การวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่

นายวัชร สายเชื้อ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

COST ANALYSIS OF AES IMPLEMENTATION ON MOBILE PROCESSOR

Mr. Vatchara Saicheur

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

หัวข้อวิทยานิพนธ์	การวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่
โดย	นายวัชร สายเชื้อ
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. วีระ เหมืองสิน)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร. พงศ์ธวัช ชีพพิมลชัย)

วัชร สายเชื้อ : การวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ (COST ANALYSIS OF AES IMPLEMENTATION ON MOBILE PROCESSOR) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. เกริก ภิรมย์โสภา, 56 หน้า.

การศึกษานี้ได้ทำการวิเคราะห์ค่าใช้จ่ายของการเข้ารหัสแบบ AES โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่แบบ 32 บิต และ 64 บิต โดยใช้ iPhone5C และ iPhone7 ผลการศึกษาพบว่าการขยายขนาดบล็อกจาก 128 บิต (AES-128) เป็น 512 บิต (AES-512) ช่วยเพิ่มประสิทธิภาพของอัลกอริทึม AES ได้ รวมทั้งการเพิ่มความยาวของกุญแจจากมาตรฐานเดิม ให้เป็น 512 และ 1024 บิต ช่วยให้มีความปลอดภัยของข้อมูลมากยิ่งขึ้น การวิเคราะห์เปรียบเทียบค่าใช้จ่ายของการเข้ารหัสโดยใช้อัลกอริทึมทั้ง 2 แบบ พบว่าการขยายขนาดบล็อกทำให้มีสปีดอัพเพิ่มขึ้น 1.21 – 1.64 เท่าบน iPhone5C และ 1.19 – 1.55 เท่าบน iPhone7 ขึ้นอยู่กับขนาดกุญแจที่ใช้ เมื่อพิจารณา CPU time ที่ใช้ในการเข้ารหัสพบว่า AES-512 ใช้เวลาน้อยกว่า AES-128 ในขณะที่การใช้หน่วยความจำของอัลกอริทึมทั้ง 2 แบบไม่แตกต่างกัน ซึ่ง iPhone7 จะมีการใช้หน่วยความจำมากกว่า iPhone5C จากการศึกษาสามารถสรุปได้ว่าการขยายขนาดบล็อกสามารถเพิ่มประสิทธิภาพในการเข้ารหัสบนอุปกรณ์แบบเคลื่อนที่ได้โดยมีค่าใช้จ่ายที่น้อยกว่า ผลการศึกษาครั้งนี้จะเป็นประโยชน์ในการพัฒนาการรักษาความปลอดภัยของข้อมูลโดยใช้อุปกรณ์แบบเคลื่อนที่ต่อไปได้

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2559

5670370621 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: AES / CRYPTOGRAPHY / COST ANALYSIS / MOBILE DEVICES

VATCHARA SAICHEUR: COST ANALYSIS OF AES IMPLEMENTATION ON MOBILE PROCESSOR. ADVISOR: ASST. PROF. KRERK PIROMSOPA, Ph.D., 56 pp.

This study is the cost analysis of two Advanced Encryption Standard (AES) algorithms on 32-bit and 64-bit Apple mobile processor by using iPhone5C and iPhone7. Our analysis shows increasing in performance when expanding the block size from 128 bits (AES-128) to 512 bits (AES-512). Similarly increasing the length of encryption key to 512 bits and 1024 bits yields stronger security. Our aim is to analyze the encryption cost different between the original AES-128 and the AES-512. The results showed that increasing block size will give 1.21 – 1.64 speed up on iPhone5C and 1.19 – 1.55 speed up on iPhone7 depending on the key length. Moreover, AES 512-bit block size shows faster CPU time in encryption than 128-bit block size. However, the memory usage for encryption on all key size are similar. iPhone7 used more memory than iPhone5C. In conclusion, expanding block size to 512 bits can increase performance while this is also lower the cost on mobile device. This result may have the benefit in improving the security of personal data by using mobile phone.

Department: Computer Engineering Student's Signature .....
Field of Study: Software Engineering Advisor's Signature .....
Academic Year: 2016

กิตติกรรมประกาศ

ขอขอบคุณ ผศ. ดร. เกริก ภิรมย์โสภาที่ช่วยให้คำแนะนำหลายๆ อย่างกับผม ในช่วงปีสองผมได้ลงเรียนวิชา Security กับอาจารย์ด้วย สนุกและได้ความรู้มาก ๆ เลยครับ การที่งานวิจัยชิ้นนี้ได้ขึ้นฐานข้อมูลถึงสองที่และมีรูปเล่มวิทยานิพนธ์ขึ้นมาได้เช่นนี้ ต้องขอบคุณอาจารย์มาก ๆ เลยครับ

ขอขอบคุณ รศ. ดร. วันชัย ริ้วไพบูลย์ ที่ช่วยเริ่มต้นให้ผมช่วงเด็กใหม่ปีหนึ่งก่อนที่ท่านจะเกษียณ ต้องขอขอบคุณมาก ๆ เลยครับผม

ขอขอบคุณผศ. ดร. ณัฐวุฒิ หนูไพโรจน์, ผศ. ดร. วีระ เหมืองสิน และอาจารย์ ดร. พงศ์ธวัช ชีพพิมลชัย ที่มาเป็นกรรมการสอบให้ผมตั้งแต่ครั้งสอบ Proposal เมื่อสองปีที่แล้ว และอีกสองปีถัดมาในการสอบวิทยานิพนธ์นี้ด้วยครับ การทำงานไปด้วยเรียนไปด้วยผมเลยอาจจะใช้เวลานานไปซักหน่อย แต่สุดท้ายก็มาถึงปลายทางการเขียนเล่มวิทยานิพนธ์นี้จนได้ ขอขอบคุณคำแนะนำจากอาจารย์มาก ๆ เลยครับ ผมมาอัพเดทงานวิจัยกับอาจารย์เกริก เป็นประจำทุกอาทิตย์เลยครับอาจารย์

ขอขอบคุณ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้ผมได้มีโอกาส มาศึกษาในระดับปริญญาโท ณ ที่แห่งนี้ ขอบคุณมาก ๆ ครับที่ให้โอกาสผม

สุดท้ายนี้ ขอขอบคุณ พ่อ, แม่, พี่ชาย และแฟนของผมที่เป็นกำลังใจ ช่วยประคับประคองผมเรื่อยมาจนถึงวันนี้

ด้วยความเคารพอย่างสูง

วัชร สายเชื้อ

สารบัญ

หน้า

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ผู้วิจัยต้องการนำเสนอการวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ เนื่องจากในปัจจุบันนี้อุปกรณ์แบบเคลื่อนที่ ไม่ว่าจะเป็น สมาร์ทโฟน แทปเล็ต สมาร์ทวอช มีเทคโนโลยีใหม่ๆ ออกมาอยู่เสมอทำให้การดำเนินชีวิตมีความสะดวกสบายมากยิ่งขึ้น ไม่ว่าจะช่วยในการทำงาน ช่วยในการสื่อสาร ช่วยในการเก็บบันทึกและส่งผ่านข้อมูลต่างๆ และเมื่อผู้คนต่างก็ใช้อุปกรณ์เคลื่อนที่เพื่อการสื่อสาร เพื่อการส่งข้อมูล หรือเพื่อการทำงานผ่านทางเครือข่ายอินเตอร์เน็ตมากขึ้น ก็ย่อมมีความเสี่ยงที่ข้อมูลอาจมีการรั่วไหลหรือถูกโจรกรรมโดยผู้ที่ไม่ประสงค์ดีได้ โดยเฉพาะข้อมูลที่มีความสำคัญซึ่งควรจะได้รับการปกป้อง ดังนั้นวิธีการหนึ่งที่จะช่วยปกป้องข้อมูลของเราให้ปลอดภัยขึ้นได้คือการเข้ารหัสข้อมูลนั้น วิธีที่ใช้ในการเข้ารหัสข้อมูลนั้นก็มีหลายวิธี แต่วิธีที่ได้รับความนิยมคือการเข้ารหัสโดยใช้อักกอริทึม AES (Advanced Encryption Standard) ซึ่งเป็นอัลกอริทึมการเข้ารหัสมาตรฐานที่ได้รับการรับรองจากรัฐบาลสหรัฐอเมริกาสำหรับเข้ารหัสและจัดเก็บข้อมูลส่วนบุคคลโดยมีชื่อเดิมว่า Rijndael [1] ต่อมาสถาบันมาตรฐานและเทคโนโลยีแห่งชาติสหรัฐอเมริกา (National Institute of Standards and Technology) ได้ดัดแปลงแก้ไขใหม่และเปลี่ยนชื่อเป็น Advanced Encryption Standard หรือชื่อย่อ AES โดยเริ่มใช้ครั้งแรกตั้งแต่ปี ค.ศ.2001 จนถึงปัจจุบัน

การเข้ารหัสแบบ AES นั้นเป็นการเข้ารหัสเพื่อปกปิดข้อมูลด้วยกุญแจเดี่ยวแบบสมมาตร (Symmetric-key) โดยใช้บล๊อกขนาด 128 บิตซึ่งมีลักษณะเป็นอาร์เรย์ 2 มิติ ขนาด 4x4 ร่วมกับการใช้กุญแจที่มีขนาด 128, 192, 256 บิต ทำการเข้ารหัสเป็นจำนวน 10, 12, 14 รอบ ตามขนาดของกุญแจที่ใช้ [2] ซึ่งในปัจจุบันนั้นผ่านมามากกว่า 10 ปีแล้วเมื่อเปรียบเทียบกับความก้าวหน้าของเทคโนโลยีในปัจจุบัน อาจจะทำให้ AES แบบเดิมนั้นมีข้อจำกัดที่มากขึ้น และมีโอกาสที่จะสามารถถอดรหัสได้เร็วและง่ายขึ้นกว่าเดิม ประกอบกับในปัจจุบันสมาร์ทโฟน iPhone และ Android เป็นที่นิยมเป็นอย่างมาก เหตุด้วยมีความสามารถพื้นฐานใกล้เคียงกับคอมพิวเตอร์ แต่มีขนาดเพียงโทรศัพท์มือถือที่พกพาสะดวก โดยที่ผ่านมายังไม่มีงานวิจัยใดที่ทำการเพิ่มขนาดบล็อก AES และขนาดกุญแจ 512, 1024 บิต บนอุปกรณ์ iPhone และ นำเสนอการวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสมาก่อน ดังนั้นจากที่กล่าวมาข้างต้น ผู้วิจัยจึงต้องการสร้างและวิเคราะห์ (Implement & Analysis) ในงานวิจัยชิ้นนี้ ควบคู่กับการนำมาใช้บนอุปกรณ์สมาร์ทโฟน iPhone

1.2 วัตถุประสงค์ของการวิจัย

นำเสนอการวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต

1.3 ขอบเขตของการวิจัย

1) ต้องการนำเสนอการวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต
2) ปรับปรุงประสิทธิภาพอัลกอริทึม AES โดยใช้บล็อกขนาด 128 บิต กับ 512 บิต
3) เพิ่มขนาดกุญแจ 512 บิต และ 1024 บิต สำหรับงานความมั่นคงระดับสูง (TOP SECRET)
4) อัลกอริทึม AES ที่ปรับปรุงประสิทธิภาพนี้ สามารถเพิ่มปริมาณงานที่ได้ต่อหนึ่งหน่วยเวลา (throughput) โดยใช้ Block Size ขนาด 512 บิต
5) อัลกอริทึมสามารถเพิ่ม-ลดชนิดรูปแบบกุญแจได้ตามความเหมาะสมของงานตามที่ผู้ใช้ต้องการ อันประกอบไปด้วย
   5.1 กุญแจที่มีลักษณะเกิดจากการรู้จำของผู้ตั้งรหัสผ่านเอง
   5.2 กุญแจที่มีลักษณะเป็นอุปกรณ์ Token Hardware หรือลักษณะเฉพาะเอกลักษณ์บุคคล อาทิเช่น NFC, Bluetooth, Fingerprint เป็นต้น
6) ผลสำเร็จของงานวิทยานิพนธ์นี้อยู่ในรูปแบบ อัลกอริทึม AES ที่มีบล็อกขนาด 128 บิต กับ 512 บิตให้เลือกใช้ ที่ถูกใช้งานภายใต้เครื่องมือยูทิลิตี้ต้นแบบ (Tool Utility prototype)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้ผลวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต
2) ได้อัลกอริทึม AES ที่มีบล็อกขนาด 512 บิตที่ให้ปริมาณงานที่ได้ต่อหนึ่งหน่วยเวลามากกว่าบล็อกขนาด 128 บิต
3) ได้อัลกอริทึม AES ที่มีบล็อกขนาด 512 บิตที่ให้ความซับซ้อนมากกว่าบล็อกขนาด 128 บิต
4) ได้อัลกอริทึมเสริมประสิทธิภาพ AES ที่มีกุญแจชนิด128, 192, 256, 512, 1024 บิต
5) ได้เครื่องมือยูทิลิตี้ต้นแบบสาธิตการทำงาน (Tool Prototype utility) บนอุปกรณ์ iPhone
6) แก้ปัญหาถูกจำกัดเทคโนโลยีเข้ารหัสเนื่องด้วยข้อจำกัดของกฎหมายโดย NIST สหรัฐอเมริกา

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

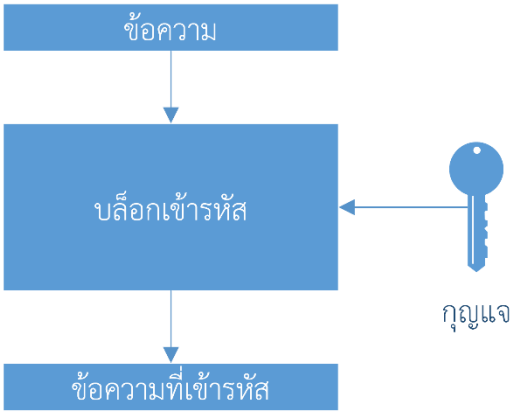
ในบทนี้จะอธิบายถึงทฤษฎีที่เป็นส่วนสำคัญสำหรับนำไปประยุกต์ใช้ในงานวิทยานิพนธ์ โดยแสดงองค์รวมความรู้ดังต่อไปนี้

- ศาสตร์แห่งการเขียนความลับ ด้วยกุญแจเดี่ยว (Symmetric Key Cryptography)
- มาตรฐานและการรับรอง (Standards & Certification)
- ศาสตร์การถอดรหัส และ การโจมตีความมั่นคงของข้อมูล (Cryptanalysis Security attacks)
- ความเป็นส่วนตัว (Privacy)

2.1.1 ศาสตร์แห่งการเขียนความลับ ด้วยกุญแจเดี่ยว (Symmetric Key Cryptography)

หลักการของการเข้ารหัสเพื่อปกปิดข้อมูลด้วยกุญแจเดี่ยว (Symmetric Key) คือจะทำการเข้ารหัสทีละบล็อก (Block) โดยหนึ่งบล็อกประกอบด้วยหลายไบค์ เช่น 32-64 ไบต์ เป็นต้น โดยการใช้คีย์ลับ (Secret-Key) เดียวกันในการเข้ารหัสข้อมูลและถอดรหัสข้อมูล (Encryption/Decryption) [3]

ตัวอย่างเช่น เรามีบล็อกเข้ารหัส ขนาด 128 บิต และมีข้อความขนาด 64 ไบต์ (โดย 1 ไบต์ มี 8 บิต) ดังนั้นข้อความนี้จึงมีขนาด 512 บิต ซึ่ง 512 บิต นี้จะถูกนำไปแบ่งลงตาม บล็อกที่มีความยาว 128 บิต ได้ 4 บล็อก หลังจากนั้นจึงทำการเข้ารหัสกับคีย์ ดังภาพที่ 1



ภาพที่ 1 แสดงบล็อกการเข้ารหัสข้อมูล

ซึ่งในกระบวนการถอดรหัสข้อมูลก็กระทำในลักษณะเดียวกันแต่เป็นการทำย้อนกลับกระบวนการซึ่งเขียนให้อยู่ในรูปสมการได้ดังนี้

X : Plaintext (ข้อความดังเดิม) = $[X_1,X_2,X_3,X_4.....X_n]$;

Y : Ciphertext (ข้อความปกปกหรือเข้ารหัสไว้) = $[Y_1,Y_2,Y_3,Y_4.....Y_n]$;

K : Key (กุญแจลับ) = $[K_1,K_2,K_3,K_4.....K_n]$;

สมการเริ่มต้นของการเข้ารหัสปกปิดข้อมูล $Y = E_k(X)$; | E : Encryption Function

สมการเริ่มต้นของการถอดรหัสข้อมูลปกปิด $X = D_k(X)$; | D : Decryption Function

หรือสามารถเขียนได้ในเทอมของ $X = D_k(E_k(X))$;

โดยปกติแล้วคีย์ที่เข้ารหัสก็มีขนาดแตกต่างกันไปตามอัลกอริทึมที่นำการเข้ารหัสแบบบล็อกไปใช้ ซึ่งอัลกอริทึมที่ใช้หลักการเข้ารหัสแบบบล็อกที่มีชื่อเสียง อาทิเช่น Data Encryption Standard (DES) [4] และ Rijndael : Advanced Encryption Standard (AES) [1] ทั้งสองอัลกอริทึมได้รับเลือกตามมาตรฐาน FIPS ของสหรัฐอเมริกาและใช้กันแพร่หลายทั่วโลก ซึ่งปัญหาความปลอดภัยของข้อมูลของผู้ใช้นั้นจะมากน้อยเท่าไรก็จะขึ้นอยู่กับความหลากหลายของรูปแบบการเข้ารหัสที่ใช้

จากหลักการข้างต้น ผู้วิจัยนำสมการและความรู้การเข้ารหัสพื้นฐาน มาใช้ในงานวิทยานิพนธ์

2.1.2 มาตรฐานและการรับรอง (Standards & Certification)

1) National Institute of Standards and Technology (NIST) เป็นสถาบันที่กำหนดมาตรฐานของเทคโนโลยี การนำไปใช้ หรือกระบวนการทำงานต่าง ๆ โดย NIST เป็นหน่วยงานราชการสังกัดกระทรวงพาณิชย์ของสหรัฐอเมริกา มีหน้าที่ในการรักษาความปลอดภัยให้รัฐบาลสหรัฐอเมริกา ซึ่งมาตรฐานรักษาความปลอดภัยที่หน่วยงานของรัฐบาลกลางจะใช้งานได้ต้องได้รับการรับรองจาก NIST ก่อนเสมอ และอันเนื่องมาจากธุรกิจทางเทคโนโลยีในสหรัฐอเมริกามีอิทธิพลสูง มาตรฐานที่ NIST ยอมรับก็มักจะกลายเป็นมาตรฐานกลางของทั่วโลก เช่น อัลกอริทึมย่อยสลายข้อมูลแบบ SHA [5] และการเข้ารหัสปกปิดแบบ AES

