ระบบสารสนเทศผ่านเว็บเซอร์วิส

WEB SERVICES INFORMATION SYSTEM

นายวรเมธ วุฑฒะวนิช
นายวิชาติ อำไพภิญโญกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

ระบบสารสนเทศผ่านเว็บเซอร์วิส

WEB SERVICES INFORMATION SYSTEM

โดย

นายวรเมธ วุฑฒะวนิช

นายวิชาติ อำไพภิญโญกุล

อาจารย์ที่ปรึกษา

ดร.วรวัฒน์ ลิ้มโภคา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบสารสนเทศผ่านเว็บเซอร์วิส

Web Services Information System

ผู้จัดทำ

1. นายวรเมธ วุฑฒะวนิช รหัสประจำตัว 44010413
2. นายวิชาติ อำไพภิญโญกุล รหัสประจำตัว 44010445

_____อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโภคา)

ระบบสารสนเทศผ่านเว็บเซอร์วิส

นายวรเมธ วุฑฒะวนิช 44010413
นายวิชาติ อำไพภิญโญกุล 44010445
ดร.วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

เว็บเซอร์วิสถูกนำมาใช้ในการพัฒนาอย่างรวดเร็วและแพร่หลาย เนื่องจากมีความสามารถที่ช่วยให้นักพัฒนาโปรแกรมสามารถพัฒนาและเรียกใช้แอพลิเคชั่นร่วมกันได้อย่างอิสระโดยไม่ขึ้นกับแพลตฟอร์ม และไม่ผูกติดกับผู้ผลิตรายใดรายหนึ่ง เว็บเซอร์วิสสามารถที่จะทำงานในสภาวะแวดล้อมที่แตกต่างกันโดยผ่านโปรโตคอลกลาง แต่เนื่องจากบางระบบจะต้องใช้ความปลอดภัยค่อนข้างสูงอย่างเช่นระบบเกี่ยวกับการเงิน จึงจำเป็นต้องมีการรักษาความปลอดภัยของข้อมูลด้วย เทคโนโลยีที่ใช้ในด้านความปลอดภัยมีหลายอย่างแต่เทคโนโลยีที่ในด้านความปลอดภัยที่ใช้แพร่หลายอยู่ในปัจจุบันนี้คือ เอสเอสแอล แต่ก็ยังไม่สามารถตอบสนองความต้องการด้านความปลอดภัยในเว็บเซอร์วิสได้ทั้งหมด จึงได้มีการจัดทำเว็บเซอร์วิสซีเคียวริตี้ ขึ้นมา ซึ่งสามารถตอบสนองความต้องการในด้านรักษาความปลอดภัยบนเว็บเซอร์วิสได้มากกว่าแบบเดิม

จุดประสงค์ของปริญญานิพนธ์นี้คือการทำระบบเว็บเซอร์วิสให้มีความปลอดภัยโดยพัฒนาด้วยภาษาซีชาร์ป อีกทั้งยังมีการสาธิตวิธีการสร้างความปลอดภัยให้กับเว็บเซอร์วิส โดยการสร้างแอพพลิเคชั่นระบบสั่งซื้ออุปกรณ์คอมพิวเตอร์ และมีการใช้โปรโตคอล เว็บเซอร์วิสซีเคียวริตี้ ในการรักษาความปลอดภัยให้กับแอพพลิเคชั่น

Web Services Information System

Woramet Wuttawanit 44010413
Wichat Ampaipinyokul 44010445
Dr. Voravat Limpoka Advisor
Academic Year 2004

ABSTRACT

Web services has been more rapidly and widely because it can help programmer to develope program independently , independent platform. The web service can work in the different environment through standard Protocol. Because in some system must have a high security such as financial system ,etc., so we have to use technology in security. One of technology in security that used widely is SSL (Secure Socket Layer), but it can't support you completely. Therefore, we use WS-Security which can response for needed in security more than SSL. In this reason, we try to build application , web services and apply WS-Security.

The purpose of this thesis is to make the security on the web services application developed by. C#.net and We demonstrate how to provide the security to the web services application by developing security web services application, purchasing computer online system and use the ws-security.

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอกราบขอบพระคุณบิดามารดาที่ช่วยอบรมสั่งสอน เลี้ยงดูคณะผู้จัดทำ ส่งเสริมด้านการศึกษาหาความรู้ต่างๆ รวมถึงกำลังใจจนกระทั่งช่วยให้คณะผู้จัดจบการศึกษาสำเร็จด้วยดี

ขอขอบคุณพระคุณคร.วรวัฒน์ ลิ้มโภคาที่คอยให้คำปรึกษาและคำแนะนำต่างๆ เกี่ยวกับโครงงานนี้เป็นอย่างมาก ทำให้โครงงานครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี ตลอดจนคณาจารย์ทุกท่าน

นายวรเมธ วุฑฒะวนิช
นายวิชาติ อำไพภิญโญกุล

สารบัญ

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

สารบัญภาพ (ต่อ)

หน้า

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การดำเนินงานของผู้ใช้อินเตอร์เน็ตเริ่มเปลี่ยนรูปแบบ โดยเฉพาะการโต้ตอบแบบสองทางได้จึงสร้างบทบาทที่สำคัญหลายอย่างตามมา โดยเฉพาะการดำเนินงานทางธุรกิจ เริ่มตั้งแต่การโฆษณาประชาสัมพันธ์ผ่านทางเว็บ การให้เป็นสถานที่สำหรับการติดต่อหรือบริการข้อมูลข่าวสาร การทำธุรกรรมการซื้อขายในรูปแบบอีคอมเมิร์ซ (E-Commerce) การทำธุรกรรมอิเล็กทรอนิกส์ ซึ่งเป็นการแลกเปลี่ยนข้อมูลข่าวสารระหว่างกันบนเครือข่ายหรือที่เรียกว่า อีบิสสิเนส (e–business)

ปัจจุบันการพัฒนาเว็บได้ก้าวหน้าขึ้นไปมาก เพื่อรองรับการใช้งานที่มีความต้องการสูงขึ้น โดยเฉพาะการดำเนินการทางธุรกิจการค้าต่างๆ ที่เพิ่มมากขึ้นบนเครือข่าย ลักษณะงานที่ต้องการมีลักษณะการทำงานร่วมกันระหว่างองค์กร (interoperability) โดยให้โปรแกรมประยุกต์ขององค์กรหนึ่งส่งคำขอผ่านเครือข่ายอินเตอร์เน็ตด้วยโปรโตคอล HTTP (Hypertext Transfer Protocol) ไปยังเว็บบริการขององค์กรอีกองค์กรหนึ่ง มีการโต้ตอบเพื่อรองรับข้อมูลระหว่างกันแบบอัตโนมัติได้

ปัจจุบันบริษัทต่างๆ ได้ขยายคุณประโยชน์ของการดำเนินการแบบอัตโนมัติไปสู่ลูกค้าของบริษัทด้วยการสร้างโปรแกรมสำหรับให้บริการทางด้านข้อมูลและให้บริการทางด้านการประมวลผลโดยใช้โปรแกรมบริษัท และอนุญาตให้เครื่องคอมพิวเตอร์ของลูกค้าของบริษัทสามารถติดต่อสื่อสาร ขอบริการข้อมูลและบริการโปรแกรมจากเครื่องคอมพิวเตอร์ของบริษัทที่ติดตั้งโปรแกรมที่ให้บริการโดยตรง เพื่อดำเนินกิจกรรมต่างๆ ตามขั้นตอนที่ได้ออกแบบไว้ ซึ่งการให้บริการดังกล่าวก็คือ “เว็บเซอร์วิส(Web Service” นั่นเอง

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่ออำนวยความสะดวกแก่ผู้ใช้ในการทำงานข้ามแพลตฟอร์ม
2. เพื่อประโยชน์ในการใช้และการจัดการข้อมูลหรือซอฟต์แวร์ร่วมกัน
3. เพื่อศึกษาและพัฒนาเว็บเซอร์วิส
4. เพื่อศึกษาการเรียกใช้เว็บเซอร์วิสที่พัฒนาโดย .NET และ JAVA
5. เพื่อศึกษาถึงความปลอดภัยในการส่งผ่านเครือข่าย

1.3 ขอบเขตของงานวิจัย

1. สร้างระบบการซื้อขายที่สามารถติดต่อกับเว็บเซอร์วิส เช่น เซอร์วิสในการเช็คบัตรเครดิต เซอร์วิสในการขนส่ง เป็นต้น

2. สร้างระบบรักษาความปลอดภัยในการรับส่งข้อมูลไปยังเว็บเซอร์วิส

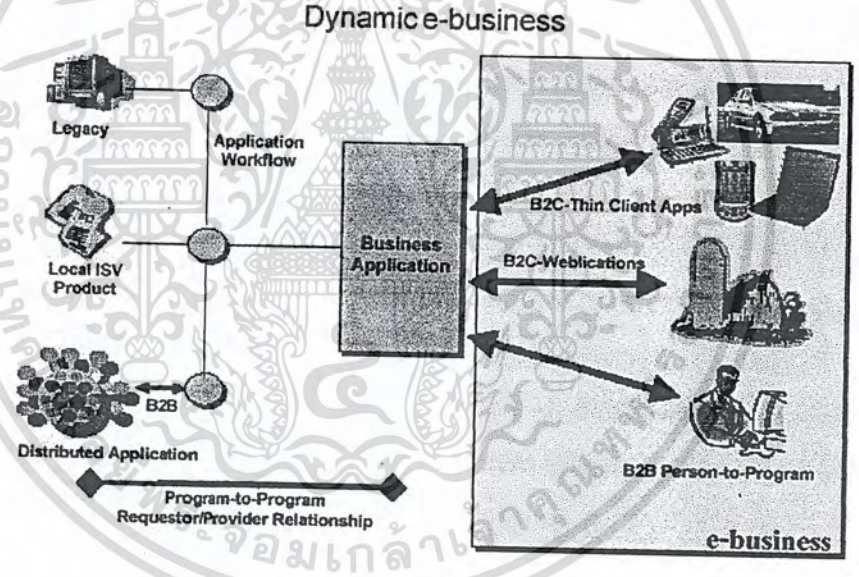
1.4 วิธีการดำเนินงาน

1. ศึกษาการทำงาน เครื่องมือและเทคโนโลยีต่างๆ ที่ใช้ในเว็บเซอร์วิส
2. ศึกษาและออกแบบการทำงานของโปรแกรม โดยออกแบบฐานข้อมูล เว็บแอพพลิเคชั่น
3. สร้างฐานข้อมูล เว็บแอพพลิเคชั่น และเว็บเซอร์วิส
4. ศึกษาและสร้างความปลอดภัยในการรับส่งข้อมูล
5. ทำการทดสอบและปรับปรุงแก้ไข

บทที่ 2
เว็บเซอร์วิส

2.1 หลักการของธุรกิจผ่านสื่ออิเล็กทรอนิกส์แบบไดนามิก

ไดนามิก-อีบิสซิเนสจะมุ่งความสนใจไปยังการยกระดับโครงสร้างพื้นฐาน และการบูรณาการด้านพาณิชย์อิเล็กทรอนิกส์แบบเชื่อมโยงธุรกิจกับธุรกิจ (Business-to-Business (B2B)) โดยการใช้ประโยชน์จากมาตรฐานบนอินเตอร์เน็ตและโครงสร้างพื้นฐานของอินเตอร์เน็ตให้เกิดประสิทธิภาพสูงสุด ทั้งภายในองค์กรและระหว่างองค์กร ไดนามิก-อีบิสซิเนสเกิดขึ้นโดยมีความมุ่งหวังว่าธุรกิจที่ดำเนินการผ่านทาง อินเตอร์เน็ตจะสามารถทำการติดต่อสื่อสารกันแบบอัตโนมัติ โดยเป็นการติดต่อกันระหว่างโปรแกรมกับโปรแกรม (P2P) หรือระหว่างแอพพลิเคชั่นกับแอพพลิเคชั่น (A2A) ดังรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างพื้นฐานการทำธุรกิจผ่านสื่ออิเล็กทรอนิกส์

หลักการพื้นฐานที่จะช่วยให้เราสามารถจัดการความยุ่งยากซับซ้อนของการบูรณาการดำเนินธุรกิจแบบ B2B

1) การรวมหรือการบูรณาการของซอฟต์แวร์ต่างระบบกันนั้น จะต้องอนุญาตให้แต่ละระบบเหล่านี้มีความเป็นอิสระจากกัน (Loosely Coupled)

2) อินเทอร์เฟสทางด้านการบริการของซอฟท์แวร์ที่จะนำมาทำการบูรณาการ ควรจะเผยแพร่สู่สาธารณชน และเปิดโอกาสของการเข้าถึงได้ง่าย

3) เมสเสจ (Message) ที่ใช้ติดต่อกันของการทำงานแบบโปรแกรมกับโปรแกรม (P2P) ต้องสอดคล้องกับมาตรฐาน เปิดบนอินเตอร์เน็ต

4) แอพพลิเคชั่นสามารถสร้างได้จากการใช้ซอฟท์แวร์คอมโพเน้นท์ (Software Component) จากภายนอกองค์กร โดยยึดตามแนวทางการดำเนินธุรกิจหลักขององค์กร

5) แหล่งซอฟต์แวร์คอมโพเน้นท์ (Software Component) ที่หาได้ง่าย ช่วยเพิ่มความยืดหยุ่นและเพิ่มคุณสมบัติ ส่วนตัวของกระบวนการทางธุรกิจ

6) การนำซอฟต์แวร์จากภายนอกองค์กรกลับมาใช้ใหม่ ช่วยให้เกิดการลดต้นทุนและ/หรือช่วยปรับปรุงประสิทธิภาพ ในการบริหารแก่ลูกค้า

7) ซอฟต์แวร์สามารถขายบริการได้

2.2 วิวัฒนาการของเว็บเซอร์วิส

เว็บเซอร์วิสนั้นอาจกล่าวได้ว่าเป็นยุคที่สามของอินเตอร์เน็ต กล่าวคือในยุคแรก การติดต่อสื่อสารบนเครือข่ายอินเตอร์เน็ตนั้น จะมีการนำเสนอข้อมูลข่าวสารต่างๆ อยู่บนเว็บซึ่งมีลักษณะเป็นแบบสแตติก (Static) ซึ่งข้อมูลต่างๆ ได้รับการออกแบบและกำหนดให้มีโครงสร้างที่แน่นอน และถูกเก็บไว้ในรูปแบบของเอกสาร HTML เป็นหลัก ข้อมูลเหล่านี้จะถูกเก็บไว้ในเซิร์ฟเวอร์และติดต่อกันผ่านทางโปรโตคอล HTTP เพื่อนำข้อมูลจากเซิร์ฟเวอร์มาแสดงผลผ่านเว็บบราวเซอร์

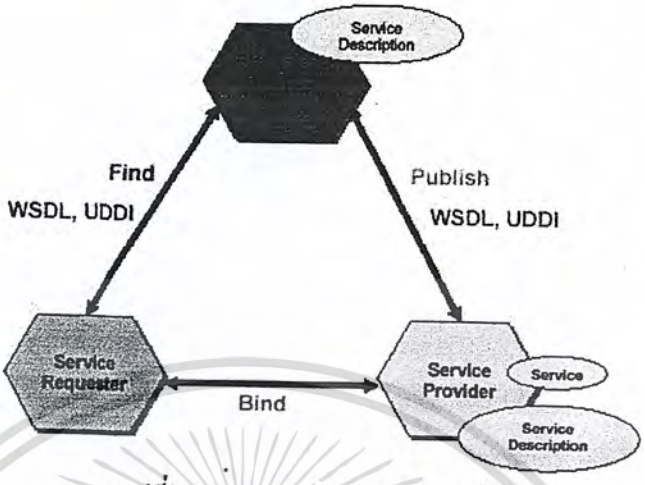
เมื่อเข้าสู่ยุคที่สองเป็นยุคที่เว็บมีการเชื่อมต่อกับระบบฐานข้อมูล และมีการประมวลผลบางอย่างทำให้ลักษณะของข้อมูลที่แสดงผลออกมามีลักษณะเป็นแบบไดนามิก มีลักษณะของการใช้งานภาษาสำหรับการโปรแกรมมากขึ้น แต่โปรแกรมต่างๆที่เขียนขึ้นส่วนใหญ่จะอยู่ที่เซิร์ฟเวอร์ และทางฝั่งผู้ใช้งานยังต้องใช้งานโปรแกรมเว็บบราวเซอร์ เพื่อแสดงผลที่ใช้งานอยู่เหมือนเดิม

ปัจจุบันได้ก้าวเข้าสู่ยุคที่สามของอินเตอร์เน็ต บริษัทหรือหน่วยงานต่างๆ ได้มีการสร้างโปรแกรมเพื่ออำนวยความสะดวกให้กับลูกค้าหรือผู้ใช้งาน สำหรับการบริการข้อมูลหรือการประมวลผลต่างๆ โดยใช้โปรแกรมที่สร้างขึ้น และอนุญาตให้เครื่องคอมพิวเตอร์ของลูกค้าหรือผู้ใช้งานสามารถติดต่อสื่อสาร ขอบริการข้อมูลและการประมวลผลโปรแกรมจากเครื่องคอมพิวเตอร์ของบริษัทหรือหน่วยงานที่ติดตั้งโปรแกรมที่ให้บริการโดยตรง เพื่อดำเนินงานต่างๆ ตามขั้นตอนที่ได้ออกแบบไว้ ซึ่งบริการดังกล่าวก็คือ "เว็บเซอร์วิส" นั่นเอง

2.3 เว็บเซอร์วิสคืออะไร

เว็บเซอร์วิส คือ แอพพลิเคชั่นหรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอพพลิเคชั่น อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นสื่อในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้ คอมโพเน้นท์ใด ๆ ก็ได้ ในแพลตฟอร์ม ใด ๆ ก็ได้ บนโปรโตคอล HTTP

2.4 แบบจำลองเว็บเซอร์วิส



รูปที่ 2.2 แสดงแบบจำลองเว็บเซอร์วิส

- ผู้ให้บริการ (Service Provider) จะทำการประกาศ(publish) บริการขององค์กรไปยังไดเร็กทอรี่ที่เก็บระเบียนของการบริการ (Service Registory)

- ผู้ขอใช้บริการ (Service Requestor) จะทำการค้นหา (find) บริการที่ต้องการ และเมื่อพบเห็นก็จะทำการเรียกใช้ (bind) ไปยังผู้ให้บริการนั้น

- ตัวแทนของผู้ให้บริการ(Service Broker) หรือที่อาจเรียกว่า "ไดเร็กทอรี่ของบริการ"

- UDDI เป็นวิธีการมาตรฐานสำหรับจัดเก็บและรวบรวมบริการต่าง ๆ ที่ให้บริการในรูปของไดเร็กทอรี่เซอร์วิส (Directory service) แต่ UDDI จะเกิดขึ้นได้ต้องอาศัยผู้ให้บริการจำนวน มาก เสนอบริการทางด้านซอฟต์แวร์หรือ โปรแกรมของตนเอง แล้วเราจะต้องประกาศ (publish) บริการเหล่านี้ไปบนอินเตอร์เน็ต

- WSDL เป็นมาตรฐานที่ใช้อธิบายคุณลักษณะของการเรียกใช้บริการของเว็บเซอร์วิสและวิธีการติดต่อกับเว็บเซอร์วิส

2.5 คุณลักษณะของเว็บเซอร์วิส

1) รายละเอียดในการสร้างและพัฒนาเว็บเซอร์วิสจะถูกซ่อนไว้(Encapsulated) เพื่อไม่ให้มองเห็นได้จากภายนอก ผู้เรียกใช้เว็บเซอร์วิสจะรู้จักเพียงอินเตอร์เฟซที่ผู้ให้บริการประกาศไว้เท่านั้น กล่าวคือเว็บเซอร์วิสเป็นประตูกั้นระหว่างระบบงานภายในกับผู้ใช้ภายนอก

2) ซอฟต์แวร์ที่ทำงานบนเว็บเซอร์วิสสามารถนำมาแก้ไขรายละเอียดภายในได้ โดยไม่ส่งผลกระทบต่อออกไปเป็นลูกโซ่ ทำให้การออกแบบซอฟต์แวร์เป็นไปได้ง่าย และผู้ใช้ที่ปลายทางไม่จำเป็นต้องโหลดซอฟต์แวร์ไว้มากเกินความจำเป็น

3) โปรแกรมที่เรียกใช้เว็บเซอร์วิสจะรับรู้ได้เอง ว่าเซอร์วิสที่กำลังจะเรียกใช้นั้นมีการกำหนดพารามิเตอร์อินพุตและเอาต์พุตอย่างไร

4) อยู่บนพื้นฐานของภาษา XML

5) สามารถแก้ไขได้ตลอดเวลา จึงสามารถอัพเดตได้อย่างรวดเร็ว และไม่จำเป็นต้องเรียก แอพพลิเคชั่นที่แก้ไขแล้วมารันใหม่

6) สนับสนุนการค้นหาและเรียกใช้แบบไดนามิก(Dynamic Discovery and Invocation) ด้วยเทคโนโลยี UDDI แอพพลิเคชั่น จึงค้นหาและเรียกใช้เว็บเซอร์วิสได้ในขณะรันไทม์ ซึ่งเพิ่มความยืดหยุ่นให้กับนักพัฒนาซอฟต์แวร์ เนื่องจากไม่จำเป็นต้องกำหนดการเรียกใช้เว็บเซอร์วิสไว้ก่อน บางทีก็เรียกคุณสมบัติข้อนี้กันว่า Just-in-Time

2.6 ประโยชน์ของเว็บเซอร์วิส

1) พันธมิตรทางการค้าสามารถเกิดขึ้นได้ตลอดเวลา โดยการค้นหาจาก UDDI

2) การดำเนินธุรกิจการค้าและบริการสามารถเป็นไปได้แบบอัตโนมัติในระดับของแอพลิเคชั่นกับแอพพลิเคชั่น โดยการแลกเปลี่ยนข้อมูลผ่านเว็บเซอร์วิส

3) เว็บเซอร์วิสสามารถใช้เป็นส่วนหนึ่งในการดำเนินธุรกิจตามเฟรมเวิร์คของ ebXML

4) ง่ายต่อการนำไปใช้งาน เนื่องจากในปัจจุบันมีเครื่องมือที่ใช้ช่วยเหลือในการพัฒนาเว็บเซอร์วิสมากมาย

5) ลดต้นทุนในการพัฒนาระบบบางอย่างที่ไม่จำเป็น โดยขอบริการจากเว็บเซอร์วิสของพันธมิตรทางการค้า

2.7 เทคโนโลยีพื้นฐานของเว็บเซอร์วิส

Service Directory	:	UDDI
Service Description	:	WSDL
Service Interaction	:	SOAP
Data Format	:	XML
Communication Protocol	:	HTTP
Communication Network	:	Internet

2.7.1 XML (Extensible Markup Language)

เป็นภาษามาร์คอัพที่เป็นเท็กซ์เบส (text-based) ที่ใช้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเตอร์เน็ตในปัจจุบัน ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) และอย่างที่เราทราบกันดีอยู่แล้วว่า XML คือมาตรฐานในการระบุโครงสร้างข้อมูลในรูปแบบแท็กซ์ด้วยความยืดหยุ่นที่สูงมากจึงก่อให้เกิดทั้งประโยชน์และโทษ เพราะในงานแบบเดียวกัน XML อาจจะมีได้หลายรูปแบบ การทำงานร่วมกันจึงยากขึ้น

2.7.2 SOAP (Simple Object Access Protocol)

เป็นโปรโตคอลที่ใช้ภาษา XML เป็นพื้นฐาน หรือเป็นเมสเสจจิ่งโปรโตคอล (Messaging protocol) สำหรับใช้ในการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (Distributed Environment) SOAP ได้กำหนดเมสเสจจิ่งโปรโตคอลระหว่างผู้ขอบริการ และผู้ให้บริการ ในการติดต่อสื่อสารกัน เช่น กำหนดให้ผู้ขอบริการต้องส่งข้อมูลที่ระบุฟังก์ชันและค่าพารามิเตอร์ต่างๆ ที่จำเป็นต้องใช้ในแอพลิเคชั่นที่ร้องขอ ส่งไปให้กับผู้ให้บริการ ซึ่งแอพลิเคชั่นของผู้ให้บริการก็จะทำงานตามกระบวนการที่ถูกร้องขอมา

SOAP ใช้โปรโตคอล HTTP ธรรมดาในการสื่อสาร ฉะนั้นจึงไม่ค่อยมีปัญหาในการเข้าออกระบบเครือข่ายที่ใช้ไฟร์วอลล์ เพราะว่าไฟร์วอลล์ส่วนใหญ่ต่างก็อนุญาตให้ข้อมูลที่ใช้โปรโตคอล HTTP (พอร์ต 80) ผ่านได้อยู่แล้ว นอกจากนี้มันยังถูกออกแบบให้สามารถใช้งานร่วมกับ SMTP และ MIME ได้ด้วย ฉะนั้นการทำงานระหว่างโปรแกรมต่างเครื่องต่างเครือข่ายก็ทำได้สะดวกขึ้น

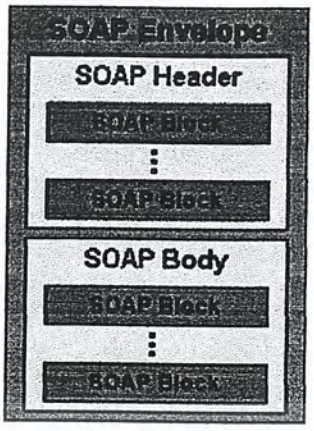
รูปที่ 2.3 แสดงการทำงานของ SOAP

ขั้นตอนการทำงานของ SOAP

1) แอพพลิเคชั่นของผู้ขอบริการสร้าง SOAP เมสเสจ เพื่อเรียกใช้บริการของเว็บเซอร์วิส
2) เว็บเซอร์วิสของผู้ให้บริการ ได้รับ SOAP เมสเสจ จากผู้ร้องขอ ซึ่งอยู่ในรูปแบบ XML
3) เว็บเซอร์วิสประมวลผลตามคอมโพเน้นท์ที่ให้บริการ แล้วผู้ให้บริการก็จะสร้าง SOAP เมสเสจที่มีผลลัพธ์นั้นกลับไปยังผู้ร้องขอบริการ
4) แอพพลิเคชั่นของผู้ขอบริการได้รับผลลัพธ์ที่เป็น SOAP เมสเสจ แล้วทำการแปลงให้อยู่ในรูปแบบที่ต้องการ เพื่อนำไปประมวลผลต่อ

SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP envelope
2. SOAP header
3. SOAP body



รูปที่ 2.4 แสดงโครงสร้างของเอกสาร SOAP

2.7.2.1 Envelope

SOAP เมสเสจ ทั้งหมดจะถูกห่อหุ้มในเอ็นวีโลบ (Envelope) ซึ่งมันจะเป็นอีลีเมนต์ (Element) ที่อยู่บนสุดของ SOAP เมสเสจ ในเอ็นวีโลบ นั้นจะต้องมีบอดี้อีลีเมนต์อยู่ 1 อีลีเมนต์เป็นอย่างน้อย ส่วนเฮดเดอร์อีลีเมนต์ (Header Element) นั้นอาจจะมีหรือไม่มีก็ได้ และเราสามารถที่จะสร้างอีลีเมนต์อื่นๆ นอกจากนี้ก็ได้ แต่จะต้องตามด้วยบอดี้

2.7.2.2 Header

เฮดเดอร์อีลีเมนต์ จะถูกใช้ในการส่งเมทตาอินฟอร์เมชั่น (meta information) ที่เกี่ยวกับ SOAP เมสเสจ แต่เมทตาอินฟอร์เมชั่นนี้จะไม่มีความจำเป็นสำหรับส่วนของการเรียกใช้งานฟังก์ชันใน SOAP เมสเสจจะมีการกำหนดค่า mustUnderstand attribute ให้อยู่ในส่วนของ เฮดเดอร์อีลีเมนต์ ซึ่งจะใช้กรณีที่เซิร์ฟเวอร์เกิดไม่เข้าใจอีลีเมนต์นั้นก็จะมีการส่ง เมสเสจแสดงความผิดพลาด(fault message)กลับมาบอก ซึ่งกระบวนการนี้จะทำให้เครื่องไคลเอนต์มั่นใจได้ว่า เครื่องเซิร์ฟเวอร์สามารถที่จะประมวลผลได้ โดยค่าแอททริบิวท์ตัวนี้จะมีค่าได้ 2 ค่า คือ 0 แสดงว่าเกิดความผิดพลาด และ 1 แสดงว่าปกติ ดังนั้นตัวอย่างข้างล่างนี้จะใช้เฮดเดอร์สำหรับส่งค่าไอดี(ID) จากเครื่องไคลเอนต์ไปยังเว็บเซอร์วิส ซึ่งเกี่ยวกับเรื่องระบบรักษาความปลอดภัย

```
<SOAP:ENV Envelope
        xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Header>
        <MYNS:CallerID
                xmlns:MYNS= "MY URI"
                SOAP-ENV: mustUnderstand = "1">
                WroxDomain/KayR
```

```
        </MYNS:CallerID>
        </SOAP-ENV:Header>
        <SOAP-ENV:BODY>
            <!-Call information-- >
        </SOAP-ENV:BODY>
</SOAP-ENV : Envelope>
```

2.7.2.3 Body

เป็นส่วนที่ต้องมีอยู่ใน SOAP เมสเสจเสมอ และจะบรรจุใจความสำคัญของ SOAP เมสเสจ ซึ่งชนิดของเมสเสจนั้นจะบรรจุส่วนของการร้องขอใช้งานเมธอดโดยใช้ชื่อของเมธอด และพารามิเตอร์ของเมธอดนั้น ซึ่งจะอยู่ในรูปแบบของ XML โดยเฉพาะรูปแบบของพารามิเตอร์ที่ส่งจะเป็นส่วนสำคัญมากสำหรับการส่งข้อมูลที่ประมวลผลที่ได้จากผู้ให้บริการกลับมาในรูปแบบ XML และถ้าหากมีข้อผิดพลาดก็จะบอกในส่วนนี้

ตัวอย่างแสดง SOAP รีเควส์ตเมสเสจ

```
<SOAP:ENV Envelope
        xmlns:SOAP-ENV = " http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Body xmlns:addNum= "Some-URI">
            < addNum:AddNumbers >
                <nNum1>10</nNum1>
                <nNum2>10</nNum2>
            < /addNum:AddNumbers>
        </SOAP-ENV:Body>
</SOAP-ENV : Envelope>
```

ตัวอย่างแสดง SOAP เรสพอนส์เมสเสจ

```
<SOAP:ENV Envelope
        xmlns:SOAP-ENV = " http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP-ENV:Body xmlns:addNum= "Some-URI">
            < addNum:AddNumberResponse >
                <return>20</return>
            < /addNum:AddNumberResponse>
        </SOAP-ENV:Body>
</SOAP-ENV : Envelope>
```

2.7.2.4 SOAP Intermediaries

SOAP อินเตอร์มีเดียรี่จะทำหน้าที่เป็นตัวผ่าน หรือตัวกลางในการส่งเอกสาร SOAP จากต้นทาง ในที่นี้คือ ผู้ขอใช้บริการไปยังผู้ให้บริการ เพื่อสร้างกระบวนการบางอย่างกับเอกสาร ซึ่งในการสร้างเส้นทางในการส่งแบบนี้อาจมีจุดประสงค์ดังต่อไปนี้คือ

1. อินเตอร์มีเดียรี่ ทำหน้าที่เป็นครอสซิ่งทรัสโดเมน (Crossing trust domain) เนื่องจากการทำงานของเว็บเซอร์วิสเป็นการเรียกใช้งานแบบระยะไกล หรือมีการทำงานแบบระบบกระจาย (Distributed System) ดังนั้นวิธีการในด้านความปลอดภัยจึงเป็นส่วนหนึ่งที่ต้องคำนึงถึง ซึ่งวิธีการหนึ่งก็คือให้มีการส่งเอกสารมายังจุดหนึ่งที่มีความน่าเชื่อถือได้ก่อน แล้วส่งต่อไปยังปลายทางอีกที่หนึ่ง ซึ่งหลักการใช้ ทรัสโดเมนเช่น ในองค์กรเราอาจจะมีเครือข่ายอินทราเน็ต อยู่ภายในองค์กร เครื่องคอมพิวเตอร์แต่ละเครื่องอาจจะสามารถติดต่อไปยังเซิร์ฟเวอร์ได้โดยตรง แต่ถ้าต้องการออกไปยังอินเตอร์เน็ต หรือเอกซ์ทราเน็ตที่อยู่นอกองค์กรจะต้องผ่านทรัสโดเมนก่อน ในที่นี้อาจจะเป็นไฟร์วอลล์ หรือทางเข้าเครือข่ายเสมือน (Virtual Private Network (VPN) Gateway) เพื่อความปลอดภัย

2. อินเตอร์มีเดียรี่ ทำหน้าที่เป็นตัวขยายการทำงานแบบระบบกระจาย ถ้าเรามองดูเฉพาะแง่ของการขอใช้บริการจาก ผู้ขอใช้บริการไปยังผู้ให้บริการ โดยผู้ให้บริการจะมีการระบวนการทำงานบางอย่างภายในและตอบกลับไปยังผู้ขอใช้ ซึ่งการทำงานแบบนี้ก็อยู่บนพื้นฐานของการร้องขอและการตอบสนองนั้นเอง แต่ถ้าระบบของเรามีการขยายใหญ่ขึ้นสิ่งที่เราอาจจะต้องดูเพิ่มขึ้นก็คือความสามารถในการรองรับงานของผู้ให้บริการด้วย ดังนั้นก่อนที่จะมีการส่งเอกสาร SOAP ไปยังเซิร์ฟเวอร์ของผู้ให้บริการนั้นอาจจะมีการผ่านอินเตอร์มีเดียรี่เซอร์เวอร์ก่อน เพื่อทำหน้าที่เป็นบัฟเฟอร์(buffer) รับเอกสาร SOAP มาพักไว้และส่งเอกสารให้ผู้ให้บริการ อีกทอดหนึ่ง การทำงานเช่นนี้จะช่วยให้เซอร์เวอร์ของผู้ให้บริการไม่รับภาระงานมากจนเกินไปในช่วงเวลาใดเวลาหนึ่งจนอาจทำให้เกิดความเสียหายได้ และอินเตอร์มีเดียรี่ อาจจะมีการรวบรวมเอกสารแล้วส่งมาให้ผู้ให้บริการเป็นลักษณะแบทช์ (Batch) เพื่อเพิ่มประสิทธิภาพการใช้งาน อินเตอร์มีเดียรี่ ลักษณะนี้ผู้ขอใช้บริการอาจจะไม่เห็นอินเตอร์มีเดียรี่ แต่จะมองเห็นว่าติดต่ออยู่กับผู้ให้บริการโดยตรง

