การสร้างระบบรหัสลับแบบ อาร์เอสเอ

(RSA Cryptosystem Implementation)

นางสาวมุทิตา นุชอุดม

นางสาวอรวรรณ ลิ้วาทินแสนสุข

นางสาวอรอุมา สำราญผล

นายอานนท์ สุขเสถียรวงศ์

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

RSA Cryptosystem Implementation

Miss Nutita Nuch-udom

Miss Orawan Leevatinsansuk

Miss On-uma Sumranpol

Mr. Anon Sukstreinwong

A Special Project Submitted in Partial Fulfillment of the

Requirement for the Degree of Bachelor of Science

Department of Mathematics and Computer Sciences

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1992

หัวข้อปัญหาพิเศษ การสร้างระบบรหัสลับแบบ อาร์เอสเอ

RSA Cryptosystem Implementation

โดย นางสาวมุทิตา นุชอุดม

นางสาวอรวรรณ ลิ้วาทินแสนสุข

นางสาวอรอุมา สำราญผล

นายอานนท์ สุขเสถียรวงศ์

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์นัชรี เลิศวิจิตรศิลป์

อาจารย์วีระ บุญจริง

ภาควิชา คณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติให้นับปัญหาพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

(รศ.วิเชียร ศรีเสือขาม) หัวหน้าภาควิชา ฯ

คณะกรรมการปัญหาพิเศษ

(รศ.ดร.ไมตรี โพธิ์สุข) ประธานกรรมการ

(ผศ.พัชรินทร์ เหมโชติ) กรรมการ

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ การสร้างระบบรหัสลับแบบ อาร์เอสเอ

นักศึกษา นางสาวมุทิตา นุชอุดม
นางสาวอรวรรณ ลิ้วาทินแสนสุข
นางสาวอรอุมา สำราญผล
นายอานนท์ สุขเสถียรวงศ์

อาจารย์ที่ปรึกษา อาจารย์นัชรี เลิศวิจิตรศิลป์
อาจารย์วีระ บุญจริง

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์

ปีการศึกษา 2535

บทคัดย่อ

การทำปัญหาพิเศษนี้มีจุดมุ่งหมายเพื่อจัดสร้างระบบรหัสลับแบบ อาร์เอสเอ ในรูปของซอฟท์แวร์ หลังจากที่ได้ออกแบบระบบแล้ว ได้ทำการสร้างระบบบนไมโครคอมพิวเตอร์ พบว่าสามารถทำการเข้ารหัสและถอดรหัสได้ดี

Special Project Title RSA Cryptosystem Implementation

Name Miss Mutita Nuch-udom
Miss Orawan Leevatinsansuk
Miss On-uma Sumranpol
Mr. Anon Sukstreinwong

Special Project Advisor Mrs. Patcharee Lertwichitsil
Mr. Veera Boonjing

Department Mathematics and Computer Sciences

Academic Year 1992

Abstract

This study aims at RSA cryptosystem software implementation. After designing, the implementation made on microcomputer. As the result, it can encrypt and decrypt excellently.

กิตติกรรมประกาศ

คณะผู้จัดทำปัญหาพิเศษฉบับนี้ ขอขอบพระคุณอาจารย์ทุก ๆ ท่านที่ได้ช่วยให้คำแนะนำแนวทางต่าง ๆ ในการทำปัญหาพิเศษ โดยเฉพาะอย่างยิ่ง อาจารย์พัชรี เลิศวิจิตรศิลป์ ซึ่งเป็นอาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้กรุณาช่วยให้คำปรึกษา แนะนำ และ ค้นคว้าเอกสารต่าง ๆ ที่ใช้ประกอบการทำปัญหาพิเศษฉบับนี้แก่คณะผู้จัดทำ และอาจารย์วีระ บุญจริง ที่ได้ช่วยให้คำปรึกษาเกี่ยวกับแนวทางในการออกแบบซอฟท์แวร์

ขอขอบพระคุณคณะกรรมการทุกท่านที่ได้กรุณาสละเวลาอันมีค่าของท่าน เพื่อตรวจสอบปัญหาพิเศษฉบับนี้จนกระทั่งสำเร็จลงด้วยดี

ขอขอบคุณ เพื่อน ๆ พี่ ๆ และน้อง ๆ ทุกคนที่ช่วยให้กำลังใจมาโดยตลอด ท้ายสุดนี้ขอกราบขอบพระคุณ พ่อ แม่ ครู อาจารย์ และ ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญรูป

หน้า

สารบัญ

บทที่ 1

บทนำ

ความสำคัญและ ที่มาของปัญหา

ปัจจุบันมีการใช้คอมพิวเตอร์ในการจัดการข้อมูลและข่าวสารกัน อย่างแพร่หลาย ในทุกวงการ ทั้งในด้านการจัดเก็บและการจัดส่ง ในบางครั้งข้อมูลเป็นข้อมูลลับ การทำให้ข้อมูลเป็นข้อมูลลับ คือ การทำให้ผู้ไม่มีสิทธิไม่สามารถตีความข้อมูลที่ได้รับได้ วิชาการแขนงนี้รู้จักกันดี ในชื่อวิชารหัสลับ (cryptography)ซึ่งมีทฤษฎีที่ประยุกต์วิชาการด้านคณิตศาสตร์จำนวนมาก แต่อย่างไรก็ตามการสร้างระบบรหัสลับตามทฤษฎีในประเทศไทย ค่อนข้างจำกัด ปัญหาพิเศษนี้จึงได้จัดสร้างระบบรหัสลับ ในรูปของซอฟท์แวร์ที่คาดว่าสามารถนำไปประยุกต์ใช้กับงานข้อมูลลับได้อย่างแพร่หลายต่อไป

วัตถุประสงค์ของปัญหา

เพื่อศึกษาทฤษฎีรหัสลับต่าง ๆ และจัดสร้างระบบรหัสลับในรูปของซอฟท์แวร์

ขอบเขตของปัญหา

ระบบรหัสลับตามทฤษฎีมีจำนวนมาก แต่ปัญหาพิเศษนี้จะจัดสร้างระบบรหัสลับในรูปของซอฟท์แวร์ โดยใช้ระบบรหัสลับแบบ อาร์เอสเอ (Rivest-Shamir-Adelman Cryptosystem)

ขั้นตอนในการดำเนินงาน

1. ศึกษาทฤษฎีรหัสลับ
2. ออกแบบซอฟท์แวร์เพื่อการจัดทำระบบรหัสลับแบบ อาร์เอสเอ
3. จัดทำระบบรหัสลับแบบ อาร์เอสเอ
4. จัดทำคู่มือการใช้งานระบบรหัสลับแบบ อาร์เอสเอ
5. สรุปและเสนอแนะ

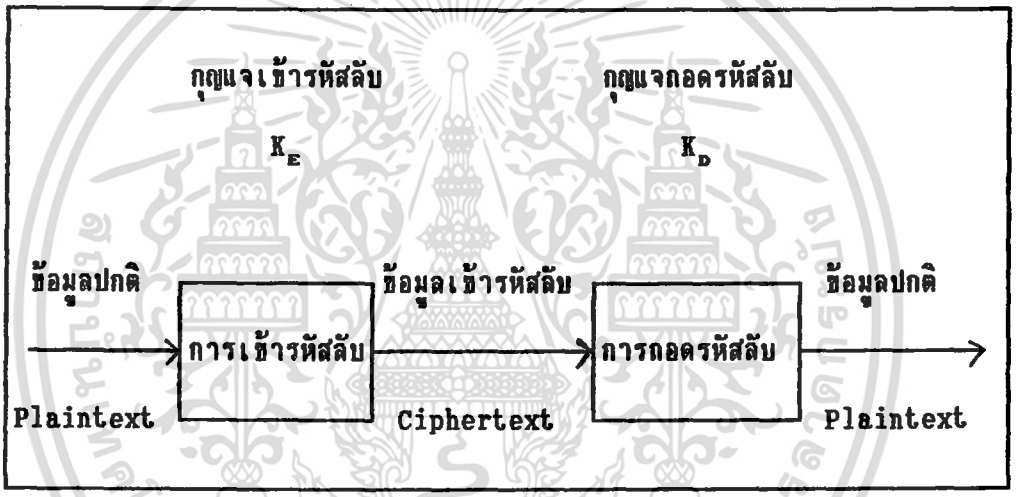
ประโยชน์ที่คาดว่าจะได้รับ

ระบบรหัสลับที่ได้ สามารถประยุกต์ใช้งานได้กับซอฟท์แวร์ทุกประเภทบนไมโครคอมพิวเตอร์ และสามารถขยายเพื่อนำไปใช้ในระบบที่ใหญ่ขึ้นได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวกับระบบรหัสลับแบบ อาร์เอสเอ

การเข้ารหัสลับ (encryption) เป็นกรรมวิธีการเข้ารหัสข้อมูลข่าวสาร ทำให้ความหมายของข่าวสารเดิมแปรเปลี่ยนไป ส่วน การถอดรหัสลับ (decryption) เป็นกรรมวิธีที่ตรงกันข้ามที่แปลงข่าวสารจากการเข้ารหัสลับให้กลับไปเป็นข้อมูลเดิม ในระบบที่ประกอบด้วยทั้งส่วนการเข้ารหัสลับและการถอดรหัสลับ ดังรูป 2.1

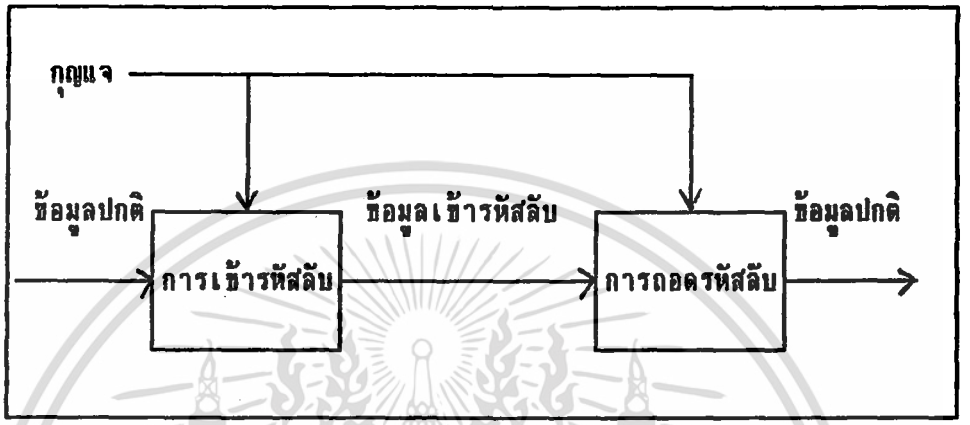


รูปที่ 2.1 ระบบการสร้างรหัสลับ

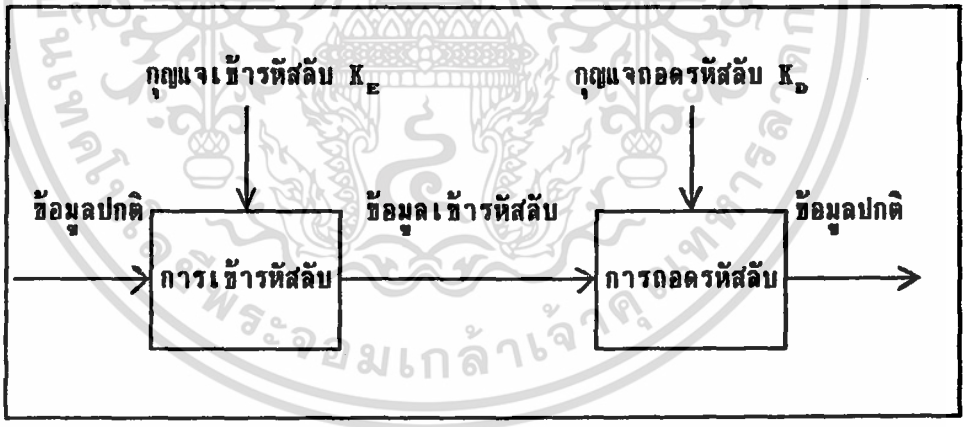
ข้อมูลข่าวสารปกติ (plaintext) เขียนย่อว่า P จะเป็นการเรียงต่อกันของตัวอักษรต่าง ๆ $P = [p_1, p_2, p_3, \ldots p_n]$ และข้อมูลที่ผ่านการเข้ารหัสลับ (ciphertext) จะเขียนได้เป็น $C = [c_1, c_2, c_3, \ldots, c_n]$ ขั้นตอนการเข้ารหัสลับก็คือ การแปลงระหว่างข้อมูลปกติไปเป็นข้อมูลรหัสลับ จะเขียนแทนด้วยสมการได้เป็น $C = E(P)$ ขณะเดียวกันขั้นตอนการถอดรหัสลับจะเขียนสมการได้เป็น $P = D(C)$ และระบบการสร้างรหัสลับจะเขียนเป็นสมการรวมได้เป็น $P = D(E(P))$

หลักการการเข้ารหัสลับ

หลักการการเข้ารหัสลับ แบ่งออกเป็น 2 ประเภท คือ การเข้ารหัสโดยใช้กุญแจในการเข้ารหัสและถอดรหัสเดียวกัน ดังรูป 2.2 ส่วนอีกประเภทหนึ่งใช้กุญแจในการเข้ารหัสและถอดรหัสต่างกัน ดังรูป 2.3 ซึ่งทั้งสองประเภทมีรายละเอียดดังนี้



รูปที่ 2.2 ระบบการสร้างรหัสลับด้วยกุญแจรหัสตัวเดียวกัน



รูปที่ 2.3 ระบบการสร้างรหัสลับด้วยกุญแจรหัสต่างกัน

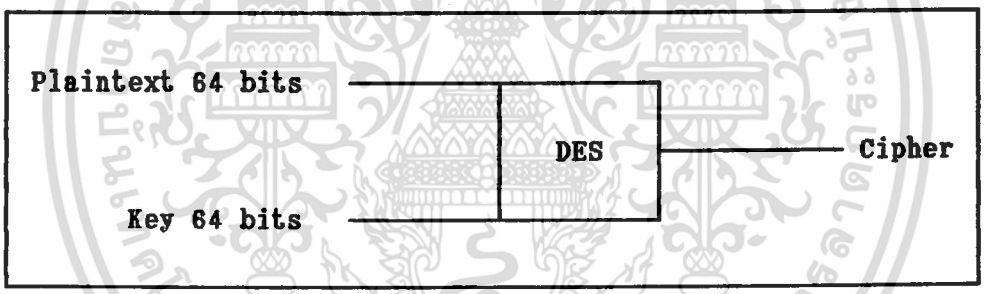
1. อัลกอริทึมกุญแจเดียวกัน (Symmetric Cryptographic algorithm)

