บทที่ 5

แนวทางในการออกแบบและรายละเอียดการทำงาน

แนวทางการออกแบบ

การออกแบบเอนคริปเตอร์และออเทนทิเคเตอร์นี้ได้ออกแบบให้ฮาร์ดแวร์และซอฟท์แวร์สามารถเข้ารหัสได้ทุก ๆ โหมด คือ

อิเล็กทรอนิกส์โค้ดบุ๊ค (Electronic Codebook) หรือ ECB

ไซเฟอร์บล็อคเชนนิ่ง (Cipher Block Chaining) หรือ CBC

ไซเฟอร์ฟีดแบ็ค (Cipher Feedback) หรือ CFB

เอาท์พุทฟีดแบ็ค (Output Feedback) หรือ OFB

โดยในแต่ละโหมดมีคุณสมบัติดังนี้ [Caelli et al.,1989]

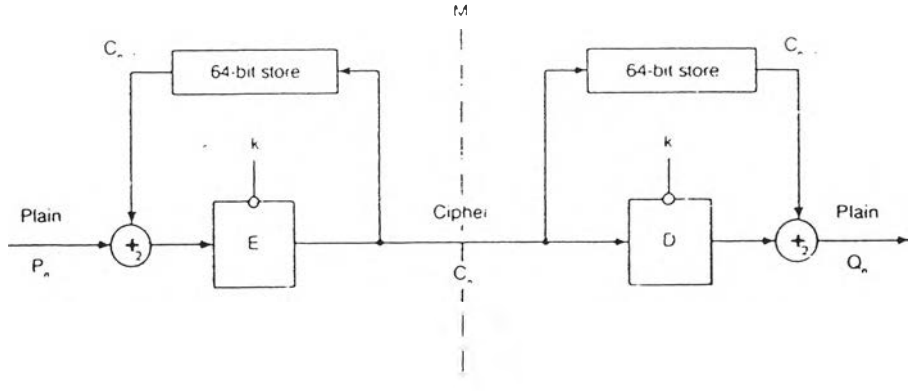
1. อิเล็กทรอนิกส์โค้ดบุ๊ค

ECB คือ โหมดในการใช้ DES ในการเข้ารหัสแบบตรงไปตรงมาที่สุดคือเมื่อมีข้อมูลเข้ามา 64 บิตผ่านDES ด้วย key 56 บิต ก็จะได้ข้อมูลขาออก ซึ่งเราเรียกว่า ไซเฟอร์เท็กซ์ (Ciphertext) จำนวน 64 บิต ข้อเสียเปรียบซึ่งถือเป็นจุดอ่อนของการทำงานในโหมดนี้คือ ถ้าหากเพลนเท็กซ์เป็นข้อมูลที่มีรูปแบบ (Format) ซ้ำกันมาก ๆ ไซเฟอร์เท็กซ์ที่ออกมาก็จะเป็นข้อมูลที่มีรูปแบบเช่นเดียวกัน ทำให้เป็นการง่ายที่จะจับคู่หาความสัมพันธ์ของเพลนเท็กซ์ และไซเฟอร์เท็กซ์ และแปลความหมายของข้อมูลนั้น ซึ่งก็จะเป็นการไม่ปลอดภัย

2. ไซเฟอร์บล็อคเชนนิ่ง

ไซเฟอร์บล็อคเชนนิ่ง คือ โหมดการทำงานของ DES ที่จะแก้จุดอ่อนของการเข้ารหัสแบบอิเล็กทรอนิกส์ โค้ดบุ๊ค (Electronic Codebook) ไซเฟอร์เท็กซ์เอาท์พุท (Ciphertext Output) จะขึ้นอยู่กับคีย์ (key)และกลุ่มของข้อมูลอินพุท(Plaintext Block) เพราะฉะนั้นข้อความหรือข้อมูลที่มีความมี

รูปแบบ(format)มากก็ไม่ทำให้เกิดการมีรูปแบบของไซเฟอร์เท็กซ์เหมือนในโหมดการทำงานแบบอิเล็กทรอนิกส์โค้ดบุ๊ค การทำงานของโหมดไซเฟอร์บล็อคเชนนิ่งนี้ก็จะแสดงได้ดังรูป 5.1



รูป 5.1 การเข้ารหัสในโหมดไซเฟอร์บล็อคเชนนิ่ง [Caelli et al.,1989]

ชิพรีจิสเตอร์ขนาด64บิตจะถูกโหลดไว้ด้วยกลุ่มของข้อมูลที่เรียกว่าเวกเตอร์เริ่มต้น (Initialization Vector) IV ข้อมูลอินพุท64บิตแรกก็จะถูกทำการบวกแบบโมดูโล-2กับข้อมูลในชิพรีจิสเตอร์และเอาท์พุทที่ได้ออกมาก็จะถูกเข้ารหัสผ่านกระบวนการ DES ด้วยคีย์ ไซเฟอร์เท็กซ์เอาท์พุทที่ได้ก็จะมีขนาด 64 บิต จะถูกส่งเข้าชิพรีจิสเตอร์ เพื่อจะได้ใช้เป็นตัวที่จะบวกแบบโมดูโล-2กับข้อมูลอินพุทชุดต่อไปและในขณะเดียวกันก็จะถูกส่งออกไปยังเครื่องรับปลายทางด้วยทางด้านเครื่องรับก็จะมีชิพรีจิสเตอร์ซึ่งถูกโหลดไว้ด้วยเวกเตอร์เริ่มต้น IV เช่นเดียวกับทางด้านเครื่องส่ง กลุ่มของข้อมูลอินพุทชุดแรกที่รับได้ก็จะถูกถอดรหัส (decrypt) และเอาท์พุทที่ได้ก็จะถูกบวกแบบโมดูโล-2กับ IV ในชิพรีจิสเตอร์ ทำให้ได้ข้อมูลที่เป็นข้อมูลอินพุทชุดแรกที่ถูกส่งมาจากทางด้านเครื่องส่งและในขณะเดียวกันข้อมูลอินพุทชุดแรกที่เครื่องรับรับได้ ก็จะเข้าไปแทนที่ข้อมูล IV ที่มีอยู่ในชิพรีจิสเตอร์เพื่อใช้เป็น IV สำหรับบวกแบบโมดูโล-2กับข้อมูลอินพุทชุดต่อไป

การทำงานในโหมด CBC นี้ สามารถแสดงได้ด้วยสมการทางคณิตศาสตร์ดังนี้

$$C_n = E_k(P_n + C_{n-1})$$

โดยที่ $E_k(P)$ = ไซเฟอร์เท็กซ์ที่เกิดขึ้นจากการเข้ารหัส P ด้วย คีย์ k เอาท์พุทของการถอดรหัส คือ

$$Q_n = D_k(C_n) + C_{n-1}$$

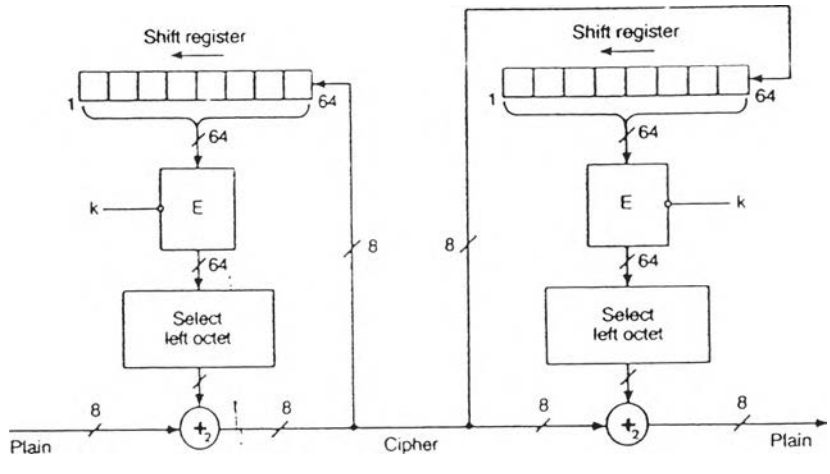
โดยที่ $D_k(C_n)$ กลุ่มข้อมูลที่เกิดจากการถอดรหัส ไซเฟอร์เท็กซ์ด้วย คีย์ k ดังนั้น

$$\begin{aligned} Q_n &= D_k(E_k(P_n + C_{n-1})) + C_{n-1} \\ &= P_n + C_{n-1} + C_{n-1} \\ &= P_n \end{aligned}$$

ทั้งทางด้านเครื่องรับและเครื่องส่งจะต้องมีค่า IV เดียวกัน โดยที่ส่งมาจากทางด้านเครื่องส่ง ซึ่งโดยปกติแล้วจะส่งมาในรูปของไซเฟอร์เท็กซ์ที่เข้ารหัสด้วยรหัสลับแต่ก็ไม่จำเป็นต้องส่งมาในรูปของไซเฟอร์เท็กซ์เสมอไปจะส่งมาเป็นเพลนเท็กซ์ก็ได้แต่ข้อสำคัญคือจะต้องป้องกันไม่ให้ IV นั้นถูกแก้ไขหรือเปลี่ยนแปลง ถ้าหากมีการเปลี่ยนแปลง IV ให้ทางด้านเครื่องรับมีค่าไม่ตรงกับเครื่องส่ง จะทำให้เอาท์พุทที่ได้ออกมาทางเครื่องรับไม่ตรงกับต้นทางที่ส่งมาผลกระทบนี้จะมีเฉพาะกลุ่มแรกของข้อมูลเท่านั้น (64 บิต) ข้อมูลกลุ่มต่อไปจะไม่ถูกกระทบกระเทือน

CBC ต้องการข้อมูลที่มีขนาดเป็นจำนวนเท่าของ 64 บิต ในกรณีที่ขนาดของข้อมูลไม่ได้เป็นจำนวนเท่าของ 64 บิต กลุ่มสุดท้ายของข้อมูลจะต้องถูกเติมด้วย 0 จนกว่าจะครบ 64 บิต

3. ไซเฟอร์ฟีดแบ็ค



รูป 5.2 การเข้ารหัสในโหมดไซเฟอร์ฟีดแบ็ค [Caelli et al.,1989]

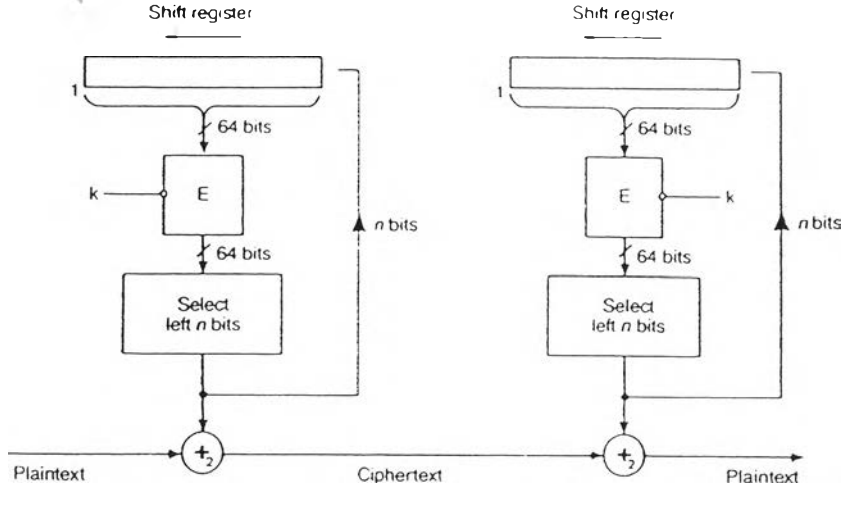
CFB เป็นโหมดการทำงานที่เหมาะสำหรับข้อมูลที่ไม่เป็นจำนวนเต็มของ 64 บิต เช่น ในการส่งตัวอักษรจากเทอร์มินอล แต่ละตัวอักษรจะใช้ 8 บิต ดังนั้นข้อมูลที่ส่งจึงอาจจะไม่เป็นจำนวนเต็มของ 64 ตามต้องการ ในโหมดการทำงานแบบ CFB บิตที่เกิดขึ้นแบบกึ่งสุ่ม (Pseudorandom) ของ

