การใช้ลายมือชื่อดิจิทัลเพื่อพิสูจน์ตัวจริงใน Microsoft SQL Server

Digital Signature for Authentication in Microsoft SQL Server

โดย

นางสาว ธิดารัตน์ ขุนเอม

รหัส 44067041

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การใช้ลายมือชื่อดิจิทัลเพื่อพิสูจน์ตัวจริงใน Microsoft SQL Server
นักศึกษา	นางสาวธิดารัตน์ ขุนเอม
อาจารย์ที่ปรึกษา	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

การรักษาความปลอดภัยเป็นเรื่องที่สำคัญเมื่อต้องการนำ Microsoft SQL Server ไปใช้งานบนเครือข่ายที่ไม่มีความปลอดภัย จึงมีการนำเอาเทคโนโลยีของลายมือชื่อดิจิทัล ซึ่งมีความสามารถในการพิสูจน์ตัวจริงรายบุคคลได้มาประยุกต์ใช้ในการเข้าสู่การใช้งานในฐานข้อมูล Microsoft SQL Server ซึ่งจะนำไปเป็นเครื่องมือช่วยในการรักษาความปลอดภัยในด้านการเข้าถึงระบบฐานข้อมูลของผู้ใช้ให้มีความแข็งแกร่งมากขึ้น โดยการออกแบบระบบงานรักษาความปลอดภัย มุ่งเน้นไปที่การพิสูจน์ตัวจริงรายบุคคล การกำหนดสิทธิ์ในการเข้าใช้งานระบบฐานข้อมูล รวมไปถึงการตรวจสอบการใช้งานของผู้ใช้ ซึ่งเป็นผลมาจากการพิสูจน์ตัวจริงโดยใช้ลายมือชื่อดิจิทัล เพื่อให้ฐานข้อมูล Microsoft SQL Server มีความปลอดภัยที่แข็งแรงมากขึ้น

Title	Digital Signature for Authentication in Microsoft SQL Server
Student	Miss.Tidarat Khunem
Advisor	Chanboon Sathiviriyawong,Ph.D
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

Abstart

In the unsecure network,Security in Relational Database is important for Microsoft SQL Server. Technology of digital signature can prove the authentication method that used in SQL Server system. Digital signature is used to protect SQL Server for stong security. The system will be designed by focusing on the authentication of user , authorization and audit mechanism built into the system from using digital signature. They make the Microsoft SQL Server have the strong authentication system.

กิตติกรรมประกาศ

โครงการศึกษากรณีพิเศษนี้ ผู้เขียนได้รับการสนับสนุนและความช่วยเหลือในการให้คำแนะนำจากคณาจารย์ และเพื่อน ๆ เป็นอย่างดียิ่ง จึงทำให้โครงการศึกษากรณีพิเศษสำเร็จลุล่วงได้ด้วยดี ผู้เขียนขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการทำโครงการศึกษากรณีพิเศษนี้ โครงการนี้เกิดขึ้นมาได้โดยเฉพาะ ดร. จันทร์บูรณ์ สถิตวิริยวงศ์ เป็นอาจารย์ที่ควบคุมโครงการนี้ ซึ่งได้ให้คำแนะนำ ข้อคิดเห็น และแนวทางที่เป็นประโยชน์ในการทำโครงการ รวมทั้งตรวจสอบ หาจุดบกพร่องเพื่อทำการแก้ไขในทุกขั้นตอนการดำเนินการเป็นอย่างดี

ผู้เขียนขอกราบขอบพระคุร คุณพ่อ คุณแม่ ที่เห็นความสำคัญของการศึกษาเล่าเรียนเป็นอย่างยิ่ง และคอยเป็นกำลังใจที่สำคัญมาโดยตลอด จนกระทุ่งโครงการศึกษากรณีพิเศษนี้สำเร็จลุล่วงอย่างสมบูรณ์

ธิดารัตน์ ขุนเอม

กุมภาพันธ์ 2546

สารบัญ

หน้าที่

สารบัญ(ต่อ)

หน้าที่

บทที่ 1
บทนำ

1.1 ความเป็นมา

ฐานข้อมูล Microsoft SQL Server เป็นระบบการจัดการดาต้าเบสแบบรีเลชันแนล (Relational Database Management System) สามารถติดตั้งและทำงานได้กับระบบปฏิบัติการ Windows 95/98, Windows NT 4.0 และ Windows 2000 โดย SQL Server ออกแบบมาให้ทำงานในลักษณะที่เป็น Client/Server Database จึงสามารถรองรับการทำงานจากเครื่องไคลเอนต์ได้เป็นจำนวนมากที่ต่อผ่านทางระบบเครือข่ายหรือเน็ตเวิร์กเข้ามา นอกจากนี้ SQL Server ยังถูกออกแบบมาเพื่อใช้งานที่เป็นแบบ Stand-Alone Database ได้ด้วย โดยติดตั้งลงบนเครื่องที่ใช้ระบบปฏิบัติการ Windows 95/98 การที่ SQL Server เป็นระบบจัดการดาต้าเบสแบบ Client-Server Relational Database ทำให้ช่วยเพิ่มประสิทธิภาพในการทำงาน และยังมีระบบจัดการเป็นแบบควบคุมจากศูนย์กลาง (Centralized Management)

สำหรับ SQL Server ประกอบด้วยส่วนต่าง ๆ ดังนี้

1.1.1 Server เป็นเครื่องที่ติดตั้งโปรแกรมการทำงานของ SQL Server โดย SQL Server ทางฝั่งนี้ทำหน้าที่จัดเก็บรวบรวม ค้นหา เรียงลำดับ เรียกดู และจัดการข้อมูล นอกจากนี้การที่ SQL Server เป็น Client-Server Relational Database ส่วนที่เป็นระบบจัดการดาต้าเบส และไฟล์ต่าง ๆ ที่เกี่ยวข้องกับดาต้าเบสทั้งหมดก็ถูกเก็บอยู่บนเครื่องที่เป็นเซิร์ฟเวอร์ด้วย

1.1.2 Client เป็นเครื่องที่ติดตั้งโปรแกรมใช้งาน ที่พัฒนาด้วยภาษาต่าง ๆ เช่น Visual Basic, Delphi และเชื่อมต่อกับ SQL Server ทางฝั่ง Server ได้ ทั้งนี้เครื่องที่เป็นเครื่องไคลเอนต์อาจเป็นแพลทฟอร์ม(platform) ใดก็ได้ โปรแกรมทางฝั่งเครื่องไคลเอนต์นี้ ทำหน้าที่ส่งและรับข้อมูลจากฐานข้อมูล และมีโปรแกรมสำหรับเรียกดูและจัดการข้อมูล โดยจะต้องมีสิทธิ์ในการเรียกใช้ข้อมูลบนเซิร์ฟเวอร์ได้

1.1.3 เครือข่ายการสื่อสาร (Network) การติดต่อระหว่างเซิร์ฟเวอร์และไคลเอนต์จะอาศัยผ่านเครือข่ายการสื่อสาร(Communication Network) ซึ่งมีบทบาทสำคัญในการทำงานให้เซิร์ฟเวอร์และไคลเอนต์สามารถแลกเปลี่ยนข้อมูลและการรับหรือส่งคำสั่งระหว่างกันได้

จะเห็นว่าข้อมูลที่ส่งออกไปยังเครือข่าย อาจไม่มีความปลอดภัย ดังนั้นจึงต้องมีการรักษาความปลอดภัยในการใช้ข้อมูลจากฐานข้อมูลที่ดี ในการให้บริการข้อมูลของฐานข้อมูลเชิงสัมพันธ์ SQL Server ที่มีผู้ใช้เป็นจำนวนมากเข้ามาใช้ข้อมูลในฐานข้อมูล ดังนั้นจึงต้องมีการระบุว่าผู้ใช้แต่ละคนและกำหนดสิทธิ์ในการเข้าใช้ฐานข้อมูลของแต่ละบุคคล เพื่อเพิ่มความปลอดภัยให้กับข้อมูลในฐานข้อมูลมากขึ้น หากมีการนำเทคโนโลยีการรักษาความปลอดภัยเข้ามาใช้ ซึ่งนำส่วนของลายมือชื่อดิจิทัลเข้ามาใช้สำหรับการพิสูจน์ตัวจริงของแต่ละบุคคล

1.2 เป้าหมาย

การนำลายมือชื่อดิจิทัลเข้ามาประยุกต์ใช้ในระบบฐานข้อมูล Microsoft SQL Server นั้น เพื่อสร้างระบบการพิสูจน์ตัวจริง และการกำหนดสิทธิ์ให้กับระบบฐานข้อมูลให้แข็งแกร่งขึ้น รวมไปถึงผู้ใช้สามารถตรวจสอบการใช้งานของตนเองได้ โดยนำลายมือชื่อดิจิทัลมาประยุกต์ใช้

1.3 วัตถุประสงค์ในการสร้างความปลอดภัย

1.3.1 เพื่อให้ผู้ใช้ฐานข้อมูลสามารถสร้างกุญแจขึ้นเองได้

1.3.2 เพื่อให้ระบบฐานข้อมูลมีการตรวจสอบลายมือชื่อดิจิทัลจากผู้ใช้ได้

1.3.3 เพื่อให้ระบบสามารถระบุตัวจริงของผู้ใช้ได้

1.3.4 เพื่อให้ระบบสามารถกำหนดสิทธิ์การใช้งานของผู้ใช้ได้

1.3.5 เพื่อให้ผู้ใช้สามารถตรวจสอบการใช้งานของตัวเองได้

1.4 ขั้นตอนในการสร้างระบบ

การสร้างระบบพิสูจน์ตัวจริงสำหรับระบบนี้เป็นการสร้างในรูปแบบ Client/Server บนเครือข่าย LAN ซึ่งมีขั้นตอนในการสร้างระบบดังนี้

1.4.1 การศึกษาวิธีการของวิทยาการเข้ารหัสลับ (Cryptography) ซึ่งมีการศึกษาวิธีการต่าง ๆ ที่เกี่ยวข้องกับการพิสูจน์ตัวจริงดังนี้

- PKI(Public-key Infrastructure)
- ลายมือชื่อดิจิทัล (Digital Signature)
- CryptoAPI

1.4.2 การวิเคราะห์และออกแบบระบบการพิสูจน์ตัวจริง ซึ่งมีการศึกษาการทำงานของส่วนต่าง ๆ ดังนี้

- การศึกษาฐานข้อมูลเชิงสัมพันธ์ ในที่นี้ใช้ Microsoft SQL Server

- การศึกษา WCCO Object สำหรับการใช้ CryptoAPI บน Visual Basic
- การศึกษา Visual Basic

1.5 องค์ประกอบและเครื่องมือที่ใช้ในการสร้างระบบ

1.5.1 องค์ประกอบทางด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ เชื่อมต่อเครือข่ายสำหรับทำเป็นฐานข้อมูลเซริฟเวอร์
- เครื่องคอมพิวเตอร์เป็นเครื่องลูกข่ายเชื่อมต่อมายังเครื่องเซริฟเวอร์

1.5.2 องค์ประกอบทางด้านซอฟท์แวร์

- ระบบปฏิบัติการเครือข่าย (Network Operating System)
  - Microsoft Windows2000
- ฐานข้อมูล
  - Microsoft SQL Server
- เครื่องมือและซอฟท์แวร์ที่ใช้ในการสร้างระบบ
  - Microsoft Enhanced 128 bit
  - Microsoft Visual Basic 6.0
  - WCCO Object

1.6 ประโยชน์ที่คาดว่าจะได้รับ

การสร้างขั้นตอนการพิสูจน์ตัวจริงให้กับ Microsoft SQL Server ให้แก่ผู้ปฏิบัติการทั้งผู้ใช้และผู้ให้บริการข้อมูล เพื่อเพิ่มความมั่นใจว่ามีเพียงผู้ใช้ที่ได้รับการอนุญาติเท่านั้นที่สามารถเข้ามาใช้ในฐานข้อมูลได้

บทที่ 2

แผนดำเนินการและขอบเขตระบบงาน

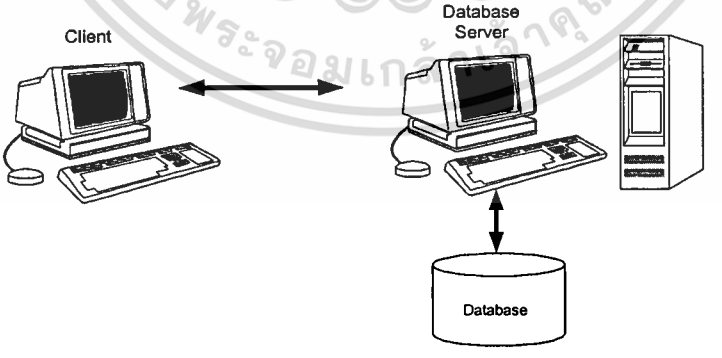
2.1 การรักษาความปลอดภัยของระบบ

การรักษาความปลอดภัยของฐานข้อมูล SQL Server ซึ่งระบบความปลอดภัยใน SQL Server มี login Authentication ซึ่งเป็นวิธีการตรวจสอบผู้ใช้ที่ล็อกอินเข้ามาในระบบ SQL Server แบ่งเป็น 2 แบบด้วยกัน ดังนี้

2.1.1 SQL Server Authentication เป็นระบบการรักษาความปลอดภัยที่ SQL Server ใช้รหัสล็อกอินที่สร้างด้วย SQL Server เอง ซึ่งวิธีนี้ ไม่มีการตรวจสอบผู้ใช้โดย Windows Login

2.1.2 Windows Authentication เป็นระบบการรักษาความปลอดภัยที่กำหนดได้โดยผู้ใช้ของวินโดว์ ซึ่งสามารถล็อกอินเข้าสู่ระบบ SQL Server ได้เลยโดยไม่ต้องมีการล็อกอินอีกครั้ง ซึ่งเป็นการลดภาระในการสร้างรหัสล็อกอินหลายชื่อให้กับผู้ใช้คนเดียว

จะเห็นว่าในส่วนของ SQL Server Authentication ที่ใช้รหัสล็อกอินที่สร้างด้วย SQL Server เอง หากนำวิธีการพิสูจน์ตัวจริงอย่างอื่นเข้ามาร่วม นั่นคือ การนำลายมือชื่อดิจิทัลเข้ามาเป็นส่วนหนึ่งของการพิสูจน์ตัวจริงในการเข้าใช้ฐานข้อมูลที่อยู่ใน SQL Server จะทำให้ขั้นตอนการพิสูจน์ตัวจริงของ SQL Server มีความแข็งแรงมากขึ้น



รูปที่ 2.1 แสดงการทำงานของ Microsoft SQL Server

2.2 แผนการดำเนินการศึกษา

สำหรับระบบการพิสูจน์ตัวจริง เป็นการทำให้ฝั่งผู้ให้บริการฐานข้อมูลสามารถพิสูจน์ข้อบ่งชี้ของผู้ขอรับข้อมูลได้ ซึ่งในการดำเนินการศึกษาครั้งนี้ จะนำลายมือชื่อดิจิทัลเข้ามาประยุกต์ใช้ในการพิสูจน์ตัวจริง เพราะลายมือชื่อดิจิทัลนั้นมีคุณสมบัติในการพิสูจน์ตัวจริงได้ โดยมีขั้นตอนในการศึกษา ซึ่งมารายละเอียดต่าง ๆ ดังนี้

2.2.1 ศึกษาและวิเคราะห์ ระบบความปลอดภัยในด้านการพิสูจน์ตัวจริงของ SQL Server เพื่อหาความปลอดภัยในด้านการพิสูจน์ตัวจริงวิธีอื่น เพื่อเพิ่มเติมความแข็งแกร่งในการพิสูจน์ตัวจริงให้แก่ SQL Server ให้มากยิ่งขึ้น

2.2.2 ค้นหาวิธีการของการพิสูจน์ตัวจริงที่จะทำให้แข็งแกร่งมากขึ้น โดยศึกษาจากทฤษฎีวิทยาการเข้ารหัสลับ (Cryptography)

2.2.3 ดำเนินการออกแบบระบบการพิสูจน์ตัวจริง ซึ่งพิจารณาในส่วนของการลงทะเบียนการใช้งานในฐานข้อมูลของผู้ใช้ รวมไปถึง วิธีการเก็บกุญแจคู่ และข้อมูลของผู้ใช้ที่ได้มีการลงทะเบียนไว้

2.2.4 ดำเนินการสร้างระบบการพิสูจน์ตัวจริงใหม่ ซึ่งเป็นขั้นตอนในการเขียนโปรแกรม โดยพิจารณาถึงภาษาที่ใช้ และจัดสร้างระบบตามที่ออกแบบเอาไว้

2.3 ขอบเขตการสร้างระบบงาน

ในการสร้างระบบการพิสูจน์ตัวจริงขึ้นสำหรับฐานข้อมูล Microsoft SQL Server มีขอบเขตของการสร้างระบบดังนี้

2.3.1 การสร้างระบบงานอยู่ในรูปแบบของ Client/Server แบบ 2-tier ประกอบด้วย ฐานข้อมูลเซิร์ฟเวอร์ ใช้ SQL Server ซึ่งในการสร้างระบบงานนี้เป็นการพิสูจน์ตัวจริงของผู้ใช้ โดยปฏิบัติการที่เครื่องของ client ร่วมกับการปฏิบัติการที่ฐานข้อมูลเซิร์ฟเวอร์

2.3.2 การปฏิบัติการของการพิสูจน์ตัวจริง มีส่วนประกอบดังนี้

- การสร้างกุญแจคู่ของผู้ใช้
- การลงทะเบียนของผู้ใช้
- การแลกเปลี่ยนข้อมูลต่าง ๆ ของผู้ใช้ระหว่าง Client กับฐานข้อมูล SQL Server
- การตรวจสอบตัวบุคคลของผู้ใช้

2.3.3 การตรวจสอบสิทธิ์ในการเข้าใช้ฐานข้อมูล SQL Server เป็นการทำให้ระบบสามารถตรวจสอบสิทธิ์ของผู้ใช้ก่อนที่จะอนุญาติให้ผู้ใช้เข้ามาใช้งานข้อมูลในฐานข้อมูล SQL Server ของแต่ละคนได้

2.3.4 การตรวจสอบการใช้งาน เพื่อทำให้ระบบมีความสามารถบอกรายละเอียดในการใช้งานของผู้ใช้แต่ละคนได้

2.4 ประสิทธิภาพของระบบ

สำหรับความปลอดภัยที่แข็งแรงขึ้น ผลที่ตามมา คือ ทำให้เกิดความยุ่งยากต่อการใช้งานมากขึ้น ทั้งในแง่ของความยากต่อการใช้งาน และเวลาที่ใช้ในการเข้ารหัสด้วย ดังนั้นจึงต้องคำนึงถึงเวลาที่ใช้สำหรับในการสร้างกุญแจ และการลงนาม เข้ารหัสข้อมูล เพราะความซับซ้อนในการสร้างกุญแจและการเข้ารหัสนั้นมีมาก ทำให้เวลาการทำงานในแต่ละขั้นตอนใช้เวลาที่นาน แต่สิ่งที่ได้ ก็คือสามารถเพิ่มเติมความปลอดภัยต่อฐานข้อมูลให้มากขึ้น

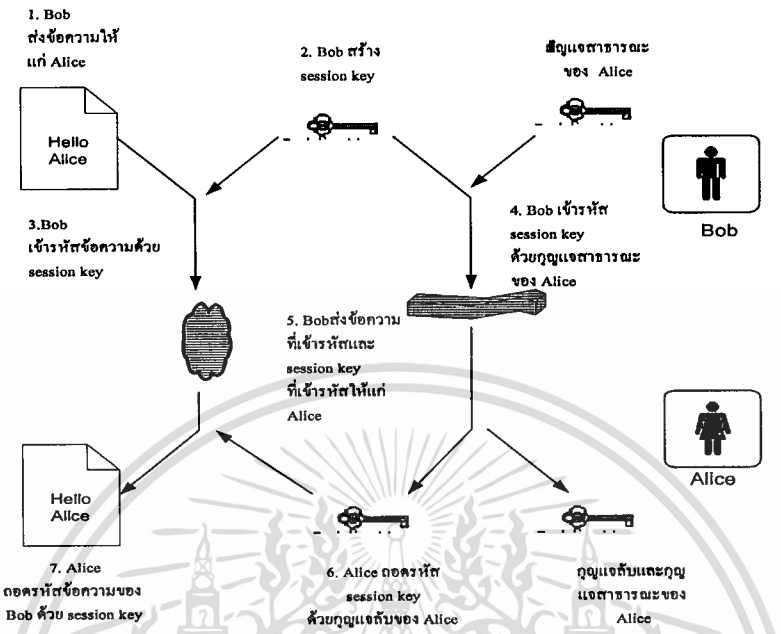
บทที่ 3
ทฤษฎีที่ใช้ในการสร้างระบบพิสูจน์ตัวจริง

3.1 วิทยาการเข้ารหัสลับ (Cryptography)

