วิธีการสอบถามเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ

Querying Encrypted XML Document Approach for a Database-service-provider

ต่วนกัสฟี หะมะ

Tuankasfee Hama

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
2553

ชื่อวิทยานิพนธ์ วิธีการสอบถามเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ

ผู้เขียน นายต่วนกัสฟี หะมะ

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....

(ผู้ช่วยศาสตราจารย์ ดร. ศิริรัตน์ วณิชโยบล)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

…….....

(ดร.ลัดดา ปรีชาวีรกุล)

คณะกรรมการสอบ

……..ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุขุมาล กิติสิน)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล)

…….....กรรมการ

(ดร.ลัดดา ปรีชาวีรกุล)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. ภัทร อัยรักษ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

……..……

(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ วิธีการสอบถามเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ

ผู้เขียน นายต่วนกัสฟี หะมะ

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2552

บทคัดย่อ

ในปัจจุบันข้อมูลข่าวสารในโลกอินเตอร์เน็ตเพิ่มขึ้นเป็นอย่างมาก นอกจากนี้ยังมีการติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลก็เพิ่มขึ้นมากเช่นเดียวกัน จากเดิมที่มีการใช้งานเครื่องคอมพิวเตอร์ส่วนตัวก็ถูกเปลี่ยนมาเป็นการใช้งานผ่านผู้ให้บริการต่าง ๆ ในการจัดการฐานข้อมูลก็เช่นเดียวกันได้มีการเปลี่ยนแปลงมาเป็นการรับบริการจากผู้ให้บริการฐานข้อมูล (Database-service-provider) นอกจากนี้ XML ยังเป็นเอกสารที่ใช้เป็นตัวกลางในการแลกเปลี่ยนข้อมูล ปัญหาสำคัญในการใช้ XML ในการจัดเก็บข้อมูลคือ ความปลอดภัยในการจัดเก็บ และประสิทธิภาพในการค้นหาข้อมูล XML Encryption Standard เป็นมาตรฐานที่ได้ออกแบบมาเพื่อความปลอดภัยในการจัดเก็บข้อมูล แต่มีปัญหาในเรื่องประสิทธิภาพในการค้นหาข้อมูล ในงานวิจัยนี้ออกแบบวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ โดยที่ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย ซึ่งจากผลการทดลองพบว่าจำนวนกุญแจที่ใช้มีจำนวนน้อยกว่า เวลาที่ใช้ในการค้นหาข้อมูลน้อยกว่าวิธีการ XML Encryption Standard และวิธีการ SemCrypt นอกจากนี้ความปลอดภัยในการทำงานไม่แตกต่างกัน

Thesis Title	Querying Encrypted XML Document Approach for a Database-service-provider
Author	Mr. Tuankasfee Hama
Major Program	Computer Science
Academic Year	2009

ABSTRACT

At present, Internet grows rapidly. The number of exchanging information is also increased as well. In addition, many servers are used for information management such as database-service-provider for database management. The popular standard used for exchanging information is XML. However, major problems of using XML are security and efficiency of searching. XML Encryption Standard is a W3C the standard designed for security management, encrypting XML elements. However, it has trouble in searching.

In this Thesis, we design a method for storing and querying encrypted XML documents at a database-service-provider. The same plain text will be encrypted to different cipher texts to increase the safety. The experimental results showed that the number of encrypted keys and the query processing time used in our proposed method is less than the XML Encryption Standard and SemCrypt while maintaining the same security level.

สารบัญ

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

รายการตาราง

รายการตาราง (ต่อ)

ตาราง **หน้า**

รายการภาพประกอบ

รายการภาพประกอบ (ต่อ)

บทที่ 1

บทนำ

ในปัจจุบันมีการใช้งานเครือข่ายอินเทอร์เน็ตอย่างกว้างขวางและมีการใช้งานในหลากหลายรูปแบบ เช่น การใช้งานในการติดต่อสื่อสารแลกเปลี่ยนข้อมูล การใช้งานทางด้านการศึกษา การใช้งานด้านวิทยาศาสตร์ การใช้งานทางด้านธุรกิจ หรือแม้กระทั่งการใช้งานทางด้านการทหาร การใช้งานในเครือข่ายอินเทอร์เน็ตมักจะอยู่ในรูปแบบของ Client-Server ซึ่งในการติดต่อสื่อสารจะมีฝ่ายหนึ่งทำหน้าที่เป็นผู้รับบริการและอีกฝ่ายหนึ่งทำหน้าเป็นผู้ให้บริการ เนื่องจากผู้ให้บริการสามารถทำงานได้อย่างมีประสิทธิภาพและรวดเร็วกว่าผู้รับบริการทำให้ผู้รับบริการจะส่งข้อมูลไปดำเนินการในฝั่งของผู้ให้บริการเป็นส่วนใหญ่ เช่น การส่งข้อมูลไปประมวลผลเพื่อหาคำตอบ การฝากข้อมูลที่มีขนาดใหญ่ในฝั่งของผู้ให้บริการ หรือการส่งคำสั่งในการค้นหาไปยังผู้ให้บริการเพื่อค้นหาข้อมูลที่ต้องการ เป็นต้น

ผู้ให้บริการฐานข้อมูล (Database-service-provider) (Hacigumus *et al*., 2002) เป็นผู้ให้บริการที่มีประสิทธิภาพในการจัดการฐานข้อมูลไม่ว่าจะเป็นการจัดเก็บ การเพิ่ม ลบ การแก้ไข หรือแม้กระทั่งการค้นหาข้อมูลที่ต้องการ แต่การฝากข้อมูลไว้ในฝั่งของผู้ให้บริการเกิดปัญหาเกี่ยวกับการเปิดเผยรายละเอียดของข้อมูล ซึ่งข้อมูลบางชนิดเป็นข้อมูลที่มีความเปราะบางและเป็นข้อมูลที่เป็นความลับ ตัวอย่างเช่น ข้อมูลทางการเงิน ข้อมูลทางการแพทย์ หรือข้อมูลทางการทหาร เป็นต้น ดังนั้นจะมีวิธีการอย่างไรในการฝากข้อมูลที่มีความสำคัญไว้กับผู้ให้บริการที่ไม่น่าเชื่อถือ

รูปแบบการจัดเก็บข้อมูลที่ใช้กันในเครือข่ายอินเทอร์เน็ตมีอยู่หลากหลายรูปแบบ หนึ่งในนั่นคือ XML (W3C, 2009) ซึ่งเป็นมาตรฐานที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่อง XML มีรูปแบบเดียวกับภาษา HTML ใช้เป็นตัวกลางในการแลกเปลี่ยนข้อมูลระหว่างระบบต่างๆ ทำให้สะดวกในการทำงานร่วมกัน (Interoperability) ข้อดีอีกอย่างหนึ่งของ XML ที่แตกต่างจาก HTML คือ เป็นข้อมูลที่มนุษย์และเครื่องคอมพิวเตอร์สามารถทำความเข้าใจได้โดยง่าย นอกจากนี้ HTML เป็นภาษาที่เหมาะสำหรับการแสดงผล

วิธีการหนึ่งในการปกปิดเนื้อหาของข้อมูลคือ การเข้ารหัสข้อมูล ซึ่งการเข้ารหัสข้อมูลก็มีอยู่หลากหลายวิธี รูปแบบการเข้ารหัสข้อมูลสามารถแบ่งออกเป็น 2 รูปแบบคือ การเข้ารหัสข้อมูลทั้งหมด และการเข้ารหัสข้อมูลเพียงบางส่วน หากการเข้ารหัสข้อมูลทั้งหมดแล้วนำไปฝากไว้ที่ผู้ให้บริการก็จะทำให้สามารถปกปิดข้อมูลที่สำคัญได้ แต่ปัญหาที่ตามมาคือ การค้นหาข้อมูลจะทำได้ยากและใช้เวลานานเนื่องจากต้องดึงข้อมูลที่เข้ารหัสทั้งหมดกลับมาในฝั่ง

ของผู้รับบริการและทำการถอดรหัสข้อมูลทั้งหมดก่อนจะสามารถค้นหาข้อมูลได้ ในส่วนของการเข้ารหัสข้อมูลเพียงบางส่วนเหมาะกับกรณีที่ข้อมูลบางส่วนสามารถเปิดเผยได้แต่ข้อมูลอีกส่วนไม่สามารถเปิดเผยได้

ปัญหาอีกประการหนึ่งในการเข้ารหัสข้อมูลคือหากข้อมูลใดเป็นข้อมูลที่เหมือนกัน เมื่อมีการเข้ารหัสก็จะทำให้ได้ข้อมูลที่ถูกเข้ารหัสเหมือนกัน และเมื่อนำไปจัดเก็บในฝั่งของผู้ให้บริการก็จะทำให้ผู้ให้บริการทราบจำนวนข้อมูลที่เหมือนกัน มีการแก้ไขปัญหาด้วยการใช้วิธีการเข้ารหัสข้อมูลโดยใช้กุญแจที่แตกต่างกัน ซึ่งในกรณีที่มีข้อมูลซ้ำกันเป็นจำนวนมากก็ต้องใช้กุญแจเป็นจำนวนมากเช่นเดียวกัน ทำให้เพิ่มพื้นที่ในการจัดเก็บกุญแจ และเมื่อต้องการเข้าถึงข้อมูลที่ซ้ำกันจะต้องมีการส่งข้อมูลเพื่อไปสอบถามเท่ากับจำนวนข้อมูลที่ซ้ำกันทั้งหมด ดังนั้นในงานวิจัยนี้จึงมุ่งเน้นเสนอวิธีการจัดการเอกสาร XML ที่ถูกเข้ารหัสบางส่วน ซึ่งประกอบด้วยวิธีการจัดเก็บและวิธีการค้นหาที่มีประสิทธิภาพในการดำเนินการ โดยที่จัดเก็บข้อมูลไว้ในฝั่งของผู้ให้บริการและดำเนินการค้นหาในฝั่งของผู้ให้บริการ นอกจากนี้ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย

1.1 การตรวจเอกสาร

1.1.1 การติดต่อสื่อสาร

ไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) (Gallaugher, 1996) เป็นสถาปัตยกรรมหนึ่งที่ใช้ในเครือข่ายอินเตอร์เน็ตในปัจจุบัน โดยเซิร์ฟเวอร์หรือผู้ให้บริการจะทำหน้าที่ในการจัดการทรัพยากรต่างๆ รวมทั้งการจัดเก็บข้อมูล ซึ่งผู้ให้บริการจะปฏิบัติงานตามคำร้องขอของไคลเอนต์หรือผู้รับบริการ โดยเซิร์ฟเวอร์สามารถรองรับการทำงานร่วมกับผู้รับบริการหลายคนพร้อมกัน

หลักการทำงานของไคลเอนต์-เซิร์ฟเวอร์ คือ เมื่อผู้รับบริการต้องการข้อมูลหรือผลลัพธ์จากการคำนวณ ผู้รับบริการจะส่งคำร้องขอไปยังผู้ให้บริการ เมื่อผู้ให้บริการได้รับคำร้องขอก็จะดำเนินการประมวลผลคำร้องขอนั้นเพื่อหาผลลัพธ์ที่ผู้รับบริการต้องการ หลังจากที่ได้ผลลัพธ์แล้วผู้ให้บริการก็จะส่งผลลัพธ์กลับไปยังผู้รับบริการ ซึ่งการทำงานในลักษณะนี้เป็นเรื่องปกติที่เกิดขึ้นในเครือข่ายอินเตอร์เน็ต

ในส่วนของโพรโตคอล (Protocol) ที่ใช้ในการติดต่อสื่อสารในระบบเครือข่ายที่ใช้สถาปัตยกรรมแบบ Client-Server ที่สำคัญมีอยู่หลายโพรโตคอล ตัวอย่างเช่น HTTP, FTP และ DNS เป็นต้น

1.1.2 ภาษา XML (Extensible Markup Language)

ภาษา XML (W3C, 2009) เป็นภาษาที่ใช้ในการนิยามข้อมูลและมีลักษณะเป็นภาษา Markup ซึ่งถูกกำหนดเป็นมาตรฐานโดย W3C โดยที่ผู้ใช้สามารถกำหนดแท็ก (Tag) และอิลิเมนต์ (Element) ขึ้นมาเองได้และทำให้เข้าใจความหมายของข้อมูล ซึ่งแตกต่างจากภาษา HTML ที่ไม่สามารถกำหนดแท็กขึ้น นอกจากนี้แล้วสามารถใช้ในการแลกเปลี่ยนข้อมูลกับบุคคลอื่นได้โดยง่าย โดยในภาษา XML จะมีแท็กเปิดและแท็กปิดเช่นเดียวกับภาษา HTML

เอกสาร XML สามารถถูกนิยามด้วย DTD (Document Type Definition) หรือ XML Schema นอกจากนี้การเข้าถึงสามารถใช้ XML Parser ซึ่งแบ่งตามวิธีสำรวจเอกสารได้ 2 ชนิด คือ DOM (Document Object Model) และ SAX (Simple API for XML) การสืบค้นข้อมูลจะใช้ XPath เพื่อระบุตำแหน่งของข้อมูลที่ต้องการ และ XQuery ในการดึงข้อมูลลักษณะเดียวกับภาษา SQL

การรักษาความปลอดภัยของเอกสาร XML (Ekelhart, 2007) สามารถแบ่งออกเป็น 3 มาตรฐานสำคัญคือ XML Encryption, XML Signature และ XML Key Management โดยที่แต่ละมาตรฐานจะมีลักษณะการนำไปใช้ที่แตกต่างกัน โดยที่ทั้ง 3 มาตรฐานเป็นมาตรฐานที่ถูกกำหนดโดย W3C ในส่วนของ XML Encryption จะเป็นมาตรฐานที่ใช้สำหรับการเข้ารหัสข้อมูล และปกปิดเนื้อหาของข้อมูลในเอกสาร XML ในส่วนของ XML Signature จะใช้สำหรับการตรวจสอบว่าเอกสาร XML ที่ได้รับนั้นถูกแก้ไขก่อนที่จะมาถึงผู้รับหรือไม่ นอกจากนี้ยังใช้ในการตรวจสอบความถูกต้องของเอกสารและสามารถระบุผู้ส่งเอกสาร ในส่วนของ XML Key Management เป็นมาตรฐานที่ใช้ในการจัดการกุญแจที่ใช้ใน XML Encryption และ XML Signature โดยที่จะมีการรักษาความปลอดภัยในการแลกเปลี่ยนกุญแจ

1.1.3 ความปลอดภัย (Security)

ความปลอดภัยเป็นประเด็นสำคัญในการแลกเปลี่ยนข้อมูลระหว่างผู้รับกับผู้ส่ง เนื่องจากในปัจจุบันมีการแลกเปลี่ยนข้อมูลระหว่างระบบเครือข่ายเป็นจำนวนมาก ผลที่ตามมาก็คือมีผู้ที่ประสงค์ร้ายต่อข้อมูลและต่อการได้รับข้อมูลเพื่อมาใช้ประโยชน์โดยที่ตนเองไม่สิทธิในการเข้าถึงข้อมูลนั้น การที่จะทำให้ข้อมูลเกิดความปลอดภัยได้นั้นจำเป็นที่จะต้องมีพื้นฐานสำคัญในการรักษาความปลอดภัย ซึ่งประกอบด้วย การพิสูจน์ตัวตน (Authentication) การรักษาความลับ (Confidentiality) ความคงสภาพ (Integrity) และ การไม่ปฏิเสธแหล่งที่มา (Non-repudiation)

การรักษาความลับเป็นพื้นฐานสำคัญที่ใช้ในการปกปิดข้อมูล ซึ่งวิธีการนั่นคือการเข้ารหัสข้อมูลซึ่งสามารถแบ่งออกเป็น 2 ประเภทคือ การเข้ารหัสแบบกุญแจสมมาตร และการเข้ารหัสแบบกุญแจอสมมาตร

1.2 วัตถุประสงค์

ออกแบบ พัฒนา และทดสอบวิธีการจัดการเอกสาร XML ที่ถูกเข้ารหัสบางส่วน ซึ่งประกอบด้วยวิธีการจัดเก็บและวิธีการค้นหาที่มีประสิทธิภาพในการดำเนินการ โดยที่จัดเก็บข้อมูลไว้ในฝั่งของผู้ให้บริการและดำเนินการค้นหาในฝั่งของผู้ให้บริการ นอกจากนี้ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย

1.3 ขั้นตอนและระยะเวลาการดำเนินงาน

1.3.1 ขั้นตอนการดำเนินงาน

1) ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง ดังนี้
    1.1) การติดต่อสื่อสาร
    1.2) ภาษา XML
    1.3) ความปลอดภัย
    1.4) การเข้ารหัสเอกสาร XML
2) ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย
3) วิเคราะห์และออกแบบขั้นตอนวิธี
4) พัฒนาและทดสอบขั้นตอนวิธีตามที่ได้ออกแบบไว้
5) เขียนบทความวิจัย
6) จัดทำเอกสารวิทยานิพนธ์

1.3.2 ระยะเวลาการดำเนินงาน

พฤษภาคม 2552 - มีนาคม 2553

1.3.3 แผนการดำเนินการวิจัย

ระยะเวลาการดำเนินการวิจัยแสดงดังตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาการดำเนินการวิจัย

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน										
	2552								2553		
	5	6	7	8	9	10	11	12	1	2	3
1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง	←	—	—	—	—	—	—	→			
2. ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย		←	—	—	→						
3. วิเคราะห์และออกแบบกลไกการทำงาน		←	—	—	→						
4. พัฒนาและทดสอบกลไกการทำงาน			←	—	—	—	—	→			
5. เขียนบทความวิจัย			←	—	—	—	—	—	→		
6. จัดทำเอกสารวิทยานิพนธ์							←	—	—	—	→

1.4 ขอบเขตการดำเนินงาน

1.4.1 ออกแบบวิธีการจัดการเอกสาร XML ที่ถูกเข้ารหัสบางส่วน ซึ่งประกอบด้วยวิธีการจัดเก็บและวิธีการค้นหาที่มีประสิทธิภาพในการดำเนินการ โดยที่จัดเก็บข้อมูลไว้ในฝั่งของผู้ให้บริการและดำเนินการค้นหาในฝั่งของผู้ให้บริการ นอกจากนี้ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

1) ออกแบบวิธีการจัดเก็บข้อมูล

2) ออกแบบวิธีการในการค้นหาข้อมูล

1.4.2 พัฒนาและทดสอบวิธีการจัดการเอกสาร XML ที่ถูกเข้ารหัสบางส่วน ซึ่งประกอบด้วยวิธีการจัดเก็บและวิธีการค้นหาที่มีประสิทธิภาพในการดำเนินการ โดยที่จัดเก็บข้อมูลไว้ในฝั่งของผู้ให้บริการและดำเนินการค้นหาในฝั่งของผู้ให้บริการ นอกจากนี้ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย

1.5 สถานที่และเครื่องมือที่ใช้

1.5.1 สถานที่

ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการประยุกต์ (CS207) ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.5.2 เครื่องมือที่ใช้

1) ด้านฮาร์ดแวร์

เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผลกลางขนาด 3.00 GHz หน่วยความจำขนาด 2 GB และฮาร์ดดิสก์ความจุ 250 GB สำหรับใช้ในการพัฒนาและทดสอบระบบ

2) ด้านซอฟต์แวร์

2.1) ระบบปฏิบัติการ Microsoft Windows XP

2.2) ภาษาที่ใช้ในการพัฒนาระบบ Java

2.3) ระบบจัดการฐานข้อมูล MySQL

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการจัดการเอกสาร XML ที่ถูกเข้ารหัสบางส่วน ซึ่งประกอบด้วยวิธีการจัดเก็บและวิธีการค้นหาที่มีประสิทธิภาพในการดำเนินการ โดยที่จัดเก็บข้อมูลไว้ในฝั่งของผู้ให้บริการและดำเนินการค้นหาในฝั่งของผู้ให้บริการ นอกจากนี้ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

บทนี้จะกล่าวถึงทฤษฎีต่าง ๆ ที่ใช้ในการออกแบบและพัฒนาวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ ซึ่งประกอบด้วย การติดต่อสื่อสาร ภาษา XML (Extensible Markup Language) ความปลอดภัย (Security) และการเข้ารหัสเอกสาร XML (XML Encryption)

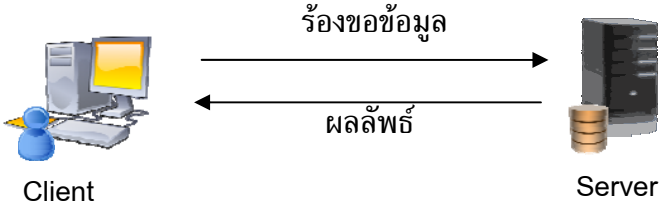
2.1 การติดต่อสื่อสาร

การให้บริการบนเครือข่ายคอมพิวเตอร์ มีวัตถุประสงค์เพื่อใช้ในการแลกเปลี่ยนข้อมูลร่วมกันระหว่างเครือข่าย โดยแบ่งการทำงานออกเป็น 2 ฝั่ง คือ เครื่องคอมพิวเตอร์ในฝั่งผู้รับบริการ (Client) และเครื่องคอมพิวเตอร์ในฝั่งผู้ให้บริการ (Server)

2.1.1 TCP/IP

การติดต่อสื่อสารในเครือข่ายอินเทอร์เน็ตในปัจจุบันจะใช้โปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นหลักในการติดต่อสื่อสาร ซึ่งมีความยืดหยุ่นในการในการทำงาน และสามารถแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่ายได้ นอกจากนี้ยังสามารถใช้ในการส่งข้อมูลได้หลากหลายรูปแบบไม่ว่าจะเป็นข้อความหรือแม้กระทั่งข้อมูลทางด้านมัลติมีเดีย เช่น เสียงหรือภาพ เป็นต้น (Socolofsky, 1991)

2.1.2 สถาปัตยกรรมไคลเอนต์-เซิร์ฟเวอร์ (Client-Server)



ภาพประกอบ 2.1 การติดต่อสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์

ไคลเอนต์-เซิร์ฟเวอร์ (Client-Server) (Gallaugher, 1996) เป็นสถาปัตยกรรมหนึ่งที่ใช้ในเครือข่ายอินเตอร์เน็ตในปัจจุบัน โดยเซิร์ฟเวอร์หรือผู้ให้บริการจะทำหน้าที่ในการจัดการทรัพยากรต่างๆ รวมทั้งการจัดเก็บข้อมูล ซึ่งผู้ให้บริการจะปฏิบัติงานตามคำร้องขอของไคลเอนต์หรือผู้รับบริการ โดยเซิร์ฟเวอร์สามารถรองรับการทำงานร่วมกับผู้รับบริการหลายคนพร้อมกัน

หลักการทำงานของไคลเอนต์-เซิร์ฟเวอร์ จะเป็นดังภาพประกอบ 2.1 เมื่อผู้รับบริการต้องการข้อมูลหรือผลลัพธ์จากการคำนวณ ผู้รับบริการจะส่งคำร้องขอไปยังผู้ให้บริการ เมื่อผู้ให้บริการได้รับคำร้องขอก็จะดำเนินการประมวลผลคำร้องขอนั้นเพื่อหาผลลัพธ์ที่ผู้รับบริการต้องการ หลังจากที่ได้ผลลัพธ์แล้วผู้ให้บริการก็จะส่งผลลัพธ์กลับไปยังผู้รับบริการ ซึ่งการทำงานในลักษณะนี้เป็นเรื่องปกติที่เกิดขึ้นในเครือข่ายอินเตอร์เน็ต

ในส่วนของโพรโตคอล (Protocol) ที่ใช้ในการติดต่อสื่อสารในระบบเครือข่ายที่ใช้สถาปัตยกรรมแบบ Client-Server ที่สำคัญมีอยู่หลายโพรโตคอล ตัวอย่างเช่น HTTP, FTP และ DNS เป็นต้น

2.1.3 Database-service-provider

เครือข่ายอินเทอร์เน็ตในปัจจุบันมีผู้ให้บริการอยู่หลากหลายรูปแบบ หนึ่งในรูปแบบที่ได้รับความนิยมคือ การให้บริการจัดการข้อมูล (Database-service-provider) (Hacigumus *et al*., 2002) รูปแบบของการให้บริการคือ

1) จัดเตรียมโปรแกรมในการจัดการฐานข้อมูลรองรับการทำงานของผู้รับบริการ
2) จัดเตรียมพื้นที่สำหรับรองรับข้อมูลของผู้รับบริการ
3) บริการแก้ไขข้อมูลตามที่ผู้รับบริการร้องขอ
4) บริการค้นหาข้อมูลตามคำร้องขอของผู้รับบริการ
5) สามารถใช้งานฐานข้อมูลตลอดเวลาและทุกสถานที่ตามที่ต้องการ

นอกจากนี้ประโยชน์ที่ได้รับจากการใช้บริการมีอยู่มากมายหลายประการ เช่น ประหยัดค่าใช้จ่ายในการจ้างผู้ดูแลระบบ สามารถเข้าถึงข้อมูลได้ตลอดเวลา และสามารถรองรับการทำงานของผู้รับบริการหลายคนพร้อมกันได้ เป็นต้น ในทางตรงข้ามการฝากข้อมูลไว้ที่ผู้ให้บริการก็อาจเกิดผลเสียในเรื่องความปลอดภัยของข้อมูล เนื่องจากผู้รับบริการนำข้อมูลของตนเองไปฝากไว้ที่ผู้ให้บริการ ซึ่งในกรณีที่เป็นข้อมูลสำคัญก็จะทำให้ผู้ให้บริการทราบรายละเอียดของข้อมูลและส่งผลเสียต่อผู้รับบริการ

2.2 ภาษา XML (Extensible Markup Language)

XML (W3C, 2009) เป็นมาตรฐานที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่อง นอกจากนี้แล้ว XML ยังมีรูปแบบเดียวกับภาษา HTML ที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างอินเตอร์เน็ต ทำให้สะดวกในการทำงานร่วมกัน ข้อดีอีกอย่างหนึ่งของ XML ที่แตกต่างจาก HTML คือ เป็นข้อมูลที่มนุษย์และเครื่องคอมพิวเตอร์สามารถทำความเข้าใจได้โดยง่าย

2.2.1 องค์ประกอบสำคัญของเอกสาร XML

W3C โดยที่ผู้ใช้สามารถกำหนดแท็ก (Tag) อิลิเมนต์ (Element) และแอททริ-บิวต์ (Attribute) ขึ้นมาเองได้และทำให้เข้าใจความหมายของข้อมูล ซึ่งแตกต่างจาก ภาษา HTML ที่ไม่สามารถกำหนดแท็กขึ้นเองได้

1) แท็ก (Tag) และอิลิเมนต์ (Element)

แท็ก และ อิลิเมนต์ เป็นส่วนประกอบสำคัญของภาษา XML โดยในภาษา XML ก็จะมีแท็กเปิดและแท็กปิดเช่นเดียวกับภาษา HTML โดยที่ผู้ใช้สามารถกำหนดชื่อแท็กได้เอง ตัวอย่างเช่น

```
<lastname>Bob</lastname>
```

ภาพประกอบ 2.2 ตัวอย่างอิลิเมนต์ของ XML

จากภาพประกอบ 2.2 แสดงส่วนที่เป็นแท็ก คือ **<lastname>** (แท็กเริ่มต้น) และ **</lastname>** (แท็กสิ้นสุด) ส่วน **<lastname>Bob</lastname>** จะเรียกว่าอิลิเมนต์

<u>ไวยากรณ์ที่เกี่ยวข้องกับอิลิเมนต์</u>

1. เอกสาร XML จะมีอิลิเมนต์ราก (Root element) ได้เพียงหนึ่งเดียวเท่านั้น ซึ่งจะครอบคลุมเอกสารทั้งหมด

2. แท็กเปิดและแท็กปิดจะต้องเหมือนกัน

3. ห้ามมีการซ้อนกันของแท็ก เช่น ภาพประกอบ 2.3 แสดงตัวอย่างการซ้อนกันของแท็กที่ถือว่าไม่ถูกต้อง

```
<patient><name>Bob </patient></name>
```

ภาพประกอบ 2.3 ตัวอย่างการซ้อนกันของแท็ก

4. ชื่อของแท็กมีคุณสมบัติเป็น case-sensitive กล่าวคือ ตัวอักษรตัวเล็กและตัวใหญ่มีความหมายที่แตกต่างกัน เช่น <patient> และ <Patient> ถือว่าแตกต่างกัน

5.แท็กข้อมูลที่เป็นค่าว่าง เขียนได้ 2 ลักษณะ คือ <patient/> และ <patient></patient>

6. ค่าของแอททริบิวต์จะต้องอยู่ในเครื่องหมาย “” หรือ ‘’ อย่างใดอย่างหนึ่งเท่านั้น

7. มีตัวอักขระสงวน 5 อักขระในภาษา XML ได้แก่ < & > “ และ ‘ ซึ่งในเอกสารที่จำเป็นที่จะต้องใช้เครื่องหมายเหล่านี้จะให้ใช้อักขระพิเศษแทน

< แทนด้วย <

& แทนด้วย &

> แทนด้วย >

“ แทนด้วย "

‘ แทนด้วย '

2) แอททริบิวต์ (Attribute)

แอททริบิวต์เป็นการระบุคุณสมบัติเพิ่มเติมให้กับอิลิเมนต์ บางเอกสารอาจไม่จำเป็นต้องใช้งานแอททริบิวต์ ตัวอย่างของการใช้แอทริบิวต์ เช่น

```
<medicine id=“1234”>AZT</medicine>
```

ภาพประกอบ 2.4 ตัวอย่างแอททริบิวต์ของ XML

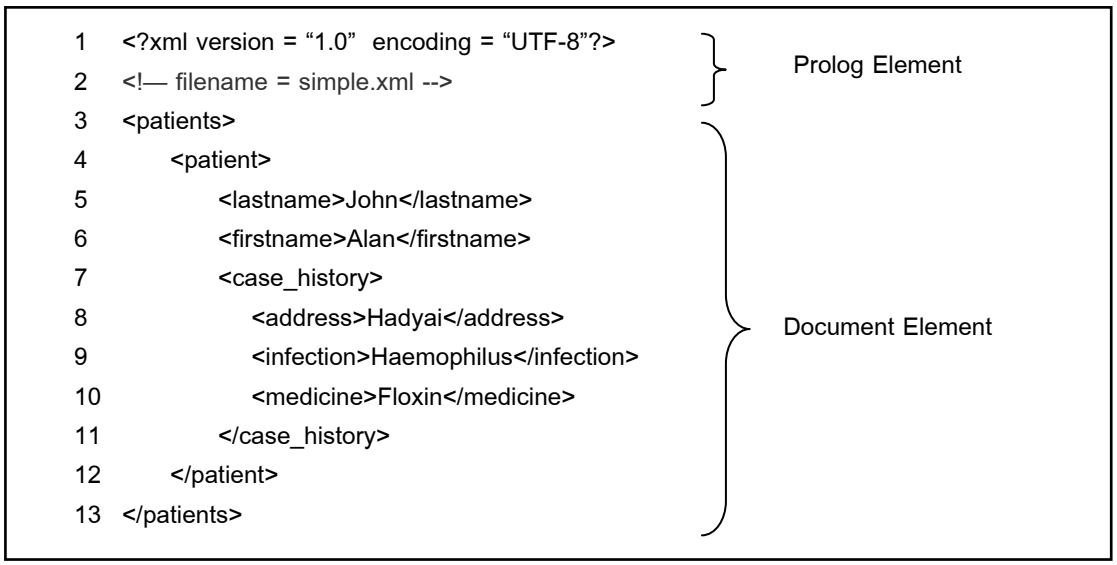
จากภาพประกอบ 2.4 ส่วนที่เป็นแอททริบิวต์ คือ id=“1234”

2.2.2 โครงสร้างของเอกสาร XML

XML มีโครงสร้างหลัก 2 ส่วน คือ Prolog และ Document Element โดย XML สามารถมีอิลิเมนต์รากได้เพียงหนึ่งอิลิเมนต์ แต่สามารถมีอิลิเมนต์ย่อยได้หลากหลาย โครงสร้างเอกสาร XML ดังภาพประกอบ 2.5 สามารถอธิบายได้ดังนี้