2) Federal Information Processing Standards (FIPS) คือมาตรฐานของรัฐบาลสหรัฐอเมริกาด้านเทคโนโลยีสารสนเทศและระบบความปลอดภัยคอมพิวเตอร์ โครงการ FIPS กำกับดูแลโดย NIST ซึ่งจะเป็นผู้ออกใบรับรอง FIPS 140 เพื่อกำหนดมาตรฐานการเข้ารหัสข้อมูลภายใต้ข้อกำหนดของรัฐบาลกลางสหรัฐอเมริกาเพื่อปกป้องข้อมูลที่มีความสำคัญ ผลิตภัณฑ์ทั้งหมดของหน่วยงานพลเรือนและหน่วยงานความมั่นคงของรัฐบาลกลางสหรัฐอเมริกาที่ใช้เทคโนโลยีการเข้ารหัสข้อมูล จะต้องได้รับใบรับรองมาตรฐาน FIPS 140 ทั้งนี้การขอการรับรองมาตรฐาน FIPS 140 กำหนดให้ต้องผ่านกระบวนการทดสอบที่เข้มงวดโดยห้องปฏิบัติการทดสอบที่ผ่านการรับรอง โดยในปัจจุบันใช้การรับรองมาตรฐาน FIPS 140-2 [6] โดยแบ่งได้เป็น 4 ระดับ คือ

ระดับ 1 : เป็นระดับต่ำสุดของการรักษาความปลอดภัย ใช้โมดูลการเข้ารหัสแบบพื้นฐาน โดยไม่มีกลไกรักษาความปลอดภัยทางกายภาพที่จำเพาะ ยกตัวอย่างเช่น การเข้ารหัสข้อมูลส่วนบุคคลบนอุปกรณ์คอมพิวเตอร์

ระดับ 2 : ปรับปรุงตามกลไกการรักษาความปลอดภัยทางกายภาพของโมดูลการเข้ารหัส รักษาความปลอดภัยระดับ 1 โดยการกำหนดคุณสมบัติด้านกายภาพที่เฉพาะเจาะจงเพิ่มขึ้นมาในการจัดเก็บหรือปกปิดข้อมูล

ระดับ 3 : เพิ่มเติมจากระดับ 1 และ 2 ในประเด็นป้องกันผู้บุกรุกจากการเข้าถึงข้อมูลที่ถูกปกปิด มีการตรวจจับและตอบสนองต่อความพยายามที่จะเข้าถึงข้อมูล หรือปรับเปลี่ยนโมดูลของการเข้ารหัสปกปิดต่อการพยายามเข้าถึงทางกายภาพ

ระดับ 4 : ระดับสูงสุดของการรักษาความปลอดภัย ให้ความสำคัญต่อโมดูลการเข้ารหัสและคุณสมบัติด้านกายภาพของการป้องกัน และตรวจจับหรือตอบสนองต่อความพยายามที่ไม่ได้รับอนุญาตในการเข้าถึงทางข้อมูล โดยออกแบบมาเพื่อทนทานต่อความผันผวนที่เกิดขึ้น และผ่านการทดสอบความล้มเหลวด้านสิ่งแวดล้อมอย่างเข้มงวดเพื่อให้ความเชื่อมั่นว่าโมดูลจะไม่ได้รับผลกระทบจากปัจจัยอันเป็นความเสี่ยงต่อข้อมูลซึ่งถูกปกปิด

จากมาตรฐานที่กล่าวมาข้างต้น รูปแบบของการศึกษาในวิทยานิพนธ์นี้มีการเข้ารหัสข้อมูลโดยใช้กุญแจทางกายภาพประกอบกับอัลกอริทึม จึงเทียบได้กับมาตรฐาน FIPS 140-2 ที่ระดับ 2

2.1.3 ศาสตร์การถอดรหัส และ การโจมตีความมั่นคงของข้อมูล (Cryptanalysis Security attacks)

กระบวนการรักษาความปลอดภัยควรต้องครอบคลุม 5 ปัจจัยหลักด้วยกัน [3] คือ

1) การรักษาความลับ (Confidence/Privacy) คือ การป้องกันไม่ให้ผู้ที่ไม่เกี่ยวข้อง หรือไม่มีสิทธิ์ทราบถึงข้อมูลข่าวสารอันเป็นความลับ

2) ความสามารถพร้อมใช้งาน (Availability) คือ การรักษาทรัพยากรข้อมูลให้ผู้มีสิทธิ์ใช้งาน สามารถเข้าถึงข้อมูลและข่าวสารได้ตลอดตามที่ตกลง

3) ความเชื่อถือได้หรือการรักษาความสมบูรณ์ (Integrity) คือ การปกป้องข้อมูลจากการเข้าถึง หรือดัดแปลงแก้ไขจากผู้ที่ไม่มีสิทธิ์ในการเข้าถึงข้อมูล

4) การห้ามปฏิเสธความรับผิดชอบ (Non-repudiation) คือ วิธีการสื่อสารซึ่งผู้ส่งข้อมูลได้รับหลักฐานว่าได้มีการส่งข้อมูลแล้วและผู้รับก็ได้รับการยืนยันว่าผู้ส่งเป็นใคร ดังนั้นทั้งผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่มีความเกี่ยวข้องกับข้อมูลดังกล่าวในภายหลัง

5) การพิสูจน์ตัวตน (Authentication) คือ การตรวจสอบว่าเป็นผู้มีสิทธิ์ในการเข้าถึงข้อมูลข่าวสารที่จัดเก็บไว้

และหากกล่าวถึงศาสตร์การถอดรหัสข้อมูล โดยผู้ชำนาญด้านวิเคราะห์รหัสปกปิด (Cryptanalysis) จะสามารถจำแนกชนิดการโจมตีออกได้ 4 ระดับ [3] ดังนี้

1) ระดับข้อความปกปิดเท่านั้น (Ciphertext only)
   - ฝ่ายตรงข้ามมีเพียงข้อความเข้ารหัสปกปิดเท่านั้น
   - ฝ่ายตรงข้ามมีข้อมูลในการวิเคราะห์น้อย
   - ความแข็งแกร่งขึ้นอยู่กับความทนทานของอัลกอริทึมที่เข้ารหัสปกปิด
2) ระดับมีข้อมูลจำนวนหนึ่ง (Know Plaintext)
   - ฝ่ายตรงข้ามมีข้อความดังเดิมและข้อความปกปิดที่เข้าคู่กันอยู่จำนวนหนึ่ง
   - ฝ่ายตรงข้ามทราบรูปแบบ (Pattern) ในการเข้ารหัสและถอดรหัส
3) ระดับเข้าถึงอัลกอริทึมปกปิด (Chosen Plaintext)
   - ฝ่ายตรงข้ามทราบอัลกอริทึมเข้ารหัสปกปิด (Encryption Algorithm)
   - ผ่ายตรงข้ามเลือกข้อความดังเดิม X และ สร้างข้อความปกปิด Y ได้
4) ระดับเข้าถึงอัลกอริทึมถอดรหัสปกปิด (Chosen Ciphertext)
   - ฝ่ายตรงข้ามทราบอัลกอริทึมถอดรหัสปกปิด (Decryption algorithm)
   - ผ่ายตรงข้าม ถอดรหัสได้

2.1.4 ความเป็นส่วนตัว (Privacy)

ความเป็นส่วนตัว เป็นแนวคิดที่มุ่งเน้น ความมั่นคงของข้อมูลหรือความลับใด ๆ ของเจ้าของ โดยมีประเด็นเกี่ยวกับความเป็นส่วนตัวหรือการไม่เปิดเผยตัวตนผู้ใช้งานบนเครือข่ายอินเทอร์เน็ต โดยตั้งอยู่บนพื้นฐานแนวความคิดเสรีนิยม (Libertarianism Democracy) ที่ให้ความสำคัญกับสิทธิ์และเสรีภาพของปัจเจกบุคคล (Individuals) เป็นหลัก [7]

ในระดับนานาชาติประเด็นสิทธิความเป็นส่วนตัวถูกกล่าวถึงในการประชุม Internet Governance Forum ในเดือนกันยายน 2557 โดยหน่วยงาน Association for Progressive Communication (APC) และหน่วยงาน Humanist Institute Cooperation with Developing Countries (Hivos) ได้นำเสนอรายงานที่มีชื่อว่า Global Information Society Watch 2014 [8] กล่าวถึง การสอดแนมข้อมูลสื่อสารในยุคดิจิตอล โดยรายงานฉบับดังกล่าวนำเสนอสถานการณ์การ

ติดตาม การเฝ้าระวัง และการสอดแนมจากรัฐใน 53 ประเทศทั่วโลกรวมถึงประเทศไทย และมีสรุปสถานการณ์ภาพรวมในประเด็นที่เกี่ยวข้อง อาทิเช่น

- การเฝ้าระวังบนโลกออนไลน์กับสิทธิมนุษยชน
- การเฝ้าสังเกตและตรวจสอบประชาชนบนโลกออนไลน์
- ภัยคุกคามจากการสอดแนมความเป็นส่วนตัวของประชาชน
- การรวมกลุ่มกัน แสดงออกทางความคิดหรือการกระทำโดยสิทธิ์และเสรีภาพของปัจเจกบุคคล

ยกตัวอย่างการนำประเด็นการคุ้มครองความเป็นส่วนตัวและสิทธิส่วนบุคคลไปใช้ที่เห็นได้ชัดเจน เช่นกรณี มาตรฐานนโยบายคุ้มครองสิทธิส่วนบุคคล (Platform for Privacy Preferences : P3P) ซึ่งกำกับดูแลโดย เวิลด์ไวด์เว็บคอนซอร์เทียม (World Wide Web Consortium : W3C) ได้เสนอคำแนะนำในทางปฏิบัติที่จะให้ผู้ใช้กำหนดและแบ่งปันข้อมูลส่วนบุคคลที่ยอมรับได้กับเว็บไซต์หรือผู้ให้บริการซึ่งมีชื่อเรียกกว่า นโยบายความเป็นส่วนตัว (Privacy Policy Statement : PPS) โดยต้องมีความเข้าใจที่ตรงกันระหว่างผู้ใช้บริการและผู้ให้บริการ ซึ่งรายละเอียดโดยร่วมกล่าวถึง การยอมรับการเข้าถึงข้อมูลส่วนตัวหรือแบ่งบันข้อมูลส่วนบุคคลของเรากับผู้ให้บริการ [9] โดยในปัจจุบันเครือข่ายสังคมออนไลน์และเทคโนโลยีมีการเปลี่ยนแปลงที่รวดเร็ว ประกอบกับรูปแบบการเก็บข้อมูลที่หลากหลาย จึงทำให้ผู้บริโภคให้ความสนใจกับประเด็นความเป็นส่วนตัวมากยิ่งขึ้น

จากทฤษฎีนี้ ผู้วิจัยจึงได้ใช้หลักการของความเป็นส่วนตัวเป็นพื้นฐาน และเชื่อมโยงกับเทคโนโลยี เพื่อสร้างรูปแบบของการรักษาความปลอดภัยของข้อมูลในระดับสูงในวิทยานิพนธ์นี้

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะอธิบายถึงงานวิจัยที่เกี่ยวข้องและศึกษาต่อยอด ที่เป็นส่วนสำคัญสำหรับนำไปประยุกต์ใช้ในงานวิทยานิพนธ์ โดยแสดงองค์รวมความรู้ดังต่อไปนี้

- งานวิจัย The Rijndael algorithm
- งานวิจัย The Twofish Encryption Algorithm: A 128-Bit Block Cipher
- งานวิจัย การทำรหัสลับแบบ AES บนหน่วยประมวลผลหลายแกนเพื่อเพิ่มประสิทธิภาพ
- งานวิจัย AES-512 : 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation
- งานวิจัย Engineer Privacy

2.2.1 งานวิจัย The Rijndael algorithm [2]

ในปี ค.ศ. 1998 Joan Daemen และ Vincent Rijimen นักวิจัยชาวเบลเยียม ได้เสนออัลกอริทึมเข้ารหัสแบบชนิดกุญแจสมมาตรที่มีชื่อว่า Rijndael ซึ่งต่อมาในปี ค.ศ .2001 อัลกอริทึมนี้ได้รับคัดเลือกจาก สถาบันกำหนดมาตรฐานเทคโนโลยีสหรัฐอเมริกา (NIST) ให้เป็นมาตรฐานการเข้ารหัสแบบชนิดกุญแจสมมาตรแทนที่อัลกอริทึม 3DES เดิม ที่เริ่มไม่เพียงพอต่อการโจมตี (Break) และเปลี่ยนชื่อจาก Rijndael เป็น Advanced Encryption Standard หรือที่รู้จักกันในชื่อย่อ AES ถึงแม้ Rijndael ไม่ใช้อัลกอริทึมที่แข็งแกร่งที่สุด แต่มีข้อดีเรื่องความเร็วและใช้งานง่ายในทางปฏิบัติ ดังนั้น Rijndael จึงถูกนำมาใช้กันอย่างแพร่หลายในปัจจุบัน

ตารางที่ 1 คุณลักษณะของอัลกอริทึม AES

ขนาดบล็อก	128 บิต
ขนาดกุญแจ	128, 192, 256 บิต
จำนวนรอบเข้ารหัส	10, 12, 14 รอบ ตามขนาดกุญแจที่ใช้
โครงสร้าง	Substitution-Permutation Network

โดยหลักการทำงานของ Rijndael จะแบ่งออกเป็น 4 ขั้นตอนหลักๆ คือ

1. SubBytes เป็นกระบวนการแทนที่ไบต์โดยใช้ตารางการแทนที่
2. ShiftRows การเลื่อนไบต์ในแนวแถวของอาร์เรย์ State ด้วยออฟเซตที่ต่างกันไปในแต่ละแถว
3. MixColumns ผสมผสานข้อมูลภายในคอลัมน์แต่ละคอลัมน์ของอาร์เรย์ State
4. AddRound key บวกค่ากุญแจในแต่ละรอบกับ อาร์เรย์ State โดยในรอบสุดท้ายจะใช้ AddRoundKey แทน MixColums

แต่ว่า AES นั้น ก็ยังมีประเด็นน่าสนใจที่ถูกค้นพบมาเรื่อย ๆ ตั้งแต่ปี 2001 ที่เริ่มมีเทคโนโลยีการประมวลผลแบบ 64 บิต จนถึงปัจจุบัน อาทิเช่น

- ตัวอัลกอริทึมที่ออกแบบถูกสร้างและรับรอบมาตรฐาน โดยกระบวนการทั้งหมดเกิดขึ้นในสมัยที่ใช้เทคโนโลยีการประมวลผลแบบ 32 บิต ดังนั้นในยุคปัจจุบัน อัลกอริทึมการประมวลผลเข้ารหัสและถอดรหัส ไม่สามารถดึงประสิทธิภาพ เทคโนโลยีการประมวลผลแบบ 64 บิต ได้อย่างเต็มที่
- ด้วยตัวบทกฎหมายของสหรัฐอเมริกา ที่ควบคุมหรือห้ามส่งออกเทคโนโลยีคุณภาพสูงออกนอกประเทศ ทำให้อัลกอริทึม AES ด้วยประสิทธิภาพด้านกุญแจกว่าที่ควรจะเป็น

- กุญแจความยาว 256 ที่รับรองโดย NISTและ NSA อาจไม่เพียงพอต่อ วิธีการโจมตีแบบคาดเดารหัสผ่าน (Brute force attack) มีประสิทธิภาพเพิ่มมากขึ้นตามกฎของมัวร์ (Moore's law) [4] อาทิเช่น การเช่าAmazon Web Services (AWS) มาทำฟาร์มเซิร์ฟเวอร์ (Cloud Server Farm) ช่วยประมวลผล หรือเทคโนโลยีการแยกส่วนช่วยกันคำนวณปัญหาหนึ่งๆ แบบ P2P ที่ใช้บน Bitcoin
- การโจมตีแบบ Cache Timing Attack ที่ใช้ปริมาณความถี่ของข้อมูลเข้าไปขัดจังหวะ ซึ่งต้องใช้ Chosen Plaintexts จำนวนมากกว่าหลักร้อยล้านขึ้นไป ซึ่งตรงประเด็นนี่เป็นเรื่องยากที่จะเกิดได้จริงในทางปฏิบัติ

จากงานวิจัยนี้ ผู้จัดทำวิทยานิพนธ์ต้องการออกแบบอัลกอริทึมที่เสริมประสิทธิภาพ ที่มั่งคงมากกว่ามาตรฐาน AES เดิมที่ผ่านการรับรองตั้งแต่ปี 2001 ในประเด็น ปริมาณงานที่ได้ต่อหนึ่งหน่วยเวลา (throughput) และประเด็นความทนทานยากต่อการโจมตีด้วยวิธีคาดเดารหัส (Brute force attack) เพื่อให้สอดคล้องกับการใช้งานบนเทคโนโลยีการประมวลผลในยุคปัจจุบัน โดยที่ไม่ติดปัญหาตัวบทกฎหมายของสหรัฐอเมริกา

2.2.2 งานวิจัย The Twofish Encryption Algorithm: A 128-Bit Block Cipher [10]

ผู้เชี่ยวชาญด้านการเข้ารหัส Bruce Schneier และคณะ ได้พัฒนาอัลกอริทึม TwoFish ต่อยอดจากอัลกอริมเดิม Blowfish ที่สร้างขึ้นมาทดแทน 3DES ซึ่งอัลกอริทึม TwoFish ที่ได้รับการเพิ่มความสามารถนี้มีประสิทธิในแง่ความเร็วและความทนทานที่เพิ่มขึ้น ซึ่งต่อมา TwoFish เป็นหนึ่งในอัลกอริทึมที่เข้ารอบในงานแข่งขัน AES ซึ่งถูกจัดโดย NIST โดยมีลำดับเป็นที่ 2 รองจาก Rijndael

ตารางที่ 2 เปรียบเทียบคุณลักษณะของ Blowfish และ Twofish Algorithm

คุณลักษณะ	Blowfish Algorithm	Twofish Algorithm
ขนาดบล็อก (Block size)	64 bit	128 bit
ขนาดคีย์ (Key size)	32-448 bit เพิ่มทีละ 8 bit ค่าปกติ 128 bit	128, 192, 256 bit
จำนวนรอบเข้ารหัส	16 รอบ	16 รอบ
โครงสร้าง	Feistel Network	Feistel Network

แต่เดิม Blowfish ใช้วิธีผสมผสานเทคนิค Feistel Network, Key-Dependent S-Box และ Non-Invertible F Function เข้าด้วยกัน ต่อมาอัลกอริทึม TwoFish ได้เพิ่มขั้นตอน เหล่านี้เข้าไปจากเดิมที่มีอยู่

- Pre-Computed Key-Dependent S-Boxes
- Relatively Complex Key Schedule
- Pseudo-Hadamard Transform (PHT) ซึ่งเป็นเทคนิคจากอัลกอริทึมตระกูล SAFER ซึ่งเป็นหนึ่งในผู้เข้าแข่งขันAESเช่นกัน

ผู้จัดทำวิทยานิพนธ์ศึกษางานของ Mr. Bruce Schneier ซึ่งมีอิทธิพลต่อวงการเข้ารหัสอย่างมาก ทุกการกระทำจึงเป็นเรื่องที่น่าสนใจ เช่นเดียวกับประเด็นจำนวนรอบของการเข้ารหัสมีการถกเถียงกันอย่างมาก ตัวอย่างคือ Blowfish และ TwoFish ที่ยึดถือจำนวนรอบไว้ที่ 16 รอบ ก็เพียงพอแล้วด้วยเหตุผลการคำนวณทางคณิตศาสตร์สมัยใหม่ แต่ก็ยังมีอีกแนวความคิดแบบอนุรักษ์นิยม ยกตัวอย่างอัลกอริทึม Serpent ซึ่งเป็นอันดับ 3 ในการแข่งขัน AES ถึงแม้โดยรวมจะแพ้อัลกอริทึม Rijndael แต่ประเด็นความทนทานมีมากกว่า ด้วยการเข้ารหัสถึง 2 เท่าของ 16 ซึ่งก็คือ 32 รอบ

2.2.3 งานวิจัย การทำรหัสลับแบบ AES บนหน่วยประมวลผลหลายแกนเพื่อเพิ่มประสิทธิภาพ [11]