การเข้ารหัสแบบนำกุญแจเข้า มาร่วมกับข้อมูลปกติเพื่อไปเป็นข้อมูลเข้ารหัสลับ กรณีนี้สมการสำหรับการเข้ารหัสลับเป็น $C = E(K,P)$ โดย กุญแจ จะเป็นค่าหรือขั้นตอนเฉพาะหนึ่ง ซึ่งสำหรับหลักการนี้การเข้ารหัสและถอดรหัสจะใช้กุญแจเดียวกัน และสมการ

สำหรับการถอดรหัสจะเขียนได้เป็น P = D(K,E(K,P)) ในการใช้กุญแจเดียวกันนี้ ทำให้มีโอกาสในการทราบกุญแจในการเข้ารหัสสูง ดังนั้น ความปลอดภัยของข้อมูลจึงมีน้อย ตัวอย่างการเข้ารหัสด้วยวิธีดังกล่าวนี้ ได้แก่ วิธีการเข้ารหัสแบบ ดีอีเอส (data encryption standard) โดยมีหลักการ ดังนี้

อัลกอริทึม ดีอีเอส ได้นำวิธีการพื้นฐานของการเข้ารหัสมาใช้ คือ วิธีการแทนที่ข้อมูล, วิธีการสลับเปลี่ยนตำแหน่งข้อมูล และ วิธีการมอดูโลสอง หรือ การเอกซ์คลูซีฟออร์ โดยที่การทำงานจะกระทำในระดับบิตของข้อมูล

โดยการเข้ารหัสของ ดีอีเอส นั้น ข้อมูลในการเข้ารหัสและถอดรหัส จะถูกแบ่งเป็นกลุ่ม ๆ ละ 64 บิต และมี กุญแจ สำหรับเข้ารหัส 64 บิต แต่ตัดเป็นพาริตี้บิต (parity bit) จำนวน 8 บิต ดังนั้นจะเหลือ 56 บิต และได้ผลลัพธ์เป็นข้อมูลที่เข้ารหัสแล้ว 64 บิต ดังรูป 2.4



รูปที่ 2.4 ระบบเข้ารหัสแบบ ดีอีเอส

การเข้ารหัสลับแบบ ดีอีเอส นี้ใช้หลักการ 2 แบบ คือ

1.)ข้อมูลที่ผ่านการเข้ารหัส (product cipher) ซึ่งใช้หลักการการสลับที่ และการแทนที่ของข้อมูล

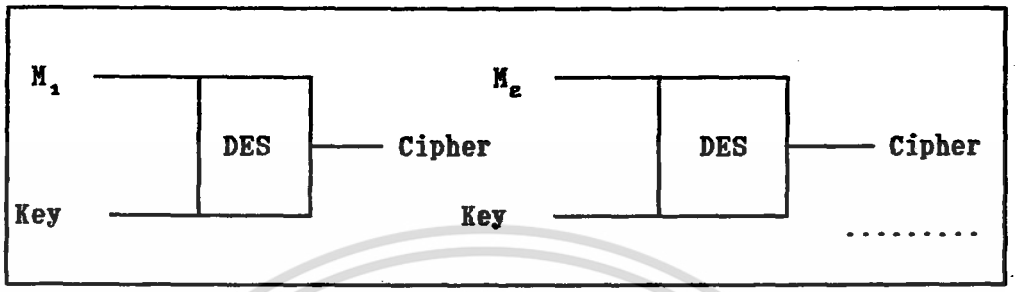
M = Message 12 bit

$C = E_k(M) = S_kP_kS_{k-1}P_{k-1}\ldots S_2P_2S_1P_1(M)$ โดยทำ k รอบ

2.)กลุ่มรหัสลับ (block cipher) ซึ่งการเข้ารหัสและการถอดรหัสจะกระทำเป็นกลุ่ม

$M = M_1, M_2, \ldots$; M คือ ข้อมูลที่จะเข้ารหัสเป็น กลุ่ม

โดยข้อมูลเข้า และ กุญแจ จะเข้าทำในอัลกอริทึม ทีละ กลุ่ม ซึ่งถ้าข้อมูลต่างกัน กุญแจ ก็จะต่างกันด้วย ดังรูป 2.5



รูปที่ 2.5 แสดงกลุ่มรหัสลับ

$$E_k(M) = E_k(M_1)\, E_k(M_2) \ldots\ldots\ldots$$

2. <u>อัลกอริทึมกุญแจต่างกัน (Asymmetric algorithm)</u>

ในการเข้ารหัสและถอดรหัส จะใช้ กุญแจต่างกัน โดย กุญแจ ที่ใช้ในการเข้ารหัส เรียกว่า กุญแจสาธารณะ (public key (K_E)) และกุญแจที่ใช้ในการถอดรหัสเรียกว่า กุญแจลับ (secret key (K_D)) โดยสมการสำหรับการเข้ารหัสเป็น $C = E(K_E, P)$ และสมการการถอดรหัสเป็น $P = D(K_D, E(K_E, P))$ เมื่อ K_E และ K_D แทนกุญแจ สำหรับการเข้ารหัสและการถอดรหัสตามลำดับ ซึ่งกุญแจลับจะต้องเก็บเป็นความลับ โดยจะได้มาจากการคำนวณของ กุญแจสาธารณะ ตัวอย่างการเข้ารหัสด้วยวิธีดังกล่าวนี้ได้แก่วิธีการเข้ารหัสแบบ อาร์เอสเอ ซึ่งหลักการและรายละเอียดต่าง ๆ จะกล่าวในลำดับต่อไป

การเข้ารหัสลับด้วยการใช้กุญแจจะช่วยเพิ่มความปลอดภัยให้กับข้อมูลมากขึ้น เพราะบุคคลภายนอกหรือนักโจรกรรมจะต้องรู้ค่าหรือขั้นตอนเฉพาะเสียก่อน จึงจะแกะข่าวสารนั้นได้ตามปกติ บุคคลที่ทำหน้าที่เข้ารหัสลับและทำการถอดรหัสลับจะเรียกว่า นักเข้ารหัสลับ (cryptographer) แต่บุคคลภายนอกที่พยายามจะเข้าไปถอดรหัสลับ โดยไม่ได้รับอนุญาตจะเรียกว่า นักวิเคราะห์รหัสลับ (cryptanalysis) ซึ่งอาจเป็นการพยายามถอดรหัสลับข่าวสารเดียว หรือพยายามหาขั้นตอนการเข้ารหัสลับเพื่อไปถอดรหัสลับของข่าวสารต่อๆไป

หรือพยายามศึกษาขั้นตอนการเข้ารหัสลับเพื่อหาจุดบกพร่องของวิธีต่าง ๆ การแกะวิธีการเข้ารหัสลับเป็นเรื่องที่เป็นไปได้มาก ถ้ามีเวลาและข้อมูลเพียงพอ ดังนั้นการเข้ารหัสลับที่ใช้ในปัจจุบันจึงค่อย ๆ ซับซ้อนมากขึ้น เพื่อป้องกันการเข้าไปแทรกแซงระบบ

ทฤษฎีทางคณิตศาสตร์ที่เกี่ยวข้อง

ระบบรหัสลับสร้างขึ้นภายใต้ทฤษฎีทางคณิตศาสตร์ ดังนี้

1. เลขคณิตมอดุลาร์ (Modular Arithmetic)

นิยาม 2.1 กำหนดให้ a,b เป็นจำนวนเต็ม และ n เป็น จำนวนเต็มบวก เรียก a เป็นคอนกรูเอนซ์กับ b มอดูโล n (a is congruent to b modulo n) ก็ต่อเมื่อ $n \mid (a-b)$ (หมายถึง n หาร (a-b) ลงตัว) ที่เขียนแทนด้วย $a \equiv b \pmod{n}$

จากนิยามการหารลงตัว เราจะได้ว่า $a \equiv b \pmod{n}$ ก็ต่อเมื่อ $a = b + kn$ สำหรับ k บางค่า

ถ้า $a \equiv b \pmod{n}$ จะเรียก b ว่าเศษเหลือ ของ a (mod n) เช่น $17 \equiv 5 \pmod{12}$ นั่นคือ 5 เป็นเศษเหลือของ 17 (mod 12)

กำหนดให้ $A = \{r_1, r_2, \ldots, r_n\}$ เป็นเซตของเศษเหลือทั้งหมด ใน มอดูโล n สำหรับจำนวนเต็ม a ใด ๆ จะหาค่า r_i สำหรับบางค่า $i \leq n$ ในเซต A เพียงค่าหนึ่งและค่าเดียวเท่านั้นที่ซึ่ง

$$a \equiv r_i \pmod{n}$$

ดังนั้นจะได้ว่า เซตของจำนวนเต็มที่เรียงติดต่อกัน n จำนวนเป็น ระบบเศษเหลือสมบูรณ์ มอดูโล n (Complete Residues System modulo n : C.R.S.)

โดยปกติแล้ว ระบบเศษเหลือสมบูรณ์ มอดูโล n ที่นิยมใช้กันมากก็มี

เซตของ $0,1,2,\ldots,n-1$

เซตของ $0,1,2,\ldots,n$

และเซตของ $-(\frac{n-1}{2}), -(\frac{n-1}{2})+1,\ldots,-1,0,1,\ldots,(\frac{n-1}{2})-1,(\frac{n-1}{2})$

(สำหรับ n ที่เป็นจำนวนคี่บวก) เช่นเมื่อ n = 5 จะได้ {-2,-1,0,1,2} เป็นเซตของระบบเศษเหลือสมบูรณ์ มอดูโล 5

จากความจริงที่ว่า จำนวน 0,1,2,...,n-1 เป็น ระบบเศษเหลือสมบูรณ์ มอดูโล n ทำให้สรุปต่อไปได้ว่า จำนวนเหล่านี้ เมื่อนำมาบวกหรือลบ หรือคูณกัน แล้วจะเป็นคอนกรูเอนซ์มอดูโล n กับจำนวนเต็มจำนวนเดียว ใน ระบบเศษเหลือสมบูรณ์ มอดูโล n นี้ จึงนำไปสู่แนวความคิดในการสร้าง เลขคณิตมอดูโล n (Arithmetic Modulo n) หรือบางครั้งเรียกว่า เลขคณิตมอดูลาร์

ดังนั้นจะได้ว่ากฎทางคณิตศาสตร์ที่เกี่ยวกับ การบวก และการคูณของจำนวนเต็ม ยังจริงอยู่ในเลขคณิตมอดูโล n เช่น

ถ้า $a \equiv b \pmod n$ และ $c \equiv d \pmod n$ แล้ว

$a+c \equiv b+d \pmod n$

$ac \equiv bd \pmod n$

$a+c \equiv c+a \pmod n$

$ac \equiv ca \pmod n$

$a+(b+c) \equiv (a+b)+c \pmod n$

$a(bc) \equiv (ab)c \pmod n$

$a(b+c) \equiv ab+ac \pmod n$

$(b+C)a \equiv ba+ca \pmod n$

อาจใช้วิธีการคัดเก้าออก (Casting out nines) มาทดสอบคุณสมบัติเบื้องต้นได้ คือเมื่อกำหนดให้ m เป็นจำนวนเต็มบวกจำนวนหนึ่ง แล้ว m จะเป็นคอนกรูเอนซ์มอดูโล 9 กับผลบวกของตัวเลขโดด (digit) ทั้งหมดที่ประกอบเป็น m

เช่น 46909818 หารด้วย 9 ลงตัว

เพราะว่า $46909818 = 4\times10^7+6\times10^6+9\times10^5+9\times10^3+8\times10^2+1\times10+8$

จาก $10 \equiv 1 \pmod 9$

$10^2 \equiv (10 \pmod 9)(10 \pmod 9)$

$$\equiv 1 \pmod 9$$

$$10^3 \equiv (10^2 \pmod 9)(10 \pmod 9)$$

$$\equiv 1 \pmod 9$$

.

.

.

เพราะฉะนั้น

$$46909818 = 4+6+9+8+1+8$$

$$= 45 = 4+5 = 9 \equiv 0 \pmod 9$$

ในทางปฏิบัติแล้ว ไม่จำเป็นต้องบวก 9 หรือ ผลรวมของจำนวนใดที่เป็น 9 เข้าไปด้วย นั่นคือ จำนวน 46909818 สามารถตัด 9,8,1 ออกได้ทันทีเลย จึงเหลือเพียง 4,6,8 ซึ่งมาคิดได้ผลเป็น $46909818 = 4+6+8 = 18 \equiv 0 \pmod 9$

นอกจากนี้ เลขคณิตมอดูโล n ยังใช้ได้กับเลขชี้กำลัง เช่น

$$3^{12} \equiv (3^2)^6 \pmod 7$$

$$\equiv (3^2 \pmod 7)^6$$

$$\equiv (2 \pmod 7)^6$$

$$\equiv 2^6 \pmod 7$$

$$= 64 \equiv 1 \pmod 7$$

และ

$$2^{5 \pmod 3} \bmod 3 \equiv 2^{2 \pmod 3} \bmod 3 \equiv 4 \bmod 3 \equiv 1 \pmod 3$$

แต่

$$2^5 \equiv 2 \pmod 3$$

2. ล็อกการิทึมเต็มหน่วย (Discrete Logarithm)

ถ้าเราต้องการทราบค่า $7^5 \pmod 9$ จะคำนวณได้จาก

$$7^2 = 49 \equiv 4 \pmod 9$$

$$7^4 = 4^2 = 16 \equiv 7 \pmod 9$$

$$7^5 = 7^2 \equiv 4 \pmod 9$$

$$7^5 \pmod 9 \equiv 4 \pmod 9$$

ในทางกลับกัน ถ้าต้องการทราบค่าเลขชี้กำลัง ใน เลขคณิตมอดูลาร์ นั่นคือ การหาจำนวน ล็อกการิทึมเต็มหน่วย หมายถึงต้องการที่จะหาค่า x เมื่อกำหนดสมการ

$$a^x \equiv b \pmod n$$

เช่นต้องการหาค่า x,y เมื่อกำหนดสมการดังต่อไปนี้

$$3^x \equiv 4 \pmod{13}$$

$$2^y \equiv 3 \pmod{13}$$

พิจารณา $3^1 \equiv 3$, $3^2 \equiv 9$, $3^3 \equiv 1$, $3^4 \equiv 3, \ldots \pmod{13}$

จะสังเกตโดยง่ายว่าสมการ $3^x \equiv 4 \pmod{13}$ ไม่มีค่า x ที่เป็นผลเฉลยของสมการ

พิจารณาสมการ $2^y \equiv 3 \pmod{13}$ จาก

$$2^1 \equiv 2 , 2^2 \equiv 4 , 2^3 \equiv 8 , 2^4 \equiv 3,$$

$$2^5 \equiv 6 , 2^6 \equiv 12, 2^7 \equiv 11, 2^8 \equiv 9,$$

$$2^9 \equiv 5 , 2^{10} \equiv 10, 2^{11} \equiv 7, 2^{12} \equiv 1 \pmod{13}$$

จะสังเกตว่า $2^4 \equiv 3 \pmod{13}$ ดังนั้นค่า y = 4

3.การคำนวณตัวผกผัน (Computing Inverse)

