บทที่ 4

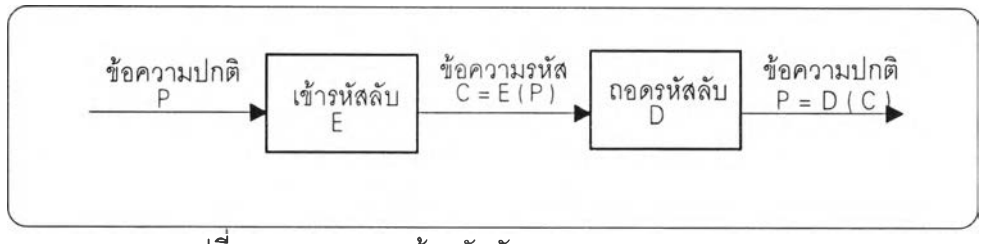
โปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูล

บทนี้จะกล่าวถึงแนวคิดเหตุผลและทฤษฎีที่ใช้ในการพัฒนาโปรแกรมอรรถประโยชน์สำหรับจัดการแฟ้มเอกสารของจุฬาจารึก เพื่อเสริมคุณสมบัติให้จุฬาจารึกทางด้านการเข้ารหัสลับแฟ้มข้อมูล นอกจากนี้ยังกล่าวถึงการออกแบบและพัฒนาโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูล บทนี้จึงแบ่งเป็น 4 ส่วน ส่วนแรกกล่าวถึงทฤษฎีพื้นฐานการเข้ารหัสลับข้อมูล ส่วนที่สองจะกล่าวถึงการเข้ารหัสลับแบบสัญนิยม ส่วนที่สามจะกล่าวถึงการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก และส่วนสุดท้ายโครงสร้างข้อมูลที่ใช้ในโปรแกรมและการพัฒนาโปรแกรมด้านการเข้ารหัสแฟ้มข้อมูลจุฬาจารึก

ทฤษฎีพื้นฐานการเข้ารหัสลับข้อมูล

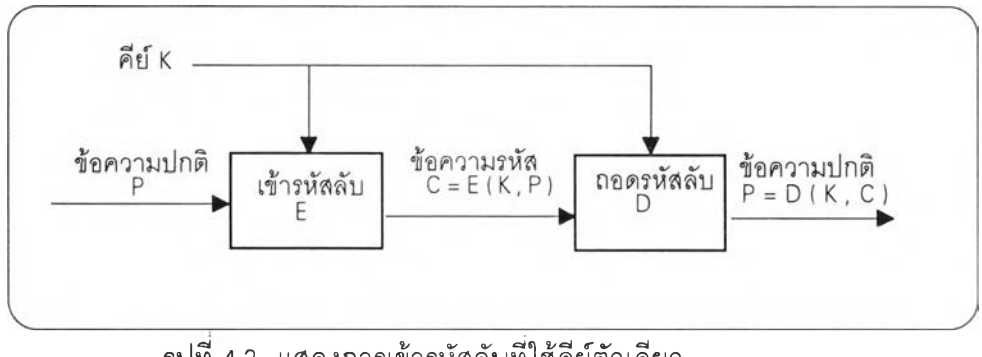
ในปัจจุบันจำเป็นต้องเพิ่มมาตราการป้องกันความปลอดภัยของข้อมูลที่สำคัญ การเข้ารหัสลับ (encryption) เป็นวิธีการหนึ่งที่สามารถป้องกันข้อมูลที่สำคัญให้เป็นความลับมิให้ผู้อื่นนำไปใช้ในทางที่ผิดได้ โดยเปลี่ยนข้อมูลที่อ่านเข้าใจได้เป็นรูปแบบอื่นที่ไม่สามารถอ่านเข้าใจก่อนที่จะเก็บข้อมูลเหล่านั้นลงในสื่อบันทึกข้อมูลหรือส่งออกไปยังที่อื่นๆผ่านเครือข่ายสื่อสาร การเข้ารหัสลับมีใช้กันมาตั้งแต่สมัยอียิปต์โบราณกว่า 4000 ปีมาแล้ว เดิมใช้วิธีสลับตำแหน่งข้อมูล (transposition cipher) หรือวิธีการแทนที่ข้อมูล (substitution cipher) วิธีสลับตำแหน่งข้อมูลจะเรียงบิตหรือตัวอักษรของข้อความเดิมในตำแหน่งใหม่จนได้ข้อความที่ไม่สามารถอ่านเข้าใจได้ เช่นข้อความเดิม "SECRET MESSAGE" จัดเรียงตำแหน่งใหม่เป็น "STSGEEEASRMSE" เป็นต้น ส่วนวิธีแทนที่ข้อมูลจะแทนที่บิตหรือตัวอักษรเดิมของข้อความเป็นบิตหรือตัวอักษรใหม่จนได้ข้อความที่ไม่สามารถอ่านเข้าใจได้ วิธีแทนที่ข้อมูลที่เป็นที่รู้จักคือ Caesar substitution จะเลื่อนตัวอักษรไปข้างหน้าเป็นจำนวน K ตำแหน่งเช่น ให้ K เป็น 3 ดังนั้น A จะถูกแทนด้วย D, B จะถูกแทนด้วย E,....., Z จะถูกแทนด้วย C นอกจากเลื่อนตัวอักษรไปข้างหน้า K ตำแหน่งแล้วยังเขียนตัวเลขแทนตัวอักษรที่เลื่อนตำแหน่งนั้นด้วย เช่น ข้อความเดิม "PRIVATE" จะถูกแทนที่เป็น "23 17 9 11 133 1" เป็นต้น (David A. Curry, 1992) ในสมัยก่อนวิธีเหล่านี้ได้รับการยอมรับว่าสามารถป้องกันข้อมูลให้ปลอดภัยได้ แต่ความเจริญทางเทคโนโลยีในปัจจุบันและการนำคอมพิวเตอร์มา

ใช้งานทำให้วิธีเหล่านี้ไม่เพียงพอที่จะป้องกันข้อมูลให้ปลอดภัยได้ จึงจำเป็นต้องศึกษาวิธีการเข้ารหัสลับข้อมูลให้มีความปลอดภัยมากขึ้น cryptography เป็นวิชาที่ศึกษาเกี่ยวกับการเข้ารหัสลับเพื่อป้องกันข้อมูลให้ปลอดภัย จะศึกษาวิธีเข้ารหัสลับ (encryption) และวิธีถอดรหัสลับ (decryption) วิธีเข้ารหัสลับเป็นกระบวนการเปลี่ยนข้อความปกติ (plaintext) ที่อ่านเข้าใจได้ให้เป็นข้อความรหัส (ciphertext) ที่ไม่สามารถอ่านเข้าใจได้ ส่วนวิธีถอดรหัสลับจะเป็นกระบวนการย้อนกลับจะเปลี่ยนจากข้อความรหัสกลับเป็นข้อความปกติตามเดิม การเข้ารหัสลับแสดงดังรูปที่ 4.1

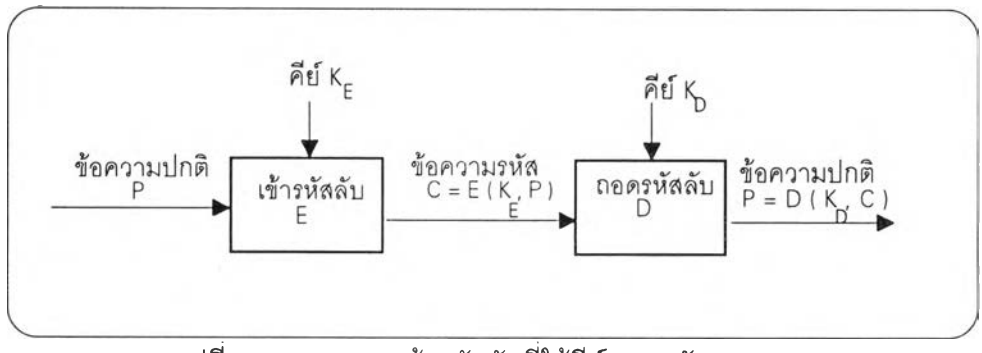


รูปที่ 4.1 แสดงการเข้ารหัสลับ

การเข้ารหัสลับจะใช้คีย์ (key) k ควบคุมวิธีเข้ารหัสลับและวิธีถอดรหัสลับ คีย์เป็นข้อมูลที่เก็บเป็นความลับเพื่อใช้ในวิธีเข้า/ถอดรหัสลับ ข้อความปกติอันหนึ่งนำมาเข้ารหัสลับด้วยคีย์ที่ต่างกันจะได้ข้อความรหัสที่ต่างกันไปด้วย ทำนองเดียวกันข้อความรหัสนำมาถอดรหัสลับด้วยคีย์ที่ต่างกันจะได้ข้อความที่ต่างกันไป ข้อความรหัสจะต้องถอดรหัสลับด้วยคีย์ที่ถูกต้องเท่านั้นจึงจะได้ข้อความปกติที่ถูกต้องตามเดิม ดังนั้นคีย์จึงป้องกันความปลอดภัยให้ข้อมูลได้ ถ้าผู้อื่นได้ข้อความรหัสไป ผู้อื่นก็ไม่สามารถอ่านข้อความนั้นได้จำเป็นต้องนำไปถอดรหัสลับก่อน แม้ผู้อื่นจะทราบวิธีถอดรหัสลับแต่ไม่ทราบคีย์จึงไม่สามารถถอดรหัสลับเป็นข้อความที่ถูกต้องได้ วิธีเข้ารหัสลับและวิธีถอดรหัสลับอาจใช้คีย์ตัวเดียวกัน แสดงดังรูปที่ 4.2 วิธีเข้ารหัสลับและวิธีถอดรหัสลับอาจใช้คีย์คนละตัว แสดงดังรูปที่ 4.3



รูปที่ 4.2 แสดงการเข้ารหัสลับที่ใช้คีย์ตัวเดียว



รูปที่ 4.3 แสดงการเข้ารหัสลับที่ใช้คีย์คนละตัว

จะเห็นว่าการเข้ารหัสลับจะป้องกันข้อมูลให้ปลอดภัยได้นั้นขึ้นอยู่กับคีย์ที่ต้องเก็บเป็นความลับมิให้ผู้อื่นรู้ และขั้นตอนวิธีการ (algorithm) จะต้องซับซ้อนและสามารถเข้ารหัสลับได้อย่างมีประสิทธิภาพ แม้ผู้อื่นได้ข้อความที่เข้ารหัสลับไว้แต่ไม่สามารถถอดรหัสลับได้หรือใช้เวลานานและทรัพยากรต่างๆเป็นจำนวนมากจนไม่สามารถถอดรหัสลับได้

นอกจาก cryptography ซึ่งเป็นวิชาที่ศึกษาเกี่ยวกับการเข้ารหัสลับเพื่อป้องกันข้อมูลให้ปลอดภัย ยังมี cryptanalysis ซึ่งเป็นวิชาที่ศึกษาเกี่ยวกับการเข้ารหัสลับเพื่อทำลายการเข้ารหัสลับ (breakable encryption) นั้น โดยศึกษาวิธีเข้า/ถอดรหัสลับ หรือลักษณะข้อความเข้ารหัสไว้หรือลักษณะข้อความปกติและข้อความเข้ารหัสไว้คู่กัน แล้วลดจำนวนที่เป็นไปได้ของคีย์ที่ใช้ถอดรหัสลับ และอื่นๆ การเข้ารหัสลับแบ่งตามลักษณะคีย์ที่ใช้ในการเข้ารหัสลับได้ 2 ประเภทคือ

1. การเข้ารหัสลับแบบสัญนิยม (conventional system หรือ symmetric system)

