



กทปส

รายงานฉบับสมบูรณ์

โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนากิจการ
กระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ

โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัว ของข้อมูลสำหรับระบบ IoT

Privacy as a Service for IoT

ดร.กณิกา สุขสมบูรณ์ และคณะ

มกราคม ๒๕๖๗

กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

รายงานฉบับสมบูรณ์

โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัว ของข้อมูลสำหรับระบบ IoT

Privacy as a Service for IoT

คณะนักวิจัย

- | | |
|---------------------------------|------------------|
| ๑. ดร.พนิดา พงษ์ไพบูลย์ | ที่ปรึกษาโครงการ |
| ๒. ดร.กณิกา สุขสมบูรณ์ | หัวหน้าโครงการ |
| ๓. ดร.เอ็มอัสนา นิรันตสุขรัตน์ | นักวิจัยร่วม |
| ๔. ดร.กุลชาติ มีทรัพย์หลาก | นักวิจัยร่วม |
| ๕. ดร.โสภณ มงคลลักษณ์ | ผู้ทรงคุณวุฒิ |
| ๖. น.ส.จิรัฐติกา ททรัพย์สมบูรณ์ | ผู้ประสานงาน |

ได้รับทุนอุดหนุนจาก

กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม
เพื่อประโยชน์สาธารณะ (สำนักงาน กสทช.)

มกราคม ๒๕๖๗

บทสรุปผู้บริหาร
โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT
มกราคม ๒๕๖๗

- | | | | |
|--------------------------|--|--------|----------------|
| ๑. ชื่อองค์กร | ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ | | |
| ๒. ชื่อหัวหน้าโครงการ | นางสาวกสิกา สุขสมบูรณ์ | | |
| ๓. ระยะเวลาดำเนินโครงการ | เริ่มดำเนินโครงการ | วันที่ | ๑๕ มกราคม ๒๕๖๕ |
| | สิ้นสุดโครงการ | วันที่ | ๑๔ มกราคม ๒๕๖๗ |

๔. กิจกรรมที่ดำเนินโครงการ

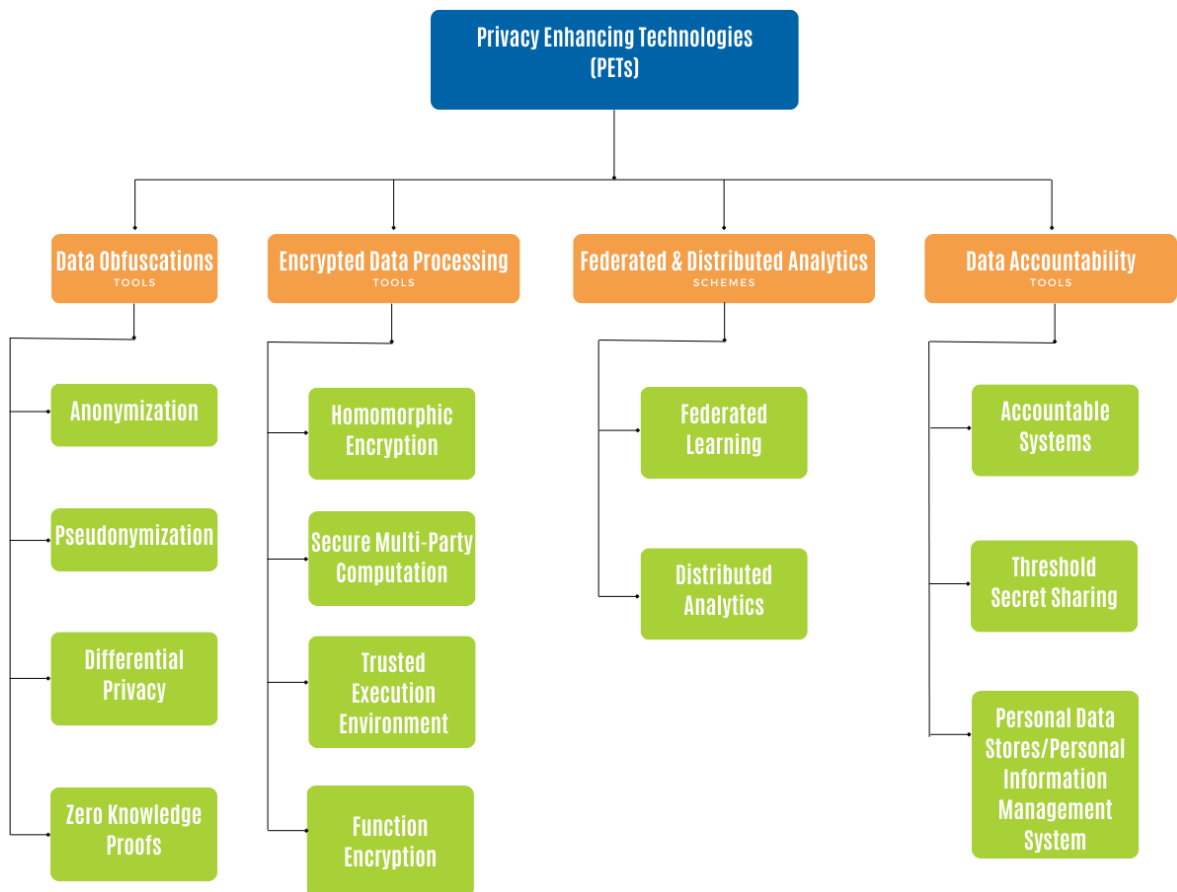
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติได้ดำเนินกิจกรรม เพื่อการส่งมอบผลงานที่มีคุณภาพ โดยมีขั้นตอนดังนี้

๑. ศึกษาความเป็นไปได้ และการพัฒนาระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล
๒. การพัฒนาและออกแบบแอปพลิเคชันสำหรับระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล
๓. การพัฒนาส่วนเชื่อมต่อระหว่างระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลมาผสานรวมเข้ากับ IoT Cloud Platform
๔. กิจกรรมการสร้าง Testbed และทำการทดสอบระบบ
๕. การจัดฝึกอบรมเชิงปฏิบัติการ และการระดมความคิด
 - ๕.๑) การจัดฝึกอบรมเชิงปฏิบัติการ
 - ๕.๒) การประชุมระดมความคิดเห็น
 - ๕.๓) การประชาสัมพันธ์ จัดทำสื่อโปสเตอร์ และ สื่อวิดีโอสำหรับเผยแพร่
๖. ผลงานอื่นๆ ที่ส่งมอบ
 - ๖.๑) ต้นแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที
 - ๖.๒) การติดตั้งและอบรมการใช้งานระบบ IoT Smart Tank Farm ที่โรงงาน บริษัท TVOP
 - ๖.๓) คู่มือการใช้งาน Node-RED และคู่มือการใช้งาน CYBLION
 - ๖.๔) ผลงานตีพิมพ์ในงานประชุมวิชาการ
 - ๖.๕) บทความทางวิชาการระดับนานาชาติ (ฉบับยื่นเสนอบทความ)

โดยมีรายละเอียดการดำเนินกิจกรรมดังต่อไปนี้

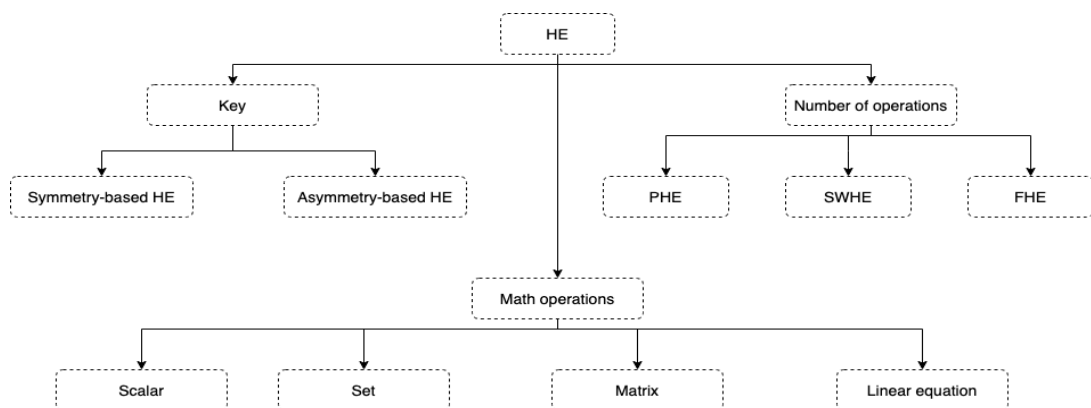
๑) ศึกษาความเป็นไปได้ และการพัฒนาระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติได้ทำการศึกษาวิเคราะห์เทคโนโลยีที่จะเข้ามาช่วยปกป้องคุ้มครองความเป็นส่วนตัวของข้อมูล โดยสามารถสรุปได้ดังรูปแสดงไดอะแกรมภาพรวมเทคโนโลยีสำหรับปกป้องความเป็นส่วนตัวของข้อมูล (Privacy Enhancing Technologies (PETs))



รูปที่ ก.๑ ไดอะแกรมภาพรวมเทคโนโลยีสำหรับปกป้องความเป็นส่วนตัวของข้อมูล
(Privacy Enhancing Technologies (PETs))

ทีมวิจัยได้เลือกเทคนิคการเข้ารหัสแบบโฮโมมอร์ฟิก (Homomorphic Encryption (HE)) ซึ่งเหมาะสมกับการประยุกต์ใช้งานกับข้อมูลอุตสาหกรรมไอโอทีที่ต้องการบริการคำนวณบนคลาวด์ภายนอก สำหรับการเข้ารหัสแบบโฮโมมอร์ฟิกสามารถจำแนกตามจำนวนโอเปอเรชั่นทางคณิตศาสตร์ได้ ๓ ประเภท คือ Partial Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE) และ Fully Homomorphic Encryption (FHE) ตามรูปแสดงแผนผังการจำแนกประเภทของโฮโมมอร์ฟิกส์ (Homomorphic Encryption (HE))



รูปที่ ก.๒ แผนผังการจำแนกประเภทของโฮโมมอร์ฟิกส์ (Homomorphic Encryption (HE))

ทีมวิจัยได้เลือกใช้เทคนิคการเข้ารหัสประเภท FHE ที่อนุญาตให้สามารถทำโอเพอร์เรชันทางคณิตศาสตร์ได้ทั้งการบวกและการคูณ และการสะสมของข้อมูลรบกวนภายหลังการทำโอเพอร์เรชันทางคณิตศาสตร์น้อยกว่าประเภท SWHE เพื่อเลือกใช้เทคนิคที่เหมาะสมกับระบบอุตสาหกรรมไอโอทีที่โครงการนี้จึงได้ทำการศึกษาเปรียบเทียบเทคนิค FHE โดยพิจารณาถึงชนิดของข้อมูลที่สามารถเข้ารหัสได้และวิธีการที่ช่วยลดข้อมูลรบกวนที่เกิดจากการทำโอเพอร์เรชันทางคณิตศาสตร์ดังแสดงใน ตารางเปรียบเทียบวิธีเข้ารหัสแบบ FHE ตามชนิดข้อมูลที่รองรับการเข้ารหัสและวิธีการลด Noise ดังนั้นจะเห็นได้ว่าวิธี CKKS/RNS-CKKS เป็นวิธีที่เหมาะสมกับการนำไปใช้งานกับแอปพลิเคชันอุตสาหกรรมไอโอทีที่การคำนวณไม่ถูกจำกัดอยู่ที่จำนวนเต็มบวกเท่านั้น

ตารางที่ ก. ๑ เปรียบเทียบวิธีเข้ารหัสแบบ FHE ตามชนิดข้อมูลที่รองรับการเข้ารหัสและวิธีการลด Noise

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

จากนั้นทางศูนย์ฯ ได้พิจารณาชุด Compiler และ ไลบรารีสำหรับ FHE แบบ CKKS/RNS-CKKS ที่สามารถนำมาพัฒนาต่อยอดได้ง่ายและสมบูรณ์ ในตารางแสดง Compiler และ ไลบรารีสำหรับเทคนิคการเข้ารหัสแบบ HE พบว่าในช่วงเวลาที่ทีมวิจัยได้ทำการศึกษาเพื่อค้นหา Compiler และ ไลบรารีสำหรับ FHE ที่รองรับเทคนิคการเข้ารหัสแบบ CKKS/RNS-CKKS นั้น Microsoft SEAL เป็นไลบรารีที่ถูกพัฒนาด้วยภาษา C++ และมีโอเพอเรชันให้ใช้งานค่อนข้างครบถ้วนกว่าไลบรารีอื่นๆ ทางโครงการนี้จึงเลือกพัฒนาต่อยอดจาก Microsoft SEAL

ตารางที่ ก. ๒ Compiler และ ไลบรารีสำหรับเทคนิคการเข้ารหัสแบบ HE

FHE Library	อ้างอิง	เทคนิคที่ใช้	โปรแกรมภาษาที่ใช้
Microsoft SEAL	[54]	SWHE, BFV, CKKS, RNS-CKKS	C++
HElib	[28]	CKKS, BFV	C++
PALISADE	[48]	BGV, BFV, CKKS, FHEW,	Cross-platform C++
Pyfhel	[49]	Afhe1, SEAL, PALISADE	Python
PySEAL	[50]	SWHE, BFV, SEAL	Python
PySyft	[72]	SWHE	Python
FV-NFLlib,	[26]	SWHE, FV	C++
HEAT	[27]	HE API	C++
latticego	[73]	BFV, CKKS, RNS-CKKS	Go
HEAAN	[74]	CKKS, RNS-CKKS	C++
concrete	[75]	GSW	Rust

๒) การพัฒนาและออกแบบแอปพลิเคชันสำหรับระบบบริการคุ้มครองความเป็นส่วนตัวของข้อมูล

ทีมวิจัยให้ความสำคัญกับการพัฒนาแพลตฟอร์มบริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอทีที่ยั่งยืน และสอดคล้องกับระบบนิเวศน์ไอโอที ที่เก็บข้อมูลจากเซนเซอร์ต่าง ๆ และเข้ารหัสบน Edge Computing เพื่อส่งข้อมูลที่เข้ารหัสแล้วไปคำนวณบนคลาวด์ที่รองรับสภาพแวดล้อมการคำนวณแบบโฮโมมอร์ฟิกส์ และข้อมูลจะถูกเรียกใช้โดยเครื่องรับผลที่ผู้ใช้งานโดยแสดงข้อมูลที่ถูกลอกรหัสแล้ว เพื่อให้การใช้งานที่ง่ายและเหมาะสมกับการใช้งาน สำหรับโรงงานอุตสาหกรรม (System Integrators (SI)) และวิศวกรผู้ดูแลและพัฒนาระบบไอโอทีในโรงงานอุตสาหกรรม ที่ไม่จำเป็นต้องมีความรู้และความเชี่ยวชาญด้านวิทยาการเข้ารหัสข้อมูล (Cryptographic) การพัฒนาแพลตฟอร์มนี้อยู่ภายใต้เทคโนโลยีการเข้ารหัสแบบโฮโมมอร์ฟิกส์แบบ RNS-CKKS ที่สามารถเข้ารหัสข้อมูลบนจำนวนจริงได้ ซึ่งแตกต่างจากการเข้ารหัสโฮโมมอร์ฟิกส์วิธีอื่นๆ ที่ใช้ได้เฉพาะข้อมูลเลขจำนวนเต็มหรือค่าบูลีนเท่านั้น โดยแพลตฟอร์มถูกพัฒนาอยู่บนพื้นฐานของไลบรารีของ Microsoft SEAL และการใช้งานเครื่องมือการเข้ารหัสและการคำนวณทางคณิตศาสตร์บน Node-RED ซึ่งเป็นโปรแกรมสำหรับนักพัฒนาไอโอทีแบบ Low-code ที่ไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโปรแกรมคอมพิวเตอร์ก็สามารถพัฒนาต่อยอดสำหรับแอปพลิเคชันไอโอทีได้ง่าย

แพลตฟอร์มที่พัฒนาขึ้น ภายใต้ชื่อแบรนด์ CYBLION ที่มาจากคำว่า CYBER และ BRILLIANT ซึ่งอ่านว่า “ไซ-บี-เลียน” จึงเป็นที่มาของโลโก้สัญลักษณ์ฮัสกี ไซบีเลียน โดยที่แพลตฟอร์ม CYBLION ประกอบด้วย ๓ ส่วน หลัก

- ๑) Edge Computing ที่มีส่วนของ Code สำหรับการเข้ารหัส บนข้อมูลและส่งข้อมูลออกไปยังคลาวด์ผ่านโพรโทคอล MQTT ที่รองรับข้อมูลชนิดโฮโมมอร์ฟิกส์ ซึ่งผู้ใช้งานสามารถดาวน์โหลดได้จาก Node-RED Pallet โดยเลือกไปที่ Node-RED-cyblion-edge เพื่อมาติดตั้งในโปรแกรม Node-RED ภายในอุปกรณ์ Edge Computing ที่เชื่อมต่อกับเซนเซอร์
- ๒) Cloud Computing (CYBLION.io) ที่พัฒนาในรูปแบบของ Web Application ที่ผู้ใช้งานสามารถเข้าถึงได้ผ่าน Web Browser โดยผู้ใช้งานจะสามารถกำหนดและสร้างสูตรคำนวณได้เองบนเครื่องมือการคำนวณแบบโฮโมมอร์ฟิกส์ โดยสภาพแวดล้อมการคำนวณจะมีความเป็นส่วนตัวสำหรับผู้ใช้งานแต่ละบัญชีผู้ใช้งาน (Account) และเฉพาะเจาะจงสำหรับแต่ละโปรเจกต์ (Project) ที่ใช้เฉพาะงานไอโอทีแต่ละกรณี ทำให้การจัดการสร้างหรือลบโปรเจกต์ทำได้ง่ายและสะดวก
- ๓) Mobile Application (CYBLION App) ถูกพัฒนาให้ใช้งานได้ทั้งบนสมาร์ตโฟนระบบปฏิบัติการ iOS และ Android โดยตัว CYBLION App ทำให้ผู้ใช้งานสามารถควบคุมความเป็นส่วนตัวของข้อมูลไอโอทีของตนเองได้ง่าย ซึ่งผู้ใช้งานสามารถสร้าง และลบ โปรเจกต์ได้ โดยภายในโปรเจกต์ผู้ใช้สามารถสร้าง Edge ซึ่งเป็นอิมเมจเสมือนของ Edge ที่ติดตั้งในโลกกายภาพ โดยภายใน Edge จะใช้ในการจัดการด้านการเข้ารหัสข้อมูลและการส่งข้อมูล และผู้ใช้งานสามารถสร้างชุดกฎแตรหัสลับสำหรับแต่ละโปรเจกต์ได้ โดยการเลือกขนาดของกฎแตรหัสลับให้เหมาะสมกับความต้องการใช้งานการคำนวณผ่าน CYBLION App ทำให้การกำหนดค่าพารามิเตอร์ที่เหมาะสมสำหรับการเข้ารหัสโฮโมมอร์ฟิกส์เป็นเรื่องง่ายสำหรับผู้ใช้งานทั่วไป โดยเลือกชุดพารามิเตอร์ผ่านกฎแตรหัสลับที่มีขนาด เล็ก กลาง และใหญ่ (S, M และ L) ซึ่งขนาดของกฎแตรหัสลับจะสอดคล้องกับความซับซ้อนของการคำนวณและความถูกต้องของผลลัพธ์ภายหลังการถอดรหัส นอกจากนี้ ผู้ใช้งาน CYBLION App สามารถลักษณะการแสดงผลที่ที่ต้องการ ทั้งแบบรูปภาพ และข้อมูลที่ต้องการให้แสดง

๓) การพัฒนาส่วนเชื่อมต่อระหว่างระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลมาผสานรวมเข้ากับ IoT Cloud Platform

หัวใจหลักของการผสานรวมส่วนต่างๆ ของแพลตฟอร์ม CYBLION ให้ทำงานร่วมกันได้อย่างมีประสิทธิภาพ และการเชื่อมโยงเชื่อมต่อกับ NETPIE นั่นคือ ส่วน API ซึ่งทางศูนย์ฯ ได้พัฒนาส่วนเชื่อมต่อทั้งสามองค์ประกอบนี้ขึ้น เรียกว่า Mobile Gateway ที่ทำหน้าที่เชื่อมต่อระหว่าง ๑) Edge Computing ๒) Cloud Computing 3) Mobile Application และ NETPIE โดยระบบการยืนยันตัวตนของ แพลตฟอร์ม CYBLION อาศัยการ Authentication กับ ระบบ Authentication ของ NETPIE เพื่อให้ NETPIE ทำการตรวจสอบสิทธิ์ในการใช้งานและโควต้าของแต่ละ Account ดังนั้นผู้ที่ต้องการใช้งานแพลตฟอร์ม CYBLION สามารถทำได้โดยการสร้างบัญชีผู้ใช้งานบน NETPIE.io และส่งคำร้องขอลงทะเบียนการเข้าถึงมาทางผู้พัฒนา CYBLION เพื่อเพิ่มชื่อบัญชีผู้ใช้งานใน Whitelist โดยผู้พัฒนาเปิดให้ใช้งานฟรีด้วยโควต้าจำนวน ๑ ชื่อบัญชีต่อ ๓ โปรเจ็คและ ๑.๐ Edge เป็นระยะเวลา ๓ เดือน

MQTT broker เป็นส่วนสำคัญของระบบการสื่อสารของอุปกรณ์ IoT เพื่อให้ข้อมูลจากอุปกรณ์ IoT สามารถสื่อสารระหว่าง Cloud Computing และ Mobile Application ได้ในรูปแบบชุดข้อมูลที่เข้ารหัสแบบไฮโมเมอร์ฟิสมันจำเป็นต้องมีการออกแบบ MQTT broker ใหม่ที่มีประสิทธิภาพเพียงพอในการรองรับข้อมูลขนาดใหญ่ได้ เนื่องจากโดยปกติข้อมูลไอโอทีที่ส่งผ่าน MQTT จะมีขนาดเล็ก ดังนั้นในโครงการนี้ได้พัฒนา FHE.MQTT broker ขึ้นใหม่ที่แยกจาก MQTT broker ของ NETPIE

๔) กิจกรรมการสร้าง Testbed และทำการทดสอบระบบ

ทีมพัฒนาให้ความสำคัญกับการทดสอบใช้งานเครื่องมือเข้ารหัส และเครื่องมือคำนวณต่างๆ บนโปรแกรม Node-RED จึงพัฒนาชุดเครื่องมือ Node-RED-cipherflow ที่เปิดบริการฟรีดาวน์โหลดติดตั้งลงบนโปรแกรม Node-RED เพื่อการเรียนรู้ โดยชุดเครื่องมือนี้ผู้ใช้งานสามารถติดตั้งและเรียนรู้การใช้งานได้ด้วยตนเอง โดยไม่ต้องมีบัญชีการใช้งานแต่อย่างใด เครื่องมือ Node-RED-cipherflow นี้จะช่วยให้ผู้ใช้งานเข้าใจกระบวนการทำงาน การคำนวณข้อมูลแบบเข้ารหัสข้อมูลและทดสอบสูตรคำนวณและผลลัพธ์ของการคำนวณที่ต้องการตลอดจนการเลือกขนาดของกฎเกณฑ์ที่เหมาะสมกับความต้องการใช้งานก่อนเริ่มไปใช้งานจริงบนแพลตฟอร์ม CYBLION

การทดสอบแพลตฟอร์ม CYBLION ที่ได้ถูกพัฒนาเป็น CYBLION Application ให้ใช้งานได้ทั้งระบบ Android และ iOS โดยสร้างข้อมูลบน Node-RED และส่งข้อมูลไปคำนวณบนคลาวด์ และเรียกตรวจสอบผลบนสมาร์ทโฟนที่ติดตั้ง CYBLION Application จากการทดสอบในห้องปฏิบัติการ โดยการทดสอบกับ Edge computing ด้วยอุปกรณ์ขนาดเล็กต่างๆ ได้แก่ Raspberry Pi 3 B+, Raspberry Pi 4, Jetson Nano และ IoT2050 ที่ติดตั้ง Node-RED-Cyblion-Edge และได้ทำการเชื่อมต่อ Sensor Radar จริงพร้อมทดสอบในห้องปฏิบัติการเพื่อเปรียบเทียบค่าที่ได้จากการแสดงผลผ่าน NETPIE Application เพื่อทดสอบทั้งความถูกต้อง และประสิทธิภาพของการส่งข้อมูลทั้งระบบ

๕) การจัดฝึกอบรมเชิงปฏิบัติการ การจัดประชุมระดมความคิด และการประชาสัมพันธ์เผยแพร่ผลงาน

เพื่อถ่ายทอดความรู้ พัฒนาและ ฝึกปฏิบัติการใช้แพลตฟอร์ม การให้บริการคุ้มครองความเป็นส่วนตัว ของ ข้อมูลไอโอที ที่พัฒนาขึ้น จากการดำเนินโครงการ และได้รับผลการจัดอบรมเชิงปฏิบัติการดังนี้

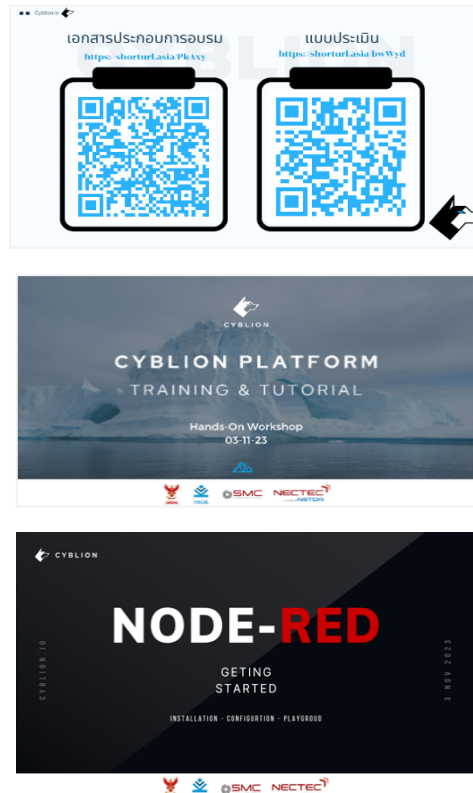
๕.๑) การจัดอบรมเชิงปฏิบัติการ

หัวข้อ “Privacy as a Service For IoT/IoT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IoT ด้วย CYBLION Platform” วันศุกร์ที่ ๓ พฤศจิกายน ๒๕๖๖ เวลา ๐๙.๐๐ - ๑๖.๐๐ น. ณ ห้องประชุม Auditorium (CO-๑๑๓) อาคารสำนักงานกลาง อุทยานวิทยาศาสตร์ประเทศไทย

กลุ่มเป้าหมาย ผู้แทนจาก ภาครัฐกิจ/โรงงานอุตสาหกรรม ได้แก่ ผู้ใช้งานในภาคอุตสาหกรรมและผู้ใช้งานระดับนักพัฒนาในภาคอุตสาหกรรม ผู้ให้บริการ IoT Cloud Platform นักพัฒนาและผู้ผลิตอุปกรณ์ IoT มีหน่วยงานที่สนใจเข้ารับการอบรม ๑๒ หน่วยงาน และมีจำนวนผู้เข้ารับการอบรมจำนวน ๕๖ คน



รูปที่ ก. ๓ บรรยากาศการประชุมเชิงปฏิบัติการและระดมความคิดเห็น



รูปที่ ก. ๔ เอกสารที่ใช้ประกอบการประชุมปฏิบัติการและระดมความคิดเห็น

๕.๒) การจัดประชุมระดมความคิดเห็น

ส่วนของการสำรวจความตระหนักรู้และระดมความคิดเห็นเกี่ยวกับการติดตั้งและใช้งานระบบ ในการประชุมเชิงปฏิบัติการ สามารถสรุปได้ดังนี้

- ด้านการใช้งานแอปพลิเคชันบนสมาร์ตโฟน

ผู้เข้าอบรมมีความต้องการสร้างและคำนวณข้อมูลด้วยสมาร์ตโฟนหลายเครื่อง เพื่อประเมินผลบนคลาวด์ต่อไป แต่การคำนวณข้อมูลในหัวข้อนั้นต้องเป็นข้อมูลชนิดเดียวกัน รวมถึงมีความต้องการให้ผู้บริหารหรือบุคคลอื่นในองค์กรสามารถดูข้อมูลที่คำนวณได้ และต้องการส่งออกข้อมูล ที่รวบรวมมาไปเก็บไว้ที่อื่น ซึ่งพีเจอร์ในปัจจุบันยังไม่ได้รองรับการดูข้อมูลได้หลายเครื่องหรือส่งออกข้อมูลไปไว้ที่อื่น และเป็นโจทย์ที่น่าสนใจในการพัฒนาเป็น Multi-Key ต่อไป

- ด้านอุปกรณ์ที่รองรับการใช้งาน

สำหรับ CYBLION Platform ปัจจุบันยังคงต้องทำบนแอปพลิเคชันบนสมาร์ตโฟน และมีข้อจำกัดที่ไม่สามารถ Export Key จากสมาร์ตโฟนเครื่องหนึ่งไปอีกเครื่องหนึ่ง หรือไม่สามารถแชร์ Secret Key ข้ามเครื่องได้ ทั้งนี้ หากเปลี่ยนเครื่องใช้งาน จำเป็นต้องสร้างกุญแจใหม่ (Key Generation) ใหม่

- ด้านการใช้ Node-Red และการจัดเตรียมข้อมูล

สำหรับการใช้งานแอปพลิเคชันต้องมีความรู้พื้นฐานในการใช้ Node-Red เนื่องจาก CYBLION platform พัฒนาอยู่บนพื้นฐานของการใช้งาน Node-RED

- ด้านการใช้ประยุกต์ใช้งานกับ Use Case

ผู้เข้าร่วมอบรมมีความสนใจนำ CYBLION platform ไปใช้งานกับฐานข้อมูลที่นอกเหนือจากเซนเซอร์ และมีความสนใจเกี่ยวกับการวัดประสิทธิภาพของกระบวนการทำงาน (Process) โดยการใช้การทยอยส่งค่าของข้อมูล ซึ่งตามคุณสมบัติพื้นฐานของแอปพลิเคชันสามารถทำได้ หากใช้ Node-RED และจัดเตรียมข้อมูลให้มีรูปแบบให้สามารถเข้ากับแอปพลิเคชัน

นอกจากนี้สิทธิประโยชน์ของผู้ร่วมอบรมที่เป็นสมาชิก SMC และ IDA Platform ในการใช้งานแอปพลิเคชัน ก็เป็นสิ่งที่ควรนำมาพิจารณาแผนงานการให้บริการในอนาคต เนื่องจากมีผู้เข้าร่วมอบรมที่เป็นสมาชิก IDA มีความสนใจที่จะใช้งานแอปพลิเคชันนี้

๕.๓) การประชาสัมพันธ์ จัดทำสื่อโปสเตอร์ และ สื่อวีดิทัศน์สำหรับเผยแพร่

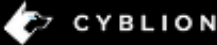
โครงการนี้ได้จัดให้มีการประชาสัมพันธ์ จัดทำสื่อวีดิทัศน์สำหรับเผยแพร่ผลงาน และการให้ความรู้สำหรับการใช้งานแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ดังนี้

๕.๓.๑) การออกบูธแสดงนิทรรศการและให้ความรู้ประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการ AINTEC 2023 เมื่อวันที่ ๑๒ - ๑๔ ธันวาคม ๒๕๖๖ เมืองฮานอย ประเทศเวียดนาม ดังรูปแสดงการนิทรรศการประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการ AINTEC 2023 โดยมีผู้เข้าร่วมงานจากนานาชาติ เช่น ประเทศไทย เวียดนาม ญี่ปุ่น ฟิlippินส์ มาเลเซีย สิงคโปร์ ไต้หวัน จีน สหรัฐอเมริกา อังกฤษ ฝรั่งเศส ออสเตรเลีย และ นิวซีแลนด์ รวมไม่น้อยกว่า ๕๒ ท่าน



รูปที่ ก.๕ นิทรรศการประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการนานาชาติ AINTEC 2023

๕.๓.๒) จัดทำโปสเตอร์ประชาสัมพันธ์แนะนำแพลตฟอร์มขึ้นโดยได้นำไปเผยแพร่ผลงาน ตามรูปแบบโปสเตอร์ประชาสัมพันธ์การประชุมเชิงปฏิบัติการ



Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0

Authors
 Kalika Suksomboon⁽¹⁾, Nataset Tanabodee⁽²⁾, Eakarut Saktaewomlerd⁽³⁾,
 Almaschana Niruntasakrat⁽¹⁾, Chaveee Issariyapat⁽⁴⁾, Sophon Mongkoluksamee⁽⁴⁾,
 Natapon Tansangworn⁽²⁾, Sukumal Kitisin⁽²⁾
 (1) NECTEC, (2) Kasetsart University, (3) Srinakharinwirot University, (4) NEXOME .co. Ltd

Introduction

“A recent cloud data leak made the factory owners rethink whether the move to Industry 4.0 might just be a trap.”

“Thus, there is a need to answer this one quick question: *How to protect industrial individual data privacy while letting factories enjoy Industry 4.0 benefits?*”

Goal

- To develop an IoT platform that allows IIoT/IoT developers to preserve and control data privacy via a mobile app while outsourcing computation to the cloud.

Key Technologies

- Homomorphic encryption (HE) has been implemented for data privacy protection, preventing the third-party cloud from learning anything during computation outsourcing.
- Node-RED is a low-code platform for accelerating the development of IIoT applications.
- CipherFlow is a HE toolkit for the RNS-CKKS-based HE scheme on Node-RED.

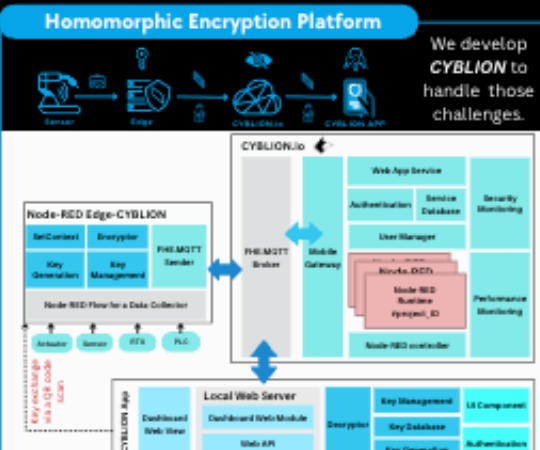
Challenges

CipherFlow proved to be a capable tool to protect IIoT data privacy based on HE; however, it initially was not designed to handle the following issues which required further extension:

- multiple users sharing one server
- transmitting HE data over MQTT protocol
- real-time data visualization
- automatic parametrization for non-cryptographic specialists
- privacy control via a mobile app

Homomorphic Encryption Platform

We develop **CYBLION** to handle those challenges.



Demonstration and Discussion

IIoT Use Case: Privacy-Preserving a Smart Tank Farm



Discussion

- Creating a Node-RED runtime project on the cloud takes 2-10 sec after a user clicks to create a project on the mobile app.
- The dashboard takes 3 sec to start displaying the decrypted data after the sensor transmits data to be encrypted at the edge, and the edge to the cloud.

Parameter	Small	Medium	Large
Creation Time (sec)	2.5	3.5	4.5
Decryption Time (sec)	3.0	3.5	4.0
Storage (KB)	100	200	300
Throughput (KB/s)	1.0	2.0	3.0

References
 [1] Jung Hye Chon, Andrey Kim, Min-Gu, and Dongsoo Song, "Homomorphic Encryption for Addition of Approximate Numbers" in ASIS/DBPPT 2015.
 [2] Nakul Tansubee and et al., "CipherFlow: A Framework for Developing Privacy Preserving IIoT on Node-RED" in MITSOT 23, 18-19.

Acknowledgement
 This work was funded and financially supported by NBTC, Thailand, under the grant A63-3-(2)-037.

Download CYBLION App

Download on the App Store

GET IT ON Google Play





รูปที่ ก. ๖ โปสเตอร์ประชาสัมพันธ์งานประชุมวิชาการนานาชาติ AINTEC 2023

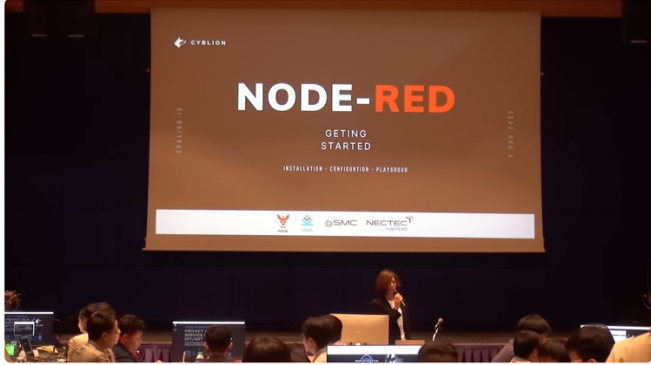
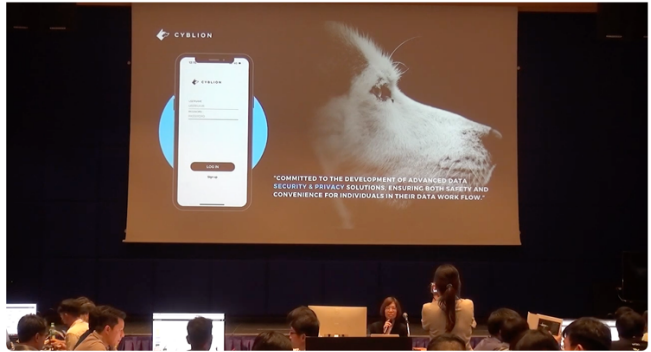


รูปที่ ก. ๗ โปสเตอร์ประชาสัมพันธ์การประชุมเชิงปฏิบัติการ

๕.๓.๓) การจัดทำสื่อวีดิทัศน์สำหรับเผยแพร่ข้อมูลการอบรม และการให้ความรู้สำหรับการใช้งานแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ตามตารางแสดงรายการสื่อวีดิทัศน์ สำหรับประชาสัมพันธ์ เผยแพร่และจัดอบรม

ตารางที่ ก. ๓ รายการสื่อวีดิทัศน์สำหรับประชาสัมพันธ์ เผยแพร่และจัดอบรม

หัวข้อบรรยาย	ลิงค์เผยแพร่	จำนวนผู้เข้าชม
Privacy-as-a-Service for IoT	https://youtu.be/a19Xd5FPTVQ 	145 ครั้ง
Node-RED และ	https://youtu.be/M0kWEfsh6II	368 ครั้ง

CipherFlow	 [20231103] การใช้งาน โปรแกรม Node-RED NECTEC 10.9K subscribers 348 views · 2 weeks ago	
CYBLION	https://youtu.be/CCbvJuQCfC0  [20231103] แนะนำวิธีการใช้งาน CYBLION Platform NECTEC 10.9K subscribers 135 views · 2 weeks ago	136 ครั้ง
PET as a Service for IoT	https://www.facebook.com/csitnu/videos/567955892168564/?_rdc=1&_rdr  WHAT IS CYBLION? JCSSE 2023 ช่วงที่ 2 CSIT มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี 20 June 2023 · 20 views	2,400 ครั้ง

๖) ผลงานอื่นๆ ที่ได้รับ

๖.๑) ต้นแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที

ผลงานวิจัยและพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ที่สามารถประยุกต์ใช้งานสำหรับโรงงานอุตสาหกรรมและสถานประกอบการ เพื่อให้หน่วยงานที่สนใจ ทั้งผู้ให้บริการวางระบบ โรงงาน สถานประกอบการสามารถนำไปศึกษา และพัฒนาต่อยอดเพื่อใช้งานหรือให้บริการ โดยผู้ใช้งาน

ทั่วไป สามารถดาวน์โหลดแพลตฟอร์ม ที่เผยแพร่มาใช้งานได้ และไม่เก็บค่าใช้จ่ายในการนำไปประยุกต์ใช้งาน โดยมีข้อมูลเผยแพร่ตามตารางแสดงแอปพลิเคชันที่ได้รับจากการดำเนินโครงการ

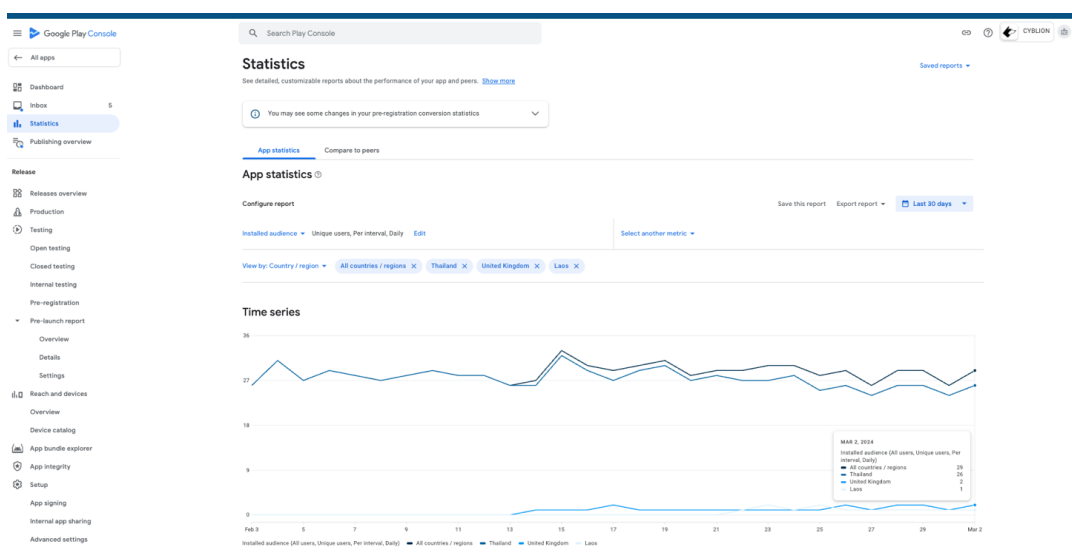
ตารางที่ ก. ๔ แอปพลิเคชันที่ได้รับจากการดำเนินโครงการ

ลำดับ	ผลงาน	GitHub/Publish
๑.	Node-RED-CipherFlow	https://flows.nodered.org/node/node-red-cipherflow
๒.	node-red-cyblion-edge	https://flows.nodered.org/node/node-red-cyblion-edge/scorecard
๓.	CYBLION App (Android)	https://play.google.com/store/apps/details?id=cps.nectec.seal.mobile&hl=en-TH
๔.	CYBLION App (iOS)	https://apps.apple.com/th/app/cyblion/id6446020335
๕.	CYBLION.io (Cloud)	https://cyblion.io/

ทีมผู้วิจัยและพัฒนา กสิกา สุขสมบุญ, เอมอัสนา นรินทร์สุวรรณ์, สุรพันธ์ ทองรังสี, ตะวัน ห่อหุ้ม, “แพลตฟอร์มไอโอทีคลาวด์ที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัส,” ต้นแบบของผลิตภัณฑ์หรือกระบวนการขั้นสุดท้าย ได้ผ่านการสาธิตและพิสูจน์การใช้งานในสภาวะทำงานจริง

๖.๒) สถิติการเข้าเยี่ยมชมและติดตั้งแอปพลิเคชัน CYBLION

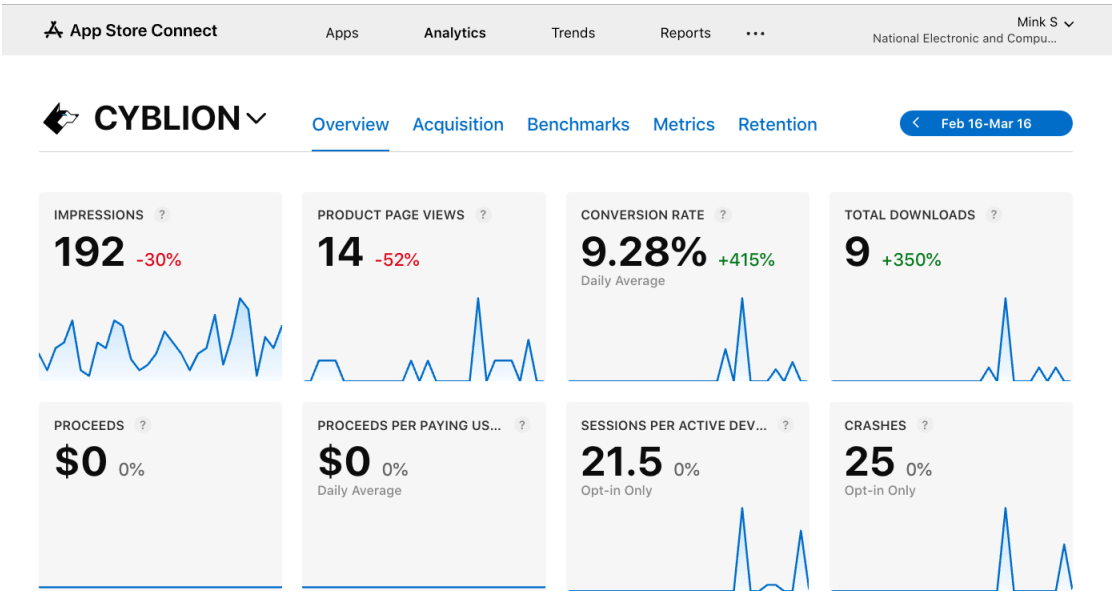
รูปที่ ก. ๘ ถึง ๑๒ แสดงสถิติการเข้าเยี่ยมชมและติดตั้งแอปพลิเคชัน CYBLION บน Google Playstore และ Apple Store จากสถิติพบว่า บน Google Playstore มีการดาวน์โหลดโดยประเทศไทย ๒๖ ครั้ง ประเทศอังกฤษ ๒ ครั้ง และ บน Apple Store มีการดาวน์โหลดโดยประเทศในทวีปเอเชีย ๓๘ ครั้ง ประเทศอังกฤษ ๔ ครั้ง และมีผู้เข้าชมทั้งหมดโดยประเทศในทวีปเอเชีย ๘๓๖ ครั้ง ประเทศอังกฤษ ๗๑ ครั้ง



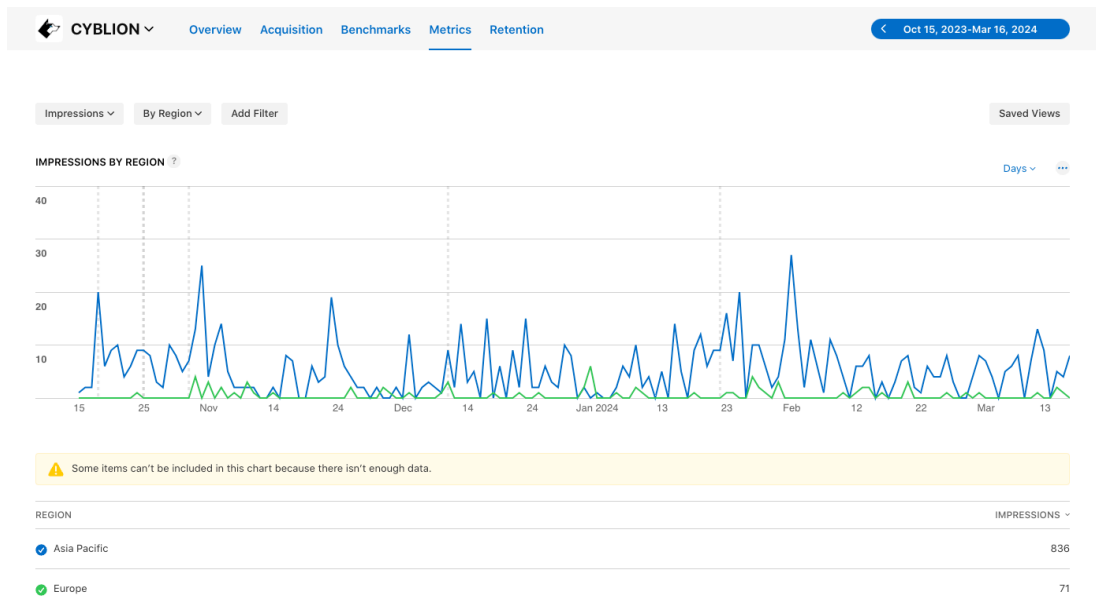
รูปที่ ก. ๘ สถิติการดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ Android ณ วันที่ 2 มีนาคม พ.ศ. ๒๕๖๗ ดาวน์โหลดโดยประเทศไทย 26 ครั้ง ประเทศอังกฤษ 2 ครั้ง และประเทศลาว 1 ครั้ง

Days	All countries / regions	Thailand	United Kingdom	Laos
Mar 2, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Mar 1, 2024	26	24	1	1
Percentage of total	100%	92.31%	3.85%	3.85%
Feb 29, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 28, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 27, 2024	26	24	1	1
Percentage of total	100%	92.31%	3.85%	3.85%
Feb 26, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 25, 2024	28	25	1	2
Percentage of total	100%	89.29%	3.57%	7.14%
Feb 24, 2024	30	28	1	1
Percentage of total	100%	93.33%	3.33%	3.33%
Feb 23, 2024	30	27	1	2
Percentage of total	100%	90%	3.33%	6.67%
Feb 22, 2024	29	27	1	1
Percentage of total	100%	93.1%	3.45%	3.45%

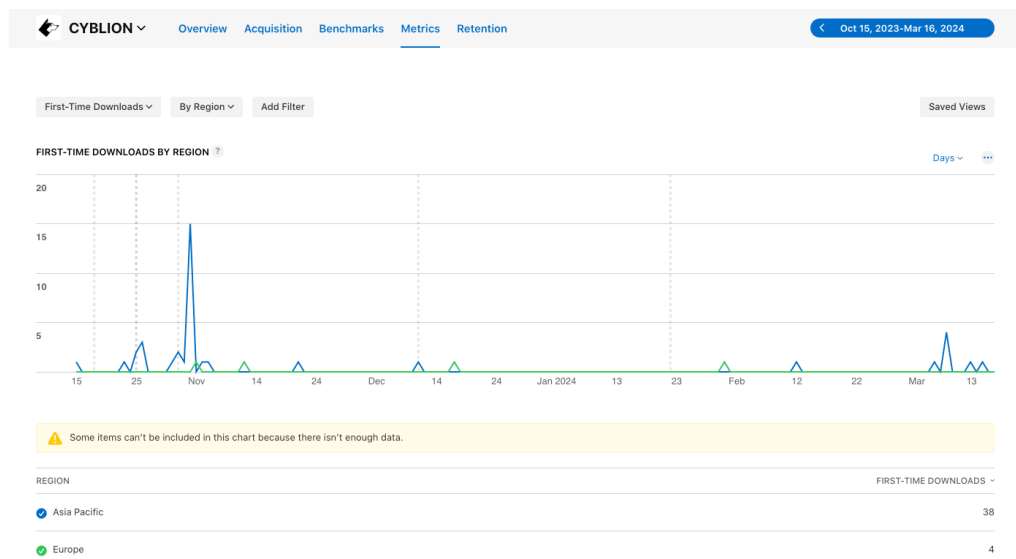
รูปที่ ก. ๙ สถิติการดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ Android จำแนกตามประเทศ



รูปที่ ก. ๑๐ ภาพรวมสถิติการเข้าเยี่ยมชมและดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ iOS



รูปที่ ก. ๑๑ สถิติการเข้าเยี่ยมชม CYBLION Application บนระบบปฏิบัติการ iOS



รูปที่ ก. ๑๒ สถิติการเข้าดาวน์โหลดและติดตั้ง CYBLION Application บนระบบปฏิบัติการ iOS

๖.๓) สถิติการใช้งาน CYBLION.io

๑๓ แสดงสถิติผู้ใช้งานบน CYBLION.io ณ วันที่ ๑๙ มีนาคม ๒๕๖๗ โดยระยะเวลาที่ใช้งานในระบบสูงสุด ๒๗๑ วัน และมีผู้ใช้งานใหม่ระยะเวลา ๓ วัน ซึ่งผู้ใช้งานในระบบเป็นแบบจำกัดจำนวนเนื่องจากขนาดทรัพยากรของคลาวด์ที่ให้บริการมีจำนวนจำกัด


```

root@cipherflow:/home/kalika# kubectl get pods --all-namespaces |grep "cyblion"
cyblion      database-gateway-684cbb6b87-ztm89      1/1      Running      3 (7d7h ago)      136d
cyblion      database-gateway-684cbb6b87-kn6mb      1/1      Running      3 (7d7h ago)      136d
cyblion      groupstore-redis-849958d995-4sj4m      1/1      Running      14 (7d7h ago)      378d
cyblion      nodered-rabbitmq-0                      1/1      Running      14 (7d7h ago)      378d
cyblion      cyblion-session-redis-0                1/1      Running      14 (7d7h ago)      378d
cyblion      edgebroker-redis-5588c78444-16x5n      1/1      Running      14 (7d7h ago)      378d
cyblion      database-gateway-684cbb6b87-xskcx      1/1      Running      3 (7d7h ago)      136d
cyblion      mysql-59fb556cb8-t6hcl                1/1      Running      14 (7d7h ago)      378d
cyblion      groupstore-6bff5bbd7b-18g52            1/1      Running      14 (7d7h ago)      378d
cyblion      nodered-controller-8b8666d6b-q24nv     1/1      Running      5 (7d7h ago)      170d
cyblion      chatchamon-p743062619213-nodered-6fc56d4f5f-tb5hn 1/1      Running      16 (7d7h ago)      167d
cyblion      eakarat-p477894720506-nodered-7f5cb54cf7-twrk     1/1      Running      2 (7d7h ago)      53d
cyblion      cybliontvp-p945763475236-nodered-5566f8966c-klz9g 1/1      Running      28 (7d7h ago)      217d
cyblion      plume142-p946106028223-nodered-656cf8cbbb-bcssm   1/1      Running      14 (7d7h ago)      140d
cyblion      cyblionpentester2-p710866874810-nodered-f6bd88d99-k8rr2 1/1      Running      10 (7d7h ago)      139d
cyblion      eakarat-p151748976058-nodered-7c6457b775-45fwr    1/1      Running      10 (7d7h ago)      136d
cyblion      web-server-849fd4b876-914hp            1/1      Running      21 (7d7h ago)      198d
cyblion      cyblionwsk1-p539072358151-nodered-57f776f5fd-77b2k 1/1      Running      11 (7d7h ago)      136d
cyblion      plume142-p568630801653-nodered-58d4597dbd-d8hbhj 1/1      Running      15 (7d7h ago)      140d
cyblion      cybliontester4-p948727399714-nodered-798cc7845d-v12qs 1/1      Running      13 (7d7h ago)      145d
cyblion      cyblion-p102818583568-nodered-5f75677f49-8sl9d    1/1      Running      30 (7d7h ago)      228d
cyblion      cybliontester4-p986965858910-nodered-7fd99fcb57-nmnj5 1/1      Running      27 (7d7h ago)      228d
cyblion      cyblionpentester-p050289124243-nodered-7c89f7b87c-2q6hj 1/1      Running      13 (7d7h ago)      145d
cyblion      edgebroker-0                            3/3      Running      27 (7d7h ago)      223d
cyblion      cybliontester-p210508897403-nodered-5c58f969bc-gb6w9 1/1      Running      10 (7d7h ago)      136d
cyblion      cyblion-p238808386764-nodered-5f58f9f7bd-nd462    1/1      Running      2 (7d7h ago)      97d
cyblion      cybliontester4-p587675092544-nodered-868584c666-npgp7 1/1      Running      15 (7d7h ago)      144d
cyblion      cyblionwsk1-p327794429068-nodered-5c584c7fb8-kztxn 1/1      Running      10 (7d7h ago)      136d
cyblion      cpssecteam-p968779617716-nodered-65d89dc7f8-1h7jt 1/1      Running      31 (7d7h ago)      228d
cyblion      cyblionmaster-p867946906445-nodered-5576f89767-qcpbt 1/1      Running      27 (7d7h ago)      227d
cyblion      iot-rabbitmq-0                          1/1      Running      14 (7d7h ago)      378d
cyblion      mobile-gateway-6c99b6b56d-xmw86        1/1      Running      0                6d12h
cyblion      mobile-gateway-6c99b6b56d-tdgr4        1/1      Running      0                6d12h
cyblion      mobile-gateway-6c99b6b56d-2nqgh        1/1      Running      0                6d12h
cyblion      i-p264514939583-nodered-bf59c889c-8szl2 1/1      Running      0                5d18h
cyblion      cyblionmaster-p194726227525-nodered-7ccdd99b85-mcxhp 1/1      Running      0                3d7h
cyblion      cyblion-wsk4-p644955397500-nodered-77bcf5c88d-g9pb5 1/1      Running      0                3d7h
cyblion      cyblionwsk6-p323379371239-nodered-5c8ff67e8-1r7ht 0/1      Pending      0                3d7h

```

รูปที่ ก. ๑๓ สถิติ User ที่ Active อยู่บน CYBLION.io ณ วันที่ ๑๙ มีนาคม ๒๕๖๓

๖.๔) การอบรมการใช้งานระบบบริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT

จากการศึกษาความต้องการใช้งานระบบ IoT ภายในโรงงานอุตสาหกรรมเป้าหมายและกรณีศึกษาของโรงงานอุตสาหกรรมที่ต้องการเข้าร่วมทดสอบตามหลักการที่ใช้คัดเลือกสำหรับการติดตั้งทดสอบในโรงงานนำร่องตามที่กำหนด โดยคำนึงถึงปัจจัยความพร้อมของโรงงานคู่ทดสอบ ทำให้เลือกโรงงานอุตสาหกรรมต้นแบบ เพื่อทำการวางระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล อบรมการติดตั้งใช้งาน การตรวจสอบผลการติดตั้ง การใช้งานระบบ การดูแลรักษา รวมถึงปรึกษาหารือสร้างความเข้าใจในการนำเข้าสู่ข้อมูล การเข้ารหัส และการรับข้อมูลเพื่อการเรียกใช้ข้อมูลผ่านระบบที่ติดตั้งไว้ จำนวน ๒ โรงงาน ได้แก่

๖.๓.๑ โรงงาน Thanakorn Vegetable Oil Products Co., Ltd.



รูปที่ ก. ๑๔ การทดสอบระบบและความเร็วของสัญญาณ 5G ที่ใช้ในการสื่อสาร



รูปที่ ก. ๑๕ การอบรมสอนใช้งานแพลตฟอร์ม CYBLION ให้กับวิศวกรบริเวณพื้นที่ติดตั้ง Edge Computing

๖.๓.๒ โรงงาน บริษัท เดอะ เพ็ท จำกัด



รูปที่ ก. ๑๖ พื้นที่โกดังน้ำมันพืช โกดังกากพืชน้ำมัน ฟาร์มไซโลน้ำมัน และส่วนที่ต้องการติดตั้งกล้อง CCTV
ในบริเวณโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP)

๖.๔) เอกสารประกอบการบรรยายการจัดอบรม

ทีมงานได้จัดทำเอกสารประกอบการบรรยายการจัดอบรมเชิงปฏิบัติการโดยส่งมอบเอกสารให้แก่ผู้เข้ารับการอบรม ดังนี้

๑. เอกสารประกอบการอบรม Privacy as a Service for IIoT/IIoT
๒. เอกสารประกอบการอบรม Node-RED (Node-RED Get Started)
๓. เอกสารประกอบการอบรม CYBLION Platform Training & Tutorial
๔. คู่มือการใช้งานแพลตฟอร์ม CYBLION (USER Guide Manual VOL.0.1)

๖.๕) ผลงานตีพิมพ์ในงานประชุมวิชาการ

ทีมงานได้นำเสนอผลการดำเนินโครงการและมีผลงานที่ตีพิมพ์ในงานประชุมวิชาการ จำนวน ๔ เรื่อง โดยชื่อผลงานที่ตีพิมพ์ตามตารางดังนี้

ตารางที่ ก. ๕ ผลงานตีพิมพ์ในงานประชุมวิชาการ

ลำดับ	บทความ	ประเภท
๑.	T. Hohum, K. Suksomboon, A. Niruntasukrat, S. Mongkolluksamee, N. Tansangworn, "ScottishFold: CatBoost-Enabled Lightweight Autonomous Smart Home Device Classification," Proceedings of IEEE Globecom Workshops (GC Wkshps), December 2021.	งานประชุมวิชาการ นานาชาติ (Full Paper) เอกสารตอบรับใน ภาคผนวก จ.
๒.	N. Tanabodee, K. Suksomboon, C. Issariyapat, S. Mongkolluksamee, A. Niruntasukrat, N. Tansangworn, S. Kitisin, "CipherFlow: A Playground for Developing Privacy-Preserving IoT in Node-RED" Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '22), December 2022.	งานประชุมวิชาการ นานาชาติ (Full Paper) เอกสารตอบรับใน ภาคผนวก จ.
๓	K. Suksomboon, N. Tanabodee, E. Saktawornlerd, A. Niruntasukrat, C. Issariyapat, S. Mongkolluksamee, N. Tansangworn, S. Kitisin, "Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0," Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '23), December 2023	งานประชุมวิชาการ นานาชาติ (Poster) เอกสารตอบรับใน ภาคผนวก จ.
๔.	C. Phoojaroenchanachai, K. Kaemarungsi and P. Pongpaibool, "5G Use Cases for Thailand's Smart Factory/Manufacturing," Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '23), December 2023	งานประชุมวิชาการ นานาชาติ (Poster) เอกสารตอบรับใน ภาคผนวก จ.

๖.๖) บทความทางวิชาการระดับชาติ (ฉบับยื่นเสนอบทความ)

ทีมวิจัยได้จัดเขียนบทความวิชาการระดับนานาชาติ เรื่อง "แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IIoT," NBTC Journal 2024.

ตารางที่ ก. ๖ บทความทางวิชาการระดับชาติ

ลำดับ	บทความ	ประเภท
๑.	Kalika Suksomboon, Aimaschana Nirantasukrat, Nataset Tanabodee, Eakarat Saktawornlerd, Chaveee Issariyapat, Tawan Hohum, Sophon Mongkolluksamee, Natapon Tansangworn, Sukumal Kitisin, "Privacy as a Service for IoT Platform," NBTC Journal 2024	วารสารวิชาการ ระดับชาติ (Journal) เอกสารตอบรับการ ส่งในภาคผนวก จ.

๖.๗) บทความทางวิชาการระดับนานาชาติ (ฉบับยื่นเสนอบทความ)

ทีมวิจัยได้จัดเขียนบทความวิชาการระดับนานาชาติ เรื่อง "CipherFlow: A Homomorphic Operation Toolkit for Privacy-Preserving Industrial IoT Application," IEEE Internet of Things Journal

ตารางที่ ก. ๗ บทความทางวิชาการระดับนานาชาติ

ลำดับ	บทความ	ประเภท
๑.	K. Suksomboon, N. Tanabodee, A. Niruntasukrat, S. Kitisin, J. Kurihara, T. Tanaka, S. Mongkolluksamee, N. Tansangworn, C. Issariyapat, "CipherFlow: A Homomorphic Operation Toolkit for Privacy-Preserving Industrial IoT Application," IEEE Internet of Things Journal	วารสารวิชาการ นานาชาติ (Journal) อยู่ในระหว่าง เตรียมการจัดส่ง

บทคัดย่อ

โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT

อุตสาหกรรม ๔.๐ ได้รับพลังขับเคลื่อนจากอุตสาหกรรมของอินเทอร์เน็ตของสรรพสิ่ง (Industrial Internet of Things (IIoT)) ซึ่งทำให้การสื่อสารระหว่างโลกดิจิทัลและกายภาพหลอมรวมกัน อย่างไรก็ตามความเป็นส่วนตัวของข้อมูล IIoT ยังคงเป็นอุปสรรคที่สำคัญต่อการเปลี่ยนแปลงไปสู่ อุตสาหกรรม ๔.๐ เนื่องจากการนำข้อมูลเหล่านั้นไปคำนวณที่คลาวด์ของบุคคลที่สามจำเป็นต้องเปิดเผยข้อมูล ซึ่งสร้างความกังวลใจในการรักษาความลับของข้อมูล ทั้งนี้ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์ (NECTEC) รับรู้ถึงปัญหานี้จึงทำหน้าที่รับผิดชอบในการดำเนินโครงการวิจัยที่ได้รับทุนจาก กทปส. เพื่อแก้ปัญหานี้ โดยการเพิ่มศักยภาพการใช้บริการคลาวด์และระบบไอโอทีด้วยเทคโนโลยีการเข้ารหัสโฮโมมอร์ฟิก (Homomorphic Encryption (HE)) เพื่อให้การคำนวณที่คลาวด์เป็นไปได้โดยข้อมูลไม่ถูกเปิดเผย ในขณะที่นักพัฒนาโซลูชัน IoT สามารถให้ความสำคัญกับการพัฒนา IoT โดยไม่ต้องมีความเชี่ยวชาญด้านวิทยาการเข้ารหัส โครงการนี้เสนอแนวทางการรักษาความเป็นส่วนตัวสำหรับไอโอทีภายใต้ชื่อแพลตฟอร์ม "CYBLION" เพื่อให้การคำนวณข้อมูลที่คลาวด์ไม่ต้องถูกถอดรหัสโดยใช้เทคโนโลยี HE ในขณะที่แอปพลิเคชันบนมือถือถูกใช้ในการควบคุมความเป็นส่วนตัวของข้อมูล IIoT ที่สามารถควบคุมการสร้างความความเป็นส่วนตัวเฉพาะเจาะจงต่อโปรเจกต์ เอ็ดจ์ และการจัดการกุญแจรหัสลับ แอปพลิเคชันมือถือจะทำการถอดรหัสผลลัพธ์ของข้อมูลที่ถูกเข้ารหัสและแสดงข้อมูลในโหมดเรียลไทม์ แพลตฟอร์ม CYBLION ถูกพัฒนาขึ้นบนโปรแกรม Node-RED และ HE ที่เข้ามาแก้ไขปัญหาที่ท้าทายสำหรับการพัฒนา HE ให้สามารถใช้งานได้กับระบบไอโอที ซึ่งมีทั้งหมด ๔ ข้อด้วยกัน ได้แก่

- ๑) การคำนวณทั่วไปสำหรับโปรแกรม IIoT
- ๒) การส่งข้อมูลที่ถูกเข้ารหัสผ่านโปรโตคอล MQTT
- ๓) การแสดงผลข้อมูลแบบเรียลไทม์
- ๔) การทำให้ง่ายสำหรับผู้เชี่ยวชาญที่ไม่ใช่การเข้ารหัส

ด้วย CYBLION นักพัฒนาโซลูชัน IoT สามารถให้ความสำคัญกับการพัฒนา IoT โดยไม่ต้องค้นหาค่าพารามิเตอร์ที่เหมาะสมสำหรับการเข้ารหัส ผลสำเร็จของ CYBLION พิสูจน์ได้จากการสาธิตผ่านการนำไปใช้งานจริงในกรณีศึกษาของโรงงานอุตสาหกรรมสองแห่ง ได้แก่ ๑) TVOP สำหรับระบบตรวจสอบระดับน้ำมันในไซโล และ ๒) PET สำหรับระบบตรวจสอบอุณหภูมิและความชื้น โดยการใช้เทคโนโลยี HE ในการเพิ่มความเป็นส่วนตัวให้กับ IIoT นอกจากนี้การนำเสนอผลงานผ่านการจัดกิจกรรมสัมมนาเชิงปฏิบัติการได้ถูกดำเนินการเพื่อส่งเสริมผู้มีส่วนได้ส่วนเสีย ได้แก่ นักพัฒนาระบบไอโอที และวิศวกรโรงงาน เพื่อให้พวกเขาเรียนรู้ถึงความสำคัญของความเป็นส่วนตัวของข้อมูล IIoT และวิธีการใช้ CYBLION เพื่อช่วยลดข้อกังวลเกี่ยวกับการรั่วไหลข้อมูล จากการสำรวจโครงการ ผู้เข้าร่วมงานทั้งหมด ๙๓.๓% ประทับใจกับแอปพลิเคชัน CYBLION และมีมากกว่า ๕๐% ทราบว่าปัญหาเรื่องความเป็นส่วนตัวเป็นสิ่งสำคัญสำหรับการมุ่งพัฒนาโรงงานของตนไปสู่โรงงานอัจฉริยะ

Abstract

Industry 4.0 and beyond are empowered by the Industrial Internet of Things (IIoT), enabling seamless communication between the physical and digital worlds. Nevertheless, IIoT privacy remains a significant obstacle because outsourcing computation to a third-party cloud raises data leak concerns for manufacturing owners. The National Electronics and Computer Technology Center (NECTEC) has realized this issue and taken responsibility for conducting a research project funded by the Broadcasting and Telecommunications Research and Development Fund for Public Interest (BTFP) under the supervision of Office of The National Broadcasting and Telecommunications Commission (NBTC) on enhancing homomorphic encryption (HE) technology to enable privacy-preserving outsourced computation on the cloud. At the same time, IoT solution developers can focus on IoT development rather than cryptographic optimization. This project proposes a privacy-preserving industry 4.0 and beyond that is empowered by the Industrial Internet of Things (IIoT), enabling seamless communication between the physical and digital worlds. Nevertheless, IIoT privacy remains a significant obstacle because outsourcing computation to a third-party cloud raises data leak concerns for manufacturing owners. The National Electronics and Computer Technology Center (NECTEC) has realized this issue and taken responsibility for conducting a research project funded by NBTC. This project proposes a privacy-preserving-as-a-service for IoT platform called “CYBLION,” enabling privacy-preserving outsourced computation on the cloud powered by a HE technology while a mobile application is used to control the privacy of IIoT data specific to a project, an edge, and key management. The mobile application decrypts the results of computed encrypted data and displays the data in real-time. CYBLION is developed based on the Node-RED program and HE that overcomes four main challenges:

- 1) common arithmetic for IIoT applications,
- 2) transmitting encrypted data over MQTT protocol,
- 3) real-time data visualization, and
- 4) simplification for non-cryptographic specialists.

With CYBLION, IoT solution developers can focus on IoT development rather than cryptographic optimization. The practicality of CYBLION is demonstrated by implementing it through two real smart factory use cases: 1) TVOP for a smart tank farm monitoring system and 2) The PET for a temperature and humidity monitoring system. The successful deployment of HE is a promising technology enhancing privacy to IIoT applications. In addition, the hands-on workshop has been conducted to encourage stakeholders, such as system integrators and factory engineers, to

realize the importance of the privacy of IIoT data and how to use CYBLION to help alleviate their concerns about data leakage. From the project's survey, 93.3% of attendees were impressed with the CYBLION application, and more than 50% realized that privacy issues are important for moving toward smart factories.

สารบัญ

บทสรุปผู้บริหาร.....	ค
บทคัดย่อ โครงการพัฒนาการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IOT	ธ
ABSTRACT	น
บทที่ ๑ บทนำ	๑
๑.๑ ที่มา และความสำคัญของโครงการ	๑
๑.๒ สารสำคัญของโครงการ.....	๓
๑.๒.๑ วัตถุประสงค์	๓
๑.๒.๒ เป้าหมาย	๓
๑.๒.๓ ผลผลิตสำคัญ.....	๔
๑.๒.๔ ประโยชน์ที่คาดว่าจะได้รับ	๔
บทที่ ๒ ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	๖
๒.๑ ทฤษฎี และแนวความคิด.....	๖
๒.๑.๑ IoT Cloud Platform.....	๖
๒.๑.๒ เทคโนโลยีคุ้มครองความปลอดภัยของข้อมูลใน IoT Cloud Platform.....	๗
๒.๒ ผลงานวิจัยที่เกี่ยวข้องกับเทคโนโลยีการเข้ารหัสแบบ HE.....	๑๒
๒.๒.๑ ประเภทของการเข้ารหัสแบบ HE.....	๑๔
๒.๒.๒ เทคนิคการเข้ารหัสแบบ HE	๑๕
๒.๒.๓ ฟังก์ชันการทำโอเพอร์เรชันพื้นฐาน	๑๖
๒.๒.๔ แนวทางการนำไปประยุกต์ใช้งานเทคนิคการเข้ารหัสแบบ HE.....	๑๗
๒.๒.๕ HE Libraries.....	๒๕
๒.๓ ผลงานศึกษาการนำเทคโนโลยี HE มาให้บริการในต่างประเทศ.....	๒๖
๒.๓.๑ วิเคราะห์บริษัทที่ให้บริการคุ้มครองความเป็นส่วนตัว (Privacy as a Service).....	๒๖
๒.๓.๒ วิเคราะห์ตลาด Privacy as a Service ต่างประเทศ.....	๓๐
๒.๓.๓ วิเคราะห์กลุ่มลูกค้าในตลาดต่างประเทศ	๓๘
๒.๔.๔ วิเคราะห์เทคโนโลยีการให้บริการคุ้มครองข้อมูลส่วนบุคคลในต่างประเทศมาใช้กับ NETPIE	๓๙
บทที่ ๓ ระเบียบวิธีวิจัย/วิธีการและขั้นตอนการดำเนินงานวิจัย.....	๔๑
๓.๑ ศึกษาความต้องการใช้งานระบบไอโอทีภายในโรงงานอุตสาหกรรม	๔๑
๓.๑.๑ โรงงาน Thanakorn Vegetable Oil Products Co., Ltd.	๔๑
๓.๑.๒ โรงงาน บริษัท เดอะ เพ็ท จำกัด	๔๔

๓.๒ ศึกษาการทำงานของ CODE LIBRARY และ COMPILER ของ HE เพื่อการพัฒนาต่อยอด	๔๕
๓.๓ ออกแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอที	๔๖
๓.๔ การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับ IoT	๔๙
บทที่ ๔ การสร้างแพลตฟอร์ม ผลการวิจัย และผลการติดตั้งใช้งานจริง	๕๒
๔.๑ การสร้างแพลตฟอร์มบริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT	๕๒
๔.๑.๑ Software Toolkit บน Node-RED (CipherFlow).....	๕๒
๔.๑.๒ แพลตฟอร์ม CYBLION.....	๖๑
๔.๒ ผลการทดสอบแพลตฟอร์มในห้องปฏิบัติการ	๗๓
๔.๒.๑ ผลการทดสอบพีเจอาร์การใช้งาน.....	๗๓
๔.๒.๒ ผลการทดสอบประสิทธิภาพของแพลตฟอร์ม.....	๗๔
๔.๒.๓ ผลการทดสอบของพารามิเตอร์ของการเข้ารหัสและขนาดของกุญแจ	๗๔
๔.๒.๔ ผลการทดสอบประสิทธิภาพของพารามิเตอร์ของการเข้ารหัสและขนาดของกุญแจ.....	๗๕
๔.๓ ผลการทดสอบแพลตฟอร์มภาคสนาม	๗๗
๔.๓.๑ การทดสอบใช้งานแพลตฟอร์ม Cyblion กับบริษัท ธนาคารผลิตภัณฑ์น้ำมันพืช จำกัด	๗๘
๔.๓.๒ การทดสอบใช้งานแพลตฟอร์ม Cyblion กับ บริษัท เดอะ เพ็ท จำกัด	๙๐
บทที่ ๕ การเผยแพร่ผลการดำเนินงานวิจัยการ	๙๖
๕.๑ การจัดฝึกอบรมเชิงปฏิบัติการ และการระดมความคิด	๙๖
๕.๒ สรุปผลการประเมินการใช้งานแพลตฟอร์ม	๙๘
๕.๓ การระดมความคิดเห็น	๑๐๓
๕.๔ การประชาสัมพันธ์ จัดทำสื่อโปสเตอร์ และ สื่อวีดิทัศน์สำหรับเผยแพร่.....	๑๐๔
๕.๕ ผลงานอื่นๆ	๑๐๗
๕.๕.๑ ต้นแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที	๑๐๗
๕.๕.๒ สถิติการเข้าเยี่ยมชมและติดตั้งแอปพลิเคชัน CYBLION	๑๐๘
๕.๕.๓ สถิติการใช้งาน CYBLION.io	๑๑๐
๕.๕.๔ การติดตั้งและอบรมการใช้งานระบบบริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT	๑๑๑
๕.๕.๕ เอกสารประกอบการบรรยายการจัดอบรม.....	๑๑๒
๕.๕.๖ ผลงานตีพิมพ์ในงานประชุมวิชาการ	๑๑๒
๕.๕.๔ บทความทางวิชาการระดับชาติ (ฉบับยื่นเสนอบทความ)	๑๑๓
บทที่ ๖ สรุปผลการวิจัย และข้อเสนอแนะ.....	๑๑๕
๖.๑ สรุปผลการวิจัย.....	๑๑๕
๖.๒ ข้อเสนอแนะ.....	๑๑๖

บรรณานุกรม	๑๑๘
ภาคผนวก ก. กำหนดการและรายชื่อผู้เข้าร่วมประชุมเชิงปฏิบัติการ.....	๑๒๕
ภาคผนวก ข. เอกสารประกอบการบรรยาย การฝึกอบรมเชิงปฏิบัติการ	๑๓๑
ภาคผนวก ค. บทความในงานประชุมวิชาการ	๑๕๗
ภาคผนวก ง. บทความระดับนานาชาติ	๑๗๕
ภาคผนวก จ. เอกสารตอบรับบทความ	๑๙๒
1. แบบตอบรับการนำส่งบทความ NBTC JOURNAL	๑๙๓
2. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ AINTEC 2023.....	๑๙๔
3. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ AINTEC 2022.....	๑๙๔
4. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ IEEE GLOBECOM 2021	๑๙๕
ประวัตินักวิจัย	๑๙๖

สารบัญตาราง

ตารางที่ ๑ การเปรียบเทียบเทคนิคการคุ้มครองความเป็นส่วนตัวของข้อมูลที่ถูกส่งออกไปประมวลผล	๑๑
ตารางที่ ๒ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ MACHINE LEARNING และ DATA MINING	๑๘
ตารางที่ ๓ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ IMAGE PROCESSING	๑๙
ตารางที่ ๔ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ BIOMETRIC AUTHENTICATION	๒๒
ตารางที่ ๕ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ LOCATION-BASED SERVICE	๒๓
ตารางที่ ๖ การประยุกต์ใช้งานเทคนิค HE กับ IOT DEVICES	๒๔
ตารางที่ ๗ COMPILER และ ไบรารีสำหรับเทคนิคการเข้ารหัสแบบ HE	๒๕
ตารางที่ ๘ ผลการทดสอบประสิทธิภาพของเทคนิค NUFHE	๒๙
ตารางที่ ๙ ชุด CODE LIBRARY และ COMPILER ที่รองรับเทคนิคการเข้ารหัสแบบ RNS-CKKS	๔๖
ตารางที่ ๑๐ การกำหนดพารามิเตอร์สภาพแวดล้อม HE สำหรับการเข้ารหัสแบบ CKKS/RNS-CKKS ตาม	
มาตรฐานความปลอดภัย [78]	๕๔
ตารางที่ ๑๑ พารามิเตอร์สภาพแวดล้อม FHE แบบ CKKS/RNS-CKKS ตามมาตรฐาน SECURITY-LEVEL 128	๕๖
ตารางที่ ๑๒ เปรียบเทียบระยะเวลาการรอส่งข้อมูลแบบข้อมูลเดียวและแบบ BATCHING	๕๙
ตารางที่ ๑๓ เปรียบเทียบ MQTT BROKER ที่ให้บริการในปัจจุบันกับ CYBLION	๖๕
ตารางที่ ๑๔ ผลการทดสอบฟีเจอร์การใช้งานแพลตฟอร์ม	๗๓
ตารางที่ ๑๕ ผลการทดสอบฟีเจอร์การใช้งาน MOBILE APP	๗๓
ตารางที่ ๑๖ พารามิเตอร์สำหรับสร้างสภาพแวดล้อมโฮมมอร์ฟิกแบบ CKKS/RNS-CKKS	๗๔
ตารางที่ ๑๗ สื่อวีดิทัศน์สำหรับเผยแพร่ข้อมูลการอบรม	๑๐๖
ตารางที่ ๑๘ แอปพลิเคชันที่ได้รับจากการดำเนินโครงการ	๑๐๗
ตารางที่ ๑๙ ผลงานตีพิมพ์ในงานประชุมวิชาการ	๑๑๓
ตารางที่ ๒๐ บทความทางวิชาการระดับชาติ	๑๑๓
ตารางที่ ๒๑ บทความทางวิชาการระดับนานาชาติ	๑๑๔

สารบัญภาพ

รูปที่ ๑ ไดอะแกรมแสดงการส่งข้อมูลจากอุปกรณ์ IOT เพื่อไปประมวลผลในคลาวด์	๗
รูปที่ ๒ ตัวอย่างการส่งข้อมูลจากอุปกรณ์ IOT ไปประมวลผลในคลาวด์แบบที่ใช้กันทั่วไป	๘
รูปที่ ๓ ตัวอย่างการนำ ARM TRUSTZONE ไปใช้งานกับ IOT [9]	๙
รูปที่ ๔ ตัวอย่างการนำ INTEL SGX ไปใช้งานกับ IOT [10]	๑๐
รูปที่ ๕ โมเดลระบบการส่งข้อมูลไปประมวลผลบนคลาวด์ด้วยเทคโนโลยี HE แบบทั่วไป	๑๓
รูปที่ ๖ แผนผังการจำแนกประเภทของ HE	๑๔
รูปที่ ๗ แผนภาพวิวัฒนาการของการพัฒนาอัลกอริทึมการเข้ารหัสแบบ HE [69]	๑๕
รูปที่ ๘ กระบวนการให้บริการการประมวลผล MACHINE LEARNING และ DATA MINING บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๑๙
รูปที่ ๙ กระบวนการให้บริการการประมวลผลการดึงข้อมูล IMAGE FEATURE EXTRACTION บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๐
รูปที่ ๑๐ กระบวนการให้บริการการประมวลผลการ IMAGE RETRIEVAL บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๑
รูปที่ ๑๑ กระบวนการให้บริการการประมวลผลการ IMAGE WATERMARKING บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๑
รูปที่ ๑๒ กระบวนการให้บริการการประมวลผลการ BIOMETRIC AUTHENTICATION บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๒
รูปที่ ๑๓ กระบวนการให้บริการการประมวลผลการสร้างโมเดล K-NN บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๔
รูปที่ ๑๔ กระบวนการให้บริการการประมวลผลข้อมูลจากอุปกรณ์ IOT บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล	๒๕
รูปที่ ๑๕ สถาปัตยกรรมการคำนวณอย่างปลอดภัย เพื่อบริการคุ้มครองความเป็นส่วนตัว ส่วนตัวของข้อมูลบริษัท INPHER	๒๗
รูปที่ ๑๖ การคำนวณอย่างปลอดภัยสำหรับบริการ DATA-BLIND MATCHING ของบริษัท INPHER	๒๘
รูปที่ ๑๗ วิธีการทำงานของระบบการเข้ารหัสแบบ FHE	๒๘
รูปที่ ๑๘ ไดอะแกรมการ TRAIN ML MODEL โดยให้ความปลอดภัยและความคุ้มครองข้อมูลส่วนบุคคล ภายใต้ผลิตภัณฑ์ของบริษัท ZAMA	๒๙
รูปที่ ๑๙ การ COMPLIE MLโมเดลที่สร้างไว้แล้วให้เป็น ML โมเดลที่ถูกเข้ารหัส	๒๙
รูปที่ ๒๐ บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน MICROSOFT AZURE MARKETPLACE	๓๐
รูปที่ ๒๑ รายละเอียดด้านเทคนิคสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ENVEIL ZERO REVEAL	๓๑

รูปที่ ๒๒ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ENVEIL ZERO REVEAL	๓๑
รูปที่ ๒๓ รายละเอียดด้านเทคนิคสำหรับบริการคุ้มครองความเป็นส่วนตัวของ OPEN MINED GRID DOMAIN	๓๒
รูปที่ ๒๔ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล OPENMINED GRID DOMAIN	๓๒
รูปที่ ๒๕ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล TRIPLEBLIND	๓๓
รูปที่ ๒๖ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล OPENMINED GRID DOMAIN	๓๓
รูปที่ ๒๗ บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน AWS MARKETPLACE	๓๔
รูปที่ ๒๘ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ENVEIL ZERO REVEAL CLIENT	๓๕
รูปที่ ๒๙ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ENVEIL ZERO REVEAL SERVER	๓๕
รูปที่ ๓๐ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ENVEIL TRIPLEBLIND	๓๖
รูปที่ ๓๑ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล INPHER บน AWS MARKETPLACE	๓๗
รูปที่ ๓๒ รายละเอียดด้านราคาสำหรับบริการคุ้มครองความเป็นส่วนตัวของ ข้อมูล CRYPTO-LAB บน AWS MARKETPLACE	๓๘
รูปที่ ๓๓ มูลค่าทางการตลาดของ DATA PROTECTION AS A SERVICE ในตลาดโลก [76]	๓๙
รูปที่ ๓๔ สัดส่วนทางการตลาดของ DATA PROTECTION AS A SERVICE ในตลาดโลก [76]	๓๙
รูปที่ ๓๕ แสดงไฮไลต์เกี่ยวกับวัตถุดิบของโรงงาน บริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (บริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด, ม.ป.ป.)	๔๒
รูปที่ ๓๖ พื้นที่โกดังน้ำมันพืช โกดังกากพืชน้ำมัน ฟาร์มไซโลน้ำมัน และส่วนที่ต้องการติดตั้งกล้อง CCTV ในบริเวณโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP)	๔๓
รูปที่ ๓๗ ระบบ CONDITION-BASED MONITORING สำหรับ SMART TANK FARM	๔๓
รูปที่ ๓๘ ระบบ CONDITION-BASED MONITORING สำหรับระบบวัดอุณหภูมิ (TEMPERATURE) และความชื้น (HUMIDITY) ของเครื่องจักร	๔๕
รูปที่ ๓๙ ไดอะแกรมการทำงานของ RNS-CKKS	๔๗
รูปที่ ๔๐ ไดอะแกรมการสร้างพารามิเตอร์ CONTEXT	๔๗
รูปที่ ๔๑ ไดอะแกรมการสร้างชุดกุญแจสำหรับ RNS-CKKS	๔๗
รูปที่ ๔๒ ไดอะแกรมการ ENCODE ข้อความ MESSAGE	๔๗

รูปที่ ๔๓ ไดอะแกรมการเข้ารหัส	๔๘
รูปที่ ๔๔ ไดอะแกรมการคำนวณด้วยข้อมูลที่ถูกรหัส	๔๘
รูปที่ ๔๕ ไดอะแกรมการถอดรหัส	๔๘
รูปที่ ๔๖ สถาปัตยกรรมแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที	๔๙
รูปที่ ๔๗ สถาปัตยกรรมของ NODE-RED CIPHERFLOW	๕๒
รูปที่ ๔๘ NODE-RED-CIPHERFLOW	๕๓
รูปที่ ๔๙ ตัวอย่างการใช้งาน NODE-RED CIPHERFLOW	๕๔
รูปที่ ๕๐ การ ENCODE/DECODE ข้อมูล	๕๕
รูปที่ ๕๑ การ ENCRYPT/DECRYPT ข้อมูล	๕๕
รูปที่ ๕๒ การคูณข้อมูลเข้ารหัส (e^* , e_1 , e_2 คือ NOISE ที่เกิดขึ้นจากการเข้ารหัส)	๕๖
รูปที่ ๕๓ การเลือกพารามิเตอร์ RNS-CKKS บน NODE INPUT	๕๗
รูปที่ ๕๔ การเลือกพารามิเตอร์ RNS-CKKS บน NODE ENCRYPT	๕๗
รูปที่ ๕๕ การเลือกพารามิเตอร์ RNS-CKKS บน NODE DECRYPT	๕๗
รูปที่ ๕๖ การเลือกพารามิเตอร์ RNS-CKKS บน NODE SETCONTEXT	๕๘
รูปที่ ๕๗ การสร้างกุญแจ	๕๘
รูปที่ ๕๘ การเข้ารหัสข้อมูลโฮโมมอร์ฟิกแบบเดี่ยว	๕๙
รูปที่ ๕๙ การเข้ารหัสข้อมูลโฮโมมอร์ฟิกแบบ BATCHING	๕๙
รูปที่ ๖๐ การจัดเรียงข้อมูลไอโอทีใน ARRAY สำหรับการส่งข้อมูลแบบ BATCH	๖๐
รูปที่ ๖๑ การจัดเรียงข้อมูลแบบ FEED ลงบนกราฟแสดงผลตามแกนเวลาสำหรับการส่งข้อมูลแบบเดี่ยว สำหรับการส่งข้อมูลที่ไม่เข้ารหัส	๖๐
รูปที่ ๖๒ การจัดเรียงข้อมูลแบบ FEED ลงบนกราฟแสดงผลตามแกนเวลา สำหรับการส่งข้อมูลแบบ BATCHING ของข้อมูลเข้ารหัส	๖๑
รูปที่ ๖๓ NODE CYBLION ENCRYPT	๖๒
รูปที่ ๖๔ การตั้งค่า NODE CYBLION ENCRYPT	๖๒
รูปที่ ๖๕ การแลกกุญแจผ่าน QR CODE	๖๒
รูปที่ ๖๖ จำนวนผู้ใช้งานบนคลาวด์ ๑๐ ราย	๖๓
รูปที่ ๖๗ การเข้าถึงพื้นที่สำหรับสร้างสูตรคำนวณด้วย PROJECT พร้อมแสดงจำนวน EDGE DEVICE ในโปรเจกต์	๖๓
รูปที่ ๖๘ การสร้างสูตรคำนวณข้อมูลเข้ารหัสในโปรเจกต์	๖๔
รูปที่ ๖๙ ตัวอย่างสูตรคำนวณข้อมูลเข้ารหัสในโปรเจกต์ ($y = 30x - 118.02$)	๖๔
รูปที่ ๗๐ การสื่อสารข้อมูลไอโอทีด้วย MQTT PROTOCOL	๖๕
รูปที่ ๗๑ การพัฒนา MOBILE APPLICATION ด้วย XAMARIN.FORMS	๖๗
รูปที่ ๗๒ DIAGRAM การพัฒนา IOS APPLICATION ด้วย XAMARIN	๖๗

รูปที่ ๗๓ DIAGRAM การพัฒนา ANDROID APPLICATION ด้วย XAMARIN	๖๘
รูปที่ ๗๔ ข้อความแจ้งเตือน PREPARED ARCHIVE FOR VALIDATION FAILED	๖๘
รูปที่ ๗๕ ไดอะแกรมกระบวนการ BUILD APPLICATION และอัปโหลดขึ้น TESTFLIGHT	๖๙
รูปที่ ๗๖ UI สำหรับการสร้าง PROJECT	๖๙
รูปที่ ๗๗ UI สำหรับการสร้าง EDGE COMPUTING	๗๐
รูปที่ ๗๘ UI สำหรับการสร้าง KEYS	๗๐
รูปที่ ๗๙ UI สำหรับการส่ง KEY ไปยังอุปกรณ์ EDGE COMPUTING	๗๑
รูปที่ ๘๐ UI สำหรับการส่ง KEY ไปยัง CLOUD	๗๑
รูปที่ ๘๑ UI สำหรับการแสดงผลข้อมูลแบบ GAUGE	๗๒
รูปที่ ๘๒ UI สำหรับการแสดงผลข้อมูลแบบ GRAPH	๗๒
รูปที่ ๘๓ ทริพยากรที่ถูกใช้งานในคลาวด์	๗๔
รูปที่ ๘๔ ความเร็วในการประมวลผลฟังก์ชัน HE บน IOT 2050	๗๕
รูปที่ ๘๕ ความเร็วในการประมวลผลฟังก์ชัน HE บน JETSON NANO	๗๖
รูปที่ ๘๖ ความเร็วในการประมวลผลฟังก์ชัน HE บน RASPBERRY PI 3 MODEL B	๗๖
รูปที่ ๘๗ ความเร็วในการประมวลผลฟังก์ชัน HE บน RASPBERRY PI 4	๗๗
รูปที่ ๘๘ ความเร็วในการประมวลผลฟังก์ชัน HE บน CLOUD	๗๗
รูปที่ ๘๙ การเชื่อมต่อระบบไอโอทีสำหรับการมอนิเตอร์ระดับน้ำมันในไซลอน้ำมันกับแพลตฟอร์มไอโอทีคลาวด์	๗๘
รูปที่ ๙๐ ความสัมพันธ์ระหว่างค่า SENSOR และระยะทาง	๘๐
รูปที่ ๙๑ การทดสอบระบบ IOT ในห้องปฏิบัติการ	๘๐
รูปที่ ๙๒ หน้าต่างการเข้าถึงโปรแกรม NODE-RED ใน EDGE COMPUTING	๘๑
รูปที่ ๙๓ การสร้าง DATA FLOW สำหรับการส่งข้อมูล SENSOR ไปยัง NETPIE IOT CLOUD	๘๑
รูปที่ ๙๔ การกำหนด PAYLOAD สำหรับ TOPIC “TEMP_MODULE”	๘๒
รูปที่ ๙๕ การกำหนด PAYLOAD สำหรับ TOPIC “SENSOR1”	๘๒
รูปที่ ๙๖ การกำหนด PAYLOAD สำหรับ TOPIC “SENSOR2”	๘๒
รูปที่ ๙๗ กำหนดให้ส่งข้อมูลเข้าสู่ @SHADOW/DATA/UPDATE ของ NETPIE	๘๓
รูปที่ ๙๘ กำหนดให้ส่งข้อมูลเข้าสู่ MQTT BROKER ของ NETPIE	๘๓
รูปที่ ๙๙ การติดตั้ง NODE สำหรับการเข้ารหัสเพิ่มเติมสำหรับการเข้ารหัสข้อมูลไอโอที และส่งออกไปยังคลาวด์ CYBLION	๘๔
รูปที่ ๑๐๐ การตั้ง TOPIC สำหรับส่งข้อมูลเข้ารหัสเพื่อส่งออกไปยังคลาวด์ CYBLION ผ่าน EDGE BROKER @CSTORE/DATA/UPDATED/	๘๔
รูปที่ ๑๐๑ การสร้างสูตรการคำนวณบนคลาวด์ CYBLION โดยข้อมูลที่ต้องการคำนวณจะผ่าน FLOW ของการคำนวณและ ข้อมูล TIMESTAMP จะไม่ผ่าน FLOW การคำนวณ	๘๕

รูปที่ ๑๐๒ การติดตั้ง SENSOR บนโซลีน้ำมันขนาดความสูง ๑๕ เมตร	๘๖
รูปที่ ๑๐๓ การติดตั้ง SENSOR บนโซลีน้ำมันขนาดความสูง ๑๕ เมตร	๘๖
รูปที่ ๑๐๔ การติดตั้งอุปกรณ์สำหรับ EDGE COMPUTING ใกล้โซลีน้ำมันที่ต้องการตรวจวัด	๘๗
รูปที่ ๑๐๕ ตู้เก็บอุปกรณ์สำหรับ EDGE COMPUTING ที่ออกแบบให้ทนความร้อนและความชื้น เพื่อติดตั้งกลางแจ้ง มีระบบไฟฟ้าสำหรับจ่ายอุปกรณ์ IT ต่างๆ ภายในตู้และพัดลมระบายความร้อน 4 ตัวรอบทิศทาง	๘๗
รูปที่ ๑๐๖ การเข้า CONFIGURATION ในพื้นที่หน้างาน	๘๘
รูปที่ ๑๐๗ การเข้าเชื่อมต่อ NETWORK กับ EDGE COMPUTING เพื่อการแลกเปลี่ยน	๘๘
รูปที่ ๑๐๘ ตรวจสอบความแม่นยำในการถอดรหัสจากจอมอนิเตอร์ของ VEGA SENSOR	๘๘
รูปที่ ๑๐๙ ตรวจสอบการรับค่ามาถอดรหัสและแสดงผลผ่าน MOBILE APPLICATION	๘๙
รูปที่ ๑๑๐ เปรียบเทียบค่าที่ได้จาก CYBLION APP	๘๙
รูปที่ ๑๑๑ การเข้าพื้นที่โรงงาน THE PET	๙๑
รูปที่ ๑๑๒ รวบรวมหรือ USE CASE ที่ต้องการใช้ทดสอบ	๙๒
รูปที่ ๑๑๓ EDGE COMPUTING ที่ใช้ติดตั้ง NODE สำหรับเข้ารหัสทดสอบ	๙๒
รูปที่ ๑๑๔ DATA FLOW เดิมที่สำหรับการตรวจวัดค่าเซนเซอร์ของโรงงาน THE PET	๙๓
รูปที่ ๑๑๕ นักวิจัยสอนการใช้งานติดตั้ง NODE สำหรับการเข้ารหัสให้กับโรงงาน THE PET	๙๓
รูปที่ ๑๑๖ วิศวกรของโรงงาน THE PET ติดตั้ง MOBILE APPLICATION สำหรับใช้งาน	๙๔
รูปที่ ๑๑๗ วิศวกรสามารถใช้งาน MOBILE APPLICATION ได้และข้อมูลเข้ารหัสได้ถูกถอดรหัสและ แสดงผล ถูกต้องผ่าน MOBILE APPLICATION	๙๔
รูปที่ ๑๑๘ ภาพบรรยากาศการประชุมเชิงปฏิบัติการและระดมความคิดเห็น	๙๗
รูปที่ ๑๑๙ แสดงเอกสารที่ใช้ประกอบการอบรมเชิงปฏิบัติการ	๙๘
รูปที่ ๑๒๐ ประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการ AINTEC 2023	๑๐๕
รูปที่ ๑๒๑ โปสเตอร์ประชาสัมพันธ์การประชุมเชิงปฏิบัติการ	๑๐๕
รูปที่ ๑๒๒ สถิติการดาวน์โหลด CYBLION APPLICATION บนระบบปฏิบัติการ ANDROID ณ วันที่ ๒ มีนาคม พ.ศ. ๒๕๖๗ ดาวน์โหลดโดยประเทศไทย ๒๖ ครั้ง ประเทศอังกฤษ ๒ ครั้ง และประเทศลาว ๑ ครั้ง	๑๐๘
รูปที่ ๑๒๓ สถิติการดาวน์โหลด CYBLION APPLICATION บนระบบปฏิบัติการ ANDROID จำแนกตามประเทศ	๑๐๙
รูปที่ ๑๒๔ ภาพรวมสถิติการเข้าเยี่ยมชมและดาวน์โหลด CYBLION APPLICATION บนระบบปฏิบัติการ IOS	๑๐๙
รูปที่ ๑๒๕ สถิติการเข้าเยี่ยมชม CYBLION APPLICATION บนระบบปฏิบัติการ IOS	๑๑๐
รูปที่ ๑๒๖ สถิติการเข้าดาวน์โหลดและติดตั้ง CYBLION APPLICATION บนระบบปฏิบัติการ IOS	๑๑๐
รูปที่ ๑๒๗ สถิติ USER ที่ ACTIVE อยู่บน CYBLION.IO ณ วันที่ ๑๙ มีนาคม ๒๕๖๗	๑๑๑
รูปที่ ๑๒๘ การทดสอบระบบและความเร็วของสัญญาณ 5G ที่ใช้ในการสื่อสาร	๑๑๑
รูปที่ ๑๒๙ การติดตั้งและตรวจสอบข้อมูล NODE-RED บน EDGE COMPUTING	๑๑๒

