กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

A Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices

วิชุตา แก้วนพรัตน์

Wichuta Kaewnopparat

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Prince of Songkla University
2552

ชื่อวิทยานิพนธ์ กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP
ผู้เขียน นางสาววิชุตา แก้วนพรัตน์
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....
(ดร.ลัดดา ปรีชาวีรกุล)

คณะกรรมการสอบ

……..ประธานกรรมการ
(ดร.ฐิมาพร เพชรแก้ว)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

…….....กรรมการ
(ดร.ลัดดา ปรีชาวีรกุล)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

……..……
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP
ผู้เขียน นางสาววิชุตา แก้วนพรัตน์
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2551

บทคัดย่อ

ในยุคสารสนเทศ ผู้รับบริการใช้เทคโนโลยี RSS (Really Simple Syndicate) เพื่อรับข้อมูลข่าวสารที่ทันสมัยจากเว็บไซต์ต่าง ๆ เนื่องจากเข้าใช้งานเพียงเว็บไซต์เดียว ข้อมูลข่าวสารที่ต้องการจะถูกรวบรวมและส่งไปยังผู้รับบริการ นอกจากนี้ยังนิยมใช้อุปกรณ์สื่อสารเคลื่อนที่เพื่อติดตามข้อมูลข่าวสารอีกด้วย หลายองค์กรจึงหันมาให้ความสำคัญในการเผยแพร่สารสนเทศขององค์กรผ่าน RSS และอุปกรณ์สื่อสารเคลื่อนที่มากขึ้น อย่างไรก็ตามสารสนเทศที่เผยแพร่ผ่าน RSS โดยมากเป็นเพียงสารสนเทศทั่วไปที่ต้องการประชาสัมพันธ์ให้บุคคลทั่วไปทราบ สำหรับสารสนเทศที่ต้องการความปลอดภัย เช่น ข้อมูลบัตรเครดิต ข้อมูลการทำธุรกรรมทางการเงิน และข้อมูลข่าวสารส่วนบุคคล RSS ไม่ได้จัดเตรียมกลไกเพื่อเผยแพร่ข้อมูลข่าวสารดังกล่าวไว้ วิทยานิพนธ์นี้จึงเสนอกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP (A Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices: SInfoNM) ซึ่งประยุกต์ใช้วิทยาการเข้ารหัสลับ เพื่อทำให้ข้อมูลข่าวสารมีความปลอดภัยก่อนแจ้งไปยังผู้ที่เกี่ยวข้อง และนำ XSL มาใช้สำหรับสืบค้นข้อมูลที่ถูกเข้ารหัสไว้ ผลลัพธ์ที่ได้จะถูกส่งไปถอดรหัสก่อนแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้รับบริการ โดยมีการนิยามโครงสร้างเอกสาร RSS ด้วย XML Schema ร่วมกับ SchemaPath ผลการศึกษาแสดงให้เห็นว่า SInfoNM สามารถรวบรวมและแจ้งข้อมูลข่าวสารไปยังผู้ที่เกี่ยวข้องได้ อีกทั้งยังเพิ่มความสามารถของ RSS สำหรับแจ้งข้อมูลข่าวสารส่วนบุคคลได้อย่างปลอดภัยอีกด้วย

Thesis Title	A Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices
Author	Miss Wichuta Kaewnopparat
Major Program	Computer Science
Academic Year	2008

ABSTRACT

In the Information Age, people use RSS (Really Simple Syndication) Technology to help them get the latest contents from websites easily only accessing a single website as well as using mobile devices. Several companies have started to use RSS for distributing their information to customers. However, most contents published via RSS technology are public, not confidential such as credit card information, financial business information and etc. Since the RSS technology does not have a mechanism to ensure that the incoming information is really secure, and then this thesis proposes a Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices (SInfoNM). We applied the RSS technology together with the cryptography to make any RSS document being secure before disseminating it to relevant users. The SInfoNM also uses XSL to apply for private information retrieval and the XML schema and SchemaPath definition have been created for validation. The results displayed on a user's mobile device, giving users the latest information. The study results confirm that our mechanism is able to aggregate RSS documents and disseminate information to each user. The SInfoNM fulfills RSS technology for private information which can be distributed securely.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

รายการตาราง

รายการภาพประกอบ

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

รายการภาพประกอบ (ต่อ)

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

บทที่ 1

บทนำ

ในยุคสารสนเทศ อินเทอร์เน็ตเข้ามามีบทบาทสำคัญสำหรับเผยแพร่สารสนเทศขององค์กรไปยังผู้รับบริการ ทำให้สามารถติดตามข่าวสารความเคลื่อนไหวขององค์กรต่าง ๆ ได้สะดวกและง่ายยิ่งขึ้น แต่เนื่องจากจำนวนเว็บไซต์และปริมาณข้อมูลข่าวสารที่เพิ่มมากขึ้น ทำให้เป็นเรื่องยากและใช้เวลานานในการพิจารณาว่าเว็บไซต์เหล่านั้นมีข้อมูลข่าวสารอะไรใหม่หรือไม่ จึงได้มีการเสนอเทคโนโลยีเพื่อรวบรวมข้อมูลข่าวสารจากเว็บไซต์ที่ผู้รับบริการชื่นชอบ และส่งไปยังผู้รับบริการเมื่อมีการอัพเดทข้อมูลข่าวสารของเว็บไซต์ โดยไม่ต้องเข้าไปเยี่ยมชมทีละเว็บไซต์ขึ้นซึ่งเรียกว่า เทคโนโลยี RSS (Really Simple Syndication)

RSS ถูกพัฒนาด้วยภาษา XML (Extensible Markup Language) เพื่อรวบรวมและเผยแพร่ข้อมูลข่าวสารที่ทันสมัยให้กับผู้รับบริการผ่านอินเทอร์เน็ต ด้วยการเข้าใช้งานเพียงที่เดียวข้อมูลข่าวสารที่ต้องการจะถูกรวบรวมและส่งมาให้กับผู้รับบริการ โดย RSS ไม่ได้จำกัดการใช้งานเฉพาะเผยแพร่ข่าวสารเท่านั้น แต่ยังสามารถใช้งานเพื่อเผยแพร่สารสนเทศได้หลากหลายตามความต้องการของผู้ใช้ หลายองค์กรจึงหันมาให้ความสำคัญในการเผยแพร่สารสนเทศขององค์กรผ่าน RSS มากขึ้น อย่างไรก็ตามสารสนเทศที่ใช้งานผ่าน RSS โดยมากเป็นเพียงสารสนเทศทั่วไปที่ต้องการประชาสัมพันธ์ให้บุคคลทั่วไปทราบ สำหรับสารสนเทศที่ต้องการความปลอดภัย ตัวอย่างเช่น ข้อมูลบัตรเครดิต ข้อมูลการทำธุรกรรมทางการเงิน และข้อมูลข่าวสารส่วนบุคคล เป็นต้น RSS ไม่ได้จัดเตรียมกลไกสำหรับเผยแพร่ข้อมูลข่าวสารดังกล่าวไว้ และแม้ว่า XML จะมีแท็ก (Tag) ที่เกี่ยวข้องกับความปลอดภัยของข้อมูลก็ตาม แต่แท็กเหล่านั้นก็ไม่ได้ถูกเรียกใช้งานใน RSS เพราะไม่ใช่แท็กที่ RSS กำหนดไว้

นอกจากนี้ปัจจุบันอุปกรณ์สื่อสารเคลื่อนที่ ตัวอย่างเช่น พีดีเอ (Personal Digital Assistant: PDA) โทรศัพท์มือถือ (Mobile Phone) และโน้ตบุ๊ก (Notebook) เข้ามามีบทบาทสำคัญกับชีวิตประจำวันในการติดตามข้อมูลข่าวสารมากขึ้น เนื่องจากทำให้ผู้รับบริการไม่พลาดข้อมูลข่าวสารความเคลื่อนไหวต่าง ๆ

วิทยานิพนธ์นี้จึงเสนอกลไกการทำงาน เพื่อรวบรวมและแจ้งข้อมูลข่าวสารไปยังผู้ที่เกี่ยวข้อง และทำให้ RSS สามารถแจ้งสารสนเทศที่ต้องการความปลอดภัยได้ โดยเสนอกลไกการทำงานบนพื้นฐานของอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP เพื่อให้ผู้รับบริการได้รับสารสนเทศที่ทันสมัยสะดวกยิ่งขึ้น

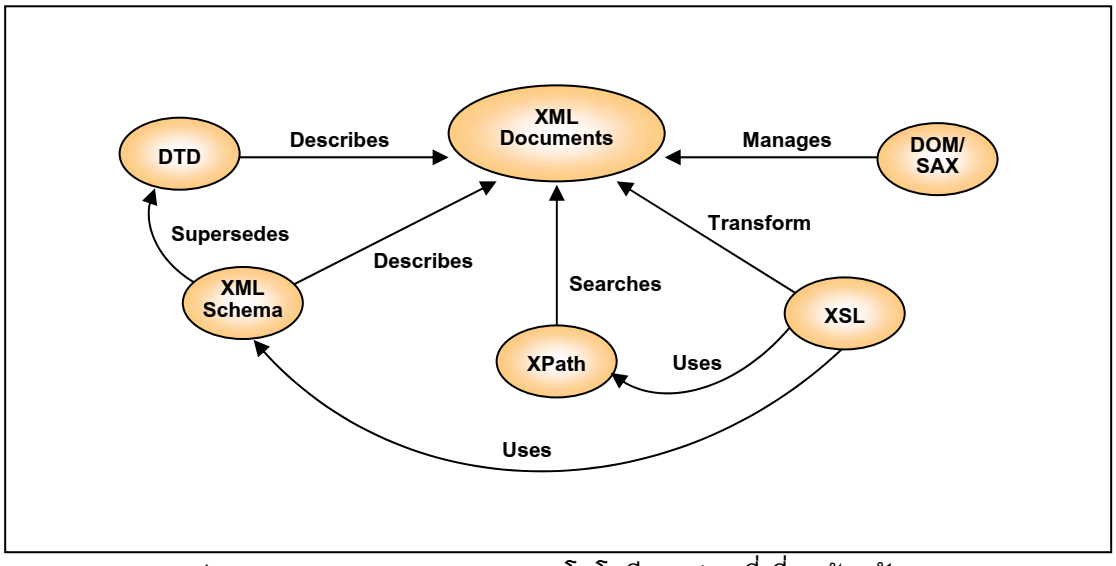
1.1 การตรวจเอกสาร

RSS ถูกพัฒนาขึ้นด้วยภาษา XML ดังนั้นการจัดการเอกสาร RSS จึงเกี่ยวข้องกับมาตรฐานและเทคโนโลยีต่าง ๆ ของ XML ดังแสดงในภาพประกอบ 1.1 และเพื่อให้สามารถเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัยผ่าน RSS ได้นั้น จึงนำวิทยาการเข้ารหัสลับ (Cryptography) มาประยุกต์ใช้ เพื่อทำให้ข้อมูลภายในเอกสาร RSS มีความปลอดภัยก่อนถูกเผยแพร่และส่งไปยังผู้รับบริการที่เกี่ยวข้อง

1.1.1 ภาษา XML (Extensible Markup Language)

ภาษา XML เป็นภาษามาร์คอัพ (Markup) ที่ใช้วิธีระบุเนื้อหาและจัดรูปแบบด้วยไฟล์ข้อความ (Text File) โดยถูกออกแบบมาเพื่อใช้แท็กอธิบายความหมายของข้อมูลและอนุญาตให้ผู้ใช้กำหนดแท็กใช้งานได้ตามต้องการ ทำให้ XML มีความยืดหยุ่นและใช้งานได้หลากหลาย จึงถูกนำมาใช้เป็นสื่อกลางในการแลกเปลี่ยนข้อมูลผ่านระบบอินเทอร์เน็ตมากขึ้น

เอกสาร XML ถูกนิยามโครงสร้างด้วย DTD (Document Type Definition) หรือ XML Schema และสามารถเข้าถึงด้วย XML Parser ซึ่งแบ่งตามวิธีสำรวจเนื้อหาของเอกสารออกเป็น 2 ชนิด คือ DOM (Document Object Model) และ SAX (Simple API for XML) การสืบค้นข้อมูลจะใช้ XPath เพื่อระบุตำแหน่งของข้อมูลที่ต้องการ ส่วนการแสดงผลเอกสาร XML นั้น จะใช้ XSL (Extensible Stylesheet Language) สำหรับจัดรูปแบบการแสดงผล (Benz and Durant, 2003; Dykes and Tittel, 2005)



ภาพประกอบ 1.1 มาตรฐานและเทคโนโลยีบางส่วนที่เกี่ยวข้องกับ XML

1.1.2 เทคโนโลยี RSS (Really Simple Syndication)

RSS เป็นรูปแบบหนึ่งในการรวบรวมและเผยแพร่เนื้อหาของเว็บไซต์ ถูกพัฒนาขึ้นตามแนวคิดของเทคโนโลยี Push (Umbach, 1997) เพื่อรวบรวมข้อมูลข่าวสารจากแหล่งผู้ให้บริการข้อมูลข่าวสารต่าง ๆ และแจ้งให้ผู้รับบริการทราบเมื่อมีการอัพเดทข้อมูลข่าวสารโดยไม่ต้องมีการร้องขอ อีกทั้งยังช่วยลดปัญหาในเรื่องการละเมิดลิขสิทธิ์ และทำให้ผู้พัฒนาเว็บไซต์ไม่ต้องเสียเวลาปรับปรุงเว็บเพจเมื่อผู้ให้บริการมีการปรับปรุงข้อมูลข่าวสาร

เทคโนโลยี RSS ถูกนำมาใช้อย่างแพร่หลายในกลุ่มของเว็บข่าว (News Site) และเว็บล็อก (Web log) แต่ยังมีงานวิจัยหลายงานด้วยกันที่ประยุกต์ใช้ RSS กับงานด้านอื่น ๆ ตัวอย่างทางด้านการศึกษา ได้แก่ การใช้งาน RSS เป็นเครื่องมือในการแบ่งปันข้อมูลสำหรับทำวิจัยของกลุ่มนักศึกษา โดยข้อมูลที่นักศึกษาได้รวบรวมไว้ เช่น วารสารทางวิชาการ ผลงานที่ได้รับการตีพิมพ์ บล็อกและข้อมูลจากแหล่งข้อมูลต่าง ๆ จะถูกนำมาแบ่งปันเพื่อใช้ประโยชน์ในการทำวิจัย (Cold, 2006) และมีการเสนอการใช้งาน RSS เพื่อแจ้งข้อมูลต่าง ๆ ของรายวิชาเรียน เช่น กำหนดการสอบ เอกสารประกอบการเรียน เป็นต้น ทำให้นักศึกษาได้รับข้อมูลข่าวสารเกี่ยวกับวิชาที่เรียนสะดวกและรวดเร็วขึ้น (Gltzbach *et al*., 2007) นอกจากนี้ RSS ยังถูกนำไปประยุกต์ใช้เพื่อปรับปรุงการแสดงเนื้อหาบนอุปกรณ์สื่อสารเคลื่อนที่ เนื่องจาก RSS เป็นข้อมูลรายละเอียดโดยย่อ จึงเหมาะสำหรับแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ที่มีทรัพยากรจำกัด (Blekus *et al*., 2006)

1.1.3 ความปลอดภัย (Security)

ความปลอดภัยของข้อมูลถูกนำมาพิจารณาเป็นประเด็นหลัก สำหรับแลกเปลี่ยนข้อมูลข่าวสารผ่านอินเทอร์เน็ต เนื่องจากอินเทอร์เน็ตเป็นเครือข่ายสาธารณะทำให้การป้องกันการบุกรุกจากผู้ที่ไม่ได้รับอนุญาตในการเข้าถึงข้อมูล โดยเฉพาะข้อมูลที่ค่อนข้างอ่อนไหวเป็นเรื่องสำคัญอย่างยิ่ง เพื่อให้มั่นใจได้ว่าข้อมูลที่สำคัญต่าง ๆ จะไม่ถูกทำลาย เปลี่ยนแปลง หรือรั่วไหลไปยังบุคคลที่ไม่เกี่ยวข้อง เทคนิคสำคัญที่ทำให้ข้อมูลข่าวสารมีความปลอดภัย คือ การประยุกต์ใช้วิทยาการเข้ารหัสลับ ซึ่งเป็นเครื่องมือในการเข้ารหัสและถอดรหัสข้อมูล

ตัวอย่างงานวิจัยที่ประยุกต์ใช้วิทยาการเข้ารหัสลับกับเอกสาร XML ได้แก่ การเข้ารหัสเอกสาร XML เฉพาะส่วน ที่เรียกว่า Element-Wise XML Encryption โดยผลลัพธ์ที่ได้คือ เอกสาร XML ที่มีข้อมูลบางส่วนถูกเข้ารหัสไว้ (Maruyama and Imamura, 2000) การใช้ XSLT เพื่อเข้ารหัสและถอดรหัสเอกสาร XML ซึ่งเป็นการขยายกลไกการทำงานของ XSLT สำหรับเข้ารหัสเอกสาร XML (Bartlett and Cook, 2002) และมีงานวิจัยที่ได้ออกแบบและประยุกต์ใช้ API สำหรับความปลอดภัยของเอกสาร XML โดยได้นิยามภาษาที่เรียกว่า DSL

(Document Security Language) เพื่อใช้ในการเข้ารหัสและถอดรหัสเอกสาร XML ซึ่งมีแนวคิดพื้นฐานมาจาก XSLT (Chang and Hwang, 2007)

นอกจากนี้ได้มีการประยุกต์ใช้ Greasemonkey ซึ่งเป็นส่วนขยายของ Mozilla Firefox เพื่อถอดรหัสข้อมูลภายในเอกสาร RSS ผลลัพธ์ที่ได้จากการถอดรหัสจะถูกแทนที่ลงในหน้าเว็บเพจที่เปิดใช้งานก่อนแสดงผลบนเบราว์เซอร์ (Browser) ของผู้ใช้ โดยได้เสนอ Greasemonkey Script เพื่อสืบค้นข้อมูลที่ถูกเข้ารหัส และถอดรหัสข้อมูลด้วยอัลกอริทึม Blowfish ซึ่งการทำงานถูกออกแบบมาสำหรับใช้งานบนเบราว์เซอร์ Mozilla Firefox ทำให้ผู้ใช้ต้องติดตั้ง Greasemonkey Script บนเบราว์เซอร์ก่อนจึงจะถอดรหัสเอกสาร RSS ได้ (Gregorio, 2005)

จากงานวิจัยที่กล่าวมาข้างต้นเห็นได้ว่าโดยมาก RSS ถูกนำไปใช้สำหรับเผยแพร่ข้อมูลข่าวสารทั่วไป ถึงแม้ว่าจะมีการเสนอการใช้ Greasemonkey เพื่อถอดรหัสข้อมูลข่าวสารที่เป็นความลับภายในเอกสาร RSS ก็ตาม แต่การทำงานยังขึ้นอยู่กับเบราว์เซอร์ที่ใช้ วิทยานิพนธ์นี้จึงเสนอกลไกที่ทำให้ RSS สามารถแจ้งสารสนเทศที่ปลอดภัยไปยังผู้ที่เกี่ยวข้อง โดยสามารถใช้งานได้กับเบราว์เซอร์ที่หลากหลายทั้งบนเครื่องคอมพิวเตอร์ทั่วไปและอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP เพื่อให้ผู้ใช้ติดตามข้อมูลข่าวสารที่ทันสมัยได้สะดวกยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อออกแบบกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

1.2.2 เพื่อพัฒนากลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

1.3 ขอบเขตการดำเนินงาน

1.3.1 ออกแบบกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP ดังนี้

1) การเผยแพร่สารสนเทศที่ต้องการความปลอดภัย
2) การรวบรวมเอกสาร RSS
3) การสร้างเอกสาร RSS เฉพาะผู้ใช้
4) การเขียน RSS Schema เพื่อกำหนดโครงสร้างเอกสาร RSS สำหรับข้อมูลข่าวสารที่ต้องการความปลอดภัย

1.3.2 พัฒนาและทดสอบกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP ตามที่ได้ออกแบบไว้

1.4 ขั้นตอนและระยะเวลาการดำเนินงาน

1.4.1 ขั้นตอนการดำเนินงาน

1) ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง ดังนี้
   1.1) ภาษา XML
   1.2) เทคโนโลยี RSS
   1.3) วิทยาการเข้ารหัสลับ
   1.4) เทคโนโลยีอื่น ๆ ที่เกี่ยวข้อง
2) ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย
3) วิเคราะห์และออกแบบกลไกการทำงาน
4) พัฒนาและทดสอบกลไกการทำงานตามที่ได้ออกแบบไว้
5) เขียนบทความวิจัย
6) จัดทำเอกสารวิทยานิพนธ์

1.4.2 ระยะเวลาการดำเนินงาน

มกราคม 2551 – เมษายน 2552

1.4.3 แผนการดำเนินการวิจัย

ระยะเวลาการดำเนินการวิจัยแสดงดังตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาการดำเนินการวิจัย

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน															
	2551												2552			
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง	←											→				
2. ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย				←					→							
3. วิเคราะห์และออกแบบกลไกการทำงาน					←				→							
4. พัฒนาและทดสอบกลไกการทำงาน							←						→			
5. เขียนบทความวิจัย							←						→			
6. จัดทำเอกสารวิทยานิพนธ์													←			→

1.5 สถานที่และเครื่องมือที่ใช้

1.5.1 สถานที่

ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการประยุกต์ (CS207) ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.5.2 เครื่องมือที่ใช้

1) ด้านฮาร์ดแวร์

เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยความจำขนาด 1 GB และฮาร์ดดิสก์ความจุ 120 GB สำหรับพัฒนาและทดสอบระบบ

2) ด้านซอฟต์แวร์

2.1) ระบบปฏิบัติการ Microsoft Windows XP

2.2) โปรแกรมเว็บเซิร์ฟเวอร์ Apache 2.2.4

2.3) ระบบจัดการฐานข้อมูล MySQL 5.0.45

2.4) ภาษา PHP 5.2.3 และ JavaScript

2.5) โปรแกรม Windows Mobile 6.1 Emulator สำหรับจำลองการทำงานบนอุปกรณ์สื่อสารเคลื่อนที่

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ได้กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีต่าง ๆ ที่ใช้ในการออกแบบและพัฒนากลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP ประกอบด้วย แนวคิดพื้นฐานเกี่ยวกับเว็บ ภาษา XML (Extensible Markup Language) การนิยามโครงสร้างเอกสาร XML เทคโนโลยี RSS (Really Simple Syndication) ความปลอดภัย (Security) และอุปกรณ์สื่อสารเคลื่อนที่ (Mobile Device)

2.1 แนวคิดพื้นฐานเกี่ยวกับเว็บ

เว็บเป็นเทคโนโลยีทางด้านเครือข่ายคอมพิวเตอร์ ที่นำเอาเครือข่ายคอมพิวเตอร์ต่าง ๆ มาเชื่อมต่อกัน มีวัตถุประสงค์เพื่อแลกเปลี่ยนและใช้ข้อมูลร่วมกันระหว่างเครือข่าย โดยแบ่งการทำงานออกเป็น 2 ฝั่ง คือ ฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้รับบริการหรือไคลเอนต์ (Client) และฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้ให้บริการหรือเซิร์ฟเวอร์ (Server)

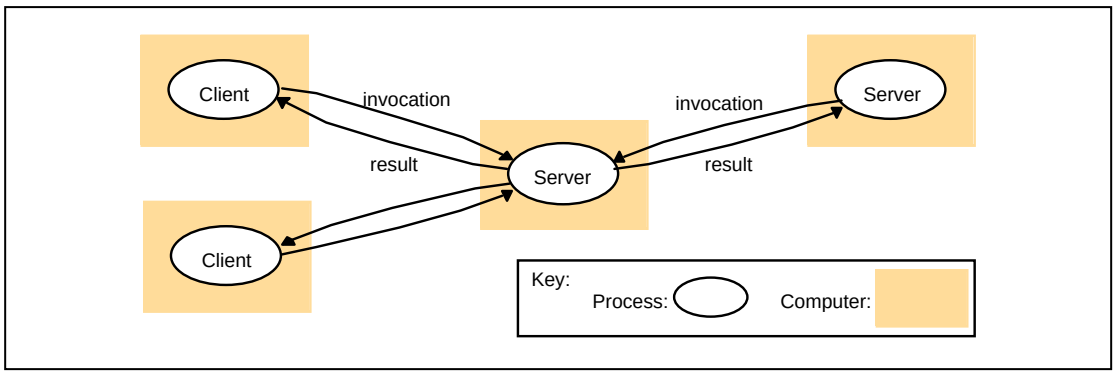
2.1.1 TCP/IP

การติดต่อสื่อสารผ่านเครือข่ายอินเทอร์เน็ตที่มีเครื่องคอมพิวเตอร์และผู้ใช้บริการจำนวนมากนั้นจำเป็นต้องมีโพรโทคอล (Protocol) ซึ่งเป็นข้อกำหนดมาตรฐานหรือข้อตกลงที่ใช้ในการสื่อสารระหว่างคอมพิวเตอร์ในระบบเครือข่ายที่เรียกว่า TCP/IP (Transmission Control Protocol/Internet Protocol) เข้ามาใช้เพื่อกำหนดกฎเกณฑ์ รูปแบบการเชื่อมต่อเครื่องคอมพิวเตอร์ในเครือข่าย การโอนย้ายข้อมูล การแสดงสถานะที่ใช้ในการเชื่อมต่อข้อมูลระหว่างเครื่องต้นทางและเครื่องปลายทางที่มีความแตกต่างกันในเรื่องของชนิดฮาร์ดแวร์ และระบบปฏิบัติการ ให้สามารถติดต่อสื่อสารทำงานร่วมกันได้อย่างถูกต้องและมีประสิทธิภาพ (Stevens, 1994)

2.1.2 สถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ (Client-Server)

ไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) เป็นสถาปัตยกรรมหนึ่งทางด้านคอมพิวเตอร์ที่มีความสำคัญ และถูกนำมาใช้อย่างแพร่หลายในระบบอินเทอร์เน็ต โดยมี

เซิร์ฟเวอร์ทำหน้าที่หลักในการจัดการข้อมูลและทรัพยากรต่าง ๆ ให้กับไคลเอนต์ เพื่อให้เกิดการใช้ข้อมูลและทรัพยากรร่วมกันระหว่างเครื่องคอมพิวเตอร์ในระบบเครือข่าย (Coulouris *et al*., 2001)

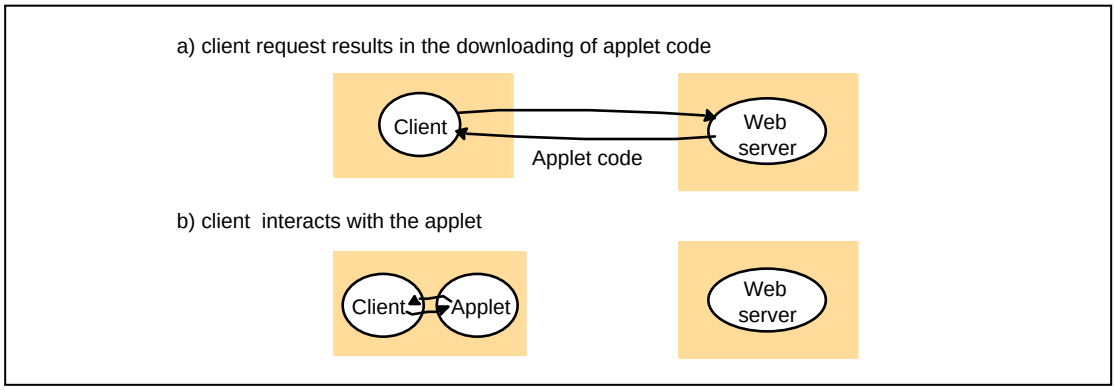


ภาพประกอบ 2.1 สถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ (Coulouris *et al*., 2001)

จากภาพประกอบ 2.1 ไคลเอนต์ส่งคำร้องขอ (Invocation) ไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ประมวลผลเรียบร้อยแล้วจะส่งผลลัพธ์ (Result) กลับไปยังไคลเอนต์ โดยเซิร์ฟเวอร์อาจประพฤติตนเป็นไคลเอนต์เพื่อร้องขอการทำงานไปยังเซิร์ฟเวอร์อื่น ๆ ยกตัวอย่างเช่น Search Engines ผู้ใช้จะได้ผลลัพธ์จากการค้นหาข้อมูลของเว็บไซต์ที่มีอยู่ในอินเทอร์เน็ต โดยผลลัพธ์ที่ได้นั้นมาจากการทำงานของโปรแกรมที่เรียกว่า Web Crawlers ซึ่งทำหน้าที่อยู่เบื้องหลัง Search Engines โดยใช้ HTTP (Hypertext Transfer Protocol) ร้องขอการเข้าถึงเว็บเซิร์ฟเวอร์อื่น ๆ ที่มีในอินเทอร์เน็ตเพื่อสืบค้นข้อมูลที่ผู้ใช้ต้องการ

2.1.3 Mobile Code

Mobile Code คือ ชุดคำสั่งที่สามารถส่งจากคอมพิวเตอร์เครื่องหนึ่งไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง โดยการประมวลผลชุดคำสั่งจะกระทำที่ฝั่งไคลเอนต์ ทำให้สามารถลดภาระการทำงานของเซิร์ฟเวอร์ได้



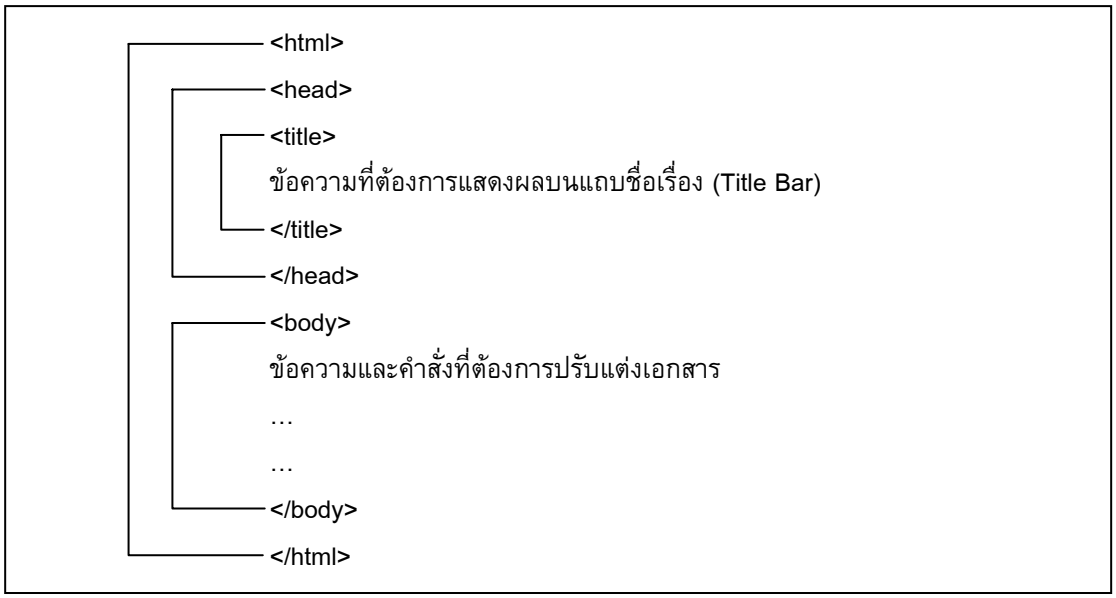
ภาพประกอบ 2.2 ตัวอย่างการทำงานของ Mobile Code (Coulouris *et al*., 2001)

ตัวอย่างของ Mobile Code ได้แก่ Java Applets, ActiveX controls และ JavaScript เป็นต้น (Wikipedia, 2008: Online) จากภาพประกอบ 2.2 เมื่อผู้ใช้เปิดใช้งานเว็บเบราว์เซอร์และเข้าไปยังเว็บเซิร์ฟเวอร์ที่เก็บ Applet code ไว้ ชุดคำสั่งเหล่านั้นจะถูกโหลดมายังเบราว์เซอร์ และประมวลผลที่ฝั่งเบราว์เซอร์ของผู้ใช้

2.1.4 ภาษา HTML (HyperText Markup Language)

HTML (HyperText Markup Language) ถูกนำมาใช้ในการพัฒนาเว็บเพจเพื่อแสดงผลข้อมูลบนอินเทอร์เน็ตผ่านโปรแกรมเว็บเบราว์เซอร์ โดยมีไฮเปอร์ลิงค์ (Hyperlink) เป็นตัวเชื่อมโยงเอกสารบนอินเทอร์เน็ตเข้าด้วยกัน เอกสาร HTML มีลักษณะเป็นไฟล์ข้อความที่ต้องอาศัยการแปลความจากเว็บเบราว์เซอร์ ตัวอย่างเว็บเบราว์เซอร์ที่เป็นที่รู้จักได้แก่ Microsoft Internet Explorer, Mozilla Firefox และ Opera เป็นต้น

เอกสาร HTML มีองค์ประกอบ 2 ส่วน คือ ส่วนที่เป็นเนื้อหาและส่วนที่เป็นคำสั่งหรือแท็ก (Tag) โดยทั่วไปแท็ก HTML จะอยู่ในรูปแบบ <...>...</...> ซึ่งเว็บเบราว์เซอร์จะแปลงแท็กเหล่านี้แล้วจึงแสดงผลลัพธ์ที่ได้ (ไพศาล โมลิสกุลมงคล, 2538; สุวัฒนา สุขสมจินตน์, 2545) โครงสร้างของเอกสาร HTML แสดงดังภาพประกอบ 2.3 และตัวอย่างแท็กพื้นฐานของเอกสาร HTML แสดงดังตารางที่ 2.1



ภาพประกอบ 2.3 โครงสร้างเอกสาร HTML

ตารางที่ 2.1 ตัวอย่างแท็กพื้นฐานของเอกสาร HTML

คำสั่ง/แท็ก (Tag)	คำอธิบาย
<html>…</html>	คำสั่งเริ่มต้นและสิ้นสุดของเอกสาร HTML
<head>…</head>	กำหนดข้อความในส่วนที่เป็นชื่อเรื่องมีคำสั่งย่อยคือ <title>
<title>…</title>	เป็นส่วนที่ใช้แสดงชื่อของเอกสารบนแถบชื่อเรื่อง (Title Bar)
<body>…</body>	ส่วนเนื้อหาของเอกสาร HTML ประกอบด้วยแท็กและข้อมูลต่าง ๆ ที่ต้องการแสดงผล
<p>	คำสั่งสำหรับขึ้นย่อหน้าใหม่
<img src= “…”>	คำสั่งสำหรับแสดงรูปภาพ
…	กำหนดไฮเปอร์ลิงค์ เพื่อเชื่อมโยงไปยังเว็บไซต์และเอกสารต่าง ๆ

2.2 ภาษา XML (Extensible Markup Language)

ภาษา XML ถูกนำเสนอโดย W3C (World Wide Web Consortium) เป็นภาษามาร์คอัพที่ใช้วิธีระบุเนื้อหาและจัดรูปแบบด้วยไฟล์ข้อความ ถูกออกแบบมาเพื่อใช้แท็กในการอธิบายความหมายของข้อมูล ผู้ใช้สามารถกำหนดแท็กที่ใช้งานได้ตามต้องการ ทำให้ XML มีความยืดหยุ่นสามารถใช้งานได้หลากหลายซึ่งแตกต่างจากภาษา HTML ที่ถูกออกแบบมาเพื่อแสดงผลข้อมูลผ่านเว็บ และสามารถอธิบายข้อมูลได้เฉพาะส่วนที่อยู่ในโครงสร้างหลัก ๆ ของเอกสารเท่านั้น ทำให้เป็นเรื่องยากที่จะดึงส่วนของข้อมูลออกจากโครงสร้างเอกสารมาประมวลผล (สุธี พงศาสกุลชัย, 2550)

ประโยชน์ของภาษา XML ที่เห็นได้ชัด คือ สามารถใช้สร้างข้อมูลที่อธิบายความหมายด้วยตัวมันเองได้ และเขียนโปรแกรมดึงข้อมูลไปใช้งานได้ง่าย เอกสาร XML จึงมีคุณสมบัติที่ทำให้เครื่องสามารถเข้าใจได้ง่าย (Machine Readable) และมนุษย์สามารถเข้าใจได้ง่าย (Human Readable) เช่นกัน ทำให้ถูกนำมาใช้เป็นสื่อกลางในการแลกเปลี่ยนสารสนเทศมากขึ้น ซึ่งสามารถใช้งานได้กับทุกแพลตฟอร์ม (Platform) เนื่องจากเป็นเพียงไฟล์ข้อความธรรมดา

นอกจากนี้ภาษา XML ยังถูกนำไปพัฒนาเป็นภาษาสำหรับลักษณะงานเฉพาะด้านอีกจำนวนมาก ได้แก่ ภาษา WML ที่นำมาใช้สร้าง WAP สำหรับอุปกรณ์มือถือ (WAP Forum, 2001) ภาษา MathML ที่ใช้ในวงการคณิตศาสตร์ (Carlisle *et al*., 2003) และเทคโนโลยี RSS ที่ใช้ในรวบรวมข้อมูลข่าวสารของเว็บไซต์ต่าง ๆ (Finkelstein, 2005) เป็นต้น

2.2.1 องค์ประกอบสำคัญของเอกสาร XML

1) แท็ก (Tag)

แท็กเป็นส่วนประกอบสำคัญของภาษามาร์คอัพ การกำหนดแท็กเริ่มต้น (Start Tag) ชื่อแท็กจะอยู่ภายในเครื่องหมาย “<” และ “>” เช่น <book> ส่วนการกำหนดแท็กสิ้นสุด (End Tag) จะกำหนดชื่อของแท็กอยู่ภายในเครื่องหมาย “</” และ “>” เช่น </book> โดยจะมีเครื่องหมาย “/” แทรกอยู่ด้านหน้า และตั้งแต่แท็กเริ่มต้นไปจนถึงแท็กสิ้นสุดจะถูกเรียกว่า “อิลิเมนต์ (Element)”

```
<first>John</first>
```

ภาพประกอบ 2.4 ตัวอย่างอิลิเมนต์ของ XML

จากภาพประกอบ 2.4

<first>	คือ แท็กเริ่มต้น
</first>	คือ แท็กสิ้นสุด
<first>John</first>	คือ อิลิเมนต์

2) แอททริบิวต์ (Attribute)

แอททริบิวต์ คือ การระบุคุณสมบัติเพื่ออธิบายส่วนเพิ่มเติมให้กับอิลิเมนต์ ซึ่งการกำหนดแอททริบิวต์จะไม่มีผลกระทบกับบางแอพพลิเคชัน (Application) ที่อาจไม่จำเป็นต้องใช้งานแอททริบิวต์นั้น ๆ โดยรูปแบบการกำหนดแอททริบิวต์ แสดงดังภาพประกอบ 2.5 และตัวอย่างแสดงดังภาพประกอบ 2.6

```
<element attribute = “value” | ‘value’>text</element>
```

ภาพประกอบ 2.5 รูปแบบการกำหนดค่าแอททริบิวต์

```
<book ISBN = “974-94136-7-8”>Beginning XML</book>
หรือ
<book ISBN = ‘974-94136-7-8’>Beginning XML</book>
```

ภาพประกอบ 2.6 ตัวอย่างการกำหนดค่าแอททริบิวต์

จากภาพประกอบ 2.6 เป็นการกำหนดค่าแอททริบิวต์ สำหรับระบุรหัส ISBN ของหนังสือให้กับอิลิเมนต์ <book>

3) Entity

Entity คือ ข้อมูลที่ถูกกำหนดความหมายไว้แล้ว ทำให้สามารถนำข้อมูลเหล่านั้นกลับมาใช้ใหม่โดยไม่ต้องเขียนบ่อย ๆ สำหรับ Entity ของ XML มีอยู่ด้วยกัน 2 ประเภท ดังนี้

3.1) Entity ที่กำหนดโดย W3C ทำให้ XML Parser สามารถประมวลผล Entity นั้น ๆ ได้อย่างถูกต้อง โดย Entity ที่สำคัญมีดังนี้

&lt;	มีค่าเท่ากับ	<
&gt;	มีค่าเท่ากับ	>
&amp;	มีค่าเท่ากับ	&
&quot;	มีค่าเท่ากับ	”
&apos;	มีค่าเท่ากับ	’

3.2) Entity ที่ผู้ใช้กำหนดเอง ผู้พัฒนาสามารถกำหนด Entity ขึ้นใช้เองได้ เพื่อประโยชน์ในการเรียกใช้ซ้ำ แต่ต้องมีการประกาศความหมายของ Entity ไว้ในส่วนของ Document Type Definition (DTD) ก่อน

2.2.2 โครงสร้างเอกสาร XML

เอกสาร XML มีโครงสร้างหลัก 2 ส่วน คือ Prolog และ Document Element โดยเอกสาร XML สามารถมี Root Element ได้เพียงหนึ่งอิลิเมนต์ แต่สามารถมีอิลิเมนต์ย่อยได้ไม่จำกัด

```
1   <?xml version = “1.0”  encoding = “UTF-8”?>          } Prolog Element
2   <!— filename = detailbook.xml -->
3   <books>                                              }
4       <book isbn = “974-94136-7-8”>                    }
5           <title>Beginning XML</title>                 }
6           <author>David</author>                       }
7       </book>                                          } Document Element
8       <book isbn = “974-94136-7-5”>                    }
9           <title>Web Programming with Ajax</title>     }
10          <author>John</author>                        }
11      </book>                                          }
12  </books>                                             }
```

ภาพประกอบ 2.7 ตัวอย่างเอกสาร XML

เอกสาร XML ดังภาพประกอบ 2.7 สามารถอธิบายส่วนต่าง ๆ ได้ดังนี้

บรรทัดที่ 1-2 คือ ส่วนของ Prolog Element เป็นส่วนที่ใช้ประกาศเอกสาร XML ซึ่งบรรจุส่วนต่าง ๆ ได้แก่ เวอร์ชั่น หมายเหตุ และ DTD เป็นต้น

บรรทัดที่ 3-12 คือ ส่วนของ Document Element เป็นอิลิเมนต์ที่ประกอบด้วยอิลิเมนต์อื่น ๆ ซ้อนกันอย่างเป็นลำดับ โดยมีอิลิเมนต์ <books> เป็น Root Element อิลิเมนต์ <book> เป็น Child Element และมี Subchild Element คือ อิลิเมนต์ <title> และ <author>

2.2.3 กฎพื้นฐานในการเขียน XML (Well-Formed)

1) ทุกอิลิเมนต์ของ XML ต้องประกอบด้วยแท็กเริ่มต้นและแท็กสิ้นสุด โดยทั้งสองแท็กต้องมีชื่อเหมือนกัน เช่น <book>…</book>

2) การกำหนดชื่อแท็กคำนึงถึง Case Sensitive คือ ตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กมีความหมายแตกต่างกัน

3) เอกสาร XML จะต้องมี Root Element และมีได้เพียงหนึ่ง Root เท่านั้น โดยเป็นแท็กที่อยู่บนสุดตามหลังส่วนของการประกาศ XML

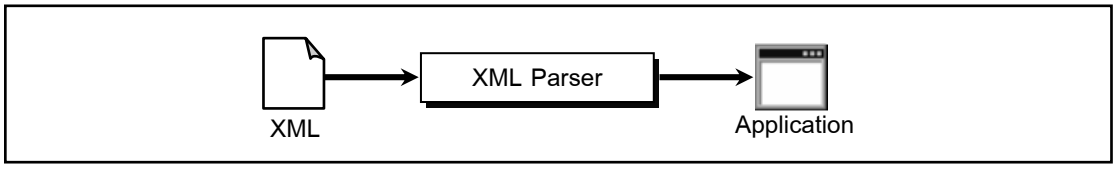
4) อิลิเมนต์ของ XML ต้องซ้อนกันอย่างเป็นลำดับ โดยไม่สามารถสลับตำแหน่งของแท็กปิดได้ เช่น <book><title>…</title></book> เป็นต้น

5) XML จะไม่ตัดส่วนที่เป็น White Space ในข้อความออก เช่น การเว้นวรรค เป็นต้น

6) การตั้งชื่ออิลิเมนต์ของเอกสาร XML สามารถใช้อักขระตัวเลขและอักขระพิเศษได้ ยกเว้นเครื่องหมาย “&” และไม่สามารถใช้ตัวเลขหรือตัวอักขระพิเศษนำหน้าชื่อของอิลิเมนต์ นอกจากนี้ยังห้ามเว้นช่องว่างระหว่างชื่ออิลิเมนต์อีกด้วย

2.2.4 XML Parser

XML Parser คือ ตัวแปลภาษา XML ทำหน้าที่อ่าน แปลความหมาย วิเคราะห์โครงสร้าง รวมถึงตรวจสอบความถูกต้องของเอกสาร XML การเข้าถึงข้อมูลในเอกสาร XML นั้น XML Parser จะเป็นตัวกลางระหว่างเอกสาร XML และแอพพลิเคชันที่จะนำข้อมูลของเอกสาร XML ไปใช้ ดังแสดงในภาพประกอบ 2.8



ภาพประกอบ 2.8 กระบวนการทำงานของ XML Parser

โดยสามารถจำแนกชนิดของ XML Parser ได้หลายวิธี แต่หลักเกณฑ์ที่นิยมมี 2 วิธี คือ

1) จำแนกตามการตรวจสอบเอกสาร XML

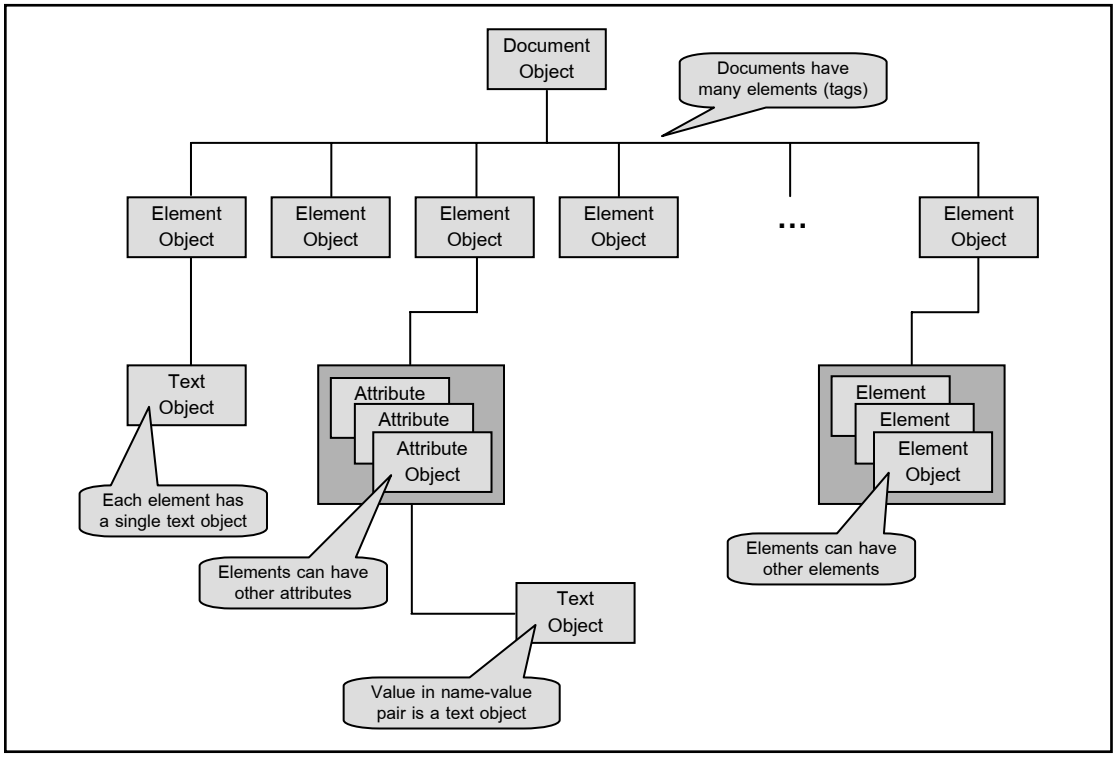
1.1) Non-validating Parser เป็นการตรวจสอบเฉพาะโครงสร้างพื้นฐาน และไวยากรณ์ของเอกสาร XML ว่ามีคุณสมบัติ well-formed หรือไม่

1.2) Validating Parser เป็นตรวจสอบความถูกต้องของเอกสาร XML ตามกฎของ DTD หรือ Schema รวมทั้งตรวจสอบโครงสร้างพื้นฐานและไวยากรณ์ของเอกสาร XML ว่ามีคุณสมบัติ well-formed ด้วยหรือไม่

2) จำแนกตามวิธีสำรวจเนื้อหาเอกสาร

2.1) Tree-based Parser เป็นการอ่านข้อมูลจากแฟ้ม XML ขึ้นมาทั้งหมด แล้วจัดองค์ประกอบต่าง ๆ ของ XML ให้อยู่ในรูปต้นไม้ (Tree) เก็บไว้ในหน่วยความจำหลักของคอมพิวเตอร์ ได้แก่ DOM (Document Object Model)

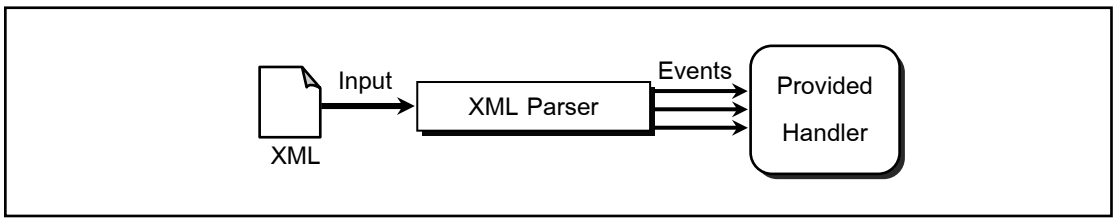
DOM ได้รับการรับรองเป็นมาตรฐานโดย W3C โครงสร้างต้นไม้ของ DOM หรือ DOM Tree ประกอบด้วยข้อมูลต่าง ๆ เช่น อิลิเมนต์ (Element) แอททริบิวต์ (Attribute) และข้อความ (Text) เป็นต้น โดย DOM สามารถอ่าน และจัดการเพิ่ม ลบ หรือแก้ไขข้อมูลในเอกสาร XML ได้ โครงสร้างของ DOM Tree แสดงดังภาพประกอบ 2.9



ภาพประกอบ 2.9 โครงสร้าง DOM Tree

2.2) Event-based Parser เป็นการอ่านเอกสาร XML และตอบสนองต่อสิ่งที่อ่านได้โดยถือเป็นเสมือนเหตุการณ์ ได้แก่ SAX (Simple API for XML)

