟังก์ชันทางเดียวนำไปสู่การคิดค้นวิธีการเข้ารหัสลับแบบคีย์ต่าง โดยระบบการเข้ารหัสลับแบบคีย์ต่างที่นิยมใช้กันอย่างแพร่หลายคือ อาร์เอสเอ (RSA) โดยตั้งชื่อตามอักษรย่อของผู้คิดค้น 3 คนคือ ไรเวสต์ (Rivest), ชาเมียร์(Shamir) และอาเดลแมน (Adelman) ที่สถาบันเทคโนโลยีแมซซาชูเซทซ์ (Massachusetts Institute of Technology หรือ MIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
เอกสารสิทธิ์ (Certificate)

เอกสารสิทธิ์เป็นเหมือนกับบัตรประจำตัวของบุคคลนั้น ซึ่งจะบ่งบอกรายละเอียดของบุคคล เราสามารถส่งเอกสารสิทธิ์ไปพร้อมกับลายมือชื่อดิจิตอลเพื่อใช้ในการอ้างถึงผู้ส่ง การสร้างเอกสารสิทธิ์ทำโดยผู้ใช้ทุกคนทำการติดต่อกับองค์กรพิสูจน์สิทธิ์ (Certificate Authority) โดยการส่งคีย์สาธารณะของตนเองไปและทำการขอเอกสารสิทธิ์มา การติดต่อต้องทำเป็นการส่วนตัวหรือติดต่อผ่านระบบการพิสูจน์บุคคลที่ปลอดภัย เราสามารถกำหนดสิ่งที่จำเป็นในการสื่อสารดังต่อไปนี้

1. ผู้ใช้ทุกคนสามารถคำนวณหาชื่อและคีย์สาธารณะของเจ้าของเอกสารสิทธิ์ได้
2. ผู้ใช้ทุกคนสามารถตรวจสอบได้ว่าเอกสารสิทธิ์มาจากองค์กรผู้รับรองสิทธิ์จริงๆ ไม่ได้ถูกปลอมแปลงมา
3. ผู้ใช้สามารถตรวจสอบได้ว่าเอกสารสิทธิ์นั้นหมดอายุหรือไม่
4. ผู้ที่สามารถสร้างและอัพเดตเอกสารสิทธิ์ได้มีเพียงองค์กรผู้มีอำนาจในการรับรองสิทธิ์เท่านั้น
5. ผู้ใช้ทุกคนสามารถตรวจสอบเอกสารสิทธิ์ได้เป็นประจำ

7.1 ความสำคัญของเอกสารสิทธิ์ดิจิตอล (Digital Certificate)

ในระบบการเข้ารหัสแบบคีย์ต่างนั้นเราจะต้องมีการสร้างทั้งคีย์ส่วนตัวและคีย์สาธารณะ ซึ่งโดยทั่วไปการสร้างคีย์จะทำโดยโปรแกรมที่จะใช้คีย์นั้นเช่น โปรแกรมเว็บบราวเซอร์หรือโปรแกรมติดต่ออิเล็กทรอนิกส์เมล์ หลังจากที่สร้างคีย์ทั้งสองเรียบร้อยแล้ว เราจะต้องเก็บรักษาคีย์ส่วนตัวไว้ให้ดีอย่าให้ใครมาแอบเห็นหรือขโมยไปได้ จากนั้นจะเป็นการตัดสินใจว่าจะทำการแจกจ่ายคีย์สาธารณะของเราไปสู่ผู้อื่นด้วยวิธีใด เช่นอาจแจกคีย์โดยส่งอิเล็กทรอนิกส์เมล์ไปให้เพื่อนหรือบุคคลที่ติดต่อกับเรา แต่วิธีนี้เราอาจส่งคีย์ไปให้ไม่ครบทุกคน และยังต้องเป็นภาระคอยจัดการส่งคีย์ให้กับบุคคลใหม่ๆ ที่ต้องการติดต่อด้วย นอกจากนี้ยังไม่สามารถทำให้ผู้รับมั่นใจได้ว่าคีย์ที่ส่งไปให้นั้นเป็นคีย์ของเราจริงๆ เนื่องจากอาจมีผู้อื่นแอบสร้างคีย์โดยใช้ชื่อเราและแอบอ้างส่งคีย์ดังกล่าวให้กับผู้อื่นเพื่อให้เข้าใจว่าเป็นคีย์ของเราก็ได้

วิธีที่ดีกว่าก็คือการใช้ระบบแจกจ่ายคีย์ที่เชื่อถือได้ ซึ่งจะมีผู้ออกคีย์ที่เรียกว่า ผู้ออกเอกสารสิทธิ์ดิจิตอล (Digital certificate authority) โดยผู้ออกเอกสารสิทธิ์นี้จะตรวจสอบคีย์สาธารณะของเราด้วยหลักฐานว่าคีย์นั้นเป็นของเราจริงๆ พร้อมทั้งตรวจสอบข้อมูลส่วนตัวของเรา (ข้อมูลที่ตรวจสอบจะมากน้อยแค่ไหนขึ้นอยู่กับระดับชั้นของการรับรอง) เมื่อผู้อื่นได้รับคีย์ของเราก็สามารถที่จะตรวจสอบกับผู้ออกเอกสารสิทธิ์นี้ว่าคีย์ที่ได้รับเป็นของเราจริงหรือไม่ ซึ่งตัวเอกสารสิทธิ์นี้จะเปรียบเสมือนบัตรประชาชนดิจิตอลของเราที่ใช้บอกได้ว่าเราเป็นบุคคลที่อ้างจริงๆในระบบการค้าอิเล็กทรอนิกส์

ในปัจจุบันบริษัทหลักๆ ที่ออกเอกสารสิทธิ์ดิจิตอล (Digital Certificate) คือ บริษัท Verisign, Cyvertrust และ Nortel โดยในเอกสารสิทธิ์ดิจิตอลจะประกอบด้วยข้อมูลต่างๆดังนี้ ชื่อของผู้ถือเอกสารสิทธิ์ ชื่อของบริษัทที่ออกเอกสารสิทธิ์ คีย์สาธารณะของผู้ถือเอกสารสิทธิ์ วันหมดอายุของเอกสารสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(โดยทั่วไปจะกำหนดระยะเวลา 6 เดือนหรือหนึ่งปี) ระดับชั้นของเอกสารสิทธิ์ และเลขหมายของตัวเอกสารสิทธิ์ดิจิตอลนั้นเอง

เอกสารสิทธิ์ดิจิตอลแบ่งออกได้เป็นสี่ระดับชั้นตามระดับการตรวจสอบข้อมูลของเจ้าของเอกสารสิทธิ์ ระดับชั้นที่หนึ่งเป็นชั้นที่ออกเอกสารสิทธิ์ได้ง่ายที่สุดเนื่องจากมีการตรวจสอบน้อยที่สุด โดยจะตรวจสอบแค่ชื่อผู้ถือเอกสารสิทธิ์ และที่อยู่อิเล็กทรอนิกส์เมล์ (e-mail address) ว่าถูกต้องจริงเท่านั้น ในเอกสารสิทธิ์ชั้นที่สองจะตรวจสอบเลขประจำตัวประชาชน เลขประจำตัวของระบบสวัสดิการหรือประกันสังคม (social security number) และวันเดือนปีเกิด ชั้นที่สามจะมีการตรวจสอบเพิ่มเติมเกี่ยวกับประวัติการใช้เครดิตและการชำระเงิน สำหรับเอกสารสิทธิ์ชั้นที่ 4 นั้นยังไม่มีการออกมาเป็นมาตรฐานอย่างแน่ชัด แต่จะเป็นการตรวจสอบข้อมูลเพิ่มเติมเกี่ยวกับตำแหน่งงานในองค์กรด้วย

7.2 การขอเอกสารสิทธิ์

การขอหนังสือรับรองจากสถาบันผู้มีสิทธิออกหนังสือรับรอง สถาบันผู้ออกหนังสือเหล่านั้นจะตรวจสอบว่าท่าน ว่ามีชื่อ ที่อยู่และเอกสารอื่นๆ ครบถ้วนถูกต้องหรือไม่ ซึ่งเอกสารที่สถาบันแต่ละแห่งต้องการอาจแตกต่างกันไปในแต่ละที่ เช่นที่บริษัท Thawte กำหนดให้ผู้ขอหนังสือรับรองส่งเอกสารประกอบการขอ 2 อย่างคือ

1.หนังสือรับรองบริษัทหรือองค์กร

2.รายละเอียดยืนยันความเป็นเจ้าของชื่อโดเมนที่ต้องการจะนำหนังสือรับรองนี้ไปใช้ คือนำเอาต์พุตจากการใช้คำสั่ง whois แล้วตามด้วยชื่อโดเมนของท่าน เอาต์พุตนี้จะแสดงรายละเอียดเจ้าของชื่อโดเมน รวมทั้งองค์กรที่จดทะเบียนชื่อโดเมนของท่านด้วย

หากสนใจจะขอหนังสือรับรองจากสถาบันใด กรุณาตรวจสอบเสียก่อนว่า สถาบันนั้นๆ ต้องการเอกสารประกอบอะไรบ้างและเตรียมให้พร้อม (เอกสารทุกชิ้นต้องเป็นภาษาอังกฤษ)

สร้างคำร้องขอมีคีย์(Certificate Signing Request : CSR)

CSR จะต้องถูกสร้างจากเครื่องเว็บเซิร์ฟเวอร์ โดยซอฟต์แวร์เพื่อนำคีย์ที่ได้ไปใช้ในการขอหนังสือรับรอง

ตัวอย่างการสร้างคีย์ใน Windows NT 4.0

ถ้าท่านติดตั้ง Microsoft Internet Information Server(IIS) แล้วใน Toolbar เลือก Microsoft Internet Server แล้วเลือก Key Manager เมื่อโปรแกรมถูกเปิดขึ้นมา ให้เลือก “WWW” แล้วไปที่เมนูคีย์ เลือก “Create New Key”

ขอเอกสารสิทธิ์

เมื่อได้ไฟล์ CSR ตามขั้นที่ 3 แล้ว ต้องนำรายละเอียดที่เขียนไว้ในไฟล์ไปใส่ในแบบฟอร์มการขอมีหนังสือรับรอง ซึ่งท่านจะต้องเข้าไปยังโฮมเพจของสถาบันผู้ออกหนังสือ

ในการขอเอกสารสิทธิ์นั้น ผู้ขอจะเสียค่าธรรมเนียมให้แก่ผู้ออกเอกสารสิทธิ์เพื่อใช้เป็นค่าใช้จ่ายในการตรวจสอบประวัติของผู้ขอเอกสารสิทธิ์ ค่าธรรมเนียมจะแพงขึ้นตามระดับชั้นของเอกสารสิทธิ์ที่ขอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากทางผู้ออกเอกสารสิทธิ์ต้องมีการตรวจสอบข้อมูลเพิ่มเติมของผู้ขอมากขึ้น ระดับชั้นของเอกสารสิทธิ์ยิ่งสูงขึ้นยิ่งทำให้แน่ใจได้ว่าเอกสารสิทธิ์นั้นแสดงถึงบุคคลที่อ้างจริงๆ

นอกจากการออกเอกสารสิทธิ์แล้ว บริษัทผู้ออกเอกสารสิทธิ์ยังมีหน้าที่ดูแลเอกสารสิทธิ์ที่ออกไปแล้ว รวมถึงการแสดงรายชื่อของเอกสารสิทธิ์ที่ถูกถอดถอนจากการใช้งานอีกด้วย รายชื่อนี้จะเรียกว่ารายชื่อการถอดถอนเอกสารสิทธิ์ (Certificate Revocation List (CRL)) โดยจะทำให้ผู้ใช้สามารถตรวจสอบได้ว่าเอกสารสิทธิ์ใดใช้การไม่ได้แล้ว สำหรับข้อมูลใน CRL นั้นจะเป็นรายชื่อของเอกสารสิทธิ์ที่ถูกขโมย สูญหาย หรือเอกสารสิทธิ์ที่ผู้ขอเอกสารสิทธิ์ออกจากบริษัท โดยจะไม่รวมรายชื่อของเอกสารสิทธิ์ที่หมดอายุตามกำหนดไว้ด้วยเนื่องจากในตัวเอกสารสิทธิ์จะบอกวันหมดอายุในตัวเองอยู่แล้ว

