บทที่ 2

หลักการพื้นฐานของกุญแจอิเล็กทรอนิกส์

จากบทที่แล้ว ผู้วิจัยได้กล่าวถึงความเป็นมาของปัญหาที่พบในการใช้งานซอฟแวร์ประยุกต์ที่จำเป็นต้องจำกัดสิทธิ์ผู้ใช้งาน โดยมุ่งเน้นความสนใจไปที่การสร้างกุญแจอิเล็กทรอนิกส์ติดต่อผ่านทาง พอร์ต USB ของเครื่องคอมพิวเตอร์ส่วนบุคคล ดังนั้นเพื่อสร้างความเข้าใจมากขึ้นในการสร้างกุญแจอิเล็กทรอนิกส์เนื้อหาในบทนี้จึงได้อธิบายถึงพอร์ต USB การติดต่อกับพอร์ต USB อุปกรณ์ที่เชื่อมต่อกับพอร์ต USB และพื้นฐานการเข้ารหัสลับ การเข้ารหัสลับแบบ RSA อัลกอริทึมการลดทอนแบบ Montgomery ซึ่งเป็นแนวทางการออกแบบกุญแจอิเล็กทรอนิกส์ที่จะกล่าวในบทที่ 3

2.1 พอร์ต USB

2.1.1 ลักษณะทั่วไปของพอร์ต USB

พอร์ตต่าง ๆ ในเครื่องคอมพิวเตอร์ส่วนบุคคล ถูกออกแบบมาเพื่อใช้งานเฉพาะอย่าง ทำให้อุปกรณ์แต่ละตัวต้องเลือกใช้พอร์ตต่างชนิดกันไป เช่น พอร์ตอนุกรม ใช้ต่อกับ โมเดม และ เมาส์ ส่วนพอร์ตขนานใช้ต่อกับเครื่องพิมพ์ ทำให้ผู้ใช้งานทั่วไปเกิดความยุ่งยาก และไม่แน่ใจถึงความเสียหายที่อาจเกิดขึ้น หากต่ออุปกรณ์กับพอร์ตผิดชนิด นอกจากนี้ในการออกแบบคอมพิวเตอร์หากมีพอร์ตมากขึ้น ดังนั้นโอกาสจะเกิดปัญหาการแย่งกันใช้การร้องขอสัญญาณขัดจังหวะ (Interrupt Request) ซึ่งเป็นส่วนหนึ่งที่จำกัดการติดต่อกับอุปกรณ์ภายนอกของคอมพิวเตอร์ ทำให้เกิดแนวคิดที่จะกำหนดมาตรฐานเพื่อสร้างเป็นพอร์ต ที่ทำให้การติดต่อกับอุปกรณ์ภายนอกอยู่ในรูปแบบเดียวกันง่ายสำหรับผู้ใช้ทั่วไป และไม่มีข้อจำกัดในการใช้ร้องขอสัญญาณขัดจังหวะนั้นคือ Universal Serial Bus (USB) มีลักษณะเด่นดังนี้

- ใช้ตัวเชื่อมต่อเพียงชนิดเดียว ต่อกับอุปกรณ์ได้ทุกชนิดเช่น เมาส์,แผงแป้นอักขระ, จานบันทึกแบบแข็งและเครื่องพิมพ์ เป็นต้น
- ไม่เกิดการแย่งกันใช้การร้องขอสัญญาณขัดจังหวะ เนื่องจากอุปกรณ์ทุกตัวจะมีหมายเลขประจำตัว(Product ID) ทำให้แม้ว่า USB จะเป็นการสื่อสารแบบอนุกรม แต่ข้อมูลจะถึงปลายทางได้โดยเลขประจำตัวของอุปกรณ์
- สามารถรองรับอุปกรณ์ได้มากที่สุด 127 อุปกรณ์
- ทำงานอัตโนมัติเมื่อต่ออุปกรณ์เข้ากับเครื่องคอมพิวเตอร์

- มีแรงดันจ่ายให้อุปกรณ์ 5 โวลต์ เพื่ออำนวยความสะดวกกับอุปกรณ์แบบพกพา เช่น ipod MP3, จานบันทึกแบบแข็ง USB เป็นต้น
- มีกระแสจ่ายให้อุปกรณ์ 500 มิลลิแอมแปร์

2.1.2 การติดต่อกับ USB Port

กำหนดให้ เครื่องแม่ข่าย คือ เครื่องคอมพิวเตอร์
อุปกรณ์ คือ กุญแจอิเล็กทรอนิกส์

หลักการทำงานของเครื่องคอมพิวเตอร์เมื่อต้องการติดต่อกับอุปกรณ์ USB ในที่นี้จะกล่าวถึงเฉพาะระบบปฏิบัติการ วินโดส์ 98 SE ขึ้นไป เนื่องจากวินโดส์ที่ต่ำกว่านี้ไม่สามารถใช้อุปกรณ์ USB ได้หรือใช้ได้แต่ไมโครซอฟต์ไม่แนะนำ ส่วนระบบปฏิบัติการอื่น ๆจะไม่ขอกล่าวถึง

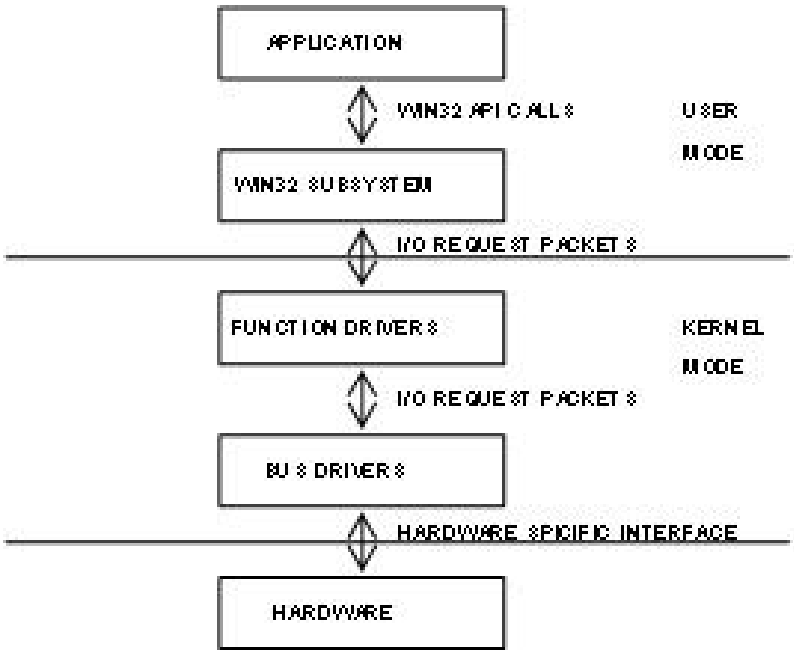
เมื่อโปรแกรมประยุกต์ต้องการติดต่อกับฮาร์ดแวร์ซึ่งเป็นอุปกรณ์ USB จะต้องผ่านขั้นตอนกระบวนการที่ไมโครซอฟต์กำหนดไว้ดังนี้

โปรแกรมประยุกต์ ซึ่งจะอยู่ด้านบนสุดจะต้องติดต่อผ่านไดนามิกส์ลิงค์ไลบรารีของวินโดวส์ซึ่งเตรียมไว้ให้ผู้ใช้พร้อมใช้งาน ประกอบด้วย 3 ไฟล์หลักดังนี้

1. hid.dll ใช้สำหรับติดต่อกับอุปกรณ์ USB ที่เสียบเข้ามาใหม่
2. setupapi.dll ใช้ในการหาลักษณะเฉพาะของอุปกรณ์เช่น หมายเลขประจำผลิตภัณฑ์ (Product Identification :PID),หมายเลขประจำตัวผู้ผลิต (Vender Identification:VID)
3. kernel32.dll ใช้ในการแลกเปลี่ยนข้อมูลกับอุปกรณ์