ในปีการศึกษา 2556นายศุภชัย ทองสุข ได้นำเสนอวิธีการเข้ารหัส AES ด้วยกุญแจขนาด 128 บิต บนหน่วยประมวลผลหลายแกน ที่ชื่อว่า S2 โดยวิเคราะห์ประสิทธิภาพหน่วยประมวลผลแบบหลายแกนเทียบกับแบบแกนเดียว ซึ่งผลที่ได้ สามารถเพิ่ม ปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลาได้ดี อีกทั้งงานวิจัยนี้ได้กล่าวถึงงานวิจัยของ Mr. Angelo Barnes [12] ที่นำเสนอการเพิ่มปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลา (Throughput) ของการเข้ารหัส AES บนหน่วยประมวลแบบหลายแกน ด้วยวิธีแบ่งข้อมูลตามจำนวนของหน่วยประมวลผล โดยผลที่ได้สามารถเพิ่มงานที่ทำได้ต่อหนึ่งหน่วยเวลาได้อย่างมีประสิทธิภาพเช่นเดียวกัน ซึ่งทั้งสองงานวิจัย ให้ผลลัพท์ที่ดีไปในทางเดียวกันเกี่ยวกับประเด็นการประมวลผลชนิดแบบหลายแกน

จากงานวิจัยข้างต้น ทำให้ผู้จัดทำวิทยานิพนธ์เล็งเห็นว่า การเพิ่มปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลา (Throughput) บนหน่วยประมวลผลแบบหลายแกนทำได้จริงและเป็นเรื่องที่ดี แต่ยังไม่สามารถดึงประสิทธิภาพเทคโนโลยีการประมวลผลแบบ 64 บิต มาใช้งานได้ ด้วยข้อจำกัดของการนำอัลกอริทึม AES มาใช้ด้วยขนาด Block Size ที่ 128 บิต เท่านั้น ดังนั้นวิทยานิพนธ์นี้จึงต้องการต่อยอดงานวิจัยบนพื้นฐานเทคโนโลยีการประมวลผลชนิด 64 บิต บนหน่วยประมวลผลหลายแกน ด้วยอัลกอริทึม AES ที่ถูกเพิ่มประสิทธิภาพให้ดีกว่ามาตรฐานและข้อจำกัดที่ถูกกำหนดโดย NIST

2.2.4 งานวิจัย AES-512 : 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation [13]

Abidalrahman Moh'd, Yaser Jararweh และ Lo'ai Tawalbeh ได้วิจัยการเพิ่มประสิทธิภาพอัลกอริทึม AES โดยได้นำเสนอการเข้ารหัสแบบ AES ในรูปแบบใหม่ซึ่งจะช่วยเพิ่มระดับความปลอดภัยของการเข้ารหัส งานวิจัยนี้ศึกษาโดยใช้บล๊อกขนาด 512 บิต และใช้คีย์ขนาด 512 บิต บน VHDL นำมาเข้ารหัสด้วยอัลกอริทึม AES จำนวน 10 รอบ เมื่อเปรียบเทียบวิธีการนี้กับการเข้ารหัส AES-128 แบบเดิมด้วยฮาร์ดแวร์แบบเดียวกันนั้นพบว่า การเพิ่มขนาดบล๊อกทำให้ปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลา มีค่าเพิ่มขึ้นประมาณ 230% และจำนวนคีย์ที่ยาวขึ้นช่วยให้อัลกอริทึมมีความปลอดภัยในระดับที่สูงขึ้น

จากงานวิจัยข้างต้น จะเห็นได้ว่าการเพิ่มประสิทธิภาพอัลกอริทึม AES ที่ขนาดบล๊อก 512 บิต มีประสิทธิภาพที่ดีขึ้นจริง ดังนั้นผู้จัดทำวิทยานิพนธ์จึงนำแนวคิดนี้มาศึกษาเพิ่มเติมบนหน่วยประมวลผลหลายแกนชนิด 64 บิต เพื่อให้ปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลาเพิ่มขึ้น และเพิ่มขนาดของกุญแจ พร้อมกับกระบวนการยืนยันตัวตนหลายปัจจัย เพื่อให้มีความปลอดภัยมากยิ่งขึ้น

2.2.5 งานวิจัย Engineer Privacy [14]

Sarah Spiekermann และ Lorrie Faith Cranor ได้นำเสนอประเด็นด้านความเป็นส่วนตัวและแนวทางปฏิบัติสำหรับการสร้างระบบความเป็นส่วนตัว โดยกล่าวถึงปัญหาความเป็นส่วนตัวที่เกิดขึ้นในสังคมชาวอเมริกันทีมีความกังวลมากขึ้น เกี่ยวกับกฎหมายที่มีอยู่เดิมและการปฏิบัติในระดับที่เหมาะสมกับความเป็นส่วนตัวของผู้บริโภค

วิศวกรรมความเป็นส่วนตัว (Engineering Privacy) ได้แบ่งออกเป็น 2 ประเภท คือ

1) ความเป็นส่วนตัวโดยนโยบาย (Privacy-by-Policy) คือ วิธีการเก็บข้อมูลส่วนบุคคลโดยมุ่งเน้นไปที่นโยบายดำเนินการแจ้งให้ทราบ

2) ความเป็นส่วนตัวโดยสถาปัตยกรรม (Privacy-by-Architecture) คือ การลดการเก็บข้อมูลส่วนบุคคลที่สามารถระบุตัวตนได้และเน้นการเก็บข้อมูลแบบไม่ระบุตัวตน (Anonymization) บนการจัดเก็บข้อมูลและประมวลผลทางฝั่งผู้ใช้งาน (Client) ซึ่งจะมีความเป็นส่วนตัวระดับสูงและน่าเชื่อถือมาก

จากการศึกษานี้พบว่าในปัจจุบันหลักการความเป็นส่วนตัว โดยนโยบาย (Privacy-by-Policy) ได้รับการตอบรับจากภาคธุรกิจจำนวนมากเพราะว่าไม่ได้รบกวนรูปแบบธุรกิจที่มีอยู่เดิม โดย

อาศัยหลักการนโยบายความเป็นส่วนตัวซึ่งกำหนดขึ้นโดยบริษัทหรือผู้ให้บริการ เพื่อให้สามารถเข้าถึงข้อมูลความเป็นส่วนตัวของผู้ใช้บริการได้

จากงานวิจัยนี้ ทำให้ผู้จัดทำวิทยานิพนธ์เล็งเห็นว่า จากเครือข่ายสังคมในยุคปัจจุบัน (Social network) ที่ผู้บริโภคบางส่วนพร้อมที่จะเสียสละความเป็นส่วนตัวของพวกเขาเพื่อแลกกับความสะดวกสบาย เช่น บริการเครือข่ายทางสังคมออนไลน์ เฟสบุ๊ค (Facebook) หรือ บริการฝากไฟล์ของกูเกิ้ล (Google) โดยในการใช้งานบางฟังก์ชั่นผู้ใช้งานจะต้องยินยอมให้ผู้อื่นสามารถเข้าถึงข้อมูลส่วนตัวของตนเองได้ และบางผู้ให้บริการเองก็มีสิทธิ์ในการเข้าถึงไฟล์นั้นๆ ได้ เนื่องด้วยนโยบายของผู้ให้บริการเรื่องลิขสิทธิ์ของไฟล์ โดยที่ผู้บริโภคไม่สนใจที่จะอ่านนโยบาย (Policy) ของการใช้งาน ดังนั้นผู้วิจัยจึงคาดหวังว่าเครื่องมือต้นแบบ (Tool Utility Prototype) ที่บรรจุอัลกอริทึมเสริมประสิทธิภาพ AES จากวิทยานิพนธ์นี้ จะช่วยในกระบวนการปกป้องความเป็นส่วนตัว (Privacy) โดยการเข้ารหัสไฟล์หรือเอกสารใดๆ ก่อนส่งขึ้นไปจัดเก็บยังผู้ให้บริการต่าง ๆ อาทิเช่น กูเกิ้ล ดรอบบ๊อก (Dropbox) ซึ่งสามารถทำได้ง่ายขึ้น บนอุปกรณ์สมาร์ทโฟน (iPhone, iPad or Android)

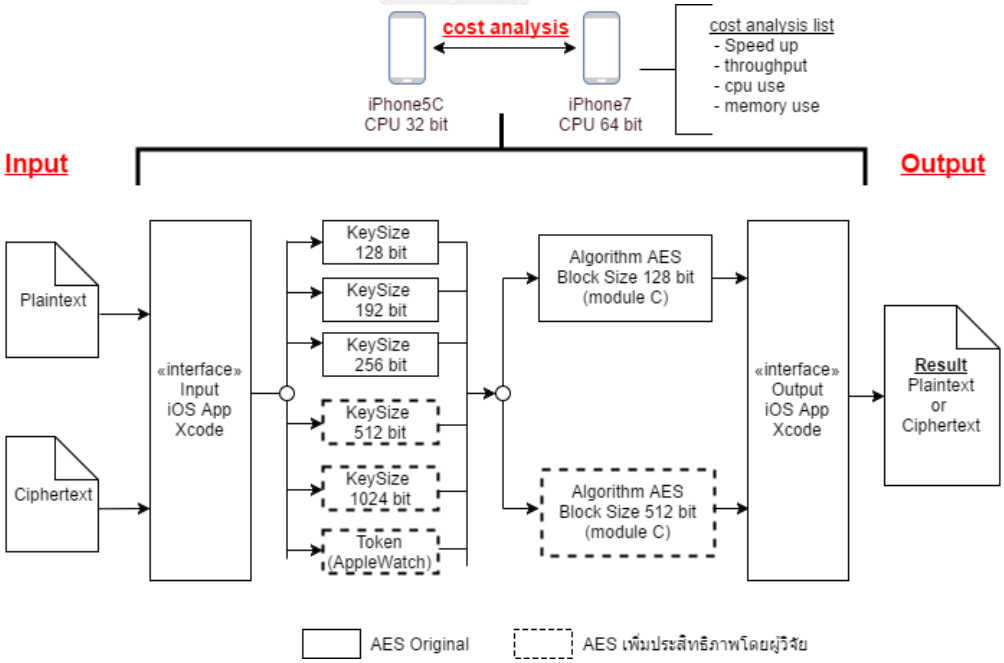
บทที่ 3 วิธีดำเนินการวิจัย

3.1 แนวความคิด

จากทฤษฎีและงานวิจัยที่กล่าวมาข้างต้นผู้วิจัยมีความสนใจที่จะสร้างและนำเสนอการวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต ซึ่งมี flow architecture ดังนี้

1. Interface การใช้งานเป็น iOS Application เขียนโดย XCode ทำงานบน iPhone โดยมีอัลกอริทึม AES-128 และ AES-512 ที่เขียนโดยภาษา C เป็น Library เบื้องหลัง
2. อินพุต (Input) จะใช้ Plaintext หากต้องการเข้ารหัส หรือ ใช้ Ciphertext หากต้องการถอดรหัส
3. ผู้ใช้เลือกขนาดกุญแจที่จะเข้ารหัสหรือถอดรหัส และขนาด Block-Size จากหน้า Interface
4. เอาต์พุต (Output) จะเป็น Ciphertext กรณีเข้ารหัส หรือ เป็น Plaintext กรณีถอดรหัส
5. แสดงผล Cost Analysis ที่เกิดจากการประมวลผลบนอุปกรณ์ iPhone

ดังแสดงในภาพที่ 2



ภาพที่ 2 แสดง flow architecture ภาพรวมของการเปรียบเทียบการเข้ารหัสระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต

3.2 การออกแบบ

กระบวนการออกแบบอัลกอลิทึมประกอบด้วย 2 ส่วน คือ การ implement algorithm AES เป็นรูปแบบ 512 บิต และ กระบวนการสร้างกุญแจโทเค็นโดยวิธีสกายเทล

3.2.1 การ implement algorithm AES เพิ่มประสิทธิภาพเป็นรูปแบบ 512 บิต

การ implement algorithm AES เป็นรูปแบบ 512 บิต มี 7 กระบวนการ ดังนี้

1) กระบวนการขยายบล็อก

เป็นกระบวนการเพื่อเพิ่มประสิทธิภาพปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลา ซึ่งแต่เดิมการประมวลผลที่ 1 state ของ AES จะมีขนาด 128 บิต โดยมีลักษณะเป็นอาร์เรย์ 2 มิติ บนข้อจำกัดมาตรฐานว่าในแต่ละแถวจะมีอยู่ Nb ไบต์ โดย Nb = 4 ดังนั้นการจะทำให้ state มี throughput ที่มากขึ้น จึงต้องการเพิ่มแถวและคอลัมน์เป็นสองเท่าจากเดิม ดังแสดงใน ตารางที่ 3

ตารางที่ 3 เปรียบเทียบขนาดของ State ระหว่างอัลกอริทึม AES-128 และ AES-512

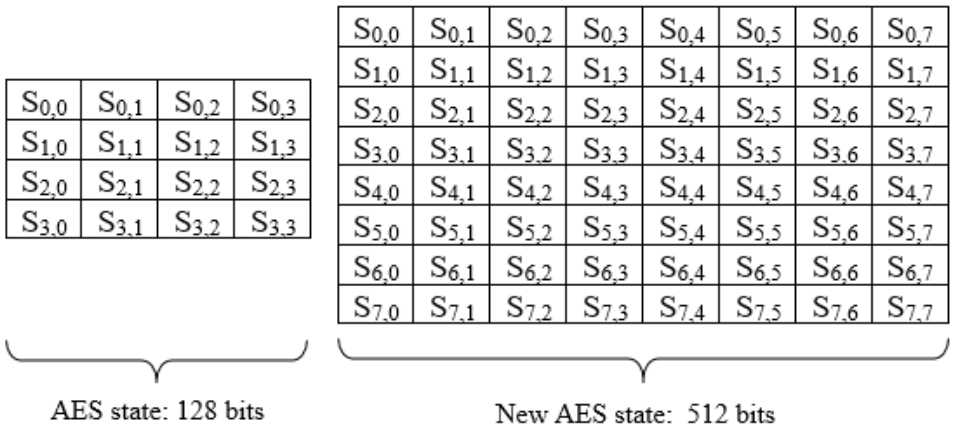
AES	AES Blocksize 128 bit	AES Blocksize 512 bit
Nb	4	8
Row	4	8
Column	4	8
Byte	16	64
Bit	128	512

จากตารางข้างต้นอัลกอริทึมเสริมประสิทธิภาพใหม่สามารถเขียนอยู่ในรูปอาร์เรย์ 2 มิติ ได้ดังต่อไปนี้

กำหนดให้ S = ไบต์แต่ละตัวในอาร์เรย์ ;

r = หมายเลขแถว | 0 <= r < 8 ;

c = หมายเลขคอลัมน์ | 0 <= c < 8;



ภาพที่ 3 แสดงลักษณะ state เปรียบเทียบขนาดของบล็อก ระหว่าง AES-128 และ AES-512

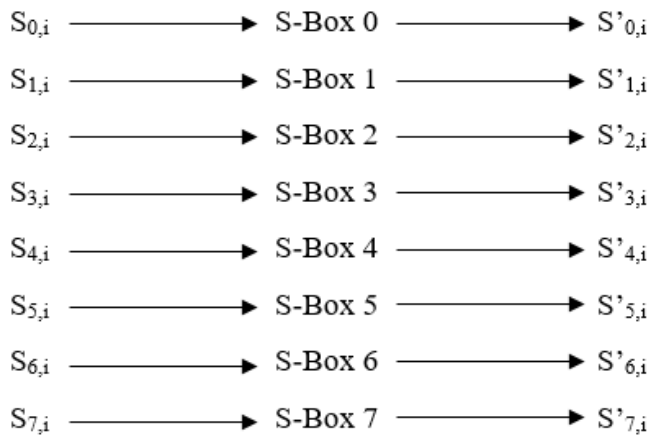
หลังจากได้ state หรือบล๊อกของอัลกอริทึมเสริมประสิทธิภาพใหม่ขนาด 512 บิต แล้ว ผู้วิจัยจึงนำมาสร้างเป็นฟังก์ชั่น AESBlock512_Encryption() สำหรับการเข้ารหัสโดยสามารถเขียนในรูป pseudo code ได้ดังนี้

```
( byte output[8*Nb] ) AESBlock512_Encryption ( byte input[8*Nb] ,Word
w[Nb*(Nr+1)] )
BEGIN
        byte state[8,Nb]
        State = input
        AddRoundKey(state, w[0,Nb-1])
        FOR(round = 1 step 1 to Nr-1 )
                SubBytes(state)
                ShiftRow(state)
                Mixcolumns(state)
                AddRoundKey(state, w[round*Nb, (round+1)*Nb-1] )
        END FOR LOOP
                SubBytes(state)
                ShiftRow(state)
                AddRoundKey(state, w[round*Nb, (round+1)*Nb-1] )
        Return output = state;
END
```

โดยผู้วิจัยจะอธิบายลำดับ SubBytes(), ShiftRow(), Mixcolumns(), AddRoundKey() ของอัลกอริทึม AES ในลำดับถัดไป

1) กระบวนการแทนที่ไบต์ (SubBytes)

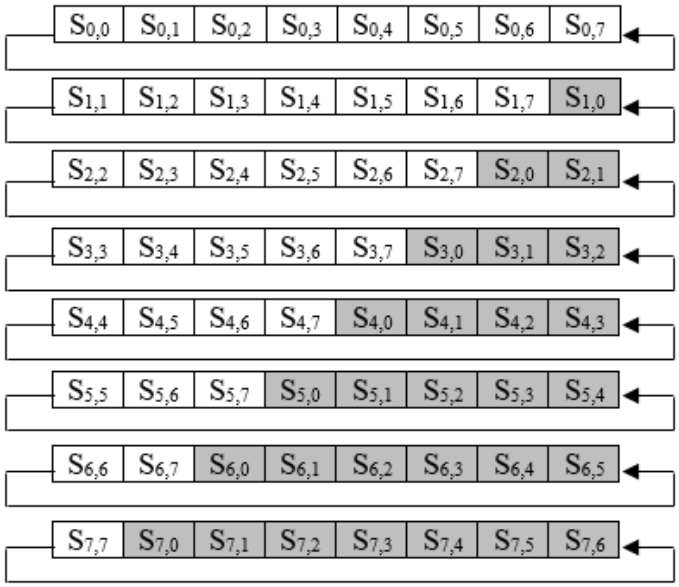
หลังจากผู้วิจัยได้สร้างบล๊อกอาร์เรย์ 2 มิติ ขนาด 512 บิต หรือ 64 ไบต์ ขึ้นมาแล้วนั้น ซึ่งในแต่ละไบต์ จะถูกแทนที่อย่างอิสระต่อกัน โดยใช้ตารางการแทนที่ S-Box จนครบทั้ง 64 ไบต์ ดังแสดงในภาพที่ 4



ภาพที่ 4 แสดงตัวอย่างการแทนที่แต่ละบิตโดยใช้ตารางการแทนที่ S-Box

2) กระบวนการเลื่อนแถว (ShiftRow)

หลังจากผ่านกระบวนการแทนที่ไบต์ เพื่อเปลี่ยนค่าบิตของบล็อก ผลลัพธ์ที่ได้ จะนำไปสู่กระบวนการเลื่อนแถว ซึ่งมีวิธีการคือ ทำการเลื่อนไบต์ แต่ละแถวไปทางซ้ายในลักษณะวนกลับ โดยที่แถวแรกยังคงตำแหน่งเดิม แถวที่ 2 จะเลื่อนไป 1 ตำแหน่ง แถวที่ 3 จะเลื่อนไป 2 ตำแหน่ง แถวถัดไปก็จะเลื่อนเพิ่มขึ้นอีกแถวละ 1 ตำแหน่ง ไปจนถึงแถวสุดท้าย ดังแสดงในภาพที่ 5