ไม่เฉพาะใน เลขคณิตจำนวนเต็ม(integer arithmetic)เท่านั้น ที่หาตัวผกผันได้ บางครั้งในเลขคณิตมอดูลาร์ ก็สามารถที่จะหาตัวผกผันได้เช่นกัน

ให้ $a \in \{0,1,\ldots,n-1\}$ $\exists x \in \{0,1,\ldots,n-1\}$ ที่ซึ่ง

$$ax \equiv 1 \pmod n$$

เช่น

$$3x7 = 21 \equiv 1 \pmod{10}$$

เราเขียนสัญลักษณ์ gcd(a,n) แทน ตัวหารร่วมมาก (greatest common divisor) ของ a และ n

เช่น gcd(6,3) = 3

gcd(6,5) = 1

<u>บทแทรก 2.1</u> ถ้า $\gcd(a,n) = 1$ ดังนั้น

$$ai \pmod n \neq aj \pmod n \qquad 0<i<j<n \ , \ i \neq j$$

<u>พิสูจน์</u> สมมุติว่า $ai \equiv aj \pmod n$

นั่นคือ $n \mid a(i-j)$ $\because n \nmid a$ $\therefore n \mid (i-j)$

ดังนั้น $i = j$ ซึ่งขัดแย้งกับข้อสมมติที่ว่า $0<i<j<n$

ดังนั้น $ai \pmod n$, $i = 0,1,\ldots,n-1$ เมื่อ $\gcd(a,n) = 1$ คือ ระบบเศษเหลือสมบูรณ์ มอดูโล n ของ $0,1,\ldots,n-1$ #

<u>ตัวอย่างที่ 2.1</u> ถ้า $a = 3$ และ $n = 7$ ดังนั้น

$3i \pmod 7$, $i = 0,1,2,\ldots,6$ คือ $0,3,2,6,5,1,4$

จะไม่จริงเมื่อ $\gcd(a,n) \neq 1$

เช่น $a = 2$, $n = 6$ แล้ว

$2i \pmod 6$, $i = 0,1,\ldots,5$ ได้ $0,2,4,0,2,4$

<u>ทฤษฎี 2.1</u>

ถ้า $\gcd(a,n) = 1$ แล้วจะหา a^{-1} เมื่อ $0<a^{-1}<n$ ซึ่ง

$$aa^{-1} \equiv 1 \pmod n$$

<u>พิสูจน์</u>

จากบทแทรก 2.1 จะได้ $ai \pmod n \in \{0,1,\ldots,n-1\}$ สำหรับ $0<i<n-1$

$\because 1 \in \{0,1,\ldots,n-1\}$

ดังนั้นจะมี $i \leq n-1$ ที่ซึ่ง $ai \equiv 1 \pmod n$ #

กำหนดให้ A เป็น ระบบเศษเหลือสมบูรณ์ มอดูโล n เมื่อสับเซต A' ของ A มีคุณสมบัติว่า

a' เป็นสมาชิกของ A' ก็ต่อเมื่อ a' เป็นสมาชิกของ A และ $\gcd(a',n)=1$

ก็จะเรียกว่า A' เป็น ระบบเศษเหลือลดรูป มอดูโล n (Reduced Residue System

modulo n : R.R.S.)

ดังนั้น $A' = \{a \in A \mid \gcd(a,n) = 1\}$

<u>ตัวอย่างที่ 2.2</u> ระบบเศษเหลือสมบูรณ์ มอดูโล 10 คือ $A = \{0,1,2,3,4,5,6,7,8,9\}$

มี $A' = \{1,3,7,9\}$ เป็น ระบบเศษเหลือลดรูป มอดูโล 10

กำหนดให้ A เป็นระบบเศษเหลือลดรูป มอดูโล n และ a เป็นจำนวนเต็มที่ $\gcd(a,n) = 1$ แล้ว a เป็นคอนกรูเอนซ์ มอดูโล n กับสมาชิกเพียงจำนวนเดียวใน A'

กำหนดให้ A' และ A'' เป็น ระบบเศษเหลือลดรูป มอดูโล n แล้ว สมาชิกของ A' และ A'' ต้องเท่ากัน

ถ้าให้ A' และ A'' แต่ละเซตมีสมาชิกเป็น k จำนวน เนื่องจากสมาชิกที่แตกต่างกันของ A'' เป็นคอนกรูเอนซ์มอดูโล n กับสมาชิกที่แตกต่างกันของ A'

ดังนั้น ใน มอดูโล n จะได้ว่าเซต A' และ A'' มีสมาชิกอย่างเดียวกัน (เมื่อไม่คิดในเรื่องลำดับที่)

<u>ตัวอย่างที่ 2.3</u> ระบบเศษเหลือสมบูรณ์ มอดูโล 6 คือ $\{0,1,2,3,4,5\}$

$R_1 = \{1,5\}$, $R_2 = \{-1,1\}$

$R_3 = \{7,-1\}$, $R_4 = \{19,-7\}$

เป็นเซตของ ระบบเศษเหลือลดรูป มอดูโล 6

กำหนดให้ n เป็นจำนวนเต็มที่ $n > 1$ จะเขียนแทนจำนวนสมาชิกของ ระบบเศษเหลือลดรูป มอดูโล n ด้วย $\phi(n)$ ซึ่งมีชื่อว่า ออยเลอร์ ϕ ฟังก์ชัน หรือ ออยเลอร์โทเทียนท์ ฟังก์ชัน(Euler's ϕ Function or Euler's Totient Function)

เนื่องจากจำนวนเต็มบวกทุกจำนวน ที่น้อยกว่าจำนวนเฉพาะ p จะเป็นจำนวนเฉพาะสัมพัทธ์กับ p ดังนั้น $\phi(p) = p-1$ (Relative prime)

และ $\phi(n)$ เป็น ฟังก์ชันผลคูณ (Multiplicative function) คือ

$$\phi(mn) = \phi(m)\phi(n)$$

ตัวอย่างที่ 2.4 $\phi(21) = \phi(3x7) = \phi(3)\phi(7) = (3-1)(7-1)$

$= 2x6 = 12$

เพราะฉะนั้น

$\phi(p^a) = p^a - p^{a-1} = (p-1)(p^{a-1})$ เมื่อ pเป็น จำนวนเฉพาะ

เมื่อ a เป็นจำนวนเต็มบวก

<u>ออยเลอร์โทเทียนท์ ฟังก์ชัน</u> คือ จำนวนของจำนวนต่างๆ ในเซต $\{1,2,\ldots,n\}$ ที่เป็นจำนวนเฉพาะสัมพัทธ์กับ n ซึ่งแสดงดังตาราง

ออยเลอร์โทเทียนท์ ฟังก์ชัน

n	Reduced set (R.R.S.)	$\phi(n)$
n prime	$\{1,2,\ldots,n-1\}$	$n-1$
n^2 (n prime)	$\{1,2,\ldots,n-1,n+1,\ldots,2n-1,2n+1,\ldots,n^2-1\}$	$n(n-1)$
. . .	. . .	. . .
n^r (n prime)	$\{1,2,\ldots,n^r-1$ - multiples of $n(n^r)\}$	$(n^r-1)-(n^{r-1}-1)$ $= n^{r-1}(n-1)$
pq (p,q prime)	$\{1,2,\ldots,pq-1$ - multiple of p - multiple of q$\}$	$(pq-1)-(q-1)-(p-1)$ $= (p-1)(q-1)$
. . . $\prod_{i=1}^{t} P_i^{e_i}$ (Pi primes)	. . .	$\prod_{i=1}^{t} P_i^{e_i-1}(P_i-1)$

<u>ทฤษฎี 2.2</u> ถ้า $\{a_1,a_2,\ldots,a_{\phi(m)}\}$ และ $\{b_1,b_2,\ldots,b_{\phi(m)}\}$ เป็น ระบบเศษเหลือลดรูป มอดูโล m

แล้วสมการ

$$\prod_{i=1}^{\phi(m)} a_i \equiv \prod_{i=1}^{\phi(m)} b_i \pmod m$$

<u>พิสูจน์</u> เพราะว่า a แต่ละตัวเป็นจำนวนเฉพาะสัมพัทธ์กับ m ดังนั้น จะคอนกรูเอนซ์กับ b ตัวใดตัวหนึ่ง และไม่ซ้ำกันคือ

$$a_1 \equiv b_{i1} \pmod m$$

$$a_2 \equiv b_{i2} \pmod m$$

$$\vdots$$

$$a_{\phi(m)} \equiv b_{i\phi(m)} \pmod m$$

เมื่อ $i_1,i_2,\ldots,i_{\phi(m)}$ เป็นจำนวนเต็ม $1,2,\ldots,\phi(m)$ ในบางลำดับ ดังนั้นเมื่อคูณทุกคอนกรูเอนซ์ก็จะได้ผลตามต้องการ #

<u>ทฤษฎี 2.3</u> กำหนดให้ gcd (a,n) = 1 ถ้าเซตของจำนวนเต็ม $a_1,\ldots,a_n$ เป็น ระบบเศษเหลือสมบูรณ์ มอดูโล n แล้ว สำหรับจำนวนเต็ม b ทุกจำนวน จะได้ว่า

$aa_1 + b, \ldots, aa_n + b$ เป็น ระบบเศษเหลือสมบูรณ์ มอดูโล n และยิ่งไปกว่านั้น ถ้า $\{a_1,\ldots,a_{\phi(n)}\}$ เป็น ระบบเศษเหลือลดรูป มอดูโล n แล้ว $\{aa_1,\ldots,aa_{\phi(n)}\}$ เป็น ระบบเศษเหลือลดรูป มอดูโล n

<u>ทฤษฎี 2.4</u> ทฤษฎีออยเลอร์ (Euler's theorem)

ถ้า gcd(a,m) = 1 แล้วจะได้ $a^{\phi(m)} \equiv 1 \pmod m$

<u>พิสูจน์</u>

ให้ $\{r_1,r_2,\ldots,r_{\phi(m)}\}$ เป็นระบบเศษเหลือลดรูป มอดูโล m

เพราะว่า (a,m) = 1 ดังนั้น $\{ar_1,ar_2,\ldots,ar_{\phi(m)}\}$ และเป็น

ระบบเศษเหลือลดรูปด้วยดังนั้นเราจะได้

$$\prod_{i=1}^{\phi(m)} ar_i \equiv \prod_{i=1}^{\phi(m)} r_i \pmod m$$

$$a^{\phi(m)} \prod_{i=1}^{\phi(m)} r_i \equiv \prod_{i=1}^{\phi(m)} r_i \pmod m$$

จากที่ $(r_i, m) = 1$ เราได้

$$\left(\prod_{i=1}^{\phi(m)} r_i, m \right) = 1$$

ดังนั้นเราตัดออกได้เป็น $a^{\phi(m)} \equiv 1 \pmod m$ #

<u>ทฤษฎี 2.5</u> ทฤษฎีเฟอร์แมท (Fermat's theorem)

ถ้า a เป็นจำนวนจริง และ p เป็นจำนวนเฉพาะ ซึ่ง $p \nmid a$ แล้ว

$p \mid (a^{p-1} - 1)$ นั่นคือ $a^{p-1} \equiv 1 \pmod p$

หรือ $a^{p-1} \pmod p = 1$

<u>ตัวอย่างที่ 2.5</u> การหา gcd(5,23) โดยใช้ ยูคลีเดียนอัลกอริทึม (Euclid's algorithm)

$23 = 4x5+3$

$5 = 1x3+2$

$3 = 1x2+1$

$2 = 2x1+0$

ดังนั้น gcd(5,23) = 1

ตัวอย่างที่ 2.6 การหา gcd(6,22) โดยใช้ ยูคลีเดียนอัลกอริทึม

$22 = 3x6+4$

$6 = 1x4+2$

$4 = 2x2+0$

ดังนั้น gcd(6,22) = 2

ซึ่งจะใช้หลักการนี้นำไปหาตัวผกผัน

ตัวอย่างที่ 2.7 จงหาตัวผกผันของ 5 (mod 23)

$\therefore 3 \equiv 23 \pmod 5$ $\quad 3 = (23) - 4x(5)$

$2 \equiv 5 \pmod 3$ $\quad 2 = (5) - 1x(3)$

$= 5 - 1x(23 - 4x5) = 5x5 - 1x23$

$1 \equiv 3 \pmod 2$ $\quad 1 = (3) - 1x(2)$

$= 23 - 4x5 - 1x(5x5 - 1x23)$

$= 2x23 - 9x5$

ดังนั้น

$1 \equiv -9x5 \pmod{23}$

และ

$-9 \equiv 14 \pmod{23}$ ดังนั้น $5^{-1} \equiv 14 \pmod{23}$

ตัวอย่างที่ 2.8 จงหาผลเฉลยของ $11x \equiv 1 \pmod{26}$

วิธีทำ พิจารณา (11,26) ; $11x-1 = k(26)$

$\therefore 4 \equiv 26 \pmod{11}$ $\quad 4 = 26 - 2x11$

$3 \equiv 11 \pmod 4$ $\quad 3 = 11 - 2x4$

$= 11 - 2(26-2x11)$

$= 5x11 - 2x26$

$1 \equiv 4 \pmod 3$ $\quad 1 = 4 - 3$

$= (26 - 2x11) - (5x11 -2x26)$

$= -7x11 + 3x26$

ดังนั้น $x = -7 \equiv 19 \pmod{26}$

และ $11^{-1} \equiv 19 \pmod{26}$

เราจะใช้หลักการในการหาตัวผกผันเพื่อแก้ปัญหา

$$ax \equiv b \pmod{n}$$

ขั้นแรกจะต้องหาตัวผกผัน

$$ay \equiv 1 \pmod{n}$$

จากนั้นจะต้องหา

$$x = yb$$

ตัวอย่างที่ 2.9 จงหาผลเฉลยของ $5x \equiv 9 \pmod{23}$

ขั้นแรกเราจะหา

$$5y \equiv 1 \pmod{23}$$

ดังนั้นเราจะได้ $y = 14$ และ $x = 14 \times 9 \equiv 11 \pmod{23}$

ทฤษฎี 2.6 ถ้า $g = \gcd(a,n)$ และ $g \mid b$ แล้ว $ax \equiv b \pmod{n}$ มีรากทั้งหมดคือ

$$x \equiv \left[\frac{b}{g}\right] x_o + t \left[\frac{n}{g}\right] \pmod{n} \quad ; t = 0,1,\ldots,g-1$$