IV ที่ผ่านชุดเอนคริปชั่น จำนวน 8 บิต จะถูกบวกแบบโมดูโล-2กับเพลนเท็กซ์ เพื่อสร้างให้เกิดไซเฟอร์เท็กซ์ ส่งออกไปยังเครื่องรับ

ในขณะเดียวกันไซเฟอร์เท็กซ์ชุดนี้ก็จะถูกป้อนกลับไปยังชิพรีจิสเตอร์ โดยการชิพไปทางซ้าย 8 บิต ขณะนี้เราจะได้ IV ชุดใหม่เพื่อที่จะผ่านชุดเข้ารหัสไปบวกแบบโมดูโล-2กับเพลนเท็กซ์ชุดต่อไป ทางด้านเครื่องรับ เมื่อรับไซเฟอร์เท็กซ์ชุดที่หนึ่งมา ไซเฟอร์เท็กซ์ชุดนี้จะถูกนำไปบวกแบบโมดูโล-2กับเอาท์พุทของชุดเข้ารหัส IV ทางด้านเครื่องรับ พื่อให้เกิดเป็นเพลนเท็กซ์ชุดที่หนึ่งที่เครื่องส่งส่งมาและไซเฟอร์เท็กซ์ชุดแรกนี้ก็ถูกป้อนไปยังชิพรีจิสเตอร์ ทำให้ชิพรีจิสเตอร์ชิพไปทางซ้าย 8 บิต เพื่อให้ข้อมูลในตัวมันรอที่จะผ่านการเข้ารหัสไปบวกแบบโมดูโล-2กับไซเฟอร์เท็กซ์ชุดต่อไปดังรูป 5.2

4. เอาท์พุทฟีดแบ็ค

เอาท์พุทฟีดแบ็ค เป็นโหมดการทำงานของ DES แบบ สตรีมไซเฟอร์ (Stream Cipher) ใน OFB นี้ คริฟโตกราฟฟิคบิตสตรีม (Cryptographic Bit Stream) จะถูกบวกแบบโมดูโล-2กับเพลนเท็กซ์เพื่อสร้างให้เกิดไซเฟอร์เท็กซ์ส่งออกไปจากเครื่องส่งทางด้านเครื่องรับซึ่งมีลักษณะเหมือนกันทุกประการก็จะกำเนิดคริฟโตกราฟิคบิตสตรีมเพื่อบวกแบบโมดูโล-2กับไซเฟอร์เท็กซ์ที่รับได้ให้เกิดเพลนเท็กซ์เหมือนกันกับเพลนเท็กซ์ทางด้านเครื่องส่งลักษณะการทำงานดังกล่าวแสดงได้ดังรูปดังรูป 5.3



รูป 5.3 การเข้ารหัสในโหมดเอาท์พุทฟีดแบ็ค [Caelli et al.,1989]

ข้อดีที่สำคัญของ OFB เมื่อเปรียบเทียบกับ CFB คือ การที่ OFB ไม่มีการขยายความผิด (error extension) ไซเฟอร์เท็กซ์ที่ผิดไป 1 บิต ก็จะทำให้เกิดเพลนเท็กซ์ที่ผิดพลาดไปเพียง 1 บิตเช่นกัน ในขณะที่ใน CFB ถ้าไซเฟอร์เท็กซ์ผิดไป 1 บิต จะทำให้เพลนเท็กซ์ผิดไป 8 บิต

ในส่วนของออเทนทิเคเตอร์ นอกจากจะมีการออกแบบให้มีการทำการสร้างรหัสรับของข้อความให้กับข้อมูลที่ต้องการจะส่งผ่านช่องสื่อสารแล้วก่อนหน้าที่จะทำการนี้เราจะเห็นได้ว่า ในกรณีที่ข้อมูลของเราถูกเก็บอยู่ในรูปเพลนไฟล์ในหน่วยความจำสำรอง เช่น ฟลอปปีดิสด์ หรือ ฮาร์ดดิสด์ จะมีความเสี่ยงสูงที่จะถูกทำการแก้ไข โดยที่เราไม่สามารถจะล่วงรู้ได้ว่าข้อมูลมีการเปลี่ยนแปลงไป เนื่องจากสาเหตุนี้เราจึงได้สร้างอัลกอริธึมในการทำการลงทะเบียนข้อความหรือไฟล์ (File Register) เพื่อใช้สำหรับยืนยันว่าข้อมูลนั้นถูกสร้างขึ้นมาโดยผู้สร้างเองและพร้อมที่จะทำการส่งผ่านไปยังผู้รับได้ ถ้าหากยังไม่มีการส่งไปยังผู้รับก็สามารถที่จะเก็บไว้ในหน่วยความจำสำรองที่ไม่มีการควบคุมการเข้าถึง (Access Control) ได้

ในการลงทะเบียนข้อความและการรับรองข้อความ เราใช้การเข้ารหัสในโหมด CBC เนื่องจากเหตุผลดังนี้

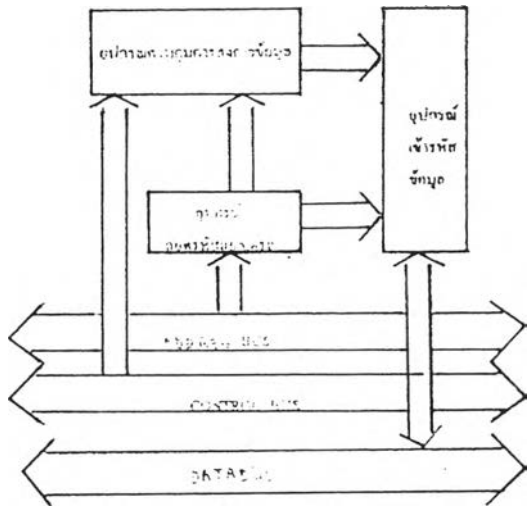
1. มีการแพร่กระจายความผิดพลาด
2. ไซเฟอร์ไฟล์ไม่คงความเป็นรูปแบบของเพลนไฟล์
3. มีความรวดเร็วในการทำงานเมื่อเทียบกับ CFB และ OFB

ทางด้านตัวเครื่องเอนคริปเตอร์และออร์เทนทิเคเตอร์เราได้ออกแบบให้มีลักษณะเป็นคาร์ดที่สามารถเชื่อมต่อเข้ากับช่องขยาย (Expansion Slot) ของเครื่องไมโครคอมพิวเตอร์ได้ โดยไม่ต้องมีสายเชื่อมโยง เพื่อความสะดวกในการใช้งาน และความแน่นอนในการทำงาน (Reliability) ที่ดี

สำหรับการควบคุมและใช้งานเครื่อง เราจะส่งผ่านทางโปรแกรมที่ได้พัฒนาขึ้นมา โดยมีลักษณะเป็น Menu Driven เพื่อความสะดวกของผู้ใช้งาน

<u>รายละเอียดทางด้านฮาร์ดแวร์</u>

บล็อคไดอะแกรมของการทำงานของเครื่องเอนคริปเตอร์และออร์เทนทิเคเตอร์แสดงดังในรูป 5.4



รูป 5.4 บล็อคไดอะแกรมเครื่องออร์เทนทิเคเตอร์และเอนคริปเตอร์

1. อุปกรณ์ถอดรหัสแอดเดรส [ธานินทร์ ถาวรศาสนวงศ์,ทินกร ดุ๊ก]

ในการควบคุมและตรวจสอบสภาวะการทำงานรวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพซัพพอร์ท หรือคาร์ดต่าง ๆ ที่ใช้ในระบบของ IBM/PC นั้น จะกระทำโดยผ่านทางพอร์ท I/O ของระบบ

สำหรับแอดเดรสของพอร์ท I/O ต่าง ๆ นั้น จะเป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะ คือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาดและมีแอดเดรสสำหรับใช้กับพอร์ทI/O อยู่ทั้งสิ้น 65,536 หรือ 64K แอดเดรส (ในขณะที่มีแอดเ ดรสสำหรับหน่วยความจำอยู่ 1Mbyte) ซึ่งทำให้การอ้างแอดเดรสของพอร์ท I/O ต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้น คือ A0-A15 แต่สำหรับใน IBM/PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ10 เส้นล่าง คือ A0-A9 เท่านั้น ดังนั้นในการอ้างถึงแอดเดรสของพอร์ทของอุปกรณ์ หรือชิพซัพพอร์ทใด ๆ ที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วย โดยเส้นแอดเดรสที่เหลือคือ A10-A15 นั้นจะไม่ถูกนำไปใช้งาน อย่างไรก็ตามถึงแม้ว่าเส้นแอดเดรส A10-A15 นี้จะไม่ถูกนำไปใช้งานแต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วย เพียงแต่ไม่ได้ถูกนำมาดีโค้ดร่วมกับแอดเดรส A0-A9 เท่านั้นตัวอย่างเช่นในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0010H นั้นจะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ทที่ตรงกับ

แอดเดรส 0410H, 0810, 0C10H ทั้งนี้เนื่องจากแอดเดรส 6 บิทบนไม่ได้ถูกใช้งานจึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10-A15 นั้น ไม่ทำให้เกิดความแตกต่างใด ๆ

เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0-A9) ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท (จากจำนวน 64K พอร์ท) เท่านั้น นอกจากนี้ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ทของ IBM/PC ข้อมูลในบิท A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1024 พอร์ทออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ท) อีกด้วย กล่าวคือถ้าข้อมูลในบิท A9 เป็น "0" แล้ว เราจะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์หรือชิพซัพพอร์ทต่าง ๆ ที่อยู่บนเมนบอร์ด (Main Board) ของ IBM/PC เช่น 8253-5, 8237-5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิท A9 นี้เป็น "1" ก็จะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนคาร์ดต่าง ๆ เท่านั้น

จากที่ได้กล่าวมานั้นจะสรุปได้ว่าพอร์ทบน IBM/PC ทั้ง 1024 พอร์ทถูกแบ่งออกเป็น 2 กลุ่ม โดยที่กลุ่มแรกเป็นกลุ่มของพอร์ทที่อยู่บนเมนบอร์ด และกลุ่มที่สองเป็นกลุ่มที่จัดเตรียมไว้สำหรับพอร์ทที่อยู่บนคาร์ดต่าง ๆ

