

การออกแบบและพัฒนาโปรแกรมรักษาความปลอดภัยให้กับระบบ
แฟ้มข้อมูลด้วยเทคโนโลยีโครงสร้างพื้นฐานระบบกุญแจสาธารณะ

Design and Implement PKI-enabled Security

Application for File Systems



วัน เดือน ปี.....	24 มี.ค. 2550
เลขทะเบียน.....	01980
เลขเรียกหนังสือ.....	วท. / 11884 2550
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชา โครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2545
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การออกแบบและพัฒนาโปรแกรมรักษาความปลอดภัยให้กับระบบ เพิ่มข้อมูลด้วยเทคโนโลยีโครงสร้างพื้นฐานระบบกฎหมาย
นักศึกษา	นางกรรณรัตน์ คามตะศิลา
อาจารย์ที่ปรึกษา	ผศ.ดร. โชติพัชร ภรณ์วลัย
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

ปัจจุบัน เอกสารในรูปแบบอิเล็กทรอนิกส์มีความสำคัญในการดำเนินธุรกรรมอิเล็กทรอนิกส์ผ่านเครือข่ายอินเทอร์เน็ต ผู้ใช้จำเป็นต้องรักษาองค์ประกอบความปลอดภัยในการสื่อสารไว้ให้ครบถ้วน ลายมือชื่อดิจิทัลเป็นที่นิยมใช้เพื่อรับประกันว่าผู้ประกอบธุรกรรมไม่สามารถปฏิเสธความรับผิดชอบต่อธุรกรรมที่กระทำ ช่วยให้สามารถตรวจสอบความถูกต้องแท้จริงของข้อมูลอิเล็กทรอนิกส์และยืนยันบุคคลผู้สร้างได้ นอกจากนี้ การรักษาความลับของข้อมูลอิเล็กทรอนิกส์ก็จำเป็นไม่ยิ่งหย่อนไปกว่ากัน สามารถกระทำได้โดยการเข้ารหัสลับเพิ่มข้อมูลเพื่อป้องกันการเปิดอ่านโดยมิได้รับอนุญาต

โครงการพัฒนาระบบงานนี้ แบ่งออกเป็น 2 ส่วนหลัก ส่วนที่ 1 เป็นการปรับแต่งและทดสอบระบบผู้ให้บริการออกใบรับรอง (Certification Authority) ด้วยซอฟต์แวร์แบบ Open Source อย่าง OpenCA และ OpenSSL ส่วนที่ 2 เป็นการออกแบบและพัฒนาโปรแกรมรักษาความปลอดภัยให้กับเพิ่มข้อมูลด้วยเทคโนโลยีโครงสร้างพื้นฐานกฎหมาย โปรแกรมสามารถสร้าง ตรวจสอบลายมือชื่อดิจิทัลให้กับเพิ่มข้อมูล เข้าและถอดรหัสลับเพิ่มข้อมูลโดยใช้ใบรับรองอิเล็กทรอนิกส์ที่สร้างจากการทำงานในส่วนที่ 1 ได้ จุดประสงค์ของโครงการพัฒนาระบบงานนี้ คือ เพื่อศึกษาค้นคว้าและทดลองให้บริการออกใบรับรองอิเล็กทรอนิกส์ และสร้างโปรแกรมประยุกต์ที่รองรับการใช้งานใบรับรองอิเล็กทรอนิกส์ เพื่อให้เกิดการสื่อสารผ่านสื่ออิเล็กทรอนิกส์ได้อย่างปลอดภัย ทั้งยังส่งเสริมการใช้สื่ออิเล็กทรอนิกส์และใบรับรองอิเล็กทรอนิกส์โดยสร้างความมั่นใจในการแลกเปลี่ยนข้อมูลอย่างปลอดภัยให้กับผู้ใช้งานมากขึ้น

Title	Design and Implement PKI-enabled Security Application for File Systems
Student	Mrs. Krongrat Kamtasila
Advisor	Asst. Prof. Dr. Chotipat Pornavalai
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

Abstract

Presently, there is more concern of electronic documents security because Internet is widely used as the public channel for electronic transaction. There is need to secure transferred data. Digital signature is used to guarantee data integrity, non-repudiation and authorization. Confidentiality is accomplished by data encryption. Public Key Infrastructure and usage of Digital certificate are the acceptable technologies for securing transferred data.

This system development project is divided into two parts. On the server side, the Certification Authority has been set up using OpenCA, OpenSSL and OpenLDAP. The basic characteristics and usage of these open source software as the Certification Authority on Linux-based system are elaborated. On the client side, the project deals with the problem of design and implementation of PKI-enabled Security Application for File Systems. Two technologies, digital signature and encryption, are applied to solve the problem. As a result, parties can communicate more securely. Furthermore, this will encourage people to use digital certificate and have more confident to make electronic transaction.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานนี้ ผู้เขียนได้ทำการพัฒนาระบบจนเสร็จสมบูรณ์โดยได้รับความช่วยเหลือจากหลายท่าน โดยเฉพาะอย่างยิ่งอาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำ ข้อเสนอแนะ ชี้แจงข้อบกพร่องต่างๆ รวมทั้งเพื่อนร่วมงานที่ช่วยแบ่งปันความรู้ความเข้าใจเกี่ยวกับเทคโนโลยีที่ใช้ในโครงการ เพื่อให้โครงการสำเร็จได้อย่างสมบูรณ์

ขอขอบคุณพี่ First สำหรับแหล่งความรู้ ให้คำปรึกษา กำลังใจ การสนับสนุนและการผลักดัน ขอขอบคุณคุณพ่อ คุณแม่สำหรับกำลังใจที่มีให้เสมอมา



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาของโครงการ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตในการพัฒนาระบบงาน.....	2
1.4 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน.....	3
1.5 องค์ประกอบของระบบงาน.....	3
1.6 ขั้นตอนในการพัฒนาระบบงาน.....	4
1.7 ประโยชน์ที่คาดว่าจะได้รับ.....	5
2. ทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบงาน.....	6
2.1 เทคโนโลยีระบบรหัส.....	6
2.2 โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure - PKI) คืออะไร.....	7
2.3 ส่วนประกอบของโครงสร้างพื้นฐานกุญแจสาธารณะ.....	8
2.4 ใบรับรองอิเล็กทรอนิกส์มาตรฐาน X.509.....	12
2.5 การประยุกต์ใช้งานใบรับรองอิเล็กทรอนิกส์.....	13
2.6 อัลกอริทึมระบบรหัสแบบสมมาตร.....	20

บทที่

2.7 เทคโนโลยี Microsoft Cryptography API.....	20
3. การติดตั้ง ปรับแต่ง และทดสอบใช้งานระบบผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์..	25
3.1 การวางแผนปฏิบัติงาน.....	25
3.2 การวางแผนโครงสร้างของระบบ.....	27
3.3 การวางแผนโมเดลการรับรอง (Trust Model).....	27
3.4 การเริ่มต้นใช้งานซอฟต์แวร์ OpenCA.....	28
3.5 การประยุกต์ใช้งาน.....	32
3.6 การใช้งานระบบไคลเอนท์.....	37
3.7 ผลการใช้งาน.....	38
4. การวิเคราะห์และออกแบบโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล.....	40
4.1 การทำงานของโปรแกรม.....	40
4.2 Use Case Diagram.....	42
4.3 การวิเคราะห์ขั้นตอนการทำงานที่เกิดใน Use Case.....	43
4.4 การออกแบบ Element ต่างๆ ในระบบ.....	52
5. การพัฒนาโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล.....	65
5.1 ซอฟต์แวร์สำหรับการพัฒนาโปรแกรม.....	65
5.2 การนำ DLL - CryptAPI Wrapper version 1.2 มาใช้งาน.....	65
5.3 การใช้งาน CAPICOM 2.0.0.1.....	65
5.4 การพัฒนาโปรแกรม.....	67
5.5 อัลกอริทึมที่ใช้ในโปรแกรม.....	76
6. บทสรุปและแนวทางการพัฒนาในอนาคต.....	80
บรรณานุกรม.....	81
ภาคผนวก ก.....	83
ประวัติผู้เขียน.....	85

สารบัญตาราง

หน้า

ตารางที่

2.1 ฟังก์ชันเกี่ยวกับระบบรหัสของไมโครซอฟท์ (Cryptographic Function)..... 23

4.1 การอธิบาย Layer ต่างๆ ในโปรแกรม..... 54

4.2 การอธิบาย Method ต่างๆ ของ Class ใน Presentation Layer..... 56

4.3 การอธิบาย Method ต่างๆ ของ Class ใน BusinessLogics Layer..... 56

4.4 การแบ่งกลุ่มของ Class ใน CAPICOM..... 58

4.5 ประเภทของ Class ในกลุ่ม *Certificate Store Object*..... 58

4.6 ประเภทของ Class ในกลุ่ม *Digital Signature Object*..... 59

4.7 ประเภทของ Class ในกลุ่ม *Enveloped Data Object*..... 59

4.8 ประเภทของ Class ในกลุ่ม *Data Encryption Object*..... 60

4.9 ประเภทของ Class ในกลุ่ม *Auxiliary Object*..... 60

4.10 ประเภทของ Class ในกลุ่ม *Interoperability Interface*..... 61

4.11 Property ต่างๆ ใน CryptoAPI Wrapper version 1.2..... 61

4.12 การอธิบาย Method ของ CryptoAPI Wrapper Class ใน BusinessObjects Layer.. 62

4.13 Class ที่เรียกใช้จาก Xceed..... 64

สารบัญภาพ

หน้า

ภาพที่

2.1 Self-signed Certificate.....	8
2.2 รูปแบบของ Certificate Revocation List – CRL.....	10
2.3 กระบวนการสร้างลายมือชื่อดิจิทัล.....	14
2.4 การเข้าและถอดรหัสลับข้อมูล.....	15
2.5 SSL ทำงานบน TCP/IP และอยู่ใต้ Application Layer.....	16
2.6 SSL Handshake.....	18
2.7 สถาปัตยกรรมของ CryptoAPI.....	21
2.8 Key Database.....	22
3.1 โครงสร้างของ OpenCA.....	27
3.2 Trust Model แบบ Hierarchy.....	28
3.3 แสดงหน้าจอแรกสำหรับการ Initialize เซิร์ฟเวอร์ผู้ให้บริการออกใบรับรอง.....	28
3.4 แสดงหน้าจอแรกหลังจากการ Initialize เซิร์ฟเวอร์ผู้ให้บริการออกใบรับรอง.....	29
3.5 แสดงหน้าจอหลังจาก Initialize Database เสร็จสิ้นแล้ว.....	29
3.6 แสดงหน้าจอหลังจากสร้างคู่กุญแจแล้ว.....	30
3.7 แสดงหน้าจอหลังจากสร้าง Certificate Request.....	31
3.8 แสดงหน้าจอหลังจากสร้าง Certificate แล้ว.....	31
3.9 แสดงหน้าจอแรกกระบวนการขึ้นตอนการขอใบรับรอง.....	33
3.10 แสดงหน้าจอหลังจากที่ผู้ใช้กรอกแบบฟอร์มขอใบรับรองแล้ว.....	33
3.11 แสดงหน้าจอ Pending Request สำหรับเจ้าหน้าที่ออกใบรับรอง.....	34
3.12 แสดงหน้าจอรายละเอียดของใบรับรองอิเล็กทรอนิกส์.....	34
3.13 หน้าจอแรกเมื่อเปิดเว็บไซด์.....	35
3.14 หน้าจอแรกหลังจากยอมรับการใช้งานใบรับรองจากเซิร์ฟเวอร์.....	35
3.15 หน้าจอแสดงรายชื่อใบรับรองฝั่งไคลเอ็นท์.....	36
3.16 สำหรับกรอกรหัสผ่านเข้าถึงกุญแจส่วนตัว.....	36

ภาพที่

3.17 แสดงหน้าจอโปรแกรม LDAP Browser/Editor.....	37
3.18 แสดงหน้าจอแสดงรายละเอียดใบรับรองอิเล็กทรอนิกส์.....	37
3.19 วงจรชีวิตของใบรับรองอิเล็กทรอนิกส์.....	38
4.1 การสร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูล.....	40
4.2 การตรวจสอบลายมือชื่อดิจิทัลของแฟ้มข้อมูล.....	41
4.3 การเข้ารหัสลับแฟ้มข้อมูล.....	41
4.4 การถอดรหัสลับแฟ้มข้อมูล.....	42
4.5 Use Case Model ส่วนผู้ส่ง.....	42
4.6 Use Case Model ส่วนผู้รับ.....	43
4.7 Basic Flow : Sequence Diagram ของ SignFile Use Case.....	44
4.8 Basic Flow : Collaboration Diagram ของ SignFile Use Case.....	44
4.9 Basic Flow : Sequence Diagram ของ VerifyFile Use Case.....	45
4.10 Basic Flow : Collaboration ของ VerifyFile Use Case.....	45
4.11 Basic Flow :Sequence Diagram ของ EncryptFile Use Case.....	46
4.12 Basic Flow :Sequence Diagram ของ EncryptFile Use Case.....	46
4.13 Basic Flow : Sequence Diagram ของ DecryptFile Use Case.....	47
4.14 Basic Flow : Collaboration Diagram ของ DecryptFile Use Case.....	47
4.15 SignFile Use Case – VOPC.....	48
4.16 VerifyFile Use Case – VOPC.....	49
4.17 EncryptFile Use Case – VOPC.....	50
4.18 DecryptFile Use Case – VOPC.....	51
4.19 Architectural Layers and Their Dependencies.....	53
4.20 Class Diagram.....	55
5.1 แสดงหน้าหลักของโปรแกรม.....	67
5.2 แสดงหน้าจอส่วนการสร้างและตรวจสอบลายมือชื่อดิจิทัล.....	67
5.3 แสดงหน้าจอการสร้างลายมือชื่อดิจิทัล.....	68
5.4 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัว.....	68

ภาพที่

5.5 แสดงผลการสร้างลายมือชื่อดิจิทัล.....	68
5.6 แสดงหน้าจอการตรวจสอบลายมือชื่อดิจิทัล.....	69
5.7 แสดงผลตรวจสอบลายมือชื่อดิจิทัล.....	69
5.8 แสดงส่วนการเข้าและถอดรหัสลับเพิ่มข้อมูล.....	69
5.9 แสดงหน้าจอการเข้ารหัสลับข้อมูล.....	70
5.10 แสดงผลการเข้ารหัสลับเพิ่มข้อมูล.....	70
5.11 แสดงเพิ่มข้อมูลที่ผ่านการเข้ารหัสลับแล้ว.....	70
5.12 แสดงหน้าจอการถอดรหัสลับ.....	71
5.13 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัว.....	71
5.14 หน้าจอแสดงส่วนการรักษาความปลอดภัยสองแบบควบคู่กัน.....	72
5.15 หน้าจอแสดงการสร้างลายมือชื่อดิจิทัลและการเข้ารหัสลับเพิ่มข้อมูล.....	72
5.16 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัวของ User 1.....	73
5.17 หน้าจอแสดงผลหลังจากการสร้างลายมือชื่อดิจิทัลและเข้ารหัสลับ.....	73
5.18 หน้าจอแสดงการถอดรหัสลับและตรวจสอบลายมือชื่อดิจิทัล.....	74
5.19 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัวของ User 2.....	74
5.20 หน้าจอแสดงผลหลังจากการถอดรหัสลับและตรวจสอบ.....	75
5.21 แสดงหน้าจอเกี่ยวกับใบรับรองอิเล็กทรอนิกส์.....	75
5.22 แสดงหน้าจอรายการใบรับรองอิเล็กทรอนิกส์ส่วนบุคคล.....	75
5.23 แสดงข้อมูลใบรับรองอิเล็กทรอนิกส์.....	76

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในปัจจุบัน เอกสารในรูปแบบอิเล็กทรอนิกส์มีความสำคัญยิ่งขึ้นในการดำเนินธุรกรรมอิเล็กทรอนิกส์ผ่านเครือข่ายอินเทอร์เน็ต ผู้ดำเนินธุรกรรมต้องรักษาองค์ประกอบความปลอดภัยในการสื่อสารไว้ให้ครบถ้วน ไม่ว่าจะเป็นการรักษาความลับของข้อมูลอิเล็กทรอนิกส์ (Data Confidentiality) การตรวจสอบความถูกต้องแท้จริงของข้อมูลอิเล็กทรอนิกส์ (Data Integrity) การยืนยันตัวตนบุคคลผู้สร้างหรือผู้ส่งข้อมูลอิเล็กทรอนิกส์ (Authentication) และการยืนยันว่าผู้ประกอบธุรกรรมไม่สามารถปฏิเสธความรับผิดชอบต่อธุรกรรมได้ (Non-Repudiation)

แม้ระบบคอมพิวเตอร์ในปัจจุบันจะมีระบบรักษาความปลอดภัยในขั้นพื้นฐานอยู่แล้ว แต่ในกรณีที่ข้อมูลมีความสำคัญมาก ก็จำเป็นที่จะต้องมีการป้องกันการเข้าถึงข้อมูลที่ปลอดภัยมากขึ้น ทั้งนี้ เทคโนโลยีรักษาความปลอดภัยของข้อมูลอย่างโครงสร้างพื้นฐานกุญแจสาธารณะ (PKI – Public key Infrastructure) และการใช้งาน ใบรับรองอิเล็กทรอนิกส์ (Digital Certificate) เป็นเทคโนโลยีที่ได้รับการยอมรับกันในวงกว้างว่ามีความน่าเชื่อถือและเป็นที่ยอมรับใช้

การนำใบรับรองอิเล็กทรอนิกส์ไปประยุกต์ใช้งาน ทำได้ 2 ลักษณะ คือ การสร้างลายมือชื่อดิจิทัลและการเข้ารหัสลับ ลายมือชื่อดิจิทัลเป็นที่ยอมรับใช้เพื่อรับประกันว่าผู้ประกอบธุรกรรมไม่สามารถปฏิเสธความรับผิดชอบต่อธุรกรรมที่ตนกระทำได้ เหมาะสมกับงานด้านเอกสารทางกฎหมายหรือข้อตกลงทางธุรกิจหรืออื่นๆ ทั้งยังช่วยให้สามารถตรวจสอบความถูกต้องแท้จริงของข้อมูลและยืนยันตัวตนบุคคลผู้สร้างข้อมูลอิเล็กทรอนิกส์ได้ด้วย ก่อให้เกิดประโยชน์ในการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ระหว่างผู้ใช้งาน นอกจากนั้น การรักษาความลับของข้อมูลอิเล็กทรอนิกส์ก็จำเป็นไม่ยิ่งหย่อนไปกว่ากัน สามารถกระทำได้โดยการเข้ารหัสลับข้อมูลเพื่อป้องกันการเปิดอ่านโดยมิได้รับอนุญาต

โครงการพัฒนาระบบงานนี้ แบ่งออกเป็น 2 ส่วนหลัก ส่วนที่ 1 เป็นการปรับแต่งและทดสอบระบบผู้ให้บริการออกใบรับรองด้วยซอฟต์แวร์แบบ Open Source อย่าง OpenCA และ OpenSSL ส่วนที่ 2 เป็นการออกแบบและพัฒนาโปรแกรมรักษาความปลอดภัยให้กับแฟ้มข้อมูลด้วยเทคโนโลยีโครงสร้างพื้นฐานกุญแจสาธารณะ โดยอาศัยเทคโนโลยี 2 ข้อด้วยกัน คือ การสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายมือชื่อดิจิทัลและการเข้ารหัสลับเพิ่มข้อมูล การสร้างลายมือชื่อดิจิทัลให้กับเพิ่มข้อมูลทำให้สามารถตรวจสอบความถูกต้อง ยืนยันตัวบุคคลผู้สร้างและเป็นหลักฐานให้ไม่สามารถปฏิเสธความรับผิดชอบต่อเพิ่มข้อมูลนั้น ส่วนการเข้ารหัสลับเพิ่มข้อมูลจะช่วยรักษาความลับของข้อมูลอิเล็กทรอนิกส์ได้ โครงการพัฒนาระบบงานนี้ มีจุดประสงค์เพื่อศึกษาค้นคว้า ทดลองให้บริการออกใบรับรองอิเล็กทรอนิกส์ และสร้างโปรแกรมประยุกต์ที่รองรับการใช้งานใบรับรองอิเล็กทรอนิกส์ เพื่อให้เกิดการสื่อสารผ่านสื่ออิเล็กทรอนิกส์ได้อย่างปลอดภัย นอกจากนั้น ยังส่งเสริมการใช้สื่ออิเล็กทรอนิกส์และใบรับรองอิเล็กทรอนิกส์โดยสร้างความมั่นใจในการแลกเปลี่ยนข้อมูลอย่างปลอดภัยให้กับผู้ใช้งานมากขึ้น

1.2 วัตถุประสงค์

- 1 เพื่อศึกษาค้นคว้าเรื่องโครงสร้างพื้นฐานกฎและมาตรฐาน
- 2 เพื่อศึกษาและทดลองเป็นหน่วยผู้ให้บริการออกใบรับรองด้วยซอฟต์แวร์ OpenCA และ OpenSSL
- 3 เพื่อประหยัดค่าใช้จ่ายด้านซอฟต์แวร์ผู้ให้บริการออกใบรับรอง (Certification Authority - CA) ด้วยการใช้ซอฟต์แวร์แบบ Open Source
- 4 เพื่อพัฒนาโปรแกรมประยุกต์ที่ใช้ในการสร้างลายมือชื่อดิจิทัลให้กับเพิ่มข้อมูลอิเล็กทรอนิกส์ ซึ่งจะทำได้สามารถรับรองความถูกต้องแท้จริงของเพิ่มข้อมูล ยืนยันตัวบุคคลผู้สร้างหรือผู้ส่งข้อมูลอิเล็กทรอนิกส์และยืนยันว่าผู้สร้างหรือส่ง ไม่สามารถปฏิเสธความรับผิดชอบต่อเพิ่มข้อมูลได้
- 5 เพื่อพัฒนาโปรแกรมประยุกต์ที่ใช้ในการเข้าและถอดรหัสลับของข้อมูล เพื่อเป็นการรักษาความลับของเพิ่มข้อมูลมิให้บุคคลที่ไม่ได้รับอนุญาตสามารถเปิดอ่านเพิ่มข้อมูลได้
- 6 เพื่อสนับสนุนการใช้งานใบรับรองอิเล็กทรอนิกส์ให้แพร่หลายมากขึ้นอีกทางหนึ่ง เนื่องจากปัจจุบันยังมีโปรแกรมประยุกต์ที่รองรับการใช้งานอยู่ไม่มากนัก

1.3 ขอบเขตในการพัฒนาระบบงาน

- 1 ติดตั้งระบบผู้ให้บริการออกใบรับรองด้วยซอฟต์แวร์ OpenCA และ OpenSSL บนระบบปฏิบัติการลินุกซ์ เพื่อทำหน้าที่ออกใบรับรองอิเล็กทรอนิกส์สำหรับนำมาใช้งานในโปรแกรมรักษาความปลอดภัยให้กับเพิ่มข้อมูล
- 2 ติดตั้งระบบไคลเอนต์เพื่อเป็นทั้งจัดเก็บและเผยแพร่ข้อมูลบุคคลและใบรับรองอิเล็กทรอนิกส์ผ่านเครือข่ายคอมพิวเตอร์

- 3 การยืนยันตัวบุคคลด้วยใบรับรองอิเล็กทรอนิกส์ (Client Authentication) เพื่อเข้าใช้งานระบบผู้ให้บริการออกใบรับรองผ่านเว็บ
- 4 สร้างโปรแกรมรักษาความปลอดภัยให้กับแฟ้มข้อมูลด้วยเทคโนโลยีโครงสร้างพื้นฐานกุญแจสาธารณะโดยใช้ใบรับรองอิเล็กทรอนิกส์ โปรแกรมมีความสามารถในการทำงานดังนี้
 - สร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูลประเภทต่างๆ
 - ตรวจสอบความถูกต้องของลายมือชื่อดิจิทัล
 - ทำการเข้าและถอดรหัสลับแฟ้มข้อมูลประเภทต่างๆ

1.4 ทฤษฎีที่ใช้ในการพัฒนาระบบงาน

- 1 ศึกษาหลักการ วิธีการติดตั้งและใช้งานซอฟต์แวร์ต่างๆ ที่ใช้สำหรับการเป็นผู้ให้บริการออกใบรับรอง (Certification Authority) ได้แก่ OpenCA, OpenSSL, OpenLDAP และ Apache Web Server
- 2 ศึกษาเทคโนโลยีระบบรหัสและเทคโนโลยีการรักษาความปลอดภัยของข้อมูล
- 3 ศึกษารายละเอียดของใบรับรองอิเล็กทรอนิกส์มาตรฐาน X.509
- 4 ศึกษาหลักการนำใบรับรองอิเล็กทรอนิกส์มาใช้ในการยืนยันตัวบุคคลเพื่อเข้าใช้งานเว็บไซต์ (Server Authentication และ Client Authentication)
- 5 ศึกษาหลักการวิเคราะห์และออกแบบซอฟต์แวร์แบบ Object-Oriented Technology
- 6 ศึกษาการใช้งานโปรแกรม Visual Basic 6.0 เพื่อสร้างโปรแกรมประยุกต์
- 7 ศึกษาวิธีการเรียกใช้งานฟังก์ชันระบบรหัสต่างๆ บนระบบปฏิบัติการไมโครซอฟท์ และศึกษารายละเอียดของอัลกอริทึมระบบรหัสแต่ละแบบ

1.5 องค์ประกอบของระบบงาน

ระบบงาน ประกอบไปด้วยองค์ประกอบต่างๆ ดังต่อไปนี้

- 1 เครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ระบบปฏิบัติการ Linux Redhat 7.2
ทำหน้าที่
 - เครื่องให้บริการออกใบรับรองอิเล็กทรอนิกส์ (Certification Authority Server)
 - เครื่องให้บริการเว็บสำหรับติดต่อระบบให้บริการออกใบรับรองอิเล็กทรอนิกส์ (Web Server)

มีการติดตั้งซอฟต์แวร์เพื่อให้บริการ ดังนี้

- ซอฟต์แวร์ระบบผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์ เลือกใช้ OpenCA เวอร์ชัน 0.9.1
 - ซอฟต์แวร์ SSL/TLS เลือกใช้ OpenSSL เวอร์ชัน 0.9.6
 - เว็บเซิร์ฟเวอร์ที่รองรับมาตรฐาน SSL/TLS เลือกใช้ Apache เวอร์ชัน 1.3.27
 - ไคลเอนต์เซิร์ฟเวอร์มาตรฐาน LDAP เลือกใช้ OpenLDAP เวอร์ชัน 2.0.25
- 2 เครื่องคอมพิวเตอร์ไคลเอนต์ ระบบปฏิบัติการ Windows 2000 Professional ทำหน้าที่
- เครื่องสำหรับพัฒนาโปรแกรมรักษาความปลอดภัยให้กับแฟ้มข้อมูล
 - เครื่องไคลเอนต์สำหรับติดต่อกับเครื่องให้บริการออกใบรับรองอิเล็กทรอนิกส์ เมื่อต้องการทำงานร่วมกัน ได้แก่ การขอใบรับรองผ่านเว็บ การอนุมัติคำขอผ่านเว็บ การออกใบรับรองผ่านเว็บ

มีการติดตั้งซอฟต์แวร์ ดังนี้

- โปรแกรม Visual Basic 6.0 สำหรับพัฒนาโปรแกรมรักษาความปลอดภัยให้กับแฟ้มข้อมูล
- Platform SDK ของไมโครซอฟท์เพื่อปรับปรุงฟังก์ชันงานด้านระบบรหัส Cryptography API ให้เป็นเวอร์ชันล่าสุด
- ActiveX สำหรับการทำงานระบบรหัส คือ CAPICOM 2.0.0.1 จากไมโครซอฟท์
- โปรแกรมเรียกดูเว็บ ในที่นี้จะใช้โปรแกรมอินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) ที่มีความสามารถในการเชื่อมต่อ SSL ได้ที่ระดับ 128 บิต

1.6 ขั้นตอนในการพัฒนาระบบงาน

ประกอบไปด้วยขั้นตอนต่างๆ ดังนี้

- 1 การศึกษาความเป็นไปได้ในการพัฒนา

เป็นการศึกษาความเป็นไปได้ในการพัฒนาโครงการ กำหนดขอบเขตของปัญหา และวางแผนวิธีการพัฒนาโปรแกรม รวมถึงกำหนดเป้าหมายในการพัฒนาโครงการ
- 2 การศึกษาและวิเคราะห์ทฤษฎีที่เลือกใช้

ศึกษาอัลกอริทึมระบบรหัสที่มีใช้งานในปัจจุบัน ศึกษาการใช้งาน Cryptography API ของไมโครซอฟท์
- 3 การวิเคราะห์และออกแบบ

ทำการวิเคราะห์และออกแบบโครงการพัฒนาระบบงาน

- 4 การพัฒนาและทดสอบ
 - ทำการติดตั้งระบบผู้ให้บริการออกใบรับรองและทดสอบการทำงาน
 - ทำการพัฒนาโปรแกรมและทดสอบการทำงานในฟังก์ชันต่างๆ
- 5 การทดลองใช้งานและปรับปรุงแก้ไข

นำโปรแกรมมาทดลองใช้งานและปรับปรุงแก้ไขเพื่อให้สามารถใช้งานได้ง่ายขึ้น

1.7 ประโยชน์ที่คาดว่าจะได้รับ

- 1 ได้พัฒนาความรู้ความเข้าใจเรื่องโครงสร้างพื้นฐานกฎหมายสาธารณะ
- 2 ได้พัฒนาความรู้ความเข้าใจเรื่องใบรับรองอิเล็กทรอนิกส์
- 3 ได้พัฒนาความรู้ ความสามารถในการวิเคราะห์ ออกแบบและพัฒนาระบบงาน และสามารถนำไปใช้ประโยชน์ต่อการทำงานในอนาคตได้
- 4 ได้โปรแกรมประยุกต์ที่รองรับการใช้งานใบรับรองอิเล็กทรอนิกส์ที่ทำความเข้าใจและใช้งานได้ง่าย
- 5 ได้สนับสนุนให้เกิดโปรแกรมรองรับการใช้งานใบรับรองอิเล็กทรอนิกส์เพิ่มขึ้นอีกหนึ่งทางเลือก

เนื้อหาในบทต่างๆ มีดังต่อไปนี้

- บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบงาน
- บทที่ 3 กล่าวถึงการติดตั้ง ปรับแต่ง และทดสอบใช้งานระบบผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์ (Certification Authority)
- บทที่ 4 กล่าวถึงการวิเคราะห์และออกแบบโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล
- บทที่ 5 กล่าวถึงการพัฒนาโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล
- บทที่ 6 กล่าวถึงบทสรุปและแนวทางการพัฒนาในอนาคต