การเข้ารหัสลับแบบนี้จะใช้คีย์สำหรับเข้ารหัสลับและถอดรหัสลับเป็นคีย์ตัวเดียวกัน หรือคีย์หนึ่งสามารถคำนวณจากอีกคีย์ได้ และคีย์จะต้องเก็บเป็นความลับมิให้ผู้อื่นรู้ ถ้าผู้อื่นรู้คีย์นี้จะสามารถถอดรหัสลับข้อมูลได้ทันที บางครั้งเรียกการเข้ารหัสลับแบบนี้ว่า single-key system หรือ secret-key system การเข้ารหัสลับแบบนี้แสดงดังรูปที่ 4.2 การเข้ารหัสลับแบบสัญนิยมที่นิยมใช้อย่างกว้างขวางคือ วิธีเข้ารหัสลับ DES (Data Encryption Standard) ซึ่งมีความปลอดภัยพอสมควรแต่ไม่เหมาะกับข้อมูลที่มีมูลค่าสูง วิธีเข้ารหัสลับ TripleDES ซึ่งปรับปรุงจากวิธี DES ให้มีความซับซ้อนมากขึ้น ขณะนี้วิธีเข้ารหัสลับ IDEA (International Data Encryption Algorithm) ซึ่งเป็นวิธีที่ผู้เชี่ยวชาญด้านการเข้ารหัสลับเชื่อว่าเป็นวิธีที่ความปลอดภัย (William Stallings, 1995)

2. การเข้ารหัสลับแบบคีย์สาธารณะ (public-key system)

การเข้ารหัสลับแบบนี้จะใช้คีย์สำหรับเข้ารหัสลับและถอดรหัสลับจำนวน 2 คีย์ที่ใช้งานคู่กัน คีย์ทั้งสองมีค่าต่างกันแต่มีความสัมพันธ์กัน คีย์ที่ใช้เข้ารหัสลับเรียกว่าคีย์สาธารณะ (public key) เป็นคีย์ที่ไม่จำเป็นต้องเก็บเป็นความลับ ส่วนคีย์ที่ใช้ถอดรหัสลับเรียกว่าคีย์ที่เป็นความลับ (secret key) ต้องไม่ให้ผู้อื่นรู้คีย์นี้ คีย์ทั้งสองมักจะคำนวณมาจากฟังก์ชันคณิตศาสตร์ที่ซับซ้อนและมีความสัมพันธ์กัน แม้ผู้อื่นรู้คีย์สาธารณะจะไม่สามารถคำนวณคีย์ที่เป็นความลับได้ จึงไม่สามารถถอดรหัสลับได้ การเข้ารหัสลับแบบคีย์สาธารณะที่นิยมใช้อย่างกว้างขวางคือ RSA (Rivest-Shamir-Adleman)

การเข้ารหัสลับแบบคีย์สาธารณะจะทำงานช้ากว่าการเข้ารหัสลับแบบสัญนิยมมากเช่น RSA ทำงานช้ากว่า DES ประมาณ 100 เท่า (Bruce Schneier, 1994) เป็นต้น การเข้ารหัสแบบสัญนิยมจะใช้เข้ารหัสลับข้อความทั้งหมดและการเข้ารหัสลับแบบคีย์สาธารณะมักใช้ในการจัดการคีย์เท่านั้น เช่นการส่งข้อมูลที่เป็นความลับระหว่าง 2 องค์กรซึ่งตกลงคีย์สาธารณะและคีย์ที่เป็นความลับที่ใช้ร่วมกันไว้ ผู้ส่งจะเข้ารหัสลับข้อมูลทั้งหมดด้วยวิธี DES จากนั้นนำคีย์ดังกล่าวเข้ารหัสลับอีกครั้งด้วยวิธี RSA โดยใช้คีย์สาธารณะ แล้วส่งคีย์และข้อมูลที่เข้ารหัสลับไว้ให้ผู้รับ เมื่อผู้รับได้รับแล้วจะเริ่มถอดรหัสคีย์ด้วยวิธี RSA โดยใช้คีย์ที่เป็นความลับ จากนั้นนำคีย์ที่ได้มาถอดรหัสลับข้อมูลทั้งหมดด้วยวิธี DES เนื่องจาก RSA ทำงานช้ากว่า DES มากจึงใช้ RSA เข้ารหัสลับคีย์เท่านั้นและ DES เข้ารหัสลับข้อมูลทั้งหมด

จากการเข้ารหัสลับแบบคีย์สาธารณะทำงานช้าจึงใช้จัดการคีย์เท่านั้น ส่วนข้อมูลทั้งหมดจะเข้ารหัสลับแบบสัญนิยม ซึ่งจะกล่าวเฉพาะการเข้ารหัสลับแบบสัญนิยมต่อไป

การเข้ารหัสลับแบบสัญนิยม

1. วิธีเข้ารหัสลับ DES (Data Encryption Standard Algorithm)

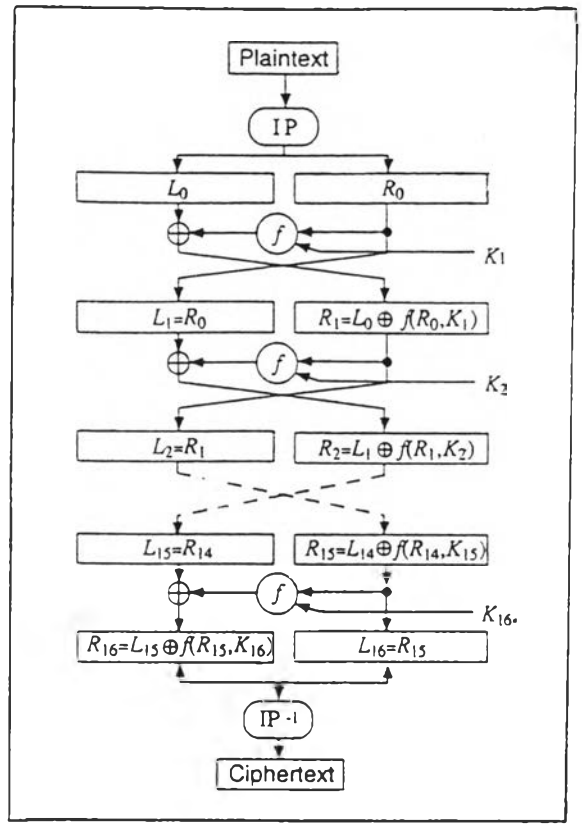
จากการพัฒนาเทคโนโลยีที่ก้าวหน้าและการเห็นความสำคัญของความปลอดภัยของข้อมูล จึงมีการพัฒนาการเข้ารหัสลับมากมาย แต่มีปัญหาเรื่องการติดต่อสื่อสารเพราะใช้วิธีเข้ารหัสลับและอุปกรณ์ที่ต่างกัน จึงได้มีการสร้างมาตราฐานสำหรับการเข้ารหัสลับขึ้น หน่วยงานที่รับผิดชอบคือ NBS (National Bureau of Standard) โดยในปีค.ศ. 1972 NBS ได้ชักชวนให้เสนอวิธีเข้ารหัสลับเพื่อใช้ในการป้องกันข้อมูลที่ส่งผ่านเครือข่ายการสื่อสาร หรือข้อมูลที่เก็บไว้ในสื่อ

ต่างๆ แต่วิธีที่เสนอมายัง NBS มีคุณสมบัติไม่ตรงกับที่ NBS ได้กำหนดไว้ จนกระทั่งปีค.ศ. 1974 บริษัทไอบีเอ็มได้เสนอวิธีเข้ารหัสลับ Lucifer ซึ่งมีคุณสมบัติตรงกับที่ NBS ได้กำหนดไว้ ต่อมา NSA (National Security Agency) และกลุ่มผู้เชี่ยวชาญได้ศึกษาการเข้ารหัสลับวิธีนี้อย่างละเอียด และปรับเปลี่ยนให้เหมาะสมจนได้วิธีเข้ารหัสลับ DES วิธีนี้ได้รับการยอมรับให้เป็นมาตราฐานในปีค.ศ. 1976 หลังจากนั้นวิธีนี้นำมาใช้งานในด้านธุรกิจการค้าอย่างกว้างขวางเช่น โปรแกรมเข้ารหัสลับพาร์ติชัน SecureDrive, เทปสำรองข้อมูล Contemporary Cybernetics เข้ารหัสลับข้อมูลขณะที่สำรองข้อมูล, ISDN เข้ารหัสลับข้อมูลและเสียงที่รับ/ส่งระหว่างเครื่องปลายทาง, ระบบเครือข่าย Semaphore Communications Corp. มีผลิตภัณฑ์ที่เข้ารหัสลับแบบ RSA และ DES เพื่อเพิ่มความปลอดภัยของข้อมูล เป็นต้น

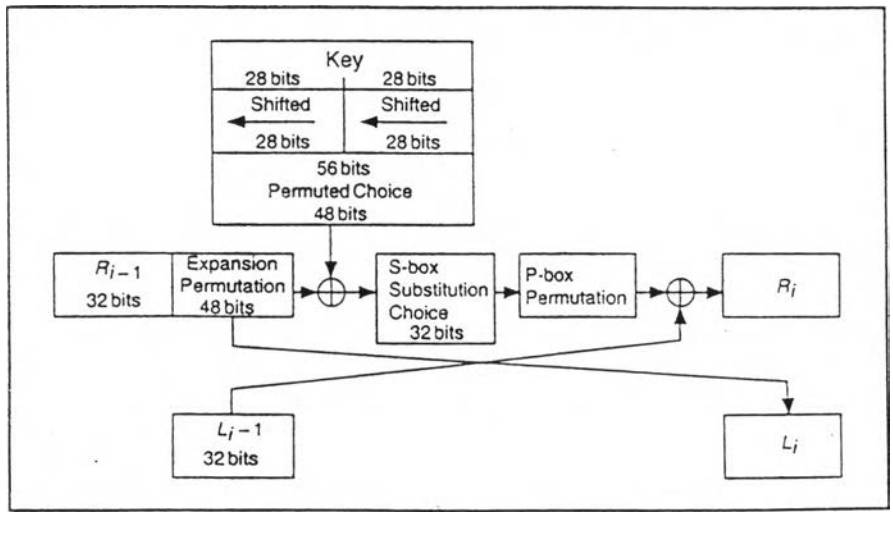
การทำงานของ DES ใช้วิธีสลับตำแหน่งข้อมูลและวิธีแทนที่ข้อมูล แล้วยังใช้การเอ็กซ์คลูซีฟออร์ (Exclusive-Or) วิธีเข้ารหัสลับด้วยวิธี DES จะทำงานกับบล็อกข้อมูล 64 บิตและใช้คีย์ 56 บิต แสดงดังรูปที่ 4.4 เริ่มจากบล็อกข้อมูลขนาด 64 บิตจะเรียงตำแหน่งบิตเริ่มต้น (initial permutation : IP) แล้วแบ่งเป็นบล็อกซ้าย 32 บิต (L_0) และบล็อกขวา 32 บิต (R_0) จากนั้นบล็อกซ้ายและบล็อกขวาจะผ่านฟังก์ชัน f ที่เป็นฟังก์ชันมีค่าขึ้นอยู่กับค่าคีย์ที่ใช้เข้ารหัสลับ ผลลัพธ์จากฟังก์ชัน f จะนำมาเอ็กซ์คลูซีฟออร์กับบล็อกซ้ายได้เป็นบล็อกขวาใหม่ (R_1) และบล็อกขวาเดิมจะกลายเป็นบล็อกซ้ายใหม่ (L_1) ทำงานซ้ำกันเป็นจำนวน 16 รอบ หลังนั้นรวมบล็อกซ้ายและบล็อกขวาแล้วเรียงตำแหน่งบิตผกผันกับแบบเริ่มต้น (inverse initial permutation : IP^{-1}) จะได้บล็อกข้อมูลที่เข้ารหัสลับไว้ขนาด 64 บิต