เมื่อ x_o คือ รากของสมการ

$$\left[\frac{a}{g}\right] x \equiv 1 \left[\bmod \frac{n}{g}\right]$$

ส่วนกรณีอื่นไม่มีผลเฉลย

พิสูจน์

ถ้า $ax \equiv b \pmod{n}$ มีผลเฉลยในเซต $\{1,\ldots,n-1\}$

แล้ว $n \mid (ax-b)$ $\therefore b = ax-kn$ สำหรับบางค่า k ที่เป็นจำนวนเต็ม

$\because g \mid n$ และ $g \mid a$ $\therefore g \mid b$

จาก $ax \equiv b \pmod{n}$ และ $\gcd(a,n) = g \wedge g \mid b$

$$\therefore \frac{a}{g} x \equiv \frac{b}{g} \left[\bmod \frac{n}{g}\right] \qquad \text{----------(1)}$$

$$\therefore \left[\frac{a}{g}\right] x \equiv 1 \left[\bmod \frac{n}{g}\right] \qquad \text{----------(2)}$$

มีผลเฉลยเพียงชุดเดียวในเซต $\{1,\ldots, \underline{n}-1\}$

ให้ x_0 คือรากของสมการ (2) $\quad \therefore \frac{a}{g} x_0 \equiv 1 \left[\mod \frac{n}{g}\right]$

$\therefore$ ผลเฉลยของสมการ (1) คือ $x_1 \equiv \frac{b}{g} x_0 \left[\mod \frac{n}{g}\right]$

$$\frac{a}{g}x_1 - \frac{b}{g} = kn \qquad \text{----------(3)}$$

สำหรับบางค่า k ที่เป็นจำนวนเต็ม

คูณ g ลงใน (3) $\quad ax_1 - b = k'n$

ดังนั้น x_1 เป็นผลเฉลยของ $ax \equiv b \pmod n$ สำหรับ $x \in \{1,..,n-1\}$

ใดๆ ที่ซึ่ง $x \equiv x_1 \left[\mod \frac{n}{g}\right]$ จะเป็นผลเฉลยด้วย ดังนั้นผลเฉลยทั้งหมดคือ

$$x = x_1 + t\left[\frac{n}{g}\right] \quad ; t = 0,1,\ldots,g-1 \qquad \#$$

<u>ตัวอย่างที่ 2.10</u> สมมติว่าจะหาผลเฉลยของ $9x \equiv 6 \pmod{12}$

เราให้ $g = \gcd(9,12) = 3$

และ 3 หาร 6 ลงตัว จะได้ผลเฉลย 3 ค่า , ค่าแรกคือ $3x_1 \equiv 2 \pmod 4$

หาผลเฉลยที่ x_0 , $3x_0 \equiv 1 \pmod 4$ $\quad x_0 \in [1,3]$

จะได้ $x_0 = 3$ ดังนั้น $x_1 = 3x2 = 6 \equiv 2 \pmod 4$

$x = 2 + tx4$, $t = 0,1,2$

$x = 2,6,10 \quad (x \in [1,11])$

<u>ทฤษฎี 2.7</u> ให้ $p_1,\ldots,p_r$ เป็นจำนวนเฉพาะสัมพัทธ์แบบคู่ (pairwise relatively prime) ให้ $n = p_1p_2\ldots p_r$

แล้ว $f(x) \equiv 0 \pmod n$ ก็ต่อเมื่อ $f(x) \equiv 0 \pmod{p_i}$

ทฤษฎี 2.8 ทฤษฎีเศษเหลือไชนิสส์ (Chinese Remainder theorem : C.R.T.)

ให้ $p_1,..,p_r$ เป็นจำนวนเฉพาะสัมพัทธ์แบบคู่ เมื่อ $n = p_1p_2...p_r$

แล้วสมการ r สมการ ที่เขียนได้เป็น $x \equiv x_i \pmod{p_i}$; $i = 1,...,r$

มีรากร่วมเพียงรากเดียวใน มอดูโล n

พิสูจน์

สำหรับ i แต่ละตัว , $\gcd\left[p_i, \frac{n}{g}\right] = 1$

จะมี y_i ซึ่ง

$$\left[\frac{n}{p_i}\right] y_i \equiv 1 \pmod{p_i}$$

หรือ

$$\left[\frac{n}{p_i}\right] y_i \equiv 0 \pmod{p_j} \quad j \neq i \ , p_j \left| \frac{n}{p_i} \right.$$

ให้ $x = \left[\sum \left[\frac{n}{p_i}\right] y_i x_i\right] \pmod{n}$

ซึ่งจะมีรากร่วมใน มอดูโล n ∎

ตัวอย่างที่ 2.11 จงหาผลเฉลยของ $x \equiv 1 \pmod 5$ และ $x \equiv 10 \pmod{11}$

เพื่อหาผลเฉลย มอดูโล 55

วิธีทำ พิจารณา $\frac{55}{5} y_1 \equiv 1 \pmod 5$ หรือ $11y_1 \equiv 1 \pmod 5$

$1y_1 \equiv 1 \pmod 5$

$y_1 = 1$

ทำนองเดียวกันจะได้ $\frac{55}{11} y_2 \equiv 10 \pmod{11}$, $y_2 = 9$

จากโจทย์ได้ค่า $x_1=1$, $x_2=10$

ให้ $x = \frac{55}{5} y_1x_1 + \frac{55}{11} y_2x_2$

$= 11x1x1 + 5x9x10$

ดังนั้น $x = 11+10$

$\equiv 21 \pmod{55}$

ตัวอย่างที่ 2.12 จงหาตัวผกผันของ 7 (mod 65)

วิธีทำ

$$7x \equiv 1 \pmod{65}$$

สังเกตุว่า

$$65 = 5(13)$$

$$7x \equiv 1 \pmod{5}$$

$$7x \equiv 1 \pmod{13}$$

จะได้ว่า

$$x = x_1 \equiv 3 \pmod{5}$$

$$x = x_2 \equiv 2 \pmod{13}$$

จะหาค่า y_1, y_2 ซึ่ง

$\left[\frac{65}{5}\right] y_1 = 13y_1 \equiv 1 \pmod{5}$ ดังนั้น $y_1 = 2$

$\left[\frac{65}{13}\right] y_2 = 5y_2 \equiv 1 \pmod{13}$ ดังนั้น $y_2 = 8$

ให้ $x = \left[\frac{65}{5}\right] x_1y_1 + \left[\frac{65}{13}\right] x_2y_2$

$= 13x3x2 + 5x2x8$

$\equiv 28 \pmod{65}$

ดังนั้น $7^{-1} \equiv 28 \pmod{65}$

ระบบรหัสลับแบบ อาร์เอสเอ

ระบบรหัสลับแบบ อาร์เอสเอ ได้ถูกคิดค้นขึ้นโดย Rivest Shamirและ Adleman ดังที่กล่าวมาแล้วว่า ระบบรหัสนี้มีกุญแจที่ใช้ในการเข้ารหัสและถอดรหัสเป็นคนละตัวกัน เรียกว่า กุญแจสาธารณะ(public key) และกุญแจลับ(secret key)โดยกุญแจแต่ละตัวเป็นสมาชิกของเซตจำนวนจริง Z_n ซึ่ง $Z_n = \{1,...,N-1\}$ และ $N = pxq$ เมื่อ p และ q เป็นจำนวนเฉพาะ และเซต นี้สามารถทำการบวกและคูณ มอดูโล N ได้โดยใช้กุญแจสาธารณะ สำหรับ การเข้ารหัสจะทำโดยการแปลงข้อมูลให้เป็นรหัสลับ C โดยคู่อันดับของ กุญแจสาธารณะ (K_B) และ ข้อมูล(M) ดังสมการ

$$C = E_K(M) = E_B(M) \equiv M^e \pmod{N} \qquad \text{--------- (4)}$$

ในทางกลับกัน

การถอดรหัสจะทำการแปลงรหัสลับ(C) ให้กลับไปเป็นข้อมูลเดิมโดยใช้คู่อันดับของกุญแจลับ (k_B) และ รหัสลับ (C) ดังสมการ

$$M = D_k(C) = D_B(M) \equiv C^d \pmod{N} \qquad \text{--------- (5)}$$

ข้อมูลที่ถูกแปลงเป็นรหัสที่เหมาะสมจะสามารถทำให้เป็นข้อมูลเดิมหลังจากขบวนการถอดรหัส ดังนั้น

$$M = D_B(E_B(M)) \qquad \text{--------- (6)}$$

แทนสมการที่ 4 และ 5 ลงในสมการที่ 6 จะได้

$$(M^e)^d \equiv M \pmod{N} \qquad \text{--------- (7)}$$

เมื่อรู้ว่าค่า N เป็นจำนวนเฉพาะ คอนกรูเอ็นซ์ ในสมการ (7) ควรจะมีผลเฉลยก็ต่อเมื่อ $K_Bk_B \equiv 1 \pmod{N-1}$

ในกรณีนี้ เมื่อ $N = pxq$ (p,q เป็นจำนวนเฉพาะ) จะมีผลเฉลยก็ต่อเมื่อ

$$K_Bk_B \equiv 1 \pmod{\varphi(N)} \qquad \text{--------- (8)}$$

เมื่อ $\varphi(N) = lcm(p-1,q-1)$, lcm = least common multiple (ค.ร.น)

ผู้รับสามารถที่จะเลือกสุ่มค่าของ กุญแจสาธารณะ (K_B) ได้ในขณะที่ กุญแจลับ (k_B) จะสามารถหาได้โดยการคำนวณในสมการ (8) และเป็นการง่ายขึ้นสำหรับผู้รับ คือเมื่อรู้ค่า p และ q ก็สามารถหาค่าของ $\varphi(N)$ ได้

สำหรับการหาคู่อันดับของ กุญแจสาธารณะ และ กุญแจลับ จะใช้หลักการของยูคลีเดียนอัลกอริทึม โดยการหาค่า ห.ร.ม ของจำนวนเต็ม (a,b) ซึ่ง

$$s_{-1} = 1 \ , \ t_{-1} = 0 \ , \ r_{-1} = a$$

$$s_0 = 0 \ , \ t_0 = 1 \ , \ r_0 = b$$

$$s_i = s_{i-2} - q_{i-1}s_{i-1}$$

$$t_i = t_{i-2} - q_{i-1}t_{i-1}$$

$$r_i = r_{i-2} - q_{i-1}r_{i-1}$$

เมื่อ $q_{i-1} = [\frac{r_{i-2}}{r_{i-1}}]$

r_i จะอยู่ในรูปของ ลำดับเศษเหลือ (remainder sequence) สำหรับการคำนวณหา ห.ร.ม r_i ตัวสุดท้ายที่ไม่เป็น 0 คือ ห.ร.ม และ

$$s_i r_{-1} + t_i r_0 = r_i \ ; \ i = 0,1,..$$

จากการพิสูจน์ดังนั้น

$$r_{i+1} = r_{i-1} - q_i r_i = s_{i-1}r_{i-1} + t_{i-1}r_0 - q_i(s_i r_{-1} + t_i r_0)$$

$$= (s_{i-1} - q_i s_i)r_{-1} + (t_{i-1} - q_i t_i)r_0 = s_{i+1}r_{-1} + t_{i+1}r_0$$

ดังนั้น เมื่อได้ค่าของ กุญแจสาธารณะ(K_B) และ $\phi(N)$ ผู้รับจะสามารถคำนวณ ห.ร.ม ของ (K_B, $\phi(N)$) ได้

ถ้า $r_{i+1} = 1$ จะได้

$$s_{i+1}K_B + t_{i+1}\phi(N) = r_{i+1} = 1$$

สุดท้ายเราจะได้ กุญแจลับ (k_B) = s_{i+1}

<u>ตัวอย่างที่ 2.13</u> สมมติ K_B ถูกเลือกมามีค่าเท่ากับ 21 ขณะที่ $\phi(N)$ = 34 จะหากุญแจลับ (k_B) จากอัลกอริทึม

s_{-1}, t_{-1}, r_{-1} ==> 1,0,34 = $\phi(N)$

s_0, t_0, r_0 ==> 0,1,21 = K_B

s_1, t_1, r_1 ==> 1,-1,13

s_2, t_2, r_2 ==> -1,2,8

s_3, t_3, r_3 ==> 2,-3,5

s_4, t_4, r_4 ==> -3,5,3

s_5, t_5, r_5 ==> 5,-8,2

s_6, t_6, r_6 ==> -8,13,1

จากแถวสุดท้าย

$$(-8)(34)+(13)(21) = 1$$

สามารถเขียนใหม่ได้ดังนี้

$$(13)(21) \equiv 1 \pmod{34}$$

ดังนั้น กุญแจลับ $(k_B) = 13$

โดยสรุป จะสังเกตว่า ผู้รับ B สามารถสร้างระบบโดยป้องกันให้ กุญแจลับ กับ คู่อันดับ (p,q) ซึ่งจะให้ค่า N เป็นความลับ และให้ จำนวนเต็ม N กับ กุญแจสาธารณะ ไม่เป็นความลับ สามารถเปิดเผยได้

<u>ตัวอย่างที่ 2.14</u> ให้ p = 5, q = 7 คำนวณ $\phi(N) = \text{lcm}(4,6) = 12$ กุญแจสาธารณะ (K_B) ที่สุ่มได้เท่ากับ 17 จะคำนวณหาค่า กุญแจลับ (k_B) โดยใช้สมการที่ 8 จะได้

$$17k_B \equiv 1 \pmod{12} \quad ; k_B = 5$$

คู่ลำดับ $(N = 35, K_B = 17)$ เมื่อถูกส่งผ่านไป จะสามารถเข้ารหัสได้

$$C = M^{K_B} = 33^{17} \equiv 3 \pmod{35}$$

และ

$$M = C^{k_B} = 3^5 \equiv 33 \pmod{35}$$

บทที่ 3

การออกแบบระบบและการสร้างระบบรหัสลับแบบ อาร์เอสเอ

การออกแบบระบบ

การออกแบบระบบรหัสลับแบบ อาร์เอสเอ นี้สามารถทำงานได้ดีภายใต้ข้อจำกัดที่ว่าทำงานได้เฉพาะบนไมโครคอมพิวเตอร์ที่มีประสิทธิภาพสูงและเป็น จอภาพสี และได้แบ่งการทำงานของระบบออกเป็น 3 ส่วนใหญ่ ๆ คือ ส่วนติดต่อผู้ใช้ ส่วนการเข้ารหัสและส่วนของการถอดรหัส

การออกแบบส่วนของการเข้ารหัสและการถอดรหัสทำโดยใช้อัลกอริทึมของ อาร์เอสเอ ส่วนการออกแบบในส่วนติดต่อกับผู้ใช้มีการแบ่งหน้าจอออกเป็น 3 ส่วน คือ ส่วนการเข้ารหัส ส่วนการถอดรหัส และส่วนการออกจากระบบ สำหรับส่วนแรก ส่วนการเข้ารหัสแสดงหน้าจอได้ดังรูป 3.1 จากรูปเห็นว่าการเข้ารหัสได้แบ่งออกเป็น 3 ส่วนย่อยคือส่วนไฟล์ (file) เป็นการเลือกไฟล์ที่จะเข้ารหัส ส่วนรหัสผ่าน (password) เป็นการใส่รหัสผ่านเพื่อนำรหัสนี้ไปแปลงเป็นกุญแจสาธารณะที่ใช้ในการเข้ารหัส และส่วนการทำงานคือการเข้ารหัส (run) การเข้ารหัสจะทำงานได้ก็ต่อเมื่อใส่ข้อมูลของ 2 ส่วนบนถูกต้อง ส่วนที่สอง การถอดรหัส แสดงหน้าจอได้ดังรูป 3.2 ซึ่งได้มีการแบ่งออกเป็น 3 ส่วนย่อยเช่นกัน คือต้องมีการเลือกไฟล์ที่ต้องการถอดรหัสและใส่รหัสผ่าน สำหรับการถอดรหัสจะทำงานได้ก็ต่อเมื่อเลือกไฟล์และใส่รหัสผ่านถูกต้อง และส่วนสุดท้ายแสดงหน้าจอดังรูป 3.3 เมื่อต้องการออกจากระบบจะมาทำงานยังส่วนนี้