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

ส่วนของ Document Element คือ บรรทัดที่ 3-12 คือ ซึ่งเป็นอิลิเมนต์ที่ซ้อนกันอย่างเป็นลำดับ โดยมีอิลิเมนต์ <patients> เป็นอิลิเมนต์ราก



ภาพประกอบ 2.5 ลักษณะโครงสร้างของเอกสาร XML

2.2.3 XML Parser

XML Parser ทำหน้าที่ในการอ่าน แปลความ วิเคราะห์โครงสร้าง และตรวจสอบความถูกต้องของเอกสาร นอกจากนี้ยังจะทำหน้าที่เป็นตัวกลางระหว่างเอกสาร XML และโปรแกรมประยุกต์ที่ต้องการใช้งานข้อมูลของ XML

XML Parser มีอยู่ 2 รูปแบบที่ได้รับความนิยม คือ Document Object Model (DOM) และ Simple API for XML (SAX)

1) DOM

DOM (W3C, 2005) เป็นมาตรฐานที่ถูกกำหนดโดย W3C ในปี 2001 หลักการของ DOM คือ การแปลงเอกสาร XML ให้อยู่ในรูปของโครงสร้างต้นไม้ซึ่งเป็นรูปแบบหนึ่งของโครงสร้างข้อมูล (Data Structure) โดยที่การเข้าถึงข้อมูลจะเรียกว่า การเดินสำรวจ (Traverse) เราสามารถเรียก DOM ได้ว่าเป็น Tree-Based Parser

DOM จะโหลดข้อมูลทั้งหมดเข้ามาในหน่วยความจำ และเริ่มต้นเข้าถึงโหนดรากของโครงสร้างต้นไม้ซึ่งได้มาจากเอกสาร XML

2) SAX

SAX (Megginson, 2004) เป็นอีกรูปแบบหนึ่งของกระบวนการดึงข้อมูล ซึ่งแตกต่างจาก DOM ในเรื่องการโหลดข้อมูลเข้าสู่หน่วยความจำ DOM จะโหลดข้อมูลทั้งหมดเข้าสู่หน่วยความจำแต่ SAX จะไม่โหลดข้อมูลมาทั้งหมดแต่จะขึ้นอยู่กับเหตุการณ์ต่างๆ ที่เกิดขึ้นในเอกสาร เช่น กรณีที่เป็นแท็กเปิดให้ดำเนินการอย่างใดอย่างหนึ่ง เราสามารถเรียก SAX ได้ว่าเป็น Event-Driven Parser

ทั้งสองวิธีการมีความแตกต่างในการทำงานและประโยชน์ ดังตารางที่ 2.1 ซึ่งเป็นการเปรียบเทียบการทำงานระหว่าง DOM และ SAX

ตารางที่ 2.1 เปรียบเทียบการทำงานระหว่าง DOM และ SAX

รูปแบบ	DOM	SAX
วิธีการสำรวจ	Tree-Base Parser	Event-Driven Parser
การอ่านข้อมูลเข้ามาทำงาน	อ่านเอกสารทั้งหมดมาเก็บไว้ในหน่วยความจำ ซึ่งสามารถเรียกใช้งานได้ตลอดเวลา	อ่านข้อมูลในเอกสารทีละชุด และต้องอ่านข้อมูลเดิมซ้ำอีกครั้งเมื่อต้องการใช้งานข้อมูลนั้น
การดึงข้อมูล	ต้องอ่านเอกสารทั้งหมดก่อน ถึงจะดึงข้อมูลได้	สามารถดึงข้อมูลเฉพาะที่ต้องการได้
หน่วยความจำ	ใช้พื้นที่ของหน่วยความจำค่อนข้างมากในการจัดเก็บโครงสร้างต้นไม้	ไม่จำเป็นที่จะต้องโหลดข้อมูลทั้งหมดเข้ามาเก็บไว้ในหน่วยความจำ
วิธีการเข้าถึงข้อมูล	Random	Sequential
การจัดการข้อมูล	สามารถเพิ่ม ลบ แก้ไขและเปลี่ยนแปลงโครงสร้างเอกสาร XML ได้	อ่านได้อย่างเดียวเท่านั้น
โครงสร้างเอกสาร	จำเป็นต้องทราบรายละเอียดของโครงสร้างเอกสารทั้งหมด	ไม่จำเป็นต้องทราบรายละเอียดของโครงสร้างเอกสารทั้งหมด

2.2.4 การเข้าถึงเอกสาร XML

วิธีการในการเข้าถึงข้อมูลในเอกสาร XML ที่ได้รับความนิยมมีอยู่ 2 รูปแบบ คือ XPath และ XQuery

1) XPath

XPath (W3C, 2007) เป็นภาษาที่ใช้ในระบุตำแหน่งบนเอกสาร XML ตามที่ต้องการและสามารถกำหนดเงื่อนไขในการค้นหา สัญลักษณ์ต่างๆที่ใช้ใน XPath แสดงดังตารางที่ 2.2 และตัวอย่างการระบุข้อมูลที่ต้องการภายในเอกสาร XML ด้วย XPath แสดงดังตารางที่ 2.3

ตารางที่ 2.2 สัญลักษณ์ต่าง ๆ ของ XPath

สัญลักษณ์	คำอธิบาย
nodename	เข้าถึงโหนดลูกทุกตัวของโหนดที่มีชื่อตามที่กำหนด
/	อ้างถึงโหนดรากเอกสาร XML
//	อ้างโหนดโดยไม่สนใจว่าข้อมูลระหว่างกลางในเอกสาร XML
.	อ้างถึงโหนดที่อยู่ปัจจุบัน
..	อ้างถึงโหนดพ่อแม่ของโหนดที่อยู่ปัจจุบัน
@	เข้าถึงแอททริบิวต์ของอิลิเมนต์

ตารางที่ 2.3 การระบุข้อมูลด้วย XPath

ตัวอย่าง	คำอธิบาย
address	เข้าถึงโหนดลูกทุกตัวของอิลิเมนต์ address
/address	เข้าถึงโหนด address โดยโหนด address ต้องเป็นอิลิเมนต์ราก
address/street	เข้าถึงโหนด street ที่เป็นสมาชิกของ address
//street	เข้าถึงโหนด street ทุกตัวที่อยู่ในเอกสาร XML
address//street	เข้าถึงโหนด street ที่เป็นลูกของ address จากทุก ๆ ตำแหน่งที่มีลักษณะดังกล่าว
//@name	เข้าถึงโหนดทุกตัวที่มีชื่อแอททริบิวต์ว่า name

2) XQuery

XQuery (W3C, 2009) เป็นภาษาที่ช่วยการค้นหาข้อมูลในเอกสาร XML ซึ่งมีลักษณะที่คล้ายกับภาษา SQL ที่ใช้ในการค้นหาข้อมูลเช่นกัน โดยที่ XQuery มองเอกสารเป็นฐานข้อมูล นอกจากนี้แล้ว W3C ยังได้รับรองในปี 2007 และได้รับการสนับสนุนจาก IBM, Oracle และ Microsoft ประโยชน์ของการใช้ XQuery ได้แก่ ใช้ในการดึงข้อมูลจากเว็บ ใช้ในการสร้างรายงาน และใช้ในการแปลง XML ให้เป็น XHTML เป็นต้น ซึ่งลักษณะของภาษา XQuery ประกอบด้วยคำสั่ง FOR, LET, WHERE, ORDER BY และ RETURN

ลักษณะไวยากรณ์ของภาษา XQuery

1) for ตัวแปร in doc(ชื่อไฟล์ XML) หรือ Path ที่ต้องการ

2) let กำหนดตัวแปรใหม่

3) where เงื่อนไข

4) order by เงื่อนไข

5) return ค่าที่ต้องการส่งกลับ

2.2.5 การนิยามโครงสร้างเอกสาร XML

การกำหนดนิยามของโครงสร้างเอกสาร XML เป็นสิ่งที่สำคัญซึ่งจะนำมาใช้ในการตรวจสอบความถูกต้องของเอกสารและทำให้ง่ายต่อการจัดการเอกสาร โดยที่ประเภทของการนิยามโครงสร้างเอกสาร XML ที่ได้รับความนิยมสามารถแบ่งออกเป็น 2 ประเภท คือ DTD และ XMLSchema

2.2.5.1 DTD (Document Type Definition)

Document Type Declaration (DTD) เป็นรูปแบบการกำหนดโครงสร้าง เพื่อให้สะดวกในการแลกเปลี่ยนเอกสารระหว่างหน่วยงาน โดยที่จะกำหนดว่าเอกสารประกอบด้วย แท็ก แอททริบิวต์ และลักษณะการซ้อนกันของแท็ก ซึ่งการประกาศ DTD นั้นมีอยู่ 2 รูปแบบ คือ แบบ Internal เป็นการประกาศภายในเอกสาร XML และแบบ external เป็นการประกาศโดยที่อ้างอิงมาจากเอกสารภายนอก

```
1 <!ELEMENT Patients (Patient)*>
2 <!ELEMENT Patient (Name, Blood_Group)>
3 <!ELEMENT Name (#PCDATA)>
4 <!ELEMENT Blood_Group (#PCDATA)>
```

ภาพประกอบ 2.6 ตัวอย่างการประกาศ DTD

จากภาพประกอบ 2.6 แสดงตัวอย่างการประกาศ DTD อธิบายได้ดังนี้

บรรทัดที่ 1 เป็นการประกาศว่าในแท็ก Patients จะประกอบด้วยแท็ก Patient

บรรทัดที่ 2 เป็นการประกาศว่าในแท็ก Patient จะประกอบด้วยแท็ก Name และ Blood_Group

บรรทัดที่ 3 เป็นการประกาศว่าในแท็ก Name จะจัดเก็บข้อมูลข้อความธรรมดา

บรรทัดที่ 4 เป็นการประกาศว่าในแท็ก Blood_Group จะจัดเก็บข้อมูลความธรรมดา

เนื่องจาก DTD มีข้อจำกัดอยู่หลายประการ เช่น มีชนิดของการจัดเก็บข้อมูลไม่กี่ชนิด และตัวของ DTD ไม่อยู่ในรูปของแท็ก ซึ่งทำให้เกิดความยากลำบากในการเขียนโปรแกรมจัดการข้อมูล DTD และข้อมูลเอกสาร XML ในตัวเดียวกัน

2.2.5.2 XMLSchema

XMLSchema (W3C, 2001) เป็นรูปแบบการกำหนดโครงสร้างที่เป็นมาตรฐานจาก W3C ในปี 2001 ซึ่งมีข้อดีมากกว่า DTD เช่น สามารถกำหนด

รูปแบบข้อมูลตามต้องการ สร้างชนิดข้อมูลใหม่โดยอ้างอิงจากชนิดข้อมูลที่มีอยู่เดิม และสามารถกำหนดจำนวนและลำดับของอิลิเมนต์ลูกได้ ดังภาพประกอบ 2.7

```
<?xml version="1.0"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
<xs:element name="Patient">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Name" type="xs:string"/>
   <xs:element name="Blood_Group" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
 </xs:element>
</xs:schema>
```

ภาพประกอบ 2.7 ตัวอย่างการประกาศ XML Schema

2.2.6 รูปแบบการจัดเก็บเอกสาร XML

วิธีการในการจัดเก็บเอกสาร XML (Silberschatz, 2006) แบ่งออกเป็น 2 ประเภท คือ

1.การจัดเก็บข้อมูลแบบ Flat File เป็นวิธีการจัดเก็บข้อมูลในรูปแบบของเท็กซ์ไฟล์ (Text File) โดยที่ไฟล์จะมีนามสกุล .xml ซึ่งสามารถเข้าถึงเอกสาร XML ได้โดยตรง และสามารถใช้งานร่วมกับโปรแกรมประยุกต์ต่างๆได้หลากหลาย นอกจากนี้ยังสามารถรองรับการทำงานร่วมกับภาษาคอมพิวเตอร์ได้หลายภาษา เช่น C, C++, Java เป็นต้น ซึ่งในการเขียนโปรแกรมจัดการฐานข้อมูลจะมีการจัดการข้อมูลในลักษณะของ Object-oriented Database

2.การจัดเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ในการจัดเก็บข้อมูลโดยวิธีการนี้จะต้องมีการแปลงข้อมูลจากเอกสาร XML ให้มาอยู่ในรูปแบบตาราง โดยที่สามารถจัดเก็บได้หลากหลายรูปแบบ เช่น การสร้างฐานข้อมูล โดยที่ในตารางของฐานข้อมูลนั้นจะมีขอบเขตข้อมูลที่เป็นชนิดข้อความซึ่งใช้สำหรับจัดเก็บข้อมูลของเอกสาร XML นอกจากรูปแบบนี้แล้วยังมีวิธีการจัดเก็บข้อมูลโดยการแปลงข้อมูลเอกสาร XML ให้อยู่ในรูปของต้นไม้ก่อน แล้วจึงนำไปจัดเก็บในฐานข้อมูล การจัดเก็บข้อมูลเอกสาร XML ไว้ในฐานข้อมูลเชิงสัมพันธ์จะเพิ่มประสิทธิภาพในการเพิ่ม ลบ แก้ไข และค้นหาข้อมูล

3. การจัดเก็บข้อมูลในรูปแบบ XML Database เป็นการจัดเก็บสำหรับการจัดเก็บข้อมูลในรูปแบบกึ่งโครงสร้างซึ่งสามารถค้นหาข้อมูล แต่มีปัญหาในเรื่องการแก้ไขข้อมูล

2.3 ความปลอดภัย (Security)

ความปลอดภัยเป็นประเด็นสำคัญในการแลกเปลี่ยนข้อมูลระหว่างผู้รับกับผู้ส่ง เนื่องจากในปัจจุบันมีการแลกเปลี่ยนข้อมูลระหว่างระบบเครือข่ายเป็นจำนวนมาก ผลที่ตามมาก็คือมีผู้ที่ประสงค์ร้ายต่อข้อมูลและต่อการได้รับข้อมูลเพื่อมาใช้ประโยชน์โดยที่ตนเองไม่สิทธิในการเข้าถึงข้อมูลนั้น การที่จะทำให้ข้อมูลเกิดความปลอดภัยได้นั้นจำเป็นที่จะต้องมีพื้นฐานสำคัญในการรักษาความปลอดภัย (Kahate, 2006)

2.3.1 พื้นฐานความปลอดภัย

1) การพิสูจน์ตัวตน (Authentication) เป็นวิธีการในการตรวจสอบว่าบุคคลคนนี้คือใคร ตัวอย่างที่มีใช้ในโลก เช่น บัตรประชาชน บัตรสมาชิก เป็นต้น ส่วนในระบบเครือข่ายอินเตอร์เน็ตมักจะใช้ Username และ Password ในการพิสูจน์ตัวตนว่าผู้ที่เข้ามาในระบบเป็นใคร

2) การรักษาความลับ (Confidentiality) เป็นวิธีการในการปกปิดข้อมูลให้เป็นความลับอยู่เสมอ โดยที่ผู้ที่มีสิทธิ์ในการเข้าถึงเท่านั้นที่สามารถทราบรายละเอียดหรือเนื้อหาภายในนั้น ตัวอย่างของการรักษาความลับในระบบเครือข่ายอินเตอร์เน็ตคือ การเข้ารหัสข้อมูลแล้วอนุญาตเฉพาะผู้ที่มีกุญแจที่ใช้สำหรับการถอดรหัสเท่านั้นที่สามารถเข้าถึงหรืออ่านเนื้อหาของข้อมูลนั้น

3) ความคงสภาพ (Integrity) เป็นวิธีการที่ทำให้ผู้ส่งข้อมูลและผู้รับข้อมูลมั่นใจได้ว่าข้อมูลที่ถูกส่งผ่านตัวกลางจะไม่ถูกแก้ไขระหว่างการการส่ง โดยที่ข้อมูลที่ผู้ส่งได้ส่งออกไปนั้นจะต้องเหมือนกับข้อมูลที่ผู้รับได้รับมา

4) การไม่ปฏิเสธแหล่งที่มา (Non-repudiation) เมื่อมีบุคคลใดได้ส่งข้อมูลไปยังผู้รับ บุคคลนั้นจะไม่สามารถปฏิเสธว่าตัวเองไม่ใช่ผู้ที่ส่งข้อมูลออกไป

2.3.2 ประเภทของผู้โจมตี

ประเภทของผู้โจมตี (Bouganim *et al.*, 2002) สามารถแบ่งเป็น 3 กลุ่มคือ

1) ผู้บุกรุก (Intruder) เป็นบุคคลที่อยู่ภายนอกกลุ่มและต้องทำลายความปลอดภัยของข้อมูล ซึ่งจะมีวิธีการในการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

2) บุคคลในกลุ่มเดียวกัน (Insider) เป็นบุคคลที่อยู่ในกลุ่มเดียวกันและสิทธิในการเข้าถึงข้อมูล โดยที่แต่ละคนจะมีสิทธิในการเข้าถึงข้อมูลที่แตกต่างกัน

3) ผู้ดูแลระบบ (Administrator) เป็นบุคคลที่มีสิทธิในการเข้าถึงข้อมูลทุกประการในฐานข้อมูล และสามารถกำหนดสิทธิ์ให้แก่ผู้อื่น

2.3.3 วิธีการโจมตีต่อแหล่งข้อมูล

วิธีการโจมตี (Agrawal et al., 2004) แบ่งประเภทเป็น 3 ประเภท คือ

1) โจมตีแหล่งเก็บข้อมูลทางตรง โดยไม่ผ่านการทำงานของโปรแกรมที่ช่วยในการจัดการฐานข้อมูล

2) โจมตีแหล่งเก็บข้อมูลทางอ้อม โดยผ่านการทำงานของโปรแกรมที่ช่วยในการจัดการฐานข้อมูล

3) โจมตีหน่วยความจำหลัก โดยการเข้าถึงข้อมูลที่ถูกจัดเก็บในหน่วยความจำหลักในขณะที่กำลังดำเนินการ

2.3.4 วิทยาการเข้ารหัสลับ (Cryptography)

ภาพประกอบ 2.8 กระบวนการเข้ารหัสและถอดรหัสข้อมูล

การเข้ารหัสข้อมูล (Kahate, 2006) เป็นกลไกหนึ่งในการรักษาความลับของข้อมูล และเป็นกลไกที่สามารถสร้างความปลอดภัยในการส่งข้อมูลระหว่างระบบเครือข่ายอินเตอร์เน็ต การเข้ารหัสข้อมูลมีการพัฒนามาโดยตลอดตั้งแต่อดีตจนถึงปัจจุบัน

การรักษาความปลอดภัยของวิธีการนี้จะแบ่งออกเป็น 2 ส่วน คือ การเข้ารหัสลับข้อมูล (Encryption) และ การถอดรหัสลับข้อมูล (Decryption) ตามภาพประกอบ 2.8

การเข้ารหัสลับข้อมูล หมายถึง การแปลงข้อมูลที่สามารถอ่านเข้าใจ (Plain Text) ให้เป็นข้อมูลที่ไม่สามารถอ่านเข้าใจ (Cipher Text)

การถอดรหัสลับข้อมูล หมายถึง การแปลงข้อมูลที่ไม่สามารถอ่านเข้าใจ (Cipher Text) ให้กลับมาเป็นข้อมูลที่สามารถอ่านเข้าใจได้ (Plain Text)

วิธีการในการเข้ารหัสข้อมูล (PaperMaster, 2009) ออกเป็น 5 วิธี คือ

1) วิธีการแทนที่เป็นวิธีการที่จะต้องกำหนดกระบวนการในการเปลี่ยนแปลงข้อมูลจากเดิมเป็นข้อมูลใหม่ และเมื่อมีการแทนที่ข้อมูลใด จะต้องสามารถแทนที่ข้อมูลนั้นให้กลับมาเป็นเหมือนเดิมได้

2) วิธีการเข้ารหัสข้อมูลเป็นกลุ่ม เมื่อต้องการเข้ารหัสข้อมูลใด จะต้องมีการแบ่งข้อมูลออกเป็นกลุ่ม โดยที่แต่ละกลุ่มมีขนาดเท่ากัน และนำข้อมูลแต่ละกลุ่มไปผ่านการดำเนินการซึ่งอาจจะเป็น XOR โดยทั่วไปอาจจะมีการเชื่อมโยงระหว่างข้อมูลแต่ละกลุ่ม

3) วิธีการเรียงสับเปลี่ยน เป็นกระบวนการในการสับเปลี่ยนข้อมูลที่ต้องการ โดยที่จะมีดัชนีที่ช่วยในการสับเปลี่ยน และจะมีการจัดเก็บดัชนีในทั้ง 2 ฝั่ง ซึ่งความปลอดภัยของวิธีการนี้จะขึ้นอยู่กับความซับซ้อนในการสับเปลี่ยนและความปลอดภัยของดัชนี

4) วิธีการขยายข้อมูล เป็นวิธีการเพิ่มข้อมูลแทรกเข้าไปในข้อมูลเดิมซึ่งทำให้ข้อมูลที่ถูกเข้ารหัสมีขนาดใหญ่กว่าข้อมูลเดิม โดยที่จะต้องมีการกำหนดรูปแบบในการเพิ่มข้อมูลและสามารถกำจัดข้อมูลเพิ่มในกรณีที่ต้องการข้อมูลเดิม ความปลอดภัยของวิธีการนี้จะขึ้นอยู่กับตำแหน่งของการเพิ่มข้อมูล

5) วิธีการลดขนาดข้อมูล เป็นการลดข้อมูลจากเดิมให้มีขนาดเล็กลง ลดทรัพยากรที่ใช้ในการทำงาน พื้นที่ในการจัดเก็บลดลง รวมทั้งสามารถลดขนาดของการถ่ายโอนข้อมูล แต่วิธีการนี้อาจจะมีข้อมูล 2 ชุดที่ผ่านกระบวนการลดขนาดแล้วมีผลลัพธ์ที่เหมือนกัน จึงทำให้ยากในการทำกระบวนการย้อนกลับมาให้เป็นข้อมูลเดิม ดังนั้นวิธีการจึงเหมาะสำหรับใช้เป็นฟังก์ชันทางเดียว

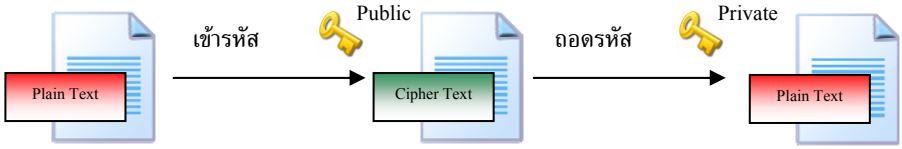
2.3.5 ประเภทของการเข้ารหัสข้อมูล

1) ระบบเข้ารหัสแบบกุญแจสมมาตร (Symmetric-key cryptography) เป็นกรรมวิธีการในการเข้ารหัสข้อมูล และการถอดรหัสข้อมูลโดยใช้กุญแจ (Key) ตัวเดียวกัน หรืออาจจะเรียกอีกอย่างหนึ่งได้ว่า การเข้ารหัสและถอดรหัสแบบสมมาตร (Symmetric) ตัวอย่างของกรรมวิธีการเข้ารหัสเช่น Data Encryption Standard (DES) และ Advanced Encryption Standard (AES) เป็นต้น เนื่องจากการใช้กุญแจตัวเดียวกันในการเข้าและถอดรหัส จึงทำให้เกิดปัญหาที่สำคัญนั่นคือ การแลกเปลี่ยนกุญแจ ดังตัวอย่างในภาพประกอบ 2.9

ภาพประกอบ 2.9 กระบวนการเข้ารหัสและถอดรหัสแบบสมมาตร

2) ระบบเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric-key cryptography) เป็นวิธีการเข้ารหัสและถอดรหัสโดยใช้กุญแจคนละตัว โดยที่ในกระบวนการเข้ารหัสจะใช้กุญแจสาธารณะ (Public Key) ของผู้รับ ซึ่งเป็นข้อมูลที่เปิดเผยต่อสาธารณะ ส่วนกระบวนการถอดรหัสนั้นจะใช้กุญแจส่วนตัวของผู้รับ ซึ่งเป็นข้อมูลที่ถูกเก็บไว้ที่ผู้รับและไม่เปิดเผยต่อสาธารณะ ซึ่งเป็นวิธีที่สามารถแก้ปัญหาการแลกเปลี่ยนกุญแจระหว่างผู้รับและผู้ส่ง ตัวอย่าง

ของวิธีการเข้ารหัสเช่น RSA Diffie-Hellman และDigital Signature Algorithm (DSA) เป็นต้น ดังตัวอย่างในภาพประกอบ 2.10



ภาพประกอบ 2.10 กระบวนการเข้ารหัสและถอดรหัสแบบอสมมาตร

2.3.6 วิธีการเข้ารหัสลับ

วิธีการเข้ารหัสลับมีอยู่หลากหลายวิธีที่ได้ถูกนำเสนอ ตัวอย่างเช่น

1) AES

AES (Paar และ Pelzl, 2009) หรือ Advanced Encryption Standard ถูกสร้างและปรับปรุงโดยรัฐบาลกลางของสหรัฐอเมริกา โดยที่เป็นวิธีการเข้ารหัสแบบกุญแจสมมาตร ซึ่งในการเข้ารหัสนั้นจะแบ่งข้อมูลออกเป็นกลุ่มย่อย ขนาดของกลุ่มย่อยมีอยู่ทั้งหมด 3 ขนาด คือ 128 192 และ 256

วิธีการนี้จะแบ่งข้อมูลให้อยู่ในอาร์เรย์ขนาด 4x4 และมีการดำเนินการเป็นรอบแตกต่างกัน ขนาด 128 จะดำเนินการ 10 รอบ ขนาด 192 จะดำเนินการ 12 รอบ และ ขนาด 256 จะดำเนินการ 14 รอบ การทำงานจะแบ่งออกเป็นดังนี้

1) Key Addition เป็นการรวมข้อมูลที่ต้องการกับกุญแจ

2) Byte Substitution เป็นการแทนที่ข้อมูลตามรหัสของข้อมูลในตาราง

3) Diffusion เป็นแบ่งออกเป็น 2 ขั้นตอนย่อยคือ ShiftRows และ MixColumn

2) RSA

RSA (Cormen และคณะ, 2001) เป็นวิธีการที่ตั้งชื่อตามผู้ออกแบบนั้นคือ Rivest Shamir และ Adleman โดยที่เป็นวิธีการเข้ารหัสแบบกุญแจอสมมาตร ซึ่งผู้ใช้ทั้ง 2 ฝั่งจะมีกุญแจที่ใช้ในการเข้ารหัสและถอดรหัสที่แตกต่างกัน วิธีการนี้สามารถช่วยในการแก้ปัญหาในการแลกเปลี่ยนกุญแจระหว่างผู้รับและผู้ส่งข้อมูล ในวิธีการนี้จำเป็นที่จะต้องหาจำนวนเฉพาะที่แตกต่างกัน 2 จำนวนเพื่อนำมาใช้ในการคำนวณหาค่ากุญแจที่จะใช้เป็นกุญแจสาธารณะและกุญแจส่วนตัว ซึ่งความสัมพันธ์เป็นไปตามสมการ $de \equiv 1 \bmod (p-1)(q-1)$ โดยที่ d คือกุญแจที่ใช้ในการถอดรหัส e คือกุญแจที่ใช้ในการเข้ารหัส p และ q เป็นจำนวนเฉพาะ การเข้ารหัสจะใช้สมการ $c = m^e \bmod pq$ ในส่วนของการถอดรหัสจะใช้สมการ $m = c^d \bmod pq$ โดยที่ c คือข้อมูลที่ถูกเข้ารหัสและ m คือข้อมูลเดิม

3) One-time-pad

One-time-pad (Katz และ Lindell, 2007) เป็นอีกหนึ่งขั้นตอนวิธีในการเข้ารหัสข้อมูลเอกสาร ซึ่งจุดเด่นจะมีการสุ่มกุญแจขึ้นมาเพื่อใช้ในการเข้ารหัสข้อมูลในแต่ละครั้ง ทำให้ Cipher Text ที่ได้ออกมาแต่ละครั้งมีลักษณะที่แตกต่างกันยากต่อการคาดเดาข้อมูล แต่ข้อจำกัดของวิธีการนี้คือ จะต้องมี Key Pad ที่มีการกระจายความน่าจะเป็นในทุกกุญแจที่เป็นไปได้ และมีความยากลำบากในการแลกเปลี่ยน Key Pad

2.3.7 วิธีการโจมตีต่อการเข้ารหัสข้อมูล

วิธีการโจมตีต่อการเข้ารหัสข้อมูล (Unay และ Gundem, 2008) สามารถแบ่งออกได้เป็น 8 รูปแบบ

1) Brute Force Attacks เป็นวิธีการโจมตีโดยอาศัยกุญแจทั้งหมดที่เป็นไปได้ในการทดลองถอดรหัสข้อมูลที่ถูกเข้ารหัส ซึ่งจะใช้เวลานานมากน้อยเพียงใดขึ้นอยู่กับความยาวของกุญแจที่ใช้

2) Cipher Text Only Attacks เป็นวิธีการโจมตีที่ผู้โจมตีมุ่งเน้นในการค้นหาความสัมพันธ์ระหว่างข้อมูลที่ถูกเข้ารหัสเท่านั้น ซึ่งผู้โจมตีจะไม่ทราบรายละเอียดของข้อมูลที่แท้จริง

3) Known Plaintext Attacks เป็นวิธีการโจมตีที่ผู้โจมตีทราบข้อมูลรายละเอียดของข้อมูลจริงที่ตรงและข้อมูลที่เข้ารหัสของข้อมูลนั้นและพยายามค้นหากุญแจที่ใช้ในการถอดรหัสข้อมูล

4) Chosen Plaintext Attacks เป็นวิธีการโจมตีซึ่งผู้โจมตีจะเลือกข้อมูลใดๆ และสังเกตว่าข้อมูลนั้นเมื่อผ่านการเข้ารหัสจะได้ข้อมูลออกมาเป็นแบบใด ซึ่งผู้โจมตีจะอาศัยข้อมูลในส่วนนี้ในการโจมตีระบบ

5) Adaptive chosen plaintext attacks เป็นวิธีการโจมตีที่มีลักษณะคล้ายกับ Chosen plaintext attacks แต่จะแตกต่างกันตรงที่วิธีการนี้จะพยายามสังเกตว่าข้อมูลที่ถูกเข้ารหัสในแต่ละรอบการเข้ารหัสมีความแตกต่างกันอย่างไร

6) Chosen Cipher Text Attacks เป็นวิธีการโจมตีซึ่งผู้โจมตีจะเลือกข้อมูลที่ถูกเข้ารหัสใดๆและสังเกตว่าข้อมูลนั้นเมื่อผ่านการเข้ารหัสจะได้ข้อมูลออกมาเป็นแบบใด ซึ่งผู้โจมตีจะอาศัยข้อมูลในส่วนนี้ในการโจมตีระบบ

7) Frequency Based Attack เป็นวิธีการโจมตีโดยการสังเกตจำนวนข้อมูลที่ซ้ำกันในระบบที่มีการจัดเก็บข้อมูลที่ถูกเข้ารหัสไว้ ซึ่งจะนำมาใช้ในการคาดเดาจำนวนข้อมูลจริง

8) Size-based Attack เป็นวิธีการโจมตีโดยการสังเกตความยาวของข้อมูลที่ถูกเข้ารหัสเปรียบเทียบกับข้อมูลจริง

2.3.8 ฟังก์ชันแฮช (Hash Function)

