แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบไอโอที ด้วยการเข้ารหัสแบบโฮโมมอร์ฟิก

DATA PRIVACY-PROTECTION SERVICE PLATFORM FOR IOT SYSTEMS USING HOMOMORPHIC ENCRYPTION

กลิกา สุขสมบูรณ์[1], เอมอัชนา นิรันตสุขรัตน์[2], ณัฐเสฐ ธนบดี[3], รุจน์กวิน สวัสดิ์จิรธำรง[4], ชาวีร์ อิสริยภัทร์[5], ตะวัน ห่อหุ้ม[6], โสภณ มงคลลักษมี[7], ณฐพล ตันสังวรณ์[8], สุขุมาล กิติสิน[9]

Kalika Suksomboon[1], Aimaschana Niruntasukrat[2], Nataset Tanabodee[3], Rujgavin Sawatjirathamrong[4], Chavee Issariyapat[5], Tawan Hohum[6], Sophon Mongkolluksamee[7], Natapon Tansangworn[8], Sukumal Kitisin[9]

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ปทุมธานี 12120[1 ถึง 2, 6 และ 8]
มหาวิทยาลัยเกษตรศาสตร์ กรุงเทพฯ 10900[3 ถึง 4 และ 9]
บริษัท เน็กซ์พาย จำกัด กรุงเทพฯ 10310[5]
มหาวิทยาลัยศรีนครินทรวิโรฒ กรุงเทพฯ 10110[7]

National Electronics and Computer Technology Center, Pathum Thani 12120 Thailand[1 to 2, 6 and 8]
Kasetsart University, Bangkok 10900 Thailand[3 to 4 and 9]
NEXPIE Co., Ltd., Bangkok 10310 Thailand[5]
Srinakharinwirot University, Bangkok 10110 Thailand[7]

Corresponding E-mail : kalika.suk@nectec.or.th

Received Date March 19, 2024
Revised Date April 30, 2025
Accepted Date May 1, 2025

บทคัดย่อ

บทความนี้นำเสนอการศึกษาพัฒนาแพลตฟอร์มให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบไอโอทีที่สามารถประยุกต์ใช้กับงานภาคอุตสาหกรรมและให้ผู้ใช้งานทั่วไปสามารถใช้งานได้ ศึกษาโดยใช้วิธีวิจัยประยุกต์เทคโนโลยีการเข้ารหัสโฮโมมอร์ฟิกที่ทำให้ข้อมูลเข้ารหัสถูกส่งไปคำนวณบนคลาวด์ได้โดยไม่ต้องถอดรหัส ผลการศึกษาครั้งนี้สามารถพัฒนาแพลตฟอร์มควบคุมการคำนวณข้อมูลของอุปกรณ์ไอโอทีบนคลาวด์ที่ใช้กับงานภาคอุตสาหกรรมด้วยการสร้างและจัดการกุญแจรหัสลับผ่านแอปพลิเคชันบนโทรศัพท์มือถือโดยง่าย แม้ผู้ใช้งานไม่มีความเชี่ยวชาญด้านเทคโนโลยี ผลสำเร็จของแพลตฟอร์มพิสูจน์ได้จากการนำไปใช้งานจริงในกรณีศึกษาของโรงงาน 2 แห่ง สำหรับระบบตรวจสอบระดับน้ำมันในไซโลและสำหรับระบบตรวจสอบสภาวะของเครื่องจักร ซึ่งผลการทดสอบประสิทธิภาพในการเข้ารหัส ถอดรหัส และความถูกต้องอยู่ในระดับที่นำไปใช้งานได้จริง โดยระดับความถูกต้องในการทดสอบภาคสนามอยู่ที่ทศนิยม 4 ตำแหน่งสำหรับการใช้กุญแจเข้ารหัสขนาดเล็ก และทศนิยม 6 ตำแหน่งสำหรับการใช้กุญแจเข้ารหัสขนาดกลางและขนาดใหญ่ อีกทั้งรองรับการคำนวณข้อมูลด้วยความเร็วสูงสุดในการส่งข้อมูลที่ 0.5 วินาทีต่อการส่งข้อมูล 1 ชุด สำหรับการใช้กุญแจเข้ารหัสขนาดเล็กและขนาดกลาง และ 5 วินาทีต่อการส่งข้อมูล 1 ชุด สำหรับการใช้กุญแจเข้ารหัสขนาดใหญ่

คำสำคัญ: การเข้ารหัสแบบโฮโมมอร์ฟิก คลาวด์คอมพิวติง ความเป็นส่วนตัวของข้อมูล อินเทอร์เน็ตสรรพสิ่ง (ไอโอที)

Abstract

This article presents the study on development of a privacy-protection service platform for industrial IoT systems and general users, leveraging homomorphic encryption to enable secure data processing in the cloud without requiring decryption. From this study, a privacy-protection service platform for industrial IoT was developed. The platform was designed to be user-friendly and accessible via a mobile application, allowing even non-technical users to manage encryption keys securely. The platform was used as case studies at two factories: one for an oil monitoring system in silos, and the other for a machinery inspection system. The results showed that the platform could perform encryption, decryption, and data transmission with high accuracy and efficiency. Field tests demonstrated accuracy levels of up to four decimal places with small encryption keys, and up to six decimal places with medium and large keys. Data transmission latency was measured at 0.5 seconds per data set for small and medium keys, and five seconds for large keys.

Keywords: homomorphic encryption, cloud computing, privacy of data, Internet of Things (IoT)

1. บทนำ

แนวทางการขับเคลื่อนอุตสาหกรรมไทยไปสู่อุตสาหกรรม 4.0 ทำได้โดยการผนวกเทคโนโลยีสารสนเทศเข้ากับระบบกายภาพ เพื่อให้เกิดการเชื่อมโยงข้อมูลของอินเทอร์เน็ตสรรพสิ่ง หรือ ไอโอที (Internet of Things: IoT) ซึ่งก่อให้เกิดระบบไซเบอร์กายภาพ (Cyber-Physical System: CPS) ทำให้สามารถดึงเอาข้อมูลจากสิ่งกายภาพ เช่น อุปกรณ์ เครื่องจักร มาวิเคราะห์ในไซเบอร์เพื่อควบคุมป้อนกลับไปยังสิ่งกายภาพได้แบบอัตโนมัติและมีประสิทธิภาพ เทคโนโลยีไอโอทีซึ่งเป็นหัวใจสำคัญหนึ่งของระบบไซเบอร์กายภาพจึงกลายมาเป็นเทคโนโลยีแกนหลักของอุตสาหกรรม 4.0 ที่ทำให้การเชื่อมต่อข้อมูลต่าง ๆ ในอุตสาหกรรมผ่านเครือข่ายไปยังไอโอทีคลาวด์ (IoT cloud) ทำให้เกิดการแลกเปลี่ยนข้อมูลระหว่างสรรพสิ่ง เพื่อนำข้อมูลเหล่านั้นมาประมวลผลและควบคุมระบบอุตสาหกรรมให้มีความอัจฉริยะ

ปัจจุบันมีความนิยมในการใช้บริการไอโอทีผ่านคลาวด์สาธารณะ (public cloud) กันมาก เนื่องจากลดภาระของผู้ใช้งานในการดูแลระบบหลังบ้าน บริการเหล่านี้ล้วนรองรับการส่งผ่านข้อมูลจากอุปกรณ์ไอโอทีหรืออุปกรณ์ไอโอทีเกตเวย์ (IoT gateway) ไปยังคลาวด์ด้วยช่องทางการสื่อสารแบบปลอดภัย (secure session) ได้แก่ Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) โดยข้อมูลจะถูกเข้ารหัส (encryption) เพื่อป้องกันการถูกโจมตีทางไซเบอร์จากภายนอก เช่น การดักฟังข้อมูล (eavesdropping) ระหว่างทาง นอกจากนี้ยังมีระบบการระบุตัวตน (authentication) และการควบคุมการเข้าถึงข้อมูล (access control) เพื่อป้องกันทรัพยากรและข้อมูลที่เก็บบนคลาวด์จากการถูกโจมตี อย่างไรก็ตาม วิธีการเข้ารหัส (encryption algorithms) ที่ใช้ในปัจจุบันข้างต้น ไม่เอื้ออำนวยให้ข้อมูลที่ถูกเข้ารหัสแล้วนำไปประมวลผลต่อได้ ดังนั้น ข้อมูลที่ถูกส่งไปประมวลผลที่คลาวด์จำเป็นต้องถูกเปิดเผยหรือถอดรหัส (decrypt) ก่อนเสมอ และผลลัพธ์ที่ได้จากการประมวลผล

ดังกล่าวอาจถูกนำไปควบคุมอุปกรณ์หรือส่งไปแสดงบนแดชบอร์ดที่ติดตั้งอยู่บนอุปกรณ์ของผู้ใช้งานเอง แต่ผู้ให้บริการคลาวด์กลับสามารถล่วงรู้ข้อมูลที่ถูกส่งมายังคลาวด์รวมทั้งผลลัพธ์ของการประมวลผล จึงก่อให้เกิดความกังวลของภาคอุตสาหกรรม โดยเฉพาะอย่างยิ่ง ในกรณีที่เป็นข้อมูลอ่อนไหวซึ่งหากรั่วไหลไปสู่คู่แข่งอาจนำมาซึ่งความเสียหายทางธุรกิจ เช่น ตัวเลขการผลิต และอาจกลายเป็นอุปสรรคสำคัญในการนำเทคโนโลยีไอโอทีมาใช้ ซึ่งจะส่งผลกระทบไปถึงการดำเนินงานตามกรอบนโยบายของภาครัฐที่มุ่งพัฒนาให้อุตสาหกรรมไทยก้าวไปสู่อุตสาหกรรม 4.0 เป็นไปอย่างล่าช้า ดังนั้น จึงมีความจำเป็นอย่างยิ่งที่จะต้องมีการพัฒนาระบบบริการแพลตฟอร์มไอโอทีคลาวด์ ให้สามารถปกป้องความเป็นส่วนตัวของข้อมูลในระบบไอโอทีไปพร้อมกับยังรักษาความสามารถของคลาวด์ในการประมวลผลได้อย่างมีประสิทธิภาพ เพื่อให้ผู้ประกอบการหันมาใช้บริการไอโอทีคลาวด์และนำเทคโนโลยีไอโอทีมาใช้ในโรงงานของตนได้อย่างเชื่อมั่น

ต่างประเทศเริ่มมีความตื่นตัวในการคิดค้นนวัตกรรมที่เกี่ยวข้องกับการคุ้มครองความเป็นส่วนตัวของข้อมูล เช่น Intel ได้ผลิตอุปกรณ์เซิร์ฟเวอร์ Intel® SGX ที่มีสถาปัตยกรรมรองรับการประมวลผลข้อมูลที่คุ้มครองความเป็นส่วนตัวของข้อมูลภายใต้เทคโนโลยีสภาพแวดล้อมการปฏิบัติการที่เชื่อถือได้ (Trusted Execution Environments: TEE) ซึ่งเทคโนโลยีนี้ได้ถูกเสนอเป็นทางเลือกให้กับผู้ใช้บริการคลาวด์รายใหญ่ ๆ เช่น AWS และ Azure ในการปกป้องข้อมูล โดยการให้บริการนั้นจำกัดอยู่กับฮาร์ดแวร์จำเพาะที่รองรับ Intel® SGX เท่านั้น (Intel, n.d.) นอกจากนี้ ยังมีบริษัทสตาร์ตอัปต่าง ๆ ในสหรัฐอเมริกา เช่น Enveil, Inpher, และ TripleBlind ที่นำเทคโนโลยีการเข้ารหัสโฮโมมอร์ฟิก (Homomorphic Encryption: HE) มาใช้งาน โดยมีกลุ่มเป้าหมายลูกค้าระดับองค์กรที่ต้องการนำข้อมูลไปประมวลผลบนคลาวด์ซึ่งจะรองรับการสื่อสารในรูปแบบของไคลเอนต์หรือเซิร์ฟเวอร์ (client/server) เท่านั้น (Enveil, n.d.; Inpher, n.d; TripleBlind, n.d.) ข้อจำกัดด้านฮาร์ดแวร์จำเพาะที่มีราคาสูงและต้องอาศัยระบบประมวลผลขนาดใหญ่ในการเข้ารหัสโฮโมมอร์ฟิก เช่น เซิร์ฟเวอร์ x86_64 ทำให้แพลตฟอร์มที่มีอยู่ในท้องตลาดปัจจุบันยังไม่เหมาะสมในการนำมาใช้งานกับระบบไอโอทีซึ่งมักประกอบด้วยอุปกรณ์ขนาดเล็ก มีขนาดทรัพยากรที่ใช้ประมวลผลจำกัด จึงเป็นความท้าทายอย่างยิ่งในการพัฒนานวัตกรรมใหม่สำหรับให้บริการความคุ้มครองข้อมูลส่วนบุคคลในระบบไอโอที และทำให้ระบบที่พัฒนาขึ้นสามารถทำงานเข้ากันได้กับการให้บริการบนแพลตฟอร์มไอโอทีคลาวด์ เพื่อลดต้นทุนอุปกรณ์เซิร์ฟเวอร์ให้กับผู้ประกอบการและเพิ่มขีดความสามารถในการแข่งขันด้านนวัตกรรมกับต่างประเทศ รวมทั้งยังสอดรับกับพระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล พ.ศ. 2562 และเมื่อพิจารณาผู้ใช้งานในภาคอุตสาหกรรมที่ไม่มีความรู้ความเชี่ยวชาญด้านวิทยาการเข้ารหัสทำให้เกิดความท้าทายอีกประการหนึ่งคือ การออกแบบแพลตฟอร์มให้ใช้งานง่ายสำหรับผู้ใช้งานระดับทั่วไปอีกด้วย

