การผสมความลับในการรักษาความปลอดภัยข้อมูลแบบไม่เท่าเทียมและการเข้ารหัสแบบไฮบริด

สรไกร ไกรปุย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา
พฤษภาคม 2560

คณะกรรมการควบคุมวิทยานิพนธ์ และคณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของ สรไกร ไกรปุย ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

.....อาจารย์ที่ปรึกษาหลัก
(ดร. อภิรัฐ ลิ่มมณี)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. วิมล แสนอุ้ม)

.....กรรมการ
(ดร. อภิรัฐ ลิ่มมณี)

.....กรรมการ
(รองศาสตราจารย์ วิรุฬห์ ศรีบริรักษ์)

.....กรรมการ
(รองศาสตราจารย์ ดร. กิตติพงศ์ บุญโล่ง)

คณะวิศวกรรมศาสตร์อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า ของมหาวิทยาลัยบูรพา

..... คณบดีคณะวิศวกรรมศาสตร์
(ดร. อาณัติ ดีพัฒนา)
วันที่ 30 เดือน พฤษภาคม พ.ศ. 2560

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยการได้รับความกรุณาให้คำปรึกษา เสนอแนะแนวทางที่ถูกต้องและตรวจแก้ไขข้อบกพร่องต่าง ๆ อย่างดียิ่งจาก ดร. อภิรัฐ ลิ่มมณี อาจารย์ที่ปรึกษาหลัก ที่ให้คำแนะนำชี้แนะแนวทาง และ รองศาสตราจารย์ วิรุฬห์ ศรีบริรักษ์ ที่สนับสนุนช่วยเหลือและเป็นกำลังใจ ประธานคณะกรรมการสอบวิทยานิพนธ์ คณะกรรมการสอบวิทยานิพนธ์ที่ให้คำแนะนำ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณทุกท่านเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณผู้ทรงคุณวุฒิทุกท่านที่ได้กรุณาตรวจสอบความสมบูรณ์และให้คำแนะนำแก้ไขเครื่องมือในการวิจัย รวมทั้งผู้เชี่ยวชาญทุกท่านที่กรุณาตอบแบบสอบถาม

ท้ายที่สุดผู้วิจัยขอกราบขอบพระคุณ บิดา มารดา พี่ น้อง และเพื่อน ๆ ที่ได้ให้ความช่วยเหลือและกำลังใจ ตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่มิได้กล่าวถึงในที่นี้

คุณค่าและประโยชน์ของวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบเป็นกตัญญูกตเวทิตาแด่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่าน ที่ได้อบรมสั่งสอน และให้กำลังใจแก่ผู้วิจัยเสมอมา

สรไกร ไกรปุย

55910297: สาขาวิชา: วิศวกรรมไฟฟ้า; วศ.ม. (วิศวกรรมไฟฟ้า)
คำสำคัญ: การเข้ารหัส/ การผสมความลับ/ การเข้ารหัสแบบสมมาตร/ การเข้ารหัสแบบอสมมาตร / การเข้ารหัสแบบไฮบริด

สรไกร ไกรปุย: การผสมความลับในการรักษาความปลอดภัยข้อมูลแบบไม่เท่าเทียมและการเข้ารหัสแบบไฮบริด (SECRET MIXING IN UNEQUAL SECURITY PROTECTION (USP) AND HYBRID ECRYPTION) คณะกรรมการควบคุมวิทยานิพนธ์: อภิรัฐ ลิ่มมณี, ปร.ด. 43 หน้า. ปี พ.ศ. 2560.

ระบบรักษาความปลอดภัยของข้อมูลบนอุปกรณ์พกพาต่าง ๆ ในปัจจุบัน สามารถใช้การเข้ารหัสแบบสมัยใหม่ได้ ซึ่งมีความปลอดภัยสูงมาก เช่น Advanced encryption standard (AES) และอื่น ๆ แต่วิธีการเข้ารหัสแบบสมัยใหม่นี้ถ้าใช้กับไฟล์ที่มีขนาดใหญ่จำเป็นต้องใช้กับฮาร์ดแวร์ที่มีประสิทธิภาพในการประมวลผลสูง เพื่อให้ได้ผลลัพธ์ที่รวดเร็วแต่การเข้ารหัสบนอุปกรณ์พกพานั้น มีข้อจำกัดทางด้านฮาร์ดแวร์ ซึ่งส่งผลให้มีการประมวลผลที่ช้ากว่าการประมวลผลด้วยคอมพิวเตอร์ ดังนั้นจึงต้องมีการหาเทคนิคการเข้ารหัสที่เหมาะสมกับการใช้บนอุปกรณ์พกพา โดยการใช้เทคนิคการเข้ารหัสแบบไฮบริด (Hybrid encryption) ซึ่งประกอบด้วย AES และ Elliptic curve cryptography (ECC) ร่วมกับเทคนิค Secret mixing เพื่อลดเวลาในการประมวลผล แต่ยังคงมีประสิทธิภาพด้านความปลอดภัยสูง

55910297: MAJOR: ELECTRICAL ENGINEERING; M.Eng.
(ELECTRICAL ENGINEERING)
KEYWORDS: ENCRYPTION/ SECRET MIXING/ SYMMETRIC ENCRYPTION/
ASYMMETRIC ENCRYPTION/ HYBRID ENCRYPTION
SORAKRAI KRAIPUI: SECRET MIXING IN UNEQUAL SECURITY
PROTECTION (USP) AND HYBRID ECRYPTION. ADVISORY COMMITTEE: APIRATH
LIMMANEE, Ph.D. 43 P. 2017.

Data security systems in today's portable devices use modern encryption techniques which are very secure, such as Advance Encryption Standard (AES). However, these modern encryption techniques, when employed for large files, require highly computationally efficient hardware. On portable devices, unfortunately, there is hardware limitation which results in slower data processing than that processed by computers. Therefore, we propose a suitable encryption technique for portable devices. The technique is composed of hybrid encryption, which is a combination of AES and Elliptic Curve Cryptography (ECC), and secret mixing, which can reduce processing time while maintaining high security level.

สารบัญ

หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

สารบัญ..... ฉ

สารบัญตาราง..... ซ

สารบัญภาพ..... ฌ

บทที่

1 บทนำ..... 1

ความเป็นมาและความสำคัญของปัญหา..... 1

วัตถุประสงค์ของการวิจัย..... 1

ประโยชน์ที่ได้รับ..... 1

ขอบเขตของการวิจัย..... 2

ขั้นตอนและวิธีการดำเนินการวิจัย..... 2

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง..... 3

เทคโนโลยีการเข้ารหัสข้อมูล..... 3

SYMMETRIC ENCRYPTION..... 3

DATA ENCRYPTION STANDARD..... 3

ADVANCE ENCRYPTION STANDARD..... 4

SYMMETRIC ENCRYPTION COMPARISON..... 7

ASYMMETRIC ENCRYPTION..... 10

RIVEST-SHAMIR-ADLEMAN..... 10

ELLIPTIC CURVE CRYPTOGRAPHY..... 12

HYBRID ENCRYPTION..... 19

HYBRID ENCRYPTION ALGORITHMS OF AES AND ECC..... 19

SECURE NETWORK CODING..... 21

3 ขั้นตอนและวิธีการดำเนินงาน..... 23

ภาพรวมของระบบ..... 23

กระบวนการดำเนินงาน..... 24

สารบัญ (ต่อ)

บทที่		หน้า
	KEY ENCAPSULATION MECHANISM	24
	ขั้นตอนการพัฒนาของระบบ	29
	SECRET MIXING AND DEMIXING	31
	UNEQUAL SECURITY PROTECTION	32
	USP SYSTEM DESIGN	32
4	ผลการวิจัย	35
	เครื่องมือที่ใช้ในการทดลอง	35
	วิธีการทดลอง	35
	ผลการทดลอง	36
5	สรุปผล อภิปรายผล และข้อเสนอแนะ	39
	สรุปผลการศึกษา	39
	ข้อเสนอแนะแนวทางในอนาคต	40
บรรณานุกรม		41
ประวัติย่อของผู้วิจัย		43

สารบัญตาราง

ตารางที่ หน้า

2-1 ผลการทดสอบอัลกอริทึมของ Symmetric ในแบบต่าง ๆ 8
2-2 จุด GF (23) ทั้งหมด 16
2-3 ระยะเวลาที่ใช้ในการทำงานของ ECC 17
2-4 ระยะเวลาที่ใช้ในการทำงานของ RSA 18
2-5 ประสิทธิภาพของขนาดกุญแจที่ใช้รหัสของ RSA และ ECC 19
5-1 แบบของการทำ Hybrid Encryption 39
5-2 การเปรียบเทียบรูปแบบของ Hybrid Encryption ในแต่แง่มุม 40

สารบัญภาพ

ภาพที่ หน้า

2-1 แผนภาพการทำงานของ DES Algorithm ในการเข้ารหัสข้อมูล..... 4

2-2 ขั้นตอนการทำงานของการเข้ารหัสและถอดรหัสด้วย AES..... 5

2-3 กระบวนการแทนค่าข้อมูลจาก S-box..... 6

2-4 ตาราง S-box ของ Rijndael..... 6

2-5 กระบวนการ ShiftRows ในแต่ละแถว..... 7

2-6 กระบวนการ MixColumn ในแต่ละแถว..... 7

2-7 ผลการทดสอบระหว่าง AES กับ DES..... 9

2-8 กราฟแสดงความสัมพันธ์ของสมการ Elliptic curve..... 12

2-9 กราฟของ Elliptic curve over GF (2^m)..... 13

2-10 กราฟของ Elliptic curve over GF (23)..... 14

2-11 ขั้นตอนการเข้ารหัสแบบ Hybrid..... 20

2-12 ขั้นตอนการถอดรหัสแบบ Hybrid..... 20

2-13 รูปแบบของ Secure network coding ที่ระดับต่าง ๆ..... 21

3-1 ภาพรวมของระบบ e-book online streaming..... 23

3-2 Key encapsulate mechanism without secret mixing..... 26

3-3 Key encapsulation mechanism with secret mixing..... 28

3-4 ขั้นตอนการส่งข้อมูล..... 29

3-5 ขั้นตอนการรับข้อมูล..... 30

3-6 System model of pure AES encryption in baseband..... 32

3-7 System model One-Half AES encryption with secret mixing in baseband..... 33

3-8 System model One-Half AES encryption and One-Half RC4 encryption in baseband.. 33

3-9 System model One-Half AES encryption and One-Half RC4 encryption with secret mixing in baseband..... 34

4-1 Packet windows scale encryption..... 35

4-2 Computation time of USP encryption process..... 37

4-3 ผลลัพธ์สำหรับการแบ่ง Key เป็น 2 ชิ้น (n=2) (a) และแบบมี Secret mixing (b)..... 38

4-4 ผลลัพธ์สำหรับการแบ่ง Key เป็น 4 ชิ้น (n=4) (a) และแบบมี Secret mixing (b)..... 38

บทที่ 1
บทนำ

ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการใช้งานอุปกรณ์จำพวกโมบายมีอัตราการใช้งานเติบโตสูงขึ้นเป็นอย่างมาก โดยอุปกรณ์เหล่านั้นมีความสามารถมากขึ้นจนทำให้การใช้งานในหลาย ๆ อย่าง สามารถทำงานแทนคอมพิวเตอร์ส่วนบุคคลได้ แต่เรื่องความปลอดภัยของข้อมูลที่อยู่บนอุปกรณ์เหล่านั้นยังใช้งานให้มีประสิทธิภาพเทียบเท่าคอมพิวเตอร์ไม่ได้ เพราะมีข้อจำกัดทางด้านทรัพยากรต่าง ๆ เช่น CPU, RAM, Storage เป็นต้น ดังนั้นจึงจำเป็นต้องหาเทคนิคด้านความปลอดภัยให้มีประสิทธิภาพและเหมาะสมกับการใช้งานบนอุปกรณ์พกพาเหล่านั้น

ระบบรักษาความปลอดภัยบนของข้อมูลบนอุปกรณ์พกพาต่าง ๆ ในปัจจุบัน สามารถใช้การเข้ารหัสแบบสมัยใหม่ได้ซึ่งมีความปลอดภัยสูงมาก เช่น Advance encryption standard (AES) (FIPS 197, 2001) และอื่น ๆ แต่วิธีการเข้ารหัสแบบสมัยใหม่นี้ถ้าใช้กับไฟล์ที่มีขนาดใหญ่จำเป็นต้องใช้กับฮาร์ดแวร์ที่มีประสิทธิภาพในการประมวลผลสูง เพื่อให้ได้ผลลัพธ์ที่รวดเร็ว แต่การเข้ารหัสบนอุปกรณ์พกพานั้นมีข้อจำกัดทางด้านฮาร์ดแวร์ ซึ่งส่งผลให้มีการประมวลผลที่ช้ากว่าการประมวลผลด้วยคอมพิวเตอร์ ดังนั้นจึงต้องมีการหาเทคนิคการเข้ารหัสที่เหมาะสมกับการใช้บนอุปกรณ์พกพา โดยการใช้เทคนิคการเข้ารหัสแบบไฮบริด (Hybrid encryption) (Hu & Ma , 2010) ซึ่งประกอบด้วย AES และ Elliptic curve cryptography (ECC) ร่วมกับ Secret mixing เพื่อลดเวลาในการประมวลผล แต่ยังคงมีประสิทธิภาพด้านความปลอดภัยสูง

วัตถุประสงค์การวิจัย

1. เพื่อศึกษาเทคนิคการเข้ารหัสต่าง ๆ ที่สามารถนำประยุกต์ใช้งานกับอุปกรณ์จำพวกโมบายได้อย่างเหมาะสม
2. เพื่อนำเทคนิคการเข้ารหัสมาใช้งานร่วมกันเพื่อความเร็วในการทำงานแต่ยังคงประสิทธิภาพด้านความปลอยภัยอยู่

ประโยชน์ที่ได้รับ

1. สามารถเข้าใจการทำงานของการเข้ารหัสและถอดรหัสในแบบต่าง ๆ

2. สามารถสร้างระบบการเข้ารหัสและถอดรหัสที่มีความปลอดภัยมากขึ้นหรือทำงานได้เร็วขึ้น

3. สามารถนำระบบการเข้ารหัสไปประยุกต์ใช้ให้เหมาะสมกับอุปกรณ์โมบายหรือบอร์ด Embedded ที่มีข้อจำกัดด้านทรัพยากรได้

ขอบเขตของการวิจัย

การวิจัยครั้งนี้ ผู้วิจัยได้ทำการศึกษาและพัฒนาเทคนิคการเข้ารหัสข้อมูลให้เหมาะสำหรับการนำไปใช้บนอุปกรณ์โมมาย ที่มีทรัพยากรจำกัด โดยมีขอบเขตการวิจัยดังนี้

1. ทดสอบเทคนิคการเข้ารหัสต่าง ๆ บนอุปกรณ์โมบาย เพื่อเปรียบเทียบความเร็วในการทำงาน

2. เลือกเทคนิคของการเข้ารหัสที่ต้องการใช้งานร่วมกัน นำมาทดสอบเปรียบเทียบกับเทคนิคการเข้ารหัสเพียงตัวเดียว

ขั้นตอนและวิธีการดำเนินการศึกษา

1. ศึกษาอัลกอริทึมการเข้ารหัสทั้งแบบ Symmetric และ Asymmetric

2. ออกแบบการทดลองและเขียนโปรแกรมเพื่อทดสอบประสิทธิภาพการเข้ารหัสที่สภาพแวดล้อมต่าง ๆ

3. สรุปผลการทำงานเพื่อนำมาวิเคราะห์และหารูปแบบการเข้ารหัสที่เหมาะสมกับอุปกรณ์โมบายและบอร์ด Embedded

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เทคโนโลยีการเข้ารหัสข้อมูล

สำหรับเทคโนโลยีการเข้ารหัสข้อมูลจะเป็นออกเป็น 2 กลุ่มใหญ่ ๆ คือ Symmetric encryption และ Asymmetric encryption ซึ่งแต่ละวิธีจะมีข้อดี ข้อเสียแตกต่างกันไปขึ้นอยู่กับการใช้งานโดยจะอธิบายดังต่อไปนี้