ฟังก์ชันแฮช (Cormen *et al.*, 2001) เป็นฟังก์ชันที่ใช้ในการแปลงข้อมูลหนึ่งให้เป็นอีกข้อมูลหนึ่ง ซึ่งข้อมูลที่ได้มาจากฟังก์ชันแฮชมักจะมีขนาดเล็กกว่าหรือเท่ากับข้อมูลเดิม นอกจากนี้แล้วมักจะนำฟังก์ชันแฮชมาช่วยในการสร้างดัชนีเพื่อการค้นหาข้อมูลและใช้เป็นลายมือชื่อดิจิตอลเพื่อรับรองเอกสารอิเล็กทรอนิกส์ โดยส่วนใหญ่แล้วฟังก์ชันแฮชจะทำการแบ่งข้อมูลเดิมให้กลายเป็นข้อมูลย่อย แล้วใช้ในการกำหนดรหัส หลังจากนั้นจะนำข้อมูลทั้งหมดมารวมกันเพื่อสร้างผลลัพธ์หรือรหัสสุดท้าย เมื่อข้อมูลผ่านฟังก์ชันแฮชแล้วควรได้ค่าออกมาแตกต่างกัน เพื่อความรวดเร็วในการค้นหาและตรวจสอบความถูกต้องของข้อมูล ตัวอย่างของฟังก์ชันแฮชที่ใช้กันอย่างแพร่หลายเช่น SHA-1, MD5 และ CRC32 เป็นต้น ข้อควรระวังในการเลือกใช้ฟังก์ชันแฮชมีอยู่ 2 ประการ คือ ปัญหาในการชนกันของข้อมูล (Collision) และข้อความที่ถูกย่อยผ่านฟังก์ชันแฮชจะไม่สามารถย้อนกลับมาเป็นข้อความเดิมได้

2.3.9 การค้นหาข้อมูลที่ถูกเข้ารหัส

วิธีการเข้ารหัสข้อมูลและสามารถค้นหาข้อมูลที่ถูกเข้ารหัสได้ (Song *et al.*, 2000) ซึ่งเหมาะสำหรับการเข้ารหัสข้อมูลและนำไปฝากไว้ที่ผู้ให้บริการ นอกจากนี้แล้วขั้นตอนการค้นหาของวิธีการนี้มีประสิทธิภาพ เนื่องจากเมื่อมีข้อมูลจำนวน *n* ชุด จะใช้เวลาในการค้นหาเท่ากับ *n* เช่นเดียวกัน

ในขั้นตอนวิธีนี้ประกอบด้วย 2 ส่วนหลัก คือวิธีการที่ใช้ในการเข้ารหัสข้อมูลและวิธีการในการค้นหาข้อมูล โดยที่ในส่วนของขั้นตอนในการเข้ารหัสข้อมูลจะประกอบไปด้วย

1) เข้ารหัสข้อมูลโดยใช้กุญแจส่วนตัวของผู้รับบริการ

2) นำข้อมูลที่ถูกเข้ารหัสแบ่งออกเป็น 2 ส่วนคือข้อมูลฝั่งซ้ายและขวา

3) สุ่มข้อมูลมา 1 ชุด ซึ่งมีขนาดเท่ากับข้อมูลที่ถูกเข้ารหัสฝั่งซ้าย

4) เข้ารหัสข้อมูลที่สุ่มโดยใช้ข้อมูลที่เข้ารหัสก่อนหน้านี้เป็นกุญแจ

5) ดำเนินการ XOR ระหว่างข้อมูลที่ถูกเข้ารหัสในฝั่งซ้ายกับข้อมูลที่สุ่ม และระหว่างข้อมูลที่ถูกเข้ารหัสฝั่งขวากับข้อมูลที่ได้จากขั้นตอนที่ 4

6) นำข้อมูลที่ผ่านขั้นตอนที่ 5 ไปเก็บไว้ในฝั่งของผู้ให้บริการ

ในส่วนของขั้นตอนการค้นหาจะประกอบไปด้วย

1) ผู้ใช้เข้ารหัสข้อความที่ต้องการค้นหาโดยใช้กุญแจส่วนตัว

2) ส่งข้อมูลที่ผ่านขั้นตอนที่ 1 ไปยังผู้ให้บริการ

3) ผู้ให้บริการจะนำข้อมูลที่ได้รับไปดำเนินการ XOR กับข้อมูลที่เก็บไว้ ถัดจากนั้นจะพิจารณาว่าข้อมูลใดที่เมื่อแบ่งเป็น 2 ฝั่งแล้ว เมื่อนำฝั่งซ้ายผ่านกระบวนการเข้ารหัสโดยใช้ข้อมูลที่รับมาจากขั้นตอนที่ 2 เป็นกุญแจในการเข้ารหัส แล้วผลลัพธ์ที่ได้ออกมาเหมือนกับข้อมูลที่ถูกเข้ารหัส

2.4 การเข้ารหัสเอกสาร XML (XML Encryption)

การเข้ารหัสข้อมูลในเอกสาร XML มีอยู่มากมายหลายรูปแบบหนึ่งในนั้นเป็นมาตรฐานที่ถูกกำหนดโดย W3C นั่นคือ XML Encryption Standard และส่วนที่เหลือเป็นงานวิจัยที่ถูกเสนอโดยนักวิจัยหลายๆท่าน

2.4.1 มาตรฐานการเข้ารหัสเอกสาร XML (XML Encryption Standard)

W3C ได้มีกำหนดมาตรฐานในการเข้ารหัสเอกสาร XML (XML Encryption Standard) (Imamura *et al.*, 2002) ซึ่งจะมีการแทนที่ข้อมูลที่ต้องการเข้ารหัสด้วยแท็ก EncrytpedData โดยที่ภายในแท็กจะประกอบไปด้วย 4 ส่วนย่อยคือ 1) EncryptionMethod ใช้สำหรับจัดเก็บวิธีการในการเข้ารหัส KeyInfo ใช้สำหรับจัดเก็บชื่อของกุญแจที่ใช้ในการเข้าและถอดรหัส CipherData ใช้สำหรับจัดเก็บข้อมูลที่ถูกเข้ารหัส และ EncryptionProperties ใช้สำหรับจัดเก็บข้อมูลเสริมเกี่ยวกับข้อมูลที่ถูกเข้ารหัส ภาพประกอบ 2.11 แสดง syntax ของ XML Encryption และตัวอย่างของการแทนที่แท็กที่ถูกเข้ารหัสดังภาพประกอบ 2.12

```
<EncryptedData Id? Type? MimeType? Encoding?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
    <ds:KeyName>?
    <ds:RetrievalMethod>?
    <ds:*>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>?
    <CipherReference URI?>?
  </CipherData>
  <EncryptionProperties>?
</EncryptedData>
```

ภาพประกอบ 2.11 Syntax ของ XML Encryption Standard

```
<EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#" Id="Test"
Type="http://www.w3.org/2001/04/xmlenc#Element">
    <EncryptionMethod />
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <KeyName>theKey</KeyName>
    </ds:KeyInfo>
    <CipherData>
      <CipherValue>ATX7rKlOLs2cjoGgLf5tJfY7mq+Rudaq</CipherValue>
    </CipherData>
</EncryptedData>
```

ภาพประกอบ 2.12 ตัวอย่างของแท็กของ XML encryption ในเอกสาร XML

2.4.2 วิธีการเข้ารหัสเอกสาร XML ในงานวิจัยอื่น ๆ

1) Efficient Processing of Secured XML Metadata (Feng และ Jonker, 2003)

ในปี ค.ศ. 2003 Feng และ Jonker ได้เสนอวิธีการจัดเก็บ mapped values ไว้ในฝั่งผู้ให้บริการและใช้ข้อมูล XML DTD ในงานวิจัยนี้ได้แบ่งขั้นตอนการทำงานออกเป็น 3 ช่วง คือ Query Preparation, Query Pre-Processing, และ Query Execution ในขั้นตอน Query Preparation มีการสร้างตาราง Path ซึ่งเป็นการเข้ารหัส Path โดยใช้ Hash Function ซึ่งสามารถค้นหาข้อมูล XML ที่เป็นจำนวนตัวเลข โดยในการเข้ารหัสจะมีการแบ่งข้อมูลตัวเลขออกเป็นกลุ่มแล้วจึงนำมาเข้ารหัส ขั้นตอน Query Pre-Processing จะเป็นขั้นตอนที่มีการแปลง XPath ออกเป็น Path ต่างๆ แล้วใช้ข้อมูลในตาราง Path ที่สร้างมาจากขั้นตอนที่ 1 มาช่วยในค้นหาข้อมูล ในส่วนสุดท้ายคือ Query Execution ซึ่งเป็นขั้นตอนการค้นหาข้อมูลในฐานข้อมูล

ข้อดี

1. สามารถค้นหาข้อมูลที่เข้ารหัส และกรองข้อมูลผลลัพธ์ให้ตรงตามความต้องการของผู้ใช้
2. สามารถใช้ XPath ในการค้นหาข้อมูล
3. สามารถค้นหาข้อมูล XML ที่เป็นจำนวนตัวเลข โดยการแบ่งเป็นช่วงของข้อมูลจำนวนตัวเลข

ข้อเสีย

ต้องเข้ารหัสเอกสารทั้งหมด ไม่สามารถเข้ารหัสเพียงบางส่วน

2) SemCrypt Ensuring Privacy of Electornic Documents Through Semantic Based Encrypted Query Processing (Schrefl, Gruen และ Dorn, 2005)

ในปี ค.ศ. 2005 Schrefl, Gruen และ Dorn ได้เสนอวิธีการเข้ารหัสข้อมูลเอกสาร XML โดยที่จะมีการสร้างข้อมูลที่ช่วยในการเข้ารหัส และแบ่งข้อมูลเป็น 2 ส่วนซึ่งถูกจัดเก็บไว้ที่ฝั่งผู้ใช้และผู้ให้บริการ ในระหว่างการเรียกข้อมูลจะมีการแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้รับบริการเพื่อช่วยในการค้นหา นอกจากนี้ยังสนับสนุนการเพิ่มข้อมูลและการแก้ไขข้อมูลในเอกสาร XML ที่ถูกเข้ารหัส โดยมีผลกระทบต่อดัชนีที่จัดเก็บไว้ก่อนหน้านี้ วิธีการนี้สามารถใช้ในการเข้ารหัสข้อมูลเพียงบางส่วนได้

ข้อดี

1.สนับสนุนการเพิ่มข้อมูลและการแก้ไขข้อมูลในเอกสาร XML ที่ถูกเข้ารหัส

2.สามารถลดการซ้ำซ้อนของเอกสาร XML

3.สามารถป้องกันการโจมตีแบบ Frequency Based

ข้อเสีย

มีการแลกเปลี่ยนข้อมูลระหว่างผู้รับบริการและผู้ให้บริการหลายครั้งในการตอบสนองคำสั่ง 1 คำสั่ง

3) Secure query processing against encrypted XML data using Query-Aware Decryption (Lee และ Whang, 2005)

ในปี ค.ศ. 2005 Lee และ Whang ได้เสนอวิธีการเข้ารหัสข้อมูลเอกสาร XML โดยที่จะมุ่งเน้นทางด้านการใช้กุญแจที่แตกต่างกันในการแยกแยะสิทธิ์ของผู้ใช้ โดยที่ผู้ใช้ที่มีกุญแจแตกต่างกันจะเข้าถึงข้อมูลเอกสาร XML ที่แตกต่างกัน ในการเข้ารหัสข้อมูลจะมีการกำหนดหมายเลขให้กับโหนดข้อมูลแต่ละโหนดเรียกว่า Dewey Number หลังจากนั้นจะมีการสร้างดัชนีเข้ารหัสซึ่งประกอบด้วย 3 ส่วนคือ Key Name, Element Type, และ Occurrences ในส่วนของ Key Name จะเป็นการจัดเก็บข้อมูลชื่อของกุญแจที่ใช้ในการเข้ารหัสโหนดข้อมูล Element Type ใช้สำหรับจัดเก็บข้อมูลชื่อโหนด และในส่วนสุดท้ายคือ Occurrences จะเป็นข้อมูลของ Dewey Number ของแต่ละโหนด

ข้อดี

1.ใช้กุญแจในการเข้ารหัสข้อมูลหลายชุด ซึ่งสามารถใช้แยกแยะสิทธิ์ในการเข้าถึงข้อมูลของผู้ใช้แต่ละคน

2.สามารถเข้ารหัสข้อมูลเพียงบางส่วน

3.ดัชนีมีขนาดเล็ก

ข้อเสีย

มีการเข้ารหัสข้อมูลหลายชั้น และเสียเวลาในการถอดรหัสข้อมูลในแต่ละชั้น

4) Querying Encrypted XML Documents (Jammalamadaka และ Mehrotra, 2006)

ในปี ค.ศ. 2006 Jammalamadaka และ Mehrotra ได้เสนอวิธีการเข้ารหัสเอกสาร XML โดยแบ่งรูปแบบการเข้ารหัสออกเป็น 3 รูปแบบ คือ การเข้ารหัสโครงสร้างเอกสาร การเข้ารหัสค่าข้อมูล และการเข้ารหัสแท็กข้อมูล ในส่วนของข้อมูลที่ใช้ช่วยในการค้นหาจะถูกจัดเก็บไว้ที่แต่ละโหนดเรียกว่า Ancillary Information

ข้อดี

1. สามารถกำหนดรูปแบบของการเข้ารหัสได้ 3 รูปแบบ ได้แก่ การเข้ารหัส node ข้อมูลทั้งหมด การเข้ารหัสเพื่อซ่อนความสัมพันธ์ระหว่าง node และการเข้ารหัสข้อมูล tag

2. สามารถเข้ารหัสข้อมูลที่เป็นจำนวนตัวเลข โดยที่ยังคงรักษาการเรียงลำดับของข้อมูล เพื่อให้ง่ายต่อการค้นหาข้อมูล

ข้อเสีย

มีการจัดเก็บข้อมูลที่ใช้ช่วยในการค้นหาเป็นจำนวนมาก ทำให้เพิ่มพื้นที่ในการจัดเก็บข้อมูล

5) Efficient Secure Query Evaluation over Encrypted XML Databases (Wang และ Lakshmanan, 2006)

ในปี ค.ศ. 2006 Wang และ Lakshmanan ได้เสนอวิธีการจัดเก็บ Metadata ที่ใช้สำหรับช่วยในการประมวลผล Query ไว้ในฝั่งผู้ให้บริการ เพื่อลดการทำงานในฝั่งผู้รับบริการ และยังสามารถกำหนดกฎเกณฑ์ในการเข้ารหัสข้อมูลเพียงบางส่วน ในส่วนของ DSI Index Table จะเป็นดัชนีที่สร้างจากขอบเขตของแต่ละอิลิเมนต์ของเอกสาร XML นอกจากนี้แล้วสามารถป้องกันการโจมตีแบบ Frequency-based และ Sized-based โดยการแบ่งข้อมูลออกเป็นกลุ่ม และเพิ่มข้อมูลหลอกเข้าไปเพื่อทำให้ข้อมูลที่ถูกเข้ารหัสมีขนาด ขั้นตอนการแปลงคำร้องในการค้นหาจะเกิดขึ้นในฝั่งของผู้ให้บริการเพื่อความสะดวกและลดการดำเนินการของผู้ใช้

ข้อดี

1.มีการจัดเก็บ Metadata ที่ใช้สำหรับช่วยในการประมวลผล Query ไว้ในฝั่งผู้ให้บริการ เพื่อลดการทำงานในฝั่งผู้รับบริการ

2.สามารถกำหนดกฎเกณฑ์ในการเข้ารหัสข้อมูลเพียงบางส่วน (ไม่ต้องเข้ารหัสทั้งเอกสาร)

3.สามารถป้องกันการโจมตีแบบ Frequency-based และ Sized-based โดยการแบ่งข้อมูลออกเป็นกลุ่ม และเพิ่มข้อมูลหลอกเข้าไปเพื่อทำให้ข้อมูลที่ถูกเข้ารหัสมีขนาดเท่ากัน

4.สามารถเข้ารหัสข้อมูลที่เป็นจำนวนตัวเลข โดยที่ยังคงรักษาการเรียงลำดับของข้อมูล เพื่อให้ง่ายต่อการค้นหาข้อมูล เช่น จำนวนที่มากที่สุด จำนวนที่น้อยที่สุด เป็นต้น วิธีการที่นำมาใช้คือ Order Preserving Encryption Schema (OPES)

ข้อเสีย

ใช้กุญแจหลายกุญแจในการเข้ารหัสเพื่อให้ข้อมูลที่ถูกเข้ารหัสมีความแตกต่างกัน

6) An Efficient Approach to Support Querying Secure Out Sourced XML Information (Yang และคณะ, 2006)

ในปี ค.ศ. 2006 Y. Yang และคณะ ได้เสนอวิธีการหนึ่งซึ่งมีการจัดเก็บข้อมูล Index ไว้ในฝั่งผู้ใช้ ซึ่งเป็นวิธีการในการบีบอัดข้อมูลโครงสร้างข้อมูลให้มีขนาดเล็กลงโดยใช้ Vectorization และ Skeleton Compression เพื่อที่จะสามารถประมวลผลในหน่วยความจำหลักของผู้รับบริการ นอกจากนี้แล้วยังมีกระบวนการแปลงข้อมูล XML ให้เป็น Relational Data และกระบวนการแปลง Query ให้อยู่ในรูปของ SQL

ข้อดี

1.เน้นการบีบอัดข้อมูลดัชนีให้มีขนาดเล็ก เพื่อที่จะสามารถประมวลผลในหน่วยความจำของเครื่องผู้ใช้

2. สามารถกำหนดการเข้ารหัสข้อมูลเพียงบางส่วน

3. ผู้ให้บริการไม่สามารถทราบรายละเอียดของข้อมูล

ข้อเสีย

ในบางครั้งข้อมูลดัชนีมีขนาดใหญ่เกินกว่าที่จะสามารถจัดเก็บในหน่วยความจำของผู้รับบริการ

7) XFlat: Query-friendly encrypted XML view publishing (Gao, Wang และ Yang, 2008)

ในปี ค.ศ. 2008 Gao, Wang และ Yang ได้เสนอวิธีการในการเข้ารหัสเอกสาร XML ที่ใช้สำหรับการเผยแพร่ โดยที่ผู้ใช้แต่ละกลุ่มจะมีสิทธิ์ในการเข้าถึงของเอกสาร XML ที่เผยแพร่แตกต่างกัน วิธีการนี้ไม่จำเป็นที่จะต้องคัดลอกเอกสาร XML ให้เท่ากับจำนวนกลุ่มผู้ใช้

ซึ่งวิธีการนี้จะแบ่งเอกสาร XML ออกเป็นโครงสร้างต้นไม้หลายส่วน เมื่อผู้ใช้ต้องการเข้าถึงข้อมูลจะมีการตรวจสอบสิทธิ์และประกอบโครงสร้างต้นไม้ หลังจากนั้นจะถอดรหัสข้อมูลตามสิทธิ์ของผู้ใช้ในแต่ละกลุ่ม

ข้อดี

1. ใช้เนื้อที่ในการจัดเก็บน้อย
2. สามารถใช้ XPath ในการค้นหาข้อมูล

ข้อเสีย

มีปัญหาในการค้นหาเมื่อข้อมูลอยู่ในระดับลึกของโครงสร้างต้นไม้

8) A Survey on Querying Encrypted XML Documents for Databases as a Service (Unay และ Gundem, 2008)

ในปี ค.ศ. 2008 Unay และ Gundem ได้รวบรวมงานวิจัยที่เกี่ยวกับการเข้ารหัสข้อมูลเอกสาร XML โดยที่ได้แบ่งประเภทของงานวิจัยตามลักษณะของดัชนีที่ใช้ช่วยในการค้นหา ซึ่งแบ่งออกเป็น 3 ประเภท คือ ดัชนีจัดเก็บไว้ในฝั่งของผู้ให้บริการ ดัชนีจัดเก็บไว้ในฝั่งของผู้รับบริการ และดัชนีจัดเก็บไว้ทั้งสองฝั่ง นอกจากนี้ยังได้สรุปลักษณะของสถาปัตยกรรมดังนี้

ขั้นตอน Querying โดยทั่วไปในเอกสาร XML

1. ผู้รับบริการกำหนด Query ที่ต้องการและส่งไป Query Translator เพื่อแปลง Query ให้อยู่ในรูปที่ผู้ให้บริการสามารถนำไปใช้ในการค้นหาข้อมูลในเอกสาร XML ที่ถูกเข้ารหัส
2. ผู้รับบริการส่ง Query ที่ถูกแปลงแล้วไปยังผู้ให้บริการ
3. ผู้ให้บริการใช้ข้อมูล Index ช่วยในการประมวลผล Query และค้นหาข้อมูลในเอกสาร XML ที่ถูกเข้ารหัส
4. ผู้ให้บริการส่งข้อมูลในเอกสาร XML ที่ถูกเข้ารหัสกลับไปยังผู้รับบริการ
5. ผู้รับบริการถอดรหัสข้อมูล เพื่อให้ได้ข้อมูลที่แท้จริงที่ต้องการ

จากตารางที่ 2.4 เป็นการสรุปเปรียบเทียบวิธีการของแต่ละงานวิจัยที่ได้กล่าวมาข้างต้น โดยมีการเปรียบเทียบในประเด็นต่างๆ เพื่อให้เห็นถึงความเหมือนหรือความแตกต่างในแต่ละวิธีการ

ตารางที่ 2.4 สรุปเปรียบเทียบวิธีการของแต่ละงานวิจัย

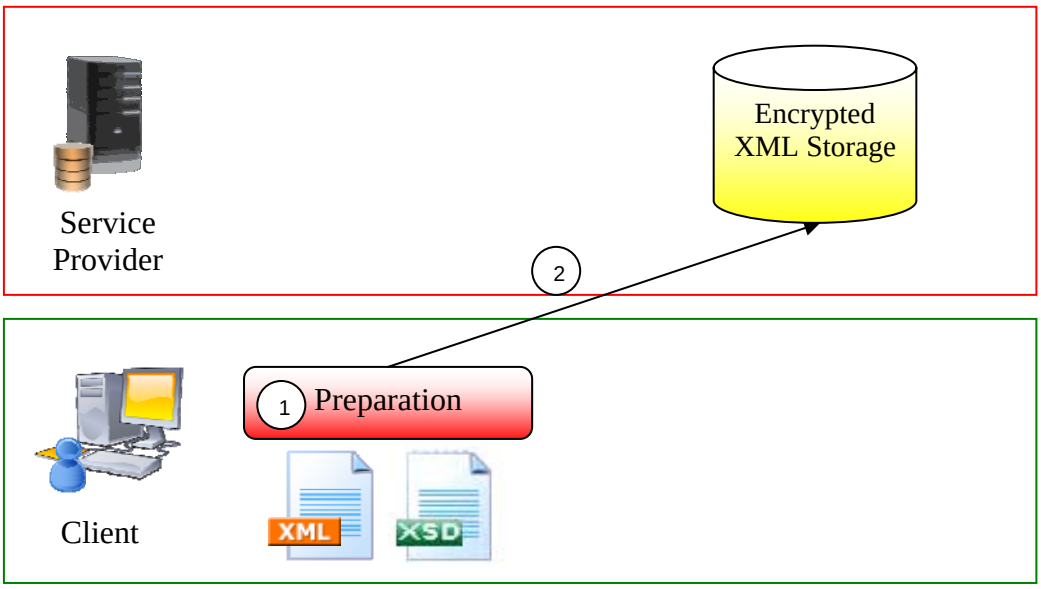
ปี ค.ศ.	นักวิจัย	การจัดเก็บดัชนี	การควบคุมการเข้าถึง	การจัดการดัชนี	การจัดเก็บข้อมูลที่ถูกเข้ารหัส	ลักษณะการเข้ารหัส	ใช้ข้อมูลโครงสร้าง XML
2003	Feng และ Jonker	ผู้ให้บริการ	ไม่ใช้	Hash Function	ฐานข้อมูล	บางส่วน	ใช้ DTD
2005	Schrefl, Grun และ Dorn	ผู้รับและผู้ให้บริการ	ใช้กุญแจในการแยกแยะ	Hash Function	ฐานข้อมูล	ทั้งหมด	ใช้ DTD หรือ XMLSchema
2005	Lee และ Whang	ผู้ให้บริการ	ใช้กุญแจในการแยกแยะ	ไม่ใช้	Flat File	บางส่วน	ไม่ใช้
2006	Jammalamadaka และ Mehrotra	ผู้ให้บริการ	ไม่ใช้	ไม่ใช้	Flat File	บางส่วน	ใช้ XMLSchema
2006	Wang และ Lakshmanan	ผู้ให้บริการ	ไม่ใช้	Tree Structure	ฐานข้อมูล	บางส่วน	ไม่ใช้
2006	Y. Yang, W. Ng, H. L. Lau, และ J. Cheng	ผู้รับบริการ	ไม่ใช้	Tree Structure	Flat File	บางส่วน	ไม่ใช้
2008	Gao, Wang และ Yang	ผู้ให้บริการ	ใช้กุญแจในการแยกแยะ	ไม่ใช้	Flat File	บางส่วน	ใช้ DTD

บทที่ 3

วิธีการเข้ารหัสข้อมูลเอกสาร XML

บทนี้จะกล่าวถึงวิธีการออกแบบวิธีการเข้ารหัสข้อมูล (qDAS: query Database As a Service) ซึ่งจะมีการจัดเก็บข้อมูลไว้ในฝั่งผู้ให้บริการและเข้ารหัสข้อมูลเฉพาะส่วนที่ต้องการเท่านั้น โดยใช้กุญแจเดียวในการเข้ารหัสข้อมูลที่เหมือนกันให้มีความแตกต่างกันเพื่อป้องกันไม่ให้ผู้โจมตีสามารถคาดเดาจำนวนข้อมูลที่เหมือนกัน เป็นการประหยัดพื้นที่ แต่ยังคงประสิทธิภาพในการค้นหา การทำงานแบ่งออกเป็น 2 กระบวนการ คือกระบวนการจัดเก็บข้อมูลเอกสาร XML (XML Data Storage) และกระบวนการค้นหาข้อมูล (Querying)

3.1 กระบวนการจัดเก็บข้อมูลเอกสาร XML (XML Data Storage)



ภาพประกอบ 3.1 กระบวนการจัดเก็บข้อมูลเอกสาร XML

กระบวนการจัดเก็บข้อมูลดังภาพประกอบ 3.1 ผู้รับบริการ (Client) จะทำการเตรียมข้อมูลและเข้ารหัสข้อมูลเฉพาะส่วนที่ต้องการในเอกสาร XML จากนั้นข้อมูลที่ได้จะถูกส่งไปจัดเก็บไว้ในฝั่งของผู้ให้บริการ (Service Provider) โดยที่ในกระบวนการเตรียมข้อมูล (Preparation) (1) ประกอบด้วย ขั้นตอนการอ่านเอกสาร XML (Read Document) ขั้นตอนการ

กำหนดหมายเลข (Assign Number) และขั้นตอนการบันทึกข้อมูล (Insert Data) ดังแสดงในภาพประกอบ 3.2

ขั้นตอนที่ 1: ขั้นตอนการอ่านเอกสาร XML (Read Document)
1.1 อ่านเอกสาร XML (.xml) 1.2 อ่านเอกสารนิยามโครงสร้างเอกสาร XML หรือ XMLSchema (.xsd)
↓
ขั้นตอนที่ 2: ขั้นตอนการกำหนดหมายเลข (Assign Number)
2.1 แปลงเอกสารให้อยู่ในรูปโครงสร้างข้อมูลแบบต้นไม้ 2.2 กำหนดหมายเลข
↓
ขั้นตอนที่ 3: ขั้นตอนการบันทึกข้อมูล (Insert Data)
3.1 เตรียมข้อมูลเพื่อการจัดเก็บ 3.2 บันทึกข้อมูลในตาราง Path 3.3 บันทึกข้อมูลในตาราง Value

ภาพประกอบ 3.2 ขั้นตอนของกระบวนการเตรียมข้อมูล

3.1.1. ขั้นตอนการอ่านเอกสาร XML (Read Document)

ในขั้นตอนนี้เอกสาร XML ในรูปของ Text File หรือ ไฟล์นามสกุล .xml จะถูกอ่านเข้ามาพร้อมกับ XML Schema ของเอกสารนั้น (ดูภาพประกอบ 3.3) โดยที่ข้อมูลในเอกสาร XML จะต้องเป็นข้อมูลที่มีความถูกต้องตามไวยากรณ์ของภาษา XML การตรวจสอบความถูกต้องจะใช้ข้อมูลจาก XML Schema

ภาพประกอบ 3.3 ขั้นตอนการอ่านเอกสาร

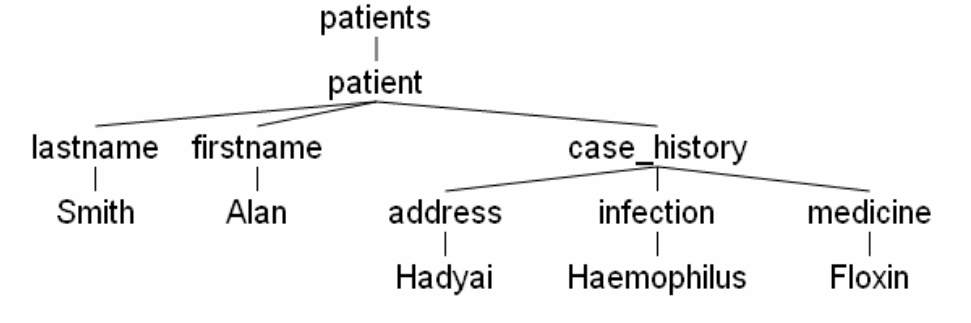
3.1.2. ขั้นตอนการกำหนดหมายเลข (Assign Number)

ในขั้นตอนนี้ข้อมูลในเอกสาร XML ดังภาพประกอบ 3.4 จะถูกแปลงให้อยู่ในรูปโครงสร้างข้อมูลแบบต้นไม้ (XML Tree) โดยใช้ XML Parser ประเภท Document Object Model (DOM) (ดูภาพประกอบ 3.5) หลังจากนั้นจะมีการกำหนดหมายเลขให้กับแต่ละโหนดโดยใช้วิธีการแบบ Prefix (Chan, 1996) และการเข้าถึงข้อมูลแบบเรียกตัวเอง (Recursive) ดัง

ภาพประกอบ 3.6 การกำหนดหมายเลขจะเริ่มต้นจากโหนดราก ซึ่งคือหมายเลข 1 และไปเรื่อยๆ จากโหนดข้อมูลซ้ายสุดไปยังขวาสุด หากข้ามไปยังโหนดลูกก็จะทำการเพิ่มหมายเลขอีก 1 ตัวต่อท้ายจะหมายเลขเดิมในทุกครั้งที่มีการเข้าถึงข้อมูลโหนดลูกของโหนดใดๆ แต่ในส่วนของข้อมูลที่อยู่ในโหนดที่ไม่มีโหนดลูก (Leaf Node) จะไม่มีการกำหนดหมายเลขให้กับโหนดนั้น วิธีการนี้จะทำให้แต่ละโหนดข้อมูลจะมีหมายเลขที่ไม่ซ้ำกัน ซึ่งจะได้ผลลัพธ์ภาพประกอบ 3.7

```
<patients>
        <patient>
                <lastname>Smith</lastname>
                <firstname>Alan</firstname>
                <case_history>
                        <address>Hadyai</address>
                        <infection>Haemophilus</infection>
                        <medicine>Floxin</medicine>
                </case_history>
        </patient>
</patients>
```

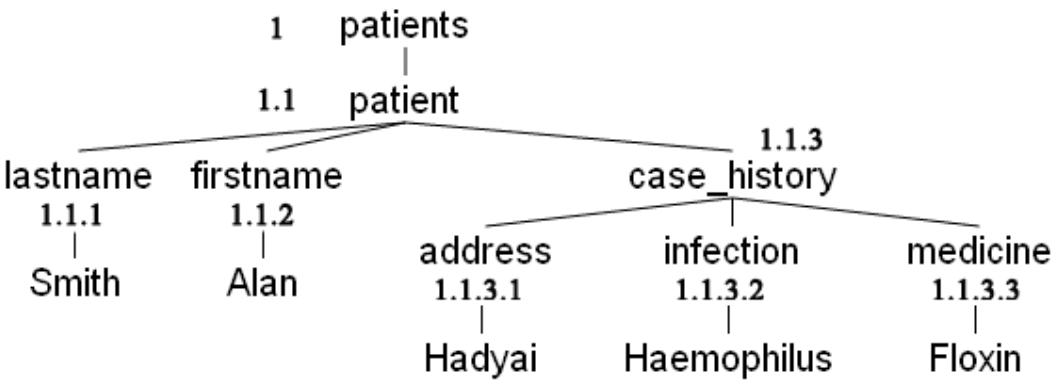
ภาพประกอบ 3.4 ตัวอย่างเอกสาร XML



ภาพประกอบ 3.5 ตัวอย่างเอกสาร XML ในรูปแบบโครงสร้างต้นไม้

```
Algorithm : Assign Number
Input : Unassign Number Tree
Output : Assign Number Tree
1. root.id = ”1”
2. identify (root, root.id)
3. identify (node, id) {
4.          children = node.getChildNodes()
5.          if (children != null) {
6.                    for (i = 1; i < children.getLength(); i++) {
7.                              children.item[i].id = id+i.toString()
8.                              identify(children.item[i], children.item[i].id)
9.                    }
10.         }else {
11.                   break;
12.          }
13.}
```

