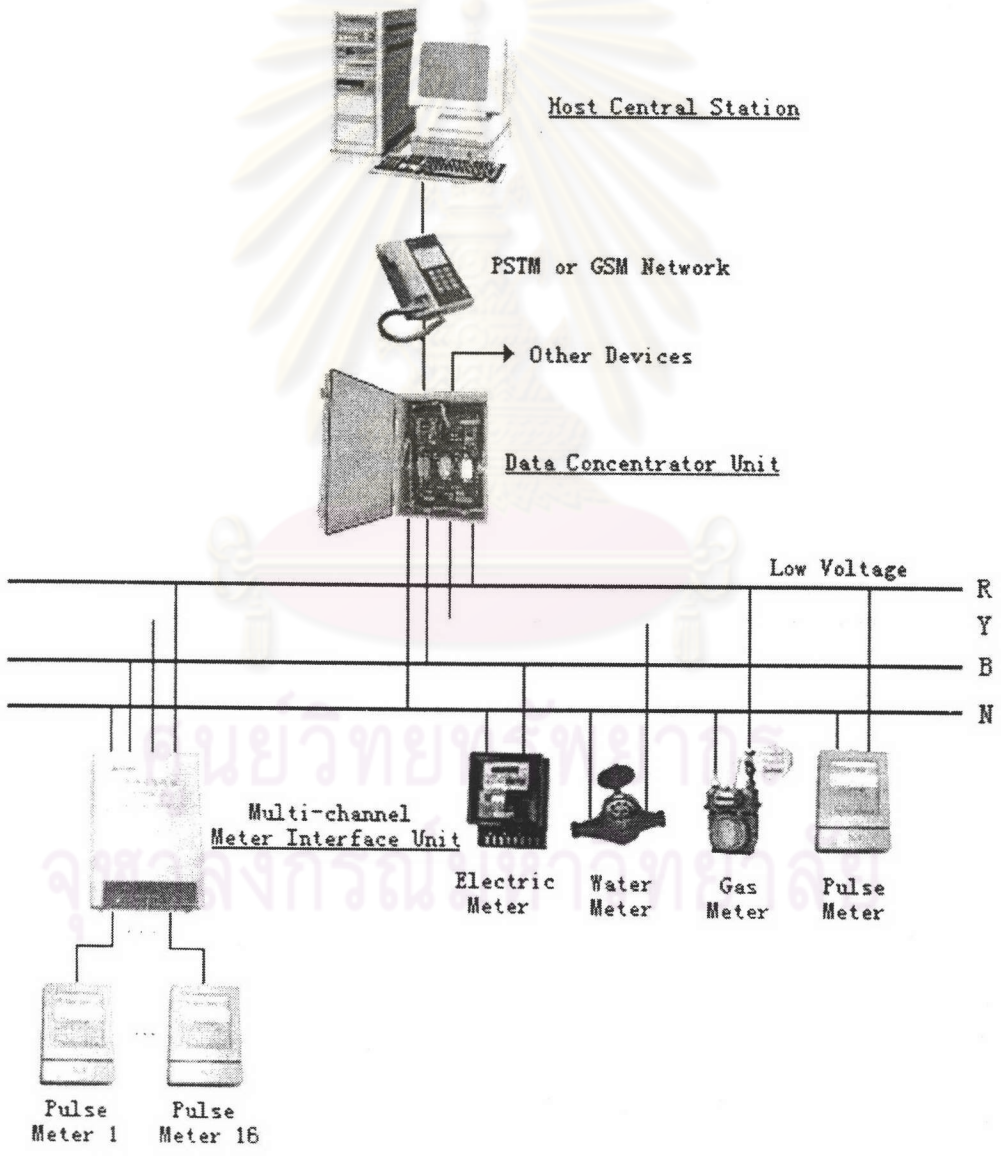
บทที่ 2

ความรู้พื้นฐานและหลักการที่เกี่ยวข้อง

2.1 องค์ประกอบของระบบ AMR

องค์ประกอบของระบบ AMR โดยทั่วไปมีประกอบด้วยส่วนต่างๆ ดังแสดงในรูปที่ 2.1 โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 2.1 ตัวอย่างแสดงส่วนประกอบของระบบ AMR

1) มิเตอร์

มีหน้าที่วัดพลังงานไฟฟ้าที่ถูกใช้ไปโดยผู้ใช้ไฟฟ้าแต่ละราย ถ้าเป็นมิเตอร์รุ่นเก่าต้องมีการใช้งานร่วมกับส่วนจำเพาะเชื่อมต่อกับมิเตอร์ (Meter Interface Module) ด้วยเพื่อให้สามารถใช้งานในระบบ AMR ได้ ถ้าเป็นมิเตอร์รุ่นใหม่จะมีส่วนจำเพาะเชื่อมต่อกับมิเตอร์อยู่แล้วในตัว

2) ส่วนจำเพาะเชื่อมต่อกับมิเตอร์ (Meter Interface Module)

ทำหน้าที่รับข้อมูลจากมิเตอร์ และแปลงข้อมูลเหล่านี้ให้อยู่ในรูปแบบมาตรฐานก่อนที่จะถูกส่งต่อไปยังส่วนกลางโดยเครื่องอ่าน โดยทั่วไปเป็นการสื่อสารแบบ 2 ทาง

3) ตัวรวบรวมข้อมูล (Data Concentrator Unit)

มีหน้าที่รวบรวมข้อมูลที่มาจากส่วนจำเพาะเชื่อมต่อกับมิเตอร์ บนมิเตอร์หลายตัวจากแต่ละพื้นที่ และส่งข้อมูลต่อไปยังส่วนกลางผ่านทางเครือข่ายการสื่อสาร

4) เครือข่ายการสื่อสาร (Communication Network)

เป็นเส้นทางสำหรับส่งข้อมูลจากตัวรวบรวมข้อมูลไปยังส่วนกลางซึ่งมีอยู่หลายวิธีด้วยกัน เช่น การใช้เครือข่ายโทรศัพท์ (Public Switch Telephone Network) การใช้สายส่งกำลังเป็นพาหะ (Power Line Carrier) การใช้คลื่นวิทยุ และการใช้อุปกรณ์แบบพกพาได้เป็นต้น

5) อุปกรณ์ส่วนกลาง (Central Office System Equipment)

ได้แก่คอมพิวเตอร์ควบคุมการทำงานของระบบ AMR และ Utility Terminal Unit (UTU) ซึ่งทำหน้าที่เชื่อมต่อ และจัดการข้อมูลที่ถูกส่งมาจากตัวรวบรวมข้อมูลในพื้นที่ต่างๆ ให้กับโปรแกรมควบคุม

6) โปรแกรมควบคุม

ทำหน้าที่ควบคุมการทำงานของระบบ AMR รวมทั้งจัดการกับข้อมูลที่อ่านมาจากมิเตอร์ เช่นการเก็บข้อมูลลงในฐานข้อมูล และการคิดค่าไฟฟ้า

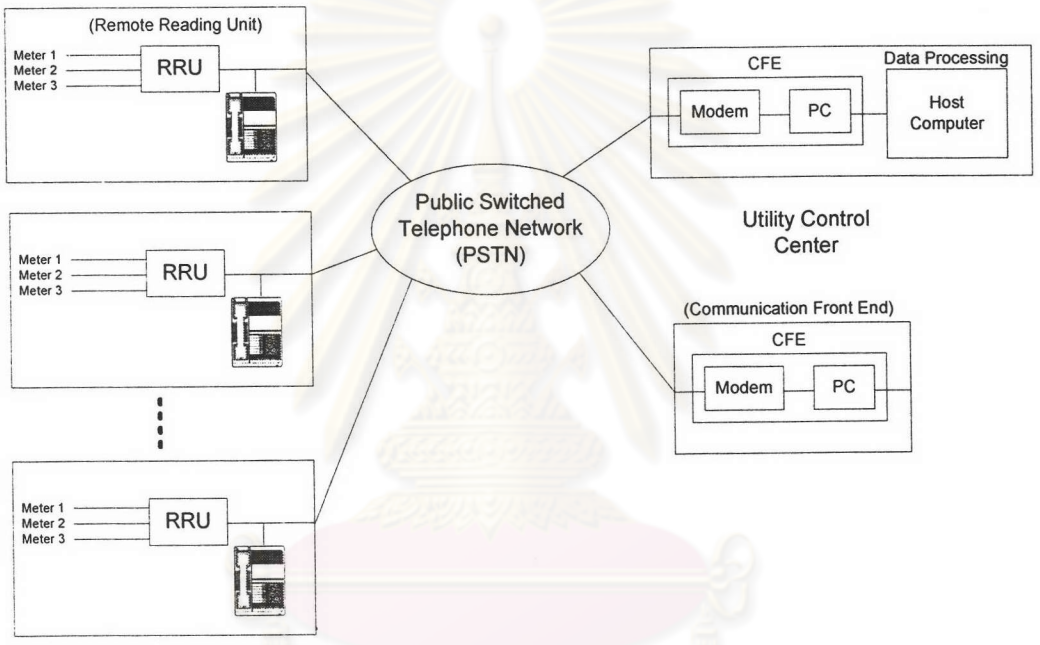
2.2 ระบบ AMR แบบต่างๆ

ตามที่ได้ศึกษาจากเอกสารอ้างอิงต่างๆ [1] และ [2] ทำให้ทราบถึงรายละเอียดของการพัฒนาระบบ AMR โดยการใช้เทคนิคและตัวกลางในการสื่อสารแบบต่างๆ ซึ่งมีทั้งข้อดีข้อเสียต่างกันไป การพิจารณาความเหมาะสมและการเลือกใช้เทคนิคต่างๆ สำหรับระบบ AMR นั้นขึ้นอยู่กับสภาพแวดล้อมของแต่ละพื้นที่เป็นสำคัญ จึงไม่มีรูปแบบ หรือเทคนิคของระบบ AMR แบบใดที่เหมาะสมกับทุกพื้นที่ รายละเอียดของระบบ AMR แบบต่างๆ ตามที่ได้ศึกษามามีดังนี้

1) การใช้ระบบโทรศัพท์

เป็นการใช้ประโยชน์จากเครือข่ายโทรศัพท์ [3] ซึ่งผู้ใช้ไฟฟ้าตามบ้านส่วนใหญ่มีหมายเลขโทรศัพท์เป็นของตนเองอยู่แล้ว ข้อมูลจากมิเตอร์จะถูกส่งไปยังศูนย์ควบคุมผ่านทางสายโทรศัพท์โดยใช้โมเด็ม ดังแสดงในรูปที่ 2.2