นอกจากบริษัทที่เป็นผู้รับรองออกใบเอกสารสิทธิ์ หรือหน่วยงานของรัฐบาลแล้ว บริษัทอื่นๆ ก็สามารถเป็นผู้ออกเอกสารสิทธิ์ได้ โดยซื้อเครื่องเซิร์ฟเวอร์จากผู้ขายเครื่องที่ได้รับการรับรองจากผู้มีอำนาจในการออกเอกสารสิทธิ์ การมีเครื่องเซิร์ฟเวอร์เป็นของตนเองจะมีข้อดีในกรณีที่ทางบริษัทต้องการใช้เอกสารสิทธิ์ของพนักงานเป็นจำนวนมากๆ เนื่องจากยิ่งมีการใช้มากขึ้นเท่าใด การควบคุมการออก การค้นหาข้อมูลของเอกสารสิทธิ์ และการใช้เอกสารสิทธิ์ก็จะยิ่งมีความสำคัญต่อบริษัทมากขึ้นเท่านั้น ในปัจจุบันทางรัฐบาลสหรัฐกำลังติดตั้งระบบเพื่อเป็นที่เก็บคีย์สาธารณะของประชาชนสำหรับเป็นโครงสร้างของระบบเอกสารสิทธิ์ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

คริปโตเอพีไอ (CryptoAPI – Cryptographic Application Programming Interface)

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

8.1 การเรียกใช้คริปโตเอพีไอ

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

8.2 การติดต่อกับซีเอสพี

ในการติดต่อกับซีเอสพีนั้นจะเริ่มจากการที่โปรแกรมใดๆ นั้นได้เรียกใช้คริปโตเอพีไอโดยการผ่านคำสั่ง CryptAcquireContext เพื่อที่จะใช้จัดการกับซีเอสพีใด ๆ โดยเมื่อเรียกใช้คำสั่งนี้แล้วจะเป็นการบ่งถึง ซีเอสพี ที่กำหนดไว้ในพารามิเตอร์ที่ส่งไป ซึ่งสิ่งที่ส่งไปไม่ได้มีเพียงแค่ชื่อของซีเอสพีเท่านั้นแต่จะรวมถึงส่วนที่ใช้เก็บคีย์ที่อยู่ภายในซีเอสพีนั้นด้วย

8.3 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ***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 8.1 ชื่อที่ใช้บ่งบอกว่าเป็น ซีเอสพี ตัวใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

8.5 การติดต่อกับซีเอสพี

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

8.6 การใช้คริปโตเอพีไอในการทำงานกับเอกสารสิทธิ์

คำศัพท์ที่เกี่ยวข้อง

CRL – Certificate Revocation List คือ รายชื่อของเอกสารสิทธิ์ที่ถูกนำกลับมาใช้ใหม่ เช่น ในกรณีที่เอกสารสิทธิ์นั้น เจ้าของอาจจะไม่ต้องการใช้อีกต่อไปก็จะสามารถนำกลับไปให้ผู้อื่นใช้ต่อไปได้

CTL – Certificate Trust List คือ รายชื่อของเอกสารสิทธิ์ที่สามารถเชื่อถือได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
โอแอลอี (OLE)

9.1 โอแอลอี

โปรแกรมมีความจำเป็นจะต้องจัดการกับวัตถุ (object) ต่างๆ ภายในโปรแกรมที่เป็นโอแอลอีไคลเอนต์ (OLE Client) ดังนั้นเราจึงจำเป็นจะต้องเขียนโปรแกรมให้อยู่ในลักษณะของโอแอลอีเซิร์ฟเวอร์ (OLE Server) เพื่อให้โอแอลอีไคลเอนต์สามารถนำวัตถุของลายมือชื่อดิจิตอลไปเพิ่มไว้ในเอกสารได้ ส่วนการสั่งงานกับโอแอลอีไคลเอนต์จะต้องใช้คุณลักษณะของโอแอลอีอย่างหนึ่งที่เรียกว่าโอแอลอีออโตเมชัน เพื่อที่จะสามารถสั่งงานให้กับโอแอลอีไคลเอนต์ได้

9.2 โอแอลอีออโตเมชั่น (OLE Automation)

โอแอลอีออโตเมชันเป็นวิธีการในการจัดการกับวัตถุของโปรแกรม (Application) ที่สนใจ โดยสามารถควบคุมได้จากนอกโปรแกรมนั้น ๆ ซึ่งในการที่จะสามารถใช้งานได้นั้น โปรแกรมที่เขียนขึ้นมาจำเป็นที่จะต้องติดต่อกับโปรแกรมที่สนใจโดยการใช้คอมอินเตอร์เฟส (COM interface) ซึ่งการใช้อินเตอร์เฟสก็จะประกอบไปด้วยพรอพเพอร์ตี (Property) และเมดทอดต่าง ๆ เราสามารถเรียกโปรแกรมที่เขียนขึ้นมาเพื่อใช้จัดการกับโปรแกรมอื่นว่าออโตเมชันไคลเอนท์ (Automation Client) เพราะเราได้เรียกใช้เมดทอดจากโปรแกรมอื่น และในแนวคิดเดียวกันเราสามารถเรียกโปรแกรมที่เราไปติดต่อด้วยว่า ออโตเมชันเซิร์ฟเวอร์ (Automation Server) เพราะสามารถให้บริการต่าง ๆ กับเรา

อินเตอร์เฟสต่าง ๆ จะถูกประกาศไว้ในรูปแบบที่เรียกว่า ODL (Object Description Language) ซึ่ง ไคลเอนต์สามารถเรียกใช้ได้จากไฟล์ชนิดของไลบรารี (Type Library) โปรแกรมต่าง ๆ ที่สามารถเรียกใช้ชนิดของไลบรารีได้จะทำการสร้างซอร์สขึ้นมาเพื่อใช้กับออโตเมชันไคลเอนท์

9.2.1 การสร้างโปรแกรมให้ทำงานกับโอแอลอีออโตเมชั่นโดยใช้เอ็มเอฟซี (MFC) และชนิดของไลบรารี (Type Library)

1. ทำการนำชนิดของไลบรารีเข้า (Import Type Libray) จากคลาสวิซาร์ด (Class Wizard) โดยการเลือกหัวข้อออโตเมชันภายในคลาสวิซาร์ด
2. ทำการเพิ่มคลาส (Add Class) จากชนิดของไลบรารีจากนั้นเลือกไฟล์ไลบรารีของโปรแกรมที่ต้องการจะติดต่อด้วย ในที่นี้จะเลือกไฟล์ชนิดของไลบรารีของโปรแกรมไมโครซอฟท์เวิร์ด (Microsoft Word) โดยต้องเลือกไฟล์ C:\Program Files\Microsoft Office\Office\Msword8.olb
3. จากนั้นโปรแกรมจะทำการสร้างซอร์สโค้ด (Source Code) ของคลาสที่เราเลือกให้สร้าง ผลที่ได้ออกมาจะเป็นไฟล์ .cpp และ .h ของออโตเมชันเซิร์ฟเวอร์
4. ทำการอินคลูด (include) ไฟล์เฮดเดอร์ของออโตเมชันเซิร์ฟเวอร์
5. เริ่มการติดต่อกับออโตเมชันเซิร์ฟเวอร์ โดยการสั่งคำสั่ง AfxOleInit() เพื่อเป็นการจัดตั้งค่าเริ่มต้นต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากนั้นจะต้องใส่คำสั่ง AfxEnableControlContainer(); เพื่อที่จะทำให้โปรแกรมสามารถเข้าควบคุมออโตเมชันได้
7. สร้างวัตถุของโปรแกรมที่เป็นออโตเมชันเซิร์ฟเวอร์ โดยการประกาศ _Application app;
8. ทำการกำหนดโปรแกรมที่เราต้องการให้เป็นออโตเมชันเซิร์ฟเวอร์ให้กับวัตถุของออโตเมชันเซิร์ฟเวอร์ โดยการใช้คำสั่ง GetActiveObject();
9. สามารถเริ่มการติดต่อกับโปรแกรมออโตเมชันเซิร์ฟเวอร์ได้

9.3 โอแอลอีเซิร์ฟเวอร์

โปรแกรมแบบโอแอลอีเซิร์ฟเวอร์เป็นโปรแกรมที่ทำงานเกี่ยวกับวัตถุต่างๆ ซึ่งวัตถุเหล่านี้จำเป็นจะต้องมีคลาสที่ใช้ในการสร้างวัตถุโดยโปรแกรมทั้งหมดที่ทำงานบนวินโดว์ (window) จะรู้จักคลาสเหล่านี้ โดยการไปดูที่รีจีสทรีในส่วนของคีย์ HKEY_CLASSES_ROOT\CLSID ซึ่งเปรียบเสมือนกับตัวเลขที่ใช้บ่งบอกถึงคลาสทั้งหมดที่วินโดว์รู้จักโอแอลอีเซิร์ฟเวอร์จะต้องเป็นโปรแกรมที่ให้บริการในเรื่องของการเพิ่มวัตถุลงในเอกสารของโปรแกรมที่สนับสนุนการทำงานแบบโอแอลอีและต้องสามารถให้บริการในเรื่องของการลิงค์วัตถุได้ด้วย

9.3.1 ขั้นตอนการสร้างโปรแกรมแบบโอแอลอีเซิร์ฟเวอร์

1. จะให้โปรแกรมทำงานแบบ SDI(Single Data Interface)
2. คลาสที่ใช้สำหรับแสดงผล หน้าจอของโปรแกรมโอแอลอีเซิร์ฟเวอร์จะสืบทอดมาจากคลาส CListView เพื่อใช้ในการแสดงผลไอคอน (icon) ของวัตถุลายมือชื่อ (signature object) แต่ละตัว
3. ในเมดทอด InitInstance() จะต้องเรียกใช้คำสั่ง AfxOleInit() เพื่อเริ่มต้นการทำงานของโอแอลอี
4. ทำการเพิ่มคลาสไอดี (Class ID) เพื่อเป็นการบอกระบบให้รู้ว่ามีคลาสใหม่ที่เป็นของโอแอลอีเซิร์ฟเวอร์
5. จะต้องทำการแก้ไขคลาสวิว (class view) ที่ใช้ในการแสดงผล ในกรณีที่มีการแทรกวัตถุของลายมือชื่อลงในเอกสารใดๆ เพื่อใช้แสดงผลของวัตถุนั้น

9.3.2 คลาสต่าง ๆ ที่เกิดจากการสร้างโปรแกรมแบบโอแอลอีเซิร์ฟเวอร์

CmainFrame เป็นคลาสที่จะใช้เก็บหน้าต่างหลักของโปรแกรมในขณะที่โปรแกรมทำงานแยกต่างหากจากโอแอลอีไคลเอนต์

CserverApp เป็นคลาสที่ใช้สำหรับโปรแกรมหลัก เพื่อเป็นการกำหนดการทำงานของโปรแกรมหลัก

CserverDoc เป็นคลาสที่ใช้สำหรับเก็บข้อมูลของโปรแกรมหลักเพื่อนำไปบันทึกเป็นไฟล์ได้

CserverItem เป็นคลาส ที่ใช้เก็บวัตถุของไอเท็ม (Item) ต่าง ๆ ที่ถูกใช้โดยโอแอลอีไคลเอนต์