SAX เป็น API (Application Programming Interface) ที่ทำงานโดยการแปลความหมายของเหตุการณ์ที่เกิดขึ้นเป็นหลัก เมื่อ parser อ่านข้อมูลจากเอกสาร XML ในแต่ละครั้งจะจดจำโครงสร้างไวยากรณ์ของเอกสาร XML ไว้ และตอบสนองต่อเหตุการณ์ที่อ่านได้ด้วยวิธีการที่กำหนด ซึ่งกระบวนการทำงานแสดงดังภาพประกอบ 2.10



ภาพประกอบ 2.10 กระบวนการทำงานของ SAX

การใช้งานของ DOM และ SAX มีข้อดีและเสียแตกต่างกันขึ้นอยู่กับลักษณะงาน และเอกสาร XML ที่ต้องการประมวลผล โดยสามารถเปรียบเทียบการทำงานระหว่าง DOM และ SAX (Haw and Rao, 2007) ได้ดังตารางที่ 2.2

ตารางที่ 2.2 เปรียบเทียบการทำงานระหว่าง DOM และ SAX

เรื่อง	DOM	SAX
วิธีการสำรวจข้อมูล	Tree-base Parser	Event-base Parser
การอ่านข้อมูล	อ่านทั้งหมดเพียงครั้งเดียว เก็บไว้ในหน่วยความจำ ทำให้สามารถใช้งานได้ตลอด โดยไม่ต้องอ่านข้อมูลซ้ำอีก จึงสะดวกในการทำงาน กรณีต้องการทำงานกับข้อมูลมากกว่า 1 ครั้ง	อ่านข้อมูลทีละชุด ทำให้ต้องอ่านข้อมูลซ้ำเมื่อต้องการข้อมูลชุดใหม่
การดึงข้อมูล	ต้องอ่านเอกสารทั้งหมดก่อน ถึงจะดึงข้อมูลได้	สามารถดึงข้อมูลเฉพาะที่ต้องการได้
หน่วยความจำ	ใช้หน่วยความจำค่อนข้างมากเพราะต้องอ่านเอกสารทั้งหมดเก็บไว้เป็นโครงสร้างต้นไม้ในหน่วยความจำ	ไม่มีการโหลดข้อมูลทั้งหมดเข้าในหน่วยความจำ ทำให้ใช้หน่วยความจำน้อยกว่า DOM
วิธีการเข้าถึงข้อมูล	สามารถเข้าถึงแบบสุ่ม (Random Access) ได้	เข้าถึงแบบ Sequential เท่านั้น ไม่มีการเข้าถึงข้อมูลแบบสุ่ม

ตารางที่ 2.2 เปรียบเทียบการทำงานระหว่าง DOM และ SAX (ต่อ)

เรื่อง	DOM	SAX
การจัดการข้อมูล	สามารถเพิ่ม ลบ แก้ไข และเปลี่ยนแปลงโครงสร้างเอกสาร XML ได้	อ่านได้อย่างเดียว
ความเร็วในการเข้าถึงข้อมูล	การเรียกใช้งานครั้งแรกจะช้า หลังจากนั้นการเข้าถึงจุดต่าง ๆ จะเร็วขึ้นเพราะข้อมูลถูกเก็บอยู่ในหน่วยความจำไว้แล้ว	ทำงานเร็วกว่า DOM ในการเข้าถึงชุดข้อมูลที่ต้องการ
ขนาดเอกสาร	ทำงานได้ดีกับเอกสารขนาดเล็ก เพราะเอกสารขนาดใหญ่ต้องใช้หน่วยความจำและเวลามากขึ้นตามขนาดของเอกสาร	ทำงานได้ดีกับเอกสารขนาดใหญ่ โดยเฉพาะกรณีมีข้อจำกัดของทรัพยากร
โครงสร้างเอกสาร XML	ทราบรายละเอียดและโครงสร้างของเอกสาร XML ทั้งหมด	ไม่ทราบรายละเอียดของโครงสร้างเอกสาร XML ทั้งหมด
การเขียนโปรแกรม	เขียนโปรแกรมง่าย (เข้าใจง่าย)	เขียนโปรแกรมยาก (เข้าใจยาก)

2.2.5 การแสดงผลเอกสาร XML

เอกสาร XML เป็นไฟล์ข้อความธรรมดา ทำให้ไม่สามารถแสดงผลในรูปแบบที่ผู้ใช้ต้องการได้ จึงต้องอาศัยเทคโนโลยีอื่น ๆ เข้ามาเกี่ยวข้อง ได้แก่ CSS (Cascading Style Sheets) และ XSL (Extensible Stylesheet Language) เป็นต้น

1) CSS (Cascading Style Sheets)

CSS เป็นเทคโนโลยีที่ได้รับการพัฒนาขึ้นสำหรับจัดรูปแบบการแสดงผลเอกสาร HTML แต่สามารถนำมาประยุกต์ใช้งานร่วมกับ XML ได้ โดย CSS จะช่วยในการจัดการองค์ประกอบต่าง ๆ เช่น ขนาดตัวอักษร ลักษณะตัวอักษร เป็นต้น การใช้งาน CSS ร่วมกับ XML จะต้องระบุคำสั่ง ดังภาพประกอบ 2.11 ไว้ในเอกสาร XML เพื่อเรียกใช้งานเอกสาร CSS

```
<?xml-stylesheet type="text/css" href="ไฟล์ css" />
```

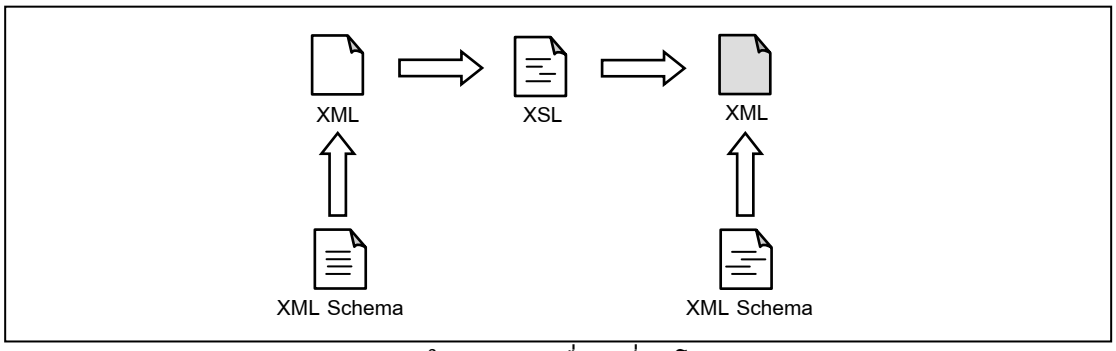
ภาพประกอบ 2.11 รูปแบบการเรียกใช้ไฟล์ CSS เพื่อแสดงผลเอกสาร XML

2) XSL (Extensible Stylesheet Language)

XSL เป็นเทคโนโลยีที่นำมาใช้จัดการรูปแบบการแสดงผลและแปลงเอกสาร XML ให้อยู่ในรูปแบบเอกสารที่ต้องการ โดย XSL ประกอบด้วย 2 ส่วน ดังนี้ (สุธี พงศาสกุลชัย, 2550)

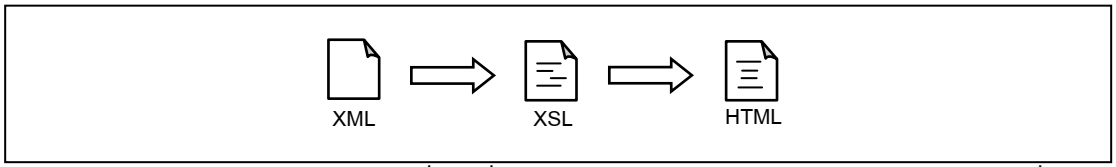
2.1) ภาษาสำหรับแปลงรูปแบบเอกสาร (XSL Transformations) ทำหน้าที่แปลงเอกสาร XML ต้นฉบับให้อยู่ในรูปแบบที่ต้องการ ถูกกำหนดเป็นมาตรฐานโดยองค์กร W3C เรียกว่า “XSLT (Extensible Stylesheet Language Transformations)” การแปลงรูปแบบเอกสาร XML ใช้แนวคิดการจับคู่ตามรูปแบบ (Pattern Matching) และแม่แบบ (Templates) โดยขณะที่เอกสาร XML ถูกประมวลผลตัวประมวลผล XSL (XSL Processor) จะค้นหารูปแบบ (Pattern) และข้อมูลในเอกสาร XML ตามตำแหน่งที่ระบุด้วยไวยากรณ์ของ XPath (Clark *et al*., 1999: Online) และจับคู่รูปแบบกับข้อมูลในเอกสาร XML ที่ตรงกัน จากนั้นจึงแสดงผลลัพธ์ตามรูปแบบที่กำหนด โดย XSLT มีหน้าที่หลัก 2 ประการ คือ

2.1.1) Structural Transformation คือ การแปลงโครงสร้างเอกสาร XML จากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่ง ดังแสดงในภาพประกอบ 2.12



ภาพประกอบ 2.12 การใช้ XSLT เพื่อเปลี่ยนโครงสร้างเอกสาร XML

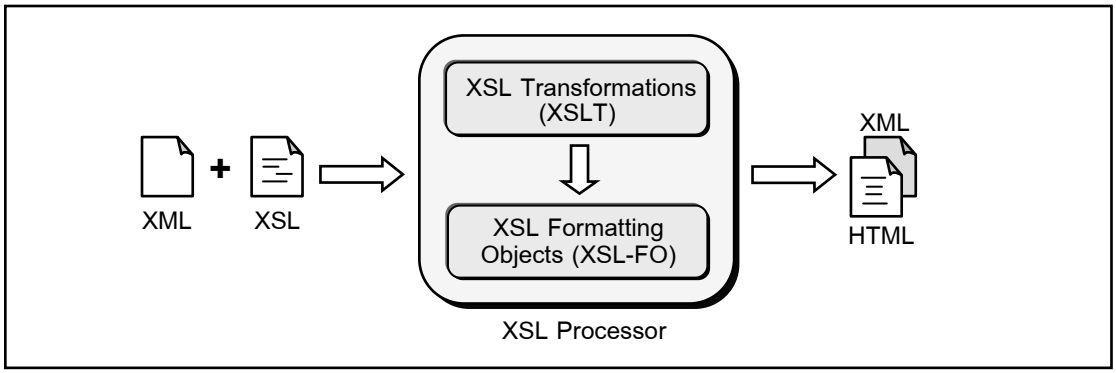
2.1.2) Aesthetic Transformation คือ การเปลี่ยนรูปแบบเอกสาร XML ให้แสดงผลในรูปแบบอื่น ๆ ที่สามารถเข้าใจได้ เช่น เปลี่ยนเอกสาร XML เป็น HTML แสดงดังภาพประกอบ 2.13 เป็นต้น



ภาพประกอบ 2.13 การใช้ XSLT เพื่อเปลี่ยนการแสดงผลเอกสาร XML ให้อยู่ในรูปแบบอื่น ๆ

2.2) ภาษาสำหรับจัดรูปแบบ (XSL Formatting Objects: XSL-FO) ทำหน้าที่จัดรูปแบบเอกสารซึ่งมีลักษณะการทำงานคล้ายกับ CSS

การทำงานโดยรวมของ XSL สามารถอธิบายได้ดังภาพประกอบ 2.14



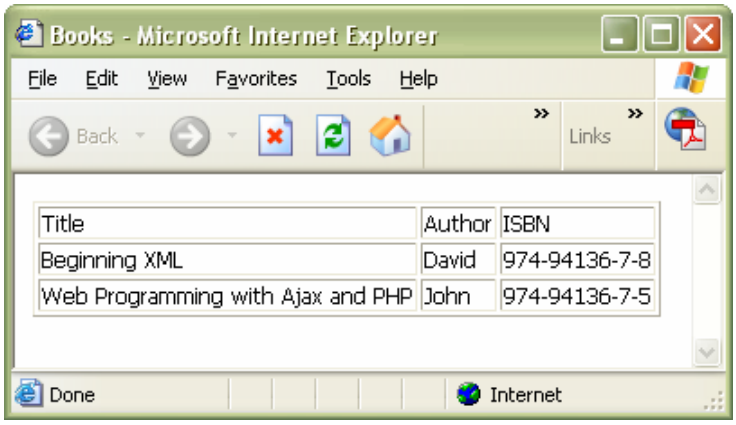
ภาพประกอบ 2.14 กระบวนการทำงานของ XSL

จากภาพประกอบ 2.14 แสดงกระบวนการทำงานของ XSL โดยตัวประมวลผล XSL ประมวลผลเอกสาร XML และเอกสาร XSL เพื่อเปลี่ยนรูปแบบเอกสาร XML ด้วย XSLT และส่งต่อไปให้ XSL-FO สำหรับจัดรูปแบบเอกสารตามที่ได้ระบุไว้ในเอกสาร XSL จากนั้นจึงแสดงผลลัพธ์ ตัวอย่างเอกสาร XSL แสดงดังภาพประกอบ 2.15

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <head><title>Books</title></head>
  <body>
    <xsl:for-each select="books">
    <table border="1">
      <tr><th>Title</th><th>Author</th><th>ISBN</th></tr>
      <xsl:for-each select="book">
      <tr>
        <td><xsl:for-each select="title"><xsl:apply-templates /></xsl:for-each></td>
        <td><xsl:for-each select="author"><xsl:apply-templates /></xsl:for-each></td>
        <td><xsl:for-each select="@isbn"><xsl:value-of select="." /></xsl:for-each></td>
      </tr>
      </xsl:for-each>
    </table>
    </xsl:for-each>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

ภาพประกอบ 2.15 ตัวอย่างเอกสาร XSL

จากเอกสาร XML ดังแสดงในภาพประกอบ 2.7 เมื่อนำ XSL ดังภาพประกอบ 2.15 มาจัดรูปแบบ และแสดงผลด้วย Microsoft Internet Explorer จะได้ผลลัพธ์ดังภาพประกอบ 2.16



ภาพประกอบ 2.16 ผลลัพธ์จากการจัดรูปแบบการแสดงผลเอกสาร XML ด้วย XSL

2.2.6 XPath

XPath (XML Path Language) คือ ภาษาที่ใช้ระบุหรือสืบค้นข้อมูลภายในเอกสาร XML กำหนดมาตรฐานโดย W3C ซึ่งเป็นสิ่งสำคัญสำหรับ XSLT เพื่อระบุข้อมูลที่ต้องการแสดงผล สัญลักษณ์ต่าง ๆ ที่ใช้ใน XPath แสดงดังตารางที่ 2.3 โดยแต่ละส่วนของเอกสาร XML จะถูกมองเป็นโหนดของวัตถุ

ตารางที่ 2.3 สัญลักษณ์ต่าง ๆ ของ XPath

สัญลักษณ์	คำอธิบาย
nodename	เข้าถึงโหนดลูกทุกตัวของโหนดที่มีชื่อเช่นเดียวกับชื่อที่กำหนด
/	การอ้างถึงโหนดราก หรือ Root Element ของเอกสาร XML
//	การอ้างโหนดโดยไม่สนใจว่าข้อมูลจะอยู่ที่ใดในเอกสาร XML
.	การอ้างถึงโหนดปัจจุบัน
..	การอ้างถึงโหนดพ่อแม่ของโหนดปัจจุบัน
@	เข้าถึงสมาชิกที่เป็นแอททริบิวต์ของอิลิเมนต์

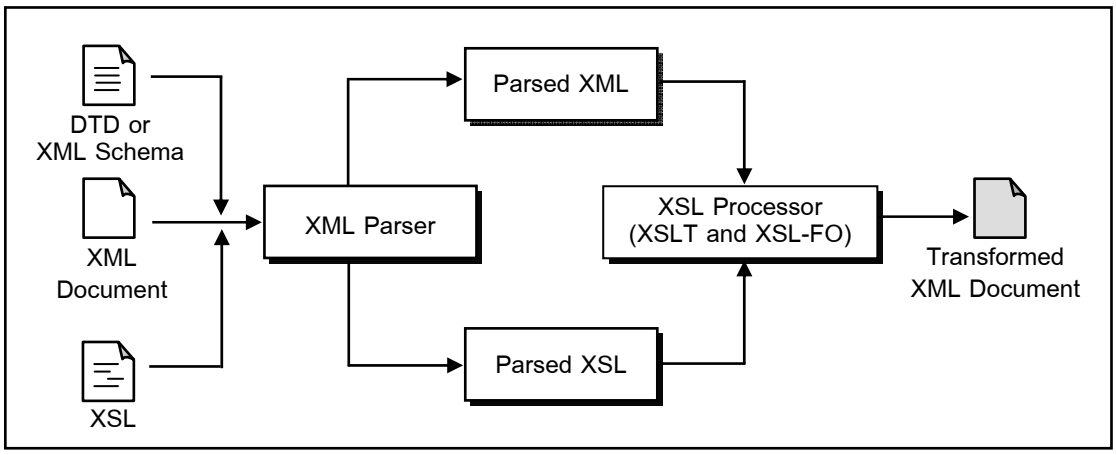
ตัวอย่างการระบุข้อมูลที่ต้องการภายในเอกสาร XML ด้วย XPath แสดงดังตารางที่ 2.4

ตารางที่ 2.4 ตัวอย่างการระบุข้อมูลด้วย XPath

ตัวอย่าง	คำอธิบาย
bookstore	เข้าถึงโหนดลูกทุกตัวของอิลิเมนต์ bookstore
/bookstore	เข้าถึงโหนด bookstore โดยโหนด bookstore ต้องเป็น Root Element
bookstore/book	เข้าถึงอิลิเมนต์ book ที่เป็นสมาชิกของ bookstore
//book	เข้าถึงอิลิเมนต์ book ทุกตัวไม่ว่าจะอยู่ที่ไหนในเอกสาร XML
bookstore//book	เข้าถึงอิลิเมนต์ book ที่เป็นอิลิเมนต์ลูกของ bookstore จากทุก ๆ ตำแหน่งที่มีลักษณะดังกล่าว
//@lang	เข้าถึงอิลิเมนต์ทุกตัวที่มีชื่อแอททริบิวต์ว่า lang

2.2.7 การประมวลผลเอกสาร XML

การประมวลผลเอกสาร XML โดยทั่วไปมีกระบวนการทำงานดังภาพประกอบ 2.17



ภาพประกอบ 2.17 การประมวลผลเอกสาร XML

จากภาพประกอบ 2.17 เอกสาร XML ถูกส่งเข้าไปประมวลด้วย XML Parser พร้อมกับเอกสาร DTD หรือ XML Schema และเอกสาร XSL เพื่อตรวจสอบโครงสร้างและกำหนดรูปแบบผลลัพธ์ โดยผลลัพธ์ที่ได้จาก XML Parser จะถูกส่งไปยัง XSL Processor เพื่อแปลงโครงสร้างและจัดรูปแบบการแสดงผลตามที่ได้กำหนดไว้ในเอกสาร XSL ผลลัพธ์สุดท้ายที่ได้ คือ เอกสาร XML ที่ถูกแปลงให้อยู่ในรูปแบบที่ผู้ใช้ต้องการ

2.3 การนิยามโครงสร้างเอกสาร XML

การนิยามโครงสร้างเอกสาร XML คือ การกำหนดรูปแบบโครงสร้างเอกสาร XML ว่าประกอบด้วยอะไรบ้าง ซึ่งจะถูกนำไปเป็นกฎเกณฑ์ในการตรวจสอบไวยากรณ์ของเอกสาร XML โดยเรียกเอกสาร XML ที่มีรูปแบบตามโครงสร้างที่นิยามไว้ว่า เอกสาร XML นั้นมีคุณสมบัติ “Valid” โดยภาษาสำหรับนิยามโครงสร้าง (Schema Language) สามารถแบ่งออกเป็น 2 ประเภท คือ Grammar-based Language ได้แก่ DTD, XML Schema และ RELAX NG เป็นต้น ส่วนอีกประเภท คือ Rule-based Language ได้แก่ Schematron และ xlinkit เป็นต้น (Marinelli *et al*., 2004) ซึ่งในที่นี้จะกล่าวถึง DTD และ XML Schema เพราะเป็นวิธีนิยามโครงสร้างเอกสาร XML ที่เป็นที่รู้จักและใช้งานแพร่หลาย

2.3.1 DTD (Document Type Definition)

DTD มีองค์ประกอบหลัก ๆ ได้แก่ Elements, Attributes, Entities, PCDATA และ CDATA โดยการนิยามโครงสร้างขององค์ประกอบต่าง ๆ สามารถทำได้ดังนี้

1) การประกาศอิลิเมนต์จะใช้คีย์เวิร์ด ELEMENT ในการประกาศชื่อ ชนิด และอิลิเมนต์ย่อยของอิลิเมนต์ต่าง ๆ ตัวอย่างของการนิยามโครงสร้างอิลิเมนต์ด้วย DTD และเอกสาร XML แสดงดังภาพประกอบ 2.18

<!ELEMENT note (to, from, heading, body)> <!ELEMENT to (#PCDATA)> <!ELEMENT from (#PCDATA)> <!ELEMENT heading (#PCDATA)> <!ELEMENT body (#PCDATA)>	<?xml version=“1.0”?> <!DOCTYPE note SYSTEM “note.dtd”> <note> <to>Tove</to> <from>Jani</from> <heading>Reminder</heading> <body>Don’t forget me this weekend!</body> </note>
note.dtd	**exam.xml**

ภาพประกอบ 2.18 ตัวอย่างการประกาศอิลิเมนต์

จากภาพประกอบ 2.18 เอกสาร exam.xml นิยามโครงสร้างตามเอกสาร note.dtd ดังนั้นคอมพิวเตอร์สามารถเข้าใจได้ว่า note เป็น Root Element ที่ประกอบด้วยอิลิเมนต์ย่อย คือ <to> <from> <heading> และ <body> โดยทุกอิลิเมนต์ย่อยกำหนดให้เป็น PCDATA (Parsed Character Data) คือ กลุ่มข้อมูลที่ต้องการให้มีการวิเคราะห์หรือประมวลผล

ด้วย XML Parser เพื่อแยกแยะว่าอะไรคือข้อมูล อะไรคืออิลิเมนต์ มีการจัดวางถูกต้องตามกฎเกณฑ์ที่กำหนดไว้หรือไม่ ซึ่งโดยปกติอาจกำหนดเป็น CDATA (Character Data) ได้เช่นกัน คือ กำหนดให้เป็นกลุ่มข้อมูลที่ไม่ต้องการให้มีการวิเคราะห์หรือประมวลผลด้วย XML Parser ได้แก่ สคริปต์ หรือฟังก์ชันการทำงานบางอย่างที่ไม่ใช่มาตรฐานของ XML ซึ่งอาจมีเครื่องหมายที่ทำให้การวิเคราะห์โครงสร้างเกิดความผิดพลาดขึ้น

2) การประกาศแอททริบิวต์จะใช้คีย์เวิร์ด ATTLIST ในการประกาศชื่อ ค่า และ/หรือค่าตั้งต้น (Default) ของแอททริบิวต์ เพื่อกำหนดข้อมูลเพิ่มเติมให้กับอิลิเมนต์

```
1   <?xml version="1.0" encoding = "window-874">
2   <!DOCTYPE books [
3   <!ELEMENT books(book)
4   <!ELEMENT> book EMPTY>
5   <!ATTLIST book ISBN CDATA "000-00000-0-0" >
6   ]>
7   <books>
8       <book ISBN = "974-94832-8-6" />
9       <book ISBN = "974-94743-0-9" />
10      <book ISBN = "974-94452-1-2" />
11  </books>
```

ภาพประกอบ 2.19 ตัวอย่างการประกาศแอททริบิวต์

จากภาพประกอบ 2.19 บรรทัดที่ 5 คือ การกำหนดแอททริบิวต์ที่ชื่อว่า ISBN ให้กับอิลิเมนต์ book และกำหนดให้เป็น CDATA นอกจากนี้ยังสามารถกำหนดค่าตั้งต้นให้กับแอททริบิวต์ได้ดังแสดงในตารางที่ 2.5

ตารางที่ 2.5 ค่าตั้งต้นของแอททริบิวต์

ค่าตั้งต้น	คำอธิบาย
#REQUIRED	เป็นแอททริบิวต์บังคับที่ต้องระบุให้กับอิลิเมนต์
#IMPLIED	เป็นแอททริบิวต์ที่ไม่ได้บังคับ จะมีหรือไม่ก็ได้
#FIXED value	เป็นการกำหนดค่าคงที่ให้กับแอททริบิวต์
Default	เป็นค่าตั้งต้นของแอททริบิวต์ซึ่งถูกดึงมาใช้งานอัตโนมัติ หากไม่มีการกำหนดค่าเป็นอย่างอื่น

3) การประกาศ Entity จะใช้คีย์เวิร์ด ENTITY ในการประกาศชื่อและค่า

```
1   <?xml version="1.0"  encoding="UTF-8"?>
2   <!— filename=detailMessage.xml -->
3   <! DOCTYPE message [
4       <!ELEMENT message (person)>
5       <!ELEMENT person (to, data)>
6       <!ELEMENT to (#PCDATA)>
7       <!ELEMENT data (#PCDATA)>
8       <!ENTITY text "Latest News">
9   ]>
10  <message>
11      <person>
12          <to>Wichuta</to>
13          <data>&text;</data>
14      <person>
12  </message>
```

ภาพประกอบ 2.20 ตัวอย่างการประกาศ Entity

จากภาพประกอบ 2.20 บรรทัดที่ 3-9 คือ ส่วนของ DTD ซึ่งมีการกำหนด Entity ที่ชื่อว่า text ไว้ในบรรทัดที่ 8 สำหรับเก็บข้อความ "Latest News" โดยมีการเรียกใช้ในบรรทัดที่ 13 เพื่อกำหนดค่าให้กับ อิลิเมนต์ <data> การเรียกใช้ Entity ต้องขึ้นต้นด้วยเครื่อง & ตามด้วยชื่อ Entity และจบด้วยเครื่องหมาย ; เช่น &text;

2.3.2 XML Schema

การนิยามโครงสร้างเอกสาร XML ด้วย DTD ยังมีข้อจำกัดในด้านต่าง ๆ ดังแสดงในตารางที่ 2.6 ทำให้ยากต่อการเรียนรู้และนำไปใช้งาน ต่อมา W3C ได้กำหนดมาตรฐานสำหรับตรวจสอบโครงสร้างเอกสาร XML ขึ้นมาใหม่ เรียกว่า "XML Schema" ทำให้สามารถจัดการกับโครงสร้างเอกสาร XML ได้ดียิ่งขึ้น เนื่องจาก XML Schema ถูกเขียนด้วยภาษา XML จึงสามารถนำไปประยุกต์ใช้งานร่วมกับเทคโนโลยีที่เกี่ยวข้องกับ XML ได้ อีกทั้งยังรองรับการใช้งาน Namespace และชนิดข้อมูลได้มากขึ้นอีกด้วย

ตารางที่ 2.6 เปรียบเทียบ DTD กับ XML Schema

DTD	XML Schema
- รองรับชนิดข้อมูลได้น้อยกว่า XML Schema - ไม่ได้อยู่ภายใต้ไวยากรณ์ของ XML - ไม่รองรับการใช้งาน Namespace	- รองรับชนิดข้อมูลได้มากกว่า DTD - อยู่ภายใต้ไวยากรณ์ของ XML - รองรับการใช้ Namespace - กำหนดจำนวนและลำดับของอิลิเมนต์ลูกได้

1) โครงสร้างของ XML Schema

XML Schema ประกอบด้วยส่วนต่าง ๆ ดังต่อไปนี้ (วิทูร ชื่นวชิรศิริ, 2550)

1.1) Schema Element คือ ส่วนที่บอกว่าเป็นไฟล์ XML Schema รวมทั้งใช้สำหรับระบุ URL ของ Namespace ที่มีรายละเอียดคำศัพท์ของ XML Schema แสดงดังภาพประกอบ 2.21

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

ภาพประกอบ 2.21 Schema Element

1.2) Element เป็นการประกาศชื่อ และชนิดข้อมูลของอิลิเมนต์ โดยสามารถแบ่งได้ดังนี้

1.2.1) Simple Type Element เป็นอิลิเมนต์ที่มีเฉพาะข้อมูลเท่านั้นไม่มีอิลิเมนต์ย่อยหรือแอททริบิวต์

1.2.2) Complex Type Element เป็นอิลิเมนต์ที่มีแอททริบิวต์ หรืออิลิเมนต์อื่น ๆ อยู่ภายใน

XML:

```
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

XML Schema:

```
1   <xs:element name="note">
2       <xs:complexType>
3           <xs:sequence>
4               <xs:element name="to" type="xs:string"/>
5               <xs:element name="from" type="xs:string"/>
7               <xs:element name="heading" type="xs:string"/>
8               <xs:element name="body" type="xs:string"/>
9           </xs:sequence>
10      </xs:complexType>
11  </xs:element>
```

ภาพประกอบ 2.22 Simple Typle Element และ Complex Type Element

XML Schema ดังแสดงในภาพประกอบ 2.22 สามารถอธิบายได้ดังนี้

บรรทัดที่ 1-11 คือ การนิยามโครงสร้างของอิลิเมนต์ที่ชื่อว่า note ซึ่งเป็นอิลิเมนต์แบบ Complex Type Element ที่มีอิลิเมนต์ย่อยอยู่ภายใน

บรรทัดที่ 2-10 คือ การนิยามอิลิเมนต์ย่อยของอิลิเมนต์ note ได้แก่ to, from heading และ body โดยอิลิเมนต์ย่อยเหล่านี้เป็นอิลิเมนต์แบบ Simple Type Element เพราะไม่มีอิลิเมนต์ย่อยอยู่ภายใน

1.3) Attribute เป็นการประกาศชื่อ และชนิดข้อมูลของแอททริบิวต์ โดยแอททริบิวต์จะถูกนิยามในรูปของ Simple Type Element ซึ่ง Complex Type Element เท่านั้นที่สามารถมีแอททริบิวต์ได้ โดยแอททริบิวต์จะถูกนิยามไว้หลังอิลิเมนต์ทั้งหมด และแอททริบิวต์ที่นิยามขึ้นจะเป็นของอิลิเมนต์ที่อาศัยอยู่

XML:

```
<person sex="Female">Wichuta Kaewnopparat</person>
```

XML Schema:

```
<xs:element name="person">
    <xs:complexType mixed="true">
        <xs:attribute name="sex" type="xs:string" />
    </xs:complexType>
</xs:element>
```

ภาพประกอบ 2.23 ตัวอย่างการนิยามแอททริบิวต์

จากภาพประกอบ 2.23 กำหนดให้อิลิเมนต์ person ประกอบด้วยแอททริบิวต์ที่ชื่อว่า sex และกำหนดให้มีชนิดข้อมูลเป็น string

นอกจากชื่อและชนิดข้อมูลแล้ว ยังสามารถกำหนดค่าเพิ่มเติมให้กับแอททริบิวต์ได้ดังนี้

use ใช้สำหรับระบุระดับความจำเป็นของแอททริบิวต์ โดยค่าที่เป็นไปได้คือ require (ต้องมี) optional (มีหรือไม่ก็ได้) และ prohibited (ไม่ต้องมี)

default ใช้สำหรับกำหนดค่าตั้งต้นของแอททริบิวต์

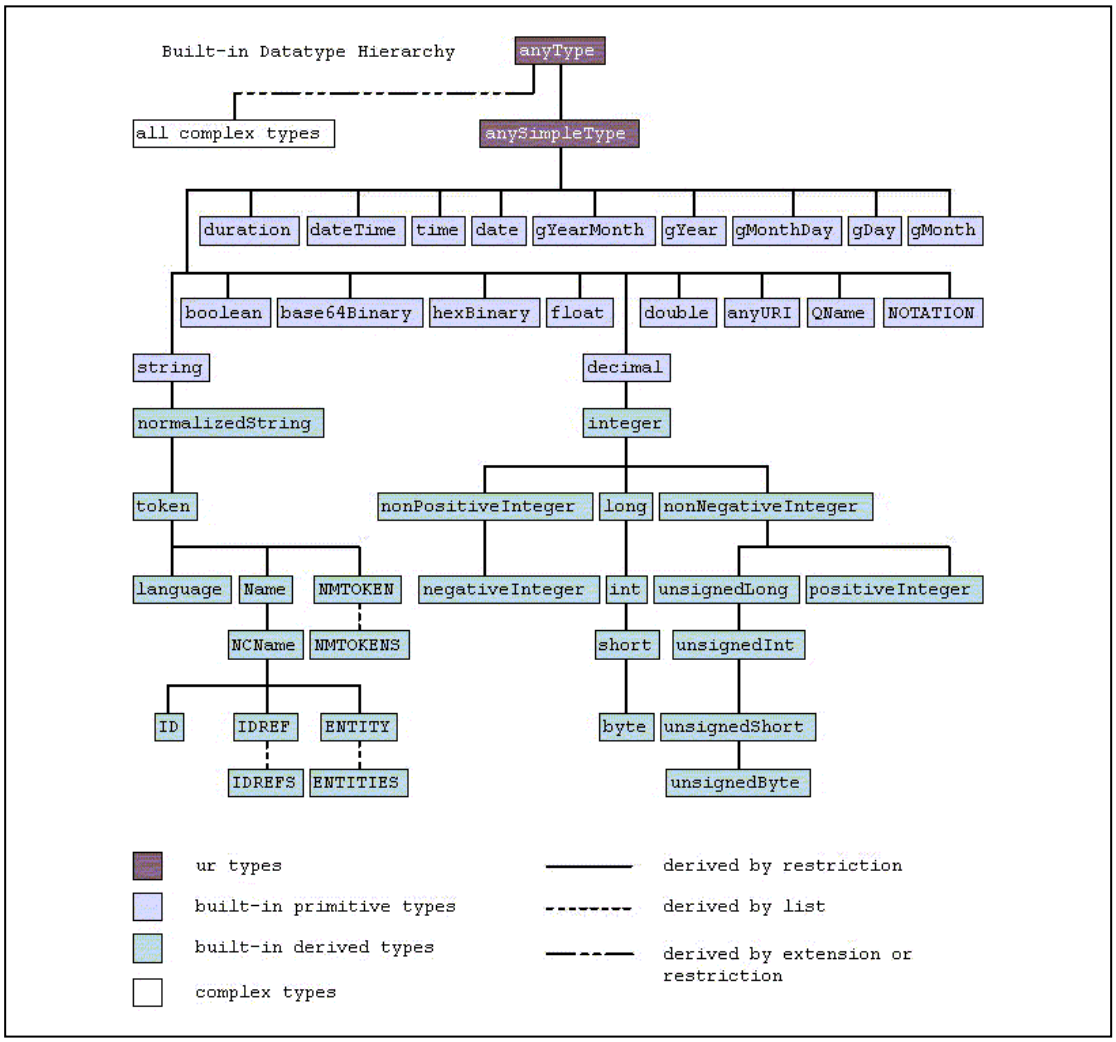
fixed ใช้สำหรับกำหนดค่าคงที่ให้กับแอททริบิวต์

1.4) Annotations ใช้สำหรับอธิบายรายละเอียดเพิ่มเติมที่เป็นประโยชน์ต่อผู้พัฒนาโปแกรมและแอพพลิเคชัน ซึ่งระบุไว้ในส่วน documentation และ appInfo ตามลำดับ ตัวอย่างแสดงดังภาพประกอบ 2.24

```
<xsd:annotation>
        <xsd:documentation xml:lang="en">
                Here is the documentation text for our schema
        </xsd:documentation>
</xsd:annotation>
```

ภาพประกอบ 2.24 ตัวอย่างการนิยาม Annotations

1.5) Datatype คือ ชนิดข้อมูลในการนิยามโครงสร้างเอกสาร XML โดย Simple Type ของ XML Schema แสดงดังภาพประกอบ 2.25



ภาพประกอบ 2.25 Simple Type ของ XML Schema (Biron *et. al.*, 2004: Online)

ตัวอย่างชนิดข้อมูลและค่าที่เป็นไปได้ของ XML Schema แสดงดังตารางที่ 2.7

ตารางที่ 2.7 ตัวอย่างชนิดข้อมูล

Data Type	ตัวอย่าง
xs:string	Hello, Wichuta
xs:dicimal	0, 4.567
xs:integer	-4, 58990
xs:boolean	TURE, FALSE, 0, 1
xs:date	2009-04-15
xs:time	16:20

2) การสร้าง Simple Type ด้วย Facet

Facet คือ การสร้างข้อกำหนดให้กับอิลิเมนต์ที่เป็นชนิด Simple Type ขึ้นมาใหม่ ตัวอย่าง Facet ที่ถูกนำมาใช้กับข้อมูลประเภทข้อความ ได้แก่ pattern (กำหนดรูปแบบ) enumeration (กำหนดค่าที่เป็นไปได้) length (กำหนดความยาว) minLength (กำหนดความยาวต่ำสุด) และ maxLength (กำหนดความยาวสูงสุด) เป็นต้น

```
1   <xs:element name="note">
2       <xs:complexType>
3           <xs:sequence>
4               <xs:element name="name" type="xs:string"/>
5               <xs:element name="telephone" type="TelephoneNumber"/>
6           </xs:sequence>
7       </xs:complexType>
8   </xs:element>
9   <xs:simpleType name="TelephoneNumber" >
10      <xs:restriction base="xs:string">
11          <xs:length value="11">
12          <xs:pattern value="\d{1}-\d{4}-\d{4}">
13      </xsd:restriction>
14  </xsd:simpleType>
```

ภาพประกอบ 2.26 การสร้างข้อกำหนดให้อิลิเมนต์ด้วย Facet

จากภาพประกอบ 2.26 สามารถอธิบายส่วนต่าง ๆ ได้ดังนี้

บรรทัดที่ 1-8 คือ การกำหนดอิลิเมนต์ที่ชื่อว่า note ประกอบด้วยอิลิเมนต์ย่อย คือ name มีชนิดข้อมูลเป็น string และอิลิเมนต์ telephone มีชนิดข้อมูลเป็น TelephoneNumber ซึ่งเป็นชนิดข้อมูลที่ผู้ใช้กำหนดเอง ระบุไว้ในบรรทัดที่ 9-14

บรรทัดที่ 9-14 เป็นการกำหนดชนิดข้อมูลที่ชื่อว่า TelephoneNumber เพื่อเก็บหมายเลขโทรศัพท์

บรรทัดที่ 10 คือ การกำหนดให้ชนิดข้อมูลพื้นฐานที่ใช้เป็น String

บรรทัดที่ 11 กำหนดความยาวของหมายเลขโทรศัพท์เท่ากับ 11 ตัวอักษร

บรรทัดที่ 12 กำหนดรูปแบบของหมายเลขโทรศัพท์เป็นข้อความชนิดตัวเลข เช่น 0-8123-4567

2.3.3 SchemaPath

SchemaPath คือ ส่วนขยายของ XML Schema ถูกออกแบบมาเพื่อใช้ในการจัดการเงื่อนไขข้อบังคับ สำหรับกำหนดชนิดข้อมูลให้กับอิลิเมนต์และแอททริบิวต์ ซึ่ง XML Schema ไม่ได้จัดเตรียมการทำงานส่วนนี้ไว้ ยกตัวอย่างเช่น ถ้ากำหนดแอททริบิวต์ A ให้กับอิลิเมนต์แล้วห้ามมีแอททริบิวต์ B ปรากฏในอิลิเมนต์นั้นอีก หรือการกำหนดชนิดข้อมูลให้กับอิลิเมนต์ตามค่าที่ระบุในแอททริบิวต์ เป็นต้น (Marinelli *et al*., 2004)

1) ไวยากรณ์ของ SchemaPath

SchemaPath ใช้ไวยากรณ์ของ XPath เพื่อระบุเงื่อนไขที่ต้องการ และยังคงใช้ไวยากรณ์ของ XML Schema เพื่อนิยามโครงสร้างเอกสาร XML โดยเพิ่มไวยากรณ์ ดังแสดงในตารางที่ 2.8 เพื่อกำหนดเงื่อนไขต่าง ๆ

ตารางที่ 2.8 ไวยากรณ์ของ SchemaPath

อิลิเมนต์	แอททริบิวต์	คำอธิบาย
<alt>		ใช้เพื่อประกาศเงื่อนไขสำหรับกำหนดชนิดข้อมูลให้กับอิลิเมนต์และแอททริบิวต์
	cond	ใช้สำหรับกำหนดเงื่อนไขด้วยการระบุไวยากรณ์ของ XPath
	type	ใช้สำหรับระบุชนิดข้อมูลของ XML Schema ตามเงื่อนไขที่กำหนด
	priority	ใช้สำหรับระบุลำดับความสำคัญของการกำหนดชนิดข้อมูลให้กับอิลิเมนต์และแอททริบิวต์ ในกรณีที่หลายเงื่อนไขมีโอกาสเกิดขึ้นพร้อมกัน
xsd:error		ใช้สำหรับแจ้งข้อผิดพลาดที่เกิดขึ้นในขั้นตอนตรวจสอบความถูกต้องของเอกสาร XML

ตัวอย่างการนิยามโครงสร้างอิลิเมนต์และแอททริบิวต์ด้วย SchemaPath แสดงดังภาพประกอบ 2.27 และ 2.28

```
1   <xsd:element name="x">
2       <xsd:alt cond="(@a and @b)"   type="myType"/>
3       <xsd:alt                      type="xsd:string"/>
4   </xsd:element>
```

ภาพประกอบ 2.27 การนิยามชนิดข้อมูลของอิลิเมนต์จากแอททริบิวต์ที่กำหนด

จากภาพประกอบ 2.27 เป็นการกำหนดชนิดข้อมูลให้กับอิลิเมนต์ที่ชื่อว่า x โดยมีเงื่อนไข คือ ถ้ากำหนดแอททริบิวต์ a และ b ให้กับอิลิเมนต์ x แล้ว ให้ชนิดข้อมูลของอิลิเมนต์ x คือ myType แต่ถ้าไม่ใช่กำหนดให้อิลิเมนต์ x มีชนิดข้อมูลเป็น string

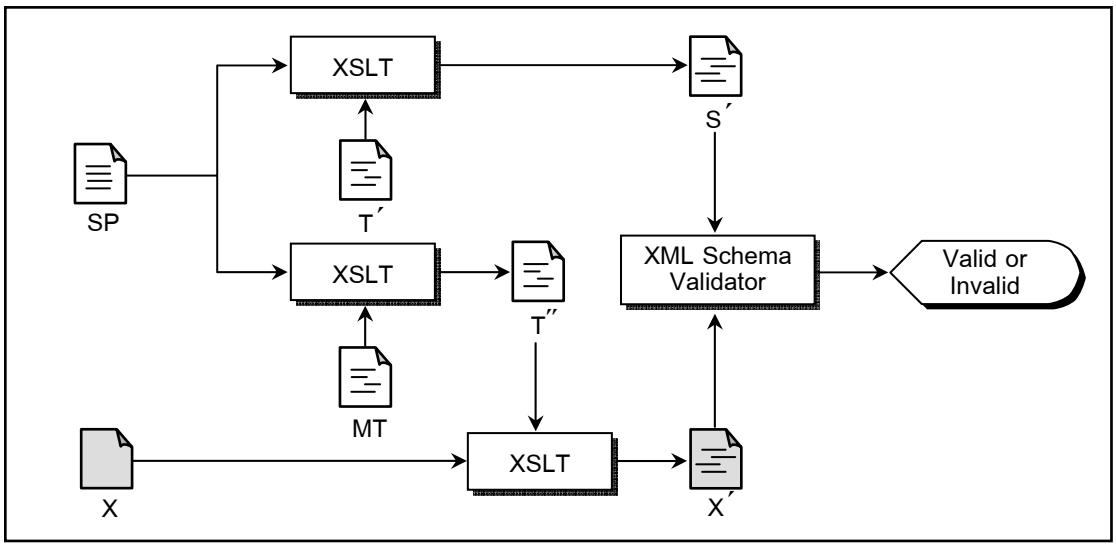
```
1   <xsd:element name="unit" type="unitType"/>
2   <xsd:element name="quantity">
3       <xsd:alt cond="../unit='items'"   type="xsd:integer"/>
4       <xsd:alt cond="../unit='meters'" type="xsd:decimal"/>
5   </xsd:element>
```

ภาพประกอบ 2.28 การนิยามชนิดข้อมูลของอิลิเมนต์จากค่าที่กำหนดให้กับอิลิเมนต์อื่น

จากภาพประกอบ 2.28 เป็นการกำหนดชนิดข้อมูลให้กับอิลิเมนต์ที่ชื่อว่า quantity โดยมีเงื่อนไขตามค่าที่ระบุไว้ในอิลิเมนต์ unit คือ ถ้าอิลิเมนต์ unit มีค่าเท่ากับ items ให้อิลิเมนต์ quantity มีชนิดข้อมูลเป็น integer แต่ถ้าอิลิเมนต์ unit มีค่าเท่ากับ meters ให้อิลิเมนต์ quantity มีชนิดข้อมูลเป็น decimal

2) การตรวจสอบความถูกต้องของเอกสาร XML ด้วย SchemaPath

เนื่องจาก SchemaPath ยังไม่ได้เป็นข้อกำหนดมาตรฐาน ทำให้ไม่สามารถนำนิยามโครงสร้างที่เขียนด้วยไวยากรณ์ของ SchemaPath ไปตรวจสอบความถูกต้องของเอกสาร XML กับ XML Schema Validator ทั่วไปที่มีได้ ดังนั้นการตรวจสอบความถูกต้องของเอกสาร XML ด้วย SchemaPath จึงต้องใช้ XSLT เพื่อแปลงโครงสร้างเอกสาร XML และเอกสาร SchemaPath ให้อยู่ในรูปแบบที่สามารถนำไปใช้งานกับ XML Schema Validator ที่มีอยู่ทั่วไปได้



ภาพประกอบ 2.29 การตรวจสอบความถูกต้องเอกสาร XML ด้วย SchemaPath

จากภาพประกอบ 2.29 กระบวนการตรวจสอบความถูกต้องเอกสาร XML ประกอบด้วยเอกสาร XSL ที่เกี่ยวข้อง 3 เอกสาร คือ T´ ใช้สำหรับแปลงโครงสร้างเอกสาร SchemaPath (SP) ไปเป็นเอกสาร XML Schema ทั่วไป และ MT ใช้สำหรับตรวจสอบกฎที่กำหนดไว้ในเอกสาร SchemaPath (SP) และนำมาสร้างเป็นแม่แบบ (Template) เก็บไว้ในเอกสาร T″ เพื่อนำไปแปลงโครงสร้างเอกสาร XML ซึ่งเอกสารอื่น ๆ ที่เกี่ยวข้องสามารถดาวน์โหลดได้ที่เว็บไซต์ http://tesi.fabio.web.cs.unibo.it/Tesi/SchemaPathImplementations โดยขั้นตอนการตรวจสอบความถูกต้องสามารถทำได้ดังนี้

2.1) แปลงโครงสร้างเอกสาร SchemaPath (SP) ให้อยู่ในรูปแบบ XML Schema ทั่วไป ด้วยเอกสาร XSL T´ ได้เป็นเอกสาร XML Schema S´

2.2) นำเอกสาร XSL MT ไปตรวจสอบกฎในเอกสาร SchemaPath (SP) และสร้างเป็นแม่แบบเก็บไว้ในเอกสาร XSL T″ สำหรับแปลงโครงสร้างเอกสาร XML

2.3) แปลงโครงสร้างเอกสาร XML X ตามแม่แบบที่กำหนดไว้ในเอกสาร XSL T″ ได้เป็นเอกสาร XML X´

2.4) นำเอกสาร XML X´ ไปตรวจสอบคุณสมบัติ Valid ตามนิยามโครงสร้างที่กำหนดไว้ในเอกสาร XML Schema S´ ด้วย XML Schema Validator ที่มีอยู่ทั่วไป

ตัวอย่างเอกสาร SchemaPath และเอกสาร XML เมื่อถูกแปลงโครงสร้างเรียบร้อยแล้ว พร้อมที่จะนำไปตรวจสอบคุณสมบัติ Valid กับ XML Schema Validator ที่มีอยู่ทั่วไป แสดงดังภาพประกอบ 2.30 และภาพประกอบ 2.31 ตามลำดับ

<table>
<tr>
<td>

```
<?xml version="1.0"?>
<xsd:schema
xmlns:xsd="http://www.cs.unibo.it/SchemaPath/1.0">
<xsd:element name="doc">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="invoiceLine"
        type="invoiceLineType"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="invoiceLineType">
  <xsd:sequence>
    <xsd:element name="unit" type="unitType"/>
    <xsd:element name="quantity">
        <xsd:alt cond="../unit='items'"
          type="xsd:integer"/>
        <xsd:alt cond="../unit='meters'"
          type="xsd:decimal"/>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="unitType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="items"/>
    <xsd:enumeration value="meters"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

เอกสาร SchemaPath (SP)