รูปที่ ๑๓๐ แสดงพื้นที่โกดังน้ำมันพืช โกดังกากพืชน้ำมัน ฟาร์มไซโลน้ำมัน และส่วนที่ต้องการติดตั้งกล้อง CCTV ใน
บริเวณโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP)

๑๑๒

บทที่ ๑

บทนำ

๑.๑ ที่มา และความสำคัญของโครงการ

การขับเคลื่อนอุตสาหกรรมไทยไปสู่อุตสาหกรรม ๔.๐ เพื่อเข้าสู่ระบบอุตสาหกรรมอัจฉริยะ แนวทางหนึ่งคือการผนวกรวมเทคโนโลยีสารสนเทศเข้ากับระบบกายภาพ เพื่อให้เกิดการเชื่อมโยงข้อมูลของสรรพสิ่ง (Internet of Things (IoT)) ซึ่งก่อให้เกิดนวัตกรรมใหม่ที่เรียกว่า ระบบไซเบอร์กายภาพ (Cyber-Physical System (CPS)) ซึ่งเป็นระบบ ที่ทำให้สามารถดึงเอาข้อมูลจากสรรพสิ่งมาใช้ควบคุม ตรวจสอบ และวิเคราะห์ระบบสายผลิตแบบอัตโนมัติและมีประสิทธิภาพ ดังนั้น เทคโนโลยี IoT และระบบการกระจายตัวข้อมูลและการประมวลผลข้อมูลผ่านระบบ cloud IoT จึงเป็นหัวใจสำคัญของอุตสาหกรรม ๔.๐ ที่ทำให้การเชื่อมต่อข้อมูลการผลิต ข้อมูลวัตถุดิบ และข้อมูลเครื่องจักรกับระบบเครือข่ายเป็นไปได้ ทำให้เกิดการแลกเปลี่ยนข้อมูลระหว่างสรรพสิ่งเพื่อนำข้อมูลต่าง ๆ มาประมวลผลและควบคุมระบบอุตสาหกรรมให้มีความอัจฉริยะ

การใช้บริการ IoT ผ่าน Cloud (IoT Cloud Services) ของภาคอุตสาหกรรมในปัจจุบัน สามารถรองรับการส่งผ่านข้อมูลจากอุปกรณ์ IoT (หรือ IoT Gateway) ไปยัง Cloud Server ได้อย่างปลอดภัย โดยอาศัยช่องทางการสื่อสารแบบปลอดภัย (Secure Session ได้แก่ Secure Sockets Layer (SSL)/ Transport Layer Security (TLS)) โดยข้อมูลที่ส่งผ่านช่องทางการสื่อสารนี้จะถูกเข้ารหัส (Encryption) ทำให้ปลอดภัยจากการถูกโจมตีทางไซเบอร์จากผู้โจมตีภายนอก เช่น การดักฟังข้อมูล (Eavesdropping) ได้ นอกจากนี้ทางผู้ให้บริการคลาวด์ (Cloud IoT Service) ได้เพิ่มการระบุตัวตน (Identification) และการควบคุมการเข้าถึงข้อมูล (Access Control Security) เพื่อสร้างความปลอดภัยจากการถูกโจมตีทางไซเบอร์ (Cyber Attack)

ด้วยข้อจำกัดของวิธีการเข้ารหัส (Encryption Algorithms) ที่ใช้ในปัจจุบัน ไม่เอื้ออำนวยให้ข้อมูลที่ถูกรหัสแล้ว (Encrypted Data) นำไปประมวลผลต่อได้ ดังนั้นข้อมูลจากอุปกรณ์ IoT ที่ถูกส่งไปประมวลผลในคลาวด์จำเป็นต้องถูกเปิดเผย หรือ ถอดรหัส (Decrypt) เพื่อทำการประมวลผลด้วยข้อมูลจริงที่ไม่ถูกเข้ารหัสในคลาวด์ และผลลัพธ์ที่ได้จากการประมวลผลจะถูกนำไปแสดงผลยังส่วนแสดงผล (Data Visualization) บน Dashboard ของผู้ใช้งาน รวมทั้งการนำผลที่ได้จากการประมวลผลข้อมูลดังกล่าวไปควบคุมระบบต่อไป ด้วยเหตุนี้ผู้ให้บริการคลาวด์เองสามารถเรียนรู้ หรือล่วงรู้ข้อมูลที่ต้องการประมวลผลในคลาวด์และผลของการประมวลผลในคลาวด์ ข้อมูลเหล่านั้นอาจบ่งชี้ตัวเลขทางธุรกิจ หากรั่วไหลไปสู่ธุรกิจคู่แข่งก็จะนำมาซึ่งความเสียหายทางธุรกิจ และผลกระทบต่อความเชื่อมั่นในการใช้งานระบบ Cloud IoT ได้ จากเหตุผลดังกล่าวก่อให้เกิดความกังวลของผู้ใช้บริการ Cloud IoT Services จากการสำรวจความคิดเห็นจากโรงงานอุตสาหกรรมในประเทศไทยพบว่า มีผู้ประกอบการส่วนหนึ่งแสดงความกังวลด้านความปลอดภัยทางไซเบอร์ของข้อมูลและด้านความเป็นส่วนตัวของข้อมูลภายในโรงงานรั่วไหลหากนำเทคโนโลยี IoT มาใช้งานภายในโรงงานอุตสาหกรรม ด้วยเหตุนี้เอง ส่งผลให้ผู้ประกอบการอุตสาหกรรมลังเลที่จะหันมาใช้เทคโนโลยี IoT และระบบ Cloud IoT Service ในการพัฒนาอุตสาหกรรมของตนเองไปสู่อุตสาหกรรม 4.0 ซึ่งเป็นผลให้การขยายผลการดำเนินงานตามกรอบนโยบายของภาครัฐที่มุ่งพัฒนาให้อุตสาหกรรมไทยก้าวไปสู่อุตสาหกรรม 4.0 เป็นไปอย่างล่าช้า

นอกจากนี้ ยังมีแรงขับเคลื่อนด้านความต้องการคุ้มครองความเป็นส่วนตัวของข้อมูลจากทางภาครัฐกิจด้านการให้บริการด้านสุขภาพ เช่น ผู้ประกอบการด้านสถานพยาบาล ผู้ผลิตอุปกรณ์เซนเซอร์และให้บริการด้านสุขภาพ (Health Care Service) ที่มีการเก็บข้อมูลด้านสุขภาพของบุคคล เช่น เซนเซอร์ตรวจวัดคลื่นหัวใจ เซนเซอร์ตรวจวัดระดับน้ำตาลในผู้ป่วยโรคเบาหวาน หรือ การใช้เทคโนโลยี IoT ในห้องควบคุมความดันลบ (Negative Pressure Room) สำหรับรักษาผู้ป่วยติดเชื้อ COVID-19 ข้อมูลของผู้ป่วย และข้อมูลสำหรับอุปกรณ์ควบคุมในห้องผู้ป่วย ซึ่งข้อมูลเหล่านี้เป็นข้อมูลที่มีความอ่อนไหว (Sensitive Data) จำเป็นต้องได้รับการคุ้มครองความเป็นส่วนตัวของข้อมูล ในขณะที่การควบคุมอุปกรณ์ต่างๆ ภายในห้องผู้ป่วยจำเป็นต้องได้รับการประมวลผลเพื่อนำข้อมูลที่ได้มาวิเคราะห์ด้าน Medical Informatics โดยข้อมูลจากอุปกรณ์ IoT ต่างๆ เหล่านั้นจำเป็นต้องถูกส่งไปประมวลผลในคลาวด์ที่รองรับการประมวลผลข้อมูลขนาดใหญ่เพื่อประสิทธิภาพในการประมวลผล (เช่น การทำนายการเกิดโรค (Data Prediction), การแจ้งเตือนระดับน้ำตาลในผู้ป่วยโรคเบาหวาน), การควบคุมระดับออกซิเจนของผู้ป่วยตามอัตราการหายใจของผู้ป่วย เป็นต้น นอกจากนี้ ระบบ Health Care Service ยังจำเป็นต้องพึ่งพาการกระจายตัวของข้อมูลผ่านระบบ Cloud IoT Service และเพื่อให้การเข้าถึงข้อมูลที่ได้รับการประมวลผลแล้วเป็นไปได้อย่างสะดวก รวดเร็ว และไม่มีข้อจำกัดด้านเขตพื้นที่ใช้งาน อย่างไรก็ตาม ความกังวลด้านข้อมูลส่วนบุคคลด้านสุขภาพรั่วไหลไปสู่บุคคลภายนอกเป็นสาเหตุสำคัญที่ทำให้ผู้ใช้งานไม่มีความเชื่อมั่นในการขอรับบริการ และเป็นส่วนหนึ่งที่ทำให้การขยายตัวด้านธุรกิจการให้บริการด้านสุขภาพผ่านระบบ IoT Cloud Platform เป็นไปได้ช้า นอกจากนี้ ผู้ประกอบการเองยังมีความกังวลในการบังคับใช้กฎหมาย ตาม พ.ร.บ. คุ้มครองข้อมูลส่วนบุคคล พ.ศ. ๒๕๖๒ ซึ่งมีผลบังคับให้ผู้ประกอบการต้องรักษาความเป็นส่วนตัวของข้อมูลส่วนบุคคลของผู้ขอรับบริการ

ดังนั้น จึงมีความจำเป็นเร่งด่วนที่จะต้องมีการพัฒนาระบบ IoT และ Cloud IoT ให้สามารถรองรับการปกป้องความเป็นส่วนตัวของข้อมูลของอุปกรณ์ IoT และพัฒนาให้เกิดกลไกการทำงานร่วมกันระหว่างการคุ้มครองความเป็นส่วนตัวของข้อมูลและการประมวลผลในคลาวด์ได้อย่างมีประสิทธิภาพ เพื่อสร้างความเชื่อมั่นให้กับผู้ประกอบการอุตสาหกรรม และผู้ประกอบการด้านการให้บริการ Health Care ในการใช้ระบบ IoT และบริการ Cloud IoT

แม้ว่าในต่างประเทศเริ่มต้นตัวในการคิดค้นนวัตกรรมใหม่ที่สามารถให้ความคุ้มครองความเป็นส่วนตัวของข้อมูล เช่น บริษัท Intel ได้ผลิตอุปกรณ์ Server (Intel SGX) ที่มีสถาปัตยกรรมรองรับการประมวลผลข้อมูลที่คุ้มครองความเป็นส่วนตัวของข้อมูลภายใต้เทคโนโลยี (Trusted Execution Environments (TEE)) ซึ่งเทคโนโลยีนี้ได้ถูกนำมาเป็นทางเลือกให้กับผู้ใช้บริการคลาวด์ใน AWS และ Azure ที่ต้องการเลือกปกป้องข้อมูล โดยการให้บริการนั้นจำกัดอยู่ที่ฮาร์ดแวร์เฉพาะที่รองรับ Intel SGX เท่านั้น นอกจากนี้ยังมีบริษัท Startup ต่างๆ เช่น บริษัท Enveil, Inpher, และ TripleBlind ในประเทศสหรัฐอเมริกาได้นำเทคโนโลยีเข้ามาเพื่อให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลด้วยนวัตกรรมการเข้ารหัสแบบ Homomorphic Encryption (HE) ทั้งนี้ บริษัทดังกล่าวมีกลุ่มเป้าหมายลูกค้าระดับ Enterprise ที่ต้องการนำข้อมูลของบริษัทไปประมวลผลใน Cloud โดยการให้บริการนั้นจัดทำในรูปแบบของ Client/Server แต่รูปแบบการให้บริการนั้นยังจำกัดอยู่ที่ความต้องการฮาร์ดแวร์

จำเพาะที่รองรับ Intel SGX และระบบการเข้ารหัสด้วยเทคนิคการเข้ารหัสแบบ HE ที่ต้องอาศัยระบบประมวลผลขนาดใหญ่ (เช่น Server x86_64) ดังนั้น ด้วยเงื่อนไขดังกล่าวทำให้การให้บริการคุ้มครองความปลอดภัยข้อมูลที่มีอยู่ในท้องตลาดขณะนี้ยังไม่เหมาะสมในการนำมาใช้งานกับระบบ IoT ซึ่งมีข้อจำกัดด้านขนาดของทรัพยากรที่ใช้ประมวลผลในอุปกรณ์ขนาดเล็ก ด้วยเหตุนี้ จึงเป็นความท้าทายอย่างยิ่งที่จะพัฒนานวัตกรรมใหม่บนพื้นฐานของการเข้ารหัสข้อมูลแบบ Homomorphic Encryption เพื่อให้บริการความคุ้มครองข้อมูลส่วนบุคคลในระบบ IoT และทำให้ระบบที่พัฒนาขึ้นสามารถทำงานเข้ากันได้กับการให้บริการบน IoT Cloud Platform อีกทั้งยังเป็นโอกาสอันดีที่ประเทศไทยจะมีช่องทางการพัฒนาระบบให้บริการคุ้มครองความเป็นส่วนตัวข้อมูลสำหรับ IoT และสามารถทำงานร่วมกับ Cloud IoT ได้เพื่อเพิ่มขีดความสามารถในการแข่งขันด้านนวัตกรรมกับต่างประเทศ และพัฒนานวัตกรรมเพื่อให้บริการความคุ้มครองข้อมูลส่วนบุคคลในระบบ IoT นี้สอดคล้องกับ พ.ร.บ. คุ้มครองข้อมูลส่วนบุคคล พ.ศ. ๒๕๖๒

๑.๒ สารสำคัญของโครงการ

๑.๒.๑ วัตถุประสงค์

- ๑) เพื่อพัฒนาระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล (Privacy Preservation) ของอุปกรณ์ IoT ที่สามารถใช้งานได้กับระบบ IoT Cloud Platform เช่น NETPIE
- ๒) เพื่อสร้างความเชื่อมั่นให้กับภาคอุตสาหกรรมของไทยในการใช้งานระบบ IoT ที่ให้ความคุ้มครองความเป็นส่วนตัวและความปลอดภัยของข้อมูล IoT
- ๓) เพื่อสร้างองค์ความรู้และการพัฒนาต่อยอดระบบ IoT ของไทยให้มีศักยภาพทัดเทียมนานาประเทศ
- ๔) เพื่อสร้างความตระหนักรู้ และถ่ายทอดองค์ความรู้ในการใช้งานโมดูลการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลของอุปกรณ์ IoT ให้กับนักพัฒนา และผู้ใช้งาน IoT ภาคอุตสาหกรรมของไทย

๑.๒.๒ เป้าหมาย

- ๑) ผู้ใช้งานระบบ IoT ในภาคอุตสาหกรรมตระหนักถึงความปลอดภัยทางไซเบอร์และความเป็นส่วนตัวของข้อมูลของอุปกรณ์ IoT
- ๒) นักเรียน นักศึกษา และ นักพัฒนา IoT ตระหนักถึงการพัฒนาระบบ IoT ที่คำนึงถึงความปลอดภัยไซเบอร์และความเป็นส่วนตัวของข้อมูล
- ๓) สร้างองค์ความรู้ และการพัฒนาระบบ IoT ให้มีความปลอดภัยทางไซเบอร์และปกป้องความเป็นส่วนตัวของข้อมูลได้
- ๔) ลดความกังวล และส่งเสริมความเชื่อมั่นด้านความปลอดภัยทางไซเบอร์และปกป้องความเป็นส่วนตัวของข้อมูล IoT ในโรงงานให้กับผู้ประกอบการอุตสาหกรรมที่จะนำอุปกรณ์ IoT เข้ามาใช้พัฒนาระบบอุตสาหกรรมของตนเอง
- ๕) เพิ่มมูลค่าทางการตลาดสำหรับการให้บริการระบบ IoT cloud ที่รองรับกับกลุ่มผู้ใช้งาน IoT ที่มีความต้องการปกป้องความเป็นส่วนตัวของข้อมูล

๑.๒.๓ ผลผลิตสำคัญ

ที่	ชื่อผลผลิต	หน่วย วัด	ตัวชี้วัด (เชิงปริมาณ/เชิงคุณภาพ)
๑	ต้นแบบผลิตภัณฑ์ ระบบ Privacy as a Service for IoT	ระบบ	ต้นแบบภาคสนามระบบเพื่อคุ้มครองข้อมูลส่วนบุคคล (Privacy Preservation) จำนวน ๑ ระบบ
๒	องค์ความรู้รายงานผลการวิจัยระเบียบวิธีการ เข้ารหัสและถอดรหัสแบบ Homomorphic Encryption	เล่ม	รายงานสรุปผลการวิจัยวิธีการเข้ารหัสและถอดรหัส แบบ Homomorphic Encryption จำนวน ๑ เล่ม
๓	การใช้ประโยชน์เชิงสาธารณะ ๓.๑ การฝึกอบรมเชิงปฏิบัติการเพื่อให้ความรู้ เกี่ยวกับการติดตั้งและถ่ายทอดเทคโนโลยีการ ใช้งานระบบ Privacy as a Service ๓.๒ การถ่ายทอดเทคโนโลยีผ่านสื่อการ เรียนรู้ระบบ Privacy as a Service	ครั้ง ชุด	๓.๑ การฝึกอบรมเชิงปฏิบัติการเพื่อให้ผู้ใช้งานได้รับ ความรู้ความเข้าใจเกี่ยวกับเทคโนโลยีที่พัฒนาขึ้น จำนวน ๑ ครั้ง และมีผู้เข้าร่วมไม่ต่ำกว่า ๕๐ ราย ๓.๒ สื่อการเรียนรู้ระบบ Privacy as a Service จำนวน ๑ ชุด
๔	บทความทางวิชาการระดับนานาชาติ	เรื่อง	หนังสือแจ้งตอบรับวารสารวิชาการในระดับนานาชาติ จำนวน ๑ เรื่อง
๕	การเสนอผลงานในการประชุมวิชาการระดับ นานาชาติ	เรื่อง	หนังสือแจ้งตอบรับบทความในระดับนานาชาติ จำนวน ๑ เรื่อง

๑.๒.๔ ประโยชน์ที่คาดว่าจะได้รับ

๑) โรงงานอุตสาหกรรม: โดยทั่วไปแล้วโรงงานอุตสาหกรรมมีการติดตั้งระบบเซนเซอร์เพื่อบันทึกข้อมูลการผลิต ข้อมูลตรวจวัดค่าการใช้งานของเครื่องจักร และข้อมูลความเสียหายของเครื่องจักร หากข้อมูลต่างๆ เหล่านี้ถูกส่งออกไปประมวลผลในคลาวด์ อาจสร้างความกังวลให้กับผู้ประกอบการโรงงานอุตสาหกรรมเอง ในการเปิดเผยข้อมูลที่เป็นความลับทางธุรกิจต่อบุคคลภายนอก ดังนั้น หากมีการติดตั้งระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT ในโรงงานอุตสาหกรรมจะสามารถช่วยแก้ปัญหาดังนี้

- สามารถแก้ปัญหาข้อมูลการผลิต ข้อมูลตรวจวัดอุปกรณ์ ข้อมูลตรวจจับความเสียหายของอุปกรณ์การผลิตภายในโรงงานรั่วไหลไปสู่บุคคลภายนอก หรือผู้ให้บริการคลาวด์ได้
- สามารถนำข้อมูลภายในโรงงานอุตสาหกรรมไปประมวลผลในคลาวด์โดยข้อมูลที่ถูกประมวลผลไม่ถูกเปิดเผยต่อผู้ให้บริการคลาวด์
- สร้างความเชื่อมั่นให้กับผู้ประกอบการภาคอุตสาหกรรมในการเข้าใช้บริการ IoT Cloud Platform ที่สามารถคุ้มครองความเป็นส่วนตัวของข้อมูล

๒) IoT Cloud Platform: ผู้ให้บริการ Cloud IoT สร้าง Value Added ให้กับ Platform โดยมีบริการคุ้มครองความเป็นส่วนตัวของข้อมูลเป็นบริการทางเลือกให้กับกลุ่มลูกค้าที่ต้องการใช้บริการนี้ได้ ซึ่งจะส่งผลให้มีฐานลูกค้ารายใหม่เพิ่มขึ้น และเพื่อเพิ่มขีดความสามารถของ IoT Cloud Platform ของไทยให้ทัดเทียมกับ IoT Cloud Platform ของต่างประเทศ

๓) นักพัฒนาและผู้ผลิตอุปกรณ์ IoT : ช่วยเพิ่มมูลค่าให้กับอุปกรณ์ IoT เพื่อตอบโจทย์ความต้องการของลูกค้าที่มีความกังวลด้านการแลกเปลี่ยนข้อมูลด้านสุขภาพซึ่งเป็นข้อมูลส่วนบุคคลที่มีความอ่อนไหวกับระบบคลาวด์ที่ให้บริการ

บทที่ ๒

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

๒.๑ ทฤษฎี และแนวความคิด

ในส่วนนี้จะอธิบายถึงงานวิจัยในอดีต และโครงการวิจัยที่เกี่ยวข้องกับโครงการนี้ โดยมีเป้าประสงค์ที่จะชี้ให้เห็นถึงส่วนที่งานวิจัยในอดีตที่เกี่ยวข้องเหล่านั้นไม่สามารถตอบโจทย์การให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลได้ และเหตุผลที่ส่วนขาดหายไปของงานวิจัยดังกล่าวสามารถถูกเติมเต็มได้ด้วยทฤษฎีและหลักการที่โครงการวิจัยนี้นำเสนอขึ้นภายใต้โครงการนี้ โดยจะอธิบายถึงหลักการทำงานของ NETPIE ซึ่งเป็น IoT Cloud Platform ของไทยถูกพัฒนาขึ้นโดยทีมวิจัยและพัฒนาของเนคเทค ซึ่งโครงการวิจัยนี้นำมาเป็นต้นแบบในการพัฒนาต่อยอดในส่วนการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล จากนั้นได้อธิบายถึงเทคโนโลยีการคุ้มครองความปลอดภัยของข้อมูลใน IoT Cloud Platform ที่ใช้ทั่วไปในปัจจุบัน และจะชี้ให้เห็นถึงสาเหตุที่ทำให้บริการ IoT Cloud Platform เหล่านั้นไม่สามารถให้ความคุ้มครองความเป็นส่วนตัวของข้อมูลได้ จากนั้นในหัวข้อถัดมาได้อภิปรายถึงงานวิจัยและเทคโนโลยีที่ถูกนำเสนอเพื่อช่วยคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับการส่งข้อมูลออกไปประมวลผลบนคลาวด์ และสุดท้ายจะอธิบายถึงหลักการและทฤษฎีการเข้ารหัสแบบ Homomorphic Encryption (HE) ซึ่งเป็นทฤษฎีหลักที่นำมาใช้ในโครงการนี้

๒.๑.๑ IoT Cloud Platform

NETPIE เป็น IoT Cloud Platform ของไทยที่ให้บริการในรูปแบบ platform-as-a-service สำหรับอำนวยความสะดวกให้กับนักพัฒนา IoT สามารถพัฒนาให้อุปกรณ์เซนเซอร์ต่างๆ สามารถเชื่อมต่อ แลกเปลี่ยนข้อมูล และมีปฏิสัมพันธ์กันได้ในแบบ Internet of Things โดย NETPIE มีองค์ประกอบ ๒ ส่วนคือ Cloud Platform ที่ทำหน้าที่รองรับการเชื่อมต่อ IoT และการคุ้มครองการส่งผ่านข้อมูลและจัดเก็บข้อมูล IoT เหล่านั้นด้วยการยืนยันตัวตนผ่าน Access Token ของแต่ละอุปกรณ์ Edge Device ที่อยู่ภายใต้การจัดการของโปรเจ็ค ความปลอดภัย (Access Control Security) ของอุปกรณ์จะถูกพิสูจน์ตัวตนด้วย Device Token ในรูปแบบ JWT Token และ User Token ภายใต้ Account ที่ได้ลงทะเบียนไว้กับ NETPIE

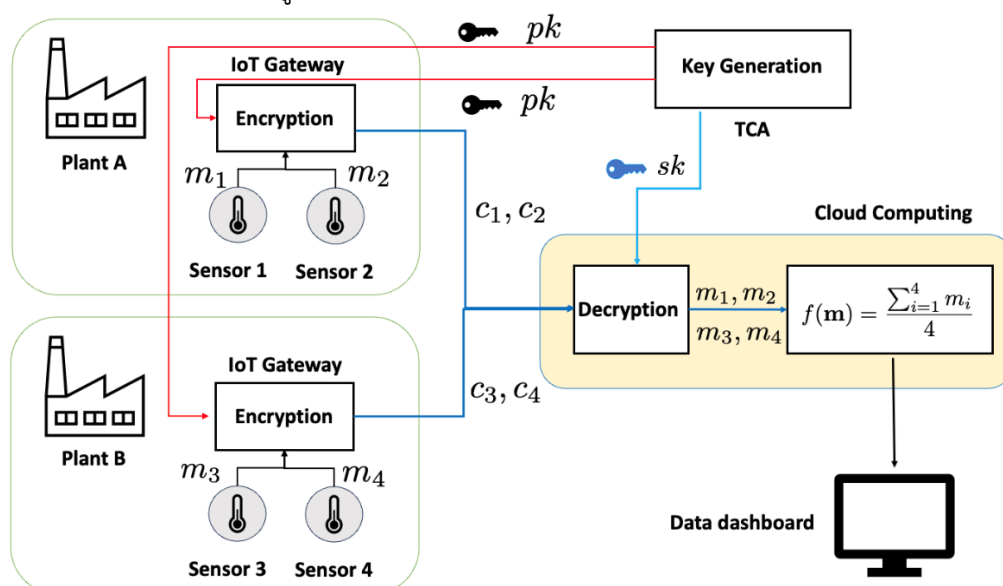
โดยอุปกรณ์ในโลกทางกายภาพจะสื่อสารและส่งข้อมูลมาประมวลผลที่คลาวด์ของ NETPIE โดยอาศัยโพรโทคอล MQTT ซึ่งข้อมูลที่สื่อสารผ่านโพรโทคอล MQTT จะไม่มีการเข้ารหัส ดังนั้นข้อมูลจะสามารถถูกดักฟังได้ หากต้องการสื่อสารในรูปแบบข้อมูลที่ถูกเข้ารหัสช่องสัญญาณ ผู้ใช้งานสามารถเลือกใช้ โพรโทคอล MQTTs ในการสื่อสารได้ ซึ่งรองรับการเชื่อมต่ออย่างปลอดภัยด้วยการสร้างเซสชัน SSL/TLS แต่อย่างไรก็ตาม ข้อมูลที่ถูกส่งผ่านโพรโทคอล MQTTs ข้อมูลก็จะถูกถอดรหัสช่องสัญญาณ และข้อมูลจะถูกจัดเก็บไว้ในดาต้าเบส (Device Shadow) บนคลาวด์ของ NETPIE ในรูปแบบที่ไม่เข้ารหัสข้อมูล และเมื่อข้อมูลถูกนำไปประมวลผล เช่น การสร้างส่วน การแจ้งเตือน การเปลี่ยนหน่วยของอุณหภูมิ เป็นต้น ข้อมูลที่ถูกนำมาคำนวณจะเป็นข้อมูลจริงที่ไม่ถูกเข้ารหัส และผลลัพธ์ที่ถูกคำนวณแล้วจะถูกเข้ารหัสช่องสัญญาณเพื่อส่งต่อไปยังอุปกรณ์แสดงผลปลายทาง

ถึงแม้ว่า NETPIE จะมีบริการให้ความปลอดภัยในการเข้าถึงข้อมูลบนคลาวด์ ด้วยรองรับการยืนยันตัวตน (Authentication) และกระบวนการจัดการสิทธิ (Authorization) สำหรับการเข้าใช้ของตัวอุปกรณ์นั้น แต่

NETPIE ยังไม่มีการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล ทำให้ข้อมูลที่ถูกส่งเข้ามาประมวลผลในคลาวด์ของ NETPIE ต้องถูกเปิดเผยข้อมูล ดังนั้น การให้บริการ Cloud IoT ของ NETPIE ในปัจจุบันอยู่ภายใต้สมมติฐานของความเชื่อใจในการปกปิดความเป็นส่วนตัวของข้อมูลของ IoT

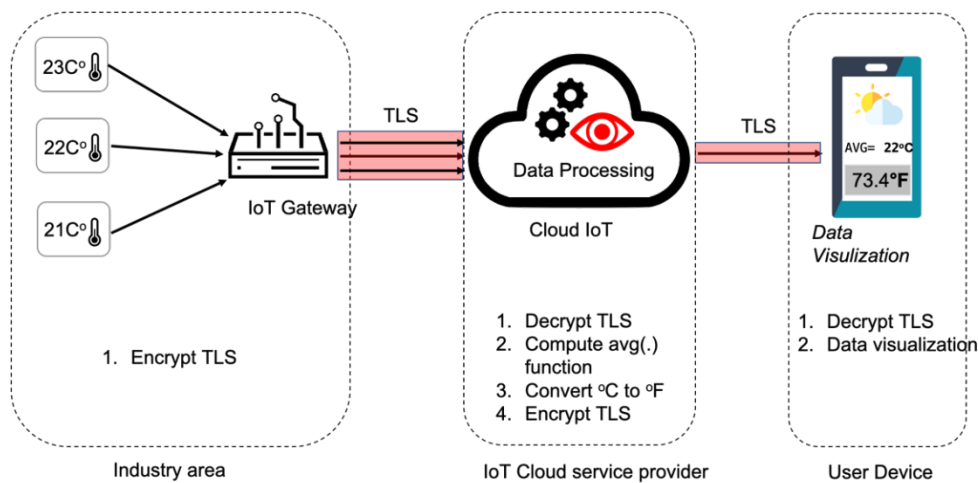
๒.๑.๒ เทคโนโลยีคุ้มครองความปลอดภัยของข้อมูลใน IoT Cloud Platform

ปัญหาความปลอดภัยของข้อมูลและการสูญเสียความเป็นส่วนตัวของข้อมูลสำหรับอุปกรณ์ IoT เกิดจากการประมวลผลข้อมูลอุปกรณ์ IoT ถูกส่งไปประมวลผลในคลาวด์เพื่อนำไปแสดงผล ใช้ในการควบคุมระบบ หรือจัดเก็บข้อมูลในคลาวด์ โดยข้อมูลที่ถูกประมวลผลต้องถูกเปิดเผยให้กับกับคลาวด์ ทั้งนี้ เทคนิคการส่งข้อมูลจากอุปกรณ์ IoT เข้าสู่คลาวด์เพื่อประมวลผลในคลาวด์ทั่วไปแล้วการส่งข้อมูล IoT Cloud Platform ได้รองรับการส่งข้อมูลผ่านข้อความผ่าน MQTT และ MQTT นั้นอยู่บนพื้นฐานการส่งข้อมูลด้วยโปรโตคอล TCP ทั้งนี้เพื่อความปลอดภัยของการส่งข้อมูล IoT Cloud Platform และ MQTT broker เช่น NETPIE, Google cloud IoT, IBM Watson IoT, AWS cloud IoT และ Azure IoT Hub เป็นต้น รองรับการส่งข้อมูลผ่าน SSL/TLS ระบบจะสร้างเซสชัน SSL/TLS ระหว่างอุปกรณ์ หรือ IoT Gateway กับ Cloud เพื่อการเชื่อมต่ออย่างปลอดภัย การส่งข้อมูลผ่าน TLS ทำให้มั่นใจได้ว่าการแลกเปลี่ยนข้อมูลระหว่าง Cloud และ IoT Gateway ปลอดภัยด้วยการแลกเปลี่ยน Certification และการตรวจสอบ Certification ของทั้งสองฝ่าย หลังจากนั้น IoT Gateway ส่ง Credential เพื่อให้ Server นำมาสร้าง Public Key (pk) และ Private Key ด้วยวิธีการเข้ารหัสแบบ Asymmetric (Asymmetrical Cryptography) และ Cloud Server จะส่ง pk ไปให้ IoT Gateway เพื่อใช้สำหรับเข้ารหัสข้อมูลก่อนส่งไปยังคลาวด์ จากนั้นอาจจะมีการทำข้อตกลงกันระหว่าง Cloud และ IoT Gateway เพื่อสร้าง Single Secure Session ด้วยวิธีการเข้ารหัสแบบ Symmetric Cryptography โดยจะมีการแลกเปลี่ยน Symmetric Key (sk) ระหว่าง Cloud และ IoT Gateway ซึ่งทำให้ข้อมูลที่ถูกส่งผ่านช่องทางนี้จะถูกเข้ารหัสด้วย sk แทนการเข้ารหัสด้วย pk ทั้งนี้ เนื่องจากการเข้ารหัสด้วย pk ทำให้ข้อมูลมีขนาดใหญ่ขึ้นมาก ในขณะที่การเข้ารหัสด้วย sk ให้ข้อมูลมีขนาดเล็กกว่าและขั้นตอนมีความซับซ้อนน้อยกว่าการเข้ารหัสแบบ pk



รูปที่ ๑ ไดอะแกรมแสดงการส่งข้อมูลจากอุปกรณ์ IoT เพื่อไปประมวลผลในคลาวด์

ตัวอย่างการส่งข้อมูลจากอุปกรณ์ IoT ไปประมวลผลใน Cloud โดยข้อมูลที่ถูกส่งเข้ารหัสดังรูปที่ ๑ ซึ่งพบว่า การประมวลผลข้อมูลจะทำได้ก็ต่อเมื่อต้องอนุญาตให้คลาวด์ถอดรหัสเพื่อทำการประมวลผลฟังก์ชัน $F(m)$ ทำให้ข้อมูลถูกเปิดเผยให้กับคลาวด์ เพื่อให้เข้าใจง่ายขึ้นในที่นี้จะขอยกตัวอย่างการส่งข้อมูลจากอุปกรณ์ IoT ในโรงงานอุตสาหกรรม Plant A และ Plant B เพื่อไปประมวลผลบนคลาวด์ และข้อมูลที่ได้จากการประมวลผลถูกส่งเข้าไปแสดงผลยังอุปกรณ์แสดงผลของผู้ใช้ปลายทาง รูปที่ ๒ แสดงถึงตัวอย่างการส่งข้อมูลจากอุปกรณ์ IoT ไปประมวลผลในคลาวด์แบบที่ใช้กันทั่วไป (ข้อมูลจะถูกเข้ารหัสเพื่อสร้างช่องสัญญาณปลอดภัย TLS และข้อมูลจะถูกส่งไปประมวลผลใน Cloud ซึ่งข้อมูลจะถูกถอดรหัสในคลาวด์และทำการคำนวณ และหลังจากนั้นผลการคำนวณที่ได้จะถูกเข้ารหัสอีกครั้งเพื่อส่งไปยังอุปกรณ์แสดงผลปลายทาง)



รูปที่ ๒ ตัวอย่างการส่งข้อมูลจากอุปกรณ์ IoT ไปประมวลผลในคลาวด์แบบที่ใช้กันทั่วไป

อย่างไรก็ตาม จะเห็นได้ว่าวิธีการส่งข้อมูลไปประมวลผลในคลาวด์ที่ใช้กันอยู่ทั่วไปนั้น ทำให้เกิดปัญหาการรั่วไหลของข้อมูลไปสู่ผู้ให้บริการคลาวด์ได้ ดังนั้นโครงการนี้ได้พิจารณาเทคนิคที่จะนำมาช่วยแก้ปัญหาการสูญเสียความเป็นส่วนตัวของข้อมูลอันเนื่องมาจากการประมวลผลข้อมูลในคลาวด์

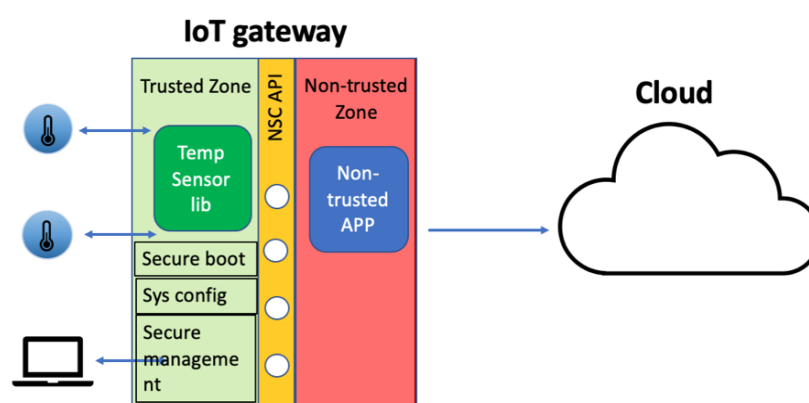
นักวิจัยต่าง ๆ ได้คิดค้นพัฒนาวิธีที่สามารถใช้ประโยชน์จากพลังการคำนวณมหาศาลของคลาวด์และยังคงไว้ซึ่งความเป็นส่วนตัวของข้อมูล ภายใต้สมมติฐานของผู้ให้บริการประมวลผลข้อมูลว่า ผู้ให้บริการคลาวด์ (Cloud Service Providers (CSPs)) เป็นผู้ที่เชื่อถือได้ครึ่งเดียว (Semi-trusted) นั่นหมายถึงผู้ให้บริการคลาวด์จะให้ผลลัพธ์การคำนวณที่ถูกต้องตามที่ตกลงกันไว้กับผู้ขอรับบริการ อย่างไรก็ตามผู้ให้บริการคลาวด์เองไม่สามารถเห็นและเรียนรู้ข้อมูลที่ถูกประมวลผลบนคลาวด์ได้

จากผลการสำรวจพบว่าเทคโนโลยีที่ช่วยคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับการส่งข้อมูลไปประมวลผลบนคลาวด์ (Privacy Enhancing Technologies (PTEs)) นั้นมีด้วยกันหลายวิธี ได้แก่ Trusted execution environment (TEE) [6], Pseudorandom Functions/Pseudorandom Permutations (PRFs/PRPs) [45], Secure Multi-party Computation (SMPC) [20], Perturbation Operations [57], Differential Privacy [M. Gong-2020], และ Homomorphic Encryption (HE) [39] เป็นต้น โดยมีรายละเอียด ดังนี้

๒.๑.๒.๑ การประมวลผลในสิ่งแวดล้อมที่เชื่อถือได้ (Trusted execution environment)

เทคนิคการประมวลผลในสิ่งแวดล้อมที่เชื่อถือได้ (TEE) [6] มีแนวคิดที่เกิดจากการออกแบบสถาปัตยกรรมคอมพิวเตอร์ใหม่เพื่อรองรับการคำนวณข้อมูลที่มีความอ่อนไหว (Sensitive Data) หรือข้อมูลที่ต้องการคุ้มครองความเป็นส่วนตัว (Privacy Preserving) โดยที่การออกแบบสถาปัตยกรรมคอมพิวเตอร์ใหม่นี้จะทำให้การประมวลผลข้อมูลบนพื้นที่ที่เชื่อถือได้ ไม่ขึ้นอยู่กับระบบปฏิบัติการ OS และ Hypervisor โดยฮาร์ดแวร์นี้ได้แก่ Arm TrustZone, Intel SGX, AMD Encryption, AMD Platform secure processor, x_86 system management mode, และ Intel management engine โดยเทคนิค Intel SGX และ Arm TrustZone เป็นเทคนิคที่นิยมใช้ในปัจจุบัน [6] ซึ่งอยู่บนพื้นฐานของการสร้างสิ่งแวดล้อมที่เชื่อถือได้เพื่อรองรับการประมวลผลข้อมูลที่ต้องการความคุ้มครองความปลอดภัย ในทางปฏิบัติ Arm TrustZone [9] ถูกนำมาใช้เพื่อพัฒนาเป็น IoT gateway ดังแสดงในรูปที่ ๓ ซึ่งสามารถป้องกันการโจมตีจาก Malware ได้ แต่การประยุกต์ใช้งาน Arm TrustZone ไม่ครอบคลุมการให้ความคุ้มครองความเป็นส่วนตัวในการประมวลผลในคลาวด์

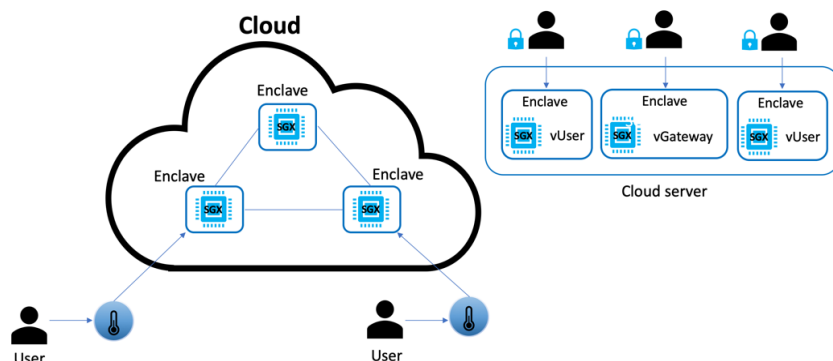
วิธี Software Guard Extensions (SGX) [30] เป็นวิธีที่อยู่ภายใต้เทคนิคของ Intel Processor ซึ่งรวมทั้งพื้นที่สำหรับการคำนวณที่ปลอดภัยทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์ ภายใต้พื้นที่ปลอดภัยนี้ถูกเรียกว่า Enclave พื้นที่ในส่วนนี้จะถูกปกป้องการเข้าถึงทั้งในรูปแบบ OS และ VM ที่แม้กระทั่ง Root ก็ไม่สามารถเข้าถึงได้ ข้อมูลที่ถูกส่งเข้าไปจะถูกถอดรหัสภายใน Enclave และคำนวณบนข้อมูลที่ไม่ถูกเข้ารหัสผลที่ได้จากการคำนวณจะถูกเข้ารหัสและส่งออกมาจาก Enclave เพื่อส่งไปยังผู้ใช้บริการ แต่อย่างไรก็ตามวิธีนี้ผู้ให้บริการคลาวด์จำเป็นต้องมี Hardware และ Software ที่รองรับ SGX (AWS เป็นตัวอย่างผู้ให้บริการคลาวด์หนึ่งที่มีเทคโนโลยี SGX ไว้ให้บริการ) นอกจากนี้ วิธีนี้ไม่ได้รับประกันความปลอดภัยของข้อมูลถ้าหาก OS ของ Server ที่ให้บริการถูกยึดจากผู้โจมตี



รูปที่ ๓ ตัวอย่างการนำ Arm TrustZone ไปใช้งานกับ IoT [9]

การนำ Intel SGX มาใช้ในระบบ IoT เพื่อการประมวลผลในคลาวด์ ดังแสดงในรูปที่ ๔ โดยผู้ให้บริการคลาวด์ต้องใช้ Server ที่มีระบบ Intel SGX โดยหลักการทำงานคือ การทำ Virtual User ในพื้นที่ที่เชื่อถือได้ภายใน Server หรือที่เรียกว่า Enclave โดยการประมวลผลข้อมูลจะทำได้ใน Enclave ของผู้ใช้แต่ละราย โดยผู้ให้บริการคลาวด์จะให้สิทธิในการเข้าใช้ของผู้ใช้แต่ละราย ดังนั้นข้อมูลที่ถูกเข้ารหัสที่ส่งมาประมวลผลใน Server

ภายใต้ Enclave จะถูกถอดรหัสและทำการประมวลผลในพื้นที่ที่เชื่อถือได้นี้ โดยผู้ให้บริการคลาวด์ หรือ บุคคลภายนอกไม่สามารถเข้าถึงข้อมูลจริงได้



รูปที่ ๔ ตัวอย่างการนำ Intel SGX ไปใช้งานกับ IoT [10]

อย่างไรก็ตามเทคนิคทางด้านฮาร์ดแวร์นี้ ไม่เหมาะสมในการนำมาใช้งานกับระบบ IoT ในโรงงานอุตสาหกรรม ที่ต้องการส่งข้อมูลไปประมวลผลในคลาวด์เพื่อลดภาระของ IoT Gateway/ Edge computing เนื่องจากการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลขึ้นอยู่กับฮาร์ดแวร์ (เช่น Intel SGX และ Arm TrustZone) ที่นำมาใช้เป็น IoT Gateway หรือ Server ที่ใช้ประมวลผลในคลาวด์ อีกทั้งยังเป็นการผูกขาดกับเทคโนโลยี ทำให้การขยายตัวของผู้ใช้งานอยู่ในวงจำกัด และเป็นการจำกัดการให้บริการคุ้มครองความเป็นส่วนตัวเฉพาะผู้ให้บริการคลาวด์ที่มีฮาร์ดแวร์นี้ให้บริการ

๒.๑.๒.๒ Differential Privacy และ Federated Learning

เทคนิคการป้องกันข้อมูลรั่วไหลโดยใช้การปนเปื้อนข้อมูลด้วยข้อมูลรบกวน (Noise) โดยการเลือกปริมาณข้อมูลรบกวนที่เหมาะสมจะช่วยให้ข้อมูลปลอดภัยและสามารถนำไปคำนวณต่อยังพื้นที่การคำนวณที่ไม่ปลอดภัยได้ โดยเทคนิค Differential Privacy จะทำการป้องกันข้อมูลของแต่ละบุคคลรั่วไหลจากการเพิ่มข้อมูลรบกวนที่คำนวณและปรับอย่างเหมาะสมกับชุดข้อมูลนั้นๆ เพื่อให้สามารถนำไปคำนวณได้อย่างถูกต้อง การเพิ่มข้อมูลรบกวนที่มากเกินไปจะส่งผลให้ข้อมูลไม่สามารถนำไปใช้งานต่อได้ซึ่งส่งผลให้ผลลัพธ์ที่ได้ไม่ถูกต้อง ในขณะที่เทคนิคนี้เหมาะกับการนำไปใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่สำหรับ Machine Learning ไม่เหมาะกับข้อมูลขนาดเล็กที่ต้องการคำนวณทั่วไป ส่วนเทคนิค Federated Learning มีข้อแตกต่างจากเทคนิค Differential Privacy เนื่องจากผู้ใช้งานเทคนิค Federated Learning เจ้าของข้อมูลจะเป็นผู้ประมวลผลข้อมูลเองโดยจะรับเฉพาะค่าพารามิเตอร์ของ Machine Learning Model ที่ถูกเทรนจากผู้ใช้งานอื่นมาอัปเดตโมเดลของตนเอง ทำให้ Federated Learning เหมาะกับการทำงานแบบ Distributed System ที่ผู้ใช้งานใช้ทรัพยากรของตนเองในการเทรนโมเดลแต่จะมีการแลกเปลี่ยนข้อมูลของพารามิเตอร์สำหรับการเทรนเพื่อให้ได้ Machine Learning Model ที่เสมือนได้รับการเทรนจากข้อมูลขนาดใหญ่จากหลายแหล่งข้อมูล ดังนั้นทั้งสองวิธีนี้ไม่เหมาะสมกับการนำมาใช้งานสำหรับการคำนวณค่า IoT ขนาดเล็กทั่วไป

๒.๑.๒.๓ Secure Multi-Party Computation (SMPC) และ Perturbation Method (PRM)

วิธี SMPC และ Perturbation Method (PRM) เป็นวิธีที่สามารถช่วยให้ข้อมูลที่ถูกส่งไปคำนวณภายนอกไม่สูญเสียความเป็นส่วนตัว แต่วิธีทั้งสองวิธีนี้ได้รับการประเมินจากนักวิจัยต่างๆ ว่าเป็นวิธีที่มีระดับความปลอดภัยทางไซเบอร์ต่ำ ซึ่งทำให้ผู้ไม่หวังดีถอดรหัสข้อมูลผลลัพธ์ได้ง่าย [69] โดยวิธีนี้จะรองรับความปลอดภัยของข้อมูลส่วนบุคคลได้ถ้ามีผู้ร่วมการคำนวณร่วมมือกันไม่เกิน 1/3 ของผู้ที่เข้าร่วมการคำนวณ [14]

๒.๑.๒.๔ Homomorphic Encryption (HE)

วิธี HE [38] เป็นวิธีที่ใช้เทคนิคของ Cryptographic ที่ได้พิสูจน์แล้วว่าเป็นวิธีที่ทำให้ข้อมูลที่เข้ารหัสมีความปลอดภัยทางไซเบอร์สูงและสามารถปกป้องความเป็นส่วนตัวของข้อมูลได้ [35] แต่อย่างไรก็ตามวิธี HE ในยุคแรกๆ ยังคงมีข้อจำกัดด้านประสิทธิภาพของการคำนวณในการใช้ทรัพยากรการคำนวณมากและประมวลผลช้า ซึ่งทำให้เกิดข้อสงสัยในการนำไปใช้งานในเชิงปฏิบัติ ในช่วงปี 2020 มาขึ้นการเติบโตของเทคโนโลยี IoT ในภาคอุตสาหกรรม (Industrial IoT) เพิ่มขึ้นอย่างมาก ทำให้ความจำเป็นในการพึ่งพาการประมวลผลบนคลาวด์มีความชัดเจนมากขึ้น อีกทั้งสืบเนื่องจากตัวอุปกรณ์ IoT มีหน่วยประมวลผลขนาดเล็กจึงทำให้ต้องอาศัยพลังการคำนวณจากทรัพยากรภายนอก เช่น คลาวด์ และภาคอุตสาหกรรมเองมีความกังวลเรื่องการรั่วไหลของข้อมูล IoT ไปสู่ภายนอก ทำให้นักวิจัยทั้งหลายหันมาให้ความสนใจพัฒนาเทคโนโลยีการเข้ารหัสแบบ HE ที่มีประสิทธิภาพมากยิ่งขึ้นและสามารถนำไปใช้งานกับอุปกรณ์ IoT ได้

เพื่อทำการศึกษาคูณสมบัติของเทคนิคที่เหมาะสมเพื่อนำมาพัฒนาต่อยอดในโครงการฯ จึงทำการเปรียบเทียบคุณสมบัติของเทคนิคที่ใช้ในการคุ้มครองความเป็นส่วนตัวของข้อมูลที่ถูส่งออกไปประมวลผลบนคลาวด์ต่างๆ ดังตารางที่ ๑

ตารางที่ ๑ การเปรียบเทียบเทคนิคการคุ้มครองความเป็นส่วนตัวของข้อมูลที่ถูส่งออกไปประมวลผลบนคลาวด์ [69]

เทคนิค	Cryptographic หรือ non-Cryptographic	Storage overhead	การ interaction ผู้ใช้งานหลายคน	จุดแข็ง	จุดอ่อน
SMPC	Cryptographic	สูง	ต้องการ	สามารถคำนวณอย่างปลอดภัยโดยอาศัยผู้ใช้งานหลายคน ไม่มี Third-party มาเกี่ยวข้อง	ต้องทำแบบออนไลน์ และใช้ทรัพยากรของผู้ใช้งาน
Pseudorandom Functions/Pseudorandom Permutations (PRFs/PRPs)	Cryptographic	ต่ำ	ไม่ต้องการ	ง่าย และมีประสิทธิภาพในการตรวจสอบ	มีข้อจำกัดในการใช้งานบนแอปพลิเคชัน
Differential Privacy	Cryptographic	สูง	ต้องการ	สามารถคำนวณบนฐานข้อมูลขนาดใหญ่	ต้องใช้ทรัพยากรของผู้ใช้งาน
SGX	Cryptographic	สูง	ไม่ต้องการ	มีพื้นที่การคำนวณที่ปลอดภัย และการ	มีช่องโหว่ในการโจมตีทั้งด้าน

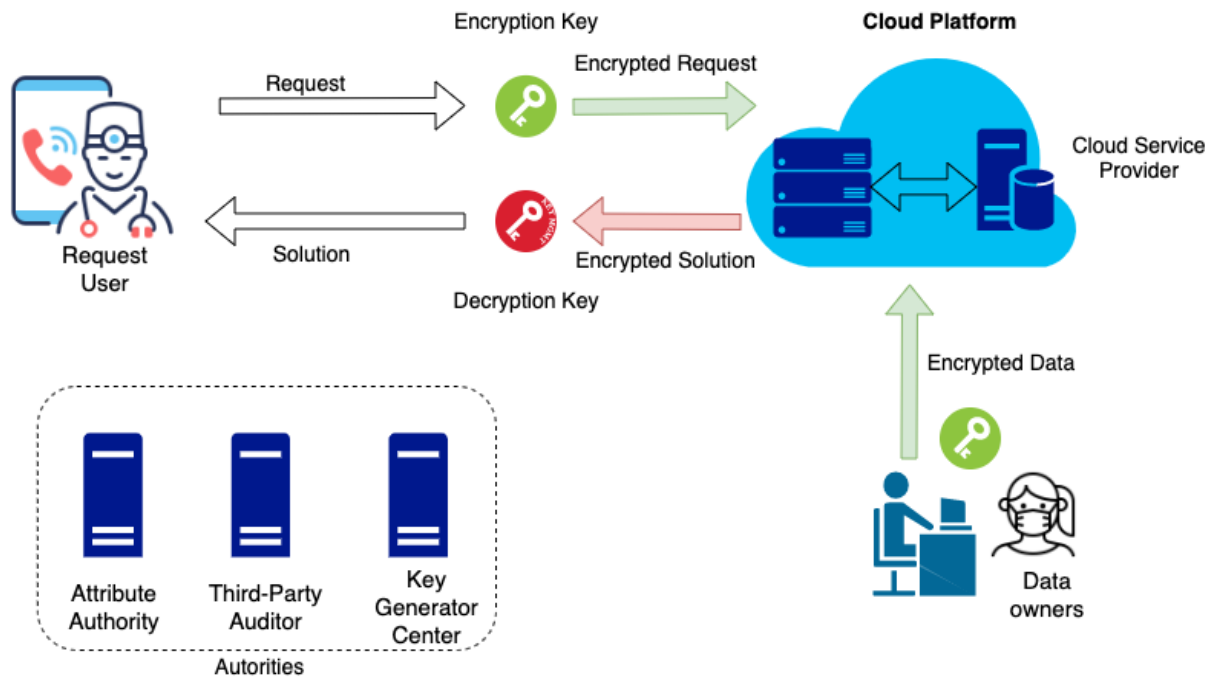
เทคนิค	Cryptographic หรือ non- Cryptographic	Storage overhead	การ interaction ผู้ใช้งานหลายคน	จุดแข็ง	จุดอ่อน
				คำนวณบนข้อมูลที่ไม่ เข้ารหัส	ซอฟต์แวร์และทาง กายภาพ
Perturbation Method (PRM)	Non- Cryptographic	ต่ำ	ไม่ต้องการ	ง่ายและมี ประสิทธิภาพในการ ตรวจสอบ	มีข้อจำกัดในการใช้ งาน และความ ปลอดภัยน้อย
HE	Cryptographic	สูง	ไม่ต้องการ	มีความปลอดภัยสูง และใช้งานได้ทั่วไป	มีความซับซ้อนสูง และมีขนาดข้อมูล ขนาดใหญ่ และ ประมวลผลช้า

๒.๒ ผลงานวิจัยที่เกี่ยวข้องกับเทคโนโลยีการเข้ารหัสแบบ HE

งานวิจัยที่เกี่ยวข้องกับการพัฒนาเทคโนโลยีการเข้ารหัสแบบ HE และการนำเทคโนโลยีนี้ไปประยุกต์ใช้งานกับแอปพลิเคชันอื่นๆ รวมทั้ง IoT ได้ถูกรวบรวมไว้ในส่วนนี้ โดยวัตถุประสงค์เพื่อต้องการวิเคราะห์ให้เห็นถึงแนวทางที่โครงการนี้เลือกใช้เทคโนโลยีการเข้ารหัสแบบ HE ที่เหมาะสมกับระบบ IoT ในภาคอุตสาหกรรม

เทคโนโลยีการเข้ารหัสแบบ HE คือ เทคโนโลยีการเข้ารหัสข้อมูลด้วยเทคนิค Cryptography ที่อยู่บนพื้นฐานการคำนวณแบบวงล้อกลุ่มทางคณิตศาสตร์ (Cyclic group) ที่อนุญาตให้ข้อมูลที่ถูกรหัสด้วยกุญแจสำหรับเข้ารหัส (Public key/symmetric key) แล้วเมื่อนำไปประมวลผลทางคณิตศาสตร์โดยไม่มีการถอดรหัสก่อนแล้วยังคงได้ผลลัพธ์ (Ciphertext) ที่เมื่อได้รับการถอดรหัสด้วยกุญแจถอดรหัสแล้วจะมีค่าเท่ากับผลลัพธ์ที่ได้จากการคำนวณด้วยข้อมูลที่ไม่ได้เข้ารหัส

โมเดลของระบบการส่งข้อมูลไปประมวลผลบนคลาวด์ โดยใช้เทคโนโลยีการเข้ารหัสแบบ HE สำหรับแอปพลิเคชันทางการแพทย์เป็นดังรูปที่ ๕ โดยภายใต้โมเดลนี้ต้องมีผู้ให้สิทธิ (Authority) รวมอยู่ด้วย โดยจะมีส่วนกลางเป็นผู้ทำหน้าที่สร้างกุญแจ จัดการกุญแจและการกำหนดสิทธิการเข้าถึงข้อมูล ขึ้นอยู่กับสมมติฐานในการออกแบบระบบ



รูปที่ ๕ โมเดลระบบการส่งข้อมูลไปประมวลผลบนคลาวด์ด้วยเทคโนโลยี HE แบบทั่วไป

สมมติฐานภัยคุกคามความปลอดภัยทางไซเบอร์สำหรับการออกแบบระบบความคุ้มครองความเป็นส่วนตัวและความปลอดภัยทางไซเบอร์มีอยู่ด้วยกัน ๒ ชนิด

๑) Honest-but-curious หรือ Semi-honest ภายใต้สมมติฐานนี้จะสมมติว่าผู้ให้บริการคลาวด์หรือผู้ประมวลผลมีความซื่อสัตย์ในการประมวลผลอย่างถูกต้องแต่ยังสงสัยหรือต้องการเรียนรู้ข้อมูลที่น่าสนใจมาประมวลผลและหาผลประโยชน์จากผลลัพธ์ของการประมวลผล

๒) Malicious ภายใต้สมมติฐานนี้ผู้ให้บริการคลาวด์หรือผู้ประมวลผลเชื่อถือไม่ได้ซึ่งให้ผลลัพธ์ของการประมวลผลที่ไม่ถูกต้องพร้อมกับต้องการหาผลประโยชน์จากข้อมูลที่ได้มาหรือเก็บผลคำนวณที่ถูกต้องไว้ใช้เอง

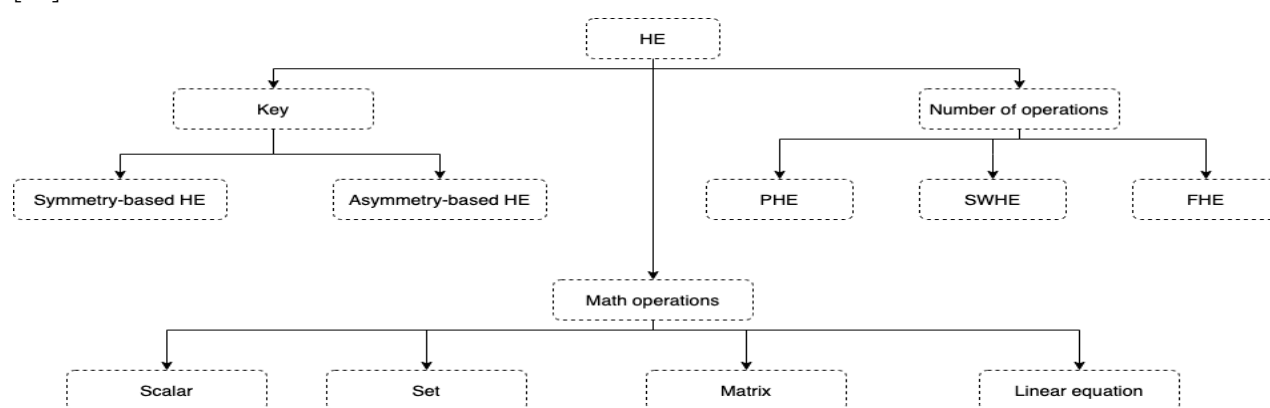
เงื่อนไขสำคัญที่ใช้สำหรับพิจารณาเลือกใช้การประมวลผลบนคลาวด์ต้องคำนึงถึงทั้งความปลอดภัยและประสิทธิภาพของการคำนวณ โดยประสิทธิภาพของการคำนวณหมายถึงค่าใช้จ่ายที่ใช้ในการคำนวณ ได้แก่ ปริมาณทรัพยากรการคำนวณที่ใช้ ระยะเวลาที่ใช้ในการคำนวณ

การประเมินความปลอดภัยในการคำนวณบนคลาวด์ต้องคำนึงถึงปัจจัยต่าง ๆ ดังนี้

- ๑) Data confidentiality หมายถึง ความสามารถในการป้องกันข้อมูลรั่วไหลไปสู่คลาวด์หรือผู้ที่ไม่ได้รับสิทธิ์ในการเข้าถึงข้อมูล
- ๒) Data integrity หมายถึง ความสามารถในการตรวจสอบผลลัพธ์ของการคำนวณที่ได้มีความถูกต้องครบถ้วน
- ๓) Data access controllability หมายถึง ความสามารถในการอนุญาตให้เฉพาะผู้ที่มีสิทธิ์เข้าถึงข้อมูลเข้าถึงข้อมูลจริงและผลลัพธ์ได้ พร้อมทั้งความสามารถในการเรียกสิทธิ์คืนได้ (Revoke)

๒.๒.๑ ประเภทของการเข้ารหัสแบบ HE

เทคนิคการเข้ารหัสแบบ HE สามารถแบ่งออกเป็นประเภทต่าง ๆ ได้ตามวิธีการจำแนกดังรูปที่ ๖ โดยการจำแนกแบ่งออกเป็น การจำแนกตามชนิดของ key ที่ใช้เข้ารหัสและถอดรหัส ซึ่งแบ่งได้เป็น ๒ ชนิด ได้แก่ Symmetric-key-based HE และ Asymmetric-key-based HE ซึ่งวิธี Symmetric-key-based HE เป็นวิธีที่มีความซับซ้อนน้อย แต่ไม่นิยมเนื่องจากมีระบบการจัดการ key ที่ยุ่งยาก, Overhead สูง, และมีความปลอดภัยต่ำ เนื่องจากการรั่วไหลของ key ได้ง่าย ทำให้ไม่นิยมนำมาใช้ได้จริงในทางปฏิบัติ ดังนั้นในปัจจุบัน นักวิจัยหันมาสนใจพัฒนาวิธีการเข้ารหัสแบบ Asymmetric-key-based HE ซึ่งมีความปลอดภัยในการจัดเก็บและแลกเปลี่ยนมากกว่าแบบแรก แต่อย่างไรก็ตาม ยังมีงานวิจัยบางส่วนนำเอาเทคนิคการเข้ารหัสทั่วไปมาช่วยเพิ่มความปลอดภัยในการแลกเปลี่ยนกุญแจสำหรับวิธี Symmetric-key-based HE เช่น เข้ารหัสกุญแจด้วยวิธี ASE และ KATAN [40] เป็นต้น และยังคงไว้ซึ่งความซับซ้อนน้อย

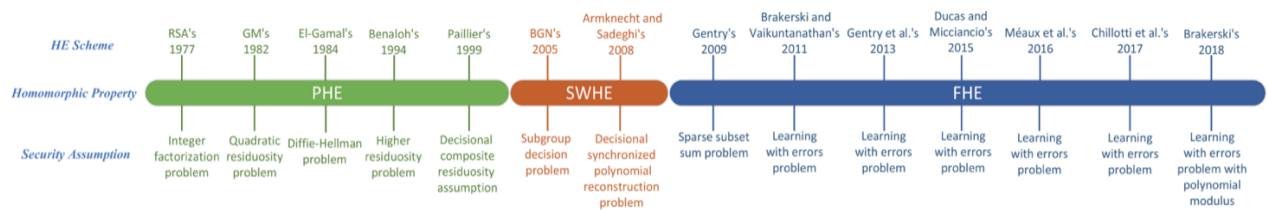


รูปที่ ๖ แผนผังการจำแนกประเภทของ HE

นอกจากนั้น HE ยังสามารถจำแนกได้ออกเป็น ๓ ประเภท ตามความสามารถในการทำโอเปอร์เรชันได้ ซึ่งได้แก่ Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE) และ Fully Homomorphic Encryption (FHE)

- ๑) PHE: เป็นวิธีการเข้ารหัสแบบ HE ที่รองรับการทำโอเปอร์เรชันเดียว เช่น โอเปอร์เรชันการบวก หรือ โอเปอร์เรชันการคูณ บน ciphertext ไม่จำกัดจำนวนครั้ง ดังนั้น เราสามารถจำแนกชนิดของ PHE ออกเป็น HE การบวก (Additive homomorphic encryption (AHE)) และการคูณ (Multiplicative homomorphic encryption (MHE)) โดยวิธี PHE นี้เป็นวิธีที่ได้รับความนิยมในการนำมาใช้งานในทางปฏิบัติเนื่องจากมีความซับซ้อนน้อยและใช้ทรัพยากรในการเก็บข้อมูลน้อย
- ๒) SWH: เป็นวิธีการเข้ารหัสแบบ HE ที่รองรับการทำโอเปอร์เรชันทั้งการบวกและการคูณบน Ciphertext แบบจำกัดจำนวนครั้ง เป็นวิธีที่มีข้อจำกัดในการใช้งาน แต่ยังคงเหมาะสมกับการใช้งานที่มีจำนวนครั้งในการคำนวณแบบจำกัด
- ๓) FHE: เป็นวิธีการเข้ารหัสแบบ HE ที่รองรับการทำโอเปอร์เรชันทั้งการบวกและการคูณบน Ciphertext แบบไม่จำกัดจำนวนครั้ง เป็นวิธีที่ยังไม่ได้รับความนิยมในการนำมาใช้งานจริงเนื่องจากความซับซ้อนของการคำนวณสูงและต้องใช้ทรัพยากรในการคำนวณมาก

การจำแนก HE ตามความสามารถในการทำโอเปอเรชันตามลำดับช่วงเวลาในการพัฒนาสามารถสรุปได้ดังรูปที่ ๗



รูปที่ ๗ แผนภาพวิวัฒนาการของการพัฒนาอัลกอริทึมการเข้ารหัสแบบ HE [69]

จากผลการศึกษาสามารถสรุปจุดอ่อนสำคัญของวิธี HE ได้ดังนี้

- ๑) มีค่าใช้จ่ายในการคำนวณสูง (High computation cost) : ถึงแม้ว่าวิธี PHE และ SWHE จะมีขั้นตอนการคำนวณที่น้อยกว่าเทคนิคแบบ FHE แต่ยังคงต้องอาศัยโมดูล Exponent arithmetic หลายโมดูลในการออกแบบโพรโทคอลที่ปลอดภัย สำหรับ FHE เป็นเทคนิคที่ต้องทำ Bootstrap ping เพื่อลด Noise ใน Ciphertext ซึ่งปริมาณ Noise ที่เกิดขึ้นในข้อมูลจะส่งผลเสียอย่างมากกับการถอดรหัสข้อมูลกลับคืนมาได้อย่างถูกต้อง
- ๒) ต้องการพื้นที่ในการเก็บข้อมูลขนาดใหญ่ (Large storage overhead) : เนื่องจากขนาดของ Ciphertext ที่ขยายออกหลายเท่าเมื่อเปรียบเทียบกับข้อมูล Plaintext ดังนั้นจึงจำเป็นต้องการพื้นที่ในการเก็บข้อมูลขนาดใหญ่
- ๓) ต้องการ Trusted Authority (TA) : ในกระบวนการเข้ารหัสจำเป็นต้องการคนกลางที่ไว้วางใจได้ในการสร้างและกระจาย Key ไปยังทุกส่วนที่เกี่ยวข้องในระบบ

๒.๒.๒ เทคนิคการเข้ารหัสแบบ HE

โดยทั่วไปเทคนิคการเข้ารหัสแบบ HE จะถูกนิยามภายใต้คุณสมบัติพื้นฐานของ Ring Number ซึ่งภายใต้ นิยามนี้การทำโอเปอเรชันบนจำนวนที่ไม่ถูกเข้ารหัส (ซึ่งโดยทั่วไปจะรองรับการเข้ารหัสจำนวนเต็มบวก

กำหนดให้ $E(\cdot)$ เป็นอัลกอริทึมสำหรับ Encryption และ $E(x)$ เป็นการเข้ารหัสข้อความ x นั่นคือ $E(x)$ คือข้อความ x ที่ถูกเข้ารหัส (Ciphertext) และกำหนดให้ M คือเซตของข้อความ Plaintext โดย นิยาม Θ_m และ Θ_c คือ โอเปอเรชันบน Plaintext และ Ciphertext ตามลำดับ
นิยาม : วิธีการเข้ารหัสจะถูกเรียกว่าการเข้ารหัสแบบ HE บนโอเปอเรชัน Θ_m ถ้าเป็นไปตามสมการต่อไปนี้

$$E(m_1)\Theta_c E(m_2) = E(m_1\Theta_m m_2), \forall m_1, m_2 \in M$$

ถ้ากำหนดให้ Θ_m เป็นโอเปอเรชันการบวกอัลกอริทึมจะเป็นไปตามคุณสมบัติการบวกกันแบบ Homomorphism (Additive homomorphism) ถ้ากำหนดให้ Θ_m เป็นโอเปอเรชันการคูณ และ ถ้ากำหนดให้ Θ_m เป็นโอเปอเรชันการคูณ อัลกอริทึมจะเป็นไปตามคุณสมบัติการคูณกันแบบ Homomorphism (Multiplication homomorphism)

วิธีการเข้ารหัสแบบ HE โดยทั่วไปประกอบด้วย 4 ขั้นตอน ดังนี้

- ๑) การสร้าง Key (Key Generation): การสร้าง key ด้วยฟังก์ชันการสร้าง key $Key(\lambda)$ โดยรับพารามิเตอร์อินพุต λ โดยอัลกอริทึมนี้จะสร้างกุญแจคู่ออกมา pk, sk โดยที่ pk คือ public key สำหรับการเข้ารหัส และ sk คือ secret key สำหรับการถอดรหัส
- ๒) การเข้ารหัส (Encryption): การเข้ารหัสทำได้ด้วยอัลกอริทึมการเข้ารหัส $Enc(pk, m)$ โดยรับพารามิเตอร์อินพุต pk และ ข้อความที่ต้องการเข้ารหัส m และเอาต์พุตที่ได้จะเป็นข้อความที่ถูกเข้ารหัส Ciphertext c ของข้อความ m
- ๓) การถอดรหัส (Decryption): การถอดรหัสสามารถทำได้ด้วยอัลกอริทึมการถอดรหัส $Dec(sk, c)$ โดยข้อความที่ถูกเข้ารหัส Ciphertext c สามารถถูกถอดรหัสได้ด้วยกุญแจ secret key sk
- ๔) การทำโอเปอเรชันบนข้อมูลที่ถูกเข้ารหัส: การทำโอเปอเรชันบน Ciphertext โดยใช้ $Eval(f(\cdot, \cdot), c_1, c_2)$ โดยรับข้อความเข้ารหัสที่ต้องการทำโอเปอเรชัน c_1, c_2 และ ฟังก์ชันสำหรับการทำโอเปอเรชัน และจะได้ผลลัพธ์เท่ากับ $f(c_1, c_2) = c$ ที่นำไปถอดรหัสได้

๒.๒.๓ ฟังก์ชันการทำโอเปอเรชันพื้นฐาน

ฟังก์ชันพื้นฐานสำหรับการทำโอเปอเรชันทางคณิตศาสตร์ ได้แก่

๑) Scalar โอเปอเรชัน

เป็นโอเปอเรชันทางคณิตศาสตร์ขั้นพื้นฐานที่พบได้ทั่วไป โดยสามารถแบ่งเป็น ๒ ชนิด ได้แก่ ๑. โอเปอเรชันพื้นฐาน (Basic operation) เช่น การบวก ลบ คูณ หาร เปรียบเทียบ การเรียงลำดับ และการหารากที่สอง เป็นต้น และ ๒. โอเปอเรชันผสม (Blend operation) เช่น การทำ Polynomial function ซึ่งต้องทำการคำนวณหลายๆ โอเปอเรชัน ตัวอย่างอัลกอริทึมการคำนวณ Scalar โอเปอเรชันได้แก่ Paillier ซึ่งมีอัลกอริทึมที่สามารถคำนวณทั้งในแบบ Positive integer และ Floating-point number นอกจากนี้วิธีการคำนวณแบบ Batch ช่วยทำให้การคำนวณเร็วขึ้น ทั้งนี้ สิ่งที่ทำนายของการนำอัลกอริทึม Paillier ไปประยุกต์ใช้ในการ outsource การคำนวณไปยังผู้ให้บริการคลาวด์รายเดียวทำให้สามารถถูกโจมตีได้จากปัญหาการโจมตีโดย Multi-tenancy attack ดังนั้น จึงทำให้นักวิจัยเสนอวิธีการกระจายกุญแจ (Distributed two trapdoor public-key cryptosystem (DT-PKC)) [63]

๒) Set โอเปอเรชัน

การทำ Set โอเปอเรชัน ได้แก่ การทำ Set intersection และ การทำ Set union ในการคำนวณนี้มีสมมติฐานว่าข้อมูลที่ผู้ร้องขอใช้งานต้องการรู้คำตอบของผลการทำ Intersection และ Union ของ Set แต่ไม่สามารถเข้าถึงข้อมูลของแต่ละ set ได้ ตัวอย่างวิธีการทำ Set โอเปอเรชัน ได้แก่ [46]

$$\text{กำหนดให้ } A = (a_1, a_2, \dots, a_n) \text{ และ } B = (b_1, b_2, \dots, b_n) \text{ ผลของการทำ}$$
$$|A \cap B| = |a_1, a_2, \dots, a_n| \cap (b_1, b_2, \dots, b_n) = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

และการใช้คุณสมบัติของเซต $|A \cap B| = |A| + |B| - |A \cup B|$

ทำให้คำนวณหาค่า Intersection และ Union ของ Set ได้จากวิธี Additive และ Multiplicative ได้

๓) Matrix โอเพอเรชั่น

การทำเมตริกซ์โอเพอเรชั่นระหว่าง ๒ เมตริกซ์ ได้แก่ ผู้ใช้งานมีเมตริก A และ B โดยต้องการคำนวณผลลัพธ์ของการคูณกันของทั้งสองเมตริกซ์ $C = AB$ โดยในงานวิจัย [D. Benjamin-2008] ได้ออกแบบโปรโตคอลสำหรับการคูณของเมตริกซ์ ตัวอย่างเช่น เมตริกซ์ X และ Y ขนาด $n \times n$

$$Enc(X \cdot Y)[i, j] = Enc \left(\sum_{k=1}^n X[i, k]Y[k, j] \right)$$

$$Enc(X \cdot Y)[i, j] = Enc \left(\sum_{k=1}^n X[i, k]Y[k, j] \right)$$

ภายหลังจากการคำนวณผลลัพธ์ ผู้ใช้งานสามารถตรวจสอบความถูกต้องของผลลัพธ์ที่ได้จากการคำนวณ โดยการสร้างเวกเตอร์ v สุ่มขนาด $n \times 1$ และตรวจสอบผลลัพธ์ได้โดย $x = Cv$ และตรวจสอบกับค่าผลลัพธ์ $y = A(Bv)$ ถ้าหากค่า $x \neq y$ จะหมายถึงผลการคำนวณไม่ถูกต้อง ต่อมาได้มีการพัฒนาให้วิธีการคำนวณเมตริกซ์เร็วขึ้นด้วยเทคนิคการ packing และการคำนวณผลคูณของหลายเมตริกซ์สามารถทำได้โดยใช้ BGV Cryptosystem และวิธีนี้ได้มีการนำมาใช้งานจริงภายใต้ HELib ไบบารี [28] และต่อมาได้มีการนำเทคนิค Chinese-Remainder Theorem (CRT) packing มาใช้เพื่อลดความซับซ้อนและ overhead ในการคำนวณ

๔) Linear Equation

การคำนวณแก้สมการเชิงเส้น (Linear Equation) ประกอบด้วย $Ax = b$ เมื่อ x เป็นเวกเตอร์ที่ต้องการหาค่า และ A คือ non-singular เมตริกซ์ และ b คือเวกเตอร์ที่มีค่าคงที่ ดังนั้น เมื่อ A เป็น non-singular เมตริกซ์ ขนาด $n \times n$ เราจะสามารถหาอินเวอร์สเมตริกซ์ A^{-1} ได้ ดังนั้นเราสามารถคำนวณค่า $x = A^{-1}b$ ในรูปแบบของการเข้ารหัสได้จาก

$$Enc(A^{-1})Enc(b) = Enc(A^{-1}b) = Enc(x)$$

ในงานวิจัยของ Kiltz et al [23] ได้นำเสนอวิธี PHE เพื่อแก้สมการเชิงเส้น ต่อมา [C. Wang -2011] ได้เสนอวิธี AHE โดยใช้เทคนิคการคำนวณแบบวนซ้ำ (Iterative) เพื่อลดความซับซ้อนของการหาคำตอบซึ่งทำให้มีค่าความซับซ้อนอยู่ที่ $O(n)$ และวิธีนี้ยังทำให้ผู้ใช้งานสามารถตรวจสอบความถูกต้องได้

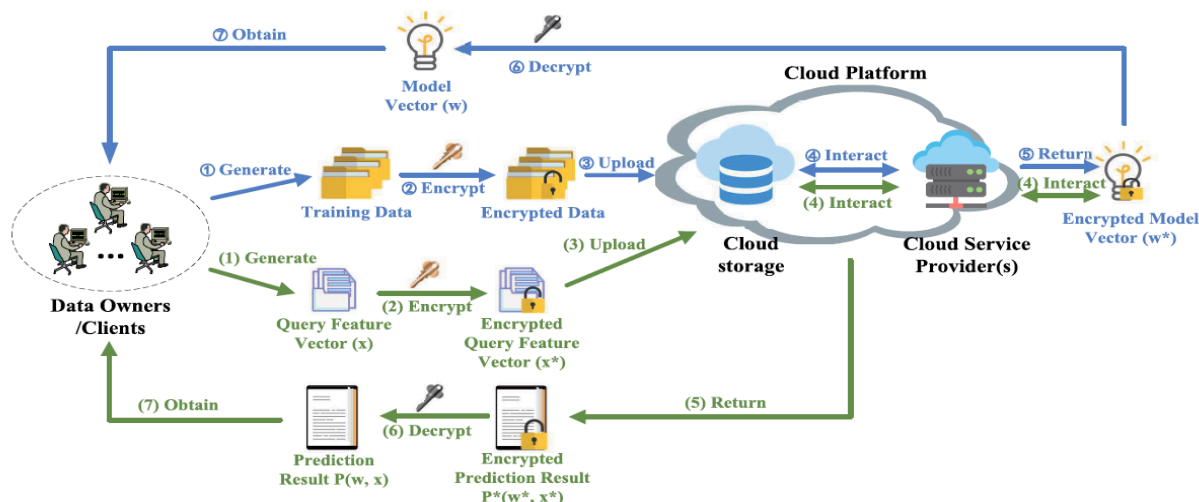
๒.๒.๔ แนวทางการนำไปประยุกต์ใช้งานเทคนิคการเข้ารหัสแบบ HE

๑) การประยุกต์ใช้งาน HE กับ Machine Learning และ Data Mining

เทคนิคการคุ้มครองข้อมูลส่วนบุคคลแบบ HE ถูกนำมาประยุกต์ใช้งานกับงานด้าน Machine Learning และ Data Mining สามารถสรุปได้ดังตารางที่ ๒

ตารางที่ ๒ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ Machine Learning และ Data Mining

งาน		อ้างอิง	เทคนิคที่ใช้	สมมติฐานความปลอดภัย	ตรวจสอบได้หรือไม่
Regression analysis	Linear / ridge regression	[67]	PHE + DP	Semi-honest	ไม่ได้
		[58]	SWHE	Semi-honest	ไม่ได้
		[60]	Linearly HE + GCs	Malicious	ได้
	Logistic regression	[66]	PHE	Semi-honest	ไม่ได้
	Logistic regression	[17]	SWHE	Semi-honest	ไม่ได้
	Logistic regression	[33]	Approximate HE	Semi-honest	ไม่ได้
Classification algorithms	Support vector machine	[62]	PHE	Semi-honest	ไม่ได้
	Naïve Bayesian, Decision trees	[15]	FHE	Semi-honest	ไม่ได้
ANN		[25]	PHE	Semi-honest	ไม่ได้
Data mining	K-means clustering	[21]	FHE	Semi-honest	ไม่ได้
	Association rule mining	[24]	PHE	Semi-honest	ไม่ได้



รูปที่ ๘ กระบวนการให้บริการการประมวลผล Machine learning และ Data mining บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล

ในกระบวนการส่งข้อมูลเข้าไปประมวลผล Machine Learning และ Data Mining บนคลาวด์อย่างปลอดภัยและคุ้มครองข้อมูลส่วนบุคคลสามารถสรุปได้ดังรูปที่ ๘ โดยในกระบวนการแบ่งออกเป็นสองส่วน คือ ๑) ขั้นตอนการเรียนรู้ (Learning phase) โดยขั้นตอนนี้เริ่มต้นจากเจ้าของข้อมูล (Data owner) ส่งข้อมูล Training Data เข้าไปผ่านกระบวนการเข้ารหัส จากนั้นข้อมูลที่เข้ารหัสแล้วจะถูกส่งขึ้นไปเก็บไว้ที่ Cloud storage และจากนั้น ภายใน cloud จะมีการแลกเปลี่ยนข้อมูลระหว่าง Cloud storage และ หน่วยให้บริการประมวลผลของคลาวด์ หลังจากเสร็จสิ้นการ Train จะได้ Model ที่อยู่ในรูปแบบของการเข้ารหัส และเมื่อ Model ที่ถูกเข้ารหัสถูกส่งมาถึงผู้ใช้งาน หรือเจ้าของข้อมูล ผู้ใช้งานจะมีกุญแจสำหรับถอดรหัส ๒) ขั้นตอนการทำนาย (Prediction phase) เจ้าของจะทำการส่ง model ที่เข้ารหัสหรือ Query feature ที่ใช้ในการทำนายแบบเข้ารหัสไปยังคลาวด์ และจากนั้น คลาวด์จะส่งผลการทำนายในรูปแบบการเข้ารหัสมายังผู้ใช้งาน และผู้ใช้งานที่มีกุญแจสำหรับถอดรหัสจะสามารถถอดรหัสเพื่อแสดงผลลัพธ์จากการทำนายได้

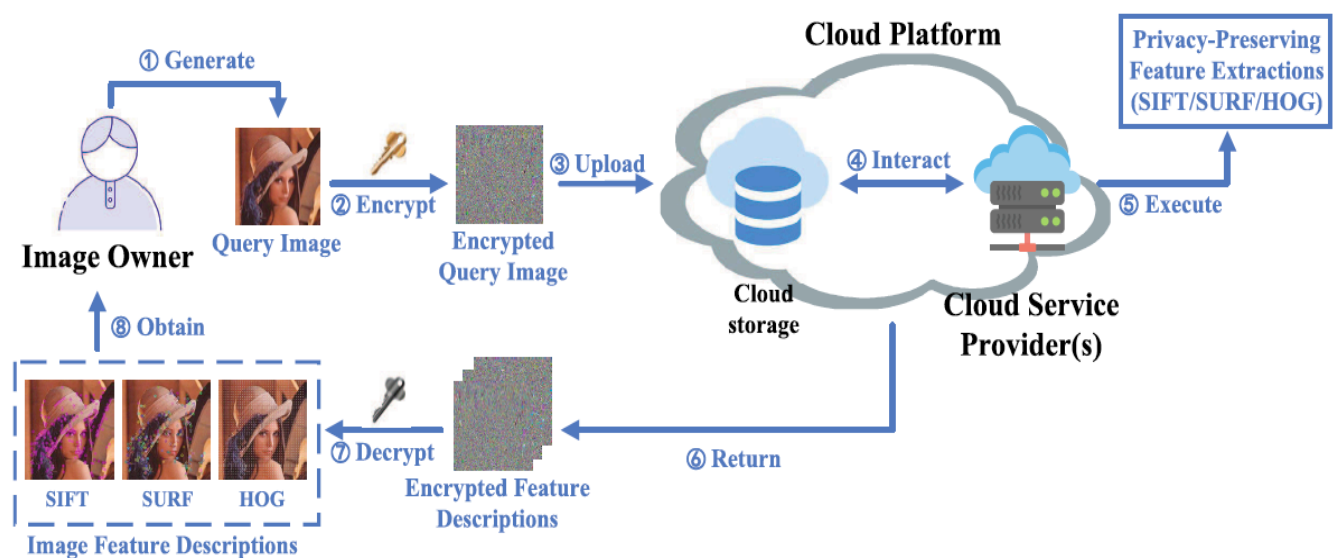
๒) การประยุกต์ใช้งานกับ Image Processing

เนื่องจากงานด้านการประมวลผลภาพ (Image Processing) มีการเติบโตอย่างมากซึ่งโดยมากจะใช้วิธีการประมวลผลบนคลาวด์ แต่เนื่องจากภาพที่ต้องการนำไปใช้งานด้านการประมวลผลภาพส่วนใหญ่มีความอ่อนไหวและต้องการความคุ้มครองข้อมูล ดังนั้นจึงมีงานวิจัยจำนวนมากพัฒนาเทคนิคการเข้ารหัสแบบ HE เพื่อนำมาใช้กับงานประมวลผลภาพ โดยตัวอย่างการประยุกต์ใช้งานเป็นดังตารางที่ ๓

ตารางที่ ๓ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ Image Processing

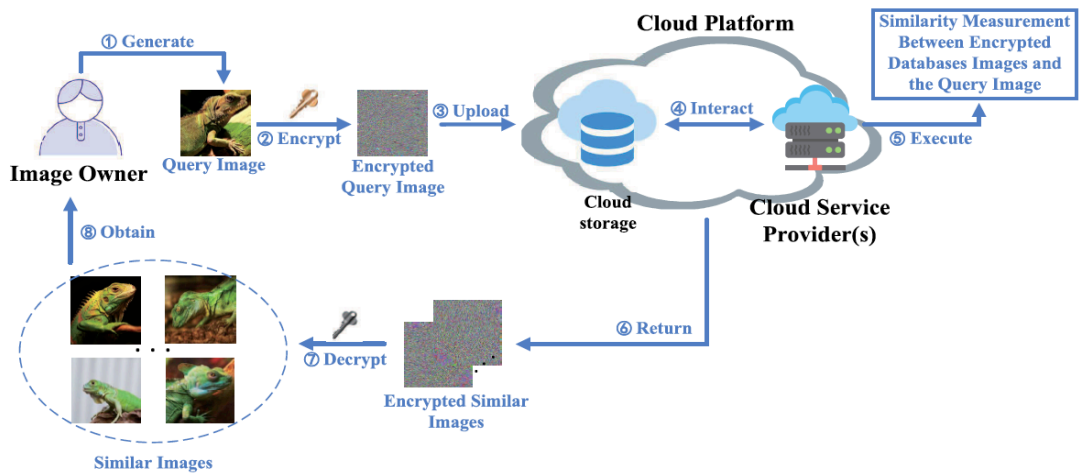
งาน		อ้างอิง	เทคนิคที่ใช้	คุณลักษณะของอัลกอริทึม
Image feature extraction	SIFT (Scale-invariant feature transform)	[22]	PHE	ปกป้องข้อมูล Location ของ image keypoints
	SURF (Speeded-up	[68]	PHE	Execute interest points

งาน		อ้างอิง	เทคนิคที่ใช้	คุณลักษณะของอัลกอริทึม
	robust features)			extraction, accurate keypoint localization
	HOG (Histogram of oriented gradients)	[59]	FHE	Higher accuracy
Content-based image retrieval		[68]	PHE	Image similarity measurement
Image watermarking		[16]	PHE	DWT method

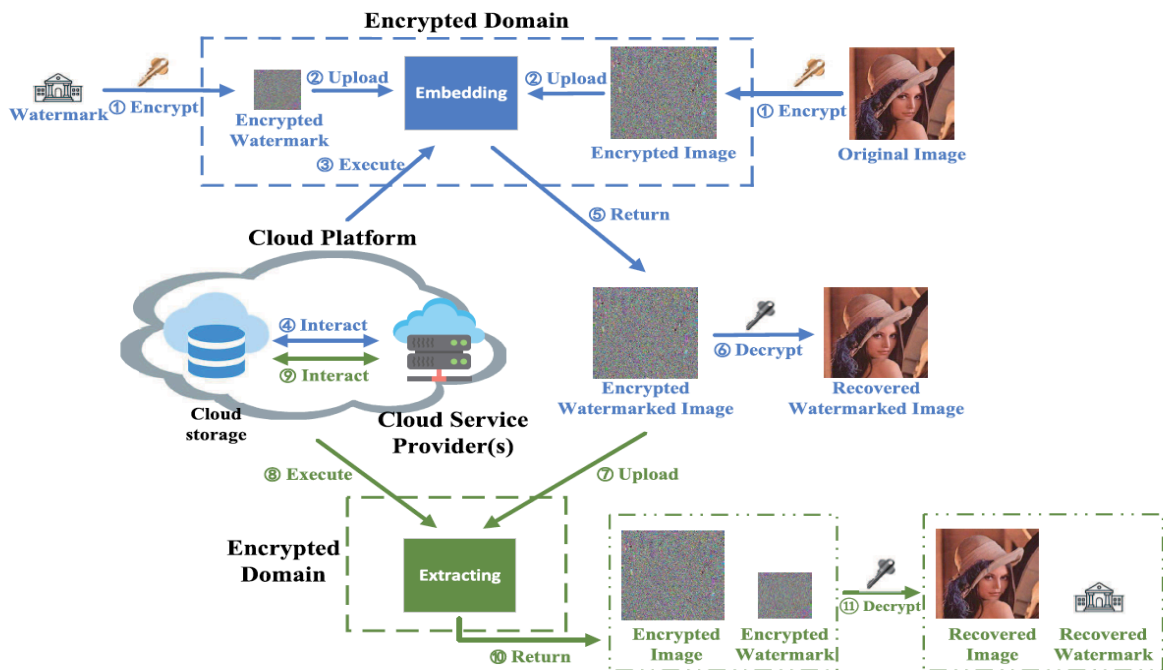


รูปที่ ๙ กระบวนการให้บริการการประมวลผลการดึงข้อมูล Image feature extraction บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล

ในกระบวนการประมวลผลการดึงข้อมูล feature extraction จากภาพที่ส่งไปประมวลผลบนคลาวด์เป็นดังรูปที่ ๙ โดยภาพของเจ้าของภาพจะถูกเข้ารหัสก่อนและส่งขึ้นไปเก็บไว้ในฐานเก็บข้อมูลในคลาวด์เพื่อรอนำไปประมวลผล โดยคลาวด์จะทำการดึง Feature ของภาพด้วยวิธี SIFT, SURF หรือ HOG ในรูปแบบข้อมูลที่ถูกรหัส ผลลัพธ์จะเป็น Feature ของภาพที่ถูกเข้ารหัส และจะถูกส่งมาถอดรหัสที่ผู้ร้องขอใช้งาน หรือเจ้าของข้อมูล ในทำนองเดียวกันกระบวนการประมวลผล Image retrieval และ Image watermarking เป็นดังรูปที่ ๑๐ และรูปที่ ๑๑ ตามลำดับ



รูปที่ ๑๐ กระบวนการให้บริการการประมวลผลการ Image retrieval
บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล



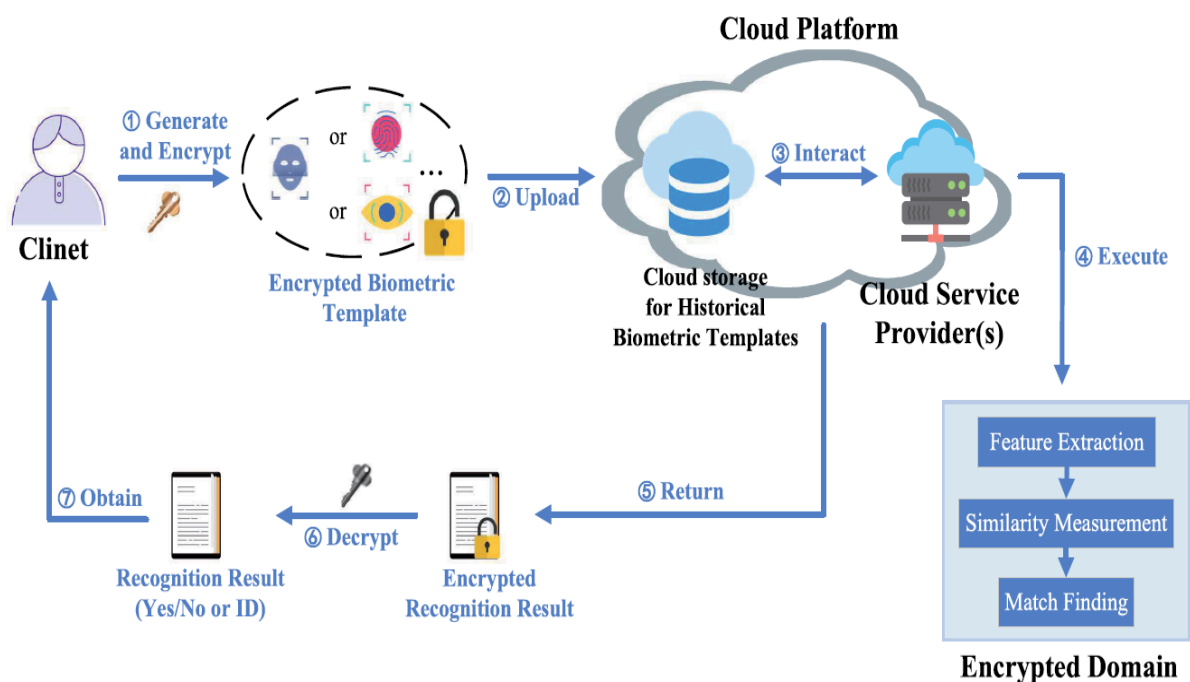
รูปที่ ๑๑ กระบวนการให้บริการการประมวลผลการ Image watermarking
บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล

๓) การประยุกต์ใช้งานกับ Biometric authentication

งานที่ถูกหยิบยกขึ้นมาเป็นประเด็นในการให้ความสำคัญกับการคุ้มครองข้อมูลส่วนบุคคลที่กล่าวถึงเป็น
อย่างมาในปัจจุบันคือ งานด้าน Biometric authentication หรือการให้คลาวด์เรียนรู้จำใบหน้า หรือข้อมูลด้านชี
วิภาพของมนุษย์เพื่อใช้ในการระบุตัวตน ดังนั้นจึงมีงานวิจัยมากมายที่นำเทคนิคการเข้ารหัสแบบ HE มาช่วยใน
การคุ้มครองความเป็นส่วนตัวของข้อมูลดังตัวอย่างในตารางที่ ๔

ตารางที่ ๔ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ Biometric authentication

งาน		อ้างอิง	เทคนิคที่ใช้	สมมติฐาน ความปลอดภัย
Secure genome analysis	Genome-wide association study	[71]	FHE	Semi-honest
	Whole genome sequencing	[35]	PHE + FHE	Semi-honest
	Edit distance or Hamming distance of genetic data	[33]	SWHE	Semi-honest
	Pattern matching on genetic data	[43]	SWHE	Semi-honest
Biometric	Biometric authentication	[41]	PHE	Semi-honest
	Face recognition	[61]	FHE	Semi-honest
	Fingerprint / iris recognition	[37]	PHE	Semi-honest



รูปที่ ๑๒ กระบวนการให้บริการการประมวลผลการ Biometric Authentication

บนคลาวด์อย่างปลอดภัยและป้องกัน การรั่วไหลของข้อมูลส่วนบุคคล

กระบวนการนำข้อมูล Biometric ไปประมวลผลบนคลาวด์เพื่อใช้ระบุตัวตนอย่างปลอดภัยสามารถสรุปได้ดังรูปที่ ๑๒ โดยเจ้าของข้อมูลจะเป็นผู้สร้างกุญแจสำหรับการเข้ารหัสและ เข้ารหัสข้อมูล Biometric หลังจากนั้นข้อมูลที่ถูกเข้ารหัสแล้วจะถูกส่งไปยังคลาวด์เพื่อนำไปจัดเก็บและประมวลผลเพื่อทำการจับคู่กับข้อมูลในฐานข้อมูลเพื่อระบุตัวตน ผลการระบุตัวตนจะถูกส่งกลับมายังผู้ร้องขอบริการและถอดรหัส ตัวอย่างที่มีให้เห็นใช้

งานทั่วไปคือ การใช้ใบหน้าระบุการเข้าถึงการใช้งานของมือถือสมาร์ทโฟน การเข้าถึงการใช้งานแอปพลิเคชันต่างๆ เป็นต้น

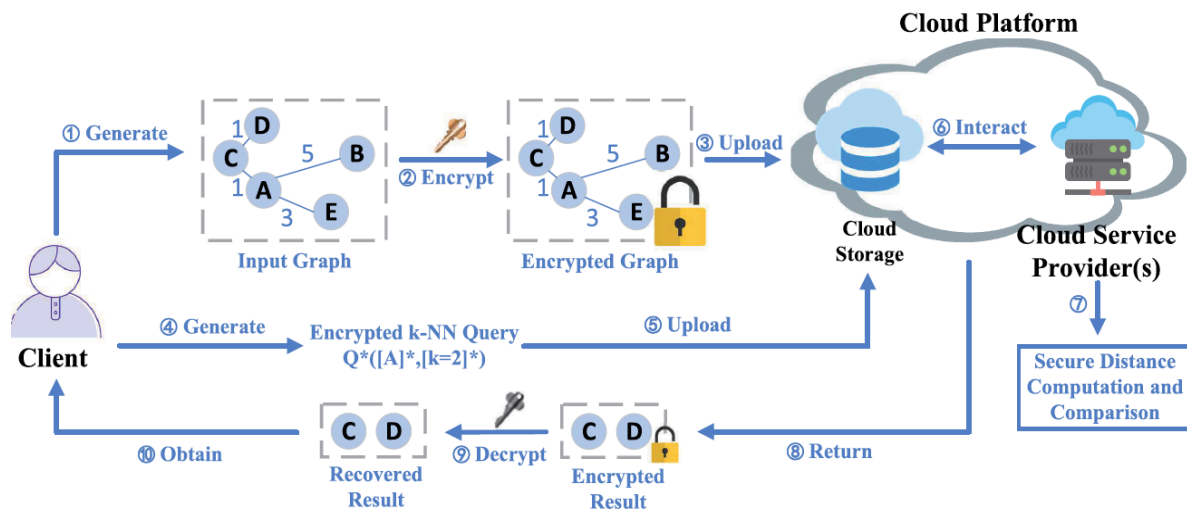
๔) การประยุกต์ใช้งานกับ Location-based service