เพื่อให้ผู้ใช้งานอุปกรณ์ เขียนโปรแกรมติดต่อกับอุปกรณ์ได้สะดวกยิ่งขึ้นวินโดวส์ได้เตรียมส่วนต่อประสานโปรแกรมประยุกต์ (Application Program Interface: API) เพื่อดึงค่าฟังก์ชันต่าง ๆในไดนามิกส์ลิงค์ไลบรารี่ทั้ง 3 ไฟล์นี้มาใช้งานได้ เมื่อเรียกใช้ส่วนต่อประสานโปรแกรมประยุกต์ในส่วนของซอฟแวร์ของวินโดวส์จะจัดการกับระบบเอง โดยจะติดต่อกับใจกลาง (Kernel) ไปจนถึงฮาร์ดแวร์ซึ่งในที่นี้คือกุญแจอิเล็กทรอนิกส์แสดงในภาพประกอบ 2-1

ในการสั่งงานจริงส่วนต่อประสานโปรแกรมประยุกต์ที่เรียกใช้จากวินโดส์ค่อนข้างจะเป็นวิธีเรียกใช้แบบทั่วไป แต่ในการส่งข้อมูลไปให้อุปกรณ์นั้น จำเป็นต้องทราบว่าอุปกรณ์ที่มีอยู่นั้นรับส่งข้อมูลในรูปแบบใด ซึ่งบริษัทผู้ผลิตอุปกรณ์จะให้ข้อมูลส่วนนี้ กับผู้ที่ต้องการนำเอาอุปกรณ์ไปพัฒนา จึงจะสามารถแลกเปลี่ยนข้อมูลกับอุปกรณ์ได้



ภาพประกอบ 2-1 ลักษณะการติดต่อระหว่างซอฟต์แวร์กับฮาร์แวร์

(ที่มา Axelson Jan,2001)

2.1.3 อุปกรณ์ต่อประสานกับพอร์ต USB

เพื่อให้เครื่องแม่ข่ายติดต่อกับกุญแจอิเล็กทรอนิกส์โดยผ่านอุปกรณ์ต่อประสานตัวนี้ ซึ่งอุปกรณ์ต่อประสานกับพอร์ต USB มีคุณสมบัติดังนี้

- สนับสนุนการถ่ายโอนแบบควบคุมและแบบขัดจังหวะ ซึ่งการถ่ายโอนแบบควบคุมใช้สำหรับการติดต่อครั้งแรกที่มีการเสียบอุปกรณ์เข้ากับเครื่องแม่ข่าย เพื่อให้เครื่องแม่ข่ายทราบถึง อุปกรณ์ที่เสียบเข้ามาในเครื่องแม่ข่ายเป็นอุปกรณ์ที่อยู่ในชั้นใด ดังนั้นจึงจำเป็นต้องมีการถ่ายโอนแบบควบคุมในทุก ๆ อุปกรณ์ USB และการถ่ายโอนแบบขัดจังหวะใช้สำหรับวนอ่านข้อมูลใน อุปกรณ์
- มี Endpoint ที่พร้อมใช้งาน 2 Endpoint ซึ่ง Endpoint คือตำแหน่งสุดท้ายที่เครื่องแม่ข่ายติดต่อมายังอุปกรณ์ ดังนั้นต้องใช้ 1 Endpoint ในการถ่ายโอนแบบควบคุมเพื่ออ่านค่าอุปกรณ์และ 1 Endpoint สำหรับการถ่ายโอนแบบขัดจังหวะเพื่อรับส่งข้อมูล
- ต้องการแรงดันขณะทำงานไม่เกิน 5 โวลต์และกินกระแสน้อยกว่า 500 มิลลิแอมแปร์ ซึ่งเป็นข้อจำกัดของพอร์ต USB
- สามารถต่อประสานกับไมโครคอนโทรลเลอร์อีกตัวหนึ่งได้โดยผ่านทางพอร์ตขนาน
- มีขนาดเล็กเพื่อให้เหมาะกับงานประเภทนี้

- อุปกรณ์ที่มีลักษณะสอดคล้องตามที่กล่าวมาได้แก่ ชิปของบริษัท NetChip เบอร์ NET2888 ,ชิปของบริษัท ไมโครชิป เบอร์PIC16C745 และชิปของบริษัท Philips เบอร์ PDIUSBD11, D12 เป็นต้น

2.1.4 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ทำหน้าที่ประมวลผลการเข้าและถอดรหัสลับแบบ RSA พร้อมตรวจสอบความถูกต้องของกุญแจอิเล็กทรอนิกส์และเก็บข้อมูล สำหรับให้เครื่องแม่ข่ายวนมาอ่านเพื่อแสดงให้เห็นว่าผู้ใช้งานเสียบกุญแจอิเล็กทรอนิกส์เข้ากับระบบก่อนการใช้งาน ซึ่งไมโครคอนโทรลเลอร์ที่ใช้ในการเก็บข้อมูลดังกล่าว จะต้องมีคุณสมบัติดังนี้

- มีขนาดของหน่วยความจำโปรแกรมและหน่วยความจำชั่วคราวขนาดใหญ่
- มีความสามารถในการป้องกันการอ่านข้อมูลที่อยู่ภายในตัวอุปกรณ์ได้ ซึ่งเป็นผลให้ไม่สามารถทำกุญแจอิเล็กทรอนิกส์ซ้ำขึ้นมาใหม่ได้
- มีขนาดเล็ก ต้องการกระแสน้อยกว่า 500 มิลลิแอมแปร์
- อุปกรณ์ที่มีลักษณะสอดคล้อง ตามที่กล่าวมาได้แก่ PIC16F87xA ,PIC18F242

2.1.5 USB ไมโครคอนโทรลเลอร์

USB ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ที่รวมเอาทั้งข้อ 2.1.3 และ 2.1.4 มารวมกัน ซึ่งลักษณะการทำงานก็จะคล้ายกัน เพียงแต่รวมเอาทั้งอุปกรณ์ดังกล่าวมารวมเข้าด้วยกันเพื่อประหยัดต้นทุนในการสร้าง ทั้งนี้อุปกรณ์ USB ไมโครคอนโทรลเลอร์ที่ใช้ทำเป็นกุญแจอิเล็กทรอนิกส์ต้องมีคุณสมบัติคล้ายคลึงกับที่กล่าวมาอุปกรณ์ USB ประเภทนี้ได้แก่ ชิปของบริษัท Cypress เบอร์ CY7C637xx, ชิปของบริษัท ScanLogic เบอร์ SL11R ซึ่งในงานวิจัยนี้ไม่สามารถนำอุปกรณ์ประเภทนี้มาประยุกต์ใช้ได้ เนื่องจากไม่สามารถหาซื้อได้ภายในประเทศ รวมถึงอุปกรณ์มีขนาดใหญ่ไม่เหมาะกับงานประเภทนี้

2.1.6 ชนิดการถ่ายโอนข้อมูลภายในบัส USB (Transfer Type)