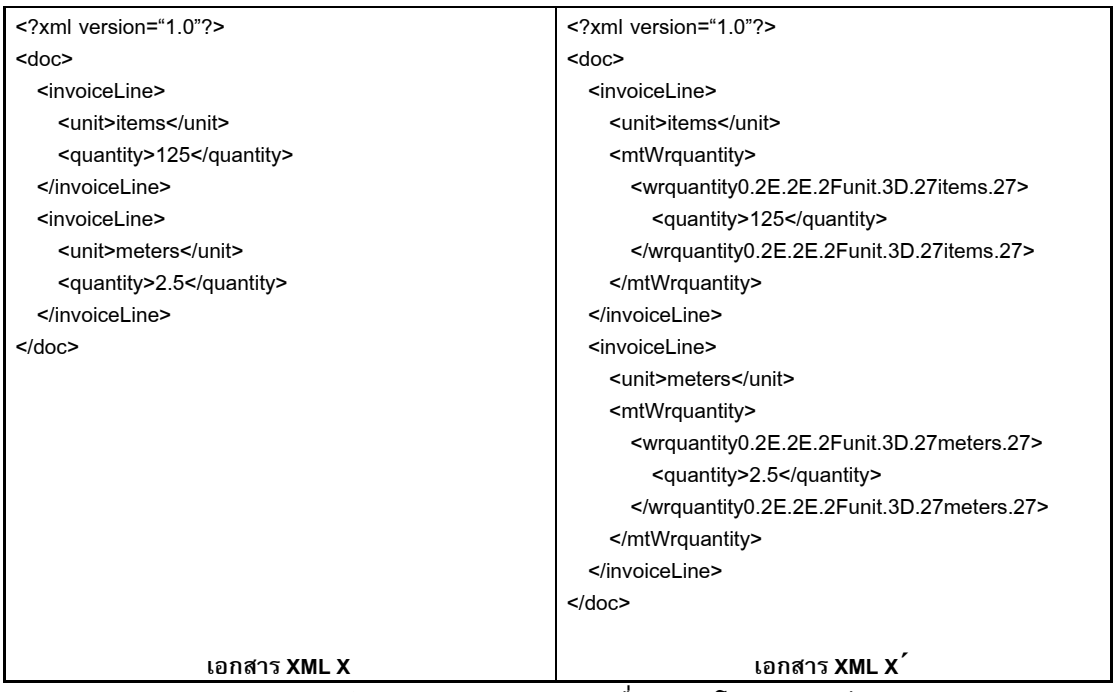
</td>
<td>

```
<?xml version="1.0"?>
<xsd:schema
xmlns:xsd="http://www.cs.unibo.it/SchemaPath/1.0">
<xsd:element name="doc">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="invoiceLine"
        type="invoiceLineType"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="invoiceLineType">
  <xsd:sequence>
    <xsd:element name="unit" type="unitType"/>
    <xsd:element name="mtWrquantity">
      <xsd:complexType>
        <xsd:choice>
          <xsd:element name="
            wrquantity0.2E.2E.2Funit.3D.27items.27">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="quantity"
                type="xsd:integer"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="
            wrquantity0.2E.2E.2Funit.3D.27meters.27">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="quantity"
                type="xsd:decimal"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </sequence>
</xsd:complexType>
<xsd:simpleType name="unitType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="items"/>
    <xsd:enumeration value="meters"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

เอกสาร XML Schema S´

</td>
</tr>
</table>

ภาพประกอบ 2.30 ตัวอย่างเอกสาร SchemaPath เมื่อแปลงโครงสร้างเรียบร้อยแล้ว



ภาพประกอบ 2.31 ตัวอย่างเอกสาร XML เมื่อแปลงโครงสร้างเรียบร้อยแล้ว

2.3.4 การนิยามภาษาโปรแกรม

ภาษาโปรแกรมต้องมีการนิยามภาษาโดยระบุในรูปของวากยสัมพันธ์ (Syntax) และความหมาย (Semantics) โดยผู้ที่นำนิยามของภาษาไปใช้ ได้แก่ ผู้ออกแบบภาษา ผู้พัฒนาภาษา และโปรแกรมเมอร์

การอธิบายวากยสัมพันธ์ของภาษาสามารถทำได้ในรูปของไวยากรณ์ (Grammar) โดยไวยากรณ์ คือ ภาษาที่ใช้ในการอธิบายภาษา (Language-Description-Language) หรือเรียกว่า Meta Language ซึ่งใช้กำหนดสตริงที่สามารถประกอบกันเป็นโปรแกรมที่ถูกต้องได้ (ทัศนวรรณ ศูนย์กลาง, 2552)

1) BNF (Backus-Naur Form)

BNF เป็น production ของไวยากรณ์ที่อยู่ในรูป A → ω หรือกฎที่ใช้ในการเขียน เพื่อนำมาใช้ในการกำหนดวากยสัมพันธ์ของภาษาโปรแกรม โดย A หรือด้านซ้ายมือ (Left-Hand Side: LHS) เป็น nonterminal symbol N หมายถึง กลุ่มของ Identifier, Integer, Expression, Statement และ Program โดยมี start symbol S ที่หมายถึงโครงสร้างหลักของภาษาและใช้ในการกำหนด production แรก และส่วนที่สอง คือ ω หรือด้านขวามือ (Right-Hand Side: RHS) ประกอบด้วย terminal symbol T คือ ตัวอักษรพื้นฐานที่ประกอบกันเป็นโปรแกรม และ nonterminal symbol N ตัวอย่างการเขียน BNF ในการกำหนดวากยสัมพันธ์ของ binaryDigit สามารถกำหนดได้ดังนี้

binaryDigit → 0 | 1

จากตัวอย่างเป็นการกำหนดว่า binaryDigit มีค่าเป็น 0 หรือ 1 เท่านั้น nonterminal symbol คือ สัญลักษณ์ทุกตัวที่ปรากฏทางด้านซ้ายมือของ production ซึ่งในที่นี้มีเพียงตัวเดียว คือ binaryDigit ส่วน termainal symbol คือ สัญลักษณ์อื่น ๆ ที่เหลือที่ปรากฏใน production จากตัวอย่างคือ 0 และ 1 โดยมีเครื่องหมาย | แทนความหมาย “หรือ” เรียกสัญลักษณ์ → และ | ว่าเป็น metasymbol คือ สัญลักษณ์ที่เป็นส่วนหนึ่งของ Meta Language แต่ไม่ได้เป็นส่วนหนึ่งของภาษาที่ถูกกำหนด ด้านขวามือของ BNF production อาจเป็นลำดับใด ๆ ของ nonterminal symbol และ terminal symbol ตัวอย่างเช่น การกำหนดไวยากรณ์ที่ใช้กับ Integer โดยกำหนดว่าเป็นลำดับของ Digit

Integer	→ Digit | Integer Digit
Digit	→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

ภาพประกอบ 2.32 ไวยากรณ์ของ Integer

จากตัวอย่างในภาพประกอบ 2.32 จะเห็นว่า production ที่สองเป็นการกำหนดเลขฐาน 10 สามารถมีค่าได้ตั้งแต่ 0 ถึง 9 และ production แรก เป็นการกำหนด Integer ว่าอาจเป็น Digit เพียงตัวเดียว หรือเป็น Integer ตามด้วย Digit ก็ได้ จะเห็นว่าการกำหนดแบบนี้เป็นลักษณะที่วนซ้ำโดยเรียกตัวเอง (Recursion) นั่นคือ Integer จะประกอบด้วยตัวเลข Digit อย่างน้อย 1 ตัวหรือมากกว่าก็ได้

2) Derivation

การพิจารณาว่าสตริงของสัญลักษณ์หนึ่งจัดอยู่ในกลุ่มไวยากรณ์ใด ทำได้โดยการ derivation ซึ่งเป็นการตรวจสอบว่าสตริงนั้นได้มาจากการใช้กฎหรือ production ของกลุ่มไวยากรณ์นั้น ๆ ตัวอย่างเช่น พิจารณาว่า 352 เป็น Integer หรือไม่ สามารถทำ derivation ได้โดยกฎของไวยากรณ์ที่ใช้กับ Integer จากภาพประกอบ 2.32 ได้ดังภาพประกอบ 2.33

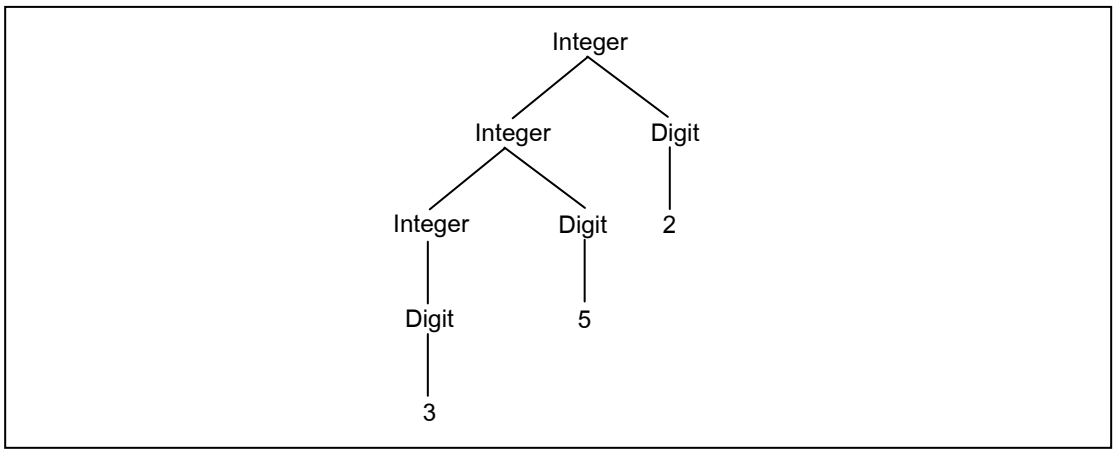
Leftmost Derivation:	**Rightmost Derivation:**
Integer => Integer Digit => Integer Digit Digit => Digit Digit Digit => 3 Digit Digit => 3 5 Digit => 3 5 2	Integer => Integer Digit => Integer 2 => Integer Digit 2 => Integer 5 2 => digit 5 2 => 3 5 2

ภาพประกอบ 2.33 Derivation ตามไวยากรณ์ของ Integer

จากภาพประกอบ 2.33 สามารถพิสูจน์ได้ว่า 352 เป็น Integer โดย Leftmost Derivation คือ การทำ derivation จากซ้ายไปขวา และ Rightmost Derivation คือ การทำ derivation จากขวาไปซ้าย

3) Parse Tree

การพิจารณาว่าสตริงของสัญลักษณ์หนึ่งจัดอยู่ในกลุ่มของภาษาที่กำหนดไว้ในรูปของ BNF หรือไม่ นอกจากทำ derivation แล้วยังสามารถเขียนให้อยู่ในรูปของ Parse Tree ได้อีกด้วย ตัวอย่างแสดงดังภาพประกอบ 2.34



ภาพประกอบ 2.34 Parse Tree ตามไวยากรณ์ของ Integer

4) Extended BNF (EBNF)

การเขียนไวยากรณ์ในรูปแบบ BNF สามารถเขียนให้อยู่ในรูปที่สั้นกะทัดรัด ด้วย Extended BNF หรือ EBNF โดยมี metasymbol เพิ่มขึ้น คือ

{ } ใช้บ่งบอกการปรากฏของสัญลักษณ์ ที่อยู่ภายในเครื่องหมายว่าอาจจะไม่มีหรือมีมากกว่าหนึ่งก็ได้

( ) ใช้บ่งบอกว่าต้องเลือกตัวใดตัวหนึ่ง จากตัวเลือกทั้งหมดที่อยู่ภายในเครื่องหมายนี้

[ ] ใช้บ่งบอกถึงลำดับของสัญลักษณ์ที่เป็นทางเลือก อาจจะเขียนหรือไม่ก็ได้ เช่น IFStatement → if (Expr) Statement [else Statement]

พิจารณาตัวอย่างของการเขียน BNF ดังแสดงในภาพประกอบ 2.35 สามารถเขียนกฎข้อแรกในลักษณะ EBNF ได้ดังภาพประกอบ 2.36

```
Expr  → Expr + Term | Expr – Term | Term
Term  → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Expr
```

ภาพประกอบ 2.35 ตัวอย่างการเขียน BNF

```
Expr  → Term { (+ | - ) Term }
```

ภาพประกอบ 2.36 ตัวอย่างการเขียน EBNF จาก BNF

การใช้ EBNF เพื่ออธิบายไวยากรณ์ที่เป็นทางการ (Formal Grammar) ของภาษา XML (Bray *et al*., 2008) แต่ละกฎของไวยากรณ์จะถูกเขียนอยู่ในรูป

symbol ::= expression

โดย symbol หมายถึง nonterminal symbol และ expression คือ การกำหนดนิพจน์สำหรับตรวจสอบสตริง ตัวอย่างแสดงดังตารางที่ 2.9 ซึ่ง expression จะประกอบด้วย nonterminal และ terminal symbol ยกตัวอย่างเช่น

```
Eq  ::=  S? '=' S?
S   ::=  (#x20 | #x9 | #xD | #xA)+
```

ตารางที่ 2.9 ตัวอย่างการกำหนดนิพจน์สำหรับตรวจสอบสตริง

รูปแบบ	คำอธิบาย
#xN	เลขจำนวนเต็มฐานสิบหกที่ใช้แทนอักขระตามมาตรฐาน ISO/IEC 10646 ซึ่งจะถูกแทนค่าลงใน N เช่น #x20 แทน ช่องว่าง เป็นต้น
[a-zA-Z], [#xN-#xN]	อักขระใดอักขระหนึ่งที่อยู่ในช่วงที่กำหนด
[abc], [#xN#xN#xN]	อักขระใดอักขระหนึ่งจากอักขระที่กำหนด
[^a-z], [^#xN-#xN]	อักขระใด ๆ ที่ไม่อยู่ในช่วงที่กำหนด
[^abc], [^#xN#xN#xN]	อักขระใด ๆ ที่ไม่ใช่อักขระที่กำหนด
“string”	ข้อความที่ตรงกับข้อความภายในเครื่องหมาย double quotes
‘string’	ข้อความที่ตรงกับข้อความภายในเครื่องหมาย single quotes
A?	ปรากฏ A หรือไม่ก็ได้ ถ้าปรากฏมีได้เพียง 1 ครั้ง
A B	ปรากฏ A แล้วตามด้วย B
A | B	ปรากฏ A หรือ B
A - B	ปรากฏ A แต่ต้องไม่ปรากฏ B
A+	ปรากฏ A อย่างน้อย 1 ครั้ง
A*	ปรากฏ A หรือไม่ก็ได้ ถ้าปรากฏมีได้มากกว่า 1 ครั้ง

2.4 เทคโนโลยี RSS (Really Simple Syndication)

RSS เป็นรูปแบบหนึ่งในการรวบรวมและเผยแพร่เนื้อหาของเว็บไซต์ ถูกพัฒนาขึ้นตามแนวคิดของเทคโนโลยี Push เพื่อรวบรวมข้อมูลข่าวสารจากแหล่งผู้ให้บริการข้อมูลข่าวสารต่าง ๆ และแจ้งให้ผู้รับบริการทราบโดยไม่ต้องมีการร้องขอเมื่อมีการอัพเดทข้อมูลข่าวสาร อีกทั้งยังช่วยลดปัญหาในเรื่องการละเมิดลิขสิทธิ์ และทำให้ผู้พัฒนาเว็บไซต์ไม่ต้องเสียเวลาปรับปรุงเว็บเพจเมื่อผู้ให้บริการมีการปรับปรุงข้อมูลข่าวสาร อย่างไรก็ตาม RSS ยังไม่ได้เป็นเทคโนโลยี Push แบบเต็มรูปแบบเนื่องจากจะทราบว่ามีข้อมูลใหม่หรือไม่นั้น ผู้รับบริการต้องตรวจสอบข้อมูลข่าวสารเป็นช่วงเวลา (Finkelstein, 2005)

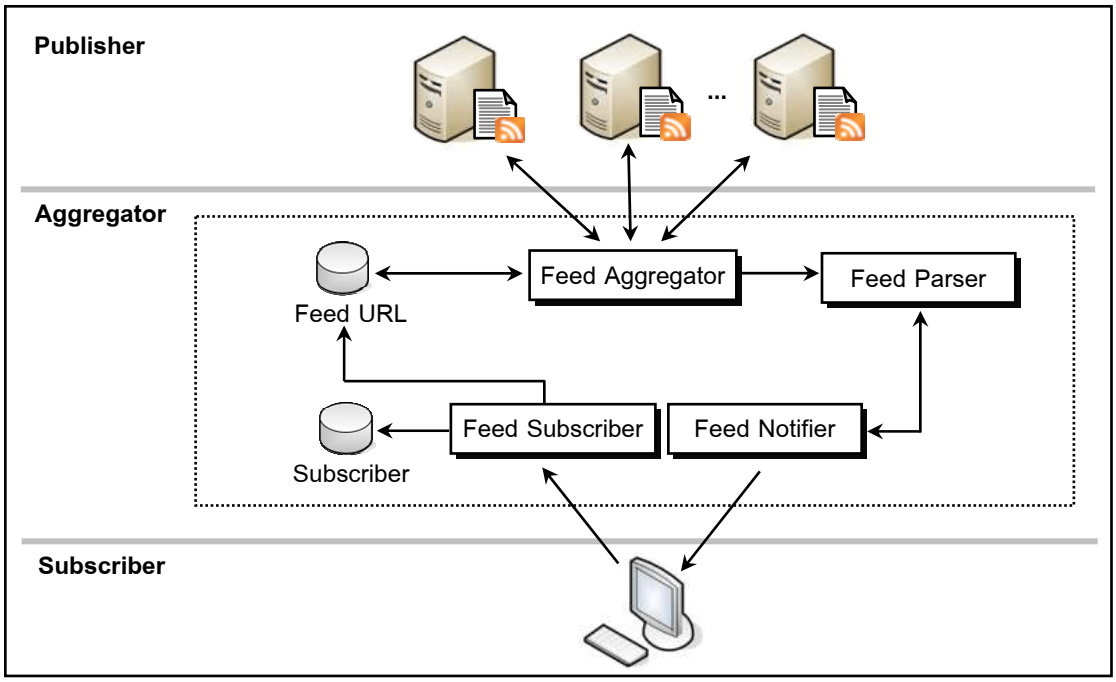
2.4.1 โครงสร้างการทำงานของ RSS

โครงสร้างการทำงานของ RSS ประกอบด้วย 3 ส่วน คือ

1) ผู้เผยแพร่ข้อมูลข่าวสาร (Publisher) คือ เว็บไซต์ที่ให้บริการข้อมูลข่าวสารในรูปแบบเอกสาร RSS

2) ผู้รวบรวมข้อมูลข่าวสาร (Aggregator) คือ แอพพลิเคชันที่ทำหน้าที่เป็นตัวแทนรวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่าง ๆ

3) ผู้รับบริการข้อมูลข่าวสาร (Subscriber) คือ ผู้รับบริการข้อมูลข่าวสารจากเว็บไซต์ต่าง ๆ



ภาพประกอบ 2.37 โครงสร้างการทำงานของ RSS

จากภาพประกอบ 2.37 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1) ผู้รับบริการลงทะเบียนรับข้อมูลข่าวสารไปยัง Feed Subscriber จากนั้นข้อมูลของผู้รับบริการจะถูกบันทึกลงฐานข้อมูล Subscriber

2) ผู้รับบริการระบุ URL ที่อยู่ของเอกสาร RSS ที่ต้องการเก็บไว้ในฐานข้อมูล Feed URL

3) เมื่อผู้รับบริการเปิดอ่านข้อมูลข่าวสาร Feed Aggregator จะทำการรวบรวมเอกสาร RSS จากเว็บไซต์ต่าง ๆ ตาม URL ที่ผู้ใช้ได้ระบุไว้ และส่งไปยัง Feed Parser เพื่อวิเคราะห์โครงสร้าง (Parse) ข้อมูล RSS และจัดรูปแบบผลลัพธ์

4) Feed Notifier แสดงผลลัพธ์ที่ได้จากขั้นตอนที่ 3) ไปยังผู้รับบริการ

2.4.2 ขั้นตอนพื้นฐานสำหรับสร้างและรับเอกสาร RSS

1) ผู้เผยแพร่ข่าวสารระบุเนื้อหาของเว็บไซต์ ที่ต้องการเผยแพร่ให้บุคคลทั่วไปทราบ โดยมากจะเป็นเนื้อหาที่มีการเปลี่ยนแปลงบ่อย ๆ

2) สร้างเอกสาร RSS ที่ระบุลิงค์เชื่อมโยงไปยังเนื้อหาของข้อมูลข่าวสารทั้งหมด

```
<?xml version = “1.0” ?>
<rss version = “2.0” >
    <channel>
        <title>…</title>
        <link>…</link>
        <description>…</description>
        <item>
            <title>…</title>
            <link>…</link>
            <description>…</description>
            <pubDate>…</pubDate>
            ...
        </item>
        …
    </channel>
</rss
```

ภาพประกอบ 2.38 รูปแบบเอกสาร RSS 2.0

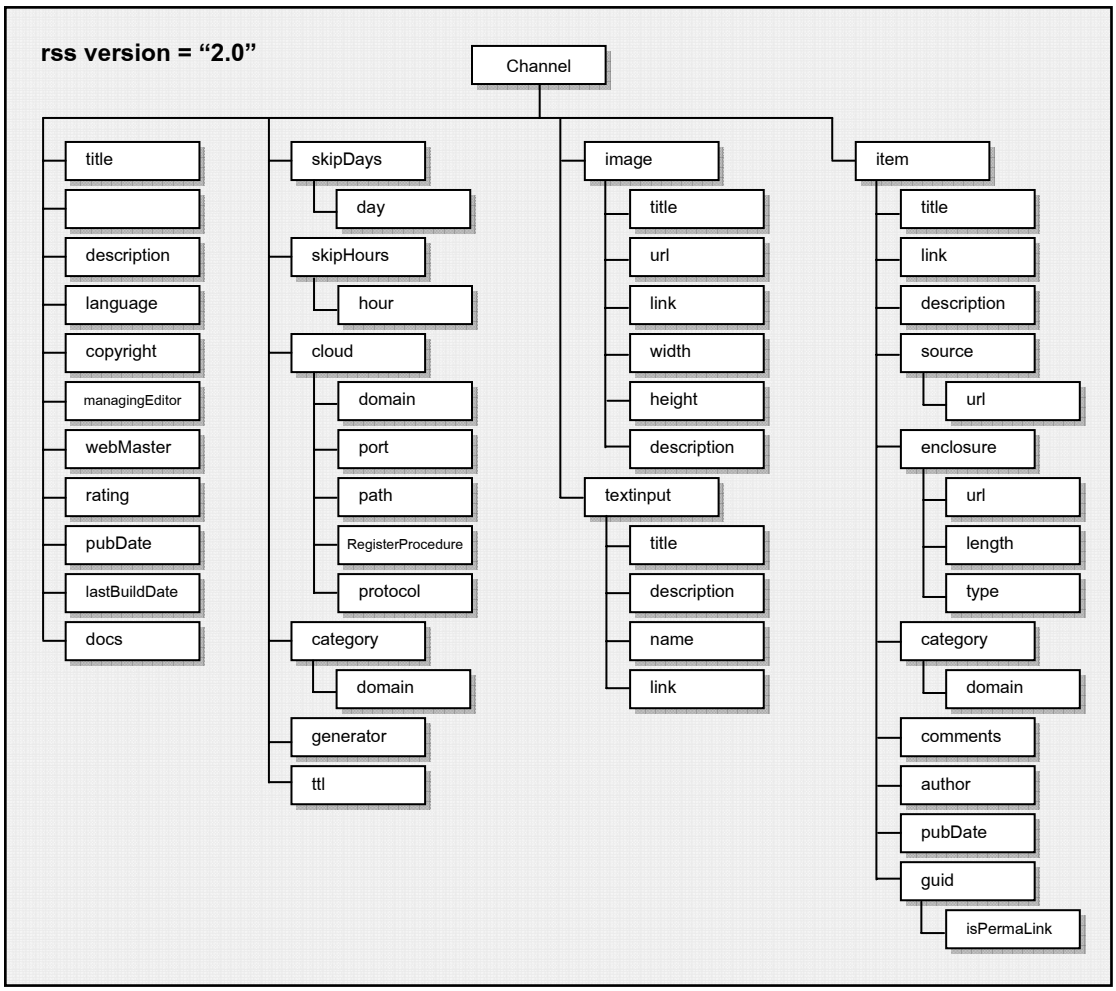
จากภาพประกอบ 2.38 แสดงรูปแบบเอกสาร RSS 2.0 โดยมีแท็ก **<rss>** บอกจุดเริ่มต้น ตามด้วยแท็ก **<channel>** เก็บข้อมูลต่าง ๆ ของ RSS ไว้ แสดงดังตารางที่ 2.10 และมีแท็ก **<item>** เป็นส่วนสำคัญทำหน้าที่เก็บรายการข้อมูลข่าวสารต่าง ๆ ดังแสดงในตารางที่ 2.11 โครงสร้างแท็กต่าง ๆ ของเอกสาร RSS แสดงดังภาพประกอบ 2.39 และตัวอย่างเอกสาร RSS แสดงดังภาพประกอบ 2.40

ตารางที่ 2.10 แท็กย่อยภายในแท็ก <channel> ของเอกสาร RSS

แท็ก	คำอธิบาย
<title>	หัวเรื่องของ Channel
<link>	ลิงค์เชื่อมโยงไปยังข้อมูลหลัก
<description>	คำอธิบายโดยย่อของเอกสาร RSS
<language>	ภาษาของข้อมูลภายในเอกสาร RSS
<copyright>	ข้อมูลลิขสิทธิ์
<managingEditor>	ที่อยู่อีเมล์ไปยังบรรณาธิการผู้ดูแลเนื้อหาของเอกสาร RSS
<webMaster>	ที่อยู่ของผู้พัฒนาเว็บไซต์ที่เผยแพร่ข้อมูลข่าวสาร
<rating>	ระบุการจัดลำดับความนิยมของข้อมูลในเอกสาร RSS
<pubDate>	วันเวลาที่ทำการเผยแพร่ข้อมูล
<lastBuildDate>	วันเวลาล่าสุดที่ทำการปรับปรุงข้อมูลภายในเอกสาร RSS
<docs>	ระบุ URL ของเอกสารที่ใช้กำหนดรูปแบบเอกสาร RSS
<skipDays>	ระบุวันเพื่อ RSS Reader จะได้ไม่ต้องตรวจสอบการอัพเดทเนื้อหา
<skipHours>	ระบุเวลาที่ RSS Reader ไม่ต้องตรวจสอบการอัพเดทเนื้อหา
<cloud>	ขั้นตอนการลงทะเบียนเมื่อผู้เผยแพร่มีการอัพเดทข้อมูลใหม่
<category>	กำหนดหมวดหมู่ให้กับข้อมูลภายในเอกสาร RSS
<generator>	ระบุโปรแกรมที่ใช้แสดงเอกสาร RSS
<ttl>	ช่วงระยะเวลาที่ข้อมูลยังคงใช้ได้ ก่อนจะมีการเปลี่ยนแปลงใหม่
<image>	รูปภาพอธิบายเอกสาร RSS
<textinput>	แถบข้อความเข้าที่แสดงในเอกสาร RSS
<item>	รายการข้อมูล

ตารางที่ 2.11 แท็กย่อยภายในแท็ก <item> ของเอกสาร RSS

แท็ก	คำอธิบาย
<title>	หัวเรื่องรายการข้อมูล
<link>	ลิงค์เชื่อมโยงไปยังข้อมูลหลัก
<description>	รายละเอียดข้อมูลโดยย่อ
<source>	แหล่งที่มาของรายการข้อมูล
<enclosure>	ไฟล์มัลติมีเดียที่แนบมาพร้อมกับรายการข้อมูล
<category>	ประเภทของข้อมูล
<comments>	ลิงค์ไปยังคำวิจารณ์เกี่ยวกับรายการข้อมูล
<author>	ข้อมูลผู้ประกาศ
<pubDate>	วันเวลาที่เผยแพร่ข้อมูล
<guid>	globally unique identifier ลิงค์สำหรับระบุแต่ละรายการข้อมูล



ภาพประกอบ 2.39 โครงสร้างแท็กต่าง ๆ ของเอกสาร RSS 2.0 (Hammersley, 2003)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rss version="2.0">
    <channel>
        <title>CNN.com - Technology</title>
        <link>http://edition.cnn.com/TECH/?eref=edition_technology</link>
        <description>CNN.com delivers up-to-the-minute news and information on the latest top
            stories, weather, entertainment, politics and more.</description>
        <language>en-us</language>
        <copyright>2009 Cable News Network LP, LLLP.</copyright>
        <pubDate>Fri, 17 Apr 2009 09:18:04 EDT</pubDate>
        <ttl>10</ttl>
        <image>
            <title>CNN.com - Technology</title>
            <link>http://edition.cnn.com/TECH/?eref=edition_technology</link>
            <url>http://i2.cdn.turner.com/cnn/.element/img/1.0/logo/cnn.logo.rss.gif</url>
            <width>144</width>
            <height>33</height>
            <description>CNN.com delivers up-to-the-minute news and information on the latest
                topstories, weather, entertainment, politics and more.</description>
        </image>
        <item>
            <title>Four sentenced to jail in landmark piracy case</title>
            <guid isPermaLink="false">http://edition.cnn.com/2009/TECH/04/17/
                sweden.piracy.jail/index.html?eref=edition_technology</guid>
            <link>http://rss.cnn.com/~r/rss/edition_technology/~3/zAZZ4K0eDVc/
                index.html</link>
            <description>A Swedish judge has found four men involved in a file-sharing Web
                site guilty of collaborating to violate copyright law and sentenced each one to a
                year in prison and 30 million kronor ($3.6 million) in damages.&lt;img
                src="http://feeds2.feedburner.com/~r/rss/edition_technology/
                ~4/zAZZ4K0eDVc" height="1" width="1"/&gt;</description>
            <pubDate>Fri, 17 Apr 2009 09:16:17 EDT</pubDate>
        </item>
            .
            .
            .
    </channel>
</rss>
```

ภาพประกอบ 2.40 ตัวอย่างเอกสาร RSS (CNN.com, 2009: Online)

3) ผู้รับบริการข่าวสารลงทะเบียนรับข้อมูลข่าวสาร และเพิ่ม URL เพื่อบอกที่อยู่ของเอกสาร RSS ที่ต้องการลงใน RSS Reader โดยพิจารณาจากสัญลักษณ์ที่ปรากฏอยู่ในเว็บไซต์ ดังแสดงในภาพประกอบ 2.41

ภาพประกอบ 2.41 สัญลักษณ์ RSS ที่ปรากฏในหน้าเว็บไซต์ต่าง ๆ

4) ผู้รับบริการข่าวสารอ่านข้อมูลข่าวสารภายในเอกสาร RSS ด้วย RSS Reader ซึ่งโดยทั่วไปแบ่งออกเป็น 2 ประเภท คือ

4.1) Software Reader ที่ต้องติดตั้งบนเครื่องคอมพิวเตอร์ก่อนการใช้งาน เช่น FeedReader และ RSSReader สำหรับเครื่องคอมพิวเตอร์ทั่วไป และ LiteFeeds สำหรับอุปกรณ์สื่อสารเคลื่อนที่ เป็นต้น นอกจากนี้ปัจจุบันเบราว์เซอร์ยังได้มีการพัฒนา RSS Reader โดยฝังเข้ากับตัวเบราว์เซอร์ ได้แก่ Mozilla Firefox เป็นต้น

4.2) Web-based RSS Reader ผู้รับบริการสามารถใช้งานผ่านเว็บไซต์ โดยไม่ต้องติดตั้งโปรแกรม เช่น Bloglines.com เป็นต้น

2.4.3 ข้อดีของการรับข้อมูลข่าวสารด้วย RSS (Finkelstein, 2005)

1) สามารถรับข้อมูลข่าวสารที่สนใจได้จากหลาย ๆ เว็บไซต์ด้วยการเข้าใช้งานที่ใดที่หนึ่งเท่านั้น ทำให้ลดเวลาในการเข้าถึงข้อมูลข่าวสาร

2) สามารถรับข้อมูลข่าวสารได้เมื่อต้องการ เพียงเข้าไปยัง RSS Reader ทำให้ไม่ต้องเสียเวลารอให้มีการส่งข้อมูลข่าวสารมายังผู้รับบริการ

3) สามารถรับข้อมูลข่าวสารเฉพาะเรื่องที่สนใจได้ เพราะ RSS แสดงพาดหัวข่าวและรายละเอียดย่อ ๆ ทำให้ผู้ใช้สามารถเลือกอ่านเฉพาะเรื่องที่สนใจได้

4) ผู้รับบริการสามารถใช้ RSS เพื่อรวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่าง ๆ แล้วนำมาเผยแพร่ในเว็บไซต์ที่สร้างขึ้น โดยไม่เกิดปัญหาในเรื่องการละเมิดลิขสิทธิ์แต่อย่างใด

2.4.4 ข้อดีของการเผยแพร่ข้อมูลข่าวสารด้วย RSS (Finkelstein, 2005)

1) ไม่ต้องบำรุงรักษาฐานข้อมูลผู้รับบริการข้อมูลข่าวสาร เนื่องจากผู้ใช้สามารถดึงเอกสาร RSS ที่เกี่ยวข้องกับเรื่องที่สนใจได้เอง

2) สามารถสร้างเอกสารได้ง่าย ถ้ามีข้อมูลข่าวสารใหม่ ๆ เกี่ยวกับเว็บไซต์ เพราะรูปแบบของเอกสาร RSS ประกอบด้วยหัวเรื่องและรายละเอียดเพียงสั้น ๆ เพื่อเชื่อมโยงไปยังเนื้อหาทั้งหมด

3) ผู้รับบริการมีความรู้สึกดีต่อผู้ให้บริการข้อมูลข่าวสาร เพราะสามารถเลือกข้อมูลข่าวสารที่ต้องการได้ด้วยตนเอง

4) มีผู้เข้าเยี่ยมชมเว็บไซต์มากขึ้น เพราะเอกสาร RSS มีลิงค์เชื่อมโยงเพื่อย้อนกลับไปยังเว็บไซต์ปลายทางที่มีเนื้อหาทั้งหมดอยู่

5) เว็บไซต์มีข้อมูลข่าวสารที่ทันสมัยอยู่เสมอ

2.5 ความปลอดภัย (Security)

ความปลอดภัยของข้อมูลถูกนำมาพิจารณาเป็นประเด็นหลัก สำหรับการแลกเปลี่ยนข้อมูลข่าวสารผ่านอินเทอร์เน็ต เนื่องจากอินเทอร์เน็ตเป็นเครือข่ายสาธารณะทำให้การป้องกันการบุกรุกจากผู้ที่ไม่ได้รับอนุญาตในการเข้าถึงข้อมูล โดยเฉพาะข้อมูลที่มีความสำคัญ เช่น ข้อมูลส่วนตัว เป็นเรื่องสำคัญอย่างยิ่ง เพื่อให้มั่นใจได้ว่าข้อมูลจะไม่ถูกทำลาย เปลี่ยนแปลง หรือรั่วไหลไปยังบุคคลที่ไม่เกี่ยวข้อง

2.5.1 องค์ประกอบพื้นฐานที่ทำให้ข้อมูลมีความปลอดภัย

โดยองค์ประกอบพื้นฐานที่ทำให้ข้อมูลมีความปลอดภัย มีดังนี้

1) การมีบูรณภาพ (Integrity) คือ การรับรองว่าข้อมูลที่ได้รับนั้นจะไม่ถูกเปลี่ยนแปลงหรือทำลายในระหว่างการส่งข้อมูลจากผู้ส่งไปจนถึงผู้รับ

2) การรักษาความลับ (Confidentiality) คือ การรับรองว่าจะมีการเก็บข้อมูลไว้เป็นความลับหรือปกปิดข้อมูลที่รับส่งผ่านสื่อต่าง ๆ และอนุญาตให้ผู้ที่มีสิทธิ์เท่านั้นจึงจะเข้าถึงข้อมูลนั้นได้

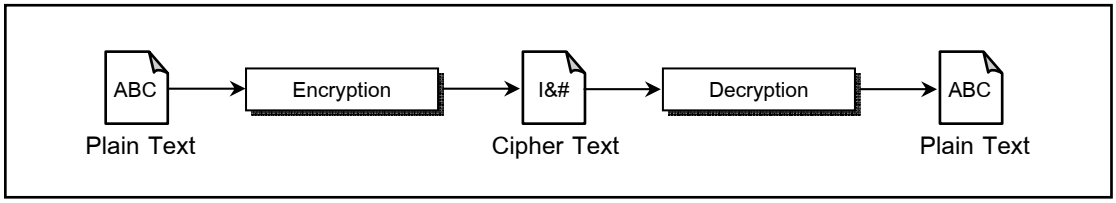
3) การระบุตัวตน (Authentication) คือ วิธีการยืนยันความถูกต้องของหลักฐาน (Identity) ที่แสดงว่าเป็นบุคคลที่กล่าวอ้างจริง เพื่อยืนยันตัวบุคคลที่สามารถเข้าถึงข้อมูลได้

4) การไม่ปฏิเสธการกระทำ (Non-repudiation) คือ วิธีที่ทำให้ผู้ส่งข้อมูลไม่สามารถปฏิเสธได้ว่าเป็นผู้ส่งข้อมูลนั้น

เทคนิคสำคัญที่ทำให้ข้อมูลข่าวสารมีความปลอดภัย คือ การประยุกต์ใช้วิทยาการเข้ารหัสลับ ซึ่งเป็นเครื่องมือในการเข้ารหัสและถอดรหัสข้อมูล

2.5.2 วิทยาการเข้ารหัสลับ (Cryptography)

วัตถุประสงค์ของวิทยาการเข้ารหัสลับ คือ การทำให้บุคคล 2 ฝ่ายสามารถติดต่อสื่อสารข้อมูลระหว่างกันได้อย่างปลอดภัย (ลัญฉกร วุฒิสิทธิกุลกิจ และคณะ, 2548) ซึ่งมีองค์ประกอบสำคัญ คือ อัลกอริทึม (Algorithm) และกุญแจ (Key) ที่ใช้ในการเข้ารหัสและถอดรหัสข้อความ โดยกระบวนการเข้ารหัสและถอดรหัสข้อความแสดงดังภาพประกอบ 2.42



ภาพประกอบ 2.42 กระบวนการเข้ารหัสและถอดรหัสข้อความ

จากภาพประกอบ 2.42 ผู้ส่งเข้ารหัส (Encryption) ข้อความต้นฉบับ (Plain Text) ด้วยอัลกอริทึมและกุญแจในการเข้ารหัสได้เป็นข้อความไซเฟอร์ (Cipher Text) เพื่อส่งไปยังฝั่งผู้รับ จากนั้นผู้รับถอดรหัส (Decryption) ข้อความไซเฟอร์ที่ได้ด้วยอัลกอริทึมและกุญแจในการถอดรหัสได้เป็นข้อความต้นฉบับอีกครั้ง

2.5.3 ประเภทของการเข้ารหัสลับ

วิทยาการเข้ารหัสลับที่เป็นที่รู้จักกันในปัจจุบันแบ่งออกเป็น 2 ประเภท คือ

1) การเข้ารหัสลับแบบกุญแจลับ (Secret Key Cryptography) หรือการเข้ารหัสลับแบบกุญแจสมมาตร (Symmetric Key Cryptography) เป็นวิธีการที่ทั้งการเข้ารหัส (Encryption) และการถอดรหัส (Decryption) จะใช้กุญแจหรือรหัสลับเดียวกัน ผู้ส่งและผู้รับต้องตกลงกันว่าจะใช้กุญแจเดียวกันก่อนเริ่มส่งข้อความ ซึ่งการเข้ารหัสแบบนี้ทำให้เกิดปัญหาในเรื่องของการแลกเปลี่ยนกุญแจและจำนวนกุญแจที่ใช้ในการเข้ารหัส อย่างไรก็ตาม วิทยาการเข้ารหัสแบบกุญแจสมมาตรก็ยังถูกนำมาใช้อย่างกว้างขวาง ตัวอย่าง

อัลกอริทึมของวิทยาการเข้ารหัสแบบกุญแจสมมาตร ได้แก่ DES (Data Encryption Standard), RC5, Blowfish และ Advanced Encryption Standard (AES) เป็นต้น

2) การเข้ารหัสลับแบบกุญแจสาธารณะ (Public Key Cryptography) หรือการเข้ารหัสลับแบบกุญแจอสมมาตร (Asymmetic Key Cryptography) วิธีนี้จะประกอบด้วยกุญแจ 2 ตัว คือกุญแจสาธารณะ (Public Key) และกุญแจส่วนตัว (Private Key) โดยกุญแจสาธารณะจะประกาศให้บุคคลทั่วไปทราบ ขณะที่กุญแจส่วนตัวจะถูกเก็บไว้เป็นความลับ ซึ่งการเข้ารหัสแบบกุญแจอสมมาตรนี้ ช่วยแก้ปัญหาในเรื่องของการแลกเปลี่ยนกุญแจและจำนวนกุญแจที่ใช้ได้ ตัวอย่างอัลกอริทึมของการเข้ารหัสวิธีนี้ ได้แก่ RSA และอัลกอริทึมลายมือชื่อดิจิตอล DSA (Digital Signature Algorithm) เป็นต้น

วิทยาการเข้ารหัสลับแบบกุญแจสมมาตรและแบบกุญแจอสมมาตรมีลักษณะเด่นที่แตกต่างกัน โดยวิทยาการเข้ารหัสลับแบบกุญแจอสมมาตรสามารถแก้ปัญหาเรื่องการแลกเปลี่ยนกุญแจได้ แต่การทำงานจะช้ากว่าและข้อความไซเฟอร์มีขนาดใหญ่กว่าการเข้ารหัสด้วยวิธีแบบกุญแจสมมาตร คุณลักษณะของวิทยาการเข้ารหัสลับแบบกุญแจสมมาตรและแบบกุญแจอสมมาตร แสดงดังตารางที่ 2.12 (ลัดดา ปรีชาวีรกุล, 2551)

ตารางที่ 2.12 คุณลักษณะของวิทยาการเข้ารหัสลับแบบกุญแจสมมาตรและอสมมาตร

คุณลักษณะ	แบบกุญแจสมมาตร	แบบกุญแจอสมมาตร
กุญแจที่ใช้เข้าและถอดรหัส	ใช้กุญแจเดียวกัน	ใช้กุญแจตัวหนึ่งเข้ารหัสและอีกตัวหนึ่งถอดรหัส
เวลาที่ใช้ในการเข้าและถอดรหัส	ใช้เวลาน้อย	ใช้เวลามากกว่า
ขนาดข้อความไซเฟอร์	โดยทั่วไปขนาดเท่ากับหรือน้อยกว่าข้อความต้นฉบับ	ขนาดใหญ่กว่าข้อความต้นฉบับ
การแลกเปลี่ยนกุญแจ	เป็นปัญหาใหญ่	ไม่มีปัญหา
จำนวนกุญแจ	เท่ากับจำนวนผู้ติดต่อ	ไม่เกินจำนวนผู้ติดต่อ
การใช้งาน	ทำให้ข้อมูลเป็นความลับ แต่ไม่สามารถใช้สำหรับลายมือดิจิตอลเพื่อให้เกิดการมีบูรณภาพและการไม่ปฏิเสธการกระทำได้	ทำให้ข้อมูลเป็นความลับ และสามารถใช้ สำหรับการมีบูรณภาพและการไม่ปฏิเสธการกระทำได้

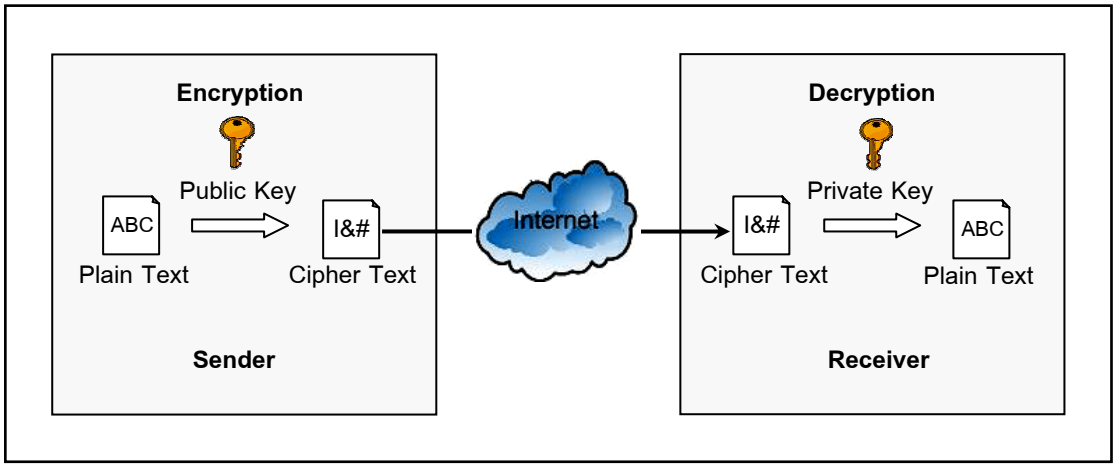
2.5.4 ตัวอย่างอัลกอริทึมเข้ารหัสลับ

1) อัลกอริทึม Blowfish

Blowfish เป็นอัลกอริทึมเข้ารหัสแบบกุญแจสมมาตรที่มีความรวดเร็วในการทำงานมีขนาดเล็กกะทัดรัด และใช้การเข้ารหัสแบบบล็อก (Block Cipher) โดยข้อมูลในการเข้ารหัสจะถูกแบ่งออกเป็นบล็อก ๆ ละ 64 บิต ขนาดของกุญแจสามารถเปลี่ยนแปลงได้ตั้งแต่ขนาดไม่มากนักไปจนถึงขนาด 448 บิต ทำให้มีความยืดหยุ่นในการเลือกใช้กุญแจ อีกทั้งอัลกอริทึมยังได้รับการออกแบบมาให้ทำงานอย่างเหมาะสมกับหน่วยประมวลผลขนาด 32 หรือ 64 บิต

2) อัลกอริทึม RSA

RSA เป็นอัลกอริทึมเข้ารหัสลับแบบกุญแจอสมมาตร โดยการเข้ารหัสจะใช้กุญแจหนึ่งสำหรับเข้ารหัสและใช้อีกกุญแจหนึ่งสำหรับถอดรหัส ซึ่งกุญแจทั้งสองมีความสัมพันธ์กันในทางคณิตศาสตร์ โดย RSA สามารถใช้ประโยชน์ได้หลายด้าน เช่น การเข้ารหัสข้อมูล และการแจกจ่ายกุญแจ ซึ่งมีขั้นตอนการทำงานดังแสดงในภาพประกอบ 2.43 (จตุชัย แพงจันทร์, 2550)



ภาพประกอบ 2.43 การเข้ารหัสลับด้วยอัลกอริทึม RSA

โดยขั้นตอนการทำงานของอัลกอริทึม RSA (Kahate, 2003) มีดังนี้

1) สุ่มเลือกจำนวนเฉพาะขนาดใหญ่ 2 จำนวน กำหนดให้เป็น P และ Q จากนั้นคำนวณค่า N เพื่อใช้ในการเข้ารหัสและถอดรหัสข้อความด้วยสมการที่ (2.1)

$$N = P \times Q \tag{2.1}$$

2) เลือกกุญแจสาธารณะ (E) เพื่อใช้เข้ารหัสข้อความ โดยต้องไม่เป็นตัวประกอบของผลคูณของ (P-1) กับ (Q-1)

3) เลือกกุญแจส่วนตัว (D) เพื่อใช้ถอดรหัสข้อความ ซึ่งทำให้สมการที่ (2.2) เป็นจริง

$$(D \times E) \bmod (P-1) \times (Q-1) = 1 \tag{2.2}$$

4) เข้ารหัสข้อความต้นฉบับ (M) ได้เป็นข้อความไซเฟอร์ (C) ด้วยสมการที่ (2.3)

$$C = M^E \bmod N \tag{2.3}$$

5) ส่งข้อความไซเฟอร์ไปยังผู้รับ และถอดรหัสข้อความไซเฟอร์ได้เป็นข้อความต้นฉบับด้วยสมการที่ (2.4)

$$M = C^D \bmod N \tag{2.4}$$

2.6 อุปกรณ์สื่อสารเคลื่อนที่ (Mobile Device)

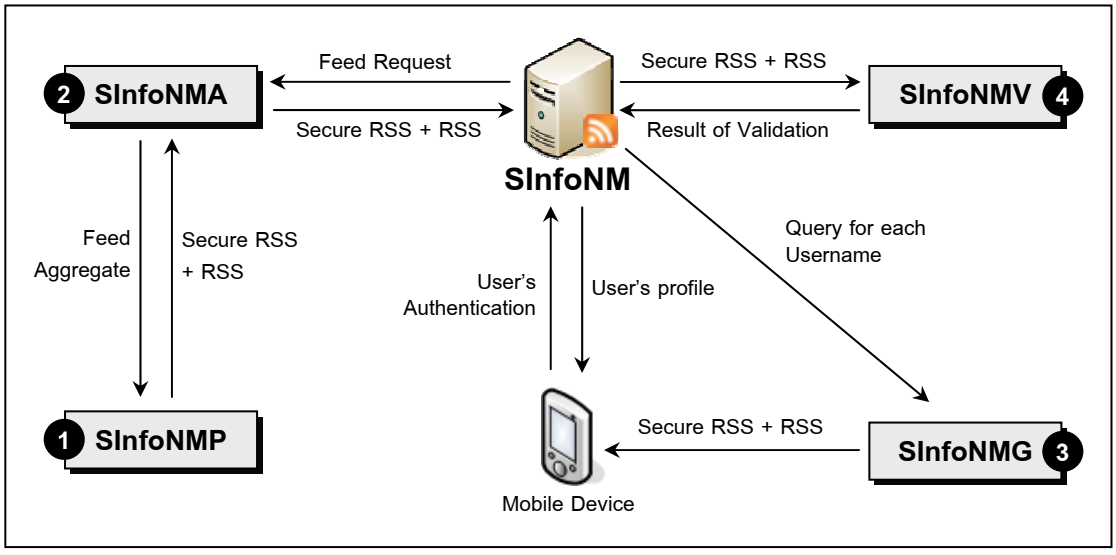
Mobile Device หมายถึง อุปกรณ์ขนาดเล็กที่สามารถพกพาได้ เช่น โทรศัพท์มือถือ พีดีเอ เป็นต้น และด้วยความก้าวหน้าของเทคโนโลยีทางด้าน Mobile ทำให้ปัจจุบันโทรศัพท์เคลื่อนที่เป็นอุปกรณ์ที่จำเป็นในชีวิตประจำวัน เพราะช่วยอำนวยความสะดวกในการติดต่อสื่อสาร นอกจากนี้ยังมีการใช้งานพีดีเอเพื่อรับส่งอีเมล์ จัดตารางนัดหมายหรือเก็บข้อมูลส่วนตัว และปัจจุบันเครื่องคอมพิวเตอร์โน้ตบุ๊กมีขนาดเล็กลงทำให้สามารถพกพาและใช้งานได้สะดวก อีกทั้งยังสามารถเชื่อมต่อกับเครือข่ายไร้สายเพื่อเข้าใช้งานอินเทอร์เน็ตได้อีกด้วย ทำให้ผู้ใช้งานอุปกรณ์สื่อสารเคลื่อนที่สามารถติดตามข้อมูลข่าวสารต่าง ๆ ได้สะดวกยิ่งขึ้น

บทที่ 3

กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

วิทยานิพนธ์นี้ได้ออกแบบกลไกการทำงาน เพื่อทำให้ RSS สามารถแจ้งสารสนเทศที่ต้องการความปลอดภัยไปยังผู้ที่เกี่ยวข้องได้ ซึ่งประยุกต์ใช้เทคโนโลยี RSS ร่วมกับวิทยาการเข้ารหัสลับ โดยเสนอกลไกการทำงานบนพื้นฐานของอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP เพื่อความสะดวกในการรับสารสนเทศที่ทันสมัยของผู้รับบริการ

แบบจำลองการทำงานโดยรวมของกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP (A Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices: SInfoNM) ดังแสดงในภาพประกอบ 3.1 แบ่งการทำงานออกเป็น 4 ส่วน คือ 1) กลไกการเผยแพร่สารสนเทศที่ต้องการความปลอดภัย (SInfoNMP) 2) กลไกการรวบรวมเอกสาร RSS (SInfoNMA) 3) กลไกการสร้างเอกสาร RSS เฉพาะผู้ใช้ (SInfoNMG) และ 4) กลไกการตรวจสอบความถูกต้องของเอกสาร RSS (SInfoNMV) โดยสามารถอธิบายกลไกการทำงานแต่ละส่วนได้ดังนี้



ภาพประกอบ 3.1 แบบจำลองการทำงานโดยรวมของ SInfoNM

3.1 กลไกการเผยแพร่สารสนเทศที่ต้องการความปลอดภัย (SInfoNM Publisher: SInfoNMP)

การเผยแพร่สารสารสนเทศที่ต้องการความปลอดภัยได้ประยุกต์ใช้วิทยาการเข้ารหัสลับ เพื่อทำให้สารสนเทศมีความปลอดภัยก่อนถูกเผยแพร่ผ่านเทคโนโลยี RSS ประกอบด้วยส่วนต่าง ๆ ดังนี้