นอกจากงานด้านการระบุตัวตนแล้วการใช้งานแอปพลิเคชันในการระบุสถานที่ของผู้ใช้งาน และสิ่งของต่างๆ ก็มีใช้อย่างแพร่หลายมากขึ้น เช่น การใช้ Google Maps เพื่อระบุเส้นทาง การใช้แอปพลิเคชันในการรับส่งสินค้าและบริการต่างๆ การระบุตำแหน่งของอุปกรณ์ เช่น Airtag เป็นต้น ข้อมูลตำแหน่งที่อยู่ของบุคคลและสิ่งต่างๆ อาจจะถือว่าเป็นข้อมูลส่วนบุคคล ดังนั้น จึงมีงานวิจัยที่นำเอาวิธีการเข้ารหัสด้วยเทคนิค HE มาประยุกต์ใช้ในการระบุตำแหน่งเพื่อป้องกันไม่ให้ คลาวด์ใช้ประโยชน์จากข้อมูลเหล่านี้ได้ ตัวอย่างของการประยุกต์ใช้เป็ดังตารางที่ ๕

ตารางที่ ๕ ตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับ Location-based service

งาน		อ้างอิง	เทคนิคที่ใช้	สมมติฐาน ความปลอดภัย
K-nearest neighbor		[32]	PHE	Semi-honest
		[65]	PHE	Semi-honest
Shortest path query	Approximate shortest distance	[64]	SE + SWHE + PRFs	Semi-honest
	Exact shortest distance	[51]	PHE + GCs + PRFs+ OT	Semi-honest
	Obstructed shortest distance	[36]	PHE + MPC + PT	Semi-honest
	Constrained shortest distance	[42]	ORE + SWHE + PRFs	Semi-honest
Group location services		[34]	SWHE + PBM	Semi-honest

การระบุตำแหน่งของวัตถุสามารถใช้เทคนิคของ k-NN โดยในรูปที่ ๑๓ แสดงให้เห็นกระบวนการประมวลผล k-NN บนคลาวด์อย่างปลอดภัย โดยข้อมูลของกราฟเพื่อใช้ในการระบุตำแหน่งถูกเข้ารหัสก่อนนำไปประมวลผลเพื่อหาระยะทางในคลาวด์ และผลการคำนวณจะถูกนำไปถอดรหัสที่ผู้ใช้งาน



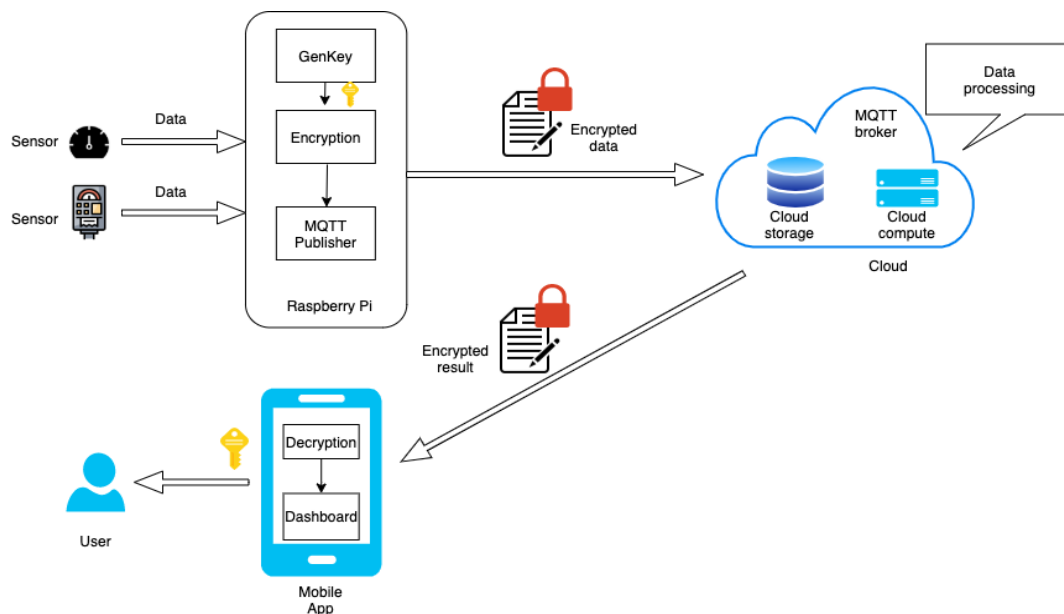
รูปที่ ๑๓ กระบวนการให้บริการการประมวลผลการสร้างโมเดล k-NN
บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล

๕) การประยุกต์ใช้งานกับ IoT

เนื่องจากปัจจุบันมีความต้องการนำอุปกรณ์ IoT เข้ามาเป็นส่วนหนึ่งของในชีวิตประจำวันเพื่อช่วยให้เกิดความสะดวกสบาย Smart life, Smart city และ Smart factory และข้อมูลที่ได้จากอุปกรณ์ IoT ที่นำไปประมวลผลส่วนใหญ่มีความอ่อนไหวไม่ว่าจะเป็นข้อมูลส่วนบุคคลของผู้ใช้งาน ข้อมูลด้านการผลิตและพลังงานในโรงงานอุตสาหกรรม ดังนั้น จึงมีงานวิจัยเริ่มให้ความสนใจในการนำเอาเทคนิค HE เข้ามาช่วยให้ความปลอดภัยและความเป็นส่วนตัวของข้อมูลในขณะเดียวกัน การนำเทคนิค HE มาใช้กับ IoT มีความท้าทายเป็นอย่างมาก เนื่องจากอุปกรณ์ IoT เป็นอุปกรณ์ที่มีหน่วยประมวลผลจำกัด มีพื้นที่ในการจัดเก็บข้อมูลน้อย ดังนั้นงานวิจัยส่วนใหญ่จะมุ่งเน้นไปที่การพัฒนาเทคนิคการเข้ารหัสที่ปลอดภัยและใช้ทรัพยากรในการคำนวณน้อย ตารางที่ ๖ แสดงตัวอย่างการประยุกต์ใช้งานเทคนิค HE กับอุปกรณ์ IoT

ตารางที่ ๖ การประยุกต์ใช้งานเทคนิค HE กับ IoT devices

งาน	อ้างอิง	เทคนิคที่ใช้	สมมติฐานความปลอดภัย	ตรวจสอบได้หรือไม่
IoT based green house	[56]	PHE	Semi-honest	ไม่ได้
Data aggregation for fog computing for hybrid IoT devices	[52]	PHE + CR + Hashing	Semi-honest	ได้
MQTT message บน IoT devices	[13]	SWHE	Semi-honest	ไม่ได้
Speed up data encryption for IoT devices	[40]	ASE + FHE, KATAN + FHE	Semi-honest	ไม่ได้
IoT devices	[31]	Polymorphic Encryption	Semi-honest	ไม่ได้



รูปที่ ๑๔ กระบวนการให้บริการการประมวลผลข้อมูลจากอุปกรณ์ IoT
บนคลาวด์อย่างปลอดภัยและป้องกันการรั่วไหลของข้อมูลส่วนบุคคล

รูปที่ ๑๔ แสดงตัวอย่างกระบวนการนำข้อมูลจากอุปกรณ์ Sensor ไปประมวลผลบนคลาวด์อย่างปลอดภัย โดยอุปกรณ์ Raspberry Pi ทำหน้าที่เป็นอุปกรณ์ Edge computing ขนาดเล็กที่ใช้ในการสร้างกุญแจและเข้ารหัสข้อมูลที่ได้รับมาจากอุปกรณ์ Sensor และข้อมูลที่ถูกเข้ารหัสจะถูกส่งไปประมวลผลที่คลาวด์ และข้อมูลผลลัพธ์ที่ได้จะถูกส่งมาถอดรหัสและแสดงผลที่แอปพลิเคชันปลายทางของผู้ใช้งาน

๒.๒.๕ HE Libraries

เทคนิคการเข้ารหัสแบบ HE ได้ถูกนำมาใช้งานจริงในแอปพลิเคชันต่างๆ อย่างแพร่หลายมากขึ้น ทำให้มีผู้พัฒนา Compiler และ ไลบรารีสำหรับการคำนวณ HE แบบต่างๆ ดังตารางที่ ๗

ตารางที่ ๗ Compiler และ ไลบรารีสำหรับเทคนิคการเข้ารหัสแบบ HE

FHE Library	อ้างอิง	เทคนิคที่ใช้	โปรแกรมภาษาที่ใช้
Microsoft SEAL	[54]	SWHE, BFV, CKKS, RNS-CKKS	C++
HElib	[28]	CKKS, BFV	C++
PALISADE	[48]	BGV, BFV, CKKS, FHEW,	Cross-platform C++
Pyfhel	[49]	Afhe1, SEAL, PALISADE	Python
PySEAL	[50]	SWHE, BFV, SEAL	Python
PySyft	[72]	SWHE	Python
FV-NFLlib,	[26]	SWHE, FV	C++
HEAT	[27]	HE API	C++
lattigo	[73]	BFV, CKKS, RNS-CKKS	Go

FHE Library	อ้างอิง	เทคนิคที่ใช้	โปรแกรมภาษาที่ใช้
HEAAN	[74]	CKKS, RNS-CKKS	C++
concrete	[75]	GSW	Rust

บทสรุป

เทคนิคการเข้ารหัสแบบ HE เป็นเทคนิคที่ช่วยให้ข้อมูลที่ถูกเข้ารหัสสามารถคำนวณทางคณิตศาสตร์ได้โดยไม่ต้องถอดรหัส ซึ่งเป็นหนึ่งในเทคนิคที่สามารถนำมาประยุกต์ใช้งานในการแก้ปัญหาความต้องการปกป้องความเป็นส่วนตัวเป็นส่วนตัวของข้อมูลในขณะเดียวกันต้องการพึ่งพาพลังการคำนวณของหน่วยประมวลผลบนคลาวด์ การนำข้อมูลส่วนบุคคลไป เทคนิคการเข้ารหัส HE มีด้วยกันหลายวิธี ซึ่งสามารถจำแนกได้ออกเป็น 3 ประเภทตามจำนวนความสามารถในการทำโอเพอร์เรชั่นทางคณิตศาสตร์ วิธี PHE เป็นวิธีที่ง่ายและซับซ้อนน้อยที่สุด ซึ่งส่วนใหญ่จะถูกนำมาใช้ในกรณีที่ความต้องการคำนวณทางคณิตศาสตร์ที่ซับซ้อนและในกรณีที่ข้อได้เปรียบที่ใช้ทรัพยากรการคำนวณน้อยจึงเหมาะสมในการนำมาใช้ประมวลผลบนอุปกรณ์ IoT อย่างไรก็ตาม หากมีความต้องการประมวลผลทางคณิตศาสตร์ที่ซับซ้อน SWHE เป็นอีกทางเลือกที่นิยมนำมาใช้ในการประมวลผลข้อมูลที่ได้จากอุปกรณ์ IoT และ ในปัจจุบันมีไลบรารีสำหรับการเข้ารหัสด้วยเทคนิค HE มากมายโดยจำแนกได้ตามโปรแกรมภาษาที่ต้องการใช้งาน เช่น C++, Python เป็นต้น ดังนั้นความท้าทายของงานวิจัยจึงขึ้นอยู่กับการออกแบบระบบให้เหมาะสมกับปัญหาที่ต้องการ และการเลือกใช้วิธีเข้ารหัสที่มีประสิทธิภาพ

๒.๓ ผลงานศึกษาการนำเทคโนโลยี HE มาให้บริการในต่างประเทศ

ในส่วนนี้ได้รวบรวมผลการสำรวจและวิเคราะห์ระบบการให้บริการคุ้มครองความเป็นส่วนตัวที่นำเทคโนโลยี HE มาใช้ในตลาดต่างประเทศ เพื่อนำมาเป็นแนวทางในการพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีสำหรับโครงการฯ โดยมีเป้าหมายในการพัฒนาและประยุกต์เทคโนโลยี HE ให้เหมาะสมกับระบบ IoT และ IoT cloud platform ของไทย

จากการสำรวจพบว่าปัจจุบันเริ่มมีบริษัทต่างๆ ในต่างประเทศนำเทคโนโลยี HE ไปใช้งานในเชิงพาณิชย์มากยิ่งขึ้นโดยเฉพาะอย่างยิ่งการนำไปใช้ประโยชน์ในการคำนวณ AI การสร้าง Machine Learning (ML) Model การใช้งานบนคลาวด์

๒.๓.๑ วิเคราะห์บริษัทที่ให้บริการคุ้มครองความเป็นส่วนตัว (Privacy as a Service)

๑) บริษัท Inpher เป็นบริษัทที่ได้รับเงินลงทุนจากกองทุน Amazon Alexa และ JPMorgan Chase & Co. โดยบริษัท Inpher ได้จดทะเบียนตั้งอยู่ที่เมือง New York ประเทศสหรัฐอเมริกา โดยมีเงินลงทุนมูลค่า ๑๔ ล้าน USD บริษัท Inpher ได้นำเทคโนโลยีการเข้ารหัส FHE และ SMPC ไปให้บริการในเพื่อคุ้มครองความเป็นส่วนตัวของข้อมูลในการประมวลผล Machine Learning ใน Cloud โดย เทคนิคการคุ้มครองความเป็นส่วนตัวของข้อมูล ประกอบด้วย เทคนิค Secure Multi-Party Computation (SMPC) และ FHE รูปที่ ๑๕ แสดงให้เห็นถึงสถาปัตยกรรมโครงสร้างการคำนวณอย่างปลอดภัยเพื่อคุ้มครองความเป็นส่วนตัวของข้อมูล

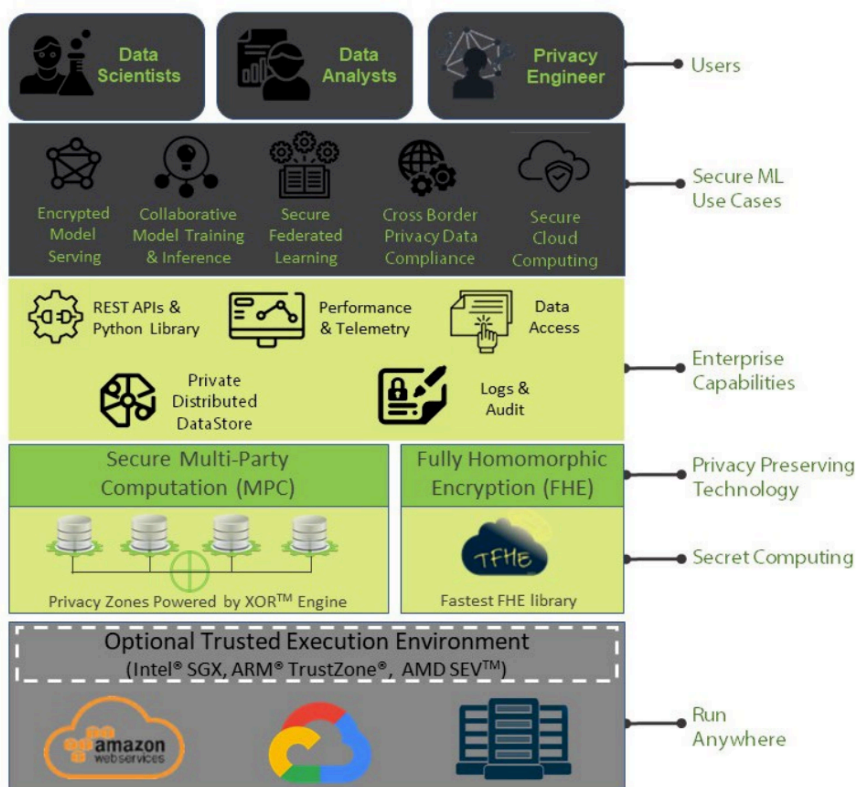
การให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลของบริษัท Inpher

๑) การร่วมมือกันคำนวณ ML model อย่างปลอดภัยโดยสามารถคุ้มครองความเป็นส่วนตัวของข้อมูล (Collaborative Model Training and Inference) โดยใช้เทคนิค SMPC

๒) การร่วมกันคำนวณจากหลายกลุ่มด้วยเทคนิค Federated Learning อย่างปลอดภัย (Secure Federated Learning)

๓) การเข้ารหัส ML Model (Encrypted Model Serving) ด้วย FHE

๔) การรับประกันการคุ้มครองข้อมูลส่วนบุคคลข้ามเขตแดน (Cross Border Data Privacy Compliance)

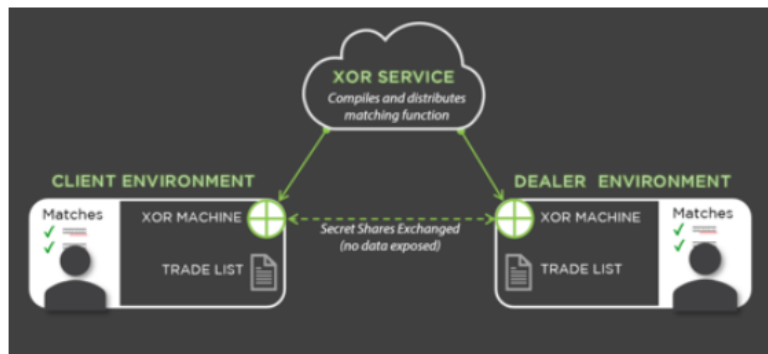


รูปที่ ๑๕ สถาปัตยกรรมการคำนวณอย่างปลอดภัย เพื่อบริการคุ้มครองความเป็นส่วนตัวของข้อมูลบริษัท Inpher

การนำไปใช้งานจริงเชิงพาณิชย์ของบริษัท Inpher

๑) ใช้สำหรับการคำนวณต้นทุนและประสิทธิภาพการทำงานของเครื่องบินรบ โดยการทำ Predictive maintenance สำหรับการซ่อมบำรุงเครื่องยนต์ของเครื่องบินรบนั้น ข้อมูลประสิทธิภาพและการทำงานของเครื่องยนต์ต้องได้รับการคุ้มครองความเป็นส่วนตัวของข้อมูล เนื่องจากข้อมูลเหล่านี้มีผลต่อการรบในสมรภูมิรบ โดยไม่สามารถให้ข้อมูลรั่วไหลไปสู่ภายนอกได้ จึงมีความจำเป็นอย่างยิ่งที่ต้องได้รับการคุ้มครองความเป็นส่วนตัวของข้อมูล แต่ทั้งนี้การประเมินความเสี่ยงของประสิทธิภาพของเครื่องยนต์ในเครื่องบินรบมีความสำคัญอย่างยิ่งต่อทหารในสมรภูมิรบและข้อมูลการประมวลผลเหล่านี้จำเป็นต้องพึ่งพาความสามารถในการคำนวณของคลาวด์ อีกทั้งการส่งข้อมูลการใช้งานเครื่องยนต์ของเครื่องบินรบด้วยระบบ IoT จำเป็นต้องพึ่งพา IoT cloud platform ซึ่งหลีกเลี่ยงไม่ได้เลยที่จะต้องส่งข้อมูลของเครื่องยนต์ไปประมวลผลบนคลาวด์ การเข้ารหัสข้อมูลด้วยเทคนิค FHE และการใช้เทคนิค SMPC มาช่วยในการกระจายการคำนวณแบบ Ad-hoc network ในสมรภูมิรบช่วยให้ศูนย์บัญชาการสามารถประเมินสถานการณ์ของทรัพยากรการรบได้แม่นยำและปลอดภัยมากยิ่งขึ้น

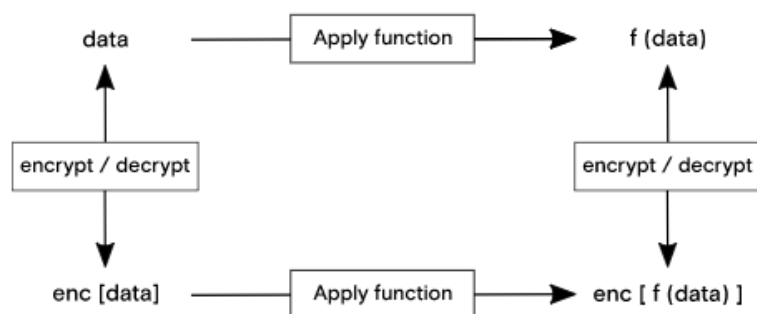
๒) ใช้ในการให้บริการ Financial โดยลูกค้าเป็นกลุ่มธนาคารพาณิชย์ที่ต้องการอาศัยคลาวด์ในการคำนวณข้อมูลการเงินของลูกค้าในการให้บริการ Fraud Data Collaboration, Data-Blind Trade Matching, Consumer Credit Modeling และการสร้างกฎทางการเงิน Uncleared Margin Rules ทั้งนี้ การร่วมกันคำนวณจากฐานข้อมูลลูกค้าข้ามธนาคารจะช่วยให้สามารถสร้างโมเดลการมอบคุณค่าให้กับลูกค้าได้ถูกต้องมากยิ่งขึ้น แต่การแชร์ข้อมูลลูกค้าข้ามธนาคารเป็นการละเมิดความเป็นส่วนตัวของลูกค้า ดังนั้น เทคนิค SMPC และ FHE จึงถูกนำมาช่วยในการคุ้มครองความเป็นส่วนตัวของลูกค้า



รูปที่ ๑๖ การคำนวณอย่างปลอดภัยสำหรับบริการ Data-blind matching ของบริษัท Inpher

๒) บริษัท CryptoExperts ตั้งอยู่ที่เมืองปารีสประเทศฝรั่งเศส ได้นำเสนอผลิตภัณฑ์ White-Box Cryptography โดยอาศัยเทคนิคการเข้ารหัสแบบ PHE ซึ่งรูปแบบการคำนวณภายใต้ผลิตภัณฑ์ของบริษัท CryptoExperts ได้มาตรฐาน ISO/IEC 18033-6:2019 [ISO/IEC] ซึ่งเป็นมาตรฐานเกี่ยวกับระเบียบวิธีการเข้ารหัสแบบ HE โดยบริการที่ทางบริษัท CryptoExperts นำเสนอคือการให้บริการออกแบบระบบการให้บริการคุ้มครองความปลอดภัยของข้อมูลที่เข้ากับ Application และความต้องการใช้งานของลูกค้า ซึ่งแตกต่างจากการให้บริการของบริษัท Inpher ที่สร้างผลิตภัณฑ์ให้สามารถนำไปใช้งานได้ในระบบ Cloud computing

๓) บริษัท NuCypher ได้สร้างผลิตภัณฑ์ชื่อ NuFHE ซึ่งอยู่บนพื้นฐานของ FHE Library TFHE ที่สามารถเร่งความเร็วในการคำนวณด้วย GPU และใช้ CUDA และ OpenCL ด้วยเทคนิคนี้ทำให้ NuFHE ใช้วิธีการคำนวณด้วย Fast-Furier Transform (FFT) และการคำนวณ integer Number Theroetic Transforms (NTT) เพื่อให้การคำนวณบนข้อมูลที่เข้ารหัสทำได้รวดเร็วมากยิ่งขึ้น



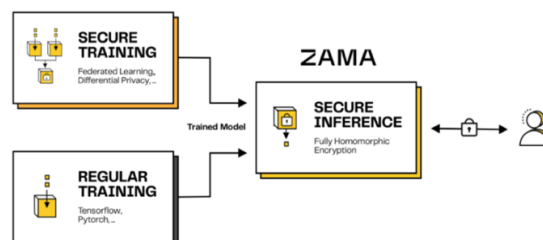
รูปที่ ๑๗ วิธีการทำงานของระบบการเข้ารหัสแบบ FHE

ผลการทดสอบประสิทธิภาพของเทคนิค NuFHE เปรียบเทียบกับเทคนิคการเข้ารหัสแบบ FHE อื่น แสดงดังตารางที่ ๘

ตารางที่ ๘ ผลการทดสอบประสิทธิภาพของเทคนิค NuFHE

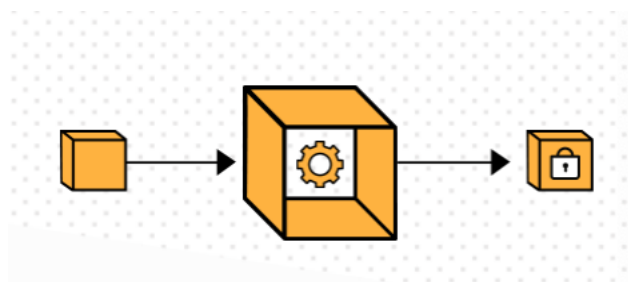
Platform	Library	Performance (ms/bits)	
		Binary Gate	MUX Gate
Single Core/Single GPU-FFT	TFHE (CPU)	13	26
	nuFHE	0.13	0.22
	Speedup	100.9	117.7
Single Core/Single GPU-NTT	cuFHE	0.35	N/A
	nuFHE	0.35	0.67
	Speedup	1.0	-

๔) บริษัท Zama [Zama] ได้นำเสนอการให้บริการสร้าง ML model ด้วยเทคโนโลยี FHE โดยรูปแบบการสร้างโมเดลเป็นดังรูปที่ ๑๘ ซึ่งเป็นเทคนิคที่สามารถผสมผสานการเทรนโมเดลด้วยข้อมูลที่ถูกเข้ารหัสและข้อมูลที่ไม่ได้ถูกเข้ารหัสได้



รูปที่ ๑๘ ไดอะแกรมการ Train ML model โดยให้ความปลอดภัยและความคุ้มครองข้อมูลส่วนบุคคล ภายใต้ผลิตภัณฑ์ของบริษัท Zama

นอกจากการให้บริการสร้าง Train ML model แล้วยังมีหน้าที่ให้บริการ Compile โมเดลเดิมที่สร้างไว้แล้วด้วยข้อมูลที่ไม่ได้เข้ารหัสไปสู่รูปแบบโมเดลที่เข้ารหัสด้วยเทคนิค FHE



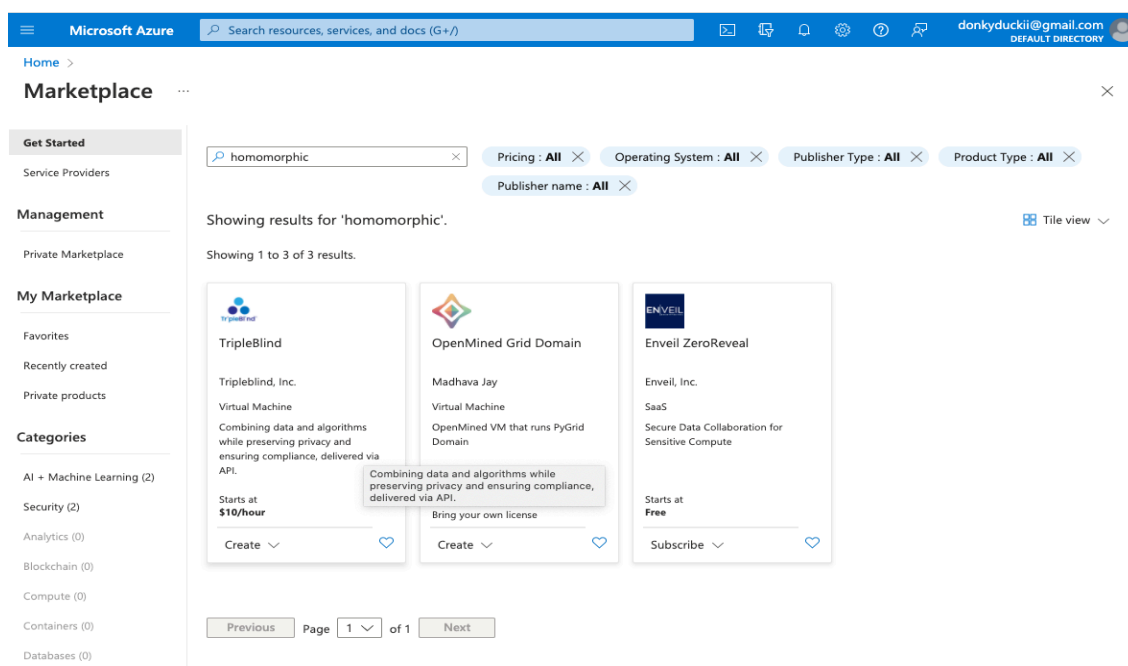
รูปที่ ๑๙ การ Compile MLโมเดลที่สร้างไว้แล้วให้เป็น ML โมเดลที่ถูกเข้ารหัส

๒.๓.๒ วิเคราะห์ตลาด Privacy as a Service ต่างประเทศ

เทคโนโลยีการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลในประเทศ ได้ถูกนำไปเสนอเป็นทางเลือกให้กับผู้ให้บริการคลาวด์บน Microsoft Azure Marketplace และ AWS Marketplace โดยมีรายละเอียดดังนี้

๑) Microsoft Azure Marketplace

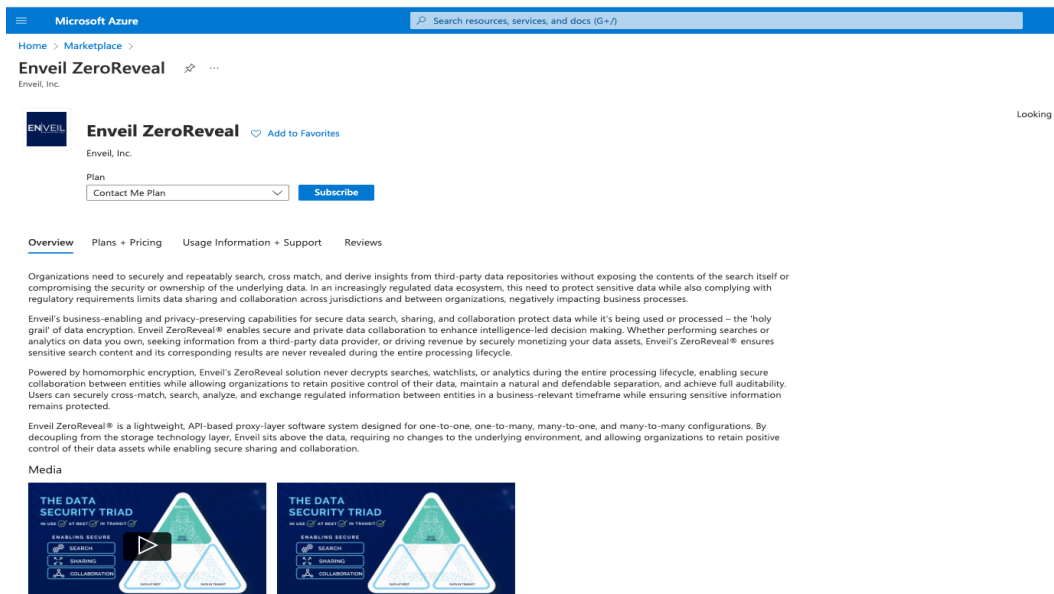
จากการสำรวจการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน Microsoft Azure Marketplace พบว่ามีผู้ให้บริการ ๓ รายที่ใช้เทคโนโลยี HE และ SMPC ในการให้บริการ ได้แก่ ๑ Enveil ZeroReveal, ๒) OpenMinded Grid Domain, และ ๓) TripleBlind ดังรูปที่ ๒๐



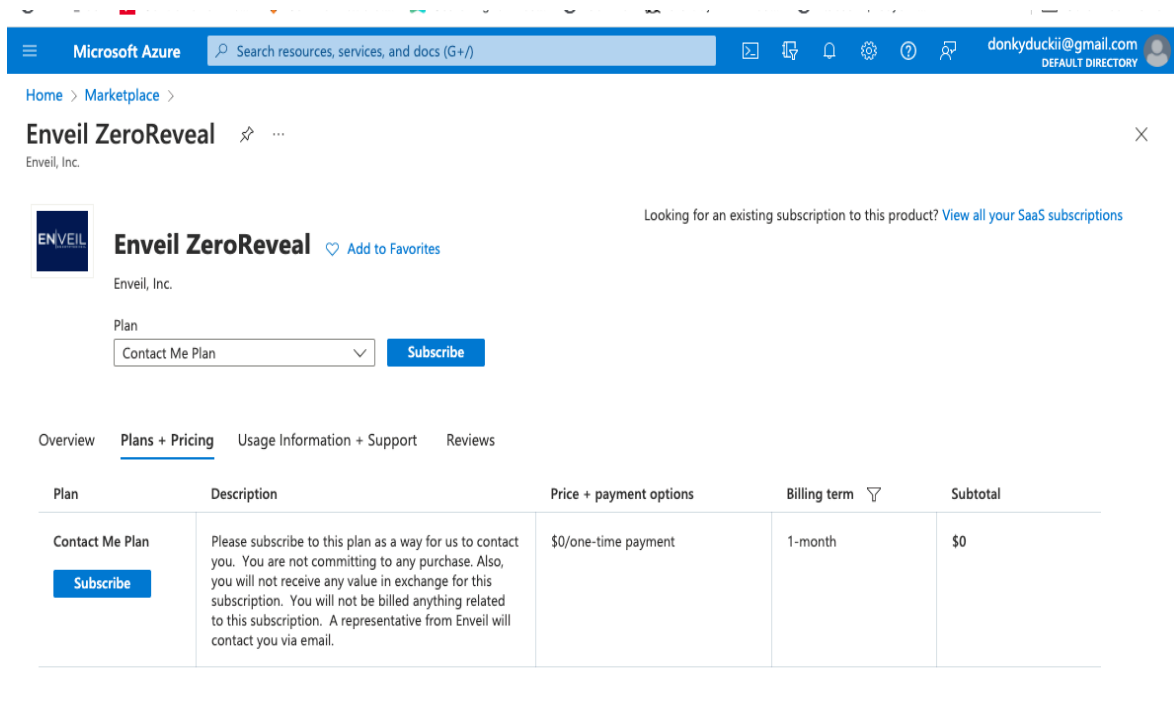
รูปที่ ๒๐ บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน Microsoft Azure Marketplace

๑.๑) Enveil ZeroReveal

บริษัท Enveil ได้เปิดให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลให้กับลูกค้าของ Microsoft Azure Marketplace ภายใต้เทคนิค HE และ SMPC โดยไม่ได้เปิดเผยราคาในการใช้งาน ซึ่ง Application ที่แนะนำให้นำไปใช้งานได้แก่ Search, Share และ Collaboration ระหว่างองค์กร



รูปที่ ๒๑ รายละเอียดด้านเทคนิคสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล Enveil ZeroReveal

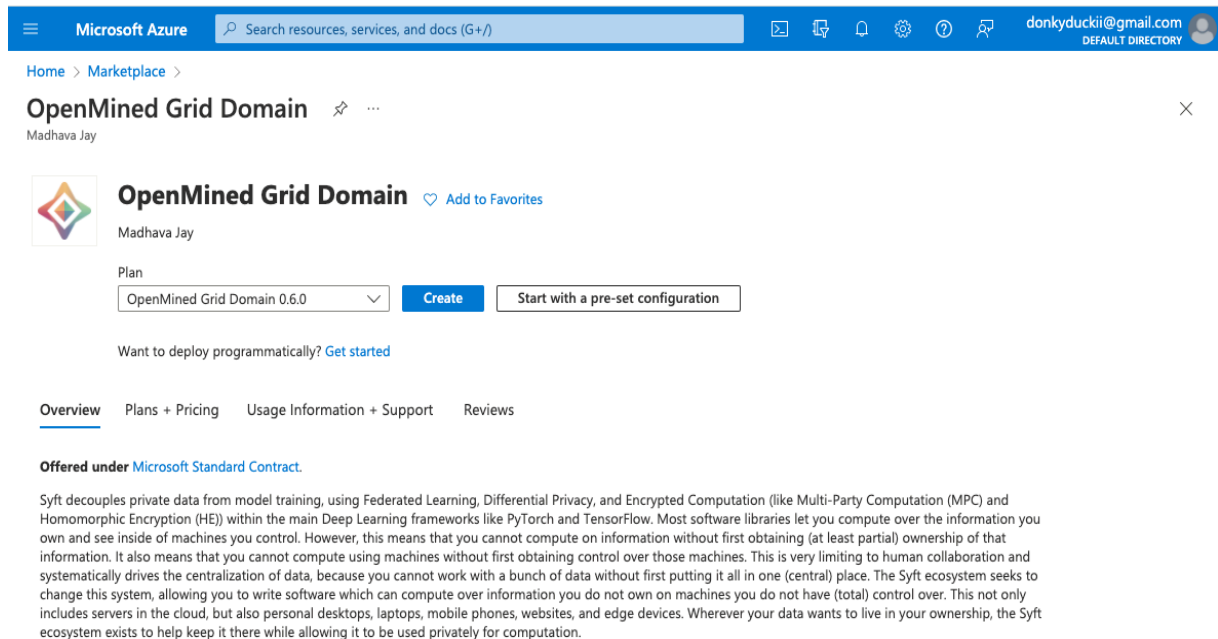


รูปที่ ๒๒ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล Enveil ZeroReveal

๑.๒) OpenMined Grid Domain

บริษัท OpenMined ได้เปิดตัว OpenMined Grid Domain ที่ไว้ให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลภายใต้ Library Syft ที่ ที่พัฒนาขึ้น โดยเทคนิคที่ใช้คือ Federated Learning, MPC และ HE ซึ่งการ

ให้บริการของ OpenMined Grid Domain จะเน้นไปที่ Deep Learning Application ทั้งนี้ในส่วนรายละเอียดด้านราคายังไม่มีการเปิดเผยข้อมูล



Microsoft Azure Search resources, services, and docs (G+)

Home > Marketplace >

OpenMined Grid Domain

Madhava Jay

Plan

OpenMined Grid Domain 0.6.0 [Create](#) [Start with a pre-set configuration](#)

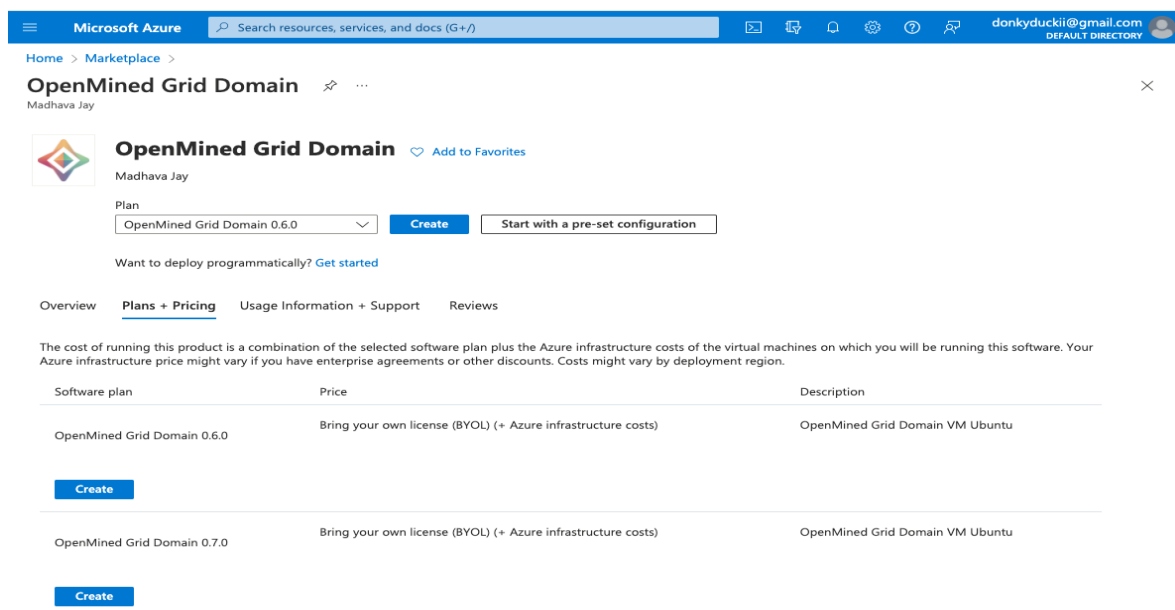
Want to deploy programmatically? [Get started](#)

Overview Plans + Pricing Usage Information + Support Reviews

Offered under [Microsoft Standard Contract](#).

Syft decouples private data from model training, using Federated Learning, Differential Privacy, and Encrypted Computation (like Multi-Party Computation (MPC) and Homomorphic Encryption (HE)) within the main Deep Learning frameworks like PyTorch and TensorFlow. Most software libraries let you compute over the information you own and see inside of machines you control. However, this means that you cannot compute on information without first obtaining (at least partial) ownership of that information. It also means that you cannot compute using machines without first obtaining control over those machines. This is very limiting to human collaboration and systematically drives the centralization of data, because you cannot work with a bunch of data without first putting it all in one (central) place. The Syft ecosystem seeks to change this system, allowing you to write software which can compute over information you do not own on machines you do not have (total) control over. This not only includes servers in the cloud, but also personal desktops, laptops, mobile phones, websites, and edge devices. Wherever your data wants to live in your ownership, the Syft ecosystem exists to help keep it there while allowing it to be used privately for computation.

รูปที่ ๒๓ รายละเอียดด้านเทคนิคสำหรับบริการคุ้มครองความเป็นส่วนตัวของ Open Mined Grid Domain



Microsoft Azure Search resources, services, and docs (G+)

Home > Marketplace >

OpenMined Grid Domain

Madhava Jay

Plan

OpenMined Grid Domain 0.6.0 [Create](#) [Start with a pre-set configuration](#)

Want to deploy programmatically? [Get started](#)

Overview **Plans + Pricing** Usage Information + Support Reviews

The cost of running this product is a combination of the selected software plan plus the Azure infrastructure costs of the virtual machines on which you will be running this software. Your Azure infrastructure price might vary if you have enterprise agreements or other discounts. Costs might vary by deployment region.

Software plan	Price	Description
OpenMined Grid Domain 0.6.0	Bring your own license (BYOL) (+ Azure infrastructure costs)	OpenMined Grid Domain VM Ubuntu
Create		
OpenMined Grid Domain 0.7.0	Bring your own license (BYOL) (+ Azure infrastructure costs)	OpenMined Grid Domain VM Ubuntu
Create		

รูปที่ ๒๔ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล OpenMined Grid Domain

๑.๓) TripleBlind

บริษัท TripleBlind ได้เปิดให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลผ่านทาง API ที่เน้นไปทางการใช้งานกับลูกค้ากลุ่ม Healthcare และ Financial โดยมีผู้ร่วมทุนคือ บริษัท Accenture, General Catalyst และ The Mayo Clinic เทคนิคที่ใช้ในการให้บริการคำนวณอยู่ภายใต้ HE

TripleBlind
Tripleblind, Inc.

Plan
TripleBlind 30 Day Evaluation

Want to deploy programmatically? [Get started](#)

Overview Plans + Pricing Usage Information + Support Reviews

Offered under Microsoft Standard Contract.

TripleBlind has created the most complete and scalable solution for privacy enhancing computation. The TripleBlind solution is software-only and delivered via a simple API. It solves for a broad range of use cases, with current focus on healthcare and financial services. The company is backed by Accenture, General Catalyst and The Mayo Clinic. TripleBlind's innovations build on well understood principles, such as federated learning and multi-party compute. Our innovations radically improve the practical use of privacy preserving technology, by adding true scalability and faster processing, with support for all data and algorithm types. TripleBlind natively supports major cloud platforms, including availability for download and purchase via cloud marketplaces. TripleBlind unlocks the intellectual property value of data, while preserving privacy and ensuring compliance with HIPAA and GDPR. TripleBlind compares favorably with other privacy preserving technologies, such as homomorphic encryption, synthetic data, and tokenization and has documented use cases for more than two dozen mission critical business problems.

Media

for a live demo or free hands-on workshop!

How Data Ecosystems are Broken Today

DISCUSSION: What's Your Pain Point?

1. How are you ingesting third party (external) data from third parties?
2. How are you sharing your data (or algorithms) with third parties?
3. What challenges are you facing in these processes? (Legal, operational, financial, time)
4. How are you controlling the privacy of that data today and what could be going better?
 - Have agreements with vendors, implement encryption, federated learning, secure enclaves, homomorphic encryption, synthetic data, tokenization, etc.
5. Are there opportunities you are not pursuing or not seeing? A dream data set?

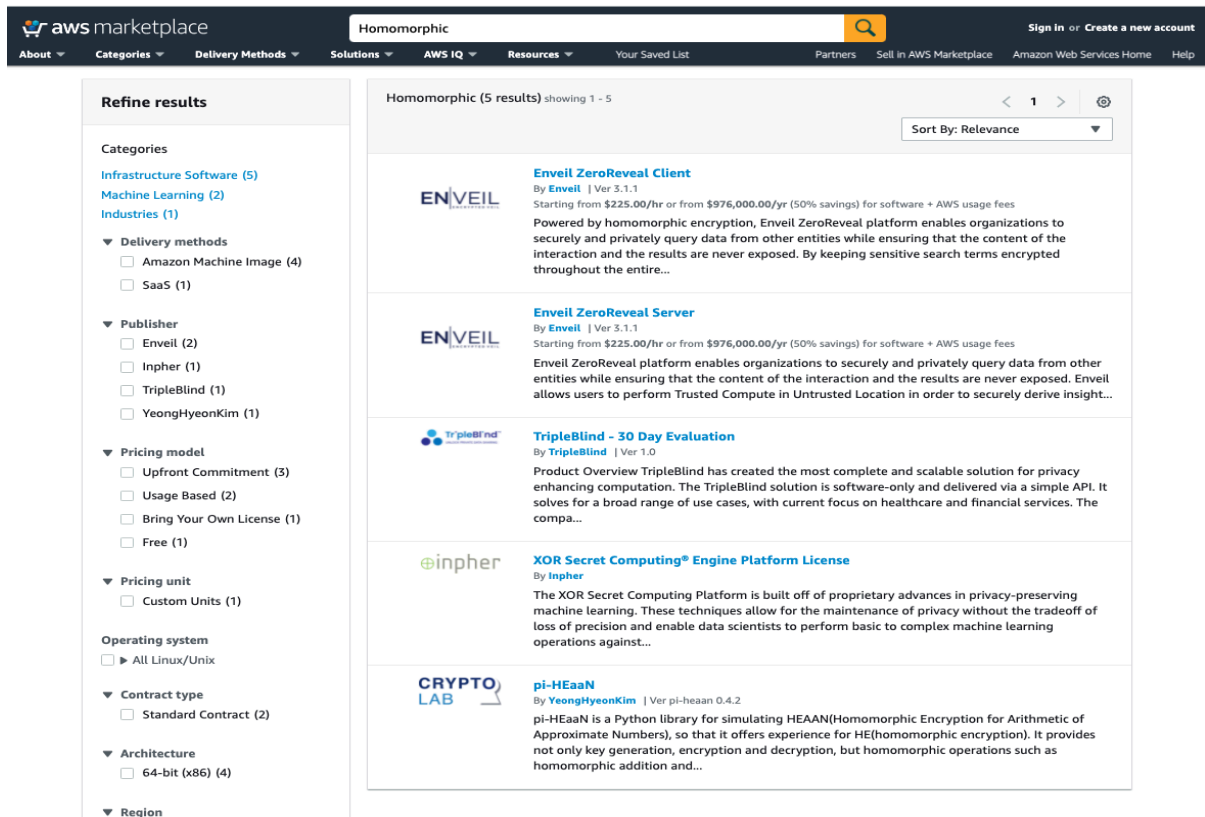
รูปที่ ๒๕ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล TripleBlind

Software plan	Price	Description
TripleBlind 30 Day Evaluation	Bring your own license (BYOL) (+ Azure infrastructure costs)	Select this plan to take TripleBlind for a test drive and learn how the API can be quickly and easily implemented. You will be contacted to arrange payment to get started easily. This plan uses the standard MSFT Azure licensing agreement. Please select a recommended VM configuration and attach a data disk with at least 100GBs of storage.
TripleBlind Annual Enterprise License	Bring your own license (BYOL) (+ Azure infrastructure costs)	TripleBlind annual enterprise license, enjoy the business benefits of leveraging privacy to derive more value from your data and algorithms. You will be contacted to arrange payment. This plan uses the MSFT Azure standard license agreement. Please select a recommended VM configuration and attach a data disk with at least 100GBs of storage.
TripleBlind Hourly Software Cost	Starting at \$250/hour (+ Azure Infrastructure costs)	TripleBlind per hour based pricing plan, pay as you go based on the number of hours that your VM is running. Please select a recommended VM configuration and attach a data disk with at least 100GBs of storage. An onboarding process is required before you can begin using the TripleBlind solution. We highly recommend during the subscription process that you setup the auto-shutdown feature, or stop the VM manually after subscribing, to minimize your cost during the onboarding process.
TripleBlind Hourly Based Eval Plan	Starting at \$10/hour (+ Azure Infrastructure costs)	TripleBlind per hour based pricing plan, pay as you go based on the number of hours that your VM is running. Please select a recommended VM configuration and attach a data disk with at least 100GBs of storage. An onboarding process is required before you can begin using the TripleBlind solution. We highly recommend during the subscription process that you setup the auto-shutdown feature, or stop the VM manually after subscribing, to minimize your cost during the onboarding process.

รูปที่ ๒๖ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล OpenMined Grid Domain

๒) AWS Marketplace

จากการสำรวจการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน Microsoft Azure Marketplace พบว่ามีผู้ให้บริการ ๓ รายที่ใช้เทคโนโลยี HE และ SMPC ในการให้บริการ ได้แก่ ๑) Enveil ZeroReveal, ๒) TripleBlind, ๓) Inpher และ ๔) CryptoLAB ดังรูปที่ ๒๗



รูปที่ ๒๗ บริการคุ้มครองความเป็นส่วนตัวของข้อมูลบน AWS Marketplace

๒.๑) Enveil ZeroReveal

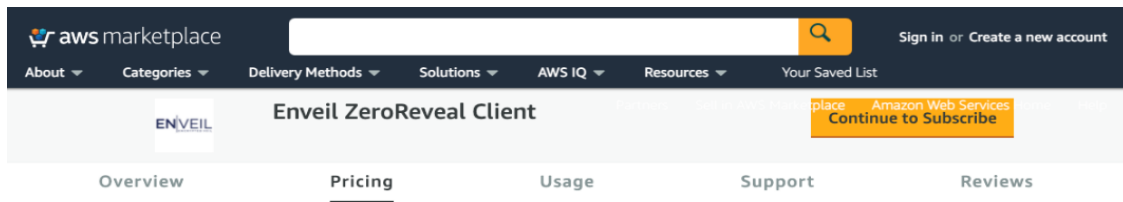
รายละเอียดการให้บริการของ Enveil ZeroReveal ใน AWS Marketplace คล้ายกันกับใน Microsoft Azure Marketplace แต่แบ่งการให้บริการออกเป็น Client และ Server

- Enveil ZeroReveal Client

ในส่วนของ Client จะมีค่าใช้จ่ายในการให้บริการแบ่งเป็นส่วนของ Software ต่อการใช้งาน 1 ชั่วโมง โดยราคาที่ 225 USD และการใช้งานในส่วนของการคำนวณบนคลาวด์ 0.816 USD/hr การให้บริการในส่วนของฝั่ง Client Application จะให้บริการ API ที่ทำให้ผู้ใช้งานสามารถใช้งานบน Application การ search และการทำ Data Analytic โดย ฝั่ง Server จะเป็นผู้คำนวณโดยปราศจากการ Decrypt ข้อมูล และส่วนของ Client จะทำการดึงข้อมูลผ่าน API

- Enveil ZeroReveal Server

ในส่วนของ Server จะมีค่าใช้จ่ายในการให้บริการแบ่งเป็นส่วนของ Software ต่อการใช้งาน 1 ชั่วโมง โดยราคาที่ 225 USD และการใช้งานในส่วนของการคำนวณบนคลาวด์ 0.816 USD/hr การทำงานของ Server จะต้องมีการจับคู่ (Pair) ระหว่าง Application ในส่วนของ Client และ ในส่วนของ Server



Pricing Information

Use this tool to estimate the software and infrastructure costs based on your configuration choices. Your usage and costs might be different from this estimate. They will be reflected on your monthly AWS billing reports.

Estimating your costs

Choose your region and fulfillment option to see the pricing details. Then, modify the estimated price by choosing different instance types.

Region
AWS GovCloud (US) East

Fulfillment Option
64-bit (x86) Amazon Machine Image (AMI)

Software Pricing Details

Enveil ZeroReveal Client	\$225.00 /hr
running on c5.4xlarge	

Infrastructure Pricing Details

Estimated Infrastructure Cost	\$0.816 EC2/hr
--------------------------------------	-----------------------

The table shows current software and infrastructure pricing for services hosted in **AWS GovCloud (US) East**. Additional taxes or fees may apply.

Enveil ZeroReveal Client

Switch to annual pricing for savings up to 50%

Hourly Annual

EC2 Instance type	Software/hr	EC2/hr	Total/hr
c5.4xlarge			
★Vendor Recommended	\$225.00	\$0.816	\$225.816

รูปที่ ๒๘ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล Enveil ZeroReveal Client

Pricing Information

Use this tool to estimate the software and infrastructure costs based on your configuration choices. Your usage and costs might be different from this estimate. They will be reflected on your monthly AWS billing reports.

Estimating your costs

Choose your region and fulfillment option to see the pricing details. Then, modify the estimated price by choosing different instance types.

Region
AWS GovCloud (US) East

Fulfillment Option
64-bit (x86) Amazon Machine Image (AMI)

Software Pricing Details

Enveil ZeroReveal Server	\$225.00 /hr
running on c5.4xlarge	

Infrastructure Pricing Details

Estimated Infrastructure Cost	\$0.816 EC2/hr
--------------------------------------	-----------------------

The table shows current software and infrastructure pricing for services hosted in **AWS GovCloud (US) East**. Additional taxes or fees may apply.

Enveil ZeroReveal Server

Switch to annual pricing for savings up to 50%

Hourly Annual

EC2 Instance type	Software/hr	EC2/hr	Total/hr
c5.4xlarge			
★Vendor Recommended	\$225.00	\$0.816	\$225.816

รูปที่ ๒๙ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล Enveil ZeroReveal Server

๒.๒) TripleBlind

รายละเอียดการให้บริการของ TripleBlind ใน AWS Marketplace คล้ายกันกับใน Microsoft Azure Marketplace แต่รายละเอียดทางด้านราคาของบริการที่อยู่ใน AWS Marketplace เป็นดังรูปที่ ๓๐ ซึ่งจะให้

ทดลองใช้ฟรี 30 วัน และราคาจะขึ้นกับขนาดของหน่วยประมวลผลและพื้นที่ให้บริการบนคลาวด์ คิดโดยประมาณ อยู่ที่ 0.166 USD ต่อ EC2 ต่อชั่วโมง

Estimating your costs

Choose your region and fulfillment option to see the pricing details. Then, modify the estimated price by choosing different instance types.

Region: US East (N. Virginia)

Fulfillment Option: 64-bit (x86) Amazon Machine Image (AMI)

Software Pricing Details

TripleBlind - 30 Day Evaluation \$0 /hr > *running on t3.xlarge*

Infrastructure Pricing Details

Estimated Infrastructure Cost \$0.166 EC2/hr >

BYOL Available for customers with current licenses purchased via other channels.

The table shows current software and infrastructure pricing for services hosted in US East (N. Virginia). Additional taxes or fees may apply.

Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

EC2 Instance type	Software/hr	EC2/hr	Total/hr
☒ t3.xlarge ★ Vendor Recommended	\$0	\$0.166	\$0.166
☐ t3.2xlarge	\$0	\$0.333	\$0.333
☐ t3a.xlarge	\$0	\$0.15	\$0.15
☐ t3a.2xlarge	\$0	\$0.301	\$0.301

รูปที่ ๓๐ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล Enveil TripleBlind

๒.๓) inpher

การให้บริการ XOR Secret Computing Platform ภายใต้บริษัท inpher บน AWS Marketplace เป็นการให้บริการการทำ Machine Learning บนข้อมูลที่มีความ Sensitive ซึ่งต้องการคุ้มครองความเป็นส่วนตัวของข้อมูล โดยรายละเอียดของค่าใช้จ่ายในการให้บริการคิดเป็นต่อ ๑๒ เดือน อยู่ที่ 240,000 USD, ต่อ ๒๔ เดือน อยู่ที่ 456,000 USD และ ต่อ ๓๖ เดือน อยู่ที่ 648,000 USD ทั้งนี้ในการใช้งาน ผู้ใช้งานต้องลง XOR ในส่วนของ Client และ Cloud

XOR Secret Computing® Engine Platform License
Sold by: [Inpher](#)

The XOR Secret Computing® Platform enables privacy-preserving machine learning and analytics via encryption-in-use technology. It is fast, precise and supports a large number of

[Show more](#)

[Continue to Subscribe](#)

[Save to list](#)

Product Overview

The XOR Secret Computing Platform is built off of proprietary advances in privacy-preserving machine learning. These techniques allow for the maintenance of privacy without the tradeoff of loss of precision and enable data scientists to perform basic to complex machine learning operations against single or multiple data sources without ever revealing the inputs, requiring any data transfer, and without involvement of a third-party.

With XOR you can perform privacy preserving analytics and machine learning on sensitive data to meet compliance requirements, maintain customer confidentiality and build better data models.

The Platform License is designed for users in a production environment with multiple internal or external privacy zones where XOR needs to be installed on the client's architecture.

The Production License includes 10 authenticated users. The License includes a sandbox environment with three Inpher-hosted XOR Machines (virtual machines). The License also includes a separate production environment with 2 client-installed XOR Machines that can be used for internal use or external deployment to data sources. Platform Licensees can purchase additional XOR Machines if needed. With this license type, XOR Machines can be deployed to external (outside your organization) privacy zones to access more data. Clients who have requirements beyond what the Platform License offers, for example additional XOR Machines, can contact sales@inpher.io.

Highlights

- Supports analysis across multiple data sets with different features.
- Has mutual privacy controls for both the dataowner and analyst.
- Supports advanced machine learning functions and statistical inference predictions.

Pricing Information

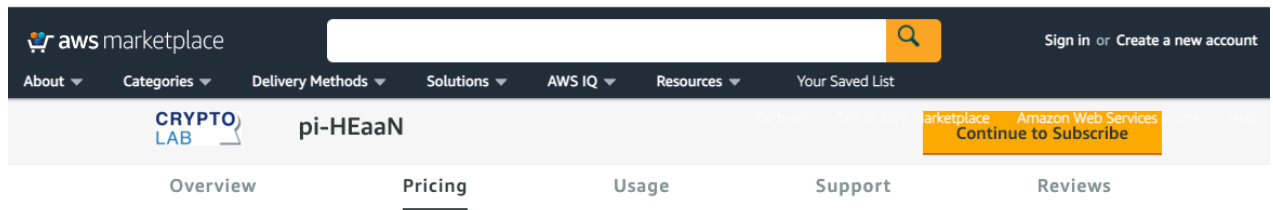
Below are the total costs for these different subscription durations. Additional taxes or fees may apply.

XOR Secret Computing® Engine Platform License		12 MONTHS	24 MONTHS	36 MONTHS
Units	Description			
Platform License	XOR Secret Computing Platform License	\$240,000	\$456,000	\$648,000

รูปที่ ๓๑ รายละเอียดด้านราคาสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล inpher บน AWS Marketplace

๒.๔) CRYPTO LAB

CRYPTO LAB เป็น Platform ที่ให้บริการภายใต้ HE ที่เขียนด้วยโปรแกรมภาษา Python ใน Library ชื่อ pi-HEaaN [pi-HEaaN] ซึ่งการคำนวณรองรับการทำโอเพอร์เรชั่นการคูณ และการบวก โดย Application ที่รองรับคือการทำ Data Analytics และการทำ Security ในที่นี้ได้กำหนดให้มีค่าใช้จ่ายในการให้บริการเป็นดังรูปที่ ๓๒ ซึ่งราคาจะขึ้นอยู่กับเขตพื้นที่ของคลาวด์ที่เลือกใช้ และขนาดของหน่วยประมวลผลและเวลาในการใช้งาน



Pricing Information

Use this tool to estimate the software and infrastructure costs based on your configuration choices. Your usage and costs might be different from this estimate. They will be reflected on your monthly AWS billing reports.

Estimating your costs

Choose your region and fulfillment option to see the pricing details. Then, modify the estimated price by choosing different instance types.

Region
US East (N. Virginia)

Fulfillment Option
64-bit (x86) Amazon Machine Image (AMI)

Software Pricing Details

pi-HEaaS \$0 /hr >
running on t2.micro

Infrastructure Pricing Details

Estimated Infrastructure Cost \$0.012 EC2/hr >

Free Tier EC2 charges for Micro instances are free for up to 750 hours a month if you qualify for the [AWS Free Tier](#).

The table shows current software and infrastructure pricing for services hosted in **US East (N. Virginia)**. Additional taxes or fees may apply.

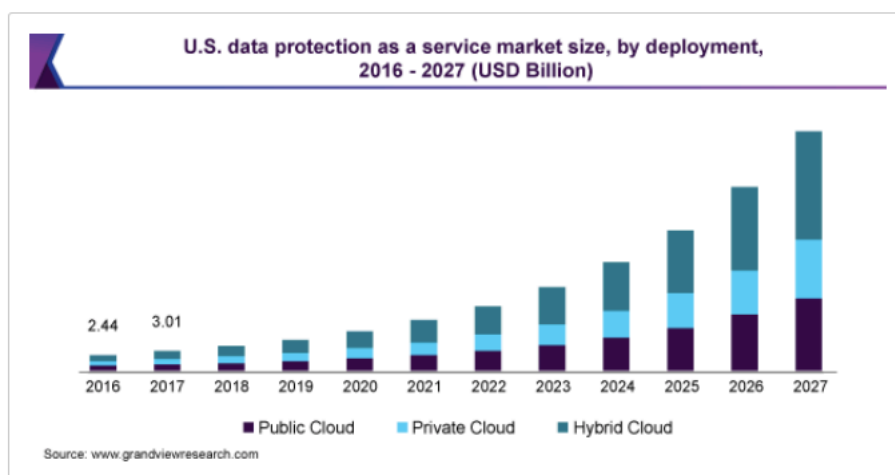
Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

pi-HEaaS			
EC2 Instance type	Software/hr	EC2/hr	Total/hr
☐ t2.nano	\$0	\$0.006	\$0.006
☒ t2.micro ★Vendor Recommended	\$0	\$0.012	\$0.012
☐ t2.small	\$0	\$0.023	\$0.023
☐ t2.medium	\$0	\$0.046	\$0.046
☐ t2.large	\$0	\$0.093	\$0.093

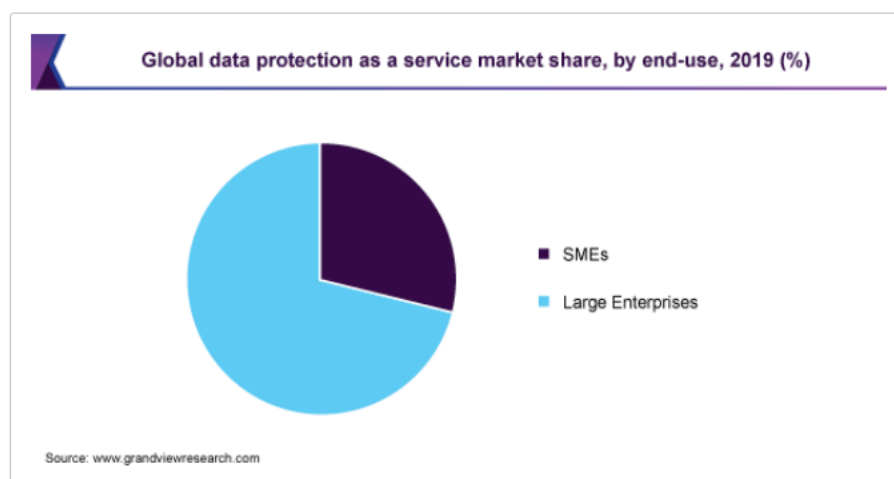
รูปที่ ๓๒ รายละเอียดด้านราคาสำหรับบริการคุ้มครองความเป็นส่วนตัวของข้อมูล CRYPTO-LAB บน AWS Marketplace

๒.๓.๓ วิเคราะห์กลุ่มลูกค้าในตลาดต่างประเทศ

จากการสำรวจบริษัทที่เปิดให้บริการกลุ่มลูกค้าคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับการประมวลผลบนคลาวด์สาธารณะ พบว่าบริการประกอบด้วยกลุ่มด้าน Application ด้านข้อมูลสุขภาพ (Healthcare), กลุ่มด้านการเงิน, กลุ่มบริหารจัดการเงินเดือนพนักงาน, กลุ่มทางการแพทย์ และการค้นหาข้อมูล ผลการสำรวจมูลค่าการให้บริการปกป้องข้อมูลในตลาดโลกพบว่าการเติบโตขึ้นอย่างก้าวกระโดด โดยมีอัตราการเติบโตอยู่ที่ ร้อยละ ๓๑ จากปี ๒๐๒๐ ถึงปี ๒๐๒๗ ดังรูปที่ ๓๓ ซึ่งการเติบโตของตลาดในกลุ่มการให้บริการนี้สืบเนื่องมาจากผลบังคับใช้ GDPR ในหลายประเทศ และความเป็นกังวลของลูกค้าในกลุ่มองค์กรด้านความปลอดภัยของข้อมูลรั่วไหลสู่ภายนอก โดยแนวโน้มการใช้บริการเป็นสัดส่วนของ SME และ องค์กรขนาดใหญ่ อยู่ที่เกือบ ๑/๓ ของมูลค่าตลาดทั้งหมดเป็นดังรูป ๓๔ โดยนักวิเคราะห์มองว่าตลาดการให้บริการ Privacy as a service ยังเป็นที่ต้องการเพิ่มมากขึ้นในอนาคต



รูปที่ ๓๓ มูลค่าทางการตลาดของ Data Protection as a Service ในตลาดโลก [76]



รูปที่ ๓๔ สัดส่วนทางการตลาดของ Data Protection as a Service ในตลาดโลก [76]

๒.๔.๔ วิเคราะห์เทคโนโลยีการให้บริการคุ้มครองข้อมูลส่วนบุคคลในต่างประเทศมาใช้กับ NETPIE

จากการสำรวจเทคนิคที่นำมาใช้ในการให้บริการคุ้มครองข้อมูลส่วนบุคคลของบริษัทในต่างประเทศพบว่า เทคนิคดังกล่าวมีข้อจำกัดในการนำมาประยุกต์ใช้กับ NETPIE เนื่องจาก NETPIE เป็นคลาวด์ที่ให้บริการเชื่อมต่อ IoT ที่เน้นไปที่การเก็บข้อมูลและกระจายข้อมูล IoT ไปยังอุปกรณ์ IoT และอุปกรณ์แสดงผลของผู้ใช้งาน โดยการคำนวณข้อมูลบน NETPIE จะทำใน Device Shadow ซึ่งรองรับการคำนวณพื้นฐาน เช่น บวก ลบ คูณ และหาร และ NETPIE ไม่อนุญาตให้มีการสร้าง Environment อื่นใน Device Shadow จึงทำให้เทคนิคของการให้บริการที่มีอยู่ เช่น inpher, OpenMined และ Enveil ไม่สามารถใช้งานได้บน NETPIE ซึ่งในการใช้งานระบบของต่างประเทศดังกล่าวจำเป็นต้องมีการติดตั้ง Environment Engine ในคลาวด์ ดังนั้นหากต้องการนำมาประยุกต์ใช้กับ NETPIE จึงจำเป็นต้องทำการแก้ไข Device Shadow ใน NETPIE เพื่อให้รองรับการติดตั้ง Environment ต่างๆ ได้

ทั้งนี้เทคนิคการให้บริการดังกล่าวไม่มีปัญหาในการใช้งานกับคลาวด์ทั่วไปเช่น AWS และ Microsoft Azure เนื่องจาก AWS และ Microsoft Azure สามารถติดตั้ง Environment Engine ได้เช่นการเขียนโค้ดลงใน Lambda บน AWS หรือติดตั้ง Docker ใน Microsoft Azure เพื่อสร้าง HE Environment

บทที่ ๓ ระเบียบวิธีวิจัย/วิธีการและขั้นตอนการดำเนินงานวิจัย

โครงการวิจัยนี้มีเป้าหมายในการพัฒนาแพลตฟอร์มสำหรับการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT เพื่อนำไปให้บริการกับระบบ IoT ของผู้ประกอบการภาคอุตสาหกรรมเพื่อนำไปสู่ Smart Factory โดยในโครงการนี้ได้ใช้เทคโนโลยี HE สำหรับการป้องกันข้อมูลไอโอทีรั่วไหลขณะที่นำข้อมูลไปประมวลผลบนคลาวด์ของผู้ให้บริการภายนอกโรงงานอุตสาหกรรม โดยขั้นตอนการออกแบบจะกล่าวถึงหลักการในการเลือกชุด Code Library และ Compiler ที่ใช้ในการพัฒนาต่อยอด และการออกแบบให้ครอบคลุม Eco-system ของระบบ IoT เพื่อสอดคล้องกับกรณีการใช้งานทั่วไปในระบบอุตสาหกรรมไอโอที รวมทั้งการพัฒนาเพื่อให้การทำงานสอดคล้องกับกลุ่มผู้ใช้งาน NETPIE โครงการนี้จึงได้ศึกษาส่วนต่อขยายและเชื่อมต่อกับ NETPIE โดยขั้นตอนการศึกษาเริ่มจากศึกษาและพูดคุยกับเจ้าของโรงงานอุตสาหกรรมในประเทศไทยเพื่อหาหรือถึงกรณีศึกษาที่ต้องการใช้งานระบบไอโอทีภายในโรงงานอุตสาหกรรม

๓.๑ ศึกษาความต้องการใช้งานระบบไอโอทีภายในโรงงานอุตสาหกรรม

การเลือกโรงงานอุตสาหกรรมเป้าหมายและกรณีศึกษาของโรงงานอุตสาหกรรมที่ต้องการเข้าร่วมทดสอบมีหลักการในการคัดเลือกโดยคำนึงถึงปัจจัยความพร้อมของโรงงานคู่ทดสอบดังนี้

๑) ความพร้อมและศักยภาพของโรงงานอุตสาหกรรม เนื่องจากในสถานการณ์ปัจจุบัน เกิดสถานการณ์ Covid-19 ความพร้อมในการให้ทีมวิจัยเข้าพื้นที่เพื่อทำการสำรวจ และความพร้อมในการจัดสรรพื้นที่เพื่อทำการทดสอบงานวิจัยต้นแบบมีความสำคัญอย่างมาก โดยทีมวิจัยประเมินจากความต้องการและศักยภาพของโรงงานอุตสาหกรรมเป็นหลัก

๒) ความต้องการนำเทคโนโลยีการให้บริการคุ้มครองข้อมูลส่วนบุคคลไปใช้งานจริง โดยทีมวิจัยทำการประเมิน และสอบถามความต้องการเบื้องต้นจากเจ้าของโรงงานอุตสาหกรรม

๓) ความพร้อมในด้านความเข้าใจเทคโนโลยี IoT ของโรงงานอุตสาหกรรม ทีมวิจัยเลือกพิจารณาจากโรงงานอุตสาหกรรมที่เคยร่วมงานในโครงการวิจัยด้าน IoT ที่มีอยู่ และแนวโน้มจะร่วมงานกันในอนาคต เช่น โรงงานอุตสาหกรรมที่อยู่ในโครงการ IDA และโครงการ 5G Smart Factories

๔) ความพร้อมด้านระบบ IT ในโรงงานอุตสาหกรรม ทีมวิจัยเลือกพิจารณาจากโรงงานอุตสาหกรรมที่มีความพร้อมด้านระบบ IT ซึ่งโรงงานอุตสาหกรรมจะมีผู้เชี่ยวชาญด้าน IT ดูแลระบบที่ต้องการทดสอบได้

จากการศึกษาความต้องการใช้งานระบบ IoT ภายในโรงงานอุตสาหกรรมเป้าหมายและกรณีศึกษาของโรงงานอุตสาหกรรมที่ต้องการเข้าร่วมทดสอบตามหลักการที่ใช้คัดเลือกตามที่กำหนด โดยคำนึงถึงปัจจัยความพร้อมของโรงงานคู่ทดสอบ ทำให้เลือกโรงงานอุตสาหกรรมต้นแบบ เพื่อทำการวางระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT จำนวน ๒ โรงงาน ได้แก่

๓.๑.๑ โรงงาน Thanakorn Vegetable Oil Products Co., Ltd.

ในการหารือร่วมกับบริษัท Thanakorn Vegetable Oil Products Co., Ltd. ได้พูดคุยถึงประเด็นความสนใจในการใช้งานระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT โดยทางประธาน

บริษัท ได้ให้ความสนใจที่จะร่วมมือให้เป็นพื้นที่ทดสอบ โดยระบบที่มีความต้องการใช้ทดสอบ คือ ระบบ Condition-Based Monitoring ซึ่งเป็นระบบเปิดปิดวาล์วของไซโล (SILO) เก็บน้ำมันขนาดใหญ่ เนื่องจากระบบเปิด/ปิดวาล์วที่ใช้กันอยู่เดิมเป็นระบบ Manual ที่ต้องใช้พนักงานเข้าไปเดินตรวจสอบแรงดันของไซโลและระดับของเหลวในไซลอน้ำมัน และพนักงานทำการเปิด/ปิด ตามรอบการเดินตรวจพื้นที่ เนื่องจากการใช้คนในการตรวจวัดระดับน้ำมันในไซโล(ที่มีไม่ต่ำกว่า ๖๐ ทำให้เกิดการเหนื่อยล้าจากการทำงาน เสี่ยงต่อความไม่ปลอดภัยต่อพนักงาน นอกจากนั้นทางบริษัทมีแผนในการเพิ่มขีดความสามารถในการแข่งขันทั้งในและต่างประเทศ ไม่ว่าจะเป็นด้านเทคโนโลยีและบุคลากร เพื่อทดแทนการนำเข้าจากฟิซน้ำมัน และการจำหน่ายผลิตภัณฑ์ทั้งหมดของบริษัท ทางบริษัทจึงเล็งเห็นถึงความต้องการในการใช้เทคโนโลยีและนวัตกรรมไอโอทีเพื่อเพิ่มประสิทธิภาพการผลิต รวมถึงกระบวนการทำงาน ต้องการให้มีการทำงานอย่างมีประสิทธิภาพ รวดเร็ว มีระบบอัตโนมัติมาสนับสนุน

จากการหารือเบื้องต้นทางโรงงานมีความต้องการปรับปรุงการทำงานในส่วนของการตรวจจับระดับของเหลวในไซลอน้ำมันฟิซทั้งขนาดเล็กและใหญ่ด้วยเซนเซอร์แบบเรดาร์ในพื้นที่ฟาร์มไซลอน้ำมัน (Tank Farm) ทั่วทั้งโรงงาน โดยมีพื้นที่หลักดังที่แสดงในแผนที่ภาพถ่ายดาวเทียมของโรงงาน บริษัท ธนากรผลิตภัณฑ์น้ำมันฟิซ จำกัด (TVOP) ดังรูปที่ ๓๕

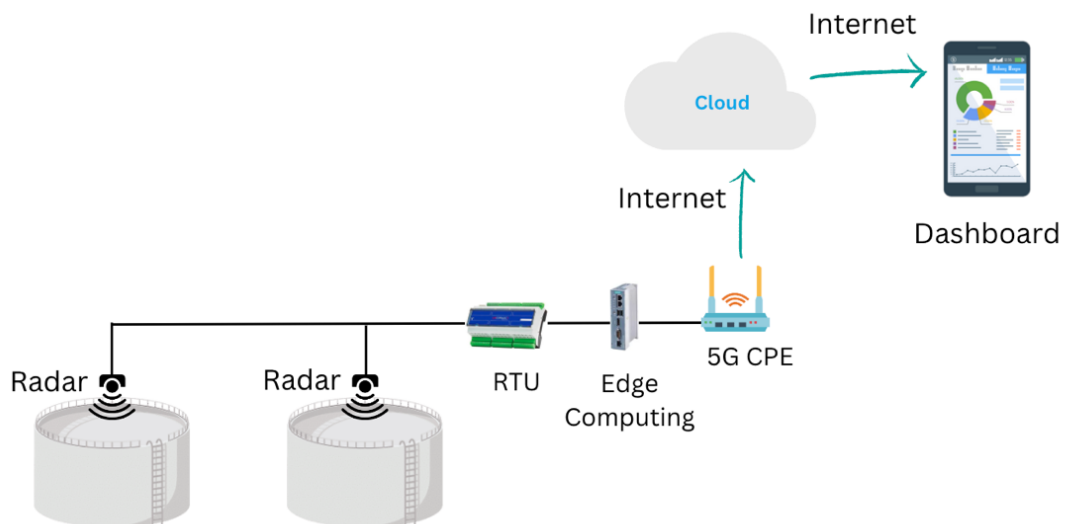


รูปที่ ๓๕ แสดงไซโลเก็บวัตถุดิบของโรงงาน บริษัท ธนากรผลิตภัณฑ์น้ำมันฟิซ จำกัด
(บริษัท ธนากรผลิตภัณฑ์น้ำมันฟิซ จำกัด, ม.ป.ป.)



รูปที่ ๓๖ พื้นที่โกดังน้ำมันพืช โกดังกากพืชน้ำมัน ฟาร์มไซโลน้ำมัน และส่วนที่ต้องการติดตั้งกล้อง CCTV ในบริเวณโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP)

ภายหลังจากการหารือกับโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP) และเข้าสำรวจพื้นที่หน้างานเพื่อพูดคุยกับทางวิศวกรผู้รับผิดชอบหน้างานแล้วจึงทำการออกแบบระบบ IoT สำหรับกรณีศึกษาดังรูปที่ ๓๗



รูปที่ ๓๗ ระบบ Condition-based Monitoring สำหรับ Smart Tank Farm

การออกแบบสำหรับกรณีนี้ อาศัยเซนเซอร์ Radar สำหรับตรวจวัดระดับน้ำมันในไซโลซึ่งข้อมูลจะถูกส่งผ่านโปรโตคอล HEART ๔-๒๐ mA ซึ่งเป็นข้อมูลอะนาล็อก เพื่อให้สามารถส่งต่อข้อมูลในรูปแบบ IoT ได้ต้องอาศัยการแปลง

โพรโทคอลดังกล่าวให้อยู่ในรูปของดิจิทัล โดยในที่นี้ โครงการฯ เลือกใช้ RTU ในการแปลงโพรโทคอล HEART ไปเป็น Modbus และอาศัยการแปลงโพรโทคอลจาก Modbus TCP ให้อยู่ในรูปของโพรโทคอล MQTT เพื่อสื่อสารการส่งข้อมูลไปยังคลาวด์ NETPIE โดยอุปกรณ์ Edge computing หรือ IoT Gateway ในที่นี้เพื่อความสะดวกในการพัฒนาโปรแกรมสำหรับวิศวกรในโรงงานที่คุ้นเคยกับโปรแกรม Node-RED ซึ่งเป็นโปรแกรมแบบ Low Code ที่มีเครื่องมือต่างๆ สำหรับการรองรับโพรโทคอลการสื่อสารแบบโอโอทีมาให้อยู่แล้ว ทำให้ง่ายต่อการนำไปใช้งาน ดังนั้นภายในอุปกรณ์ Edge computing นี้จึงได้ถูกติดตั้งโปรแกรม Node-RED ที่วิศวกรได้เขียนโปรแกรมการทำงานสำหรับรับค่าข้อมูลจาก RTU มาแปลงเป็นข้อความสำหรับส่งผ่านโพรโทคอล MQTT

อย่างไรก็ตาม หากลองเปรียบเทียบสถาปัตยกรรมระบบเครือข่าย IoT ของไทยที่ใช้กันทั่วไปในปัจจุบันการออกแบบระบบให้ IoT เซนเซอร์เชื่อมต่อคลาวด์ผ่านระบบ Wi-Fi มีข้อดีคือ ไม่จำเป็นต้องวางสายเคเบิลเพื่อเชื่อมต่อกับ Router/ IoT gateway ทำให้ประหยัดค่าใช้จ่ายในการดูแลระบบสายเชื่อมต่อ และการเชื่อมต่อมีความยืดหยุ่นสูงในการปรับเปลี่ยนจุดเชื่อมต่อเซนเซอร์ แต่การออกแบบเช่นนี้ยังมีข้อเสีย ดังนี้

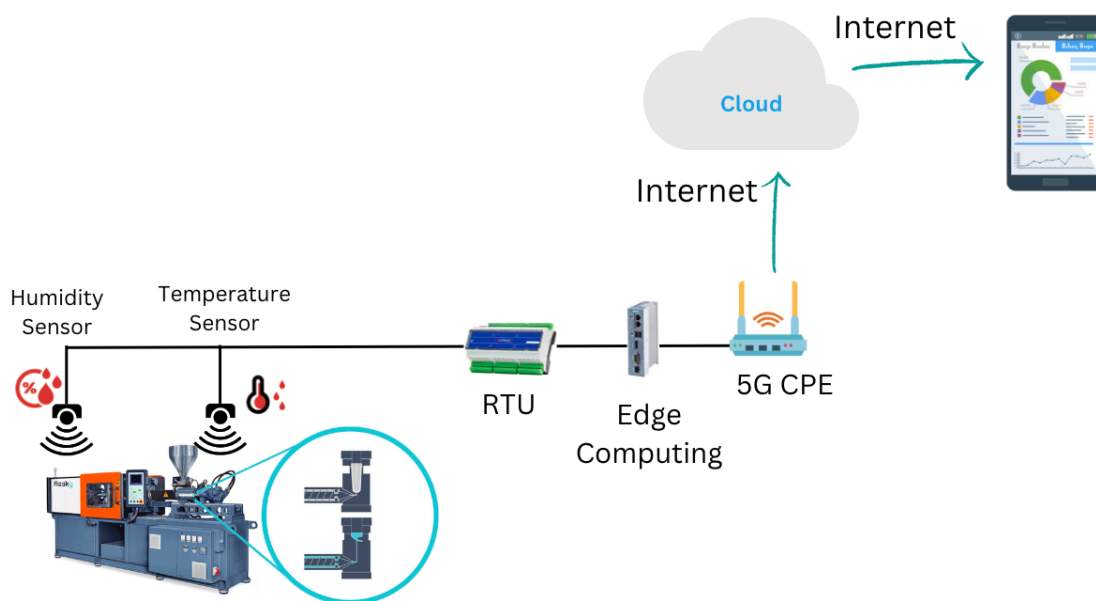
- การเชื่อมต่อ IoT เซนเซอร์ผ่านอุปกรณ์ Wi-Fi router อาจไม่ครอบคลุมทั่วทั้งพื้นที่ใช้งาน เช่น Tank farm site#1 อยู่ห่างจาก Tank farm site# ๒ และอาจมีจุดอับสัญญาณ Wi-Fi
- การเชื่อมต่อข้อมูลผ่านระบบ Wi-Fi ไม่ปลอดภัยทางไซเบอร์เนื่องจากระบบ Wi-Fi สามารถถูกดักฟังข้อมูลได้ง่าย
- ข้อมูลที่ส่งไปประมวลผลบนคลาวด์ถูกเปิดเผยให้กับคลาวด์เมื่อต้องนำไปประมวลผล ทำให้เกิดปัญหาความไม่เป็นส่วนตัวของข้อมูล หรือข้อมูลรั่วไหลสู่ภายนอก

ดังนั้น การพัฒนาแพลตฟอร์มในโครงการนี้จึงได้กำหนดให้มีการเข้ารหัสข้อมูลด้วยเทคโนโลยี HE ที่ Edge Computing และการสร้างสิ่งแวดล้อมการคำนวณแบบ HE บนคลาวด์ โดยที่ส่วนคลาวด์จะทำหน้าที่คำนวณตามโอเพอร์เรชันที่ผู้ใช้งานกำหนด จากนั้นผลลัพธ์ที่ได้จากการคำนวณภายใต้ข้อมูลที่เข้ารหัสจะถูกส่งมาถอดรหัสที่ส่วนแสดงผล ในโครงการนี้ได้ออกแบบให้ส่วนถอดรหัสอยู่ใน Mobile Application และแสดงผลบน Mobile Application แบบเวลาจริง (Real Time)

๓.๑.๒ โรงงาน บริษัท เดอะ เพ็ท จำกัด

ในการหารือร่วมกับบริษัท เดอะ เพ็ท จำกัด ได้พูดคุยถึงประเด็นความสนใจในการใช้งานระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT กับหัวหน้าวิศวกรของโรงงาน โดยทางโรงงานมีกรณีศึกษาสำหรับการติดตั้งระบบโอโอทีอยู่เดิมเป็นระบบวัดอุณหภูมิ (Temperature) และความชื้น (Humidity) ด้วยเซนเซอร์ในระบบโรงงานอุตสาหกรรมการผลิตขวดพลาสติกเพื่อคำนวณค่า Dew Point Temperature ในระบบอุตสาหกรรม โดยระบบเดิมนั้นทางโรงงานได้พัฒนาการดึงข้อมูลจากเซนเซอร์ของเครื่องจักรด้วยโปรแกรม Node-RED ใน Edge computing และส่งค่าดังกล่าวไปจัดเก็บเพื่อแสดงผลบน NETPIE ดังรูปที่ ๓๘ จากการหารือแนวทางความร่วมมือเบื้องต้นทางโรงงานพบปัญหาของระบบ IoT ที่ดำเนินการอยู่เดิม คือ

- ข้อมูลของอุปกรณ์ IoT สำหรับตรวจเช็คอุณหภูมิ (Temperature) และความชื้น (Humidity) ด้วยเซนเซอร์ในระบบโรงงานอุตสาหกรรมการผลิตขวดพลาสติกเพื่อคำนวณค่า Dew Point Temperature ในระบบอุตสาหกรรมผ่านเครือข่าย Internet ถูกดักฟังได้ง่าย และต้องเปิดเผยข้อมูลให้คลาวด์
- เดิมวิศวกรโรงงานได้พัฒนาสูตรการคำนวณบน Node-RED ที่ถูกติดตั้งใน Edge Computing และหากมีการแก้ไขสูตร หรือต้องการเพิ่มเติมเซนเซอร์บนโพล์การทำงานใน Node-RED ทำให้ข้อมูลอื่นๆ หยุดชะงัก ส่งผลกระทบกับการทำงานของระบบในโรงงาน และระบบในการดึงค่าเซนเซอร์ในโรงงาน



รูปที่ ๓๘ ระบบ Condition-based Monitoring สำหรับระบบวัดอุณหภูมิ (Temperature) และความชื้น (Humidity) ของเครื่องจักร

ดังนั้นการพัฒนาแพลตฟอร์มในโครงการนี้จึงได้กำหนดให้มีการเข้ารหัสข้อมูลด้วยเทคโนโลยี HE ที่ Edge Computing และการจัดการเปลี่ยนแปลงสูตรการคำนวณบนคลาวด์ โดยอิสระจากส่วนการรวบรวมข้อมูลที่ Edge Computing ผลลัพธ์ที่ได้จากการคำนวณภายใต้ข้อมูลที่เข้ารหัสจะถูกส่งมาถอดรหัสที่ส่วนแสดงผล ในโครงการนี้ได้ออกแบบให้ส่วนถอดรหัสอยู่ใน Mobile Application และแสดงผลบน Mobile Application แบบเวลาจริงโดยผู้ควบคุมข้อมูลที่มีสิทธิ์ในการเข้าถึงข้อมูลเท่านั้นที่สามารถตรวจสอบข้อมูลได้

๓.๒ ศึกษาการทำงานของ Code Library และ Compiler ของ HE เพื่อการพัฒนาต่อยอด

จากการศึกษากรณีศึกษาของโรงงานอุตสาหกรรมที่ต้องการใช้งานระบบโอโอทีนั้น พบว่าการคำนวณข้อมูลโอโอทีจะมีหลากหลายสูตรคำนวณ ทั้งนี้ขึ้นอยู่กับกรณีที่จะนำข้อมูลเหล่านั้นไปใช้งาน อีกทั้งข้อมูลโอโอทีที่ต้องการนำไปใช้อยู่ในรูปแบบของจำนวนจริง (Real Number) ดังนั้นในโครงการนี้จึงเลือกใช้เทคนิคการเข้ารหัส RNS-CKKS ที่สามารถรองรับการเข้ารหัสบนข้อมูลจริงได้ โดยอาศัยเทคนิคการทำ Encoding จำนวนจริงให้อยู่ในรูป

ของจำนวนเต็มบวกก่อน จากนั้นจึงนำข้อมูลที่ถูก Encode แล้วมาทำการ Encrypt เพื่อให้ได้ข้อมูลที่เข้ารหัสแบบ HE โดยชุด Code Library และ Compiler ที่รองรับเทคนิคการเข้ารหัสแบบ RNS-CKKS มีด้วยกันดังตารางที่ ๙

ตารางที่ ๙ ชุด Code Library และ Compiler ที่รองรับเทคนิคการเข้ารหัสแบบ RNS-CKKS

FHE Library	โปรแกรมภาษาที่ใช้	Compiler
Microsoft SEAL	C++	Node SEAL, EVA, CHET, SEALion, nGraph-HE, Marble
Lattigo	Go	Lattigo
HEAAN	C++	CHET

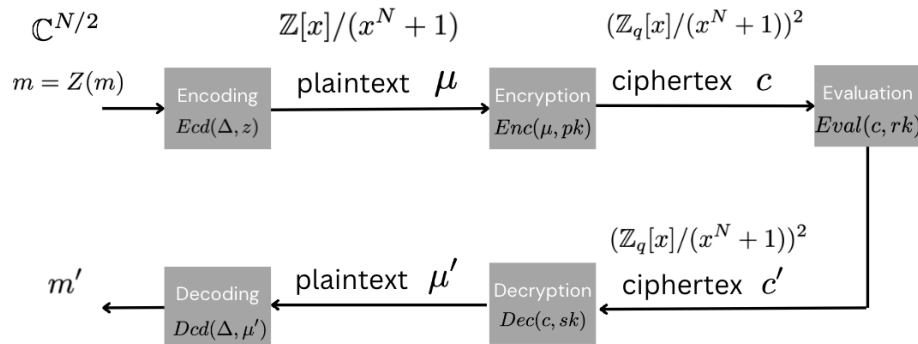
อย่างไรก็ตาม Code Library ดังกล่าวอยู่ในรูปแบบของ Low level programming language ที่นักพัฒนาไอโอทีต้องอาศัยความสามารถในการพัฒนาโค้ดเอง และการกำหนดพัฒนาต่อยอดสำหรับนักพัฒนาไอโอที หรือวิศวกรในโรงงานอุตสาหกรรมเป็นเรื่องยาก นอกจากต้องคำนึงถึงการเขียนโค้ดที่ถูกต้องโดยไม่มีบั๊ก ยังต้องมีความรู้ความเข้าใจในการเลือกกำหนดพารามิเตอร์ HE ที่เหมาะสมกับการใช้งานแต่ละกรณีเพื่อให้คงความปลอดภัยและมีประสิทธิภาพใช้งานได้ในทางปฏิบัติ ทำให้การเข้าถึงเทคโนโลยี HE เป็นเรื่องยาก ดังนั้นโครงการนี้ได้พัฒนาให้แพลตฟอร์มมีการใช้งานได้ง่ายและเหมาะสมกับนักพัฒนาไอโอทีและวิศวกรโรงงานที่คุ้นเคยกับการใช้งานโปรแกรม Node-RED อยู่แล้วใช้งานได้ง่าย

โครงการนี้เลือกพัฒนาต่อยอดบน Code Library ของ Microsoft SEAL ที่อยู่บนพื้นฐานของภาษา C++ โดยจากการศึกษาพบว่าได้มีผู้พัฒนา Compiler Node SEAL ซึ่งทำให้ Code Abstract มากขึ้นด้วยโปรแกรม NodeJS ซึ่งเป็นพื้นฐานโค้ดของโปรแกรม Node-RED อยู่แล้ว ดังนั้นในโครงการนี้จึงสามารถนำชุด Compiler Node SEAL มาพัฒนาต่อยอดบน Node-RED เพื่อใช้เป็นเครื่องมือในการปฏิบัติการฟังก์ชัน HE

๓.๓ ออกแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอที

การออกแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอที เริ่มต้นจากการพิจารณาส่วนโมดูลต่างๆ ของ RNS-CKKS ของ Microsoft SEAL โดยได้อะแกรมการทำงานแสดงดังรูปที่ ๓๙ โดยการเลือกพารามิเตอร์สำหรับ RNS-CKKS มีพารามิเตอร์ที่เกี่ยวข้อง ๒ ส่วนคือ พารามิเตอร์สำหรับการ Encode และพารามิเตอร์สำหรับการ Encrypt โดยเมื่อข้อมูลเลขจำนวนจริงถูก Encode เพื่อทำให้กลายเป็น Plaintext ที่ช่วยเปลี่ยนให้เลขที่มีทศนิยมสามารถทำการเข้ารหัสแบบ Homomorphic ได้ โดยพารามิเตอร์สำหรับการเข้ารหัสจะเปลี่ยนข้อมูล Plaintext ให้เป็น Ciphertext ด้วยกุญแจเข้ารหัส ในขั้นตอนการคำนวณหรือทำโอเปอเรชันการคำนวณ นั้นจะนำเอากุญแจ Relinear Key (rk) มาช่วยลดขนาดของข้อมูลเนื่องจากภายหลังการทำโอเปอเรชันการคูณแล้วขนาดของข้อมูลจะมีขนาดโตขึ้นทำให้ไม่สามารถนำไปคำนวณต่อได้ จากนั้นผลลัพธ์จากการคำนวณที่อยู่ในรูปของ Ciphertext จะถูกส่งไปถอดรหัสด้วยกุญแจสำหรับถอดรหัส Secret Key (sk) และผลที่ได้จากการถอดรหัสจะได้ Plaintext ดังนั้นเพื่อให้ได้ข้อมูลจำนวนจริงกลับมา Plaintext จะถูกนำไป Decode โดยผลของความถูกต้องของผลลัพธ์ที่ได้ภายหลังจากการ Decode จะขึ้นอยู่กับ การเลือกพารามิเตอร์สำหรับการ Encode และ Encrypt โดยสามารถตรวจสอบความคลาดเคลื่อนได้จากความ

แตกต่างจากการคำนวณบนจำนวนจริงและผลลัพธ์ที่ได้จากการ Decode ดังนั้น พารามิเตอร์ที่เกี่ยวข้องในกระบวนการ HE ประกอบด้วย พารามิเตอร์ Polymodulus N , พารามิเตอร์ Scale Factor Δ , Parameter Coefficient Modulus q



รูปที่ ๓๙ ไต่อะแกรมการทำงานของ RNS-CKKS

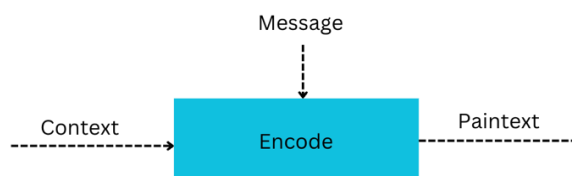
เมื่อพิจารณาโมดูลการทำงานด้วย Microsoft SEAL จะสามารถแบ่งการทำงานออกตามโมดูลต่างๆ ดังรูปที่ ๔๐ ถึง ๔๕ ดังนั้น จะเห็นได้ว่าเราสามารถแยกส่วนประกอบของการทำงานของแต่ละโมดูลออกจากกันและนำมาจัดเรียงในส่วนต่างๆ ตามแพลตฟอร์มที่เราต้องการออกแบบ



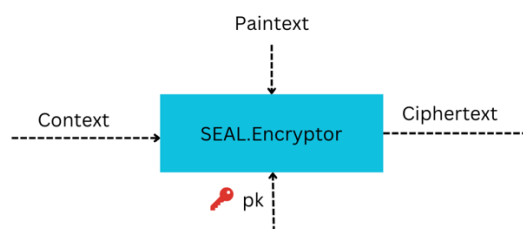
รูปที่ ๔๐ ไต่อะแกรมการสร้างพารามิเตอร์ Context



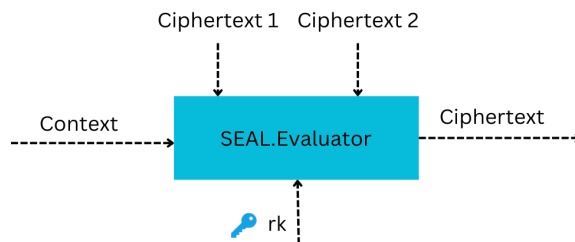
รูปที่ ๔๑ ไต่อะแกรมการสร้างชุดกุญแจสำหรับ RNS-CKKS



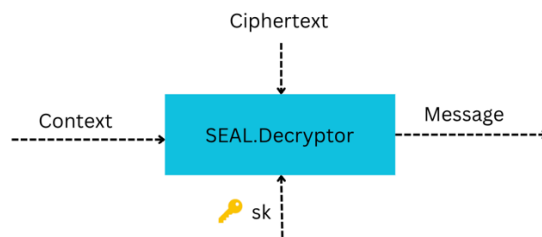
รูปที่ ๔๒ ไต่อะแกรมการ Encode ข้อความ Message



รูปที่ ๔๓ ไดอะแกรมการเข้ารหัส



รูปที่ ๔๔ ไดอะแกรมการคำนวณด้วยข้อมูลที่ถูกรหัส



รูปที่ ๔๕ ไดอะแกรมการถอดรหัส

โดยองค์ประกอบของแพลตฟอร์มบริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอทีประกอบด้วย ๓ ส่วน