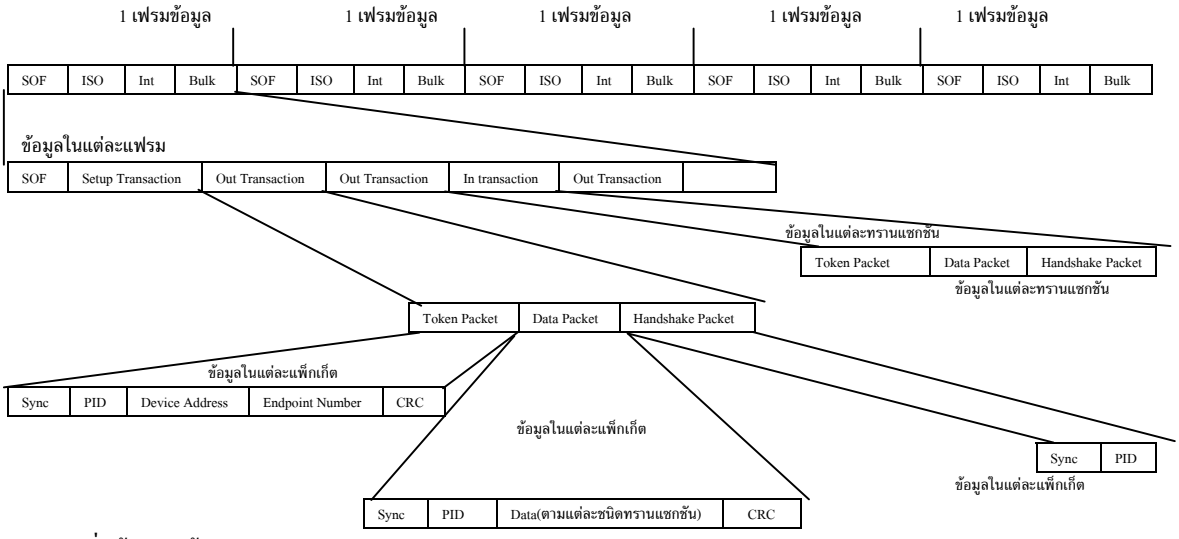
2.1.6.1 การถ่ายโอนแบบควบคุม (Control Transfer)

อุปกรณ์ USB ทุกชนิดจะต้องมีการถ่ายโอนแบบนี้ ก่อนที่จะเริ่มต้นการถ่ายโอนแบบอื่น เมื่อเริ่มที่จะอ่านข้อมูลดิสคริปเตอร์ต่าง ๆ เครื่องแม่ข่ายได้ข้อมูลมาจากอุปกรณ์ เครื่องแม่ข่ายจะพิจารณาว่าสามารถรองรับการทำงานได้หรือไม่ และจะทำการตั้งค่าการทำงานต่าง ๆรวมถึงการกำหนดชั้นของอุปกรณ์ เมื่อตั้งค่าได้แล้วอุปกรณ์จะเริ่มทำงาน ทั้งหมดนี้ใช้ผ่าน Endpoint 0 ภายใต้การถ่ายโอนแบบควบคุม

2.1.6.2 การถ่ายโอนแบบขัดจังหวะ (Interrupt Transfer)

การรับส่งข้อมูลชนิดนี้ สร้างขึ้นมาเพื่อเลียนแบบการสร้างสัญญาณแบบขัดจังหวะของอุปกรณ์ให้แก่ระบบ วิธีเลียนแบบนี้จะอาศัยการวนอ่านข้อมูลจากตัวอุปกรณ์ต่าง ๆ ในระยะเวลาที่กำหนดอย่างสม่ำเสมอหรือเรียกว่า การหยั่งสัญญาณ (Polling) โดยอัตราการวนอ่านข้อมูลนี้ จะต้องไม่ช้าเกินไปเพราะจะทำให้เกิดการสูญเสียข้อมูลที่อ่านไม่ทัน และต้องไม่เร็วเกินไป เพราะจะทำให้สูญเสียความกว้างของแถบความถี่ และหากอุปกรณ์ไม่มีข้อมูลที่ต้องการส่งกลับยังเครื่องแม่ข่าย อุปกรณ์จะส่งสัญญาณ No Acknowledge (NAK) ตอบกลับ

คาบเวลาในการวนอ่านข้อมูลอุปกรณ์แต่ละตัว จะขึ้นอยู่กับความต้องการของอุปกรณ์แต่ละชนิด คาบเวลานี้สามารถเปลี่ยนได้ในช่วงกว้างตั้งแต่การอ่านทุก ๆ 1 เฟรม (1 มิลลิวินาที) ไปจนถึงทุก ๆ 255 เฟรม เครื่องแม่ข่ายจะรู้คาบเวลาในการวนอ่านของ Endpoint ที่ใช้วิธีการถ่ายโอนแบบขัดจังหวะจากการอ่าน เอ็นด์พอยต์ดิสคริปเตอร์



SOF = เริ่มต้นเฟรมข้อมูล

ISO = การถ่ายทอดข้อมูลแบบไอโซโครนัส

Int = การถ่ายทอดข้อมูลแบบอินเตอร์รัฟต์

Bulk = การถ่ายทอดข้อมูลแบบบัลค์

Con = การถ่ายทอดข้อมูลควบคุม

ภาพประกอบ 2-2 โครงสร้างการส่งข้อมูลของ USB

การถ่ายโอนแบบขัดจังหวะนี้ แม้ว่าต้องการอัตราการวนอ่านข้อมูลที่คงที่ หากเกิดความผิดพลาดของการส่งข้อมูลซึ่งก็สามารถส่งใหม่ได้ ทำให้การถ่ายโอนแบบนี้จึงต้องมีการตรวจสอบความถูกต้องของอุปกรณ์ที่ใช้การถ่ายโอนแบบนี้ และเมื่อเรากำหนดให้มีการตรวจสอบ รหัสประจำตัวของผู้ใช้ในกุญแจอิเล็กทรอนิกส์สามารถทำได้ด้วยวิธีนี้

2.1.6.3 คลาส HID (Human Interface Device Class)

ในการติดต่อผ่าน USB นั้นสามารถแบ่งคลาสออกเป็นหลาย ๆคลาส ดังนี้

- อุปกรณ์คลาสแสดงผล ได้แก่ จอภาพ
- อุปกรณ์คลาสการสื่อสาร ได้แก่ โมเด็ม
- อุปกรณ์คลาสเสียง ได้แก่ ลำโพง
- อุปกรณ์คลาสจัดเก็บข้อมูล ได้แก่ จานแม่เหล็กแข็ง, แฮนด์ดี้ไดร์ ,ดิสก์ไดร์
- อุปกรณ์คลาส HID ได้แก่ เมาส์,แผงแป้นอักขระ,จอยสติกส์

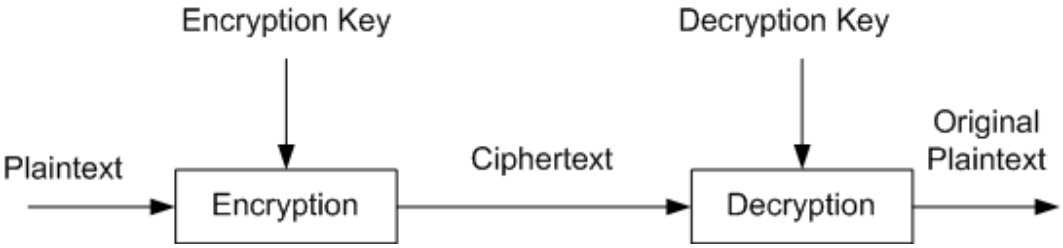
สำหรับเอกสารนี้จะกล่าวถึงเฉพาะคลาส HID เนื่องจากกุญแจอิเล็กทรอนิกส์จัดอยู่ในคลาสนี้ในการจัดให้อุปกรณ์ USB อยู่ในคลาส HID นั้นจะให้ความสำคัญไปที่ขนาดของข้อมูลที่สื่อสารระหว่าง เครื่องแม่ข่ายกับอุปกรณ์โดยจะแตกต่างกันไปตามความเร็วของอุปกรณ์ USB ซึ่งกุญแจอิเล็กทรอนิกส์เป็นอุปกรณ์ UBS ความเร็วต่ำมีขนาดของข้อมูลใน 1 การถ่ายโอนเท่ากับ 8 ไบต์ กุญแจอิเล็กทรอนิกส์จะติดต่อกับ HID ได้ 2 ลักษณะคือ สัญญาณควบคุมเพื่อกำหนดให้มีการวนรออ่านข้อมูลโดยใช้ Endpoint O และสัญญาณ Interrupt เพื่อใช้ในการรับส่งข้อมูลใน Endpoint 1 ซึ่ง HID นี้สามารถใช้ วิชวล C++ หรือวิชวลเบสิก 6.0 ในการเขียนโปรแกรมกำหนดค่าต่าง ๆ ได้ตามต้องการ