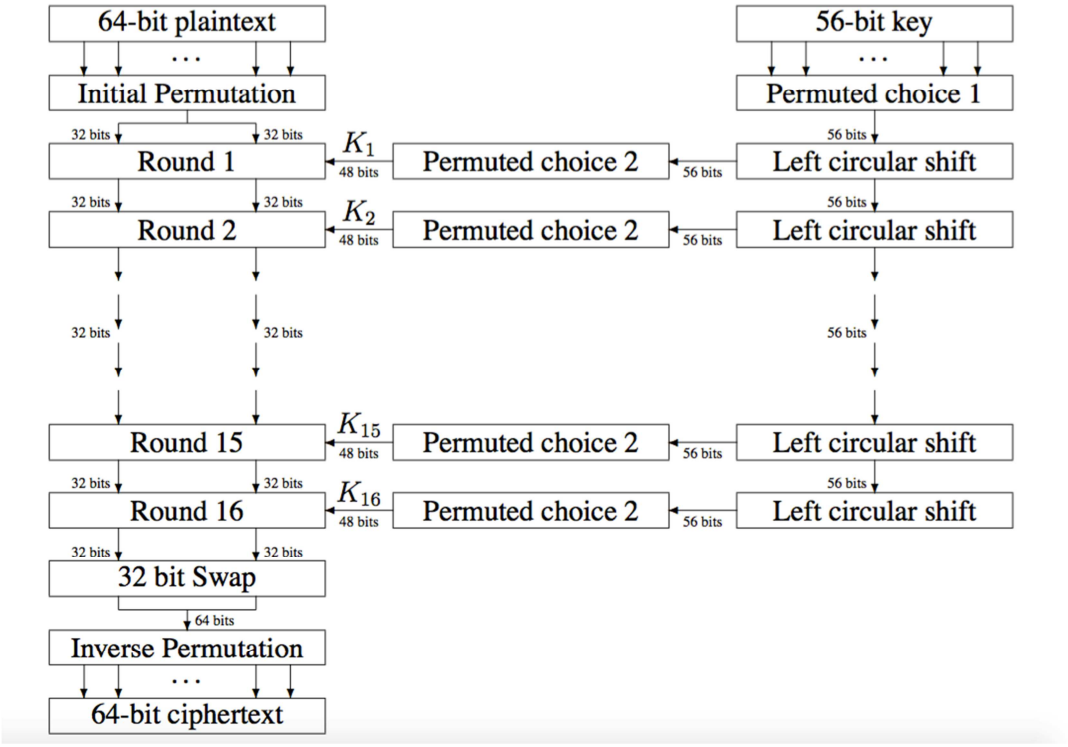
SYMMETRIC ENCRYPTION

การเข้ารหัสแบบ Symmetric นั้นรูปแบบจะเป็นการเข้ารหัสและถอดรหัสด้วยกุญแจตัวเดียวกัน โดยจะมีอยู่ด้วยกันหลายอัลกอริทึม การนำไปใช้ก็เพื่อให้เหมาะสมในแต่ละงาน เพราะแต่ละอัลกอริทึมจะมีข้อดีข้อเสียแตกต่างกัน ซึ่งอัลกอริทึมที่ได้รับความนิยมก็มีอยู่หลายตัว

DATA ENCRYPTION STANDARD

มาตรฐานการเข้ารหัสแบบ Data encryption standard (DES) เป็นอัลกอริทึมที่พัฒนาโดย IBM และได้การรับรองโดยรัฐบาลสหรัฐอเมริกาในปี ค.ศ. 1977 ให้เป็นข้อมูลมาตรฐานการเข้ารหัสข้อมูลสำหรับหน่วยงานรัฐทั้งหมด

DES เป็นอัลกอริทึมแบบ Block cipher ที่มีขนาด Block 64 บิตและใช้กุญแจที่มีขนาดความยาว 56 บิต ซึ่งถือว่าเป็นอัลกอริทึมที่มีความปลอดภัยสูงในสมัยนั้น แต่ด้วยปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์ที่พัฒนาไปอย่างรวดเร็ว ทำให้อัลกอริทึมนี้ที่มีความยาวของกุญแจเพียง 56 บิต อาจจะไม่เพียงพอสำหรับการปกป้องข้อมูลที่สำคัญเพราะสามารถถูกถอดรหัสได้ไม่ยากมากนัก

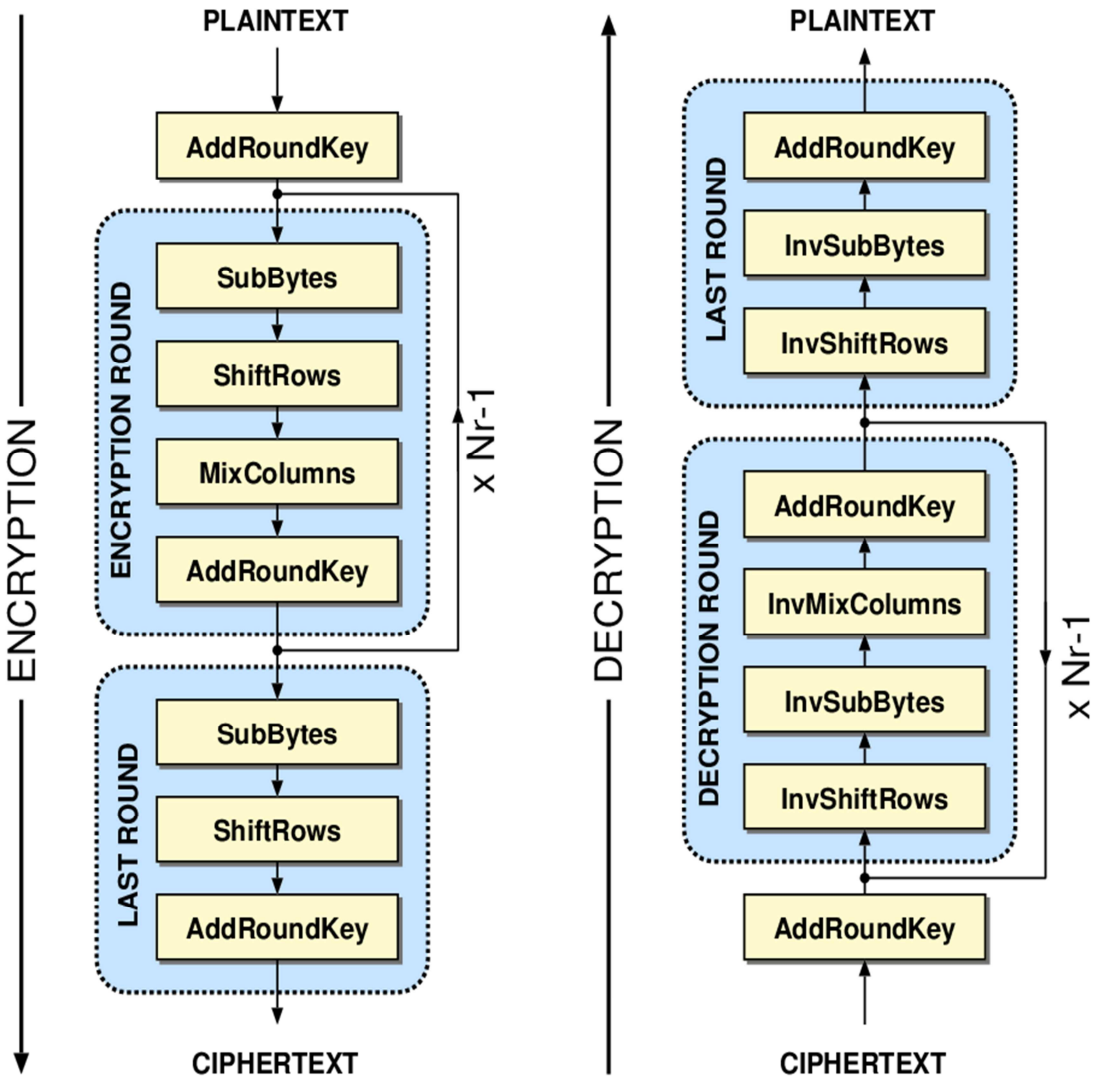


ภาพที่ 2-1 แผนภาพการทำงานของ DES Algorithm ในการเข้ารหัสข้อมูล

ADVANCE ENCRYPTION STANDARD

มาตรฐานการเข้ารหัสแบบ Advance encryption standard (AES) ถูกพัฒนาโดย Joan Daemen และ Vincent Rijmem ในเดือนมกราคมปี 1997 สถาบันมาตรฐานแห่งชาติและเทคโนโลยี (NIST) ประกาศการเริ่มต้นที่จะพัฒนา AES อัลกอริทึม โดยผลของการวิจัยได้ศึกษาอัลกอริทึม MARS, RC6TM, Rijndael เพื่อที่จะใช้กับการเข้ารหัสแบบ AES ทาง NIST ได้ตัดสินใจให้อัลกอริทึมของ Rijndael นำมาใช้เป็นการเข้ารหัสแบบ AES โดยประกาศรับรองอย่างเป็นทางการในปี 2000

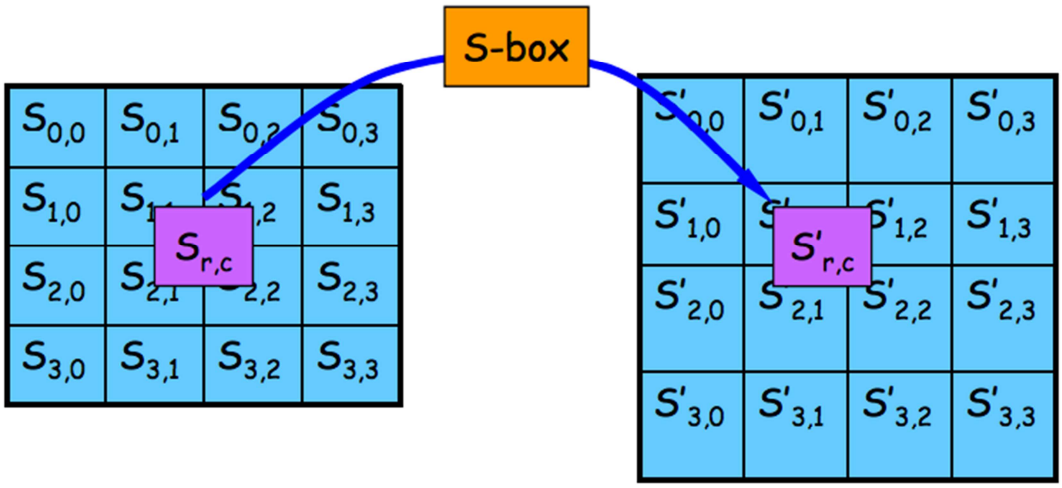
AES เป็นอัลกอริทึมแบบ Block cipher ขนาด 128 บิต แต่สามารถใช้กับกุญแจที่มีขนาด 128, 192 และ 256 บิต ได้ และ AES ได้รับความนิยมมากในปัจจุบันเพราะมีความปลอดภัยที่สูง และทำงานได้รวดเร็ว อัลกอริทึมของ AES ได้แสดงดังภาพที่ 2-2 โดยเริ่มต้นด้วย Add round key, Sub bytes, Shift rows, Mix columns ในแต่ละรอบส่วนรอบสุดท้ายจะมีเพียงแค่ Sub bytes, Shift rows และ Add round key เท่านั้น



ภาพที่ 2-2 ขั้นตอนการทำงานของการเข้ารหัสและถอดรหัสด้วย AES

AES นี้จะประกอบไปด้วย 4 ขั้นตอน ดังนี้

1. Sub bytes ในทุก ๆ Block จะถูกแทนที่ด้วยข้อมูลจาก S-box ที่ตำแหน่งเดียวกัน

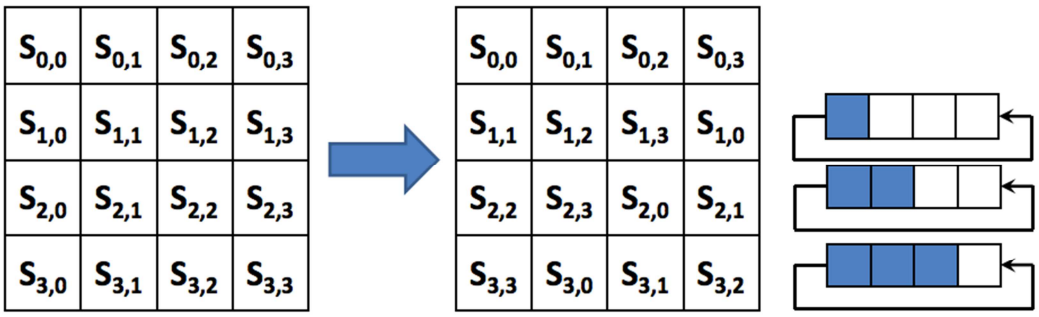


ภาพที่ 2-3 กระบวนการแทนค่าข้อมูลจาก S-box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