CserverView เป็นคลาสที่ใช้เก็บส่วนของการแสดงผลของหน้าจอหลักของโปรแกรม

CinplaceFrame เป็นคลาสที่ใช้จัดการกับวิชวลอีดิตติง (Visual Editing) ที่เกิดจากโอแอลอีไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
การออกแบบโปรแกรม

10.1 คุณลักษณะของโปรแกรม

- เป็นโปรแกรมที่สามารถสร้างลายมือชื่อดิจิตอลได้ โดยจะต้องสร้างลายมือชื่อดิจิตอลไว้ในเอกสารของโปรแกรมไมโครซอฟท์เวิร์ด
- ตัวโปรแกรมสามารถเพิ่มลายมือชื่อดิจิตอลของผู้ใช้ ใหม่ และสามารถลบลายมือชื่อดิจิตอลที่ไม่ต้องการใช้ออกไปได้
- ในการเพิ่มลายมือชื่อดิจิตอลเข้าไปจะทำให้มีการสร้างวัตถุของลายมือชื่อดิจิตอล เพิ่มเข้าไปอยู่ในไฟล์นั้น ๆ
- โปรแกรมจะสามารถตรวจสอบลายมือชื่อดิจิตอลของไฟล์ได้

10.2 การออกแบบโปรแกรมในส่วนติดต่อกับผู้ใช้

- โปรแกรมจะใช้การติดต่อกับผู้ใช้ในลักษณะกราฟฟิก โดยการใช้เมาส์เลือกที่เมนู หรือการกดปุ่มที่คีย์บอร์ดเป็นการสั่งงานโปรแกรม
- ในหน้าต่างของโปรแกรมหลักจะใช้แสดงไอคอนของบุคคลที่สามารถจะเลือกลงลายมือชื่อได้ ซึ่งผู้ใช้ได้เคยสร้างไว้ในเครื่องก่อนหน้านี้แล้ว ในที่นี้จะขอเรียกไอคอนนี้ว่า ไอคอนของผู้ใช้ เพื่อความสะดวก
- ผู้ใช้สามารถจะเลือกสร้าง และลบไอคอนของผู้ใช้ที่เคยสร้างไว้ได้จากเมนู Signature ซึ่งการจัดการต่าง ๆ กับไอคอนผู้ใช้จะมีผลต่อข้อมูลต่างของผู้ใช้ผู้นั้น
- ที่ไอคอนของผู้ใช้ จะสามารถสั่งการด้วยคีย์บอร์ดได้คือ การสั่งลบข้อมูลของผู้ใช้ผู้นั้นโดยการกดปุ่ม Delete บนคีย์บอร์ด

10.3 การออกแบบโปรแกรมในส่วนของการสร้างผู้ใช้ใหม่

- โปรแกรมต้องการข้อมูลของผู้ใช้ คือชื่อ และ นามสกุลของผู้ใช้ และถ้าหากผู้ใช้ต้องการรูปภาพที่แสดงถึง ลายมือชื่อของตนเอง ก็สามารถเพิ่มเข้าไปได้โดยการกดปุ่ม Browse…
- ผู้ใช้สามารถเลือกระดับความปลอดภัยของการใช้คีย์ส่วนตัวได้ โดยการกำหนดค่าในส่วนของ Set Security Level
- หลังจากสร้างผู้ใช้ใหม่แล้ว ผู้ใช้จะต้องเห็นทันทีว่าไอคอนของผู้ใช้ได้ถูกสร้างขึ้นมาใหม่แล้ว พร้อมกับมีชื่อผู้ใช้แสดงให้เห็น
- โปรแกรมจะทำการสร้างคู่ของพับลิกและไพรเวทคีย์ขึ้นมา เก็บไว้ในภายในรีจิสทรีของระบบ และผู้ใช้จะใช้คีย์เหล่านี้ในการลงลายมือชื่อ
- ถ้าผู้ใช้ได้เลือกไฟล์รูปภาพในการแสดงลายมือชื่อของตนเอง โปรแกรมจะเก็บชื่อของไฟล์พร้อมทั้งสร้างลายมือชื่อดิจิตอลให้กับข้อมูลของไฟล์รูปภาพด้วย เพื่อความปลอดภัยของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลรูปภาพของผู้ใช้ ในกรณีที่ข้อมูลรูปภาพมีการเปลี่ยนแปลง ก็จะไม่สามารถใช้ไพรเวทคีย์ในการลงลายมือชื่อได้

- โปรแกรมจะทำการสร้างส่วนที่ใช้เก็บข้อมูลของผู้ใช้แต่ละบุคคลให้ ซึ่งจะเก็บไว้ในรีจิสทรีหลักของโปรแกรมในส่วนของ HKEY_LOCAL_MACHINE\Software\IsagSign

10.4 การออกแบบโปรแกรมในส่วนของการลบชื่อผู้ใช้ออก

- ผู้ใช้ต้องสามารถลบชื่อผู้ใช้ออกได้ โดยจะต้องเป็นเจ้าของรายชื่อผู้ใช้นั้น ๆ พิสูจน์ได้จากการให้ใส่รหัสผ่าน ที่เป็นรหัสผ่านเดียวกันกับในขั้นตอนการสร้างไพรเวทคีย์ ถ้าไม่ถูกต้องโปรแกรมจะไม่อนุญาตให้ลบรายชื่อนั้นได้
- การสั่งให้ลบรายชื่อผู้ใช้นั้นออกนั้น ทำได้โดยการคลิ๊กที่ไอคอนของผู้ใช้นั้น ๆ แล้วเลือกเมนู Signature / Remove Signature หรือทำโดยการเลือกเมนู Edit / Cut หรือวิธีสุดท้ายโดยการกดปุ่ม Delete บนคีย์บอร์ด
- โปรแกรมสามารถสั่งลบรายชื่อผู้ใช้ออกหลายรายพร้อมกันได้ ด้วยการคลิ๊กเลือกรายชื่อผู้ใช้ที่ต้องการลบทั้งหมด โดยการกดปุ่ม Ctrl หรือปุ่ม Shift บนคีย์บอร์ดร่วมด้วย และทำการลบโดยวิธีดังกล่าวมาแล้วข้างต้น
- โปรแกรมจะทำการลบส่วนของคีย์คอนเทนเนอร์ของผู้ใช้นั้น ๆ ออกไปถ้าใส่รหัสผ่านได้ถูกต้อง แต่ถ้าหากผู้ใช้นั้นไม่ได้กำหนดรหัสผ่านไว้ โปรแกรมจะทำการลบทันที
- โปรแกรมต้องลบค่าของข้อมูลผู้ใช้แต่ละบุคคลที่เก็บไว้ในรีจิสทรีให้หมด

10.5 การออกแบบโปรแกรมในส่วนของการทำงานในไมโครซอฟท์เวิร์ด

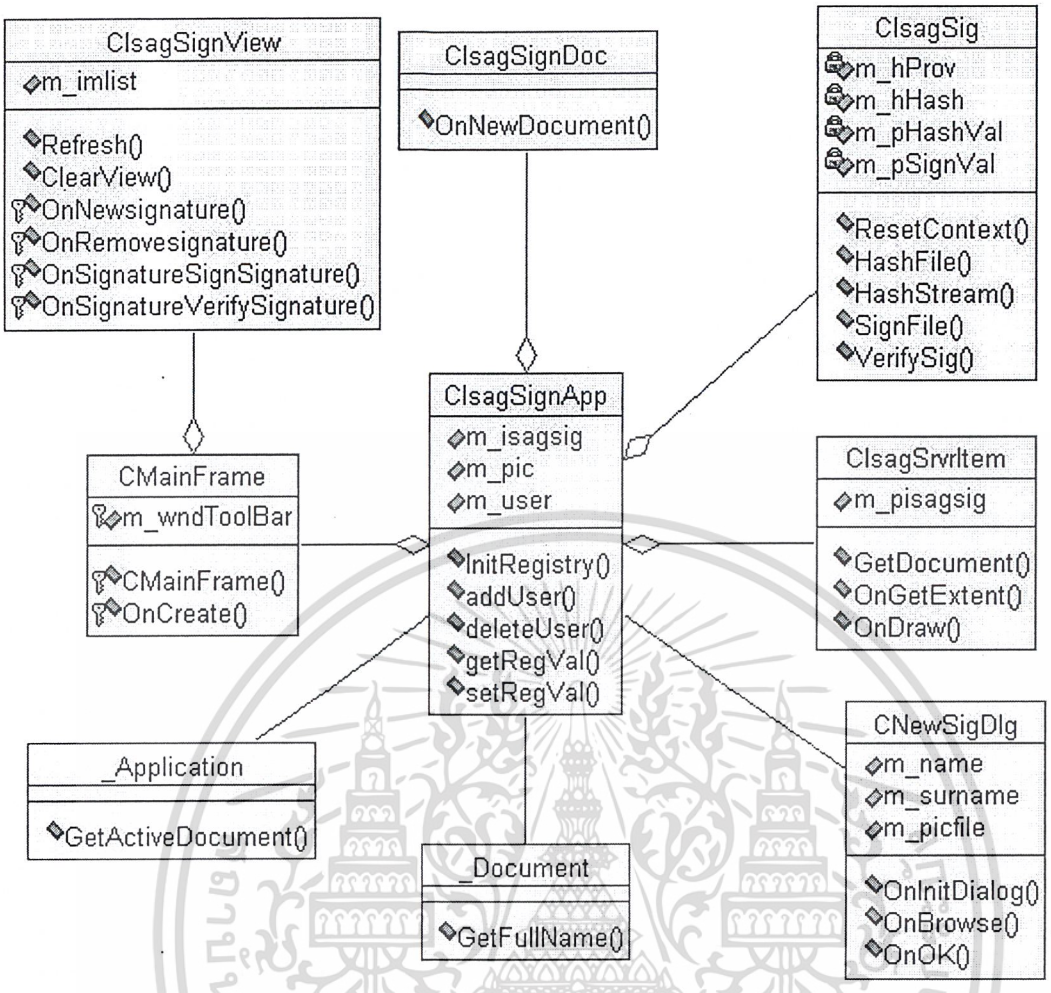
- โปรแกรมจะต้องสร้างเมนูของโปรแกรมเองขึ้น ในไมโครซอฟท์เวิร์ด โดยเป็นเมนูที่ชื่อ IsagSign
- ภายในเมนูของ IsagSign จะมีเมนูของโปรแกรมคือ Sign Signature และ Verify Signature
- เมื่อเลือกเมนู Sign Signature โปรแกรมจะทำการสร้างออปเจกต์ของลายมือชื่อดิจิตอลมาใส่ไว้ในเอกสารนั้น ๆ
- ในขณะการลงลายมือชื่อดิจิตอล โปรแกรมจะทำการอ่านข้อมูลในหน้าเอกสารนั้น ๆ ซึ่งข้อมูลที่อ่านได้จะเป็นข้อมูลปัจจุบันที่มีอยู่ในเอกสารนั้น ๆ โดยจะทำการติดต่อกับอินเทอร์เฟสของโปรแกรมไมโครซอฟท์เวิร์ดที่ชื่อว่า IDataObject และอ่านข้อมูลออกมาทำการหาลายมือชื่อดิจิตอลออกมา
- การลงลายมือชื่อดิจิตอลของ IsagSign จะมีผลต่อข้อมูลที่เป็นตัวอักษรเท่านั้น ข้อมูลรูปภาพและข้อมูลอื่น ๆ จะไม่ถูกรวมไปในการลงลายมือชื่อด้วย
- ผู้ใช้จำเป็นที่จะต้องใส่รหัสผ่านของการใช้คีย์ส่วนตัวให้ถูกต้อง เพื่อทำการลงลายมือชื่อดิจิตอล ไม่เช่นนั้นโปรแกรมจะไม่ทำการลงลายมือชื่อให้
- ถ้าหากผู้ใช้ผู้นั้นมีรูปภาพที่ใช้ในการแสดงลายมือชื่อ โปรแกรมจะทำการตรวจสอบลายมือชื่อของรูปภาพเอง ถ้าถูกต้องจะทำให้การลงลายมือชื่อสำเร็จ แต่ถ้าหากไม่ถูกต้องการลงลายมือชื่อก็จะไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าการลงลายมือชื่อสำเร็จ โปรแกรมจะทำการอ่านข้อมูลรูปภาพที่แสดงถึง รูปลายมือชื่อของผู้ใช้ผู้นั้นออกมาและทำการแสดงรูปภาพนั้นออกในหน้าของเอกสาร
- เมื่อเวิร์ดมีการบันทึกข้อมูลลงในไฟล์ จะทำให้โปรแกรม IsagSign เพิ่มข้อมูลเพิ่มเติมลงไปในเอกสารเวิร์ด โดยการเพิ่มสตอเรจ (Storage) ของลายมือชื่อดิจิตอล เข้าไปกับเอกสารเวิร์ดนั้น ๆ โดยสตอเรจที่เพิ่มเข้าไปจะประกอบไปด้วยสตรีม (Stream) ของข้อมูลลายมือชื่อดิจิตอล ซึ่งมีทั้งหมด 6 ส่วน ดังนี้
    1. รายชื่อผู้ลงลายมือชื่อดิจิตอล
    2. วิธีการในการหาแมซเซจไดเจส
    3. ขนาดของลายมือชื่อดิจิตอล เป็นจำนวนไบต์
    4. ข้อมูลของลายมือชื่อดิจิตอล
    5. ขนาดของพับลิกคีย์บล๊อบ (Public Key Blob) เป็นจำนวนไบต์
    6. พับลิกคีย์ที่อยู่ในรูปพับลิกคีย์บล๊อบ
- ในการเปิดใช้ออปเจกต์ของ IsagSign นั้นทำได้โดยการดับเบิ้ลคลิ้กที่รูปของออปเจกต์ในเอกสารเวิร์ด
- เมื่อออปเจกต์ถูกเปิดขึ้นมา โปรแกรมจะทำการอ่านข้อมูลจากสตอเรจที่อยู่ภายในเอกสารเวิร์ดนั้น ๆออกมา โดยอ่านสตรีมออกมา 6 สตรีมเหมือนกับในส่วนของการเก็บข้อมูลลงไฟล์ ถ้าหากไม่มีข้อมูลใด ๆ อยู่แสดงว่าไม่เคยลงลายมือชื่อมาก่อน
- ในการตรวจสอบลายมือชื่อดิจิตอลนั้น โปรแกรมจะทำการตรวจสอบว่าข้อมูลได้มีการลงลายมือชื่อไว้หรือไม่ ถ้ายังไม่มีการลงลายมือชื่อ โปรแกรมจะไม่ทำการตรวสอบให้
- การตรวจสอบลายมือชื่อ โปรแกรมจะทำการอ่านข้อมูลเฉพาะส่วนที่เป็นตัวอักษรออกมาทางอินเทอร์เฟส IDataObject และทำการหาค่าแฮชแวลลูออกมาเปรียบเทียบกับ ลายมือชื่อดิจิตอลที่ได้มีการลงไว้แล้ว