3.1.1 แจกจ่ายกุญแจสำหรับเข้ารหัส

ระบบแจกจ่ายกุญแจสาธารณะ (Public Key) สำหรับเข้ารหัสกุญแจเข้ารหัสข้อมูลข่าวสารส่วนบุคคลให้กับหน่วยงานที่ต้องการเผยแพร่สารสนเทศที่ต้องการความปลอดภัย

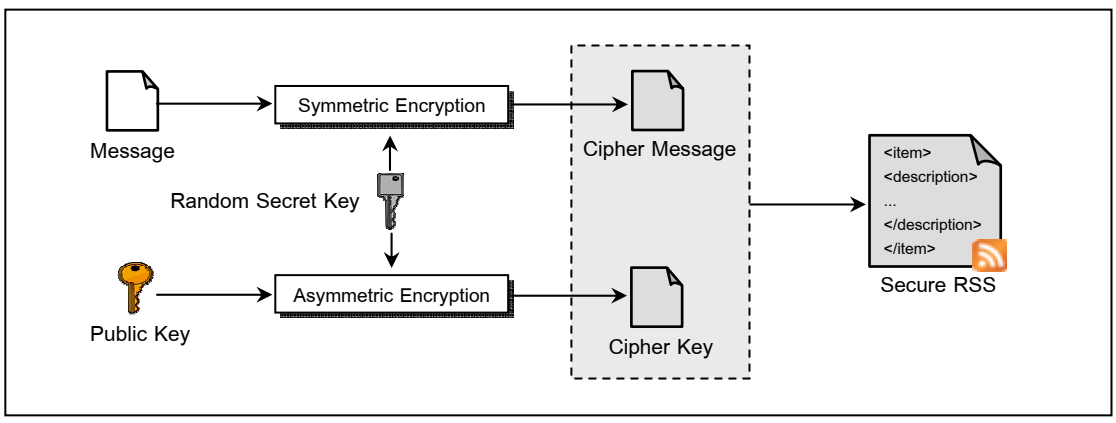
3.1.2 เข้ารหัสข้อมูลข่าวสารที่ต้องการความปลอดภัย

การเข้ารหัสข้อมูลข่าวสารที่ต้องการเผยแพร่ผ่านเอกสาร RSS ดังแสดงในภาพประกอบ 3.2 จะเข้ารหัสเฉพาะข้อมูลข่าวสารส่วนบุคคล โดยมีกระบวนการทำงานดังนี้

1) เข้ารหัสข้อความต้นฉบับ (Message) แต่ละรายการ (แท็ก <Item>) ภายในเอกสาร RSS โดยใช้ขั้นตอนวิธีเข้ารหัสลับแบบกุญแจสมมาตร ด้วยกุญแจที่เกิดจากการสุ่ม (Random Secret Key) ได้เป็นข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล (Cipher Message)

2) เข้ารหัสกุญแจที่ใช้ถอดรหัสข้อความต้นฉบับ ด้วยกุญแจสาธารณะที่ได้รับจากระบบ ได้เป็นข้อความไซเฟอร์ของกุญแจถอดรหัส (Cipher Key)

3) ข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคลและข้อความไซเฟอร์ของกุญแจถอดรหัส จะถูกนำไปแทนที่ข้อความต้นฉบับภายในแท็ก <description> ของแต่ละรายการ



ภาพประกอบ 3.2 การเข้ารหัสข้อมูลข่าวสารส่วนบุคคล

3.1.3 สร้างเอกสาร RSS

เอกสาร RSS ที่สร้างขึ้นจะประกอบด้วยข้อมูลข่าวสารทั่วไป และข้อมูลข่าวสารส่วนบุคคลหรือข้อมูลข่าวสารที่ต้องการความปลอดภัย (Secure RSS) โดยรายละเอียดของข้อมูลข่าวสารจะถูกระบุไว้ที่แท็ก <description> ของแต่ละรายการ

ดังนั้นเพื่อให้ RSS สามารถเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัยได้ วิทยานิพนธ์นี้จึงสร้างแอททริบิวต์ (Attribute) เพิ่มเติม ดังแสดงในตารางที่ 3.1 สำหรับระบุค่าต่าง ๆ ให้กับแท็ก <description> ของเอกสาร RSS แต่ละรายการ

ตารางที่ 3.1 แอททริบิวต์ต่าง ๆ ของแท็ก <description>

แอททริบิวต์	คำอธิบาย
class	ระบุว่าข้อมูลภายในแท็ก <description> เป็นข้อมูลข่าวสารทั่วไป (mesg) หรือข้อมูลข่าวสารส่วนบุคคล (cipher)
username	ชื่อผู้รับบริการข้อมูลข่าวสาร
algorithm	อัลกอริทึมที่ใช้เข้ารหัสข้อมูลข่าวสารส่วนบุคคล

จากตารางที่ 3.1 การกำหนดแอททริบิวต์ให้กับแท็ก <description> แต่ละรายการสามารถทำได้ดังนี้

1) ข้อมูลข่าวสารทั่วไป ระบุเฉพาะแอททริบิวต์ class โดยมีรูปแบบดังนี้

```
<description class = “mesg” >Message</description>
```

class = “mesg” คือ การระบุว่าข้อมูลภายในแท็ก <description> เป็นข้อมูลข่าวสารทั่วไป

Message คือ รายละเอียดข้อมูลข่าวสารทั่วไป

2) ข้อมูลข่าวสารส่วนบุคคล ระบุแอททริบิวต์ทั้ง 3 โดยมีรูปแบบดังนี้

```
<description class = “cipher” username = “Username” algorithm = “Encryption Algorithm” >
Cipher Key:Cipher Message</description>
```

class = “cipher”	คือ การระบุว่าข้อมูลภายในแท็ก <description> เป็นข้อมูลข่าวสารส่วนบุคคล
Username	คือ ชื่อผู้รับบริการข้อมูลข่าวสาร
Encryption Algorithm	คือ อัลกอริทึมที่ใช้เข้ารหัสข้อมูลข่าวสารส่วนบุคคล
Cipher Key	คือ ข้อความไซเฟอร์ของกุญแจถอดรหัส
Cipher Message	คือ ข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล

3.1.4 เผยแพร่เอกสาร RSS

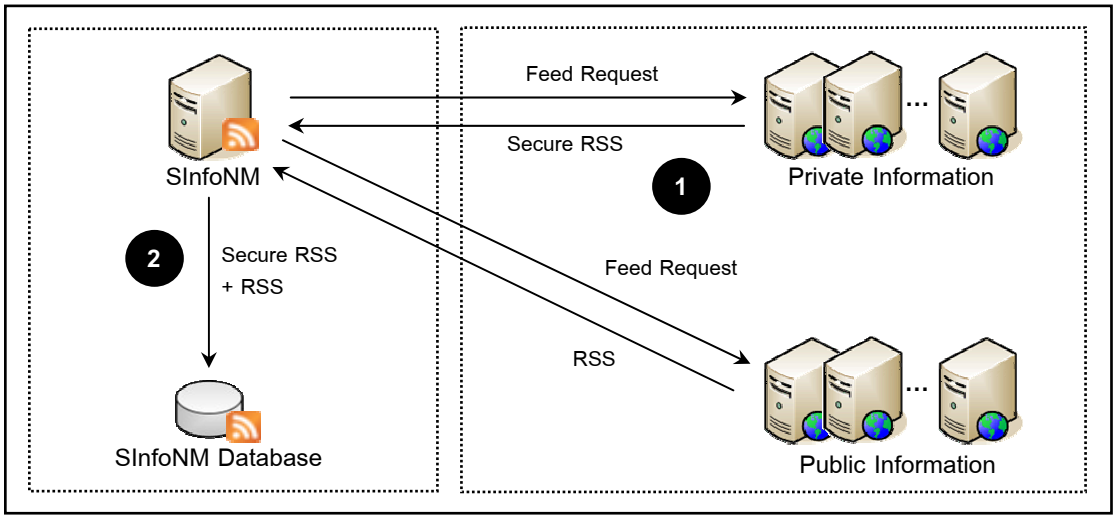
เมื่อกำหนดรูปแบบข้อมูลข่าวสารแต่ละรายการภายในเอกสาร RSS ดังหัวข้อ 3.1.3 เรียบร้อยแล้ว จะได้เอกสาร RSS ที่ประกอบด้วยข้อมูลข่าวสารทั่วไป และข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้ โดยการเผยแพร่เอกสาร RSS สามารถกระทำได้เช่นเดียวกับการเผยแพร่เอกสาร RSS ทั่วไป

3.2 กลไกการรวบรวมเอกสาร RSS (SInfoNM Aggregator: SInfoNMA)

กลไกการรวบรวมเอกสาร RSS ทำหน้าที่รวบรวมเอกสาร RSS จากเว็บไซต์ต่าง ๆ เก็บไว้ในฐานข้อมูลเพื่อคัดกรองข้อมูลข่าวสารที่เกี่ยวข้อง ก่อนส่งไปยังผู้รับบริการ

3.2.1 แบบจำลองการทำงานส่วนรวบรวมเอกสาร RSS

แบบจำลองการทำงานส่วนรวบรวมเอกสาร RSS ดังแสดงในภาพประกอบ 3.3 ประกอบด้วยขั้นตอนการทำงาน 2 ขั้นตอนดังนี้



ภาพประกอบ 3.3 แบบจำลองการทำงานส่วนรวบรวมเอกสาร RSS

1) รวบรวมเอกสาร RSS

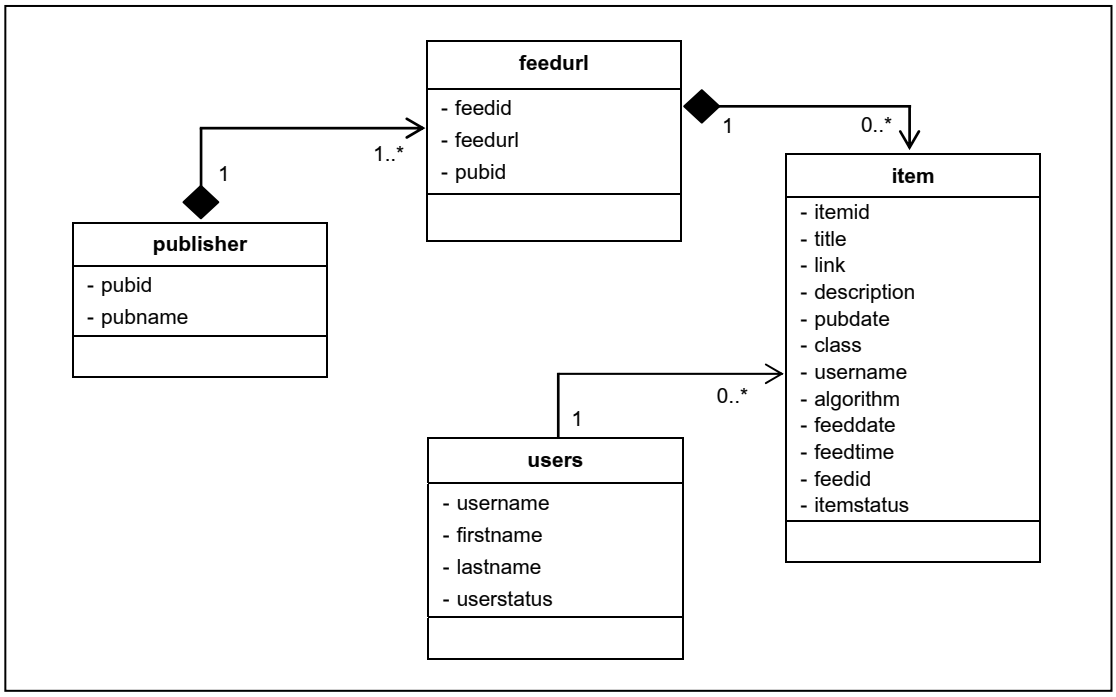
SInfoNM ร้องขอการทำงาน (Feed Request) ไปยังเว็บไซต์ต่าง ๆ เพื่อรวบรวบเอกสาร RSS ที่ประกอบด้วยข้อมูลข่าวสารทั่วไป (Public Information) และข้อมูลข่าวสารส่วนบุคคล (Private Information)

ในกรณีหน่วยงานไม่ได้จัดเตรียมเอกสาร RSS ไว้ ระบบจะสร้างข้อมูล RSS อัตโนมัติด้วยวิธีการค้นหาแท็ก HTML ต่าง ๆ ภายในหน้าเว็บเพจ โดยใช้ Regular Expression กำหนดให้แท็ก <title> ใน RSS channel สกัดจากแท็ก <title> ของ HTML และ URL ของเว็บเพจจะถูกนำมาใส่ไว้ในแท็ก <link> ของ RSS channel หากมี Metadata ใน HTML ที่เกี่ยวกับ description จะถูกนำมาใส่ไว้ในแท็ก <description> ของ RSS โดย RSS item จะพิจารณาจากแท็ก <a> ภายในหน้าเว็บเพจ เพื่อนำมาสร้างเป็นส่วน <title> และ <link> ในแต่ละแท็ก <item>

2) บันทึกข้อมูลเอกสาร RSS

2.1) นำเอกสาร RSS มาผ่านกระบวนการวิเคราะห์โครงสร้าง เพื่อสกัดข้อมูลภายในแท็กและแอททริบิวต์ที่จำเป็นของแต่ละรายการ

2.2) บันทึกข้อมูลที่สกัดได้ลงฐานข้อมูล (SInfoNM Database) โดยโครงสร้างการจัดเก็บข้อมูล RSS แสดงดังภาพประกอบ 3.4



ภาพประกอบ 3.4 โครงสร้างการจัดเก็บข้อมูล RSS

จากภาพประกอบ 3.4 โครงสร้างข้อมูลตารางต่าง ๆ มีดังนี้

2.2.1) ตาราง publisher ใช้สำหรับเก็บข้อมูลต่าง ๆ ของผู้เผยแพร่ข้อมูลข่าวสาร แสดงดังตารางที่ 3.2

ตารางที่ 3.2 โครงสร้างข้อมูลตาราง publisher

แอททริบิวต์	คำอธิบาย	ประเภท	ขนาด	หมายเหตุ
pubid	รหัสผู้เผยแพร่ข้อมูลข่าวสาร	INT	3	PK
pubname	ชื่อผู้เผยแพร่ข้อมูลข่าวสาร	VARCHAR	200	

2.2.2) ตาราง items ใช้สำหรับเก็บรายการข้อมูลข่าวสารของแท็ก <item> ภายในเอกสาร RSS และรายละเอียดเพิ่มเติมในการบันทึกรายการ เช่น วันเวลาที่บันทึกรายการ เป็นต้น แสดงดังตารางที่ 3.3

ตารางที่ 3.3 โครงสร้างข้อมูลตาราง items

แอททริบิวต์	คำอธิบาย	ประเภท	ขนาด	หมายเหตุ
itemid	รหัสรายการข่าว	INT	10	PK
title	หัวข้อข่าว	VARCHAR	200	
link	ที่อยู่ของรายละเอียดข่าว	VARCHAR	200	
description	รายละเอียดข่าวโดยย่อ	LONGTEXT	-	
pubdate	วันเวลาเผยแพร่รายการข่าว	VARCHAR	100	
class	ระบุว่าเป็นข้อมูลข่าวสารทั่วไป หรือข้อมูลข่าวสารส่วนบุคคล	VARCHAR	10	
username	ชื่อสำหรับใช้งานระบบ	VARCHAR	20	FK
algorithm	อัลกอริทึมที่ใช้เข้ารหัสข้อมูล	VARCHAR	200	
feeddate	วันที่บันทึกรายการข่าว	VARCHAR	8	
feedtime	เวลาบันทึกรายการข่าว	VARCHAR	8	
feedid	รหัสที่อยู่เอกสาร RSS	INT	3	FK
itemstatus	สถานะรายการข่าว	ENUM	‘Y’, ‘N’	

2.2.3) ตาราง feedurl ใช้สำหรับเก็บที่อยู่เอกสาร RSS หรือเว็บเพจของผู้เผยแพร่ข้อมูลข่าวสาร โดยผู้เผยแพร่สามารถมีที่อยู่เอกสาร RSS หรือเว็บเพจได้มากกว่าหนึ่งที่อยู่ แสดงดังตารางที่ 3.4

ตารางที่ 3.4 โครงสร้างข้อมูลตาราง feedurl

แอททริบิวต์	คำอธิบาย	ประเภท	ขนาด	หมายเหตุ
feedid	รหัสที่อยู่เอกสาร RSS	INT	3	PK
feedurl	ที่อยู่เอกสาร RSS	LONGTEXT	-	
pubid	รหัสผู้เผยแพร่ข้อมูลข่าวสาร	INT	2	FK

2.2.4) ตาราง users ใช้สำหรับเก็บข้อมูลผู้รับบริการข้อมูลข่าวสาร แสดงดังตารางที่ 3.5

ตารางที่ 3.5 โครงสร้างข้อมูลตาราง users

แอททริบิวต์	คำอธิบาย	ประเภท	ขนาด	หมายเหตุ
username	ชื่อสำหรับใช้งานระบบ	VARCHAR	20	PK
firstname	ชื่อผู้รับข้อมูลข่าวสาร	LONGTEXT	-	
lastname	นามสกุลผู้รับข้อมูลข่าวสาร	INT	2	
userstatus	สถานะผู้รับข้อมูลข่าวสาร	ENUM	'Y', 'N'	

3.2.2 ขั้นตอนวิธีรวบรวมเอกสาร RSS

ขั้นตอนวิธีโดยรวมของการรวบรวมเอกสาร RSS แสดงดังภาพประกอบ 3.5

```
1   Method SInfoNMA (feedUrl, PubInfo, SecInfo)
2       for each feedUrl
3             download feed content (PubInfo and SecInfo) from feedUrl
4           for each item in the feed
5             get data in <title>, <description>, <link>, <pubDate>, class, username, algorithm
6             store in SInfoNM database
7           end for
8       end for
9   end method
```

ภาพประกอบ 3.5 ขั้นตอนวิธีรวบรวมเอกสาร RSS

จากภาพประกอบ 3.5 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 2–8 คือ การรวบรวมเอกสาร RSS จาก URL ที่บันทึกไว้ในตาราง feedurl ทีละเว็บไซต์

บรรทัดที่ 4–7 คือ กระบวนการวิเคราะห์โครงสร้าง เพื่อสกัดข้อมูลจากเอกสาร RSS และบันทึกลงฐานข้อมูล

3.3 กลไกการสร้างเอกสาร RSS เฉพาะผู้ใช้ (SInfoNM Generator: SInfoNMG)

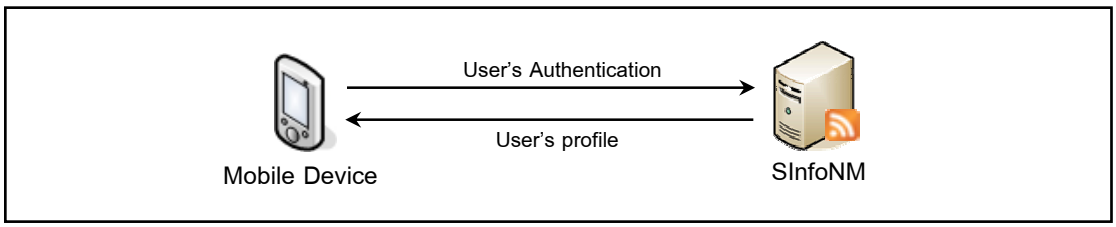
กลไกการสร้างเอกสาร RSS เฉพาะผู้ใช้ ทำหน้าที่รวบรวมข้อมูลข่าวสารที่เกี่ยวข้องกับผู้ใช้จากฐานข้อมูล และสร้างเป็นเอกสาร RSS ส่งกลับไปยังผู้ใช้ ประกอบด้วยส่วนต่าง ๆ ดังนี้

3.3.1 ลงทะเบียนรับข้อมูลข่าวสาร

การลงทะเบียนรับข้อมูลข่าวสารเป็นการยืนยันตัวบุคคล เพื่อนำไปใช้ในการสืบค้นข้อมูลข่าวสารส่วนบุคคลภายในเอกสาร RSS แสดงดังภาพประกอบ 3.6 โดยมีขั้นตอนการทำงานดังนี้

1) ผู้ใช้สมัครเข้าใช้งานระบบด้วย Username และ Password ที่องค์กรกำหนดให้

2) ระบบสร้างไฟล์ข้อมูลส่วนตัวผู้ใช้ (User's profile) ไว้บนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้ ซึ่งประกอบด้วยชื่อผู้ใช้ (Username) และกุญแจส่วนตัว (Private Key) ที่ถูกเข้ารหัสไว้ สำหรับถอดรหัสข้อมูลข่าวสารส่วนบุคคลภายในเอกสาร RSS

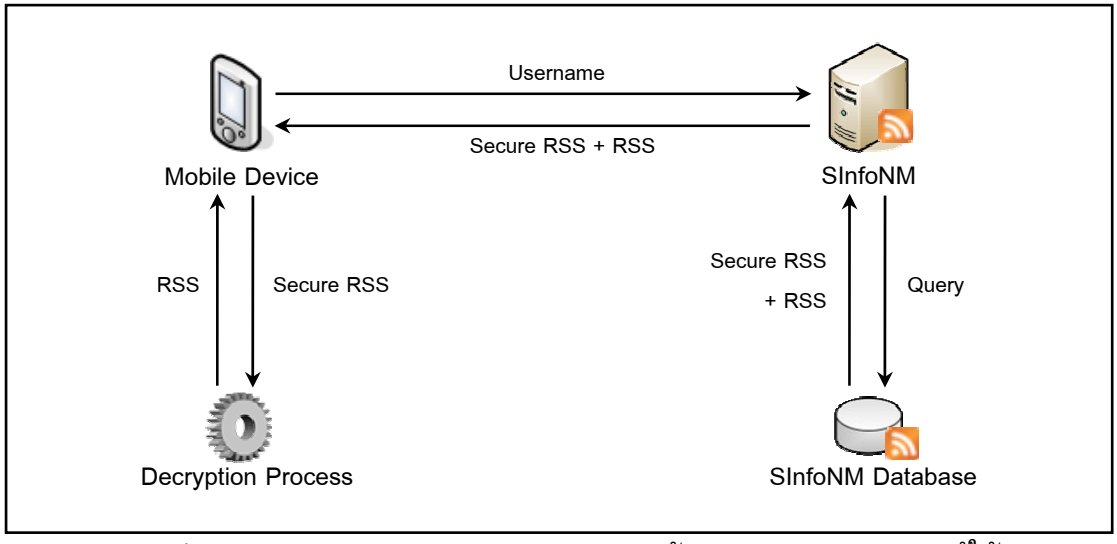


ภาพประกอบ 3.6 ลงทะเบียนรับข้อมูลข่าวสาร

โดยการเข้าใช้งานครั้งต่อไประบบจะทำการอ่านข้อมูลจากไฟล์ข้อมูลส่วนตัวผู้ใช้เพื่อยืนยันตัวบุคคล และสืบค้นข้อมูลข่าวสารจากฐานข้อมูล

3.3.2 สร้างเอกสาร RSS เฉพาะผู้ใช้

เมื่อผู้ใช้ลงทะเบียนรับข้อมูลข่าวสารเรียบร้อยแล้ว ระบบจะสร้างเอกสาร RSS ส่งกลับไปยังอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้ โดยการทำงานสามารถสร้างเป็นแบบจำลองได้ดังภาพประกอบ 3.7



ภาพประกอบ 3.7 แบบจำลองการทำงานส่วนสร้างเอกสาร RSS เฉพาะผู้ใช้

แบบจำลองการทำงานส่วนสร้างเอกสาร RSS เฉพาะผู้ใช้ ดังภาพประกอบ 3.7 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1) SInfoNM ตรวจสอบ Username จากไฟล์ข้อมูลส่วนตัวบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้ เพื่อยืนยันตัวบุคคล

2) สืบค้นข้อมูล RSS โดยสอบถาม (Query) ไปยังฐานข้อมูลข่าวสาร (SInfoNM Database) ด้วย Username ของผู้ใช้

3) สร้างเอกสาร RSS สำหรับผู้ใช้ ประกอบด้วยข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล ส่งกลับไปยังผู้ใช้

4) ข้อมูลข่าวสารส่วนบุคคลที่ผู้ใช้ได้รับ จะถูกส่งไปยังกระบวนการถอดรหัส

5) เมื่อถอดรหัสเรียบร้อยแล้ว ผลลัพธ์ที่ได้จะถูกแสดงบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้

ขั้นตอนวิธีโดยรวมของการสร้างเอกสาร RSS เฉพาะผู้ใช้ แสดงดังภาพประกอบ 3.8

```
1   Method SInfoNMG (username)
2     search username in SInfoNM database to generate RSS document
3     send RSS document to user
4     if (class = “cipher”)
5         Call decryption function (cipher key, cipher message, algorithm)
6     end if
7     show content to user
8   end method
```

ภาพประกอบ 3.8 ขั้นตอนวิธีสร้างเอกสาร RSS เฉพาะผู้ใช้

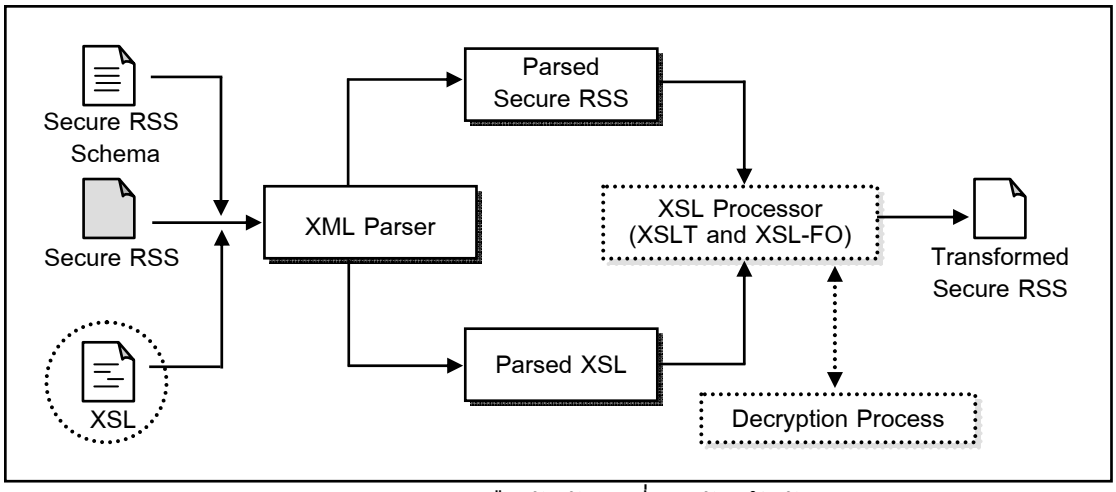
จากภาพประกอบ 3.8 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 2–3 คือ การสร้างเอกสาร RSS และส่งเอกสาร RSS ที่ได้ไปยังผู้ใช้

บรรทัดที่ 4–6 คือ การเรียกใช้งานฟังก์ชันถอดรหัส เพื่อถอดรหัสข้อความไซเฟอร์ข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้

3.3.3 การสืบค้นข้อมูลที่ถูกเข้ารหัส

วิทยานิพนธ์นี้ได้นำวิธีการแสดงผลเอกสาร XML ด้วย XSL มาใช้เพื่อสืบค้นข้อมูลที่ถูกเข้ารหัสไว้ แสดงดังภาพประกอบ 3.9

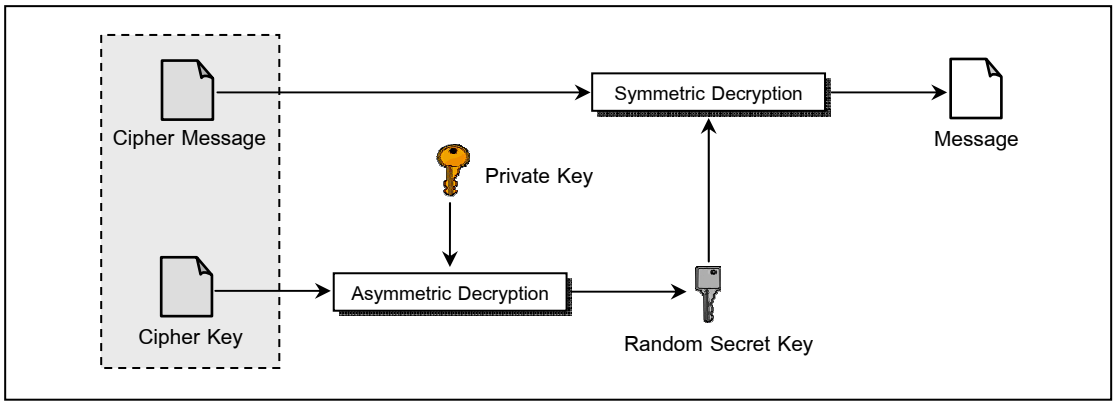


ภาพประกอบ 3.9 การสืบค้นข้อมูลที่ถูกเข้ารหัสด้วย XSL

จากภาพประกอบ 3.9 การสืบค้นข้อมูลที่ถูกเข้ารหัสใช้ไวยากรณ์ของ XPath ระบุรูปแบบข้อมูลที่ถูกเข้ารหัสไว้ในเอกสาร XSL เพื่อให้ XSL Processor สืบค้นข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้ในเอกสาร RSS และส่งข้อมูลที่สืบค้นได้ไปยังกระบวนการถอดรหัส ผลลัพธ์ที่ได้จะถูกแทนที่ลงในเอกสาร RSS ก่อนแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้

3.3.4 กระบวนการถอดรหัส

กระบวนการถอดรหัสข้อมูลข่าวสารส่วนบุคคล แสดงดังภาพประกอบ 3.10



ภาพประกอบ 3.10 การถอดรหัสข้อมูลข่าวสารส่วนบุคคลภายในเอกสาร RSS

จากภาพประกอบ 3.10 กระบวนการถอดรหัสข้อมูลข่าวสารส่วนบุคคลภายในเอกสาร RSS จะทำงานที่ฝั่งเครื่องผู้ใช้ โดยมีขั้นตอนการทำงานดังนี้

1) อ่านกุญแจถอดรหัสจากไฟล์ข้อมูลส่วนตัวบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้

2) ถอดรหัสกุญแจที่ได้จากขั้นตอนที่ 1) ได้เป็นกุญแจส่วนตัว (Private Key) เพื่อใช้ถอดรหัสข้อความไซเฟอร์ของกุญแจถอดรหัส (Cipher Key)

3) นำกุญแจส่วนตัวที่ได้ไปถอดรหัสข้อความไซเฟอร์ของกุญแจถอดรหัส (Cipher Key) ได้เป็นกุญแจที่เกิดจากการสุ่ม (Random Secret Key)

4) นำกุญแจที่เกิดจากการสุ่มไปถอดรหัสข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล (Cipher Message) ได้เป็นข้อความต้นฉบับ (Message)

3.3.5 การแสดงข้อมูลเอกสาร RSS

การแสดงข้อมูลข่าวสารทั่วไป และข้อมูลข่าวสารส่วนบุคคล ภายในเอกสาร RSS ใช้ XSL เพื่อเปลี่ยนเอกสาร RSS ให้อยู่ในรูปแบบเอกสาร HTML ก่อนแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้

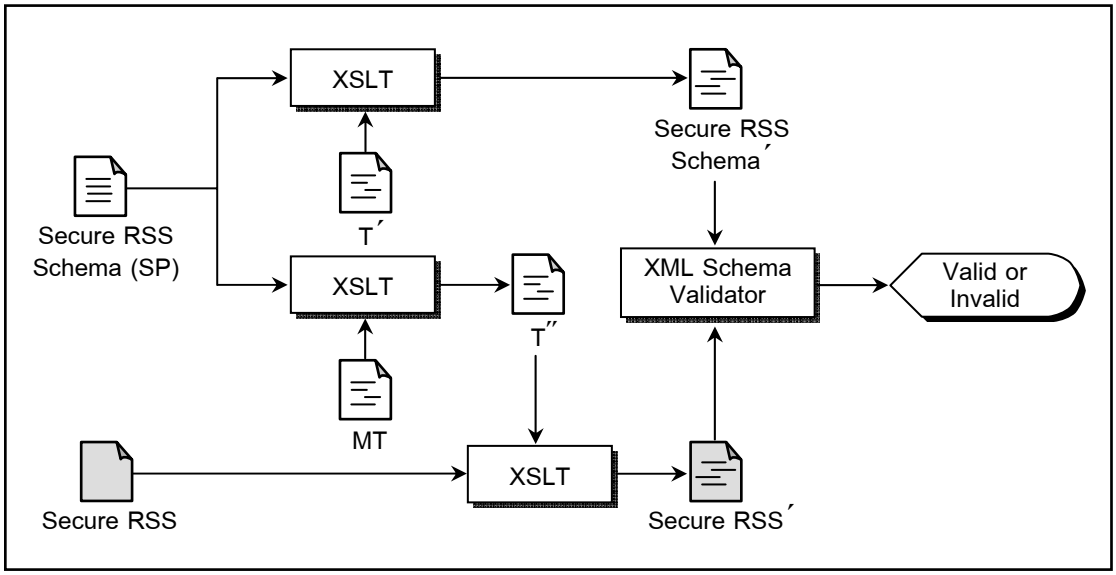
3.4 กลไกการตรวจสอบความถูกต้องเอกสาร RSS (SInfoNM Validator: SInfoNMV)

กลไกการตรวจสอบความถูกต้องเอกสาร RSS เป็นการนิยามโครงสร้างเอกสาร RSS สำหรับสารสนเทศที่ต้องการความปลอดภัย วัตถุประสงค์เพื่อทำให้เอกสาร RSS มีรูปแบบถูกต้องตรงตามที่ได้กำหนดไว้หรือมีคุณสมบัติ Valid พร้อมที่จะนำไปใช้งานสำหรับเผยแพร่สารสนเทศที่ต้องการความปลอดภัย

วิทยานิพนธ์นี้ใช้ XML Schema เพื่อนิยามโครงสร้างเอกสาร RSS ที่ต้องการความปลอดภัย (Secure RSS Schema) อย่างไรก็ตาม XML Schema ยังมีข้อจำกัดในเรื่องการกำหนดชนิดข้อมูลและรูปแบบให้กับอิลิเมนต์ตามแอททริบิวต์ที่กำหนด วิทยานิพนธ์นี้จึงนำ SchemaPath ซึ่งเป็นส่วนขยายสำหรับกำหนดเงื่อนไขของ XML Schema มาใช้งานเพื่อกำหนดรูปแบบการเปลี่ยนโครงสร้างเอกสาร Secure RSS Schema และเอกสาร Secure RSS ด้วย XSLT ให้อยู่ในรูปแบบที่สามารถตรวจสอบคุณสมบัติ Valid กับ XML Schema Validator ที่มีอยู่ทั่วไปได้

3.4.1 การตรวจสอบคุณสมบัติ Valid ด้วย Secure RSS Schema

ขั้นตอนการตรวจสอบคุณสมบัติ Valid ด้วย Secure RSS Schema แสดงดังภาพประกอบ 3.11



ภาพประกอบ 3.11 ขั้นตอนการตรวจสอบคุณสมบัติ Valid ด้วย Secure RSS Schema

จากภาพประกอบ 3.11 การตรวจสอบคุณสมบัติ Valid ของเอกสาร RSS มีขั้นตอนการทำงานดังนี้

1) แปลงโครงสร้างเอกสาร Secure RSS Schema ที่เขียนด้วยไวยากรณ์ของ SchemaPath (SP) ให้อยู่ในรูปแบบของ XML Schema ทั่วไป ด้วยเอกสาร XSL T′ ได้เป็นเอกสาร Secure RSS Schema′

2) นำเอกสาร XSL MT ไปตรวจสอบกฎที่เขียนด้วยไวยากรณ์ของ SchemaPath ในเอกสาร Secure RSS Schema และสร้างเป็นแม่แบบเก็บไว้ในเอกสาร XSL T″ สำหรับแปลงโครงสร้างเอกสาร Secure RSS

3) แปลงโครงสร้างเอกสาร Secure RSS ตามแม่แบบที่กำหนดไว้ในเอกสาร XSL T″ ได้เป็นเอกสาร Secure RSS′

4) นำเอกสาร Secure RSS′ ไปตรวจสอบคุณสมบัติ Valid ตามนิยามโครงสร้างที่กำหนดไว้ในเอกสาร Secure RSS Schema′ ด้วย XML Schema Validator ที่มีอยู่ทั่วไป

3.4.2 นิยามโครงสร้างเอกสาร RSS ด้วย Secure RSS Schema

วิทยานิพนธ์นี้ได้เพิ่มชนิดและรูปแบบข้อมูลของแท็ก <description> ดังภาพประกอบ 3.12 และภาพประกอบ 3.13 ลงในเอกสาร RSS Schema ทั่วไปที่มี (Thelin, 2008: Online) สำหรับเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัย

```
1    <xsd:element name="description" >
2        <xsd:alt cond="@class='mesg'" type="rssMesgType" />
3        <xsd:alt cond="@class='cipher'" type="rssCipherType"/>
4    </xsd:element>
```

ภาพประกอบ 3.12 ไวยากรณ์ SchemaPath เพื่อกำหนดชนิดข้อมูลแท็ก <description>

จากภาพประกอบ 3.12 ใช้ไวยากรณ์ SchemaPath เพื่อกำหนดชนิดข้อมูลให้กับแท็ก <description> ดังนี้

บรรทัดที่ 1 คือ การเริ่มต้นกำหนดแท็ก <description>

บรรทัดที่ 2 คือ การกำหนดให้ข้อมูลข่าวสารทั่วไปมีชนิดข้อมูลเป็น rssMesgType โดยรายละเอียดของชนิดข้อมูลแสดงดังภาพประกอบ 3.13 บรรทัดที่ 1-8

บรรทัดที่ 3 คือ การกำหนดข้อมูลข่าวสารส่วนบุคคลมีชนิดข้อมูลเป็น rssCipherType โดยรายละเอียดของชนิดข้อมูลแสดงดังภาพประกอบ 3.13 บรรทัดที่ 9-36

บรรทัดที่ 4 คือ การสิ้นสุดการกำหนดแท็ก <desciption>

```
1   <xsd:complexType name="rssMesgType">
2         <xsd:simpleContent>
3               <xsd:extension base="xsd:string">
4                     <xsd:attribute name="class" type="xsd:string" use="required"
5                           fixed="mesg"/>
6               </xsd:extension>
7         </xsd:simpleContent>
8   </xsd:complexType>
9   <xsd:complexType  name="rssCipherType" >
10        <xsd:simpleContent>
11              <xsd:restriction base="atrrRssCipherType">
12              <xsd:pattern value="([a-zA-Z0-9\+/][a-zA-Z0-9\+/=\s]*[a-zA-Z0-9\+/=]):
13                    ([a-zA-Z0-9\+/][a-zA-Z0-9\+/=\s]*[a-zA-Z0-9\+/=])"/>
14                    <xsd:attribute name="class" type="xsd:string" use="required"
15                          fixed="cipher"/>
16                    <xsd:attribute name="algorithm" type="xsd:string" use="required"/>
17                    <xsd:attribute name="username" type="userNameType"
18                          use="required"/>
19              </xsd:restriction>
20        </xsd:simpleContent>
21  </xsd:complexType>
22  <xsd:complexType  name="atrrRssCipherType">
23        <xsd:simpleContent>
24              <xsd:extension base="xsd:string">
25                    <xsd:attribute name="class" type="xsd:string" use="required"
26                          fixed="cipher"/>
27                    <xsd:attribute name="algorithm" type="xsd:string" use= "required"/>
28                    <xsd:attribute name="username" type="userNameType" use="required"/>
29              </xsd:extension>
30        </xsd:simpleContent>
31  </xsd:complexType>
32  <xsd:simpleType name="userNameType" >
33        <xsd:restriction base="xsd:string">
34              <xsd:pattern value="([a-zA-Z0-9_\-])([a-zA-Z0-9_\-\.]*)"/>
35        </xsd:restriction>
36  </xsd:simpleType>
```

ภาพประกอบ 3.13 ชนิดข้อมูลของแท็ก <description>

จากภาพประกอบ 3.13 ใช้ไวยากรณ์ของ XML Schema เพื่อกำหนดชนิดข้อมูลให้กับอิลิเมนต์ <description> ดังนี้

บรรทัดที่ 1-8 คือ ชนิดข้อมูลของข้อมูลข่าวสารทั่วไป ประกอบด้วยแอททริบิวต์ class มีค่าเท่ากับ "mesg" และข้อมูลภายในแท็ก <description> กำหนดให้มีชนิดข้อมูลเป็น string

บรรทัดที่ 9-36 คือ ชนิดข้อมูลของข้อมูลข่าวสารส่วนบุคคล ประกอบด้วยแอททริบิวต์ class มีค่าเท่ากับ "cipher" แอททริบิวต์ username ใช้สำหรับกำหนดชื่อผู้ใช้และแอททริบิวต์ algorithm ใช้สำหรับระบุอัลกอริทึมที่ใช้เข้ารหัสข้อมูล โดยข้อมูลภายในแท็ก <description> กำหนดให้มีรูปแบบดังบรรทัดที่ 12 และ 13

3.4.3 นิยามโครงสร้างเอกสาร RSS ด้วย EBNF

ส่วนนี้จะใช้ EBNF เพื่ออธิบายไวยากรณ์ของอิลิเมนต์ description ภายในเอกสาร RSS สำหรับเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัย ซึ่งอ้างอิงถึงนิยามโครงสร้างเอกสาร XML บางส่วนที่ถูกกำหนดโดย W3C ดังภาพประกอบ 3.14 และวิทยานิพนธ์นี้ได้นิยามไวยากรณ์ของอิลิเมนต์ <description> เพิ่มเติมดังภาพประกอบ 3.15

[2]	Char	::=	#x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
[3]	S	::=	(#x20 | #x9 | #xD | #xA)+
[4]	NameStartChar	::=	":" | [A-Z] | "_" | [a-z] | [#xC0-#xD6] | [#xD8-#xF6] | [#xF8-#x2FF] | [#x370-#x37D] | [#x37F-#x1FFF] | [#x200C-#x200D] | [#x2070-#x218F] | [#x2C00-#x2FEF] | [#x3001-#xD7FF] | [#xF900-#xFDCF] | [#xFDF0-#xFFFD] | [#x10000-#xEFFFF]
[4a]	NameChar	::=	NameStartChar | "-" | "." | [0-9] | #xB7 | [#x0300-#x036F] | [#x203F-#x2040]
[5]	Name	::=	NameStartChar (NameChar)*
[10]	AttValue	::=	'"' ([^<&"] | Reference)* '"' | "'" ([^<&'] | Reference)* "'"
[25]	Eq	::=	S? '=' S?
[66]	CharRef	::=	'&#' [0-9]+ ';' | '&#x' [0-9a-fA-F]+ ';'
[67]	Reference	::=	EntityRef | CharRef
[68]	EntityRef	::=	'&' Name ';'

ภาพประกอบ 3.14 นิยามโครงสร้างเอกสาร XML ด้วย EBNF (บางส่วน)

(Bray *et al*., 2008: Online)

```
DescItemElem      ::=   DescItemSTag DescItemType DescItemETag
DescItemSTag      ::=   '<description' S
DescItemETag      ::=   '</description' S? '>'
DescItemType      ::=   MesgAttr MesgContent | CipherAttr CipherContent
MesgAttr          ::=   'class' Eq MesgValue S? '>'
MesgValue         ::=   "'" mesg "'" | '"' mesg '"'
MesgContent       ::=   Char (Char)*
UserNameSChar     ::=   [a-z] | [A-Z] | [0-9] | "_" | "-"
UserNameChar      ::=   UserNameSChar | "."
UserNameValue     ::=   "'" UserNameStartChar (UserNameChar)* "'"
                        | '"' UserNameStartChar (UserNameChar)* '"'
CipherAttr        ::=   'class' Eq CipherValue S 'username' Eq UserNameValue S
                        'algorithm' Eq AttValue S? '>'
CipherValue       ::=   "'" cipher "'" | '"' cipher '"'
CipherSChar       ::=   [a-z] | [A-Z] | [0-9] | "+" | "/"
CipherEChar       ::=   [a-z] | [A-Z] | [0-9] | "_" | "-" | "="
CipherChar        ::=   CipherSChar | "=" | #x20
CipherContent     ::=   CipherSChar (CipherChar)* CipherEChar ':' CipherSChar
                        (CipherChar)* CipherEChar
```

ภาพประกอบ 3.15 นิยามโครงสร้างแท็ก <description> ด้วย EBNF

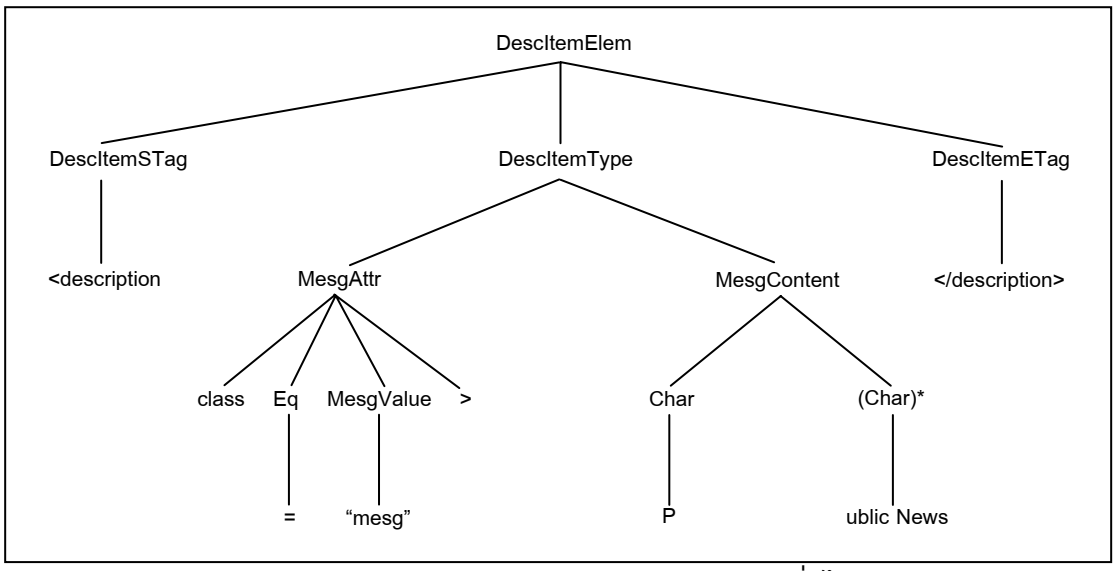
ตัวอย่างการ Derivation ตามไวยากรณ์ที่ได้กำหนด สำหรับข้อมูลข่าวสารทั่วไป แสดงดังภาพประกอบ 3.16 และสามารถอธิบายไวยากรณ์ของการ Derivation ด้วย Parse Tree ดังภาพประกอบ 3.17

Example element: <description class="mesg">Public News</description>

Derivation:

```
DescItemElem  =>  DescItemSTag DescItemType DescItemETag
              =>  <description DescItemType DescItemETag
              =>  <description MesgAttr MesgContent DescItemETag
              =>  <description class Eq MesgValue> MesgContent DescItemETag
              =>  <description class = MesgValue> MesgContent DescItemETag
              =>  <description class = "mesg"> MesgContent DescItemETag
              =>  <description class = "mesg"> Char (Char)* DescItemETag
              =>  <description class = "mesg"> P (Char)* DescItemETag
              =>  <description class = "mesg"> Public News DescItemETag
              =>  <description class= "mesg"> Public News </description>
```

ภาพประกอบ 3.16 Derivation ข้อมูลข่าวสารทั่วไป



ภาพประกอบ 3.17 Parse Tree ข้อมูลข่าวสารทั่วไป

ตัวอย่างการ Derivation ตามไวยากรณ์ที่ได้กำหนด สำหรับข้อมูลข่าวสารส่วนบุคคล แสดงดังภาพประกอบ 3.18 และสามารถอธิบายไวยากรณ์ของการ Derivation ด้วย Parse Tree ดังภาพประกอบ 3.19

Example element: <description class=“cipher” username=“5010220120” algorithm= “RSA”>F633:2e5e</description>

Derivation:

DescItemElem => DescItemSTag DescItemType DescItemETag
=> <description DescItemType DescItemETag
=> <description CipherAttr CipherContent DescItemETag
=> <description class Eq CipherValue username Eq UserNameValue algorithm Eq AttValue> CipherContent DescItemETag
=> <description class = CipherValue username Eq UserNameValue algorithm Eq AttValue> CipherContent DescItemETag
=> <description class = “cipher” username Eq UserNameValue algorithm Eq AttValue> CipherContent DescItemETag
=> <description class = “cipher” username = UserNameValue algorithm Eq AttValue> CipherContent DescItemETag
=> <description class = “cipher” username = “UserNameSChar (UserNameChar)*” algorithm Eq AttValue> CipherContent DescItemETag

ภาพประกอบ 3.18 Derivation ข้อมูลข่าวสารส่วนบุคคล

=> <description class = “cipher” username = “5 (UserNameChar)*” algorithm Eq AttValue> CipherContent DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm Eq AttValue> CipherContent DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = AttValue> CipherContent DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”> CipherContent DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”> CipherSChar (CipherChar)* CipherEChar : CipherSChar (CipherChar)* CipherEChar DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F (CipherChar)* CipherEChar : CipherSChar (CipherChar)* CipherEChar DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F66 CipherEChar : cipherSChar (CipherChar)* CipherEChar DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F663: cipherSChar (CipherChar)* CipherEChar DescItemETag