สำหรับในกรณีของการส่งข้อมูลให้กับพอร์ททั้ง 1024 พอร์ท เราสามารถที่จะเลือกส่งไปยังพอร์ทใด ๆ ใน IBM/PC ได้ ดังนั้นการเลือกแอดเดรสสำหรับพอร์ทที่อยู่บนคาร์ดจึงสามารถทำได้โดยสะดวก แต่อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงก็คือ ถ้าแอดเดรสที่เราเลือกให้กับพอร์ทนี้ตรงกับค่าแอดเดรสเดิมที่มีอยู่บนเมนบอร์ดแล้ว เมื่อเราทำการส่งข้อมูลให้กับพอร์ทที่อยู่ในตำแหน่งแอดเดรสนี้ก็จะเท่ากับเป็นการส่งข้อมูลให้กับทั้งพอร์ทที่อยู่บนเมนบอร์ดและพอร์ทที่อยู่บคาร์ดด้วย ซึ่งในกรณีเช่นนี้อาจจะก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรสให้กับพอร์ทที่ถูกสร้างขึ้นบนคาร์ดต่าง ๆ จึงควรจะใช้ค่าแอดเดรสที่แอดเดรสบิท A9 มีค่าเป็น "1" คือ แอดเดรส 0 FE00H จนถึง 0FFFFH เท่านั้น (แอดเดรสบิท A10-A15 ไม่ถูกใช้ในการดีโค้ด แต่เพื่อความสะดวกจึงกำหนดให้มีค่าเป็น "1" ในฐานสองทั้งหมด แต่ในการใช้งานจริงอาจเปลี่ยนให้แอดเดรส A10-A15 แต่ละบิทมีค่าเป็น "1" หรือ "0" ก็ได้)

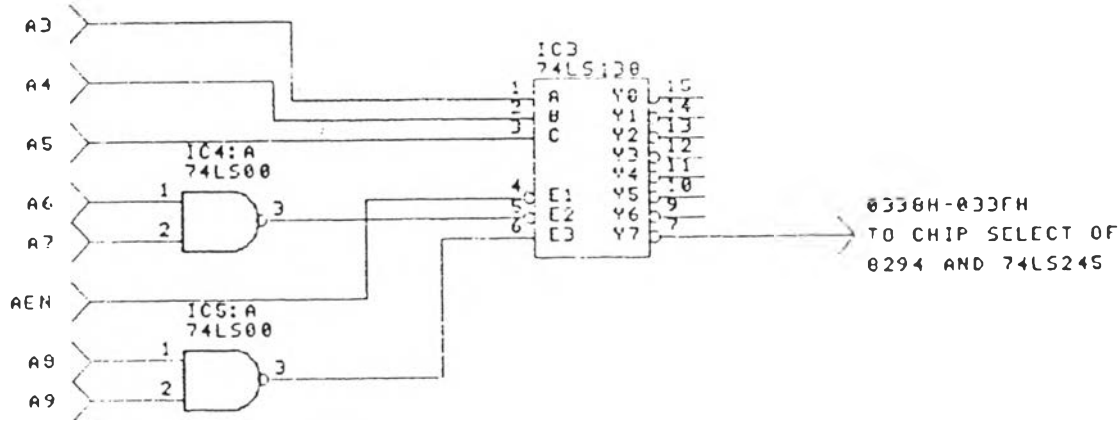
จากที่ผ่านมานั้น พอร์ท I/O ทั้ง 1024 พอร์ทใน IBM/PC จะถูกแบ่งออกเป็น 2 กลุ่ม ๆ ละ 512 พอร์ท สำหรับในหัวข้อนี้จะกล่าวถึงการใช้งานพอร์ทต่าง ๆ เหล่านี้ โดยจะแบ่งออกเป็น 2 กลุ่มตามที่ได้อธิบายไว้ในหัวข้อที่ผ่านมาดังนี้

1. ในกลุ่มแรกนี้เป็นกลุ่มของพอร์ท I/O ที่อยู่บนเมนบอร์ดของ IBM/PC ซึ่งจะมีแอดเดรสอยู่ในตำแหน่ง 0000H จนถึง 01FFH หรือแอดเดรสที่มีบิท A9 เป็น "0" นั่นเอง

2. ในกลุ่มที่สองนี้ จะเป็นกลุ่มของพอร์ท I/O ที่ถูกใช้งานอยู่บนคาร์ดที่ใช้เสียบบนสล็อตต่าง ๆ ของ IBM/PC สำหรับแอดเดรสของพอร์ทเหล่านี้จะเริ่มต้นจากแอดเดรส 0200H จนถึง 03FFH ซึ่งก็คือแอดเดรสที่มีบิท A9 เป็น "1" นั่นเอง สำหรับการใช้งานแอดเดรสของพอร์ท I/O จะแสดงได้ดังตาราง 5.1

อย่างไรก็ตามการใช้งานแอดเดรสในกลุ่มนี้อาจจะเปลี่ยนแปลงไปได้ ทั้งนี้ขึ้นอยู่กับการใช้งานคาร์ดต่าง ๆ ร่วมกับ IBM/PC โดยคาร์ที่ถูกออกแบบผลิตขึ้นใหม่นั้น อาจจะใช้ค่าแอดเดรสต่าง ๆ ที่เหลืออยู่นี้ได้ ดังนั้นก่อนที่จะทำการออกแบบวงจรอินเทอร์เฟสที่จำเป็นต้องใช้ค่าแอดเดรสสำหรับพอร์ท I/O จึงควรจะตรวจสอบดูก่อนว่าคาร์ดต่าง ๆ ที่ใช้อยู่ในระบบของ IBM/PC ที่เราใช้งานอยู่นั้นมีคาร์ดใดบ้าง และคาร์ดเหล่านี้ใช้งานแอดเดรสใดบ้าง จากนั้นจึงทำการออกแบบวงจรอินเทอร์เฟสโดยเลือกใช้เฉพาะแอดเดรสที่ยังไม่ถูกใช้งาน

สำหรับแอดเดรสที่เราเลือกใช้ในงานวิจัยนี้คือ แอดเดรส 0338H-033FH วงจรการเลือกใช้แอดเดรสดังกล่าวแสดงอยู่ในรูป 5.5



รูป 5.5 วงจรถอดรหัสแอดเดรส

จากรูปจะเห็นว่าวงจรที่ใช้นี้เป็นวงจรที่สามารถทำการถอดรหัสแอดเดรสได้ 8 กลุ่ม โดยแต่ละกลุ่มจะมีจำนวนแอดเดรส 8 แอดเดรส ซึ่งแอดเดรสทั้ง 8 กลุ่ม จะแสดงได้ดังตาราง 5.2

ตาราง 5.1 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนคาร์ดต่าง ๆ

หมายเลข พอร์ทรหัสฐานสิบหก	ชื่ออุปกรณ์
000-01F	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 1,8237A-5
020-03F	อินเตอร์รัพต์คอนโทรลเลอร์หมายเลข1,8259A
040-05F	ไทเมอร์ 8254-2
060-06F	8042 คีย์บอร์ด
070-07F	นาฬิกา และ NMI และซีมอสแรม
080-09F	DMA เพจรีจิสเตอร์
0A0-0BF	อินเตอร์รัพต์คอนโทรลเลอร์หมายเลข 2,8259A
0C0-0DF	ดีเอ็มเอคอนโทรลเลอร์หมายเลข 2,8237A-5
0F0	เคลียร์โปรเซสเซอร์คณิตศาสตร์
0F1	รีเซตโปรเซสเซอร์คณิตศาสตร์
0F8-0FF	โปรเซสเซอร์คณิตศาสตร์
1F0-1F8	ฮาร์ดดิสก์
200-207	เกมไอโอ
278-27F	พอร์ตเครื่องพิมพ์หมายเลข 2
2F8-2FF	พอร์ตอนุกรมหมายเลข 2
300-31F	โปรโตไทป์การ์ด
360-36F	สำรอง
378-37F	พอร์ตเครื่องพิมพ์หมายเลข 1
380-38F	SDLC, ไบซิงค์ 2
3A0-3AF	ไบซิงค์ 1
3B0-3BF	โมโนโครมและเครื่องพิมพ์
3C0-3CF	สำรอง
3D0-3DF	จอภาพสี
3F0-3F7	ควบคุมดิสเกตต์

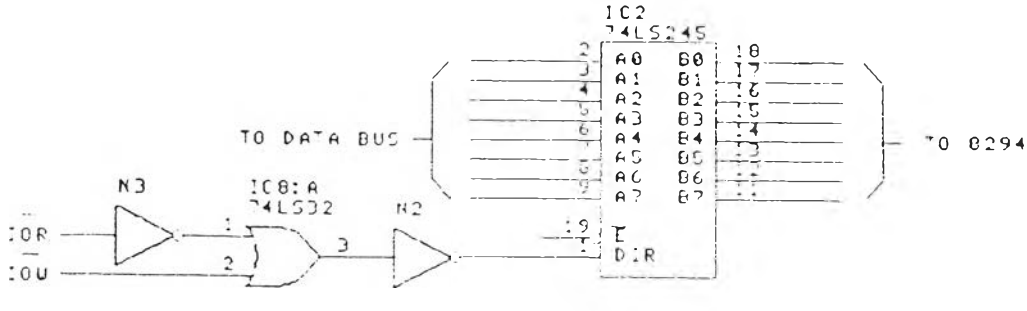
ตารางที่ 5.2 ตำแหน่งแอดเดรสในแต่ละกลุ่ม

กลุ่ม	แอดเดรส
0 (Y0)	0300H - 0307H
1 (Y1)	0308H - 030FH
2 (Y2)	0310H - 0317H
3 (Y3)	0318H - 031FH
4 (Y4)	0320H - 0327H
5 (Y5)	0328H - 032FH
6 (Y6)	0330H - 0337H
7 (Y7)	0338H - 033FH

เราใช้การถอดรหัสแอดเดรสในกลุ่ม 7 หรือ Y7 คือใช้สัญญาณเอาท์พุทจากขา Y7 ของ IC 74LS138 ไปเป็นตัวเลือกให้ IC 8294 หรือตัวเข้ารหัสทำงาน (Chip Select หรือ CS)

2. วงจรสร้างสัญญาณควบคุมการส่งถ่ายข้อมูล

วงจรสร้างสัญญาณควบคุมการส่งถ่ายข้อมูลทำหน้าที่รับสัญญาณอ่านหรือเขียนอุปกรณ์ I/O (IOR หรือ IOW) จากบัสควบคุม (Control Bus) ของเครื่องคอมพิวเตอร์ มาทำการจัดวงจรตามรูป ให้เกิดช่วงเวลาที่เหมาะในการส่งถ่ายข้อมูลนอกจากนี้ยังเป็นตัวกำหนดทิศทางการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์และ IC 8294 ด้วย โดยอาศัย IC 74LS08 และ 74LS32 มาเป็นตัวควบคุมดังรูป 5.6

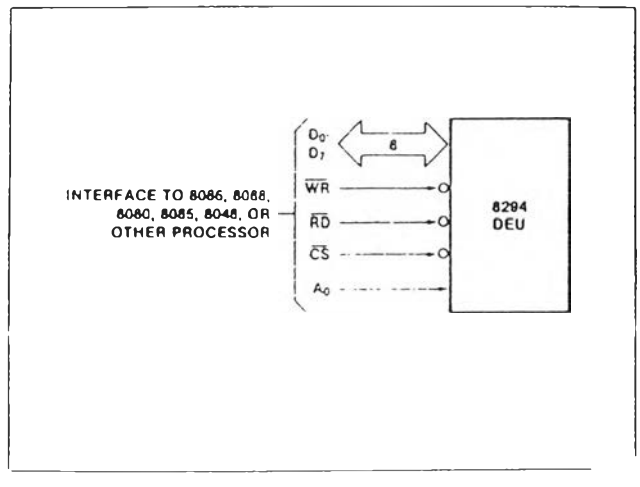


รูป 5.6 อุปกรณ์สร้างสัญญาณควบคุมการส่งถ่ายข้อมูล

3. วงจรเข้ารหัสข้อมูล

เป็นอุปกรณ์ที่มีหน้าที่หลักในการทำการเข้าหรือถอดรหัสข้อมูล โดยอาศัยการควบคุมการทำงานจากโปรแกรมผ่านทางอุปกรณ์ในหัวข้อ 5.2.1 และ 5.2.2 ในการออกแบบเราใช้ IC เบอร์ 8294 ซึ่งเป็น Data Encryption Unit โดยมีอัลกอริธึ่มในการทำ DES ที่ออกแบบโดย NBS มีความสามารถในการเข้าหรือถอดรหัสด้วยอัตรา 80 ไบท์/วินาที รายละเอียดส่วนต่าง ๆ ของ IC 8294 นี้ดูได้จากภาคผนวก ก.