รูป 3.1 ส่วนการเข้ารหัส

รูปที่ 3.2 ส่วนการถอดรหัส

รูปที่ 3.3 ส่วนการออกจากระบบ

การสร้างระบบ

ส่วนของขบวนการการเข้ารหัสและถอดรหัสจะมีอัลกอริทึมที่เกี่ยวข้องดังนี้

1. อัลกอริทึมรับรหัสผ่านจากหน้าจอและหาค่ากุญแจสาธารณะ
2. อัลกอริทึมแปลงค่ารหัสผ่านและเขียนลงไฟล์ประวัติเพื่อนำไปใช้ในการถอดรหัส
3. อัลกอริทึมเข้ารหัส
4. อัลกอริทึมแปลงชื่อไฟล์และเขียนลงไฟล์ประวัติเพื่อนำไปใช้ในการถอดรหัส
5. อัลกอริทึมหากุญแจลับ
6. อัลกอริทึมเปรียบเทียบรหัสผ่านและชื่อไฟล์ที่ถอดรหัสกับไฟล์ประวัติ
7. อัลกอริทึมถอดรหัส

อัลกอริทึมแต่ละอัลกอริทึมแสดงรายละเอียดได้ดังนี้

1. อัลกอริทึมรับรหัสผ่านจากหน้าจอและหาค่ากุญแจสาธารณะ

   1) รับค่า รหัสผ่านจากหน้าจอ : K[I] ; 1<I<20

   2) ที่ 1<I<5 K[I] = K[I]+3

   3) K[1] = K[4]

       K[2] = K[1]

       K[3] = K[3]

       K[4] = K[5]

       K[5] = K[2]

   4) แปลง K[I] ; 1<I<5 จาก ตัวอักษรเป็นตัวเลข

       K[I] ==> KK[I]

   5) K_E = (KK[1]KK[2] + KK[3]) (mod (KK[4] + KK[5]))

   6) ถ้า K_E ไม่เป็น เลขจำนวนเฉพาะ เพิ่มค่า K_E ขึ้นทีละ 1 จนกระทั่ง K_E เป็น เลขจำนวนเฉพาะ ซึ่ง K_E นี้คือ กุญแจสาธารณะ (public key)

อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*****************************************************/
/***          Find Public key (Algorithm 1)          **/
/*****************************************************/
change()
   {
    int i,val,temp[5],cha[5],x[5],y[2];

       for(i=0;i<5;i++)
       {
         temp[i] = temp[i] + 3;
          if (temp[i] > '~')
             temp[i] -= 95;
       }
         cha[0]=temp[0];
         temp[0]=temp[3];
         cha[1]=temp[1];
         temp[1]=cha[0];
         temp[3]=temp[4];
         temp[4]=cha[1];
         for(i=0;i<5;i++)
          {
            x[i] = Int(temp[i]);
          }
   y[0] = x[0]+x[1]+x[2];
   y[1] = x[3]+x[4];
   val = main_exp(y[0],1,y[1]);
   key[0] = main_prime(val);  /* key[0] is Public Key */
}
/*****************************************************/
```

2. อัลกอริทึมแปลงค่ารหัสผ่าน และเขียนลงไฟล์ประวัติเพื่อนำไปใช้ในการถอดรหัส

1) แปลงรหัสผ่านที่รับจากหน้าจอของอัลกอริทึมที่ 1 : K[I] ; 1<I<5 จากตัวอักษรเป็นตัวเลข K[I] ==> K1[I] ; 1<I<5

2) K1[I] = K1[I] + I ; 1<I<5 และแปลงตัวเลขกลับไปเป็นตัวอักษร K1[I] ==> K[I] และเขียน K[I] ลงไฟล์ประวัติ ทำซ้ำจนกระทั่ง I = 5 อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*********************************************************/
/**          change key and write to  History file      **/
/**                for check in encoding                **/
/**                    (Algorithm 2)                    **/
/*********************************************************/

change_key()
{
     int xx,i;
       for(i=0;i<5;i++)
       {
          xx = Int(key3[i]) + i+1;
          if (xx  > 95)
             xx -= 95;
          key3[i] = Char(xx);
          putc(key[3],f1);  /* f1 = History file */
        }

}

/*********************************************************/
```

3. อัลกอริทึมการเข้ารหัส

1) สุ่ม p , q ที่เป็น เลขจำนวนเฉพาะ

2) N = p x q

3) แปลง M[I] (ข้อมูลที่เข้ารหัสตัวที่ I ; I = 1,2,...) ให้เป็นตัวเลข

4) C[I] = $M[I]^{K_E}$ (mod N) ; K_E คือ กุญแจสาธารณะ และ C[I] คือ รหัสลับ และเขียน C[I] ลงไฟล์ใหม่ซึ่งใช้ชื่อเดิมแต่เปลี่ยนนามสกุลเป็น .ENC ทำซ้ำจนกระทั่งจบ file

5) เขียน N ลงในไฟล์ประวัติ

อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*********************************************************/
/**          Encryption  (Algorithm 3)                  **/
/*********************************************************/
find_key(int key[2])
{
int i,N,n,p,q,r[2],key[2];

 randomize();
        for (i=0;i<2;i++)
          {
            do
               r[i] = rand();
            while(r[i]>15);
            if(r[i] == 0)
            r[i] = r[i] + 2;
            if(r[i] == 1)
            r[i] = r[i] + 1;
            r[i] = main_prime(r[i]);
              /*random p and q such that prime number  */
            }
    key[2] = r[0]*r[1];  /* the value of N */
    p = r[0] -1;
    q = r[1] -1;
    n = main_lcm(p,q); /* the value of  (N) */
    key[1]  = main_inverse(key[0],n); /* The value of secret Key */
    Inv_Key = Char(key[1]);
    return(key[2]);
    if(flag_inv == 'N')     /* have inverse */
       {
         xxx[h-65] = main_exp(h,key[0],key[2]);
         dd[h-65]  = main_exp(xxx[h-65],key[1],key[2]);
         if(dd[h-65] != h)
           {
              h = 65;
              flag4 = 'N';
           }
       }
/*********************************************************/
```

```
/*****************************************************/
/**            Function main_exp                    **/
/**     fastexp - return a^z (mod n)                **/
/*****************************************************/
#include <conio.h>
#include <math.h>
#include <stdio.h>
extern int main_exp(int a,int z,int n);
int  main_exp(a,z,n)
  {
     int x = 1;
       while (z)
          {
            while (!(z % 2))
              {
                z /= 2;
                a = ((a%n)*(a%n)) % n;
              }
            z--;
            x = ((x % n)* (a % n)) % n;
          }
       return (x);
   }
/*****************************************************/
/*****************************************************/
/**                Function  main_lcm                 **/
/**  For find least common multiple (lcm) of (a,b)    **/
/*****************************************************/

int main_lcm(int a,int b)
{
 float k1,k2;
 int  i=2,m=1,d,great;
    if( a>b)
       great = a;
    else  great = b;
  while (i <= great)
  {
   k1 = (float)a/(float)i - a/i;
   k2 = (float)b/(float)i - b/i;
   if( k1> 0 || k2 > 0) i++;
   else
    {
     m = m*i;
     a= a/i;
     b= b/i;
     i++;
    }
  }
  m= m*a*b;
  return(m);
}
/*****************************************************/
```

```
/************************************************/
/**         Function   main_prime              **/
/**        For find prime number               **/
/************************************************/

int main_prime(int a)
{
 int k = 0;

 while(k == 0)
 {
  k = prime(a);
  a++;
  }
  return(--a);
}

int prime(int a)
{
 int flag4 = 0;
 float k1;
 int  i=2,great;
 great = a;
 if(a != 1 )
 {

  while (i < great)
  {
   k1 = (float)a/(float)i - a/i;
   if( k1> 0 ) i++;
   else
   {
    flag4 = 0;
    i = great+1 ;
   }

  }
  }
     if (i <= great)
     flag4 = 1;
     if (a == 1)
     flag4 = 1;
     return(flag4);
}
/************************************************/
```

4. อัลกอริทึมแปลงชื่อไฟล์และเขียนลงไฟล์ประวัติเพื่อนำไปใช้ในการถอดรหัส

1) เมื่อรับชื่อไฟล์ที่ต้องการเข้ารหัสจากหน้าจอ M[I]; 1<I<15 จะแปลงชื่อไฟล์โดยแปลงตัวอักษรให้เป็น ตัวเลข M[I] ==> MM[I]

2) MM[I] = MM[I] + I ; 1<I<15 และแปลง ตัวเลขให้เป็นตัวอักษร MM[I] ==> M[I] จากนั้นเขียน M[I] ลงไฟล์ประวัติ ทำซ้ำจนกระทั่ง I = 15 อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*************************************************/
/**    Change file name and write to History file  **/
/**            For check in decoding               **/
/**              (Algorithm 4)                     **/
/*************************************************/
Write_File(char kk[15])
{
  int xx;
  int j,i;
  for(i=0;i < 15;i++)
    if (kk[i] > '~')
      kk[i] -= 95;
      j=strlen(kk)-4;
    for(i=0;i< j;i++)
     {
       xx    = Int(kk[i]) + i+1;
       if(xx > 95) xx -= 95;
         kk[i] = Char(xx);
       putc(kk[i],f1);
      }
   for(i=j;i< j+4;i++)
        putc(kk[i],f1);
   for(i= j+4;i < 15;i++)
        putc(' ',f1);  /* Write 15 charector to file History.txt*/
}
/*************************************************/
```

5. อัลกอริทึมหากุญแจลับ (Secret key)

1) $\phi(n) = LCM\ [(p-1),(q-1)]$; LCM = ค.ร.น

2) $K_E \times K_D = 1 (\mathrm{mod}\ \phi(n))$

3) กุญแจลับ = K_D

4) แปลงค่าของกุญแจลับโดยแปลงตัวเลขให้เป็นตัวอักษร K_D ==> KK_D

5) เขียนกุญแจลับที่แปลงแล้ว ลงไฟล์ประวัติ

อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*************************************************/
/**      Find  Secret key which is key[1] in    **/
/**   Algorithm 3 by function main_inverse      **/
/**           ( Algorithm 5 )                   **/
/*************************************************/
/*************************************************/
/*                Function   main_inverse        */
/*      For Inversing  - return ax mod n = 1     */
/*************************************************/
#define j  100
int main_inverse(int a,int n)
  {
  int g[j],u[j],v[j],i,x,y,inv;
  flag_inv =  'Y';
      if(check(a,n) == 0)
          {
           flag_inv = 'Y';          /* not have inverse */
           return(0);
           }
        else flag_inv = 'N';
      g[0] = n;
      g[1] = a;
      u[0] = 1;
      u[1] = 0;
      v[0] = 0;
      v[1] = 1;
       i  = 1;
      while ( g[i] !=0   )
       {
          y = g[i-1] / g[i];
          g[i+1] = g[i-1] - (y * g[i]);
          u[i+1] = u[i-1] - (y * u[i]);
          v[i+1] = v[i-1] - (y * v[i]);
          i = i+1;
        }
       x = v[i-1];
       if (x >= 0)
          inv = x;
       else
          inv = x+n;
       return(inv);
   }

/************************/
int check(int a,int n)
 {
  int  k = 1;
  while(k  != 0 )
   {
    k = n % a;
    n = a;
    a = k;
    if(k == 1 ) return(k);
   }
 if(k == 0) return(0);
 /*if k = 0 not have inverse */
 }

/*************************************************/
```

6. อัลกอริทึมเปรียบเทียบรหัสผ่านและชื่อไฟล์ที่ถอดรหัสกับไฟล์ประวัติ

1) รับค่ารหัสผ่านและชื่อไฟล์จากหน้าจอในส่วนของการถอดรหัส

2) แปลงรหัสผ่านตามอัลกอริทึมที่ 1 และแปลงชื่อไฟล์ตามอัลกอริทึมที่ 4

3) นำรหัสผ่านและชื่อไฟล์ที่ได้แปลงแล้วไปค้นหาในไฟล์ประวัติ ถ้าพบไปอัลกอริทึมที่ 7 ถ้าไม่พบแสดงข้อความบอกข้อผิดพลาด

อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*************************************************/
/**         Compare file and key which decode     **/
/**        with file and key in history file      **/
/**                ( Algorithm 6  )               **/
/*************************************************/

  if(( f3 = fopen("A:History.txt","r")) == NULL)
    {
      printf(" can't open file A:History.txt");
      exit(0);
    }
 while((strcmp(Fi,File1) != 0))
 {
 fscanf(f3,"%s%s%s%d",&Fi,&ke,&inv,&N);
    j=strlen(Fi)-4;
    for(i=0;i< j;i++)
     {
       Fi[i] = (Fi[i]) - i;
       if(Fi[i] < 0) Fi[i] += 95;
     }
 }
 inverse_value = inv-31; /* change inverse to integer */
 strcpy(bv,key1);
 for(i =0;i< 5;i++)
 {
   ke[i] = ke[i]-i;
   if(ke[i] < 0) ke[i] += 95;
  }
 /*************************************************/
```

7. อัลกอริทึมการถอดรหัส

1) อ่านค่า N และ KK_D จากไฟล์ประวัติ

2) แปลงค่า KK_D จากตัวอักษรเป็นตัวเลข KK_D ==> K_D

3) M[I] = C[I]^{K_D} (mod N) ; C[I] คือ ข้อมูลที่เข้ารหัสตัวที่ I เขียน M[I] ลงไฟล์ใหม่ซึ่งมีชื่อเดิมและเปลี่ยนนามสกุลเป็น.CRY ทำซ้ำจนกระทั่งจบไฟล์ ซึ่งจะได้ข้อความเดิมก่อนเข้ารหัส