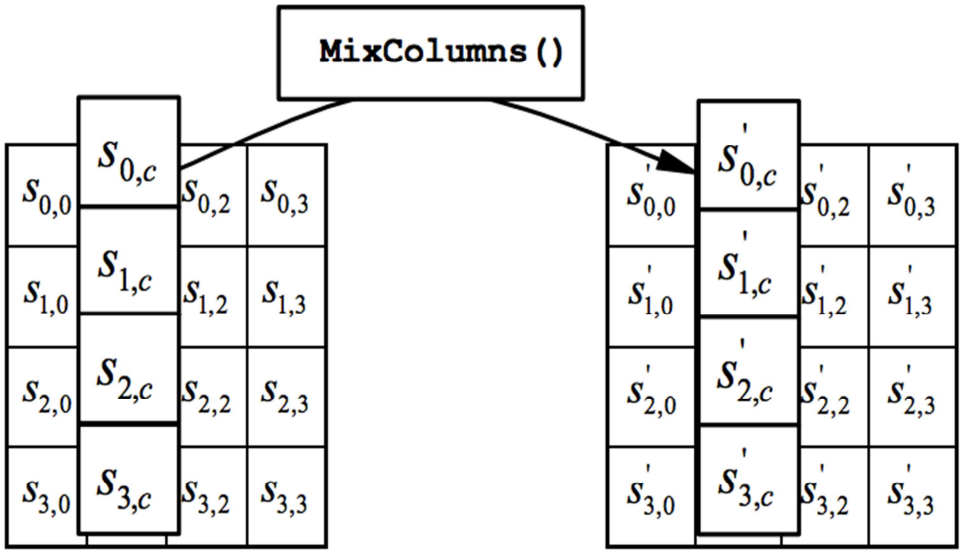
ภาพที่ 2-4 ตาราง S-box ของ Rijndael

2. ShiftRows ข้อมูลในแต่ละแถวจะถูกเลื่อนไปทางซ้ายโดยจำนวนที่ถูกเลื่อนจะเท่ากับหมายเลขของแถวนั้น



ภาพที่ 2-5 กระบวนการ ShiftRows ในแต่ละแถว

3. MixColumn เป็นกระบวนการนำในแต่ละแนวหลักของตารางข้อมูล ไปผสมกับค่าคงที่ ในตาราง S-box ของ Rijndael



ภาพที่ 2-6 กระบวนการ MixColumn ในแต่ละแถว

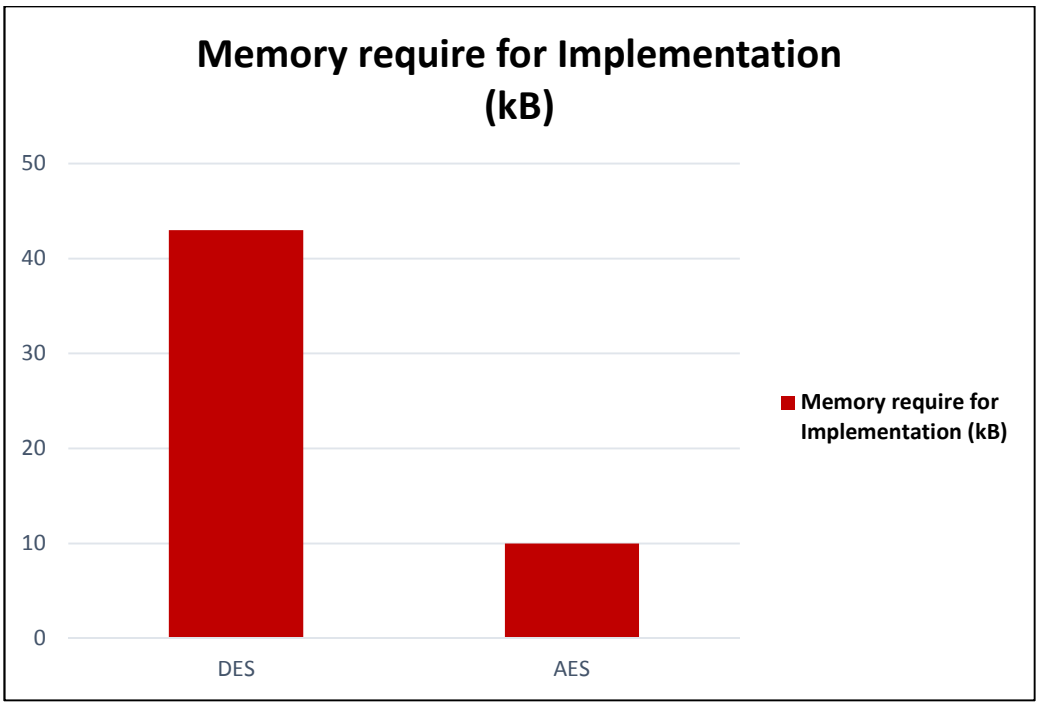
SYMMETRIC ENCRYPTION COMPARISON

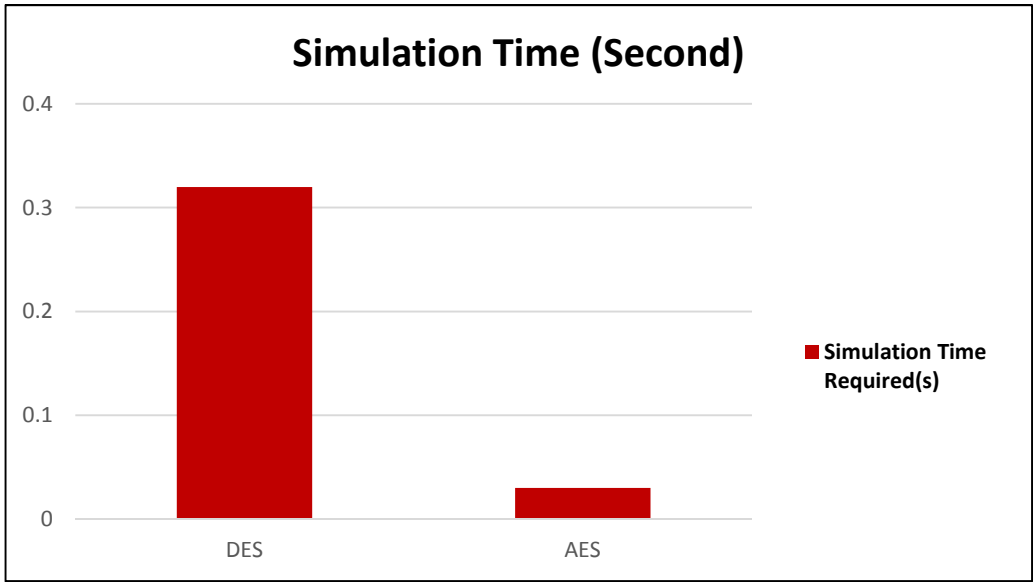
โดย (Abdul Elminaam, Abdul Kader, & Hadhoud , 2009) ได้ทำการทดสอบและเปรียบเทียบประสิทธิภาพของอัลกอริทึมแต่ละตัวดังตารางที่ 2-1

ตารางที่ 2-1 ผลการทดสอบอัลกอริทึมของ Symmetric ในแบบต่าง ๆ

Algorithm	Platform	Encryption (kbit/s)	Key-setup (ms)
AES	MSP430	290.0	0.6
Twofish	ATmega	235.0	271.9
SAFER	MSP430	233.0	1.7
AES	ATmega	223.0	49.0
Twofish	MSP430	212.0	15.0
RC2	MSP430	161.0	1.2
SAFER	ATmega	158.0	1.8
Noekeon	MSP430	139.0	0.1
RC2	ATmega	138.4	1.4
DES	MSP430	123.0	16.6
RC5	MSP430	121.0	3.3
Kasumi	MSP430	120.0	0.3

ซึ่งจากการทดสอบ AES นั้นเป็นอัลกอริทึมที่มีประสิทธิภาพในการเข้ารหัสสูงที่สุด และยังมี (Akash Kumar Mandal, Chandra Parakash, & Archana Tiwari, 2012) ได้ทำการทดสอบเปรียบเทียบกันระหว่าง DES กับ AES ซึ่งได้ผลดังภาพที่ 2-7





ภาพที่ 2-7 ผลการทดสอบระหว่าง AES กับ DES

ซึ่งผลลัพธ์ที่ได้นั้น AES มีประสิทธิภาพสูงกว่า DES ทั้งในเรื่องของ Memory และเวลาที่ใช้ในการทำงานจากการทดลองนั้น AES จึงถือได้ว่าเป็น Symmetric encryption ที่มีประสิทธิภาพที่สูงสุดในขณะนี้

ASYMMETRIC ENCRYPTION

การเข้ารหัสแบบ Asymmetric นั้นจะแตกต่างกับ Symmetric ตรงที่กุญแจที่ใช้เข้ารหัสนั้นจะมีอยู่ 2 ตัว คือ Public key กับ Private key ซึ่งการเข้ารหัสจะใช้ Public key ในการเข้ารหัส ส่วนการถอดรหัสจะใช้ Private key จะเห็นได้ว่าการเข้ารหัสแบบ Asymmetric นี้จะใช้กุญแจในการเข้าและถอดรหัสคนละตัวกัน ทำให้มีข้อได้เปรียบกว่า Symmetric encryption ในด้านของการจัดการกุญแจที่ดีกว่า สำหรับ Asymmetric encryption นั้นก็มีอัลกอริทึมอยู่หลายชนิดเช่นกันแต่ที่นิยมใช้กันอย่างแพร่หลายในตอนนี้ คือ Rivest-Shamir-Adleman (RSA) (Ren, & Miao, 2010) และมี Elliptic curve cryptography (ECC)

RIVEST-SHAMIR-ADLEMAN

Rivest-Shamir-Adleman (RSA) เป็นอัลกอริทึมในการเข้ารหัสแบบ Asymmetric ที่ถูกใช้งานได้ทั่วไปเป็นตัวแรก ซึ่งถูกคิดค้นโดย Ron Rivest, Adi Shamir และ Leonard Adleman จาก MIT ในปี 1977 โดยนำตัวอักษรตัวแรกของนามสกุลแต่ละคนมาตั้งเป็นชื่ออัลกอริทึม การสร้าง Public key และ Private key ของ RSA นั้นเป็นการสร้างจากจำนวนเฉพาะที่มีขนาดใหญ่มาก 512-4,096 bits ซึ่งทำให้การโจมตีเพื่อหาค่าของจำนวนเฉพาะนี้ทำได้ยากและใช้เวลานานมากเมื่อเทียบกับการใช้คอมพิวเตอร์ในสมัยนั้นทำการคำนวณ

Public key cryptography เป็นระบบการเข้ารหัสโดยจะมีการสร้างกุญแจ 2 ดอก ที่เป็นคู่กัน คือ Public key และ Private key โดยที่ Public key นั้นจะสามารถเปิดเผยให้ภายนอกมองเห็นได้ ส่วน Private key นั้นจะมีแต่ผู้สร้างกุญแจเท่านั้น โดยการเข้ารหัสในลักษณะนี้จะนิยมใช้ในการเข้ารหัสข้อความเพื่อยืนยันตัวตนผ่านระบบเครือข่ายซึ่งจะมีแต่ผู้ที่มี Private key เท่านั้นถึงจะสามารถถอดรหัสได้

การทำงานของ RSA นั้นจะทำการสร้างคู่ของกุญแจมาจากจำนวนเฉพาะ เหตุผลที่เลือกใช้จำนวนเฉพาะก็เพราะว่าคุณสมบัติพิเศษอันหนึ่งของจำนวนเฉพาะเมื่อเอามาคูณกัน จะมีแค่จำนวนเฉพาะคู่นั้นที่หารผลคูณได้ Key ส่วนใหญ่จะสร้างโดยใช้จำนวนเฉพาะสองตัวคูณกันเป็นเลขยกกำลังแล้วเอาผลลัพธ์มาทำการคำนวณร่วมกับการ Mod การจะหา Key ได้จะต้องเอาผลคูณมาแยกตัวประกอบซึ่งทำได้ยาก เพราะมีตัวประกอบเพียงสองตัว ซึ่งกระบวนการทำงานนั้นสามารถแบ่งออกได้เป็น 4 ขั้นตอน ดังนี้

1. Key generation

1.1 เลือกจำนวนเฉพาะที่มีขนาดใหญ่ 2 จำนวนให้เป็น p และ q โดยขึ้นอยู่กับขนาดของกุญแจ เช่น 512 1,024 2,048 เป็นต้น

1.2 คำนวณหา $n = pq$ และ $\varphi(n) = (p-1)(q-1)$

1.3 เลือกจำนวนเต็มมาค่าหนึ่งโดยที่ $1 < e < \varphi(n)$ โดยที่ e เป็นจำนวนเฉพาะสัมพัทธ์กับ $\varphi(n)$

1.4 หาค่า d ซึ่ง $ed = 1 \ mod(\varphi(n))$

1.5 Public key คือ (n, e) ส่วน Private key คือ (n, d)

2. Key distribution

ทำการแลกกุญแจกันระหว่างผู้รับกับผู้ส่งโดยสมมติให้ Alice คือ ผู้รับและ Bob คือ ผู้ส่ง โดย Bob จะต้องเข้ารหัสด้วย Public key ของ Alice ก่อนจึงจะส่งให้กับ Alice เพื่อนำไปถอดรหัสด้วย Private key ของตัวเอง

3. Encryption

ในกรณีนี้ Bob ต้องการส่งข้อความ M ให้กับ Alice ในขั้นแรก Bob จะต้องแปลงข้อความ M ให้กลายเป็นตัวเลข m โดยที่ $0 \leq m < n$ และ $\gcd(m, n) = 1$ โดยใช้วิธีที่เรียกว่า padding scheme และเมื่อได้ค่า e, d, n มาแล้วก็จะสามารถนำค่าเหล่านี้มาเข้ารหัสได้ดังนี้

$$c = m^2 mod\ n \tag{2-1}$$

โดยที่ c คือ ข้อความที่เข้ารหัสแล้ว

4. Decryption

การถอดรหัสนั้นจะมีขั้นตอนคล้ายกับการเข้ารหัสแต่จะมีการนำค่า d ที่หาไว้มาเป็นเลขชี้กำลังของ C ดังสมการ

$$c^d = (m^e)^2 = m \tag{2-2}$$

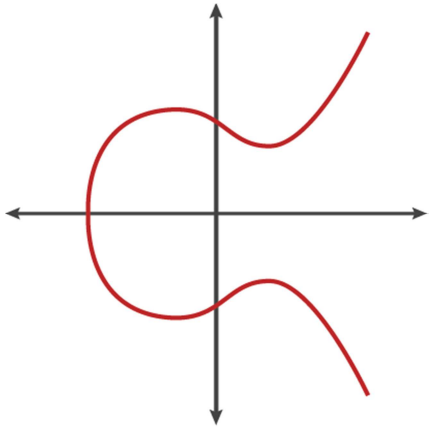
จากสมการจะเห็นว่าเมื่อนำ c^d หารด้วย n จะได้ m ซึ่งสามารถนำไปแปลงกลับเป็นข้อความ M ได้

ความปลอดภัยของอัลกอริทึมนี้ขึ้นอยู่กับความยากในการแยกตัวประกอบของเลขจำนวนเฉพาะที่มีค่ามาก ๆ ซึ่งถ้าเป็นการแยกตัวประกอบจากกุญแจที่มีขนาดเล็กจะทำให้แยกตัวประกอบได้ง่าย เช่น ในปี ค.ศ. 1999 มีผู้สามารถแยกตัวประกอบของตัวเลขขนาด 512 บิท ได้ในเวลา 7 เดือน ดังนั้นทางแล็ปของ RSA จึงแนะนำให้ใช้กุญแจขนาด 1,024 หรือ 2,048 ขึ้นไปในการเข้ารหัสเพื่อความปลอดภัยในปัจจุบัน

ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography (ECC) เป็นอัลกอริทึมการเข้ารหัสแบบ Asymmetric ได้รับการนำเสนอโดย Neal Koblitz และ Victor S. Miller ในปี 1985 โดยอัลกอริทึมการเข้ารหัสแบบ ECC นี้ได้พัฒนามาจากสมการของเส้นโค้งวงรี

$$y^2 = x^3 + ax + b \tag{2-3}$$

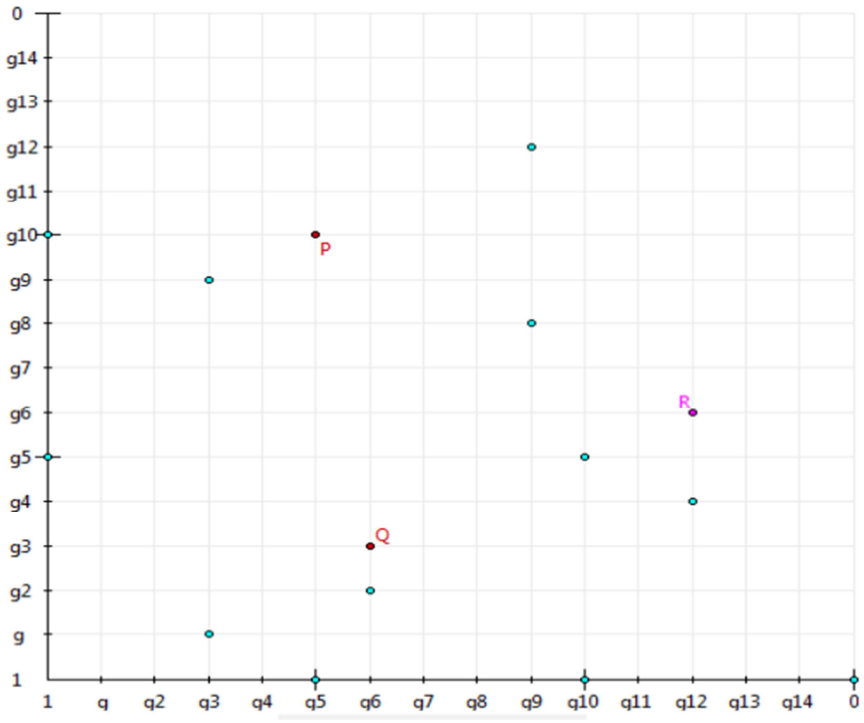


ภาพที่ 2-8 กราฟแสดงความสัมพันธ์ของสมการ Elliptic curve

อัลกอริทึม ECC นั้นจะมีข้อดีที่เหนืออัลกอริทึมแบบ Asymmetric ก่อนหน้านี้ เช่น RSA คือ ที่ความปลอดภัยเท่ากันจะใช้ขนาดของกุญแจที่เล็กกว่า ซึ่งในทางเดียวกันถ้าใช้กุญแจที่มีขนาดเท่ากัน ECC จะให้ระดับความปลอดภัยที่มากกว่า หากใช้การโจมตีแบบ Brute-Force จะใช้เวลามากกว่าของ RSA เนื่องจาก ECC นั้นมีขนาดของกุญแจที่เล็กกว่า RSA มากและมีการใช้เวลาในการคำนวณที่รวดเร็ว ใช้พลังงานต่ำ ดังนั้น ECC จึงเหมาะกับการใช้งานในการเข้ารหัสบนอุปกรณ์ที่มีทรัพยากรจำกัดเมื่อเทียบกับคอมพิวเตอร์ทั่วไป เช่น โทรศัพท์สมาร์ทโฟนหรือบอร์ดสมองกลฝังตัว เป็นต้น