ทำงานแต่ละรอบของฟังก์ชัน f ที่ใช้คีย์ 56 บิตแสดงดังรูปที่ 4.5 เริ่มจากคำนวณคีย์ย่อย (subkey) โดยคีย์ขนาด 56 บิตแบ่งเป็น 2 ส่วนๆละ 28 บิต แต่ละส่วนจะเลื่อนตำแหน่งบิตแบบหมุนเวียนไปทางซ้าย (circular left shift) ในแต่ละรอบจะนำผลลัพธ์ทั้งสองส่วนมารวมกันได้ 56 บิต แล้วจัดเรียงตำแหน่งบิตใหม่โดยไม่ใช้บางบิต จะได้คีย์ย่อยขนาด 48 บิตจะทำซ้ำเดิม 16 รอบจึงได้เป็นคีย์ย่อย 16 คีย์คือ $K_1, K_2, K_3, \ldots, K_{16}$ แต่ละคีย์ย่อยจะนำมาใช้ในฟังก์ชันแต่ละรอบ

การทำงานในฟังก์ชัน f แสดงดังรูปที่ 4.5 บล็อกขวา (R_i) ขนาด 32 บิต จะมีการจัดเรียงตำแหน่งใหม่โดยขยายเป็นบล็อกขนาด 48 บิต จากนั้นนำมาเอ็กซ์คลูซีฟออร์กับคีย์ย่อย K_i จากนั้นนำผลลัพธ์มาแบ่งเป็น 8 กลุ่มๆละ 6 บิต แล้วนำแต่ละกลุ่มมาผ่านฟังก์ชัน S-box



รูปที่ 4.4 แสดงการทำงานของการเข้ารหัสลับวิธี DES



รูปที่ 4.5 แสดงการทำงานแต่ละรอบของฟังก์ชัน f

และเลือกให้เหลือเพียง 32 บิตแล้วจัดเรียงตำแหน่งใหม่โดยใช้ P-box permutation จึงได้เป็นผลลัพธ์ของฟังก์ชัน f แต่ละรอบ จากนั้นนำผลลัพธ์ของฟังก์ชัน f มาเอ็กซ์คลูซีฟออร์กับบล็อกซ้าย ผลลัพธ์ที่ได้จะเป็นบล็อกขวาของรอบต่อไป

การถอดรหัสลับจะทำงานเช่นเดียวกับการเข้ารหัสลับ ข้อแตกต่างคือคีย์ย่อยที่ใช้ในการถอดรหัสจะใช้ในลักษณะตรงข้ามกับที่ใช้เข้ารหัสลับคือ คีย์ย่อย K_{16} จะทำงานในรอบที่ 1, คีย์ย่อย K_{15} จะทำงานในรอบที่ 2 เช่นนี้เรื่อยไปจนถึงรอบที่ 16 จะใช้คีย์ย่อย K_1

วีคคีย์	เซมิวีคคีย์
0101 0101 0101 0101	01FE 01FE 01FE 01FE กับ FE01 FE01 FE01 FE01
FEFE FEFE FEFE FEFE	1FE0 1FE0 0EF1 0EF1 กับ E01F E01F F10E F10E
1F1F 1F1F 1F1F 1F1F	01E0 01E0 01F1 01F1 กับ E001 E001 F101 F101
E0E0 E0E0 E0E0 E0E0	1FFE 1FFE 0EFE 0EFE กับ FE1F FE1F FE0E FE0E
	011F 011F 010E 010E กับ 1F01 1F01 0E01 0E01
	E0FE E0FE F1FE F1FE กับ FEE0 FEE0 FEF1 FEF1

ตารางที่ 4.1 แสดงวีคคีย์และเซมิวีคคีย์ของการเข้ารหัสลับ DES

การทำงานของ DES มีประสิทธิภาพและซับซ้อนกล่าวคือฟังก์ชัน f ทำงานซ้ำเดิม 16 รอบ แต่ละรอบจะสลับตำแหน่งข้อมูล, แทนที่ข้อมูลและเอ็กซ์คลูซีฟออร์ข้อมูลกับค่าคีย์ จึงยากแก่การทำลาย เนื่องจากการแทนที่ข้อมูลจะป้องกันการนำคณิตศาสตร์มาวิเคราะห์หาความสัมพันธ์ระหว่างข้อมูลปกติ ข้อมูลเข้ารหัสและคีย์ ส่วนการสลับตำแหน่งข้อมูลจะป้องกันการนำสถิติมาวิเคราะห์หาความสัมพันธ์ดังกล่าวเช่นกัน และการเอ็กซ์คลูซีฟออร์ข้อมูลปกติกับคีย์ทำให้ข้อมูลเข้ารหัสเปลี่ยนแปลงตามข้อมูลปกติและคีย์ นอกจากนี้ DES ทำงานซ้ำเดิม 16 รอบจึงทำให้การทำงานซับซ้อน คีย์ที่ใช้ในการเข้ารหัสลับ DES จะต้องหลีกเลี่ยงวีค์คีย์ (weak key) ซึ่งเป็นคีย์ที่นำมาคำนวณได้เป็นคีย์ย่อยที่เหมือนกัน วีคคีย์ดังกล่าวจึงลดประสิทธิภาพและความซับซ้อนของการทำงาน ข้อมูลปกติที่เข้ารหัสลับด้วยวีคคีย์แล้วนำมาเข้ารหัสซ้ำอีกครั้งด้วยวีคคีย์เดิมจะได้ข้อความปกติตามเดิม วีคคีย์มีจำนวน 4 ตัวแสดงในตารางที่ 4.1 ซึ่งนำมาคำนวณเป็นคีย์ย่อยที่เป็น 0 หรือ 1 ทั้งหมด นอกจากนี้ยังมีคีย์ที่ใช้เข้ารหัสลับที่เป็นคู่เรียกว่า เซมิวีคคีย์ (semi-weak key) โดยแต่ละคู่ของเซมิวีคคีย์จะคำนวณเป็นคีย์ย่อยที่เหมือนกันแต่เรียงตรงข้ามกัน คีย์ย่อย K_1

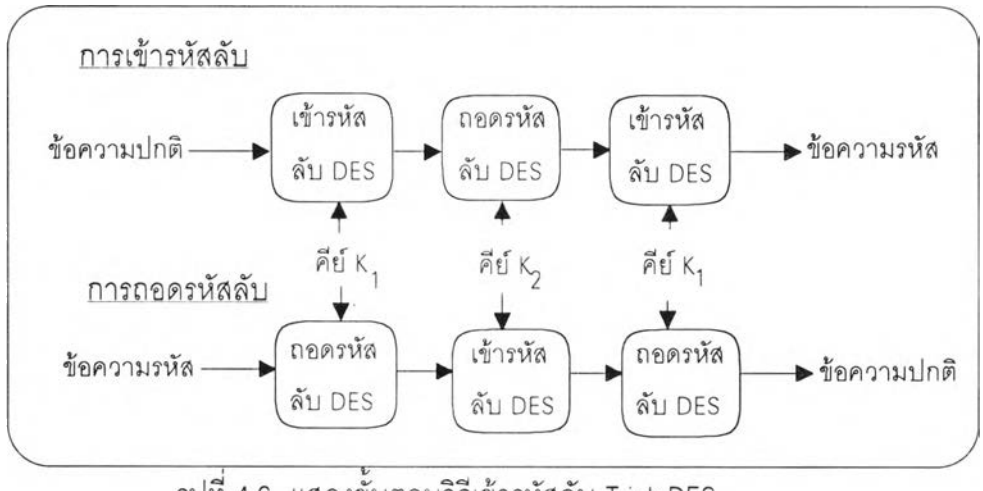
จากคีย์หนึ่งจะเหมือนคีย์ย่อย K_{16} จากอีกคีย์, คีย์ย่อย K_2 จากคีย์หนึ่งจะเหมือนคีย์ย่อย K_{15} จากอีกคีย์ เช่นนี้เรื่อยไป ข้อมูลปกติที่เข้ารหัสลับด้วยเซมิวีคคีย์แล้วนำมาเข้ารหัสซ้ำอีกครั้งด้วยเซมิวีคคีย์อีกตัวจะได้ข้อความปกติตามเดิม เซมิวีคคีย์มีจำนวน 6 คู่แสดงในตารางที่ 4.1 (สมศรี จตุรพิธพรชัย, 2533)

เนื่องจากการเข้ารหัสลับ DES ได้รับการยอมรับให้เป็นมาตราฐานจาก NBS จึงมีผู้เชี่ยวชาญหลายท่านตรวจสอบขั้นตอนการทำงานอย่างละเอียด แต่ไม่พบจุดอ่อนสำคัญที่จะทำให้ DES ไม่เป็นที่ยอมรับ อย่างไรก็ตามการทำลายการเข้ารหัสลับวิธีหนึ่งคือการบังคับโดยพลการ (brute force) เป็นการทำลายโดยถอดรหัสลับด้วยคีย์ที่เป็นไปได้ทั้งหมดเพื่อหาคีย์เข้ารหัสลับที่แท้จริง การบังคับโดยพลการนี้มีสมมติฐานว่าผู้ทำลายต้องทราบข้อมูลปกติและข้อมูลเข้ารหัสที่คู่กัน คีย์ที่เป็นไปได้ทั้งหมดของ DES มีจำนวน 2^{56} หรือ $7x10^{16}$ คีย์ การบังคับโดยพลการจะเริ่มจากเข้ารหัสลับข้อมูลปกติด้วยคีย์ที่เป็นไปได้ แล้วนำผลลัพธ์มาเปรียบเทียบกับข้อมูลเข้ารหัสที่ทราบ ถ้าข้อมูลเข้ารหัสทั้งสองตรงกันแสดงว่าคีย์นี้เป็นคีย์เข้ารหัสลับที่แท้จริง ถ้าข้อมูลเข้ารหัสทั้งสองไม่ตรงกันจะทดลองเข้ารหัสลับด้วยคีย์ที่เป็นไปได้อื่นต่อไป การบังคับโดยพลการนี้ใช้เวลาหาคีย์เข้ารหัสที่แท้จริงขึ้นอยู่กับความเร็วของเครื่องคอมพิวเตอร์ ในปีค.ศ. 1977 Diffie และ Hellman ได้เสนอความเห็นว่าความยาวของคีย์ 56 บิตยังให้ความปลอดภัยไม่เพียงพอ Diffie และ Hellman กล่าวว่าการบังคับโดยพลการเพื่อหาคีย์เข้ารหัสที่แท้จริงต้องสร้างเครื่องมือพิเศษที่ใช้เวลาหาคีย์เข้ารหัสลับที่แท้จริงในเวลา 1 วัน เครื่องมือพิเศษดังกล่าวต้องเสียค่าใช้จ่ายประมาณ 20 ล้านดอลลาร์ ซึ่งราคาแพงและใช้ทรัพยากรจำนวนมาก ดังนั้นความยาวคีย์ 56 บิตยังให้ความปลอดภัยเพียงพอ Diffie และ Hellman เชื่อว่าการพัฒนาด้านเทคโนโลยีมีความก้าวหน้ามากขึ้นทำให้เครื่องคอมพิวเตอร์ทำงานเร็วขึ้นและราคาถูกลง จึงเป็นไปได้ที่จะสามารถทำลายการเข้ารหัสลับ DES ด้วยการบังคับโดยพลการ (Bruce Schneier, 1994) ตารางที่ 4.2 แสดงเวลาที่ใช้หาคีย์เข้ารหัสที่แท้จริงด้วยการบังคับโดยพลการ สมมติว่าความเร็วเครื่องคอมพิวเตอร์ในปัจจุบันใช้เวลา 1 ไมโครวินาทีเข้ารหัสลับได้ 1 คีย์ และการพัฒนาด้านการประมวลผลแบบขนาน (parallel processing) ที่สามารถพัฒนาให้เป็นจริงได้ในอนาคตจะใช้เวลา 1 ไมโครวินาทีเข้ารหัสได้ 1 ล้านคีย์ จะเห็นว่าการหาคีย์เข้ารหัสที่แท้จริงด้วยการบังคับโดยพลการของความยาวคีย์ 56 บิตจะใช้เวลาเพียง 10.01 ชั่วโมงเท่านั้น ดังนั้นเทคโนโลยีด้านการประมวลผลแบบขนานในอนาคตจะทำให้ความยาวคีย์ 56 บิตไม่ปลอดภัย (William Stallings, 1995) อย่างไรก็ตามการเข้ารหัสลับ DES สามารถใช้งานได้อีกหลายปี

