ความปลอดภัยของบิ๊กดาตา

BIG DATA SECURITY

เจต โกมลวนิช

เจษฎา รัตนจรัสกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ความปลอดภัยของบิ๊กดาตา

BIG DATA SECURITY

ผู้จัดทำ

1. นายเจต	โกมลวนิช	รหัสนักศึกษา	54010233
2. นายเจษฎา	รัตนจรัสกุล	รหัสนักศึกษา	54010244

..... อาจารย์ที่ปรึกษา

(อาจารย์อัครเดช วัชระภูพงษ์)

ความปลอดภัยของบิ๊กดาตา

นาย เจต	โกมลวนิช	54010233
นาย เจษฎา	รัตนจรัสกุล	54010244
อ.อัครเดช	วัชระภูพงษ์	อาจารย์ที่ปรึกษา

ปีการศึกษา 2557

บทคัดย่อ

ในปัจจุบันข้อมูลต่างๆ ในระบบสารสนเทศมีขนาดและจำนวนเพิ่มมากขึ้นอย่างรวดเร็ว จึงมีความจำเป็นในการนำเทคโนโลยี และอุปกรณ์ที่ทันสมัยมาปรับใช้ในการบริหารจัดการข้อมูล กระบวนการบริหารจัดการบิ๊กดาต้าจึงเกิดขึ้น สิ่งที่มีความสำคัญไม่น้อยไปกว่ากระบวนการบริหารจัดการคือการรักษาความปลอดภัยของข้อมูลบิ๊กดาตา จากปัญหา และการเปลี่ยนแปลงต่างๆ ที่เกิดขึ้น จึงเป็นที่มาและความสำคัญของโครงงาน “ความปลอดภัยของบิ๊กดาตา(Big Data Security)” เพื่อเป็นการศึกษาปัญหา และเพิ่มประสิทธิภาพการรักษาความมั่นคงปลอดภัยข้อมูล โครงงานนี้ให้ความสำคัญไปที่การรักษาความถูกต้อง และครบถ้วนของข้อมูล (Data Integrity) คณะผู้จัดทำจึงศึกษา และค้นหาข้อมูลทั้งที่เป็นเอกสาร หนังสือ หรือบทความทางวิชาการต่างๆ จากหลากหลายแหล่งข้อมูล เพื่อทำการรวบรวม และสังเคราะห์เพื่อหากระบวนการที่เหมาะสม เรียบเรียงกระบวนการดังกล่าว พบว่า “การสร้างลายน้ำของข้อมูล (Watermarking Algorithm)” ซึ่งเป็นกระบวนการหนึ่งในการตรวจสอบความถูกต้องครบถ้วนของข้อมูล การสร้างลายน้ำของข้อมูลมีความเหมาะสมกับกระบวนการบริหารจัดการข้อมูลบิ๊กดาตามากกว่าการสร้างลายเซ็นดิจิตัล(Digital Signature Algorithm) ซึ่งเป็นที่นิยม และมีการใช้กันอย่างกว้างขวาง ข้อดีของการสร้างลายน้ำของข้อมูลคือไม่ใช้กุญแจ ใช้ทรัพยากรน้อย กระบวนการไม่ซับซ้อนมาก ทำได้รวดเร็ว สามารถทำย้อนกลับได้ง่าย และที่สำคัญคือมีประสิทธิภาพมากในการรักษาความถูกต้องครบถ้วนของข้อมูล จึงนับว่าการสร้างลายน้ำของข้อมูลเป็นกระบวนการที่ใช้ในการรักษาความถูกต้องครบถ้วนของข้อมูลที่เหมาะสมกับกระบวนการบริหารจัดการข้อมูลบิ๊กดาตา

BIG DATA SECURITY

Mr. Jate	Komolvanich	54010233
Mr. Jetsada	Rattanacharatkun	54010244
Mr. Akkradach	Watcharapupong	Advisor

Academic Year 2014

ABSTRACT

Nowadays, Data of information technology systems, always grow up and up. The new hardwares and high technologies, are applied for data management system. Now, "Big data" becomes famous in the world. Not only data management, but also information security is very important too. Because of many problems and changes, that why my project works on "Big Data Security". We studied its problems and found any ways to improve the information security. We focused on big data integrity checking. We studied many educational articles from many resources and synthesized. Then, we found "The Watermarking Algorithm (Watermarking Embedding Processes)", the one of data integrity checking. The watermarking algorithm is more appropriate with big data than the digital signature algorithm, the well-known and popular algorithm for data integrity checking. The watermarking algorithm advantages are no key, lightweight, easy algorithm, faster and very efficient to ensure big data integrity. Conclusion, the watermarking is appropriate with big data security.

กิตติกรรมประกาศ

โครงงาน และปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ เนื่องจากคำแนะนำ และการให้คำปรึกษาที่ดีจากท่านอาจารย์ที่ปรึกษาคือ อาจารย์อัครเดช วัชระภูพงษ์ ที่คอยแนะนำ เป็นที่ปรึกษา และเอาใจใส่กับการทำโครงงานเป็นอย่างดี ซึ่งทางคณะผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาเป็นอย่างสูง

ขอขอบพระคุณคณาจารย์ทุกๆ ท่าน ในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นอย่างยิ่งที่ได้ช่วยประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ อีกทั้งภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้เอื้อเฟื้ออุปกรณ์ ทรัพยากร สถานที่ และอำนวยความสะดวกต่างๆ ในโครงงานนี้ด้วย

ขอขอบพระคุณเพื่อนๆ และน้องๆ สมาชิกห้องวิจัยและพัฒนาการรักษาความปลอดภัยข้อมูล (Information Security Advisory Group) และคนอื่นๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งได้ช่วยเหลือในการทำงาน และแก้ไขปัญหา อุปสรรคต่างๆ ให้ผ่านพ้นไปได้ด้วยดี

สุดท้ายนี้ต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดอันได้แก่ บิดา และมารดาของคณะผู้จัดทำที่เคารพและเป็นที่รักยิ่ง ผู้ที่ให้กำเนิด สั่งสอน และให้การศึกษาอย่างสูงสุด พร้อมทั้งสนับสนุนสิ่งต่างๆ ตลอดการทำโครงงาน และปริญญานิพนธ์นี้ นับเป็นพระคุณอย่างสูงสุดหาที่เปรียบมิได้ คณะผู้จัดทำขอระลึกพระคุณอันยิ่งใหญ่สุดประมาณนี้ไว้กว่าชีวิตจะหาไม่ และขอกราบขอบพระคุณ ทุกที่กล่าวถึงและ ไม่ได้กล่าวถึงไว้ ณ ที่นี้ด้วย

นาย เจต โกมลวนิช

นาย เจษฎา รัตนจรัสกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ในยุคปัจจุบันเทคโนโลยีมีความก้าวหน้าเป็นอย่างมาก ทำให้ข้อมูล และสารสนเทศต่างๆ จำเป็นต้องทำการเปลี่ยนข้อมูลต่างๆ จากเอกสาร มาเป็นรูปแบบดิจิตัล เพื่อให้สะดวกและง่ายต่อการบริหารจัดการด้วยเทคโนโลยีสารสนเทศอันทันสมัย ประกอบกับกฎหมายที่มีความเกี่ยวข้องกับการกระทำความผิดทางคอมพิวเตอร์ประกาศใช้ออกมาอย่างต่อเนื่อง อาทิเช่นประกาศคณะกรรมการธุรกรรมทางอิเล็คทรอนิกส์ เรื่อง มาตรฐานการรักษาความมั่นคงปลอดภัยของระบบสารสนเทศตามวิธีแบบปลอดภัย ที่มีการกำหนดให้บริษัทหรือองค์กร ต้องมีการเก็บ Log ทั้งของผู้ใช้งาน และผู้ดูแลระบบ ทำให้ทุกๆ องค์กรมีการตื่นตัวตอบรับกับกฎหมาย หรือการเก็บข้อมูลต่างๆ ของธุรกิจที่มีความจำเป็นต้องใช้ในการวางแผนต่างๆ ทั้งด้านกลยุทธ์ และการตลาดของธุรกิจ แต่ละองค์กรทั้งของภาครัฐ และเอกชน จึงมีข้อมูลต่างๆ เพิ่มเข้ามาในระบบสารสนเทศมากขึ้น การบริหารจัดการข้อมูลสารสนเทศที่เกิดขึ้นนั้นมีความแตกต่างไปจากวิธีการเดิมที่ใช้กับข้อมูลจำนวนน้อย ซึ่งการบริหารจัดการข้อมูลสารสนเทศตามปกติ ไปจึงไม่เหมาะสมกับการบริหารจัดการข้อมูลสารสนเทศต่างๆ ที่กำลังจะเพิ่มขึ้นในอนาคต จึงเป็นสาเหตุให้มีการค้นคว้า และพัฒนากระบวนการต่างๆ ในการนำเข้าประมวลผล จัดเก็บ และแสดงข้อมูล ให้มีประสิทธิภาพเพิ่มมากขึ้นทั้งมีสมรรถนะในการประมวลผลที่ดี และข้อมูลมีความถูกต้อง และอีกสิ่งหนึ่งที่มีความสำคัญนั่นคือความมั่นคงปลอดภัยของข้อมูล เพราะข้อมูลสารสนเทศของบริษัท หรือองค์กรต่างๆ นั้นต้องเป็นความลับ มีความถูกต้องครบถ้วน และรักษาสภาพพร้อมใช้งานตามหลักของการรักษาความมั่นคงปลอดภัยของข้อมูลสารสนเทศ ซึ่งการรักษาความปลอดภัยข้อมูลจึงเป็นอีกกระบวนการหนึ่งที่มีความสำคัญ และจำเป็นต้องมีการพัฒนาไปพร้อมกันกับประสิทธิภาพของกระบวนการอื่นๆ ในการบริหารจัดการข้อมูลดังกล่าวในข้างต้น จึงเป็นที่มาและความสำคัญของโครงงาน “ความปลอดภัยของบิ๊กดาตา(Big Data Security)”

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาปัญหาของการรักษาความมั่นคงปลอดภัยให้กับกระบวนการบริหารจัดการบิ๊กดาตา

2) เพื่อศึกษาการเพิ่มประสิทธิภาพการรักษาความมั่นคงปลอดภัยให้กับกระบวนการบริหารจัดการบิ๊กดาตา

1.3 ขอบเขตของโครงงาน

1) ศึกษาปัญหาและสาเหตุที่เกิดขึ้นจากกระบวนการบริหารจัดการบิ๊กดาตา โดยการเริ่มต้นในปัญหาที่เกี่ยวข้องกับความถูกต้องครบถ้วนของข้อมูล

2) โครงงานนี้มุ่งเน้นไปที่การศึกษา Streaming data integrity checking

3)เป็นการศึกษา และทดสอบการทำงานของอัลกอริทึม ไม่ใช่การสร้างโปรแกรม หรือแอพพลิเคชั่นที่สามารถทำงานได้ทันที

4) ระบบมีความต้องการบางอย่างที่ต้องมีการตั้งค่าเพิ่มเติมจากในคู่มือการติดตั้งและการใช้งาน

5) ระบบมีความจำเป็นต้องใช้โปรแกรม VMware fusion เพราะฉะนั้นควรติดตั้งโปรแกรมก่อนการใช้งานระบบ

1.4 วิธีการดำเนินการ

1) ทำการศึกษาทฤษฎีพื้นฐาน และทำความเข้าใจบิ๊กดาตา

2) ศึกษาทฤษฎีพื้นฐาน และค้นหาข้อมูลที่เกี่ยวกับความปลอดภัยของบิ๊กดาตา

3) ศึกษาข้อมูลที่ได้จากบทความของ CSA (ฉบับแรก) และ CSA (ฉบับขยายความ)

4) ทำการวิเคราะห์ปัญหาที่ได้มาจากบทความ

5) ศึกษา Kylin OLAP by eBay Inc., NoSQL และ Non-Relational Database

6) ค้นหาข้อมูล และศึกษาเทคโนโลยี Keyless

7) ออกแบบ และจัดทำ Poster สำหรับการนำเสนอแนวคิดของโครงงาน

8) ศึกษา Keyless Algorithms ต่างๆ เพิ่มเติม

9) ทดลองปรับใช้ Keyless กับ Big Data Integrity

10) ค้นหาข้อมูล และศึกษาเทคโนโลยีที่เกี่ยวข้องกับ Wireless Sensor Network

11) ทำการศึกษาเทคโนโลยีที่เกี่ยวข้องกับการทำ Watermarking Algorithm ในรูปแบบต่างๆ

12) เขียนโปรแกรมเพื่อสร้างความเข้าใจ และสร้าง Watermarking Algorithm เพื่อทดลองกับ Streaming data

13) จำลองการทดลองเพื่อทดสอบประสิทธิภาพในรูปแบบต่างๆ ของ Watermarking Algorithm และทำการสรุปผลการศึกษา

14) รวบรวมข้อมูลเพื่อจัดทำรูปเล่ม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) รับรู้ รับทราบ และเข้าใจถึงกระบวนการบริหารจัดการบิ๊กดาตา ซึ่งเป็นการรวมเอาเทคโนโลยีต่างๆ มาประยุกต์ใช้รวมกัน

2) รับรู้ รับทราบ และเข้าใจในแนวคิด ความเป็นมา และความสำคัญของวิวัฒนาการ และการพัฒนาในการบริหารจัดการบิ๊กดาตา

3) รับรู้ รับทราบ และเข้าใจถึงปัญหาด้านการรักษาความมั่นคงปลอดภัยที่เกิดขึ้นในกระบวนการบริหารจัดการบิ๊กดาตาในระดับสากล

4) ศึกษา และพัฒนากลไก หรือกระบวนการที่เป็นไปได้ในการรักษาความมั่นคงปลอดภัยในกระบวนการบริหารจัดการบิ๊กดาตา

5) เพิ่มประสิทธิภาพ และความน่าเชื่อถือให้กับกระบวนการรักษาความมั่นคงปลอดภัยของบิ๊กดาตา

1.6 ส่วนประกอบของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาโดยทั่วไปออกเป็น 5 บท คือ บทนำ ทฤษฎีที่เกี่ยวข้อง การออกแบบและพัฒนา การทดลองและผลการทดลอง และบทสรุป โดยสามารถจำแนกรายละเอียดได้ดังนี้

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงงาน การให้ความหมาย และคำจำกัดความของบิ๊กดาตาที่มีการอ้างอิงในโครงงาน หลักการของเรื่องต่างๆ รวมถึงซอฟต์แวร์ และภาษาต่างๆ ที่นำมาใช้ในโครงงานนี้ เช่น Python และผลิตภัณฑ์ของ Apache ต่างๆ เป็นต้น

บทที่ 3 การออกแบบและพัฒนา กล่าวถึงขั้นตอนในการศึกษา ลำดับของการใช้คำสำคัญต่างๆ เพื่อใช้ค้นหาข้อมูล รวมไปถึงแนวคิด และขั้นตอนต่างๆ ในการพัฒนาโปรแกรมเพื่อใช้ในการทดลอง และทำความเข้าใจอัลกอริทึม

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง วิธีการทดลอง และผลลัพธ์ที่ได้จากการทดลองรวมถึงการกระทำต่างๆ ที่สามารถนำไปสู่ข้อสรุปของโครงงานนี้

บทที่ 5 บทสรุปและข้อเสนอแนะ กล่าวถึง ผลที่ได้จากการทำโครงงาน ปัญหา อุปสรรค แนวทางการแก้ไข และแนวทางในการพัฒนาต่อยอด

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บิ๊กดาตา(Big Data)

2.1.1 ทำความรู้จักบิ๊กดาตา (Big Data)

บิ๊กดาตา(Big Data) คืออะไร สิ่งที่จำเป็นต้องกล่าวถึงก่อนที่จะอธิบายคำๆ นี้คงต้องกำหนดความหมายของคำว่า “บิ๊ก(Big)” ซึ่งถ้าแปลเป็นภาษาไทยพจนานุกรมจะให้ความหมายของคำนี้ว่า “ใหญ่” การทำให้ได้มาซึ่งข้อมูลขนาดใหญ่สำหรับระบบสารสนเทศนั้นมีอยู่ 3 ลักษณะด้วยกัน คือ

1). ปริมาตร(Volume)

ข้อมูลนั้นมีจำนวนเท่าไหร่ บิ๊กดาตานั้นจะมีปริมาณของข้อมูลมากมายมหาศาลอยู่ในระดับหลายเทราไบท์ด้วยกันโดยที่ปริมาณของข้อมูลนั้นจะเติบโตขึ้นเรื่อยๆ

2). ความเร็ว(Velocity)

ข้อมูลนั้นมีการนำเข้า และประมวลผลเร็วขนาดไหน บิ๊กดาตานั้นจะมีการเปลี่ยนแปลงของข้อมูลอยู่ตลอดเวลา มีการนำข้อมูลเข้าสู่ระบบสารสนเทศครั้งละมากๆ และรวดเร็ว

3). ความหลากหลาย(Variety)

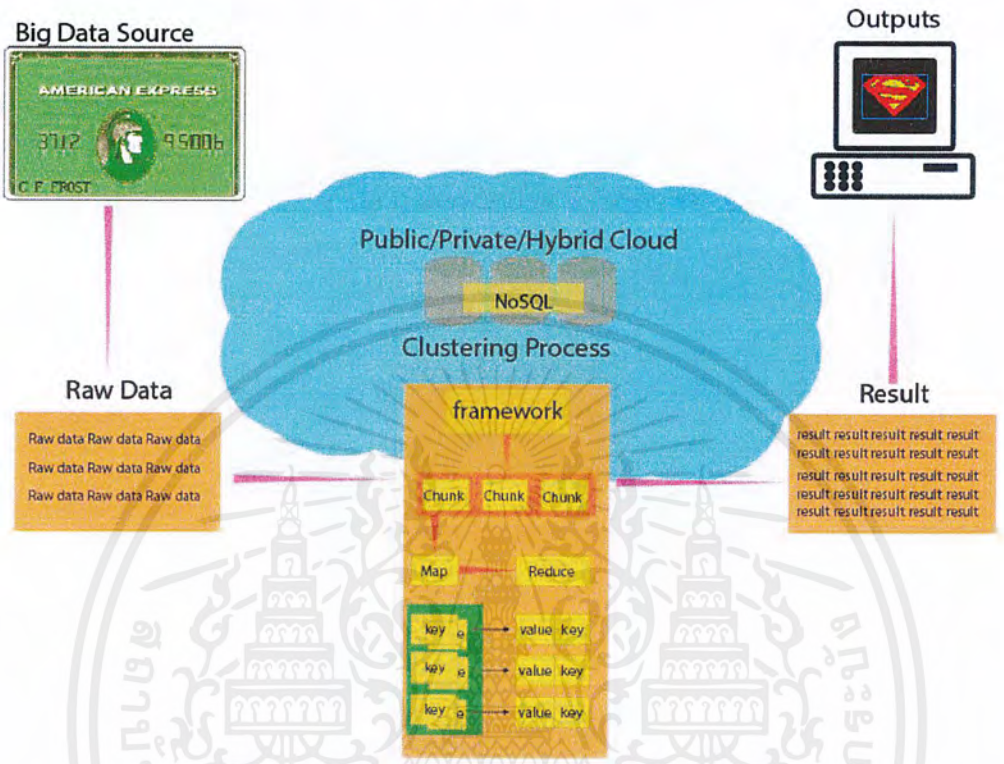
ชนิดของข้อมูลนั้น มีความหลากหลายมากเพียงใด บิ๊กดาตานั้นจะมีรูปแบบของข้อมูลอยู่หลากหลายรูปแบบประกอบเข้าด้วยกัน อาทิเช่นข้อความ รูปภาพ มัลติมีเดียต่างๆ รวมไปถึง Logs ต่างๆ ด้วย

การบริหารจัดการข้อมูลที่มีปริมาตรจำนวนมาก มีความเร็วในการนำเข้ามาก ต้องการประมวลผลเร็วมาก และชนิดของข้อมูลนั้นมีความหลากหลายมาก จึงมีความแตกต่างไปจากการบริหารจัดการข้อมูลแบบเดิมๆ ดังนั้นการบริหารจัดการนั้นจึงประกอบไปด้วยหลากหลายเทคโนโลยี เพื่อทำให้ระบบสามารถรองรับการบริหารดังที่กล่าวมาในข้างต้นได้

จึงสามารถกล่าวได้ว่า บิ๊กดาตา คือความสามารถในการบริหารจัดการข้อมูลขนาดใหญ่ ข้อมูลที่มีการนำเข้า และต้องการประมวลผลเร็ว และข้อมูลนั้นมีชนิดที่หลากหลาย ดังนั้นบิ๊กดาตาจึงเป็นกระบวนการที่รวมเอาหลากหลายเทคโนโลยีมาช่วยในการบริหารจัดการข้อมูลต่างๆ ที่จะมีเพิ่มมากขึ้นตามวิทยาการที่ก้าวหน้าในอนาคต

ระบบการบริหารจัดการบิ๊กดาตาจึงประกอบไปด้วยข้อมูลขนาดใหญ่ หรือข้อมูลจำนวนมากมหาศาล ซึ่งข้อมูลจำนวนมหาศาลนั้นจะมีการเปลี่ยนแปลงอยู่ตลอดเวลา ยกตัวอย่างเช่น Transaction Logs ของธนาคาร ข้อมูลการสนทนาบนโซเชียลเนตเวิร์ค(Social Network) หรือยังรวมไปถึงการบันทึกการซื้อสินค้าย้อนหลังของบริษัทด้วย

ในระบบการบริหารจัดการบิ๊กดาตานั้น จะประกอบไปด้วย



รูปที่ 2.1 แผนภาพของระบบบริหารจัดการบิ๊กดาตา

ซึ่งภาพแผนภาพของระบบการบริหารจัดการบิ๊กดาตาทางด้านบนนั้น รูปจะเริ่มต้นที่ Big Data Sources ซึ่งทำหน้าที่เป็นแหล่งข้อมูลที่ส่งข้อมูลเข้าสู่ระบบการบริหารจัดการบิ๊กดาตา ดังยกตัวอย่างในแผนภาพโดยการใช้รูปของบัตรเครดิต(Credit Card) เพราะในระบบบริหารจัดการบิ๊กดาตาในปัจจุบันนั้นจะนิยมใช้ Input sources หลากหลายแหล่งพร้อมๆ กัน จึงมีความเกี่ยวข้องกันกับแนวคิด Bring Your Own Devices (BYOD) ซึ่งจะมีการกล่าวถึงในภายหลัง การใช้บัตรเครดิตนั้นเป็นหนึ่งในตัวอย่างการใช้ระบบการบริหารจัดการบิ๊กดาตาที่ชัดเจน เนื่องมาจากในปัจจุบันเพื่อความสะดวกในการจ่ายเงิน หรือชำระค่าสินค้า และบริการที่เป็นที่นิยมของเทคโนโลยีปัจจุบัน ซึ่งในหนึ่งวันนั้นบัตรเครดิตของแต่ละผู้ให้บริการบัตรเครดิตนั้นจะต้องมีการรับ Transaction จำนวนมาก ซึ่งจากการใช้บัตรเครดิตปัจจุบันจะเห็นได้ว่าข้อมูลจะถูกอัพเดตได้เกือบจะทันที(Real-time) หรือมีความรวดเร็วมาก สามารถดูได้จากการที่เราไปทำการปรับสมุดคู่ฝาก หรือการตรวจสอบ Transaction Logs ของตนเองผ่านระบบบริการลูกค้าของผู้ให้บริการ เช่น Internet Banking หรือทาง Smart Devices ซึ่งจะพบว่าข้อมูลการใช้จ่ายดังกล่าวจะถูกบันทึกลงฐานข้อมูลออนไลน์ของผู้ให้บริการทันที ลำดับต่อมาข้อมูลที่เป็น Raw Data เป็นข้อมูลที่ถูกนำเข้ามาจาก Input Sources ต่างๆ จะถูกลงเข้า

มาที่ระบบประมวลผล ซึ่งระบบประมวลผลที่นิยมในระบบบริหารจัดการบิ๊กดาตานั้น ในปัจจุบันจะมีอยู่สองประเภทที่จะกล่าวถึงรายละเอียดในภายหลัง แต่ในแผนภาพทางด้านบนจะอ้างอิงระบบที่เป็นการประมวลผลด้วยเทคโนโลยี Clustering Processing เช่นเทคโนโลยี Hadoop ที่เป็น Distributed Computing พัฒนาโดย Apache ซึ่งจะกล่าวถึงในรายละเอียดในส่วนของ Apache Hadoop ซึ่งเทคโนโลยีที่นิยมใช้ทำงานร่วมกันกับระบบประมวลผลนั้น ในปัจจุบันเทคโนโลยีที่นิยมใช้ในระบบบริหารจัดการบิ๊กดาตานั้น คือเทคโนโลยี Public/ Private/ Hybrid Cloud Computing ซึ่งผลลัพธ์ที่ได้จากการประมวลผลออกมาเป็น Results นั้น ขึ้นอยู่กับเทคโนโลยีที่ใช้ประมวลผลว่าผลลัพธ์ที่ได้ออกมานั้นจะทำให้ข้อมูลนั้นมีปริมาณมากขึ้น หรือลดลง แต่จะทำให้สะดวกและง่ายในการจัดเก็บลงส่วนบันทึกข้อมูลและการเข้าถึง เพื่อนำออกไปสู่ในส่วนแสดงผลของระบบ

จากที่กล่าวมาในข้างต้นคือการอธิบายกระบวนการ และเทคโนโลยีที่ใช้ในระบบการบริหารจัดการบิ๊กดาตาอย่างคราวๆ เพื่อให้เกิดความเข้าใจในความสำคัญของการเปลี่ยนแปลงจากระบบทั่วๆ ไป และการทำงานคร่าวๆ พร้อมยกตัวอย่างที่เกี่ยวข้องในกระบวนการบริหารจัดการบิ๊กดาตา

2.1.2 ส่วนประกอบต่างๆ ของกระบวนการบริหารจัดการบิ๊กดาตา

2.1.2.1 ส่วนนำข้อมูลเข้าสู่ระบบ(Input Source)

ส่วนนำข้อมูลเข้าสู่ระบบการบริหารจัดการบิ๊กดาตา(Big Data Sources) เป็นส่วนที่ทำหน้านำข้อมูลดิบ(Raw Data) เข้ามาสู่ระบบ ซึ่งอุปกรณ์ที่ทำหน้าที่เป็น Big Data Sources นั้น เป็นอุปกรณ์ที่ทำหน้าสร้างหรือรับข้อมูล และนำส่งเข้ามาสู่แหล่งข้อมูลเข้าส่วนกลาง(Centralization) โดยทั่วๆ ไปแล้วสำหรับอุปกรณ์ที่ทำหน้าที่เป็นส่วนนำข้อมูลเข้าสู่ระบบนั้น จะเป็นอุปกรณ์หลายๆ ชิ้นที่ส่งข้อมูลเข้ามาสู่ส่วนกลางของระบบเดียวกัน ซึ่งจะเป็นอุปกรณ์ของส่วนกลางผลิต และแจกจ่ายออกมาเอง อาทิเช่น การใช้งานบัตรอิเล็คทรอนิกส์ประเภท Debit Card และ Credit Card ดังตัวอย่างข้างต้น บัตรที่สามารถใช้งานได้อย่างถูกต้องคือต้องมีการลงทะเบียนเพื่อรับบัตรจากธนาคารหรือผู้ให้บริการบัตรแต่ละแห่งนั้นเป็นผู้ออกบัตรให้ ถึงแม้ในทางปฏิบัติจะสามารถทำบัตรขึ้นมาใช้เองได้ แต่บัตรนั้นก็จะไม่ได้รับการยอมรับจากคนในสังคม ซึ่งบัตรอิเล็คทรอนิกส์ที่ออกมาจากผู้ให้บริการแต่ละแห่ง เมื่อทำการส่งข้อมูล เบื้องหลังอาจจะประกอบไปด้วย Transactions มากกว่า 1 Transactions แต่สิ่งที่ผู้ใช้บริการจะรับรู้คือข้อมูลการใช้งาน ปลายทางต้องถูกส่งไปที่ผู้ให้บริการที่เป็นคนออกบัตรให้เพียงเท่านั้น หรืออีกตัวอย่างหนึ่งที่ชัดเจนคือการใช้อุปกรณ์ติดตามตัว หรือการติดแท็กให้กับพะยูนของกรมอุทยานแห่งชาติ สัตว์ป่า และพันธุ์พืช เพื่อใช้ในการศึกษาพฤติกรรมการดำรงชีวิต และตำแหน่งที่อยู่ของพะยูนในท้องทะเลอ่าวไทย ซึ่งอุปกรณ์ดังกล่าวจะต้องมีการส่งข้อมูลพิกัดเพื่อบอกตำแหน่งของพะยูนที่มีอุปกรณ์ติดอยู่ โดยเป้าหมายของการศึกษาคือการค้นหาเส้นทางหรืออาณาบริเวณที่ใช้ในการดำรงชีวิตตลอดทั้งวันของพะยูน สิ่งที่เกิดขึ้นคืออุปกรณ์ต้องทำการส่ง