บทที่ 2

ทฤษฎีที่เกี่ยวข้องในการพัฒนาระบบงาน

2.1 เทคโนโลยีระบบรหัส

เทคโนโลยีระบบรหัสเป็นเทคโนโลยีที่มีการนำมาใช้ในการรักษาความปลอดภัยของข้อมูลอย่างแพร่หลาย โดยทั่วไป แบ่งออกเป็น 2 ประเภทหลัก ได้แก่ ระบบรหัสแบบสมมาตร (Symmetric-Key Cryptography) และ ระบบรหัสแบบอสมมาตร (Asymmetric-Key Cryptography)

2.1.1 ระบบรหัสแบบสมมาตร (Symmetric-Key Cryptography)

ประกอบด้วย 2 ขั้นตอนหลัก ได้แก่ การเข้ารหัสลับ (Data Encryption) และการถอดรหัสลับ (Data Decryption) การเข้ารหัสลับเป็นการแปลงข้อมูลให้เปลี่ยนจากรูปแบบเดิมไปเป็นอีกรูปแบบหนึ่ง โดยนำข้อมูลอิเล็กทรอนิกส์ธรรมดา (Plain Text) กับกุญแจ (Key) ซึ่งเป็นรหัสทางคอมพิวเตอร์มาผ่านกระบวนการทางคณิตศาสตร์ ผลที่ได้คือข้อมูลที่เข้ารหัส (Cipher Text) ซึ่งเป็นข้อมูลที่ไม่สามารถอ่านเข้าใจได้ เมื่อส่งไปยังผู้รับ ผู้รับก็จะทำการถอดรหัสลับข้อมูล ซึ่งเป็นกระบวนการแปรสภาพข้อมูลที่เข้ารหัสนี้ให้กลับคืนสู่ข้อมูลเดิมโดยใช้กุญแจชุดเดียวกัน การใช้กุญแจชุดเดียวกันทั้งด้านผู้ส่งและผู้รับนี้เองเป็นที่มาของชื่อระบบรหัสแบบสมมาตร เนื่องจากคำว่า “สมมาตร” เป็นการอธิบายถึงความเท่ากันหรือเหมือนกันของสองข้าง ซึ่งในที่นี้ก็คือตัวกุญแจนั่นเอง โดยกุญแจซึ่งอยู่ในรูปรหัสคอมพิวเตอร์นี้ เป็นตัวแปรสำคัญสำหรับการเข้าและถอดรหัสลับข้อมูล ซึ่งขนาดของกุญแจ (มีหน่วยเป็นบิต : Bit) จะแสดงถึงระดับความปลอดภัยของข้อมูลที่ได้รับการเข้ารหัสลับ โดยการใช้กุญแจที่มีความยาวหรือจำนวนบิตสูง จะทำให้การเข้ารหัสลับข้อมูลนั้นมีความปลอดภัยมากยิ่งขึ้น โดยขนาดของกุญแจที่นิยมใช้ในระบบรหัสแบบสมมาตร คือ 40 – 256 บิต

ทั้งนี้ ข้อดีของระบบรหัสแบบนี้ คือ สามารถเข้าและถอดรหัสลับได้รวดเร็ว (เมื่อเทียบกับระบบรหัสแบบอสมมาตร) และใช้ในการเก็บรักษาความลับของข้อมูลได้ ส่วนข้อเสีย คือ บริหารจัดการกุญแจ (Key Management) ยาก เนื่องจากผู้ใช้ต้องมีกุญแจ 1 ดอกสำหรับติดต่อกับบุคคล 1 คน ดังนั้นหากติดต่อกับบุคคลจำนวนมาก ก็จะต้องเก็บกุญแจเพิ่มขึ้นตามจำนวนผู้ที่ติดต่อ และการส่งกุญแจไปยังบุคคลที่ติดต่อด้วยนั้น มีความเสี่ยงในการที่ผู้อื่นอาจลักลอบขโมยกุญแจไประหว่างการสื่อสารส่งกุญแจดังกล่าว ตัวอย่างวิธีการเข้าและถอดรหัสลับของระบบรหัสแบบสมมาตร

เอกสารนี้เป็นเอกสารของงานวิจัยที่ได้รับทุนสนับสนุนจากสำนักงานการวิจัยแห่งชาติ (วช.) กระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม ไม่สามารถเผยแพร่หรือใช้ประโยชน์อื่นใดได้โดยไม่ได้รับอนุญาต
ไม่ว่าการตีพิมพ์นี้ในสิ่งอื่นใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น Data Encryption Standard (DES), Triple DES (3DES), Advanced Encryption Standard (AES), CAST, RC2 เป็นต้น

2.1.2 ระบบรหัสแบบอสมมาตร (Asymmetric-Key Cryptography)

ระบบรหัสแบบอสมมาตรเป็นระบบที่ใช้กุญแจคู่ (Key Pair) ซึ่งประกอบด้วยกุญแจส่วนตัว (Private Key) และกุญแจสาธารณะ (Public Key) กล่าวคือผู้ใช้ 1 คน (เจ้าของกุญแจคู่) จะใช้กุญแจคู่นี้ในการติดต่อสื่อสารกับบุคคลอื่น โดยไม่จำเป็นต้องมีกุญแจเป็นจำนวนมากเช่นเดียวกับกรณีของระบบรหัสแบบสมมาตร ในการเก็บรักษากุญแจนั้น กุญแจส่วนตัวจะต้องเก็บรักษาไว้กับเจ้าของกุญแจและห้ามไม่ให้ผู้อื่นล่วงรู้ได้ ส่วนกุญแจสาธารณะนั้น ต้องมีการประกาศให้ผู้อื่นรับรู้หรือเก็บไว้ในที่ซึ่งบุคคลอื่นมานำเอาไปใช้ได้ ในการเข้ารหัสข้อมูลจะต้องใช้กุญแจดอกหนึ่งในการเข้ารหัส และใช้กุญแจอีกดอกหนึ่งที่เป็นคู่กันในการถอดรหัส เช่น หากใช้กุญแจสาธารณะในการเข้ารหัสก็จะต้องใช้กุญแจส่วนตัวในการถอดรหัส ในทางกลับกัน หากใช้กุญแจส่วนตัวในการเข้ารหัสก็จะต้องใช้กุญแจสาธารณะในการถอดรหัส ซึ่งวิธีการเลือกใช้กุญแจในการเข้ารหัสจะขึ้นอยู่กับประเภทของการใช้งาน เช่น นำไปประยุกต์ใช้งานในการรักษาความลับของข้อมูล หรือใช้งานในการสร้างลายมือชื่อดิจิทัล (Digital Signature) ขนาดของกุญแจที่นิยมใช้ในระบบรหัสแบบอสมมาตร คือ 512 – 2048 บิต

ข้อดีของระบบรหัสแบบอสมมาตร คือ สามารถบริหารกุญแจ (Key Management) ได้ดีและเป็นระบบ เนื่องจากผู้ใช้แต่ละคนต้องการกุญแจ 1 คู่ (กุญแจส่วนตัวและกุญแจสาธารณะ) เท่านั้น สามารถใช้งานเพื่อการรักษาความลับของข้อมูล การระบุตัวบุคคลผู้ส่ง รวมไปถึงการยืนยันความถูกต้องของข้อมูลและการป้องกันการปฏิเสธความรับผิดชอบได้ แต่ข้อเสียคือ เข้ารหัสและถอดรหัสได้ช้า (เมื่อเทียบกับระบบรหัสแบบสมมาตร) ตัวอย่างวิธีที่ใช้ในระบบรหัสแบบอสมมาตร เช่น RSA, Digital Signature Algorithm (DSA) เป็นต้น

2.2 โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure - PKI) คืออะไร

ส่วนประกอบหลักของโครงสร้างพื้นฐานกุญแจสาธารณะ คือ ผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์ (Certification Authority) ระบบไดเรกทอรีเพื่อจัดเก็บและเผยแพร่ใบรับรองอิเล็กทรอนิกส์ (Directory Services) และระบบจัดการบริหารกุญแจ (Certificate Management System) โดยแนวคิดหลักที่มีบทบาทสำคัญคือ ใช้เทคโนโลยีระบบรหัสแบบอสมมาตร (Asymmetric-Key Cryptography) ซึ่งผู้ใช้งานจะมีกุญแจ 2 ดอก (Key Pair) คือ กุญแจสาธารณะที่เปิดเผยให้ผู้อื่นนำไปใช้งานได้ ซึ่งได้รับการจัดเก็บไว้บนใบรับรองอิเล็กทรอนิกส์ และกุญแจส่วนตัวที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เจ้าของต้องเก็บเป็นความลับ ผู้เป็นเจ้าของกุญแจคู่สามารถนำไปใช้เข้าและถอดรหัสด้วยระบบรหัสแบบอสมมาตร หรือสร้างลายมือชื่อดิจิทัล (Digital Signature) ได้ นอกจากนั้น ผู้ให้บริการออกใบรับรองจะดูแลการเผยแพร่ใบรับรองอิเล็กทรอนิกส์ ตรวจสอบและประกาศสถานะของใบรับรองอิเล็กทรอนิกส์ที่หมดอายุด้วย

2.3 ส่วนประกอบของโครงสร้างพื้นฐานกุญแจสาธารณะ

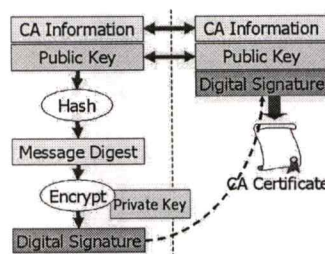
โครงสร้างพื้นฐานกุญแจสาธารณะมีส่วนประกอบหลายส่วนที่ทำงานร่วมกันเพื่อสร้าง (Issue) แจกจ่าย (Distribute) และจัดการเกี่ยวกับการใช้งาน (Manage) ใบรับรองอิเล็กทรอนิกส์ กล่าวคือ ผู้ให้บริการออกใบรับรอง (Certification Authority - CA) หน่วยรับลงทะเบียน (Registration Authority - RA) กระบวนการเพิกถอนใบรับรองอิเล็กทรอนิกส์ (Certificate Revocation) และระบบจัดเก็บข้อมูลใบรับรองอิเล็กทรอนิกส์ (Certificate Repository) ในที่นี้จะขอกล่าวถึงส่วนประกอบสำคัญ ดังนี้

2.3.1 ผู้ให้บริการออกใบรับรอง (Certification Authority - CA)

คือ ผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์ที่เป็นหน่วยงานภายนอกที่ได้รับความเชื่อถือ (Trusted Third Party) มีหน้าที่หลักคือ เป็นผู้ออกใบรับรองอิเล็กทรอนิกส์ให้กับบุคคลหรือสิ่งอื่นๆ (ในที่นี้จะเรียกรวมว่าเป็น Subject) ทั้งนี้ ผู้ให้บริการออกใบรับรองจะมีใบรับรองอิเล็กทรอนิกส์ของตนเอง ที่อาจเกิดจากการสร้างเอง (Self-signed Certificate) หรือให้ผู้ให้บริการออกใบรับรองอื่นสร้างให้ สิ่งนี้จะใช้ระบุตัวตนของผู้ให้บริการออกใบรับรองเอง ดังนั้นผู้ให้บริการออกใบรับรองจึงเป็นจุดเริ่มต้นของการสร้างระบบการรับรอง (Trust) แต่ละ Subject จะต้องยอมรับผู้ให้บริการออกใบรับรองและบุคคลอื่นหรือสิ่งใดก็ตามที่ผู้ให้บริการออกใบรับรองได้ออกใบรับรองอิเล็กทรอนิกส์ให้

2.3.1.1 การสร้าง Self-signed Certificate

ในการเริ่มต้นเป็นผู้ให้บริการออกใบรับรอง จะต้องสร้างใบรับรองอิเล็กทรอนิกส์ของผู้ให้บริการออกใบรับรองเองก่อน ทั้งนี้ขั้นตอนการสร้าง Self-signed Certificate แสดงดังในภาพที่ 2.1



ภาพที่ 2.1 Self-signed Certificate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่ควรเปิดเผยให้บุคคลภายนอกเห็น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ให้บริการออกใบรับรองเป็นหัวใจสำคัญของโครงสร้างพื้นฐานกุญแจสาธารณะ และรับผิดชอบหน้าที่สำคัญ คือ การจัดการใบรับรองอิเล็กทรอนิกส์ (Certificate Management) กล่าวคือ การออกใบรับรอง (Issuance) การเพิกถอนใบรับรอง (Revocation) การให้ใบรับรองใหม่ (Renewal) การเผยแพร่ใบรับรอง การสร้างรายชื่อการเพิกถอนใบรับรอง (Certificate Revocation List - CRL) และทำการบันทึกเหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบ

2.3.1.2 ขั้นตอนในการออกใบรับรองอิเล็กทรอนิกส์ให้กับ Subject ใดๆ ประกอบไปด้วย

1. Subject ส่งคำร้องขอใบรับรอง (Certificate Request)
2. Subject เป็นผู้สร้างกุญแจคู่โดยผู้ให้บริการออกใบรับรองเป็นผู้จัดเตรียมฟังก์ชันนี้ให้
3. ผู้ให้บริการออกใบรับรองจะจัดการออกใบรับรองอิเล็กทรอนิกส์ให้ ทั้งนี้ คำร้องดังกล่าวอาจเป็นไปได้ตั้งแต่ขอใหม่ การปรับเปลี่ยนหรือขอให้ออกให้ใหม่
4. รอกการอนุมัติจากผู้ให้บริการออกใบรับรอง
5. ผู้ให้บริการออกใบรับรองตรวจสอบ Identity ของผู้ร้องขอ เมื่อได้รับการอนุมัติแล้ว ผู้ให้บริการออกใบรับรองจะทำการรับรองคำร้องนั้นๆ (Sign Request) โดยการสร้างใบรับรองอิเล็กทรอนิกส์ให้กับ Subject
6. ผู้ให้บริการออกใบรับรองเผยแพร่ใบรับรองอิเล็กทรอนิกส์ดังกล่าว เพื่อให้ผู้ร้องขอมารับไป

2.3.2 หน่วยรับลงทะเบียน (Registration Authority – RA)

คือ หน่วยงานหรือบุคคลผู้มีหน้าที่ติดต่อกับ Subject และตรวจสอบความถูกต้องแท้จริงของ Subject ว่าเป็นผู้ที่กล่าวอ้างว่าเป็นจริง เนื่องจาก ในการดำเนินงานจริง ผู้ให้บริการออกใบรับรองอาจแบ่งหน้าที่บางอย่างให้กับหน่วยรับลงทะเบียน ทั้งนี้ ใน RFC 2510 Internet Public Key Infrastructure Certificate Management Protocols อ้างถึงหน้าที่ของหน่วยรับลงทะเบียนไว้ว่า คือ การตรวจสอบความถูกต้องของบุคคล (Personal Authentication) การแจกจ่าย Token (Token Distribution) รายงานการเพิกถอนใบรับรอง (Revocation Reporting) การกำหนดชื่อ (Name Assignment) การสร้างกุญแจคู่ (Key Generation) การจัดเก็บกุญแจคู่ (Archival of Key Pairs) เป็นต้น ในทางปฏิบัติแล้ว หน่วยรับลงทะเบียนจะทำหน้าที่ตรวจสอบความถูกต้องของบุคคลหรือ Subject ใดๆ ระหว่างกระบวนการออกใบรับรองอิเล็กทรอนิกส์ (Certificate Enrollment Process)

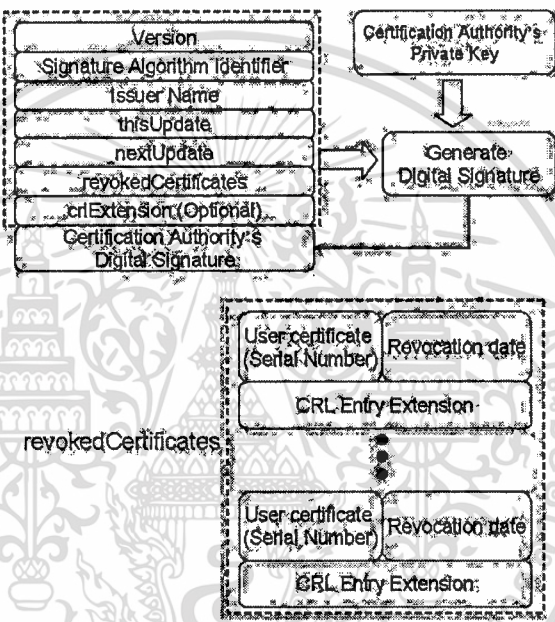
2.3.3 กระบวนการเพิกถอนใบรับรองอิเล็กทรอนิกส์ (Certificate Revocation)

คือ กระบวนการเพิกถอนใบรับรองอิเล็กทรอนิกส์ (Certificate Revocation) เนื่องจากใบรับรองอิเล็กทรอนิกส์หมดอายุหรือถูกผู้อื่นนำไปใช้ วิธีที่นิยมใช้ ได้แก่

2.3.3.1 CRL (Certificate Revocation List)

เอกสารนี้เป็นเอกสารทบทวนเนื้อหาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRL เป็นแฟ้มข้อมูลแสดงรายการของใบรับรองอิเล็กทรอนิกส์ที่ถูกเพิกถอน รูปแบบของ CRL แสดงดังในภาพที่ 2.2 ทั้งนี้ในฟิลด์ข้อมูล revokedCertificates คือ รายการของใบรับรองอิเล็กทรอนิกส์ที่ถูกเพิกถอน แสดงตาม Serial Number ของใบรับรองอิเล็กทรอนิกส์ (ผู้ให้บริการออกใบรับรองจะให้เลข Serial Number กับใบรับรองไม่ซ้ำกัน) ผู้ให้บริการออกใบรับรองจะปรับปรุงรายการ CRL เป็นระยะ ผู้ใช้บริการจะต้องเป็นผู้ดาวน์โหลดเพื่อนำมาปรับปรุงรายการใบรับรองบนเครื่องคอมพิวเตอร์ของตนเอง



ภาพที่ 2.2 รูปแบบของ Certificate Revocation List - CRL

2.3.3.2 OCSF (Online Certificate Status Protocol)

การปฏิบัติงานบนโครงสร้างพื้นฐานกุญแจสาธารณะกับงานบางประเภท ช่วงเวลาระหว่างการปรับปรุง CRL แต่ละครั้งไม่เป็นที่ยอมรับ ดังนั้นจึงเกิด OCSF ขึ้น ซึ่งเป็นกลไกในการตรวจสอบการเพิกถอนของใบรับรองอิเล็กทรอนิกส์แบบเรียลไทม์ ผู้ใช้บริการจะส่ง Certificate Status Check ไปยัง OCSF Responder ที่อาจเป็นผู้ให้บริการออกใบรับรองหรือหน่วย OCSF Responder อื่นเมื่อต้องการใช้งานใบรับรองอิเล็กทรอนิกส์นั้น ทั้งนี้ ใบรับรองอิเล็กทรอนิกส์จะไม่ได้รับการยอมรับจนกว่า OCSF Responder จะตอบ Message มายืนยันหรือปฏิเสธสถานะของใบรับรองอิเล็กทรอนิกส์ดังกล่าว

2.3.4 ระบบจัดเก็บข้อมูลใบรับรองอิเล็กทรอนิกส์ (Certificate Repository)

ใช้สำหรับจัดเก็บและเผยแพร่ใบรับรองอิเล็กทรอนิกส์ให้สาธารณะรับทราบ ในการพัฒนาระบบเล็กๆ ส่วนนี้อาจจะไม่จำเป็น เพราะอาจทำการแลกเปลี่ยนใบรับรองหรือ CRL ผ่านทางจดหมายอิเล็กทรอนิกส์ได้ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขอิเล็กทรอนิกส์ได้ แต่เมื่อต้องเกี่ยวข้องกับใบรับรองจำนวนมาก ก็จำเป็นต้องมีการจัดเก็บให้เป็นไปในทิศทางเดียวกัน ซึ่งอาจจะเลือกใช้ระบบไดเรกทอรี (Directory Services) หรือจัดเก็บในระบบไฟล์ที่สามารถเข้าถึงผ่าน FTP หรือ HTTP ได้

2.3.5 Certificate Management Protocol (CMP)

คือ ขั้นตอนหรือโปรโตคอลในการจัดการใบรับรองอิเล็กทรอนิกส์ โดยระบุถึงขั้นตอนการออกใบรับรองอิเล็กทรอนิกส์ (Certificate Enrollment) ขั้นตอนการเพิกถอนใบรับรอง (Certificate Revocation Process) เวนเดอร์บางแห่งเพิ่มเติมฟังก์ชันการทำงานของโปรโตคอลนอกเหนือจากนี้ เช่น Key recovery และ Automated certificate renewal เป็นต้น

2.3.5.1 Public Key Cryptographic Standard (PKCS)

คือ Certificate Management Protocol ซึ่งเป็นชุดโปรโตคอลมาตรฐานของ RSA Security ที่ระบุวิธีการรักษาความปลอดภัยของการแลกเปลี่ยนข้อมูลระหว่างกัน PKCS #1, #3, #5, #6, #7, #8, #9, #10, #11, #12 และ #15 เป็นมาตรฐานที่มีการใช้งานในปัจจุบัน ส่วน PKCS #13 และ #14 ยังคงอยู่ในระหว่างการอนุมัติ ส่วน PKCS #2 และ #4 ได้ถูกรวมเข้ากับ PKCS #1 แล้ว แต่ละข้อมีรายละเอียดดังนี้

PKCS #1	RSA Cryptography Standard
PKCS #2	รวมเข้ากับ PKCS #1 แล้ว
PKCS #3	Diffie-Hellman Key Agreement Standard
PKCS #4	รวมเข้ากับ PKCS #1 แล้ว
PKCS #5	Password-Based Cryptography Standard
PKCS #6	Extended-Certificate Syntax Standard
PKCS #7	Cryptographic Message Syntax Standard
PKCS #8	Private-Key Information Syntax Standard
PKCS #9	Selected Attribute Types
PKCS #10	Certification Request Syntax Standard
PKCS #11	Cryptographic Token Interface Standard
PKCS #12	Personal Information Exchange Syntax Standard
PKCS #13	Elliptic Curve Cryptography Standard
PKCS #15	Cryptographic Token Information Format Standard

2.4 ใบรับรองอิเล็กทรอนิกส์มาตรฐาน X.509

ในชีวิตประจำวัน เมื่อมีการติดต่อกับบุคคลหรือองค์กรที่ไม่เคยมีความสัมพันธ์กันมาก่อน ถ้าต้องการตรวจสอบสถานะมักจะต้องใช้เอกสารที่เชื่อถือได้เพื่อยืนยันสถานะ เช่น การยืนยันสถานะของบริษัท ใช้เอกสารจากกรมพัฒนาธุรกิจการค้า การยืนยันตัวบุคคลภายในประเทศ ใช้บัตรประชาชน การยืนยันตัวบุคคลระหว่างประเทศ ใช้หนังสือเดินทาง เป็นต้น เอกสารหรือหนังสือเหล่านี้คือตัวอย่างของใบรับรองประเภทต่าง ๆ ที่ใช้รับรองสถานะหรือความสามารถของบุคคลหรือนิติบุคคล สำหรับโลกอิเล็กทรอนิกส์ สามารถทำการตรวจสอบสถานะของผู้ที่ต้องการจะติดต่อได้โดยตรวจสอบจากใบรับรองอิเล็กทรอนิกส์ซึ่งใช้สำหรับการยืนยันตัวบุคคลทางโลกอิเล็กทรอนิกส์ โดยเป็นเอกสารอิเล็กทรอนิกส์ที่ออกให้โดยผู้ให้บริการออกใบรับรอง

ใบรับรองอิเล็กทรอนิกส์ X.509 จะจับคู่ระหว่าง Subject ใดๆ กับกุญแจคู่ ซึ่งประกอบไปด้วยกุญแจสาธารณะ (Public Key) กับกุญแจส่วนตัว (Private Key) โดยกุญแจสาธารณะจะเปิดเผยต่อทุกคน สามารถใช้ในการตรวจสอบข้อมูลที่ได้รับการรับรอง (Sign) โดยกุญแจส่วนตัวหรือเพื่อเข้ารหัสข้อมูลที่จะส่งให้ผู้เป็นเจ้าของกุญแจซึ่งจะมีกุญแจส่วนตัวที่ใช้ถอดรหัสข้อมูลนั้นได้ ผู้เป็นเจ้าของกุญแจจะต้องเก็บรักษากุญแจส่วนตัวที่สามารถใช้เข้ารหัสและรับรองข้อมูล โดยป้องกันจากคนอื่นๆ ที่ไม่ใช่เจ้าของและไม่มีสิทธิในการใช้กุญแจส่วนตัวมิให้ล่วงรู้ได้

ปัจจุบัน มีการใช้งานใบรับรองอิเล็กทรอนิกส์ส่วนมากบนเว็บเบราว์เซอร์ ที่รองรับ Secure Socket Layer - SSL แต่อย่างไรก็ตาม ยังมีเทคโนโลยีอื่นๆ ที่หันมาใช้งานใบรับรองอิเล็กทรอนิกส์เช่นกัน ไม่ว่าจะเป็นการนำไปรับรองแอปพลิเคชันต่างๆ การใช้งานกับจดหมายอิเล็กทรอนิกส์ หรือใช้ร่วมกับการดำเนินธุรกรรมต่างๆ ในพาณิชย์อิเล็กทรอนิกส์

2.4.1 รูปแบบของใบรับรองอิเล็กทรอนิกส์มาตรฐาน X.509

ใบรับรองอิเล็กทรอนิกส์ประกอบด้วยข้อมูลดังนี้

1. เวอร์ชัน (Version) ซึ่งจะบอกได้ถึงส่วนประกอบของข้อมูลในใบรับรอง ปัจจุบันมีทั้งหมด 3 เวอร์ชัน

- X.509 เวอร์ชัน 1 คิดค้นขึ้นในปี ค.ศ. 1988 ปัจจุบันได้รับความนิยมใช้ไม่มากนัก
- X.509 เวอร์ชัน 2 เริ่มใช้แนวคิดของ Subject และ Issuer Unique Identifier เพื่อให้สามารถนำ Subject และ Issuer Name กลับมาใช้ได้ใหม่ แต่เอกสาร Certificate Profile ส่วนใหญ่ไม่แนะนำให้ใช้ Name กลับมาใช้ใหม่และใบรับรองก็ไม่ควรนำ Unique Identifier กลับมาใช้ใหม่ด้วย อย่างไรก็ตาม เวอร์ชันนี้ไม่เป็นที่นิยมนัก
- X.509 เวอร์ชัน 3 เป็นเวอร์ชันล่าสุด (ปี ค.ศ. 1996) และรองรับ Extension เพิ่มเติม ใครก็ตามสามารถออกแบบ Extension และเพิ่มเข้าไปในตัวใบรับรองได้เอง ยกตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่หน่วยงานกลางหรือหน่วยงานที่เกี่ยวข้องออกให้เพื่อใช้ในการยืนยันตัวตน ไม่สามารถแก้ไขเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KeyUsage ซึ่งจำกัดลักษณะการนำไปใช้งาน เช่น กำหนดว่าให้ใช้เฉพาะในการรับรองเพียงอย่างเดียว AlternativeNames คือ การอนุญาตให้ Identity อื่น เช่น DNS name, e-mail address, IP address มาทำงานร่วมกับ Public Key ได้ ทั้งนี้ สามารถระบุไว้ได้ด้วยว่า Extension ใดมีความสำคัญ จำเป็นต้องตรวจสอบและใช้งานด้วย

2. Serial Number ผู้ที่เป็นคนออกใบรับรองจะเป็นผู้กำหนดหรือให้เลข Serial Number ทั้งนี้เพื่อระบุตัวคนที่เป็นหนึ่งเดียวของใบรับรอง แยกให้แตกต่างกับใบรับรองอื่นๆ ที่ผู้ออกใบรับรองได้เคยออกไป ข้อมูลนี้ถูกนำไปใช้ประโยชน์หลายอย่าง เช่น เมื่อมีการออกใบรับรองใหม่ทดแทนใบเดิมก็จะนำ Serial Number เดิมไปไว้บนข้อมูลประกาศสถานะของใบรับรอง นั่นคือ Certificate Revocation List (CRL)

3. Signature Algorithm Identifier เพื่อระบุอัลกอริทึมที่ผู้ให้บริการออกใบรับรองใช้ในการรับรอง (Sign) ใบรับรองอิเล็กทรอนิกส์

4. Issuer Name ชื่อผู้ออกใบรับรองในรูปแบบของ Distinguished Name ตามมาตรฐานไคเรกทอรีปกติก็คือ ผู้ให้บริการออกใบรับรองนั่นเอง

5. Validity Period ใบรับรองอิเล็กทรอนิกส์ทุกใบ จะมีช่วงเวลาจำกัดในการใช้งาน ซึ่งอยู่ในรูปแบบของวันที่-เวลาเริ่มต้นและสิ้นสุดการใช้งาน การกำหนดช่วงเวลานี้ ขึ้นอยู่กับหลายปัจจัย เช่น ความยาวของอายุแก่ที่ใช้ ยิ่งยาวก็จะต้องยากในการลักลอบถอดรหัสข้อมูล ไม่จำเป็นต้องเปลี่ยนบ่อยเป็นต้น

6. Subject Name ชื่อของผู้ที่เป็นเจ้าของใบรับรองอิเล็กทรอนิกส์ จะอยู่ในรูปแบบของ Distinguished Name ตามมาตรฐานไคเรกทอรี

7. Subject Public Key Information คือ ญาแจะสาธารณะกับข้อมูลระบุอัลกอริทึมที่ใช้ในระบบรหัส รวมไปถึงพารามิเตอร์ต่างๆ ของญาแจะ

ทั้งนี้ ข้อมูลทั้งหมดในใบรับรองอิเล็กทรอนิกส์ จะใช้มาตรฐานการจัดข้อมูล 2 มาตรฐาน คือ ASN.1 (Abstract Syntax Notation 1) และ DER (Definite Encoding Rules) โดยมาตรฐานแรกใช้อธิบายข้อมูล ส่วนอันที่สองจะอธิบายวิธีการเก็บและถ่ายโอนข้อมูล

2.5 การประยุกต์ใช้งานใบรับรองอิเล็กทรอนิกส์

2.5.1 ลายมือชื่อดิจิทัล (Digital Signature)