2. วัตถุประสงค์

2.1 เพื่อพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลของอุปกรณ์ไอโอทีด้วยการเข้ารหัสแบบโฮโมมอร์ฟิกที่สามารถประยุกต์ใช้กับงานภาคอุตสาหกรรม จากนั้นทดสอบประสิทธิภาพการใช้งานจริงของแพลตฟอร์ม

2.2 เพื่อพัฒนาสร้างแพลตฟอร์มที่ผู้ใช้งานไม่จำเป็นต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัส

3. วิธีการศึกษา

การศึกษานี้ใช้การวิจัยประยุกต์ โดยมีขั้นตอนดังนี้

3.1 ทบทวนวรรณกรรมที่เกี่ยวข้องกับเทคโนโลยีเข้ารหัสแบบโฮโมมอร์ฟิกและแพลตฟอร์มไอโอทีที่ให้บริการในปัจจุบัน เพื่อนำมาเป็นแนวทางพัฒนาแพลตฟอร์มให้รองรับการใช้งานกับระบบไอโอทีสำหรับภาคอุตสาหกรรม

3.2 กำหนดขอบเขตของการพัฒนาครั้งนี้ให้รองรับการส่งข้อมูลไอโอทีด้วยโพรโทคอล MQTT (Message Queuing Telemetry Transport protocol) และออกแบบให้ผู้ใช้งานสามารถจัดการสูตรการคำนวณเองได้

3.3 ออกแบบแพลตฟอร์ม โดยอธิบายถึงองค์ประกอบรวมของแพลตฟอร์ม และนำเสนอผลการทดสอบประสิทธิภาพของการใช้งานได้จริงในโรงงานอุตสาหกรรม

3.4 ศึกษาข้อจำกัดของแพลตฟอร์มและแนวทางการพัฒนาต่อในอนาคต

4. การทบทวนวรรณกรรม

เทคโนโลยีเข้ารหัสแบบโฮโมมอร์ฟิกและแพลตฟอร์มไอโอทีที่ให้บริการในปัจจุบัน

การเข้ารหัสแบบโฮโมมอร์ฟิก จำแนกตามการดำเนินการ (operation) ทางคณิตศาสตร์ได้ 3 ประเภท ได้แก่

1) การเข้ารหัสโฮโมมอร์ฟิกบางส่วน (Partial Homomorphic Encryption: PHE)

2) การเข้ารหัสโฮโมมอร์ฟิกเกือบสมบูรณ์ (Somewhat Homomorphic Encryption: SHE)

3) การเข้ารหัสโฮโมมอร์ฟิกสมบูรณ์ (Fully Homomorphic Encryption: FHE)

PHE เป็นวิธีการเข้ารหัสแบบโฮโมมอร์ฟิกซึ่งรองรับการบวกหรือการคูณข้อมูลที่เข้ารหัสอย่างใดอย่างหนึ่งเพียงวิธีเดียวแต่ไม่จำกัดจำนวนครั้ง ส่วน SHE รองรับฟังก์ชันทางคณิตศาสตร์ที่มีทั้งการบวกและการคูณแต่จำกัดจำนวนครั้ง ในขณะที่ FHE รองรับทั้งการบวกและการคูณโดยไม่จำกัดจำนวนครั้ง เหตุผลที่เป็นเช่นนี้เพราะผลของการเข้ารหัสโฮโมมอร์ฟิกจะเพิ่มข้อมูลรบกวน (noise) เข้าไปในข้อมูลเดิม และการคูณจำนวนที่เข้ารหัสเข้าด้วยกันจะทำให้ข้อมูลรบกวนนี้ถูกสะสมเพิ่มขึ้นเรื่อย ๆ ตามจำนวนครั้งของการคูณ เมื่อข้อมูลรบกวนมากเกินระดับที่การเข้ารหัสรองรับได้จะไม่สามารถถอดรหัสข้อมูลกลับมาได้ถูกต้อง ดังนั้น ในทางทฤษฎีแม้ว่า FHE จะอนุญาตให้บวกและคูณข้อมูลที่เข้ารหัสได้โดยไม่จำกัด โดยจะมีขั้นตอนในการลดข้อมูลรบกวนภายหลังการดำเนินการคูณแต่ละครั้ง แต่ในทางปฏิบัติยังคงจำเป็นต้องออกแบบ FHE โดยจำกัดจำนวนครั้งของการคูณในฟังก์ชันการคำนวณ เพราะหากต้องการคูณแบบไม่จำกัดจำนวนครั้ง ข้อมูลเข้ารหัส FHE จะต้องมีขนาดใหญ่มหาศาลและต้องใช้ทรัพยากรคำนวณมากตามไปด้วย

สำหรับเทคนิคการจัดการข้อมูลรบกวนที่เกิดขึ้นหลังจากนำข้อมูลเข้ารหัสด้วยโฮโมมอร์ฟิกมาคูณหรือบวกกันมีหลายวิธี แต่วิธีหลักที่จะขอกล่าวถึงในบทความนี้คือ เทคนิคการบูตสแตรปปิง (bootstrapping) หรือการทำให้ข้อมูลเสมือนเพิ่งเข้ารหัสใหม่ ซึ่งถูกนำเสนอโดย Craig Gentry ใน พ.ศ 2552 (ค.ศ. 2009) โดยจุดประสงค์เพื่อนำมาใช้กับวิธีการเข้ารหัส FHE ที่เสนอโดย Craig Gentry ซึ่งอยู่บนพื้นฐานทางคณิตศาสตร์ของ Ideal Lattices เป็นผลบวกเชิงเส้นของเวกเตอร์ฐาน (basis vector) ที่เป็นอิสระจากกัน (Gentry, 2009) เทคนิคการบูตสแตรปปิงจะทำให้ข้อมูลเข้ารหัสที่มีข้อมูลรบกวนสูงกลับมาเสมือนข้อมูลเพิ่งเข้ารหัสใหม่ (fresh ciphertext) โดยการเข้ารหัสใหม่ แต่วิธีการทำบูตสแตรปปิงมีขั้นตอนที่ซับซ้อน กล่าวคือ 1) ต้องทำการเปลี่ยนข้อมูลเข้ารหัสที่ต้องการทำการบูตสแตรปปิงให้อยู่ในรูปของข้อมูลที่สามารถทำการบูตสแตรปปิงได้เสียก่อน ซึ่งเรียกว่าเทคนิคสควอชชิง (squashing) คือการคูณข้อมูลเข้ารหัสที่มีข้อมูลรบกวนสูงด้วยเซตของเวกเตอร์ที่ผลรวมมีค่าเท่ากับส่วนผกผันการคูณของกุญแจสำหรับถอดรหัส (multiplicative inverse of the secret key) 2) เมื่อข้อมูลเข้ารหัสดังกล่าวถูกเปลี่ยนแล้วจะทำการเข้ารหัสอีกครั้งด้วยกุญแจชุดใหม่คู่ขนานกับชุดแรก ทำให้ได้ข้อมูลเข้ารหัสที่เสมือนเพิ่งเข้ารหัสใหม่ที่ข้อมูลรบกวนได้ถูกกำจัดไปแล้ว ดังนั้น ถึงแม้ว่าเทคนิคการบูตสแตรปปิงนี้ สามารถทำให้ข้อมูลที่เข้ารหัสโฮโมมอร์ฟิกด้วยวิธีของ Craig Gentry จะสามารถดำเนินการคำนวณได้ไม่จำกัดแต่กลับส่งผลให้ค่าใช้จ่ายในการคำนวณสูง ดังจะเห็นได้จากผลการทดลองในตารางที่ 3 ใน Acar et al. (2017) แสดงให้เห็นว่าการเข้ารหัสใหม่ใช้เวลา 31 นาที และใช้เวลาในการสร้างกุญแจเข้ารหัส 2.2 ชั่วโมง จึงทำให้ไม่สามารถนำเทคนิคนี้มาใช้งานได้ในทางปฏิบัติ

ต่อมา Brakerski et al. (2014) เสนอเทคนิค leveled-FHE ที่ไม่ต้องอาศัยเทคนิคการบูตสแตรปปิง แต่ใช้แนวคิดการคาดการณ์ไว้ก่อนว่าจะมีการดำเนินการกับข้อมูลเข้ารหัสแบบโฮโมมอร์ฟิกกี่ครั้ง (เช่น L-1 ครั้ง) การเข้ารหัสข้อมูลแบบ leveled-FHE จากนั้นจะเข้ารหัสแบบชุดข้อมูล (batch) และอาศัยการทำโมดูลัสสวิตชิง (modulus switching) เพื่อทำให้ข้อมูลมีขนาดคงที่ โดยจำนวนครั้งการทำโมดูลัสสวิตชิงจะถูกกำหนดด้วยขนาดความลึกของการเข้ารหัสข้อมูล จำนวนที่ได้จากการคาดการณ์ (level) ของการเข้ารหัสวิธีนี้จะอยู่บนพื้นฐานของ Ring-Learning With Error (RLWE) ดังนั้นจึงช่วยทำให้เทคนิค leveled-FHE ลดความซับซ้อนของการลดข้อมูลรบกวนที่เกิดจากการคูณได้

FHE ที่ใช้งานกันอยู่ในปัจจุบันมีด้วยกันหลายแบบ เช่น BFV GSW TFHE และ RNS-CKKS โดยผลเปรียบเทียบคุณสมบัติแสดงในตารางที่ 1 พบว่า RNS-CKKS เป็นวิธีเดียวที่รองรับการคำนวณข้อมูลตัวเลขจำนวนเต็ม (integers) และจำนวนตรรกยะ (rational numbers) (Cheon et al., 2017) ดังนั้นเทคนิค FHE แบบ RNS-CKKS จึงเหมาะสมที่สุดกับการนำไปใช้งานกับแอปพลิเคชันไอโอที (IoT application) สำหรับข้อมูลในอุตสาหกรรมโดยทั่วไปที่ไม่ได้จำกัดอยู่แค่จำนวนเต็มบวกเท่านั้น

ตารางที่ 1 การเปรียบเทียบวิธีเข้ารหัสแบบ FHE ตามชนิดข้อมูลที่รองรับและวิธีการลดข้อมูลรบกวน

FHE scheme	Types of Messages			Noise Reduction	
	Integer	Boolean	Fixed-Point Number	Bootstrapping	Leveled
BFV	x	-	-	-	x
GSW	x	-	-	x	-
TFHE	-	x	-	x	-
RNS-CKKS	x	x	x	-	x