ข้อมูลตลอดทั้งวันเพื่อให้ได้ข้อมูลที่มีความชัดเจนมากที่สุดเพื่อใช้ในการศึกษา อุปกรณ์ดังกล่าวต้องเป็นอุปกรณ์ที่ถูกสร้างขึ้นโดยกรมอุทยานแห่งชาติ สัตว์ป่า และพันธุ์พืชเพียงเท่านั้น เพื่อส่งข้อมูลไปยังเครื่องแม่ข่ายส่วนกลางที่ได้มีการกำหนดเอาไว้อย่างถูกต้อง

หรืออีกหนึ่งรูปแบบคือนำอุปกรณ์หลายๆ เครื่องที่เป็นของผู้ให้บริการเอง และอาจจะเป็นเครื่องส่วนบุคคลที่ต้องนำมาลงทะเบียน สร้างบัญชีผู้ใช้ เพื่อติดต่อกับเครื่องแม่ข่ายของส่วนกลาง ในแนวคิดของการนำอุปกรณ์ส่วนตัวมาใช้หรือเรียกว่า “Bring Your Own Device” (BYOD) ตัวอย่างที่ชัดเจน เช่น การใช้ Social Networking ต่างๆ เช่น Messaging, Posting และ Sharing เป็นต้น ผู้ให้บริการไม่จำเป็นต้องแจกจ่ายอุปกรณ์ เพื่อให้นำมาใช้ในการติดต่อกับเครื่องแม่ข่าย แต่สามารถนำเครื่องใดๆ ก็ได้ นำมาใช้ติดต่อกับเครื่องแม่ข่ายโดยการนำมาลงทะเบียน เพื่อสร้างบัญชีที่ใช้ติดต่อกับเครื่องแม่ข่าย จากนั้นจึงใช้บัญชีนี้ในการติดต่อกับเครื่องแม่ข่ายผ่านทางอุปกรณ์ใดๆ ก็ได้

การใช้อุปกรณ์ต่างๆ ที่นำมาเป็นแหล่งนำข้อมูลเข้าสู่ระบบของกระบวนการบริหารจัดการบิ๊กดาตานั้น ไม่ว่าจะเป็นอุปกรณ์ที่ส่วนกลางเป็นผู้ผลิต และแจกจ่ายเอง หรือเป็นอุปกรณ์ที่ใช้เป็นอุปกรณ์ส่วนบุคคลที่นำมาเชื่อมต่อกับเครื่องแม่ข่าย สิ่งที่สำคัญคือเครื่องแม่ข่ายจะทำวิธีการใด เพื่อทำให้มั่นใจได้ว่า เครื่องที่เป็นแหล่งนำข้อมูลเข้านั้นเป็นเครื่องของผู้ให้บริการจริง หรือเป็นเครื่องส่วนบุคคลที่นำมาลงทะเบียนกับเครื่องแม่ข่ายที่ถูกต้องแล้วจริง ซึ่งจะอธิบายถึงรายละเอียดของประเด็นปัญหาดังกล่าวในหัวข้ออื่นถัดๆ ไป

2.1.2.2 ส่วนประมวลผล

สำหรับส่วนประมวลผลนั้น โดยในที่นี้จะกล่าวถึงส่วนประมวลผลเป็นหลัก เพราะส่วนประมวลผลเปรียบเสมือนส่วนสำคัญที่ทำให้กระบวนการบริหารจัดการบิ๊กดาตานั้นสามารถบริหารจัดการข้อมูลที่เข้ามาในระบบได้หรือไม่ เพราะส่วนที่ต่างกันคือการประสิทธิภาพในการประมวลผล ส่วนประมวลผลของกระบวนการบริหารจัดการบิ๊กดาตาในปัจจุบันนั้นมี 2 เทคโนโลยีซึ่งเป็นที่นิยมอยู่ในขณะนี้ คือ In-Memory Computing Technology และ Distributed Computing Technology หรือ Clustering Processing Technology ซึ่งจะอธิบายในรายละเอียดต่อไป

แนวคิดของ In-Memory Computing Technology คือการใช้เครื่องแม่ข่ายที่มีสมรรถนะสูงมาก มีขนาดของ Memory ที่มากเพียงพอที่จะนำข้อมูลเข้าทีละมากๆ และนำมาใช้ในการประมวลผลด้วย Central Processing Unit(CPU) ที่มีความสามารถมาก ทำให้กระบวนการบริหารจัดการบิ๊กดาตานั้นมีประสิทธิภาพมาก ซึ่งการสร้างระบบเพื่อนำกระบวนการบริหารจัดการบิ๊กดาตามาปรับใช้กับการประมวลผลด้วย In-Memory Computing นั้น จำเป็นต้องมีต้นทุนที่ใช้การลงทุนสูงมาก เพื่อให้เครื่องแม่ข่ายนั้นมีประสิทธิภาพ และสมรรถนะที่ดี เพื่อให้ได้ผลลัพธ์ของการประมวลผลตามที่ต้องการ ทั้งในเรื่องของการให้ข้อมูลที่ได้จากการประมวลผลที่ถูกต้อง และความเร็วในการ

ประมวลผลที่ใช้เวลาในการประมวลผลน้อย การใช้กระบวนการบริหารจัดการบิ๊กดาตาที่มีการนำ In-Memory Computing Technology มาประยุกต์ใช้ด้วยนั้น ผู้ที่นำมาใช้จะต้องมีเงินทุนมากพอที่จะสามารถจัดหาเครื่องแม่ข่ายตามที่ In-Memory Computing Technology ซึ่งการใช้กระบวนการบริหารจัดการบิ๊กดาตาด้วยแนวคิดนี้ จะมีการใช้ในเชิงพาณิชย์เพียงเท่านั้น เช่นระบบการประมวลผลด้วย SAP HANA ซึ่งนำมาใช้กับระบบการบริหารจัดการ SAP ซึ่งเป็น Enterprise Resource Planning(ERP) ชื่อดังซึ่งเป็นที่นิยมในวงการธุรกิจปัจจุบัน ซึ่ง SAP HANA นั้นมีการใช้ In-Memory Computing ร่วมกับ Column-Oriented หรือ Relational Database Management System เพื่อให้ง่ายต่อการพัฒนา

สำหรับแนวคิดต่อไปคือการใช้ Distributed Computing Technology หรือเรียกว่า Clustering Processing Technology ซึ่งเป็นเทคโนโลยีที่เป็นที่นิยมมากกว่าในสังคมทั่วๆ ไป เพราะเป็น Open Source ที่สามารถดาวน์โหลดให้ใช้ได้โดยไม่มีค่าใช้จ่าย จึงเป็นที่นิยม และมีใช้มากเพื่อการศึกษาทดลองกระบวนการบริหารจัดการบิ๊กดาตาในรูปแบบต่างๆ แนวคิดของ Distributed Computing Technology คือการแบ่งข้อมูลขนาดใหญ่ออกเป็นส่วนๆ หรือเป็นก้อนย่อยๆ (Clustering) จากนั้นจึงนำไปเข้าสู่การประมวลผลที่ละก้อนโดยกระบวนการคือ เมื่อข้อมูลจำนวนมากผ่านเข้ามา Distributed Computing Framework จะทำหน้าที่ในการแบ่งข้อมูลออกเป็น Cluster(Clustering) อาจจะเรียก Clusters เหล่านั้นเป็นชิ้นๆ (Chunks) จากนั้นจึงขึ้นอยู่กับแต่ละผลิตภัณฑ์ว่าจะจัดการชิ้นๆ ของข้อมูลเหล่านั้นต่อไปด้วยวิธีการเช่นใด ตัวอย่างของเทคโนโลยีที่มีการใช้ Distributed Computing และ Clustering Processing ที่เป็นที่นิยมคือ Apache Hadoop ซึ่งสำหรับใน Hadoop นั้น เมื่อข้อมูลผ่าน Framework เป็นที่เรียบร้อยแล้ว จะเข้าสู่กระบวนการที่เรียกว่า Map Reduce Processes ซึ่งประกอบไปด้วย Mapping ทำหน้าที่สร้าง List ของ Keys กับ Values ดูแลโดย Mapper และ Reducing ซึ่งทำหน้าที่จับคู่ Key และ Value เพื่อออกเป็นผลลัพธ์ของการประมวลผล ดูแลโดย Reducer จากนั้นจึงออกมาเป็นผลลัพธ์ เพื่อนำออกไปสู่ส่วนที่ทำหน้าที่เก็บ และบันทึกข้อมูล หรือส่วนแสดงผลต่อไป

2.1.2.3 ส่วนบันทึกข้อมูล

สำหรับส่วนบันทึกข้อมูลของกระบวนการบริหารจัดการบิ๊กดาตานั้น ส่วนที่สำคัญอยู่ที่การใช้เทคโนโลยีเพื่อบริหารจัดการข้อมูลขนาดใหญ่ สิ่งที่ทำให้แตกต่างกันระหว่างการเก็บข้อมูลสำหรับข้อมูลธรรมดาทั่วไปกับข้อมูลที่มีปริมาตรมาก มีความเร็วนำเข้าข้อมูลสูง และมีชนิดของข้อมูลมีความหลากหลาย ยกตัวอย่างให้เข้าใจง่ายขึ้น เช่นการบริหารจัดการข้อมูลด้วยรูปแบบของโครงตาราง ซึ่งมีการใช้ Relational Databse การจัดเก็บข้อมูลด้วยตารางที่เป็น Relation นั้นสามารถบริหารจัดการได้ง่ายเมื่อมีข้อมูล 100 Rows ซึ่งตารางดังกล่าวอาจจะประกอบไปด้วยข้อมูล 1,000 หรือ 10,000

Rows แต่ถ้าลองจินตนาการถึงข้อมูลการใช้งานบัตรเครดิตในหนึ่งวันของธนาคารแห่งหนึ่ง ซึ่งเมื่อลองคิดดูแล้วธนาคารคงต้องมีการรองรับ Transactions จำนวนมากที่หลั่งไหลเข้ามาในแต่ละวัน เมื่อเทียบ 1 Transaction ต่อ 1 การใช้งานของบัตรเครดิต 1 ใบ ถ้าธนาคารต้องการจะเก็บทุก Transactions ลงตารางที่เป็น Relation ในหนึ่งวันจำนวนของ rows ในตารางคงมีมากในระดับหนึ่ง แต่ถ้าเก็บทุกวัน หนึ่งสัปดาห์จะมีปริมาณข้อมูลที่ต้องเก็บมากขึ้น 7 เท่า ถ้าคิดในหนึ่งเดือนจะมีข้อมูลที่ต้องเก็บมากกว่าในหนึ่งวันจำนวน 30 เท่า สมมติว่าลูกค้าหนึ่งรายต้องการยกเลิกรายการชำระเงินเมื่อบิลมาเรียกเก็บในแต่ละรอบเดือน ธนาคารต้องทำการค้นหาเพื่อทำการยกเลิกและระงับการชำระเงินของ Transaction นั้นๆ ซึ่งต้องทำ sequential searching ทั้งตารางเพื่อหาข้อมูลของ Transaction นั้นๆ เพื่อทำการยกเลิก ดังนั้นถ้าธนาคารมีการจัดเก็บ Transaction ในรูปแบบของตารางแล้ว ในแต่ละวันคงจะมีการประมวลผลและการบริหารจัดการข้อมูลที่สามารถเรียกเข้าถึงได้ด้วยใช้เวลามาก

จากที่กล่าวมาในตัวอย่างข้างต้นนั้น เป็นการยกกรณีศึกษาเพื่อบอกให้เข้าใจได้ว่ากระบวนการบริหารจัดการข้อมูลบิ๊กดาตานั้น ไม่เหมาะสมกับการใช้ Relational Database ซึ่งวิธีการที่นิยมในปัจจุบันคือการใช้ Non-Relational Database ซึ่งสำหรับ Non-Relational Database นั้นมีอยู่หลากหลายแนวคิดด้วยกัน แต่ที่นิยมใช้ในปัจจุบันจะมี 4 ประเภทได้แก่

Column-oriented (Wide-table) data stores

แนวคิดของ Column-oriented data store คือการเปลี่ยนตารางที่เป็น row-oriented data store เป็นการเก็บโดยยึด column เป็นสำคัญ และข้อมูลจะแยกออกเป็นแต่ละ column อย่างเป็นอิสระต่อกัน การเข้าถึงข้อมูลจะสามารถทำได้โดยการเข้าถึง column ที่ต้องการ จากนั้นจะสามารถเข้าถึงข้อมูลได้เลย โดยไม่ต้องทำการค้นหาทั้งตารางเหมือนกับ row-oriented dates store ผลิตภัณฑ์ตัวอย่างของ column-oriented data store เช่น HBase, Hypertable และ Cassandra เป็นต้น

Key-Value Databases

แนวคิดของ Key-Value Databases มีลักษณะการใช้งานเป็น associative array ที่ทำหน้าที่เก็บ Key และ Value ที่คู่กัน ซึ่ง key แต่ละตัวจะแตกต่างกัน และมีหน้าที่ใช้ในการเข้าถึงข้อมูล ลักษณะของ Key-Value database มีความคล้ายคลึงกันกับตารางที่เป็น relational ที่ประกอบไปด้วยหลายๆ rows แต่มีเพียง 2 columns คือ key และ value ผลิตภัณฑ์ตัวอย่างของ Key-Value databases เช่น Riak, Voldemort และ Redis เป็นต้น

Document Databases

แนวคิดของ Document Databases คือการเก็บ Document เสมือนเป็นการเก็บ Data ลงใน database การใช้ Document Databases จะช่วยให้ประหยัด และลดการสูญเสียพื้นที่ที่ไม่

จำเป็น เพราะ Document Database มีความยืดหยุ่นในการเก็บ Fields ที่แตกต่างกันได้ใน Document ซึ่งทุกๆ Documents จะถูกแสดงด้วย Keys ต่างๆ ที่แตกต่างกันเช่นเดียวกันกับ Key-Value ผลิตภัณฑ์ตัวอย่างของ Document Databases เช่น MongoDB, CouchDB และ RavenDB เป็นต้น

Graph Databases

แนวคิดของ Graph Databases นั้นมีลักษณะเหมือนกับโครงสร้างของระบบเครือข่ายที่ประกอบไปด้วย Nodes และ Edges ที่มีความเกี่ยวข้องกันระหว่าง Nodes เพื่อที่จะแสดงความสัมพันธ์ที่เกิดขึ้นระหว่าง Nodes ซึ่งแต่ละ Node จะเก็บคุณสมบัติต่างๆ เพื่ออธิบายข้อมูลที่แท้จริงของแต่ละ object ความสัมพันธ์ที่เกิดขึ้นระหว่างสอง nodes นั้นๆ และอาจจะบอกถึงทิศทางที่จะเพิ่มเติมความหมายของความสัมพันธ์เหล่านั้นด้วย ถ้าเปรียบเทียบกันกับ Entity-Relational Model(ER Model) นั้น Node จะเปรียบเสมือนกับ entity, property of a node to an attribute และ entities to relationship between nodes ผลิตภัณฑ์ตัวอย่างของ Graph Databases เช่น Neo4j เป็นต้น

ซึ่งจากที่กล่าวมาในข้างต้น เป็นกระบวนการบริหารจัดการบิ๊กดาตาที่เกิดขึ้นบน On-site storage แต่ในปัจจุบันมีเทคโนโลยี Cloud Computing ที่กำลังเป็นที่นิยมมากขึ้น ซึ่งก็มีการนำมาปรับใช้ในกระบวนการบริหารจัดการบิ๊กดาตาด้วยเช่นกัน ทั้งในรูปแบบของ Public, Private และ Hybrid Cloud Computing อีกด้วย

2.1.3 ปัญหาที่เกิดขึ้นในกระบวนการบริหารจัดการบิ๊กดาตา

จากการศึกษาข้อมูลจากบทความในหัวข้อ Top Ten Big Data Security and Privacy Challenges ทั้งในฉบับปกติ และขยายความ(Expanded) จัดทำโดย Big Data Working Group ของ Cloud Security Alliance(CSA) ซึ่งได้กล่าวถึงสิ่งที่ท้าทายด้านการรักษาความมั่นคงปลอดภัยและความเป็นส่วนตัวในกระบวนบริหารจัดการบิ๊กดาตา ได้แยกออกมาเป็น 10 หัวข้อได้แก่

1.ความปลอดภัยในการประมวลผลบนการเขียนเฟรมเวิร์คของการเขียนโปรแกรมแบบกระจาย(Secure Computations in distributed programming frameworks)

2.ความปลอดภัยที่ดีที่สุดที่สุดสามารถทำได้สำหรับส่วนเก็บข้อมูลแบบไม่มีความสัมพันธ์กัน (Security best practices for non-relational data stores)

3.ความปลอดภัยของส่วนเก็บข้อมูล และรายการการเปลี่ยนแปลง (Secure data storage and transactions logs)

4.การทำให้ถูกต้อง และการคัดกรองข้อมูลที่นำเข้าจากอุปกรณ์ปลายทาง(End-Point input validation/ filtering)

5.การตรวจสอบความปลอดภัยตามเวลาจริง(Real-time security monitoring)

6.การขยายขนาด และการประกอบการสงวนความเป็นส่วนบุคคลของการทำเหมืองข้อมูล และการวิเคราะห์ (Scalable and composable privacy-preserving data mining and analytics)

7.การบังคับใช้การเข้ารหัสการควบคุมการเข้าถึง และความปลอดภัยของการสื่อสาร (Cryptographically enforced data centric security)

8.การควบคุมการเข้าถึงในส่วนย่อย (Granular access control)

9.การตรวจสอบส่วนย่อย (Granular audits)

10.ที่มาของข้อมูล (Data Provenance)

ทั้ง 10 หัวข้อนี้ครอบคลุมทุกๆ ส่วนตั้งแต่ส่วนนำข้อมูลเข้าสู่ระบบ ส่วนประมวลผล ส่วนบันทึกข้อมูล และส่วนแสดงผล ซึ่งเมื่อทำการวิเคราะห์แล้ว จะสามารถจำแนกทั้ง 10 หัวข้อออกมาได้เป็น 4 กลุ่ม กว้างๆ ที่ประกอบไปด้วยแต่ละหัวข้อภายในดังนี้

Infrastructure Security

Secure Computations in Distributed Programming Frameworks

Security Best Practices for Non-Relational Data Stores

Data Privacy

Privacy Preserving Data Mining and Analytics

Cryptographically Enforced Data Centric Security

Granular Access Control

Data Management

Secure Data Storage and Transaction Logs

Granular Audits

Data Provenance

Integrity and Reactive Security

End-Point Validation and Filtering

Real Time Security Monitoring

ทั้ง 10 หัวข้อถูกนำมาอ้างอิงอีกหลายบทความ และมีการพูดถึงวิธีทางในการแก้ไข และบรรเทาด้วยกลไกต่างๆ เมื่อนำมาบทความจาก Expanded Top Ten Big Data Security and Privacy Challenges จาก Cloud Security Alliance มาพิจารณาแล้ว จึงดำเนินการแยกออกเป็นหัวข้อต่างๆ ที่สำคัญอันได้แก่

Challenge Topic เพื่ออธิบายหัวข้อปัญหา

Problems/Weaknesses เพื่อให้รายละเอียดเพิ่มเติมถึงปัญหา และจุดอ่อนของหัวข้อปัญหา

Solutions เพื่ออธิบายแนวทางการแก้ไขที่ควรจะทำ หรือสามารถรับมือ บรรเทาปัญหาที่เกิดขึ้น

Practical เพื่ออธิบายแนวทางการแก้ไขปัญหา หรือการรับมือ บรรเทาปัญหาที่เกิดขึ้นที่สามารถนำมาปฏิบัติใช้ได้จริง

เมื่อนำบทความมาทำการศึกษาและวิเคราะห์เพื่อจำแนกข้อมูลที่ได้จากบทความ และนำผลของข้อมูลที่ได้จากการวิเคราะห์บทความลงตารางตามหัวข้อที่ได้กล่าวไว้ในข้างต้นดังนี้ (ในตารางจะแสดงข้อมูลเป็นภาษาอังกฤษเพื่อให้สะดวกในการทำความเข้าใจ)

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
1	Secure Computations in Distributed Programming Frameworks	- Malfunctioning Compute Worker Nodes - Infrastructure Attacks - Rogue Data Nodes	- Securing the mappers - Securing the data in the presence of an untrusted mapper"	Trust establishment and Mandatory Access Control (MAC)
2	Security Best Practices for Non-Relational Data Stores	- Transactional Integrity - Lax Authentication Mechanisms - Inefficient Authorization Mechanisms - Susceptibility to injection Attacks - Lack of Consistency - Insider Attacks	- Techniques like Architectural Trade-off - HTTP Basic- or Digest-based authentication - Enforcing the authorization on a per-database level rather authorization at lower layers - Current Hashing Algorithms entrusted to replicate data across the cluster nodes crumple in the event of single node failure, resulting in load imbalance among the cluster nodes. - Lenient security mechanisms can be leveraged to achieve insider attacks.	- Using secure hashing algorithms. - Data stored in the database should never be left in the clear. - Weak authentication and authorization techniques employed. - Hardware appliance-based encryption/decryption and bulk file-based encryption. - Appropriate logging mechanisms and data tagging techniques with time stamp.
3	Secure Data Storage and Transactions Logs	- Confidentiality and Integrity - Provenance - Availability - Consistency - Roll-back Attacks - Disputes	- Confidential and Integrity can be achieved with robust encryption techniques and message-digests. - The exchange of signed messagedigests can be used to address potential disputes. - User freshness and writeserializability can be solved by periodic audit and chain hash or persistent authenticated dictionary(PAD).	- Dynamic Data Operations : An extended dynamic version of a PDP scheme achieves higher efficiency because it only relies on symmetry-key cryptography. To support both public verifiability and data dynamic in cloud storage. - Privacy Preservation

ตารางที่ 2.1 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
			- Secure untrusted data repository(SUNDR) can be used to detect fork consistency attack and write serializability. - Broadcast encryption and key rotation can be used to improve scalability. - Data availability can be improved through proof of retrievability(POR) or provable data procession(PDP) methods with high probability. - Digital rights management can prevent collusion attacks.	- Secure Manipulations on Encrypted Data : The fully homomorphic encryption scheme makes these operations possible because more complex functions are supported. - Cryptographic Cloud Storage
4	End-Point Input Validation/Filtering	- A key challenge in data collection process is input validation as how can we trust the data? How can we validate that a source of input data is not malicious? - Untrusted Input Source - A device which data is collected or application running on the device to provide malicious input to central data collection system.	- To prevent an adversary from generating and sending malicious input to the central collection system. - To detect and filter malicious input at the central system. - Preventing the sending malicious input requires tamper-proof software and defenses against Sybil attacks. - Tamper-proof secure software. For best practices have been developed to identify and remove vulnerabilities from software.	- Hybrid approach to be implemented in practice. - To develop secure data collection platforms and applications. - To consider the BYOD scenario in which their application would run on untrusted devices. - To identify plausible Sybil attacks and ID spoofing attacks on their system and then identify cost-effective ways to mitigate the attacks. - To acknowledge and be able to send malicious input to their central collection system.

ตารางที่ 2.2 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
		- ID cloning attacks on a data collection system by creating multiple fake identities and by then providing malicious input from the faked identities. - Sybil attacks become more acute in a Bring-your-owndevice(BYOD) scenario. - To manipulate the input sources of sensed data. - Data in transmission from a benign source to the central collection system as by performing a man-in-the-middle attack or replay attack.	- Be able to compromise mobile devices and the applications running on them. - Using the Trusted Platform Modules(TPMs) to guarantee the integrity of raw sensor data. - To manipulate the sensor inputs. - Defense schemes against ID cloning attacks and Sybil system by proposed in diverse areas such as peer-to-peer systems, vehicular networks, and wireless sensor networks.	- To develop algorithms to detect and filter out malicious input from an adversary.
5	Real-Time Security/Compliance Monitoring	- Monitoring the big data infrastructure itself - Using the same infrastructure for data analytics. - Common uses include utilizing the technology to answer questions such as, "Who is accessing which data from which	To provide a reduction in the number of false positives and/or increase in the quality of the true positives. - To monitor anomalous connections to the cluster and mine logging events to identify suspicious activities. - To implement mining and analytics algorithm. - To mitigate potential evasion or poisoning attacks.	- To implement the front-end systems to monitor Hadoop. - Database Activity Monitoring proxy of firewall. - Security controls - Real-time Streaming Application

ตารางที่ 2.3 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
		resource at what time" or "Are we under attack?" - The threats to big data infrastructure include rogue admin access to applications or nodes, (web) application threats, and eavesdropping on the line. - The security of the public cloud - The security of the Hadoop cluster - The security of the monitoring application itself - The security of the input sources	- To improve data management	
6	Scalable and Compassable Privacy-Preserving Data Mining and Analytics	anonymizing data for analytics is not enough to maintain user privacy. A malicious insider or untrusted partner can abuse these data sets and extract private information from customers.	to protect the user privacy, best practices in the prevention and detection of abuse by continuous monitoring must be implemented	- Differential privacy - outsourced computational resources is universal homomorphic encryption
7	Cryptographically Enforced Data-Centric Security	Big Data comes from diverse endpoints and contains more personal data, it is becoming increasingly essential to tether the visibility of the data at the source	- cryptographically-enforced access control method using encryption, the adversary should not be able to identify the corresponding plaintext data by looking at the cipher text	- current algorithms to implement identity/attribute based encryption schemes and group signatures use elliptic curve groups that support bilinear pairing maps.

ตารางที่ 2.4 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
			- cryptographic protocol for searching and filtering encrypted data, the adversary should not be able to learn anything about the encrypted data beyond whether the corresponding predicate was satisfied. - For a cryptographic protocol for computation on encrypted data, the adversary should not be able to identify the corresponding plaintext data by looking at the ciphertext - For a cryptographic protocol ensuring the integrity of data coming from an identified source, there could be a range of threat models.	
8	Granular Access Control	Legal and policy restrictions on data come from numerous sources -keeping track of secrecy requirements for individual data elements. - keeping track of roles and authorities for users. -properly implementing secrecy requirements with mandatory access control.	The first challenge is to pick the appropriate level of granularity required for a given domain.	authentication and mandatory access control.
9	Granular Audits	- More data objects distributed - Completeness of the required audit information	- Auditing capabilities need to be enabled across the Big Data infrastructure as log information	- Implementation of audit features starts on the individual component level.