โปรแกรมจะแบ่งส่วนของโปรแกรมออกเป็นออปเจกต์ต่าง ๆ โดยแต่ละออปเจกต์จะมีความสัมพันธ์กันอยู่ โดยได้แสดงไว้ในรูปของคลาสไดอะแกรมดังรูปที่ 10.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.1 ส่วนประกอบต่าง ๆ ของโปรแกรม จะประกอบไปด้วยคลาสหลาย ๆ คลาส

โดยแต่ละคลาสมีการทำงานหลัก ๆ ดังนี้

- CisagSignApp เป็นคลาสหลักของโปรแกรม โดยจะมีหน้าที่ในการจัดการเกี่ยวกับการเริ่มต้นการทำงานของโปรแกรมคือ การกำหนดค่าต่าง ๆ ที่จำเป็นลงใน รีจิสทรีหลักของโปรแกรม (HKEY_LOCAL_MACHINE\Software\IsagSign) , การเพิ่มลายมือชื่อดิจิตอลใหม่และ การลบลายมือชื่อดิจิตอลที่ไม่ใช้แล้ว นอกจากนี้แล้ว คลาสนี้ยังรับผิดชอบถึงการสร้างคีย์ส่วนตัว และคีย์สาธารณะให้กับผู้ใช้ด้วย ซึ่งจะมีตัวแปรที่สำคัญดังนี้
    - m_isagsig เก็บออบเจ็กต์ที่ใช้จัดการเกี่ยวกับเรื่องลายมือชื่อดิจิตอล
    - m_pic เก็บรูปภาพที่ใช้แสดงลายมือชื่อ
    - m_user เก็บชื่อของผู้ที่ลงลายมือชื่อ

  มีฟังก์ชันหลัก ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- InitRegistry() สำหรับกำหนดค่าเริ่มต้นเก็บไว้ในรีจิสทรี
- addUser() เพิ่มรายชื่อผู้ใช้ใหม่เข้าไป
- deleteUser() ลบรายชื่อผู้ใช้ออกจากเครื่อง
- getRegVal() อ่านค่าจากรีจิสทรีออกมา
- setRegVal() เปลี่ยนค่าที่เก็บอยู่ในรีจิสทรี

- CisagSig เป็นคลาสที่จะเก็บส่วนของการทำงานต่าง ๆ เกี่ยวกับคีย์ส่วนตัว และคีย์สาธารณะ ซึ่งการทำงานดังกล่าว จะประกอบไปด้วยการสร้างไดเจส, การสร้างลายมือชื่อดิจิตอล และการตรวจสอบ โดยมีตัวแปรที่สำคัญดังนี้
  - m_hProv เก็บแฮนเดิลสำหรับซีเคียวริตี้โพรไวเดอร์
  - m_hHash เก็บแฮนเดิลสำหรับแฮช
  - m_pHashVal เก็บค่าตำแหน่งแฮชที่ได้ออกมาจากการคำนวณ
  - m_pSignVal เก็บตำแหน่งที่ชี้ไปยังลายมือชื่อดิจิตอล

  ฟังก์ชันสำหรับคลาสนี้ มีฟังก์ชันหลัก ๆ คือ
  - ResetContext() ฟังก์ชันสำหรับกำหนดค่าใหม่ให้กับแฮนเดิลของซีเคียวริตี้โพรไวเดอร์
  - HashFile() หาค่าของแฮชที่ได้จากการอ่านข้อมูลไฟล์
  - HashStream() หาค่าของแฮชที่ได้จากการอ่านข้อมูลในหน่วยความจำ
  - SignFile() ใช้สำหรับการลงลายมือชื่อให้กับข้อมูลของไฟล์
  - VerifySig() ใช้ตรวจสอบลายมือชื่อที่ได้ลงไว้
- CnewSigDlg เป็นคลาสของหน้าต่างการสร้างลายมือชื่อดิจิตอลใหม่ ซึ่งจะต้องรับชื่อและนามสกุลของผู้สร้างลายมือชื่อใหม่เข้าไปและนำไปสร้างลายมือชื่อขึ้นมา ประกอบด้วยตัวแปรต่าง ๆ คือ
  - m_name สำหรับเก็บชื่อของผู้ใช้ใหม่
  - m_surname สำหรับเก็บนามสกุลของผู้ใช้ใหม่
  - m_picfile สำหรับเก็บชื่อไฟล์ที่แสดงรูปภาพลายมือชื่อ (ถ้ามี)

  และฟังก์ชันต่าง ๆ มีดังนี้
  - OnInitDialog() ฟังก์ชันสำหรับเริ่มต้นการทำงานของหน้าต่างเพิ่มผู้ใช้ใหม่
  - OnBrowse() สำหรับผู้ใช้เมื่อกดปุ่ม Browse เพื่อเลือกชื่อไฟล์รูปภาพ
  - OnOK() สำหรับการยืนยันการสร้างชื่อผู้ใช้ใหม่
- CisagSignView เป็นคลาสที่ใช้สำหรับการแสดงผลของโปรแกรม ซึ่งจะแสดงไว้ในรูปไอคอนของลายมือชื่อดิจิตอลแต่ละคน นอกจากนี้แล้วคลาสนี้จะเป็นส่วนที่โปรแกรมมีไว้สำหรับให้ผู้ใช้ติดต่อกับโปรแกรมได้เพราะจะมีส่วนของการรอรับเหตุการณ์ที่เกิดจากผู้ใช้ ซึ่งเป็นเหตุการณ์ที่เกิดขึ้นกับคีย์บอร์ด และเมาส์เป็นต้น โดยคลาสนี้จะมีการเรียกใช้ฟังก์ชันต่าง ๆ เมื่อผู้ใช้ได้เลือกเมนู โดยมีตัวแปรที่สำคัญดังนี้
  - m_imlist สำหรับเก็บรูปภาพที่ใช้แสดงไอคอนของผู้ใช้แต่ละบุคคล

  และฟังก์ชันสำคัญดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Refresh() สำหรับแสดงผลหน้าจอใหม่ โดยจะตรวจว่ามีผู้ใช้กี่คน
- ClearView() สำหรับลบไอคอนของผู้ใช้อื่น ให้เหลือแต่ของผู้ลงลายมือชื่อ
- OnNewsignature() สำหรับสร้างรายชื่อผู้ใช้ใหม่เพิ่มไปในเครื่อง
- OnRemovesignature() สำหรับลบรายชื่อผู้ใช้ที่ไม่ต้องการออก
- OnSignatureSignsignature() สำหรับการลงลายมือชื่อใหม่
- OnSignatureVerifysignature() สำหรับการตรวจสอบลายมือชื่อที่ได้ลงไว้แล้ว

- CisagSignDoc คลาสเอกสารของโปรแกรม ซึ่งจะต้องถูกสร้างขึ้นเมื่อถูกนำไปแทรกไว้ในเอกสารของโปรแกรมอื่น โดยคลาสนี้จะรวมการทำงานทุกอย่างเกี่ยวกับการเก็บข้อมูลของลายมือชื่อดิจิตอลลงในเอกสารอื่น ๆ ซึ่งข้อมูลที่จะต้องเก็บลงไปประกอบไปด้วยข้อมูลสำคัญ 6 ส่วนคือ
  - ชื่อของผู้ลงลายมือชื่อ
  - วิธีการในการหาแมซเซจไดเจส
  - ความยาวของลายมือชื่อเป็นจำนวนไบต์
  - ลายมือชื่อที่ได้ลงไว้ในเอกสาร
  - ความยาวของคีย์สาธารณะเป็นจำนวนไบต์
  - คีย์สาธารณะของผู้ลงลายมือชื่อ

  ซึ่งจะมีฟังก์ชันที่สำคัญดังนี้คือ
  - OnNewDocument() สำหรับการเปิดเอกสารใหม่ของ IsagSign
- CmainFrame คลาสนี้เป็นคลาสหน้าต่างหลักของโปรแกรม สำหรับการสร้างแถบเครื่องมือ และเมนูต่าง ๆ ของโปรแกรมมีตัวแปรสำคัญคือ
  - m_wndToolBar สำหรับเก็บทูลบาร์ในการนำไปแสดงในหน้าจอโปรแกรมอื่น

  มีฟังก์ชันหลัก ๆ ดังนี้
  - CmainFrame() เป็นคอนสตรัคเตอร์สำหรับคลาส
  - OnCreate() สำหรับการสร้างหน้าต่างหลัก
- _Application คลาสนี้จะใช้สำหรับโปรแกรมที่เรียกใช้งานลายมือชื่อดิจิตอล โดยจะสามารถใช้คุณสมบัติต่าง ๆ ที่โปรแกรมนั้นจัดหาไว้ได้ โดยในโครงงานนี้ได้จัดให้คลาสนี้เป็นคลาสของโปรแกรมไมโครซอฟท์เวิร์ด 97
- _Document เป็นคลาสสำหรับจัดการกับเอกสารที่โปรแกรมอื่นที่เรียกใช้ลายมือชื่อดิจิตอล ซึ่งคลาสนี้ของโครงงานกำหนดให้เป็นคลาสสำหรับเอกสารที่สร้างจากโปรแกรมไมโครซอฟท์เวิร์ด 97
- CisagSrvrItem เป็นคลาสสำหรับวัตถุของลายมือชื่อดิจิตอลที่นำเข้าไปใส่ในเอกสารอื่น ซึ่งคลาสนี้จัดการในเรื่องของการวาดรูปร่างของวัตถุที่อยู่ในเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11