2.2 การเข้ารหัสลับข้อมูล

ในระบบจริงจะเกิดการสื่อสารระหว่างแม่ข่ายกับอุปกรณ์ USB จึงต้องป้องกันข้อมูลระหว่างการสื่อสารผ่านสายนำสัญญาณ เพื่อมิให้ ผู้ไม่ประสงค์ดี นำข้อมูลไปใช้ได้ วิธีการป้องกัน คือ นำข้อมูลมาเข้ารหัสลับ (Encryption) นั้นคือทำให้ข้อมูลนั้นเป็นความลับ ซึ่งผู้ที่มีสิทธิ์เท่านั้นที่สามารถอ่านข้อมูลและเข้าใจความหมายของข้อมูลนั้นได้ด้วยการถอดรหัสลับ (Decryption) เป็นผลให้ สามารถรักษาข้อมูลให้เป็นความลับได้ (Confidentiality) และกำหนดผู้มีสิทธิ์ (Authentication & Authorization) สำหรับการเข้ารหัสลับและถอดรหัสลับจะอาศัยสมการทางคณิตศาสตร์ที่ซับซ้อนและต้องอาศัยกุญแจซึ่งอยู่ในรูปของพารามิเตอร์ที่กำหนดไว้ (สำหรับตัวกุญแจจะมีความยาวเป็นบิต หากกุญแจมีความยาวมาก ปลอดภัยจะมากตามไปด้วย เนื่องจากจะต้องใช้เวลาเพิ่มขึ้นในการคาดเดากุญแจของผู้บุกรุก) ในการเข้าและถอดรหัสลับ สามารถแบ่งออกเป็น 2 ประเภทคือ การเข้ารหัสลับแบบกุญแจสมมาตร (Symmetric Key Cryptography หรือ Secret Key Cryptography) และการเข้ารหัสลับแบบอสมมาตร (Asymmetric Key Cryptography หรือ Public Key Cryptography) ซึ่งในที่นี้จะกล่าวถึงเฉพาะการเข้ารหัสลับแบบอสมมาตร

2.2.1 การเข้ารหัสลับแบบกุญแจอสมมาตร

การเข้าและถอดรหัสลับ ด้วยกุญแจต่างกัน มีขั้นตอนดังแสดงในภาพประกอบ 2-3 กล่าวคือเมื่อเครื่องแม่ข่ายเป็นผู้ส่ง ทำการเข้ารหัสลับข้อความ (Plaintext) ไปเป็นข้อความที่เข้ารหัสแล้ว (Ciphertext) ด้วยกุญแจสาธารณะของผู้รับซึ่งเมื่อผู้รับรับข้อมูลที่ถูกเข้ารหัสแล้ว จะนำกุญแจในการถอดรหัส ซึ่งเป็นกุญแจของตนเองทำการถอดรหัสลับข้อมูลที่ได้รับเป็นข้อความเดิมอีกครั้งหนึ่ง



ภาพประกอบ 2-3 ระบบการเข้าและถอดรหัสลับ แบบกุญแจอสมมาตร

เมื่อนำมาประยุกต์ใช้กับกุญแจอิเล็กทรอนิกส์ซึ่ง PC จะทำหน้าที่เป็นเครื่องแม่ข่าย โดยจะเข้ารหัสลับข้อมูล เป็นข้อมูลที่ถูกเข้ารหัสและถูกส่งไปยังกุญแจอิเล็กทรอนิกส์ เมื่อกุญแจอิเล็กทรอนิกส์รับมาแล้ว จะถอดรหัสลับข้อความด้วยกุญแจส่วนตัวของกุญแจอิเล็กทรอนิกส์ในการส่งข้อความด้วยการเข้ารหัสลับกุญแจแบบอสมมาตร ผู้รับจะใช้กุญแจส่วนตัวในการถอดรหัสลับ ปัจจุบันการเข้ารหัสลับแบบอสมมาตรมีหลายชนิด แต่ในที่นี่จะกล่าวถึงเฉพาะการเข้ารหัสลับแบบ RSA

2.3 การเข้ารหัสลับแบบ RSA

อัลกอริทึม RSA ได้รับการพัฒนาขึ้นที่มหาวิทยาลัย MIT ในปี 1977 โดยศาสตราจารย์ 3 คน ซึ่งประกอบด้วย Ronald Rivest, Adi Shamir และ Leonard Adleman ชื่อของอัลกอริทึมได้รับการตั้งชื่อตามตัวอักษรตัวแรกของนามสกุลของศาสตราจารย์ทั้งสามคน ซึ่งกำหนดให้ n = p*q โดย p และ q เป็นจำนวนเฉพาะที่ไม่ซ้ำกันและเขียนเป็นฟังก์ชันทอเทียนต์ออยเลอร์ (Euler totient) ได้เป็น

$$\phi(n) = (p-1)(q-1) \tag{2.1}$$

เลือกตัวเลข $1<e<\phi(n)$ ได้เป็น

$$\gcd(e, \phi(n)) = 1 \tag{2.2}$$

คำนวณค่า d จาก $d = e^{-1} \bmod \phi(n)$ (2.3)

โดยที่ e เป็นกุญแจสาธารณะและ d เป็นกุญแจส่วนตัว โดยทั่วไปจะเลือกค่า e น้อย ๆ เช่น $e = 2^{16} + 1$ ส่วนค่าของ d ,p และ q ต้องเป็นความลับ เพื่อใช้สำหรับคำนวณสมการ $C = M^e \pmod n$ เมื่อ M คือข้อมูลดิบซึ่งเป็นค่า $0 \leq M < n$ ค่าของ C เป็นค่าที่เข้ารหัสลับแล้ว และสามารถถอดรหัสลับได้จากสมการ $M = C^d \pmod n$

ขั้นตอนการเข้าและถอดรหัสลับ

1. สุ่มเลข จำนวนเฉพาะมา 2 ตัวในที่นี้กำหนดให้เป็น p,q
2. $n = p*q$
3. เลือกเลข จำนวนเฉพาะมา 1 ตัวคือ d ซึ่งต้องเป็นไปตามเงื่อนไขนี้ $1<d<\phi(n)$
4. หาค่าผกผันของ d คือ $ed \bmod \phi(n) = 1$

การเข้ารหัสลับสามารถทำได้โดย นำข้อมูลที่ต้องการเข้ารหัสลับเช่นต้องการเข้ารหัสลับคำว่า RENA โดยกำหนดให้ A=00,B=01,C=02,…,Z=25 จากนั้นทำตามขั้นตอนทั้ง 4 ข้างต้น คือ p = 53 และ q = 61 ได้ n = 3233 และ $\phi(n) = 3120$ เลือก d = 791ได้ e = 71 นำค่าต่าง ๆ มาเข้ารหัสลับด้วยสมการ $C = M^e \bmod n$ ได้ RE=1704 และ NA=1300 จากนั้นแทนในสมการได้ $1704^{71} \bmod 3233 = 3106$ และค่า $1300^{71} \bmod 3233 = 0100$ นั้นหมายความว่าข้อมูลที่เข้ารหัสลับแล้วคือ 3106 0100 และในการถอดรหัสลับใช้สมการ $M = C^d \bmod n$ ได้ $3106^{791} \bmod 3233 = 1704$ ซึ่งเป็นข้อมูลที่ถูกต้อง