Elliptic Curve Group Over Finite Fields

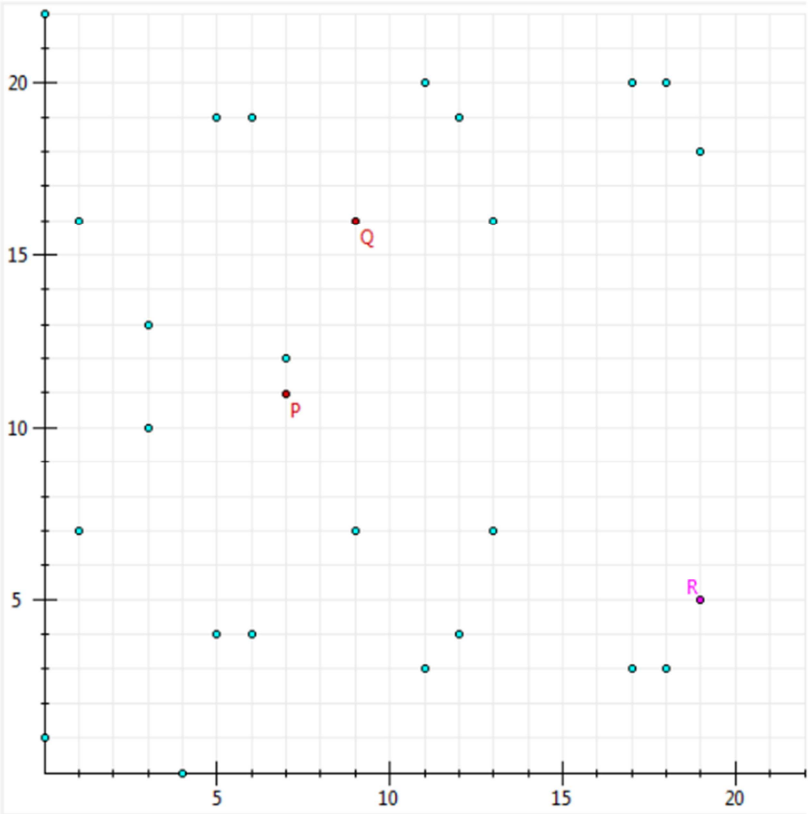
1. Over GF (2^m) Polynomials เป็นกราฟที่ได้จากสมการ $y^2 + xy = x^3 + ax + b$ โดยที่ $b \neq 0$ กำหนดให้ $P = (x_1, y_1), Q = (x_2, y_2)$ คือ จุดบนกราฟของสมการ ดังแสดงในตัวอย่างภาพที่ 2-9



ภาพที่ 2-9 กราฟของ Elliptic Curve Over GF(2^m)

$$\begin{aligned} P &= (g5, g10) \\ Q &= (g6, g3) \\ R &= P + Q = (g12, g6) \\ y^2 + xy = x^3 + x^2 + 1; &Polynom\ f = x^4 + x + 1; m = 4 \end{aligned} \tag{2-4}$$

2. Over GF (p) (Prime Number) เป็นกราฟที่ได้จากสมการ $y^2\ mod\ p = (x^3 + ax + b)\ mod\ p$ โดยที่ $4a^3 + 27b^2\ mod\ p \neq 0$ ดังแสดงในภาพตัวอย่างที่ 2-10



ภาพที่ 2-10 กราฟของ Elliptic Curve Over GF(23)

$$
\begin{aligned}
&P = (7,11)\\
&Q = (9,16)\\
&R = P + Q = (19,15)\\
&y^2 mod\ 23 = (x^3 + x + 1) mod\ 23
\end{aligned}
\tag{2-5}
$$

3. กฎการบวกระหว่างจุดบนกราฟ GF(p)

กำหนดให้ $P = (x_1, y_1), Q = (x_2, y_2)$ คือ จุดบนกราฟของสมการ $P + Q = (x_3, y_3)$ โดยที่

$$x_3 = \lambda^2 - x_1 - x_2, y_3 = \lambda(x_1 - x_3) - y_1 \tag{2-6}$$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & if\ P \neq Q \\ \dfrac{2x^2 + a}{2y_1}, & if\ P = Q \end{cases} \tag{2-7}$$

4. กฎการลบระหว่างจุดบนกราฟ GF (p)

กำหนดให้ $P = (x_1, y_1), Q = (x_2, y_2)$ คือ จุดบนกราฟของสมการ $P - Q = P + (-Q)$ โดยที่

$$-Q = (x_2, y_2) mod\ p \tag{2-8}$$

5. กฎการคูณค่าคงที่กับจุดบนกราฟ GF (p)

กำหนดให้ $P = (x_1, y_1), Q = (x_2, y_2)$ คือ จุดบนกราฟของสมการ ถ้า $P = Q$ จะได้ $P + P = 2P = R = (x_3, y_3)$

เมื่อ k คือ จำนวนเต็มบวกใดๆจะได้

$$Q = kP = \underbrace{P + P + \ldots + P}_{k} \tag{2-9}$$

เช่น ถ้า $k = 9, Q = kP = 9P = 2\big(2(2P)\big) + P$

6. การเข้ารหัสและถอดรหัส

การเข้ารหัสข้อความนั้นผู้ส่ง A (Alice) จะนำข้อความ P_m มาทำการคำนวณหาข้อความที่เข้ารหัส C_m แล้วส่งไปยังผู้รับ B (Bob) ซึ่ง

$$c_m = \{kG, P_m + kP_B\} \tag{2-10}$$

โดยที่

G คือ จุดที่ได้จากการ Generate บน Elliptic Curve

k คือ ตัวเลขสุ่มจำนวนเต็มบวกที่เลือกโดย A

P_B คือ P ของ B ซึ่ง $P_B = n_B \times G$

n_B คือ Private key ของ B

การถอดรหัส B จะนำ Private key มาคูณค่าจุดแรกและนำผลลัพธ์ไปลบออกจากค่าจุดที่สองดังต่อไปนี้

$$P_m + kP_B - n_N(kG) = P_m + k(n_B)G - n_B(kG) = P_m \tag{2-11}$$

ตัวอย่างการเข้ารหัสและถอดรหัส GF(p)

$E_p(a, b) = E_{23}(1,1)$ จะได้ $a = 1, b = 1, p = 23$ สามารถเขียนเป็นสมการได้ดังนี้

$$y\ mod\ 23 = (x^3 + x + 1)\ mod\ 23 \tag{2-12}$$

โดยจุดบนกราฟทั้งหมด GF (23) แสดงดังตารางที่ 2-2

ตารางที่ 2-2 จุด GF (23) ทั้งหมด

จุด P	จุด Q	จุด R
(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

กำหนดให้

เลือกใช้จุด $G = (1,7)$

$P = (9,7)$ ซึ่งแทนด้วยตัวอักษร "M"

ผู้ส่ง A

Private Key $= n_A = 3$

Public Key $= n_A \times G = 3 * (1,7) = (18,20)$

ผู้รับ B

Private Key $= n_p = 5$

Public Key $= n_p xG = 5 \times (1,7) = (0,1)$

เมื่อ A ต้องการส่งข้อความให้ B

1. A สุ่มตัวเลขได้ k = 9
2. คำนวณ $C = \{kG, P_m + kP_B\}$

$$\begin{aligned} C_m &= \{9(1,7), (9,7) + 9(0,1)\} \\ &= \{(9,16), (9,7) + (19,18)\} \\ &= \{(9,16), (13,7)\} \\ &= \{(9,16), (13,7)\} \end{aligned} \tag{2-13}$$

เมื่อ B ได้รับข้อความจากสมการ

$$\begin{aligned} P_m &= (13,7) - 5(9,16) \\ &= (13,7) - (19,18) \\ &= (13,7) + (19,-18) \\ &= (13,7) + (19,5); \quad (5 = -18 \ mod \ 23) \\ &= (9,7) \end{aligned} \tag{2-14}$$

B ได้รับจุด (9, 7) แทนด้วยอักษร "M"

โดยได้ทำการทดสอบประสิทธิภาพของอัลกอริทึมแบบ Asymmetric ดังตารางที่ 2-3 และ 2-4 (Vijayalakshmi, & Bommanna Raja, 2012)

ตารางที่ 2-3 ระยะเวลาที่ใช้ในการทำงานของ ECC

Process / key size	128 bits (milliseconds)	1024 bits (milliseconds)
World: java		
Private key	6.5	19.5
Public key	19.8	59.3
Encryption	9.8	29.3
Decryption	10.1	30.4
World: javaworld		
Private key	0.2	0.5
Public key	7.1	21.2
Encryption	6.9	20.6
Decryption	7.3	21.8
World: javaworldwelcome		
Private key	6.6	19.7
Public key	21.0	62.9
Encryption	10.1	30.2
Decryption	10.5	31.5

ตารางที่ 2-4 ระยะเวลาที่ใช้ในการทำงานของ RSA

Process / Key size	128 bits (Milliseconds)	1024 bits (Milliseconds)
Word: java		
Private key	62.0	186.1
Public key	61.7	185.0
Encryption	72.3	217.0
Decryption	73.1	219.4
Word: javaworld		
Private key	76.1	228.2
Public key	75.5	226.6
Encryption	87.3	261.9
Decryption	88.1	264.3
Word: javaworldwelcome		
Private key	66.9	200.7
Public key	66.5	199.4
Encryption	76.9	230.7
Decryption	77.8	233.7

จากการทดสอบทำให้สรุปได้ว่า ECC นั้นใช้เวลาในการทำงานน้อยกว่า RSA ซึ่งแสดงให้เห็นว่า ECC นั้นมีประสิทธิภาพที่ดีว่า RSA ในด้านของระยะเวลาในการทำงาน ส่วนในเรื่องของความปลอดภัยนั้นทาง NIST ซึ่งเป็นสถาบันที่กำหนดมาตรฐานเทคโนโลยี การนำไปใช้หรือกระบวนการทำงานต่างของสหรัฐอเมริกา ได้มีการทดสอบถึงประสิทธิภาพด้านความปลอดภัยของการเข้ารหัสดังตารางที่ 2-5

ตารางที่ 2-5 ประสิทธิภาพของขนาดกุญแจที่ใช้รหัสของ RSA และ ECC (National Security Agency, 2009)

Symmetric key size (bits)	RSA and Diffle-Hellman key size (bits)	Elliptic curve key size (bits)
80	1,024	160
112	2,048	224
128	3,072	256
192	7,680	384
256	15,360	521

จะเห็นได้ว่า ECC นั้นใช้ขนาดกุญแจที่เล็กกว่า RSA มากแต่ให้ประสิทธิภาพความปลอดภัยที่เท่ากัน จากกุญแจที่มีขนาดเล็กกว่าก็จะทำให้ใช้เวลาในการทำงานนั้นน้อยลง แสดงให้เห็นว่า ECC นั้นมีประสิทธิภาพสูงกว่า RSA

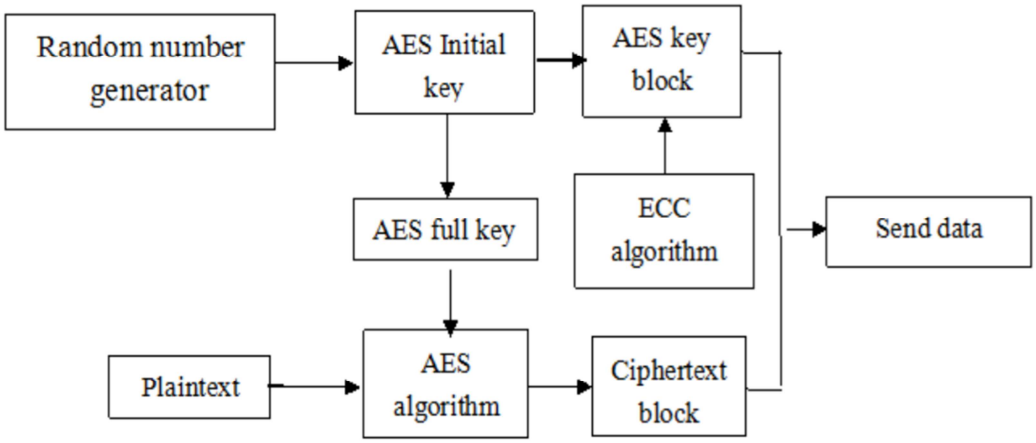
HYBRID ENCRYPTION

ในปัจจุบันนั้นมีการใช้งานอินเตอร์เน็ตกันมากขึ้นหรือมีการส่งข้อมูลผ่าน network กันมากขึ้น ดังนั้นการปกป้องข้อมูลที่สำคัญนั้นจึงมีความสำคัญเป็นอย่างยิ่ง โดยอย่างที่กล่าวในตอนต้นว่าการเข้ารหัสข้อมูลนั้นถูกแบ่งออกเป็น 2 แบบ คือ Symmetric Encryption และ Asymmetric Encryption การที่จะใช้เทคโนโลยีการเข้ารหัสอย่างใดอย่างหนึ่งอาจจะไม่เพียงพอต่อการป้องกันเพราะการส่งข้อมูลผ่าน network มีโอกาสถูกโจมตีข้อมูลมากฉะนั้นจึงมีการนำเทคโนโลยีการเข้ารหัสทั้ง 2 ชนิด มาใช้ร่วมกันโดยนำจุดดีของแต่ละชนิดมาใช้งานเพื่อการปกป้องข้อมูลที่ดียิ่งขึ้น

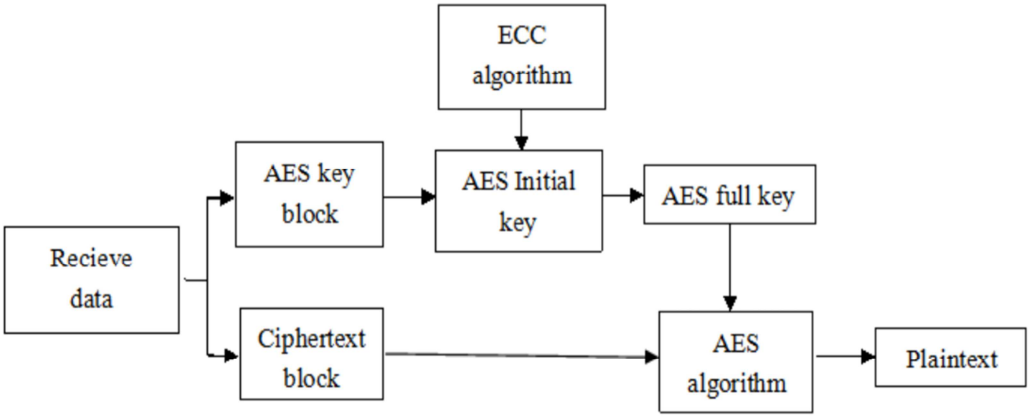
HYBRID ENCRYPTION ALGORITHMS OF AES AND ECC

การใช้อัลกอริทึม AES และ ECC ควบคู่กันในการเข้ารหัสข้อมูลโดยเป็นที่ทราบกันว่า AES นั้นมีความปลอดภัยสูง เข้ารหัสได้เร็วแต่จะมีข้อด้อยเรื่องการจัดการกับกุญแจเพราะทั้ง 2 ฝั่งผู้รับและผู้ส่งจะต้องใช้กุญแจที่เหมือนกันในการเข้ารหัสและถอดรหัสทำให้การส่งกุญแจผ่านระบบ Network มีความเสี่ยงที่จะถูกโจมตีได้ง่าย ดังนั้นความสามารถของ AES นั้นจึงเหมาะสมกับ