การทดลองและผลการทดลอง

11.1 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

11.2 ระบบที่ใช้ทดสอบ

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

11.3 การทดสอบการรันโปรแกรมในครั้งแรก

เมื่อมีการรันโปรแกรมในครั้งแรกนั้น โปรแกรมจะทำการสร้างคีย์คอนเทนเนอร์ชื่อ IsagSign ซึ่งจะใช้เป็นดีฟอลต์คีย์คอนเทนเนอร์ของโปรแกรม ดังนั้นโปรแกรมครั้งแรกจะสร้างคีย์ส่วนตัว และคีย์สาธารณะเก็บไว้ในคีย์คอนเทนเนอร์ IsagSign จึงทำให้โปรแกรมทำการคำนวณชั่วขณะหนึ่งซึ่งผู้ใช้จะต้องรอการทำงานในจุดนี้ให้เสร็จ จึงจะเริ่มการทำงานกับโปรแกรมได้ เมื่อการสร้างดีฟอลต์คีย์คอนเทนเนอร์เสร็จเรียบร้อยแล้ว โปรแกรมจะแสดงหน้าต่างดังรูปที่ 11.1

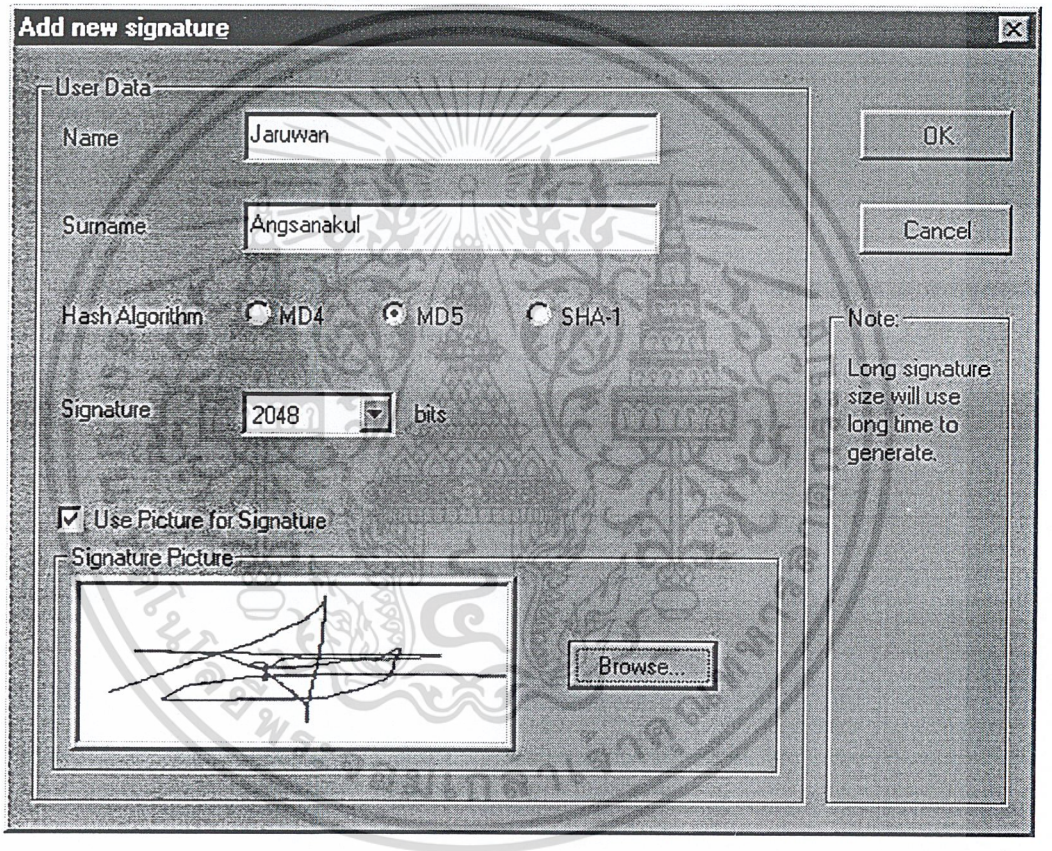
รูปที่ 11.1 แสดงรายชื่อผู้ใช้ที่มีอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 11.1 จะเห็นว่ามีไอคอนที่ชื่อ IsagSign ซึ่งเป็นไอคอนหลักของโปรแกรม เพื่อแสดงว่าขณะนี้ได้มีการสร้างคีย์คอนเทนเนอร์ชื่อ IsagSign ไว้เป็นที่เรียบร้อย ผู้ใช้จะสามารถนำคีย์ดังกล่าวนี้ไปใช้ในการลงลายมือชื่อดิจิตอลได้

11.4 การทดสอบการสร้างลายมือชื่อดิจิตอลใหม่

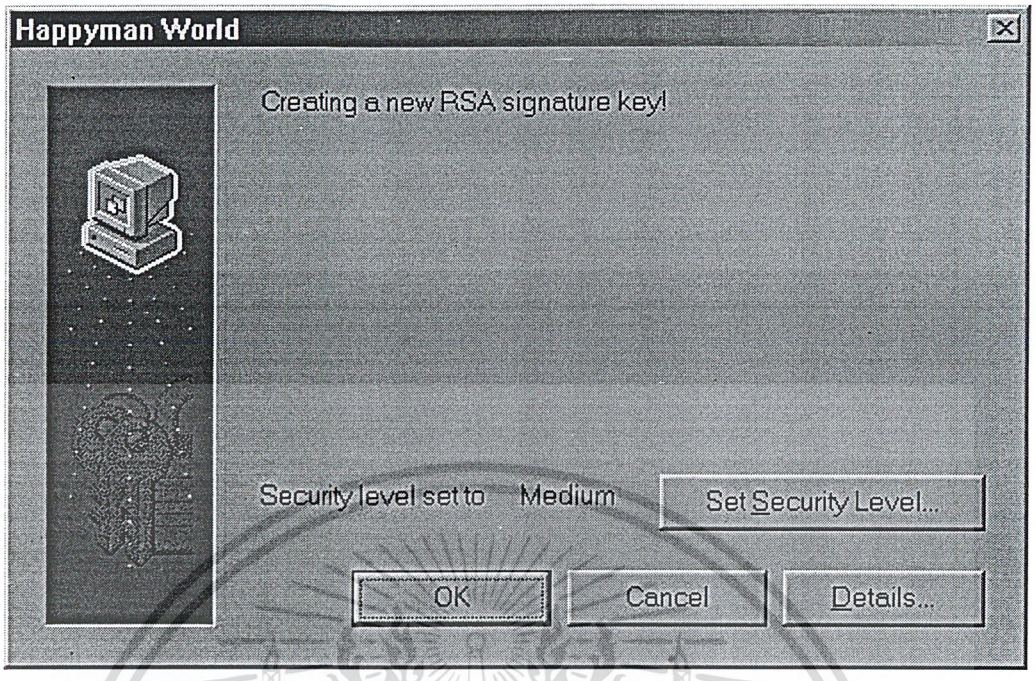
- ผู้ใช้จะต้องเลือกเมนู Signature / New Signature
- โปรแกรมจะขึ้นหน้าต่างของการสร้างลายมือชื่อดิจิตอลใหม่ดังรูปที่ 11.2 ให้ผู้ใช้ใส่ชื่อ และนามสกุลของผู้ใช้ลงไป และถ้าผู้ใช้คนใดมีไฟล์รูปภาพที่จะใช้แสดงเมื่อเกิดการลงลายมือชื่อดิจิตอล ก็ให้กดที่ปุ่ม Browse จากนั้นให้เลือกไฟล์รูปภาพ ซึ่งจะมีรูปภาพตัวอย่างแสดงในหน้าต่าง เมื่อผู้ใช้ใส่ข้อมูลครบแล้วให้กดปุ่ม OK



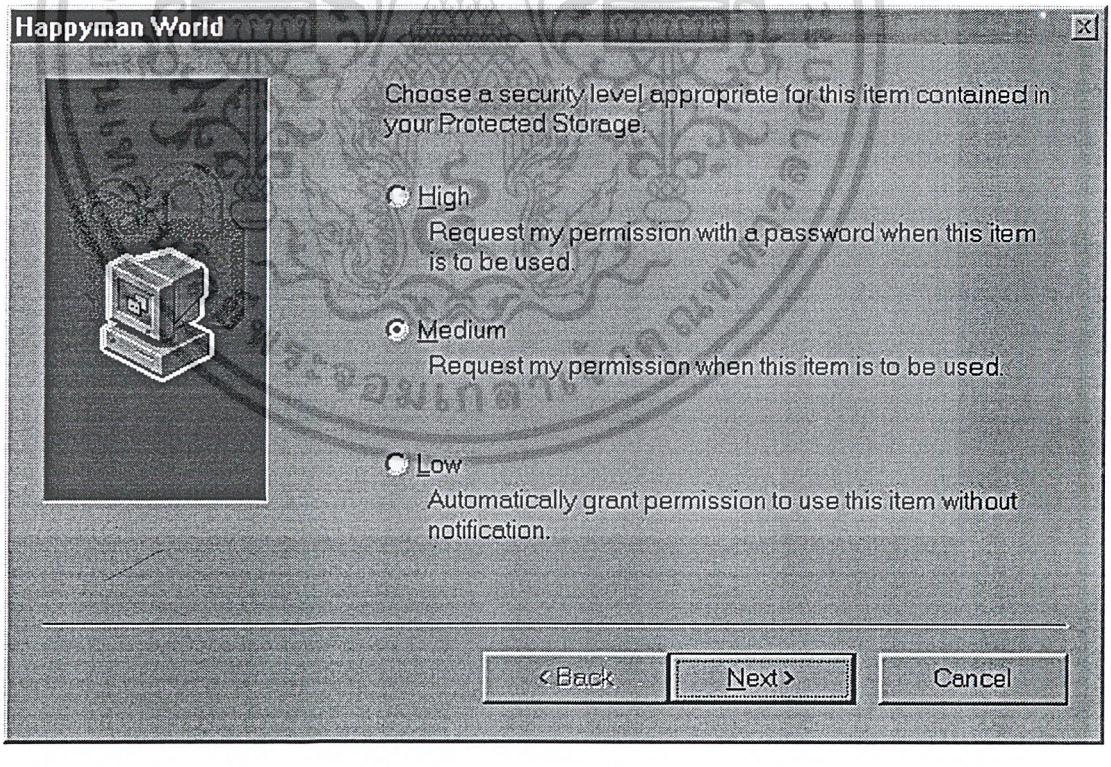
รูปที่ 11.2 หน้าจอเมื่อทำการเพิ่มลายมือชื่อ

- โปรแกรมจะขึ้นหน้าต่างเพื่อถามว่าต้องการความปลอดภัยจากการใช้คีย์ส่วนตัวระดับใด ดังรูปที่ 11.3 โดยการกดที่ปุ่ม Set Security Level จะได้หน้าจอของโปรแกรมดังรูปที่ 11.4 โดยมีให้เลือก 3 ระดับคือ ระดับต่ำ ผู้ใช้จะสามารถใช้คีย์ส่วนตัวได้โดยไม่มีการแจ้งเตือนใด ๆ ทั้งสิ้น, ระดับกลาง ผู้ใช้จะได้รับการแจ้งเตือนเมื่อมีการใช้คีย์ส่วนตัว และสุดท้ายระดับสูง ซึ่งจะต้องมีการใส่รหัสผ่านเพื่อกำหนดค่าไว้สำหรับเป็นความลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