3. การใช้งานอินเตอร์มีเดียรี่ ในลักษณะเพิ่มคุณค่า (Value-added) ให้กับระบบ เช่น ใช้อินเตอร์มีเดียรี่ ในการเข้ารหัสข้อมูล (Encrypts) และเซ็นลายเซ็นอิเล็กทรอนิคส์ (Digital signs) ในการส่งและรับ หรืออีกตัวอย่างหนึ่งเช่นต้องการเก็บข้อมูลเพิ่มนำไปวิเคราะห์การทำงานของระบบโดยการรับส่งเอกสารจะผ่านอินเตอร์มีเดียรี่ แต่ไม่ได้ทำการประมวลผลเอกสารแต่จะเก็บของข้อมูลบางอย่างไว้เท่านั้นเช่นนำไปวิเคราะห์ค่า QoS (Quality of Service) หรือคอขวด (Botton Neck) ของระบบ เป็นต้น

เอกสาร SOAP จะถูกส่งจากผู้ขอใช้บริการผ่าน ไปจนถึงปลายทางคือผู้ให้บริการ ดังนั้นเอกสาร SOAP ก็ถูกเปิดเผยทั้งหมดกับอินเตอร์มีเดียรี่ ด้วย แต่ส่วนที่ อินเตอร์มีเดียรี่ สนใจจะอยู่เฉพาะส่วนที่เป็นเฮดเดอร์เท่านั้น โดยในเฮดเดอร์ จะมีแอททริบิวท์หนึ่งเพื่อใช้ในการระบุว่าใครที่ควรจะทำงานกับเฮดเดอร์นี้ ซึ่งแอททริบิวท์ นั้นก็คือ แอกเตอร์(actor) โดยค่าที่แอกเตอร์จะต้องใส่ก็คือ URI (Universal Resource Indicator) ของอินเตอร์มีเดียรี่นั้นเอง ดังตัวอย่าง

```
<soap:Header>
    <i:Authentication>
        xmlns:i= "http://www.i3t.or.th/ws/security"
        soap:actor= "urn:I3T:SecurityGateWay"
        soap:mustUnderstand= "1">
        <i:username>ACompany</i:username>
        <i:password>SOAPpath</i:password>
    </i:Authentication>
</soap: Header>
```

ดังนั้นเมื่ออินเตอร์มีเดียรี่รับเอกสาร SOAP โดยถ้ามีชื่อแอททริบิวท์แอกเตอร์ตรงกับของตัวเอง อินเตอร์มีเดียรี่ จะเปลี่ยนเนื้อหาในเฮดเดอร์ที่ได้เพื่อส่งต่อไปยังอินเตอร์มีเดียรี่อื่นหรือไปยังผู้ให้บริการจุดประสงค์หลักที่มีการออกแบบการทำงานแบบนี้ เพื่อลดความซับซ้อนของเอกสารลง นอกจากนี้แล้วยังมี URI ของแอททริบิวท์แอกเตอร์ ชนิดพิเศษอีกหนึ่งค่าคือ "http://schemas.xmlsoap.org/soap/actor/next" เพื่อบอกว่าผู้รับเอกสาร SOAP ในจุดต่อไปจะเป็นผู้ที่จะประมวลผลของเอกสาร ซึ่งกระบวนการนี้จะมีประโยชน์อย่างมากในการส่งเอกสาร SOAP เป็นทอดๆ (hop-by-hop)

2.7.3 WSDL (Web Services Description Language)

WSDL นั้นเป็นภาษาที่ใช้ XML เป็นพื้นฐาน ซึ่งใช้ในการบรรยายเว็บเซอร์วิสหรือเน็ตเวิร์คเอ็นพอยต์ (network endpoint) เพื่อเป็นการเผยแพร่ข้อมูลการให้บริการแก่ระบบภายนอกโดยที่ทำงานผ่านระบบเครือข่าย โดยปราศจากการแทรกแซงของคน WSDLนั้นยังสามารถที่จะบรรยายการส่งเมสเสจระหว่างเว็บเซอร์วิส ระบุตำแหน่งที่อยู่ของเว็บเซอร์วิสและรวมทั้งโปรโตคอลที่ใช้ในการติดต่อสื่อสารกันของเว็บเซอร์วิส WSDL นั้นจะทำงานรวมกันกับ SOAP และ UDDI เพื่อที่จะทำให้เว็บเซอร์วิสติดต่อกับเว็บเซอร์วิส อื่นๆ ได้บนระบบอินเตอร์เน็ต ถ้าไม่มี WSDL แล้วการกระทำเมสเสจจิ่งอินเตอร์เฟส (messaging interface) นั้นจะต้องทำเอง WSDL คือ มาตรฐานสำหรับการประกาศโปรเซสที่จำเป็นในการเรียกใช้เซอร์วิส SOAP

โครงสร้างเอกสาร WSDL

Element	Definition
<portType>	เป็นส่วนที่สำคัญที่สุดในWSDL อีลีเมนต์ อธิบายโอเปอเรชั่นที่เว็บเซอร์วิส มีให้บริการและเมสเสจที่เกี่ยวข้องเทียบได้กับฟังก์ชันไลบรารี่ หรือ โมดูล(module) หรือคลาสในการเขียนโปรแกรม

<operation>	อธิบายเมธอดที่ให้บริการ เว็บเซอร์วิสหนึ่งจะมีเมธอดจำนวนกี่เมธอดก็ได้
<message>	อธิบายดาต้าอีลีเมนต์ (data elements) ของโอเปอเรชั่นแต่ละเมสเสจ อาจมีมากกว่าหนึ่งส่วนเทียบได้กับพารามิเตอร์ของฟังก์ชันในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลที่เว็บเซอร์วิสใช้ เพื่อความเป็นกลาง WSDL ใช้ XML สคีมาซินแทกซ์ (Schema syntax) ในการระบุชนิดข้อมูล
<binding>	อธิบายรูปแบบของเมสเสจและโปรโตคอล รายละเอียดในแต่ละพอร์ต(port)
<service>	สำหรับเว็บเซอร์เวอร์จะมีเว็บเซอร์วิสจำนวนกี่บริการก็ได้ และ ชื่อเว็บเซอร์วิส ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน

ตารางที่ 2.1 แสดงโครงสร้างเอกสาร WSDL

ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการเว็บเซอร์วิสได้มากกว่า 1 บริการ โดยแต่ละเว็บเซอร์วิสจะมีพอร์ตสื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

2.7.4 UDDI

UDDI (Universal Description, Discovery, and Integration) เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI ใช้สำหรับค้นหาบริการ ที่ต้องการและเมื่อได้มาแล้ว UDDI ยังจัดหาข้อตกลงในวิธีการที่จะใช้งาน รูปแบบการทำงานของUDDI จะเปลี่ยนแปลงวิธีการค้นหาข้อมูลบนอินเตอร์เน็ตที่เราใช้อยู่ในปัจจุบันจากการพิมพ์ชื่อ โดเมนเนมในช่องแอดเดรสบาร์ ไปเป็นการใช้คำสามัญหรือชื่อเฉพาะในการค้นหา และเข้าสู่เว็บไซต์แทน UDDI จะทำหน้าที่เหมือนสมุดหน้าเหลืองที่จำแนกผู้จดทะเบียนตามรูปของธุรกิจแยก ประเภทตามกิจการ

โดยสรุป UDDI เสมือนคือโครงสร้างพื้นฐานที่มารองรับการค้นหาข้อมูลแบบใหม่ โดยใช้ตัวเลขหรือคำสามัญแทนโดเมนเนม สำหรับโดเมนเนมเองก็ยังคงอยู่ไม่ได้หายไปไหน เพียงแต่UDDI จะมาเป็นอีกมาตรฐานหนึ่งในการติดต่อกับไอพีแอดเดรส(IP Address) เหมือนที่เราเคยมีโดเมนเนมหรือยูอาร์แอลในการทำให้ผู้ที่เข้าสู่เว็บไซต์รู้จักและจดจำได้โดยง่าย

บทที่ 3
ความปลอดภัยในการส่งผ่านเครือข่าย

3.1 นิยามของความมั่นคงปลอดภัยในคอมพิวเตอร์

ในปัจจุบันระบบคอมพิวเตอร์ได้ถูกคุกคามมากขึ้นทั้งจากไวรัสคอมพิวเตอร์หรือจากผู้ไม่ดี ซึ่งความมั่นคงปลอดภัยคอมพิวเตอร์ (Computer Security) ช่วยปกป้องเครื่องคอมพิวเตอร์รวมไปถึงอุปกรณ์ต่างๆที่เกี่ยวข้อง และที่สำคัญยังสามารถช่วยปกป้องข้อมูลที่ได้จัดเก็บไว้ภายในระบบหรือใช้ในความปลอดภัยทางข้อมูลสารสนเทศ (Information Security) ก็ได้

จุดประสงค์หลักของความปลอดภัยทางข้อมูลคือ

- การรักษาความลับ (Confidentiality) คือการรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับ และผู้มีสิทธิเท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้

- การรักษาความสมบูรณ์ (Integrity) คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไม่ว่าจะเป็นโดย อุบัติเหตุหรือโดยเจตนา

- ความพร้อมใช้ (Availability) คือการรับรองว่าข้อมูลและบริการการสื่อสารต่าง ๆ พร้อมที่จะใช้ได้ในเวลาที่ต้องการใช้งาน

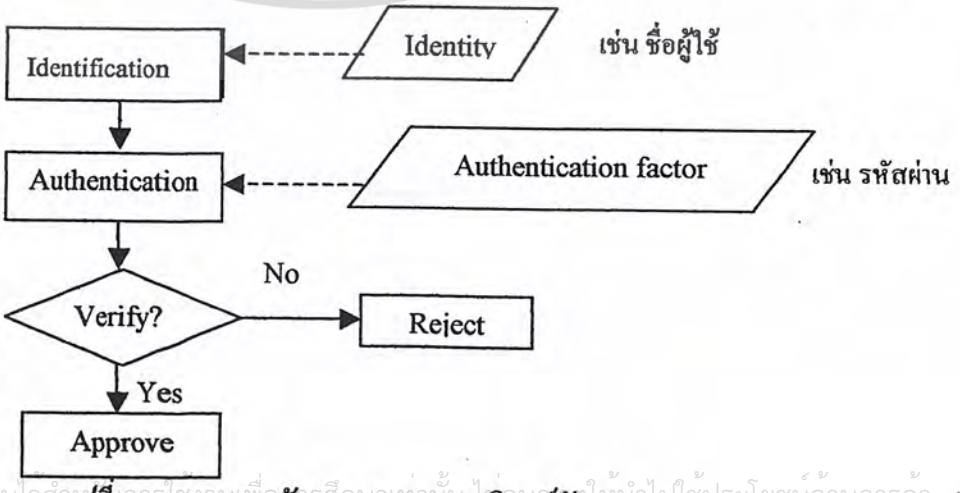
- การห้ามปฏิเสธความรับผิดชอบ (Non-Repudiation) คือวิธีการสื่อสารซึ่งผู้ส่งข้อมูลได้รับหลักฐานว่าได้มีการส่งข้อมูลแล้วและผู้รับก็ได้รับการยืนยันว่าผู้ส่งเป็นใคร ดังนั้นทั้งผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่มีความเกี่ยวข้องกับข้อมูลดังกล่าวในภายหลัง

3.1.1 การพิสูจน์ตัวตน (Authentication)

การพิสูจน์ตัวตน คือขั้นตอนการยืนยันความถูกต้องของหลักฐาน(Identity) ที่แสดงว่าเป็นบุคคลที่กล่าวอ้างจริง ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

1) การระบุตัวตน (Identification) คือขั้นตอนที่ผู้ใช้แสดงหลักฐานว่าตนเองคือใคร เช่น ชื่อผู้ใช้

2) การพิสูจน์ตัวตน (Authentication) คือขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลที่กล่าวอ้างจริง



รูปที่ 3.1 แสดงแผนผังกระบวนการการพิสูจน์ตัวตน

จากแผนผังแสดงกระบวนการพิสูจน์ตัวตน ในขั้นแรกผู้ใช้จะทำการแสดงหลักฐานที่ใช้ในการพิสูจน์ตัวตนต่อระบบ ซึ่งในขั้นนี้คือการระบุตัวตน และในขั้นตอนต่อมาระบบจะทำการตรวจสอบหลักฐานที่ผู้ใช้นำมากล่าวอ้างซึ่งก็คือการพิสูจน์ตัวตน หลังจากระบบได้ทำการตรวจสอบหลักฐานเรียบร้อยแล้วถ้าหลักฐานที่นำมากล่าวอ้างถูกต้องจึงอนุญาตให้เข้าสู่ระบบได้ หากหลักฐานที่นำมากล่าวอ้างไม่ถูกต้องผู้ใช้จะถูกปฏิเสธจากระบบหลักฐานที่ผู้ใช้นำมากล่าวอ้างที่เกี่ยวกับเรื่องของความปลอดภัยนั้นจำแนกได้ 2 ชนิด

1) ระบุแท้จริง (Actual identity) คือหลักฐานที่สามารถบ่งบอกได้ว่าในความเป็นจริงบุคคลที่กล่าวอ้างนั้นเป็นใคร เช่น ลายนิ้วมือ

2) ระบุทางอิเล็กทรอนิกส์ (Electronic identity) คือหลักฐานทางอิเล็กทรอนิกส์ซึ่งสามารถบ่งบอกข้อมูลของบุคคลนั้นได้ แต่ละบุคคลอาจมีหลักฐานทางอิเล็กทรอนิกส์ได้มากกว่าหนึ่งหลักฐาน ตัวอย่างเช่น บัญชีชื่อผู้ใช้

กลไกของการพิสูจน์ตัวตน (Authentication mechanisms) สามารถแบ่งออกได้เป็น 3 คุณลักษณะคือ

1) สิ่งที่คุณมี (Possession factor) เช่น กุญแจหรือเครดิตการ์ด เป็นต้น

2) สิ่งที่คุณรู้ (Knowledge factor) เช่น รหัสผ่านหรือการใช้พิน (PINs) เป็นต้น

3) สิ่งที่คุณเป็น (Biometric factor) เช่น ลายนิ้วมือ รูปแบบเรตินา (retinal patterns) หรือใช้รูปแบบเสียง (voice patterns) เป็นต้น

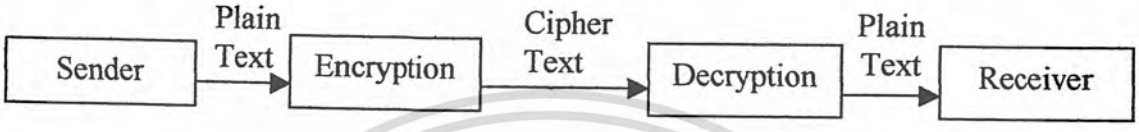
กระบวนการพิสูจน์ตัวตนนั้นจะนำ 3 ลักษณะข้างต้นมาใช้ในการยืนยันหลักฐานที่นำมากล่าวอ้าง ทั้งนี้ขึ้นอยู่กับระบบ วิธีการที่นำมาใช้เพียงลักษณะอย่างใดอย่างหนึ่ง(Single-factor authentication) นั้นมีข้อจำกัดในการใช้ ตัวอย่างเช่น สิ่งที่คุณมีนั้นอาจจะสูญหายหรือถูกขโมยได้ สิ่งที่คุณรู้ อาจจะถูกดักฟังเดา หรือถูกขโมยจากเครื่องคอมพิวเตอร์ สิ่งที่คุณเป็นจัดได้ว่าเป็นวิธีที่มีความปลอดภัยสูง อย่างไรก็ตามการที่จะใช้เทคโนโลยีนี้ได้นั้นจำเป็นต้องมีการลงทุนที่สูง เป็นต้น ดังนั้นจึงได้มีการนำแต่ละคุณลักษณะมาใช้ร่วมกัน (multi-factor authentication) ตัวอย่างเช่น ใช้สิ่งที่คุณมีกับสิ่งที่คุณรู้มาใช้ร่วมกัน เช่น การใช้ลายมือชื่อร่วมกับการใช้บัตรเครดิตหรือการใช้รหัสผ่านร่วมกับการใช้บัตร ATM เป็นต้น การนำแต่ละลักษณะของการพิสูจน์ตัวตนมาใช้ร่วมกันมากกว่าหนึ่งลักษณะจะช่วยเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูล

3.1.2 การกำหนดสิทธิ์ (Authorization)

การกำหนดสิทธิ์ (Authorization) คือขั้นตอนในการอนุญาตให้แต่ละบุคคลสามารถเข้าถึงข้อมูลหรือระบบใดได้บ้าง ก่อนอื่นต้องทราบก่อนว่าบุคคลที่กล่าวอ้างนั้นคือใครตามขั้นตอนการพิสูจน์ตัวตนและต้องให้แน่ใจด้วยว่าการพิสูจน์ตัวตนนั้นถูกต้อง

3.1.3 การเข้ารหัส (Encryption)

การเข้ารหัส คือ กระบวนการในการเปลี่ยนข้อมูลต้นฉบับให้อยู่ในอีกรูปแบบหนึ่งที่ไม่สามารถเข้าใจได้โดยง่าย ผลลัพธ์ที่ได้จากการเข้ารหัสจะสามารถกลับไปเป็นข้อมูลต้นฉบับได้นั้นต้องใช้การถอดรหัส (Decryption) โดยเราจะเรียกข้อมูลต้นฉบับที่จะทำการเข้ารหัสว่า เคลียร์เท็กซ์ (Clear text) หรือ เพลนเท็กซ์ (Plain text) และเราจะเรียกข้อมูลที่ทำการเข้ารหัสเรียบร้อยแล้วว่า ไซเฟอร์เท็กซ์ (Cipher text) การเข้ารหัสและถอดรหัสสามารถเขียนไดอะแกรมแสดงได้ดังนี้



รูปที่ 3.2 แสดงการเข้ารหัสและถอดรหัส

หลักการของการเข้ารหัสมี 2 ประเภท คือ

1) การแทนที่ (Substitution) เป็นการแทนที่บิตใดๆ ด้วยข้อมูลอื่น ทำให้ข้อมูลมีความสับสน ยากต่อการถอดรหัส เช่น เรามีข้อความว่า PRIVATE แล้วใช้หลักการเพิ่มค่ารหัสแอสกี (ASCII) ของตัวอักษรแต่ละตัวในข้อความไปอีก 3 แล้วแทนที่ในข้อความต้นฉบับจะได้ข้อความออกมาเป็น SULYDWH เป็นต้น

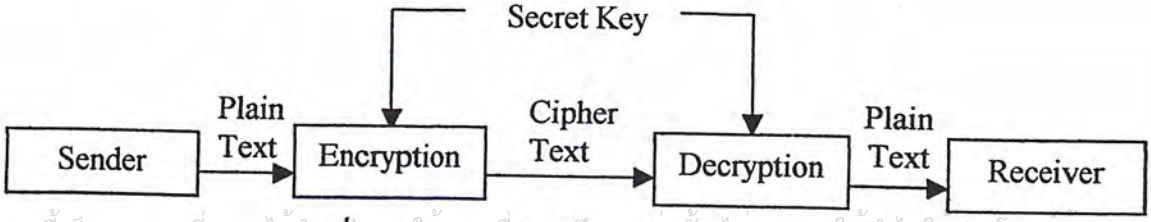
2) การสับเปลี่ยนตำแหน่ง (Permutation) เป็นการสับเปลี่ยนตำแหน่งใดๆ ของข้อมูล เมื่อมีการสับเปลี่ยนตำแหน่งมากๆ ทำให้ข้อมูลมีความซับซ้อนยากต่อการถอดรหัส เช่น เรามีข้อความว่า PRIVATE จะได้ข้อความที่จากการเข้ารหัสเป็น VITREAP เป็นต้น

การเข้ารหัสนั้นวิธีการหนึ่งที่ทำได้คือการเข้ารหัสและถอดรหัสโดยใช้กุญแจ มี 2 แบบ

1) การเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem)
2) การเข้ารหัสแบบไม่สมมาตร (Aymmetric Cryptosystem)

3.1.3.1 การเข้ารหัสแบบสมมาตร (Symmetric Cryptosystem)

การเข้ารหัสแบบสมมาตรทำงานโดยใช้กุญแจในการเข้ารหัสจะใช้กุญแจเข้าร่วมกับอัลกอริทึมในการเข้ารหัสและเมื่อทำการถอดรหัสจะใช้กุญแจเดียวกับที่ใช้ในการเข้ารหัสร่วมกับอัลกอริทึมแบบเดียวกันแสดงการทำงานดังรูปที่ 3.3



รูปที่ 3.3 แสดงการเข้ารหัสและถอดรหัสแบบสมมาตร

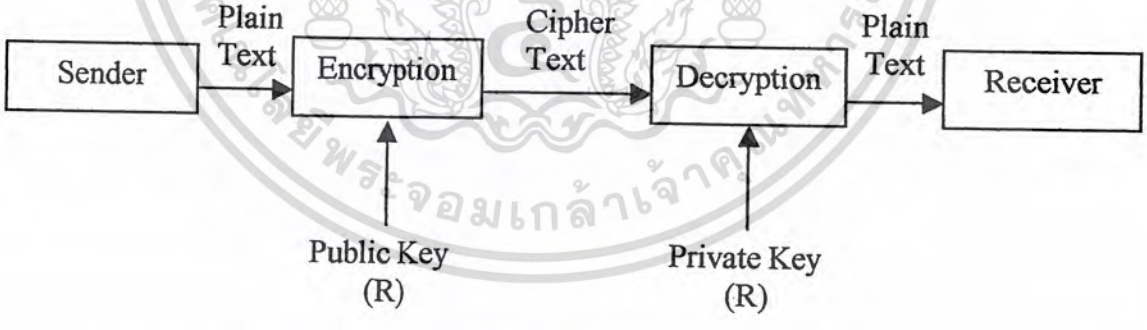
ลักษณะของการเข้ารหัสแบบสมมาตร

1. จะต้องมีอัลกอริทึมการเข้ารหัสที่แข็งแกร่งเพียงพอ กล่าวคือไม่ว่าผู้ใดรู้ถึงอัลกอริทึมการเข้ารหัสจะต้องไม่สามารถหาเพลนเท็กซ์และกุญแจลับจากไซเฟอร์เท็กซ์ ทำให้การส่งข้อมูลจะต้องมีเฉพาะผู้รับและผู้ส่งเท่านั้นที่รู้กุญแจลับ ถ้าเมื่อใดที่ผู้อื่นรู้กุญแจลับและรู้อัลกอริทึมที่ใช้ในการถอดรหัสก็จะสามารถหาเพลนเท็กซ์ได้ การแปลงเพลนเท็กซ์ไปเป็นไซเฟอร์เท็กซ์ ใช้การกระทำ 2 รูปแบบคือ ทั้งการแทนที่และการเปลี่ยนตำแหน่ง
2. การเข้ารหัสและการถอดรหัสใช้กุญแจเดียวกัน
3. การเข้ารหัสสามารถใช้บล็อกไซเฟอร์ในการเข้ารหัสได้

3.1.3.2 การเข้ารหัสแบบไม่สมมาตร (Asymmetric Cryptosystem)

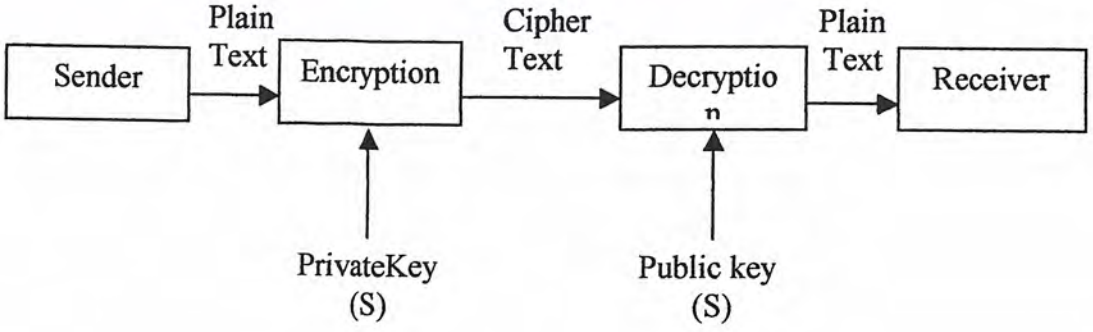
วิธีการนี้เป็นการเข้ารหัสและการถอดรหัสที่ใช้กุญแจคนละดอกกัน โดยกุญแจทั้งสองต้องเป็นกุญแจคู่ (Key Pair) กัน คือ คีย์สาธารณะ (Public Key) ซึ่งเป็นคีย์ที่ทำการแจกให้ผู้อื่น และคีย์ส่วนตัว (Private Key) เป็นคีย์ที่เก็บไว้เป็นความลับ ความสามารถของการเข้ารหัสแบบไม่สมมาตรมีอยู่ด้วยกัน 3 ข้อคือ

1) ความลับ หมายถึง ไม่ยอมให้มีบุคคลที่ไม่มีสิทธิ์เข้ามาดูข้อมูลได้ ซึ่งสามารถทำได้โดยผู้ส่งเข้ารหัสโดยใช้กุญแจสาธารณะของผู้รับ ซึ่งทำให้มีแต่ผู้รับที่มีกุญแจส่วนตัวที่เป็นกุญแจคู่ของมันเท่านั้นที่สามารถทำการถอดรหัสได้ ถึงแม้กุญแจสาธารณะมีบุคคลอื่นรู้ก็ไม่สามารถทำการถอดรหัสได้ดังรูป



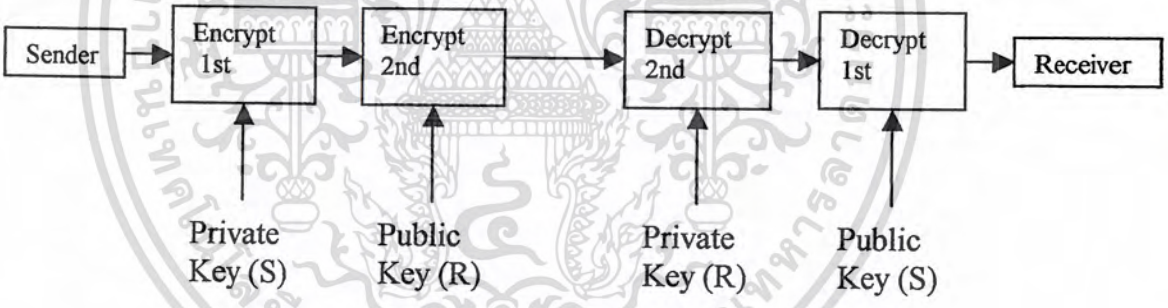
รูปที่ 3.4 แสดงการเข้ารหัสแบบไม่สมมาตรแบบความลับ

2) การพิสูจน์บุคคล หมายถึง การตรวจสอบที่มาของข้อมูล ว่าถูกส่งมาจากผู้ส่งคนนั้นจริงหรือไม่ ซึ่งทำโดยเข้ารหัสข้อมูลโดยใช้กุญแจส่วนตัวของผู้ส่ง การตรวจสอบทำได้โดยใช้กุญแจสาธารณะที่เป็นกุญแจคู่ทำการถอดรหัส ซึ่งผู้ที่จะสามารถเข้ารหัสได้นั้นต้องเป็นผู้ที่เป็นเจ้าของกุญแจส่วนตัวไปให้ผู้รับ โดยผู้รับนำเอาข้อมูลมาทำการถอดรหัสแล้วตรวจดูว่าข้อมูลส่วนตัวของผู้ส่งเป็นจริงหรือไม่โดยการเข้ารหัสเพื่อพิสูจน์ตัวบุคคลแสดงดังรูปที่ 3.5



รูปที่ 3.5 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคล

3) การพิสูจน์บุคคลและความลับ (Authenticity and Secrecy) หมายถึง การตรวจสอบที่มาของข้อมูลและจำกัดสิทธิ์ข้อมูลให้ผู้รับที่สามารถอ่านข้อมูลได้เท่านั้น สามารถทำได้โดยผู้ส่งทำการนำข้อมูลมาเข้ารหัสครั้งแรกด้วยกุญแจส่วนตัวของตนเพื่อเป็นการพิสูจน์ตัวบุคคลจากนั้นนำมาเข้ารหัสโดยกุญแจ เมื่อผู้รับได้รับไซเฟอร์เท็กมาก็นำมาถอดรหัสโดยใช้กุญแจส่วนตัวของตัวเองเปิด จากนั้นนำมาถอดรหัส โดยใช้กุญแจสาธารณะที่เป็นกุญแจคู่ของผู้ส่ง ดังแสดงดังรูปที่ 3.6



รูปที่ 3.6 แสดงการเข้ารหัสแบบไม่สมมาตรแบบพิสูจน์บุคคลและความลับ

3.1.4 การรักษาความสมบูรณ์ (Integrity)

การรักษาความสมบูรณ์ คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไปจากต้นฉบับไม่ว่าจะเป็นโดยบังเอิญหรือดัดแปลงโดยเจตนาที่อาจส่งผลเสียต่อข้อมูลการคุกคามความสมบูรณ์ของข้อมูลคือการที่บุคคลที่ไม่ได้รับอนุญาตสามารถที่จะเข้าควบคุมการจัดการของข้อมูลได้

3.1.5 การตรวจสอบ (Audit)

การตรวจสอบ คือการตรวจสอบหลักฐานทางอิเล็กทรอนิกส์ ซึ่งสามารถใช้ในการติดตามการดำเนินการเพื่อตรวจสอบความถูกต้องและแม่นยำ ตัวอย่างเช่นการตรวจสอบบัญชีชื่อผู้ใช้ โดยผู้ตรวจบัญชี ซึ่งการตรวจสอบความถูกต้องของการดำเนินการเพื่อให้แน่ใจว่าหลักฐานทางอิเล็กทรอนิกส์นั้นได้ถูกสร้างและสั่งให้ทำงานโดยบุคคลที่ได้รับอนุญาต และในการเชื่อมต่อเหตุการณ์เข้ากับบุคคลจะต้องทำ

การตรวจสอบหลักฐานของบุคคลนั้นด้วย ซึ่งถือเป็นหลักการพื้นฐานของขั้นตอนการทำงานของการพิสูจน์ตัวตนด้วยการพิสูจน์ตัวตนจัดเป็นการตรวจสอบหลักฐานขั้นพื้นฐานที่สำคัญที่สุดใน 5 ระดับชั้นของการควบคุมความปลอดภัย ดังนั้นการพิสูจน์ตัวตนดีจะช่วยเพิ่มความมั่นคงปลอดภัยขั้นพื้นฐานให้กับระบบมากยิ่งขึ้น

3.2 ประเภทของการพิสูจน์ตัวตน

3.2.1 ไม่มีการพิสูจน์ตัวตน (No Authentication)

ตามหลักการแล้วการพิสูจน์ตัวตนไม่มีความจำเป็น ถ้าเงื่อนไขต่อไปนี้เป็นจริง

- ข้อมูลเหล่านั้นเป็นข้อมูลสาธารณะ ที่อนุญาตให้ทุกคนเข้าใช้บริการและเปลี่ยนแปลงได้ หรือ
- ข้อมูลข่าวสารหรือแหล่งของข้อมูลนั้นๆ สามารถเข้าถึงได้เฉพาะบุคคลที่ได้รับอนุญาตเท่านั้น

3.2.2 การพิสูจน์ตัวตนโดยใช้รหัสผ่าน (Authentication by Passwords)

รหัสผ่านเป็นวิธีการที่ใช้มานานและนิยมใช้กันแพร่หลาย รหัสผ่านควรจำกัดให้เฉพาะผู้ใช้ที่มีสิทธิเท่านั้นที่ทราบแต่ว่าในปัจจุบันนี้ การใช้แค่รหัสผ่านไม่มีประสิทธิภาพมากพอที่จะรักษาความมั่นคงปลอดภัยให้กับระบบคอมพิวเตอร์หรือระบบเครือข่ายคอมพิวเตอร์ เนื่องจากการตั้งรหัสผ่านที่ง่ายเกินไปและวิทยาการและความรู้ที่ก้าวหน้าทำให้รหัสผ่านอาจจะถูกขโมยโดยระหว่างการสื่อสารผ่านเครือข่ายได้

3.2.3 การพิสูจน์ตัวตนโดยใช้ PIN (Authentication by PIN)

PIN (Personal Identification Number) เป็นรหัสลับส่วนบุคคลที่ใช้เป็นรหัสผ่านเพื่อเข้าสู่ระบบ ซึ่ง PIN ใช้อย่างแพร่หลายโดยเฉพาะการทำธุรกรรมทางด้านธนาคารเช่นบัตร ATM และเครดิตการ์ดต่างๆ การใช้ PIN ทำให้มีความปลอดภัยในการสื่อสารข้ามระบบเครือข่ายสาธารณะมากขึ้น เนื่องจาก PIN จะถูกเข้ารหัสเอาไว้และจำเป็นต้องมีเครื่องมือที่สามารถถอดรหัสนี้ออกมาได้ เช่นฮาร์ดแวร์ที่ออกแบบมาโดยเฉพาะ และถูกติดตั้งไว้ในเครื่องของผู้รับและผู้ส่งเท่านั้น

3.2.4 การพิสูจน์ตัวตนโดยใช้ Password Authenticators หรือ Tokens (Authentication by Password Authenticators or Tokens)

Authenticator หรือ โทเคน(Token) เป็นฮาร์ดแวร์พิเศษที่ใช้สร้าง "รหัสผ่านซึ่งเปลี่ยนแปลงได้ (dynamic password)" ในขณะที่กำลังเข้าสู่ระบบเครือข่ายมี 2 วิธี
คือ ซิงโครนัส และ อะซิงโครนัส

3.2.4.1 การพิสูจน์ตัวตนแบบซิงโครนัส

การพิสูจน์ตัวตนแบบซิงโครนัส แบ่งออกเป็น 2 ประเภทตามลักษณะของการใช้งาน คือ

1) การพิสูจน์ตัวตนแบบซิงโครนัสโดยขึ้นอยู่กับสถานการณ์ (Event-synchronous authentication)

เมื่อผู้ใช้ต้องการที่จะเข้าสู่ระบบ ผู้ใช้จะต้องกดโทเคน เพื่อให้โทเคนสร้างรหัสผ่านให้ จากนั้นผู้ใช้นำรหัสผ่านที่แสดงหลังจากกดโทเคน ใส่ลงในฟอร์ม เพื่อเข้าสู่ระบบ ระบบจะทำการตรวจสอบกับเซิร์ฟเวอร์ก่อน ว่ารหัสผ่านที่ใส่มีอยู่ในเซิร์ฟเวอร์จริง จึงจะยินยอมให้ผู้ใช้เข้าสู่ระบบ