ในการนำไปใช้กับกุญแจอิเล็กทรอนิกส์ไม่สามารถนำวิธีนี้ไปใช้ได้โดยตรง เนื่องจากจำนวนของตัวเลขที่คำนวณมีค่ามากเช่น $21^{31} = 9.74536560714604461109210780048 87e+40$ สังเกตว่าตัวเลขมากเกินกว่าที่ไมโครคอนโทรลเลอร์จะคำนวณได้ ดังนั้นจึงดำเนินการศึกษาสมการรูปแบบเลขชี้กำลังมอดุลาร์

2.4 การลดทอนสมการรูปแบบ RSA

การคำนวณสมการการเข้ารหัสลับแบบ RSA โดยทั่วไปจะถูกเรียกว่าสมการเลขชี้กำลังมอดุลาร์ ซึ่งประกอบด้วยส่วนชี้กำลังคือ M^e และส่วนที่เป็นมอดูโลคือ $C := (M^e) \bmod n$ ผลการคำนวณในแต่ละขั้นตอนที่ทำการหาผลลัพธ์จำเป็นต้องลดลง ในการคำนวณวิธีการต่าง ๆ ซึ่งจะไม่ยกกำลังสมการโดยตรง สามารถทำได้ด้วยวิธีการดังนี้

2.4.1 วิธีการทวินิยม (Binary Method)

วิธีการนี้จะสแกนบิตของเลขชี้กำลัง จากซ้ายไปขวาหรือขวาไปซ้ายโดยการคำนวณแต่ละขั้นตอนจะขึ้นอยู่กับค่าของบิตที่สแกนได้โดยกำหนดให้ k เป็นจำนวนบิตของ e เช่น $k = 1+[\log_2 e]$ และสามารถเขียนค่า e ได้ใหม่ดังนี้

$$e = (e_{k-1}e_{k-2}\ldots e_1e_0) = \sum_{i=0}^{k-1} e_i\, 2^i \qquad (2.4)$$

เมื่อ k คือจำนวนบิตของ e

ดังนั้น $e_i \in \{0,1\}$ วิธีการทวินิยมสามารถคำนวณได้จาก

The Binary Method
Input: M,e,n Output: $C=M^e \pmod n$ 1. if $e_{k-1} = 1$ then C := M else C: = 1 2. for i = k -2 down to 0 2a. C:= C*C (mod n) 2b. if $e_i = 1$ then C:=C*M (mod n) 3. return C

ตัวอย่างเช่น ให้ e = 250= (11111010) ซึ่งได้ k = 8 โดยเริ่มให้ C := M เมื่อ $e_{k-1}= e_7 = 1$ จากนั้นทำตามขั้นตอนวิธีการทวินิยมซึ่งจะได้ผลลัพธ์ดังตารางประกอบ 2-1

ตารางประกอบ 2-1 ตัวอย่างการคำนวณของวิธีการทวินิยม

i	e_i	Step 2a	Step 2b
6	1	$(M^2) = M^2$	$M^2*M = M^3$
5	1	$(M^3)^2=M^6$	$M^6*M = M^7$
4	1	$(M^7)^2 = M^{14}$	$M^{14}*M = M^{15}$
3	1	$(M^{15})^2 = M^{30}$	$M^{30}*M = M^{31}$
2	0	$(M^{31})^2 = M^{62}$	M^{62}
1	1	$(M^{62})^2 = M^{124}$	$M^{124}*M = M^{125}$
0	0	$(M^{125})^2 = M^{250}$	M^{250}

จากตารางประกอบ 2-1 จะเห็นได้ว่าจำนวนครั้งในการคูณกันระหว่าง C*C และ C*M ที่เกิดขึ้นจากการคำนวณโดยใช้วิธีการทวินิยมค่าของ M^{250} พบว่าจะเท่ากับ 7+5 = 12 ครั้งซึ่งพิจารณาจากขั้นตอน 2a และขั้นตอน 2b เมื่อกำหนดให้ $e_{k-1} = 1$ เป็นดังนี้

Maximum:(k-1) + (k-1) = 2(k-1) = 14 ครั้ง (2.5)

Minimum:(k-1) + 0= k-1= 7 ครั้ง (2.6)

Average:(k-1)+(k-1) $= \frac{3}{2}$ (k-1) = 11.5 ครั้ง (2.7)

2.4.2 วิธีการ m-ary

วิธีการ m-ary เป็นการปรับปรุงวิธีการทวินิยม โดยแบ่งการสแกนบิตของ e ได้ดังนี้

- สแกน 2 บิต เรียกว่า วิธีจตุนิยม (Quaternary)
- สแกน 3 บิต เรียกว่า วิธีอัฐนิยม (Octal)

ในการแบ่งบิตออกเป็นกลุ่มโดยแทน s ด้วยบล็อกของ r เช่น s*r = k ถ้า r ไม่ใช่ตัวหารของ k ซึ่งสามารถเขียนได้ใหม่ดังนี้

$$F_i = (e_{ir+r-1}e_{ir+r-2}\ldots e_{ir}) = \sum_{j=0}^{r-1} e_{ir+j}2^i \quad (2.8)$$

เมื่อ $0 \le Fi \le m-1$ และ $e = \sum_{i=0}^{s-1} F_i 2^{ir}$ โดยที่วิธีการ m-ary จะคำนวณค่าของ M^w (mod n) เป็นอันดับแรกเมื่อ w = 2,3,…,m-1 และค่าของ Fi จะไม่เป็นศูนย์

The m-ary Method
Input: M,e,n Output: C = M^e mod n 1. Computer and store M^w (mod n) for all w= 2,3,4,…m-1 2. Decompose e into r-bit words F_i for i = 0,1,2,…s-1 3. C := M^{Fs-1} (mod n) 4. for i = s-2 downto 0 4a. C :=C^{2r} (mod n) 4b. if $F_i \neq 0$ then C := C*M^{Fi} (mod n) 5. return C

2.4.2.1 วิธีการจตุนิยม

วิธีการนี้จะกำหนดให้ค่า e เป็นกลุ่มในการ Scan ครั้งละ 2 บิตโดยกำหนดให้ (00) = 0, (01)=1, (10)=2 และ (11)=3 ใช้สำหรับ Step 4b ที่อาจต้องการค่า M^0,M^1,M^2 และ M^3 ดังตัวอย่าง เมื่อให้ e = 250 และแบ่งกลุ่มบิต กลุ่มละ 2 บิต

e = 250 = 11 11 10 10 (2.9)

เมื่อ s = 4 ซึ่งเป็นจำนวนของกลุ่ม ซึ่งได้มาจาก s = k/r = 8/2 = 4 และสามารถเขียนเป็นตารางประกอบ 2-2 ก่อนการคำนวณจริง ได้ดังนี้

ตารางประกอบ 2-2 การเตรียมค่า M^w ก่อนการคำนวณและแบ่ง e ออกเป็นกลุ่มๆ ละ 2 บิต

bits	w	M^w
00	0	1
01	1	M
10	2	$M*M=M^2$
11	3	$M^2*M=M^3$

วิธีจตุนิยมกำหนดให้ $C := M^{F3} = M^3 \pmod n$ และสามารถประมวลผลตามขั้นตอนวิธีการของ m-ary ได้ดังนี้

ตารางประกอบ 2-3 ตัวอย่างการคำนวณวิธีการจตุนิยม

i	Fi	Step 4a	Step 4b
2	11	$(M^3)^4 = M^{12}$	$M^{12}*M^3=M^{15}$
1	10	$(M^{15})^4=M^{60}$	$M^{60}*M^2=M^{62}$
0	10	$(M^{62})^4=M^{248}$	$M^{248}*M^2=M^{250}$

จากตารางประกอบ 2-3 สังเกตได้ว่าจำนวนการคูณแบบมอดุลาร์ต้องการเท่ากับ 11 ครั้ง โดยได้มาจากการเตรียมการแบ่งกลุ่ม 2 ครั้ง + ความยาวของ e มีหน่วยเป็นบิต + ขั้นตอนใน Step 4b จะได้ 2+6+3 = 11 ครั้ง