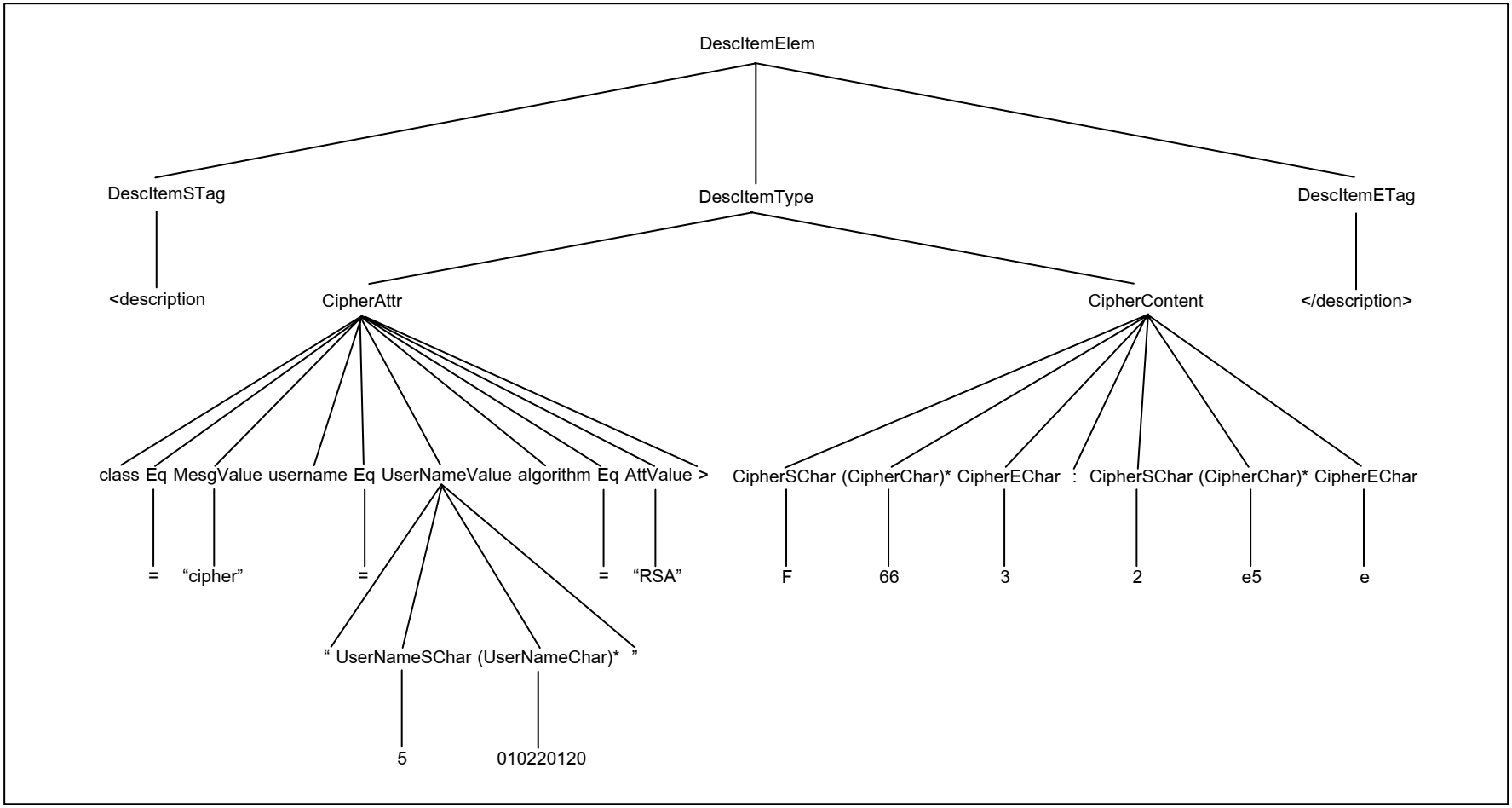
=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F663:2 (CipherChar)* CipherEChar DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F663:2e5 CipherEChar DescItemETag

=> <description class = “cipher” username = “5010220120” algorithm = “RSA”>F663:2e5e DescItemETag

=> <description class=“cipher” username=“5010220120” algorithm=“RSA”>F663:2e5e</description>

ภาพประกอบ 3.18 Derivation ข้อมูลข่าวสารส่วนบุคคล (ต่อ)



ภาพประกอบ 3.19 Parse Tree ข้อมูลข่าวสารส่วนบุคคล

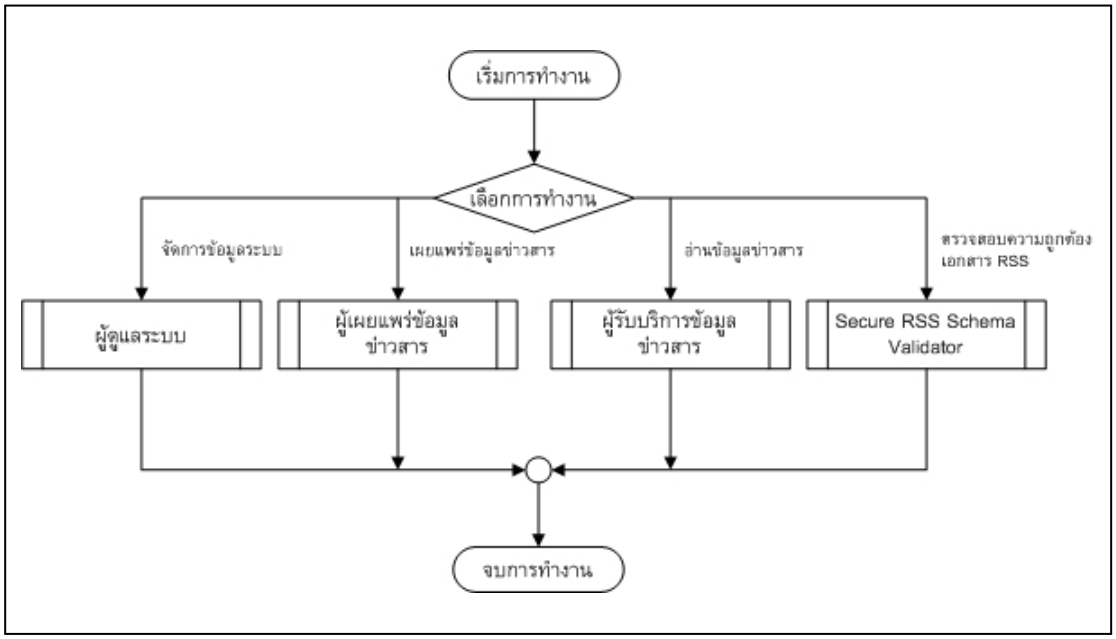
บทที่ 4

การพัฒนาระบบและผลการศึกษา

วิทยานิพนธ์นี้ได้ออกแบบกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP (SInfoNM) ดังรายละเอียดในบทที่ 3 และเพื่อให้กลไกที่ออกแบบไว้สามารถทำงานได้ตามวัตถุประสงค์ จึงพัฒนาระบบเพื่อทดสอบกลไกการทำงานดังกล่าว โดยสามารถอธิบายส่วนต่าง ๆ ของระบบได้ดังนี้

4.1 ผังการทำงานของระบบ

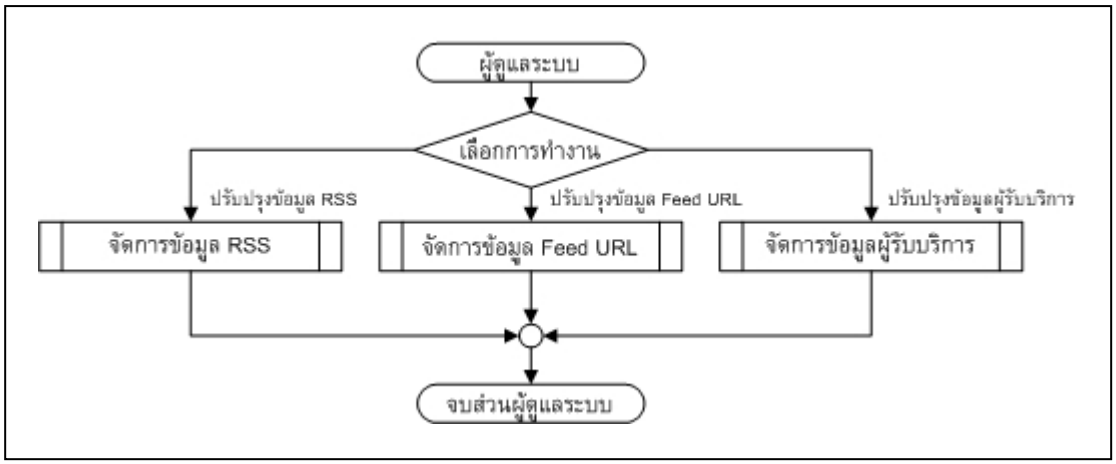
ระบบแบ่งการทำงานออกเป็น 4 ส่วน คือ ส่วนผู้ดูแลระบบ ส่วนผู้เผยแพร่ข้อมูลข่าวสาร ส่วนผู้รับบริการข้อมูลข่าวสาร และส่วน Secure RSS Schema Validator แสดงดังภาพประกอบ 4.1 โดยสามารถอธิบายรายละเอียดการทำงานส่วนต่าง ๆ ได้ดังนี้



ภาพประกอบ 4.1 ผังงานระบบ

4.1.1 ส่วนผู้ดูแลระบบ

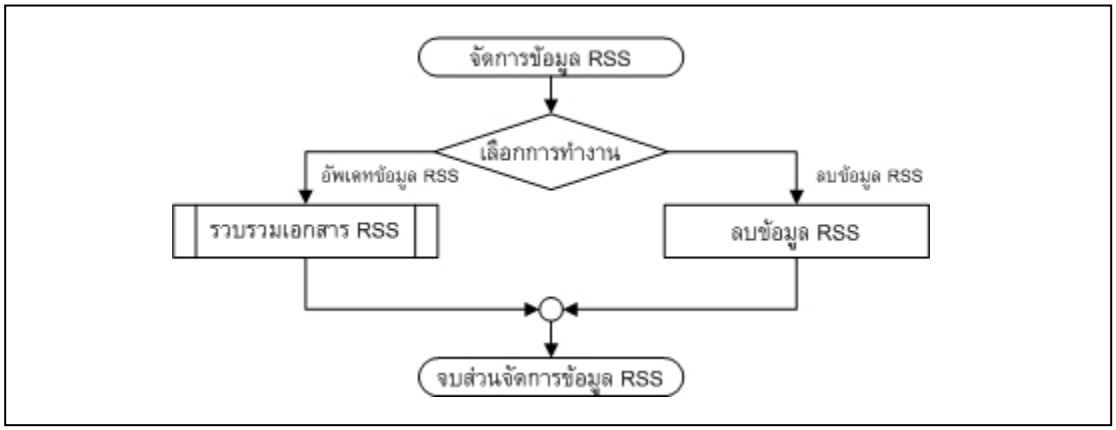
ผู้ดูแลระบบมีหน้าที่จัดการข้อมูลต่าง ๆ ของระบบ ซึ่งประกอบด้วยข้อมูลเอกสาร RSS ข้อมูล Feed URL และข้อมูลผู้รับบริการ โดยผังงานจัดการข้อมูลระบบ แสดงดังภาพประกอบ 4.2



ภาพประกอบ 4.2 ผังงานผู้ดูแลระบบ

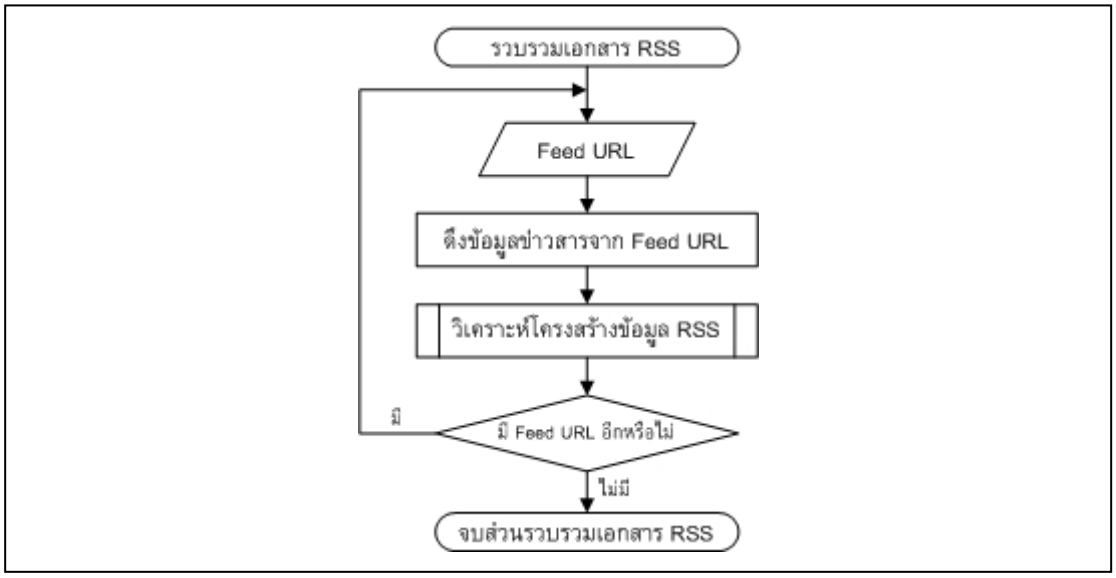
1) จัดการข้อมูล RSS

การจัดการข้อมูล RSS เป็นส่วนที่ผู้ดูแลระบบใช้เพื่อรวบรวมและลบข้อมูล RSS ในกรณีที่ต้องการปรับปรุงฐานข้อมูล ณ เวลาปัจจุบัน โดยผังงานจัดการข้อมูล RSS แสดงดังภาพประกอบ 4.3

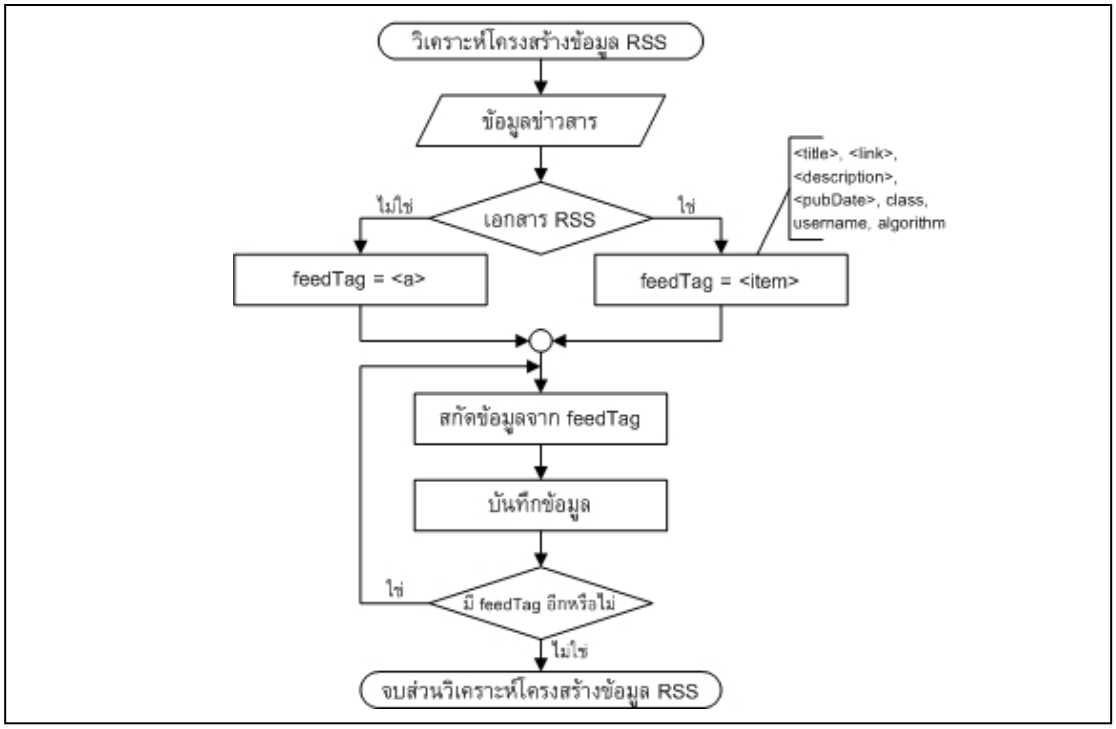


ภาพประกอบ 4.3 ผังงานจัดการข้อมูล RSS

จากภาพประกอบ 4.3 สามารถอธิบายการทำงานของส่วนรวบรวมเอกสาร RSS ได้ดังภาพประกอบ 4.4 และผังงานวิเคราะห์โครงสร้างข้อมูล RSS แสดงดังภาพประกอบ 4.5 โดยกรณีไม่ได้จัดเตรียมเอกสาร RSS ไว้ ระบบจะวิเคราะห์โครงสร้างข้อมูลจากแท็ก **<a>** ของ HTML และกำหนดข้อมูลที่ได้ตามโครงสร้างของเอกสาร RSS อัตโนมัติ



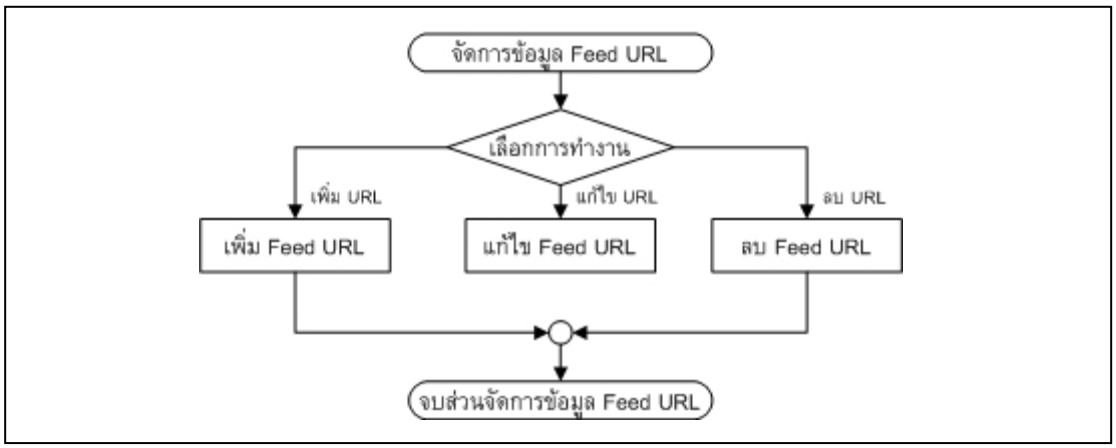
ภาพประกอบ 4.4 ผังงานรวบรวมเอกสาร RSS



ภาพประกอบ 4.5 ผังงานวิเคราะห์โครงสร้างข้อมูล RSS

2) จัดการข้อมูล Feed URL

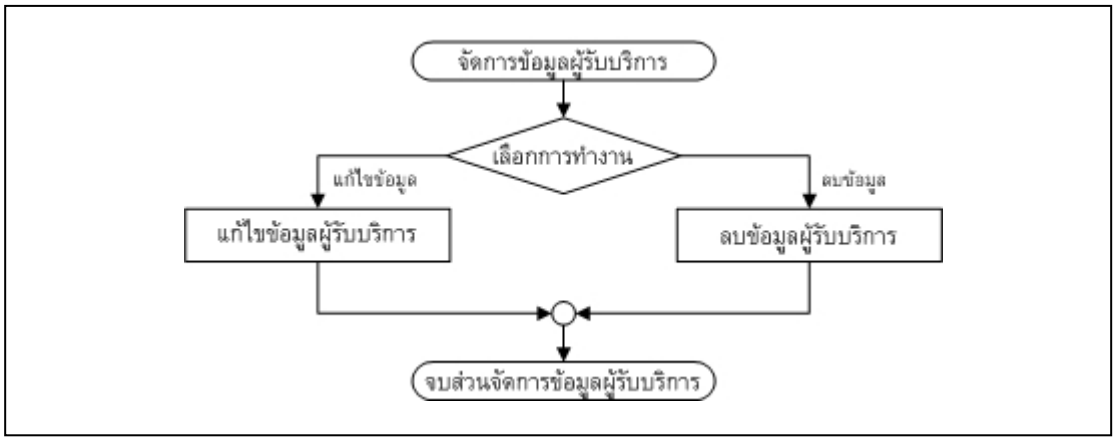
การจัดการข้อมูล Feed URL เป็นการเพิ่ม ลบ และแก้ไข URL ที่อยู่ของเอกสาร RSS ที่ต้องการรวบรวม ซึ่งผังการทำงานแสดงดังภาพประกอบ 4.6



ภาพประกอบ 4.6 ผังงานจัดการข้อมูล Feed URL

3) จัดการข้อมูลผู้รับบริการ

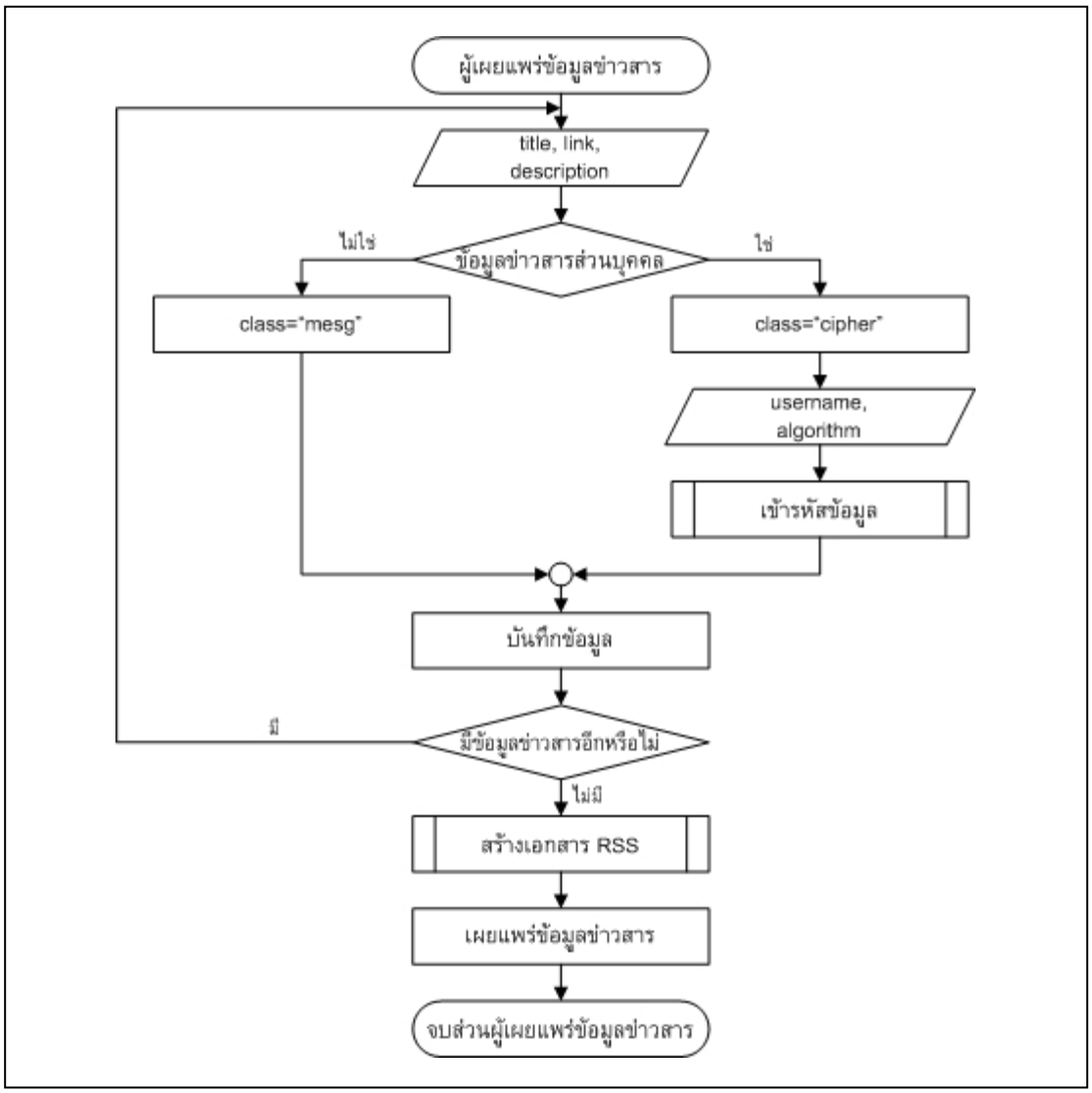
การจัดการข้อมูลผู้รับบริการ เป็นการแก้ไขและลบข้อมูลผู้รับบริการ แสดงดังภาพประกอบ 4.7



ภาพประกอบ 4.7 ผังงานจัดการข้อมูลผู้รับบริการ

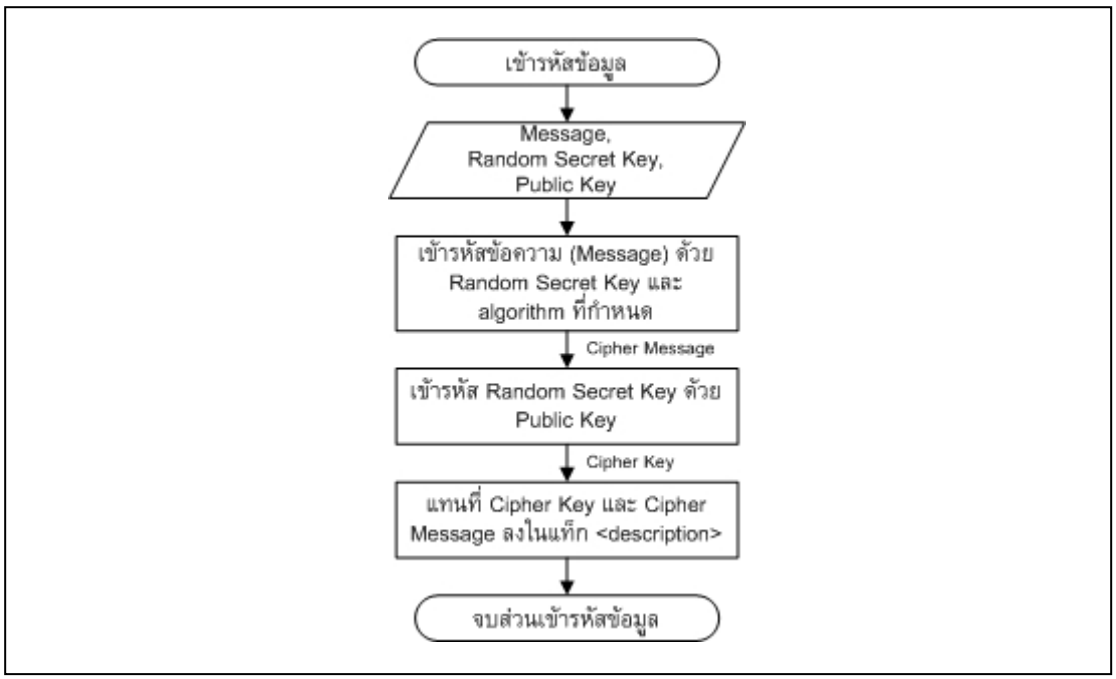
4.1.2 ส่วนผู้เผยแพร่ข้อมูลข่าวสาร

ผู้เผยแพร่ข้อมูลข่าวสารสร้างเอกสาร RSS ที่ประกอบด้วยข้อมูลข่าวสารทั่วไป และข้อมูลข่าวสารส่วนบุคคล เพื่อเผยแพร่ข้อมูลข่าวสารผ่านหน้าเว็บไซต์



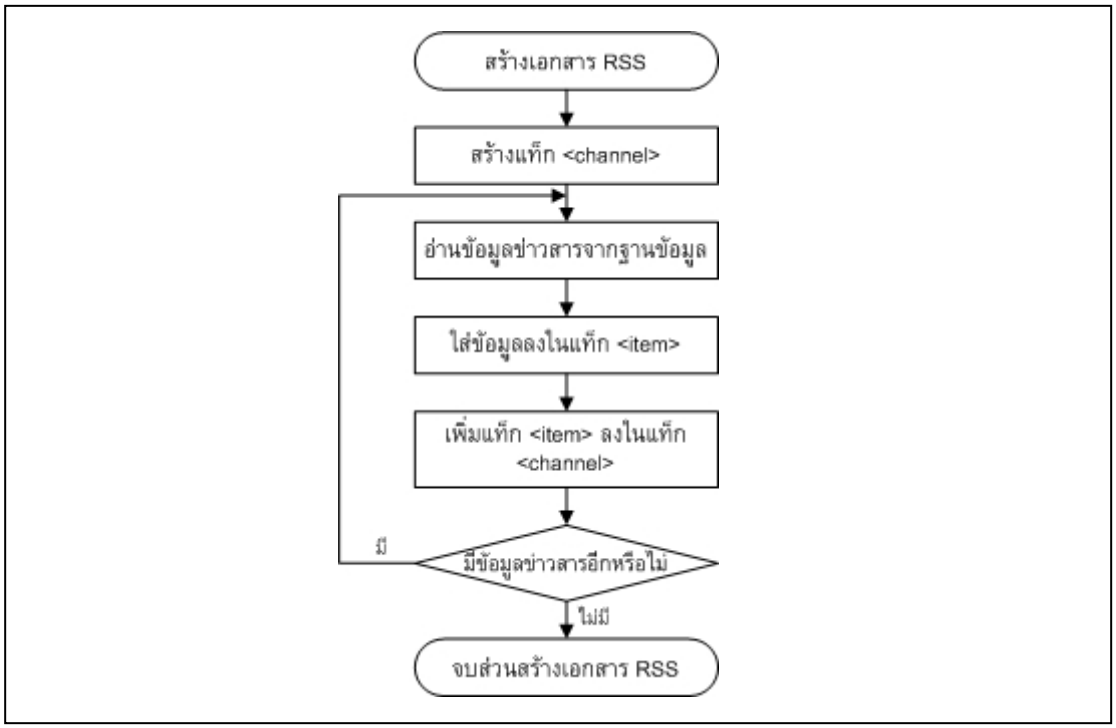
ภาพประกอบ 4.8 ผังงานผู้เผยแพร่ข้อมูลข่าวสาร

จากภาพประกอบ 4.8 ข้อมูลข่าวสารส่วนบุคคลจะถูกเข้ารหัสก่อนบันทึกและเผยแพร่ผ่านเว็บไซต์ โดยขั้นตอนการเข้ารหัสแสดงดังภาพประกอบ 4.9



ภาพประกอบ 4.9 ผังงานเข้ารหัสข้อมูล

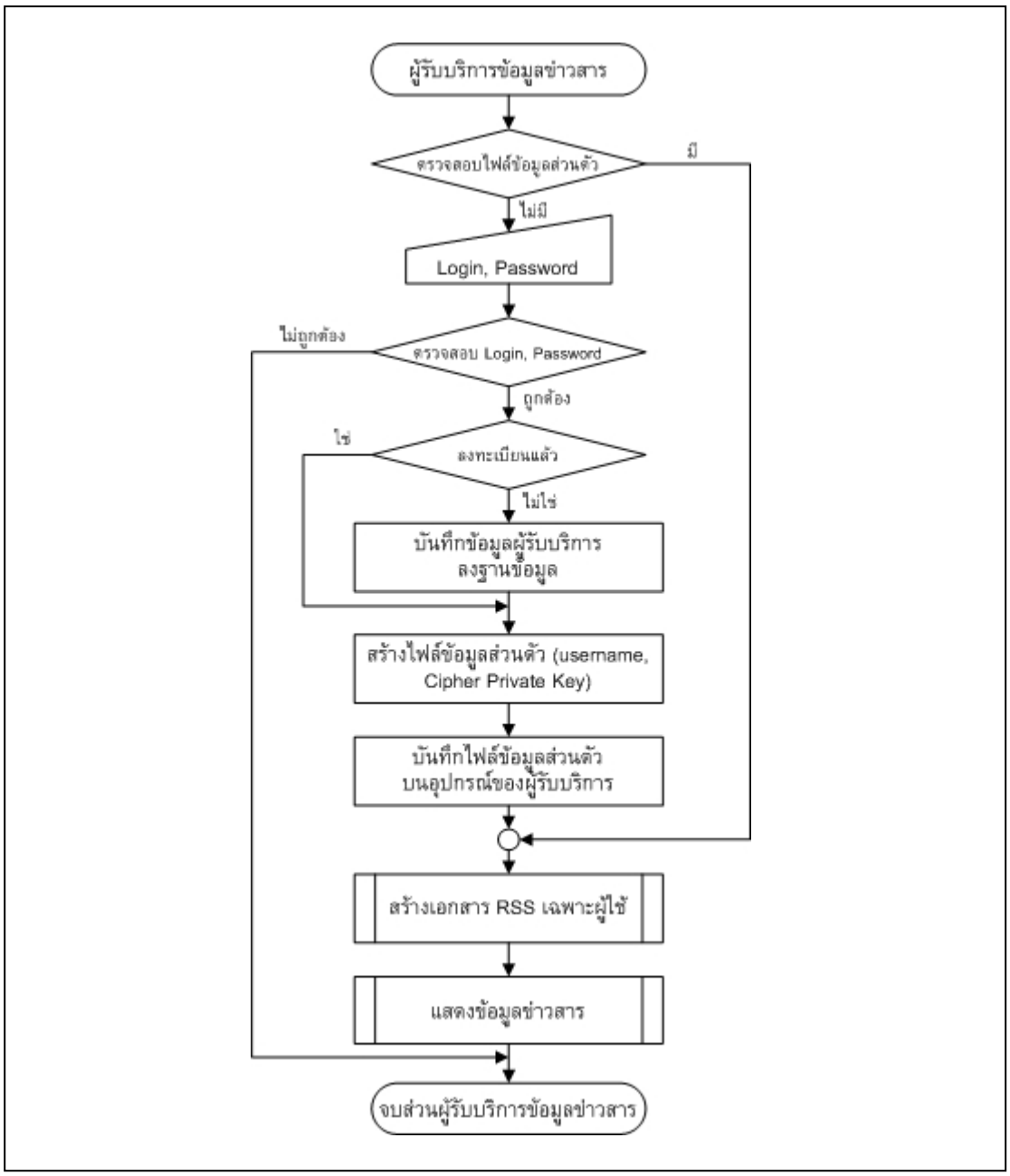
ข้อมูลข่าวสารที่บันทึกไว้จะถูกนำไปเผยแพร่ในรูปแบบเอกสาร RSS บนเว็บไซต์ของผู้เผยแพร่ โดยกระบวนการสร้างเอกสาร RSS แสดงดังภาพประกอบ 4.10



ภาพประกอบ 4.10 ผังงานสร้างเอกสาร RSS

4.1.3 ส่วนผู้รับบริการข้อมูลข่าวสาร

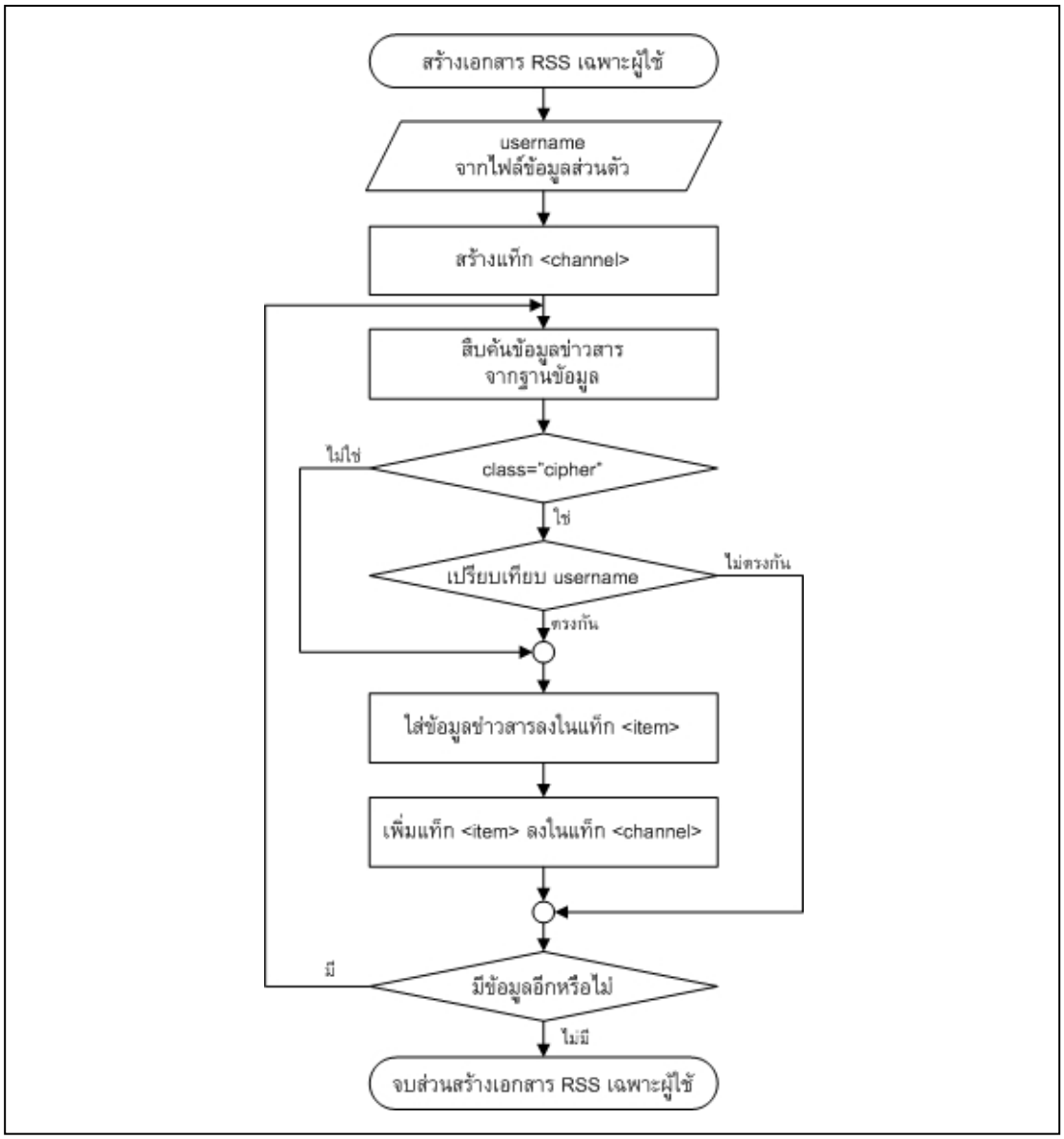
ผู้รับบริการข้อมูลข่าวสารเข้าใช้งานระบบเพื่ออ่านข้อมูลข่าวสารต่าง ๆ ที่ระบบได้รวบรวมไว้ โดยการทำงานแสดงดังภาพประกอบ 4.11



ภาพประกอบ 4.11 ผังงานผู้รับบริการข้อมูลข่าวสาร

1) สร้างเอกสาร RSS เฉพาะผู้ใช้

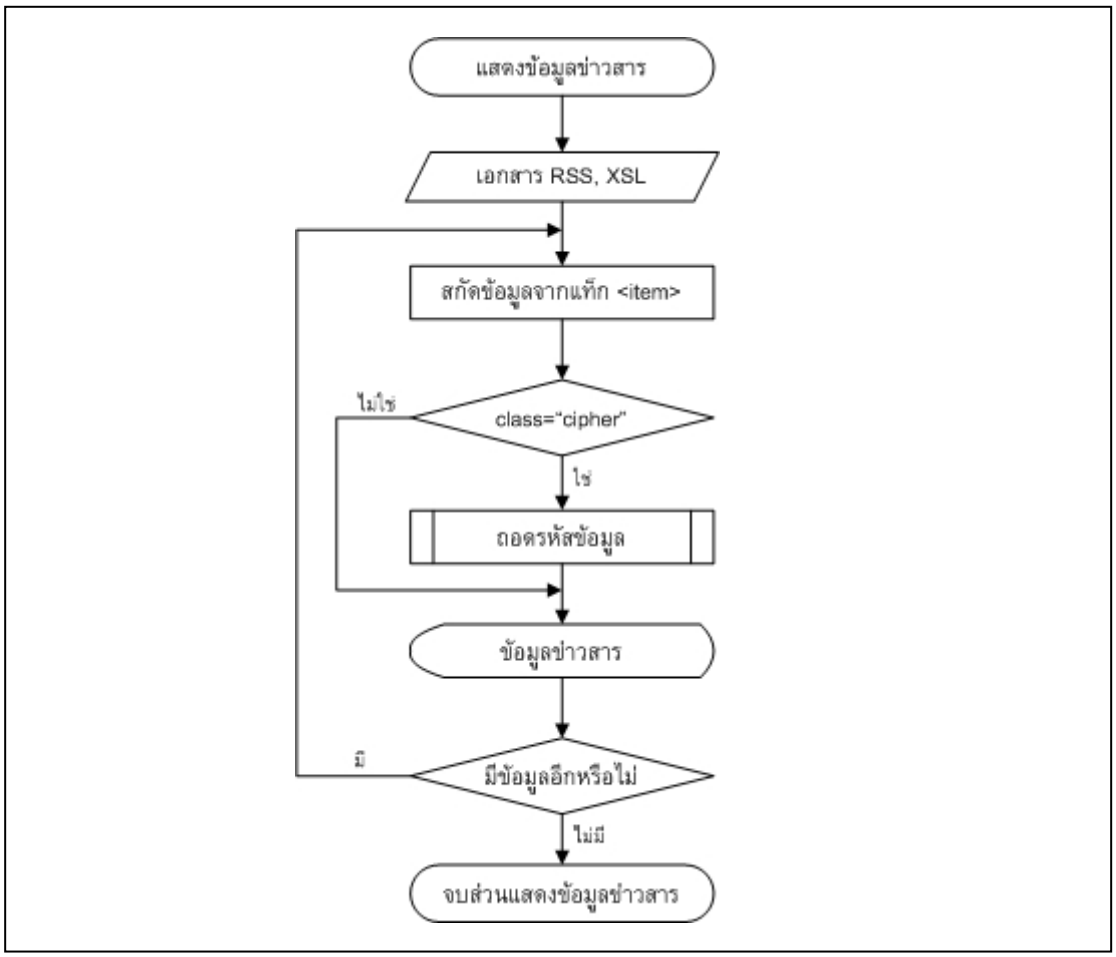
การสร้างเอกสาร RSS เฉพาะผู้ใช้เป็นการสืบค้นข้อมูลข่าวสารที่เกี่ยวข้องกับผู้รับบริการแล้วสร้างเป็นเอกสาร RSS ส่งกลับไปยังผู้รับบริการ โดยการทำงานแสดงดังภาพประกอบ 4.12



ภาพประกอบ 4.12 ผังงานสร้างเอกสาร RSS เฉพาะผู้ใช้

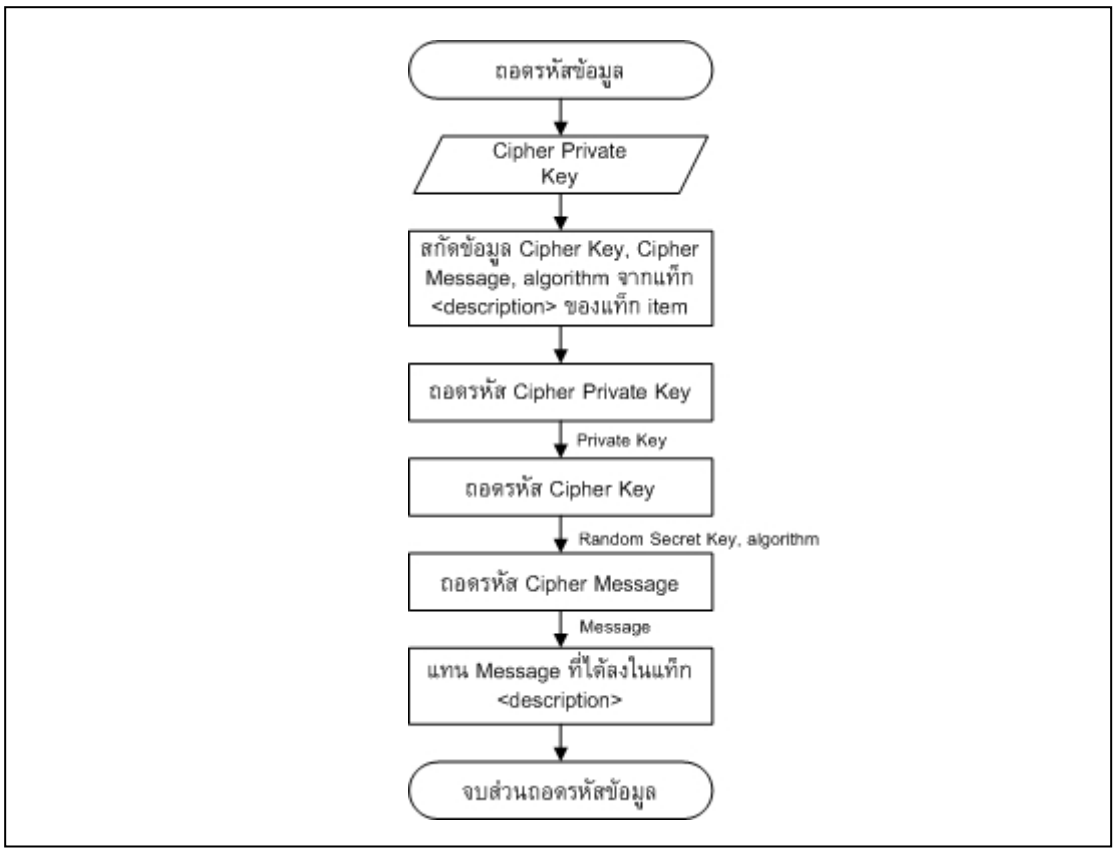
2) แสดงข้อมูลข่าวสาร

การแสดงข้อมูลข่าวสารเป็นการแสดงข้อมูลเอกสาร RSS ที่ได้รับจากระบบบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้รับบริการ



ภาพประกอบ 4.13 ผังงานแสดงข้อมูลข่าวสาร

จากภาพประกอบ 4.13 ข้อมูลข่าวสารที่ถูกเข้ารหัสไว้จะถูกส่งไปยังกระบวนการถอดรหัส โดยผังงานการถอดรหัสแสดงดังภาพประกอบ 4.14

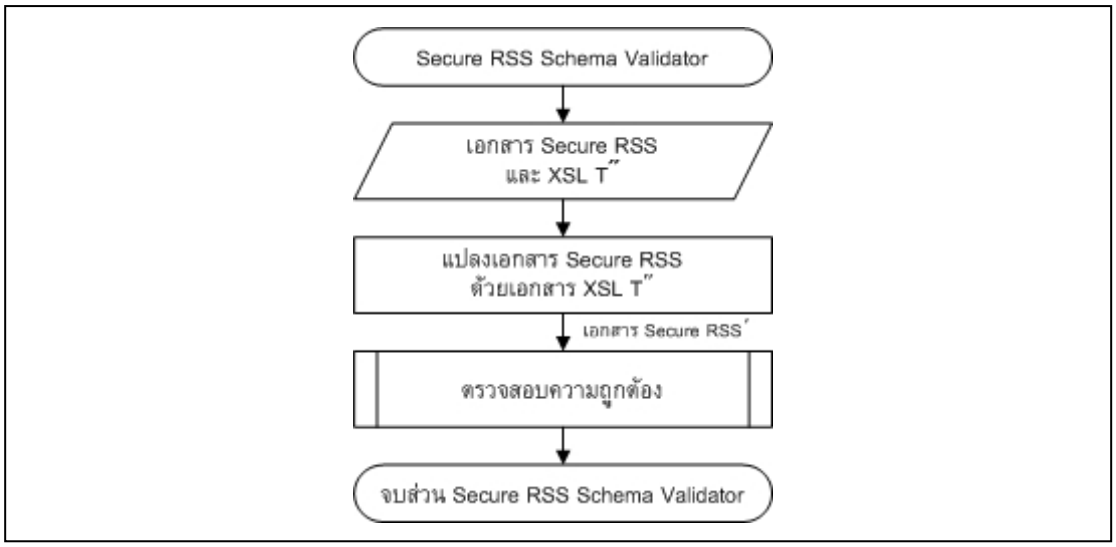


ภาพประกอบ 4.14 ผังงานถอดรหัสข้อมูล

จากภาพประกอบ 4.14 การถอดรหัส Cipher Private Key คือ การถอดรหัสกุญแจส่วนตัวที่ถูกเข้ารหัสไว้ในไฟล์ข้อมูลส่วนตัวบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้รับบริการ เพื่อนำมาใช้ถอดรหัสข้อความไซเฟอร์ของกุญแจถอดรหัส (Cipher Key) ได้เป็น Random Secret Key เพื่อถอดรหัสข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล (Cipher Message) ผลลัพธ์ที่ได้คือ ข้อความต้นฉบับ (Message) ซึ่งจะถูกนำไปแทนที่ลงในแท็ก <description> ก่อนแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้รับบริการ

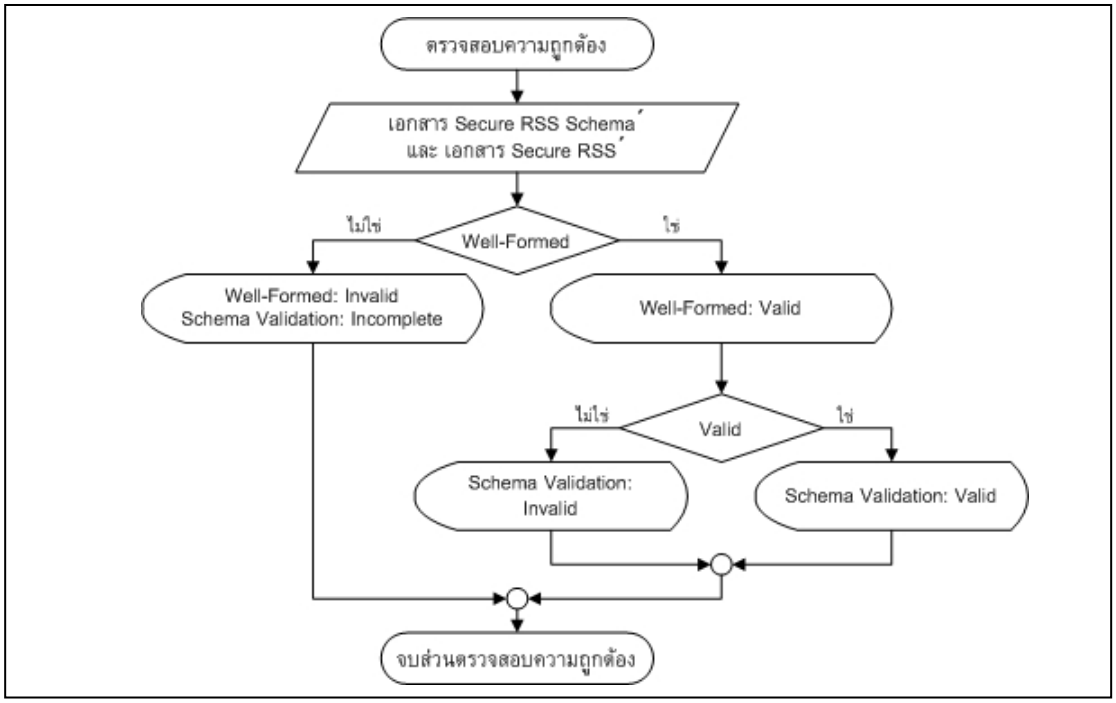
4.1.4 ส่วน Secure RSS Schema Validator

Secure RSS Schema Validator ถูกสร้างขึ้นเพื่อใช้ตรวจสอบความถูกต้องเอกสาร RSS สำหรับเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัยตามกลไกที่ได้ออกแบบไว้ โดยการทำงานของ Secure RSS Schema Validator สามารถอธิบายได้ดังภาพประกอบ 4.15



ภาพประกอบ 4.15 ผังงาน Secure RSS Schema Validator

การตรวจสอบความถูกต้องเอกสาร RSS จะตรวจสอบว่ามีคุณสมบัติ Well-Formed และ Valid หรือไม่ แสดงดังภาพประกอบ 4.16