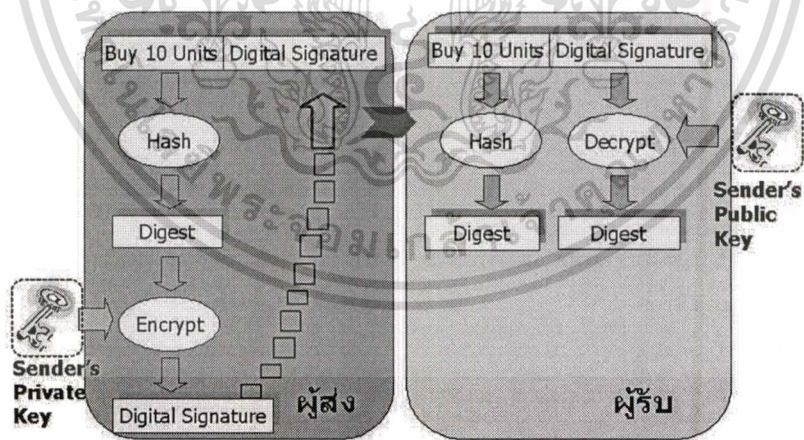
ลายมือชื่อดิจิทัลเป็นข้อมูลในรูปแบบอิเล็กทรอนิกส์ที่สร้างขึ้นโดยใช้สมการทางคณิตศาสตร์ เป็นการนำใบรับรองอิเล็กทรอนิกส์ไปประยุกต์ใช้งานในรูปแบบหนึ่ง มีวัตถุประสงค์เพื่อระบุตัวผู้ส่งข้อมูลอิเล็กทรอนิกส์นั้น ทั้งนี้ การลงลายมือชื่อดิจิทัลจะก่อให้เกิดความปลอดภัยกับ

เอกสารอิเล็กทรอนิกส์เหล่านี้เพื่อเป็นการยืนยันตัวตนของผู้ส่งและผู้รับ โดยผู้ส่งและผู้รับจะต้องมีการยืนยันตัวตนกันก่อน ไม่อย่างนั้นแล้วทั้งผู้ส่งและผู้รับก็อาจถูกหลอกลวงได้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลได้ 3 ประการ คือ ใช้ระบุตัวผู้ส่งข้อมูล (Authentication) ตรวจสอบความถูกต้องของข้อมูล (Data Integrity) และป้องกันไม่ให้มีการปฏิเสธความรับผิดชอบในการส่งหรือรับ (Non-repudiation)

ขั้นตอนในการลงลายมือชื่อดิจิทัลปรากฏตามภาพที่ 2.3 กล่าวคือ ในการสร้างลายมือชื่อดิจิทัลนั้น ผู้ส่งจะนำข้อมูลที่ต้องการส่งมาผ่านกระบวนการย่อข้อมูล (Hash Function) เพื่อให้มีขนาดเล็กลงและสามารถใช้แทนข้อมูลเดิม สิ่งที่ได้จากการย่อข้อมูลเรียกว่า Message Digest จากนั้นจะนำกุญแจส่วนตัวของผู้ส่งมาเพื่อทำกลไกทางคณิตศาสตร์กับ Message Digest ผลลัพธ์ที่ได้จะเป็นลายมือชื่อดิจิทัล ซึ่งส่งไปให้ผู้รับพร้อมกับข้อมูลอิเล็กทรอนิกส์ ข้อสังเกตคือการใช้กุญแจส่วนตัวของผู้ส่งในการสร้างลายมือชื่อดิจิทัล เป็นการยืนยันได้ว่ามาจากผู้ส่งเนื่องจากผู้ส่งคนเดียวเท่านั้นที่มีหรือรู้ถึงกุญแจส่วนตัว

ในการตรวจสอบลายมือชื่อดิจิทัล ผู้รับจะนำข้อมูลอิเล็กทรอนิกส์มาผ่านขั้นตอนการย่อแบบเดียวกับที่ผู้ส่งใช้เพื่อสร้าง Message Digest และนำกุญแจสาธารณะของผู้ส่งมาทำกลไกทางคณิตศาสตร์ (ตรวจสอบ) กับลายมือชื่อดิจิทัลที่แนบมากับข้อมูล จากนั้นนำผลลัพธ์ที่ได้จากกระบวนการทั้งสองอย่างดังกล่าวมาเปรียบเทียบกัน หากเหมือนกันทุกประการก็หมายความว่าข้อมูลที่ส่งมาเป็นข้อมูลที่ถูกส่งมาจากผู้ที่ยืนยันจริง และไม่ได้ถูกเปลี่ยนแปลงหรือแก้ไขใด ๆ (หากมีการเปลี่ยนแปลงข้อมูลอิเล็กทรอนิกส์ของเดิม ตัว Message Digest ที่ได้จะแตกต่างจากเดิม)



ภาพที่ 2.3 กระบวนการสร้างลายมือชื่อดิจิทัล

2.5.2 การเข้ารหัสลับข้อมูล (Data Encryption)

ในการเข้ารหัสลับข้อมูล ข้อมูลปกติ (Plain Text) จะได้รับการเปลี่ยนแปลงให้อยู่ในรูปที่ไม่สามารถอ่านเข้าใจและเปลี่ยนรูปกลับไม่ได้หากไม่มีกุญแจที่ใช้ร่วมในการเข้าและถอดรหัสลับ

ผู้ใช้แต่ละคนจะมีคู่กุญแจ กล่าวคือ กุญแจสาธารณะ (Public Key) และกุญแจส่วนตัว (Private Key) ในการใช้งานนั้น สามารถใช้กุญแจสาธารณะเพื่อการเข้ารหัสลับข้อมูล และใช้กุญแจส่วนตัวเพื่อการถอดรหัสลับข้อมูลได้ ซึ่งลักษณะนี้จะเป็นระบบรหัสแบบอสมมาตร

นอกจากนั้น ยังสามารถใช้ประโยชน์จาก Session Key ซึ่งใช้สำหรับการเข้าและถอดรหัสลับข้อมูลในระบบรหัสแบบสมมาตรได้ สามารถสร้างขึ้นเองจากแอปพลิเคชันโดยใช้ฟังก์ชันใน Cryptography API ของไมโครซอฟท์และจะจัดเก็บไว้ในที่เฉพาะเพื่อรักษาให้ปลอดภัยได้ สิ่งที่แตกต่างกันไปจากคู่กุญแจ (Key Pair) คือ Session Key นั้น เปลี่ยนแปลงในทุกครั้งที่มีการเรียกใช้งาน ในการสร้างแต่ละครั้งจะได้ Session Key ใหม่เสมอ แอปพลิเคชันสามารถเก็บ Session Key เหล่านี้สำหรับการใช้งานต่อหรือส่งไปยังผู้อื่นได้ โดยการนำออก (Export) จากที่เก็บมายังแอปพลิเคชัน ในรูปของ Encrypted Key Binary Large Object หรือ Key Blob ได้

คำจำกัดความ

- Message ส่วนของข้อมูล ประกอบไปด้วยข้อความแบบ ASCII เพิ่มข้อมูลของระบบฐานข้อมูล หรือข้อมูลอื่นๆ
- Plain Text ข้อมูลที่ยังมิได้ผ่านการเข้ารหัสลับ
- Cipher Text ข้อมูลที่ผ่านการเข้ารหัสลับแล้ว

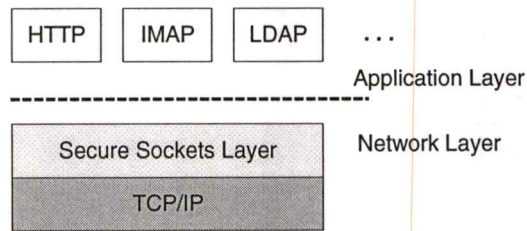


ภาพที่ 2.4 การเข้าและถอดรหัสลับข้อมูล

2.5.3 Secure Sockets Layer (SSL V3.0) Protocol

SSL เป็นโพรโตคอลที่พัฒนาโดยบริษัทเน็ตสเคป (Netscape) ใช้เพื่อสร้างความปลอดภัยในการสื่อสารระหว่างเว็บเซิร์ฟเวอร์และไคลเอ็นต์ผ่านเครือข่ายอินเทอร์เน็ต การทำงานโดยย่อคือ มีการใช้กุญแจสาธารณะในการเข้ารหัสข้อมูลแล้วส่งผ่านข้อมูลไปบนการเชื่อมต่อ SSL เว็บเบราว์เซอร์อย่างอินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer) และเน็ตสเคป (Netscape) รองรับ SSL ในปัจจุบัน หลายเว็บไซต์จะรับข้อมูลที่เป็นความลับของผู้ใช้ผ่าน SSL สังเกตได้ง่ายคือเว็บไซต์เหล่านั้นจะใช้ “https” แทนที่จะเป็น “http”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 SSL ทำงานบน TCP/IP และอยู่ใต้ Application layer

จุดประสงค์หลักของ SSL คือ เพื่อรักษาความเป็นส่วนตัวและความน่าเชื่อถือในการสื่อสารระหว่างไคลเอ็นท์และเซิร์ฟเวอร์ ป้องกันมิให้เกิดปัญหาการลักลอบอ่านหรือเปลี่ยนแปลงแก้ไขข้อมูล โปรโตคอลนี้ประกอบไปด้วย 2 เลเยอร์ คือ

1. **SSL Record Protocol** อยู่ในเลเยอร์ต่ำกว่าติดกับ TCP ทำหน้าที่ห่อหุ้ม (Encapsulate) โปรโตคอลอื่นๆ ที่สูงกว่าและระบุรูปแบบของการรับส่งข้อมูล
2. **SSL Handshake Protocol** ซึ่งทำให้ไคลเอ็นท์และเซิร์ฟเวอร์สามารถตรวจสอบสิทธิของกันและกัน (Authentication) และตกลงอัลกอริทึมในการเข้ารหัสที่จะใช้ รวมถึงกุญแจที่ใช้ในระบบรหัสก่อนที่จะตกลงแลกเปลี่ยนข้อมูลกันได้ ประโยชน์ข้อหนึ่งของ SSL คือ ไม่ขึ้นต่อการทำงานของเลเยอร์ที่สูงกว่า (Application Protocol Independent) กล่าวคือ โปรโตคอลใดๆ ในเลเยอร์ที่สูงกว่าสามารถทำงานอยู่บน SSL ได้

2.5.3.1 คุณสมบัติของการสื่อสารผ่าน SSL

- การเชื่อมต่อเป็นส่วนตัว หลังจากตกลงกันถึงกุญแจ (Secret key) ที่จะใช้แล้ว จะมีการเข้ารหัสข้อมูล ซึ่งจะใช้ระบบรหัสแบบสมมาตร เช่น DES หรือ RC4 เป็นต้น
- การระบุช่องทางสื่อสาร (Peer's Identity) จะมีการตรวจสอบสิทธิโดยใช้ระบบรหัสแบบอสมมาตร เช่น RSA หรือ DSS เป็นต้น
- การเชื่อมต่อมีความน่าเชื่อถือ ข้อมูลที่ส่งไปจะมี Message Integrity Check ที่ใช้ Keyed MAC การคำนวณเกี่ยวกับ MAC นี้จะใช้ Secure Hash Function เช่น SHA หรือ MD5 เป็นต้น

2.5.3.2 ความสามารถในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต

- **SSL Server Authentication** ทำให้ผู้ใช้ (ไคลเอ็นท์) สามารถตรวจสอบตัวตนของเซิร์ฟเวอร์ได้ โดยใช้เทคนิคมาตรฐานของกุญแจสาธารณะ ตรวจสอบว่าใบรับรองอิเล็กทรอนิกส์ของเซิร์ฟเวอร์ใช้งานได้และออกให้โดยผู้ประกอบการรับรองที่ผู้ใช้ให้เอกสารนี้เป็นเอกสารที่สวามิฉันชอบสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเชื่อถืออยู่เช่นกัน ความสามารถนี้จำเป็นในการส่งข้อมูลที่เป็นความลับจากผู้ใช้ไปยังเซิร์ฟเวอร์

- **SSL Client Authentication** ทำให้เซิร์ฟเวอร์สามารถตรวจสอบตัวตนของผู้ใช้ได้ ใช้เทคนิคเดียวกันกับข้อที่แล้ว โดยตรวจสอบใบรับรองอิเล็กทรอนิกส์ของไคลเอ็นท์ใช้งานได้ และออกให้โดยผู้ประกอบการรับรองที่เซิร์ฟเวอร์ให้ความเชื่อถืออยู่เช่นกัน ความสามารถนี้จำเป็นในการส่งข้อมูลที่เป็นความลับจากเซิร์ฟเวอร์ไปยังผู้ใช้
- **Encrypted SSL Connection** ทำให้ข้อมูลที่รับส่งระหว่างเซิร์ฟเวอร์และไคลเอ็นท์ที่ได้รับการเข้ารหัส ใช้ซอฟต์แวร์ที่ฝั่งส่งและรับ สนับสนุนให้เกิดรักษาความลับของข้อมูลได้ดี นอกจากนี้ ยังได้รับการปกป้องจากกลไกที่ตรวจจับได้ว่าข้อมูลได้รับการปรับเปลี่ยนระหว่างการรับส่งหรือไม่

2.5.3.3 Cipher ที่ใช้กับ SSL

SSL รองรับการใช้งาน Cipher หรืออัลกอริทึมระบบรหัสหลายแบบ เพื่อใช้ในขั้นตอนการตรวจสอบตัวตนเซิร์ฟเวอร์และไคลเอ็นท์ การส่งใบรับรองอิเล็กทรอนิกส์และการตั้งกุญแจสำหรับ Session ทั้งนี้ อาจแบ่งออกคร่าวๆ ได้ 2 รูปแบบ คือ Cipher suites with RSA Key Exchange และ FORTEZZA Cipher suites

2.5.3.3.1 Cipher suites with RSA Key Exchange รองรับใน SSL เวอร์ชัน 2 และ 3

ส่วนที่ใช้เฉพาะในสหรัฐอเมริกาเท่านั้น

- Triple DES รองรับการใช้รหัสด้วยกุญแจ 168 บิต และ SHA-1 Message Authentication
 - RC4 รองรับการใช้รหัสด้วยกุญแจ 128 บิตและ MD5 Message Authentication
 - RC2 รองรับการใช้รหัสด้วยกุญแจ 128 บิตและ MD5 Message Authentication
 - DES รองรับการใช้รหัสด้วยกุญแจ 56 บิตและ SHA-1 Message Authentication
- ส่วนที่สามารถนำไปใช้นอกสหรัฐอเมริกาได้
- RC4 รองรับการใช้รหัสด้วยกุญแจ 40 บิตและ MD5 Message Authentication
 - RC2 รองรับการใช้รหัสด้วยกุญแจ 40 บิตและ MD5 Message Authentication
- ส่วนที่มีความแข็งแกร่งน้อยที่สุด

- ไม่มีการเข้ารหัสและใช้ MD5 Message Authentication

2.5.3.3.2 FORTEZZA Cipher suites รองรับใน SSL เวอร์ชัน 3

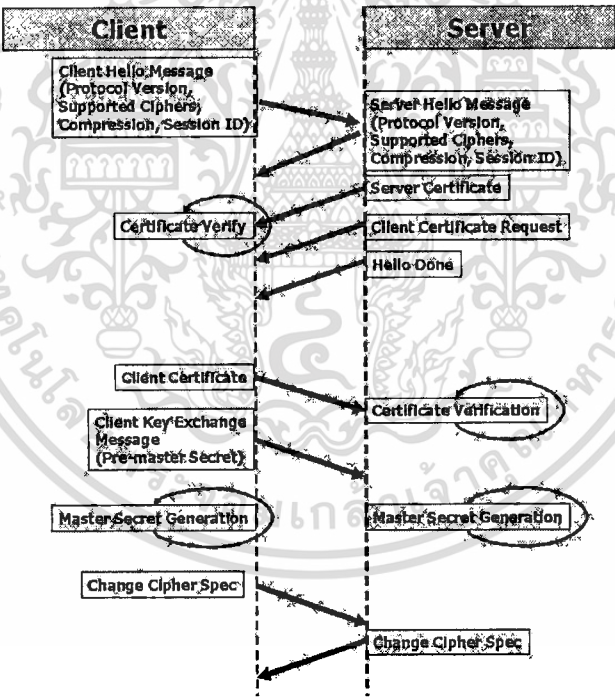
เป็นระบบการเข้ารหัสที่ใช้โดยรัฐบาลสหรัฐอเมริกาเพื่อจัดการข้อมูลที่เป็นความลับแต่ไม่ได้จำแนกประเภท ทั้งนี้ FORTEZZA ใน SSL ใช้ KEA (Key Exchange Algorithm) แทน RSA Key Exchange และใช้ FORTEZZA Card และ DSA สำหรับการตรวจสอบฟังก์ชันโคลเอ็นท์ ส่วนที่ใช้เฉพาะในสหรัฐอเมริกาเท่านั้น

- RC4 รองรับการเข้ารหัสด้วยกุญแจ 128 บิตและ SHA-1 Message Authentication
- RC4 รองรับการเข้ารหัสด้วยกุญแจ SKIPJACK 80 บิตและ SHA-1 Message Authentication

ส่วนที่มีความแข็งแกร่งน้อยที่สุด

- ไม่มีการเข้ารหัสและใช้ SHA-1 Message Authentication เท่านั้น

2.5.3.4 การทำงานของ SSL ในขั้นตอนการทำ Handshake



ภาพที่ 2.6 SSL Handshake

จากภาพที่ 2.6 สามารถอธิบายได้ดังนี้

- โคลเอ็นท์ส่งหมายเลขเวอร์ชันของ SSL, Cipher setting, ข้อมูลที่ผู้มสรสร้างขึ้นและข้อมูลอื่นๆ ที่เซิร์ฟเวอร์ต้องใช้ระหว่างการสื่อสารต่อไปยังเซิร์ฟเวอร์

- เซิร์ฟเวอร์ส่งหมายเลขเวอร์ชันของ SSL, Cipher setting, ข้อมูลที่สุ่มสร้างขึ้นและข้อมูลอื่นๆ ที่ไคลเอ็นต์ต้องการใช้ระหว่างการสื่อสาร ทั้งนี้ ยังส่งไปรับรองอิเล็กทรอนิกส์ของเซิร์ฟเวอร์ให้ด้วยและหากไคลเอ็นต์ร้องขอ
- ไคลเอ็นต์ใช้ข้อมูลบางอย่างที่ส่งมาจากเซิร์ฟเวอร์ในการตรวจสอบตัวตนเซิร์ฟเวอร์ หากไม่สามารถตรวจสอบได้ ผู้ใช้จะได้รับการเตือนและแจ้งให้ทราบว่าการเชื่อมต่อแบบที่มีการตรวจสอบตัวตนก่อนและมีการเข้ารหัสไม่สามารถทำได้ และข้ามไปทำงานดังข้อ 4
- ไคลเอ็นต์สร้าง Premaster secret สำหรับ Session แล้วเข้ารหัสด้วยกุญแจสาธารณะของเซิร์ฟเวอร์ (ได้จากใบรับรองอิเล็กทรอนิกส์ของเซิร์ฟเวอร์ ที่ส่งมาให้ในข้อ 2) และส่ง Premaster secret ที่เข้ารหัสแล้วไปยังเซิร์ฟเวอร์ ทั้งนี้ จะทำได้โดยอาศัยข้อมูลที่สร้างมาจากขั้นตอนข้างต้น และความร่วมมือจากเซิร์ฟเวอร์ ขึ้นอยู่กับ Cipher ที่ใช้
- ถ้าเซิร์ฟเวอร์ร้องขอที่จะตรวจสอบตัวตนของไคลเอ็นต์ด้วย ไคลเอ็นต์จะรับรองข้อมูลส่วนหนึ่งที่ Unique ใน Handshake และทั้งเซิร์ฟเวอร์และไคลเอ็นต์รู้จัก ในกรณีนี้ ไคลเอ็นต์จะส่งทั้งข้อมูลที่รับรองแล้วและใบรับรองอิเล็กทรอนิกส์ของไคลเอ็นต์ไปยังเซิร์ฟเวอร์พร้อมกับ Premaster secret ที่เข้ารหัสแล้วด้วย
- ถ้าเซิร์ฟเวอร์ร้องขอที่จะตรวจสอบตัวตนของไคลเอ็นต์ แล้วไม่สามารถทำได้ Session จะจบลง แต่ถ้าสามารถทำได้ เซิร์ฟเวอร์จะใช้กุญแจส่วนตัวถอดรหัส Premaster secret จากนั้นจะเกิดขึ้นขั้นตอนมาอีกหลายข้อ ซึ่งจะทำการทั้งบนเซิร์ฟเวอร์และไคลเอ็นต์ ทั้งนี้เพื่อสร้าง Master secret ขึ้น
- ทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะใช้ Master secret สร้าง Session key ซึ่งเป็นกุญแจเดียวที่ใช้สำหรับเข้ารหัสและถอดรหัสข้อมูลที่แลกเปลี่ยนกันผ่าน SSL Session และใช้เพื่อตรวจสอบความถูกต้องข้อมูลด้วย คือตรวจหาว่าข้อมูลเปลี่ยนแปลงไปบ้างหรือไม่ระหว่างการส่ง
- ไคลเอ็นต์ส่ง Message มายังเซิร์ฟเวอร์เพื่อแจ้งว่าข้อมูลที่จะส่งหลังจากนี้ไปจะได้รับการเข้ารหัสด้วย Session key จากนั้นจะส่ง Message อีกส่วนที่เข้ารหัสไว้ไปเพื่อบอกว่าส่วนการทำ Handshake ของไคลเอ็นต์สิ้นสุดลงแล้ว
- เซิร์ฟเวอร์ส่ง Message ไปยังไคลเอ็นต์แจ้งว่าข้อมูลที่จะส่งหลังจากนี้ไปจะได้รับการเข้ารหัสด้วย Session key จากนั้นจะส่ง Message อีกส่วนที่เข้ารหัสไว้ไปเพื่อบอกว่าส่วนการทำ Handshake ของเซิร์ฟเวอร์สิ้นสุดลงแล้ว
- SSL handshake สิ้นสุดลง และ SSL session ได้เริ่มต้นแล้ว ไคลเอ็นต์และเซิร์ฟเวอร์จะใช้ Session key เพื่อเข้ารหัสและถอดรหัสข้อมูลที่รับส่งระหว่างกันและเพื่อตรวจสอบความถูกต้อง

ขั้นตอนการแลกเปลี่ยนรหัสลับใช้เวลาพอสมควร ซึ่งโปรโตคอล SSL ก็มีคุณสมบัติที่ช่วยลดขั้นตอนในการทำงานนี้ลงไปได้ ถ้ามีเว็บเบราว์เซอร์ตัวใหม่ติดต่อไปยังเว็บเซิร์ฟเวอร์ตัวเดิมที่ทำงานกับ SSL ได้ เว็บเบราว์เซอร์อาจเพียงส่งแค่หมายเลขประจำตัวของช่องการสื่อสารให้กับเซิร์ฟเวอร์ และถ้าเซิร์ฟเวอร์นั้นยอมรับหมายเลขประจำตัวที่ส่งมา การติดต่อระหว่างเซิร์ฟเวอร์ตัวเดิมกับเว็บเบราว์เซอร์ตัวใหม่ก็สามารถใช้อัลกอริทึมเดิมในการเข้ารหัสและบีบอัดข้อมูล รวมไปถึงรหัสเดิมที่เคยตกลงกันไว้ก่อนหน้านี้ โดยไม่ต้องทำขั้นตอนแลกเปลี่ยนรหัสครั้งใหม่เลย เว็บเบราว์เซอร์ของเน็ต สเคปยังสามารถส่งข้อความที่ชื่อ "Keep Alive" ที่เป็นคำร้องขอให้เปิดช่องการสื่อสาร TCP ค้างเอาไว้สักพักแม้ว่าจะแลกเปลี่ยนข้อมูลเสร็จสิ้นแล้วก็ตาม ทำให้การสื่อสารครั้งใหม่ที่จะเกิดขึ้นในอนาคตไม่จำเป็นต้องเสียเวลากับการแลกเปลี่ยนรหัสซ้ำอีก

2.6 อัลกอริทึมระบบรหัสแบบสมมาตร (Cryptographic Algorithm)

อัลกอริทึมระบบรหัสแบบสมมาตร (Symmetric Algorithm) นั้นเป็นอัลกอริทึมที่ใช้งานในการเข้ารหัสลับโดยทั่วไปที่ใช้เทคโนโลยีระบบรหัสแบบสมมาตร กล่าวคือใช้ กุญแจเดียวสำหรับการทั้งการเข้ารหัสและถอดรหัสลับ มีการเปลี่ยนแปลงกุญแจที่ใช้บ่อยครั้ง กล่าวได้ว่า สามารถสร้างขึ้นเองเพื่อใช้งานเฉพาะการสื่อสารนั้นแล้วทิ้งไปได้ เรียกได้ว่าเป็น Session Key มีความรวดเร็ว เหมาะสำหรับการเข้ารหัสลับข้อมูลจำนวนมาก ยกตัวอย่างอัลกอริทึม เช่น RC2 RC4 หรือ DES เป็นต้น

อัลกอริทึมระบบรหัสแบบสมมาตรมีอยู่ด้วยกัน 2 ประเภท ได้แก่ Block Cipher และ Stream Cipher

- Block Cipher เป็นการทำงานกับ Block ของ Plain Text และ Cipher Text โดยปกติ Block ดังกล่าวจะมีความยาว 64 บิต แต่ก็อาจจะยาวมากกว่านั้นก็ได้ ทั้งนี้ เมื่อเกิดการเข้ารหัสลับ Plain Text Block ใดๆ แล้ว Cipher Text Block ที่ได้ หากนำมาถอดรหัสลับกลับก็จะยังคงได้ Plain Text Block อันเดิม
- Stream Cipher เป็นการทำงานกับ Plain Text และ Cipher Text แบบแปลง Plain Text ให้เป็น Cipher Text ทีละบิต

2.7 เทคโนโลยี Microsoft Cryptography API

Microsoft Cryptography API ใช้เพื่อสร้างความปลอดภัยให้กับข้อมูลอิเล็กทรอนิกส์โดยอาศัยเทคโนโลยีกุญแจสาธารณะ (Public Key Security) มี Service Provider Model ในเรื่องระบบรหัส (Cryptography) โดยสามารถเลือกใช้อัลกอริทึมได้หลากหลาย สนับสนุนความยาวของกุญแจหลายขนาด และสามารถจัดการใบรับรองอิเล็กทรอนิกส์ได้ (Certificate Management)

เอกสารนี้เป็นเอกสารที่เผยแพร่เพื่อใช้ในการศึกษาเท่านั้น ไม่ควรใช้เพื่อวัตถุประสงค์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cryptography API ประกอบไปด้วยฟังก์ชันที่ช่วยให้สามารถสร้างแอปพลิเคชันสำหรับการเข้ารหัสลับหรือสร้างลายมือชื่อดิจิทัลให้กับ (Sign) ข้อมูลได้ในหลายรูปแบบ โดยป้องกันข้อมูลกุญแจส่วนตัวให้เป็นความลับได้ด้วย ทั้งนี้ Operation ต่างๆ ในด้าน Cryptographic ทุกอย่างจะแยกออกเป็นโมดูล เรียกว่า Cryptographic Service Providers (CSP) ซึ่งอย่างหนึ่งที่รวมไว้ในระบบปฏิบัติการของไมโครซอฟท์ คือ Microsoft RSA Base Provider

CSP แต่ละตัวมีการอิมพลีเมนต์ Cryptography API Layer ที่ต่างกัน บางอย่างใช้อัลกอริทึมที่แข็งแรง (Strong) มาก บางอย่างสามารถใช้กับฮาร์ดแวร์ได้ เช่น บัตรสมาร์ทการ์ด บางครั้งก็ใช้ติดต่อกับผู้ใช้โดยตรง เช่นการสร้างลายมือชื่อดิจิทัลให้กับข้อมูลโดยการใช้งานกุญแจส่วนตัวของผู้ใช้นั่นเอง

2.7.1 สถาปัตยกรรมของ CryptoAPI 2.0



ภาพที่ 2.7 สถาปัตยกรรมของ CryptoAPI

2.7.2 บริการใน CryptoAPI

- การสร้างและบริหารจัดการกุญแจ
- การสร้าง Hash
- การสร้างลายมือชื่อดิจิทัล (Digital Signature) และตรวจสอบความถูกต้อง (Verification) ของลายมือชื่อดิจิทัล
- การแลกเปลี่ยนกุญแจ (Key Exchange)

เอกสารนี้เป็นเอกสารการเข้ารหัสลับข้อมูล (Bulk Encryption / Decryption) เสนอแนะให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 หน้าที่ของ Cryptographic Service Provider

- ทำงานโดยใช้อัลกอริทึมและความแข็งแกร่งของกุญแจต่างๆ กัน
- เลือกได้ว่าจะทำงานกับฮาร์ดแวร์หรือซอฟต์แวร์

2.7.4 Microsoft Software Provider

- Provider พื้นฐาน (Base Provider) คือ 512 Bit RSA และ 40 Bit RC2/RC4
- Advanced Provider คือ มากกว่า 512 Bit RSA, 128 Bit RC2/RC4, DES และ 3DES
- DSS Provider และ D-H Provider

2.7.5 การจัดการใบรับรอง (Certificate Handling)

- จัดการใบรับรองอิเล็กทรอนิกส์มาตรฐาน X 509 เวอร์ชัน 3 และ Certificate Chain
- การอ่านข้อมูลในใบรับรองอิเล็กทรอนิกส์ (ASN.1)

2.7.6 ที่จัดเก็บใบรับรอง (Certificate Storage)

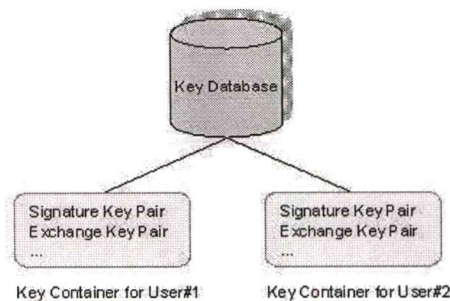
- เป็นที่เก็บ (Repository) ของใบรับรองผู้ให้บริการออกใบรับรองที่เป็น Root และอื่นๆ
- เป็นที่เก็บข้อมูลใบรับรองของผู้ใช้
- จัดการจับคู่ระหว่างใบรับรองและคู่กุญแจ

2.7.7 รูปแบบข้อมูลมาตรฐานสำหรับออกใบรับรองอิเล็กทรอนิกส์

- PKCS #10 เป็นรูปแบบข้อมูลขอใบรับรอง (Certificate Request)
- PKCS #7 ซึ่งเป็นข้อมูลใบรับรองและ Chain ของใบรับรอง
- PKCS #12 ซึ่งเป็นรูปแบบการสำรองจัดเก็บใบรับรองและคู่กุญแจ