ข้อดีของระบบ AMR แบบนี้คือ ประหยัดค่าใช้จ่ายเนื่องจากใช้ระบบเดิมที่มีอยู่แล้ว และสามารถส่งได้ในระยะไกลจึงไม่จำเป็นต้องใช้ตัวรวบรวมข้อมูล สำหรับเก็บข้อมูลจากมิเตอร์ในบริเวณใกล้เคียงก่อน



รูปที่ 2.2 ระบบ AMR ที่ใช้เครือข่ายโทรศัพท์

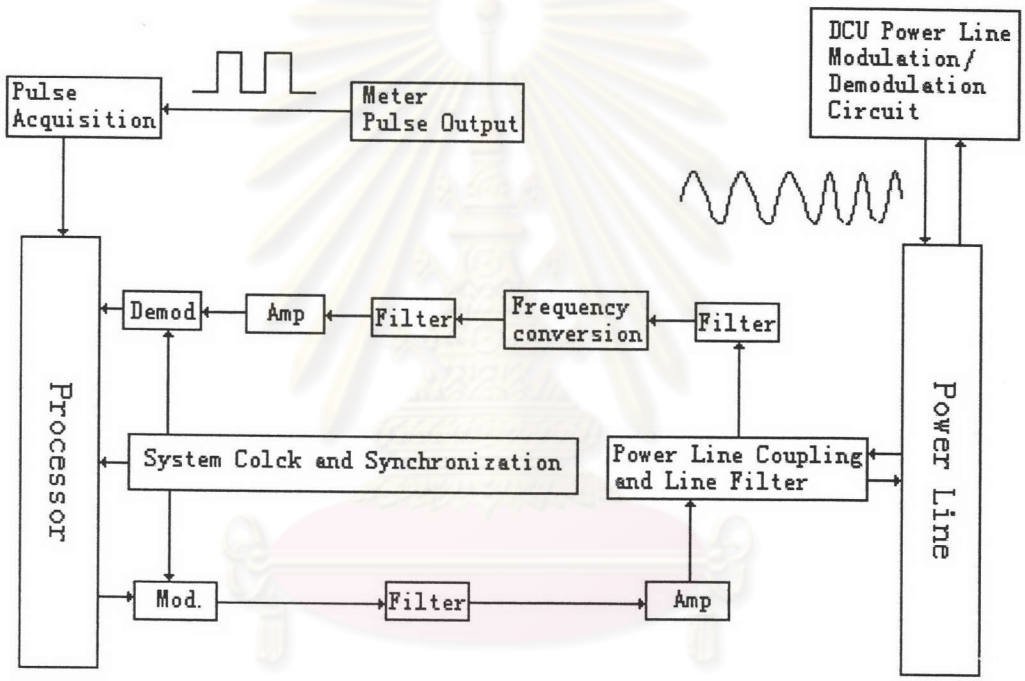
ระบบ AMR ที่ใช้สายโทรศัพท์สามารถแบ่งออกได้เป็น 3 ประเภทคือ

- *มิเตอร์เป็นฝ่ายโทรออก (Inbound communication)* มิเตอร์เป็นฝ่ายโทรไปหาศูนย์ตามวันเวลาที่ถูกตั้งเอาไว้ซึ่งมีข้อดีคือ ไม่ต้องมีอุปกรณ์เพิ่มเติมเหมือนแบบที่ใช้ศูนย์ควบคุมเป็นฝ่ายโทรออก แต่มีข้อเสียคือศูนย์ไม่สามารถติดต่อกับมิเตอร์ได้โดยทันที
- *ศูนย์ควบคุมเป็นฝ่ายโทรออก (Outbound communication)* มีข้อดีคือฝั่งศูนย์ควบคุมสามารถติดต่อกับตัวมิเตอร์ได้ตลอดเวลา แต่มีข้อเสียคือระบบโทรศัพท์ต้องติดตั้งอุปกรณ์บางอย่างเพิ่มเข้าไปเพื่ออนุญาตให้ศูนย์ควบคุมสามารถหมุนหมายเลขโทรศัพท์เพื่อติดต่อกับมิเตอร์ได้โดยปราศจากเสียงเรียกเข้าดังขึ้นที่เครื่องโทรศัพท์ที่ต่อพ่วงอยู่กับมิเตอร์

- *แบบที่สามารถโทรออกได้ทั้งสองทาง (Bi-directional communication)* เป็นการนำคุณสมบัติของทั้ง 2 แบบข้างต้นมารวมกัน

2) การใช้สายส่งกำลัง (Power Line Carrier: PLC)

เป็นอีกวิธีหนึ่งที่ใช้ประโยชน์จากสายส่งกำลังที่มีอยู่แล้วมาใช้เป็นตัวนำในการสื่อสารของระบบ AMR โดยใช้เทคนิคการส่งสัญญาณพาหะ ที่เป็นไฟสลับความถี่สูง (โดยทั่วไปอยู่ในย่าน 9 kHz ไปจนถึง 20 MHz) เข้าไปในสายส่งกำลังและการมอดูเลตสัญญาณพาหะนี้เข้ากับข้อมูลที่มาจากส่วนจำเพาะเชื่อมต่อกับมิเตอร์หรือจากตัวรวบรวมข้อมูลโดยใช้โมเด็มชนิดส่งข้อมูลผ่านสายส่งกำลัง (Power Line Modem) ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างของระบบ AMR ที่ใช้สายส่งกำลังเป็นตัวกลางในการรับส่งข้อมูล

อย่างไรก็ตามวิธีนี้มีปัญหาอยู่มาก [4] ยกตัวอย่างเช่น ไม่สามารถส่งได้ในระยะไกล เพราะสายส่งสามารถเป็นตัวลดทอนสัญญาณพาหะที่มีความถี่สูงได้ สัญญาณไม่สามารถถูกส่งผ่านข้ามหม้อแปลงแรงดันสูงของการไฟฟ้าได้ นอกจากนี้ยังทำให้เกิดการแทรกสอดไปรบกวนกับคลื่นวิทยุได้ และถูกรบกวนจากสัญญาณรบกวนจากแหล่งต่างๆ ได้ง่าย เป็นต้น

3) การใช้คลื่นวิทยุ

เป็นการใช้คลื่นวิทยุเป็นตัวกลางในการสื่อสารกับมิเตอร์ [1] และ [2] แต่ก็อาจทำให้เกิดปัญหาของการแทรกสอดหรือรบกวนขึ้นได้หากในพื้นที่ไม่มีการจัดสรรความถี่ไว้อย่างเหมาะสม ระบบ AMR โดยใช้คลื่นวิทยุถูกแบ่งเป็น 2 ประเภทหลัก ได้แก่

- *แบบอยู่กับที่ (Fixed Radio)* ตัวรับส่งคลื่นวิทยุ (Radio transceiver) ที่ตัวรวบรวมข้อมูลซึ่งมักถูกติดตั้งอยู่ตามเสาหรือตามตึกที่อยู่ใกล้กับมิเตอร์จะคอยทำหน้าที่รับสัญญาณที่ถูกส่งมาจากส่วนจำเพาะเชื่อมต่อกับมิเตอร์ และส่งสัญญาณนี้ต่อไปยังศูนย์ควบคุมส่วนกลางต่อไปโดยอาจใช้เครือข่ายการสื่อสารแบบอื่นร่วมด้วยเช่น เครือข่ายโทรศัพท์ เป็นต้น
- *แบบเคลื่อนที่ (Mobile Radio)* เป็นการใช้วิธีที่ให้ตัวอ่านมิเตอร์ซึ่งมีตัวรับส่งคลื่นวิทยุติดอยู่เคลื่อนที่ผ่านเข้าไปใกล้ส่วนจำเพาะเชื่อมต่อกับมิเตอร์ในระยะปฏิบัติการ โดยอุปกรณ์อาจมีลักษณะที่สามารถพกพาได้ (Portable device) หรือถูกติดตั้งอยู่บนรถเพื่อความสะดวกในการเคลื่อนที่ไปยังพื้นที่ต่างๆ

4) อุปกรณ์มือถือโดยใช้อินฟราเรด

เป็นการใช้อุปกรณ์มือถือที่มีช่องทางแสงอินฟราเรดสำหรับติดต่อสื่อสารกับตัวมิเตอร์ในระยะใกล้ในระยะไม่กี่เมตร ข้อดีของการประยุกต์ใช้อินฟราเรด กับระบบ AMR คือ อุปกรณ์แบบมือถือมีต้นทุนในการผลิตต่ำเมื่อเทียบกับการติดตั้งระบบ AMR แบบอื่น ใช้กำลังงานต่ำ และมีการทำงานที่ไม่ซับซ้อน [5] เมื่อพนักงานเก็บข้อมูลจากมิเตอร์ทุกตัวในพื้นที่จนครบแล้ว จึงนำอุปกรณ์ดังกล่าวเชื่อมต่อกับคอมพิวเตอร์เพื่อดึงข้อมูลที่อ่านมาได้เก็บลงในฐานข้อมูล

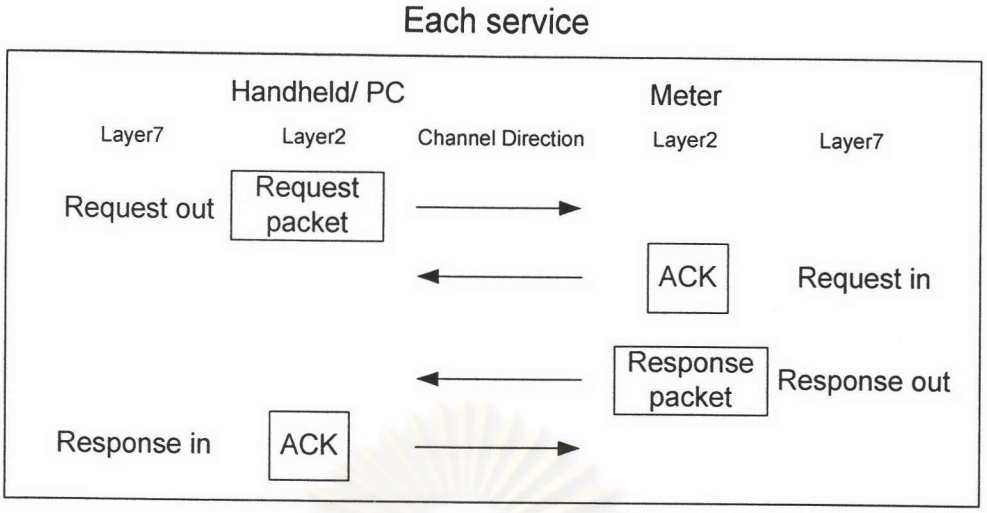
จากข้อดีดังกล่าวของอุปกรณ์มือถือชนิดอินฟราเรด ในวิทยานิพนธ์นี้จึงได้พัฒนาต้นแบบของระบบ AMR ด้วยเครื่องอ่านที่สามารถพกพาได้ซึ่งใช้แสงอินฟราเรดเป็นช่องทางการสื่อสารกับมิเตอร์ขึ้น ดังจะกล่าวถึงรายละเอียดต่อไปในบทที่ 3 และ 4