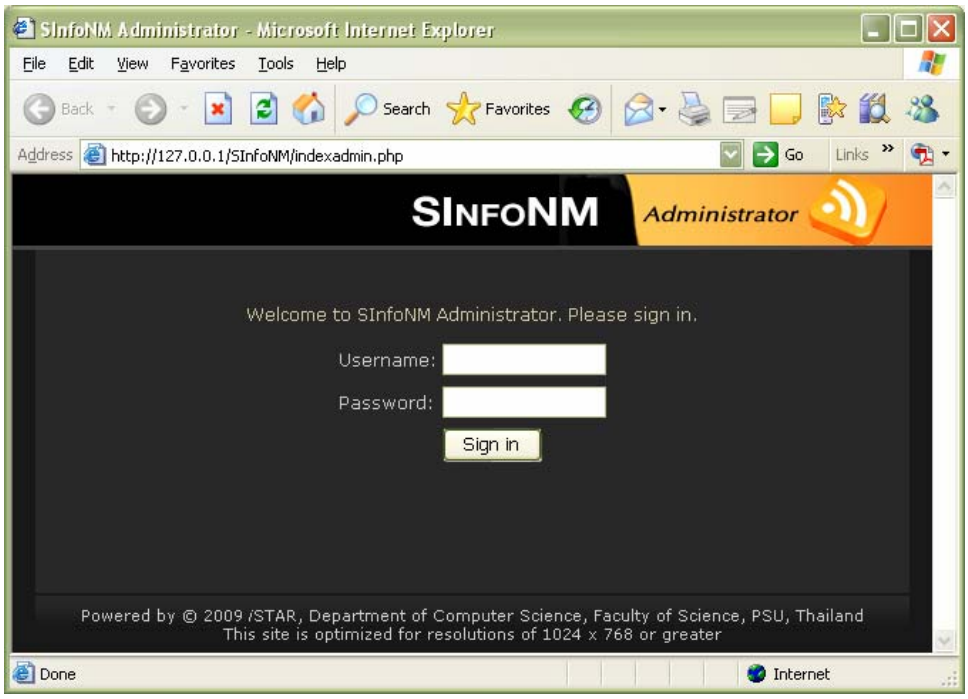
ภาพประกอบ 4.16 ผังงานตรวจสอบความถูกต้อง

4.2 ระบบแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP (SInfoNM)

ระบบ SInfoNM พัฒนาด้วยภาษา PHP และ JavaScript โดยใช้ Apache เป็นโปรแกรมเว็บเซิร์ฟเวอร์ จัดเก็บข้อมูลลงในฐานข้อมูล MySQL ซึ่งสามารถอธิบายการทำงานส่วนต่าง ๆ ได้ดังนี้

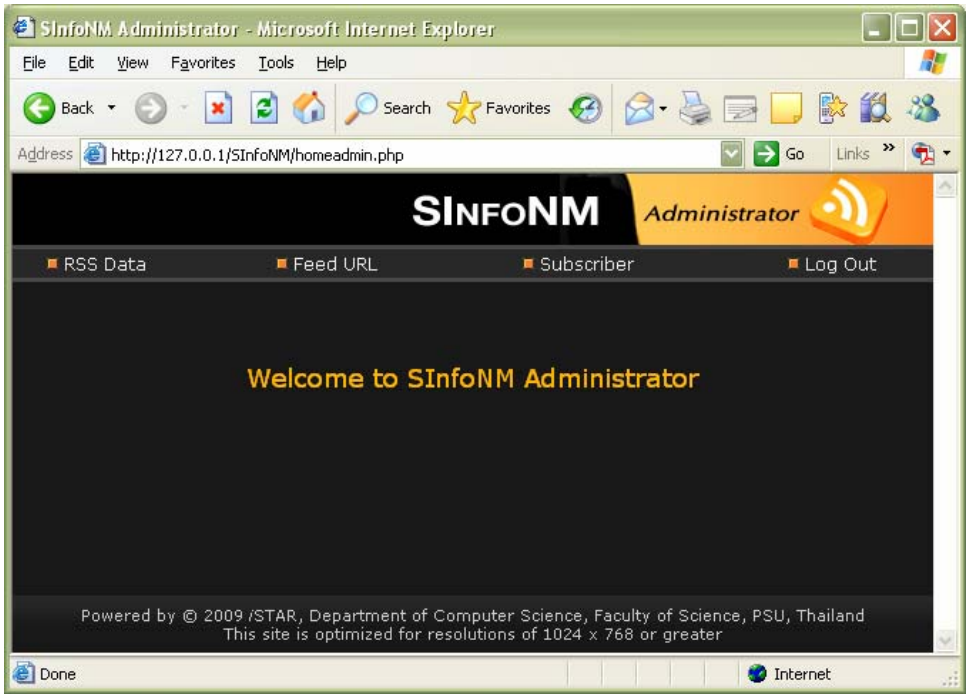
4.2.1 ส่วนผู้ดูแลระบบ

เมื่อเข้าสู่หน้าจอของผู้ดูแลระบบจะปรากฏหน้าต่างเพื่อยืนยันตัวตน ดังภาพประกอบ 4.17



ภาพประกอบ 4.17 หน้าต่างยืนยันตัวตนของผู้ดูแลระบบ

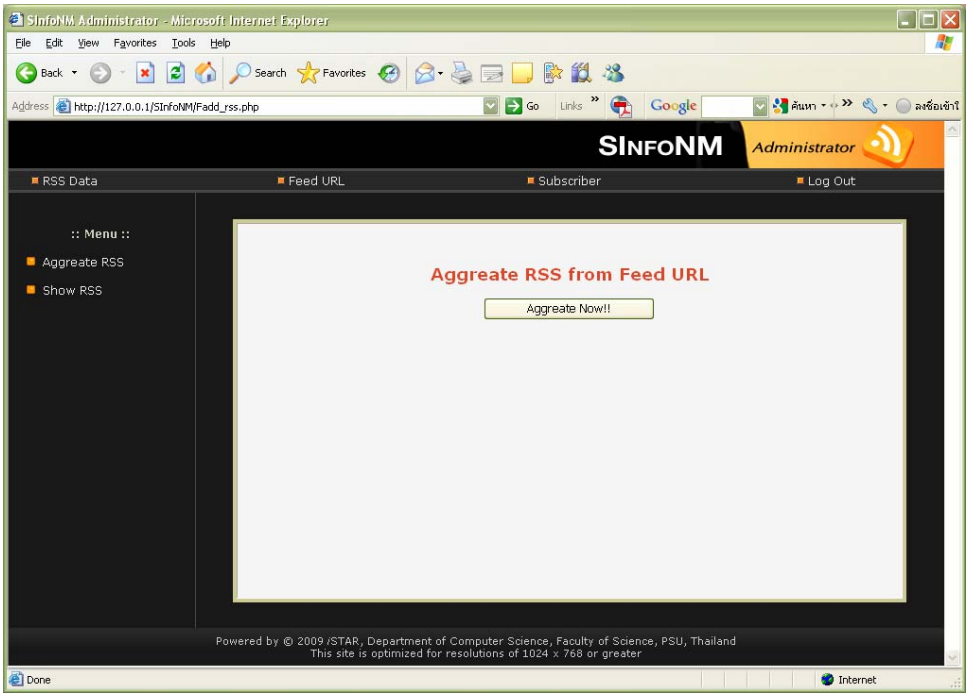
เมื่อตรวจสอบตัวตนของผู้ดูแลระบบเรียบร้อยแล้วจะปรากฏหน้าต่างเพื่อจัดการส่วนต่าง ๆ ของระบบ ซึ่งประกอบด้วย 4 เมนู คือ RSS Data, Feed URL, Subscriber และ Log out ดังภาพประกอบ 4.18



ภาพประกอบ 4.18 หน้าต่างต้อนรับเมื่อผู้ดูแลระบบยืนยันตัวตนเรียบร้อยแล้ว

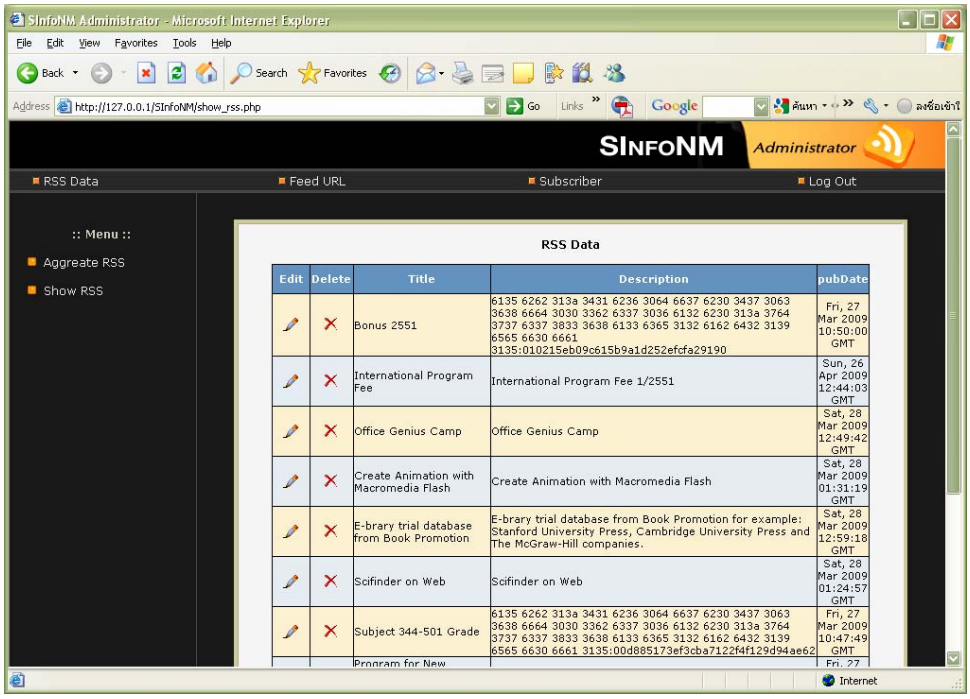
1) เมนู RSS Data

เมนู RSS Data ใช้สำหรับรวบรวมเอกสาร RSS ตามที่อยู่ที่ได้ระบุไว้ใน Feed URL ดังแสดงในภาพประกอบ 4.19



ภาพประกอบ 4.19 หน้าต่างสำหรับรวบรวมเอกสาร RSS โดยผู้ดูแลระบบ

จากภาพประกอบ 4.19 ผู้ดูแลระบบสามารถรวบรวมเอกสาร RSS (อัพเดทข้อมูล RSS) ด้วยการคลิกปุ่ม “Aggregate Now!!” ซึ่งปกติกำหนดให้อัพเดทข้อมูลอัตโนมัติทุกชั่วโมง เนื่องจากหลายเว็บไซต์มีการอัพเดทข้อมูล RSS ทุกช่วงเวลาดังกล่าว (ชารวีย์ แสงขำ, 2552) นอกจากนี้เมนู RSS Data ยังใช้เพื่อแสดง แก้ไข และลบข้อมูล RSS จากฐานข้อมูล ดังแสดงในภาพประกอบ 4.20

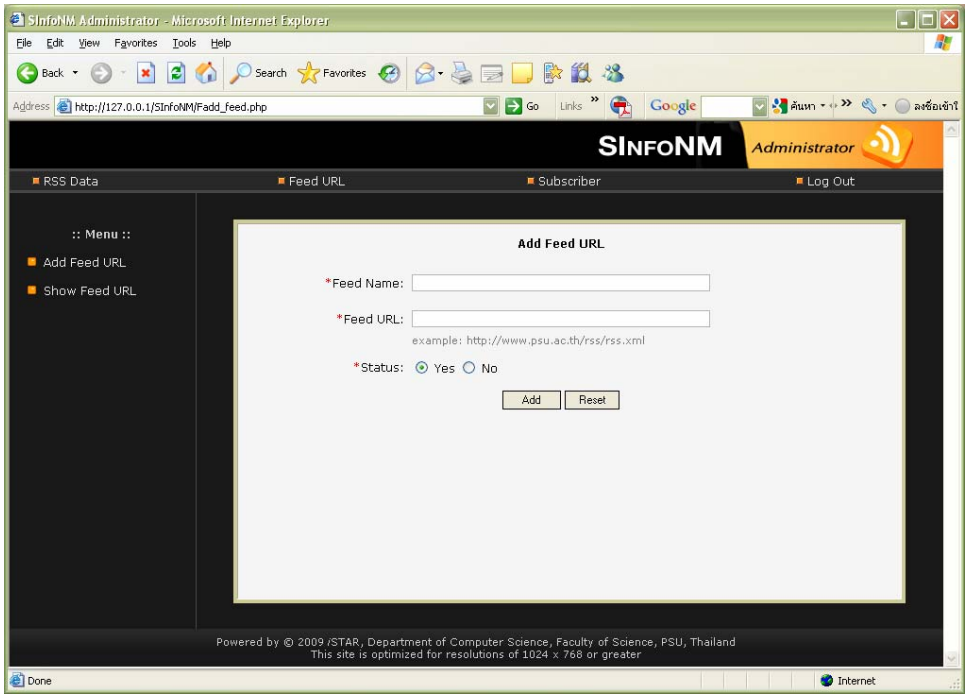


ภาพประกอบ 4.20 หน้าต่างแสดง แก้ไข และลบข้อมูล RSS จากฐานข้อมูล

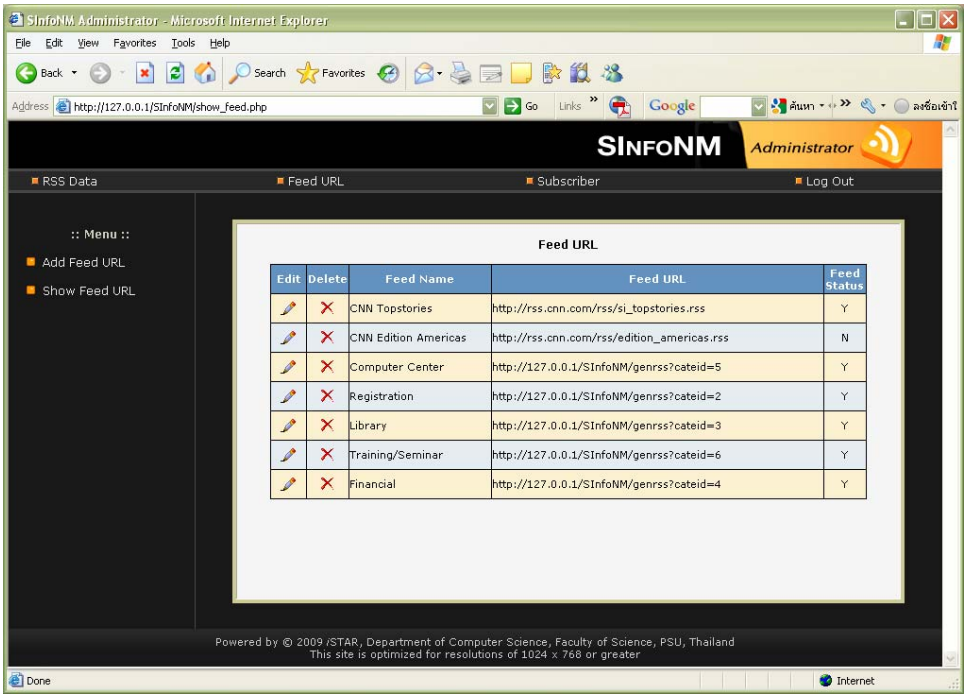
จากภาพประกอบ 4.20 เมื่อคลิกไอคอน และ จะปรากฏหน้าต่างสำหรับแก้ไขและลบข้อมูล RSS จากฐานข้อมูลตามลำดับ

2) เมนู Feed URL

เมนู Feed URL ใช้สำหรับเพิ่ม URL ที่อยู่เอกสาร RSS ดังแสดงในภาพประกอบ 4.21 และยังใช้เพื่อแสดง แก้ไข และลบข้อมูล Feed URL แสดงดังภาพประกอบ 4.22



ภาพประกอบ 4.21 หน้าต่างสำหรับเพิ่มข้อมูล Feed URL

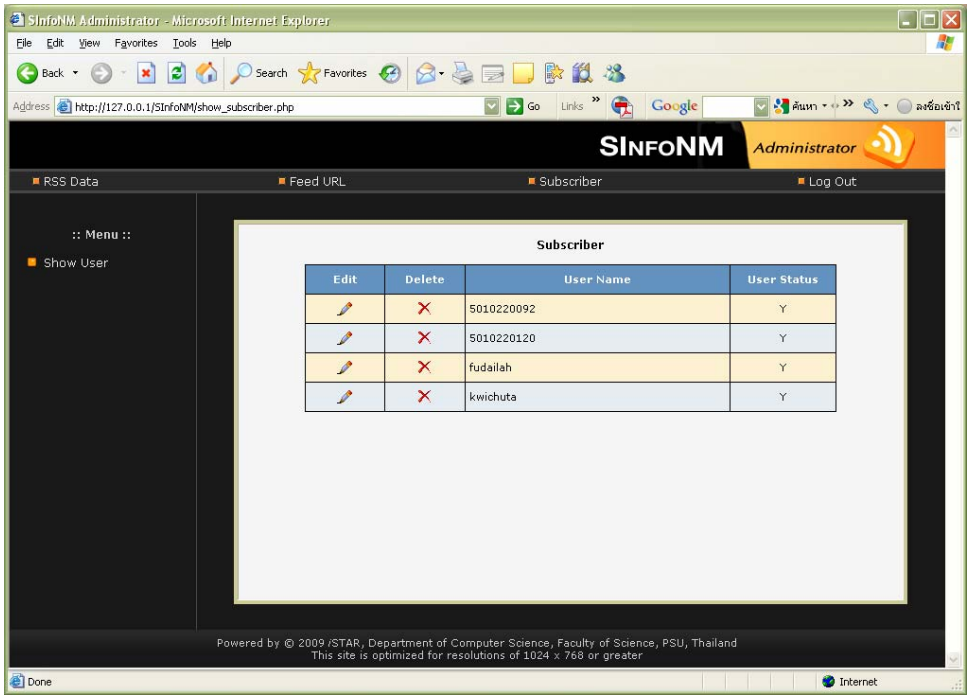


ภาพประกอบ 4.22 หน้าต่างสำหรับแสดง แก้ไข และลบข้อมูล Feed URL

จากภาพประกอบ 4.22 เมื่อคลิกไอคอน  และ  จะปรากฏหน้าต่างสำหรับแก้ไขและลบข้อมูล Feed URL จากฐานข้อมูลตามลำดับ

3) เมนู Subscriber

เมนู Subscriber ใช้สำหรับแก้ไขและลบข้อมูลผู้รับบริการ ดังแสดงในภาพประกอบ 4.23



ภาพประกอบ 4.23 หน้าต่างสำหรับแก้ไขและลบข้อมูลผู้รับบริการ

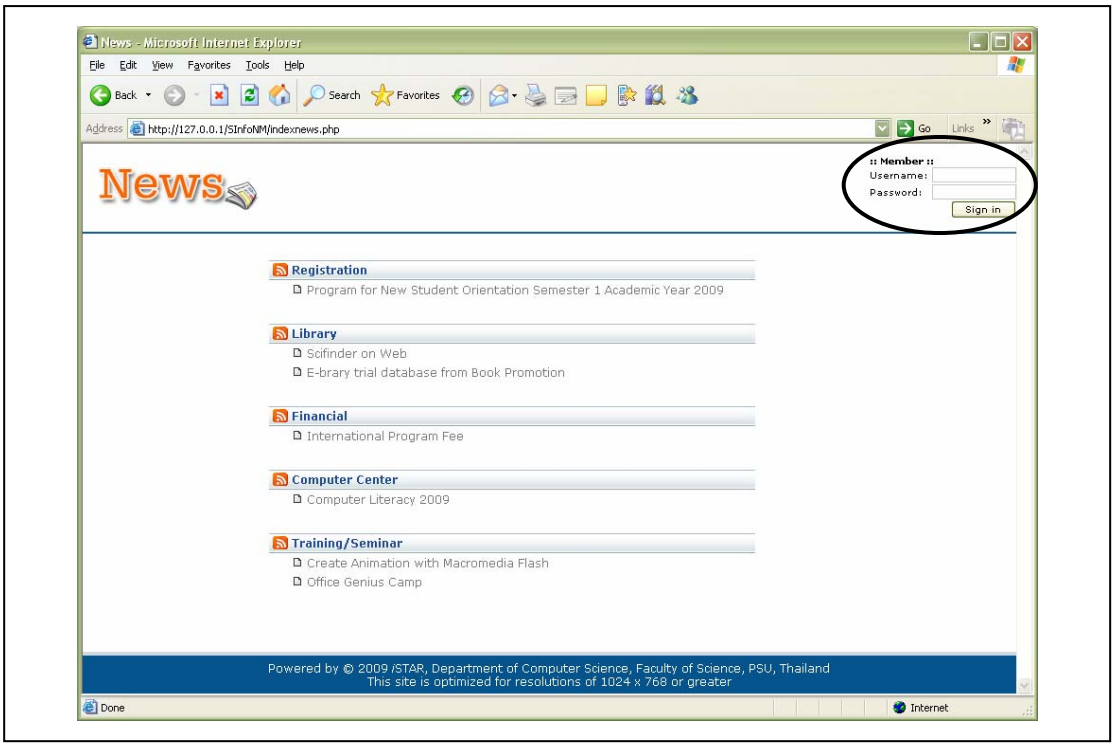
จากภาพประกอบ 4.23 เมื่อคลิกไอคอน  และ  จะปรากฏหน้าต่างสำหรับแก้ไขและลบข้อมูลผู้บริการจากฐานข้อมูลตามลำดับ

4) เมนู Log Out

เมนู Log Out ใช้เมื่อต้องการออกจากส่วนผู้ดูแลระบบและกลับไปยังหน้าแรกของผู้ดูแลระบบ

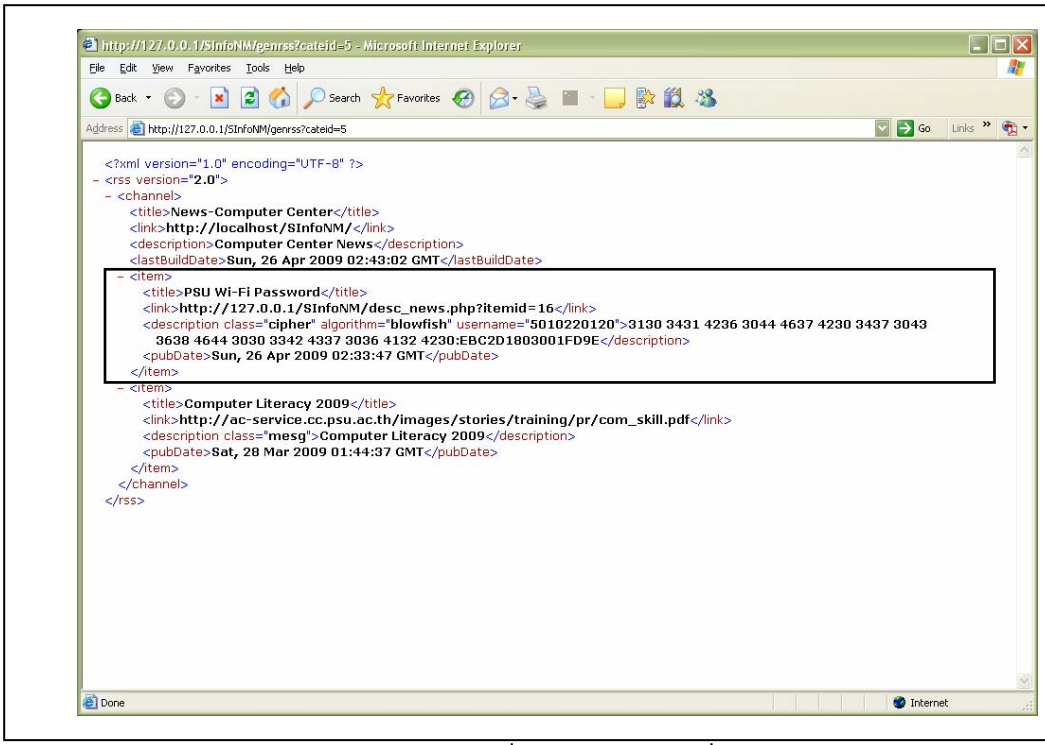
4.2.2 ส่วนผู้เผยแพร่ข้อมูลข่าวสาร

ส่วนผู้เผยแพร่ข้อมูลข่าวสารเป็นการจำลองเว็บไซต์ของหน่วยงานที่มีการประกาศข่าวในรูปแบบเอกสาร RSS แสดงดังภาพประกอบ 4.24 ซึ่งประกอบด้วยข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคลตามนิยามโครงสร้างเอกสาร RSS ที่ได้กำหนดไว้



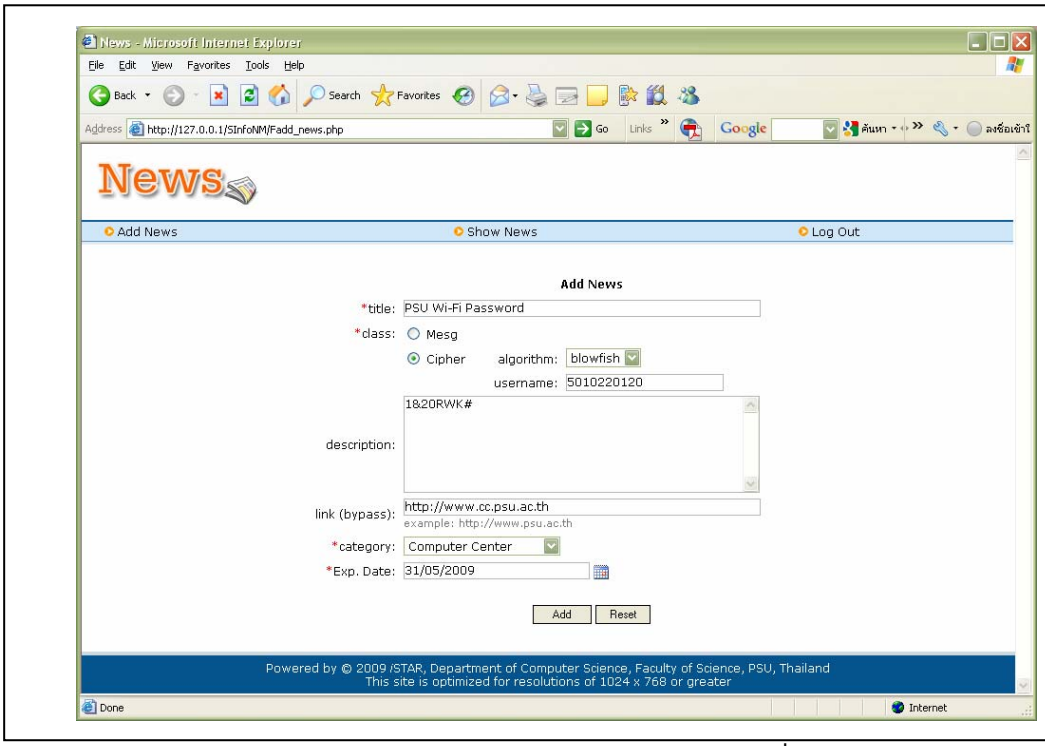
ภาพประกอบ 4.24 หน้าต่างการประกาศข่าวในรูปแบบเอกสาร RSS

จากภาพประกอบ 4.24 ข้อมูลข่าวสารถูกเผยแพร่โดยจัดกลุ่มตามหน่วยงานย่อยภายในองค์กร ซึ่งผู้เผยแพร่ข้อมูลข่าวสารต้องตรวจสอบตัวตนด้วยการระบุ Username และ Password ที่ช่องมุมขวาบนก่อนการประกาศข่าว โดยข้อมูลข่าวสารในรูปแบบเอกสาร RSS ดังภาพประกอบ 4.25 จะถูกแสดงเมื่อคลิกที่ไอคอน



ภาพประกอบ 4.25 ตัวอย่างเอกสาร RSS ที่มีข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล

เมื่อผู้เผยแพร่ตรวจสอบตัวตนเพื่อประกาศข่าวเรียบร้อยแล้วจะปรากฏหน้าต่าง ดังภาพประกอบ 4.26 เพื่อบันทึกข้อมูลข่าวสารที่ต้องการเผยแพร่



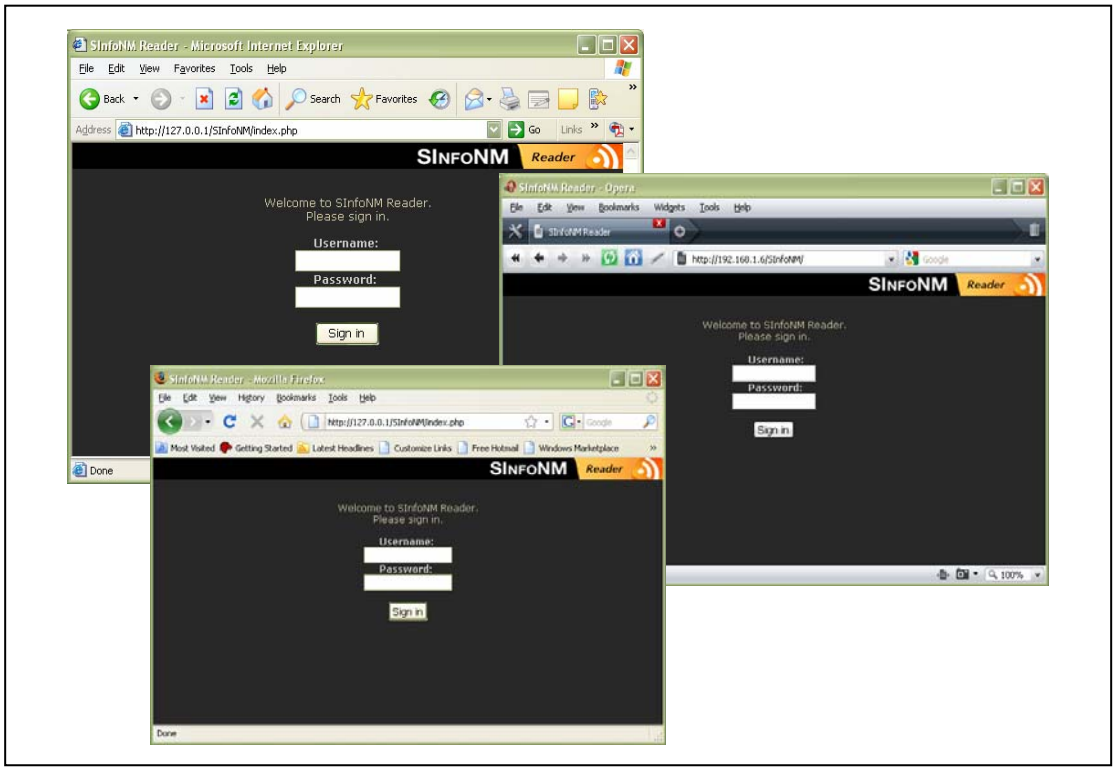
ภาพประกอบ 4.26 หน้าต่างสำหรับบันทึกข้อมูลข่าวสารที่ต้องการเผยแพร่

จากภาพประกอบ 4.26 หากข้อมูลข่าวสารที่ต้องการเผยแพร่เป็นข้อมูลข่าวสารทั่วไปกำหนด class เท่ากับ mesg หากเป็นข้อมูลข่าวสารส่วนบุคคลกำหนด class เท่ากับ cipher พร้อมทั้งเลือก algorithm และระบุ username ของผู้รับข้อมูลข่าวสาร เมื่อกรอกข้อมูลเรียบร้อยแล้วคลิกปุ่ม Add ระบบจะเข้ารหัสข้อมูลข่าวสารส่วนบุคคลที่ระบุไว้ในช่อง description ด้วย algorithm ที่กำหนดอัตโนมัติก่อนบันทึกลงฐานข้อมูล

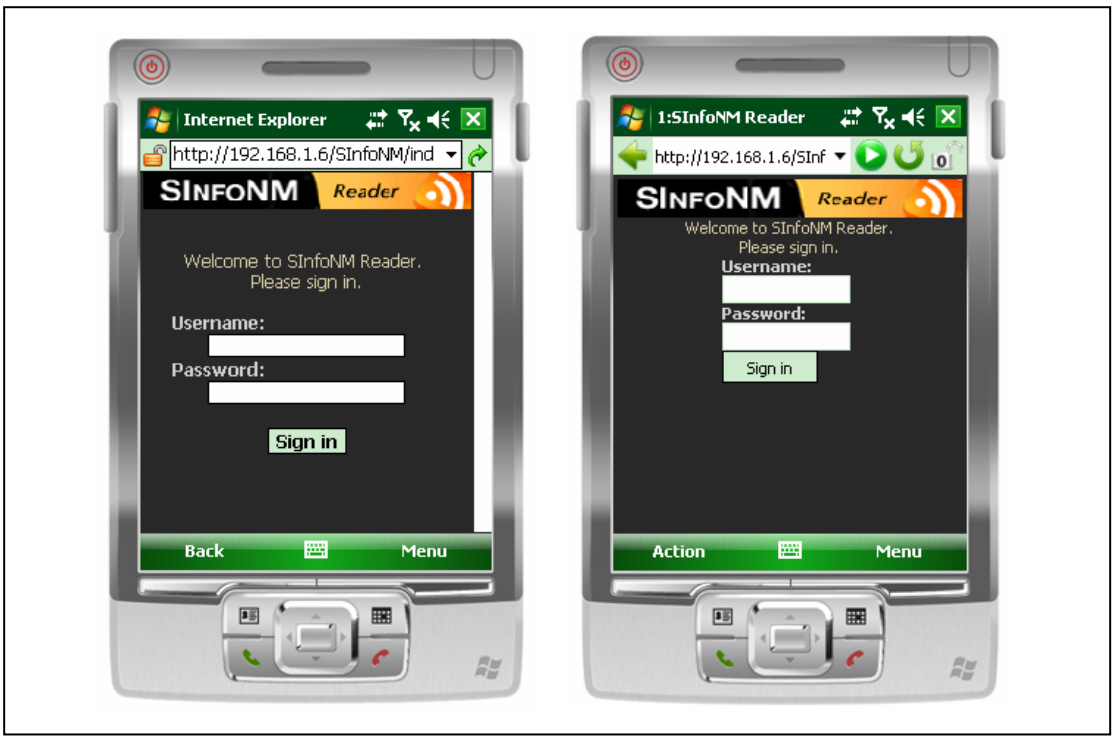
4.2.3 ส่วนผู้รับบริการข้อมูลข่าวสาร

ส่วนผู้รับบริการข้อมูลข่าวสารถูกออกแบบให้สามารถทำงานได้ทั้งบนเครื่องคอมพิวเตอร์ทั่วไป และบนอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP โดยการทำงานบนเครื่องคอมพิวเตอร์ทั่วไปใช้ Internet Explorer, Mozilla FireFox และ Opera เป็นเบราว์เซอร์สำหรับแสดงผล ส่วนการทำงานบนอุปกรณ์สื่อสารเคลื่อนที่ซึ่งในที่นี้ได้แก่ พีดีเอและสมาร์ทโฟน (Smart Phone) เป็นต้น ใช้ Windows Mobile 6.1 Emulator (Microsoft, 2008) เป็นระบบปฏิบัติการบนอุปกรณ์สื่อสารเคลื่อนที่สำหรับทดสอบการทำงาน โดยใช้ Internet Explorer Mobile และ Opera Mobile เป็นเบราว์เซอร์สำหรับแสดงผล

เมื่อเข้าสู่หน้าจอสำหรับผู้รับบริการข้อมูลข่าวสารจะปรากฏหน้าต่างเพื่อตรวจสอบตัวตน แสดงดังภาพประกอบ 4.27 และภาพประกอบ 4.28



ภาพประกอบ 4.27 หน้าต่างยืนยันตัวตนผู้รับบริการข้อมูลข่าวสาร
สำหรับเครื่องคอมพิวเตอร์ทั่วไป

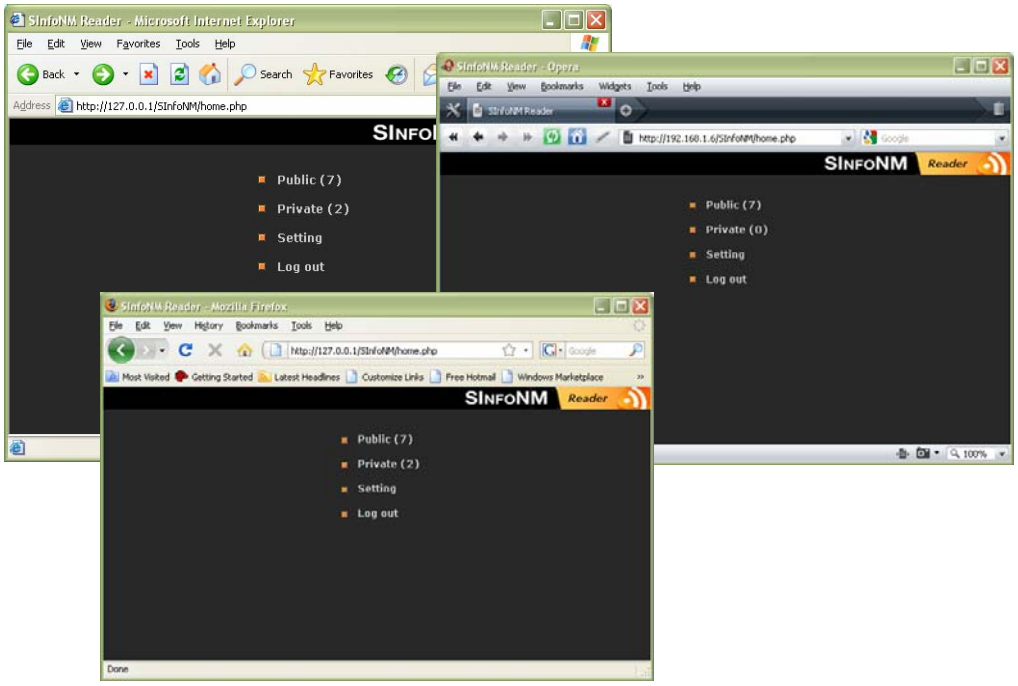


ภาพประกอบ 4.28 หน้าต่างยืนยันตัวตนผู้รับบริการข้อมูลข่าวสาร
สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

จากภาพประกอบ 4.28 ภาพด้านซ้ายแสดงผลด้วย Internet Explorer Mobile และภาพด้านขวาแสดงผลด้วย Opera Mobile

เมื่อยืนยันตัวตนเรียบร้อยแล้ว ผู้รับบริการที่ยังไม่ได้ลงทะเบียนรับข้อมูลข่าวสารจะปรากฏหน้าต่างเพื่อลงทะเบียนรับข้อมูลข่าวสารก่อนการใช้งาน เมื่อคลิกปุ่ม “Please register before working…Click Here” ระบบจะทำการบันทึกข้อมูลผู้รับบริการลงฐานข้อมูล และสร้างไฟล์คุกกี้ (Cookie) เพื่อเก็บข้อมูล Username และกุญแจถอดรหัสไว้บนเครื่องคอมพิวเตอร์ของผู้รับบริการสำหรับใช้ยืนยันตัวตน และถอดรหัสข้อมูลในการใช้งานครั้งต่อไป

สำหรับผู้รับบริการที่ลงทะเบียนไว้แล้วจะปรากฏเมนูเพื่อเลือกการทำงาน ประกอบด้วย 4 เมนู คือ Public, Private, Setting และ Log out ดังภาพประกอบ 4.29 และภาพประกอบ 4.30 โดยสามารถอธิบายการทำงานของแต่ละเมนูได้ดังนี้

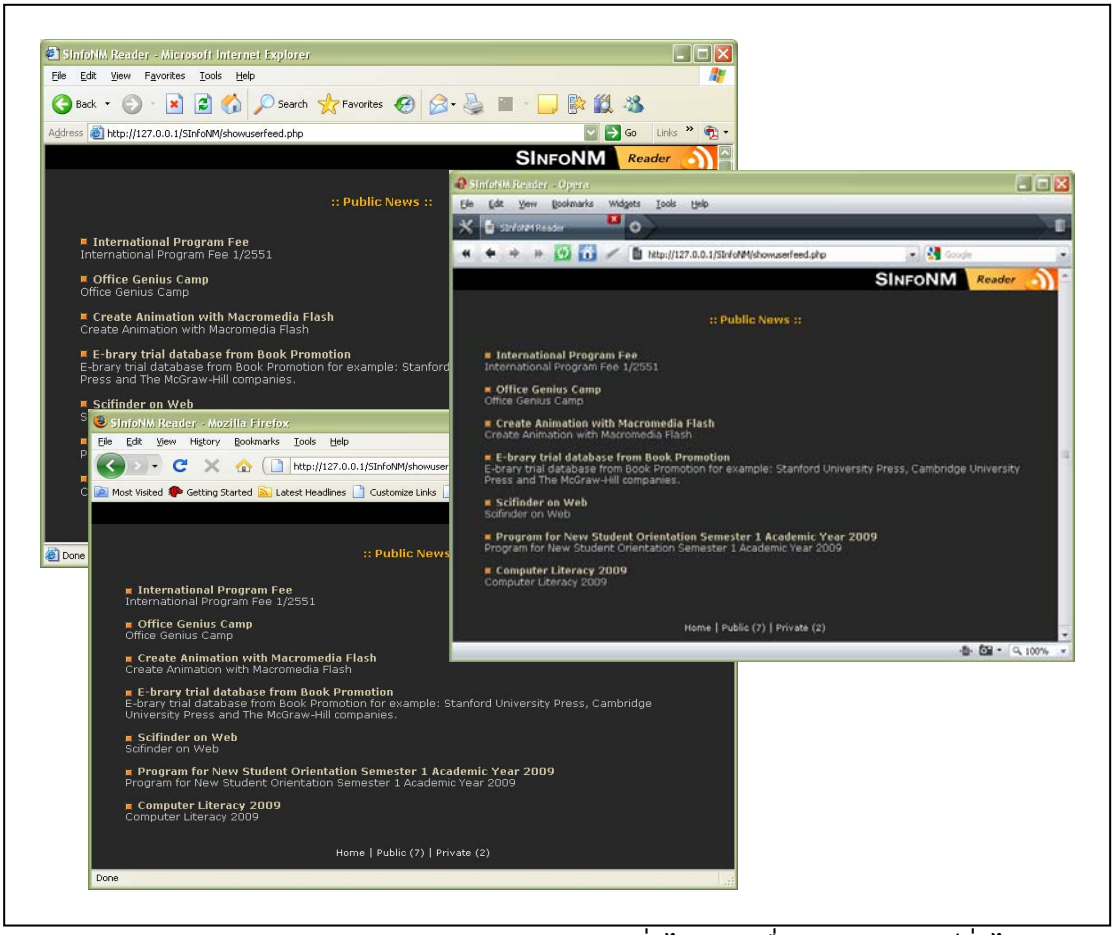


ภาพประกอบ 4.29 หน้าต่างแสดงเมนูการทำงานสำหรับผู้รับบริการข้อมูลข่าวสาร
บนเครื่องคอมพิวเตอร์ทั่วไป

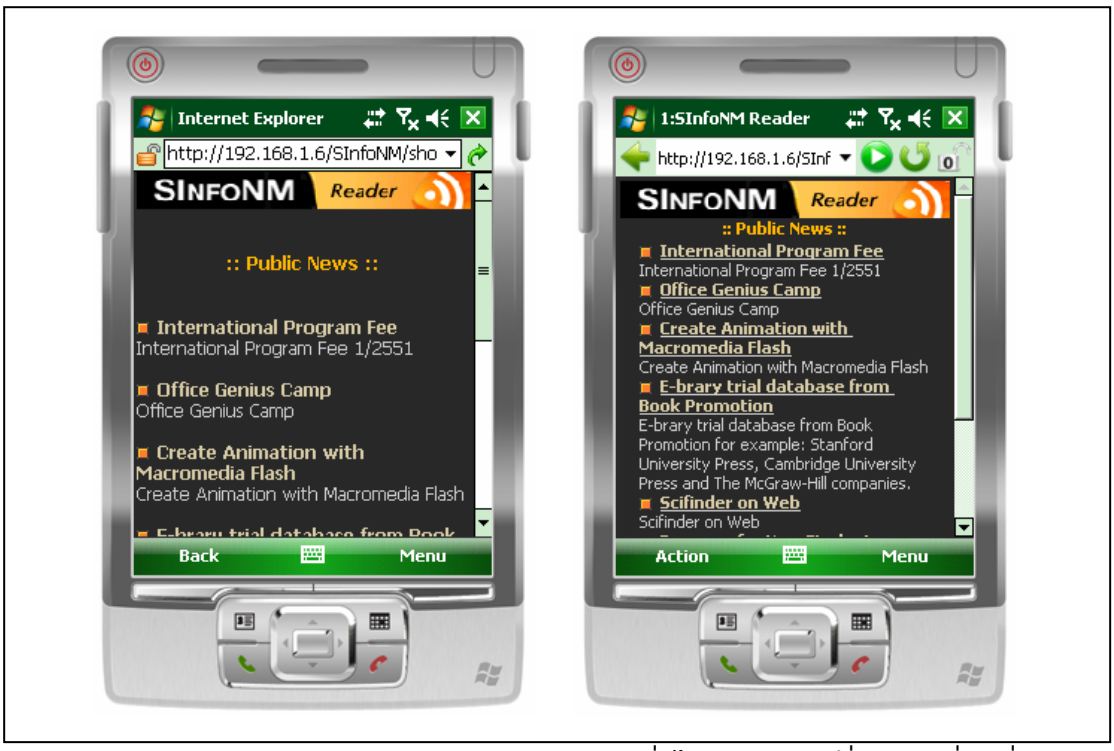
ภาพประกอบ 4.30 หน้าต่างแสดงเมนูการทำงานสำหรับผู้รับบริการข้อมูลข่าวสาร
บนอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

1) เมนู Public

เมนู Public ใช้สำหรับแสดงข้อมูลข่าวสารทั่วไป ดังภาพประกอบ 4.31 และภาพประกอบ 4.32



ภาพประกอบ 4.31 หน้าต่างแสดงข้อมูลข่าวสารทั่วไปบนเครื่องคอมพิวเตอร์ทั่วไป



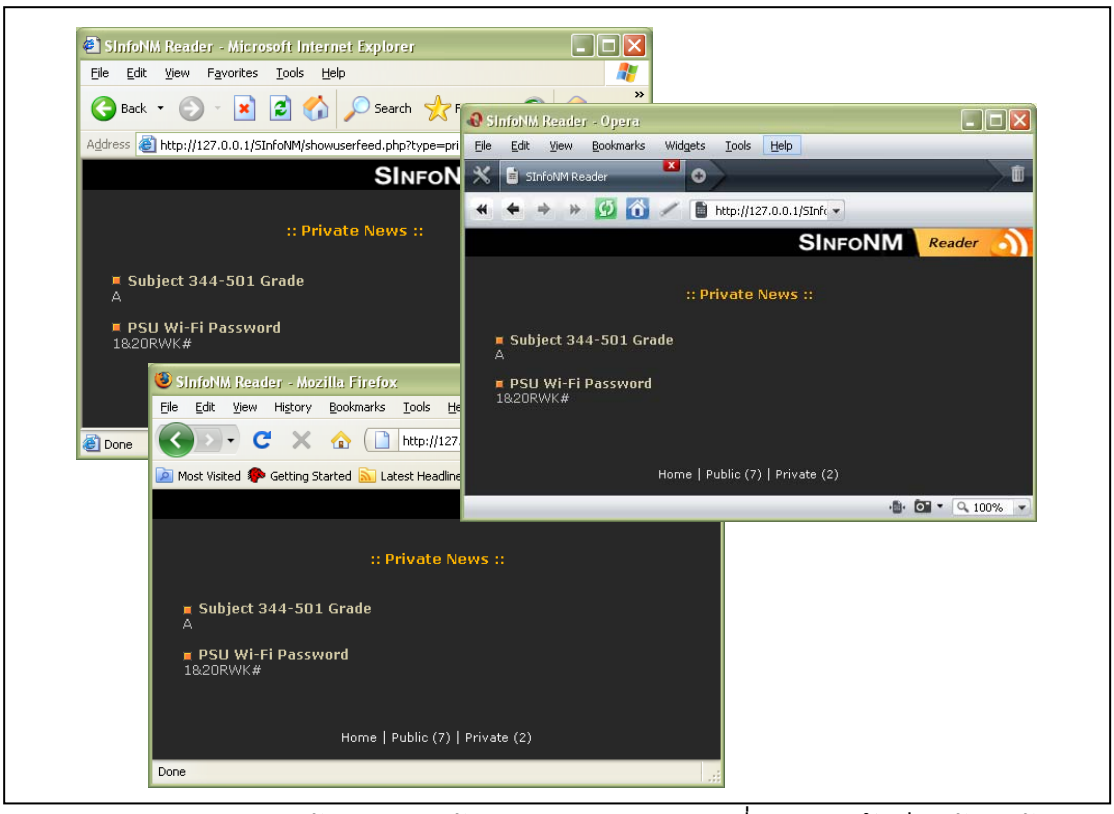
ภาพประกอบ 4.32 หน้าต่างแสดงข้อมูลข่าวสารทั่วไปบนอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

2) เมนู Private

เมนู Private ใช้สำหรับแสดงข้อมูลข่าวสารส่วนบุคคล โดยใช้เอกสาร XSL ดังภาพประกอบ 4.33 สำหรับจัดรูปแบบการแสดงผล และสืบค้นข้อมูลที่ถูกเข้ารหัสไว้ จากนั้นส่งข้อมูลที่สืบค้นได้ไปยังฟังก์ชันถอดรหัสที่พัฒนาด้วยภาษา JavaScript เพื่อถอดรหัสบนเครื่องคอมพิวเตอร์ของผู้รับบริการ ผลลัพธ์ที่ได้จะถูกแสดงบนเครื่องคอมพิวเตอร์ของผู้รับบริการ ดังภาพประกอบ 4.34 และภาพประกอบ 4.35 นอกจากนี้เอกสาร XSL ยังถูกใช้เพื่อแสดงผลข้อมูลข่าวสารทั่วไปอีกด้วย