ตารางที่ 2.5 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
		- Timely access to audit information especially important in case of forensics. - Integrity of the information (Audit information that has not been tampered with.) - Authorized access to the audit information - Threats (as unauthorized access, removal of data, tempering with log files) to those key factors will jeopardize the audit data and process.	from network components, applications, OS, Databases. - The challenge is to create a cohesive audit view of an attack using the available audit information of the different components.	- A forensics or SIEM tool collects, analyzes and processes this information. - The audit data might have the characteristics of Big Data infrastructure and the audit of this infrastructure. - To have the forensics/SIEM tool implemented. - Another approach would be to create an "Audit Layer/ Orchestrator".
10	Data Provenance	- Provenance metadata - Detecting metadata dependencies for security and/or confidentiality applications is computationally intensive. - The digital record for key security - These security assessments are time-sensitive in nature and require fast algorithms to handle the provenance metadata containing this information,	- To ensure the trustworthiness and usability of secure provenance in Big Data applications. - To secure provenance collection, the source components should be first authenticated. - Periodic status updates should be generated to ensure the health of the source components. - To guarantee the accuracy of the provenance records should be taken though an integrity check to assure that it is not forged or modified. - Data should be verified.	- Verification - Audit Trails - Assurance of reproducibility - Trust and fault detection in applications - To retrofit provenance in existing cloud infrastructure. - To secure the provenance collection - Fast and lightweight authentication technique - To secure channels - end-to-end security - Fine-grained access control

ตารางที่ 2.6 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

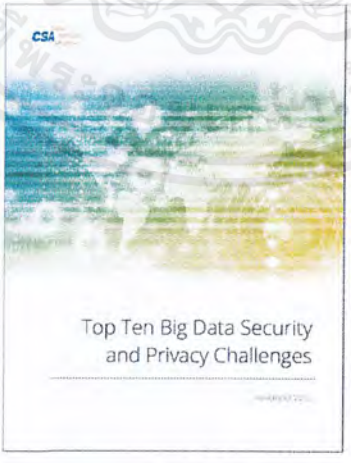
Number	Challenge Topic	Problems/Weaknesses	Solutions	Practical
		- To require the provenance records to be reliable, provenanceintegrated, privacy-preserving and access-controllable. - Malfunctioning Infrastructure Components - Infrastructure Outside Attacks — An outside attacker can forge, modify, replay, or unduly delay. - Infrastructure Inside Attacks	- The sensitive information pertaining to the data encryption techniques. - The provenance collection is secure against malfunctioning infrastructure components and outside attacks. - Fine-grained access control to inside attacker resistant.	

ตารางที่ 2.7 ตารางแสดงข้อมูลที่ได้จากการวิเคราะห์จากบทความ(ต่อ)

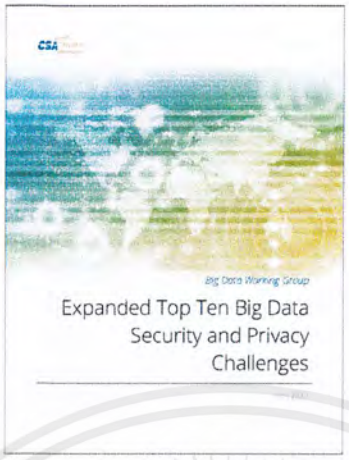
เมื่อทำการวิเคราะห์ และทำความเข้าใจปัญหาต่างๆ ที่ได้ทำการศึกษามา หัวข้อที่สนใจจึงเลือกที่จะมุ่งเป้าหมายของการศึกษาไปที่หัวข้อในกลุ่มของ Data Management อันได้แก่ Secure Data Storage and Transaction Logs, Infrastructure Security อันได้แก่ Security Best Practices for Non-Relational Data Stores และ Integrity and Reactive Security อันได้แก่ End-Point Validation and Filtering และ Real Time Security Monitoring

จากการตั้งเป้าหมายแล้ว จึงทำการศึกษารายละเอียดข้อมูลพื้นฐาน และทฤษฎีความเป็นไปของแต่ละหัวข้อ เมื่อได้ทำการศึกษาทฤษฎีอยู่จนมีความเข้าใจ อันได้อธิบายเอาไว้ในส่วนประกอบต่างๆ ของกระบวนการบริหารจัดการบิ๊กดาตา ซึ่งการเก็บข้อมูลในกระบวนการบริหารจัดการบิ๊กดาตานั้นนิยมใช้เทคโนโลยี Non-Relational Databases เป็นส่วนบันทึกข้อมูลในกระบวนการ และในส่วนของ Integrity and Reactive Security ทั้งสองหัวข้อนั้น ยังไม่มีความชัดเจนมากนักในการทำให้ข้อมูลในกระบวนการบริหารจัดการบิ๊กดาตานั้นมีความถูกต้องครบถ้วน เป้าหมายเริ่มต้นของขอบเขตการศึกษาคือ การศึกษาปัญหาและสาเหตุที่เกิดขึ้นจากกระบวนการบริหารจัดการบิ๊กดาตา โดยการเริ่มต้นในปัญหาที่เกี่ยวข้องกับความถูกต้องครบถ้วนของข้อมูล

จุดเริ่มต้นของปัญหาคือเราหรือผู้ดูแลระบบบริหารจัดการบิ๊กดาตาจะสามารถใช้วิธีการ หรือกระบวนการใด เพื่อทำให้มั่นใจได้ว่า ข้อมูลดิบ(Raw Data) ที่ถูกนำเข้ามาจากส่วนนำข้อมูลเข้าสู่ระบบนั้น เป็นข้อมูลที่ได้จากอุปกรณ์ต่างๆ นั้น เป็นข้อมูลที่ระบบต้องการจริง หรือเป็นข้อมูลที่ถูกปลอมแปลง หรือเปลี่ยนแปลงเนื้อความก่อนนำเข้ามาสู่ระบบหรือไม่ และจะมีวิธีการ หรือกระบวนการใดเพื่อทำให้มั่นใจได้ว่าข้อมูลเหล่านั้นมีความถูกต้องครบถ้วนจริง เมื่อทำการศึกษาแล้วจึงได้ทำความรู้จักเพิ่มเติมกับเทคโนโลยีไร้กุญแจหรือเรียกว่า Keyless Technology ที่จะได้กล่าวถึงในหัวข้อถัดไป



รูปที่ 2.2 บทความ Top Ten Big Data Security and Privacy Challenges



รูปที่ 2.3 บทความ Expanded Top Ten Big Data Security and Privacy Challenges

2.2 เทคโนโลยีการรักษาความมั่นคงปลอดภัยของข้อมูลด้วยกลไกการเข้า และถอดรหัสลับด้วยกลไกแบบใช้กุญแจเดี่ยว กุญแจคู่ และไร้กุญแจ (Keyless)

ก่อนที่จะมีการกล่าวถึงเทคโนโลยีการรักษาความมั่นคงปลอดภัยของข้อมูลด้วยเข้าและถอดรหัสลับรหัสข้อมูลแบบไร้กุญแจ ในที่นี้จึงจะกล่าวถึงกลไกการรักษาความมั่นคงปลอดภัยของข้อมูลด้วยการเข้าและถอดรหัสด้วยกลไกต่างๆ ที่ใช้กันโดยปกติทั่วไป ซึ่งในที่นี้จะกล่าวถึงการเข้าและถอดรหัสข้อมูลสองกลไกหลักคือกลไกการใช้กุญแจเดี่ยว(Symmetric Key Cryptography) และกุญแจคู่(Asymmetric Key Cryptography) แล้วจึงเป็นการกล่าวถึงกลไกการเข้า และถอดรหัสลับแบบไร้กุญแจซึ่งจะมีการกล่าวถึงรายละเอียดในลำดับถัดไป

2.2.1 ทำความรู้จักการเข้า และถอดรหัสลับของข้อมูล(Cryptography)

การเข้า และถอดรหัสลับข้อมูล หรือในภาษาอังกฤษใช้คำว่า “Cryptography” มาจากภาษากรีกสองคำคือ “**κρυπτο**”(krypto) ที่มีความหมายว่า การซ่อนหรือความลับ(hidden or secret) และคำว่า “**γράφη**”(gráfi) ที่มีความหมายว่า การเขียน(writing) ดังนั้น Cryptography จึงเป็นเสมือนศิลปะของการเขียนอะไรก็ตามที่เป็นความลับ โดยคนทั่วๆ ไปจะความเข้าใจว่า Cryptography นั้นคือศิลปะของการตัด ฉีก หรือตกแต่งข้อมูล ให้ปรากฏออกมาแล้วไม่สามารถที่จะเข้าใจได้ จะมีการอนุญาตด้วยวิธีการที่เป็นความลับ จึงจะสามารถแก้การตัด ฉีก หรือตกแต่งข้อมูลให้กลับมาปรากฏในรูปที่สามารถเข้าใจได้

การเข้า และถอดรหัสลับของข้อมูลโดยเบื้องต้นแล้วจะใช้ในการส่งข้อมูลกันระหว่างคนสองคนในวิถีทางที่จะป้องกันการอ่านข้อมูลนั้นจากผู้อื่น ซึ่งการเข้า และถอดรหัสลับนั้นยังสามารถ

นำมาใช้กระบวนการอื่นๆ นอกจากการใช้งานเบื้องต้น เช่น Integrity Checking (คือการทำการตรวจสอบข้อมูลที่ได้มาว่ามีความถูกต้องครบถ้วนหรือไม่) และ Authentication (เป็นกลไกในการทำการยืนยันตัวตนเพื่อให้สามารถแสดงสิทธิ์ในการเข้าใช้ระบบได้) เป็นต้น

ในกระบวนการปกติ ข้อความต้นฉบับจะถูกเรียกว่า Plaintext และข้อความที่ถูกตกแต่งแล้วจะถูกเรียกว่า Ciphertext ซึ่งกระบวนการที่เปลี่ยนจาก Plaintext เพื่อให้กลายเป็น Ciphertext จะเรียกว่า การเข้ารหัสลับ(Encryption) และกระบวนการย้อนกลับของ Encryption ที่ทำการเปลี่ยนจาก Ciphertext กลับมาเป็น Plaintext จะถูกเรียกว่า การถอดรหัสลับ(Decryption)

รูปที่ 2.4 กระบวนการ Encryption และ Decryption

สำหรับกระบวนการเข้าและถอดรหัสลับนั้น สิ่งที่จำเป็นต้องรักษาเอาไว้คือ Algorithm และ secret Value ซึ่ง Secret Value นั้นเป็นที่รู้จักกันในคำว่า “กุญแจ” หรือ “Key” ซึ่งการสร้าง และการใช้ Key นั้นจะมีการกล่าวถึงในรายละเอียดในลำดับถัดไป

2.2.2 ข้อมูล(รหัส) อันเป็นความลับ (Secret Code)

สำหรับ Secret code หรือเรียกว่า Cipher นั้น เป็นคำที่ใช้เรียกข้อความ ข้อมูล หรือเอกสารที่ผ่านกระบวนการเข้ารหัสลับมาเป็นที่เรียบร้อยแล้ว

โดยแรกเริ่มต้นนั้นมีการค้นพบเอกสารที่ถูกเข้ารหัสนั้นอ้างอิงไปถึง Julius Caesar วิธีการนี้จึงถูกเรียกว่า Caesar Cipher ซึ่งการจะทำวิธีการนี้ข้อความจำเป็นต้องอยู่ในภาษาอังกฤษเท่านั้น โดยอาศัยการแทนที่ตัวอักษรแต่ละตัวในข้อความด้วยตัวอักษรถัดไปอีกสามตัว โดยพิจารณาจากตัวอักษร A ถึง Z ด้วยวิธีการนี้ตัวอักษร A ใน Plaintext ก็จะกลายมาเป็นตัวอักษร D ใน Ciphertext ยกตัวอย่างเช่น Plaintext คือข้อความ DOZEN จะกลายมาเป็น Ciphertext คือข้อความ GRCHQ ซึ่งจะเห็นว่าวิธีการของ Caesar Cipher นั้นง่ายต่อการอ่านข้อความ Ciphertext ที่ผ่านการเข้ารหัสลับมาแล้ว (โดยอันที่จริงแล้ววิธีการของ Caesar Cipher นั้นมีการใช้ในภาษากรีก)

Caesar Cipher ได้ถูกพัฒนาเพื่อเพิ่มประสิทธิภาพให้ดีขึ้นในช่วงปี 1940 ดังเช่นอุปกรณ์ที่มีชื่อว่า Captain Midnight Secret Decoder rings ซึ่งโดยวิธีการจะมีการกำหนดตัวแปร(Variant) ขึ้นมาค่าหนึ่งซึ่งเป็นความลับ โดยสมมติว่าเป็น n ซึ่งกำหนดค่าให้ตัวแปร n มีค่าอยู่ระหว่าง 1 ถึง 25 แทนที่จะมีค่าเป็น 3 เพียงค่าเดียว เพื่อใช้ในการกำหนดตัวอักษรที่จะมาแทนที่ โดยการแทนที่

ตัวอักษรนั้นจะมีค่าตาม n ยกตัวอย่างเช่น ถ้าให้ n เป็น 2 เมื่อตัวอักษรใน Plaintext เป็น A ตัวอักษรที่ปรากฏใน Ciphertext แทนที่ตัว A คือตัวอักษร C

การทำกระบวนการเข้า และถอดรหัสลับที่ถูกพัฒนาต่อมา และเป็นที่รู้จักคือ Monoalphabetic cipher กล่าวคือการจับคู่กันของตัวอักษรแต่ละตัวอย่างอิสระ ดังนั้นจะสามารถเกิดคู่ของตัวอักษรได้ทั้งหมด 26! ซึ่งจะคิดได้ประมาณ 4x1026 กรณีที่อาจเกิดขึ้นได้ ถ้าคิดว่าการลองหนึ่งครั้งใช้เวลา 1 microsecond จะต้องใช้เวลาประมาณ 10 ล้านล้านปี ซึ่งนับว่าเป็นระยะเวลาที่ยาวนานมาก ถ้าคิดถึงการใช้คอมพิวเตอร์ในสมัยโบราณกาล

แต่ในปัจจุบันการทำ Secret Code แบบต่างๆ ที่กล่าวมาในข้างต้นนั้นไม่มีความปลอดภัยอีกต่อไปเพราะคอมพิวเตอร์มีความซับซ้อน และสามารถทำงานได้เร็วมากยิ่งขึ้น ทำให้ต้องมีกระบวนการหรือแนวคิดอื่นๆ เพื่อมาทำให้ข้อมูลมีความมั่นคงปลอดภัยมากยิ่งขึ้น

2.2.3 การเข้า และถอดรหัสลับด้วยกุญแจเดี่ยว(Symmetric Key Cryptography)

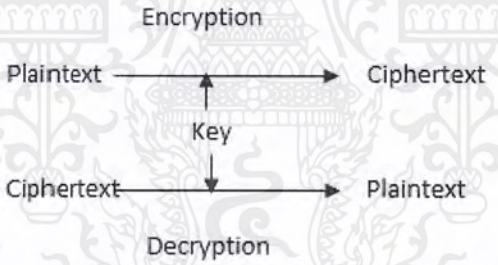
Symmetric key cryptography หรืออาจจะเรียกว่า Secret key cryptography และ Conventional cryptography เป็นกระบวนการที่มีการใช้กุญแจเดี่ยว (Single key) ความหมายคือในกระบวนการเข้า และถอดรหัสลับนั้นมีการใช้กุญแจในการเข้ารหัสลับ และถอดรหัสลับเป็นข้อมูลชุดเดียวกัน ตัวอย่างของอัลกอริทึมในกลุ่มของ Symmetric Cryptography คือ Data Encryption Standard (DES)

Data Encryption Standard(DES) เป็นอัลกอริทึมที่ใช้ในการเข้ารหัสที่มีการใช้งานอย่างแพร่หลาย คิดค้นขึ้นในปี 1976 โดยใช้ Key ในการเข้ารหัสและถอดรหัสคือ key เดียวกัน (Symmetric Cryptography) เนื่องจาก DES มีการใช้งาน Key ที่มีขนาด 56 บิต การถอดรหัสลับโดยไม่ทราบ Key จึงต้องมีการสุ่ม Key ทั้งหมด 72,000 ล้านล้าน Key ซึ่งถือว่าเป็นอัลกอริทึมที่มีความปลอดภัยสูง ในกระบวนการเข้ารหัส DES จะทำการแบ่งข้อมูลออกเป็นบล็อค บล็อคละ 64 บิต แล้วทำการเข้ารหัสแต่ละบล็อคโดยใช้ Key 56 บิต กระบวนการดังกล่าวจะทำการเข้ารหัสทั้งหมด 16 รอบตามกระบวนการของ DES ถึงแม้ว่า DES จะถือว่ามีความปลอดภัยสูง แต่ก็ยังมีการปรับปรุง DES ให้มีความปลอดภัยสูงขึ้นโดยการเปลี่ยนเป็น Triple DES ซึ่งสามารถใช้ Key ทั้งหมด 3 ชุด

แม้ว่า DES จะมีความปลอดภัยสูงแต่ก็ยังสามารถถอดรหัสลับออกมาได้ โดยในปี 1997 มีนักคณิตศาสตร์สามคนชื่อ Rivest-Sharmir-Adleman (ซึ่งภายหลังคิดค้น RSA Algorithm) โดยทำการถอดรหัสลับข้อมูลโดยได้รับความร่วมมือจากผู้ใช้คอมพิวเตอร์ประมาณ 14,000 เครื่องในอินเทอร์เน็ตร่วมกันถอดรหัสลับข้อมูลเพื่อหา Key ในการถอดรหัสลับ ซึ่งภายหลังสามารถถอดรหัสลับได้โดยการสุ่มตรวจ Key ทั้งสิ้น 18,000 ล้านล้านคีย์ จนได้รับรางวัล 10,000 เหรียญสหรัฐ

ภายหลังมีการคิดค้นการถอดรหัสลับข้อมูล DES ได้มากขึ้นจึงมีการคิดค้นกระบวนการในการเข้ารหัสใหม่ คือ Advanced Encryption Standard(AES) ซึ่งอัลกอริทึมนี้ได้พัฒนาโดย Joan Daemen และ Vincent Rijmen ในปี 2000 อัลกอริทึมนี้เป็นที่ยอมรับโดยหน่วยงานมาตรฐาน และเทคโนโลยีของสหรัฐ หรือ National Institute of Standard and Technology(NIST) ให้เป็นมาตรฐานในการเข้ารหัสของประเทศ เป็นอัลกอริทึมที่มีความเร็วสูง และมีขนาดกะทัดรัดโดยสามารถใช้กุญแจที่มีความยาวขนาด 128, 192 และ 256 บิต เพื่อเพิ่มความปลอดภัยให้สูงขึ้น นอกจากนี้ยังมีอัลกอริทึมอื่นๆ ที่ได้รับการสนับสนุนให้นำไปใช้ให้แพร่หลายอีกเช่น RC6, Serpent, MARS และ Twofish

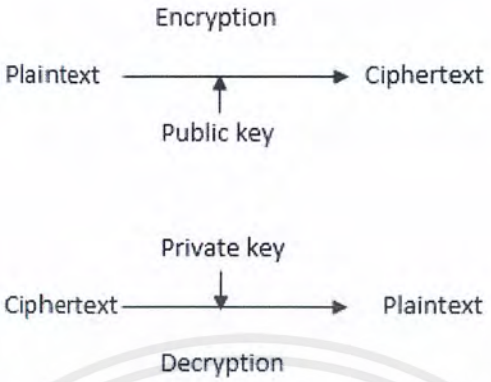
การใช้งานของ Symmetric Key Cryptography จะมีการใช้งานเพื่อการเข้ารหัสลับของข้อมูลโดยกุญแจที่ใช้ต้องเป็นกุญแจเดียวกันกับตอนที่ใช้ถอดรหัส โดยการใช้งานนั้นผู้ที่ทำหน้าที่ส่งสารเมื่อทำการเข้ารหัสข้อมูลด้วยกุญแจ และทำการส่ง ciphertext ไปให้กับผู้รับสารเป็นที่เรียบร้อยแล้ว ทางฝั่งผู้รับสารเมื่อได้รับสารเป็นที่เรียบร้อยแล้ว จะต้องใช้กุญแจชุดเดียวกันซึ่งเป็นชุดเดียวกันกับที่ผู้ส่งสารเข้ารหัสลับข้อมูลมา จึงสามารถทำกระบวนการถอดรหัสลับข้อมูลเพื่อให้ได้ข้อความที่ต้องการสื่อสารกันกลับมาได้



รูปที่ 2.5 กระบวนการทำงานของ Symmetric Cryptography

2.2.4 การเข้า และถอดรหัสลับด้วยกุญแจคู่ (Asymmetric Key Cryptography)

Asymmetric Cryptography คือกระบวนการเข้ารหัสลับที่มีการใช้ Key ในการเข้ารหัสกับ Key ในการถอดรหัสต่างกัน ในการใช้งาน หากใช้ Key ใดในการเข้ารหัสลับจะใช้ Key อีก Key หนึ่งในการถอดรหัส สำหรับ Key ที่ใช้ทั้งสอง Key จะมีชื่อเรียกว่า Private key และ Public key โดย Private key จะเป็น Key ประจำตัวของผู้ใช้งานจะถูกเก็บรักษาไว้เป็นความลับ ส่วน Public key จะเป็น Key ในการเข้ารหัสข้อมูลเพื่อส่งให้กับเจ้าของ Key สามารถแจกจ่ายให้กับบุคคลทั่วไปได้ Asymmetric Cryptography จึงมีการใช้งานในอีกชื่อหนึ่งคือ Public Key Cryptography



รูปที่ 2.6 กระบวนการทำงานของ Public Key Cryptography

ในการใช้งาน ผู้ใช้งานจะสามารถดำเนินการได้ใน 2 รูปแบบคือ
การ Sign ข้อมูลที่จะส่งด้วย Private key ของผู้ส่ง ทำให้ผู้รับสามารถมั่นใจได้ว่าข้อมูลที่ได้รับมานั้นจะเป็นข้อมูลที่ถูกต้องโดยการตรวจสอบความถูกต้องโดยใช้ Public Key ของผู้ส่ง
ทำการเข้ารหัสลับข้อมูลที่จะส่งโดยใช้ Public Key ของผู้รับ ทำให้ผู้ที่สามารถถอดรหัสข้อมูลและใช้งานข้อมูลนั้นๆ ได้คือผู้รับเท่านั้น โดยผู้รับจะทำการถอดรหัส และนำข้อมูลไปใช้งานโดยใช้ private key ของตนเอง

ในการใช้งานในระบบจริง การใช้งาน Asymmetric Cryptography นั้นประสบความสำเร็จได้เนื่องจากมีระบบการบริหารจัดการ Key (Key Management System) ซึ่งเป็นระบบเกี่ยวกับการสร้าง การเก็บ และการแจกจ่าย Public key ของผู้ใช้งานระบบได้โดยง่าย และน่าเชื่อถือ โดยระบบบริหารจัดการ Key ที่ใช้กันอย่างแพร่หลายในปัจจุบันคือ Public key Infrastructure หรือ PKI ตัวอย่างของอัลกอริทึมในการเข้ารหัสแบบ Asymmetric key cryptography คือ RSA

RSA

RSA เป็นอัลกอริทึมในการเข้ารหัสข้อมูลโดยการใช้ Key ในการเข้ารหัสกับถอดรหัสคนละ Key กัน ซึ่งจากการทำงานดังกล่าวทำให้อัลกอริทึมนี้มีการใช้งานอย่างแพร่หลายโดยเฉพาะในการสร้าง Digital Signature ของข้อมูลต่างๆ RSA ถูกคิดค้นขึ้นมาโดย Ron Rives, Adi Shamir และ Len Adleman สำหรับชื่อ RSA นั้นมาจากการนำเอาตัวอักษรตัวแรกของผู้คิดค้นขึ้นมาเรียงต่อกันตามลำดับ สำหรับแนวคิดของ RSA คือการคิดว่าการแยกตัวประกอบของตัวเลขจำนวนเฉพาะ 2 จำนวนใดๆ เป็นสิ่งที่ทำได้ยาก

Public Key Infrastructure: PKI

ระบบโครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure) PKI คือ ระบบป้องกันข้อมูลที่รับส่งกันผ่านเครือข่ายอินเทอร์เน็ต ในการทำงานของ PKI ทำได้โดยการใช้หลักการของ Asymmetric Encryption โดยการสร้าง Public Key และ Private Key ในการเข้ารหัส และถอดรหัสข้อมูล โดยกุญแจทั้งสองนี้จะได้มาพร้อมกับใบรับรองที่ Certificate Authority(CA) เป็นผู้ออกให้โดย Private Key จะถูกเก็บไว้ที่เจ้าของใบรับรองเท่านั้น ส่วน Public Key จะถูกแจกจ่ายโดย CA เพื่อนำไปใช้ในการติดต่อกับเจ้าของใบรับรองทำให้การรับส่งข้อมูลใดๆ มีความน่าเชื่อถือมากขึ้น

Certificate Authority

ในชีวิตจริงเราจะเห็นได้ว่าความน่าเชื่อถือในตัวบุคคลต่างๆ มีค่ามากหลายๆ หน่วยงานจะเชื่อถือในหน่วยงานของรัฐหรือหน่วยงานต้นสังกัดของคนๆ นั้นเป็นหลักทำให้ในการทำธุรกรรมต่างๆ ไม่ว่าจะเป็นสมัครงาน สมัครเพื่อเรียนต่อ ซื้อทรัพย์สิน กู้เงินเปิดบัญชีธนาคาร ฯลฯ จำเป็นต้องใช้บัตรประชาชนซึ่งออกให้โดยกรมการปกครองกระทรวงมหาดไทยหรือบัตรอื่นๆ ที่ออกโดยหน่วยงานนั้นๆ จึงจะสามารถทางธุรกรรม หากหน่วยงานต่างๆ เชื่อถือในตัวประชาชนคนนั้นๆ จริงๆ จะต้องสามารถดำเนินการธุรกรรมต่างๆ ได้โดยไม่จำเป็นต้องใช้บัตรประชาชนเลย

นั่นหมายความว่าการทำธุรกรรมต่างๆ จะมีความเชื่อถือกรมการปกครองกระทรวงมหาดไทยมากกว่าตัวบุคคลนั้นๆ ซึ่งเมื่อมองในความเป็นจริงก็เป็นสิ่งที่ปฏิเสธไม่ได้เนื่องจากรูปลักษณ์ภายนอกของแต่ละคนสามารถปลอมแปลงกันได้ไม่ยาก

ในการทำธุรกรรมอิเล็กทรอนิกส์ให้มีความปลอดภัยสูงการไว้ใจให้ผู้ใช้บริการสามารถทำธุรกรรมได้ด้วยตนเองโดยการสร้างคู่กุญแจในการเข้ารหัส และถอดรหัสด้วยตนเองนั้น เป็นสิ่งที่มีความเสี่ยงสูงมาก ปัจจุบันจึงมีการตั้งหน่วยงานกลางในการสร้างการรับรองและการแจกจ่ายคู่กุญแจเหล่านั้นแทนที่จะให้ผู้ใช้งานสร้างขึ้นเอง ด้วยเหตุผลหลักเพียงข้อเดียวคือไม่เชื่อถือในผู้ใช้งานแต่เชื่อถือในผู้ประกอบการรับรอง (Certificate Authority: CA) เท่านั้น