2.3 โพรโตคอลสำหรับติดต่อสื่อสารกับมิเตอร์

การสื่อสารกับมิเตอร์ด้วยอุปกรณ์มือถือที่พัฒนาขึ้นยึดรูปแบบของโพรโตคอลตามมาตรฐาน ANSI C12.18-1996 [11] เป็นหลัก ซึ่งกล่าวถึงรายละเอียดโดยการแบ่งเป็นลำดับตามรูปแบบของการเชื่อมต่อระหว่างระบบเปิด (Open system Interconnection หรือ OSI) ด้วยกันทั้งหมด 3 ระดับคือ ระดับ Physical ระดับ Data Link และระดับ Application

2.3.1 รายละเอียดโดยสังเขปของระดับ Application

ในหัวนี้จะกล่าวถึงกระบวนการ (Service) ต่างๆ ที่ใช้ในการสื่อสารระหว่างตัวมิเตอร์กับอุปกรณ์ที่จะมาอ่าน โดยในแต่ละกระบวนการจะประกอบไปด้วย 2 ขั้นตอนย่อยคือ ขั้นตอนร้องขอ (Request) และขั้นตอนตอบสนอง (Response) ดังแสดงในรูปที่ 2.4 กระบวนการต่างๆ ได้แก่



รูปที่ 2.4 แสดงขั้นตอนย่อยในแต่ละกระบวนการของระดับ Application

1) กระบวนการระบุ (Identification service)

เป็นกระบวนการที่ต้องทำเป็นลำดับแรกหลังจากที่มีการเชื่อมต่อช่องทางการสื่อสารขึ้น กระบวนการนี้ใช้สำหรับร้องขอดูรายละเอียดของรุ่น (Version) และรุ่นของการปรับปรุง (Revision) ของโพรโตคอลบนตัวมิเตอร์ก่อนที่จะเริ่มกระบวนการต่อไป

2) กระบวนการเจรจา (Negotiate service)

เป็นกระบวนการที่ใช้สำหรับตั้งหรือเปลี่ยนค่าพารามิเตอร์ที่ใช้ในการสื่อสารกับมิเตอร์ไปจากค่าเริ่มต้น (Default value) ค่าพารามิเตอร์เหล่านี้ได้แก่ อัตราเร็วในการรับส่งข้อมูล ขนาดและจำนวนของกลุ่มข้อมูล (packet) มากสุดที่รองรับได้

3) กระบวนการลงบันทึกเปิด (Logon service)

ใช้สำหรับร้องขอการเริ่มต้นเข้าสู่ช่วงเวลาสื่อสาร (session state) โดยการส่งหมายเลขของผู้ใช้ และชื่อของผู้ใช้ที่ทำการลงบันทึกเปิดไปในขั้นตอนร้องขอด้วย ข้อมูลของผู้ใช้เหล่านี้จะถูกเก็บอยู่ในส่วนลงบันทึกเหตุการณ์ (History logs) บนตัวมิเตอร์ด้วย

4) กระบวนการรักษาความปลอดภัย (Security service)

ถูกใช้สำหรับการอนุญาตหรือกำหนดสิทธิในการเข้าถึงข้อมูลให้กับแต่ละตารางข้อมูลบนตัวมิเตอร์โดยการส่งรหัสผ่านไปในขั้นตอนร้องขอ

5) กระบวนการอ่าน (Read service)

เป็นกระบวนการที่ใช้อ่านข้อมูลที่อยู่ในตารางข้อมูลแต่ละชุดบนตัวมิเตอร์ออกมา ในการอ่านข้อมูลจะต้องทำการระบุหมายเลขของตารางข้อมูลไปด้วย นอกจากนี้ยังสามารถเลือกได้ว่าต้องการอ่านข้อมูลทั้งหมดในตารางข้อมูลหรืออ่านเป็นบางส่วนตามรูปแบบของขั้นตอนร้องขอของกระบวนการอ่าน

6) กระบวนการเขียน (Write service)

ใช้สำหรับเขียนข้อมูลลงไปในตารางข้อมูลที่อยู่บนตัวมิเตอร์ตามหมายเลขตารางข้อมูลที่อยู่ในขั้นตอนร้องขอ เป็นกระบวนการที่ทำหน้าที่ตรงข้ามกับกระบวนการอ่านข้อมูล

7) กระบวนการลงบันทึกปิด (Logoff service)

ตรงข้ามกับกระบวนการลงบันทึกเปิดคือ ใช้ร้องขอการออกจากช่วงเวลาการสื่อสารซึ่งถูกสร้างขึ้นโดยกระบวนการลงบันทึกเปิด โดยค่าพารามิเตอร์ที่ถูกเปลี่ยนแปลงโดยกระบวนการเจรจาจะยังคงเดิมอยู่

8) กระบวนการทำให้สิ้นสุด (Terminate service)

ใช้สั่งยกเลิกช่องทางการสื่อสารที่ถูกสร้างขึ้นโดยทันที และค่าพารามิเตอร์ที่ใช้ในการสื่อสารซึ่งถูกเปลี่ยนแปลงโดยกระบวนการเจรจาจะถูกเซตเป็นค่าเริ่มต้นใหม่หมด

ดังนั้นในการสื่อสารเพื่อติดต่อกับตัวมิเตอร์โดยทั่วไปนั้นจะประกอบด้วยกระบวนการต่างๆ ตามลำดับต่อไปนี้

กระบวนการระบุ -> กระบวนการเจรจา -> กระบวนการลงบันทึกเปิด -> กระบวนการรักษาความปลอดภัย -> กระบวนการอ่านหรือเขียน -> กระบวนการทำให้สิ้นสุด

2.3.2 <u>รายละเอียดของกลุ่มข้อมูลในระดับ Data Link</u>

รูปแบบของกลุ่มข้อมูลตามมาตรฐาน ANSI C12.18-1996 ที่ใช้ติดต่อสื่อสารระหว่างอุปกรณ์แบบพกพาที่พัฒนาขึ้นกับมิเตอร์ประกอบไปด้วยเขตข้อมูล (field) ต่างๆ ดังแสดงในรูปที่ 2.5 โดยแต่ละเขตข้อมูลมีรายละเอียดดังนี้

Stp	Reserved	Ctrl	Seq_nbr	Length	Data	CRC
1 byte	1 byte	1 byte	1 byte	2 bytes	x bytes	2 bytes

รูปที่ 2.5 รูปแบบของกลุ่มข้อมูลตามมาตรฐาน ANSI C12.18-1996

- <Stp> มีค่าเป็น EE_H เพื่อบอกให้รู้ว่าเป็นไบต์แรกของกลุ่มข้อมูล (Start of packet)
- <*Ctrl*> *บิตที่ 7* ใช้บอกชนิดของกลุ่มข้อมูลว่าเป็นแบบกลุ่มข้อมูลเดี่ยว (Single packet) หรือเป็นส่วนหนึ่งของกลุ่มข้อมูลแบบหลายกลุ่ม (Multiple packet)

  บิตที่ 6 หากมีค่าเป็น "1" แสดงว่าเป็นกลุ่มข้อมูลกลุ่มแรกของกลุ่มข้อมูลแบบหลายกลุ่ม

  บิตที่ 5 ถูกใช้เป็นบิตสลับ (toggle) เพื่อป้องกันการรับกลุ่มข้อมูลซ้ำ บิตนี้จะถูกสลับไปมาทุกครั้งที่มีการส่งกลุ่มข้อมูลใหม่ออกมา

- *<Seq_nbr>* แสดงถึงลำดับของกลุ่มข้อมูลในกรณีของการรับสั่งข้อมูลแบบหลายกลุ่มข้อมูล โดยกลุ่มข้อมูลแรกจะมีหมายเลขลำดับเท่ากับ (จำนวนกลุ่มข้อมูลทั้งหมด – 1) ส่วนกลุ่มข้อมูลสุดท้ายหรือกลุ่มข้อมูลเดี่ยวจะมีค่าเป็น 00_H
- *<Length>* ใช้บอกความยาวของเขตข้อมูล <Data> โดยมีหน่วยเป็นไบต์
- *<Data>* เป็นเขตของข้อมูลที่แท้จริงที่ถูกนำไปใช้ในกระบวนการต่างๆ ของระดับ Application
- *<CRC>* เป็นค่าการตรวจสอบด้วยส่วนซ้ำซ้อนแบบวน (Cyclic Redundancy Check) ของกลุ่มข้อมูลกลุ่มข้อมูลนั้นเพื่อใช้ตรวจสอบความถูกต้องของกลุ่มข้อมูล

2.3.3 <u>การคำนวณหาค่าซีอาร์ซี</u>

ค่าซีอาร์ซี คือเศษเหลือจากการหารค่าฟังก์ชัน Polynomial ต่อไปนี้ [6]

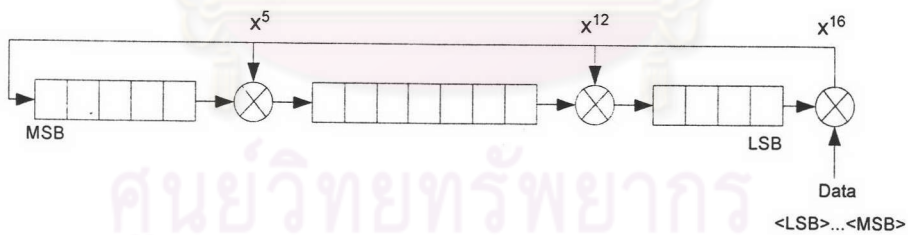
$$\text{ค่าซีอาร์ซี} = \text{เศษของการหารค่า} \frac{x^{n-p}[P(x)]}{G(x)}$$

