การเข้ารหัสลับบนบัตรเลือกตั้งเพื่อป้องกันการปลอมแปลงและทุจริต

นายศิริพงษ์ ประยูรหงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2544
ISBN 974-03-0470-2

ENCRYPTION ON BALLOT PAPERS FOR FORGERY AND DISHONESTY PREVENTION

MR. SIRIPHONG PRAYOONHONG

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2001

ISBN 974-03-0470-2

หัวข้อวิทยานิพนธ์	การเข้ารหัสลับบนบัตรเลือกตั้งเพื่อป้องกันการปลอมแปลงและทุจริต
โดย	นายศิริพงษ์ ประยูรหงษ์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	อาจารย์สุวิทย์ นาคพีระยุทธ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประสิทธิ์ ประพิณมงคลการ)

..... อาจารย์ที่ปรึกษา
(อาจารย์สุวิทย์ นาคพีระยุทธ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ลัญฉกร วุฒิสิทธิกุลกิจ)

ศิริพงษ์ ประยูรหงษ์ : การเข้ารหัสลับบนบัตรเลือกตั้งเพื่อป้องกันการปลอมแปลงและทุจริต. (ENCRYPTION ON BALLOT PAPERS FOR FORGERY AND DISHONESTY PREVENTION) อ. ที่ปรึกษา : อาจารย์สุวิทย์ นาคพีระยุทธ 85 หน้า. ISBN 974-03-0470-2.

วิทยานิพนธ์นี้นำเสนอการเข้ารหัสลับบนบัตรเลือกตั้งเพื่อป้องกันการปลอมแปลงและทุจริต เนื่องจากบัตรเลือกตั้งในปัจจุบันสามารถปลอมแปลงได้ง่ายแต่การตรวจสอบว่าเป็นบัตรปลอมหรือไม่นั้นทำได้ยาก

บัตรเลือกตั้งแต่ละใบจะมีรหัสลับซึ่งได้จากการเข้ารหัสลับแบบกุญแจสาธารณะด้วยวิธี RSA ทำให้ไม่สามารถสร้างรหัสลับปลอมได้ มีการดึงคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตรเลือกตั้งมาเป็นส่วนหนึ่งของรหัสลับ เพื่อป้องกันการคัดลอกรหัสลับจากบัตรเลือกตั้งจริงไปยังบัตรเลือกตั้งปลอมซึ่งมีคุณลักษณะเฉพาะตัวของกระดาษแตกต่างออกไป มีข้อมูลที่มาของบัตรทำให้สามารถตรวจสอบที่มาของบัตรได้ มีระบบต้นขั้วเพื่อตรวจสอบจำนวนบัตรที่ใช้ไป มีลายพิมพ์นิ้วมือของผู้มาใช้สิทธิ์ มีรหัสแท่งของรหัสลับเพื่อให้สะดวกในการสุ่มตรวจสอบบัตรเลือกตั้ง

กระดาษที่ใช้พิมพ์บัตรเลือกตั้งจะมีวัตถุลักษณะเป็นเส้นฝังอยู่ ซึ่งพิกัดของจุดปลายของวัตถุจะถูกใช้เป็นคุณลักษณะเฉพาะตัวของกระดาษ การสร้างกุญแจ การพิมพ์รหัสลับบนบัตรเลือกตั้งและการตรวจสอบบัตรเลือกตั้งที่เขตเลือกตั้งซึ่งเป็นที่นับคะแนน จะใช้โปรแกรมที่พัฒนาขึ้นบนไมโครคอมพิวเตอร์ ส่วนการตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้ง จะใช้เครื่องถอดรหัสลับที่พัฒนาขึ้นโดยใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 ซึ่งสามารถทำงานได้เร็วเพียงพอที่จะนำไปใช้งานจริง

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา 2544 .....