หมายเหตุ: [x] หมายถึง รองรับ
[-] หมายถึง ไม่รองรับ

จากการสำรวจคอมไพเลอร์และไลบรารี HE เช่น Microsoft SEAL (Microsoft, n.d.), Lattigo (Mouchet et al., 2020), HEAAN (Cheon et al., 2017) และ PySEAL (Stavish, 2017) ที่สามารถนำมาพัฒนาต่อยอดได้ในปัจจุบัน พบว่าในกลุ่มที่รองรับ FHE แบบ RNS-CKKS นั้น Microsoft SEAL ซึ่งถูกพัฒนาด้วยภาษา C++ มีการดำเนินการให้ใช้งานครบถ้วนกว่าไลบรารีอื่น ๆ ผู้วิจัยจึงเลือก Microsoft SEAL มาใช้เป็นเครื่องมือพัฒนา FHE ของแพลตฟอร์ม ในปัจจุบันมีผู้ให้บริการคำนวณข้อมูลที่เข้ารหัสแบบโฮโมมอร์ฟิกอยู่เพียงไม่กี่รายและทั้งหมดเป็นโมเดลแบบไคลเอนต์หรือเซิร์ฟเวอร์ที่มีการสื่อสารรูปแบบร้องขอหรือตอบรับ (request response) ทั้งสิ้น ซึ่งไม่รองรับการส่งข้อความแบบเวลาจริง (real-time messaging) ที่เป็นการสื่อสารหลักที่นิยมใช้กันในระบบไอโอที นอกจากนี้ยังไม่รองรับอุปกรณ์ปลายทางขนาดเล็กและอุปกรณ์สื่อสารเคลื่อนที่ รวมทั้งมุ่งเป้าไปที่ให้บริการกลุ่มธุรกิจด้านการแพทย์และการเงินเป็นหลัก โดยเน้นไปที่การเก็บข้อมูลหรือวิเคราะห์ข้อมูลที่ไม่ได้ต้องการความรวดเร็วในระดับเวลาจริง เมื่อพิจารณาเทคนิคการเข้ารหัสโฮโมมอร์ฟิกที่ใช้ ทุกบริการล้วนมีข้อจำกัด เนื่องจากไม่มีบริการใดที่รองรับการคำนวณบนข้อมูลตัวเลขตรรกยะ หรือข้อมูลที่มีทศนิยมตายตัว (fixed-point numbers) ได้ ทำให้ยากต่อการนำไปประยุกต์ใช้งานโดยตรงกับภาคอุตสาหกรรม เช่น การคำนวณค่าเซนเซอร์ เนื่องจากเจ้าของข้อมูลจำเป็นต้องสเกลข้อมูลที่ต้นทางและปลายทางให้เป็นจำนวนเต็มเอง แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลที่พัฒนาขึ้นภายใต้ชื่อ CYBLION เป็นบริการเดียวที่พัฒนาต่อยอดจากวิธีการ RNS-CKKS ให้รองรับข้อมูลจำนวนตรรกยะที่ถูกประมาณค่าให้อยู่ในรูปของจำนวนที่มีทศนิยมตายตัวได้ จึงพร้อมประยุกต์ใช้กับการคำนวณในงานภาคอุตสาหกรรมได้ทันที

5. นิยามศัพท์เฉพาะ

การเข้ารหัสโฮโมมอร์ฟิก หมายถึง การเข้ารหัสข้อมูลที่สามารถนำไปคำนวณทางคณิตศาสตร์ต่อได้โดยไม่ต้องถอดรหัสและผลลัพธ์ที่ได้จากการคำนวณเมื่อถอดรหัสแล้วมีค่าใกล้เคียงกับผลลัพธ์จากการคำนวณบนข้อมูลที่ไม่เข้ารหัส

แพลตฟอร์มเน็ตพาย (NETPIE) หมายถึง แพลตฟอร์มไอโอทีคลาวด์สาธารณะของไทยที่ให้บริการรับส่งข้อมูลไอโอทีด้วยโพรโทคอล MQTT แบบไม่เข้ารหัสข้อมูล โดยสามารถเข้าถึงบริการได้จาก netpie.io

MQTT เป็นโพรโทคอลการส่งข้อความที่อ้างอิงตามมาตรฐานการสื่อสารระหว่างเครื่องต่อเครื่อง โดยมากใช้กับอุปกรณ์ไอโอทีที่ต้องส่งและรับข้อมูลผ่านเครือข่ายที่มีข้อจำกัดด้านทรัพยากร โดยทำงานบนหลักการเผยแพร่ (publish) และ สมัครรับ (subscribe) เพื่อที่จะสามารถแยกผู้ส่งข้อความ (ผู้เผยแพร่) ออกจากผู้รับข้อความ (ผู้สมัครรับ) และมีโบรกเกอร์ (broker) เป็นตัวกลางจัดการการสื่อสารระหว่างผู้เผยแพร่และผู้สมัครรับข้อมูล โดยจะทำหน้าที่กรองข้อความขาเข้าทั้งหมดจากผู้เผยแพร่ และกระจายไปยังผู้สมัครรับข้อความอย่างถูกต้อง

6. กรอบแนวคิด/สมมติฐานการวิจัย

แพลตฟอร์มสำหรับให้บริการความคุ้มครองข้อมูลส่วนบุคคลในระบบไอโอทีที่ผู้วิจัยพัฒนาขึ้น เรียกว่า “CYBLION” อยู่บนพื้นฐานของเทคนิคการเข้ารหัสแบบโฮโมมอร์ฟิกซึ่งเหมาะสมกับเป้าหมายคือ การรักษาความเป็นส่วนตัวของข้อมูลไอโอทีที่ต้องการรับบริการคำนวณโดยคลาวด์ภายนอก รายละเอียดดังนี้

6.1 การออกแบบและพัฒนาแพลตฟอร์ม CYBLION

ผู้วิจัยได้ศึกษาและหารือกับโรงงานเป้าหมาย เพื่อสำรวจความต้องการใช้งานไอโอที ซึ่งสามารถนำมาวิเคราะห์และสรุปได้ดังนี้

6.1.1 โรงงานมีความต้องการนำไอโอทีไปใช้งานเพื่อปรับปรุงกระบวนการทำงานหรือการผลิตและอำนวยความสะดวกให้กับผู้ปฏิบัติงานในโรงงาน โดยการปรับปรุงดังกล่าวต้องส่งผลกระทบต่อการทำงานของระบบเดิมน้อยที่สุดตามความจำเป็น เช่น หยุดระบบเพื่อติดตั้งอุปกรณ์เพิ่มเติม

6.1.2 ในกรณีที่โรงงานมีการใช้งานไอโอทีเดิม ระบบมักจะประกอบด้วยการรวบรวมข้อมูลจากเซนเซอร์ผ่านอุปกรณ์เอดจ์คอมพิวติง (edge computing) ที่เขียนด้วยโฟลว์ (flow) และมีแดชบอร์ดการทำงานบนโปรแกรมโนดเรด (Node-RED) จากนั้น ข้อมูลจะถูกส่งด้วยโพรโทคอล MQTT ไปยังคลาวด์เพื่อการแสดงผลบนมอนิเตอร์และแจ้งเตือน

6.1.3 ผู้ประกอบการมีความกังวลเรื่องการรั่วไหลของข้อมูลบางประเภท เช่น ข้อมูลการผลิตและข้อมูลการใช้พลังงานหากต้องส่งข้อมูลไปคำนวณบนคลาวด์ภายนอก

ด้วยเหตุนี้ ผู้วิจัยจึงออกแบบสถาปัตยกรรมไอโอทีคลาวด์แพลตฟอร์มที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัสหรือแพลตฟอร์ม CYBLION โดยได้นำเทคนิคการเข้ารหัสแบบโฮโมมอร์ฟิก แบบ RNS-CKKS มาใช้ และพัฒนาต่อยอดให้ทำงานได้บนโปรแกรม Node-RED เพื่อให้นำไปบูรณาการกับระบบไอโอทีเดิมของโรงงานที่มีอยู่ เพียงแค่เพิ่มโนดเข้ารหัสแบบโฮโมมอร์ฟิกส์ที่ผู้วิจัยพัฒนาเข้าไปในโปรแกรมเดิมที่ใช้งานอยู่แล้วโดยไม่ต้องรอเปลี่ยนซอฟต์แวร์ของอุปกรณ์ ส่วนการคำนวณซึ่งอยู่บนคลาวด์ จะอยู่ในรูปแบบโฟลว์ของ Node-RED ซึ่งผู้ใช้งานสามารถเข้าไปปรับแต่งโฟลว์ของสูตรการคำนวณได้ด้วยตัวเอง ทั้งหมดนี้ช่วยให้การติดตั้งง่าย รวดเร็ว และเกิดผลกระทบต่อกระบวนการหรือระบบไอโอทีเดิมน้อยที่สุด นอกจากนี้ ผู้วิจัยยังได้พัฒนาแอปพลิเคชันบนโทรศัพท์บนมือถือเพื่อให้ผู้ใช้งานสามารถดูผลลัพธ์ของการคำนวณ รวมทั้งควบคุมความเป็นส่วนตัวของข้อมูลและความปลอดภัยของข้อมูลได้ด้วยตัวเอง เช่น การควบคุมและจัดการโครงการหรือโพรเจกต์ (project) การจัดการอุปกรณ์เอดจ์คอมพิวติง การสร้างและกระจายกุญแจรหัสลับ ทั้งนี้ ผู้วิจัยออกแบบให้จัดเก็บกุญแจดังกล่าวไว้ในอุปกรณ์สมาร์ตโฟนของผู้ใช้งานซึ่งจะช่วยป้องกันการโจรกรรมข้อมูลหากชื่อบัญชีผู้ใช้งาน (account username) และพาสเวิร์ด (password) ถูกคอมโพรไมส์ (compromise) เนื่องจากการถอดรหัสข้อมูลสามารถทำได้เฉพาะบนอุปกรณ์สมาร์ตโฟนที่เก็บกุญแจรหัสลับสำหรับถอดรหัสได้เท่านั้น ด้วยสถาปัตยกรรมที่ออกแบบมาให้ยืดหยุ่นและปลอดภัยของ CYBLION จะช่วยลดข้อกังวลของโรงงานและช่วยให้เกิดการขยายผลการใช้งานไอโอทีและเทคโนโลยีรักษาความเป็นส่วนตัวของข้อมูลมากขึ้นในอุตสาหกรรม

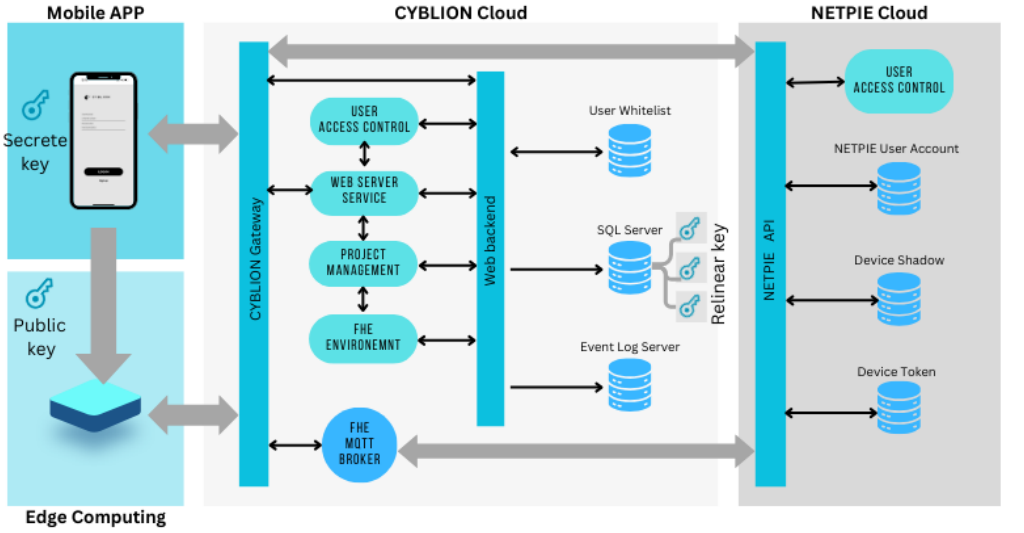
6.2 การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอที

องค์ประกอบหลักของการทำงานบนแพลตฟอร์ม CYBLION ประกอบด้วย 3 ส่วน ได้แก่

6.2.1 อุปกรณ์เอดจ์คอมพิวติง ทำหน้ารับข้อมูลไอโอทีมาเข้ารหัสโฮโมมอร์ฟิกด้วยกุญแจเข้ารหัส

6.2.2 คลาวด์ CYBLION (cyblion.io) ทำหน้ารับข้อมูลไอโอทีที่เข้ารหัสโฮโมมอร์ฟิกแล้วมาคำนวณโดยไม่ถอดรหัสและส่งผลลัพธ์การคำนวณไปยังแอปพลิเคชันบนโทรศัพท์มือถือ