อัลกอริทึมนี้เขียนเป็นโปรแกรมได้ดังนี้

```
/*************************************************/
/**       Decoding  (Algorithm 7)               **/
/*************************************************/
 if(strcmp(bv,ke)==0)
 {
  while((ch = getc(f2)) != EOF)
    {
     if( flag1 == 'Y')
        {
         leng = INT(ch);flag1 = 'N';
         k = leng-1;
        }
      else
        {
          oldvalue = INT(ch)*pow(10,k--);
          value = oldvalue + value;
          if(k< 0)
          {
          flag1 = 'Y';
          oldvalue = 0; /* value is ciphertext */
          m = main_exp(value,inverse_value,N);
                /* inverse_value is secert key */
          M = m+63; /* M is plaintext */
          putc(M,f1);
          value = 0;
          }
        }
    }
/*************************************************/
```

บทที่ 4

ผลการทำงานของระบบ

ข้อจำกัดการใช้งานของระบบอาร์เอสเอ

1. รหัสผ่าน ที่ใส่ในการเข้ารหัสต้องมีความยาวอย่างน้อย 5 ตัวอักษร และไม่เกิน 20 ตัวอักษร

2. ข้อมูลที่จะเข้ารหัสจะต้องเป็นไฟล์ชนิด ไฟล์ข้อความ (text file) เท่านั้น

3. ข้อมูลที่เข้ารหัส และ รหัสผ่านที่ใช้ในการเข้ารหัส จะอยู่ในช่วงเลขฐานสิบ ตั้งแต่ 32 - 122 ของตารางรหัสแอสกี้ ดังนั้นสัญลักษณ์นอกจากนี้จะไม่คำนึงถึงในการเข้ารหัส

4. ฮาร์ดแวร์ที่ใช้กับระบบนี้มีลักษณะ ซีพียู(cpu) รุ่น 286 ขึ้นไป ฮาร์ดดิสก์ 20 เมกกะไบต์ขึ้นไป จอภาพสี วีจีเอ (vga)

ผลการทำงานของระบบ

ระบบรหัสลับแบบอาร์เอสเอ ที่ได้สร้างขึ้นนี้มีความสามารถในการเข้ารหัส โดยจะทำการแปลงข้อมูลให้เป็นรหัสลับ และสามารถถอดรหัสลับกลับไปเป็นข้อมูลเดิมได้ กุญแจที่ใช้ในการเข้ารหัสและถอดรหัสต่างกัน และจากการทดลองใช้ระบบนี้ พบว่า

1. เมื่อสุ่มค่าของ p และ q ได้เหมาะสมสำหรับการหา กุญแจลับ จะทำให้การทำงานของระบบเป็นไปอย่างรวดเร็ว แต่ถ้า ค่าของ p และ q ไม่เหมาะสมคือ ไม่สามารถหา กุญแจลับที่เหมาะสมได้ จะทำให้การทำงานของระบบช้าลง เนื่องจาก ต้องทำการสุ่มค่า p และ q ใหม่ จนกว่าจะได้ค่า p และ q ที่เหมาะสม

2. การทำงานของระบบมีการคำนวณตัวเลขทางคณิตศาสตร์ และทำซ้ำกันหลาย ๆ รอบ ดังนั้น จึงต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง จึงจะทำให้ระบบทำงานได้อย่างรวดเร็ว

3. การทำงานของระบบรหัสลับแบบ อาร์เอสเอ นี้จะทำงานได้ช้าเนื่องจากมีการคำนวณทางคณิตศาสตร์มาก

บทที่ 5

บทสรุปและข้อเสนอแนะ

สรุปผลการทำปัญหาพิเศษ

การทำปัญหาพิเศษนี้มีจุดมุ่งหมายเพื่อจัดสร้างระบบรหัสลับแบบ อาร์เอสเอ ในรูปของซอฟท์แวร์ หลังจากที่ได้ออกแบบระบบแล้ว ได้ทำการสร้างระบบบนไมโครคอมพิวเตอร์ พบว่าระบบสามารถทำงานได้ดี โดยสามารถแปลงข้อความปกติให้เป็นรหัสลับและแปลงรหัสลับกลับให้เป็นข้อความเดิมได้

ปัญหาและข้อเสนอแนะ

การจัดทำระบบนี้ได้พบปัญหาต่าง ๆ ดังนี้

1. การหาค่ายกกำลังของ มอดูโล จะได้ผลลัพธ์ผิดพลาด เมื่อตัวเลขในการคำนวณมาก

2. การหาค่าตัวผกผันจะมีบางค่าที่ไม่สามารถหาค่าได้ จึงทำให้การทำงานของโปรแกรมช้าเพราะต้องมีการคำนวณหาค่าตัวผกผันใหม่

3. เนื่องจากระบบรหัสลับแบบ อาร์เอสเอ นี้เมื่อเข้ารหัสจะทำให้ไฟล์รหัสลับมีขนาดใหญ่ กว่าไฟล์ข้อมูลเดิม

และถ้าต้องการให้ระบบนี้มีประสิทธิภาพมากยิ่งขึ้น ควรปรับปรุงและแก้ไขดังนี้

1. ข้อมูล และ กุญแจ ที่ใช้ในการเข้ารหัส ควรจะกระทำได้กับตัวอักขระทุกตัวในตารางรหัสแอสกี้

2. ปรับปรุงระบบ ให้สามารถใช้กับไฟล์ข้อมูลภาษาไทยได้

3. ปรับปรุงระบบ ให้เป็นระบบงานประเภทฝังตัว (resident)

4. ระบบนี้จะทำงานได้อย่างมีประสิทธิภาพและรวดเร็ว เมื่อใช้ ฮาร์ดแวร์ที่มีความเร็วสูง

5. ไฟล์รหัสลับควรมีขนาดเล็กกว่าหรือเท่ากับไฟล์ข้อมูลเดิม

6. ระบบควรจะมีส่วนที่เป็น เอดิเตอร์ (editor) เพื่อใช้ประโยชน์ในการดูผลในการเข้ารหัสและถอดรหัส หรือเพื่อการแก้ไขข้อมูลที่จะเข้ารหัสได้ทันที โดยไม่ต้องผ่านเอดิเตอร์ของโปรแกรมอื่น

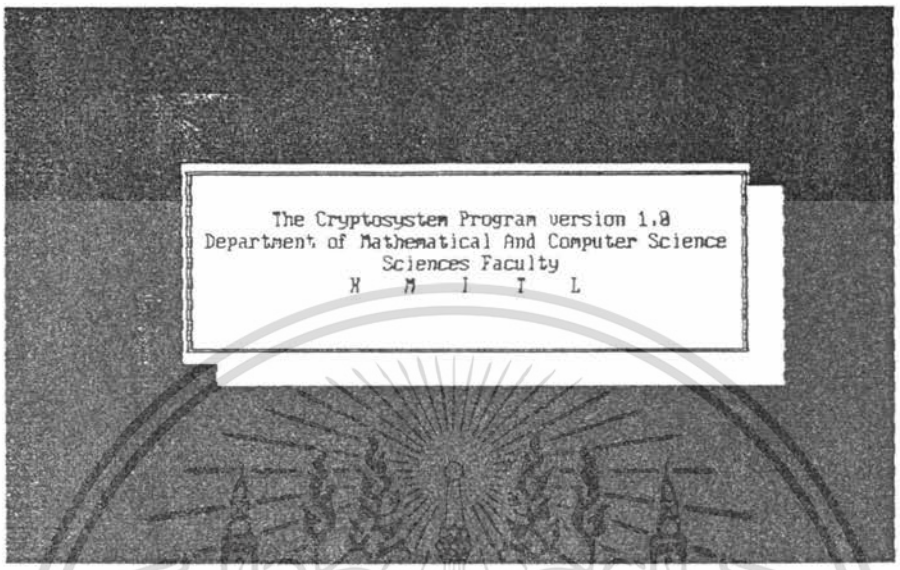
ภาคผนวก

คู่มือการใช้งานระบบ Cryptosystem version 1.0

การเข้าสู่ระบบ Cryptosystem โดยใช้คำสั่ง

A:> crypto

ก็จะปรากฏภาพที่หน้าจอดังรูปที่ 1



รูปที่ 1 ภาพบนจอเมื่อเข้าสู่ระบบ Cryptosystem

จากนั้นให้กดคีย์ใด ๆ ก็ได้ ก็จะปรากฏภาพบนจอภาพดังรูปที่ 2

รูปที่ 2 ภาพบนจอเมื่อกดคีย์แล้ว

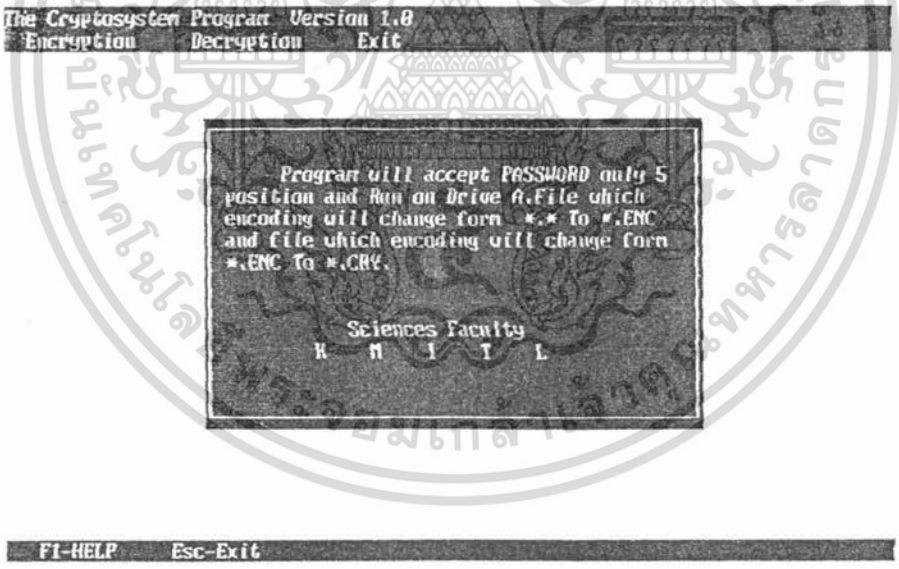
จากรูปที่ 2 แสดงว่าขณะนี้อยู่ที่เมนเมนู (main menu) ขอให้สังเกตุดูจะเห็นว่าได้แบ่งจอภาพออกเป็น 3 ช่องหน้าต่าง (window) คือ

ส่วนบน ช่องหน้าต่างนี้มีเพียงบรรทัดเดียว อยู่ส่วนบนของจอภาพ โดยจะมีข้อความว่า "The Cryptosystem program version 1.0"

ส่วนกลาง เมนเมนู ช่องหน้าต่างนี้มีเพียงบรรทัดเดียว อยู่ส่วนบนของจอภาพ โดยจะมีข้อความว่า Encryption Decryption และ Exit ขณะนี้เคอร์เซอร์ทับจะอยู่ตรงคำว่า Encryption ซึ่งช่องหน้าต่างนี้จะมีสีพื้นเป็นสีแดง และ เคอร์เซอร์เป็นสีเขียว การเลือกรายการใดก็ทำได้ด้วยการเลื่อนเคอร์เซอร์ด้วยการกดคีย์ลูกศร เช่น

<-- หรือ --> (มีอยู่ 4 คีย์ อยู่ทางขวาคีย์บอร์ด) ไปตรงรายการที่ต้องการแล้วกดคีย์ Enter

ส่วนล่าง ฟังก์ชั่นคีย์ ช่องหน้าต่างนี้มีเพียงบรรทัดเดียวเช่นกัน อยู่ที่ส่วนล่างสุดของจอภาพ ที่มีคำว่า F1-Help ซึ่งจะเป็นฟังก์ชันที่ช่วยแนะนำผู้ใช้ให้เข้าใจระบบมากขึ้น เมื่อกด ฟังก์ชั่นคีย์ F1 จะขึ้นหน้าจอดังรูปที่ 3



รูปที่ 3 แสดงหน้าจอเมื่อกดฟังก์ชั่นคีย์ F1

และ ESC-Exit เพื่อบอกว่าต้องการจะกลับสู่ระบบเดิมอย่างไร

เมื่อต้องการจะกลับไปสู่ระบบเดิมก่อนเข้าระบบ Cryptosystem ก็สามารถทำได้โดยกดคีย์ Esc แล้วจะมีข้อความขึ้นว่า "Are you sure (Y/N) ?" ถ้าผู้ใช้ต้อง

การออกจากระบบก็กดคีย์ตัวอักษรตัว Y ก็จะกลับสู่ระบบเดิม แต่ถ้ากดคีย์ตัวอักษรตัว N ก็จะปรากฏหน้าจอดังรูปที่ 2 ดังเดิม หรืออาจทำได้โดยกดคีย์ลูกศรของหน้าต่างที่สองไปทางขวามือสุด คือ Exit แล้วกดคีย์ Enter ก็จะกลับไปสู่ระบบเดิมเช่นกัน

รูปที่ 4 แสดงหน้าต่างของระบบ Cryptosystem มีอยู่ 3 หน้าต่าง

สรุปส่วนประกอบของหน้าต่าง

1.หน้าต่างส่วนบน ชื่อระบบ และ เวอร์ชัน (version) อยู่บรรทัดบนสุด (บรรทัดเดียว)

2.หน้าต่างส่วนกลาง Main menu อยู่บรรทัดที่สอง ส่วนบนของหน้าจอ

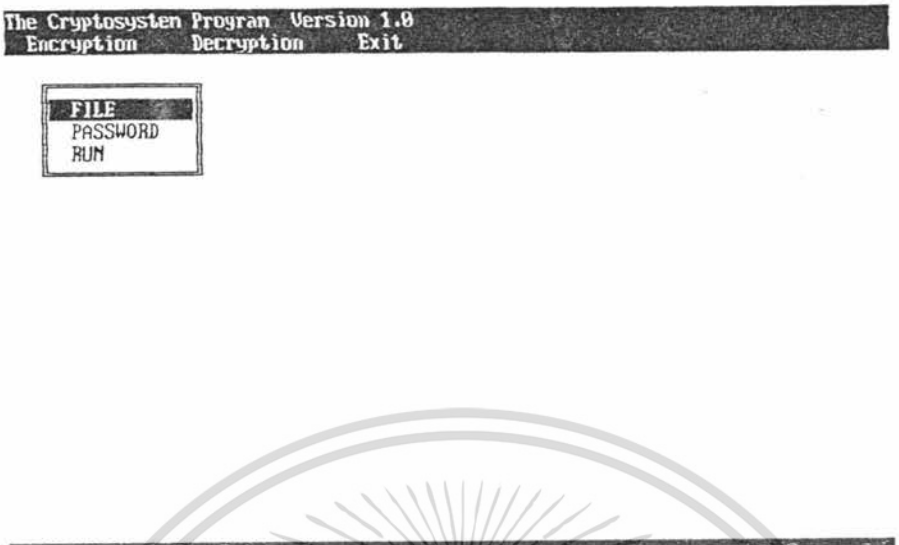
3.หน้าต่างส่วนล่าง Function key อยู่บรรทัดล่างสุด(บรรทัดเดียว)