ผู้ประกอบการรับรอง (Certification Authority) หรือผู้ให้บริการรับรอง(Certification Service Provider) ซึ่งจะทำหน้าที่เป็นตัวกลางในการให้บริการโครงสร้างพื้นฐานกุญแจสาธารณะ (PKI) เพื่อตอบสนองความต้องการพื้นฐานด้านความปลอดภัยของการทำธุรกรรมอิเล็กทรอนิกส์ CA คือผู้ประกอบกิจการรับรองการใช้ Key pairs ในรูปแบบของใบรับรองอีกนัยก็คือผู้ที่รับรองความปลอดภัยของข้อมูลอิเล็กทรอนิกส์ และยืนยันความมีตัวตนของเจ้าของใบรับรองในการทำธุรกรรมได้โดยหน้าที่ของผู้ออกใบรับรองฯ มีดังนี้

1. สร้างคู่กุญแจ (Key pairs) ของผู้ใช้บริการ
2. ออกใบรับรองฯ เพื่อยืนยันตัวผู้ใช้บริการ

3. จัดเก็บและเผยแพร่กุญแจสาธารณะ

หากมีการร้องขอให้ยืนยันตัวบุคคลเจ้าของกุญแจจะดำเนินการยืนยัน หรือปฏิเสธความเป็นเจ้าของกุญแจสาธารณะตามคำขอของบุคคลทั่วไป

เปิดเผยรายชื่อใบรับรองฯ ที่ถูกยกเลิกแล้ว (Certificate Revocation List หรือ CRL) เพื่อเป็นการบอกแก่สาธารณชนว่าใบรับรองฯ นั้น ไม่สามารถนำมาใช้ได้อีกต่อไป

2.2.5 การเข้า และถอดรหัสลับด้วยกลไกไร้กุญแจ (Keyless Cryptography)

จากที่กล่าวมาในข้างต้นถึง algorithms ต่างๆ ที่เกี่ยวข้องกับการใช้งานเพื่อการเข้า และถอดรหัสลับด้วยกลไกต่างๆ ซึ่งเป็นที่นิยมในปัจจุบันทั้ง Symmetric และ Asymmetric key cryptography ทั้งสองวิธีนี้นั้นจำเป็นต้องมีกุญแจ หรือ key ที่ใช้ในการเข้า และถอดรหัสลับ เพื่อตกแต่งหรือปกปิดข้อมูลที่ต้องการจะสื่อสาร ปัญหาของ Symmetric key cryptography ในเรื่องเกี่ยวกับการบริหารจัดการกุญแจ ปัญหาที่เกิดขึ้นคือการแจกจ่ายกุญแจไปให้ผู้ที่ต้องการจะติดต่อด้วย ปัญหานี้สามารถแก้ไขได้โดยการใช้ Asymmetric key cryptography ซึ่งกลไกทั้งสองวิธีที่กล่าวมาในข้างต้นนั้นสามารถที่จะแก้ปัญหาได้เกือบทั้งหมดในการรักษาความมั่นคงปลอดภัยของข้อมูลด้วยการใช้กระบวนการเข้า และถอดรหัสลับของข้อมูล แต่ทั้งสองวิธีนั้นยังไม่สามารถแก้ปัญหาการบริหารจัดการข้อมูลในกระบวนการบริหารจัดการข้อมูลของบิ๊กดาตา ซึ่งปัญหาดังกล่าวอาทิเช่น ปัญหาของการทำ Integrity Checking ของ Big Data ที่ได้กล่าวไว้ในตอนต้น

ปัญหาของการทำ Integrity Checking สำหรับการบริหารจัดการข้อมูลทั่วไปสามารถโดยใช้กลไกการเข้า และถอดรหัสด้วย Asymmetric key cryptography เพื่อเข้ามาใช้ในกระบวนการทำ Digital Signature การเข้ารหัสลับของข้อมูลทั่วไปนั้นสามารถทำได้โดยการนำคีย์มาเข้ารหัสกับข้อมูลทั้งหมด ซึ่งกับข้อมูลที่อยู่ในกระบวนการบริหารจัดการบิ๊กดาตานั้นก็สามารถทำได้เช่นกัน แต่อาจจะต้องใช้เครื่องประมวลผลกลางที่มีประสิทธิภาพมาก สมรรถนะสูง หรือถ้าเครื่องประมวลผลกลางมีประสิทธิภาพ และสมรรถนะปานกลาง คงต้องแลกกับระยะเวลาในการประมวลผลที่ยาวนานมากขึ้น ดังนั้นการทำ Integrity Checking ให้กับการบริหารจัดการบิ๊กดาตาด้วยการทำ Digital Signature หรือการนำ key มาเข้ารหัสลับกับข้อมูลทั้งหมด นั้นจึงเป็นเรื่องที่ยากลำบากถ้าหากไม่มีความพร้อมทางด้านทุนทรัพย์

ดังนั้นกลไกการเข้า และถอดรหัสลับข้อมูลที่อยู่ภายในกระบวนการบริหารจัดการบิ๊กดาตานั้น หากสามารถกระทำได้โดยไม่ใช้กุญแจ หรือไม่ต้องนำกุญแจมาใช้ในการเข้ารหัสกับข้อมูลทั้งหมดก็จะสามารถช่วยบรรเทาปัญหาที่เกิดเหล่านี้ได้ กลไกการเข้า และถอดรหัสลับโดยไม่ใช้กุญแจจึงเป็นที่นิยมมากขึ้น ปัจจุบันมีการนำกลไกการเข้า และถอดรหัสแบบไร้กุญแจ(keyless) มาทำการศึกษาในหลากหลายด้าน จากการศึกษาจากฐานข้อมูลห้องสมุดดิจิตัลของ Institute of Electrical and

Electronics Engineers(IEEE) ค้นพบว่ามีคำค้นหาที่มีส่วนเกี่ยวข้องกับ Keyless อาทิเช่น Keyless Entry System, Keyless Steganography, Keyless Encryption, Keyless Car Entry, Keyless Security, Keyless Password-based Access Control System, Keyless Secure Acoustic Communication, Keyless Authentication เป็นต้น ซึ่งจากที่ยกตัวอย่างมาจะเห็นได้ว่า คำค้นหาว่า keyless นั้นถูกใช้อย่างแพร่หลายในหลายสาขา ทั้งที่เกี่ยวข้องกับการเข้า และถอดรหัสข้อมูล หรือการทำ Integrity Checking และไม่เกี่ยวเช่น Keyless Entry ต่างๆ จะเกี่ยวข้องกับเทคโนโลยีการเปิดประตูรถโดยไม่ใช้กุญแจ ซึ่งมีหลากหลายบทความดังจะยกตัวอย่างในลำดับต่อไป

2.2.5.1 ตัวอย่างการใช้ Keyless Algorithm ในบทความทางการศึกษา

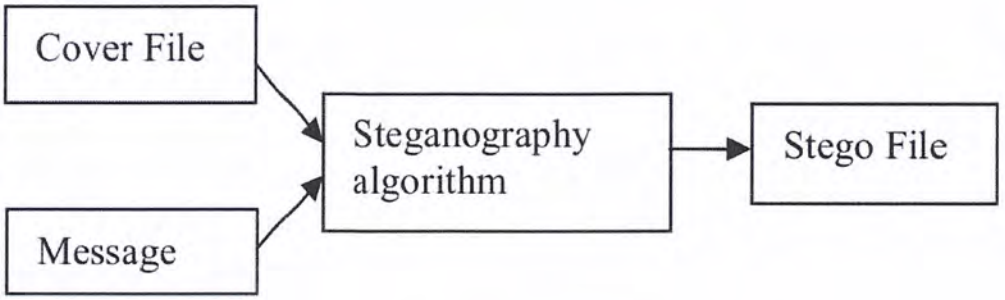
A Novel Keyless Algorithm for Steganography

(Supriya Rai and Ruchi Dubey)


A Novel Keyless Algorithm for Steganography

รูปที่ 2.7 A Novel Keyless Algorithm for Steganography

จากการศึกษาบทความ A Novel Keyless Algorithm for Steganography โดยแนวคิดคือการซ่อนข้อมูลบางอย่างลงไปในไฟล์อีกไฟล์หนึ่งที่ต้องการจะส่งไปยังผู้รับ โดยการเอาข้อความที่จะซ่อนนั้นใส่ลงไปใน Cover File โดยผ่านกระบวนการที่เกิดจาก Steganography Algorithm ซึ่งจะได้ Stego file ออกมาตามแผนภาพ



รูปที่ 2.8 Steganography Process

สิ่งที่แนวคิดของ Steganography Algorithm คือการซ่อนข้อมูลโดยการแก้ไข เปลี่ยนแปลงที่ LSB(Least Significant Bit) ซึ่งตัวอย่างที่ใช้ในบทความนี้ได้แก่การทำ Steganography กับไฟล์รูปภาพ bitmap 24 bit ที่ 1 pixel ประกอบไปด้วย R(Red) 8 bits, G(Green) 8 bits, B(Blue) 8 bits ซึ่งจากศึกษาจะพบว่าสีต่างๆ นั้นที่สีเฉดเดียวกันนั้นจะมี MSB(Most Significant Bit) ดังนั้นหากมีการเปลี่ยนแปลงที่ LSB จะทำให้รูปที่ถูกเปลี่ยนแปลงนั้นไม่สามารถแยกความแตกต่างได้ด้วยตาของมนุษย์ ข้อความที่ถูกซ่อนเอาไว้นั้นจะกระจายอยู่ใน Pixel ต่างๆ ซึ่งทำได้โดยการแบ่งรูปภาพออกเป็นบล๊อค โดยที่แต่ละบล๊อคนั้นมีขนาด 256x256 ตารางพิกเซล โดยการกำหนดพื้นที่ออกเป็นบล๊อคนั้นเพื่อใช้ในการกำหนดตำแหน่งเป็น Coordinate ให้กับพิกเซลที่จะถูกเปลี่ยนแปลง LSB ซึ่งพิกเซลที่ถูกเปลี่ยนแปลงนั้นจะเป็นตัวบอกไปถึงตำแหน่งของ pixel ถัดไปที่ถูกเปลี่ยนแปลงเช่นกัน ซึ่งสำหรับภาพที่เป็น RGB นั้นจะมีการใช้พิกัด coordinate ที่ประกอบไปด้วย (horizontal, vertical)

กรณีศึกษาของบทความนั้นได้ใช้ข้อมูลจากพิกเซลที่ทำการเปลี่ยนแปลง LSB เป็นที่เรียบร้อยแล้ว ในการกำหนดพิกเซลที่จะถูกเปลี่ยนแปลงถัดไปดังนี้

Horizontal ของพิกเซลถัดไปนั้นอ้างอิงมาจาก 5 bits ของสีแดง(R) ได้แก่ LSB ถึง LSB-4 และ 3 bits จากสีน้ำเงิน (B) ได้แก่ LSB-3 ถึง LSB-5

Vertical ของพิกเซลถัดไปนั้นอ้างอิงมาจาก 5 bits ของสีเขียว (G) ได้แก่ LSB ถึง LSB-4 และ 3 bits จากสีน้ำเงิน (B) ได้แก่ LSB ถึง LSB-2

และ LSB ของสีน้ำเงิน(B) ที่ทำหน้าที่บอกว่าบล๊อคที่จะถูกเปลี่ยนแปลง LSB ของ pixel นั้นคือบล๊อคใด ถ้า LSB = 0 คือบล๊อคถัดไปที่ติดกัน และ LSB = 1 คือบล๊อคที่ถัดไปสองบล๊อค

ข้อความที่ซ่อนอยู่ภายใต้ cover file นั้นแต่ละตัวอักษรจะถูกตีความ และแปลความหมายด้วย ASCII Code โดยการซ่อนอักษรหนึ่งตัวจะต้องอาศัยทั้งหมด 3 พิกเซล เพื่อให้ครอบคลุมครบทั้ง 7 บิตของ ASCII Code

ตัวอย่างการใช้งานอ้างอิงจากในบทความ การซ่อน ASCII ของ m = 109 = 1101101 โดยสมมติว่า พิกเซลแรกที่ได้รับการเปลี่ยนแปลงคือ (1, 1) ที่อยู่ในบล็อคแรก

I	II	III	IV
V	VI	VII	VIII
IX	X	XI	XII

รูปที่ 2.9 รูปของการแบ่งบล็อค

กำหนด Original Pixel Value : 01110000: 10010001: 11000110

Modified Pixel Value : 01110001: 10010001: 11000110

ดังนั้น Horizontal location coordinate of next pixel = 10001000 = 136

Vertical location coordinate of next pixel = 10001011 = 139

Block number of next pixel = 0

เมื่อ End coordinates ของบล็อคแรกคือ (1, 1), (256, 1), (256, 256), (1, 256)

ดังนั้นจากข้อมูลที่กำหนดด้านบน pixel ถัดไปจะอยู่ที่ตำแหน่ง (136, 256+139) = (136, 395) ในบล็อคที่ 2

Pixel location		Original pixel value	Modified pixel value	message bit	Message
(1,1) Block I	R	01110000	01110001	1	m
	G	10010001	10010001	1	
	B	11000110	11000110	0	
(136, 395) Block II	R	10111000	10111001	1	
	G	11001101	11001101	1	
	B	11011110	11011110	0	
(158, 691) Block III	R	11011101	11011101	1	
	G	11101010	11101010	--	
	B	11110000	11110000	--	

ตารางที่ 2.8 ตารางแสดงตัวอย่างของการซ่อนตัวอักษรจากข้อความลงในพิกเซลที่กำหนด

สรุปความโดยรวมที่ได้จากบทความในแนวคิดคือทำการเปลี่ยนแปลง LSB ในแต่ละพิกเซล เพื่อซ่อนข้อมูลเอาไว้ จากนั้นจึงให้แต่ละพิกเซลที่ถูกเปลี่ยนแปลงนั้นชี้ตำแหน่งของพิกเซลถัดไปเป็นลำดับด้วยข้อมูลของ current pixel ความยากในการที่จะรับรู้ รับทราบ และติดตามข้อความที่ถูกซ่อนเอาไว้ใน stego file นั้นเกิดจากการที่ไม่รู้ตำแหน่งพิกเซลแรกที่ทำหน้าที่เป็นพิกเซลเริ่มต้น การจะรู้ตำแหน่งของพิกเซลถัดไปจึงจำเป็นต้องรู้ทั้ง coordinates และ block ของพิกเซลนั้นๆ ซึ่งการจะทำการสุ่มหาพิกเซลที่เก็บ LSB ที่ถูกเปลี่ยนแปลงเอาไว้นั้น จะเป็นการยากลำบากมาก และต้องใช้เวลานาน

สิ่งที่สามารถนำมาปรับใช้ในกระบวนการบริหารจัดการบิ๊กดาตาได้จากบทความนี้ คือการเปลี่ยนข้อมูลบางส่วน หรือเพิ่มข้อมูลบางอย่างเข้าไปในข้อมูลบางส่วนที่สามารถชี้ตำแหน่งถึงกันได้ ระบบสามารถที่จะทำการประมวลผลข้อมูลในกระบวนการบางส่วนเพื่อซ่อนข้อมูลบางอย่างเอาไว้ เพื่อที่จะให้สามารถทำ Integrity Checking ให้กับกระบวนการบริหารจัดการบิ๊กดาตาได้ เพราะไม่จำเป็นต้องทำการเข้ารหัสลับระหว่างกุญแจกับข้อมูลทั้งหมด ซึ่งจะต้องทำการศึกษาและทดลองความเป็นไปได้ต่อไป

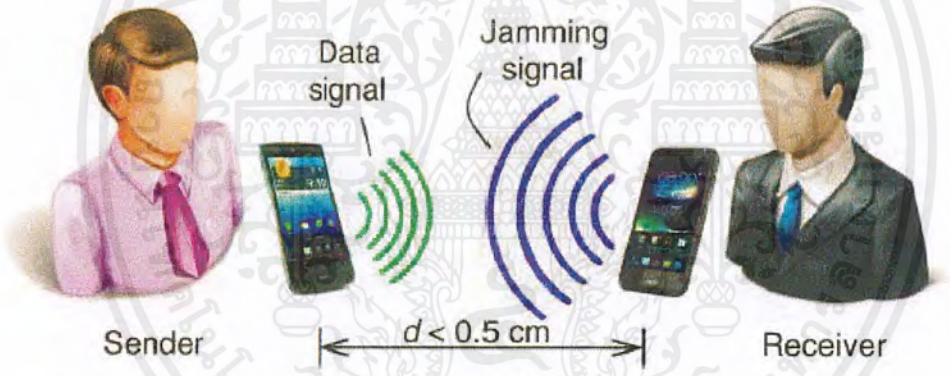
PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones

(Bingsheng Zhang and Qin Zhan)


PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones

รูปที่ 2.10 Enabling Keyless Secure Acoustic Communication for Smartphones

เทคโนโลยีการสื่อสารไร้สายระยะใกล้นั้นมีใช้อยู่มากมายในสมาร์ทโฟนปัจจุบัน ซึ่งจะมีบางอย่างที่มีความเสี่ยงสูงซึ่งจำเป็นจะต้องมีการรักษาความปลอดภัย ไม่ว่าจะเป็นการใช้จ่ายออนไลน์หรืออื่นๆ ซึ่งปกติจะเป็นการทำการใช้ กุญแจในการเข้ารหัสลับ ซึ่ง PriWhisper เป็นโปรแกรมที่ใช้ในการรักษาความปลอดภัยของสมาร์ทโฟนซึ่งไม่มีกุญแจในการเข้ารหัสลับนั่นเอง (Keyless) ซึ่งตัว PriWhisper นั้นจะใช้ Near Field Communication (NFC) ในการรักษาความปลอดภัย โดยที่ NFC นั้นจะส่งคลื่นวิทยุติดต่อกันระหว่างสมาร์ทโฟน 2เครื่อง ซึ่งเป็นเทคโนโลยีที่พบได้ทั่วไปในปัจจุบัน ตัวอย่างเช่น Google Wallet ซึ่งทำให้สมาร์ทโฟนสามารถเก็บข้อมูลบัตรเครดิต/เดบิต ไว้บนเซิฟเวอร์ของ Google และทำการใช้งาน NFC เพื่อใช้เป็นเสมือนบัตรเครดิตในการซื้อของได้ ซึ่งPriWhisper นั้นใช้การรบกวนของคลื่นสัญญาณวิทยุเป็นตัวรักษาความปลอดภัยของข้อมูลเพื่อทำให้เกิดการเข้ารหัสลับแบบไร้กุญแจ โดยจะให้ทางฝั่งผู้รับเป็นฝ่ายส่งสัญญาณรบกวน(artificial noise) ออกไป และฝ่ายผู้ส่งทำการส่งข้อมูลออกไป ดังภาพ



รูปที่2.11 แผนภาพการทำงานของPriWhisper

ซึ่งวิธีการดังกล่าวจะทำให้ไม่มีใครสามารถถอดรหัสข้อมูลที่ถูกสัญญาณรบกวนได้ ยกเว้นฝั่งผู้รับอยู่แล้วที่รู้ว่าสัญญาณรบกวนเป็นเช่นไร ทางฝั่งผู้รับจึงสามารถลบสัญญาณรบกวนออกไปได้เพื่อที่จะได้ข้อมูลมา โดยที่ PreWhisper นั้นได้ถูกออกแบบให้ใช้งานกับสมาร์ทโฟนเพื่อเชื่อต่อกับสมาร์ทโฟน และสมาร์ทโฟนกับอุปกรณ์อื่นๆ โดยที่ PreWhisper นั้นได้ถูกออกแบบมาให้ใช้งานได้โดยการติดตั้งเพียงแค่Software อย่างเดียวโดยไม่ต้องใช้อุปกรณ์อื่นๆเข้าช่วยเพิ่มเติม ซึ่งสิ่งที่จำเป็นสำหรับ PreWhisper ก็คือ NFC เพราะฉะนั้น สมาร์ทโฟนที่ใช้งานจึงจะต้องมี NFC ด้วย และเพื่อความปลอดภัยสูงสุดควรจะเชื่อมต่อเพื่อส่งข้อมูลกันโดยอุปกรณ์ทั้ง2 ทั้งผู้ส่งและผู้รับควรจะอยู่ห่างกันไม่เกิน 0.5 เซนติเมตร

2.3 โปรแกรม Apache Hadoop

Hadoop เป็น Open source Software ของ Apache สำหรับการประมวลผลข้อมูลแบบกระจาย สำหรับข้อมูลจำนวนมากผ่าน clusters ของคอมพิวเตอร์โดยมีการออกแบบให้สามารถใช้ได้ตั้งแต่เซิฟเวอร์เพียง1ตัว หรือมากกว่านั้นซึ่งแต่ละเครื่องนั้นจะมี ที่เก็บข้อมูลและการคำนวณแยกกันแทนที่จะพึ่งฮาร์ดแวร์เพียงอย่างเดียว โดย Hadoop นั้นถูกสร้างขึ้นมาสำหรับประมวลผลข้อมูลจำนวนมหาศาล ซึ่งถูกใช้งานอย่างแพร่หลายในปัจจุบัน เช่น Facebook, eBay และบริษัทอื่นๆอีกมากมาย เป็นต้น

รูปที่2.12 สัญลักษณ์โปรแกรม Apache Hadoop

2.3.1 Architecture

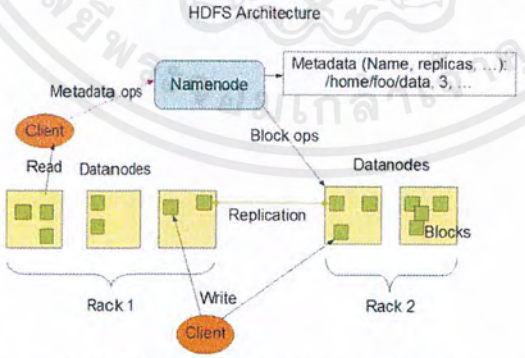
Hadoop ประกอบด้วย

2.3.1.1 Hadoop Common

เป็นโปรแกรมอรรถประโยชน์ต่างๆ ที่คอยสนับสนุนส่วนอื่นๆของ Hadoop

2.3.1.2 Hadoop Distributed File System (HDFS)

เป็น distributed file system ที่มีไว้สำหรับ Hadoop framework เป็นส่วนที่ทำหน้าที่ในการเข้าถึงข้อมูลโดย HDFS มีโครงสร้างดังนี้



รูปที่2.13 แผนภาพโครงสร้างของ Hadoop Distributed File System

2.3.1.3 Hadoop YARN

Yet Another Resource Negotiator (YARN) เป็น framework สำหรับจัดการ cluster และจัดตารางงานสำหรับ Hadoop โดยที่เมื่อใช้ YARN จะทำให้สามารถใช้งานได้

หลายโปรแกรมที่ใช้ทรัพยากรร่วมกันพร้อมกันบน Hadoop โดยที่ตัว YARN นั้นจะคอยทำหน้าที่บริหารจัดการ CPU และ Memory

2.3.1.4 Hadoop MapReduce

MapReduce เป็นโปรแกรมที่ใช้ในการวิเคราะห์ข้อมูลขนาดใหญ่ โดยจะเป็นการประมวลผลแบบขนาน โดยจะเป็นส่วนที่มีหน้าที่ลดขนาดของข้อมูลลงด้วยการMap เพื่อกระจายข้อมูลออกไปประมวลผลแบบขนานพร้อมๆกัน โดยที่ MapReduce นั้นจะแบ่งออกเป็นการ Map และการ Reduce โดยการ Map จะเป็นการนำข้อมูลที่ได้รับมาแบ่งอีกทีเพื่อทำการกระจายข้อมูลออกไปประมวลผล หลังจากนั้นจึงนำมารวมกันเพื่อทำการ Reduce อีกทีซึ่งการ Reduce คือการนำผลลัพธ์ที่ได้จากการ Map กลับมารวมกันเพื่อมาประมวลและสรุปผลให้ได้ผลลัพธ์ที่ต้องการออกมา ซึ่ง MapReduce นั้นจะเหมาะแก่การใช้งานในตอนที่เรามีข้อมูลจำนวนมหาศาล ที่ไม่สามารถใช้คอมพิวเตอร์เครื่องเดียวประมวลผลได้หมดโดยจะทำการ Map และ Reduce เพื่อลดปริมาณข้อมูลลง

2.4 โปรแกรม Apache Hive

Apache Hive คือ ซอฟต์แวร์ที่ทำงานอยู่บน Apache Hadoop เป็น data warehouse ซึ่งมีหน้าที่ในการ สรุป จัดเรียงและวิเคราะห์ข้อมูล โดยมีความสามารถในการใช้คำสั่ง SQL พื้นฐานเช่น insert, select เป็นต้น โดยที่ Apache Hive นั้นเหมาะแก่การทำ data warehouse สำหรับข้อมูลจำนวนมหาศาลไม่เหมาะแก่การนำไปใช้กับข้อมูลปกติ

รูปที่2.14 สัญลักษณ์โปรแกรม Apache Hive

2.5 โปรแกรม Apache Hbase

Apache Hbase เป็น non-relational database (NoSQL) ที่ทำงานอยู่บนโปรแกรม Apache Hadoop โดยที่ Hbase นั้นทำให้เราสามารถสุ่มเข้าถึงข้อมูลบิ๊กดาตาได้แบบ real time โดย Hbase จะใช้ Log Structured Merge trees (LSM trees) ในการเก็บและจัดเรียงข้อมูลต่างๆ

รูปที่2.15 สัญลักษณ์โปรแกรม Apache Hbase

2.6 โปรแกรม Apache Kylin

Kylin ได้ถูกตอบรับให้เป็น Apache Incubator Project ในวันที่ 25 พฤศจิกายน 2557 ซึ่ง Kylin นั้นในภาษาไทย แปลว่ากิเลน ซึ่งกเลนคือสัตว์ในตำนานของจีน ที่เกิดจากสัตว์หลายๆตัวมาผสมกัน โดยที่ตัวโปรแกรม Apache Kylin เป็น Open source Distributed Analytics Engine จาก eBay Inc. โดยตัว Apache Kylin นั้นมีหน้าที่ให้บริการ SQL interface และ multi-dimensional analysis(OLAP) บน Hadoop

รูปที่2.16 สัญลักษณ์โปรแกรม Apache Kylin

โดยที่ตัวโปรแกรม Kylin นั้น โดยที่ Kylin จะทำงานกับ HDFS ,Hive, Hbase ,MapReduce

ในส่วนการติดตั้ง Kylin นั้น เราจำเป็นจะต้องมี

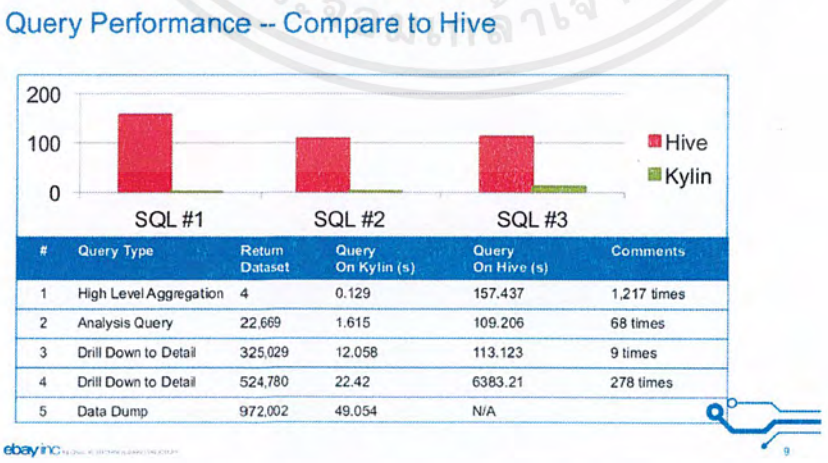
1. Hadoop 2.2.0.2.06.0-61 ขึ้นไป
2. Hive 0.12.0.2.0.6.0-61 ขึ้นไป
3. Hbase 0.96.0.2.6.0-61-hadoop2