6.2.3 แอปพลิเคชัน CYBLION ทำหน้าที่สร้างชุดกุญแจโฮโมมอร์ฟิกซึ่งประกอบด้วยกุญแจ 3 ชนิด คือ กุญแจสาธารณะ (public key: pk) ซึ่งจะต้องถูกส่งไปให้อุปกรณ์เอดจ์คอมพิวติงสำหรับการเข้ารหัส กุญแจรีลีเนียร์ (relinear key: rk) ซึ่งจะถูกส่งไปให้คลาวด์เพื่อใช้ในการลดข้อมูลรบกวนหลังการคำนวณ และกุญแจสำหรับถอดรหัส (secret key: sk) ซึ่งจะถูกเก็บไว้ในอุปกรณ์ที่ติดตั้งแอปพลิเคชันเท่านั้น ภายในแอปพลิเคชันยังมีแดชบอร์ดเวลาจริง (real-time dashboard) ที่ผู้ใช้สามารถปรับแต่งเพื่อแสดงผลการคำนวณที่ถอดรหัสแล้วได้หลายรูปแบบ เช่น เกจ (gauge) ตัวอักษร (text) กราฟฟีด (feed graph)



ภาพที่ 1 สถาปัตยกรรมการเชื่อมต่อระบบแพลตฟอร์ม CYBLION ที่เสนอในงานวิจัยนี้

การเชื่อมต่อการทำงานของแต่ละองค์ประกอบถูกออกแบบให้ทำผ่านส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface: API) และมีการเชื่อมต่อฐานข้อมูลผู้ใช้งานกับแพลตฟอร์มเน็ตพาย (NETPIE) เพื่อให้การบูรณาการระบบเป็นไปอย่างไร้รอยต่อ และลดความสับสนยุ่งยากของผู้ใช้งาน (NETPIE, 2020) เนื่องจากโรงงานเป้าหมายที่มีระบบไอโอทีอยู่แล้ว ล้วนใช้บริการแพลตฟอร์ม เน็ตพายอยู่เดิม ผังการเชื่อมต่อของแพลตฟอร์มแสดงดังภาพที่ 1 ขั้นตอนการใช้งานถูกกำหนดให้เริ่มต้นจากการเข้าใช้ล็อกอินแอปพลิเคชัน CYBLION และข้อมูลการยืนยันตัวตนด้วยยูสเซอร์เนม (username) และพาสเวิร์ด ซึ่งจะถูกส่งไปตรวจสอบกับฐานข้อมูล User Whitelist และส่งไปตรวจสอบกับแพลตฟอร์มเน็ตพายผ่านส่วนต่อประสานโปรแกรมประยุกต์ของเน็ตพาย เมื่อผู้ใช้งานได้รับการยืนยันตัวตน ผู้ใช้งานจะเข้าระบบเพื่อทำการสร้างโพรเจกต์ประกอบด้วยกลุ่มของอุปกรณ์เอดจ์คอมพิวติงเสมือน (virtual edge computing devices) บนแอปพลิเคชันโทรศัพท์มือถือ ภายในโพรเจกต์ผู้ใช้สามารถสร้างเอดจ์คอมพิวติงเสมือนได้หลายตัวตามจำนวนโควตาที่กำหนด และภายในโพรเจกต์ผู้ใช้งานสามารถสร้างชุดกุญแจโฮโมมอร์ฟิกซึ่งจะเฉพาะเจาะจงกับโพรเจกต์นั้น จากนั้นผู้ใช้จะส่งกุญแจสาธารณะไปยังเอดจ์คอมพิวติงกายภาพที่ติดตั้งเชื่อมต่อกับอุปกรณ์ไอโอทีหรือเซนเซอร์ และจะส่งกุญแจรีลีเนียร์ไปยังคลาวด์ CYBLION เมื่อเอดจ์คอมพิวติงรับข้อมูลดิบจากเซนเซอร์ที่เชื่อมต่อมาจะเข้ารหัสข้อมูลด้วยกุญแจสาธารณะ และส่งข้อมูลที่เข้ารหัสแล้วไปยังโบรกเกอร์ FHE MQTT บนคลาวด์ผู้ใช้จะล็อกอินเข้าไปยังคลาวด์ผ่านเว็บเพื่อเลือกรับข้อมูลที่เข้ารหัสมาคำนวณและสร้างสูตรคำนวณที่ต้องการ ข้อมูลผลลัพธ์จะถูกส่งออกไปยังโบรกเกอร์ MQTT เพื่อรอให้แอปพลิเคชัน CYBLION นำไปถอดรหัสด้วยกุญแจสำหรับถอดรหัส และแสดงผลบนแดชบอร์ด

6.3 ปัญหาและความท้าทายในการออกแบบแพลตฟอร์ม

6.3.1 โบรกเกอร์ MQTT ทั่วไปไม่รองรับข้อมูลที่เข้ารหัสแบบโฮโมมอร์ฟิก เนื่องจากข้อมูลที่เข้ารหัสมีขนาดใหญ่ ประมาณ 356 KB ถึง 2.2 MB ต่อ 1 ข้อความ ในขณะที่โบรกเกอร์ MQTT ทั่วไปรองรับขนาดข้อความได้ไม่เกิน 128 KB ผู้วิจัยจึงต้องพัฒนาโบรกเกอร์ MQTT ขึ้นมาเพื่อให้รองรับการส่งข้อมูลเข้ารหัส FHE ที่มีขนาดใหญ่ได้

6.3.2 การถอดรหัสข้อมูลที่เข้ารหัสแบบโฮโมมอร์ฟิกและแสดงข้อมูลแบบเรียลไทม์ เดิมทีข้อมูลไอโอทีจะถูกส่ง ข้อมูล 1 จุดต่อ 1 เพย์โหลด (payload) แต่การเข้ารหัส FHE แบบ RNS-CKKS เป็นแบบชุดข้อมูลเพื่อเพิ่มประสิทธิภาพทั้งในด้านการคำนวณเข้ารหัสและแง่การส่งข้อมูล แต่ข้อเสียคือ การรอเข้ารหัสและส่งข้อมูลเมื่อครบชุด ในขณะที่ประโยชน์หลักของการใช้ระบบไอโอทีคือ การติดตามสถานะของข้อมูลได้ตามเวลาจริง ดังนั้น ผู้วิจัยต้องหาสมดุลระหว่างการประหยัดแบนด์วิดท์ (bandwidth) จากการบันทึกข้อมูลลงชุดข้อมูล (batch size) เพื่อที่จะเข้ารหัสและส่งข้อมูลพร้อมกันครั้งละจำนวนมากและความเป็นปัจจุบันของข้อมูลของฝั่งผู้รับเพื่อที่จะนำผลข้อมูลล่าสุดมาแสดงบนแดชบอร์ดแบบเวลาจริง

6.3.3 การกำหนดพารามิเตอร์สำหรับการสร้างสภาพแวดล้อมโฮโมมอร์ฟิกที่เหมาะสมกับฟังก์ชันทางคณิตศาสตร์ เนื่องจากพารามิเตอร์สำหรับการสร้างสภาพแวดล้อมโฮโมมอร์ฟิก มีความสำคัญอย่างยิ่งต่อการนำข้อมูลที่ถูกเข้ารหัสไปคำนวณ อีกทั้งยังเป็นตัวกำหนดระดับความปลอดภัยของการเข้ารหัสข้อมูล ซึ่งต้องสอดคล้องกับมาตรฐานความปลอดภัยการเข้ารหัสโฮโมมอร์ฟิก (Albrecht et al., 2019; Bossuat et al., 2024) ดังตารางที่ 2

ตารางที่ 2 มาตรฐานความปลอดภัยของการเข้ารหัสโฮโมมอร์ฟิก

N	จำนวนบิตของ *Q*		
	128-bit security	192-bit security	256-bit security
1024	27	19	14
2048	54	37	29
4096	109	75	58
8192	218	152	118
16384	438	300	237
32768	881	600	476

ที่มา: Bossuat et al. (2024)

พารามิเตอร์ที่เกี่ยวข้องในกระบวนการเข้ารหัสโฮโมมอร์ฟิกแบบ RNS-CKKS ประกอบด้วย โพลีโมดูลัส N สเกลแฟกเทอร์ Δ โมดูลัสสัมประสิทธิ์ Q เนื่องจากวิธี RNS-CKKS จะมีการปรับขนาดข้อมูลรบกวนในแต่ละลำดับชั้นของการเข้ารหัส ทำให้การกำหนดค่าโมดูลัสสัมประสิทธิ์ Q ซึ่งเป็นผลคูณของค่าสัมประสิทธิในแต่ละขั้น q_i จะต้องสอดคล้องกับค่าสเกลแฟกเทอร์ Δ ที่ใช้กำหนดขนาดความทนทานต่อข้อมูลรบกวน นอกจากนี้พารามิเตอร์กระบวนการเข้ารหัสโฮโมมอร์ฟิกแบบ RNS-CKKS ยังต้องสอดคล้องกับพารามิเตอร์ความปลอดภัย λ ที่แสดงถึงจำนวนบิตความปลอดภัย โดยอ้างอิงตามระดับความปลอดภัยของการเข้ารหัสแบบ AES จำนวนบิตจะเป็นตัวกำหนดค่าโมดูลัสสัมประสิทธิ์ Q และค่าโพลีโมดูลัส N โดยที่ขนาดของ Q จะเป็นตัวกำหนดขนาดของข้อมูลที่สามารถทนทานต่อข้อมูลรบกวนและจำนวนชั้นความลึกของข้อมูลเข้ารหัส (Level) ที่กำหนดจำนวนการดำเนินการของการคูณ นั่นคือ $Q = \prod_{i=0}^{L-1} q_i$ เมื่อ i คือ ชั้นของการเข้ารหัส และ L คือ จำนวนลำดับชั้นสูงสุดของการเข้ารหัส และ $[q_0, \ldots, q_L]$ คือ โซ่โมดูลัสสัมประสิทธิ์ (Coefficient Modulus Chain) ภายหลังการคูณกันของข้อมูลเข้ารหัสจะต้องทำโมดูลัสสวิตชิงด้วยการหารด้วยสเกลแฟกเทอร์ Δ ซึ่งเสมือนข้อมูลเข้ารหัสถูกเลื่อนระดับชั้น จาก i ไปยัง $i-1$ ดังนั้นการเลือกสเกลแฟกเทอร์จะมีผลต่อการควบคุมขนาดของข้อมูลรบกวนในแต่ละการทำการดำเนินการทางคณิตศาสตร์ นอกจากนี้ค่าโมดูลัสสัมประสิทธิ์ Q ที่เลือกจะเป็นตัวกำหนดขนาดของโพลีโมดูลัส N ซึ่งหมายถึง จำนวนข้อมูลที่รองรับในแต่ละชุดข้อมูล (batch) แต่การกำหนดพารามิเตอร์ที่เหมาะสมดังกล่าวไม่ใช่เรื่องง่ายสำหรับผู้ใช้งานทั่วไปที่ไม่มีความเชี่ยวชาญด้านการเข้ารหัสโฮโมมอร์ฟิก หากกำหนดพารามิเตอร์ไม่ถูกต้องอาจจะทำให้ไม่สามารถนำข้อมูลที่เข้ารหัสไปคำนวณได้หรืออาจจะทำให้ข้อมูลไม่ปลอดภัย ดังนั้น ความท้าทายในการออกแบบให้แพลตฟอร์มนี้สามารถใช้งานได้จริงในเชิงพาณิชย์ นั่นคือต้องออกแบบอย่างไร ให้ผู้ใช้สามารถเลือกพารามิเตอร์ดังกล่าวอย่างเหมาะสมกับฟังก์ชันการคำนวณใด ๆ ที่ผู้ใช้งานต้องการ ในขณะที่ผู้ใช้งานไม่ต้องมีความเชี่ยวชาญวิทยาการเข้ารหัสลับ (Cryptography)

6.4 ระเบียบวิธีแก้ปัญหา

6.4.1 การออกแบบโบรกเกอร์ MQTT ของแพลตฟอร์ม CYBLION ให้รองรับข้อมูลเข้ารหัสโฮโมมอร์ฟิก