สำหรับรูปแบบของการต่อวงจรใช้งานนั้น เราเลือกการเชื่อมต่อแบบ Polling Interface [ภาคผนวก ก.] โดยมีลักษณะดังรูป 5.7



รูป 5.7 การต่อวงจรแบบ Polling Interface

ซึ่งการทำงานของการต่อลักษณะนี้คือการ ใช้โปรแกรมวนรอสัญญาณที่ส่งออกมาจากรีจิสเตอร์ แสดงสถานะของDEU เข้ามาเป็นเงื่อนไขในการทำงานต่อไป ก่อนที่จะกล่าวถึงสถานะต่าง ๆ ที่แสดงออกมา จะขอกล่าวถึงรีจิสเตอร์ต่าง ๆ ของ DEU ก่อน

รีจิสเตอร์ภายใน DEU จะมี 4 รีจิสเตอร์ เป็นอินพุท 2 รีจิสเตอร์ และเอาท์พุท 2 รีจิสเตอร์ ดังตารางที่ 5.3

ตาราง 5.3 รีจิสเตอร์ภายใน DEU

RD	WR	CS	A_0	Register
1	0	0	0	Data Input Buffer
0	1	0	0	Data Ouput Buffer
1	0	0	1	CommandInputBuffer
0	1	0	1	Status Output Buffer
X	X	1	X	Don'tCare

หน้าที่ของแต่ละรีจิสเตอร์มีดังนี้

Data Input Buffer ข้อมูลที่ถูกเขียนมาที่รีจิสเตอร์นี้ จะถูกตีความได้ว่าจะเป็นอะไรใน 3 กรณีนี้ขึ้นอยู่กับคำสั่งที่ส่งมาที่ Command Input Buffer

1. คีย์
2. ข้อมูลที่จะถูกเข้าหรือถอดรหัส
3. จำนวนบล็อคของกระบวนการ DMA (ในการจัดวงจรของเราไม่ได้ใช้กระบวนการของ DMA)

Data Output Buffer ข้อมูลที่ถูกอ่านจากรีจิสเตอร์นี้จะเป็นเอาท์พุทของการเข้าหรือถอดรหัส

Command Input Buffer คำสั่งที่ส่ง DEU จะถูกเขียนลงมาที่รีจิสเตอร์นี้ ซึ่งจะประกอบด้วยการใส่คีย์ การสั่งให้เข้ารหัสหรือถอดรหัส

Status Output Buffer สถานะของ DEU จะแสดงอยู่ในรีจิสเตอร์ตัวนี้ ซึ่งจะเป็นที่ที่โปรแกรมจะมาอ่านสถานะนี้ไปเป็นเงื่อนไขการทำงาน ในกรณีที่ต่อวงจรแบบ Polling Interface

Status Bit	: 7	6	5	4	3	2	1	0
Function	: X	X	X	KPE	CF	DEC	IBF	OBF

OBF : Output Buffer Full แสดงว่ามีข้อมูลรอให้อ่านที่ Data Output Buffer

IBF : Input Buffer Full แสดงว่ามีข้อมูลเข้าไปที่ Data Input Buffer หรือ Command Input Buffer

DEC : Decryption ถ้าบิตนี้อยู่ในสถานะ"HIGH"แสดงว่าอยู่ในโหมดของการถอดรหัส

CF : Completion Flag แสดงว่าการทำการส่งถ่ายข้อมูลหรือการใส่คีย์ถูกต้องสมบูรณ์

KPE : Key Parity Error ถ้าบิตนี้อยู่ในสถานะ"HIGH"แสดงว่าการใส่คีย์มีข้อผิดพลาด

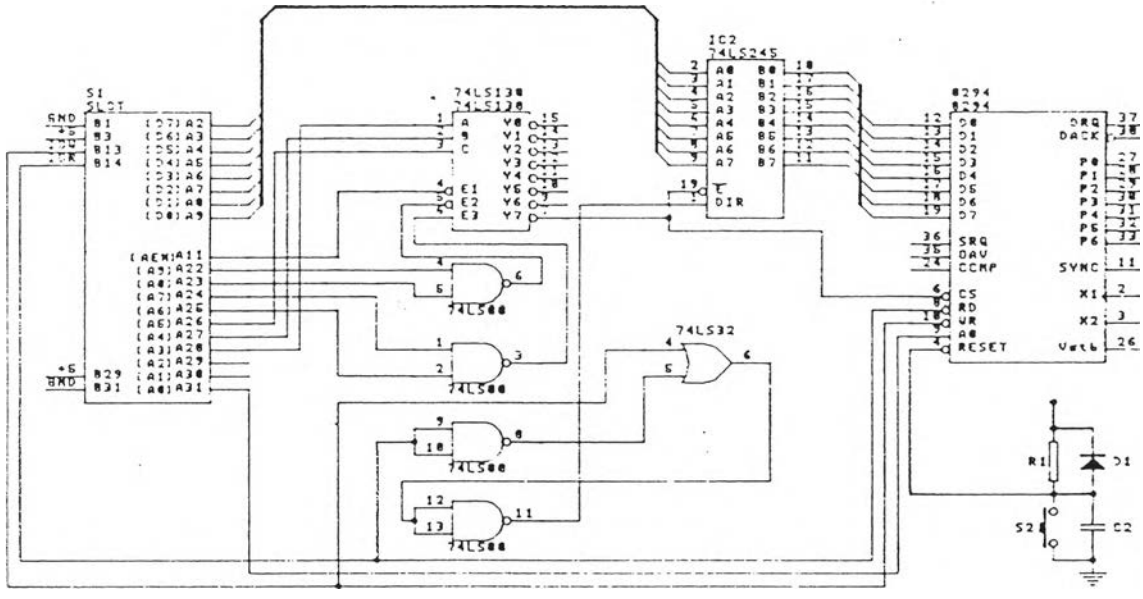
จากรายละเอียดของรีจิสเตอร์ทั้ง 4 ตัว ทำให้เราสามารถกำหนด Address ที่จะใช้ในการควบคุมการทำงานของ DEU ได้ โดยดูจากตาราง 5.3 (รีจิสเตอร์ภายใน DEU) จะเห็นว่า รีจิสเตอร์ทั้ง 4 ตัวจะทำงานได้ CS จะต้อง Active นั้น คือขา Y1 ในหัวข้อ 5.2.1 จะต้องมีสถานะเป็น "LOW" ซึ่งก็อยู่ในกลุ่มของ Address 0338H-033FH หรือ

A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	0	0	1	1	1	X	X	X
3		3				8-F			

เราจะเป็นผู้เลือกว่าจะติดต่อกับรีจิสเตอร์ตัวใดใน DEU โดยการใช้ขา A_0 ร่วมกับขา RD และ WR จะสังเกตได้ว่าถ้า A_0 เป็น 0 จะเป็นติดต่อกับ Data Buffer Register ส่วนจะเป็น Data Input Buffer หรือ Data Output Buffer จะกำหนดได้ด้วยขา WR หรือ RD ในลักษณะเดียวกัน เมื่อ A_0 เป็น 1 เราจะติดต่อกับ Command Input Butter หรือ Status Output Buffer ได้โดยอาศัย WR และ RD เป็นตัวควบคุมดังนั้นขา A_2 และ A_1 จึงไม่ถูกใช้งาน หรือจะเป็นสถานะอะไรก็ได้ แต่เพื่อความไม่สับสนในการเขียนโปรแกรม เรากำหนดให้รีจิสเตอร์ภายใน DEU มี Address ประจำตัวดังนี้

Data Input Buffer	อยู่ที่แอดเดรส	0338H
Data Output Buffer	อยู่ที่แอดเดรส	033AH
Command Input Buffer	อยู่ที่แอดเดรส	0339H
Status Output Buffer	อยู่ที่แอดเดรส	033BH

จากรายละเอียดในหัวข้อ1(วงจรถอดรหัสแอดเดรส) หัวข้อ2(วงจรสร้างสัญญาณควบคุมการส่งถ่ายข้อมูล)และรายละเอียดในหัวข้อนี้ทำให้เราได้วงจรที่สมบูรณ์ดังรูป 5.8



รูป 5.8 วงจรสมบูรณ์ของเอนคริปเตอร์และออเทนทิเคเตอร์

<u>รายละเอียดทางด้านซอฟท์แวร์</u>

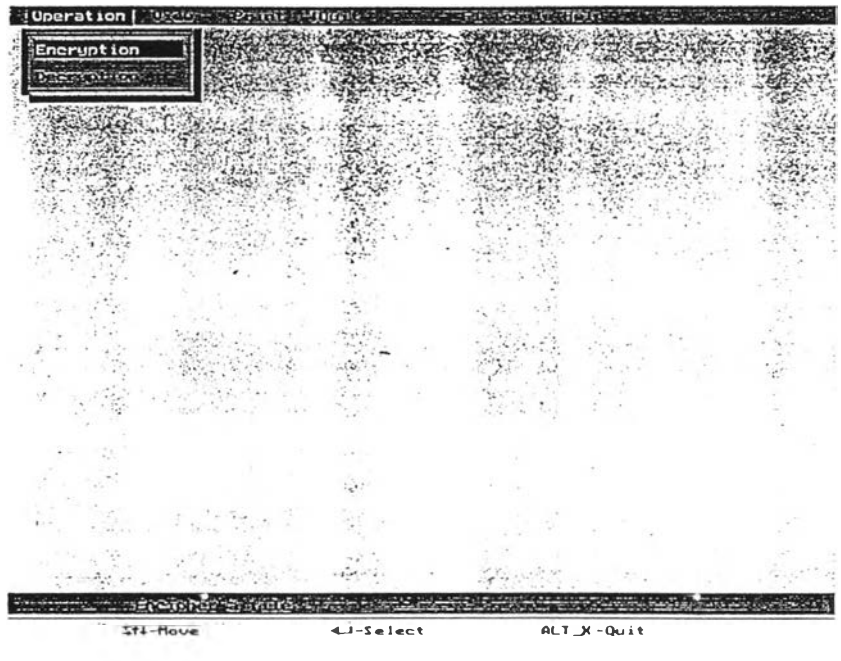
การเขียนโปรแกรมเพื่อควบคุมการทำงานของเครื่องเอนคริปเตอร์ และออเทนทิเคเตอร์ จะใช้ภาษาซี ซึ่งมีข้อดีคือ เป็นภาษาที่มีลักษณะเป็นโครงสร้าง สามารถออกแบบให้การทำงานในแต่ละส่วนสามารถเขียนเป็นฟังก์ชันเล็ก ๆ ได้ ซึ่งจะทำให้โปรแกรมมีความยืดหยุ่น สามารถแก้ไขหรือเพิ่มเติมความสามารถต่าง ๆ ได้โดยง่าย นอกจากนี้ตัวคอมไพเลอร์ ยังมีส่วนอรรถประโยชน์ (Utility) ที่ช่วยให้การพัฒนาโปรแกรมต่าง ๆ เป็นไปได้โดยสะดวก