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <table border="0" align="center" width="90%"><tr><td>
        <xsl:for-each select="rss/channel/item">
            <img src="img/but5.gif" border="0"/>
            <xsl:element name="a">
                <xsl:attribute name="href">
                    <xsl:value-of select="link" />
                </xsl:attribute>
                <b><font color="#D3CBA7"><xsl:value-of select="title" /></font></b><br/>
            </xsl:element>
            <xsl:choose>
                <xsl:when test="description/@class='cipher'">
                    <script language="JavaScript1.2">
                        message=Decrypt("<xsl:value-of select="description" />","<xsl:value-of
                        select="description/@algorithm" />");
                        document.write(message);
                    </script>
                </xsl:when>
                <xsl:when test="description/@class='mesg'">
                        <xsl:value-of select="description"/>
                </xsl:when>
            </xsl:choose>
            <br/><br/>
        </xsl:for-each>
        </td></tr></table>
    </xsl:template>
</xsl:stylesheet>
```

ภาพประกอบ 4.33 เอกสาร XSL สำหรับจัดรูปแบบการแสดงผลข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล พร้อมทั้งถอดรหัสข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้



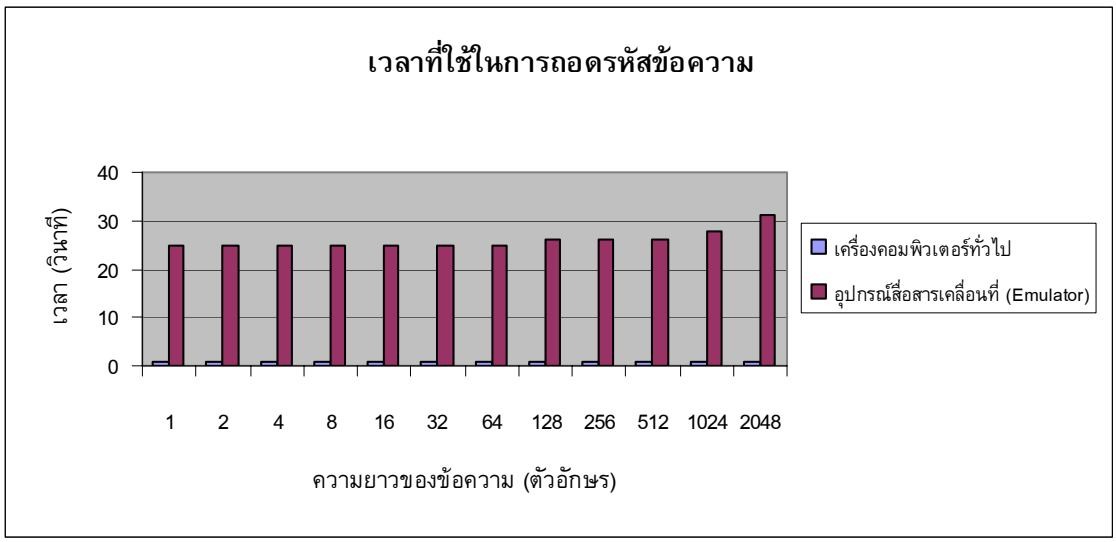
ภาพประกอบ 4.34 หน้าต่างแสดงข้อมูลข่าวสารส่วนบุคคลที่ถูกถอดรหัสเรียบร้อยแล้ว บนเครื่องคอมพิวเตอร์ทั่วไป

ภาพประกอบ 4.35 หน้าต่างแสดงข้อมูลข่าวสารส่วนบุคคลที่ถูกถอดรหัสเรียบร้อยแล้ว บนอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

สำหรับเวลาในการถอดรหัสข้อความได้ทำการทดสอบโดยการสุ่มข้อความที่มีความยาวขนาดต่างกันเพื่อกำหนดเป็นข้อมูลข่าวสารส่วนบุคคล จากนั้นนำข้อความที่ได้มาเข้ารหัสโดยใช้อัลกอริทึม Blowfish ด้วยกุญแจขนาด 128 บิต และใช้อัลกอริทึม RSA สำหรับเข้ารหัสกุญแจที่ใช้เข้ารหัสข้อความด้วยกุญแจขนาด 128 บิต โดยทดสอบการทำงานบนเครื่องคอมพิวเตอร์หน่วยประมวลผลกลางรุ่น Intel® Core™ 2 และหน่วยความจำขนาด 1 GB ผลการทดสอบแสดงให้เห็นว่าเครื่องคอมพิวเตอร์ทั่วไปใช้เวลาถอดรหัสข้อความค่อนข้างน้อย แต่สำหรับอุปกรณ์สื่อสารเคลื่อนที่ใช้เวลาถอดรหัสค่อนข้างมาก นอกจากนี้เมื่อความยาวของข้อความมีขนาดเพิ่มขึ้นมีแนวโน้มที่จะใช้เวลาถอดรหัสข้อความมากขึ้นด้วย ผลการทดสอบแสดงดังตารางที่ 4.1 และภาพประกอบ 4.36

ตารางที่ 4.1 เวลาที่ใช้ในการถอดรหัสข้อความ

ความยาวของข้อความ (ตัวอักษร)	เวลาที่ใช้ (วินาที)	
	เครื่องคอมพิวเตอร์ทั่วไป	อุปกรณ์สื่อสารเคลื่อนที่ (Emulator)
1	0.80	25.00
2	0.80	25.00
4	0.80	25.00
8	0.80	25.00
16	0.80	25.00
32	0.81	25.00
64	0.81	25.00
128	0.81	26.00
256	0.83	26.00
512	0.84	26.00
1024	0.89	28.00
2048	0.98	31.00



ภาพประกอบ 4.36 เวลาที่ใช้ในการถอดรหัสข้อความ

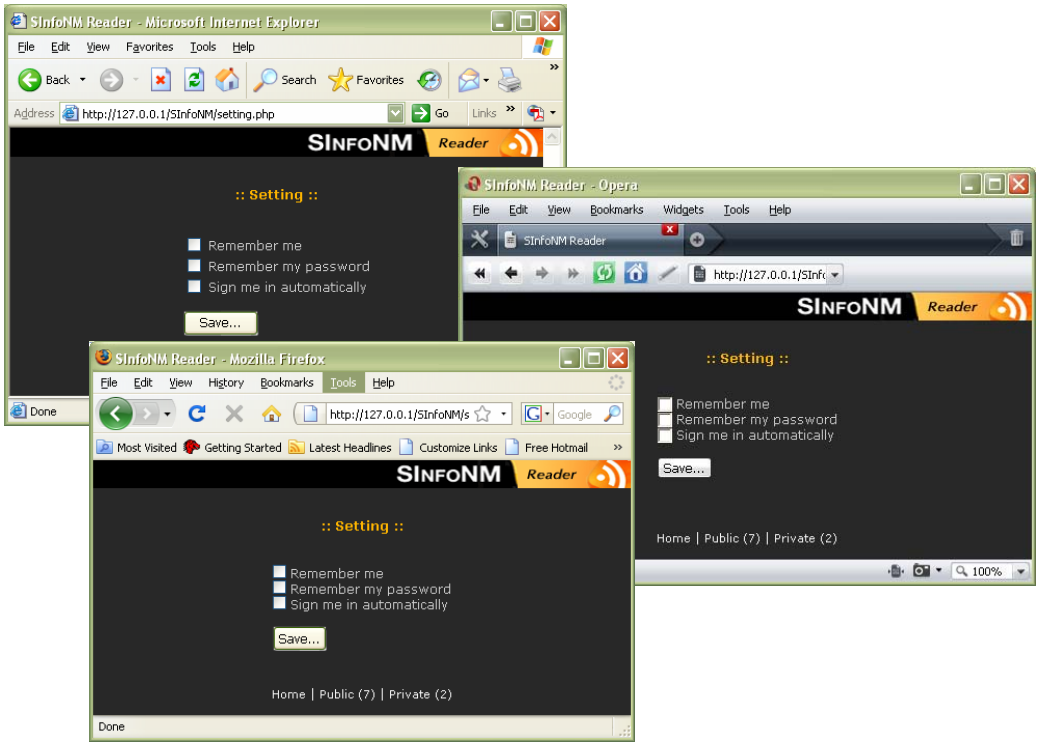
3) เมนู Setting

เมนู Setting ใช้เพื่ออำนวยความสะดวกให้กับผู้รับบริการสำหรับกำหนดค่ายืนยันตัวตนก่อนเข้าใช้งานระบบ ดังแสดงในภาพประกอบ 4.37 และภาพประกอบ 4.38 โดยสามารถกำหนดค่าได้ 3 ลักษณะ ดังนี้

3.1) Remember me คือ ระบบบันทึก Username ของผู้รับบริการไว้ ทำให้การใช้งานครั้งต่อไปผู้รับบริการไม่ต้องกรอก Username

3.2) Remember my password คือ ระบบบันทึก Password ของผู้รับบริการไว้ ทำให้การใช้งานครั้งต่อไปผู้รับบริการไม่ต้องกรอก Username และ Password

3.3) Sign me in automatically คือ ระบบบันทึก Username และ Password ไว้ ทำให้การใช้งานครั้งต่อไปผู้รับบริการจะเข้าสู่ระบบอัตโนมัติเมื่อเปิดใช้งานระบบ



ภาพประกอบ 4.37 หน้าต่างกำหนดค่าสำหรับยืนยันตัวตนบนเครื่องคอมพิวเตอร์ทั่วไป

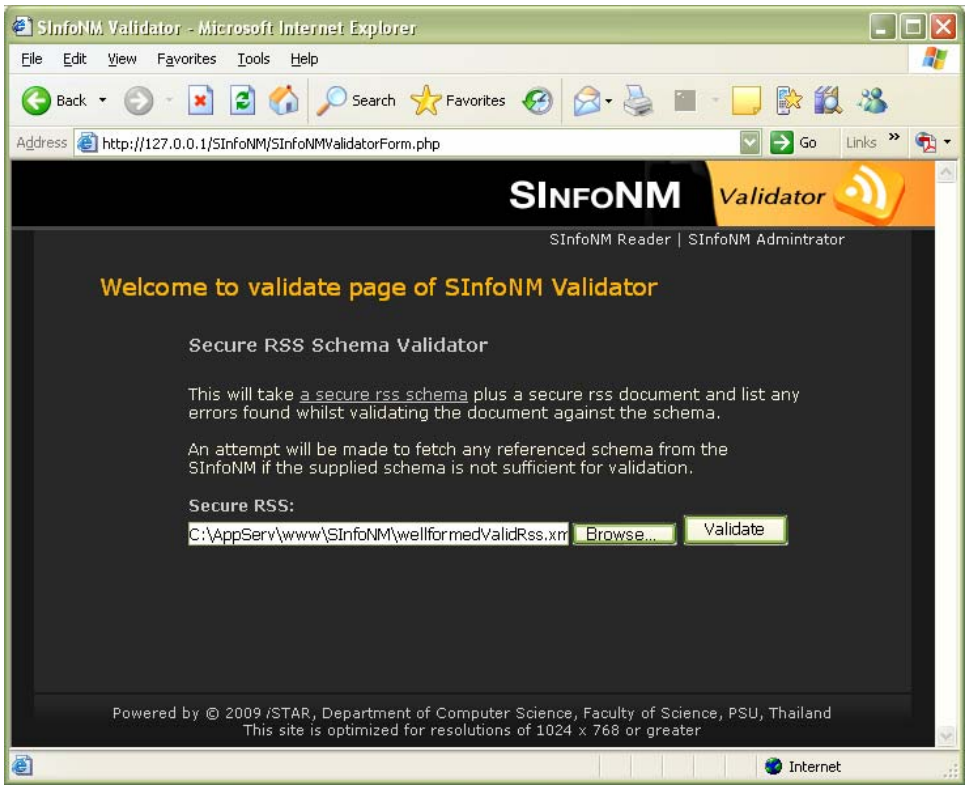
ภาพประกอบ 4.38 หน้าต่างกำหนดค่าสำหรับยืนยันตัวตนบนอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

4) เมนู Log Out

เมนู Log Out ใช้สำหรับออกจากส่วนผู้รับบริการข้อมูลข่าวสาร และกลับไปยังหน้าแรกของผู้รับบริการข้อมูลข่าวสาร

4.2.4 ส่วน Secure RSS Schema Validator

ส่วน Secure RSS Schema Validator ใช้สำหรับตรวจสอบเอกสาร RSS ว่าถูกต้องตามนิยามที่ได้กำหนดไว้หรือไม่ โดยเมื่อเข้าสู่ระบบจะปรากฏหน้าต่าง ดังภาพประกอบ 4.39



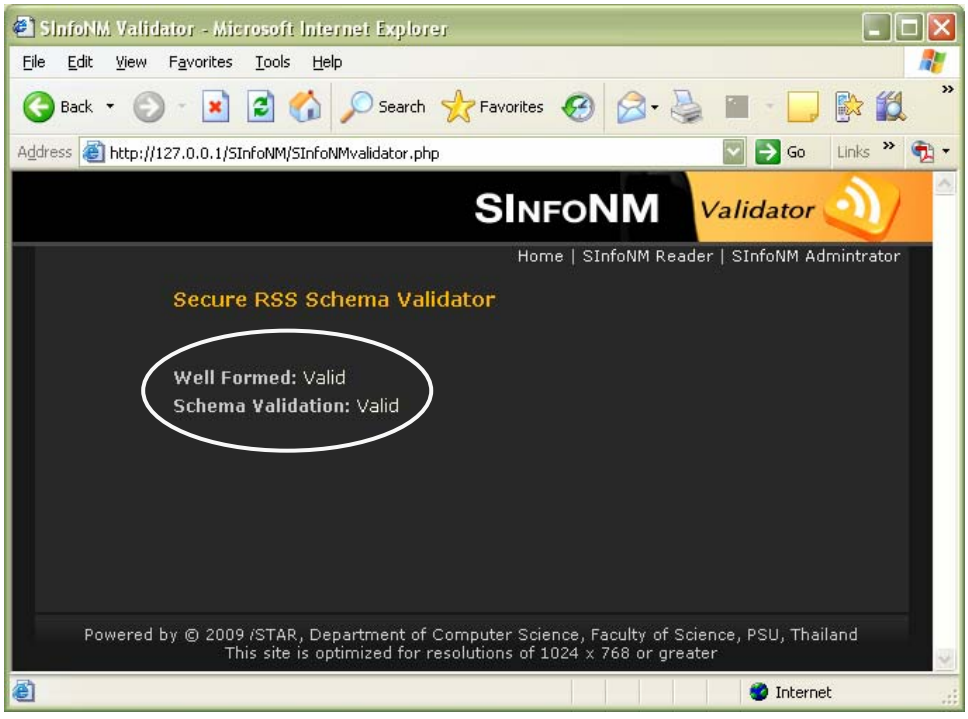
ภาพประกอบ 4.39 หน้าต่างสำหรับเลือกเอกสาร RSS เพื่อตรวจสอบความถูกต้อง

จากภาพประกอบ 4.39 คลิกปุ่ม “Browse” เพื่อเลือกเอกสาร RSS ที่ต้องการตรวจสอบ จากนั้นคลิกปุ่ม “Validate” เพื่อเริ่มตรวจสอบ โดยพิจารณาได้ดังกรณีต่อไปนี้

1) กรณีเอกสาร RSS มีคุณสมบัติ Well-Formed และคุณสมบัติ Valid หรือนิยามโครงสร้างได้ถูกต้อง ดังแสดงในภาพประกอบ 4.40 เมื่อตรวจสอบความถูกต้องเรียบร้อยแล้วจะปรากฏหน้าต่างดังภาพประกอบ 4.41

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rss version="2.0">
3  <channel>
4        <title>News-Computer Center</title>
5        <link>http://localhost/SInfoNM/</link>
6        <description>Computer Center News</description>
7        <lastBuildDate>Sun, 26 Apr 2009 01:11:01 GMT</lastBuildDate>
8        <item>
9              <title>PSU Wi-Fi Password</title>
10             <link>http://127.0.0.1/SInfoNM/desc_news.php?itemid=16</link>
11             <description class="cipher" algorithm="blowfish" username="5010220120">3130 3431
12                   4236 3044 4637 4230 3437 3043 3638 4644 3030 3342 4337 3036 4132
13                   4230:EBC2D1803001FD9E</description>
14             <pubDate>Sun, 26 Apr 2009 02:33:47 GMT</pubDate>
15       </item>
16       <item>
17             <title>Computer Literacy 2009</title>
18             <link>http://ac-service.cc.psu.ac.th/images/stories/training/pr/com_skill.pdf</link>
19             <description class="mesg">Computer Literacy 2009</description>
20             <pubDate>Sat, 28 Mar 2009 01:44:37 GMT</pubDate>
21       </item>
22 </channel>
23 </rss>
```

ภาพประกอบ 4.40 เอกสาร RSS ที่มีคุณสมบัติ Well-Formed และคุณสมบัติ Valid

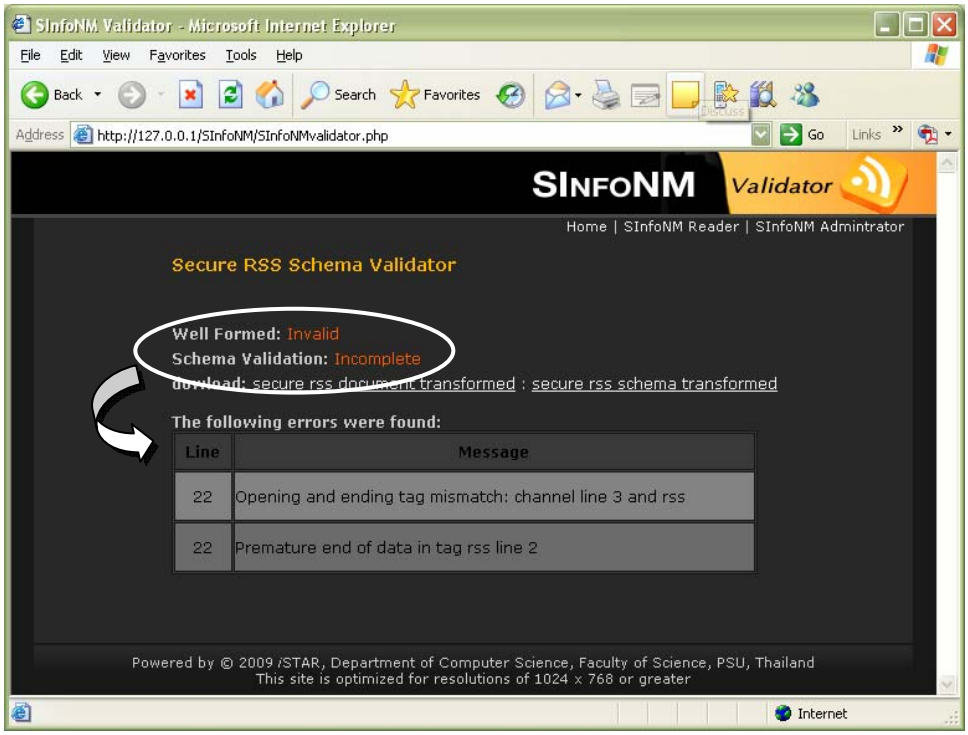


ภาพประกอบ 4.41 หน้าต่างแสดงผลการตรวจสอบเอกสาร RSS
ที่มีคุณสมบัติ Well-Formed และคุณสมบัติ Valid

2) กรณีเอกสาร RSS ไม่เป็นไปตามคุณสมบัติ Well-Formed ดังแสดงในภาพประกอบ 4.42 ซึ่งขาดแท็กสิ้นสุด </channel> ทำให้ไม่สามารถตรวจสอบคุณสมบัติ Valid ได้ ดังนั้นเมื่อนำไปตรวจสอบความถูกต้องจะปรากฏหน้าต่างดังแสดงในภาพประกอบ 4.43 เพื่อแจ้งข้อผิดพลาดที่เกิดขึ้นพร้อมกับหมายเลขบรรทัดที่ทำให้เกิดข้อผิดพลาดนั้น ๆ

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rss version="2.0">
3  <channel>
4      <title>News-Computer Center</title>
5      <link>http://localhost/SInfoNM/</link>
6      <description>Computer Center News</description>
7      <lastBuildDate>Sun, 26 Apr 2009 01:11:01 GMT</lastBuildDate>
8      <item>
9          <title>PSU Wi-Fi Password</title>
10         <link>http://127.0.0.1/SInfoNM/desc_news.php?itemid=16</link>
11         <description class="cipher" algorithm="blowfish" username="5010220120">3130 3431
12             4236 3044 4637 4230 3437 3043 3638 4644 3030 3342 4337 3036 4132
13             4230:EBC2D1803001FD9E</description>
14         <pubDate>Sun, 26 Apr 2009 02:33:47 GMT</pubDate>
15     </item>
16     <item>
17         <title>Computer Literacy 2009</title>
18         <link>http://ac-service.cc.psu.ac.th/images/stories/training/pr/com_skill.pdf</link>
19         <description class="mesg">Computer Literacy 2009</description>
20         <pubDate>Sat, 28 Mar 2009 01:44:37 GMT</pubDate>
21     </item>
22 </rss>
```

ภาพประกอบ 4.42 เอกสาร RSS ที่ไม่เป็นไปตามคุณสมบัติ Well-Formed

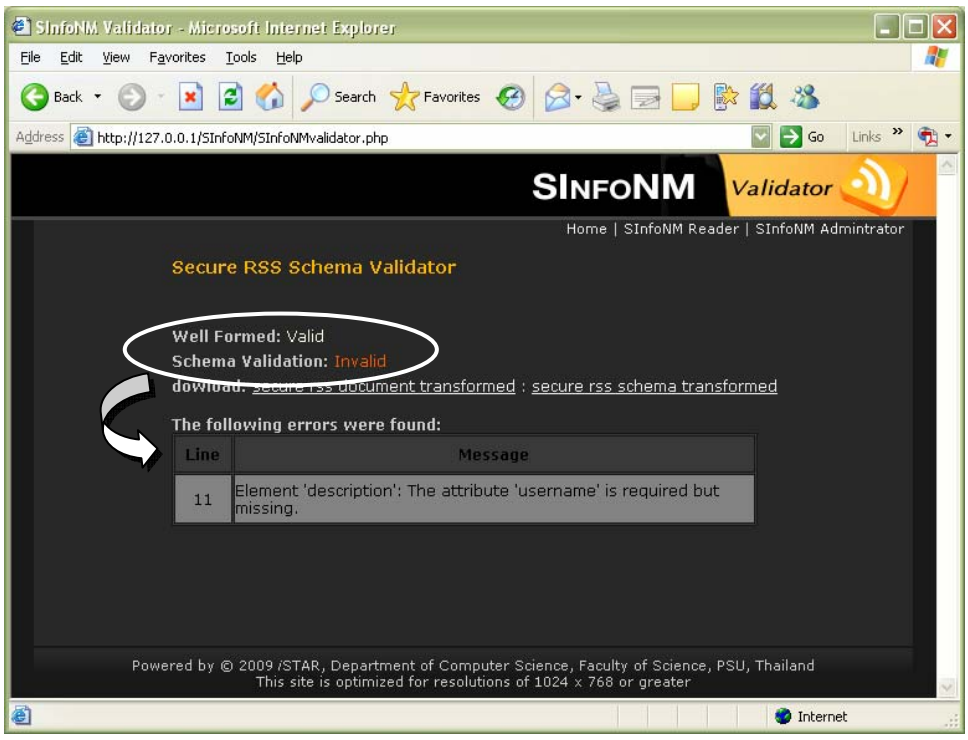


ภาพประกอบ 4.43 หน้าต่างแสดงผลการตรวจสอบเอกสาร RSS
ที่ไม่เป็นไปตามคุณสมบัติ Well-Formed

3) กรณีเอกสาร RSS มีคุณสมบัติ Well-Formed แต่ไม่มีคุณสมบัติ Valid ดังแสดงในภาพประกอบ 4.44 เนื่องจากไม่ได้กำหนด username ให้กับข้อมูลข่าวสารส่วนบุคคล เมื่อนำไปตรวจสอบความถูกต้องจะปรากฏหน้าต่าง ดังภาพประกอบ 4.45 เพื่อแจ้งข้อผิดพลาดที่เกิดขึ้นพร้อมกับหมายเลขบรรทัดที่ทำให้เกิดข้อผิดพลาดนั้น ๆ

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rss version="2.0">
3  <channel>
4       <title>News-Computer Center</title>
5       <link>http://localhost/SInfoNM/</link>
6       <description>Computer Center News</description>
7       <lastBuildDate>Sun, 26 Apr 2009 01:11:01 GMT</lastBuildDate>
8       <item>
9            <title>PSU Wi-Fi Password</title>
10           <link>http://127.0.0.1/SInfoNM/desc_news.php?itemid=16</link>
11           <description class="cipher" algorithm="blowfish">3130 3431
12                4236 3044 4637 4230 3437 3043 3638 4644 3030 3342 4337 3036 4132
13                4230:EBC2D1803001FD9E</description>
14           <pubDate>Sun, 26 Apr 2009 02:33:47 GMT</pubDate>
15      </item>
16      <item>
17           <title>Computer Literacy 2009</title>
18           <link>http://ac-service.cc.psu.ac.th/images/stories/training/pr/com_skill.pdf</link>
19           <description class="mesg">Computer Literacy 2009</description>
20           <pubDate>Sat, 28 Mar 2009 01:44:37 GMT</pubDate>
21      </item>
22 </channel>
23 </rss>
```

ภาพประกอบ 4.44 เอกสาร RSS ที่มีคุณสมบัติ Well-Formed แต่ไม่มีคุณสมบัติ Valid



ภาพประกอบ 4.45 หน้าต่างแสดงผลการตรวจสอบเอกสาร RSS
ที่มีคุณสมบัติ Well-Formed แต่ไม่มีคุณสมบัติ Valid

บทที่ 5

บทสรุปและข้อเสนอแนะ

วิทยานิพนธ์นี้ได้ออกแบบกลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP (A Secure Information Notified Mechanism with RSS Technology for TCP/IP-based Mobile Devices: SInfoNM) เพื่อทำให้ RSS สามารถแจ้งสารสนเทศที่ต้องการความปลอดภัยไปยังผู้ที่เกี่ยวข้องได้ โดยเสนอกลไกการทำงานบนพื้นฐานของอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP ทำให้ผู้รับบริการสามารถติดตามข้อมูลข่าวสารที่ทันสมัยสะดวกยิ่งขึ้น นอกจากนี้ยังได้พัฒนาระบบเพื่อทดสอบกลไกการทำงานตามที่ออกแบบไว้ ผลการศึกษาแสดงให้เห็นว่ากลไกดังกล่าวทำให้ RSS สามารถแจ้งสารสนเทศที่ต้องการความปลอดภัยไปยังผู้ที่เกี่ยวข้องได้

5.1 สรุปผลการวิจัย

กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP หรือเรียกว่า “SInfoNM” ประยุกต์ใช้วิทยาการเข้ารหัสลับเพื่อทำให้ข้อมูลข่าวสารมีความปลอดภัยก่อนแจ้งไปยังผู้ที่เกี่ยวข้อง และนำ XSL ซึ่งเป็นเทคโนโลยีที่เกี่ยวข้องกับ XML ที่เป็นที่รู้จักมาใช้เพื่อสืบค้นข้อมูลที่ถูกเข้ารหัสไว้ โดยข้อมูลที่สืบค้นได้จะถูกส่งไปถอดรหัสก่อนแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้รับบริการ ซึ่งมีการนิยามโครงสร้างเอกสาร RSS ด้วย XML Schema ร่วมกับ SchemaPath โดยสามารถพิจารณา SInfoNM ในมุมมองผู้ที่เกี่ยวข้องกับโครงสร้างการทำงานของ RSS ได้ดังนี้

5.1.1 ผู้เผยแพร่ข้อมูลข่าวสาร (Publisher)

1) สามารถแจ้งข้อมูลข่าวสารไปยังผู้รับบริการทั่วไป และเจาะจงเฉพาะผู้รับบริการที่เกี่ยวข้องโดยใช้เทคโนโลยี RSS ได้

2) สามารถสร้างเอกสาร RSS สำหรับเผยแพร่ข้อมูลข่าวสารที่ต้องการความปลอดภัยได้ง่าย เพียงระบุแอททริบิวต์เพื่อกำหนดคุณสมบัติเพิ่มเติมให้กับแท็ก <description> ของแท็ก <item> และเข้ารหัสข้อมูลภายในแท็ก <description> เท่านั้น

5.1.2 ผู้รวบรวมข้อมูลข่าวสาร (Aggregator)

1) กลไกการทำงานใช้มาตรฐานและเทคโนโลยีทั่วไปของ XML ที่เป็นที่รู้จักทำให้ผู้พัฒนา RSS Reader หรือ Aggregator สามารถนำไปประยุกต์ใช้งานได้ง่าย

2) ไม่ต้องปรับปรุง RSS Reader หากผู้รับบริการต้องการอ่านเฉพาะข้อมูลข่าวสารทั่วไป เนื่องจากเป็นการเพิ่มแอททริบิวต์เพื่อกำหนดคุณสมบัติเพิ่มเติมให้กับแท็ก <description> ของแท็ก <item> ซึ่งไม่มีผลกระทบกับกระบวนการวิเคราะห์โครงสร้างข้อมูลของเอกสาร RSS

5.1.3 ผู้รับบริการข้อมูลข่าวสาร (Subscriber)

1) สามารถอ่านข้อมูลข่าวสารทั่วไป และข้อมูลข่าวสารส่วนบุคคลได้ด้วยการเข้าใช้งานเพียงที่เดียว

2) กลไกที่นำเสนอทำให้ผู้รับบริการสามารถอ่านข้อมูลข่าวสารส่วนบุคคลโดยใช้เบราว์เซอร์ได้หลากหลาย ทั้งบนเครื่องคอมพิวเตอร์ทั่วไปและอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP

3) สามารถนำเอกสาร RSS ที่นิยามโครงสร้างตามที่ได้ออกแบบไว้ไปใช้งานกับ RSS Reader ที่มีอยู่เพื่ออ่านข้อมูลข่าวสารทั่วไปได้

5.2 ปัญหาและอุปสรรค

5.2.1 เนื่องจาก RSS ถูกพัฒนาด้วยภาษา XML ทำให้การจัดการเอกสาร RSS จึงเกี่ยวข้องกับมาตรฐานและเทคโนโลยีของ XML ได้แก่ XML Parser, XSL, XPath และ XML Schema เป็นต้น โดยแต่ละเทคโนโลยีมีวิธีการทำงานที่หลากหลายทำให้ใช้เวลาศึกษาค่อนข้างมาก

5.2.2 เนื่องจาก XML Schema ไม่รองรับเงื่อนไขสำหรับกำหนดชนิดข้อมูลให้กับอิลิเมนต์และแอททริบิวต์ ทำให้การนิยามโครงสร้างเอกสาร RSS ตามกลไกที่ออกแบบไว้ไม่สามารถใช้ XML Schema เพียงอย่างเดียวได้ จึงต้องนำ SchemaPath มาประยุกต์ใช้งานร่วมกัน

5.2.3 การแจ้งข้อมูลข่าวสารที่ต้องการความปลอดภัยหรือข้อมูลข่าวสารที่เป็นความลับ ใช้วิทยาการเข้ารหัสลับเพื่อเข้ารหัสและถอดรหัสข้อมูล ทำให้ต้องศึกษาขั้นตอนวิธีต่าง ๆ ซึ่งมีความซับซ้อนจึงใช้เวลาศึกษาค่อนข้างนาน

5.3 ข้อเสนอแนะ

5.3.1 กลไกที่นำเสนอเหมาะกับการแจ้งข้อมูลข่าวสารส่วนบุคคลที่มีลักษณะเป็นข้อความสั้น ๆ เนื่องจากอุปกรณ์สื่อสารเคลื่อนที่มีทรัพยากรในการประมวลผลจำกัด ทำให้การถอดรหัสข้อมูลใช้เวลาค่อนข้างมาก

5.3.2 กลไกการทำงานที่นำเสนอเหมาะกับองค์กรที่มีหน่วยงานย่อยภายในซึ่งกำหนดให้ใช้ username เดียวกัน สำหรับเข้าถึงข้อมูลของหน่วยงานต่าง ๆ เพื่อทำให้การรวบรวมและแสดงข้อมูลข่าวสารจากหน่วยงานไปยังผู้ที่เกี่ยวข้องสามารถทำได้โดยใช้เพียง username เดียว

5.3.3 กลไกที่ออกแบบรองรับการทำงานเฉพาะภาษาอังกฤษเท่านั้น จึงควรมีการพัฒนาให้สามารถใช้งานได้กับภาษาไทยด้วย

บรรณานุกรม

จตุชัย แพงจันทร์. 2550. Master in Security. พิมพ์ครั้งที่ 1. ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์: นนทบุรี.

เดวิด ฮันเตอร์. 2545. คัมภีร์การใช้ XML ฉบับสมบูรณ์. ซีเอ็ดยูเคชั่น: กรุงเทพฯ.

ชารวีย์ แสงขำ. 2552. เทคนิคการคัดกรองวิดีโออัตโนมัติสำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโปรโตคอล TCP/IP. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา.

ทัศนวรรณ ศูนย์กลาง. 2552. วากยสัมพันธ์. [ออนไลน์] เข้าถึงได้จาก: http://www.cp.su.ac.th/~tasanawa/cs517211/chap9.pdf (วันที่สืบค้น 18 เมษายน 2552).

ไพศาล โมลิสกุลมงคล. 2538. พัฒนา Web Database ด้วย PHP. ไทยเจริญการพิมพ์: กรุงเทพฯ.

ลัดดา ปรีชาวีรกุล. 2551. วิทยาการเข้ารหัสลับเบื้องต้น. พิมพ์ครั้งที่ 1. โรงพิมพ์ดิจิตอล คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์: สงขลา.

วิทูร ชื่นวชิรศิริ. 2550. การสร้าง XML Schema. [ออนไลน์] เข้าถึงได้จาก: http://gdi.mict.go.th /images_2007/ GDI2_Day1-2_XML_schema.pdf (วันที่สืบคืน 16 เมษายน 2552).

สราวุธ อ้อยศีสกุล. 2551. เริ่มคิด-เริ่มสร้าง-เริ่มใช้ XML 2nd edition. พิมพ์ครั้งที่ 1. วิตตี้ กรุ๊ป: กรุงเทพฯ.

สุธี พงศาสกุลชัย. 2550. การพัฒนาระบบด้วยสถาปัตยกรรมเชิงบริการบนเทคโนโลยีของ Web Service. พิมพ์ครั้งที่ 1. ไทยเจริญการพิมพ์: กรุงเทพฯ.

Bartlett, R. G., and Cook, M. W. 2002. XML Security Using XSLT. 36th Hawaii International Conference on System Sciences. Vol. 4, No 4. pp.122b.

BBC. 2008. NEWS. http://news.bbc.co.uk/2/hi/science/nature/default.stm (accessed 11/9/2008).

Benz, B., and Durant J. R. 2003. XML Programming Bible. Wiley Publishing: NY.

Biron, V. P., Permanente, K., and Malhotr, A. 2004. XML Schema Part 2: Datatypes Second Edition. http://www.w3.org/TR/xmlschema-2/ (accessed 16/4/2009).

Blekas, A., Garofalakis, J., and Stefanis, V. 2006. Use of RSS feeds for Content Adaptation in Mobile Web Browsing. WWW2006. pp.79-85.

Bloglines. 2008. Bloglines. http://www.bloglines.com (accessed 11/9/2008).

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., and Yergeau, F. 2008. Extensible Markup Language (XML) 1.0 (Fifth Edition). http://www.w3.org/TR/2008/REC-xml-20081126/ (accessed 14/4/2009).

Carlisle, D., Lon, Patrick., Miner, R., and Popelier, N. 2003. Mathematical Markup Language (MathML) Version 2.0 (Second Edition). http://www.w3.org/TR/MathML/ (accessed 13/01/2009).

Ch7.com. 2008. Channel7. http://www.ch7.com/website/ (accessed 11/9/2008).

Chang, T., and Hwang, G. 2004. Using the Extension Function of XSLT and DSL to Secure XML Documents. The 18th International Conference on Advanced Information Networking and Application. pp.1-6.

Chang, T., and Hwang, G. 2006. The design and implementation of an application program interface for securing XML documents. The Journal Systems and Software. August, 2007. pp.1362-1374.

Clark, J., and DeRose, S. 1999. XML Path Language (XPath) Version 1.0. http://www.w3.org/TR/xpath (accessed 28/5/2008).

CNN. 2009. CNN RSS. http://edition.cnn.com/services/rss/?iref=rssmorenews (accessed 18/4/2009).

Cold, S. J. 2006. Using Really Simple Syndication (RSS) to Enhance Student Research. ACM SIGITE Newsletter. Vol.3, No.1, January, 2006. pp.6-9.

Coulouris, G., Dollimore, J., and Kindberg, T. 2001. Distributed Systems Concept and Design Third edition. Addison-Wesley.

Dykes, L., and Tittel, E. 2005. XML For Dummies 4th Edition. Wiley Publishing: NJ.

Finkelstein, E. 2005. Syndicating Web Sites with RSS Feeds For Dummies. Wiley Publishing: NJ.

Forouzan, B. A. 2008. Introduction to Cryptography and Network Security. McGraw-Hill Publishing: NY.

Glotzbach, R. J., Mohler, J. L., and Radwan, J. E. RSS as a Course Information Delivery Method. International Conference on Computer Graphics and Interactive Techniques. San Diego, California, August 5-9, 2007.

Gregorio, J. 2005. Secure RSS Syndication. http://www.xml.com/pub/a/2005/07/13/secure-rss.html (accessed 15/3/2008).

Hammersley, B. 2003. Content Syndicating with RSS. O'Reilly & Associates: CA.

Haw, S. C., and Rao, G. S. V. R. K. 2007. A Comparative Study and Benchmarking on XML Parsers. ICACT2007. February 12-14, 2007. pp.321-325.

Kahate, A. 2003. Cryptography and Network Security. McGraw-Hill Publishing: Singapore.

Manager Online. 2008. Manager Online. http://www.manager.co.th/ (accessed 11/9/2008).

Marinelli, P., Coen, C. S., and Vitali, F. 2004. SchemaPath, a Minimal Extension to XML Schema for Conditional Constraints. WWW2004. New York, NY USA. May 17-22, 2004. pp.164-174.

Maruyama, H., and Imamura, T. 2000. Element-wise XML Encryption. http://lists.w3.org/Archives/Public/xml-encryption/2000Apr/att-0005/01-xmlenc (accessed 15/1/2008).

Microsoft. 2008. Windows Mobile 6.1 Emulator Images. http://www.microsoft.com/downloads/details.aspx?FamilyId=3D6F581E-C093-4B15-AB0C-A2CE5BFFDB47&displaylang=en (accessed 15/05/2008).

Medicalnewstoday.com. 2008. Medical RSS/XML News Feeds http://www.medicalnewstoday.com/index.php?page=newsfeed (accessed 11/9/2008).

Ohdave.com. 2008. RSA in JavaScript. http://ohdave.com/rsa/ (accessed 6/7/2008).

PGP. An Introduction to Cryptography. http://download.pgp.com/pdfs/Intro_to_Crypto_040600_F.pdf (accessed 11/9/2008).

RSS Thai. 2008. Latest News. http://www.rssthai.com/ (accessed 11/9/2008).

SAP. 2009. 'Opening' a Digital Envelope (Develope). http://help.sap.com/saphelp_nw04/Helpdata/EN/b8/822000dadd11d2a60a0000e835363f/content.htm (accessed 14/4/2009).

Sebesta, R. W. 2005. Concept of Programming Languages 7th EDITION. Addison-Wesley: Colorado.

Stevens, W. R. 1994. TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley.

Thelin, J. RSS 2.0 Schema. http://www.thearchitect.co.uk/schemas/rss-2_0.xsd (accessed 31/10/ 2008).

Umbach, K. W. 1997. What is "Push Technology"?. http://www.library.ca.gov/crb/97/notes/V4n6.pdf (accessed 12/11/2008).

Vitali, F. 2004. SchemaPath: an extension of XML Schema. http://tesi.fabio.web.cs.unibo.it/Tesi/SchemaPath (accessed 21/3/2009).

W3Schools. 2008. XML Schema Tutorial. http://www.w3schools.com/schema/default.asp (accessed 13/4/2009).

WAP Forum. 2001. Wireless Markup Language Version 2.0. http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf (accessed 13/01/2009).

Wikipedia. 2008. Mobile Code. http://en.wikipedia.org/wiki/Mobile_code (accessed 4/9/2008).

ภาคผนวก

ภาคผนวก ก

ผลงานตีพิมพ์

เรื่อง	กลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่
งานประชุมวิชาการ	วิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 12 (NCSEC 2008)
สถานที่	จังหวัดชลบุรี ประเทศไทย
วันที่	20 - 21 พฤศจิกายน 2551

กลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่

A Notified Secure Information Mechanism with RSS Technology for Mobile Users

วิชุตา แก้วนพรัตน์ และ ลัดดา ปรีชาวีรกุล

ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

ตำบลคอหงส์ อำเภอหาดใหญ่ จังหวัดสงขลา 90112 โทรศัพท์: 0-7428-8581 โทรสาร: 0-7444-6917

E-mail: {s5010220120, ladda.p}@psu.ac.th

บทคัดย่อ

ในยุคข้อมูลข่าวสาร หลายองค์กรให้ความสำคัญในการเผยแพร่ข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล เช่น ข้อมูลการใช้บัตรเครดิต ข้อมูลการทำธุรกรรมทางการเงิน และข้อมูลอื่น ๆ ผ่านอินเทอร์เน็ตบนอุปกรณ์สื่อสารเคลื่อนที่มากขึ้น ซึ่ง RSS (Really Simple Syndication) เป็นเทคโนโลยีหนึ่งที่ถูกนำมาใช้เพื่อเผยแพร่ข้อมูลข่าวสารที่ทันสมัยให้กับผู้ใช้ อย่างไรก็ตาม RSS ไม่ได้จัดเตรียมกลไกที่ทำให้ผู้ใช้มั่นใจว่า ข้อมูลข่าวสารที่ได้รับมีความปลอดภัย บทความนี้จึงนำเสนอกลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS โดยได้ออกแบบและพัฒนากลไกสำหรับการใช้งานบนอุปกรณ์สื่อสารเคลื่อนที่ เพื่อให้ผู้ใช้ได้รับข้อมูลข่าวสารที่ทันสมัยจากองค์กรต่าง ๆ

คำสำคัญ: ความปลอดภัย, RSS, Really Simple Syndication, อุปกรณ์สื่อสารเคลื่อนที่

Abstract

In the Information age, many organizations focus on disseminating information both general and sensitive information, such as credit card information, financial business information, etc. In addition using mobile device is growing rapidly. Really Simple Syndication (RSS) is used to distribute latest information over the Internet to users. However, the RSS technology does not have a mechanism to ensure that the incoming information is really secure. Therefore, we propose a Notified Secure Information Mechanism with RSS Technology for Mobile Users (NSIM) and develop the system as design.

Keywords: Security, RSS, Really Simple Syndication, Mobile device

1. บทนำ

RSS [1] เป็นเทคโนโลยีการสื่อสารข้อมูลด้วยมาตรฐาน XML (Extensible Markup Language) [2] ถูกนำมาใช้อย่างแพร่หลายในกลุ่มของเว็บล็อก (Web log) และเว็บไซต์ผู้ให้บริการข่าว (News Site) เพื่อเผยแพร่ข่าวสารต่าง ๆ ของเว็บไซต์ และด้วยความสามารถของ RSS ที่ไม่ได้จำกัดการใช้งานเฉพาะข่าวสารเท่านั้น แต่ยังสามารถใช้งาน RSS เพื่อเผยแพร่เนื้อหาที่หลากหลายตามความต้องการของผู้ใช้ได้อีกด้วย หลายองค์กรจึงหันมาให้ความสำคัญในการเผยแพร่ข้อมูลขององค์กรผ่าน RSS มากขึ้น อย่างไรก็ตามข้อมูลข่าวสารที่ใช้งานผ่าน RSS โดยมากเป็นเพียงข้อมูลข่าวสารที่ต้องการประชาสัมพันธ์ให้บุคคลทั่วไปทราบ จึงไม่ใช่ข้อมูลข่าวสารที่เป็นความลับแต่อย่างใด ดังนั้นการใช้งาน RSS ที่ไม่ได้จัดเตรียมกลไกสำหรับความปลอดภัยเพื่อเผยแพร่ข้อมูลข่าวสารส่วนบุคคลผ่านเครือข่ายอินเทอร์เน็ต จึงเป็นประเด็นสำคัญที่ต้องพิจารณา และแม้ว่า XML จะมีมาตรฐานรองรับความปลอดภัยของข้อมูล [3-4] ก็ตาม แต่ด้วยข้อจำกัดของแท็ก (Tag) ที่กำหนดไว้ในเอกสาร RSS feed ทำให้ไม่สามารถใช้งานมาตรฐานที่มี เพื่อจัดการกับข้อมูลบางส่วนที่เป็นความลับในเอกสาร RSS feed ได้

บทความนี้จึงนำเสนอกลไกในการรวบรวมข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคลขององค์กรต่าง ๆ ในรูปแบบเอกสาร RSS feed โดยทำการเข้ารหัสข้อมูลข่าวสารส่วนบุคคล เพื่อให้มีความปลอดภัยก่อนเผยแพร่ไปยังผู้ที่เกี่ยวข้อง จากนั้นจึงทำการถอดรหัสข้อมูลและแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้ กลไกดังกล่าวทำให้ผู้ใช้ได้รับข้อมูลข่าวสารที่ทันสมัยและมีความปลอดภัยจากองค์กรต่าง ๆ

เนื้อหาของบทความในส่วนที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง ส่วนที่ 3 อธิบายการทำงานของกลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่ ในส่วนที่ 4 แสดงการพัฒนาระบบและผลลัพธ์ ส่วนที่ 5 เป็นบทสรุป

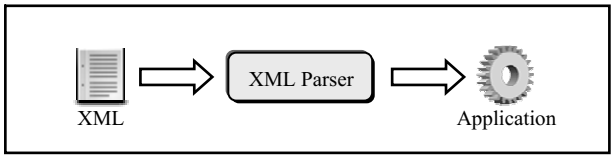
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ภาษา XML (Extensible Markup Language)

ภาษา XML ถูกนำมาใช้เป็นสื่อกลางในการแลกเปลี่ยนข้อมูลข่าวสารผ่านระบบอินเทอร์เน็ต เนื่องจากเป็นภาษาที่ได้รับการออกแบบเพื่อให้สามารถอธิบายความหมายของข้อมูล และยังอนุญาตให้ผู้ใช้กำหนดแท็กได้ตามต้องการ จึงทำให้ XML มีความยืดหยุ่นและใช้งานได้หลากหลาย

2.1.1 XML Parser

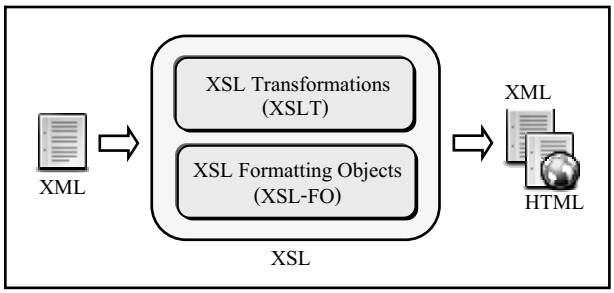
คือ ตัวแปลภาษา XML ทำหน้าที่อ่าน แปลความหมาย วิเคราะห์โครงสร้าง รวมถึงตรวจสอบความถูกต้องของเอกสาร XML โดยสามารถจำแนก XML Parser ตามวิธีการสำรวจเนื้อหาของเอกสารออกเป็น 2 ชนิด คือ DOM (Document Object Model) ซึ่งเป็นแบบ Tree-based Parser และ SAX (Simple API for XML) เป็นแบบ Event-driven Parser โดยกระบวนการทำงานของ XML Parser แสดงดังรูปที่ 1



รูปที่ 1 กระบวนการทำงานของ XML Parser

2.1.2 XSL (Extensible Stylesheet Language)

XSL [5] เป็นเทคโนโลยีที่นำมาใช้จัดการรูปแบบการแสดงผลและแปลงเอกสาร XML ให้อยู่ในรูปแบบเอกสารที่ต้องการ กระบวนการทำงานของ XSL แสดงดังรูปที่ 2 โดย XSL ประกอบด้วย 2 ส่วน คือ ภาษาสำหรับแปลงรูปแบบเอกสาร (XSL Transformations) และภาษาสำหรับจัดรูปแบบ (XSL Formatting Objects)



รูปที่ 2 กระบวนการทำงานของ XSL

XSLT (Extensible Stylesheet Language Transformations) ทำหน้าที่แปลงเอกสาร XML ตามแบบอย่างที่กำหนดไว้ โดย XSLT Processor ทำการประมวลผลไวยากรณ์ของ XPath [6] ที่ระบุตำแหน่งในการค้นหาข้อมูลของเอกสาร XML ไว้ จากนั้นจึงแสดงผลลัพธ์ตามรูปแบบที่ต้องการ

2.2 เทคโนโลยี RSS

RSS เป็นเทคโนโลยีการสื่อสารข้อมูลด้วยมาตรฐาน XML ใช้สำหรับเผยแพร่ข่าวสารของเว็บไซต์ ทำให้ผู้ใช้ได้รับข้อมูลข่าวสารที่ทันสมัยจากเว็บไซต์ที่ให้บริการ โดยผู้ใช้ไม่จำเป็นต้องเข้าไปยังเว็บไซต์ต่าง ๆ เพื่อดูว่ามีการปรับปรุงข้อมูลใหม่หรือไม่ อีกทั้งยังช่วยลดปัญหาในเรื่องการละเมิดลิขสิทธิ์ และทำให้ผู้พัฒนาเว็บไซต์ไม่ต้องเสียเวลาปรับปรุงเว็บเพจเมื่อผู้ให้บริการมีการปรับปรุงข้อมูลข่าวสาร ซึ่งผู้ใช้บริการสามารถเรียกดูข้อมูล RSS ได้จากโปรแกรมรวบรวมข่าวสารที่เรียกว่า Reader หรือ Aggregator แบ่งออกเป็น 2 ประเภท คือ Software Reader และ Web-based RSS Reader โดยผู้ให้บริการ RSS ต้องจัดเตรียมเอกสาร XML ที่เรียกว่า RSS feed ไว้สำหรับผู้ใช้

```
<?xml version = "1.0" ?>
<rss version = "2.0" >
    <channel>
        <title>…</title>
        <link>…</link>
        <description>…</description>
        …
        <item>
            <title>…</title>
            <link>…</link>
            <description>…</description>
            <pubDate>…</pubDate>
            ...
        </item>
        …
    </channel>
</rss
```

รูปที่ 3 โครงสร้างทั่วไปตามรูปแบบเอกสาร RSS 2.0

จากรูปที่ 3 แสดงโครงสร้างทั่วไปของเอกสาร RSS 2.0 [7] โดยมีแท็ก <rss> บอกจุดเริ่มต้น ตามด้วยแท็ก <channel> เก็บข้อมูลต่าง ๆ ของ RSS ไว้ และมีแท็ก <item> เป็นส่วนสำคัญทำหน้าที่เก็บรายการข้อมูล ซึ่งประกอบด้วยแท็กต่าง ๆ ที่อยู่ภายในเพื่อบอกรายละเอียดข้อมูลแต่ละรายการ โดยตัวอย่างและคำอธิบายแท็กภายในแท็ก <item> แสดงดังตารางที่ 1 นอกจากนี้ในแต่ละแท็กยังสามารถกำหนดแอททริบิวต์ (Attribute) เพื่ออธิบายข้อมูลเพิ่มเติมได้อีกด้วย

ตารางที่ 1 ตัวอย่างแท็กภายในแท็ก <item> ของ RSS feed

RSS item tag	คำอธิบาย
<title>	หัวเรื่องรายการข้อมูล
<link>	ลิงค์เชื่อมโยงข้อมูลหลัก
<description>	รายละเอียดข้อมูลโดยย่อ
<pubDate>	วันเวลาที่เผยแพร่ข้อมูล
<category>	ประเภทของข้อมูล

2.3 ความปลอดภัย (Security)