ผู้วิจัยได้ออกแบบพัฒนาโบรกเกอร์ MQTT ซึ่งเป็นส่วนหนึ่งของคลาวด์ CYBLION ขึ้นมาใหม่ภายใต้ชื่อเอดจ์โบรกเกอร์ (edge broker) ทำหน้าที่เป็นตัวกลางการสื่อสารสองทางสำหรับการรับส่งข้อความตามเวลาจริง โดยมีความสามารถในการกระจายข้อความที่มีเพย์โหลดขนาดใหญ่ที่เกิดจากการเข้ารหัสแบบชุดของ RNS-CKKS โดยกำหนดขนาดสูงสุดไว้ที่ 4 MB เมื่อพิจารณามาตรฐาน OASIS MQTT เวอร์ชัน 3.1.1 พบว่ามีการจำกัดค่าเพย์โหลดของข้อความ MQTT สูงสุดไว้ที่ 260 MB ซึ่งยังมีขนาดใหญ่กว่าขนาดของข้อความเข้ารหัสบนแพลตฟอร์ม CYBLION ดังนั้น การเพิ่มขนาดเพย์โหลดขึ้นเป็น 4 MB จึงยังอยู่ภายใต้มาตรฐานและไม่กระทบกับคุณภาพของบริการ (QoS) ของโบรกเกอร์ อุปกรณ์เอดจ์คอมพิวติงสามารถเชื่อมต่อกับเอดจ์โบรกเกอร์ด้วยโพรโทคอล MQTT และ MQTT over websocket ทำให้รองรับการเชื่อมต่อทั้งแบบ plain MQTT/WS และแบบ over TLS ด้วย MQTTS/WSS

6.4.2 การถอดรหัสข้อมูลโฮโมมอร์ฟิกแบบ RNS-CKKS และการแสดงข้อมูลตามเวลาจริง

RNS-CKKS จะเข้ารหัสข้อมูลแบบชุด กล่าวคือ ข้อมูลหลาย ๆ ตัวจะถูกเรียงกันในรูปแบบแถวลำดับ (array) ก่อนนำไปเข้ารหัสแต่ละครั้ง ดังนั้น จะเห็นได้ว่าในแง่การเข้ารหัสและสื่อสารข้อมูล การบรรจุข้อมูลให้เต็มแถวลำดับเพื่อนำไปเข้ารหัสและส่งรวมกันจะได้ประสิทธิภาพสูงสุด อย่างไรก็ตาม การนำวิธีการดังกล่าวมาใช้งานในระบบไอโอทีโดยไม่ประยุกต์เลยไม่ได้ เนื่องจากข้อมูลของโรงงานอุตสาหกรรมที่ส่งในระบบไอโอทีส่วนใหญ่เป็นค่าที่เก็บจากเซนเซอร์เป็นอนุกรมเวลา เช่น 1 ค่า ต่อ t วินาที เช่น ข้อมูลการวัดค่าการใช้พลังงาน ข้อมูลผลผลิต หรือข้อมูลพารามิเตอร์ต่าง ๆ ของเครื่องจักร จากการสำรวจการใช้งานไอโอทีสำหรับโรงงานอุตสาหกรรมที่เข้าร่วมโครงการทดลองและถ่ายทอดการประยุกต์ใช้เทคโนโลยี 5G สำหรับ smart factory/manufacturing[1] และโครงการ IDA Platform[2] อัตราการส่งข้อมูลอยู่ที่ 1 ข้อมูล ต่อ 15 วินาที หรือ 4 ครั้งต่อนาที หมายความว่า ถ้าต้องการรอให้ข้อมูลถูกบรรจุลงในแถวลำดับซึ่งมีความยาว $\frac{N}{2}$ จนครบ จะต้องใช้เวลา $15 \times \frac{N}{2}$ วินาที (หมายเหตุ N คือ โพลีโมดูลัสของการเข้ารหัสโฮโมมอร์ฟิกและเป็นตัวกำหนดความยาวของแถวลำดับของชุดข้อมูล) ตามที่แพลตฟอร์ม CYBLION ได้กำหนดไว้ คือ $N=2^{13}=8192$ สำหรับพารามิเตอร์การสร้างสภาพแวดล้อมการเข้ารหัสขนาดเล็กและกลาง และ $N=2^{14}=16384$ สำหรับพารามิเตอร์ขนาดใหญ่ ดังนั้น หากรอบรรจุแถวลำดับจนครบก่อนเข้ารหัสและส่งไปยังคลาวด์ ผู้ใช้ปลายทางจะเห็นข้อมูลใหม่ใน 1-2 วัน ซึ่งไม่สอดคล้องกับการใช้งานในระบบไอโอที จึงต้องมีการออกแบบการส่งข้อมูลใหม่โดยไม่ต้องรอข้อมูลถึง $N/2$ ตัว และเพิ่มความเชื่อถือได้ (reliability) ด้วยการส่งข้อมูลด้วยการส่งซ้ำ (redundancy) บางส่วนในแต่ละแถวลำดับ

6.4.3 การกำหนดพารามิเตอร์สำหรับการสร้างสภาพแวดล้อมโฮโมมอร์ฟิกให้เหมาะสมกับฟังก์ชันทางคณิตศาสตร์

เมื่อพิจารณาโมดูลของการเข้ารหัสโฮโมมอร์ฟิกแบบ RNS-CKKS กระบวนการจะเริ่มจากการแปลงข้อมูลดิบซึ่งเป็นเลขจำนวนจริง จะถูกแปลงเป็นรหัส (encode) ให้กลายเป็นเพลนเท็กซ์ (plain text) ในขั้นนี้ มีการคูณข้อมูลตัวเลขทศนิยมด้วยสเกลแฟกเทอร์ เพื่อให้อยู่ในรูปของจำนวนเต็ม ขนาดของสเกลแฟกเทอร์เป็นตัวกำหนดจำนวนหลักทศนิยมและความทนทานต่อข้อมูลรบกวนที่จะเกิดขึ้นภายหลังกระบวนการคำนวณ จากนั้นเพลนเท็กซ์ที่ได้จะถูกนำไปเข้ารหัสด้วยกุญแจสาธารณะ (pk) ให้กลายเป็นไซเฟอร์เท็กซ์ ซึ่งสามารถนำไปคำนวณต่อได้บนคลาวด์โดยมีกุญแจรีลีเนียร์ (rk) ทำหน้าที่ลดขนาดของข้อมูล ภายหลังการดำเนินการคูณเนื่องจากขนาดของข้อมูลจะเพิ่มขึ้นทำให้ไม่สามารถนำไปคำนวณต่อได้ จากนั้นผลลัพธ์จากการคำนวณที่อยู่ในรูปของไซเฟอร์เท็กซ์จะถูกส่งไปยังแอปพลิเคชันบนโทรศัพท์มือถือเพื่อถอดรหัสด้วยกุญแจรหัสลับ (sk) ออกมาเป็นเพลนเท็กซ์ และถูกแปลงกลับ (decoding) เพื่อให้ได้ผลลัพธ์การคำนวณกลับมา ความถูกต้องแม่นยำของผลลัพธ์ที่ได้ขึ้นอยู่กับพารามิเตอร์การแปลงเป็นรหัสและการเข้ารหัส เพราะการเปลี่ยนจำนวนจริงให้เป็นจำนวนเต็มหรือจำนวนที่มีทศนิยมตายตัว เป็นการประมาณค่าที่ก่อให้เกิดความคลาดเคลื่อนอย่างหลีกเลี่ยงไม่ได้ แต่การกำหนดพารามิเตอร์ที่เหมาะสมจะช่วยลดความคลาดเคลื่อนดังกล่าวให้อยู่ในช่วงที่ยอมรับและใช้งานได้

[1] ศึกษาเพิ่มเติม https://btfp.nbtc.go.th/Newsandactivities/activity/2162.aspx

[2] ศึกษาเพิ่มเติม https://www.nectec.or.th/smc/ida-platform/

การกำหนดค่าพารามิเตอร์สำหรับการสร้างสภาพแวดล้อมโฮโมมอร์ฟิกมีผลต่อความแม่นยำของผลลัพธ์ภายหลังการถอดรหัส ความปลอดภัยจากการถูกโจมตีทางไซเบอร์ และสัมพันธ์โดยตรงกับขนาดทรัพยากรการคำนวณที่ต้องใช้ ดังนั้น การเลือกค่าพารามิเตอร์สำหรับ RNS-CKKS จึงมีความซับซ้อนและละเอียดอ่อน ต้องอาศัยผู้เชี่ยวชาญในวิทยาการเข้ารหัสแบบ RNS-CKKS เพราะนอกจากปัจจัยเรื่องประสิทธิภาพและทรัพยากรแล้ว การกำหนดค่าพารามิเตอร์โดยพื้นฐานจำเป็นต้องเป็นไปตามมาตรฐานความปลอดภัยทางไซเบอร์ดังแสดงในตารางที่ 3

ตารางที่ 3 ตัวเลือกพารามิเตอร์สภาพแวดล้อมโฮโมมอร์ฟิกที่เสนอ

Parameter Size (KB)	Polymodulus N	Coefficient Modulus Chain $[q_0,\ldots,q_L]$	Scale Factor Δ	No. Multiply Operations	Encrypted Data Size (KB)	Key Size (KB)		
						Public Key	Relinear Key	Secret Key
Small 140	8192	$[2^{50}, 2^{30}, 2^{30}, 2^{50}]$	2^{30}	2	356	258	771	255
Medium 146	8192	$[2^{60}, 2^{40}, 2^{40}, 2^{60}]$	2^{40}	2	446	314	940	314
Large 174	16384	$[2^{60}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{60}]$	2^{40}	7	2,200	1,300	10,200	1,300

การเลือกค่าสเกลแฟกเทอร์ Δ มีผลต่อความทนทานต่อข้อมูลรบกวนที่เกิดจากกระบวนการคำนวณ ในขณะที่โพลีโมดูลัส N มีผลต่อความแม่นยำในการถอดรหัส และมีผลต่อขนาดของข้อมูลที่เข้ารหัสอีกด้วย ถ้าหากกำหนดให้ N มีค่าน้อยก็จะส่งผลทำให้ค่า Q มีลิมิตน้อยตามไปด้วย เช่น ถ้ากำหนดให้ N=1024 จะทำให้ค่า Q ที่ทำให้การเข้ารหัสตามมาตรฐานความปลอดภัยระดับ 128 บิต มีค่าได้ไม่เกิน 2^{27} ซึ่งจะส่งผลไปถึง q_i และ L ตามไปด้วย โดยที่ L เป็นตัวกำหนดจำนวนครั้งที่สามารถทำการคูณได้ ดังนั้น การเลือกค่า N น้อย จะเพิ่มข้อจำกัดการใช้งานด้านการคำนวณ แต่หากเลือกค่า N ให้สูง ก็จะส่งผลให้ข้อมูลที่เข้ารหัสมีขนาดใหญ่และมีข้อมูลรบกวนสูงตามไปด้วย

จะเห็นได้ว่า เป็นไปไม่ได้ที่ผู้ใช้งานทั่วไปจะสามารถกำหนดพารามิเตอร์สภาพแวดล้อมโฮโมมอร์ฟิกได้อย่างถูกต้องด้วยตนเอง ดังนั้น นักวิจัยจึงต้องพัฒนาวิธีการเลือกพารามิเตอร์ที่เหมาะสมกับฟังก์ชันการคำนวณได้อย่างง่าย และพารามิเตอร์ที่ให้เลือกต้องมีความยืดหยุ่นต่อการปรับแต่งฟังก์ชันการคำนวณได้อีกด้วย โดยกำหนดให้มีตัวเลือกพารามิเตอร์สภาพแวดล้อมโฮโมมอร์ฟิกจำนวน 3 ชุด ได้แก่ ขนาดเล็ก (small) ขนาดกลาง (medium) และขนาดใหญ่ (large) ดังตารางที่ 3 ซึ่งแสดงผลการทดลองและหาจุดสมดุลของค่าพารามิเตอร์ทั้ง 3 ขนาด ให้เหมาะสมกับการใช้งาน โดยผู้ใช้งานสามารถเลือกใช้งานได้ตามจำนวนครั้งของการคูณที่ต้องการได้ เช่น หากสูตรที่ต้องการคำนวณมีจำนวนครั้งของการคูณไม่เกิน 3 ครั้ง สามารถเลือกใช้งานพารามิเตอร์สภาพแวดล้อมโฮโมมอร์ฟิกขนาดเล็กหรือขนาดกลางได้ ทั้งนี้ ขึ้นกับความแม่นยำของผลลัพธ์ที่ต้องการ ในขณะที่พารามิเตอร์ขนาดใหญ่เหมาะสมกับกรณีในสูตรคำนวณมีการหาร