จากความเป็นโครงสร้างของภาษาซี จึงได้นำข้อดีดังกล่าวมาเป็นต้นแบบในการพัฒนาโปรแกรม โดยสามารถแบ่งโปรแกรมออกเป็น 3 ส่วนหลัก ๆ ได้ดังนี้คือ

1. โปรแกรมหลัก
2. โปรแกรมที่ทำหน้าที่เข้ารหัส ถอดรหัส และรับรองข้อความ
3. โปรแกรมอรรถประโยชน์

1. โปรแกรมหลัก

โปรแกรมหลักทำหน้าที่เป็นตัวกำหนดตัวแปรต่าง ๆ ทั้งหมดที่ใช้ในโปรแกรมทำการตรวจจับว่ามอนิเตอร์ที่ใช้เป็นชนิดใด เพื่อที่จะเลือกค่าพารามิเตอร์สำหรับโปรแกรมให้ตรงกับมอนิเตอร์ที่ใช้งาน หลังจากนั้นก็จะแสดงเมนูหลักที่หน้าจอ โดยมีลักษณะเป็น Menu Driven สามารถให้ผู้ใช้เลือกทำงานต่อไปได้โดยการใช้ปุ่มลูกศรที่แป้นพิมพ์นอกจากนี้ยังจัดการเกี่ยวกับการแสดงข้อความแนะนำสั้น ๆ สำหรับในแต่ละเมนูที่เราเลื่อนแถบเน้น (Highligh Bar) ผ่านไว้ที่ด้านล่างของหน้าจอ ดังรูป 5.9



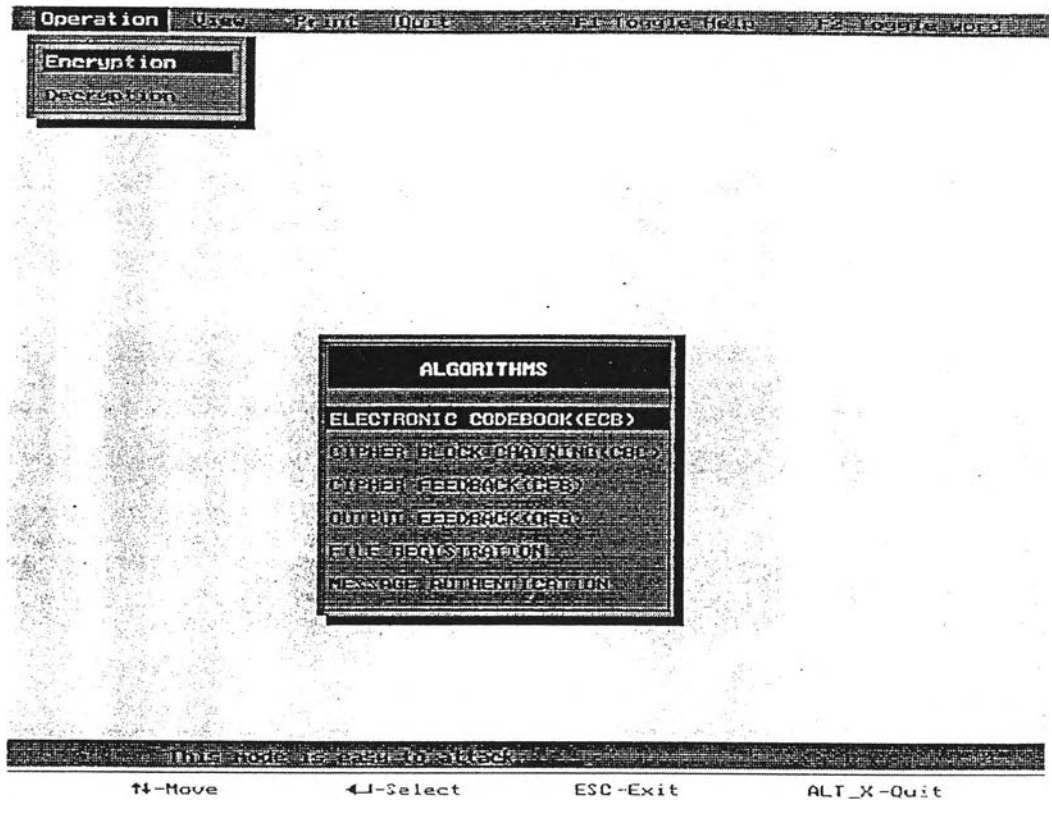
รูป 5.9 เมนูหลัก

2. โปรแกรมที่ทำหน้าที่เข้ารหัส ถอดรหัส และรับรองข้อความ

โปรแกรมในส่วนนี้พัฒนาขึ้นมาเพื่อเป็นการควบคุมและทำงานร่วมกับเอนคริปเตอร์และออเทนทิเคเตอร์โดยสามารถจะเข้ามาใช้โปรแกรมส่วนนี้ได้ทางเมนู Operation ที่อยู่ในเมนูหลัก ในเมนู Operation จะแบ่งออกเป็น 2 เมนูย่อย คือ

1. Encryption
2. Decryption

เมื่อเลือกการทำงานโดยการเลื่อนแถบดำมาที่ต้องการแล้วกดEnterแล้วโปรแกรมจะแสดงหน้าต่างตามที่เลือกไว้ ดังรูป 5.10



รูป 5.10 เมนู Encryption

เมื่อเข้าสู่เมนูย่อยในลำดับขั้นของEncryptionแล้วหากต้องการเปลี่ยนเป็นการDecryptionเราสามารถจะกระทำได้โดยการ กดคีย์ ESC เพื่อกลับไปสู่เมนูก่อนหน้านี้ แล้วเลือก Decryption

เมื่อเลือก Operation ของการ Encryption หรือ Decryption แล้ว ตัวโปรแกรมจะรอรับการเลือกในลำดับขั้นต่อไป และอัลกอริธึ่มสำหรับการรับรองข้อความอีก 2 เมนู คือ

1. Electronic Codebook (ECB)
2. Cipher Block Chaining (CBC)
3. Cipher Feedback (CFB)
4. Output Feedback (OFB)
5. File Registration
6. Message Authentication

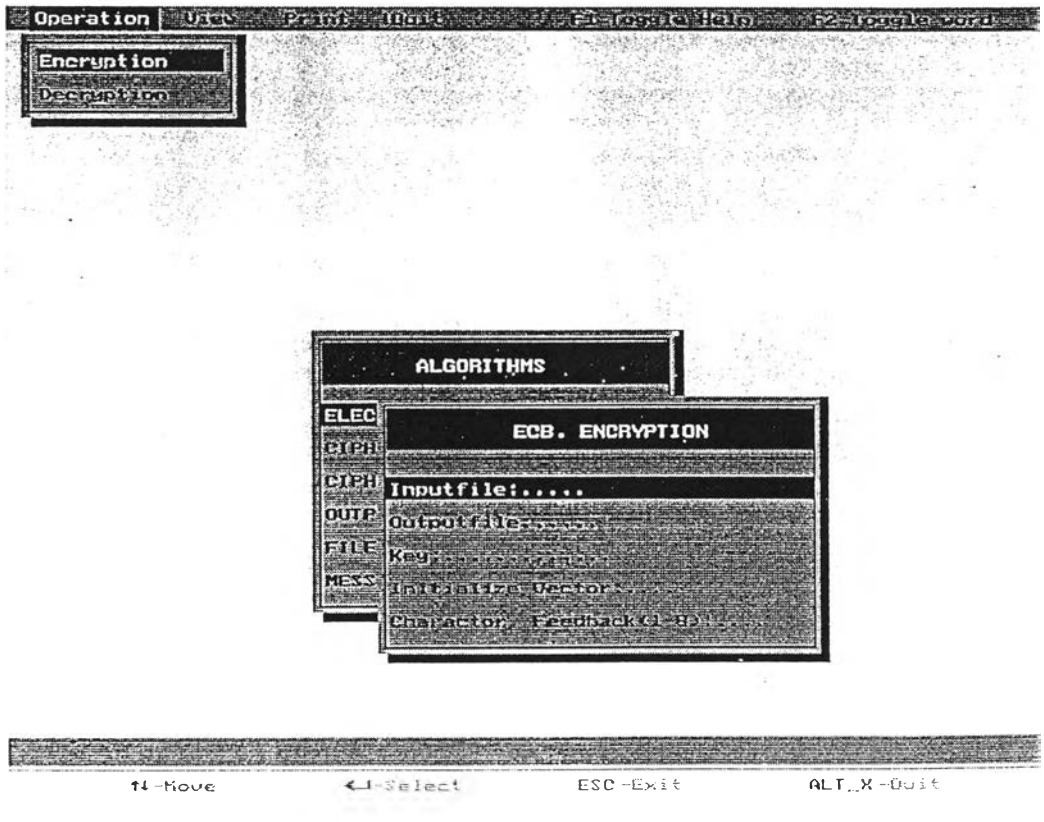
รายละเอียดของการทำงานของเมนูต่าง ๆ มีดังนี้

2.1 Electronic Codebook (ECB)

เมื่อเลือกเมนูนี้จะเป็นการสั่งให้เอนคริปเตอร์ทำงานในโหมด ECB ซึ่งเป็นโหมดการเข้ารหัสนี้เป็นพื้นฐานที่สุด โดยพารามิเตอร์ที่ใช้ในการเข้ารหัสโหมดนี้มีเพียงคีย์เท่านั้นนั่นคือโปรแกรมจะเรียกหน้าต่างสำหรับใส่ค่าพารามิเตอร์ขึ้นมา ซึ่งประกอบด้วย

1. Input File หรือ ไฟล์ที่ใช้สำหรับจะทำการเข้าหรือถอดรหัส
2. Output File หรือ ไฟล์ที่ใช้สำหรับเก็บค่าไฟล์ที่เข้าหรือถอดรหัสแล้ว
3. Key หรือ กุญแจที่ใช้สำหรับเข้าหรือถอดรหัส ซึ่งต้องใช้ทั้งหมด 8 ตัวอักษร (64 บิต)

เมนูในข้อ 1.1 ถึง 1.3 นั้น จะแสดงด้วยตัวอักษรสีดำส่วนเมนูที่เหลือซึ่งไม่ได้ใช้งานในโหมดนี้ จะแสดงไว้ด้วยตัวอักษรสีแดง หรือสีที่จางกว่า ดังรูป 5.11



รูป 5.11 เมนู Encryp/ECB

2.2 Cipher Block Chaining (CBC)

เมื่อเลือกเมนูนี้ จะเป็นการสั่งให้เอนคริปเตอร์ทำงานในโหมด CBC ซึ่งเป็นการเข้ารหัสเป็นบล๊อคที่มีการเชนนิ่งเพื่อขจัดปัญหาการมีรูปแบบของข้อมูลในการทำการเชนนิ่ง ต้องมีการกำหนดค่าเวกเตอร์เริ่มต้น ดังนั้นหน้าต่างที่ปรากฏจะต้องใส่พารามิเตอร์ 4 ตัว คือ

1. Input File
2. Output File
3. Key
4. Initialize Vector หรือค่าเริ่มต้นที่ประกอบด้วยตัวอักษร 8 ตัว (64 บิต) เพื่อทำการ Exclusive of กับเพลนเท็กซ์ชุดแรก ก่อนที่จะทำการเข้ารหัส