ความปลอดภัยในการแลกเปลี่ยนข้อมูลผ่านระบบอินเทอร์เน็ต ถูกนำมาพิจารณาเป็นประเด็นหลัก เพื่อให้มั่นใจได้ว่าข้อมูลที่สำคัญจะไม่รั่วไหลไปยังบุคคลที่ไม่เกี่ยวข้อง โดยองค์ประกอบพื้นฐานสำคัญที่ทำให้ข้อมูลมีความปลอดภัย คือ การมีบูรณภาพ (Integrity) การรักษาความลับ (Confidentiality) การพิสูจน์ตัวตน (Authentication) และการไม่ปฏิเสธการกระทำ (Non-repudiation)

2.3.1 วิทยาการเข้ารหัสลับ (Cryptography)

วัตถุประสงค์ของวิทยาการเข้ารหัสลับ คือ การทำให้บุคคล 2 ฝ่ายสามารถติดต่อสื่อสารข้อมูลระหว่างกันได้อย่างปลอดภัย ซึ่งมีองค์ประกอบสำคัญ คือ อัลกอริทึม (Algorithm) และกุญแจ (Key) ที่ใช้ในการเข้ารหัสและถอดรหัสข้อความ โดยมีขั้นตอนการทำงานดังนี้

1) ผู้ส่งเข้ารหัสข้อความต้นฉบับ (Plain Text) ด้วยอัลกอริทึมและกุญแจในการเข้ารหัส ได้เป็นข้อความไซเฟอร์ (Cipher Text) เพื่อส่งไปยังฝั่งผู้รับ

2) ผู้รับทำการถอดรหัสข้อความไซเฟอร์ที่ได้ ด้วยอัลกอริทึมและกุญแจในการถอดรหัสได้เป็นข้อความต้นฉบับ

2.3.2 อัลกอริทึม RSA

RSA [8] เป็นอัลกอริทึมเข้ารหัสลับแบบกุญแจอสมมาตร (Asymmetric Key Cryptography) โดยการเข้ารหัสจะใช้กุญแจหนึ่งสำหรับเข้ารหัสและใช้อีกกุญแจหนึ่งสำหรับถอดรหัส ซึ่งกุญแจทั้งสองมีความสัมพันธ์กันในทางคณิตศาสตร์ โดยขั้นตอนการทำงานของอัลกอริทึม RSA มีดังนี้

1) สุ่มเลือกจำนวนเฉพาะขนาดใหญ่ 2 จำนวน กำหนดให้เป็น P และ Q จากนั้นคำนวณค่า N เพื่อใช้ในการเข้ารหัสและถอดรหัสข้อความด้วยสมการที่ 1

$$N = P \times Q \tag{1}$$

2) เลือกกุญแจสาธารณะ (E) เพื่อใช้เข้ารหัสข้อความ โดยต้องไม่เป็นตัวประกอบของผลคูณของ (P-1) กับ (Q-1)

3) เลือกกุญแจส่วนตัว (D) เพื่อใช้ถอดรหัสข้อความ ซึ่งทำให้สมการที่ 2 เป็นจริง

$$(D \times E) \bmod (P-1) \times (Q-1) = 1 \tag{2}$$

4) เข้ารหัสข้อความต้นฉบับ (M) ได้เป็นข้อความไซเฟอร์ (C) ด้วยสมการที่ 3

$$C = M^E \bmod N \tag{3}$$

5) ส่งข้อความไซเฟอร์ไปยังผู้รับ และถอดรหัสข้อความไซเฟอร์ได้เป็นข้อความต้นฉบับ ด้วยสมการที่ 4

$$M = C^D \bmod N \tag{4}$$

2.4 งานวิจัยที่เกี่ยวข้อง

เทคโนโลยี RSS ถูกนำมาใช้อย่างแพร่หลายสำหรับเว็บข่าวและเว็บล็อก แต่ยังมีงานวิจัยหลายงานด้วยกันที่ประยุกต์ใช้ RSS กับงานด้านอื่น ๆ ตัวอย่างทางด้านการศึกษา ได้แก่งานวิจัยของ Glotzbach และคณะ [9] ใช้เทคโนโลยี RSS เพื่อแจ้งข้อมูลต่าง ๆ ของรายวิชาเรียน เช่น ข้อมูลเกี่ยวกับการสอบ เอกสารประกอบการเรียน เป็นต้น ทำให้นักศึกษาได้รับข้อมูลข่าวสารเกี่ยวกับวิชาที่เรียนได้สะดวกและรวดเร็วขึ้น นอกจากนี้ Cold [10] เสนอการใช้ RSS เป็นเครื่องมือในการแบ่งปันข้อมูลสำหรับทำวิจัยของกลุ่มนักศึกษา โดยข้อมูลที่นักศึกษาได้รวบรวมไว้ เช่น วารสารทางวิชาการ ผลงานที่ได้รับการตีพิมพ์ บล็อกและข้อมูลจากแหล่งข้อมูลต่าง ๆ จะถูกนำมาแบ่งปันเพื่อใช้ประโยชน์ในการทำวิจัย

ด้านความปลอดภัย Maruyama และคณะ [11] ได้เสนอการเข้ารหัสเอกสาร XML เฉพาะส่วน เรียกว่า Element-Wise XML Encryption ผลลัพธ์ที่ได้ คือ เอกสาร XML ที่มีข้อมูลบางส่วนถูกเข้ารหัสไว้ Bartlett และ Cook [12] ได้เสนอการใช้ XSLT เพื่อเข้ารหัสและถอดรหัสเอกสาร XML โดยได้เสนอกลไกที่ขยายการทำงานของ XSLT สำหรับเข้ารหัสเอกสาร XML และยังมีงานวิจัยของ Chang และ Hwang [13] ได้เสนอการออกแบบและประยุกต์ใช้ API สำหรับความปลอดภัยเอกสาร XML โดยได้นิยามภาษาที่เรียกว่า DSL (Document Security Language) เพื่อใช้ในการเข้ารหัสและถอดรหัสเอกสาร XML ซึ่งมีแนวคิดพื้นฐานมาจาก XSL

นอกจากนี้ Gregorio [14] ได้ประยุกต์ใช้ Greasemonkey ซึ่งเป็นส่วนขยายของ Mozilla Firefox เพื่อถอดรหัสข้อมูลภายในเอกสาร RSS feed ผลลัพธ์ที่ได้จากการถอดรหัสจะถูกแทนที่ลงในหน้าเว็บเพจที่เปิดใช้งานก่อนการแสดงผลบนเบราว์เซอร์ของผู้ใช้ และได้เสนอ Greasemonkey Script สำหรับสืบค้นข้อมูลที่ถูกเข้ารหัสไว้ และถอดรหัสข้อมูลที่ได้ด้วยอัลกอริทึม Blowfish ซึ่งเป็นอัลกอริทึมเข้ารหัสลับแบบกุญแจสมมาตร (Symmetric Key Cryptography) โดยผู้ใช้ต้องทำการติดตั้งสคริปต์บนเบราว์เซอร์ Mozilla Firefox ก่อนถึงจะถอดรหัสข้อมูลภายในเอกสาร RSS feed ได้

จากที่กล่าวมา เห็นได้ว่ามีการประยุกต์ใช้เทคโนโลยี RSS ในหลาย ๆ ด้าน อย่างไรก็ตามงานวิจัยส่วนใหญ่ยังไม่ได้กล่าวถึงการใช้เทคโนโลยี RSS เพื่อแจ้งข่าวสารที่ปลอดภัยสำหรับอุปกรณ์สื่อสารเคลื่อนที่ ดังนั้นงานวิจัยนี้จึงเสนอกลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่ เพื่อให้องค์กรต่าง ๆ สามารถแจ้งข่าวสารไปยังผู้ที่เกี่ยวข้องได้อย่างทั่วถึงและมีความปลอดภัย

3. กลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่

กลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่ (A Notified Secure Information Mechanism with RSS Technology for Mobile Users: NSIM) มีวัตถุประสงค์เพื่อให้องค์กรต่าง ๆ สามารถแจ้งข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคลได้ทั่วถึงและมีความปลอดภัย นอกจากนี้ยังทำให้ผู้ใช้ได้รับข้อมูลข่าวสารที่ทันสมัยอีกด้วย

3.1 แนวคิดสำหรับการออกแบบ

เพื่อให้เป็นไปตามวัตถุประสงค์ดังกล่าวข้างต้น การออกแบบจึงต้องคำนึงถึงสิ่งต่าง ๆ ดังต่อไปนี้

1) สามารถรวบรวมและเผยแพร่ข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคลได้

2) การเผยแพร่ข้อมูลข่าวสารส่วนบุคคลต้องมีความปลอดภัย

3) ผู้ใช้ต้องได้รับข้อมูลข่าวสารที่ทันสมัยจากองค์กรต่าง ๆ

3.2 แบบจำลองการทำงานของ NSIM

การทำงานของ NSIM เป็นแบบ Client-Server ประกอบด้วย 2 ส่วนสำคัญ คือ ส่วนรวบรวม RSS feed (NSIM Aggregator: NSIMAg) และส่วนสร้าง RSS feed สำหรับผู้ใช้ (Generate RSS for User: GRSS4U) ซึ่งแบบจำลองการทำงานโดยรวมแสดงดังรูปที่ 4

3.2.1 ส่วนรวบรวม RSS feed (NSIMAg)

ส่วนรวบรวม RSS feed ทำหน้าที่รวบรวมเอกสาร RSS feed จากองค์กรต่าง ๆ โดยแต่ละองค์กรอาจมีทั้งข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล ดังแสดงในรูปที่ 5 โดยมีขั้นตอนการทำงานดังต่อไปนี้

1) รวบรวบเอกสาร RSS feed ที่ประกอบด้วยข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคลจากองค์กรต่าง ๆ

2) Parse ข้อมูล RSS feed เก็บไว้ในฐานข้อมูลข่าวสาร

3.2.2 ส่วนสร้าง RSS feed สำหรับผู้ใช้ (GRSS4U)

ผู้ใช้สมัครใช้งานระบบด้วย Username และ Password ขององค์กร โดย NSIM จะทำการสร้างไฟล์ข้อมูลส่วนตัวผู้ใช้ (User's Profile) ไว้บนอุปกรณ์สื่อสารเคลื่อนที่ และสร้างเอกสาร RSS feed ส่งกลับไปยังผู้ใช้ ดังแสดงในรูปที่ 6 โดยสามารถอธิบายขั้นตอนการทำงานได้ดังนี้

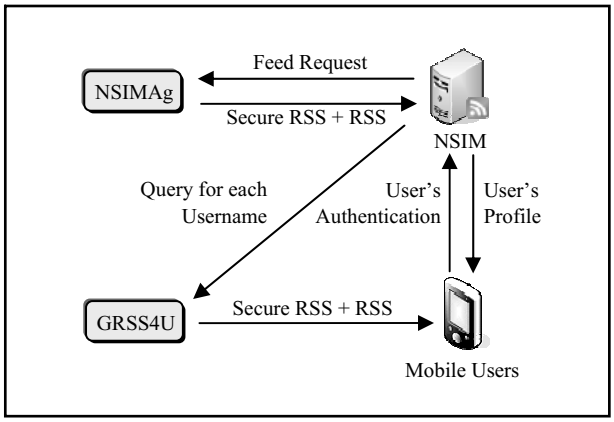
1) NSIM ตรวจสอบ Username จากไฟล์ข้อมูลส่วนตัวบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้ เพื่อยืนยันตัวบุคคล

2) สืบค้นข้อมูล RSS feed โดยสอบถาม (Query) ไปยังฐานข้อมูลข่าวสารด้วย Username ของผู้ใช้

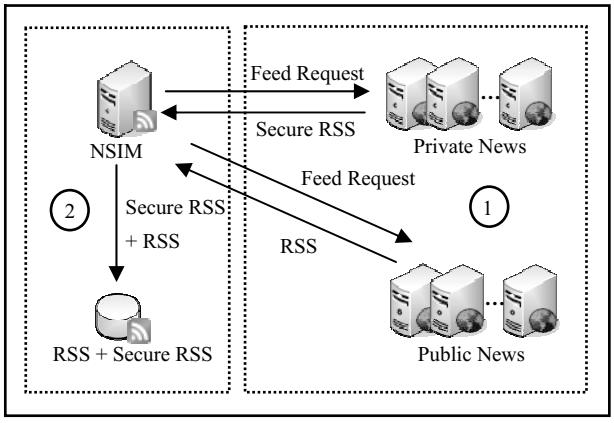
3) สร้างเอกสาร RSS feed สำหรับผู้ใช้ ประกอบด้วยข้อมูลข่าวสารทั่วไปและข้อมูลข่าวสารส่วนบุคคล ส่งกลับไปยังผู้ใช้

4) ข้อมูลข่าวสารส่วนบุคคลที่ผู้ใช้ได้รับ จะถูกส่งเข้าสู่กระบวนการถอดรหัส

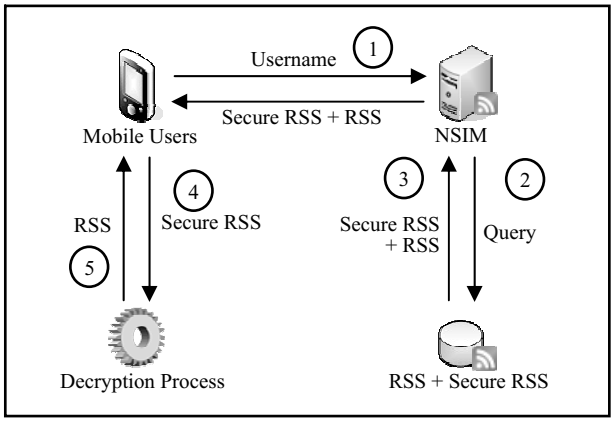
5) เมื่อถอดรหัสเรียบร้อยแล้ว ผลลัพธ์ที่ได้จะถูกแสดงบนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้



รูปที่ 4 แบบจำลองการทำงานโดยรวมของ NSIM

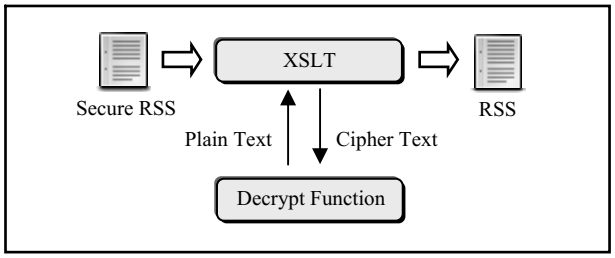


รูปที่ 5 ส่วนรวบรวม RSS feed (NSIMAg)



รูปที่ 6 ส่วนสร้าง RSS feed สำหรับผู้ใช้ (GRSS4U)

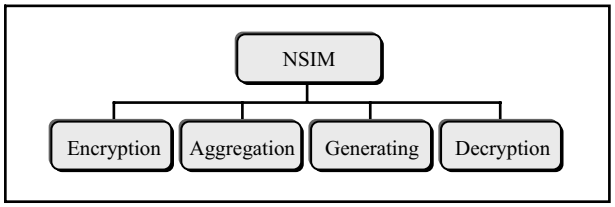
กระบวนการถอดรหัสดังแสดงในรูปที่ 7 ใช้ไวยากรณ์ของ XPath ที่ระบุไว้ใน XSLT เพื่อค้นหาข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้ โดยส่งข้อความไซเฟอร์ไปยังฟังก์ชันการถอดรหัส ได้เป็นข้อความต้นฉบับ และแทนที่ลงในเอกสาร RSS feed ก่อนถูกแสดงผล



รูปที่ 7 กระบวนการถอดรหัสข้อมูลข่าวสารส่วนบุคคล

4. การพัฒนาระบบและผลลัพธ์

การพัฒนาระบบใช้ภาษา PHP และ JavaScript ทดสอบการทำงานด้วย Internet Explorer บน Windows Mobile 5 Emulator โดยแบ่งการทำงานออกเป็น 4 โมดูล แสดงดังรูปที่ 8



รูปที่ 8 โมดูลการทำงานของ NSIM

4.1 การเข้ารหัสข้อมูล RSS feed

การเข้ารหัสข้อมูลภายในเอกสาร RSS feed นั้น องค์กรต่าง ๆ จะได้รับกุญแจที่ใช้ในการเข้ารหัสด้วยอัลกอริทึม RSA จากระบบ โดยกำหนดให้เอกสาร RSS feed ที่สร้างขึ้นต้องระบุ Username ของผู้ใช้ที่ต้องการแจ้งข่าวสารไว้ในแอททริบิวต์ id บนแท็ก <description> ของ RSS item โดยมีรูปแบบดังแสดงในรูปที่ 9 และตัวอย่างแสดงดังรูปที่ 10 ซึ่งข้อมูลข่าวสารทั่วไปกำหนดให้แอททริบิวต์ id = “public”

<description id = “*Username*”>*Cipher* | *Message*</description>

รูปที่ 9 รูปแบบการกำหนดชื่อผู้ใช้

จากรูปที่ 9 กำหนดให้

Username	คือ ชื่อผู้ใช้
Cipher	คือ ข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล
Message	คือ ข้อมูลข่าวสารทั่วไป

<description id = “Bob”>004073eb 0031e994 000a9a00 000f9e24 00213748 0057f693 000d013e 0023e9bd 001a28e4 0011b8e2</description>

รูปที่ 10 เข้ารหัสข้อมูลข่าวสารส่วนบุคคลและระบุชื่อผู้ใช้

4.2 การรวบรวม RSS feed

ระบบรวบรวมเอกสาร RSS feed จากองค์กรต่าง ๆ ที่ได้ลงทะเบียน URL ของเว็บไซต์ไว้ และ Parse ข้อมูลเก็บไว้ในฐานข้อมูล ซึ่งประกอบด้วยข้อมูลส่วนต่าง ๆ ของเอกสาร RSS feed และ Username ที่ได้จากแอททริบิวต์ id ในแท็ก <description> ของ RSS item โดยกำหนดให้มีการปรับปรุงฐานข้อมูลของระบบอัตโนมัติทุก 3 ชั่วโมง เมื่อมีผู้เข้าใช้งาน

เนื่องจากมีบางองค์กรไม่ได้จัดเตรียมเอกสาร RSS feed ไว้ ดังนั้นระบบจึงออกแบบเพื่อให้สามารถสร้างเอกสาร RSS feed อัตโนมัติด้วยวิธีการค้นหาแท็ก HTML ต่าง ๆ ภายในหน้าเว็บเพจ โดยใช้ Regular Expression กำหนดให้ส่วน title ใน RSS channel สกัดจากแท็ก <title> ของ HTML และ URL ของเว็บเพจจะถูกนำมาใส่ไว้ใน link ของ RSS channel หากมี Metadata ใน HTML ที่เกี่ยวกับ description ก็จะถูกนำมาใส่ไว้ใน description ของ RSS channel ซึ่งคล้ายกับวิธีการใน [15] โดยส่วนของ RSS item จะพิจารณาจากแท็ก <a> ภายในหน้าเว็บเพจ เพื่อนำมาสร้างเป็นส่วน <title> และ <link> ในแต่ละ item

4.3 การสร้าง RSS feed สำหรับผู้ใช้

เมื่อผู้ใช้เข้าใช้งาน ระบบจะทำการตรวจสอบไฟล์ข้อมูลส่วนตัวของผู้ใช้ และนำ Username ที่ได้ไปสืบค้นข้อมูลข่าวสารส่วนบุคคล พร้อมกับข้อมูลข่าวสารทั่วไป ด้วยการสอบถามไปยังฐานข้อมูลข่าวสาร จากนั้นจึงสร้างเป็นเอกสาร RSS feed โดยมีรูปแบบดังแสดงในรูปที่ 11 และตัวอย่างแสดงดังรูปที่ 12

<description class = “cipher”>*Cipher Key*:*Cipher Mesg*</description>
หรือ
<description class = “mesg”>*Message*</description>

รูปที่ 11 รูปแบบแท็ก <description> ของ RSS item

จากรูปที่ 11 กำหนดให้

class = “cipher”	คือ รายละเอียดสำหรับข้อมูลข่าวสารส่วนบุคคล
class = “mesg”	คือ รายละเอียดสำหรับข้อมูลข่าวสารทั่วไป
Cipher Key	คือ ข้อความไซเฟอร์ของกุญแจถอดรหัส
Cipher Mesg	คือ ข้อความไซเฟอร์ของข้อมูลข่าวสารส่วนบุคคล
Message	คือ ข้อมูลข่าวสารทั่วไป

```
1  <item>
2  <title>Private News</title>
3  <link>http://sis.psu.ac.th/</link>
4  <description class="cipher">3331 3362 6239 323a 6664
   3638 3a33 3637 6334 3936:004073eb 0031e994 000a9a00
   000f9e24 00213748 0057f693 000d013e 0023e9bd 001a28e4
   0011b8e2 </description>
5  </item>
6  <item>
7  <title>Public News</title>
8  <link>http://www.psu.ac.th </link>
9  <description class="mesg">Public News</description>
10 </item>
```

รูปที่ 12 ตัวอย่าง RSS item สำหรับผู้ใช้

4.4 การถอดรหัสข้อมูล RSS feed

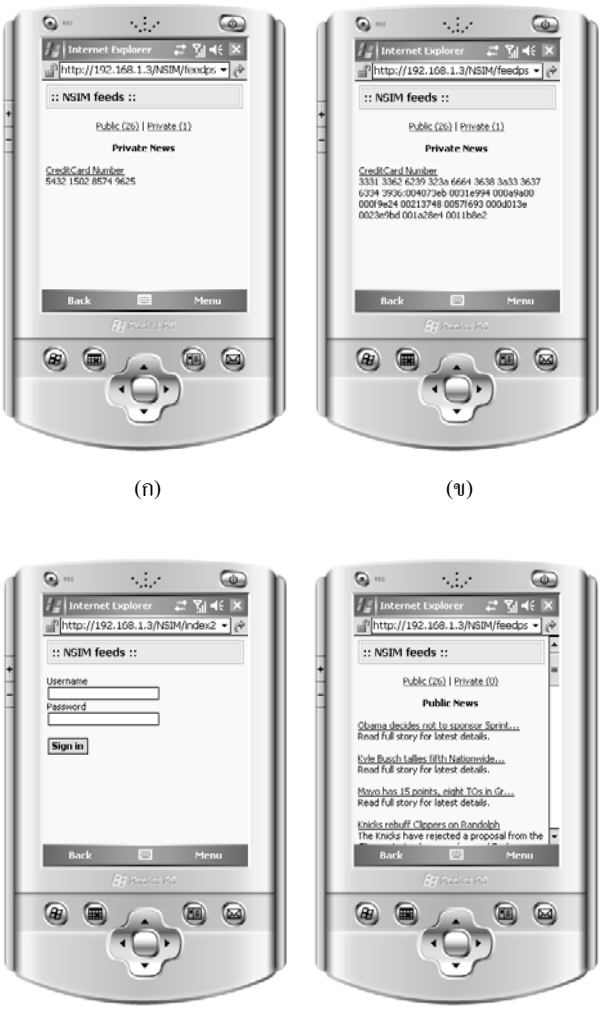
การถอดรหัสข้อมูลใช้ DOM และ XSLT ดังแสดงในรูปที่ 13 เพื่อค้นหาข้อมูลที่เข้ารหัสไว้ โดยพิจารณาจากแอททริบิวต์ class = “cipher” ในแท็ก <description> จากนั้นนำข้อความไซเฟอร์และกุญแจถอดรหัส ที่ได้จากไฟล์ข้อมูลส่วนตัวผู้ใช้ ส่งไปยังฟังก์ชันการถอดรหัสที่พัฒนาด้วย JavaScript ตามรูปแบบคำสั่งที่ระบุไว้ในเอกสาร XSLT โดยตัวอย่างข้อมูลที่ถูกเข้ารหัสไว้แสดงดังรูปที่ 12 ในบรรทัดที่ 4

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
 <xsl:for-each select="rss/channel/item">
    <xsl:element name="a">
      <xsl:attribute name="href">
           <xsl:value-of select="link" />
      </xsl:attribute>
      <xsl:value-of select="title" />
   </xsl:element>
   <xsl:choose>
     <xsl:when test="description/@class='cipher'">
      <script language="JavaScript1.2">
        mess=decrypt("<xsl:value-of select="description " />");
        document.write(mess);
      </script>
     </xsl:when>
    <xsl:when test="description/@class='mesg'">
        <xsl:value-of select="description "/>
    </xsl:when>
   </xsl:choose>
 </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

รูปที่ 13 ตัวอย่าง XSLT สำหรับถอดรหัส RSS feed

4.5 ผลลัพธ์จากการทดลอง

ทดสอบการทำงานโดยดึงเอกสาร RSS feed จากเว็บข่าว CNN [16] สำหรับข้อมูลข่าวสารทั่วไป และเว็บข่าวทดสอบที่ได้จัดเตรียม RSS feed ที่ปลอดภัยเพื่อแจ้งข้อมูลบัตรเครดิตแก่ผู้ใช้บริการ ผลลัพธ์ที่ได้แสดงดังรูปที่ 14 (ก) และ (ข) เพื่อเปรียบเทียบในกรณีที่มีและไม่มีการถอดรหัสข้อมูลข่าวสารส่วนบุคคล (Private News) ตามลำดับ และหากผู้ใช้ยังไม่ได้ลงทะเบียนเพื่อรับข้อมูลข่าวสารจะปรากฏหน้าต่างดังแสดงในรูปที่ 14 (ค) เพื่อให้ผู้ใช้ลงทะเบียนแล้วจึงแสดงข้อมูลข่าวสารทั่วไป (Public News) ดังรูปที่ 14 (ง)



(ก) (ข)

(ค) (ง)

รูปที่ 14 ผลลัพธ์จากการทดลอง

(ก) กรณีถอดรหัสข้อมูลข่าวสารส่วนบุคคลเรียบร้อยแล้ว

(ข) กรณีไม่ได้ถอดรหัสข้อมูลข่าวสารส่วนบุคคล

(ค) ลงทะเบียนรับข้อมูลข่าวสาร

(ง) ข่าวสารทั่วไปที่ได้จากการรวบรวมเอกสาร RSS feed

จากผลการทดลองเห็นได้ว่า ระบบสามารถรวบรวม RSS feed และเผยแพร่ข้อมูลข่าวสารไปยังผู้ใช้เฉพาะบุคคลได้อย่างถูกต้อง โดยข้อมูลข่าวสารส่วนบุคคลถูกทำให้ปลอดภัยด้วยการเข้ารหัสก่อนส่งไปยังผู้ใช้ เมื่อฝั่งผู้ใช้ได้รับข้อมูลจะทำการถอดรหัสข้อความและแสดงผลบนอุปกรณ์สื่อสารเคลื่อนที่ จึงนับได้ว่าระบบสามารถทำให้ผู้ใช้ได้รับข้อมูลข่าวสารที่ทันสมัยและมีความปลอดภัย นอกจากนี้เพื่อให้ผู้ใช้รับทราบข้อมูลข่าวสารอย่างรวดเร็ว เมื่อผู้ให้บริการมีการปรับปรุงข้อมูล จึงจำเป็นต้องพัฒนาระบบให้สามารถวิเคราะห์เวลาที่เหมาะสมในการดึงเอกสาร RSS feed แบบอัตโนมัติ ซึ่งส่วนนี้กำลังอยู่ในขั้นตอนการพัฒนาและปรับปรุง

5. บทสรุป

บทความนี้นำเสนอกลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่ ทำให้องค์กรต่าง ๆ สามารถแจ้งข้อมูลข่าวสารไปยังผู้ที่เกี่ยวข้องได้อย่างปลอดภัย โดยประยุกต์ใช้เทคโนโลยี RSS ร่วมกับการเข้ารหัสลับ ด้วยอัลกอริทึม RSA เพื่อทำให้เอกสาร RSS feed มีความปลอดภัยก่อนเผยแพร่ไปยังผู้ที่เกี่ยวข้อง และได้ใช้ความสามารถของ DOM และ XSLT สืบค้นข้อมูลข่าวสารส่วนบุคคลที่ถูกเข้ารหัสไว้ โดยแทรกสคริปต์สำหรับสืบค้นข้อมูลดังกล่าวไว้กับ XSLT จากนั้นนำข้อมูลที่ได้จากการสืบค้นไปผ่านกระบวนการถอดรหัส แล้วจึงแสดงผลลัพธ์บนอุปกรณ์สื่อสารเคลื่อนที่ของผู้ใช้

เอกสารอ้างอิง

[1] E. Finkelstein, *Syndicating Web Sites with RSS Feeds For Dummies*, Wiley Publishing, Inc., Hoboken, NJ, 2005.

[2] W3C, “Extensible Markup Language (XML)”, July 2008, [Online]. Available: http://www.w3.org/XML/ [May 28, 2008].

[3] W3C, “XML Encryption Syntax and Processing”, December 2002, [Online]. Available: http://www.w3.org/TR/xmlenc-core/ [May 28, 2008].

[4] W3C, “XML Signature Syntax and Processing (Second Edition)”, June 2008, [Online]. Available: http://www.w3.org/TR/xmldsig-core/ [May 28, 2008].

[5] L. Dykes and E. Tittel, *XML For Dummies 4th Edition*, Wiley Publishing, Inc., Hoboken, NJ, 2005.

[6] W3C, “XML Path Language (XPath) Version 1.0”, November 1999, [Online]. Available: http://www.w3.org/TR/xpath [May 28, 2008].

[7] W3Schools, “RSS Tutorial”, [Online]. Available: http://www.w3schools.com/rss/ [April 20, 2008].

[8] A. Kahate, *Cryptography and Network Security*, McGraw-Hill Publishing Company, 2003.

[9] R. J. Glotzbach, J. L. Mohler and J. E. Radwan, “RSS as a Course Information Delivery Method”, International Conference on Computer Graphics and Interactive Techniques, San Diego, California, August 05-09, 2007.

[10] S. J. Cold, “Using Really Simple Syndication (RSS) to Enhance Student Research”, *ACM SIGITE Newsletter*, Vol.3, No.1, January, 2006, pp. 6-9.

[11] H. Maruyama and T. Imamura, “Element-wise XML Encryption”, April 2000, [Online]. Available: http://lists.w3.org/Archives/Public/xml-encryption/2000Apr/att-0005/01-xmlenc [January 5, 2008].

[12] R. G. Bartlett and M. W. Cook, “XML Security Using XSLT”, 36th Hawaii International Conference on System Sciences (HICSS’03), IEEE, 2002.

[13] T. Chang and G. Hwang, “The design and implementation of an application program interface for securing XML documents”, *The Journal Systems and Software*, August, 2007, pp. 1362–1374.

[14] J. Gregorio, “Secure RSS Syndication”, July 2005, [Online]. Available: http://www.xml.com/pub/a/2005/07/13/secure-rss.html.

[15] J. Wang, K. Uchino, T. Takahashi and S. Okamoto, “RSS Feed Generation from Legacy HTML Pages”, APWeb, LNCS 3841, 2006, pp. 1071–1082.

[16] CNN.com [Online]. Available: http://rss.cnn.com/rss/si_topstories.rss [July 15, 2008].

ประวัติผู้เขียนบทความ

วิชุตา แก้วนพรัตน์ นักศึกษาปริญญาโท ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ งานวิจัยที่สนใจ ได้แก่ RSS Technology, Web Intelligent, Information Retrieval

ดร.ลัดดา ปรีชาวีรกุล อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ งานวิจัยที่สนใจ ได้แก่ Internet Computing, Image and Video Retrieval, Information Security

ภาคผนวก ข

ผลงานตีพิมพ์

เรื่อง	A Novel Approach: Secure Information Notifying System using RSS Technology
งานประชุมวิชาการ	2009 International Conference on Future Networks (ICFN 2009)
สถานที่	กรุงเทพฯ ประเทศไทย
วันที่	7 - 9 มีนาคม 2552

2009 International Conference on Future Networks

A Novel Approach: Secure Information Notifying System using RSS Technology

Ladda Preechaveerakul and Wichuta Kaewnopparat
Computer Science Department
Faculty of Science, Prince of Songkla University
Hat Yai, Songkhla, Thailand
e-mail: {ladda.p, s5010220120}@psu.ac.th

***Abstract*—Internet affects our daily lives; people use it to contact each other. Many organizations also make use of it to supply information both general and sensitive information such as credit card information, financial business information to users as well as in using mobile devices. Really Simple Syndication (RSS) is also used to distribute the latest information over the Internet. However, the RSS technology does not have a mechanism to ensure that the incoming information is really secure. Therefore, we developed a Secure Information Notifying System with RSS Technology for Mobile Users (SInfoNS).**

Keywords-Mobile device; Really Simple Syndication; RSS

I. INTRODUCTION

RSS [1] is an XML-based information technology widely used in Web log and news web sites for disseminating the latest news. Several companies have started to use RSS for spreading their information to customers. However, most contents published via RSS technology are public, not confidential. Therefore, an RSS secure mechanism to send private information is now needed. Although, XML [2] has an XML encryption [3, 4] for sensitive data, a limitation of syntax in the RSS feed document does not match with confidential information.

In this paper, we propose a system that aggregates both general and sensitive information in organizations by using RSS feed documents and transmit those to relevant users. The system which gives the incoming information to a user will be secure by encrypting some parts of an RSS document, especially private information. Decryption occurs when a mobile user needs to see his or her private information.

This paper gives the background and related work in Section 2. In Section 3, we describe how a Secure Information Notifying System with RSS Technology for Mobile Users (SInfoNS) works. The implementation and some results are shown in Section 4. Section 5 presents our conclusions and future work.

II. BACKGROUND AND RELATED WORK

A. Extensible Markup Language (XML)

XML provides an easy way to manage and share information via the Internet. An XML document is accessed and manipulated by one of two types: Document Object Model (DOM) and Simple API for XML (SAX). Since XML was designed to carry data not to display data, then XSL [5] is used to turn the XML document into required form. At present, there are already several specific markup languages called XML applications, which we can use within our documents.

B. RSS Technology

RSS or Really Simple Syndication is an XML application used to publish frequently updated contents such as blog entries, news headlines, etc. in a standardized format. RSS automatically aggregates contents from multiple web sites in one place as well as avoids piracy. Its content can be read using an RSS reader or an aggregator, which can be web-based or at the software reader. The RSS current version is 2.0. Its general structure [6] is shown in Fig. 1.

In Fig. 1, the first line in the document - the XML declaration - defines the XML version. The second line is the RSS declaration, which identifies an RSS document (in this case, RSS version 2.0). The next line contains the <channel> element used to describe the RSS feed. The <channel> element has three required child elements:

1. <title> - Defines the title of the channel
2. <link> - Defines the hyperlink to the channel
3. <description> - Describes the channel

```
<?xml version = "1.0" ?>
<rss version = "2.0" >
    <channel>
        <title>…</title>
        <link>…</link>
        <description>…</description>
        <item>
            <title>…</title>
            <link>…</link>
            <description>…</description>
            <pubDate>…</pubDate>
            ...
        </item>
        …
    </channel>
</rss>
```

Figure 1. General structure of RSS 2.0 document.

978-0-7695-3567-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ICFN.2009.35

Each <channel> element can also have one or more <item> elements. Table 1 shows some RSS item tags.

TABLE I. EXAMPLE OF SOME RSS ITEM TAGS

RSS item tag	Description
<title>	Defines the title of the item
<link>	Defines the hyperlink to the item
<description>	Describes the item
<pubDate>	Defines the last publication date for the item
<category>	Defines one or more categories the item belongs to

C. Security

The rapid growth and widespread use of electronic data processing through the Internet fuels the need for protecting the information they store, process and transmit. There have been many examples of insufficient security in applications developed for the Internet. Information exchange through the Internet should be focused on four core principles of security: confidentiality, integrity, authentication and non-repudiation. Thus, the main technique for securing information is to apply cryptography.

Cryptography is a tool to encrypt and decrypt data. Every encryption and decryption process has two aspects: the algorithm and the key. In essence, a sender encrypts his message with the algorithm and the key, get the result as cipher text, and send it to a recipient. The recipient decrypts an incoming cipher text with the algorithm and the key and restores original message. The popular techniques in cryptography are symmetric cryptography and asymmetric cryptography. DES and RSA [7] are the examples of symmetric and asymmetric cryptography, respectively.

D. Related Work

RSS technology is widely used for news web sites and web log. However, there are applied researches using RSS. For example, Glotzbach et al. [8] applied RSS to provide students with a method of receiving course announcement. Cold [9] used RSS to enhance research methods for students by gleaning current information from online journal, publications, web logs and other sources without visiting the sites daily.

Maruyama et al. [10] proposed a subtree Element-Wise encryption that encrypts some part of XML document for security. Barrett and Cook [11] presented XML Security using XSLT regarded encryption and decryption as another XML document transformation operation. Chang and Wang [12] devised an application program interface for securing XML documents called Document Security Language (DSL). In addition, Gregorio [13] applied Greasemonkey (a Mozilla Firefox extension) to decrypt RSS feed documents by writing a symmetric key algorithm. However, this approach also works well with Mozilla Firefox browser.

As mentioned above, we found that RSS technology is applied for several applications. However, most researches do not propose securing RSS feed documents. Therefore, we propose a Secure Information Notifying System with RSS Technology for Mobile Users (SInfoNS).

III. SECURE INFORMATION NOTIFYING SYSTEM WITH RSS TECHONOLOGY FOR MOBILE USERS

The SInfoNS aims to enable each organization to notify both public and private information as well as receive updated information on mobile devices.

A. Design Concept

The SInfoNS was designed under the following requirements:

1. The system should be able to aggregate and disseminate both public and private information.
2. Users should ensure that their private information is being secure.
3. Users should be ensured that they will receive the latest information.

B. SInfoNS Model

The SInfoNS is a client/server model with 2 main components: SInfoNS Aggregator (SInfoNSA) and SInfoNS Generator (SInfoNSG). Fig. 2 illustrates the SInfoNS Model.

The SInfoNSA aggregates RSS feed from organizations of which each organization may have both public and private information, as shown in Fig. 3 with the following operations:

1. SInfoNSA aggregates RSS feed both for public and private information from organizations.
2. The XML parser reads an RSS document, identifies all the RSS tags, validates it and stores the data to the SInfoNS database. Since SInfoNS was designed to ensure the users' private information is being secure and RSS technology does not match, the XML schema definition has been created for validation.

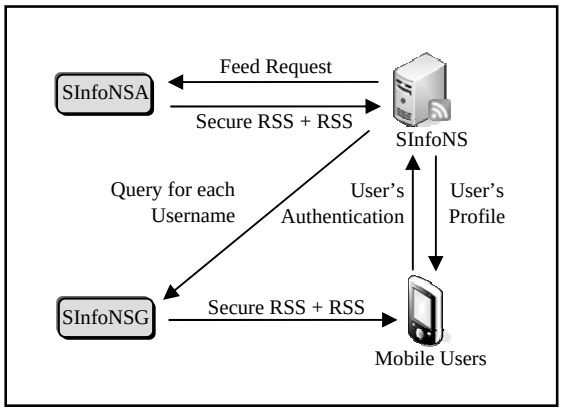


Figure 2. The SInfoNS Model.

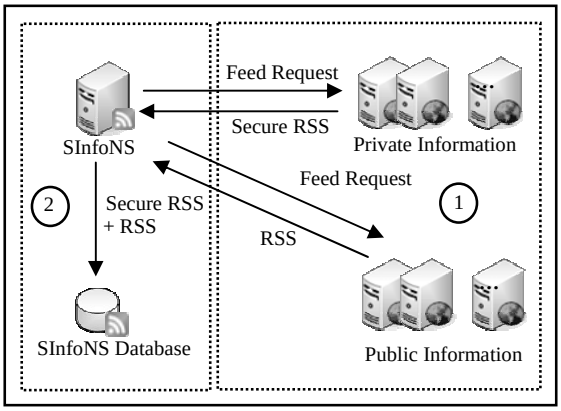


Figure 3. The SInfoNS Aggregator (SInfoNSA).

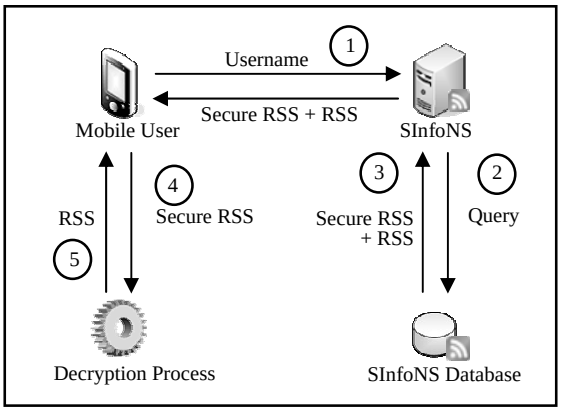


Figure 4. The SInfoNS Generator (SInfoNSG).

The SInfoNSG generates RSS feed for each user. The SInfoNSG will activate when a user accesses it with his or her username and password given by the organization. Then, SInfoNS creates a user's profile on his mobile device, and generate a RSS feed document for that user as shown in Fig. 4. The process to obtain the RSS feed document runs as follows:

1. The SInfoNS verifies the username from the client's profile on the user's mobile device for authentication.

2. The SInfoNS queries the SInfoNS database for retrieving a user's latest RSS feed.

3. The SInfoNS generates the user's RSS feed documents which are public and private information and send it to that user.

4. In case of private information, the decryption process in Fig. 5 is invoked.

5. When finishing the decryption of the information, the result will be displayed on the user's mobile device.

The decryption process as illustrated in Fig. 5 uses XPath [14] defined in XSLT to search a user's cipher text, decrypts and replaces it in the RSS feed document before displaying the result.

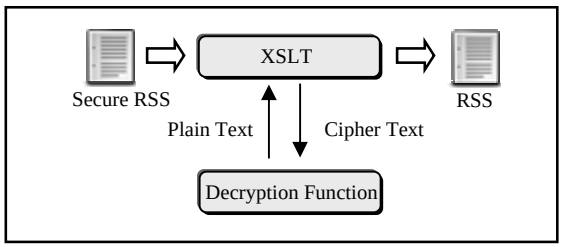


Figure 5. The decryption process for private information.

C. *Secure Information Notifying Algorithm*

After RSS feeds from multiple sources have been retrieved, the information in XML tags can be extracted to be stored in the SInfoNS database, for example, <title>, <description>, <link>. The secure information notifying algorithm is shown in Fig. 6.

D. *SInfoNS Database*

The SInfoNS database contains the information from RSS feeds that are necessary for users. Fig. 7 shows the design of database. Each RSS feed has one feedurl, each of which contains any number of items. In addition, one organization has at least one url and each user has any number of items.

IV. IMPLEMENTATION

The system is developed for the use of PHP and JavaScript. The result is shown in Internet Explorer on Windows Mobile 5 Emulator. The main function of SInfoNS can be divided into 4 modules: Encryption, Aggregation, Generating, and Decryption module.

```
1   Method SInfoNT (feedUrl, PubInfo, SecInfo)
2     for each feedUrl
3       download feed content (PubInfo and SecInfo)
4       from feedUrl
5       for each item in the feed
6         get data in <title>, <description> <link> and
7               <username>
8         store in SInfoNS database
9       end for
10    end for
11    if (search username in SInfoNS database to
12      generate RSS feed )
13      if (SecInfo)
14        class = "cipher"
15      else
16        class = "mesg"
17      end if
18    end if
19    send RSS feed to user
20    if (class = "cipher")
21      Call decryption function
22    end if
23    show content to user
24  end method
```

Figure 6. The secure information notifying algorithm.

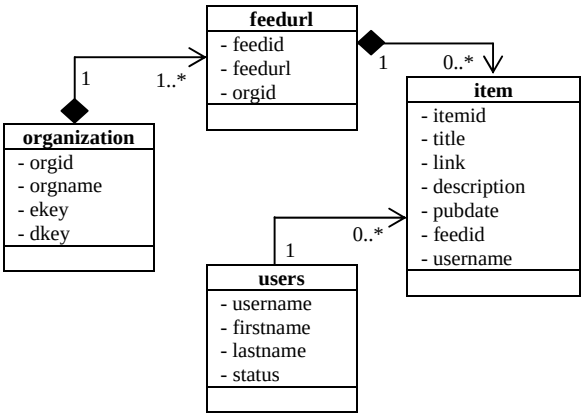


Figure 7. SInfoNS database structure.

A. *Aggregation Module*

The SInfoNS aggregates RSS feeds from organizations which subscribe their URL and stores data in the SInfoNS database. In case an organization do not prepare an RSS feed document, the system automatically creates an RSS feed by using a regular expression to extract the content from the HTML tag such as *title*, *url*, *meta*, etc. and put those in the RSS channel. In addition, tag <a> in the web page is used to be parts of tag <title> and <link> in RSS item.

B. *Generating Module*

When a user accesses the system, a user's profile is invoked to retrieve his private information as well as his public information by querying the SInfoNS database and creating an RSS feed document as shown in Fig. 8 and Fig. 9. In Fig. 8, there are two classes for defining type of information; class = "cipher" defines for private information, and class = "mesg" defines for public information. The *Cipher Key* is decryption key and the message to send from the user may be *Cipher Mesg* for cipher text or *Message* for general information.

C. *Decryption Module*

We use DOM together with XSLT for the decryption process. Fig. 10 describes how to search encrypted information from an attribute: class = "cipher" in <description>. The system then sends a cipher text with the decryption key, obtained from the Generating module, to the decryption function written by JavaScript.

```
<description class = "cipher">Cipher Key:Cipher Mesg
</description>
Or
<description class = "mesg">Message</description>
```

Figure 8. RSS item: <description> format.

```
<item>
<title>Private News</title>
<link>http://sis.psu.ac.th/</link>
<description class="cipher">3331 3362 6239 323a 6664
3638 3a33 3637 6334 3936:004073eb 0031e994 000a9a00
000f9e24 00213748 0057f693 000d013e 0023e9bd 001a28e4
0011b8e2 </description>
</item>
<item>
<title>Public News</title>
<link>http://www.psu.ac.th </link>
<description class="mesg">Public News</description>
</item>
```

Figure 9. Example of RSS item for a user.

```
<xsl:choose>
<xsl:when test="description/@class='cipher'">
<script language="JavaScript1.2">
mess=decrypt("<xsl:value-of select="description " />");
document.write(mess);
</script>
</xsl:when>
<xsl:when test="description/@class='mesg'">
<xsl:value-of select="description "/>
</xsl:when>
</xsl:choose>
```

Figure 10. XSLT example for RSS feed decryption.

D. *Experimental Results*

We implemented the SInfoNS via feeding general news from CNN website [15] and generated a secure web site to notify credit card information to users. The results in Fig. 11 a) and b) show private information with and without decryption.

a) b)

Figure 11. Private information with and without decryption.

V. CONCLUSIONS AND FUTURE WORK

The proposed SInfoNS helps organizations to ensure that information to each user, especially sensitive information, will be secure. We applied the RSS technology together with the cryptography algorithm to make any RSS feed document being secure before to disseminate it to relevant users. The SInfoNS also uses DOM and XSLT to apply for private information retrieval written in JavaScript. The results displayed on a user's mobile device give users, the latest information.

The experimental results confirm that our system is able to aggregate RSS feed and disseminate information to each user. The SInfoNS fulfills RSS technology for private information which can be distributed securely by using the RSA as an example of cryptography algorithm. The system is designed to support TCP/IP-based users, thus not only for mobile users, but also Internet-based users. We consider the appropriate time for feeding RSS documents automatically to users, so this will be our future work.

REFERENCES

[1] E. Finkelstein, Syndicating Web Sites with RSS Feeds for Dummies, Wiley Publishing, Inc., Hoboken, NJ, 2005.

[2] W3C, "Extensible Markup Language (XML)", July 2008. Available from: http://www.w3.org/XML/ [May 28, 2008].

[3] W3C, "XML Encryption Syntax and Processing", December 2002. Available from: http://www.w3.org/TR/xmlenc-core/ [May 28, 2008].

[4] W3C, "XML Signature Syntax and Processing (Second Edition)", June 2008. Available from: http://www.w3.org/TR/xmldsig-core/ [May 28, 2008].

[5] L. Dykes and E. Tittel, XML For Dummies 4th Edition, Wiley Publishing, Inc., Hoboken, NJ, 2005.

[6] W3Schools, "RSS Tutorial". Available from: http://www. w3schools.com/rss/ [April 20, 2008].

[7] A. Kahate, Cryptography and Network Security, McGraw-Hill Publishing Company, 2003.

[8] R. J. Glotzbach, J. L. Mohler and J. E. Radwan, "RSS as a Course Information Delivery Method", International Conference on Computer Graphics and Interactive Techniques, San Diego, California, August 05-09, 2007.

[9] S. J. Cold, "Using Really Simple Syndication (RSS) to Enhance Student Research", ACM SIGITE Newsletter, Vol.3, No.1, January, 2006, pp. 6-9.

[10] H. Maruyama and T. Imamura, " Element-wise XML Encryption", April 2000. Available from: http://lists.w3.org/Archives/ Public/xml-encryption/2000Apr/att-0005/01-xmlenc [January 5, 2008].

[11] R. G. Bartlett and M. W. Cook, "XML Security Using XSLT", The 36th Hawaii International Conference on System Sciences (HICSS'03), IEEE, 2002.

[12] T. Chang and G. Hwang, "The design and implementation of an application program interface for securing XML documents", The Journal Systems and Software, August, 2007, pp. 1362–1374.

[13] J. Gregorio, "Secure RSS Syndication", July 2005. Available from: http://www.xml.com/pub/a/2005/07/13/secure-rss.html [May 28, 2008].

[14] W3C, "XML Path Language (XPath) Version 1.0", November 1999. Available from: http://www.w3.org/ TR/xpath [May 28, 2008].

[15] CNN. Available from: http://rss.cnn.com/rss/si_top stories.rss [July 15, 2008].

ประวัติผู้เขียน

ชื่อ สกุล นางสาววิชุตา แก้วนพรัตน์

รหัสประจำตัวนักศึกษา 5010220120

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
บธ.บ. (ระบบสารสนเทศ)	สถาบันเทคโนโลยีราชมงคล	2546

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

1. ทุนมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร ประจำปีการศึกษา 2550 จากมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร
2. ทุนอุดหนุนระดับบัณฑิตศึกษา จากมหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี

ตำแหน่งและสถานที่ทำงาน

ตำแหน่ง นักวิทยาศาสตร์

สถานที่ทำงาน ศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี

การตีพิมพ์เผยแพร่ผลงาน

1. วิชุตา แก้วนพรัตน์ และ ลัดดา ปรีชาวีรกุล. 2551. กลไกแจ้งข้อมูลข่าวสารที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับผู้ใช้อุปกรณ์สื่อสารเคลื่อนที่. การประชุมวิชาการวิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 12 (NCSEC 2008). ชลบุรี, ประเทศไทย, 20-21 พฤศจิกายน 2551. หน้า 483-489.
2. Preechaveerakul, L., and Kaewnopparat, W. 2009. A Novel Approach: Secure Information Notifying System using RSS Technology. 2009 International Conference on Future Networks (ICFN 2009). Bangkok, Thailand, March 7–9, 2009. pp.95-99.