- ๑) Edge Computing: ใน Edge Computing จะดำเนินการฟังก์ชันการ Encode ดังรูปที่ ๔๒ และ ฟังก์ชันการเข้ารหัสดังรูปที่ ๔๓
- ๒) Cloud Computing: ใน Cloud Computing จะดำเนินการฟังก์ชันการคำนวณดังรูปที่ ๔๔
- ๓) Mobile Application: ในส่วน Mobile Application จะเป็นส่วนที่ติดต่อกับผู้ใช้งานและเป็นผู้ควบคุมสิทธิ์ในการเข้าถึงข้อมูล ดังนั้นจะดำเนินการฟังก์ชันการกำหนดค่าพารามิเตอร์ RNS-CKKS เพื่อส่งต่อพารามิเตอร์ในรูปแบบของ Context ให้ฟังก์ชันอื่นๆ ดังรูปที่ ๔๐ และผู้ใช้งานจะทำหน้าที่ดำเนินการสร้างกุญแจด้วยฟังก์ชันสร้างกุญแจดังรูปที่ ๔๑ กุญแจและ Context จะถูกส่งต่อไปยังฟังก์ชันต่างๆ นอกจากนี้ใน Mobile Application ยังมีส่วนที่ทำหน้าที่ฟังก์ชันถอดรหัสดังรูปที่ ๔๕ เพื่อนำมาแสดงผลให้กับผู้ใช้งาน

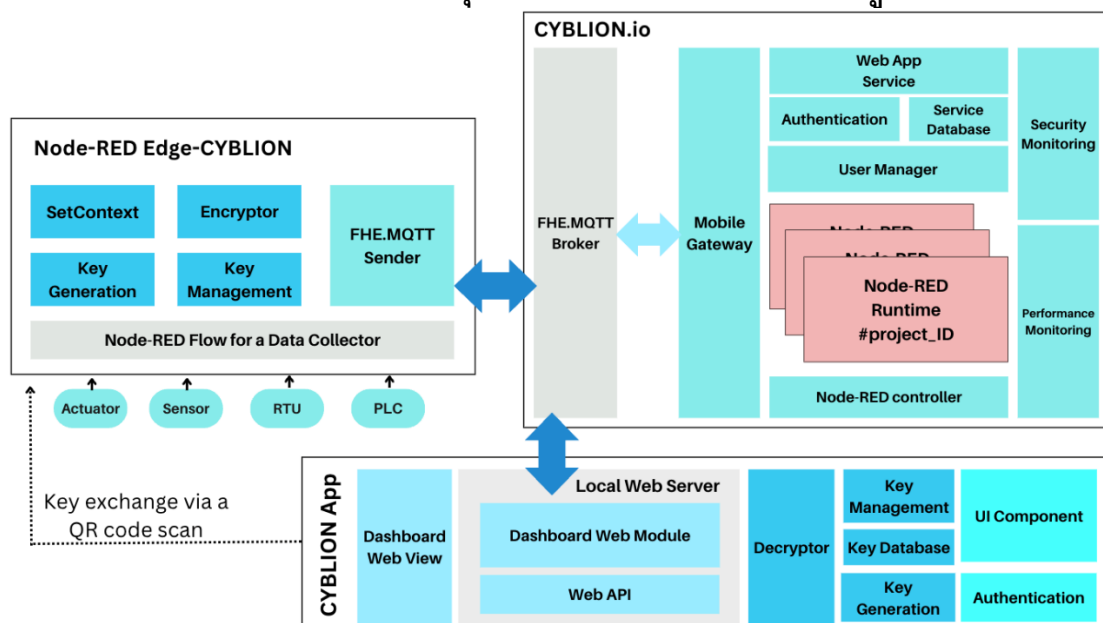
จากการวางองค์ประกอบของฟังก์ชันการทำงานของ RNS-CKKS ในส่วนต่างๆ ของระบบไอโอทีแล้วในการออกแบบแต่ละส่วนได้คำนึงถึงการใช้งานตามฟังก์ชันที่สอดคล้องกับ Code Library ของ Microsoft SEAL และยังคำนึงผู้ใช้งานเป็นหลัก โดยการออกแบบแพลตฟอร์มมีหลักการดังนี้

- คำนึงถึงความง่ายต่อการนำไปใช้งานในอุปกรณ์ Edge Computing ที่เหมาะสมกับวิศวกรโรงงาน และนักพัฒนาไอโอที
- ยืดหยุ่นต่อการปรับเปลี่ยนฟังก์ชันการคำนวณให้เป็นไปตามกรณีที่ต้องการใช้งาน
- ขยายได้เพื่อรองรับจำนวนผู้ใช้งานจำนวนมากและข้อมูลจำนวนมาก
- ประสิทธิภาพในการประมวลผลการเข้ารหัสและถอดรหัสบนอุปกรณ์ที่มีหน่วยประมวลผลขนาดเล็ก

- ความสวยงามน่าใช้ของ Mobile Application และเข้ากันได้กับอุปกรณ์สมาร์ตโฟนทั้งระบบ iOS และ Android
- ความปลอดภัยในการยืนยันตัวตน ความปลอดภัยในการเข้ารหัสข้อมูล และการแลกเปลี่ยนกุญแจระหว่างอุปกรณ์

จากการออกแบบแพลตฟอร์มดังกล่าว ทีมวิจัยได้เลือกพัฒนาส่วนโปรแกรมสำหรับทำงานบน Edge Computing ด้วยโปรแกรม Node-RED ให้ใช้งานร่วมกับ Edge Computing ที่โรงงานใช้งาน โดยไม่ต้องเพิ่มอุปกรณ์ Edge Computing ใหม่ ส่วนฟังก์ชันการทำงานบนคลาวด์ ได้เลือกพัฒนามบนโปรแกรม Node-RED เช่นกัน ผู้ใช้งานจึงมีความคุ้นเคย ทั้งนี้การพัฒนา Node-RED สำหรับผู้ใช้งานบนคลาวด์จะอยู่ใน Environment ส่วนตัวแยกจากผู้ใช้งานอื่นๆ และส่วนโปรเจ็คอื่นๆ ทางโครงการนี้จึงเลือกใช้ Docker ในส่วน Mobile Application ได้พัฒนามบน Xamarin.Forms เพราะเป็น Cross Platform และรองรับการทำงานของ Microsoft SEAL Library ทำให้การพัฒนา Mobile Application บนระบบปฏิบัติการ iOS สามารถแชร์ Code กันได้กับระบบปฏิบัติการ Android ในส่วนการยืนยันตัวตน ทีมงานได้ออกแบบให้ยืนยันตัวตนผ่าน NETPIE Account และระบบการยืนยันตัวตนของ Device ในแพลตฟอร์ม โดยยืนยันตัวตนของ Device ใน NETPIE ทำให้ผู้ใช้งานสามารถตรวจสอบข้อมูลของ Project และ Device ผ่าน NETPIE ได้ รวมทั้งออกแบบให้ผู้ใช้งานสามารถสร้างชุดกุญแจบน Mobile Application และการแลกเปลี่ยนกุญแจเข้ารหัสและกุญแจ Relinear สำหรับการคำนวณบน Mobile Application เพื่อลดความซับซ้อนในการตรวจสอบ Integrity ของกุญแจสำหรับเข้ารหัสบน Edge Computing โครงการนี้ยังได้ออกแบบการแลกเปลี่ยนกุญแจเข้ารหัสระหว่าง Mobile Application และ Edge Computing ผ่านการสแกน QR code และการส่งกุญแจ Relinear ไปยังคลาวด์ผ่าน HTML โดยคลาวด์มีความสามารถในการตรวจสอบค่า Hash ของกุญแจและตรวจเช็คกับ ID ของโปรเจ็คที่ต้องการแลกเปลี่ยนได้

๓.๔ การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับ IoT



รูปที่ ๔๖ สถาปัตยกรรมแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที

การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีเป็นไปตามการออกแบบตามฟังก์ชันการทำงานบนพื้นฐานของ RNS-CKKS สำหรับ HE บน Microsoft SEAL Library ซึ่งแผนการพัฒนาดังกล่าวถูกเสนอขึ้นสำหรับโครงการนี้ในรูปแบบของสถาปัตยกรรมดังรูปที่ ๔๖ โดยแต่ละส่วนจะประกอบไปด้วยฟังก์ชันการทำงานสำหรับ ของ RNS-CKKS และส่วนเชื่อมต่อระหว่างอุปกรณ์ และส่วนส่งต่อข้อมูล โดยการพัฒนาแต่ละส่วนมีรายละเอียดดังนี้

- ๑) Edge Computing: ส่วนนี้ถูกพัฒนาฟังก์ชันสำหรับการทำ HE บนโปรแกรม Node-RED โดยสร้าง Node สำหรับการใช้งานเพิ่มเติมใน Node-RED extension ที่รองรับการรับพารามิเตอร์ Context สำหรับ RNS-CKKS และ กุญแจ pk สำหรับเข้ารหัสข้อมูล ส่วนการรับข้อมูลพารามิเตอร์ Context และ กุญแจ pk ได้ถูกพัฒนาให้รับผ่านการสแกน QR Code โดย ในขั้นตอนการ Scan QR code ประกอบด้วย QR code จะส่งข้อมูล IP address ของ อุปกรณ์ Edge Computing ให้กับ Mobile Application เพื่อทำการส่ง Request ผ่าน HTML หลังจากนั้นข้อมูล Context สำหรับ RNS-CKKS และ กุญแจ pk จะถูกส่งจาก Mobile Application ไปยัง Node ที่รับข้อมูลใน Node-RED บน Edge computing ภายหลังการส่งข้อมูลเรียบร้อยแล้ว Edge Computing จะสามารถเข้ารหัสข้อมูลที่ถูกรวบรวมมาด้วย Node-RED Flow ก่อนหน้า ส่วนของ Node การเข้ารหัสข้อมูลจะที่อยู่ปลายของ Node-RED Flow และข้อมูลที่ถูก Encrypt จะถูกส่งผ่าน Message MQTT ไปยัง MQTT broker ที่รองรับข้อมูลชนิด HE
- ๒) Cloud Computing: การพัฒนาส่วนนี้แบ่งเป็นสองส่วนย่อย ได้แก่ ส่วน MQTT broker เนื่องจาก MQTT broker เดิมของ NETPIE ไม่สามารถรองรับไฟล์ข้อมูลขนาดใหญ่ได้ โครงการนี้จึงได้พัฒนาส่วน MQTT broker ขึ้นมาใหม่ โดย MQTT broker นี้จะรองรับการ publish/subscribe ข้อมูลขนาดใหญ่ ชนิด HE ได้ ส่วนถัดมาเป็นส่วนฟังก์ชันการเข้าถึงของผู้ใช้งาน ในโครงการนี้ได้พัฒนาให้ผู้ใช้งานเข้าถึงการจัดการสร้างสูตรคำนวณบนโปรเจ็คไอโอทีของตนเองผ่าน Web Application โดยภายใต้ Web Application นั้นจะส่งคำสั่งไปเรียกใช้โปรแกรม Node-RED ที่มี Node-RED extension สำหรับการรับข้อมูลและการทำโอเพอร์เรชันการคำนวณ การเชื่อมต่อระหว่าง Cloud, Edge Computing และ Mobile Application แต่ละส่วนนี้ต้องมีตัวกลางทำหน้าที่ติดต่อกับส่วนต่างๆ ทั้งภายในและภายนอก คลาวด์ ดังนั้นจึงกำหนดให้พัฒนา Mobile Gateway ทำหน้าที่เชื่อมต่อประสานการทำงานกับทุกส่วนทั้งภายนอกและภายในคลาวด์ การพัฒนาส่วนคลาวด์เพื่อให้สามารถขยายได้ตามจำนวนผู้ใช้งาน โครงการนี้จึงเลือกพัฒนามบน Kubernetes K8 โดยมีระบบการจัดการแบบ Namespace
- ๓) Mobile Application: การพัฒนาในส่วนนี้เป็นส่วน User Interface ทำหน้าที่เชื่อมต่อกับผู้ใช้งาน โดยพัฒนาอยู่บน Xamarin.Forms ที่รองรับการทำงานของ Microsoft SEAL Library ทั้งนี้การทำงานในส่วนของการจัดการฟังก์ชัน HE จะติดต่อกับส่วน Mobile Gateway เพื่อทำหน้าที่เชื่อมต่อไปยัง MQTT broker ที่รองรับ publish/subscribe ข้อมูลขนาดใหญ่ชนิด HE ในส่วนแสดงผลบน Dashboard ได้เลือกให้พัฒนาขึ้นในรูปแบบของ WebView ซึ่งเป็นการทำให้ Web Application สามารถรันอยู่บน

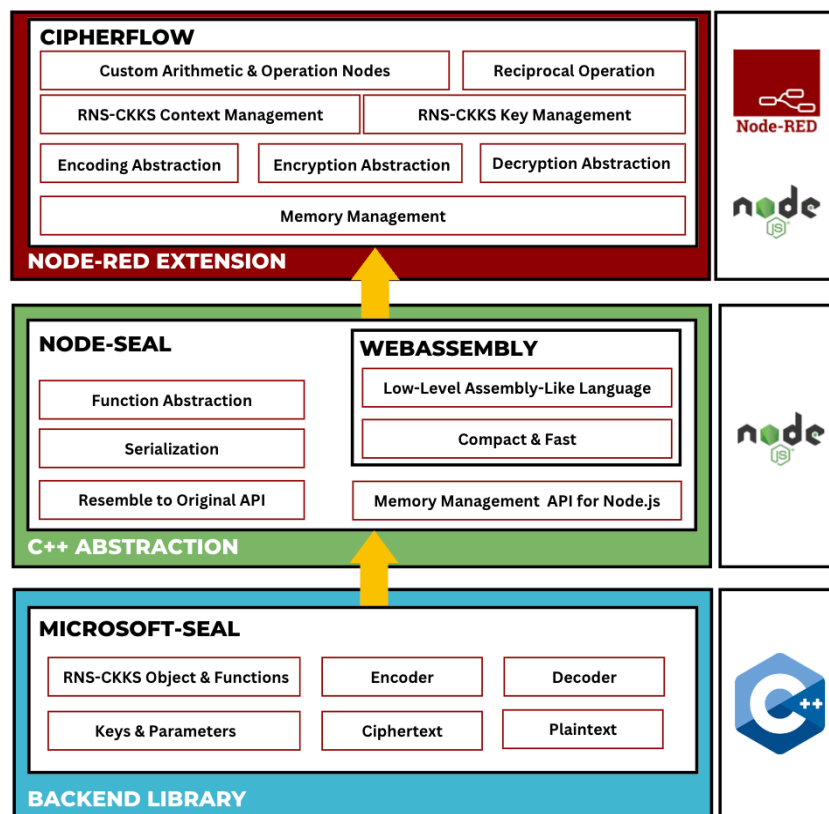
Mobile Application ได้ การเชื่อมต่อส่วนนี้กระทำผ่าน API ที่พัฒนาขึ้นเพื่อเป็นตัวกลางระหว่าง Xamarin.Forms และ WebView ส่วนการแสดงผลถูกพัฒนาให้ผู้ใช้สามารถเลือกรูปแบบของการนำเสนอข้อมูลและข้อมูลที่ต้องการนำเสนอได้ การจัดการกุญแจ (Key Management) ถูกพัฒนาให้จัดเก็บกุญแจไว้ในส่วน Key Chain ของสมาร์ทโฟนที่มีความปลอดภัยโดยไม่จัดเก็บ API Key ในชั้นของ Application การพัฒนาในส่วน UI ของ Mobile Application ใช้การออกแบบ Wireflow ก่อนการพัฒนาจริงด้วยโปรแกรม Adobe XD และทำการ Export ออกมาให้อยู่ในฟอร์แมตของ Xamarin.Forms ทั้งในส่วน of Android และ iOS

บทที่ ๔

การสร้างแพลตฟอร์ม ผลการวิจัย และผลการติดตั้งใช้งานจริง

๔.๑ การสร้างแพลตฟอร์มบริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT

แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอทีถูกพัฒนาขึ้นตามระเบียบวิธีการออกแบบในบทที่ ๓ โดยส่วนที่เป็นหัวใจหลักของการพัฒนาเครื่องมือสำหรับการใช้งาน HE แบบ RNS-CKKS นี้ คือ การพัฒนา Software Toolkit บน Node-RED ซึ่งการพัฒนาที่มีความท้าทายเนื่องจาก Microsoft SEAL Library ไม่ได้มีส่วนเชื่อมต่อกับโปรแกรม Node-RED ได้โดยตรง ดังนั้น โครงการนี้จึงได้นำเสนอสถาปัตยกรรมการพัฒนา RNS-CKKS บน Microsoft SEAL โดยเพิ่มชั้นของ NodeJS abstraction ที่ใช้ Web Assembly ในการรวบรวมเอา Code Microsoft SEAL Library ส่วน Backend ไว้ และทำให้สามารถพัฒนา Node ที่ทำงานฟังก์ชัน HE ใน Node-RED บน NodeJS ดังรูปที่ ๔๗

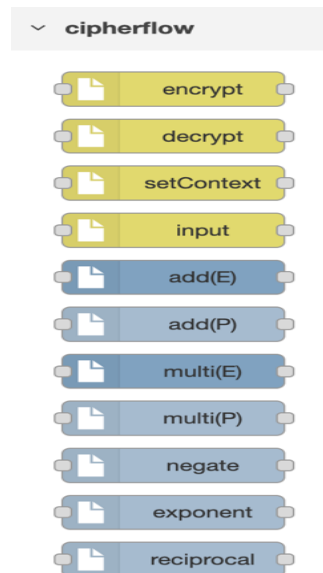


รูปที่ ๔๗ สถาปัตยกรรมของ Node-RED CipherFlow

๔.๑.๑ Software Toolkit บน Node-RED (CipherFlow)

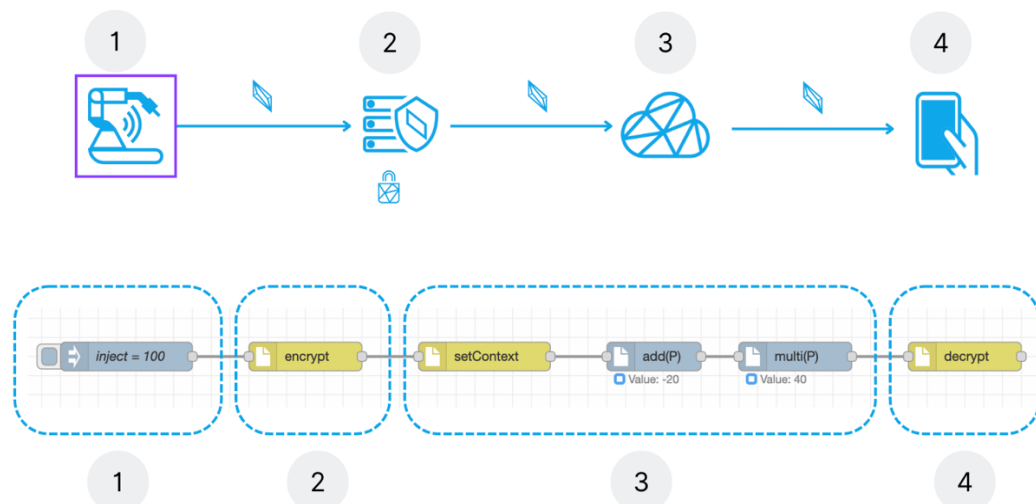
โครงการนี้พัฒนาส่วน Software Toolkit บน Node-RED ที่รวบรวม Node ที่เป็นเครื่องมือในการใช้สำหรับการเรียนรู้การใช้งาน HE ฟังก์ชันต่างๆ บน Node-RED โดยใช้ชื่อว่า CipherFlow ซึ่งถูกพัฒนาให้เป็น Playground เพื่อให้ผู้ใช้งานคุ้นเคยกับการใช้ HE ฟังก์ชันบน Node-RED และเรียนรู้ในการสร้างสูตรการคำนวณ และตรวจสอบความถูกต้องของการเลือกตั้งค่าพารามิเตอร์ที่เหมาะสมกับระบบไอโอทีที่ต้องการใช้งาน โดยภายใน

CipheFlow ประกอบด้วย Node ต่างๆ ดังนี้ Node ในกลุ่มการทำ HE ฟังก์ชัน ซึ่งใช้สัญลักษณ์สีเหลือง ได้แก่ encrypt, decrypt, setContext, และ input และ Node ในกลุ่มการทำโอเปอร์เรชันการคำนวณ ได้แก่ add(E), add(P), multi(E), multi(P) negate, exponent, และ reciprocal ดังรูปที่ ๔๘



รูปที่ ๔๘ Node-RED-Cipherflow

การใช้งาน Node เป็นดังตัวอย่างรูปที่ ๔๙ ผู้ใช้งานสามารถดาวน์โหลดได้จากการเลือก Node-RED-cipherflow ใน Node-RED Pallet ตัวอย่างแสดงให้เห็น Node ที่สื่อให้เห็นถึงส่วนต่างๆ ในระบบไอโอที นั่นคือ ส่วนที่ ๑ เป็นส่วนที่แสดงถึงการป้อนข้อมูลจาก Sensor เข้าไปสู่ Edge Computing ที่ติดตั้ง Node Encrypt ใน ส่วนที่ ๒ และในส่วนที่ ๓ เป็นส่วนที่ทำหน้าที่รับข้อมูลที่เข้ารหัสจาก Edge Computing มาคำนวณโดยมี Node add(P) และ multi(P) จากนั้นข้อมูลที่ผ่านการคำนวณแล้วจะถูกส่งไปถอดรหัสในส่วนที่ ๔ ด้วย Node Decrypt โดยตัวอย่างสูตรการคำนวณ คือ นำค่าจากเซนเซอร์ที่ถูกเข้ารหัสมาทำการบวกด้วยค่าคงที่ค่าหนึ่ง และหลังจากนั้น จะถูกนำไปคูณกับค่าคงที่ค่าหนึ่ง โดยข้อความใต้ Node แสดงถึงค่าคงที่ที่ใช้สำหรับ Node นั้นๆ การทำเช่นนี้จะช่วยให้ผู้ใช้งานสามารถตรวจสอบค่าคงที่ที่ตั้งไว้ได้โดยไม่ต้องคลิกเข้าไปใน Node เพื่อตรวจสอบ



รูปที่ ๔๙ ตัวอย่างการใช้งาน Node-RED Cipherflow

การเลือกค่าพารามิเตอร์สำหรับ Node-RED-cipherflow นั้น สามารถทำได้จาก Node encrypt, Node setContext และ Node decrypt ซึ่ง การทำเช่นนี้เปรียบเสมือนการส่งต่อค่า Context ที่เก็บข้อมูลพารามิเตอร์ของ RNS-CKKS ไว้ให้กับฟังก์ชัน HE เพื่อสร้างสภาพแวดล้อม HE ที่ตรงกัน ดังไดอะแกรมในรูปที่ ๕๐ ถึง ๕๒ เนื่องจากการเลือกค่าพารามิเตอร์สำหรับ RNS-CKKS มีความซับซ้อน ซึ่งทำให้ผู้ใช้งาน Microsoft SEAL Library โดยตรงนั้นจำเป็นต้องมีความเข้าใจในวิทยาการเข้ารหัสแบบ RNS-CKKS การกำหนดค่าพารามิเตอร์จำเป็นต้องเป็นไปตามมาตรฐานความปลอดภัยทางไซเบอร์ ดังตารางที่ ๑๐

ตารางที่ ๑๐ การกำหนดพารามิเตอร์สภาพแวดล้อม HE สำหรับการเข้ารหัสแบบ CKKS/RNS-CKKS ตามมาตรฐานความปลอดภัย [78]

n	Bit-length of coefficient modulus 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

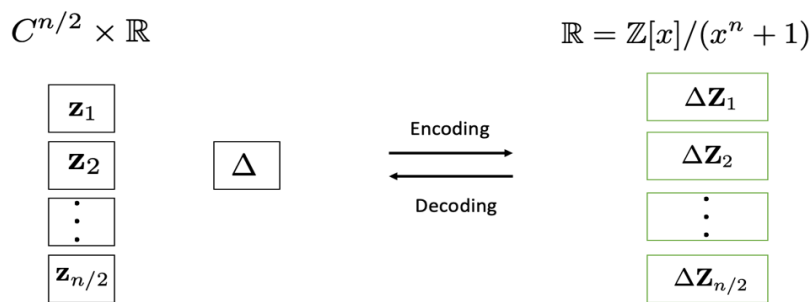
การกำหนดพารามิเตอร์สภาพแวดล้อม HE สำหรับการเข้ารหัสแบบ CKKS/RNS-CKKS กำหนดด้วยพารามิเตอร์ดังนี้ พารามิเตอร์ความปลอดภัยของข้อมูลขึ้นอยู่กับจำนวนบิตที่ใช้ในการสร้างพารามิเตอร์ $\lambda = 128$ บิตเป็นพารามิเตอร์ที่ใช้อธิบาย security-level ของการเข้ารหัสข้อมูลตามมาตรฐานความปลอดภัยของการเข้ารหัส หมายถึงความพยายามของผู้โจมตีที่ต้องใช้เวลาในการคำนวณ ๒,๑๒๘ ครั้งในการ Break security โดย

การเลือกพารามิเตอร์ให้เป็นไปตามมาตรฐานความปลอดภัย security-level 128 นั้นต้องทำการเลือกดีกรีของ Ring ที่ใช้กำหนดโดยค่าโพลีโมดูลัส n ที่โพลีโนเมียล $x^n + 1$ พร้อมกับค่าโคเอฟฟิเชียนสูงสุดที่ใช้ได้ $Q = \prod_{i=1}^L q_i$ ดังตารางที่ ๑๐ ทั้งนี้ในการกำหนดค่า L ขึ้นอยู่กับจำนวนครั้งของความต้องการทำโอเพอร์เรชันการคูณ โดยการเลือกค่า q_i มีผลต่อความแม่นยำในการคำนวณการเข้ารหัส และการกำหนดค่า Scale Factor Δ เพื่อช่วยในการกำหนดขีดและหัดของขอบเขตของพื้นที่ข้อมูล (Budget) สำหรับการ Encode

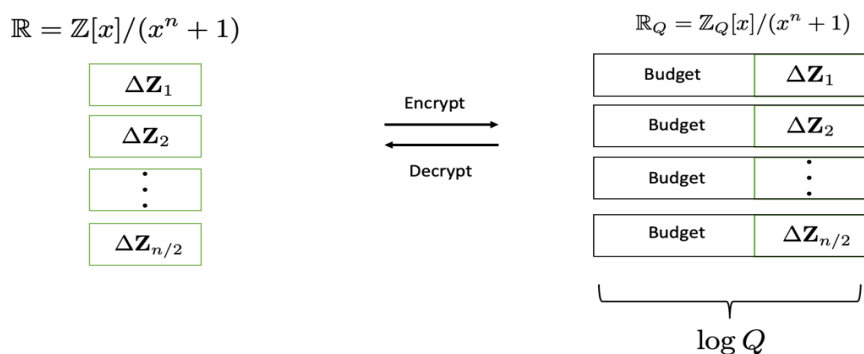
ดังนั้น พารามิเตอร์ที่ใช้ในการกำหนดสภาพแวดล้อม HE สำหรับ CKKS/RNS-CKKS คือ

$$P = (n, Q = \prod_{i=1}^L q_i, \Delta)$$

เมื่อ $p_i = (n, q_i, \Delta)$ สำหรับ $p_i \in P$ เมื่อกำหนดให้ลำดับชั้นของ Modulus Chain มีค่า L โดยที่ค่า Polynomial Modulus Chain เป็นดังนี้ $p_L \rightarrow p_{L-1} \rightarrow \dots \rightarrow p_1$ ซึ่งการกำหนด ค่าพารามิเตอร์ในแต่ละลำดับชั้นต้องสอดคล้องกับค่า Scale Factor และต้องเป็นไปตามมาตรฐานความปลอดภัย Security Level ที่กำหนด

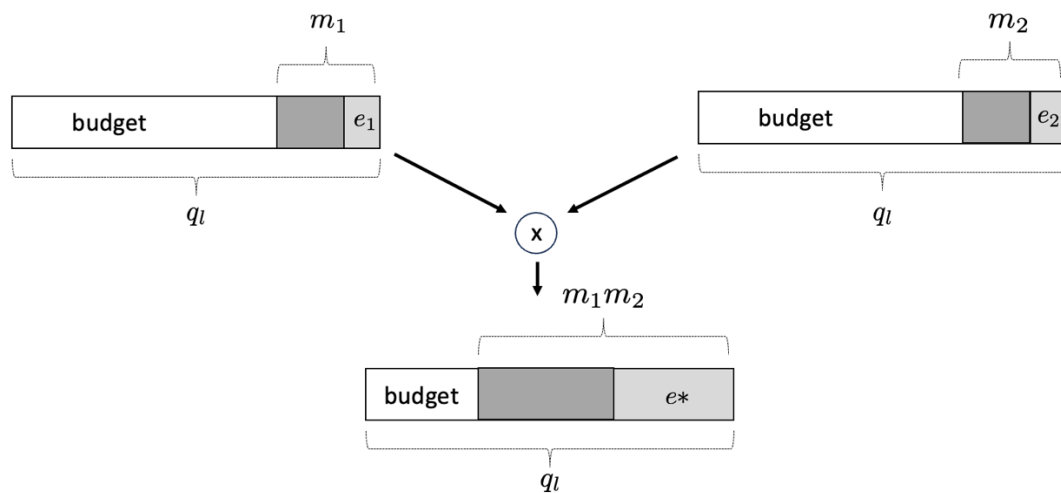


รูปที่ ๕๐ การ Encode/Decode ข้อมูล



รูปที่ ๕๑ การ Encrypt/Decrypt ข้อมูล

การกำหนดค่าพารามิเตอร์ที่เหมาะสมมีความสำคัญอย่างยิ่งต่อการนำข้อมูลที่เข้ารหัสไปคำนวณด้วยโอเพอร์เรชันทางคณิตศาสตร์ เนื่องจากเงื่อนไขของการคำนวณข้อมูลที่เข้ารหัสด้วยการบวก และการคูณทำให้ขนาดของข้อมูลที่เข้ารหัสเปลี่ยนไป และทำให้มี Noise เกิดขึ้นภายหลังการทำโอเพอร์เรชันทางคณิตศาสตร์ ดังรูปที่ ๕๒



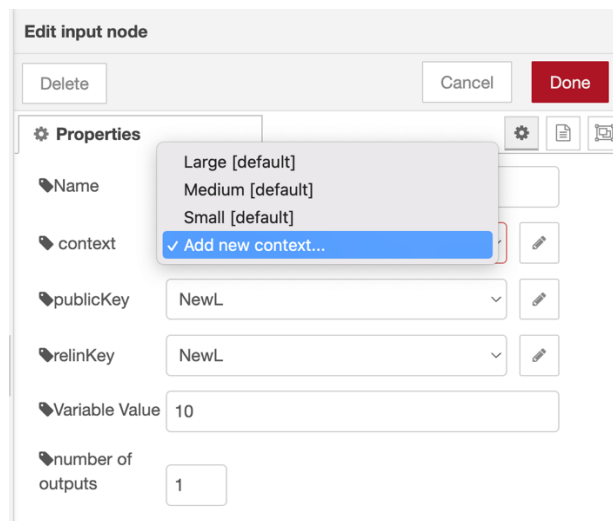
รูปที่ ๕๒ การคูณข้อมูลเข้ารหัส (e^* , e_1 , e_2 คือ Noise ที่เกิดขึ้นจากการเข้ารหัส)

ดังนั้นจะเห็นได้ว่าการกำหนดพารามิเตอร์สภาพแวดล้อม HE มีความซับซ้อนสำหรับผู้ใช้งานทั่วไปหากผู้ใช้งานกำหนดไม่ถูกต้องจะทำให้ไม่สามารถนำข้อมูลที่เข้ารหัสไปคำนวณได้และอาจจะทำให้การเข้ารหัสไม่ปลอดภัย ดังนั้นโครงการนี้มีวัตถุประสงค์เพื่อให้สามารถนำแพลตฟอร์มที่พัฒนาขึ้นไปใช้งานได้จริงในเชิงพาณิชย์ โดยกลุ่มเป้าหมายผู้ใช้งานคือนักพัฒนาไอโอที และเจ้าของโรงงานอุตสาหกรรม ซึ่งไม่มีความรู้ด้าน Cryptography นักวิจัยจึงพัฒนาให้ผู้ใช้งานสามารถเลือกพารามิเตอร์สำหรับการสร้างสภาพแวดล้อมโฮโมมอร์ฟิกที่เหมาะสมกับฟังก์ชันการคำนวณได้ง่ายโดยไม่ต้องอาศัยความเชี่ยวชาญด้าน Cryptography และพารามิเตอร์ที่มีให้เลือกใช้ต้องมีความยืดหยุ่นต่อการปรับแต่งฟังก์ชันการคำนวณได้อีกด้วย ดังนั้นโครงการนี้จึงได้พัฒนาให้การเลือกพารามิเตอร์ทำได้ง่ายมากขึ้น โดยกำหนดค่าพารามิเตอร์ Default ไว้เพียงสามทางเลือก นั่นคือ Small, Medium, Large ดังตารางที่ ๑๑ และตัวอย่างการเลือกค่าพารามิเตอร์สำหรับ Node Input, Encrypt, Decrypt และ setContext ดังรูปที่ ๕๓ ถึง ๕๖

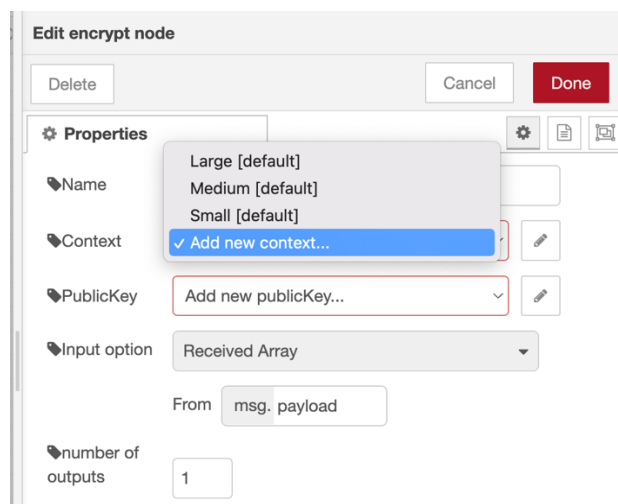
ตารางที่ ๑๑ พารามิเตอร์สภาพแวดล้อม FHE แบบ CKKS/RNS-CKKS ตามมาตรฐาน Security-Level 128

ขนาดของพารามิเตอร์	โพลีโมดูลัส	โคเอฟฟิเชียนโมดูลัส	สเกลเพคเตอร์	จำนวนครั้งของการคูณ
Small 140 kB	8192	[250,230,230,250]	230	3
Medium 146 kB	8192	[260,240,240,260]	240	3

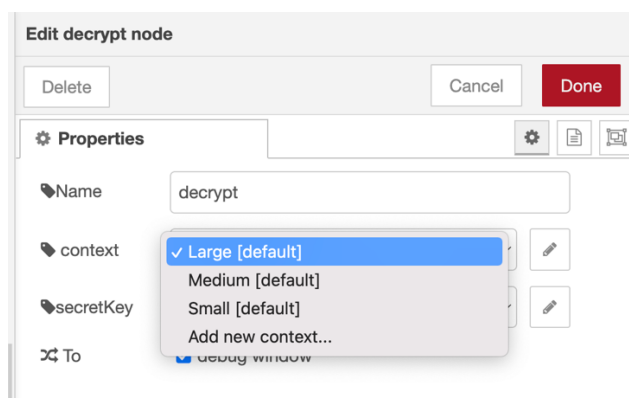
Large 174 kB	16384	[260,240,240,240,240,240,240,240,260]	240	7
-----------------	-------	---------------------------------------	-----	---



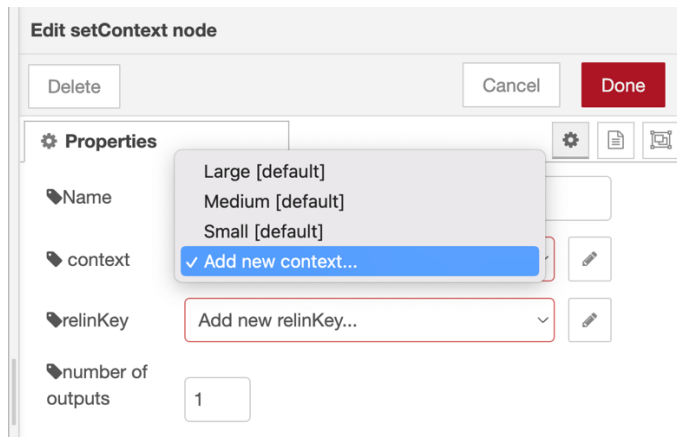
รูปที่ ๕๓ การเลือกพารามิเตอร์ RNS-CKKS บน Node Input



รูปที่ ๕๔ การเลือกพารามิเตอร์ RNS-CKKS บน Node Encrypt

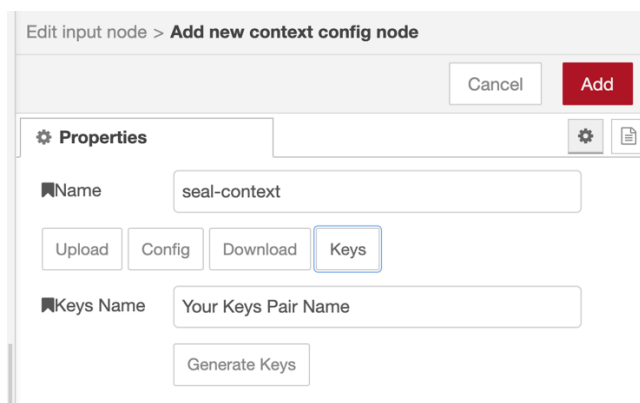


รูปที่ ๕๕ การเลือกพารามิเตอร์ RNS-CKKS บน Node Decrypt



รูปที่ ๕๖ การเลือกพารามิเตอร์ RNS-CKKS บน Node setContext

นอกจากนี้ ภายใน Node ยังมีฟีเจอร์สำคัญคือ การสร้างกุญแจ ดังแสดงในรูปที่ ๕๗ โดยผู้ใช้งานสามารถตั้งชื่อชุดกุญแจได้เอง และการสร้างชุดกุญแจจะสร้างตามพารามิเตอร์ Context ที่ได้เลือกไว้ก่อนหน้านี้ ผู้ใช้งานสามารถสร้างกุญแจได้หลายชุด และสามารถลบกุญแจที่ไม่ต้องการได้ โดยกุญแจจะถูกจัดเก็บไว้ในหน่วยความจำของ Node



รูปที่ ๕๗ การสร้างกุญแจ

Node สำหรับการทำโอเพอร์เรชันการคำนวณ ถูกพัฒนาขึ้นตามฟังก์ชันที่มีใน Microsoft SEAL Library แต่อย่างไรก็ตามโอเพอร์เรชันการคำนวณมีเพียงโอเพอร์เรชันการบวก การคูณ และการลบเท่านั้น ซึ่งไม่สามารถครอบคลุมการคำนวณทั่วไปของการใช้งานวิเคราะห์ข้อมูลโอไอทีได้ ดังนั้น ในโครงการนี้จึงได้พัฒนาอัลกอริทึมการหารสำหรับข้อมูลเข้ารหัสด้วยเทคนิคการประมาณค่าด้วยการคูณแบบวนซ้ำ โดยรายละเอียดของเทคนิคที่นำเสนออยู่ใน [77] โดยข้อจำกัดของการเลือกโอเพอร์เรชันการคำนวณการหาร (reciprocal) นี้คือ ผู้ใช้งานต้องเลือกพารามิเตอร์ RNS-CKKS ขนาด Large เท่านั้น

การส่งข้อมูลโอไอทีไปเข้ารหัสแบบโฮโมมอร์ฟิกส์แบบ RNS-CKKS ต้องเผชิญกับความท้าทายในการออกแบบนั่นคือ การออกแบบให้ Mobile Application สามารถรับข้อมูลเข้ารหัสที่ผ่านการคำนวณแล้วจากคลาวด์ส่งต่อให้โมดูลการถอดรหัสโฮโมมอร์ฟิกเพื่อนำไปแสดงผลบน Dashboard ของ Mobile Application เป็นความท้าทายอีกประการหนึ่ง เนื่องจากข้อมูลที่ถูกเข้ารหัสโฮโมมอร์ฟิกจะถูกบันทึกข้อมูลแบบ Batching นั่นคือ ข้อมูลหลายๆ ตัวจะถูกผนวกรวมในรูปแบบ Array ในการการเข้ารหัสแต่ละครั้ง รูปที่ ๕๘ และ ๕๙ แสดงการ

เปรียบเทียบการเข้ารหัสข้อมูลไฮโมเมอร์ฟิก (E_k) แบบเดี่ยวและแบบ Batching โดย $Ecd(M, \Delta)$ คือการแปลงข้อมูลเป็นรหัส (Encoding) ให้ Array M เป็นข้อความเดียวโดยใช้สเกล Δ

$$m_1 \rightarrow E_k(m_1) \rightarrow c_1$$

รูปที่ ๕๘ การเข้ารหัสข้อมูลไฮโมเมอร์ฟิกแบบเดี่ยว

$$M = [m_1, m_2, \dots, m_N] \rightarrow Ecd(M; \Delta) \rightarrow E_k(Ecd(M; \Delta)) \rightarrow c$$

รูปที่ ๕๙ การเข้ารหัสข้อมูลไฮโมเมอร์ฟิกแบบ Batching

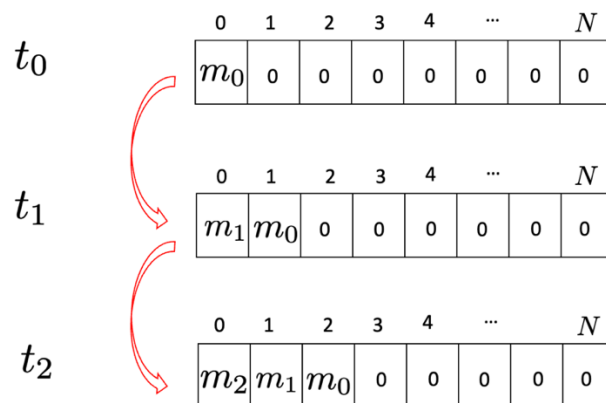
ดังนั้นจะเห็นได้ว่าในแง่การเข้ารหัสและสื่อสารข้อมูล การบรรจุข้อมูลให้เต็ม Array เพื่อนำไปเข้ารหัสและส่งรวมกันจะได้ประสิทธิภาพสูงสุด อย่างไรก็ตามการนำวิธีการดังกล่าวมาใช้งานในระบบไอโอทีโดยไม่ประยุกต์ดัดแปลงใดๆ เลยไม่สามารถทำได้ด้วยเหตุผลดังนี้

ข้อมูลของโรงงานอุตสาหกรรมที่ส่งในระบบไอโอทีมักเป็นค่าที่เก็บจากเซนเซอร์เป็นอนุกรมเวลาเช่น ๑ ค่าต่อ t วินาที ตัวอย่างเช่น ข้อมูลการวัดค่าการใช้พลังงาน ข้อมูลผลผลิต หรือข้อมูลพารามิเตอร์ต่างๆ ของเครื่องจักร เป็นต้น จากการสำรวจการใช้งานไอโอทีสำหรับโรงงานอุตสาหกรรม (อ้างอิงโครงการ IDA และ 5G Smart Factory) อัตราการส่งข้อมูลอยู่ที่ ๑ ข้อมูลต่อ ๑๕ วินาทีหรือ ๔ ครั้งต่อนาที นั้นหมายความว่าถ้าต้องการรอให้ข้อมูลถูกบรรจุลงใน Array ความยาว $N/2$ จนครบ จะต้องรอเป็นระยะเวลา $๑๕ \times N/๒$ วินาที (หมายเหตุ N คือโพลีโมดูลัสของการเข้ารหัสไฮโมเมอร์ฟิกและเป็นตัวกำหนดความยาวของ Array ของ Batch) ตารางที่ ๑๒ แสดงการเปรียบเทียบระยะเวลาการรอส่งข้อมูลแบบข้อมูลเดี่ยวและแบบ Batching ที่ขนาด Array ต่างๆ ตามที่แพลตฟอร์ม Cyblion ได้กำหนดไว้คือ $N = 2^{13} = 8192$ สำหรับพารามิเตอร์การสร้างสภาพแวดล้อมการเข้ารหัสขนาดเล็กและกลาง และ $N = 2^{14} = 16384$ สำหรับพารามิเตอร์ขนาดใหญ่ จะเห็นได้ว่าหากรอบรรจุ Array จนครบก่อนเข้ารหัสและส่งไปยังคลาวด์ ผู้ใช้ปลายทางจะเห็นข้อมูลใหม่ ๑-๒ วันในแต่ละครั้ง ซึ่งเป็นไปไม่ได้เลยในการใช้งานในระบบไอโอทีที่มีจุดเด่นหนึ่งคือการสื่อสารให้เห็นข้อมูลได้แบบ Real Time

ตารางที่ ๑๒ เปรียบเทียบระยะเวลาการรอส่งข้อมูลแบบข้อมูลเดี่ยวและแบบ Batching

ขนาดของพารามิเตอร์	โพลีโมดูลัส	ระยะเวลาการรอส่งข้อมูลแบบข้อมูลเดี่ยว	ระยะเวลาการรอส่งข้อมูลข้อมูลแบบ Batching
Small 140 kB	8192	15 วินาที	7.11 ชั่วโมง
Medium 146 kB	8192	15 วินาที	7.11 ชั่วโมง
Large 174 kB	16384	15 วินาที	34.13 ชั่วโมง

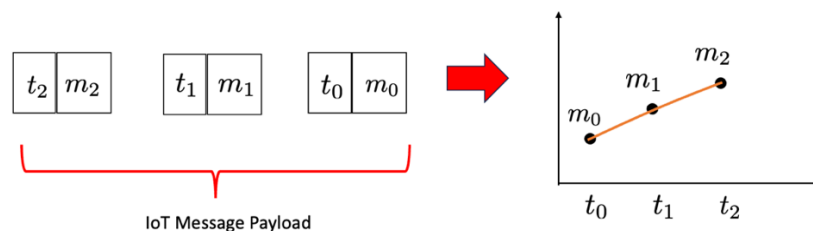
ดังนั้นจึงต้องมีการออกแบบการส่งข้อมูลใหม่โดยไม่ต้องรอข้อมูลถึง N ตัว และเพิ่ม Reliability ของการส่งข้อมูลจากการส่งซ้ำ (Redundancy) ข้อมูลบางส่วนในแต่ละ Array ดังรูปที่ ๖๐



รูปที่ ๖๐ การจัดเรียงข้อมูลไอโอทีใน Array สำหรับการส่งข้อมูลแบบ Batch

ในภาครับสำหรับการนำข้อมูลที่ถอดรหัสมาแสดงผลสำหรับการแสดงผลแบบ Text และ Gauge ซึ่งเป็นค่าล่าสุด โปรแกรมจะเลือกข้อมูล index ที่ศูนย์มาแสดงผล

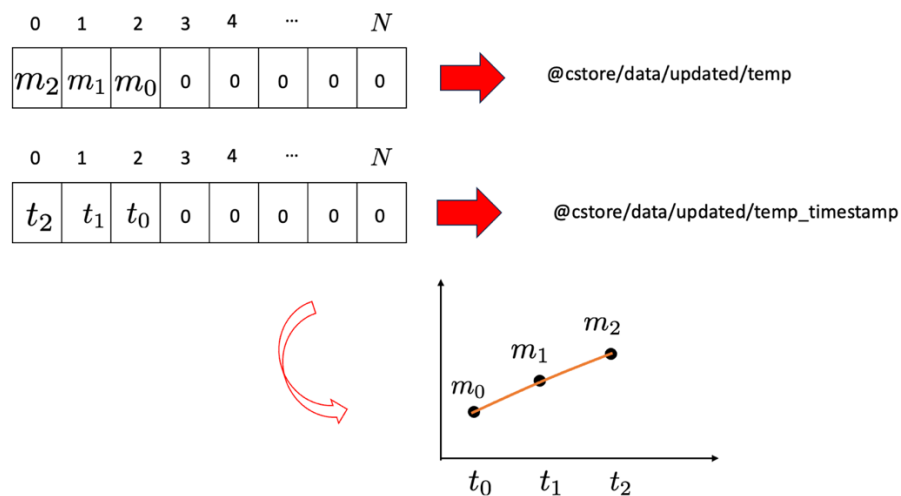
ส่วนการแสดงผลข้อมูลแบบ Feed ในรูปแบบกราฟแสดงค่าข้อมูลตามแกนเวลา มีความท้าทายคือการส่งข้อมูลไอโอทีโดยปกติที่ไม่มีการเข้ารหัส จะมีการส่งค่า Timestamp พร้อมมากับข้อมูลแบบ ๑ ต่อ ๑ ดังนั้นฝั่งรับจึงได้รับข้อมูลที่ครบถ้วน (เวลาและค่าของข้อมูล) ในการรับข้อมูล ๑ ครั้งต่อการเพิ่ม ๑ จุดลงในกราฟ ดังแสดงในรูปที่ ๖๑



รูปที่ ๖๑ การจัดเรียงข้อมูลแบบ Feed ลงบนกราฟแสดงผลตามแกนเวลาสำหรับการส่งข้อมูลแบบเดี่ยว สำหรับการส่งข้อมูลที่ไม่เข้ารหัส

สำหรับแพลตฟอร์ม Cyblion ซึ่งข้อมูลไอโอทีที่ถูกบรรจุแบบ Batching ทำให้ไม่สามารถระบุค่า Timestamp ของข้อมูลแต่ละตัวใน Array ได้ เนื่องจากหากทำการบรรจุข้อมูล Timestamp ควบคู่ลงไปกับข้อมูลไอโอที ข้อมูล Timestamp จะถูกนำไปแปลงเป็นรหัสรวมเข้าไปกับข้อมูลไอโอทีด้วย เมื่อนำไปเข้ารหัสโหมมอร์ฟิกก็จะไม่สามารถนำไปคำนวณตามฟังก์ชันทางคณิตศาสตร์ต่อได้ ดังนั้นทีมวิจัยจึงกำหนดให้อุปกรณ์ต้นทางส่งข้อมูลไอโอทีที่เข้ารหัส และข้อมูล Timestamp ที่ถูกเข้ารหัสแยกจากกันโดยแยกชนิด Array และแยก Topic

เพื่อให้ Mobile Application ที่เครื่องปลายทางสามารถเลือกข้อมูลมาแสดงตามแกน x หรือ y ได้อย่างถูกต้อง ดังแสดงในรูปที่ ๖๒



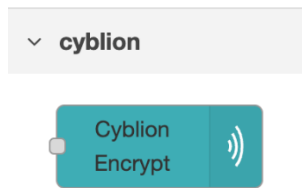
รูปที่ ๖๒ การจัดเรียงข้อมูลแบบ Feed ลงบนกราฟแสดงผลตามแกนเวลา
สำหรับการส่งข้อมูลแบบ Batching ของข้อมูลเข้ารหัส

๔.๑.๒ แพลตฟอร์ม CYBLION

CYBLION เป็นแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับไอโอทีที่พัฒนาต่อยอดมาจาก CipherFlow โดยมีการแยกส่วนประกอบของ Node-RED ออกจากกัน แบ่งเป็น 3 ส่วน ได้แก่ Edge Computing, Cloud Computing และ Mobile Application

๑) Edge Computing

สำหรับการทำ Encrypt และ การส่งข้อมูลด้วย MQTT ไปยัง MQTT Broker ที่พัฒนาขึ้นโดยเฉพาะ สำหรับการติดตั้งใน Node-RED บน Edge Computing โดยข้อแตกต่างจาก Node Encrypt ใน CipherFlow และ Node CYBLION Encrypt คือ ภายใน Node และ Node CYBLION Encrypt จะไม่มีการสร้างกุญแจ แต่จะรับกุญแจมาจาก Mobile Application ผ่านทาง QR Code ดังรูปที่ ๖๓ ถึง ๖๕ แสดง Node CYBLION Encrypt และรายละเอียดการติดตั้ง ส่วนการส่งข้อมูล MQTT จะสามารถกำหนด Topic ของข้อมูลที่ต้องการส่งพร้อมกันนั้นข้อมูล Time stamp จะถูกกำหนดให้ส่งไปใน Topic หนึ่งแยกจากกัน ทั้งนี้จะแตกต่างจากการส่งข้อมูลแบบ MQTT ทั่วไปที่ Time stamp จะถูกส่งรวมไปกับข้อมูล Topic เดียวกันได้ เหตุผลในการออกแบบให้การส่งข้อมูล Time Stamp แยกจากข้อมูลที่ต้องการคำนวณเพราะข้อมูลจะถูกเข้ารหัสและจะถูกคำนวณด้วยโอเพอร์เรชั่นต่างๆ ที่กำหนดไว้ในคลาวด์ทำให้คลาวด์ไม่สามารถแยก Time Stamp ออกจากการคำนวณบนข้อมูลที่เข้ารหัสได้ ทำให้เมื่อถึง Mobile Application ปลายทาง ข้อมูล Time Stamp จะถูกเปลี่ยนแปลงไปตามสูตรที่คำนวณ ทำให้นำข้อมูลนั้นไปใช้ในการแสดงผลกราฟตามแกนเวลาไม่ได้ ดังนั้น ในภาคส่งข้อมูลจำเป็นต้องเข้ารหัสข้อมูล และเข้ารหัส Time Stamp แยกจากกัน และส่งด้วย Topic ที่แตกต่างกัน



รูปที่ ๖๓ Node CYBLION Encrypt

 This is a screenshot of the 'Edit Cyblion Encrypt node' configuration window. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. The 'Properties' section includes a 'Name' text field, a 'Device' dropdown menu currently set to 'No Device', and radio buttons for 'sync' and 'connect' (the 'connect' option is selected). Below these is an 'Add Device' button. The 'MQTT TOPIC' section has two checkboxes: 'Send the encrypted value to' and 'Send the encrypted timestamp to'. A yellow box provides an example of an encrypted value topic: '@cstore/data/updated/home/bedroom/temperature' and '@cstore/data/updated/oil/height'.

รูปที่ ๖๔ การตั้งค่า Node CYBLION Encrypt

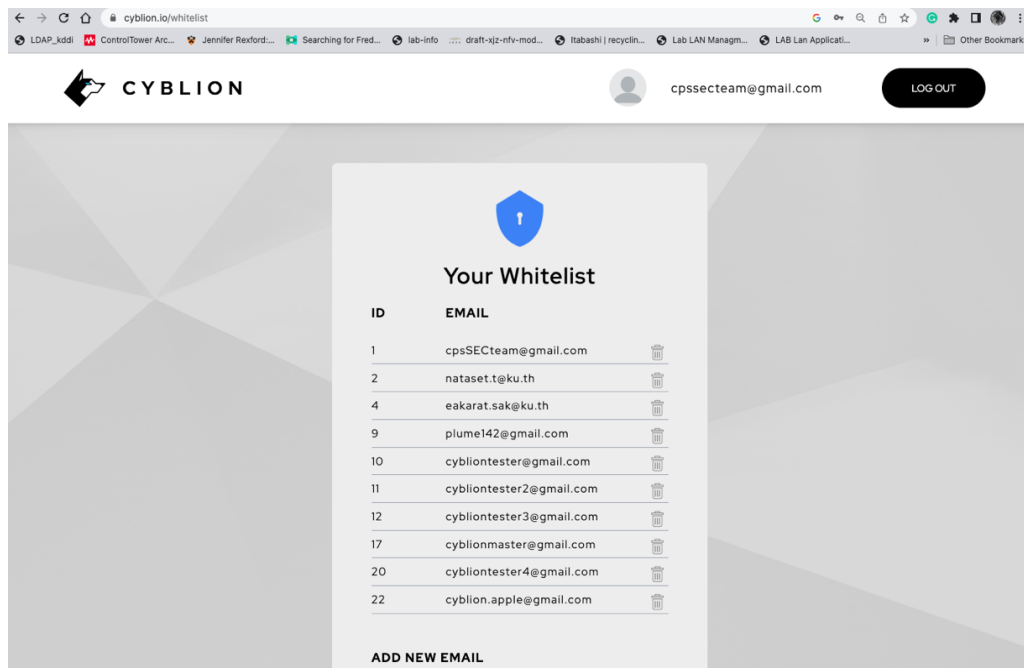
 This is a screenshot of the 'Add new cyblion-mqtt-broker config node' configuration window. It has 'Cancel' and 'Add' buttons at the top. The 'Properties' section shows 'Sync' and 'Config' tabs, with 'Sync' currently active. The 'Cyblion Sync Device' section instructs to 'Synchronize Device Config with QR-Code' and displays a large QR code. Below the QR code is an 'Allow Sync' button. At the bottom, there is a status bar showing 'Enabled', '0 nodes use this config', and a dropdown menu set to 'On all flows'.

รูปที่ ๖๕ การแลกกุญแจผ่าน QR Code

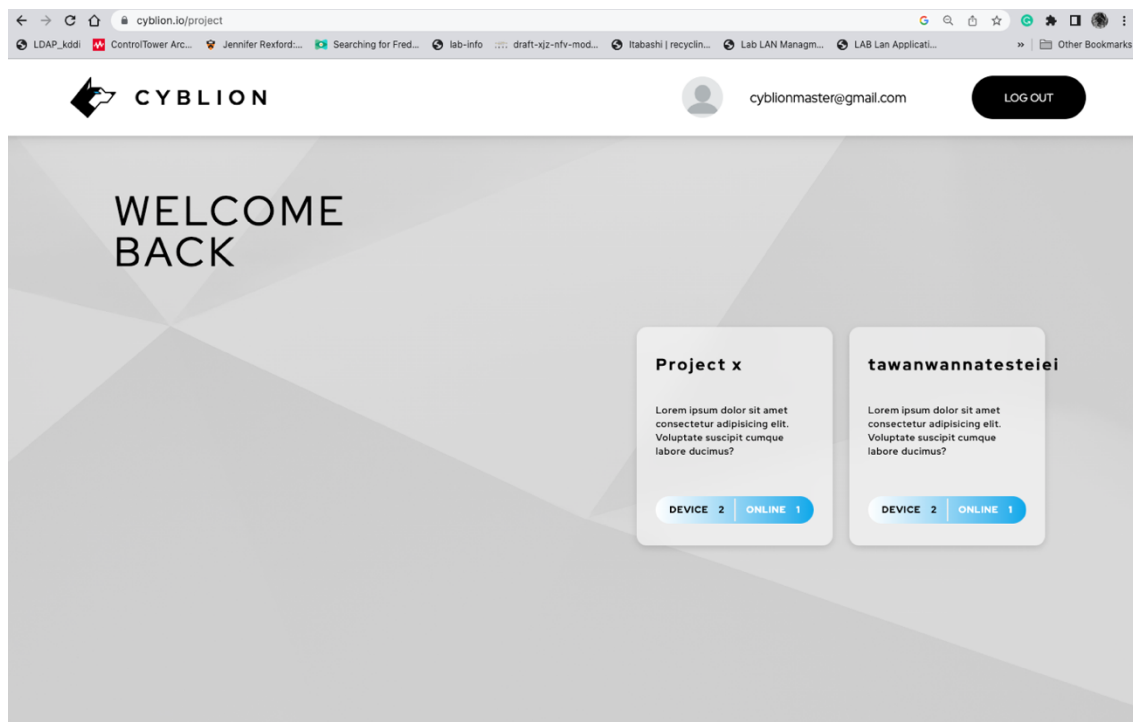
๒) Cloud Computing

การพัฒนาในส่วนของ Cloud เป็นในรูปแบบของ Web Application โดยใช้โดเมนเนม Cyblion.io และพัฒนา UI สำหรับผู้ใช้งาน ดังรูปที่ ๖๖ โดยมีส่วนหน้าต่างของ Admin สำหรับการเพิ่มชื่อผู้ลงทะเบียนใช้งานบน Web Application ใน Whitelist สำหรับ Admin User ส่วนการใช้งาน Login ใช้งาน Web Application เพื่อตรวจสอบสิทธิ์ในการเข้าถึง Project ที่ถูกสร้างไว้ สำหรับ User ทั่วไป เป็นดังรูปที่ ๖๗ จากนั้นผู้ใช้งาน

สามารถเข้าไปยัง Project ที่ถูกสร้างขึ้นจาก Mobile Application โดยผู้ใช้งานสามารถเข้าไปสร้างสูตรการคำนวณในแต่ละ Project ได้แยกอิสระจากกัน



รูปที่ ๖๖ จำนวนผู้ใช้งานบนคลาวด์ ๑๐ ราย

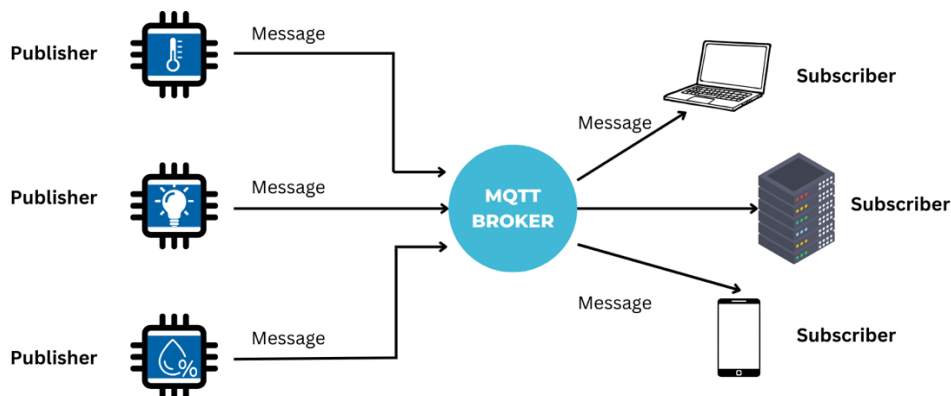


รูปที่ ๖๗ การเข้าถึงพื้นที่สำหรับสร้างสูตรคำนวณด้วย Project พร้อมแสดงจำนวน Edge Device ในโปรเจ็ค

ตารางที่ ๑๓ เปรียบเทียบ MQTT Broker ที่ให้บริการในปัจจุบันกับ CYBLION

MQTT Broker	Payload Size	Address and Port	SSL Support	WebSocket	Scalability	FHE
AWS IoT Core	128 kB	Managed	Yes	Dynamically assigned	Auto scale	No
Azure IoT Hub	256 kB	Managed	Yes	Dynamically assigned	Auto scale	No
MindSphere	1 MB	8883, 443	Yes	mindconnectmqtt.eu1. mindsphere.io	Broker- Clustering	No
NETPIE	64 kB	1883, 8883, 443	Yes	Mqtt.netpie.io	Broker- Clustering	No
CYBLION	4 MB	18883, 8084, 8883, 443	Yes	fhe.netpie.io	Auto scale	Yes

ดังนั้น ทีมวิจัยจึงได้ออกแบบพัฒนา MQTT Broker ขึ้นมาใหม่ให้สามารถรองรับข้อมูลที่เข้ารหัสแบบ FHE แบบ RNS-CKKS ได้ รวมทั้งสามารถขยายตัวได้แบบอัตโนมัติตามมาตรฐานคลาวด์แพลตฟอร์มสาธารณะทั่วไป โดยการพัฒนาในรูปแบบของ Microservices ภายใต้ชื่อ EdgeBroker



รูปที่ ๗๐ การสื่อสารข้อมูลไอโอทีด้วย MQTT Protocol

การออกแบบ EdgeBroker ยังคงอยู่บนพื้นฐานของ MQTT โดยการเป็นตัวกลางการสื่อสารสองทางแบบ Real-Time Messaging ในขณะที่ต้องมีความสามารถในการกระจายข้อความที่มี Payload ขนาดใหญ่ที่เกิดจากการเข้ารหัสแบบ Block Cipher ของ FHE algorithm แบบ CKKS/RNS-CKKS โดยตามมาตรฐาน OASIS MQTT Version 3.1.1 [8] มีการจำกัดค่า Payload ของข้อความ MQTT สูงสุดที่ ๒๖๐ MB ซึ่งมีขนาดใหญ่กว่าขนาดของข้อความที่แพลตฟอร์ม Cyblion ต้องการส่ง ดังนั้นการเพิ่มขนาด Payload ขึ้นเป็น ๔ MB จึงยังอยู่ภายใต้มาตรฐานและไม่กระทบกับ QoS ของ MQTT Broker

ทีมวิจัยได้ออกแบบ EdgeBroker ให้รับการเชื่อมต่อและรับข้อมูลจากอุปกรณ์ Edge Computing ด้วยโพรโทคอล MQTT และ MQTT over WebSocket ซึ่งทำให้องค์กรบริการเชื่อมต่อแบบ Plain MQTT/WS และแบบ over TLS ด้วย MQTTS/WSS

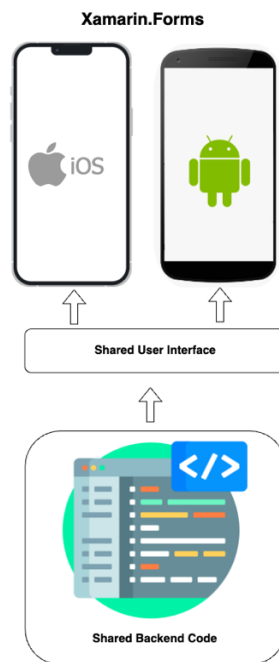
ความท้าทายอีกประการของการออกแบบ EdgeBroker คือการทำให้ EdgeBroker สามารถทำงานได้แบบกระจายตัวและขยายตัวได้อัตโนมัติ นั่นคือ สามารถเปิด Broker (ที่อาจกระจายตัวอยู่ในที่ต่างๆ) เพิ่มขึ้นตามจำนวนอุปกรณ์ Edge Computing และจำนวนผู้ใช้งานที่เพิ่มขึ้น ในขณะที่ Broker ทุกตัวยังคงทำงานแบบหลอมรวมกันเสมือนมี Broker เพียงตัวเดียว ทีมวิจัยจึงเลือกใช้สถาปัตยกรรมแบบ Microservice ที่พัฒนาบน Kubernetes (K8s) โดย Microservice EdgeBroker จะถูก Build เป็น Container Image เพื่อที่จะสามารถรันในระบบ Container Orchestration ได้ เนื่องจาก Kubernetes มีระบบอัตโนมัติที่ช่วยดูแลเงื่อนไขการทำงานของ Broker ให้เป็นไปตามที่กำหนดไว้ได้ เช่น Worker Pod ซึ่งหากมีความผิดปกติเกิดขึ้นกับ Node ที่รัน Pod อยู่ระบบจะทำการกู้คืน Pod ไว้ที่ Node อื่นที่ยังทำงานปกติ

Image ของ EdgeBroker ประกอบด้วย 3 ส่วนหลัก คือ ๑) EdgeBroker Topic Rewriter ๒) VerneMQ [12] ซึ่งเป็น Open-Source MQTT Broker ที่เขียนด้วยภาษา Erlang และ ๓) Redis Cache ที่ใช้ในการแคช Credential ต่างๆ เพื่อลดภาระโหลดของ Database และยังใช้เก็บข้อมูล cstore อีกด้วย Image ของ EdgeBroker ทั้งหมดอยู่บน <https://api.daysie.io/> ซึ่งเป็น Harbor Container Registry ของ CPS ซึ่งสร้างเป็น Robot Account ให้สิทธิ์ Kubernetes ในคลาวด์ของ Cyblion ในการ Pull Image ไปติดตั้ง สำหรับทางเข้าของบริการของ EdgeBroker จะเข้าผ่าน NGINX Proxy เข้าไปยัง Node Port ของ Kubernetes Service

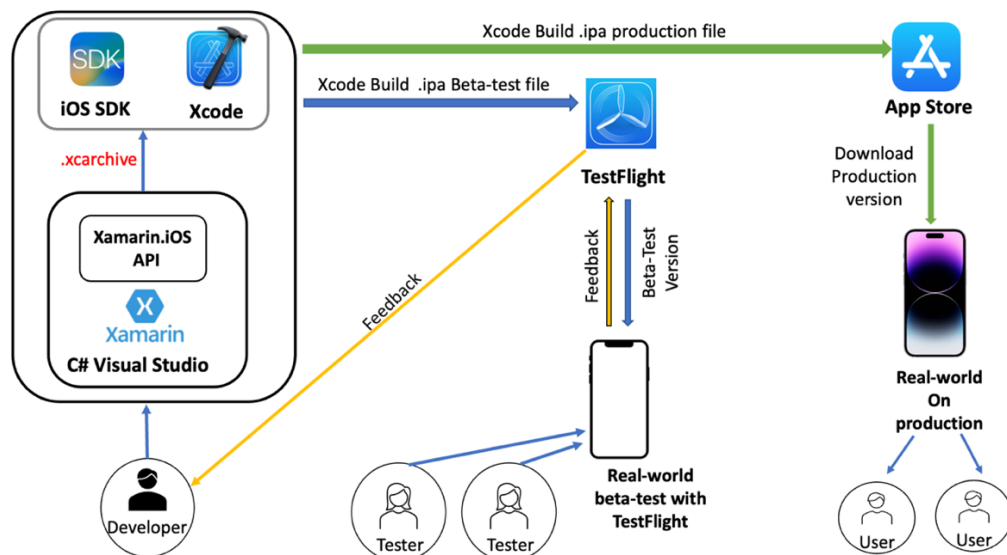
๓) Mobile Application (CYBLION)

ทีมวิจัยได้พัฒนา Mobile Application บนแพลตฟอร์มของ Xamarin.Forms เนื่องจากเป็นโปรแกรมเดียวที่ใช้งานได้กับ Source Code Library ของโปรแกรม Microsoft SEAL ซึ่งเป็นหัวใจของการทำงานของระบบการใช้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล เพื่อการเข้ารหัสข้อมูลแบบ HE และเลือกพัฒนาบน Xamarin.Forms ที่อนุญาตให้แชร์ส่วน Backend Code และ User Interface ทำให้การพัฒนา Backend และการออกแบบ UI ของระบบปฏิบัติการได้ทั้ง iOS และ Android เพื่อลดการทำงานที่ซ้ำซ้อน ดังรูปที่ ๗๑ ดังนั้นเมื่อทีมพัฒนาสามารถพัฒนาส่วนของ Backend และการออกแบบ UI แล้วเสร็จจึงทำการแชร์ใช้ Source Code ให้กับนักพัฒนา iOS และ Android เพื่อไม่ให้เกิดความซ้ำซ้อนในขั้นตอนการพัฒนาและปรับปรุง Backend และ UI โครงการนี้ได้เริ่มจากการทดสอบ Build Mobile Application บน iOS โดยขั้นตอนการ Build Mobile Application สำหรับใช้ทดสอบและกระบวนการทดสอบเป็นดังรูปที่ ๗๒ โดยเป้าหมายเพื่อต้องการลดความซ้ำซ้อนของงานและการขัดแย้งของการปรับ Backend Code และเมื่อการปรับปรุงส่วนของ Backend Code และ UI เสร็จสิ้นแล้ว ทางทีมพัฒนาจึงเริ่มทดสอบ Build Mobile Application บน Android ดังรูปที่ ๗๓ โดยทางโครงการได้กำหนดให้ผู้ทดสอบระบบไม่ใช่ผู้พัฒนา Mobile Application เพื่อให้แน่ใจว่าการทดสอบจะครอบคลุมทุกฟังก์ชันการใช้งาน และเป็นการตรวจสอบความง่ายของการใช้งานเสมือนผู้ใช้งานจริงไม่ต้องมีความรู้

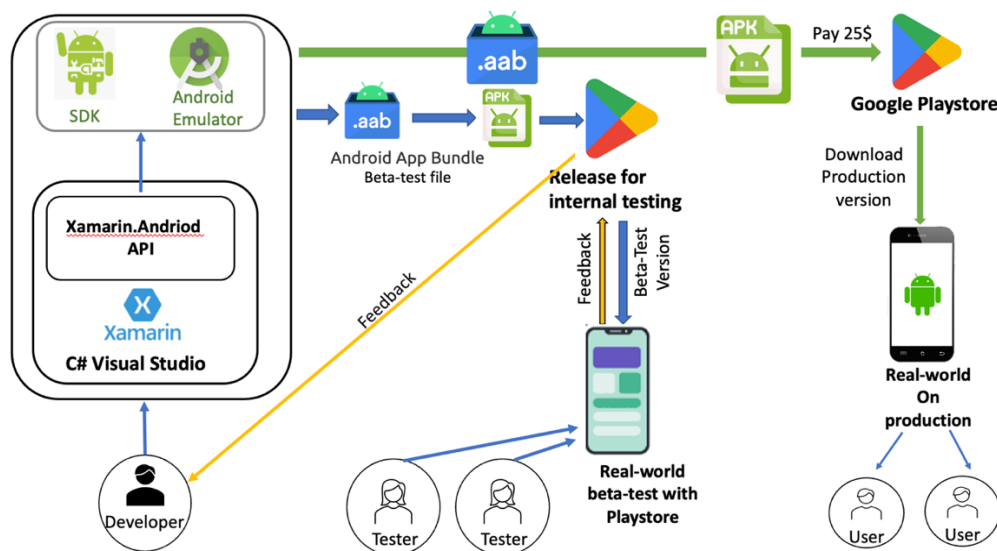
หรืออ่านคู่มือการใช้งานมาก่อน ผู้ทดสอบจะทำการ Report ข้อผิดพลาดและ Bug ต่างๆ ไม่ว่าจะเป็นข้อความแจ้งเตือนที่ผิดพลาด และ Bug ที่เกิดจากโปรแกรม อีกทั้งการปรับปรุงเพื่อให้เกิด User Experience ที่ดีที่สุด



รูปที่ ๗๑ การพัฒนา Mobile Application ด้วย Xamarin.Forms

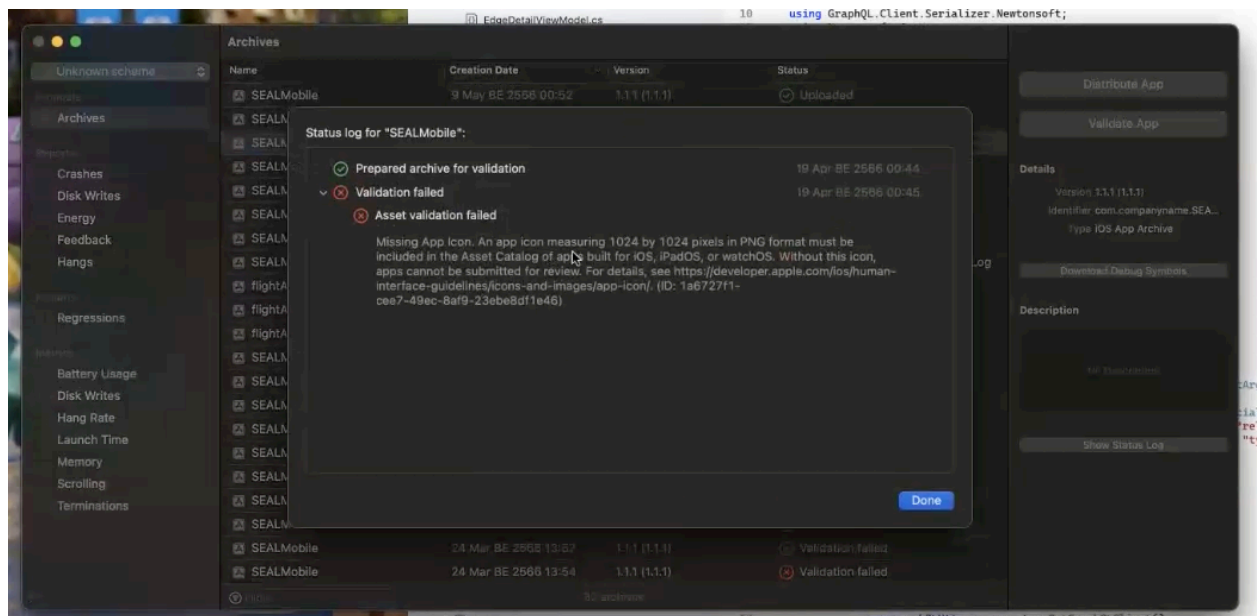


รูปที่ ๗๒ Diagram การพัฒนา iOS Application ด้วย Xamarin



รูปที่ ๗๓ Diagram การพัฒนา Android Application ด้วย Xamarin

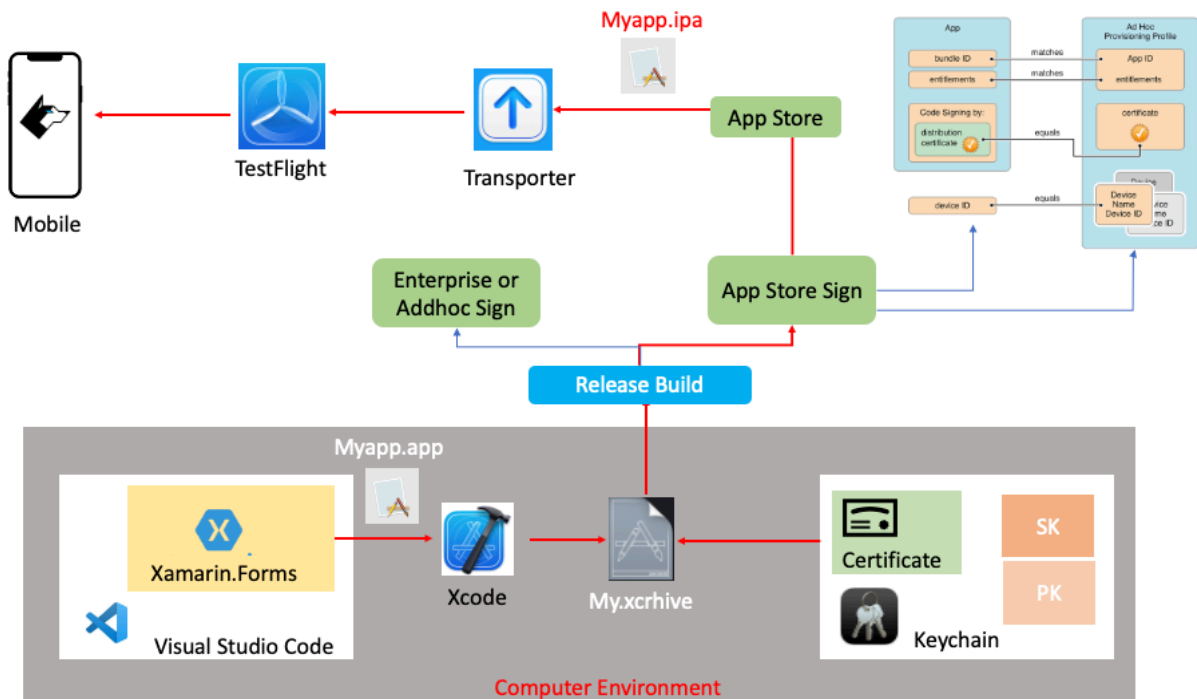
แต่อย่างไรก็ตาม ในระหว่างการพัฒนา Mobile Application ทางทีมพัฒนาได้พบปัญหาในการพัฒนา iOS Application บน Xamarin.Forms นั่นคือ Validation Failed เนื่องจาก Asset Validation Failed ในขั้นตอนการนำ Mobile Application ที่พัฒนาแล้วเข้าสู่โปรแกรม TestFlight ของ iOS เพื่อกระจายให้ Tester ได้ทำการทดสอบ Mobile Application เพื่อตรวจสอบความถูกต้องก่อนนำขึ้น Apple Store นั้น จะต้องผ่านขั้นตอนการทำ Archives โดยปัญหาที่ทีมพัฒนาพบ คือ ไม่สามารถผ่านขั้นตอนนี้ได้ โดยผลของการรัน Archives ได้แสดง Error Message ดังรูปที่ ๗๔



รูปที่ ๗๔ ข้อความแจ้งเตือน Prepared archive for validation failed

แนวทางแก้ไขปัญหาที่พบเนื่องจาก archive for validation failed และ ไม่สามารถแก้ไขตาม Error Message ได้ เนื่องจาก Error Message ไม่สามารถแจ้งรายละเอียด ดังนั้นจึงเสนอวิธีการแก้ไขโดยติดตั้ง

โปรแกรม Transporter (<https://developer.apple.com/news/?id=10152019a>) เพื่อใช้ในการอัปเดต Mobile Application เพื่อทำการกระจาย Mobile Application ให้ Tester สำหรับ Internal Testing ก่อนการส่ง Mobile Application ไปยังโปรแกรม TestFlight เนื่องจาก Error Message จาก Transporter จะให้ข้อมูลที่ละเอียดกว่าการส่งโปรแกรมขึ้น TestFlight โดยตรง ดังนั้นได้อะแกรมแสดงขั้นตอนการอัปเดต Mobile Application ไปยัง TestFlight จึงได้ถูกปรับใหม่ดังรูปที่ ๗๕

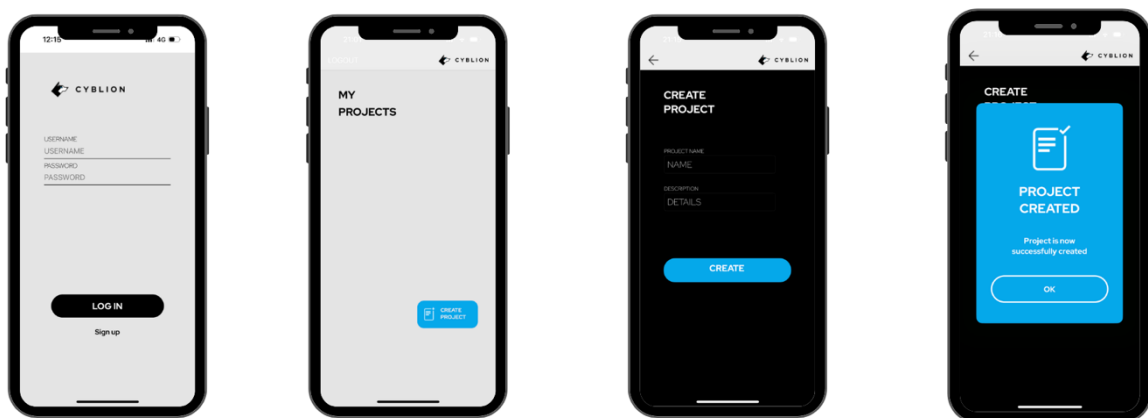


รูปที่ ๗๕ ไดอะแกรมกระบวนการ Build Application และอัปเดตขึ้น TestFlight

การออกแบบ UI ของ Mobile Application โดยแสดงในส่วนฟิจเจอร์ต่างๆ เป็นดังนี้

- การสร้าง Project

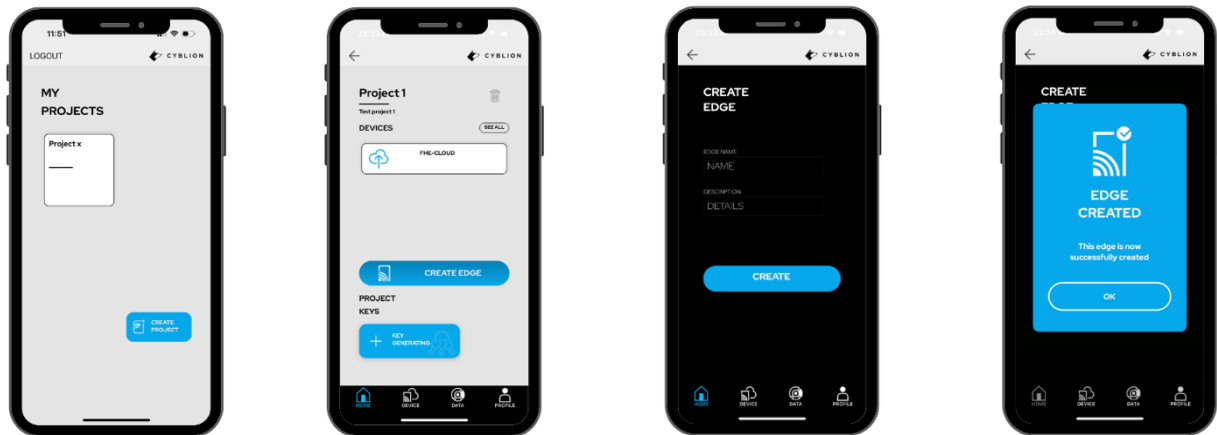
โครงการนี้ออกแบบ UI สำหรับขั้นตอนการสร้าง Project บน Mobile Application ดังนี้ โดยขั้นตอนการสร้างเป็นไปตามลำดับดังรูปที่ ๗๖



รูปที่ ๗๖ UI สำหรับการสร้าง Project

- การสร้าง Edge Computing

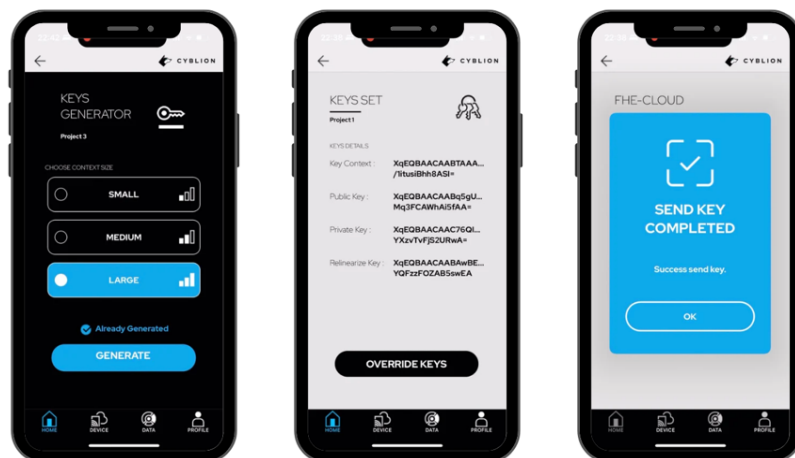
โครงการนี้ออกแบบ UI สำหรับขั้นตอนการสร้าง Edge Computing บน Mobile Application ดังนี้ โดยขั้นตอนการสร้างเป็นไปตามลำดับดังรูปที่ ๗๗



รูปที่ ๗๗ UI สำหรับการสร้าง Edge Computing

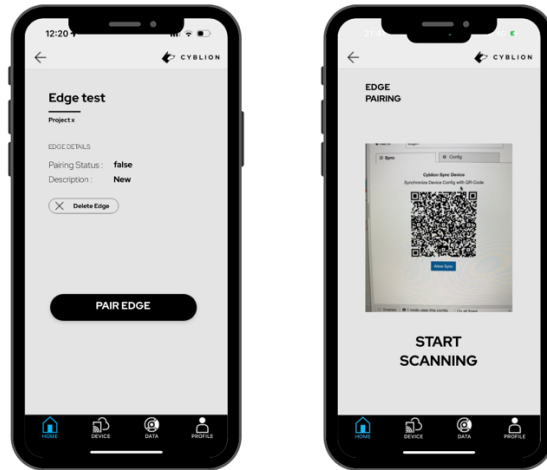
- การสร้าง Keys

โครงการนี้ออกแบบ UI สำหรับขั้นตอนการสร้าง Keys โดยผลการสร้างจะได้กุญแจสามชนิด sk, pk, และ rk สามารถตรวจสอบความถูกต้องของกุญแจได้จากค่า Hash ที่แสดงในหน้าต่าง Keyset และเมื่อต้องการเปลี่ยนชุดกุญแจ สามารถทำการสร้างกุญแจใหม่ได้โดยกดปุ่ม Override Keys ดังรูปที่ ๗๘



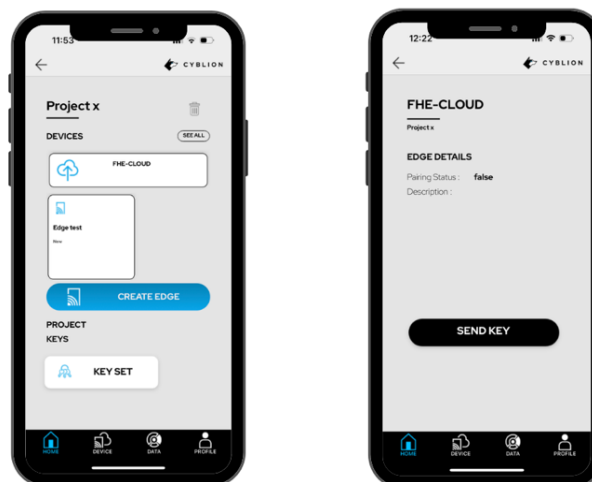
รูปที่ ๗๘ UI สำหรับการสร้าง Keys

- การส่ง Key ไปยังอุปกรณ์ Edge computing ภายนอก
 โครงการนี้ออกแบบ UI สำหรับขั้นตอนการส่ง pk ไปยังอุปกรณ์ Edge computing ภายนอก ด้วยการ Scan QR code บน Node-RED Cyblion encrypt บนโปรแกรม Node-RED ที่ติดตั้งบนอุปกรณ์ Edge computing ดังรูปที่ ๘๙



รูปที่ ๘๙ UI สำหรับการส่ง Key ไปยังอุปกรณ์ Edge computing

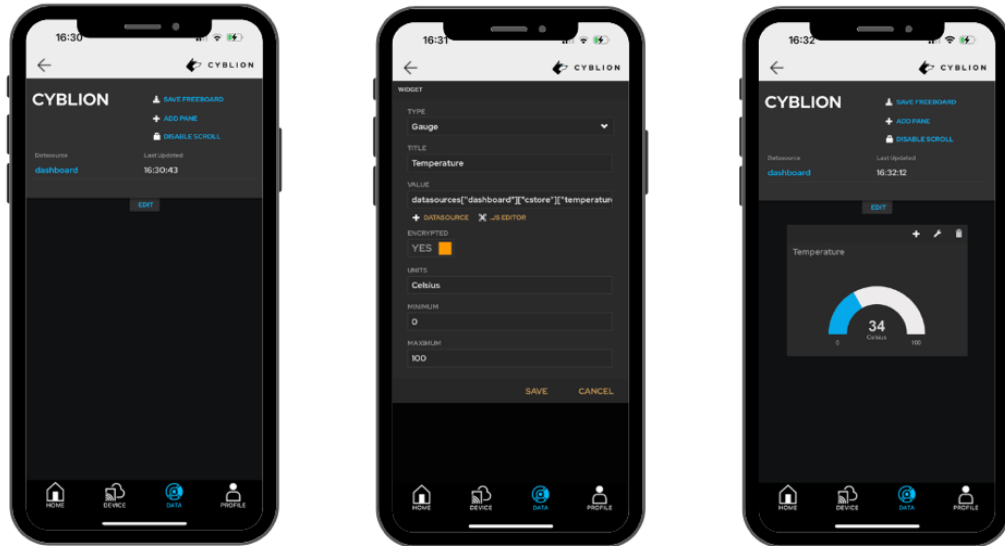
- การส่ง Key ไปยัง Cloud
 โครงการนี้ออกแบบ UI สำหรับขั้นตอนการส่งกุญแจ rk ไปยัง CYBLION.io โดยให้ผู้ใช้กดปุ่ม FHE-CLOUD เพื่อทำการส่งกุญแจ rk



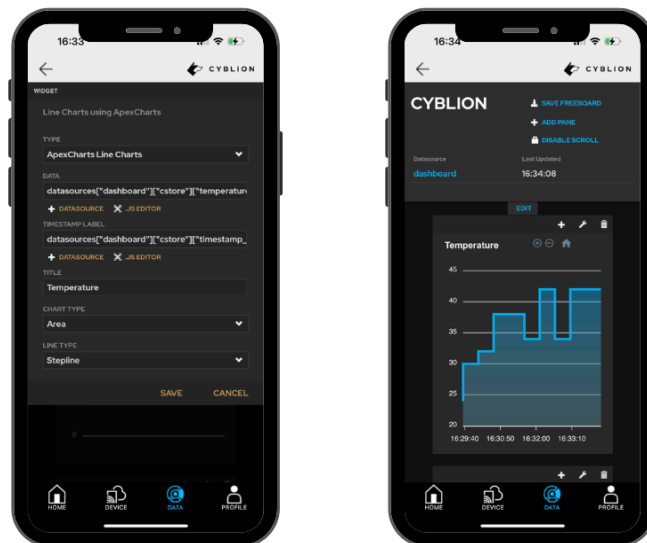
รูปที่ ๘๐ UI สำหรับการส่ง Key ไปยัง Cloud

- การติดตั้งส่วนแสดงผลข้อมูล

โครงการนี้ออกแบบ UI สำหรับส่วนแสดงผลข้อมูล โดยให้ผู้ใช้เป็นผู้ตั้งค่าและเลือกข้อมูลที่ต้องการแสดงด้วยตัวเอง



รูปที่ ๘๑ UI สำหรับการแสดงผลข้อมูลแบบ Gauge



รูปที่ ๘๒ UI สำหรับการแสดงผลข้อมูลแบบ Graph

๔.๒ ผลการทดสอบแพลตฟอร์มในห้องปฏิบัติการ

ผลการทดสอบแพลตฟอร์ม CYBLION มีด้วยกันหลายส่วน ดังนี้

๔.๒.๑ ผลการทดสอบพีเจอรการใช้งาน

ในการทดสอบใช้งานที่นำเสนอในรายงานนี้เป็นพีเจอรหลักซึ่งเพียงส่วนหนึ่งภายหลังการทดสอบจาก Tester ได้ทำการทดสอบทุกๆ ฟังก์ชันแล้ว

ตารางที่ ๑๔ ผลการทดสอบพีเจอรการใช้งานแพลตฟอร์ม

พีเจอรการใช้งาน	ผลการทดสอบ	พีเจอรการใช้งาน	ผลการทดสอบ
การเข้ารหัสข้อมูลบน Edge Computing	ผ่าน	การส่งข้อมูลเข้ารหัสไปยัง Mobile application	ผ่าน
การรับกุญแจรหัสลับสำหรับเข้ารหัส Edge Computing	ผ่าน	การสร้างฟังก์ชันการคำนวณในคลาวด์	ผ่าน
การส่งข้อมูลเข้ารหัสไปยังคลาวด์	ผ่าน	การลงทะเบียนเข้าใช้งาน Mobile application	ผ่าน
คลาวด์รับข้อมูลเข้ารหัสจาก Edge computing แสดง Topic ตรงตามต้นทาง	ผ่าน	การถอดรหัสข้อมูลเพื่อแสดงผลใน Mobile application	ผ่าน
การคลาวด์รับกุญแจรหัสลับสำหรับคำนวณ	ผ่าน	การเลือกขนาดพารามิเตอร์สร้างสภาพแวดล้อมโฮโมมอร์ฟิก	ผ่าน
การลงทะเบียนเข้าใช้งานคลาวด์	ผ่าน	การสร้างกุญแจรหัสลับ	ผ่าน

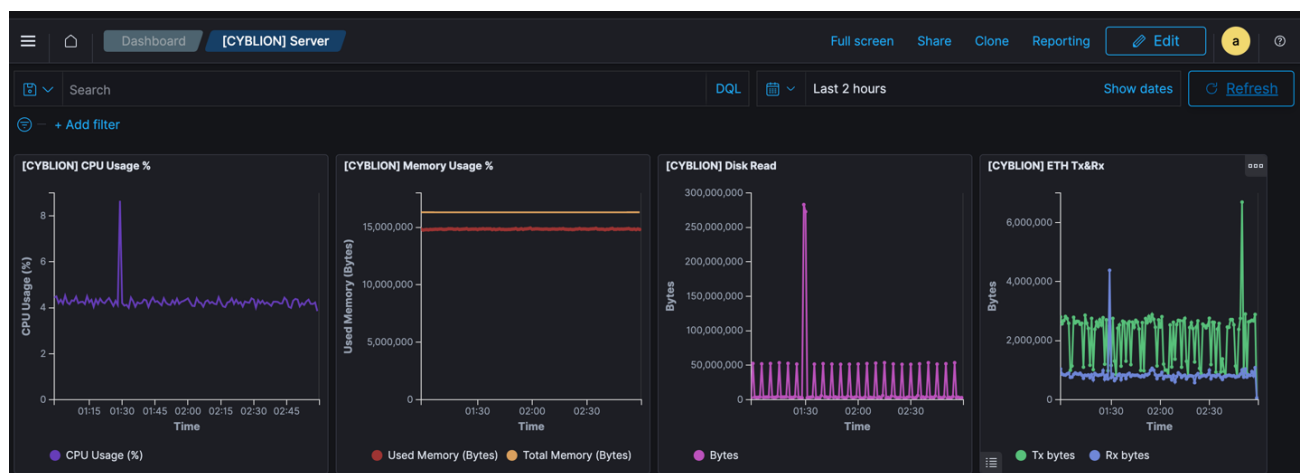
ตารางที่ ๑๕ ผลการทดสอบพีเจอรการใช้งาน Mobile App

พีเจอรการทำงานของ Mobile App	Link VDO
การแสดงหน้าโพรไฟล์ของผู้ใช้งาน	https://youtu.be/T2xvhvn7u-s
การสร้างโปรเจ็ค	https://youtu.be/pj_xGAwhcSI
การลบโปรเจ็ค	https://youtu.be/xemFj8wrHcA
การสร้าง Edge Device	https://youtu.be/36WJ8tbUICA
การลบ Edge Device	https://youtu.be/tueRFbm_msA
การสร้าง Key Generating	https://youtu.be/DJuHw1mCTsQ
การสร้าง Key ทับ Key เดิม	https://youtu.be/sAOPnEYBiew
การส่ง Public Key ไปยัง Edge Device (Pair EDGE)	https://youtu.be/fRGnUNpZrq0
การส่ง Relinear Key ไปยังคลาวด์ FHE-Cloud (SEND KEY)	https://youtu.be/7FnVwW95vhM
แสดงผล Device List	https://youtu.be/6v1-Dn3B8b4
DATA - ADD PANE	https://youtu.be/3LhCmWkDGmk
DATA - SAVE FREEBOARD	https://youtu.be/Sk_Va5Lgloo

ฟีเจอร์การทำงานของ Mobile App	Link VDO
DATA - DISABLE SCROLL	https://youtu.be/WSXQstAxRqk
DATA - WIDGET Type: Text	https://youtu.be/j4mx6tEctPo
DATA - WIDGET Type: Gauge	https://youtu.be/nlhXyFUM5o8
DATA - WIDGET Type: ApexCharts Line Charts	https://youtu.be/2PUDHp2LgTM

๔.๒.๒ ผลการทดสอบประสิทธิภาพของแพลตฟอร์ม

ผลการทดสอบปริมาณทรัพยากรที่ใช้รองรับผู้ใช้งานในคลาวด์ ได้แก่ CPU utilization, Memory, Disk Read และ Tx/Rx ปริมาณ Tx/Rx ที่เกิดขึ้นเนื่องจากการส่งข้อมูลของ Edge computing ไปยัง MQTT broker ที่อยู่ภายในคลาวด์ และรวมไปถึงกราฟฟิคการทำงานของ Mobile Application ไปยัง Mobile Gateway เป็นดังรูปที่ ๘๓ จะเห็นได้ว่า CPU ที่ถูกใช้งานไปอยู่ในระดับต่ำ แต่ส่วนทรัพยากรที่ถูกใช้เป็นจำนวนมากคือ Memory