ภาพประกอบ 3.6 ขั้นตอนวิธีการกำหนดหมายเลข

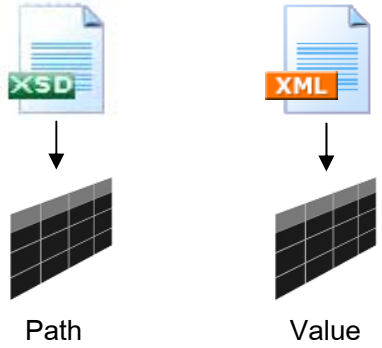


ภาพประกอบ 3.7 ตัวอย่างเอกสาร XML ในรูปแบบโครงสร้างต้นไม้เมื่อผ่านกระบวนการกำหนดหมายเลข

การกำหนดหมายเลขจะนำมาช่วยในการเข้ารหัสข้อมูลเพื่อให้ข้อมูลที่เหมือนกันถูกเข้ารหัสแล้วได้ผลลัพธ์ที่แตกต่างกัน ตัวอย่างเช่น ข้อมูล ‘yes’ กับ ‘no’ หากเข้ารหัสโดยตรงจะได้ผลลัพธ์ (Cipher Text) ที่เหมือนกันสองแบบ แต่ถ้านำข้อมูล ‘yes’ และ ‘no’ มารวมกับหมายเลขโหนดของข้อมูลแล้วนำไปเข้ารหัสจะทำให้ได้ Cipher Text ที่แตกต่างกัน นอกจากนี้สามารถนำหมายเลขที่กำหนดให้แต่ละโหนดมาช่วยในการหาความสัมพันธ์ระหว่างโหนดเพื่อช่วยในการค้นหาข้อมูล กล่าวคือสามารถระบุความเป็นพ่อแม่และลูกของแต่ละโหนดจากหมายเลขที่ถูกกำหนด โดยที่ถ้าตัดหมายเลขของโหนดลูก 1 ตำแหน่งก็จะได้หมายเลขโหนดพ่อแม่ของโหนดนั่น เช่น โหนด address มีหมายเลข 1.1.3.1 หากเราตัดหมายเลขออก 1 ตำแหน่งจะได้ 1.1.3 ซึ่งเป็นหมายเลขของ case_history (โหนดพ่อแม่ของโหนด address) ข้อดีของวิธีการนี้คือเมื่อมีการแก้ไขโครงสร้างของเอกสารเช่น การเพิ่มหรือลบข้อมูลไม่ส่งผลกระทบต่อการกำหนดหมายเลขให้กับข้อมูลใหม่ที่เข้ามา เมื่อมีข้อมูลเข้ามาใหม่ การกำหนดหมายเลขก็จะใช้หมายเลขที่ถัดจากหมายเลขสุดท้ายเดิมมากำหนดให้กับข้อมูลใหม่

3.1.3. ขั้นตอนการบันทึกข้อมูล (Insert Data)

การบันทึกข้อมูลจะบันทึกลงในตารางฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เพื่อเพิ่มความสะดวกในการจัดการและประสิทธิภาพในการค้นหาข้อมูล (Min และคณะ, 2008) ในวิทยานิพนธ์นี้จะมีการบันทึกข้อมูลลงใน 2 ตาราง คือ ตาราง Path และ ตาราง Value ดังภาพประกอบ 3.8



ภาพประกอบ 3.8 การบันทึกข้อมูล

1) ตาราง Path ประกอบด้วย 3 ฟิลด์ (Fields) ดังนี้ Path(Path_Id, Path_Name, Encrypted) โดยที่รายละเอียดต่างๆแสดงดังตารางที่ 3.1

2) ตาราง Value ประกอบด้วย 5 ฟิลด์ (Fields) ซึ่งแต่ละเรคคอร์ด (Record) จะเก็บค่าข้อมูลแต่ละโหนดของ XML Tree ที่ได้จากขั้นตอนที่ 2 โครงสร้างของตาราง Value คือ Value(Node_Id, Path_Id, Key_Name, Hash_Value , Node_Data) โดยที่รายละเอียดต่างๆ แสดงดังตารางที่ 3.2

ตารางที่ 3.1 รายละเอียดของตาราง Path

ข้อมูล	รายละเอียด
Path_Id	จัดเก็บข้อมูลหมายเลขที่ใช้สำหรับระบุพาธ โดยที่จะทำหน้าที่เป็น PrimaryKey ของตารางข้อมูลนี้และมีการเพิ่มค่าขึ้นตามลำดับการเข้ามาของข้อมูล
Path_Name	จัดเก็บข้อมูลชื่อของพาธที่ได้จาก XMLSchema โดยที่แต่ละพาธจะเป็นพาธที่มีค่าข้อมูลจัดเก็บอยู่หรือเป็นพาธที่มี Leaf Node
Encrypted	จัดเก็บข้อมูลที่ใช้ในการระบุการเข้ารหัสของ path (โดยที่ 'n' หมายถึงพาธที่ไม่มีการเข้ารหัส และ 'y' หมายถึงพาธที่มีการเข้ารหัส)

ตารางที่ 3.2 รายละเอียดของตาราง Value

ข้อมูล	รายละเอียด
Node_Id	จัดเก็บข้อมูลหมายเลขของโหมดจาก XML Tree ที่ได้จากการใช้ DOM ในการเข้าถึงและขั้นตอนการกำหนดหมายเลข (Assign Number) โดยที่จะไม่มีข้อมูลของ Node_Id ซ้ำกันและใช้เป็น Primary Key ของตารางข้อมูล
Path_Id	จัดเก็บข้อมูลหมายเลขที่ใช้สำหรับระบุพาธ และใช้เป็น Foreign Key ของระหว่างตาราง Value และตาราง Path
Key_Name	จัดเก็บชื่อกุญแจที่ใช้ในการเข้าและถอดรหัสข้อมูลในส่วนของ Value (กรณีที่ใช้มากกว่าหนึ่งกุญแจในการเข้ารหัสข้อมูล)
Hash_Value	เก็บข้อมูลผลลัพธ์ที่ได้จากการผ่านค่าข้อมูล (Node_Data) กับ Node_Id ของโหนดที่มีข้อมูลที่ต้องการปกปิดไปยังสมการที่ 1 โดยค่า Hash_Value นี้จะนำมาใช้เพื่อช่วยในการค้นหาข้อมูลในตาราง (เป็นตัวแทนของข้อมูลที่มีการเข้ารหัส)
Node_Data	จัดเก็บค่าข้อมูลของโหนดต่างๆ ในกรณีที่เป็นข้อมูลของพาธที่เข้ารหัส ข้อมูลในส่วนนี้จะเป็นข้อมูลที่เข้ารหัสและไม่สามารถอ่านเข้าใจได้ นอกจากนี้ข้อมูลที่ถูกเข้ารหัสถึงแม้จะมีค่าเหมือนกันแต่ผลลัพธ์ของการเข้ารหัสจะไม่เหมือนกัน

ในส่วนของฟังก์ชัน Hash สามารถเลือกใช้ฟังก์ชันใดๆ ซึ่งจะต้องไม่สามารถคำนวณข้อมูลที่ถูกย่อยแล้วกลับมาให้เป็นข้อมูลเดิม ซึ่งในงานวิจัยนี้เลือกฟังก์ชัน Hash Shift-

add-XOR (Ramakrishna และ Zobel, 1997) ดังภาพประกอบ 3.9 และฟังก์ชันการคำนวณหาค่า Hash_Value เป็นไปดังสมการที่ 1

$$Hash\_Value = \text{Hash}(Node\_Id + \text{Hash}(Node\_Data)) \quad (1)$$

```
Shift_add_XOR (message){
  h = 0
  for ( i=0; i < message.length; i++ )
    h ^= Shift_left(h) + Shift_right(h) + message[i]
  return h
}// ^ symbol is XOR operation
```

ภาพประกอบ 3.9 ขั้นตอนวิธี Shift-add-XOR

ในส่วนของการเข้ารหัสข้อมูลในงานวิจัยนี้จะใช้การเข้ารหัสข้อมูลแบบกุญแจสมมาตร โดยข้อมูลที่ต้องการเข้ารหัสจะถูกนำมารวมกับหมายเลขของโหนดนั้น (หมายเลขโหนดเป็นข้อมูลที่ไม่ซ้ำกัน) จะทำให้เมื่อมีการเข้ารหัสข้อมูลจะได้ Cipher Text ที่แตกต่างกัน แม้ว่าจะเป็นข้อมูลเดียวกันและใช้กุญแจเดียวกันในการเข้ารหัส โดยใช้ฟังก์ชัน Encryption (สมการที่ 2) ในการเข้ารหัสข้อมูล

$$Encrypted\_Data = \text{Encrytion}(Node\_Id + Node\_Data, Key\_Name) \quad (2)$$

ในส่วนของการถอดรหัสข้อมูลซึ่งจะเกิดขึ้นในกรณีที่ค้นหาข้อมูลแล้วได้ผลลัพธ์เป็นข้อมูลที่มีการเข้ารหัส ผู้รับบริการจะต้องดำเนินการถอดรหัสข้อมูลโดยใช้ฟังก์ชัน Decryption (สมการที่ 3) ซึ่งจะใช้กุญแจเดียวกันกับที่ใช้ในการเข้ารหัส

$$Node\_Id+Node\_Data = \text{Decrytion}(Encrypted\_Data, Key\_Name) \quad (3)$$

ตัวอย่างเช่น เอกสาร XML ดังภาพประกอบ 3.4 หลังจากผ่านกระบวนการจัดเก็บข้อมูลจะได้ตาราง Path และตาราง Value ดังแสดงในตารางที่ 3.3 และ 3.4 ตามลำดับ ซึ่งตาราง Path ได้มาจากข้อมูลใน XML Schema และตาราง Value ได้มาจาก XML Tree สองตารางนี้ จะถูกจัดเก็บไว้ที่ฝั่งผู้ให้บริการ รายละเอียดของข้อมูลอธิบายได้ดังนี้ ถ้าข้อมูลเป็นข้อมูลที่ไม่เข้ารหัส (ไม่ปกปิด) เช่น โหนด lastname ในตาราง Path จะจัดเก็บค่า Path_Id ของโหนด lastname เป็น 1 ค่า Path_Name เป็น Patients/Patient/lastname และค่า Encrypted

เป็น ‘n’ ในตาราง Value จะจัดเก็บค่า Node_Id เป็น 1.1.1 ค่า Path_Id เป็น 1 และ ค่า Node_Data เป็น “Smith” โดยที่ไม่ต้องเก็บค่า Key_Name และ Hash_Value

ถ้าข้อมูลเป็นข้อมูลที่เข้ารหัส (ปกปิด) เช่น โหนด medicine ในตาราง Path จะจัดเก็บค่า Path_Id ของโหนด medicine เป็น 5 ค่า Path_Name เป็น Patients/Patient/case_history/medicine และค่า Encrypted เป็น ‘y’ ในตาราง Value จะจัดเก็บค่า Node_Id เป็น 1.1.3.3 ค่า Path_Id เป็น 5 ค่า Key_Name เป็น “Name_key1” ค่า Hash_Value เป็น 326514789 (นั่นคือ Hash(“1.1.3.3”+ Hash(“Floxin”))) และค่า Node_Data เป็น “&*()_+” (นั่นคือ Encryption(“1.1.3.3Floxin”, “Name_key1”))

ตารางที่ 3.3 ตัวอย่างตาราง Path

Path_Id	Path_Name	Encrypted
1	/Patients/Patient/lastname	n
2	/Patients/Patient/firstname	n
3	/Patients/Patient/case_history/address	n
4	/Patients/Patient/case_history/infection	y
5	/Patients/Patient/case_history/medicine	y

ตารางที่ 3.4 ตัวอย่างตาราง Value

Node_Id	Path_Id	Key_Name	Hash_Value	Node_Data
1.1.1	1			Smith
1.1.2	2			Alan
1.1.3.1	3			Hadyai
1.1.3.2	4	Name_key1	142357869	!@#$%^
1.1.3.3	5	Name_key1	326514789	&*()_+

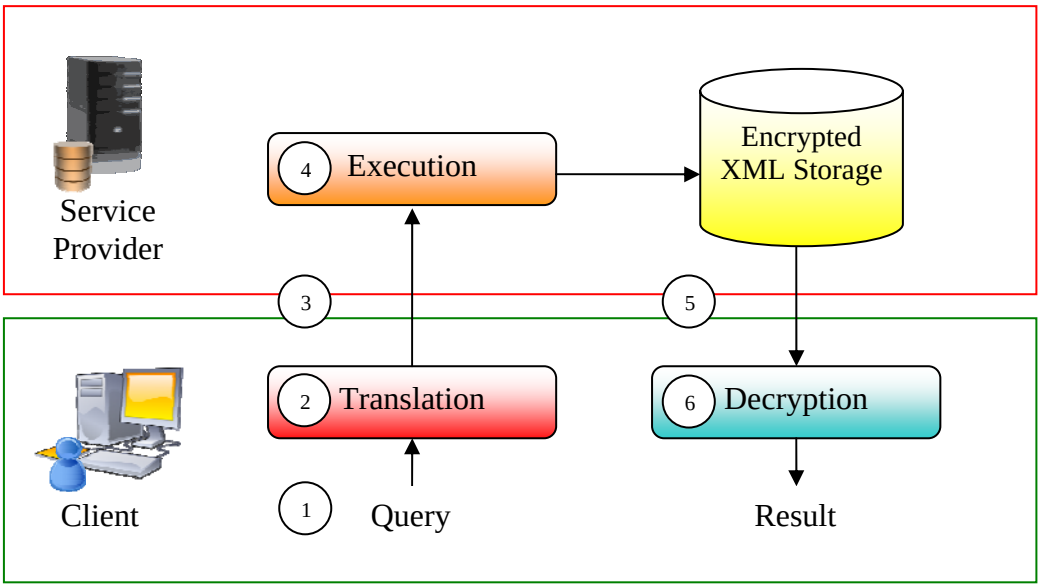
3.1.4. การจัดเก็บข้อมูลในฝั่งของผู้ให้บริการและผู้รับบริการ

จากภาพประกอบ 3.1 หลังจากกระบวนการเตรียมข้อมูลจะเป็นการจัดเก็บข้อมูล (2) โดยจะมีการแบ่งการจัดเก็บข้อมูลดังนี้

1) ข้อมูลที่จัดเก็บไว้ในฝั่งผู้ให้บริการประกอบด้วย ตาราง Path, ตาราง Value และฟังก์ชัน Hash

2) ข้อมูลที่จัดเก็บไว้ในฝั่งผู้รับบริการประกอบด้วย กุญแจ, ตาราง Path_Client (ข้อมูลเฉพาะส่วนพาธที่เข้ารหัส), ฟังก์ชันเข้ารหัสข้อมูล (Encryption), ฟังก์ชันถอดรหัสข้อมูล (Decryption) และฟังก์ชัน Hash

3.2 กระบวนการค้นหาข้อมูล (Querying)



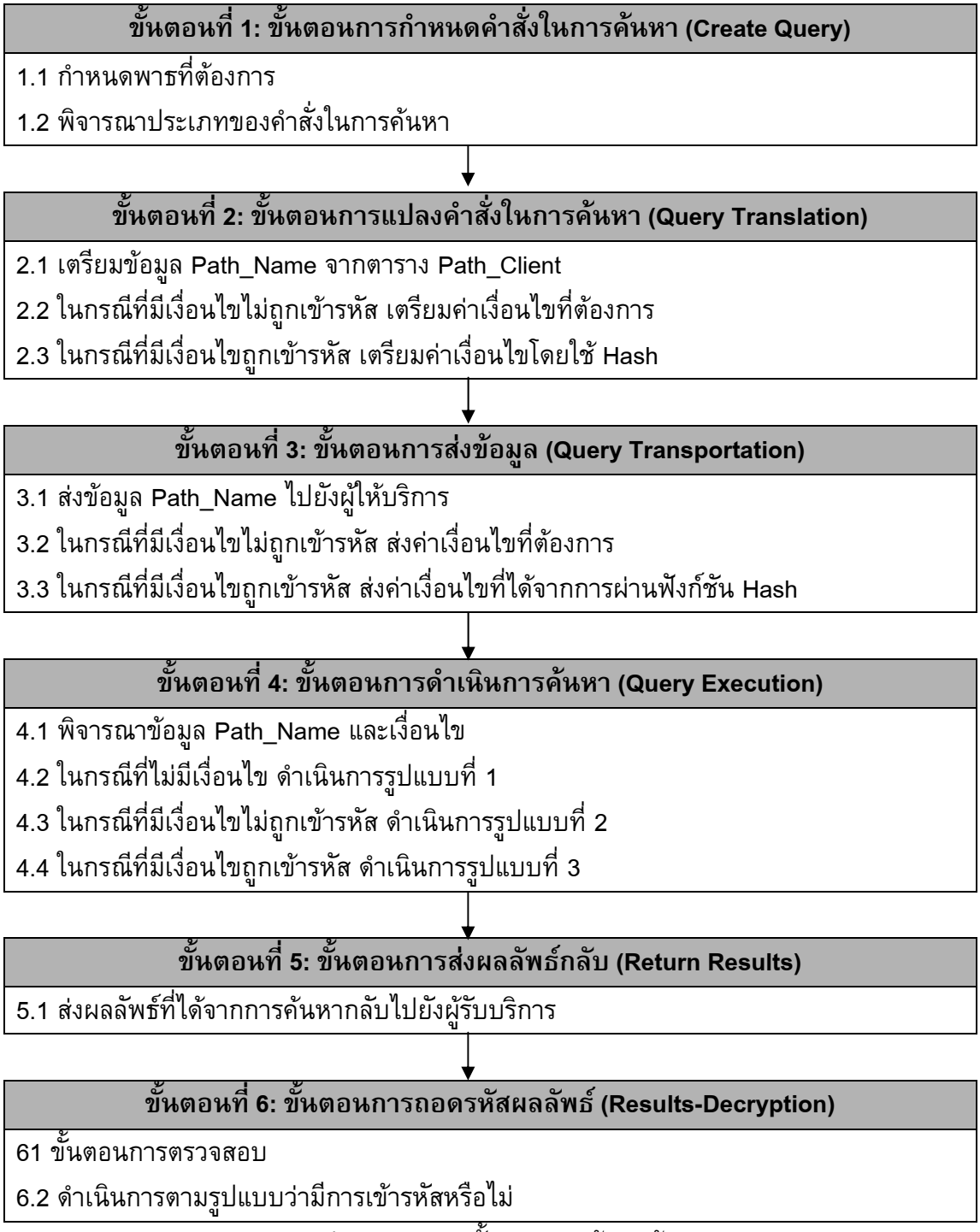
ภาพประกอบ 3.10 สถาปัตยกรรมการค้นหา
(หมายเลขแสดงลำดับการดำเนินการ)

ในกระบวนการค้นหาข้อมูลนี้จะต้องมีการแปลงข้อมูลก่อนนำส่งไปยังผู้ให้บริการ เพื่อให้สามารถค้นหาข้อมูลในฝั่งของผู้ให้บริการ เนื่องจากข้อมูลบางส่วนนั้นเป็นข้อมูลที่มีการเข้ารหัส นอกจากนี้รูปแบบในการค้นหาข้อมูลในเอกสาร XML มีอยู่หลากหลายรูปแบบ การค้นหาข้อมูลโดยทั่วไปของเอกสาร XML จะใช้ XPath และ XQuery ซึ่งเมื่อเข้ารหัสข้อมูลด้วยวิธีการ XML Encryption จะไม่สามารถค้นหาข้อมูลได้โดยตรงจะต้องถอดรหัสข้อมูลก่อน แต่วิธีการนี้สามารถค้นหาข้อมูลโดยที่ไม่ต้องถอดรหัสและส่งคำร้องในการค้นหาเพียงครั้งเดียวเมื่อต้องการผลลัพธ์ทั้งหมด ซึ่งสถาปัตยกรรมการค้นหาแสดงดังภาพประกอบ 3.10 และภาพประกอบ 3.11 อธิบายขั้นตอนการค้นหาข้อมูล

3.2.1 ขั้นตอนการกำหนดคำสั่งในการค้นหา (Create Query)

ในขั้นตอนนี้ผู้รับบริการจะมีการกำหนดคำสั่งในการค้นหา (Query) เพื่อให้ได้ผลลัพธ์ตามที่ต้องการโดยคำสั่งในการค้นหาจะแบ่งออกเป็น 2 รูปแบบคือ คำสั่งในการค้นหา

แบบไม่มีเงื่อนไข (Unconditional Query: UQ) และคำสั่งในการค้นหาแบบมีเงื่อนไข (Conditional Query: CQ) ตารางที่ 3.5 แสดงตัวอย่าง Query โดยที่ Q1-Q2 เป็นแบบ UQ ส่วน Q3-Q6 เป็นแบบ CQ ซึ่งคำสั่งแต่ละรูปแบบจะมีการดำเนินการในขั้นตอนต่อไปที่แตกต่างกัน โดยทั่วไปแล้วสามารถแบ่งรูปแบบการค้นหาข้อมูลได้ 6 รูปแบบ ดังตารางที่ 3.6



ภาพประกอบ 3.11 ขั้นตอนการค้นหาข้อมูล

ตารางที่ 3.5 ตัวอย่าง Query

Q1	/Patients/Patient/lastname
Q2	/Patients/Patient/case_history/infection
Q3	/Patients/Patient[lastname='Smith']/case_history/address
Q4	/Patients/Patient[lastname='Smith']/case_history/infection
Q5	/Patients/Patient/case_history[infection='Haemophilus']/address
Q6	/Patients/Patient/case_history[infection='Haemophilus']/medicine

จากตารางที่ 3.5 คำอธิบายของคำสั่งในการค้นหามีดังต่อไปนี้

1) การค้นหาประเภทไม่มีเงื่อนไขในการค้นหาและผลลัพธ์เป็นข้อมูลที่ไม่ถูกเข้ารหัส (Unconditional query with unencrypted result)

ในการค้นหาประเภทนี้จะไม่มีเงื่อนไขในการสอบถามข้อมูลและต้องการแสดงข้อมูลทั้งหมดที่อยู่ใน path โดยที่ path ที่ถูกระบุจะเป็น path ที่ไม่มีการเข้ารหัสข้อมูล ตัวอย่างเช่น /Patients/Patient/lastname ซึ่งคำตอบของ path นี้จะเป็นชื่อสกุลของผู้ป่วยทั้งหมดที่มีอยู่ในเอกสารโดยที่ข้อมูลชื่อเป็นข้อมูลที่ไม่มีการเข้ารหัส

2) การค้นหาประเภทไม่มีเงื่อนไขในการค้นหาและผลลัพธ์เป็นข้อมูลที่ถูกเข้ารหัส (Unconditional query with encrypted result)

ในการค้นหาประเภทนี้จะไม่มีเงื่อนไขในการสอบถามข้อมูลและต้องการแสดงข้อมูลทั้งหมดที่อยู่ใน path โดยที่ path ที่ถูกระบุจะเป็น path ที่มีการเข้ารหัสข้อมูลไว้ ตัวอย่างเช่น /Patients/Patient/case_history/infection ซึ่งคำตอบของ path นี้จะเป็นข้อมูลการติดเชื้อทั้งหมดที่มีอยู่ในเอกสารโดยที่ข้อมูลการติดเชื้อเป็นข้อมูลที่มีการเข้ารหัส

3) การค้นหาประเภทมีเงื่อนไขในการค้นหาโดยเงื่อนไขนั้นไม่ถูกเข้ารหัสและผลลัพธ์เป็นข้อมูลที่ไม่ถูกเข้ารหัส (Unencrypted conditional query with unencrypted result)

ในการค้นหาประเภทนี้จะมีการระบุเงื่อนไขในการสอบถามข้อมูลโดยจะแสดงข้อมูลเฉพาะส่วนที่ตรงตามเงื่อนไข ซึ่งผลลัพธ์ที่ต้องการเป็นข้อมูลที่ไม่มีการเข้ารหัส ตัวอย่างเช่น /Patients/Patient[lastname='Smith']/case_history/address ซึ่งเงื่อนไขของ path นี้คือบุคคลที่มีชื่อสกุล Smith และต้องการทราบที่อยู่ โดยที่ข้อมูลชื่อสกุลและข้อมูลที่อยู่เป็นข้อมูลที่ไม่ถูกเข้ารหัส

4) การค้นหาประเภทมีเงื่อนไขในการค้นหาโดยเงื่อนไขนั้นไม่ถูกเข้ารหัสและผลลัพธ์เป็นข้อมูลที่ถูกเข้ารหัส (Unencrypted conditional query with encrypted result)

ในการค้นหาประเภทนี้จะมีการระบุเงื่อนไขในการสอบถามข้อมูลโดยจะแสดงข้อมูลเฉพาะส่วนที่ตรงตามเงื่อนไข ซึ่งผลลัพธ์ที่ต้องการเป็นข้อมูลที่มีการเข้ารหัสไว้ ตัวอย่างเช่น /Patients/Patient[lastname='Smith']/case_history/infection ซึ่งเงื่อนไขของ path นี้คือบุคคลที่มีชื่อสกุล Smith และต้องการทราบการติดเชื้อ โดยที่ข้อมูลชื่อสกุลเป็นข้อมูลที่ไม่ถูกเข้ารหัส แต่ข้อมูลการติดเชื้อเป็นข้อมูลที่ถูกเข้ารหัส

5) การค้นหาประเภทมีเงื่อนไขในการค้นหาโดยเงื่อนไขนั้นถูกเข้ารหัสและผลลัพธ์เป็นข้อมูลที่ไม่ถูกเข้ารหัส (Encrypted conditional query with unencrypted result)

ในการค้นหาประเภทนี้จะมีการระบุเงื่อนไขในการสอบถามข้อมูลโดยจะแสดงข้อมูลเฉพาะส่วนที่ตรงตามเงื่อนไขโดยที่เงื่อนไขเป็นข้อมูลที่เข้ารหัส และผลลัพธ์ที่ต้องการเป็นข้อมูลที่ไม่มีการเข้ารหัส ตัวอย่างเช่น /Patients/Patient/case_history[infection='Haemophilus']/address ซึ่งเงื่อนไขของ path นี้คือบุคคลที่ติดเชื้อ Haemophilus และต้องการทราบที่อยู่ โดยที่ข้อมูลการติดเชื้อเป็นข้อมูลที่ถูกเข้ารหัส แต่ข้อมูลที่อยู่เป็นข้อมูลที่ไม่ถูกเข้ารหัส

6) การค้นหาประเภทมีเงื่อนไขในการค้นหาโดยเงื่อนไขนั้นถูกเข้ารหัสและผลลัพธ์เป็นข้อมูลที่ถูกเข้ารหัส (Encrypted conditional query with encrypted result)

ในการค้นหาประเภทนี้จะมีการระบุเงื่อนไขในการสอบถามข้อมูลโดยจะแสดงข้อมูลเฉพาะส่วนที่ตรงตามเงื่อนไขโดยที่เงื่อนไขเป็นข้อมูลที่เข้ารหัส และผลลัพธ์ที่ต้องการเป็นข้อมูลที่มีการเข้ารหัสไว้ ตัวอย่างเช่น /Patients/Patient/case_history[infection='Haemophilus']/medicine ซึ่งเงื่อนไขของ path นี้คือบุคคลที่ติดเชื้อ Haemophilus และต้องการทราบยาที่ใช้ในการรักษา โดยที่ข้อมูลการติดเชื้อและข้อมูลยาเป็นข้อมูลที่ถูกเข้ารหัส

โดยสรุปแล้วคำสั่งในการค้นหาข้อมูลสามารถแสดงได้ดังตารางที่ 3.6

ตารางที่ 3.6 สรุปรูปแบบการค้นหาข้อมูล

Result \ Condition	**คำสั่งแบบไม่มีเงื่อนไข (UQ)**	**คำสั่งแบบมีเงื่อนไข (CQ)**	
		ไม่เข้ารหัส	**เข้ารหัส**
ผลลัพธ์แบบไม่เข้ารหัส	Q1	Q3	Q5
ผลลัพธ์แบบเข้ารหัส	Q2	Q4	Q6

3.2.2 ขั้นตอนการแปลงคำสั่งในการค้นหา (Query Translation)

ในขั้นตอนนี้ผู้รับบริการเตรียมค่าข้อมูล Path_Name จากตาราง Path_Client โดยพิจารณาว่าพาธที่ต้องการเป็นพาธที่มีการเข้ารหัสหรือไม่ โดยข้อมูลในตาราง Path_Client จะจัดเก็บเฉพาะข้อมูลของพาธที่มีการเข้ารหัสเท่านั้น ในกรณีที่เงื่อนไขถูกเข้ารหัสจะมีการนำค่าเงื่อนไขผ่านฟังก์ชัน Hash และได้ผลลัพธ์ที่จะถูกส่งเพื่อเป็นตัวแทนในการเปรียบเทียบข้อมูลที่ถูกเข้ารหัส

3.2.3 ขั้นตอนการส่งข้อมูล (Query Transportation)

เมื่อผู้รับบริการเตรียมข้อมูลคำสั่งที่จะใช้ในการค้นหาเสร็จสิ้น หลังจากนั้นผู้รับบริการส่งข้อมูลไปยังผู้ให้บริการ ซึ่งข้อมูลที่ถูกส่งออกไปจะมีลักษณะที่แตกต่างกันขึ้นอยู่กับประเภทของคำสั่งในการค้นหา

3.2.4 ขั้นตอนการดำเนินการค้นหา (Query Execution)

ในขั้นตอนการค้นหาซึ่งจะดำเนินการในฝั่งของผู้ให้บริการ โดยที่จะแบ่งรูปแบบการค้นหาออกตามประเภทออกเป็น 6 ประเภท คือ

1) กรณีที่ไม่มีเงื่อนไขและผลลัพธ์แบบไม่เข้ารหัส (Q1) จะมีการค้นหาข้อมูลโดยอ้างอิงจากหมายเลข Path_Id โดยที่จะดำเนินการค้นหาหมายเลขพาธจากตาราง Path ในฝั่งของผู้ให้บริการ เมื่อได้หมายเลขพาธแล้วก็จะนำไปค้นหาในตาราง Value อีกครั้ง โดยที่มีคำสั่งในการค้นหาดังภาพประกอบ 3.12

```
1. path_id = get_id(Path_Name) // Find Path_Id from Path table
2. execute SQL select Node_Data from path_value where Path_Id = path_id
```

ภาพประกอบ 3.12 ขั้นตอนการค้นหากรณีที่ไม่มีเงื่อนไขและผลลัพธ์แบบไม่เข้ารหัส (Q1)

2) กรณีที่ไม่มีเงื่อนไขและผลลัพธ์แบบเข้ารหัส (Q2) จะมีการดำเนินการเช่นเดียวกับ Q1 แต่จะมีการค้นหา Node_Id และ Key_Name เพิ่มเติมสำหรับใช้ในการถอดรหัส โดยที่มีคำสั่งในการค้นหาดังภาพประกอบ 3.13

1. *path_id* = get_id(*Path_Name*) // Find Path_Id from Path table
2. execute SQL select Node_Data, Node_Id, KeyName from path_value where Path_Id = *path_id*

ภาพประกอบ 3.13 ขั้นตอนการค้นหากรณีที่ไม่มีเงื่อนไขและผลลัพธ์แบบเข้ารหัส (Q2)