โดยที่ Kylin มีแผนภาพการทำงานดังนี้

รูปที่2.17 แผนภาพการทำงานของ Apache Kylin

โดยที่ Kylin จะแบ่งการประมวลผลข้อมูลเป็น2ส่วนโดยส่วนแรก Kylin จะทำการอ่านข้อมูลจาก Hive และทำการ MapReduce ข้อมูลหลังจากนั้นจึงทำการบันทึกข้อมูลลงบนHbase เป็น OLAP cubes หลังจาก OLAP cubes พร้อมใช้งานผู้ใช้งานสามารถเชื่อมต่อไปยัง Kylin's REST server ได้

ตัวโปรแกรม Kylin นั้นสามารถเชื่อมต่อกับ BI tools ต่างๆได้ เช่น Tableau เป็นต้น โดยที่ Kylin นั้นมี Web interface ที่ง่ายต่อการใช้งาน และมี ACL ในการรักษาความปลอดภัยให้ในระดับ Cube/Project ของตัว Kylin นอกจากนี้ Kylin ยังสามารถทำการ query ได้เร็วกว่า Hive อีกด้วย เปรียบเทียบได้จากตารางดังต่อไปนี้



#	Query Type	Return Dataset	Query On Kylin (s)	Query On Hive (s)	Comments
1	High Level Aggregation	4	0.129	157.437	1,217 times
2	Analysis Query	22,669	1.615	109.206	68 times
3	Drill Down to Detail	325,029	12.058	113.123	9 times
4	Drill Down to Detail	524,780	22.42	6383.21	278 times
5	Data Dump	972,002	49.054	N/A	

ตารางที่ 2.9 ตารางเปรียบเทียบความเร็วในการQuery Kylin กับ Hive

2.7 ภาษาไพทอน (Python programming language)

เป็นภาษาโปรแกรมระดับสูงที่ถูกสร้างขึ้นมาโดย กีโด ฟาน รอสซัม (Guido van Rossum) ในปี พศ.2533 ซึ่งปัจจุบันถูกดูแลโดย มูลนิธิซอฟต์แวร์ไพทอน (Python Software Foundation) ซึ่งไพทอนเป็นภาษาสคริปต์ (Scripting Language) ซึ่งทำงานโดยมีอินเทอร์พรีเตอร์ (Interpreter) ในการแปลงคำสั่งเป็นภาษาเครื่อง ซึ่งจะทำงานโดยแปลงคำสั่งทีละบรรทัด โดยที่ภาษาไพทอนนั้นจะไม่ยึดติดกับแพลตฟอร์ม หมายความว่าสามารถรันโปรแกรมภาษาไพทอนได้บนระบบปฏิบัติการที่หลากหลาย ไม่ว่าจะเป็นระบบปฏิบัติการ ลินุกซ์ (Linux), ยูนิกซ์ (Unix), วินโดวส์ (Windows)

รูปที่2.18 สัญลักษณ์ Python

2.7.1 หลักการทำงานของภาษาไพทอน

ภาษาไพทอนนั้นแปลคำสั่งโค้ดโดยใช้อินเทอพรีเตอร์ซึ่งจะทำงานโดยการแปลคำสั่ง บรรทัดต่อบรรทัด คือ อ่านคำสั่งขึ้นมาแล้วทำงานเลยทีละบรรทัด

2.8 โปรแกรมไพชาร์ม (PyCharm)

เป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมภาษาไพทอนซึ่งช่วยให้การเขียนโปรแกรมภาษาไพทอนง่ายขึ้นโดยที่โปรแกรมไพชาร์มนั้นสามารถรันโค้ดภาษาไพทอนมาเพื่อทดสอบได้ นอกจากนี้ยังมี Code Editor ซึ่งช่วยในการเขียนโค้ดภาษาไพทอนที่มาพร้อมกับคุณสมบัติต่างๆมากมายที่ช่วยอำนวยความสะดวกให้แก่ผู้ใช้งานในการเขียนโค้ดเพื่อความสะดวกที่มากขึ้น ไม่ว่าจะเป็นระบบการไฮไลท์syntax, ระบบเว้นบรรทัดอัตโนมัติ, การคอมเมนต์โค้ดหลายบรรทัดพร้อมกัน, การไฮไลท์เออเรอร์ ที่เกิดขึ้นจากการเขียนโค้ด เป็นต้น

รูปที่2.19 สัญลักษณ์โปรแกรม PyCharm

2.9 Cloudera Manager

Cloudera Manager เป็นแพลตฟอร์มที่ช่วยในการบริหารจัดการบิ๊กดาตาโดยมีหน้าที่ช่วยให้การใช้งานโปรแกรมต่างๆนั้นง่ายขึ้น เช่น HDFS, Apache Hive, Apache Hbase, Apache Spark เป็นต้น

นอกเหนือจากนี้ Cloudera มีหน้าที่ในการบริหารจัดการ คลัสเตอร์ต่างๆ ภายในระบบ เพื่อให้เราสามารถ ตรวจสอบคลัสเตอร์แต่ละตัวได้อย่างมีประสิทธิภาพและสามารถควบคุมการทำงานต่างๆของคลัสเตอร์ทั้งหมดได้อย่างง่ายดายเนื่องจากมีหน้าต่างผู้ใช้งานที่เป็นมิตรแก่ผู้ใช้งาน

รูปที่2.20 สัญลักษณ์โปรแกรม Cloudera

2.10 โปรแกรม Apache Spark

Apache Spark เป็นโปรแกรม Framework Open Source ที่ช่วยในการประมวลผลข้อมูล เพื่อทำการจัดการกับข้อมูลจำนวนมหาศาลซึ่ง Apache Spark จะช่วยในการ Streaming ข้อมูลเข้าไปเพื่อประมวลผลซึ่ง Apache Spark นั้นจะมี APIs มาให้ ใช้งานได้หลายภาษาไม่ว่าจะเป็น Java, Python, Scala เป็นต้น

รูปที่2.21 สัญลักษณ์โปรแกรม Apache Spark

2.11 การตรวจสอบความถูกต้องของข้อมูลโดยวิธีการสร้างลายน้ำ (Integrity Checking using Watermarking Algorithm)

กระบวนการตรวจสอบความถูกต้องของข้อมูลโดยวิธีการสร้างลายน้ำ (Integrity Checking using Watermarking Algorithm) นั้นเป็นเทคโนโลยีที่มีการใช้งานกันในระบบเครือข่ายของตัวตรวจจับไร้สาย (Wireless Sensor Network) ซึ่งจะมีการกล่าวถึง และนำเทคโนโลยีดังกล่าวมาอ้างอิงในโครงงานนี้

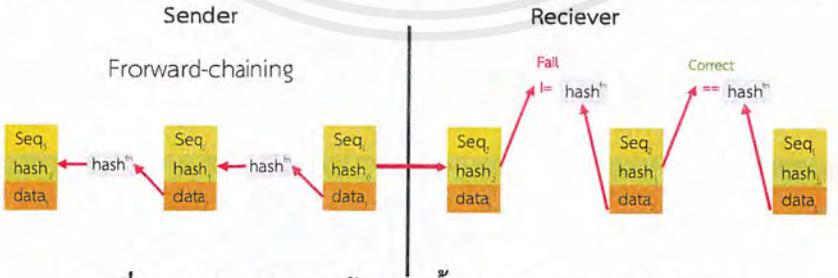
ความถูกต้องครบถ้วนของข้อมูลนั้นถือเป็นความต้องการหลักๆ ของการรักษาความปลอดภัยให้กับระบบสารสนเทศต่างๆ ความผิดพลาดหรือข้อมูลที่ถูกปลอมแปลงนั้นจะส่งผลให้ผลของการตัดสินใจต่างๆ ไม่ถูกต้อง หรืออาจเกิดความเสียหายในเชิงพาณิชย์ ซึ่งนั้นก็ถือเป็นความท้าทายหลักๆ ของการรักษาความปลอดภัยให้กับระบบสารสนเทศ กล่าวคือในทางหนึ่งนั้นอุปกรณ์มีจำกัด เช่น ความสามารถในการคำนวณ พลังงานที่ต้องใช้ รวมไปถึงปริมาณหน่วยความจำ และในอีกทางหนึ่งนั้นคือความต้องการทางด้านความปลอดภัยเองก็เป็นสิ่งสำคัญ ส่วนใหญ่แล้วการรักษาความปลอดภัยในระบบสารสนเทศ โดยทั่วไปจะใช้การเข้า และถอดรหัสลับของข้อมูล ซึ่งเทคนิคนี้จะทำงานโดยใช้หลากหลายคำสั่ง และการคำนวณต่างๆ เช่น modular exponentiation ทำให้การใช้เทคนิคนี้มี

ราคาแพง และไม่เหมาะสมสำหรับระบบที่มีการประมวลผลตลอดเวลา หรือมีข้อมูลไหลเข้าอย่างต่อเนื่อง

การสร้างลายน้ำดิจิตัล(digital watermarking) นั้นเป็นเทคนิคที่ไม่ซับซ้อนมาก ซึ่งสามารถที่จะทำให้เกิดการปกป้องการละเมิดลิขสิทธิ์ และรักษาความถูกต้องครบถ้วนของข้อมูลได้อย่างมีประสิทธิภาพ แนวคิดของการสร้างลายน้ำดิจิตัลนั้นเป็นการฝังข้อมูลอันเป็นความลับบางอย่างเข้าไปที่กลุ่มของข้อมูลโดยตรง ซึ่งไม่ทำให้ขนาดของแพ็คเกจ(packet) เพิ่มขึ้นมากนัก โดยในแนวคิดหลักของการสร้างลายน้ำดิจิตัลนั้นคือการฝังชิ้นข้อมูลอันเป็นความลับบางอย่างเข้าไปในข้อมูลที่กำลังไหล ซึ่งการกระทำทำเช่นนี้เพื่อที่จะเปลี่ยนแปลงด้วยการเพิ่มลายน้ำของข้อมูลกับข้อมูลดั้งเดิมที่อาจจะถูกโจมตีได้ การกระทำเช่นนี้จะสามารถทำให้ข้อมูลที่ถูกส่งมีความถูกต้องครบถ้วนได้ ซึ่งการโจมตีที่ได้กล่าวมาในข้างต้นนั้น ยกตัวอย่างเช่น การโจมตีด้วยการเปลี่ยนแปลงแก้ไขข้อมูล(Data Modification Attack) ซึ่งเป็นการเปลี่ยนแปลงค่าบางอย่างของข้อมูลที่ส่ง, การแทรกข้อมูลอันเป็นเท็จ(False Data Insertion) เป็นการแทรกโหนดของข้อมูลอันเป็นเท็จเพิ่มเข้ามาในระหว่างการส่งข้อมูล, การลบข้อมูล (Data deletion) เป็นการลบโหนดของข้อมูลไประหว่างการส่ง เป็นต้น

การสร้างลายน้ำของข้อมูลที่กล่าวถึงนี้ เริ่มต้นที่การแยกข้อมูลที่กำลังไหลออกไปเป็นกลุ่มๆ ซึ่งอาจมีขนาดแตกต่างกันได้อย่างหลากหลาย และฟังก์ชั่นแฮชที่ปลอดภัย(Secure Hash Function) ถูกนำมาประยุกต์ใช้กับแต่ละกลุ่มของข้อมูลที่กำลังไหลอยู่ ซึ่งโดยทั่วๆ ไปแล้วเราจะใช้ค่าแฮชของเลขศูนย์(0) เป็นค่าที่บ่งบอกจุดสิ้นสุดของชุดข้อมูล

กระบวนการสร้างลายน้ำของข้อมูล (Watermarking Embedding Processes) เริ่มต้นที่การส่งข้อมูลชุดที่ 1 (Data1) เข้าไปที่ฟังก์ชั่นแฮชที่ปลอดภัย(Secure Hash Function) เพื่อที่จะคำนวณให้ได้ค่าแฮชที่มีลักษณะเฉพาะ จากนั้นนำค่าแฮชที่คำนวณได้เก็บไว้ที่บัฟเฟอร์ แล้วนำข้อมูลชุดถัดไป(Data2) มาเข้าแฮชฟังก์ชั่นเดียวกัน นำค่าแฮชที่ได้จากข้อมูลชุดที่ 1 มาต่อที่หัว(Header) ของข้อมูลชุดที่ 2 และนำค่าแฮชของข้อมูลชุดที่ 2 มาเก็บในบัฟเฟอร์แทนที่ค่าแฮชของข้อมูลชุดที่ 1



รูปที่2.22 กระบวนการสร้างลายน้ำแบบ Forward-chaining

ตัวอย่างของฟังก์ชั่นแฮชที่ปลอดภัย(Secure Hash) เช่น MD5, SHA-1, SHA-512 เป็นต้น ซึ่งประเด็นสำคัญอยู่ที่ฝั่งผู้ส่งข้อมูล และฝั่งผู้รับข้อมูลต้องเลือกใช้ฟังก์ชั่นแฮชเดียวกัน เพื่อให้ผลลัพธ์หรือค่าที่ได้จากการแฮชด้วยข้อมูลเดียวกันมีค่าเท่ากัน

2.12 ความสัมพันธ์ของกระบวนการบริหารจัดการข้อมูลบิ๊กดาตา การรักษาความปลอดภัย และตรวจสอบความถูกต้องครบถ้วนของข้อมูล ที่มีการนำมาใช้กับผลิตภัณฑ์ต่างๆ ในปัจจุบัน

ในปัจจุบันมีการใช้บริการต่างๆ เพื่ออำนวยความสะดวกให้กับการใช้ชีวิตประจำวัน ความสำคัญของการบริหารจัดการข้อมูลต่างๆ ในการดำรงชีวิตจึงมีเพิ่มมากขึ้น สภาวการณ์ของการอยู่ร่วมกันในสังคมปัจจุบัน ข้อมูลต่างๆ ที่มีความเป็นส่วนตัวนั้น ไม่ว่าเราจะเลือกซื้อสินค้าต่างๆ ผ่านบัตรเครดิต เดินทางไปยังที่ต่างๆ โดยรถบริการสาธารณะ รวมไปถึงการอาศัยอยู่ตามที่ต่างๆ มีความจำเป็นต้องเปิดเผยให้กับใคร หรือองค์การใดๆ รับรู้ รับทราบ เพื่อแลกกับความสะดวกในการรับบริการอันเป็นสาธารณะดังที่กล่าวไปในข้างต้น ข้อมูลต่างๆ อันเป็นส่วนบุคคลจึงถูกละเมิดความเป็นส่วนตัวโดยชอบธรรม จากองค์การที่หวังประโยชน์ในเชิงพาณิชย์ และมีแนวโน้มเพิ่มมากขึ้นเรื่อยๆ ต่อไปในอนาคต

2.12.1 การนำกระบวนการบริหารจัดการข้อมูลบิ๊กดาตา การรักษาความปลอดภัย และการตรวจสอบความถูกต้องครบถ้วนของข้อมูลที่มีการนำมาใช้กับ Uber

Uber เป็นบริษัทที่ทำงานเกี่ยวกับการให้บริการพนักงานขับรถส่วนตัว โดยให้บริการผ่านทางแอพพลิเคชั่น(Application) บนระบบปฏิบัติการต่างๆ ของโทรศัพท์เคลื่อนที่แต่ละชนิด ผู้ใช้งานสามารถดาวน์โหลดมาติดตั้งบนเครื่องโทรศัพท์เคลื่อนที่ส่วนตัวได้โดยไม่เสียค่าใช้จ่าย และทำการสมัครสมาชิก ซึ่งในขั้นตอนของการสมัครสมาชิกนั้นจะมีการร้องขอให้ผู้สมัครบริการได้อนุญาตให้ระบบได้รับสิทธิ์บ้างอย่างเพื่อเข้าถึงข้อมูลส่วนบุคคลของผู้สมัครเพื่อรับบริการ ซึ่งข้อมูลส่วนบุคคลดังกล่าวจะมีการกล่าวถึงในภายหลัง ซึ่งแน่นอนว่าการร้องขอข้อมูลต่างๆ ดังกล่าวนั้นเป็นการขอให้เชิงบังคับเพราะถ้าไม่ตกลงก็จะไม่สามารถใช้บริการได้ เมื่อการสมัครบริการเสร็จสิ้นเป็นที่เรียบร้อยแล้ว จะสามารถใช้บริการได้โดยการใช้รหัสบัตรเครดิตในระบบเพื่อใช้ในการชำระค่าบริการ แอพพลิเคชั่นใช้ในการดูข้อมูลส่วนตัวของผู้ใช้งาน เลือกประเภทรถที่จะใช้บริการ(ราคามีความแตกต่างกัน) เลือกสถานที่ที่รถจะไปรับ และเลือกสถาที่ที่ต้องการให้รถไปส่ง ระบบจะทำการคำนวณค่าเดินทางตามระยะทาง และเวลาโดยประมาณที่ใช้ในการเดินทาง รวมไปถึงระบบจะแจ้งตำแหน่งของรถประเภทที่เลือก คันที่อยู่ใกล้ที่สุด และบอกว่าจะมาถึงจุดนัดหมายภายในกี่นาที ซึ่งผู้โดยสารหรือผู้รับบริการจะได้รับความสะดวกเป็นอย่างมากจากบริการที่เป็นระบบจาก Uber ดังที่ได้กล่าวมาแล้วในข้างต้นซึ่งทุกอย่างดูดี และเป็นสิ่งที่น่าใช้เป็นอย่างยิ่งสำหรับทุกๆ คน

สิ่งที่น่าสนใจคือข้อมูลที่จำเป็นต้องเปิดเผยให้กับแอพพลิเคชั่นของระบบ Uber ส่วนแรกเมื่อทำการเข้ามาในระบบคือการขออนุญาตให้ระบบสามารถเข้าถึงตำแหน่งที่อยู่ในปัจจุบันได้ ไม่เพียงแต่การประมวลผลเพียงเท่านั้น แต่ยังรวมไปถึงการบันทึกเก็บเอาไว้เป็นข้อมูลเพื่อใช้ในการวิเคราะห์เพื่อการทำนายอนาคตต่างๆ ซึ่งจะมีการให้รายละเอียดเป็นลำดับต่อไป ซึ่งทางคณะผู้จัดทำได้ทำการวิเคราะห์ในมุมมองของบุคคล 3 ประเภทอันได้แก่ มุมมองจากคนภายในของ Uber, มุมมองของผู้พัฒนาโปรแกรม และมุมมองของผู้ใช้ทั่วๆ ไป ซึ่งทั้งสามมุมมองนี้จะมีการกล่าวถึงจุดที่น่าสนใจในแต่ละสิ่งที่แตกต่างกันดังนี้

มุมมองที่ 1: มุมมองของพนักงานภายในของ Uber (http://blog.uber.com)

ในหน้าเว็บไซต์ของ Uber เองจะมีการพูดถึงสิ่งต่างๆ ที่ Uber ได้กระทำ และมองไว้ว่าเป็นประโยชน์ต่อผู้บริโภค และสาธารณชน โดยจัดออกมาในลักษณะของหน้าส่วนตัวของ Uber เอง หรือเรียกว่า Uber's blog ซึ่งใน blog จะมีการใช้ติดป้าย(Tag) (ซึ่งในที่นี้จะขอเรียกว่า "ติดแท็ก") ว่า #UBERDATA ซึ่งเป็นการบ่งบอกว่า Uber นั้นมีการนำข้อมูลไปใช้อย่างไร และเป็นประโยชน์ต่อสาธารณชนอย่างไร ดังเช่นการยกตัวอย่างต่อไปนี้

ตัวอย่างที่ 1: กรณีการทำงานร่วมกันของ Uber และระบบขนส่งมวลชนของเมืองลอสแอนเจลิส (Uber and LA's public transportation working together) ซึ่งตีพิมพ์ออกมาวันที่ 26 กุมภาพันธ์ 2558 โดย Lucy

จาก Santa Monica ไปยัง WeHo หรือไปจนถึงหมู่บ้านต่างๆ ในตัวเมือง ผู้คนในเมืองลอสแอเจนลิสต่างเริ่มใช้บริการ Uber มาตั้งแต่ปี 2012 เป็นเวลากว่าสามปีที่ผ่านมา Uber มีการดำเนินงานเพื่อให้ Uber มีความปลอดภัย และความน่าเชื่อถือมากที่สุด ไม่ว่าจะเดินทางไปที่ไหนก็ตาม เมื่อคุณต้องการที่จะเดินทางไปที่ต่างๆ คุณจะได้พบกับความตื่นเต้นกับการเดินทางด้วยบริการใหม่ๆ ที่เพิ่มขึ้นเช่น UberLUX, UberFresh และ UberPOOL ซึ่งล้วนเป็นนวัตกรรมที่ดึงดูดผู้คนในเมืองลอสแอนเจลิสแห่งนี้

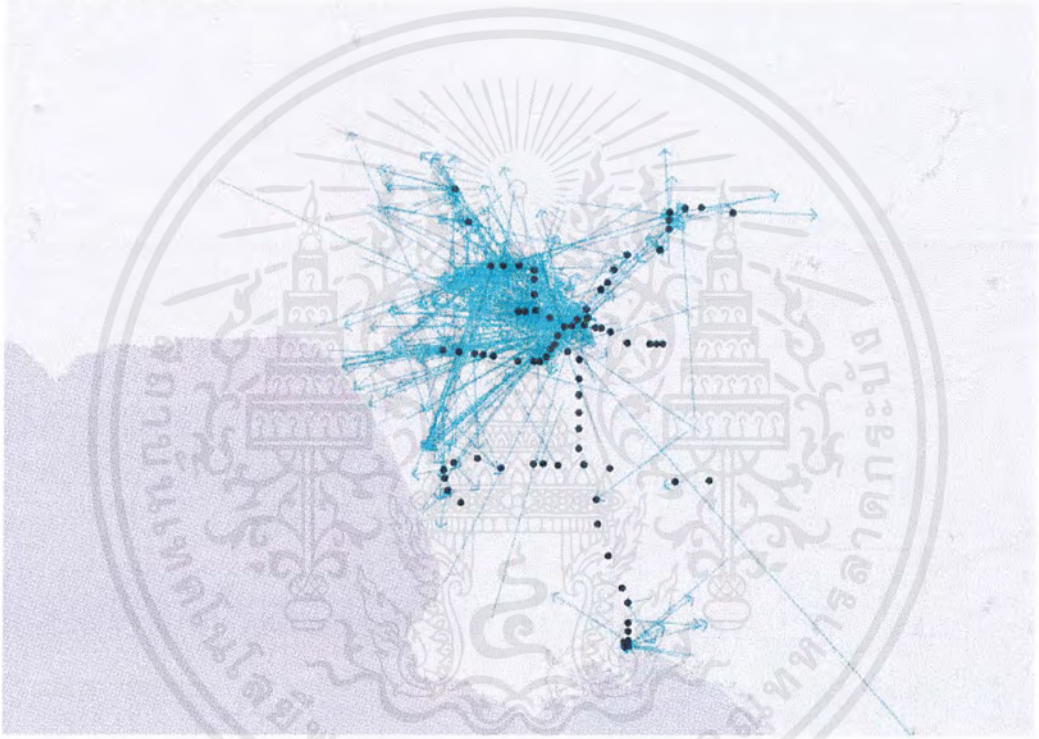
แต่ Uber ไม่ได้เพิ่งเกิดขึ้นเพียงชั่วข้ามคืน หรือเพียงไม่กี่สัปดาห์ มันเป็นการดำเนินการเพื่อให้ประชาชนสามารถติดต่อกันภายในลอสแอนเจลิส เพื่อให้ง่ายและสะดวกต่อการเดินทางไปโรงเรียน หรือไปทำงานโดยการเชื่อมต่อระบบโครงสร้างพื้นฐานของระบบการขนส่งสาธารณะ ทำให้เมืองนี้สามารถมีความสะดวกทั้งกับพลเมือง ผู้ที่มาติดต่อ รวมไปถึงนักท่องเที่ยวด้วย

Uber ทำการบันทึกจุดเริ่มต้น และจุดหมายของช่องทางการเดินทางของติดต่อกันใน 2 ปัจจัยด้วยกัน ได้แก่

1.ช่องทางการเดินทางด้วยระบบขนส่งสาธารณะ กลายมาเป็นชุมทางสำหรับ Uber ซึ่ง Uber จะไปในยังที่ที่รถประจำทาง และระบบการขนส่งอื่นๆ ไม่สามารถไปถึงได้

2.พวกเราช่วยอำนวยการเดินทางสาธารณะ ทำให้มันเป็นสิ่งที่เป็นทางเลือกมากกว่าโครงสร้างพื้นฐานของระบบการขนส่งที่มีอยู่ ซึ่งเราไปถึงที่ไหนก็ตามที่ไม่เคยมีมาก่อน

จากการประเมินและคาดคะเนดูพฤติกรรมพื้นฐานของประชาชนในลอสแอนเจลิสนั้น มีการรวมเอา Uber เข้ามาเป็นส่วนหนึ่งของการติดต่อกันในชีวิตประจำวัน พวกเรามองไปที่เดือนแรกของการให้บริการ และวันสุดทางของการดำเนินการของ 1 ใน 4 ของเส้นทางจากจุดหมายต่างๆ ภายในเมือง และนี่คือสิ่งที่เราพบ



รูปที่2.23 รูปแสดงเส้นทางที่ใกล้เคียงกันของจุดหมายต่างๆสำหรับผู้ใช้ 16% ของผู้ใช้งาน Uber

และพวกเรายังให้ความสนใจไปยังประชาชนที่บริการ Uber และพบว่าส่วนใหญ่ของผู้ใช้งานจากหนึ่งในสี่ของผู้ใช้งานทั้งหมดในชั่วโมงเร่งด่วน พบว่า 22% ของการเดินทางในลอสแอนเจลิส นั้นมี 1 ใน 4 ของจำนวนทั้งหมดเกิดขึ้นในระหว่าง 7.00 - 10.00 น. และ 16.00 - 19.00 น. ของวันจันทร์ถึงวันศุกร์

ด้วยราคาที่มีอัตราต่ำกว่ารถ Taxi และอุปสรรคทางเศรษฐกิจที่จะสามารถเดินทางอย่างสะดวก และปลอดภัยทุกเวลา และทุกเส้นทาง ซึ่งนั่นก็หมายความว่ามีประชาชนส่วนน้อยที่ต้องการใช้รถส่วนตัว แต่ไม่สามารถที่จะจ่ายได้ และเลือกที่จะใช้รถบริการสาธารณะแทน

พวกเราเริ่มให้บริการมายาวนานตั้งแต่ปี 2012 และตื่นเต้นเสมอเมื่อมีบริการใหม่ๆ ในลอสแอนเจลิส พวกเราให้บริการ uberPOOL ที่กำลังเชื่อมโยงผู้ขับขี่แต่ละรายเข้าด้วยกัน เพื่อลดความแออัดลง สามารถเข้าใกล้กันได้มากขึ้นจากจุดเริ่มต้นจนถึงปลายทาง และทำให้การเดินทางมีความปลอดภัยและน่าเชื่อถือมากยิ่งขึ้นกว่าที่เคยเป็น Uber มีความมุ่งมั่นที่จะบริการให้ดีมากขึ้น และเข้าถึงหลายๆ ปลายทางมากขึ้นในหลายๆ พื้นที่ของลอสแอนเจลิสโดยการเพิ่มจำนวนผู้โดยสาร และพนักงานขับรถให้มากขึ้นในปีที่กำลังจะมาถึง

มุมมองที่ 2: มุมมองด้านความปลอดภัยจากนักพัฒนาระบบ (https://developer.uber.com)

จากมุมมองในด้านของนักพัฒนา เมื่อเรามาพิจารณาไปยังข้อมูลที่ Uber ต้องการใช้ เก็บและเข้าถึงความเป็นส่วนตัวของผู้ที่มีส่วนเกี่ยวข้อง(ทั้งผู้โดยสาร และพนักงานขับรถ) นั้น ข้อมูลใดที่เป็นเหตุให้เกิดการละเมิดข้อมูลอันเป็นส่วนบุคคลโดยชอบธรรมจากการขอรับบริการ Uber

เมื่อเข้าไปดูที่ Parameter Categories ซึ่งเป็นหัวข้อที่กล่าวถึงฟิลด์ต่างๆ ที่ Uber ต้องการเก็บเพื่อให้บริการผู้โดยสารได้อย่างปลอดภัย และมีความน่าเชื่อถือ พบว่ามีฟิลด์จำนวนมาก และหลากหลายฟิลด์ในหลาย Categories นั้นมีการเข้าถึง และบันทึกข้อมูลที่มีความเป็นส่วนตัวอยู่ตลอดเวลา

Categories/ Functions	Name	Type	Parameter
Endpoint	latitude	float	Latitude component of location.
Endpoint	longtitude	float	Longitude component of location
Product type	product_id	string	Unique identifier representing a specific product for a given latitude & longitude. For example, uberX in San Francisco will have a different product_id than uberX in Los Angeles.
Product type	description	string	Description of production.