2) การพิสูจน์ตัวตนแบบซิงโครนัสโดยขึ้นอยู่กับเวลา (Time-synchronous authentication)

เป็นวิธีการที่สร้างรหัสผ่านโดยมีการกำหนดช่วงระยะเวลาการใช้งานโดยปกติแล้วรหัสผ่านจะถูกเปลี่ยนทุกๆ หนึ่งนาที การสร้างรหัสผ่านจะเป็นไปอย่างต่อเนื่อง ทำให้บางครั้งรหัสผ่านที่สร้างออกมาอาจจะซ้ำกันกับรหัสผ่านตัวอื่นที่เคยสร้างมาแล้วก็ได้เมื่อผู้ใช้ ต้องการเข้าสู่ระบบก็ใส่รหัสผ่านและเวลาที่รหัสผ่านตัวนั้นถูกสร้างขึ้นมา (รหัสผ่านจะถูกสร้างขึ้นมาจากโทเคน) ลงในฟอร์มเพื่อเข้าสู่ระบบ ระบบจะทำการตรวจสอบเวลาและรหัสผ่านที่ผู้ใช้ใส่ลงไป กับเซิร์ฟเวอร์ว่ารหัสผ่านที่ใส่ตรงกับเวลาที่โทเคนสร้างและมีอยู่ในเซิร์ฟเวอร์จริง จึงยินยอมให้ผู้ใช้เข้าสู่ระบบ

3.2.4.2 การพิสูจน์ตัวตนแบบอะซิงโครนัส

การพิสูจน์ตัวตนแบบอะซิงโครนัส หรือเรียกอีกอย่างหนึ่งว่า "challenge-response" ถูกพัฒนาขึ้นเป็นลำดับแรกๆ ของระบบการใช้ "รหัสผ่านซึ่งเปลี่ยนแปลงได้" ซึ่งถือได้ว่าเป็นการป้องกันการจู่โจมที่ปลอดภัยที่สุด เพราะเนื่องจากว่าเมื่อผู้ใช้ต้องการจะเข้าสู่ระบบ ผู้ใช้จะต้องทำการร้องของไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์ก็จะส่ง challenge string มาให้ผู้ใช้ เพื่อให้ผู้ใช้ใส่ลงในโทเคนที่ผู้ใช้ถืออยู่ จากนั้นโทเคน จะทำการคำนวณรหัสผ่านออกมาให้ผู้ใช้ ผู้ใช้จึงสามารถนำรหัสผ่านนั้นใส่ลงในฟอร์มเพื่อเข้าสู่ระบบได้

การพิสูจน์ตัวตนแบบซิงโครนัสทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะมีรหัสผ่านเก็บเอาไว้ แต่แบบอะซิงโครนัส ไคลเอ็นต์จะต้องติดต่อเซิร์ฟเวอร์ก่อน ก่อนจะได้รับรหัสผ่านจริง ทำให้การพิสูจน์ตัวตนแบบอะซิงโครนัสมีขั้นตอนที่ซับซ้อนกว่าแบบซิงโคนัส

3.2.5 การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล (Authentication by Biometric traits)

ลักษณะทางชีวภาพของแต่ละบุคคลเป็นลักษณะเฉพาะและลอกเลียนแบบกันไม่ได้ การนำมาใช้ในการพิสูจน์ตัวตนจะเพิ่มความน่าเชื่อถือได้มากขึ้นเช่นการใช้ลายนิ้วมือ เสียง ม่านตา เป็นต้น จึงมีการนำเทคโนโลยีนี้มาช่วยในการพิสูจน์ตัวตน เพื่อเพิ่มความปลอดภัยก่อนเข้าสู่ระบบ เช่นการใช้ควบคู่กับการใช้รหัสผ่าน

3.2.6 การพิสูจน์ตัวตนโดยใช้รหัสผ่านที่ใช้เพียงครั้งเดียว (One-Time Password: OTP)

OTP ถูกพัฒนาขึ้นเพื่อหลีกเลี่ยงปัญหาที่เกิดจากการใช้รหัสผ่านเพียงตัวเดียวซ้ำๆกัน OTP จะทำให้ระบบมีความปลอดภัยมากขึ้น เพราะรหัสผ่านจะถูกเปลี่ยนทุกครั้งก่อนที่ผู้ใช้จะเข้าสู่ระบบ การทำงานของ OTP คือเมื่อผู้ใช้ต้องการจะเข้าใช้ระบบ ผู้ใช้จะทำการร้องขอไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะส่ง

challenge string กลับมาให้ผู้ใช้ จากนั้นผู้ใช้จะนำ challenge string และรหัสลับที่มีอยู่กับตัวของผู้ใช้นำไปเข้าแฮชฟังก์ชันแล้วออกมาเป็นค่า response ผู้ใช้ก็จะส่งค่านั้นกลับไปยังเซิร์ฟเวอร์ แล้วเซิร์ฟเวอร์จะทำการตรวจสอบค่าที่ผู้ใช้ส่งมาเปรียบเทียบกับค่าที่เซิร์ฟเวอร์เองคำนวณได้ โดยเซิร์ฟเวอร์ก็ใช้วิธีคำนวณเดียวกันกับผู้ใช้ เมื่อได้ค่าที่ตรงกันเซิร์ฟเวอร์ก็จะยอมรับให้ผู้ใช้เข้าสู่ระบบ

3.2.7 การพิสูจน์ตัวตนโดยการเข้ารหัสโดยใช้กุญแจสาธารณะ (Public-key cryptography)

เป็นการรักษาความปลอดภัยของข้อมูลระหว่างการส่งข้ามเครือข่ายวิธีหนึ่งที่นิยมใช้กันอยู่ในปัจจุบัน การเข้ารหัสแบบคู่รหัสกุญแจนี้จะมีความปลอดภัยมากกว่าการเข้ารหัสข้อมูลแบบธรรมดา แต่ก็ไม่ได้หมายความว่าการเข้ารหัสแบบคู่รหัสกุญแจนี้จะเป็นวิธีที่เหมาะสมที่สุดของวิธีการเข้ารหัส ทั้งนี้ขึ้นอยู่กับประเภทงานของแต่ละองค์กรหรือบุคคล

การเข้ารหัสโดยใช้กุญแจสาธารณะ ประกอบไปด้วยกุญแจ 2 ชนิด ที่ต้องใช้คู่กันเสมอในการเข้ารหัสและถอดรหัสคือ

- กุญแจสาธารณะ (public key) เป็นกุญแจที่ผู้สร้างจะส่งออกไปให้ผู้ใช้อื่นๆ ทราบหรือเปิดเผยได้
- กุญแจส่วนตัว (private key) เป็นกุญแจที่ผู้สร้างจะเก็บไว้ โดยไม่เปิดเผยให้คนอื่นรู้

กระบวนการของการเข้ารหัสแบบคู่รหัสกุญแจ มีดังนี้

1. ผู้ใช้แต่ละคนจะสร้างคู่รหัสกุญแจของตัวเองขึ้นมา เพื่อใช้สำหรับการเข้ารหัสและการถอดรหัส
2. กุญแจสาธารณะจะถูกส่งออกไปยังผู้ใช้คนอื่นๆ แต่กุญแจส่วนตัวจะถูกเก็บที่ตนเอง
3. เมื่อจะส่งข้อมูลออกไปหาผู้ใช้คนใด ข้อมูลที่ส่งจะถูกเข้ารหัสด้วยกุญแจสาธารณะ ก่อนถูกส่งออกไป
4. เมื่อผู้รับได้รับข้อความแล้วจะใช้กุญแจส่วนตัวซึ่งเป็นคู่รหัสกันถอดรหัสออกมา

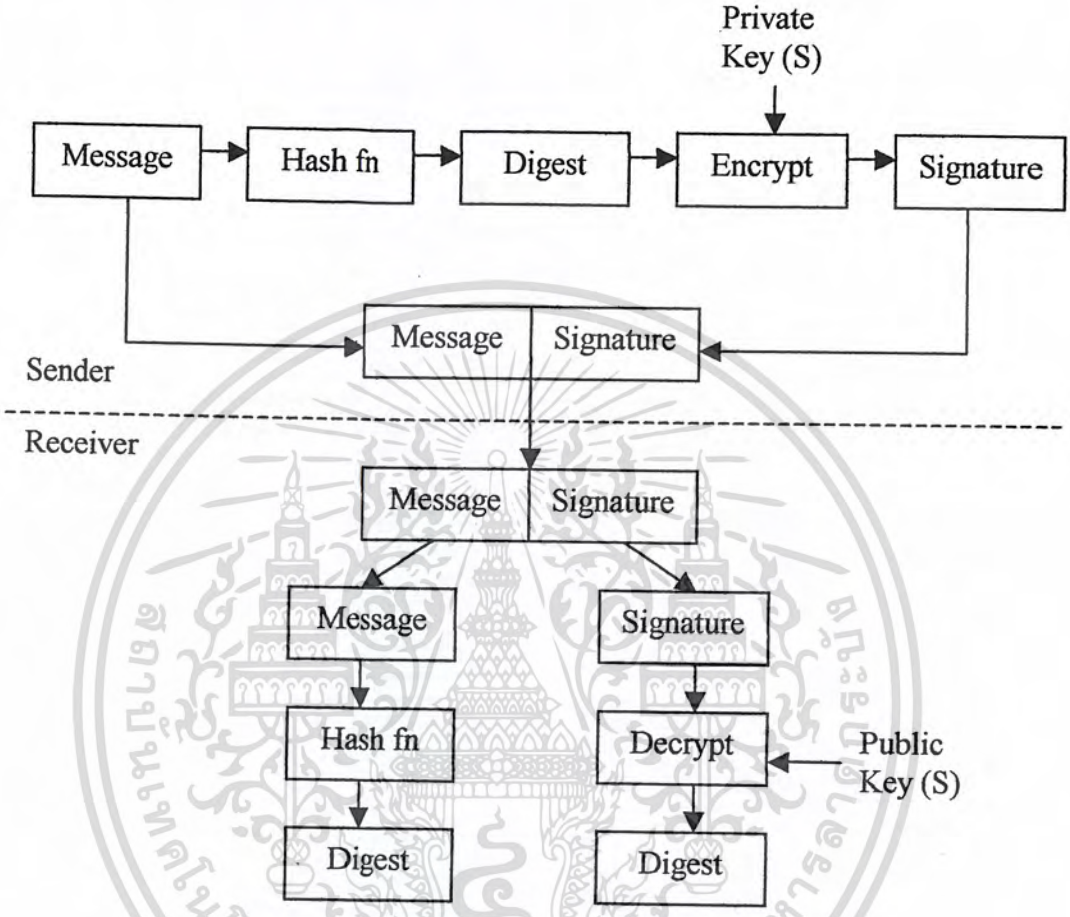
การเข้ารหัสโดยใช้กุญแจสาธารณะสามารถใช้ได้ทั้งในการเข้ารหัส และการพิสูจน์ตัวตน การประยุกต์ใช้ในการเข้ารหัสข้อมูล เป็นการนำข้อมูลที่จะส่งไปยังผู้รับมาเข้ารหัสด้วยกุญแจสาธารณะของผู้รับ และเมื่อผู้รับได้รับข้อความนั้นแล้ว จะถอดรหัสออกมาด้วยกุญแจส่วนตัว จึงจะเห็นได้ว่ามีเพียงผู้รับเท่านั้นที่จะสามารถถอดรหัสออกมาได้ การประยุกต์ใช้ในการพิสูจน์ตัวตน เป็นการนำข้อมูลที่ผู้ส่งต้องการส่งมาเข้ารหัสด้วยกุญแจส่วนตัวของผู้ส่ง แล้วนำข้อมูลนั้นส่งไปยังผู้รับ ซึ่งผู้รับจะใช้กุญแจสาธารณะซึ่งเป็นคู่รหัสกันถอดรหัสออกมา ผู้รับก็สามารถรู้ได้ว่าข้อความนั้นถูกส่งมาจากผู้ส่งคนนั้นจริง ถ้าสามารถถอดรหัสข้อมูลได้อย่างถูกต้อง

3.2.8 การพิสูจน์ตัวตนโดยการใช้ลายเซ็นอิเล็กทรอนิคส์ (Digital Signature)

เป็นการนำหลักการของการทำงานของระบบการเข้ารหัสแบบใช้คู่รหัสกุญแจเพื่อการพิสูจน์ตัวตนมาประยุกต์ใช้

คุณสมบัติที่สำคัญของลายมือชื่อดิจิตอลมีดังนี้

1. สามารถยืนยันได้ว่าข้อมูลที่ได้รับมานั้นมีการเปลี่ยนแปลงระหว่างการส่ง
2. สามารถยืนยันได้ว่าข้อมูลนั้นได้รับการยืนยันจากผู้ลงลายมือชื่อจริงๆ



รูปที่ 3.7 วิธีการสร้างลายมือชื่อดิจิตอลโดยใช้แฮชฟังก์ชันและกุญแจต่างๆ

ระบบของลายเซ็นดิจิตอลสามารถแบ่งเป็นขั้นตอนได้ดังนี้

1. เมื่อผู้ใช้ต้องการจะส่งข้อมูลไปยังผู้รับ ข้อมูลนั้นจะถูกนำไปเข้าฟังก์ชั่นทางคณิตศาสตร์ที่เรียกว่า "แฮชฟังก์ชัน" ได้เมสเสจไดเจสต์ (Message Digest) ออกมา

2. การใช้กุญแจส่วนตัวเข้ารหัสข้อมูล หมายถึงว่าผู้ส่งได้ลงลายเซ็นดิจิตอล ยินยอมที่จะให้ผู้รับสามารถทำการตรวจสอบด้วยกุญแจสาธารณะของผู้ส่งเพื่อพิสูจน์ตัวตน

3. การตรวจสอบข้อมูลว่าถูกส่งมาจากผู้ส่งคนนั้นจริงในด้านผู้รับ โดยการนำข้อมูลมาผ่านแฮชฟังก์ชันเพื่อคำนวณหาค่าเมสเสจไดเจสต์ และถอดรหัสลายเซ็นอิเล็กทรอนิกส์ด้วยกุญแจสาธารณะของผู้ส่ง ถ้าสามารถถอดได้อย่างถูกต้อง จะเป็นการยืนยันข้อมูลจากผู้ส่งคนนั้นจริง และถ้าข้อมูลเมสเสจไดเจสต์ที่ได้จากการถอดรหัสเท่ากันกับค่าเมสเสจไดเจสต์ในตอนต้นที่ทำการคำนวณได้ จะถือว่าข้อมูลดังกล่าวนั้นถูกต้อง

3.2.9 การพิสูจน์ตัวตนโดยใช้การถาม - ตอบ (zero-knowledge proofs)

เป็นวิธีการพิสูจน์ตัวตนโดยใช้การถาม - ตอบ เมื่อผู้ใช้เข้ามาในระบบแล้ว ระบบจะแน่ใจได้อย่างไรว่าผู้ใช้คนนั้น เป็นคนที่ได้รับอนุญาตให้เข้ามาใช้ระบบได้จริง การใช้ชื่อผู้ใช้และรหัสผ่าน ในปัจจุบันนี้ไม่มีความปลอดภัยเพียงพอต่อการเข้าใช้ระบบ เนื่องจากความรู้และวิทยาการที่ก้าวหน้า ทำให้เกิดผู้ที่ต้องการจะเข้ามาละเมิดระบบต่างๆมีมากขึ้น ทำให้ชื่อผู้ใช้และรหัสผ่านอาจจะถูกลักลอบดักข้อมูลระหว่างการสื่อสารกันได้ การที่จะทำให้ระบบมั่นใจได้ว่า ผู้ที่เข้าไปในระบบผู้นั้นเป็นผู้ที่ได้รับอนุญาตจริง นั่นก็คือ ระบบจะใช้การถาม - ตอบ ซึ่งคำถามและคำตอบเหล่านี้ ผู้ใช้จะเป็นคนสร้างคำถามและคำตอบขึ้นมาเองจากนั้นจะส่งให้กับเซิร์ฟเวอร์ ซึ่งคำถาม - คำตอบที่ผู้ใช้สร้างขึ้นมา ผู้ใช้เท่านั้นจะเป็นคนที่ทราบคำตอบของแต่ละคำถามที่ถูกสร้าง และเมื่อผู้ใช้คนนั้นๆ เข้าสู่ระบบได้ ระบบจะถามสุ่มคำถามเหล่านั้นที่ผู้ใช้คนนั้นๆ สร้างขึ้นมา ถามผู้ใช้คนนั้นๆก่อนที่จะยอมให้เข้าใช้ระบบได้จริงการให้ใช้ระบบได้จริงจะได้รับการยินยอมก็ต่อเมื่อการตอบคำตอบที่ผู้ใช้ตอบ นั้นสัมพันธ์กับคำตอบที่มีอยู่ในเซิร์ฟเวอร์ ยกตัวอย่างเช่น นาย ก. กับ นาย ข. รู้จักกันมานานและสนิทกัน นาย ก. และ นาย ข. ย่อมมีความสนิทกันเป็นส่วนตัวเมื่อนาย ก. และนาย ข. เล่น MSN กัน ต่างฝ่ายต่างจะแน่ใจได้อย่างไรว่าคนที่ตนคุยอยู่เป็นบุคคลเดียวกันกับที่ตนรู้จัก เพราะว่านาย ก. หรือ นาย ข. อาจจะทำการเข้าระบบทิ้งไว้ หรือ อาจจะมีบุคคลอื่นสามารถดักจับหลักฐานและข้อมูลที่สามารถเข้าสู่ระบบของคนใดคนหนึ่งไว้ได้ แล้วทำการสวมรอยแทนนั่นก็คือการใช้คำถามและคำตอบที่มีเพียงนาย ก. และ นาย ข. เท่านั้นที่ทราบ วิธีการพิสูจน์ตัวตนวิธีนี้ เป็นวิธีการที่ต้องใช้ความรู้ขั้นสูงในการนำมาใช้ เนื่องจากระบบจะใช้การเรียนรู้จากข้อมูลที่ได้รับ อาจจะเรียกระบบนี้ได้ว่าเป็นการนำความรู้ด้าน AI (Artificial Intelligence) มาใช้นั่นเอง

ตารางเปรียบเทียบข้อดีข้อเสียของการพิสูจน์ตัวตนแต่ละชนิด

การพิสูจน์ตัวตน	ข้อดี	ข้อเสีย
ไม่มีการพิสูจน์ตัวตนตน	ง่ายต่อการใช้งานและค่าใช้จ่ายต่ำ	ความปลอดภัยของข้อมูลจะขึ้นอยู่กับผู้ใช้ว่าจะนำข้อมูลเหล่านั้นไปใช้ในทางที่ควรหรือไม่
การพิสูจน์ตัวตนโดยใช้รหัสผ่าน	สามารถใช้ได้กับทุกระบบ	จะไม่ปลอดภัยเมื่อมีการส่งข้ามระบบเครือข่ายที่เป็นสาธารณะหรือไม่มีการเข้ารหัสข้อมูล
การพิสูจน์ตัวตนโดยใช้ PIN	- ง่ายต่อการจำและความปลอดภัยค่อนข้างดี (บัตร ATM) -สามารถสื่อสารข้ามเครือข่ายสาธารณะได้อย่างปลอดภัย	- ต้องใช้ฮาร์ดแวร์เฉพาะในการอ่าน PIN - ไม่สามารถใช้กับต่างระบบกันได้ - ราคาแพง

การพิสูจน์ตัวตนโดยใช้ password authenticators หรือ tokens แบบซิงโครนัส	- มีความปลอดภัยมากกว่าการใช้การจำรหัสผ่าน แบบธรรมดา - ไม่ต้องใช้เครื่องอ่านการ์ด - ผู้ที่ละเมิดเข้ามาไม่สามารถจะเข้ามาจู่โจมได้	- การใช้งานยุ่งยากกว่าแบบจำรหัสผ่าน - authenticator เป็นวัตถุจึงง่ายต่อการสูญหายและการถูกขโมยได้
การพิสูจน์ตัวตนตนโดยใช้ password authenticators หรือ tokens แบบอะซิงโครนัส	- มีความปลอดภัยมากกว่าการใช้การจำรหัสผ่าน แบบธรรมดา - ไม่ต้องใช้เครื่องอ่านการ์ด - เป็นวิธีการป้องกันที่ดีที่สุดเมื่อเปรียบเทียบกับ วิธีการใช้การพิสูจน์ตัวตน โดยใช้ รหัสผ่าน authenticators หรือtoken	- การใช้งานยุ่งยากกว่าแบบจำรหัสผ่าน - authenticator เป็นวัตถุจึงง่ายต่อการสูญหาย และการถูกขโมยได้ไม่สามารถป้องกันผู้ที่ละเมิดเข้ามาในระบบได้ - การใช้งานค่อนข้างยุ่งยากกว่าวิธีการใช้ "รหัสผ่านซึ่งเปลี่ยนแปลงได้ (dynamic password)" วิธีอื่นๆ
การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทาง ชีวภาพของแต่ละบุคคล	มีความปลอดภัยสูงเพราะเลียนแบบกันได้ยาก	- ระบบมีความซับซ้อนสูง - ยังไม่ได้รับความนิยมกันอย่างแพร่หลาย - ค่าใช้จ่ายสูง
การพิสูจน์ตัวตนโดยวิธี One-Time Password	ทำให้การเดาหรือขโมยรหัสผ่านเป็นไปได้ยาก	- ไม่สะดวกต่อการใช้งาน เพราะผู้ใช้ต้องจำรหัสผ่านหลายตัว - ถ้าผู้ใช้จำรหัสผ่านไม่ได้ หรือ ทำรหัสผ่านสูญหาย ก็ไม่สามารถเข้าใช้ระบบได้
การพิสูจน์ตัวตนโดยการเข้ารหัสแบบ คู่รหัสกุญแจ	- การจัดการกุญแจทำได้ปลอดภัย เพราะ ใช้กุญแจในการเข้ารหัสและถอดรหัสต่างกัน - สามารถระบุผู้ใช้โดยการใช้ร่วมกับลายมือชื่อ อิเล็กทรอนิกส์	- ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การคำนวณอย่างมาก - ต้องใช้ระบบที่สนับสนุนการทำงาน
การพิสูจน์ตัวตนโดยการใช้ลายเซ็นดิจิตอล	- สามารถระบุตัวผู้ส่งได้ชัดเจน - ป้องกันข้อมูลถูกแก้ไขระหว่างการส่งได้ หรือสามารถตรวจสอบ	ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การคำนวณอย่างมาก

	ข้อมูลได้ว่าผ่านการแก้ไขมาหรือไม่	
การพิสูจน์ตัวตนโดยวิธี zero-knowledge proofs	ความปลอดภัยค่อนข้างสูง เพราะคำถามและคำตอบจะมีเพียงผู้ใช้และเซิร์ฟเวอร์เท่านั้นที่ทราบ	ความซับซ้อนของระบบเพิ่มขึ้นตามความฉลาดของระบบ

ตารางที่ 3.1 แสดงการเปรียบเทียบข้อดีข้อเสียของการพิสูจน์ตัวตนแต่ละชนิด

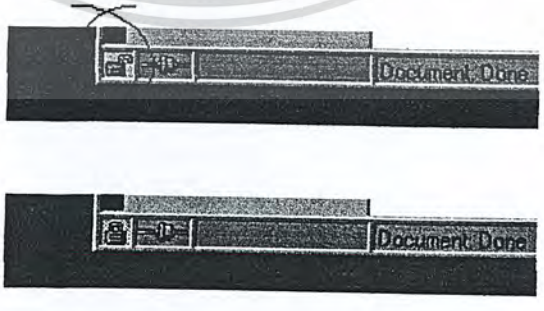
3.3 Secure Socket Layer (SSL)

SSL ย่อมาจากคำว่า Secure Sockets Layer เป็นโปรโตคอลที่เพิ่มความปลอดภัยในการรับส่งข้อมูลผ่านระบบเครือข่าย ทำให้เราสามารถส่งข้อมูลที่เป็นความลับเช่น รหัสผ่าน หรือหมายเลขบัตรเครดิตผ่านระบบเครือข่ายได้ด้วยความปลอดภัย นอกจากผู้ส่งและผู้รับข้อมูลแล้วไม่มีใครในระบบเครือข่ายสามารถดึงข้อมูลที่เป็นความลับไปใช้ได้

3.3.1 การใช้งาน SSL

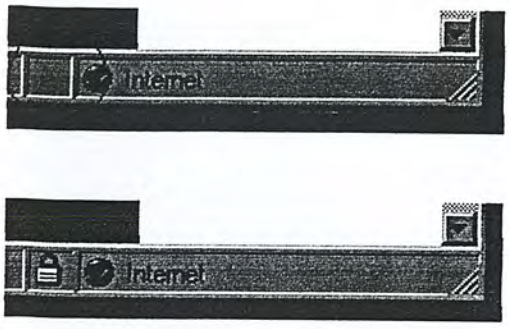
ปัจจุบัน SSL ถูกนำมาใช้อย่างมากในเครือข่าย WWW โดยถูกผนวกเข้าไปกับบราวเซอร์ ปัจจุบันบราวเซอร์ที่รองรับ SSL ได้แก่ เน็ตสเคป และอินเตอร์เน็ตเอ็กพลอเรอร์ โดยผู้ใช้จะใช้งานเหมือนบราวเซอร์ปกติ ยกเว้น WWW page ที่เป็น SSL จะใช้ https นำหน้าแทน http หมายความว่าให้บราวเซอร์ดึงข้อมูลผ่านโปรโตคอล https (SSL http) ที่หมายเลข tcp port 443 แทน port 80 ที่เป็นของ http

สำหรับเน็ตสเคปให้สังเกตที่มุมล่างซ้ายของโปรแกรม ในส่วนที่เป็นรูปแม่กุญแจจะแสดงสถานะของ SSL ในขณะนั้นว่าข้อมูลที่ผู้ใช้กำลังอ่านอยู่ถูกส่งมาโดยผ่าน SSL หรือไม่ รูปบนแสดงว่าส่งผ่านมาทาง http ธรรมดา ส่วนรูปล่างส่งผ่าน SSL โดย https



รูปที่ 3.8 แสดงNetscape ที่มีการใช้ SSL

ทางด้านอินเตอร์เน็ตเอ็กพลอเรอร์ ก็มีลักษณะเหมือนกัน เพียงแต่สัญลักษณ์แสดงสถานะของ SSL จะปรากฏอยู่ ณ มุมขวาล่างของโปรแกรมดังรูปที่3.9



รูปที่ 3.9 แสดง Internet Explorer ที่มีการใช้ SSL

3.3.2 ประโยชน์ของการใช้ SSL

SSL สามารถเพิ่มความปลอดภัยของการส่งข้อมูลผ่านระบบเครือข่ายได้ในด้าน

1. การกำหนดสิทธิ์หรือการรักษาความลับ
2. การพิสูจน์ตัวตน
3. การรักษาความสมบูรณ์

อย่างไรก็ตามการใช้ SSL มิได้หมายความว่าข้อมูลที่เป็นความลับจะรั่วไหลออกไปภายนอกไม่ได้เลย เพราะจุดที่รั่วไหลก็อาจจะมาจากที่ตัวผู้รับหรือผู้ส่งข้อมูลเอง เช่นผู้รับข้อมูลนำข้อมูลไปเปิดเผยต่อ หรือการเก็บข้อมูลไว้บนเครื่องคอมพิวเตอร์โดยไม่ได้ป้องกันการคัดลอกข้อมูลโดยผู้ใช้คนอื่น

3.3.3 ขั้นตอนการทำงานของ SSL

1) ผู้ใช้งานเริ่มกระบวนการติดต่อ ไปยังเว็บเซิร์ฟเวอร์ที่มีระบบ SSL

2) เซิร์ฟเวอร์จะส่งใบรับรอง (Server Certificate) กลับมาพร้อมกับเข้ารหัส ด้วยกุญแจสาธารณะของเซิร์ฟเวอร์

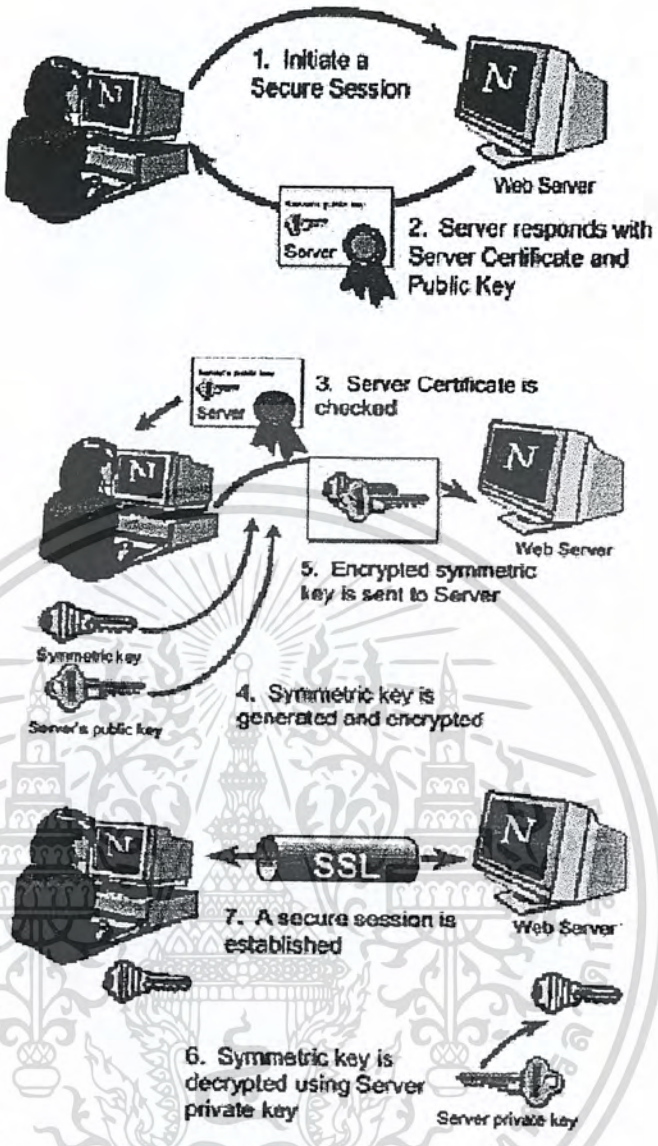
3) ฝั่งผู้รับจะทำการตรวจสอบใบรับรองนั้นอีกทีเพื่อตรวจสอบตัวตนของฝั่งผู้ค้า

4) สร้างกุญแจสมมาตร

5) ทำการเข้ารหัสกุญแจสมมาตรด้วยกุญแจสาธารณะของเซิร์ฟเวอร์ที่ได้รับมาและส่งกลับไปยังเซิร์ฟเวอร์

6) ทำการถอดรหัสด้วยกุญแจส่วนตัว ก็จะได้กุญแจสมมาตรของลูกค้ามาไว้ใช้ในการติดต่อสื่อสาร

7) การติดต่อสื่อสารกันก็ใช้การเข้ารหัสติดต่อสื่อสารกันได้อย่างปลอดภัย



รูปที่ 3.10 แสดงขั้นตอนการทำงานของ SSL

3.4 WS-Security

ในการพัฒนาแอพพลิเคชั่นให้สามารถเรียกใช้งานได้ในขณะรันไทม์คล้ายกับ CRL เราควรจะพัฒนาโดยใช้เว็บเซอร์วิสซึ่งสามารถทำงานภายใต้สภาพแวดล้อมที่แตกต่างกันได้ เนื่องจากการพัฒนาเว็บเซอร์วิสจะใช้มาตรฐานเดียวกันในการพัฒนานั่นคือมาตรฐาน XML นอกจากนี้เว็บเซอร์วิสยังเป็นตัวอย่างการพัฒนาแอพพลิเคชั่นแบบกระจาย หัวใจสำคัญก็คือ โปรโตคอลSOAP ซึ่งนิยามภายใต้มาตรฐาน XML ทำให้โปรแกรมที่พัฒนาด้วยภาษาที่แตกต่างกันสามารถที่จะทำการติดต่อกันได้ นอกจากนี้ยังเป็นโปรโตคอลที่ไม่ขึ้นกับโปรโตคอลในระดับทรานสปอร์ต ความจริงแล้ว SOAP เมสเสจแสดงอยู่ในรูปของ XML และจะต้องมีการรักษาความปลอดภัยในข้อมูลที่มีความสำคัญ ปัจจุบันมีโปรโตคอล SSLซึ่งสามารถรักษาความปลอดภัยได้ในระดับหนึ่งเท่านั้นแต่ การที่จะนำ SSL มาใช้กับเว็บเซอร์วิส จะทำให้การรักษาความปลอดภัยไม่คลอบคลุมการทำงานของเว็บเซอร์วิสทั้งหมด เนื่องจาก SSL เป็นการรักษาความปลอดภัยแบบจุดต่อจุด การส่งข้อมูลจะมีความปลอดภัย ถ้าข้อมูลนั้นไม่ผ่านตัวกลางอื่น และเนื่องจาก SSL เป็น

โปรโตคอลที่ต้องส่งผ่าน โปรโตคอล HTTP ดังนั้นถ้าพัฒนาเว็บเซอร์วิส โดยมีการส่ง SOAP ไปบนโปรโตคอลที่ไม่ใช่ HTTP โดยที่ใช้ SSL ในการรักษาความปลอดภัยจะไม่สามารถทำได้

WS-Security เป็นเทคโนโลยีใหม่ที่รักษาความปลอดภัยในเว็บเซอร์วิส ผู้สร้างเว็บเซอร์วิสจะต้องตระหนักถึงเรื่องของความปลอดภัยเป็นหลัก โดยที่ผู้สร้างเว็บเซอร์วิสไม่จำเป็นต้องสร้างระบบรักษาความปลอดภัยอันใหม่ขึ้นมาใช้งาน แต่สามารถที่จะนำระบบรักษาความปลอดภัยที่มีอยู่เดิมมาใช้งานร่วมกับ WS-Security ได้ เช่น UsernameToken, Kerberos , PKI ซึ่งจะเลือกใช้ระบบรักษาความปลอดภัยแบบใดก็ขึ้นกับจุดประสงค์และขอบเขตที่นำไปใช้ในเว็บเซอร์วิส

รายละเอียดที่เกี่ยวข้องกับความปลอดภัยของระบบ มีดังนี้

1) WS-Security เป็นพื้นฐานของระบบต่างๆ ซึ่งใช้ SOAP ในการรองรับ security tokens ซึ่งบ่งบอกถึงความปลอดภัย และการพิสูจน์ตัวตนของผู้ใช้ และแน่ใจว่าข้อความนั้นมีความถูกต้อง และมีการจัดหาข้อความที่เชื่อถือได้ ซึ่งจะทำการเพิ่มเติมข้อมูลต่างๆ ลงในเอกสารต้นฉบับ