3) กรณีที่มีเงื่อนไขซึ่งไม่เข้ารหัสและผลลัพธ์แบบไม่เข้ารหัส (Q3) เริ่มต้นจะแบ่งคำสั่งในการค้นหาออกเป็น 2 ส่วนคือ ส่วนที่เป็นเงื่อนไขและส่วนที่เป็นผลลัพธ์ที่ต้องการ หลังจากนั้นจะมีการค้นหาหมายเลขพาธของเงื่อนไขและผลลัพธ์จากตาราง Path ในฝั่งของผู้ให้บริการ เมื่อได้หมายเลขพาธทั้งสองแล้วจะมีการดำเนินการเพื่อหาความสัมพันธ์ระหว่างโหนดข้อมูลของแต่ละพาธว่าโหนดที่พิจารณาอยู่นั้นเป็นโหนดที่มีโหนดพ่อแม่เดียวกันหรือไม่ โดยที่มีคำสั่งในการค้นหาดังภาพประกอบ 3.14 ในที่นี้ interval หมายถึงความแตกต่างระหว่างพาธที่เป็นเงื่อนไขและพาธที่เป็นผลลัพธ์ ตัวอย่างเช่น Patients/Patient/lastname (มี 3 ระดับ) และ Patients/Patient/case_history/address (มี 4 ระดับ) ดังนั้น interval จะมีค่าเท่ากับ 1

1. split Condition Query into Condition and Unconditon Query
2. *condition_id* = get_id(*Condition_Path*) // Find Path_Id from Path table
3. *result_id* = get_id(*Uncondition_Query*) // Find Path_Id from Path table
4. excute SQL select *v1*.Node_Data from Value *v1*, Value *v2* where
SUBSTR(*v1*.Node_Id, 1, Length(*v1*.Node_Id) - (*interval* + 1))
like SUBSTR(*v2*.Node_Id, 1, Length(*v2*.Node_Id))
AND *v1*.Path_Id =*result_id*
AND *v2*.Value=*Condition*
AND *v2*.Path_Id=*condition_id*

ภาพประกอบ 3.14 ขั้นตอนการค้นหากรณีที่มีเงื่อนไขซึ่งไม่เข้ารหัสและผลลัพธ์แบบไม่เข้ารหัส (Q3)

4) กรณีที่มีเงื่อนไขซึ่งไม่เข้ารหัสและผลลัพธ์แบบไม่เข้ารหัส (Q4) จะมีการดำเนินการเช่นเดียวกับ Q3 แต่จะมีการค้นหา Node_Id และ Key_Name เพิ่มเติมสำหรับใช้ในการถอดรหัส โดยที่มีคำสั่งในการค้นหาดังภาพประกอบ 3.15

5) กรณีที่มีเงื่อนไขซึ่งเข้ารหัสและผลลัพธ์แบบไม่เข้ารหัส (Q5) เริ่มต้นจะแบ่งคำสั่งในการค้นหาออกเป็น 2 ส่วนและค้นหาหมายเลขพาธของเงื่อนไขและผลลัพธ์จากตาราง Path ในฝั่งของผู้ให้บริการเช่นเดียวกับกรณี 2 หลังจากนั้นจะค้นหาหมายเลยโหนดที่ตรงตามเงื่อนไข

และใช้ค่า Hash_Value ร่วมในการค้นหา ถัดจากนั้นจะการดำเนินการเพื่อหาความสัมพันธ์ระหว่างโหนดข้อมูลของแต่ละพาธว่าโหนดที่พิจารณาอยู่นั้นเป็นโหนดที่มีโหนดพ่อแม่เดียวกันหรือไม่ โดยที่มีการกำหนดคำสั่งในการค้นหาดังภาพประกอบ 3.16

1. split Condition Query into Condition and Unconditon Query
2. *condition_id* = get_id(*Condition_Path*) // Find Path_Id from Path table
3. *result_id* = get_id(*Uncondition_Query*) // Find Path_Id from Path table
4. excute SQL select *v1*.Node_Data, v1.Key_Name, v1.Node_Id from Value *v1*, Value *v2* where SUBSTR(*v1*.Node_Id, 1, Length(*v1*.Node_Id) - (*interval* + 1))
like SUBSTR(*v2*.Node_Id, 1, Length(*v2*.Node_Id))
AND *v1*.Path_Id =*result_id*
AND *v2*.Value=*Condition*
AND *v2*.Path_Id=*condition_id*

ภาพประกอบ 3.15 ขั้นตอนการค้นหากรณีที่มีเงื่อนไขซึ่งไม่เข้ารหัสและผลลัพธ์แบบเข้ารหัส (Q4)

1. split Condition Query into Condition and Unconditon Query
2. *condition_id* = get_id(*Condition_Path*)
3. *result_id* = get_id(*Uncondition_Query*)
4. *Node_id* = get_set_node_id(*condition_id*)
5. do until last of set *Node_id*
5.1 *Hash_Value* = get_Hash_Value(*Node_id*)
5.2 if *Hash_Value* equal Hash(*Node_id* + Hash(*Node_Data*))
5.3 excute SQL select *v1*.Node_Data from Value *v1*, Value *v2* where SUBSTR(*v1*.Node_Id, 1, Length(*v1*.Node_Id) - (*interval* + 1))
like SUBSTR(*v2*.Node_Id, 1, Length(*v2*.Node_Id))
AND *p2*.Path_Id = *condition_id*
AND *p2*.Node_Id = *Node_id*
AND *p1*.Path_Id = *result_id*

ภาพประกอบ 3.16 ขั้นตอนการค้นหากรณีที่มีเงื่อนไขซึ่งเข้ารหัสและผลลัพธ์แบบไม่เข้ารหัส (Q5)

6) กรณีที่มีเงื่อนไขซึ่งเข้ารหัสและผลลัพธ์แบบเข้ารหัส (Q6) จะมีการดำเนินการเช่นเดียวกับ Q5 แต่จะมีการค้นหา Node_Id และ Key_Name เพิ่มเติมสำหรับใช้ในการถอดรหัส โดยที่มีคำสั่งในการค้นหาดังภาพประกอบ 3.17

1. split Condition Query into Condition and Unconditon Query
2. *condition_id* = get_id(*Condition_Path*)
3. *result_id* = get_id(*Uncondition_Query*)
4. *Node_id* = get_set_node_id(*condition_id*)
5. do until last of set *Node_id*

5.1 *Hash_Value* = get_Hash_Value(*Node_id*)

5.2 if *Hash_Value* equal Hash(*Node_id* + Hash(*Node_Data*))

5.3 excute SQL select *v1*.Node_Data, *v1*.Key_Name, *v1*.Node_Id from Value *v1*, Value *v2* where SUBSTR(*v1*.Node_Id, 1, Length(*v1*.Node_Id) - (*interval* + 1))
like SUBSTR(*v2*.Node_Id, 1, Length(*v2*.Node_Id))
AND *p2*.Path_Id = *condition_id*
AND *p2*.Node_Id = *Node_id*
AND *p1*.Path_Id = *result_id*

ภาพประกอบ 3.17 ขั้นตอนการค้นหากรณีที่มีเงื่อนไขซึ่งเข้ารหัสและผลลัพธ์แบบเข้ารหัส (Q6)

3.2.5 ขั้นตอนการส่งผลลัพธ์กลับ (Return Results)

เมื่อได้ผลลัพธ์ตามคำสั่งที่ผู้รับบริการร้องขอมา ผู้ให้บริการส่งผลลัพธ์กลับไปยังผู้รับบริการ โดยรูปแบบของผลลัพธ์จะอยู่ 2 รูปแบบคือ ผลลัพธ์ที่มีการเข้ารหัสและผลลัพธ์ที่ไม่มีการเข้ารหัส

1) กรณีผลลัพธ์ที่มีการเข้ารหัสจะมีการส่งข้อมูล Node_Data, Node_Id และ Key_Name ที่ได้จากการค้นหากลับไปยังผู้รับบริการ

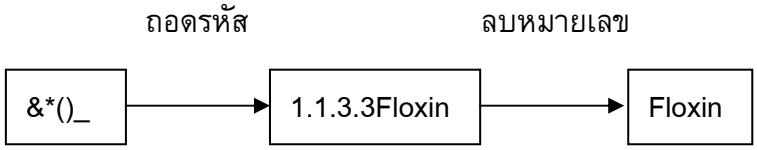
2) กรณีผลลัพธ์ที่ไม่มีการเข้ารหัสจะมีการส่งข้อมูลเฉพาะ Node_Data เท่านั้น

3.2.6 ขั้นตอนการถอดรหัสผลลัพธ์ (Results-Decryption)

เนื่องจากข้อมูลที่เป็นผลลัพธ์ที่ได้จากผู้ให้บริการในบางครั้งเป็นข้อมูลที่มีการเข้ารหัส โดยที่ขั้นตอนนี้จะเกิดขึ้นเฉพาะกรณีที่ข้อมูลมีการเข้ารหัส ดังนั้นก่อนการแสดงผลจะต้องมีการถอดรหัสข้อมูลให้อยู่ในรูปที่สามารถอ่านได้เข้าใจ ขั้นตอนกระบวนการถอดรหัสข้อมูลมีดังภาพประกอบ 3.18 และ 3.19 ซึ่งสมการ (3) เป็นสมการที่ใช้ในการถอดรหัสข้อมูล

ขั้นตอนที่ 6.1: ขั้นตอนการตรวจสอบ	
● พิจารณาว่าข้อมูลที่ได้รับกลับมาเป็นข้อมูลที่มีการเข้ารหัสหรือไม่	
ขั้นตอนที่ 6.2: กรณีที่ข้อมูลไม่เข้ารหัส	**ขั้นตอนที่ 6.2: กรณีที่ข้อมูลเข้ารหัส**
● แสดงผลลัพธ์ของข้อมูล	● ถอดรหัสข้อมูลโดยค้นหากุญแจจากชื่อกุญแจที่ได้รับจากผู้ให้บริการ โดยใช้สมการ 3 ● ลบข้อมูลหมายเลขโหนดออกจากข้อมูล ● แสดงผลลัพธ์ของการค้นหา

ภาพประกอบ 3.18 ขั้นตอนกระบวนการถอดรหัสข้อมูล



ภาพประกอบ 3.19 ตัวอย่างการถอดรหัสข้อมูล

เนื่องจากรูปแบบค้นหาข้อมูลที่หลากหลาย ทำให้การค้นหาในแต่ละรูปแบบมีความแตกต่างกันขึ้นอยู่ลักษณะของ Query โดยในขั้นตอนที่ 2-4 จะมีรายละเอียดแตกต่างกันไปดังนี้

1) กรณีที่ Query เป็นแบบ UQ เช่น Q1 และ Q2 ในตารางที่ 3 โดยที่ Q1 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q2 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส จะมีการเปลี่ยนแปลงขั้นตอนดังนี้

ขั้นตอนที่ 2 เตรียมค่า Path_Name

ขั้นตอนที่ 3 ส่งค่า Path_Name ไปยังผู้ให้บริการ

ขั้นตอนที่ 4 ค้นหาข้อมูลโดยใช้ค่า Path_Name โดยใช้ขั้นตอนวิธีดังภาพประกอบ 3.12 ในกรณีที่เป็น Q1 หรือใช้ขั้นตอนวิธีดังภาพประกอบ 3.13 ในกรณีที่เป็น Q2

2) กรณีที่ Query เป็นแบบ CQ เช่น Q3-Q6 ในตารางที่ 3

เงื่อนไขไม่ถูกเข้ารหัส เช่น Q3-Q4 โดยที่ Q3 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q4 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส จะมีการเปลี่ยนแปลงขั้นตอนดังนี้

ขั้นตอนที่ 2 เตรียมค่า Path_Name จากตาราง Path_Client และ ค่าข้อมูลของเงื่อนไข (ค่า Node_Data)

ขั้นตอนที่ 3 ส่งค่า Path_Name พร้อมกับค่าข้อมูลของเงื่อนไข

ขั้นตอนที่ 4 นำค่า Path_Name ไปค้นหา Node_Id จากนั้นใช้ขั้นตอนวิธีดังภาพประกอบ 3.14 ในกรณีที่เป็น Q3 หรือใช้ขั้นตอนวิธีดังภาพประกอบ 3.15 ในกรณีที่เป็น Q4

เงื่อนไขถูกเข้ารหัส เช่น Q5-Q6 โดยที่ Q5 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q6 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส จะมีการเปลี่ยนแปลงขั้นตอนดังนี้

ขั้นตอนที่ 2 เตรียมค่า Path_Name จากตาราง Path_Client และ Hash(ค่า Node_Data)

ขั้นตอนที่ 3 ส่งค่า Path_Name พร้อมกับ ผลลัพธ์ของค่า Hash(ค่า Node_Data)

ขั้นตอนที่ 4 นำค่า Path_Name ไปค้นหา Node_Id จากนั้นทำการคำนวณหาค่า Hash_Value โดยใช้สมการ 1 (นำแต่ละค่าของ Node_Id ไปรวมกับ ผลลัพธ์ของค่า Hash(ค่า Node_Data) แล้วผ่านฟังก์ชัน Hash) และทำการค้นหาข้อมูลในตาราง Value โดยใช้ขั้นตอนวิธีดังภาพประกอบ 3.16 ในกรณีที่เป็น Q5 หรือใช้ขั้นตอนวิธีดังภาพประกอบ 3.17 ในกรณีที่เป็น Q6

บทที่ 4

ผลการทดลองและบทวิจารณ์

บทนี้จะนำเสนอชุดข้อมูลที่ใช้ในการทดลอง การออกแบบการทดลองและผลลัพธ์ที่ได้จากการทดลองของระบบที่ได้ออกแบบไว้ โดยเปรียบเทียบกับวิธีการของ XML Encryption ซึ่งเป็นมาตรฐานที่ถูกกำหนดโดย W3C และเปรียบเทียบกับวิธีการ SemCrypt ซึ่งเป็นวิธีการที่สามารถเข้ารหัสข้อมูลเหมือนกันให้มีความแตกต่างกัน แต่มีข้อจำกัดในการแสดงผลข้อมูลที่เข้ารหัสทั้งหมด โดยในการทดลองจะใช้ชุดข้อมูลที่สร้างขึ้นจาก XMark (Schmidt, 2001) ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างเอกสาร XML สำหรับการทดลอง

วิธีการ XML Encryption (W3C, 2002) เป็นมาตรฐานในการเข้ารหัสเอกสาร XML แต่จะไม่สามารถค้นหาข้อมูลบนเอกสารที่เข้ารหัสได้โดยตรง โดยวิธีการนี้จะต้องถอดรหัสข้อมูลก่อนการค้นหา

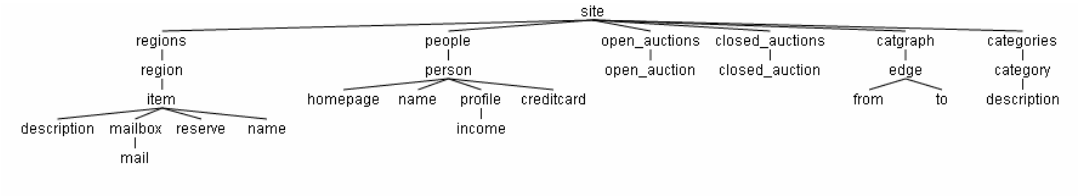
วิธีการ SemCrypt (Schrefl, Gruen และ Dorn, 2005) เป็นวิธีการที่มีความใกล้เคียงกับวิธีการที่ได้ออกแบบไว้กล่าวคือ สามารถเข้ารหัสข้อมูลเหมือนกันให้มีความแตกต่างกัน แต่จะมีข้อจำกัดในการทำงานและไม่สามารถดำเนินการคำสั่งในการค้นหาบางประเภทได้ โดยวิธีการนี้จะมีการแบ่งข้อมูลและจัดเก็บไว้ใน 3 ตาราง คือ ตาราง Path ตาราง PSVHash ซึ่งจัดเก็บข้อมูลสำหรับอ้างอิงไปยังข้อมูลจริง และตาราง PlHash ซึ่งจัดเก็บข้อมูลจริง ในการค้นหาข้อมูลด้วยวิธีการนี้จะมีการส่งคำสั่งในการค้นหา 2 ครั้งต่อ 1 ผลลัพธ์

4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในการทดลองนี้คือชุดข้อมูลที่สร้างขึ้นจาก XMark ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างเอกสาร XML สำหรับการทดลองได้ถูกนำมาใช้ในงานวิจัยที่เกี่ยวข้องกับการเข้ารหัสข้อมูลในเอกสาร XML (Lee, 2006), (Wang, 2006)

4.1.1 โครงสร้างของเอกสาร

โครงสร้างของเอกสาร XML ที่ถูกสร้างขึ้นโดย XMark จะมีลักษณะดังภาพประกอบ 4.1



ภาพประกอบ 4.1 ตัวอย่างโครงสร้างเอกสาร XML จาก XMark

4.1.2 การกำหนดค่า Factor

โปรแกรมที่ใช้ในการสร้างเอกสาร XML สามารถกำหนดค่า Factor ซึ่งเป็นค่าที่ระบุขนาดของเอกสาร XML โดยที่แต่ละ Factor จะให้ขนาดของเอกสารแตกต่างกันดังตารางที่ 4.1

ตารางที่ 4.1 ความสัมพันธ์ระหว่างค่า Factor และขนาดของเอกสาร

Factor	ขนาดของเอกสาร
0.01	1 MB
0.1	10 MB
1	100 MB
10	1 GB

4.2 การออกแบบการทดลอง

4.2.1 การทดลองเปรียบเทียบการจัดเก็บข้อมูล

ในการจัดเก็บข้อมูลจะจัดเก็บข้อมูลตามแต่ละขั้นตอนวิธีและมีการเข้ารหัสข้อมูลในส่วนของอิลิเมนต์ creditcard และ education โดยที่ในวิธีการของ XML Encryption จะมีการจัดเก็บข้อมูลอยู่ในรูปแบบของ Text File แต่ในส่วนของวิธีการ SemCrypt และวิธีการของงานวิจัยจะใช้การจัดเก็บข้อมูลลงในฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

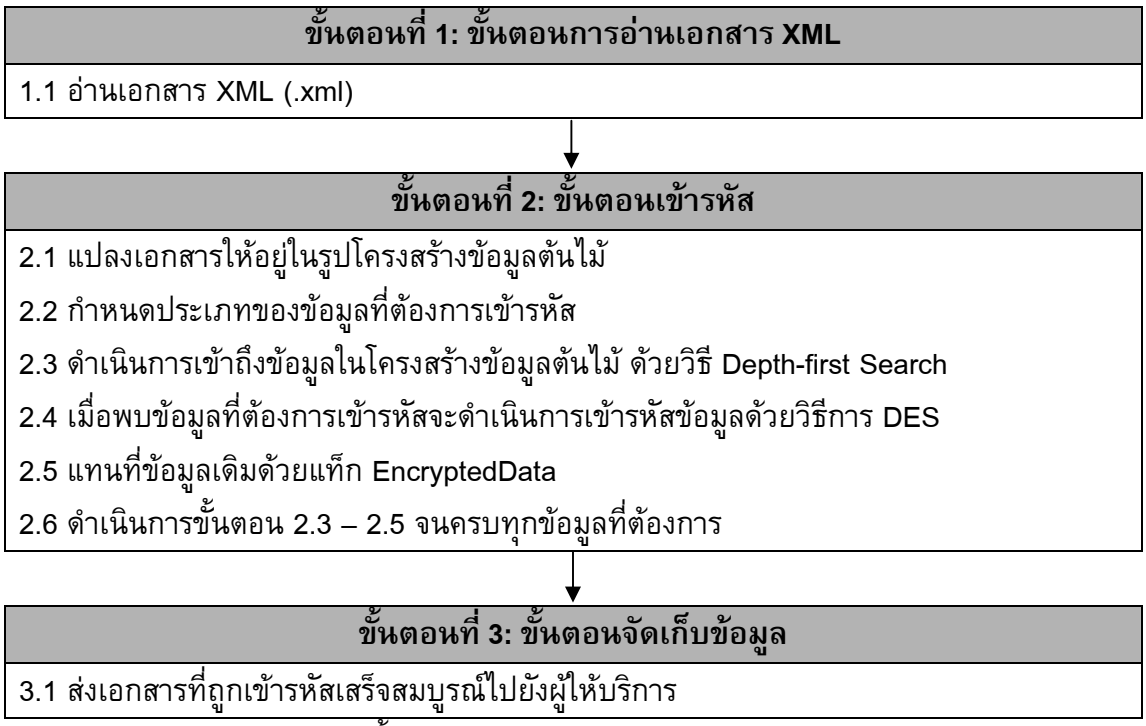
ในส่วนของการเข้ารหัสข้อมูลจะใช้วิธีการเข้ารหัสข้อมูลที่เหมือนกันนั่นคือ DES (Data Encryption Standard) โดยที่ลักษณะทั่วไปของแต่ละวิธีอธิบายดังตารางที่ 4.2

ตารางที่ 4.2 ลักษณะทั่วไปของแต่ละวิธีการจัดเก็บข้อมูล

วิธีการ	XML Encryption	SemCrypt	qDAS
การจัดเก็บข้อมูลในฝั่งผู้รับบริการ	ไม่มีการจัดเก็บ	มีการจัดเก็บข้อมูลพาธทั้งหมด	มีการจัดเก็บข้อมูลพาธ ที่มีการเข้ารหัส
การจัดเก็บข้อมูลในฝั่งผู้ให้บริการ	มีการจัดเก็บข้อมูลทั้งหมด	มีการจัดเก็บข้อมูลทั้งหมด	มีการจัดเก็บข้อมูลทั้งหมด
รูปแบบการจัดเก็บข้อมูล	Text File	ฐานข้อมูลเชิงสัมพันธ์	ฐานข้อมูลเชิงสัมพันธ์
การสร้าง XML Tree Model	ใช้ DOM ในการสร้าง	ใช้ DOM ในการสร้าง	ใช้ DOM ในการสร้าง
ฟังก์ชันเข้ารหัส	DES	DES	DES
ฟังก์ชัน Hash	ไม่ใช้	Shift-add-XOR	Shift-add-XOR
เอกสารนิยามโครงสร้าง	ไม่ใช่	XML Schema	XML Schema
กุญแจในการเข้ารหัส	ขึ้นอยู่จำนวนข้อมูลที่ซ้ำกันในเอกสาร	อย่างน้อย 1 กุญแจ	อย่างน้อย 1 กุญแจ
จำนวนตารางข้อมูล	ไม่มีการจัดเก็บข้อมูลในตารางข้อมูล	3 ตาราง	2 ตาราง

1) การจัดเก็บข้อมูลด้วยวิธีการ XML Encryption

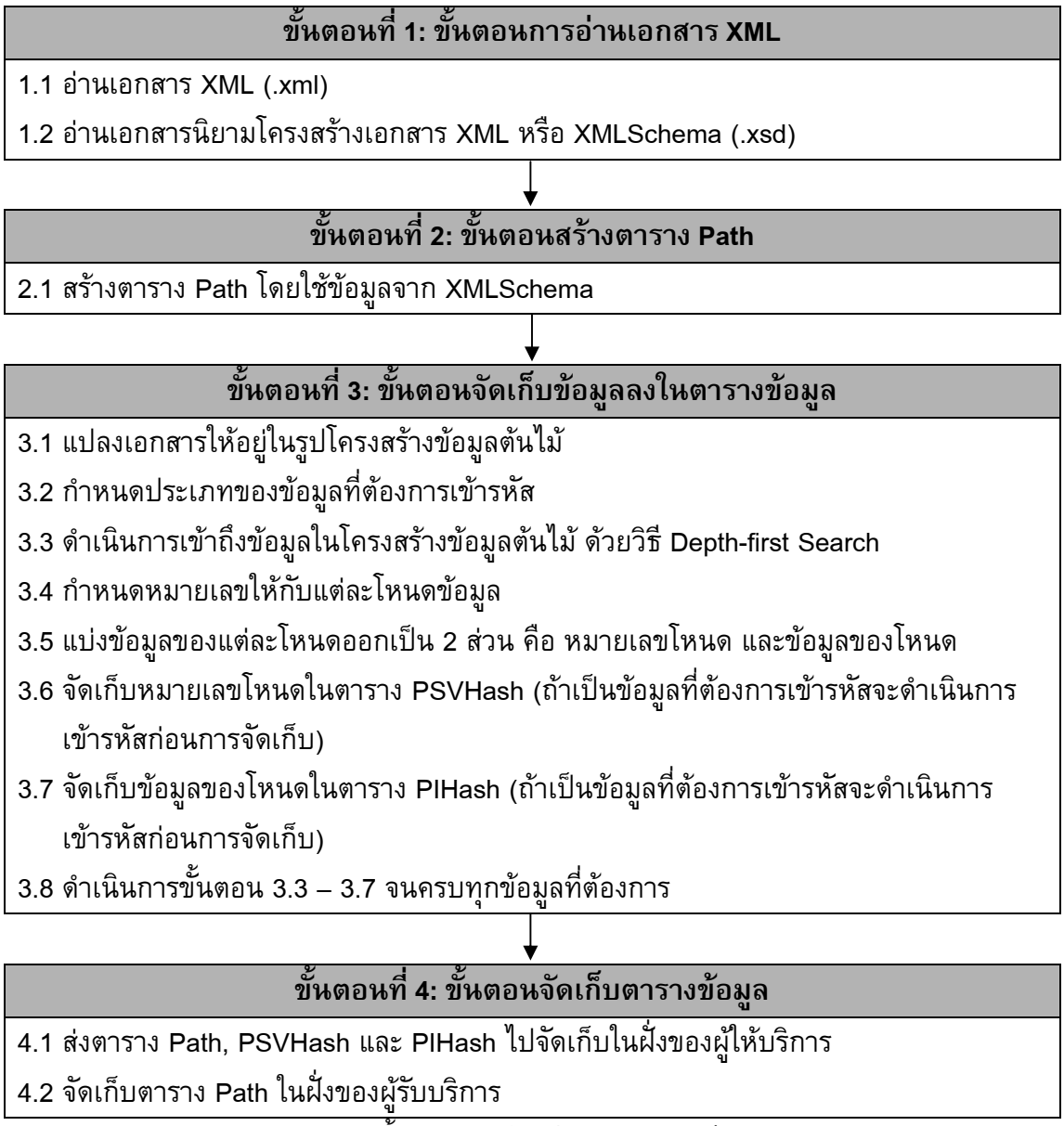
ในขั้นตอนวิธีการจัดเก็บข้อมูลของวิธีการ XML Encryption จะมีการอ่านข้อมูลในเอกสารที่ต้องการเข้ารหัส หลังจากนั้นผู้รับบริการกำหนดประเภทของข้อมูลที่ต้องการเข้ารหัสซึ่งสามารถกำหนดได้มากกว่า 1 ตำแหน่ง การอ่านข้อมูลในเอกสารจะใช้ DOM ในการสร้าง XML Tree Model เมื่อดำเนินการเข้าถึงข้อมูล Depth-first Search และพบประเภทของข้อมูลที่ตรงกับที่ต้องการเข้ารหัสก็จะดำเนินการสร้างข้อมูลใหม่ซึ่งเป็นข้อมูลที่ถูกเข้ารหัสและนำไปแทนที่ข้อมูลเดิม ทำเช่นนี้ไปเรื่อยจนครบทุกข้อมูลตามที่ต้องการ เมื่อเข้ารหัสข้อมูลเสร็จสิ้นผู้รับบริการจะส่งเอกสารที่ได้เข้ารหัสแล้วไปจัดเก็บในฝั่งของผู้ให้บริการ ซึ่งวิธีการจัดเก็บข้อมูลของ XML Encryption แสดงดังภาพประกอบ 4.2 การจับเวลาในการจัดเก็บข้อมูลจะเริ่มต้นตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้าย



ภาพประกอบ 4.2 ขั้นตอนการจัดเก็บข้อมูลด้วยวิธีการ XML Encryption

2) การจัดเก็บข้อมูลด้วยวิธีการ SemCrypt

ในขั้นตอนวิธีการจัดเก็บข้อมูลของวิธีการ SemCrypt เริ่มต้นจะมีการอ่านเอกสาร XML (.xml) และอ่านเอกสารนิยามโครงสร้างเอกสาร XML (.xsd) แล้วจากนั้นจะสร้างตารางสำหรับเก็บข้อมูลของแต่ละพาธที่มีข้อมูล โดยใช้ XMLSchema หลังจากนั้นจะแปลงเอกสาร XML ให้เป็นโครงสร้างข้อมูลต้นไม้ (XML Tree Model) เมื่อได้โครงสร้างต้นไม้จะดำเนินการจัดเก็บข้อมูลลงในตารางข้อมูล โดยข้อมูลแบ่งออกเป็น 2 ประเภทคือ ข้อมูลที่ใช้สำหรับอ้างอิง และข้อมูลจริง ในกรณีของข้อมูลที่ต้องการเข้ารหัส ข้อมูลที่ใช้สำหรับอ้างอิงของข้อมูลนั้นจะมีการเข้ารหัสเช่นเดียวกัน นอกจากนี้ในกรณีที่ข้อมูลที่มีค่าเหมือนกัน ข้อมูลที่ใช้สำหรับอ้างอิงของข้อมูลนั้นจะถูกจัดเก็บอยู่ในกลุ่มเดียวกัน เมื่อดำเนินการเข้ารหัสข้อมูลเสร็จสิ้นก็จะส่งตารางข้อมูลไปจัดเก็บในฝั่งของผู้ให้บริการ ซึ่งวิธีการจัดเก็บข้อมูลของ SemCrypt แสดงดังภาพประกอบ 4.3 การจับเวลาในการจัดเก็บข้อมูลจะเริ่มต้นตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้าย



ภาพประกอบ 4.3 ขั้นตอนการจัดเก็บข้อมูลด้วยวิธีการ SemCrypt

4.2.2 การทดลองเปรียบเทียบการค้นหาข้อมูล

ในการทดลองเปรียบเทียบประสิทธิภาพในการค้นหาข้อมูลบนเอกสาร XML ที่ถูกเข้ารหัสของวิธีที่ได้นำเสนอไว้ในงานวิจัยนี้กับวิธีการของงานวิจัยก่อนหน้าโดยแต่ละวิธีการมีลักษณะการทำงานที่ต่างกันแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ลักษณะทั่วไปของแต่ละวิธีการในการค้นหาข้อมูล

วิธีการ	XML Encryption	SemCrypt	qDAS
การถอดรหัสข้อมูล	ถอดรหัสทั้งเอกสารก่อนการค้นหา	ถอดรหัสเฉพาะข้อมูลที่สนใจ	ถอดรหัสเฉพาะข้อมูลที่สนใจ
การติดต่อผู้ให้บริการ	1 ครั้ง	ขึ้นอยู่จำนวนผลลัพธ์	1 ครั้ง
คำสั่งในการค้นหา	XPath	แปลง XPath เป็น SQL	แปลง XPath เป็น SQL

1) การค้นหาข้อมูลด้วยวิธีการ XML Encryption

ในการค้นหาข้อมูลด้วยวิธีการ XML Encryption เริ่มต้นจะพิจารณาพาธที่รับเข้ามาว่าเป็นพาธที่มีการเข้ารหัสหรือไม่ ถ้าเป็นพาธที่ไม่เข้ารหัสก็จะดำเนินการค้นหาโดยใช้ XPath และแสดงผลลัพธ์ของคำสั่งในการค้นหา แต่ถ้าหากเป็นกรณีของพาธที่เข้ารหัสก็จะต้องดำเนินการถอดรหัสข้อมูลทั้งหมดก่อนการค้นหา เมื่อถอดรหัสเสร็จสิ้นจึงใช้ XPath ในการค้นหาข้อมูลที่ต้องการ ซึ่งวิธีการค้นหาแสดงดังภาพประกอบ 4.4

ขั้นตอนที่ 1: ขั้นตอนการกำหนดคำสั่งในการดึงเอกสาร

1.1 ผู้รับบริการกำหนดคำสั่งในการดึงเอกสาร
1.2 ผู้ให้บริการส่งเอกสารทั้งหมดกลับไปให้ผู้รับบริการ

↓

ขั้นตอนที่ 2: ขั้นตอนการพิจารณาประเภทของคำสั่งในการค้นหา

2.1 ผู้รับบริการพิจารณาประเภทของคำสั่งในการค้นหา
2.2 ถ้าเป็นคำสั่งซึ่งประกอบด้วยพาธที่ไม่เข้ารหัส
    2.2.1 ค้นหาข้อมูลที่ต้องการโดยใช้ XPath
2.3 ถ้าเป็นคำสั่งซึ่งประกอบด้วยพาธที่เข้ารหัส
    2.3.1 ดำเนินการถอดรหัสข้อมูลทั้งหมดที่ถูกเข้ารหัสในเอกสาร
    2.3.2 ค้นหาข้อมูลที่ต้องการโดยใช้ XPath