4170543121 : MAJOR ELECTRICAL ENGINEERING

KEY WORD: BALLOT PAPER / PUBLIC-KEY CRYPTOGRAPHY / PUBLIC KEY / PRIVATE KEY / FEATURE EXTRACTION / 8051 MICROCONTROLLER

SIRIPHONG PRAYOONHONG : ENCRYPTION ON BALLOT PAPERS FOR FORGERY AND DISHONESTY PREVENTION. THESIS ADVISOR : SUVIT NAKPEERAYUTH, 85 pp. ISBN 974-03-0470-2.

This thesis proposes encryption on ballot papers to prevent forgery and check dishonesty because the current ballot papers can be easily forged but difficult to prove if they are forged.

Each ballot paper has a secret code generated by public-key encryption using RSA, therefore the code cannot be forged. Feature data extracted from that piece of paper which is used as the ballot paper is included in the secret code to prevent copying the secret code from the real one to the forged one which has different feature data. Stubs are used to check the number of ballot papers used in an election. Fingerprints of voters are required. A barcode of the secret code is used to check the samples of ballot papers conveniently.

Some small line-shaped objects are embedded in paper used to make ballot papers. Coordinates of these objects are used as paper features. Key generation, secret code printing on ballot papers and ballot paper check at the voting district which is the vote-counting place can be done using the program developed on a microcomputer. Ballot paper check at the voting place can be done using a decryption device based on a 8051 microcontroller which can run fast enough for practical use.

Department Electrical Engineering Student's signature .....

Field of study Electrical Engineering Advisor's signature .....

Academic year 2001 .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ อาจารย์สุวิทย์ นาคพีระยุทธ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำ ข้อคิดเห็น และสนับสนุนอุปกรณ์เครื่องมือต่างๆ ในการทำวิจัยมาโดยตลอด ผู้วิจัยจึงขอกราบขอบพระคุณมา ณ ที่นี้

ขอขอบคุณ เพื่อนพี่น้องนิสิตที่อยู่ภายในห้องปฏิบัติการไฟฟ้าสื่อสาร ที่ได้ช่วยเหลือและเป็นกำลังใจที่ดียิ่งต่อผู้วิจัย

ท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดามารดา ที่ให้การสนับสนุนแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สารบัญ

หน้า

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

บทที่ หน้า

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

จากเหตุการณ์การทุจริตเลือกตั้งที่ผ่านมา ทำให้เราจำเป็นต้องกลับมาทบทวนกระบวนการเลือกตั้งกันใหม่ว่า จะสามารถปรับปรุงวิธีดำเนินการเลือกตั้งได้อย่างไรบ้าง กระบวนการเลือกตั้งมีเป้าหมายหลักอยู่สองประการ คือ

- ให้ประชาชนสามารถใช้สิทธิ์ออกเสียงได้อย่างเป็นความลับ เพื่อป้องกันการตรวจสอบจากผู้ที่ซื้อเสียงหรือข่มขู่คุกคามได้ นั่นคือ ในระดับบุคคลจะต้องรับประกันได้ว่าไม่มีใครสามารถรู้ได้ว่าใครเลือกใคร และในระดับชุมชนหรือหน่วยเลือกตั้งนั้น ในรัฐธรรมนูญฉบับปัจจุบันได้รับประกันความลับในการใช้สิทธิ์ด้วย โดยระบุให้นำบัตรมานับรวมกันที่อำเภอหรือจังหวัดที่เป็นเขตเลือกตั้ง เพื่อให้ไม่สามารถตรวจสอบได้ว่าหน่วยเลือกตั้งใดได้คะแนนเท่าไร
- แต่ในขณะเดียวกันกระบวนการเลือกตั้งจะต้องสามารถป้องกันการทุจริต โดยสามารถตรวจสอบยืนยันความถูกต้องได้อย่างเปิดเผยในทุกขั้นตอน รวมทั้งสามารถสืบสาวไปยังต้นตอของการกระทำผิดในภายหลังอย่างน้อย 3-6 เดือน หลังการเลือกตั้งไปแล้วได้