2) นโยบายของ WS-SecurityPolicy ซึ่งจะระบุและอธิบายการรักษาความปลอดภัยได้อย่างชัดเจนในเว็บเซอร์วิสได้ตามต้องการ โดยจะมีการกำหนดลักษณะการรักษาความปลอดภัยไว้ในโพลีซี ไฟล์ เมื่อมีการติดต่อกับเว็บเซอร์วิสที่มีการกำหนดโพลีซีไฟล์เว็บเซอร์วิสจะมาตรวจสอบที่ไฟล์นี้ก่อนว่าตรงกับนโยบายที่ได้ไว้หรือไม่

3) WS-Trust ซึ่งจะอธิบายถึงการได้รับ SecurityToken

4) WS-SecureConversation ซึ่งใช้ในการสร้างข้อความสำหรับการติดต่อที่เป็นกรณีพิเศษ และใช้ในการสร้างกุญแจ ซึ่งใช้ในข้อความนั้น

สิ่งที่เป็นพื้นฐานของ WS-Security เป็นส่วนประกอบซึ่งเป็นส่วนหนึ่งของ SOAP โครงสร้างข้อมูลของ SOAP จะรวมอยู่ใน Security Element ดังตัวอย่างรูปที่ 3.11 นี้

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">
<s:Header>
 <wsse:Security> ••• </wsse:Security>
</s:Header>
<s:Body> ••• </s:Body>
</s:Envelope>
```

รูปที่ 3.11 แสดงโครงสร้างของSecurity Element

โครงสร้างพื้นฐานของ WS-Security สามารถอธิบายโดย namespace ของมัน ผู้สร้าง ws-security จะขึ้นกับมาตรฐานที่มีอยู่ และเทคโนโลยีต่างๆ ที่เป็นไปได้

WS-Security อธิบายได้จากความมั่นคงที่ได้รับ ซึ่งอนุญาตจากผู้รับถึงความถูกต้องของข้อมูลที่ได้รับว่าไม่เปลี่ยนแปลงในขณะที่ทำการส่ง และมีสภาพที่เป็นความลับ ซึ่งรับรองว่าข้อมูลไม่ถูกเปิดเผยในระหว่างที่ทำการส่ง สามารถอธิบายวิธีการส่งของซีเคียวริตี้โทเคน เช่น การใช้ username และ password ,Kerberos ticket หรือ การรับรองจาก X.509 ในการอธิบาย

3.4.1 การส่งซีเคียวริตี้โทเคน

คำถามของผู้รับส่วนใหญ่ก็คือ ฉันกำลังคุยกับใคร เป็นคำถามที่ง่ายแต่ตอบยาก ในทางเน็ตเวิร์คมีหลายวิธีที่จะใช้ในการตรวจสอบเช่น username password ,Kerberos,X.509 certificate Username password จะใช้รหัสผ่านในการตรวจสอบ เป็นต้น ตัวอย่างของ usernametoken

```
<SOAP:Envelop xmlns:SOAP="..">
<SOAP:Header>
<wsse:Security SOAP:mustUnderstand="1" xmlns:wsse="..">
<wsse:UsernameToken xmlns:wsu=".."wsu:Id="..">
<wsse:Username>John Doe</wsse:Username>
<wsse:Password Type="wsse:PasswordDigest">SeseKRseOes...</wsse:Password>
<wsse:Nonce> WsYrGcesIiLeWs ...</wsse:Nonce>
<wsu:Created> ....</wsu:Created>
</wsse:UsernameToken>
</wsse:Security>
</SOAP:Header>
<SOAP:Body> ... </SOAP:Body>
</SOAP:Envelope>
```

คำอธิบาย

- wsu:UernameToken : แสดงถึงการใช้ ยูสเซอร์เนมโทเคน
- wsu:Id : เป็นตัวที่ระบุ uri เพื่อใช้ในการระบุโทเคน
- wsu:Username : แสดงชื่อ ยูสเซอร์เนม
- wsu:Password : แสดงพาสเวิร์ดของยูสเซอร์
- wsse:Password/@Type : อธิบายประเภทของพาสเวิร์ด
- wsse: Nonce : เป็นค่าที่มีไว้เพื่อป้องกันการ โจมตีแบบซ้ำๆ (replay attack)
- wsu:Created : เป็นค่าที่บอกว่า เมสเสจนี้ถูกสร้างขึ้นเมื่อไร

BinarySecurityToken เป็นการนิยามโครงสร้างที่ภาย โดยที่ซีเคียวริตี้ภายในไม่ได้อยู่บนพื้นฐานของความปลอดภัยของ XML

<wsse:BinarySecurityToken ValueType="wsse:X509v3" EncodingType="wsse:Base64Binary" wsu:Id=" .."> MimeCcese </wsse:BinarySecurityToken>

wsse:BinarySecurityToken : แสดงถึงไบนารีโทเคน

wsse: BinarySecurityToken/@ValueType : บอกประเภทของ โทเคนที่ใช้เช่น x.509

wsse: BinarySecurityToken/@EncryptingType : บอกว่าเข้ารหัสด้วยวิธีใด เช่น Base64

3.4.2 การรักษาความสมบูรณ์ของข้อมูล

การรักษาความถูกต้องข้อมูลเพื่อตรวจสอบว่ามีการเปลี่ยนแปลงของข้อมูลหรือไม่ในระหว่างการส่งข้อมูล การรักษาความถูกต้องของข้อมูลจะทำผ่าน XML signature ซึ่งเป็นมาตรฐานของ W3C เราสามารถทำลายเซ็นอิเล็กทรอนิกส์โดยการใช้เทคโนโลยีที่มีอยู่ในปัจจุบันได้เช่น สามารถใช้ Kerberos token มาใช้ในการทำลายเซ็นอิเล็กทรอนิกส์ได้ นอกจาก Kerberos แล้วยังสามารถใช้ username password หรือใช้เทคโนโลยีกุญแจสาธารณะ มาทำงานในลักษณะนี้ได้ด้วย

อีลีเมนต์ที่สำคัญในการทำลายเซ็นอิเล็กทรอนิกส์คือ < signature> เป็นอีลีเมนต์เริ่มต้นที่จะประกอบด้วยอีลีเมนต์ย่อย ที่จะใช้ในการทำลายเซ็นอิเล็กทรอนิกส์ โดยมี อีลีเมนต์ <Reference> เป็นตัวที่ใช้ในการระบุว่าอีลีเมนต์ใดใน SOAP ที่ถูกทำลายเซ็นอิเล็กทรอนิกส์ โดยจะมีอีลีเมนต์ <KeyInfo> เป็นตัวที่จะบอกว่าควรจะใช้กุญแจคอกไหนในการในการทำลายเซ็นอิเล็กทรอนิกส์

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility">
<s:Header>
 <wsse:Security>
  <wsse:BinarySecurityToken
   ValueType="wsse:X509v3"
   EncodingType="wsse:Base64Binary"
   wsu:Id="X509Cert">
   KkFPle ...
  </wsse:BinarySecurityToken>
```

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <ds:SignedInfo>
  <ds:CanonicalizationMethod
   Algorithm="http://www.w3.org/2001/10/xml-exc-c14N"/>
  <ds:SignatureMethod
   Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#MessageBody">
   <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <ds:DigestValue>
    aOb4Luuk...
   </ds:DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>
  A9qqIrtE3xZ...
 </ds:SignatureValue>
 <ds:KeyInfo>
  <wsse:SecurityTokenReference>
   <wsse:Reference URI="#X509Cert"/>
  </wsse:SecurityTokenReference>
 </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</s:Header>
<s:Body wsu:Id="MessageBody">
 •••
</s:Body>
</s:Envelope>
```

รูปที่ 3.12 แสดงการสร้างลายเซ็นเอ็กทรอนิกส์

3.4.3 การรักษาความลับ

การเก็บข้อมูลให้ปลอดภัยจากคนที่สอดรู้สอดเห็นในขณะที่มันกำลังอยู่ในระหว่างการส่งข้อมูลเป็นเรื่องที่สำคัญ WS-Security สามารถจัดการเกี่ยวกับการเข้ารหัสข้อมูลบางส่วน หรือทั้งหมดของ SOAP เมสเสจ ก่อนที่จะทำการส่งต่อไป ผู้สร้าง WS-Security จะเป็นผู้สร้างมาตรฐานสำหรับความไว้วางใจได้ ซึ่งมาตรฐานหนึ่งที่ถูกเลือกนำมาใช้คือ XML Encryption ซึ่งมาจาก W3C การใช้มาตรฐานนี้นั้น WS-Security อนุญาตให้เข้ารหัสทั้งหมด หรือเฉพาะบางส่วนของข้อมูลใน เฮดเดอร์ บอดี้ และข้อมูลที่แนบมาของ SOAP message นั้น XML Encryption ไม่ได้มีความซับซ้อนมากเท่าใดนัก สำหรับตัวอย่างที่ใช้สามอีลีเมนต์ของ XML ในการอธิบายมาตรฐานนี้คือ <EncryptedData>, <EncryptedKey> และ <ReferenceList>

สิ่งที่สำคัญที่สุดของทั้งสามสิ่งนี้คือ <EncryptedData> element นี้ประกอบด้วยข้อมูลที่ถูกเข้ารหัสแล้วอยู่ในอีลีเมนต์ย่อยซึ่งจะมีชื่อเรียกว่า <CipherData> ในอีลีเมนต์นี้จะประกอบด้วย อีลีเมนต์ ย่อย ซึ่งจะแสดงอัลกอริทึม ในการเข้ารหัส กุญแจที่จะใช้ในการเข้ารหัส และอื่นๆ

ในส่วนของบอดี้ของเมสเสจ จะประกอบด้วย <EncryptedData> กับอีลีเมนต์ย่อยอีก 3 อีลีเมนต์ คือ

1.อีลีเมนต์แรกคือ <EncryptionMethod> จะบอกถึงอัลกอริทึมในการเข้ารหัสข้อมูล ในกรณีนี้อัลกอริทึม ชนิด TripleDES จะถูกเลือก และกุญแจที่เหมือนกันจะถูกใช้ในการเข้ารหัส และถอดรหัส

2. อีลีเมนต์ ที่สองคือ <KeyInfo> จะมีการใช้ข้อมูลจาก XML Signature มันจะสมมติว่าทั้งสองฝั่งที่ติดต่อกันนั้นรู้กุญแจที่ใช้ในการเข้ารหัส และถอดรหัส ตัวอย่างชื่อของกุญแจ เช่น "CN=Key13, C=US" ไม่ว่าชื่อใดก็ตามที่ถูกใช้นั้น สิ่งที่สำคัญที่สุดคือทั้งสองฝั่งต้องเข้าใจที่กุญแจนี้อ้างถึง

3.อีลีเมนต์สุดท้ายคือ <CipherData> เป็นอีลีเมนต์ย่อยที่จะมี <CipherValue> เป็นอีลีเมนต์ย่อยอีกทีที่จะเป็นการเข้ารหัสข้อมูลที่แท้จริง

มันเป็นไปได้ที่จะส่งกุญแจที่ถูกเข้ารหัสในเมสเสจ เดียวกันที่ข้อมูลนั้นถูกเข้ารหัสด้วยกุญแจ ตัวอย่างในกรณีนี้เมื่อข้อมูลถูกเข้ารหัสโดยใช้กุญแจสมมาตร โดยใช้ TripleDES ดังรูปที่3.13 นี้

```
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
<s:Body>
<xenc:EncryptedData>
 <EncryptionMethod
  Algorithm='http://www.w3.org/2001/04/xmlenc#tripledes-cbc'/>
 <ds:KeyInfo>
```

```
<ds:KeyName>
 CN=Key13, C=US
</ds:KeyName>
</ds:KeyInfo>
<xenc:CipherData>
<xenc:CipherValue>
 r5KipsDV . . .
</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</s:Body>
</s:Envelope>
```

รูปที่ 3.13 แสดงการเข้ารหัส SOAP MESSAGE

แล้วเมื่อกุญแจสมมาตรถูกเข้ารหัสโดยใช้กุญแจสาธารณะ แล้วทำการส่งข้อมูลต่อไป เมื่อได้รับเมสเสจ แล้วผู้รับสามารถที่จะใช้กุญแจส่วนตัวในการถอดรหัสเพื่อดึงเอากุญแจสมมาตรออกมา แล้วใช้กุญแจสมมาตรนี้ในการถอดรหัสข้อมูลที่แท้จริง การส่งกุญแจที่เข้ารหัสนั้น XML Encryption จะกำหนดอีลีเมนต์ กับชื่อที่ชัดเจนของ <EncryptedKey> เพื่อใช้กับ SOAP เมสเสจ ซึ่ง WS-Security จะกำหนดว่าอีลีเมนต์นี้ควรจะอยู่ใน <Security> เฮดเดอร์ ดังนั้น SOAP เมสเสจ จะมีทั้งข้อมูลที่เข้ารหัสแล้ว และกุญแจสมมาตรที่เข้ารหัสแล้ว ดูตัวอย่างได้จากรูปที่ 3.14 ดังนี้