โดย P(x) คือ Polynomial function ของข้อมูลที่จะถูกนำมาคำนวณ

G(x) คือ Generator Polynomial function ซึ่งเป็นค่าคงที่ค่าหนึ่งซึ่งมีค่าแตกต่างกันไปตามมาตรฐานเช่น มาตรฐานของ CRC-CCITT ซึ่งเป็นแบบที่ใช้คำนวณค่าซีอาร์ซีให้กับแต่ละกลุ่มข้อมูลตามมาตรฐาน C12.18-1996 ใช้ค่า $G(x) = x^{16} + x^{12} + x^5 + 1$ ในการคำนวณ

n คือจำนวนบิตของข้อมูลทั้งหมดในเฟรมที่รวมจำนวนบิตของค่าซีอาร์ซีเข้าไปด้วย

p คือจำนวนบิตของข้อมูล P(x) เพียงอย่างเดียว



รูปที่ 2.6 หลักการคำนวณค่าซีอาร์ซีแบบ CRC-CCITT โดยใช้รีจิสเตอร์แบบเลื่อน

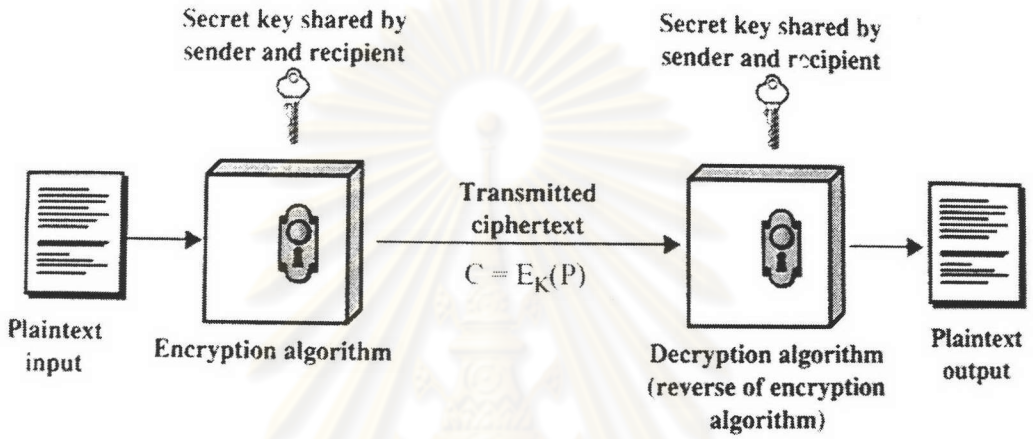
เพื่อให้การคำนวณค่าซีอาร์ซี สามารถนำไปประยุกต์ใช้โดยการออกแบบฮาร์ดแวร์หรือการเขียนโปรแกรมได้โดยง่าย การคำนวณค่าซีอาร์ซีจึงสามารถคำนวณได้โดยใช้โครงสร้างของรีจิสเตอร์แบบเลื่อนทำงานร่วมกับตัวออร์เฉพาะ (Exclusive-OR) ดังแสดงในรูปที่ 2.6 ซึ่งแสดงการหาค่าซีอาร์ซีแบบ CRC-CCITT โดยตำแหน่งของออร์เฉพาะในวงจรรีจิสเตอร์แบบเลื่อนจะสัมพันธ์กับ G(x) ที่ใช้

ตามที่มาตรฐาน ISO/IEC 3309:1993(E) ระบุไว้ ค่าเริ่มต้นในวงจรรีจิสเตอร์แบบเลื่อนก่อนการคำนวณจะถูกเซตเป็น “1” ทุกบิต จากนั้นข้อมูลจะถูกป้อนเข้าไปทีละบิตโดยเริ่มจากบิต

ต่ำสุด (Least Significant Bit) ก่อนจนครบ ค่าสุดท้ายที่ปรากฏอยู่ในวงจรรีจิสเตอร์แบบเลื่อนจะถูกนำไปกลับ (Invert) บิตอีกครั้ง ค่าสุดท้ายที่ได้คือค่าซีอาร์ซีของแต่ละกลุ่มข้อมูล

2.4 การเข้าและถอดรหัสลับแบบสมมาตร (Symmetric Cipher Model)

เป็นการใช้กุญแจตัวเดียวกันสำหรับทั้งในการเข้าและถอดรหัสข้อมูล กระบวนการเข้าและถอดรหัสลับแบบสมมาตรสามารถอธิบายได้ด้วยรูปที่ 2.7 โดยมีองค์ประกอบและรายละเอียดดังต่อไปนี้ [14] และ [15]



รูปที่ 2.7 แสดงกระบวนการเข้ารหัสแบบสมมาตร

1) *ข้อมูลที่จะนำมาเข้ารหัส (Plaintext)* เป็นข้อมูลที่ถูกนำเข้าสู่ขั้นตอนวิธีเข้ารหัส
2) *ขั้นตอนวิธีเข้ารหัส (Encryption algorithm)* ทำหน้าที่สลับหรือแปลงข้อมูลนำเข้าให้เป็นข้อมูลใหม่ $C = E_K(P)$
3) *กุญแจ (Secret key)* เป็นข้อมูลนำเข้าให้กับทั้งการเข้าและถอดรหัส
4) *ข้อมูลที่ถูกเข้ารหัสแล้ว (Ciphertext)* เป็นข้อมูลนำออกที่ผ่านขั้นตอนวิธีเข้ารหัสมาแล้ว
5) *ขั้นตอนวิธีถอดรหัส (Decryption algorithm)* มีหน้าที่สลับหรือแปลงข้อมูลกลับออกมาให้เหมือนเดิมก่อนทำการเข้ารหัส $P = D_K(C) = D_K(E_K(P))$

2.4.1 รหัสลับแบบแทนที่ (Substitution Cipher)

เป็นกลวิธีพื้นฐานของการเข้ารหัสที่นิยมใช้กันในช่วงเริ่มต้นของการคิดค้นกระบวนการเข้ารหัสโดยการแทนที่อักษรแต่ละตัวของข้อมูลด้วยอักษรอีกตัวหนึ่ง ถูกแบ่งเป็น 2 แบบย่อยคือ

1) การแทนตัวอักษรแบบหนึ่งต่อหนึ่ง (Monoalphabetic) เป็นการแทนที่ตัวอักษรตัวหนึ่งด้วยตัวอักษรอีกค่าหนึ่งเช่น แทนอักษร a ด้วย Q แทนอักษร b ด้วย W เป็นต้น ดังตัวอย่างของรหัสลับแบบ Caesar ในรูปที่ 2.8

```
plaintext:    a b c d e f g h i j k l m n o p q r s t u v w x y z
ciphertext:   Q W E R T Y U I O P A S D F G H J K L Z X C V B N M
```

รูปที่ 2.8 แสดงการเข้ารหัสแบบ Caesar

2) การแทนตัวอักษรแบบหลากหลาย (Polyalphabetic) [13] เป็นการใช้การแทนตัวอักษรแบบหนึ่งต่อหนึ่งหลายชุดรวมกันเพื่อเข้ารหัสให้กับข้อมูล ตัวอย่างของการเข้ารหัสแบบนี้คือรหัสลับแบบ Vigenere ดังรูปที่ 2.9

```
key:          deceptivewearediscoveredsav
plaintext·    wearediscoveredsaveyourself
ciphertext:   ZICVTWQNGKZEIIGASXSTSLVVWLA
```

Encryption: $C_i = M_i + K_i \pmod{26}$

Decryption: $M_i = C_i - K_i \pmod{26}$

รูปที่ 2.9 แสดงการเข้ารหัสแบบ Vigenere

2.4.2 <u>รหัสลับแบบย้ายข้าง (Transposition Cipher)</u>

กลวิธีดังกล่าวสามารถแสดงได้โดยรูปที่ 2.10 โดยข้อความที่ใช้เป็นกุญแจต้องไม่มีตัวอักษรซ้ำกันดังตัวอย่างซึ่งใช้คำว่า MEGABUCK หลังจากจัดเรียงข้อมูลให้เป็นไปตามรูปแบบแล้วจึงเริ่มดึงข้อมูลจากแต่ละสดมภ์ออกมาจัดเรียงใหม่โดยเรียงลำดับตามตัวอักษรของข้อความที่เป็นกุญแจ

```
M E G A B U C K
7 4 5 1 2 8 3 6
p l e a s e t r
a n s f e r o n
e m i l l i o n
d o l l a r s t
o m y s w i s s
b a n k a c c o
u n t s i x t w
o t w o a b c d
```

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

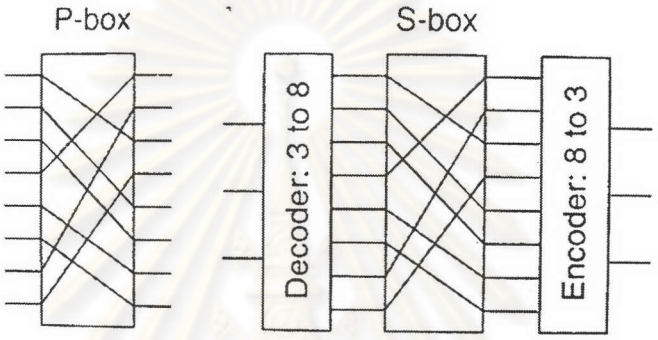
AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

รูปที่ 2.10 ตัวอย่างการเข้ารหัสแบบย้ายข้าง

อย่างไรก็ตามกลวิธีในหัวข้อ 2.4.1 และ 2.4.2 ไม่เหมาะสมที่จะนำมาใช้ในปัจจุบัน เนื่องจากความสามารถของเครื่องคอมพิวเตอร์ประสิทธิภาพสูงและกลวิธีการวิเคราะห์รหัสลับ (Cryptanalysis) ที่หลากหลายทำให้เสี่ยงต่อการถูกถอดรหัสได้โดยง่าย

2.4.3 พื้นฐานของการเข้ารหัสลับแบบบล็อก (Block Cipher)