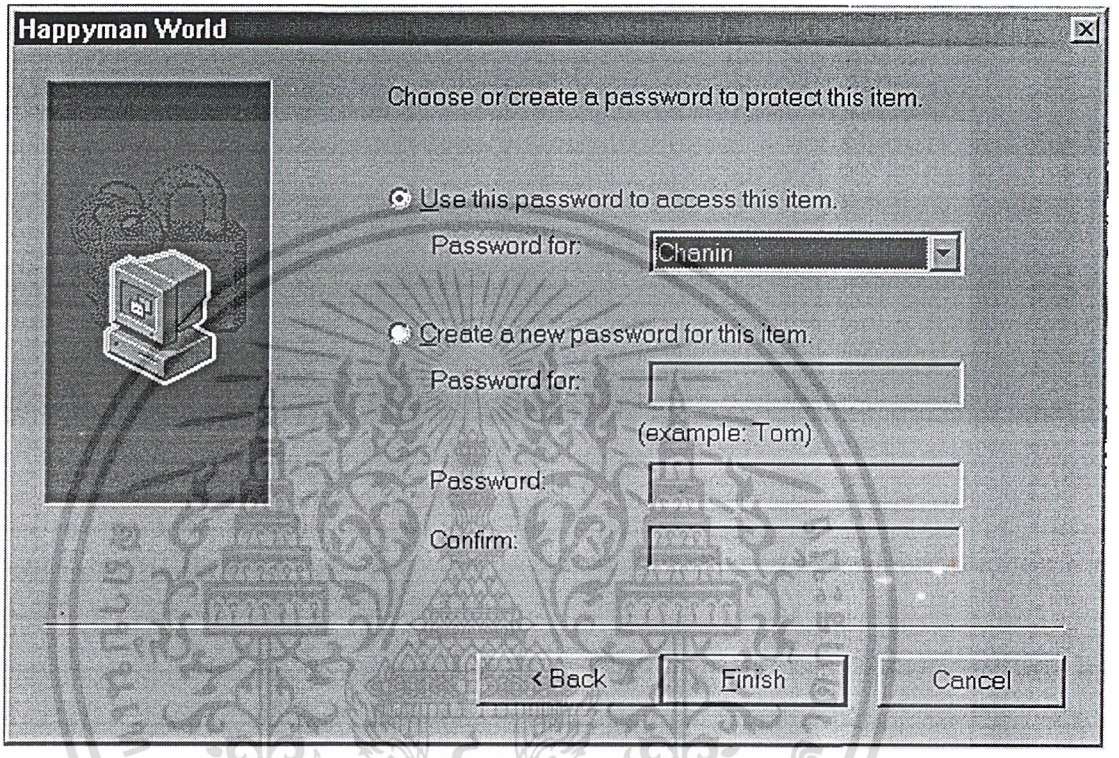
รูปที่ 11.3 หน้าจอเมื่อกดยืนยันการสร้าง



รูปที่ 11.4 หน้าจอเมื่อกดปุ่ม Set Security Level

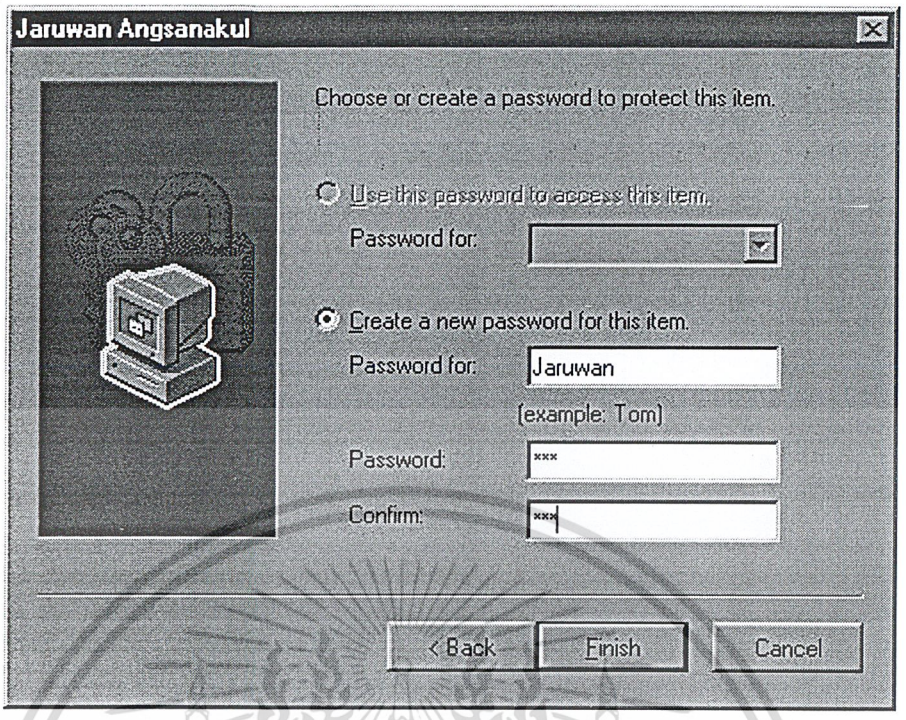
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าผู้ใช้เลือกความปลอดภัยระดับต่ำและกลาง จะสามารถจบขั้นตอนของการสร้างลายมือชื่อได้เลย แต่หากผู้ใช้เลือกระดับสูง จะต้องมีการกำหนดรหัสผ่านซึ่งการสร้างรหัสผ่านจะสามารถสร้างขึ้นมาใหม่ หรือจะใช้รหัสผ่านที่เคยสร้างไว้แล้วก็ได้ ซึ่งถ้าต้องการใช้รหัสผ่านที่มีอยู่แล้ว ดังรูปที่ 11.5 โปรแกรมจะให้เลือกชื่อของรหัสผ่านที่เคยสร้างมาแล้ว แต่ถ้าต้องการสร้างใหม่ผู้ใช้จะต้องสร้างชื่อรหัสผ่านขึ้นมาใหม่ และกำหนดรหัสผ่านให้ดังรูปที่ 11.6



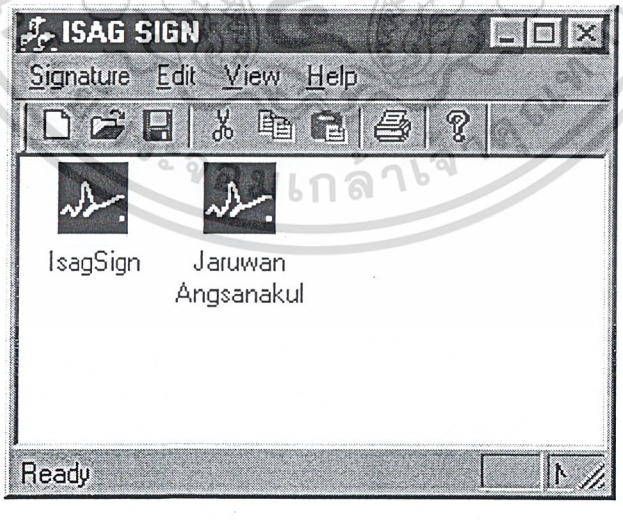
รูปที่ 11.5 หน้าจอเลือกว่ายืนยันใช้รหัสผ่านอันนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11.6 รูปแสดงการสร้างรหัสผ่านอันใหม่

- โปรแกรมจะให้ใส่รหัสผ่านอีกครั้งเพื่อเป็นการสร้างคีย์ส่วนตัว และคีย์สาธารณะ
- จะได้ลายมือชื่อดิจิตอลที่สร้างขึ้นมาตามรูปที่ 11.7



รูปที่ 11.7 ผลของการสร้างลายมือชื่ออันใหม่

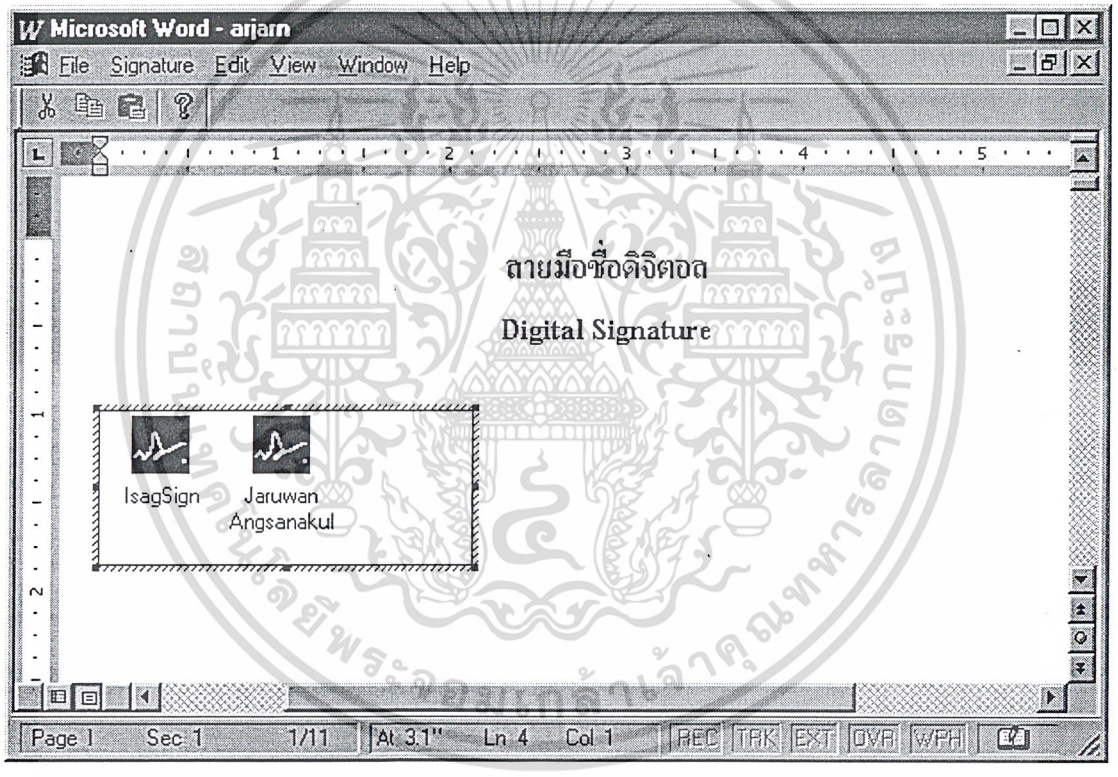
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11.5 การลบลายมือชื่อดิจิตอลที่ไม่ใช้

- เริ่มจากผู้ใช้จะต้องทำการเลือกแถบสีให้กับไอคอนของลายมือชื่อดิจิตอลที่ต้องการลบทิ้ง
- โปรแกรมจะทำการลบคีย์ส่วนตัว และคีย์สาธารณะของลายมือชื่อดิจิตอลนั้น ถ้าหากในขั้นตอนการสร้างลายมือชื่อดิจิตอล มีการกำหนดรหัสผ่านไว้ ในขั้นตอนนี้จะต้องมีการใส่รหัสผ่านนั้นด้วยเพื่อเป็นการยืนยัน

11.6 การลงลายมือชื่อดิจิตอล

- ในโปรแกรมที่ต้องการลงลายมือชื่อดิจิตอล ให้เลือกที่เมนู IsagSign จากนั้นเลือก Sign Signature
- โปรแกรมจะทำการหาว่าในเครื่องนั้น ๆ ได้มีการสร้างลายมือชื่อดิจิตอลใดไว้บ้าง และจะนำมาแสดงในรูปของไอคอนดังแสดงในรูปที่ 11.8