การเข้ารหัสข้อมูลที่มีขนาดใหญ่ จึงนำการเข้ารหัสแบบ ECC มาใช้ในการเข้ารหัสกุญแจของ AES เพื่อให้การจัดการส่งกุญแจของ AES ผ่านระบบ Network นั้นปลอดภัยยิ่งขึ้นเนื่องการ ECC เป็นการเข้ารหัสแบบ Asymmetric คือ จะมีการสร้างคู่ของกุญแจระหว่างผู้รับและผู้ส่งซึ่งประกอบด้วย Public key และ Private key โดยผู้รับและผู้ส่งจะทำการแลกเพียง Public key เท่านั้น ส่วนผู้รับจะทำการถอดรหัสโดยใช้ Private key ของตนเองทำให้การรับส่งข้อมูลผ่านระบบ Network มีความปลอดภัยยิ่งขึ้น ซึ่งกระบวนการทำงานของ Hybrid encryption สามารถแสดงได้ดังภาพที่ 2-11 และ 2-12



ภาพที่ 2-11 ขั้นตอนการเข้ารหัสแบบ Hybrid

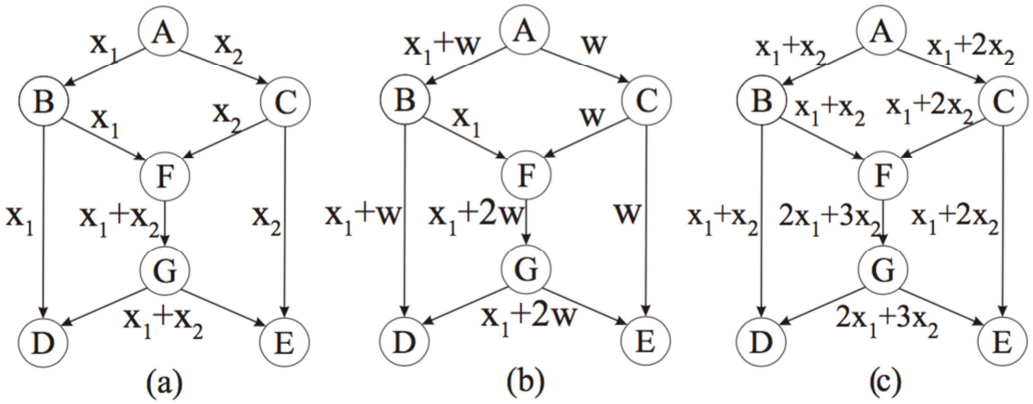


ภาพที่ 2-12 ขั้นตอนการถอดรหัสแบบ Hybrid

SECURE NETWORK CODING

เทคนิค Network coding นั้นเป็นวิธีการเพิ่มประสิทธิภาพของการส่งข้อมูลผ่านระบบ Network ช่วยให้การส่งข้อมูลดีขึ้นเพิ่มความเสถียรและความทนทานของข้อมูล ซึ่งเทคนิคนี้ได้ถูกคิดค้นโดย Ahlswede ในปี 2000 จากเทคนิคนี้ได้ถูกพัฒนามาให้เป็นการเพิ่มความปลอดภัยของข้อมูลที่ถูกส่งออกไป ซึ่งจะถูกเรียกว่า Secure network coding (Cai, & Yeung, 2002)

หลักการของ Secure network coding นั้นจะทำการแบ่งข้อมูลออกเป็น n ส่วนจากนั้นจะสร้างสมการคณิตศาสตร์แล้วนำข้อมูลที่ถูกแบ่งออกผสมกันตามสมการที่สร้างขึ้น เพื่อให้ข้อมูลที่ถูกส่งออกไปนั้นมีความปลอดภัยจากการถูกผู้ไม่ประสงค์ดีดักข้อมูลไปเพราะเนื่องจากข้อมูลมีการผสมกับค่าคงที่จากสมการคณิตศาสตร์ทำให้ยากต่อการเดาข้อมูลที่ได้ไป



ภาพที่ 2-13 รูปแบบของ Secure network coding ที่ระดับต่าง ๆ

จากภาพที่ 2-13 จะเห็นได้เมื่อมีการส่งสัญญาณแบบ Multicast จากโหนด A ไปยัง D และ E นั้นเราจะสามารถทำการเข้ารหัสแบบ Network coding ได้หลายรูปแบบ ในภาพที่ 2-13 (a) นั้นข้อมูล x_1 และ x_2 ที่ส่งออกจากโหนด A จะถูกบวกหรือ XOR กันที่โหนด F ซึ่งในภาพที่ 2-13 (a) นั้นถือเป็นการส่งสัญญาณที่ไม่ปลอดภัย เนื่องจากผู้ดักฟังที่บางเส้นเชื่อมระหว่างโหนดจะสามารถได้รับข้อมูลบางอย่างได้ เช่น ถ้าดักฟังที่เส้นเชื่อม BD ก็จะทราบข้อมูล x_1

หากจะทำการเข้ารหัสแบบ Network coding ให้ปลอดภัยนั้นจะต้องมีการผสมข้อมูลเข้ากับสัญญาณอื่น ตัวอย่างเช่น ในภาพ 2-13 (b) มีการผสม x_1 เข้ากับสัญลักษณ์ที่สุ่มขึ้นมา คือ W จะเห็นว่าถ้าผู้ดักฟัง ดักฟังเพียงเส้นเชื่อมเดียวจะไม่สามารถทราบข้อมูล x_1 ได้เลย ต้องดักฟังตั้งแต่สองเส้นเชื่อมขึ้นไป เทคนิคนี้เรียกว่า Secure network coding

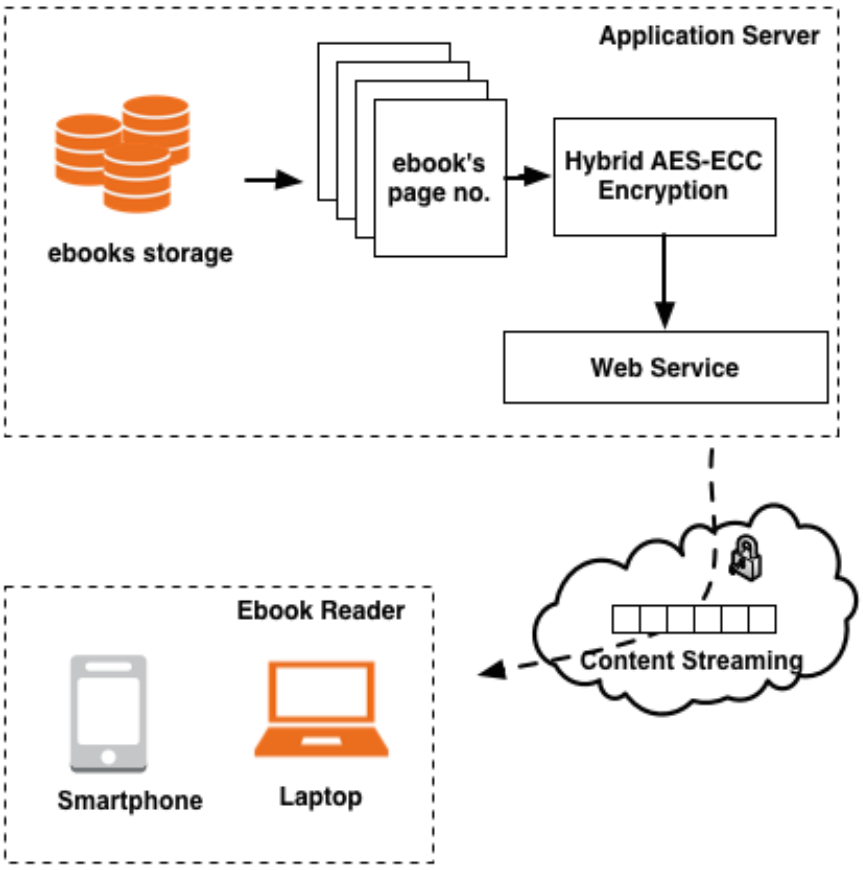
สำหรับในภาพ 2-13 (c) จะเป็นการผสมข้อมูลที่ต้องการส่งเข้าด้วยกันโดยไม่ต้องมี

การสุ่มสัญลักษณ์อื่นขึ้นมาข้อดี คือ สามารถส่งทั้ง x_1 และ x_2 ได้ในเวลาเดียวกัน ข้อเสียคือ แม้ว่าผู้ดักฟังที่เส้นเชื่อมจะไม่ทราบว่า x_1 และเส้นใดเส้นหนึ่ง คืออะไร แต่ก็จะทราบผลลัพธ์ของการผสมข้อมูลบางอย่าง เช่น $x_1 + x_2$ เราจึงเรียกการเข้ารหัสแบบนี้ว่า Weakly secure network coding

บทที่ 3
ขั้นตอนและวิธีการดำเนินงาน

ภาพรวมของระบบ

ระบบการปกป้องลิขสิทธิ์ของไฟล์ดิจิตอลผ่านระบบอินเตอร์เน็ตนั้นถูกออกแบบให้มีการรักษาความปลอดภัยของตัวไฟล์เองสูงและลดความเสี่ยงของการส่งผ่านระบบอินเตอร์เน็ต โดยระบบนี้จะมีการใช้เทคนิคการเข้ารหัสทั้งแบบ Symmetric และ Asymmetric ควบคู่กันอีกทั้งยังมีการใช้เทคนิคของ Network coding เข้ามาช่วยเพิ่มความปลอดภัยของข้อมูลในระหว่างการรับ-ส่งผ่านระบบอินเตอร์เน็ต



ภาพที่ 3-1 ภาพรวมของระบบ e-book online streaming

ระบบ e-book online streaming ได้ถูกออกแบบโดยที่ผู้ใช้สามารถอ่าน e-book ผ่านอินเตอร์เน็ตได้โดยไม่จำเป็นต้องดาวน์โหลดไฟล์มาเก็บไว้ในพื้นที่จัดเก็บข้อมูลบนคอมพิวเตอร์หรือสมาร์ทโฟน แต่ในระบบนี้ยังมีการปกป้องลิขสิทธิ์ (DRM) เพื่อป้องกันการละเมิดลิขสิทธิ์จากการนำไปอ่านหรือเผยแพร่โดยไม่ได้รับอนุญาต หลักการทำงานของระบบนี้เริ่มจากเมื่อมีผู้ใช้จะเปิดอ่าน e-book ตัวโปรแกรมอ่าน e-book จะทำการร้องขอไปยัง Server จากนั้น Server จะทำการส่งข้อมูลของ e-book มาทีละหน้าโดยที่ทำการเข้ารหัสข้อมูลก่อนที่จะส่งออกมายังโปรแกรมอ่าน e-book เพื่อทำการถอดรหัสและแสดงผลต่อไป จากภาพที่ 3-1 เป็นการแสดงภาพรวมของระบบ e-book online streaming ซึ่งในแต่ละหน้าที่ส่งออกมานั้นถูกเข้ารหัสด้วยเทคนิค Hybrid encryption โดยเป็นการผสมผสานระหว่างอัลกอริทึมแบบ Symmetric และ Asymmetric เพื่อเพิ่มความปลอดภัยของข้อมูลมากยิ่งขึ้น สำหรับอัลกอริทึมแบบ Symmetric ได้เลือกใช้ Advance encryption standard (AES) และ Asymmetric ใช้ Elliptic curve cryptography (ECC) อีกทั้งยังได้มีการใช้เทคนิค Secret mixing เข้ามาใช้เพื่อเพิ่มความปลอดภัยยิ่งขึ้น

กระบวนการดำเนินงาน

สำหรับการออกแบบในกระบวนการดำเนินงานนั้นจากภาพรวมของระบบหัวใจหลักของระบบนี้จะอยู่ที่เทคนิค Hybrid encryption โดยอัลกอริทึมในส่วนแรก AES ขนาด 256 บิท จะใช้สำหรับในการเข้ารหัสไฟล์ e-book เพราะมีความปลอดภัยสูง ส่วน ECC ขนาด 224 บิท ซึ่งจะเทียบเท่ากับ RSA ขนาด 2,048 บิท จะใช้สำหรับเข้ารหัสกุญแจ AES ในระหว่างการส่งกุญแจผ่านอินเตอร์เน็ตซึ่งมีชื่อว่า Key encapsulate mechanism (KEM) ในบทนี้เราได้พัฒนาระบบการรับส่งที่แตกต่างระบบ Hybrid encryption โดยทั่วไป โดยมีการเสริมเทคนิคด้าน Secret mixing เพื่อให้ขั้นตอนของ KEM นั้นปลอดภัยมากยิ่งขึ้น อีกทั้งในส่วนของการเข้ารหัสเนื้อหาข้อมูลได้มีการใช้เทคนิค Unequal security protection (USP) มาช่วยในเรื่องของการเพิ่มความเร็วให้กับระบบที่มีทรัพยากรจำกัด เช่น สมาร์ทโฟน หรือบอร์ดสมองกลฝังตัว

KEY ENCAPSULATION MECHANISM (KEM)

ระบบสำหรับการส่งกุญแจสำหรับการถอดรหัสผ่านอินเตอร์เน็ตอย่างปลอดภัย โดยทั่วไปแล้วการส่ง AES key จะมีการเข้ารหัส Key ทั้งหมดแล้วส่งออกไป ทำให้มีโอกาสถูกโจมตีโดยผู้ไม่ประสงค์ดีได้ในครั้งเดียว แต่สำหรับ KEM นี้จะทำการแบ่ง AES key ออกเป็น

หลาย ๆ ชิ้นแล้วทำการเข้ารหัสแยกกันและส่งออกไปยังผู้รับ ซึ่งอาจจะช่วยลดโอกาสการถูกโจมตีจากผู้ไม่ประสงค์ดีได้ดีขึ้น โดยมีผู้ส่งจะใช้ชื่อแทนว่า “Bob” ส่วนฝั่งผู้รับจะใช้ชื่อแทนว่า “Alice” ซึ่ง KEM นี้ได้มีการออกแบบมา 2 แบบ ดังนี้

1. Key encapsulation mechanism (KEM) Without secret mixing การทำงานของ KEM นั้นประกอบไปด้วย 3 ขั้นตอนย่อย คือ 1) Alice และ Bob สร้าง Public key และแลกเปลี่ยนกัน 2) ทำการเข้ารหัส AES key ของทางฝั่ง Bob 3) Alice ทำการถอดรหัสและนำไปใช้งานต่อไป

โดยเราสามารถอธิบายขั้นตอนการทำงานได้ดังต่อไปนี้

1.1 Public key generation of Alice and Bob, Alice และ Bob ต่างสร้าง Private key ของตัวเองขึ้นมา d_A และ d_B นำค่า d_A และ d_B มาสร้าง Public key ขึ้นมาจากค่าสุ่ม G

$$\begin{aligned} Q_A &= d_A G, \\ Q_B &= d_B G, \end{aligned} \tag{3-1}$$

Alice และ Bob แลกเปลี่ยน Public key กัน (R’s PK#1, S’s PK#1) ดังภาพที่ 3-2 โดย Public key ของ Alice จะถูกใช้ในกระบวนการ KEM ส่วนของ Bob จะใช้สำหรับสร้าง Digital signature สำหรับข้อมูลที่ทำการส่งออกไป

1.2 Key encapsulation of Bob โดย Bob สร้าง Shared secret x จากค่าสุ่ม r กับ Alice Public key Q_A

$$x, y \ = rQ_A = rGd_A = Rd_A \tag{3-2}$$

Bob ทำการสร้าง AES key K_{AES} ใช้อัลกอริทึม Elliptic curve encryption E_k ทำการเข้ารหัสโดยมี Input x และ K_{AES} ซึ่งจะได้ Output ของการเข้ารหัส AES key เป็น C_K จากนั้นจะส่งไปให้ Alice ต่อไป