ตอนนี้เราจะขอกลับไปอธิบาย เมนเมนู ดังรูปที่ 2 โดยที่บรรทัดที่ 2 มีคำว่า Encryption Decryption และ Exit โดยที่รายการ Encryption Decryption และ Exit มีรายการรองให้เลือกอีกและเมื่อเลือกรายการใด เช่น เลือก Encryption ก็จะถือว่าอยู่ที่เมนู Encryption ซึ่งจะแสดงรายการรองให้เลือกต่อไปอีกในลักษณะที่เรียกว่า pull down menu

อีกเรื่องหนึ่งที่ควรทราบก่อนคือ เมื่ออยู่ที่เมนูรองต้องการกลับไปที่เมนูหลักให้กดคีย์ Esc ในทำนองเดียวกัน เมื่ออยู่ที่เมนูรองของเมนูรองอีกที ให้กดคีย์นี้เพื่อกลับไปที่เมนูรอง เมื่อกดคีย์ Esc อีกครั้งก็จะกลับไปที่เมนูหลัก

1. Encryption

เมื่อเลือกรายการ Encryption ระบบจะแสดงเมนูรองดังรูปที่ 4.1



รูป 4.1 แสดงเมนูของการเข้ารหัส

เมื่อเลือกรายการ Encryption ระบบจะแสดงเมนูรองดังนี้

FILE เพื่อกำหนดว่าจะให้ทำการเข้ารหัส (encryption) ในไฟล์ใดโดยระบุชื่อไฟล์(file) และ ไดเร็กทอรี่(directory) ซึ่งถ้ากดคีย์ Enter จะแสดงหน้าจอดังรูป 4.1.1-1

รูปที่ 4.1.1-1 แสดงเมนูรองของไฟล์ในการเข้ารหัส

จากรูปที่ 4.1.1-1 จะเห็นว่า

1.เมื่อกดคีย์ Enter ระบบจะแสดงชื่อไฟล์ในไดเร็กทอรีปัจจุบัน สามารถจะเลือกโหลดไฟล์ใดก็ได้โดยเลื่อนเคอร์เซอร์ทิบไปยังชื่อไฟล์นั้น แล้วกดคีย์ Enter หรือจะเลือกไดเร็กทอรีรองเพื่อเลือกไฟล์ในไดเร็กทอรีรองต่อไปก็ได้เช่นกัน

2. หากป้อนไดเร็กทอรี รวมทั้งไดเร็กทอรีรองแล้วตามด้วยชื่อไฟล์ก็จะโหลดไฟล์ในไดเร็กทอรีนั้นลงหน่วยความจำ แต่ถ้าป้อนไดเร็กทอรีตามด้วย wild card(*) ก็จะแสดงชื่อไฟล์ในไดเร็กทอรีนั้น

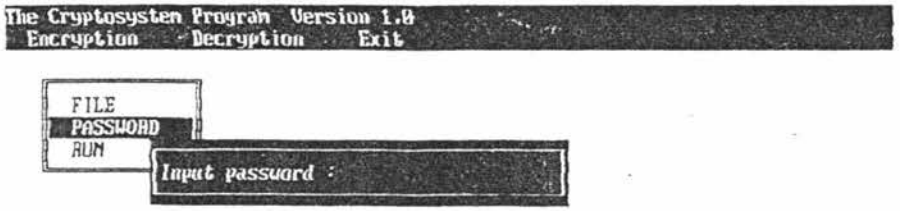
3. หากป้อนชื่อไฟล์(ไม่มีนามสกุล) แล้วกดคีย์ Enter ระบบจะตรวจว่ามีไฟล์ชื่อนี้โดยจะขึ้นชื่อไฟล์นี้ กับ นามสกุลที่มีอยู่ทั้งหมด (ถ้ามีมากกว่า 1 ชื่อ : 1 นามสกุล)

4. หากป้อนชื่อหรือทำการเลือกไฟล์ที่ต้องการแล้ว ไม่มีไฟล์ในแฟ้มข้อมูลที่ต้องการ ระบบจะแสดงหน้าจอ ดังรูปที่ 4.1.1-2

รูป 4.1.1-2 แสดงหน้าจอเมื่อไม่พบแฟ้มข้อมูลที่ต้องการ

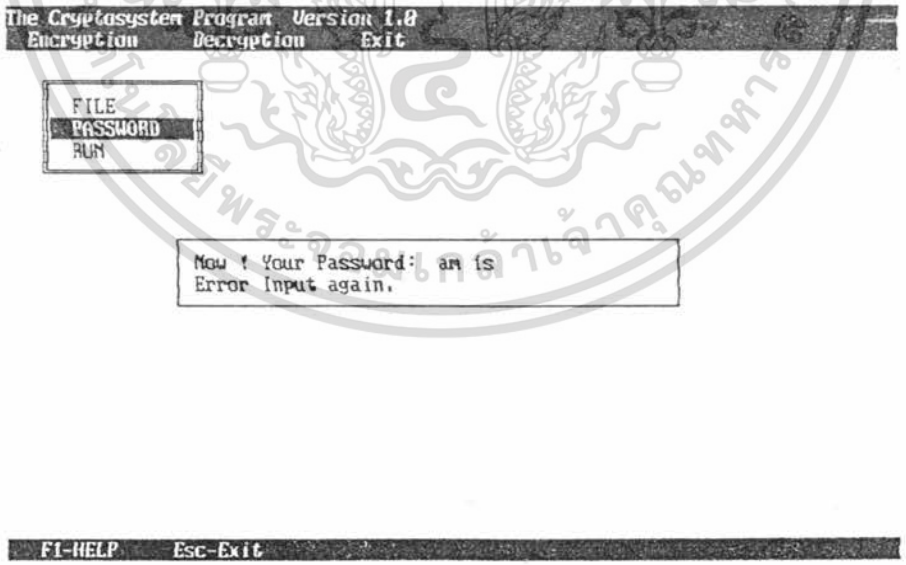
PASSWORD เป็นส่วนหนึ่งของขั้นตอนในการเข้ารหัสลับ ซึ่งจะช่วยเพิ่มความปลอดภัยให้กับข้อมูลมากขึ้นโดยผู้ทำการเข้ารหัส จะเป็นผู้ใส่รหัสผ่านจำนวนตั้งแต่ 5 ตัวอักษรขึ้นไป โดยเงื่อนไขของการใส่รหัสผ่านคือ ใส่รหัสผ่านใด ๆ ก็ได้ (ยกเว้นคีย์หน้าที่พิเศษ เช่น F1-F10 ,Insert,home) และถ้าใส่เกิน 5 ตัวจะถือว่ามีแค่ 5 ตัวแรกเท่านั้นที่เป็นรหัสผ่านที่ต้องการเพื่อนำไปใช้เป็นกุญแจสาธารณะ เมื่อเข้าเมนูรองโดยเลือก

PASSWORD ซึ่งถ้ากดคีย์ Enter จะแสดงหน้าจอดังรูป 4.1.2-1



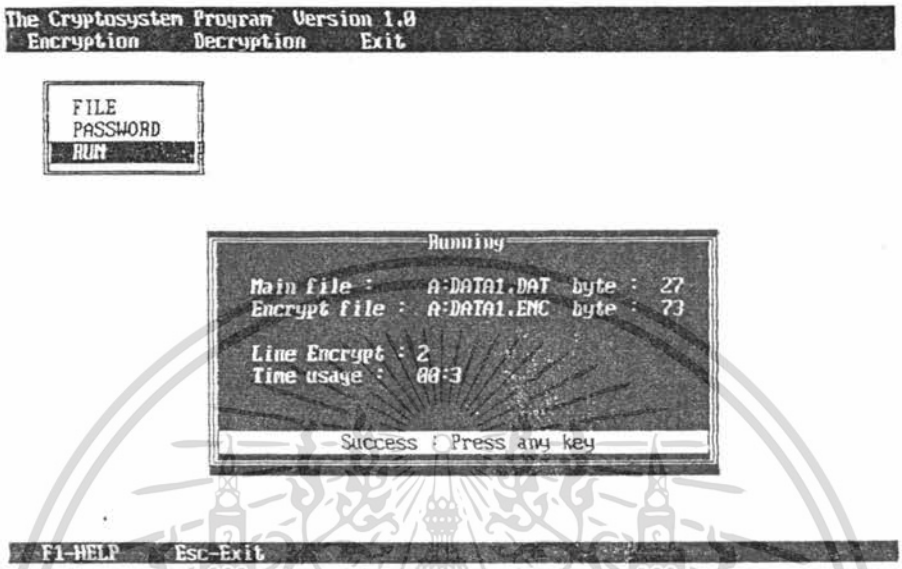
รูปที่ 4.1.2-2 แสดงเมนูรองของ "PASSWORD"

หากใส่รหัสผ่าน (password) ผิดเงื่อนไขการใส่รหัสผ่าน จะขึ้นหน้าจอดังรูปที่ 4.1.2-2



รูปที่ 4.1.2-2 แสดงหน้าจอเมื่อใส่รหัสผ่านผิดเงื่อนไข

RUN เป็นการเริ่มทำการเข้ารหัส โดยจะเป็นการแสดงว่าเป็นเข้ารหัสของไฟล์ใด ไฟล์หนึ่ง ที่เราป้อนชื่อไฟล์ หรือทำการเลือกไว้ในเมนู "FILE" โดยจะต้องผ่านขั้นตอนการใส่รหัสผ่านมาก่อนในขั้นตอนเมนู "PASSWORD" โดยทำการเลือกเมนู "RUN" แล้วกดคีย์ Enter ระบบจะทำการเข้ารหัสไฟล์นั้น ๆ โดยจะแสดงหน้าจอการประมวลผลดังรูป 4.1.3-1

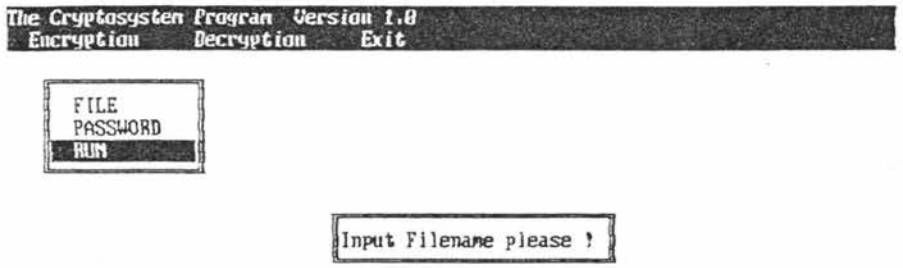


รูป 4.1.3-1 แสดงหน้าจอเมื่อกดคีย์ Enter ของเมนู "RUN"

ถ้าไฟล์ ที่ต้องการทำการเข้ารหัสนั้นได้ทำการเข้ารหัสเรียบร้อยแล้ว ผลการทำงานของระบบแสดงหน้าจอดังรูป 4.1.3-2

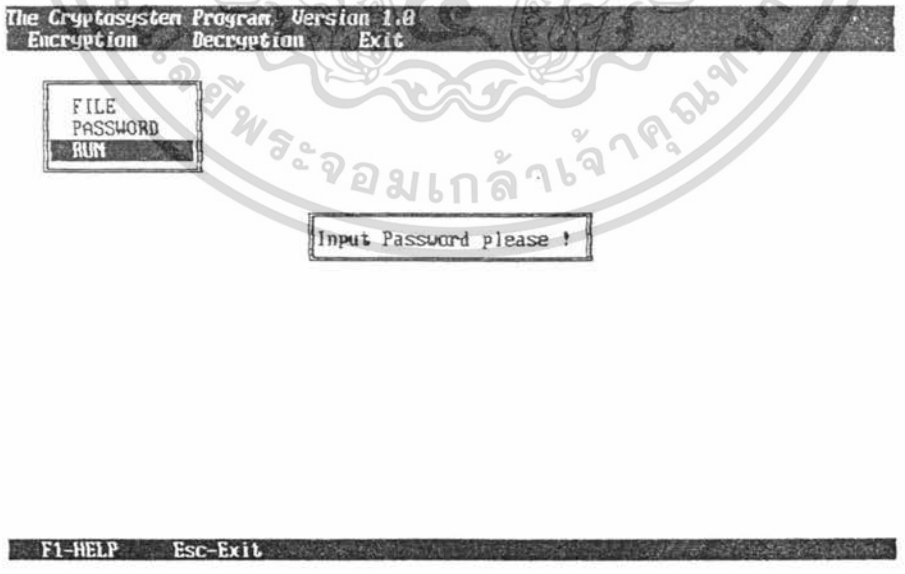
รูปที่ 4.1.3-2 แสดงหน้าจอเมื่อมีการเข้ารหัสของไฟล์ ซ้ำซ้อน

ถ้าเราข้ามขั้นตอนการป้อนชื่อไฟล์ หรือ ทำการเลือกไฟล์ที่ต้องการ โดยข้ามมาเลือกเมนู "RUN" เลย ระบบจะแสดงหน้าจอดังรูป 4.1.3-3



รูปที่ 4.1.3-3 แสดงหน้าจอของระบบเมื่อข้ามขั้นตอนการป้อนชื่อไฟล์ หรือทำการเลือกไฟล์ในเมนู "FILE"

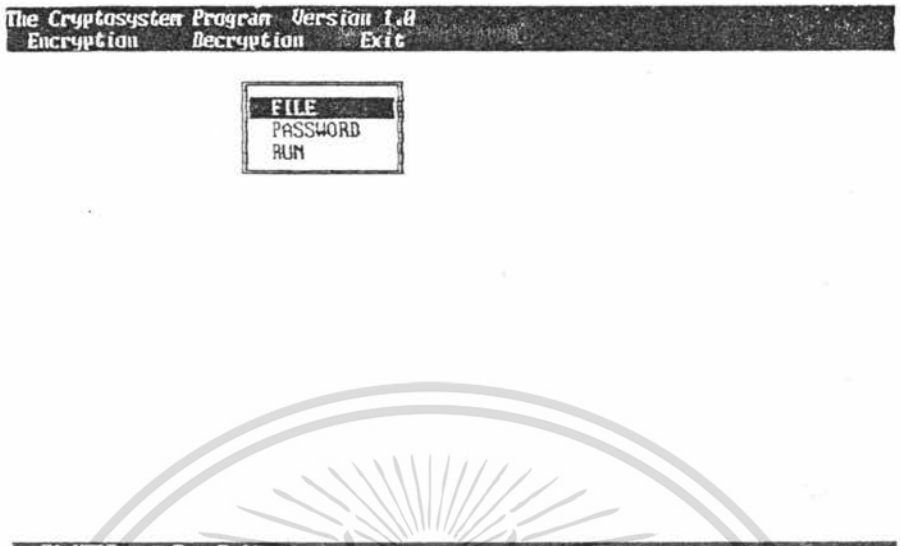
ถ้าเราข้ามขั้นตอนการใส่รหัสผ่านมาก่อนในขั้นตอนเมนู "PASSWORD" โดยข้ามมาเลือกเมนู "RUN" เลย ระบบจะแสดงหน้าจอดังรูป 4.1.3-4