6.4.5 การเชื่อมต่อผู้ใช้งานคลาวด์ CYBLION กับคลาวด์เน็ตพาย

เนื่องจากผู้พัฒนาต้องการให้ผู้ใช้งานบนแพลตฟอร์มเน็ตพายเดิม มีทางเลือกในการใช้บริการข้อมูลไอโอทีอย่างปลอดภัยและคุ้มครองความเป็นส่วนตัวของข้อมูล ดังนั้น แพลตฟอร์ม CYBLION จึงถูกออกแบบให้การควบคุมการเข้าใช้งานคลาวด์ CYBLION ของผู้ใช้ เป็นไปตามกฎระเบียบของคลาวด์เน็ตพายและคลาวด์ CYBLION กล่าวคือ ผู้ใช้งานที่ลงทะเบียนกับแพลตฟอร์ม CYBLION จะถูกบันทึกใน Whitelist การยืนยันตัวตนของผู้ใช้งาน แพลตฟอร์ม CYBLION จะต้องผ่านการตรวจสอบสิทธิ์ในฐานข้อมูล Whitelist และมีการส่ง User token ไปตรวจสอบกับคลาวด์เน็ตพาย

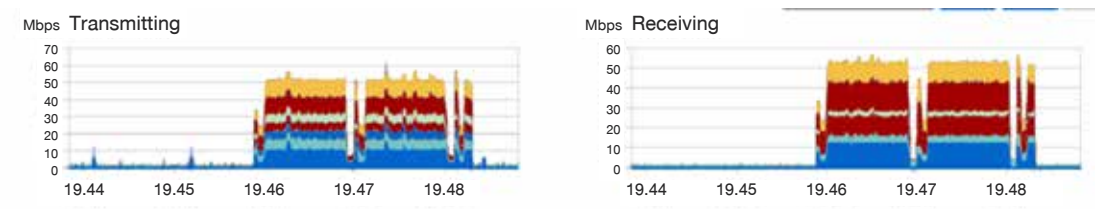
7. ผลการศึกษา

7.1 การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลของอุปกรณ์ไอโอทีที่สามารถประยุกต์ใช้กับงานภาคอุตสาหกรรมและทดสอบประสิทธิภาพการใช้งานจริงของแพลตฟอร์ม

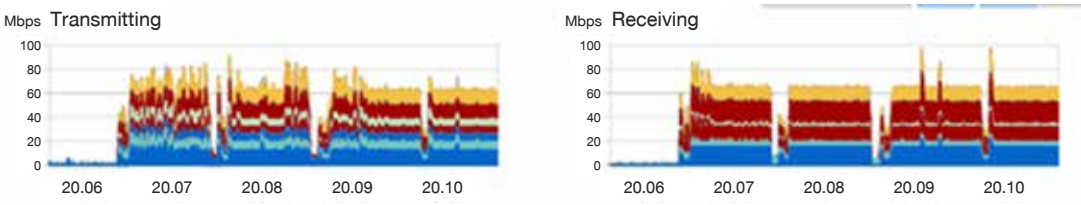
การศึกษาในบทความนี้ประกอบด้วยการทดสอบประสิทธิภาพของแพลตฟอร์มที่พัฒนาขึ้นเพื่อหาข้อจำกัดของแพลตฟอร์มด้วยการทดสอบด้านความสามารถของคลาวด์ CYBLION ในการรับข้อมูลมาประมวลผล และความถูกต้องในการถอดรหัสข้อมูล นอกจากนี้เพื่อแสดงให้เห็นว่าแพลตฟอร์มที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้งานได้จริงกับงานภาคอุตสาหกรรม

7.1.1 การทดสอบประสิทธิภาพของแพลตฟอร์ม

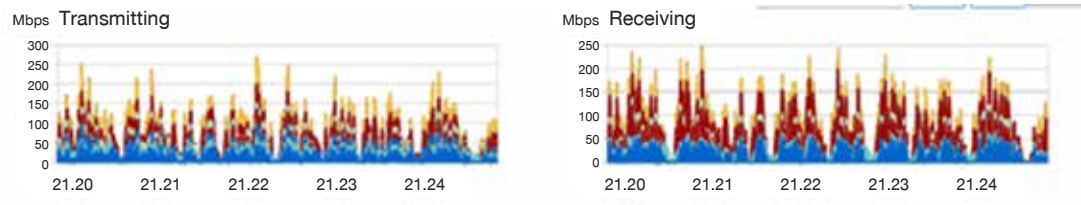
ผู้วิจัยได้ทำการทดสอบโดยให้เอดจ์คอมพิวติงเข้ารหัสด้วยกุญแจขนาดเล็ก ขนาดกลาง และขนาดใหญ่ ส่งข้อมูลด้วยความเร็วที่ 60, 30, 15, 5, 3, 2, 1 และ 0.5 วินาทีต่อข้อมูล มายังคลาวด์ CYBLION และวัดปริมาณข้อมูลที่คลาวด์รับมาประมวลผล (receiving) และส่งออกไปยังแอปพลิเคชันบนโทรศัพท์มือถือ CYBLION (transmitting) ผลที่ได้จากการทดสอบดังภาพที่ 2 ถึงภาพที่ 4 ความเร็วสูงสุดของการส่งข้อมูลเข้ารหัสจากเอดจ์คอมพิวติงที่คลาวด์สามารถประมวลผลได้โดยไม่มี packet drop เมื่อข้อมูลถูกเข้ารหัสด้วยกุญแจขนาดเล็กและขนาดกลาง ความเร็วสูงสุดของการส่งข้อมูลอยู่ที่ 0.5 วินาทีต่อข้อมูล ในขณะที่เมื่อส่งข้อมูลที่เข้ารหัสด้วยกุญแจขนาดใหญ่ ความเร็วสูงสุดของการส่งข้อมูลอยู่ 5 วินาทีต่อข้อมูล



ภาพที่ 2 อัตราการรับ-ส่งข้อมูลของคลาวด์ CYBLION เมื่อเอดจ์คอมพิวติงส่งข้อมูลด้วยความเร็ว 0.5 วินาทีต่อข้อมูล เมื่อเข้ารหัสด้วยกุญแจขนาดเล็ก



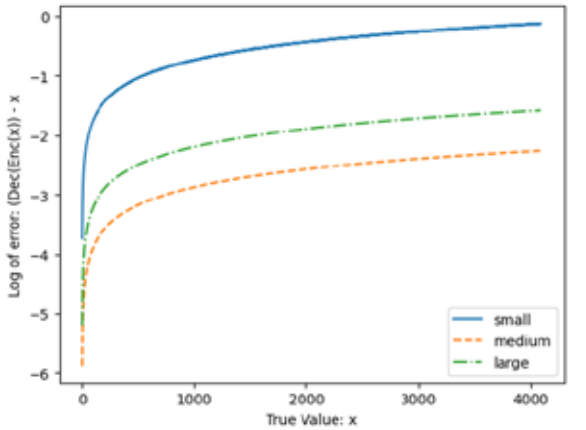
ภาพที่ 3 อัตราการรับ-ส่งข้อมูลของคลาวด์ CYBLION เมื่อเอดจ์คอมพิวติงส่งข้อมูลด้วยความเร็ว 0.5 วินาทีต่อข้อมูล เมื่อเข้ารหัสด้วยกุญแจขนาดกลาง



ภาพที่ 4 อัตราการรับ-ส่งข้อมูลของคลาวด์ CYBLION เมื่อเอดจ์คอมพิวติงส่งข้อมูลด้วยความเร็ว 5 วินาทีต่อข้อมูล เมื่อเข้ารหัสด้วยกุญแจขนาดใหญ่

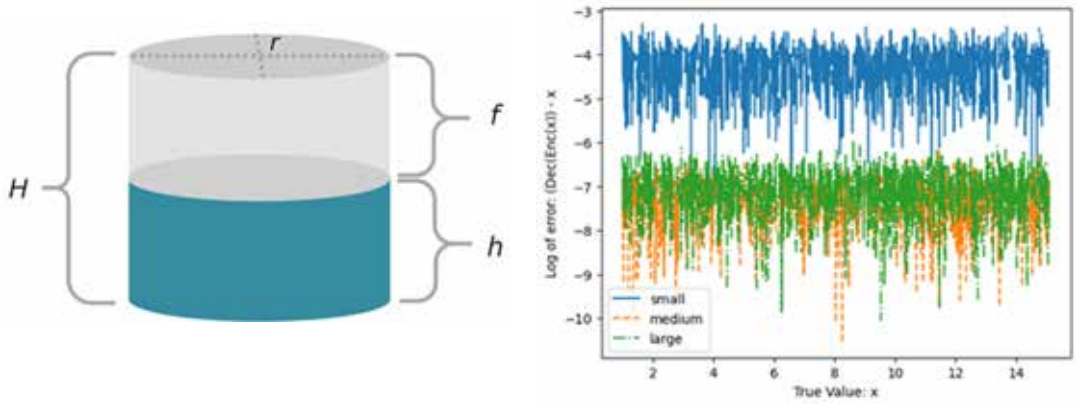
7.1.2 การทดสอบความแม่นยำของการถอดรหัส

การศึกษานี้ได้เสนอผลการทดสอบช่วงการใช้งานของแพลตฟอร์ม CYBLION โดยทดสอบความถูกต้องของผลลัพธ์ที่ได้จากการคำนวณข้อมูลที่เข้ารหัสเมื่อข้อมูลถูกเข้ารหัสด้วยกุญแจขนาดเล็ก ขนาดกลาง และขนาดใหญ่ โดยทดสอบเฉพาะการคูณ เนื่องจากการคูณจะส่งผลต่อความคลาดเคลื่อนสูงกว่าการบวก ฟังก์ชันที่ใช้ทดสอบเป็นไปตามสมการดังนี้ $z=Enc_k(x) \times Enc_k(y)$ โดย x มีค่าในช่วง 0 ถึง 4,000 และ y มีค่าเท่ากับ 2 ผลที่ได้จากการทดสอบค่าความคลาดเคลื่อนของการคำนวณข้อมูลเข้ารหัสด้วยกุญแจขนาดต่าง ๆ เปรียบเทียบกับการคำนวณบนค่าจริง แสดงผลในรูปของลอการิทึม (logarithm) ฐานสิบของผลต่างของทั้งสองค่า ซึ่งแสดงถึงจำนวนทศนิยมที่เริ่มคลาดเคลื่อน ดังภาพที่ 5



ภาพที่ 5 ผลการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณ เมื่อข้อมูลถูกเข้ารหัสด้วยกุญแจขนาดเล็ก ขนาดกลาง และขนาดใหญ่ กับค่าที่ได้จากการคำนวณค่าจริง ตามสมการ $z=Enc_k(x) \times Enc_k(y)$ เมื่อ y มีค่าเท่ากับ 2

จากนั้นทดสอบความถูกต้องความแม่นยำในการถอดรหัสจากผลการคำนวณตามสมการที่ใช้ในกรณีศึกษาการวัดปริมาตรน้ำมันในถังน้ำมัน โดยเซนเซอร์จะวัดค่าระยะทางจากขอบถังถึงผิวของเหลว f และสามารถหาความสูงของน้ำมันในถัง h ได้จากความสูงของถัง H และค่าปริมาตรของน้ำมันในถังคำนวณได้จากสมการ $v = \pi r^2 (H - f)$ (หน่วยลูกบาศก์เมตร) ผลที่ได้จากการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณเมื่อข้อมูลถูกเข้ารหัสด้วยกุญแจขนาดเล็ก ขนาดกลาง และขนาดใหญ่ กับค่าที่ได้จากการคำนวณค่าจริงเป็นดังภาพที่ 6 โดย ก. แสดงตัวแปรสำหรับการคำนวณปริมาตรของน้ำมันในถัง และ ข. แสดงค่าความคลาดเคลื่อนจากการถอดรหัสผลลัพธ์จากการคำนวณ โดยผลลัพธ์ข้อมูลที่ถูกเข้ารหัสด้วยกุญแจขนาดเล็ก มีค่าความคลาดเคลื่อนโดยเฉลี่ยอยู่ที่ทศนิยม 4 ตำแหน่ง ในขณะที่ค่าความคลาดเคลื่อนของผลลัพธ์ข้อมูลที่ถูกเข้ารหัสด้วยกุญแจขนาดกลางและขนาดใหญ่ มีค่าความคลาดเคลื่อนเฉลี่ยที่ทศนิยม 7 ตำแหน่ง