2.3 Cipher Feedback (CFB)

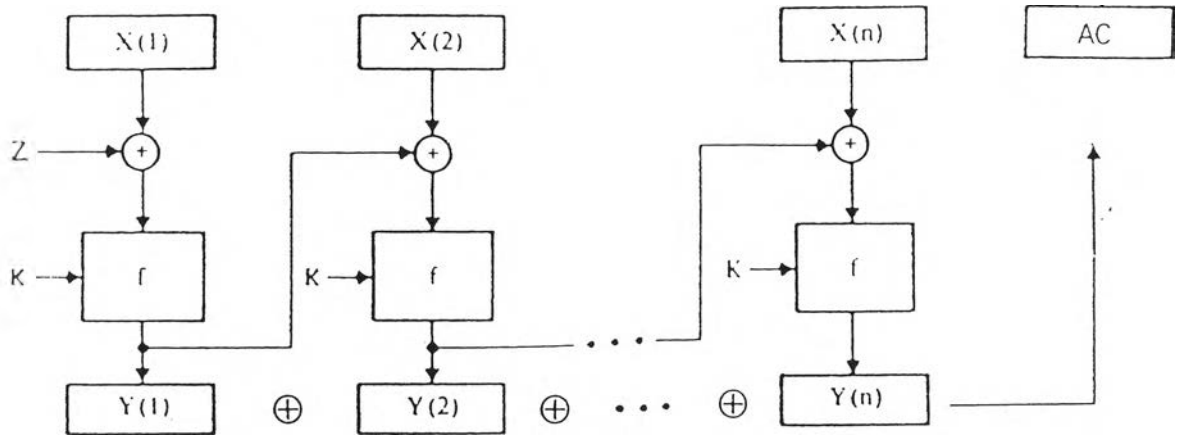
เมื่อเลือกเมนูนี้ จะเป็นการเข้าหรือถอดรหัสในโหมด CFB ซึ่งเป็นการเข้ารหัสแบบสตรีมไซเฟอร์ที่มีการเชนนิ่งการเข้ารหัสวิธีนี้จะใช้พารามิเตอร์เหมือนกับในโหมดCBCแต่เนื่องจากเป็นสตรีมไซเฟอร์การเชนนิ่งจะเกิดจากทุกตัวอักษร (8 บิต) ดังนั้นจำนวนครั้งที่ใช้ในการเข้ารหัสจึงมากกว่าแบบบล๊อคเชนนิ่ง8เท่าซึ่งก็ทำให้เวลาที่ใช้ในการเข้ารหัสมากกว่าประมาณ 8 เท่าด้วย

2.4 Output Feedback (OFB)

เมื่อเลือกเมนูนี้ต้องใส่ค่าพารามิเตอร์ทั้งหมด 5 ตัวกล่าวคือต้องใส่ค่าพารามิเตอร์ในเมนู Charactor Feedback ซึ่งหมายถึง จำนวนตัวอักษรที่จะทำการป้อนกลับมายังด้านอินพุทในแต่ละครั้งของการเข้ารหัส ซึ่งมีจำนวนตั้งแต่ 1 ถึง 8 ตัว (8 ถึง 64 บิต) ซึ่งค่านี้จะมีผลต่อความเร็วในการเข้ารหัสของเครื่องเข้ารหัส กล่าวคือ จะทำให้เข้ารหัสได้ช้ากว่า 8 ถึง 1 เท่า เมื่อเทียบกับ CBC

2.5 File Registration and Message Authentication

จาก [Meyer and Matyas, 1982] การทำ Message Authentication หรือ การรองรับข้อความนั้น AC จะถูกสร้างขึ้นจาก การเพลนเท็กซ์มาบวกกันแบบโมดูโล-2 ดังรูป 5.12



รูป 5.12 การรับรองข้อความโดยสร้าง AC จากเพลนเท็กซ์ [Meyer and Matyas, 1982]

จะเห็นว่า การที่จะทำการปลอมแปลง AC (ในรูปคือ $\Delta(M)$) หรือเปลี่ยนแปลงเพลนเท็กซ์ โดยที่ AC ยังคงเดิมทำได้โดยง่ายโดยการแก้ไขข้อในตำแหน่งเดียวกันของ2 บล็อคของเพลนเท็กซ์ใดๆ หรือโดยการเพิ่มเพลนเท็กซ์บล็อคใดๆที่เหมือนกัน เป็นจำนวนคู่ของบล็อค (เช่น 2, 4, ...) โดยที่ไม่สามารถที่จะตรวจสอบได้ว่าเพลนเท็กซ์ เกิดการเปลี่ยนแปลงขึ้น ซึ่งกรณีดังกล่าวสามารถแสดงได้โดยสมการทางคณิตศาสตร์ดังนี้

$$x = x_1, x_2, x_3, \ldots, x_n \tag{5.1}$$

x = เพลนเท็กซ์

เมื่อทำการรับรองข้อความด้วยวิธีในรูป 4.5 แล้วส่งไปยังผู้รับ

$$E_k(x) = E_k(x_1, x_2, x_3, \ldots, x_n, AC)$$
$$Y = y_1, y_2, y_3, \ldots, y_n, E_k(AC) \tag{5.2}$$

y คือ ไซเฟอร์เท็กซ์

AC คือ $x_1 \oplus x_2 \oplus x_3 \oplus \ldots \oplus x_n$

เมื่อผู้รับถอดรหัส $D_k(y) = D_k(y_1, y_2, y_3, \ldots, y_n, E_k(AC))$

$$x = x_1, x_2, x_3, \ldots, x_n, AC \qquad (5.3)$$

เมื่อนำ $x_1 \oplus x_2 \oplus x_3 \oplus \ldots \oplus x_n$ มาเปรียบเทียบกับ AC จะได้ค่าที่เท่ากัน แสดงว่าข้อมูลหรือไฟล์ไม่ถูกแก้ไข แต่ถ้าเพลนเท็กซ์ X ในฝ่ายผู้ส่งถูกแก้ไขในบล็อค 2 บล็อค โดยการแก้ไขดังกล่าวมีเงื่อนไขคือ

$$x_i \oplus x_j = x_i' \oplus x_j'$$

$x_{i,j}'$ คือ บล็อคที่ถูกแก้ไขแล้วเมื่อส่งข้อมูลดังกล่าวไปยังผู้รับ

$$E_k(x') = E_k(x_1', x_2', x_3, \ldots, x_n, AC)$$

$$Y = y_1', y_2', y_3, \ldots, y_n, E_k(AC)$$

(5.4)

เมื่อผู้รับถอดรหัส

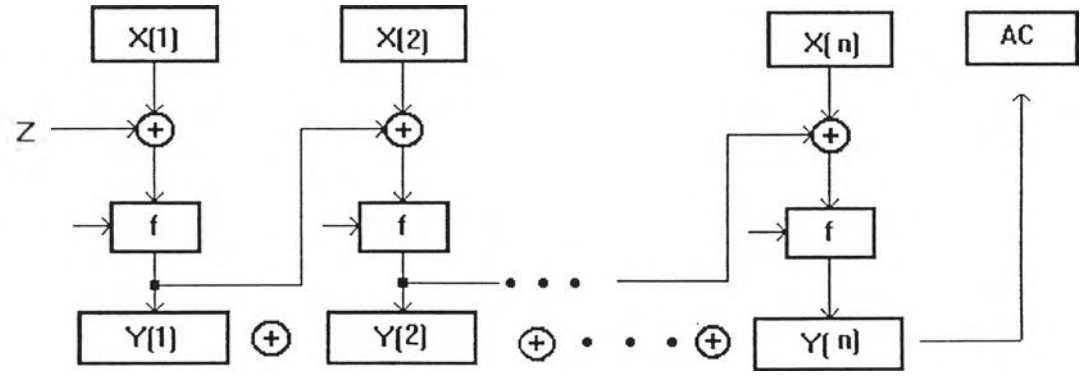
$$D_k(Y) = D_k(y_1', y_2', y_3, \ldots y_n, E_k(AC))$$

$$x' = x_1', x_2', x_3, \ldots, x_n, AC$$

(5.5)

เมื่อนำ $x_1' \oplus x_2' \oplus x_3 \oplus \ldots \oplus x_n$ มาเปรียบเทียบกับ AC ก็จะไม่แตกต่างกัน ซึ่งผู้รับก็จะยอมรับว่าข้อมูลที่ได้รับ คือข้อมูลที่ถูกต้อง ทั้งที่ถูกแก้ไขมาแล้ว ซึ่งในกรณีนี้ถือเป็นข้อด้อยของอัลกอริธึ่มนี้

ดังนั้นในวิทยานิพนธ์ฉบับนี้จึงได้เสนอวิธีแก้ไขกรณีดังกล่าว โดยการสร้าง AC จากไซเฟอร์เท็กซ์ ซึ่งเกิดขึ้นอย่างสุ่ม แทนการใช้เพลนเท็กซ์ ดังแสดงในรูป 5.13



รูป 5.13 การสร้าง AC ใน File Registration และ Message Authentication

นอกจากนั้นแล้ว ยังได้เสนอให้มีการทำ File Registration หรือ ลงทำเบียนไฟล์เพื่อป้องกันข้อมูลซึ่งอยู่ในระหว่างการรอการทำการรับรองข้อความถูกแก้ไขไปในกรณีที่ข้อมูลนั้นถูกเก็บไว้ในที่ที่ไม่มีการควบคุมการเข้าถึง

การลงทะเบียนไฟล์ คือการสร้าง AC จากข้อมูลในไฟล์เพื่อใช้เป็นตัวตรวจสอบว่าหลังจากลงทะเบียนไฟล์แล้วข้อมูลถูกแก้ไขหรือไม่ AC นี้จะถูกสร้างตามรูป 5.13 โดยที่

กำหนดเพลนเท็กซ์

$$x = x_1, x_2, x_3, \ldots, x_n \tag{5.6}$$

เมื่อทำการลงทะเบียนไฟล์ จะได้เพลนเท็กซ์ x_R

$$x_R = x_1, x_2, x_3, \ldots, x_n, AC_R \tag{5.7}$$

โดยที่

$$AC_R = (E_k(x_1) \oplus E_k(x_2) \oplus \ldots \oplus E_k(x_n)) \tag{5.8}$$

สมมติว่าเกิดการแก้ไขเพลนเท็กซ์ X_R เป็น $x_R^{'}$ โดย

$$x_R^{'} = x_1^{'}, x_2^{'}, x_3, \ldots, x_n, AC_R \tag{5.9}$$

เมื่อเพลนเท็กซ์ $x_R^{'}$ หรือไฟล์ที่ลงทะเบียนแล้วถูกแก้ไข ถูกนำมาทำการรับรองข้อความเพื่อส่งออกไปยังผู้รับ โดยวิธีการคือสร้าง $AC_R^{'}$ ขึ้นมาเปรียบเทียบกับ AC_R ที่ได้จากการลงทะเบียนในสมการที่ (5.8)

$$AC_R^{'} = (E_k(x_1^{'}) \oplus E_k(x_2^{'}) \oplus \ldots \oplus E_k(x_n)) \tag{5.10}$$

เนื่องจาก AC_R ในสมการ (5.8) และ $AC_R^{'}$ ในสมการ (5.10) ต่างมีขนาด 64 บิต ดังนั้นโอกาสที่จะแก้ไข $x_R^{'}$ เพื่อให้ $AC_R^{'}$ มีค่าเท่ากับ AC_R จึงมีความน่าจะเป็นเพียง 2^{-64} ซึ่งหมายความว่าไฟล์ที่ผ่านการทำการลงทะเบียนแล้วถูกแก้ไขจะมีความเป็นไปได้น้อยมากที่จะตรวจสอบไม่พบ

หลังจากที่ทำการลงทะเบียนไฟล์ไว้แล้ว เพลนเท็กซ์จะมีค่าตามสมการ 5.7 เราจะทำการรับรองข้อความนี้โดยใช้ อัลกอริธึ่มเดียวกับการลงทะเบียนไฟล์ เพื่อสร้างรหัสรับรองข้อความ