ความยาวคีย์ (บิต)	จำนวนคีย์ที่เป็นไปได้	เวลาที่ใช้ (1 เข้ารหัส/ไมโครวินาที)	เวลาที่ใช้ (10^6 เข้ารหัส/ไมโครวินาที)
32	$2^{32} = 4.3 \times 10^{9}$	2^{31} ไมโครวินาที = 35.8 นาที	2.15 ไมโครวินาที
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} ไมโครวินาที = 1142 ปี	10.01 ชั่วโมง
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} ไมโครวินาที = 5.7×10^{24} ปี	5.4×10^{18} ปี

ตารางที่ 4.2 แสดงเวลาที่ใช้หาคีย์เข้ารหัสที่แท้จริงด้วยการบังคับโดยพลการ

จากความเห็นว่าความยาวของคีย์ 56 บิตยังให้ความปลอดภัยไม่เพียงพอ จึงพัฒนาวิธีเข้ารหัสลับ TripleDES ที่เพิ่มประสิทธิภาพของความยาวคีย์ให้มากขึ้น วิธีนี้นำการเข้ารหัสลับ DES มาทำงาน 3 รอบโดยใช้คีย์ 2 ตัว จึงเสมือนใช้คีย์ขนาด 112 บิต การทำงานวิธีนี้แสดงดังรูปที่ 4.6



รูปที่ 4.6 แสดงขั้นตอนวิธีเข้ารหัสลับ TripleDES

ขั้นตอนการเข้ารหัสลับเริ่มจากนำข้อความปกติมาเข้ารหัสลับด้วยคีย์ตัวแรก จากนั้นนำผลลัพธ์มาถอดรหัสลับด้วยคีย์ตัวที่สอง สุดท้ายนำผลลัพธ์มาเข้ารหัสอีกครั้งด้วยคีย์ตัวแรก ผลลัพธ์สุดท้ายเป็นข้อความที่เข้ารหัสจากวิธี TripleDES ส่วนขั้นตอนการถอดรหัสลับจะทำงานย้อนกลับ เริ่มจากนำข้อความรหัสมาถอดรหัสลับด้วยคีย์ตัวแรก จากนั้นนำผลลัพธ์มาเข้ารหัสลับด้วยคีย์ตัวที่สอง สุดท้ายนำผลลัพธ์มาถอดรหัสอีกครั้งด้วยคีย์ตัวแรก ผลลัพธ์สุดท้ายเป็นข้อความปกติตามเดิม บางครั้งเรียกวิธีการนี้ว่า encrypt-decrypt-encrypt (EDE)

เนื่องจากวิธีเข้ารหัสลับ DES ยังทำงานอย่างมีประสิทธิภาพและซับซ้อนอยู่ แต่มีความวิตกว่าความยาวของคีย์ 56 บิตยังให้ความปลอดภัยไม่เพียงพอ ดังนั้นการนำวิธีเข้ารหัสลับ

DES มาทำงานวน 3 รอบโดยใช้คีย์ 2 ตัวเสมือนเพิ่มความยาวคีย์เป็น 112 บิตจึงลดความวิตกนี้ได้ นอกจากนี้วิธี TripleDES ใช้ขั้นตอนการทำงานเดิมและคีย์ขนาดเดิมจึงสามารถพัฒนาบนฮาร์ดแวร์และซอฟท์แวร์ที่มีอยู่ได้ทันที อย่างไรก็ตามการเข้ารหัสวิธีนี้ต้องหลีกเลี่ยงวีคคีย์และเซมิวีคคีย์เช่นเดียวกับการเข้ารหัส DES

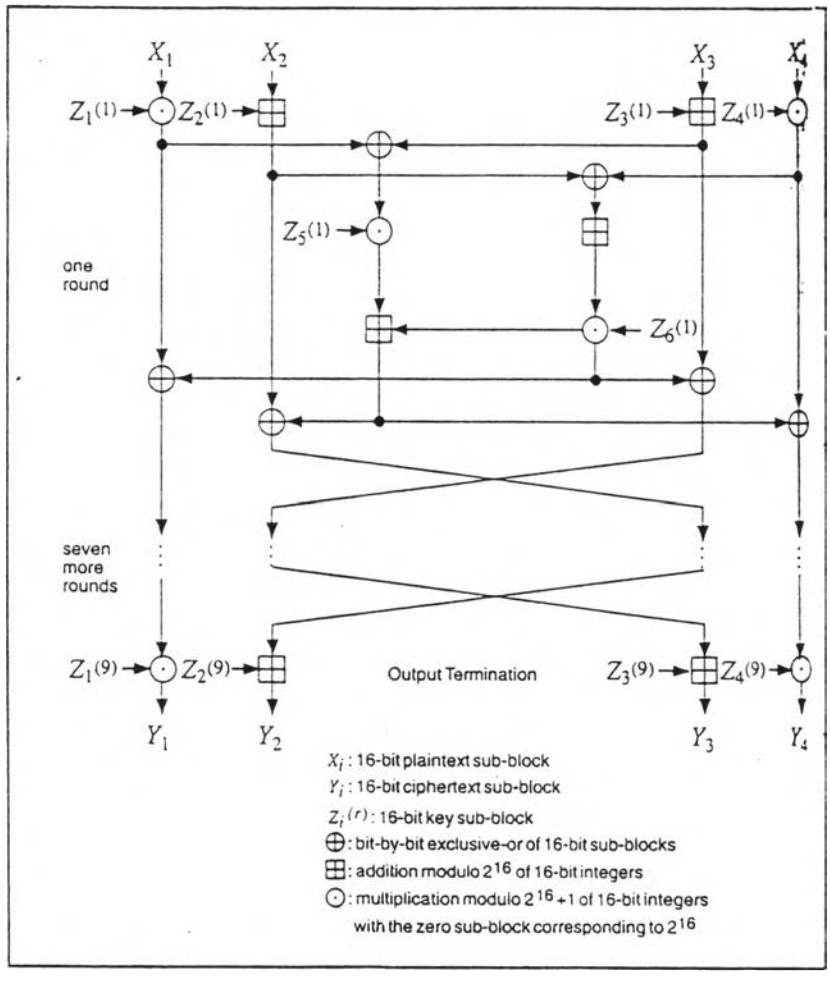
2. การเข้ารหัสลับ IDEA (International Data Encryption Algorithm)

การเข้ารหัสลับ IDEA เกิดครั้งแรกในปี 1990 โดย Xuejia Lai และ James Massey โดยตั้งชื่อว่า PES (Proposed Encryption Standard) ในปีต่อมาได้รับการพัฒนาเพิ่มเติมพร้อมทั้งเปลี่ยนชื่อเป็น IPES (Improved Proposed Encryption Standard) และเปลี่ยนชื่ออีกครั้งในปี 1992 เป็น IDEA (International Data Encryption Algorithm)

การเข้ารหัสลับ IDEA ทำงานกับบล็อกข้อมูล 64 บิตและใช้คีย์ขนาด 128 บิต ซึ่งมีขนาดเป็นสองเท่าของคีย์ที่ใช้ในวิธีเข้ารหัสลับ DES การทำงานของ IDEA จะเข้ารหัสลับด้วยวิธีทางคณิตศาสตร์ดังนี้ 1. เอ็กซ์คลูซีฟออร์ (exclusive-or : XOR) 2. การบวกมอดดูโล 2^{16} (addition modulo 2^{16}) 3. การคูณมอดดูโล $2^{16}+1$ (multiplication modulo $2^{16}+1$)

การทำงานของ IDEA เริ่มจากนำคีย์มาสร้างคีย์ย่อยจำนวน 52 คีย์เพื่อใช้ในการทำงานแต่ละรอบของ IDEA การสร้างคีย์ย่อยเริ่มจากแบ่งคีย์ขนาด 128 บิตเป็นคีย์ย่อยขนาด 16 บิตจำนวน 8 คีย์ จากนั้นเลื่อนบิตแบบหมุนเวียนไปทางซ้ายจำนวน 25 บิตแล้วแบ่งเป็นคีย์ย่อยขนาด 16 บิตจำนวน 8 คีย์อีก หลังจากนั้นเลื่อนบิตแล้วแบ่งเป็นคีย์ย่อยขนาด 16 บิตจนครบจำนวน 52 คีย์

การทำงานของ IDEA แสดงดังรูปที่ 4.7 จะทำงาน 8 รอบและอีก 1 รอบสุดท้ายจึงทำงานทั้งหมด 9 รอบ คีย์ย่อยขนาด 16 บิตจำนวน 52 คีย์จะนำมาใช้ใน 8 รอบแรก แต่ละรอบใช้คีย์ย่อยจำนวน 6 คีย์คือ $Z1^{(1)}$, $Z2^{(1)}$, ..., $Z6^{(1)}$ ส่วนรอบสุดท้ายจะใช้คีย์ย่อยเพียง 4 คีย์คือ $Z1^{(9)}$, $Z2^{(9)}$, ..., $Z4^{(9)}$ การทำงานของ IDEA เริ่มจากบล็อกข้อมูล 64 บิตจะนำมาแบ่งเป็นบล็อกข้อมูลย่อย 16 บิตจำนวน 4 บล็อกคือ X1, X2, X3, X4 เข้าสู่การทำงาน 8 รอบ แต่ละรอบบล็อกข้อมูลย่อยทั้ง 4 บล็อกจะ XOR, บวกและคูณกับบล็อกข้อมูลย่อยอื่นหรือคีย์ย่อย หลังการทำงานแต่ละรอบบล็อกข้อมูลย่อย X2 และบล็อกข้อมูลย่อย X3 จะสลับตำแหน่งกัน การทำงาน 8 รอบแรกจะมีการทำงานแต่ละรอบดังนี้



รูปที่ 4.7 แสดงขั้นตอนวิธีเข้ารหัสลับ IDEA

1. บล็อกข้อมูลย่อย X1 คูณกับคีย์ย่อย Z1
2. บล็อกข้อมูลย่อย X2 บวกกับคีย์ย่อย Z2
3. บล็อกข้อมูลย่อย X3 บวกกับคีย์ย่อย Z3
4. บล็อกข้อมูลย่อย X4 คูณกับคีย์ย่อย Z4
5. (ผลลัพธ์ข้อ 1) XOR (ผลลัพธ์ข้อ 3)
6. (ผลลัพธ์ข้อ 2) XOR (ผลลัพธ์ข้อ 4)
7. (ผลลัพธ์ข้อ 5) คูณกับคีย์ย่อย Z5
8. (ผลลัพธ์ข้อ 6) บวกกับ (ผลลัพธ์ข้อ 4)