เป้าหมายทั้งสองค่อนข้างขัดแย้งกันอยู่ในตัว โดยเฉพาะอย่างยิ่ง การนำบัตรเลือกตั้งมานับรวมกันจะทำให้การตรวจสอบความถูกต้องทำได้ยากขึ้น ปัญหาใหญ่ของระบบการเลือกตั้งด้วยบัตรกระดาษคือ การปลอมแปลงบัตรทำได้โดยง่าย แต่การตรวจสอบว่าเป็นบัตรปลอมหรือไม่นั้นทำได้ยาก ดังนั้นการเลือกตั้งที่ผ่านมาจึงเน้นที่ “ระบบการป้องกัน” ไม่ให้บัตรที่ถูกหย่อนลงในหีบถูกนำออกไปหรือเพิ่มบัตรเข้ามาได้โดยง่าย ด้วยการใช้พยานบุคคลที่ไว้ใจได้ คอยจับตาดูหีบบัตรอยู่ตลอดเวลา แทนที่จะใช้ “ระบบที่สามารถตรวจสอบได้” กระบวนการในปัจจุบันจึงมีจุดอ่อนอยู่มากในการตรวจสอบภายหลังว่า ได้มีการทุจริตที่เล็ดลอดสายตาหรือรู้เห็นเป็นใจกันบ้างหรือไม่ ตัวอย่างที่เห็นได้เด่นชัดคือ หากมีใครคนหนึ่งสามารถหย่อนบัตรที่กาแล้วจำนวนมากลงไปในหีบต่อหน้าต่อตา ก็ไม่มีวิธีการใดที่จะสามารถแยกบัตรที่เพิ่งปนกันนั้นออกจากกันได้ นอกจากยกเลิกบัตรทั้งหีบ และถ้าไม่มีใครเห็นการกระทำก็ยิ่งไม่อาจรู้ได้เลยว่ามีการทุจริตเกิดขึ้นหรือไม่ นอกจากนี้พยานบุคคลจะสามารถจับตาดูได้เพียงเฉพาะวันเลือกตั้งเท่านั้น หลังจากนั้นบัตรในหีบที่เก็บไว้ก็อาจมีการลักลอบเปลี่ยนแปลงได้ไม่ยากนัก ทำให้การฟ้องร้องเพื่อตรวจนับคะแนนใหม่ไม่เกิดประโยชน์มากนัก

สมมติว่าบัตรเลือกตั้งถูกพิมพ์ให้ไม่สามารถปลอมแปลงได้เหมือนกับการพิมพ์ธนบัตร ก็อาจแก้ปัญหาบัตรปลอมได้บางส่วน เนื่องจากไม่สามารถป้องกันการนำบัตรจริงจากเขตอื่นมาใช้ข้ามเขตได้ และจะมีปัญหาเรื่องค่าใช้จ่ายที่สูงมาก รวมทั้งไม่สามารถตรวจสอบหรือสืบสาวไปถึงผู้ทำการทุจริตได้ ดังนั้น วิทยานิพนธ์นี้จึงมีแนวคิดที่จะทำให้บัตรปลอมแปลงได้ยากขึ้นในราคาที่ถูกและสามารถตรวจสอบได้ โดยนำระบบการเข้ารหัสลับมาใช้กับบัตรเลือกตั้ง

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อออกแบบและพัฒนาบัตรเลือกตั้งที่ยากต่อการปลอมแปลงโดยนำเอาระบบการเข้ารหัสลับมาใช้
2. เพื่อออกแบบและพัฒนาระบบการเลือกตั้งที่สอดคล้องกับบัตรเลือกตั้ง
3. เพื่อพัฒนาต้นแบบบัตรเลือกตั้งที่นำระบบการเข้ารหัสลับมาใช้
4. เพื่อพัฒนาโปรแกรมสำหรับอุปกรณ์เข้าและถอดรหัสลับที่ใช้กับบัตรเลือกตั้ง