วิทยาการเข้ารหัสลับ เป็นวิชาที่ว่าด้วยการพัฒนาขั้นตอน วิธี หรืออัลกอริซึมสำหรับการปกปิดเนื้อหาของข้อความจากทุกคน ยกเว้นผู้ส่งและผู้รับ รวมไปถึงการทวนสอบความถูกต้องของข้อความให้กับผู้รับ ซึ่งวิทยาการเข้ารหัสลับถูกนำมาใช้ในระบบคอมพิวเตอร์และเครือข่าย เพื่อทำให้การสื่อสารบนเครือข่ายมีความปลอดภัย ซึ่งในระหว่างการส่งข้อมูลในเครือข่ายนั้น ข้อมูลที่ถูกส่งไปอาจถูกดักจับ หรือถูกเปลี่ยนแปลงข้อมูล โดยที่ผู้รับไม่ทราบ ดังนั้น จึงต้องหาวิธีการเพื่อให้ผู้รับและผู้ส่งมั่นใจว่า ข้อมูลที่ส่งไปยังเครือข่ายนั้นมีความปลอดภัย ซึ่งวิทยาการเข้ารหัสลับสามารถแก้ไขสิ่งที่กล่าวมาได้ และอีกสิ่งหนึ่งที่ต้องพิจารณาในการติดต่อสื่อสารระหว่างกัน คือ หากมีผู้ประสงค์ร้ายต้องการปลอมตัวมาเป็นผู้รับข้อมูล หรือส่งข้อมูลไปให้ผู้รับที่ผิดคน ทำให้ข้อมูลที่ต้องการปกปิดให้เป็นความลับนั้นเปิดเผยต่อผู้อื่นได้ จึงต้องมีวิธีการเพื่อให้มั่นใจว่าข้อมูลส่งถึงผู้รับตัวจริง ดังนั้น จึงมีวิธีการในการพิสูจน์ตัวจริง และข้อมูลต้องไม่ถูกเปลี่ยนแปลงระหว่างทาง

การเข้ารหัสลับข้อมูล เป็นวิธีการรักษาความลับให้กับข้อมูลที่ต้องการปกปิดเป็นความลับ ซึ่งการเข้ารหัสลับนี้เป็นกระบวนการแปลงรูปแบบข้อมูลที่เป็นภาษา หรือข้อความกระจ่าง (plaintext) ไปอยู่ในรูปแบบที่ไม่เป็นภาษา หรือข้อความเข้ารหัส (ciphertext) ซึ่งในการเข้ารหัสลับข้อมูลขึ้นอยู่กับอัลกอริซึมที่ใช้ในการเข้ารหัสและกุญแจ ซึ่งกุญแจที่ใช้เป็นรหัสดิจิทัลที่ใช้ในการเข้ารหัส ถอดรหัส และลงลายชื่อแบบดิจิทัลให้กับข้อมูล

(Microsoft Corporation.2002)ปัจจุบันมีการนำวิธีกุญแจสาธารณะ(Public-key Infrastructure: PKI) เข้ามาใช้ในการรักษาความปลอดภัยของข้อมูลระหว่างส่งไปในเครือข่ายที่ไม่มีความปลอดภัย หลักการของกุญแจสาธารณะ มีการใช้กุญแจสำหรับเข้ารหัสและถอดรหัสที่แตกต่างกัน นั่นคือ จะต้องมีการสร้างกุญแจคู่เพื่อใช้ในการเข้ารหัสข้อความที่จะต้องการส่ง ซึ่งการเข้ารหัสแบบนี้ อาจเรียกได้ว่าเป็นการเข้ารหัสแบบไม่สมมาตร(Asymmetric key) นั่นเอง แต่ในการเข้ารหัสแบบนี้ จำเป็นที่จะต้องใช้เวลาในการเข้าและถอดรหัสที่นาน ดังนั้นการนำวิธีกุญแจสาธารณะเข้ามาใช้ควรคำนึงถึงขนาดของข้อมูลที่ต้องการเข้ารหัสด้วย หากเป็นข้อมูลขนาดใหญ่ซึ่งทำให้การเข้ารหัสด้วยกุญแจคู่นี้ใช้เวลาที่นาน วิธีการทำงานของเทคโนโลยีกุญแจสาธารณะดูได้ดังรูปที่ 3.1



รูปที่ 3.1 วิธีการของกุญแจสาธารณะ

3.2 การพิสูจน์ตัวจริงและการไม่ปฏิเสธรับผิด (Authentication and Nonrepudiation)

การรักษาความปลอดภัยไม่ใช่เพียงแค่เข้ารหัสข้อมูลเพื่อให้ข้อมูลไปถึงยังผู้รับที่มีกุญแจที่สามารถถอดรหัสได้เท่านั้น แต่จำเป็นจะต้องมีการพิสูจน์ตัวจริงสำหรับผู้รับและผู้ส่งข้อความนั้น ๆ ถูกส่งมายังผู้รับ หรือส่งมาจากผู้ส่งตัวจริงหรือไม่ ดังนั้นการนำวิธีการของการเข้ารหัสแบบกุญแจสาธารณะเข้ามาร่วมกับ Hash function ทำให้สามารถยืนยันข้อบ่งชี้ของผู้ส่งได้ว่า ผู้ส่งได้ว่า ผู้ส่งเป็นใคร และผู้ส่งไม่สามารถปฏิเสธรับผิดว่าไม่ได้เป็นคนส่งข้อความนั้นมา เพราะมีเพียงผู้ส่งคนเดียวที่สามารถลงนามข้อมูลได้โดยกุญแจลับของตัวเอง

จากที่กล่าวมาข้างต้น อาจเรียกได้ว่าเป็นการลงนามข้อมูลแบบดิจิทัล ซึ่งข้อมูลที่ถูกลงนามเรียกได้ว่า เป็นลายมือชื่อดิจิทัล (Digital Signature) ซึ่งวิธีการของ Hash function คือ เป็นการย่อข้อความขนาดใด ๆ ให้มีขนาดคงที่ ซึ่ง Hash function ถูกใช้ในขั้นตอนของการลงลายมือชื่อดิจิทัลด้วย อัลกอริซึมลายมือชื่อดิจิทัล ซึ่งค่าแฮช(hash value) ถูกสร้างขึ้นด้วย ฟังก์ชั่นแฮช(Hash Function) ได้ค่าดังนี้คือ

h = H(M)

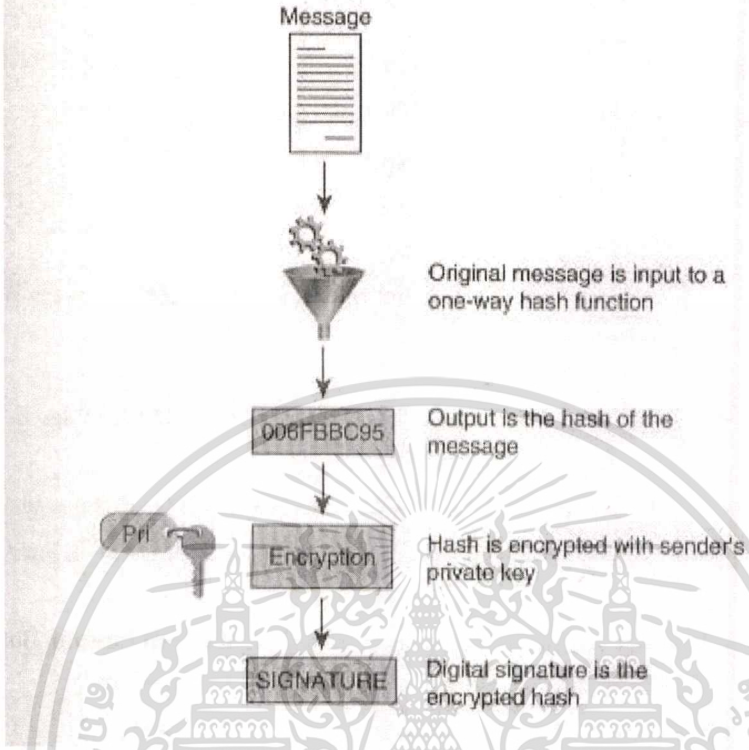
H	หมายถึง	Hash Function
h	หมายถึง	hash value เป็น ค่าแฮชที่มีความยาวคงที่
M	หมายถึง	message

ค่าแฮชถูกนำไปผนวกกับข้อความที่จะถูกส่งจากต้นทาง ซึ่งคุณสมบัติของฟังก์ชั่นแฮช ที่สำคัญคือ ค่าแฮชใด ๆ จะไม่สมารถคำนวนย้อนกลับเพื่อหาค่า M ที่ทำให้ H(M) = h ได้ จึงเรียกได้ว่าเป็น One-way Hash function ซึ่งอัลกอริซึมของแฮช ที่ใช้เช่น MD5 จะสร้างค่า hash ขนาด 160 bit เป็นเอาท์พุทของฟังก์ชั่นแฮช(หรือเรียกอีกอย่างได้ว่าเป็น message digetest) และ มีขนาดของอินพุทเป็น 521 bit block

สำหรับ hash ที่ใช้นั้นสามารถทำให้ข้อมูลที่ถูก hash มีความบูรณภาพได้ นั่นคือ ข้อมูลที่ถูก hash จะสามารถตรวจ check ได้ว่า ข้อมูลนั้นถูกเปลี่ยนแปลง หรือถูกเพิ่มเติมอะไรบางอย่างหรือไม่ โดยการนำค่าแฮชมาแล้วมา ตรวจสอบกับข้อมูล หากข้อมูลมีการเปลี่ยนแปลง ค่าแฮชที่ได้จากข้อมูลกับค่าแฮชเดิมจะไม่ตรงกัน จึงสามารถตรวจสอบความบูรณภาพของข้อมูลได้

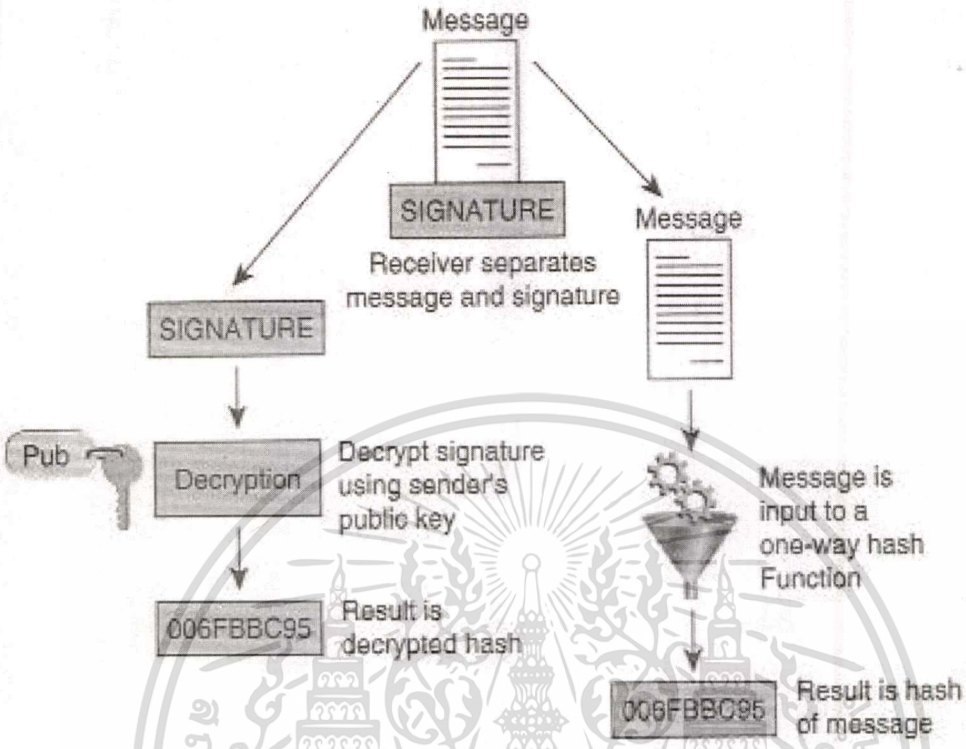
ดังนั้นในการสร้างลายมือชื่อดิจิทัล เป็นการรวม One-way hash function กับเทคโนโลยีการเข้ารหัสแบบกุญแจสาธารณะไว้ ซึ่งมีความสามารถในการพิสูจน์ตัวจริงรวมไปถึงการไม่ปฏิเสธรับผิดของผู้ส่งอีกด้วย วิธีการคือ ผู้ส่งสามารถใช้กุญแจลับ ที่ถูกสร้างขึ้นมาพร้อมกันกับกุญแจสาธารณะ ซึ่งกุญแจลับและกุญแจสาธารณะมีความสัมพันธ์กัน คือ หากข้อความถูกเข้ารหัสด้วยกุญแจสาธารณะ การที่จะถอดรหัสได้ มีเพียงวิธีเดียวคือ ถอดรหัสด้วยกุญแจลับที่มีความสัมพันธ์กับกุญแจสาธารณะที่ทำการเข้ารหัสข้อความมา ในการใช้กุญแจลับของผู้ส่งข้อความเข้ารหัสข้อมูลเพื่อส่งไปยังผู้รับ ดังนั้นผู้รับสามารถถอดรหัสข้อความได้ด้วยกุญแจสาธารณะของผู้ส่งได้เท่านั้น ในที่นี้จะใช้อัลกอริซึมของ RSA ในการเข้ารหัส ซึ่งขั้นตอนในการสร้างลายมือชื่อดิจิทัล มีดังรูป

จากรูป 3.2 จะเห็นว่า มีการนำ one-way hash function เข้ามาทำให้อินพุทที่เข้าไป กลายเป็น message digest เพื่อถูกใช้ในการเข้ารหัสโดยกุญแจลับของผู้ส่งให้กลายเป็นลายมือชื่อดิจิทัล หรือ ค่าแฮชที่ถูกเข้ารหัสนั่นเอง



รูป 3.2 การสร้างลายมือชื่อดิจิทัล

เมื่อลายมือชื่อดิจิทัลถูกส่งมายังผู้รับ ผู้รับทำการตรวจสอบ(Verify)ลายมือชื่อดิจิทัล ดังรูป ที่ 3.3 การตรวจสอบลายมือชื่อดิจิทัลจากผู้ส่ง โดยจะเห็นว่ามีการใช้กุญแจสาธารณะของผู้ส่งที่มีความสัมพันธ์กับกุญแจลับที่ใช้เข้ารหัสค่าแฮช ดังนั้นก่อนหน้านี้ผู้ส่งกับผู้รับ ต้องมีการติดต่อกันก่อนเพื่อนำเอากุญแจสาธารณะของผู้ส่งมาเก็บไว้ทางฝั่งผู้รับ หรือใช้ third-party เข้ามาช่วยในการแจกจ่ายกุญแจสาธารณะให้แก่กัน เพราะมั่นใจว่า กุญแจสาธารณะที่ใช้ถอดรหัส



รูป 3.3 การตรวจสอบลายมือชื่อดิจิทัล

จากรูปที่ 3.3 แสดงถึงการตรวจสอบลายมือชื่อดิจิทัลทางฝั่งผู้รับ มีขั้นตอนดังนี้

1. ผู้รับแยกข้อความออกเป็นเอกสาร(pliantext) และบล็อกลายมือชื่อ(Signature Block) ซึ่งบล็อกลายมือชื่อเป็นค่าแฮชที่ถูกเข้ารหัสด้วยกุญแจลับของผู้ส่ง
2. ผู้รับใช้กุญแจสาธารณะในการถอดรหัสบล็อกลายมือชื่อ ซึ่งผลที่ได้เป็น original message digest
3. ผู้รับนำเอกสาร(plaintext) เข้าในฟังก์ชั่นแฮช สิ่งที่ได้คือ message digest
4. แล้วนำ message digest ทั้งสองมาเปรียบเทียบกัน หากทั้งสองมีค่าที่เหมือนกัน แสดงว่าข้อความที่ส่งมาไม่ได้ถูกเปลี่ยนแปลง และมั่นใจได้ว่าเป็นข้อความจากผู้ส่งตัวจริง

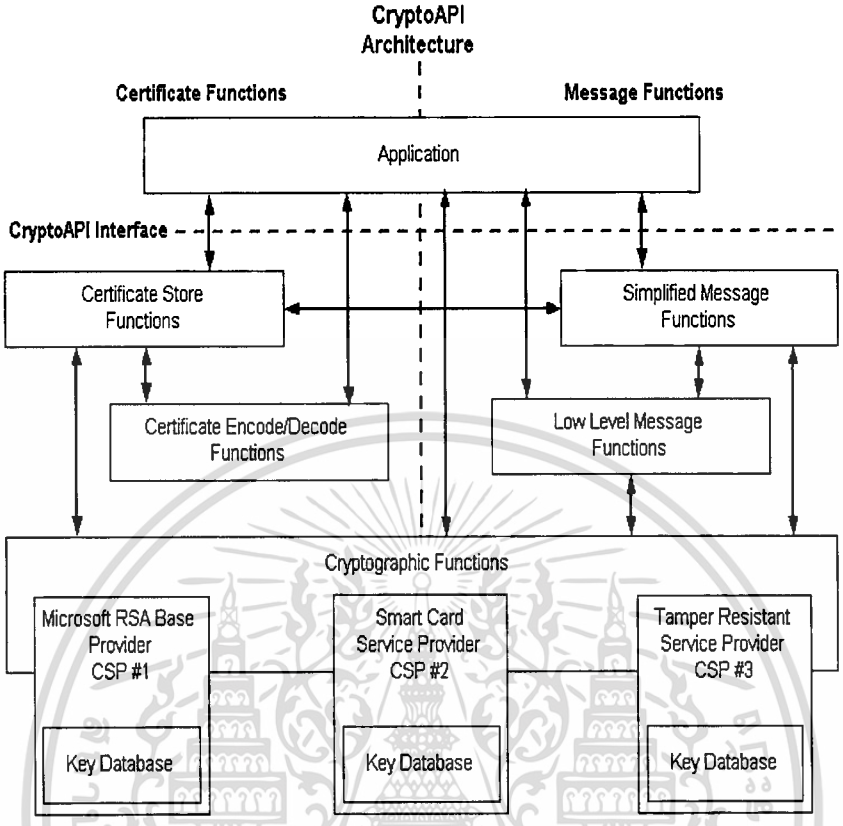
3.3 CryptoAPI

(Microsoft Corporation.2002) CryptoAPI เป็นการรวมเอาฟังก์ชั่นต่าง ๆ ซึ่งเป็นส่วนของระบบปฏิบัติการบนวินโดว์ หรืออาจกล่าวได้ว่า CryptoAPI เป็นการรวมอัลกอริซึมที่ต่างกันเอาไว้และรวมเอาฟังก์ชั่นไว้ในที่เดียวกัน โดย CryptoAPI ถูกออกแบบมาเพื่อนักพัฒนาโปรแกรม เพื่อใช้

ในการสร้างความปลอดภัยให้กับข้อมูลที่ต้องการให้เป็นความลับ ซึ่งในการใช้ CryptoAPI นี้มีพื้นฐานของวิทยาการเข้ารหัสลับเป็นหลัก สำหรับการพิสูจน์ตัวจริง เพื่อให้ผู้รับข้อมูลที่ส่งมามั่นใจว่าข้อมูลที่ได้รับเป็นข้อมูลจากผู้ส่งตัวจริงและข้อมูลไม่ได้ถูกดัดแปลงหรือมีอะไรแปลกปลอมปนเข้ามา

CryptoAPI มีความสามารถในการในการลงนามข้อมูลเพื่อการพิสูจน์ตัวจริง และมีการทวนสอบลายมือชื่อ รวมไปถึงการที่ CryptoAPI มีความสามารถรองรับความบูรณภาพข้อมูล ซึ่งเป็นผลกระทบจากการลงนามแบบดิจิทัล ทำให้เกิดลายมือชื่อดิจิทัลขึ้น นั่นคือ ผู้รับสามารถทวนสอบได้ว่าข้อมูลที่ได้รับมาไม่ได้ถูกเปลี่ยนแปลงตั้งแต่ถูกลงนาม ซึ่งความบูรณภาพของข้อมูล CryptoAPI มีการใช้ลายมือชื่อดิจิทัลและฟังก์ชั่นแฮชเพื่อรองรับการลงนามและตรวจสอบข้อมูล สำหรับส่วนประกอบของ CryptoAPI มีอยู่ 5 ส่วน ดังรูปที่ 3.4 เป็นรูปสถาปัตยกรรม CryptoAPI

1. Base Cryptography function เป็นส่วนการทำงานที่สำคัญใน CryptoAPIสำหรับใช้ในการติดต่อกับ CSP ซึ่งการทำงานนี้ ทำให้แอพพริชั่นสามารถเลือก CSP ตามที่ต้องการได้ สำหรับการทำงานของฟังก์ชั่นนี้มีความเกี่ยวข้องกับการสร้างและเก็บกุญแจ เพราะเป็นการทำงานที่ใช้ติดต่อกับ CSP(Cryptography Service Provider) ซึ่ง CSP เป็นสิ่งที่ช่วยในการสร้างและเก็บกุญแจคู่
2. Ceritificate encode/decode function เป็นการทำงานที่ใช้ในการเข้ารหัสและถอดรหัสข้อมูล รวมไปถึงการทำข้อมูลให้เป็น message digest
3. Simplified message function ในส่วนฟังก์ชั่นนี้ เป็นการเข้ารหัสและถอดรหัสข้อมูล รวมไปถึง การลงนามทั้งข้อความและเอกสาร หรือข้อมูลต่าง ๆ ที่ต้องการเข้ารหัส ซึ่งการทำงานนี้ สามารถตรวจสอบการพิสูจน์ตัวจริงของลายมือชื่อบนข้อความที่ได้รับและข้อมูลต่าง ๆ ที่เกี่ยวข้อง
4. Certificate store function เป็นการทำงานที่เกี่ยวกับการเก็บใบรับรองที่ใช้ประกอบการพิจารณากุญแจสาธารณะ
5. Low-level function สำหรับฟังก์ชั่นนี้ มีการทำงานเหมือนกันกับ Simplified message function ทุกอย่าง ซึ่ง low-level message function จะมีความสามารถที่ยืดหยุ่นกว่า Simplified message function แต่มีการใช้ฟังก์ชั่นมากกว่า ซึ่งจะส่งผลให้เกิดความยุ่งยากมากกว่า