ภาพประกอบ 4.4 ขั้นตอนการค้นหาข้อมูลด้วยวิธีการ XML Encryption

2) การค้นหาข้อมูลด้วยวิธีการ SemCrypt

ในการค้นหาข้อมูลด้วยวิธีการ SemCrypt เริ่มต้นจากผู้รับบริการกำหนดคำสั่งในการค้นหา หลังจากนั้นเชื่อมต่อไปยังผู้ให้บริการ และส่งพาธที่ต้องการไปยังผู้ให้บริการเพื่อ

ร้องขอข้อมูล PathSchemaId (ข้อมูลสำหรับอ้างอิง) เมื่อได้ข้อมูล PathSchemaId หลังจากนั้นจะร้องขอ PathInstance (ข้อมูล) ที่ตรงกับค่าเงื่อนไข (ในกรณีที่มีเงื่อนไขในการค้นหา) ถัดจากนั้นร้องขอค่าข้อมูลที่ตรงกับ PathInstance ซึ่งในกรณีที่เป็นข้อมูลที่ถูกเข้ารหัสก็จะมีการถอดรหัสก่อนการแสดงผลลัพธ์ แต่ถ้าผลลัพธ์ของการค้นหามีมากกว่า 1 ผลลัพธ์จะทำให้ได้ PathInstance เท่ากับจำนวนผลลัพธ์ และเมื่อต้องการผลลัพธ์ทั้งหมดก็จำเป็นที่จะต้องส่ง PathInstance ทั้งหมด เมื่อร้องขอข้อมูลเสร็จสิ้นก็จะปิดการเชื่อมต่อกับฐานข้อมูล ซึ่งวิธีการค้นหาแสดงดังภาพประกอบ 4.5

ขั้นตอนที่ 1: ขั้นตอนการกำหนดคำสั่งในการค้นหา

1.1 ผู้รับบริการกำหนดคำสั่งในการค้นหา
1.2 เปิดการเชื่อมต่อกับฐานข้อมูล
1.3 ผู้รับบริการร้องขอ PathSchemaId จากผู้ให้บริการ

↓

ขั้นตอนที่ 2: ขั้นตอนการพิจารณาประเภทของคำสั่งในการค้นหา

2.1 ผู้รับบริการพิจารณาประเภทของคำสั่งในการค้นหา
2.2 ถ้าเป็นคำสั่งที่ไม่มีเงื่อนไข
    2.2.1 ผู้รับบริการส่งข้อมูล PathSchemaId
    2.2.2 ผู้ให้บริการส่งผลลัพธ์กลับ
2.3 ถ้าเป็นคำสั่งที่มีเงื่อนไข
    2.3.1 ผู้รับบริการส่งข้อมูล PathSchemaId รวมกับเงื่อนไข
    2.3.2 ผู้ให้บริการส่ง PathInstance
    2.3.3 ผู้รับบริการส่งข้อมูล PathInstance ทั้งหมด
    2.3.4 ผู้ให้บริการส่งผลลัพธ์ทั้งหมดกลับ

ภาพประกอบ 4.5 ขั้นตอนการค้นหาข้อมูลด้วยวิธีการ SemCrypt

4.3 ผลการทดลอง

ผลการทดลองในวิทยานิพนธ์นี้จะแบ่งออกเป็น 3 หัวข้อได้แก่ การทดลองเปรียบเทียบประสิทธิภาพในการจัดเก็บข้อมูล การทดลองเปรียบเทียบประสิทธิภาพในการค้นหาข้อมูล และการเปรียบเทียบความปลอดภัย

4.3.1 ผลการทดลองเปรียบเทียบประสิทธิภาพในการจัดเก็บข้อมูล

ในการทดลองเปรียบเทียบประสิทธิภาพในการจัดเก็บข้อมูลจะแบ่งออกเป็น 4 ประเด็น คือ เวลาที่ใช้ในการจัดเก็บข้อมูล พื้นที่ในการจัดเก็บข้อมูล จำนวนกุญแจที่ใช้ในการเข้ารหัสข้อมูล และกรณีที่ขนาดของเอกสารเพิ่มขึ้น

โดยที่ในการทดลองนี้จะใช้เอกสาร XML ที่ถูกสร้างขึ้นโดย XMark ซึ่งมีขนาด 1 MB (Factor = 0.01) สาหรับประเด็น 1 ถึง 3

1) ประเด็นเวลาที่ใช้ในการจัดเก็บข้อมูล

ตารางที่ 4.4 เปรียบเทียบเวลาที่ใช้ในการจัดเก็บข้อมูล

วิธีการ	เวลา (วินาที)
XML Encryption	1.594
SemCrypt	255.58
qDAS	126.14

จากตารางที่ 4.4 พบว่าเวลาที่ใช้ในการเข้ารหัสข้อมูลของแต่ละวิธีการมีความแตกต่างกัน ซึ่งวิธีการ XML Encryption ใช้เวลาน้อยที่สุด ตามมาด้วย qDAS และ SemCrypt ตามลำดับ เนื่องจากวิธีการ XML Encryption มีการเข้ารหัสข้อมูลและจัดเก็บลงใน Text File นอกจากนี้วิธีการ XML Encryption ไม่ต้องเสียเวลาการคำนวณหาค่า Hash ซึ่งแตกต่างกับอีก 2 วิธีที่ต้องใช้เวลาส่วนใหญ่ในการติดต่อกับฐานข้อมูลเชิงสัมพันธ์และคำนวณค่า Hash

ในส่วนของวิธีการ SemCrypt และ qDAS จะมีความแตกต่างกันในเรื่องของการคำนวณค่า Hash เนื่องจาก SemCrypt จะคำนวณค่า Hash 2 ครั้งสำหรับข้อมูลที่ต้องการเข้ารหัสและ 1 ครั้งสำหรับข้อมูลที่ไม่ต้องการเข้ารหัส แต่ qDAS จะมีการเข้ารหัส 2 ครั้งสำหรับข้อมูลที่ต้องการเข้ารหัสและไม่มีการคำนวณค่า Hash สำหรับข้อมูลที่ไม่ต้องการเข้ารหัส นอกจากนี้วิธีการ SemCrypt จะมีการจัดเก็บข้อมูลลงใน 2 ตาราง ซึ่งจะต้องติดต่อกับฐานข้อมูล 2 ครั้งต่อการจัดเก็บ 1 ระเบียน แต่ qDAS จะมีการจัดเก็บข้อมูลลงในตารางเดียว (การดำเนินในส่วนนี้จะไม่นับการจัดเก็บข้อมูลในตาราง Path เนื่องจากเป็นการจัดเก็บที่แยกส่วนการทำงาน)

2) ประเด็นพื้นที่ในการจัดเก็บข้อมูล

ตารางที่ 4.5 เปรียบเทียบพื้นที่ในการจัดเก็บข้อมูล

วิธีการ	ขนาด (MB)	เพิ่มขึ้น %
Original File	1.12	
XML Encryption	1.25	12 %
SemCrypt	1.8	60 %
qDAS	1.4	25 %

จากตารางที่ 4.5 พบว่าพื้นที่ในการจัดเก็บข้อมูลโดยใช้วิธีการ XML Encryption ใช้พื้นที่น้อยที่สุด ตามมาด้วย qDAS และ SemCrypt ตามลำดับ เนื่องจากวิธีการ XML Encryption ไม่มีการจัดเก็บข้อมูลที่ช่วยในการค้นหา ซึ่งแตกต่างกับอีก 2 วิธีที่มีการคำนวณค่า Hash เพื่อใช้ช่วยในการค้นหาข้อมูล โดยที่ SemCrypt จะจัดเก็บข้อมูล Hash 2 ค่าต่อ 1 ข้อมูล แต่ qDAS จะจัดเก็บข้อมูล Hash 1 ค่าต่อ 1 ข้อมูล

3) ประเด็นจำนวนกุญแจที่ใช้ในการเข้ารหัสข้อมูล

ตารางที่ 4.6 เปรียบเทียบจำนวนกุญแจที่ใช้ในการเข้ารหัสข้อมูลในกรณีที่ข้อมูลแตกต่างกัน

วิธีการ	จำนวนกุญแจ
XML Encryption	137
SemCrypt	1
qDAS	1

จากตารางที่ 4.6 พบว่าจำนวนกุญแจที่ใช้ในการเข้ารหัสข้อมูลเพื่อให้ผลลัพธ์ของการเข้ารหัส (Cipher Text) มีความแตกต่างกันด้วยวิธีการ SemCrypt และ qDAS มีค่าเท่ากับ 1 กุญแจซึ่งน้อยกว่าวิธีการ XML Encryption ซึ่งใช้กุญแจอย่างน้อย 137 กุญแจในการที่จะทำให้ได้ผลลัพธ์ที่เหมือนกันเมื่อผ่านการเข้ารหัสแล้วได้ Cipher text ที่แตกต่างกัน

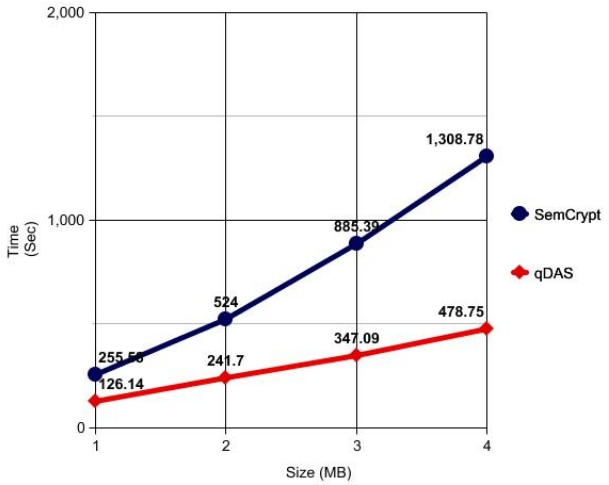
วิธีการ SemCrypt และ qDAS จะมีลักษณะที่เหมือนกันคือจะนำข้อมูลที่ต้องการเข้ารหัสไปร่วมกับค่าเฉพาะค่าหนึ่งแล้วจึงดำเนินการเข้ารหัส ซึ่งวิธีการทั้งสองจะได้ Cipher text แตกต่างกันโดยใช้กุญแจเพียงกุญแจเดียว

4) ประเด็นเมื่อขนาดของเอกสารเพิ่มขึ้น

ในการทดลองนี้เปรียบเทียบเฉพาะวิธี SemCrypt และ qDAS เท่านั้น เนื่องจากผลการทดลองก่อนหน้านี้แสดงให้เห็นว่าวิธีการ XML Encryption ใช้เวลาน้อยมากซึ่งแตกต่างกันหลายเท่าตัว

ตารางที่ 4.7 เปรียบเทียบผลการทดลองเมื่อเอกสารมีขนาดที่แตกต่างกัน

ขนาด / วิธีการ	1 MB	2 MB	3 MB	4 MB
SemCrypt	255.58 Sec	524.0 Sec	885.39 Sec	1308.78 Sec
qDAS	126.14 Sec	241.70 Sec	347.09 Sec	478.75 Sec



ภาพประกอบ 4.6 การเปรียบเทียบเวลาที่ใช้ในการจัดเก็บข้อมูลเมื่อเอกสารมีขนาดเพิ่มขึ้น

จากตารางที่ 4.7 และภาพประกอบ 4.6 พบว่าเมื่อเอกสารมีขนาดใหญ่ขึ้น วิธี qDAS และ SemCrypt ใช้เวลาในการดำเนินการเพื่อจัดเก็บข้อมูลเพิ่มขึ้นตามขนาดของเอกสาร โดยที่แนวโน้มของการเพิ่มขึ้นของวิธีการ qDAS มีลักษณะเป็นแบบเชิงเส้น (Linear) แต่วิธีการ SemCrypt มีแนวโน้มในลักษณะเป็นโพลีโนเมียล (Polynomial) เนื่องจากในวิธีการ SemCrypt เมื่อข้อมูลที่ซ้ำกันเพิ่มจำนวนขึ้นจะมีการเพิ่มข้อมูลสำหรับการค้นหาต่อท้ายข้อมูลเดิมและดำเนินการเข้ารหัสซ้ำอีกครั้ง

4.3.2 ผลการทดลองเปรียบเทียบประสิทธิภาพในการค้นหาข้อมูล

ในการทดลองเปรียบเทียบการค้นหาข้อมูล โดยที่จะดำเนินการครั้งละ 1 คำสั่ง ซึ่งแบ่งออกเป็น 7 กรณีได้แก่

1) กรณีการค้นหาข้อมูลแบบ Unconditional query with unencrypted result

ตารางที่ 4.8 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Unconditional query with unencrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	0.234
SemCrypt	0.03
qDAS	0.015

จากตารางที่ 4.8 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Unconditional query with unencrypted result โดยใช้ /site/people/person/name เป็นพาธในการค้นหาพบว่าวิธีการ qDAS ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ XML Encryption จะค้นหาข้อมูลในเอกสาร XML โดยใช้การสำรวจตลอดเอกสารและเมื่อพบข้อมูลที่ต้องการก็จะนำมาแสดง ในทางตรงข้ามวิธีการ SemCrypt และ qDAS จะใช้การค้นหาข้อมูลในฐานข้อมูลโดยใช้ภาษา SQL เวลาในการค้นหาของ SemCrypt จะมีลักษณะที่คล้ายกับวิธีการ qDAS แต่จะแตกต่างกันตรงที่ SemCrypt จะมีค่า CryptoHashPI ซึ่งเป็นค่าที่ช่วยค้นหาที่มีความยาวมากกว่าทำให้เสียเวลาในส่วนนี้เพิ่มขึ้นจากวิธีการ qDAS

2) กรณีการค้นหาข้อมูลแบบ Unconditional query with encrypted result

ตารางที่ 4.9 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Unconditional query with encrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	9.3
SemCrypt	ไม่สามารถดำเนินการได้
qDAS	0.156

จากตารางที่ 4.9 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Unconditional query with encrypted result โดยใช้ /site/people/person/creditcard เป็นพาธในการค้นหาพบว่าวิธีการ qDAS ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ XML Encryption จะต้องทำการ

ถอดรหัสข้อมูลในส่วนของ creditcard ทั้งหมดก่อนการค้นหาข้อมูลในเอกสาร XML ซึ่งเวลา 9.3 วินาทีใช้ไปสำหรับการถอดรหัสข้อมูล

ในกรณีวิธีการ SemCrypt ไม่สามารถดำเนินการค้นหาข้อมูลในประเภทนี้ได้ เนื่องจากไม่สามารถคำนวณเพื่อหาค่าข้อมูลที่จะนำมาช่วยค้นหา (CrytoHashPI) ซึ่งได้จากฟังก์ชัน Hash ของหมายเลขพาธรวมกับหมายเลขของแต่ละโหนด โดยที่ในที่นี้ไม่สามารถค้นหาหมายเลขของแต่ละโหนดได้

3) กรณีการค้นหาข้อมูลแบบ Unencrypted conditional query with unencrypted result

ตารางที่ 4.10 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Unencrypted conditional query with unencrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	0.219
SemCrypt	0.405
qDAS	0.031

จากตารางที่ 4.10 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Unencrypted conditional query with unencrypted result โดยใช้ /site/people/person/profile [gender='male']/business เป็นพาธในการค้นหาพบว่าวิธีการ qDAS ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ qDAS ส่งคำสั่งในการค้นหาไปยังฐานข้อมูลเพียงคำสั่งเดียว ในขณะที่ SemCrypt จะต้องส่งคำสั่งในการค้นหาเท่ากับจำนวนบุคคลที่เป็นเพศชายซึ่งในข้อมูลมีอยู่หลายบุคคล

4) กรณีการค้นหาข้อมูลแบบ Unencrypted conditional query with encrypted result

ตารางที่ 4.11 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Unencrypted conditional query with encrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	9.3
SemCrypt	0.392
qDAS	0.141

จากตารางที่ 4.11 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Unencrypted conditional query with encrypted result โดยใช้ /site/people/person/profile [gender='male']/education เป็นพาธในการค้นหาพบว่าวิธีการ qDAS ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ qDAS ส่งคำสั่งในการค้นหาไปยังฐานข้อมูลเพียงคำสั่งเดียว ในขณะที่ SemCrypt จะต้องส่งคำสั่งในการค้นหาเท่ากับจำนวนบุคคลที่เป็นเพศชายซึ่งในข้อมูลมีอยู่หลายบุคคล ในส่วนของวิธีการ XML Encryption จะต้องทำการถอดรหัสข้อมูลในส่วนของ education ทั้งหมดก่อนการค้นหาข้อมูลในเอกสาร XML

5) กรณีการค้นหาข้อมูลแบบ Encrypted conditional query with unencrypted result

ตารางที่ 4.12 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Encrypted conditional query with unencrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	9.3
SemCrypt	0.001
qDAS	0.015

จากตารางที่ 4.12 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Encrypted conditional query with unencrypted result โดยใช้ /site/people/person[creditcard='5821 1584 3395 4744']/name เป็นพาธในการค้นหาพบว่าวิธีการ SemCrypt ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ SemCrypt ลดเวลาในการเชื่อมข้อมูลในตารางฐานข้อมูล (Joint Table) แต่ในส่วนวิธี qDAS จะมีการเชื่อมข้อมูลในตารางข้อมูลที่มากกว่าทำให้เวลาในการค้นหามากกว่า

6) กรณีการค้นหาข้อมูลแบบ Encrypted conditional query with encrypted result

ตารางที่ 4.13 เปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลแบบ Encrypted conditional query with encrypted result

วิธีการ	เวลา (วินาที)
XML Encryption	9.3
SemCrypt	0.015
qDAS	0.142

จากตารางที่ 4.13 ซึ่งเป็นการทดลองค้นหาข้อมูลแบบ Encrypted conditional query with encrypted result โดยใช้ /site/people/person[creditcard='1681 7357 9954 2844']/profile/education เป็นพาธในการค้นหาพบว่าวิธีการ SemCrypt ใช้เวลาในการค้นหาน้อยที่สุด เนื่องจากวิธีการ SemCrypt ลดเวลาในการเชื่อมข้อมูลในตารางฐานข้อมูล (Joint Table) แต่ในส่วนวิธี qDAS จะมีการเชื่อมข้อมูลในตารางข้อมูลที่มากกว่าทำให้เวลาในการค้นหามากกว่า นอกจากนี้ยังพบความแตกต่างของเวลาระหว่าง Q5 และ Q6 เนื่องจากในส่วนของ Q6 จะเพิ่มเวลาในส่วนของการถอดรหัส

7) ประเด็นเวลาในการค้นหาข้อมูลเมื่อเอกสารมีขนาดเพิ่มขึ้น

ในการทดลองนี้เปรียบเทียบเฉพาะวิธี SemCrypt และ qDAS เท่านั้น เนื่องจากผลการทดลองก่อนหน้านี้แสดงให้เห็นว่าวิธีการ XML Encryption ใช้เวลามากกว่าหลายเท่าตัวและเสียเวลาในการถอดรหัสข้อมูล

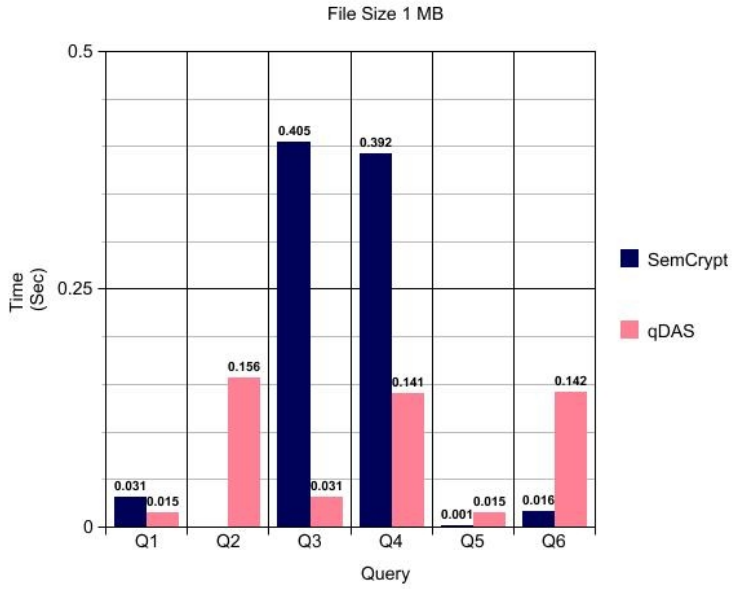
ตารางที่ 4.14 เปรียบเทียบการค้นหาด้วยวิธี SemCrypt เมื่อเอกสารมีขนาดที่ต่างกัน

Query / Size	Q1	Q2	Q3	Q4	Q5	Q6
1 MB	0.031 Sec	-	0.405 Sec	0.392 Sec	0.001 Sec	0.016 Sec
2 MB	0.047 Sec	-	0.813 Sec	0.797 Sec	0.001 Sec	0.016 Sec
3 MB	0.063 Sec	-	1.546 Sec	1.516 Sec	0.001 Sec	0.016 Sec
4 MB	0.078 Sec	-	3.188 Sec	3.125 Sec	0.001 Sec	0.016 Sec

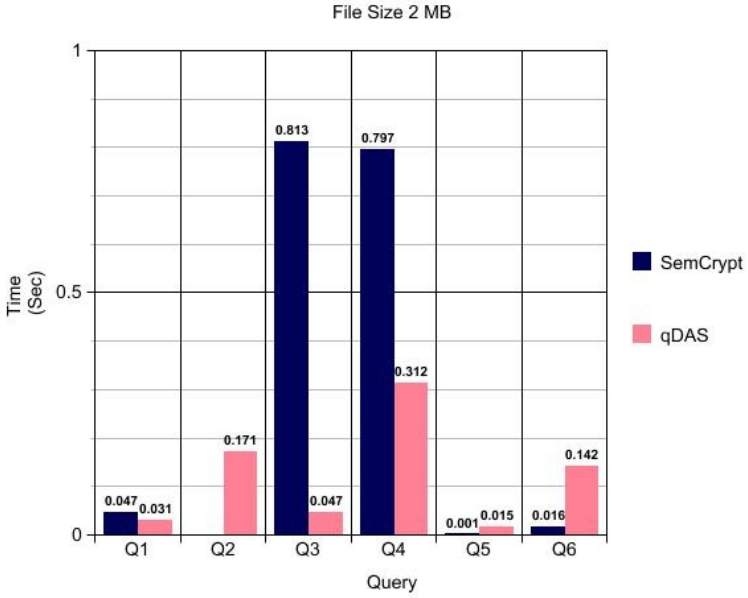
ตารางที่ 4.15 เปรียบเทียบการค้นหาด้วยวิธี qDAS เมื่อเอกสารมีขนาดที่ต่างกัน

Query / Size	Q1	Q2	Q3	Q4	Q5	Q6
1 MB	0.015 Sec	0.156 Sec	0.031 Sec	0.141 Sec	0.015 Sec	0.142 Sec
2 MB	0.031 Sec	0.171 Sec	0.047 Sec	0.312 Sec	0.015 Sec	0.142 Sec
3 MB	0.047 Sec	0.188 Sec	0.093 Sec	0.634 Sec	0.015 Sec	0.142 Sec
4 MB	0.062 Sec	0.219 Sec	0.156 Sec	0.945 Sec	0.015 Sec	0.142 Sec

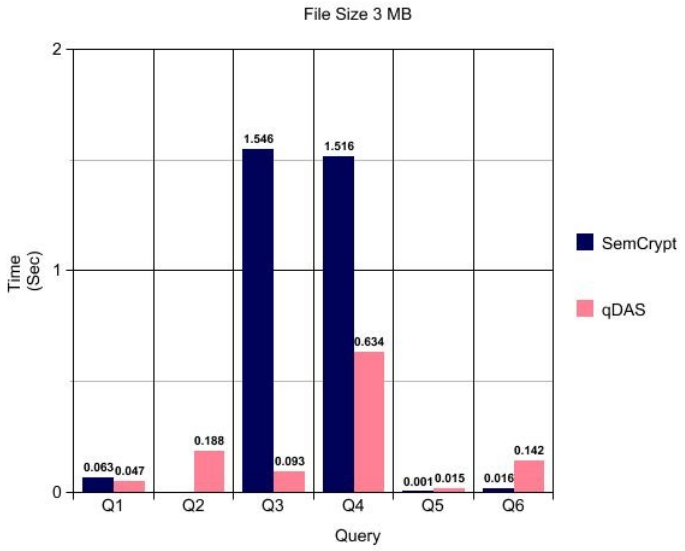
จากตารางที่ 4.14 และ 4.15 พบว่าเมื่อเอกสารมีขนาดใหญ่ขึ้น จะทำให้ข้อมูลที่ต้องการค้นหาผลลัพธ์จะเพิ่มขึ้น ซึ่งส่งผลให้เวลาในการค้นหาเพิ่มขึ้นตามไปด้วย ยกเว้นในส่วนของ Q5 และ Q6 ที่ใช้เวลาในการค้นหาไม่เปลี่ยนแปลงเนื่องจากผลลัพธ์ของการค้นหามีเพียงผลลัพธ์เดียว



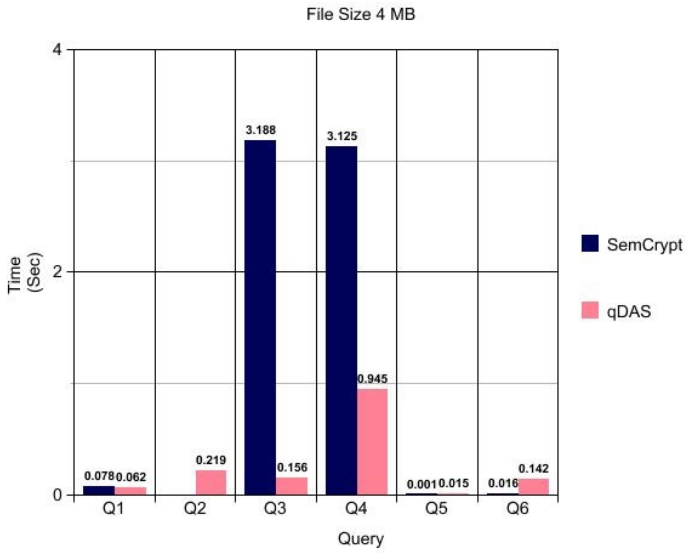
ภาพประกอบ 4.7 เปรียบเทียบเวลาในการค้นหาในรูปแบบต่างๆ กรณีที่เอกสารมีขนาด 1 MB



ภาพประกอบ 4.8 เปรียบเทียบเวลาในการค้นหาในรูปแบบต่างๆ กรณีที่เอกสารมีขนาด 2 MB



ภาพประกอบ 4.9 เปรียบเทียบเวลาในการค้นหาในรูปแบบต่างๆ กรณีที่เอกสารมีขนาด 3 MB



ภาพประกอบ 4.10 เปรียบเทียบเวลาในการค้นหาในรูปแบบต่างๆ กรณีที่เอกสารมีขนาด 4 MB

จากภาพประกอบ 4.7 – 4.10 เมื่อเปรียบเทียบระหว่างวิธีการ qDAS และ SemCrypt ในกรณี Q1-Q4 พบว่า วิธีการ qDAS สามารถดำเนินการค้นหาข้อมูลได้รวดเร็วกว่า แต่ในส่วนของกรณี Q5-Q6 พบว่า วิธีการ SemCrypt สามารถดำเนินการค้นหาข้อมูลได้รวดเร็วกว่า เมื่อพิจารณาถึงรูปแบบของคำสั่งในการค้นหาพบว่าในกรณี Q1-Q4 เป็นการค้นหาที่ได้ผลลัพธ์เป็นจำนวนมาก แต่กรณี Q5-Q6 เป็นการค้นหาที่ได้ผลลัพธ์เพียงผลลัพธ์เดียว ดังนั้น สามารถสรุปได้ว่า ในกรณีที่มีผลลัพธ์เป็นจำนวนมาก วิธีการ qDAS จะสามารถค้นหาข้อมูลได้รวดเร็วกว่า แต่ในทางกลับกันหากผลลัพธ์มีจำนวนน้อย วิธีการ SemCrypt จะสามารถค้นหาข้อมูลได้รวดเร็วกว่า

จากผลการทดลองในการค้นหาข้อมูลทั้งหมดข้างต้น สามารถวิเคราะห์ได้ว่าเวลาในการค้นหาข้อมูลจะขึ้นอยู่กับปัจจัยต่าง ๆดังต่อไปนี้

1. จำนวนผลลัพธ์ของคำสั่งในการค้นหา มีความสัมพันธ์กับลักษณะของข้อมูล เมื่อผลลัพธ์เพิ่มขึ้นเวลาในการค้นหาก็จะเพิ่มขึ้นตามไปด้วย ซึ่งเกิดจากการที่ต้องใช้เวลาในการค้นหาในตารางข้อมูลเพิ่มขึ้น

2. จำนวนครั้งในการติดต่อสื่อสาร เมื่อมีการติดต่อสื่อสารกับฐานข้อมูลมากขึ้นจะทำให้ใช้เวลาในการค้นหาเพิ่มขึ้น เนื่องจากในการติดต่อไปยังผู้ให้บริการฐานข้อมูลจำเป็นที่จะต้องมีการเริ่มต้นการติดต่อและจบการติดต่อทุกครั้ง

3. เวลาที่ใช้ในการถอดรหัส ในกรณีที่ข้อมูลไม่ต้องถอดรหัสจะใช้เวลาในการค้นหาน้อยกว่ากรณีที่ข้อมูลจะต้องถอดรหัส ซึ่งถ้าหากมีการถอดรหัสข้อมูลเป็นจำนวนมากก็จะส่งผลต่อเวลาในการค้นหา

4.3.3 เปรียบเทียบความปลอดภัย

ความปลอดภัยในการจัดการข้อมูลของแต่ละวิธีการที่ได้ทดลองมีความแตกต่างกัน ในส่วนนี้จะเป็นการเสนอข้อเปรียบเทียบในเรื่องความปลอดภัยของแต่ละวิธีการ โดยแสดงดังตารางที่ 4.16

1) ประเด็นเรื่องการรักษาความลับของเนื้อหา

การเข้ารหัสข้อมูลมีวัตถุประสงค์หลักคือการปกปิดความลับของข้อมูลเพื่อไม่ให้ผู้อื่นเข้าใจเนื้อหาของข้อมูล ซึ่งวัตถุประสงค์ของวิธีการ XML Encryption SemCrypt และ qDAS คือการปกปิดความลับเนื้อหาของข้อมูล โดยที่ความปลอดภัยของการรักษาความลับจะขึ้นอยู่กับฟังก์ชันที่แต่ละวิธีการเลือกใช้ในการเข้ารหัส ตัวอย่างของฟังก์ชันที่ใช้ในการเข้ารหัสข้อมูล เช่น DES AES Blowfish และ RSA เป็นต้น

2) ประเด็นเรื่องการป้องกันไม่ให้ทราบจำนวนข้อมูลที่ซ้ำกัน

การอาศัยจำนวนซ้ำของข้อมูลที่ถูกเข้ารหัสในการโจมตีการรักษาความปลอดภัย เรียกว่า Frequency-based Attack โดยทั่วไปแล้วการป้องกันการโจมตีรูปแบบนี้ที่ง่ายที่สุดคือ การเข้ารหัสข้อมูลโดยใช้กุญแจที่แตกต่างกันในการเข้ารหัส ซึ่งสามารถป้องกันปัญหานี้ได้ดีที่สุด แต่ประสิทธิภาพในการค้นหาจะลดลง นอกจากนี้ยังเสียเวลาในการถอดรหัสข้อมูลที่ไม่จำเป็น ซึ่งวิธีการ XML Encryption ไม่ได้ออกแบบมาเพื่อการป้องกันปัญหานี้จึงจำเป็นที่จะต้องใช้วิธีที่ได้กล่าวมาข้างต้นในการป้องกันปัญหา แต่วิธีการ SemCrypt และ qDAS ป้องกันปัญหานี้โดยการเพิ่มข้อมูลที่เป็นข้อมูลเฉพาะเข้าไปรวมกับข้อมูลเดิมก่อนการดำเนินการเข้ารหัสจึงทำให้ผลลัพธ์ของการเข้ารหัส (Cipher Text) มีความแตกต่างกัน แต่จะมีการเพิ่มพื้นที่ในการจัดเก็บข้อมูลที่ใช้สำหรับช่วยในการค้นหา