1.3 ขอบเขตของการวิจัย

1. สร้างต้นแบบที่เป็นส่วนประกอบของระบบเลือกตั้ง ประกอบด้วย
   1.1. โปรแกรมบนไมโครคอนโทรลเลอร์ สำหรับถอดรหัสลับและจัดการกุญแจที่หน่วยเลือกตั้ง
   1.2. โปรแกรมบนไมโครคอมพิวเตอร์สำหรับการดึงข้อมูลที่เป็นคุณลักษณะเฉพาะตัวของกระดาษ, การเข้ารหัสลับ, การพิมพ์รหัสแถบ, การตรวจสอบความถูกต้องของบัตรเลือกตั้งที่เขตเลือกตั้ง และการจัดการกุญแจ
2. ปรับกระบวนการเลือกตั้งให้สอดคล้องกับระบบ จนสามารถนำไปใช้ได้โดยมีช่องโหว่น้อยที่สุด พร้อมการวิเคราะห์จุดอ่อนของระบบ

1.4 วิธีดำเนินการวิจัย

1. ศึกษาวิทยาการเข้ารหัสลับ
2. ออกแบบระบบทั้งหมดในทุก ๆ รายละเอียด ขั้นตอนต่าง ๆ ให้ไม่มีช่องโหว่
3. พัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์ เพื่อใช้สำหรับถอดรหัสลับและจัดการกุญแจ
4. ออกแบบและปรับปรุงวิธีการดึงข้อมูลที่เป็นคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

5. พัฒนาโปรแกรมบนไมโครคอมพิวเตอร์ เพื่อใช้สำหรับดึงข้อมูลที่เป็นคุณลักษณะเฉพาะตัวของกระดาษ, เข้ารหัสลับ, พิมพ์รหัสแถบ, ตรวจสอบความถูกต้องของบัตรเลือกตั้ง และจัดการกุญแจ
6. ทดสอบและปรับปรุงอุปกรณ์ต้นแบบ
7. สรุปผลการทดสอบและเขียนวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำไปพัฒนาต่อเพื่อใช้แทนระบบเลือกตั้งในปัจจุบัน เพื่อป้องกันการปลอมแปลงและทุจริตได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้ ได้กล่าวถึงทฤษฎีต่าง ๆ ที่เกี่ยวข้องในการออกแบบและสร้างต้นแบบระบบเลือกตั้ง ได้แก่ ระบบการเข้ารหัสลับแบบกุญแจสาธารณะด้วยวิธี RSA วิธีการคำนวณการยกกำลังและหาเศษจากการหารซึ่งใช้กับ RSA ไมโครคอนโทรลเลอร์ตระกูล 8051 ซึ่งใช้ในการถอดรหัสลับด้วยวิธี RSA การทำโครงร่างภาพ ซึ่งใช้ในการดึงข้อมูลเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง และสุดท้ายเป็นเรื่องรหัสแท่ง Code 128 ซึ่งใช้ในการพิมพ์รหัสลับลงบนบัตรเลือกตั้ง

2.1 ระบบการเข้ารหัสลับแบบกุญแจสาธารณะ (Public-key Cryptosystem)

ระบบการเข้ารหัสลับแบบเดิมนั้น เรียกว่า ระบบการเข้ารหัสลับแบบกุญแจลับ (Secret-key Cryptosystem) เป็นระบบที่ใช้กุญแจลับ (Secret key) เพียงตัวเดียว เมื่อผู้ส่งต้องการส่งข้อความไปยังผู้รับก็จะใช้กุญแจลับเพื่อเข้ารหัสลับข้อความที่ต้องการส่ง และผู้รับจะใช้กุญแจลับตัวเดียวกันในการถอดรหัสลับข้อความ ซึ่งปัญหาหลักของระบบนี้คือ การตกลงเรื่องกุญแจลับระหว่างผู้ส่งกับผู้รับโดยไม่มีผู้อื่นรู้เห็น ถ้าทั้งคู่มิได้อยู่ในที่แห่งเดียวกัน ก็จะต้องมีช่องทางการสื่อสารที่ปลอดภัยที่ผู้อื่นไม่อาจรู้ได้ ซึ่งถ้ามีช่องทางดังกล่าวจริง ทำไมไม่ใช้ช่องทางนั้นส่งข้อความที่ต้องการสื่อสารกันเลย ไม่จำเป็นต้องใช้ระบบการเข้ารหัสลับ จะเห็นว่า ระบบการเข้ารหัสลับแบบกุญแจลับจะมีความยุ่งยากในการจัดการกุญแจ โดยเฉพาะในระบบที่มีผู้ใช้จำนวนมาก ก็จะต้องใช้กุญแจเป็นจำนวนมากด้วย ดังนั้น ระบบการเข้ารหัสลับแบบกุญแจสาธารณะจึงเกิดขึ้นในปี ค.ศ. 1976 โดย Whitfield Diffie และ Martin Hellman [1] เพื่อแก้ปัญหาในเรื่องการจัดการกุญแจ