2.7.8 ระบบฐานข้อมูลสำหรับจัดเก็บกุญแจ (Key Database)

Cryptographic Service Provider แต่ละตัวจะมี Key Database สำหรับเก็บ Persistent Cryptographic Key ซึ่งในแต่ละ Key Database อาจจะมี Key Container มากกว่าหนึ่งก็ได้ ใน Key Container จะมี Key Pair ของผู้ใช้แต่ละคน ซึ่งมี Unique Name ประจำ Key Container



ภาพที่ 2.8 Key Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.9 ฟังก์ชันเกี่ยวกับระบบรหัสของไมโครซอฟท์ (Cryptographic Function)

ฟังก์ชันต่างๆ เกี่ยวกับระบบรหัสใน Wincrypt.h ทำให้สามารถสร้างแอปพลิเคชันที่จำเป็นต้องเกี่ยวข้องกับระบบรหัสได้ง่ายขึ้น และช่วยอำนวยความสะดวกในการติดต่อกับ Cryptographic Service Provider (CSP) แสดงดังในตารางที่ 2.1

ตารางที่ 2.1 ฟังก์ชันระบบรหัสของไมโครซอฟท์

ลำดับ	ฟังก์ชัน	คำอธิบาย
1	CryptAcquireContext	เข้าถึง Key Container ภายใน CSP ที่ต้องการ
2	CryptContextAddRef	เพื่อ Reference Count ในการ Handle HCRYPTPROV
3	CryptCreateHash	เริ่มต้นการ Hash ข้อมูล
4	CryptDecrypt	ถอดรหัสลับข้อมูล
5	CryptEnumProviderTypes	ประเภทของ Crypt ที่ระบบปฏิบัติการบนคอมพิวเตอร์รองรับ
6	CryptExportKey	ทำการ Export กุญแจออกจาก CSP อย่างปลอดภัย
7	CryptFormatObject	จัดรูปแบบของข้อมูลที่ถูกเข้ารหัส ให้เป็น Unicode String ใน Buffer (Buffer ตาม Certificate Encoding Type)
8	CryptGenKey	สร้างกุญแจด้วยการสุ่มสำหรับใช้งานกับ CSP Module
9	CryptGenRandom	สร้าง Buffer ขนาดตามที่สุ่มขึ้น
10	CryptGetDefaultProvider	บอก CSP Default ของผู้ใช้งานปัจจุบันหรือคอมพิวเตอร์ (แต่ละ Provider Type)
11	CryptGetHashParam	ดึงข้อมูลเกี่ยวกับการควบคุมกระบวนการ Hash
12	CryptGetKeyParam	ดึงข้อมูลเกี่ยวกับการควบคุมกระบวนการจัดการ Key
13	CryptGetProvParam	ดึงพารามิเตอร์ที่ควบคุมกระบวนการของ CSP
14	CryptGetUserKey	ควบคุม Permanent User Key Pair เช่น User's Signature Key Pair
15	CryptHashData	คำนวณ Hash ให้กับข้อมูลใดๆ
16	CryptHashSessionKey	คำนวณ Hash ให้กับ Key Object
17	CryptImportKey	ถ่ายโอน Cryptographic key จาก Key Blob มายัง CSP
18	CryptProtectData	เข้ารหัสลับข้อมูลใน DATA_BLOB นำไปใช้ประโยชน์ด้านการค้า

19	CryptQueryObject	รับค่า CERT_BLOB เป็น File Name จากนั้นคืนค่าเป็นข้อมูลเกี่ยวกับ Content ของไฟล์นั้น
20	CryptReleaseContext	ยกเลิกการควบคุม CSP และ Key Container
21	CryptSetHashParam	ดัดแปลงกระบวนการของ Hash Object แต่ยังสามารถใช้งานได้ไม่สมบูรณ์นัก (ตอนนี้รับพารามิเตอร์ได้ค่าเดียว)
22	CryptSetKeyParam	ดัดแปลง Key Operation
23	CryptSetProvider	ระบุ Default CSP ของผู้ใช้
24	CryptSetProviderEx	กำหนด Default CSP ของผู้ใช้หรือของคอมพิวเตอร์
25	CryptSetProvParam	ดัดแปลงกระบวนการ CSP
26	CryptSignHash	สร้างลายมือชื่อดิจิทัลให้กับข้อมูล (Sign)
27	CryptUnprotectData	ถอดรหัสลับและทำ Integrity Check ของข้อมูลใน DATA_BLOB
28	CryptVerifySignature	ตรวจสอบลายมือชื่อดิจิทัลจาก Hash Object ไว้

บทที่ 3

การติดตั้ง ปรับแต่ง และทดสอบใช้งานระบบผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์

เทคโนโลยีกุญแจสาธารณะ (Public Key Infrastructure - PKI) เป็นเทคโนโลยีที่ได้รับการยอมรับในวงกว้าง แต่ปัญหาที่พบคือ ระบบโครงสร้างพื้นฐานเพื่อรองรับเทคโนโลยีกุญแจสาธารณะแบบองค์กรที่จัดจำหน่ายในเชิงพาณิชย์มีราคาค่อนข้างสูง จึงมีบุคคลบางกลุ่มที่พยายามสร้างทางเลือกอื่น โดยการสร้างซอฟต์แวร์แบบ Open Source อย่าง OpenCA หรือ OpenSSL เพื่อสร้างระบบผู้ให้บริการออกใบรับรอง (Certificate Authority - CA) เอง ทั้งนี้ แนวโน้มในปัจจุบันหน่วยงานไอทีต่างๆ เริ่มหันมาสนับสนุนการทำงานด้วยซอฟต์แวร์ Open Source เพื่อลดค่าใช้จ่ายขององค์กรกันมากขึ้น

3.1 การวางแผนปฏิบัติงาน

ได้เลือกใช้ระบบปฏิบัติการและซอฟต์แวร์ดังต่อไปนี้ (สามารถอ่านวิธีการติดตั้งได้จากภาคผนวก ก ขั้นตอนการติดตั้งระบบ)

1. ระบบปฏิบัติการ เลือกใช้ Linux Redhat 7.2

เป็นระบบปฏิบัติการแบบยูนิกซ์ที่ได้รับความนิยมอย่างแพร่หลาย ออกแบบไว้สำหรับการใช้งานบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ต้องการใช้ระบบปฏิบัติการที่ไม่ต้องเสียค่าใช้จ่าย มีส่วนติดต่อผู้ใช้แบบกราฟฟิก เช่น X Window System รวมทั้งส่วนประกอบอื่นๆ ที่จะพบในระบบปฏิบัติการแบบยูนิกซ์ทั่วไป เช่น TCP/IP เป็นต้น

2. ซอฟต์แวร์ผู้ให้บริการออกใบรับรองเลือกใช้ OpenCA เวอร์ชัน 0.9.1

OpenCA Lab เกิดจาก OpenCA Project ซึ่งเป็นองค์กรที่มุ่งจะพัฒนากรอบงาน (Framework) สำหรับการศึกษาเกี่ยวกับ PKI และการพัฒนาโครงการที่เกี่ยวข้อง แต่เนื่องจากมีผู้ให้ความสนใจมาตรฐาน PKI กันมากและเพิ่มจำนวนขึ้นอย่างรวดเร็ว ผู้ก่อตั้งจึงตัดสินใจที่จะแยกโครงการเดิมออกเป็นโครงการย่อยที่เล็กลงเพื่อความรวดเร็วในการพัฒนาและจัดโครงสร้าง ทั้งนี้บางโครงการก็ได้เริ่มต้นและรับการวิจัยไปแล้ว ในขณะที่บางโครงการยังคงอยู่ระหว่างการตัดสินใจขั้นสุดท้าย

โครงการที่อยู่ระหว่างการดำเนินงานในขณะนี้โครงการหนึ่ง คือ PKI Development เป็นความร่วมมือในการสร้างซอฟต์แวร์ผู้ให้บริการออกใบรับรองแบบ

Open Source ที่พร้อมใช้งาน โดยใช้งานโปรโตคอลและระบบรหัสที่ใช้งานกันทั่วโลก ทั้งนี้ OpenCA มีพื้นฐานมาจากซอฟต์แวร์ Open Source หลายตัวด้วยกัน ไม่ว่าจะเป็น OpenLDAP, OpenSSL, Apache Project หรือ Apache_mod_ssl ทั้งนี้ได้แบ่งการพัฒนาภายในโครงการออกเป็น 2 ส่วนได้แก่ การศึกษาและสรุปโครงสร้างด้านความปลอดภัยที่เหมาะสมจะเป็นโมเดลที่ดีที่สุดสำหรับผู้ให้บริการออกใบรับรองและการพัฒนาซอฟต์แวร์ที่สามารถติดตั้งและจัดการให้เป็นผู้ให้บริการออกใบรับรองได้โดยง่าย

3. ซอฟต์แวร์ SSL/TLS เลือกใช้ OpenSSL เวอร์ชัน 0.9.6

โครงการ OpenSSL เป็นความพยายามในการพัฒนาทูลคิดแบบ Open Source ที่มีประสิทธิภาพเทียบเท่าผลิตภัณฑ์เชิงพาณิชย์ โดยมีการอิมพลีเมนต์โปรโตคอล Secure Sockets Layer (SSL v2/v3) และ Transport Layer Security (TLS v1) รวมไปถึงไลบรารีระบบรหัสต่างๆ (Cryptography) ซึ่งโครงการนี้ได้รับความร่วมมือจากอาสาสมัครทั่วโลกที่มาจากเครือข่ายอินเทอร์เน็ต ในการสื่อสาร วางแผนและพัฒนาทูลคิด OpenSSL และเอกสารอื่นๆ ที่เกี่ยวข้อง OpenSSL มีพื้นฐานมาจากไลบรารี SSLeay ที่พัฒนาโดย Eric A. Young และ Tim J. Hudson มีการจดลิขสิทธิ์ภายใต้รูปแบบเดียวกับ Apache สามารถนำไปใช้ได้ทั้งในเชิงพาณิชย์และไม่ใช้

4. เว็บเซิร์ฟเวอร์ที่รองรับ SSL/TLS เลือกใช้ Apache เวอร์ชัน 1.3.27

Apache HTTP Server เป็นซอฟต์แวร์สำหรับให้บริการเว็บเซิร์ฟเวอร์ (HTTP/Web Server) ผ่านทางโปรโตคอล HTTP โดยเป็นซอฟต์แวร์แบบ Open Source สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่าย เนื่องจากกลุ่มอาสาสมัครจากทั่วโลกได้ร่วมกันพัฒนามาจนกระทั่งในปัจจุบันเป็นเวอร์ชันที่ 2.0.43 แล้ว

5. ไดรกทอรีเซิร์ฟเวอร์มาตรฐาน LDAP เลือกใช้ OpenLDAP เวอร์ชัน 2.0.25

เป็นซอฟต์แวร์ Open Source ที่พัฒนาอ้างอิงจากมาตรฐาน Lightweight Directory Access Protocol (LDAP) โดยในชุดของซอฟต์แวร์ประกอบไปด้วย slapd คือ stand-alone LDAP Server, slurpd คือ stand-alone LDAP replication Server และไลบรารีที่ใช้สำหรับติดต่อผ่านโปรโตคอล LDAP รวมไปถึงเครื่องมือต่างๆ และตัวอย่างไคลเอ็นท์

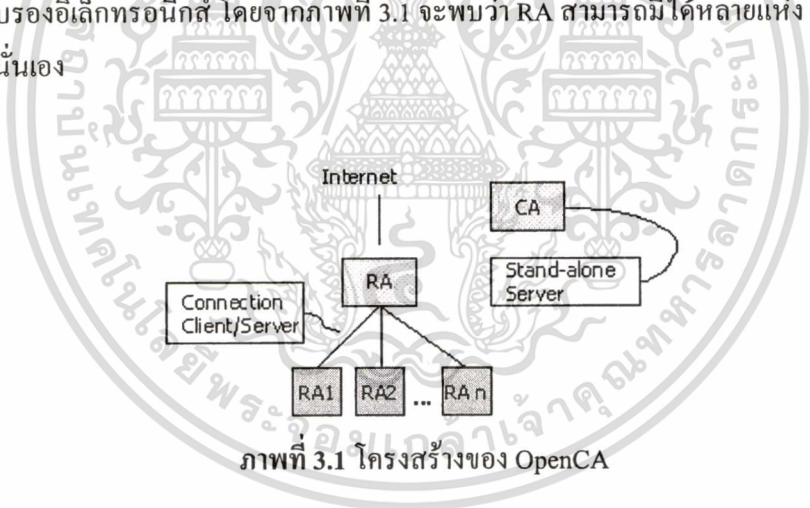
Slapd เป็นเดมอน stand-alone LDAP ซึ่งจะคอยรับการเชื่อมต่อผ่านโปรโตคอล LDAP ผ่านพอร์ตใดๆ (โดยปกติคือ 389) ตอบสนองต่อ LDAP Operation ต่างๆ ที่ได้รับโดยปกติแล้ว slapd จะเริ่มทำงานตอนเครื่องเริ่มทำงาน ทั่วไปอยู่นอกเหนือจาก /etc/rc.local หลังจากเริ่มต้นทำงาน slapd จะ Fork และ Disassociate ตัวเองจาก Invoking tty

ส่วน slurpd เป็นเดมอน stand-alone LDAP Replication Server ดูแลเกี่ยวกับการทำซ้ำข้อมูลระหว่าง LDAP Server (Replication) กล่าวคือ เมื่อมีการสร้าง LDAP Server ให้ทำงานพร้อมกัน 2 เซิร์ฟเวอร์ขึ้นไป แล้วได้ตั้งให้เซิร์ฟเวอร์ 2 เครื่องนั้นมีการทำซ้ำข้อมูลระหว่างกันตลอดเวลา เมื่อเกิดการอัปเดตที่เครื่องแม่ เครื่องลูกก็จะต้องเกิดการอัปเดตข้อมูลตาม

3.2 การวางแผนโครงสร้างของระบบ

จากภาพที่ 3.1 โครงสร้างของ OpenCA ส่วนที่เป็น CA จะแยกออกมาเป็นเซิร์ฟเวอร์ที่ไม่เชื่อมต่อกับเซิร์ฟเวอร์อื่นหรือเครือข่ายอินเทอร์เน็ต เนื่องจากต้องการความปลอดภัยสูงเพราะเป็นเซิร์ฟเวอร์ที่ติดตั้งซอฟต์แวร์ของ CA และเก็บกุญแจส่วนตัว (Private Key) ของ CA ไว้ด้วย

เซิร์ฟเวอร์ RA จะมีความซับซ้อนมากกว่า กล่าวคือ จะมีเว็บเซิร์ฟเวอร์และไคลเอนต์เซิร์ฟเวอร์รันอยู่ โดยอย่างแรกจะทำหน้าที่รองรับ Request จาก Subject ใดๆ ในการขอใบรับรองอิเล็กทรอนิกส์ ก่อนที่ RA จะเป็นผู้นำไปให้ CA รับรองต่อ ส่วนอย่างหลัง จะทำหน้าที่เก็บและเผยแพร่ใบรับรองอิเล็กทรอนิกส์ โดยจากภาพที่ 3.1 จะพบว่า RA สามารถมีได้หลายแห่ง ทั้งนี้เพื่อแบ่งเบาภาระนั่นเอง

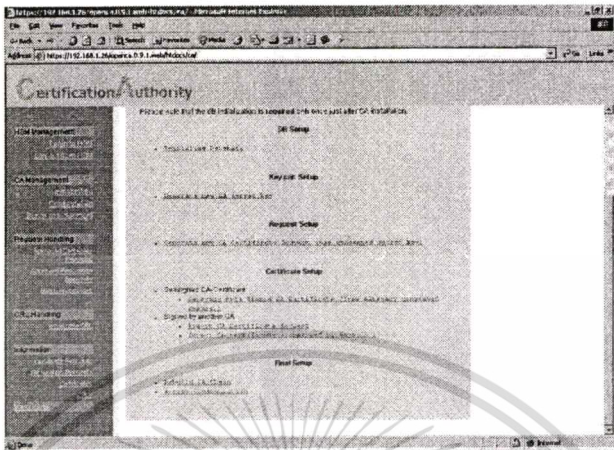


ภาพที่ 3.1 โครงสร้างของ OpenCA

3.3 การวางแผนโมเดลการรับรอง (Trust Model)

ในการสร้างผู้ให้บริการออกใบรับรองนั้น หากเป็นการสร้างเพื่อใช้กันภายในองค์กรเองแล้ว CA นั้นก็ย่อมจะได้รับความไว้วางใจจากบุคลากรในองค์กร แต่หากบุคคลในองค์กรต้องการที่จะติดต่อสื่อสารกับบุคคลภายนอกโดยใช้ใบรับรองอิเล็กทรอนิกส์ที่ออกให้โดยองค์กรของตน บุคคลภายนอกอาจไม่รู้จักและไม่ให้ความไว้วางใจ CA ภายใน ดังนั้น บางครั้ง การรับรองถูกต้องแท้จริงสร้างเองก็ไม่ใช่ที่ยอมรับ จำเป็นต้องขอให้ CA อื่นที่เป็นรู้จักและได้รับความไว้วางใจมากกว่ารับรองให้ ดังในภาพที่ 3.2 Root CA จะเป็นผู้รับรองให้ Subordinate CA1 และ 2 ใครก็ตามที่มอบ

- จากภาพที่ 3.3 ให้เลือก Phase 1 คลิกที่ปุ่ม Initialize ระบบจะแสดงหน้าจอ ดังภาพที่ 3.4



ภาพที่ 3.4 แสดงหน้าจอแรกหลังจากการ Initialize เซิร์ฟเวอร์ผู้ให้บริการออกใบรับรอง

- เมื่อคลิกเลือกสร้างฐานข้อมูล (Initialize Database) ระบบจะทำการสร้างฐานข้อมูลใหม่ และเมื่อเสร็จสิ้น ระบบจะแสดงหน้าจอ ดังภาพที่ 3.5



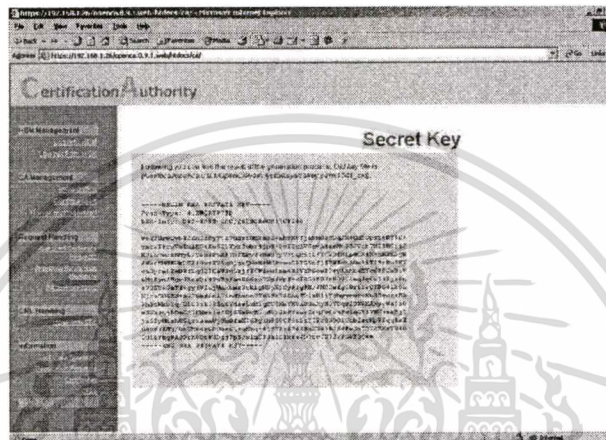
ภาพที่ 3.5 แสดงหน้าจอหลังจาก Initialize Database เสร็จสิ้นแล้ว

- จากนั้น ผู้ให้บริการออกใบรับรองได้สร้างกุญแจส่วนตัว (Private Key) และ Certificate Request และทำการสร้างใบรับรองอิเล็กทรอนิกส์ต่อไป จากภาพที่ 3.4 เมื่อคลิกปุ่มสร้างกุญแจส่วนตัว (Generate new CA secret key) จะปรากฏหน้าต่างให้ระบุรายละเอียดของกุญแจดังนี้
 - Encryption Algorithm : DES3

- Key Length : 1024 Bit
- รหัสผ่านเพื่อเข้าถึงกุญแจส่วนตัว (Private Key) คือ “qaz1234”
- ผลลัพธ์ที่ได้ ไฟล์ Secret Key : cakey.pem จัดเก็บไว้ที่

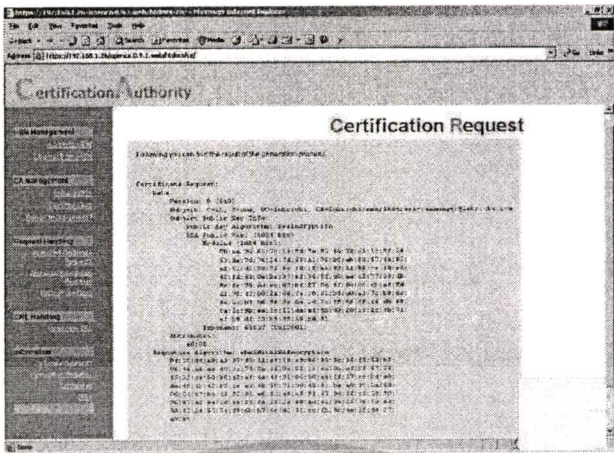
/usr/local/openssl/OpenCA/var/crypto/keys/cakey.pem

เมื่อระบบสร้างกุญแจส่วนตัวเสร็จสิ้น ระบบจะแสดงหน้าจอดังในภาพที่ 3.6



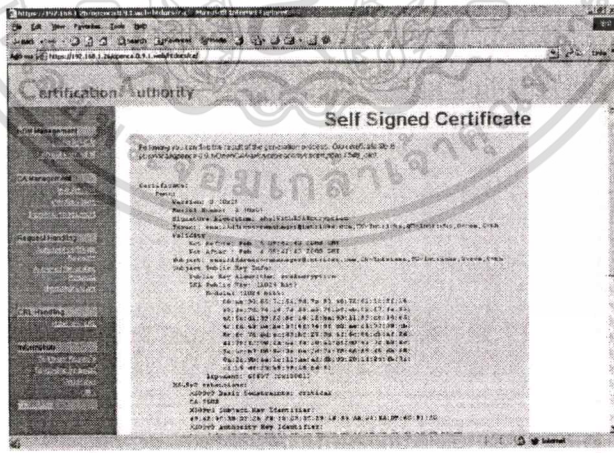
ภาพที่ 3.6 แสดงหน้าจอหลังจากสร้างคู่กุญแจแล้ว

- จากนั้น จากภาพที่ 3.4 เมื่อคลิกปุ่มสร้าง Certificate Request จะปรากฏหน้าต่างให้ระบุรายละเอียดของ Certificate Request ดังนี้
 - e-mail Address : camanager@intricks.com
 - CommonName : Intricks
 - Organizational Unit : Intricks
 - Organization : com
 - ISO 3166 Country Code : th
 - Distinguished Name :
emailAddress=camanager@intricks.com,CN=Intricks,OU=Intricks,O=com,C=th
 - รหัสผ่านคือ “qaz1234”
 - ผลลัพธ์ที่ได้ ไฟล์ Request : /usr/local/openssl/OpenCA/var/crypto/reqs/careq.pem
- หลังจากสร้าง Certificate Request เสร็จสิ้น ระบบจะแสดงหน้าจอดังในภาพที่ 3.7



ภาพที่ 3.7 แสดงหน้าจอหลังจากสร้าง Certificate Request

- จากนั้น จากภาพที่ 3.4 เมื่อคลิกปุ่มสร้าง Self-Signed Certificateจะปรากฏหน้าต่างให้ระบุรายละเอียดต่างๆ ของใบรับรองอิเล็กทรอนิกส์ที่จะสร้าง ดังนี้
 - อายุของใบรับรอง 1825 วัน (CA Certificate Validity (In days from now) : 1825)
 - กรอกรหัสผ่าน “qaz1234”
 - ผลลัพธ์ที่ได้ - ไฟล์ cacert.der, cacert.pem : จัดเก็บไว้ที่ /usr/local/openca/OpenCA/var/crypto/cacerts
- จากนั้น ระบบจะแสดงหน้าจอใบรับรองอิเล็กทรอนิกส์ ดังในภาพที่ 3.8



ภาพที่ 3.8 แสดงหน้าจอหลังจากสร้าง Certificate แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การประยุกต์ใช้งาน

3.5.1 สร้างใบรับรองอิเล็กทรอนิกส์

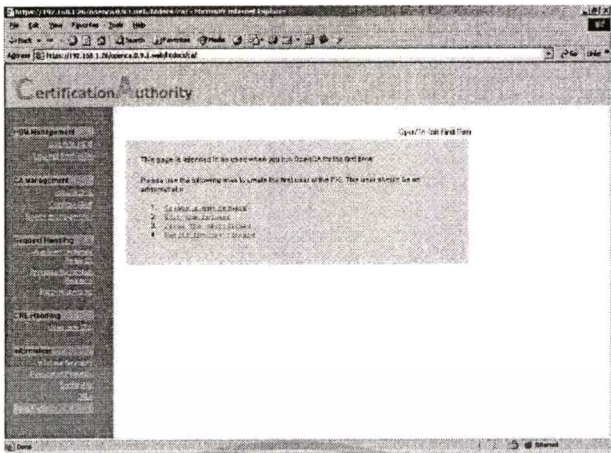
- ผู้ที่ปฏิบัติงานเป็น CA/RA Operator จะต้องได้รับใบรับรองอิเล็กทรอนิกส์ที่ได้รับการรับรอง (Sign) จากผู้ให้บริการออกใบรับรองใหม่นี้ด้วยเพื่อนำไปใช้งานต่อไป
- ใบรับรองของ Operator นี้จะต้องถูก Import นำไปใช้งานร่วมกับเว็บเบราว์เซอร์ ทั้งนี้เพื่อระบุตัวตนกับผู้ให้บริการออกใบรับรอง

3.5.2 การให้บริการของผู้ใช้ทั่วไป

- ผู้ใช้เปิดโปรแกรมเว็บเบราว์เซอร์แล้วเชื่อมต่อมายังหน้าเว็บของเซิร์ฟเวอร์ผู้ให้บริการออกใบรับรอง
- RA Operator ทำการตรวจสอบ Certificate Request เป็นระยะเพื่อทำการอนุมัติการร้องขอใบรับรองอิเล็กทรอนิกส์ผ่านเว็บเบราว์เซอร์ โดยตรวจสอบสิทธิด้วยใบรับรองของ RA Operator เอง
- RA Operator ทำการ Export ตัว Request ไปสู่ดิสก์เก็ต
- นำดิสก์เก็ตนั้นไปยังเซิร์ฟเวอร์ผู้ให้บริการออกใบรับรองเพื่อ Import เข้าสู่เซิร์ฟเวอร์เพื่อทำการอนุมัติ
- มีการสร้าง CRL ใหม่หากมีการออกใบรับรองใหม่หรือเปลี่ยนสถานะใบรับรองเก่า
- ทำการ Export ใบรับรองและ CRL ไปยังดิสก์เก็ตเพื่อนำไป Import เข้าสู่ RA Server อีกครั้งหนึ่งเพื่อให้ผู้ใช้สามารถดาวน์โหลดไปใช้งานได้

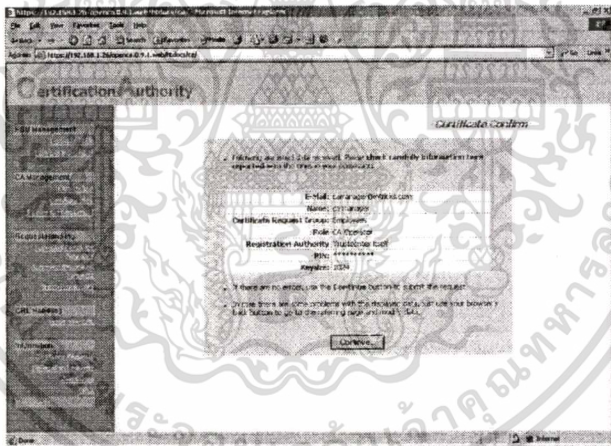
การออกใบรับรองอิเล็กทรอนิกส์ให้กับผู้ใช้ ไม่ว่าทำหน้าที่ด้านใด จะมีขั้นตอนการทำงานคล้ายกัน ซึ่งจะอธิบายได้ดังต่อไปนี้

- จากภาพที่ 3.3 เมื่อคลิก Phase 2 ระบบจะแสดงหน้าจอดังในภาพที่ 3.9 ระบุขั้นตอนการขอใบรับรอง ได้แก่
 - สร้าง Certificate Request ใหม่
 - แก้ไข Certificate Request
 - ออกใบรับรองอิเล็กทรอนิกส์
 - จัดการเกี่ยวกับใบรับรองอิเล็กทรอนิกส์



ภาพที่ 3.9 แสดงหน้าจอแรกระบบขั้นตอนการขอใบรับรอง

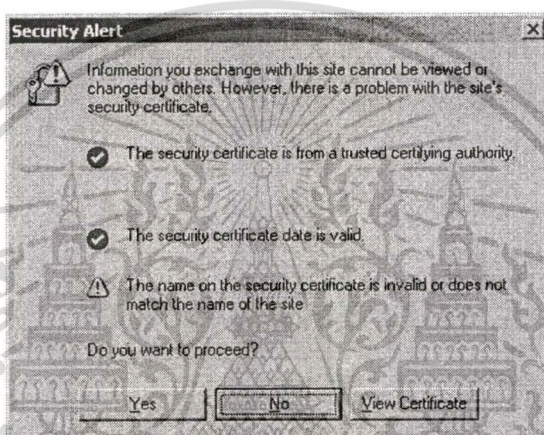
- เมื่อผู้ใช้ทำการกรอกแบบฟอร์มเรียบร้อยแล้ว ระบบจะแสดงหน้าจอเพื่อยืนยันความถูกต้องของข้อมูลอีกครั้งหนึ่ง ดังแสดงในภาพที่ 3.10



ภาพที่ 3.10 แสดงหน้าจอหลังจากที่ผู้กรอกแบบฟอร์มขอใบรับรองแล้ว

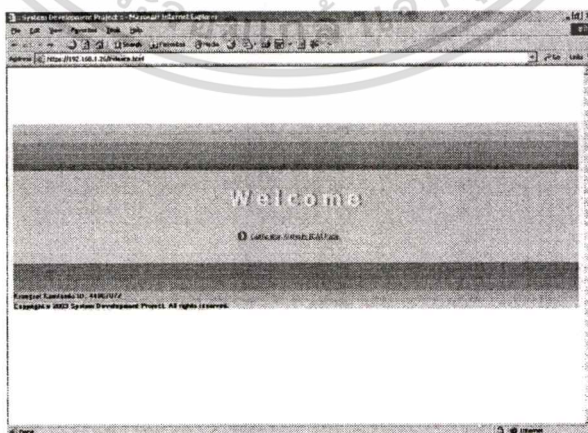
- จากนั้น ระบบจะแสดงหน้าจอ Certificate Request ที่รอการอนุมัติ (Pending Request) จากเจ้าหน้าที่ออกใบรับรอง ดังแสดงในภาพที่ 3.11