Product type	display_name	string	Display name of product.
Product type	capacity	int	Capacity of product, for example, 4 people.
Product type	image	string	Image URL representing the product.
Price Estimates	start_latitude	float	Latitude component of start location.
Price Estimates	start_longtitude	float	Longitude component of start location.
Price Estimates	end_latitude	float	Latitude component of end location.
Price Estimates	end_longitude	float	Longitude component of end location.
Price Estimates	product_id	string	Unique identifier representing a specific product for a given latitude & longitude. For example, uberX in San Francisco will have a different product_id than uberX in Los Angeles.
Price Estimates	currency_code	string	ISO 4217 currency code.
Price Estimates	display_name	string	Display name of product.
Price Estimates	estimate	string	Formatted string of estimate in local currency of the start location. Estimate could be range, a

			single number(flat rate) or "Metered" for Taxi.
Price Estimates	low_estimate	int	Lower bound of the estimated price.
Price Estimates	high_estimate	int	Upper bound of the estimated price.
Price Estimates	super_multiplier	float	Expected surge multiplier. Surge is active if surge_multiplier is greater than 1. Price estimate already factors in the surge multiplier.
Price Estimates	duration	int	Expected activity duration (in seconds). Always show duration in minutes.
Price Estimates	distance	float	Expected activity distance (in miles).
Time Estimates	start_latitude	float	Latitude component.
Time Estimates	start_longtitude	float	Longitude component.
Time Estimates	customer_uuid (optional)	string	Unique customer identifier to be used for experience customization.
Time Estimates	product_id (optional)	string	Unique identifier representing a specific

			product for a given latitude & longitude.
Time Estimates	display_name	string	Display name of product.
Time Estimates	estimate	int	ETA for the product (in seconds). Always show estimate in minutes.
Promotions	start_latitude	float	Latitude component of start location.
Promotions	start_longtitude	float	Longitude component of start location.
Promotions	end_latitude	float	Latitude component of end location.
Promotions	end_longitude	float	Longitude component of end location.
Promotions	display_text	string	A localized string we recommend to use when offering the promotion to users.
Promotions	localized_value	string	The value of the promotion that is available to a user in this location in the local currency.
Promotions	type	string	The type of the promo which is either “trip_credit” or “account_credit”.

User Activity (history)	offset	integer	Offset the list of returned results by this amount. Default is zero. Position in pagination.
User Activity (history)	limit	integer	Number of items to retrieve. Default is 5, maximum is 50.
User Activity (history)	count	integer	Total number of items available.
User Activity (history)	uuid	string	Unique activity identifier.
User Activity (history)	request_time	integer	Unix timestamp of activity request time.
User Activity (history)	product_id	string	Unique identifier representing a specific product for given latitude & longitude. For example, uberX in San Francisco will have a different product_id than uberX in Los Angles.
User Activity (history)	status	string	Status of the activity. Only returns completed for now.
User Activity (history)	distance	float	Length of activity in miles.
User Activity (history)	start_time	integer	Unix timestamp of activity start time.

User Activity (history)	end_time	integer	Unix timestamp of activity end time.
User Profile	first_name	string	First name of the Uber user.
User Profile	last_name	string	Last name of the Uber user.
User Profile	email	string	Email address of the Uber user.
User Profile	picture	string	Image URL of the Uber user.
User Profile	promo_code	string	Promo code of the Uber user.
User Profile	uuid	string	Unique identifier of the Uber user.
Request	product_id	string	The unique ID of the product being requested.
Request	start_latitude	float	The beginning or "pickup" latitude.
Request	start_longtitude	float	The beginning or "pickup" longitude.
Request	end_latitude	float	The final or destination latitude.
Request	end_longitude	float	The final or destination longitude.

Request	surge_confirmation_id (optional)	string	The unique identifier of the surge session for a user. Required when returned from a 409 Conflict response on previous POST attempt.
Request	request_id	string	The unique ID of the Request.
Request	status	string	The status of the Request indicating state.
Request	vehicle	object	The object that contains vehicle details.
Request	driver	object	The object that contains driver details.
Request	location	object	The object that contains the location information of the vehicle and driver.
Request	eta	integer	The estimated time of vehicle arrival in minutes.
Request	surge_multiplier	float	The surge pricing multiplier used to calculate the increased price of a Request. A multiplier of 1.0 means surge pricing is not in effect.

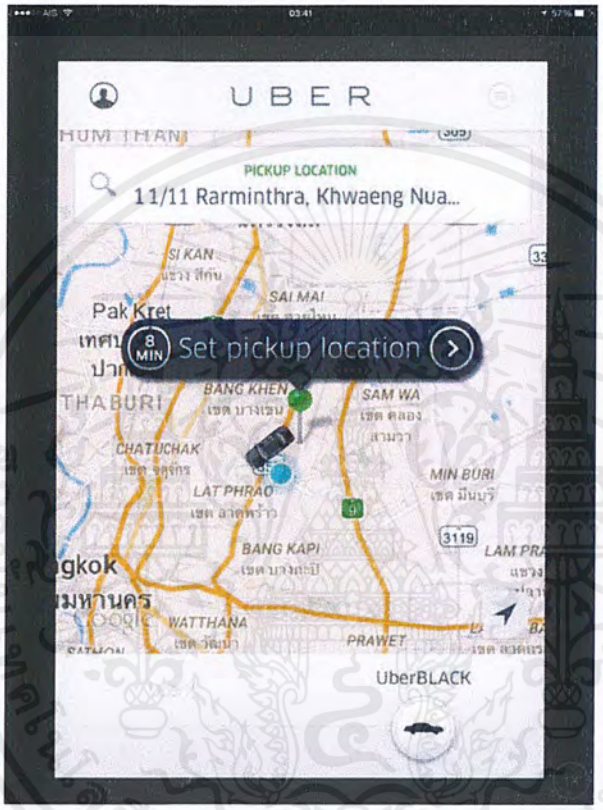
Request (error response)	surge_confirmaiton_id	string	The surge confirmation identifier used to make a Request after a user has accepted surge pricing.
Request (error response)	href	string	The URL a user must visit to accept surge pricing.
Request - Details	request_id	string	Unique identifier representing a Request.
Request - Details	status	string	The status of the Request indicating state.
Request - Details	vehicle	object	The object that contains vehicle details.
Request - Details	vehicle.make	string	The vehicle make of brand.
Request - Details	vehicle.model	string	The vehicle make of type.
Request - Details	vehicle.license_plate	string	The license plate number of the vehicle.
Request - Details	driver	object	The object that contains driver details.
Request - Details	driver.phone_number	string	The formatted phone number for contacting the driver.
Request - Details	driver.rating	float	The driver's star rating out of 5 stars.
Request - Details	driver.picture_url	string	The URL to the photo of the driver.

Request - Details	driver.name	string	The first name of the driver.
Request - Details	location	object	The object that contains the location information of the vehicle and driver.
Request - Details	location.latitude	float	The current latitude of the vehicle.
Request - Details	location.longitude	float	The current longitude of the vehicle.
Request - Details	eta	integer	The estimated time of vehicle arrival in minutes.
Request - Details	surge_multiplier	float	The surge pricing multiplier used to calculate the increased price of a Request. A multiplier of 1.0 means surge pricing is not in effect.
Request - Cancel	request_id	string	Unique identifier representing a Request.
Request - Map	request_id	string	Unique identifier representing a Request.

ตารางที่ 2.10 ตารางแสดงQuery Parameter ของ Uber

จากความเป็นจริงแล้วก่อนการใช้แอพพลิเคชั่นของ Uber (อ้างอิงจากระบบปฏิบัติการ iOS) พบว่าเมื่อทำการติดตั้งแอพพลิเคชั่นเป็นที่เรียบร้อยแล้ว ครั้งแรกที่เข้าใช้งานแอพพลิเคชั่นนั้น จะมีการถามจากแอพพลิเคชั่นว่าผู้ใช้งานจะอนุญาตให้ Uber นั้นสามารถเข้าถึงข้อมูลของผู้ใช้งานได้

หรือไม่ โดยข้อมูลที่สำคัญดังกล่าวได้แก่การแบ่งปันที่อยู่ปัจจุบันให้กับ Uber (current Location Service: Allow) ทำให้ Uber สามารถเข้าถึงตำแหน่งที่อยู่ปัจจุบันของผู้ใช้งานได้อย่างแม่นยำ ซึ่งสามารถดูได้จากฟิลด์ที่อยู่ภายใน Categories เช่น User Account, User Profile, Price Estimates หรือ Request ในรูปแบบต่างๆ เป็นต้น รวมไปถึงฟิลด์บางส่วนใน Time Estimates ด้วยเช่นกัน



รูปที่2.10 รูปแสดงผลของการรับส่งข้อมูลแบบ Real Time

มุมมองที่ 3: มุมมองของผู้ใช้งานทั่วไป

ในส่วนนี้จะกล่าวถึงส่วนที่เกี่ยวข้องกับผู้ใช้งานทั่วไป และคำสำคัญคือ “Big Data Company” โดยอ้างอิงจากบทความของ Ron Hirson ที่ชื่อว่า “Uber: The Big Data Company” โดยในเนื้อหาได้กล่าวถึงการใช้บริการของ Uber และการใช้บริการของโรงแรมในเครือข่ายของ Starwoods ซึ่งผู้ที่เขียนบทความเองเป็นสมาชิกของทั้งคู่ และมีการใช้บริการอย่างสม่ำเสมอ

Ron ค้นพบว่าสิ่งที่เกิดขึ้นกับเขาคือการได้รับข้อเสนอพิเศษจากบริการของทั้งสองโดยอัตโนมัติทันทีที่ได้แจ้งขอรับบริการจาก Uber และ Uber รู้โดยตลอดว่าเขาจะเดินทางไปที่ไหน และเมื่อใด โดยสามารถที่จะมีรถมารอที่สนามบินปลายทางก่อนที่เที่ยวบินจะไปถึง และไปส่งยังโรงแรมปลายทาง ที่เขาวางแผนจะเข้าพัก ซึ่งมันเป็นบริการที่เขาหวังว่าจะได้รับโดยไม่คาดคิด นั่นหมายความ

ว่าเขาได้รับความสะดวกมากขึ้น แต่ก็เป็นความจริงที่ปฏิเสธไม่ได้ว่าข้อมูลการเดินทางของเขาถูกส่งต่อให้กันระหว่างทั้งสองบริษัท Ron พอใจกับการบริการ แต่นั่นก็หมายความว่าการเดินทางของเขาไม่เป็นความลับอีกต่อไป

Big Data Company ทั้งสองนั้นมีการติดต่อระหว่างกันบางอย่าง ทำให้มีการแบ่งปันข้อมูลระหว่างกัน ซึ่งในอนาคตนั้นจะมีการกระทำในลักษณะนี้เพิ่มขึ้น และเพิ่มขึ้นเรื่อยๆ เพื่อความสะดวกของบริการสาธารณะ นั่นก็หมายความว่าข้อมูลส่วนบุคคลบางอย่างของผู้รับบริการนั้นก็คงจะกลายเป็นสาธารณะเช่นกัน

บทที่ 3
การออกแบบและพัฒนา

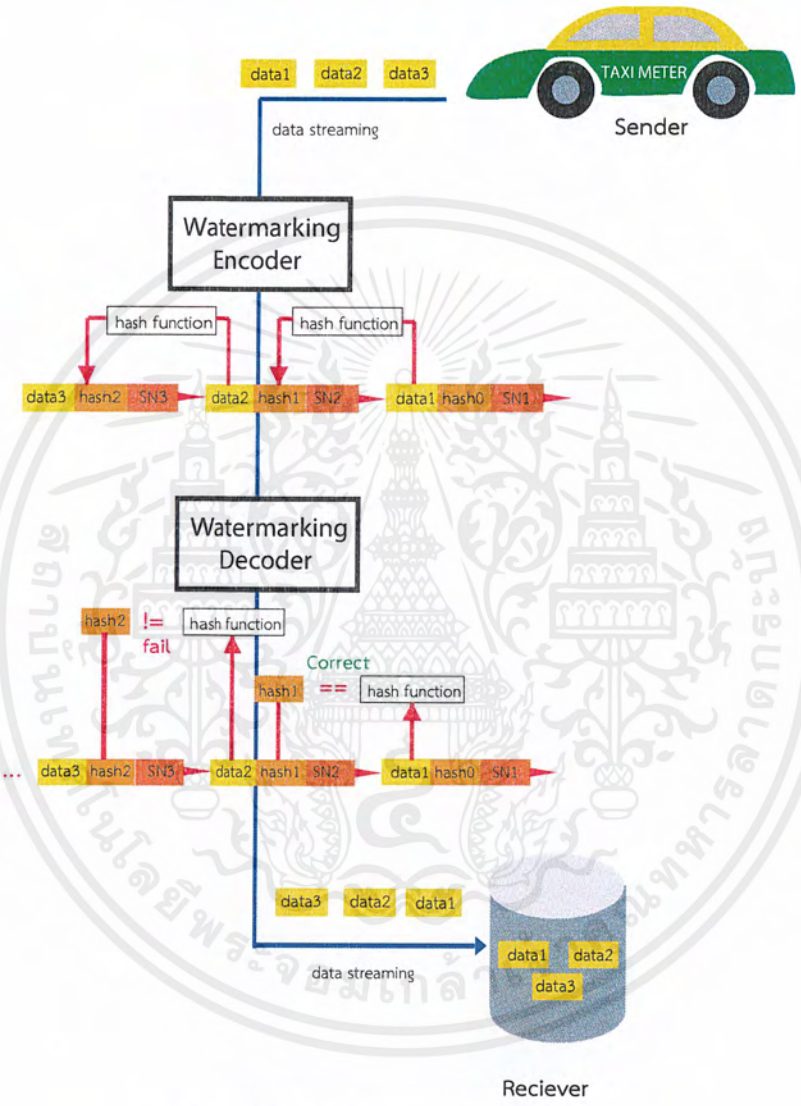
ในการพัฒนา และออกแบบระบบเพื่อการรักษาความปลอดภัยของบิ๊กดาตาโดยมุ่งเน้นไปที่การรักษาความถูกต้องครบถ้วนของข้อมูลที่ถูกส่งจากผู้ส่งสารไปยังผู้รับสาร เพื่อให้ขั้นตอนของการศึกษา และการพัฒนาระบบเพื่อการทดลองเป็นไปอย่างราบรื่น ซึ่งในเนื้อหาภายในบทนี้จะอธิบายถึงรายละเอียด และขั้นตอนในการศึกษา ออกแบบ พัฒนา และทดลองการทำงาน รวมไปถึงโครงสร้างหลักของระบบ

3.1 แนวคิดในการพัฒนา

โครงงานชิ้นนี้จะมีลักษณะเน้นไปที่การศึกษา เพื่อหาวิถีทางที่เหมาะสมกับการรักษาความปลอดภัยของระบบบริหารจัดการบิ๊กดาตา ทางคณะผู้จัดทำจึงเริ่มต้นที่การค้นหาข้อมูล เพื่อทำความเข้าใจกับเทคโนโลยีในปัจจุบันที่กำลังเป็นที่นิยมในการประยุกต์ใช้กับกระบวนการบริหารจัดการของบิ๊กดาตา ในโครงงานชิ้นนี้จึงมุ่งเป้าหมายเริ่มต้นไปที่การรักษาความถูกต้องครบถ้วนของข้อมูลที่มีการส่งตลอดเวลา และมีการไหลของข้อมูลเข้ามาในระบบอย่างต่อเนื่อง(Real-time streaming data) จึงเลือกวัตถุประสงค์ และกำหนดขอบเขตของการศึกษาในการศึกษาปัญหา และหาวิธีในการเพิ่มประสิทธิภาพของการรักษาความถูกต้องครบถ้วนของข้อมูลที่เข้ามาสู่ระบบบริหารจัดการบิ๊กดาตา โดยเริ่มต้นที่การใช้คำสำคัญในการค้นหาต่างๆ จนมาถึงคำว่า “Keyless” และ “Watermerking” ตามลำดับ

โครงสร้างหลักของระบบประกอบไปด้วยสามส่วน ได้แก่ ส่วนที่หนึ่งคือโปรแกรมเพื่อจำลองการทำงานของการสร้างลายน้ำของข้อมูลซึ่งในที่นี้จะแบ่งเป็นสองส่วนย่อยคือโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำ และโปรแกรมถอดรหัสเพื่อการตรวจสอบ เพื่อความเข้าใจในโครงงานนี้ต่อไปนี้จะขอเรียกว่า Watermarking Encoder และ Watermarking Decoder ตามลำดับ ส่วนที่สองคือโปรแกรมเพื่อใช้ติดต่อกับเครื่องแม่ข่ายจำลอง และส่วนที่สามคือระบบจำลองเครื่องแม่ข่ายที่มีการติดตั้งบริการต่างๆ ที่ใช้สำหรับกระบวนการบริหารจัดการบิ๊กดาตา โดยเครื่องมือที่ใช้สำหรับพัฒนาส่วนที่หนึ่งคือโปรแกรม Pycharm IDE ซึ่งภาษาที่ใช้ในการพัฒนาคือ Python 2.7.9 เครื่องมือสำหรับใช้พัฒนาส่วนที่สองคือโปรแกรม Eclipse ซึ่งภาษาที่ใช้ในการพัฒนาคือ Java และเครื่องมือที่ใช้สำหรับส่วนที่สามคือการสร้างสิ่งแวดล้อมต่างๆ เพื่อติดตั้งเครื่องแม่ข่ายจำลอง โปรแกรมที่ใช้คือ Vmware Fusion 6 ที่ทำงานบนระบบปฏิบัติการ OS X รุ่น 10.10.3

สำหรับโครงงานนี้ใช้สิ่งแวดล้อมของเครื่องแม่ข่ายสำเร็จรูปจาก Cloudera โดยใช้ผลิตภัณฑ์ชื่อ QuickStart VMs for CDH 5.4.x ซึ่งภายในผลิตภัณฑ์นี้ประกอบไปด้วย Apache Hadoop cluster ชนิดโหนดเดี่ยว, ตัวอย่างข้อมูล และ Cloudera Manager เป็นที่เรียบร้อยแล้ว



รูปที่ 3.1 โครงสร้างโดยรวมของระบบทดลอง

3.2 แนวคิดในการพัฒนาโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำ

สำหรับแนวคิดในการพัฒนาโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำ โปรแกรมจะทำหน้าที่สร้างสิ่งแวดล้อมจำลองเพื่อทดลองอัลกอริทึมในการสร้างลายน้ำให้กับข้อมูลที่ส่งตลอดเวลาและมีการไหลของข้อมูลอย่างต่อเนื่อง แล้วจึงนำข้อมูลไปเข้าฟังก์ชั่นแฮชที่มีความปลอดภัย จากนั้นจึงสร้างลายน้ำ

ให้กับข้อมูลชนิดที่นำค่าแฮชไปกับข้อมูลกลุ่มถัดไปที่ต่อเนื่องกัน(Forward-Chaining Watermarking Embedding Process) จากนั้นจึงนำข้อมูลที่จะส่งเขียนเป็นไฟล์เพื่อสมมติการส่งข้อมูลลักษณะการฝากข้อความ(Message Passing) เพื่อให้โปรแกรมถอดรหัสเพื่อการตรวจสอบมาทำการอ่านไฟล์และดำเนินกระบวนการต่างๆ ต่อไป โดยโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำมีขั้นตอนต่างๆ ในการทำงานดังต่อไปนี้

1. สร้างข้อมูลจำลองเพื่อใช้เป็นข้อมูลในการทดลอง สร้างข้อมูลที่ส่งตลอดเวลา และมีการไหลอย่างต่อเนื่อง
2. มีการจำลองบัฟเฟอร์เพื่อใช้ในพักข้อมูลระหว่างการส่ง และการสร้างลายน้ำ
3. มีการใช้ฟังก์ชั่นแฮชที่ปลอดภัย นำค่าแฮชที่ได้มาใช้ในการสร้างลายน้ำของข้อมูล ซึ่งในที่นี้ใช้ SHA-512
4. สร้างลายน้ำของข้อมูล
5. นำข้อมูลที่ได้จากการสร้างลายน้ำ เขียนเป็นไฟล์โดยกำหนดให้หนึ่งไฟล์จำลองเป็นหนึ่งแพ็คเกตที่ประกอบไปด้วยส่วนหัว(Header) และข้อมูล(Payload)

3.3 ลำดับการทำงานของโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำ

1. โปรแกรมจะเริ่มต้นด้วยส่วนที่ส่งข้อมูลตัวอย่างที่จำลองด้วยการใช้ไลบารี่ Fake-Factory ในการสร้างข้อมูลแบบส่งตลอดเวลาและมีการไหลของข้อมูลอย่างต่อเนื่อง โดยสร้างข้อมูลบางอย่างเพื่อยกตัวอย่างข้อมูลให้สอดคล้องกับกรณีศึกษาตัวอย่าง การใช้งานไลบารี่ สามารถเรียกใช้งานได้ดังต่อไปนี้

```
from faker import Faker
fake = Faker()

names1 = fake.name()
addr1  = fake.street_address()
lat1   = fake.latitude()
long1  = fake.longitude()
phone1 = fake.phone_number()

names2 = fake.name()
addr2  = fake.street_address()
```

```
lat2      = fake.latitude()
long2     = fake.longitude()
phone2    = fake.phone_number()

block = [names1, phone1, addr1, lat1, long1]
block2 = [names2, phone2, addr2, lat2, long2]
```

2. มีการสร้างบัฟเฟอร์จำลองเพื่อใช้ในการพักข้อมูลระหว่างการส่ง และการสร้างลายน้ำของข้อมูล โดยเมื่อนำข้อมูลที่สร้างขึ้นมา เมื่อนำเข้าสู่ฟังก์ชั่นการทำงานแล้วจึงทำการเปลี่ยนชื่อตัวแปรแทนการสร้างบัฟเฟอร์ เมื่อรับค่าเข้ามาในฟังก์ชั่นจึงเปลี่ยนข้อมูลเป็นข้อความ (String) โดยกำหนดการทำงานดังนี้

```
pblock = str(block)
currentBlock = str(block2)
```

3. มีการใช้ฟังก์ชั่นแฮชที่ปลอดภัย โดยในโครงงานนี้ใช้ SHA-512 ซึ่งสามารถเรียกใช้จากไลบารี่ hashlib และจึงนำข้อมูลที่ต้องการสร้างลายน้ำมาเข้าฟังก์ชั่นแฮชที่ปลอดภัย และนำค่าใส่ไว้ในตัวแปร

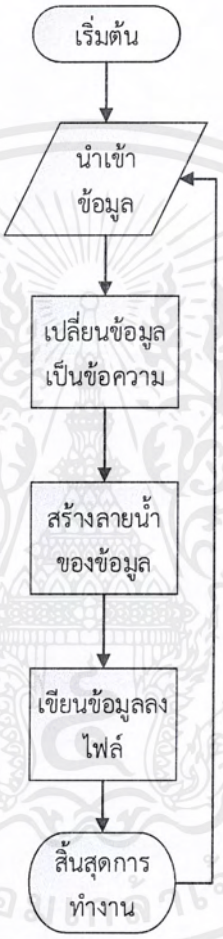
```
m  = hashlib.sha512(pblock)
m2 = m.digest()
```

4. ขั้นตอนในการสร้างลายน้ำของข้อมูล คือการนำค่าแฮชที่ได้จากข้อที่ 3 มาทำการจัดเรียงตำแหน่งเพื่อวางค่าแฮชที่เก็บไว้ในลักษณะของส่วนหัวของแพ็คเกต จากนั้นจึงนำข้อมูลชุดถัดไปที่พักรออยู่ในบัฟเฟอร์ เพื่อเตรียมใช้ในการสร้างแพ็คเกต (อ้างอิงตัวแปร m2 มาจากข้อที่ 3) โดยในตัวอย่างส่วนของโปรแกรมที่ยกมานั้นมีการกำหนดตัวเลขแสดงลำดับ(Sequence Number - sn) ซึ่งส่วนของโปรแกรมดังกล่าวมีแสดงตัวเลขโดยการใช้ตัวเลขที่แสดงด้วยเลขฐานสองจำนวน 64 ไบต์

```
m3 = '{0:064b}'.format(sn) + m2 + currentBlock
```

5. นำข้อมูลที่ได้จากการสร้างลายน้ำ และเขียนไฟล์โดยกำหนดให้หนึ่งไฟล์จำลองเป็นหนึ่งแพ็คเกตที่ประกอบไปด้วยส่วนหัว(Header) และข้อมูล(Payload) โดยการเขียนไฟล์จะเขียนลงไปในโฟลเดอร์ที่สร้างเอาไว้จำลองการเป็นกล่องข้อความเพื่อให้มีลักษณะการติดต่อแบบการฝากข้อความ (Message Passing) โดยดังแสดงด้วยส่วนของโปรแกรมต่อไปนี้(อ้างอิงตัวแปร m3 จากข้อที่ 4)

```
filepath = str(sn)+".txt"
file = open(filepath, "wb+")
with file:
file.write(m3)
```



รูปที่ 3.2 แผนผังแสดงการทำงานของโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำ

3.4 แนวคิดในการพัฒนาโปรแกรมถอดรหัสเพื่อการตรวจสอบ

แนวคิดในการพัฒนาโปรแกรมถอดรหัสเพื่อการตรวจสอบ โปรแกรมจะทำหน้าที่สร้างสิ่งแวดล้อมจำลองเพื่อทดลองอัลกอริทึมในการตรวจสอบลายน้ำของข้อมูลที่รับตลอดเวลาและมีการไหลเข้าของข้อมูลอย่างต่อเนื่อง แล้วจึงนำข้อมูลที่รับเข้ามาไปทำการแบ่งข้อมูลออกเป็นส่วนๆ คือการแยกค่าแฮชออกจากส่วนของข้อมูลดั้งเดิมที่มากับแพ็คเกตที่รับเข้ามา ส่วนที่หนึ่งคือส่วนของข้อมูล