รูปที่ ๘๓ ทรัพยากรที่ถูกใช้งานในคลาวด์

๔.๒.๓ ผลการทดสอบของพารามิเตอร์ของการเข้ารหัสและขนาดของกุญแจ

ผลการทดสอบขนาดของพารามิเตอร์ ขนาดกุญแจ และขนาดของข้อมูลเข้ารหัสด้วยกุญแจแยกตาม Small, Medium และ Large ดังตารางที่ ๑๖ จากการทดสอบพบว่าขนาดของข้อมูล ขนาดของกุญแจสำหรับการเลือกใช้พารามิเตอร์ Large มีขนาดใหญ่มาก ส่งผลกระทบต่อการส่งข้อมูลผ่านช่องสัญญาณแบนด์วิดท์ต่ำ

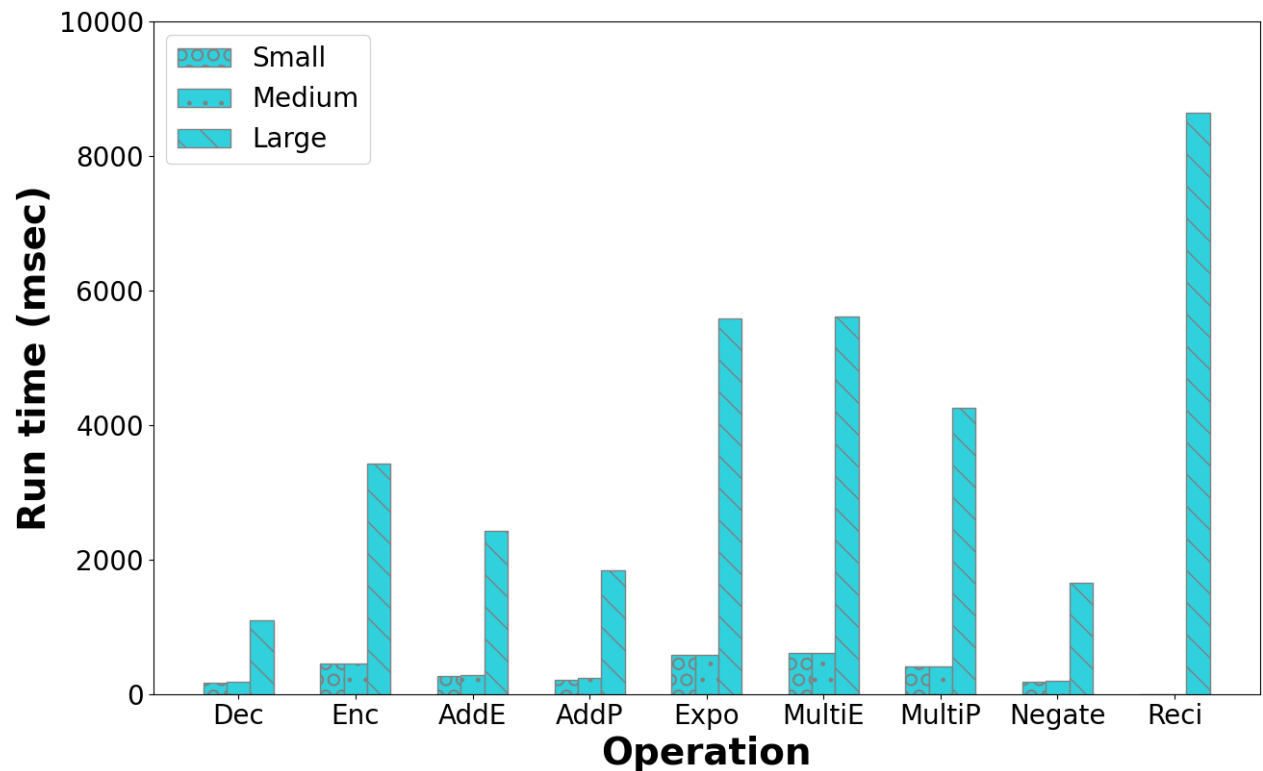
ตารางที่ ๑๖ พารามิเตอร์สำหรับสร้างสภาพแวดล้อมโฮโมมอร์ฟิกแบบ CKKS/RNS-CKKS

ขนาดของพารามิเตอร์	โพลีโมดูลัส	โคเอฟฟิเชียนต์โมดูลัส	สเกลเพคเตอร์	ขนาดข้อมูลที่เข้ารหัส	ขนาดกุญแจรหัสลับ (kByte)		
					Encryption	Relinearization	Decryption
Small	8192	[250,230,230,250]	230	356kB	258	771	255

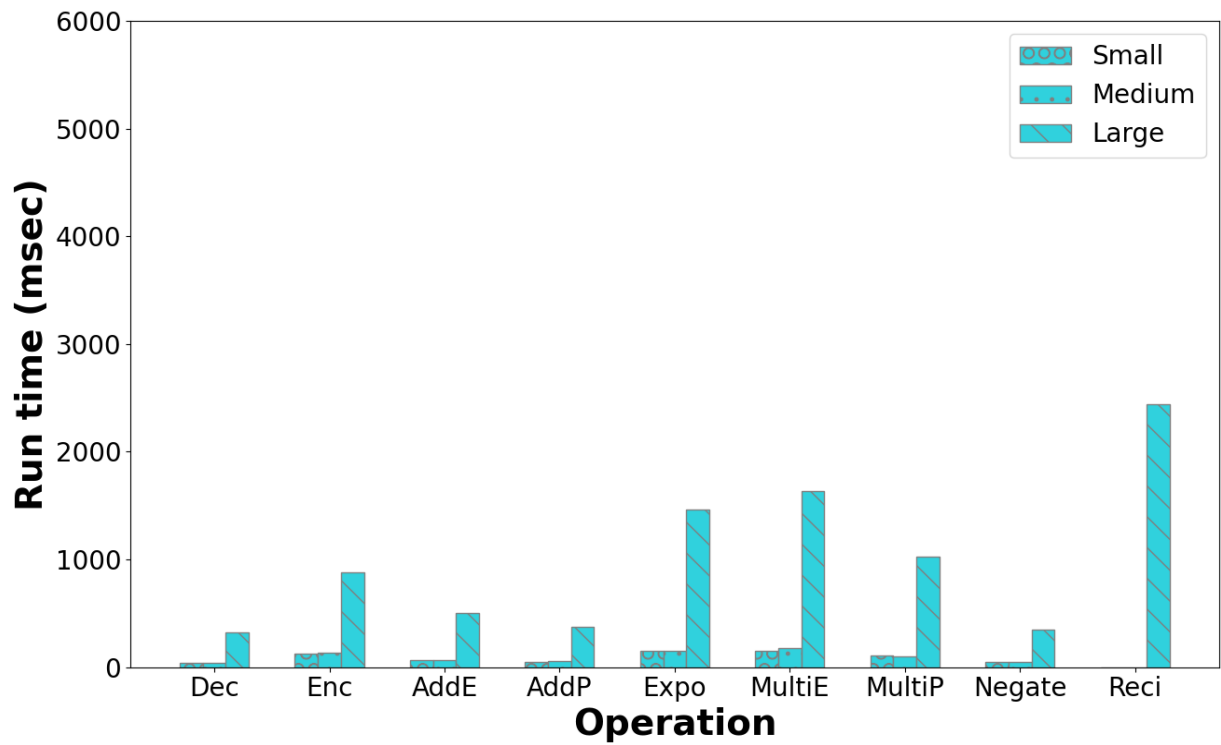
140 kB							
Medium 146 kB	8192	[260,240,240,260]	240	446kB	314	940	314
Large 174 kB	1638 4	[260,240,240,240,240,240,240,240,260]	240	2.2MB	1,300	10,200	1,300

๔.๒.๔ ผลการทดสอบประสิทธิภาพของพารามิเตอร์ของการเข้ารหัสและขนาดของกุญแจ

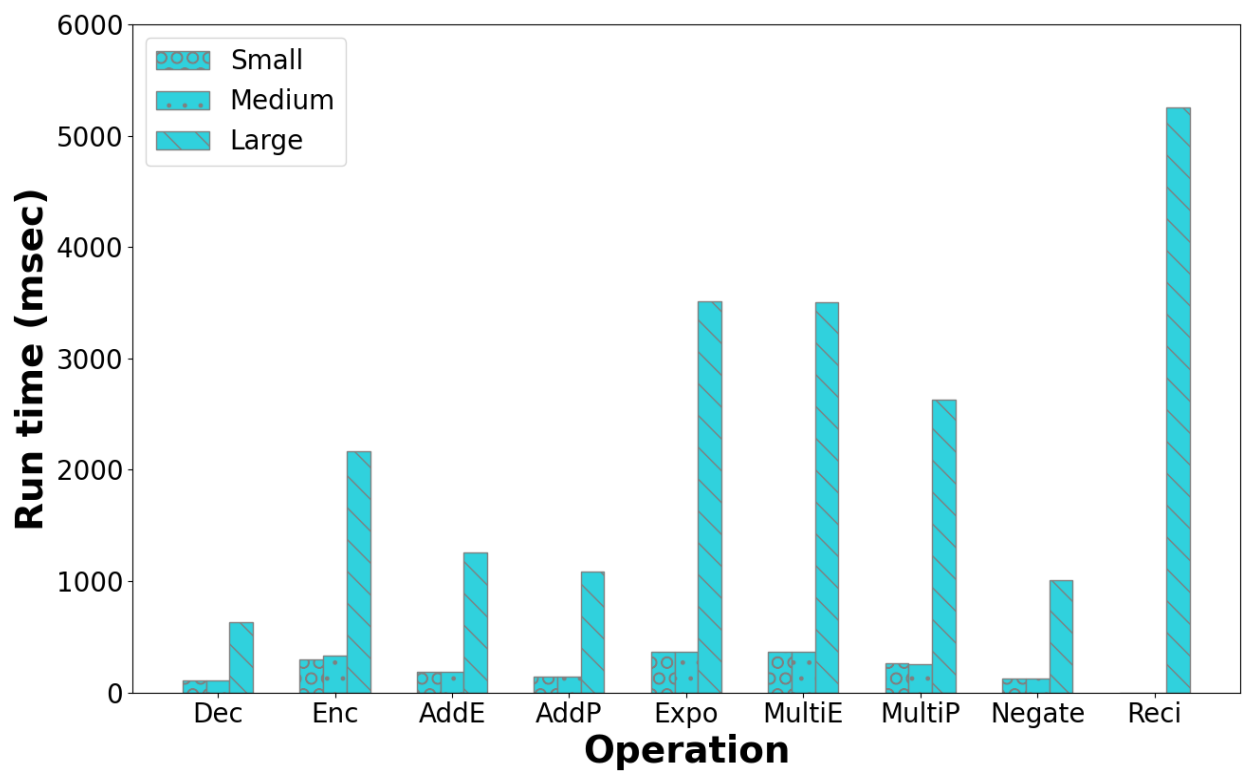
ผลการทดสอบความเร็วในการประมวลฟังก์ชัน HE บนอุปกรณ์ Edge device ประเภทต่างๆ ได้แก่ IoT ๒๐๕๐, Raspberry Pi 3 Model B+, Raspberry Pi 4, Jestson Nana และบนคลาวด์ เป็นดังรูปที่ ๘๔ ถึง ๘๘



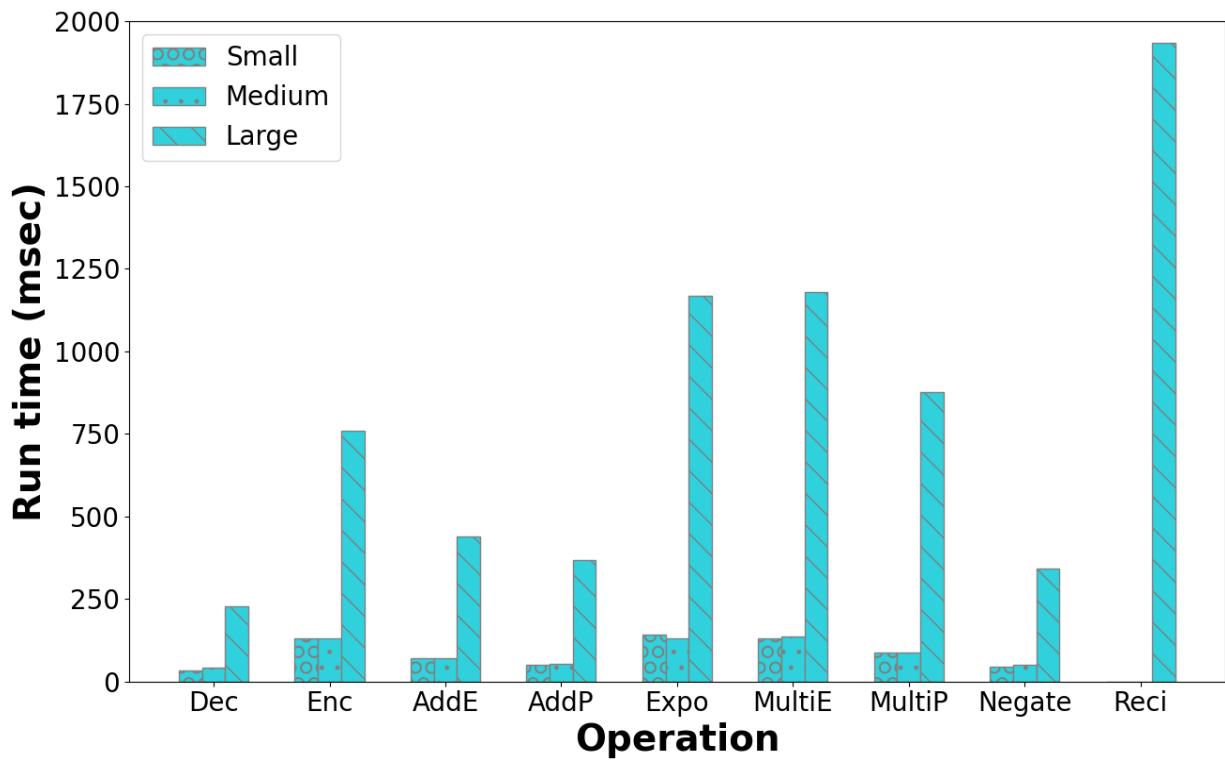
รูปที่ ๘๔ ความเร็วในการประมวลผลฟังก์ชัน HE บน IoT 2050



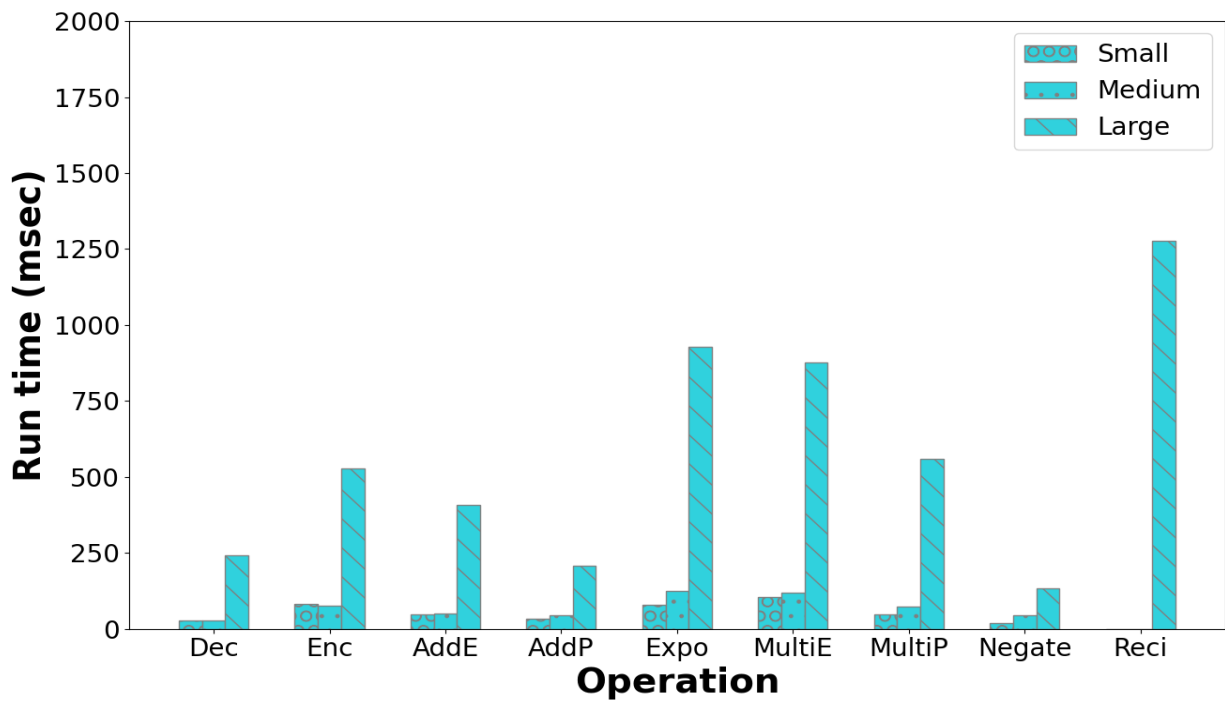
รูปที่ ๘๕ ความเร็วในการประมวลผลฟังก์ชัน HE บน. Jetson Nano



รูปที่ ๘๖ ความเร็วในการประมวลผลฟังก์ชัน HE บน. Raspberry Pi 3 Model B



รูปที่ ๘๗ ความเร็วในการประมวลผลฟังก์ชัน HE บน.Raspberry Pi 4



รูปที่ ๘๘ ความเร็วในการประมวลผลฟังก์ชัน HE บน.Cloud

๔.๓ ผลการทดสอบแพลตฟอร์มภาคสนาม

การทดสอบการใช้งานแพลตฟอร์มไอโอทีคลาวด์ที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัส (CYBLION) ได้ทำการร่วมทดสอบกับบริษัทเอกชน ดังนี้

๑. บริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด และ

๒. บริษัท เดอะ เพ็ท จำกัด

โดยมีรายละเอียดของการทดสอบแยกตามบริษัท ดังนี้

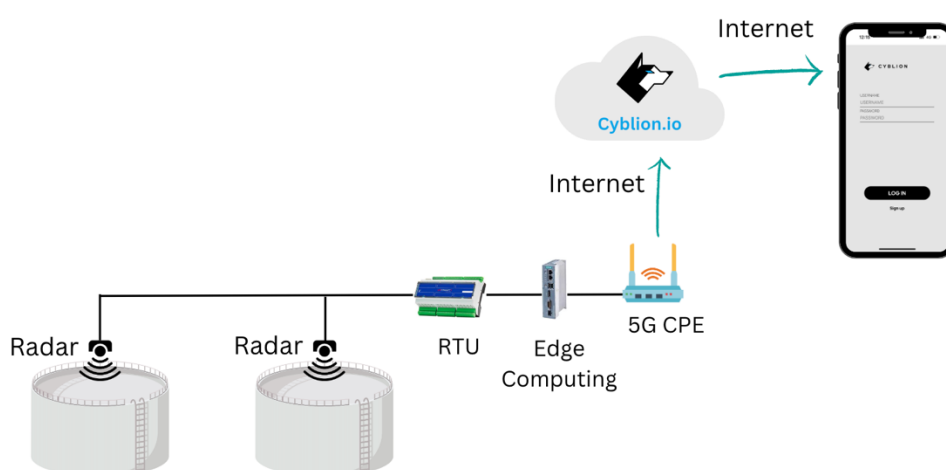
๔.๓.๑ การทดสอบใช้งานแพลตฟอร์ม Cyblion กับบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด

๑) วัตถุประสงค์

- เพื่อทำการออกแบบและติดตั้งระบบ IoT สำหรับตรวจวัดค่าระดับของน้ำมันในไซลอน้ำมันจำนวน ๒ ไซโล ด้วย Radar Sensor สองชนิด
- เพื่อทดสอบการใช้งานแพลตฟอร์มไอโอทีคลาวด์ที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัส (Cyblion) ระบบ 5G Smart Tank Farm สำหรับการตรวจวัดระดับน้ำมันในไซโลแบบเรียลไทม์
- เพื่อทดสอบประสิทธิภาพของการส่งข้อมูลแบบเข้ารหัสด้วยเทคนิคโฮโมมอร์ฟิกแบบ CKKS/RNS-CKKS ด้วยระบบการสื่อสารไร้สาย 5G
- เพื่อเปรียบเทียบการใช้งานของแพลตฟอร์มไอโอทีคลาวด์ที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัส (Cyblion) กับ NETPIE IoT Cloud Platform ที่ให้บริการไอโอทีคลาวด์แบบไม่เข้ารหัสข้อมูล

๒) การออกแบบในห้องปฏิบัติการ

การออกแบบระบบ IoT สำหรับตรวจวัดค่าระดับของน้ำมันในไซลอน้ำมันจำนวน ๒ ไซโล โดยข้อมูลจาก Sensor ถูกส่งไปยัง IoT cloud เพื่อทำการประมวลผล และข้อมูลที่ได้รับการประมวลผลแล้วถูกส่งไปแสดงผลผ่าน Mobile Application ผ่านเครือข่าย 5G เป็นดังรูปที่ ๘๙ โดย Sensor ทำการวัดระยะจากตัว Sensor ถึงระดับน้ำมันในไซโลด้วยระบบเรดาร์ โดยตัวเซนเซอร์ยังส่งอินฟราเรดลงไปกระทบน้ำมันในไซโลและรอรับแสงที่สะท้อนกลับมายังตัวเซนเซอร์



รูปที่ ๘๙ การเชื่อมต่อระบบไอโอทีสำหรับการมอนิเตอร์ระดับน้ำมันในไซลอน้ำมันกับแพลตฟอร์มไอโอทีคลาวด์

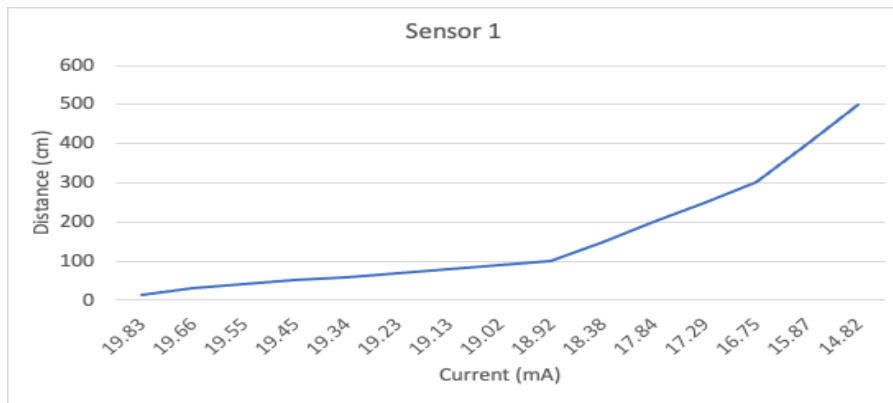
ในการทดสอบติดตั้งเซนเซอร์บนถังน้ำมันในโรงงานของ บริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด ในการทดสอบนี้นักวิจัยได้เลือกใช้ Sensor ๒ ยี่ห้อ ได้แก่

- VEGA Radar ที่ใช้โปรโตคอลในการสื่อสารด้วย ๔-๒๐ mA (Heart protocol)
- Irissa Radar ที่ใช้การส่งข้อมูลเป็นระยะทางระดับความละเอียดขนาดไมโครเมตร ด้วยรูปแบบการสื่อสารผ่าน UART
- เนื่องจาก Sensor VEGA Radar ส่งข้อมูลเป็นสัญญาณอะนาล็อกกระแสไฟฟ้า ๔-๒๐ mA ทำให้ต้องแปลงค่ากระแสไฟฟ้า ๔-๒๐ mA ให้อยู่ในรูปแบบข้อมูล Modbus TCP ด้วย uRTU (uRconnect) และให้ Edge Computing ทำการแปลงค่า Modbus TCP เป็น MQTT protocol เพื่อส่งข้อมูลไปยัง IoT Cloud Platform ในการทดสอบนี้ได้กำหนดให้ Edge Computing ส่งข้อมูล ๒ ชนิด
- ข้อมูลที่ไม่มีการเข้ารหัสไปยัง NETPIE Cloud ในรูปแบบของข้อมูล MQTT โดยไม่มีการเข้ารหัสเพื่อไปคำนวณบนคลาวด์เพื่อทำการแปลงค่าระยะห่างของเซนเซอร์กับของเหลวในไซโลน้ำมันเป็นค่าปริมาตรของน้ำมันในไซโล และส่งผลการคำนวณไปยัง Mobile Application โดยข้อมูลที่เข้ารหัสมีขนาดเล็กประมาณ ๑๖ KB
- ข้อมูลที่มีการเข้ารหัสที่ Edge Computing ในรูปแบบ Homomorphic Encryption เพื่อป้องกันข้อมูลรั่วไหล โดยส่งข้อมูลไปยัง Cyblion Cloud เพื่อทำการคำนวณ และส่งผลการคำนวณไปยัง Mobile Application โดยขนาดของข้อมูลที่เข้ารหัสมีขนาดประมาณ ๓๖๕ KB

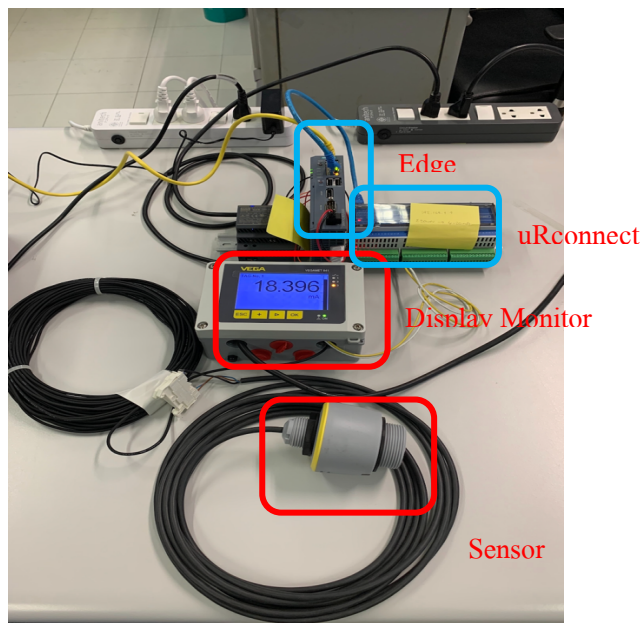
สูตรการคำนวณสำหรับการแปลงค่า ๔-๒๐ mA ไปเป็นระยะทางนั้นเป็นสมการเชิงเส้น เนื่องจากระยะห่างของ Sensor และน้ำมันแปรผันตรงกับค่าปริมาตรน้ำมันในไซโล โดยตั้งค่า offset ที่ระยะต่ำสุด ๔ mA ที่ ๔ เมตรและสูงสุด ๒๐ mA ที่ ๑๕ เมตร

๓) การทดสอบในห้องปฏิบัติการ (ก่อนนำไปติดตั้งในพื้นที่โรงงานจริง)

เมื่อได้ทำการทดสอบค่าที่ได้จาก Sensor VEGA ในห้องปฏิบัติการเทียบกับระยะทางเป็นดังรูป ซึ่งทำให้สามารถประมาณค่าเป็น Function $f(x) = ๖.๒๕x - ๒๕$ เมื่อ x เป็นค่า Sensor ที่วัดได้ในหน่วย mA แต่อย่างไรก็ตามในการทดลองในห้องปฏิบัติการไม่ได้ทดสอบในระบบปิด ซึ่งเป็นการทดสอบในพื้นที่เปิดโล่ง โดยการวัดระยะระหว่าง Sensor และผนังที่วัดจึงทำให้มีผลความผิดพลาดจากการทดลองเนื่องจากค่าการสูญเสียทางแสงจากปัจจัยภายนอก ผลการทดสอบหาความสัมพันธ์ระหว่าง Sensor และระยะทางเป็นดังรูปที่ ๙๐

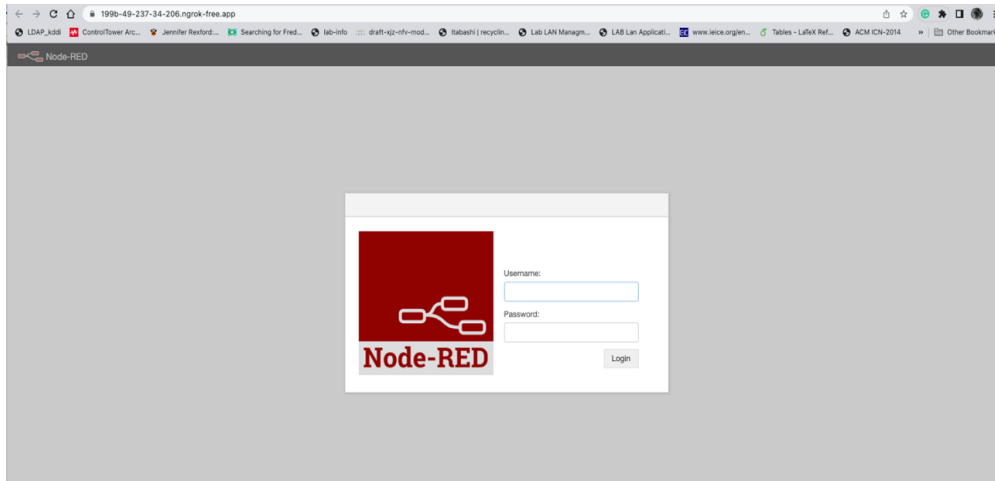


รูปที่ ๙๐ ความสัมพันธ์ระหว่างค่า Sensor และระยะทาง



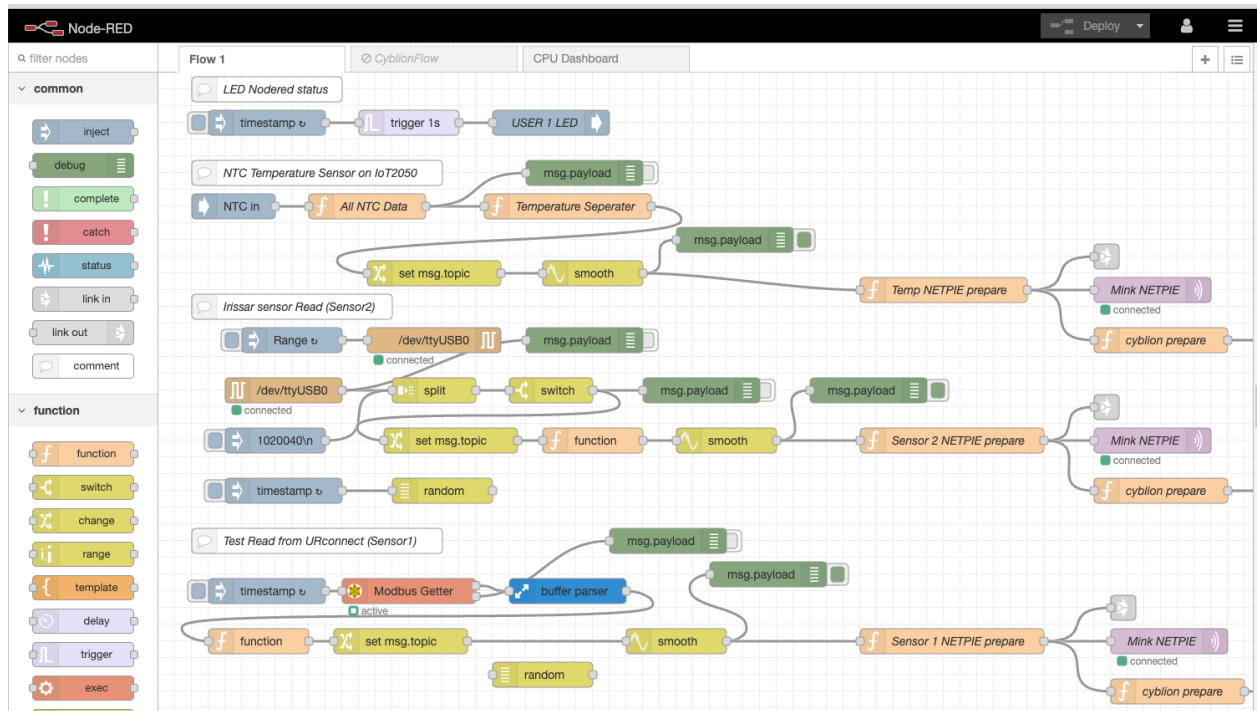
รูปที่ ๙๑ การทดสอบระบบ IoT ในห้องปฏิบัติการ

จากนั้นทำการสร้าง Data Flow ใน Node-RED ใน Edge Computing โดยทำการเชื่อมต่อผ่าน port 1880 ของ IP address ของอุปกรณ์ Edge Computing โดยทางนักวิจัยได้ออกแบบให้มีหน้า Login สำหรับการเข้าถึง Node-RED ใน Edge Computing เพิ่มเติมจากการติดตั้ง Node-RED local ทั่วไป ทั้งนี้เพื่อป้องกันผู้ไม่ได้รับอนุญาตให้เข้าถึงข้อมูลทำการเปลี่ยนแปลง Node-RED flow ได้

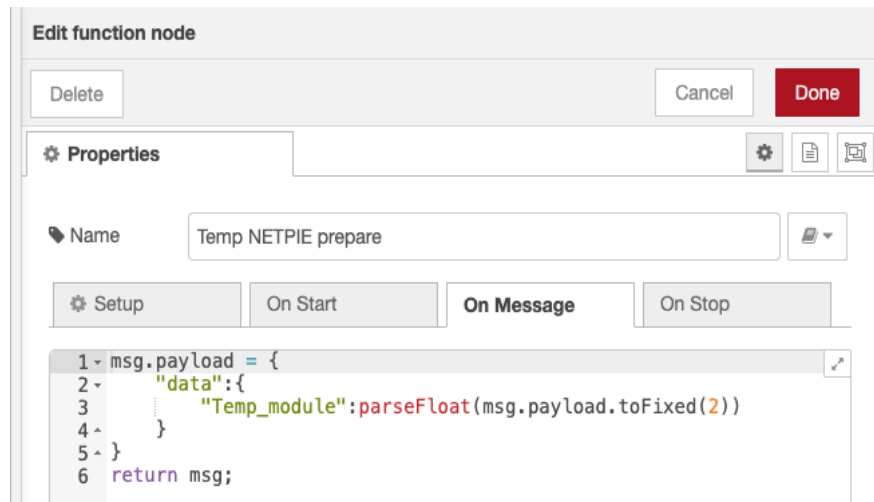


รูปที่ ๔๒ หน้าต่างการเข้าถึงโปรแกรม Node-RED ใน Edge Computing

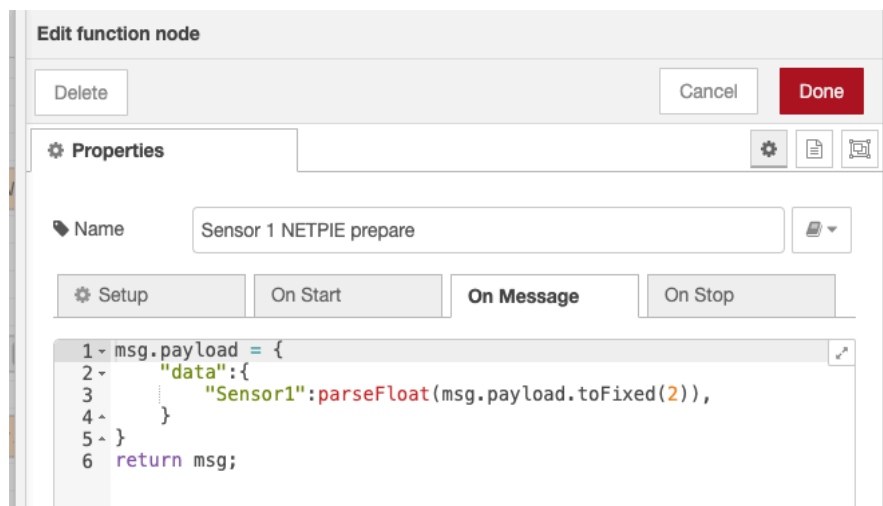
จากนั้นทำการสร้าง Data Flow ใน Node-RED ให้ดึงข้อมูลจาก Sensor ผ่าน Modbus TCP protocol ที่ได้รับจาก uRConnect และทำการแปลงให้อยู่ในรูปแบบ MQTT protocol เพื่อส่งข้อมูลเข้าสู่ NETPIE IoT cloud เพื่อนำข้อมูลมาเปรียบเทียบระหว่างข้อมูลที่ได้รับแบบไม่เข้ารหัส และข้อมูลที่ได้รับจากแพลตฟอร์ม CYBLION ที่เข้ารหัส ดังรูปที่ ๔๓



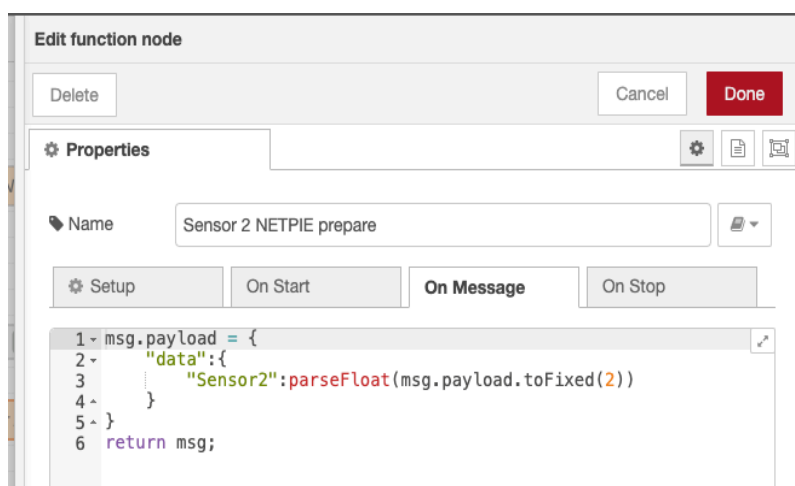
รูปที่ ๔๓ การสร้าง Data Flow สำหรับการส่งข้อมูล Sensor ไปยัง NETPIE IoT Cloud



รูปที่ ๔๔ การกำหนด Payload สำหรับ Topic “Temp_module”



รูปที่ ๔๕ การกำหนด Payload สำหรับ Topic “Sensor1”



รูปที่ ๔๖ การกำหนด Payload สำหรับ Topic “Sensor2”

Edit mqtt out node

Delete Cancel Done

Properties

Server: Sensor1

Topic: @shadow/data/update

QoS: 0 Retain: ☒

Name: Mink NETPIE

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

รูปที่ ๙๗ กำหนดให้ส่งข้อมูลเข้าสู่ @shadow/data/update ของ NETPIE

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Properties

Name: Sensor1

Connection

Server: mqtt.netpie.io Port: 1883

☐ Use TLS

Protocol: MQTT V3.1.1

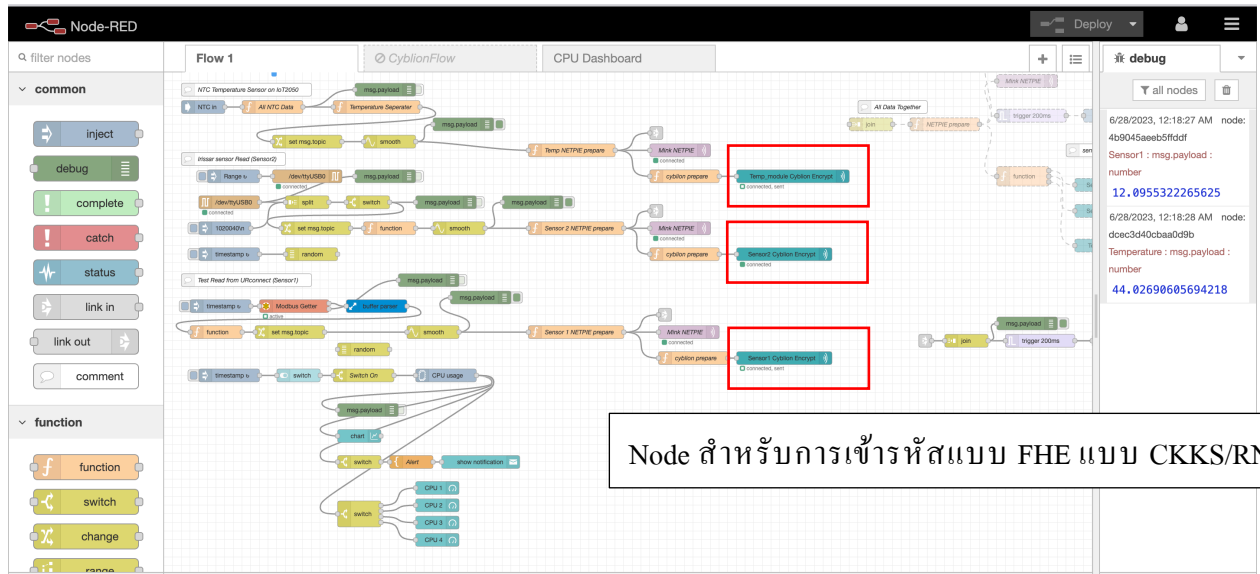
Client ID: a271ed1e-1d19-40e3-899b-0c70d6075f8a

Keep Alive: 60

Session: ☒ Use clean session

รูปที่ ๙๘ กำหนดให้ส่งข้อมูลเข้าสู่ MQTT Broker ของ NETPIE

จากนั้นทำการติดตั้ง Node สำหรับเข้ารหัสข้อมูลแบบ HE แบบ RNS-CKKS เพิ่ม ใน Node-RED flow ดังรูปที่ ๙๙ โดยการเพิ่มเพียง Node เดียวที่ปลายทางของข้อมูลไอโอที่ที่ต้องการส่งออก



รูปที่ ๙๙ การติดตั้ง Node สำหรับการเข้ารหัสเพิ่มเติมสำหรับการเข้ารหัสข้อมูลไอโอที และส่งออกไปยังคลาวด์ Cyblion

Edit Cyblion Encrypt node

Delete Cancel Done

Properties

Name: Sensor1 Cyblion Encrypt

Device: IoT2050

sync connect

Sync or Edit Device

MQTT TOPIC

☒ Send the encrypted value to

Topic: @cstore/data/updated/ oil/Sensor1

☒ Send the encrypted timestamp to

Topic: @cstore/data/updated/ oil/time_s1

Example encrypted value topic

@cstore/data/updated/home/bedroom/temperature

@cstore/data/updated/oil/height

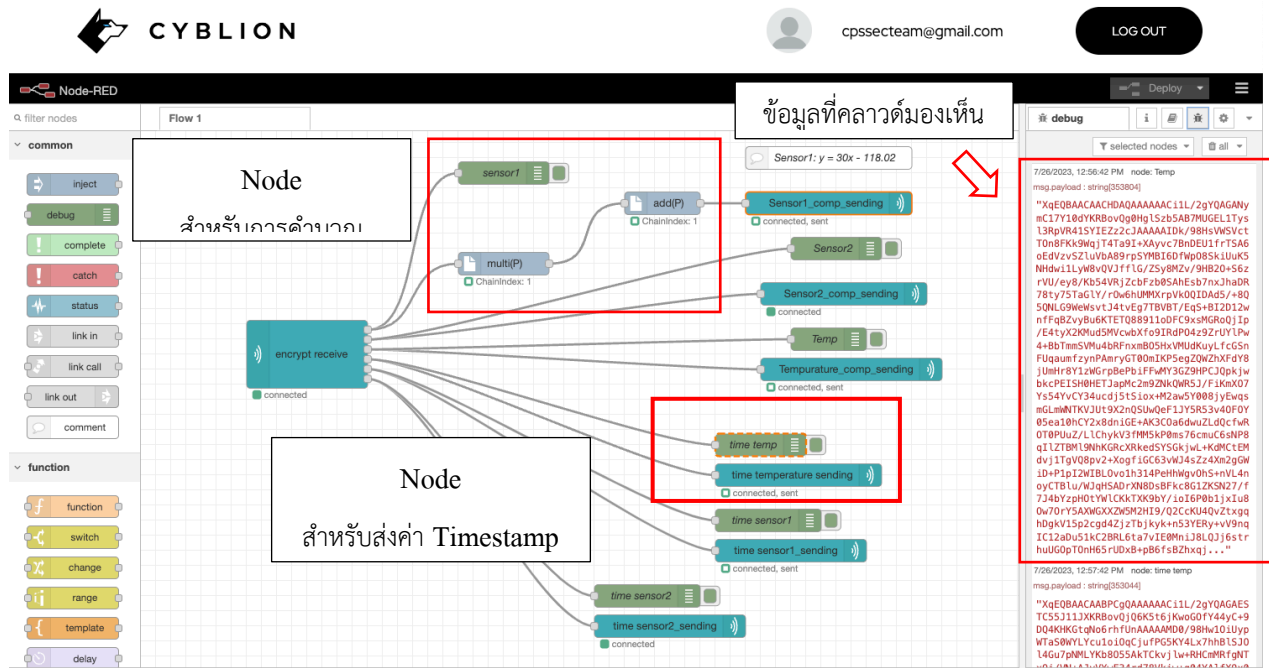
Example encrypted timestamp topic

@cstore/data/updated/home/bedroom/time

@cstore/data/updated/oil/time

Enabled

รูปที่ ๑๐๐ การตั้ง Topic สำหรับส่งข้อมูลเข้ารหัสเพื่อส่งออกไปยังคลาวด์ Cyblion ผ่าน EdgeBroker @cstore/data/updated/

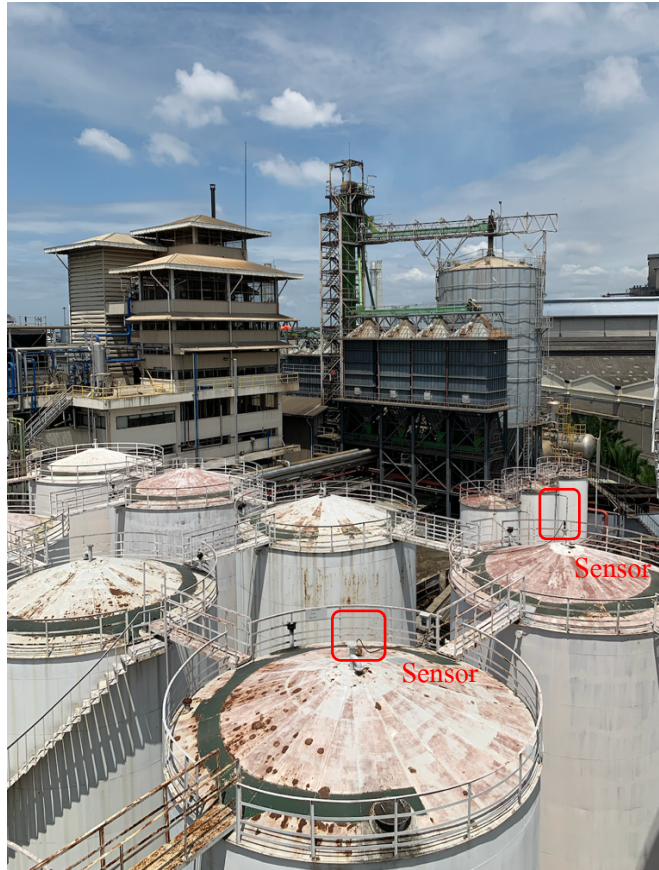


รูปที่ ๑๐๑ การสร้างสูตรการคำนวณบนคลาวด์ Cyblion โดยข้อมูลที่ต้องการคำนวณจะผ่าน Flow ของการคำนวณและ ข้อมูล Timestamp จะไม่ผ่าน Flow การคำนวณ

ผลการทดสอบส่งค่า IoT ไปยัง Cloud และแสดงผลความหน่วงต่ำซึ่งอยู่ในระดับต่ำกว่า ๑ msec เนื่องจากในพื้นที่ห้องปฏิบัติการมีสัญญาณ 5G สูง ค่าความหน่วงของข้อมูลไม่ส่งผลกระทบต่อผู้ใช้งานเนื่องจากต่ำกว่าค่าระยะเวลาที่ต้องการส่งข้อมูล โดยกำหนดให้ส่งข้อมูลทุกๆ ๓๐ วินาที ถึง ๑ นาที

๔) การติดตั้งภาคสนาม (ในพื้นที่โรงงาน TVOP)

การดำเนินงานติดตั้งชุดอุปกรณ์ 5G Smart Tank Farm ในโรงงาน TVOP ส่วนของ Tank น้ำมัน และเข้าตรวจสอบและ Configure อุปกรณ์ Edge Computing เพื่อให้รับค่า Sensor จาก และตรวจสอบความถูกต้องของค่า Sensor ที่รับมาจาก Sensor และค่าที่แสดงผลบน IoT Cloud Platform (NETPIE) บน Dashboard ในรูปแบบ Web Application และ Mobile Application โดยภายใน Edge Computing (SIEMENS IoT2050) ได้ติดตั้งโปรแกรม Node-RED สำหรับการแปลงค่าและเข้ารหัส และส่งค่าที่ต้องการไปยังคลาวด์



รูปที่ ๑๐๒ การติดตั้ง Sensor บนไซโลน้ำมันขนาดความสูง ๑๕ เมตร



รูปที่ ๑๐๓ การติดตั้ง Sensor บนไซโลน้ำมันขนาดความสูง ๑๕ เมตร



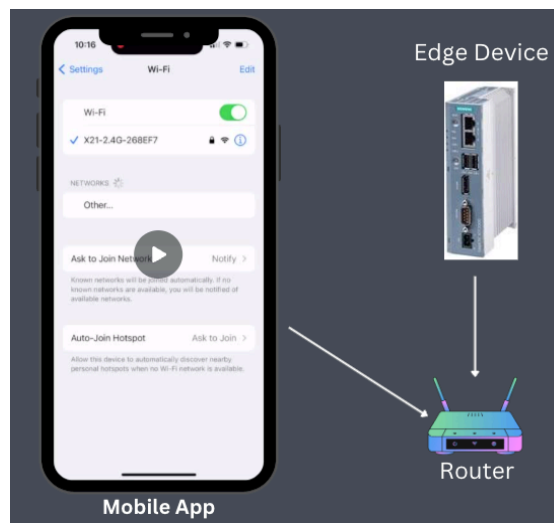
รูปที่ ๑๐๔ การติดตั้งอุปกรณ์สำหรับ Edge Computing ใกลิโซโลน้ำมันที่ต้องการตรวจวัด



รูปที่ ๑๐๕ ตู้เก็บอุปกรณ์สำหรับ Edge Computing ที่ออกแบบให้ทนความร้อนและความชื้น เพื่อติดตั้งกลางแจ้ง มีระบบไฟฟ้าสำหรับจ่ายอุปกรณ์ IT ต่างๆ ภายในตู้และพัฒนาระบายความร้อน 4 ตัวรอบทิศทาง



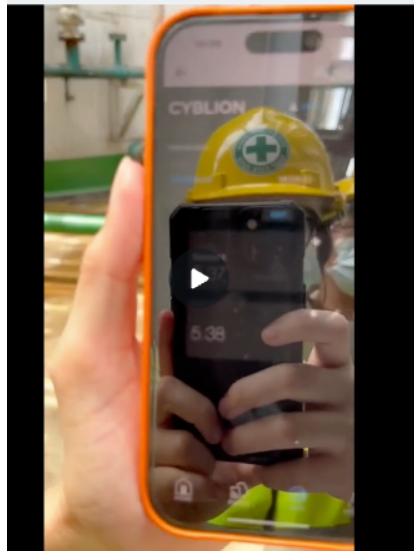
รูปที่ ๑๐๖ การเข้า Configuration ในพื้นที่หน้างาน



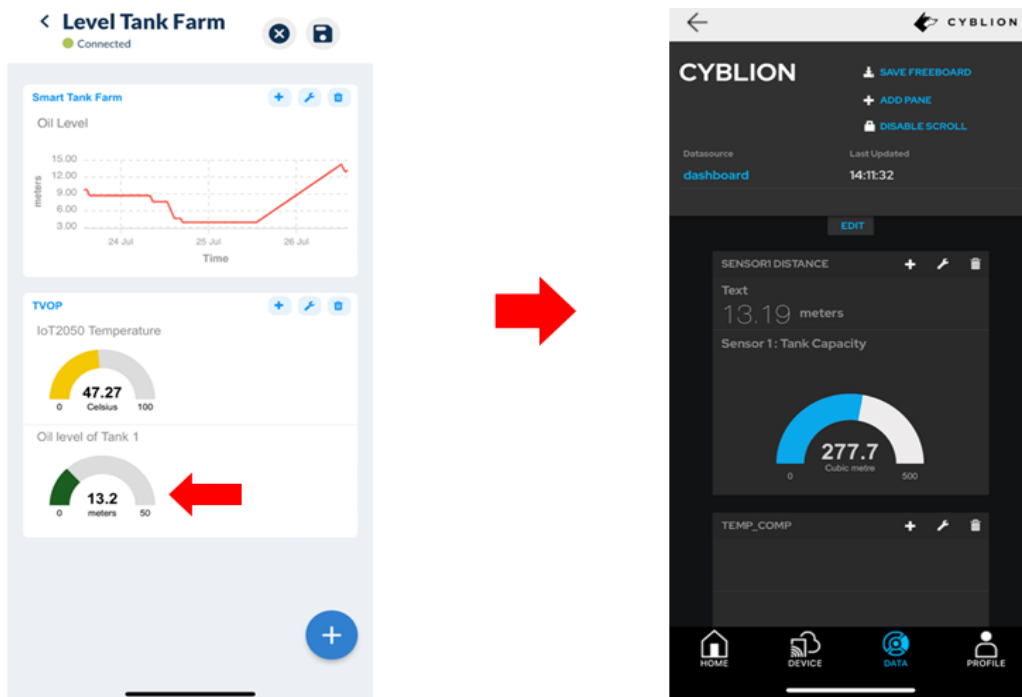
รูปที่ ๑๐๗ การเข้าเชื่อมต่อ Network กับ Edge Computing เพื่อการแลกเปลี่ยน



รูปที่ ๑๐๘ ตรวจสอบความแม่นยำในการถอดรหัสจากจอมอนิเตอร์ของ VEGA sensor



รูปที่ ๑๐๙ ตรวจสอบการรับค่ามาถอดรหัสและแสดงผลผ่าน Mobile Application



รูปที่ ๑๑๐ เปรียบเทียบค่าที่ได้จาก Cyblion App

ปัญหาและอุปสรรค

๑. ค่า Sensor ของ Irrissa ไม่เสถียรเนื่องจากการติดตั้งไม่สมบูรณ์ เนื่องจากระบบที่ติดตั้งเป็น Outdoor บนที่มีความร้อนและโดนความชื้นจากฝน แต่การติดตั้งมีรอยรั่วทำให้น้ำเข้าไปยังตัว Sensor ทำให้ระบบเสียหาย จึงทำให้ไม่สามารถอ่านค่าได้

๒. สัญญาณ 5G ภายในโรงงานในบริเวณที่ทดสอบไม่เสถียร และโดนรบกวนจาก Tank สูงซึ่งส่งผลให้ข้อมูลเข้ารหัสที่มีขนาดใหญ่ที่ถูกส่งไปยังคลาวด์ไม่สมบูรณ์ (Packet Drop) และมีความหน่วงสูงส่งผลให้ Mobile Application ถอดรหัสผิดพลาด
๓. การเข้าไป Configure ค่าที่ต้องการเปลี่ยนแปลงการส่งข้อมูลใน Node-RED ใน Edge Computing ไม่สามารถทำได้บ่อยเนื่องจากต้องเข้าไปแก้ไขที่พื้นที่ ทางทีมนักวิจัยจึงได้ทำวาง node agent ไว้ใน Edge Computing เพื่อให้สามารถเข้าไป Configure ได้ระยะไกล
๔. การเข้าไปเปลี่ยนแปลง และตรวจเช็คระบบ ต้องการพนักงาน และวิศวกรในโรงงานผู้ดูแลให้ความร่วมมือในการพาเข้าพื้นที่
๕. การเข้าไปตรวจเช็คความเสียหายของ Sensor ต้องรอการติดต่อกลับจากผู้ผลิต
๖. ต้องทำการรีเซตระบบ Edge Computing เมื่อเกิดความเสียหายอันเนื่องจากระบบ disconnect โดยมี ๒ วิธี Remote reboot ในกรณี Node-Red ใน Edge computing มีความผิดพลาดไม่ส่งข้อมูล และ Reset ระบบแบบ Switch off เมื่อ Sensor ไม่ส่งข้อมูล

๔.๓.๒ การทดสอบใช้งานแพลตฟอร์ม Cyblion กับ บริษัท เดอะ เพ็ท จำกัด

๑) วัตถุประสงค์ของการดำเนินงาน

- เพื่อติดตั้งซอฟต์แวร์ให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT ให้กับบริษัท เดอะ เพ็ท จำกัด ซึ่งประกอบด้วย ๑) ซอฟต์แวร์เข้ารหัสข้อมูลที่ติดตั้งในอุปกรณ์ Edge Computing (ในที่นี้คือ อุปกรณ์ Mini PC) โดยอุปกรณ์ Edge Computing ดังกล่าวได้ติดตั้งระบบตรวจวัดข้อมูลของอุณหภูมิ และความชื้น ของระบบในโรงงานอุตสาหกรรม โดยซอฟต์แวร์เข้ารหัสข้อมูลจะทำหน้าที่เข้ารหัสข้อมูลก่อนการส่งข้อมูลอุณหภูมิ และความชื้น ก่อนส่งผ่านการสื่อสารไร้สาย WiFi ออกสู่ Internet ไปยัง IoT Cloud Platform ๒) ซอฟต์แวร์คำนวณข้อมูลที่เข้ารหัสแล้วบน IoT Cloud Platform ที่ทีมวิจัยได้พัฒนาขึ้น และ ๓) ซอฟต์แวร์สร้างกุญแจรหัสลับ และถอดรหัสข้อมูลและแสดงผล ที่ติดตั้งบน Smart Phone เพื่อใช้ตรวจสอบและสั่งการ โดยผู้มีอำนาจ และ/หรือ เชื่อมต่อและสั่งการให้เครื่องมือที่ต้องการควบคุมหยุดการทำงาน ชะลอการทำงาน หรือให้ผู้เกี่ยวข้องเข้าบำรุงรักษาก่อนเกิดความเสียหาย

- เพื่อส่งมอบ Account สำหรับใช้งานซอฟต์แวร์ให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล สำหรับระบบ IoT ให้กับโรงงานอุตสาหกรรม ๑ Account

- เพื่อเพิ่มการคุ้มครองความเป็นส่วนตัว ให้กับระบบควบคุมและตรวจสอบเครื่องจักร ป้องกันการถูกผู้ไม่พึงประสงค์บุกรุก หรือนำข้อมูลที่ส่งผ่านระบบ IoT ไปใช้ประโยชน์ในทางมิชอบ

- เพื่อคุ้มครองความเป็นส่วนตัวของข้อมูลของอุณหภูมิ และความชื้น ของระบบในโรงงานอุตสาหกรรมที่ส่งผ่านการสื่อสารไร้สาย WiFi ออกสู่ Internet เพื่อนำไปคำนวณบน IoT Cloud Platform ที่พัฒนาขึ้นของโครงการฯ แบบเข้ารหัส

- เพื่อให้ผู้ปฏิบัติงานของผู้ร่วมทดสอบสามารถสำรวจข้อมูล ของอุณหภูมิ และความชื้นของระบบ ในโรงงานอุตสาหกรรม ผ่านโมบายแอปพลิเคชันที่พัฒนาขึ้นของโครงการฯ เพื่อนำข้อมูลมาใช้ตรวจสอบ ควบคุม และจัดเก็บสำหรับการวิเคราะห์ วางแผนและสั่งการ เป็นต้น

- เพื่อเก็บข้อมูลก่อนและหลังการใช้งานระบบวัดอุณหภูมิ และความชื้น ด้วยเซนเซอร์ในประเด็นที่เกี่ยวข้องกับประสิทธิภาพการทำงานของแพลตฟอร์มที่พัฒนาขึ้น และประโยชน์ในการใช้งาน

- เพื่อสรุปผลการทดลองร่วมกันและกำหนดข้อมูลที่สามารถเผยแพร่ต่อสาธารณะ ในด้านผลการทดลองใช้แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ

๒) การออกแบบการทดสอบในห้องปฏิบัติการ

เนื่องจากโจทย์ของบริษัท The PET เป็นการนำแพลตฟอร์ม Cyblion ไปใช้งานต่อยอดจากการใช้งาน IoT ที่ติดตั้งใช้งานอยู่แล้วในโรงงานอุตสาหกรรม จึงทำให้ทางนักวิจัยไม่ต้องออกแบบระบบ IoT ให้กับโรงงานอุตสาหกรรมใหม่ การนำไปใช้งานจึงเป็นเพียงการติดตั้ง Node สำหรับเข้ารหัสแบบ FHE แบบ CKKS/RNS-CKKS เพิ่มเติมลงใน Data Flow เดิมใน Edge Computing ของโรงงานอุตสาหกรรม



รูปที่ ๑๑๑ การเข้าพื้นที่โรงงาน The PET

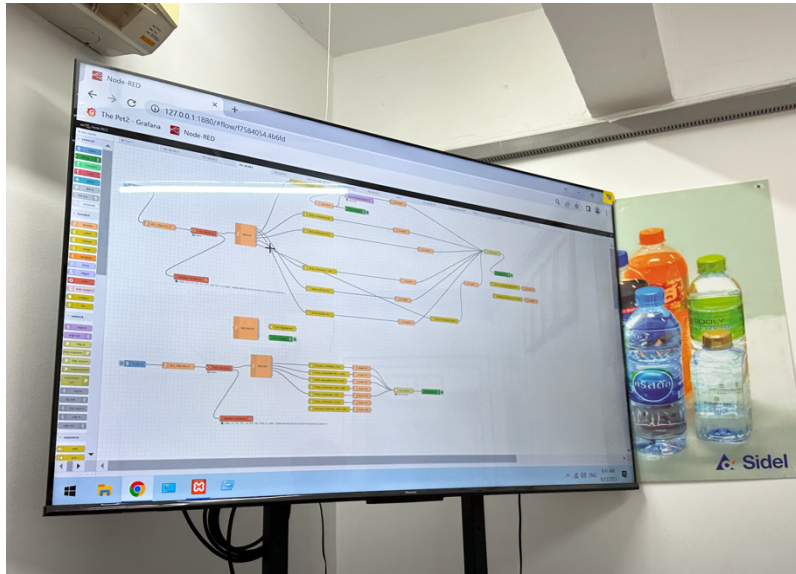


รูปที่ ๑๑๒ ร่วมหาหรือ Use Case ที่ต้องการใช้ทดสอบ

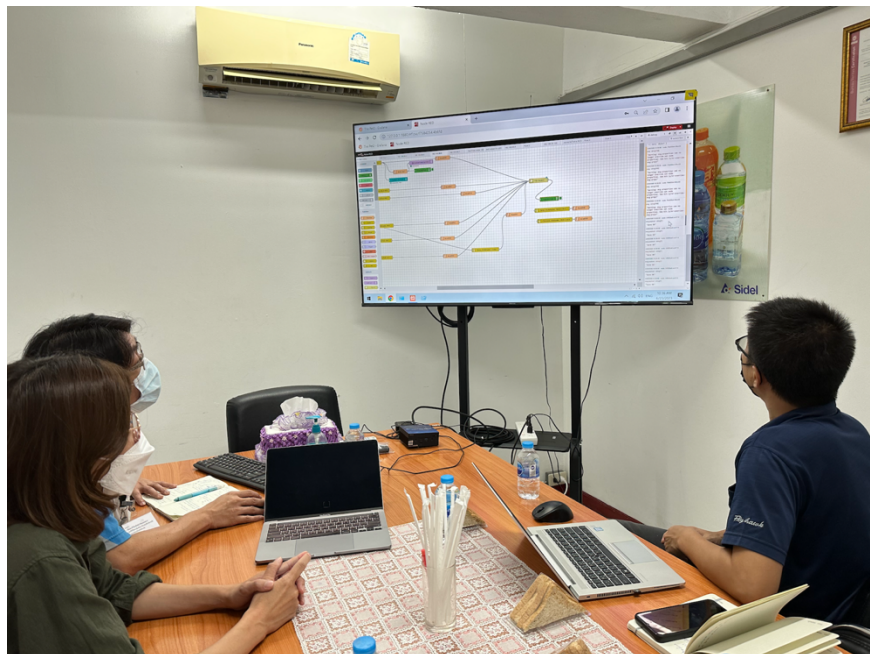
๓) การทดสอบภาคสนามในพื้นที่โรงงาน



รูปที่ ๑๑๓ Edge Computing ที่ใช้ติดตั้ง Node สำหรับเข้ารหัสทดสอบ



รูปที่ ๑๑๔ Data Flow เดิมที่สำหรับการตรวจวัดค่าเซนเซอร์ของโรงงาน The PET



รูปที่ ๑๑๕ นักวิจัยสอนการใช้งานติดตั้ง Node สำหรับการเข้ารหัสให้กับโรงงาน The PET



รูปที่ ๑๑๖ วิศวกรของโรงงาน The PET ติดตั้ง Mobile Application สำหรับใช้งาน



รูปที่ ๑๑๗ วิศวกรสามารถใช้งาน Mobile Application ได้และข้อมูลเข้ารหัสได้ถูกถอดรหัสและแสดงผลถูกต้องผ่าน Mobile Application

ปัญหาและอุปสรรค

๑. เนื่องจาก Edge computing ที่โรงงานใช้รวมการใช้งานของ Data Flow อื่นๆ ของโรงงานไว้ทำให้ไม่สามารถเข้าไปอัปเดตและตรวจเช็คข้อมูลได้หลังจากส่งมอบให้วิศวกรของโรงงานดำเนินงาน
๒. การอัปเดตหรือเปลี่ยนแปลงสูตรการคำนวณสามารถทำได้บนคลาวด์แต่ผลการคำนวณผู้ที่สามารถถอดรหัสได้มีแค่ Mobile Application ของวิศวกรโรงงานเท่านั้นทำให้กวีจียไม่สามารถตรวจเช็คข้อมูลได้ต่อเนื่อง
๓. การติดตั้งเปิดให้ดาวน์โหลด Mobile Application ใน Google Play Store เท่านั้น

บทที่ ๕

การเผยแพร่ผลการดำเนินงานวิจัยการ

ทีมวิจัยได้ดำเนินการเผยแพร่ผลการดำเนินโครงการ เพื่อสร้างความตระหนักรู้ถึงความปลอดภัยและความเป็นส่วนตัวของข้อมูลไอโอที แนวทางการป้องกัน รวมทั้งจัดทำสื่อสิ่งพิมพ์และสื่อวีดิทัศน์ เพื่อเป็นการเผยแพร่องค์ความรู้และตระหนักรู้ถึงความปลอดภัยและความเป็นส่วนตัวของข้อมูลไอโอที พร้อมทั้งเป็นการถ่ายทอดการใช้งานแพลตฟอร์มให้กับวิศวกรโรงงานอุตสาหกรรม นักพัฒนาไอโอที และผู้ประกอบการที่สนใจ โดยใช้แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที และระดมความคิดเห็นเพิ่มเติม เพื่อให้ทราบถึงความรู้และความเข้าใจที่เกี่ยวข้องกับ “ความตระหนักรู้ถึงความปลอดภัยและความเป็นส่วนตัวของข้อมูลไอโอที” โดยมีการดำเนินกิจกรรมดังนี้

๕.๑ การจัดฝึกอบรมเชิงปฏิบัติการ และการระดมความคิด

เพื่อถ่ายทอดความรู้ พัฒนาและฝึกปฏิบัติการใช้แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ที่พัฒนาขึ้น จากการดำเนินโครงการ และได้รับผลการจัดอบรมเชิงปฏิบัติการดังนี้

๑) การจัดอบรมเชิงปฏิบัติการ “Privacy as a Service For IoT/IoT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IoT ด้วย CYBLION Platform” วันศุกร์ที่ ๓ พฤศจิกายน ๒๕๖๖ เวลา ๐๙.๐๐ - ๑๖.๐๐ น. ณ ห้องประชุม Auditorium (CO-๑๑๓) อาคารสำนักงานกลาง อุทยานวิทยาศาสตร์ประเทศไทย

กลุ่มเป้าหมาย ผู้แทนจาก ภาคธุรกิจ/โรงงานอุตสาหกรรม ได้แก่ ผู้ใช้งานในภาคอุตสาหกรรมและผู้ใช้งานระดับนักพัฒนาในภาคอุตสาหกรรม ผู้ให้บริการ IoT Cloud Platform นักพัฒนาและผู้ผลิตอุปกรณ์ IoT มีหน่วยงานที่สนใจเข้ารับการอบรม ๑๒ หน่วยงาน และมีจำนวนผู้เข้ารับการอบรมจำนวน ๕๖ คน โดยมีกำหนดการ รายชื่อหน่วยงานและผู้เข้าร่วมอบรมเชิงปฏิบัติการ ตามภาคผนวก ก. และเอกสารประกอบการบรรยาย ตามภาคผนวก ข.

สรุปเนื้อหาการฝึกอบรมเชิงปฏิบัติการ

๑.๑) แนะนำ “Privacy Enhancement Technologies (PETs) สำหรับ IoT/IoT เพื่อความเป็นส่วนตัวของข้อมูล IoT/IoT และ CYBLION Platform”

๑.๒) การฝึกอบรมและใช้งาน “Hands-on Workshop: CYBLION” โดยแบ่งกลุ่มได้ทั้งสิ้น ๑๒ กลุ่ม และมีผู้ช่วยวิทยากร ช่วยฝึกปฏิบัติและแก้ไขปัญหาระหว่างการปฏิบัติการ ประจำกลุ่มกลุ่มละ ๑-๒ คน

๑.๓) การทดสอบใช้งาน “Hands-on Workshop: CYBLION” โดยกำหนด Use Case ที่ใช้ทำการทดลอง และเริ่มต้นทดลองใช้งาน CYBLION บน Mobile APP

๑.๔) รวมถึงระดมความคิดเห็นเกี่ยวกับการติดตั้งและการใช้งานระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลระบบ IoT พร้อมเปิดโอกาสให้ผู้เข้าร่วมทำแบบประเมินความพึงพอใจผู้ใช้งาน CYBLION Platform สรุปและตอบข้อซักถามโดยนักวิจัยและวิทยากร



รูปที่ ๑๑๘ ภาพบรรยากาศการประชุมเชิงปฏิบัติการและระดมความคิดเห็น



รูปที่ ๑๑๙ แสดงเอกสารที่ใช้ประกอบการอบรมเชิงปฏิบัติการ

๕.๒ สรุปผลการประเมินการใช้งานแพลตฟอร์ม

โครงการได้จัดทำแบบสอบถาม เพื่อติดตามและตรวจสอบผลการใช้งานแพลตฟอร์ม CYBLION และ CipherFlow เพื่อขอรับการประเมินจากผู้เข้ารับการอบรม ทั้งนี้มีผู้เข้าร่วมตอบแบบสอบถามจำนวน ๔๔ ท่าน และสามารถสรุปผลการสำรวจได้ดังนี้

ตอนที่ ๑ ความคิดเห็นเกี่ยวกับการใช้งาน CYBLION App (Mobile Application)

๑) ความสะดวกในการติดตั้งโปรแกรม CYBLION App บนมือถือของท่าน มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๔๕.๕๐ % |
| ๑.๒) เห็นด้วยมาก | ๔๗.๗๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๖.๘๐ % |

๒) มีขั้นตอนการใช้งาน CYBLION App ที่เข้าใจง่าย มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๒๐.๕๐ % |
| ๑.๒) เห็นด้วยมาก | ๔๐.๗๙ % |
| ๑.๓) เห็นด้วยปานกลาง | ๓๘.๖๐ % |

๓) มีการออกแบบที่สวยงามน่าใช้ มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๑๓.๖๐ % |
| ๑.๒) เห็นด้วยมาก | ๕๒.๓๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๓๑.๘๐ % |
| ๑.๔) เห็นด้วยน้อย | ๒.๓๐ % |

๔) ความเร็วในการตอบสนองของโปรแกรม มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๖.๙๐ % |
| ๑.๒) เห็นด้วยมาก | ๒๙.๕๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๔๗.๗๐ % |
| ๑.๔) เห็นด้วยน้อย | ๑๕.๙๐ % |

๕) ปัญหาและอุปสรรคในการใช้งาน CYBLION App

ปัญหาที่ผู้เข้ารับการอบรมพบและทำการเสนอแนะ ได้แก่

- ระบบอินเทอร์เน็ตช้ามาก จนทำให้แอปพลิเคชันมีการทำงานที่เกิดปัญหาแอปพลิเคชันค้างออกเองภายหลังการเชื่อมต่ออินเทอร์เน็ตเกิดการติดขัด
- อุปกรณ์ Raspberry Pi ที่นำมาใช้เรียนมีขนาดเล็กประสิทธิภาพต่ำทำให้การเข้าใช้งานอุปกรณ์ Raspberry Pi ได้ช้าและติดขัด เนื่องจากหน่วยความจำของ Raspberry Pi มีขนาดเล็กและได้มีการติดตั้งโปรแกรมอื่นใน Raspberry Pi อื่นๆ นอกเหนือจากโปรแกรม Node-RED สำหรับการใช้งาน CYBLION App
- ผู้เรียนรู้สึกยุ่งยากกับการ Pair Key ระหว่าง CYBLION App และ Edge Computing ที่จำเป็นต้องบังคับให้อุปกรณ์ทั้งสองเชื่อมต่อกับ WiFi เพื่อให้อยู่ในเน็ตเวิร์กเดียวกัน
- การใช้งานร่วมกับ Raspberry Pi ค่อนข้างยาก
- อยากให้มี Username Account เพียงพอต่อผู้เข้าอบรม
- ปุ่มกดภายใน CYBLION App เล็กเกินไป

๖) ข้อเสนอแนะเพื่อปรับปรุงจากผู้ใช้งาน มีดังนี้

- ปรับปรุงสัญญาณอินเทอร์เน็ต
- ต้องการให้ขั้นตอนการใช้งานซับซ้อนน้อยลง
- เพิ่มการใช้งานแบบ Multi-Users
- ปรับปรุงความเร็วของ Server
- เพิ่มฟังก์ชัน
- ปรับปรุงปุ่มที่มีขนาดเล็ก

ตอนที่ ๒ ความคิดเห็นเกี่ยวกับการใช้งาน Node-RED-CYBLION-Edge

๑) ความสะดวกในการใช้งาน มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|---------------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๒๐.๕๐ % |
| ๑.๒) เห็นด้วยมาก | ๕๙.๑๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๑๕.๙๐ % |
| ๑.๔) เห็นด้วยน้อย และน้อยที่สุด | ๔.๕๐ % |

๒) ขั้นตอนการใช้งานเข้าใจง่าย มีผู้ตอบแบบสอบถาม ๔๔ ราย

- ๑.๑) เห็นด้วยมากที่สุด ๒๗.๓๐ %
- ๑.๒) เห็นด้วยมาก ๕๖.๘๐ %
- ๑.๓) เห็นด้วยปานกลาง ๑๕.๙๐ %

๓) มีการออกแบบสวยงามน่าใช้ มีผู้ตอบแบบสอบถาม ๔๔ ราย

- ๑.๑) เห็นด้วยมากที่สุด ๑๘.๒๐ %
- ๑.๒) เห็นด้วยมาก ๕๖.๘๐ %
- ๑.๓) เห็นด้วยปานกลาง ๒๕.๐๐ %

๔) ความเร็วในการตอบสนองของโปรแกรม มีผู้ตอบแบบสอบถาม ๔๔ ราย

- ๑.๑) เห็นด้วยมากที่สุด ๙.๐๐ %
- ๑.๒) เห็นด้วยมาก ๕๐.๑๐ %
- ๑.๓) เห็นด้วยปานกลาง ๓๖.๔๐ %
- ๑.๔) เห็นด้วยน้อย และน้อยที่สุด ๔.๕๐ %

๕) ปัญหาและอุปสรรคในการใช้งาน Node-RED-CYBLION-Edge

ปัญหาที่ผู้ได้รับการอบรมพบและทำการเสนอแนะ ได้แก่

- ยังไม่ค่อยเข้าใจ Flow ในการทำงาน
- Internet ช้า
- อุปกรณ์ประมวลผลช้า
- Raspberry Pi ไม่เสถียร และช้า
- ขั้นตอนยุ่งยาก
- เป็น Library ที่ดี แต่อยากได้คู่มือการใช้งานที่ละเอียดมากขึ้น

๖) ข้อเสนอแนะเพื่อปรับปรุงจากผู้ใช้งาน มีดังนี้

- ควรให้ทดสอบทุกเครื่อง
- Flow การทำงานค่อนข้างเข้าใจได้ยาก
- เปลี่ยน Raspberry Pi
- ใช้คอมพิวเตอร์ผู้อบรมทำเป็น Edge แทน Raspberry Pi
- ถ้าแสดงเป็นวิดีโอการใช้งานจะดีมาก
- เปิดให้ใช้งานเยอะๆ จะได้ขอ Feedback ไปพัฒนาต่อยอด
- ควรใช้อุปกรณ์ที่เสถียรมากกว่านี้
- ไม่แน่ใจว่า สามารถเพิ่ม node ได้เองหรือไม่ หรือเพิ่ม UI สำหรับแสดงผลโดย Node-RED

ตอนที่ ๓ ความคิดเห็นเกี่ยวกับการใช้งาน CYBLION.io

๑) ความสะดวกในการใช้งาน มีผู้ตอบแบบสอบถาม ๔๔ ราย

- | | |
|---------------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๒๕.๕๐ % |
| ๑.๒) เห็นด้วยมาก | ๕๒.๔๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๑๗.๖๐ % |
| ๑.๔) เห็นด้วยน้อย และน้อยที่สุด | ๔.๕๐ % |
- ๒) ขั้นตอนการใช้งานเข้าใจง่าย มีผู้ตอบแบบสอบถาม ๔๔ ราย
- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๒๙.๖๐ % |
| ๑.๒) เห็นด้วยมาก | ๕๖.๕๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๑๓.๙๐ % |
- ๓) มีการออกแบบสวยงามน่าใช้ มีผู้ตอบแบบสอบถาม ๔๔ ราย
- | | |
|------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๑๘.๑๐ % |
| ๑.๒) เห็นด้วยมาก | ๕๗.๙๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๒๔.๐๐ % |
- ๔) ความเร็วในการตอบสนองของโปรแกรม มีผู้ตอบแบบสอบถาม ๔๔ ราย
- | | |
|---------------------------------|---------|
| ๑.๑) เห็นด้วยมากที่สุด | ๑๐.๑๐ % |
| ๑.๒) เห็นด้วยมาก | ๔๗.๗๐ % |
| ๑.๓) เห็นด้วยปานกลาง | ๓๙.๗๐ % |
| ๑.๔) เห็นด้วยน้อย และน้อยที่สุด | ๔.๕๐ % |
- ๕) ปัญหาและอุปสรรคในการใช้งาน Node-RED-CYBLION-Edge
- ปัญหาที่ผู้เข้ารับการอบรมพบและทำการเสนอแนะ ได้แก่
- ยังไม่ค่อยเข้าใจ Flow ในการทำงาน
 - Internet ช้า
 - ต้องการคู่มือการใช้งานแบบวิดีโอ
 - ขั้นตอนยุ่งยากซับซ้อน
- ๖) ข้อเสนอแนะเพื่อปรับปรุงจากผู้ใช้งาน มีดังนี้
- ควรให้ทดสอบทุกเครื่อง
 - Flow การทำงานค่อนข้างเข้าใจได้ยาก
 - ถ้าแสดงเป็นวิดีโอการใช้งานจะดีมาก
 - เปิดให้ใช้งานเยอะๆ จะได้ขอ Feedback ไปพัฒนาต่อยอด
 - ควรใช้อุปกรณ์ที่เสถียรมากกว่านี้

ตอนที่ ๔ ข้อเสนอแนะเกี่ยวกับ Use Case

- ๑) ข้อเสนอแนะเกี่ยวกับ Use Case ที่ต้องการใช้งาน มีดังนี้
- กรณีส่งข้อมูลควบคุมอุปกรณ์ หรือระบบสำคัญ

- ใช้งานรับส่งข้อมูลผ่านโรบอท
- การดึงข้อมูลจาก SCADA ขึ้น Cloud
- รักษาข้อมูล เครื่องจักรที่อยู่บนคลาวด์
- สนใจนำมาใช้งานกับอุปกรณ์ IoT ในโรงงาน
- PM Machine, Productivity
- IoT ที่ต้องการระบบความปลอดภัย
- การส่งข้อมูลเข้าไปคำนวณบนคลาวด์
- IoT ในโรงงาน
- การส่งข้อมูลที่ต้องการความปลอดภัยสูง เช่น ข้อมูลการผลิตของโรงงาน
- งานที่ใช้ Low Compute Logic
- Machine efficiency/ Productivity
- ระบบ IoT ในการตรวจวัดการคำนวณไฟฟ้าจากโซลาเซลล์
- การทำงานของ Robot
- ระบบ IoT ที่ต้องการป้องกันข้อมูลที่เป็นความลับเช่น สูตร ปริมาณ ส่วนผสม
- Sensor

๒) จากข้อที่ผ่านมา กรุณาระบุเหตุผลที่ท่านมีความสนใจ/มีความต้องการนำ CYBLION Platform มาใช้งานกับ Use Case นั้น

- อนาคตต้องมี SI หลายเจ้าเข้ามาดึงข้อมูล อนาคตข้อมูลอาจรั่วไหลได้
- เพิ่มความปลอดภัยของข้อมูล
- Productivity progress
- การดูข้อมูลแบบ Realtime
- อยากให้ขยายผลไปยัง Web App ด้วยนอกเหนือจาก Mobile App
- Power Monitor
- คำนวณค่าไฟของโรงงาน
- คำนวณประสิทธิภาพของเครื่องจักร

การวิจารณ์ผลจากการตอบแบบสอบถาม

ผลจากการพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล (CYBLION) และ Playground (CipherFlow) และการจัดอบรมการใช้งานผ่านการประชุมเชิงปฏิบัติการให้กับผู้เข้ารับการอบรมกลุ่มเป้าหมาย และเมื่อพิจารณาผลการตอบรับการประเมินความพึงพอใจในการใช้งาน ผู้เข้ารับการอบรมส่วนใหญ่มีความเห็นด้วยในการออกแบบการใช้งาน และการติดตั้งอยู่ในเกณฑ์ดี และเห็นถึงประโยชน์ของการใช้งานด้าน IoT และด้านการวิเคราะห์ข้อมูล ในโรงงานอุตสาหกรรม ซึ่งสร้างความตระหนักรู้ถึงความปลอดภัยและความ

เป็นส่วนตัวของข้อมูล ในส่วนข้อเสนอแนะได้มีความเห็นจากผู้เข้ารับการอบรมให้ปรับปรุงเพิ่มเติมเรื่องของการใช้งานที่ยังมีความซับซ้อน และมีคู่มือการใช้งานที่ละเอียดเพิ่มมากขึ้น รวมไปถึงการปรับปรุงให้สามารถทำงานแบบ Multi-User และการแลกเปลี่ยนกุญแจระหว่าง CYBLION App และ Edge Computing มีความซับซ้อน เนื่องจากต้องเชื่อมต่ออยู่ในเครือข่ายวงแลนเดียวกันเท่านั้น อีกทั้ง ปัญหาหลักของการอบรมคือ อุปกรณ์ Raspberry Pi และเครือข่ายอินเทอร์เน็ตที่ใช้ทดสอบมีประสิทธิภาพต่ำ ทั้งนี้ เนื่องจากการอบรมจริง จำนวนผู้ใช้งานเครือข่ายที่ใช้ทำการอบรมมีมากกว่าที่คาดการณ์ไว้ โดยทั้งนี้ผู้เข้าอบรมได้ทำการเชื่อมต่ออุปกรณ์สมาร์ตโฟนและอุปกรณ์คอมพิวเตอร์ส่วนตัวของตนเองร่วมด้วย จึงทำให้ทราบฟิสิกส์ที่ใช้งานจริงมีค่าสูงกว่าระบบเครือข่ายที่ได้ออกแบบไว้ จึงทำให้เกิด Network congestion ที่ระบบ WiFi อีกทั้ง Raspberry Pi โมเดล 3 โดยมี Memory ขนาด ๑ GB ที่ใช้ทดสอบมีขนาดเมมโมรี่น้อยและประกอบกับเน็ตเวิร์คที่มีการแย่งใช้แบนด์วิดท์สูงจึงส่งผลกระทบต่อความเร็วในการตอบสนองของโปรแกรม เนื่องจากการรับคำสั่งต่างๆ ผ่าน CYBLION App จะมีการส่งคำร้องขอคำสั่งไปยัง Mobile Gateway ที่อยู่ในคลาวด์ทำให้เมื่อ Mobile Gateway ได้รับคำสั่งที่ล่าช้า จึงส่งผลกระทบต่อความรู้สึกต่อการตอบสนองกับผู้ใช้งาน ทั้งนี้ทางทีมพัฒนาได้ทำการตรวจสอบ Response Time ของ Mobile Gateway ในขณะที่ทำการใช้งานพร้อมๆ กัน ผลขอความหน่วงเวลาจะเพิ่มขึ้นจาก ๒ วินาที เป็น ๑๖ วินาที และส่งผลต่อการสร้างโปรเจกต์ในคลาวด์ซึ่งการสร้างโปรเจกต์ในคลาวด์ใช้ระบบการเข้าคิวของคำสั่ง โดยระยะเวลาในการตอบสนองของคำสั่งการสร้างโปรเจกต์จาก Mobile Gateway จนถึงการสร้างโปรเจกต์สำเร็จมีระยะเวลาทั้งสิ้น ๒ นาที ดังนั้น โดยรวมของการใช้งานแพลตฟอร์มอยู่ในเกณฑ์ดี แต่ยังคงต้องได้รับการพัฒนาและปรับปรุงให้สามารถรับโหลดได้เพิ่มมากขึ้น และรองรับการใช้งานหลากหลายมากขึ้น

๕.๓ การระดมความคิดเห็น

ส่วนของการสำรวจความตระหนักรู้และระดมความคิดเห็นเกี่ยวกับการติดตั้งและการใช้งานระบบ ในการประชุมเชิงปฏิบัติการ สามารถสรุปได้ดังนี้

๑. แชร์ประสบการณ์ตรงจากบริษัทที่เคยทดลองใช้งานแพลตฟอร์ม CYBLION โดยบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด และ บริษัท เดอะเพ็ท จำกัด ซึ่งบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด ให้ความคิดเห็นเกี่ยวกับการใช้แพลตฟอร์มว่ามีหลักแนวคิดที่สามารถเข้าใจได้ง่าย และมีความต้องการทราบคุณสมบัติพื้นฐานของโทรศัพท์ที่สามารถถอดรหัสได้ โดยต้องการให้ทีมวิจัยพัฒนาต่อไป ในขณะที่ บริษัท เดอะเพ็ท จำกัด ให้ความเห็นว่าโครงการนี้เป็นโครงการที่ดีที่จะช่วยปกป้องข้อมูลของบริษัท และช่วยลดความกังวลที่เกิดขึ้นจากการนำข้อมูลขึ้นคลาวด์ ส่งผลให้ข้อมูลดังกล่าวมีความปลอดภัยมากขึ้น

๒. ช่วงก่อนการอบรมผู้เข้าร่วมอบรมต่างให้ความสนใจและต้องการทราบหลักการและแนวคิดในการใช้งานภาพรวมตัวระบบ ตลอดจนการนำไปคำนวณบนคลาวด์ รวมถึงมีความคาดหวังและความเชื่อมั่นในแอปพลิเคชัน ว่าจะสามารถเสริมสร้างความปลอดภัยของข้อมูลได้

๓. ประเด็นคำถามและข้อเสนอแนะ

- ด้านการใช้งานแอปพลิเคชันบนสมาร์ตโฟน

ผู้เข้าอบรมมีความต้องการสร้างและคำนวณข้อมูลด้วยสมาร์ตโฟนหลายเครื่อง เพื่อประมวลผลบนคลาวด์ต่อไป แต่การคำนวณข้อมูลในหัวข้อนั้นต้องเป็นข้อมูลชนิดเดียวกัน รวมถึงมีความต้องการให้ผู้บริหารหรือ

บุคคลอื่นในองค์กรสามารถดูข้อมูลที่คำนวณได้ และต้องการส่งออกข้อมูลที่รวบรวมมาไปเก็บไว้ที่อื่น ซึ่งพีเจอรี่ในปัจจุบันยังไม่ได้รองรับการดูข้อมูลได้หลายเครื่องหรือส่งออกข้อมูลไปไว้ที่อื่น และเป็นโจทย์ที่น่าสนใจในการพัฒนาเป็น Multi-Key ต่อไป

- ด้านอุปกรณ์ที่รองรับการใช้งาน

สำหรับ CYBLION Platform ปัจจุบันยังคงต้องทำบนแอปพลิเคชันบนสมาร์ทโฟน และมีข้อจำกัดที่ไม่สามารถ Export Key จากสมาร์ทโฟนเครื่องหนึ่งไปอีกเครื่องหนึ่ง หรือไม่สามารถแชร์ Secret Key ข้ามเครื่องได้ ทั้งนี้ หากเปลี่ยนเครื่องใช้งาน จำเป็นต้องสร้างกุญแจใหม่ (Key Generation) ใหม่

- ด้านการใช้ Node-Red และการจัดเตรียมข้อมูล

สำหรับการใช้งานแอปพลิเคชันต้องมีความรู้พื้นฐานในการใช้ Node-Red เนื่องจาก CYBLION platform พัฒนาอยู่บนพื้นฐานของการใช้งาน Node-RED

- ด้านการใช้ประยุกต์ใช้งานกับ Use Case

ผู้เข้าร่วมอบรมมีความสนใจนำ CYBLION platform ไปใช้งานกับฐานข้อมูลที่นอกเหนือจากเซนเซอร์ และมีความสนใจเกี่ยวกับการวัดประสิทธิภาพของกระบวนการทำงาน (Process) โดยใช้การทยอยส่งค่าของข้อมูล ซึ่งตามคุณสมบัติพื้นฐานของแอปพลิเคชันสามารถทำได้ หากใช้ Node-RED และจัดเตรียมข้อมูลให้มีรูปแบบให้สามารถเข้ากับแอปพลิเคชัน

นอกจากนี้สิทธิประโยชน์ของผู้ร่วมอบรมที่เป็นสมาชิก SMC และ IDA Platform ในการใช้งานแอปพลิเคชัน ก็เป็นสิ่งที่ควรนำมาพิจารณาแผนงานการให้บริการในอนาคต เนื่องจากมีผู้ร่วมอบรมที่เป็นสมาชิก IDA มีความสนใจที่จะใช้งานแอปพลิเคชันนี้

๕.๔ การประชาสัมพันธ์ จัดทำสื่อโปสเตอร์ และ สื่อวีดิทัศน์สำหรับเผยแพร่

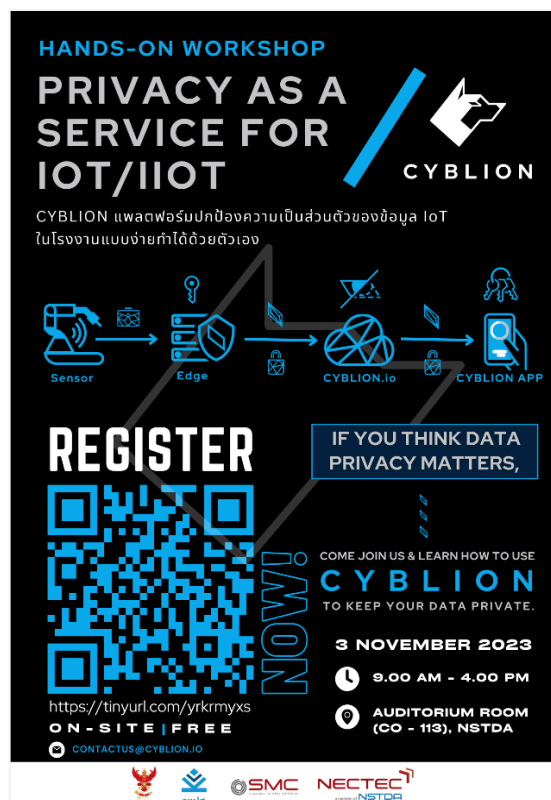
โครงการนี้ได้จัดให้มีการประชาสัมพันธ์ จัดทำสื่อวีดิทัศน์สำหรับเผยแพร่ผลงาน และการให้ความรู้สำหรับการใช้งานแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ดังนี้

๑) การออกบูธแสดงนิทรรศการและให้ความรู้ประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการ AINTEC 2023 เมื่อวันที่ ๑๒ - ๑๔ ธันวาคม ๒๕๖๗ เมืองฮานอย ประเทศเวียดนาม ดังรูปที่ ๑๒๑



รูปที่ ๑๒๐ ประชาสัมพันธ์แพลตฟอร์ม CYBLION ในงานประชุมวิชาการ AINTEC 2023

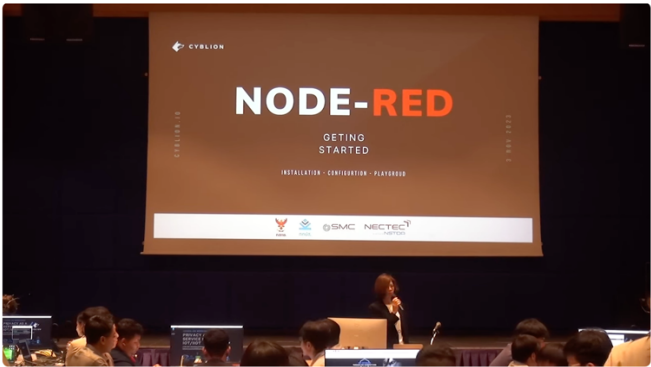
๒) จัดทำโปสเตอร์ประชาสัมพันธ์แนะนำแพลตฟอร์มขึ้นโดยได้นำไปเผยแพร่ผลงาน ตามรูปที่ ๑๒๑

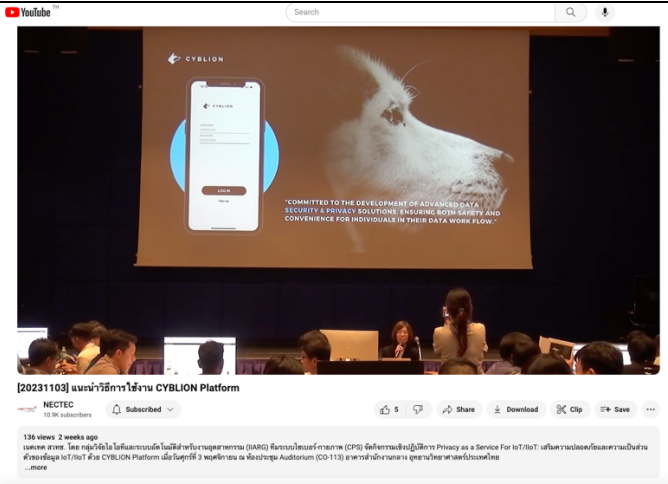
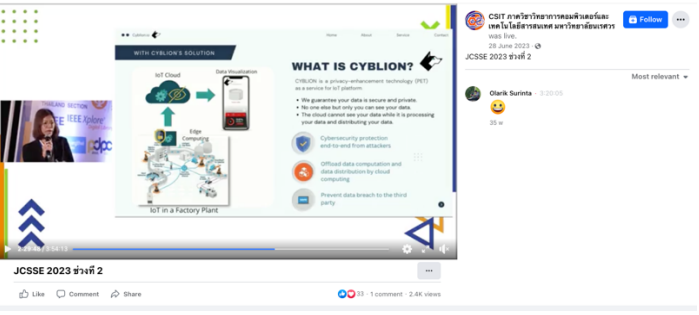


รูปที่ ๑๒๑ โปสเตอร์ประชาสัมพันธ์การประชุมเชิงปฏิบัติการ

๓) การจัดทำสื่อวีดิทัศน์สำหรับเผยแพร่ข้อมูลการอบรม และการให้ความรู้สำหรับการใช้งานแพลตฟอร์ม การให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ดังตารางที่ ๑๗

ตารางที่ ๑๗ สื่อวีดิทัศน์สำหรับเผยแพร่ข้อมูลการอบรม

หัวข้อบรรยาย	ลิงค์เผยแพร่	จำนวนผู้เข้าชม
Privacy-as-a-Service for IoT	https://youtu.be/a19Xd5FPTVQ 	145 ครั้ง
Node-RED และ CipherFlow	https://youtu.be/M0kWEfsh6II 	368 ครั้ง
CYBLION	https://youtu.be/CCbvJuQCfC0	136 ครั้ง

		
PET as a Service for IoT	https://www.facebook.com/csitnu/videos/567955892168564/?_rdc=1&_rdr 	2,400 ครั้ง

๕.๕ ผลงานอื่นๆ

๕.๕.๑ ต้นแบบแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที

ผลงานวิจัยและพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอที ที่สามารถประยุกต์ใช้งานสำหรับ โรงงานอุตสาหกรรมและสถานประกอบการ เพื่อให้หน่วยงานที่สนใจ ทั้งผู้ให้บริการวางระบบ โรงงาน สถานประกอบการสามารถนำไปศึกษาและพัฒนาต่อยอดเพื่อใช้งานหรือให้บริการ โดยผู้ใช้งานทั่วไปสามารถดาวน์โหลดแพลตฟอร์มที่เผยแพร่มาใช้งานได้และไม่เก็บค่าใช้จ่ายในการนำไปประยุกต์ใช้งาน โดยมีข้อมูลเผยแพร่ตามตารางที่ ๑๘