```
<s:Envelope
xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext"
xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
<s:Header>
<wsse:Security>
<xenc:EncryptedKey>
<xenc:EncryptionMethod
 Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
<ds:KeyInfo>
```

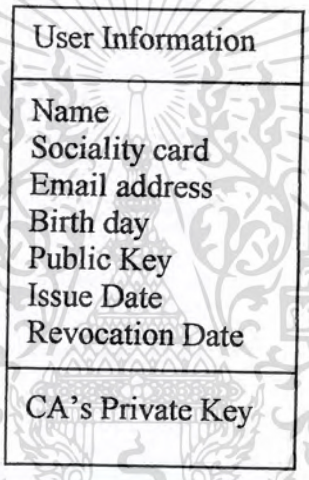
```
<ds:KeyName>
CN=Key13, C=US
</ds:KeyName>
</ds:KeyInfo>
<xenc:CipherData>
<xenc:CipherValue>
ir4DfG . . .
</xenc:CipherValue>
</xenc:CipherData>
<xenc:ReferenceList>
<xenc:DataReference URI="#EncryptedBody"/>
</xenc:ReferenceList>
</xenc:EncryptedKey>
</wsse:Security>
</s:Header>
<s:Body>
<xenc:EncryptedData wsu:Id="EncryptedBody">
<xenc:EncryptionMethod
Algorithm='http://www.w3.org/2001/04/xmlenc#tripledes-cbc'/>
<xenc:CipherData>
<xenc:CipherValue>
r5KipsDV . . .
</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</s:Body>
</s:Envelope>
```

รูปที่ 3.14 แสดงการเข้ารหัส SOAP Message ด้วยกุญแจ

บทที่ 4
เอกสารสิทธิ์

4.1 ลักษณะของเอกสารสิทธิ์ดิจิตอล (Digital Certificate)

การรับรองและพิสูจน์ตนในการติดต่อสื่อสาร เมื่อผู้ส่งส่งข้อมูลมายังผู้รับ ผู้รับต้องตรวจสอบว่าผู้ส่งคือใครและสามารถเชื่อถือข้อมูลที่ส่งมาได้มากน้อยเพียงใด การรับรองและการพิสูจน์ตนในการส่งข้อมูลแบบอิเล็กทรอนิกส์นั้นเป็นเรื่องที่ซับซ้อนมาก เพราะเราไม่สามารถรู้ว่าใครเป็นผู้ส่งที่แท้จริง เพราะสามารถทำการดักจับข้อมูลมาแก้ไข รวมถึงแอบอ้างสิทธิ์ของผู้ส่งเอง



รูปที่ 4.1 แสดงตัวอย่างของเอกสารสิทธิ์

เอกสารสิทธิ์เป็นเหมือนกับบัตรประจำตัวของบุคคลนั้น ซึ่งจะบ่งบอกรายละเอียดของบุคคล เราสามารถส่งเอกสารสิทธิ์ไปพร้อมกับลายมือชื่อดิจิตอลเพื่อใช้ในการอ้างถึงผู้ส่ง การสร้างเอกสารสิทธิ์ทำโดยผู้ใช้ทุกคนทำการขอสิทธิ์กับองค์กรพิสูจน์สิทธิ์ (Certificate Authority) โดยการส่งกุญแจสาธารณะและข้อมูลตามที่องค์กรพิสูจน์สิทธิ์นั้นกำหนดไปและทำการขอเอกสิทธิ์มา การติดต่อต้องทำเป็นการส่วนตัวหรือติดต่อผ่านระบบการพิสูจน์บุคคลที่ปลอดภัย เราสามารถกำหนดสิ่งที่จำเป็นในการสื่อสารดังต่อไปนี้

1. ผู้ใช้ทุกคนสามารถคำนวณหาชื่อและกุญแจสาธารณะของเจ้าของเอกสารสิทธิ์ได้
2. ผู้ใช้ทุกคนสามารถตรวจสอบได้ว่าเอกสารสิทธิ์มาจากองค์กรพิสูจน์สิทธิ์จริงๆ ไม่ได้ถูกปลอมแปลมมา
3. ผู้ใช้สามารถตรวจสอบได้ว่าเอกสารสิทธิ์นั้นหมดอายุหรือไม่
4. ผู้ที่สามารถสร้างและอัพเดตเอกสารสิทธิ์ได้มีเพียงองค์กรผู้มีอำนาจในการรับรองสิทธิ์เท่านั้น

5. ผู้ใช้ทุกคนสามารถตรวจสอบเอกสารสิทธิ์ได้เป็นประจำ

4.2 ความสำคัญของเอกสารสิทธิ์ดิจิตอล

ในระบบการเข้ารหัสแบบกุญแจต่างนั้นเราจะต้องมีการสร้างทั้งกุญแจส่วนตัวและกุญแจสาธารณะ ซึ่งโดยทั่วไปการสร้างกุญแจจะทำโดยโปรแกรมที่จะใช้กุญแจนั้น เช่น โปรแกรมเว็บบราวเซอร์หรือโปรแกรมติดต่ออิเล็กทรอนิกส์เมล์ หลังจากที่สร้างกุญแจทั้งสองเรียบร้อยแล้ว จะต้องเก็บรักษากุญแจส่วนตัวเอาไว้ให้ดีอย่าให้ใครขโมยไปได้ จากนั้นจะเป็นการตัดสินใจว่าจะแจกกุญแจสาธารณะของเราไปสู่ผู้อื่นด้วยวิธีใด เช่น อาจแจกกุญแจโดยส่งอิเล็กทรอนิกส์เมลล์ไปให้เพื่อนหรือบุคคลที่เราติดต่อ แต่วิธีนี้เราอาจส่งกุญแจไปให้ไม่ครบทุกคน และยังจะต้องเป็นภาระคอยจัดการส่งกุญแจให้กับบุคคลใหม่ๆ ที่ต้องการติดต่อด้วย นอกจากนี้ยังไม่สามารถทำให้ผู้รับมั่นใจได้ว่ากุญแจที่ส่งไปให้นั้นเป็นกุญแจของเราจริงๆ เนื่องจากอาจมีผู้อื่นแอบสร้างกุญแจโดยใช้ชื่อเราและแอบอ้างส่งกุญแจดังกล่าวให้กับผู้อื่นเพื่อให้เข้าใจว่าเป็นกุญแจเราก็ได้ วิธีที่ใช้อยู่ในปัจจุบันก็คือการใช้ระบบแจกจ่ายกุญแจที่เชื่อถือได้โดยจะมีองค์กรที่ทำหน้าที่เฉพาะเป็นองค์กรที่สาม (Third Party) ในการรับรองและระงับการรับรองกุญแจที่เรียกว่า องค์กรพิสูจน์สิทธิ์ (Certificate Authority – CA) โดยองค์กรพิสูจน์สิทธิ์นี้จะตรวจสอบกุญแจสาธารณะของเราด้วยหลักฐานว่ากุญแจนั้นเป็นของเราจริงๆ พร้อมทั้งตรวจสอบข้อมูลส่วนตัวของเรา (ข้อมูลที่ตรวจสอบจะมากน้อยแค่ไหนขึ้นอยู่กับระดับชั้นของการรับรอง) เมื่อผู้อื่นได้รับกุญแจของเราก็สามารถที่จะตรวจสอบกับผู้ออกเอกสารสิทธิ์นี้ว่ากุญแจที่ได้รับเป็นของเราจริงหรือไม่ ซึ่งตัวเอกสารสิทธิ์นี้จะเปรียบเสมือนบัตรประชาชนดิจิตอลของเราที่ใช้บอกได้ว่าเราเป็นบุคคลที่อ้างจริงๆ ในระบบเครือข่ายหรือการส่งข้อมูลทางอิเล็กทรอนิกส์

ในปัจจุบันบริษัทหลักๆ ที่ออกเอกสารสิทธิ์ดิจิตอล คือบริษัท verisign ,Cybertrust , Globla Sign และ Nortel โดยในเอกสารสิทธิ์ดิจิตอลจะประกอบด้วยข้อมูลต่างๆ ดังนี้ ชื่อของผู้ถือเอกสารสิทธิ์ ชื่อของบริษัทของบริษัทที่ออกเอกสารสิทธิ์ กุญแจสาธารณะของผู้ถือเอกสารสิทธิ์ วันหมดอายุของเอกสารสิทธิ์ ระดับชั้นของเอกสารสิทธิ์ และเลขหมายของตัวเอกสารสิทธิ์ดิจิตอล

เอกสารสิทธิ์ดิจิตอลแบ่งออกได้เป็นสี่ระดับชั้นตามระดับการตรวจสอบข้อมูลของเจ้าของเอกสารสิทธิ์

1. ระดับชั้นที่หนึ่งเป็นชั้นที่ออกเอกสารสิทธิ์ได้ง่ายที่สุด เนื่องจากมีการตรวจสอบน้อยที่สุด โดยจะตรวจสอบแค่ชื่อผู้ถือเอกสารสิทธิ์ และที่อยู่อิเล็กทรอนิกส์ (e-mail address) ว่าถูกต้องจริงเท่านั้น

2. ระดับชั้นที่สองจะตรวจสอบเลขบัตรประจำตัวประชาชน เลขประจำตัวของระบบสวัสดิการหรือประกันสังคม และวันเดือนปีเกิด

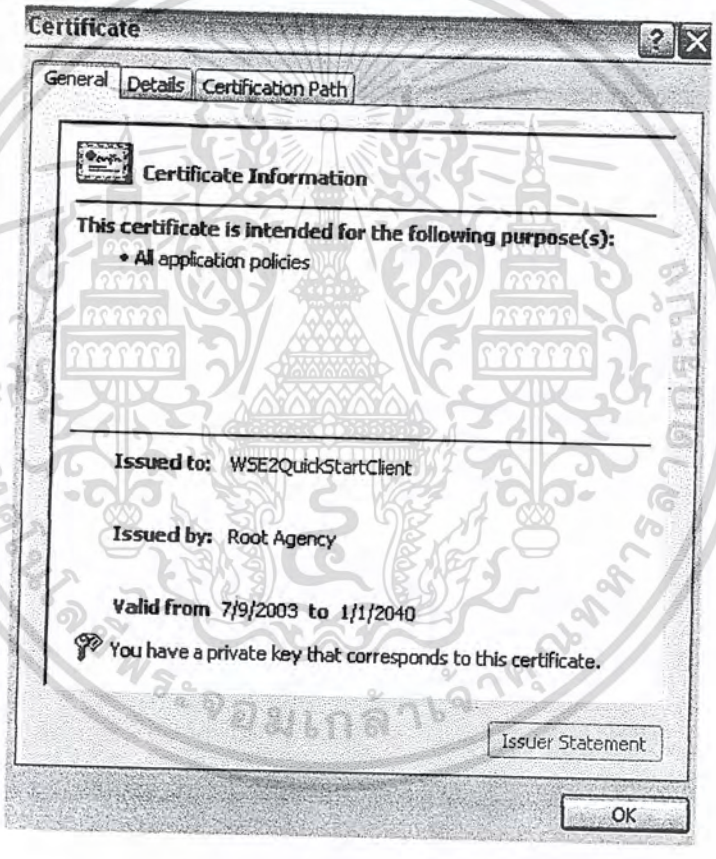
3. ระดับชั้นที่สามจะมีการตรวจสอบเพิ่มเติมเกี่ยวกับประวัติการใช้เครดิตและการชำระเงิน

4. ระดับชั้นที่สี่นั้นยังไม่มีการออกมาเป็นมาตรฐานอย่างแน่ชัด แต่จะเป็นการตรวจสอบข้อมูลเพิ่มเติมเกี่ยวกับตำแหน่งงานในองค์กรด้วย

ในการขอเอกสารสิทธิ์นั้นจะต้องกระทำการบนกระบวนการที่ปลอดภัย ซึ่งปัจจุบันมี 2 วิธีคือ

1. การขอเอกสารสิทธิ์ผ่านโปรแกรมประเภทบราวเซอร์ เช่น อินเตอร์เน็ตเอ็กซ์พลอเรอร์ เน็ตสเคป เป็นต้น โดยผู้ใช้เข้าไปยังโฮสต์ที่ให้บริการการขอเอกสารสิทธิ์หรือเว็บที่เปิดให้บริการดังกล่าวอยู่ ตัวโปรแกรมประเภทบราวเซอร์จะทำการสร้างกุญแจคู่ กุญแจส่วนตัวจะเก็บไว้ที่เครื่องของเรา จากนั้นจะส่งกุญแจสาธารณะและข้อมูลส่วนตัวของเราไปยังโฮสต์หรือไซต์ที่เป็นขององค์กรพิสูจน์ จากนั้นองค์กรพิสูจน์จะทำการรับรองข้อมูลแล้วจึงส่งเอกสารสิทธิ์กลับมาที่เครื่องเรา

2. การขอเอกสารสิทธิ์อีกรูปแบบหนึ่งคือการส่งเอกสารสิทธิ์ที่ไม่ผ่านระบบเครือข่าย กล่าวคือผู้ขอต้องไปทำการขอที่สำนักงานขององค์กรพิสูจน์สิทธิ์โดยตรง โดยทำเรื่องขอเอกสารสิทธิ์และกรอกข้อมูลส่วนตัว องค์กรพิสูจน์สิทธิ์จะทำการรับรองและบันทึกข้อมูลของผู้ขอแล้วทำการส่งเอกสารสิทธิ์พร้อมทั้งกุญแจส่วนตัวให้กับผู้ขอในรูปของไฟล์แผ่นดิสก์ให้ผู้ขอนำไปบันทึกในเครื่องของตนต่อไป



รูปที่ 4.2 แสดงตัวอย่างเอกสารสิทธิ์โดยใช้โปรแกรมประเภทบราวเซอร์

4.3 การบริการพิสูจน์สิทธิ์แบบ X.509 (X.509 Authentication Service)

ระบบ X.509 เป็นระบบพิสูจน์สิทธิ์ที่สำคัญมากในระบบเครือข่าย โดย X.509 เป็นอนุกรมย่อยของ X.500 ซึ่งกำหนดมาตรฐาน ITU-T โดยขณะที่ X.500 เป็นตัวกำหนดโครงสร้างในลักษณะที่เป็นไดเร็กทอรี่หรือโครงสร้างนั้น X.509 จะทำหน้าที่ในการพิสูจน์สิทธิ์ให้กับส่วนต่างๆ ของไดเร็กทอรี่นั้น สำหรับรูปแบบการใช้งานจะเน้นไปที่การพิสูจน์บุคคล เพื่อยืนยันการติดต่อเป็นสำคัญ

การทำงานของ X.509 จะมีโครงสร้างการทำงานที่เป็นไดเร็กทอรี่ โดยที่นี้ไดเร็กทอรี่จะทำหน้าที่เป็นที่เก็บข้อมูลที่ใช้ในการยืนยัน ซึ่งโดยทั่วไปจะอยู่ในรูปของเอกสารสิทธิ์ซึ่งในเอกสารสิทธิ์จะบรรจุกุญแจสาธารณะของผู้ใช้ที่เข้ารหัสโดยกุญแจส่วนตัวขององค์กรที่จ่ายใบเอกสารสิทธิ์มาให้ สำหรับการทำงานของ X.509 นั้นจะมีขอบเขตการนำไปใช้งานที่กว้างขวางมาก เช่น ใช้ในการทำ Mail Security ใช้ในการทำ IP Security ใช้ในการทำ Web Security หรือหากจะกล่าวว่าเมื่อใดที่ต้องการพิสูจน์บุคคลหรือยืนยันเครื่องคอมพิวเตอร์แล้ว ก็มักจะอยู่ขอบข่ายการทำงานของ X.509 เสมอ

X.509 ได้ถูกนำเสนอเมื่อปี 1988 จากนั้นได้ผ่านการปรับปรุงเป็นลำดับขั้น ในประเด็นต่างๆ รวมทั้งเรื่องความปลอดภัยด้วย จากนั้นก็ได้ออกมาเป็นข้อเสนอที่ปรับปรุงแล้วในปี 1993 และปรับปรุงอีกครั้งในปี 1995 โดยการทำงานของ X.509 จะใช้การเข้ารหัสแบบกุญแจสาธารณะและใช้มาตรฐานลายมือชื่อดิจิตอลในการรับรองข้อมูล สำหรับอัลกอริทึมนั้นไม่ได้ระบุแน่นอนโดยสามารถเลือกใช้ได้หลายตัว แต่ที่แนะนำคือ RSA

ในการใช้งานเอกสารสิทธิ์จะมีช่วงเวลาใช้งานที่จำกัดแน่นอน ดังนั้นหากผู้ใช้ต้องการใช้เอกสารสิทธิ์ต่อไปก็จะต้องขอต่ออายุเอกสารสิทธิ์ก่อนที่จะหมดอายุ แต่หากมีการหมดอายุโดยที่ไม่ขอต่อหรือมีการเลิกใช้เอกสารสิทธิ์อาจจะเนื่องมาจากพนักงานลาออก หรืออาจจะเนื่องจากเอกสารสิทธิ์นี้ไม่ปลอดภัยแล้วก็ต้องทำการเรียกคืน(Revoke) และองค์กรพิสูจน์สิทธิ์จะต้องมีการจัดทำรายการเอกสารสิทธิ์ที่ถูกเรียกคืน (Certificate Revocation List – CRL) ซึ่งจะเก็บไว้ในไดเร็กทอรี่และรับรองโดยองค์กรพิสูจน์สิทธิ์ ซึ่งผู้ที่ต้องการตรวจสอบเอกสารสิทธิ์ว่าเป็นเอกสารสิทธิ์ที่ไม่ใช้งานแล้วหรือไม่ ก็ต้องขอรายการเอกสารสิทธิ์ที่ถูกเรียนคืนไปตรวจสอบ

และเนื่องจากเอกสารสิทธิ์ไม่สามารถปลอมแปลงได้ ดังนั้นการเก็บเอกสารสิทธิ์ไว้ที่องค์กรพิสูจน์สิทธิ์ จึงไม่ต้องมีกลไกพิเศษมาป้องกันแต่อย่างใด กล่าวคือผู้ใช้คนใดที่เป็นสมาชิกขององค์กรพิสูจน์สิทธิ์ก็สามารถเข้าถึงเอกสารสิทธิ์ของผู้ใช้คนอื่นๆได้ทุกคน โดยเอกสารสิทธิ์นี้จะเก็บอยู่ในไฟล์เพียงไฟล์เดียวที่มีขนาดเล็ก นอกจากจะสามารถขอเอกสารสิทธิ์จากองค์กรพิสูจน์สิทธิ์แล้ว ผู้ใช้ยังสามารถส่งเอกสารสิทธิ์ไปให้กันเองได้อีกด้วย โดยผ่านทางสื่อต่างๆ เช่น จดหมายอิเล็กทรอนิกส์ ส่งผ่านแผ่นดิสก์เก็ต เป็นต้น

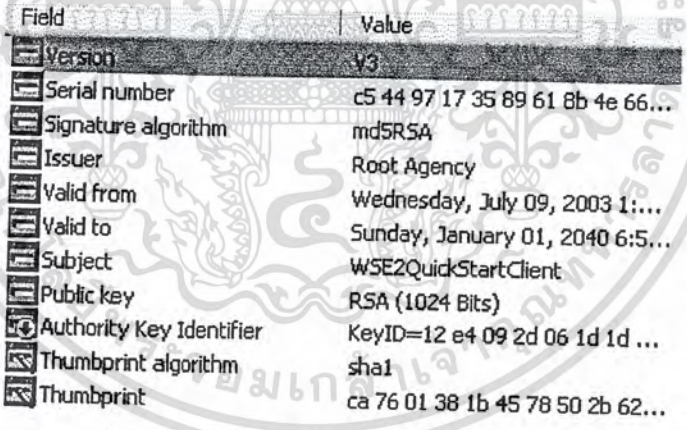
อย่างไรก็ตามเนื่องจากระบบเครือข่ายในปัจจุบันมีขนาดใหญ่โตกว้างขวางและการติดต่อสื่อสารก็ไม่ได้มีลักษณะเฉพาะกลุ่มอีกแล้ว ดังนั้นการที่จะให้ผู้ใช้ทุกคนมาใช้เอกสารสิทธิ์ที่รับรองโดยองค์กรพิสูจน์สิทธิ์เดียวกันทั้งหมดก็อาจเป็นเรื่องยาก หากผู้ใช้ 2 คน ใช้เอกสารสิทธิ์ที่รับรองจากองค์กรพิสูจน์สิทธิ์คนละแห่งก็จะไม่สามารถตรวจสอบเอกสารสิทธิ์ของอีกฝ่ายได้ว่าเป็นฉบับจริงหรือไม่ ซึ่งในกรณีเช่นนี้ก็อาจจะใช้วิธีสำเนากุญแจสาธารณะขององค์กรพิสูจน์สิทธิ์ของผู้ใช้อีกคนหนึ่งมาทำการตรวจสอบเองก็สามารถทำได้ เช่น กำหนดให้มี CA-A และ CA-B โดยให้บริการกับผู้ใช้ A และ B แต่เนื่องจากในครั้งแรกที่ A สำเนากุญแจสาธารณะของ CA-B มานั้นอาจเกิดการปลอมแปลงได้ เพราะสิ่งที่เรามีอยู่ก็คือกุญแจสาธารณะของ CA-A ของเรา แต่เอกสารสิทธิ์ของ CA-B ซึ่งบรรจุกุญแจสาธารณะของ CA-B นั้นจะรับรองการเข้ารหัสด้วยกุญแจส่วนตัวของ CA-B ทำให้เราไม่สามารถตรวจสอบว่าเอกสารสิทธิ์ที่ได้รับมานั้นเป็นฉบับที่ถูกต้องหรือไม่ ดังนั้นวิธีดังกล่าวจึงถือว่ามีความปลอดภัยไม่เพียงพอ

สำหรับอีกวิธีการอีกแบบ คือให้ CA-A เก็บเอกสารสิทธิ์ CA-B เอาไว้ด้วยและ CA-B ก็เก็บเอกสารสิทธิ์ของ CA-A เอาไว้เช่นกัน ด้วยวิธีนี้เราก็สามารถให้องค์กรพิสูจน์สิทธิ์ตรวจสอบเอกสารสิทธิ์ได้ไม่ว่าเอกสารสิทธิ์นั้นจะรับรองจาก CA-A หรือ CA-B ก็ตาม เช่นผู้ใช้ A ต้องการตรวจสอบเอกสารสิทธิ์ที่รับรองจาก CA-A ก็สามารถทำได้เลยเพราะรู้คีย์สาธารณะของ CA-A อยู่แล้ว เนื่องจากเป็นสมาชิกของ CA-A และหากผู้ใช้ A ต้องการตรวจสอบเอกสารสิทธิ์ที่รับรองโดย CA-B ผู้ใช้ A ก็ขอเอกสารสิทธิ์ของ CA-B จาก CA-A โดยเอกสารสิทธิ์ดังกล่าวจะรับรองโดย CA-A ดังนั้นจึ่งแน่ใจได้ว่าเอกสารสิทธิ์ของ CA-Bที่ได้รับนั้นเป็นของจริงและคีย์สาธารณะของ CA-B ก็เป็นของจริง

จากนั้นจึงนำเอาคีย์สาธารณะของ CA-Bไปตรวจสอบเอกสารสิทธิ์อีกทีก็จะทราบได้ว่าเอกสารสิทธิ์นั้นเป็นของจริงหรือไม่

และนอกเหนือจาก CA-A และ CA-B แล้วการเชื่อถือกันเช่นนี้ ยังสามารถกระทำกับองค์กรพิสูจน์สิทธิ์อื่นๆ ไปเรื่อยๆ อย่างไรก็ตาม หากองค์กรพิสูจน์สิทธิ์มีจำนวนมากๆ แล้ว ก็มีความจำเป็นที่จะต้องจัดโครงสร้างการเชื่อถือกันขององค์กรพิสูจน์สิทธิ์ให้เป็นระบบ ไม่เช่นนั้นก็อาจมีการเชื่อถือกันแบบยุ่งเหยิงและทำให้การทำงานเป็นไปอย่างไม่มีประสิทธิภาพได้

4.4 ส่วนประกอบของเอกสารสิทธิ์



รูปที่ 4.3 แสดงรายละเอียดของฟิลด์ในเอกสารสิทธิ์

1. เวอร์ชัน (Version) แสดงหมายเลขเวอร์ชันเพราะในแต่ละเวอร์ชันจะมีรูปแบบของข้อมูลที่ไม่เหมือนกันก็ได้ โดยปกติจะเป็นเวอร์ชัน 1 แต่หากในเอกสารสิทธิ์มีการใช้

2. หมายเลขลำดับ (Serial Number) เป็นเลขจำนวนเต็ม โดยจะต้องไม่ซ้ำกันในองค์กรที่ออกเอกสารสิทธิ์ โดยเลขนี้จะเป็นเลขที่จะใช้อ้างถึงแต่ละเอกสารสิทธิ์ที่สร้างขึ้นมา

3. อัลกอริทึมที่ใช้สร้าง (Signature Algorithm Identifier) เป็นฟิลด์ที่ระบุอัลกอริทึมที่ใช้ในการสร้างเอกสารสิทธิ์

4. ชื่อผู้ออกเอกสารสิทธิ์ (Issue Name) เป็นชื่อขององค์กรที่ออกเอกสารสิทธิ์

5. ช่วงเวลาที่รับรองเอกสารสิทธิ์ (Period of Validity) เป็นตัวบอกว่าให้ใช้เอกสารสิทธิ์นี้ตั้งแต่วันที่เท่าไรและสิ้นสุดวันที่เท่าไร

6. ชื่อเจ้าของเอกสารสิทธิ์ (Subject Name) เป็นชื่อของบุคคลที่เอกสารสิทธิ์ใบนี้อ้างถึงหรือแทนตัวบุคคลนั้น

7. ข้อมูลของกัญแจสาธารณะ (Subject's Public Key Information) เป็นฟิลด์ที่เก็บกุญแจสาธารณะและระบุถึงอัลกอริทึมที่ใช้กับกุญแจนี้ขึ้นมา รวมถึงพารามิเตอร์อื่นๆด้วย

8. ตัวระบุผู้ออกเอกสารสิทธิ์ (Issuer Unique Identifier) เป็นฟิลด์ออปชันที่ใช้ในการระบุถึงองค์กรที่ออกเอกสารสิทธิ์ ในกรณีที่ชื่อ X.500 มีการนำไปใช้กับส่วนอื่นๆ

9. ตัวระบุผู้ขอเอกสารสิทธิ์ (Subject Unique Identifier) เป็นฟิลด์ที่ใช้ในการระบุถึงตัวบุคคลที่เป็นเจ้าของเอกสารสิทธิ์ ในกรณีที่ชื่อ X.500 มีการนำไปใช้กับส่วนอื่นๆ

10. ส่วนขยาย (Extension) เป็นกลุ่มของฟิลด์ที่เพิ่มเติมข้อมูลอื่นๆ เข้ามาด้วย

11. ลายมือชื่อ (Signature) จะบรรจุเมสเซจไดเจสต์ของข้อมูลในทุกฟิลด์ที่เข้ารหัสด้วยกุญแจส่วนตัวขององค์กรพิสูจน์สิทธิ์ เพื่อเป็นการยืนยันว่าเอกสารสิทธิ์นี้สร้างมาจากองค์กรดังกล่าวจริงๆ โดยจะมีข้อมูลที่ระบุถึงวิธีการแฮชและวิธีการเข้ารหัสด้วย

บทที่ 5
ขั้นตอนและวิธีการดำเนินการวิจัย

5.1 ระบบสั่งซื้อคอมพิวเตอร์ออนไลน์

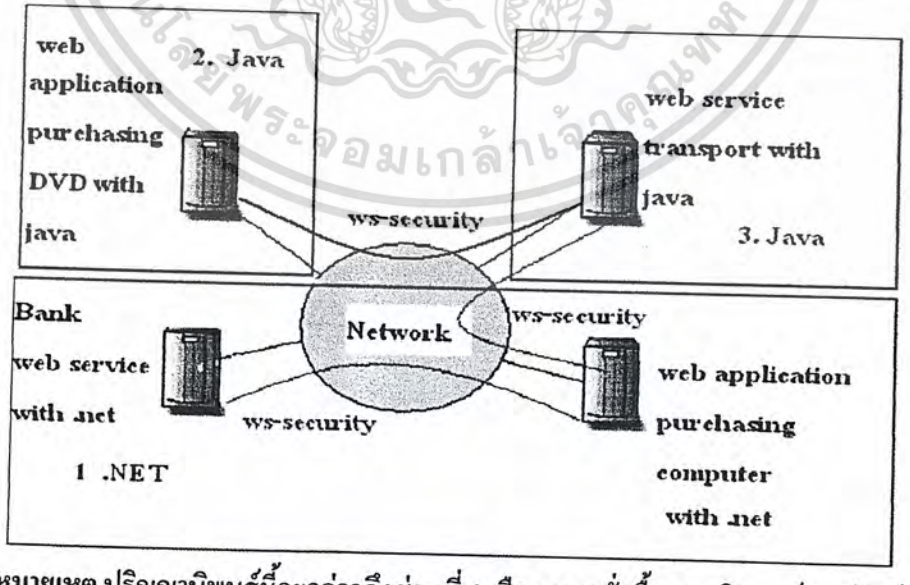
ระบบสั่งซื้อคอมพิวเตอร์ออนไลน์เป็นระบบให้บริการซื้อคอมพิวเตอร์ โดยเป็นระบบที่พัฒนาโดยใช้ asp.net และติดต่อกับเว็บเซอร์วิสในการชำระค่าสินค้า ซึ่งการติดต่อระหว่างแอพพลิเคชั่นกับเว็บเซอร์วิส นอกจากนี้ยังมีการติดต่อกับเว็บเซอร์วิสที่เป็นระบบขนส่ง ที่พัฒนาด้วยภาษา java และมีการประยุกต์ใช้โปรโตคอล ws-security ซึ่งเป็นโปรโตคอลที่เกิดขึ้นมาสำหรับเว็บเซอร์วิสโดยเฉพาะ

สำหรับระบบการสั่งซื้อสินค้าลูกค้าจะต้องเข้ามาเลือกสินค้า ที่ต้องการจะสั่งซื้อ แต่ก่อนการสั่งซื้อสินค้าจะต้องมีการลงทะเบียนลูกค้าเสียก่อนจึงจะสามารถสั่งซื้อได้ เมื่อลูกค้าชำระค่าบริการเรียบร้อยแล้วระบบเว็บเซอร์วิสที่ทำหน้าที่เกี่ยวกับการชำระเงินจะทำการเช็คบัตรเครดิต ถ้าบัตรถูกต้อง (Valid) ก็ทำธุรกรรมนี้ได้เสร็จสมบูรณ์

5.2 ส่วนค้นหาสินค้าในระบบสั่งซื้อคอมพิวเตอร์ออนไลน์

เป็นระบบที่มีการให้บริการค้นหาข้อมูล จะมีการแบ่งประเภทการค้นหาได้ตามชนิดของอุปกรณ์ เช่น ถ้าต้องการค้นหาเกี่ยวกับคอมพิวเตอร์โน๊ตบุ๊ก ก็สามารถที่จะเลือกค้นหายี่ห้อ ซีพียู หน่วยความจำ ฮาร์ดดิส ราคา ฯลฯ

5.3 โครงสร้างของระบบทั้งหมด



หมายเหตุ ปริญญานิพนธ์นี้จะกล่าวถึงส่วนที่ 1 คือ ระบบสั่งซื้อคอมพิวเตอร์ออนไลน์เท่านั้น

รูปที่ 5.1 แสดงภาพรวมของระบบทั้งหมด

5.4 วิธีการดำเนินงาน

วิธีการดำเนินงานมี 4 ขั้นตอนดังนี้

1) ขั้นตอนวิเคราะห์และออกแบบระบบสั่งซื้อคอมพิวเตอร์

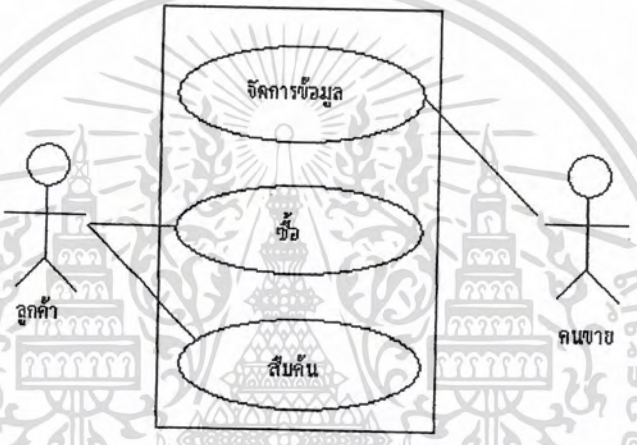
2) ขั้นตอนการสร้างเว็บแอพพลิเคชั่นของระบบสั่งซื้อคอมพิวเตอร์

3) ขั้นตอนการสร้างแบงค์เว็บเซอร์วิสและการเรียกใช้เว็บเซอร์วิส

4) ขั้นตอนการสร้างเว็บเซอร์วิสซีเคียวริตี้

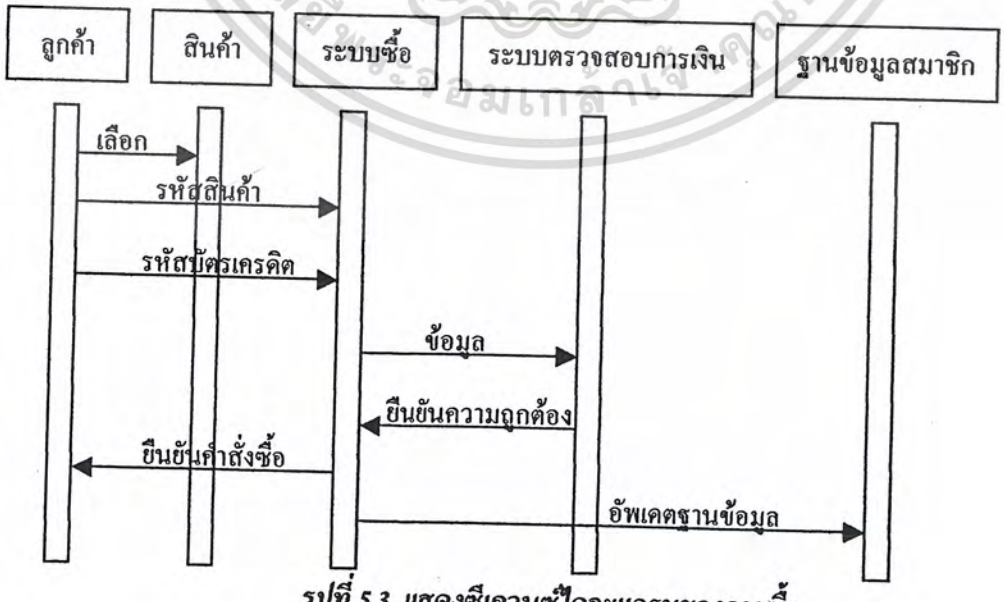
5.4.1 ขั้นตอนวิเคราะห์และออกแบบระบบสั่งซื้อคอมพิวเตอร์

5.4.1.1 แผนผังระบบ (Use case)



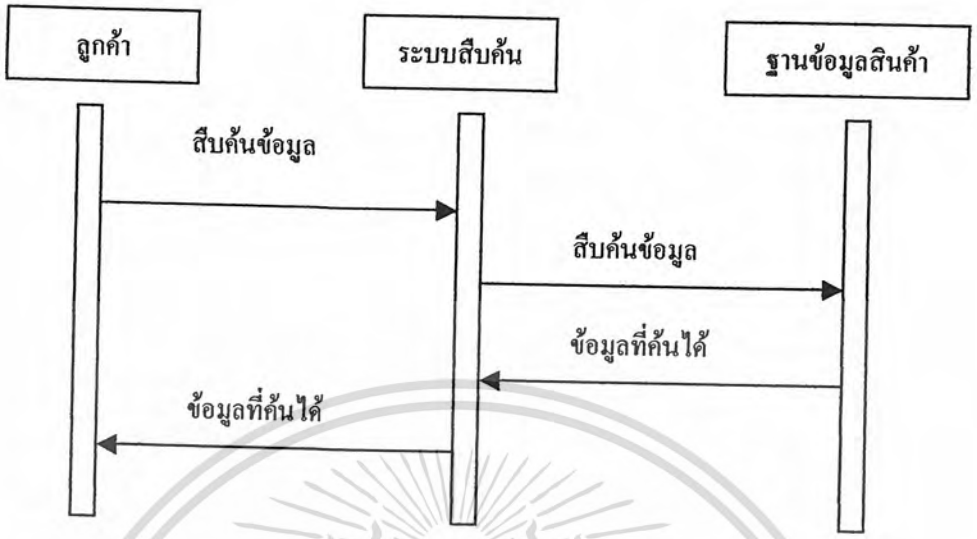
รูปที่ 5.2 แสดงแผนผังของระบบ(Use case)

5.4.1.2 ซีเควนซ์ไดอะแกรมของการซื้อ



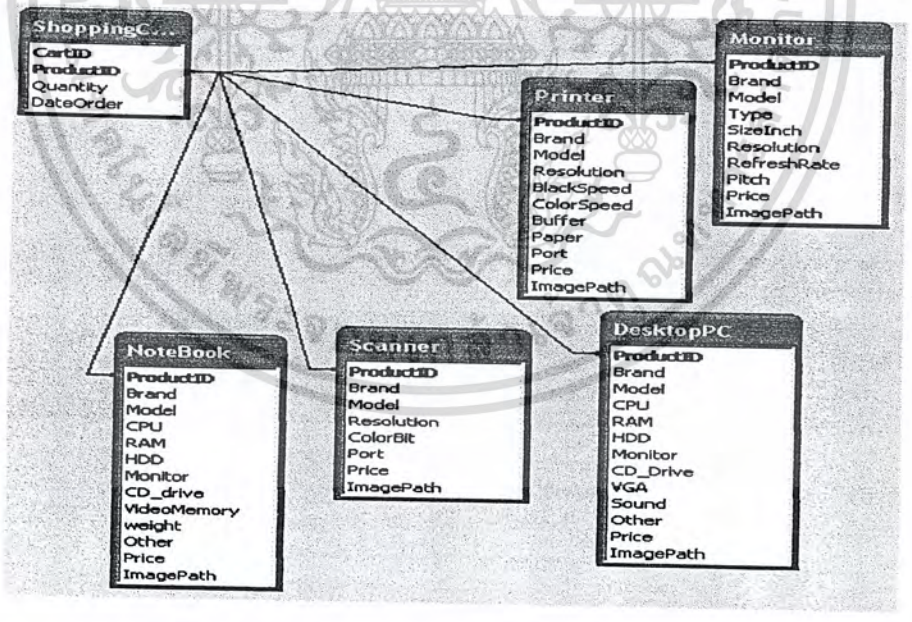
รูปที่ 5.3 แสดงซีเควนซ์ไดอะแกรมของการซื้อ

5.4.1.3 ซีเควนซ์ไดอะแกรมของการสืบค้น

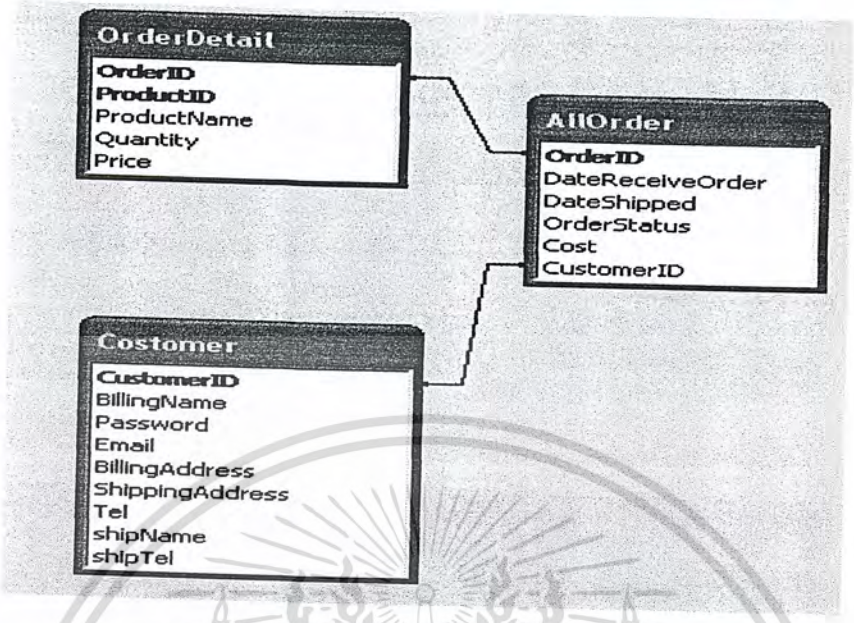


รูปที่ 5.4 แสดงซีเควนซ์ไดอะแกรมของการสืบค้น

5.4.1.4 ออกแบบดาต้าเบส



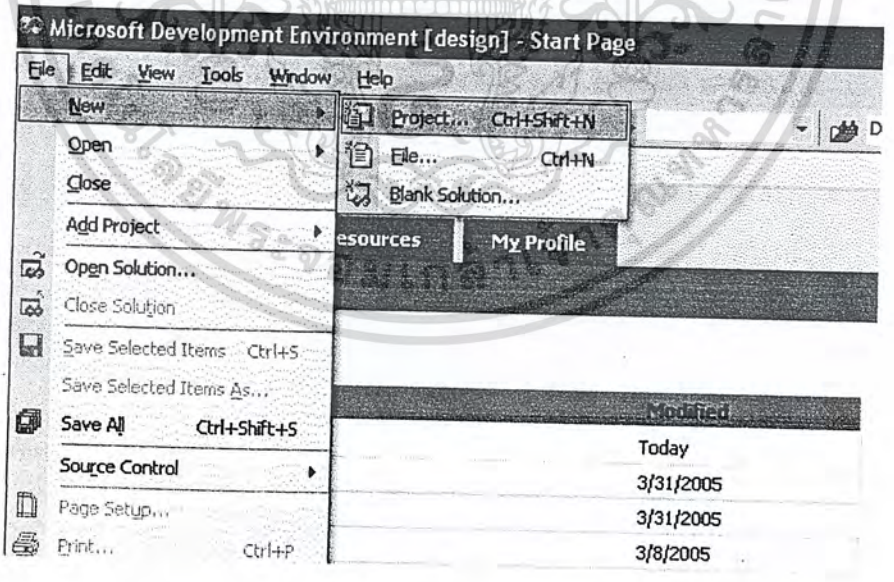
รูปที่ 5.5 แสดงการออกแบบดาต้าเบสส่วนที่ 1



รูปที่ 5.6 แสดงการออกแบบดาต้าเบสส่วนที่ 2

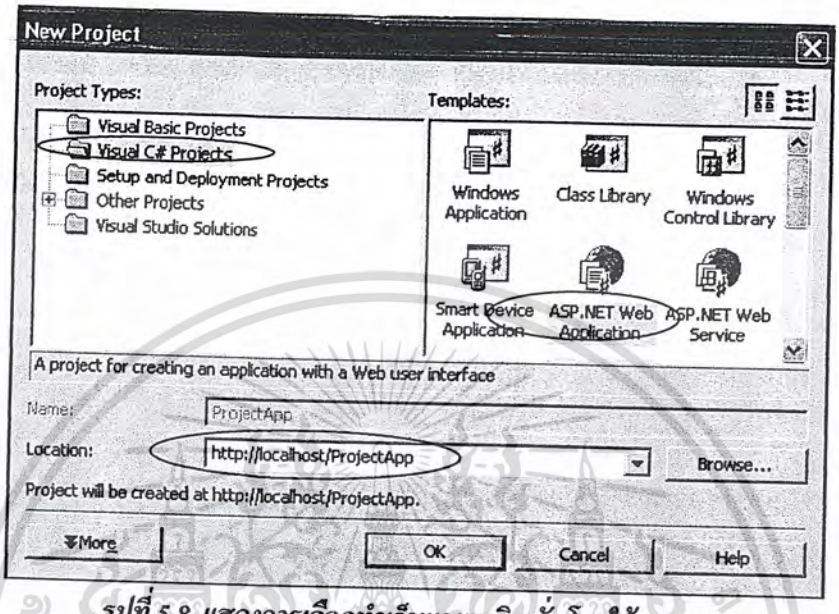
5.4.2 ขั้นตอนการสร้างเว็บแอพพลิเคชั่นของระบบสั่งซื้อคอมพิวเตอร์

1. เปิดโปรแกรม Visual Studio.NET และไปที่ File -> New -> Project แสดงดังรูปที่ 5.7

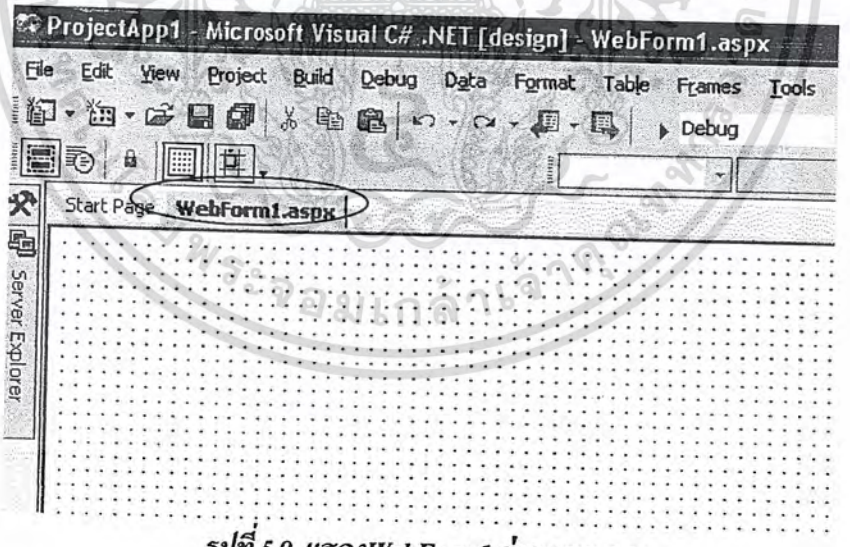


รูปที่ 5.7 แสดงการเริ่มสร้างโปรเจคใหม่

2. เลือกภาษาที่ใช้ในช่อง Project Types ในที่นี้เลือกภาษา C# และในช่อง Templates เลือก ASP.NET Web Application ที่ช่อง Location ให้ใส่ http://localhost/ProjectApp (ProjectApp เป็นชื่อที่เราตั้งขึ้น) แสดงดังรูปที่ 5.8 หลังจากกดปุ่ม OK จะได้รูปที่ 5.9

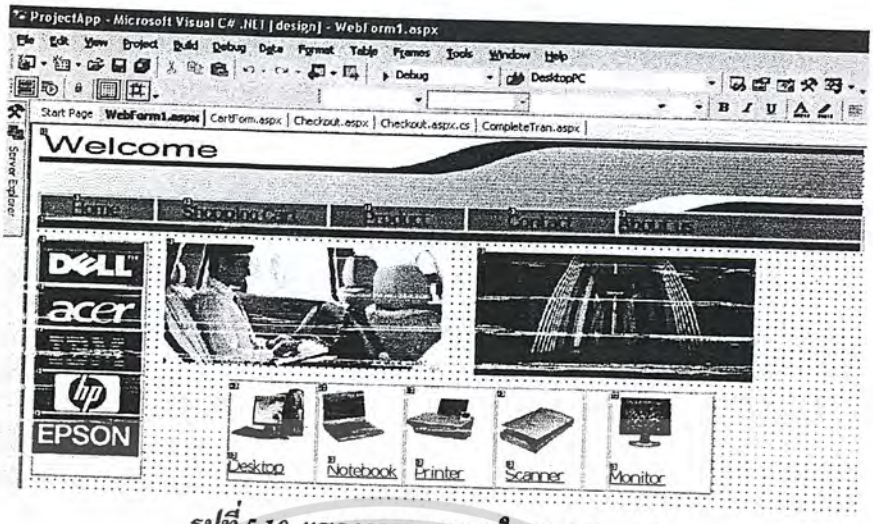


รูปที่ 5.8 แสดงการเลือกทำเว็บแอพพลิเคชั่นโดยใช้ภาษาC#



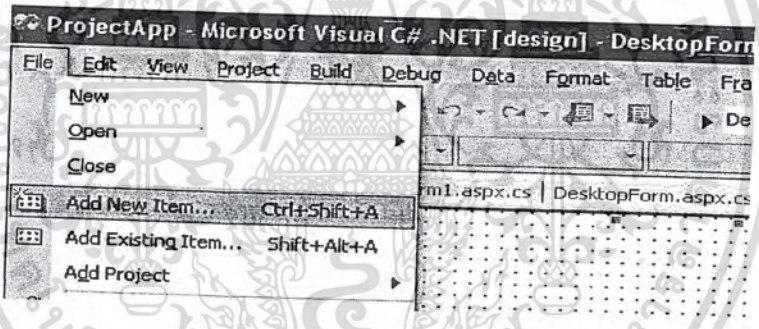
รูปที่ 5.9 แสดงWebForm1 ก่อนออกแบบ

3. เมื่อได้เว็บฟอร์มแล้วให้ทำการออกแบบหน้าเว็บแรกดังรูปที่ 5.10

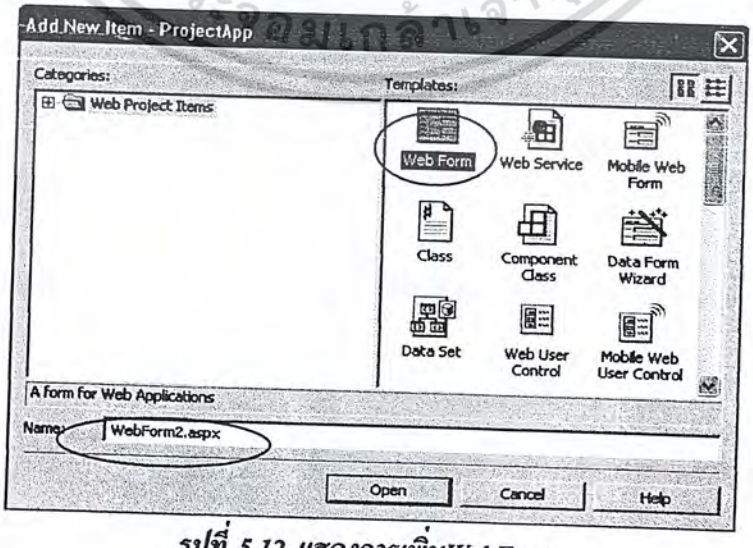


รูปที่ 5.10 แสดงการออกแบบใน WebForm1

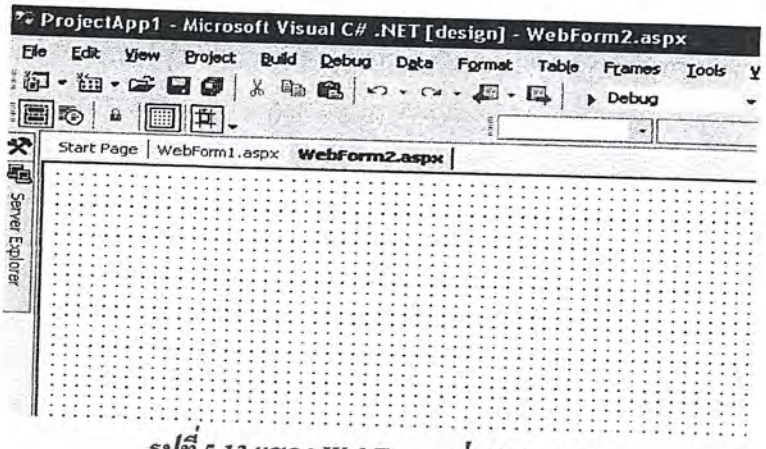
4. ถ้าหากเว็บไซต์ของเรามีหลายหน้าก็ต้องทำการสร้างเว็บฟอร์มขึ้นมาใหม่เพื่อให้เว็บฟอร์มอื่นสามารถอ้างอิงถึงได้ วิธีการสร้างเว็บฟอร์มใหม่ ให้ทำการคลิกที่ File -> Add New Item แสดงดังรูปที่ 5.11 แล้วจะได้รูปที่ 5.12 ตั้งชื่อเว็บฟอร์ม แล้วกดปุ่ม Open จะได้รูปที่ 5.13



รูปที่ 5.11 แสดงการเพิ่ม Item ใหม่



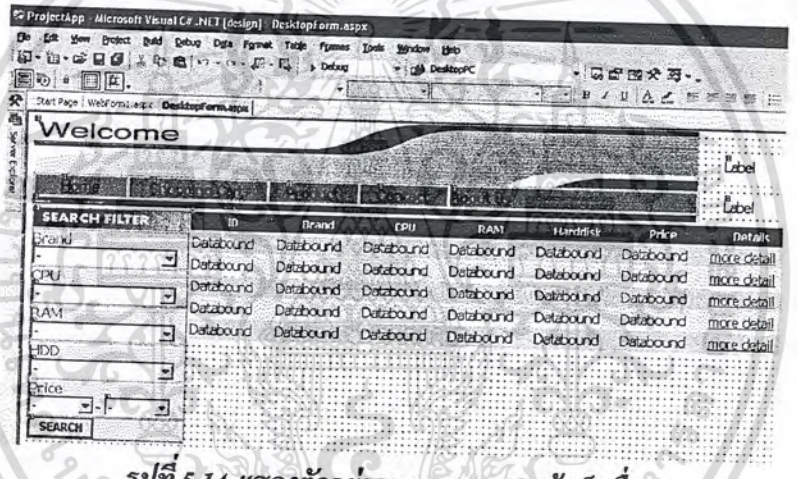
รูปที่ 5.12 แสดงการเพิ่มWebForm



รูปที่ 5.13 แสดง WebForm2 ก่อนออกแบบ

การสร้างหน้าเว็บเพจอื่นๆ ก็ทำในลักษณะเดียวกัน

5. เมื่อได้เว็บฟอร์มแล้วให้ทำการออกแบบหน้าเว็บอื่นดังรูปที่ 5.14



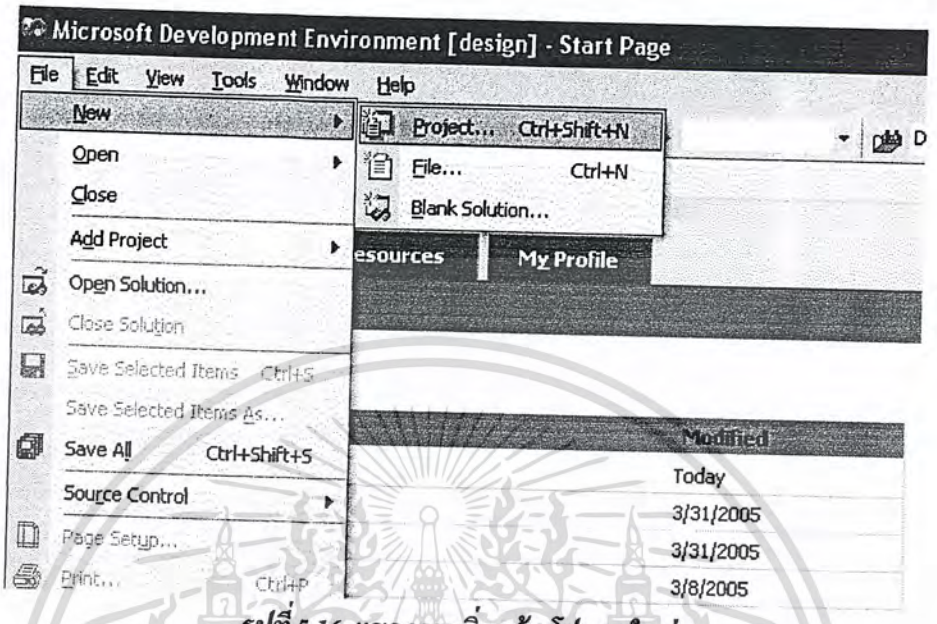
รูปที่ 5.14 แสดงตัวอย่างการออกแบบหน้าเว็บอื่น

6. หลังจากทำหน้าเว็บเพจต่างๆ เสร็จแล้วก็ไปที่ Build แล้วเลือก Build Solution เพื่อคอมไพล์ เมื่อคอมไพล์ผ่านแล้วให้ไปที่ debug แล้วเลือก start without debugging จะได้ผลลัพธ์ดังรูปที่ 5.15

รูปที่ 5.15 แสดงผลการรัน

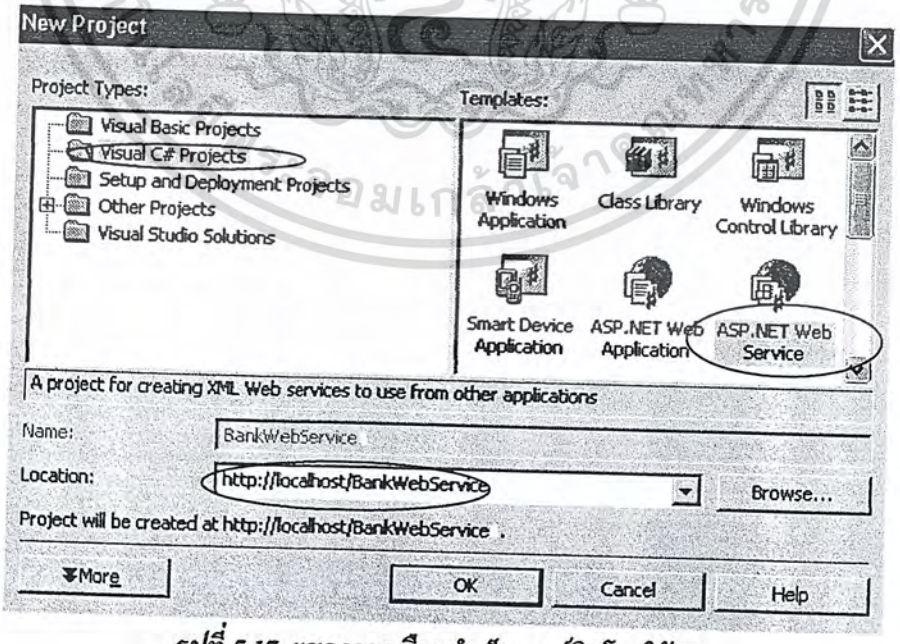
5.4.3 ขั้นตอนการสร้างเว็บเซอร์วิสและการเรียกใช้เว็บเซอร์วิส

1. เปิดโปรแกรม Visual Studio.NET และไปที่ File -> New -> Project แสดงดังรูปที่ 5.16



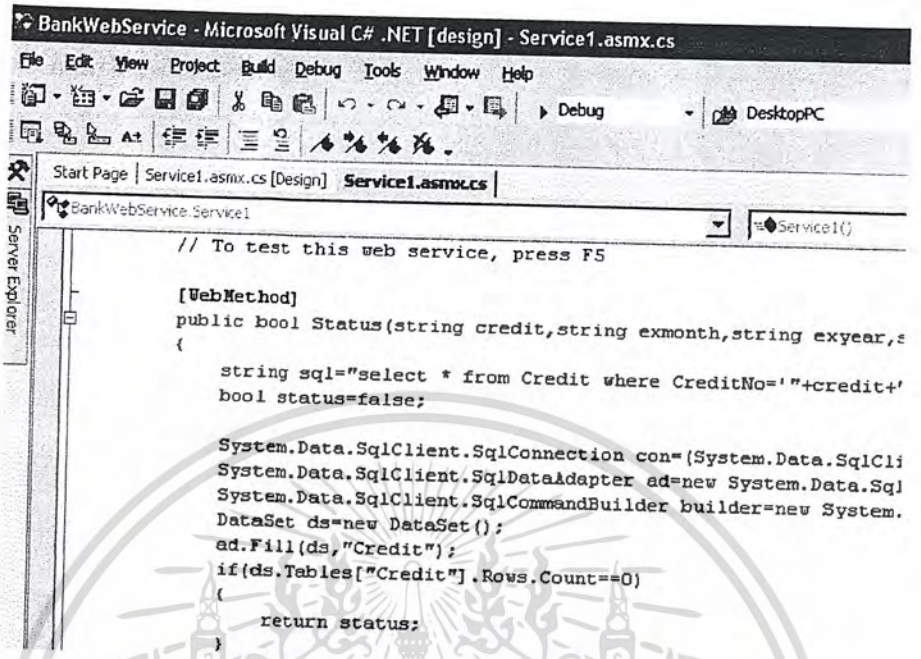
รูปที่ 5.16 แสดงการเริ่มสร้างโปรเจคใหม่

2. เลือกภาษาที่ใช้ในช่อง Project Types ในที่นี้เลือกภาษา C# และในช่อง Templates เลือก ASP.NET Web Service ที่ช่อง Location ให้ใส่ http://localhost/BankWebService (BankWebService เป็นชื่อที่เราตั้งขึ้น) แสดงดังรูปที่ 5.17



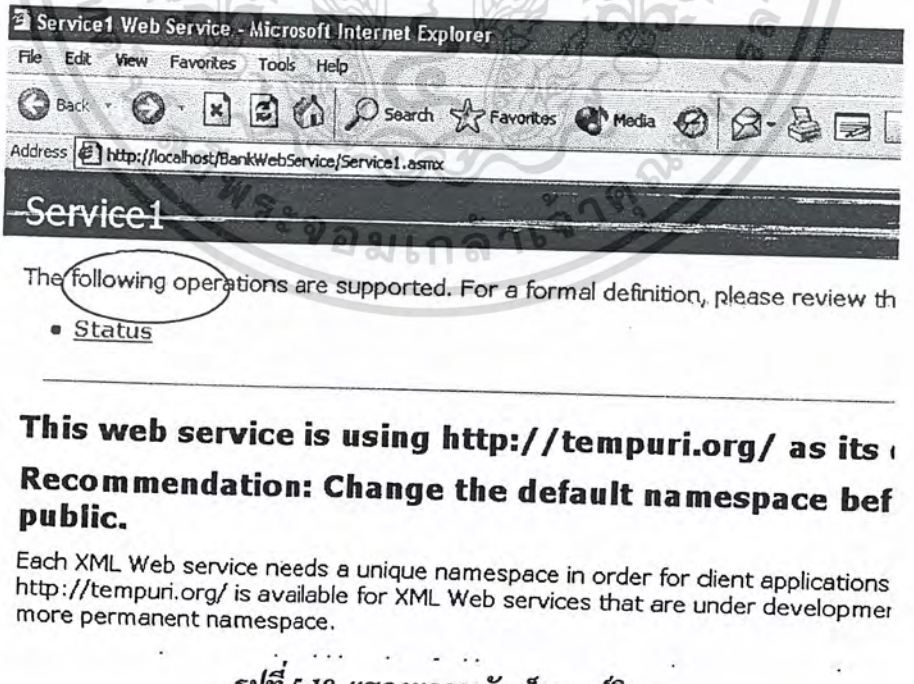
รูปที่ 5.17 แสดงการเลือกทำเว็บเซอร์วิสโดยใช้ C#

3. เขียนโค้ดของเว็บเซอร์วิส ดังรูปที่ 5.18



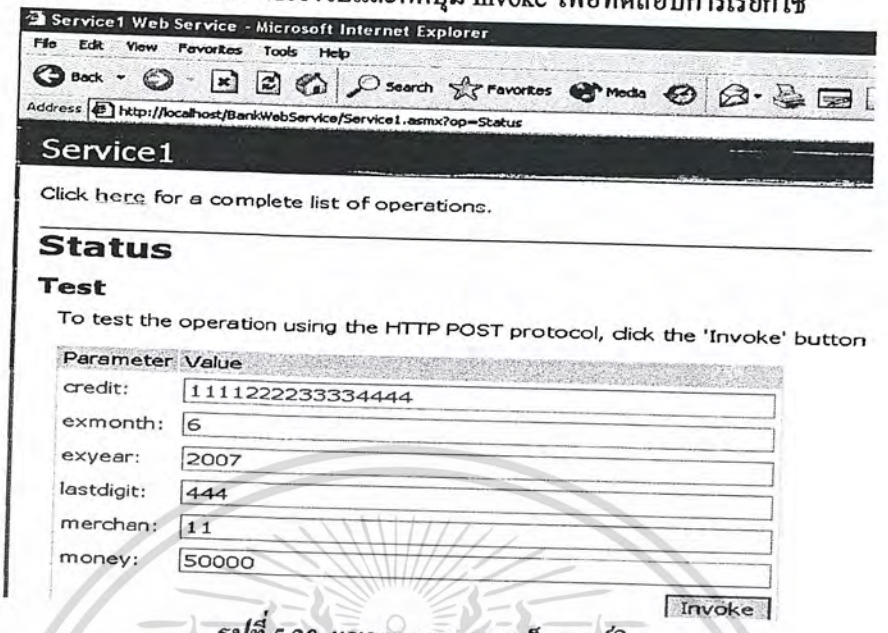
รูปที่ 5.18 แสดงโค้ดของแบงค์เว็บเซอร์วิส

3. เมื่อทำการคอมไพล์เว็บเซอร์วิสสำเร็จ ให้คลิกซ้ายที่ Debug จากนั้นเลือก Start without debugging จะได้ผลลัพธ์ดังรูปที่ 5.19 แล้วคลิก เมธอด Status จะได้รูปที่ 5.20



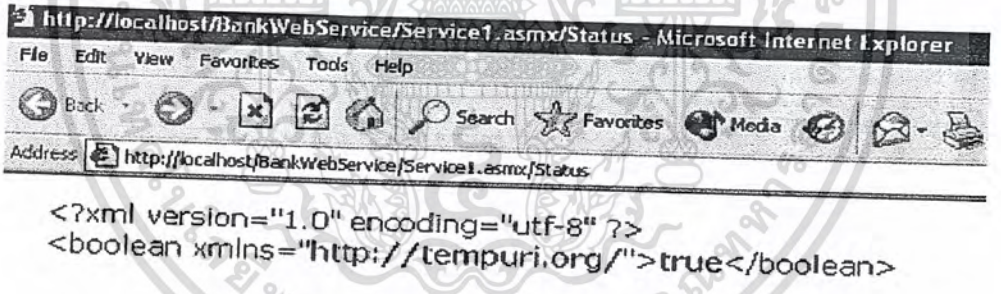
รูปที่ 5.19 แสดงผลการรันเว็บเซอร์วิส

4. จากรูปที่ 5.20 ให้ทำการใส่ผลลัพธ์เข้าไปและกดปุ่ม Invoke เพื่อทดสอบการเรียกใช้



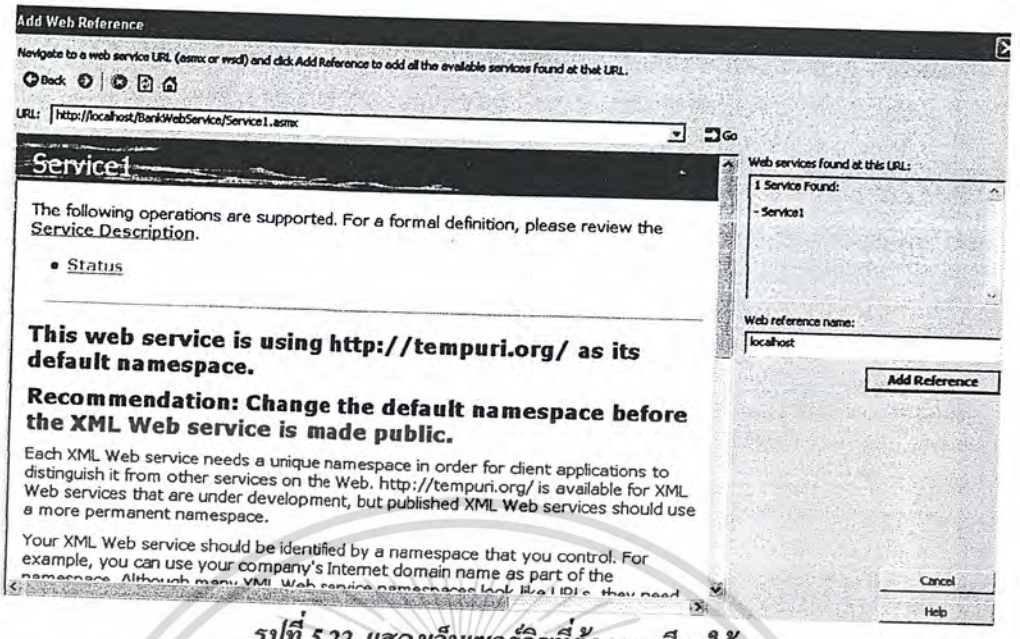
รูปที่ 5.20 แสดงการทดสอบเว็บเซอร์วิส

5. หลังจากที่มีการทดสอบเว็บเซอร์วิสแล้วและ เป็นบัตรเครดิตที่ สามารถใช้งานได้จะทำให้เว็บเซอร์วิสส่งค่ากลับมาเป็น true ดังรูปที่ 5.21 ก็เป็นอันเสร็จสิ้นในการสร้างเว็บเซอร์วิส



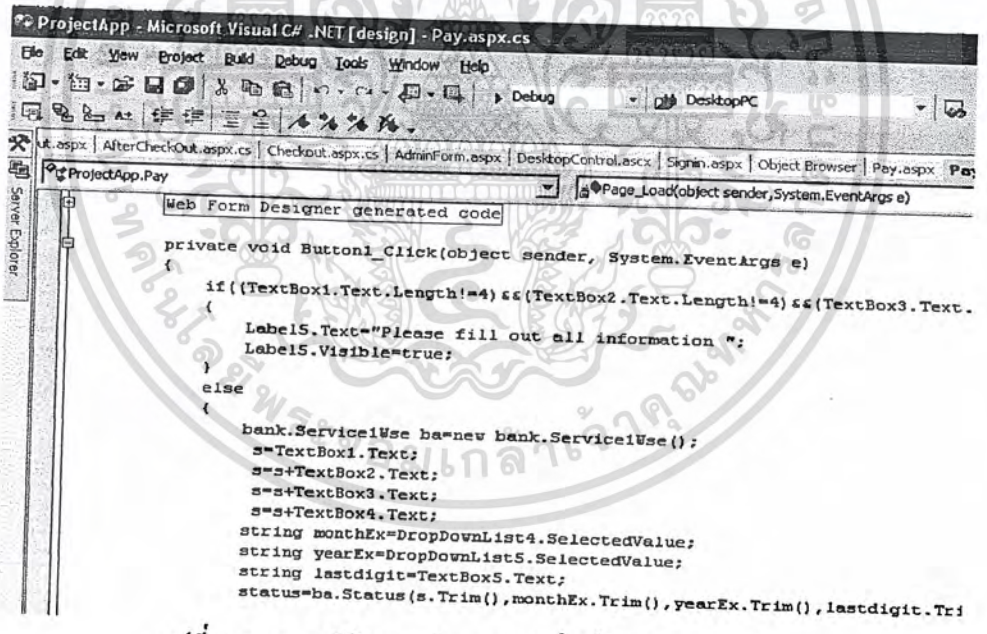
รูปที่ 5.21 แสดงผลการทดสอบ

6. เมื่อต้องการเรียกใช้เว็บเซอร์วิส ให้ทำการเปิดหรือสร้างแอพพลิเคชั่นขึ้นมาแล้วทำการเรียกเว็บเซอร์วิส โดยเลือก Project->Add Web Reference.. และเลือก Service ที่ต้องการเรียก ดังแสดงรูปที่ 5.22 และกดปุ่ม Add Reference



รูปที่ 5.22 แสดงเว็บเซอร์วิสที่ต้องการเรียกใช้

7. เขียนโค้ดที่มีการเรียกใช้ BankWebService ดังแสดงในรูปที่ 5.23



รูปที่ 5.23 แสดงโค้ดของเว็บแอพลิเคชั่นที่อ้างถึงเว็บเซอร์วิส

5.4.4 ขั้นตอนการสร้างเว็บเซอร์วิสซีเคียวริตี้

ดาวน์โหลดโปรแกรมที่ชื่อว่า WSE2.0 SP2 จากเว็บไซต์ของไมโครซอฟต์ ทำการติดตั้งโปรแกรมในการติดตั้งให้เลือกการติดตั้งเป็นแบบ Developer ในการทำ เว็บเซอร์วิสซีเคียวริตี้ มี 2 วิธี

1. วิธีการใช้ไฟล์โพลิซี(policy file)

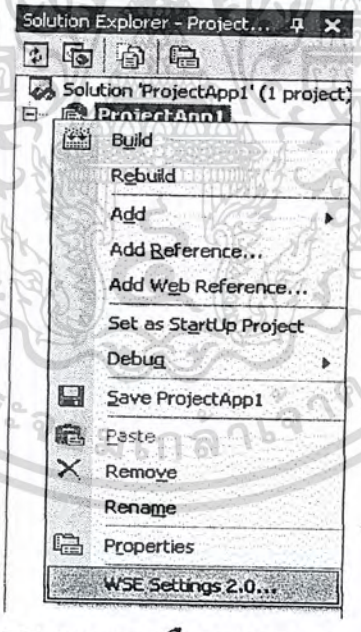
2. วิธีการเขียนโปรแกรมโดยการอ้างถึงเนมสเปส(namespace)ที่เกี่ยวกับเว็บเซอร์วิสซีเคียวริตี้

แต่ละวิธีมีข้อดีข้อเสียแตกต่างกันคือ การเขียนโปรแกรมโดยการอ้างถึงเนมสเปซที่เกี่ยวกับเว็บเซอร์วิสซีเคียวริตี้ มีข้อเสียตรงที่ต้องยุ่งยากในการเขียนโปรแกรม ส่วนข้อดีคือสามารถทำการเข้ารหัสและทำลายเซ็นอิเล็กทรอนิกส์ในบางอีลีเมนต์ของ SOAP เมสเสจได้ ส่วนการใช้ ไฟล์โพลิซีนั้นมีข้อดีตรงที่ง่ายต่อการใช้งาน ส่วนข้อเสียคือไม่สามารถที่จะทำการเข้ารหัสหรือทำลายเซ็นอิเล็กทรอนิกส์เพียงบางส่วนได้

5.4.4.1 วิธีการใช้ไฟล์โพลิซี(policy file)

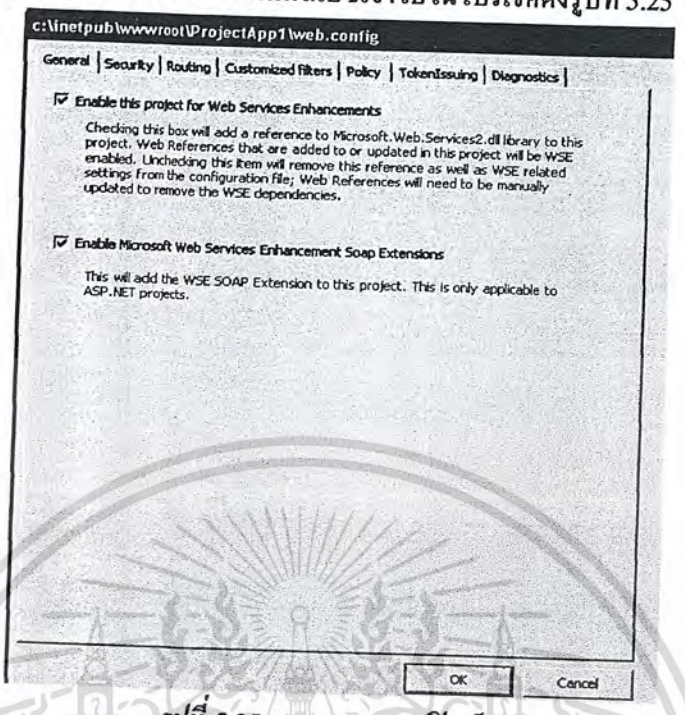
จากแอพพลิเคชั่นที่ได้ทำไว้แล้วให้ทำตามขั้นตอนดังนี้

1. ไปที่เว็บแอพพลิเคชั่น คลิกขวาที่ชื่อโปรเจคในช่อง Solution Explorer ดังรูปที่ 5.24 แล้วเลือกที่ wse setting 2.0 จะปรากฏไดอะล็อกดังรูปที่ 5.25



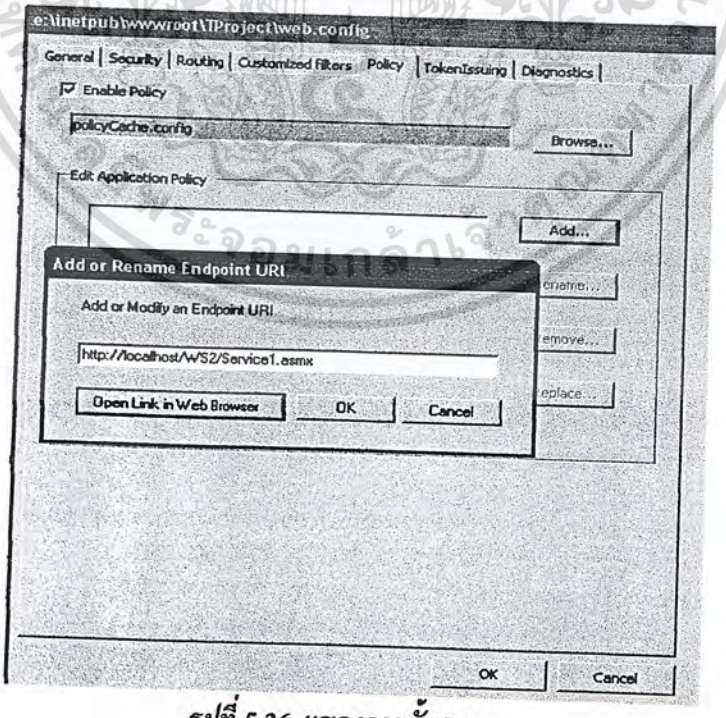
รูปที่ 5.24 แสดงการเลือก WSE Settings 2.0

2. ทำการเลือกที่เช็กบอกซ์ เพื่อให้มีการเพิ่มเนมสเปซเข้าไปในโปรเจคดังรูปที่ 5.25



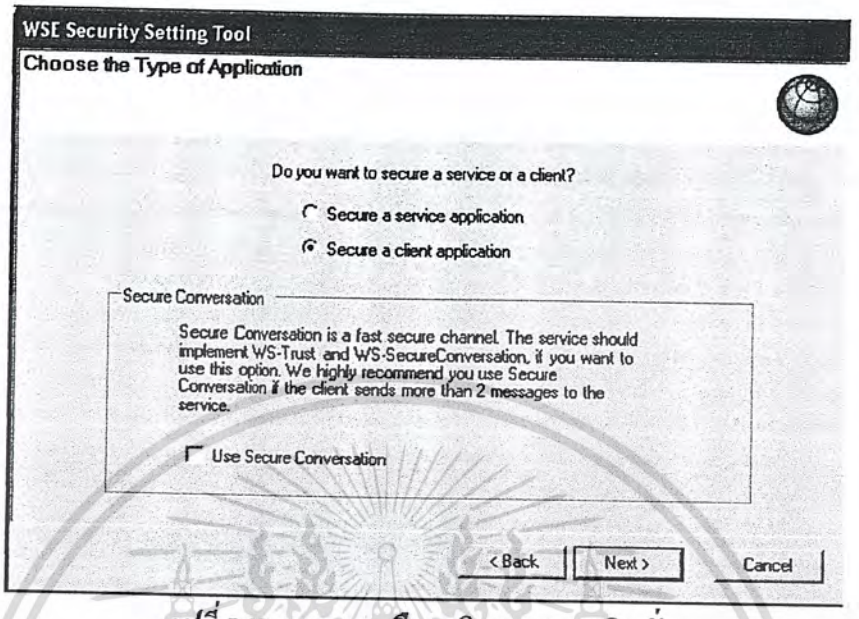
รูปที่ 5.25 แสดงการคอนฟิกเว็บ

3. คลิกที่แท็บ Policy คลิกเช็คบ็อกซ์หน้า Enable Policy กดปุ่ม Add... จะปรากฏไดอะล็อกบ็อกซื่อ Add or Rename Endpoint URI แล้วใส่ชื่อเว็บเซอร์วิสที่เราต้องการใช้ลงไป ดังรูปที่ 5.26 แล้วกด OK แล้วจะเข้าสู่ WSE Security Setting Tool



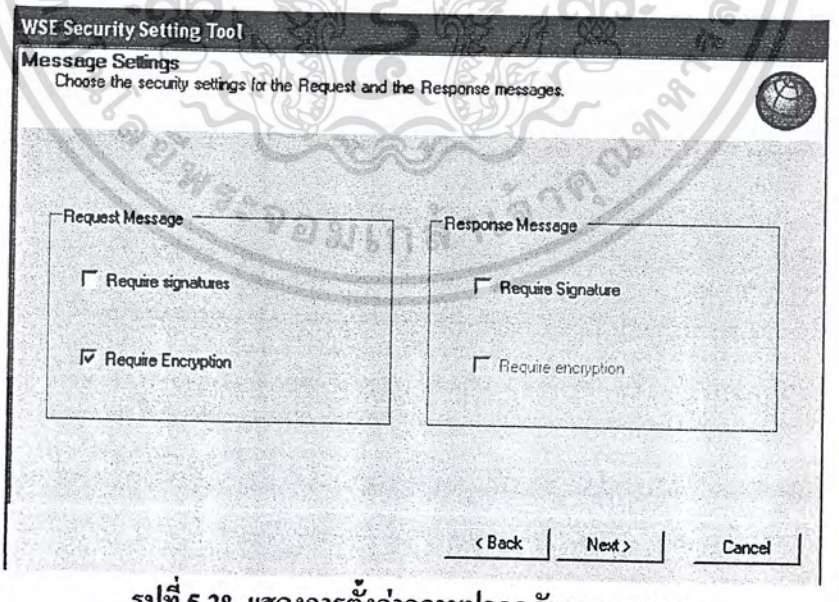
รูปที่ 5.26 แสดงการตั้ง Policy

4. เมื่อเข้าสู่ WSE Security Setting Tool แล้วคลิก Next จะได้รูปที่ 5.27 โดยเลือก Secure a client Application แล้วคลิก Next



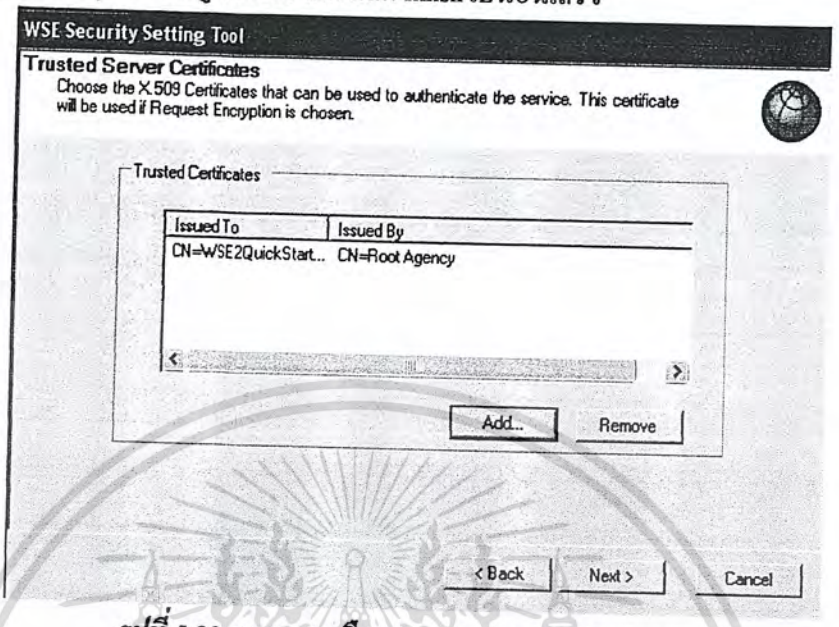
รูปที่ 5.27 แสดงการเลือกชนิดของแอพพลิเคชั่น

5. หลังจากนั้นเลือกการเข้ารหัสจาก ไคล์เอนไปยัง เว็บเซอร์วิส โดยมีการเช็กบอกซ์ที่ Require Encryption ดังรูปที่ 5.28

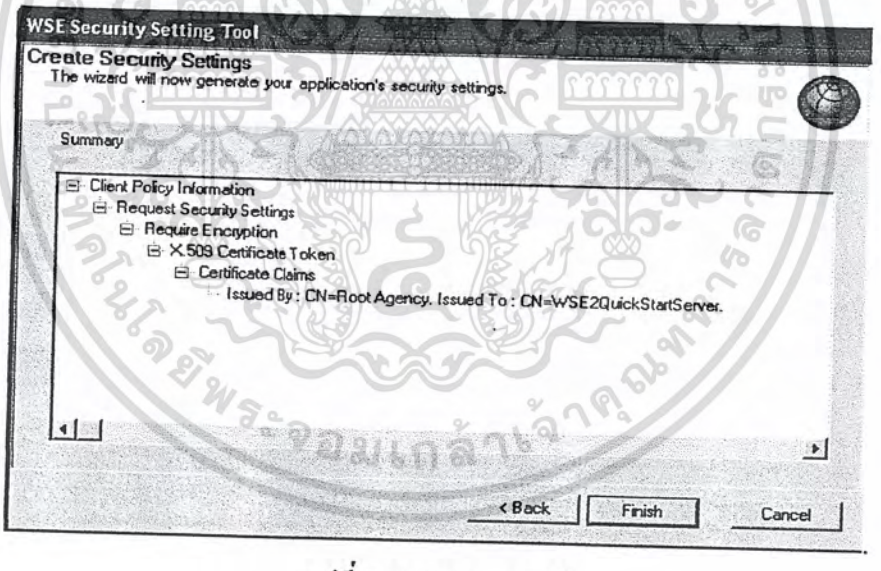


รูปที่ 5.28 แสดงการตั้งค่าความปลอดภัยของเมสเสจ

6. จากนั้นทำการเลือกเอกสารสิทธิ์ของฝั่งเว็บเซอร์วิสที่จะใช้ในการเข้ารหัส ดังรูปที่ 5.29 คลิก Next แล้วจะเข้าสู่ไดอะล็อกสุดท้าย ดังรูปที่ 5.30 และคลิก finish เป็นอันเสร็จ



รูปที่ 5.29 แสดงการเลือก Trust Server Certificates



รูปที่ 5.30 แสดงผลสรุป

ที่โพลิซีแคชจะมีการปรับแต่งไฟล์โดยมีข้อความดังต่อไปนี้