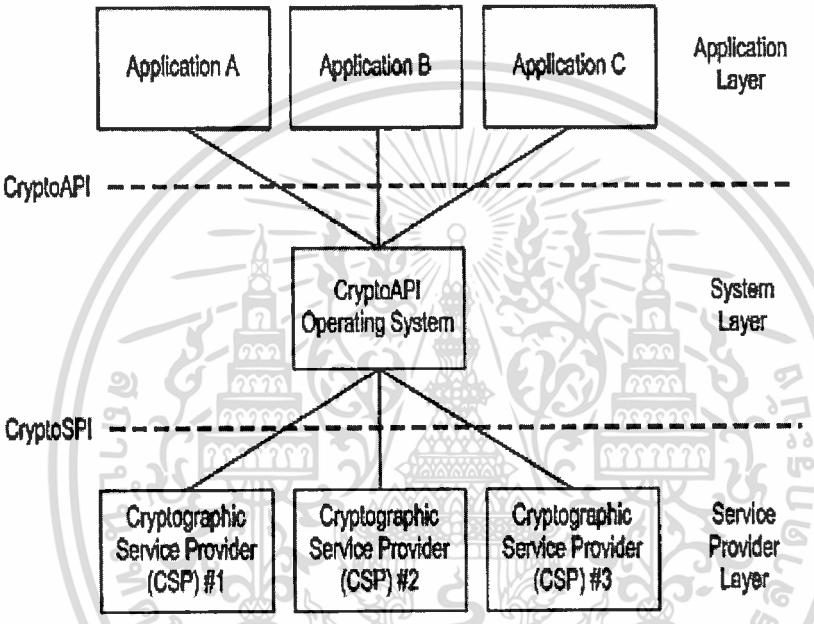


รูปที่ 3.4 สถาปัตยกรรม CryptoAPI

จากรูป Base Cryptography Function เป็นส่วนการทำงานที่สำคัญ มีการทำงานหลัก ๆ ที่ใช้สำหรับการพิสูจน์ตัวจริงดังนี้

1. Provider Function เป็นฟังก์ชั่นเพื่อใช้จัดการข้อมูลต่าง ๆ ที่เกี่ยวกับ Provider ที่ติดตั้ง และ Key container เช่น การเข้าครอบครอง Key container เมื่อต้องการใช้ container นั้นๆ , การปล่อย Key container เมื่อเลิกใช้งาน container นั้นๆ
2. Key Function เป็นฟังก์ชั่นเพื่อการสร้าง และทำลายกุญแจ ทั้งกุญแจแบบสมมาตร และอสมมาตร รวมไปถึงการ Import/Export key และการจัดการเรื่องคุณสมบัติต่าง ๆ ของกุญแจ
3. Hash and Digital signature Function เป็นฟังก์ชั่นสำหรับการลงนามกุญแจหรือข้อมูลให้กลายเป็นลายมือชื่อดิจิทัล
4. Encrypt/Decrypt Function เป็นฟังก์ชั่นใช้สำหรับเข้ารหัสกุญแจ หรือ ข้อมูลต่าง ๆ

เมื่อ CryptoAPI ใช้ CSP ในการเข้าและถอดรหัสข้อมูลและจัดสรรที่เก็บของกุญแจ ซึ่ง CSP เป็นขั้นตอนที่เป็นอิสระต่อส่วนของแอพพลิเคชั่น ดังนั้นแอพพลิเคชั่นสามารถทำงานบน CSP ได้หลากหลายแบบ โดยลักษณะการใช้งาน CSP คือCryptoAPI ใช้ Base Cryptography Function เพื่อติดต่อกับ CSP(Cryptography Service Provider) ต่าง ๆ ได้ ซึ่งการทำงานนี้ ทำให้แอพพลิเคชั่นสามารถเลือก CSP ตามที่ต้องการได้ โดยการทำงานทุก ๆ การติดต่อของแอพพลิเคชั่นกับ CSP จะต้องผ่านการทำงานของ Base Cryptography Function ก่อน ดังรูป 3.5



รูปที่ 3.5 สถาปัตยกรรม CSP

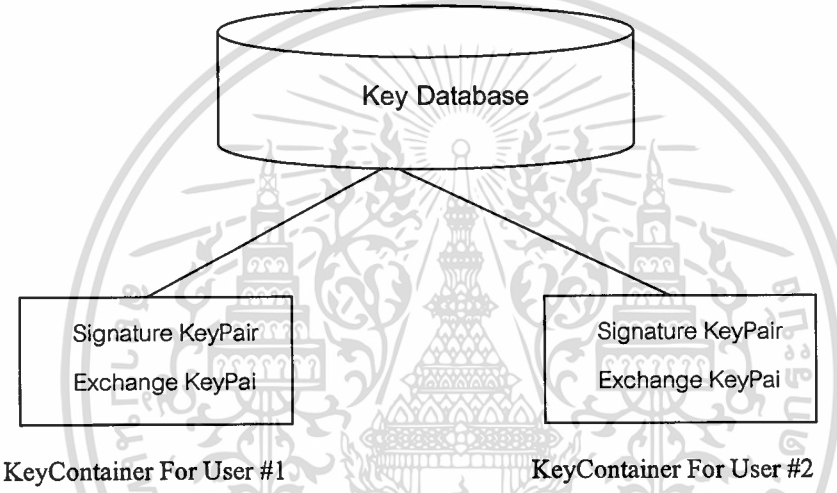
เราอาจสรุปได้ว่า CryptoAPI เป็นชุดของฟังก์ชั่นทางด้านวิทยาการเข้ารหัสลับ ที่ใช้ฟังก์ชั่นเดียวกันสามารถใช้อัลกอริซึมที่ต่างกันได้ เช่น RC2,DES สามารถใช้ฟังก์ชั่น CrypGenKey ในการสร้างกุญแจได้ สำหรับ CSP เปรียบเสมือน ตัวเก็บอัลกอริซึมของวิทยาการเข้ารหัสลับเอาไว้เป็นชุด หากต้องการอัลกอริซึมแบบใด ก็สามารถเลือกใช้ CSP ที่แตกต่างกันได้ โดยใช้ฟังก์ชั่นจาก CryptoAPI เรียกมาที่ CSP ได้

สำหรับแต่ละ provider มีมาตรฐานสำคัญที่เหมือนกันดังนี้

1. แต่ละ provider มีเพียงอัลกอริซึมแบบอสมมาตรชนิดเดียวสำหรับการเข้ารหัสกุญแจ เรียกได้ว่า key exchange algorithm

2. แต่ละ provider มีเพียงอัลกอริซึมแบบอสมมาตรชนิดเดียวสำหรับลายมือชื่อดิจิทัล ซึ่งเรียกได้ว่า Digital signature algorithm
3. แต่ละ provider มีรูปแบบของไฟล์ที่เป็นรูปแบบเดียว สำหรับเป็นรูปแบบที่ใช้ในการส่งหรือนำเข้ากุญแจ ซึ่งในที่นี้ ใช้รูปแบบ BLOB(binary Large Object)

ในแต่ละ CSP มี Key Database เพื่อการเข้ารหัสลับของกุญแจต่าง ๆ ที่สร้างขึ้น ซึ่งใน Key Database จะมีกุญแจคู่อยู่สองคู่ โดยข้อมูลของกุญแจถูกเข้ารหัสและถูกเก็บไว้ให้อยู่ในรูปแบบที่ปลอดภัยใน Key Database ซึ่ง CryptoAPI มีฟังก์ชั่นที่เรียกว่าk CryptAcquireContext สำหรับใช้ในการควบคุม Key container ดังรูปที่ 3.6 แสดงถึงลักษณะของ Key database



รูป 3.6 Key Database

จากรูป 3.6 สำหรับผู้ใช้หนึ่งคน จะสามารถมี Key container ของตนเองได้เพียง Key container เดียว ภายในแต่ละ Key container มีกุญแจคู่อยู่สองชนิดด้วยกัน คือ Exchange Key มีไว้สำหรับเข้ารหัส Session key และ กุญแจคู่ที่สอง คือ Signature key ไว้สำหรับลงนามข้อมูล

3.4 Microsoft Visual Basic

การพัฒนาโปรแกรมบนวินโดว์ในปัจจุบัน กระทำได้ง่าย และสะดวกขึ้น เนื่องจากมีการใช้เทคโนโลยีทางด้าน Visualize เข้ามาประกอบในการออกแบบจอภาพ ซึ่งต่างจากในอดีต ที่การพัฒนาโปรแกรมบนวินโดว์นั้นค่อนข้างจะทำได้ยาก เนื่องจากการพัฒนาโปรแกรมหนึ่ง ๆ ให้แล้วเสร็จ

โปรแกรมเมอร์ต้องเขียน Routine ต่างๆ ขึ้นเป็นจำนวนมาก ซึ่ง Visual Basic จัดเป็นภาษาหนึ่งที่ได้รับความนิยม และถูกนำมาใช้ในการพัฒนาโปรแกรมเพื่อใช้งานบนวินโดว์

Visual Basic เป็นภาษาคอมพิวเตอร์ที่ได้รับความนิยมนำมาใช้ในการพัฒนาโปรแกรมบนวินโดว์ เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแต่เลือก Control ที่เหมาะสม แล้ววาดลงบน Form ก็สามารถสร้างจอภาพที่ใช้งานสำหรับติดต่อกับผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับ Control ต่าง ๆ ที่สร้างขึ้นตามเหตุการณ์ (Event) ต่าง ๆ ที่เกิดขึ้น สำหรับการสร้างระบบการพิสูจน์ตัวจริงด้วย Visual Basic จะใช้ WCCO object ซึ่งเป็น Reference ที่ใช้สำหรับ CryptoAPI ซึ่งปรกติแล้ว CryptoAPI ถูกเขียนขึ้นด้วยภาษา C หรือ C++

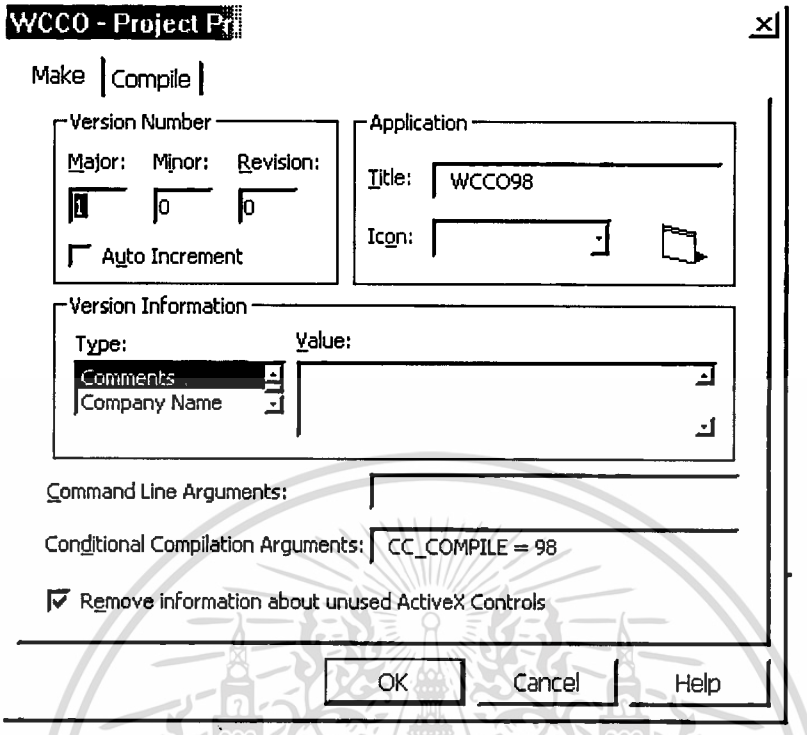
ในการสร้างระบบพิสูจน์ตัวจริงให้กับ SQL Server ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ในระดับเซิร์ฟเวอร์ การเขียนโปรแกรมต้องมีการติดต่อกับฐานข้อมูล SQL Server ด้วย ดังนั้น จึงต้องมี ODBC หรือ OLEDD เพื่อเป็นตัวกลางในการติดต่อกับฐานข้อมูลทางฝั่งเซิร์ฟเวอร์

3.5 WCCO Object

(Bondi. 2002)Wiley CryptoAPI COM Objects หรือ WCCO เป็น COM ที่ใช้ใน Visual Basic 6.0 สำหรับ Base Cryptography Function ของ CryptoAPI ซึ่ง WCCO source code จะต้องถูกคอมไพล์ในรูปแบบที่แตกต่างกันไปของระบบปฏิบัติการวินโดว์ที่ต่างแพลทฟอร์ม เช่น Windos98, WindowsNT รวมไปถึง Windows2000

สำหรับการคอมไพล์โค๊ดของ WCCO จะต้องตั้งค่าของ CC_COMPILE ซึ่งในการตั้งค่าในการคอมไพล์นี้ ขึ้นอยู่กับวินโดว์ว่าเป็นเวอร์ชั่นอะไร ซึ่งจะตั้งค่าของ CC_COMPILE ดังนี้

Windows95	ใช้	CC_COMPILE = 97
Windows98	ใช้	CC_COMPILE = 98
WindowsNT	ใช้	CC_COMPILE = 100
Windows2000	ใช้	CC_COMPILE = 104



รูปที่ 3.7 แสดงการคอมไพล์ WCCO โดยใช้ Visual Basic

บน Project Properties Dialog Box ผลที่ได้จากการคอมไพล์ จะได้ WCCO98.DLL เพื่อเป็น Reference ไว้ใน Visual Basic และเมื่อนำ WCCO มาใช้ จะต้องมีการรีจีสเตอร์ DLL เสียก่อน โดยใช้คำสั่ง "REGSVR32 WCCOxxx.DLL" และการใช้งานบน Windows95,98 จะต้องรีกีสเตอร์ด้วยคำสั่ง "C:\WINDOWS\SYSTEM\REGSVR WCCOxxx.DLL" ในการใช้ WCCO object เพื่อให้เกิดการเข้ารหัสที่แข็งแรง จึงมีการนำเอา CryptoAPI's Microsoft Enhanced Provider เข้ามาใช้ ซึ่งเป็น 128-bit Encryption สำหรับส่วนต่าง ๆ ของ WCCO object ที่ถูกใช้ในการสร้างระบบพิสูจน์ตัวจริง จะถูกอธิบายในตอนต่อไป

สำหรับขั้นตอนที่สำคัญของ WCCO Object มีหลายขั้นตอนดังนี้

3.5.1. ขั้นตอนการสร้างกุญแจคู่

สำหรับ WCCO Object ในการสร้างลายมือชื่อดิจิทัล จำเป็นต้องมีการสร้าง Signature Key ขึ้นมาก่อน ซึ่ง ขั้นตอนในการสร้างมีดังนี้

1. เข้าครอบครอง Context ที่ต้องการเสียก่อน นั่นคือ เข้าครอบครอง Provider ที่ต้องการ
2. กำหนด Container ที่จะเก็บกุญแจเสียก่อนว่าชื่ออะไร

3. ทำลายกุญแจต่าง ๆ หรือปลดปล่อยกุญแจต่าง ๆ ที่ได้ครอบครองไว้ในตอนแรกก่อนทำการสร้างกุญแจ
4. กำหนดค่า dwflags ให้ ซึ่ง High Byte ของมันจะเป็นค่าของขนาดกุญแจที่จะสร้าง
5. ในการสร้างกุญแจคู่ จะส่งค่าตัวแปรต่าง ๆ ดังนี้
   - m_oPrivider.Contexthandle นั่นคือ Context นี้ใช้ Provider อะไรซึ่งในที่นี้จะใช้ RSA_FULL
   - m_nExOrSigkey จะบ่งบอกว่ากุญแจที่ถูกสร้างขึ้นนี้เป็นประเภทของ Exchange Key หรือ Signature Key
   - dwflags ตัวแปรนี้บ่งบอกถึงค่า dwflags ซึ่งค่า dwflags นี้จะเชื่อมโยงมาจากการกำหนดความยาวของกุญแจว่าให้มีขนาดเท่าไหร่

เมื่อมีการสร้างกุญแจเก็บเอาไว้แล้ว ก็จะถูกนำเข้าใน Container ที่เปิดเอาไว้ในตอนสร้างกุญแจนั้น ๆ และกุญแจจะถูกแยกแยะด้วยว่าเป็นประเภทของ Exchange key หรือ Signature Key ในการเก็บกุญแจเอาไว้ใน Container จะถูกเข้ารหัสโดยตัวของ Windows ซึ่งจะกล่าวในภายหลังว่าถูกเข้ารหัสได้อย่างไร ดังนั้นจึงมั่นใจได้ว่า จะไม่มีการเรียกใช้กุญแจผิดประเภท หรือกุญแจถูกนำมาใช้โดยบุคคลอื่นที่ไม่ใช่เจ้าของ

3.5.1. ขั้นตอนการลงนามข้อมูล

ในการลงนามข้อมูบของ WCCO Object นี้ ก่อนจะเรียกใช้ขั้นตอนการลงนามจะต้องเริ่มโหลดกุญแจลายมือชื่อ เข้าไปใน CRSAKeyPair เสียก่อน ซึ่งประเภทของกุญแจลายมือชื่อนี้ จะเป็นประเภทกุญแจลับ(Private Key) ซึ่งจะถูกใช้ในการลงนามข้อมูล ซึ่งอัลกอริซึมของ Hash Object มีค่า default เป็น “MD5”

ขั้นตอนของการลงนามข้อมูลมีดังนี้

1. การตรวจสอบ Object นี้มีกุญแจลับเพื่อการลงนามหรือไม่ และตรวจสอบดูว่าอัลกอริซึมที่ใช้ Hash ที่กำหนดไว้ถูกต้องหรือไม่ (ในที่นี้ใช้ MD5 สำหรับอัลกอริซึมของ Hash)
2. นำข้อมูลที่ต้องการลงนาม ซึ่งข้อมูลอาจเป็นกุญแจ หรือข้อมูลต่าง ๆ แต่ไม่ควรมีขนาดที่ใหญ่นัก เพราะจะทำให้เกิดความล่าช้าในขั้นตอนต่าง ๆ ซึ่งข้อมูลนี้จะถูกเก็บให้อยู่ใน Cmessagetext เพื่อรอการลงนาม แต่ต้องคำนึงถึงรูปแบบของข้อมูลด้วยว่าอยู่ในรูปแบบที่พร้อมหรือไม่ ซึ่งในการโหลดข้อมูลเข้า

Cmessagetext นั้น จะเป็นการ convert ข้อมูลให้อยู่ในรูปแบบที่พร้อมโดยอัตโนมัติ

3. ทำการ Hash ข้อมูลที่ได้จากขั้นตอนที่ 2. ซึ่งการทำในขั้นตอนนี้จะต้อง Clear Hash Object ต่าง ๆ ที่เคยเข้าครอบครองไว้ก่อน แล้วค่อยมาทำการ Hash ซึ่งผลที่ได้ จะเป็น hash value ซึ่งค่าที่ได้นี้ จะถูกครอบครองเอาไว้เสียก่อน
4. ในขั้นตอนนี้ จะทำการลงนาม hash value ที่ได้จากขั้นตอนที่ 3. ด้วย กุญแจลับที่อยู่ใน container ที่กำลังเปิดอยู่ ซึ่งมีการตรวจสอบก่อนว่าเป็น Exchange Key หรือ Signature Key ถ้า CRSAKeyPair เป็น Signature Key Object ก็จะใช้ Private Signature Key ในการลงนาม
5. ผลที่ได้เป็น Signature Block ซึงการได้ Signature Block อาจสามารถเปลี่ยนแปลงให้อยู่ในรูปแบบของไฟล์แบบเลขฐานสิบหกก็ได้ แล้วแต่ต้องการ
6. ทำลาย hash object ที่สร้างขึ้นจากขั้นตอนที่ 3 เพื่อไม่ให้ผู้อื่นนำไปใช้ประโยชน์ได้

จะเห็นได้ว่า ผลที่ได้เป็นเพียง Signature Block และเมื่อกลับไปดูทฤษฎีของลายมือชื่อดิจิทัลแล้ว จะเห็นว่าเป็นการนำ Signature Block เข้ารวมกับข้อมูลก่อนการลงนามมารวมกันนั่นเอง จึงได้เป็นลายมือชื่อดิจิทัลขึ้นมา

3.5.3. ขั้นตอนตรวจสอบลายมือชื่อดิจิทัล

เมื่อสร้างลายมือชื่อดิจิทัลเสร็จแล้วจะต้องมีขั้นตอนการตรวจสอบลายมือชื่อดิจิทัลเพื่อการพิสูจน์ตัวจริงของผู้ส่ง ก่อนที่ขั้นตอนการตรวจสอบลายมือชื่อดิจิทัลจะเกิดขึ้นจะต้องมีการโหลด Public Key Signature ก่อนที่จะใช้ขั้นตอนการตรวจสอบลายมือชื่อดิจิทัล ซึ่งมีขั้นตอนดังนี้