ภาพที่ 5 แสดงกระบวนการเลื่อนแถว

3) กระบวนการผสมคอลัมน์ (MixColumns)

หลังจากผ่านกระบวนการเลื่อนแถวเพื่อเปลี่ยนค่าบิตของบล็อก ผลลัพธ์ที่ได้จะนำไปสู่กระบวนการแปลงข้อมูลที่ละคอลัมน์ ซึ่งมีวิธีการคือ ในแต่ละคอลัมน์จะประกอบด้วยชุดข้อมูลจำนวน 8 ไบต์ ที่พิจารณาเป็นพหุนามบนฟิลด์ GF(2^n) โดยนำพหุนามนี้ไปคูณด้วยพหุนาม a(x) ที่มีค่าตายตัวเท่ากับ

$$a(x) = [02]x^7 + [01]x^6 + [03]x^5 + [01]x^4 + [01]x^3 + [01]x^2 + [01]x^1 + [01]x^0$$

ซึ่งสามารถจัดในรูปของเมทริกซ์ได้เป็น

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \\ S'_{4,c} \\ S'_{5,c} \\ S'_{6,c} \\ S'_{7,c} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 03 & 01 & 01 & 01 & 01 & 01 \\ 01 & 03 & 01 & 01 & 01 & 01 & 01 & 02 \\ 03 & 01 & 01 & 01 & 01 & 01 & 02 & 01 \\ 01 & 01 & 01 & 01 & 01 & 02 & 01 & 03 \\ 01 & 01 & 01 & 01 & 02 & 01 & 03 & 01 \\ 01 & 01 & 01 & 02 & 01 & 03 & 01 & 01 \\ 01 & 01 & 02 & 01 & 03 & 01 & 01 & 01 \\ 01 & 02 & 01 & 03 & 01 & 01 & 01 & 01 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \\ S_{4,c} \\ S_{5,c} \\ S_{6,c} \\ S_{7,c} \end{bmatrix}$$

ภาพที่ 6 แสดงเมตริกซ์การคูณข้อมูลในคอลัมน์ด้วยพหุนาม a(x)

หรือสามารถเขียนในรูปการคูณพหุนามได้ดังนี้

$$S'_{0,c} = ([02]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([03]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([01]*S_{7,c})$$

$$S'_{1,c} = ([01]*S_{0,c}) \oplus ([03]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([02]*S_{7,c})$$

$$S'_{2,c} = ([03]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([02]*S_{6,c}) \oplus ([01]*S_{7,c})$$

$$S'_{3,c} = ([01]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([02]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([03]*S_{7,c})$$

$$S'_{4,c} = ([01]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([02]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([03]*S_{6,c}) \oplus ([01]*S_{7,c})$$

$$S'_{5,c} = ([01]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([02]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([03]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([01]*S_{7,c})$$

$$S'_{6,c} = ([01]*S_{0,c}) \oplus ([01]*S_{1,c}) \oplus ([02]*S_{2,c}) \oplus ([01]*S_{3,c}) \oplus ([03]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([01]*S_{7,c})$$

$$S'_{7,c} = ([01]*S_{0,c}) \oplus ([02]*S_{1,c}) \oplus ([01]*S_{2,c}) \oplus ([03]*S_{3,c}) \oplus ([01]*S_{4,c}) \oplus ([01]*S_{5,c}) \oplus ([01]*S_{6,c}) \oplus ([01]*S_{7,c})$$

4) กระบวนการบวกค่ากุญแจ (AddRoundKey)

เป็นการบวกค่ากุญแจเข้าไปในบล๊อก ในแต่ละรอบที่ทำการเข้ารหัส จำนวนรอบที่ใช้ในการเข้ารหัสแสดงดังตารางที่ 3 ค่าของกุญแจที่ผู้ใช้กำหนดจะถูกนำไปผ่านกระบวนการขยายขนาดอักขระกุญแจ (Key Expansion) อยู่ในรูปแบบตารางอักขระกุญแจซึ่งจะมีทั้งหมด Nb(Nr+1) เวิร์ด ซึ่งตารางอักขระกุญแจที่ได้จะนำไปใช้ทำ XOR กับ State ทีละคอลัมน์ จนครบ Nb คอลัมน์ในแต่ละรอบของการ AddRoundKey ดังแสดงในสมการด้านล่าง

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, \ldots, S'_{7,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, \ldots, S_{7,c}] \oplus [W_{round*Nb+c}]$$

ตารางที่ 4 แสดงจำนวนรอบในการเข้ารหัสของกุญแจขนาดต่าง ๆ

Key size	Round Number AES-128 bits	Round Number AES-512 bits
128bits	10	4
192 bits	12	5
256 bits	14	6
512 bits	22	10
1024 bits	38	18

5) กระบวนการขยายขนาดกุญแจ (Key Expansion)

เป็นกระบวนการเตรียมตารางอักขระกุญแจเพื่อใช้ในกระบวนการเข้าและถอดรหัสในแต่ละรอบ ตารางกุญแจหลังจากผ่านขั้นตอนการขยายขนาดแล้วจะมีทั้งหมด Nb(Nr+1) เวิร์ด ซึ่งเขียนแสดงด้วย [wi] โดยให้กลุ่มของเวิร์ดที่มีค่า i น้อยกว่า Nk ให้มีค่าเท่ากับค่ากุญแจไซเฟอร์ตามที่ผู้ใช้กำหนดส่วนค่าอื่น ๆ ที่เหลือจะถูกแบ่งเป็น 2 กลุ่ม คือ

1. กลุ่มของเวิร์ดที่ค่า i เป็นจำนวนเท่าของ Nk ให้คำนวณดังนี้

   $w[i] = \{Subword(RotWord(w[i-1])) \oplus Rcon[i/Nk]\} \oplus w[i-Nk]$

2. กลุ่มของเวิร์ดที่เหลือที่ค่า i ไม่เป็นจำนวนเท่าของ ให้คำนวณดังนี้

   $w[i] = w[i-1] \oplus w[i-Nk]$

ซึ่งในกระบวนการนี้จะมีการใช้ฟังก์ชั่น 2 ฟังก์ชั่น ได้แก่ SubWord() และ RotWord() โดยฟังก์ชั่น SubWord() จะรับเวิร์ดขนาด 8 ไบต์เพื่อนำไปแปลงโดยตารางแทนที่ S-Box ส่วนฟังก์ชั่น RotWord() จะรับเวิร์ด $[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$ เข้ามาแล้วทำการเลื่อนตำแหน่งแบบวนกลับได้เป็น

$[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_0]$ นอกจากนี้ยังมีการใช้อาร์เรย์ของเวิร์ดที่เป็นค่าคงที่เรียกว่า Rcon[i] ซึ่งอาร์เรย์นี้บรรจุค่า $[x^{i-1}, \{0,0\}, \{0,0\}, \{0,0\}]$ โดยที่ x มีค่าเป็น 02_{hex} และ i มีค่าตั้งแต่ 1 ถึง 10

สามารถเขียนในรูปแบบ *pseudo code* ได้ดังนี้

```
KeyExpansion ( byte key[8*Nk] ,word w[Nb*(Nr+1)] )
BEGIN
        word = temp
        i = 0
        WHILE ( i < Nk )
                w[i] = word(key[8*1] , key[8*i+1] , key[8*1+2] , key[8*1+3] ,
                        key[8*1+4] , key[8*1+5] , key[8*1+6] , key[8*1+7])
                i = i+ 1
        END WHILE
       i = Nk
       WHILE ( i < Nb * (Nr+1))
                temp = w[i-1]
                IF (i mod Nk == 0)
                        temp = SubWord(RotWord(temp)) XOR Rcon[i/NK]
                END IF
                w[i] = w[i-Nk] XOR temp;
                 i  =  i + 1
     END WHILE
END
```

6) กระบวนการถอดรหัส(decryption)

เป็นกระบวนการนำไซเฟอร์ที่เข้ารหัสมาทำกระบวนการย้อนกลับขั้นตอนเข้ารหัสที่กล่าวมาในช่วงแรกทั้งหมด โดยเขียนใหม่ได้เป็น

- InvShiftRows() เป็นกระบวนการแปลงผกผันกับ ShiftRows() โดยการขยับกลับด้านมาทางขวา
- InvSubBytes() เป็นกระบวนการแปลงผกผันกับ SubBytes() ซึ่งเป็น *S-Box* อีกตาราง

- InvMixColumns() เป็นกระบวนการแปลงผกผันกับ MixColumns() ซึ่งอยู่ในรูปของพหุนามของแต่ละคอลัมน์ โดยนำพหุนามที่ได้ไปคูณด้วยพหุนาม a'(x) ที่มีค่าตายตัวเท่ากับ

$$a'(x) = [0E]x^7 + [01]x^6 + [09]x^5 + [01]x^4 + [0D]x^3 + [01]x^2 + [0B]x^1 + [01]x^0$$

- AddRoundKey() ใช้ฟังชั่นเดิมมาทำการแปลงผกผัน

กระบวนการถอดรหัส สามารถเขียนในรูป pseudo code ได้ดังนี้

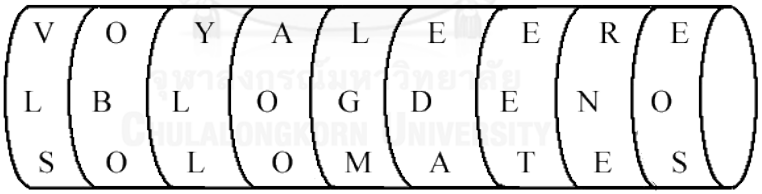
```
( byte output[8*Nb] ) AESBlock512_Decryption ( byte input[8*Nb] ,Word
w[Nb*(Nr+1)] )
BEGIN
        byte state[8,Nb]
        State = input
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
        FOR(round = Nr-1 step -1 downto 1 )
                InvShiftRows(state)
                InvSubBytes(state)
                AddRoundKey(state, w[round*Nb, (round+1)*Nb-1] )
                InvMixColumns(state)
        END FOR
                InvShiftRows(state)
                InvSubBytes(state)
                AddRoundKey(state,w[0,Nb-1])
         Return output = state;
END
```

3.2.2 กระบวนการสร้างกุญแจโทเค็นโดยวิธีสกายเทล (Token scytale generator)

การกำหนดและจดจำรหัสผ่าน (password) ของผู้ใช้งานนั้นโดยปกติ ควรมีความยาวไม่ต่ำกว่า 8 ตัวอักษร และไม่เป็นกลุ่มคำที่มีความหมายหรือคาดเดาได้ง่ายเพื่อหลีกเลี่ยงการเดากุญแจรหัสผ่าน หรือการสุ่มเดารหัสผ่านจากเครื่องมือ (dictionary attack) ซึ่งในทางปฏิบัติแล้วอาจเป็นเรื่องยาก เพราะผู้ใช้งานไม่ได้กำหนดและจำ password แค่สิ่งๆเดียว และปริมาณจำนวน password ที่เยอะและยาวอาจทำให้ผู้ใช้เกิดความสับสนหรือลืม ซึ่งก็เป็นเหตุการณ์ที่อาจเกิดขึ้นได้จริงเมื่อ

ระยะเวลาผ่านไปนาน ดังนั้นการใช้ Token ช่วยในการเก็บรหัสผ่านจึงเป็นที่นิยมแนวทางหนึ่ง เพื่อช่วยให้ผู้ใช้ไม่ต้องจดจำรหัสผ่าน

โดยพื้นฐาน กระบวนการสร้างรหัส (key-password) จะอยู่บนแนวคิดที่ว่า จำนวน key ที่ยาวและซับซ้อน มีผลโดยตรงต่อกระบวน คาดเดารหัสผ่าน (brute-force) ดังนั้นเนื้อหาในส่วนนี้จึงนำเสนออัลกอริทึมสร้างชุดกุญแจรหัสผ่าน โดยอาศัย กระบวนการสร้างรหัส 2 ประเภท คือรหัสผ่านที่เกิดจากการกำหนดจากผู้ใช้งานหรือดึงสตริง อัตลักษณ์ (Identity) จากผู้ใช้งาน และอย่างที่สองคือรหัสผ่านที่เกิดจากการสุ่มชุดสตริงจากสถาปัตยกรรมฮาร์ดแวร์ เช่น Intel, AMD, ARM โดยผู้วิจัยนำแนวคิดทั้งสองมารวมกัน (Hybrid) ในการสร้างกุญแจ (token generator) ออกมาหนึ่งชุด โดยผู้วิจัยนำไอเดียหลักการเข้ารหัสแบบโบราณ (Classic Cryptography) ที่มีชื่อว่า สกายเทล (Scytale) มาปรับใช้เพื่อสร้างโทเค็นให้มีความแตกต่างจากการสุ่มชุดสตริงจากสถาปัตยกรรมฮาร์ดแวร์เพียงอย่างเดียว หลักการดั้งเดิมของสกายเทล คือจะนำแผนผ้ามาพันรอบแท่งถอดรหัสเพื่ออ่านข้อความ ตัวอักษรตำแหน่งที่มีระยะห่างเท่ากับเส้นรอบวงของแท่งถอดรหัสจะเรียงต่อกันและถูกอ่านได้ ดังนั้นผู้วิจัยจึงนำหลักการนี้มาใช้เพื่อผสานรหัสทั้งสองชนิดโดยใช้ชุดอักขระที่ถูกบรรจุอยู่ใน โทเค็นเป็นฐานของรหัสผ่าน โดยแต่ละตัวอักษรของรหัสจากกระบวนการจำจะถูกแทรกลงไป ซึ่งแต่ละไบต์มีระยะห่างเท่ากับความยาวของรหัสนั้น เช่น รหัสจากกระบวนการจำมีความยาวเท่ากับ 7 ตัวอักษร ดังนั้น อักขระตัวแรกจะถูกแทรกลงไปยังตำแหน่งที่ 7 ของชุดอักขระจาก โทเค็นอักขระตัวที่สองจะถูกแทรกลงไปยังตำแหน่งที่ 14 ของอักขระจาก โทเค็นเป็นเช่นนี้ไปจนถึงตัวสุดท้าย



ภาพที่ 7 ลักษณะการเข้ารหัสตามหลักการของ Scytale

ตารางที่ 5 เปรียบเทียบลักษณะของ Token และ Scytale

Token	เปรียบเสมือน	ชุดอักขระบนเส้นผ้าของสกายเทล
ชุดอักขระที่เป็นกระบวนการจำ	เปรียบเสมือน	อักขระรหัสผ่านที่ฝังอยู่บนเส้นผ้าของสกายเทล
ความยาวของชุดอักขระที่เป็นกระบวนการจำ	เปรียบเสมือน	เส้นรอบวงของสกายเทล

ตัวอย่างขั้นตอนลำดับการผสานอักขระแสดงดังตารางที่ 6

ตารางที่ 6 แสดงตัวอย่าง ลำดับการผสานชุดอักขระ

1	Password String Ex : “LoveCat” (7 byte)	Token 512 bit (64 byte) and 1024 bit (128 byte) B_0 B_1 B_2 B_3 B_4 B_5 B_6 B_7 .... B_n
2	InsertKeytoToken	B_0 .... B_5 **L** B_7 .... B_{12} **o** B_{14} .... B_{47} **t** B_{49} .... B_n
3	Result key 512 bit	a .... Y **L** u .... M **o** @ .... 8 **t** $ .... B_{63}
	Result key 1024 bit	a .... Y **L** u .... M **o** @ .... 8 **t** $ .... B_{127}

จากกระบวนการข้างต้น สามารถเขียนในรูปรหัสเทียม (Pseudo code) ได้ดังนี้

```
(byte newpassword)Gen_KeyScyTale(byte token, byte password)
BEGIN
    Int : len_token = length(token)
    int : len_password = length(password)
    int : x = 0
    FOR i = all the characters in the password
        x = x + len_password
        IF x <= len_token
            token[x-1] = password[i]
        ENDIF
    ENDFOR
    newpassword = token
END
```

โดยกระบวนการสร้างรหัสโทเค็นที่นำเสนอนี้มีกระบวนการซับซ้อนกว่าวิธีการการนำชุดอักขระทั้งสองชนิดมาต่ออักขระกันอย่างเรียบง่าย ซึ่งอาจมีจุดบกพร่องตรงอักขระที่กำหนดโดยผู้ใช้งานทั่วไป (User Error) กำหนดเป็นคำหรือประโยคที่มีความหมายเช่น “ILOVECAT”, เมตริกซ์

(*matrix)* “MyBirthDay1978” ซึ่งนำไปสู่การทำสถิติเพื่อคาดเดารหัสผ่าน (dictionary attack) และแยกแยะ ชุดอักขระทั้งสองชนิดได้โดยง่าย

3.3 การพัฒนา

จากกระบวนการออกแบบและขยายขนาดบล็อกที่กล่าวมาข้างต้น ผู้วิจัยได้สร้างเครื่องมือ (Utility Tool) ขึ้นมาเพื่อทดสอบกระบวนการ และวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ โดยเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต ได้ดังนี้

3.3.1 เครื่องมือที่ใช้

ภาษาในการพัฒนา

- C language
- Objective-C (Xcode)

อุปกรณ์ที่ใช้ทดสอบ

- iPhone5C
- iPhone7

3.3.2 วิธีการพัฒนา

ในกระบวนการ implement อัลกอริทึม AES ขนาดบล็อก 512 บิต บนอุปกรณ์มือถือ iPhone7 และ iPhone 5C ซึ่งมี specification ดังตารางที่ 7 นั้น สามารถแบ่งการพัฒนาได้เป็น สองส่วน คือ ส่วน interface ซึ่งเป็นรูปแบบการใช้งานบนอุปกรณ์มือถือ และส่วนการประมวลผลของอัลกอริทึม AES ที่ใช้สำหรับการเข้าและถอดรหัส

ตารางที่ 7 แสดงข้อมูล Specification ของ iPhone5C และ iPhone7

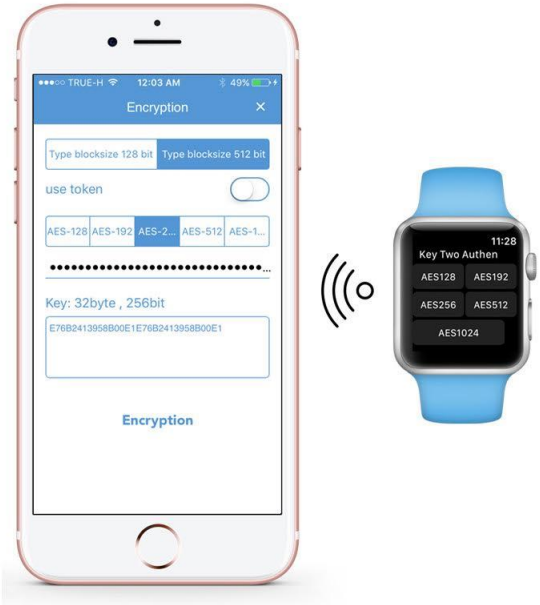
Specification	iPhone5C	iPhone7
CPU	ARM A6 32-bit architecture Dual-core processor 1.3 GHz	ARM A10 64-bit architecture Quad-core processor 2.34 GHz
Ram	1 GB	2 GB
iOS	10.2	10.2