```
<?xml version="1.0" encoding="utf-8"?>
<policyDocument xmlns="http://schemas.microsoft.com/wse/2003/06/Policy">
 <mappings xmlns:wse="http://schemas.microsoft.com/wse/2003/06/Policy">
  <!--The following policy describes the policy requirements for the service:
http://localhost/WS2/Service1.asmx .-->
   <endpoint uri="http://localhost/WS2/Service1.asmx">
```

```
      <defaultOperation>
       <request policy="#Encrypt-X.509" />
       <response policy="" />
       <fault policy="" />
      </defaultOperation>
     </endpoint>
    </mappings>
    <policies xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy"
    xmlns:wssp="http://schemas.xmlsoap.org/ws/2002/12/secext"
    xmlns:wse="http://schemas.microsoft.com/wse/2003/06/Policy" xmlns:wssc="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
     <wsp:Policy wsu:Id="Encrypt-X.509">
      <!--The Confidentiality assertion is used to ensure that the SOAP Body is encrypted.-->
      <wssp:Confidentiality wsp:Usage="wsp:Required">
       <wssp:KeyInfo>
        <!--The SecurityToken element within the KeyInfo element describes which token type must be used for Encryption.-->
        <wssp:SecurityToken>
         <wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</wssp:TokenType>
         <wssp:TokenIssuer>CN=Root Agency</wssp:TokenIssuer>
         <wssp:Claims>
          <!--By specifying the SubjectName claim, the policy system can look for a certificate with this subject name in the certificate store indicated in the application's configuration, such as LocalMachine or CurrentUser. The WSE X.509 Certificate Tool is useful for finding the correct values for this field.-->
          <wssp:SubjectName
    MatchType="wssp:Exact">CN=WSE2QuickStartServer</wssp:SubjectName>
          <wssp:X509Extension OID="2.5.29.14"
    MatchType="wssp:Exact">bBwPfltvKp3b6TNDq+14qs58VJQ=</wssp:X509Extension>
         </wssp:Claims>
        </wssp:SecurityToken>
       </wssp:KeyInfo>
```

```
    <wssp:MessageParts
Dialect="http://schemas.xmlsoap.org/2002/12/wsse#part">wsp:Body()</wssp:MessageParts>
    </wssp:Confidentiality>
  </wsp:Policy>
 </policies>
</policyDocument>
```

จะเห็นว่าที่โพลิซีไฟล์นั้นเราไม่ต้องเขียนโปรแกรมเพิ่มเติม เพียงแต่ต้องเรียนรู้การใช้เครื่องมือในการสร้างโพลิซีซึ่งทำให้ประหยัดเวลาในการทำงานเป็นอย่างมาก

จากโพลิซีไฟล์ข้างบนพอจะสรุปได้ว่า

<wssp:Confidentiality wsp:Usage="wsp:Required"> เป็นการบอกว่ามีการเข้ารหัสข้อมูล

<wssp:TokenType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3</wssp:TokenType> เป็นการบอกว่าโทเคนที่ใช้เป็น x509

<wssp:TokenIssuer>CN=Root Agency</wssp:TokenIssuer> บอกว่าผู้ออกใบรับรองนี้คือ Root Agency

<wssp:SubjectName MatchType="wssp:Exact">CN=WSE2QuickStartServer</wssp:SubjectName>
<wssp:X509Extension OID="2.5.29.14"
MatchType="wssp:Exact">bBwPfltvKp3b6TNDq+14qs58VJQ=</wssp:X509Extension>
เป็นการบอกว่าใบรับรองนี้ชื่อ wse2quickstartserver และมีหมายเลขประจำตัวเป็น bBwPfltvKp3b6TNDq+14qs58VJQ= ซึ่งทุกครั้งที่มีการเข้ารหัสและถอดรหัสจะต้องมีการอ้างถึง หมายเลขนี้เพื่อจับคู่ระหว่าง ใบรับรองสาธารณะ กับ ใบรับรองที่เป็นส่วนตัว

5.4.4.2 วิธีการเขียนโปรแกรมโดยการอ้างถึงเนมสเปส

1.จัดเตรียมทำเว็บแอพพลิเคชั่น และ เว็บเซอร์วิส โดยที่เว็บแอพพลิเคชั่นต้องสามารถติดต่อกับเว็บเซอร์วิสได้

2. ทำการเพิ่มเนมสเปซที่ส่วนหัวของเว็บแอพพลิเคชั่น

```
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Security;
using Microsoft.Web.Services2.Security.Tokens;
using Microsoft.Web.Services2.Security.X509;
```

3. สร้างเมธอดใช้ในการรับค่าซีเคียวริตี้โทเคน

```
public X509SecurityToken GetToken()
{
X509SecurityToken sec=null;
X509CertificateStore
store=X509CertificateStore.CurrentUserStore(X509CertificateStore.MyStore);
bool open=store.OpenRead();
X509CertificateCollection
certs=store.FindCertificateBySubjectString("WSE2QuickStartServer");
X509Certificate cert=(X509Certificate)certs[0];
sec=new X509SecurityToken(cert);
return sec;
}
```

4. ที่ Solution Explorer ให้คลิกที่ Show All File จากนั้นไปยัง web reference ดับเบิลคลิกที่ localhost (ในกรณีนี้ตอน add web reference ใช้ชื่อเป็น localhost) ดับเบิลคลิกที่ reference.map จะพบกับไฟล์ reference.cs เปลี่ยน