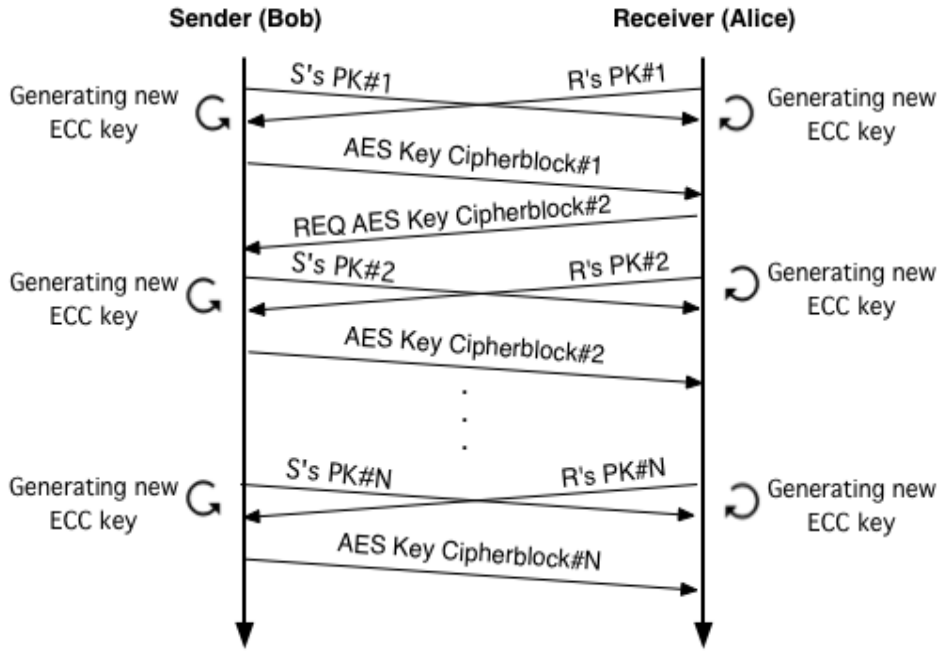
$$C_K = E_K(x, K_{AES}) \tag{3-3}$$

1.3 Key decapsulation of Alice ซึ่ง Alice กำหนดให้ Shared secret x โดยใช้ค่า R จาก Bob และ Private key d_A

$$x, y \ = Rd_A \tag{3-4}$$

ใช้ Shared secret x กับ C_K เป็น Input ให้กับการถอดรหัสด้วย Elliptic curve decryption เพื่อให้ได้ K_{AES} ต่อไป

$$K_{AES} = D_K(x, C_K) \tag{3-5}$$



ภาพที่ 3-2 Key encapsulate mechanism without secret mixing

2. Key encapsulation mechanism (KEM) With secret mixing สำหรับ KEM โดยเพิ่มความสามารถด้วยวิธี Secret mixing เข้าไปจะแตกต่างกับตัวอย่างที่แล้วบางขั้นตอน ดังนี้

2.1 Public key generation of Alice and Bob โดยให้ Alice และ Bob ต่างสร้าง Private key ของตัวเองขึ้นมา $d_{A,i}$ และ $d_{B,I}$ โดยที่ i = 1,2,..n จากนั้นนำค่า $d_{A,i}$ และ $d_{B,i}$ มาสร้าง Public key ขึ้นมาจากค่าสุ่ม G

$$Q_{A,i} = d_{A,i}G, \\ Q_{B,i} = d_{B,i}G, \tag{3-6}$$

Alice และ Bob แลกเปลี่ยน Public key กัน (R's PK#11, S's PK#11) ดังภาพที่ 3-2 โดย Public key ของ Alice จะถูกใช้ในกระบวนการ KEM ส่วนของ Bob จะใช้สำหรับสร้าง Digital signature สำหรับข้อมูลที่ทำการส่งออกไป

2.2 Key encapsulation of Bob โดยให้ Bob สร้าง Shared secret x_i , i = 1, 2, … , n จากค่าสุ่ม r_i กับ Alice Public key $Q_{A,i}$

$$x_i, y_i \ = r_i Q_{A,i} = r_i G d_{A,i} = R_i d_{A,i} \tag{3-7}$$

Bob ทำการสร้าง AES key K_{AES} ทำการเป็น AES key เป็นจำนวน n

$$K_{AES} = [K_{AES,1}, K_{AES,2}, \dots, K_{AES,n}] \quad (3\text{-}8)$$

นำ AES key มาผสมรวมกันกับค่าคงที่ซึ่งจะได้ค่า secret m_i, i = 1,2,..,n

$$m_i = a_{i,1}K_{AES,1} + a_{i,2}K_{AES,2} + \cdots + a_{i,n}K_{AES,n} \quad (3\text{-}9)$$

ใช้อัลกอริทึม Elliptic curve encryption E_k ทำการเข้ารหัสโดยมี Input x_i และ m_i ซึ่งจะได้ Output ของการเข้ารหัส AES key เป็น $C_{K,i}$ จากนั้นจะส่งไปให้ Alice ต่อไป

$$C_{K,i} = E_K(x_i, m_i) \quad (3\text{-}10)$$

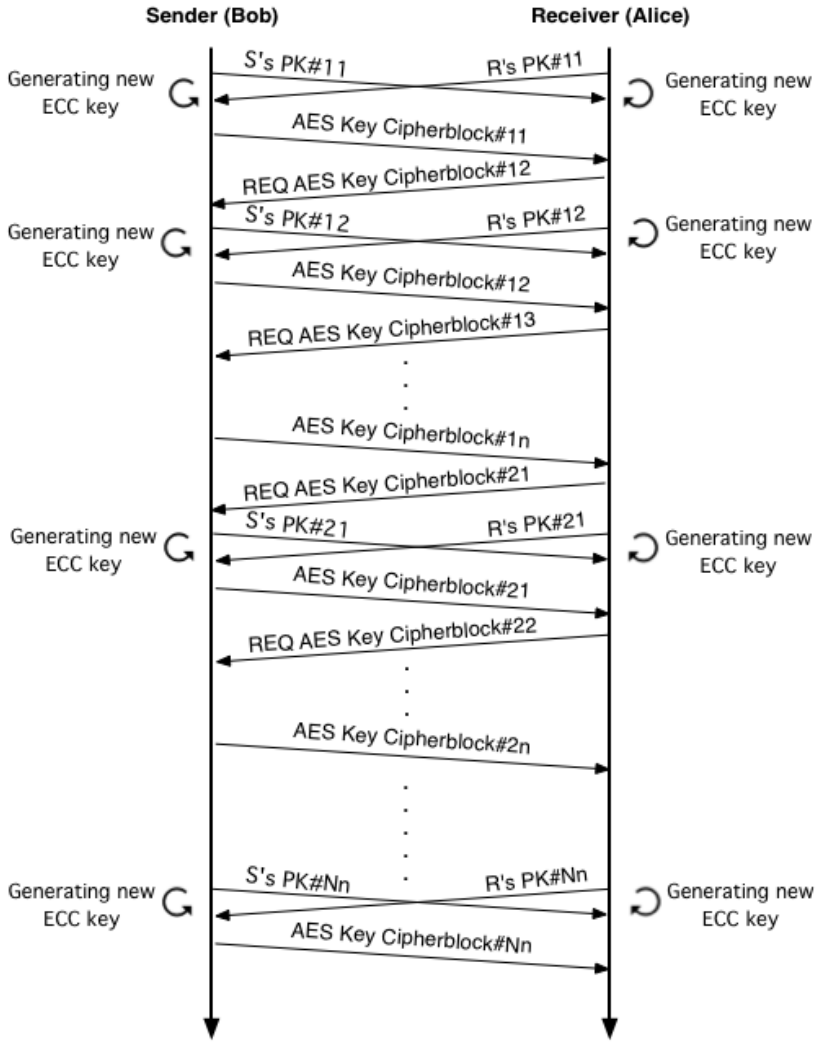
2.3 Key Decapsulation of Alice โดยให้ Alice กำหนดให้ Shared secret x_i , i = 1,2,…,n โดยใช้ค่า R_i จาก Bob และ Private key $d_{A,i}$

$$x_i, y_i = R_i d_{A,i} \quad (3\text{-}11)$$

ใช้ Shared secret x_i กับ $C_{K,i}$ เป็น Input ให้กับการถอดรหัสด้วย Elliptic curve decryption เพื่อให้ได้ Mixed secret m_i ต่อไป

$$m_i = D_k(X_i, C_{K,i}) \quad (3\text{-}12)$$

นำค่า m_i ที่ได้มาหาค่า $K_{AES,\,i}$ โดยหาได้จาก $a_{i,k}$ ที่กำหนดไว้ จากนั้นนำ $K_{AES,\,I}$ มารวมเพื่อให้ได้ K_{AES} และนำไปใช้งานต่อไป

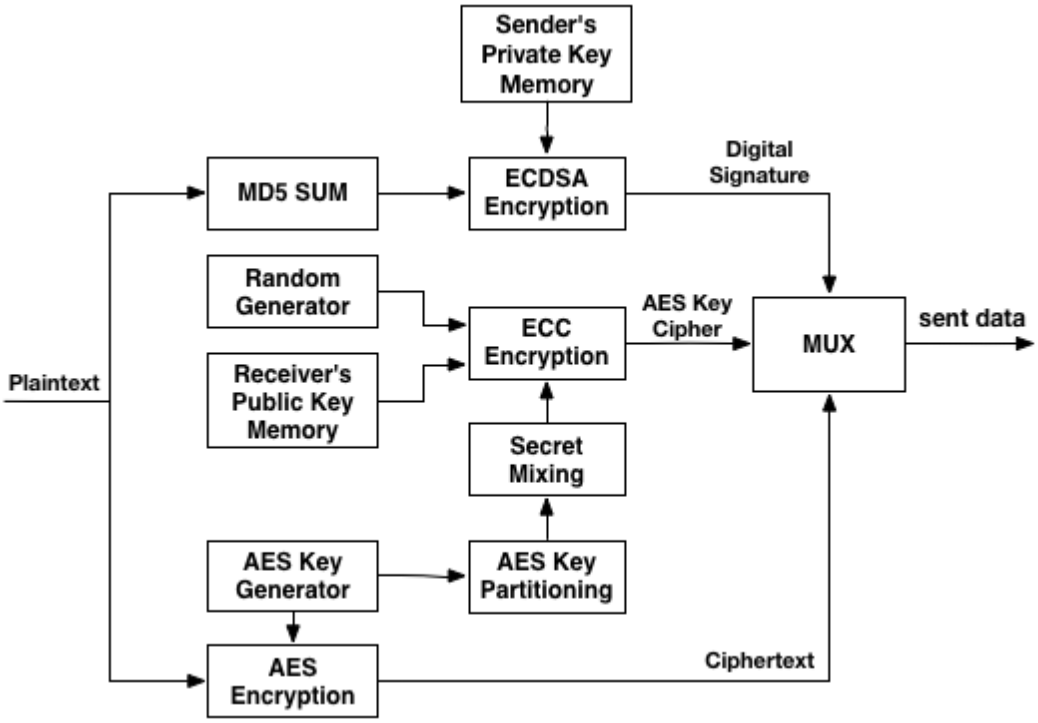


ภาพที่ 3-3 Key encapsulation mechanism with secret mixing

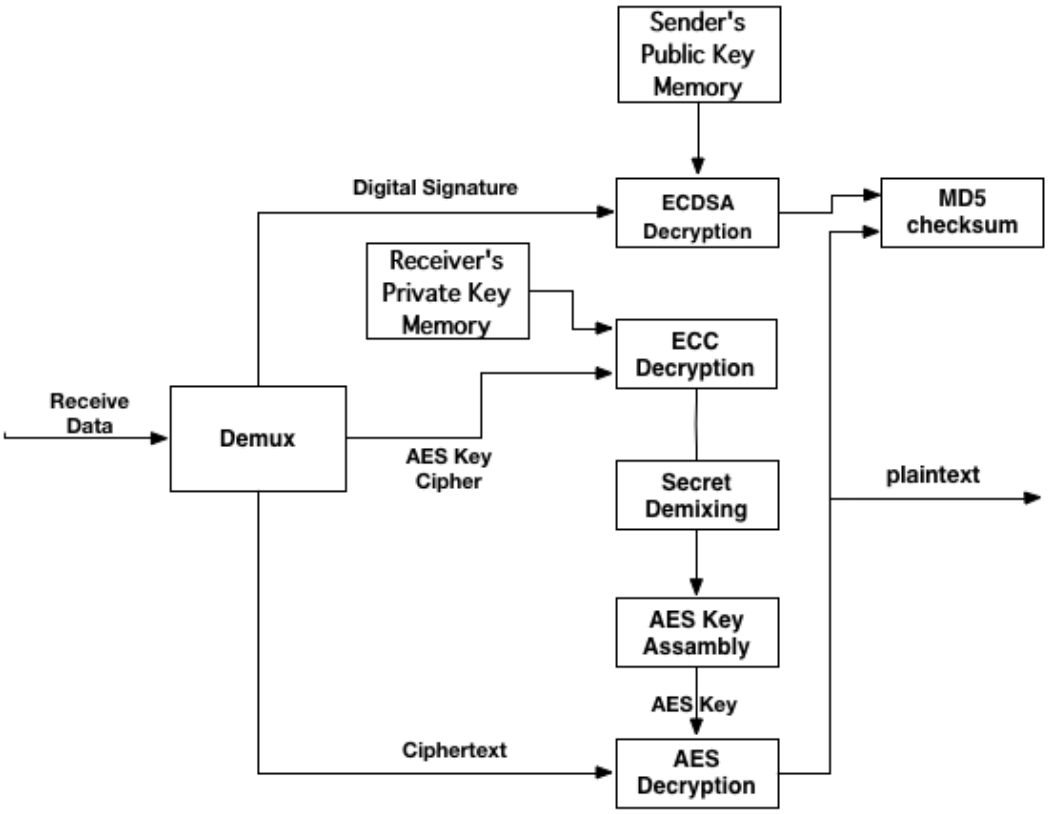
ในการส่ง AES key โดยใช้วิธี KEM นั้นขั้นแรกของกระบวนการจะทำการแยก AES key ออกเป็นหลาย ๆ ชิ้นทางต้องการจากนั้นจังหวะที่มีการร้องขอ AES key จากทางฝั่ง Alice ทั้ง Bob และ Alice จะสร้าง Public key ขึ้นมาและทำการแลกเปลี่ยนกัน จากนั้น Bob จะทำการเข้ารหัส AES key ชิ้นที่ 1 และสร้าง Digital signature จาก Public key ของ Alice และส่ง AES key ที่เข้ารหัสแล้วชิ้นที่ 1 พร้อมกับ Digital signature ไปให้ Alice เมื่อ Alice ได้รับข้อมูลแล้วและทำการตรวจสอบข้อมูลที่ได้นั้นถูกต้อง Alice จะทำการร้องขอ AES key ชิ้นต่อไปจาก Bob ซึ่งจะมีการทำเช่นนี้ไปจนกว่าจะได้รับ AES key ชิ้นสุดท้ายเป็นที่เรียบร้อยแล้วจึงจะเสร็จสิ้นกระบวนการของ KEM

ขั้นตอนการพัฒนาของระบบ

ขั้นตอนการพัฒนาระบบนั้นสามารถแบ่งออกได้เป็น 2 ส่วนหลัก ดังนี้



ภาพที่ 3-4 ขั้นตอนการส่งข้อมูล



ภาพที่ 3-5 ขั้นตอนการรับข้อมูล

จากภาพที่ 3-4 และ 3-5 จะอธิบายถึงขั้นตอนในการรับ-ส่งข้อมูล ซึ่งสามารถอธิบายได้ดังต่อไปนี้

1. การส่งข้อมูล

1.1 นำข้อมูลที่ต้องการส่ง (Plain text) มาทำการเข้ารหัสด้วยวิธี MD5 เพื่อนำค่าที่ได้ไปใช้ในขั้นตอนการสร้าง Digital signature ต่อไป

1.2 ทำการสร้าง AES key ขึ้นมาและนำ key ไปใช้เข้ารหัสข้อมูลที่ต้องการจะส่งออกไปยังฝั่งผู้รับ

1.3 นำ AES key ที่ได้ไปแยกออกเป็นหลาย ๆ ชิ้นด้วยและนำไปเข้ากระบวนการ Secret mixing

1.4 นำ ญublic key ของผู้รับกับค่า Random ค่าหนึ่งนำไปใช้ในการเข้ารหัสชิ้นส่วนของ AES key ด้วยเทคนิคของ ECC

1.5 นำข้อมูลที่ได้เข้ารหัสแล้ว (Cipher text) ชิ้นส่วน AES key ที่ถูกเข้ารหัส (AES key cipher และ Digital signature ส่งไปยังผู้รับต่อไป