9. (ผลลัพธ์ข้อ 8) คูณกับคีย์ย่อย Z6

10. (ผลลัพธ์ข้อ 7) บวกกับ (ผลลัพธ์ข้อ 9)

11. (ผลลัพธ์ข้อ 1) XOR (ผลลัพธ์ข้อ 9)

12. (ผลลัพธ์ข้อ 3) XOR (ผลลัพธ์ข้อ 9)

13. (ผลลัพธ์ข้อ 2) XOR (ผลลัพธ์ข้อ 10)

14. (ผลลัพธ์ข้อ 4) XOR (ผลลัพธ์ข้อ 10)

ผลลัพธ์จากข้อ 11, 12, 13, 14 จะใช้เป็นบล็อกข้อมูลย่อย 16 บิตจำนวน 4 บล็อก X1, X2, X3, X4 ในรอบต่อไป โดยสลับตำแหน่งบล็อกข้อมูลย่อย X2 และบล็อกข้อมูลย่อย X3 ยกเว้นรอบที่ 8 ไม่ต้องสลับตำแหน่งบล็อกข้อมูลย่อยทั้งสอง หลังจากทำงานครบ 8 รอบแล้วจะทำงานรอบสุดท้ายดังนี้

1. บล็อกข้อมูลย่อย X1 คูณกับคีย์ย่อย Z1
2. บล็อกข้อมูลย่อย X2 บวกกับคีย์ย่อย Z2
3. บล็อกข้อมูลย่อย X3 บวกกับคีย์ย่อย Z3
4. บล็อกข้อมูลย่อย X4 คูณกับคีย์ย่อย Z4

จากนั้นนำบล็อกข้อมูลย่อย 16 บิตจำนวน 4 บล็อก X1, X2, X3, X4 มารวมกันเป็น ข้อความรหัส การถอดรหัสลับจะทำงานเช่นเดียวกับการเข้ารหัสลับ ข้อแตกต่างคือคีย์ย่อยที่ใช้ในการถอดรหัสจะใช้ในลักษณะตรงข้ามกับที่ใช้เข้ารหัสลับคือ คีย์ย่อยตัวที่ 1 ในรอบที่ 9 จะทำงานในรอบที่ 1, คีย์ย่อยตัวที่ 1 ในรอบที่ 8 จะทำงานในรอบที่ 2, ... เรื่อยไป นอกจากนี้คีย์ย่อยที่ใช้ถอดรหัสลับเป็นค่าผกผันการบวก (addtive inverse) และค่าผกผันการคูณ(multiplicative inverse) ของคีย์ย่อยที่ใช้เข้ารหัสลับ ตารางที่ 4 3 แสดงคีย์ย่อยของแต่ละรอบที่ใช้เข้ารหัสลับและถอดรหัสลับ (Bruce Schneier, 1994)

แม้การเข้ารหัสลับ IDEA ยังใหม่แต่ผู้เชี่ยวชาญหลายท่านได้ตรวจสอบการทำงานแล้วเชื่อว่าเป็นวิธีเข้ารหัสลับที่ปลอดภัย การเข้ารหัสลับ IDEA ได้นำมาใช้ในผลิตภัณฑ์ PGP (Pretty Good Privacy) ซึ่งพัฒนาในปี 1991 โดย Philip Zimmermann เป็นมาตราฐานสากลสำหรับการเข้ารหัสลับข้อมูลที่รับส่งในอินเตอร์เน็ต เช่นอิเล็กทรอนิกส์เมล์ ปัจจุบัน PGP ใช้งานได้บนหลายระบบปฏิบัติการเช่น DOS, Macintosh, UNIX, Vax, Amiga เป็นต้น องค์กรที่ใช้ PGP เช่น

Computer Professionals for Social Responsibility (CPSR), The Electronic Frontier Foundation (EFF), The Society for Electronic Access (SEA)

รอบ	คีย์ย่อยที่ใช้เข้ารหัสลับ	คีย์ย่อยที่ใช้ถอดรหัสลับ
1	$Z1^{(1)}$ $Z2^{(1)}$ $Z3^{(1)}$ $Z4^{(1)}$ $Z5^{(1)}$ $Z6^{(1)}$	$Z1^{(9)}$-1 $-Z2^{(9)}$ $-Z3^{(9)}$ $Z4^{(9)}$-1 $Z5^{(8)}$ $Z6^{(8)}$
2	$Z1^{(2)}$ $Z2^{(2)}$ $Z3^{(2)}$ $Z4^{(2)}$ $Z5^{(2)}$ $Z6^{(2)}$	$Z1^{(8)}$-1 $-Z2^{(8)}$ $-Z3^{(8)}$ $Z4^{(8)}$-1 $Z5^{(7)}$ $Z6^{(7)}$
3	$Z1^{(3)}$ $Z2^{(3)}$ $Z3^{(3)}$ $Z4^{(3)}$ $Z5^{(3)}$ $Z6^{(3)}$	$Z1^{(7)}$-1 $-Z2^{(7)}$ $-Z3^{(7)}$ $Z4^{(7)}$-1 $Z5^{(6)}$ $Z6^{(6)}$
4	$Z1^{(4)}$ $Z2^{(4)}$ $Z3^{(4)}$ $Z4^{(4)}$ $Z5^{(4)}$ $Z6^{(4)}$	$Z1^{(6)}$-1 $-Z2^{(6)}$ $-Z3^{(6)}$ $Z4^{(6)}$-1 $Z5^{(5)}$ $Z6^{(5)}$
5	$Z1^{(5)}$ $Z2^{(5)}$ $Z3^{(5)}$ $Z4^{(5)}$ $Z5^{(5)}$ $Z6^{(5)}$	$Z1^{(5)}$-1 $-Z2^{(5)}$ $-Z3^{(5)}$ $Z4^{(5)}$-1 $Z5^{(4)}$ $Z6^{(4)}$
6	$Z1^{(6)}$ $Z2^{(6)}$ $Z3^{(6)}$ $Z4^{(6)}$ $Z5^{(6)}$ $Z6^{(6)}$	$Z1^{(4)}$-1 $-Z2^{(4)}$ $-Z3^{(4)}$ $Z4^{(4)}$-1 $Z5^{(3)}$ $Z6^{(3)}$
7	$Z1^{(7)}$ $Z2^{(7)}$ $Z3^{(7)}$ $Z4^{(7)}$ $Z5^{(7)}$ $Z6^{(7)}$	$Z1^{(3)}$-1 $-Z2^{(3)}$ $-Z3^{(3)}$ $Z4^{(3)}$-1 $Z5^{(2)}$ $Z6^{(2)}$
8	$Z1^{(8)}$ $Z2^{(8)}$ $Z3^{(8)}$ $Z4^{(8)}$ $Z5^{(8)}$ $Z6^{(8)}$	$Z1^{(2)}$-1 $-Z2^{(2)}$ $-Z3^{(2)}$ $Z4^{(2)}$-1 $Z5^{(1)}$ $Z6^{(1)}$
9	$Z1^{(9)}$ $Z2^{(9)}$ $Z3^{(9)}$ $Z4^{(9)}$	$Z1^{(1)}$-1 $-Z2^{(1)}$ $-Z3^{(1)}$ $Z4^{(1)}$-1

ตารางที่ 4.3 แสดงคีย์ย่อยของแต่ละรอบที่ใช้เข้ารหัสลับและถอดรหัสลับ

การทำงานของ IDEA ทำลายยากกว่าการทำงานของ DES เนื่องจากใช้วิธีทางคณิตศาสตร์กระทำระหว่างบล็อกข้อมูลย่อยกับคีย์ย่อย การเข้ารหัสลับ IDEA ใช้วิธีทางคณิตศาสตร์จึงพัฒนาบนฮาร์ดแวร์และซอฟท์แวร์ได้ง่ายกว่าการเข้ารหัสลับ DES และใช้เวลาทำงานน้อยกว่า DES และ TripleDES วิธีนี้มีวีคคีย์ที่นำมาสร้างเป็นคีย์ย่อยที่เหมือนกันเพียงตัวเดียวคือ 0 ทั้งหมด จากตารางที่ 4.2 จะเห็นว่าเทคโนโลยีด้านการประมวลผลแบบขนาน (parallel processing) ที่เป็นจริงได้ในอนาคตจะหาคีย์เข้ารหัสที่แท้จริงด้วยการบังคับโดยพลการต้องใช้เวลาถึง 5.4×10^{18} ปี ดังนั้นความยาวคีย์ 128 บิตจึงปลอดภัยต่อการทำลายด้วยการบังคับโดยพลการ (William Stallings, 1995) ดังนั้นการเข้ารหัสลับ IDEA จึงเป็นวิธีที่เหมาะสมสำหรับการพัฒนาโปรแกรมอรรถประโยชน์สำหรับจัดการแฟ้มเอกสารของจุฬาจารึกทางด้านการเข้ารหัสลับข้อมูล

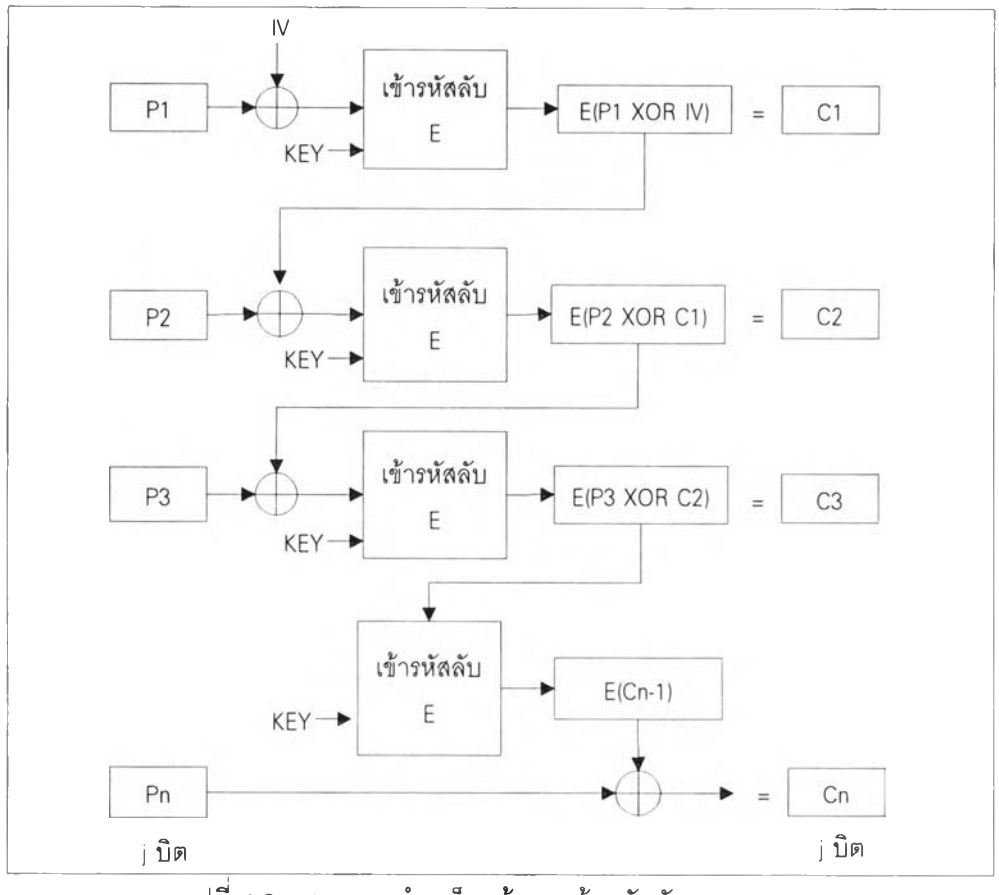
<u>การเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก</u>