เป็นกลวิธีการเข้ารหัสที่ถูกใช้อย่างแพร่หลายในปัจจุบัน [14] ก่อนเริ่มต้นการเข้ารหัสด้วยวิธีนี้ ข้อมูลที่จะเข้ารหัสจะต้องถูกแบ่งออกเป็นบล็อกๆ โดยมีขนาดบล็อกละ n บิตเสียก่อน เนื่องจากกระบวนการเข้ารหัสต้องทำกับข้อมูลทีละ n บิตเช่น 64 หรือ 128 บิต ซึ่งแตกต่างจาก 2 แบบแรกที่กล่าวมาที่มีการดำเนินการแบบทีละ 1 ตัวอักษรหรือ 8 บิตเท่านั้น ดังนั้นกลวิธีการเข้ารหัสแบบบล็อกจึงมีความซับซ้อนกว่ามาก โครงสร้างพื้นฐานของการเข้ารหัสแบบบล็อกประกอบด้วย 2 ส่วนดังรูปที่ 2.11

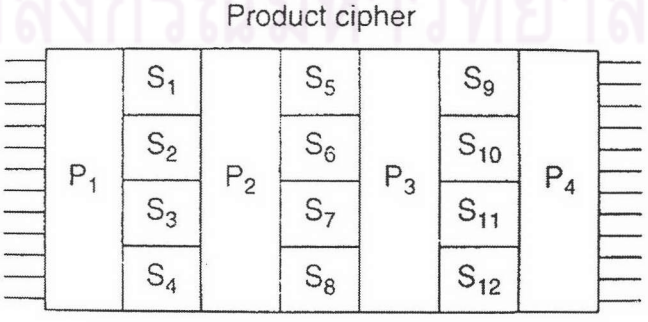


รูปที่ 2.11 ตัวอย่างโครงสร้างของส่วนสับเปลี่ยน และส่วนแทนค่า

1) *กล่องสับเปลี่ยน (Permutation box หรือ P-box)* เป็นการนำหลักการเข้ารหัสแบบย้ายข้างมาใช้คือ นำข้อมูลขาเข้ามาทำการสลับสับเปลี่ยนตำแหน่งบิตข้อมูลเสียใหม่

2) *กล่องแทนค่า (Substitution box หรือ S-box)* ใช้หลักการเดียวกับการเข้ารหัสแบบแทนที่คือ ข้อมูลขาเข้าจะถูกแทนที่ด้วยข้อมูลค่าใหม่

โครงสร้างพื้นฐานดังกล่าวข้างต้นสามารถนำมาประกอบรวมกันหลายๆ ชั้นเพื่อให้มีความซับซ้อนมากขึ้นจนกลายเป็น Product cipher ดังตัวอย่างในรูปที่ 2.12 ซึ่งเป็นพื้นฐานของการเข้ารหัสแบบบล็อกชนิดต่างๆ



รูปที่ 2.12 ตัวอย่างของ Product cipher ที่เกิดจากการรวมกันของโครงสร้างพื้นฐาน

2.4.4 ขั้นตอนวิธีการเข้ารหัสลับแบบบล็อกชนิด Blowfish

Blowfish เป็นขั้นตอนวิธีการเข้ารหัสแบบบล็อกชนิดหนึ่งซึ่งถูกใช้ในงานวิจัยนี้ คุณสมบัติและจุดเด่นของการเข้ารหัสด้วยวิธีนี้ได้แก่ [12] และ [15]

- *บล็อกข้อมูลมีขนาด 64 บิต*
- *ความยาวของกุญแจมีขนาดได้ตั้งแต่ 32 บิตจนถึง 448 บิต* ซึ่งยาวกว่าการเข้ารหัสแบบ DES ที่มีได้ยาวสุดเพียง 56 บิต ทำให้มีความปลอดภัยมากกว่า
- *เร็วเมื่อเทียบกับการเข้ารหัสแบบบล็อกชนิดอื่น* ดังข้อมูลในตารางที่ 2.1
- *ใช้ทรัพยากรน้อย* ใช้หน่วยความจำไม่ถึง 5 กิโลไบต์สำหรับเก็บกุญแจย่อย (Subkey)
- *สะดวกในการประยุกต์ใช้* ทั้งโดยทางฮาร์ดแวร์และซอฟต์แวร์
- *ไม่มีการจดสิทธิบัตรหรือลิขสิทธิ์* จึงใช้กันอย่างแพร่หลาย

ตารางที่ 2.1 เปรียบเทียบความเร็วของการเข้ารหัสแบบบล็อกแบบต่างๆ

Speed Comparisons of Block Ciphers on a Pentium				
Algorithm	Clock cycles per round	# of rounds	# of clock cycles per byte encrypted	Notes
Blowfish	9	16	18	Free, unpatented
Khufu/Khafre	5	32	20	Patented by Xerox
RC5	12	16	23	Patented by RSA Data Security
DES	18	16	45	56-bit key
IDEA	50	8	50	patented by Ascom-Systec
Triple-DES	18	48	108	

ส่วนประกอบสำคัญของการเข้ารหัสแบบ Blowfish คือ กลุ่มของกุญแจย่อย (Subkeys) ซึ่งต้องถูกสร้างเอาไว้ก่อนที่จะถูกใช้ในการเข้าหรือถอดรหัส อันประกอบไปด้วย

1) *กลุ่มของกล่องสับเปลี่ยนขนาด 32 บิต จำนวน 18 ชุด* ได้แก่

$$P_1, P_2, ..., P_{18}$$

2) *กลุ่มของกล่องแทนค่าขนาด 32 บิต จำนวน 256*4 ชุด* ได้แก่

$$S_{1,0}, S_{1,1}, ..., S_{1,254}, S_{1,255}$$

$$S_{2,0}, S_{2,1}, ..., S_{2,254}, S_{2,255}$$

$$S_{3,0}, S_{3,1}, ..., S_{3,254}, S_{3,255}$$

$$S_{4,0}, S_{4,1}, ..., S_{4,254}, S_{4,255}$$

การสร้างกลุ่มของกุญแจย่อยมีขั้นตอนดังนี้

1. กำหนดค่าเริ่มต้นให้กับทั้งกลุ่มของกล่องสับเปลี่ยนและแทนค่า

2. ทำการออร์เฉพาะ P1 กับค่ากุญแจขนาด 32 บิตแรก ออร์เฉพาะ P2 กับค่ากุญแจขนาด 32 บิตถัดไป และทำซ้ำไปเรื่อยๆ จนถึง P18

3. ทำการเข้ารหัสข้อมูลที่มีค่าเป็นศูนย์หมด (all-zero string) ด้วยขั้นตอนวิธี Blowfish โดยใช้กุญแจย่อยที่ถูกสร้างขึ้นด้วยขั้นตอนที่ 1 และ 2

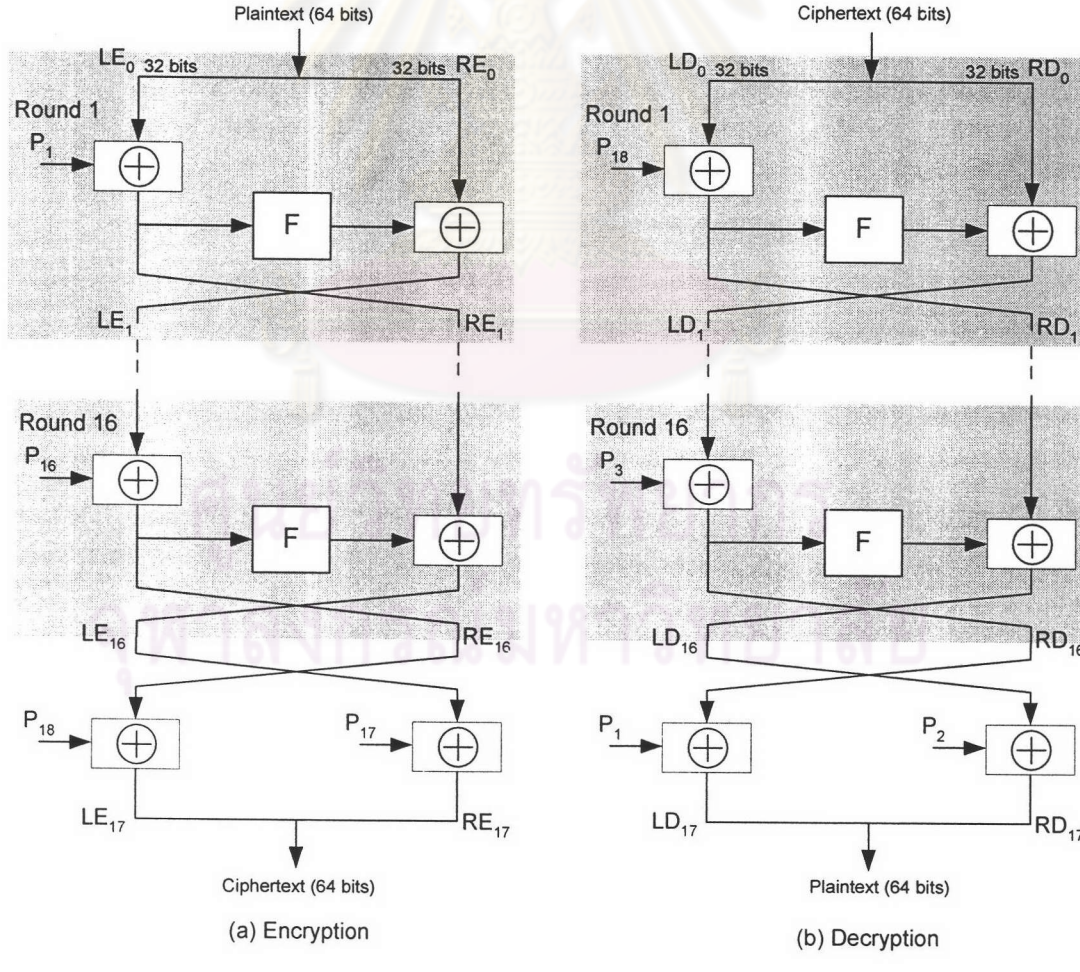
4. แทนค่า P1 และ P2 ด้วยผลลัพธ์ที่ได้จากการเข้ารหัสในขั้นตอนที่ 3