```
public class Service1 : System.Web.Services.Protocols.SoapHttpClientProtocol
```

ให้เป็น

```
public class Service1 : Microsoft.Web.Services2.WebServicesClientProtocol
```

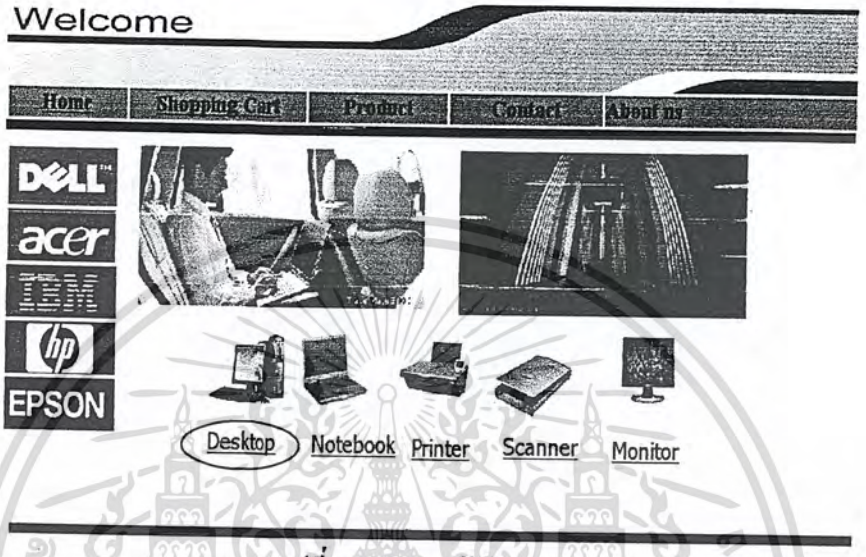
5. ในการเรียกใช้งานถ้าต้องการจะทำเว็บเซอร์วิสซีเคียวริตี้จะต้องเขียนโปรแกรมดังต่อไปนี้ก่อนที่จะทำการเรียกเว็บเซอร์วิส