รูปที่ 4.1.3-4 แสดงหน้าจอของระบบเมื่อข้ามขั้นตอนการใส่รหัสผ่าน

2. Decryption

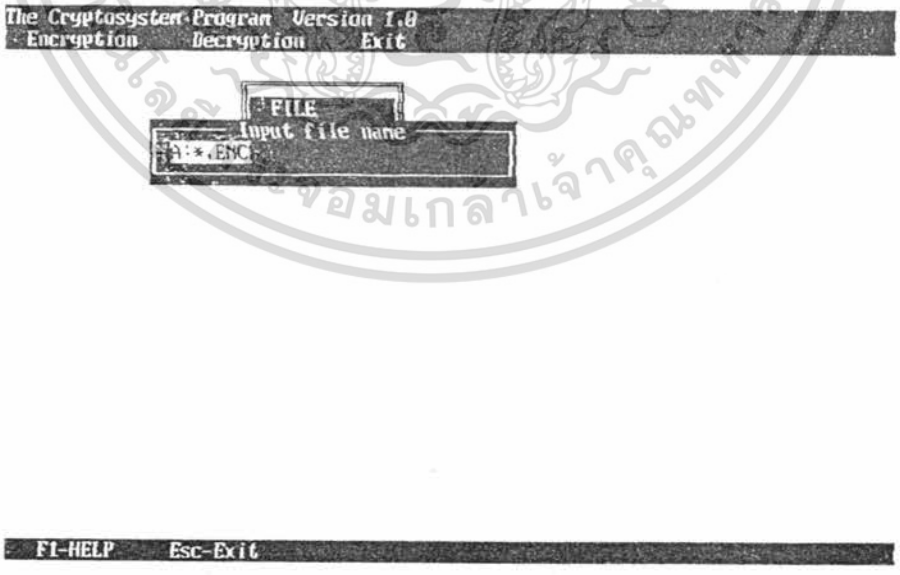
เมื่อเลือกรายการถอดรหัส(decryption) ระบบจะแสดงเมนูรองดังรูปที่ 4.2



รูป 4.2 แสดงเมนูของการถอดรหัส

เมื่อเลือกรายการถอดรหัส (decryption) ระบบจะแสดงเมนูรองดังนี้

FILE เพื่อกำหนดว่าจะให้ทำการถอดรหัส (decryption) ในไฟล์ใด โดยระบุชื่อไฟล์และไดเร็กทอรี ซึ่งถ้ากดคีย์ Enter จะแสดงหน้าจอดังรูป 4.2.1-1



รูปที่ 4.2.1-1 แสดงเมนูรองของไฟล์ในการถอดรหัส

จากรูปที่ 4.2.1-1 จะเห็นว่าจะเป็นเช่นเดียวกันกับการเข้ารหัส

1. เมื่อกดคีย์ Enter ระบบจะแสดงชื่อไฟล์ในไดเร็กทอรีปัจจุบัน สามารถจะเลือกโหลดไฟล์ใดก็ได้โดยเลื่อนเคอร์เซอร์ทิบไปยังชื่อไฟล์นั้น แล้วกดคีย์ Enter หรือจะเลือกไดเร็กทอรีรองเพื่อเลือกไฟล์ในไดเร็กทอรีรองต่อไปก็ได้เช่นกัน

2. หากป้อนไดเร็กทอรี รวมทั้งไดเร็กทอรีรองแล้วตามด้วยชื่อไฟล์ ก็จะโหลดไฟล์ในไดเร็กทอรีนั้นลงหน่วยความจำ แต่ถ้าป้อนไดเร็กทอรีตามด้วย wild card (*) ก็จะแสดงชื่อไฟล์ในไดเร็กทอรีนั้น

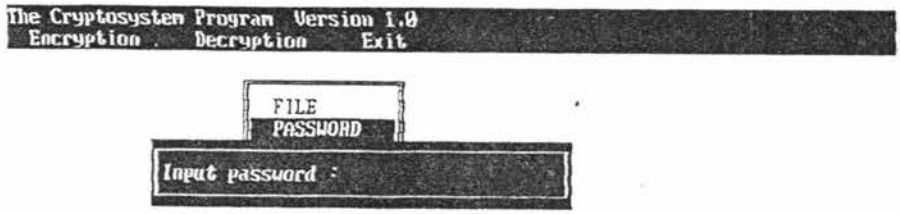
3. หากป้อนชื่อไฟล์(ไม่มีนามสกุล) แล้วกดคีย์ Enter ระบบจะตรวจว่ามีไฟล์ชื่อนี้โดยจะขึ้นชื่อไฟล์นี้ กับ นามสกุลที่มีอยู่ทั้งหมด (ถ้ามีมากกว่า 1 ชื่อ : 1 นามสกุล)

4. หากป้อนชื่อหรือทำการเลือกไฟล์ที่ต้องการแล้ว ไม่มีไฟล์ในแฟ้มข้อมูลที่ต้องการ ระบบจะแสดงหน้าจอ ดังรูปที่ 4.2.1-2

รูป 4.2.1-2 แสดงหน้าจอเมื่อไม่พบแฟ้มข้อมูลที่ต้องการ

PASSWORD เป็นส่วนหนึ่งของขั้นตอนในการถอดรหัสลับ เช่นเดียวกับการเข้ารหัสลับซึ่งต้องมีการใส่รหัสผ่าน โดยผู้ทำการถอดรหัส จะเป็นผู้ใส่รหัสผ่าน ซึ่งจะต้องเป็นรหัสเดียวกับรหัสผ่านในการเข้ารหัสโดยมีเงื่อนไขการใส่รหัสผ่านเช่นเดียวกัน

เมื่อเข้าเมนูรองโดยเลือก "PASSWORD" ซึ่งถ้ากดคีย์ Enter จะแสดงหน้าจอดังรูป 4.2.2-1



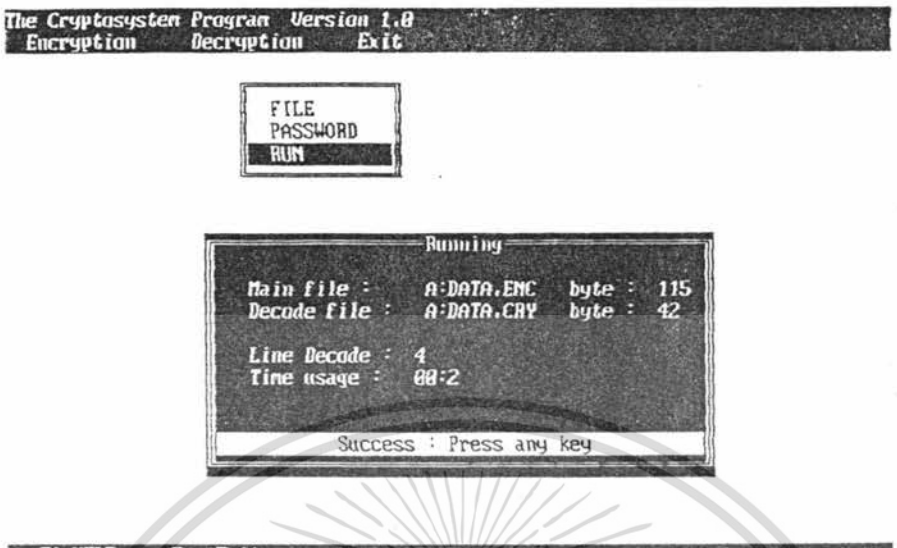
รูปที่ 4.2.2-1 แสดงเมนูรองของรหัสผ่านในการถอดรหัส

หากป้อนรหัสผ่านผิดเงื่อนไขการใส่รหัสผ่านจะขึ้นหน้าจอดังรูปที่ 4.2.2-2

รูปที่ 4.2.2-2 แสดงหน้าจอเมื่อใส่คีย์ผิดเงื่อนไข

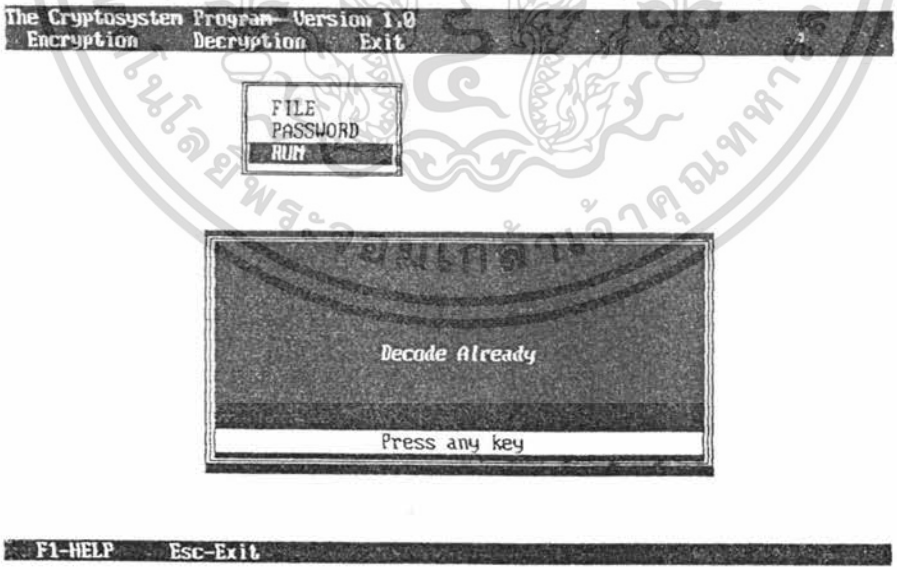
RUN เป็นการเริ่มทำการถอดรหัส โดยจะเป็นการแสดงว่าเป็นถอดรหัสของไฟล์ใด ไฟล์ หนึ่ง ที่เราป้อนชื่อไฟล์ หรือทำการเลือกไว้ในเมนู "FILE" โดยจะต้องผ่านขั้นตอนการใส่รหัสผ่านมาก่อนในขั้นตอนเมนู "PASSWORD" โดยทำการเลือกเมนู "RUN" แล้ว

กดคีย์ Enter ระบบจะทำการถอดรหัสไฟล์นั้น ๆ โดยจะแสดงหน้าจอการประมวลผลดังรูป 4.2.3-1



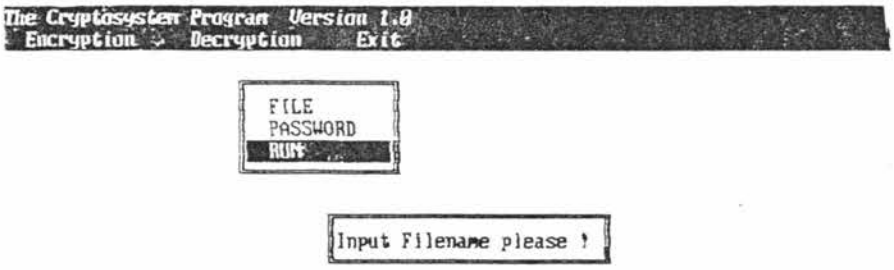
รูป 4.2.3-1 แสดงหน้าจอเมื่อกดคีย์ Enter ของเมนู "RUN"

ถ้าไฟล์ ที่ต้องการทำการถอดรหัสนั้น ได้ทำการถอดรหัสเรียบร้อยแล้ว ผลการทำงานของระบบแสดงหน้าจอดังรูป 4.2.3-2



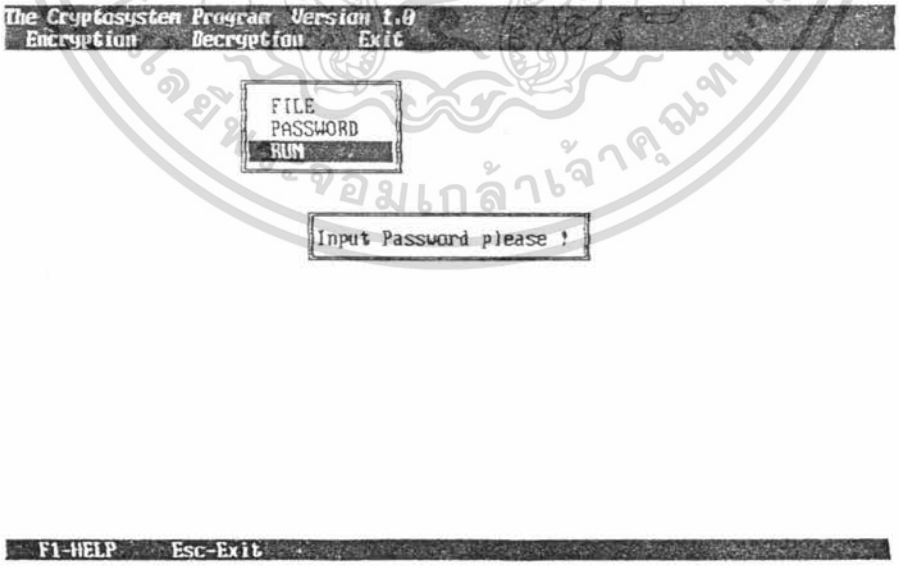
รูปที่ 4.2.3-2 แสดงหน้าจอเมื่อมีการถอดรหัสของไฟล์ ซ้ำซ้อน

ถ้าเราข้ามขั้นตอนการป้อนชื่อไฟล์ หรือ ทำการเลือกไฟล์ที่ต้องการ โดยข้ามมาเลือกเมนู "RUN" เลย ระบบจะแสดงหน้าจอดังรูป 4.2.3-3



รูปที่ 4.2.3-3 แสดงหน้าจอของระบบเมื่อข้ามขั้นตอนการป้อนชื่อไฟล์ หรือทำการเลือกไฟล์ในเมนู "FILE"

ถ้าเราข้ามขั้นตอนการใส่รหัสผ่าน มาก่อนในขั้นตอนเมนู "PASSWORD" โดยข้ามมาเลือกเมนู "RUN" เลย ระบบจะแสดงหน้าจอดังรูป 4.2.3-4



รูปที่ 4.2.3-4 แสดงหน้าจอของระบบเมื่อข้ามขั้นตอนการใส่รหัสผ่าน

3. Exit

เมื่อต้องการที่จะออกจากระบบ Cryptosystem แล้วกลับสู่ระบบเดิม โดยเลือกเมนู "Exit" แล้วกดคีย์ Enter ระบบจะกลับสู่ระบบเดิม ดังในรูปที่ 4.3.1

```
The Cryptosystem Program version 1.0 ,copyright(c)  1992.
A:\>>
```

รูปที่ 4.3.1 แสดงหน้าจอ เมื่อเลือก "Exit" แล้วกดคีย์ Enter

เอกสารอ้างอิง

Charles P.Pfleeger in Security In Computing, Prentice-Hall International Editions, pp. 1-128, A Division Of Simon & Sehuster Englewood Cliffs, N.J. 1989.

Jennifer Seberry and Josef Pieprzyk in Cryptography An Introduction To Computer Security, Prentice - Hall Advance In Computer Science Series, pp. 1-130, Richard P.Brent-Editor, Australia, 1989.