ก. ถังใส่น้ำมัน ข. ผลการเปรียบเทียบค่าความคลาดเคลื่อน

ภาพที่ 6 ผลการเปรียบเทียบค่าความคลาดเคลื่อนของการคำนวณเมื่อข้อมูลถูกเข้ารหัสด้วยกุญแจขนาดเล็ก ขนดกลาง และขนาดใหญ่ กับค่าที่ได้จากการคำนวณค่าจริงตามสมการ $v = \pi r^2 (H - f)$

7.1.3 การทดสอบการใช้งานแพลตฟอร์มกับกรณีศึกษาของโรงงานอุตสาหกรรม

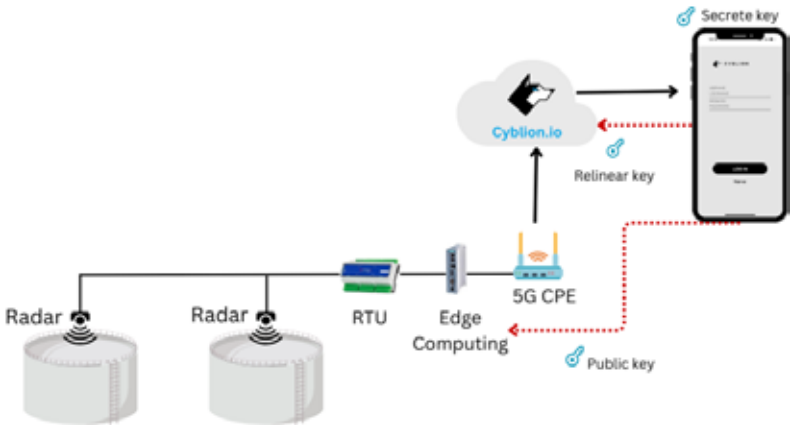
ผู้วิจัยทำการทดสอบกับกรณีศึกษาของโรงงานอุตสาหกรรม โดยนำไปทดสอบกับระบบไอโอทีในโรงงานอุตสาหกรรมสองแห่ง ได้แก่

7.1.3.1 การนำไปประยุกต์ใช้งานกับกรณีศึกษาของบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด สำหรับระบบตรวจสอบระดับน้ำมันในไซโล

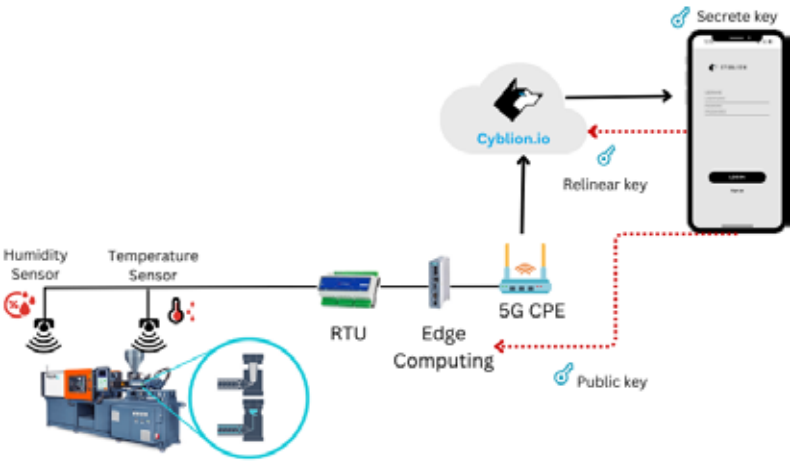
7.1.3.2 การนำไปประยุกต์ใช้งานกับกรณีศึกษาของบริษัท เดอะเพ็ท จำกัด สำหรับระบบตรวจสอบสภาวะของเครื่องจักร

เริ่มต้นจากขั้นตอนการออกแบบไดอะแกรมการเชื่อมต่อของอุปกรณ์ต่าง ๆ เพื่อนำไปติดตั้งระบบไอโอทีสำหรับตรวจสอบระดับน้ำมันเป็นดังภาพที่ 7 โดยอุปกรณ์เซนเซอร์สำหรับรับข้อมูล ในที่นี้ผู้วิจัยได้ติดตั้งเรดาร์ (radar) สำหรับวัดปริมาณของเหลวด้วยแสงอินฟราเรด (infrared) และข้อมูลจะถูกส่งไปเข้ารหัส

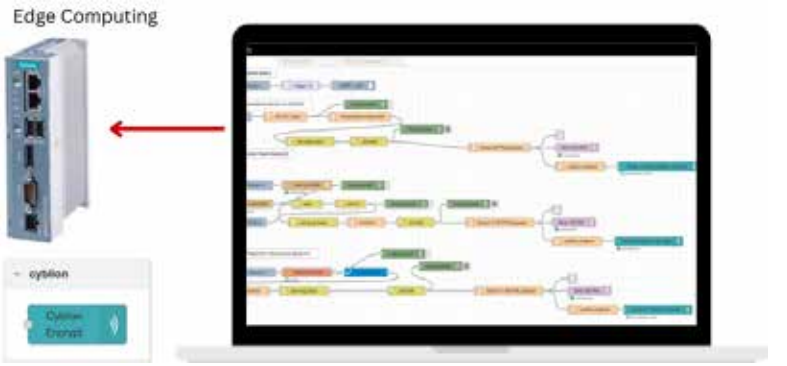
ที่เอดจ์คอมพิวติง จากนั้นข้อมูลที่เข้ารหัสจะถูกส่งไปคำนวณบนคลาวด์แพลตฟอร์ม CYBLION ในทำนองเดียวกัน ผู้วิจัยได้ออกแบบระบบเพื่อนำไปติดตั้งระบบไอโอที สำหรับตรวจสอบสภาวะของเครื่องจักรเป็นดังภาพที่ 8 ข้อมูลจากไอโอทีจะถูกเข้ารหัสบนเอดจ์คอมพิวติง โดยผู้ใช้งานสามารถเข้าไปกำหนดข้อมูลที่ต้องการเข้ารหัสได้ ดังภาพที่ 9 ผู้ใช้งานสามารถเข้าไปกำหนดสูตรการคำนวณเองได้บนคลาวด์ดังภาพที่ 10 และภาพแสดงข้อมูลที่คลาวด์มองเห็นเป็นข้อมูลที่ถูกเข้ารหัส ผลลัพธ์ที่ได้จากคลาวด์จะถูกส่งมาถอดรหัสที่แอปพลิเคชันบนโทรศัพท์มือถือ การทดสอบทำโดยส่งข้อมูลไอโอทีด้วยความเร็ว 1 ข้อความต่อ 5 วินาที และตรวจสอบข้อมูลที่ได้จากการประมวลผลบนคลาวด์และการแสดงผลการถอดรหัสบนแอปพลิเคชันโทรศัพท์มือถือ ผลที่ได้จากการคำนวณข้อมูลแบบเข้ารหัสโฮโมมอร์ฟิกบนคลาวด์แพลตฟอร์ม CYBLION และถอดรหัสซึ่งแสดงผลผ่านแอปพลิเคชันบนโทรศัพท์มือถือ เปรียบเทียบกับผลที่ได้จากค่าจริงดังภาพที่ 11 โดยผลที่ได้มีค่าเท่ากันเมื่อแสดงระดับทศนิยม 2 ตำแหน่ง



ภาพที่ 7 ไดอะแกรมระบบตรวจสอบระดับน้ำมันในไซโลด้วยไอโอทีสำหรับแพลตฟอร์มที่เสนอ



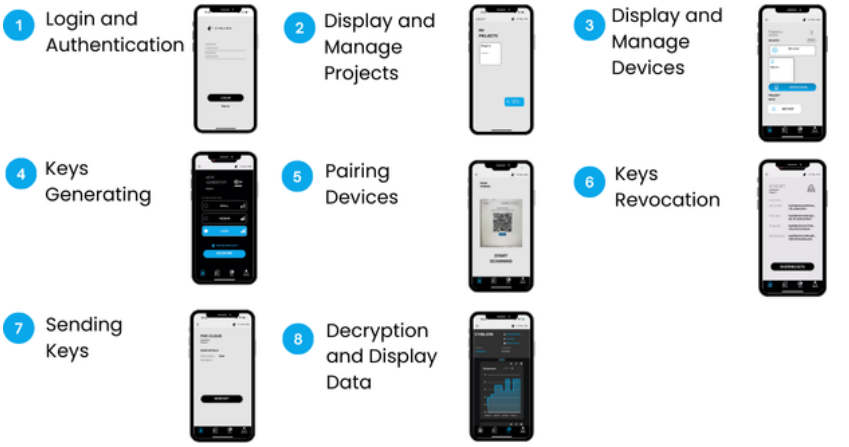
ภาพที่ 8 ไดอะแกรมระบบตรวจสอบสภาวะของเครื่องจักรด้วยไอโอทีสำหรับแพลตฟอร์มที่เสนอ



ภาพที่ 9 ข้อมูลไอโอทีสำหรับเข้ารหัสที่เอดจ์คอมพิวติง

ภาพที่ 10 ข้อมูลที่เข้ารหัสถูกคำนวณบนคลาวด์แพลตฟอร์ม

ภาพที่ 11 เปรียบเทียบข้อมูลที่ถูกถอดรหัสที่แสดงผลบนแอปพลิเคชัน CYBLION
และข้อมูลจริงบนจอแสดงผลของอุปกรณ์เรดาร์



ภาพที่ 12 การควบคุมการเข้ารหัสผ่านแอปพลิเคชันบนโทรศัพท์มือถือที่มีหน้าต่างอินเทอร์เฟซ (UI)

7.2 การพัฒนาแพลตฟอร์มที่ผู้ใช้งานไม่จำเป็นต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัส

การออกแบบแพลตฟอร์ม CYBLION สำหรับการใช้งานเพื่อให้ผู้ใช้งานไม่จำเป็นต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัสสามารถใช้งานได้ด้วยตนเอง โดยผู้วิจัยได้พัฒนาให้การควบคุมการเข้ารหัสผ่านแอปพลิเคชันบนโทรศัพท์มือถือที่มีหน้าต่างอินเทอร์เฟซ (UI) ดังภาพที่ 12 ให้ผู้ใช้สามารถเลือกพารามิเตอร์การเข้ารหัสข้อมูลจากสามตัวเลือก (S, M, L) ตามขนาดความซับซ้อนของสูตรที่ต้องการคำนวณ ทำให้ผู้ใช้งานไม่ต้องมีความรู้เรื่องการกำหนดพารามิเตอร์การเข้ารหัสแบบโฮโมมอร์ฟิกที่เหมาะสมเอง และอีกทั้งขั้นตอนการเรียกคืนกุญแจ (key revocation) สามารถทำได้ผ่านแอปพลิเคชันบนโทรศัพท์มือถือเช่นกัน โดยภาพที่ 13 แสดงให้เห็นถึงการทดสอบให้วิศวกรในโรงงานอุตสาหกรรมทดลองใช้งานแพลตฟอร์มโดยที่วิศวกรดังกล่าวไม่มีความเชี่ยวชาญด้านวิทยาการเข้ารหัส

ภาพที่ 13 การทดสอบใช้งานแพลตฟอร์ม โดยวิศวกรโรงงานที่ไม่มีความเชี่ยวชาญด้านวิทยาการเข้ารหัส

8. การอภิปรายผล

8.1 การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลของอุปกรณ์ไอโอทีที่สามารถประยุกต์ใช้กับงานภาคอุตสาหกรรม

ผลการทดสอบชี้ให้เห็นว่าแพลตฟอร์ม CYBLION ใช้ประโยชน์ได้จริง ผลที่ได้จากการทดสอบแสดงถึงข้อจำกัดของการนำไปใช้ในการคำนวณความถูกต้องระดับทศนิยมไม่เกิน 4 ตำแหน่ง สำหรับการเข้ารหัสด้วยกุญแจขนาดเล็ก และความถูกต้องระดับทศนิยมไม่เกิน 7 ตำแหน่งสำหรับการเข้ารหัสด้วยกุญแจขนาดกลางและขนาดใหญ่ ความเร็วในการส่งข้อมูลไปคำนวณบนคลาวด์ถูกจำกัดที่การส่ง 0.5 วินาทีต่อ 1 ชุดข้อมูลเมื่อเข้ารหัสด้วยกุญแจขนาดเล็ก และ 5 วินาทีต่อข้อมูลเมื่อเข้ารหัสด้วยกุญแจขนาดใหญ่