2.4.2.2 วิธีการอัฐนิยม

วิธีการนี้แบ่งกลุ่มของ e ออกเป็นกลุ่มละ 3 บิตดังตัวอย่างเช่น e = 250 จะได้

$$e = 250 = \underline{011}\ \underline{111}\ \underline{010} \tag{2.10}$$

โดยกำหนดให้ s = k/r = 9/3 = 3 และก่อนการคำนวณจริงต้องสร้างตาราง ก่อนการคำนวณจริง ในขั้นตอนต่างๆ ของค่า $M^w \pmod n$ สำหรับค่า w = 2, 3, 4, 5, 6, 7 ดังนี้

ตารางประกอบ 2-4 การเตรียมค่า M^w ก่อนคำนวณและแบ่ง e ออกเป็นกลุ่มละ 3 บิต

Bits	w	M^w
000	0	1
001	1	M
010	2	$M*M=M^2$
011	3	$M^2*M=M^3$
100	4	$M^3*M=M^4$
101	5	$M^4*M=M^5$
110	6	$M^5*M=M^6$
110	7	$M^6*M=M^7$

ก่อนการคำนวณจริงต้องสร้างตารางประกอบ 2-4 เพื่อให้ทราบถึงค่ายกกำลังของ M จากนั้นกำหนดให้ $C := M^{F2} = M^3 \pmod n$ และสามารถคำนวณค่า $M^{250} \pmod n$ ได้ดังนี้

ตารางประกอบ 2-5 ตัวอย่างการคำนวณวิธีการอัฐนิยม

i	Fi	Step 4a	Step 4b
1	111	$(M^3)^8=M^{24}$	$M^{24}*M^7=M^{31}$
0	010	$(M^3)^8=M^{248}$	$M^{248}*M^2=M^{250}$

จากตารางประกอบ 2-5 สังเกตได้ว่าการคำนวณ $M^{250} \pmod n$ โดยใช้วิธีการของอัฐนิยม ต้องการการคำนวณจำนวนทั้งหมด 6+6+2 = 14 ครั้ง อย่างไรก็ตามเมื่อมองไปที่การคำควณ $M^w \pmod n$ ของค่า w ทั้งหมด w= 2, 3, 4, 5, 6, 7 ไม่ได้นำค่าของ w ทั้งหมดมาใช้ในการประมวลผลจริง ดังนั้นจึงสามารถปรับแต่งขั้นตอนที่ 1 ของวิธีการ m-ary และก่อนการคำนวณ $M^w \pmod n$ เฉพาะค่า w ที่นำมาคำนวณค่า e ตัวอย่างเช่น e = 250 สามารถคำนวณกำลังโดยใช้การคูณเพียง 4 ครั้ง

ตารางประกอบ 2-6 การเตรียมค่า M^w ก่อนคำนวณ หลังจากปรับปรุง

bits	w	M^w
000	0	1
001	1	M
010	2	$M*M=M^2$
011	3	$M^2*M=M^3$
100	4	$M^3*M=M^4$
111	7	$M^4*M^3=M^7$

จากตารางประกอบ 2-6 เป็นตารางที่ปรับปรุงการคำนวณค่าของ M^w โดยจะคำนวณเฉพาะค่า M^w ที่จำเป็นต้องใช้งานเท่านั้น ซึ่งจะลดการคำนวณได้ 2 ครั้งเมื่อเทียบกับการคำนวณค่า M^w ทั้งหมดโดยพิจารณาจาก 4+6+2 = 12 ครั้ง และเมื่อพิจารณาต่อจะพบว่าถ้า k มีค่ามาก ๆ ค่าเฉลี่ยของการคูณ และวิธีตีตารางโดยวิธีการ m-ary จะได้เป็นดังนี้

กำหนดให้ $2^r = m$ และ $\frac{k}{r}$ เป็นจำนวนเต็ม

- Preprocessing Multiplication (Step 1): $m - 2 = 2^r - 2 = 2$ เมื่อ r=2 (2.11)
- Squarings (Step 4a): $(\frac{k}{r} - 1) * r = k - r = 6$ (2.12)
- Multiplications (Step 4b): $(\frac{k}{r} - 1)(1 - \frac{1}{m}) = (\frac{k}{r} - 1)(1 - 2^{-r}) = 2.25$ (2.13)

ดังนั้นโดยทั่วไปวิธีการ m-ary ต้องการ: $2^r - 2 + k - r + (\frac{k}{r} - 1)(1 - 2^{-r}) = 10$ (2.14)

ค่าเฉลี่ยของการตีตาราง(ขั้นตอน 4a) ในวิธีการทวินิยมสามารถหาโดยการแทนค่า r = 1 และ m = 2 ซึ่งจะได้ $\frac{3}{2}(k-1)$ ฉะนั้นจึงกำหนดให้ r = r* สำหรับแต่ละค่า k ซึ่งเป็นค่าเฉลี่ยในการหา ค่าการคูณโดยวิธีการของ m-ary ที่น้อยที่สุด ตามตารางประกอบ 2-7 ซึ่งเป็นค่าเฉลี่ยของ การตีตาราง ของวิธีการทวินิยมและวิธีการ m-ary ด้วยการแทนค่า r ที่ดีที่สุด

ตารางประกอบ 2-7 การเปรียบเทียบวิธีการทวินิยมกับวิธีการ m-ary

k	binary	m-ary	r*	Savings %
8	11	10	2	9.1
16	23	21	2	8.6
32	47	43	2,3	8.5
64	95	85	3	10.5
128	191	167	3,4	12.6
256	383	325	4	15.1
512	767	635	5	17.2
1024	1535	1246	5	18.8
2048	3071	2439	6	20.6

จากตารางประกอบ 2-7 จะสังเกตได้ว่าแนวโน้มของค่าเปอร์เซ็นต์การลดทอนซึ่งใช้วิธี m-ary เปรียบเทียบกับวิธีการทวินิยมคือ 33% ซึ่งพิสูจน์ได้จาก

$$\lim_{k\to\infty} \frac{2^r - 2 + k - r + (\frac{k}{r} - 1)(1 - 2^{-r})}{\frac{3}{2}(k-1)} = \frac{2}{3}(1 + \frac{1 - 2^{-r}}{r}) \approx \frac{2}{3} \qquad (2.15)$$

2.4.3 วิธีการแบบปรับแต่ง m-ary

จากขั้นตอนการลดทอนของด้วยวิธีการของ m-ary ที่ผ่านมาพบว่าต้องเตรียมการก่อนการคำนวณจริงทุกๆ ค่า ทำให้เกิดการสูญเสียในบางค่า M^w ที่ไม่ได้ใช้งานจริง จึงจำเป็นต้องปรับแต่ง m-ary เพื่อจะคำนวณเฉพาะค่า M^w ที่จำเป็นต้องใช้งานเท่านั้น อย่างไรก็ตามในการแบ่งค่า e ออกเป็นส่วนๆ สามารถพิจารณาได้ 2 แบบคือ แบบศูนย์และไม่ใช่ศูนย์ซึ่งใช้การลดทอนในขั้นตอนที่ 4b ของวิธีการ m-ary ซึ่งในขั้นตอนต่อไปจะแสดงให้เห็นถึงวิธีการคูณและวิธีการตีตาราง