1) ส่วนของ interface ผู้ใช้สามารถเลือกขนาดบล็อก และขนาดกุญแจ สำหรับไฟล์ที่ต้องการเข้ารหัส มีการแสดงผล log file ของการเข้ารหัสและถอดรหัสแต่ละครั้งซึ่งแสดงให้เห็นถึงขนาดของข้อมูล ขนาดบล็อก และขนาดกุญแจที่ใช้ รวมถึง throughput และระยะเวลาในการประมวลผล กระบวนการนี้พัฒนาโดยใช้ Xcode แบบ Native application เพื่อให้ดึงประสิทธิ์ภาพของ iOS บน iPhone5C และ iPhone7 ได้ดีที่สุด โดยมีลำดับขั้นตอนการใช้งานดังต่อไปนี้

ลำดับขั้นตอนการใช้งานอัลกอริทึมเพื่อเข้ารหัสบน iPhone ดังแสดงในภาพที่ 8

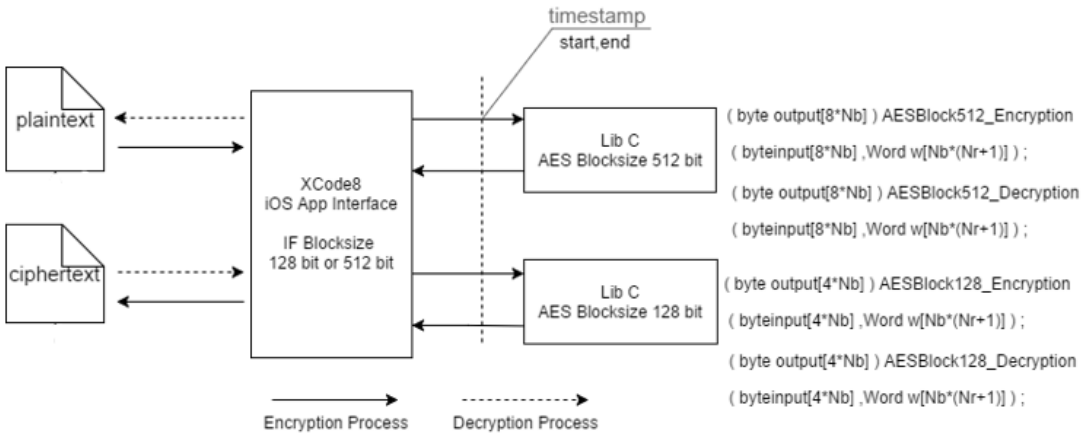
1. เลือก File ที่ต้องการเข้ารหัส/ถอดรหัส
2. เลือกขนาดบล็อกที่ต้องการเข้ารหัส คือ 128 บิต หรือ 512 บิต
3. เลือกขนาดกุญแจที่ต้องการใช้ คือ 128 บิต, 192 บิต, 256 บิต, 512 บิต หรือ 1024 บิต
4. เลือกว่าต้องการเปิดใช้ Token Authentication หรือไม่ ถ้าในกรณีเปิดใช้ จะใช้งานคู่กับ Apple Watch ในการเก็บค่า Key ที่ผ่านการสร้างกุญแจรูปแบบ Scytale ที่ได้กล่าวไปแล้วในตอนต้น

ภาพที่ 8 แสดง Interface การใช้งาน Algorithm บน iPhone



ภาพที่ 9 แสดง Interface กรณีใช้ Token Authentication

2) ส่วนของ Algorithm ในการเข้ารหัส พัฒนาโดยใช้ภาษา C มีลักษณะเป็น library ซึ่งจะถูกเรียกใช้งานโดย Xcode เมื่อทำการเข้ารหัสหรือถอดรหัส จะมีการบันทึกเวลาเริ่มต้นและเวลาสิ้นสุดเอาไว้ เพื่อใช้ในกระบวนการหาค่า throughput และค่า speedup ที่ได้ ดังแสดงในภาพที่ 10



ภาพที่ 10 แผนภาพแสดงกระบวนการทำงานในการเข้ารหัส และถอดรหัสบน iPhone

บทที่ 4 วิธีการทดลองและผลการทดลอง

4.1 วิธีการทดลอง

หลังจากทำการ Implement Algorithm ทั้งสองส่วนที่กล่าวมาข้างต้นแล้ว ผู้วิจัยจะได้ Application Tool Analysis ที่มีอัลกอริทึมเข้ารหัสแบบ AES ซึ่งทำงานบนหน่วยประมวลผลสำหรับอุปกรณ์แบบเคลื่อนที่ iPhone5C ที่ใช้ CPU 32 บิต และ iPhone7 ที่ใช้ CPU 64 บิต และทำการเปรียบเทียบระหว่างอัลกอริทึม AES บล็อกขนาด 128 บิต กับ 512 บิต โดย Tool จะแสดง Output ดังต่อไปนี้ออกทางหน้าจอ ดังภาพที่ 11

Type = Encryption / Decryption

Block Size = ขนาดบล็อกที่เลือกเข้ารหัส/ถอดรหัส

Key Size = ขนาดกุญแจที่เลือกเข้ารหัส/ถอดรหัส

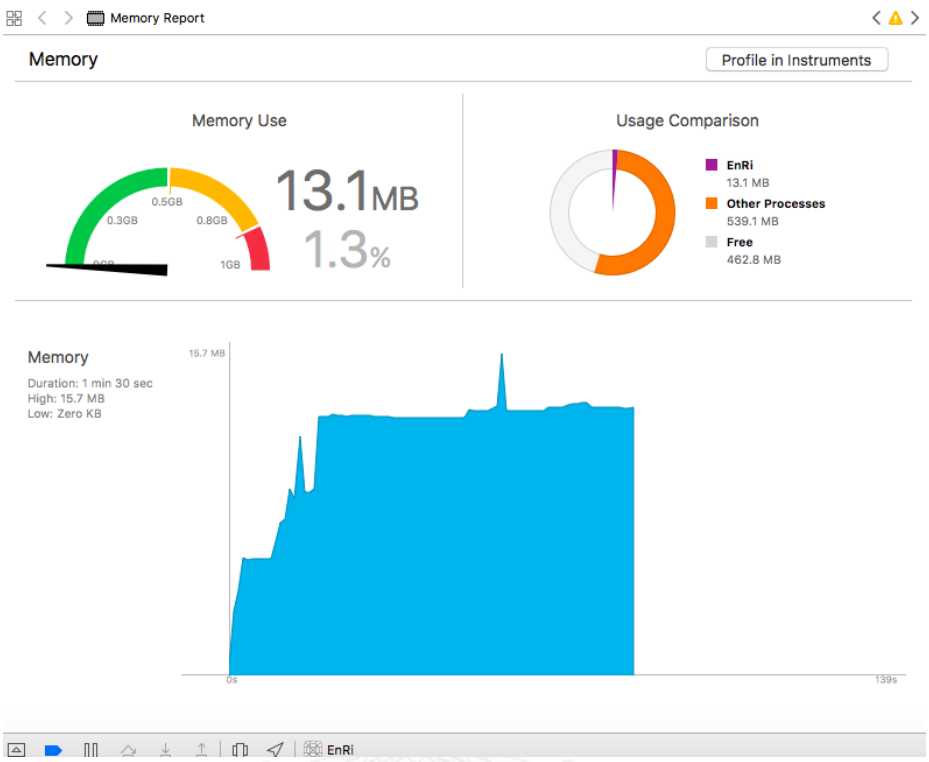
Time = เวลาที่ใช้ในการเข้ารหัส/ถอดรหัส

Throughput = ปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลา

Speed up = Throughput AES บล็อก 512 / Throughput AES บล็อก 128

ภาพที่ 11 แสดง Output interface ของ AES algorithm บน iPhone

การหาปริมาณการใช้ทรัพยากร ได้แก่ CPU และ RAM สามารถทำได้โดยใช้ Analysis tool ของโปรแกรม Xcode ดังแสดงในภาพที่ 12



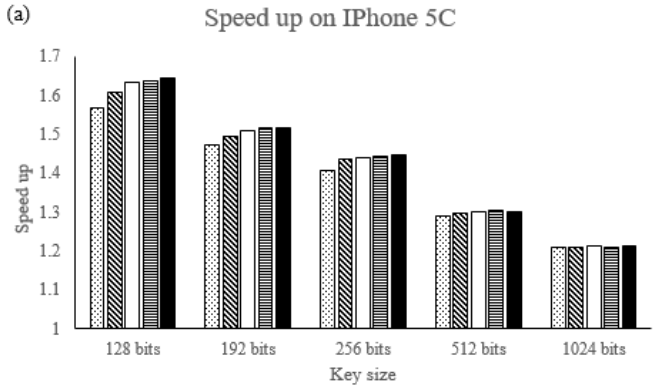
ภาพที่ 12 แสดง Analysis tool ของโปรแกรม Xcode

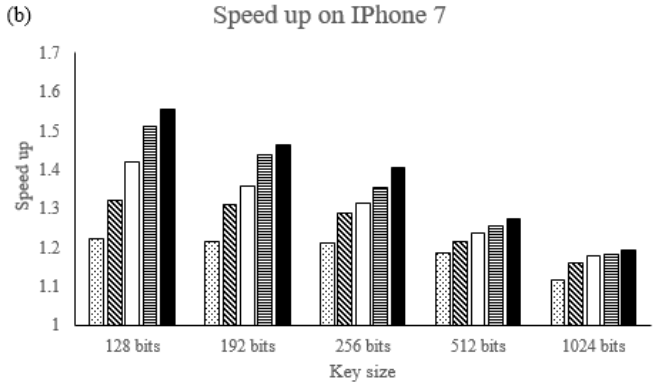
4.2 ผลการทดลอง

4.2.1 การเปรียบเทียบ Speed up ระหว่างอัลกอริทึม AES บล็อก 512 บิต และ 128 บิต

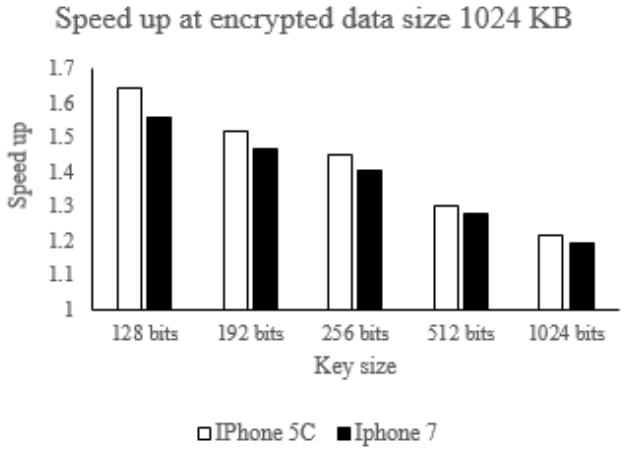
จากการศึกษาพบว่า อัลกอริทึม AES บล็อก 512 บิต มีความเร็วในการเข้ารหัสดีกว่า AES บล็อก 128 บิต ซึ่งความเร็วในการเข้ารหัสขึ้นอยู่กับอุปกรณ์ที่ใช้ เมื่อพิจารณาเปรียบเทียบ Speed up ของ AES บล็อก 512 บิต ต่อ AES บล็อก 128 บิต พบว่าจะมี Speed up เพิ่มขึ้นตามขนาดของข้อมูลที่ใช้ เมื่อเพิ่มขึ้นสูงสุดแล้วจะมีค่าคงที่ ซึ่งบน iPhone5C มีค่า Speed up สูงสุดเมื่อใช้ข้อมูลขนาด 256 KB และบน iPhone7 มีค่า Speed up สูงสุดเมื่อใช้ข้อมูลขนาด 1024 KB ดังภาพที่ 13

เมื่อทดสอบการเข้ารหัสข้อมูลขนาด 1024 KB โดยใช้กุญแจขนาด 128, 196, 256, 512 และ 1024 บิต พบว่า บน iPhone5C มีค่า Speed up 1.21 - 1.64 และบน iPhone7 มีค่า Speed up 1.19 – 1.55 โดยค่า Speed up จะลดลงตามความยาวของกุญแจที่ใช้ ดังภาพที่ 14





ภาพที่ 13 กราฟแสดง Speed up เมื่อใช้ข้อมูลในการเข้ารหัสตั้งแต่ 64 – 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7



ภาพที่ 14 กราฟแสดง Speed up เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

ตารางที่ 8 แสดงค่า Speed up เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

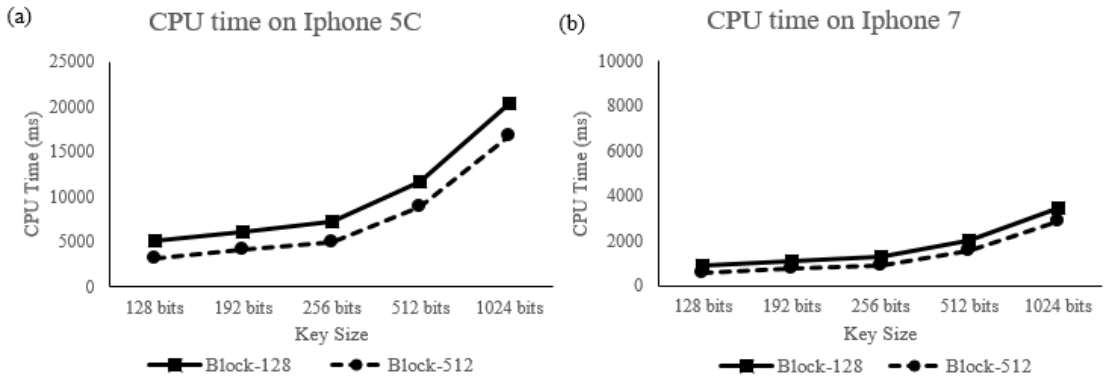
Key size	Speed up	
	iPhone5C	iPhone7
128bits	1.64	1.55
192 bits	1.51	1.46
256 bits	1.45	1.40
512 bits	1.30	1.27
1024 bits	1.21	1.19

4.2.2 การเปรียบเทียบ CPU Time ระหว่างอัลกอริทึม AES บล็อก 512 บิต และ 128 บิต

การวิเคราะห์การใช้ CPU ของอัลกอริทึม พบว่า เมื่อมีการเรียกใช้อัลกอริทึมทั้ง 2 แบบ จะมีอัตราการใช้ CPU 100% จนดำเนินการเสร็จสิ้น เมื่อพิจารณา CPU Time พบว่า

บน iPhone5C อัลกอริทึม AES บล็อก 512 บิตมี CPU Time น้อยกว่า AES บล็อก 128 บิต ประมาณ 2,008 – 3,584 มิลลิวินาที

บน iPhone7 อัลกอริทึม AES บล็อก 512 บิตมี CPU Time น้อยกว่า AES บล็อก 128 บิต ประมาณ 330 – 550 มิลลิวินาที ซึ่งการใช้กุญแจขนาดใหญ่ขึ้นจะเห็นความแตกต่างของ CPU Time ที่มากขึ้น ดังภาพที่ 15



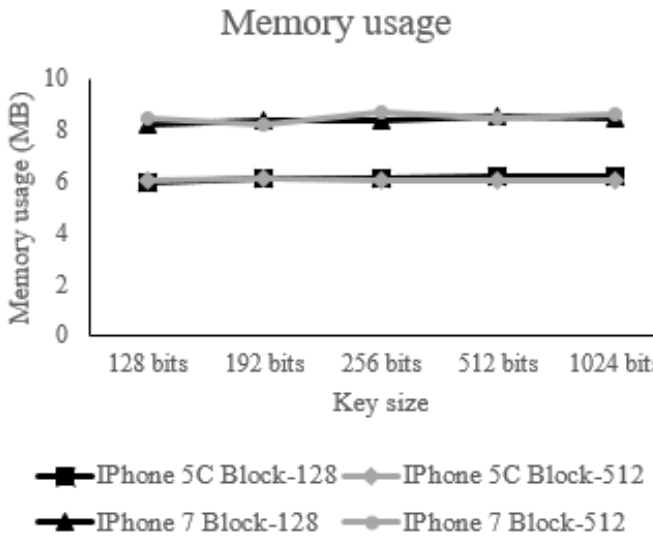
ภาพที่ 15 กราฟแสดง CPU Time เปรียบเทียบระหว่าง อัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

ตารางที่ 9 แสดงค่า CPU Time เปรียบเทียบระหว่าง อัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

Devices	Key size (bits)	CPU Time (ms)		
		AES-128	AES-512	Different time
iPhone5C	Key 128	5134	3126	2008
	Key 192	6207	4099	2108
	Key 256	7328	5060	2268
	Key 512	11664	8959	2705
	Key 1024	20356	16772	3584
iPhone7	Key 128	914	588	326
	Key 192	1106	756	350
	Key 256	1287	916	371
	Key 512	1993	1563	430
	Key 1024	3440	2887	553

4.2.3 การเปรียบเทียบการใช้หน่วยความจำระหว่างอัลกอริทึม AES บล็อก 512 บิต และ 128 บิต

ผลการศึกษาการใช้หน่วยความจำของอัลกอริทึม พบว่า เมื่อทำการเข้ารหัสข้อมูลที่ขนาดเท่ากัน อัลกอริทึมทั้ง 2 ชนิดมีการใช้หน่วยความจำไม่แตกต่างกัน แต่เมื่อพิจารณาอุปกรณ์ จะเห็นว่า iPhone7 มีการใช้หน่วยความจำมากกว่า iPhone5C ดังภาพที่ 16



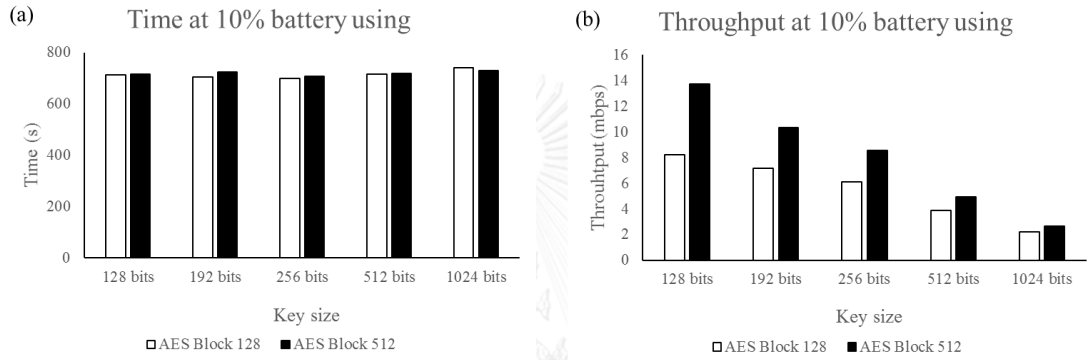
ภาพที่ 16 กราฟแสดงการใช้หน่วยความจำเปรียบเทียบระหว่าง อัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

ตารางที่ 10 แสดงค่าการใช้หน่วยความจำเปรียบเทียบระหว่าง อัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone5C และ iPhone7

Devices	Key size (bits)	Memory usage (MB)	
		AES-128	AES-512
iPhone5C	Key 128	5.97	6.07
	Key 192	6.13	6.10
	Key 256	6.13	6.06
	Key 512	6.20	6.00
	Key 1024	6.20	6.06
iPhone7	Key 128	8.17	8.46
	Key 192	8.40	8.17
	Key 256	8.33	8.67
	Key 512	8.53	8.46
	Key 1024	8.46	8.60

4.2.4 การเปรียบเทียบปริมาณการใช้แบตเตอรี่ระหว่าง AES บล็อก 512 บิต และ 128 บิต บน iPhone7