- **Client Authentication** คือ การยืนยันความถูกต้องแท้จริงของผู้ใช้ด้วยใบรับรองอิเล็กทรอนิกส์ของผู้ใช้จากเครื่องไคลเอนท์
 ทั้งนี้ การยืนยันตัวบุคคลเพื่อเข้าใช้งานระบบผู้ให้บริการออกใบรับรอง (เฉพาะผู้ใช้ที่เป็นผู้ออกใบรับรอง – CA Manager และ RA Manager) สามารถอธิบายขั้นตอนการทำงานได้ ดังนี้
- เมื่อเปิดเว็บไซต์ที่มีใบรับรองอิเล็กทรอนิกส์จะปรากฏหน้าจอเพื่อแจ้งกับไคลเอนท์ว่าเว็บไซต์นี้มีการรับรองไว้ด้วยใบรับรองอิเล็กทรอนิกส์ และถามว่าไคลเอนท์ยอมรับใบรับรองนี้หรือไม่ แสดงในภาพที่ 3.13



ภาพที่ 3.13 หน้าจอแรกเมื่อเปิดเว็บไซต์

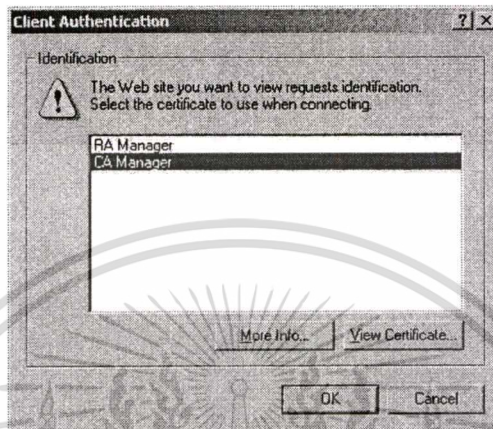
- เมื่อฝั่งไคลเอนท์ยอมรับ ให้ความเชื่อถือใบรับรองอิเล็กทรอนิกส์ของเซิร์ฟเวอร์ด้วยการคลิกปุ่ม Yes ในภาพที่ 3.13 ก็จะสามารรถเรียกดูหน้าเว็บได้ตามปกติ ดังแสดงในภาพที่ 3.14



ภาพที่ 3.14 หน้าจอแรกหลังจากยอมรับการใช้งานใบรับรองจากเซิร์ฟเวอร์

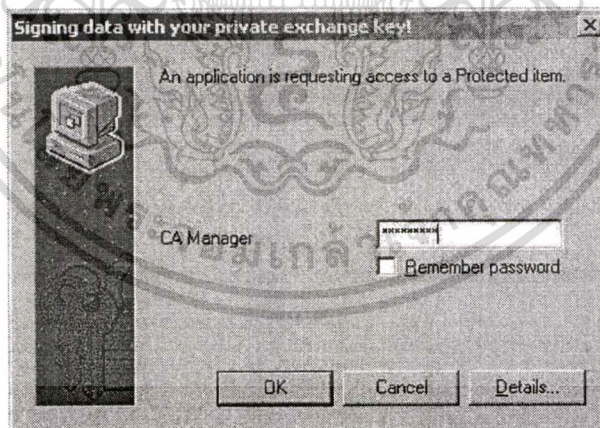
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อคลิกที่ลิงค์ Certification Authority หากได้กำหนดค่าให้เซิร์ฟเวอร์ตรวจสอบใบรับรองของไคลเอ็นต์ด้วย จะปรากฏหน้าจอให้เลือกใบรับรองของไคลเอ็นต์ ที่ไคลเอ็นต์ต้องการใช้เพื่อยืนยันความถูกต้องของตนเอง ดังแสดงในภาพที่ 3.15



ภาพที่ 3.15 หน้าจอแสดงรายชื่อใบรับรองฝั่งไคลเอ็นต์

- จากนั้น จะปรากฏหน้าจอให้ผู้ใช้ในการกรอกรหัสผ่านการเข้าถึงกุญแจส่วนตัวของคุณ ในที่นี้คือ CA Manager ดังภาพที่ 3.16



ภาพที่ 3.16 สำหรับกรอกรหัสผ่านเข้าถึงกุญแจส่วนตัว

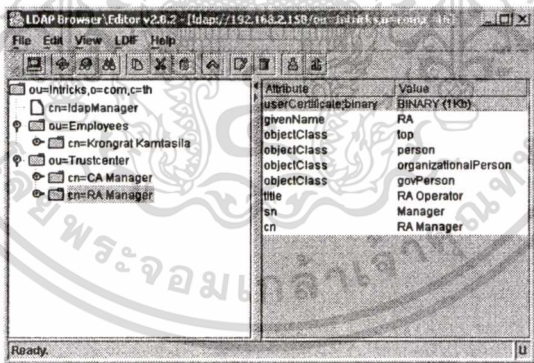
- จากภาพที่ 3.16 หากกรอกรหัสผ่านถูกต้อง แล้วคลิกปุ่ม OK จะสามารถเปิดเข้าสู่หน้าเว็บไซต์ได้ตามปกติ

3.6 การใช้งานระบบไคเรกทอรี

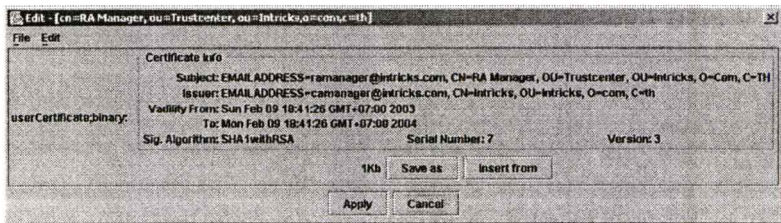
- หลังจากติดตั้ง OpenLDAP เรียบร้อยแล้ว ทำการปรับเปลี่ยนไฟล์คอนฟิก (slapd.conf) โดยทำการเปลี่ยนแปลง BaseDN, RootDN, RootDN Password, Server Address และ Port ที่จะใช้งาน ดังนี้

```
BaseDN    ou=Intricks,o=com,c=th
RootDN    cn=ldapManager,ou=Intricks,o=com,c=th
Port      389
```

- จากนั้น กำหนดไฟล์ Schema (รูปแบบของข้อมูลบนระบบไคเรกทอรี) ที่จะใช้งาน เช่น core.schema, cosine.schema หรือ inetorgperson.schema
- ภาพที่ 3.17 แสดงหน้าจอโปรแกรม LDAP Browser/Editor ที่เชื่อมต่อเข้าสู่ไคเรกทอรีเซิร์ฟเวอร์เพื่อทดสอบการเรียกดูข้อมูล
- ภาพที่ 3.18 แสดงหน้าจอเมื่อดับเบิลคลิกที่ปุ่ม userCertificate;binary: เพื่อแสดงรายละเอียดของใบรับรองอิเล็กทรอนิกส์ ทั้งนี้ ผู้ใช้สามารถเชื่อมต่อเข้าสู่ระบบไคเรกทอรีเพื่อดาวน์โหลดใบรับรองอิเล็กทรอนิกส์ของบุคคลอื่น ผ่านโปรแกรมรับส่งอีเมลอย่าง Outlook Express หรือ Netscape Messenger ได้ด้วย โดยการตั้งค่าผ่านส่วน Address Book

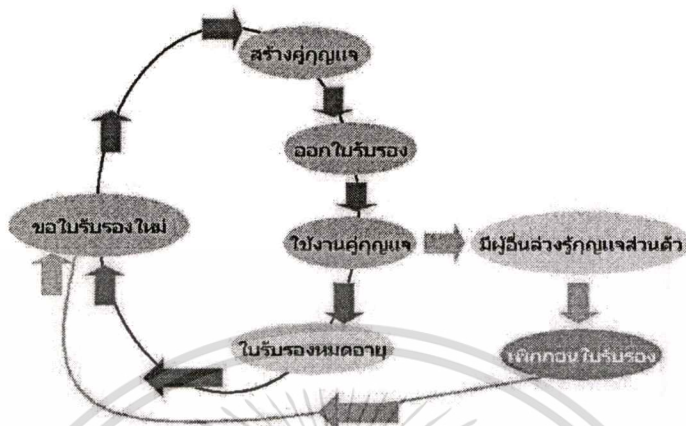


ภาพที่ 3.17 แสดงหน้าจอโปรแกรม LDAP Browser/Editor



ภาพที่ 3.18 แสดงหน้าจอแสดงรายละเอียดใบรับรองอิเล็กทรอนิกส์

3.7 ผลการใช้งาน



ภาพที่ 3.19 วงจรชีวิตของใบรับรองอิเล็กทรอนิกส์

จากภาพที่ 3.19 สามารถระบุซอฟต์แวร์ที่ใช้ในการทำงานขั้นตอนต่างๆ ได้ดังนี้

- **ขอใบรับรองใหม่** ผู้ใช้จะขอใบรับรองใหม่ผ่านเว็บอินเทอร์เฟซของ OpenCA
- **สร้างคู่กุญแจ** จากนั้น OpenCA จะเรียกใช้งานฟังก์ชันระบบรหัสของ OpenSSL ในการสร้างคู่กุญแจและสร้าง Certificate Request
- **ออกใบรับรอง** OpenCA จะเรียกใช้งานฟังก์ชันระบบรหัสของ OpenSSL ในการสร้างลายมือชื่อดิจิทัลให้กับกุญแจสาธารณะของผู้ใช้ กล่าวคือการออกใบรับรองนั่นเอง
- **จัดเก็บใบรับรองลงสู่ระบบฐานข้อมูลและระบบไคลเอนต์** ของผู้ให้บริการออกใบรับรอง เมื่อทำการออกใบรับรองให้กับผู้ใช้แล้ว ผู้ให้บริการออกใบรับรองจะจัดเก็บใบรับรองดังกล่าวลงสู่ระบบฐานข้อมูลและระบบไคลเอนต์ ซึ่งใช้ประกาศข้อมูลใบรับรอง
- **ใช้งานคู่กุญแจ** ผู้ใช้นำใบรับรองและกุญแจส่วนตัวไปใช้งาน
- **มีผู้อื่นลวงรู้กุญแจส่วนตัว** หากมีผู้อื่นลวงรู้กุญแจส่วนตัว ผู้ใช้จะต้องแจ้งผู้ให้บริการออกใบรับรองเพื่อยกเลิกการใช้งานใบรับรองนั้น
- **ใบรับรองหมดอายุ** หากมีใบรับรองหมดอายุการใช้งาน ผู้ใช้จะต้องแจ้งผู้ให้บริการออกใบรับรองเพื่อยกเลิกการใช้งานใบรับรองนั้น
- **ขอเพิกถอนใบรับรอง** เมื่อผู้ใช้ขอยกเลิกการใช้งานใบรับรอง ผู้ให้บริการออกใบรับรองจะทำการประกาศยกเลิกใบรับรองดังกล่าวไว้ใน Certificate Revocation List - CRL และจัดเก็บไว้บนระบบไคลเอนต์ ในขั้นตอนนี้ OpenCA จะเรียกใช้ฟังก์ชันระบบรหัสบน OpenSSL

หลังจากติดตั้งระบบผู้ให้บริการออกใบรับรองแล้ว ผู้พัฒนาสามารถปฏิบัติงานเป็นผู้ให้บริการออกใบรับรองอิเล็กทรอนิกส์แก่บุคคลในองค์กรได้เอง เป็นการเลือกใช้ซอฟต์แวร์ Open Source ที่ไม่เสียค่าใช้จ่ายแต่ต้องอาศัยความเชี่ยวชาญระบบยูนิกซ์และการใช้งานซอฟต์แวร์ต่างๆ อย่างไรก็ตาม ผู้ให้บริการต้องคำนึงถึงเรื่องนโยบายการปฏิบัติงานด้วย เช่น เอกสารข้อตกลงระหว่างผู้ใช้งานและผู้ให้บริการ การระบุขั้นตอนทำงานที่ชัดเจน การจัดการกับข้อผิดพลาดที่อาจเกิดขึ้นจากตัวบุคคล ซอฟต์แวร์หรือคอมพิวเตอร์ เป็นต้น นอกจากนี้ ผู้พัฒนาสามารถสร้าง PKI-Enabled Application เป็นโปรแกรมประยุกต์เพื่อใช้งานในด้านต่างๆ เช่น การแนบลายมือชื่อดิจิทัลไปกับเอกสารหรือธุรกรรมใดๆ ทางเครือข่ายคอมพิวเตอร์เพื่อยืนยันตัวผู้ส่ง การเข้ารหัสเอกสารหรือข้อมูลใดๆ ที่ส่งระหว่างทรานส์แอ็กชันต่างๆ บนเครือข่ายคอมพิวเตอร์เพื่อรักษาความลับของข้อมูล เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การวิเคราะห์และออกแบบโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล

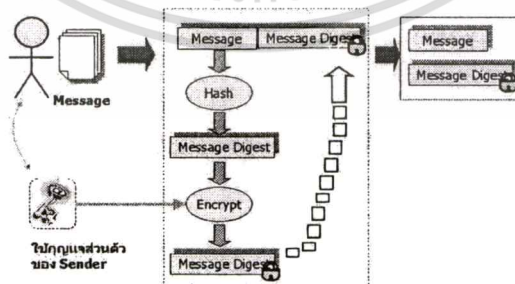
4.1 ขั้นตอนการทำงานของโปรแกรม

ได้แบ่งการทำงานของโปรแกรมออกเป็น 6 ข้อ ดังนี้

- 1 การสร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูล
- 2 การตรวจสอบความถูกต้องของลายมือชื่อดิจิทัลที่แนบกับแฟ้มข้อมูล
- 3 การเข้ารหัสลับแฟ้มข้อมูล
- 4 การถอดรหัสลับแฟ้มข้อมูล
- 5 การสร้างลายมือชื่อดิจิทัลและเข้ารหัสลับแฟ้มข้อมูล
- 6 การตรวจสอบลายมือชื่อดิจิทัลและถอดรหัสลับแฟ้มข้อมูล

4.1.1 การสร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูล แสดงในภาพที่ 4.1

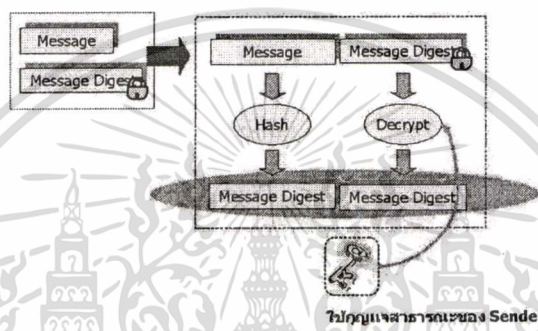
- เลือกแฟ้มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ โดยติดต่อกับที่จัดเก็บใบรับรอง (Microsoft Certificate Store) จากนั้นแสดงใบรับรองอิเล็กทรอนิกส์เรียงลำดับตามชื่อ
- นำข้อมูลผ่านการ Hash เพื่อสร้างเป็น Message Digest
- เข้ารหัสลับ Message Digest ทั้งนี้ ข้อมูลที่ได้จากการเข้ารหัสลับ Message Digest นี้ คือ ลายมือชื่อดิจิทัล



ภาพที่ 4.1 การสร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูล

4.1.2 การตรวจสอบความถูกต้องของลายมือชื่อดิจิทัลที่แนบกับแฟ้มข้อมูล แสดงในภาพที่ 4.2

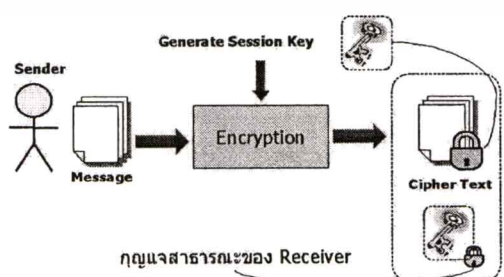
- เลือกเพิ่มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ โดยติดต่อกับที่จัดเก็บใบรับรอง (Microsoft Certificate Store) จากนั้นแสดงใบรับรองอิเล็กทรอนิกส์ เรียงลำดับตามชื่อ
- ใช้กุญแจสาธารณะของผู้ส่งถอดรหัสลายมือชื่อดิจิทัลที่แนบมากับเพิ่มข้อมูล ได้ข้อมูล Message Digest
- สร้าง Message Digest จากการ Hash ข้อมูลต้นฉบับ
- นำข้อมูล Message Digest มาเทียบกับค่า Message Digest ใหม่ที่ได้



ภาพที่ 4.2 การตรวจสอบลายมือชื่อดิจิทัลของเพิ่มข้อมูล

4.1.3 การเข้ารหัสลับเพิ่มข้อมูล แสดงในภาพที่ 4.3

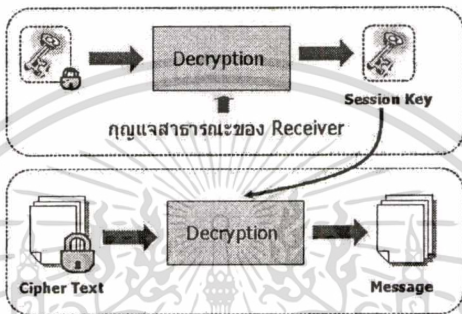
- เลือกเพิ่มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ โดยติดต่อกับที่จัดเก็บใบรับรอง (Microsoft Certificate Store) จากนั้นแสดงใบรับรองอิเล็กทรอนิกส์ เรียงลำดับตามชื่อ
- สร้าง Session Key ที่จะใช้ในการเข้ารหัสลับเพิ่มข้อมูล
- เข้ารหัสลับเพิ่มข้อมูลด้วย Session Key
- เข้ารหัสลับ Session Key ด้วยกุญแจสาธารณะของผู้รับ



ภาพที่ 4.3 การเข้ารหัสลับเพิ่มข้อมูล

4.1.4 การถอดรหัสลับเพิ่มข้อมูล แสดงในภาพที่ 4.4

- เลือกเพิ่มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ โดยติดต่อกับที่จัดเก็บใบรับรอง (Microsoft Certificate Store) จากนั้นแสดงใบรับรองอิเล็กทรอนิกส์ เรียงลำดับตามชื่อ
- ใช้กุญแจสาธารณะถอดรหัสลับ Session Key
- ใช้ Session Key ถอดรหัสลับเพิ่มข้อมูล



ภาพที่ 4.4 การถอดรหัสลับเพิ่มข้อมูล

4.1.5 การสร้างลายมือชื่อดิจิทัลและเข้ารหัสลับเพิ่มข้อมูล

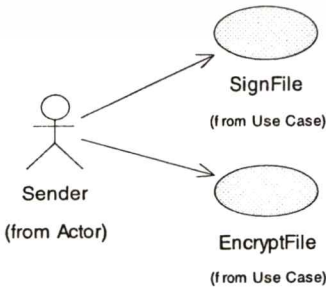
ขั้นตอนประกอบไปด้วยขั้นตอนในข้อที่ 4.1.1 และ 4.1.3 ตามลำดับ

4.1.6 การตรวจสอบลายมือชื่อดิจิทัลและถอดรหัสลับเพิ่มข้อมูล

ขั้นตอนประกอบไปด้วยขั้นตอนในข้อที่ 4.1.4 และ 4.1.2 ตามลำดับ

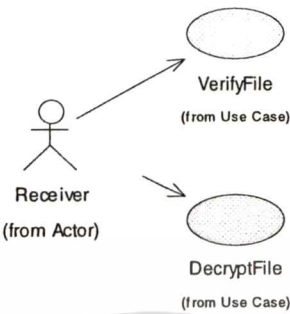
4.2 Use Case Diagram

4.2.1 ส่วนผู้สร้างหรือส่งเพิ่มข้อมูล (Sender)



ภาพที่ 4.5 Use Case Model ส่วนผู้ส่ง

4.2.2 ส่วนผู้รับเพิ่มข้อมูล (Receiver)



ภาพที่ 4.6 Use Case Model ส่วนผู้รับ

4.3 การวิเคราะห์ขั้นตอนการทำงานที่เกิดขึ้นใน Use Case

จะเป็นการวิเคราะห์กระบวนการทำงานที่เกิดขึ้นในแต่ละ Use Case

4.3.1 Use Case Realization Interaction Diagrams

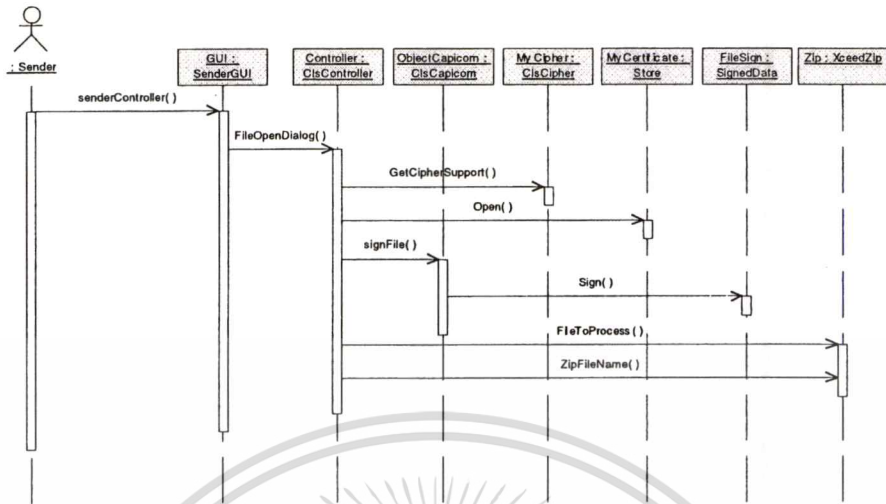
แสดงถึงลำดับการทำงานที่เกิดขึ้นในแต่ละ Use Case

4.3.1.1 SignFile

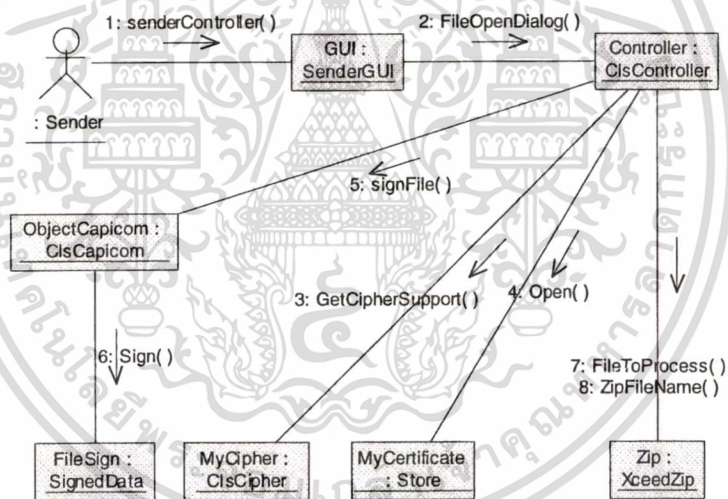
คำอธิบาย Use Case นี้มีหน้าที่สร้างลายมือชื่อดิจิทัลให้กับแฟ้มข้อมูล แสดงลำดับเหตุการณ์ดังในภาพที่ 4.7 และ 4.8 ตามลำดับ

ลำดับเหตุการณ์ (Flow of Event)

- เลือกแฟ้มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ จาก Microsoft Certificate Store
- เลือกอัลกอริทึมที่จะใช้ในการเข้ารหัสลับ
- สร้างลายมือชื่อดิจิทัล
- บีบอัด (Zip) แฟ้มข้อมูลต้นฉบับและแฟ้มข้อมูลที่เป็นลายมือชื่อดิจิทัล



ภาพที่ 4.7 Basic Flow : Sequence Diagram ของ SignFile Use Case



ภาพที่ 4.8 Basic Flow : Collaboration Diagram ของ SignFile Use Case

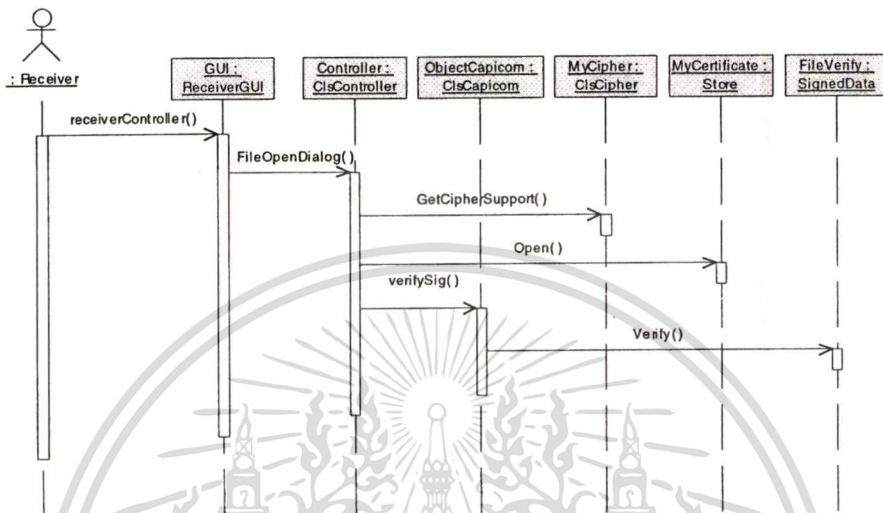
4.3.1.2 VerifyFile

คำอธิบาย Use Case นี้มีหน้าที่ตรวจสอบความถูกต้องของลายมือชื่อดิจิทัล แสดงลำดับเหตุการณ์ดังในภาพที่ 4.9 และ 4.10 ตามลำดับ

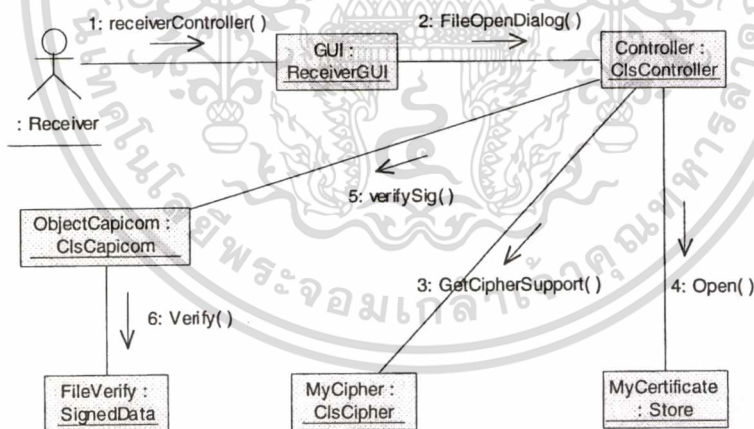
ลำดับเหตุการณ์ (Flow of Event)

- เลือกเพิ่มข้อมูล
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการ จาก Microsoft Certificate Store

- เลือกอัลกอริทึมที่จะใช้ในการเข้ารหัสลับ
- ตรวจสอบลายมือชื่อดิจิทัล



ภาพที่ 4.9 Basic Flow : Sequence Diagram ของ VerifyFile Use Case



ภาพที่ 4.10 Basic Flow : Collaboration ของ VerifyFile Use Case

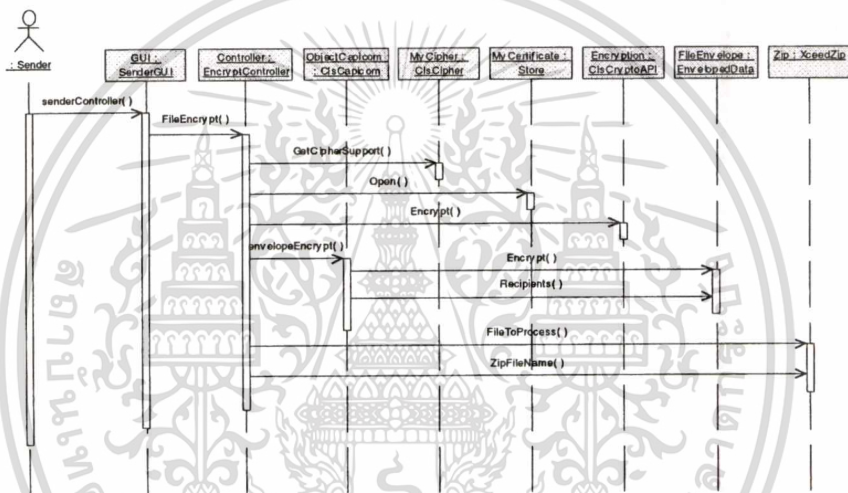
4.3.1.3 EncryptFile

คำอธิบาย Use Case นี้มีหน้าที่เข้ารหัสลับเพิ่มข้อมูล แสดงลำดับเหตุการณ์ดังในภาพที่ 4.11 และ 4.12 ตามลำดับ

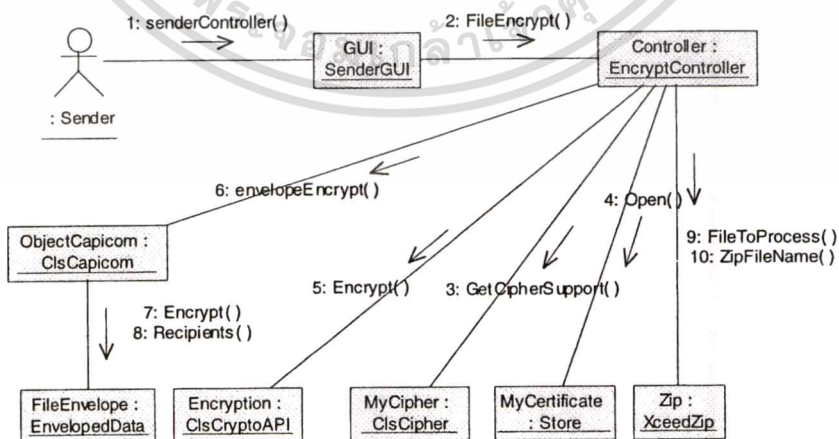
ลำดับเหตุการณ์ (Flow of Event)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกเพิ่มข้อมูล เลือกอัลกอริทึมที่ต้องการใช้
- สร้าง Session Key
- เข้ารหัสลับเพิ่มข้อมูลด้วย Session Key
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการจาก Microsoft Certificate Store
- เข้ารหัสลับ Session Key ด้วยกุญแจสาธารณะของผู้รับ
- รวมข้อมูลทั้งหมดที่ต้องใช้ในการถอดรหัสเก็บไว้ใน Envelope
- บีบอัด (Zip) เพิ่มข้อมูลที่ได้รับการเข้ารหัสลับแล้ว