แฟ้มข้อมูลจุฬาจารึกได้ออกแบบโครงสร้างแฟ้มข้อมูลให้จัดเก็บข้อมูลอย่างมีระเบียบ โดยจัดเก็บข้อมูลเป็นส่วนต่างๆ 10 ส่วนคือ ส่วน Header, ส่วน Style, ส่วน Section, ส่วน Font, ส่วน Picture, ส่วน Text, ส่วน Index, ส่วน Attribute, ส่วน Overlay Drawing และส่วน TTF Embed

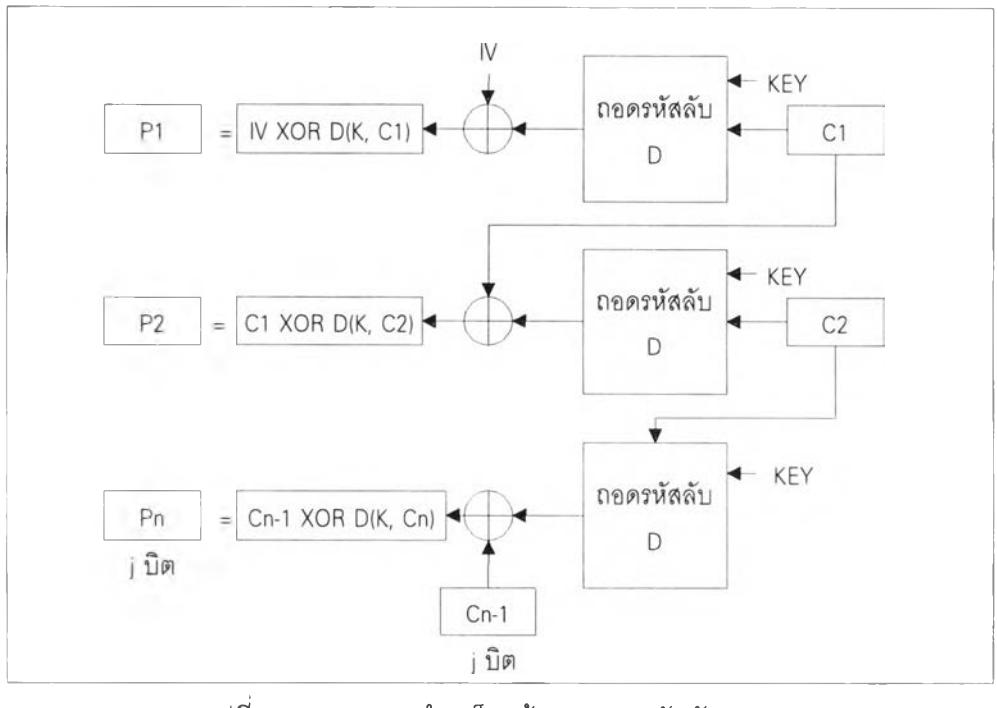
เมื่อจุฬาจารึกเปิดแฟ้มข้อมูลจะตรวจสอบว่าเป็นแฟ้มข้อมูลจุฬาจารึกหรือไม่ โดยตรวจสอบตัวระบุแฟ้มข้อมูล (0xA8CCE00) ที่ส่วน Header และตรวจสอบว่าโครงสร้างแฟ้มข้อมูลถูกต้องหรือไม่ โดยตรวจสอบจุดเริ่มต้นของแต่ละส่วน การเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึกจะเข้ารหัสลับข้อมูลเฉพาะส่วน Text, ส่วน Picture และส่วน Overlay Drawing เท่านั้น Bruce Schneir ได้ศึกษาและพัฒนาการเข้ารหัสลับต่างๆเป็นภาษาซีไว้ในหนังสือ Applied Cryptography ในการวิจัยครั้งนี้นำการเข้ารหัสลับ IDEA ซึ่ง Bruce Schneir พัฒนาด้วยภาษาซีมาดัดแปลงให้เหมาะสมสำหรับการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก

การเข้ารหัสลับจะทำงานกับข้อมูลทีละบล็อกเช่น การเข้ารหัสลับ IDEA จะทำงานกับบล็อกข้อมูล 64 บิต การนำข้อมูลมาเข้ารหัสลับนิยมนำมาทีละบล็อกตั้งแต่ต้นแฟ้มข้อมูลจนถึงท้ายแฟ้มข้อมูล โดยนำข้อมูลปกติทีละบล็อกมาเข้ารหัสลับแล้วจัดเก็บข้อมูลรหัสลงในบล็อกเดิมเช่นนี้เรื่อยไปจนจบ การนำบล็อกข้อมูลมาเข้ารหัสลับในลักษณะนี้ไม่มีประสิทธิภาพเท่าที่ควร เช่น ข้อมูลจากแบบฟอร์มจะมีข้อมูลซ้ำๆกัน เมื่อนำบล็อกข้อมูลที่ซ้ำกันนั้นมาเข้ารหัสลับจะได้บล็อกข้อมูลรหัสที่ซ้ำกันด้วย ผู้อื่นจะสามารถดักข้อมูลแล้วแก้ไขข้อมูลส่วนที่สำคัญได้ง่ายมาก จะเห็นว่าบล็อกข้อมูลรหัสข้างต้นจะมีค่าขึ้นอยู่กับบล็อกข้อมูลปกติและคีย์เท่านั้น จึงต้องปรับปรุงวิธีนำบล็อกข้อมูลมาเข้ารหัสลับให้มีประสิทธิภาพยิ่งขึ้น การปรับปรุงวิธีหนึ่งคือการนำบล็อกข้อมูลมาเข้ารหัสลับในลักษณะลูกโซ่ เรียกวิธีนี้ว่า CBC (Cipher Block Chain) กล่าวคือหลังจากเข้ารหัสลับบล็อกข้อมูลปกติแล้ว จะนำบล็อกข้อมูลรหัสที่ได้กลับมาปรับปรุงการเข้ารหัสลับของบล็อกข้อมูลถัดไป ดังนั้นบล็อกข้อมูลรหัสจะมีค่าขึ้นอยู่กับบล็อกข้อมูลปกติ, คีย์และบล็อกข้อมูลรหัสก่อนหน้า

การนำบล็อกข้อมูลมาเข้ารหัสลับแสดงดังรูปที่ 4.8 เริ่มจากกำหนดค่าเริ่มต้นลูกโซ่ (initial chaining value : IV) ในรีจีสเตอร์ จากนั้นนำบล็อกข้อมูลปกติมาเอ็กซ์คลูซีฟออร์กับค่าในรีจีสเตอร์ ผลลัพธ์ที่ได้เป็นบล็อกแรกของข้อมูลรหัส พร้อมทั้งนำผลลัพธ์นั้นเก็บในรีจีสเตอร์เพื่อนำมาเอ็กซ์คลูซีฟออร์กับบล็อกต่อไปของข้อมูล ดังนั้นบล็อกต่อไปของข้อมูลที่จะเข้ารหัสลับจะเปลี่ยนแปลงตามบล็อกข้อมูลรหัสนี้ ทำซ้ำเดิมเรื่อยไป สมการคณิตศาสตร์ของการนำบล็อกข้อมูลมาเข้ารหัสลับแบบ CBC คือ $C = E(K, P \text{ XOR } C_{j})$ บล็อกสุดท้ายอาจมีข้อมูลไม่ครบ 64 บิต สมมติว่าบล็อกสุดท้ายมีข้อมูลเพียง j บิต การนำบล็อกข้อมูลมาเข้ารหัสลับจะทำงานซ้ำเดิมจนถึงบล็อกสุดท้ายที่มีข้อมูลครบ 64 บิต จะนำบล็อกสุดท้ายของข้อมูลรหัสมาเข้ารหัสลับซ้ำอีก



รูปที่ 4.8 แสดงการนำบล็อกข้อมูลเข้ารหัสลับแบบ CBC



รูปที่ 4.9 แสดงการนำบล็อกข้อมูลถอดรหัสลับแบบ CBC

ครั้ง จากนั้นนำผลลัพธ์ที่ได้จำนวน j บิตมาเอ็กซ์คูลซีฟออร์กับบล็อกสุดท้ายของข้อมูลที่มี j บิตเช่นกัน ได้เป็นบล็อกข้อมูลรหัส j บิต

การนำบล็อกข้อมูลมาถอดรหัสลับแสดงดังรูปที่ 4.9 เริ่มจากกำหนดค่าเริ่มต้นลูกโซ่ในรีจีสเตอร์ จากนั้นถอดรหัสบล็อกแรกของข้อมูลรหัส นำผลลัพธ์ที่ได้มาเอ็กซ์คลูซีฟออร์กับค่าในรีจีสเตอร์ได้เป็นบล็อกแรกของข้อมูลปกติ พร้อมทั้งนำบล็อกแรกของข้อมูลรหัสเก็บในรีจีสเตอร์เพื่อนำมาเอ็กซ์คลูซีฟออร์กับผลลัพธ์จากการถอดรหัสบล็อกต่อไป ทำซ้ำเดิมเรื่อยไป สมการคณิตศาสตร์ของการนำบล็อกข้อมูลมาถอดรหัสลับแบบ CBC คือ P = C XOR D(K, C) เมื่อถอดรหัสถึงบล็อกสุดท้ายที่มีข้อมูลเพียง j บิต จะนำบล็อกสุดท้ายที่มีข้อมูลครบ 64 บิตมาเข้ารหัสลับนำผลลัพธ์ที่ได้จำนวน j บิตมาเอ็กซ์คลูซีฟออร์กับบล็อกสุดท้ายที่มีข้อมูล j บิตได้เป็นข้อมูลเดิม

การออกแบบและพัฒนาโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูล

โครงสร้างแฟ้มข้อมูลจุฬาจารึก 78 ได้จัดเก็บข้อมูลแต่ละส่วนแยกเป็นส่วนต่างๆ 10 ส่วนคือ ส่วน Header, ส่วน Style, ส่วน Section, ส่วน Font, ส่วน Picture, ส่วน Text, ส่วน Index, ส่วน Attribute, ส่วน Overlay Drawing และส่วน TTF Embed การเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึกจะเข้ารหัสลับข้อมูลเฉพาะส่วน Text, ส่วน Picture และส่วน Overlay Drawing

การวิจัยนี้นำเทคนิคการเข้ารหัสลับ IDEA ที่ Bruce Schneier พัฒนาเป็นภาษาซีมาดัดแปลงให้เหมาะสมสำหรับการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก โปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึกจะแบ่งเป็น 2 ส่วนคือส่วนเข้ารหัสลับแฟ้มข้อมูลเป็นขั้นตอนการเปลี่ยนข้อความปกติที่อ่านเข้าใจได้ให้เป็นข้อความรหัสที่ไม่สามารถอ่านเข้าใจได้ และส่วนถอดรหัสลับเป็นขั้นตอนย้อนกลับที่เปลี่ยนจากข้อความรหัสเป็นข้อความปกติตามเดิม วิธีเข้ารหัสลับ IDEA และการนำบล็อกข้อมูลมาเข้ารหัสลับแบบ CBC ได้กล่าวไว้แล้วในหัวข้อที่ผ่านมา การเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึกจะเก็บตัวตรวจสอบการใช้คีย์ที่ถูกต้อง encryptID ไว้ท้ายแฟ้มข้อมูล การถอดรหัสลับจะนำ encryptID มาถอดรหัสด้วยคีย์ก่อนแล้วตรวจสอบว่าคีย์ที่ใช้ถูกต้องหรือไม่

1. โครงสร้างข้อมูลที่ใช้ในโปรแกรม

โครงสร้างข้อมูลที่ใช้ในโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึกที่สำคัญดังนี้