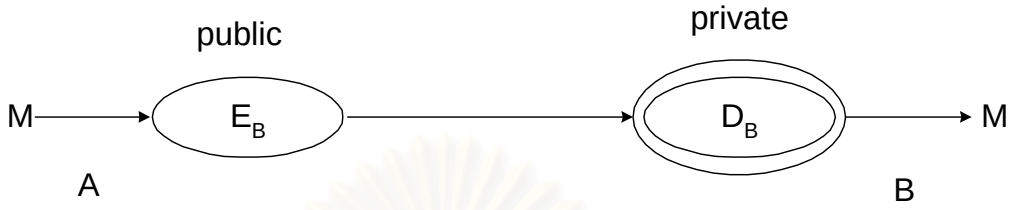
ในระบบการเข้ารหัสลับแบบกุญแจสาธารณะ ผู้ใช้แต่ละคนจะมีกุญแจอยู่ 1 คู่ คือ กุญแจสาธารณะ (Public key) และกุญแจส่วนตัว (Private key) โดยกุญแจสาธารณะจะสามารถประกาศให้ผู้อื่นรู้ได้ ส่วนกุญแจส่วนตัวจะเก็บไว้เป็นความลับ ผู้ใช้ 2 คนจะสามารถติดต่อสื่อสารกันได้เพียงแค่รู้กุญแจสาธารณะของอีกฝ่ายหนึ่งเท่านั้น ดังนั้นจึงไม่มีความจำเป็นที่จะต้องแลกเปลี่ยนข้อมูลที่เป็นความลับกันระหว่างผู้ส่งกับผู้รับเหมือนกับในระบบการเข้ารหัสลับแบบกุญแจลับ นอกจากนั้น การรู้กุญแจตัวหนึ่งจะไม่สามารถคำนวณหากุญแจอีกตัวหนึ่งที่ใช้คู่กันได้โดยง่าย นอกจากลองเดาไปเรื่อย ๆ ซึ่งใช้เวลานานมาก

ในระบบการเข้ารหัสลับแบบกุญแจสาธารณะนั้น ความต้องการในเรื่อง *ความเป็นส่วนตัว (Privacy)* และ *การยืนยันข้อมูล (Authentication)* จะแยกจากกัน ดังนี้

- **ความเป็นส่วนตัว**

เมื่อผู้ใช้ A ต้องการส่งข่าวสาร M ไปให้กับ B เมื่อ A ทราบกุญแจสาธารณะของ B ผู้ใช้ A ก็จะสามารถส่งข่าวสาร M ไปให้กับ B อย่างเป็นความลับได้ โดยการส่งข่าวสารที่เข้ารหัสลับแล้ว