ภาพที่ 4.11 Basic Flow :Sequence Diagram ของ EncryptFile Use Case



ภาพที่ 4.12 Basic Flow :Sequence Diagram ของ EncryptFile Use Case

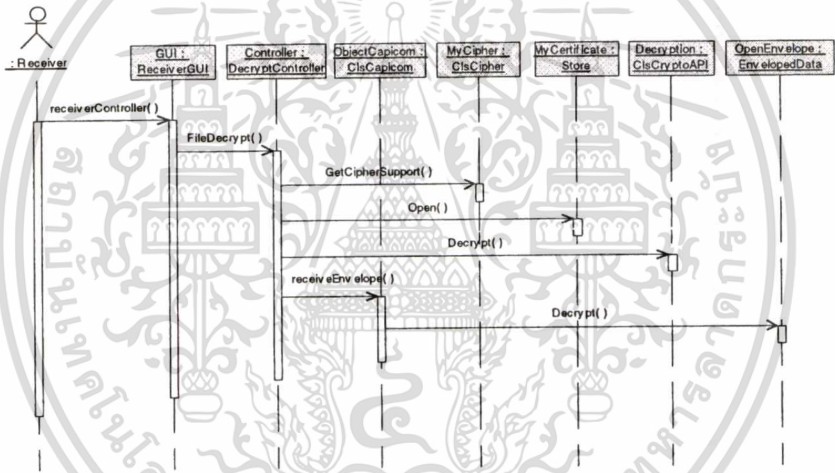
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.4 DecryptFile

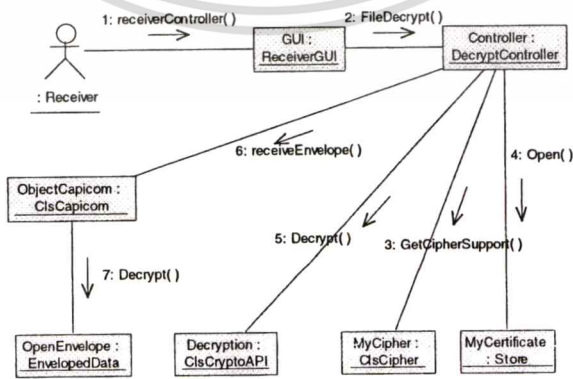
คำอธิบาย Use Case นี้มีหน้าที่ถอดรหัสลับเพิ่มข้อมูล แสดงลำดับเหตุการณ์ดังในภาพที่ 4.13 และ 4.14 ตามลำดับ

ลำดับเหตุการณ์ (Flow of Event)

- เลือกเพิ่มข้อมูล เลือกอัลกอริทึมที่ต้องการใช้
- เลือกใบรับรองอิเล็กทรอนิกส์ที่ต้องการจาก Microsoft Certificate Store
- นำข้อมูลออกจาก Envelope
- ถอดรหัสลับ Session Key ด้วยกุญแจส่วนตัวของผู้รับ
- ถอดรหัสลับเพิ่มข้อมูลด้วย Session Key



ภาพที่ 4.13 Basic Flow : Sequence Diagram ของ DecryptFile Use Case



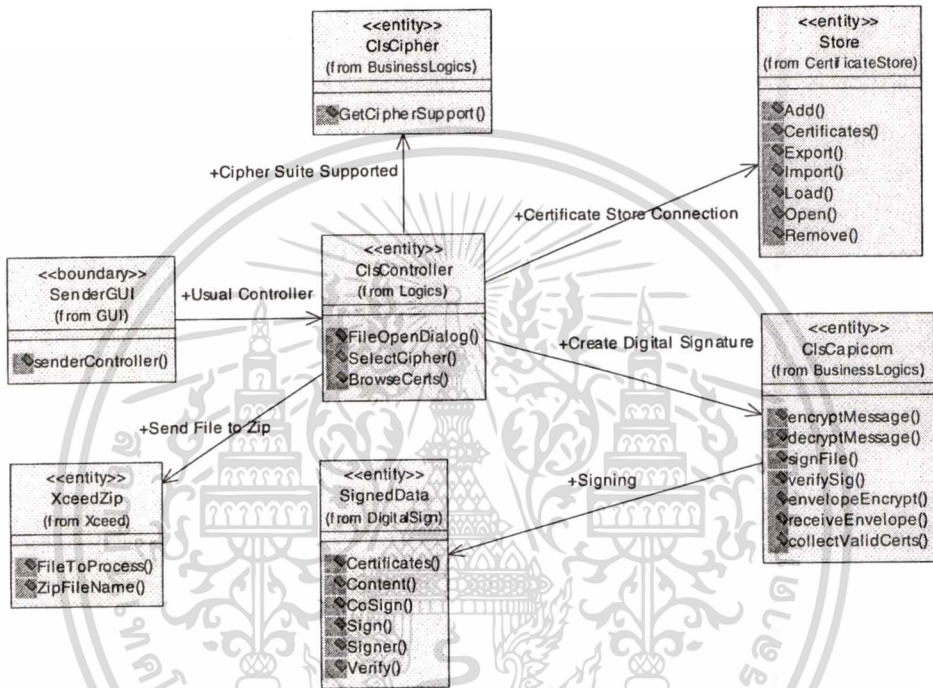
ภาพที่ 4.14 Basic Flow : Collaboration Diagram ของ DecryptFile Use Case

4.3.2 Use Case Realization View of Participating Classes (VOPC)

แสดงถึง Class ที่เกี่ยวข้องต่อการทำงานของแต่ละ Use Case

4.3.2.1 SignFile

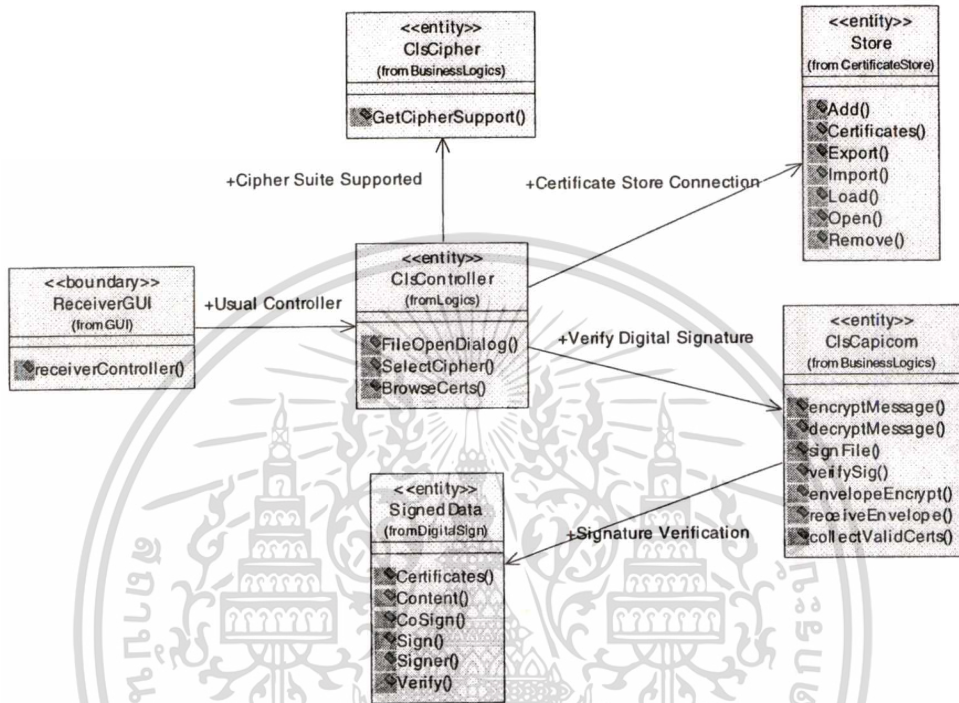
ภาพที่ 4.15 แสดงถึง Class ต่างๆ ที่ออกแบบขึ้นในส่วนของ SignFile Use Case



ภาพที่ 4.15 SignFile Use Case - VOPC

4.3.2.2 VerifyFile

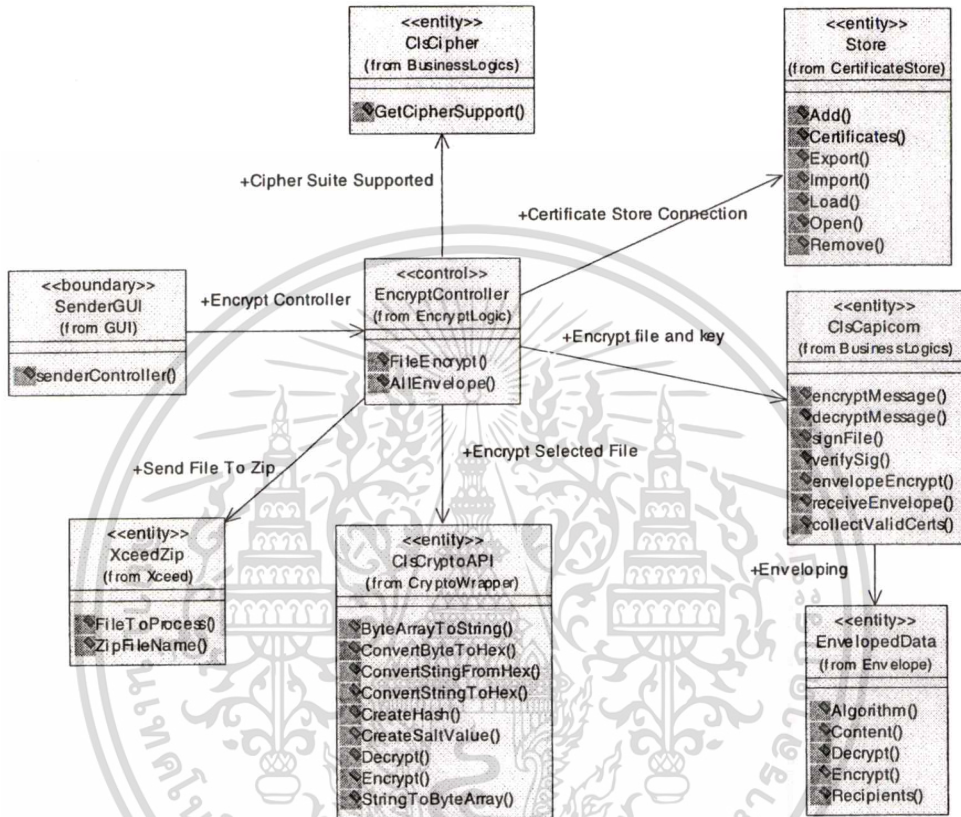
ภาพที่ 4.16 แสดงถึง Class ต่างๆ ที่ออกแบบขึ้นในส่วนของ VerifyFile Use Case



ภาพที่ 4.16 VerifyFile Use Case - VOPC

4.3.2.3 EncryptFile

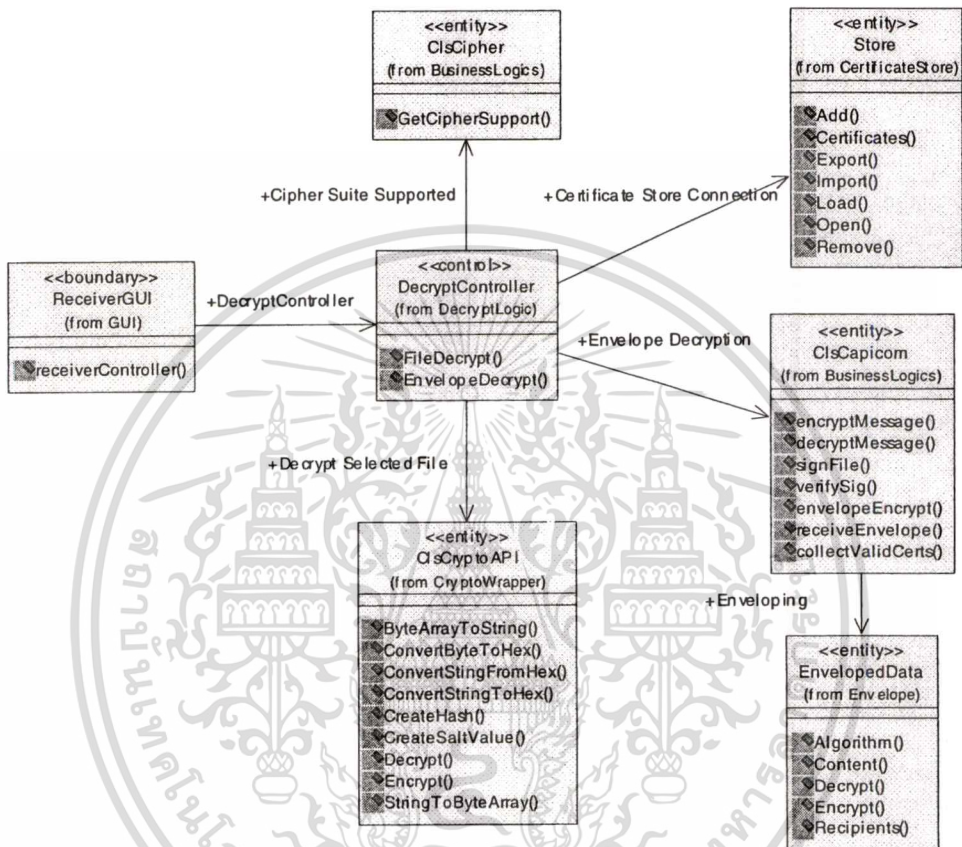
ภาพที่ 4.17 แสดงถึง Class ต่างๆ ที่ออกแบบขึ้นในส่วนของ EncryptFile Use Case



ภาพที่ 4.17 EncryptFile Use Case - VOPC

4.3.2.4 DecryptFile

ภาพที่ 4.18 แสดงถึง Class ต่างๆ ที่ออกแบบขึ้นในส่วนของ DecryptFile Use Case



ภาพที่ 4.18 DecryptFile Use Case - VOPC

4.4 การออกแบบ Element ต่างๆ ในระบบ

4.4.1 Sub System

เป็นส่วนของ Sub System ที่มีได้พัฒนาเอง มีการเรียกใช้งานในโปรแกรม

- Microsoft Cryptography API เป็น API ของไมโครซอฟท์ ประกอบไปด้วยฟังก์ชันที่ช่วยให้สามารถสร้างแอปพลิเคชันสำหรับการเข้ารหัสลับหรือสร้างลายมือชื่อดิจิทัล

```
<<subsystem>>
CryptoAPI
(from Cryptographic Logic)
```

- CryptAPI Wrapper version 1.2 เป็นไฟล์ DLL ที่เป็น Wrapper สำหรับเรียกใช้งาน advapi32.dll ซึ่งเป็นที่เก็บฟังก์ชันเกี่ยวกับ CryptoAPI ของไมโครซอฟท์

```
<<subsystem>>
CryptoWrapper
(from BusinessObjects)
```

ผู้พัฒนาโมดูล Kenneth Ives โดยมีรายละเอียดโมดูลที่พัฒนาขึ้น ดังนี้

วันที่	โมดูล
29-DEC-2000	Original module
10-JUL-2001	Converted to a DLL
09-SEP-2001	Enhanced and fixed some minor bugs

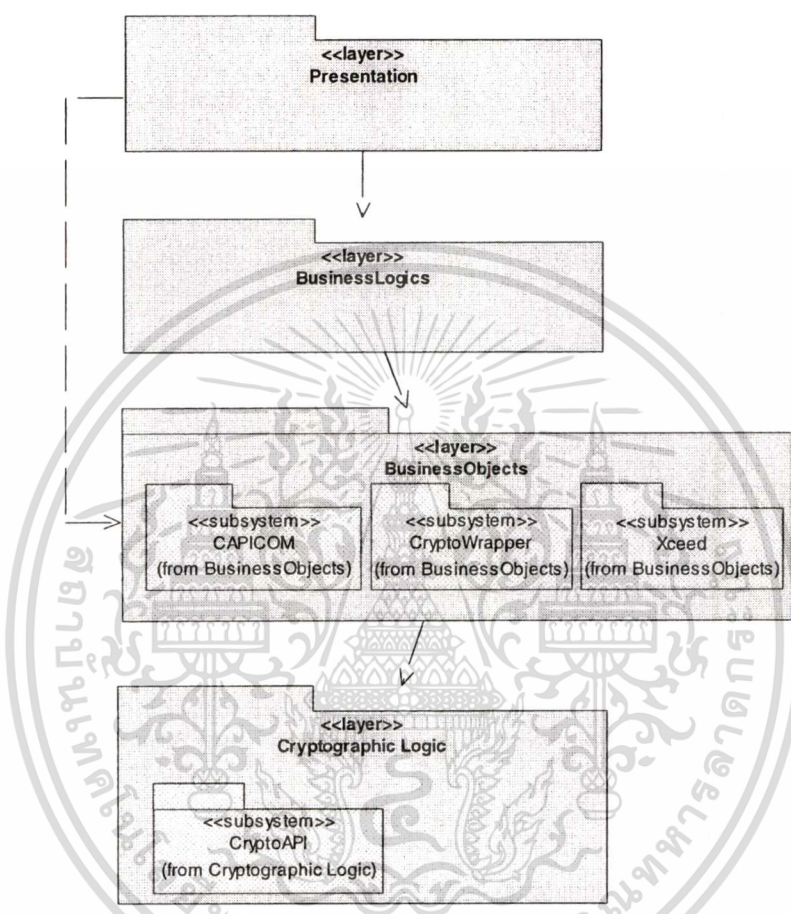
- CAPICOM เป็น ActiveX ของไมโครซอฟท์ใน Platform SDK ที่ให้ COM Interface ในการเรียกใช้งาน CryptoAPI ของไมโครซอฟท์

```
<<subsystem>>
CAPICOM
(from BusinessObjects)
```

- Xceed เป็น Component สำหรับการสร้าง Zip File กล่าวคือการบีบอัดข้อมูลให้มีขนาดเล็ก

```
<<subsystem>>
Xceed
(from BusinessObjects)
```

4.4.2 โครงสร้างของเลเยอร์และความสัมพันธ์ระหว่างกัน (Architectural Layers and Their Dependencies)



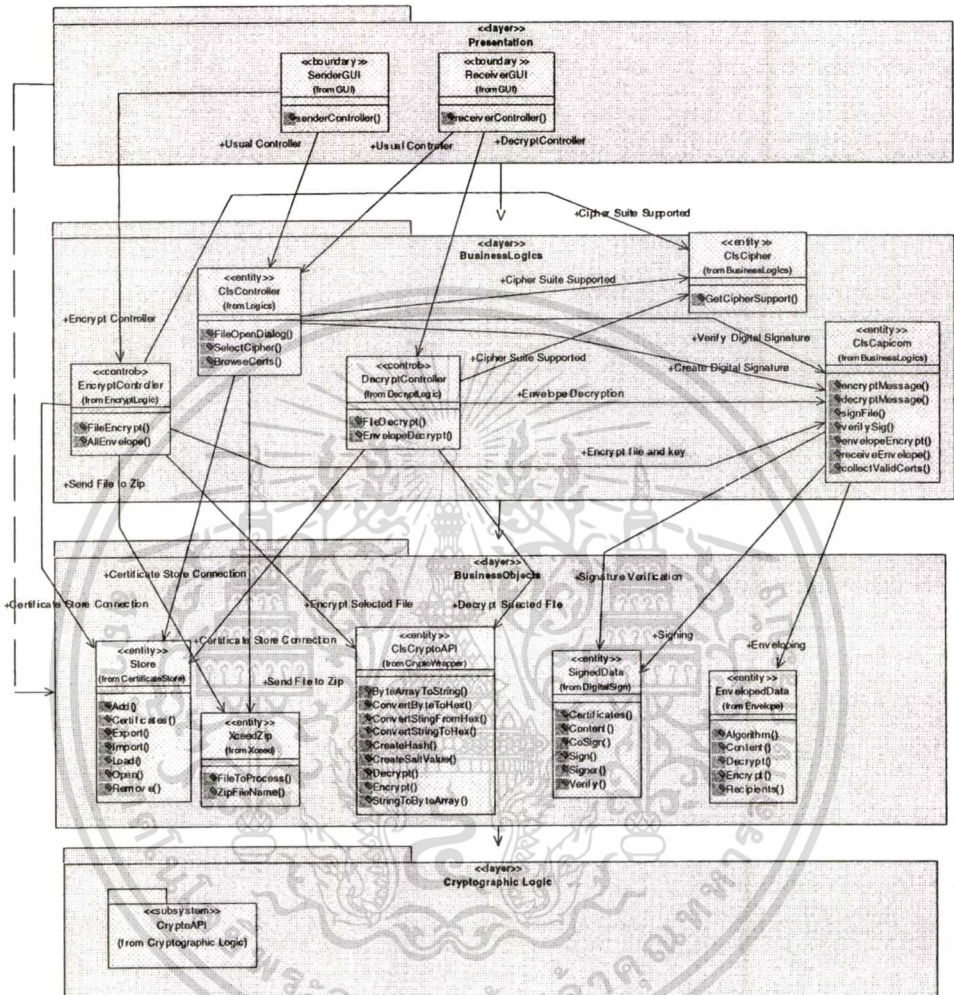
ภาพที่ 4.19 Architectural Layers and Their Dependencies

จากภาพที่ 4.19 ได้ออกแบบโครงสร้างของโปรแกรม โดยแบ่งออกเป็น Layer ที่สามารถอธิบายได้ ดังในตารางที่ 4.1

ตารางที่ 4.1 การอธิบาย Layer ต่างๆ ในการออกแบบโปรแกรม

ชื่อ Layer	หน้าที่	ส่วนประกอบ	หมายเหตุ
Presentation	- ติดต่อผู้ใช้ - เรียกใช้งาน Class จาก Layer อื่น	Element สำหรับส่วนติดต่อผู้ใช้ ติดต่อใช้งาน BusinessLogics และ BusinessObjects	โปรแกรมในส่วนนี้ ได้ทำการพัฒนาขึ้นเอง
BusinessLogics	- เลือกเพิ่มข้อมูล - เรียกใช้งาน Class จาก Layer อื่น - สร้าง Class เพื่อควบคุมการทำงานกับระบบรหัสและเพิ่มข้อมูล	ส่วนของการควบคุม การเปิดเพิ่มข้อมูล การเลือก อัลกอริทึม และการติดต่อกับคลาสต่างๆ ใน BusinessObjects Layer และ Cryptographic Logic	โปรแกรมในส่วนนี้ ได้ทำการพัฒนาขึ้นเอง
BusinessObjects	- ติดต่อฟังก์ชันระบบรหัส Cryptography API บนระบบปฏิบัติการ วินโดวส์ - บีบอัดเพิ่มข้อมูล	ส่วนของ Sub System ได้แก่ CAPICOM, CryptoWrapper และ Xceed	โปรแกรมในส่วนนี้ มิได้ทำการพัฒนาขึ้นเอง เป็นการเรียกใช้ Component สำเร็จรูป
Cryptographic Logic	- ฟังก์ชันระบบรหัส Cryptography API บนระบบปฏิบัติการ วินโดวส์	ส่วนของ Sub System ได้แก่ Microsoft Cryptography API	โปรแกรมในส่วนนี้ มิได้ทำการพัฒนาขึ้นเอง เป็นฟังก์ชันที่อยู่บนระบบปฏิบัติการวินโดวส์

4.4.3 Class Diagram



ภาพที่ 4.20 Class Diagram

จากภาพที่ 4.20 การแบ่งโครงสร้างโปรแกรมออกเป็น Layer สามารถระบุ Class ที่ทำงานใน Layer ต่างๆ ได้ดังนี้

4.4.3.1 Presentation Layer ส่วนของการติดต่อผู้ใช้ (Graphic User Interface)

ประกอบไปด้วย Class ต่างๆ ดังนี้

- SenderGUI
- ReceiverGUI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การอธิบาย Method ต่างๆ ของ Class ใน Presentation Layer

Class : SenderGUI	
Method	คำอธิบาย
SenderController	ส่วนติดต่อผู้ใช้ ติดต่อกับ Class ใน BusinessLogics Layer เพื่อควบคุมการทำงานต่างๆ ผู้ส่ง
Class : ReceiverGUI	
Method	คำอธิบาย
VerifyController	ส่วนติดต่อผู้ใช้ ติดต่อกับ Class ใน BusinessLogics Layer เพื่อควบคุมการทำงานต่างๆ ผู้รับ

4.4.3.2 BusinessLogics Layer ส่วนของการควบคุม การเปิดเพิ่มข้อมูล การเลือกอัลกอริทึม และการติดต่อกับคลาสต่างๆ ใน BusinessObjects Layer ประกอบไปด้วย Class ต่างๆ ดังนี้

- ClsCapicom
- ClsCipher
- ClsController
- DecryptController
- EncryptController

ตารางที่ 4.3 การอธิบาย Method ต่างๆ ของ Class ใน BusinessLogics Layer

Class : ClsCapicom	
Method	คำอธิบาย
signFile	ติดต่อกับ CAPICOM เพื่อสร้างลายมือชื่อดิจิทัลด้วย CAPICOM โดยส่งเพิ่มข้อมูล ญุณแจส่วนตัวและระบุประเภทอัลกอริทึมที่จะใช้งาน
verifySig	ติดต่อกับ CAPICOM เพื่อตรวจสอบลายมือชื่อดิจิทัลด้วย CAPICOM โดยส่งเพิ่มข้อมูล และข้อมูลใบรับรองอิเล็กทรอนิกส์
EncryptMessage	ติดต่อกับ CAPICOM ส่งสตริงข้อมูลไปเพื่อเข้ารหัสลับ

decryptMessage	ติดต่อกับ CAPICOM ส่งสตริงข้อมูลไปเพื่อถอดรหัสลับ
envelopeEncrypt	ติดต่อกับ CAPICOM เพื่อส่งเพิ่มข้อมูล Session Key ในรับรองอิเล็กทรอนิกส์ ไปเพื่อรวมข้อมูลส่งไปใน Envelope
receiveEnvelope	ติดต่อกับ CAPICOM เพื่อส่ง Envelope ทำการดึงข้อมูลภายในออกมา
CollectValidCerts	ตรวจสอบใบรับรองอิเล็กทรอนิกส์ว่ายังคงใช้งานได้หรือไม่
Class : ClsCipher	
Method	คำอธิบาย
getCipherSupport	แสดงประเภทอัลกอริทึมที่คอมพิวเตอร์เครื่องที่ใช้งานรองรับ
Class : ClsController	
Method	คำอธิบาย
FileOpenDialog	เปิดเพิ่มข้อมูลที่ต้องการ
browseCerts	เลือกใบรับรองอิเล็กทรอนิกส์ที่จะใช้งาน โดยติดต่อกับ CAPICOM
selectCipher	เลือกประเภทอัลกอริทึมที่จะใช้งาน
Class : DecryptController	
Method	คำอธิบาย
fileDecrypt	ติดต่อกับ CryptoWrapper ส่งเพิ่มข้อมูลไปเพื่อถอดรหัสลับ
envelopeDecrypt	ติดต่อกับ CAPICOM เพื่อส่ง Envelope ทำการดึงข้อมูลภายในออกมา
Class : EncryptController	
Method	คำอธิบาย
fileEncrypt	ติดต่อกับ CryptoWrapper ส่งเพิ่มข้อมูลไปเพื่อเข้ารหัสลับ
AllEnvelope	ติดต่อกับ CAPICOM เพื่อส่งเพิ่มข้อมูล Session Key ในรับรองอิเล็กทรอนิกส์ ไปเพื่อรวมข้อมูลส่งไปใน Envelope

4.4.3.3 BusinessObjects Layer เป็นส่วนของ Sub System ได้แก่ CAPICOM, CryptoWrapper และ Xceed

4.4.3.3.1 CAPICOM

สามารถแบ่งกลุ่มของ Class ใน CAPICOM ได้ดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 การแบ่งกลุ่มของ Class ใน CAPICOM

ประเภท	คำอธิบาย
Certificate Store Objects	Class สำหรับการเข้าถึงและเรียกใช้งานใบรับรองใน Certificate Store
Digital Signature Objects	Class สำหรับการสร้างลายมือชื่อดิจิทัลให้กับข้อมูล รวมถึงการตรวจสอบความถูกต้องของลายมือชื่อดิจิทัลนั้นด้วย (Sign and Verify)
Enveloped Data Objects	Class สำหรับการสร้าง Enveloped Data Message สำหรับรักษาความ เป็นส่วนตัวและสำหรับถอดรหัส Enveloped Message
Data Encryption Objects	Class สำหรับการเข้ารหัสและถอดรหัสลับข้อมูล
Auxiliary Objects	Class สำหรับการเปลี่ยน Default Behavior และจัดการใบรับรอง, Certificate Store, User Interface Message
Interoperability Interfaces	Interfaces ใช้สำหรับการ Derive CryptoAPI ให้สามารถทำงานร่วมกับ CAPICOM 2.0 ได้
Enumeration Types	Enumeration Type สำหรับใช้งานกับ CAPICOM

Class ในตารางที่ 4.5 ทำงานกับ Certificate Store และ Certificate ทั้งนี้ CAPICOM รองรับการ ใช้งานของ Current User, Local Machine, Memory และ Active Directory Certificate Store เท่านั้น

ตารางที่ 4.5 ประเภทของ Class ในกลุ่ม Certificate Store Object

ชื่อ Class	คำอธิบาย
Certificate	ใบรับรองอิเล็กทรอนิกส์จำนวน 1 ใบ
CertificatePolicies	กลุ่มของ Class PolicyInformation
Certificates	กลุ่มของใบรับรองอิเล็กทรอนิกส์
CertificateStatus	บอกสถานะของใบรับรองอิเล็กทรอนิกส์
Chain	สร้างและตรวจสอบ Certificate Validation Chain จากใบรับรอง อิเล็กทรอนิกส์

ExtendedProperties	แสดงกลุ่มของ Class ExtendedProperty
ExtendedProperty	แสดง Microsoft-extended Property
Extension	แสดง Extension ของใบรับรองอิเล็กทรอนิกส์
Extensions	แสดงกลุ่มของ Class ของ Extension
PrivateKey	แสดงกุญแจส่วนตัว
PublicKey	แสดงกุญแจสาธารณะในใบรับรองอิเล็กทรอนิกส์
Store	แสดง Property และ Method ให้เลือก จัดการ และใช้งาน Certificate Store และ Certificate ที่จัดเก็บอยู่
Template	แสดง Template ของ Certificate Extension

Class ในตารางที่ 4.6 ใช้สำหรับการสร้างและตรวจสอบความถูกต้องของลายมือชื่อดิจิทัล

ตารางที่ 4.6 ประเภทของ Class ในกลุ่ม *Digital Signature Object*

ชื่อ Class	คำอธิบาย
SignedCode	ให้ฟังก์ชันสำหรับการ Sign Content ด้วย Authenticode Digital Signature
SignedData	Class ที่ใช้สำหรับสร้างและตรวจสอบความถูกต้องของลายมือชื่อดิจิทัล
Signer	ข้อมูลของ Data Signer รวมไปถึงใบรับรองของ Signer ด้วย
Signers	กลุ่มของ Signer Object

Class ในตารางที่ 4.7 ใช้สำหรับการสร้าง Enveloped Data Message สำหรับรักษาความเป็นส่วนตัวและถอดรหัสลับของ Enveloped Data Message ด้วย