ผลการศึกษาปริมาณการใช้แบตเตอรี่ของอัลกอริทึม โดยเปรียบเทียบเวลาที่ทำให้แบตเตอร์รี่ลดลง พบว่า อัลกอริทึมทั้ง 2 ชนิดมีอัตราการใช้แบตเตอรี่ในการทำงานไม่แตกต่างกัน โดยอัตราการใช้พลังงานจะอยู่ที่ 11 – 12 นาที ต่อ 10% ของแบตเตอรี่ แต่ในอัตราการใช้พลังงานที่เท่ากันจะเห็นได้ว่าบล๊อกขนาด 512 บิต จะมีปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลาสูงกว่าบล๊อก 128 บิต ดังภาพที่ 17



ภาพที่ 17 กราฟแสดงเวลาที่ใช้เพื่อทำให้แบตเตอร์รี่ลดลง 10% และแสดง Throughput ที่ได้จากการทำงานของอัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone7

ตารางที่ 11 แสดงเวลาที่ใช้เพื่อทำให้แบตเตอรี่ลดลง 10% และแสดง Throughput ที่ได้จากการทำงานของอัลกอริทึม AES บล็อก 512 และ 128 บิต เมื่อใช้ข้อมูลในการเข้ารหัสขนาด 1024 KB โดยใช้กุญแจขนาด 128 – 1024 บิต บน iPhone7

Key size (bits)	Time (s)		Throughput (mbps)	
	AES-128	AES-512	AES-128	AES-512
Key 128	712 (00:11:52)	715 (00:11:55)	8.22	13.72
Key 192	703 (00:11:43)	723 (00:12:03)	7.15	10.33
Key 256	699 (00:11:39)	708 (00:11:48)	6.11	8.54
Key 512	716 (00:11:56)	718 (00:11:58)	3.86	4.94
Key 1024	740 (00:12:10)	729 (00:12:09)	2.19	2.66

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอ การวิเคราะห์ค่าใช้จ่ายของการทำการเข้ารหัสแบบ AES บนหน่วยประมวลผลสำหรับอุปกรณ์แบบเครื่องที่ ซึ่งแสดงการเปรียบเทียบทรัพยากรที่ใช้และประสิทธิภาพที่ได้ระหว่าง อัลกอริทึม AES บล็อกขนาด 512 และ 128 บิต โดยใช้กุญแจในการเข้ารหัสความยาว 128, 192, 256, 512 และ 1024 บิต บนอุปกรณ์แบบเคลื่อนที่ โดยใช้ iPhone 5C ซึ่งเป็น CPU ARM 32 บิต และ iPhone7 ซึ่งเป็น CPU ARM 64 บิต โดยสรุปประเด็นที่น่าสนใจดังต่อไปนี้

1) การเพิ่มขนาดของบล๊อกจาก 128 บิต เป็น 512 บิต

ในด้านของ Performance จะเห็นได้ว่าการเพิ่มขนาดของบล๊อกสามารถเพิ่มประสิทธิภาพของอัลกอริทึมได้ โดยผลการศึกษาแสดงให้เห็น Speed up ที่เพิ่มขึ้น และ CPU Time ที่ลดลง จึงส่งผลให้ค่าใช้จ่ายในการเข้ารหัสลดลง

ในด้านของ Security การเพิ่มขนาดของบล๊อกมีผลโดยตรงต่อการเรียงสับเปลี่ยนและความน่าจะเป็นที่เกิดขึ้นต่อกลุ่มสมาชิกของไบท์แต่ละตัวในเซตจำกัดของ block state ที่มีขนาดเพิ่มขึ้นจากเดิม 4x4 เป็น 8x8 ทำให้จำนวนของสมาชิกในเซตจำกัดจากเดิม 16 เพิ่มเป็น 64 ตัว ซึ่งทำให้การโจมตีแบบแยกโครงสร้างย้อนกลับ (Reverse Engineer) นั้นทำได้ยากในการจำลองวิธีที่เป็นไปได้ของการเรียงสับเปลี่ยนสมาชิกทั้งหมดของเซตจำกัด

2) การเพิ่มความยาวของกุญแจเป็น 512 บิต และ 1024 บิต

ในด้านของ Performance จะเห็นว่าความยาวของกุญแจทำให้มีจำนวนรอบที่ใช้ในการเข้ารหัสเพิ่มขึ้น ส่งผลต่อกระบวนการขยายขนาดกุญแจและกระบวนการบวกค่ากุญแจ ยิ่งกุญแจยาวจะใช้เวลาในการทำงานที่นานขึ้น จึงทำ Performance ลดลง

ในด้านของ Security นั้น อัลกอริทึม AES เดิม กำหนดให้ใช้ขนาดกุญแจที่ 128, 192 และ 256 บิต ในการศึกษาครั้งนี้ได้เพิ่มความยาวของกุญแจที่ใช้เข้ารหัสถึง 512 และ 1024 บิต ซึ่งทำให้การสุ่มคาดเดารหัสผ่านมีความยากขึ้น และทนทานต่อการโจมตีแบบ Brute force attack จึงช่วยให้ข้อมูลมีความปลอดภัยมากยิ่งขึ้น

3) ขนาดของข้อมูลที่ต่างกัน ทำให้ Speedup มีค่าแตกต่างกัน

สาเหตุหลักเกิดจาก Cache size ของสมาร์ทโฟน กล่าวคือ ข้อมูลที่รอการประมวลผลเข้ารหัสจะถูกเก็บไว้ในหน่วยความจำความเร็วสูง (Cache) ซึ่งเมื่อมีขนาดของข้อมูลจำนวนไม่มาก ข้อมูลทั้งหมดจะสามารถเก็บไว้ใน Cache ได้ทั้งหมดและถูกเรียกใช้ได้ทันที ทำให้ Speed up มีค่าเพิ่มขึ้นตามขนาดของข้อมูล แต่เมื่อข้อมูลที่ใช้มีขนาดใหญ่ขึ้นจนไม่สามารถเก็บใน Cache ได้ทั้งหมด

การเรียกใช้ข้อมูลจึงเรียกได้เท่ากับขนาดของหน่วยความจำ Cache ดังนั้นจึงทำให้ Speed up เริ่มมีค่าคงที่ จะเห็นได้จาก iPhone5C ซึ่งมีหน่วยความจำ Cache น้อย จะมี Speed up สูงสุดที่ข้อมูลขนาด 256 KB ในขณะที่ iPhone7 ซึ่งมีหน่วยความจำ Cache มากกว่า ทำให้มี Speed up สูงสุดที่ข้อมูลขนาด 1024 KB

4) การใช้ทรัพยากรหน่วยความจำ (Memory Use)

การศึกษาการใช้หน่วยความจำของอัลกอริทึม พบว่า เมื่อทำการเข้ารหัสข้อมูลที่ขนาดเท่ากัน อัลกอริทึมทั้ง 2 ชนิดมีการใช้หน่วยความจำไม่แตกต่างกัน แต่เมื่อพิจารณาในแง่ของ อุปกรณ์ที่ใช้เข้ารหัส จะพบว่า iPhone7 ที่ใช้หน่วยประมวลผลแบบ 64 บิต มีการใช้หน่วยความจำมากกว่า iPhone5C ที่ใช้หน่วยประมวลผลแบบ 32 บิต เนื่องด้วย การอ้างอิงหน่วยความจำของ หน่วยประมวลผลแบบ 64 บิต มีค่าสูงกว่า 32 บิต

5) การใช้ทรัพยากรแบตเตอรี่ (Battery Use)

จากผลการศึกษาพบว่าการใช้ทรัพยากรแบตเตอรี่ของอัลกอริทึมบล๊อกขนาด 128 บิต และ 512 บิต ไม่มีความแตกต่างกัน เนื่องจากการทำงานของอัลกอริทึมทั้ง 2 แบบมีการใช้ CPU ที่ 100% จนเสร็จสิ้นกระบวนการ ดังนั้นอัตราการใช้แบตเตอรี่จึงใกล้เคียงกัน โดยอัตราการใช้พลังงานจะอยู่ที่ 11 – 12 นาที ต่อ 10% ของแบตเตอรี่ แต่บล๊อกขนาด 512 บิต จะมีปริมาณงานที่ทำได้ต่อหนึ่งหน่วยเวลาสูงกว่าบล๊อก 128 บิต ซึ่งความร้อนที่เกิดกับอุปกรณ์ iPhone ส่งผลโดยตรงต่อพลังงานแบตเตอรี่ที่สูญเสียเร็วขึ้น โดยการทดลองหนึ่งครั้งจะหยุดพักก่อนทำการทดลองใหม่ เพื่อให้ได้ค่าอัตราการใช้พลังงานแบตเตอรี่ที่สมเหตุสมผลที่สุด

ก่อนหน้านี้ได้มีการนำอัลกอริทึม AES ไปประยุกต์ใช้งานบน Hardware หลายชนิด ซึ่งมีกระบวนการ implement แตกต่างกันไปตามคุณลักษณะของ Hardware นั้น ๆ ผลที่ได้จากการศึกษาในครั้งนี้ โดยการขยายขนาดบล๊อค และเพิ่มขนาดกุญแจ บนอุปกรณ์ Apple iPhone ผู้วิจัยคาดว่าน่าจะมีประโยชน์ในการนำกระบวนการนี้ไปประยุกต์ใช้เข้ารหัสบนอุปกรณ์ Smartphone ซึ่งสามารถทำได้ง่าย ทุกที่ ทุกเวลา อีกทั้งยังเป็นการช่วยต่อยอดองค์ความรู้ทางด้าน Cryptography ต่อไปในอนาคต

5.2 ข้อเสนอแนะ

เครื่องมือที่ผู้จัดวิจัยสร้าง (Implement) มีข้อจำกัดด้านความเร็วในขั้นตอนคำนวณอยู่ (encoding) เพราะผู้วิจัยใช้วิธีการเขียนโค้ดแบบเมตริกซ์ (Matrix) ดังนั้นในกรณีนำรหัสเทียม (Pseudo code) ของอัลกอริทึมเพิ่มขนาดบล็อก AES นี้ไปใช้เชิงพาณิชย์ ควรพัฒนาการเขียนโค้ดในระดับบิต (bit level encoding) ให้สอดคล้องกับสถาปัตยกรรม ฮาร์ดแวร์นั้น ๆ เพื่อดึงประสิทธิภาพการประมวลผลให้ได้มากที่สุด

รายการอ้างอิง

[1] J. Daemen and V. Rijmen, *The design of Rijndael: AES – The advanced encryption standard*. New York: Springer-Verlag, 2002.

[2] T. Jamil, "The Rijndael algorithm," *IEEE Potentials,* vol. 23, pp. 36-38, 2004.

[3] ลัญฉกร วุฒิสิทธิกุลกิจ, ธงชัย โรจน์กังสดาล, วรากร ศรีเชวงทรัพย์, นพดล พรมภักษร, and สุวิทย์ นาคพีระยุทธ, วิทยาการรหัสลับเบื้องต้น. กรุงเทพมหานคร: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2548.

[4] National Institute of Standards and Technology, "Data Encryption Standard (DES), Federal Information Processing Standards Publication (FIPS PUB) 46-3," 1999.

[5] National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," 2015.

[6] National Institute of Standards and Technology, *Security Requirements for Cryptographic Modules*. Washington: U.S. Government Printing Office, 2001.

[7] D. Bergland, *Libertarianism in one lesson*, 5 ed. California: Orpheus Publications, 1990.

[8] A. Finlay, *Global Information Society Watch 2014 Communications surveillance in the digital age*. Istanbul: APC and Hivos, 2014.

[9] Shadow Open Mar Commitee, *Policy Statement and Position Papers*. New York: Bradley Policy Research Center, 2005.

[10] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish encryption algorithm: a 128-bit block cipher*: John Wiley \& Sons, Inc., 1999.

[11] ศุภชัย ทองสุข, "การทำรหัสลับแบบ AES บนหน่วยประมวลผลหลายแกนเพื่อเพิ่มประสิทธิภาพ," วิทยานิพนธ์ปริญญาวิทยาศาสตร์มหาบัณฑิต, คณะวิศวกรรมศาสตร์, จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร, 2556.

[12] A. Barnes, R. Fernando, K. Mettananda, and R. Ragel, "Improving the throughput of the AES algorithm with multicore processors," in *2012 IEEE 7th*

International Conference on Industrial and Information Systems (ICIIS), 2012, pp. 1-6.

[13] A. Moh'd, Y. Jararweh, and L. Tawalbeh, "AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation," in *2011 7th International Conference on Information Assurance and Security (IAS)*, 2011, pp. 292-297.

[14] S. Spiekermann and L. F. Cranor, "Engineering Privacy," *IEEE Trans. Softw. Eng.,* vol. 35, pp. 67-82, 2009.

[15] The Guardian. (2013, 2013 Jun 6). *Codenamed PRISM : NSA collecting phone records of millions of Verizon customers daily*. Available: http://www.bbc.com/news/technology-29076899

[16] The Guardian. (2013, 2013 Jul 31). *XKeyscore: NSA tool collects 'nearly everything a user does on the internet'* Available: http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data

[17] The Guardian. (2013, 2013 Sep 5). *Project Bullrun – classification guide to the NSA's decryption program* Available: http://www.theguardian.com/world/interactive/2013/sep/05/nsa-project-bullrun-classification-guide

[18] The New York Times. (2013, 2013 Sep 10). *Government Announces Steps to Restore Confidence on Encryption Standards* Available: http://bits.blogs.nytimes.com/2013/09/10/government-announces-steps-to-restore-confidence-on-encryption-standards/?src=twrhp&_r=2

[19] The Guardian. (2013, 2013 Sep 30). *Codenamed Marina : NSA stores metadata of millions of web users for up to a year, secret files show* Available: http://www.theguardian.com/world/2013/sep/30/nsa-americans-metadata-year-documents

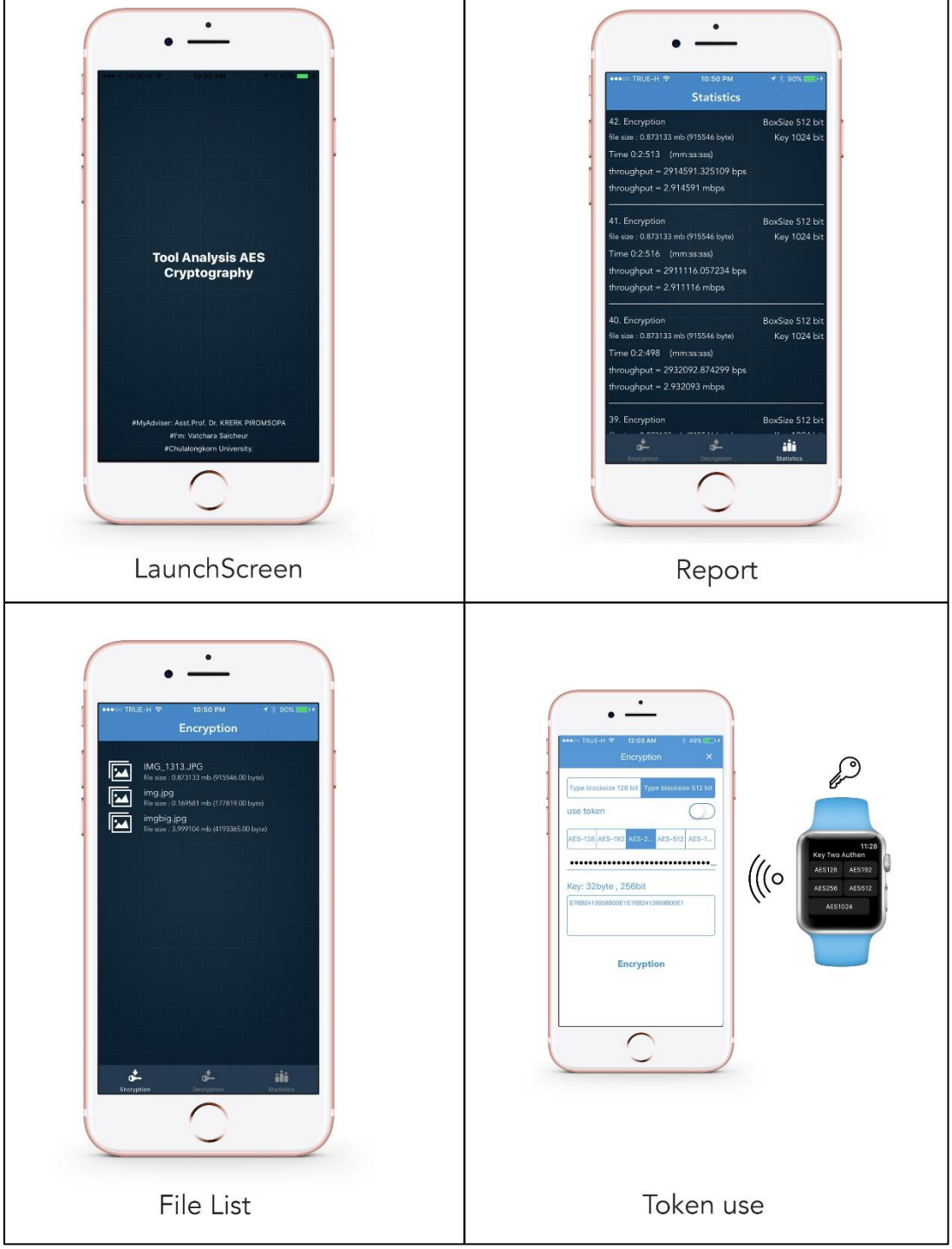
[20] The Guardian. (2013, 2013 Oct 4). *Attacking Tor: how the NSA targets users' online anonymity* Available: http://www.theguardian.com/world/2013/oct/04/tor-attacks-nsa-users-online-anonymity

[21] LeakSource. (2013, 2013 Dec 30). *NSA's ANT Division Catalog of Exploits for Nearly Every Major Software / Hardware / Firmware* Available: https://leaksource.wordpress.com/2013/12/30/nsas-ant-division-catalog-of-exploits-for-nearly-every-major-software-hardware-firmware/

[22] LeakSource. (2014, 2014 Apr 10). *Shotgiant: Snowden Docs Show Real Worry With Huawei Tech Is NSA Backdoors Not China Spying*. Available: http://leaksource.info/2014/03/23/shotgiant-snowden-docs-show-real-worry-with-huawei-tech-is-nsa-backdoors-not-china-spying/

ภาคผนวก

ภาคผนวก ก Interface ของ AES analysis tool บน iPhone



ภาพที่ 18 Interface ของ AES analysis tool บน iPhone

ภาคผนวก ข รายชื่อโครงการของสำนักงานความมั่นคงแห่งชาติ สหรัฐอเมริกา ที่มีผลต่อความมั่นคงของข้อมูล

ตารางที่ 12 แสดงรายชื่อโครงการของสำนักงานความมั่นคงแห่งชาติ สหรัฐอเมริกา ที่มีผลต่อความมั่นคงของข้อมูล