จากสมการ 5.7 $x_R = x_1, x_2, \ldots, x_n, AC_R$

เมื่อทำการรับรองข้อความโดยสร้างรหัสรับรองข้อความตามรูป 5.13 จะได้

$$Y_A = E_k(x_1), E_k(x_2), \ldots, E_k(x_n), E_k(AC_R), AC_A$$

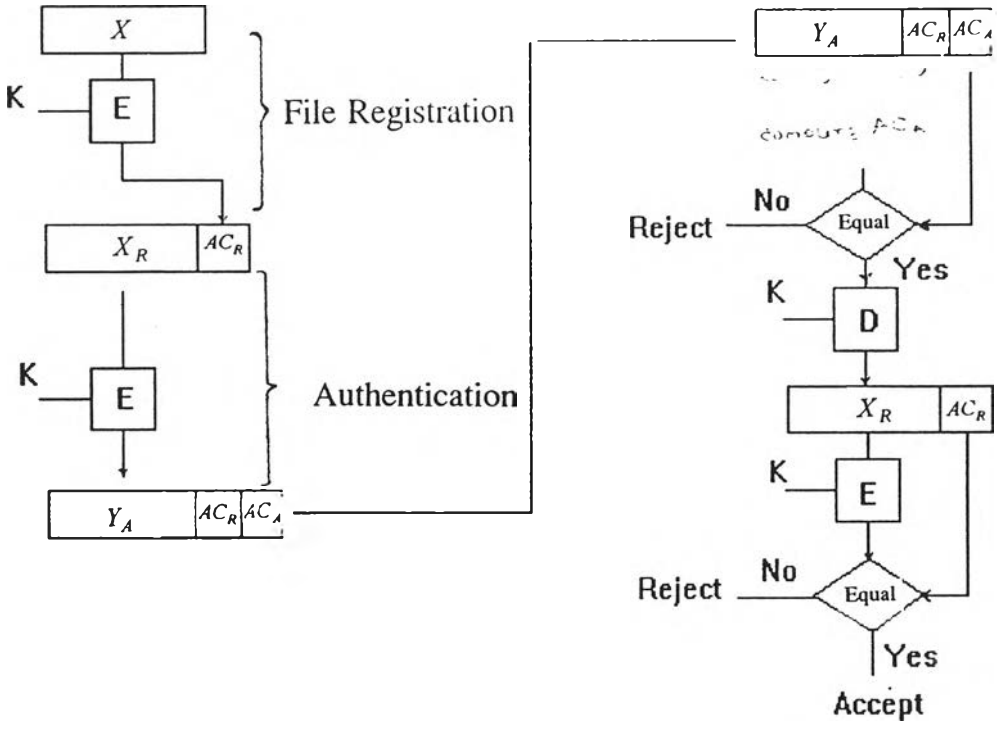
$$Y_A = y_1, y_2, \ldots, y_n, y_{AC_R}, AC_A$$

โดยที่ Y_A คือไซเฟอร์เท็กซ์ที่เกิดจากการรับรองข้อความ

และ $AC_A^{'} = y_1 \oplus y_2 \oplus \ldots \oplus y_n \oplus y_{AC_R}$

Y_A อาจจะถูกแก้ไขในระหว่างส่งผ่านช่องสื่อสารแต่ทางผู้รับสามารถที่จะตรวจสอบได้จากการเปรียบเทียบค่า AC_A ที่ส่งมากับค่าที่คำนวนขึ้นมาจากไซเฟอร์เท็กซ์ที่รับมาได้ในลักษณะบิตต่อบิตซึ่งการที่จะแก้ไข Y_A โดยที่ AC_A ไม่เปลี่ยนแปลงนั้นเป็นไปได้ยากมากเนื่องข้อมูลในแต่ละบล็อคของ Y_A เป็นข้อมูลที่เกิดจากการเข้ารหัส

วิธีการในการลงทะเบียนและรับรองข้อความที่นำเสนอนี้แสดงอยู่ในรูป 5.14



รูป 5.14 วิธีการลงทะเบียนและรับรองข้อความ

จะเห็นว่า เราต้องสร้าง AC ขึ้น 2 ค่า คือในขั้นตอนของการลงทะเบียนไฟล์ และในขั้นตอนของการทำการรับรองข้อความ ซึ่งทำให้ต้องใช้เวลาในการคำนวนมากกว่าวิธีที่เสนอใน [Meyer and Matyas, 1982] แต่ในแง่ของความปลอดภัยแล้ววิธีที่นำเสนอนี้จะมีข้อดีกว่าคือ การลงทะเบียนไฟล์สามารถป้องกันการเปลี่ยนแปลงแก้ไขข้อมูลทั้งในขณะที่ยังไม่ได้ผ่านช่องสื่อสาร และการรับรองข้อความโดยใช้ไซเฟอร์เท็กซ์ เป็นตัวสร้าง AC_A สามารถป้องกันการแก้ไขข้อมูลในขณะที่ข้อมูลถูกส่งผ่านช่องสื่อสารไปยังผู้รับ ซึ่งวิธีการลงทะเบียนไฟล์และรับรองข้อความนี้ ผู้วิจัยได้ศึกษาจากเอกสารต่างๆ (ตามรายการอ้างอิงท้ายเล่ม) แล้ว ยังไม่ปรากฏว่ามีการกล่าวถึง

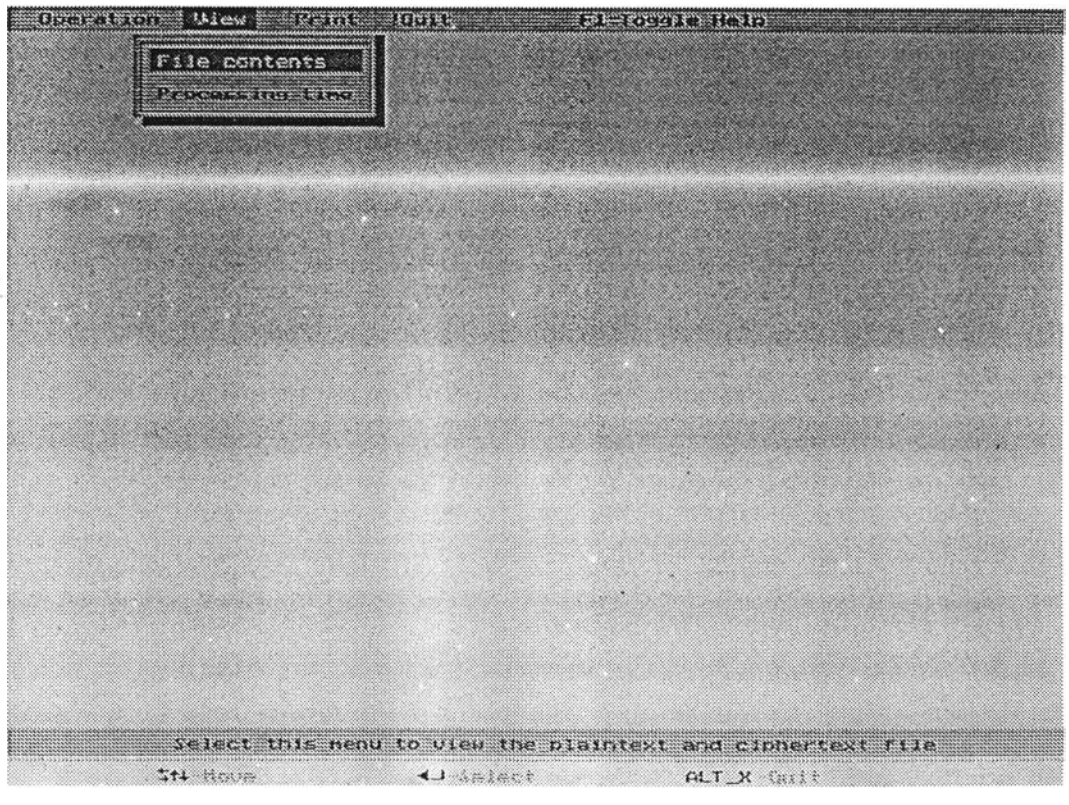
3. โปรแกรมอรรถประโยชน์

โปรแกรมอรรถประโยชน์ถูกพัฒนาขึ้นมาเพื่อช่วยให้ผู้ใช้มีความคล่องตัวในการที่จะดูข้อความก่อนหรือหลังจากการเข้ารหัส ดูเวลาที่ใช้ไป ขนาดของไฟล์ หรือพิมพ์ข้อความออกทางเครื่องพิมพ์โดยไม่ต้องใช้โปรแกรมอื่น โปรแกรมอรรถประโยชน์ที่พัฒนาขึ้นมามี 2 โปรแกรม คือ

1. โปรแกรม View
2. โปรแกรม Print

3.1 โปรแกรม View

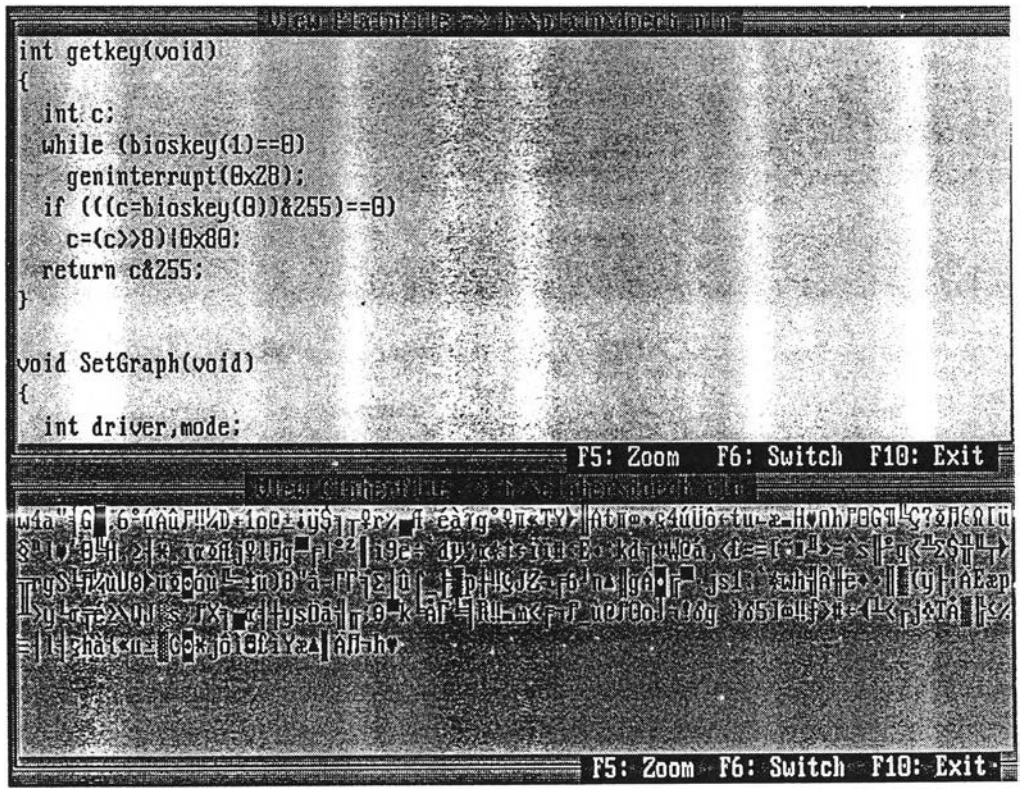
เป็นโปรแกรมที่ใช้สำหรับดูข้อความ หรือดูขนาดของไฟล์ รวมทั้งเวลาที่ใช้ในการเข้ารหัสของแต่ละโหมดการใช้งานทำได้โดยเลือกเมนู View จากเมนูหลัก จากนั้นโปรแกรมจะให้เลือกต่อไปว่าจะดูข้อความหรือเวลาที่ใช้ในการเข้ารหัส ดังแสดงในรูป 5.15