ตารางที่ ๑๘ แอปพลิเคชันที่ได้รับการดำเนินการโครงการ

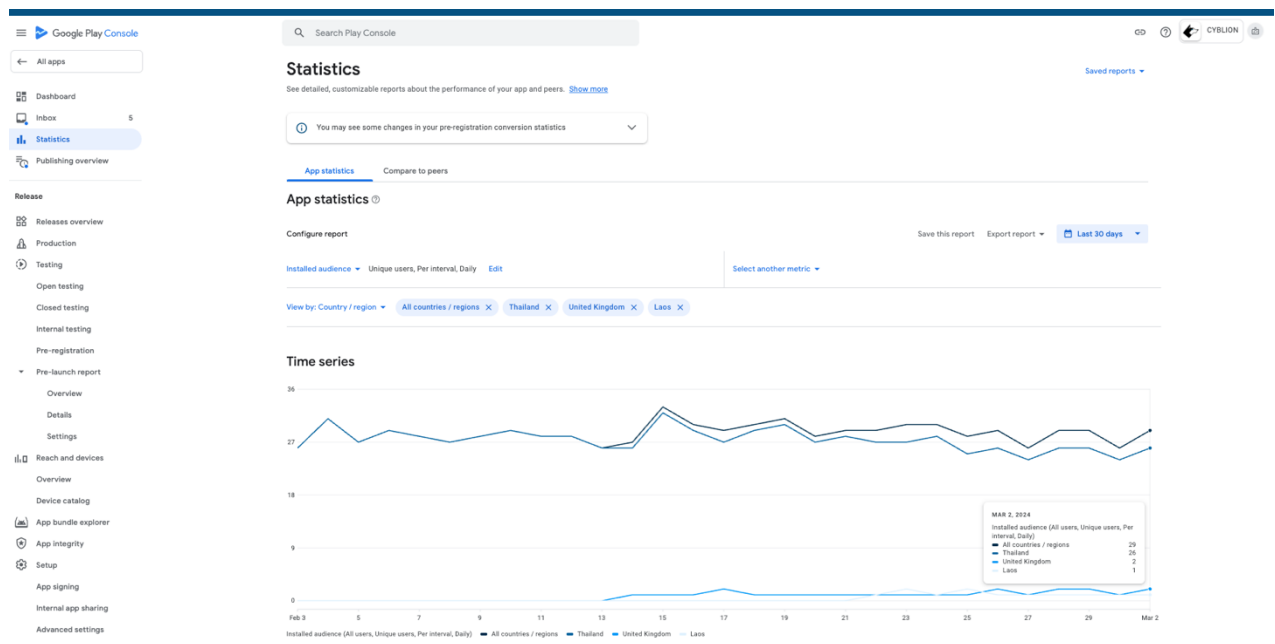
ลำดับ	ผลงาน	GitHub/Publish
๑.	Node-RED-CipherFlow	https://flows.nodered.org/node/node-red-cipherflow
๒.	node-red-cyblion-edge	https://flows.nodered.org/node/node-red-cyblion-edge/scorecard
๓.	CYBLION App (Android)	https://play.google.com/store/apps/details?id=cps.nectec.sealmobile&hl=en-TH

๔.	CYBLION App (iOS)	https://apps.apple.com/th/app/cyblion/id6446020335
๕.	CYBLION.io (Cloud)	https://cyblion.io/

ผลงานต้นแบบได้จดทะเบียนเป็นลงเป็นต้นแบบของทางศูนย์ฯ ภายใต้ชื่อผลงาน , “แพลตฟอร์มไอโอที คลาวด์ที่ให้บริการคำนวณข้อมูลเข้ารหัสโดยไม่ถอดรหัส,” โดยทีมผู้วิจัยและพัฒนา กณิกา สุขสมบูรณ์, เอมอัชนา นิรันตสุขวัฒน์, สุรพันธ์ ทองรังสี, ตะวัน ห่อหุ้ม โดยอยู่ในระดับ TRL 7 ต้นแบบของผลิตภัณฑ์หรือกระบวนการขั้นสุดท้าย ได้ผ่านการสาธิตและพิสูจน์การใช้งานในสภาวะทำงานจริง

๕.๕.๒ สถิติการเข้าเยี่ยมชมและติดตั้งแอปพลิเคชัน CYBLION

รูปที่ ๑๒๒ ถึง รูปที่ ๑๒๓ แสดงสถิติการเข้าเยี่ยมชมและติดตั้งแอปพลิเคชัน CYBLION บน Google Playstore และ Apple Store จากสถิติพบว่า บน Google Playstore มีการดาวน์โหลดโดยประเทศไทย ๒๖ ครั้ง ประเทศอังกฤษ ๒ ครั้ง และ บน Apple Store มีการดาวน์โหลดโดยประเทศในทวีปเอเชีย ๓๘ ครั้ง ประเทศอังกฤษ ๔ ครั้ง และมีผู้เข้าเยี่ยมชมทั้งหมดโดยประเทศในทวีปเอเชีย ๘๓๖ ครั้ง ประเทศอังกฤษ ๗๑ ครั้ง

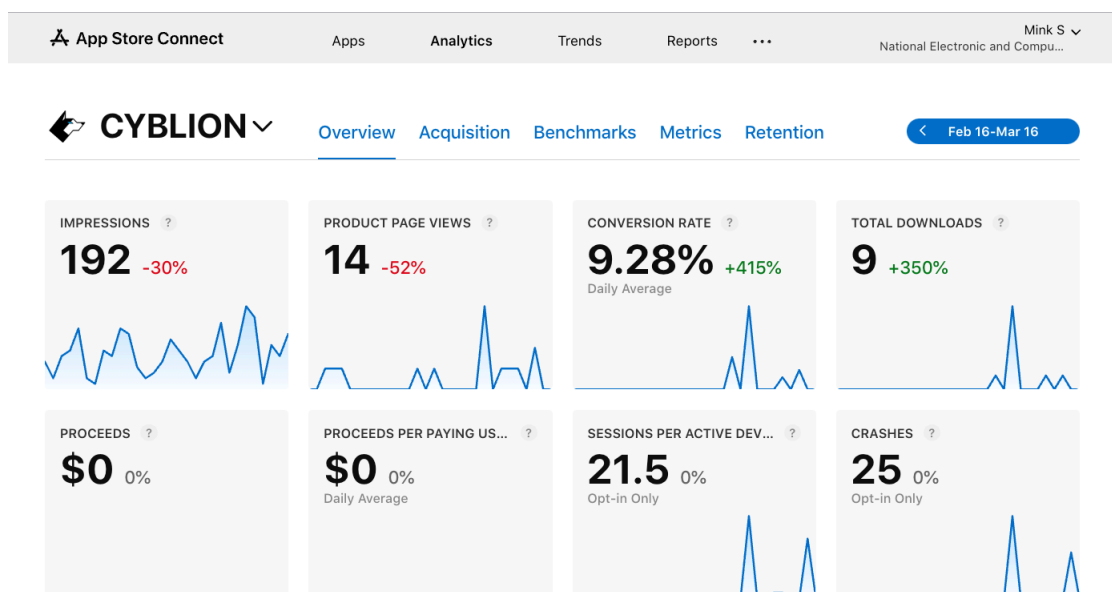


รูปที่ ๑๒๒ สถิติการดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ Android ณ วันที่ ๒ มีนาคม พ.ศ. ๒๕๖๗ ดาวน์โหลดโดยประเทศไทย ๒๖ ครั้ง ประเทศอังกฤษ ๒ ครั้ง และประเทศลาว ๑ ครั้ง

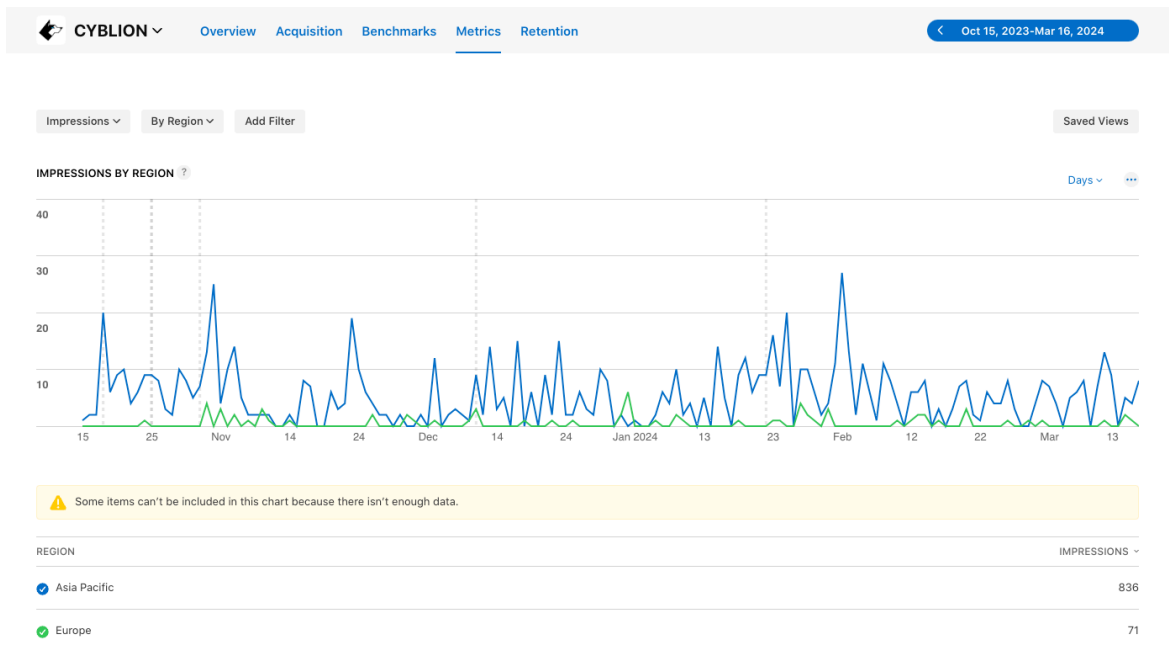
Data table

Days	All countries / regions	Thailand	United Kingdom	Laos
Mar 2, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Mar 1, 2024	26	24	1	1
Percentage of total	100%	92.31%	3.85%	3.85%
Feb 29, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 28, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 27, 2024	26	24	1	1
Percentage of total	100%	92.31%	3.85%	3.85%
Feb 26, 2024	29	26	2	1
Percentage of total	100%	89.66%	6.9%	3.45%
Feb 25, 2024	28	25	1	2
Percentage of total	100%	89.29%	3.57%	7.14%
Feb 24, 2024	30	28	1	1
Percentage of total	100%	93.33%	3.33%	3.33%
Feb 23, 2024	30	27	1	2
Percentage of total	100%	90%	3.33%	6.67%
Feb 22, 2024	29	27	1	1
Percentage of total	100%	93.1%	3.45%	3.45%

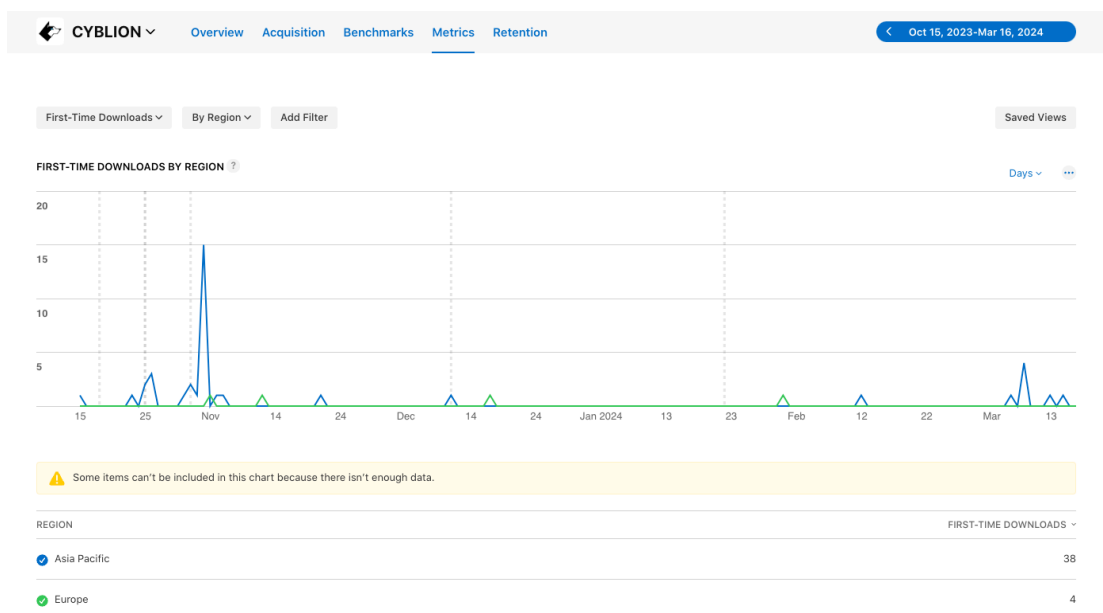
รูปที่ ๑๒๓ สถิติการดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ Android จำแนกตามประเทศ



รูปที่ ๑๒๔ ภาพรวมสถิติการเข้าเยี่ยมชมและดาวน์โหลด CYBLION Application บนระบบปฏิบัติการ iOS



รูปที่ ๑๒๕ สถิติการเข้าเยี่ยมชม CYBLION Application บนระบบปฏิบัติการ iOS



รูปที่ ๑๒๖ สถิติการเข้าดาวน์โหลดและติดตั้ง CYBLION Application บนระบบปฏิบัติการ iOS

๕.๕.๓ สถิติการใช้งาน CYBLION.io

รูปที่ ๑๒๗ แสดงสถิติผู้ใช้งานบน CYBLION.io ณ วันที่ ๑๙ มีนาคม ๒๕๖๗ โดยระยะเวลาที่ใช้งานในระบบสูงสุด ๒๗๑ วัน และมีผู้ใช้งานใหม่ระยะเวลา ๓ วัน ซึ่งผู้ใช้งานในระบบเป็นแบบจำกัดจำนวนเนื่องจากขนาดทรัพยากรของคลาวด์ที่ให้บริการมีจำนวนจำกัด

```

root@cipherflow:/home/kalika# kubectl get pods --all-namespaces |grep "cyblion"
cyblion      database-gateway-684cbb6b87-ztm89      1/1      Running      3 (7d7h ago)      136d
cyblion      database-gateway-684cbb6b87-kn6mb      1/1      Running      3 (7d7h ago)      136d
cyblion      groupstore-redis-849958d995-4sj4m      1/1      Running      14 (7d7h ago)      378d
cyblion      nodered-rabbitmq-0                      1/1      Running      14 (7d7h ago)      378d
cyblion      cyblion-session-redis-0                1/1      Running      14 (7d7h ago)      378d
cyblion      edgebroker-redis-5588c78444-16x5n      1/1      Running      14 (7d7h ago)      378d
cyblion      database-gateway-684cbb6b87-xskcx      1/1      Running      3 (7d7h ago)      136d
cyblion      mysql-59fb556cb8-t6hcl                1/1      Running      14 (7d7h ago)      378d
cyblion      groupstore-6bff5bbd7b-18g52            1/1      Running      14 (7d7h ago)      378d
cyblion      nodered-controller-8b8666d6b-q24nv     1/1      Running      5 (7d7h ago)      170d
cyblion      chatchamon-p743062619213-nodered-6fc56d4f5f-tb5hn 1/1      Running      16 (7d7h ago)      167d
cyblion      eakarar-p477894720506-nodered-7f5cb54cf7-twrk    1/1      Running      2 (7d7h ago)      53d
cyblion      cybliontvop-p945763475236-nodered-5566f8966c-klz9g 1/1      Running      28 (7d7h ago)      217d
cyblion      plume142-p946106028223-nodered-656cf8cbbb-bcssm  1/1      Running      14 (7d7h ago)      140d
cyblion      cyblionpentester2-p710866874810-nodered-f6bd88d99-k8rr2 1/1      Running      10 (7d7h ago)      139d
cyblion      eakarar-p151748976058-nodered-7c6457b775-45fwr   1/1      Running      10 (7d7h ago)      136d
cyblion      web-server-849fd4b876-914hp            1/1      Running      21 (7d7h ago)      198d
cyblion      cyblionwsk1-p539072358151-nodered-57f776f5fd-77b2k 1/1      Running      11 (7d7h ago)      136d
cyblion      plume142-p568630601653-nodered-58d4597dbd-d8hbj   1/1      Running      15 (7d7h ago)      140d
cyblion      cybliontester4-p948727399714-nodered-798cc7845d-v12qs 1/1      Running      13 (7d7h ago)      145d
cyblion      cyblion-p102818583568-nodered-5f75677f49-8sl9d    1/1      Running      30 (7d7h ago)      228d
cyblion      cybliontester4-p986965858910-nodered-7fd99fcb57-nmj5 1/1      Running      27 (7d7h ago)      228d
cyblion      cyblionpentester-p050289124243-nodered-7c89f7b87c-2q6hj 1/1      Running      13 (7d7h ago)      145d
cyblion      edgebroker-0                            3/3      Running      27 (7d7h ago)      223d
cyblion      cybliontester-p210508897403-nodered-5c58f969bc-gb6w9 1/1      Running      10 (7d7h ago)      136d
cyblion      cyblion-p238808386764-nodered-5f58f9f7bd-nd462    1/1      Running      2 (7d7h ago)      97d
cyblion      cybliontester4-p587675092544-nodered-868584c666-npgp7 1/1      Running      15 (7d7h ago)      144d
cyblion      cyblionwsk1-p327949429068-nodered-5c584c7fb8-kztxn 1/1      Running      10 (7d7h ago)      136d
cyblion      cpssecteam-p968779617716-nodered-65d89dc7f8-1h7jt 1/1      Running      31 (7d7h ago)      228d
cyblion      cyblionmaster-p867946906445-nodered-5576f89767-qcpbt 1/1      Running      27 (7d7h ago)      227d
cyblion      iot-rabbitmq-0                          1/1      Running      14 (7d7h ago)      378d
cyblion      mobile-gateway-6c99b6b56d-xmw86        1/1      Running      0              6d12h
cyblion      mobile-gateway-6c99b6b56d-tdgr4        1/1      Running      0              6d12h
cyblion      mobile-gateway-6c99b6b56d-2nqgh        1/1      Running      0              6d12h
cyblion      i-p264514939583-nodered-bf59c889c-8szl2 1/1      Running      0              5d18h
cyblion      cyblionmaster-p194726227525-nodered-7cdd99b85-mcxhp 1/1      Running      0              3d7h
cyblion      cyblion-wsk4-p644955397500-nodered-77bcf5c88d-g9pb5 1/1      Running      0              3d7h
cyblion      cyblionwsk6-p323379371239-nodered-5cf8ff67c8-lr7ht 0/1      Pending      0              3d7h

```

รูปที่ ๑๒๓ สถิติ User ที่ Active อยู่บน CYBLION.io ณ วันที่ ๑๙ มีนาคม ๒๕๖๗

๕.๕.๔ การติดตั้งและอบรมการใช้งานระบบบริการคุ้มครองความเป็นส่วนตัวของข้อมูล IoT

จากการศึกษาความต้องการใช้งานระบบ IoT ภายในโรงงานอุตสาหกรรมเป้าหมายและกรณีศึกษาของโรงงานอุตสาหกรรมที่ต้องการเข้าร่วมทดสอบตามหลักการที่ใช้คัดเลือกสำหรับการติดตั้งทดสอบในโรงงานนำร่องตามที่กำหนด โดยคำนึงถึงปัจจัยความพร้อมของโรงงานคู่ทดสอบ ทำให้เลือกโรงงานอุตสาหกรรมต้นแบบ เพื่อทำการวางระบบการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูล อบรมการติดตั้งใช้งาน การตรวจสอบผลการติดตั้งการใช้งานระบบ การดูแลรักษา รวมถึงปรึกษาหารือสร้างความเข้าใจในการนำเข้าข้อมูล การเข้ารหัส และการรับข้อมูลเพื่อการเรียกใช้ข้อมูลผ่านระบบที่ติดตั้งไว้ จำนวน ๒ โรงงาน ได้แก่

๑) โรงงาน Thanakorn Vegetable Oil Products Co., Ltd.



รูปที่ ๑๒๔ การทดสอบระบบและความเร็วของสัญญาณ 5G ที่ใช้ในการสื่อสาร



รูปที่ ๑๒๙ การติดตั้งและตรวจสอบข้อมูล Node-RED บน Edge Computing

๒) โรงงาน บริษัท เดอะ เพ็ท จำกัด



รูปที่ ๑๓๐ แสดงพื้นที่โกดังน้ำมันพืช โกดังกากพืชน้ำมัน ฟาร์มไซโลน้ำมัน และส่วนที่ต้องการติดตั้งกล้อง CCTV ในบริเวณโรงงานบริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด (TVOP)

๕.๕.๕ เอกสารประกอบการบรรยายการจัดอบรม

ทีมงานได้จัดทำเอกสารประกอบการบรรยายการจัดอบรมเชิงปฏิบัติการโดยส่งมอบเอกสารให้แก่ผู้เข้ารับการอบรม ดังนี้

๑. เอกสารประกอบการอบรม Privacy as a Service for IIoT/IoT
๒. เอกสารประกอบการอบรม Node-RED (Node-RED Get Started)
๓. เอกสารประกอบการอบรม CYBLION Platform Training & Tutorial
๔. คู่มือการใช้งานแพลตฟอร์ม CYBLION (USER Guide Manual VOL.0.1)

๕.๕.๖ ผลงานตีพิมพ์ในงานประชุมวิชาการ

ทีมงานได้นำเสนอผลการดำเนินโครงการและมีผลงานที่ตีพิมพ์ในงานประชุมวิชาการ จำนวน ๔ เรื่อง โดยชื่อผลงานที่ตีพิมพ์ตามตารางที่ ๑๙ และรายละเอียดผลงานที่ตีพิมพ์ ตามภาคผนวก ค.

ตารางที่ ๑๙ ผลงานตีพิมพ์ในงานประชุมวิชาการ

ลำดับ	บทความ	ประเภท
๑.	T. Hohum, K. Suksomboon, A. Niruntasukrat, S. Mongkolluksamee, N. Tansangworn, “ScottishFold: CatBoost-Enabled Lightweight Autonomous Smart Home Device Classification,” Proceedings of IEEE Globecom Workshops (GC Wkshps), December 2021.	งานประชุมวิชาการนานาชาติ (Full Paper) เอกสารตอบรับอยู่ในภาคผนวก จ.
๒.	N. Tanabodee, K. Suksomboon, C. Issariyapat, S. Mongkolluksamee, A. Niruntasukrat, N. Tansangworn, S. Kitisin, “CipherFlow: A Playground for Developing Privacy-Preserving IoT in Node-RED” Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '22), December 2022.	งานประชุมวิชาการนานาชาติ (Full Paper) เอกสารตอบรับอยู่ในภาคผนวก จ.
๓.	K. Suksomboon, N. Tanabodee, E. Saktawornlerd, A. Niruntasukrat, C. Issariyapat, S. Mongkolluksamee, N. Tansangworn, S. Kitisin, “Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0,” Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '23), December 2023	งานประชุมวิชาการนานาชาติ (Poster) เอกสารตอบรับอยู่ในภาคผนวก จ.
๔.	C. Phoojaroenchanachai, K. Kaemarungsi and P. Pongpaibool, “5G Use Cases for Thailand's Smart Factory/Manufacturing,” Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0,” Proceedings of the 17th Asian Internet Engineering Conference (AINTEC '23), December 2023	งานประชุมวิชาการนานาชาติ (Poster) เอกสารตอบรับอยู่ในภาคผนวก จ.

๕.๕.๔ บทความทางวิชาการระดับชาติ (ฉบับยื่นเสนอบทความ)

ทีมวิจัยได้จัดเขียนบทความวิชาการระดับนานาชาติ เรื่อง “แพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลสำหรับระบบ IoT,” NBTC Journal 2024.

ตารางที่ ๒๐ บทความทางวิชาการระดับชาติ

ลำดับ	บทความ	ประเภท
๑.	Kalika Suksomboon, Aimaschana Nirantasukrat, Nataset Tanabodee, Eakarat Saktawornlerd, Chaveee Issariyapat, Tawan Hohum, Sophon Mongkolluksamee, Natapon Tansangworn,	วารสารวิชาการระดับชาติ (Journal)

	Sukumal Kitisin, “Privacy as a Service for IoT Platform,” NBTC Journal 2024	เอกสารตอบรับการ ส่งในภาคผนวก จ.
--	---	------------------------------------

๕.๕.๕) บทความทางวิชาการระดับนานาชาติ (ฉบับยื่นเสนอบทความ)

ทีมวิจัยได้จัดเขียนบทความวิชาการระดับนานาชาติ เรื่อง “CipherFlow: A Homomorphic Operation Toolkit for Privacy-Preserving Industrial IoT Application,” IEEE Internet of Things Journal โดยมีบทความแสดง ตามภาคผนวก ง.

ตารางที่ ๒๑ บทความทางวิชาการระดับนานาชาติ

ลำดับ	บทความ	ประเภท
๑.	K. Suksomboon, N. Tanabodee, A. Niruntasukrat, S. Kitisin, J. Kurihara, T. Tanaka, S. Mongkolluksamee, N. Tansangworn, C. Issariyapat, “CipherFlow: A Homomorphic Operation Toolkit for Privacy-Preserving Industrial IoT Application,” IEEE Internet of Things Journal	วารสารวิชาการ นานาชาติ (Journal) อยู่ในระหว่าง เตรียมการจัดส่ง

บทที่ ๖

สรุปผลการวิจัย และข้อเสนอแนะ

๖.๑ สรุปผลการวิจัย

การดำเนินโครงการวิจัยและพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความปลอดภัยของไอโอที (Privacy-as-a-Service for IoT) ได้ผลผลิตงานวิจัยเชิงวิชาการที่มีผลงานตีพิมพ์ในงานประชุมวิชาการนานาชาติในรูปแบบของ Paper ๒ ฉบับและ Poster ๒ ฉบับและต้นแบบผลงานที่สามารถนำไปใช้งานทดสอบภาคสนามในโรงงานอุตสาหกรรม โดยผลงานทางวิชาการนั้นเป็นผลการศึกษาการเทคโนโลยีการเข้ารหัสแบบโฮโมมอร์ฟิกส์ที่เหมาะสมกับการนำไปพัฒนาเพื่อตอบโจทย์การใช้งานกับแอปพลิเคชันไอโอที ที่ช่วยให้ข้อมูลที่ถูกรหัสสามารถนำไปคำนวณต่อได้โดยไม่ต้องถอดรหัส และผลลัพธ์ของการคำนวณนั้นสามารถถอดรหัสมาได้ค่าผลลัพธ์ที่มีความถูกต้องสูง ทั้งนี้ระดับความถูกต้องขึ้นกับขนาดพารามิเตอร์โฮโมมอร์ฟิกส์ที่เลือกใช้ ในขณะที่การเข้ารหัสแบบโฮโมมอร์ฟิกส์มีหลายเทคนิคด้วยกัน แต่เนื่องจากความต้องการใช้งานข้อมูลไอโอทีของอุตสาหกรรมมีความหลากหลายและข้อมูลที่ต้องการประมวลผลเป็นข้อมูลเลขจำนวนจริง โดยผลการศึกษาภายใต้โครงการนี้พบว่าเทคนิคการเข้ารหัสโฮโมมอร์ฟิกส์แบบ RNS-CKKS นั้นเป็นเทคนิคที่สามารถรองรับการเข้ารหัสบนข้อมูลจำนวนจริง ซึ่งแตกต่างจากวิธีการเข้ารหัสแบบอื่นๆ ในอดีต ดังนั้นโครงการนี้จึงพัฒนาแพลตฟอร์มขึ้นบนพื้นฐานของเทคนิคการเข้ารหัสโฮโมมอร์ฟิกส์แบบ RNS-CKKS แต่อย่างไรก็ตามการเข้ารหัสด้วยเทคนิคนี้จำเป็นต้องเข้ารหัสแบบชุดข้อมูล ซึ่งทำให้ไม่เหมาะสมกับลักษณะของข้อมูลไอโอทีที่ถูกอัปเดตตามเวลาและจะถูกส่งมาครั้งละ ๑ ค่าต่อเวลา ดังนั้นเพื่อให้เกิดความคุ้มค่าของการบรรจุข้อมูลลงในข้อมูลเข้ารหัสแบบชุดข้อมูล แต่ยังคงไม่ทำให้สูญเสียการอัปเดตข้อมูลแบบเวลาจริง ทำให้โครงการนี้ได้ศึกษาและนำเสนอวิธีการจัดเรียงข้อมูลไอโอทีที่ใหม่ที่เหมาะสมกับการแสดงผลข้อมูลไอโอทีแบบเรียลไทม์ ผลงานวิจัยในโครงการนี้ได้พัฒนาต่อยอดจากเทคนิคและฟังก์ชันที่มีอยู่เดิมใน Microsoft SEAL เพื่อให้สามารถรองรับการทำโอเพอร์เรชันการหารได้ด้วยเทคนิคการคูณแบบวนซ้ำโดยผลผลิตของเทคนิคใหม่ที่ถูกคิดค้นนี้ได้ถูกตีพิมพ์ในงานประชุมวิชาการ AINTEC 2023 และอยู่ในระหว่างการยื่นจดสิทธิบัตร จากนั้นทางในโครงการได้พัฒนาอัลกอริทึมการหารด้วยเทคนิคการคูณแบบวนซ้ำและนำเอาอัลกอริทึมที่เสนอพร้อมเครื่องมือโอเพอร์เรชันการคำนวณที่มีอยู่เดิมใน Microsoft SEAL มาพัฒนาเป็นชุดเครื่องมือซอฟต์แวร์ Node-RED-Cipherflow ที่สามารถให้ผู้ใช้ทั่วไปดาวน์โหลดมาใช้งานฟรีได้บนโปรแกรม Node-RED นอกจากนี้ ทางโครงการได้พัฒนาต่อยอดจากชุดเครื่องมือดังกล่าวเป็นแพลตฟอร์ม CYBLION ที่ประกอบด้วย ๑) Node-RED-CYBLION-Edge, ๒) CYBLION.io และ ๓) CYBLION App ที่รองรับทั้งการใช้งานบนสมาร์ตโฟนระบบปฏิบัติการ iOS และ Android โดยคลาวด์ที่ให้บริการสามารถรองรับการใช้งานจากผู้ใช้งานพร้อมกันหลายคนได้ จากนั้นโครงการได้จัดให้มีการทดสอบแพลตฟอร์มที่พัฒนาขึ้นมาทั้งการทดสอบในห้องปฏิบัติการและการทดสอบภาคสนามในกรณีศึกษาของโรงงานอุตสาหกรรมสองแห่ง คือ ๑. บริษัท ธนากรผลิตภัณฑ์น้ำมันพืช จำกัด และ ๒. บริษัท เดอะ เฟ้ท จำกัด โดยผลผลิตจากการพัฒนาแพลตฟอร์มและผลการทดสอบดังกล่าว โครงการจึงได้จัดทะเบียนผลงานต้นแบบภาคสนาม จากนั้นผลการทดสอบและเผยแพร่การใช้งานแพลตฟอร์มไปยังกลุ่มผู้ใช้งานเป้าหมายจำนวน 60 คน ผ่านการจัดอบรมเชิงปฏิบัติการและการระดมความคิดเห็น

โดยผู้เข้ารับการอบรมได้ให้คำแนะนำติชมและเสนอแนะการใช้งานแพลตฟอร์มที่พัฒนาขึ้น ซึ่งมีประโยชน์ต่อการนำไปปรับปรุงและพัฒนาแพลตฟอร์มเพื่อการใช้งานต่อไป ภายหลังการจัดอบรมและเผยแพร่องค์ความรู้ในเชิงวิชาการและการใช้งานแพลตฟอร์มก่อให้เกิดการเรียนรู้และตระหนักรู้ถึงความสำคัญของความปลอดภัยและความเป็นส่วนตัวของข้อมูลไอโอทีในโรงงานอุตสาหกรรมแก่กลุ่มเป้าหมาย

๖.๒ ข้อเสนอแนะ

การพัฒนาแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีเป็นสิ่งท้าทายสำหรับนักพัฒนาเทคโนโลยีเข้ารหัสแบบโฮโมมอร์ฟิกส์ที่จะนำสิ่งที่ตนเองพัฒนานั้นมาประยุกต์ใช้งานกับแอปพลิเคชันอุตสาหกรรมไอโอที เนื่องจากการเข้ารหัสแบบโฮโมมอร์ฟิกส์มีความซับซ้อนและใช้ทรัพยากรการคำนวณสูงกว่าการส่งข้อมูลไอโอทีทั่วไปอย่างมาก การเติมเต็มช่องว่างระหว่างนักวิจัยด้านวิทยาการเข้ารหัสและนักพัฒนาไอโอทีโซลูชันยังคงเป็นสิ่งที่ต้องการ ดังนั้นในระยะเวลาที่ผ่านมาจึงไม่มีนักวิจัยและนักพัฒนาแอปพลิเคชันให้ความสนใจนำเทคโนโลยีการเข้ารหัสนี้ไปพัฒนาต่อยอดเป็นการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีในตลาดโลก ถึงแม้ว่าโปรแกรม Node-RED จะเป็นโปรแกรมที่ถูกพัฒนาขึ้นมาเพื่อลดช่องว่างระหว่างนักพัฒนา IoT Solution และนักเขียนโปรแกรม เพื่อทำให้การพัฒนา IoT Solution ทำได้ง่ายและรวดเร็วมากยิ่งขึ้น ดังนั้นโครงการนี้จึงได้เลือกพัฒนาเทคโนโลยีเข้ารหัสแบบโฮโมมอร์ฟิกส์ที่อาศัยโปรแกรม Node-RED เป็นพื้นฐานเพื่อให้นักพัฒนา IoT Solution ที่มีความคุ้นเคยกับการใช้งาน Node-RED เรียนรู้และประยุกต์ใช้งานแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีที่พัฒนาขึ้นมาได้เอง

อย่างไรก็ตาม การใช้งานระบบไอโอทีของอุตสาหกรรมไทยยังอยู่ในช่วงเริ่มต้นและเปลี่ยนผ่านจากอุตสาหกรรมดั้งเดิมไปสู่อุตสาหกรรมอัจฉริยะ จึงทำให้วิศวกรและนักพัฒนา IoT Solution ของไทยเองจำนวนมากที่ไม่มีความรู้การใช้ Node-RED ในการพัฒนา IoT Solution เพื่อการประยุกต์ใช้งาน Node-RED สำหรับไอโอทีที่ต้องการของโรงงาน ข้อเสนอแนะของการขับเคลื่อนให้มีการใช้งานแพลตฟอร์มการให้บริการคุ้มครองความเป็นส่วนตัวของข้อมูลไอโอทีที่ถูกพัฒนาขึ้นภายใต้โครงการนี้ จำเป็นต้องทำไปพร้อมๆ กับการให้ความรู้ด้านการใช้งาน Node-RED ในการพัฒนา IoT Solution การสร้าง Use Case ที่หลากหลายและสอดคล้องกับความต้องการของโรงงานอุตสาหกรรม อีกทั้งการปรับปรุงหรือพัฒนาให้แพลตฟอร์มรองรับการใช้งานแบบ Multi-User ได้นั้นอาจจะต้องแลกมากับการเปลี่ยนแปลงชุด Library ของเทคโนโลยีเข้ารหัสแบบโฮโมมอร์ฟิกส์เดิมที่นำมาใช้ในโครงการนี้ และการพัฒนาเพื่อให้ตอบโจทย์ความสะดวกสบายของการใช้งานสำหรับ Web Application อาจจะต้องแลกมากับการสูญเสียความปลอดภัย และความเป็นส่วนตัวของข้อมูล เนื่องจากถ้าหากข้อมูลต้องถูกแสดงผลผ่าน Web Application แบบดั้งเดิม หรือการแสดงผลผ่าน Grafana Cloud จะทำให้ขั้นตอนการถอดรหัสข้อมูลที่ผ่านการคำนวณแล้วต้องอยู่บน Server ที่ให้บริการ Web Application ดังนั้นการเพิ่มเติม Feature ใหม่ๆ เพื่อตอบสนองความต้องการใช้งานตามที่คุณใช้งานต้องการจำเป็นต้องอาศัยเทคโนโลยีใหม่ เช่น Web3 หรืออาจต้องออกแบบสถาปัตยกรรม Web Application ใหม่ที่ยังคงความเป็นส่วนตัวของข้อมูลได้ ควบคู่กับการให้ความรู้ความเข้าใจในเทคโนโลยีการเข้ารหัสและความปลอดภัยทางไซเบอร์แก่ผู้ใช้งานเพิ่มเติม เพื่อลดข้อ

เรียกร้องที่เป็นไปได้ต่างๆ และเพื่อให้เล็งเห็นถึงคุณประโยชน์ในการออกแบบสถาปัตยกรรมแพลตฟอร์มของโครงการมากยิ่งขึ้น

บรรณานุกรม

- [1] R. LU, A. HABIBI, A. A. GHORBANI, “A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT”, IEEE Access (SPECIAL SECTION ON SECURITY AND PRIVACY IN APPLICATIONS AND SERVICES FOR FUTURE INTERNET OF THINGS), March 28, 2017.
- [2] S. Makwana, “An Application of Homomorphic Encryption on IoT based Green House”, in Proceeding of ICECDS-2017, 01 August 2017.
- [3] I. Yoon, N. Cao, A. Amaravati, A. Raychowdhury “A 55nm 50nJ/encode 13nJ/decode Homomorphic Encryption Crypto-Engine for IoT Nodes to Enable Secure Computation on Encrypted Data”, in Proceeding of IEEE Custom Integrated Circuits Conference (CICC), 14-17 April 2019.
- [4] A. B. KORCHI and N. E. MRABET, “A practical use case of homomorphic encryption”, in Proceeding of International Conference on Cyberworlds (CW), 2-4 Oct. 2019.
- [5] Y. Doroz, E. Ozturk, and B. Sunar “Accelerating Fully Homomorphic Encryption in Hardware”, IEEE TRANSACTIONS ON COMPUTERS, VOL. 64, NO. 6, JUNE 2015.
- [6] M. A. Mukhtar, M. K. Bhatti and G. Gogniat “Architectures for Security: A comparative analysis of hardware security features in Intel SGX and ARM TrustZone,” in Proceeding of C-CODE.
- [7] S. Halevi and V. Shoup “Faster Homomorphic Linear Transformations in HElib”, [online] url: <https://eprint.iacr.org/2018/244.pdf>
- [8] T. Yiannis, Y. Moti (2006). "On the security of ElGamal based encryption". Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. 1431: 117–134. doi:10.1007/BFb0054019. ISBN 978-3-540-64693-8.
- [9] TRUSTONIC: [online] url: <https://www.trustonic.com/microchip/iot-developer-kit/#download>
- [10] R. Tahir, et. al, “Using SGX-Based Virtual Clones for IoT Security”, IEEE NCA 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA.
- [11] K. Sako (2005) GOLDWASSER-MICALI ENCRYPTION SCHEME. In: van Tilborg H.C.A. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA.

- [12] L. Fousse, P. Lafourcade., M. Alnuaimi (2011) Benaloh's Dense Probabilistic Encryption Revisited. In: Nitaj A., Pointcheval D. (eds) Progress in Cryptology – AFRICACRYPT 2011. AFRICACRYPT 2011. Lecture Notes in Computer Science, vol 6737. Springer, Berlin, Heidelberg.
- [13] A. B. Korchi, Nadia E. Mrabet, "A practical use case of homomorphic encryption," in Proc. International Conference on Cyberworlds (CW), 2019.
- [14] A. Patra and D. Ravi, "On the Power of Hybrid Networks in Multi-Party Computation," IEEE Trans. On Information Theory, vol. 64, no. 6, June 2018.
- [15] A. Khedr, G. Gulak, and V. Vaikuntanathan, "SHIELD: Scalable homomorphic implementation of encrypted data-classifiers," IEEE Trans. Comput., vol. 65, no. 9, pp. 2848-2858, Sep. 2016.
- [16] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," IEEE Trans. Multimedia, vol. 14, no. 3, pp. 703–716, Jun. 2012.
- [17] C. Bonte and F. Vercauteren, "Privacy-preserving logistic regression training," BMC Med. Genomics, vol. 11, no. 4, 2018, Art. no. 86.
- [18] C. Wang, K. Ren, J. Wang, and K. M. R. Urs, "Harnessing the cloud for securely solving large-scale systems of linear equations," in Proc. IEEE 31st Int. Conf. Distrib. Comput. Syst., Jun. 2011, pp. 549-558.
- [19] D. Benjamin and M. J. Atallah, "Private and cheat" in Proc. IEEE 6th Annu. Conf. Privacy, Secur. Trust, Oct. 2008, pp. 240—245.
- [20] D. Chaum, I. Damgård, and J. van de Graaf, "Multiparty computations ensuring privacy of each party's input and correctness of the result," in Proc. Conf. Theory Appl. Cryptogr. Techn., Santa Barbara, CA, USA, Aug. 1987, pp. 87–119.
- [21] D. Liu, E. Bertino, and X. Yi, "Privacy of outsourced k-means clustering," in Proc. 9th ACM Symp. Inf., Comput. Commun. Secur., 2014, pp. 123–134.
- [22] D. Li, X. Dong, Z. Cao, and H. Wang, "Privacy-preserving outsourced image feature extraction," J. Inf. Secur. Appl., vol. 47, pp. 59-64, Aug. 2019.
- [23] E. Kiltz, P. Mohassel, E. Weinreb, and M. Franklin, "Secure linear algebra using linearly recurrent sequences," in Proc. Theory Cryptogr. Conf. Berlin, Germany: Springer, 2007, pp. 291-310.

- [24] F. Liu, W. K. Ng, and W. Zhang, "Encrypted association rule mining for outsourced data mining," in Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl., Mar. 2015, pp. 550-557.
- [25] F. Tang, W. Wu, J. Liu, H. Wang, and M. Xian, "Privacy-preserving distributed deep learning via homomorphic re-encryption," *Electronics*, vol. 8, no. 4, p. 411, 2019.
- [26] "FV-NFLlib", [Online access] Available: <https://github.com/CryptoExperts/FV-NFLlib>
- [27] "HEAT", [Online access] Available: <https://github.com/bristolcrypto/HEAT>
- [28] "HElib," [Online access] Available: <https://github.com/homenc/HElib>
- [29] I. K. Dutta, B. Ghosh, A. H. Carlson, and M. Bayoumi, "Lightweight Polymorphic Encryption for the Data Associated with Constrained Internet of Things Devices", IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020.
- [30] I. Anati, S. Gueron, P. Johnson, and V. R. Scarlata, "Innovative Technology for CPU based attestation and Sealing," in Proc. ACM 2nd Int. Workshop Hardw. Archit. Support Secur. Privacy, New York, USA, vol. 13, 2013.
- [31] I. Kalyan Dutta, B. Ghosh, A. H. Carlson and M. Bayoumi, "Lightweight Polymorphic Encryption for the Data Associated with Constrained Internet of Things Devices," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1-6, doi: 10.1109/WF-IoT48130.2020.9221296.
- [32] I.T. Lien, Y. H. Lin, J. R. Shieh, and J. L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for K-NN search," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 6, pp. 863-873, Jun. 2013.
- [33] J. H. Cheon, D. Kim, Y. Kim, and Y. Song, "Ensemble method for privacy-preserving logistic regression based on homomorphic encryption," *IEEE Access*, vol. 6, pp. 46938-46948, 2018.
- [34] J. Wang, Y. Han, X. Yang, T. Zhou, and J. Chen, "A new group location privacy-preserving method based on distributed architecture in LBS," *Secur. Commun. Netw.*, vol. 2019, Feb. 2019, Art. no. 2414687.
- [35] J. H. Ziegeldorf, J. Pennekamp, D. Hellmanns, F. Schwinger, I. Kunze, M. Henze, J. Hiller, R. Matzutt, and K. Wehrle, "BLOOM: Bloom filter based oblivious outsourced matchings," *BMC Med. Genomics*, vol. 10, no. 2, 2017, Art. no. 44.
- [35] K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can Homomorphic Encryption be Practical?", *ACM CCSW*, 2011.

- [36] L. Zhang, J. Li, S. Yang, and B. Wang, "Privacy preserving in cloud environment for obstructed shortest path query," *Wireless Pers. Commun.*, vol. 96, no. 2, pp. 2305-2322, 2017.
- [37] [M. Barni-2010] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, and A. Piva, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proc. 4th IEEE Int. Conf. Biometrics: Theory, Appl. Syst. (BTAS)*, Sep. 2010, pp. 1-7.
- [38] [M. Gong-2020] M. Gong, Y. Xie, K. Pen, K. Feng, and A.K. Qin, "A Survey on Differentially Private Machine Learning," *IEEE Computational Intelligence Magazine (Review Article)*, vol. 15, no. 2, 2020.
- [39] [M. Muhan-2017] M. Mohan, M. K. Devi, and J. Prakash V3, "Homomorphic encryption-state of the art", *International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore, India, 2017.
- [40] [M. Matsumoto-2020] M. Matsumoto and M. Oguchi, "Speeding Up Sensor Data Encryption with a Common Key Cryptosystem combined with Fully Homomorphic Encryption on Smartphones," in *Proc. Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2019.
- [41] [M. Salem-2019] M. Salem, S. Taheri, and J.-S. Yuan, "Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system," *Computers*, vol. 8, no. 1, p. 3, 2019.
- [42] [M. Shen-2018] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 940–953, Apr. 2018.
- [43] [M. Yasuda-2014] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshihara, "Privacy-preserving wildcards pattern matching using symmetric somewhat homomorphic encryption," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2014, pp. 338-353.
- [44] [N.P. Smart-2010] N.P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes", *Proceedings of the 13th international conference on Practice and Theory in Public Key Cryptography*, May 2010.

- [45] [O. Goldreich-1986] O. Goldreich, S. Goldwasser and S. Micali, “How to construct random functions,” *Journal of The Association for Computing Machinery*, Vol. 33, No. 4, Oct. 1986, pp. 792—807.
- [46] [O. Ruan-2019] O. Ruan, Z. Wang, J. Mi, and M. Zhang, “New approach to set representation and practical private set-intersection protocols,” *IEEE Access*, vol. 7, pp. 64897–64906, 2019.
- [47] [P.K. Mishra-2018] P. K. Mishra, D. Rathee, D. H. Duong, and M. Yasuda, “Fast secure matrix multiplications over ring-based homomorphic encryption,” *IACR Cryptol. ePrint Arch.*, Tech. Rep. 663, 2018.
- [48] [PHESL] “PALISADE HOMOMORPHIC ENCRYPTION SOFTWARE LIBRARY”, version 1.11.2 [Online access] Available: <https://palisade-crypto.org/>
- [49] [Pyfhel] “Pyfhel: PYthon For Homomorphic Encryption Libraries”, version 2.3.1 [Online access] Available: <https://pypi.org/project/Pyfhel/>
- [50] [PySEAL] “PySEAL”, [Online access] Available: <https://github.com/Lab41/PySEAL>
- [51] [Q. Wang-2017] Q. Wang, K. Ren, M. Du, Q. Li, and A. Mohaisen, “SecGDB: Graph encryption for exact shortest distance queries with efficient updates,” in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2017, pp. 79-97.
- [52] [R. Lu-2017] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, “A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT”, *IEEE Access*, 2017.
- [53] [S. Benabbas-2011] S. Benabbas, R. Gennaro, and Y. Vaglis, “Verifiable Delegation of Computation over Large Datasets.” In. *Proc. Cryptol. Conf. Berlin, Germany*: Springer, 2011, pp. 111-131.
- [54] [SEAL] “Microsoft SEAL”, [Online access] Available: <https://www.microsoft.com/en-us/research/project/microsoft-seal/>
- [55] [S. Li-2021] S. Li, et al., “Lightweight Privacy-Preserving Scheme using Homomorphic Encryption in Industrial Internet of Things”, *IEEE Internet of Things Journal*, March 2021.
- [56] [S. Makwana-2017] S. Makwana, “An Application of Homomorphic Encryption on IoT based Green House”, *ICECDS*, 2017.
- [57] [S. Patel -2014] S. Patel, G. Shah, and A. Patel, “Techniques of data perturbation for privacy preserving data mining,” *Int. J. Advent Res. Comput. Electron.*, vol. 1, pp. 5–10, Mar. 2014.

- [58] [T. Morshed-2018] T. Morshed, D. Alhadidi, and N. Mohammed, "Parallel linear regression on encrypted data," in Proc. IEEE 16th Annu. Conf. Privacy, Secur. Trust (PST), Aug. 2018, pp. 1-5.
- [59] [T. Shortell-2018] T. Shortell and A. Shokoufandeh, "Secure feature extraction in computational vision using fully homomorphic encryption," in Proc. Future Technol. Conf. Cham, Switzerland: Springer, 2018, pp. 189-213.
- [60] [V. Nikolaenko-2013] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in Proc. IEEE Symp. Secur. Privacy, May 2013, pp. 334-348.
- [61] [V. N. Boddeti-2018] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Oct. 2018, pp. 1-10.
- [62] [W. Sun-2018] W. Sun, Z. L. Jiang, J. Zhang, S.-M. Yiu, Y. Wu, H. Zhao, X. Wang, and P. Zhang, "Outsourced privacy preserving SVM with multiple keys," in Proc. Int. Conf. Algorithms Archit. Parallel Process. Cham, Switzerland: Springer, 2018, pp. 415-430.
- [63] [X. Liu-2016] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," IEEE Trans. Inf. Forensics Security, vol. 11, no. 11, pp. 2401-2414, Nov. 2016.
- [64] [X. Meng-2015] X. Meng, S. Kamara, K. Nissim, and G. Kollios, "GRECS: Graph encryption for approximate shortest distance queries," in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 504-517.
- [65] [X. Yi-2014] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical k nearest neighbor queries with location privacy," in Proc. IEEE 30th Int. Conf. Data Eng., Mar./Apr. 2014, pp. 640-651.
- [66] [Y. Aono -2016] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Privacy-preserving logistic regression with distributed data sources via homomorphic encryption," IEICE Trans. Inf. Syst., vol. 99.D, no. 8, pp. 2079-2089, 2016.
- [67] [Y. Aono-2017] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Input and output privacy-preserving linear regression," IEICE Trans. Inf. Syst., vol. E100.D, no. 10, pp. 2339-2347, 2017.
- [68] [Y. Bai-2014] Y. Bai, L. Zhuo, B. Cheng, and Y. F. Peng, "Surf feature extraction in encrypted domain," in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Jul. 2014, pp. 1-6.

- [69] [Y.Yang-2019] Y. Yang, et al., “A Comprehensive Survey on Secure Outsourced Computation and Its Applications”, IEEE ACCESS, vol. 7, 2018.
- [70] [Y. Zhang-2014] Y. Zhang, L. Zhuo, Y. Peng, and J. Zhang, “A secure image retrieval method based on homomorphic encryption for cloud computing,” in Proc. IEEE 19th Int. Conf. Digit. Signal Process., Aug. 2014, pp. 269-274.
- [71] [Y. Zhang-2015] Y. Zhang, W. Dai, X. Jiang, H. Xiong, and S. Wang, “Foresee: Fully outsourced secure genome study based on homomorphic encryption,” in BMC Med. Informat. Decis. Making, vol. 15, no. 5, p. S5, 2015.
- [72] “PySyft: private Data Science in Python”, [Online access] Available: <https://openmined.github.io/PySyft/index.html>
- [73] C. Mouchet and J.-P. Bossuat, “Lattigo: A multiparty homomorphic encryption library in go,” in WAHC 2020 – 8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, 15 Dec. 2020.
- [74] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in Advances 1105 in Cryptology – ASIACRYPT 2017. Cham: Springer International Publishing, 2017, vol. 10624, pp. 409–437.
- [75] I. Chillotti, M. Joye, D. Ligier, J.-B. Orfila, and S. Tap, “CONCRETE: Concrete operates on ciphertexts rapidly by extending TfhE,” in WAHC 2020 – 8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, 15 Dec. 2020.
- [76] [Online access] Available: “<https://www.grandviewresearch.com/>”
- [77] N. Tanabodee et al., “CipherFlow: A Playground for Developing Privacy-Preserving IoT in Node-RED,” AINTEC '22: Proceedings of the 17th Asian Internet Engineering Conference, Decmber 2022.
- [78] Albrecht, et. al, “Homomorphic Encryption Standard”, HomomorphicEncryption.org, November 21, 2018, Toronto, Canada

ภาคผนวก ก.

กำหนดการและรายชื่อผู้เข้าร่วมประชุมเชิงปฏิบัติการ

กำหนดการประชุมเชิงปฏิบัติการ
วันศุกร์ที่ 3 พฤศจิกายน 2566 เวลา 09.00 – 16.00 น. ณ
ห้องประชุม Auditorium (CO-113) อาคารสำนักงานกลาง อุทยานวิทยาศาสตร์ประเทศไทย

09.00 – 09.30 น.	ลงทะเบียน
09.30 – 09.45 น. (15 นาที)	พิธีกรกล่าวต้อนรับ และกล่าวเปิดประชุม โดย <i>ดร.พนิดา พงษ์ไพบูลย์</i> รองผู้อำนวยการศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) และผู้อำนวยการศูนย์นวัตกรรมการผลิตยั่งยืน (SMC)
09.45 – 10.15 น. (30 นาที)	Privacy Enhancement Technologies (PETs) สำหรับ IoT/IIoT เพื่อความ เป็นส่วนตัวของข้อมูล IoT/IIoT ให้กับภาคอุตสาหกรรมไทย และแนะนำ CYBLION Platform ▪ สร้างความตระหนักรู้ถึงความสำคัญของ Privacy ของข้อมูลโรงงานอุตสาหกรรม ▪ ระบุปัญหา รู้ทันต่อ รู้แนวทางแก้ไข ▪ นำเสนอเทคโนโลยีใหม่ล่าสุด “CYBLION Platform” โดย <i>ดร.กสิกา สุขสมบุรณ์</i> ทีมระบบไซเบอร์-กายภาพ (CPS) กลุ่มวิจัยไอโอทีและระบบอัตโนมัติสำหรับงานอุตสาหกรรม (IIARG) ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)
10.15 – 10.30 น. (15 นาที)	พักรับประทานอาหารว่าง
10.30 – 12.00 น. (90 นาที)	Hands-on Workshop: CYBLION วิทยากรโดย <i>ดร.กสิกา สุขสมบุรณ์, NECTEC</i> <i>นายณัฐ เสฐธ ธนบดี, Bitkub Company</i> แบ่งกลุ่มและเตรียมอุปกรณ์/ระบบ ▪ ขั้นตอนการลง Node-RED ▪ ขั้นตอนการใช้งาน Node-RED เบื้องต้น ▪ ขั้นตอนการลงโปรแกรม Node-RED Edge-CYBLION, CipherFlow ▪ Playground ฝึกสร้างสูตรคำนวณด้วยตัวอย่างง่าย ๆ
12.00 – 13.00 น.	พักรับประทานอาหาร

13.00 – 15.45 น. (165 นาที)	Hands-on Workshop: CYBLION วิทยากรโดย <i>ดร.ภลิกา สุขสมบูรณ์, NECTEC</i> <i>นายณัฐเศรษฐ์ ธนบดี, Bitkub Company</i> <i>นายเอกรัตน์ ศักดิ์ถาวรเลิศ, สาขาวิทยาการจัดการคอมพิวเตอร์</i> มหาวิทยาลัยเกษตรศาสตร์ กำหนด USE CASE ที่ใช้ทำการทดลอง ▪ กำหนด USE CASE ที่ใช้ทำการทดลอง ▪ ลงโปรแกรม Mobile APP ▪ เริ่มต้นใช้งาน Mobile APP, EDGE, CLOUD ▪ การสร้าง project, edge, keys บน Mobile APP ▪ Pair Key ▪ สร้างสูตรคำนวณบน Cloud ▪ การสร้าง Dashboard
15.45 – 16.00 น. (15 นาที)	ระดมความคิดเห็น สรุปการประชุม ตอบข้อซักถาม และทำแบบสอบถาม ปิดการประชุม

"Privacy as a Service For IoT/IloT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IloT ด้วย CYBLION Platform"

วันที่ 3 พฤศจิกายน 2566 เวลา 09.00 - 16.00 น.

ณ ห้องประชุม Auditorium (CO-113) ชั้น 1 (อาคาร 1) สวทช. อุทยานวิทยาศาสตร์ประเทศไทย จังหวัดปทุมธานี

ลำดับ	ชื่อ-นามสกุล	ตำแหน่ง	ชื่อหน่วยงาน/องค์กร	อีเมล	ลงนาม
18	ไพบูลย์ ขอมเดช	วิทยากร ที่ปรึกษาโครงการ IoT			
19	สมประสิทธิ์ คตภูธร	Software Engineer			
20	วุฒิสักดิ์ หมอยา	Developer			
21	สร้อยสุดา ทรงพันธ์	MIS Software Engineer			
22	มงคล มั่นเขตกิจ	Programmer			
23	Chinnawat Chinnapun	Technology Evangelist			
24	Varottarit Kaseamsan	Senior Business Development			
25	ธนภูมิ ภูมิ	Technology Evangelist			
26	พศวีร์ พรมมหาราช	Technology Evangelist			
27	คุณานนต์ ภัทรวาทิน	Ass. Technology Associate			
28	ณัฐวุฒิ มีกล้า	รองผู้จัดการแผนก			
29	วิสิฐพร เอกวงค์มั่นคง	วิศวกร			
30	พัชร ดารงศักดิ์	รองผู้อำนวยการฝ่าย			
31	อรรถพล ฤกษ์มอริญญา	รองผู้จัดการแผนกวางแผนและควบคุมการผลิต			
32	ธนกร นาราพานิช	รองผู้จัดการ แผนกกลั่นและบรรจุน้ำมันพืช			
33	ศตวรรษ สอนเจริญ	System Support Engineer			
34	พชร บุญน้อม	System Support			
35	พงษ์เอก อุทธโยธา	SYstem Support Engineer			
36	ประวิทย์ จันทร์ประทีภ	Pre-sale engineer and support			
37	โชติระวี ยอดโพธิ์ทอง	System support engineer			

ดร.อภิสิทธิ์ ภูมิธรรม PSME
ดร.อภิสิทธิ์ ภูมิธรรม

"Privacy as a Service For IoT/IIoT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IIoT ด้วย CYBLION Platform"

วันที่ 3 พฤศจิกายน 2566 เวลา 09.00 - 16.00 น.

ณ ห้องประชุม Auditorium (CO-113) ชั้น 1 (อาคาร 1) สวทช. อุทยานวิทยาศาสตร์ประเทศไทย จังหวัดปทุมธานี

ลำดับ	ชื่อ-นามสกุล	ตำแหน่ง	ชื่อหน่วยงาน/องค์กร	อีเมล	ลงนาม
38	อัครดนัย ปุณณนท	Engineer Supervisor			
39	ศุภกร จันทรรักษ์	Engineer			
40	อติวุฒิ สถาพรบำรุงเผ่า	Engineer			
41	วัชรกร สิงหาบุตร	IT			
42	กิตติวรวิชัย ชำนาญยนต์	พนักงาน IT			
43	วราพร เสี่ยงมจิต	ผู้จัดการส่วนงานไอที			
44	ปรัชญา มหาวินัย	หัวหน้าส่วน			
45	จักรชัย กายา	หัวหน้าส่วน			
46	จุฑารัตน์ ภูไธยา	Programmer			
47	วัฒนพัชร จิตติประวัติ	system administrator			
48	yutthasak promtaeng	Business Development Engineer			
49	Ukrit Pungjitwisut	System Development Engineer			
50	วาสนา พูลผล	วิศวกร			
51	อดิรุจ สุพรหม	Product Development Engineer			
52	ภาณุพงศ์ จันทรีโส	วิศวกร			
53	ประเสริฐ รักสกลาม	System developer			
54	ประสงค์ โพธิ์สุวรรณ	MD			
55	ภัทรกฤษ พิศพรณ	ผู้จัดการฝ่ายบริหาร			
56	ทักษิณ บำเพ็ญ	ผู้จัดการแผนกวิจัยและพัฒนา			
57	อนันต์ อินทร์ทรง	GM			
58	บุญเรือน มะโนมัย	IT			

แบบลงทะเบียน

หน้าที่ 4 จาก 4

"Privacy as a Service For IoT/IIoT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IIoT ด้วย CYBLION Platform"

วันที่ 3 พฤศจิกายน 2566 เวลา 09.00 - 16.00 น.

ณ ห้องประชุม Auditorium (CO-113) ชั้น 1 (อาคาร 1) สวทช. อุทยานวิทยาศาสตร์ประเทศไทย จังหวัดปทุมธานี

ลำดับ	ชื่อ-นามสกุล	ตำแหน่ง	ชื่อหน่วยงาน/องค์กร	อีเมล	ลงนาม
59	ธีรบุณย์ เต็มวิจิตร	Manager	Digital Generation Industries Co., Ltd.	dgi.teeraboon@gmail.com	ธีรบุณย์
60	อวิวิชญ์ เกษมปัญญา	Engineer	SV Engineering Consulting	apiwich.kt@gmail.com	อวิวิชญ์
61	อัครเดช ภิรมย์	ICT	STATE ENERGY		
63	สุกฤษฎิ์ เกษมปัญญา	วิศวกร	ทีเอ็มซี/มอส์ จำกัด/มอส์ 6-1	Sekrit.krit@gmail.com	อ.สุกฤษฎิ์
64	วิศิษฐ์ การะเกตุ	ICT	"	wisutt_garagat@yahoo.com	วิศิษฐ์
65	สุเมธ คำจันทร์	Senior Training Engineer	MELFT	somate@mitsubishi.co.th	สุเมธ
66	ศุภวัฒน์ ขาสกุล	PR	NECTEC / PBRs	sa (สืบพันธุ์)	ศุภวัฒน์
67	อัครเดช ภิรมย์	วิศวกร	NECTEC / CPS	(สืบพันธุ์ มอส์ 6-1)	อ.อัครเดช
68	อวิวิชญ์ เกษมปัญญา	research assistant	NETEC	apiwat.cha@nectec.or.th	อ.อวิวิชญ์
69	สุวิมล คำจันทร์	NCR	NETEC / TA		สุวิมล
70	อัครเดช ภิรมย์	NCR	NECTEC / LAI	-	อ.อัครเดช

แบบลงทะเบียน

หน้าที่ 4 จาก 4

"Privacy as a Service For IoT/IIoT: เสริมความปลอดภัยและความเป็นส่วนตัวของข้อมูล IoT/IIoT ด้วย CYBLION Platform"

วันที่ 3 พฤศจิกายน 2566 เวลา 09.00 - 16.00 น.

ณ ห้องประชุม Auditorium (CO-113) ชั้น 1 (อาคาร 1) สวทช. อุทยานวิทยาศาสตร์ประเทศไทย จังหวัดปทุมธานี

ลำดับ	ชื่อ-นามสกุล	ตำแหน่ง	ชื่อหน่วยงาน/องค์กร	อีเมล	ลงนาม
59	ธนินทร ภิรมย์	ncr	NECTEC	Thaninrat.kla@ncr.nstda.or.th	ธนินทร
60	วราวุธ ภิรมย์	ncr	NECTEC	kwanchanok.hae@ncr.nstda.or.th	วราวุธ
62	นิราวรรณ ศรีสุข	ncr	NECTEC	nirawan.sri@ncr.nstda.or.th	นิราวรรณ
63	จิรัช ภิรมย์	ncr	NECTEC	Jirattikan.Subsomboon@nectec.or.th	จิรัช
64	สุวิมล คำจันทร์	ncr	NECTEC	suwankk.ard@ncr.nstda.or.th	สุวิมล
65	อัครเดช ภิรมย์	อ.อัครเดช ภิรมย์	NECTEC	pattra.champa@nectec.or.th	อ.อัครเดช
66					
67					
68					
69					

ภาคผนวก ข.
เอกสารประกอบการบรรยาย
การฝึกอบรมเชิงปฏิบัติการ

PRIVACY AS A SERVICE FOR IIOT/IOT



PRESERVE YOUR IOT DATA PRIVATE



Presented by
Kalika Suksomboon, PhD.
CPS – NECTEC





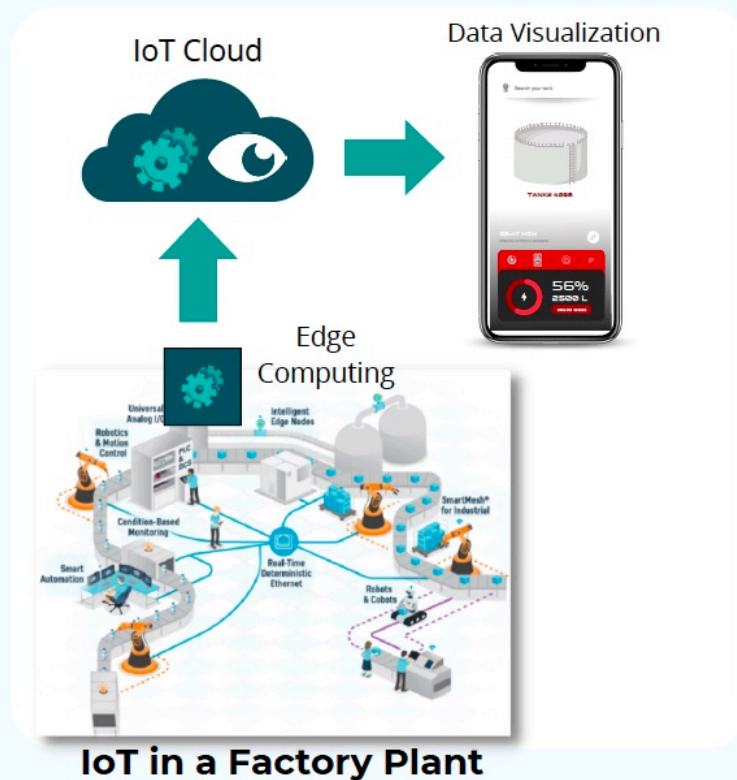
IIOT/IOT FOR SMART FACTORY

Make your factory smart empowered by data monitoring and analysis in the cloud

SHARE YOUR (SENSITIVE) DATA TO THE CLOUD



TRADITIONAL IIOT SOLUTION



Cybersecurity protection



Offload data computation and data distribution by cloud computing

WHAT'S WRONG?

- Need a trust of the third party such as the Cloud Service Provider
- Not guarantee the cloud will not be attacked (i.e., Ransomware, Data breach)
- The cloud can see your data while it is processing your data and distributing your data.



Personal Data And Order Histories Of Amazon Japan Exposed - September 2019



Amazon Data Breaches History and Full Timeline Up to 2023 3

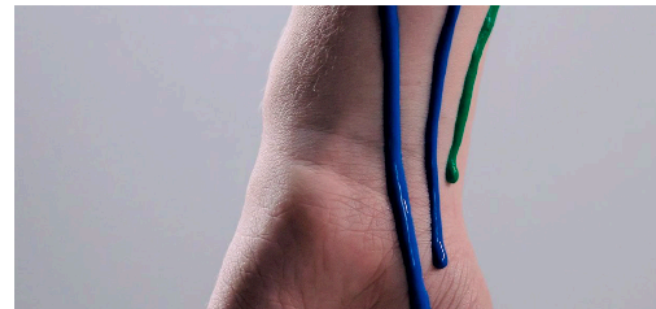
Amazon Japan users encountered a strange thing in September 2019 when the order histories of other shoppers were viewable to them.

Microsoft data breach "BlueBleed" exposes 2.4TB of customer data

Customers being told GDPR disclosure unnecessary

ELIOT BEER

October 20, 2022 . 11:21 AM — 4 min read



A misconfigured Microsoft endpoint (an Azure storage bucket) exposed 2.4TB of Microsoft customer information, including emails and signed documents, the company has admitted. Strikingly the company's support staff appear to be telling customers that Microsoft has no GDPR obligations to report it to regulators.

Ref: <https://www.thestack.technology/microsoft-data-breach-azure-bluebleed> 

PRIVACY AS A SERVICE PLATFORMS



Security Services

- Legacy privacy definition, called "**Privacy Tech 1.0**"
- To use security technologies to protect data leakage from untrusted or unauthorized parties.



Privacy by Policy

- Privacy-by-policy is used to enforce compliance and accountability of privacy protection.
- To make privacy protection is a necessity for every service provider.
- I.e., GDPR, PDPA, HIPAA



Privacy Enhancing Technologies

- Enable privacy protection strategies to be implemented in engineering.
- Use technologies such as
 - k-anonymous,
 - data marking,
 - federated learning,
 - differential privacy, TEE
 - SMPC, ZKP, **HE**, etc.

2022.09.24



EN|VIEL

Homomorphic
Encryption for
financial and
Healthcare



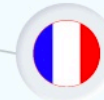
TripleBlind

Federated
computing and
secure MPC provide
API for data
scientists



INPHER

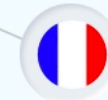
XOR with MPC and
secret computing
with TFHE



ZAMA

TFHE for ML and
enable Web2 and
Web3

ZAMA



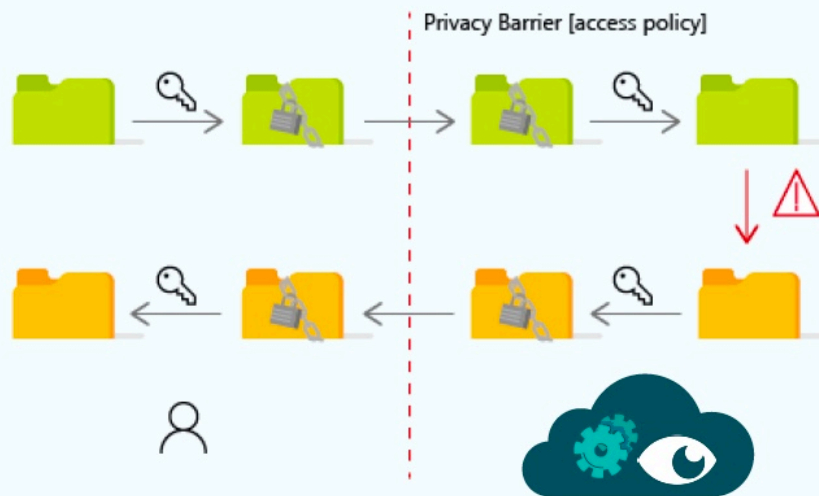
RAVEL

Ravel Homomorphic
Encryption
Provide Encrypted
Financial
Exchanges, GDPR,
Large Scale
Confidential Data
Sharing & Analytics



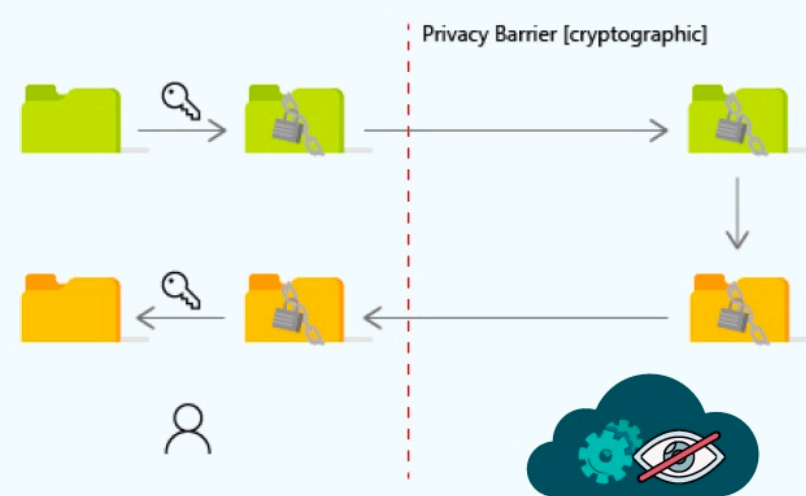
PETs

Traditional cloud computation



Fully Trusted Third Party

Homomorphic cloud computation



Semi Trusted Third Party

HOMOMORPHIC ENCRYPTION

01

PHE

PHE provides good performance and protection, but limited utility. Support either "Add" or "Multiply" operations, not both.

02

SHE

SHE permits fewer additions and multiplications on encrypted data. Noise grows due to the number of operations. So, it requires bootstrapping to reduce the noise.

03

FHE

FHE allows you to compute additions and multiplications on encrypted data. However, the more complex the operation, the more resource and time may be required.

CHALLENGES

- Most FHE schemes support only **positive integer**
- Most FHE support single only **"ADD"** and **"Multiplication"** on positive integer
- No scheme support **"Divider"** operation

HOW WE SLOVE IT

- Use the **CKKS scheme** to support Fixed-Point Number --> Real number
- Support **"Negative"** & **"Positive"** values
- Propose a New Algorithm for **Divider operation** without division!!!



CKKS IN A NUTSHELL

A security parameter is a way of measuring of how "hard" it is for an adversary to break a cryptographic scheme.

CKKS PARAMETER SETTING

λ Define Security Level

Q Coefficient modulus $Q = \prod_{i=1}^L q_i$

n Polynomial modulus

Δ Scale Factor

$P = (n, Q, \Delta)$

$p_L \rightarrow p_{L-1} \rightarrow p_{L-2} \rightarrow \dots \rightarrow p_1$

n	Bit-length of coefficient modulus Q $Q(\lambda, n)$		
	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



Step 1: Select $\lambda = 128$

Step 2: Choose coefficient modulus q_i

Step 3: Determine number of levels (operations)

Step 4: Determine polynomial modulus n

CKKS IN A NUTSHELL

CKKS PARAMETER SETTING

λ Define Security Level

Q Coefficient modulus $Q = \prod_{i=1}^L q_i$

n Polynomial modulus

Δ Scale Factor

$P = (n, Q, \Delta)$

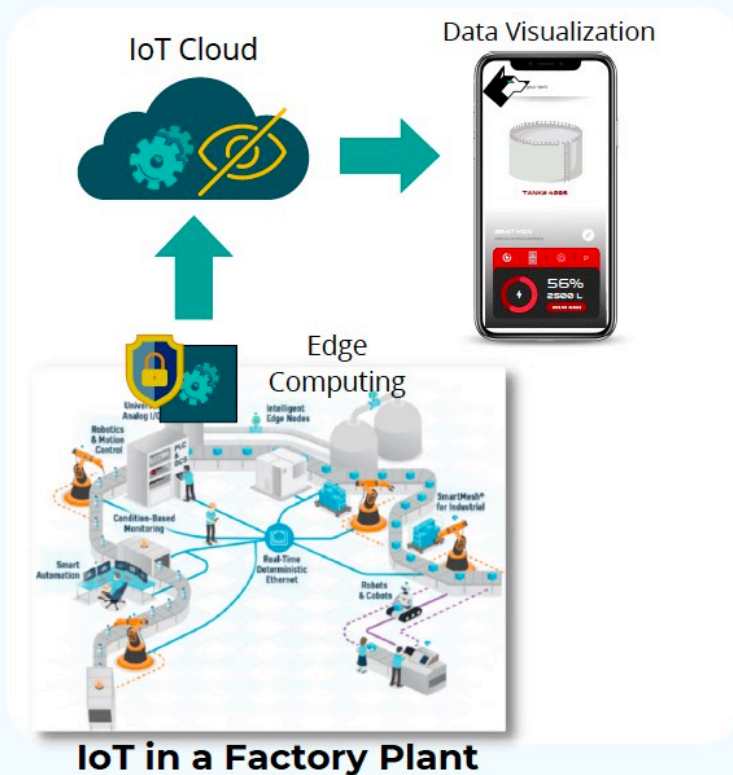
$p_L \rightarrow p_{L-1} \rightarrow p_{L-2} \rightarrow \dots \rightarrow p_1$

CYBLION Offers

ขนาดของพารามิเตอร์	โพลีโมดูลัส	โคเอฟฟิเชียนโมดูลัส	สเกล แฟคเตอร์	จำนวนครั้ง ของการคูณ
Small 140 kB	8192	$[2^{50}, 2^{30}, 2^{30}, 2^{50}]$	2^{30}	3
Medium 146 kB	8192	$[2^{60}, 2^{40}, 2^{40}, 2^{60}]$	2^{40}	3
Large 174 kB	16384	$[2^{60}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{40}, 2^{60}]$	2^{40}	7



WITH CYBLION'S SOLUTION



WHAT IS CYBLION?

CYBLION is a privacy-enhancement technology (PET) as a service for IoT platform

- We guarantee your data is secure and private.
- No one else but only you can see your data.
- The cloud cannot see your data while it is processing your data and distributing your data.



Cybersecurity protection
end-to-end from attackers

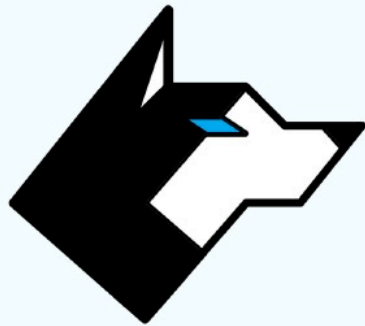


Offload data computation and
data distribution by cloud
computing



Prevent data breach to the third
party





TRUSTED HUSKY

CYBLION aims for quality privacy enhancement as a service for users. The name is composed of "Cyber" and "Brilliant," conjugated as "CYBLION."

Fortunately, the name sounds similar to the **Siberian husky** as the symbol of agility, trust, and power. The icon is simplified from a **Siberian husky** to convey a sense of modernity to fit the nature of the cybersecurity technology brand.

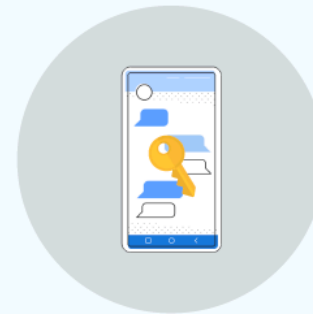
WE OFFER



Offloading computing
data to the cloud
without revealing your
data



Encrypt data at the
edge computing



Control your data
privacy via
our mobile app

HOW DOSE IT WORK?

FACTORY USE CASE



DESIGN

We design a solution to fit with the Factory's use case

SMART TANK FARM MONITORING SYSTEM



PRIVATE DATA MONITORING

The factory's staff can monitor the level of oil in their tanks in real-time via mobile phone



CYBLION REQUIREMENTS



Cyblion App in
iOS & Android

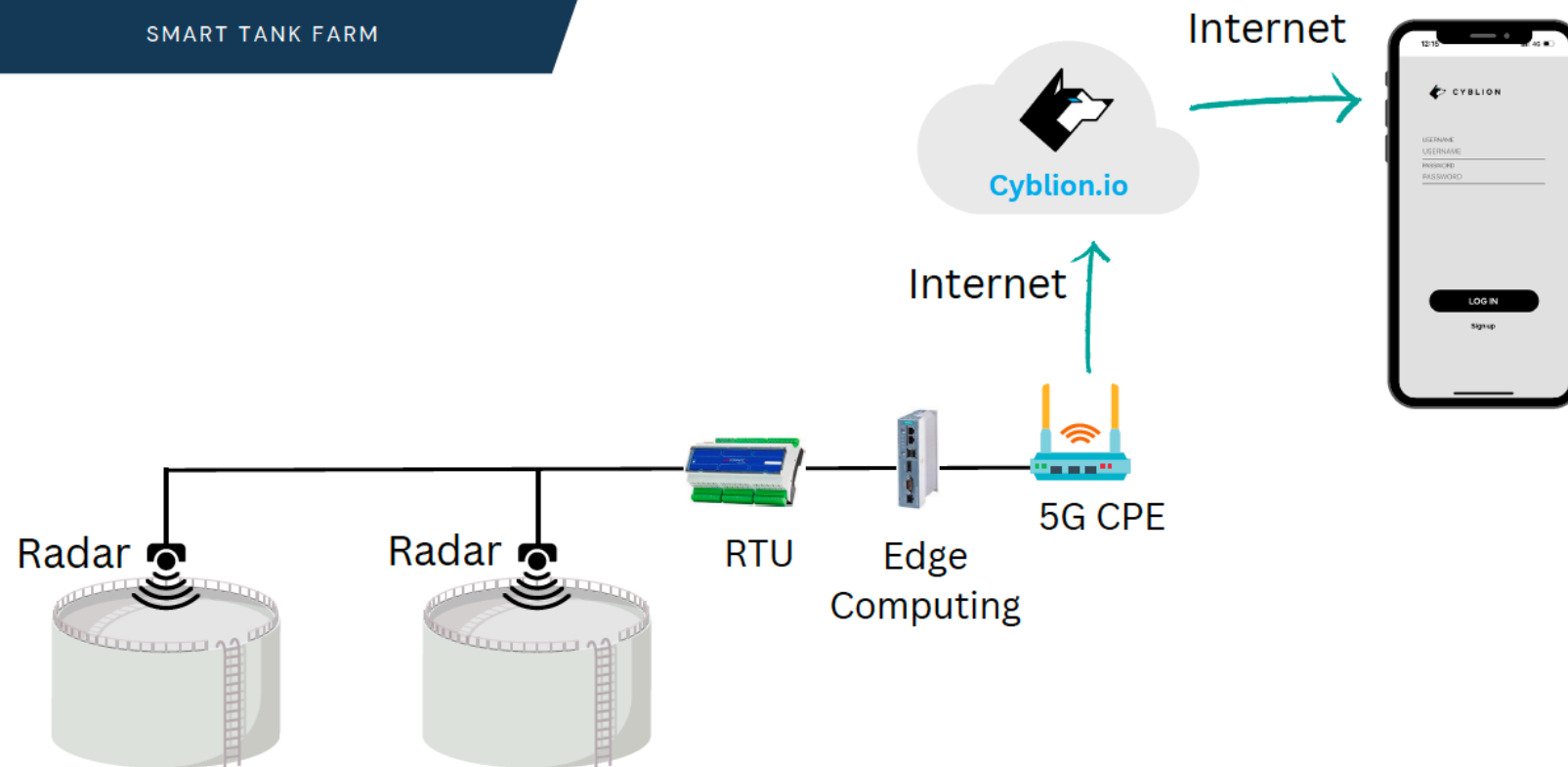


Node-RED installation
in the Edge computing

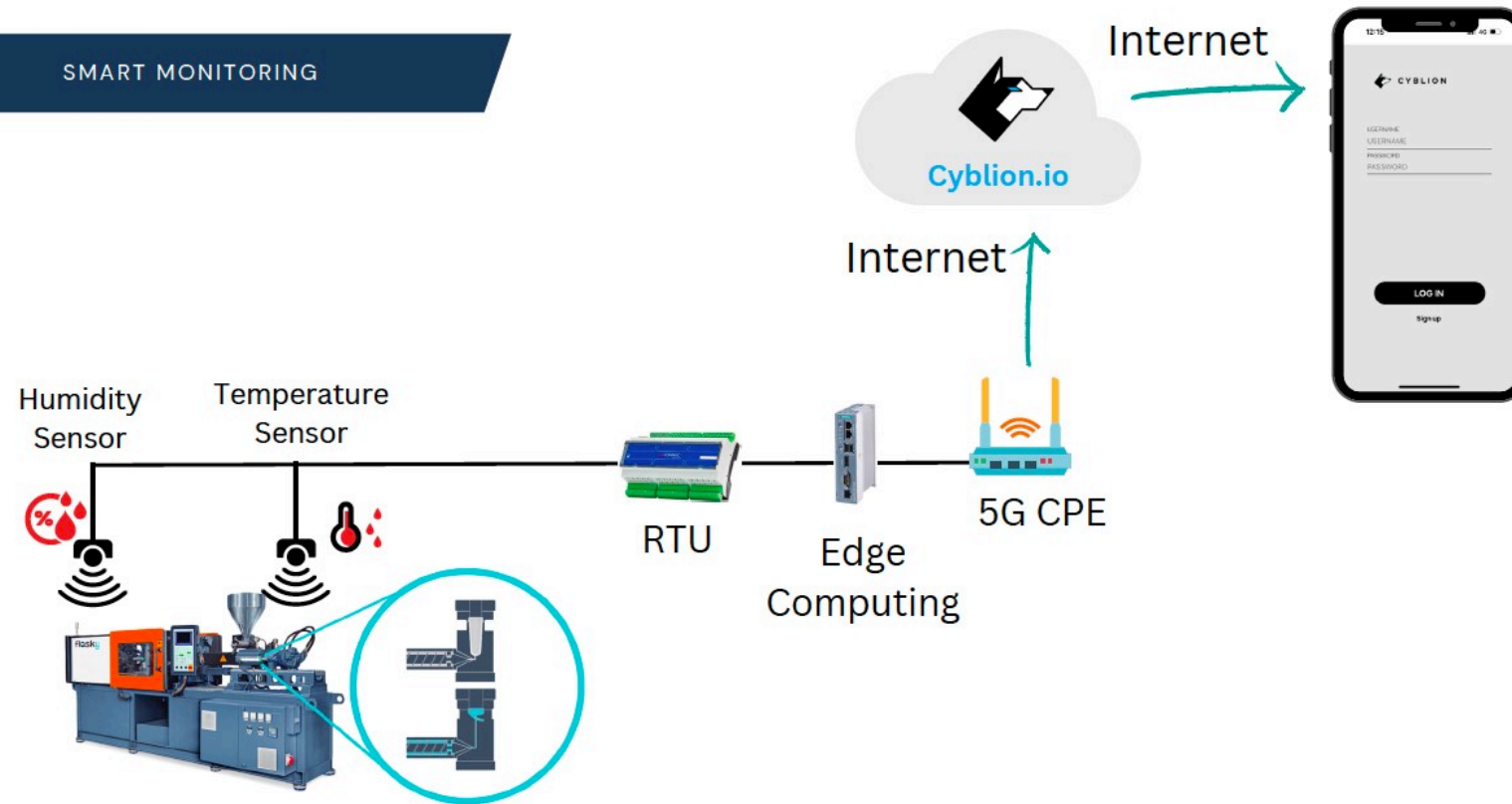


Cloud computing
with a NETPIE user account

SMART TANK FARM



SMART MONITORING



REQUIREMENTS

WE NEED FROM YOU



USE CASE

- Data monitoring, and transformation
- Data analysis



EDGE COMPUTING

- Install "node-red-cyblion-edge" from the node-RED pallet
- Add "node-red-cyblion-edge" in the end of node-red flow of the edge computing



TRY IT FREE

- We offer free trial until 31 March 2024
- 1 Cyblion Account on Cyblion.io
- 3 Projects with 10 Edges quota

INSTALLATION FREE

- Edge device (e.g., IoT2050)
- Other network devices if required



TO MAKE IT REAL

WORK CLOSELY WITH US

Require a use case and a dedicated your staff to
work closely with our deployment team



CHALLENGES FOR THE FUTURE

RESEARCH & DEVELOPMENT



PERFORMANCE

We still need improvement on the computation complexity and ciphertext size



SCALABILITY

There is some rooms to improve the scalability of distributed edge computing

OUR BEST TEAM

Project Team



**Panita
Pongpaibool**
ADVISOR

Research Team



**Kalika
Suksomboon**
CRYPTOGRAPHY/
PROJECT MANAGEMENT



**Aimaschana
Niruntasukrat**
IOT EDGE COMPUTING/
RESEARCHER



**Koonlachat
Meesublak**
IOT CLOUD PLATFORM/
RESEARCHER



**Sophon
Mongkolluksamee**
CYBERSECURITY/ RESEARCHER

OUR BEST TEAM

Development Lead



**Chavee
Issariyapat**

DEVOPS LEAD AND BACKEND
CLOUD DEV

Development Team



**Nattapon
Tansangworn**

NETPIE ARCHITECT



**Nataset
Tanabodee**

EDGE & CLOUD DEVOPS
SOFTWARE ENGINEER



Tawan Hohum

MOBILE APP DEVOPS
RESEARCH ASSISTANT



**Eakarat
Saktawornlerd**

MOBILE APP DEVOPS
SOFTWARE ENGINEER

[Home](#)

[About](#)

[Service](#)

[Contact](#)



- THANK YOU -



JOIN US!
IF YOU ARE INTERESTED



ภาคผนวก ค.
บทความในงานประชุมวิชาการ

ScottishFold: CatBoost-Enabled Lightweight Autonomous Smart Home Device Classification

Tawan Hohum[†], Kalika Suksomboon[†], Aimaschana Niruntasukrat[‡],
Sophon Mongkolluksamee[‡], Natapon Tansangworn[†]

[†]National Electronics and Computer Technology Center (NECTEC), Thailand

[‡]Srinakharinwirot University, Bangkok, Thailand

tawan.hoh@nec.nstda.or.th, kalika.suk@nectec.or.th, aimaschana.niruntasukrat@nectec.or.th,

sophon@g.swu.ac.th, natapon.tan@nectec.or.th

Abstract—Massive deployed IoT devices in smart homes pose security management challenges due to different hardware and firmware. To address these challenges, an automated IoT-device-type identification has been used as the first step in autonomous networks in which IoT security policy is automatically managed. Although machine-learning-based traffic patterns were proven high accuracy on a prediction while heavy on preprocessing, we argue that good stability and less computational cost should be considered. We focus on constructing an IoT-device-type classification based on the common features extracted from the network traffic. Inspired by the categorical boosting algorithm, we propose *ScottishFold*, a classification model, that is constructed from the dirty categorical IoT traffic features with less preprocessing. *ScottishFold* can classify the IoT device type from a single packet (either a signaling handshake or a data packet) that makes the classification process light. We boost the classification accuracy by integrating a machine reasoning approach into the classification process. The experimental results confirm the effectiveness and robustness of our proposed technique.

I. INTRODUCTION

The wide adoption of numerous heterogeneous Internet-of-Things (IoT) devices in smart home environments brings security management challenges according to their different requirements. Specifically, the difficulty poses to consumers how to efficiently manage the IoT devices' security due to their incredibly hardware and firmware diversities. Nevertheless, without wise security management, IoT devices can be easily exploited by attackers.

Through artificial intelligence (AI), an *autonomous network* enables an adaptive network that could be self-provisioning, self-optimizing, and self-diagnosing to mitigate the difficulty of security management. Alongside this vision, an automated IoT-device-type identification has been used as the first step in autonomous networks in which a security policy is automatically and optimally managed. Moreover, successfully automated identifying IoT devices can be vital in identifying illegitimate devices and unauthorized devices that risk associated with a security incident.

Traditional IoT-device-type identification approaches [1]–[5] rely on aggregated statistics of dense network traffics. These approaches require a considered amount of storage to collect the communication traffic log and cloud-computing resource. Additionally, IoT devices usually produce low traffic resulting in long waiting times to complete the device

identification. Therefore, we argue that those techniques are considered ineffective for a smart home usage.

Meanwhile, several recent works, e.g., [6], have proven the edge node is the best place to be the first-line management of IoT devices. We further ask whether the edge node can help automatically identify IoT devices due to its limited resource constraints. We address this problem by developing a lightweight automated-IoT-device-type identification so that we can utilize a paradigm edge computing and deploy the identification function close to the IoT devices. Accordingly, there is a need for a simple classification technique that can identify the IoT device types from a small number of communication packets.

To this end, we address those aforementioned issues and make contributions as follows. First, we study the effect of a classification model that has been trained with a single-device-per-type dataset (Dataset#1). The classification results turn to overfit in practices. To overcome this problem, our datasets were collected from two or more IoT devices per type and combined with the open datasets [7]. Unlike the existing work, we use different datasets for training and testing to confirm the robustness of our classification model. In addition, our datasets include all operating states of IoT devices, i.e., *initial*, *active*, and *standby*, so that the classification model, which is trained with these datasets, can identify the IoT device type from a packet of any operating states.

Second, motivated by [8], the host IP addresses can be used as a feature for the device type identification. However, IP address cannot directly be adopted as a feature of a machine learning algorithm due to facing the dirty categorical feature problem. We address this problem by leveraging the categorical boosting (CatBoost) algorithm [9], a machine learning (ML) algorithm for handling dirty categorical features. Hence, we propose, *ScottishFold*, a lightweight IoT-device-type classification based on the CatBoost algorithm.

Third, a machine reasoning approach (including two reasoning functions based on source/destination IP address mapping (sdIP2vec) and OUI translation (MAC2oui)) is proposed to boost the accuracy of classification. Finally, we evaluate the performance of our proposed method in terms of accuracy and operating costs.

The rest of this paper is organized as follows. Section II

reviews the related work. Section III proposes *ScottishFold*. Section IV presents how to collect the dataset. Section V evaluates the performance of the proposed method. Section VI concludes this paper.

II. RELATED WORK

Various approaches [1]–[5] proposed establishing the device fingerprint from collecting a long history of device communications. Y. Meidan, et al., [2], [3] utilized full TCP sessions to identify the IoT device type to create a white list of authorized devices. This method requires waiting for all packets from SYN to FIN to establish a device fingerprint. S. Marchal, et al., [1] have argued that only N packets of TCP session are enough for constructing a device fingerprint. The authors captured bidirectional packets and extracted features from the sizes of N sent and received packets, and $N-1$ inter-arrival times between those sent and received packets. However, most IoT devices often operate in *standby* states while some IoT devices do not receive any TCP session in a *standby* state. Hence, these techniques are infeasible. IoT SENTINEL [10], on the other hand, uses the packets of the *setup* state to identify an IoT device type. This technique is based on the two-fold classification system. First, IoT SENTINEL has been trained from a single class against the other classes, causing the number of binary trees scaled by the number of classes. Second, the unknown fingerprint has been treated by the *distance-based metric*. This technique requires the system to collect N packets of the *setup* state to establish 23 features and builds a single type classification by the Random Forests (RFs) algorithm. Unlike [10], S. Marchal, et al., [4] derived the device fingerprint from periodic communications of IoT device in any states of the IoT-device operations, i.e., *setup*, *standby*, and *active*. The authors used a discrete Fourier transform and signaled autocorrelation to identify several distinct periods. However, this technique requires pre-processing the network traffic and divide the traffic into different distinct flows. A simpler technique [11] identifies the IoT device type by the servers' names and their IP addresses which the IoT devices regularly communicate with. The intuition is that a particular IoT device regularly communicates with the same cloud service server. However, the different IoT device types from the same brand often communicate with the same server causing ambiguous identification. Closest to our work is SysID in [12], where the IoT device type can be identified by only a single data packet of the IoT device communication in any operating states. SysID uses the header field information in different protocols that has a unique signature. This information provides clues for classification yielding high accuracy. The SysID required two steps of classification. Genetic Algorithm (GA) is used to select the important features in protocol headers and then deployed the decision trees for classification. However, GA scales by the number of features causing the algorithm heavy. To overcome those challenges, we develop *ScottishFold*, an efficient and lightweight classification model of IoT devices based on a single data packet of any operating states of the IoT device communication.

III. THE PROPOSED SCHEME

We first present the design goals of *ScottishFold*, and then introduce a technique to handle the categorical features of IoT traffic and the prediction shift due to different databases between training and testing. Next, we propose a machine reasoning approach to boost the classification accuracy. Finally, we propose the *ScottishFold* framework.

A. Design Goals of *ScottishFold*

We leverage a ML based approach to automatically identify the IoT device types, including seen and unseen IoT devices. *ScottishFold* is expected to place at a network gateway that normally has limited resources. Hence, we derive three design goals that *ScottishFold* must satisfy.

Less preprocessing data: the classification model must require light data cleaning to reduce the preprocessing overhead.

Less buffering IoT packet streams: The classification model requires only a single TCP/IP packet of IoT traffic regardless of IoT operating states for the identification.

Robustness: The classification model can identify a new IoT device that is just introduced to the network.

B. Handling Categorical Features

Consider a dataset $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}$ for $k = 1, 2, \dots, n$ samples, where $\mathbf{x}_k = (x_k^1, x_k^2, \dots, x_k^m)$ is a vector of m attributes of IoT traffic and $y_k \in \mathbb{R}$ is a label of IoT-device-type identifier. Define x_k^i the value of data in attribute i of the k th-training example in dataset $\mathcal{D}$. The goal of learning task is to train a function $F : \mathbb{R}^m \rightarrow \mathbb{R}$ which minimizes the expected loss function $\mathcal{L}(F) := \mathbb{E}(L(y, F(\mathbf{x})))$, where $L(\cdot)$ is a smooth loss function and $(\mathbf{x}, y)$ is a test sampled from probability P independently of the training set $\mathcal{D}$.

The existing approaches [2], [10], [12], [13] have adopted ML algorithms, e.g., Support Vector Machine (SVM), RFs, Decision Trees (DT) and Gradient Boosting (GB) to construct the classification model for IoT device type identification. Most of them rely on the attributes of traffic patterns and the frequency of incoming and outgoing packets. The variables reserved for these attributes are usually real numbers. Although feature engineering based on the traffic patterns can avoid facing the dirty categorical feature problem [14], extracting the traffic pattern and packet frequencies requires a considerable amount of memory resources and time-consuming due to tracking the consecutive packets and handcrafted feature engineering on cleaning the collected data.

Unlike those previous works, we choose the TCP/IP packet attributes (e.g., IP address, MAC address, source port, destination port, packet length, and protocol) to perform the ML features. The reason is that the TCP/IP packet attributes can be obtained directly from capturing only a single data packet. However, these attributes are discrete values and non-curved data called *dirty categories*. For example, the values of the IP address attribute are not comparable to each other. Moreover, the same type of two IoT devices in the different locations may communicate to the different host IP addresses. Typically, those data are collected in the different datasets from different

research laboratories and have never been shared. Even though the typical machine learning, such as Random Forests (RFs), Decision Trees (DT), and Gradient Boosting (GB), can deal with the continuous numerical values, they fail to handle the categorical features problem. Therefore, it turns into a problem of mapping the categorical features to feature vectors.

One-hot encoding [15] and bag-of-words classifiers [16] are widely-used techniques for handling categorical features. However, they are impractical for applying to non-curved categorical data, leading to large cardinality of the categorical variable. The other challenge of applying these techniques to the dirty categorical features is that once the encoding categories do not appear in the training dataset, it will cause error-prone in an inference process. Therefore, there is a need for a simple feature-engineering approach that can replace the widely-used one-hot encoding and bag-of-words methods.

Inspired by the CatBoost algorithm [9], a novel gradient boosting technique proposed by Yandex Company [9], we provide *ScottishFold*, a lightweight automated-IoT-device-type classification, that can deal well with the dirty categorical features. The CatBoost algorithm adopts the ordered target statistics (TS) to overcome target leakage problems by utilizing a random permutation of training samples. Therefore, *ScottishFold* can scale well and robust to the different statistics of the same categorical feature even though it is applied to different training and testing datasets. Hence, the dirty categorical features, such as IP address and MAC address, can be used to train the model without preprocessing with one-hot encoding or bag-of-words classifiers.

C. Handling Prediction Shift of Datasets

A major challenge towards realizing IoT device type classification lies in how it can effectively identify the new IoT device introduced to a network. A new IoT device's data may interfere with the pattern of existing IoT traffic resulting in a distribution shift. Nevertheless, the existing works rely on only their IoT traffic datasets, which contain one IoT device per type. It is unclear whether the existing trained models are general enough to identify unseen IoT devices. In this paper, we argue that training and testing on the same dataset lead to bias results even though the dataset is split into 70%-30% for training and testing, respectively (we show the effect of this issue in Section V). Meanwhile, the existing approaches have not considered the problem of prediction shift due to the different training and testing datasets. In contrast, we train our model with the CatBoost algorithm, which has addressed the prediction shift problem by utilizing the ordered boosting technique. Intuitively, reusing the same dataset for each iteration causes the base predictor bias. Therefore, the CatBoost algorithm trains the model with a different dataset for each iteration by the permutation.