2. การรับข้อมูล เมื่อได้รับข้อมูลมาจะถูกนำมาแยกออกเป็น 3 ส่วน คือ Digital signature, AES key cipher และ Cipher text

2.1 นำ Digital signature มาถอดรหัสด้วย Public key ของผู้ส่งด้วยวิธี ECDSA เพื่อให้ได้ค่า MD5 ของข้อมูลต้นฉบับ

2.2 นำ AES key cipher มาถอดรหัสด้วย Private key ของผู้รับด้วยวิธี ECC และทำกระบวนการ Secret demixing เพื่อให้ได้ชิ้นส่วนต่าง ๆ ของ AES key และนำมารวมกันเป็น AES key ที่พร้อมจะใช้งานได้

2.3 นำ Cipher text มาถอดรหัสด้วย AES key ที่ได้มา จากนั้นนำไปตรวจสอบค่า MD5 ที่ได้รับมาว่าถูกต้องไม่ ถ้าถูกต้องสามารถจะนำข้อมูลไปใช้งานได้ต่อไป

SECRET MIXING AND DEMIXING

ในกระบวนการทำงานของระบบในส่วนของการใช้ Secret mixing สำหรับ AES key ที่ถูกแบ่งออกเป็นส่วน ๆ ด้วยการใช้กรรมวิธีทางคณิตศาสตร์ เพื่อทำการแก้ไขสมการการกระจายข้อมูล เข้ามาใช้เพื่อเพิ่มความปลอดภัยนั้น สาเหตุที่ใช้ Secret mixing ในขั้นตอนนี้เพราะถ้าในระหว่างการส่ง AES key ถูกโจมตีโดยผู้ไม่ประสงค์ดีซึ่งอาจจะรวบรวมและประกอบเป็น AES key ที่สมบูรณ์ได้แต่ถ้าเพิ่ม Secret mixing เข้าไปถึงแม้จะสามารถรวบรวมชิ้นส่วนของ AES key และถอดรหัสออกมาได้แต่ก็จะยังไม่ได้ชิ้นส่วนของข้อมูลจริงเพราะต้องนำชิ้นส่วนเหล่านั้นมาแก้สมการของ Secret mixing เพื่อที่จะได้ผลลัพธ์ออกมาเป็น AES key ที่สมบูรณ์ต่อไป โดยจะทำการทดลองออกเป็น 2 ตัวอย่าง ดังนี้

1. Secret mixing ให้ n = 2 จะได้สมการดังต่อไปนี้

$$\begin{aligned} m_1 &= K_{AES,1} + K_{AES,2} \\ m_2 &= K_{AES,1} + 2K_{AES,2} \end{aligned} \tag{3-13}$$

2. Secret mixing ให้ n = 4 จะได้สมการดังต่อไปนี้

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 8 \\ 1 & 5 & 8 & 15 \end{bmatrix} \cdot \begin{bmatrix} K_{AES,1} \\ K_{AES,2} \\ K_{AES,3} \\ K_{AES,4} \end{bmatrix} \tag{3-14}$$

UNEQUAL SECURITY PROTECTION

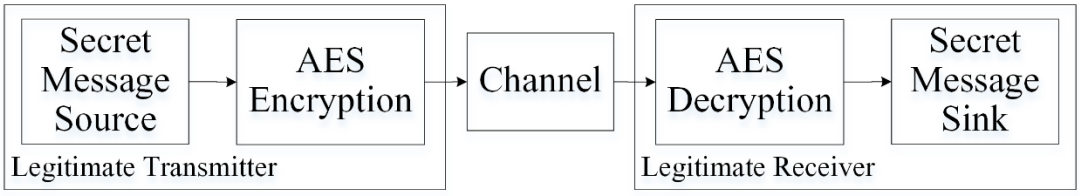
Unequal security protection (USP) นั้นออกแบบมาสำหรับการรับส่งข้อมูลที่เป็นความลับหรือต้องการความปลอดภัย แต่โดยส่วนใหญ่แล้วการเข้ารหัสข้อมูลที่มีความปลอดภัยสูงมาก ๆ นั้น จะใช้เวลาค่อนข้างมาก ทำให้การรับส่งข้อมูลทำได้ช้าลง แต่ถ้าข้อมูลที่ส่งต้องการความปลอดภัยพอประมาณแต่เน้นที่เวลาในการรับส่งที่น้อยลง ซึ่งกระบวนการนี้จะใช้ USP เข้ามาใช้งานโดยมีการทำงานดังต่อไปนี้

1. แบ่งจำนวนข้อมูลออกเป็นจำนวน n
2. ข้อมูลที่แบ่งออกนั้นจะนำไปเข้ารหัสกันแบบขนาน โดยที่มีอัลกอริทึมการเข้ารหัสที่ต่างกัน

USP SYSTEM DESIGN

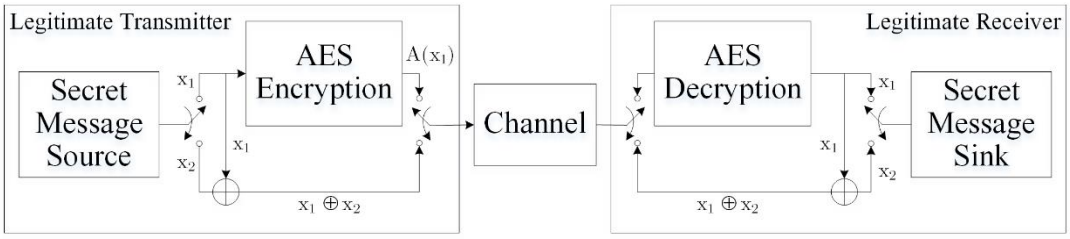
ในการออกแบบ USP นี้ซึ่งออกแบบไว้ด้วยกันอยู่ 4 แบบ ดังต่อไปนี้

1. Pure AES โดยจะทำการส่งข้อมูลโดยเข้ารหัสแบบ AES อย่างเดียวดังภาพที่ 3-6



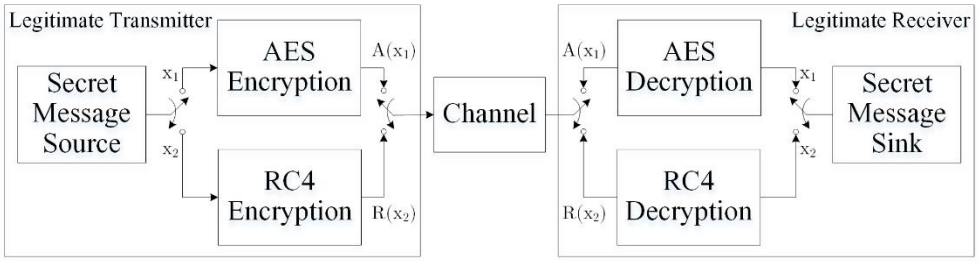
ภาพที่ 3-6 System model of pure AES encryption in baseband

2. One-Half AES encryption with secret mixing (AES/XOR) โดยจะทำการแยกข้อมูลออกเป็น 2 ทาง โดย x_1 เป็นข้อมูลที่นำไปเข้ารหัสด้วย AES ผลลัพธ์คือ $A(x_1)$ ส่วนอีกทาง x_2 เป็นข้อมูลที่จะนำไป XOR กับ x_1 ได้ผลลัพธ์เป็น $x_1 \oplus x_2$ ดังภาพที่ 3-7



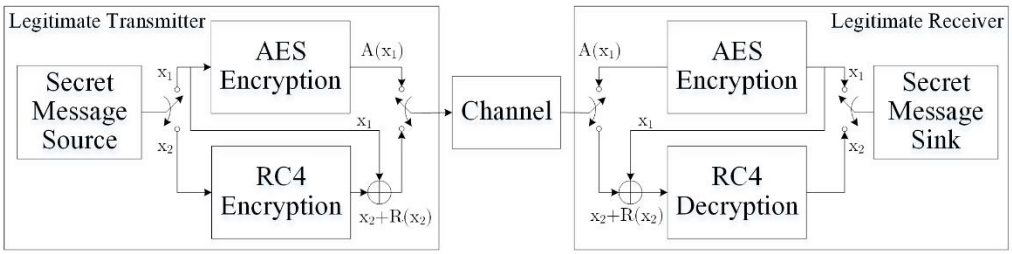
ภาพที่ 3-7 System model One-Half AES encryption with secret mixing in baseband

3. One-Half AES encryption and One-Half RC4 encryption (AES/RC4) โดยจะทำการแยกข้อมูลออกเป็น 2 ทาง โดย x_1 เป็นข้อมูลที่นำไปเข้ารหัสด้วย AES ผลลัพธ์คือ $A(x_1)$ ส่วนอีกทาง x_2 เป็นข้อมูลที่จะนำไปเข้ารหัสด้วย RC4 ได้ผลลัพธ์เป็น $R(x_2)$ ดังภาพที่ 3-8



ภาพที่ 3-8 System model One-Half AES encryption and One-Half RC4 encryption in baseband

4. One-Half AES encryption and One-Half RC4 encryption with secret mixing โดยจะทำการแยกข้อมูลออกเป็น 2 ทาง โดย x_1 เป็นข้อมูลที่นำไปเข้ารหัสด้วย AES ผลลัพธ์คือ $A(x_1)$ ส่วนอีกทาง x_2 เป็นข้อมูลที่จะนำไปเข้ารหัสด้วย RC4 หลังจากนั้นนำข้อมูลไปผ่านกระบวนการ Secret mixing ได้ผลลัพธ์เป็น $x_2 + R(x_2)$ ดังภาพที่ 3-9



ภาพที่ 3-9 System model One-Half AES encryption and One-Half RC4 encryption with secret mixing in baseband

การเข้ารหัสข้อมูลด้วย AES เพียงอย่างเดียวเทียบกับการเข้ารหัสด้วยเทคนิค USP นั้น โดยเทคนิค USP จะให้ผลในเรื่องความเร็วในการเข้ารหัสที่ดีกว่าการเข้ารหัสด้วย AES เพียงอย่างเดียวแต่ความปลอดภัยของข้อมูลอาจจะมีการลดทอนลงบ้างโดยรูปแบบของ USP ตั้งแต่แบบ 2 ถึง 4 จะเป็นการแบ่งข้อมูลออกเป็น 2 ส่วนโดยจะมีการเข้ารหัส AES เพียงฝั่งเดียว ส่วนอีกฝั่งจะเป็นเข้ารหัสอีกชนิดหนึ่งซึ่งในแบบที่ 2 จะเป็นการทำ XOR ส่วนแบบที่ 3 และ 4 จะเป็นการเข้ารหัสด้วยอัลกอริทึม RC4 และ RC4 + Secret mixing ตามลำดับ ซึ่งในแต่ละแบบจะส่งผลถึงเรื่องระดับความปลอดภัย ความเร็วและทรัพยากรของระบบที่ถูกใช้ โดยในเรื่องความปลอดภัยแบบที่ 4 จะให้ความปลอดภัยที่ดีกว่าแบบที่ 3 และ 2 ตามลำดับ ส่วนในเรื่องของความเร็วและทรัพยากรของระบบที่ถูกใช้นั้น แบบที่ 2 จะดีที่สุดตามมาด้วยแบบที่ 3 และ 4 ตามลำดับโดยหลักการทำงานของ USP นั้น ถึงแม้จะถูกผู้ไม่ประสงค์ดีดักฟังข้อมูลไปได้แต่เนื่องจากมีการผสมกันระหว่างการเข้ารหัสข้อมูลหลายชนิดอีกทั้งยังมีการผสมข้อมูลเข้าไปโดยเทคนิค Secret mixing ทำให้การถอดรหัสข้อมูลแล้วได้ข้อมูลที่ถูกต้องนั้นยากมากยิ่งขึ้น ซึ่งในบทต่อไปจะทำการทดลองเพื่อหาประสิทธิภาพการทำงานของเทคนิค Key encapsulation mechanism (KEM) และ Unequal security protection (USP)

บทที่ 4
ผลการวิจัย

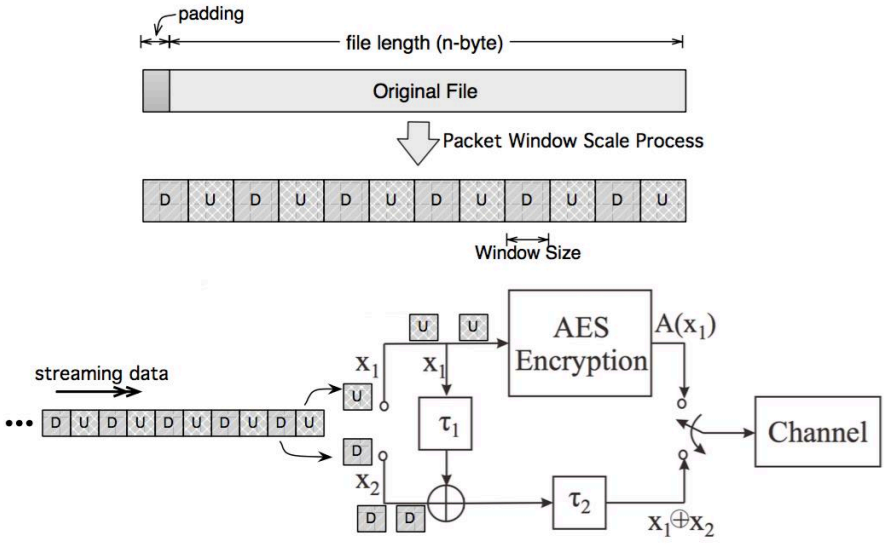
เครื่องมือที่ใช้ในการทดลอง

การทดลองนี้ได้ทำการทดลองด้วยการเขียนโปรแกรมด้วยภาษา Java และทำการรันโปรแกรมบนคอมพิวเตอร์ที่มีหน่วยประมวลผลลาง Intel Core i7 quad-core 2.3GHz หน่วยความจำ 10 GB ระบบปฏิบัติการ OSX 64bit

วิธีการทดลอง

การทดลองของ Hybrid encryption จะประกอบไปด้วย 2 เทคนิค คือ Key encapsulation mechanism (KEM) และ Unequal security protection (USP)

1. การทดลองเรื่อง Unequal security protection (USP) ในการส่งข้อมูลที่มีการรักษาความปลอดภัยผ่านระบบเครือข่ายนั้น ขั้นแรกจะทำการแบ่งข้อมูลออกเป็น Packet ขนาดเล็กด้วยเทคนิค Packet window scale encryption ดังภาพที่ 4-1



ภาพที่ 4-1 Packet windows scale encryption

จากภาพที่ 4-1 เป็น USP ในแบบที่ 2 คือ ใช้ AES + XOR ส่วนในแบบที่ 3 และ 4 เราได้ทำการทดลองแบบเดียวกันโดยจะเป็นการใช้ AES + RC4 และ AES + RC4 + Secret mixing ตามลำดับซึ่งในทุก Packet จะถูกเข้าคิวเพื่อเรียงลำดับการผ่านกระบวนการเข้ารหัส โดย Window size จะมีขนาดเริ่มต้นที่ 16 byte และเพิ่มขนาดไปจนถึงสูงสุด 1,600 byte (100 เท่า) ประโยชน์ที่ได้จากการใช้เทคนิค Packet window scale encryption นี้เป็นการทำให้การเข้ารหัส Packet ที่ส่งออกนั้นเหมาะสมกับ Bandwidth ของเครือข่ายนั้น ๆ

2. การทดลองเรื่อง Key encapsulation mechanism (KEM) ในการทดลองของ Hybrid encryption นั้นจะถูกนำมาใช้ในส่วนของการทำ Key encapsulation mechanism (KEM) ซึ่งได้แบ่งการทดลองออก 2 แบบ คือ