5. นำผลลัพธ์จากขั้นตอนที่ 3 มาเข้ารหัสต่อโดยใช้กุญแจย่อยที่ถูกเปลี่ยนแปลงแล้ว

6. แทนค่า P3 และ P4 ด้วยผลลัพธ์ที่ได้จากการเข้ารหัสในขั้นตอนที่ 5

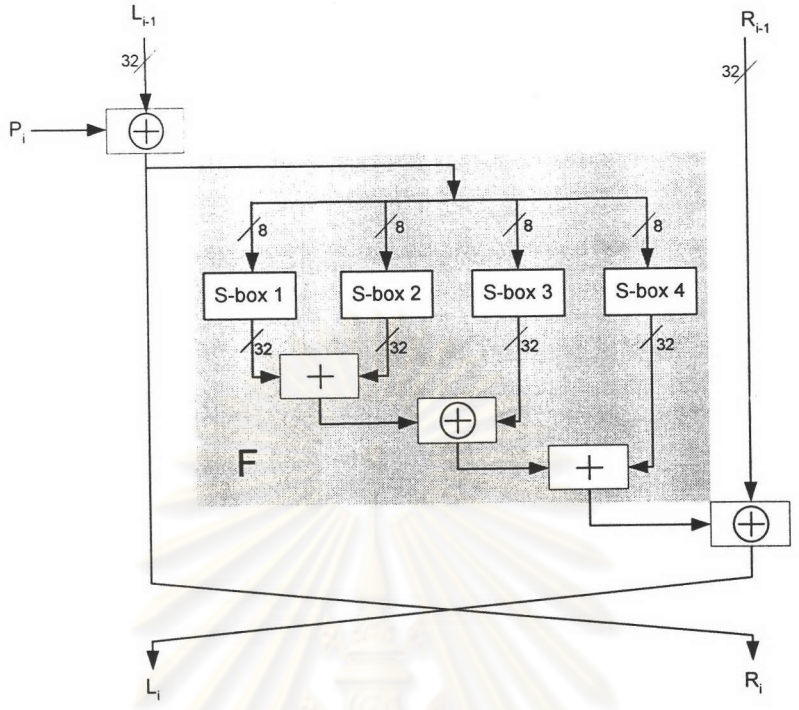
7. ทำซ้ำไปเรื่อยๆ จนกระทั่งค่าในกล่องสับเปลี่ยนและแทนค่าถูกแทนด้วยค่าใหม่จนครบทั้งหมด

ขั้นตอนวิธีการเข้าและถอดรหัสแบบ Blowfish สามารถอธิบายได้โดยรูปที่ 2.13



รูปที่ 2.13 แสดงขั้นตอนวิธีการเข้ารหัสและถอดรหัสแบบ Blowfish

โดยที่ฟังก์ชัน F มีรายละเอียดของโครงสร้างดังแสดงในรูปที่ 2.14



รูปที่ 2.14 รายละเอียดของฟังก์ชัน F

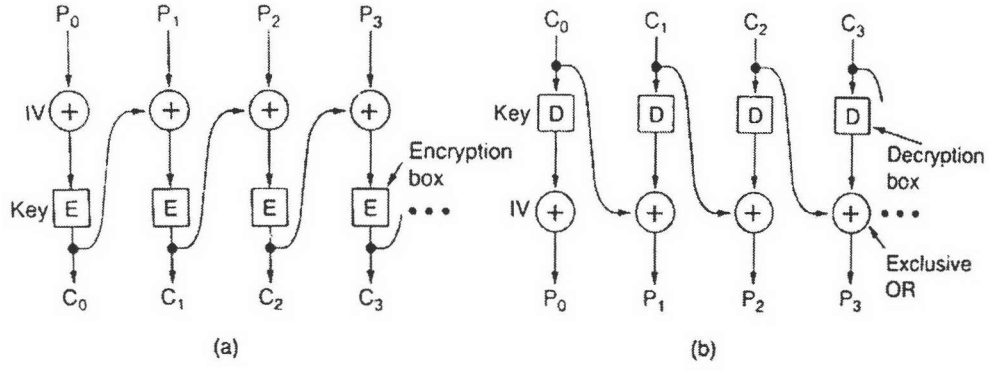
สำหรับรายละเอียดของการประยุกต์ใช้ขั้นตอนวิธีการเข้ารหัสแบบ Blowfish โดยการเขียนโปรแกรมจะอยู่ในหัวข้อ 5.3.3

2.4.5 วิธีดำเนินการกับรหัสแบบบล็อก (Block Cipher Modes of operation)

เป็นการนำข้อมูลที่ถูกเข้ารหัสแบบแยกกันแต่ละบล็อกมาผ่านกระบวนการบางอย่างเพื่อทำให้รหัสลับมีความซับซ้อนมากยิ่งขึ้น วิธีหลักๆ ที่ใช้ได้แก่ [14] และ [15]

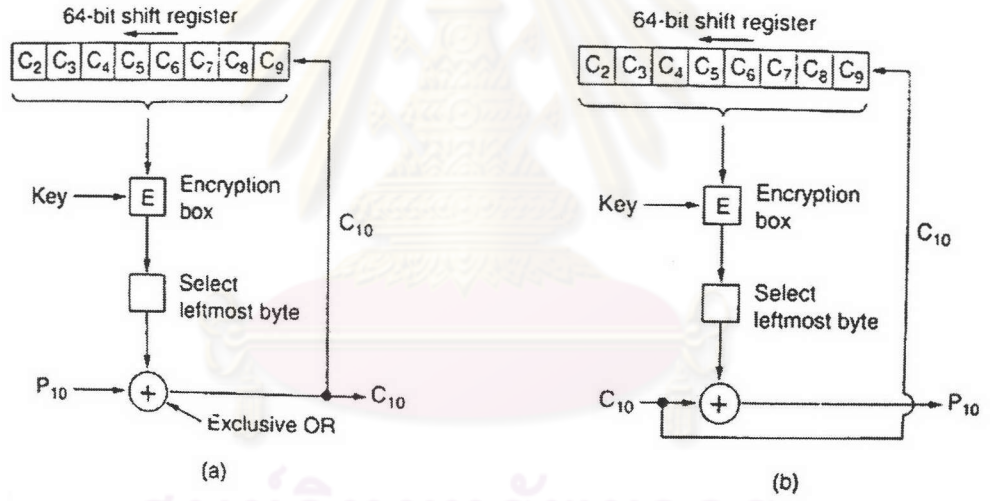
1) Electronic Codebook (ECB) ข้อมูลที่มีความยาวมากกว่า 1 บล็อกจะถูกแบ่งออกจากกัน โดยข้อมูลในแต่ละบล็อกจะถูกเข้ารหัสแยกกันโดยอิสระและไม่มีการดำเนินการใดๆ กับข้อมูลในบล็อกอื่น ดังนั้นจึงมีคุณสมบัติเช่นเดียวกับการเข้ารหัสแบบแทนที่เพียงแต่ต่างกันตรงที่ข้อมูลถูกแบ่งเป็นบล็อกแทนที่จะเป็นหนึ่งตัวอักษร ทำให้ข้อมูลมีความเสี่ยงต่อการถูกลอบถอดรหัสได้ง่ายกว่าวิธีอื่น

2) Cipher Block Chaining (CBC) บล็อกข้อมูลที่จะถูกเข้ารหัสจะถูกออร์เฉพาะกับข้อมูลที่ถูกเข้ารหัสแล้วของบล็อกก่อนหน้า ทำให้รหัสลับแต่ละบล็อกมีความเกี่ยวเนื่องกันตลอด หากมีความผิดพลาดเกิดขึ้นเพียงบิตเดียวก็จะมีผลต่อการเข้ารหัสของบล็อกที่ตามหลังมาทั้งหมด รายละเอียดของวิธีนี้สามารถแสดงได้ดังในรูปที่ 2.15



รูปที่ 2.15 โครงสร้างการเข้าและถอดรหัสแบบบล็อกด้วยวิธี CBC

3) Cipher Feedback (CFB) เนื่องจากวิธีในข้อ 2) มีข้อเสียอีกประการคือ ต้องรอให้ข้อมูลทั้งบล็อกเข้ามาหมดเสียก่อนจึงจะเริ่มถอดรหัสได้ วิธีการป้อนกลับรหัสลับนี้มีข้อดีกว่าตรงที่เป็นการเข้าและถอดรหัสแบบทีละไบต์ โดยการดึงเอาไบต์ด้านซ้ายสุดของข้อมูลที่ออกมาจากบล็อกเข้ารหัสมาทำการออร์เฉพาะกับข้อมูลที่จะถูกส่งออกไป ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 โครงสร้างการเข้าและถอดรหัสแบบบล็อกด้วยวิธี CFB

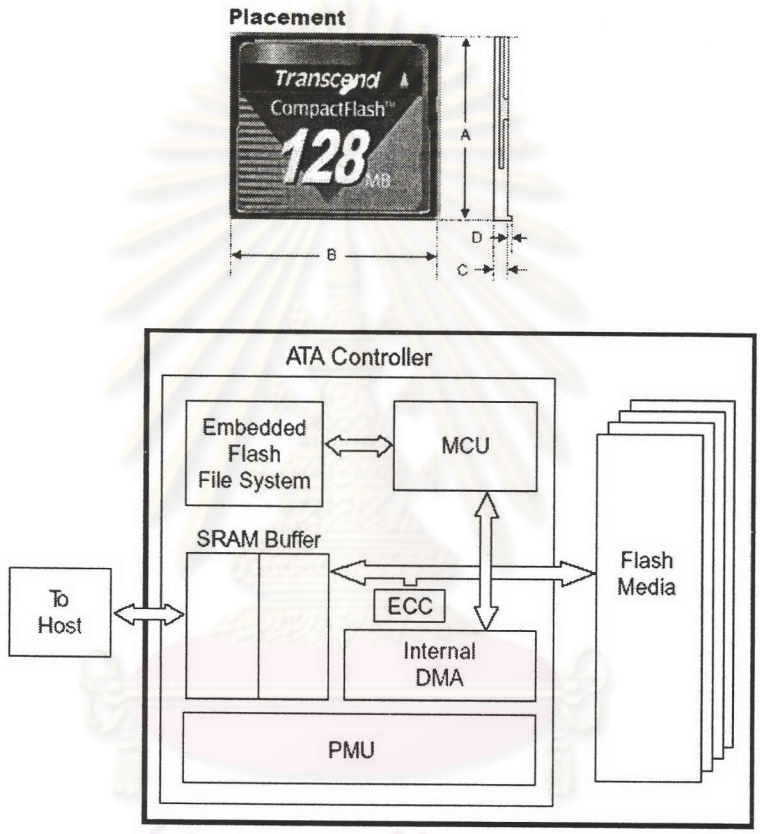
4) Output Feedback (OFB) คล้ายกับวิธีในข้อ 3) ต่างกันเพียงแค่จุดที่นำไปป้อนกลับเป็นข้อมูลจากไบต์ซ้ายสุดที่ออกมาจากบล็อกเข้ารหัสแทนที่จะเป็นรหัสลับ C_i