ตารางที่ 4.16 เปรียบเทียบความปลอดภัยในการจัดการข้อมูลและการค้นหา

ประเด็น	XML Encryption	SemCrypt	qDAS
การรักษาความลับของเนื้อหา	สามารถป้องกันปัญหา	สามารถป้องกันปัญหา	สามารถป้องกันปัญหา
การป้องกันไม่ให้ทราบจำนวนข้อมูลที่ซ้ำกัน	สามารถป้องกัน แต่ต้องใช้กุญแจหลายกุญแจในการเข้ารหัสข้อมูล	สามารถป้องกันปัญหา	สามารถป้องกันปัญหา
การพิสูจน์ตัวตนของผู้รับบริการ	ใช้การตรวจสอบสิทธิ์ในการเข้าถึงข้อมูลไฟล์	ใช้การตรวจสอบสิทธิ์ในการเข้าถึงฐานข้อมูล	ใช้การตรวจสอบสิทธิ์ในการเข้าถึงฐานข้อมูล
ความคงสภาพของข้อมูล	ใช้งานร่วมกับ XML Signature (W3C, 2002)	ใช้งานร่วมกับ Message Authentication	ใช้งานร่วมกับ Message Authentication
การรับส่งข้อมูลระหว่างผู้รับบริการและผู้ให้บริการ	ใช้ Protocol ในระบบชั้นเครือข่าย	ใช้ Protocol ในระบบชั้นเครือข่าย	ใช้ Protocol ในระบบชั้นเครือข่าย

3) ประเด็นเรื่องการพิสูจน์ตัวตนของผู้รับบริการ

เนื่องจากมีการติดต่อสื่อสารระหว่างผู้รับบริการและผู้ให้บริการในการร้องขอข้อมูลจึงทำให้เป็นช่องทางในการโจมตีของผู้ประสงค์ร้าย ดังนั้นจึงจำเป็นที่จะต้องมีการป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต ซึ่งวิธีการ XML Encryption สามารถป้องกันได้ 2 รูปแบบ คือ การตรวจสอบสิทธิ์ในการเข้าถึงไฟล์ และการตรวจสอบสิทธิ์ในการเข้าถึงเอกสาร XML (Zhu และคณะ, 2009) โดยที่ทั้งสองวิธีการนี้ขึ้นอยู่กับความเหมาะสมของการใช้งาน ในส่วนของวิธีการ SemCrypt และ qDAS จะใช้วิธีการป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตเช่นเดียวกับระบบฐานข้อมูลเชิงสัมพันธ์โดยทั่วไป เนื่องจากการจัดเก็บข้อมูลของวิธีการทั้งสองนี้อาศัยการทำงานของระบบฐานข้อมูลเชิงสัมพันธ์

4) ประเด็นเรื่องความคงสภาพของข้อมูล

ในการติดต่อสื่อสารแลกเปลี่ยนข้อมูลที่เกิดขึ้นในโลกอินเตอร์เน็ตเป็นการแลกเปลี่ยนข้อมูลที่เป็นดิจิตอล ซึ่งแตกต่างกับเอกสารกระดาษที่เป็นจดหมายหรือหนังสือ ปัญหาอย่างหนึ่งที่เกิดขึ้นคือ เราจะรู้ได้อย่างไรว่าข้อมูลที่ได้รับมาจากผู้ส่งที่เรากำลังติดต่ออยู่

ไม่ใช่เป็นการส่งข้อมูลมาจากผู้อื่น เช่นเดียวกับจดหมาย หนังสือ หรือเอกสารจำเป็นจะต้องมีการยืนยันผู้ส่ง หรือผู้เขียนด้วยการลงลายมือชื่อ ในโลกอินเตอร์เน็ตก็เช่นเดียวกันจะต้องมีการลงลายมือชื่อเพื่อยืนยันการส่งข้อมูล

วิธีการ XML Encryption ออกแบบมาเพื่อการเข้ารหัสโดยเฉพาะจึงไม่สามารถจัดการกับปัญหานี้โดยตรงจำเป็นที่จะต้องอาศัยกลไกหรือวิธีการอื่นเข้ามาช่วยในการแก้ปัญหา ซึ่ง W3C ได้เสนอ XML Signature สำหรับการยืนยันการส่งข้อมูลในรูปแบบของเอกสาร XML ในส่วนของวิธีการ SemCrypt และ qDAS จะอาศัยวิธีการที่เรียกว่า ใช้งานร่วมกับ Message Authentication ซึ่งมีมากมายหลายวิธีตัวอย่างเช่น DSA หรือ SHA เป็นต้น โดยจะนำมาใช้งานในขั้นตอนการรับส่งข้อมูลระหว่างผู้รับบริการและผู้ให้บริการ

5) ประเด็นเรื่องการรับส่งข้อมูลระหว่างผู้รับบริการและผู้ให้บริการ

ในระหว่างการติดต่อสื่อสารของผู้รับบริการและผู้ให้บริการ จะต้องมีการป้องกันการดักข้อมูลของผู้ที่ประสงค์ร้าย โดยที่การรับส่งข้อมูลระหว่างผู้รับบริการและผู้ให้บริการจะมีอยู่ด้วยกัน 2 กรณีคือ การส่งข้อมูลเพื่อนำไปจัดเก็บและการส่งข้อมูลเพื่อการสอบถามและตอบกลับ ในส่วนของการป้องกันจำเป็นที่จะต้องใช้ Protocol ในระบบเครือข่ายช่วยในการแก้ปัญหานี้ เช่น TLS เป็นต้น

โดยสรุปแล้วในประเด็นของการรักษาความปลอดภัย จำเป็นที่ต้องมีการรักษาความปลอดภัยในด้านต่าง ๆ ได้แก่ การพิสูจน์ตัวตน (Authentication) การกำหนดสิทธิ์ (Authorization) การเข้ารหัส (Encryption) และการรักษาความสมบูรณ์ (Integrity) นอกจากนี้แล้วจำเป็นที่จะต้องใช้การเก็บข้อมูลการใช้งาน (Log File) และใช้ Protocol ในระดับเครือข่ายช่วยในการรักษาความปลอดภัย

บทที่ 5

บทสรุปและข้อเสนอแนะ

บทนี้จะนำเสนอบทสรุปและข้อเสนอแนะ โดยในงานวิจัยนี้ได้เสนอวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ (qDAS) โดยที่ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย และมีประสิทธิภาพในการจัดเก็บ นอกจากนี้ใช้กุญแจเดียวในการเข้ารหัสข้อมูลที่เหมือนกันให้มีความแตกต่างกันเพื่อป้องกันไม่ให้ผู้โจมตีสามารถคาดเดาจำนวนข้อมูลที่เหมือนกันและมีประสิทธิภาพในการค้นหา ซึ่งจากผลการทดลองพบว่าจำนวนกุญแจที่ใช้มีจำนวนน้อยกว่า เวลาที่ใช้ในการค้นหาข้อมูลน้อยกว่าวิธีการ XML Encryption Standard และวิธีการ SemCrypt นอกจากนี้ความปลอดภัยในการทำงานไม่แตกต่างกัน

5.1 สรุปผลการวิจัย

จากการทดลองที่ได้ออกแบบไว้และผลการทดลองในบทก่อนหน้านี้ สามารถสรุปประเด็นได้ดังต่อไปนี้

5.1.1 ประเด็นการจัดเก็บข้อมูล

1) ขนาดของพื้นที่ในการจัดเก็บข้อมูลของแต่ละวิธีการมีความแตกต่างกัน โดยเรียงลำดับจากน้อยไปมากได้ดังต่อไปนี้ คือ XML Encryption, qDAS และ SemCrypt เนื่องจากวิธีการ XML Encryption มีการเพิ่มเพียงแค่ข้อมูลที่ถูกเข้ารหัสและชื่อกุญแจที่ใช้ในการเข้ารหัส แต่วิธีการ qDAS และ SemCrypt ขนาดของพื้นที่ที่เพิ่มขึ้นจะเป็นส่วนของข้อมูลที่ใช้ในการช่วยค้นหาข้อมูล

2) วิธีการ qDAS และ SemCrypt จะใช้จำนวนกุญแจในการเข้ารหัสข้อมูลที่เหมือนกันให้ได้ผลลัพธ์ที่แตกต่างกันเพียง 1 กุญแจ

3) เวลาที่ใช้ในการจัดเก็บข้อมูลของแต่ละวิธีการมีความแตกต่างกัน โดยเรียงลำดับจากน้อยไปมากได้ดังต่อไปนี้ คือ XML Encryption, qDAS และ SemCrypt เนื่องจากวิธีการ XML Encryption มีการดำเนินการเฉพาะการเข้ารหัสและการแทนที่ข้อมูล แต่วิธีการ qDAS และ SemCrypt นอกจากเข้ารหัสข้อมูลแล้วยังมีการดำเนินการสร้างข้อมูลช่วยในการค้นหา นอกจากนี้ยังเวลาที่ใช้ส่วนใหญ่จะเป็นการติดต่อกับฐานข้อมูลเพื่อจัดเก็บข้อมูล

4) เวลาที่ใช้ในการจัดเก็บข้อมูลของวิธีการ qDAS จะขึ้นอยู่กับขนาดของเอกสาร โดยจะมีค่าเพิ่มขึ้นในลักษณะเชิงเส้น (Linear) ตามขนาดของเอกสาร ซึ่งแตกต่างกับวิธีการ SemCrypt ที่มีค่าเพิ่มขึ้นในลักษณะโพลีโนเมียล (Polynomial)

5.1.2 ประเด็นการค้นหาข้อมูล

1) เวลาที่ใช้ในการค้นหาข้อมูลของแต่ละวิธีการมีความแตกต่างกัน โดยที่ในกรณีที่มีผลลัพธ์เพียงคำตอบเดียวหรือมีผลลัพธ์จำนวนน้อย วิธีการ SemCrypt จะใช้เวลาในการค้นหาข้อมูลน้อยที่สุด เนื่องจากสามารถลดเวลาในการเชื่อมตารางข้อมูล แต่ในกรณีที่มีผลลัพธ์หลายคำตอบหรือมีผลลัพธ์จำนวนมาก วิธีการ qDAS จะใช้เวลาในการค้นหาข้อมูลน้อยที่สุด เนื่องจากมีการติดต่อกับฐานข้อมูลเพียงครั้งเดียว (แตกต่างกับ SemCrypt ที่มีการติดต่อกับฐานข้อมูลเท่ากับจำนวนคำตอบของคำสั่งที่ใช้ในการค้นหา)

2) วิธีการ qDAS สามารถค้นหาข้อมูลได้ทุกรูปแบบของคำสั่งในการค้นหา ซึ่งวิธีการอื่นมีข้อจำกัดบางประการในการค้นหาข้อมูล เช่น วิธีการ SemCrypt จะสามารถค้นหาข้อมูลได้เฉพาะกรณีที่มีเงื่อนไขและวิธีการ XML Encryption จะมีข้อจำกัดในการค้นหากล่าวคือจะใช้เวลานานในการถอดรหัสข้อมูลก่อนดำเนินการค้นหา

3) ในกรณีที่ข้อมูลที่เป็นผลลัพธ์มีจำนวนมากขึ้น เวลาที่ใช้ในการค้นหาจะเพิ่มขึ้นตามจำนวนผลลัพธ์ ซึ่งวิธีการ qDAS จะใช้เวลาในการค้นหาน้อยที่สุด

5.1.3 ประเด็นความปลอดภัย

1) วิธีการ qDAS และ วิธีการ SemCrypt สามารถป้องกันการโจมตีแบบ Frequency-based โดยที่ไม่จำเป็นจะต้องใช้กุญแจหลายกุญแจในการเข้ารหัสข้อมูล ซึ่งแตกต่างจากวิธีการ XML Encryption

2) ความปลอดภัยในการทำงานของแต่ละวิธีการที่ได้เปรียบเทียบไม่มีความแตกต่างกัน แต่จำเป็นที่ต้องใช้วิธีการหรือกลไกอื่นๆ เข้ามาช่วยในการรักษาความปลอดภัย

5.2 ข้อเสนอแนะ

เนื่องจากความปลอดภัยของการทำงานเป็นสิ่งที่สำคัญ ในหลายๆงานวิจัยมักจะแยกประเด็นในการรักษาความปลอดภัย จึงทำให้เกิดข้อจำกัดในการทำงานร่วมกันเช่น วิธีการหนึ่งเหมาะสำหรับการปกปิดความลับ และอีกวิธีการเหมาะสำหรับการตรวจสอบสิทธิ์การใช้งาน โดยที่แต่ละวิธีการจะทำงานได้ดีในแต่ละปัญหา แต่หากนำสองวิธีการมาร่วมกันอาจเกิดปัญหาเนื่องจากรูปแบบการจัดเก็บข้อมูลที่ไม่เหมือนกัน หรืออาจเป็นการทำงานที่ซ้ำซ้อนกัน

ดังนั้นการออกแบบระบบหรือวิธีการควรที่จะสามารถแก้ไขปัญหาได้พร้อมกันหลายปัญหาหรือป้องกันการโจมตีได้หลายรูปแบบเพื่อที่จะสามารถนำไปใช้งานจริง

ในส่วนของฟังก์ชัน Hash เป็นอีกส่วนหนึ่งที่จำเป็นจะต้องศึกษาเพิ่มเติม กล่าวคือ ฟังก์ชัน Hash เป็นฟังก์ชันที่มีโอกาสที่จะเกิดการชนกันของข้อมูล นอกจากนี้ฟังก์ชัน Hash บางประเภทมีโอกาสที่จะสามารถถูกแก้ข้อมูลกลับได้ ดังนั้นจึงจำเป็นที่จะต้องมีการศึกษาเพิ่มเติมในส่วนของการเลือกใช้ฟังก์ชัน Hash ให้มีความเหมาะสมกับการทำงานให้มีประสิทธิภาพและความปลอดภัยเพิ่มขึ้น

บรรณานุกรม

Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y. 2004. Order Preserving Encryption for Numeric Data. SIGMOD 2004. Maison de la Chimie, Paris, France June 13-18, 2004. p.563 – 574.

Bartel, M., Boyer J., Fox, B., LaMacchia, B. and Simon, E. 2008. XML Signature Syntax and Processing (Second Edition). W3C Recommendation.

Bouganim, L. and Pucheral, P. 2002. Chip-Secured Data Access: Confidential Data on Untrusted Servers, in: Proc. of the 28th Int. Conf. on Very Large Data Bases (VLDB). Hong Kong, China. August 20-23, 2002. p.131–142.

Chan, L.M. 1996. Dewey Decimal Classification: A Practical Guide, second ed., OCLC Forest Press.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. 2001. Introduction to Algorithms (2e ed.). MIT Press and McGraw-Hill.

Ekelhart, A., Fenz, S., Goluch, G., Steinkellner, M. and Weippl, E. 2008. XML security - A comparative literature review. Journal of Systems and Software. v.81: p.1715-1724.

Feng, L. and Jonker, W. 2003. Efficient Processing of Secured XML Metadata. In: On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, 3-7 November 2003, Catania, Sicily, Italy. pp. 704-717.

Gallaugher, J. 1996. The Critical Choice of Client Server Architecture: A Comparison of Two and Three Tier Systems, A future issue of Information Systems Management, Auerbach Publications, New York.

Gao, J., Wang, T. and Yang, D. 2008. XFlat: Query-friendly encrypted XML view publishing. Information Sciences 178(3): p.774-787.

Hacigumus, H., Mehrotra, S. and Iyer, B. 2002. Providing Database as a Service. Proceedings of the 18th international Conference on Data Engineering. 26 February – 1 March 2002, pp. 29-40.

Imamura, T., Dillaway, B. and Simon, E. 2002. XML Encryption Syntax and Processing. W3C Recommendation.

Jammalamadaka, R.C. and Mehrotra, S. 2006. Querying Encrypted XML Documents. Proceedings of the 10th International Database Engineering and Applications Symposium. Berkeley, CA, USA, December 11-14, 2006. p.129-136.

Kahate, A. 2006. Cryptography and network security. McGraw-Hill: India.

Katz, J., Lindell, Y. 2007. ”Introduction to modern cryptography”. Chapman & Hall/CRC Press

Lee, J.G. and Whang, K.Y. 2006. Secure query processing against encrypted XML data using query-aware decryption, Inform. Sci. 176 (13) (2006), pp. 1928–1947.

Megginson, D. Simple API for XML. 2004. http://www.saxproject.org/ (accessed 18/5/2010).

Min, J., Lee, C. and Chung, C. 2008. XTRON: An XML data management system using relational databases. Information and Software Technology 50 (2008): p.462–479.

Paar, C., Pelzl, J., 2009. "The Advanced Encryption Standard". Springer.

PaperMaster. http://www.papermasters.com/encryption-techniques.html (accessed 10/3/2553).

Schmidt, A. R., Waas, F., Kersten, M. L., Florescu, D., Manolescu, I., Carey, M. J. and Busse, R.. 2001. The XML Benchmark Project. Technical Report INS-R0103, CWI, Amsterdam, The Netherlands.

Schrefl, M., Grün, K. and Dorn, J. 2005. SemCrypt–Ensuring privacy of electronic documents through semantic-based encrypted query processing, PDM 2005 International Workshop on Privacy Data Management, Workshop Dates Tokyo, Japan, April 8-9, 2005. p.1191.

Silberschatz, A., Korth, H.F., and Sudarshan S. 2006. Database system Concepts. McGraw- Hill international edition. Fifth edition. McGraw-Hill: Singapore.

Socolofsky, T., ”A TCP/IP Tutorial”, RFC 1180, Network Working Group, January 1991.

Song, D., Wagner, D. and Perrig, A. 2000. Practical techniques for searches on encrypted data in Proceedings of IEEE Symposium on Security and Privacy 2000. pp.44–55.

Ünay, O. and Gündem, T.I. 2008. A survey on querying encrypted XML documents for databases as a service. ACM SIGMOD Record. March 2008: p. 12-20.

W3C. Document Object Model (DOM). 2005. http://www.w3.org/DOM/ (accessed 18/5/2010)

W3C. Extensible Markup Language (XML). 2009. http://www.w3.org/XML/ (accessed 7/3/2010).

W3C. Recommendation XML Path Language (XPath) 2.0 Available from: http://www.w3.org/TR/xpath20/. 2007.

W3C. XML Query (XQuery). 2009. http://www.w3.org/XML/Query/ (accessed May 7/3/2010).

Wang, H. and Lakshmanan, L.V.S. 2006. Efficient secure query evaluation over encrypted XML databases. Proceedings of the 32nd international conference on Very large data bases. Seoul, Korea, September 12-15, 2006. p.127–138.

Yang, Y., Ng, W., Lau, H.L. and Cheng, J. 2006. An Efficient Approach to Support Querying Secure Outsourced XML Information. Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006. Luxembourg, Luxembourg, June 5-9, 2006. p.157-171.

Yue, L., Ren, J., Qian, Y. 2009. Storage Method of XML Documents Based-on the Pri-order Labeling Scheme. etcs, vol. 2, pp.127-131, 2009 First International Workshop on Education Technology and Computer Science.

ภาคผนวก

ภาคผนวก ก.

ผลงานตีพิมพ์

เรื่อง	Querying Encrypted XML Document Approach for a Database-service-provider
งานประชุมวิชาการ	The 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010)
สถานที่	กรุงเทพมหานคร ประเทศไทย
วันที่	12 – 14 พฤษภาคม 2553

การสอบถามเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ

Querying Encrypted XML Document Approach for a Database-service-provider

ต่วนกัสฟี หะมะ, ศิริรัตน์ วณิชโยบล และ ลัดดา ปรีชาวีรกุล

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

อำเภอหาดใหญ่ จังหวัดสงขลา 90112

E-mail: {s5110220027, sirirut.v, ladda.p}@psu.ac.th

บทคัดย่อ

ในการทำงานปัจจุบันข้อมูลข่าวสารมีปริมาณเพิ่มมากขึ้น จึงทำให้ผู้ให้บริการทางด้านฐานข้อมูลมีจำนวนเพิ่มมากขึ้นเช่นเดียวกัน นอกจากนี้ข้อมูลในปัจจุบันนั้นเป็นข้อมูลที่มีความสำคัญต่อบุคคลหรือองค์กร เช่น ข้อมูลทางด้านการเงิน เป็นต้น เอกสาร XML เป็นเอกสารที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างผู้รับบริการ ในการที่จะปกปิดเนื้อหาของข้อมูลจำเป็นที่จะต้องใช้การเข้ารหัสข้อมูล การเข้ารหัสข้อมูลทั้งหมดแล้วส่งต่อไปฝากไว้ที่ผู้ให้บริการ วิธีการนี้สามารถสร้างความปลอดภัยให้กับข้อมูล แต่จะไม่มีประสิทธิภาพในการค้นหาข้อมูล เนื่องจากต้องถอดรหัสข้อมูลทั้งหมดแล้วจึงสามารถนำมาค้นหาคำตอบในฝั่งของผู้รับบริการ ซึ่งจะเสียเวลาและทรัพยากรในการทำงาน ในงานวิจัยนี้ได้เสนอวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ โดยที่ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย และมีประสิทธิภาพโดยใช้กุญแจเดียว

คำสำคัญ: XML, ผู้ให้บริการทางด้านฐานข้อมูล

Abstract

Today, there is much information in the world. Consequently, database service provider has been increased. Moreover, some information is important for organization. XML document is standard document for exchanging data. When we would like to hide important information, we must encrypt that. All encrypted document has deal with security but it is ineffective searching. The document would be decrypted before execute on client side. In this paper, we purpose querying encrypted XML document approach for a database-service-provider. In this approach, same data will be encrypted using single key as different for security and efficiency.

Key Words: XML, Database-service-provider

1.บทนำ

ปัจจุบันระบบอินเตอร์เน็ตมีความสำคัญในการดำรงชีวิตประจำวัน โดยที่จะมีข้อมูลข่าวสารเกิดขึ้นอย่างรวดเร็วและมีปริมาณมหาศาล จึงทำให้จำเป็นที่จะต้องมีระบบในการจัดเก็บข้อมูลที่มีประสิทธิภาพ เนื่องจากประสิทธิภาพในการทำงานของเครื่องผู้รับบริการโดยทั่วไปมีต่ำกว่าประสิทธิภาพในการทำงานของเครื่องผู้ให้บริการ ดังนั้นการจัดเก็บข้อมูลในฝั่งของผู้ให้บริการ (Service Provider) จะมีประสิทธิภาพมากกว่า และได้มีการพัฒนามาอย่างต่อเนื่อง รูปแบบหนึ่งที่ได้รับความนิยมคือ Database as a service [1] เป็นการฝากข้อมูลไว้ที่ผู้

ให้บริการจะทำให้เกิดความสะดวกในการจัดเก็บข้อมูล เนื่องจากมีพื้นที่จัดเก็บข้อมูลขนาดใหญ่ การเรียกค้นหาข้อมูลได้อย่างรวดเร็ว และประหยัดค่าใช้จ่ายในเรื่องผู้ดูแลระบบ แต่การฝากข้อมูลไว้ที่ผู้ให้บริการ ก็มีปัญหาที่ตามมานั่นคือ ความปลอดภัยของข้อมูล เนื่องจากการฝากข้อมูลไว้ที่ผู้ให้บริการจะทำให้ผู้ให้บริการทราบรายละเอียดของข้อมูลที่สำคัญ เช่น ข้อมูลบัตรเครดิต เป็นต้น ดังนั้นจะมีวิธีการอย่างไรในการฝากข้อมูลที่มีความสำคัญไว้กับผู้ให้บริการที่ไม่น่าเชื่อถือ

มาตรฐานการจัดเก็บข้อมูลที่ได้รับความนิยมในขณะนี้ คือ XML [2] ซึ่งเป็นมาตรฐานที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่อง นอกจากนี้แล้ว XML ยังมีรูปแบบเดียวกับภาษา HTML ที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างอินเตอร์เน็ต ทำให้สะดวกในการทำงานร่วมกัน ข้อดีอีกอย่างหนึ่งของ XML ที่แตกต่างจาก HTML คือ เป็นข้อมูลที่มนุษย์และเครื่องคอมพิวเตอร์สามารถทำความเข้าใจได้โดยง่าย

ในการที่จะปกปิดเนื้อหาของข้อมูลจำเป็นที่จะต้องใช้วิธีการควบคุมการเข้าถึงหรือการเข้ารหัสข้อมูล ในส่วนของวิธีการเข้ารหัสข้อมูลจะเข้ารหัสข้อมูลทั้งหมดแล้วส่งต่อไปฝากไว้ที่ผู้ให้บริการ วิธีการนี้สามารถสร้างความปลอดภัยให้กับข้อมูล แต่จะไม่มีประสิทธิภาพในการค้นหาข้อมูล เนื่องจากเมื่อต้องการข้อมูลเพียงบางส่วนจะต้องถอดรหัสข้อมูลทั้งหมดแล้วจึงนำข้อมูลที่ถูกถอดรหัสมาค้นหาคำตอบในฝั่งของผู้รับบริการ ซึ่งจะเสียเวลาและทรัพยากรในการทำงาน ดังนั้นการค้นหาข้อมูลควรดำเนินการในฝั่งของผู้ให้บริการซึ่งมีเครื่องคอมพิวเตอร์ที่ทำงานได้อย่างรวดเร็วและมีประสิทธิภาพมากกว่า

นอกจากนี้ยังมีวิธีการอื่นในการเข้ารหัสเอกสาร XML [3 - 8] ซึ่งเป็นการเข้ารหัสและค้นหาข้อมูลในเอกสารที่ถูกเข้ารหัส แต่ปัญหาของวิธีการเหล่านั้นคือหากข้อมูลใดเป็นข้อมูลที่เหมือนกัน เมื่อมีการเข้ารหัสก็จะทำให้ได้ข้อมูลที่ถูกเข้ารหัสเหมือนกัน และเมื่อนำไปจัดเก็บในฝั่งของผู้ให้บริการก็จะทำให้ผู้ให้บริการทราบจำนวนข้อมูลที่เหมือนกัน สำหรับการแก้ไขปัญหานี้โดยส่วนใหญ่แล้วจะใช้วิธีการเข้ารหัสข้อมูลโดยใช้กุญแจที่แตกต่างกัน ในกรณีที่มีข้อมูลเหมือนกันเป็นจำนวนมากก็ต้องใช้กุญแจเป็นจำนวนมากเช่นเดียวกัน ทำให้เพิ่มพื้นที่ในการจัดเก็บกุญแจ และเมื่อต้องการเข้าถึงข้อมูลที่ซ้ำกันจะต้องมีการส่งข้อมูลเพื่อไปสอบถามเท่ากับจำนวนข้อมูลที่ซ้ำกันทั้งหมด

ในงานวิจัยนี้ได้มุ่งเน้นที่จะนำเสนอวิธีการในการเข้ารหัสและค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ โดยที่ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย และมีประสิทธิภาพในการจัดเก็บ และการค้นหาข้อมูลได้อย่างรวดเร็ว

เอกสารนี้ประกอบด้วย 6 ส่วน โดยส่วนที่ 2 อธิบายถึงทฤษฎีที่เกี่ยวข้อง ส่วนที่ 3 จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการเข้ารหัสเอกสาร XML ส่วนที่ 4 เป็นการนำเสนอเทคนิคการเข้ารหัสข้อมูลเอกสาร XML แบบใหม่ ส่วนที่ 5 แสดงผลการเปรียบเทียบประสิทธิภาพ และสุดท้ายส่วนที่ 6 เป็นบทสรุป

2.ทฤษฎีที่เกี่ยวข้อง

ในส่วนนี้จะเป็นการกล่าวถึง ภาษา XML และ ความปลอดภัยของเอกสาร XML อย่างสังเขป

2.1 ภาษา XML (Extensible Markup Language)

ภาษา XML เป็นภาษาที่ใช้ในการนิยามข้อมูลและมีลักษณะเป็นภาษา Markup ซึ่งถูกกำหนดเป็นมาตรฐานโดย W3C โดยที่ผู้ใช้สามารถกำหนดแท็ก (Tag) และอิลิเมนต์ (Element) ขึ้นมาเองได้และทำให้เข้าใจความหมายของข้อมูล ซึ่งแตกต่างจาก ภาษา HTML ที่ไม่สามารถกำหนดแท็กขึ้น นอกจากนี้แล้วสามารถใช้ในการแลกเปลี่ยนข้อมูลกับบุคคลอื่นได้โดยง่าย โดยในภาษา XML จะมีแท็กเปิดและแท็กปิดเช่นเดียวกับภาษา HTML รูปที่ 1 แสดงตัวอย่างเอกสาร XML ของข้อมูลคนไข้

```
<patients>
 <patient>
  <lastname>Smith</lastname>
  <firstname>Alan</firstname>
  <case_history>
   <address>Hadyai</address>
   <infection>Haemophilus</infection>
   <medicine>Floxin</medicine>
  </case_history>
 </patient>
</patients>
```

รูปที่ 1 ตัวอย่างเอกสาร XML

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
<xs:element name="Patients">
 <xs:complexType><xs:sequence>
   <xs:element name="Patient">
    <xs: complexType><xs:sequence>
      <xs:element name="lastname" type="xs:string"/>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="case_history">
       <xs: complexType><xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="infection" type="xs:string"/>
        <xs:element name="medicine" type="xs:string"/>
      .
      .
      .
 </xs:element>
</xs:schema>
```

รูปที่ 2 ตัวอย่าง XMLShcema

นอกจากนี้แล้วยังมี XML Schema ซึ่งจะนำมาใช้ในการกำหนดรูปแบบโครงสร้างของเอกสาร XML และได้รับมาตรฐานจาก W3C ในปี ค.ศ. 2001 ซึ่งมีข้อดีมากกว่า XML DTD เช่น สามารถกำหนดรูปแบบข้อมูลตามต้องการ สร้างชนิดข้อมูลใหม่โดยอ้างอิงจากชนิดข้อมูลที่มีอยู่เดิม และสามารถกำหนดจำนวนและลำดับของอิลิเมนต์ลูกได้ [9] โดยที่ XML Schema ในรูปที่ 2 เป็นของเอกสาร XML ในรูปที่ 1

XPath [10] เป็นภาษาที่ใช้ในการสอบถามข้อมูลบนเอกสาร XML ซึ่งสามารถระบุตำแหน่งที่ต้องการบนเอกสาร XML และสามารถกำหนดเงื่อนไขในการค้นหา

2.2 ความปลอดภัยของเอกสาร XML (XML Security)

เนื่องจากความไม่น่าเชื่อถือของผู้ให้บริการในการรับฝากข้อมูลเอกสาร XML จึงทำให้ผู้รับบริการจำเป็นที่จะต้องมีวิธีการในการแก้ปัญหานี้ โดยที่วิธีการในการรักษาความปลอดภัยของเอกสาร XML มีอยู่หลายวิธี

วิธีการที่สะดวกที่สุดในการรักษาความปลอดภัยของข้อมูลในกรณีที่ฝากเก็บข้อมูลไว้กับผู้อื่นคือการเข้ารหัสข้อมูลทั้งหมดแล้วนำไปจัดเก็บ ถึงแม้วิธีการนี้จะสามารถเก็บข้อมูลให้เป็นความลับ แต่จะประสบปัญหาต่างๆเช่น การค้นหาข้อมูล การเพิ่มข้อมูลใหม่ และการแก้ไขข้อมูลเดิม เป็นต้น

ในปี ค.ศ. 2002 W3C ได้มีการกำหนดมาตรฐานในการเข้ารหัสเอกสาร XML (XML Encryption) [11] ซึ่งจะมีการแปลงข้อมูลแท็กของ XML ที่ต้องการเข้ารหัส ให้เป็นข้อมูลใหม่ประกอบไปด้วย 4 ส่วนคือ 1) EncryptionMethod ใช้สำหรับจัดเก็บวิธีการในการเข้ารหัส 2) KeyInfo ใช้สำหรับจัดเก็บชื่อของกุญแจที่ใช้ในการเข้ารหัส 3) CipherData ใช้สำหรับจัดเก็บข้อมูลที่ถูกเข้ารหัส และ 4) EncryptionProperties ใช้สำหรับจัดเก็บข้อมูลเสริมเกี่ยวกับข้อมูลที่ถูกเข้ารหัส ซึ่งการจัดเก็บข้อมูลเหล่านี้จะรวมกันเป็นกลุ่มเดียวกันทำให้ไม่มีประสิทธิภาพในการค้นหา และต้องถอดรหัสก่อนค้นหา โดยที่การเข้ารหัสข้อมูลเอกสาร XML รูปที่ 1 จะได้ผลลัพธ์ดังรูปที่ 3