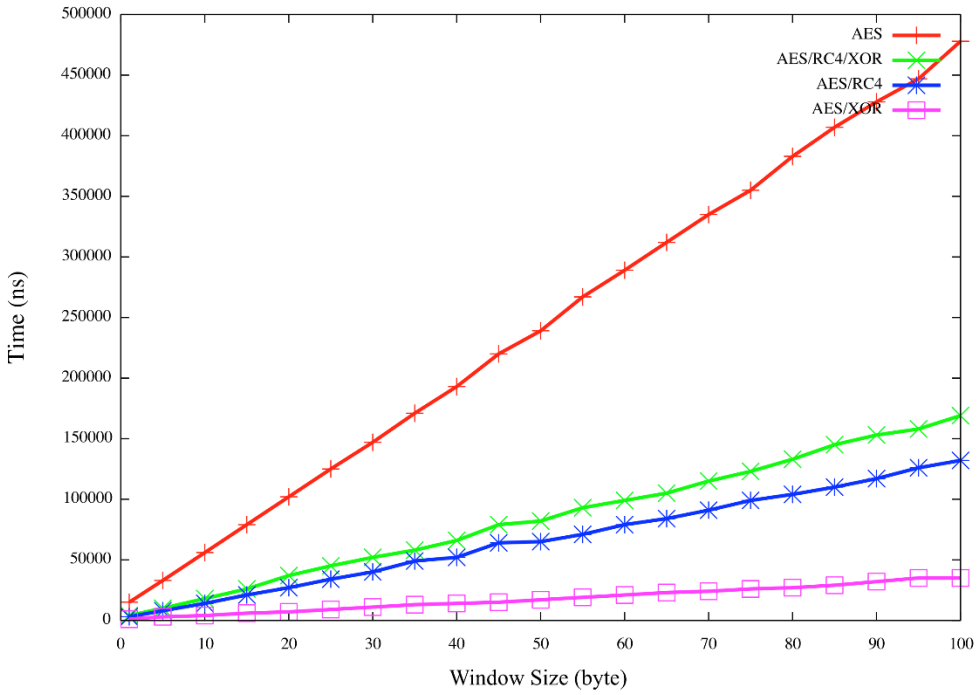
2.2 Key encapsulation mechanism

2.3 Key encapsulation mechanism + Secret mixing

โดยจะทำการหาประสิทธิภาพการทำงานโดยวัดจากการทำงานของ CPU ซึ่งการทดลองนั้นจะใช้ KEM ในการเข้ารหัส AES key โดยจะทำการแบ่ง Key ออกเป็น 2 และ 4 ชิ้น ตามลำดับ โดยในแบบแรกนั้น Key แต่ละชิ้นจะถูกเข้ารหัสแบบ ECC ก่อนทำการส่ง ส่วนในแบบที่สองมีการทำ Secret mixing เพื่อผสม Key แต่ละชิ้นเข้าด้วยกันก่อนจะเข้ารหัสโดย ECC แล้วทำการส่ง ซึ่งกระบวนการทำงานของ KEM ทั้ง 2 แบบนั้น ได้อธิบายรายละเอียดไว้ในบทที่ 3 ในหัวข้อ "Key encapsulation mechanism (KEM)"

ผลการทดลอง

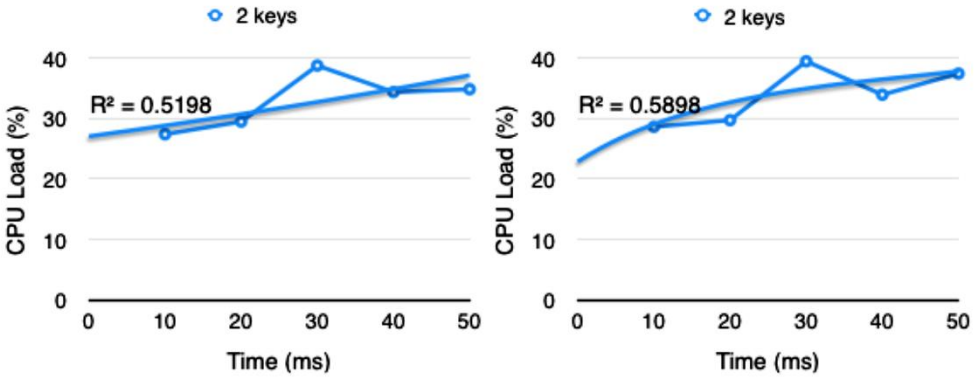
ผลลัพธ์จากการทดลองของเทคนิค USP โดยการวัดเวลาที่ใช้ในแต่ละกระบวนการแสดงดังภาพที่ 4-2



ภาพที่ 4-2 Computation time of USP encryption process

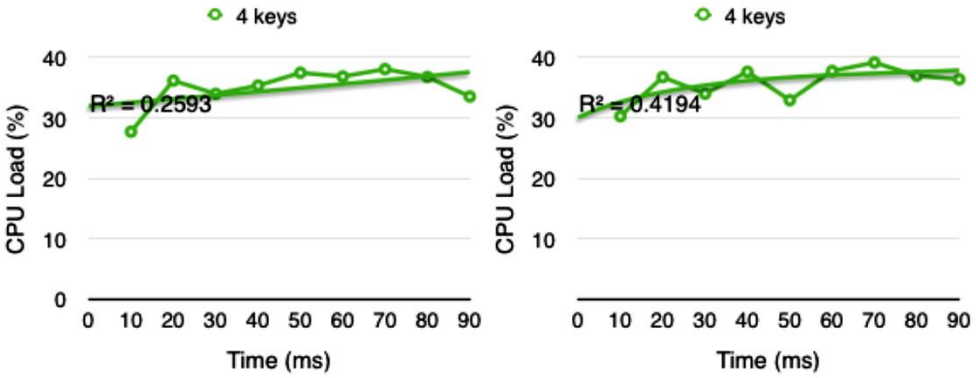
จากกราฟผลการทดลองนั้นแสดงให้เห็นถึงเวลาใช้ในการเข้ารหัสจะแปรผันตรงกับขนาดของ Window size ส่วนของการถอดรหัสก็เวลาเช่นเดียวกันกับการเข้ารหัส การเข้ารหัสด้วย AES อย่างเดียวเมื่อเปรียบเทียบวิธีของ USP นั้นจะเห็นได้ว่าใช้เวลาต่างกันมากกว่าครึ่ง แต่วิธีของ USP นั้นออกแบบมาสำหรับ Security ที่แตกต่างกันอยู่ 3 แบบ โดยในแต่ละแบบก็จะมีผลต่อเวลาที่ใช้ในงานทำงานซึ่งอยู่กับความต้องการของระบบ

ส่วนผลการทำงานของ CPU ของการทำ KEM ที่จำนวนชิ้นส่วนกุญแจ 2 และ 4 ชิ้น จะแสดงดังภาพที่ 4-3 และ 4-4 ตามลำดับ โดยภาพที่ 4-3 (a) และ 4-4 (a) เป็นผลการทำงานโดยไม่มี Secret mixing ส่วนภาพที่ 4-3 (b) และ 4-4 (b) เป็นการทำงานที่มี Secret mixing



(a) (b)

ภาพที่ 4-3 ผลลัพธ์สำหรับการแบ่ง Key เป็น 2 ชิ้น (n=2) (a) และแบบมี Secret mixing (b)



(a) (b)

ภาพที่ 4-4 ผลลัพธ์สำหรับการแบ่ง key เป็น 4 ชิ้น (n=4) (a) และแบบมี Secret mixing (b)

ค่า R^2 เป็นค่า Coefficient of determination ซึ่งเกี่ยวข้องกับความแตกต่างระหว่าง Curve fitting กับข้อมูลจริงกล่าว คือ ถ้า Curve fitting มีความใกล้เคียงกับข้อมูลจริง ค่า R^2 จะมีค่าเข้าใกล้ 1 ซึ่งจากกราฟการทดลองของ KEM นั้นจะเห็นได้ว่าการเข้ารหัสโดยแบ่ง AES key ออกเป็น 2 และ 4 ส่วนนั้นผลการทำงานของ CPU นั้นไม่ได้มีความแตกต่างกันอย่างเห็นได้ชัดและรวมถึงการใช้ Secret mixing เข้ามาช่วยในการผสมข้อมูลก็ยังไม่ได้ทำให้การทำงานของ CPU นั้นมากขึ้นเท่าใดนัก

ดังนั้นเราสามารถเพิ่มเทคนิค Secret mixing เข้าไปเพื่อเพิ่มความปลอดภัยให้กับการทำงานของ KEM โดยที่ไม่ได้กระทบกับการทำงานของ CPU

บทที่ 5
สรุปผล อภิปรายผล และข้อเสนอแนะ

สรุปผลการศึกษา

จากศึกษาการทำงานของ Hybrid encryption โดยผสมผสานกับวิธี Secret mixing ในส่วนของ KEM ใช้สำหรับการปกป้องข้อมูลที่มีการรับส่งกันผ่านระบบอินเตอร์เน็ต พบว่าการทำงานของ Hybrid encryption จะช่วยเพิ่มความปลอดภัยให้กับข้อมูลอีกทั้งยังช่วยให้การจัดการกับกุญแจที่ใช้ในการถอดรหัสข้อมูลให้ปลอดภัยมากยิ่งขึ้น อีกทั้งเมื่อมีการใช้วิธี Secret mixing ผสมเข้าไปด้วยในส่วนของ KEM ทำให้การปกป้องข้อมูลสูงยิ่งขึ้นไปอีก เนื่องจากมีการผสมข้อมูลบางอย่างเข้าด้วยกันทำให้โอกาสในการถอดรหัสนั้นลดลง และการเพิ่ม Secret mixing ลงไปมีผลกระทบต่อการทำงานของ CPU ไม่มากเมื่อเทียบกับการปกป้องที่ดีขึ้น

จากตารางที่ 5-1 เมื่อนำเอาทั้งเทคนิคของ Hybrid encryption ที่มี KEM แบบปกติ (Normal KEM) ในแบบ A และ KEM ที่มี Secret mixing ในแบบ B มาประกอบกับการเข้ารหัสในส่วน Symmetric key encryption แบบปกติ (Normal AES) หรือแบบ C และการเข้ารหัสที่ใช้ USP ที่มีการ Secret mixing ดังการทดลองในภาพที่ 4-1 (AES with secret mixing) ในแบบ D เราจะสามารถสร้างโมเดลรวมของ Hybrid encryption ได้ 4 แบบดังตารางที่ 5-1 ได้แก่ Traditional hybrid encryption (Hyb) ที่ประกอบด้วย A และ C ตามด้วย Hybrid encryption with secret mixing of key (HybMixK) ที่ประกอบด้วย B และ C ต่อด้วย Hybrid encryption with secret mixing of message (HybMixM) ที่ประกอบด้วย A และ D และปิดท้ายด้วย Hybrid encryption with secret mixing of key and message (HybMixKM) ที่ประกอบด้วย B และ D ซึ่งเราสามารถเปรียบเทียบประสิทธิภาพของทั้ง 4 โมเดลในแง่ของระดับความปลอดภัย (Security level) ความซับซ้อนในการคำนวณ (Computational complexity) ตลอดจนอัตราการส่งข้อมูลที่ต้องการ (Data rate) ได้ดังตารางที่ 5-2

ตารางที่ 5-1 รูปแบบของการทำ Hybrid encryption

Model	KEM	Symmetric Key Encryption
1. Traditional Hyb. Enc (Hyb)	Normal KEM (A)	Normal AES (C)
2. Hyb. Enc. with Secr. Mix. Of Key (HybMixK)	KEM with Secr. Mix. (B)	Normal AES (C)

ตารางที่ 5-1 (ต่อ)

Model	KEM	Symmetric Key Encryption
3. Hyb. Enc. with Secr. Mix. of Message (HybMixM)	Normal KEM (A)	AES with Secr. Mix. (D)
4. Hyb. Enc. with Secr. Mix. of Key and Message (HybMixKM)	KEM with Secr. Mix. (B)	AES with Secr. Mix. (D)

ตารางที่ 5-2 การเปรียบเทียบรูปแบบของ Hybrid encryption ในแต่แง่มุม

Aspects	Comparison
1. Security level	HybMixK ≈ HybMixKM > Hyb ≈ HybMixM
2. Comput. Comp.	HybMixM ≈ HybMixKM < Hyb ≈ HybMixK
3. Data rate	Hyb = HybMixM < HybMixK = HybMixKM

ข้อเสนอแนะแนวทางในอนาคต

จากการทดลองด้านประสิทธิภาพนั้นพบว่าการใช้การเข้ารหัสแบบ Hybrid โดยมีการผสมผสานกับการใช้ Secret mixing ด้วยนั้นให้ผลดีในเรื่องของความเร็วในการเข้ารหัสและการถอดรหัสที่เพิ่มขึ้นกับความปลอดภัยในระดับที่ต้องการได้ ซึ่งทำให้สามารถสร้างโมเดลสำหรับการเข้ารหัสและถอดรหัสแบบ Hybrid ได้ออกมา 4 แบบ แต่เนื่องจากยังไม่ได้มีการทดลองกับอุปกรณ์โมบายหรือบอร์ดสมองกลฝังตัวที่หลากหลายเพื่อผลการทดสอบที่สภาพแวดล้อมบนอุปกรณ์จริงจะได้ถูกต้องและแม่นยำขึ้นซึ่งอาจจะทำให้สามารถสร้างโมเดลการเข้ารหัสแบบ Hybrid ที่หลากหลายขึ้นเพื่อให้เหมาะสมกับอุปกรณ์โมบายหรือบอร์ดสมองกลฝังตัวแต่ละชนิดได้

บรรณานุกรม

Abdul, D. S., Elminaam, Abdul Kader, H. M., & Hadhoud, M. M. (2009). Performance Evaluation of Symmetric Encryption Algorithms. *Communications of the IBIMA* Volume 8, 2009 ISSN: 1943-7765.

Ahlswede, R., Cai, N., Li, S.-Y.R., & Yeung, R.W. (2000). Network Information Flow. *IEEE Trans. Inform. Theory, 46*(4), 1204-1216.

Akash Kumar Mandal, Chandra Parakash, Archana Tiwari. (2012). Performance Evaluation of Cryptographic Algorithms: DES and AES SCEECS.

Alkady, Y., Habib, M. I., & Rizk, R Y. (2013). A New Security Protocol Using Hybrid Cryptography Algorithms. *Computer Engineering Conference (ICENCO), IEEE*, 109-115.

Bhattad, K. & Narayanan, K.R. (2005). Weakly Secure Network Coding. *in Proc. NETCOD.*

Biryukov, A., Khovratovich, D., & Nikoli'c. DistinguisherandRelated-Key Attack on the Full AES-256.

Cai, N. & Yeung, R.W. (2002). Secure Network Coding. *Int. Symp. Informa- tion Theory*.

Cheng, H. & Li, X. (2000). Partial Encryption of Compressed Images and Videos. *IEEE Trans. Signal Processing, 48*(8), 2439-2451.

Cramer, R. & Shoup V. (2003). Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, *33*(1), 167-226.

Cramer, R. & Shoup, V. (1998). A Practical Public Key Cryptosystem Prov- ably Secure against Adaptive Chosen Ciphertext Attack. *Advances in Cryptology-CRYPTO'98, Springer*, 13-25.

El-Hadidi, M. T., Hegazi, N. H., Aslan, H. K. (1995). Implementation of a Hy- brid Encryption Scheme for Ethernet. *Computers and Communications. IEEE Symposium,* 150-156.

FIPS 197. (2001). Advanced Encryption Standard (AES).

Fluhrer, S., Mantin, I., & Shamir, A. (2001). Weakness in the Key Scheduling Algorithm of RC4, *Selected Areas in Cryptography Lecture Notes in Computer Science, 2259*, 1-24.

Hu, X. & Ma, L. (2010). A Study on the hybrid encryption technology in the security transmission of electronic documents. *Information Science and Management Engineering (ISME)*, *IEEE*, 60-63.

Massey J.L. (1988). AnIntroductiontoContemporaryCryptology. *Proc.IEEE*, *76*(5), 533-549.

National Security Agency. (2009). The Case for Elliptic Curve Cryptography.

Ren, M. & Miao Z. (2010). A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication. *Modeling, Simulation and Visualization Methods (WMSVM)*, *IEEE*, 221-225.

Shannon, C.E. (1949). CommunicationTheoryofSecrecySystems. *BellSystem Tech. J, 28,* 656 - 716.

Zhu, S.-H. (2011). Research of Hybrid Cipher Algorithm Application to Hydraulic Information Transmission Conf. on Electronics*, Communications and Control (ICECC)*