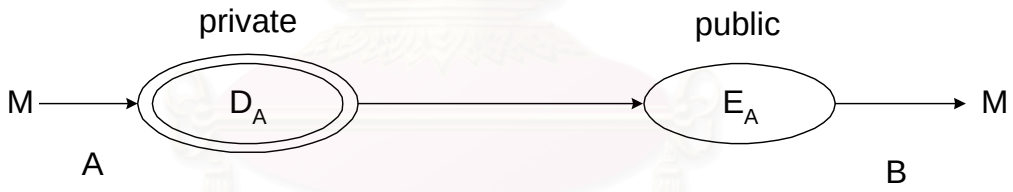
$C = E_B(M)$ เมื่อ B ได้รับก็จะถอดรหัสลับ C ด้วยกุญแจส่วนตัวของตนเอง นั่นคือ $D_B(C) = D_B(E_B(M)) = M$ ดังรูปที่ 2.1 ซึ่งผู้อื่นที่ไม่ใช่ B ถึงแม้จะดักฟังข่าวสารมาได้ แต่ก็ไม่สามารถถอดรหัสลับที่ได้มานั้นได้ เพราะไม่มีกุญแจส่วนตัวของ B จะเห็นว่าการส่งข่าวสารวิธีนี้จะไม่รับประกันในเรื่องการยืนยันข้อมูล เนื่องจากผู้ที่รู้กุญแจสาธารณะของ B สามารถปลอมข่าวสารเพื่อส่งให้ B ได้



รูปที่ 2.1 ความเป็นส่วนตัวในระบบกุญแจสาธารณะ

- **การยืนยันข้อมูล**

ความต้องการในเรื่องการยืนยันข้อมูล สามารถทำได้โดย A จะเข้ารหัสลับโดยใช้กุญแจส่วนตัวของตนเอง คือจะส่ง $C = D_A(M)$ ไปให้ B เมื่อ B ได้รับก็จะถอดรหัสโดยใช้กุญแจสาธารณะของ A นั่นคือ $E_A(C) = E_A(D_A(M)) = M$ ดังรูปที่ 2.2 เนื่องจากบทบาทของ E_A และ D_A สามารถสลับกันได้ จะเห็นว่าข่าวสารที่ได้รับมาจาก A อย่างแน่นอนเพราะมี A เพียงคนเดียวที่รู้กุญแจส่วนตัวของ A แต่วิธีนี้จะไม่รับประกันว่าข่าวสารนี้จะเป็นความลับ เนื่องจากผู้ใดก็ตามที่รู้กุญแจสาธารณะของ A จะสามารถถอดรหัสลับได้

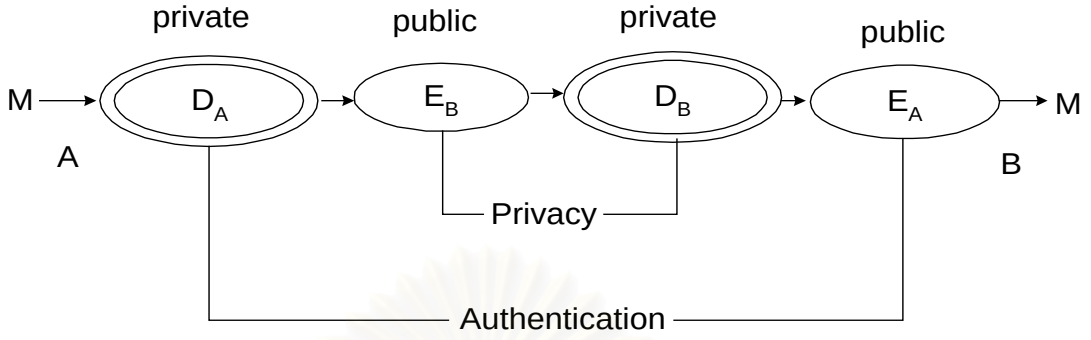


รูปที่ 2.2 การยืนยันข้อมูลในระบบกุญแจสาธารณะ

การจะทำให้ระบบการเข้ารหัสลับแบบกุญแจสาธารณะสามารถรับประกันได้ทั้งในเรื่องของความเป็นส่วนตัวและการยืนยันข้อมูล ทำได้โดยเมื่อ A ต้องการติดต่อกับ B ผู้ใช้ A จะเข้ารหัสลับโดยใช้กุญแจส่วนตัวของตนเอง จากนั้นจะเข้ารหัสลับอีกชั้นหนึ่งโดยใช้กุญแจสาธารณะของ B แล้วจึงส่งข้อความที่เข้ารหัสลับแล้ว $C = E_B(D_A(M))$ ไปให้ B เมื่อ B ได้รับก็จะถอดรหัสลับโดยใช้กุญแจส่วนตัวของตนเอง แล้วจึงใช้กุญแจสาธารณะของ A มาถอดรหัสลับอีกชั้นหนึ่ง ดังรูปที่ 2.3