ช่วงเวลา (Timeline)	ชื่อโครงการ (Code-Name)	รายละเอียดโครงการ (Detail)
มิถุนายน พ.ศ. 2556 [15] (June 2013)	โปรเจคปริซึม (PRISM)	สำนักงานความมั่นคงแห่งชาติสหรัฐอเมริกา (NSA : National Security Agency) สามารถกระทำการเข้าถึง - เนื้อหาของอีเมล์, - โปรแกรมสนทนาโดยใช้ภาพและเสียง - ข้อมูลที่ถูกเก็บไว้ในคลาวด์สตอเรจ (เช่น OneDrive, Dropbox) - ข้อมูลเครือข่ายสังคมออนไลน์และยังร่วมถึง "คำขอพิเศษ" (Special request)
กรกฎาคม พ.ศ. 2556 [16] (July 2013)	เอ็กซ์คีย์สกอร์ (XKeyscore)	หน่วยงาน NSA สามารถกระทำการค้นหากิจกรรมและสะกดรอยเป้าหมาย โดย - สามารถค้นหาจากชื่อล็อกอิน - เบอร์โทรศัพท์, อีเมล - ภาษาที่ใช้งาน
กันยายน พ.ศ. 2556 [17] (September 2013)	บูลรัน (Bullrun)	หน่วยงาน NSA สามารถกระทำการถอดรหัสการเชื่อมต่อ โปรโตคอล TLS/SSL, HTTPS, SSH, VPNs, VoIP, webmail
กันยายน พ.ศ. 2556 [18] (September 2013)	ช่องโหว่ Dual_EC_DRBG	หน่วยงาน NSA เข้าไปมีส่วนร่วมกับหน่วยงาน NIST ทำการเปลี่ยนแปลงคุณสมบัติบางส่วนเพื่อสร้างช่องโหว่ที่ใช้สร้างเลขสุ่ม ซึ่งโดยปกติจะรองรับโดยหน่วยงาน NIST
กันยายน พ.ศ. 2556 [19] (September 2013)	มาริน่า (Marina)	หน่วยงาน NSA สามารถกระทำการจัดเก็บและสืบค้นmetadata ของข้อมูลรายบุคคล เช่น - รายการเข้าดูเว็บไซต์ - รายการค้นหาสถานที่

ช่วงเวลา (Timeline)	ชื่อโครงการ (Code-Name)	รายละเอียดโครงการ (Detail)
		- พฤติกรรมการใช้อีเมล์ - หรือรหัสผ่านอีเมล์ในบางกรณี
ตุลาคม พ.ศ. 2556 [20] (October 2013)	ฟอกซ์ซาซิส (FOXACID)	หน่วยงาน NSA สามารถกระทำการเข้าถึงข้อมูลแบบตั้งเป้าหมายรายบุคคลได้หากต้องการ
ธันวาคม พ.ศ. 2556 [21] (December 2013)	แอ๊นท์ดิวิชั่น (ANT division)	ฝ่ายพัฒนาฮาร์ดแวร์ และซอฟต์แวร์ของปฏิบัติการ TAO (Tailored Access Operations) แผนกสร้างจุดดักฟังให้กับโครงการอื่นๆ เช่น โครงการปริซึม
มีนาคม พ.ศ. 2557 [22] (March 2014)	ช็อทไจแอนท์(Shotgiant)	โครงการแปลง Router Huawei ให้กลายเป็นจุดดักฟังของ NSA

ภาคผนวก ค ผลงานตีพิมพ์

ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้จัดทำเป็นผลงานวิจัยฉบับเต็มจำนวน 2 เรื่อง ได้แก่

1. หัวข้อ A Cost analysis of AES- 128and AES- 512on Apple mobile processors เพื่อนำเสนอในการประชุมวิชาการ 2017 6th International Conference on Software and Computing Technologies (ICSCT 2017)
2. หัวข้อ An implementation of AES-128 and AES-512 on Apple mobile processor เพื่อนำเสนอในการประชุมวิชาการ ECTI-CON 2017

A Cost analysis of AES-128 and AES-512 on Apple mobile processors

Vatchara Saicheur
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: vatchara.s@student.chula.ac.th

Krerk Piromsopa
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: krerk.p@chula.ac.th

***Abstract*—This paper is the cost analysis of two Advanced Encryption Standard (AES) algorithms on 32-bit and 64-bit Apple mobile processor by using iPhone5C and iPhone7. Our analysis shows increasing in performance when expanding the block size from 128 bits (AES-128) to 512 bits (AES-512). Similarly increasing the length of encryption key to 512 bits and 1024 bits yields stronger security. Our aim is to analyze the encryption cost different between the original AES-128 and the AES-512. The results showed that increasing block size will give 1.21 – 1.64 speed up on iPhone5C and 1.19 – 1.55 speed up on iPhone7 depending on the key length. Moreover, AES 512-bit block size shows faster CPU time in encryption than 128-bit block size. However, the memory usage for encryption on all key size are similar. iPhone7 used more memory than iPhone5C. In conclusion, expanding block size to 512 bits can increase performance while this is also lower the cost on mobile device. This result may have the benefit in improving the security of personal data by using mobile phone.**

Keywords-AES; cryptography; cost analysis; mobile devices

I. INTRODUCTION

Today, the Internet and technology has made our life more comfortable. Many devices are designed to help facilitate comfortable in everyday life such as smartphone, tablet, smart watch and computer. Those devices have high specification and can be used for many things. When people are using mobile devices for communication or data transmission through the internet network, it is also increase the risk of data leaked or data stolen by third party. Particularly, if important data was stolen, it may cause serious damage. A tool for protecting our data is encryption.

There are many ways to encrypt the data. The popular standard encryption is AES algorithm. The AES algorithm is a standard encryption algorithm that has been certified by the US government for encrypting and storing personal information. It is a symmetric key algorithm which has been accepted and presented by NIST since 2001[1]. However, implementing AES algorithm on mobile devices can be difficult and resource intensive. If we can improve the performance and can lower the resource utilization at the same time, it will allow applications to encrypt data on mobile device instead of off-loading to another computer.

This paper proposed to expand the AES block size to 512 bits for using in mobile devices. To promote this concept, we compare the cost between AES 128 bits and the new block size by varying different key sizes: 128, 192, 256, 512 and 1024 bits. To make our analysis more complete, the experiment is based on both 32-bit and 64-bit mobile architecture (Apple ARM A6 and Apple ARM A10 respectively).

II. RELATED WORK

Cryptography is the study of techniques for securing information in the presence of third parties such as microdots, steganography (embedding bits in an image), and other ways to hide data. Effective cryptography is associated with the development and the creation of mathematical algorithms[2].

A. AES-128 algorithm

AES is a symmetric-key encryption using the array of 4x4 of 128-bit ciphering block. It uses the key size of 128, 192, and 256 bits with 10, 12, or 14 iteration rounds, respectively. There are four major operations; byte substitution, row shifting, column mixing and adding round key[3],[4]. It is widely adopted and supported by many applications. It considered secure compared to early algorithms such as Data Encryption Standard (DES) which is certified by Federal Information Processing Standard (FIPS)[5].

B. AES-512 implementation

Several AES implementations were present in many studies. The difference applications of AES require different implementations of same algorithm. The FPGA architecture for a new version of the Advanced Encryption Standard (AES) algorithm[6] proposed the implementation of AES using input block size and key size of 512 bits developed in VHDL, and synthesized using Virtix-6 and Virtex-7 chips. The study found that this design obtained high level of security to cryptanalysis and yields the speed up of 2.3 comparing to the original AES-128 implementation[7].

III. PROPOSED METHODOLOGY

This study contains 2 steps. First is the implementation of AES for using on mobile devices with the expansion of block size to AES-512. Second is the comparison of efficacy and the cost of using algorithm by comparing speed up, CPU time and memory utilization on 32-bis and 64-bit Apple mobile processors. The specification of the processor can be described in Table 1.

TABLE I. SPECIFICATION OF IPHONE5C AND IPHONE7

Model	iPhone5C	iPhone7
CPU	ARM A6 32-bit architecture Dual-core processor 1.3 GHz	ARM A10 64-bit architecture Quad-core processor 2.34 GHz
Ram	1 GB	2 GB
iOS	10.2	10.2

The implementation of AES-512 algorithm can be explained as follow:

A. *Block size expansion*

The block state of the original AES-128 is an 4x4 array. Each row and column of input with Nb byte requires Nb = 4. In this paper, we increase the block state to 8x8 array. This also increase the input data in required Nb = 8 as shown in figure 1.

B. *Byte Substitution*

The plaintext Input are assigned in the array 8x8 of 64 bytes. Each byte is independently substituted by the values by the substitution boxes (similar to that of the AES-128) to increase the complexity and the security of algorithm as shown in figure 2.

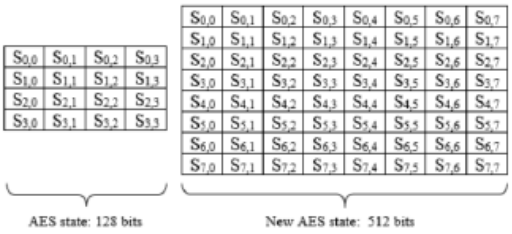


Figure 1. The block state of original AES-128 and new AES-512 bits.

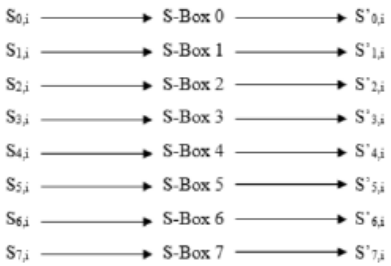


Figure 2. The byte substitution of AES-512 bits.

C. *Row shifting*

After the byte substitution process, the resulting matrix rows are shifted with simple process. The bytes in each row in the matrix will be rotated to the left. The number of left rotations will be increased by one per row. In this case, the zeroth row is not shifted; the first row is shifted by one byte; the second row is shifted by two bytes. The increasing is repeated until the last row as shown in figure 3.

Figure 3. The row shifting of AES-512 bits.

D. *Column mixing*

After the shifting row process, the result is the bit changed in the state block. Next process is converting data in columns by multiplying with a pre-defined polynomial. Each column contains 8 bytes of data which must be applied to a column of data matrix. Column mixing are based on the concept of polynomial over GF(2^n). The columns in the data matrix (shown in figure 4) will be multiplied by a fixed polynomial of a(x) as given in (1).

The multiplication result is modulo by p(x)= x8 + 1 to maintain the resulting polynomial with the degree of less than 8. The inversing of column mixing as shown in figure 5 will be multiplied with the inverse of the polynomial a'(x) as given in (2).

$$a(x) = [02]x^7 + [01]x^6 + [03]x^5 + [01]x^4 + [01]x^3 + [01]x^2 + [01]x^1 + [01]x^0 \quad (1)$$

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \\ S'_{4,c} \\ S'_{5,c} \\ S'_{6,c} \\ S'_{7,c} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 03 & 01 & 01 & 01 & 01 & 01 \\ 01 & 03 & 01 & 01 & 01 & 01 & 01 & 02 \\ 03 & 01 & 01 & 01 & 01 & 01 & 02 & 01 \\ 01 & 01 & 01 & 01 & 01 & 02 & 01 & 03 \\ 01 & 01 & 01 & 01 & 02 & 01 & 03 & 01 \\ 01 & 01 & 01 & 02 & 01 & 03 & 01 & 01 \\ 01 & 01 & 02 & 01 & 03 & 01 & 01 & 01 \\ 01 & 02 & 01 & 03 & 01 & 01 & 01 & 01 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \\ S_{4,c} \\ S_{5,c} \\ S_{6,c} \\ S_{7,c} \end{bmatrix}$$

Figure 4. The data matrix for column mixing encryption of AES-512 bits

$$a'(x) = [0E]x^7 + [01]x^6 + [09]x^5 + [01]x^4 + [0D]x^3 + [01]x^2 + [0B]x^1 + [01]x^0 \quad (2)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \\ s'_{4,c} \\ s'_{5,c} \\ s'_{6,c} \\ s'_{7,c} \end{bmatrix} = \begin{bmatrix} 0e & 01 & 09 & 01 & 0d & 01 & 0b & 01 \\ 01 & 09 & 01 & 0d & 01 & 0b & 01 & 0e \\ 09 & 01 & 0d & 01 & 0b & 01 & 0e & 01 \\ 01 & 0d & 01 & 0b & 01 & 0e & 01 & 09 \\ 0d & 01 & 0b & 01 & 0e & 01 & 09 & 01 \\ 01 & 0b & 01 & 0e & 01 & 09 & 01 & 0d \\ 0b & 01 & 0e & 01 & 09 & 01 & 0d & 01 \\ 01 & 0e & 01 & 09 & 01 & 0d & 01 & 0b \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \\ s_{4,c} \\ s_{5,c} \\ s_{6,c} \\ s_{7,c} \end{bmatrix}$$

Figure 5. The data matrix for column mixing decryption of AES-512 bits.

E. *Add round key*

In this process, each byte of key will be added to data state in each round of encryption process. The setting key is required to expand before used. The expanded key will be XORed with each state column until the last column of each round as shown in (3). A round number of encryption is different for each key size as shown in table 2.

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, \ldots, S'_{7,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, \ldots, S_{7,c}] \oplus [w_{round*Nb+c}] \quad (3)$$

F. *Key expansion*

The key expansion is the process of preparing key for each round of encryption and decryption process. The key will be kept in words of eight bytes each. Words is defined by the number of column used for each key size. It is often written in the form of number of key *(Nk)*. The algorithm will generate sub-keys *Nb(Nr+1)* words, they depend on the number of block *(Nb)* and the number of round *(Nr)*. This process requires *SubWord()* and *RotWord()* functions. *SubWord()* receives input words of 8 bytes for substituted by the values in the substitution boxes. *RotWord()* receives input word $[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$, to be rotated left to $[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_0]$. Moreover, the round constant *(Rcon[i])* should be used as defined by (4):

Let: i be the round number

$$Rcon[i] = 00000010^{(i-8)/8} \quad (4)$$

The round key expansion is performed according to the following equations:

If i is a multiple of Nk

$$w[i] = (SubWord(RotWord(w[i-1])) \oplus Rcon[i/Nk]) \oplus w[i-Nk] \quad (5)$$

If i is not a multiple of Nk

$$w[i] = w[i-1] \oplus w[i-Nk] \quad (6)$$

TABLE II. ROUND NUMBER OF AES-128 BITS AND AES-512 BITS

Key size	Round Number AES-128 bits	Round Number AES-512 bits
128 bits	10	4
192 bits	12	5
256 bits	14	6
512 bits	22	10
1024 bits	38	18

G. *Decryption process*

The encrypted cipher should be inverted by starting with *InvertShiftRows(), InvertSubBytes(), AddRoundKey()* and *InvertMixColumns()* respectively.

IV. EXPERIMENTAL RESULTS AND EVALUATION

In this study, we have implemented AES-128 and AES-512 on mobile phone by using iPhone5C and iPhone7. The results from 2-type of AES show that: AES-512 algorithms have better speed than AES-128 algorithms. The speed up depends on devices. Our experimental results showed that iPhone5C has the maximum speed up at the encrypted data size of 256 kb. However, iPhone7 has the maximum speed up at encrypted data size 1024 KB as shown in figure 6.

The AES-512 algorithms on iPhone5C yield about 1.21 – 1.64 speed up comparing to those of AES-128 algorithm. The AES-512 algorithms on iPhone7, like iPhone5c, yield about 1.19 – 1.55 speed up as shown in figure 7.

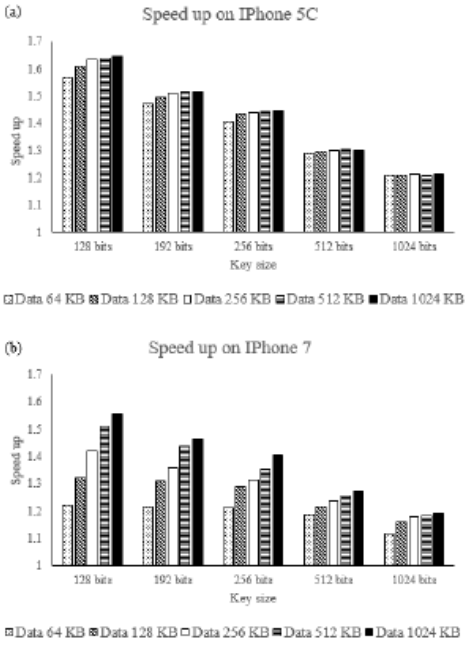


Figure 6. Speed up testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C (a) and iPhone7 (b) at encrypted data size 64, 128, 256, 512 and 1024 KB

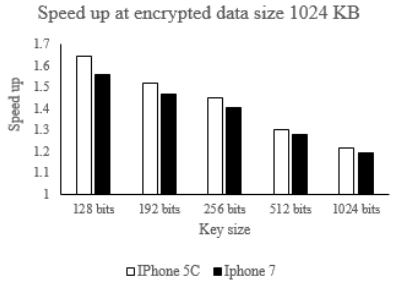


Figure 1. Speed up testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C and iPhone7 at encrypted data size 1024 KB

TABLE I. SPEED UP BETWEEN AES-512 BITS/AES-128 BITS

Key size	Speed up	
	iPhone5C	iPhone7
128 bits	1.64	1.55
192 bits	1.51	1.46
256 bits	1.45	1.40
512 bits	1.30	1.27
1024 bits	1.21	1.19

The cost analysis of CPU usage found that CPU will run at full speed when algorithms are being processed. The AES-512 algorithms use less CPU times than that of the AES-128 about 2,008 – 3,584 milliseconds on iPhone5C. Increasing the key size will result in higher different in CPU time. iPhone7 showed smaller different in CPU time. The AES-512 use less CPU time than that of the AES-128 about 330 – 550 milliseconds. The result is shown in figure 8.

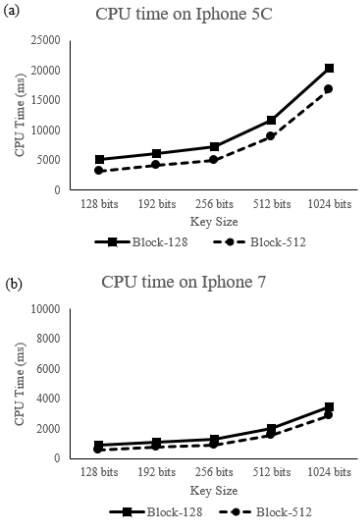


Figure 2. CPU time testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C (a) and iPhone7 (b) at encrypted data size 1024 KB

TABLE II. CPU TIMES OF AES-128 BITS AND AES-512 BITS

Devices	Key size (bits)	CPU Time (ms)		
		AES-128	AES-512	Different time
iPhone5C	**Key 128**	5134	3126	2008
	Key 192	6207	4099	2108
	Key 256	7328	5060	2268
	Key 512	11664	8959	2705
	Key 1024	20356	16772	3584
iPhone7	**Key 128**	914	588	326
	Key 192	1106	756	350
	Key 256	1287	916	371
	Key 512	1993	1563	430
	Key 1024	3440	2887	553

The analysis of memory usage shows that at the same size of encrypted data, memory usage of both algorithms is similar. However, the iPhone7 uses memory more than that of iPhone5C as shown in figure 9.

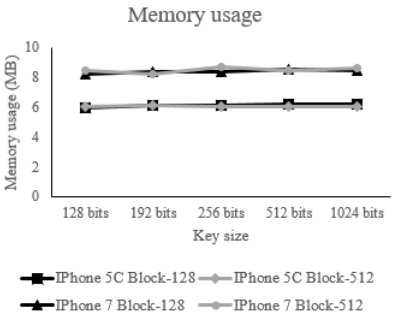


Figure 3. Memory usage testing of key size 128, 192, 256, 512 and 1024 bits compare between AES-512/AES-128 algorithms on iPhone5C and iPhone7 at encrypted data size 1024 KB

TABLE III. MEMORY USAGE OF AES-128 BITS AND AES-512 BITS

Devices	Key size (bits)	Memory usage (MB)	
		AES-128	AES-512
iPhone5C	**Key 128**	5.97	6.07
	Key 192	6.13	6.10
	Key 256	6.13	6.06
	Key 512	6.20	6.00
	Key 1024	6.20	6.06
iPhone7	**Key 128**	8.17	8.46
	Key 192	8.40	8.17
	Key 256	8.33	8.67
	Key 512	8.53	8.46
	Key 1024	8.46	8.60

I. CONCLUSION

In conclusion, we have proposed the use of AES-512 algorithm supported on mobile devices. Our study showed the cost analysis between the original AES-128 and the new AES-512 algorithms with various key size. The new algorithm showed higher speed than the original AES at lower cost. Although, encryption with longer length of key

showed less difference than that of the shorter key, the new algorithm provides stronger security. Thus it is more useful for data protection.