ดั้งเดิมจะนำไปเข้าฟังก์ชั่นแฮชที่ปลอดภัย เพื่อนำค่าที่ได้จากการแฮชมาเปรียบเทียบกับค่าแฮชที่ถูกแยกออกมาจากข้อมูลที่ได้รับเข้ามาในตอนต้น เพื่อตรวจสอบว่าข้อมูลทั้งหมดในแพ็คเกตที่ได้รับเข้ามานั้น ถูกเปลี่ยนแปลงแก้ไขในระหว่างช่องทางการส่งข้อมูลหรือไม่ จากนั้นจึงนำข้อมูลที่ได้รับการตรวจสอบแล้ว เขียนเป็นไฟล์เพื่อสมมติการส่งข้อมูลลักษณะการฝากข้อความ(Message Passing) เพื่อให้โปรแกรมที่ใช้ในการติดต่อกับเครื่องแม่ข่าย มาทำการอ่านไฟล์และดำเนินกระบวนการต่างๆ ต่อไป โดยโปรแกรมถอดรหัสเพื่อการตรวจสอบมีขั้นตอนต่างๆ ในการทำงานดังต่อไปนี้

1. รับข้อมูลที่ส่งตลอดเวลา และมีการไหลอย่างต่อเนื่อง เข้ามาเก็บในบัฟเฟอร์ที่เตรียมไว้
2. แบ่งแยกข้อมูลที่ได้รับเข้ามา ออกเป็นส่วนๆ โดยแยกเก็บระหว่างค่าแฮช และข้อมูลดั้งเดิมที่อยู่ในข้อมูลที่รับเข้ามา
3. นำข้อมูลดั้งเดิมที่อยู่ภายในข้อมูลที่รับเข้ามาที่แยกเก็บไว้มาเข้าฟังก์ชั่นแฮชที่ปลอดภัย (SHA-512)
4. นำค่าแฮชที่ได้ มาเทียบกับค่าแฮชที่แยกเก็บไว้จากข้อมูลที่รับเข้ามาเป็นลำดับถัดไป เมื่อคำค่าแฮชทั้งสองส่วนมาเทียบกัน ถ้าค่าแฮชที่ได้มีค่าเท่ากันทุกประการ จึงถือว่าข้อมูลที่รับเข้ามามีความครบถ้วนถูกต้อง
5. บันทึกข้อมูลที่ได้รับการตรวจสอบความถูกต้องแล้ว โดยวิธีการเขียนไฟล์ เพื่อเตรียมให้โปรแกรมเพื่อใช้ติดต่อเครื่องแม่ข่ายมาทำการนำเข้าไปเก็บในหน่วยบันทึกผลที่เครื่องแม่ข่าย

3.5 ลำดับการทำงานของโปรแกรมถอดรหัสเพื่อการตรวจสอบ

1. โปรแกรมจะรับข้อมูลที่ถูกส่งเข้ามาในระบบตลอดเวลา และมีการไหลอย่างต่อเนื่อง เข้ามาเก็บในบัฟเฟอร์ที่เตรียมไว้ ซึ่งที่นี้คือการอ่านไฟล์มาจากโฟลเดอร์ที่เก็บไฟล์ที่ภายในประกอบไปด้วยข้อมูลดั้งเดิม และค่าแฮช ดังแสดงส่วนของโปรแกรมดังนี้

```
partInput = str(sn)+".txt"
file = open("Hashed_file/"+partInput, “rb")
filesize = os.stat("Hashed_file/"+partInput).st_size

    with file:
    .
    .
    .
    file.close()
```

2. แบ่งแยกข้อมูลที่ได้เข้ามา ออกเป็นส่วนๆ โดยแยกออกเป็นสามส่วนได้แก่ เลขบอกลำดับ, ค่าแฮช และข้อมูลดั้งเดิม ซึ่งในที่นี้สมมติคือ fblock, sblock และ currentblock ตามลำดับ ดังแสดงส่วนของโปรแกรมดังนี้

```
fblock = file.read(64)
sblock = file.read(64)
currentBlock = file.read(filesize-128)
```

3. นำข้อมูลดั้งเดิมที่อยู่ภายในข้อมูลที่รับเข้ามา ซึ่งในที่นี้คือ currentblock มาเข้าฟังก์ชั่นที่ปลอดภัย ซึ่งเป็นฟังก์ชั่นเดียวกันกับโปรแกรมเข้ารหัสเพื่อการสร้างลายมือ สำหรับโครงงานนี้ฟังก์ชั่นแฮชที่ปลอดภัยที่ใช้คือ SHA-512 จากการใช้ไลบารี่ hashlib ดังแสดงส่วนของโปรแกรมดังนี้

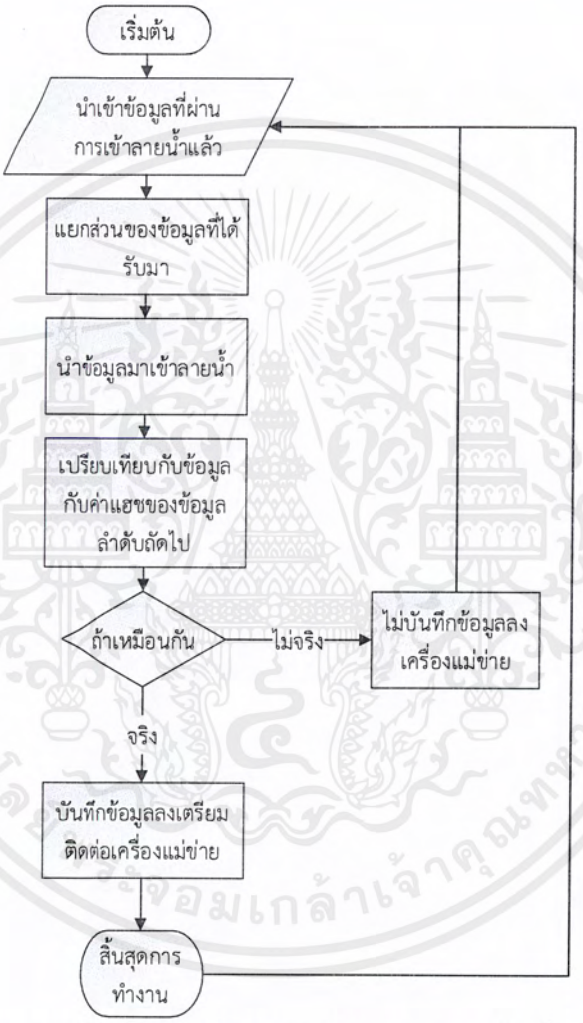
```
block = str(currentBlock)
m = hashlib.sha512(block)
m2 = m.digest()
```

4. นำค่าแฮชที่ได้ มาเทียบกับค่าแฮชที่แยกเก็บไว้จากข้อมูลที่รับเข้ามาเป็นลำดับถัดไป เมื่อนำค่าแฮชทั้งสองส่วนมาเทียบกัน ถ้าค่าแฮชที่ได้มีค่าเท่ากันทุกประการ จึงถือว่าข้อมูลที่รับเข้ามามีความครบถ้วนถูกต้อง ดังแสดงส่วนของโปรแกรมดังนี้

```
if (sblock == m2):
        print "True"
        .
        .
        .
        elif (not(sblock) ):
        print “Checked"
else:
        print “Hacked"
```

5. บันทึกข้อมูลที่ได้รับการตรวจสอบความถูกต้องแล้ว โดยวิธีการเขียนไฟล์ เพื่อเตรียมให้โปรแกรมเพื่อใช้ติดต่อเครื่องแม่ข่ายมาทำการนำเข้าไปเก็บในหน่วยบันทึกผลที่เครื่องแม่ข่ายเป็นลำดับต่อไป ดังแสดงส่วนของโปรแกรมดังนี้ ซึ่งส่วนของโปรแกรมที่แสดงนั้นจะอยู่ทำงานร่วมกันกับการใช้คำสั่งเงื่อนไขในข้อ 4

```
wfilepath = "True_"+str(sn)+".txt"
wfile = open("hadoop_temp/"+wfilepath, "wb+")
with wfile:
wfile.write(currentBlock)
```



รูปที่ 3.3 แผนผังแสดงการทำงานของโปรแกรมถอดรหัสเพื่อสร้างลายน้ำ

3.6 แนวคิดในการพัฒนาโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่าย

แนวคิดในการพัฒนาโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่าย โปรแกรมจะทำหน้าที่อ่านข้อมูลที่ได้รับการตรวจสอบความถูกต้องครบถ้วนเป็นที่เรียบร้อยแล้ว เข้าไปเก็บในหน่วยบันทึกข้อมูลของเครื่องแม่ข่าย ซึ่งสำหรับโครงงานนี้ เครื่องแม่ข่ายจะใช้เป็นสิ่งแวดล้อมสำเร็จรูปที่มีการตั้งค่ามาเป็นที่

เรียบร้อยแล้ว สามารถทำงานได้บนโปรแกรมที่ทำงานเครื่องเสมือน ซึ่งในโครงงานนี้ใช้โปรแกรม Vmware Fusion 6 ดังที่ได้กล่าวไปแล้วในข้างต้น

การทำงานของการติดต่อกันระหว่างเครื่องผู้ใช้งาน และเครื่องแม่ข่ายในโครงงานนี้จะมีลักษณะการฝากข้อความ(Message Passing) ซึ่งจำลองการติดต่อกันโดยการอ่านไฟล์จากโฟลเดอร์ โดยโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่ายมีขั้นตอนต่างๆ ในการทำงานดังต่อไปนี้

1. อ่านไฟล์ที่ประกอบไปด้วยข้อมูลที่ได้รับการตรวจสอบความถูกต้องครบถ้วนเป็นที่เรียบร้อยแล้ว จากโฟลเดอร์ต้นทาง

2. เมื่อทำเจอไฟล์ที่ประกอบไปด้วยข้อมูลที่ได้รับการตรวจสอบความถูกต้องครบถ้วนเป็นที่เรียบร้อยแล้ว จึงทำการดึงไฟล์ไปเขียนลงในหน่วยบันทึกผลของเครื่องแม่ข่าย

3.7 ลำดับการทำงานของโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่าย

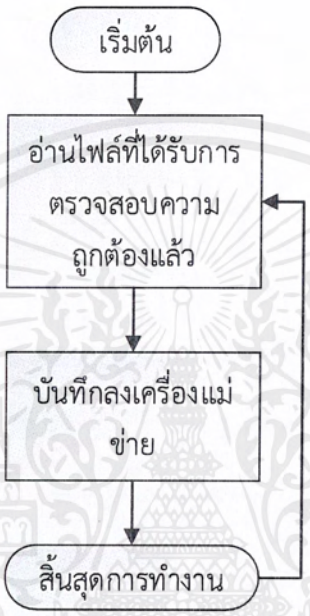
1.อ่านไฟล์ที่ประกอบไปด้วยข้อมูลที่ได้รับการตรวจสอบความถูกต้องครบถ้วนเป็นที่เรียบร้อยแล้ว จากโฟลเดอร์ต้นทาง ในโครงงานนี้โฟลเดอร์ดังกล่าวได้แก่ hadoop_temp เพื่อไปเก็บใน Hadoop File System ที่อยู่ภายในเครื่องแม่ข่าย โดยส่วนของโปรแกรมที่ทำการอ่าน และค้นหาไฟล์มีการทำงานดังแสดงต่อไปนี้

```
{
    File dir = new File("/Users/nuannapha/PycharmProjects/fdw1/hadoop_temp");
    ArrayList<File> files = new ArrayList<File>();
    for(File f : dir.listFiles()) {
        if(f.getName().contains("True"))
            files.add(f);
        else
            f.delete();
}
```

2. เมื่อโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่ายเจอไฟล์ที่ได้รับการตรวจสอบความถูกต้องครบถ้วนเป็นที่เรียบร้อยแล้ว โปรแกรมจะทำการดึงไฟล์ไปเขียนลงในหน่วยบันทึกข้อมูลของเครื่องแม่ข่าย มีการทำงานดังแสดงส่วนของโปรแกรมดังนี้

```
{
    cmd = "curl -i -X PUT -L http://192.168.34.159:50070/webhdfs/v1/user/Test/"
```

```
                + f.getName() +"?op=CREATE -T " + f.getAbsolutePath();
        output = executeCommand(cmd);
        f.delete();
}
```



รูปที่ 3.4 แผนผังแสดงการทำงานของโปรแกรมเพื่อติดต่อกับเครื่องแม่ข่าย

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองในลักษณะต่างๆ ผลการทดลอง รูปแสดงผลการทดลอง และตัวอย่างของรูปผลการทดลอง

4.1 การทดลองการทำงานแบบที่ส่งข้อมูลตลอดเวลา และมีการไหลอย่างต่อเนื่อง (Real-time processing for data streaming)

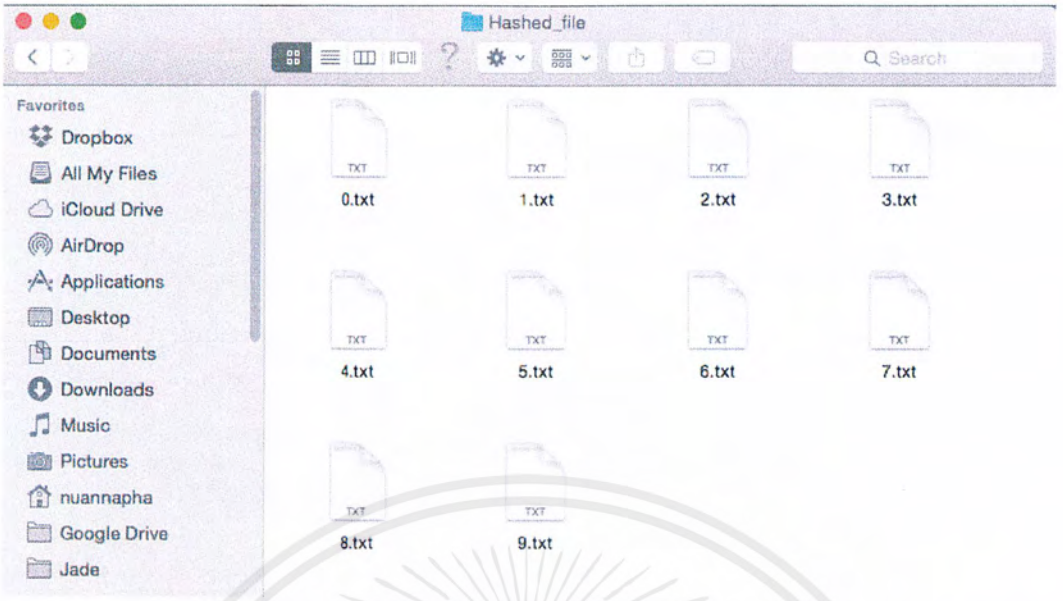
การทดลองนี้มุ่งเน้นไปที่การทดลองสร้างระบบเสมือนจริง เพื่อทดสอบความสอดคล้องกันของการทำงาน ซึ่งทดลองด้วยการทำงานโปรแกรมในลักษณะของการวนซ้ำไม่รู้จบ ทั้งสามโปรแกรม อันได้แก่ โปรแกรมที่หนึ่งคือ โปรแกรมเข้ารหัสเพื่อการสร้างลายน้ำ โปรแกรมที่สองคือ โปรแกรมถอดรหัสเพื่อการตรวจสอบ และโปรแกรมที่สามคือ โปรแกรมเพื่อติดต่อกับเครื่องแม่ข่าย

<u>ตัวอย่าง</u>

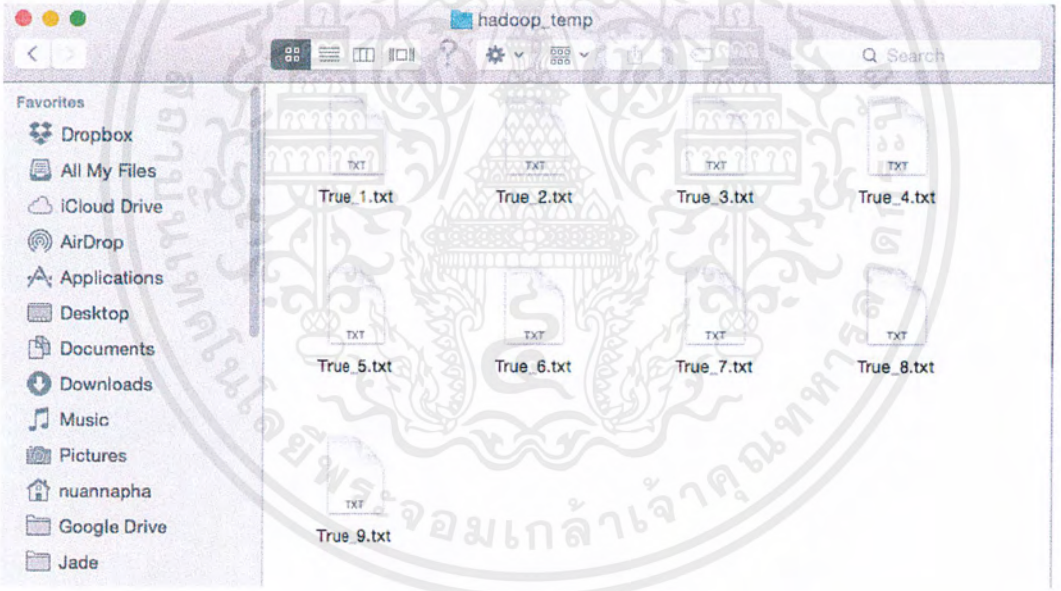
เนื่องจากในการถอดรหัสในเชิงความถี่นี้จะต้องใช้การแปลงไปและแปลงกลับเป็นส่วนสำคัญ นอกเหนือไปจากการคำนวณอื่นๆ การแปลงและการแปลงกลับจะต้องใช้การคำนวณเป็นจำนวนมาก จึงมีการนำวิธีการตัวประกอบปฐม (Prime Factor Algorithm) มาใช้เพื่อลดจำนวนการคำนวณลง โดยใช้ร่วมกับวิธีการแปลงข้อมูลจำนวนน้อยๆ (Short Length Algorithm) ในแง่ของการนำวิธีการดังกล่าวไปใช้งานจริงซึ่งจะต้องพิจารณา

```
-----------------------------------

Hashed Block >>>
0000000000000000000000000000000000000000000000000000000000000001117r(?v3D,???w:H
                                                                                .
l?ɗk?'k??u~??ᵂ???I?"L???l&?^:??J?\??ϟ?[u'Dr. Emett Dicki DVM', u'808.494.1521x25
806', u'59306 Doyle Light', Decimal('17.904844'), Decimal('159.190502')]

Hashed Block in Hexadecimal >>>
3030303030303030303030303030303030303030303030303030303030303030303030303030303
0303030303030303030303030303030303030303030303030313131cb7228bc7633442c109faec9773a48
0c2e6cfb17ca9b6b0e90276b96dd757ef213951dcaac88f2839de94986224cc1bec96c269c5e3aa
df2864a925c9ad4b4d35b752744722e20456d657474204469636b692044564d272c207527383038
2e3439342e31353231783235383036272c207527353933303620446f796c65204c69676874272c2
0446563696d616c282731372e393034383343427292c20446563696d616c28273135392e31393035
303227295d

Hashed Block Size >>>
242

-----------------------------------

Hashed Block >>>
0000000000000000000000000000000000000000000000000000000000001000P0🖃ÃU?^T-[5???
[?$?ω*c
%?
  ?A?(THQEp??-??MK[J?N?[u'Maryam Robel', u'+86(2)0961041680', u'87389 Ashanti C
lub Suite 994', Decimal('62.512019'), Decimal('-35.956591')]

Hashed Block in Hexadecimal >>>
3030303030303030303030303030303030303030303030303030303030303030303030303030303
0303030303030303030303030303030303030303030303130303050dc9230dac38355a55e54031f2d5b
358ab4d45b1f8624bccf890f2a630a25f5140cbd41bf4a082854481b915145d180d6bcf2d02db4c
6194d4b5b4a9c4ef1a25b75274d617279616d20526f62656c272c2075272b383628322930393631
303431363830272c207527383733383920417368616e746920436c756220537569746520393942
72c20446563696d616c282736322e353132303139272c20446563696d616c28272d33352e3935
36353931272 95d

Hashed Block Size >>>
244

-----------------------------------

Hashed Block >>>
0000000000000000000000000000000000000000000000000000000000001001U?🅾p?6?" ?d?O<
?lU?🖃??I
        ?       ??R??H"]L?]?ϟ?[u'Mr. Elliott Jaskolski', u'(848)241-6723x177',
u'3540 Schumm Streets Apt. 906', Decimal('-60.192546'), Decimal('-179.808881')]

Hashed Block in Hexadecimal >>>
3030303030303030303030303030303030303030303030303030303030303030303030303030303
0303030303030303030303030303030303030303030303130303155dee0b6bc8570c136942220e21364
c44f3ced817041d844f8d2913d3c0d106c55a0e1829b1f038095f907490bd409b5f3b052f9db482
25d4c11cc5dcacbab895b75274d722e20456c6c696f7474204a61736b6f6c736b69272c20752728
383438293234312d3637323378313737272c207527333534302053636875 6d6d205374726565747
3204170742e20393036272c20446563696d616c28272d36302e313932353435627292c2044656369
6d616c28272d3137392e383038383831272 95d

Hashed Block Size >>>
256

-----------------------------------
```

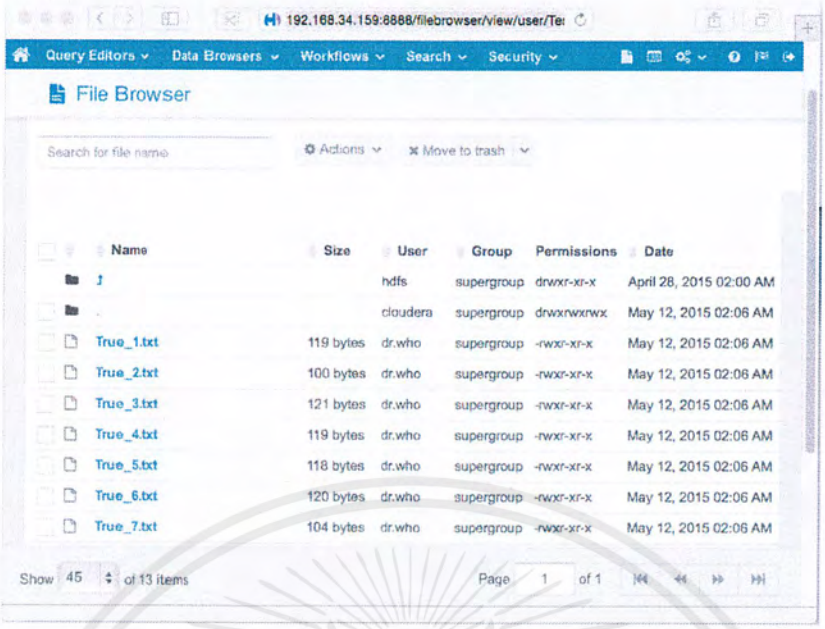
รูปที่ 4.1 ตัวอย่างข้อมูลที่ส่งออกมาจากโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำข้อมูล



รูปที่ 4.2 ตัวอย่างโฟลเดอร์ของข้อมูลที่ส่งออกมาจากโปรแกรมเข้ารหัสเพื่อสร้างลายน้ำข้อมูล



รูปที่ 4.3 ตัวอย่างโฟลเดอร์ข้อมูลที่ส่งออกมาจากโปรแกรมถอดรหัสเพื่อการตรวจสอบ



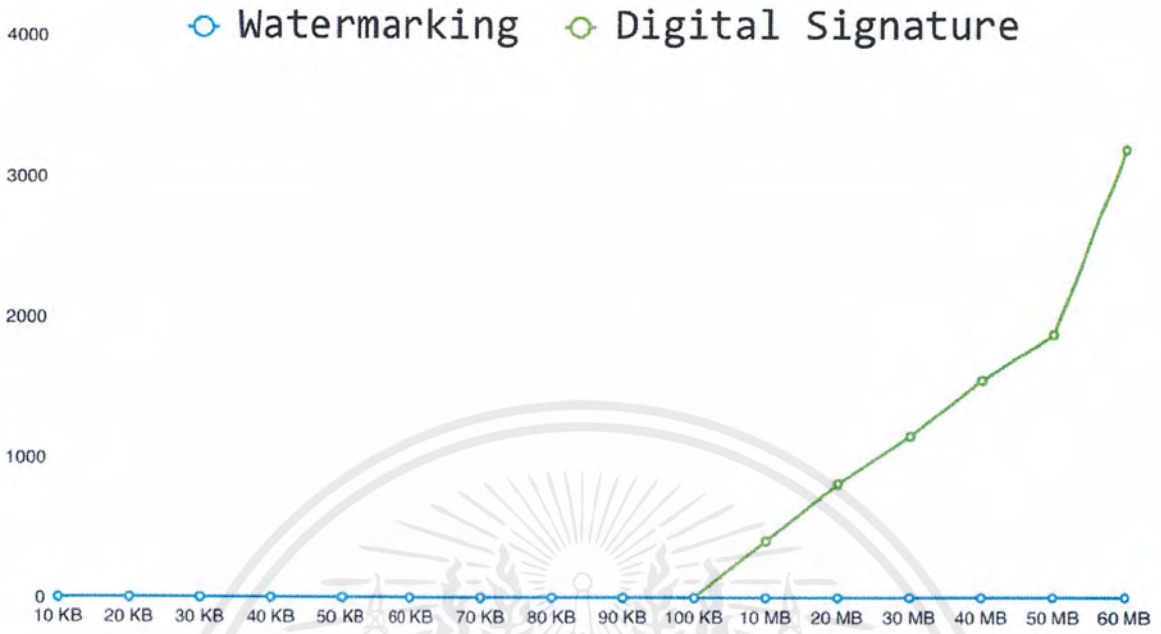
รูปที่ 4.4 ตัวอย่างหน้าต่างเพื่อติดต่อกับฐานข้อมูลของเครื่องแม่ข่าย

4.2 การทดลองการทำงานเปรียบเทียบการทำงานเข้ารหัสที่ไฟล์ขนาดต่างๆ ระหว่างการสร้างลายน้ำข้อมูล และการสร้างลายเซ็นดิจิตัล

การทดลองนี้มุ่งเน้นไปที่การทดลองเพื่อเปรียบเทียบประสิทธิภาพการทำงานของทั้งสองอัลกอริทึมที่ใช้ในการรักษาความถูกต้องควบถ้วนของข้อมูล ได้แก่การสร้างลายน้ำของข้อมูล (Watermarking) และการสร้างลายเซ็นดิจิตัล(Digital Signature) โดยการทดลองทำการจับเวลาเพื่อทดสอบความเร็วในการเข้ารหัสข้อมูลที่ขนาดต่างๆ ได้แก่ 10 KB, 20 KB, 30 KB, 40 KB, 50 KB, 60 KB, 70 KB, 80 KB, 90 KB, 100 KB, 10 MB, 20 MB, 30 MB, 40 MB, 50 MB และ 60 MB โดยที่ขนาดไฟล์หนึ่งๆ จะทำการทดลองซ้ำ 3 ครั้ง เพื่อให้ได้ค่าเฉลี่ยของการทดลอง ซึ่งผลการทดลองได้ผลดังนี้