ข้อดีของระบบการเข้ารหัสลับแบบกุญแจสาธารณะที่เหนือกว่าระบบการเข้ารหัสลับแบบกุญแจลับ (Secret-key cryptosystem) ก็คือเราสามารถให้กุญแจสาธารณะของเรากับทุก ๆ คนที่เรารู้จักหรือไม่รู้จักก็ได้ เพื่อที่จะสามารถส่งข่าวสารมาถึงเราได้ โดยจะไม่มีผู้ใดที่จะสามารถทราบข่าวสารนั้นได้นอกจากตัวเรา เพราะเราจะเป็นเพียงบุคคลเดียวที่มีกุญแจส่วนตัว แต่ในกรณีของ

ระบบการเข้ารหัสลับแบบกุญแจลับ เราจะให้กุญแจลับ (Secret key) กับบุคคลที่เราไว้ใจเท่านั้น และกุญแจลับที่ให้กับแต่ละคนก็แตกต่างกันด้วย



รูปที่ 2.3 ความเป็นส่วนตัวและการยืนยันข้อมูลในระบบกุญแจสาธารณะ

2.2 ระบบการเข้ารหัสลับด้วยวิธี RSA

RSA เป็นระบบการเข้ารหัสลับแบบกุญแจสาธารณะซึ่งสามารถใช้รองรับความต้องการได้ทั้งความเป็นส่วนตัว (Privacy) โดยใช้ *การเข้ารหัสลับ* และการยืนยันข้อมูล (Authentication) โดยใช้ *ลายเซ็นดิจิตอล* Ron Rivest, Adi Shamir และ Leonard Adleman ได้พัฒนา RSA ขึ้นในปี ค.ศ.1977 [1] คำว่า RSA มาจากอักษรตัวแรกของชื่อท้ายของผู้คิดค้นแต่ละท่าน

2.2.1 การสร้างกุญแจ

การสร้างกุญแจของ RSA มีขั้นตอนดังนี้

- หาจำนวนเฉพาะที่มีค่ามาก 2 ค่า ให้เป็น p และ q
- คำนวณผลคูณ n = pq โดยเรียก n ว่า modulus
- คำนวณค่า Euler's totient function [2] ตามสมการ

$$\phi(n) = (p-1)(q-1) \tag{2-1}$$

- เลือกค่า e ให้มีค่าน้อยกว่า n และเป็นจำนวนเฉพาะสัมพัทธ์กับ $\phi(n)$ คือ e กับ $\phi(n)$ มี ห.ร.ม. เป็น 1 สาเหตุที่ e ต้องเป็นจำนวนเฉพาะสัมพัทธ์กับ $\phi(n)$ ก็เพื่อ e จะได้มีอินเวอร์ส โดยปกติแล้วเราจะเลือกค่า e ให้มีค่าน้อย เช่น $e = 2^{16} + 1$
- หาค่า d ที่ทำให้ (ed-1) หารด้วย $\phi(n)$ ลงตัว หรือ

$$1 = ed \bmod \phi(n) \tag{2-2}$$

(นั่นคือ d เป็นอินเวอร์สของ e) ด้วยวิธี extended Euclidean algorithm [2]

- เรียกค่า e และ d ว่า ตัวยกกำลังสาธารณะ (Public exponent) และตัวยกกำลังส่วนตัว (Private exponent) ตามลำดับ

- กุญแจสาธารณะคือ (n,e) และกุญแจส่วนตัวคือ (n,d) ส่วนตัวประกอบ p และ q อาจเก็บไว้กับกุญแจส่วนตัวหรือทิ้งไปก็ได้