2.4.3.1 การลดทอนการเตรียมการ Multiplications

วิธีนี้เป็นการลดจำนวนการเตรียมการก่อนการคำนวณจริง โดยการแบ่งกลุ่มของ k ออกเป็นกลุ่มๆ ในที่นี้จะแบ่งออกเป็น 4 บิตต่อกลุ่มนั้นคือ เมื่อ k= 16, d=4 สามารถเขียนได้เป็น 1011 0011 0111 1000 ซึ่งถือได้ว่าจะเตรียมการเฉพาะ w = 3,7,8,11 ซึ่งเป็นค่าที่จำเป็นต่อการใช้งานสามารถคำนวณได้ดังนี้

$$M^2 = M * M$$
$$M^3 = M^2 * M$$
$$M^4 = M^2 * M^2$$
$$M^7 = M^3 * M^4$$
$$M^8 = M^4 * M^4$$
$$M^{11} = M^8 * M^3$$

สังเกตได้ว่าทำการคำนวณเพียง 6 ครั้ง แต่วิธีการ m-ary จะไม่คำนึงถึงความต้องการค่า ยกกำลังและจะคำนวณทุกๆ ค่าของ w เมื่อพิจารณาเพิ่มขึ้นจะพบว่าจำนวนของค่าที่สามารถประหยัดได้เป็น $m-2 = 2^d - 2$ ซึ่งกรณีนี้จะเกิดขึ้นเมื่อค่าของยกกำลังเป็น 1 คือ 0001 0001 0001 0001

2.4.3.2 Constant Length Nonzero Windows (CLNW)

การแบ่งกลุ่มในลักษณะนี้จะใช้หลักการเทคนิคพื้นฐานของสไลค์วินโดวส์ กล่าวคือจะทำการแยก e ในส่วนที่เป็น 0 และ 1 (วินโดวส์) Fi ของความยาว L(Fi) โดยที่ค่าสมาชิกในกลุ่มจะกำหนดเป็นค่าคงที่และสามารถหาการเตรียมการก่อนคำนวณเป็นจำนวนที่น้อยที่สุด ซึ่งในการแบ่งกลุ่มจะให้ความสนใจที่ บิต 1 เป็นหลัก ดังอัลกอริทึมนี้

The Sliding Windows Method

Input : M,e,n.

Output : $C = M^e \pmod n$.

1. Compute and store Mw (mod n) for all w= 3,5,7,…,2d – 1.
2. Decompose e into zero and nonzero windows Fi length L(Fi) for i = 0,1,2,…p-1.
3. C := $M^{F_{k-1}} \pmod n$.
4. for I = p-2 downto 0
   4a. $C := C^{2^{L(Fi)}} \pmod n$
   4b. if $Fi \neq 0$ then $C := C*M^{Fi} \pmod n$
5. return C

การแบ่งค่าของ e ออกเป็นส่วนๆ โดยพิจารณาถึงความเป็นไปได้ของค่า e ที่เตรียมการก่อนคำนวณน้อยที่สุดเช่น e = 111 00 101 0 001 จะเตรียมเฉพาะค่าที่ค่าของ w = 3,5,7 ตามตารางประกอบ 2- 8

ตารางประกอบ 2-8 การเตรียมการคำนวณของ CLNW

Bits	w	M^w
001	1	M
010	2	$M*M = M^2$
011	3	$M*M^2 = M^3$
111	7	$M^5*M^2 = M^7$

เมื่อเตรียมการเรียบร้อยแล้ว ค่าที่ได้จะนำไปประมวลผลต่อไปเช่น M^{3665} (mod n) สามารถคำนวณได้ดังตารางประกอบ 2-9 จะเห็นได้ว่าการเตรียมการก่อนคำนวณ = 4+9+2=15 ครั้ง

ตารางประกอบ 2-9 ตัวอย่างการคำนวณแบบ CLNW

i	Fi	L(Fi)	Step 4a	Step 4b
3	00	2	$(M^7)^4 = M^{28}$	M^{28}
2	101	3	$(M^{28})^8 = M^{224}$	$M^{224}*M^5 = M^{229}$
1	0	1	$(M^{229})^2 = M^{458}$	M^{458}
0	001	3	$(M^{458})^8 = M^{3664}$	$M^{3664} * M = M^{3665}$

2.4.4 อัลกอริทึมการลดทอนแบบ Montgomery

อัลกอริทึมการลดทอนแบบ Montgomery นี้ถูกค้นคิดพัฒนาเพื่อให้เหมาะกับไมโครโปรเซสเซอร์ ซึ่งจะมีความสามารถประมวลผลทางคณิตศาสตร์มอดุโลกำลัง 2 ได้รวดเร็ว ดังนั้นการประมวลผลของการลดทอนแบบ Montgomery จะปราศจากการนำค่ามามอดุโล n โดยตรง แต่จะทำการแปลง n ให้อยู่ในรูปของ 2 ยกกำลัง n นั้นคือ r กำหนดให้ $2^{k-1} \le n < 2^k$ เมื่อ r เป็น 2^k โดยที่ r และ n จะต้องมีความสัมพันธ์กับจำนวนเฉพาะเช่น $\gcd(2^k,n)=1$ แนวคิดของการลดทอนแบบ Montgomery จะอาศัยพื้นฐานดังนี้

ให้จำนวนเต็มบวก $a < n$ จะได้

$$\bar{a} = a \cdot r \bmod n \tag{2.16}$$

เขียนใหม่ได้ดังนี้

$$i \cdot r \bmod n \,|_{0 \le i \le n-1} \tag{2.17}$$

ในฟังก์ชั้นการคูณแบบ Montgomery (Montgomery product) ต้องการค่า $\bar{a}$ และ $\bar{b}$

$$\bar{R} = \bar{a} \cdot \bar{b} \cdot r^{-1} \bmod n \tag{2.18}$$

ค่า r^{-1} เป็นค่าส่วนกลับของ r modulo n เช่น

$$r^{-1} \cdot r = 1 \bmod n \tag{2.19}$$

ผลของจำนวนจริงเขียนได้ดังนี้

$$R = a \cdot b \bmod n \tag{2.20}$$

เมื่อ $\bar{R} = \bar{a} \cdot \bar{b} \cdot r^{-1} \bmod n$

$$= a \cdot b \cdot r \cdot r^{-1} \bmod n$$

$$= a \cdot b \cdot r \bmod n \tag{2.21}$$

การลดทอนแบบ Montgomery ต้องการค่า n' ซึ่งเป็นจำนวนจริงจากสมการ

$$r \cdot r^{-1} - n \cdot n' = 1 \tag{2.22}$$

จำนวนจริง r^{-1} และ n' ทั้งสองค่านี้สามารถหาได้จาก อัลกอริทึมส่วนขยายแบบยุคลิด (extended Euclidean) ซึ่งอัลกอริทึมการคูณแบบ Montgomery คำนวณหา u = $\bar{a} \cdot \bar{b} \cdot r^{-1} (\bmod n)$ ได้ดังนี้

Function MonPro($\bar{a}, \bar{b}$)
Step 1. t: = $\bar{a}, \bar{b}$
Step 2. m: = t * n' mod r
Step 3. u: = (t + m*n)/r
Step 4. if u $\geq$ n then return u-n else return u

ลักษณะที่สำคัญที่สุดของอัลกอริทึมการคูณแบบ Montgomery คือการประมวลผลโดย มอดุโล r และหาร r ซึ่งทั้ง 2 การประมวลผลนี้ สามารถทำงานได้รวดเร็วเนื่องจาก r คือ 2^k ซึ่งจะสังเกตได้ว่าวิธีการที่กล่าวมาข้างต้น จะปรับปรุงวิธีการคำนวณโดยการลดเตรียมการก่อนการคำนวณจริงแต่วิธีการของการลดทอนแบบ Montgomery จะเน้นไปที่การมอดุโลที่ r เป็นหลัก เนื่องจากแต่ละรอบของการคำนวณจะต้องผ่านฟังก์ชัน MonPro ทำให้หากมอดุโล r และหาร r ได้จะทำให้การประมวลผลเร็วขึ้นตามไปด้วย