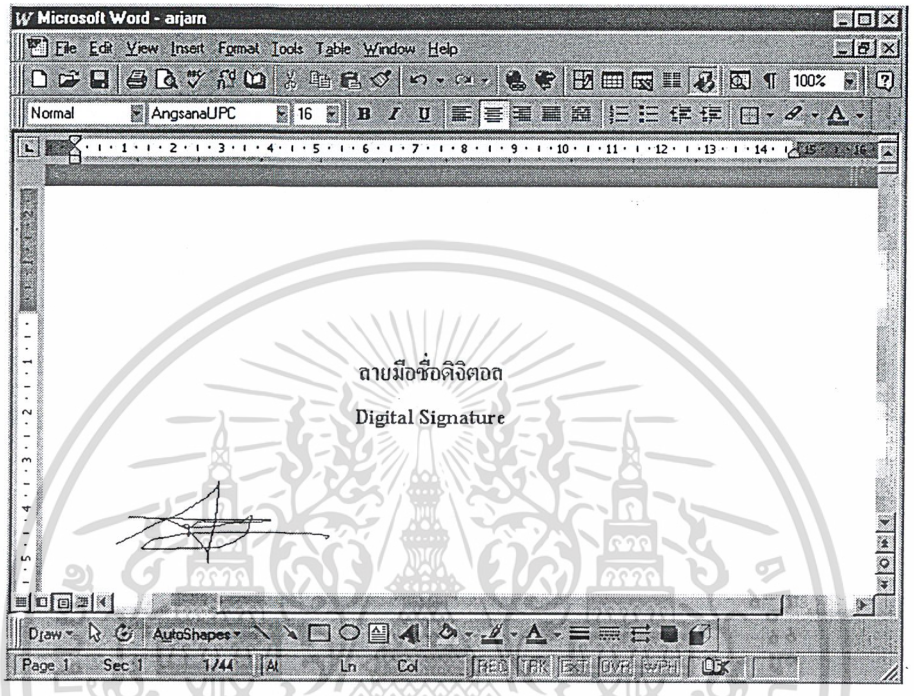


รูปที่ 11.8 หน้าจอโปรแกรมระหว่างการทำงานในโปรแกรมไมโครซอฟท์เวิร์ด

- ผู้ใช้จะต้องทำการเลือกลายมือชื่อดิจิตอลของตนเองเพื่อทำการลงลายมือชื่อดิจิตอล
- โปรแกรมจะนำคีย์ส่วนตัวของผู้ใช้ผู้นั้นออกมาทำการลงลายมือชื่อดิจิตอล ซึ่งในขั้นนี้อาจต้องมีการใส่ รหัสผ่านของการเข้าใช้คีย์ส่วนตัว ขึ้นอยู่กับขั้นตอนของการสร้างลายมือชื่อดิจิตอลนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมจะตรวจสอบว่าผู้ใช้นั้นมีการใส่รูปลายมือชื่อดิจิตอลไว้หรือไม่ ถ้ามีจะแสดงรูปของลายมือชื่อดิจิตอลนั้นออกมาดังแสดงในรูปที่ 11.9 แต่ถ้าไม่มีรูปจะแสดงไอคอนของโปรแกรมขึ้นมา

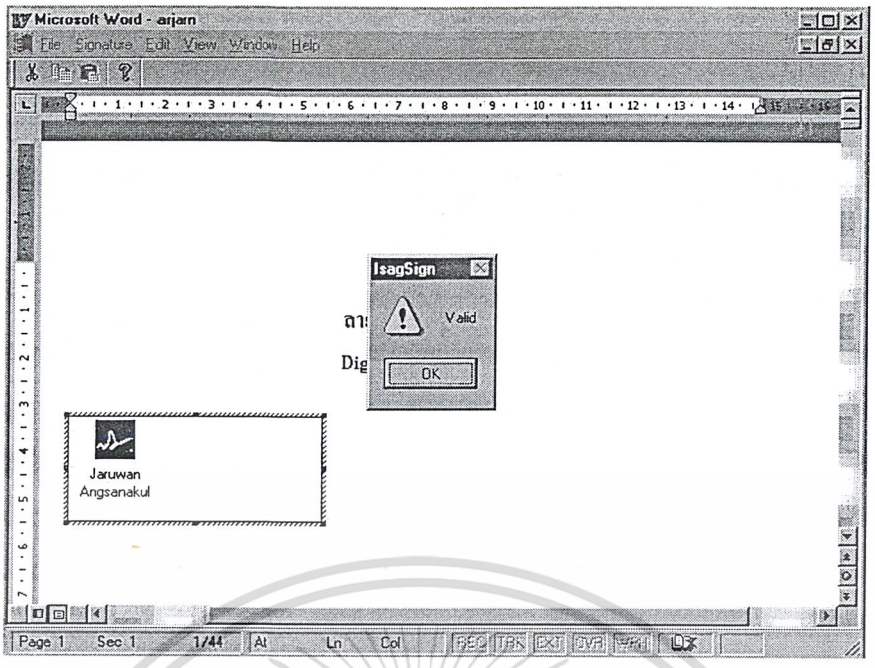


รูปที่ 11.9 ผลที่ได้จากการลงลายมือชื่อ

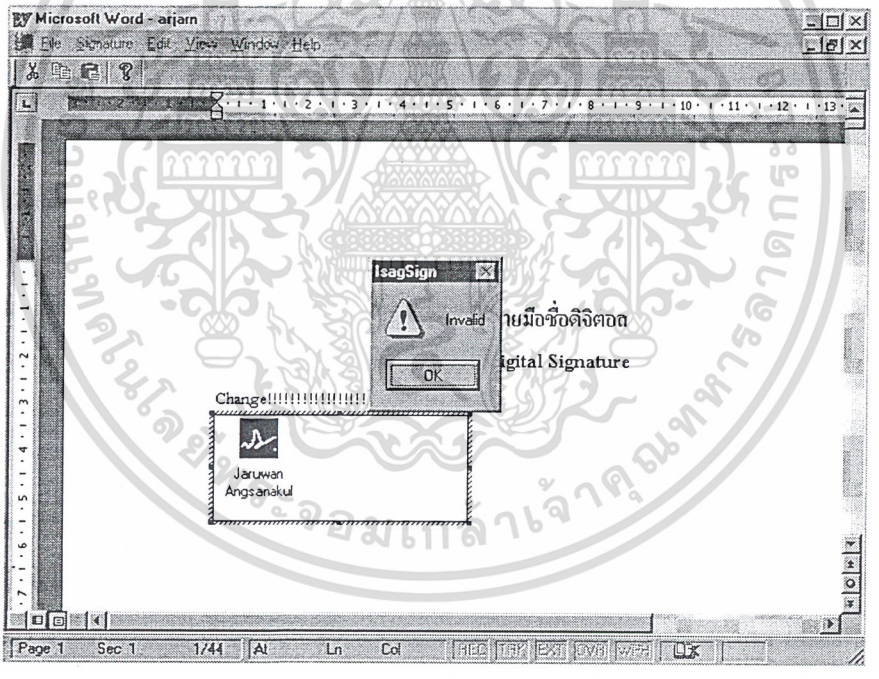
11.7 การตรวจสอบลายมือชื่อดิจิตอล

- ในเอกสารที่มีการลงลายมือชื่อดิจิตอลไว้แล้ว ให้เลือกที่เมนูของ IsagSign / Verify Signature
- โปรแกรมจะทำการตรวจสอบลายมือชื่อดิจิตอลที่ได้เคยลงไว้ กับข้อมูลปัจจุบันว่าเปลี่ยนแปลงไปหรือไม่ถ้าข้อมูลของเอกสารไม่มีการเปลี่ยนแปลง โปรแกรมจะแสดงข้อความ Signature Valid ดังรูปที่ 11.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11.10 แสดงการตรวจสอบลายมือชื่อดิจิตอลว่าถูกต้อง



รูปที่ 11.11 แสดงการตรวจสอบลายมือชื่อดิจิตอลและข้อมูลมีการแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 12
วิจารณ์และสรุป

12.1 บทวิจารณ์

จากโปรแกรมที่ได้พัฒนามาเพื่อใช้สำหรับการลงลายมือชื่อดิจิตอล ผลที่ได้ออกมาสามารถที่จะลงลายมือชื่อดิจิตอลได้ แต่ตัวโปรแกรมมีข้อจำกัดอยู่ ซึ่งโปรแกรมนี้จะสามารถตรวจได้เฉพาะข้อมูลที่เป็นตัวอักษรที่อยู่ภายในเอกสารของโปรแกรมไมโครซอฟท์เวิร์ด จากการทดสอบเมื่อลงลายมือชื่อดิจิตอลในเอกสารที่มีแต่ตัวอักษร หลังจากนั้นตรวจสอบลายมือชื่อดิจิตอล จะได้ผลลัพธ์ว่าการตรวจสอบลายมือชื่อดิจิตอลนั้นถูกต้อง หลังจากได้ทดลองแก้ไขข้อมูลที่อยู่ในเอกสาร จากนั้นตรวจสอบลายมือชื่อดิจิตอล จะได้ว่าการตรวจสอบลายมือชื่อดิจิตอลผิดพลาด ซึ่งเป็นผลลัพธ์ที่ออกมาตรงตามทฤษฎีของลายมือชื่อดิจิตอล แต่โปรแกรมนี้ก็ยังทำงานได้ไม่สมบูรณ์เนื่องจากเอกสารในไฟล์สามารถที่จะมีข้อมูลได้หลายชนิด ไม่เฉพาะเจาะจงอยู่ที่ข้อมูลชนิดตัวอักษร

12.2 แนวทางในการพัฒนาโปรแกรม

เนื่องจากโปรแกรมจะสามารถตรวจสอบข้อมูลได้เฉพาะข้อมูลที่เป็นตัวอักษร ดังนั้นแนวทางในการพัฒนาต่อ จะต้องทำให้โปรแกรมสามารถตรวจสอบข้อมูลได้ทุกชนิดที่อยู่ในไฟล์เอกสาร โปรแกรมที่จะพัฒนาต่อควรจะมีคุณสมบัติด้านความปลอดภัยสูงกว่าโปรแกรมปัจจุบัน ด้วยการยืนยันบุคคลที่เป็นเจ้าของลายมือชื่อดิจิตอลด้วยเอกสารสิทธิ์ เพราะผู้ที่มีเอกสารสิทธิ์ที่เชื่อถือได้จะต้องได้รับมาจากองค์กรพิสูจน์สิทธิ์ นอกจากนั้นแล้วโปรแกรมที่ควรจะพัฒนาต่อควรจะทำให้สามารถที่จะแทรกออปเจกต์ของลายมือชื่อดิจิตอล เข้าไปในเอกสารของโปรแกรมหลาย ๆ ตัวที่เป็นที่นิยม ไม่ยึดติดอยู่กับโปรแกรมไมโครซอฟท์เวิร์ด

12.3 บทสรุป

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
ความรู้พื้นฐานเกี่ยวกับ Microsoft Visual C++

Microsoft Visual C++ เป็นโปรแกรมที่สามารถใช้เป็นเครื่องมือ (Tool) ในการพัฒนาโปรแกรมประยุกต์บนระบบปฏิบัติการวินโดว์ (Microsoft Windows) ที่มีความสามารถในการจัดการและควบคุมการติดต่อหรือใช้งานทั้งด้านซอฟต์แวร์และทางด้านฮาร์ดแวร์สูง ซึ่งตัวโปรแกรมเองได้รับการพัฒนาความยืดหยุ่นและความมีประสิทธิภาพสูงนี้มาจากภาษา C++

โดยภาษา C++ มีโครงสร้างการทำงานเป็นแบบโอโอพี (OOP : Obfect Oriented Programming) ซึ่งเป็นวิธีการเขียนโปรแกรม โดยอาศัยแนวคิดของวัตถุ (Object) ชิ้นหนึ่ง ลักษณะการเขียนโปรแกรมประเภทนี้จะเป็นแบบโครงสร้าง (Structure) โดยการเขียนโปรแกรมแบบ OOP นี้มีความสามารถในการปกป้องข้อมูล โดยสามารถซ่อนรายละเอียด (encapsulation) และการสืบทอดคุณสมบัติ (inheritance) ของวัตถุตัวอื่นๆ ซึ่งทำให้การเขียนโปรแกรมมีความง่ายและรวดเร็วขึ้นมาก