```
<patients>
    <patient>
        <lastname>Smith</lastname>
        <firstname>Alan</firstname>
        <case_history>
        <address>Hadyai</address>
        <EncryptedData>
         <EncryptionMethod/>
        <ds:KeyInfo >
        <KeyName>theKey</KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>qvpAWAE</CipherValue>
        </CipherData>
        </EncryptedData>
        </case_history>
    </patient>
</patients>
```

รูปที่ 3 ตัวอย่างเอกสาร XML เมื่อเข้ารหัสโดยใช้ XML Encryption

3. งานวิจัยที่เกี่ยวข้อง

การเข้ารหัสเอกสาร XML และจัดเก็บไว้ที่ผู้ให้บริการ มีนักวิจัยหลายท่านได้นำเสนอวิธีการต่างๆ เช่น Wang และ Lakshmanan [7] ได้เสนอวิธีการจัดเก็บ Metadata ซึ่งเป็นดัชนีที่ใช้สำหรับช่วยในการประมวลผลคำสั่งในการค้นหา (Query) ไว้ที่ฝั่งผู้ให้บริการ เพื่อลดการทำงานในฝั่งผู้รับบริการ และยังสามารถกำหนดกฎเกณฑ์ในการเข้ารหัสข้อมูลเพียงบางส่วน นอกจากนี้วิธีการของ Feng และ Jonker [3] ได้เสนอวิธีการจัดเก็บ Mapped Values ไว้ในฝั่งผู้ให้บริการและใช้ข้อมูล XML DTD มาช่วยในการเข้ารหัสเส้นทางโดยใช้ Hash Function ซึ่งสามารถค้นหาข้อมูล XML ที่เป็นจำนวนตัวเลข โดยการแบ่งเป็นช่วงของข้อมูลจำนวนตัวเลข

นอกจากวิธีการที่มีการจัดเก็บข้อมูลดัชนีไว้ในฝั่งผู้ให้บริการแล้ว ยังมีอีกวิธีการหนึ่งซึ่งมีการจัดเก็บข้อมูล Index ไว้ในฝั่งผู้รับบริการเสนอโดย Yang และคณะ [8] ซึ่งมีวิธีการในการบีบอัดโครงสร้างข้อมูลให้มีขนาดเล็กลง โดยใช้ Vectorization และ Skeleton Compression เพื่อที่จะสามารถประมวลผลในหน่วยความจำหลักของผู้รับบริการ นอกจากนี้แล้วยังมีกระบวนการแปลงข้อมูล XML ให้เป็น Relational Data และกระบวนการแปลง Query ให้อยู่ในรูปของ SQL วิธีการที่นำเสนอโดย Schrefl และคณะ [4] เป็นวิธีการที่แตกต่างกับวิธีการอื่นนั่นคือ มีการแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้รับบริการในการเข้าถึงข้อมูล นอกจากนี้ยังสนับสนุนการเพิ่มข้อมูลและการแก้ไขข้อมูลในเอกสาร XML ที่ถูกเข้ารหัส

ในปี ค.ศ. 2008 Gao และคณะ [12] ได้เสนอวิธีการในการเข้ารหัสเอกสาร XML ที่ใช้สำหรับการเผยแพร่ โดยที่ผู้ใช้แต่ละกลุ่มจะมีสิทธิ์ในการเข้าถึงเอกสาร XML ที่เผยแพร่แตกต่างกัน โดยวิธีการนี้เมื่อผู้ใช้ต้องการเข้าถึงข้อมูลส่วนไหนจะทำการสร้างโครงสร้างต้นไม้ย่อยของข้อมูลส่วนนั้นพร้อมกับตรวจสอบสิทธิ์การใช้งานก่อนการรวมต้นไม้ย่อยเป็นหนึ่งเดียว หลังจากนั้นจะถอดรหัสข้อมูลตามสิทธิ์ของผู้ใช้ในแต่ละกลุ่ม

ปัญหาของวิธีการที่กล่าวมาข้างต้นคือหากข้อมูลใดเป็นข้อมูลที่เหมือนกัน เมื่อมีการเข้ารหัสก็จะทำให้ได้ข้อมูลที่ถูกเข้ารหัสเหมือนกัน และเมื่อนำไปจัดเก็บในฝั่งของผู้ให้บริการก็จะทำให้ผู้ให้บริการทราบจำนวนข้อมูลที่เหมือนกัน ดังนั้นในงานวิจัยนี้จึงมุ่งเน้นเสนอวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ โดยข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย และมีประสิทธิภาพในการจัดเก็บโดยใช้กุญแจเดียวในการเข้ารหัส

4. วิธีการเข้ารหัสข้อมูลเอกสาร XML

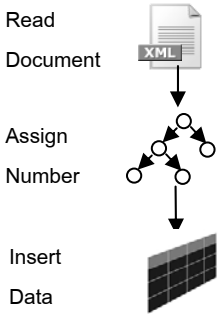
ในส่วนนี้จะเป็นการนำเสนอวิธีการเข้ารหัสข้อมูล ซึ่งจะมีการจัดเก็บข้อมูลไว้ในฝั่งผู้ให้บริการและเข้ารหัสข้อมูลเฉพาะส่วนที่ต้องการเท่านั้น โดยใช้กุญแจเดียวในการเข้ารหัสข้อมูลที่เหมือนกันให้มีความแตกต่างกันเพื่อป้องกันไม่ให้ผู้โจมตีสามารถคาดเดาจำนวนข้อมูลที่เหมือนกัน เป็นการประหยัดพื้นที่ แต่ยังคงประสิทธิภาพในการค้นหา การทำงานแบ่งออกเป็น 2 กระบวนการ คือ กระบวนการจัดเก็บข้อมูลและการค้นหาข้อมูล

4.1 กระบวนการจัดเก็บข้อมูลเอกสาร XML (XML Data Storage)

กระบวนการจัดเก็บข้อมูลนี้ผู้รับบริการจะทำการสร้างตารางเก็บข้อมูลและทำการเข้ารหัสข้อมูลเฉพาะส่วนที่ต้องการในเอกสาร XML จากนั้นผลลัพธ์ที่ได้จะถูกจัดเก็บไว้ในฝั่งของผู้ให้บริการ ในกระบวนการนี้ประกอบด้วย 3 ขั้นตอนย่อยคือ ขั้นตอนการอ่านเอกสาร XML (Read Document) ขั้นตอนการกำหนดหมายเลข (Assign Number) และขั้นตอนการบันทึกข้อมูล (Insert Data) ดังแสดงในรูปที่ 4

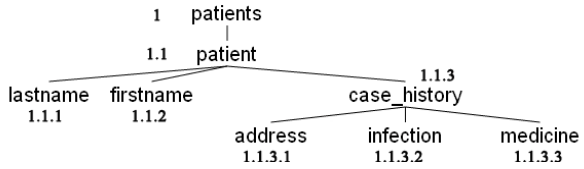
1. ขั้นตอนการอ่านเอกสาร XML ในขั้นตอนนี้เอกสาร XML ในรูปของ Text File หรือ ไฟล์นามสกุล

.xml จะถูกอ่านเข้ามาพร้อมกับ XML Schema ของเอกสารนั้น



รูปที่ 4 กระบวนการจัดเก็บข้อมูลเอกสาร XML

2. ขั้นตอนการกำหนดหมายเลข ในขั้นตอนนี้ข้อมูลในเอกสาร XML จะถูกแปลงให้อยู่ในรูปโครงสร้างข้อมูลแบบต้นไม้ (XML Tree) และมีการกำหนดหมายเลขให้กับแต่ละโหนด ในงานวิจัยทั่วไปมีวิธีการกำหนดหมายเลขให้กับโหนดของเอกสาร XML 2 รูปแบบ คือแบบ Region และแบบ Prefix ซึ่งในงานวิจัยนี้ใช้วิธีการแบบ Prefix ของ Dewey [13] โดยกำหนดหมายเลขจากบนลงล่างและซ้ายไปขวา ทำให้แต่ละข้อมูลจะมีหมายเลขที่ไม่ซ้ำกัน การกำหนดหมายเลขจะนำมาช่วยในการเข้ารหัสข้อมูลเพื่อให้ข้อมูลที่เหมือนกันถูกเข้ารหัสแล้วได้ผลลัพธ์ที่ไม่เหมือนกัน และสามารถนำหมายเลขที่กำหนดให้มาช่วยในการค้นหาข้อมูลที่ถูกเข้ารหัส ข้อดีของวิธีการนี้คือเมื่อมีการแก้ไขโครงสร้างของเอกสารเช่น การเพิ่มหรือลบข้อมูลไม่ส่งผลกระทบต่อการกำหนดหมายเลขให้กับข้อมูลใหม่ที่เข้ามา [14] รูปที่ 5 แสดงโครงสร้างต้นไม้หลังจากกำหนดหมายเลขของเอกสาร XML ในรูปที่ 1



รูปที่ 5 เอกสาร XML ในรูปแบบโครงสร้างต้นไม้ เมื่อกำหนดหมายเลข

3. ขั้นตอนการบันทึกข้อมูล ซึ่งจะเป็นการบันทึกข้อมูลลงในตารางฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เพื่อเพิ่มความสะดวกในการจัดการและประสิทธิภาพในการค้นหาข้อมูล [8, 15]

ในงานวิจัยนี้มีการบันทึกข้อมูลลงใน 2 ตาราง คือ ตาราง Path และ ตาราง Value

ตาราง Path ประกอบด้วย 3 ฟิลด์ (Fields) ดังนี้ Path(Path_Id, Path_Name, Encrypted) โดยที่

Path_Id เก็บข้อมูลรหัสของ Path Name ซึ่งมีค่าไม่ซ้ำกัน

Path_Name เก็บข้อมูล Path ทั้งหมดในเอกสาร XML ได้มาจาก XML Schema

Encrypted เก็บข้อมูลการระบุว่า Path ไหนที่มีการเข้ารหัส

ตาราง Value ประกอบด้วย 5 ฟิลด์ ซึ่งแต่ละเรคคอร์ด (Record) จะเก็บค่าข้อมูลแต่ละโหนดของ XML Tree ที่ได้จากขั้นตอนที่ 2 โครงสร้างของตาราง Value คือ Value(Node_Id, Path_Id, Key_Name, Hash_Value , Node_Data) โดยที่

Node_Id เก็บข้อมูลหมายเลขของแต่ละโหนด

Path_Id เก็บข้อมูลรหัสของ Path Name ของแต่ละโหนด

Key_Name เก็บข้อมูลชื่อกุญแจที่ใช้ในการเข้ารหัส (กรณีที่ใช้มากกว่าหนึ่งกุญแจในการเข้ารหัสข้อมูล)

Hash_Value เก็บข้อมูล ผลลัพธ์ที่ได้จากการผ่านค่าข้อมูล Node_Data กับ Node_Id ของโหนดที่มีข้อมูลที่ต้องการปกปิดไปยังสมการที่ 1 เพื่อช่วยในการค้นหาข้อมูลในตาราง

$$Hash\_Value = \text{Hash}(Node\_Id + \text{Hash}(Node\_Data)) \quad (1)$$

ผู้วิจัยเลือกใช้ฟังก์ชัน Hash โดยใช้กระบวนการ Shift-add-XOR [16] ซึ่งเป็นฟังก์ชันที่ไม่สามารถแปลงข้อมูลย้อนกลับได้และมีขั้นตอนการทำงานดังรูปที่ 6 ยกตัวอย่างเช่น Hash(“Haemophilus”)= 741258963 ซึ่งค่า 741258963 จะไม่สามารถดำเนินการย้อนกลับเพื่อให้ได้ออกมาเป็น Haemophilus

```
Hash (key){
      h = 0
      for ( i=0; i < key.length; i++ )
            h ^= Shift_left(h) + Shift_right(h) + key[i]
      return h
}
```

รูปที่ 6 ขั้นตอนวิธี Shift-add-XOR

Node_Data เก็บค่าข้อมูลซึ่งอาจเป็นค่าที่ไม่ได้เข้ารหัสหรือค่าที่เข้ารหัสแล้ว ขึ้นอยู่กับค่า Key_Name โดยค่าที่ต้องการจะปกปิดจะถูกนำมาเข้ารหัสโดยผ่านสมการที่ 2

Encrypted_Data = Encrytion(*Node_Id+Node_Data,Key_Name*) (2)

ทั้งนี้ตาราง Path และ ตาราง Value จะถูกจัดเก็บไว้ที่ฝั่งผู้ให้บริการ ส่วนในฝั่งผู้รับบริการจะเก็บตาราง Path_Client ซึ่งบรรจุเฉพาะข้อมูล Record ที่มีฟิลด์ Encrypted เท่ากับ ‘y’ ในตาราง Path และเก็บข้อมูลกุญแจพร้อมชื่อของกุญแจนั้น

ตัวอย่างเช่น เอกสาร XML ในรูปที่ 1 หลังจากผ่านกระบวนการจัดเก็บข้อมูลจะได้ตาราง Path และตาราง Value ดังแสดงในตารางที่ 1 และ 2 ตามลำดับ ซึ่งตาราง Path ได้มาจากข้อมูลใน XML Schema และตาราง Value ได้มาจาก XML Tree สองตารางนี้ จะถูกจัดเก็บไว้ที่ฝั่งผู้ให้บริการ รายละเอียดของข้อมูลอธิบายได้ดังนี้ ถ้าข้อมูลเป็นข้อมูลที่ไม่เข้ารหัส (ไม่ปกปิด) เช่น lastname ในตาราง Path จะจัดเก็บ Path_Id ของ lastname เป็น 1 Path_Name เป็น Patients/Patient/lastname และEncypted เป็น ‘n’ ในตาราง Value จะจัดเก็บ Node_Id เป็น 1.1.1 Path_Id เป็น 1 และNode_Data เป็น “Smith” โดยที่ไม่ต้องเก็บค่า Key_Name และ Hash_Value

ถ้าข้อมูลเป็นข้อมูลที่เข้ารหัส (ปกปิด) เช่น medicine ในตาราง Path จะจัดเก็บ Path_Id ของ medicine เป็น 5 Path_Name เป็น Patients/Patient/case_history/medicine และEncypted เป็น ‘y’ ในตาราง Value จะจัดเก็บ Node_Id เป็น 1.1.3.3 Path_Id เป็น 5 Key_Name เป็น “Name_key1” Hash_Value เป็น 326514789(นั่นคือ Hash(“1.1.3.3”+ Hash(“Floxin”))) และ Node_Data เป็น “&*()_+” (นั่นคือ Encryption(“1.1.3.3Floxin”, “Name_key1”))

ตารางที่ 1 ตาราง Path

Path_Id	Path_Name	Encrypted
1	Patients/Patient/lastname	n
2	Patients/Patient/firstname	n
3	Patients/Patient/case_history/address	n
4	Patients/Patient/case_history/infection	y
5	Patients/Patient/case_history/medicine	y

ตารางที่ 2 ตาราง Value

Node_Id	Path_Id	Key_Name	Hash_Value	Node_Data
1.1.1	1			Smith
1.1.2	2			Alan
1.1.3.1	3			Hadyai
1.1.3.2	4	Name_key1	142357869	!@#$%^
1.1.3.3	5	Name_key1	326514789	&*()_+

4.2 การค้นหาข้อมูล (Querying)

หลังจากจัดเก็บข้อมูลในฝั่งผู้ให้บริการ ในงานวิจัย [3 - 8] จะใช้วิธีการเข้ารหัสข้อมูลโดยใช้กุญแจที่แตกต่างกัน ซึ่งในกรณีที่มีข้อมูลซ้ำกันเป็นจำนวนมากก็ต้องใช้กุญแจเป็นจำนวนมากเช่นเดียวกัน ทำให้เพิ่มพื้นที่ในการจัดเก็บกุญแจ และเมื่อต้องการเข้าถึงข้อมูลที่ซ้ำกันจะต้องมีการส่งข้อมูลเพื่อไปสอบถามเท่ากับจำนวนข้อมูลที่ซ้ำกันทั้งหมด

ในงานวิจัยนี้ได้เสนอวิธีการเข้าถึงข้อมูลเพื่อแก้ไขปัญหาเหล่านั้น โดยที่ขั้นตอนเข้าถึงข้อมูลแสดงดังรูปที่ 7 อธิบายได้ดังนี้

1. ผู้รับบริการกำหนด Query ซึ่งมีอยู่ 2 รูปแบบคือ คำสั่งในการค้นหาแบบไม่มีเงื่อนไข (Unconditional Query: UQ) และคำสั่งในการค้นหาแบบมีเงื่อนไข (Conditional Query: CQ) ตารางที่ 3 แสดงตัวอย่าง Query โดยที่ Q1-Q2 เป็นแบบ UQ ส่วน Q3-Q6 เป็นแบบ CQ

2. Query ถูกเปลี่ยนโดยกระบวนการแปลงคำสั่งก่อนส่งไปยังผู้ให้บริการ

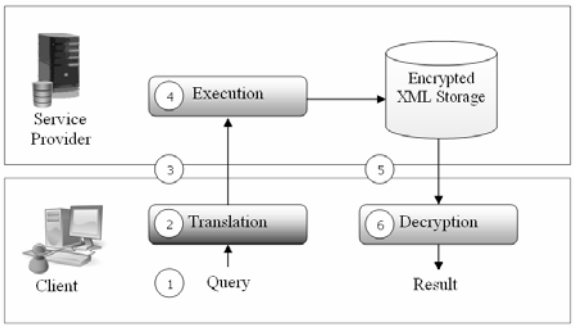
3. ผู้รับบริการส่งข้อมูลไปยังผู้ให้บริการ

4. ผู้ให้บริการค้นหาข้อมูล

5. ผู้ให้บริการส่งข้อมูล เช่น Node_Data, Node_Id และ Key_Name ที่ได้จากการค้นหากลับไปยังผู้รับบริการ ในที่นี้ผลลัพธ์ที่ต้องการเป็นข้อมูลที่ถูกเข้ารหัสจะมีการส่งชื่อกุญแจกลับไปพร้อมกับข้อมูลด้วย

6. ผู้รับบริการถอดรหัสข้อมูลโดยผ่านสมการที่ 3 และได้ผลลัพธ์

Node_Id+Node_Data=Decrytion(*Encrypted_Data ,Key_Name*) (3)



รูปที่ 7 ขั้นตอนการเข้าถึงข้อมูล

ตารางที่ 3 ตัวอย่าง Query

Q1	Patients/Patient/lastname
Q2	Patients/Patient/case_history/infection
Q3	Patients/Patient[lastname= ‘Smith’]/case_history/address
Q4	Patients/Patient[lastname= ‘Smith’]/case_history/infection
Q5	Patients/Patient/case_history[infection=’Haemophilus’] /address
Q6	Patients/Patient/case_history[infection=’Haemophilus’] /medicine

ในขั้นตอนการเข้าถึงข้อมูลในรูปที่ 7 ขั้นตอนที่ 2-4 จะมีรายละเอียดแตกต่างกันไปขึ้นอยู่ลักษณะของ Query กล่าวคือ

กรณีที่ Query เป็นแบบ UQ เช่น Q1 และ Q2 ในตารางที่ 3 โดยที่ Q1 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q2 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส

<u>ขั้นตอนที่ 2</u> เตรียมค่า Path_Name

<u>ขั้นตอนที่ 3</u> ส่งค่า Path_Name ไปยังผู้ให้บริการ

<u>ขั้นตอนที่ 4</u> ค้นหาข้อมูลโดยใช้ค่า Path_Name

กรณีที่ Query เป็นแบบ CQ เช่น Q3-Q6 ในตารางที่ 3

- *เงื่อนไขไม่ถูกเข้ารหัส* เช่น Q3-Q4 โดยที่ Q3 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q4 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส

<u>ขั้นตอนที่ 2</u> เตรียมค่า Path_Name จากตาราง Path_Client และ ค่าข้อมูลของเงื่อนไข (ค่า Node_Data)

<u>ขั้นตอนที่ 3</u> ส่งค่า Path_Name พร้อมกับค่าข้อมูลของเงื่อนไข

<u>ขั้นตอนที่ 4</u> นำค่า Path_Name ไปค้นหา Node_Id จากนั้นใช้ Algorithm ในรูปที่ 8 ทำการค้นหาข้อมูลในตาราง Value

- *เงื่อนไขถูกเข้ารหัส* เช่น Q5-Q6 โดยที่ Q5 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่ไม่เข้ารหัส และ Q6 เป็นคำสั่งสอบถามที่ต้องการผลลัพธ์เป็นค่าข้อมูลที่เข้ารหัส

<u>ขั้นตอนที่ 2</u> เตรียมค่า Path_Name จากตาราง Path_Client และ Hash(ค่า Node_Data)

<u>ขั้นตอนที่ 3</u> ส่งค่า Path_Name พร้อมกับ ผลลัพธ์ของค่า Hash(ค่า Node_Data)

<u>ขั้นตอนที่ 4</u> นำค่า Path_Name ไปค้นหา Node_Id จากนั้นทำการคำนวณหาค่า Hash_Value โดยใช้สมการ 1 (นำแต่ละค่าของ Node_Id ไปรวมกับ ผลลัพธ์ของค่าHash(ค่า Node_Data) แล้วผ่านฟังก์ชัน Hash) และทำการค้นหาข้อมูลในตาราง Value โดยใช้ Algorithm ในรูปที่ 8

```
1. Translate  CQ to CQ’ and UQ’
// CQ’ is only condition part of CQ
// UQ’ is CQ without condition
 2. If  Parent(CQ’) = Parent(UQ’) then
      use CQ’ and UQ’ to make a query from Value Table
```

รูปที่ 8 Algorithm การค้นหาข้อมูลในตาราง Value

ตัวอย่างผู้รับบริการสร้าง Query: Patients/Patient/case_history[infection="Haemophilus"]/medicine และทำการเตรียมข้อมูลโดยการคำนวณค่า Hash("Haemophilus") ซึ่งมีค่าเท่ากับ 741258963 จากนั้น

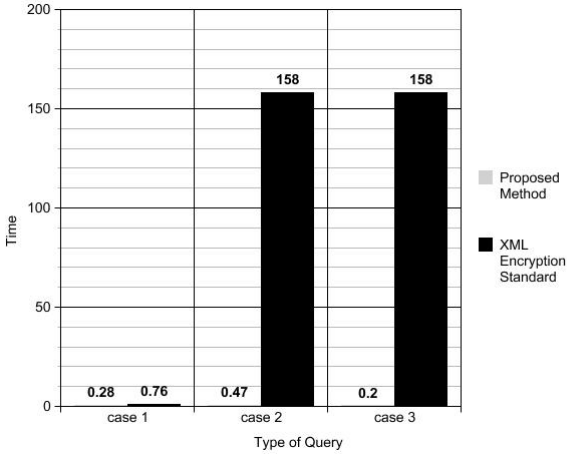
ส่งข้อมูล Path_Name พร้อม ค่า 741258963 ไปยังผู้ให้บริการ นั่นคือ
(Patients/Patient/case_history[infection=741258963]/medicine)
ผู้ให้บริการรับข้อมูลและนำค่า Path_Name ทำการค้นหา Node_Id และทำการคำนวณหาค่า Hash_Value จะได้ว่า Hash_Value มีค่าเท่ากับ 142357869 (Hash("1.1.3.2"+ 741258963) หลังจากนั้นจะค้นหาข้อมูลโดยใช้ Algorithm ในรูปที่ 8 จะได้ว่าผลลัพธ์ที่ได้คือ record ที่ 5 ในตาราง Value ดังนั้นผู้ให้บริการจะส่งค่า "&*()+", "1.1.3.3" และ Name_key1 กลับไปยังผู้รับบริการ หลังจากรับข้อมูลที่ได้จาก Query ผู้รับบริการจะทำการถอดรหัสข้อมูลและลบ Node_Id โดยใช้ข้อมูลกุญแจที่จัดเก็บไว้ในฝั่งของตนเอง ผลลัพธ์ที่ได้คือ Floxin (จากสมการที่ 3 จะได้ว่า Node_Id + Node_Data = Decrytion(&*()_ , Key_Name1) เมื่อถอดรหัสข้อมูลเสร็จสิ้นผลลัพธ์ที่ได้คือ 1.1.3.3Floxin ซึ่งจะต้องลบ Node_Id ออกจากข้อมูลก่อนที่จะนำไปแสดงผล)

5. ผลการเปรียบเทียบประสิทธิภาพ

ในส่วนนี้เป็นการวัดประสิทธิภาพในการทำงานระหว่างวิธีการที่ได้นำเสนอและวิธีการเข้ารหัสของ W3C หรือ XML Encryption Standard โดยใช้ชุดข้อมูล XMark[17] ซึ่งเป็นชุดข้อมูลที่นำมาใช้ในการเปรียบเทียบและเป็นชุดข้อมูลที่มีข้อมูลบางส่วนที่ซ้ำกัน โดยเครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาและทดสอบระบบมีหน่วยประมวลผลกลาง 3 GHz และหน่วยความจำหลักขนาด 2 GB บนระบบปฏิบัติการ WindowXP

เมื่อเปรียบเทียบจำนวนกุญแจที่ใช้ในการเข้ารหัสข้อมูลเพื่อให้ได้ผลลัพธ์ของการเข้ารหัสที่แตกต่างกัน พบว่าวิธีการที่ได้นำเสนอในงานวิจัยนี้จะใช้กุญแจเพียงกุญแจเดียว แต่วิธีการของ W3C จะใช้กุญแจจำนวน 626 กุญแจในการเข้ารหัสข้อมูล

นอกจากนี้ผลการทดลองในด้านเวลาในการค้นหาข้อมูลพบว่าวิธีการที่ได้นำเสนอในงานวิจัยนี้ใช้เวลาในการค้นหาน้อยกว่าวิธีการเข้ารหัสของ W3C



รูปที่ 9 ผลการทดลองเปรียบเทียบเวลาในการค้นหา

โดยที่ในการทดลองจะแบ่งออกเป็น 3 กรณี คือ เปรียบเทียบ 1) กรณี UQ โดยผลลัพธ์ไม่มีการเข้ารหัส 2) กรณี UQ โดยผลลัพธ์เป็นข้อมูลที่เข้ารหัส และ 3) กรณี CQ จากรูปที่ 9 แสดงผลการทดลองดังนี้ ผลการทดลองในกรณี 1) พบว่าการค้นหาของวิธีการที่ได้นำเสนอใช้เวลา 0.281 วินาทีและวิธีการของ W3C ใช้เวลา 0.765 วินาที กรณี 2) พบว่าการค้นหาของวิธีการที่ได้นำเสนอใช้ 0.468 วินาที และวิธีการของ W3C ใช้เวลา 158.185 วินาที กรณี 3) พบว่าการค้นหาของวิธีการที่ได้นำเสนอใช้ 0.203วินาที และวิธีการของ W3C ใช้เวลา 158.34 วินาที หมายเหตุในสองกรณีหลังวิธีการของ W3C ไม่สามารถค้นหาข้อมูลได้โดยตรงจำเป็นที่จะต้องมีการถอดรหัสข้อมูลก่อนดำเนินการค้นหา

6. บทสรุป

ในงานวิจัยนี้ได้มุ่งเน้นที่จะนำเสนอวิธีการในการค้นหาเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ โดยที่ข้อมูลที่เหมือนกันจะถูกเข้ารหัสให้แตกต่างกันเพื่อเพิ่มความปลอดภัย และมีประสิทธิภาพในการจัดเก็บ ซึ่งจากผลการทดลองพบว่าจำนวนกุญแจที่ใช้มี

จำนวนน้อยกว่า และเวลาที่ใช้ในการค้นหาข้อมูลน้อยกว่าวิธีการ XML Encryption Standard

เอกสารอ้างอิง

[1] H. Hacigumus, S. Mehrotra, and B. Iyer. 2002. Providing Database as a Service. Proceedings of the 18th international Conference on Data Engineering. 26 February – 1 March 2002, pp. 29-40.

[2] W3C. Extensible Markup Language (XML). 2009. http://www.w3.org/XML/ (accessed 18/5/2009).

[3] L. Feng and W. Jonker. 2003. Efficient Processing of Secured XML Metadata. In: On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, 3-7 November 2003, Catania, Sicily, Italy. pp. 704-717.

[4] M. Schrefl, K. Grün and J. Dorn. 2005. SemCrypt– Ensuring privacy of electronic documents through semantic-based encrypted query processing, PDM 2005 International Workshop on Privacy Data Management, Workshop Dates Tokyo, Japan, April 8-9, 2005. p.1191.

[5] J. G. Lee and K. Y. Whang. 2006. Secure query processing against encrypted XML data using query-aware decryption, Inform. Sci. 176 (13) (2006), pp. 1928–1947.

[6] R. C. Jammalamadaka and S. Mehrotra. 2006. Querying Encrypted XML Documents. Proceedings of the 10th International Database Engineering and Applications Symposium. Berkeley, CA, USA, December 11-14, 2006. p.129-136.

[7] H. Wang and L.V.S. Lakshmanan. 2006. Efficient secure query evaluation over encrypted XML databases. Proceedings of the 32nd international conference on Very large data bases. Seoul, Korea, September 12-15, 2006. p.127–138.

[8] Y. Yang, W. Ng, H.L. Lau and J. Cheng. 2006. An Efficient Approach to Support Querying Secure Outsourced XML Information. Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006. Luxembourg, Luxembourg, June 5-9, 2006. p.157-171.

[9] A. Algergawy, E. Schallehna and G. Saakea. 2009. Improving XML schema matching performance using Prüfer sequences. Data & Knowledge Engineering Volume 68, Issue 8, August 2009, p. 728-747.

[10] W3C Recommendation XML Path Language (XPath) 1.0, Available from: http://www.w3c.org/ TR/XPath, 1999.

[11] T. Imamura, B. Dillaway and E. Simon. 2002. XML Encryption Syntax and Processing. W3C Recommendation.

[12] J. Gao, T. Wang and D. Yang. 2008. XFlat: Query-friendly encrypted XML view publishing. Information Sciences 178(3): p.774-787.

[13] L.M. Chan 1996. Dewey Decimal Classification: A Practical Guide, second ed., OCLC Forest Press.

[14] L. Yue, J. Ren, Y. Qian. 2009. Storage Method of XML Documents Based-on the Pri-order Labeling Scheme. etcs, vol. 2, pp.127-131, 2009 First International Workshop on Education Technology and Computer Science.

[15] J. Min, C. Lee and C. Chung. 2008. XTRON: An XML data management system using relational databases. Information and Software Technology 50 (2008): p.462–479.

[16] M.V. Ramakrishna, J. Zobel. 1997. Performance in Practice of String Hashing Functions. Proceedings of the Fifth International Conference on Database Systems for Advanced Application, Melbourne, Australia, April 1-4, 1997.

[17] A. R. Schmidt, F. Waas, M. L. Kersten, D. Florescu, I. Manolescu, M. J. Carey and R. Busse. 2001. The XML Benchmark Project. Technical Report INS-R0103, CWI, Amsterdam, The Netherlands.

ประวัติผู้เขียน

ชื่อ สกุล นายด่วนกัสฟี หะมะ

รหัสประจำตัวนักศึกษา 5110220027

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ. (วิทยาการคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2550

การตีพิมพ์เผยแพร่ผลงาน

ด่วนกัสฟี หะมะ, ศิริรัตน์ วณิชโยบล และ ลัดดา ปรีชาวีรกุล. 2553. การสอบถามเอกสาร XML ที่ถูกเข้ารหัสซึ่งจัดเก็บไว้ในฝั่งของผู้ให้บริการ. The 7th International Joint Conference on Computer Science and Software Engineering (JCSSE 2010). กรุงเทพมหานคร ประเทศไทย. 12 – 14 พฤษภาคม 2553. หน้า 241-249.