1. ตรวจสอบว่า Object ที่ได้ครอบครองกุญแจสาธารณะที่ถูกต้องหรือไม่ ซึ่งการโหลดกุญแจสาธารณะก่อนหน้านี้ จะต้องโหลดกุญแจสาธารณะให้อยู่ในรูปแบบที่ใช้ได้ นั่นคือ convert กุญแจสาธารณะที่อยู่ในรูปแบบอื่น ไม่ว่าจะเป็นไฟล์ หรือเป็นสตริง ให้อยู่ในรูปแบบของ Byte Array เสียก่อน เพราะก่อนหน้านี้ กุญแจสาธารณะอาจถูกเก็บอยู่ในรูปแบบอื่นได้
2. ตรวจสอบ SigBlock ที่ส่งเข้ามาในขั้นตอนนี้ด้วย ว่ามีการเปลี่ยนให้อยู่ในรูปแบบที่ใช้งานได้หรือไม่ ซึ่งการทำงาน จะเป็นลักษณะเดียวกันกับการโหลดกุญแจสาธารณะนั่นเอง

3. ในขั้นตอนนี้จะทำการแฮชข้อมูลที่ถูกส่งเข้ามาใน Object ของการตรวจสอบลายมือชื่อ นั่นคือ จะต้องส่งข้อมูลก่อนการลงนามเข้ามาทำแฮชเพื่อตรวจสอบ ในขั้นตอนนี้อันดับแรก จะต้องเคลียร์ hash Object ก่อน ไม่ว่าจะมีการเข้าครอบครองอะไรไว้ ให้ปล่อยไปทุกอย่าง แล้วจึงตั้งค่าอัลกอริซึมของแฮชที่ใช้ตรงกับการลงนาม(ในที่นี้ใช้ MD5) โดยที่ เมื่อได้ค่าแฮช(hash value) แล้วจะต้องเข้าครอบครองค่าแฮชเอาไว้
4. การทวนสอบลายมือชื่อดิจิทัล ด้วยการใช้ Hash Object Handle จากขั้นตอนที่ 3. และ Signature Block ที่อยู่ในรูปแบบของ Byte Array จากขั้นตอนที่ 2. และกุญแจสาธารณะที่เป็นของ CRSAKeyPair Object ที่ได้เข้าครอบครองไว้ตอนโหลดกุญแจสาธารณะในตอนต้น นำค่าที่ได้ทั้งหมดเข้าทำการตรวจสอบลายมือชื่อดิจิทัล หากผลลัพท์ที่ได้เป็น “True” แสดงลายมือชื่อดิจิทัลที่ได้เป็นของจริง แต่ถ้าได้ค่า “False” แสดงว่าลายมือชื่อนี้ไม่ใช่ของจริง
5. ในขั้นตอนสุดท้ายของการตรวจสอบลายมือชื่อ จะทำลาย Object Hash ที่ได้เข้าครอบครองเอาไว้ เพื่อไม่ให้ผู้อื่นสามารถนำไปใช้ได้

3.6 Key Container

ในการสร้างกุญแจคู่เพื่อเข้ารหัสลับข้อความ ดังนั้น กุญแจคู่ที่ถูกสร้างขึ้นจำเป็นต้องคำนึงถึงเรื่องที่เก็บกุญแจไว้ให้ปลอดภัยต่อการโจรกรรม สำหรับกุญแจที่ถูกสร้างขึ้นจะถูกเชื่อมต่อกับ container ซึ่งเป็นที่บรรจุกุญแจเอาไว้ สำหรับหนึ่ง container จะเก็บกุญแจได้เพียงคู่เดียวในแต่ละประเภทของกุญแจ และไม่เก็บ session key หรือกุญแจที่ไม่สมบูรณ์ (มีเพียงกุญแจสาธารณะอย่างเดียว) แต่การที่ใช้กุญแจคู่ได้นั้น ไม่เพียงแต่เก็บเอาไว้ใน container แล้วนำมากุญแจคู่มาใช้เข้ารหัสหรือทำอย่างอื่นได้เลย แต่ต้องมีวิธีการสำหรับนำกุญแจมาใช้ อยู่ 3 วิธี คือ การสร้างกุญแจคู่ขึ้นมาใหม่ วิธีที่สอง ใช้ฟังก์ชั่น เพื่อให้ CryptoAPI รู้จัก container ที่มีกุญแจคู่ที่ต้องการใช้เสียก่อน ส่วนวิธีสุดท้าย ใช้ฟังก์ชั่นเพื่อการส่งกุญแจมายัง CryptoAPI ให้อยู่ในรูปแบบของ BLOB เสียก่อน (ซึ่งก่อนหน้านี้ ถูกเก็บไว้ในรูปแบบไฟล์)

วิธีการตั้งค่า Key Container ซึ่งการตั้งค่ามีวิธีการตั้งค่าอยู่ 2 วิธี คือ

1. ตั้งค่าให้เป็น CRYPT_MACHINE_KEYSET หากมีการตั้งค่า container ให้เป็น CRYPT_MACHINE_KEYSET ใน Registry จะเป็น HKEY_LOCAL_MACHINE การที่ container ถูกตั้งค่าให้เป็นแบบนี้ทำให้ไม่ว่าใครก็ตามที่ล็อกอินเข้ามาในเครื่องที่มี container อยู่ จะสามารถมองเห็น container ทั้งหมดได้ทุกคน

2. ถ้าไม่ตั้งค่า container ให้เป็น CRYPT_MACHINE_KEYSET ใน Registry จะเป็น HKEY_CURRENT_USER การที่ container ถูกตั้งค่าให้เป็นอย่างนี้ทำให้เมื่อ user มีการล็อกอินเข้าเครื่อง สามารถมองเห็นเพียง container ที่ถูกสร้างขึ้นภายใต้ล็อกอินของ user นี้เท่านั้น ไม่สามารถมองเห็น container ที่ถูกสร้างโดยล็อกอินอื่นได้

ซึ่ง WCCO objectsใช้ CONTAINER_ACCESS.SINGER_USER แทนการใช้ CRYPT_MACHINE_KEYSET และสำหรับการทวนสอบลายมือชื่อโดยใช้กุญแจสาธารณะ จะใช้ CONTAINER_KEYS_ACCESS.PUBLIC_KEYS_ONLY ในการทวนสอบ แต่สำหรับการใช้ CONTAINER_KEYS_ACCESS.PUBLIC_KEYS_ONLY ต้องกำหนดตั้งชื่อ container ให้เป็น " " หรือ VbnullChar ซึ่งเป็นค่า default ของ container สำหรับการจัดเก็บกุญแจ ซึ่ง WCCO มีความสามารถในการเก็บกุญแจให้อยู่ในรูปแบบได้หลายรูปแบบ ซึ่ง CryptoAPI ใช้รูปแบบของ BLOB(Binary Large object) ซึ่งเป็นรูปแบบของการเก็บกุญแจไว้ใน container ในการเก็บกุญแจ

ในการที่จะใช้กุญแจคู่จาก container จะใช้การ import key เพื่อให้เข้าสู่ CryptoAPI ในการ import key pair เป็นการ import แค่กุญแจลับ ซึ่งในการ import key เพื่อให้เข้าสู่ CryptoAPI เป็นการให้ CryptoAPI รู้ว่ามันเป็นกุญแจประเภทอะไรและของใคร ซึ่งในการให้มันรู้จักกุญแจโดยการใช้ CryptGetUserkey ในการให้ CryptoAPI รู้จักกุญแจที่มีอยู่ใน container ที่ถูกเปิดอยู่

การสร้างกุญแจขึ้นมาแต่ละคู่ เพียงใช้คำสั่ง "CrypGenKey" ของ WCCO objects ก็จะสร้างกุญแจคู่ขึ้นมาหนึ่งคู่(อาจเป็น Exchange key หรือ Signature key) การสร้างกุญแจคู่ขึ้นมายังไม่สามารถบอกได้ว่ากุญแจคู่นี้ ดอกไหนเป็นกุญแจลับ หรือดอกไหนเป็นกุญแจสาธารณะ จึงจำเป็นต้องมีการกำหนดว่ากุญแจใดเป็นกุญแจสาธารณะ โดยใช้ "CryptExportKey" ในการส่งกุญแจออกแล้วใช้ "CryptImportKey" ส่งเข้ามาเพื่อกำหนดให้เป็นกุญแจสาธารณะอีกที

เมื่อผู้ใช้ต้องการกุญแจที่ถูกเก็บอยู่ใน container ดังนั้น เราต้องทำการครอบครอง container โดยการบอกชื่อ container นั้น ๆ ก่อน การกระทำอย่างนี้เป็นการใช้ฟังก์ชั่น CryptAcquireContext ซึ่งทาง Microsoft เรียกว่า "Provider handle" แต่ในที่นี้จะเรียกว่า "Context handle" ซึ่งฟังก์ชั่นของ CryptoAPI ต้องการเข้าครอบครองอย่างนี้ เพื่อใช้ในการทำงานต่าง ๆ เช่น

1. CryptGenkey เป็นการสร้างกุญแจ ซึ่งต้องการให้มีความสอดคล้องกับอัลกอริซึมของ Provider ที่เราต้องการ
2. CryptHashData การทำงานนี้เป็นการแฮชข้อมูล ซึ่งอาจไม่เกี่ยวข้องโดยตรง แต่มันต้องการสิ่งที่ handle จาก CryptCreateHash ซึ่งไม่สามารถเกิดขึ้นได้เลย ถ้าเราไม่เข้าครอบครอง Context ก่อน

อาจกล่าวสรุปได้ว่า การเข้าครอบครอง Contextโดยใช้ฟังก์ชั่น CryptoAcquireContext และถ้าเรามีการครอบครอง Context ที่ใช้ได้(valid) ทำให้สามารถครอบครอง Context ได้

3.7 การบริหารกุญแจ (Key Management)

วิทยาการเข้ารหัสลับถูกใช้เป็นเครื่องมือสำหรับการรักษาความปลอดภัยให้กับข้อมูล แต่ในส่วนของวิทยาการเข้ารหัสลับมีปัญหาเกี่ยวกับการป้องกันกุญแจ จากการ ขโมย ดัดแปลง หรือ การสูญหายของกุญแจ ดังนั้นต้องมีวิธีการแก้ไข คือในการเก็บกุญแจที่สร้างขึ้น ซึ่ง WCCO object มีการ Unload key ใด ๆ เข้าไปใน BLOB (เป็นรูปแบบในการเก็บกุญแจ เช่น PrivateKeyBLob เป็นที่เก็บกุญแจลับที่อยู่ในรูปแบบของ Byte Array) ซึ่ง BLOB จะถูกเข้ารหัสเอาไว้ ซึ่งวิธีการเข้ารหัส Key container จะไม่ถูกเปิดเผย โดยวิธีการเข้ารหัส container จะมีวิธีการดังนี้ คือ เมื่อมีการสร้างหรือ Import key pair เข้ามายัง CryptoAPI ผู้อื่นที่ล็อกอินเข้ามาในเครื่องสามารถเห็น Key container ที่ถูกสร้างขึ้นได้โดยดูที่ Registry ดังนั้น หากการสร้างกุญแจด้วย Flag ที่ไม่ใช่ CRYPTO_MACHINE_KEYSET container ก็จะถูกเข้ารหัสภายใต้วินโดว์

สำหรับวินโดว์ 95/98 หรือ NT มี Pstore (Protected Storeage Service) และวินโดว์2000 จะมี DAPI(Data Protection API) ซึ่ง CSP ต้องการให้วินโดว์ทำการเข้ารหัสบางอย่างให้กับ Key container และกุญแจคู่โดยเฉพาะในส่วนของ Private Key Blob ซึ่ง CSP จะยอมให้วินโดว์ใช้ DAPI/Pstore ในการเข้ารหัส BLOB นั่นคือ เมื่อผู้ใช้สร้าง Key container ไว้บนล็อกอินของตัวเอง และเมื่อผู้ใช้ทำการล็อกอินเข้าสู่วินโดว์ วินโดว์จะทำการส่งกุญแจที่ถูกเข้ารหัสเอาไว้ให้แก่ผู้ใช้คนนั้นเพียงผู้เดียว หากผู้อื่นล็อกอินเข้ามาด้วยรหัสอื่น ก็ไม่สามารถจะถอดรหัสกุญแจนี้ได้ ดังนั้นกุญแจลับหรือกุญแจสาธารณะที่ถูกเก็บไว้ในเครื่องจะปลอดภัยจากผู้อื่น

การ Load/Unload key เมื่อต้องการใช้กุญแจที่สร้างขึ้น ไม่ว่าจะเป็นการนำกุญแจมาเข้ารหัสหรือถอดรหัส จะรู้ได้อย่างไรว่ากุญแจถูกส่งเข้าสู่ key object ไม่ผิด เพราะฉะนั้นใช้วิธีดูที่ BlobHeader ซึ่งเป็นส่วนหัวของกุญแจที่ถูกแปลงให้อยู่ในลักษณะของ Byte Array โดย BlobHeader มีส่วนต่าง ๆ ดังนี้

1. bType เป็นการบอกว่าเป็นกุญแจประเภทอะไร
2. bVersion เป็นการบอกว่าKey Blobใช้ Version อะไร
3. Reversed จะถูกใช้ในอนาคต แต่ตอนนี้ถูกตั้งค่าให้เป็น 0
4. AiKeyAlg ส่วนนี้บอกถึงอัลกอริซึมที่ใช้ ว่าถูกใช้สำหรับกุญแจแลกเปลี่ยน(Exchange Key) หรือกุญแจลายมือชื่อ(Signature Key)

จากทั้งสี่ส่วน ทำให้ BlobHeader มีความยาว 8 Bytes ที่กล่าวมา 4 อย่างใน BlobHeader สามารถใช้ Class ของ CkeyTools เพื่อทำการตรวจสอบ Key Blob (กุญแจที่ถูกแปลงให้อยู่ในลักษณะของ Byte Array) ว่าเป็น Header ของกุญแจประเภทไหน

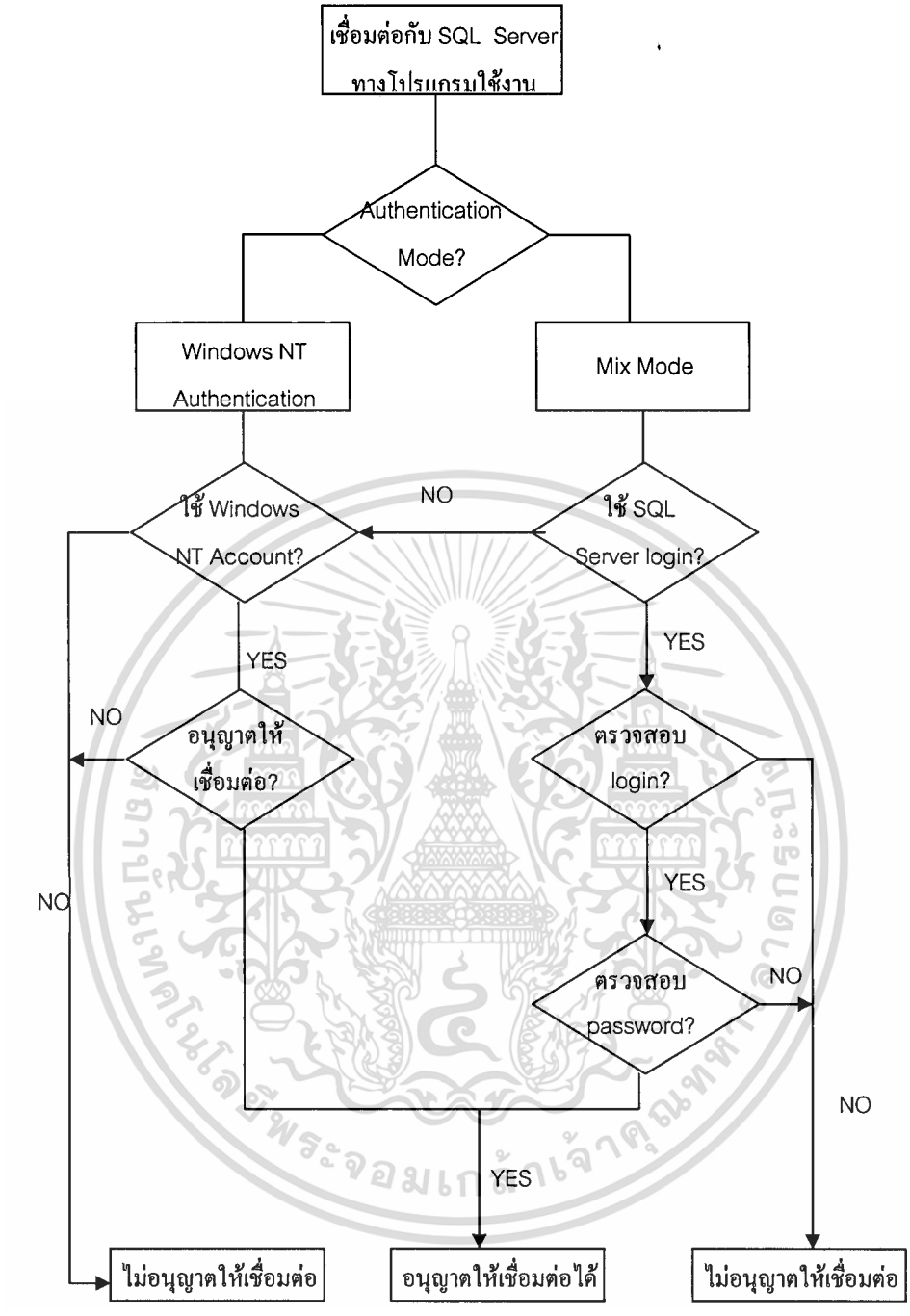
3.8 ฐานข้อมูล Microsoft SQL Server

(Microsoft Corporation. 2002)Microsoft SQL Server เป็นระบบจัดการดาต้าเบสแบบ Client/Server Relational Database ซึ่งทำงานได้ตั้งแต่เครื่องที่มีระบบปฏิบัติการวินโดว์ 95/98 ซึ่งจะใช้ได้เพียงผู้ใชคนเดียวเท่านั้น แต่ถ้าต้องการให้เป็นแบบ multi-user ต้องเลือกใช้เครื่องที่มีระบบปฏิบัติการWindows NT หรือ Windows2000 ซึ่ง Micrsoft SQL Server ได้สร้างระบบที่ควบคุมความปลอดภัยของข้อมูล ซึ่งสามารถตั้งค่าให้กับ SQL Server ได้ 2 แบบ ตาม รูป 3. 7 ความปลอดภัยของ Microsoft SQL Server ดังนี้

1. Windows Authentication Mode เป็นรูปแบบการรักษาความปลอดภัยให้กับ Microsoft SQL Server สำหรับ Mode นี้ ความปลอดภัยของ SQL Server จะอยู่บนความดูแลวินโดว์ ซึ่งผู้ใช้หรือกลุ่มของวินโดว์จะถูกอนุญาตให้สามารถเข้าใช้งาน SQL Server ได้อัตโนมัต ซึ่ง Mode นี้ ยอมให้ SQL Server อยู่บนการพิสูจน์ตัวจริงของวินโดว์ซึ่งเป็นวิธีเดียวกับแอพพลิเคชั่น ๆ สำหรับการติดต่อแบบนี้อาจเรียกได้ว่าเป็น “Trusted Connections” เมื่อเลือกใช้รูปแบบนี้ เป็นการกำหนดให้ SQL Server ยอมรับรหัสล็อกอินของผู้ใช้จากวินโดว์เท่านั้น โดยใช้ SID(Security Identifier) ในการล็อกอินเข้าวินโดว์ เมื่อใช้ SID จึงสามารถให้สิทธิ์ผู้ใช้หรือกลุ่มที่ล็อกอินบนวินโดว์เข้าใช้ SQL Server ได้โดยตรง
2. Mix Mode เป็นรูปแบบการรักษาความปลอดภัยให้กับ Microsoft SQL Server สำหรับใน Mode นี้ ความปลอดภัยของ SQL Server รับการล็อกอินจากผู้ใช้ทั้งแบบที่เป็น SQL Server Authentication และแบบ Windows Authentication ได้แล้วแต่จะกำหนด หากมีการเลือกให้อยู่ในรูปแบบของ Windows Authentication ลักษณะการรักษาความปลอดภัยต่าง ๆ ก็จะเหมือนกันกับข้อ 1. ที่กล่าวมาซึ่งอาจมีการนำ NTLM หรือ Kerberos เข้ามาใช้เพิ่มความปลอดภัยให้มากขึ้น แต่หากเลือกให้เป็น SQL Server Authentication จะทำให้ SQL Server ต้องการ Username/Password ที่เป็นรหัสล็อกอินเพื่อใช้ในการเปรียบเทียบกับรหัสล็อกอินที่ถูกเก็บไว้ในตาราง sysxlogin บนฐานข้อมูล master ใน SQL Server ซึ่งการติดต่อแบบใช้ Username/Password นี้อาจเรียกได้ว่าเป็น “Non-

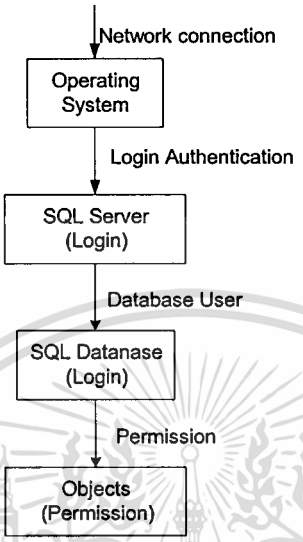
trusted connection" ซึ่งการติดตั้ง SQL Server บน windows 95/98/ME จะทำให้รูปแบบความปลอดภัยถูกเลือกให้เป็น Mix Mode