- คลาส (Class) คือ การรวมคุณลักษณะและการใช้งานของวัตถุอย่างน้อยหนึ่งอย่างมาไว้ในกลุ่มเดียวกัน
- ออบเจ็กต์(Object) คือ วัตถุที่เป็นตัวแปรของคลาส เป็นรูปแบบของคลาสที่มีตัวตน ที่เราสามารถนำไปใช้งานได้
- การสืบทอดของคลาส (Inheritance) คือ การที่คลาสแต่ละคลาสสามารถเป็นคลาสแม่ (Super Class) ได้ โดยจะสืบทอดมาเป็นคลาสใหม่ได้ ซึ่งคลาสใหม่ที่สืบทอดมานี้จะยังคงมีคุณสมบัติเหมือนกับคลาสแม่ทุกประการและเราสามารถกำหนดคุณสมบัติใหม่เพิ่มเติมลงไปได้อีกเพื่อความเหมาะสมในการใช้งาน

โปรแกรม Microsoft Visual C++ ยังสนับสนุนการพัฒนาโปรแกรมประยุกต์อื่นๆในหลายด้านด้วยกัน ไม่ว่าจะเป็นโปรแกรมประยุกต์ที่ทำงานทั่วไป เช่น การคำนวณง่ายๆ การเขียนรูปต่างๆ เป็นต้น การเขียนโปรแกรมระบบจัดการฐานข้อมูล (DBMS : Database Management System) หรือแม้กระทั่งการเขียนโปรแกรมที่ทำงานด้วยระบบมัลติมีเดีย (Multimedia Application)

ขั้นตอนแรกในการสร้างโปรแกรมประยุกต์โดยใช้โปรแกรม Microsoft Visual C++ คือ การสร้างโปรเจกต์เวิร์กสเปซ (Project Workspace) ขึ้นมาใหม่ ซึ่งเป็นการกำหนดพื้นที่ในการเก็บโปรเจกต์ (Project) หรือเก็บโปรแกรมที่เราต้องการสร้าง และใช้ในการกำหนดตัวเลือกต่างๆของโปรแกรมที่เราต้องการสร้าง เช่น เก็บรูปภาพ การกำหนดลักษณะของโปรแกรมที่สร้าง หรือข้อกำหนดต่างๆเป็นต้น

การใช้งานโปรเจกต์ของ Microsoft Visual C++ มีลักษณะการใช้งานเหมือนกับโปรเจกต์ไฟล์ทั่วไป คือ เราสามารถเปิด-ปิด (Open-Close) เซฟ (Save) หรือลบ (Delete) ไฟล์ที่ใช้งานได้ สามารถทำการคอมไพล์ (Compile) รวมทั้งการดีบัก (Debug) โปรแกรมเพื่อหาจุดผิดของโปรแกรมได้

เราจะเห็นได้ว่าในการเขียนโปรแกรมประยุกต์บนระบบปฏิบัติการดอส (Dos) เราสามารถเขียนโปรแกรมใหญ่ๆได้โดยใช้ไฟล์เดียวในการเก็บข้อมูลทุกอย่าง แต่ในการเขียนโปรแกรมประยุกต์บนระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติการวินโดว์นั้นเรามักจะต้องใช้ส่วนประกอบต่างๆมากมาย เช่น รูปภาพ หรือเมนูบาร์ (Menu Bar) เป็นต้น และใช้ไฟล์ร่วมกันหลายๆไฟล์ เช่น ไฟล์ที่ทำหน้าที่ในการเปิดรูปภาพ หรือไฟล์ที่ทำการแบ่งส่วนของรูปภาพเป็นต้น เพื่อให้เป็นสัดส่วนและสามารถแก้ไขโปรแกรมนี้ได้ง่ายยิ่งขึ้น

โปรเจกต์ไฟล์ของ Microsoft Visual C++ เวอร์ชัน 6 ใช้นามสกุล dsw ซึ่งทำหน้าที่เป็นไฟล์ที่เก็บตัวเลือกต่างๆของโปรเจกต์ดังที่กล่าวไปแล้ว และเราสามารถโหลดโปรเจกต์ไฟล์ที่เขียนด้วย Microsoft Visual C++ เวอร์ชันที่ต่ำกว่านี้ได้ เช่น mak หรือ mdp เป็นต้น

ประโยชน์ของโปรเจกต์ไฟล์สามารถสรุปได้ดังนี้คือ

1. โปรเจกต์ไฟล์จะเก็บรายชื่อของไฟล์ที่เป็นซอร์สโค้ด (Source Code) โปรแกรมทั้งหมดที่ใช้ร่วมกันในโปรเจกต์ เช่น ซอร์สโปรแกรมนามสกุล h หรือ นามสกุล cpp รวมทั้งไฟล์ฐานข้อมูลโปรแกรมที่ใช้ในคลาสวิซาร์ด (Class Wizard) เป็นต้น
2. โปรเจกต์ไฟล์จะเก็บค่าตัวเลือกสำหรับการคอมไพล์และลิงค์ (Link) กับไลบรารี (Library) ใดๆ หรือมีการสร้างส่วนประกอบ (Component) อื่นๆอีกหรือไม่ เช่น ส่วนประกอบในการดีบัก
3. โปรเจกต์จะเก็บค่าตัวเลือกที่แสดงว่าโปรเจกต์นี้เป็นโปรเจกต์แบบใดเมื่อทำการคอมไพล์ เช่น เป็นวินโดว์แอปพิเคชั่น (Windows Application) โดยมีนามสกุลเป็น exe หรือเป็นพวกไดนามิกลิงค์ไลบรารี (Dynamic-Ling Library) โดยมีนามสกุลเป็น dll เป็นต้น

Microsoft Foundation Class (MFC)

เป็นไลบรารีที่ทางบริษัท Microsoft สร้างขึ้นเพื่อช่วยให้นักพัฒนาโปรแกรมประยุกต์เขียนโปรแกรมได้ง่ายขึ้น ซึ่งภายในตัว MFC เองจะประกอบด้วยคลาสพื้นฐานต่างๆที่ต้องใช้ในการสร้างหรือแสดงผลในระบบวินโดว์ โดยจะช่วยให้โปรแกรมประยุกต์ที่เขียนขึ้นนั้นมีขนาดเล็กลงและไม่มีความซับซ้อนมาก ทำให้การเขียนโปรแกรมประยุกต์ก็จะเขียนได้ง่ายขึ้น

ได้มีการพัฒนา MFC มาตลอด จากเวอร์ชัน 1.0 จนในปัจจุบันมีการพัฒนามาจนถึงเวอร์ชัน 6.0 ซึ่งภายใน MFC ถูกบรรจุคลาสต่างๆที่สำคัญไว้มากมาย โดยเราสามารถแบ่งเป็น 2 ส่วนใหญ่ๆ คือ

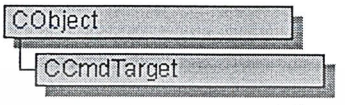
- คลาสที่มีต้นกำเนิดมาจากคลาสแม่ (Parent Class) คือ CObject
- คลาสที่เป็นคลาสโดด คือ คลาสที่ถูกสร้างขึ้นมาโดยไม่มีการสืบทอดมาจากคลาสใดๆ คลาสเหล่านี้เป็นคลาสที่สร้างขึ้นมาเพื่อใช้งานเฉพาะอย่างเท่านั้น โดยแบ่งได้เป็น 9 หมวดใหญ่ๆ ส่วนคลาสที่เราต้องใช้บ่อยๆคือ คลาส Simple Value Types ซึ่งเป็นคลาสที่ใช้สำหรับกำหนดชนิดของข้อมูล

ใน MFC จะมีคลาสชื่อ Object เป็นคลาสแม่ โดยเป็นคลาสที่ใช้ในการเขียนโปรแกรมในระบบ เช่น การสร้างวินโดว์ การสร้างคอนโทรล เป็นต้น คลาสเหล่านี้จะสืบทอดมาจากคลาส Object ทั้งสิ้น โดยคลาสที่มีความสำคัญในการสร้างโปรแกรมโดยทั่วไปได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส Cobject คลาสนี้เป็นคลาสแม่ของทุกๆคลาส ใช้ในการจัดการกับโปรแกรม เราเรียกคลาสนี้ว่าเป็นรูท (Root) หรือเป็นต้นกำเนิดของทุกๆคลาส หน้าที่ของคลาสนี้คือ จะเตรียมกระบวนการต่างๆที่ใช้ในโปรแกรมที่เราเขียนขึ้นมา เช่น กระบวนการเข้าถึงข้อมูลแบบเป็นลำดับ (Serialization) ในการเขียนข้อมูลและอ่านข้อมูลจากดิสก์ (Disk) ให้กับโปรแกรมหรือการจัดการการให้เวลาของไลบรารีในการรันโปแกรม

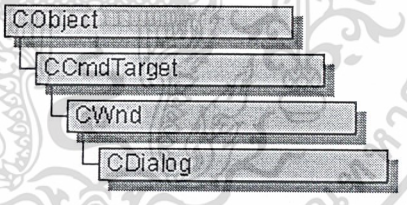
คลาส CCmdTarget



คลาส CcmdTarget เป็นคลาสที่สืบทอดมาจากคลาส Cobject ใช้สำหรับจัดการกับกระบวนการแมสเสจแมป (Message Map) ซึ่งเป็นการจัดการกับเหตุการณ์ต่างๆ เช่น การส่งข้อความติดต่อกัน การเรียกใช้คำสั่งระหว่างโปรแกรม เป็นต้น คลาส CcmdTarget นี้จะทำหน้าที่เป็นคลาสแม่ของคลาสต่างๆ ต่อไปนี้

- คลาส CwinApp ใช้สำหรับจัดการแอปพลิเคชัน เช่น กระบวนการเริ่มต้นการรันโปรแกรม
- คลาส CWnd เป็นคลาสแม่ของคลาสที่เป็นวินโดว์ทั้งหมด มีหน้าที่สร้างวินโดว์ ตลอดจนการควบคุมการทำงานของคอนโทรลในวินโดว์ด้วย
- คลาส CFrameWnd มีหน้าที่สร้างหน้าต่างวินโดว์แบบเฟรม (Frame) ซึ่งสืบทอดมาจากคลาส CWnd

คลาส CDialog



คลาส CDialog เป็นคลาสที่สืบทอดมาจากคลาส CWnd ซึ่งเป็นคลาสที่จะรับผิดชอบในการแสดงผลและควบคุมไดอะล็อกทั้งหมด เช่น การกดปุ่ม OK กับการกด Cancel และการทำงานของฟังก์ชัน OnInitDialog เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Beckett, Brian "The RSA algorithm" *Introduction to Cryptology and PC Security* McGraw-Hill, 1997

[2] Ford, W. "*Computer Communication Security*", Prentice Hall, 1994

[3] David Kosiur " Understanding Electronic Commerce", 1998

[4] Mickey Williams, David Hamilton "*Programming Windows NT4*", Sams Publishing, 1996

[5] Microsoft http://msdn.microsoft.com/library/psdk/crypto/portalapi_3351.htm?RLD=29

[6] Digital Signature Standard http://www.itl.nist.gov/fipspubs/fip186.htm

[7] Digital ID (Certificate) http://digitalid.verisign.com/client/help

[8] Digital Signature and Electronic Software http://www.elock.com

[9] Certificate http://www.verisign.com/support/tlc/per/installation/install.html

[10] Digital Signature http://www.geocities.com/NapaValley/1345/digsig.htm

[11] OLE http://www.softmosis.ca/WordFAQ.html#GetData

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้