ตารางที่ 4.7 ประเภทของ Class ในกลุ่ม *Enveloped Data Object*

ชื่อ Class	คำอธิบาย
EnvelopedData	Class สำหรับใช้สร้าง ส่งและรับ Enveloped data ซึ่งถูกเข้ารหัสลับไว้เพื่อให้ผู้รับที่อ้างถึงสามารถถอดรหัสลับได้
Recipients	กลุ่มของใบรับรองของผู้รับที่ต้องการให้ได้รับ Enveloped Message

Class ในตารางที่ 4.8 ใช้สำหรับการเข้ารหัสและถอดรหัสลับ

ตารางที่ 4.8 ประเภทของ Class ในกลุ่ม *Data Encryption Object*

ชื่อ Class	คำอธิบาย
EncryptedData	ใช้สำหรับการเข้ารหัสลับข้อมูล

Class ในตารางที่ 4.9 ใช้สำหรับการเปลี่ยน Default Behavior และจัดการใบรับรอง, Certificate Store, User Interface Message

ตารางที่ 4.9 ประเภทของ Class ในกลุ่ม *Auxiliary Object*

ชื่อ Class	คำอธิบาย
Algorithm	กำหนดอัลกอริทึมและความยาวของกุญแจที่จะใช้ในงานระบบรหัส
Attribute	ข้อมูลเกี่ยวกับลายมือชื่อดิจิทัล เช่น เวลาในตอนสร้างลายมือชื่อดิจิทัล
Attributes	กลุ่มของ Attribute
BasicConstraints	ให้อ่านอย่างเดียวย ส่วน Basic constraint บนใบรับรองของผู้ใช้
EKU	ให้การเข้าถึง EKU Property ของใบรับรอง
EKUs	กลุ่มของ EKU
EncodedData	แสดง Block ของข้อมูลที่ได้รับการเข้ารหัสลับ
ExtendedKeyUsage	ให้อ่านอย่างเดียวยที่ Extended key usage property ของใบรับรอง
HashedData	ฟังก์ชันการแปลง String ด้วยการ Hash
KeyUsage	ให้อ่านอย่างเดียวยที่ Key usage property ของใบรับรอง
OID	แสดง Object Identifier จะถูกใช้งานในหลาย CAPICOM property
OIDs	แสดงกลุ่มของ OID Object
PolicyInformation	ให้การเข้าถึง Policy OID ของ Extension
Qualifier	แสดงพอยเตอร์ไปยัง Certification Practice Statement (CPS)
Qualifiers	แสดงกลุ่มของ Qualifier
Settings	Enable หรือ Disable ส่วน Dialog Box ให้ระบุ Identity ของ Signer

	หรือ Sender หากไม่ได้ระบุไว้
Utilities	ฟังก์ชันทั่วไป

Class ในตารางที่ 4.10 เป็น Interfaces ใช้สำหรับการ Derive CryptoAPI ให้สามารถทำงานร่วมกับ CAPICOM 2.0 ได้

ตารางที่ 4.10 ประเภทของ Class ในกลุ่ม *Interoperability Interface*

Interface	คำอธิบาย
ICertContext	Provides access to the context of a CAPICOM X.509v3 Certificate object. This context allows the CAPICOM certificate to be used in other derivations of CryptoAPI
ICertStore	Provides access to the context of a CAPICOM Store object. This context allows the CAPICOM certificate store to be used in other derivations of CryptoAPI
IChainContext	Provides access to the context of a CAPICOM Chain object. This context allows the CAPICOM certificate trust chain to be used in other derivations of CryptoAPI

4.4.3.3.2 CryptoAPI Wrapper version 1.2

ตารางที่ 4.11 Property ต่างๆ ใน CryptoAPI Wrapper version 1.2

Routine	คำอธิบาย
EnhancedProvider	[Output] Return เป็น Boolean หากมีการเรียกใช้งาน Enhanced Provider อยู่ [Input] เป็น Boolean บอกว่า สามารถเรียกใช้งาน Enhanced Provider ได้หรือไม่
InputData	[Input] ข้อมูลที่จะนำไปเข้าหรือถอดรหัสลับ
OutputData	[Read Only] คืนค่าข้อมูลที่ได้รับการเข้าหรือถอดรหัสลับแล้วในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ของ Byte Array
Password	[Output] คืนค่ารหัสผ่านในรูปแบบของ Byte Array [Input] รับค่ารหัสผ่านจากผู้ใช้ในการรูปแบบของ Byte Array
UseDefaultPWD	[Input] ค่า Boolean ใช้เพื่อ Override Password Property ถ้าค่าเป็น TRUE คือใช้ Default Password ถ้าเป็น FALSE คือสร้าง Random Data Password

ตารางที่ 4.12 การอธิบาย Method ของ CryptoAPI Wrapper Class ใน BusinessObjects Layer

Class : CryptoWrapper	
Method	คำอธิบาย
ByteArrayToString	เปลี่ยน Byte Array เป็น String <Parameter> arByte() – ข้อมูลที่เข้ามาในรูปแบบของ Byte
ConvertByteToHex	เปลี่ยน Byte Array เป็นรูปแบบของ Hex (2 Character) และคืนค่าเป็น Single String <Parameter> abytData() – Array ของข้อมูลที่จะเปลี่ยนรูป
ConvertStingFromHex	เปลี่ยน Hex ให้อยู่ในรูปแบบ ASCII Decimal String <Parameter> strHex – Data String ที่จะเปลี่ยนรูป
ConvertStringToHex	นำเข้าทีละ 1 Character แล้วเปลี่ยนรูปเป็น Integer จากนั้นเป็น Hex ใส่ Prefix เป็น 00 ในกรณีที่ค่าที่ได้มีผลลัพธ์ตั้งแต่ 0x00 ถึง 0xFF เพื่อให้แสดงผลได้ชัดเจนขึ้น <Parameter> strInput – ข้อมูลที่จะเปลี่ยนรูป blnRetUpperCase – (Optional) [Default] – TRUE คือ จะเปลี่ยนรูปให้เป็น Uppercase ก่อน ถ้ากำหนดให้เป็น FALSE ก็จะไม่เปลี่ยนรูปเป็น Uppercase
CreateHash	สร้าง One-way Hash String จากข้อมูล String มีอัลกอริทึมให้ใช้งาน 4

	แบบ คือ MD5, MD4, MD2, SHA-1 <Parameter> strInText – String ที่จะนำมา Hash intHashChoice - (Optional) Numeric identifier สำหรับบอกประเภทของ Hash Algorithm ที่จะใช้ ค่า [Default] = 1 (MD5) blnConvertToHex - (Optional) [Default] TRUE คือจะเปลี่ยนรูปข้อมูลที่ได้ออกมาเป็น Hex ถ้ากำหนดเป็น FALSE จะไม่เปลี่ยนรูปข้อมูล blnAppendPassword - (Optional) [Default] FALSE คือไม่เพิ่มข้อมูล Password ไว้กับข้อมูลที่ถูก Hash แต่ถ้ากำหนดเป็น TRUE จะเพิ่มข้อมูล Password ไปกับข้อมูลที่จะถูก Hash ด้วย blnCaseSensitive - (Optional) ใช้เมื่อ blnConvertToHex ถูกกำหนดให้เป็น TRUE ถ้ากำหนดค่าเป็น [Default] FALSE จะเปลี่ยนรูปข้อมูลเป็น Uppercase แต่ถ้าเป็น TRUE จะคืนค่าข้อมูลตามรูปแบบเดิมที่ได้สร้างไว้
CreateSaltValue	สร้างข้อมูลแบบสุ่มเพื่อใช้เป็น Salt Value จะคืนค่าเป็น 0-9, A-Z, a-z หรือข้อมูลอื่นใดที่ได้จากการสุ่มสร้างขึ้น <Parameter> lngReturnLength – ความยาวของข้อมูลที่จะส่งคืน blnUseLettersNumbersOnly - (Optional) [Default] คือ TRUE จะใช้เฉพาะตัวอักษรและตัวเลขเท่านั้น ถ้ากำหนดเป็น FALSE จะใช้ข้อมูลแบบสุ่มอย่างแท้จริง
Decrypt	เรียกใช้ Decryption Routine <Parameter> intHashType - (Optional) [Default] คือ 1 จะใช้ MD5 Hash Algorithm สามารถเลือกได้ดังนี้ 1=MD5 2=MD4 3=MD2 4=SHA-1 intCipherType - (Optional) [Default] คือ 1 ใช้ RC4 Algorithm สามารถเลือกได้ดังนี้ (Default Provider) 1=RC4 2=RC2 3=DES (Enhanced Provider) 4=3DES 5=3DES_112
Encrypt	เรียกใช้ Encryption Routine <Parameter> intHashType - (Optional) [Default] คือ 1 จะใช้ MD5 Hash Algorithm

	สามารถเลือกได้ดังนี้ 1=MD5 2=MD4 3=MD2 4=SHA-1 intCipherType - (Optional) [Default]] คือ 1 ใช้ RC4 Algorithm สามารถเลือกได้ดังนี้ (Default Provider) 1=RC4 2=RC2 3=DES (Enhanced Provider) 4=3DES 5=3DES_112
StringToByteArray	เปลี่ยนรูป String เป็น Byte Array [Range 0, 255] <Parameter> strInput – String ที่จะเปลี่ยนรูปเป็น Byte Array

4.4.3.3.3 Xceed

ระบุ Class ใน Xceed ที่เรียกใช้ ดังแสดงในตารางที่ 4.13

ตารางที่ 4.13 Class ที่เรียกใช้จาก Xceed

Class : XceedZip	
Class สำหรับการบีบอัดเพิ่มข้อมูล	
Method	คำอธิบาย
FileToProcess	รับเพิ่มข้อมูลที่ต้องการนำไปบีบอัด
ZipFileName	ระบุเพิ่มข้อมูลผลลัพธ์ที่มีนามสกุลเป็น .zip

4.4.3.4 Cryptographic Layer เป็นส่วนของ Sub System ได้แก่ Microsoft Cryptography API

ได้แสดงรายละเอียดของฟังก์ชันระบบรหัสของไมโครซอฟท์ไว้ดังในตารางที่ 2.1 ฟังก์ชันระบบรหัสของไมโครซอฟท์

บทที่ 5

การพัฒนาโปรแกรมรักษาความปลอดภัยของแฟ้มข้อมูล

5.1 ซอฟต์แวร์สำหรับการพัฒนาโปรแกรม

Microsoft Visual Basic 6.0

5.2 การนำ DLL - CryptAPI Wrapper version 1.2 มาใช้งาน

เป็น Wrapper ที่ใช้เพื่อเรียกใช้ advapi32.dll ซึ่งเป็นที่เก็บฟังก์ชันเกี่ยวกับ CryptoAPI ของไมโครซอฟท์

5.3 การใช้งาน CAPICOM 2.0.0.1

CAPICOM เป็น ActiveX ของไมโครซอฟท์ที่ให้ COM Interface ในการเรียกใช้งาน CryptoAPI ของไมโครซอฟท์ ติดต่อกับกลุ่มของฟังก์ชัน CryptoAPI เพื่อให้ผู้พัฒนาสามารถสร้างแอปพลิเคชันทำงานร่วมกับ Digital Signing และการเข้ารหัส/ถอดกลับได้ โดยสามารถใช้ในการพัฒนาด้วย Microsoft Visual Basic, Visual Basic Script, Active Server Pages, Microsoft JScript, C++ และอื่นๆ CAPICOM อยู่ในรูปของ ActiveX control ทำให้สามารถใช้ในเว็บแอปพลิเคชันได้ ทั้งนี้ CAPICOM สามารถทำหน้าที่ได้ดังนี้

- สร้างลายมือชื่อดิจิทัลให้กับข้อมูลจากกุญแจส่วนตัวที่บันทึกในบัตร Smart Card หรือที่บันทึกอยู่บนเครื่องคอมพิวเตอร์
- ตรวจสอบความถูกต้องของลายมือชื่อดิจิทัล
- แสดงข้อมูลใบรับรองอิเล็กทรอนิกส์แบบกราฟฟิก
- ตรวจสอบข้อมูลของใบรับรองอิเล็กทรอนิกส์เช่น Subject Name หรือ Expiration Date เป็นต้น
- เพิ่มหรือลบใบรับรองอิเล็กทรอนิกส์จาก Certificate Store ของไมโครซอฟท์
- เข้าและถอดรหัสลับข้อมูลด้วยรหัสผ่าน
- เข้าและถอดรหัสลับข้อมูลด้วยกุญแจสาธารณะและใบรับรองอิเล็กทรอนิกส์

5.3.1 การสร้างลายมือชื่อดิจิทัล

กล่าวคือ สร้างเป็นแฟ้มข้อมูล Signed Text มีรูปแบบเป็น PKCS7 จากนั้น สามารถนำไปบนเครือข่ายอินเทอร์เน็ตได้

การสร้าง Digital Signature จะใช้กุญแจส่วนตัวของผู้สร้างหรือส่งแฟ้มข้อมูล (Signer) โดยจะสร้างได้ก็ต่อเมื่อมีทั้งใบรับรองและกุญแจส่วนตัวของ Signer ถ้าไม่ได้ระบุและไม่มีใบรับรองกับกุญแจส่วนตัวใน CAPICOM_MY_STORE กระบวนการจะล้มเหลว แต่ถ้ามีใบรับรองเพียง 1 ใบใน CAPICOM_MY_STORE รวมทั้งมีกุญแจส่วนตัวด้วย ก็จะสามารถนำมาใช้งานเพื่อสร้างลายมือชื่อดิจิทัลได้ แต่หากมีใบรับรองมากกว่า 1 ใบใน CAPICOM_MY_STORE ส่วนติดต่อผู้ใช้จะแสดงไดอะล็อกบ็อกซ์ให้เลือกใบรับรองที่ต้องการ

5.3.2 การตรวจสอบลายมือชื่อดิจิทัล

สิ่งที่ต้องตรวจสอบได้แก่

- Validity ของ Signature Hash
- Validity ของใบรับรองอิเล็กทรอนิกส์ของ Signer

ส่วนที่เป็น Signature Hash นั้นจะได้รับการถอดรหัสกลับโดยใช้กุญแจสาธารณะของ Signer ที่อยู่บนใบรับรองของ Signer หากถอดรหัสได้ออกมาตรงกับ Signature Hash ที่สร้างใหม่ จะสามารถสรุปได้ว่า Signature นั้นสร้างจากผู้ส่งจริง Content ได้รับการ Sign แล้ว SignedData object จะถูกอ่านจากไฟล์ จากนั้นจะตรวจสอบ Validity ของ Signature กับใบรับรอง

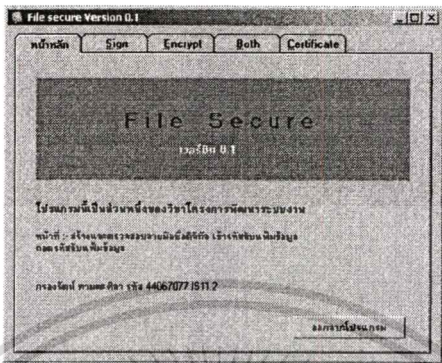
5.3.3 การเข้ารหัสลับข้อมูลด้วย CAPICOM

สามารถใช้ CAPICOM เพื่อการเข้ารหัสลับข้อมูลด้วยการใช้กุญแจสาธารณะด้วยการ Envelope Data ซึ่ง Enveloped Message นี้จะทำให้แน่ใจได้ว่าเนื้อความจะได้รับการอ่านจากเฉพาะผู้รับที่มีกุญแจส่วนตัวที่คู่กับกุญแจสาธารณะที่ผู้ส่งใช้ในการเข้ารหัสลับข้อมูลเท่านั้น เมื่อทำการ Envelope Message กุญแจ Session จะถูกสร้างอัตโนมัติแล้วนำไปใช้ในการเข้ารหัสลับข้อมูล จากนั้น กุญแจ Session จะถูกเข้ารหัสลับแยกต่างหากด้วยกุญแจสาธารณะของผู้รับ (นำมาจากใบรับรองของผู้รับ) Enveloped Message นี้จะประกอบไปด้วย Message ที่ได้รับการเข้ารหัสลับแล้ว ใบรับรองอิเล็กทรอนิกส์ของผู้รับ และกุญแจ Session ที่ได้รับการรหัสลับ ซึ่งจะมี 1 กุญแจต่อผู้ใช้ 1 คน

CAPICOM ไม่รองรับลักษณะข้อมูลแบบ PKCS7 แต่ใช้ Non-Standard ASN Structure สำหรับข้อมูลที่เข้ารหัสลับ EncryptedData ดังนั้น มีแต่ CAPICOM เท่านั้นที่สามารถถอดรหัสลับ CAPICOM EncryptedData Object ได้

5.4 การพัฒนาโปรแกรม

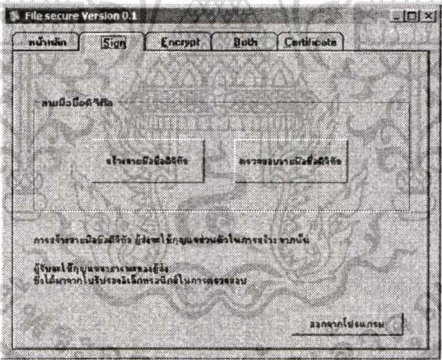
- หน้าหลักของโปรแกรม ดังในภาพที่ 5.1



ภาพที่ 5.1 แสดงหน้าต่างหลักของโปรแกรม

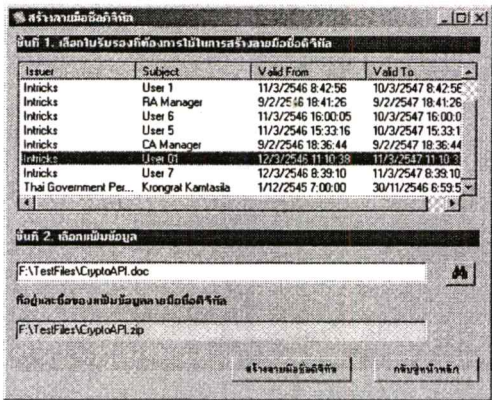
5.4.1 การสร้างลายมือชื่อดิจิทัล

- เมื่อคลิก Tab Signing จะปรากฏหน้าจอ ดังในภาพที่ 5.2



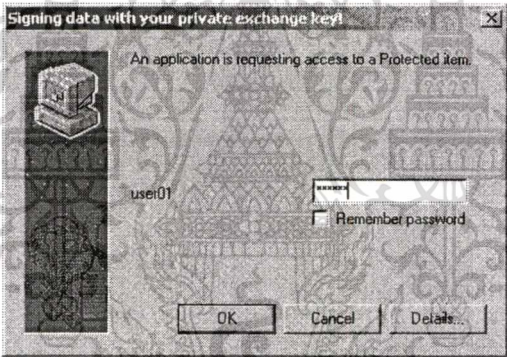
ภาพที่ 5.2 แสดงหน้าจอส่วนการสร้างและตรวจสอบลายมือชื่อดิจิทัล

- จากภาพที่ 5.2 เมื่อคลิกที่ปุ่ม “สร้างลายมือชื่อดิจิทัล” จะปรากฏหน้าจอให้เลือกใบรับรองที่จะใช้ และเลือกเพิ่มข้อมูลที่ต้องการ ดังในภาพที่ 5.3



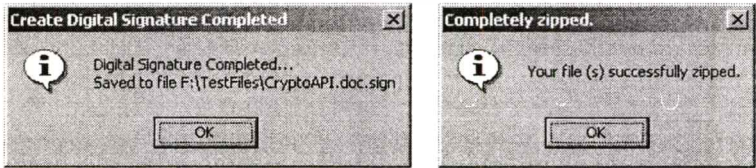
ภาพที่ 5.3 แสดงหน้าจอการสร้างลายมือชื่อดิจิทัล

- จากภาพที่ 5.3 เมื่อคลิกปุ่ม “สร้างลายมือชื่อดิจิทัล” จะปรากฏหน้าจอ ดังภาพที่ 5.4 ให้พิมพ์รหัสผ่านเพื่อใช้งานกุญแจส่วนตัว



ภาพที่ 5.4 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัว

- จากภาพที่ 5.4 เมื่อได้ใส่รหัสผ่านที่ถูกต้อง และคลิกปุ่ม OK จะปรากฏหน้าจอแสดงผลการสร้างลายมือชื่อดิจิทัลว่าได้สร้างเสร็จสิ้นและบีบอัดเพิ่มข้อมูลแล้ว ดังในภาพที่ 5.5

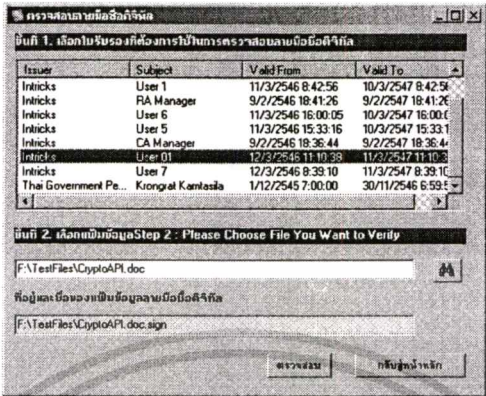


ภาพที่ 5.5 แสดงผลการสร้างลายมือชื่อดิจิทัล

5.4.2 การตรวจสอบลายมือชื่อดิจิทัล

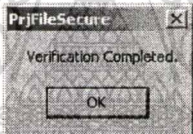
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากภาพที่ 5.2 เมื่อคลิกปุ่ม “ตรวจสอบลายมือชื่อดิจิทัล” จะปรากฏหน้าจอดังในภาพที่ 5.6



ภาพที่ 5.6 แสดงหน้าจอการตรวจสอบลายมือชื่อดิจิทัล

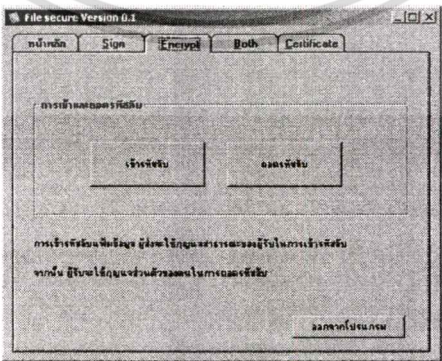
- เมื่อเลือกใบรับรองที่ต้องการและเพิ่มข้อมูลที่ต้องการตรวจสอบแล้ว คลิกปุ่ม “ตรวจสอบ” จะปรากฏหน้าจอดังในภาพที่ 5.7



ภาพที่ 5.7 แสดงผลตรวจสอบลายมือชื่อดิจิทัล

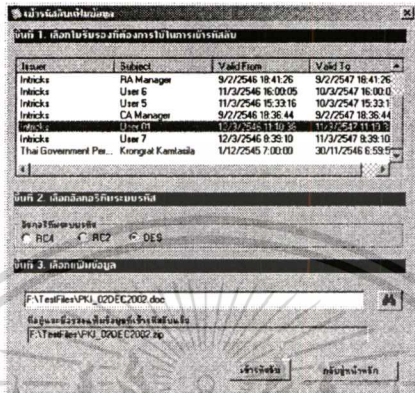
5.4.3 การเข้ารหัสลับเพิ่มข้อมูล

- จากภาพที่ 5.1 เมื่อคลิก Tab Encrypt จะปรากฏหน้าจอ ดังในภาพที่ 5.8



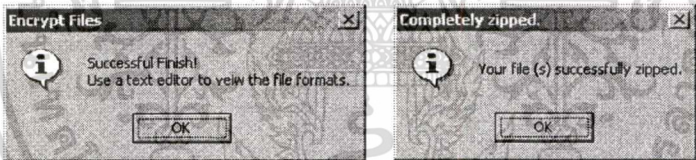
ภาพที่ 5.8 แสดงส่วนการเข้ารหัสและถอดรหัสลับเพิ่มข้อมูล

- จากภาพที่ 5.8 เมื่อคลิกปุ่ม “เข้ารหัสลับ” จะปรากฏหน้าต่างดังแสดงในภาพที่ 5.9 จากนั้นทำการเลือกใบรับรองที่จะใช้งาน (User 01) เพิ่มข้อมูลที่ต้องการนำมาเข้ารหัสลับและอัลกอริทึมที่ต้องการใช้



ภาพที่ 5.9 แสดงหน้าต่างการเข้ารหัสลับข้อมูล

- จากภาพที่ 5.9 เมื่อคลิกปุ่ม “เข้ารหัสลับ” จะปรากฏหน้าต่างดังแสดงในภาพที่ 5.10



ภาพที่ 5.10 แสดงผลการเข้ารหัสลับเพิ่มข้อมูล

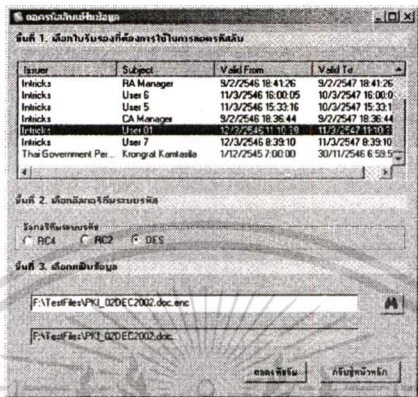
- เพิ่มข้อมูลที่ได้รับการเข้ารหัสเรียบร้อยแล้วจะได้รับการบันทึกเป็นแฟ้มข้อมูลนามสกุล .enc และทำการบีบอัดให้เป็น Zip File หากทดสอบเปิดด้วยโปรแกรม Notepad จะไม่สามารถอ่านเข้าใจได้ ดังในภาพที่ 5.11



ภาพที่ 5.11 แสดงแฟ้มข้อมูลผ่านการเข้ารหัสลับแล้ว

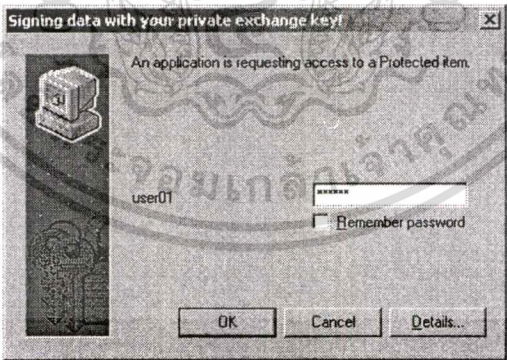
5.4.4 การถอดรหัสลับเพิ่มข้อมูล

- จากภาพที่ 5.8 เมื่อคลิกปุ่ม “ถอดรหัสลับ” จะปรากฏหน้าต่างดังในภาพที่ 5.12 ทำการเลือกเพิ่มข้อมูลนามสกุล .enc เลือกใบรับรองและอัลกอริทึมที่ต้องการใช้



ภาพที่ 5.12 แสดงหน้าจอการถอดรหัสลับ

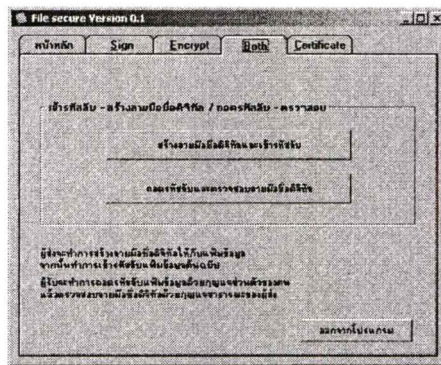
- ในการถอดรหัสลับเพิ่มข้อมูล ผู้รับจะต้องใช้กุญแจส่วนตัวในการถอดรหัสลับ ในที่นี้คือ User 01 ดังนั้นจะต้องพิมพ์รหัสผ่านเพื่อเข้าใช้งานกุญแจส่วนตัวดังแสดงในภาพที่ 5.13
- ทั้งนี้ เพิ่มข้อมูลที่ได้รับการถอดรหัสจะได้รับการบันทึกเป็นเพิ่มข้อมูลนามสกุลเดิมเหมือนต้นฉบับ



ภาพที่ 5.13 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัว

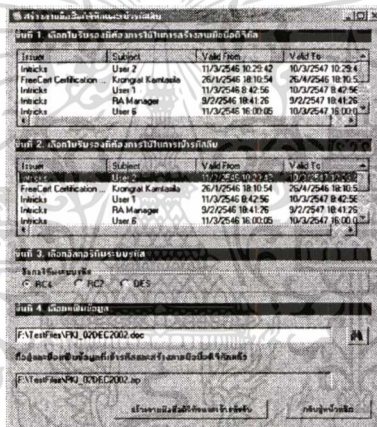
5.4.5 การสร้างลายมือชื่อดิจิทัลให้กับเพิ่มข้อมูลและเข้ารหัสลับเพิ่มข้อมูลนั้น

- จากภาพที่ 5.1 เมื่อคลิก Tab Both จะปรากฏหน้าจอ ดังในภาพที่ 5.14



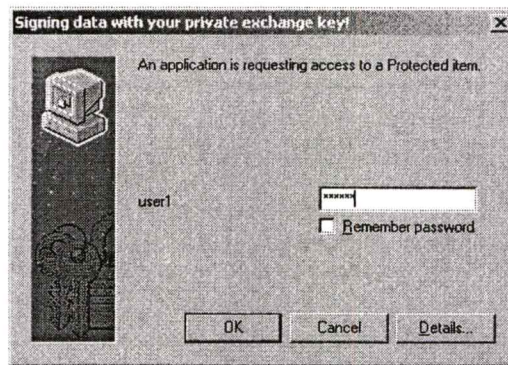
ภาพที่ 5.14 หน้าจอแสดงส่วนการรักษาความปลอดภัยสองแบบควบคู่กัน

- เมื่อคลิกปุ่ม “สร้างลายมือชื่อดิจิทัลและเข้ารหัสลับ” จะปรากฏหน้าจอ ดังในภาพที่ 5.15



ภาพที่ 5.15 หน้าจอแสดงการสร้างลายมือชื่อดิจิทัลและเข้ารหัสลับเพิ่มข้อมูล

- จากภาพที่ 5.15 สามารถอธิบายการทำงานได้ดังนี้
 - ขั้นที่ 1 จะเป็นการเลือกใบรับรองที่จะใช้สร้างลายมือชื่อดิจิทัล นั่นคือ ใบรับรองของผู้ส่ง (User 1) จะใช้กุญแจส่วนตัวในการสร้างลายมือชื่อดิจิทัล
 - ขั้นที่ 2 จะเป็นการเลือกใบรับรองที่จะใช้เข้ารหัสลับ นั่นคือ ใบรับรองของผู้รับ (User 2) จะใช้กุญแจสาธารณะในการเข้ารหัสลับ
 - ขั้นที่ 3 จะเป็นการเลือกอัลกอริทึมที่จะใช้ในการเข้ารหัสลับ
 - ขั้นที่ 4 จะเป็นการเลือกเพิ่มข้อมูลที่ต้องการ
- เมื่อคลิกปุ่ม “สร้างลายมือชื่อดิจิทัลและเข้ารหัสลับ” จะปรากฏหน้าจอ ดังในภาพที่ 5.16