SQL Server มีส่วนที่ทำหน้าที่ติดต่อสื่อสารกับทางฝั่งไคลเอนต์อยู่ด้วย แต่ก็แยกส่วนที่จัดการเน็ตเวิร์กและโปรโตคอลออกจากส่วนที่เป็นแอพพลิเคชั่น ทำให้แอพพลิเคชั่นสามารถทำงานอยู่บนเน็ตเวิร์กแบบใดก็ได้ ซึ่งแอพพลิเคชั่นเป็นโปรแกรมที่ถูกพัฒนาขึ้นเพื่อใช้ข้อมูลจากฐานข้อมูลโดยผ่านอินเตอร์เฟสของโปรแกรมที่เรียกว่า API โดยมี Database Interface ซึ่งเป็นอินเตอร์เฟสที่ใช้โดยแอพพลิเคชั่นเพื่อติดต่อไปยัง SQL Server เช่น ODBC(Open Database Connectivity เป็น engine หนึ่งในการสร้างทางติดต่อระหว่างแอพพลิเคชั่นและฐานข้อมูล) หรือ OLE DB(Object Linking and Embedding Database)



รูป 3.8 ความปลอดภัยของ Microsoft SQL Server

ในการติดต่อเพื่อให้สามารถเข้าถึง และใช้งานระบบ SQL Server ได้นั้น มีระบบรักษาความปลอดภัยตลอดจนสิทธิ์การใช้งาน ซึ่งสามารถแสดงให้เห็นภาพโดยรวมเป็นลำดับขั้น ดังรูป



รูปที่ 3.9 รูปแบบในการเข้าถึงระบบของ SQL Server

จากรูปจะเห็นได้ว่า การที่จะสามารถติดต่อ และใช้งานระบบได้นั้นต้องผ่านขั้นตอนเริ่มแรกตั้งแต่ระบบปฏิบัติการไปยัง SQL Server แล้วเข้าสู่ฐานข้อมูล จึงสามารถใช้งานแต่ละออบเจ็กต์ในที่สุด ดังนั้น จะเห็นว่า หากมี Login Authentication ที่สามารถผ่านเข้ามายัง SQL Server ได้ แต่ก็ยังไม่สามารถเข้าใช้ฐานข้อมูลต่าง ๆ ได้ หากไม่ได้รับ Login สำหรับเข้าใช้ฐานข้อมูลแต่ละฐานข้อมูล

Roles เป็นการจัดแบ่งกลุ่มตามความสามารถตามการใช้งานของแต่ละฐานข้อมูล ซึ่งสามารถสร้าง Roles เพิ่มขึ้นมาเองได้ ซึ่งในการสร้าง Role ขึ้นมาเองจะสามารถกำหนดการเข้าถึงแต่ละฐานข้อมูลได้ ซึ่งในการสร้าง Role ขึ้นเองนี้สามารถสร้างด้วย Enterprise manager ขึ้นได้

ฐานข้อมูล SQL Server มีตัวอย่างคำสั่งต่าง ๆ ที่เกี่ยวข้องต่อระบบงานที่สร้างดังนี้

การเพิ่มสมาชิกเข้าไปใน Role ที่สร้างขึ้น

```
sp_addrolemember 'Role_Name','Login_Name'
```

การลบสมาชิกออกจาก Role ที่สร้างขึ้น โดยใช้ Store Procedures

sp_droprolemember 'Role_Name','Login_Name'

จะเห็นว่า ในการเพิ่มและลบสมาชิกออกจาก Role นั้นจะมีตัวแปรอยู่สองค่า ดังนี้

- Role_name : เป็นชื่อของ Role ที่ต้องการเพิ่มหรือลบสมาชิก
- Login_name : เป็นชื่อของ Login ที่มีอยู่ ที่ต้องการเพิ่มหรือลบ Role นั้น ๆ

จากเรื่องของ Role จะสามารถนำเข้ามาประยุกต์ใช้เพื่อสร้างการพิสูจน์ตัวจริงโดยลายมือชื่อดิจิทัล โดยหากผู้ใช้คนใด ถูกพิสูจน์ตัวจริงแล้วว่าเป็นบุคคลนั้นจริง ก็จะทำการเพิ่มผู้ใช้คนนั้นเข้าไปยัง Role ที่สร้างขึ้น เพื่อให้ผู้ใช้สามารถเข้าใช้ข้อมูลในฐานข้อมูลได้

การเพิ่มข้อมูลในฐานข้อมูล SQL Server มีคำสั่งดังนี้ ซึ่งเป็นคำสั่งที่ใช้ในการใส่ข้อมูลได้เพียงหนึ่งแถวต่อหนึ่งคำสั่ง

```
INSERT [INTO]   table_name [( column_list)]
VALUES ( { DEFAULT | NULL | expression } )
```

โดยที่

- table_name : เป็นชื่อตารางที่ต้องการใส่ข้อมูล
- column_list : ชื่อคอลัมน์ที่ต้องการใส่ข้อมูล
- DEFAULT | NULL | expression : คือข้อมูลที่ใส่ลงในคอลัมน์ต่าง ๆ ตามที่ระบุใน column_list ถ้าเป็น DEFAULT คือ ใส่ค่าที่กำหนดไว้เป็นดีฟอลต์ของคอลัมน์นั้น หรือ NULL ก็จะใส่ค่า NULL และ expression คือค่าใด ๆ ที่กำหนดให้กับคอลัมน์นั้นและคั่นข้อมูลแต่ละตัวด้วย , [comma]

การ Update ข้อมูลในฐานข้อมูล SQL Server มีตัวอย่างดังนี้ ซึ่งเป็นคำสั่งที่ใช้ในการใส่ข้อมูลได้เพียงหนึ่งแถวต่อหนึ่งคำสั่ง

```
UPDATE TABLE1
SET AUDIT = UNAME
WHERE NAME = NAME1
```

- Table1 : เป็นชื่อตารางที่ต้องการใส่ข้อมูล
- Audit : เป็นชื่อของคอลัมน์ที่กำหนด

- Name : เป็นชื่อของแถวที่ต้องการ Update

ตัวอย่างการติดต่อกับ SQL Server จาก Visual Basic มีคำสั่งดังนี้

```
Dim cn  As New ADODB.Connection
.....
.....
If cn.State = adStateOpen Then cn.Close
cn.Open "Provider=SQLOLEDB.1;Data Source=JANN;Initial
catalog=test;user id=sa;password=abc;"
```

- เป็นการเปิด connection ด้วยรหัสของ “sa” ที่ฐานข้อมูล “testr”

ในการสร้างตารางขึ้นมาใหม่อาจใช้ Enterprise Manager ของ SQL Server ก็ได้ ซึ่งการสร้างตารางจะสามารถกำหนดฟิลด์และชนิดของข้อมูลได้ และในการเก็บข้อมูลต่าง ๆ ในตาราง จะต้องมีการกำหนดชนิดของแต่ละคอลัมน์ ให้เหมาะสม ซึ่งในการกำหนดชนิดของข้อมูลในตาราง หากเป็นข้อมูลธรรมดา เช่น ชื่อ จะถูกเก็บไว้ในรูปแบบของ nvarchar เพื่อความเหมาะสม จะเห็นว่า SQL Server สามาระรองรับข้อมูลได้หลายประเภท ใช้พื้นที่ในการจัดเก็บไม่เท่ากัน ดังนั้น จึงควรเลือกชนิดของข้อมูลให้ใกล้เคียงกับสิ่งที่เก็บให้มากที่สุด เพราะจะเป็นการประหยัดทรัพยากรได้

บทที่ 4
การสร้างแอพพลิเคชั่น

4.1 ขั้นตอนการเตรียมสภาพแวดล้อมสำหรับระบบพิสูจน์ตัวจริง

4.1.1. เตรียมสภาพแวดล้อมของเครื่อง Client/Server

การเตรียมสภาพแวดล้อมสำหรับระบบพิสูจน์ตัวจริงเพื่อฐานข้อมูล Microsoft SQL Server มีการเตรียมติดตั้งโปรแกรม และเครื่องมือต่าง ๆ ทั้งทางฝั่งเครื่องที่ให้บริการ(Server) และเครื่องที่ขอใช้บริการข้อมูล(Client) ดังนี้

ทางฝั่งเครื่องที่ขอใช้บริการข้อมูล มีส่วนของโปรแกรมอยู่สองส่วน ในส่วนแรกเป็นการลงทะเบียนของผู้ใช้ของแต่ละคนที่ไม่เคยลงทะเบียนใช้ฐานข้อมูลมาก่อน เพื่อให้ผู้ใช้ลงทะเบียนและสร้างกุญแจคู่ขึ้นมา และอีกส่วน ไว้สำหรับให้ผู้ใช้สร้างลายมือชื่อดิจิทัล เพื่อส่งมายังเครื่องให้บริการข้อมูล

ทางฝั่งเครื่องที่ให้บริการ มีการติดตั้ง Microsoft SQL Server และเลือกค่าความปลอดภัยให้เป็น Mix Mode และในส่วนของโปรแกรมที่ถูกสร้างเพื่อตรวจสอบลายมือชื่อดิจิทัลที่ถูกส่งมาจากเครื่องขอใช้บริการ รวมถึงการสร้างตารางใน Microsoft SQL Server ขึ้นมาเพื่อเก็บข้อมูลต่าง ๆ ของผู้ใช้ที่ได้ทำการลงทะเบียน เพื่อใช้ในการตรวจสอบ และพิสูจน์ตัวจริงของผู้ใช้

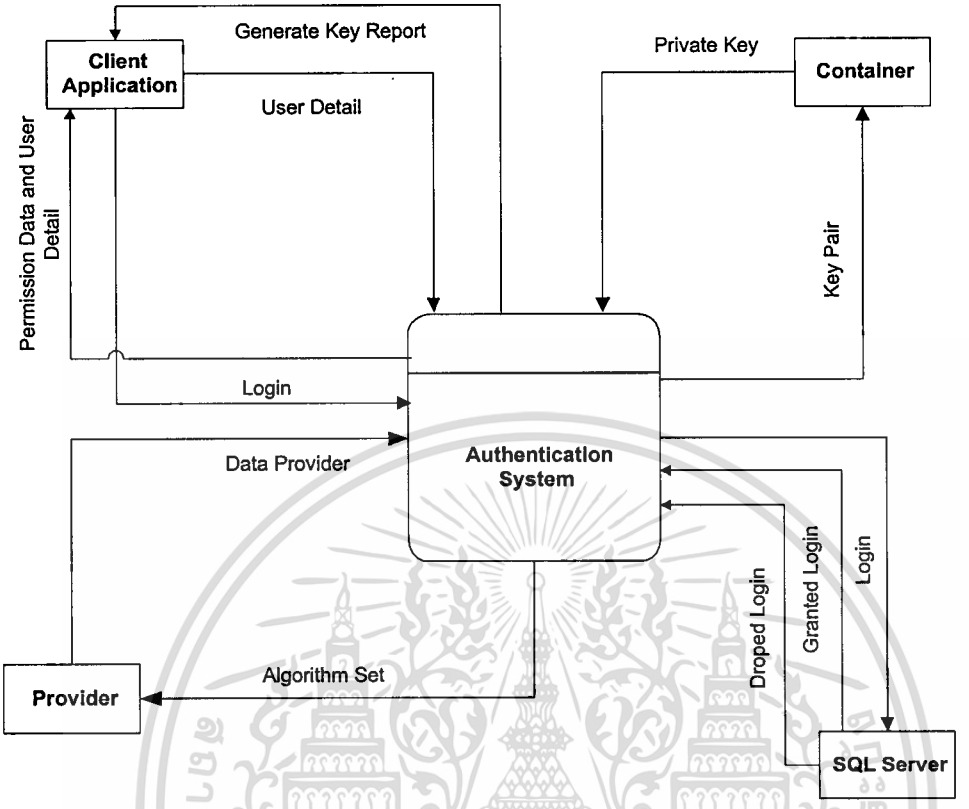
4.1.2. สร้างตาราง และจัดการ Role

สร้างตารางขึ้นมาสองตารางเพื่อไว้เก็บข้อมูลที่ลงทะเบียนของผู้ใช้งาน และเก็บค่า hash ของกุญแจสาธารณะเพื่อเป็นการตรวจสอบกุญแจสาธารณะที่ถูกเก็บไว้ ซึ่งในการสร้างตารางนี้จะต้องกำหนดค่าในแต่ละคอลัมน์ และชนิดของข้อมูล รวมไปถึงขนาดของข้อมูลที่จะใส่ลงไปในตารางด้วย

สำหรับการจัดการ Role ให้กำหนด Role และกำหนดสิทธิ์ในการเข้าถึงฐานข้อมูลต่าง ๆ ขึ้นมา เพื่อให้ ผู้ใช้สามารถถูกเพิ่มเข้ามาได้

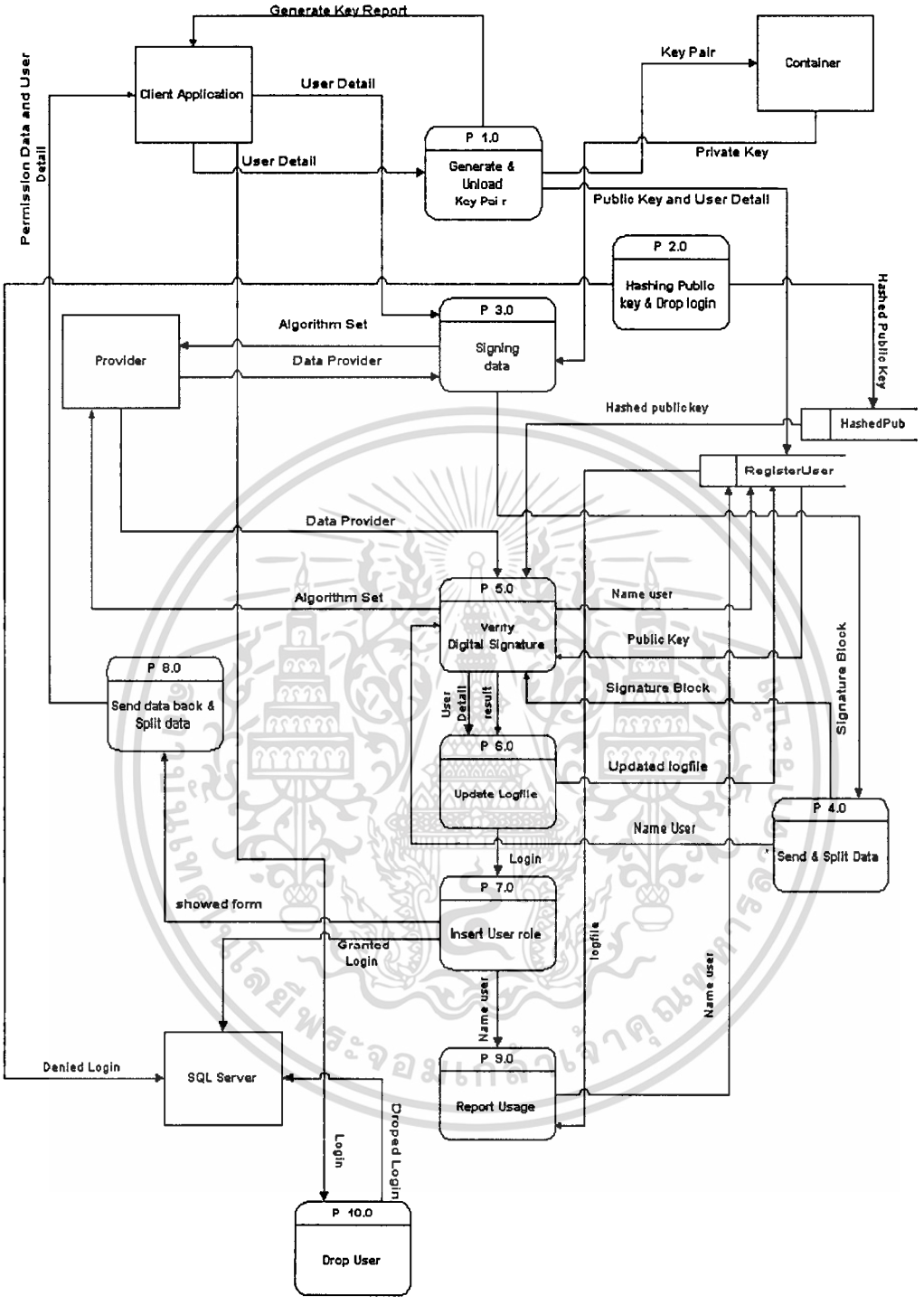
4.2 Context และ Dataflow Diagram

จากการวิเคราะห์ภาพรวมของระบบ ส่วนข้อมูลนำเข้า และส่งออก รวมทั้งส่วนที่เกี่ยวข้องกับระบบรวมใน Context Diagram มีดังนี้



รูปที่ 4.1 แสดง Context Diagram ของระบบการพิสูจน์ตัวจริง

จากรูปที่ 4.1 แสดงถึงดาต้าโฟลไดอะแกรมระดับ 0 ของระบบ โดยเริ่มจาก Client Application ส่งรายละเอียดมาจากผู้ใช้ เพื่อใช้ในการพิสูจน์ตัวจริง ซึ่งทางฝั่ง Server Application จะรับข้อมูลของลายมือชื่อดิจิทัลทั้งหมด แล้วออกคำสั่งส่งข้อมูล การตัดสินใจต่าง ๆ ว่าจะอนุญาตให้ผู้ใช้สามารถเข้าใช้ฐานข้อมูลได้หรือไม่



รูปที่ 4.2 แสดงดาต้าโฟล์ไดอะแกรมระดับ 1

จากรูปที่ 4.2 แสดงถึงดาต้าโฟลไดอะแกรมระดับ 1 ของระบบมีขั้นตอนย่อย ๆ 10 ขั้นตอน ดังนี้

- P 1.0 จัดการสร้างกุญแจ รับรายละเอียดข้อมูลผู้ใช้มาเพื่อสร้าง และเก็บกุญแจ จะส่งรายงานกลับมาหากการสร้างสำเร็จ รวมไปถึงเก็บข้อมูลต่าง ๆ ที่ฝั่งเซิร์ฟเวอร์ด้วย
- P 2.0 ขั้นตอนการแฮชและการระงับผู้ใช้ จะทำการแฮชกุญแจสาธารณะ แล้วทำการระงับ นั้นๆ ไว้ชั่วคราว ทำให้ไม่สามารถเข้าใช้ฐานข้อมูลได้
- P 3.0 ลงนามข้อมูล รับข้อมูลของผู้ใช้ เมื่อผู้ใช้ต้องการเข้าใช้ฐานข้อมูล โดยจะนำข้อมูลมาลงนาม โดยใช้กุญแจลับ
- P 4.0 ทำการแยกแยะข้อมูล ที่ส่งจาก Client ไปยัง Server ซึ่งข้อมูลในที่นี้ประกอบด้วยข้อมูลของชื่อ,อีเมลล์ และลายมือชื่อดิจิทัล
- P 5.0 ตรวจสอบลายมือชื่อ โดยนำค่าแฮชมาร่วมพิจาณา ในตรวจสอบลายมือชื่อว่าเป็นของจริงหรือไม่ ซึ่งจะต้องรับค่าจาก Provider มาพิจารณาถึงการตรวจสอบด้วย โดยที่ P1.0,P2.0 ก็จะทำเช่นกัน
- P 6.0 ขั้นตอนการเพิ่มข้อมูลการใช้งาน ให้ทำการ Update logfile เมื่อการตรวจสอบลายมือชื่อสำเร็จ ผลที่ได้จะต้องถูก Update ลงในไฟล์จากตาราง RegisterUser
- P 7.0 กำหนดให้ผู้ใช้สามารถเข้าใช้ฐานข้อมูลได้ หากการพิสูจน์ตัวจริงได้ผลลัพท์ว่าลายมือชื่อดิจิทัลเป็นของบุคคลนั้นจริง
- P 8.0 ส่งข้อมูลกลับ และแยกแยะข้อมูล เมื่อทำการตรวจสอบลายมือชื่อดิจิทัลสำเร็จจะกำหนดให้ว่าผู้ใช้สามารถเข้าใช้ข้อมูลได้หรือไม่
- P 9.0 รายงานการใช้งานด้วย logfile หากผู้ใช้มีการร้องขอดูรายงานการใช้งาน ให้ดำเนินการดึงข้อมูลจาก SQL Server เข้ามาแสดงผล
- P 10.0 ระงับการใช้งานฐานข้อมูล หลังจากผู้ใช้ทำการ Logout ออกไป ให้ทำการกำหนดไม่ให้ผู้ใช้สามารถเข้าใช้ฐานข้อมูลได้

4.3 ขั้นตอนการออกแบบระบบ

ในการออกแบบขั้นตอนการทำงานของระบบนั้นจะมีการศึกษาต่ะละขั้นตอน ซึ่งแบ่งออกเป็นแต่ะละขั้นดังนี้