File Sizes	10 KB	20 KB	30 KB	40 KB	50 KB	60 KB	70 KB	80 KB	90 KB	100 KB	10 MB	20 MB	30 MB	40 MB	50 MB	60 MB
Watermarking	0.00127	0.00162	0.00239	0.02680	0.00523	0.00505	0.00622	0.00739	0.00783	0.03017	0.9957	1.95648	3.27209	4.02052	5.12849	6.66323
	0.00128	0.00164	0.00247	0.00355	0.00418	0.00610	0.00610	0.01017	0.01117	0.00869	1.01357	1.98394	2.95400	4.13558	5.50568	7.07245
	0.001524	0.00167	0.00285	0.00334	0.00623	0.00580	0.00627	0.01231	0.01263	0.01048	1.21930	2.15776	3.09985	4.32815	5.55886	6.99522
Average	0.00136	0.00164	0.00257	0.01123	0.00521	0.00565	0.0062	0.00996	0.01054	0.01645	1.07619	2.03273	3.10865	4.16142	5.39768	6.9103
Digital Signature	1.06634	1.80520	1.65848	2..05791	2.87625	3.16504	3.29852	3.94504	4.22895	4.46202	399.40023	794.01396	1250.71413	1592.29802	1906.17232	3600.69839
	0.92745	1.36224	1.61293	1.96128	2.37212	2.72140	3.08075	3.62341	4.18997	4.34397	412.93264	847.57451	1100.25437	1516.98960	1879.64927	3596.41514
	0.98736	1.53144	1.50749	1.96929	2.52060	3.05744	3.35521	3.39200	4.11708	4.39642	405.56541	800.71261	1109.74833	1548.77803	1856.94310	2386.54570
Average	0.99372	1.56629	1.59297	1.96529	2.58966	2.98129	3.24483	3.65348	4.17867	4.4008	405.96609	814.10036	1153.57228	1552.68855	1880.92156	3194.55308

ตารางที่ 4.1 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพของการทำงานระหว่างการสร้างลายน้ำข้อมูลและการสร้างลายเซ็นดิจิตัลกับที่ขนาดไฟล์ต่างๆ



ตารางที่ 4.2 แผนภูมิเส้นแสดงผลการเปรียบเทียบการทำงานของการสร้างลายน้ำข้อมูล และการสร้างลายเซ็นดิจิตัลที่ขนาดไฟล์ต่างๆ

4.3 การทดลองการทำงานเปรียบเทียบการทำงานเข้ารหัสที่ไฟล์ขนาดต่างๆ ระหว่างการสร้างลายน้ำข้อมูล และการสร้างลายเซ็นดิจิตัล

4.3.1 ผลการทดลองการทำงานของการสร้างลายน้ำข้อมูล

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '10KB.txt', mode 'rb' at 0x1005a08a0>
0.00127005577087
<closed file '10KB.txt', mode 'rb' at 0x1005a0930>
0.00127696990967
<closed file '10KB.txt', mode 'rb' at 0x1005a08a0>
0.00152397155762
```

รูปที่ 4.5 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 10 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '20KB.txt', mode 'rb' at 0x1005a08a0>
0.00162386894226
<closed file '20KB.txt', mode 'rb' at 0x1005a0930>
0.00164389610291
<closed file '20KB.txt', mode 'rb' at 0x1005a08a0>
0.00167393684387
```

รูปที่ 4.6 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 20 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '30KB.txt', mode 'rb' at 0x1004a08a0>
0.00238513946533
<closed file '30KB.txt', mode 'rb' at 0x1004a0930>
0.00247097015381
<closed file '30KB.txt', mode 'rb' at 0x1004a08a0>
0.00285315513611
```

รูปที่ 4.7 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 30 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '40KB.txt', mode 'rb' at 0x1004a08a0>
0.0268020629883
<closed file '40KB.txt', mode 'rb' at 0x1004a0930>
0.00354504585266
<closed file '40KB.txt', mode 'rb' at 0x1004a08a0>
0.00333905220032
```

รูปที่ 4.8 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 40 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '50KB.txt', mode 'rb' at 0x1004a08a0>
0.0052330493927
<closed file '50KB.txt', mode 'rb' at 0x1004a0930>
0.00418400764465
<closed file '50KB.txt', mode 'rb' at 0x1004a08a0>
0.00622582435608
```

รูปที่ 4.9 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 50 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '60KB.txt', mode 'rb' at 0x1005a08a0>
0.00504994392395
<closed file '60KB.txt', mode 'rb' at 0x1005a0930>
0.00609707832336
<closed file '60KB.txt', mode 'rb' at 0x1005a08a0>
0.00580382347107
```

รูปที่ 4.10 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 60 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '70KB.txt', mode 'rb' at 0x1005a08a0>
0.00622200965881
<closed file '70KB.txt', mode 'rb' at 0x1005a0930>
0.00610089302063
<closed file '70KB.txt', mode 'rb' at 0x1005a08a0>
0.00627088546753
```

รูปที่ 4.11 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 70 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '80KB.txt', mode 'rb' at 0x1004a08a0>
0.00738501548767
<closed file '80KB.txt', mode 'rb' at 0x1004a0930>
0.0101721286774
<closed file '80KB.txt', mode 'rb' at 0x1004a08a0>
0.0123128890991
```

รูปที่ 4.12 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 80 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '90KB.txt', mode 'rb' at 0x1005a08a0>
0.00783085823059
<closed file '90KB.txt', mode 'rb' at 0x1005a0930>
0.011167049408
<closed file '90KB.txt', mode 'rb' at 0x1005a08a0>
0.0126311779022
```

รูปที่ 4.13 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 90 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '100KB.txt', mode 'rb' at 0x1004a08a0>
0.0301699638367
<closed file '100KB.txt', mode 'rb' at 0x1004a0930>
0.00869393348694
<closed file '100KB.txt', mode 'rb' at 0x1004a08a0>
0.0104489326477
```

รูปที่ 4.14 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 100 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '10MB.txt', mode 'rb' at 0x1004a08a0>
0.995702981949
<closed file '10MB.txt', mode 'rb' at 0x1004a0930>
1.01356983185
<closed file '10MB.txt', mode 'rb' at 0x1004a08a0>
1.21930003166
```

รูปที่ 4.15 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 10 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '20MB.txt', mode 'rb' at 0x1007a08a0>
1.95648312569
<closed file '20MB.txt', mode 'rb' at 0x1007a0930>
1.98394107819
<closed file '20MB.txt', mode 'rb' at 0x1007a08a0>
2.15776491165
```

รูปที่ 4.16 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 20 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '30MB.txt', mode 'rb' at 0x1005a08a0>
3.27209091187
<closed file '30MB.txt', mode 'rb' at 0x1005a0930>
2.95395588875
<closed file '30MB.txt', mode 'rb' at 0x1005a08a0>
3.0338549614
```

รูปที่ 4.17 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 30 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '40MB.txt', mode 'rb' at 0x1006a08a0>
4.02052688599
<closed file '40MB.txt', mode 'rb' at 0x1006a0930>
4.13557887077
<closed file '40MB.txt', mode 'rb' at 0x1006a08a0>
4.3281481266
```

รูปที่ 4.18 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 40 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '50MB.txt', mode 'rb' at 0x1006a08a0>
5.12849092484
<closed file '50MB.txt', mode 'rb' at 0x1006a0930>
5.50568199158
<closed file '50MB.txt', mode 'rb' at 0x1006a08a0>
5.55886292458
```

รูปที่ 4.19 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 50 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python fwdTest.py
<closed file '60MB.txt', mode 'rb' at 0x1005a08a0>
6.663230896
<closed file '60MB.txt', mode 'rb' at 0x1005a0930>
7.07245016098
<closed file '60MB.txt', mode 'rb' at 0x1005a08a0>
6.995221138
```

รูปที่ 4.20 ผลการทดลองสร้างลายน้ำของข้อมูลที่ไฟล์ขนาด 60 เมกะไบต์

4.3.2 ผลการทดลองการทำงานของการทำลายเซ็นดิจิตัล

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '10KB.txt', mode 'rb' at 0x10282e5d0>
1.06634497643
<closed file '10KB.txt', mode 'rb' at 0x10282e5d0>
0.927457094193
<closed file '10KB.txt', mode 'rb' at 0x10282e5d0>
0.987359046936
```

รูปที่ 4.21 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 10 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '20KB.txt', mode 'rb' at 0x10282e5d0>
1.80519509315
<closed file '20KB.txt', mode 'rb' at 0x10282e5d0>
1.36224102974
<closed file '20KB.txt', mode 'rb' at 0x10282e5d0>
1.53143715858
```

รูปที่ 4.22 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 20 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '30KB.txt', mode 'rb' at 0x10282e5d0>
1.65848517418
<closed file '30KB.txt', mode 'rb' at 0x10282e5d0>
1.61293196678
<closed file '30KB.txt', mode 'rb' at 0x10282e5d0>
1.50749492645
```

รูปที่ 4.23 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 30 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '40KB.txt', mode 'rb' at 0x10282e5d0>
2.05791211128
<closed file '40KB.txt', mode 'rb' at 0x10282e5d0>
1.9612839222
<closed file '40KB.txt', mode 'rb' at 0x10282e5d0>
1.96929216385
```

รูปที่ 4.24 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 40 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '50KB.txt', mode 'rb' at 0x10282e5d0>
2.87624788284
<closed file '50KB.txt', mode 'rb' at 0x10282e5d0>
2.37212705612
<closed file '50KB.txt', mode 'rb' at 0x10282e5d0>
2.52060103416
```

รูปที่ 4.25 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 50 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '60KB.txt', mode 'rb' at 0x10282e5d0>
3.16504502296
<closed file '60KB.txt', mode 'rb' at 0x10282e5d0>
2.72140407562
<closed file '60KB.txt', mode 'rb' at 0x10282e5d0>
3.05743503571
```

รูปที่ 4.26 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 60 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '70KB.txt', mode 'rb' at 0x10282e5d0>
3.29852294922
<closed file '70KB.txt', mode 'rb' at 0x10282e5d0>
3.08075404167
<closed file '70KB.txt', mode 'rb' at 0x10282e5d0>
3.35521006584
```

รูปที่ 4.27 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 70 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '80KB.txt', mode 'rb' at 0x10282e5d0>
3.94504189491
<closed file '80KB.txt', mode 'rb' at 0x10282e5d0>
3.62340903282
<closed file '80KB.txt', mode 'rb' at 0x10282e5d0>
3.39200210571
```

รูปที่ 4.28 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 80 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '90KB.txt', mode 'rb' at 0x10282e5d0>
4.22894501686
<closed file '90KB.txt', mode 'rb' at 0x10282e5d0>
4.18996882439
<closed file '90KB.txt', mode 'rb' at 0x10282e5d0>
4.1170759201
```

รูปที่ 4.29 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 90 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '100KB.txt', mode 'rb' at 0x10282e5d0>
4.46201896667
<closed file '100KB.txt', mode 'rb' at 0x10282e5d0>
4.34397101402
<closed file '100KB.txt', mode 'rb' at 0x10282e5d0>
4.39642000198
```

รูปที่ 4.30 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 100 กิโลไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '10MB.txt', mode 'rb' at 0x10071e5d0>
399.400231838
<closed file '10MB.txt', mode 'rb' at 0x10071e5d0>
412.932636023
<closed file '10MB.txt', mode 'rb' at 0x10071e5d0>
405.565413952
```

รูปที่ 4.31 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 10 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py

<closed file '20MB.txt', mode 'rb' at 0x10282e5d0>
794.013962984
<closed file '20MB.txt', mode 'rb' at 0x10282e5d0>
847.574511051
<closed file '20MB.txt', mode 'rb' at 0x10282e5d0>
800.712614059
```

รูปที่ 4.32 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 20 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '30MB.txt', mode 'rb' at 0x10282e5d0>
1250.71413279
<closed file '30MB.txt', mode 'rb' at 0x10282e5d0>
1100.25437307
<closed file '30MB.txt', mode 'rb' at 0x10282e5d0>
1109.74833107
```

รูปที่ 4.33 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 30 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss.py
<closed file '40MB.txt', mode 'rb' at 0x10072e5d0>
1592.29802084
<closed file '40MB.txt', mode 'rb' at 0x10072e5d0>
1516.98960304
<closed file '40MB.txt', mode 'rb' at 0x10072e5d0>
1548.77803493
```

รูปที่ 4.34 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 40 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss50.py
<closed file '50MB.txt', mode 'rb' at 0x10202e5d0>
1906.17232299
<closed file '50MB.txt', mode 'rb' at 0x10202e5d0>
1879.64927101
<closed file '50MB.txt', mode 'rb' at 0x10202e5d0>
1856.9439981
```

รูปที่ 4.35 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 50 เมกะไบต์

```
nuannaphas-MacBook-Pro:file nuannapha$ python dss60.py
<closed file '60MB.txt', mode 'rb' at 0x10282e5d0>
3600.698385
<closed file '60MB.txt', mode 'rb' at 0x10282e5d0>
3596.41514206
<closed file '60MB.txt', mode 'rb' at 0x10282e5d0>
2386.54570413
```

รูปที่ 4.36 ผลการทดลองการทำลายเซ็นดิจิตัลของข้อมูลที่ไฟล์ขนาด 60 เมกะไบต์

บทที่ 5
บทสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงบทสรุปของโครงงาน ขอบเขตและข้อจำกัดของระบบเสมือนอ้างอิง ปัญหาและอุปสรรคที่เกิดขึ้นขณะการดำเนินโครงงานและข้อแนะนำต่างๆ รวมไปถึงแนวทางการพัฒนาต่อยอดในอนาคต

5.1 บทสรุป

จากการทดลองเพื่อศึกษาปัญหา และเพิ่มประสิทธิภาพการรักษาความมั่นคงปลอดภัยของบิ๊กดาตา โดยโครงงานนี้มุ่งเน้นไปที่การรักษาความถูกต้องครบถ้วนของข้อมูลที่ถูกส่งจากแหล่งข้อมูลมายังหน่วยบันทึกข้อมูลที่เครื่องแม่ข่าย เมื่อทำการศึกษาด้วยคำสำคัญต่างๆ เพื่อค้นหาข้อมูลจากแหล่งข้อมูลต่างๆ พบว่าการสร้างลายน้ำของข้อมูล(Watermarking Embedding Processes) มีความเหมาะสมกับการรักษาความถูกต้องครบถ้วนของกระบวนการบริหารจัดการบิ๊กดาตา(Big Data Integrity) มากกว่าการสร้างลายเซ็นดิจิตัล(Digital Signature) ซึ่งเป็นอัลกอริทึมที่นิยมใช้เพื่อการรักษาความถูกต้องครบถ้วนของข้อมูลทั่วๆ ไป

เมื่อทำการทดลองเพื่อทดสอบประสิทธิภาพของการทำงานทั้งสองอัลกอริทึมกับไฟล์ที่ขนาดต่างๆ ตั้งแต่ 10 กิโลไบต์จนถึง 60 เมกะไบต์ พบว่าการสร้างลายน้ำของข้อมูลมีความรวดเร็วในการทำงานมากกว่าการสร้างลายเซ็นดิจิตัล จึงเป็นการยืนยันว่าการสร้างลายน้ำนั้นมีความเหมาะสมกับกระบวนการบริหารจัดการบิ๊กดาตามากกว่าการสร้างลายเซ็นดิจิตัล

จากสิ่งที่กล่าวมาทั้งหมดมิได้เป็นการสรุปว่ากระบวนการสร้างลายน้ำของข้อมูลนั้นเป็นแนวคิดกระบวนการรักษาความถูกต้องครบถ้วนของข้อมูลได้ดีกว่าการสร้างลายเซ็นดิจิตัลในทุกๆ กรณี แต่กระบวนการสร้างลายน้ำของข้อมูลนั้นเหมาะสมกับการรักษาความถูกต้องควบถ้วนของข้อมูลในกระบวนการบริหารจัดการบิ๊กดาตามากกว่าการสร้างลายเซ็นดิจิตัลเพียงเท่านั้น

5.2 ขอบเขตและข้อจำกัดของระบบเสมือนอ้างอิง

1.ระบบเสมือนอ้างอิงนี้จะทำงานได้ดีบนเงื่อนไข และข้อกำหนดต่างๆ ที่ได้กล่าวเอาไว้ในโครงงานฉบับนี้เพียงเท่านั้น

2.ต้องมีการลง Python 2.7.9 ก่อนการติดตั้งระบบเสมือนอ้างอิง

3.เพื่อให้การทำงานของระบบเสมือนอ้างอิงเป็นไปได้อย่างปกติ ควรตั้งค่าที่อยู่ไอพีของเครื่องแม่ข่ายเป็นแบบคงที่ หรืออาจใช้เป็นที่อยู่ไอพีส่วนตัวก็ได้

4.ข้อแนะนำควรติดตั้งระบบเสมือนอ้างอิงบนระบบปฏิบัติการที่มียูนิกซ์เป็นพื้นฐาน

5.3 ปัญหาและอุปสรรคที่เกิดขึ้นขณะการดำเนินโครงงาน

1. กระบวนการบริหารจัดการบิ๊กดาตานั้นมีการใช้ และพูดถึงกันอย่างแพร่หลายในปัจจุบัน แต่ยังไม่มีหนังสือ หรือบทความใดๆ ที่ระบุถึงเทคโนโลยีที่ดี และเหมาะสมที่สุดสำหรับกระบวนการบริหารจัดการบิ๊กดาตา ทั้งหมดเป็นเพียงสมมติฐานที่ตั้งขึ้น จึงจำเป็นต้องหาข้อมูลจากหลากหลายแหล่งข้อมูลเพื่อให้ได้สมมติฐานที่มีความชัดเจน และเป็นไปได้มากที่สุด

2. เนื่องจากเทคโนโลยีในกระบวนการบริหารจัดการบิ๊กดาตามีใช้อย่างหลากหลาย รวมถึงระบบเสมือนอ้างอิงจากหลายๆ องค์การ เพื่อให้สะดวกต่อการทำโครงงาน และการศึกษาต่อในอนาคตจึงต้องทำการศึกษา ทดลอง และลองผิดลองถูกอยู่หลายครั้ง กับหลายผลิตภัณฑ์ เพื่อให้เหมาะสมกับโครงงานมากที่สุด

3. การใช้งานกระบวนการบริหารจัดการบิ๊กดาตามีอยู่ในหลากหลายธุรกิจ หลายองค์การ ในโครงงานนี้มีการยกตัวอย่างกรณีศึกษาเพียงไม่หนึ่งถึงสองกรณีเพื่อความเข้าใจเบื้องต้น ซึ่งอาจจะเป็นการยากต่อการเข้าใจในกระบวนการบริหารจัดการบิ๊กดาตาที่แท้จริง และกว้างขวางมากยิ่งขึ้น

4.ผู้พัฒนายังไม่มีความชำนาญในการพัฒนาโปรแกรมมากนัก ทำให้ต้องใช้เวลาในการศึกษาเพิ่มเติมมากยิ่งขึ้น

5. เนื่องจากแนวคิดที่นำมาใช้ในโครงงานนี้เป็นเพียงอัลกอริทึมที่สามารถทดลองได้ด้วยหลากหลายวิธี และผู้พัฒนายังไม่มีประสบการณ์ด้านการพัฒนาโปรแกรมมาก จึงต้องทำการศึกษา และเลือกเครื่องมือมาใช้ให้เหมาะสมมากที่สุด รวมไปถึงการใช้ภาษาต่างๆ เพื่อนำมาใช้ในโครงงานนี้ด้วยเช่นกัน

5.4 ข้อเสนอแนะ

1. เพื่อให้กระบวนการบริหารจัดการบิ๊กดาตาสามารถเข้าใจได้มากขึ้น จึงควรจำกัดขอบเขตของการศึกษาให้มีขนาดเล็กลงเพื่อให้การศึกษาเป็นไปในทิศทางที่ชัดเจนขึ้น และสามารถหาข้อมูลจากแหล่งข้อมูลต่างๆ ได้รวดเร็ว และตรงประเด็นมากยิ่งขึ้น

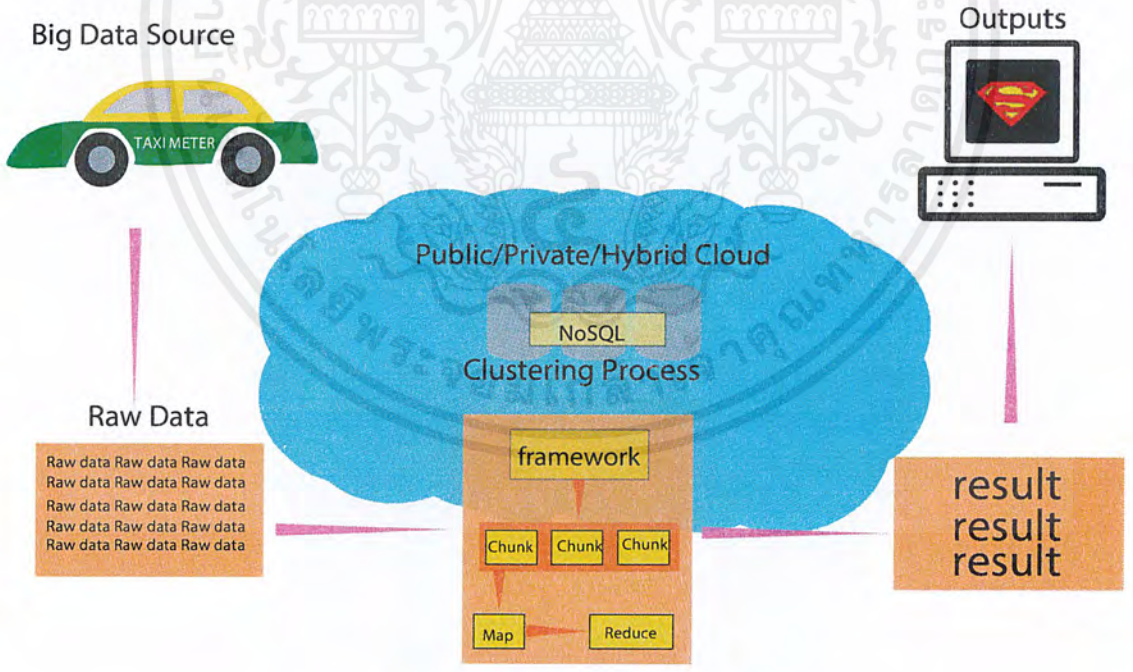
2. เนื่องจากผลิตภัณฑ์ที่สามารถใช้เป็นระบบอ้างอิงเสมือนนั้นมีหลากหลาย เพื่อให้รวดเร็วในการดำเนินงานมากขึ้นจึงควรศึกษา และหาข้อมูลของแต่ละผลิตภัณฑ์ให้รอบคอบก่อนการลงมือติดตั้งและทดลองใช้ระบบต่างๆ เพื่อให้ลดเวลาการทำงานให้น้อยลง

3. การยกตัวอย่างกรณีศึกษาควรมีหลากหลายสถานการณ์มากขึ้น เพื่อให้ครอบคลุม และสามารถเข้าใจการใช้งานกระบวนการบริหารจัดการบิ๊กดาตาที่มีในปัจจุบันมากยิ่งขึ้น

4. ในการเลือกใช้ภาษา และเครื่องมือต่างๆ ในการพัฒนาโปรแกรมเพื่อการทดลองในส่วนต่างๆ นั้น ควรศึกษาการใช้เครื่องมือต่างๆ ให้ละเอียด และรอบคอบในการเลือกใช้ภาษา และเครื่องมือต่างๆ ในการนำมาเพื่อใช้ในการพัฒนาโปรแกรมเพื่อการทดลอง

5.5 แนวทางการพัฒนา

สำหรับโครงงานเรื่องความปลอดภัยของบิ๊กดาตานั้นเป็นเพียงจุดเริ่มต้นของการศึกษาอีกแง่มุมหนึ่งที่สำคัญไม่น้อยไปกว่าการพัฒนาเพื่อการประมวลผลต่างๆ หรือการทำงานในส่วนอื่นๆ ของกระบวนการบริหารจัดการข้อมูล อาทิเช่นการจัดเก็บข้อมูล การเรียกข้อมูลจากฐานข้อมูล การจัดเรียงข้อมูล หรือการแสดงผล เป็นต้น ซึ่งมุมมองในด้านของการรักษาความปลอดภัยข้อมูลที่เริ่มต้นในโครงงานนี้การมุ่งเน้นไปที่รักษาความถูกต้องครบถ้วนของข้อมูลจากแหล่งข้อมูล(Data Sources) ไปยังฐานข้อมูล(Database) ที่อยู่ภายในเครื่องแม่ข่าย(Server) ซึ่งนับเป็นจุดเริ่มต้นของการนำข้อมูลบิ๊กดาตาเข้าสู่ระบบเพียงเท่านั้น



รูปที่ 5.1 แผนภาพกระบวนการบริหารจัดการข้อมูล

จากแผนภาพกระบวนการบริหารจัดการข้อมูลทั้งหมดจะพบว่ายังมีประเด็นปัญหาด้านการรักษาความปลอดภัยอีกหลายส่วน ซึ่งสามารถนำโครงงานนี้ไปศึกษาเพื่อเป็นองค์ความรู้ และแนวคิด

เบื้องต้นในการพัฒนาการรักษาความปลอดภัยของบิ๊กดาตาในส่วนต่างๆ ต่อไปในอนาคต เพื่อให้สอดคล้องกับแนวคิดในหลักของการรักษาความปลอดภัยของระบบสารสนเทศอันได้แก่การรักษาข้อมูลอันเป็นความลับ(Confidentiality), การรักษาความถูกต้องครบถ้วนของข้อมูล(Integrity) และการรักษาสภาพพร้อมใช้งานของระบบ(Availability) เพื่อให้ความปลอดภัยของบิ๊กดาตาเป็นไปได้อย่างเหมาะสมในระบบบริหารจัดการข้อมูลสารสนเทศในอนาคต

บรรณานุกรม

[1] ธนัญชัย ตรีภาค. **เอกสารการสอนวิชา Network Security** : ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[2] Apache Hadoop **"What is Apache Hadoop."** [Online]. Available : http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F

[3] Apache Hbase **"Apache Hbase."** [Online]. Available : http://hbase.apache.org/

[4] Apache Hive **"Apache Hive."** [Online]. Available : https://hive.apache.org/

[5] Apache Kylin **"OLAP Engine for Big Data."** [Online]. Available : http://www.kylin.io/

[6] Big Data Working Group "Top Ten Big Data Security and Privacy Challenges." **cloud security alliance.** , November 2012.

[7] Big Data Working Group "Expanded Top Ten Big Data Security and Privacy Challenges." **cloud security alliance.** ,April 2013.

[8] Bingsheng Z., Qin Z. "PriWhisper: Enabling Keyless Secure Acoustic Communication for Smartphones." IEEE pp. 1-13.

[9] Charlie Kaufman, Radia Perlman, Mike Speciner,Prentice Hall. **Network Security: Private Communication in a Public World, Second Edition** Upper Saddle River, New Jersey 07458: Pearson Education, Inc.

[10] Hortonworks **"Hadoop Distributed File Systems(HDFS)."** [Online]. Available : http://hortonworks.com/hadoop/hdfs/

[11] Jiang Xu "Kylin: Hadoop OLAP Engine – Tech Deep Dive" [Slide]. **ebay inc.**

[12] Judith Hurwitz, Alan Nugent, Dr.Fern Halper, Marcia Kaufman **Big Data for Dummies.** 111 River Street Hoboken: John Wiley & Sons, Inc. 2013.

[13] Supriya Rai, Ruchi Dubey "A Novel Keyless Algorithm for Steganography"

[14] Ibrahim Kamel, Hussam Juma "A Lightweight Data Integrity Scheme for Sensor Networks"

แผ่น CD/ DVD ประกอบการศึกษาปริญญานิพนธ์

“ความปลอดภัยของบิ๊กดาตา(Big Data Security)”