2.5 รายละเอียดของอุปกรณ์หน่วยเก็บรวมชนิดคอมแพคแฟลช (CompactFlash)

เนื่องจากอุปกรณ์แบบมือถือที่ถูกพัฒนาขึ้นนั้นต้องสามารถเก็บข้อมูลจำนวนมากจากมิเตอร์หลายตัวของแต่ละพื้นที่ในคราวเดียวกันได้ นอกจากนี้ตัวอุปกรณ์ต้องมีขนาดเล็กเพื่อความสะดวกในการใช้งานจริง ดังนั้นจึงเลือกใช้อุปกรณ์หน่วยเก็บรวมชนิดคอมแพคแฟลชซึ่งใช้มาตรฐานในการอ่านเขียนเช่นเดียวกันกับฮาร์ดดิสก์คือ โพรโตคอลและการเชื่อมต่อตามมาตรฐานไอดีอี/เอทีเอ (IDE/ATA standard Interface and protocol)

2.5.1 โครงสร้างภายในของคอมแพคแฟลช

คอมแพคแฟลชเป็นหน่วยความจำแฟลชชนิดหนึ่งที่มีคุณสมบัติและการทำงานตามมาตรฐานเอทีเอ จึงสนับสนุนกระบวนการติดต่อด้วยวิธีไอดีอีแบบแท้จริง (True IDE Mode) เช่นเดียวกับฮาร์ดดิสก์ชนิดไอดีอี ภายในตัวคอมแพคแฟลชมีหน่วยควบคุมที่ทำหน้าที่ต่อประสานระหว่างกลุ่มของหน่วยความจำแฟลชที่อยู่ภายในกับอุปกรณ์ที่จะมาเข้าถึง รูปร่างและโครงสร้างภายในของคอมแพคแฟลชเป็นดังรูปที่ 2.17 [8]



รูปที่ 2.17 รูปร่างและโครงสร้างภายในของคอมแพคแฟลช

2.5.2 การอ่านและเขียนข้อมูลลงในคอมแพคแฟลช

หน่วยย่อยที่สุดในการอ่านเขียนข้อมูลในคอมแพคแฟลชมีขนาดเท่ากันกับฮาร์ดดิสก์คือ 1 เซกเตอร์ (sector) โดยแต่ละเซกเตอร์มีขนาด 512 ไบต์ กระบวนการอ่านหรือเขียนข้อมูลจะกระทำผ่านทางชุดรีจิสเตอร์เอทีเอ (ATA Drive Register Set) ดังรายละเอียดในตารางที่ 2.2 [7] และ [8]

ตารางที่ 2.2 ชุดรีจิสเตอร์เอทีเอที่ใช้ติดต่อกับคอมแพคแฟลช

-REG	A10	A9-A4	A3	A2	A1	A0	Offset	-OE=0	-WE=0
1	0	X	0	0	0	0	0	Even RD Data	Even WR Data
1	0	X	0	0	0	1	1	Error	Features
1	0	X	0	0	1	0	2	Sector Count	Sector Count
1	0	X	0	0	1	1	3	Sector No.	Sector No.
1	0	X	0	1	0	0	4	Cylinder Low	Cylinder Low
1	0	X	0	1	0	1	5	Cylinder High	Cylinder High
1	0	X	0	1	1	0	6	Select Card/Head	Select Card/Head
1	0	X	0	1	1	1	7	Status	Command

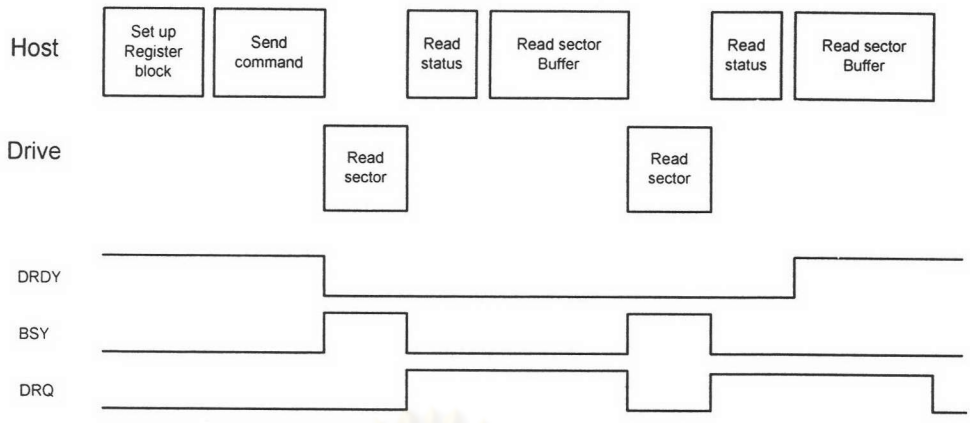
ดังนั้นในการอ่านหรือเขียนข้อมูลลงในเซกเตอร์ของคอมแพคแฟลชแต่ละครั้งจึงต้องกำหนดค่าลงในรีจิสเตอร์เหล่านี้ให้ครบทุกตัวเสียก่อนซึ่งเปรียบเสมือนเป็นการป้อนคำสั่งเป็นชุดๆ โดยระบุตำแหน่งของเซกเตอร์ที่ต้องการจะอ่านหรือเขียนผ่านทางรีจิสเตอร์ตำแหน่งที่ 2 ถึง 6 แล้วตามด้วยการป้อนคำสั่งผ่านทางรีจิสเตอร์ตำแหน่งที่ 7 ส่วนการตรวจสอบสถานะของตัวบ่งชี้ต่างๆ (Flag) สามารถตรวจสอบได้จากการอ่านค่าใน รีจิสเตอร์ตำแหน่งที่ 1 และ 7

ตัวอย่างของการป้อนชุดคำสั่งสำหรับอ่านข้อมูลจากแต่ละเซกเตอร์ (Read sector) เป็นดังรูปที่ 2.18

Bit ->	7	6	5	4	3	2	1	0
Command (7)	20H or 21H							
C/D/H (6)	1	LBA	1	Drive	Head (LBA 27-24)			
Cyl High (5)	Cylinder High (LBA 23-16)							
Cyl Low (4)	Cylinder Low (LBA 15-8)							
Sec Num (3)	Sector Number (LBA 7-0)							
Sec Cnt (2)	Sector Count							
Feature (1)	X							

รูปที่ 2.18 ตัวอย่างชุดคำสั่งสำหรับอ่านข้อมูลจากแต่ละเซกเตอร์

ส่วนการเคลื่อนย้ายข้อมูลจากเซกเตอร์ภายหลังจากที่ป้อนคำสั่งและตรวจสอบสถานะของตัวบ่งชี้ต่างๆ เป็นที่เรียบร้อยแล้วนั้น จะกระทำผ่านทางรีจิสเตอร์ตำแหน่งที่ 0 ซึ่งเป็นรีจิสเตอร์ข้อมูลเพียงตัวเดียวควบคู่ไปกับการส่งสัญญาณสโตรปจำนวน 512 ครั้งเพื่ออ่านข้อมูลจากเซกเตอร์ซึ่งถูกเก็บพักเอาไว้ในบัฟเฟอร์ในตัวคอมแพคแฟลช [9] ตัวอย่างขั้นตอนต่างๆ ที่เกิดขึ้นและสถานะของตัวบ่งชี้เนื่องจากการอ่านข้อมูลจากเซกเตอร์แต่ละครั้งเป็นดังรูปที่ 2.19 [7]



รูปที่ 2.19 ขั้นตอนต่างๆ และสถานะของตัวบ่งชี้เนื่องจากคำสั่งอ่านเซกเตอร์

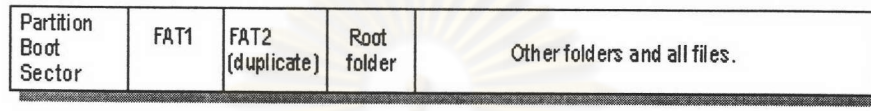
รายละเอียดของชุดคำสั่งตามมาตรฐานเอทีเอแสดงดังตารางที่ 2.3 โดยค่าในช่อง 'Code' ก็คือรหัสของคำสั่งที่จะต้องถูกป้อนเข้าไปผ่านทางรีจิสเตอร์ตำแหน่งที่ 7 นั่นเอง

ตารางที่ 2.3 ชุดคำสั่งต่างๆ ตามมาตรฐานเอทีเอ

Command	Code	FR[1]	SC[2]	SN[3]	CY[4]	DH[5]	LBA[6]
Check Power Mode	E5H or 98H	-	-	-	-	D	-
Execute Drive Diagnostic	90H	-	-	-	-	D	-
Erase Sector(s)	C0H	-	Y[7]	Y	Y	Y[8]	Y
Format Track	50H	-	Y	-	Y	Y	Y
Identify Drive	ECH	-	-	-	-	D	-
Idle	E3H or 97H	-	Y	-	-	D	-
Idle Immediate	E1H or 95H	-	-	-	-	D	-
Initialize Drive Parameters	91H	-	Y	-	-	Y	-
Read Buffer	E4H	-	-	-	-	D	-
Read Long Sector	22H or 23H	-	-	Y	Y	Y	Y
Read Multiple	C4H	-	Y	Y	Y	Y	Y
Read Sector(s)	20H or 21H	-	Y	Y	Y	Y	Y
Read Verify Sector(s)	40H or 41H	-	Y	Y	Y	Y	Y
Recalibrate	1XH	-	-	-	-	D	-
Request Sense	03H	-	-	-	-	D	-
Seek	7XH	-	-	Y	Y	Y	Y
Set Features	EFH	Y	-	-	-	D	-
Set Multiple Mode	C6H	-	Y	-	-	D	-
Set Sleep Mode	E6H or 99H	-	-	-	-	D	-
Stand By	E2H or 96H	-	-	-	-	D	-
Stand By Immediate	E0H or 94H	-	-	-	-	D	-
Translate Sector	87H	-	Y	Y	Y	Y	Y
Wear Level	F5H	-	-	-	-	Y	-
Write Buffer	E8H	-	-	-	-	D	-
Write Long Sector	32H or 33H	-	-	Y	Y	Y	Y
Write Multiple	C5H	-	Y	Y	Y	Y	Y
Write Multiple w/o Erase	CDH	-	Y	Y	Y	Y	Y
Write Sector(s)	30H or 31H	-	Y	Y	Y	Y	Y
Write Sector(s) w/o Erase	38H	-	Y	Y	Y	Y	Y
Write Verify	3CH	-	Y	Y	Y	Y	Y