4.3.1. ขั้นตอนการลงทะเบียน

ขั้นตอนการลงทะเบียน เพื่อนำเอากุญแจคู่ที่สร้างมาทำการลงนามเพื่อให้เกิดเป็นลายมือชื่อดิจิทัล ที่ใช้สำหรับการพิสูจน์ตัวจริง ซึ่งการออกแบบการสร้างขั้นตอนการลงทะเบียนสำหรับผู้ใช้ทุกคน แบ่งออกเป็น 7 ส่วน ดังนี้

1. ขั้นตอนการรับข้อมูลที่ลงทะเบียนจากผู้ใช้ ซึ่งเป็นข้อมูลของ ชื่อ,พาสเวิร์ด และอีเมลล์
2. ขั้นตอนสร้าง container จะนำชื่อที่ของผู้ลงทะเบียนที่ได้จากข้อ 1. มาเปิดเป็นชื่อของ container ซึ่ง container นี้ มีหน้าที่เก็บกุญแจต่าง ๆ ไว้ ซึ่งกุญแจจะถูกสร้างโดย WCCO Object
3. ขั้นตอนการสร้างกุญแจคู่ ซึ่งในที่นี้จะทำการสร้างกุญแจลายมือชื่อดิจิทัลขึ้นมาหนึ่งคู่ในแต่ละครั้ง ซึ่ง oSigkeys เป็น CRSAKeyPair นั่นเอง แต่ก่อนจะใช้คำสั่งนี้จำเป็นต้องให้ CryptoAPI รู้ว่าจะใช้อะไรเป็น Provider มีคำสั่งดังนี้

   ```
   If Not osigkeys.GenerateKeyPair Then Debug.Print osigkeys.LastError
   ```

   ซึ่งกุญแจที่สร้างขึ้นในที่นี้มีขนาดเท่ากับ 2048 bit ทั้งกุญแจลับและกุญแจสาธารณะ สำหรับการสร้างกุญแจในที่นี้จะใช้ Microsoft Enhanced Provider
4. ขั้นตอนโหลดกุญแจ จะทำการโหลดกุญแจคู่ที่สร้างขึ้นเข้าไว้ใน container เพื่อเก็บเอาไว้ใช้ภายหลัง

   ```
   If Not osigkeys.LoadKeyPair(vbNullString,
   DATA_FORMAT.CURRENT_KEY_CONTAINER) Then Debug.Print
   osigkeys.LastError
   ```

   ในการโหลดกุญแจเพื่อเก็บไว้ใน container ไม่ต้องเปลี่ยนแปลงรูปแบบของกุญแจ

5. ขั้นตอนการสร้าง log file ซึ่งในการสร้างไฟล์นี้ ตอนแรกของการลงทะเบียนจะเป็นเพียงการเปิดไฟล์ที่มีชื่อสอดคล้องกับชื่อผู้ใช้ เช่น ผู้ใช้ชื่อ "Somsak" จะทำให้ชื่อ log file มีชื่อว่า "Somsaklogfile" และถูกเก็บเอาไว้ในฐานข้อมูล SQL Server
6. ขั้นตอนการ Unload กุญแจสาธารณะ ที่ได้สร้างไว้ในขั้นที่ 3. ส่งไปยัง SQL Server ซึ่งในขั้นตอนของการ Unload กุญแจสาธารณะนี้ จะทำให้กุญแจสาธารณะสามารถกำหนดให้อยู่ในรูปแบบต่าง ๆ ได้ ซึ่งในที่นี้จะเปลี่ยนแปลงรูปแบบของกุญแจให้อยู่ในรูปแบบของไฟล์ เพื่อเก็บไว้ในฐานข้อมูล SQL Server ได้สะดวก ซึ่งมีคำสั่งดังนี้

```
If Not osigkeys.UnloadPublicKey(uname, DATA_FORMAT.FILE_HEX) Then
Debug.Print osigkeys.LastError
```

โดยที่ uname เป็นชื่อไฟล์ที่ต้องการจะเก็บเป็นกุญแจสาธารณะ

7. ขั้นตอนการติดต่อระหว่าง Client Application กับ Server เพื่อให้ Client Application ส่งข้อมูลที่ได้จากขั้นตอนก่อนหน้านี้ทั้งหมด เช่น ข้อมูลของชื่อผู้ใช้, อีเมลล์ และกุญแจสาธารณะเป็นต้น มายังฐานข้อมูลเพื่อเก็บไว้ในตาราง RegisterUser โดยใช้ Login ของผู้ใช้ทำการเข้าถึงฐานข้อมูล

4.3.2. ขั้นตอนการแฮชและระงับการใช้งาน

ขั้นตอนการแฮช ทำเพื่อให้มั่นใจว่า กุญแจสาธารณะที่เก็บไว้ มิได้ถูกเปลี่ยนแปลง และการระงับผู้ใช้ออก จะทำให้ผู้ใช้ไม่สามารถเข้าใช้ฐานข้อมูล ได้จนกว่าจะมีการพิสูจน์ตัวจริง

1. นำข้อมูลกุญแจสาธารณะจากที่เก็บไว้ในตาราง RegisterUser ขึ้นมาตามชื่อผู้ใช้ที่ระบุ
2. นำข้อมูลกุญแจสาธารณะเข้าแฮชฟังก์ชั่นเพื่อให้ได้ค่าแฮช
3. นำค่าแฮชที่ได้ ไปเก็บไว้ที่ฐานข้อมูล ที่ตาราง Hashedpub
4. เมื่อขั้นตอนการเก็บข้อมูลของผู้ใช้เสร็จสิ้น ให้ทำการระงับ Login ที่ใช้งานในฐานข้อมูล

4.3.3. ขั้นตอนการลงนาม

การสร้างขั้นตอนลงนามข้อมูล เพื่อให้เกิดลายมือชื่อดิจิทัลของผู้ใช้ขึ้นมา เพื่อใช้ตรวจสอบในกระบวนการพิสูจน์ตัวจริง มีขั้นตอนดังนี้

1. ขั้นตอนการรับข้อมูลของผู้ใช้ เพื่อนำมาใช้ในกระบวนการลงนามข้อมูล ซึ่งข้อมูลที่ได้รับ จะเป็นข้อมูลชื่อผู้ใช้ และอีเมลล์ของผู้ใช้
2. ขั้นตอนการตรวจหาชื่อ container ให้ตรงกับชื่อผู้ใช้ที่ระบุไว้ หากไม่มี container ตามที่ระบุเอาไว้แสดงว่า ผู้ใช้ไม่เคยลงทะเบียน หรือ container นั้นได้ถูกลบทิ้งออกไปแล้ว
3. ขั้นตอนการโหลดกุญแจเพื่อการลงนาม ในขั้นตอนนี้จะใช้กุญแจลับเพื่อการลงนามข้อมูล ซึ่งข้อมูลในที่นี้ ได้จากขั้นตอนที่ 1. คือข้อมูลของผู้ใช้และอีเมลล์
4. ขั้นตอนการโหลดข้อมูลที่จะลงนามให้อยู่ในรูปแบบที่ต้องการ นั่นคือเปลี่ยนรูปแบบของข้อมูล ดังนี้

```
If  Not omsgtxt.LoadMessageText(x, DATA_FORMAT.STRING_VB) Then
 Debug.Print omsgtxt.LastError
```

จะเห็นได้ว่าข้อมูลถูกเก็บให้อยู่ในรูปแบบของ string vb และ จะนำข้อมูลที่ได้มาโหลดเข้าเป็น Messagetext Object

5. ในการสร้างลายมือชื่อนี้ จะมีส่วนที่ใช้ hash value ดังนั้น ค่า default ของอัลกอริธึมของฟังก์ชั่นแฮชจะถูกตั้งให้เป็น MD5 เอาไว้แต่แรก
6. ขั้นตอนการลงนามข้อมูล ในส่วนนี้จะนำกุญแจที่ได้โหลดจากขั้นที่ 3. และข้อมูลจากขั้นที่ 4. มาทำการลงนามด้วยกุญแจลับ ด้วยอัลกอริธึมที่ได้ตั้งค่าไว้เป็น MD5 มีดังนี้

```
If Not osigkeys.Sign(omsgtxt, sSig, DATA_FORMAT.STRING_HEX) Then
 Debug.Print osigkeys.LastError
```

ซึ่งผลลัพท์ที่ได้จะเป็น Signature Block ที่ถูกเก็บอยู่ในรูปแบบของ string_hex

4.3.4. ขั้นตอนการส่งและแยกแยะข้อมูล

เมื่อสร้าง Signature Block ขึ้นมาเสร็จแล้ว ในขั้นตอนการส่งข้อมูลผ่านเครือข่ายนี้ จะทำการรวมเอา Signature Block มารวมกับข้อมูลของชื่อและอีเมลล์ของผู้ใช้มารวมกับเพื่อให้เกิดเป็นลายมือชื่อดิจิทัล แล้วส่งไปยัง Server Application เพื่อให้ทำการตรวจสอบ ซึ่งในการส่งข้อมูลต่าง ๆ นี้มีขั้นตอนดังนี้

1. ขั้นตอนการติดต่อ สำหรับทาง Client จะทำการเปิดช่องทางการสื่อสารด้วย winsock กับ Server เพื่อให้สามารถส่งข้อมูลไปประมวลผลทางฝั่ง Server ได้
2. ขั้นตอนการเตรียมข้อมูลในการส่ง ซึ่งข้อมูลในการส่ง ประกอบไปด้วย Signature Block รวมกับชื่อและอีเมลล์ของผู้ใช้ ส่งรวมไปเป็นข้อมูลหนึ่งชุด
3. ขั้นตอนการรับข้อมูล ที่ฝั่ง Server จะคอยรับข้อมูลที่ส่งมา และทำการแยกแยะว่าเป็นข้อมูลอะไรบ้าง

4.3.5. ขั้นตอนการทวนสอบลายมือชื่อ

กระบวนการตรวจสอบลายมือชื่อดิจิทัล มีขั้นตอนดังนี้

1. ขั้นตอนค้นหากุญแจ ทำโดยตรวจสอบตามชื่อที่ส่งมาจากไคลเอ็นต์แล้วเข้าไปค้นหาในตาราง RegisterUser หากพบชื่อให้โหลดคอลัมน์ "Pubkey" ขึ้นมา
2. ขั้นตอนโหลดกุญแจสาธารณะ จะนำกุญแจสาธารณะที่ได้จากขั้นตอนที่ 1. มาโหลดเข้า CRSAKeyPair Object ดังนี้

   ```
   If Not osigkeys.LoadPublicKey(name, DATA_FORMAT.FILE_HEX)
   Then Debug.Print osigkeys.LastError
   ```

   ซึ่ง "name" ในที่นี้ จะเป็นกุญแจสาธารณะที่โหลดได้จากตาราง RegisterUser ที่อยู่ในรูปแบบของไฟล์
3. ขั้นตอนการหาค่าแฮช จากกุญแจสาธารณะที่นำมาจากตาราง Register นำมาเข้าแฮชฟังก์ชั่น เพื่อให้ได้ค่าแฮช
4. นำข้อมูลจากตาราง Hashed pub มาเปรียบเทียบกับข้อมูลที่ได้จากขั้นตอน ที่ 3. หากได้ ตรงกัน แสดงว่าข้อมูลกุญแจสาธารณะมิได้ถูกเปลี่ยนแปลงไป
5. ขั้นตอนการ โหลดข้อมูลเข้า Messagetext Object เพื่อให้ทำการเปลี่ยนแปลงรูปแบบจากข้อมูลเดิมที่มีชนิดเป็นสตริง ให้กลายเป็น Byte Array
6. ขั้นตอนการขั้นตอนการตรวจสอบลายมือชื่อดิจิทัล จากลายมือชื่อที่ส่งมานำมาเข้าสู่การทำงานตรวจสอบ หากผลลัพท์ที่ได้เป็น "True" แสดงว่า ลายมือชื่อนั้นเป็นของจริง แต่ถ้าผลลัพท์ที่ได้เป็น "False" แสดงว่าลายมือชื่อนี้เป็นของปลอม ซึ่งมีการทำงานดังนี้

```
If Not osigkeys.VerifySignature(omsgtxt, sData,
DATA_FORMAT.STRING_HEX, ffvalid) Then
  Debug.Print osigkeys.LastError
```

จากคำสั่ง จะเห็นว่ามีการใช้ “omsgtxt” ซึ่งเป็น Messagetext Object และ “sData” ซึ่งเป็นส่วนของ Signature Block นำมาทำการตรวจสอบควบคู่ไปกับกุญแจสาธารณะที่ถูกโหลดเข้ามาก่อนหน้านี้ หากตรวจสอบแล้วจะเก็บผลลัพท์ที่ได้ไว้ใน “ffvalid”

4.3.6. ขั้นตอนการเพิ่มข้อมูลการใช้งาน

ในขั้นตอนนี้จะเป็นการ Update log file เมื่อมีการเรียกร้องการใช้งานเข้ามา

1. นำข้อมูลจากฟิลด์ที่ชื่อ “Audit” จากตาราง RegisterUser ที่สอดคล้องกับชื่อของผู้ใช้
2. ขั้นตอนนี้จะทำการเปิดไฟล์ที่ได้ เพื่อทำการ Update ข้อมูลภายใน จากผลของการเข้าใช้งานของผู้ใช้
3. ขั้นตอนการเก็บข้อมูลลงไฟล์ที่ได้เปิดเอาไว้ ให้นำค่าที่ได้จากการตรวจสอบลายมือชื่อ มา Update ลงไป

4.3.7. ขั้นตอนการกำหนดให้ผู้ใช้สามารถเข้าใช้ฐานข้อมูล

หากการพิสูจน์ตัวจริงได้ผลลัพท์ว่า ลายมือชื่อดิจิทัลเป็นของบุคคลนั้นจริง

1. ตรวจสอบผลลัพท์ที่ได้จากการตรวจสอบลายมือชื่อ
2. หากผลลัพท์ที่ได้ เป็น “True” ให้ เพิ่มชื่อของผู้ใช้คนนี้เข้าไปยังฐานข้อมูลและ Role ที่กำหนดไว้ในตอนแรก

4.3.8. ส่งข้อมูลกลับ และแยกแยะข้อมูล

สำหรับขั้นตอนนี้จะมีการส่งข้อมูล 2 ส่วน คือ ส่วนแรก เป็นการส่งข้อมูลแสดงผลของหน้าจอ ในส่วนที่สองเป็นข้อมูลของชื่อและอีเมลล์ ของผู้ใช้ ซึ่งมีขั้นตอนดังนี้

1. ขั้นตอนแรก นำค่าผลลัพท์ที่ได้จากขั้นตอนการตรวจสอบลายมือชื่อ มาใช้ตรวจสอบ

2. ตรวจหาค่า Status จากตาราง RegisterUser ที่สอดคล้องกับชื่อผู้ใช้ หาก Status มีค่าเป็น
3. ตรวจสอบอีเมลล์ปัจจุบันกับอีเมลล์ในตาราง RegisterUser หากมีข้อมูลไม่ตรงกัน ให้ Update ตามอีเมลล์ปัจจุบัน
4. นำค่าที่ได้จากทั้งสองขั้นตอนเบื้องต้นมาพิจารณาการเข้าใช้งาน
5. ส่งผลที่พิจารณาไปยัง Client Application เพื่อให้ทางฝั่ง Client แสดงผลตามข้อมูลที่ Server ส่งกลับไป

4.3.9. ขั้นตอนการรายงานการใช้งานด้วย logfile

ในการสร้างขั้นตอนตรวจสอบการใช้งานของผู้ใช้แต่ละคน โดยเก็บข้อมูลที่เข้าใช้งานของแต่ละคนเอาไว้ ที่ตาราง RegisterUser โดยขั้นตอนนี้จะเกิดขึ้นได้ต่อเมื่อ ผู้ใช้ถูกอนุญาตให้เข้าใช้งานฐานข้อมูล ซึ่งการสร้างขั้นตอนการตรวจสอบการใช้งาน มีดังนี้

1. ขั้นตอนตรวจหา ชื่อของผู้ใช้ ซึ่งข้อมูลของชื่อและอีเมลล์ของผู้ใช้ได้ถูกเก็บเอาไว้ตั้งแต่ผู้ใช้ได้เข้ามาในตอนแรก
2. ทำการติดต่อกับเซิร์ฟเวอร์โดยใช้ล็อกอิน “sa” ในการเข้าใช้ฐานข้อมูล
3. ตรวจหาคอลัมน์ “Audit” ตามชื่อของผู้ใช้
4. นำค่าที่ได้มาแสดงผลให้ผู้ใช้ทราบ นั่นคือ นำไฟล์ที่ได้จากฐานข้อมูลมาแสดงผลรายงานผู้ใช้

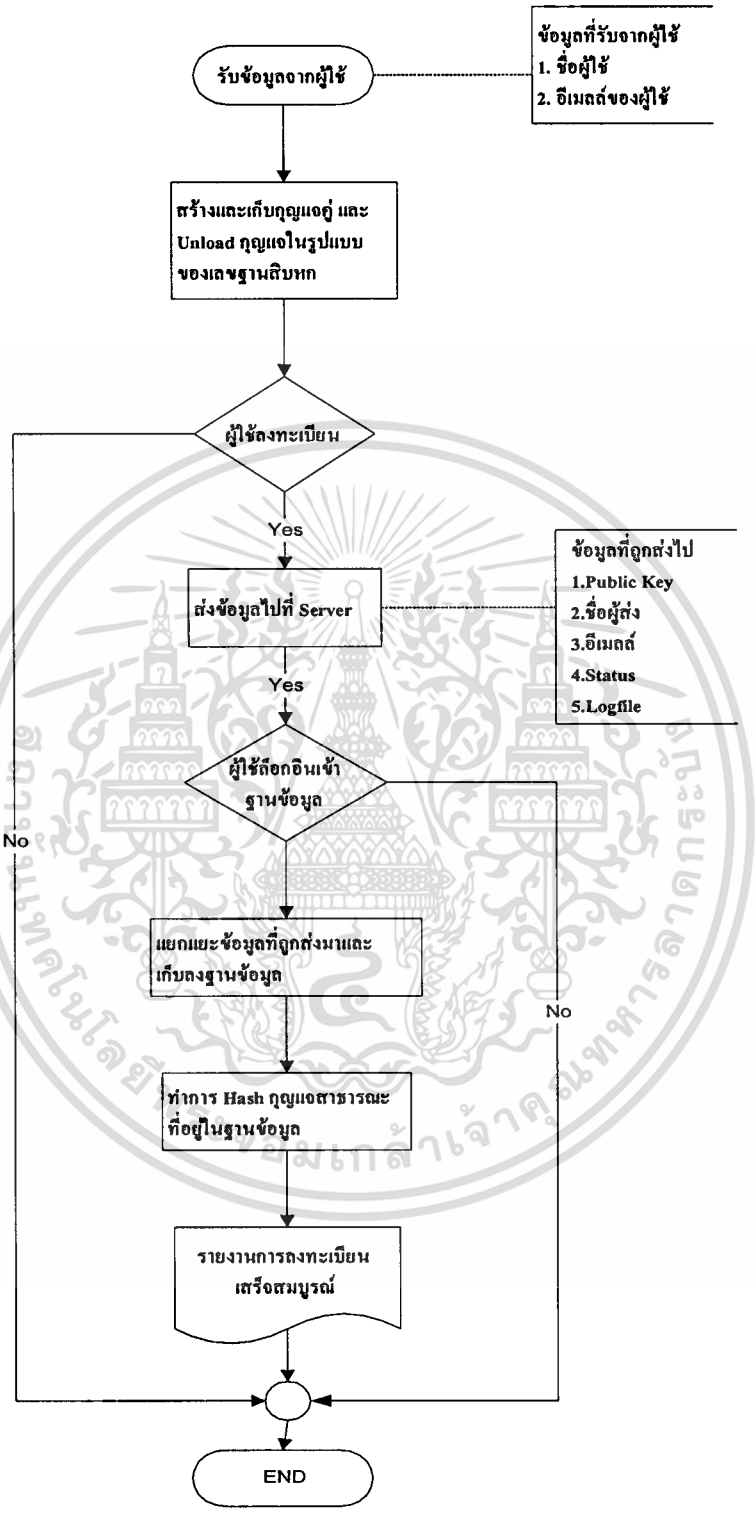
4.3.10. ขั้นตอนระงับการใช้งานฐานข้อมูล

หลังจากผู้ใช้ทำการ Logout ออกไป ให้ทำการกำหนดไม่ให้ผู้ใช้สามารถเข้าใช้ฐานข้อมูลได้