1.1 ค่าคงที่ เป็นค่าที่กำหนดไว้ตั้งแต่ต้นโปรแกรมโดยไม่มีการเปลี่ยนแปลง ค่าคงที่แสดงดังตารางที่ 4.4

ชื่อค่าคงที่	ค่าที่เก็บ	คำอธิบาย
BUFSIZE	10000	ขนาดหน่วยความจำทั้งหมดของ buffer
NUMSECTION	9	จำนวนส่วนของแฟ้มของจุฬาจารึกที่อัดข้อมูล
ROUNDS	8	จำนวนรอบการทำงานของ IDEA
KEYLEN	52	จำนวนคีย์ย่อย
IDEAKEYSIZE	16	จำนวนไบต์ของคีย์
IDEABLOCKSIZE	8	จำนวนไบต์ของบล็อกข้อมูล
ENCRYPT	0	เข้ารหัสลับ
DECRYPT	1	ถอดรหัสลับ
TEXT	2	ลำดับที่ของส่วน Text
PICTURE	6	ลำดับที่ของส่วน Picture
OVERLAY	8	ลำดับที่ของส่วน Overlay Drawing
ENCRYPTID	yaowalaklimkongk	คำที่ใช้ตรวจสอบว่าเป็นคีย์ที่ถูกต้องหรือไม่

ตารางที่ 4.4 แสดงค่าคงที่ในโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับ

1.2 ตัวแปร จะกล่าวเฉพาะตัวแปรที่สำคัญที่ใช้ในโปรแกรมอรรถประโยชน์ด้านเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก ดังนี้

header	เป็นตัวแปรโครงสร้างของส่วน Header (โครงสร้างข้อมูลกล่าวไว้ในบทที่ 2)
id	เป็นตัวแปรแถวลำดับตัวเลขขนาด 9 เป็นตัวระบุส่วนของแต่ละส่วน
s	เป็นตัวแปรชนิดตัวเลข เป็นลำดับที่ของส่วนที่ทำงานอยู่
fsize	เป็นตัวแปรชนิดตัวเลข ขนาดแฟ้มข้อมูลที่ต้องการเข้า/ถอดรหัสลับ
secsize	เป็นตัวแปรชนิดตัวเลข ขนาดส่วนที่ต้องการเข้า/ถอดรหัสลับต่อไป

buffer	เป็นตัวแปรแถวลำดับชนิดตัวอักษรขนาด10000 ใช้สำหรับอ่านข้อมูลจากแฟ้มข้อมูลและจัดเก็บข้อมูลที่เข้า/ถอดรหัสลับแล้ว
Xptr	เป็นตัวแปรพอยเตอร์ชี้บล็อกข้อมูลที่จะเข้า/ถอดรหัสลับ
cycle	เป็นตัวแปรชนิดตัวเลข จำนวนบล็อกข้อมูลที่เข้า/ถอดรหัสลับ
remain	เป็นตัวแปรชนิดตัวเลข จำนวนไบต์ของข้อมูลในบล็อกสุดท้าย
Z	เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 52 เป็นคีย์ย่อยใช้เข้า/ถอดรหัสลับ
key	เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 8 เป็นคีย์ใช้เข้า/ถอดรหัสลับ
IV	เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 4 เป็นค่าเริ่มต้นลูกโซ่
randbuf	เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 4 เก็บตัวเลขสุ่ม
dum	เป็นตัวแปรแถวลำดับชนิดตัวเลขขนาด 4 เป็นหน่วยความจำชั่วคราว
encryptID	เป็นตัวแปรแถวลำดับชนิดตัวอักษรขนาด 16 เป็นตัวตรวจสอบคีย์ที่ถูกต้อง

2. การพัฒนาโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก

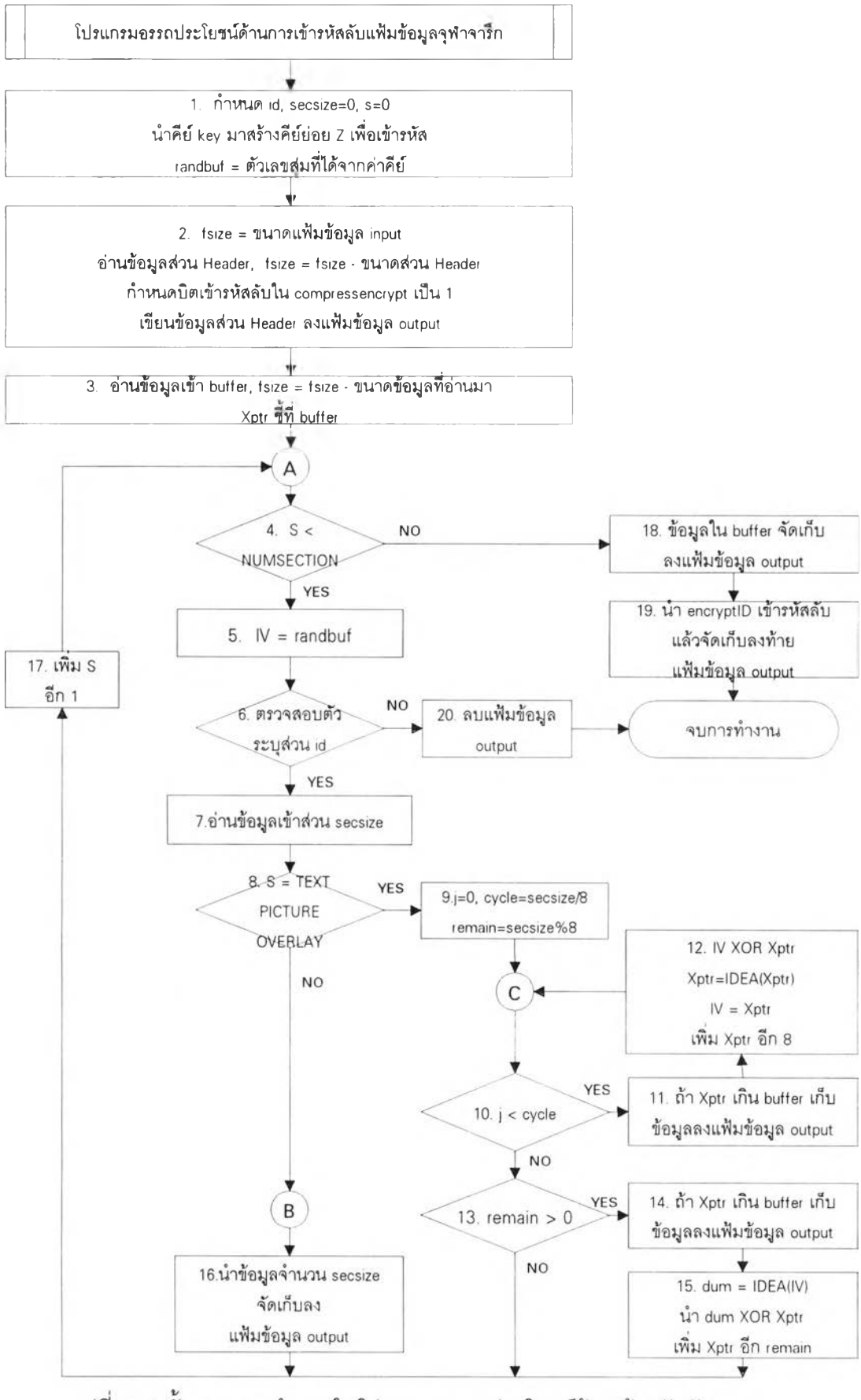
ในที่นี้จะกล่าวถึงขั้นตอนการทำงานทั้งหมดของโปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก ซึ่งแบ่งการทำงานเป็น 2 ส่วนคือส่วนการเข้ารหัสลับแฟ้มข้อมูลและส่วนการถอดรหัสลับแฟ้มข้อมูล ผังงานอธิบายการเข้ารหัสลับแฟ้มข้อมูลแสดงดังรูปที่ 4.10 และผังงานอธิบายการถอดรหัสลับแฟ้มข้อมูลแสดงดังรูปที่ 4.11 แต่ละขั้นตอนมีรายละเอียดมาก จำเป็นต้องใช้ผังงานหลักอธิบายการทำงานทั้งหมดก่อน จากนั้นใช้ผังงานย่อยอธิบายรายละเอียดต่อไป

2.1 โปรแกรมอรรถประโยชน์ด้านการเข้ารหัสลับแฟ้มข้อมูลจุฬาจารึก ผังงานของขั้นตอนการทำงานทั้งหมดแสดงดังรูปที่ 4.10 มีขั้นตอนการทำงานดังนี้

1. กำหนดค่าเริ่มต้นของตัวแปรที่ใช้ในโปรแกรม โดยกำหนดตัวระบุส่วนใน id ให้ขนาดส่วนที่อ่านเข้ามา secsize เป็น 0 และลำดับที่ของส่วน s เป็น 0 จากนั้นนำคีย์ key มาสร้างคีย์ย่อย Z เพื่อใช้เข้ารหัสลับ พร้อมทั้งหาตัวเลขสุ่มจากคีย์เก็บใน randbuf

2. หาขนาดแฟ้มข้อมูล fsize อ่านข้อมูลส่วน Header พร้อมทั้งลดขนาด fsize กำหนดบิตเข้ารหัสลับเป็น 1 จากนั้นเขียนข้อมูลส่วน Header ลงในแฟ้มข้อมูล output

3. อ่านข้อมูลเข้า buffer พร้อมทั้งลดขนาด fsize ให้ Xptr ชี้ที่ buffer



รูปที่ 4.10 ขั้นตอนการทำงานในโปรแกรมอรรถประโยชน์ด้านเข้ารหัสลับ

4. ตรวจสอบว่า S < NUMSECTION หรือไม่ ถ้าคำตอบคือใช่ จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 20

5. นำตัวเลขสุ่มจากค่าคีย์มาเป็นค่าเริ่มต้นลูกโซ่ IV

6. ตรวจสอบว่าตัวระบุส่วน id ถูกต้องหรือไม่ ถ้าคำตอบคือใช่ จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 22

7. อ่านขนาดของส่วน secsize ที่ต้องการทำงานต่อ

8. ตรวจสอบว่า S เป็น TEXT, PICTURE, OVERLAY หรือไม่ ถ้าคำตอบคือใช่ แสดงต้องการเข้ารหัสลับข้อมูลในส่วนนี้จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 16

9. หาจำนวนบล็อกข้อมูลที่ต้องเข้ารหัสลับ cycle และหาจำนวนไบต์ของบล็อกสุดท้าย remain และให้ j เป็น 0

10. ตรวจสอบว่า j < cycle หรือไม่ ถ้าคำตอบคือใช่ แสดงว่ายังมีบล็อกข้อมูลที่เข้ารหัสลับจะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่ จะข้ามไปทำงานข้อ 13

11. ถ้า Xptr ชี้เกิน buffer แสดงว่าข้อมูลใน buffer หมดแล้วต้องอ่านข้อมูลเข้ามาใหม่

12. นำค่าใน IV มาเอ็กซ์คลูซีฟออร์กับบล็อกข้อมูลที่ Xptr จากนั้นนำมาทำงานตามขั้นตอนวิธี IDEA (อธิบายในบทที่ 2) ผลลัพธ์ที่ได้เป็นข้อมูลที่เข้ารหัสลับเก็บในบล็อกข้อมูลเดิม เก็บข้อมูลที่เข้ารหัสแล้วใน IV เพื่อใช้เปลี่ยนแปลงบล็อกข้อมูลต่อไป จากนั้นเพิ่ม Xptr อีก 8 เพื่อทำงานบล็อกข้อมูลถัดไป แล้ววนกลับไปข้อ 10