```
X509SecurityToken sec=null; // กำหนดให้ซีเคียวริตี้โทเคนเป็นนัล
localhost.Service1Wse s=new localhost.Service1Wse();//สร้างอินสแตนของเว็บเซอร์วิส
SoapContext request=s.RequestSoapContext;//ทำการร้องขอ soap
request.Security.Tokens.Add(sec);//เป็นการกำหนดว่าจะใช้โทเคนนี้ในการทำซีเคียวริตี้
//MessageSignature mes=new MessageSignature(sec); กรณีที่จะทำ ดิติตอลซิกเนเจอร์
EncryptedDate enc=new EncryptedData(sec); //กำหนดให้การเข้ารหัสใช้โทเคนนี้
request.Security.Elements.Add(end); // เพิ่มซีเคียวริตี้เข้าไปใน soap
request.Security.Timestamp.TtlInSeconds=120; //กำหนด ทามเอาต์(time out)
s.HelloWorld(); เรียกเซอร์วิส โดยมีการเข้ารหัสข้อความก่อนที่จะส่งไปยังผู้ให้บริการ
```

บทที่ 6

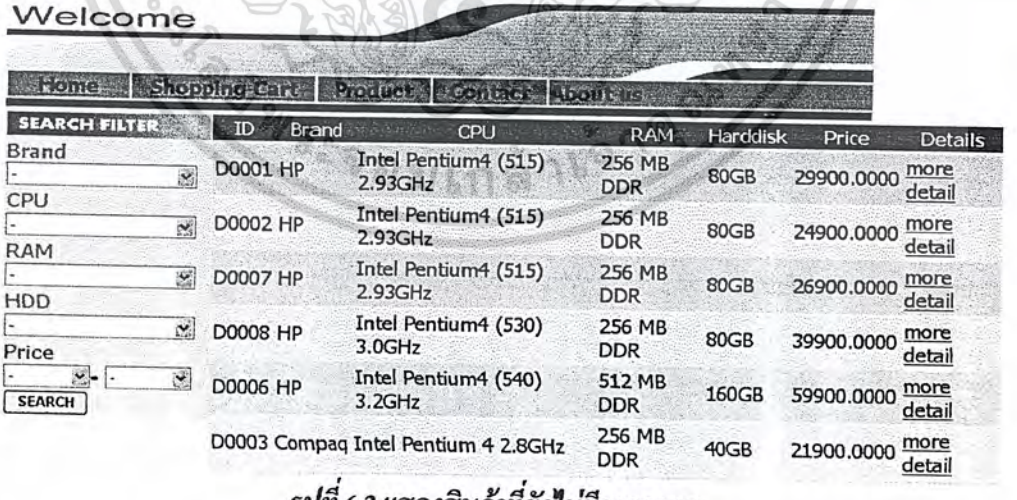
การทดลองและผลการทดลอง

1. เมื่อลูกค้าเข้าสู่หน้าเว็บเพจจะมีสินค้าให้เลือกอยู่ 5 ประเภท ดังรูปที่ 6.1 ในที่นี้เลือก Desktop

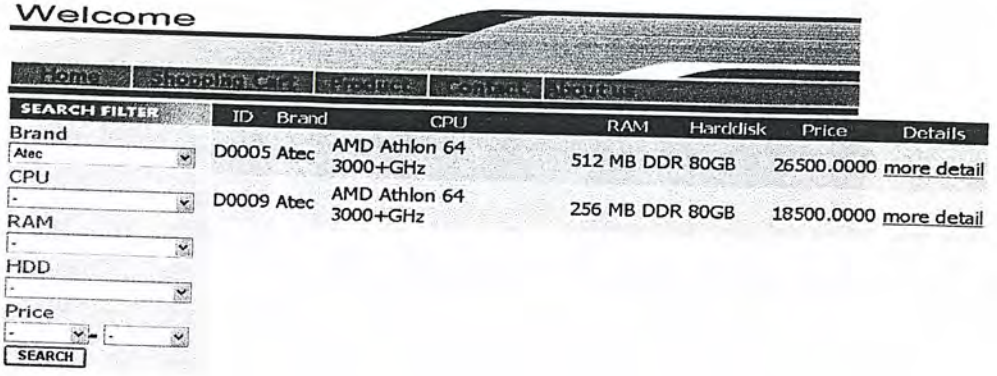


รูปที่ 6.1 แสดงหน้าโฮมเพจ

2. เมื่อเลือกสินค้าประเภท Desktop ซึ่งแสดงรายละเอียดสินค้าอย่างย่อ ดังรูปที่ 6.2 ถ้าหากลูกค้าต้องการค้นหาให้คลิก Search จะได้ข้อมูลที่ต้องการดังรูปที่ 6.3 (ในที่นี้ต้องการ Brand : Atec)

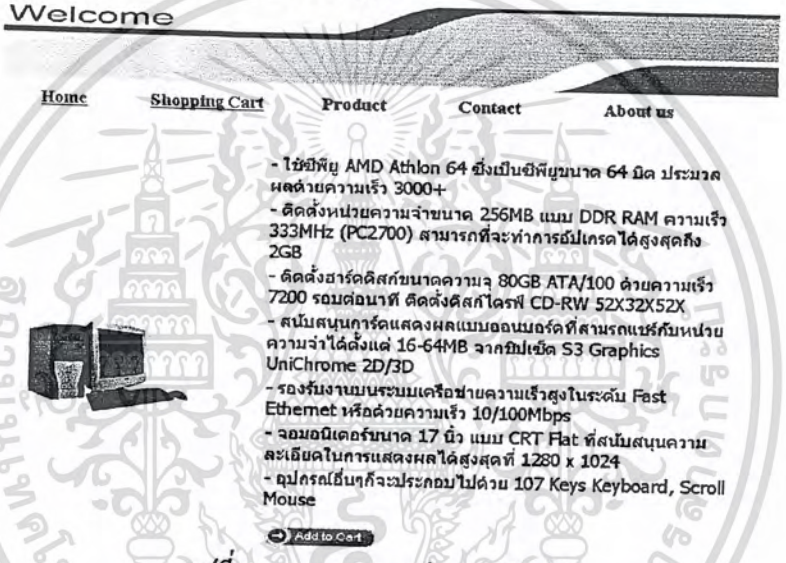


รูปที่ 6.2 แสดงสินค้าที่ยังไม่มีการกรอง



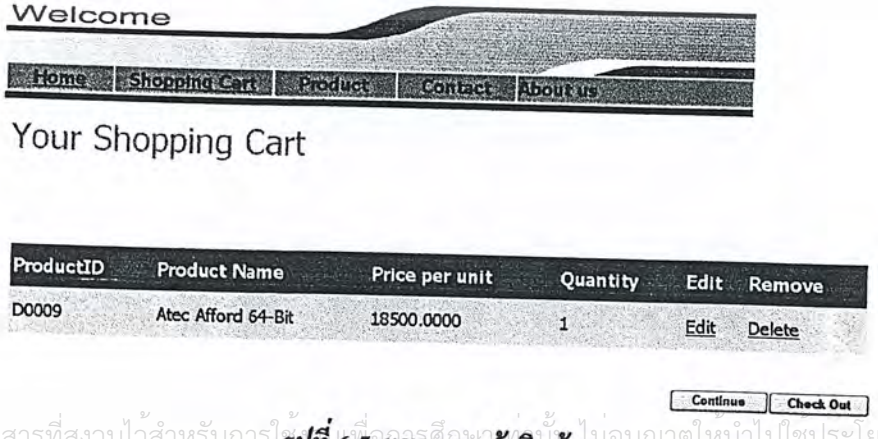
รูปที่ 6.3 แสดงสินค้าที่ผ่านการกรองแล้ว

3. ดูรายละเอียดสินค้าให้คลิกที่ more detail จะได้รูปที่ 6.4



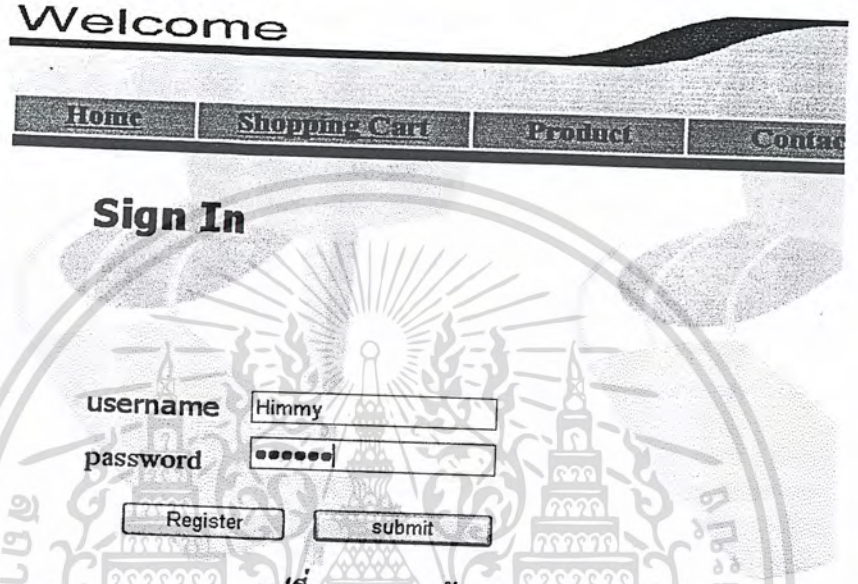
รูปที่ 6.4 แสดงรายละเอียดของสินค้า

4. เมื่อตัดสินใจเลือกสินค้าตัวนี้ลงตะกร้าให้คลิกปุ่ม Add to Cart ในรูปที่ 6.4 จะได้รูปที่ 6.5 ซึ่งเป็นตะกร้าสินค้า (Shopping Cart) สามารถแก้ไขจำนวนสินค้าที่ต้องการได้โดยทำการกด Edit หรือสามารถเอาสินค้าออก โดยกด delete และถ้าต้องการซื้อสินค้าต่อให้คลิก Continue หากต้องการจ่ายเงินให้คลิก Check out ในรูปที่ 6.5

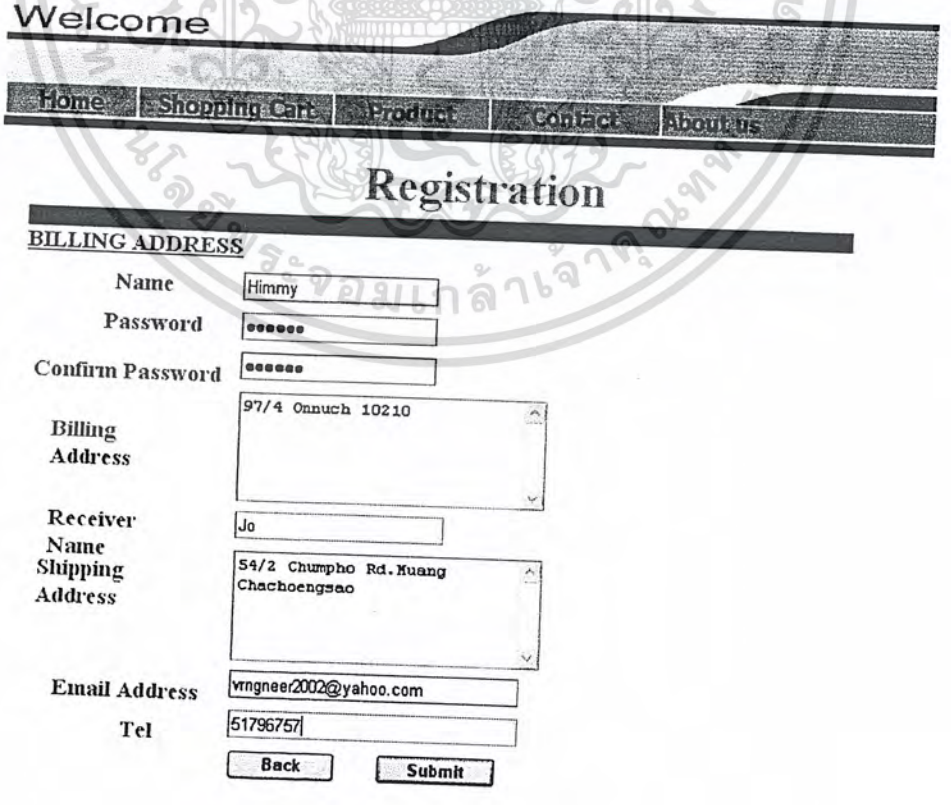


รูปที่ 6.5 แสดงตะกร้าสินค้า

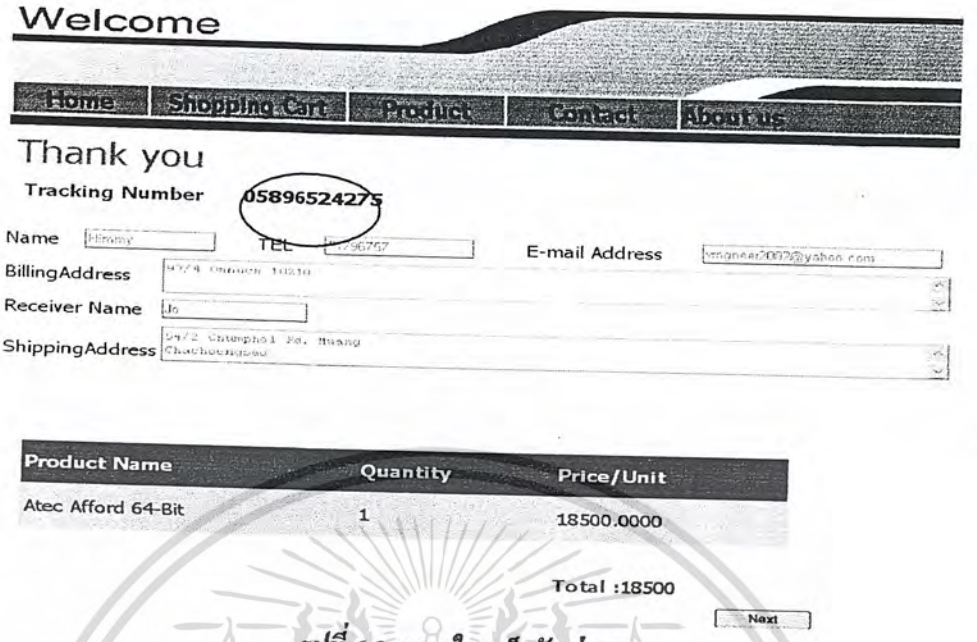
5. หลังจากคลิก Check out จะเข้าสู่หน้า Sign in ดังรูปที่ 6.6
ถ้าหากเป็นสมาชิกใหม่หรือต้องการแก้ไขข้อมูลที่เคยลงทะเบียนไว้ให้คลิก Registration จะได้รูปที่ 6.7 และกรอกข้อมูลจนครบกดปุ่ม Submit
ถ้าหากเป็นสมาชิกเก่าให้ใส่ User name และ Password ก่อนแล้วคลิก Submit จะได้รูปที่ 6.8
ในการกด Submit จะมีการเรียกเว็บเซอร์วิสขนส่ง โดยจะส่ง Tracking Numberมา



รูปที่ 6.6 แสดงหน้า Sign In

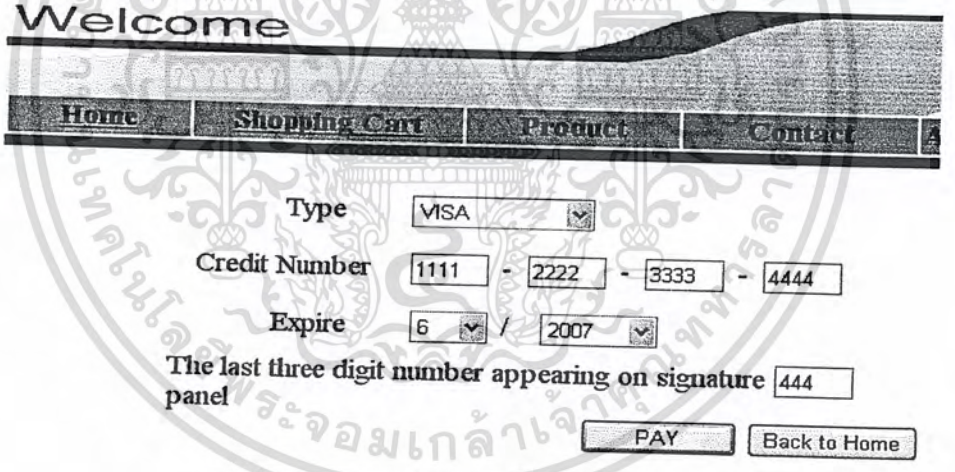


รูปที่ 6.7 แสดงหน้าการลงทะเบียน

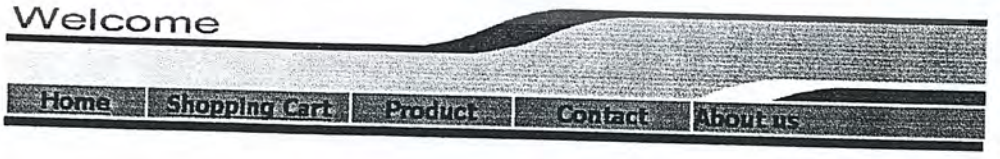


รูปที่ 6.8 แสดงใบเสร็จรับส่งของ

6. หลังจากคลิก Next ในรูปที่ 6.9 จะต้องกรอกข้อมูลเกี่ยวกับบัตรเครดิตเพื่อยืนยันความถูกต้องกับ BankWebService โดยถ้าถูกต้องเมื่อกด PAY จะได้รูปที่ 6.10



รูปที่ 6.9 แสดงการใส่ข้อมูลเกี่ยวกับบัตรเครดิต



Complete Transaction

รูปที่ 6.10 แสดงผลการซื้อสินค้าสำเร็จ

บทที่ 7
บทวิจารณ์และสรุป

7.1 บทวิจารณ์

จากโปรแกรมที่ได้พัฒนาขึ้นมาสามารถให้ผลการทดลองได้ตามการคาดหมาย แต่สิ่งที่ออกมามีข้อจำกัดคือ ในระบบการจ่ายเงินไม่สามารถใช้งานระบบการจ่ายเงินในโลกของความจริงได้ ต้องมีการจำลองการจ่ายเงินผ่านธนาคาร และจะต้องทำให้การติดต่อสื่อสารระหว่างแอพพลิเคชั่น และเว็บเซอร์วิสมีความปลอดภัยสูงสุด แต่เนื่องจากเทคโนโลยีนำมาประยุกต์ใช้เทคโนโลยีใหม่ มีเอกสารเผยแพร่ไม่มากนักทำให้การพัฒนาระบบรักษาความปลอดภัยเป็นไปอย่างช้าๆ

7.2 แนวทางในการพัฒนา

เนื่องจากโปรแกรมนี้เป็นการจำลองการค้าขายสินค้าบนอินเตอร์เน็ต ดังนั้นเราสามารถที่จะนำไปพัฒนาต่อเพื่อให้ระบบมีความปลอดภัย และสามารถที่จะนำไปใช้งานจริงได้

7.3 บทสรุป

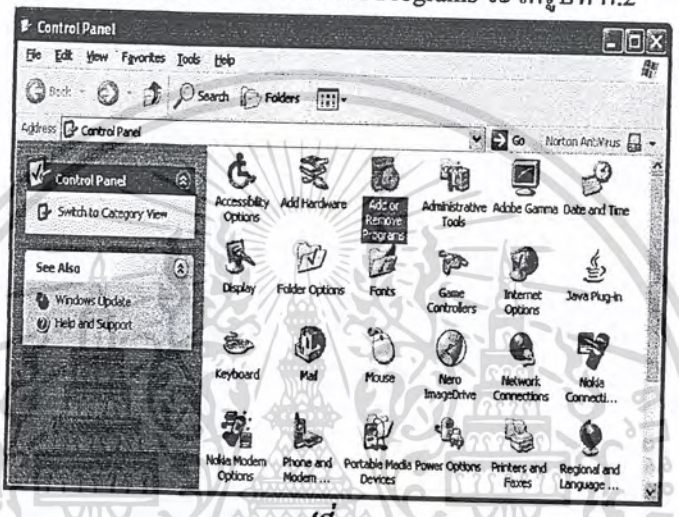
เนื่องจากสิ่งที่สำคัญที่สุดของเว็บเซอร์วิสคือ เรื่องของความปลอดภัย และถ้าปล่อยให้ใครก็ได้ทำการเรียกเว็บเซอร์วิสโดยไม่มีการตรวจสอบ จะทำให้เกิดปัญหาภายหลังได้ ดังนั้นจึงมีการนำ WS-Security มาประยุกต์ใช้ร่วมกับแอพพลิเคชั่นในการรักษาความปลอดภัย

ภาคผนวก ก.

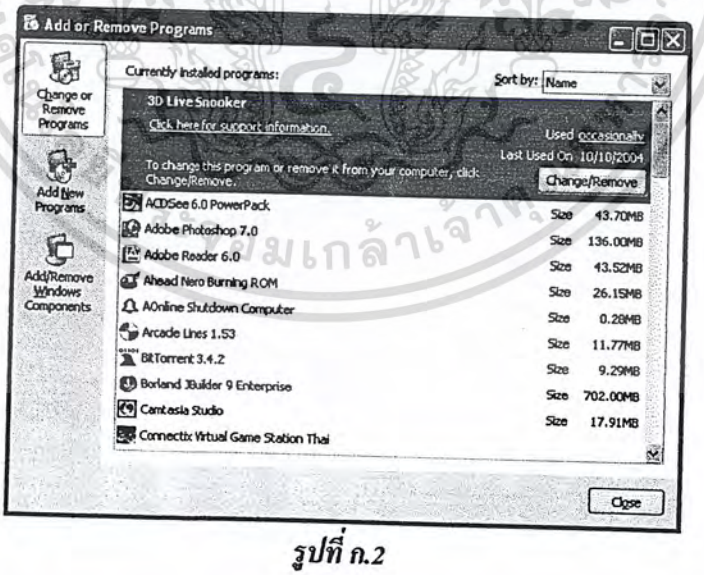
การติดตั้ง Internet Information Services(IIS)

เซิร์ฟเวอร์ชนิดนี้สามารถติดตั้งได้เฉพาะใน Windows NT/ 2000 / XP โดยสามารถติดตั้งโปรแกรมได้ดังนี้

1. นำแผ่น Windows ใส่ในช่องซีดีรอม แล้วคลิกปุ่ม Start → Setting → Control Panel จะปรากฏหน้าต่างดังรูปที่ ก.1 ให้ดับเบิลคลิกที่ไอคอน Add/Remove Programs จะได้รูปที่ ก.2

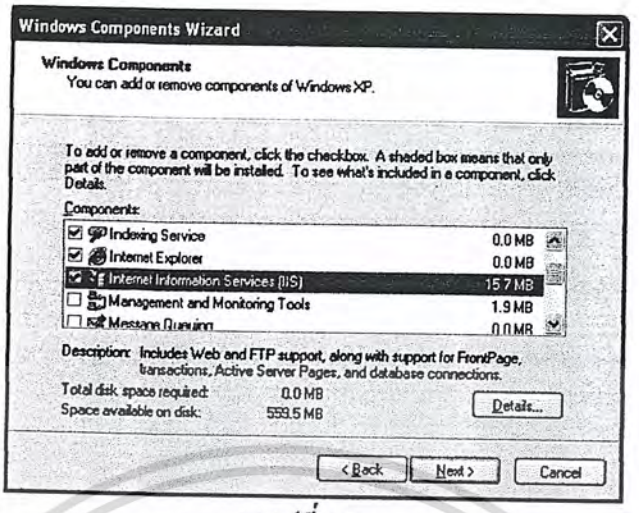


รูปที่ ก.1

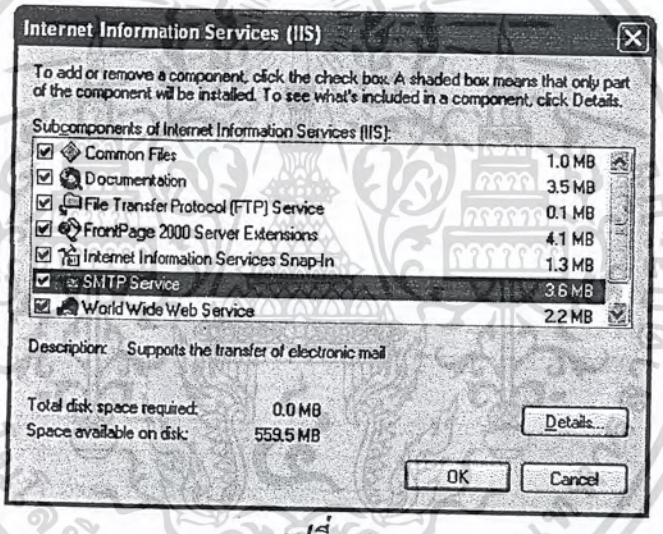


รูปที่ ก.2

2. จากรูปที่ ก.2 คลิกปุ่ม Add/Remove Windows Components จะได้รูปที่ ก.3 และให้เลือกที่ Internet Information Server (IIS) ดังรูปที่ ก.3 ซึ่งหากเราจะติดตั้ง Component เพิ่มก็ให้คลิกที่ปุ่ม Detail และเลือก Component ที่ต้องการ(แนะนำว่าควรเลือก Component ให้หมดทุกตัว) ดังรูปที่ ก.4 แล้วคลิกที่ปุ่ม OK พร้อมกับคลิกที่ปุ่ม Next เพื่อติดตั้งขั้นตอนต่อไป

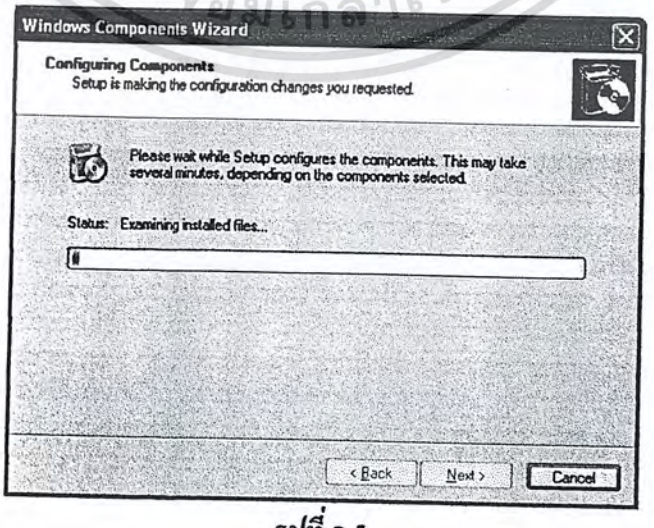


รูปที่ ก.3



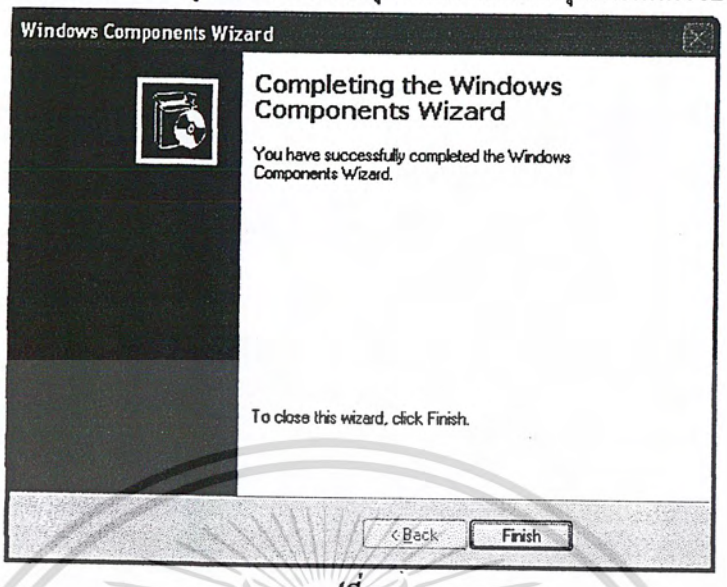
รูปที่ ก.4

3. รอการติดตั้งดังรูปที่ ก.5



รูปที่ ก.5

4. หลังจากโปรแกรมติดตั้งเสร็จสมบูรณ์แล้ว ให้คลิกปุ่ม Finish เพื่อสิ้นสุดการติดตั้งโปรแกรม



รูปที่ ก.6

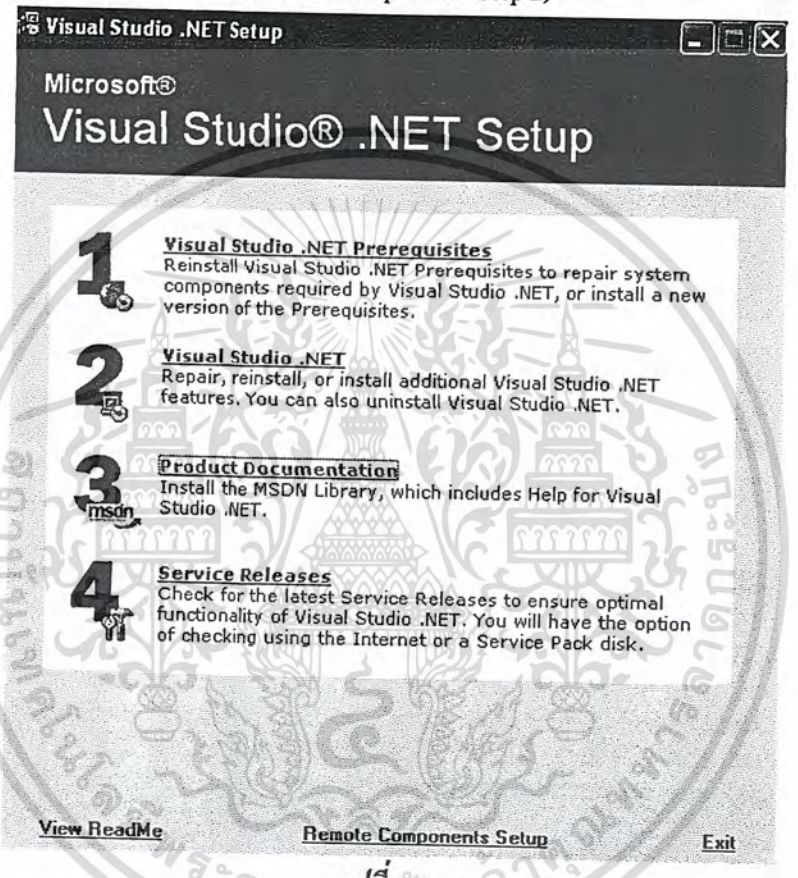
ภาคผนวก ข.

การติดตั้ง Visual Studio.NET

1. เปิดโปรแกรมแล้วเข้าสู่รูปที่ ข.1

จะมีให้อยู่ 4 Step ในการ Setup

(ในการติดตั้งนี้จะให้ใช้งานได้อย่างน้อยต้องทำ Step 1 และ Step 2)



รูปที่ ข.1

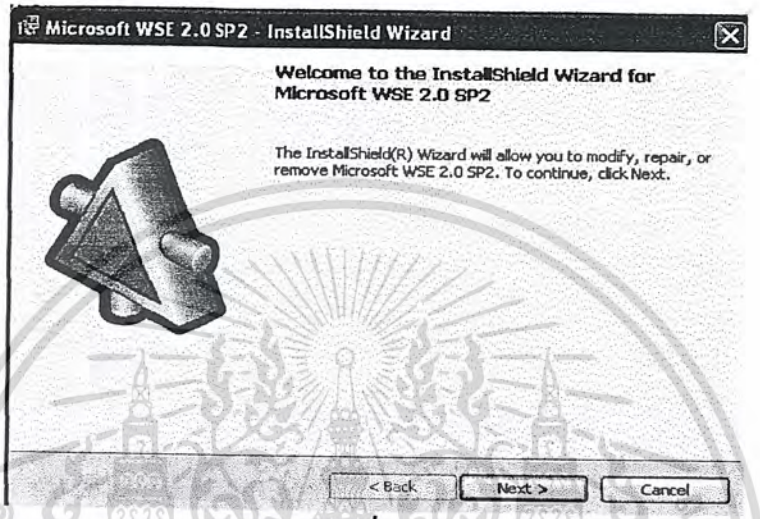
2. คลิกที่ Link Visual Studio.NET Prerequisites แล้วจะเรียกหา Prerequisite Disk ก็ใส่แผ่นที่ถามหาจนครบก็เป็นอันเสร็จสิ้นในขั้นนี้

3. คลิกที่ link Visual Studio.NET แล้วก็ใส่แผ่นที่ถามหาจนครบก็เป็นอันเสร็จสิ้นในขั้นนี้

4. Step 3 และ Step 4 ไม่จำเป็นต้องทำก็ได้

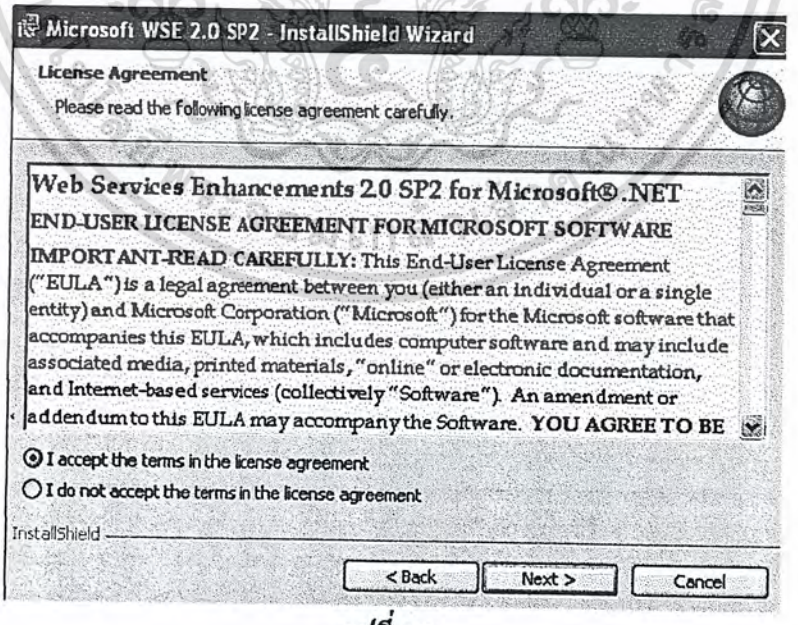
ภาคผนวก ค.
การติดตั้ง WSE2.0

1. เปิดโปรแกรมแล้วเข้าสู่การติดตั้ง WSE2.0 ดังรูปที่ ค.1 แล้วคลิก Next



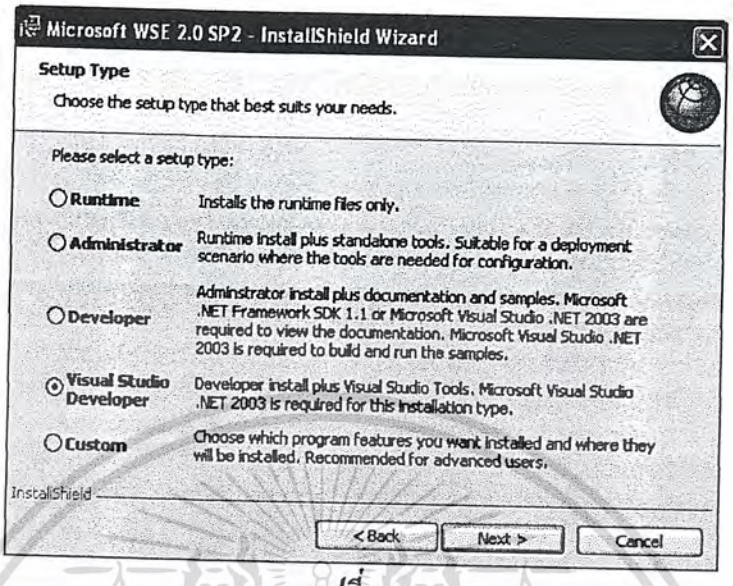
รูปที่ ค.1

2. เลือก I accept the terms in the license agreement แล้วคลิก Next ดังรูปที่ ค.2



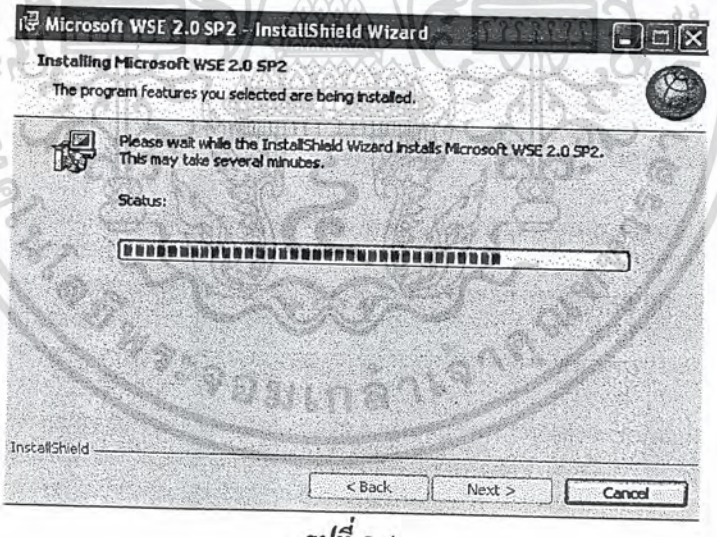
รูปที่ ค.2

3. เลือก Visual Studio Developer ดังรูปที่ ค.3



รูปที่ ค.3

4. จากรูปที่ ค.4 รอการติดตั้งหลังจากนั้นคลิก Next และ Finish ก็เป็นอันติดตั้งเสร็จ

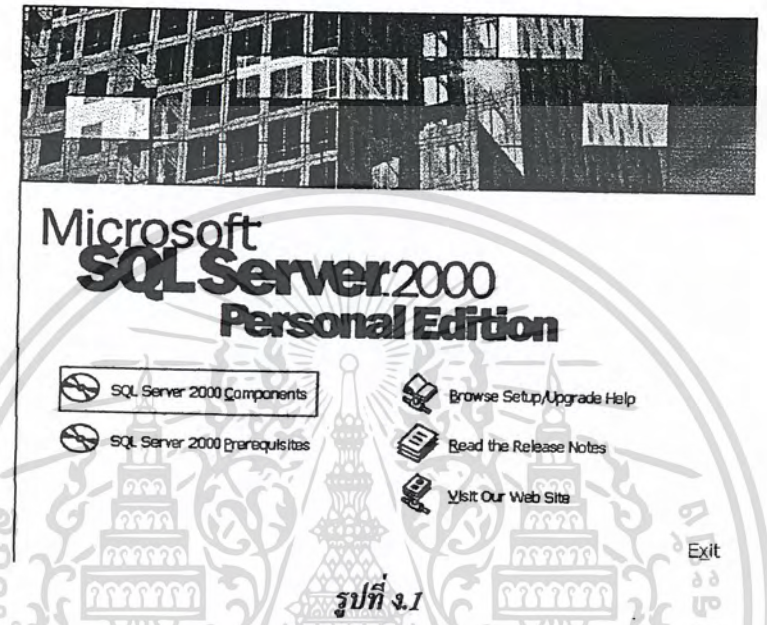


รูปที่ ค.4

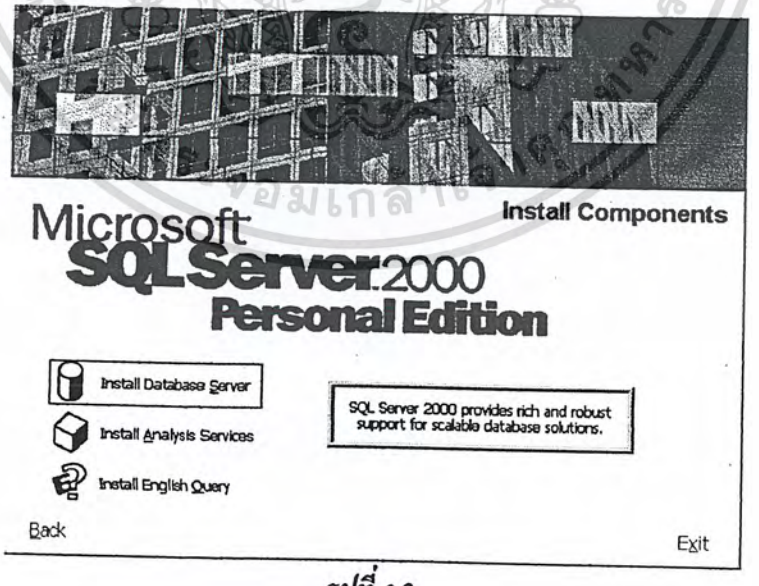
ภาคผนวก ง.

การติดตั้ง Microsoft SQL Server 2000

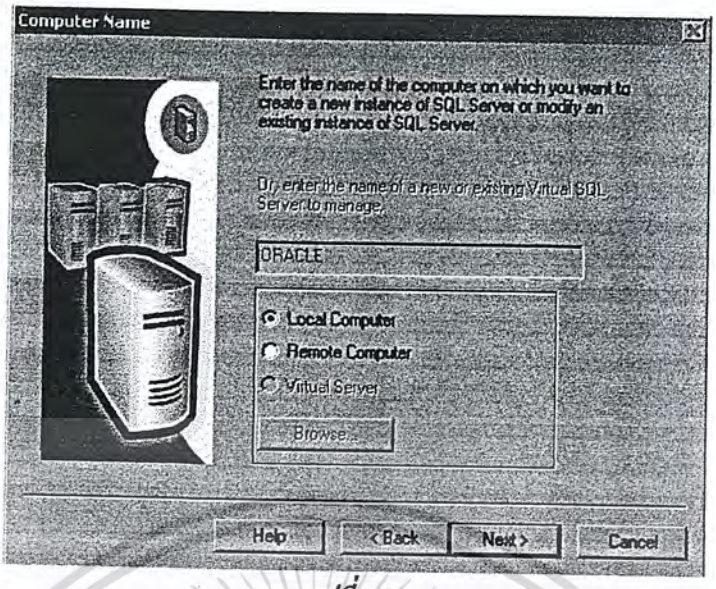
เลือก SQL Server 200 Componets แล้วทำตามขั้นตอนดังรูปที่ ง.1 ถึง รูปที่ ง.8



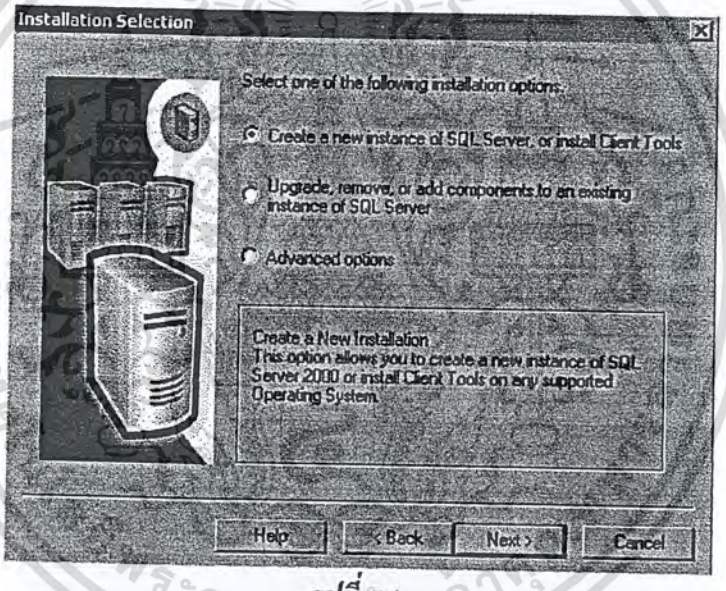
รูปที่ ง.1



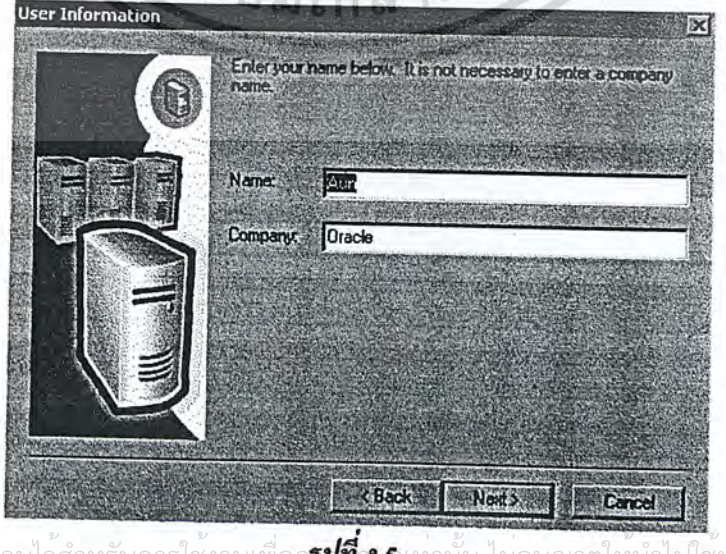
รูปที่ ง.2



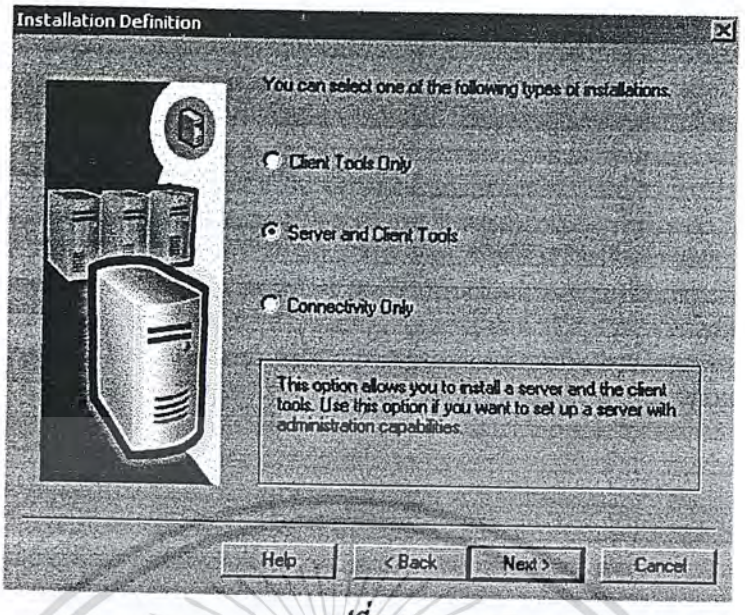
รูปที่ 3.3



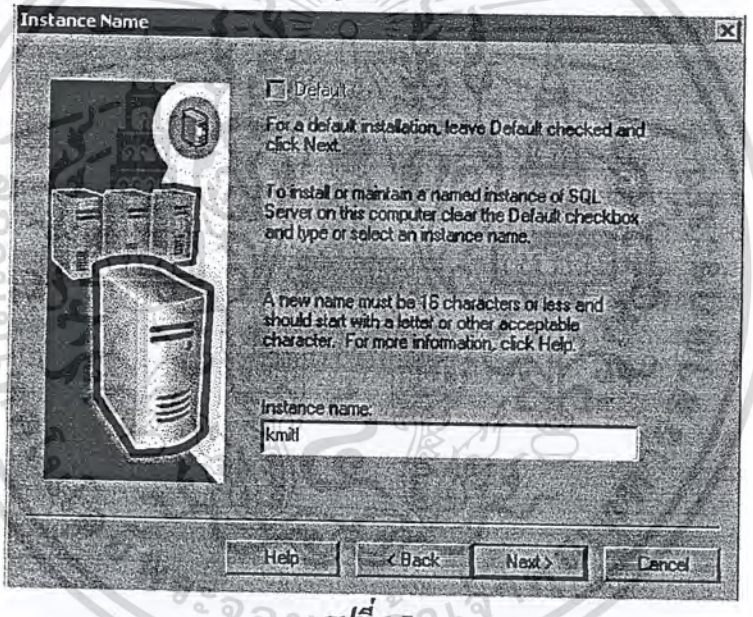
รูปที่ 3.4



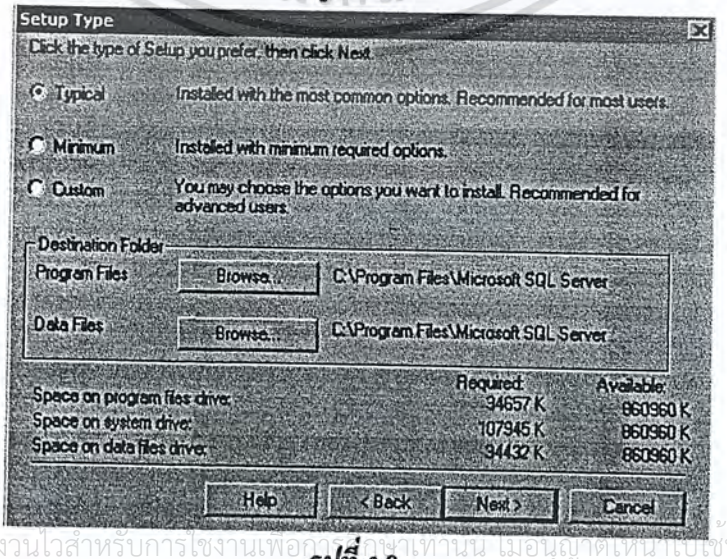
รูปที่ 3.5



รูปที่ 3.6



รูปที่ 3.7



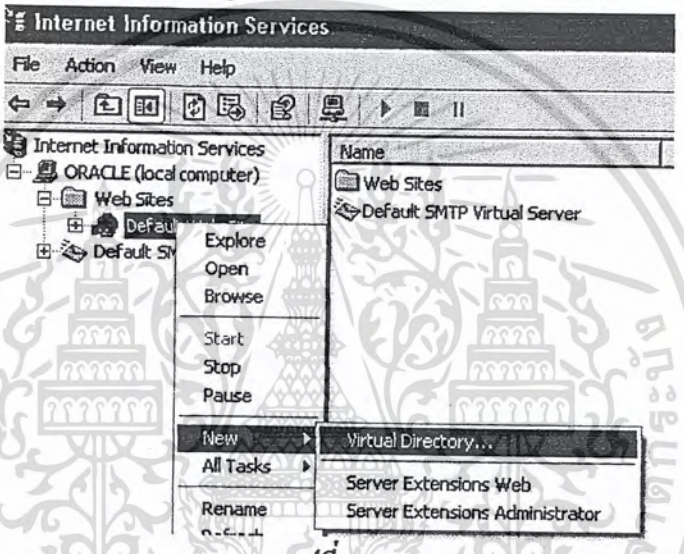
รูปที่ 3.8

ภาคผนวก จ.
การติดตั้งโปรแกรมระบบซื้อคอมพิวเตอร์ออนไลน์

การติดตั้งโฟลเดอร์ ProjectApp และ BankWebService

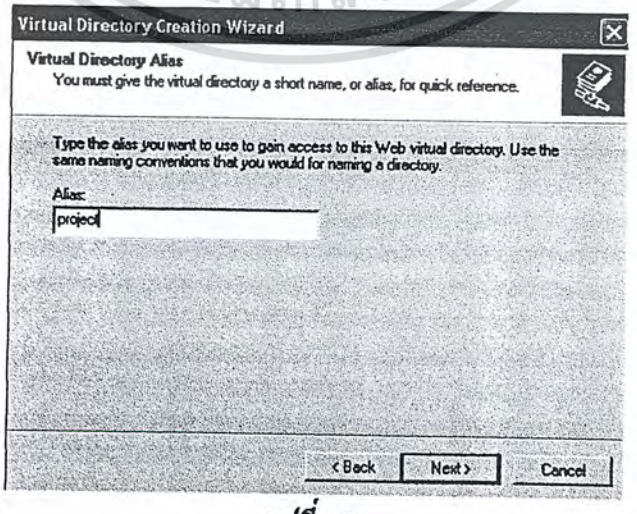
1. นำโฟลเดอร์ชื่อ ProjectApp ไปไว้ใน Directory c:\Inetpup\wwwroot

2. ไปที่ Start -> Setting -> Control Panel->Administrative Tools->Internet Information Services(IIS) แล้วเข้าไปที่ folder Web Sites จากนั้นคลิกขวาที่ Default Website จะเกิด popup แล้วเลือก New -> Virtual Directory ดังรูปที่ จ.1



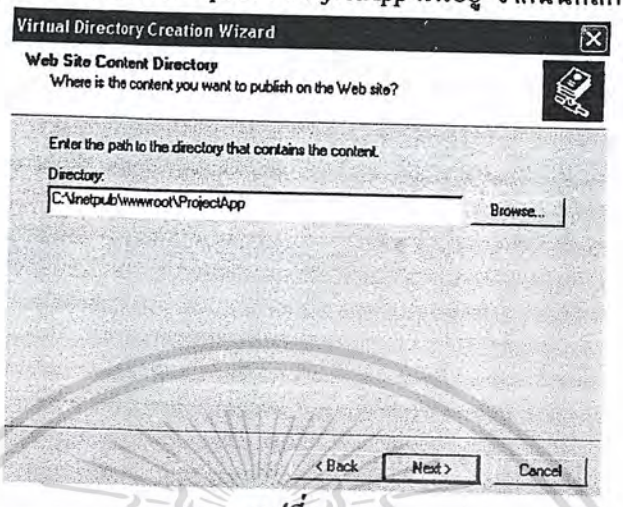
รูปที่ จ.1

3. หลังจากนั้นคลิก Next แล้วจะได้รูปที่ จ.2 ตั้งชื่อ Alias (ชื่ออะไรก็ได้) แล้วคลิก Next



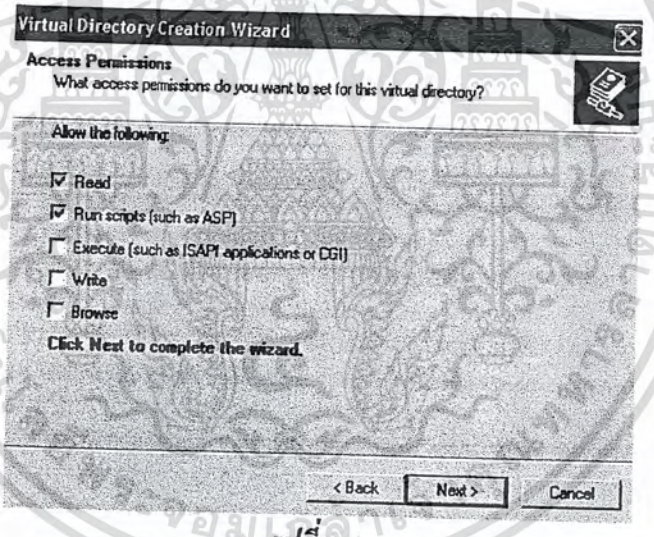
รูปที่ จ.2

4. จากรูปที่ จ.3 ให้คลิก Browse.. เพื่อเลือก path ที่ ProjectApp นั้นอยู่ จากนั้นคลิก Next



รูปที่ จ.3

5. เลือกเช็คบอกซ์ดังรูปที่ จ.4 แล้วกด Next -> finish



รูปที่ จ.4

6. โฟลเดอร์ BankWebService ก็ทำเช่นเดียวกับโฟลเดอร์ ProjectApp

การติดตั้งฐานข้อมูล

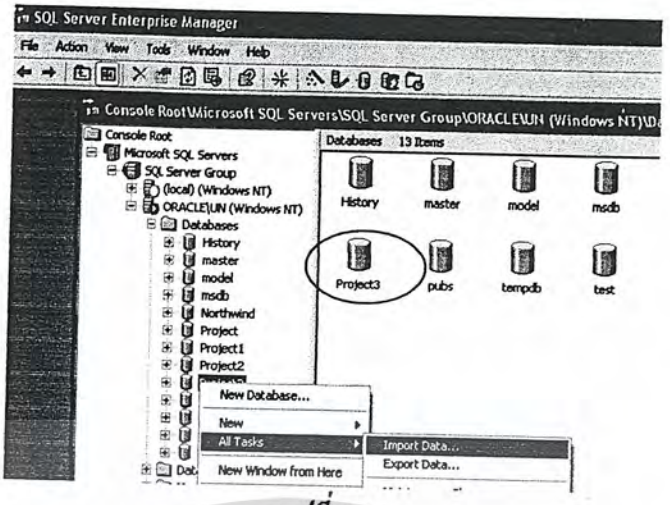
วิธี 1 ใช้โฟลเดอร์ DB

ขั้นตอน นำไฟล์ในโฟลเดอร์ DB ไว้ในโฟลเดอร์ชื่อ Data ของ Microsoft SQL Server

วิธี 2 ใช้db3 (เป็นไฟล์Microsoft Access)

ขั้นตอน

1. Microsoft SQL Server -> Enterprise Manager จะได้รูปที่ จ.5 แล้วสร้างฐานข้อมูล โดยคลิกขวาที่โฟลเดอร์ Database และเลือก New Database หลังจากนั้นก็จะมีฐานข้อมูลนั้นเกิดขึ้นในที่นี้ให้คือ Project3 หลังจากนั้นคลิกขวาที่ Project3 เลือก All Tasks -> Import Data.. จะได้รูปที่ จ.6

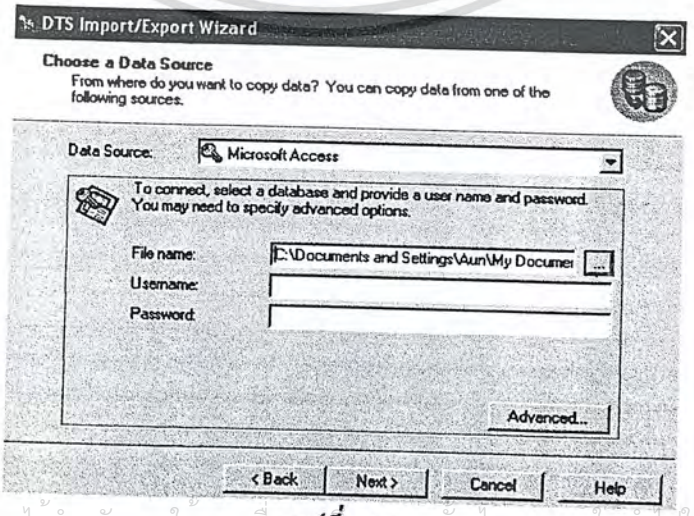


รูปที่ จ.5

2. จากรูปที่ จ.6 ให้คลิก Next

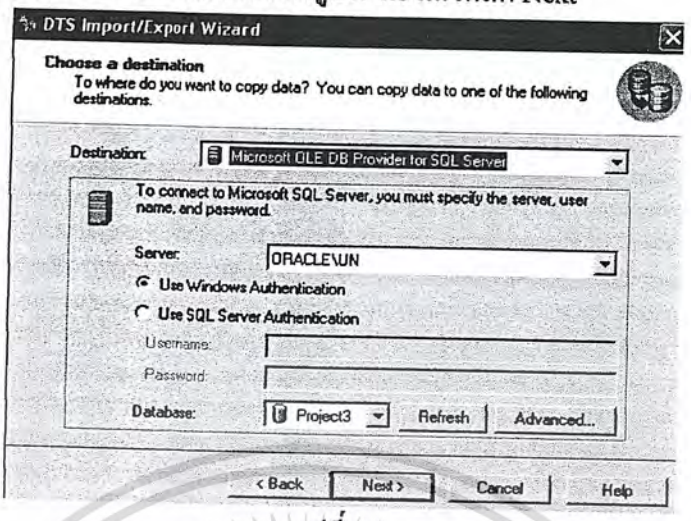
รูปที่ จ.6

3. จากรูปที่จ.7 เลือก Data Source เป็น Microsoft Access ที่ File name ให้คลิกปุ่ม ... เพื่อหา ไฟล์ db3 แล้วคลิก Next

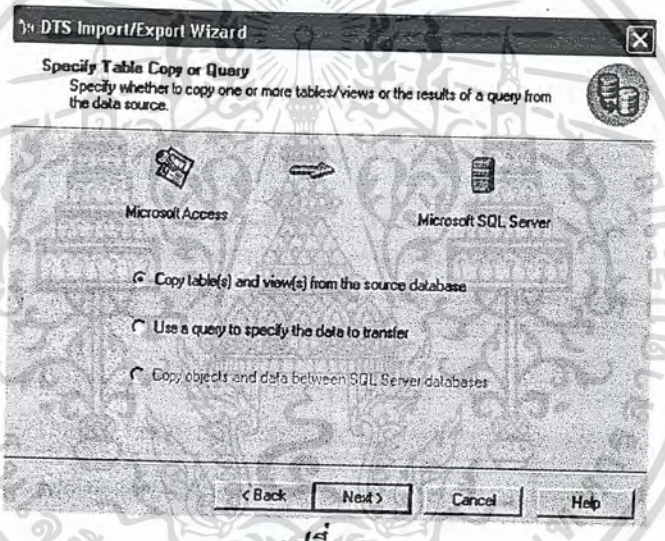


รูปที่ จ.7

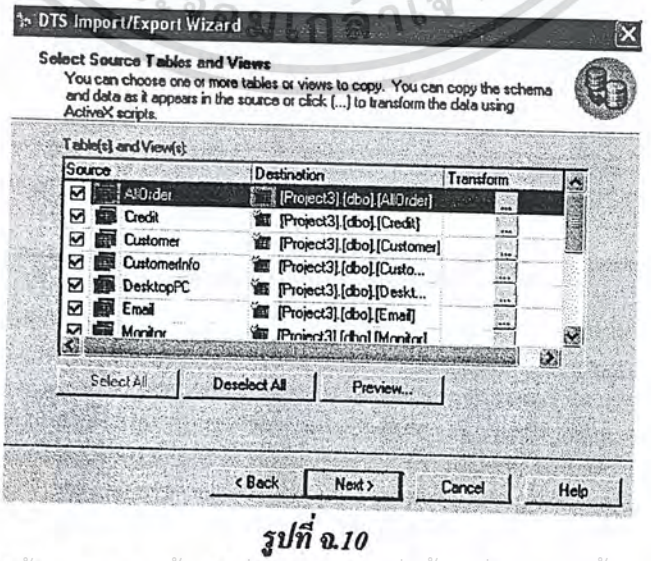
4. เลือก Destination , Server และ database ดังรูปที่ จ.8 แล้วคลิก Next



รูปที่ จ.8



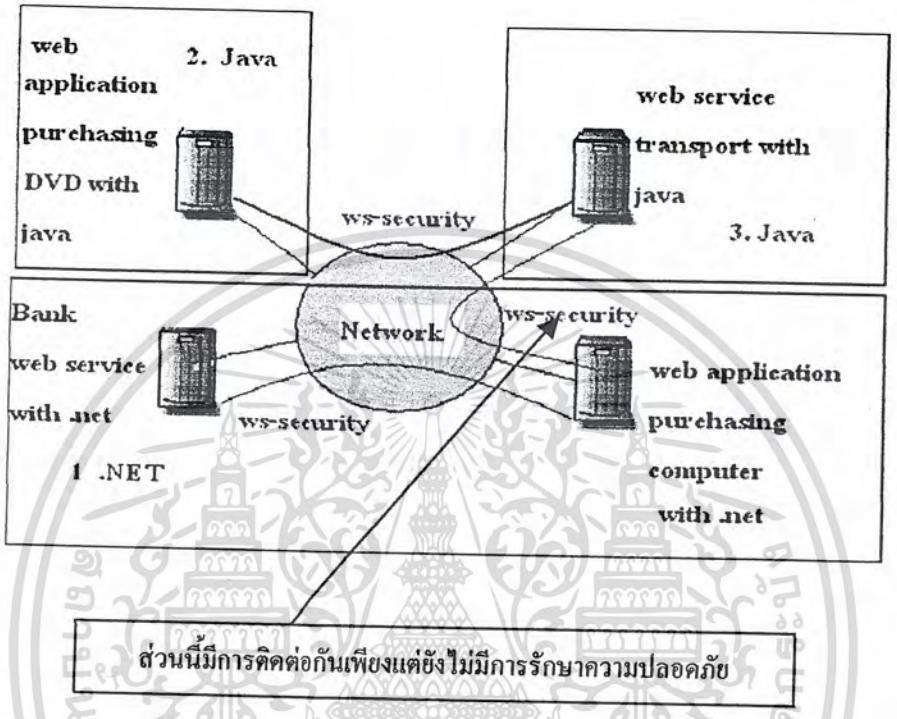
รูปที่ จ.9



รูปที่ จ.10

5.เลือก Select All แล้วคลิก Next จนกระทั่งไปจนกระทั่งเสร็จสมบูรณ์

ภาคผนวก ฉ.
ปัญหาที่แก้ไม่ได้ในงานวิจัยนี้



ปัญหาของการทำ ws-security ระหว่าง Application ที่พัฒนาด้วย แพลตฟอร์ม .NET และ Java เนื่องจากในโปรเจคนี้มีการทำ ws-security โดยใช้ PKI ซึ่งต้องมีการใช้ x.509 (ใบรับรองอิเล็กทรอนิกส์) แต่การใช้งานจริง x.509 จะต้องขอ x.509 จาก CA (ผู้ออกใบรับรองอิเล็กทรอนิกส์) ซึ่งจะต้องเสียค่าใช้จ่ายในการขอ x.509 ดังนั้น ไมโครซอฟต์ และ ซันไมโครซิสเตมส์ จึงได้สร้างแอพพลิเคชั่นสำหรับการสร้าง x.509(makecert ของบริษัทไมโครซอฟต์ , keytool ของบริษัท ซันไมโครซิสเตมส์) ขึ้นมาสำหรับใช้ในการสร้าง x.509 เพื่อทดสอบ ws-security แต่ปัญหาที่พบก็คือ x.509 ที่สร้างขึ้นจากโปรแกรม makecert และ x.509 ที่สร้างจากโปรแกรม keytool ทางคณะผู้จัดทำยังไม่สามารถนำมาใช้งานร่วมกันได้และยังไม่สามารถหาคำตอบได้ว่าปัญหาที่เกิดขึ้นเกิดจากสิ่งใด (ไมโครซอฟต์ได้เผยแพร่เอกสารที่เกี่ยวกับ ws-security และได้อ้างว่าสามารถที่จะทำ ws-security ระหว่าง .NET และ Java ได้) คณะผู้จัดทำได้ทำการค้นหาข้อมูลเกี่ยวกับการสร้าง ws-security ระหว่าง .NET และ Java จนในที่สุดก็พบวีดีโอที่เกี่ยวกับการทำ ws-security ระหว่าง .NET และ Java แต่วีดีโอที่ทางคณะผู้จัดทำไม่สามารถที่จะเปิดดูได้

บรรณานุกรม

H. Nielsen, S. Thatte, "Web Services", Microsoft, October 2001.

Mark O' neill (2003) web services security .Publisher: McGraw-Hill

W3C Working Draft, "XML Encryption Syntax and Processing," 04 March 2002.

สุวัฒนา สุขสมจินตน์ คัมภีร์การใช้ visual c# ฉบับสมบูรณ์ , กรุงเทพ : ซีเอ็ดยูเคชั่น,2546

สันติ ศรีลาศักดิ์ สร้างเว็บไซต์เพื่อขายสินค้าบนอินเตอร์เน็ต.กรุงเทพ: ออฟเซ็ท เพรส ,2547

ศุภชัย สมพานิช คู่มือการเขียนโปรแกรม visual c#.net ฉบับโปรแกรมเมอร์. นนทบุรี : อินโฟเพรส,2546