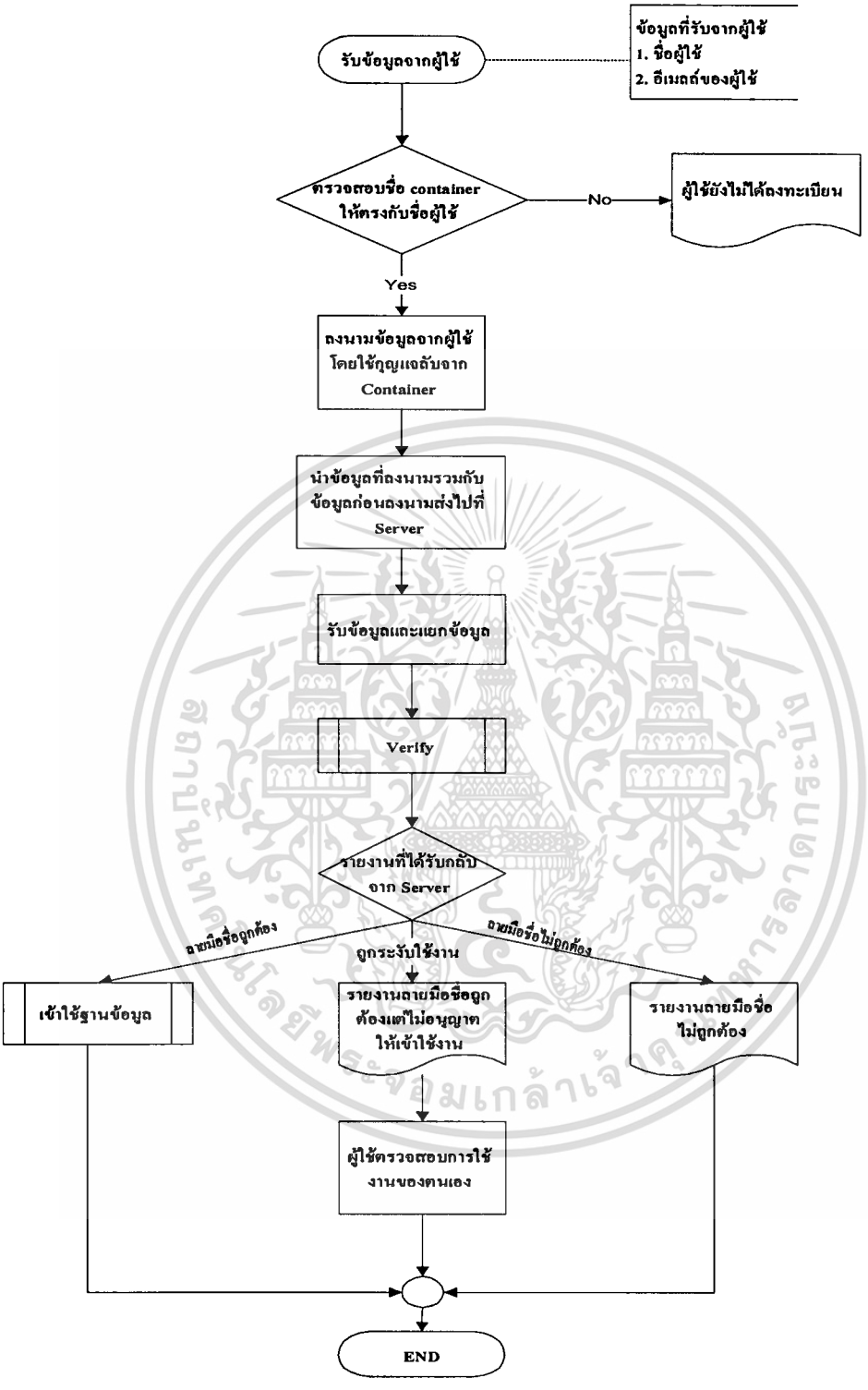
1. ส่งข้อมูลชื่อ Login ของผู้ใช้คนนั้น ไปยัง SQL Server เพื่อเป็นการระงับการใช้งานในฐานข้อมูลนั้น ๆ ของผู้ใช้ ด้วยคำสั่งที่ว่า

   sp_dropuser
2. ทำการส่งข้อมูลชื่อ Login ของผู้ใช้คนนั้น ไปยัง SQL Server เพื่อเป็นการลบชื่อ Login ของผู้ใช้ออกจาก Role ที่สร้างขึ้น ด้วยคำสั่งที่ว่า

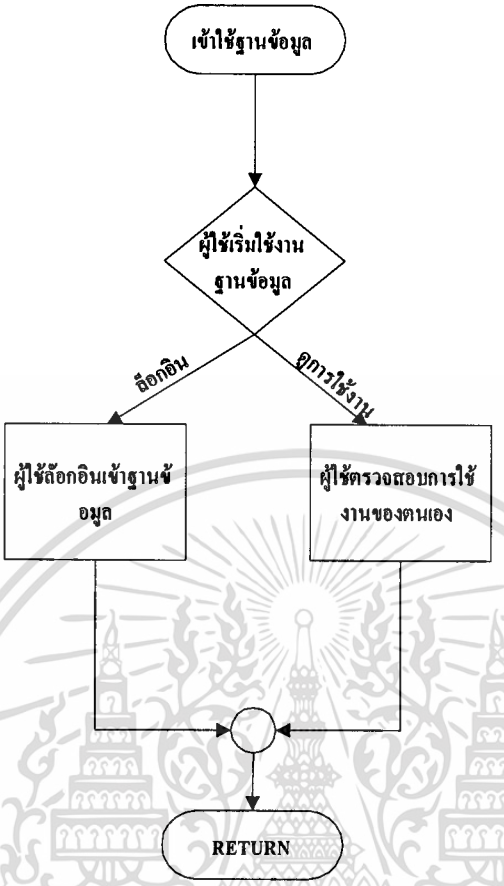
   sp_droprolemember



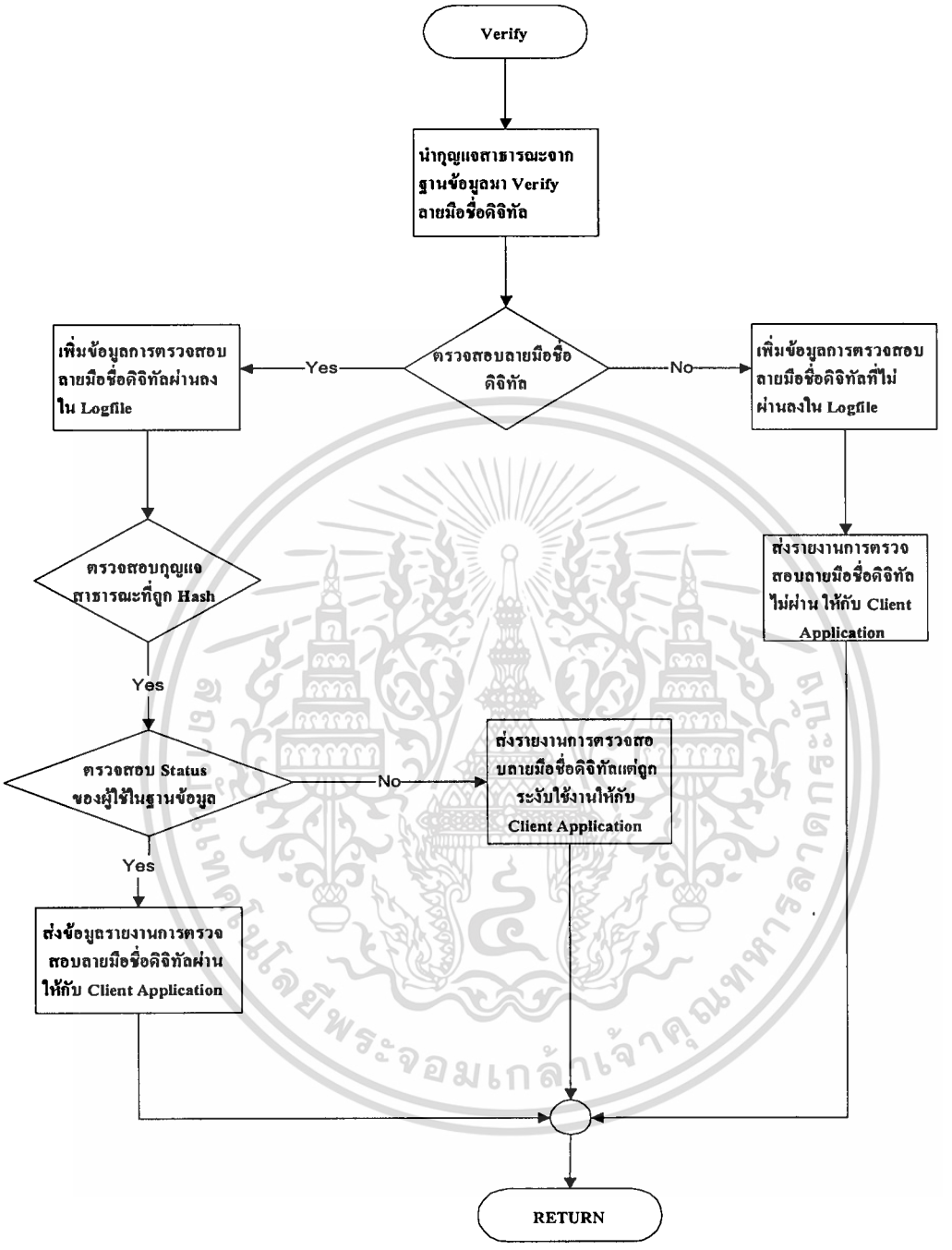
รูปที่ 4.3 แสดงFlowchart ในการลงทะเบียนผู้ใช้



รูปที่ 4.4 แสดง Flowchart การลงนาม



รูปที่ 4.5 แสดง Flowchart เข้าใช้ฐานข้อมูล



รูปที่ 4.6 แสดง Flowchart การตรวจสอบลายมือชื่อดิจิทัล

4.4 การกำหนดประเภทของตัวแปร

ในตอนแรกของการโปรแกรมที่เกี่ยวกับส่วนของ WCCO Object จะต้องกำหนดค่าตัวแปรให้อยู่ในรูปแบบดังนี้

Dim Oprov As Cprovider	กำหนดให้ Oprov เป็นส่วนของ Provider
Dim octr As Ccontainer	กำหนดให้ octr เป็นส่วนของ Container
Dim osigkeys As CRSAKeyPair	กำหนดให้ osigkeys เป็นส่วนของ CRSAKeyPair
Dim omsgtxt As Cmessagetext	กำหนดให้ omsgtxt เป็นส่วนของ Messagetext ที่ใช้สำหรับการโหลดข้อมูลเข้าเป็น Object ของมัน

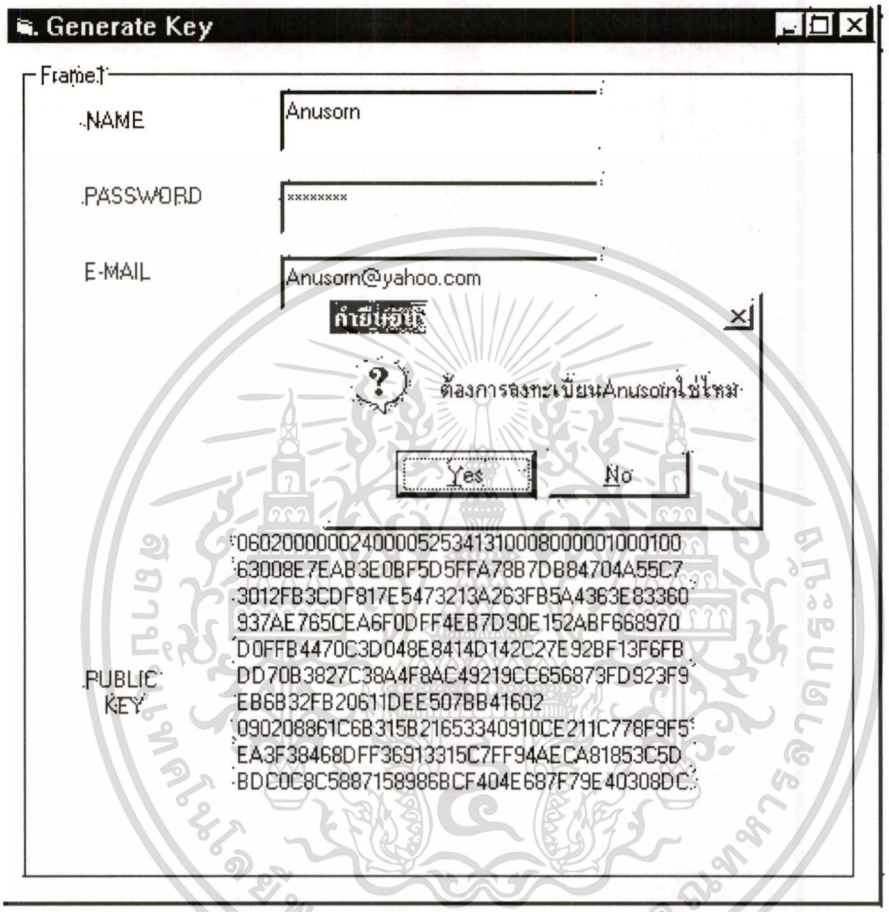
หลังจากที่เสร็จสิ้นของแต่ละขั้นตอน ให้ตั้งค่า object ต่าง ๆ ที่เคยตั้งไว้ให้เคลียร์

```
Set  oMsgtxt = Nothing
Set  oSigkeys = Nothing
Set  oCtr = Nothing
If  Not oProv.TerminateAll then Debug.Print oProv.LastError
Set  oProv = Nothing
```

4.5 ขั้นตอนการออกแบบแอพพลิเคชั่น

การออกแบบหน้าจอสำหรับผู้ใช้งานมีดังนี้

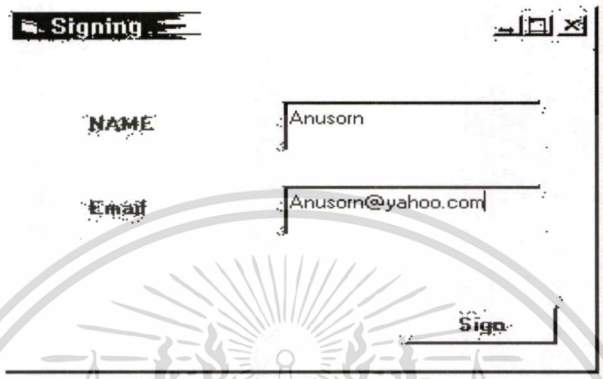
1. การลงทะเบียนเพื่อสร้างกุญแจ เป็นขั้นตอนให้ผู้ขอใช้ฐานข้อมูลเข้ามาลงทะเบียน เพื่อให้ระบบฐานข้อมูล SQL Server เก็บข้อมูลของผู้ใช้ เพื่อเอาไว้เป็นที่อ้างอิงในภายหลัง พร้อมด้วยกับการสร้างกุญแจคู่เพื่อใช้ในการสร้างลายมือชื่อดิจิทัลของผู้ใช้



รูปที่ 4.7 รูปการลงทะเบียน

จากรูปที่ 4.7 จะเห็นว่า มีการใส่ Login และ Password ของฐานข้อมูล เพื่อให้ผู้ใช้เก็บรายละเอียดข้อมูลของตัวเองได้ ถ้าหากผู้ใช้ไม่มี Login และ Password ก็ไม่สามารถจะเข้าไปทำการใด ๆ ในฐานข้อมูลได้เลย และหลังจากผู้ใช้ได้ทำการลงทะเบียนเสร็จ จะทำการลบความสามารถในการใช้ฐานข้อมูลของผู้ใช้ออก จนกว่า ผู้ใช้จะสามารถพิสูจน์ตัวจริงโดยใช้ลายมือชื่อดิจิทัลได้ จึงมีสิทธิ์ที่จะเข้าไปใช้ฐานข้อมูล

2. การพิสูจน์ตัวจริง เป็นขั้นตอนสำหรับผู้ใช้ที่ต้องการจะเข้าใช้ฐานข้อมูล SQL Server จึงต้องมีการสร้างลายมือชื่อดิจิทัล โดยข้อความที่ใช้ทำการเข้ารหัสนั้น เป็นชื่อ และอีเมลล์ของผู้ใช้เอง โดยที่หลังจากระบบสร้างลายมือชื่อดิจิทัลให้กับผู้ใช้สำเร็จ ผู้ใช้จะทำการส่งลายมือชื่อดิจิทัลมายังที่เครื่องเซิร์ฟเวอร์เพื่อทำการตรวจสอบ

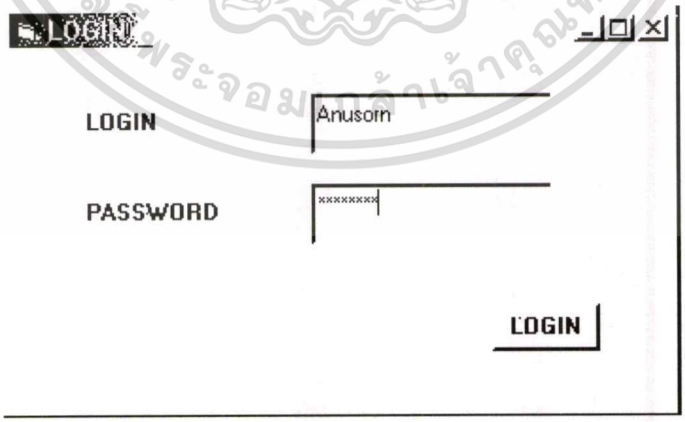


รูปที่ 4.8 การลงนามข้อมูล

3. การอนุญาตให้ใช้งานฐานข้อมูล เมื่อการทวนสอบลายมือชื่อสำเร็จ จะทำการตรวจสอบสิทธิ์ของบุคคลนั้น ๆ ว่ามีสิทธิ์เข้าใช้ฐานข้อมูลหรือไม่

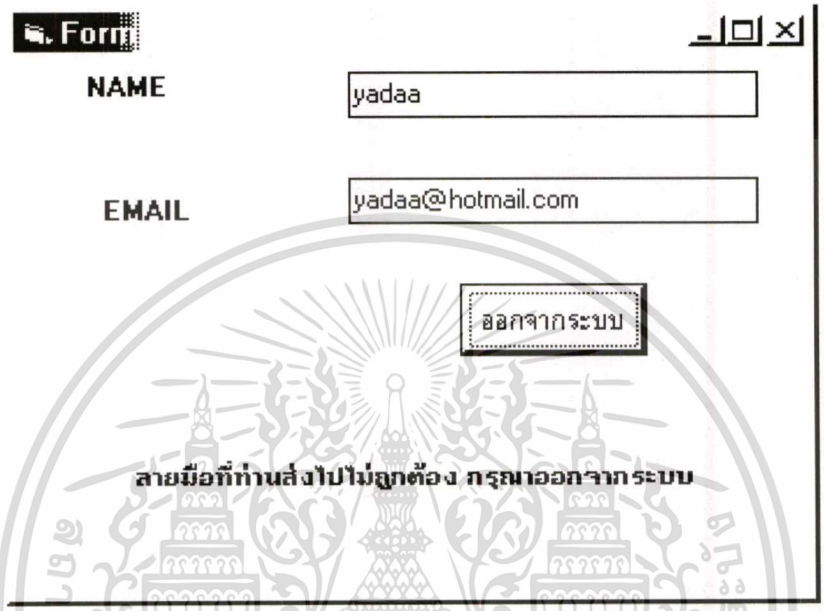
รูปที่ 4.9 การอนุญาตให้ใช้งานฐานข้อมูล

4. เมื่อผู้ใช้ถูกอนุญาติให้เข้าถึงฐานข้อมูลได้ ให้ผู้ใช้ใส่ Login และ Password ของผู้ใช้ ลงไป



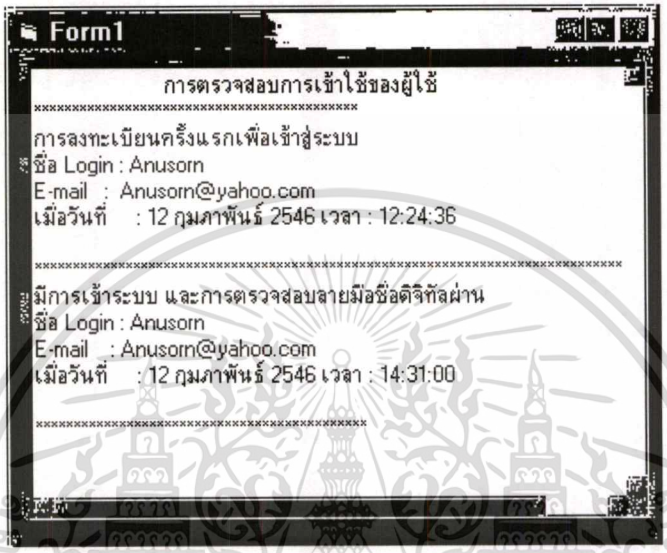
รูปที่ 4.10 การเข้าสู่ฐานข้อมูล

5. การอนุญาตให้ใช้งานฐานข้อมูล เมื่อการทวนสอบลายมือชื่อสำเร็จ จะทำการตรวจสอบสิทธิ์ของบุคคลนั้น ๆ ว่ามีสิทธิ์เข้าใช้ฐานข้อมูลหรือไม่ ซึ่งจากรูปนี้แสดงให้เห็นว่า หากพิสูจน์ลายมือชื่อออกมาแล้วเป็นของปลอม ทำให้ผู้ใช้ไม่สามารถเข้าสู่ระบบได้



รูปที่ 4.11 การตรวจสอบลายมือชื่อไม่ถูกต้อง

6. การตรวจสอบการใช้งาน สำหรับขั้นตอนนี้ เป็นขั้นตอนของการตรวจสอบและรายงานการใช้งานต่าง ๆ ของผู้ใช้ในแต่ละคน หากผู้ใช้คนใดต้องการตรวจสอบการใช้งานของตัวเอง ซึ่งผลที่ปรากฏ จะรายงานเวลาที่เข้ามาใช้ รวมไปถึงการตรวจสอบว่าผู้ใช้ถูกอนุญาติให้เข้ามาใช้ฐานข้อมูลหรือไม่