ภาพที่ 5.16 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัวของ User 1

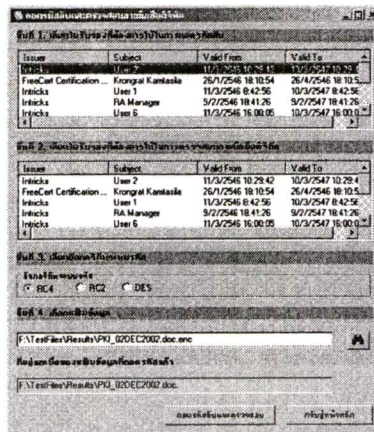
- แสดงผลลัพธ์ดังในภาพที่ 5.17



ภาพที่ 5.17 หน้าจอแสดงผลหลังจากการสร้างลายมือชื่อดิจิทัลและเข้ารหัสลับ

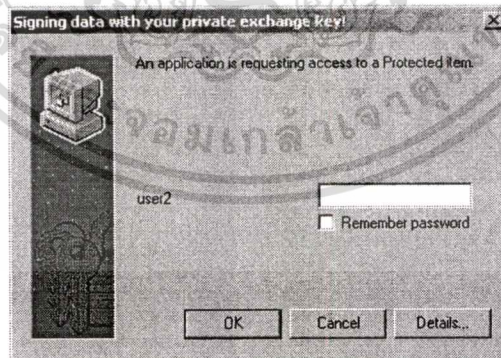
5.4.6 การถอดรหัสลับและตรวจสอบลายมือชื่อดิจิทัล

- ผู้รับจะได้รับแฟ้มข้อมูลที่บีบอัดแล้ว ภายในประกอบไปด้วยแฟ้มข้อมูลที่เข้ารหัสลับไว้ (.enc) และแฟ้มข้อมูลที่เป็นลายมือชื่อดิจิทัล (.sign)
- จากภาพที่ 5.14 เมื่อคลิกปุ่ม “ถอดรหัสลับและตรวจสอบลายมือชื่อดิจิทัล” จะปรากฏหน้าจอ ดังในภาพที่ 5.18



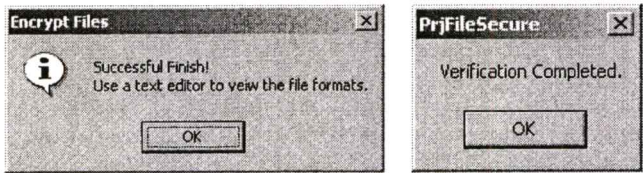
ภาพที่ 5.18 หน้าจอแสดงการถอดรหัสลับและตรวจสอบลายมือชื่อดิจิทัล

- จากภาพที่ 5.18 สามารถอธิบายการทำงานได้ดังนี้
 - ขั้นที่ 1 จะเป็นการเลือกใบรับรองที่จะใช้ถอดรหัสลับ นั่นคือ ใบรับรองของผู้รับ (User 2) จะใช้กุญแจส่วนตัวในการถอดรหัสลับ
 - ขั้นที่ 2 จะเป็นการเลือกใบรับรองที่จะใช้ตรวจสอบลายมือชื่อดิจิทัล นั่นคือ ใบรับรองของผู้ส่ง (User 1) จะใช้กุญแจสาธารณะในการตรวจสอบ
 - ขั้นที่ 3 จะเป็นการเลือกอัลกอริทึมที่จะใช้ในการเข้ารหัสลับ
 - ขั้นที่ 4 จะเป็นการเลือกเพิ่มข้อมูลที่ต้องการ (.enc)
- เมื่อคลิกปุ่ม “ถอดรหัสลับและตรวจสอบ” จะปรากฏหน้าจอ ดังในภาพที่ 5.19



ภาพที่ 5.19 หน้าจอสำหรับพิมพ์รหัสผ่านเข้าใช้งานกุญแจส่วนตัวของ User 2

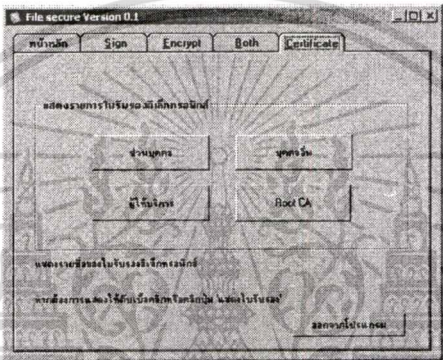
- แสดงผลลัพธ์ดังในภาพที่ 5.20



ภาพที่ 5.20 หน้าจอแสดงผลลัพธ์จากการถอดรหัสลับและตรวจสอบ

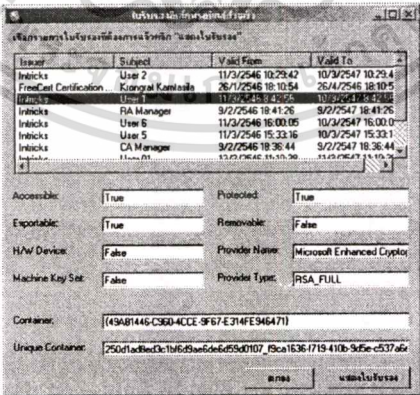
5.4.7 การเรียกดูข้อมูลใบรับรองอิเล็กทรอนิกส์

- จากภาพที่ 5.1 เมื่อคลิก Tab Certificates จะปรากฏหน้าจอ ดังในภาพที่ 5.21



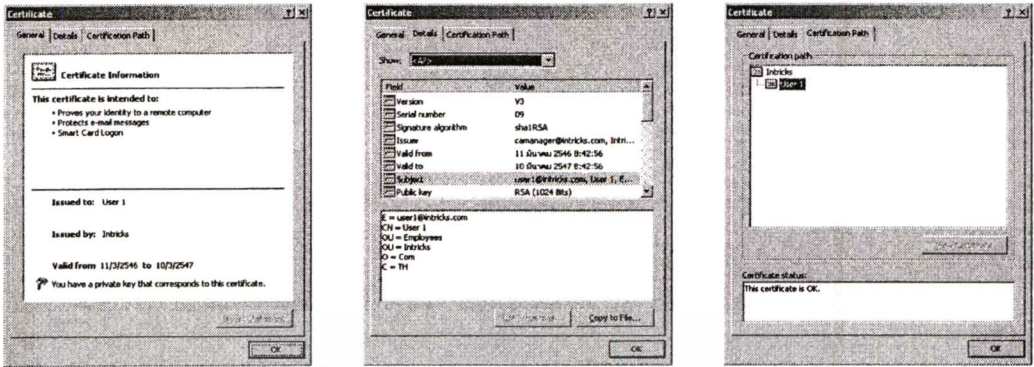
ภาพที่ 5.21 แสดงหน้าจอเกี่ยวกับใบรับรองอิเล็กทรอนิกส์

- จากภาพที่ 5.21 เมื่อคลิกปุ่ม “ส่วนบุคคล” จะปรากฏหน้าจอแสดงในภาพที่ 5.22 และหากคลิกเลือกที่ใบรับรองใบใดๆ จะปรากฏรายละเอียดดังในภาพ



ภาพที่ 5.22 แสดงหน้าจอรายการใบรับรองอิเล็กทรอนิกส์ส่วนบุคคล

- จากภาพที่ 5.22 เมื่อคลิกปุ่ม “แสดงใบรับรอง” จะปรากฏหน้าจอแสดงในภาพที่ 5.23



ภาพที่ 5.23 แสดงข้อมูลใบรับรองอิเล็กทรอนิกส์

5.5 อัลกอริทึมที่ใช้ในโปรแกรม

5.5.1 RSA

- พัฒนาขึ้น โดย Rivest, Shamir และ Adelman ในปี ค.ศ. 1978
- จะทำงานบนพื้นฐานของความยากในการหา Factor ของเลขจำนวนมาก
- ใช้เลขจำนวนเฉพาะที่มีค่ามากๆ (Large Prime Number) 2 ค่า (10^{100} หรือมากกว่า)
- RSA Algorithm ทำงานดังนี้

- เลือกเลขจำนวนเฉพาะที่มีค่ามาก 2 จำนวน ได้แก่ p และ q
- คำนวณค่า Modulus $n = p * q$
- เลือกเลข e โดยให้ e น้อยกว่า n และ e กับ $(p-1)(q-1)$ ไม่มี Common Factor ยกเว้น 1 ตัวเดียว
- เลือกเลข d ที่ $(ed-1)$ จะต้องหารโดย $(p-1)(q-1)$ ได้ลงตัว
- e กับ d เรียกว่าเป็น Public Exponent และ Private Exponent ตามลำดับ
- Public Key คือ (n,e)
- Private Key คือ (n,d)
- ทั้งนี้ Factor p, q อาจจะทำลายทิ้งไป หรือเก็บไว้กับ Private Key ก็ได้
- ที่จริงแล้ว การหา Private Key d จาก Public Key (n,e) นั้นทำได้ยากมาก อย่างไรก็ตาม ถ้าผู้ใดสามารถแยก Factor n ของ p และ q ได้ ก็จะสามารถหาค่า d ได้
- ความปลอดภัยของ RSA มีพื้นฐานว่า การหา Factor นั้นยาก หากมีการค้นพบวิธีการหา Factor ได้ง่าย นั้นหมายถึงจะสามารถทำลายความปลอดภัยของ RSA ได้

- การ Encrypt ด้วย RSA Algorithm หาก Alice ต้องการส่ง Message ไปยัง Bob จะเกิดขึ้น
ตอนดังนี้

Alice ----- > Message M ----- > Bob

1. สร้าง Cipher Text C ด้วยวิธี $C = M^e \bmod n$; $e \& n = \text{Bob's Public Key}$

จากนั้น ส่ง C ไปยัง Bob

2. Bob ถอดรหัสลับ ด้วยวิธี $M = C^d \bmod n$

- การสร้าง Digital Signature ด้วย RSA Algorithm มีขั้นตอนดังนี้

1. สร้างลายมือชื่อดิจิทัล Digital Signature S ด้วยวิธี $S = M^d \bmod n$; $d \& n = \text{Alice's Private Key}$

Alice ----- > M & S ----- > Bob

2. ตรวจสอบลายมือชื่อดิจิทัล ด้วยวิธี $M = S^e \bmod n$; $e \& n = \text{Alice's Public Key}$

- ตัวอย่างการสร้างกุญแจด้วยอัลกอริทึม RSA

- เลือกเลขจำนวนเฉพาะ (Prime Number)
- $p = 5, q = 11$; $n = p * q = 55$
- $k < \log(n) = 5$; Length of Mess
- $z = (p-1)(q-1) = 40$
- เลือก e ที่ไม่มี common factor กับ z ได้ 13
- เลือก d ที่ $ed - 1 \bmod z$ ได้ 37
- $C_{\text{mess}} = \text{mess}^e \bmod n \text{ ----- } > \text{mess}^{13} \bmod 55$
- $\text{Mess} = C_{\text{mess}}^d \bmod n \text{ ----- } > C_{\text{mess}}^{37} \bmod 55$

5.5.2 RC4

RC4 เป็น Stream Cipher ที่สามารถเปลี่ยนขนาดของกุญแจได้ พัฒนาขึ้นในปี ค.ศ. 1987 โดย Ron Rivest อัลกอริทึมทำงานใน Output Feedback Mode (OFB) และใช้การเปลี่ยนค่าแบบสุ่ม (Random Permutation) ตัว Key ไม่ขึ้นต่อ Plaintext หากใช้กุญแจความยาว 8 บิตจะต้องสร้าง S-box เป็นจำนวน $8 * 8$ S-box ได้แก่ $S_0, S_1, \dots, S_{255}$ ขั้นแรกจะทำการสร้าง Random Byte โดยกำหนดค่า I และ J เริ่มต้นให้เป็น 0

- สร้าง Random Byte ด้วยวิธีดังนี้

$$I = (I + 1) \bmod 256$$

$$J = (J + S_i) \bmod 256$$

Swap S_i กับ S_j

$$T = (S_i + S_j) \bmod 256$$

$$K = S_i$$

- จากนั้น Byte K จะถูกนำไป XOR กับ Plaintext เพื่อสร้าง Ciphertext หรือในทางกลับกันนำไป XOR กับ Ciphertext เพื่อหา Plaintext การเข้ารหัสลับแบบนี้สามารถทำได้เร็วและเร็วกว่า DES ประมาณ 10 เท่า
- ทั้งนี้ ในตอนแรกให้กำหนดค่า $S_0, \dots, S_{255}$ แบบเส้นตรง คือ $S_0 = 0, S_1 = 1, \dots, S_{255} = 255$ ตามลำดับ จากนั้นให้ใส่ 256 Byte Array ด้วย Key โดยทำซ้ำ Key จนกว่าจะสามารถใส่ทั้ง Array ได้ครบทั้งหมด ($K_0, \dots, K_{255}$) จากนั้นกำหนดค่า J ให้เป็น 0 แล้วกระทำดังนี้

$$\text{For } I = 0 \text{ to } 255$$

$$J = (J + S_i + K_i) \bmod 256$$

Swap S_i กับ S_j

- ปัจจุบัน RC4 มีสถานะพิเศษสามารถนำไปใช้นอกประเทศสหรัฐอเมริกาได้หากขนาดกุญแจยาว 40 บิตหรือน้อยกว่า

5.5.3 RC2

RC2 เป็น Block Cipher ที่สามารถใช้งานกับกุญแจขนาดใดก็ได้ ได้รับการออกแบบโดย Ronald Rivest จาก RSA Security คำว่า "RC" ย่อมาจาก "Ron's Code" หรือ "Rivest's Cipher" RC2 เป็นอัลกอริทึมที่เร็วกว่า DES และออกแบบมาใช้งานแทน DES ซึ่งสามารถนำมาใช้งานให้ปลอดภัยกว่า DES ได้ด้วยการเลือกกุญแจที่มีความยาวเหมาะสม มีขนาด Block 64 บิตและเมื่อนำไปใช้งานกับซอฟต์แวร์จะทำงานได้เร็วกว่า DES สองถึงสามเท่า ส่วนที่เรียกว่า Salt (มีความยาว 40 ถึง 88 บิต) สามารถนำมาใช้เพื่อป้องกันผู้บุกรุกที่พยายามจะถอดรหัสลับ ส่วน Salt นี้จะเพิ่มต่อกับกุญแจที่ใช้เข้ารหัสลับแล้วนำไปใช้เพื่อเข้ารหัสลับ ผู้ส่งจะส่ง Salt ไปกับข้อมูลที่ได้รับการเข้ารหัสลับด้วย (โดยมิได้เข้ารหัสลับส่วนที่เป็น Salt) ทั้งนี้ RC2 ได้รับความนิยมใช้งานกับการพัฒนา

โปรแกรมที่ต้องการนำไปใช้งานนอกประเทศสหรัฐอเมริกา เพราะ DES มีข้อจำกัดในการนำออกนอกประเทศมากกว่า

มีข้อตกลงระหว่าง Software Publishers Association (SPA) และรัฐบาลสหรัฐอเมริกาเกี่ยวกับ RC2 กล่าวคือมีสถานะพิเศษที่ได้รับอนุมัติให้นำออกนอกประเทศได้ง่ายกว่าแบบอื่น

5.5.4 DES

DES ย่อมาจาก Data Encryption Standard เป็นที่นิยมใช้ในการเข้ารหัสลับข้อมูลแบบใช้กุญแจเดียว (Symmetric Cryptography) กล่าวคือ ฟังก์ชันส่งและรับจะใช้กุญแจเดียวกันในการเข้าและถอดรหัส ทั้งนี้ มีจำนวนกุญแจกว่า 72,000,000,000,000,000 ดอกที่สามารถใช้เพื่อการเข้ารหัสลับได้ ซึ่งในการเข้ารหัสลับแต่ละครั้ง จะเลือกแบบสุ่มขึ้นมาใช้งาน

DES จะใช้กุญแจความยาว 56 บิตสำหรับเข้ารหัสลับข้อมูลทีละ Block ซึ่งมีความยาว 64 บิต ทำงาน 16 รอบ แม้ว่า DES จะเป็นอัลกอริทึมที่แข็งแกร่งมากอยู่แล้ว บางหน่วยงานก็อาจใช้ Triple DES ซึ่งจะใช้กุญแจทั้งสิ้น 3 ดอกในการเข้ารหัสลับ

ในต้นปี 1997 Rivest-Shamir-Adleman เจ้าของทฤษฎีการเข้ารหัสลับอื่นได้เสนอเงินรางวัล 10,000 เหรียญให้กับผู้ที่สามารถแกะข้อมูลที่เข้ารหัสลับแบบ DES ได้ ซึ่งในที่สุดก็มิผู้สามารถทำได้สำเร็จ ค้นพบกุญแจที่ใช้งานหลังจากสุ่มมาเพียง 18,000,000,000,000,000 จาก 72,000,000,000,000,000 ดอก

DES ได้รับการคิดค้นโดย IBM ในปี 1977 และได้นำไปใช้งานในกระทรวงกลาโหมของสหรัฐอเมริกา ได้รับการระบุไว้เป็นมาตรฐานใน ANSI X3.92, มาตรฐาน X3.106 และในมาตรฐาน Federal FIPS 46 และ 81 และถูกห้ามนำออกนอกประเทศ ทั้งนี้ สถาบัน NIST (National Institute of Standards and Technology หน่วยงานหนึ่งในกระทรวงพาณิชย์ของสหรัฐอเมริกา) ได้ระบุว่า DES จะต้องได้รับการตรวจสอบว่าเหมาะสมจะเป็นมาตรฐานหรือไม่ทุกๆ 5 ปี ในเดือนตุลาคมปี 1999 ปรากฏว่า มาตรฐาน FIPS 46-3 ได้รับรองการใช้งาน DES แต่ให้ใช้เฉพาะในระบบเก่าที่เคยใช้งานอยู่เท่านั้น และภายใน 2-3 ปี จะต้องแทนที่การใช้งาน DES และ Triple DES ด้วย AES (Advanced Encryption Standard)

5.5.5 SHA

SHA เป็นอัลกอริทึมในการสร้าง Message Digest ที่พัฒนาโดย NIST สร้างเพื่อใช้งานกับแอปพลิเคชันสร้างลายมือชื่อดิจิทัล ที่ต้องการบีบอัดข้อมูลให้เป็น Digest แล้วนำไปสร้างลายมือชื่อดิจิทัลด้วยกุญแจส่วนตัว

อัลกอริทึมนี้ จะแบ่งข้อมูล Plain Text ออกเป็น Block แต่ละ Block จะต้องมีความยาวบิตข้อมูลเท่ากันคือ 512 บิต และสร้างเป็น Digest ความยาว 160 บิต

บทที่ 6

บทสรุปและแนวทางการพัฒนาในอนาคต

การติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตเป็นการสื่อสารแบบเปิด ข้อมูลที่รับส่งถึงกันต้องเดินทางผ่านเครือข่ายคอมพิวเตอร์ขนาดใหญ่ จึงมีความเสี่ยงสูงในเรื่องความปลอดภัยของข้อมูล เทคโนโลยีระบบรหัสจึงได้ถูกนำมาประยุกต์ใช้งาน ทั้งในรูปของการเข้ารหัสและถอดรหัสหรือการสร้างลายมือชื่อดิจิทัล เพื่อรักษาความมั่นใจในความปลอดภัยของข้อมูลทั้ง 4 ด้าน ได้แก่ การรักษาความลับของข้อมูลอิเล็กทรอนิกส์ (Data Confidentiality) การตรวจสอบความแท้จริงของข้อมูลอิเล็กทรอนิกส์ (Data Integrity) การยืนยันตัวบุคคลผู้สร้างหรือผู้ส่งข้อมูลอิเล็กทรอนิกส์ (Authentication) และการยืนยันว่าผู้ประกอบธุรกรรมไม่สามารถปฏิเสธความรับผิดชอบต่อธุรกรรมอิเล็กทรอนิกส์ได้ (Non-repudiation)

เทคโนโลยีกุญแจสาธารณะ (Public-Key Infrastructure - PKI) และใบรับรองอิเล็กทรอนิกส์มาตรฐาน X.509 ได้รับการนำมาประยุกต์ใช้งาน ทั้งเพื่อการพิสูจน์ทราบตัวตน (Authentication) การให้สิทธิการใช้งาน (Authorization) หรือการทำงานเกี่ยวกับระบบรหัส สำหรับในโครงงานพัฒนาระบบงานนี้ ได้นำใบรับรองอิเล็กทรอนิกส์มาใช้ในการสร้างลายมือชื่อดิจิทัลเข้ารหัสและถอดรหัสลับเพิ่มข้อมูล เพื่อเสริมสร้างความมั่นใจในความปลอดภัยของเพิ่มข้อมูลที่ผู้ใช้งานจัดเก็บไว้บนเครื่องคอมพิวเตอร์ และส่งเสริมให้เกิดการใช้งานใบรับรองอิเล็กทรอนิกส์ในวงกว้าง

อย่างไรก็ตาม เรื่องการรักษาความปลอดภัยมักเป็นห่วงลูกโซ่ ความปลอดภัยของระบบ PKI นั้นมีจุดเชื่อมต่อหลายจุดและแน่นอนว่า ทั้งหมดนั้นไม่ได้ถูกปกป้องด้วยระบบรหัสหรือการเข้ารหัส กล่าวได้ว่า ยังคงมีจุดอ่อนและยังต้องมีการทำงานของคนเข้าไปเกี่ยวข้องด้วย จุดอ่อนสำคัญเป็นไปได้ตั้งแต่ การที่ผู้ใช้งานต้องระมัดระวังการใช้งานคอมพิวเตอร์มิให้ผู้อื่นสามารถลักลอบหรือแอบใช้งานคอมพิวเตอร์ที่จัดเก็บใบรับรองอิเล็กทรอนิกส์ของตนได้ การรักษาข้อมูลของตนให้เป็นความลับ เป็นต้น

สำหรับแนวทางการพัฒนาต่อไปในอนาคต อาจมีการใช้อัลกอริทึมที่แข็งแกร่งมากกว่านี้ หรือสามารถนำไปประยุกต์ใช้งานกับแอปพลิเคชันด้านอื่นๆ ที่ต้องการคุณสมบัติการเข้ารหัสและถอดรหัสลับ รวมถึงการสร้างลายมือชื่อดิจิทัลให้กับเพิ่มข้อมูลได้

บรรณานุกรม

- กรองรัตน์ คามตะศิลา และ โชติพัชร ภรณ์วลัย. 2545. “การออกแบบและพัฒนา OpenSSL/ openCA เพื่อเป็นหน่วยผู้ประกอบการรับรองระบบปฏิบัติการลินุกซ์”. ในการประชุมวิชาการ The First National Conference on Electronic Business หัวข้อ “Enlightening : Successful e-Business Model” (ธุรกิจผ่านสื่ออิเล็กทรอนิกส์ : รูปแบบและกลยุทธ์สู่ความสำเร็จ) ครั้งที่ 1. กรุงเทพฯ: มหาวิทยาลัยธรรมศาสตร์.
- ฝ่ายเทคโนโลยีความปลอดภัยของข้อมูล, สำนักบริการเทคโนโลยีสารสนเทศภาครัฐ. 2544. เอกสาร แจกเกี่ยวกับ PKI. กรุงเทพฯ.
- Aaron Margosis, Patrick Arnold. 1999. **Microsoft CryptoAPI Overview and an (fine) Example of CAPI Application Development**. [Online]. Available: <http://csrc.nist.gov/pki/twg/presentationws/twg-99-51.pdf>.
- Bruce Schneier, Carl M. Ellison. 2001. **Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure**. [Online]. Available: <http://www.counterpane.com/pki-risks-ft.txt>.
- Bruce Schneier. 1996. **Applied Cryptography**. Second Edition. Canada: Katherine Schowalter.
- Christopher M. King, Curtis E. Dalton and T. Ertem Osmanoglu, **Security Architecture : Design, Deployment & Operations**. Osborne: McGraw-Hill, 2001.
- Holt Sorenson. 2001, August. **An Introduction to OpenSSL (Part One)**. [Online]. Available: <http://www1.securityfocus.com/frames/?focus=sun&content=/focus/sun/articles/openssl.html>.
- Holt Sorenson. 2001, September. **An Introduction to OpenSSL, Part Two: Cryptographic Functions Continued**. [Online]. Available: <http://online.securityfocus.com/infocus/1462>.
- Holt Sorenson. 2001, September 19. **An Introduction to OpenSSL, Part Three: PKI- Public Key Infrastructure**. [Online]. Available: <http://online.securityfocus.com/infocus/1466>.
- John Lambert. May 2001. **Introducing CAPICOM**. [Online]. Available: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/intcapicom.asp>
- Kenneth Ives. 2001. **CryptoAPI Demo Version 1.7**. [Online]. Available: <http://www.freevbcode.com/ShowCode.Asp?ID=2131>

Microsoft Platform SDK Security. 2002, October. **CAPICOM Reference**. [Online]. Available:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/capicom_reference.asp

Paul E. Jones, Donald E. Eastlake. 2001, April. **US Secure Hash Algorithm 1 (SHA1)**. [Online].

Available: <http://www.securitytechnet.com/resource/ietf/ind-draft/draft-eastlake-sha1-02.txt>

R. Rivest. 1992, April. **The MD5 Message-Digest Algorithm**. [Online]. Available:

<http://www.rfc-editor.org/rfc/rfc1321.txt>

Ryan M. Hurst. 2002. **Developing PKI Application Using CAPICOM**. [Online]. Available:

http://asp01sea.activate.net/microsoft/teched2002/publish/SEC302_files/default.htm#profile=1

SearchSecurity.com Definitions. 2001, January. **Data Encryption Standard**. [Online]. Available:

http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213893,00.html

SearchSecurity.com Definitions. 2002, March. **Advanced Encryption Standard**. [Online].

Available: http://searchsolaris.techtarget.com/sDefinition/0,,sid12_gci344759,00.html

Steve Burnett, Stephen Paine, **RSA Security's Official Guide to Cryptography**. Osborne:

McGraw-Hill, 2001.

Symeon (Simos) Xenitellis. 2000. **The Open-source PKI Book A guide to PKIs and Open-source Implementations**. [Online]. Available:

<http://ospkibook.sourceforge.net/docs/OSPki-2.4.6/OSPki/ospki-book.htm>.

ภาคผนวก ก

1. การติดตั้งและปรับแต่งส่วนเซิร์ฟเวอร์ เตรียมสภาพแวดล้อมในการทำงาน

ลำดับ	ขั้นตอนการทำงาน	รายละเอียด
1	ติดตั้งระบบปฏิบัติการ Linux Redhat 7.0	
2	ทำการปรับแต่งค่ามิให้ระบบเปิดโปรแกรมที่ไม่ต้องการโดยอัตโนมัติเมื่อเปิดเครื่อง	ปิดโปรแกรมที่ไม่จำเป็นเช่น Sendmail, Crond เป็นต้น
3	ติดตั้ง Secure Shell Server SSH 3.2.0	ดาวน์โหลดได้จาก http://ftp.azc.uam.mx/mirrors/ssh/
4	ติดตั้งซอฟต์แวร์ต่างๆ ที่จำเป็นเพื่อเป็น API ให้ PHP สามารถเรียกใช้งานได้	MHASH 0.8.16, MYSQL 3.23.52, OPENLDAP 2.0.25, OPENSLL 0.9.7, LIBMCRYPT 2.5.3, MOD_SSL
5	ติดตั้งและปรับแต่งค่าเพื่อใช้งาน OpenSSL	ดาวน์โหลดได้จาก http://www.openssl.org
6	ติดตั้งและปรับแต่งค่าเพื่อใช้งาน OpenLDAP	ดาวน์โหลดได้จาก http://www.openldap.org
7	ติดตั้ง APACHE 1.3.27, mod_ssl-2.8.12-1.3.27, OpenSSL 0.9.7 และ PHP 4.1.2	ดาวน์โหลดได้จาก http://www.apache.org , http://www.php.net
8	ติดตั้ง OpenCA เพื่อสร้างเป็นหน่วยผู้ให้บริการออกใบรับรอง	ดาวน์โหลดได้จาก http://www.openca.org
9	จากนั้นติดตั้งแก้ไข Virtual Host ใน Apache Web Server เพื่อให้สามารถเรียกใช้งานหน้าเว็บของ OpenCA ได้	สร้าง Directory “ca” ภายใต้อ /usr/local/apache/cgi-bin/ แล้วนำไฟล์ ca ของ OpenCA วางในไว้ใดเรกทอรีนี้เพื่อให้เว็บของ OpenCA เรียกใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การออกแบบและพัฒนาในส่วนไคลเอ็นท์

ลำดับ	ขั้นตอนการทำงาน	รายละเอียด
1	ดาวน์โหลด Microsoft Cryptography API ได้จาก SDK Update Site	http://www.microsoft.com/msdownload/platformsdk/sdkupdate/
2	ดาวน์โหลด CAPICOM (Platform SDK Redistributable) ได้จาก Platform SDK Redistributable: CAPICOM (ActiveX Control ขนาด 440 KB)	http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=860EE43A-A843-462F-ABB5-FF88EA5896F6
3	ศึกษาวิธีการใช้งาน CAPICOM	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/creating_and_receiving_enveloped_data_messages_in_capicom.asp
4	ศึกษาวิธีการใช้งานจาก CAPICOM Discussion List	http://discuss.microsoft.com/archives/index.html
5	การลงทะเบียน (Register) เพื่อใช้งาน CAPICOM	ไปที่ Command Prompt ไปยังไดเรกทอรีที่เก็บไฟล์ CAPICOM.dll อยู่ จากนั้นพิมพ์คำสั่ง regsvr32 CAPICOM.dll
6	ติดตั้ง CAPICOM เพื่อใช้งาน	เพิ่ม CAPICOM.DLL ไว้ที่ Project References ของโปรแกรม Microsoft Visual Basic
7	ติดตั้ง Xceed เพื่อใช้งาน XceedZip	ดาวน์โหลดได้จาก http://www.xceedsoft.com/cs/

ประวัติผู้เขียน

ชื่อผู้เขียน

นางกรรณรัตน์ คามตะศิลา

วันเดือนปีเกิด

8 สิงหาคม 2520

สถานที่เกิด

จังหวัดสมุทรปราการ

วุฒิการศึกษาระดับปริญญาตรี

วท.บ. (ศาสตร์คอมพิวเตอร์)

สถานที่สำเร็จการศึกษา

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

ปีที่สำเร็จการศึกษาในระดับปริญญาตรี

2541



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้