ในการคำนวณสมการเข้ารหัสลับแบบ RSA นอกจากต้องใช้ฟังก์ชันการคูณแบบ Montgomery ซึ่งเขียนได้เป็น MonPro($\bar{a},\bar{b}$) แล้วยังจำเป็นต้องใช้ฟังก์ชันเลขชี้กำลังมอดุลาร์ เขียนได้เป็น ModExp(M,e,n) ซึ่งเป็นฟังก์ชันหลักในการเข้ารหัสลับ โดยฟังก์ชั้นนี้จะใช้หลักการพื้นฐานของวิธีการทวินิยมและ n ต้องเป็นจำนวนคี่เท่านั้น ซึ่ง n คือ p * q

Function ModExp(M,e,n) {n is odd}
Step 1. Compute n' using Euclid's algorithm Step 2. $\overline{M}$ = M*r mod n Step 3. $\overline{C}$ = r mod n Step 4. for i = k-1 down to o do Step 4a. $\overline{C}$ = MonPro($\overline{C}$, $\overline{C}$) Step 4b. if e_i=1 then $\overline{C}$ = MonPro($\overline{M}$, $\overline{C}$) Step 5. C = MonPro($\overline{C}$,1) Step 6. return C

จากฟังก์ชัน ModExp(M,e,n) จะเริ่มคำนวณจากการหาค่า $\overline{M}$ โดยใช้ตัวหารคล้าย ๆ กับขั้นตอนที่ 2 และ 3 ของฟังก์ชัน MonPro($\bar{a},\bar{b}$) และไม่จำเป็นต้องเตรียมการก่อนการคำนวณจริง เนื่องจากมันถูกคำนวณในลูปการคูณของ Montgomery ซึ่งก็คือมอดุโล 2^k และหาร 2^k

จากหลักการที่กล่าวมาข้างต้นสามารถเขียนขั้นตอนการหาค่า $C = M^e \bmod n$ ได้ดังนี้

1. $r = 2^k$ โดยหา k ได้จาก

$2^{k-1} \le n < 2^k$ เมื่อ k คือจำนวนบิตของ n

n คือ p x q

2. นำค่า r และ n หา gcd(r,n) = 1 โดยมีอัลกอริทึมในการหาดังนี้

GCD Algorithm
INPUT : two positive integers x and y with $x \ge y$ OUTPUT: gcd (x,y) 1. g = 1 2. While both x and y are even do the following : x=x/2 , y=y/2 , g=2g 3.While $x \neq 0$ do the following:

While x is even do : x=x/2 While y is even do : y =y/2 T= | x-y| /2 If x $\geq$ y then x=t; otherwise, y=t 4. Return(g*y)

3. ใช้อัลกอริทึม Extended Euclid หาค่า r^{-1} และ n' จากสมการ $r*r^{-1} - n*n' = 1$

Binary extended gcd algorithm
1. g = 1 2. While x and y are both even ,do the following : x=x/2 , y=y/2 , g=2g 3. u=x , v=y , A=1 , B=0 , C=0 , D=1 4. While u is even do the following: 4.1 u = u / 2 4.2. if A=B=0 (mod 2) then A=A/2,B=B/2 ; otherwise, A=(A+y) / 2 , B=(B-x)/2 5. While v is even do the following 5.1 v = v/2 5.2. if C=D=0 (mod 2) then C=C/2 , D=D/2 ; otherwise,C=(C+y)/2,D=(D-x)/2 6. If u > v then u = u – v , A= A-C ,B=B-D; otherwise, v = v–u , C = C- A ,D=D-B 7. If u = 0, then a = C, b=D and return (a,b,g*v); otherwise, go to step 4

4. ใช้ฟังก์ชัน MonPro($\bar{a},\bar{b}$) และฟังก์ชัน MonExp(M,e,n) เพื่อหาค่า C จากสมการ $C = M^e \bmod n$ โดยที่ e ต้องแปลงเป็นฐานสองซึ่งใช้วิธีการของบนพื้นฐานของวิธีการทวินิยม

<u>ตัวอย่าง</u> การคำนวณหาค่า $C = 7^{10} \bmod 13$ โดยใช้อัลกอริทึมการลดทอนแบบ Montgomery

วิธีทำ M = 7,e= 10 ,n=13

step 1:ได้ $r = 2^k = 16$ ใช้ Euclid's algorithm หาค่า r-1 และ n' ได้$16*9 - 13*11 = 1$นั้นคือ r-1 = 9 และ n' = 11

step 2: หา $\overline{M}$ จาก $\overline{M}$ =M*r mod n = 7*16 mod 13 = 8

step 3: หา $\overline{C}$ จาก $\overline{C}$ = r mod n = 16 mod 13 = 3 ดังนั้นได้ $\overline{C}$ = 3 และ $\overline{M}$ = 8

e_i	Step 4a	Step 4b
1 0 1 0	MonPro(3,3) =3 MonPro(8,8)=4 MonPro(4,4)=1 MonPro(7,7)=12	MonPro(8,3)=8 MonPro(8,1)=7

การคำนวณหาค่า MonPro(3,3)

t = 3*3 = 9

m = 9*11 mod 16 = 3

u =(9+3*13)/16 = 48/16=3

Step 5 : C= MonPro(12,1) = 4

Step 6 : return 4

จากตัวอย่างข้างต้นหากคำนวณค่าโดยไม่ใช้ทฤษฎีการลดทอนแบบ Montgomery จะได้ 7^{10} = 282 475 249 จะเห็นได้ว่าตัวเลขจะมากขึ้นเป็นทวีคูณซึ่งเป็นผลมาจากการยกกำลังตัวเลขและหากเลขยกกำลังมีค่ามาก ผลลัพธ์ของการยกกำลังจะมากทวีคูณตามไปด้วย ซึ่งเมื่อใช้การลดทอนแบบ Montgomery พบว่าค่าที่มากที่สุดจะเกิดขึ้นในขั้นตอนที่ 2 และขั้นตอนที่ 3 ของฟังก์ชัน Monpro = 99 ซึ่งจะเห็นได้ว่าตัวเลขที่เกิดจากการคำนวณแตกต่างกันประมาณ 100,000 เท่า เมื่อใช้การลดทอนแบบ Montgomery เข้าช่วยตัวเลขที่ยกกำลังมาก ๆ จะทำให้การวนทำซ้ำของฟังก์ชัน Monpro มีรอบทำงานหลายรอบ (เท่ากับค่าของ e) แต่ไม่มีผลให้ตัวเลขที่เกิดจากการทำซ้ำมากขึ้น ซึ่งมีความเป็นไปได้ที่การเข้ารหัสลับแบบ RSA สามารถทำได้บนไมโครคอนโทรลเลอร์เนื่องจากไมโครคอนโทรลเลอร์ไม่สามารถคำนวณตัวเลขที่มีขนาดใหญ่ได้

2.5 สรุปท้ายบท

ในการสร้างกุญแจอิเล็กทรอนิกส์ที่ติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ต USB นั้นมีขั้นตอนที่ซับซ้อนพอสมควร รวมถึงการเข้ารหัสลับแบบ RSA มีลักษณะที่เด่นคือกุญแจที่ใช้ในการเข้าและถอดรหัสลับ แตกต่างกันและในการใช้ทฤษฎีการลดทอนแบบ Montgomery จะมีกระบวนการขั้นตอนที่ซับซ้อนและต้องเป็นไปตามลำดับ โดยผ่านขั้นตอนดังนี้

1. หาค่า r
2. หาค่า gcd(r,n)
3. หาค่า n’ จาก rr-1 – nn’ = 1
4. หาค่า Monpro(a,b) และ Monexp(M,e,n)

โดยแต่ละขั้นตอนจะไม่ทำให้ตัวเลขเพิ่มขึ้นเป็นทวีคูณ และไม่จำเป็นต้องเตรียมการก่อนการคำนวณจริงต่างกับการใช้วิธีการแบบอื่น ๆ