2.5.3 การบันทึกข้อมูลในรูปของตารางการจัดสรรแฟ้มหรือแฟต (FAT)

เนื่องจากข้อมูลที่ถูกอ่านมาจากมิเตอร์ซึ่งถูกบันทึกลงในคอมแพคแฟลชบนตัวอุปกรณ์แบบพกพาจะต้องถูกโอนย้ายลงในฐานข้อมูลบนเครื่องคอมพิวเตอร์ด้วยโดยใช้ซอฟต์แวร์ที่พัฒนาขึ้นด้วย ดังนั้นเพื่อให้สะดวกต่อการพัฒนาซอฟต์แวร์เพื่อโอนย้ายข้อมูลจากคอมแพคแฟลชลงบนฐานข้อมูล อุปกรณ์แบบพกพาจึงควรทำการบันทึกข้อมูลที่อ่านมาจากมิเตอร์แต่ละตัวลงบนคอมแพคแฟลชให้อยู่ในรูปของแฟ้มข้อมูล (file) ตามรูปแบบของตารางการจัดสรรแฟ้มหรือแฟต โครงสร้างของแฟตมีลักษณะดังรูปที่ 2.20 และมีรายละเอียดดังนี้ [16] และ [17]



รูปที่ 2.20 โครงสร้างของตารางการจัดสรรแฟ้มหรือแฟต

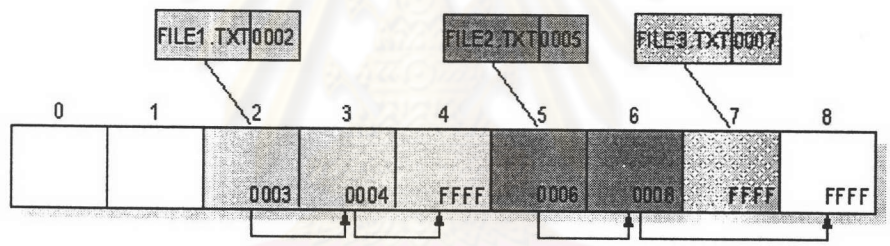
1) *บู๊ตเซกเตอร์* เป็นเซกเตอร์แรกของแต่ละการแบ่งส่วน (Partition) บู๊ตเซกเตอร์เก็บข้อมูลที่เป็นรายละเอียดต่างๆ และจำเป็นสำหรับระบบแฟ้มเช่น คำสั่งบู๊ต ขนาดของเซกเตอร์ในหน่วยไบต์ จำนวนเซกเตอร์ทั้งหมดที่มี จำนวนของตารางแฟต จำนวนเซกเตอร์ที่ใช้เก็บตารางแฟต จำนวนแฟ้มมากสุดที่มีได้บนโฟลเดอร์ราก (Directory Entry) เป็นต้น ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 รายละเอียดภายในบู๊ตเซกเตอร์

Offset	*Description*	*Size*
00h	Jump Code + NOP	3 Bytes
03h	OEM Name	8 Bytes
0Bh	Bytes Per Sector	1 Word
0Dh	Sectors Per Cluster	1 Byte
0Eh	Reserved Sectors	1 Word
10h	Number of Copies of FAT	1 Byte
11h	Maximum Root Directory Entries	1 Word
13h	Number of Sectors in Partition Smaller than 32MB	1 Word
15h	Media Descriptor (F8h for Hard Disks)	1 Byte
16h	Sectors Per FAT	1 Word
18h	Sectors Per Track	1 Word
1Ah	Number of Heads	1 Word
1Ch	Number of Hidden Sectors in Partition	1 Double Word
20h	Number of Sectors in Partition	1 Double Word
24h	Logical Drive Number of Partition	1 Word
26h	Extended Signature (29h)	1 Byte

27h	Serial Number of Partition	1 Double Word
2Bh	Volume Name of Partition	11 Bytes
36h	FAT Name (FAT16)	8 Bytes
3Eh	Executable Code	448 Bytes
1FEh	Executable Marker (55h AAh)	2 Bytes

2) *ตารางแฟต* มีหน้าที่ระบุตำแหน่งของแฟ้มที่ถูกจัดเก็บเอาไว้ว่าอยู่ที่ไหนบ้าง โดยข้อมูลที่อยู่ในแต่ละหน่วยของตารางแฟตคือตำแหน่งของกลุ่มข้อมูล (Cluster) ถัดไปที่ใช้เก็บแฟ้มนั้นๆ สำหรับ FAT16 หนึ่งหน่วยของตารางแฟตมีขนาด 2 ไบต์ซึ่งสามารถอ้างไปยังกลุ่มข้อมูลได้ทั้งสิ้น 65518 กลุ่ม ดังนั้นตารางแฟตจึงมีหน้าที่จัดเส้นทางให้กับระบบปฏิบัติการเพื่อหาตำแหน่งของกลุ่มข้อมูลต่างๆ ที่ใช้เก็บแฟ้มข้อมูลนั้นๆ ตัวอย่างลักษณะของตารางแฟต และความหมายของรหัสต่างๆ ที่อยู่ในแต่ละหน่วยของตารางแฟตชนิด FAT16 เป็นรูปที่ 2.21 และตารางที่ 2.1 ตามลำดับ



รูปที่ 2.21 ตัวอย่างการจัดเก็บตำแหน่งของกลุ่มข้อมูลในตารางแฟต

ตารางที่ 2.5 รหัสต่างๆ ที่อยู่ในตารางแฟตแบบ FAT16

FAT Code Range	***Meaning***
0000h	Available Cluster
0002h-FFEFh	Used, Next Cluster in File
FFF0h-FFF6h	Reserved Cluster
FFF7h	BAD Cluster
FFF8h-FFFF	Used, Last Cluster in File

3) *โฟลเดอร์ราก* ถูกใช้สำหรับเก็บรายละเอียดของแต่ละแฟ้มข้อมูลอันได้แก่ ชื่อและนามสกุลของแฟ้ม ประเภทของแฟ้ม วันเวลาที่สร้างแฟ้ม ตำแหน่งเริ่มต้นของกลุ่มข้อมูลที่ใช้เก็บข้อมูลในแฟ้มนั้นๆ และขนาดของแฟ้ม โดยแต่ละหน่วย (Entry) ที่ใช้เก็บข้อมูลให้กับแต่ละแฟ้มมีขนาด 32 ไบต์ รายละเอียดในแต่ละหน่วยของโฟลเดอร์รากอยู่ในตารางที่ 2.6 [18]

ตารางที่ 2.6 รายละเอียดของแต่ละหน่วยในโฟลเดอร์ราก

Offset	*Length*	*Value*
0	8 bytes	Name
8	3 bytes	Extension
11	byte	Attribute (00ARSHDV) 0: unused bit A: archive bit, R: read-only bit S: system bit D: directory bit V: volume bit
22	word	Time
24	word	Date
26	word	Starting Cluster
28	dword	File Size

4) *กลุ่มสำหรับเก็บข้อมูล (Cluster)* อยู่ถัดจากส่วนของโฟลเดอร์ราก กลุ่มข้อมูลเป็นส่วนย่อยๆ จำนวนมากที่ถูกใช้เป็นพื้นที่สำหรับเก็บข้อมูลให้กับแฟ้มข้อมูลต่างๆ โดยตำแหน่งของกลุ่มข้อมูลต่างๆ ที่ใช้เก็บข้อมูลให้กับแต่ละแฟ้มนั้นจะสัมพันธ์กับรายละเอียดในตารางแฟตเสมอ

ในส่วนของรายละเอียดของโปรแกรมย่อยที่ทำหน้าที่ควบคุมกระบวนต่างๆ สำหรับบันทึกข้อมูลลงในคอมแพคแฟลชให้อยู่ในรูปของแฟ้มข้อมูลฐานสอง (Binary file) ตามรูปแบบของระบบแฟ้มแบบแฟตอยู่ในหัวข้อ 5.4

2.6 สรุปท้ายบท

ในบทนี้กล่าวถึงความรู้พื้นฐานของระบบ AMR ซึ่งได้แก่ องค์ประกอบโดยทั่วไปและระบบ AMR ประเภทต่างๆ นอกจากนี้ยังได้กล่าวถึงความรู้และหลักการต่างๆ ที่เกี่ยวข้องซึ่งจะถูกนำมาประยุกต์ใช้ในการทำวิทยานิพนธ์ได้แก่ รายละเอียดโพรโตคอลที่ใช้ติดต่อสื่อสารกับมิเตอร์ตามมาตรฐาน ANSI C12.18-1996 ตลอดจนวิธีการคำนวณหาค่าซีอาร์ซีเพื่อใช้ตรวจสอบความถูกต้องของกลุ่มข้อมูล พื้นฐานของการเข้าและถอดรหัสลับแบบต่างๆ รายละเอียดของการเข้ารหัสแบบบล็อกด้วยขั้นตอนวิธี Blowfish ที่จะนำมาประยุกต์ใช้ และในหัวข้อสุดท้ายเป็นการอธิบายรายละเอียดและการอ่านเขียนข้อมูลลงในอุปกรณ์หน่วยเก็บรวมชนิดคอมแพคแฟลชซึ่งจะถูกนำมาใช้เก็บข้อมูลที่อ่านได้จากมิเตอร์ และการบันทึกข้อมูลในรูปของระบบแฟ้มข้อมูลชนิดแฟตเพื่อให้สะดวกต่อการพัฒนาซอฟต์แวร์ถ่ายโอนซึ่งต้องอ่านข้อมูลมาจากตัวคอมแพคแฟลชนี้ด้วย