13. เมื่อบล็อกข้อมูลที่ทำงานหมดแล้ว จะตรวจว่า remain > 0 หรือไม่ ถ้าคำตอบคือใช่ แสดงว่ามีเศษของบล็อกข้อมูลเหลืออยู่จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่ แสดงว่าข้อมูลหมดแล้วจะข้ามไปทำงานข้อ 17

14. ถ้า Xptr ชี้เกิน buffer แสดงว่าข้อมูลใน buffer หมดแล้วต้องอ่านข้อมูลเข้ามาใหม่

15. นำค่าใน IV มาทำงานตามขั้นตอนวิธี IDEA (อธิบายในหัวข้อที่ผ่านมา) ผลลัพธ์ที่ได้เก็บใน dum จากนั้นนำค่าใน dum มาเอ็กซ์คลูซีฟออร์กับเศษข้อมูลในบล็อกที่ Xptr ชี้อยู่ จากนั้นเพิ่ม Xptr อีก remain แล้วข้ามไปข้อ 17

16. เนื่องจากเป็นข้อมูลที่ไม่จำเป็นต้องเข้ารหัสลับจึงอ่านข้อมูลส่วนแล้วจัดเก็บในแฟ้มข้อมูล output

17. เพิ่มลำดับที่ส่วน s อีก 1 เพื่อทำงานส่วนถัดไป แล้ววนกลับไปข้อ 4

18. เมื่อทำงานครบทุกส่วนแล้วจะเก็บข้อมูลที่เข้ารหัสลับใน buffer ลงในแฟ้มข้อมูล output

19 นำ encryptID มาเข้ารหัสลับแล้วเก็บลงท้ายแฟ้มข้อมูลในส่วน TTFEmbed เพื่อใช้ตรวจสอบคีย์ในขั้นตอนถอดรหัสลับ แล้วออกจากการทำงาน

20. ตัวระบุส่วนไม่ถูกต้องจะลบแฟ้มข้อมูล output แล้วออกจากการทำงาน

2 2 โปรแกรมอรรถประโยชน์ด้านการถอดรหัสลับแฟ้มข้อมูลจุฬาจารึก ผังงานของขั้นตอนการทำงานทั้งหมดแสดงดังรูปที่ 4.11 มีขั้นตอนการทำงานดังนี้

1. กำหนดค่าเริ่มต้นของตัวแปรที่ใช้ในโปรแกรม โดยกำหนดตัวระบุส่วนใน id ให้ขนาดส่วนที่อ่านเข้ามา secsize เป็น 0 และลำดับที่ของส่วน s เป็น 0 จากนั้นนำคีย์ key มาสร้างคีย์ย่อย Z เพื่อใช้ถอดรหัสลับ พร้อมทั้งหาตัวเลขสุ่มจากคีย์เก็บใน randbuf จากนั้นอ่าน encryptID ท้ายแฟ้มข้อมูลขึ้นมาแล้วถอดรหัสลับ

2. ตรวจสอบว่า encryptID เท่ากับ ENCRYPTID หรือไม่ ถ้าคำตอบคือใช่ จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่ แสดงว่าใช้คีย์ไม่ถูกต้อง จะออกจากการทำงาน

3. หาขนาดแฟ้มข้อมูล fsize อ่านข้อมูลส่วน Header พร้อมทั้งลดขนาด fsize กำหนดบิตเข้ารหัสลับเป็น 0 จากนั้นเขียนข้อมูลส่วน Header ลงในแฟ้มข้อมูล output แล้วอ่านข้อมูลเข้า buffer พร้อมทั้งลดขนาด fsize ให้ Xptr ชี้ที่ buffer

4. ตรวจสอบว่า S < NUMSECTION หรือไม่ ถ้าคำตอบคือใช่ จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 20

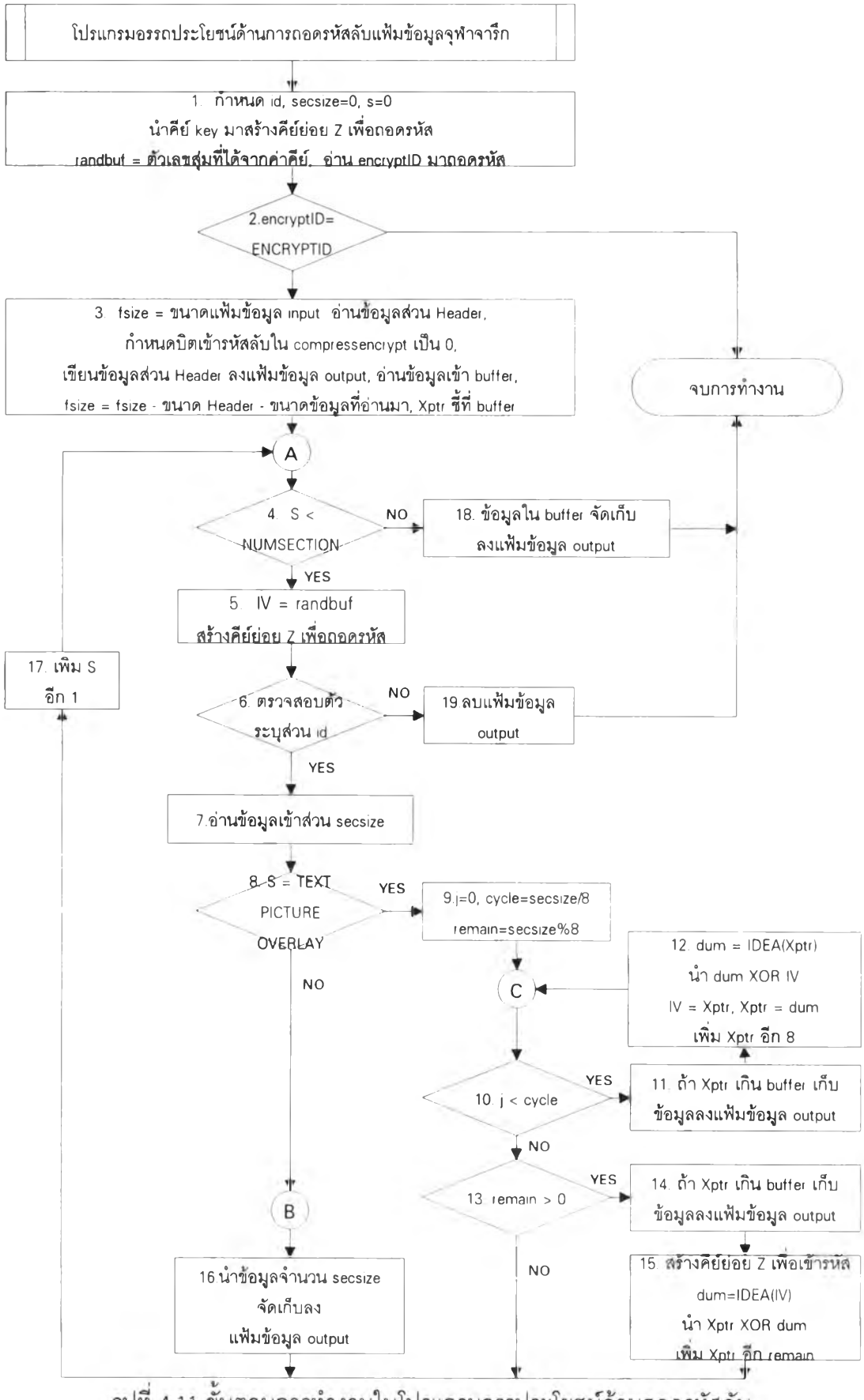
5. นำตัวเลขสุ่มจากค่าคีย์มาเป็นค่าเริ่มต้นลูกโซ่ IV สร้างคีย์ย่อย Z เพื่อใช้ถอดรหัสลับ

6. ตรวจสอบว่าตัวระบุส่วน id ถูกต้องหรือไม่ ถ้าคำตอบคือใช่ จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 21

7. อ่านขนาดของส่วน secsize ที่ต้องการทำงานต่อ

8. ตรวจสอบว่า S เป็น TEXT, PICTURE, OVERLAY หรือไม่ ถ้าคำตอบคือใช่ แสดงต้องการถอดรหัสลับข้อมูลในส่วนนี้จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่จะข้ามไปทำงานข้อ 16

9. หาจำนวนบล็อกข้อมูลที่ต้องเข้ารหัสลับ cycle และหาจำนวนไบต์ของ



รูปที่ 4.11 ขั้นตอนการทำงานในโปรแกรมอรรประโยชน์ด้านถอดรหัสลับ

บล็อกสุดท้าย remain และให้ j เป็น 0

10. ตรวจสอบว่า j < cycle หรือไม่ ถ้าคำตอบคือใช่ แสดงว่ายังมีบล็อกข้อมูลที่ถอดรหัสลับจะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่ จะข้ามไปทำงานข้อ 13

11. ถ้า Xptr ชี้เกิน buffer แสดงว่าข้อมูลใน buffer หมดแล้วต้องอ่านข้อมูลเข้ามาใหม่

12. นำบล็อกข้อมูลที่ Xptr มาทำงานตามขั้นตอนวิธี IDEA (อธิบายในบทที่ 2) เก็บใน dum พร้อมทั้งนำมาเอ็กซ์คลูซีฟออร์กับค่าใน IV ผลลัพธ์ที่ได้ใน dum เป็นข้อมูลที่ถอดรหัสลับแล้ว จากนั้นนำบล็อกข้อมูล Xptr เก็บใน IV เพื่อใช้เปลี่ยนแปลงบล็อกข้อมูลต่อไป แล้วนำข้อมูลที่ถอดรหัสแล้วใน dum มาเก็บในบล็อกข้อมูล Xptr จากนั้นเพิ่ม Xptr อีก 8 เพื่อทำงานบล็อกข้อมูลถัดไป แล้ววนกลับไปข้อ 10

13. เมื่อบล็อกข้อมูลที่ทำงานหมดแล้ว จะตรวจว่า remain > 0 หรือไม่ ถ้าคำตอบคือใช่ แสดงว่ามีเศษของบล็อกข้อมูลเหลืออยู่จะทำงานข้อต่อไป ถ้าคำตอบคือไม่ใช่ แสดงว่าข้อมูลหมดแล้วจะข้ามไปทำงานข้อ 17

14. ถ้า Xptr ชี้เกิน buffer แสดงว่าข้อมูลใน buffer หมดแล้วต้องอ่านข้อมูลเข้ามาใหม่

15. สร้างคีย์ย่อย Z เพื่อใช้เข้ารหัสลับ นำค่าใน IV มาทำงานตามขั้นตอนวิธี IDEA (อธิบายในหัวข้อที่ผ่านมา) ผลลัพธ์ที่ได้เก็บใน dum จากนั้นนำค่าใน dum มาเอ็กซ์คลูซีฟออร์กับเศษข้อมูลในบล็อกที่ Xptr ชี้อยู่ จากนั้นเพิ่ม Xptr อีก remain แล้วข้ามไปข้อ 17

16 เนื่องจากเป็นข้อมูลที่ไม่จำเป็นต้องถอดรหัสลับจึงอ่านข้อมูลส่วนแล้วจัดเก็บในแฟ้มข้อมูล output

17. เพิ่มลำดับที่ส่วน s อีก 1 เพื่อทำงานส่วนถัดไป แล้ววนกลับไปข้อ 4

18. เมื่อทำงานครบทุกส่วนแล้วจะเก็บข้อมูลที่ถอดรหัสลับใน buffer ลงในแฟ้มข้อมูล output แล้วออกจากการทำงาน

19. ตัวระบุส่วนไม่ถูกต้องจะลบแฟ้มข้อมูล output แล้วออกจากการทำงาน