D. Machine Reasoning Approach

Observation: We have examined the result of training the classification model with the CatBoost algorithm based on eight attributes (i.e., ipSrc, ipDst, macSrc, macDst, protocol,

pktLength, portSrc, and portDst) of IoT traffic. Its confusion matrix is shown in Fig. 4a, and the classification performance is presented in Table I.

The observation results (based on 37,479 tested samples) demonstrate that the classification turns to poor inference with 44.56% overall accuracy. Specifically, the classification cannot clearly distinguish between Nest IQ Cam (type#5) and Amazon echo (type#7). Similarly, 71.04% of the packets of TP-Link KC120 (type#3) have been identified as Nest IQ Cam. Although the CatBoost Algorithm can simplify how to manage the IP address and MAC address categories, it lets ipDst, macSrc and macDst features lead the classification. However, IP addresses under NAT are randomly assigned to the IoT devices that make the classification confusing. The similarity of IP addresses under NAT and MAC addresses of a network gateway as shown in Fig. 1 makes the classification harder to determine the correct type.

Our approach: To address the aforementioned problems, we develop two intermediate functions that define the reasoning process to the CatBoost algorithm.

sdIP2vec function: $\mathcal{F}_1 : \mathbb{R}^2 \rightarrow \mathbb{R}$, is defined as a mapping function of two attributes, i.e., source and destination IP addresses onto a string of an IP address excluding the IP addresses under NAT and the gateway. Let x_k^s, x_k^d be the attributes of ipSrc and ipDst of sample packet k , respectively. Define $\mathcal{N}$ be a set of IP addresses under NAT and $\mathcal{G}$ be a set of IP addresses of the gateway. The sdIP2vec takes dataset $\mathcal{D}$, $\mathcal{N}$, and $\mathcal{G}$ as the inputs. The sdIP2vec returns a vector $\hat{\mathbf{x}}$, where new attribute $\hat{x}_k^i \in \hat{\mathbf{x}}$ for $k = 1, \dots, n$ replaces the attributes of ipSrc and ipDst by a string of the IP address excluding the IP addresses under NAT and the IP address of the gateway.

MAC2oui function: $\mathcal{F}_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined as a mapping function of MAC address of IoT devices to Organizationally Unique Identifier (OUI). Let $\mathcal{M} = \{(o_j, s_j), j = 1, \dots, S\}$ be a set of tuples (o_j, s_j) , where o_j is the 3-octet of a MAC address) obtained from the OUI database and s_j is OUI j . Let x_k^i be the attribute of the MAC address of sample packet k . The MAC2oui takes dataset $\mathcal{D}$, $\mathcal{G}$ and $\mathcal{M}$ as the inputs. The MAC2oui returns a new attribute, called *mac*, where macSrc and macDst are mapped to a string of OUI in which the OUI of the gateway is excluded.

Note: Due to the space limit, we omit presenting the algorithms of these two functions in this paper, while we present them in our GitHub [17].

E. ScottishFold

We present a framework of *ScottishFold* in Fig 2. The framework is based on two intermediate functions i.e., sdIP2vec and MAC2oui concatenating with the CatBoost algorithm to train a model with a training dataset. The model is optimized by state-of-the-art parameter tuning, such as grid search. The obtained model is then validated by the testing dataset.

IV. DATASETS

This section presents our testbed architecture and the datasets using throughout this paper.

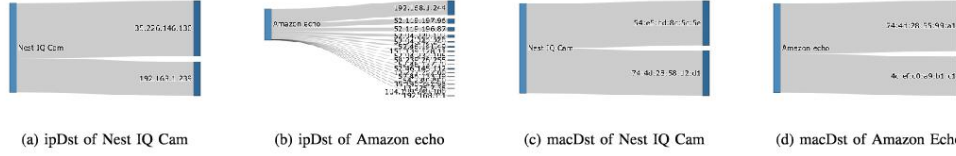


Fig. 1: Sankey diagram of network activities for Nest IQ Cam and Amazon Echo. A clear distinction is observed in terms of the IP addresses and MAC addresses of the destinations that the IoT devices used for exchanging the data.

TABLE I: The performance metrics of the observation based on Dataset#1: overall accuracy is equal to 44.56% from 37,479 total tested samples.

Type	Precision	Recall	F1-Score	Samples
0	1.00	0.74	0.85	479
1	0.71	0.98	0.82	332
2	0.15	0.84	0.26	932
3	1.00	0.29	0.45	7,853
4	1.00	0.46	0.63	908
5	0.61	0.63	0.62	13,787
6	0.59	0.41	0.48	1,230
7	0.55	0.63	0.59	11,958
micro avg	0.56	0.56	0.56	37,479
macro avg	0.70	0.62	0.59	37,479
weighted avg	0.68	0.56	0.56	37,479

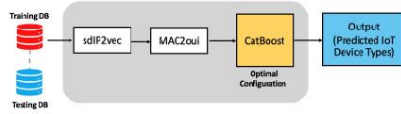


Fig. 2: ScottishFold framework.

A. Testbed architecture

Our IoT-device-type identification technique relies on passive monitoring of the network traffic at the network gateway. We set our testbed as shown in Fig 3. The smart home IoT devices connect to the network gateway (i.e., a Wi-Fi gateway). We select 18 smart home IoT devices consisting of two IP cameras, four smart speakers, three smart bulbs, a hub, seven smart plugs, and a toy robot as listed in Table II. We configure port mirroring on a network switch to capture the incoming and outgoing network traffic of the IoT devices passing through the network gateway. The source port connects to the network gateway, and the destination port connects to the database server. For analyzing network traffic passing through the ports, the network traffic is mirrored from a source port to the destination port. A Kali-Linux server connecting to a USB Wi-Fi running Aircrack-ng [18] performs a packet sniffer. Wireshark software is used for network protocol analysis. We control the smart home IoT devices via their mobile

applications on a smartphone and a tablet.

B. IoT Traffic Datasets

In this paper, we use three independent datasets collected based on the full PCAP files. **Dataset#1** consists of the PCAP file traces (for nonconsecutive three months) collected from IoT devices in our testbed as listed in Table II. **Dataset#2** shares some smart home IoT devices of Dataset#1, and we added the new smart home IoT devices in our testbed and collected their traffic in Dataset#2 in the different duration time of Dataset#1 (i.e., for another nonconsecutive four days). **Dataset#3** is obtained from the open datasets in [7]. We select only the network traffic of an Amazon echo device for testing due to its similar type as in both Dataset#1 and Dataset#2. Hence, we use Dataset#3 for inference to verify the generalization and robustness of our classification model.

The PCAP file traces of our datasets include three states of the smart home IoT device, i.e., 1) *initial* state (i.e., pairing an IoT device to its mobile application and the first time connecting to the gateway), 2) *active* state (i.e., interacting with the mobile application), and 3) *standby* state (i.e., not being interacted with the mobile application nor at the *initial* state). We will open our datasets on our GitHub [17].

We categorize our 18 smart home IoT devices into eight types according to their brands and the device functionalities as shown in Tabel II. For example, the smart plugs and smart bulbs are grouped into the same type, while the IP cameras of different brands are categorized into different types. The reason for doing so is that the smart plugs and the smart bulbs from the same brand use the same chipset and firmware. In contrast, different brands' IP cameras perform different functionalities because they have developed based on the different chipsets and firmware. Therefore, we treat them as different types.

V. PERFORMANCE EVALUATION

In this section, we aim to pursue the following goals: 1) accuracy of *ScottishFold*, 2) prediction shift due to the effect of different datasets between training and testing, and 3) the operating costs.

We conduct the experiments on a server with Intel Xeon CPU @ 2.5 GHz with 20 cores and 13.75MB L3-Cache memory. The proposed scheme is implemented using our own

TABLE II: 18 IoT devices with 8 types used in our dataset for identification.

Type	Device name	Mac address	Dataset			OUI	Device description
			#1	#2	#3		
#0	Anitech	cc:50:e3:da:00:7f	✓			Espressif Inc.	Smart wireless power strip
	Anitech	cc:50:e3:da:00:3f		✓		Espressif Inc.	Smart wireless power strip
#1	Sonoff B1	60:01:94:ac:93:31	✓			Espressif Inc.	Sonoff B1 LED Bulb
	Sonoff B1	60:01:94:ac:8f:fd		✓		Espressif Inc.	Sonoff B1 LED Bulb
	Sonoff S20	60:01:94:74:22:a6	✓	✓		Espressif Inc.	Sonoff S20 Smart plug
	Sonoff S26	cc:50:e3:00:68:e8	✓			Espressif Inc.	Sonoff S26 Smart plug
	Sonoff S26	84:f3:eb:3d:fa:f5		✓		Espressif Inc.	Sonoff S26 Smart plug
#2	TP-Link LB100	50:c7:bf:8d:87:b6	✓	✓		TP-Link Tech. Inc.	Wi-Fi LED Light Bulb
	TP-Link HS110	ac:84:c6:21:07:3e	✓			TP-Link Tech. Inc.	Wi-Fi Smart Plug
	TP-Link HS110	b0:4e:26:ae:47:e5		✓		TP-Link Tech. Inc.	Wi-Fi Smart Plug
#3	TP Link KC120	ac:84:c6:bf:fc:a5	✓	✓		TP-Link Tech. Inc.	Security IP Camera
#4	Philips Hue	00:17:88:b2:6b:0c	✓	✓		Philips Lighting BV	Philips Hue Bulbs
#5	Nest IQ Cam	18:b4:30:8f:88:a8	✓	✓		Nest Labs Inc.	Security IP Camera
#6	GuRobot	54:e5:bd:8c:5c:5e	✓	✓		NA	GuRobot
#7	Amazon echo	4c:ef:c0:a9:b1:c1	✓			Amazon Tech. Inc.	Amazon Echo
	Amazon echo dot	68:9a:87:31:d8:15		✓		Amazon Tech. Inc.	Amazon Echo
	Amazon echo spot	4c:17:44:df:1f:b6		✓		Amazon Tech. Inc.	Amazon Echo
	Amazon echo [7]	44:65:0d:56:cc:d3			✓	Amazon Tech. Inc.	Amazon Echo



Fig. 3: Testbed architecture.

program development based on Python. All datasets collected in PCAP files were interpreted in a database format (using SQLite database engine). We analyze the data using Pandas libraries in Python. We extract the features from eight attributes, i.e., source and destination of MAC addresses (macSrc and macDst), source IP address (ipSrc), destination IP address (ipDst), packet length (pktLength), source port (portSrc), destination port (portDst), and protocol. The extracted features are fed to the sdIP2vec and MAC2oui functions. The number of attributes is reduced from eight to six before training the model with the CatBoost algorithm. We ran three experiments with different combinations of datasets to demonstrate the effect of prediction shift.

Experiment 1: The classification model was developed based on Dataset#1, where 70% of data were used for training, and 30% of data were used for testing.

Experiment 2: The classification model was developed based on Dataset#1 for training and Dataset#2 for testing.

Experiment 3: The classification model was developed based on Dataset#1 for training and the combination of Dataset#2 and Dataset#3 for testing.

We ran each experiment ten times to obtain confident

results. In our experiments, the data imbalance problem was addressed by a random over-sampling method.

Figure 4b demonstrates the confusion matrix of the classification of **Experiment 1**. The overall accuracy of our model is 99.94% (98,313 tested samples). It is slightly unclear with 1% error between TP-Link smart bulb/plug (type#2) and TP-Link KC120 (type#3) due to the same OUI prefix. However, the classification can distinguish them by means of ipAddr and pktLength features. Comparing to the observation in Fig. 4a, the sdIP2vec and the MAC2oui significantly improve the classification accuracy. However, this experiment shows that training and testing on the same dataset resulting in overfitting, since the MAC address is the dominated feature with 75.85%.

We further explore the prediction shift effect due to the different datasets between training and testing in **Experiment2**. Figure 4c shows that the accuracy is dropped from 99.94% to 98.52% (48,986 tested samples). Adding a new TP-Link smart plug to the network causes the classification slightly confusion between TP-Link smart bulb/plug and TP-Link KC120 since the classification were not trained by that new device. In **Experiment 3**, we ran the test with 57,408 tested samples by adding 8,422 samples of a new Amazon echo packet from [7]. The confusion matrix in Fig. 4d shows the robustness of our classification model because it can identify a new Amazon echo from Dataset#3 with 100% accuracy. The reason is that MAC address is the dominant feature of classification, while the classification does not affect by the different source and destination IP addresses of the different network locations.

The operating cost consists of *i*) two intermediate functions and *ii*) classifier. The operating costs of two intermediate functions can be negligible for both training and inference processes. The classifier in the inference process spends 2.17 μ s, 2.33 μ s, and 1.86 μ s for **Experiment 1**, **Experiment 2**, and **Experiment 3**, respectively. The results show that our classification model is lightweight.

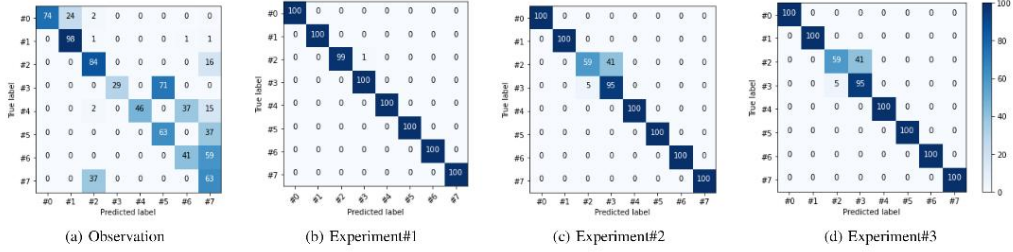


Fig. 4: **Observation:** The feature importance scores are ranked as ipDst: 45.26%, macSrc: 26.43%, macDst: 20.61%, ipSrc: 6.40%, protocol: 1.25%, pktLength: 0.042%, portSrc: 0.0%, and portDst: 0.0% and the overall accuracy is equal to 44.56% from 37,479 total tested samples. **Experiment 1:** The feature importance scores are ranked as mac: 75.85%, ipAddr: 10.83%, pktLength: 6.66%, portSrc: 3.10%, portDst: 2.89%, and protocol: 0.66%. **Experiment 2:** The feature importance scores are ranked as mac: 45.25%, ipAddr: 29.46%, pktLength: 11.57%, portDst: 7.60%, portSrc: 4.71%, and protocol: 1.38%. **Experiment 3:** the confusion matrix is the same as that of Experiment 2, with 1.00 precision, recall and F1-score for 32,266 tested samples of Amazon echo's TCP packets (8,422 new samples were added).

VI. CONCLUSION AND FUTURE WORK

This paper proposed an automated IoT-device-type classification framework, called *ScottishFold*. *ScottishFold* is aimed to be processed in an edge node since it is the best place for the first-line management of IoT devices. Nevertheless, deploying sophisticated machine learning and deep learning techniques in the edge node is infeasible due to the resource constraints of the edge node. Therefore, the development of *ScottishFold* was aimed to satisfy three requirements, i.e., less preprocessing data, less buffering IoT packet streams, and robustness. The classification model was constructed based on eight common attributes extracted from the network traffic, and it uses only a single TPC packet for the identification. By means of the CatBoost algorithm, our classification model can be constructed based on the dirty categorical features of IoT traffic attributes with less preprocessing and robustness to the unseen devices. The machine reasoning approach has been introduced (i.e., sdIP2vec and MAC2oui) to boost the classification accuracy. The evaluation results from observation and three experiments with three different datasets demonstrate the proposed method's effectiveness and robustness. The operating cost of our technique shows its lightweight classification.

In the future, we plan to develop a packet sniffer integrated with *ScottishFold* and deploy it in the edge node, e.g., Raspberry Pi. Regarding the privacy concern, the sniffed packets will not refer to human activity analysis. Moreover, we expect to incorporate with the other fine-tune packet features to make *ScottishFold* more robust for new device types.

ACKNOWLEDGMENTS

This work was funded and financially supported by NBCT, Thailand, under the grant A63-1-(2)-017.

REFERENCES

- [1] M. R. Shahid and et al., "Iot devices recognition through network traffic analysis," in *IEEE Big Data*, 2018, pp. 5187–5192.

- [2] Y. Meidan and et al., "Detection of unauthorized iot devices using machine learning techniques," *CoRR*, vol. abs/1709.04647, 2017.
- [3] Y. Meidan and et al., "Profilot: A machine learning approach for iot device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, 2017, pp. 506–509.
- [4] S. Marchal and et al., "Audi: Toward autonomous iot device-type identification using periodic communication," *IEEE JSAC IN Communications*, vol. 37, pp. 1402–1412, 2019.
- [5] I. Cvitić and et al., "Ensemble machine learning approach for classification of iot devices in smart home," *Int. J. Mach. Learn. Cyber.*, pp. 1–24.
- [6] K. Bhardwaj, J. C. Miranda, and A. Gavrilovska, "Towards iot-ddos prevention using edge computing," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge)*, 2018, pp. 1–7.
- [7] A. Sivanathan and et al., "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 8 2018, an optional note.
- [8] H. Guo and J. Heidemann, "Detecting iot devices in the internet (extended)," Tech. Rep. ISI-TR-726B, Jul. 2018.
- [9] L. Prokhorenkova and et al., "Catboost: Unbiased boosting with categorical features," in *32nd NIPS*, 2018, pp. 1–11.
- [10] M. Miettinen and et al., "Iot sentinel: Automated device-type identification for security enforcement in iot," in *IEEE ICDSCS*, 2017, pp. 2177–2184.
- [11] H. Guo and J. Heidemann, "Ip-based iot device detection," in *ACM IoT S&P*, 2019, pp. 36–42.
- [12] A. Aksoy and M. H. Gunes, "Automated iot device identification using network traffic," in *IEEE ICC*, 2019, pp. 1–7.
- [13] A. Sivanathan and et al., "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Computing*, vol. 18, pp. 1745–1759, 07 2018.
- [14] P. Cerda, G. Varoquaux, and B. Kegl, "Similarity encoding for learning with dirty categorical variables," *Mach Learn*, vol. 107, pp. 1477–1494, 2018.
- [15] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *SIGKDD Explorations*, vol. 3, pp. 27–32, 07 2001.
- [16] J. Nowak and et al., "Random forests for profiling computer network users," in *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2018, pp. 734–739.
- [17] T. Hohum, K. Suksomboon, and N. Tansangworn, (2021) *Scottishfold*. [Online]. Available: <https://github.com/donkyduck/ScottishFold-on-Edge>.
- [18] *Aircrack-NG. A Network Software Suite to Assess WiFi Network Security*, 25 Jan 20 (accessed September 1, 2020). [Online]. Available: <https://www.aircrack-ng.org/index.html>.

CipherFlow: A Playground for Developing Privacy-Preserving IoT in Node-RED

Nataset Tanabodee
nataset.t@ku.th
Kasetsart University
Bangkok, Thailand

Kalika Suksomboon
kalika.suk@nectec.or.th
NECTEC
Pathum Thani, Thailand

Chaveee Issariyapat
i@chavee.com
NEXPIE Co. Ltd.
Bangkok, Thailand

Sophon Mongkolluksamee
sophon@g.swu.ac.th
Srinakharinwirot University
Bangkok, Thailand

Aimaschana Niruntasukrat
aimaschana.niruntasukrat@nectec.or.th
NECTEC
Pathum Thani, Thailand

Natapon Tansangworn
natapon.tan@nectec.or.th
NECTEC
Pathum Thani, Thailand

Sukumal Kitisiin
sukumal.i@ku.th
Kasetsart University
Bangkok, Thailand

ABSTRACT

The emergence of privacy breaches in IoT surges demands for embedding privacy-by-design into IoT systems. Decades of work on privacy-enhancing technologies have resulted in myriad ways to secure outsourcing computation with homomorphic encryption (HE) that allows the third party (e.g., cloud) computes over encrypted data. All of them are still yet far away from the rapid development of practical privacy-preserving IoT systems. Abstraction and flexibility in development are keys to accelerating privacy-preserving IoT deployment. Nevertheless, HE tools and compilers are primarily written in low-level languages like C++, posing a high entry barrier to IoT developers. This paper proposes *CipherFlow*, the HE node extension for Node-RED, a visually and flexibly programmable dataflow tool for rapid IoT development. *CipherFlow* offers three pipeline abstractions: 1) a playground of drag-and-drop HE nodes in Node-RED, 2) arithmetic operators, and 3) optimal parameter setting guidelines. As a result, IoT developers can create and verify their secured computing functions on data flows. We demonstrate the applicability and effectiveness of *CipherFlow* through the performance evaluation. Furthermore, our evaluation shows the performance and security trade-off depending on the HE parameters.

CCS CONCEPTS

• **Networks** → Network reliability.

KEYWORDS

Privacy, IoT, homomorphic encryption, Node-RED

ACM Reference Format:

Nataset Tanabodee, Kalika Suksomboon, Chaveee Issariyapat, Sophon Mongkolluksamee, Aimaschana Niruntasukrat, Natapon Tansangworn, and Sukumal Kitisiin. 2022. CipherFlow: A Playground for Developing Privacy-Preserving IoT in Node-RED. In *The 17th Asian Internet Engineering Conference (AINTEC'22)*, December 19–21, 2022, Hiroshima, Japan. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3570748.3570752>

1 INTRODUCTION

Widely used and deployed, the Internet of Things (IoT) has shifted the predominant usage of cloud towards IoT cloud computing, such as Azure IoT hub and Amazon IoT, which provides simple management services for billions of IoT devices. The agility and simplicity of IoT cloud computing encourage newbie IoT developers to enjoy creating new IoT solutions for the market. However, recent IoT data breach news has escalated the severity of users' privacy concerns. Supported by the survey in [15], respondents have admitted that privacy is the biggest challenge for IoT adoption.

Embedding privacy by design into IoT systems is a key to lifting confidence in IoT usage. Once sensitive information in data has been adequately constrained through privacy-enhancing technologies (PETs), such as homomorphic encryption (HE) and k -anonymization, the privacy-by-policy approaches become less important. Although many efforts [13, 16, 17, 21] on PETs are underway to secure outsourcing computation with HE, they are still far away from the rapid development of practical privacy-preserving IoT systems. While many HE tools and compilers are well developed and adopted in numerous research works, most of them were written in low-level languages like C++, posing a high entry barrier to newbie IoT developers [21]. Even though they desire to consolidate their IoT solutions with a HE technique, they may need some truly bespoke tools that provide simplicity, flexibility, and abstraction.

This paper focuses on enhancing Node-RED, a low-code IoT platform for IoT developers [10], to perform HE algorithms for secure computation in IoT data. With Node-RED, the IoT developers can build their desired applications without the coding background. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AINTEC'22, December 19–21, 2022, Hiroshima, Japan
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9981-4/22/12...\$15.00
<https://doi.org/10.1145/3570748.3570752>

pre-defined pieces of code built in Node.js are wrapped in a Node-RED node to perform a specific function. The developers merely wire a collection of Node-RED nodes to perform their desired IoT solutions.

Hence, we develop *CipherFlow*, a HE playground in Node-RED for IoT developers to play and develop privacy-preserving solutions for IoT. The HE nodes in *CipherFlow* are built with *node-seal* [17], a wrapper of the Microsoft SEAL library [13]. Therefore, the *CipherFlow* does not require pre-installation of the Microsoft SEAL library. Written in Node.js, *node-seal* is seamlessly integrated into the Node-RED platform, allowing us to develop *CipherFlow*, the extension nodes to Node-RED. *CipherFlow* offers three pipeline abstractions, i.e., 1) a playground of dragging-and-dropping HE nodes in Node-RED, 2) arithmetic operators, and 3) optimal parameters setting guidelines. As a result, IoT developers can create and verify their secured computing functions on their dataflow. However, our development did not emerge overnight. To achieve design goals, i.e., simplicity, flexibility, and abstraction, developing the *CipherFlow* has to overcome the following challenges.

HE computation. The Node-RED platform does not directly support four requirements of the HE computation; (i) to unify the HE parameter set in the same Node-RED's dataflow, (ii) to import and export keys in/out of the Node-RED platform, (iii) parallel computation on arithmetic operations, and (iv) the garbage collector (GC) for optimal memory management. To fill these gaps, we carefully design the *CipherFlow* to meet these requirements (see in § 3).

Parameter setting. Despite the available dozen choices of HE schemes, the *CipherFlow* is based on the Cheon-Kim-Kim-Song (CKKS) scheme [5], which requires three main parameters to set up the HE environment: 1) polynomial modulus degree, 2) coefficient modulus and 3) scale factor. These parameters are used to determine the percentage error of ciphertext resulting from the compute operations. Although choosing the optimal CKKS parameters requires the skill of the cryptography experts [7], which is beyond the scope of this paper, we provide a hint to the *CipherFlow*'s users by presenting an indicator—called “Chain-index” value. We demonstrate how to use this indicator to fine-tune the optimal setting in § 4.

Reciprocal operation. The CKKS scheme does not support the reciprocal operation (or multiplicative inverse), a common arithmetic operator. To this end, we propose a reciprocal algorithm by modifying the Goldschmidt division iteration method [9] in § 3.5. However, it results in inherent errors due to its approximation. To deal with this shortcoming, we investigate the error bounds and provide the parameter-setting guidelines to the users (§ 5).

Contributions

The design and implementation of *CipherFlow* are proposed. Then we demonstrate the applicability and effectiveness of the *CipherFlow* through the performance evaluation in § 4 and § 5. Our evaluation shows the trade-off between the security and the performance in terms of time taken and memory consumption of *CipherFlow*.

2 PRELIMINARIES AND RELATED WORK

2.1 Node-RED

Since the third-party developers can easily extend the functionality of the platform through publishing nodes and configurations of

these nodes by being built in Node.js, several works paid attention to extending the applicability of Node-RED in different directions, e.g., [1, 8, 14, 18, 19]. However, only a few approaches, i.e., [14, 19], proposed functionalities to enhance privacy-preserving dataflows in Node-RED. RedCASTLE [19] provides k_s -anonymity in Node-RED for anonymizing the personal information in IoT streaming data. This approach cannot prevent third parties from learning the data while computing them in their environment. On the other hand, Kawamoto [14] has adopted the PASTE [20] in Node-RED to prevent the third party from learning the IoT data during data aggregation by the differential privacy and Paillier cryptosystem. With Paillier, his Node-RED extension supports only an add operation on an integer. In contrast, we provide comprehensive arithmetic operations and support computing over real numbers based on the CKKS [5] scheme. To the best of our knowledge, we propose the first fully homomorphic encryption to Node-RED.

2.2 Homomorphic Encryption

The state-of-the-art approaches focus on providing open-source software libraries, e.g., the Microsoft SEAL [13], PALISADE [6], HELib [11], for variants of HE schemes, e.g., BFV [3], BGV [2] and CKKS [5]. These open-source software libraries allow any developers freely adopt HE schemes to their applications. However, developing HE applications remains difficult due to the need for low-level programming language and cryptography knowledge. Several general-purpose compilers have been proposed in different languages, i.e., Julia, Haskell, C++, Python [7, 12], Node.js [17] (see detail in [21]) to alleviate this difficulty. The HE compilers are aimed to provide a high-level abstraction for developing HE-based applications. These tools manage key setup, encryption, decryption, and ciphertext operation in the background. Even though these programming languages can be written comfortably by most typical programmers, they are not easily learned and, hence, inconvenient for typical IoT developers. On the one hand, EVA [7] and PyFHE [12] have been designed for Python-based machine learning (ML) applications because most ML developers are familiar with Python coding. On the other hand, IoT developers are more familiar with Node-RED, which does not require hard-cored program writing skills. Therefore, this paper proposes *CipherFlow*, a HE playground for IoT developers. It offers drag-and-drop features that make users conveniently select HE nodes and define dataflows in the Node-RED platform to perform a task.

2.3 CKKS with the Microsoft SEAL

We develop the *CipherFlow* based on the CKKS scheme under the Microsoft SEAL library [13]. The CKKS scheme composes of the following stages;

1. Parameter setting. CKKS requires three parameters: (i) polynomial ring of degree n , (ii) the coefficients modulus chain $\mathbf{q} = [q_1, \dots, q_L]$, where $q_l \in \mathbf{q}$ is a coefficients modulus level l for $1 \leq l \leq L$ and L is the highest level, and (iii) scale factor Δ . These parameters are encoded in a context. This context is further used to create the HE environment.

2. Key generation. The KeyGenerator takes the context to create the HE environment and construct a tuple of keys, i.e., (a public key pk , a secret key sk , a relinear key rk).

3. Encryption. The Encryptor takes context to generate the HE environment and the public key pk . The idea of CKKS scheme is to add noise for security purposes as a part of the message we want to encrypt— so-called Ring-Learning with Errors (Ring-LWE). The encoder transforms a message m to a plaintext P , in which a real number is represented in the fixed-point number scaled by the scale factor Δ . Hence, noise is added to the plaintext due to approximation. Once the Encryptor receives the plaintext P as an input, it applies pk to encrypt P to a ciphertext c . A ciphertext c is in the form of polynomials in which the coefficient modulus bounds the ciphertext space.

4. Evaluation. The Evaluator takes the context to generate the HE environment. It supports arithmetic operations. CKKS allows the ciphertext to do arithmetic operations on both ciphertext and plaintext. Once the Evaluator takes two ciphertexts, c_1 and c_2 to do the operation (e.g., add $\oplus$ and multiply $\otimes$), it returns the result as a ciphertext, $c' = c_1 \oplus c_2$, and $c'' = c_1 \otimes c_2$.

5. MODSWITCH. MODSWITCH is used to switch the coefficient modulus q_l to q_{l-1} .

6. RESCALE. RESCALE is the heart of CKKS, which is used to reduce the magnitude of the plaintext. After multiplication with two ciphertexts, the RESCALE operation is needed. For instance, two ciphertexts with scale 2^Δ will grow to $2^\Delta * 2^\Delta = 2^{2\Delta}$ after multiplication. CKKS uses the RESCALE operation to scale down the ciphertext with scale $2^{2\Delta}$ by dividing it by 2^Δ . The RESCALE also switches the coefficient modulus from q_l to q_{l-1} .

7. SetScale. setScale is used to force the scale to the exact value.

8. Relinearize. Relinearize reshapes the ciphertext without changing the underlying message by applying key rk . This operation helps limit noise growth and the ciphertext size.

9. Decryption. Decryptor takes the context to generate the HE environment. The secret key sk is used to decrypt the ciphertext and yields the plaintext. The decoder transforms the plaintext back to the approximate real number corresponding to the ciphertext.

3 PROPOSED METHOD

3.1 Design Principles

We aim to develop the *CipherFlow* to be a playground for IoT developers to play and learn how to apply the HE functionalities to preserve the privacy of IoT applications. The *CipherFlow* is designed to meet these desired goals: simplicity, flexibility, and abstraction. To simplify a way to implement the HE scheme in IoT applications, we develop the *CipherFlow* as the node extension in the Node-RED pallet and preserve the HE functionalities with the Microsoft SEAL [13] in the backend library. On top of that, node-seal wraps the Microsoft SEAL library, allowing us to develop and integrate the HE node extension to Node-RED. Therefore, the IoT developer can install the *CipherFlow* as the Node-RED extension (see in § 4).

With *CipherFlow*, the developers can easily drag and drop the HE nodes and flexibly compose the computation functions in the Node-RED platform. Moreover, the *CipherFlow* allows developers to create a new tuple of HE keys or download the existing ones to the flows. The *CipherFlow* abstracts the HE nodes for some CKKS functions, i.e., encode, decode, RESCALE, setScale and Relinearize, that are required for encryption, decryption and evaluation. By this

abstraction, the developers do not need to determine when and how to use these functions wisely.

3.2 CipherFlow Architecture

Regarding the design principles, the *CipherFlow* architecture comprises three layers: (i) backend library, (ii) C++ abstraction, and (iii) Node-RED extension as shown in Figure 1.

Backend library. The *CipherFlow* uses the Microsoft SEAL version 4.0.0 as the backend library written in C++. We select the CKKS scheme in this library as the core of our node extension. Hence, the objects and core functions of CKKS are embedded in our backend library.

C++ abstraction. To abstract C++, the node-seal wraps CKKS core functions and objects in Node.js. With Web Assembly, node-seal is a portable target for C++ in the Microsoft SEAL library, enabling faster deployment on Node-RED. The node-seal also provides memory management and only contains a low-level API close to C++.

Node-RED extension. The *CipherFlow* presents the CKKS functionalities and arithmetic operations in the Node-RED extension that raises the level of abstraction for putting HE functionalities in the dataflows. Nodes warp Node.js code and perform individual operations. The developers merely select HE nodes by dragging and dropping them on the Node-RED's workspace. Underneath this node extension, the *CipherFlow* provides the options to import and export the keys and context to/from the Node-RED platform. Beyond the CKKS arithmetic operations in node-seal, we develop the reciprocal operation to the node extension.

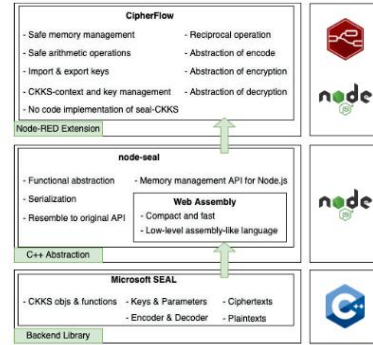


Figure 1: CipherFlow architecture.

3.3 HE Computation

The main idea behind the *CipherFlow*'s core abstractions is to unify and simplify the HE computation in Node-RED. However, the original Node-RED platform does not support the following requirements;

Unify parameter setting. Nodes are the event triggers running their functions. Each node does not share local variables with the other nodes in a dataflow, while global variables are shared among

different flows in the same workspace. This yields the mismatch of *CipherFlow*'s principles, in which simplicity in setting and flexibility in creating multiple dataflows in the same workspace cannot be met. To unify the parameter setting in the same flow without sharing it with the other flows, we warp it in a context and pass the context as a message to the nodes along the same dataflow.

Import and export keys. Common nodes in Node-RED do not provide the import and export data nor configurations to a save file. To make the *CipherFlow* flexible and portable, our HE functionality nodes support import and export keys and context in/out of the Node-RED platform as a text file.

Parallel computation. A node receives and sends events, and the configuration of this event handling defines up to one input port but any number of output ports. This allows us to pass a message to the parallel arithmetic operation nodes. For example, we want to compute $c_1 + 3$ and $c_1 * 2$ in the parallel flows. However, the message from one node cannot flow to two nodes simultaneously due to the node dependency and asynchronous computation of Node-RED events. Once a message (e.g., ciphertext c_1) arrives at an arithmetic operation node ($\text{add}(P)$), the other arithmetic operation node ($\text{multi}(P)$) needs to wait until the node ($\text{add}(P)$) has finished computing the message c_1 . As a consequence, the message c_1 was replaced by $c_1 + 3$ leading to the wrong result $((c_1 + 3) * 2$ instead of $c_1 * 2$). To solve this problem, we duplicate c_1 according to the output ports and inject them into the parallel flows.

Garbage collector. The modern language run time systems automatically manage an application's memory with garbage collection (GC). However, node-seal, which wraps C++ by Web Assembly, does not provide GC for optimal memory management. To avoid memory leaks and dangling pointers, we use manual memory management like C++/C via command ".delete" to free the unused memory of the node-seal instance.

3.4 Parameter Setting

The heart of HE computation is to choose the parameter setting optimally. First, we explore how to select the CKKS parameters matching the security level and then provide a guideline for parameter setting to avoid pitfalls in working with CKKS.

Let us consider the CKKS parameters: (i) polynomial modulus degree n , (ii) the coefficients modulus chain $\mathbf{q} = [q_1, \dots, q_L]$, where q_l is a coefficients modulus level l for $1 \leq l \leq L$, and (iii) scale factor Δ . The freshly encrypted ciphertext is said to be the level L -ciphertext. Let $Q = \prod_{l=1}^L q_l$ be the coefficient modulus product. The polynomial modulus degree n determines the security level of encryption [5] and determines the maximum coefficient modulus Ω . For instance, to guarantee 128 bits of security (TC128 [17]) for $n = 2^{13} = 8192$, $\Omega = 2^{218}$, while for $n = 2^{14} = 16384$, $\Omega = 2^{438}$. The maximum coefficient modulus Ω limits the depth of the ciphertext level in which $Q \leq \Omega$ must be satisfied. The scale factor Δ is used to map a real number to an integer by encoding the message to plaintext. Let the scale of level L -ciphertext $c_{i,L} = 2^\Delta$, where scale factor Δ is set. All freshly encrypted ciphertexts must have the same q_L , but they may have different scale factors Δ . Computing CKKS arithmetic operations have to satisfy the following constraints.

Constraint 1: q_l must be equal for two input ciphertexts for additive and multiplicative operations (i.e., $\text{add}(P)$, $\text{add}(E)$, $\text{multi}(P)$,

and $\text{multi}(E)$ (see their definitions in Table 1)).

Constraint 2: The two input ciphertexts must have the same scale for additive operations, i.e., $\text{add}(P)$ and $\text{add}(E)$.

According to these constraints, the *CipherFlow* enforces all freshly encrypted ciphertexts to be set to the same scale factor to avoid failure due to improperly set. After we apply $\text{add}(E)$ operation to two ciphertexts, it returns the same scale, while $\text{multi}(E)$ operation increases the scale. Given freshly encrypted ciphertexts with the same scale, it turns out that after we multiply two ciphertexts, we cannot do $\text{add}(E)$ with that of the result and the freshly encrypted ciphertext because it does not satisfy the **Constraint 2**. To solve this problem, we use **RESCALE** operation to scale down the scale of the ciphertext with scale factor Δ . However, by applying the **RESCALE** operation, the coefficient modulus of the ciphertext is also reduced. This causes the coefficient modulus of the multiplicative result not to match the coefficient modulus of freshly encrypted ciphertexts. Therefore, we cannot do $\text{add}(E)$ operation of these two ciphertexts since it brakes the **Constraint 1**. We, therefore, consider applying **MODSWITCH** operation to switch the coefficient modulus q_l to q_{l-1} , so that the **Constraint 1** is satisfied. Although after **MODSWITCH** is applied, the scale is still mismatched because it is divided by a prime number very close to the scale we want, but it is not exactly equal. Therefore, we enforce the scale by applying **setScale** to the exact value. In addition, after multiplication, the size of ciphertext is expanded from 2 to 3 polynomials (see detail in [5]). To reduce the size of ciphertext back to 2 polynomials, we apply **Relinearize** with the relinearize key rk to the ciphertext.

Hence, we warp the operations **Relinearize**, **RESCALE**, and **setScale** into the $\text{multi}(P)$, and $\text{multi}(E)$ operations for abstraction. In addition, we let $\text{add}(P)$ and $\text{add}(E)$ nodes check the level of coefficient modulus. If two ciphertexts are unequal, we use **MODSWITCH** to switch their coefficient modulus to the same level. Therefore, the developers do not need to apply these operations manually to avoid pitfalls. Since the $\text{multi}(P)$, and $\text{multi}(E)$ operations move the ciphertext and the coefficient modulus to the next level, i.e., from level l to $l - 1$. Once the ciphertext is at level 2, it cannot further do the $\text{multi}(P)$, and $\text{multi}(E)$ operations. Note that q_1 is preserved for setting the scale back to the same value as the fresh one. Consequently, we recommend to set $q_L = q_1$ and $q_L, q_1 > 2^\Delta$. We also recommend setting $q_l = 2^\Delta$ for $1 < l < L - 1$ to avoid error by approximation. To simplify the way to insert the parameter setting of the coefficient modulus chain $\mathbf{q}$ in the form of power of 2, we define $\tilde{\mathbf{q}}$ a degree of $\mathbf{q}$ and let the developer insert $\tilde{\mathbf{q}} = [\tilde{q}_1, \dots, \tilde{q}_L]$, where $\tilde{q}_l$ is the degree of q_l . Hence, the developers can define $\tilde{q}_1, \tilde{q}_L = \Delta + i$, where i is a positive integer. We introduce a *Chain-index* to warn a developer whether it can do more multiplication operations. Let us define $\lambda(c_{i,l}) = l - 2$ as a *Chain-index* of ciphertext i at level l . The *Chain-index* is shown under all Node-RED nodes that contain a ciphertext.

3.5 Reciprocal Algorithm for CipherFlow

The Microsoft SEAL provides additive and multiplicative operations but does not provide the reciprocal operation or the multiplicative inverse ($1/x$), a common arithmetic operation. We, therefore, develop the reciprocal operation for the *CipherFlow*. However, computing the reciprocal operation ($1/x$) in HE is not straightforward

since division in HE is not applicable. Thus, we explore two approximation methods, i.e., 1) Newton-Raphson and 2) Goldshmidt, that can perform $1/x$ without division. The Newton-Raphson and Goldshmidt methods both are iterative and quadratically convergent, allowing us to obtain the approximation of $1/x$. However, the Newton-Raphson method requires guessing an initial value, and a wrong initial value can result in the wrong convergence or requires more iterations to converge.

Therefore, we select the Goldshmidt iteration [4] method for approximating $1/x$ because the division and an initial value are not required. The Goldshmidt iteration requires only the number of iterations d in which the error rate of the output of the algorithm compared to $1/x$ is bounded by $(1-x)^{2^{d+1}}$ (see Lemma 1 in [4]). Since the number of iterations is specified by d , we can determine the required coefficient modulus chain q . Although the Goldshmidt iteration method is suitable for performing the reciprocal operation ($1/x$) in HE, it limits the working space on $x \in (0, 2)$. Thus, we generalize the Goldshmidt iteration algorithm to be applied for the other space $x \in (0, \Psi)$ for $\Psi > 0$.

Let a linear transformation $L_{\Psi \rightarrow \Gamma} : x \in (0, \Psi) \rightarrow \tilde{x} \in (0, \Gamma)$ map from a positive real number in range $(0, \Psi)$ to a positive real number in range $(0, \Gamma)$, where Ψ and Γ are the maximum values of the two considered spaces. Accordingly, we denote $L_{\Psi \rightarrow \Gamma} : x \in (0, \Psi) \rightarrow \tilde{x} \in (0, \Gamma)$, $L_{\Psi \rightarrow \Gamma}(x) = \frac{\tilde{x}}{\Psi} \times \Gamma$.

For $x_1, x_2 \in (0, \Psi)$, the output of a linear transform $L_{\Psi \rightarrow \Gamma}(x_1)$ divided by $L_{\Psi \rightarrow \Gamma}(x_2)$ is equal to x_1/x_2 , where Ψ and Γ are positive integers. We, then, propose the reciprocal algorithm (as shown in Algorithm 1) for the CKKS scheme by integrating the linear transformation $L_{\Psi \rightarrow \Gamma}$, where $\Gamma = 2$ in the Goldshmidt iteration algorithm in [4].

Algorithm 1 requires predetermining the number of iterations d and the range of space we want to map to Ψ . It is worth noticing that larger d yields less error in the approximation. However, approximation precision has to be traded with the number of multiplications, causing a larger *Chain-index*. To compromise between the large *Chain-index* and its precision, we recommend to set $d = 3$. We demonstrate this trade-off in § 5.

Algorithm 1 Reciprocal(x, d, Ψ)

```

1: input:  $0 < x < \Psi, d > 0, \Psi > 0$ 
2: output:  $a_d$ , an approximate value of  $1/x$ 
3:  $\tilde{x} \leftarrow L_{\Psi \rightarrow 2}(x)$ 
4:  $k \leftarrow L_{\Psi \rightarrow 2}(1)$ 
5:  $\hat{x} \leftarrow k/\tilde{x}$ 
6:  $a_0 \leftarrow 2 - \hat{x}$ 
7:  $b_0 \leftarrow 1 - \hat{x}$ 
8: for  $n = 1, 2, \dots, d - 1$  do
9:    $b_{n+1} \leftarrow b_n^2$ 
10:   $a_{n+1} \leftarrow a_n \cdot (1 + b_{n+1})$ 
11: end for

```

3.6 Node Pallet for CipherFlow

The *CipherFlow* introduces 13 new nodes as the Node-RED extension as presented in Table 1. We divide them into two groups: 1) HE functionalities and 2) arithmetic operations. The HE functionality

Table 1: Node-RED CipherFlow Pallet and Definitions

Operation	Node	Definition
FHE	encrypt	To encrypt the plaintext messages passing through it by applying with public key pk and output ciphertexts.
	decrypt	To decrypt ciphertexts passing through it by applying secret key sk , and output the message in a float number
	setContext	To insert the CKKS context to the arithmetic operation nodes in the same dataflow.
	input	Similar to the encrypt node, but it encrypts the value in the input field instead of the passing message
	toBase64	To transform a ciphertext to base64URL
Arithmetic	toCiphertext	To transform a base64 to a ciphertext
	add(E)	To add two passing ciphertexts
	add(P)	To add a given value in the input field and the passing ciphertext
	multi(E)	To multiply two passing ciphertexts
	multi(P)	To multiply a given value to the passing ciphertext
	negate	To convert a positive passing ciphertext to negative ciphertext, and vice versa
	exponent	To raise the passing ciphertext to a given power
	reciprocal	To multiplicative inverse a passing ciphertext

nodes provide all necessary CKKS functions, such as encryption, decryption, and setContext. In a playground, developers may focus on only testing their computation flow rather than testing the fed input messages from outside Node-RED. Thus, we provide the input node that embeds the fixed value we want to encrypt and allows the developers to create or select the existing context. The arithmetic operation nodes contain additive and multiplicative operations, such as add(P), add(E), multi(P), and multi(E). Instead of subtraction, the *CipherFlow* offers the negate node to turn a positive value into a negative one, and vice versa. Consequently, we can perform subtraction by combining the negate node and the add(P) or add(E) nodes. The exponent node resembles the multi(E) with the number of exponents. The reciprocal node performs the multiplicative inverse of a ciphertext by Algorithm 1 with fixed $d = 3$.

4 USING CIPHERFLOW

We demonstrate how to use *CipherFlow* through several use cases.

4.1 Getting Start

To start using *CipherFlow*, we first install the *CipherFlow* extension to Node-RED¹. After the installation succeeded, the *CipherFlow* nodes are listed in the node pallet. In this playground, we can play and learn how the HE is performed via the node extension.

In the *CipherFlow*, developers must generate the CKKS context and keys corresponding to the parameter setting or import them from the prior files. All nodes must be set to the same CKKS context along the same dataflow and tied with the corresponding keys. In the parameter setting stage, *CipherFlow* allows a developer to configure the following parameters;

context: to set the CKKS parameters: 1) *PolyModulus* for a polynomial modulus degree n , 2) *CoeffModulus* for coefficient modulus chain $\bar{q}$, and 3) *Scale* for a scale factor Δ .

publicKey: to be selected in the input and encrypt nodes.

secretKey: to be selected in only the decrypt node.

relinKey: to be selected in the input and setContext nodes.

We omit the parameter configuration in the arithmetic operation nodes. The developers merely wire the input to those arithmetic operation nodes, and the corresponding CKKS context obtained from the previous node is automatically injected. The *CipherFlow* requires the developer to set a constant value (referring to a plaintext) for computation in *add(P)*, and *multi(P)*.

4.2 Playground

The playground is designed for the IoT developers to play with the *CipherFlow*. The goal is to develop and test their computation functions. Specifically, input node is designed for this purpose since it does not require real IoT data to feed into the node. We demonstrate how to compose the computation functions in various examples as follows.

Linear function. Figure 2a presents function $f(x) = 3x - 2$. Let *Chain-index* be equal to 7 at the fresh ciphertext. The *Chain-index* value is reduced to 6 after passing the *multi(P)* node. The *Chain-index* value under each node is a hint that indicates the developer to fine-tune the performance trade-off between performance and security.

Polynomial functions. Figure 2b presents function $f(x) = x^3 + 0.4x + 1$. We use the *exponent* node to compute x^3 , where 3 is set to the exponent value. Since the *exponent* node requires multiplication two times, the *Chain-index* value is decreased from 7 to 5. To perform function $f(x)$, the input node of x injects to the exponent node and the *multi(P)* simultaneously. Note that the *Chain-index* value from the exponent node is not equal to that of the *multi(P)*. When the output ciphertexts of these nodes arrive the *add(E)* node, the *Chain-index* value is forced to drop to the lowest one.

Divider function. Figure 2c presents function $f(x, y) = \frac{x}{y+45}$. It is worth noticing that the *Chain-index* value is equal to zero at the *multi(E)* node. This indicates that we cannot further add more multiplicative or reciprocal nodes to this dataflow.

4.3 Edge computing

A dataflow in the edge computing is divided into two parts: the dataflow of encryption, and the dataflow of decryption.

Encryption. In Figure 3, the *injectArray* node receives the messages sent from sensors to be encrypted in the *encrypt* node. The *encrypt* node outputs the ciphertext objects corresponding to the input messages and later are converted to base64URL by the *toBase64* node before being sent to the cloud computing by the *mqtt-out* node.

Decryption. Figure 4 presents a decryption dataflow. the *mqtt-in* node receives the ciphertexts resulted by the cloud computing. The ciphertext is injected to the *decrypt* node, which outputs the real value of the ciphertext.

4.4 Cloud computing

The dataflow in cloud computing consists of the middle part of the dataflow in the playground. The *mqtt-in* node receives the Base64URL messages from the edge computing and passes them to the *toCipherText* node to convert them back to the ciphertext objects. The *setContext* node inserts the CKKS context to the dataflow, so that all the arithmetic operation nodes can perform the same HE environment. Note that the sequence of the arithmetic operation nodes is the same as in the playground. The ciphertext objects are converted to base64URL by the *toBase64* node before being sent back to edge computing. We demonstrate the examples of cloud computing according to the examples in the playground.

Linear functions. Figure 5a presents function $f(x) = 3x - 2$.

Polynomial functions. Figure 5b presents function $f(x) = x^3 + 0.4 * x + 1$.

Divider function. Figure 5c presents function $f(x, y) = \frac{x}{y+40}$.

5 EVALUATIONS

We evaluate the performance of *CipherFlow* in terms of time taken and memory consumed. We run the *CipherFlow* in a Node-RED docker on our tested machine, and then we evaluate the applicability of the proposed reciprocal algorithm. Our experiments were run on MacBook Air (M1, 2020), Mac Os Monterey 12.3.1, and Ram 8 GB.

First, we evaluate the time taken and memory consumed by the *CipherFlow* operations. We set the *Chain-index* equal to 6 because the reciprocal computation requires at least six multiplications. The CKKS parameters are set according to two scenarios;

Ex#1: $n = 2^{13}$, $\bar{q} = [40, 30, 30, 30, 30, 30, 30, 30]$, and $\Delta = 30$, and

Ex#2: $n = 2^{14}$, $\bar{q} = [60, 40, 40, 40, 40, 40, 40, 40]$, and $\Delta = 40$. This

evaluation aims to demonstrate the performance and security trade-off. In Ex#1, we set the security level to null, which means the security condition is broken due to the coefficient modulus production $\prod_{l=1}^L q_l > 2^{2^{18}}$ for $n = 2^{13}$. Note that $q_l = 2^{\bar{q}_l}$, where $\bar{q}_l \in \bar{q}$. In contrast, in Ex#2, our parameter setting satisfies the security level of TC128 in which $\prod_{l=1}^L q_l \leq 2^{438}$ for $n = 2^{14}$. We observe that all parameters in Ex#2 consume memory, and run time more than twice that of Ex#1 as shown in Table 2. The reason is that the scale factor expands the ciphertext Δ and the polynomial modulus degree n . The plaintext space is reserved by the scale factor Δ , while the ciphertext space is reserved by the polynomial modulus degree n .

Next, we evaluate our reciprocal algorithm's applicability by observing the algorithm's percentage error when the CKKS scheme is implemented and when the unencrypted value is implemented. Percentage error presents the difference between the estimated

¹<https://github.com/donkyduck/CipherFlow>.

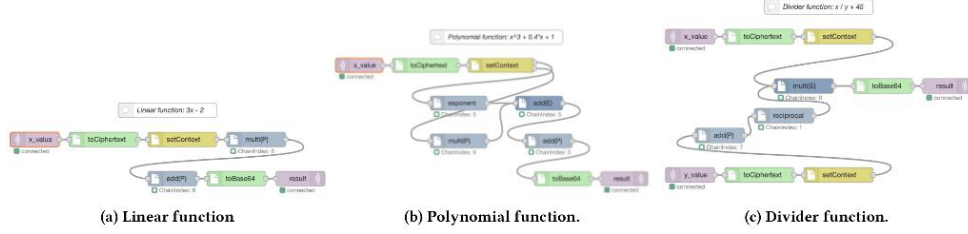


Figure 5: Cloud computing

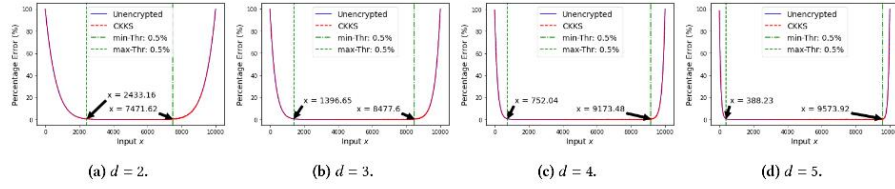


Figure 6: The percentage error of the reciprocal algorithm for CKKS: $n = 2^{14}$, $\tilde{q} = [40, 35, 35, 35, 35, 35, 35, 35, 35, 40]$, $\Lambda = 35$.

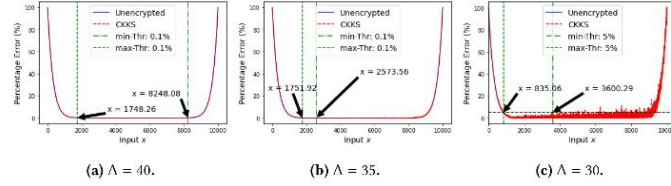


Figure 7: The percentage error of the reciprocal algorithm with $d = 3$, where (a) CKKS: $n = 2^{14}$, $\tilde{q} = [60, 40, 40, 40, 40, 40, 40, 60]$, and (b) CKKS: $n = 2^{14}$, $\tilde{q} = [60, 35, 35, 35, 35, 35, 35, 60]$, and (c) CKKS: $n = 2^{14}$, $\tilde{q} = [60, 30, 30, 30, 30, 30, 30, 60]$.

- Advances in Cryptology – ASIACRYPT 2019*, 415–445.
- [5] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, Tsuyoshi Takagi and Thomas Peyrin (Eds.), Springer International Publishing, Cham, 409–437.
 - [6] Dave Cousins, Kurt Rohloff, and Yuriy Polyakov. 2014. *PALISADE Homomorphic Encryption Software Library: An Open-Source Lattice Crypto Software Library*. Retrieved August 20, 2022 from <https://palisade-crypto.org/>
 - [7] Roshan Dathathri and et al. 2020. EVA: An Encrypted Vector Arithmetic Language and Compiler for Efficient Homomorphic Computation. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, 546–561.
 - [8] João Pedro Dia, André Restivo, and Hugo Sereno Ferreira. 2021. Empowering Visual Internet-of-Things Mashups with Self-Healing Capabilities. In *IEEE/ACM SERP4IoT*, 1–8.
 - [9] Milos Ercegovac and et al. 1999. *Improving Goldschmidt Division, Square Root and Square Root Reciprocal*. Research Report inria-00072909, INRIA, LIP.
 - [10] J. Foundation. March 2016. *Node-red, flow-based programming for the internet of things*. Retrieved June, 2022 from <https://nodered.org/>
 - [11] Shai Halevi and Victor Shoup. 2014. *HElib Implementing Homomorphic Encryption*. Retrieved August 20, 2022 from <https://homenc.github.io/HElib/index.html>
 - [12] Alberto Ibarondo and Alexander Viand. 2021. *Pyfhel: Python For Homomorphic Encryption Libraries*. In *WAHC '21*, 11–16.
 - [13] Wernsing John and et al. March 2018. *Microsoft SEAL*. Retrieved March 2, 2022 from <https://github.com/microsoft/SEAL>
 - [14] Junpei Kawamoto. 2016. An Implementation of Privacy Preserving Stream Integration System. In *AIChallenger'19*, 32–38.
 - [15] Chao Li and Balaji Palanisamy. 2019. Privacy in Internet of Things: from Principles to Technologies. *IEEE Internet of Things* 6, 1 (February 2019), 488–505. <https://doi.org/10.1109/JIOT.2018.2864168>
 - [16] Deepika Natarajan and Wei Dai. 2021. SEAL-Embedded: A Homomorphic Encryption Library for the Internet of Things. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021, 3 (Jan. 2021), 756–779.
 - [17] Angelou Nick. March 2020. *Morfis: Homomorphic Encryption as a Service*. Retrieved March 2, 2022 from <https://github.com/morfis-io>
 - [18] Joseph Noor and et al. 2019. DDFlow: visualized declarative programming for heterogeneous IoT networks. In *IoT'19*, 172–177.
 - [19] Frank Pallas, Julian Legler, Niklas Amsgruber, and Elia Grönewald. 2021. Red-CASTLE: practically applicable ks-anonymity for IoT streaming data at the edge in node-RFD. In *M4IoT '21*, 8–13.
 - [20] Vibhor Rastogi and Suman Nath. 2010. Differentially private aggregation of distributed time-series with transformation and encryption. In *ACM SIGMOD*, 735–746.
 - [21] Alexander Viand, Patrick Jattke, and Anwar Hithnawi. 2021. SoK: Fully Homomorphic Encryption Compilers. In *the 1st ACM SIGSOFT International Workshop on Ensemble-Based Software Engineering*, 1–8.

5G Use Cases for Thailand's Smart Factory/Manufacturing

Chatchamon
Phoojaroenchanachai
NECTEC
Pathum Thani, Thailand
chatchamon.pho@necr.nstda.or.th

Kamol Kaemarungsi
NECTEC
Pathum Thani, Thailand
kamol.kaemarungsi@necotec.or.th

Panita Pongpaibool
NECTEC
Pathum Thani, Thailand
panita.pongpaibool@necotec.or.th

ABSTRACT

Utilizing a 5G cellular network can help factories digitally transform their traditional processes, increasing operation efficiency, and aligning with the Industry 4.0. The 5G network is chosen over other communication technologies due to its broader service coverage, guarantee quality of services, reduced interference from other wireless signals, and enhanced security through SIM card authentication. In this project, four practical industry use cases utilizing 5G networks are presented. The use cases are smart warehousing, asset tracking in production line, mobile robot teleoperating, and condition-based monitoring. All case locations have 5G network services at band n41 with sufficient quality and coverage. The performance of each 5G network is also assessed, including data rate, latency, maximum bandwidth, and error rate. All examples benefit from 5G use case integration. The smart warehouse and asset tracking cases gain real-time location data and insights from workload analysis. The mobile robot case can help factory staff to remotely control payload transportation process, while the monitoring case transforms its manual inspection approach to real-time visualized logs.

CCS CONCEPTS

• Networks • Network types • Mobile networks

KEYWORDS

5G, Industry 4.0

ACM Reference format:

Chatchamon Phoojaroenchanachai, Kamol Kaemarungsi and Panita Pongpaibool. 2023. 5G Use Cases for Thailand's Smart Factory/Manufacturing. In *Proceedings of the 18th Asian Internet Engineering Conference (AINETC'23)*, December 12–15, 2023, AINETC, Hanoi, Vietnam, 2 pages. <https://doi.org/10.1145/1234567890>

1 Experiment Method

Before implementing each use case, the 5G network's radio quality at each site was measured to ensure network stability and sufficient coverage, and to identify performance limitations. To assess signal quality and coverage, we used Keysight FieldFox microwave analyzer N9950B [1] and a directional antenna [2]. The FieldFox can measure the 5G signal in two functions: Real Time Spectrum Analysis (RTSA) and Over-the-Air (OTA). The RTSA function captures a 5G signal in frequency domain, and the OTA

function returns 5G signal quality parameters, including Reference Signal Received Power (RSRP) in dBm, Reference Signal Received Quality (RSRQ) in dB, and Signal to Interference and Noise Ratio (SINR) in dB.

We used a laptop and 5G customer premises equipment (CPE) to test network performance metrics, including data rate, latency, maximum bandwidth, and rate of dropped packets in transmission. Data rate and latency were tested with nPerf (URL: <https://www.nperf.com/th/>), speed test command-line, and ping utility. To estimate the maximum bandwidth and packet loss rate, we used the iPerf3 network diagnostic tool [3]. The iPerf on TCP protocols can measure the maximum achievable bandwidth on IP networks, while the iPerf on UDP can measure packet loss.

With the verified 5G network, we deployed each use case with required application. The smart warehousing and asset tracking cases utilized an implementation of our ultra-wideband (UWB) based real-time location system (RTLS) and workload analysis unit. The UWB RTLS system can track a moving machine (such as forklift or an automated guided vehicle (AGV)) by attaching a UWB tag that can calculate its location based on the reference locations of surrounding installed UWB anchors. The UWB anchors were installed on the walls or structures of the warehouse and factory over the tracking area. The UWB tag determines its location using trilateration based on time of flight (ToF) that is measured over UWB pulses. The UWB tag transmits its location and additional information through 5G CPE to the cloud-based server. Staff can monitor the machine's location at the URL: <https://rtls.lailab.online>. Subsequently, data on the server is used for analyzing workload efficiency. The analyzed metrics include cumulative machine distance, machine operation time, and a timeline zone report that tracks zones visited by machines over time. These metrics are visualized in graphs and can be viewed over a 30-minute interval, daily, weekly, monthly, or cumulatively.

For the case of mobile robot teleoperating, the robot was remotely controlled over the 5G network using a joystick by staff in the control room. Concurrently, real-time video streaming from an IP camera mounted on the robot offered visual assistance to staff. For the last use case of condition-based monitoring, an IoT monitoring system using RADAR-based sensors was deployed. The radar sensors were installed at the top of the large tanks, approximately 11.4 meters above the ground. The sensor reads the liquid level in the storage tank and transmits data to an IoT gateway situated within a ground-level controller box. The gateway gathers information from two tanks alongside relevant information and

Table 1: 5G Radio Frequency Measurements at all use case locations

Test Topic	Smart Warehouse	Asset tracking	Mobile robot	Condition-based monitoring
5G signal quality - RSRP (dBm)	-95.82 Fair to Poor	-79.09 Excellent	-93.70 Fair to Poor	-91.93 Fair to Poor
RSRQ (dB)	-13.24 Good	-11.37 Good	-13.31 Good	-13.16 Good
SINR (dB)	6.54 Fair to Poor	21.53 Excellent	9.16 Fair to Poor	14.46 Good
Data Rate - DL (Mbps)	12.51	35.10	17.74	32.42
UL (Mbps)	7.63	21.88	1.79	2.93
Latency (ms)	57.52	24.35	44.18	38.42
Max Bandwidth (Mbps)	8.69	19.24	1.57	1.69
Packet loss rate in transmission (%error per target bit rate)	*36.8% /10	1.96% / 20	0% / 4	0.55% / 4

transmits it to NETPIE IoT cloud platform (URL: <https://netpie.io>). The data is transmitted, stored, and processed on the cloud in the encrypted form through a privacy-enhanced service called Cyblion. The log information is later presented on an interactive dashboard and can be accessed via smartphones or tablets with internet connectivity.

2 Results and Discussion

Utilizing the FieldFox RTSA function, all locations could receive 5G signal band n41 as shown in an example in Figure 1. The OTA results were analyzed according to the criteria of signal statistics in [4]. In Table 1, the tested OTA showed that most sites, except for the asset tracking location, received weak signal strength but maintained good signal quality (fair to poor RSRP and good RSRQ). Since none of our use cases involved extremely demanding applications, these signal strength levels were sufficient. The tested maximum bandwidths were closely aligned with the presented uplink speed at each location. For latency, both the mobile robot and warehouse cases had a slower response time due to weak SINR, and the warehouse was tested with a higher rate of dropped packets. The high error rate was because we tested at a target data rate that was too high for the supported uplink speed at the location. In a real deployment, the warehouse case will experience far fewer data packet losses. Additionally, apart from the warehouse, other locations relied on existing public 5G networks. These networks had to share their bandwidth resources with other consumer users, resulting in suboptimal performance of speed, bandwidth, latency, and loss rate. For example, the cellular networks at the mobile robot and condition-based monitoring locations could provide an uplink speed of up to 4 Mbps. However, neither location could achieve that high speed or bandwidth.

Integrating 5G cellular networks into factories can help digitally transform traditional processes as demonstrated in this project. The first two cases can improve production efficiency with data-driven insights such as location data and workload analysis. The third case switches payload transportation to teleoperation. The last case transforms its manual inspection method into automated real-time log visualization. The benefits of using 5G networks over other communication technologies are that the coverage of the 5G

network is generally wider; therefore, it could save on the cost of local network installation, especially on a factory campus with multiple buildings.

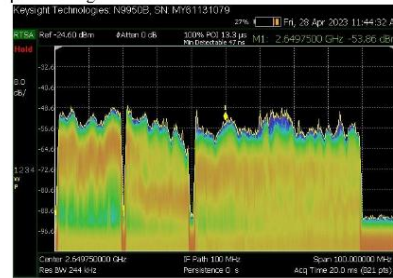


Figure 1: 5G Signal captured by RTSA at band n41

Compared with a Wi-Fi signal, a 5G signal is less susceptible to interference because only licensed devices can be used. Additionally, the 5G network provides security as it requires devices to use subscriber identity module (SIM) cards in the authentication process before accessing the network service. It is also possible to guarantee the network service level agreements (SLAs) by switching from a public 5G network to an enterprise-grade private 5G network. In the smart warehouse use case, we subscribed to 5G virtual private network type 1 provided by Advanced Info Service PLC. This specific enterprise-grade 5G service guarantees downlink speed of 20 Mbps and uplink speed of 8 Mbps with network slicing. The average latency is approximately 50 ms. The recovery time is within six hours.

REFERENCES

- [1] Keysight, "FieldFox handheld analyzers," [Online]. Available: <https://www.keysight.com/us/en/assets/7018-06516/data-sheets/>
- [2] Aaronia, "Logger antennas HYPFRI.ORG," [Online]. Available: https://downloads.aaronia.com/datasheets/antennas/HYPFRI.ORG/Aaronia_HYPFRI.ORG_30_Logger_Antennas.pdf
- [3] ESnet/iperf, "IPERF3: A TCP, UDP, and SCTP Network Bandwidth Measurement Tool," [Online]. Available: <https://github.com/esnet/iperf>
- [4] Netvault, "4G Signal Statistics Explained," [Online]. Available: <https://www.netvault.net.au/netmon-4g-signal-statistics-explained/>

Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0

Kalika Suksomboon
kalika.suk@nectec.or.th
NECTEC

Nataset Tanabodee
nataset.t@ku.th
Kasetsart University

Eakarat Saktawornlerd
eakarat.sak@ku.th
Kasetsart University

Aimaschana Niruntasukrat
aimaschana.niruntasukrat@nectec.or.th
NECTEC

Chaveee Issariyapat
i@chavee.com
NEXPIE Co. Ltd.

Sophon Mongkolluksamee
sophon@g.swu.ac.th
Srinakharinwirot University

Natapon Tansangworn
natapon.tan@nectec.or.th
NECTEC

Sukumal Kitisin
sukumal.i@ku.th
Kasetsart University

ABSTRACT

This work proposes a homomorphic encryption (HE) platform, called *CYBLION*, enabling privacy-preserving outsourced computation on the cloud while a mobile application controls the privacy of IIoT data. *CYBLION* is developed based on our Node-RED HE playground, called *CipherFlow*, to provide a HE platform for IIoT that overcomes four main challenges: common arithmetic for IIoT applications, transmitting encrypted data over MQTT protocol, real-time data visualization, and simplification for non-cryptographic specialists. We demonstrate the practicality of *CYBLION* by implementing it through a real smart factory use case.

CCS CONCEPTS

• Security and privacy → Security services.

KEYWORDS

Homomorphic encryption, privacy, IIoT, Industry 4.0

ACM Reference Format:

Kalika Suksomboon, Nataset Tanabodee, Eakarat Saktawornlerd, Aimaschana Niruntasukrat, Chaveee Issariyapat, Sophon Mongkolluksamee, Natapon Tansangworn, and Sukumal Kitisin. 2023. Protecting Privacy in Industrial IIoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0. In *The 18th Asian Internet Engineering Conference (AINTEC'23)*, December 12–14, 2023, Hanoi, Vietnam. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/xxxxxx>

1 INTRODUCTION

Industry 4.0 and beyond drive the factory towards a completely interconnected framework that allows its supply chain to create an autonomous flow of data and information for efficient decision-making. Meanwhile, IIoT/IoT makers and application developers

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AINTEC'23, December 12–15, 2023, Hanoi, Vietnam

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-xxxxx-xxxxx-\$15.00

<https://doi.org/10.1145/xxxxxx>

benefit from emerging several IIoT cloud service platforms, e.g., AWS IIoT and Azure IIoT hub, to accelerate smart factories' solutions. However, a recent cloud data leak made the factory owners rethink whether the move to Industry 4.0 was just a trap. The main causation is that IIoT data are sensitive, so the factory owners want to keep them private from a third party. Therefore, there is a need to answer a rapid-fire question: how to protect industrial individual data privacy while letting factories enjoy Industry 4.0?

Although various privacy-enhancement technologies (PETs) and several HE platforms [2, 3] have been proposed for data privacy protection, none have been adopted to an IIoT cloud platform serving IIoT applications while zero knowledge is revealed to the cloud. To this end, we develop *CYBLION*, a HE platform based on "CipherFlow", our HE Node-RED playground [4], that allows IIoT/IoT developers to create their privacy-preserving data computation with the RNS-CKKS-based HE scheme [1] on Node-RED. While "CipherFlow" can preserve privacy computing over encrypted data, extending it to handle multiple users, transmitting HE data over the MQTT protocol, real-time data visualization, simplification for non-cryptographic specialists, and privacy control via a mobile app have not yet been taken for granted. This work proposes the design and implementation of *CYBLION*, including those extensions for protecting privacy in IIoT/IoT applications.

2 HOMOMORPHIC ENCRYPTION PLATFORM

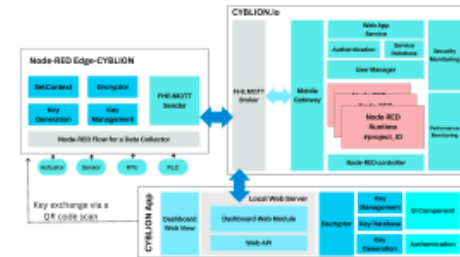


Figure 1: High-Level *CYBLION* architecture.

Architecture Design: To overcome the challenges mentioned in Section 1, we carefully design our HE platform composed of three main elements: edge computing (i.e., Node-RED-CYBLION-edge), cloud computing (i.e., CYBLION.io), and mobile application (i.e., CYBLION App) as shown in Figure 1. The Node-RED-CYBLION-edge is deployed in edge computing to encrypt IIoT data gathered from sensors and send them to the cloud. CYBLION.io is a Web service that allows users to create their computation functions on the cloud. Underneath cloud computing, an FHE.MQTT broker is developed to publish and subscribe to the encrypted messages under the FHE tag. The CYBLION App installed on users' smartphones enables them to control their IIoT data privacy and data visualization.

Node-RED-CYBLION-edge: The HE-based RNS-CKKS functions, i.e., setContext, key generation, key management, encryptor, and FHE.MQTT sender are encapsulated into a Node-RED node. This node allows the IIoT/IoT developers to merely drag and drop it at the end of the Node-RED flow to encrypt the collection of IIoT data with the public key generated by the CYBLION App. Then, the encrypted data are sent to the CYBLION cloud for computation.

CYBLION Cloud: We design the FHE.MQTT broker to handle a sequence of encrypted data chunks, attach an FHE tag, and interpret them as a single message to an evaluator in a Node-RED runtime and the decryptor in the CYBLION App. The mobile gateway also acts as an API between the CYBLION App and the CYBLION Cloud. Once a user creates an IIoT project in the CYBLION App, the mobile gateway receives a request from the App and then trick a Node-RED controller to build a Node-RED runtime accordingly. The UI of the Web Application service in the cloud allows multiple users to access and create their HE computation flows under their Node-RED workspace. We adopt "CipherFlow" [4] to provide the common HE arithmetic operations. Hence, the cloud can compute the encrypted data without decryption, so the data information does not leak into the cloud.

CYBLION App: We design the CYBLION App's UI as in Figure 2 to make non-cryptographic specialists comfortably using them. Our UI presents three options, i.e., small, medium, and large key sizes. Therefore, the user does not require knowledge of HE configuration. A users merely determine based on the number of multiplicative operations in a computation flow and the accuracy level of the results. With small and medium options, the computation flow can contain at most two multiplicative operations, while with a medium option, the user can achieve higher accuracy results. A large key size is recommended for a Node-RED flows containing a reciprocal operation (or seven multiplicative operations). HE environment is shared with the cloud and Node-RED-CYBLION-edge under the key exchange. QR code scanning plays a role of a public key exchange between Node-RED-CYBLION-edge and the CYBLION App. Once the dashboard Web module in the mobile App subscribes to the encrypted data from the cloud with the corresponding topic, it passes the encrypted data through a Web API to a decryptor for decrypting the encrypted data. Since our HE functions are based on the Microsoft SEAL Library¹ written in C#, we develop a Web API for translating Xamarin code to C# so the decryptor can perform on the CYBLION App. The decrypted data are sent to the dashboard web view for data visualization.

¹<https://github.com/microsoft/SEAL>.



Figure 2: UI for the CYBLION App; (a) key generation, (b) key lists, (c) sending a relinear key to the cloud, and (d) sending a public key to the Node-RED-CYBLION-edge via a QR code scan.

3 DEMONSTRATION AND DISCUSSION

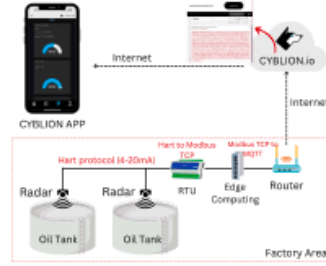


Figure 3: IIoT use case: privacy-preserving smart tank farm monitoring system.

We demonstrate the practicability of CYBLION through a smart tank farm monitoring system as shown in Figure 3. The radar sensors transmit the 4–20 mA signal to an RTU and let it convert to Modbus TPC. An IIoT gateway installed Node-RED-CYBLION-edge collects the sensor data and then converts Modbus TCP to MQTT protocol. The data is encrypted with the small-size public key generated by CYBLION App, and then, published the encrypted one to the CYBLION cloud. A computing flow is created on the cloud to convert the sensor data to the amount of oil production. The dashboard can present the decrypted data within 3 sec after a sensor sends data to the cloud. In the future, we will implement CYBLION to the other IIoT applications.²

REFERENCES

- [1] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song, 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer International Publishing, Cham, 409–437.
- [2] Roshan Dathathri and et al. 2020. EVA: An Encrypted Vector Arithmetic Language and Compiler for Efficient Homomorphic Computation. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, 546–561.
- [3] Angelou Nick. March 2020. *Morfix: Homomorphic Encryption as a Service*. Retrieved March 2, 2022 from <https://github.com/morfix-io>
- [4] Natasat Tanabodee and et al. 2022. CipherFlow: A Playground for Developing Privacy-Preserving IIoT in Node-RED. In *AINTEC'22*, 18–25.

²This work was funded and financially supported by NBTC, Thailand, under the grant A63-1-(2)-017.

ภาคผนวก ง.
บทความระดับนานาชาติ

CipherFlow: A Homomorphic Operation Toolkit for Privacy-Preserving Industrial IoT Applications

Kalika Suksomboon, *Member, IEEE*, Natasat Tanabodee, Aimaschana Niruntasukrat, *Member, IEEE*,
 Sophon Mongkolluksamee, *Member, IEEE*, Sukumal Kitisin, *Member, IEEE*, Jun Kurihara, *Member, IEEE*,
 Toshiaki Tanaka, *Member, IEEE*, Natapon Tansangworn, Chavee Issariyapat

Abstract—The emergence of privacy breaches in Industrial IoT (IIoT)/IoT surges demands embedding privacy-by-design into IoT systems. Decades of work on privacy-enhancing technologies have resulted in myriad ways to secure outsourcing computation with homomorphic encryption (HE) schemes that allow the third party (e.g., cloud) to compute over encrypted data. However, IoT programmers are still daunted by those HE tools due to their complicated parameter settings and configurations, preventing the rapid growth of privacy-preserving IIoT/IoT application developments. We consider two keys required for HE tools to accelerate these developments: 1) abstraction and flexibility to automatically manage HE's complicated mechanisms and parameters, and 2) ease of the IoT-specific computational functions such as average and Specific Energy Consumption (SEC) calculations of manufacturing applications. Meanwhile, existing HE tools and compilers are primarily written in low-level languages like C++, posing a high entry barrier to novice IoT application developers. Moreover, none of the HE platforms have been investigated and equipped with IoT-friendly computational functions, i.e., *reciprocal or multiplicative inverse*. To these ends, this paper proposes *CipherFlow*, a HE operation toolkit based on Node-RED, a visually and flexibly programmable dataflow tool for rapid IoT development. *CipherFlow* offers three pipeline abstractions: i) a playground of drag-and-drop HE nodes in Node-RED, ii) optimal parameter setting guidelines, iii) various arithmetic operations used for IoT-specific calculations. As a result, IoT developers can create and verify their secured computing functions on data flows. Through the performance evaluation on IIoT use cases, we demonstrate the applicability and effectiveness of *CipherFlow*. Furthermore, our evaluation shows the performance and security trade-off depending on the HE parameters.

Index Terms—Homomorphic Encryption, IIoT/IoT, Privacy Enhancement Technologies, Node-RED, RNS-CKKS.

I. INTRODUCTION

The transition towards Industry 4.0 sounds promising, driven by the Industrial Internet of Things (IIoT), without considering the upstream challenges of data breaches in the cloud platform. Nevertheless, the data generated in the IIoT are very sensitive; hence, the privacy and security of these data are of significant importance.

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

K. Suksomboon, A. Niruntasukratwas, and N. Tansangworn are with NECTEC, Thailand, e-mail: (kalika.suk@nectec.or.th, aimaschana.niruntasukrat@nectec.or.th and natapon.tan@nectec.or.th); S. Mongkolluksamee is with Srinakharinwirot University, Thailand, e-mail: (sophon@g.swu.ac.th); N. Tanabodee was with Kasetsart University, Thailand, e-mail: (plume142@gmail.com); S. Kitisin is with Kasetsart University, Thailand e-mail: (sukumal.i@ku.th); J. Kurihara and T. Tanaka are with Hyogo University, Japan, e-mail: (kurihara@iecc.org and Toshi@gis.u-hyogo.ac.jp)

Widely used and deployed, the Internet of Things (IoT) has shifted the predominant usage of cloud towards IoT cloud computing, such as Azure IoT hub and Amazon IoT, which provides simple management services for billions of IoT devices. The agility and simplicity of IoT cloud computing encourage novice IoT programmers to become newbie IoT developers to enjoy creating new IoT solutions for the market. However, recent IoT data breach news has escalated the severity of users' privacy concerns. Supported by the survey in [1]–[3], respondents have admitted that privacy is the biggest challenge for IoT adoption.

Embedding privacy by design into IoT systems is key to lifting confidence in IoT usage. Once sensitive information in data has been adequately constrained through privacy-enhancing technologies (PETs) [4], such as homomorphic encryption (HE), secure multi-party computation (SMPC), federated learning, and k -anonymization, the privacy-by-policy approaches become less important. Although many efforts [5]–[8] on PETs are underway to secure outsourcing computation with HE, they are still far from the rapid development of practical privacy-preserving IoT systems. While many HE tools and compilers are well developed and adopted in numerous research works, most were written in low-level languages like C++, posing a high entry barrier to newbie IoT application developers [8]. Moreover, as far as we know, there has been proposed no HE platform supporting IIoT/IoT-specific arithmetic computations, i.e., *reciprocal or multiplicative inverse*. However, such functions are widely and frequently leveraged in most IoT scenarios, like the average calculations of machine temperature and the specific energy consumption (SEC) for energy management in manufacturing [9]. Thus, even though they desire to consolidate their IoT solutions with a HE technique, they may need bespoke tools that provide simplicity, abstraction, and flexibility.

To these ends, this paper focuses on enhancing Node-RED, a low-code IoT platform for IoT developers [10], to perform HE algorithms for secure computation in IoT data. With Node-RED, IoT developers can build their desired applications without the coding background. The pre-defined code built in Node.js is wrapped in a Node-RED node to perform an IoT-specific function. The developers merely wire a collection of Node-RED nodes to perform their desired IoT solutions.

The enhancement introduced in this paper called *CipherFlow*, is an HE playground in Node-RED for IoT developers to play and develop privacy-preserving solutions for IoT. The HE nodes in *CipherFlow* are built with

node-seal [7], a wrapper of the Microsoft SEAL library [6]. Therefore, the *CipherFlow* does not require pre-installation of the Microsoft SEAL library. Written in Node.js, *node-seal* is seamlessly integrated into the Node-RED platform, allowing us to develop *CipherFlow*, the extension nodes to Node-RED. *CipherFlow* offers three pipeline abstractions and flexibility, i.e., 1) a playground of dragging-and-dropping HE nodes in Node-RED workspaces, 2) arithmetic operators designed to execute IoT-specific calculations, and 3) optimal parameter setting guidelines. As a result, IoT developers can create and verify their secured computing functions on their dataflow. However, our development did not emerge overnight. Developing *CipherFlow* has to overcome the following challenges.

HE computation. The Node-RED platform does not directly support four requirements of the HE computation: (i) to unify the HE parameter set in the same Node-RED's dataflow, (ii) to import and export keys in/out of the Node-RED platform, (iii) parallel computation on arithmetic operations, and (iv) the garbage collector (GC) for optimal memory management. To fill these gaps, we carefully design the *CipherFlow* to meet these requirements (see in § IV).

Reciprocal operation. The RNS-CKKS scheme does not support the reciprocal operation (or multiplicative inverse), a common arithmetic operator. To this end, we propose a reciprocal algorithm by modifying the Goldschmidt division iteration method [11] in § V. However, it results in inherent errors due to its approximation. To deal with this shortcoming, we investigate the error bounds and provide the parameter-setting guidelines to the users (§ VI).

Parameter setting. Despite the available dozen choices of HE schemes, the *CipherFlow* is based on the RNS-CKKS scheme [12], which requires three main parameters to set up the HE environment: 1) polynomial modulus degree, 2) coefficient modulus and 3) scale factor. These parameters determine the percentage error of ciphertext resulting from the compute operations. Although choosing the optimal RNS-CKKS parameters requires the skill of the cryptography experts [13], which is beyond the scope of this paper, we provide a hint to the *CipherFlow*'s users by presenting an indicator—called “Chain-index” value. We use this indicator to fine-tune the optimal setting in § VII.

To our knowledge, this paper makes the following contributions:

- 1) *CipherFlow* architecture design: we design the *CipherFlow* architecture with three abstraction layers that enable execution HE seamlessly on the Node-RED platform.
- 2) Homomorphic operation nodes for Node-RED: we propose the algorithms specific to the arithmetic operations on the RNS-CKKS scheme that handles the HE operation constraints and the noise management.
- 3) Reciprocal algorithm: we propose a technique to turn the HE-based RNS-CKKS scheme to do a multiplicative inverse of the input real number using the Gold Smith iteration method. The algorithm includes a noise

explosion handling method. We also analyze the noise bound of this approximation and interpret it into the reciprocal algorithm that allows users to manage their desired applicable input range.

- 4) Homomorphic encryption parameter selections: we investigate the effect of the parameter setting of RNS-CKKS on the noise of outputs. The optimum options are recommended to the users as the guideline regarding their IIoT applications' requirements. The offered options comply with the security level and the performance trade-off between the encrypted data size and the output results' precision.
- 5) Industrial use case applications: we implement the *CipherFlow* to two use cases, i.e., a smart tank farm monitoring system and an SEC for a smart energy monitoring system.

II. RELATED WORK

A. Node-RED

Since the third-party developers can easily extend the functionality of the platform through publishing nodes and configurations of these nodes by being built in Node.js, several works paid attention to extending the applicability of Node-RED in different directions, e.g., [14]–[18]. However, only a few approaches, i.e., [14], [18], proposed functionalities to enhance privacy-preserving dataflows in Node-RED. RedCASTLE [14] provides k_s -anonymity in Node-RED for anonymizing the personal information in IoT streaming data. This approach cannot prevent third parties from learning the data while computing them in their environment. On the other hand, Kawamoto [18] has adopted the PASTE [19] in Node-RED to prevent the third party from learning the IoT data during data aggregation by the differential privacy and Paillier cryptosystem. With Paillier, his Node-RED extension supports only an add operation on an integer. In contrast, we provide comprehensive arithmetic operations and support computing over real numbers (which are approximated by fixed-point numbers) by means of the RNS-CKKS scheme [12]. To the best of our knowledge, we propose the first fully homomorphic encryption to Node-RED.

B. Libraries/Compilers of Homomorphic Encryption

The state-of-the-art approaches focus on providing open-source software libraries, e.g., the Microsoft SEAL [6], PALISADE [20], HELib [21], for variants of HE schemes, e.g., BFV [22], BGV [23] and CKKS [12]. These open-source software libraries allow any developers freely adopt HE schemes to their applications. However, developing HE applications remains difficult due to the need for low-level programming language and cryptography knowledge. Several general-purpose compilers have been proposed in different languages, i.e., Julia, Haskell, C++, Python [?], [13], Node.js [7] (see detail in [8]) to alleviate this difficulty.

The HE compilers are aimed to provide a high-level abstraction for developing HE-based applications. These tools manage key setup, encryption, decryption, and ciphertext operation in the background.

Even though these programming languages can be written comfortably by most typical programmers, they are not easily learned and, hence, inconvenient for typical IoT developers. On the one hand, EVA [13], HECATE [24], [25], ELASM [26] and PYfhel [27] have been designed for Python-based machine learning (ML) applications because most ML developers are familiar with Python coding.

EVA [13], HECATE [24] and ELASM [26] have been proposed to manage noise over RNS-CKKS computation. On the other hand, IoT developers are more familiar with Node-RED, which does not require hard-coded program writing skills. Therefore, this paper proposes *CipherFlow*, a HE playground for IoT developers. It offers drag-and-drop features that make users conveniently select HE nodes and define dataflows in the Node-RED platform to perform a task.

CHET [?] is an optimized compiler for encrypted DNN inference. CHET provides a domain-specific language and transforms an input tensor circuit into a sequence of FHE operations.

nGraph-HE [?] extends Intel's nGraph [?], and existing deep learning graph compiler, to enable encrypted deep learning. The compiler implements the HE backed for nGraph to deploy neural network models with popular deep learning frameworks like TensorFlow.

AHEC [28] supports mGraph and Tensorflow as front-end and SEAL (RNS-CKKS) as back-end with supporting multiple hardware back-end through GPU-accelerated HE library.

Marble [29] Marble: Making Fully Homomorphic Encryption Accessible to All

[30] OpenFHE

[31] FHE on Go, a multiparty homomorphic encryption library in Go. Such systems, by nature, are highly interactive, concurrent and cross-platform. For this reason, implementing them may represent a significant investment in terms of time and effort when using C++, which most of the state-of-the-art HE libraries are using.

C. Homomorphic arithmetic operations

Homomorphic encryption typically allows arithmetic operations to work on multiplying and adding due to the ring properties. The limited arithmetic operations cause its applications. Most recent research approaches [13], [26], [28], [32], [33] focus on machine learning operations-specific tools rather than basic arithmetic operations, such as minus and division, which are commonly used in IIoT applications.

The common arithmetic operations have been firstly considered in [34]. This research has proposed three iterative methods for a multiplicative inverse approximation of homomorphic encryption. The first approach based on Fermat's Little Theorem allows homomorphic evaluation of the multiplicative inverse of an integer b by modulo a prime number p . However, this method cannot provide a multiplicative inverse over real numbers. The second approach is based on the Newton-Raphson iteration method. The bright sight of this method is flexible to the desired precision levels by setting the convergence threshold. However, the downside is that the number of multiplicative depths cannot be controlled leading

to the difficulty of choosing the homomorphic parameters. The third approach uses the approximate inverse of Goldschmidt's convergence method. The precision of this approximation relies on the number of iterations. and

Privacy-preserving machine learning (PPML)

[35] developers friendly. [36]

[37] A Language and Compiler for Homomorphic Encryption Made easY ... take programs written in a high-level language (e.g., Julia, C++, or a custom DSL) and transform them into arithmetic representations that can be evaluated using a backend FHE scheme

CARM [32] uses CUDA to accelerate FHE the first optimized GPU implementation.

[38] Optimize key switching

III. PRELIMINARIES

We develop the *CipherFlow* based on the RNS-CKKS scheme under the Microsoft SEAL library [6]. The RNS-CKKS scheme is composed of the following stages;

- 1) **Parameter setting.** CKKS requires three parameters: (i) polynomial ring of degree N , (ii) the coefficients modulus chain $\mathbf{q} = [q_L, q_{L-1}, \dots, q_1]$, where $q_l \in \mathbf{q}$ is a coefficients modulus level l for $1 \leq l \leq L$ and L is the highest level, and (iii) scale factor Δ . These parameters are encoded in a context. This context is further used to create the HE environment.
- 2) **RNS-CKKS Encoder.** Encoding in RNS-CKKS is a mapping function of a complex number of message z to a plaintext p such that: $Ecd : \mathbb{Z} \rightarrow \mathbb{P}$, where $z \in \mathbb{Z}$, $\mathbb{Z} \subseteq \mathbb{C}^{N/2}$ for $\mathbb{C}$ is a complex domain with dimension $N/2$. A plaintext space $\mathbb{P}$ is a set of polynomials in the cyclotomic ring $\mathcal{R} = \mathbb{Z}[X]/\Phi_M(X)$ with magnitude bounded by ciphertext modulus. The Encoder $Ecd(z, N, \Delta)$ takes a message of a complex number z to encode with polynomial modulus N and scale the encoding space by a scale factor Δ . The scale factor is used to enlarge the space to prevent error overflow during the homomorphic evaluation process. The output of encoding is a polynomial plaintext p . RNS-CKKS encoding is inevitable to encrypt a vector of multiple plaintext in a single ciphertext for efficient homomorphic computation.
- 3) **Key generation.** The KeyGenerator takes the context to create the HE environment and constructs a tuple of keys, i.e., (a public key pk , a secret key sk , a relinear key rk).
- 4) **Encryption.** The Encryption takes context to generate the HE environment and the public key pk . The idea of CKKS scheme is to add noise for security purposes as a part of the message we want to encrypt— so-called Ring-Learning with Errors (Ring-LWE). The encoder transforms a message m to a plaintext P , in which a real number is represented in the fixed-point number scaled by the scale factor Δ . Hence, noise is added to the plaintext due to approximation. Once the Encryption receives the plaintext P as an input, it applies pk to encrypt P to a ciphertext c . A ciphertext c is in the

form of polynomials in which the coefficient modulus bounds the ciphertext space.

- 5) **Evaluation.** The Evaluator takes the context to generate the HE environment. It supports arithmetic operations. CKKS allows the ciphertext to do arithmetic operations on both ciphertext and plaintext. Once the Evaluator takes two ciphertexts, c_1 and c_2 to do the operation (e.g., add $\oplus$ and multiply $\otimes$), it returns the result as a ciphertext, $c' = c_1 \oplus c_2$, and $c'' = c_1 \otimes c_2$.
- 6) **Decryption.** Decryptor takes the context to generate the HE environment. The secret key sk is used to decrypt the ciphertext and yields the plaintext. The decoder transforms the plaintext back to the approximate real number corresponding to the ciphertext.
- 7) **RNS-CKKS Decoder.** Decoding is the reverse process of encoding that transforms a plaintext polynomial p into a complex vector z .

The RNS-CKKS of Microsoft SEAL offers the control operations to manage noise and error of ciphertext after doing RNS-CKKS arithmetic operations as follows.

- 1) **Relinearize.** Relinearize reshapes the ciphertext without changing the underlying message by applying key rk . This operation helps limit noise growth and the ciphertext size.
- 2) **MODSWITCH.** MODSWITCH is used to switch the coefficient modulus q_l to q_{l-1} .
- 3) **RESCALE.** RESCALE is the heart of RNS-CKKS, which is used to reduce the magnitude of the plaintext. After multiplication with two ciphertexts, the RESCALE operation is needed. For instance, two ciphertexts with scale 2^Δ will grow to $2^\Delta * 2^\Delta = 2^{2\Delta}$ after multiplication. CKKS uses the RESCALE operation to scale down the ciphertext with scale $2^{2\Delta}$ by dividing it by 2^Δ . The RESCALE also switches the coefficient modulus from q_l to q_{l-1} .
- 4) **SETSCALE.** SETSCALE is used to force the scale to the exact value.

IV. PROPOSED CIPHERFLOW ARCHITECTURE

A. Design Principles

We aim to develop the *CipherFlow* to be a playground for IoT developers to play and learn how to apply the HE functionalities to preserve the privacy of IoT applications. The *CipherFlow* is designed to meet these desired goals: simplicity, flexibility, and abstraction. To simplify a way to implement the HE scheme in IoT applications, we develop the *CipherFlow* as the node extension in the Node-RED pallet and preserve the HE functionalities with the Microsoft SEAL [6] in the backend library. On top of that, node-seal wraps the Microsoft SEAL library, allowing us to develop and integrate the HE node extension to Node-RED. Therefore, the IoT developer can install the *CipherFlow* as the Node-RED extension (see in § VII).

With *CipherFlow*, the developers can easily drag and drop the HE nodes and flexibly compose the computation functions in the Node-RED platform. Moreover, the *CipherFlow* allows developers to create a new tuple of HE keys or download

TABLE I: Notations and definitions

Notation	Definition
λ	Security level (bits)
N	Polynomial modulus
$\mathbb{Q}(\lambda, N)$	Coefficient modulus budget
Q	Coefficient modulus
L	Maximum level of coefficient modulus chain
Δ	Scale factor
s_x	Scale of plaintext/ciphertext x
q_l	Coefficient modulus of the l -th level
δ	Chain-index, where $\delta = L - 2$
sk, pk, rk	The secret key, public key, and relinear key, respectively
c_i	Ciphertext, where $Enc(p_i, sk)$
p_i	Plaintext, where $Ecd(m_i, N, \Delta)$
σ^2	Coefficient variance of the Ring-LWE error distribution
$Enc(p, sk)$	Encryption over plaintext p with a secret key sk
$Ecd(z, N, \Delta)$	Encoding message z with polynomial modulus N and a scale factor Δ
$\oplus, \otimes, \otimes^i$	Add, multiply and exponent over i operations respectively
$\epsilon_{min}, \epsilon_{max}$	Range of usable input data for the reciprocal operation
$\Phi_M(x)$	Cyclotomic Polynomials of the root of unity M
$\mathcal{T} : \mathcal{U} \rightarrow \mathcal{V}$	A linear transform function that maps a vector space $\mathcal{U}$ to a vector space $\mathcal{V}$ with respect to the underlying linear structure of a vector space $\mathcal{V}$
$\mathcal{T}^{he} : \mathcal{U} \rightarrow \mathcal{V}$	A linear transform function under homomorphism that maps a vector space $\mathcal{U}$ to a vector space $\mathcal{V}$ with respect to the underlying linear structure of a vector space $\mathcal{V}$

the existing ones to the flows. The *CipherFlow* abstracts the HE nodes for some CKKS functions, i.e., encode, decode, RESCALE, SETSCALE and Relinearize, that are required for encryption, decryption and evaluation. By this abstraction, the developers do not need to determine when and how to use these functions wisely.

B. CipherFlow Architecture

Regarding the design principles, the *CipherFlow* architecture comprises three layers: (i) backend library, (ii) C++ abstraction, and (iii) Node-RED extension as shown in Figure 1. **Backend library.** The *CipherFlow* uses the Microsoft SEAL version 4.0.0 as the backend library written in C++. We select the CKKS scheme in this library as the core of our node extension. Hence, the objects and core functions of CKKS are embedded in our backend library.

C++ abstraction. To abstract C++, the node-seal wraps CKKS core functions and objects in Node.js. With Web Assembly, node-seal is a portable target for C++ in the Microsoft SEAL library, enabling faster deployment on Node-RED. The node-seal also provides memory management and only contains a low-level API close to C++.

Node-RED extension. The *CipherFlow* presents the CKKS functionalities and arithmetic operations in the Node-RED extension that raises the level of abstraction for putting HE functionalities in the dataflows. Nodes warp Node.js code and perform individual operations. The developers merely select HE nodes by dragging and dropping them on the Node-RED's workspace. Underneath this node extension, the *CipherFlow* provides the options to import and export the keys and context

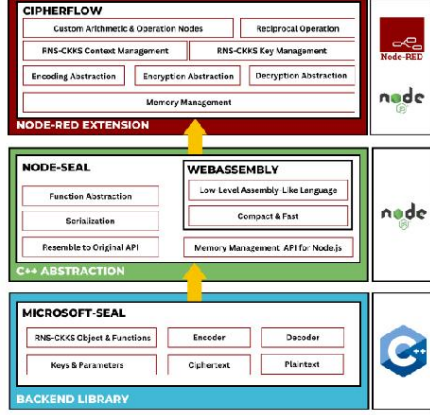


Fig. 1: CipherFlow Architecture

to/from the Node-RED platform. Beyond the CKKS arithmetic operations in node-seal, we develop the reciprocal operation to the node extension.

V. HOMOMORPHIC ARITHMETIC OPERATION NODES

A. RNS-CKKS Arithmetic Operation Constraints

Computing RNS-CKKS arithmetic operations must satisfy the following constraints.

$$l_x = l_y, \quad \text{for } x \otimes y \text{ and } x \oplus y \quad (1)$$

$$s_x = s_y, \quad \text{for } x \oplus y \quad (2)$$

B. Arithmetic Operation Nodes for Node-RED

We propose arithmetic operation algorithms for Node-RED that resemble the arithmetic operations of Microsoft SEAL and combine a sequence of noise and error control operations.