เมื่อทดสอบกับการตรวจวัดค่าของเครื่องจักรด้วยไอโอทีที่ใช้งานในโรงงานอุตสาหกรรมอยู่ในระดับที่ผู้ใช้งานรับได้ โดยประเมินจากทั้งความเร็วในการแสดงผลบนแอปพลิเคชัน ความแม่นยำของข้อมูลที่ถอดรหัสเทียบกับการคำนวณบนข้อมูลจริง ความปลอดภัยและความเป็นส่วนตัวของข้อมูลถูกสงวนไว้สำหรับเจ้าของข้อมูล เมื่อนำแพลตฟอร์ม CYBLION ไปทดสอบกับกรณีศึกษาในโรงงานอุตสาหกรรมพบว่า ระบบสามารถทำงานได้ดีและผลการถอดรหัสและแสดงผลผ่านแอปพลิเคชันบนโทรศัพท์มือถือ เมื่อเปรียบเทียบกับการใช้งานแอปพลิเคชันบนมือถือที่แสดงผลข้อมูลไอโอทีแบบไม่เข้ารหัส ผู้ใช้งานไม่รู้สึกถึงความแตกต่าง และการทดสอบแสดงให้เห็นว่าคลาวด์ไม่สามารถเรียนรู้ข้อมูลที่คำนวณได้

8.2 การพัฒนาแพลตฟอร์มที่ผู้ใช้งานไม่จำเป็นต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัส

ผลการทดสอบกับผู้ใช้งานในโรงงานอุตสาหกรรม เช่น วิศวกรควบคุมเครื่องจักร และพนักงานฝ่ายเทคโนโลยีสารสนเทศ พบว่า ผู้ใช้งานสามารถเรียนรู้และใช้งานได้โดยไม่ต้องอธิบายทฤษฎีวิทยาการเข้ารหัส FHE ทำให้ผู้ใช้งานสามารถเข้าใจวิธีการใช้งานได้อย่างรวดเร็ว ผ่านการอ่านคู่มือประกอบการใช้งาน และผลการสำรวจความพึงพอใจของการออกแบบอยู่ในระดับดี

9. ข้อสรุป

การศึกษานี้ชี้ให้เห็นถึงข้อจำกัดในการให้บริการของแพลตฟอร์มไอโอทีบนคลาวด์ทั่วไปในท้องตลาดที่ไม่สามารถป้องกันการเปิดเผยข้อมูลให้กับคลาวด์ที่ให้บริการคำนวณได้ ซึ่งไม่สอดคล้องกับความต้องการของผู้ใช้งานในภาคอุตสาหกรรมที่ต้องการใช้ประโยชน์จากระบบไอโอที โดยที่ความลับของข้อมูลยังคงถูกเก็บรักษาไว้ ด้วยเหตุนี้ ผู้วิจัยจึงพัฒนาแพลตฟอร์มให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบไอโอทีที่ให้บริการคำนวณบนข้อมูลไอโอทีที่เข้ารหัสโดยไม่ถอดรหัส ซึ่งทำให้ข้อมูลไอโอทีไม่ถูกเปิดเผยไปสู่คลาวด์ เมื่อผลลัพธ์ของการคำนวณบนข้อมูลที่เข้ารหัสดังกล่าวถูกถอดรหัสด้วยกุญแจรหัสลับจะมีค่าความถูกต้องในระดับทศนิยม 4 ตำแหน่ง กับการคำนวณบนข้อมูลจริงที่ไม่ถูกเข้ารหัสในกรณีของการสาธิต หลักการทำงานของแพลตฟอร์มที่ถูกพัฒนาขึ้นโดยผู้วิจัยภายใต้ชื่อ CYBLION อาศัยเทคโนโลยีการเข้ารหัสแบบโฮโมมอร์ฟิกสมบูรณ์แบบ RNS-CKKS ทำให้สามารถเข้ารหัสข้อมูลเลขจำนวนที่มีจุดทศนิยมตายตัวได้ โดยหลักการ

ในการออกแบบเน้นให้นักพัฒนาไอโอทีโซลูชัน และผู้ใช้งานโรงงานอุตสาหกรรมสามารถนำไปใช้งานและพัฒนาไอโอทีโซลูชันบนแพลตฟอร์มได้โดยไม่ต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัส และผู้ใช้งานสามารถสร้างสูตรคำนวณเองได้บนคลาวด์ เนื่องจากส่วนสร้างสูตรการคำนวณถูกแบ่งแยกออกจากส่วนเก็บข้อมูลต้นทาง ทำให้ผู้ใช้งานสามารถใช้งานปรับแต่งสูตรการคำนวณ โดยไม่กระทบการทำงานเดิม ข้อมูลที่เข้ารหัสสามารถถอดรหัสและแสดงผลได้ตามเวลาจริง ด้วยการจัดการบรรจุและดึงข้อมูลแบบชุดข้อมูลมาแสดง จากผลการทดสอบประสิทธิภาพในห้องทดลองและการใช้งานจริงในโรงงานอุตสาหกรรมสำหรับกรณีศึกษา แสดงให้เห็นว่าต้นแบบนี้สามารถใช้งานได้จริงและมีประโยชน์กับโรงงาน

10. ข้อเสนอแนะ

10.1 ข้อเสนอแนะสำหรับการศึกษาในอนาคต

แนวทางการศึกษาเพื่อพัฒนาต่อยอดแพลตฟอร์มให้สามารถนำไปใช้งานได้กับแอปพลิเคชันไอโอทีที่หลากหลายมากยิ่งขึ้น และสามารถรองรับการใช้งานได้สำหรับ multi-users ในการเข้าถึงข้อมูลแต่ยังคงรักษาไว้ซึ่งความเป็นส่วนตัวของข้อมูลและความปลอดภัยทางไซเบอร์ เนื่องจากในการใช้งานจริงสำหรับโรงงาน เจ้าของโรงงานอาจต้องการให้ผู้รับผิดชอบส่วนงานที่เกี่ยวข้องสามารถเข้าถึงข้อมูลไอโอทีของโรงงานได้ ซึ่งไม่จำกัดอยู่ที่ผู้ใช้รายเดียว นอกจากนี้ ควรพัฒนาให้แพลตฟอร์มสามารถรองรับการทำงานกับความต้องการประมวลผลด้วยการเรียนรู้ของเครื่อง (Machine Learning: ML) และการเรียนรู้เชิงลึก (Deep Learning: DL) กับการวิเคราะห์ข้อมูลไอโอทีด้วยการเข้ารหัสแบบโฮโมมอร์ฟิก เพื่อรองรับความต้องการใช้งานวิเคราะห์ข้อมูล (data analytics) ไอโอทีในอนาคต

10.2 ข้อเสนอแนะเชิงนโยบายสำหรับกิจการสื่อสารดิจิทัล

ในปัจจุบันยังคงมีความเข้าใจคลาดเคลื่อนเกี่ยวกับการป้องกันความปลอดภัยของข้อมูลทางไซเบอร์และการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับการส่งข้อมูลไปคำนวณบนคลาวด์สาธารณะ เนื่องจากผู้ประกอบการและผู้พัฒนาระบบให้กับผู้ประกอบการโรงงานอุตสาหกรรมเองขาดความรู้ความเข้าใจถึงความแตกต่างของภัยคุกคามทั้งสองรูปแบบ ดังนั้น การสร้างความตระหนักรู้ถึงภัยคุกคามด้านความเป็นส่วนตัวของข้อมูลที่ไม่สามารถถูกกำกับได้เพียงใช้นโยบายเท่านั้น เป็นสิ่งสำคัญที่จะต้องเร่งทำ เพื่อให้ความรู้และความเข้าใจกับผู้ประกอบการไทยให้รู้ทัดเทียมนานาประเทศ

10.3 ข้อเสนอแนะในการใช้งานแพลตฟอร์ม

แพลตฟอร์ม CYBLION สามารถนำไปใช้งานได้กับงานด้านไอโอทีที่ต้องการส่งข้อมูลจากเซนเซอร์ผ่านโพรโทคอล MQTT ที่ต้องการความปลอดภัยสูงและไม่ต้องการให้ข้อมูลรั่วไหลไปยังคลาวด์ และข้อแนะนำการเลือกกุญแจเข้ารหัสที่เหมาะสมกับฟังก์ชันที่ต้องการคำนวณ เช่น หากต้องการคำนวณข้อมูลบนคลาวด์ที่สมการมีการคูณกันไม่เกิน 3 ครั้งและไม่มีการหาร สามารถเลือกเข้ารหัสด้วยกุญแจขนาดเล็กหรือขนาดกลางได้ และถ้าต้องการความถูกต้องระดับทศนิยมไม่เกิน 4 ตำแหน่ง เลือกเข้ารหัสด้วยกุญแจขนาดเล็กจะทำให้ประหยัดทรัพยากรการคำนวณและขนาดข้อมูลเข้ารหัสได้มากกว่าการเข้ารหัสด้วยกุญแจขนาดกลาง แต่หากสมการมีการคูณกันเกิน 3 ครั้งหรือมีการหาร ผู้ใช้งานต้องเลือกเข้ารหัสข้อมูลด้วยกุญแจขนาดใหญ่เท่านั้น และอัตราการส่งข้อมูลจากเอจด์คอมพิวติงไปยังคลาวด์ CYBLION ที่ได้จากการทดสอบ จะไม่ถูกจำกัดจากระบบที่ออกแบบแต่ถูกจำกัดที่ความเร็วในการประมวลของทรัพยากรที่ใช้ในการติดตั้งคลาวด์ เช่น CPU Memory และ Network Interface Card (NIC)

กิตติกรรมประกาศ

โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอที ได้รับการสนับสนุนเงินทุนจากกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ (กทปส.) สัญญาเลขที่ A63-1-(2)-017 ระหว่าง พ.ศ. 2565 ถึง พ.ศ. 2567 ขอขอบคุณบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด และ บริษัท เดอะ เพ็ท จำกัด ที่ให้ข้อมูล สถานที่ และบุคคลากรร่วมทดสอบและทดลองใช้งานแพลตฟอร์ม CYBLION กับระบบไอโอทีของโรงงาน

รายการเอกสารอ้างอิง

Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2017). A survey on homomorphic encryption schemes: theory and implementation. *ACM Computing Surveys, 51*(4), 1-35. https://doi.org/10.1145/321430

Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., & Vaikuntanathan, V. (2019). *Homomorphic encryption standard.* Cryptology ePrint Archive. https://eprint.iacr.org/2019/939

Bossuat, J. -P., Cammarota, R., Chillotti, I., Curtis, B. R., Dai, W., Gong, H., Hales, E., Kim, D., Kumara, B., Lee, C., Lu, X., Maple, C., Pedrouzo-Ulloa, A., Player, R., Polyakov, Y., Ruiz Lopez, L. A., Song, Y., & Yhee, D. (2024). *Security guidelines for implementing homomorphic encryption.* Cryptology ePrint Archive. https://eprint.iacr.org/2024/463

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014, July 1). (Leveled) Fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT), 6*(3), 1-36. http://dx.doi.org/10.1145/2633600

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi & T. Peyrin. (Eds.), *Advances in cryptology – ASIACRYPT 2017.* (pp. 409–437). Springer Nature. https://doi.org/10.1007/978-3-319-70694-8_15

Enveil. (n.d.). *Privacy enhancing technologies (PETs).* https://www.enveil.com/privacy-enhancing-technologies/

Gentry, C. (2009). *A fully homomorphic encryption scheme* [Doctoral dissertation]. Stanford University. https://crypto.stanford.edu/craig/craig-thesis.pdf

Inpher. (n.d.). *TFHE: Fastest, fully homomorphic encryption library.* https://inpher.io/tfhe-library/

Intel. (n.d.). *Intel® Software Guard Extensions.* https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html

Microsoft. (n.d.). *Microsoft SEAL (release 3.6)* [Computer software]. https://www.microsoft.com/en-us/research/project/microsoft-seal/

Mouchet, C. V., Bossuat, J. -P., Troncoso-Pastoriza, J. R., Hubaux, J. -P., Brenner, M., & Lepoint, T. (2020). Lattigo: A multiparty homomorphic encryption library in Go. *8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. WAHC 2020* (pp. 1-6). EPFL. https://infoscience.epfl.ch/entities/publication/79285058-0d3c-464e-a805-e7bf10ed10a0

NETPIE. (2020). *Introducing NETPIE 2020.* https://netpie.io/

Stavish, T. (2017, December 22). *Lab41/PySEAL* [Computer software]. GitHub. https://github.com/Lab41/PySEAL.

TripleBlind. (n.d.). *Empowering your digital world with cutting-edge security and privacy solutions.* https://tripleblind.com/