When comparing between 32-bit and 64-bit CPU, (iPhone5C and iPhone7), we found that both devices give the same trend for both algorithms.

With stronger level of security and better performance, we expect that the use of AES-512 algorithm would be useful for people and organizations in protecting their important data.

REFERENCES

[1] Federal Information Processing Standards (FIPS), "Announcing the Advanced Encryption Standard (AES)," *National Institute of Standards and Technology (NIST)*, 2001.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[3] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Science & Business Media, 2002.

[4] T. Jamil, "The Rijndael algorithm," *IEEE Potentials*, vol. 23, no. 2, pp. 36–38, Apr. 2004.

[5] Federal Information Processing Standards (FIPS), "Data Encryption Standard (DES)," *National Institute of Standards and Technology (NIST)*, 1999.

[6] V. Arun, K. Vanisree, and D. L. Reddy, "Implementation of AES-GCM encryption algorithm for high performance and low power architecture Using FPGA," *Proc. ICETET*, vol. 29, p. 30th, 2014.

[7] A. Moh'd, Y. Jararweh, and L. 'ai Tawalbeh, "AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation," in *Information Assurance and Security (IAS), 2011 7th International Conference on*, 2011, pp. 292–297.

An implementation of AES-128 and AES-512 on Apple mobile processor

Vatchara Saicheur
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: vatchara.s@student.chula.ac.th

Krerk Piromsopa
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
Email: krerk.p@chula.ac.th

***Abstract*—This paper proposes the implementation of the Advance Encryption Standard (AES) algorithm on Apple iPhone7. We extend the standard AES-128 algorithm to support the block size of 512 bits (AES-512). There are 4 steps in the encryption process: SubBytes, ShiftRows, MixColumns and Add-round key. The comparison between original AES-128 and the new AES-512 using 128, 192, 256, 512, 1024 bits key size is presented. Our implementation shows that AES-512 has higher performance than that of AES-128. The speed up is 1.20 – 1.58 depending on key size. The 128-bit key size is the fastest. The 1024-bit key size is the slowest. We conclude that expanding the block size to 512 bits can enhance the throughput and the speed up of AES algorithm.**

Keywords- AES; cryptography; iPhone; mobile devices

I. INTRODUCTION

Nowadays, there are many news regarding the violation of personal information, data privacy, data robbery or eavesdropping. One way to protect our data is using encryption. There exist many encryption methods. One of the most popular methods is AES, which is the standard encryption algorithm certified by NIST[1]. However, the current version of AES encryption has been implemented since 2001. With advances in computing, the security level provided by the standard AES may be not be strong enough. Moreover, mobile devices such as iPhone, Android, has been developed for various applications. They are few (if any) differences between the mobile devices and desktop computers. The use of mobile devices on internet for storing, transferring, or opening data may increase the risk of data leakage if not implemented properly.

In the past, the use of encryption on mobile devices may not be not efficient due to limited resources. This paper aims to show that it is possible to efficiently implement the AES-512 encryption on mobile devices. We use iPhone7 (the latest mobile device on iOS platform) as our test device. We also want to present that increasing the block size from 128 bits to 512 bits can increase the throughput. In addition, we have increased the length of the key to 1024 bits to support more secure applications.

A symmetric key block encryption is to encrypt each block state with a secret key. The input of each block state is 32 bytes for 4X4 block or 64 bytes for 8x8 block. For example, using 64 bytes of input plain text means this plain text has 512 bits. If encrypted with 128 bits block size (4x4 block), this plain text will be divided into 4 sets. Each set will be added to each block state and be encrypted with a secret key. The decryption process is the reverse of all steps.

II. RELATED WORK

A. *Rijndael algorithm [2],[3]*

In 1998, Joan Daemen and Vincent Rijimen presented the symmetric key encryption algorithm, namely Rijndael. Later in 2001, this algorithm has been selected by NIST as standard encryption algorithm to replace the aged triple Data Encryption Standard (3DES). The name of the algorithm was then changed to Advanced Encryption Standard or AES. Comparing to DES, the AES is faster and is more practical. Since then, the algorithm has been used widely today

The process of AES algorithm is divided into four main steps. SubBytes is the process for replacing the byte by using the substitution boxes. ShiftRows is the process for moving bytes in a row of the array state with offset differ in each row. MixColumns is the process for combining data within each column of the array State. Add-round key is the process for XORing the key into each state.

B. *The 512-Bit Advanced Encryption Standard Algorithm Design and Evaluation [4],[5]*

Abidalrahman Moh'd et.al, studied the optimization of AES encryption algorithms. They used the new AES 512-bit block size with 512 bits key running on VDRL. The aim was to increase the security. When comparing the new AES-512 with the legacy AES-128 on same hardware, the larger block size increases the throughput. Thus, we choose the 512-bit block size for our implementation on mobile devices.

III. PROPOSED METHODOLOGY

There are 7 steps in our implementation of AES on mobile devices.

A. *Block-size expansion*

The Block-size expansion is the process for increasing the amount of work per unit of time. The original block size of AES is an array of 4 x 4 with 128 bit each. Each row consists of *Nb* byte with $Nb = 4$. To increase the throughput, we therefore have to increase the row and the column of each state as shown in Table I.

TABLE I. COMPARISON OF BLOCK SIZE BETWEEN AES-128 AND AES-512

AES	AES Blocksize 128 bit	AES Blocksize 512 bit
Nb	4	8
Row	4	8
Column	4	8
Byte	16	64

From Table I, new AES-512 algorithm can be written in 2D array as shown in Figure 1.

The block state of the new AES-512 algorithm can be created as a function AESBlock512_Encryption(). The pseudo code of this function is provided as follow:

```
( byte output[8*Nb] )
AESBlock512_Encryption ( byte input[8*Nb] ,Word w[Nb*(Nr+1)] )
BEGIN
        byte state[8,Nb]
        State = input
        AddRoundKey(state, w[0,Nb-1])
    FOR(round = 1 step 1 to Nr-1 )
                SubBytes(state)
                ShiftRow(state)
                Mixcolumns(state)
                AddRoundKey(state, w[round*Nb,
                                (round+1)*Nb-1] )
        END FOR LOOP
                SubBytes(state)
                ShiftRow(state)
                AddRoundKey(state, w[round*Nb,
                                (round+1)*Nb-1] )
    Return output = state;
END
```

The functions SubBytes(), ShiftRow(), Mixcolumns(), AddRoundKey() will be explained in the next topics.

B. *Byte substitution*

After adding the plain text to the 512-bit block, each byte can independently be replaced with each other by the values by the substitution boxes as shown in Figure 2.

C. *Row shifting*

After the byte substitution process for changing the value in each block, the matrix rows can be shifted to the left. The zeroth row remains at the same location. The first row is shifted by one byte; the second row is shifted by two bytes. The increment is repeated until the last row as shown in Figure 3.

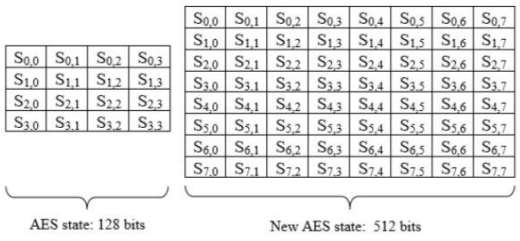


Figure 1. The comparison of the state of block size in AES-128 and AES-512

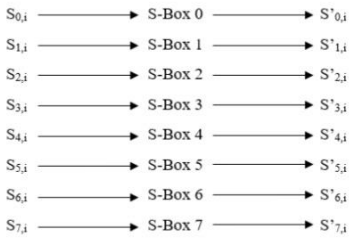


Figure 2. bytesubstitution by the values of the substitution boxes

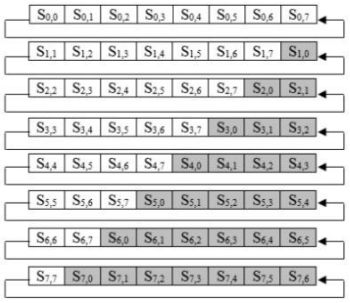


Figure 3. Row shifting process

D. *Column mixing*

After the row shifting process, the resulting matrix columns are mixed with the following process. Each column consists of 8 bytes that can be converted into a polynomial and the obtained data will be multiplied with a pre-defined polynomial. Column mixing are based on the concept of polynomial over GF(2n). The columns in the data matrix (shown in Figure 4) will be multiplied by a fixed polynomial of a(x) as given in (1)

$$a(x) = [02]x^7 + [01]x^6 + [03]x^5 + [01]x^4 + [01]x^3 + [01]x^2 + [01]x^1 + [01]x^0 \quad (1)$$

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \\ S'_{4,c} \\ S'_{5,c} \\ S'_{6,c} \\ S'_{7,c} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 03 & 01 & 01 & 01 & 01 & 01 \\ 01 & 03 & 01 & 01 & 01 & 01 & 01 & 02 \\ 03 & 01 & 01 & 01 & 01 & 01 & 02 & 01 \\ 01 & 01 & 01 & 01 & 01 & 02 & 01 & 03 \\ 01 & 01 & 01 & 01 & 02 & 01 & 03 & 01 \\ 01 & 01 & 01 & 02 & 01 & 03 & 01 & 01 \\ 01 & 01 & 02 & 01 & 03 & 01 & 01 & 01 \\ 01 & 02 & 01 & 03 & 01 & 01 & 01 & 01 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \\ S_{4,c} \\ S_{5,c} \\ S_{6,c} \\ S_{7,c} \end{bmatrix}$$

Figure 4. Multiplying the data matrix with a(x) polynomial

E. *Add-Round Key*

In this process, each byte of key will be added to the data state in each round of encryption process. The number of round for each key size is shown in Table II. The set key must be expanded before used. The expanded key will be XORed with each state column until the last column of each round as shown in (2).

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, \ldots, S'_{7,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, \ldots, S_{7,c}] \oplus [W_{round*Nb+c}] \quad (2)$$

TABLE II. ROUND NUMBER OF OF AES-128 BITS AND AES-512 BITS

Key size	Round number of AES-128 bits	Round number of AES-512 bits
128bits	10	4
192 bits	12	5
256 bits	14	6
512 bits	22	10
1024 bits	38	18

F. Key Expansion

The key expansion is the process of preparing the key for each round of encryption and decryption process. The key will be kept in words of eight bytes each. Words are defined by the the number of columns used for each key size. It is often written in the form of number of key (*Nk*). The algorithm will generate sub-keys *Nb(Nr+1)* words, they depend on the number of block (*Nb*) and the number of round (*Nr*). Let *i* be the round number, the word with *i* less than *Nk* are not change. The other words have been divided into 2 group.

If *i* is a multiple of *Nk*

$$w[i] = (SubWord(RotWord(w[i-1])) \oplus Rcon[i/Nk]) \oplus w[i-Nk] \quad (3)$$

If *i* is not a multiple of *Nk*

$$w[i] = w[i-1] \oplus w[i-Nk] \quad (4)$$

This process requires SubWord() and RotWord() functions. SubWord() receives input words of 8 bytes for substituted by the values in the substitution boxes. RotWord() receives input word $[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7]$, to be rotated left to $[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_0]$. Moreover, the round constant (Rcon[i]) should be used as defined by (5).

$$Rcon[i] = 00000010^{(i-8)/8} \quad (5)$$

The pseudo can be shown as follows.

```
KeyExpansion ( byte key[8*Nk] ,word w[Nb*(Nr+1)] )
BEGIN
        word = temp
        i = 0
        WHILE ( i < Nk )
            w[i] = word(key[8*1] , key[8*i+1] , key[8*1+2] ,
                    key[8*1+3] , key[8*1+4], key[8*1+5] ,
                    key[8*1+6] , key[8*1+7]    )
            i = i+ 1
        END WHILE
        i = Nk
        WHILE ( i < Nb * (Nr+1))
            temp = w[i-1]
            IF (i mod Nk == 0)
                temp = SubWord(RotWord(temp)) XOR Rcon[i/NK]
            END IF
            w[i] = w[i-Nk] XOR temp;
             i  =  i + 1
    END WHILE
END
```

G. Decryption process

The encrypted cipher should be inverted by the following processes. InvShiftRows() is the inversion of ShiftRows() by shift byte to the right. InvSubBytes() is the inversion of SubBytes() by using inverse S-Box. InvMixColumns(), the resulting matrix column will be multiply with a'(x) polynomial which is given in (6)

$$a'(x) = [0E]x^7 + [01]x^6 + [09]x^5 + [01]x^4 + [0D]x^3 + [01]x^2 + [0B]x^1 + [01]x^0 \quad (6)$$

Finally, AddRoundKey() is similar to that of the encryption process.

Decryption process can be shown as follows.

```
( byte output[8*Nb] ) AESBlock512_Decryption ( byte input[8*Nb]
,Word w[Nb*(Nr+1)] )
BEGIN
        byte state[8,Nb]
        State = input
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
        FOR(round = Nr-1 step -1 downto 1 )
            InvShiftRows(state)
            InvSubBytes(state)
            AddRoundKey(state, w[round*Nb, (round+1)*Nb-1] )
            InvMixColumns(state)
        END FOR
            InvShiftRows(state)
            InvSubBytes(state)
            AddRoundKey(state,w[0,Nb-1])
        Return output = state;
END
```

IV. EVALUATION AND RESULTS

In this paper, the implementation of AES-512 algorithm on iPhone7. The specification of iPhone7 is shown in Table III. There are 2 parts in our implementation. The first part is the interface for using on mobile devices. The second part is the processing of the AES algorithm is used to encryption and decryption.

TABLE III. SPECIFICATION OF IPHONE7

Device Model	iPhone 7	iPhone 7
CPU	ARM A10 64-bit architecture Quad-core 2.34 GHz	ARM A10 64-bit architecture Quad-core 2.34 GHz
Ram	2 GB	2 GB
iOS	10.2	10.2

For the Interface part, users can select a block size and a key size for encryption. There is a log file of all work that collect data size, block size, key size, throughput, and time used as shown in Figure 5. This process is developed on Xcode native language (Objective C) to obtain the best performance from the iPhone 7 running iOS.

The algorithm part is the part of encryption and decryption process which is developed in C-programming language. It is developed a library for Xcode. When starting the encryption or the decryption, we will record a timestamp for later calculating throughput and speed up as shown in Figure 6.

Between AES-128 and AES-512, we found that the AES-512 algorithm has better speed than that of the AES-128 algorithm. The AES-512 algorithm has a speed up about 1.20 – 1.58 comparing to that of AES-128 algorithm at the data size of 4096 kb. The speed up is higher when for larger files. However,

speed up are decreasing when the key sizes are increasing. The results are shown in Table IV and Figure 7.

Figure 5. Interface of AES algorithm on iPhone

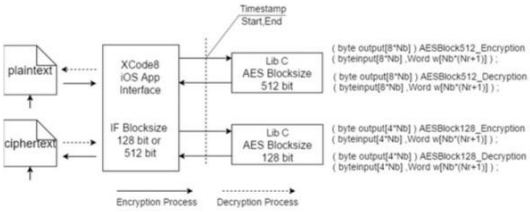


Figure 6. The process of encryption and decryption on iPhone

TABLE IV. SPEED UP BETWEEN AES-512 AND AES-128 AT KEY SIZE 128 1024 – BITS TEST WITH ENCRYPTED DATA SIZE 4096 – 64KB

Data size	Speed up				
	Key 128	Key 192	Key 256	Key 512	Key 1024
64 KB	1.22	1.22	1.21	1.19	1.12
128 KB	1.32	1.31	1.29	1.22	1.16
256 KB	1.42	1.36	1.32	1.24	1.18
512 KB	1.51	1.44	1.35	1.26	1.18
1024 KB	1.55	1.46	1.40	1.28	1.19
2048 KB	1.58	1.47	1.41	1.28	1.19
4096 KB	1.58	1.47	1.40	1.28	1.20

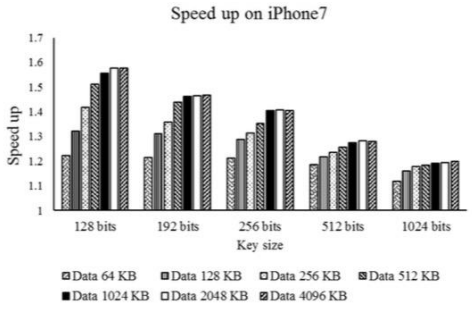


Figure 7. Speed up between AES-512 and AES-128 at key size 1024 – 128 bits test with encrypted data size 64 – 4096 KB

V. CONCLUSION

We have presented the implementation of AES algorithm on iPhone7. The algorithm is developed from AES-128 by expanded block size to 512 bits. We compare the performance between two algorithms. Our implementation showed that expanding the block can increase the performance. Moreover, the longer the key size makes it difficult to guess. This makes the data more secure. Though previous studies have shown the implementations of AES on many hardware with various implementation processes depending on hardware resources, we expect that our implementation will be useful for those who want to apply the AES algorithm on mobile devices. In particular, our implementation will serve as a basis for implementing cryptography in the future.

REFERENCES

[1] Federal Information Processing Standards (FIPS), "Announcing the Advanced Encryption Standard (AES)," National Institute of Standards and Technology (NIST), 2001.

[2] T. Jamil, "The Rijndael algorithm," IEEE Potentials, vol. 23, no. 2, pp. 36–38, Apr. 2004.

[3] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard. Springer Science & Business Media, 2002.

[4] V. Arun, K. Vanisree, and D. L. Reddy, "Implementation of AES-GCM encryption algorithm for high performance and low power architecture Using FPGA," Proc. ICETET, vol. 29, p. 30th, 2014.

[5] A. Moh'd, Y. Jararweh, and L. 'ai Tawalbeh, "AES-512: 512-bit Advanced Encryption Standard algorithm design and evaluation," in Information Assurance and Security (IAS), 2011 7th International Conference on, 2011, pp. 292–297.

ประวัติผู้เขียนวิทยานิพนธ์

ชื่อ – สกุล วัชร สายเชื้อ

วัน เดือน ปีเกิด 23 กรกฎาคม พ.ศ. 2528

จบการศึกษาระดับมัธยมศึกษาที่ โรงเรียนสวนกุหลาบวิทยาลัย รังสิต

จบการศึกษาระดับปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ เกียรตินิยมอันดับสอง มหาวิทยาลัยเทคโนโลยีมหานคร : โปรเจคจบ โทรจันควบคุมคอมพิวเตอร์ windows XP ในเครือข่าย

ในปี พ.ศ. 2556 ได้ศึกษาต่อระดับปริญญาโท สาขาวิศวกรรมซอฟต์แวร์ (SE) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย มีความสนใจด้านงานเข้ารหัสลับ (Cryptography)

ประสบการณ์การทำงาน

2551-2552 : Network Engineer CCTV

2552-2553 : Digital Signage Software Developer

2553-2554 : Game Online Developer

2554-2560 : Mobile Developer & Consult & Creative Designer

ความสนใจในอนาคต

- Artificial Intelligence Neural Networks
- Data compression
- ภาษาญี่ปุ่น

e-mail : turnofmut@gmail.com