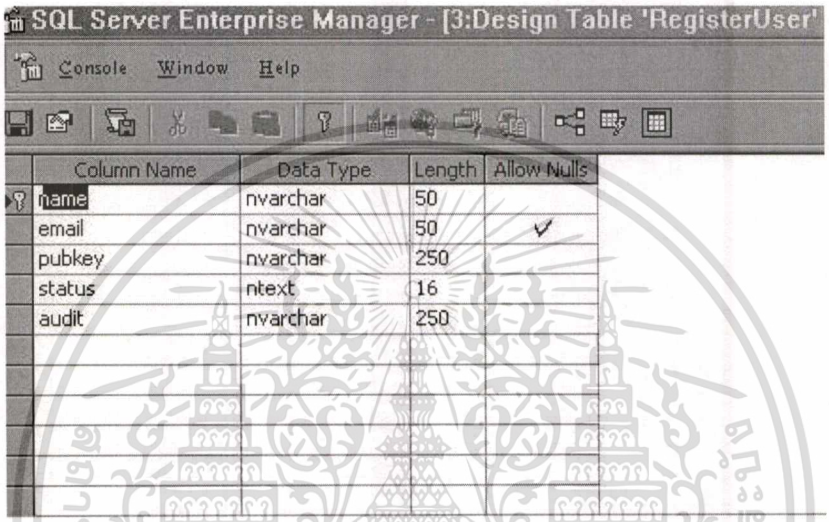


รูปที่ 4.12 การตรวจสอบการใช้งาน

4.6 ขั้นตอนการเก็บข้อมูลของผู้ใช้ใน SQL Server

เมื่อมีการลงทะเบียนของผู้ใช้แต่ละคนแล้ว ค่ารายละเอียดต่าง ๆ เช่น ชื่อ อีเมลล์ และกุญแจ จะถูกเก็บไว้ในตาราง Register User ซึ่งอยู่ในฐานข้อมูล SQL Server

มีรายละเอียดของแต่ละ field ดังนี้



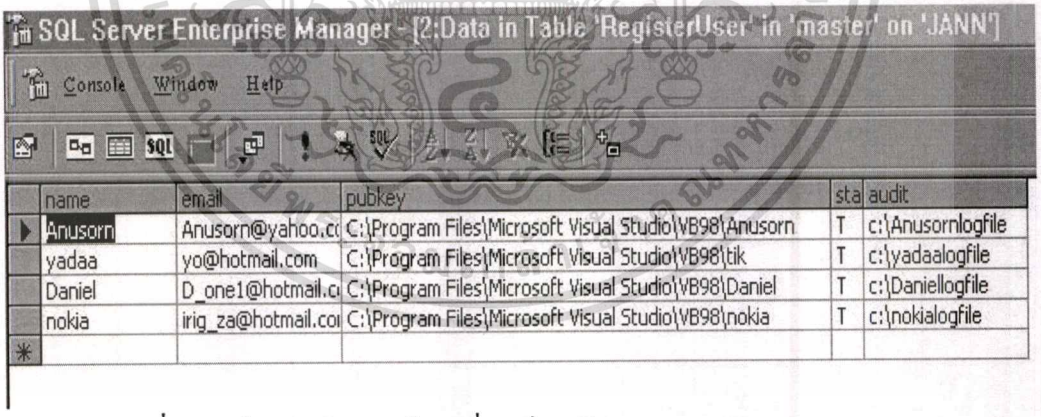
รูปที่ 4.13 การออกแบบตาราง RegisterUser

ตารางที่ 4.1 ตารางเก็บข้อมูลการลงทะเบียนของผู้ใช้

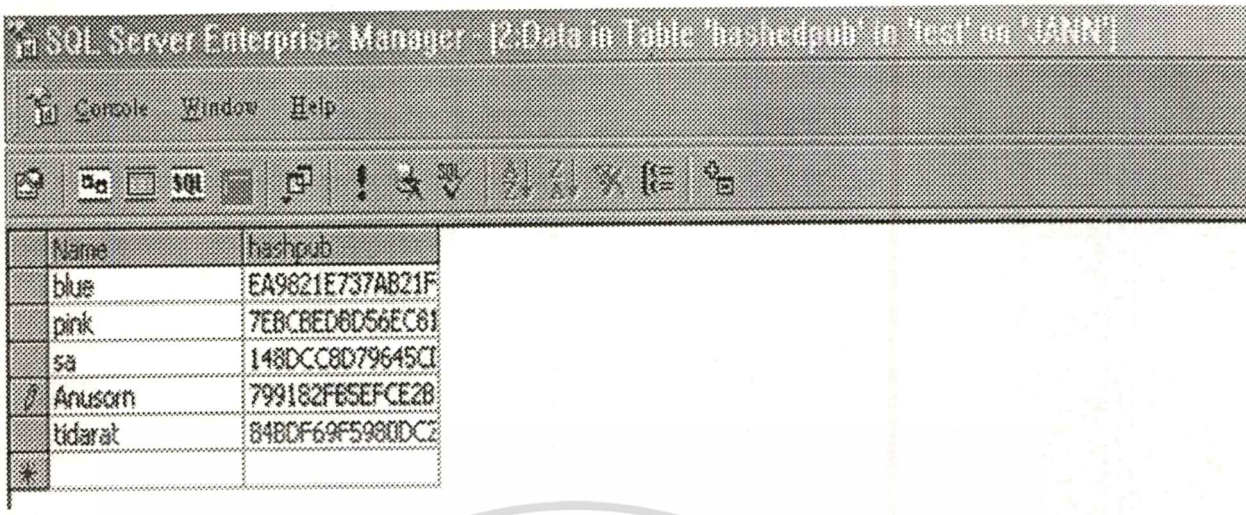
Field	รายละเอียด
Name*	ชื่อของผู้ลงทะเบียน
Email	อีเมลล์ของผู้ลงทะเบียน
PubKey	กุญแจสาธารณะที่ถูกสร้างขึ้นพร้อมกับกุญแจลับของผู้ลงทะเบียน
Status	การอนุญาตให้ใช้งานฐานข้อมูล (True/False)
Audit	การตรวจสอบผู้ใช้งาน

* เป็น Primary Key

สำหรับตารางที่ 4.1 เป็นตารางที่เก็บข้อมูลการลงทะเบียนของผู้ใช้ ซึ่งแต่ละ record จะเป็นข้อมูลของผู้ใช้ในแต่ละคนที่เข้ามาลงทะเบียนเอาไว้ ดังนั้น เวลาระบบมีการลงทะเบียนของผู้ใช้ ก็จะส่งค่า Name , E-mail และ PubKey มาเก็บไว้ยังตารางนี้ เพื่อเอาไว้อ้างอิงในภายหลัง และเมื่อระบบมีการตรวจสอบลายมือชื่อดิจิทัลที่ถูกส่งจากผู้ใช้มายังเครื่องเซิร์ฟเวอร์ จึงมีการนำเอา field ที่ชื่อว่า PubKey มาใช้งานเพื่อทวนสอบลายมือชื่อ โดยที่ตาราง Hashedpub ก็จะมีลักษณะในทำนองเดียวกันกับตารางนี้



รูปที่ 4.14 เป็นตัวอย่างของข้อมูลที่ถูกเก็บอยู่ในตาราง RegisterUser


SQL Server Enterprise Manager - [2 Data in Table 'hashedpub' in 'test' on 'JANN']

Console Window Help

	Name	hashpub
	blue	EA9821E737AB21F
	pink	7EBCBED6D56EC81
	sa	148DCC8D79645C
	Anusorn	799182FE5EFCE28
	tidarat	84BDF69F5980DC2
*		

รูปที่ 4.15 เป็นตัวอย่างของข้อมูลที่ถูกเก็บอยู่ในตาราง Hashedpub

จากรูปที่ 4.15 ตาราง Hashedpub เป็นตารางที่เก็บข้อมูลค่าแฮชที่ได้จากการนำกุญแจสาธารณะที่ได้จากตาราง Register แล้วนำเข้าแฮชฟังก์ชั่น ผลที่ได้ คือค่าแฮช แล้วนำค่าที่ได้เก็บไว้ที่ตารางนี้ เพื่อใช้ตรวจสอบกุญแจสาธารณะว่าถูกเปลี่ยนแปลงหรือไม่

บทที่ 5
สรุปผลการปฏิบัติงาน

5.1 ผลการดำเนินงาน

จากการศึกษาและสร้างระบบการพิสูจน์ตัวจริงเพื่อฐานข้อมูล Microsoft SQL Server สามารถสรุปผลการดำเนินงานในส่วนต่าง ๆ ได้ดังนี้

5.1.1 การศึกษาและสร้างโปรแกรม

- การนำลายมือชื่อดิจิทัลมาประยุกต์ใช้ ทำให้ความปลอดภัยในฐานข้อมูล SQL Server มีความปลอดภัยที่แข็งแกร่งขึ้น
- ในส่วนของการสร้างโปรแกรมโดยนำเอา WCCO Object เข้ามาใช้ทำให้การสร้างโปรแกรมง่ายขึ้น

5.1.2 การนำไปใช้งาน

- การเพิ่มความปลอดภัยเข้ามาใหม่ ทำให้เกิดความยุ่งยากมากขึ้น ต่อการเข้าใช้ระบบฐานข้อมูล
- ในแต่ละขั้นตอนอาจใช้เวลาในการปฏิบัติงานนานไม่มากเท่าไร แต่ก็ทำให้เกิดความปลอดภัยเพิ่มขึ้นจากเดิมในส่วนของ Mix Mode
- ให้ความมั่นใจต่อผู้ใช้และผู้ให้บริการว่า ข้อมูลที่ถูกต้องจะถูกใช้งานโดยผู้ที่มีสิทธิ์ถูกต้องเท่านั้น

5.2 ประโยชน์ที่ได้รับ

การสร้างขั้นตอนการพิสูจน์ตัวจริงให้กับ Microsoft SQL Server ให้แก่ผู้ปฏิบัติการทั้งผู้ใช้และผู้ให้บริการข้อมูล เพื่อเพิ่มความมั่นใจว่ามีเพียงผู้ใช้ที่ถูกต้องและมีสิทธิ์เท่านั้น จะได้รับการอนุญาตเท่านั้นที่สามารถเข้ามาใช้ในฐานข้อมูลได้ โดยขั้นตอนของการใช้งานโปรแกรมนี้ไม่ยุ่งยากและใช้เวลานานจนเกินไปเพื่อแลกกับความปลอดภัยที่ได้มา เพราะความปลอดภัยของข้อมูลเป็นสิ่งที่จำเป็น และยังมีขั้นตอนของการตรวจสอบการใช้งานของผู้ใช้ได้ หากผู้ใช้ต้องการตรวจสอบดูการใช้งานของตนเอง ระบบจะแสดงผลรายงานทั้งเวลา และกิจกรรมที่ผู้ใช้ได้ทำลงไป

เป็น และยังมีขั้นตอนของการตรวจสอบการใช้งานของผู้ใช้ได้ หากผู้ใช้ต้องการตรวจสอบดูการใช้งานของตนเอง ระบบจะแสดงผลรายงานทั้งเวลา และกิจกรรมที่ผู้ใช้ได้ทำลงไป

ซึ่งโปรแกรมง่ายต่อการปรับปรุงเปลี่ยนแปลง หากต้องการเปลี่ยนแปลงขนาดของกุญแจหรืออัลกอริซึมใหม่ เนื่องจากในส่วนของ CSP เป็นส่วนที่เป็นอิสระ ซึ่งระบบสามารถมี CSP ซึ่งเป็น provider ที่มีลักษณะต่างกันได้หลายแบบ

บรรณานุกรม

Microsoft Corporation. 2002. **About Cryptography [Online]**. Available:

Http://msdn.microsoft.com/library/en-us/security/security/about_cryptography.asp.

Microsoft Corporation. 2002. **CSP Architectural overview. [Online]**. Available:

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

Microsoft Corporation. 2002. **SQL Server 2000 Security**. [Online]. Available:

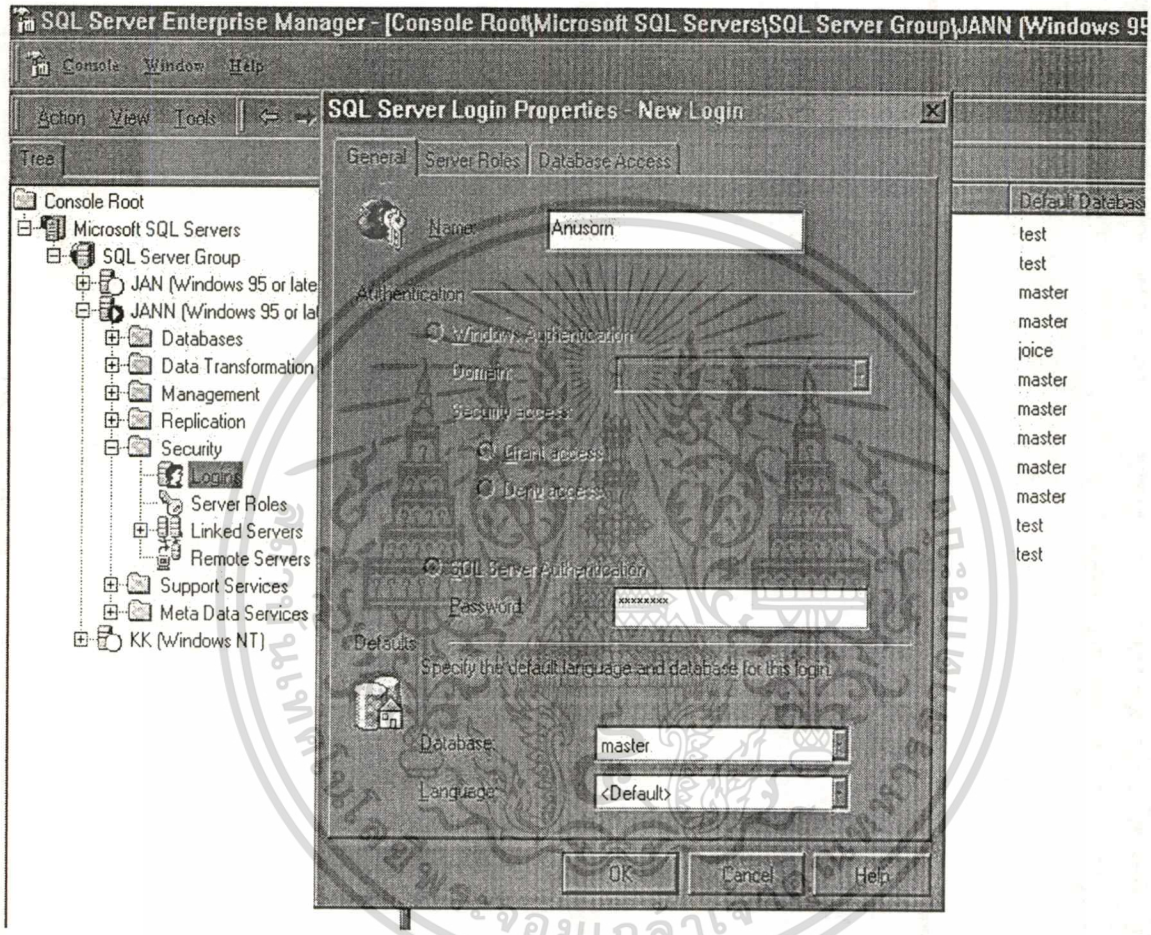
Http://www.microsoft.com/technet/protechnol/sql/maintain/security/sql2ksec.asp?frame=true.

Bondi, Richard. 2000. **Cryptography for Visual Basic.** New York : John Wiley & Sons.

ภาคผนวก

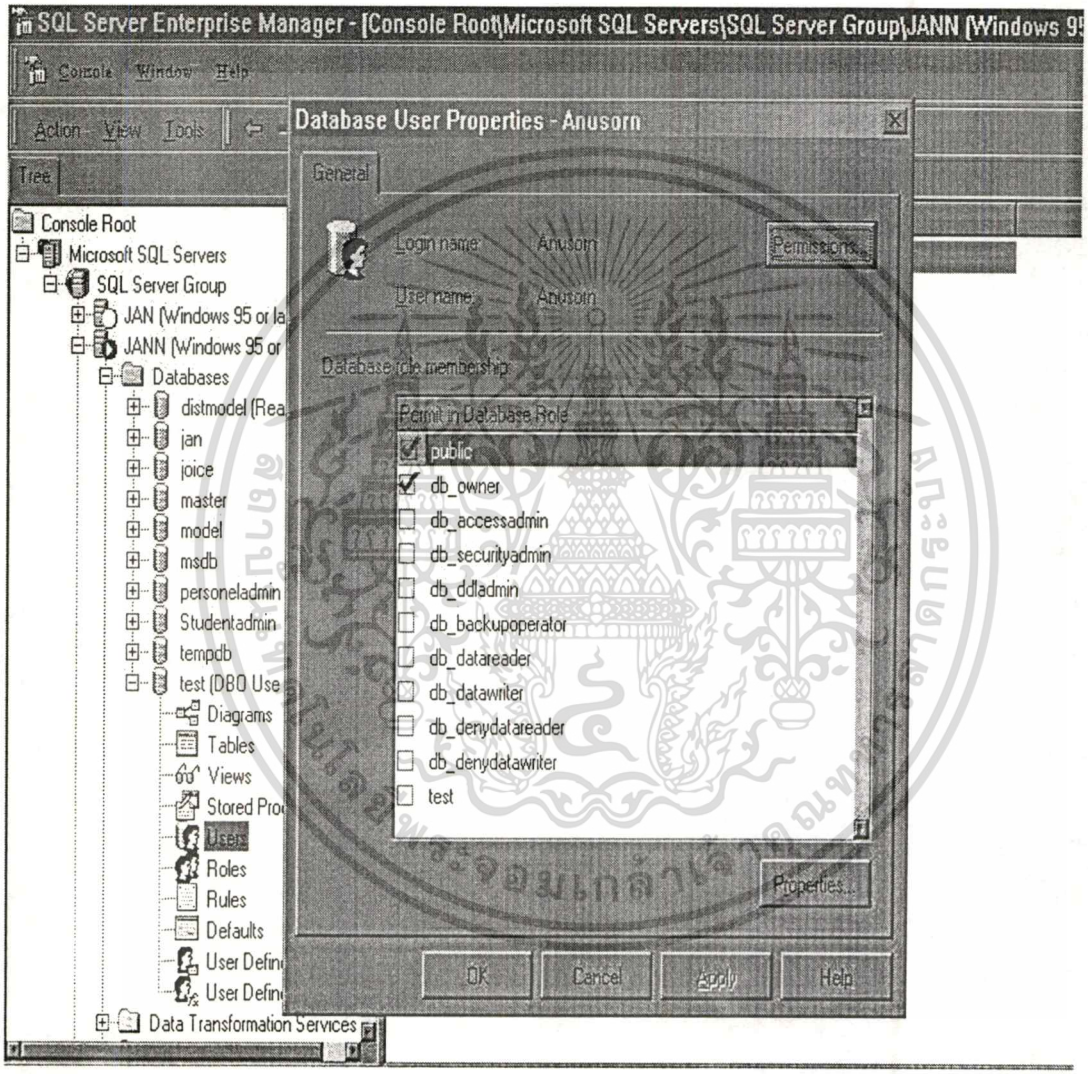
การสร้าง Login ของ Server ด้วย Enterprise manager

สำหรับการสร้าง Login เพื่อให้ผู้ใช้สามารถเข้าถึงฐานข้อมูลได้ จะทำโดยใช้ Enterprise manager ซึ่งจะเห็นว่า ถึงแม้ผู้ใช้จะสามารถล็อกอินเข้ามาภายใน SQL Server ได้ แต่หากมิได้ถูกอนุญาตให้เข้าใช้ฐานข้อมูลในนี้ ก็ทำให้ผู้ใช้เข้าใช้ข้อมูลต่าง ๆ ได้เลย ซึ่งในที่นี้ จะทำการสร้างล็อกอินเพื่อให้ผู้ใช้สามารถเก็บข้อมูลรายละเอียดต่าง ๆ ซึ่งประกอบไปด้วย ชื่อ,อีเมลล์ และกุญแจสาธารณะ



การสร้าง User ของฐานข้อมูลแต่ละอัน ด้วย Enterprise manager

หลังจากการสร้าง Login หากผู้ใช้ต้องการลงทะเบียน ให้ทำการเพิ่ม User ในฐานข้อมูลที่ชื่อว่า "test" ให้กับผู้ใช้ เพื่อให้ผู้ใช้สามารถใส่ข้อมูลของตัวเองลงไปได้ หลังจากผู้ใช้ทำการลงทะเบียนเสร็จให้ลบ Database User นี้ทิ้งเพื่อกันมิให้ผู้ใช้หรือบุคคลอื่นมาเปลี่ยนแปลงค่าในตาราง ได้ ซึ่งในการสร้าง Database User จะต้องกำหนด Permission ให้ด้วย ถึงจะสามารถทำการต่าง ๆ ในฐานข้อมูลได้



การสร้าง Database Role เพื่อให้ผู้ใช้สามารถเข้าใช้งานตาม Role ได้

หลังจากผู้ใช้สามารถพิสูจน์ตัวจริงได้แล้ว ผู้ใช้จะถูกกำหนดให้เป็น Database User นั้น ๆ และยังถูกกำหนดให้อยู่ใน Role ที่ได้สร้างขึ้น เพื่อให้ผู้ใช้สามารถเข้าถึงข้อมูลในฐานข้อมูลต่าง ๆ ได้