Algorithm 1: $AddE(x, m)$

```

Input:  $x, y$ 
Result:  $z, \delta_z$ 
/* Initialization */
1  $\delta_x \leftarrow \text{get.chain-index}(x), \delta_y \leftarrow \text{get.chain-index}(y)$ 
2  $\gamma \leftarrow |l_y - l_x|$ 
3 while  $\gamma > 0$  do
4   if  $l_y > l_x$  then
5      $y \leftarrow \text{MODSWITCH}(y)$ 
6   else
7      $x \leftarrow \text{MODSWITCH}(x)$ 
/* Evaluation */
8  $z \leftarrow x \oplus y$ 
9  $\delta_z \leftarrow \text{get.chain-index}(z)$ 

```

Algorithm 2: $AddP(x, m)$

```

Input:  $x, m$ 
Result:  $z, \delta_z$ 
/* Initialization */
1  $\delta_x \leftarrow \text{get.chain-index}(x)$ 
2  $N, \Delta \leftarrow \text{get.context}(x)$ 
3  $p \leftarrow \text{Ecd}(m, N, \Delta)$ 
4 for  $n = 1$  to  $\delta - \delta_x$  do
5    $p \leftarrow \text{MODSWITCH}(p)$ 
/* Evaluation */
6  $z \leftarrow x \oplus p$ 
7  $\delta_z \leftarrow \text{get.chain-index}(z)$ 

```

Algorithm 3: $MultiE(x, y, rk)$

```

Input:  $x, y, rk$ 
Result:  $z, \delta_z$ 
/* Initialization */
1  $\delta_x \leftarrow \text{get.chain-index}(x), \delta_y \leftarrow \text{get.chain-index}(y)$ 
2  $\gamma \leftarrow |l_y - l_x|$ 
3 for  $n = 1$  to  $\delta - \delta_x$  do
4    $p \leftarrow \text{MODSWITCH}(p)$ 
/* Evaluation */
5  $z \leftarrow x \otimes y$ 
6  $\delta_z \leftarrow \text{get.chain-index}(z)$ 
7  $z \leftarrow \text{Relinearize}(z, rk)$ 
8  $z \leftarrow \text{RESCALE}(z, \Delta)$ 
9  $z \leftarrow \text{SETSCALE}(z, \Delta)$ 

```

Algorithm 4: $MultiP(x, p, rk)$

```

Input:  $x, m, rk$ 
Result:  $z, \delta_z$ 
/* Initialization */
1  $\delta_x \leftarrow \text{get.chain-index}(x)$ 
2  $N, \Delta \leftarrow \text{get.context}(x)$ 
3  $p \leftarrow \text{Ecd}(m, N, \Delta)$ 
4 for  $n = 1$  to  $\delta - \delta_x$  do
5    $y \leftarrow \text{MODSWITCH}(y)$ 
/* Evaluation */
6  $z \leftarrow x \otimes p$ 
7  $\delta_z \leftarrow \text{get.chain-index}(z)$ 
8  $z \leftarrow \text{Relinearize}(z, rk)$ 
9  $z \leftarrow \text{RESCALE}(z, \Delta)$ 
10  $z \leftarrow \text{SETSCALE}(z, \Delta)$ 

```

Algorithm 5: $negate(x)$

```

Input:  $x$ 
Result:  $x, \delta_x$ 
/* Evaluation */
1  $x \leftarrow \neg x$ 
2  $\delta_x \leftarrow \text{get.chain-index}(x)$ 

```


C. Reciprocal Algorithm

The Microsoft SEAL provides additive and multiplicative operations but does not provide the reciprocal or multiplicative inverse operation (i.e., $1/x$), a common arithmetic operation for IoT applications. We, therefore, develop the reciprocal operation for the *CipherFlow*. However, computing the reciprocal operation ($1/x$) in HE is not straightforward since division in HE is not applicable due to the Ring properties. Thus, we explore two approximation methods, i.e., 1) Newton-Raphson and 2) Goldshmidt, that can perform $1/x$ without division. The Newton-Raphson and Goldshmidt methods are both iterative and quadratically convergent, allowing us to obtain the approximation of $1/x$. However, the Newton-Raphson method requires guessing an initial value, and a wrong initial value can result in the wrong convergence or require more iterations to converge. Therefore, with Newton-Raphson, we cannot control the number of HE operations causing the difficulty of setting the RNS-CKKS parameters beforehand.

On the other hand, the Goldshmidt iteration method [39] approximates $1/x$ does not require an initial value, but it requires the number of iterations d in which the error rate of the output of the algorithm compared to $1/x$ is bounded by $(1-x)^{2^{d+1}}$ (see Lemma 1 in [39]). Fig 2 demonstrates the percentage error of the approximate $1/x$ in different d . Since the number of iterations is specified by d , we can determine the required coefficient modulus chain $\mathbf{q}$ in the parameter setting.

Although the Goldshmidt iteration method is suitable for performing the reciprocal operation ($1/x$) in HE, it limits the working space $x \in (0, 2)$. Thus, we generalize the Goldshmidt iteration algorithm to be applied for the other space $\mathcal{U}$ for $u \in \mathcal{U}$ and $\mathcal{U}$ is the set of positive real numbers, where $u \in (0, u_{max})$ for $u_{max} > 0$.

Consider a linear transformation $\mathcal{T}_{\mathcal{U} \rightarrow \mathcal{V}} : \mathcal{U} \rightarrow \mathcal{V}$ that maps from a positive number in range $(0, u_{max})$ to a positive number in range $(0, 2)$. Accordingly, we denote $\mathcal{T}_{\mathcal{U} \rightarrow \mathcal{V}}$ by

$$\mathcal{T}_{\mathcal{U} \rightarrow \mathcal{V}}(u, \alpha) = \alpha u, \quad (3)$$

where $\alpha = \frac{2}{u_{max}}$.

Let $\mathbb{U}$ and $\mathbb{V}$ be vector space of equal dimension over a ring R , and let α be a nonzero complex number in $\mathbb{Z}$. Then a plaintext of α is a linear mapping of $\mathbb{U}$ into $\mathbb{V}$. Hence, we denote a linear transform $\mathcal{T}_{\mathbb{U} \rightarrow \mathbb{V}}^{he} : \mathbb{U} \rightarrow \mathbb{V}$ on homomorphic RNS-CKKS spaces where

$$\mathcal{T}_{\mathbb{U} \rightarrow \mathbb{V}}^{he}(u, \alpha) = \text{MultiP}(u, \alpha, rk). \quad (4)$$

We, then, propose the reciprocal algorithm (as shown in Algorithm 6) for the CKKS scheme by integrating the linear transformation $\mathcal{T}_{\mathbb{U} \rightarrow \mathbb{V}}^{he}(u, \alpha)$ in the Goldshmidt iteration algorithm in [39].

Algorithm 6 requires predetermining the number of iterations d and the range of space we want to map to. It is worth noticing that larger d yields less error in the approximation. However, approximation precision has to be traded with the number of multiplications, causing a larger *Chain-index*. To compromise between the large *Chain-index* and its precision, we recommend setting $d = 3$, which requires 6 *Chain-index* ($\delta = 6$).

The applicable range of the reciprocal algorithm can be determined by the error bound of the Goldshmidt iteration method. Let $x \in [0, 2]$ be the applicable range in the Goldshmidt iteration method. We define e as the desired error rate of the reciprocal algorithm.

Theorem V.1. For $(b_{min}, b_{max}) \subset (0, u_{max})$ and a positive integer d , the applicable range of the reciprocal algorithm regarding the error rate e is determined by

$$b_{min} > (1 - e^{\frac{1}{2^{d+1}}}) \times \left(\frac{u_{max}}{2}\right), \quad (5)$$

and

$$b_{max} < u_{max} - b_{min}. \quad (6)$$

Proof. Let $\tilde{e}$ and ϵ be the error rates of outputs from the approximate function $f(x) = 1/x$ compared to true value $1/x$ by the Goldshmidt iteration method and the noise produced by the HE operations, respectively. From Lemma 1 in [11], the error rate from approximation using the Goldshmidt iteration method is bounded by $(1-x)^{2^{d+1}}$ for $x \in (0, 2)$. The error is always negative because the approximated result is always smaller than the true value $1/x$. We thus determine x from a given $\tilde{e}$ by

$$\tilde{e} \leq (1-x)^{2^{d+1}}, \quad (7)$$

$$\tilde{e}^{\frac{1}{2^{d+1}}} \leq 1-x, \quad (8)$$

$$x \geq 1 - \tilde{e}^{\frac{1}{2^{d+1}}}. \quad (9)$$

The error of the reciprocal algorithm is added by ϵ and accumulated by the number of iterations d . Hence, the desired error rate e includes these two errors $\tilde{e}$ and ϵ , where $e > \tilde{e} + \epsilon$. Then, we have

$$1 - \tilde{e}^{\frac{1}{2^{d+1}}} > 1 - (\tilde{e} + \epsilon)^{\frac{1}{2^{d+1}}} > 1 - e^{\frac{1}{2^{d+1}}}. \quad (10)$$

Thus, we have $x > 1 - e^{\frac{1}{2^{d+1}}}$ for a given d and $x \in (0, 2)$. Since we expand the space of x from $(0, 2)$ to $(0, u_{max})$ by the linear transform $\mathcal{T}_{(0,2) \rightarrow (0, u_{max})}(x, \alpha) = \alpha x$. Hence, after we replace x by αx in (9) we can obtain the lower bound of the applicable range as

$$b_{min} > (1 - e^{\frac{1}{2^{d+1}}}) \times \frac{1}{\alpha} = (1 - e^{\frac{1}{2^{d+1}}}) \times \left(\frac{u_{max}}{2}\right), \quad (11)$$

where $\alpha = \frac{2}{u_{max}}$. Since the applicable range of the Goldshmidt iteration method is symmetric, the upper bound is calculated by

$$b_{max} < u_{max} - b_{min}. \quad (12)$$

□

The implication of Theorem V.1 leads us to be able to compute u_{max} from a given applicable range of the reciprocal algorithm (b_{min}, b_{max}) , d and the level of precision (i.e., error rate e). That is

$$u_{max} \geq (1 - e^{\frac{1}{2^{d+1}}}) \times \left(\frac{b_{min}}{2}\right). \quad (13)$$

Therefore, we can pre-compute u_{max} by (13) and use it as an input of Algorithm 6.

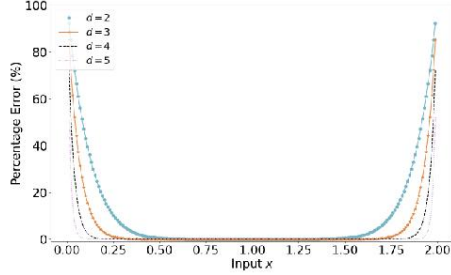


Fig. 2: The percentage errors (%) of the Goldshmidt iteration method to approximate a function $f(x) = \frac{1}{x}$, where d is the number of iterations and $x \in (0, 2)$.

Algorithm 6: *Reciprocal*(x, d, rk, u_{max})

```

Input:  $x, d, rk, u_{max}$ 
Result:  $z, \delta_z$ 
/* Initialization */
1  $\alpha \leftarrow 2/u_{max}$ 
2  $v \leftarrow \mathcal{T}_{U \rightarrow V}^{he}(x, \alpha)$ 
3  $\delta_z \leftarrow \text{get.chain-index}(z)$ 
4  $p_1 \leftarrow \text{Ecd}(1, N, \Delta)$ 
5  $p_2 \leftarrow \text{Ecd}(2, N, \Delta)$ 
6 for  $n = 1$  to  $\delta_z$  do
7    $p_1 \leftarrow \text{MODSWITCH}(p_1)$ 
8    $p_2 \leftarrow \text{MODSWITCH}(p_2)$ 
/* Evaluation */
9  $v \leftarrow \text{negate}(v)$ 
10  $b_0 \leftarrow \text{AddP}(v, p_1)$ 
11  $a_0 \leftarrow \text{AddP}(v, p_2)$ 
12 for  $n = 0$  to  $d - 1$  do
13    $b_{n+1} \leftarrow \text{MultiE}(b_n, b_n)$ 
14    $\delta_{a_{n+1}} \leftarrow \text{get.chain-index}(b_{n+1})$ 
15    $a_{n+1} \leftarrow \text{AddP}(b_{n+1}, p_1)$ 
16    $a_{n+1} \leftarrow \text{MultiE}(a_{n+1}, a_n, rk)$ 
17  $\delta_{a_n} \leftarrow \text{get.chain-index}(a_n)$   $p_\alpha \leftarrow \text{Ecd}(\alpha, N, \Delta)$ 
18 for  $n = 1$  to  $\delta_{a_n}$  do
19    $p_1 \leftarrow \text{MODSWITCH}(p_1)$ 
20  $p_1 \leftarrow \mathcal{T}_{U \rightarrow V}^{he}(p_1, \alpha)$ 
21  $z \leftarrow \text{MultiP}(a_n, p_1, rk)$ 
22  $z \leftarrow \text{Relinearize}(z, rk)$ 
23  $z \leftarrow \text{RESCALE}(z, \Delta)$ 
24  $z \leftarrow \text{SETSCALE}(z, \Delta)$ 

```

VI. HOMOMORPHIC ENCRYPTION PARAMETER SELECTION

Due to the approximate nature of the scheme, application developers using CKKS must ensure that the evaluation output is within a tolerable error of the corresponding plaintext computation. Choosing appropriate parameters requires understanding how the noise will grow through the computation. We present a comprehensive noise analysis of CKKS, considering

TABLE II: Time complexity and noise of RNS-CKKS encoding [40] and operations [26]: l is level and σ is std. deviation.

Operations	Time Complexity	Noise
encode	$O(N^3)$	$\frac{N}{\pi \Delta}$
decode	$O(N^2)$	0
negate, add, mul	$O(N \cdot l)$	0
modswitch	$O(N \cdot l)$	0
rotation, relinearize	$O(N \log N \cdot l^2)$	$\frac{8\sqrt{3}}{3} \sigma l N + \frac{8\sqrt{2}}{3} N + \sqrt{3N}$
rescale	$O(N \log N \cdot l^2)$	$\frac{8\sqrt{2}}{3} l N + \sqrt{3N}$

TABLE III: Recommended Fully Homomorphic Encryption parameters w.r.t the security level λ [41].

N	$\log_2 Q(\lambda, N)$		
	$\lambda = 128$ bits	$\lambda = 192$ bits	$\lambda = 256$ bits
$2^{10} = 1024$	27	19	14
$2^{11} = 2048$	54	37	29
$2^{12} = 4096$	109	75	58
$2^{13} = 8192$	218	152	118
$2^{14} = 16384$	438	300	237
$2^{15} = 32768$	881	600	476

noise coming from the encoding and homomorphic operations.

We consider the *precision loss* in RNS-CKKS, which we can define informally as the difference between evaluating a circuit in the clear (true value) and evaluating the same circuit homomorphically (encrypted value). A more formal description is given below.

Definition VI.1. We define the *precision loss* associated with calculating the results of the arithmetic function $f(X, Y)$ homomorphically as the logarithm of the root mean square errors (RMSE);

$$\log_{10}(\text{RMSE}) = \log_{10} \left(\sqrt{\frac{1}{n} \sum_i (\tilde{m}_i - m_i)^2} \right), \quad (14)$$

where $(\tilde{m}_1, \dots, \tilde{m}_{N/2})$ is the output of the homomorphic evaluation of $f(X, Y)$ and $(m_1, \dots, m_{N/2})$ is the output of the true values of the function $f(X, Y)$.

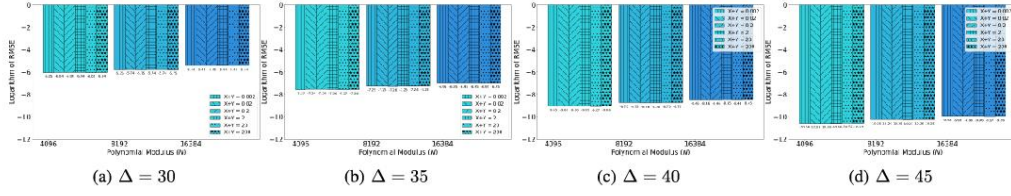
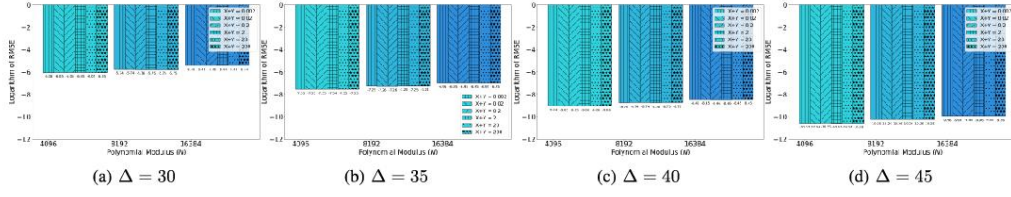
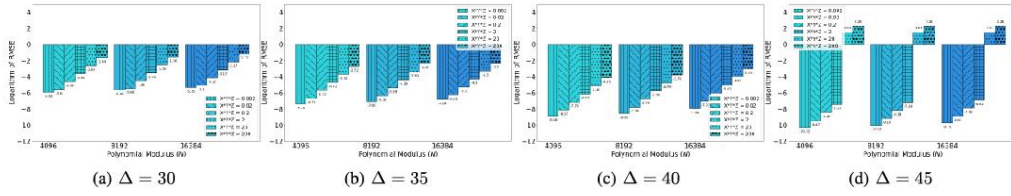
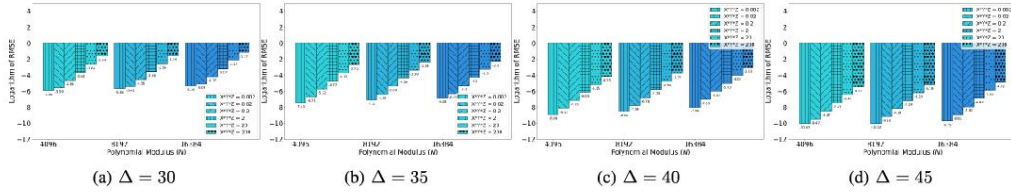
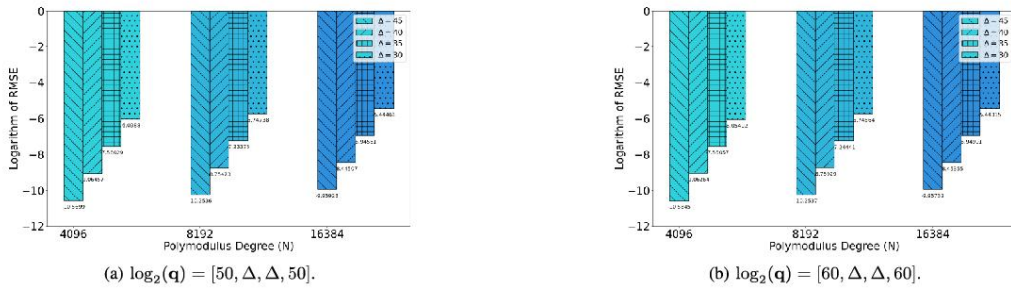
Definition VI.2. We define the *percentage errors* associated with calculating the results of the arithmetic function $f(X, Y)$ homomorphically as the ratio of the absolute difference between the decrypted results and the true values over the true values;

$$\text{Error}(\%) = \frac{\sum_i (|\tilde{m}_i - m_i|)}{\sum_i m_i} \times 100, \quad (15)$$

where $(\tilde{m}_1, \dots, \tilde{m}_{N/2})$ is the output of the homomorphic evaluation of $f(X, Y)$ and $(m_1, \dots, m_{N/2})$ is the output of the true values of the function $f(X, Y)$.

VII. USING CIPHERFLOW

We demonstrate how to use *CipherFlow* through several use cases.

Fig. 3: The precision loss of $AddE(X, Y)$, where $\log_2(\mathbf{q}) = [50, \Delta, 50]$.Fig. 4: The precision loss of $AddE(X, Y)$, where $\log_2(\mathbf{q}) = [60, \Delta, 60]$.Fig. 5: The precision loss of $MultiE(MultiE(X, Y), Z)$, where $\log_2(\mathbf{q}) = [50, \Delta, 50]$.Fig. 6: The precision loss of $MultiE(MultiE(X, Y), Z)$, where $\log_2(\mathbf{q}) = [60, \Delta, 60]$.Fig. 7: The precision loss of $AddE(X, Y)$, where $\Delta = [49, 45, 40, 35, 30]$ for $X + Y = 200$.

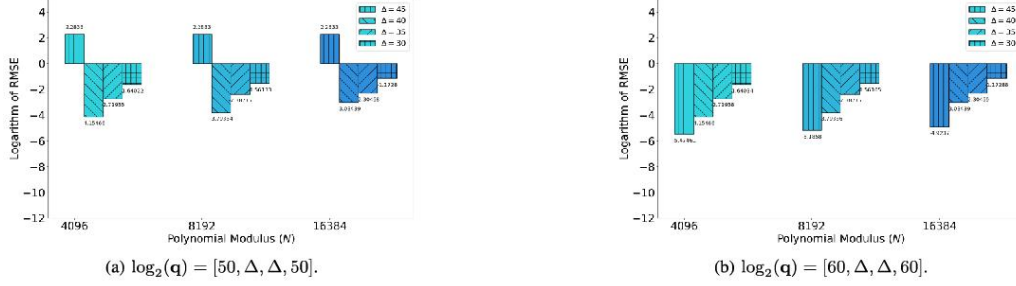


Fig. 8: The precision loss of $MultiE(MultiE(X, Y), Z)$, where $\Delta = [49, 45, 40, 35, 30]$ for $X \times Y \times Z = 200$.

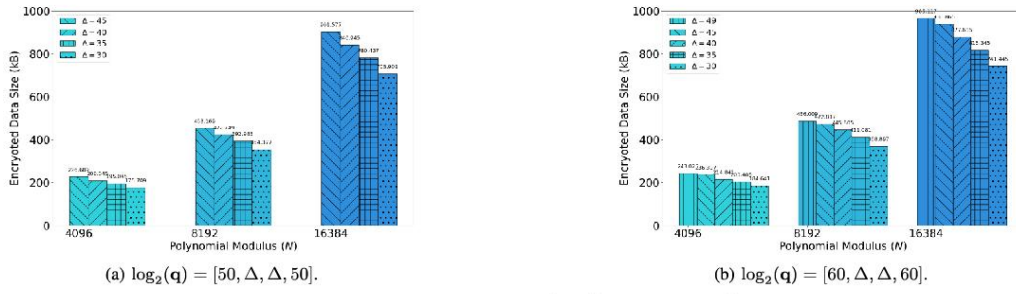


Fig. 9: Encrypted data size for $\Delta = [49, 45, 40, 35, 30]$.

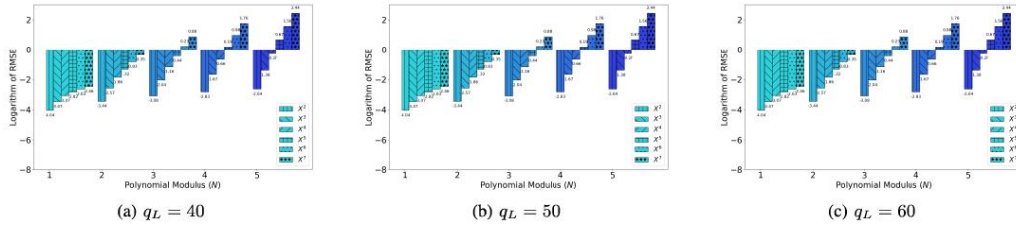


Fig. 10: The precision loss of $Exponent(X, i)$ for $i = 1, 2, \dots, 7$, $\Delta = [30]$, $\log_2(q) = [q_L, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, q_1]$ and $N = 16384$.

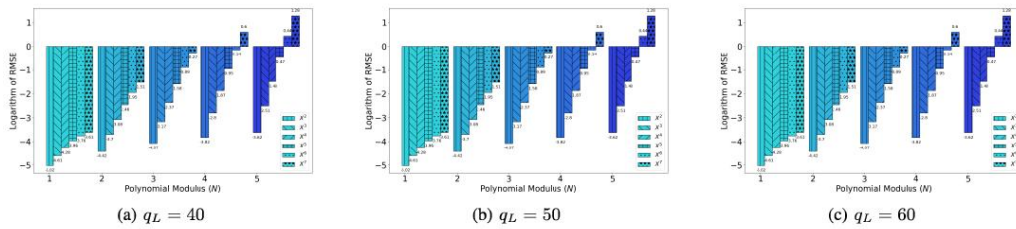
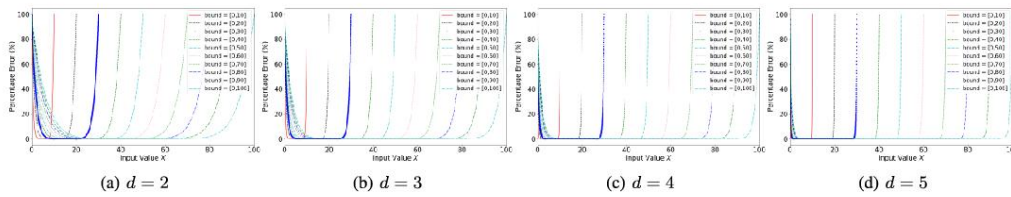
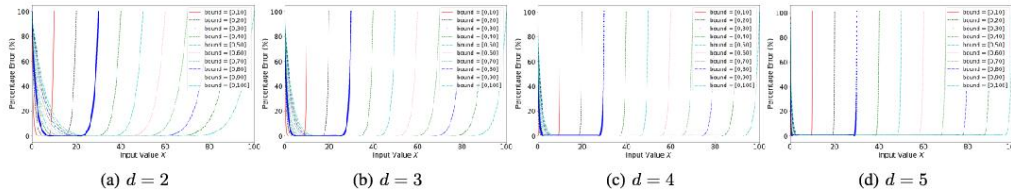
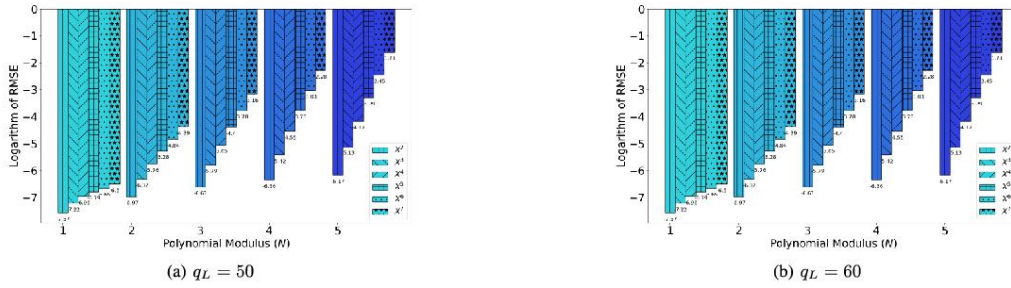
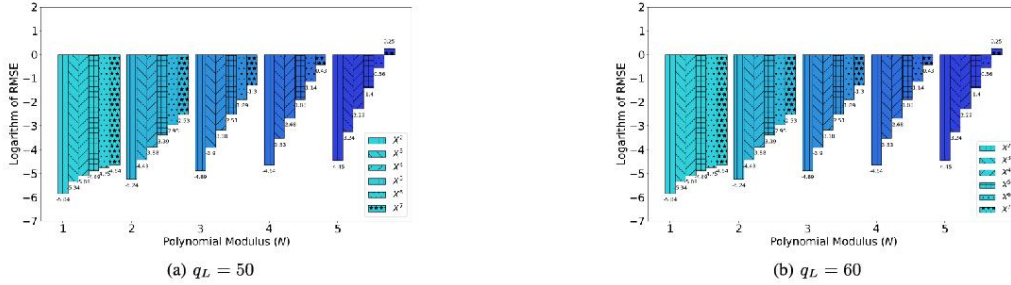


Fig. 11: The precision loss of $Exponent(X, i)$ for $i = 1, 2, \dots, 7$, $\Delta = [35]$, $\log_2(q) = [q_L, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, q_1]$ and $N = 16384$.

A. Getting Started

To start using *CipherFlow*, we first install the *CipherFlow* extension to Node-RED. After the installation succeeded,

the *CipherFlow* nodes are listed in the node pallet. In this playground, we can play and learn how the HE is performed

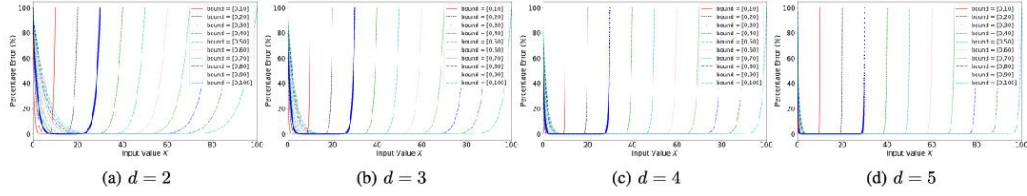
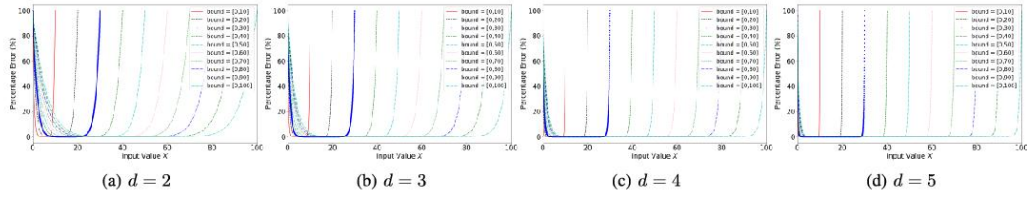
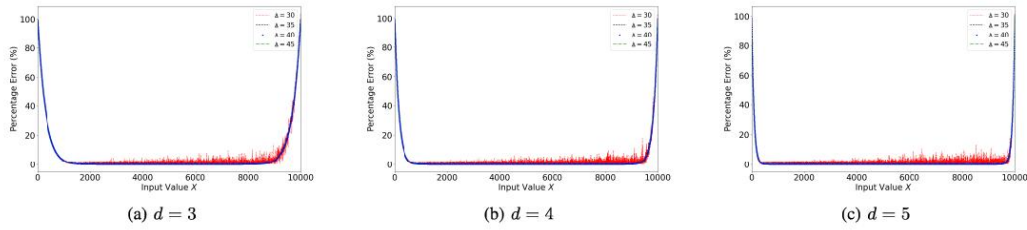
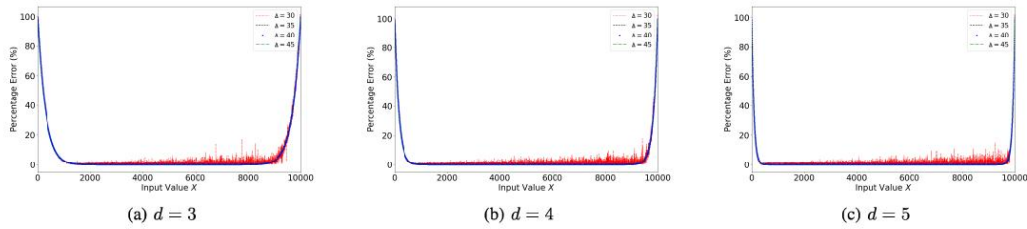
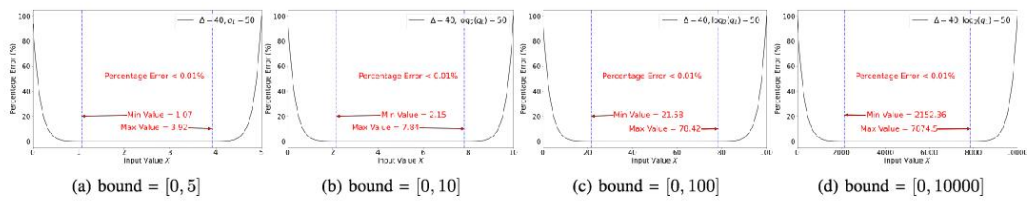


via the node extension.

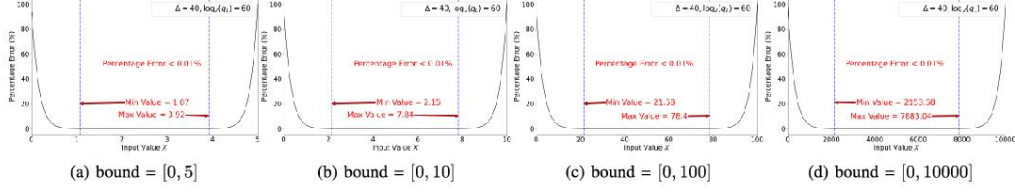
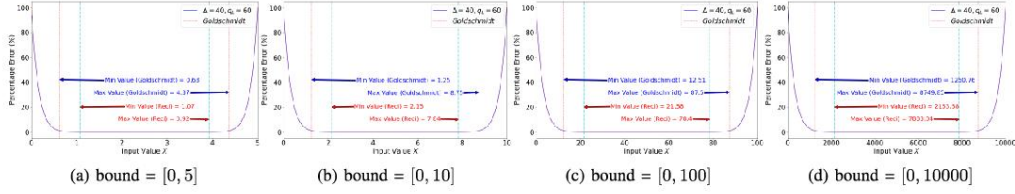
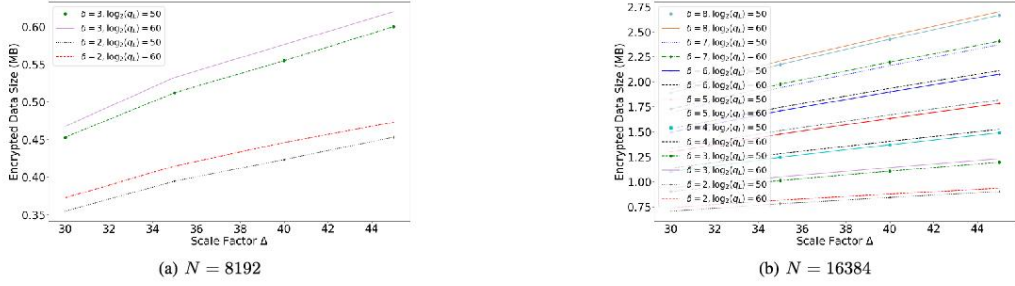
In the *CipherFlow*, developers must generate the CKKS context and keys corresponding to the parameter setting or import them from the prior files. All nodes must be set to the same CKKS context along the same dataflow and tied

with the corresponding keys. In the parameter setting stage, *CipherFlow* allows a developer to configure the following parameters;

context: to set the CKKS parameters: 1) PolyModulus for a

Fig. 16: The percentage errors (%) for $\Delta = [40]$, $\log_2(\mathbf{q}) = [50, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 50]$, $\delta = 8$ and $N = 16384$.Fig. 17: The percentage errors (%) for $\Delta = [45]$, $\log_2(\mathbf{q}) = [50, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 50]$, $\delta = 8$ and $N = 16384$.Fig. 18: The percentage errors (%) for $\Delta = [30, 35, 40, 45]$, $\log_2(\mathbf{q}) = [50, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 50]$, $\delta = 8$ and $N = 16384$.Fig. 19: The percentage errors (%) for $\Delta = [30, 35, 40, 45]$, $\log_2(\mathbf{q}) = [60, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 60]$, $\delta = 8$ and $N = 16384$.Fig. 20: The percentage errors (%) for $\Delta = [40]$, $\log_2(\mathbf{q}) = [50, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 50]$, $\delta = 7$, $d = 3$, and $N = 16384$.

polynomial modulus degree n , 2) CoeffModulus for coefficient modulus chain $\tilde{\mathbf{q}}$, and 3) Scale for a scale factor Δ .

Fig. 21: The percentage errors (%) for $\Delta = [40]$, $\log_2(q) = [60, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 60]$, $\delta = 7$, $d = 3$ and $N = 16384$.Fig. 22: Comparison between the percentage errors (%) of the reciprocal algorithm and the Goldschmidt error bound for $\Delta = [40]$, $\log_2(q) = [60, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, \Delta, 60]$, $\delta = 7$, $d = 3$ and $N = 16384$.Fig. 23: Encrypted data size (MB) for $\Delta = [30, 35, 40, 45]$.TABLE IV: Parameter setting options of *CipherFlow*. (Note: Apart of these options, *CipherFlow* also allows users to customize the parameter setting by themselves.)

Options	N	$\log_2(q)$	Δ	δ
Small	2^{13}	[50, 30, 30, 50]	30	2
Medium	2^{13}	[60, 40, 40, 60]	40	2
Large	2^{14}	[60, 40, 40, 40, 40, 40, 40, 40, 60]	40	7

publicKey: to be selected in the input and encrypt nodes.

secretKey: to be selected in only the decrypt node.

relinKey: to be selected in the input and setContext nodes.

We omit the parameter configuration in the arithmetic operation nodes. The developers merely wire the input to those arithmetic operation nodes, and the corresponding CKKS context obtained from the previous node is automatically injected. The *CipherFlow* requires the developer to set a constant value (referring to a plaintext) for computation in $\text{add}(P)$, and $\text{multi}(P)$.

VIII. PERFORMANCE EVALUATION

Our experiment involves testing nine operations on multiple widely used IoT edge devices and a cloud service. The nine operations are decrypt, encrypt, addEncrypt, addPlain, exponent, multiEncrypt, multiPlain, negate, and reciprocal. We aim to measure memory usage and running time while executing the operations. The IoT edge devices included in the experiment are IoT 2050, Jetson Nano, Raspberry Pi3, and Raspberry Pi4.

The Jetson Nano [42] is a small, powerful, affordable computer designed for AI and machine learning applications. It has a quad-core ARM Cortex-A57 CPU, a 128-core Nvidia Maxwell GPU, and 4GB of RAM. It is easy to use, supporting popular AI frameworks like TensorFlow, PyTorch, and Caffe. It can be used for various applications, such as object detection, robotics, and autonomous systems.

Raspberry Pi 3 and Pi 4 are single-board computers designed for hobbyists, educators, and DIY enthusiasts. The Raspberry Pi boards are famous for their versatility, affordability, and ease of use. They can be used for a wide range

of applications, such as media centers, home automation, and robotics. Raspberry Pi 3 has a 1.2 GHz quad-core ARM Cortex-A53 CPU and 1GB of RAM, and built-in WiFi and Bluetooth connectivity. The Raspberry Pi 4 has a 1.5 GHz quad-core ARM Cortex-A72 CPU, up to 8GB of RAM, dual 4K HDMI outputs, Gigabit Ethernet, and USB 3.0 ports. The Raspberry Pi 4 is more powerful than the Raspberry Pi 3 and can run more demanding applications, such as desktop computing, gaming, and AI.

The IoT2050 is an embedded PC designed for industrial IoT (Internet of Things) applications produced by Siemens. The IoT2050 has an Intel Atom processor, up to 8GB of RAM, and includes various communication interfaces, such as Ethernet, USB, and RS232/485. It is designed to be rugged, reliable, and easily integrated with other devices and systems, such as PLCs (Programmable Logic Controllers) and cloud platforms. The IoT2050 is also compatible with popular software tools and frameworks, such as Node-RED and Eclipse Kura, making it easy to develop and deploy IoT applications.

IX. IIOT USE CASES FOR SMART FACTORIES

IoT is a transformative technology and finds application in the industrial setting in many ways but it's still an evolving area and has its share of challenges. To successfully implement IIoT, it is necessary to identify and prioritize use cases that can benefit the organization. We list some of the leading IIoT use cases below:

- Production visibility
- Supply chain optimization
- Energy consumption management [9].

A. Node Pallet for CipherFlow

The *CipherFlow* introduces 13 new nodes as the Node-RED extension as presented in Table V. We divide them into two groups: 1) HE functionalities and 2) arithmetic operations. The HE functionality nodes provide all necessary CKKS functions, such as encryption, decryption, and setContext. In a playground, developers may focus on only testing their computation flow rather than testing the fed input messages from outside Node-RED. Thus, we provide the input node that embeds the fixed value we want to encrypt and allows the developers to create or select the existing context. The arithmetic operation nodes contain additive and multiplicative operations, such as add(P), add(E), multi(P), and multi(E). Instead of subtraction, the *CipherFlow* offers the negate node to turn a positive value into a negative one, and vice versa. Consequently, we can perform subtraction by combining the negate node and the add(P) or add(E) nodes. The exponent node resembles the multi(E) with the number of exponents. The reciprocal node performs the multiplicative inverse of a ciphertext by Algorithm 6 with fixed $d = 3$.

X. CONCLUSION

This paper proposed *CipherFlow*, a privacy-preserving IoT development playground in Node-RED. This playground was developed under Microsoft SEAL libraries, enabling secured

TABLE V: Node-RED CipherFlow Pallet and Definitions

Operation	Node	Definition
FHE	encrypt	To encrypt the plaintext messages passing through it by applying with public key pk and output ciphertexts.
	decrypt	To decrypt ciphertexts passing through it by applying secret key sk , and output the message in a float number
	setContext	To insert the CKKS context to the arithmetic operation nodes in the same dataflow.
	input	Similar to the encrypt node, but it encrypts the value in the input field instead of the passing message
Arithmetic	add(E)	To add two passing ciphertexts
	add(P)	To add a given value in the input field and the passing ciphertext
	multi(E)	To multiply two passing ciphertexts
	multi(P)	To multiply a given value to the passing ciphertext
	negate	To convert a positive passing ciphertext to negative ciphertext, and vice versa
	exponent	To raise the passing ciphertext to a given power
	reciprocal	To multiplicative inverse a passing ciphertext

computation based on Homomorphic Encryption (HE). Instead of a hard-code writing program in C++, *CipherFlow* offers three pipeline abstractions, i.e., 1) a playground of drag-and-drop HE nodes in Node-RED, 2) arithmetic operators, and 3) optimal parameters setting guidelines that allow IoT developers to create and verify their secured computing functions on dataflows. The development of *CipherFlow* has overcome three main challenges: 1) HE computation, 2) parameter setting, and 3) reciprocal operation. We carefully designed *CipherFlow* and demonstrated the applicability and effectiveness of the *CipherFlow* through various use cases and performance evaluations on the selected scenarios. Our evaluation shows the performance and security trade-off depending on the parameter setting, in which higher security and accuracy require a higher computation resource. Overall, *CipherFlow* is flexible and easy to use, and it will save time to enable new secured IoT applications.

ACKNOWLEDGEMENT

This work was funded and financially supported by NBTC, Thailand, under the grant A63-1-(2)-017 and also supported in part by NICT22401 and JSPS KAKENHI Grant Number JP22K11994, JP21H03442, JP20K23329.

REFERENCES

- [1] C. Li and B. Palanisamy, "Privacy in internet of things: from principles to technologies," *IEEE Internet of Things*, vol. 6, no. 1, pp. 488–505, February 2019.
- [2] P. Kühnreiter, V. Pak, and D. Reinhardt, "A survey on solutions to support developers in privacy-preserving iot development," *Pervasive and Mobile Computing*, vol. 85, no. 3, pp. 101656(1–10), 2022.
- [3] S. Tokas, G. Erdogan, and K. Stølen, "Privacy-aware iot: State-of-the-art and challenges," in *In Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, 2023, pp. 450–461.

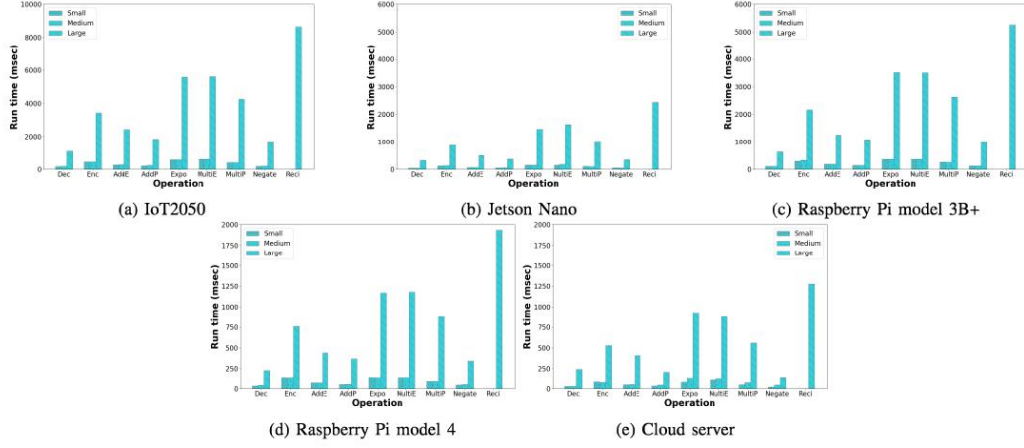


Fig. 24: Run time (msec).

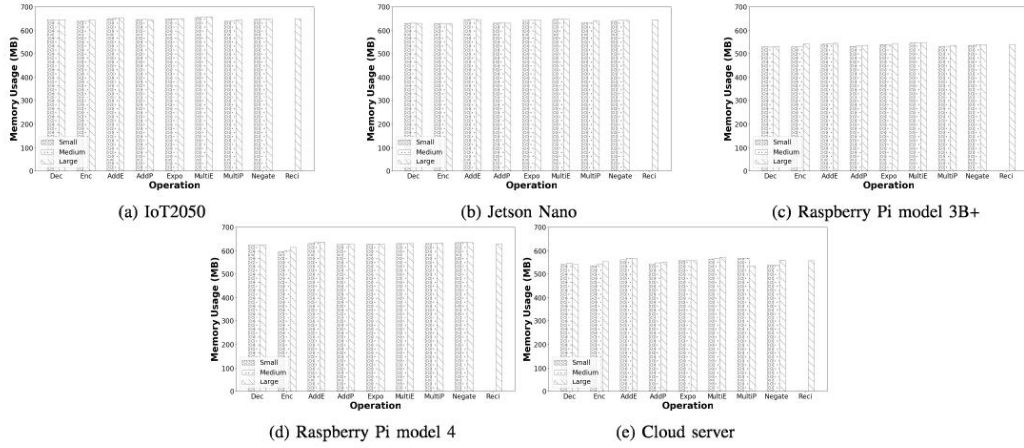


Fig. 25: Memory usage (MB).

- [4] "Emerging privacy enhancing technologies: Current regulatory and policy approaches."
- [5] D. Natarajan and W. Dai, "Seal-embedded: A homomorphic encryption library for the internet of things," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, no. 3, pp. 756–779, Jan. 2021.
- [6] W. John and et. al. (March 2018) Microsoft seal. [Online]. Available: <https://github.com/microsoft/SEAL>
- [7] A. Nick. (March 2020) Morfix: Homomorphic encryption as a service. [Online]. Available: <https://github.com/morfix-io>
- [8] A. Viand, P. Jattke, and A. Hithnawi, "Sok: Fully homomorphic encryption compilers," in *the 1st ACM SIGSOFT International Workshop on Ensemble-Based Software Engineering*, 2021, pp. 1–8.
- [9] A. Lawrence, P. Thollander, and M. Karlsson, "Specific energy consumption/use (sec) in energy management for improving energy efficiency in industry: Meaning, usage and differences," *MDPI Energies*, vol. 12(2), no. 247, pp. 1–22, January 2019.
- [10] J. Foundation. (March 2016) Node-red, flow-based programming for the internet of things. [Online]. Available: <https://nodered.org/>
- [11] M. Ercegovic and et al., "Improving goldschmidt division, square root and square root reciprocal," INRIA, LIP, Research Report inria-00072909, 1999.
- [12] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds. Cham: Springer International Publishing, 2017, pp. 409–437.
- [13] R. Dathathri and et al.
- [14] F. Pallas, J. Legler, N. Amsgruber, and E. Grönewald, "Redcastle: practically applicable ks-anonymity for iot streaming data at the edge in node-red," in *M4IoT '21*, 2021, p. 8–13.
- [15] M. Blackstock and R. Lea, "Fred: A hosted data flow platform for the iot," in *MOTA '16*, 2016, pp. 1–5.
- [16] J. P. Dia, A. Restivo, and H. S. Ferreira, "Empowering visual internet-of-things mashups with self-healing capabilities," in *IEEE/ACM SERP4IoT*, 2021, pp. 1–8.
- [17] J. Noor and et al., "Ddflow: visualized declarative programming for heterogeneous iot networks," in *IoTDI '19*, 2019, pp. 172–177.
- [18] J. Kawamoto, "An implementation of privacy preserving stream integra-

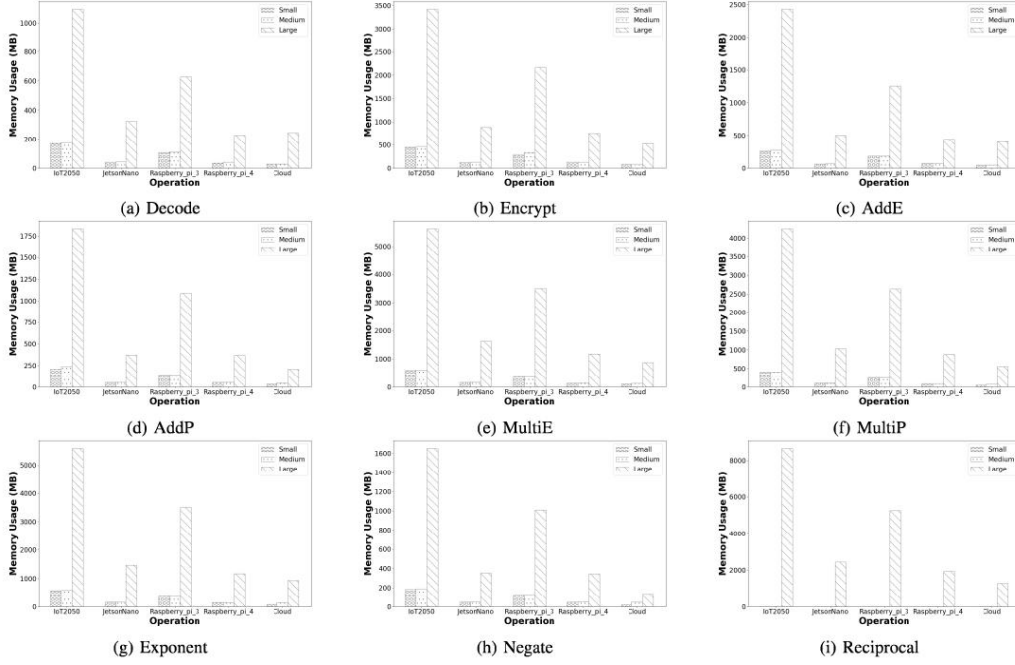


Fig. 26: Run time (msec).

- tion system,” in *AIChallengesIoT'19*, 2016, pp. 32–38.
- [19] V. Rastogi and S. Nath, “Differentially private aggregation of distributed time-series with transformation and encryption,” in *ACM SIGMOD*, 2010, pp. 735–746.
- [20] D. Cousins, K. Rohloff, and Y. Polyakov. (2014) Palisade homomorphic encryption software library: An open-source lattice crypto software library. [Online]. Available: <https://palisade-crypto.org/>
- [21] S. Halevi and V. Shoup. (2014) Helib implementing homomorphic encryption. [Online]. Available: <https://homenc.github.io/HElib/index.html>
- [22] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(leveled) fully homomorphic encryption without bootstrapping,” *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, Jan. 2014.
- [23] —, “Fully homomorphic encryption without bootstrapping,” in *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, 2012, pp. 309–325.
- [24] Lee, H. Yongwoo, C. Seonyeong, J. Seonyoung, K. Shinnung, K. Changsu, L. Eunkyung, K. Dongyoon, and Hanjun, “Hecate: Performance-aware scale optimization for homomorphic encryption compiler,” in *Proceedings of the 2022 IEEE/ACM International Symposium on Code Generation and Optimization (CGO 2022)*, 2022, pp. 193–204.
- [25] A. Viand, P. Jattke, M. Haller, and A. Hithnawi, “Heco: Fully homomorphic encryption compiler,” in *WAHC '21*, 2023, pp. 11–16.
- [26] Y. Lee, S. Cheon, D. Kim, D. Lee, and H. Kim, “ELASM: Error-Latency-Aware scale management for fully homomorphic encryption,” in *32nd USENIX Security Symposium (USENIX Security 23)*, Anaheim, CA: USENIX Association, Aug. 2023, pp. 4697–4714. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity23/presentation/lee-yongwoo>
- [27] A. Ibarrondo and A. Viand, “Pyfhel: Python for homomorphic encryption libraries,” in *WAHC '21*, 2021, pp. 11–16.
- [28] H. Chen, R. Cammarota, F. Valencia, F. Regazzoni, and F. Koushanfar, “Ahec: end-to-end compiler framework for privacy-preserving machine learning acceleration,” in *Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference (DAC '20)*. ACM/EDAC/IEEE, 2020, pp. 1–6.
- [29] A. Viand and H. Shafagh, “Marble: Making fully homomorphic encryption accessible to all,” in *Proceedings of the 6th Workshop on Encrypted Computing Applied Homomorphic Cryptography (WAHC '18)*. ACM, 2018, pp. 49–60.
- [30] A. A. Badawi and et al., “Openfhe: Open-source fully homomorphic encryption library,”
- [31] C. V. Mouchet, J.-P. Bossuat, J. R. Troncoso-Pastoriza, and J.-P. Hubaux, “Lattigo: A multiparty homomorphic encryption library in go,” in *Proceedings of the 8th ACM Workshop on Encrypted Computing Applied Homomorphic Cryptography (WAHC'20)*, 2020, pp. 64–70.
- [32] S. Shenand, H. Yang, Y. Liu, Z. Liu, and Y. Zhao, “Carm: Cuda-accelerated rms multiplication in word-wise homomorphic encryption schemes for internet of things,” *IEEE Transactions on Computers*, vol. 72, no. 7, pp. 1999–2010, July 2023.
- [33] S. Chowdhary, W. Dai, K. Laine, and O. Saarikivi, “Eva improved: Compiler and extension library for ckks,” in *Proceedings of the 9th on Workshop on Encrypted Computing Applied Homomorphic Cryptography (WAHC '21)*, 2021, pp. 43–55.
- [34] G. S. Çetin, Y. Doröz, B. Sunar, and W. J. Martin, “Arithmetic using word-wise homomorphic encryption,” 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:17174996>
- [35] M. Green and J. Hopkins, “Developers are not the enemy! the need for usable security apis,” *IEEE Security & Privacy*, vol. 14, no. 5, pp. 40–46, 2016.
- [36] N. Patnaik, J. Hallett, and A. Rashid, “Usability smells: an analysis of developers’ struggle with crypto libraries,”
- [37] Crockett, P. Eric, S. Chris, and Chad, “Alchamy: A language and compiler for homomorphic encryption made easy,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, 2018, pp. 1020–1037.
- [38] I. Hwang1, J. Seo1, and Y. Song1, “Optimizing he operations via level-aware key-switching framework,” in *Proceedings of the 11th ACM Work-*

- shop on Encrypted Computing Applied Homomorphic Cryptography (WAHC'23)*, 2023, pp. 59–67.
- [39] J. Cheon, D. Kim, D. Kim, H. Lee, and K. Lee, "Numerical method for comparison on homomorphically encrypted numbers," in *Advances in Cryptology – ASIACRYPT 2019*, 2019, pp. 415–445.
- [40] I. Syafalni, D. M. Reynaldi, R. Munir, T. Adiono, N. Sutisna, and R. Mulyawan, "Complexity analysis of encoding in ckks-fully homomorphic encryption algorithm," in *Proc. of International Symposium on Electronics and Smart Devices (ISESD)*, 2022, pp. 1–6.
- [41] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Homomorphic encryption standard," Cryptology ePrint Archive, Paper 2019/939, 2019. [Online]. Available: <https://eprint.iacr.org/2019/939>
- [42] Jetson nano. [Online]. Available: <https://developer.nvidia.com/embedded/jetson-nano>

ภาคผนวก จ.
เอกสารตอบรับบทความ

1. แบบตอบรับการนำส่งบทความ NBTC Journal

[NBTC_Journal] Submission Acknowledgement

M

Ms. Sarothorn Puspakom via Thai Journals Online (ThaiJO) <admin@tci-thaijo.org>
To: Kalika Suksomboon
Tue 3/19/2024 5:09 PM

Kalika Suksomboon:

Thank you for submitting the manuscript, "Privacy as a Service for IoT Platform" to NBTC Journal. With the online journal management system that we are using, you will be able to track its progress through the editorial process by logging in to the journal web site:

Submission URL: https://so04.tci-thaijo.org/index.php/NBTC_Journal/authorDashboard/submission/271384
Username: kalika.suk@nectec.or.th

If you have any questions, please contact me. Thank you for considering this journal as a venue for your work.

Ms. Sarothorn Puspakom

Journal Editors

Office of The National Broadcasting and Telecommunications Commission
87 Phaholythin 8 (Soi Sailom), Samsen Nai, Phayathai, Bangkok Thailand 10400
Tel : +66 271 0151 - 60, +66 2670 8888

Completed.

Thank you!

Submitted.

Reply

Forward

NBTC Journal



[Back to Submissions](#)

271384 / Kalika Suksomboon et al. / Privacy as a Service for IoT Platform

Library

Workflow

Publication

Submission

Submission Files

Q Search

	1092507 Journal_CYBLION_submission_16pages.docx	19 March 2024	Article Text
	1092514 แบบนำส่งบทความ+2567_kalika.pdf	19 March 2024	Other

Download All Files

Pre-Review Discussions

Add discussion

Name	From	Last Reply	Replies	Closed
Comments for the Editor	kalika.suk@nectec.or.th	-	0	☐
19-03-2024 04:37 PM				

2. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ AINTEC 2023

[AINTEC 2023 poster] Accepted submission #7 "Protecting Privacy in Industrial IoT..."

You forwarded this message on Wed 11/8/2023 3:52 PM

aintec2023-poster-submission@cfp.wide.ad.jp
To: Kalika Suksomboon

Tue 10/31/2023 10:31 PM

Dear author,

The **AINTEC 2023** poster program committee is delighted to inform you that your submission #7 has been accepted to appear in the conference.

* Title: Protecting Privacy in Industrial IoT with a Homomorphic Encryption Platform While Enjoying Industry 4.0
* Authors: Kalika Suksomboon (NECTEC);
Nataset Tanabodee (Kasetsart University);
Eakarat Saktawornlerd (Kasetsart University);
Aimaschana Niruntasukrat (NECTEC);
Chaveee Issariyapat (NEXPIE Co. Ltd.);
Sophon Mongkoluksamee (Srinakharinwirot University);
Natapon Tansangworn (NECTEC);
Sukumal Kitisin (Kasetsart University)
* Site:
https://linkprotect.cudasvc.com/url? a=https%3a%2f%2fcp.wide.ad.jp%2fconferences%2fainte2023poster%2fpaper%2f7%3fcap%3dhcav7xUypEboKAEKuGnASXXTfmdMC&c=E.1.ALcJ6aTvPzKjnP3i8Dgx_VFUeIMVnpb00tF4E-V-M51pzZ5f6jlgXyYeqn_lqjVEaaU0MxtqGsyFPmxMJ_DqCfql9MiQ3FhjluovRWpVba0qBsszkbwEK3Hei8.&typo=1

Visit the submission site for reviews, comments, and related information. Reviews and comments are also included below.

Contact **AINTEC 2023** poster submission <poster-**ainte**c@interlab.ait.ac.th> with any questions or concerns.

- **AINTEC 2023** poster Submissions

3. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ AINTEC 2022

AINTEC22: Additional Information Required from Authors

Pavel Farhan <pavelfarhan@ait.asia>
To: nataset.t@ku.th; Kalika Suksomboon; i@chavee.com; sophon@g.swu.ac.th; Aimaschana Niruntasukrat; +2 others
Cc: Rey Padilla <rpadilla@ait.asia>

Thu 11/3/2022 11:05 AM

Dear Authors,

I hope this email finds you well.

Congratulations again on your accepted paper! Before we move on to the next step, there is some additional information that is required from you.

Please find the shared google docs link below:

https://docs.google.com/spreadsheets/d/1GBsNVF_eEpMLZ6tW1FNBZjvqGMYxyb3I/edit?usp=sharing&oid=112887607925720849797&rtpof=true&sd=true

Kindly fill in the missing information as instructed. It's important for you to provide us with the ORCID which can be set up at orcid.org/register.

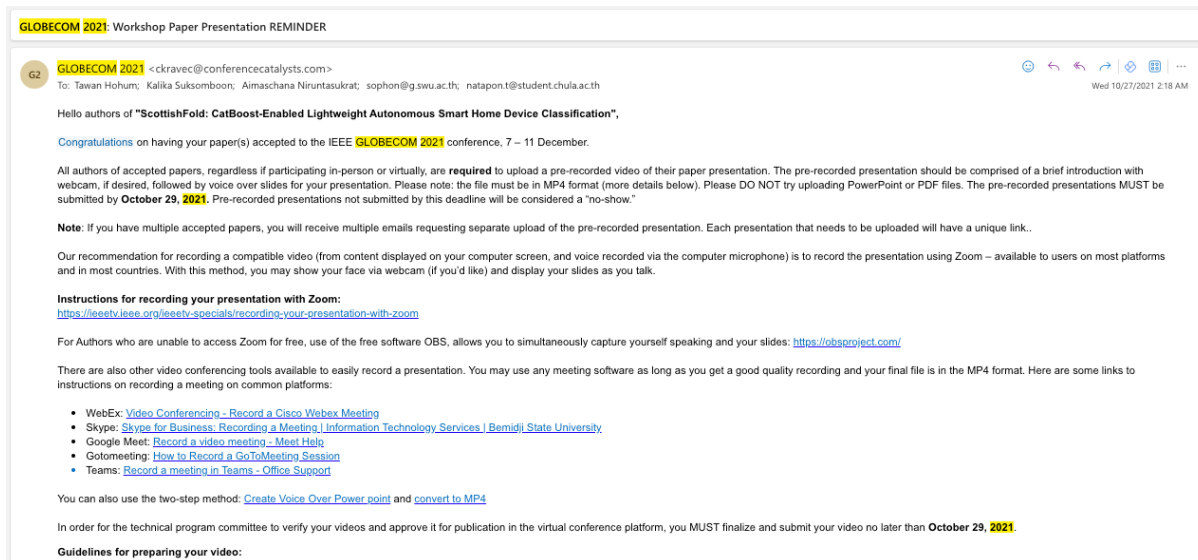
If you think that some of the information which has been pre-filled is not correct, then feel free to update that information as well.

Once we have all the information required from the authors, we will proceed to give you more details and guidelines regarding the Copyright form.

Thank you for your understanding and cooperation.

Warm regards,
Pavel Farhan
AINTEC2022 Secretariat

4. แบบตอบรับการตีพิมพ์บทความในงานประชุมวิชาการ IEEE GLOBECOM 2021



ประวัตินักวิจัย

ประวัติบุคลากร



ชื่อ - นามสกุล : ดร.พินิตา พงษ์ไพบูลย์

ที่อยู่ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

เลขที่ 112 อุทยานวิทยาศาสตร์ประเทศไทย ถ.พหลโยธิน ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120

สถานที่ทำงาน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ประวัติการศึกษา

ปริญญาเอก Electrical and Computer Engineering, Carnegie Mellon University

ปริญญาโท Electrical Engineering, Stanford University

ปริญญาตรี Electrical Engineering, Stanford University (with distinction)

ประสบการณ์การทำงาน

2561 – ปัจจุบัน รองผู้อำนวยการ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

2561 นักวิจัย กลุ่มวิจัยเอดจ์คอมพิวเตอร์สำหรับเชื่อมต่อสรรพสิ่ง ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

- วิจัยพัฒนาอุปกรณ์เอดจ์คอมพิวเตอร์สำหรับงาน Internet of Things และ Smart Factory

2559 – 2561 หัวหน้ากลุ่มวิจัยเพื่อทดลองให้บริการ NETPIE ในเชิงพาณิชย์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

- พัฒนาแผนธุรกิจสำหรับการให้บริการ คลาวด์แพลตฟอร์มสำหรับการพัฒนา Internet of Things
- สร้างพันธมิตร และขยายฐานการใช้งาน Internet of Things ภายในประเทศ

2551 – 2559 นักวิจัย ห้องปฏิบัติการวิจัยนวัตกรรมอินเทอร์เน็ต ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

- หัวหน้าโครงการวิจัยและพัฒนาแพลตฟอร์ม Internet of Things
- Lead a project that develops a suite of intelligent network management software such as network monitoring, traffic measurement, and bandwidth controller.
- Draft national ICT policies: IPv6 action plan (2013-2015), IPv6 action plan 2 (2016-2019), and digital economy strategy.
- Research and develop solution and applications for IPv6 transition such as IPv6 tunnel broker, IPv6 outlet controller.
- Drive IPv6 awareness and deployment in Thailand in collaboration with Ministry of ICT and IPv6 Forum Thailand.

2547 – 2550 นักวิจัย ห้องปฏิบัติการวิจัยเทคโนโลยีเครือข่าย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

- Promote IPv6 deployment in Thailand in collaboration with IPv6 of Forum Thailand.
- R&D responsibility in the fields of IPv6, mobile IP, traffic classification, network monitoring, and intelligent systems.

ความเชี่ยวชาญเฉพาะทาง

Cyber-Physical Systems, Internet of Things, Computer Networks, Network Security, Internet Protocols, Network Management, Optical Networks

ผลงานที่ผ่านมา

- ที่ปรึกษาบริษัทเน็กซ์พายจำกัด (ผู้ให้บริการ IoT Cloud Platform เชิงพาณิชย์ด้วยเทคโนโลยี NETPIE) 2018
- ช่วยราชการสำนักงานเลขาธิการคณะกรรมการเตรียมการดิจิทัลเพื่อเศรษฐกิจและสังคม กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร 2014-2015
- ผู้จัดการโครงการจ้างที่ปรึกษาเพื่อช่วยบริหารจัดการและดำเนินงานของศูนย์ประสานงานและปฏิบัติการ IPv6 กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร 2014-2015
- ผู้จัดการโครงการที่ปรึกษาจัดทำแผนปฏิบัติการเพื่อผลักดัน ส่งเสริม เร่งรัดและติดตามผลการดำเนินงาน IPv6 ในประเทศไทย กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร 2012-2013
- คณะทำงานพิจารณาบทวนเลขหมายโทรคมนาคม สำนักงานกิจการกระจายเสียง กิจการโทรทัศน์และกิจการโทรคมนาคมแห่งชาติ 2012
- ที่ปรึกษาโครงการการศึกษาและสำรวจความพร้อม เพื่อประกาศแผนการปรับเปลี่ยนจากโครงข่ายดั้งเดิมเป็นโครงข่ายยุคหน้าแห่งชาติ (Next Generation Network) สำนักงานคณะกรรมการกิจการโทรคมนาคมแห่งชาติ 2011 – 2012
- คณะกรรมการส่งเสริมนโยบาย IPv6 c และคณะทำงานส่งเสริมนโยบาย IPv6 กระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร 2006 – ปัจจุบัน
- อาจารย์พิเศษ คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล 2008 – ปัจจุบัน
- ที่ปรึกษาด้าน Network และ IT Security บริษัท บัตรกรุงไทย จำกัด (มหาชน) 2007
- อาจารย์พิเศษ วิศวกรรมศาสตร์ มหาวิทยาลัยรังสิต 2005-2006
- กรรมการและผู้ร่วมก่อตั้ง สมาคมไอพีวี 6 ประเทศไทย 2005-ปัจจุบัน
- Adjunct Faculty in the Internet Education & Research Laboratory, Asian Institute of Technology 2006-2007

Journal Publications

1. Suksomboon, K., Pongpaibool, P., Ji, Y., and Aswakul, C., "PC-Nash: QoS provisioning framework with Path-Classification scheme under Nash equilibrium," The Computer Journal, Vol. 54, No. 6, June 2011, pp. 931-943. First published online: December 7, 2010. Doi: 10.1093/comjnl/bxq084.

2. N. Sukityarn, P. Tangamchit, C. Issariyapat, and P. Pongpaibool, "Web Server Anomaly Detection using Principle Component Analysis", NECTEC Technical Journal, NECTEC-ACE2010 Special Edition, September 2010.
3. W. Sunrat, P. Pongpaibool, and K. Meesublak, "สถานภาพและความพร้อมใช้งานคอมพิวเตอร์และอินเทอร์เน็ตในโรงเรียน" in NECTEC Technical Journal, July 2008, pp.118-124.
4. Panita Pongpaibool and Hyong S. Kim, "Providing End-to-End Service Level Agreements across Multiple ISP Networks," Computer Networks Special Issue in Internet Economics, Vol. 46, No. 1, September 2004, pp. 3-18.

Conference Publications (10 years)

1. W. Viriyavit, V. Sornlertlamvanich, W. Kongprawechnon, P. Pongpaibool and T. Isshiki, "Neural network based bed posture classification enhanced by Bayesian approach," 2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), Chonburi, 2017, pp. 1-5.
2. W. Viriyavit, V. Sornlertlamvanich, W. Kongprawechnon, and P. Pongpaibool, "Bed Posture Classification using Noninvasive Bed Sensors for Elderly Care," The Proceedings of the 27th International Conference on Information Modeling and Knowledge Bases(EJC 2017), Krabi, 2017, pp. 396-407.
3. P. Pongpaibool, A. Nirantasukrat, C. Issariyapat, K. Meesublak, C. Mokarat, P. Aiumsupugkul, " NetHAM- nano: A Robust and Scalable Service- Oriented Platform for Distributed Monitoring," in Proceedings of AINTEC 2014, Chiang Mai, Thailand, November 2014.
4. Chuangchunsong, N. , Kamolphiwong, S. , Kamolphiwong, T. , Elz, R. , Pongpaibool, P. , " Performance Evaluation of IPv4/ IPv6 Transition Mechanisms: IPv4- in- IPv6 Tunneling Techniques," International Conference on Information Networking (ICOIN), 2014, Phuket, Thailand, pp. 238-243.
5. P. Sornlertlamvanich, S. Kamolphiwong, R. Elz, and P. Pongpaibool, "NEMO-Based Distributed Mobility Management," in Proceedings of 26th International Conference on Advanced Information Networking and Applications Workshops, Fukuoka-shi, Japan, March 26-29, 2012, pp.645-650.
6. S. Mongkolluksame, C. Issariyapat, P. Pongpaibool, K. Meesublak, N. Nulong, and S. Pukkawanna, "A Management System for Software Package Distribution," in Proceedings of PICMET 2012, Vancouver, Canada, July 2012.
7. C. Issariyapat, P. Pongpaibool and S. Mongkolluksamee, and K. Meesublak, "Using Nagios as a Groundwork for Developing a Better Network Monitoring System," in Proceedings of PICMET 2012, Vancouver, Canada, July 2012.

8. S. Mongkolluksamee, K. Fukuda, and P. Pongpaibool, "Counting NATted Hosts by Observing TCP/IP Field Behaviors," in Proceedings of ICC 2012, Ottawa, Canada, June 2012.
9. P. Chokarpa, S. Mongkolluksamee, and P. Pongpaibool, "การวัด End-to-End Available Bandwidth โดยการตรวจจับสถานะ TCP," in Proceedings of Joint Conference on Computer Science (JCSSE 2011), Thailand, May 2011.
10. V. Visoottiviset, C. Yanprasop, P. Pongpaibool, "DAA: Distributed Address Auto-configuration for Mobile Ad Hoc Networks", in Proceedings of the 1st Workshop on Ubiquitous Computing for Symbio-Information Processing (UCSIP) held in conjunction with 2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE), 11-13 May 2011, Thailand.
11. K. Suksomboon, Y. Ji, P. Pongpaibool, and C. Aswakul, "Load-Balanced Path Provisioning for Guaranteeing End- to- End QoS in Inter- Domain Networks," in Proceedings of 10th International Symposium on Communications and Information Technologies (ISCIT 2010), Tokyo, Japan, October 2010.
12. C. Yanprasop, V. Visoottiviset, and P. Pongpaibool, "Implementation of Fast IP Address Auto-configuration for Mobile Ad Hoc Networks," in Proceedings of JCSSE 2010, Bangkok, Thailand, May 2010.
13. S. Mongkolluksamee, C. Issariyapat, and P. Pongpaibool, "Strengths and Limitations of Nagios as a Network Monitoring Solution," in Proceedings of JCSSE 2010, Bangkok, Thailand, May 2010.
14. S. Mongkolluksamee, C. Issariyapat, and P. Pongpaibool, "การแจ้งเตือนปัญหาในระบบเครือข่ายแบบบูรณาการ," in Proceedings of ECTI-CARD 2010, May 2010.
15. S. Porntaweechoaksakul, S. Kitisin, and P. Pongpaibool, "การศึกษาการใช้งานจริงของไอพีแอดเดรสที่มีการจัดสรรภายในมหาวิทยาลัยเกษตรศาสตร์," in Proceedings of NCSEC 2009, November 2009.
16. C. Chanin and P. Pongpaibool, "IPv6 Tunnel Broker: ระบบอำนวยความสะดวกในการเชื่อมต่อเครือข่าย IPv6 สำหรับผู้ใช้ตามบ้าน," in Proceedings of ECTI-CARD 2009, May 2009.
17. P. Pongpaibool, S. Mongkolluksamee, and S. Pukkawanna, "NtopViewer ซอฟต์แวร์ตรวจสอบการใช้งานอินเทอร์เน็ตในโรงเรียน," in Proceedings of ECTI-CARD 2009, May 2009.
18. S. Pukkawanna, T. Kongpool, and P. Pongpaibool, "BaCon: A User-Friendly Bandwidth Controller System," in Proceedings of JCSSE 2009, Phuket, Thailand, May 2009.
19. S.Pukkawanna, P.Pongpaibool, and V.Visoottiviset, "LD2: A System for Lightweight Detection of Denial-of-Service Attacks," in Proceedings of IEEE MILCOM 2008, San Diego, CA, USA, November 2008.
20. P. Pongpaibool, K. Siriwong Na Ayutaya, K. Kanchanasut, and H. Tazaki, "Rapid IPv6 Address Autoconfiguration for Heterogeneous Mobile Technologies," in Proceedings of the 8th

International Conference of ITS Telecommunications (ITST 2008), Phuket, Thailand, October 2008.

21. P. Sotthivirat, P. Pongpaibool, S. Kitisin, and C. Srisathapornphat, "Fast and Robust Duplicate Address Detection for Mobile IPv6," in Proceedings of the 8th International Conference of ITS Telecommunications (ITST 2008), Phuket, Thailand, October 2008.
22. C. Chanin, P. Pongpaibool, and K. Chanchio, "Outbound link recommender on BGP Multihomed Ases," in Proceedings of International Joint Conference on Computer Science and Software Engineering (JCSSE2008), Kanchanaburi, Thailand, May 2008.
23. K. Suksomboon, P. Pongpaibool, and C. Asawakul, "Game-theoretic approach to prevent selfish path provisioning in interdomain networks," in Proceedings of ECTI-CON 2008, Krabi, Thailand, May 2008.
24. K. Suksomboon, P. Pongpaibool, and C. Asawakul, "An Equilibrium Policy for Providing End-to-end Service Level Agreements," in Proceedings of IEEE WCNC 2008, Las Vegas, USA, March 2008.

Patents

1. ระบบตรวจสอบและอนุญาตการเข้าถึงทรัพยากรระหว่างอุปกรณ์ผ่านเครือข่ายข้อมูลและวิธีการดังกล่าว เลขที่คำขอ 1601003026 วันที่ยื่นคำขอ 26/5/2559
2. ระบบให้บริการเชื่อมต่อสรรพสิ่ง เลขที่คำขอ 1501005546 วันที่ยื่นคำขอ 15/9/2558
3. วิธีการอนุมานผังการเชื่อมต่อระหว่างอุปกรณ์ในเครือข่ายคอมพิวเตอร์ที่สนใจโดยการวิเคราะห์ความสัมพันธ์ของปริมาณข้อมูล เลขที่คำขอ 1201003186 วันที่ยื่นคำขอ 26/6/2555
4. วิธีการสำหรับการตรวจจับความผิดปกติในเครือข่ายคอมพิวเตอร์โดยการคัดกรองสารสนเทศเครือข่ายด้วยโปรโตคอลเอสเอ็นเอ็มพีแบบป้อนกลับ เลขที่คำขอ 1201004884 วันที่ยื่นคำขอ 21/9/2555
5. วิธีการตรวจนับจำนวนเครื่องคอมพิวเตอร์ในเครือข่ายที่อยู่ภายใต้อุปกรณ์แปลงแอดเดรสเครือข่าย (Network Address Translation: NAT) ด้วยการใช้พฤติกรรมของข้อมูลใน TCP/IP โปรโตคอล เลขที่คำขอ 1201003307 วันที่ยื่นคำขอ 2/7/2555
6. วิธีการตรวจสอบการทำงานของเครือข่ายโดยการสร้างโพรไฟล์การทำงานของอุปกรณ์เครือข่าย เลขที่คำขอ 1101001597 วันที่ยื่นคำขอ 11/8/2554

Awards and Grants

- รางวัลระดับดี สาขาเทคโนโลยีสารสนเทศและนิเทศศาสตร์ ในงานวันนักประดิษฐ์ 2562 ผลงานแพลตฟอร์มสื่อสารเพื่อเชื่อมต่อสรรพสิ่ง

- รางวัลรองชนะเลิศ ประเภท Internet of Things ในงาน Thailand ICT Award 2018 ผลงานแพลตฟอร์มสื่อสารเพื่อเชื่อมต่อสรรพสิ่ง
- รางวัลชมเชยประเภท Telecom Application ในงานประกวด Telecom Innovation 2009 จัดโดยสำนักงานคณะกรรมการกิจการโทรคมนาคมแห่งชาติ สำหรับซอฟต์แวร์ดูแลสุขภาพเครือข่าย NetHAM
- Research grants from Pan-Asia ICT R&D Grants Programme 2005-2007 for the project “IPv6 Tunnel Broker: a Key for Using Next Generation Internet in Developing Countries”
- Thai Government Scholarship (Ministry of Science and Technology) 1992-2004.

รับรองสำเนาถูกต้อง



พนิตา พงษ์ไพบูลย์

ประวัติบุคลากรในโครงการ



ชื่อ – นามสกุล : ดร.กัลลิกา สุขสมบุญ

ตำแหน่งงาน : นักวิจัย

ที่อยู่ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ เลขที่ 112 อุทยานวิทยาศาสตร์ประเทศไทย ถ.พหลโยธิน ต. คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120

สถานที่ทำงาน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ประวัติการศึกษา

- ปริญญาเอก Electrical Engineering, Chulalongkorn University
- ปริญญาโท Electrical Engineering, Chulalongkorn University
- ปริญญาตรี Telecommunications Engineering, KMIT'L (2nd class honor)

ประสบการณ์การทำงาน (Researcher)

- ตุลาคม 2562 - ปัจจุบัน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
 - นักวิจัยศูนย์ทดสอบความปลอดภัยทางไซเบอร์สำหรับระบบไซเบอร์-กายภาพ
- มกราคม 2562 - ตุลาคม 2562 LeapMind Inc., Tokyo, Japan
 - Research for AI & Deep Learning for Edge Computing
- เมษายน 2560 - ธันวาคม 2561 KDDI Research Inc., Saitama, Japan
 - ICN2020(EU-JPHorizon2020) Project: Advancing ICN Towards Real-World Deployment Through Research, Innovative Applications and Global Scale Experimentation
 - Platform/network protocol design for access control security management and privacy preserving data outsourcing (with fully homomorphic encryption and broadcast encryption) for ICN-based IoT services.
- มกราคม 2557 - เมษายน 2560 KDDI Research Inc., Saitama, Japan
 - SDN/NFV project: Resource allocation and virtual network function orchestration in a core network
 - Develop the performance modeling of virtual network functions in SDN/NFV environment and mathematical models for network resource optimization in NFV and construct a testbed for NFV
- ตุลาคม 2554 - ธันวาคม 2556 National Institute of Informatics, Tokyo, Japan
 - Project: Future Network Architecture
 - Responsible for developing a mathematical model for efficient caching and routing mechanism for improving content delivery in ICN

ความเชี่ยวชาญเฉพาะทาง

Network Security & Privacy, Deep Learning, Network Function Virtualization, Network Resource Management, ICN

ผลงานที่ผ่านมา

Book Chapter:

- K. Suksumboon, (2019) Performance Modeling of Virtual Packet Forwarding Functions for Wireless Network Virtualization. In: Shen X., Lin X., Zhang K. (eds) Encyclopedia of Wireless Networks. Springer, Cham.

Journal:

1. K. Suksomboon, et al., "Configuring a software router by the Erlang-k based packet latency prediction," IEEE Journal on Selected Areas in Communications, vol. 36, no.3, March 2018. (Q1, Impact factor: 8.085, Cited by: 2)
2. K. Suksomboon, et al., "Towards performance prediction of multicore software routers," IntJ Network Mgmt., vol. 27 no.2, Feb. 2017. (Q3, Impact factor: 1.32, Cited by: 4)
3. K. Suksomboon, P. Pongpalbool, Y. Ji and C. Aswakul, "PC-Nash: QoS Provisioning Framework with Path-Classification Scheme Under Nash Equilibrium" The Computer Journal, vol. 54 no.6, June 2011. (Q2, Impact factor: 1.10, Cited by: 4)

International Conferences (Selected papers):

1. K. Suksomboon, et al., "C2P2: Content-Centric Privacy Platform for Privacy-Preserving Monitoring Services," in Proc. IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC 2019), Milwaukee Wisconsin, USA, July 2019.
2. K. Suksomboon, et al., "Erlang-k-based packet latency prediction model for optimal configuration of software routers," in Proc. IEEE NetSoft, Bologna, Italy, June 2017. (h-index: 17, top score, Cited by: 2)
3. K. Suksomboon, et al., "A dilated-CPU-consumption-based performance prediction for multi-core software routers," in Proc. IEEE NetSoft, Seoul, Korea, June 2016. (h-index: 17, best paper candidates, Cited by: 8)
4. K. Suksomboon, M. Fukushima, M. Hayashi, R. Chawuthai, and H. Takeda, "LawNFO: A decision framework for optimal location-aware network function outsourcing," in Proc. IEEE NetSoft, London, UK, June 2015.
5. K. Suksomboon, M. Fukushima, and M. Hayashi, "Optimal virtualization of functionality for customer premise equipment," in Proc. IEEE ICC, London, UK, June 2015.
6. K. Suksomboon, M. Fukushima, M. Hayashi, and Y. Ji, "Pending-interest-driven cache orchestration through network function virtualization," in Proc. IEEE GLOBECOM, Austin TX, US, Dec. 2014.
7. K. Suksomboon et al., "Popcache: Cache more or less based on content popularity for information-centric networking," in Proc. IEEE LCN, Sydney, Australia, Oct. 2013. (h-index: 19, Best paper award, Cited by: 48)
8. S. Tarnoi, K. Suksomboon, W. Kumwilaisak, and Y. Ji, "Performance of probabilistic caching and cache replacement policies for content-centric networks," in Proc. IEEE LCN, Alberta, Canada, 2014.
9. K. Suksomboon, K. Ueda, A. Tagami, "Content centric privacy model for monitoring services in surveillance systems," in Proc. ACM ICN, Boston, US, 2018.
10. K. Suksomboon, A. Tagami, A. Basu, and J. Kurihara, "In-device proxy re-encryption service for information-centric networking access control," in Proc. IEEE LCN, Chicago, US, 2018.
11. K. Suksomboon, A. Tagami, A. Basu, and J. Kurihara, "IPRES: in-device proxy re-encryption service for secure ICN," in Proc. ACM ICN, Berlin, Germany, 2017.
12. S. Tarnoi, K. Suksomboon and Y. Ji, "Cooperative Routing Protocol for Content-centric Networking," in Proc. IEEE LCN, Sydney, Australia, 2013.
13. K. Suksomboon, et al., "On Incentive Based Inter-Domain Caching Mechanism for Optimal Future Network Design," in Proc. ACM AINTEC, Bangkok, Thailand, 2012.
14. K. Suksomboon, Y. Ji, P. Pongpalbool and C. Aswakul, "Load-Balanced Path Provisioning for Guaranteeing End-to-End QoS in Inter-Domain Networks," in Proc. ISCIT, Tokyo, Japan, 2010.

Japanese Patents

1. Method, device and program for predicting delay of packet in server (JP2018142799A)
2. Control device and program for packet transfer device having multi-core CPU, (JP2017117009A)
3. Device for controlling packet transfer device that has multi-core cpu, and computer-readable storage medium (WO2017110619A1) (Cited by: 1(Honda motor industry stock company))
4. Method and program for discriminating arrangement position of function of network (JP2017175464A)
5. Network design apparatus and program, (JP6475966B2, JP6301224B2)
6. Program and the network controller (JP6238233B2)
7. Client device for acquiring content (JP2019075619A)

Awards

1. Excellent Basic Research Award 2015 from KDDI Research Inc., Japan.
2. Best Paper Award from The 38th IEEE LCN conference, 2013.

ศาสตราจารย์ ดร. กุศล สุksomboon
ผอ.ศูนย์วิจัยและพัฒนา
เทคโนโลยีเครือข่าย

ประวัติบุคลากรในโครงการ



ชื่อ – นามสกุล : ดร.กุลชาติ มีทรัพย์หลาก

ตำแหน่งงาน : นักวิจัย

ตำแหน่งในโครงการ : หัวหน้าโครงการ

ที่อยู่ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ เลขที่ 112 อุทยานวิทยาศาสตร์ประเทศไทย ถ.พหลโยธิน ต. คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120

สถานที่ทำงาน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ประวัติการศึกษา

- ปริญญาเอก Information Science, University of Pittsburgh
- ปริญญาโท Electrical Engineering (Systems), University of Michigan
- ปริญญาตรี Electrical Engineering, The Johns Hopkins University

ประสบการณ์การทำงาน

- กันยายน 2549 – ปัจจุบัน เป็นนักวิจัยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
- ตุลาคม 2561 – ปัจจุบัน หัวหน้าทีมระบบไซเบอร์-กายภาพ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ใบอนุญาต / ทะเบียน :

นักวิจัย :

ความเชี่ยวชาญเฉพาะทาง

Industrial Internet of Things, Operations Research, Optimization

ผลงานที่ผ่านมา

- A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, “Authorization Mechanism for MQTT-based Internet of Things”, ICC, May 2016.

- Y.Khaokaew, T. Anusas-amornkul, K. Meesublak, "Intrusion Detection System based on Hybrid Feature Selection and Support Vector Machine (HFS-SVM)," In Applied and Mechanics and Materials, Vol. 781, pp.125-128, 2015.
- P. Pongpaibool, A. Niruntasukrat, C. Issariyapat, K. Meesublak, C. Mokarat, and P. Aiumsupucgul, "NethAM-nano": A robust and Scalable Service-Oriented Platform for Distributed Monitoring." Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference, p.51, ACM, 2014.
- K. Meesublak, "A Stochastic Assignment Model for Transportation Network," in proceeding of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WICOM 2011), Wuhan, china, Sept. 2011.
- P. Posawang, K. Meesublak, and W. Pattara-atikorn, "Cellular-based location tracking using decision tree and neural network techniques," 17th ITS World Congress, Busan, Korea, 2010.
- K. Meesublak, A two-layer network design problem under traffic uncertainty, ECTI-CON 2009, pp.914-917, May 2009.
- K. Meesublak, Network design under demand uncertainty, Proceedings of APAN Network Research Workshop 2008, pp. 19-23.
- K. Meesublak, K. Siri Wong N.A, Tele-referral system for remote hospitals in Thailand, The 8th ASEAN Scientific and Technical Conferences, Philippines, 2008.

รับรองสำเนาถูกต้อง

น.อ.อ.อ.

นาย กุลชาติ สิริพันธ์นวก

ประวัติบุคลากรในโครงการ



ชื่อ – นามสกุล : ดร.เอมอชญา นีรัตน์สุรัตน์

ตำแหน่งงาน : นักวิจัย

ตำแหน่งในโครงการ : นักวิจัย

ที่อยู่ : ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ เลขที่ 112 อุทยานวิทยาศาสตร์ประเทศไทย ถ.พหลโยธิน
ต.คลองหนึ่ง อ.คลองหลวง จ.ปทุมธานี 12120

สถานที่ทำงาน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

ประวัติการศึกษา

- ปริญญาเอก Electrical and Computer Engineering, University of Maryland, College Park
- ปริญญาโท วศ.ม. (ไฟฟ้าสื่อสาร) จุฬาลงกรณ์มหาวิทยาลัย
- ปริญญาตรี วศ.บ. (ไฟฟ้าสื่อสาร เกียรตินิยม) จุฬาลงกรณ์มหาวิทยาลัย

ประสบการณ์การทำงาน

- มีนาคม 2554 - ปัจจุบัน ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
 - หัวหน้าทีมโครงการพัฒนา Mobile Ad Hoc Network Protocol
 - หัวหน้าทีมโครงการวิจัย Internet of Things Service Platform Security
 - หัวหน้าทีมโครงการวิจัย Internet of Things Edge Computing Platform
- สิงหาคม 2555 – ธันวาคม 2561 สถาบันเทคโนโลยีแห่งเอเชีย
 - อาจารย์สอนคอร์ส Digital Signal Processing, Communication Engineering, Information Theory และ Coding Techniques
 - เป็นอาจารย์ที่ปรึกษาและประธานกรรมการโครงการ Undergraduate students' graduation (Capstone)
 - กรรมการวิทยานิพนธ์ระดับปริญญาเอก
- กันยายน 2554 – พฤษภาคม 2557 สถาบันเทคโนโลยีแห่งเอเชีย
 - ร่วมมือกับโครงการ DUMBO-FIBO ในการพัฒนาต้นแบบ robotic router สำหรับการสื่อสาร ad hoc ในพื้นที่เสี่ยงภัย
- สิงหาคม 2549 - พฤษภาคม 2550 Department of Electrical and Computer Engineering, University of Maryland (College Park)
 - Teaching Assistant and Research Assistant

ใบอนุญาต / ทะเบียน :

นักวิจัย :

รับรองสำเนาถูกต้อง

Prof. Dr. Niruntasukrat

เจมส์ จี. นันตสุภกิจ

ความเชี่ยวชาญเฉพาะทาง

Internet of Things, Network Security, Wireless Sensor Network, Delay Tolerant Network, Signal Processing

ผลงานที่ผ่านมา

- Niruntasukrat, A., Issariyapat, C., Pongpaibool, P., Meesublak, K., Aiumsupucgul, P., & Panya, A. (2016, May). Authorization mechanism for mqtt-based internet of things. In 2016 IEEE international Conference on Communications Workshops (ICC) (pp. 290-295). IEEE.
- Naruephiphat, W., Promya, R., & Niruntasukrat, A. (2015, November). Remote air conditioning control system based on ZigBee Wireless Sensor Network for building. In 2015 International Computer Science and Engineering Conference (ICSEC) (pp. 1-6). IEEE.
- Pongpaibool, P., Niruntasukrat, A., Issariyapat, C., Meesublak, K., Mokarat, C., & Aiumsupucgul, P. (2014, November). Netham nano: A robust and scalable service-oriented platform for distributed monitoring. In Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference (p. 51). ACM.
- Klaokliang, N., Teawtim, P., Airtongkham, P., So In, C., & Niruntasukrat, A. (2018, July). A novel IoT authorization architecture on hyperledger fabric with optimal consensus using genetic algorithm. In 2018 Seventh ICT International Student Project Conference (ICT-ISPC) (pp. 1-5). IEEE.
- Sandulescu, G., Niruntasukrat, A., & Charnsripinyo, C. (2015). Guaranteed capacity bounds in intermittently-connected networks: A resource-aware, holistic evaluation. Computer Communications, 59, 12-23.
- Sandulescu, G., Niruntasukrat, A., & Charnsripinyo, C. (2013, November). Resource-aware capacity evaluation for heterogeneous, disruption-tolerant networks. In Proceedings of the 9th Asian Internet Engineering Conference (pp. 57-64). ACM.
- Sandulescu, G., Niruntasukrat, A., & Charnsripinyo, C. (2013, November). Resource-aware capacity evaluation for heterogeneous, disruption-tolerant networks. In Proceedings of the 9th Asian Internet Engineering Conference (pp. 57-64). ACM.
- Benjapolakul, W., & Niruntasukrat, A. (2004). An Application of Neuro-Fuzzy Systems for Access Control in Asynchronous Transfer Mode Networks. In Neuro-Fuzzy and Fuzzy-Neural Applications in Telecommunications (pp. 235-281). Springer, Berlin, Heidelberg.
- Niruntasukrat, A., & Benjapolakul, W. (2001, May). A novel scheme for policing mechanism in ATM networks: feedback fuzzy leaky bucket. In ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196) (Vol. 3, pp. 161-164). IEEE.
- Benjapolakul, W., & Niruntasukrat, A. (2001, May). Backward predictive congestion control notification in ATM networks using neural network prediction. In ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196) (Vol. 3, pp. 165-168). IEEE.

ประวัติบุคลากรในโครงการ



ชื่อ - นามสกุล : ดร.โสภณ มงคลลักษณ์

ตำแหน่งงาน : อาจารย์ประจำ

ตำแหน่งในโครงการ : นักวิจัย

ที่อยู่ : ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร 114 ซอยสุขุมวิท 23 เขตวัฒนา กรุงเทพมหานคร 10110

สถานที่ทำงาน คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติการศึกษา

- ปริญญาเอก ปร.ด (วิทยาการคอมพิวเตอร์) มหาวิทยาลัย มหิดล
- ปริญญาโท วท.ม. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัย มหิดล
- ปริญญาตรี วศ.บ. (คอมพิวเตอร์) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์การทำงาน

- พฤษภาคม 2562 - ปัจจุบัน อาจารย์ประจำ, ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
 - สอนในระดับปริญญาโท สาขาวิทยาการข้อมูล, สอนในระดับปริญญาตรี สาขาวิทยาการคอมพิวเตอร์ และงานทำวิจัย
 - สอนรายวิชาการโปรแกรมเชิงวัตถุ, การโปรแกรมคอมพิวเตอร์เบื้องต้น, ระบบความปลอดภัยคอมพิวเตอร์, และ Internet of Things
- เมษายน 2560 - เมษายน 2562 อาจารย์ประจำ, สถาบันเทคโนโลยีไทย-ญี่ปุ่น
 - สอนในระดับปริญญาตรี สาขาเทคโนโลยีสารสนเทศ
 - สอนรายวิชา Data Structure and Algorithms และ วิชา Internet of Things
 - เป็นอาจารย์ที่ปรึกษาในระดับปริญญาโทและตรี ทำงานวิจัยด้าน Text mining และความปลอดภัยในระบบ LORAWAN
- เมษายน 2552 - สิงหาคม 2555 ผู้ช่วยนักวิจัย, ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
 - ร่วมพัฒนาด้านแบบซอฟต์แวร์บริหารจัดการเครือข่าย NethAM พร้อมทั้งเขียนบทความวิชาการ
- พฤศจิกายน 2553 - มีนาคม 2554 Research intern Fukuda laboratory, National Institute of Informatics(NII), Tokyo, Japan
 - Research topic: "Counting NATted Hosts by Observing TCP/IP Field Behaviors"

สำเนาถูกต้อง

ดร.โสภณ มงคลลักษณ์

- มกราคม 2551 – มีนาคม 2552 โปรแกรมเมอร์(outsource), ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
 - ร่วมพัฒนาด้านแบบซอฟต์แวร์บริหารจัดการเครือข่าย NtopViewer และร่วมเขียนบทความวิชาการ
- มิถุนายน 2546 – เมษายน 2550 วิศวกรระบบคอมพิวเตอร์, ธนาคารธนชาติ, ประเทศไทย
 - บริหารจัดการระบบคอมพิวเตอร์ต่างๆเช่น Mail Server, Web Server, Domain Server, DNS Server, และ Backup System ที่เป็นระบบปฏิบัติการ Windows

ใบอนุญาต / ทะเบียน :

นักวิจัย :

ความเชี่ยวชาญเฉพาะทาง

Internet Traffic measurement and analysis, Network management, Computer network security, Internet of Things (IoT), Blockchain technology, and Data science.

อบรมและเวิร์คช็อป

1. “Offensive & Detection Course” by SEC Playground Co., Ltd., February 2018, Thailand
2. “Car Hacking via the CAN BUS” workshop, JCSSE 2018, July 2018, Thailand.

ผลงานที่ผ่านมา

- Mongkolluksamee, S., Visoottiviseth, V., & Fukuda, K. (2018, Nov). Robust Peer to Peer Mobile Botnet Detection by Using Communication Patterns. In Proceedings of the AINTEC 2018 on Asian Internet Engineering Conference (p. 1-8). ACM. (The Best Paper Award)
- Mongkolluksamee, S., Visoottiviseth, V., & Fukuda, K. (2016, Mar). Combining Communication Patterns & Traffic Patterns to Enhance Mobile Traffic Identification Performance. In Journal of Information Processing (JIP), Vol. 24, No.2. (Specially Selected Paper Award).
- Thay, C., Visoottiviseth, V., & Mongkolluksamee, S. (2015, Nov). P2P Traffic Classification for Residential Network. In Proceedings of the 19th ICSEC, IEEE.
- Mongkolluksamee, S., Visoottiviseth, V., & Fukuda, K. (2015, Jul). Enhancing the Performance of Mobile Traffic Identification with Communication Patterns. In Proceedings of IEEE COMPSAC2015. IEEE.
- Issariyapat, C., Pongpaibool, P., Mongkolluksamee, S., Pukkawanna, S., & Meesublak, K. (2012, Jul). Using Nagios as a Groundwork for Developing a Better Network Monitoring System. In Proceeding of PICMET2012.
- Mongkolluksamee, S., Fukuda, K., & Pongpaibool, P. (2012, Jun). Counting NATed Hosts by Observing TCP/IP Field Behavior. In Proceedings of IEEE ICC2012. IEEE.
- Mongkolluksamee, S., Issariyapat, C., & Pongpaibool, P. (2010, May). Strengths and Limitations of Nagios as a Network Monitoring Solution. In Proceeding of JCSSE 2010.
- Mongkolluksamee, S., & Visoottiviseth, V. (2008, Feb)DiffservConditionalized Handover for HMIPv6. In Proceedings of IEEE ICACT2008, IEEE.

สำเนาถูกต้อง

วิวัฒน์ วงศ์ฉัตรพงษ์

- พัทธน์ โชคอาภา, โสภณ มงคลลักษณ์, พนิต พงษ์ไพบูลย์. (2011, May). การวัด End-to-End Available Bandwidth โดยการตรวจนับสถานะ TCP. In Proceedings of JCSSE 2011.
- โสภณ มงคลลักษณ์, ชาวีร์ อัสริยกัทร, พนิต พงษ์ไพบูลย์. (2010, May). การแจ้งเตือนปัญหาในระบบเครือข่ายแบบบูรณาการ. In Proceeding of ECTI-CARD 2010.

สำเนาถูกต้อง

โสภณ มงคลลักษณ์



กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)