รูป 5.15 เมนูใน View

3.1.1 File Contents

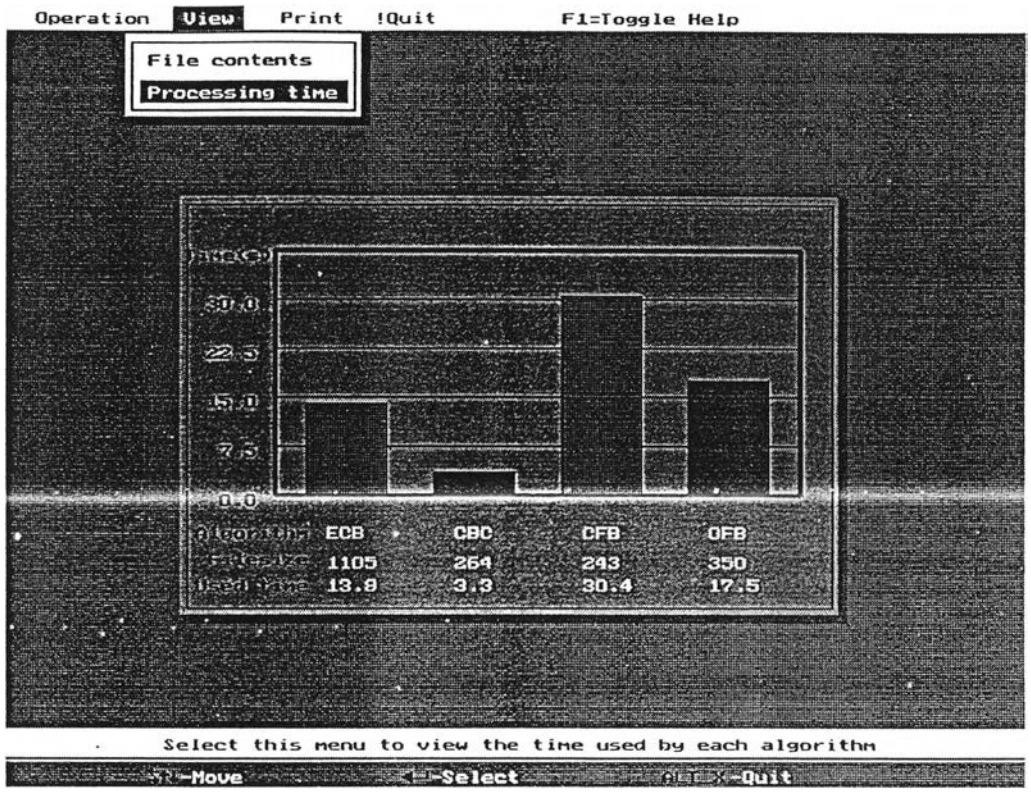
เมื่อเลือกเมนูนี้แล้วโปรแกรมจะไปเรียกใช้งานโปรแกรมภายนอก ซึ่งผู้วิจัยได้พัฒนาขึ้นมาแสดงที่หน้าจอ เพื่อรอให้ผู้ใช้ใส่ชื่อไฟล์ที่ต้องการจะดูข้อความในไฟล์ ซึ่งสามารถที่จะดูได้ทั้งก่อนที่จะทำการเข้ารหัสและหลังจากเข้ารหัสแล้วในโปรแกรมนี้ยังได้ออกแบบให้ผู้ใช้สามารถที่จะดูทั้ง 2 ไฟล์ในจอเดียวกันหรือทีละไฟล์ก็ได้และยังสามารถเลือกที่จะดูบรรทัดใดหรือหน้าใดก็ได้ ดังแสดงในรูป 5.16



รูป 5.16 โปรแกรม File Contents

3.1.2 Processing Time

เมนูนี้จะเป็นการแสดงให้เห็นถึงขนาดของไฟล์ และเวลาที่ใช้ในการเข้ารหัส โดยได้แสดงเวลาในรูปของกราฟแท่งที่มีแกนนอนเป็นโหมดของการเข้ารหัส และแกนตั้งเป็นเวลาที่ใช้ไปในหน่วยของวินาที ซึ่งค่าสูงสุดที่ได้ออกแบบมาคือ 37.5 วินาที ในกรณีที่การเข้ารหัสที่ใช้เวลาเกินกว่านี้ โปรแกรมก็จะแจ้งให้ผู้ใช้ทราบ ดังรูป 5.17

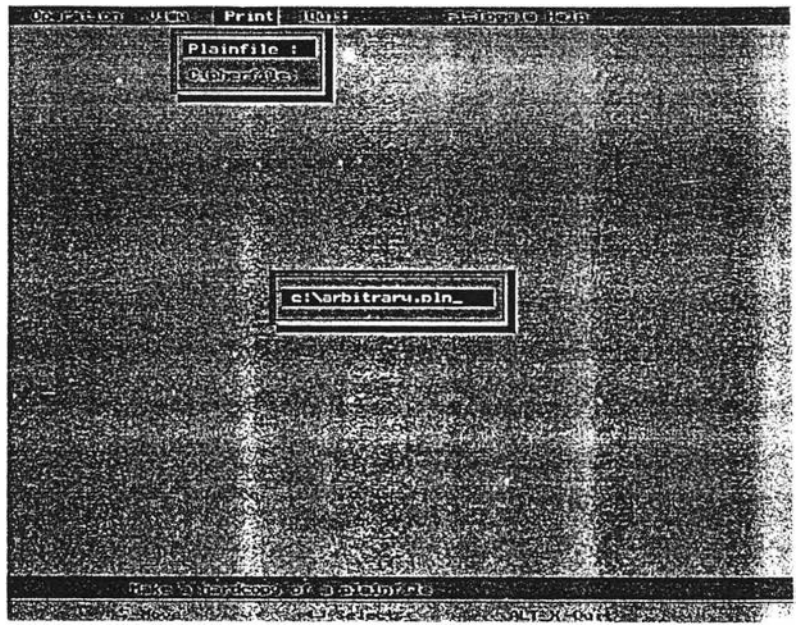


รูป 5.17 โปรแกรม Processing Time

สำหรับการกลับเข้าสู่โปรแกรมหลักของทั้งสองโปรแกรมทำได้โดยกดปุ่ม ESC

3.2 โปรแกรม Print

โปรแกรม Print ใช้สำหรับพิมพ์ไฟล์ที่ต้องการออกทางเครื่องพิมพ์โดยสามารถเลือกได้ว่าจะพิมพ์ Plain File ชื่อ Cipher File โดยการเลือกเมนูที่ปรากฏและใส่ชื่อไฟล์ที่ตต้องการพิมพ์ ดังรูป 5.18



รูป 5.18 เมนูใน Print

รูปแบบของไฟล์ที่พิมพ์ออกมาจะมีรายละเอียดทั้งชื่อไฟล์ วันที่ เวลา และบอกหมายเลขหน้าด้วย ดังรูป5.19

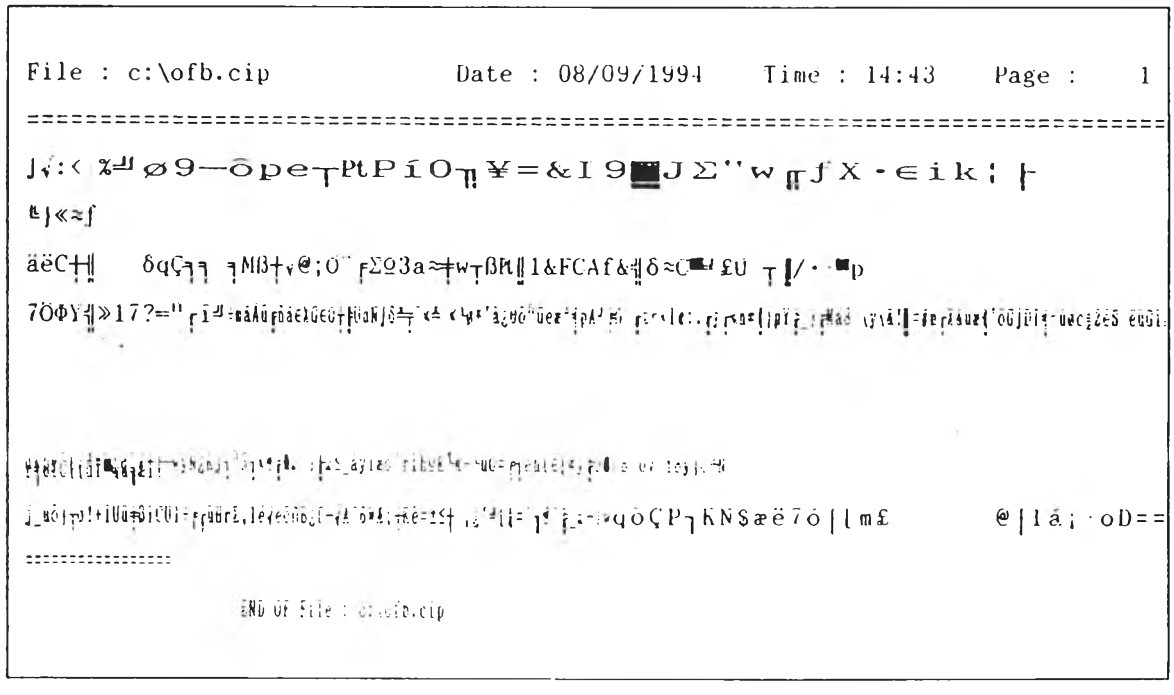
```
File : c:\autoexec.bat        Date : 08/09/1994    Time : 14:42    Page :    1
================================================================================
echo off
prompt $p$g
path = c:\tc;c:\dos5;c:\util\norton;c:\windows
c:\mouse\mouse

================================================================================
                    END OF File : c:\autoexec.bat
```

รูป 5.19 รูปแบบของการพิมพ์ในโปรแกรม Print

อย่างไรก็ตามในกรณีที่เราเลือกเมนูของการพิมพ์ไม่ตรงกับชนิดของไฟล์ที่เราต้องการจะพิมพ์ จะทำให้เครื่องพิมพ์อาจจะพิมพ์ข้อความที่ผิดพลาดได้ เนื่องจากวิธีการส่งค่าไปให้เครื่องพิมพ์ของแต่ละเมนูนั้นไม่เหมือนกัน กล่าวคือสำหรับ Plain File เราจะส่งค่าที่อ่านได้จากไฟล์ให้เครื่องพิมพ์โดยตรงแต่ใน Cipher File ค่าที่เราอ่านได้จากไฟล์บางค่าจะเป็นค่าเดียวกับที่ใช้ในการควบคุมการทำงานของเครื่องพิมพ์ ดังนั้นเมื่อได้รับค่าที่ใช้ควบคุมเครื่องพิมพ์ เราจะไม่ส่งค่านี้นออกไป แต่จะแทนด้วย "." เพื่อให้เราสามารถควบคุมการพิมพ์ของเครื่องพิมพ์ให้ออกมาเป็นรูปแบบดังรูป 5.19 ได้

ตัวอย่างการพิมพ์ของการเลือกเมนูไม่ตรงกับชนิดของไฟล์ที่จะพิมพ์ แสดงดังรูป 5.20



รูป 5.20 ตัวอย่างการพิมพ์ผิดพลาด