ในปัจจุบันนี้ ยังเป็นการยากที่จะหาค่ากุญแจส่วนตัว d จากกุญแจสาธารณะ (n,e) แต่อย่างไรก็ตาม หากสามารถแยกตัวประกอบ n เป็น p กับ q ได้ ก็จะรู้ค่ากุญแจส่วนตัว d ดังนั้นความปลอดภัยของ RSA จึงตั้งอยู่บนสมมติฐานที่ว่า การแยกตัวประกอบเป็นเรื่องยาก การค้นพบวิธีใหม่ ๆ ที่ทำให้การแยกตัวประกอบง่ายขึ้น ก็จะทำให้ไม่สามารถใช้ RSA ได้ด้วย

2.2.2 การใช้งาน

การใช้งาน RSA อาจแบ่งได้เป็น 2 ประเภท คือ การเข้ารหัสลับกับการใช้ลายเซ็นดิจิตอล (การใช้งานจริงจะใช้โพรโตคอลที่ซับซ้อนกว่าที่กล่าวไว้ในที่นี้)

- **การเข้ารหัสลับ**

สมมติว่า อลิสต้องการส่งข้อความ m ให้กับบ็อบ อลิสจะสร้างรหัสลับ c โดยการยกกำลังตามสมการ

$$c = m^e \bmod n \tag{2-3}$$

โดย e และ n คือกุญแจสาธารณะของบ็อบ เมื่อสร้างเสร็จก็ส่ง c ไปให้บ็อบ

เมื่อบ็อบได้รับรหัสลับ c ก็จะนำมาถอดรหัสลับโดยการยกกำลังเช่นกัน ตามสมการ

$$m = c^d \bmod n \tag{2-4}$$

จากทฤษฎีบทของ Euler [2] ที่ว่า

$$a^{\phi(n)} = 1 \pmod n \tag{2-5}$$

เมื่อให้ n และ a เป็นค่าบวก และเป็นจำนวนเฉพาะสัมพัทธ์กัน และจากความสัมพันธ์ของ e กับ d ตามสมการ (2-2) สามารถเขียนได้เป็น $ed = 1 + k\phi(n)$ เมื่อ k เป็นจำนวนเต็มบางค่า จะได้

$$\begin{aligned} c^d &= (m^e)^d \pmod n \\ &= m^{ed} \pmod n \\ &= m^{1+k\phi(n)} \pmod n \\ &= m \cdot (m^{\phi(n)})^k \pmod n \\ &= m \cdot 1 \pmod n \end{aligned}$$

จึงทำให้บ็อบสามารถรู้ข้อความ m ได้อย่างถูกต้อง และเนื่องจากมีบ็อบเพียงคนเดียวที่รู้ค่า d บ็อบจึงเป็นคนเดียวที่สามารถถอดรหัสลับข้อความได้

- **ลายเซ็นดิจิตอล**

ลายเซ็นดิจิตอลจะมีลักษณะแตกต่างจากลายเซ็นที่เซ็นบนกระดาษ คือ เป็นฟังก์ชันของเอกสารดิจิตอล ขณะที่ลายเซ็นบนกระดาษจะเหมือนกันทุกเอกสาร

สมมติว่า อลิสต้องการส่งข้อความ m ให้กับบ็อบในลักษณะที่รับประกันได้ว่าบ็อบจะได้ข้อความต้นฉบับที่แท้จริงไม่มีการแก้ไข และส่งมาจากอลิสจริง อลิสจะสร้างลายเซ็นดิจิตอล s โดยการยกกำลังตามสมการ $s = m^d \bmod n$ โดย d และ n คือกุญแจส่วนตัวของอลิส เมื่อสร้างเสร็จก็ส่ง m กับ s ไปให้บ็อบ

เมื่อบ็อบได้รับข้อความ m กับลายเซ็นดิจิตอล s ก็จะนำมาตรวจสอบลายเซ็นโดยการยกกำลังเช่นกัน ตามสมการ $m = s^e \bmod n$ โดย e และ n คือกุญแจสาธารณะของอลิส แล้วตรวจสอบค่า m ว่าตรงกันหรือไม่

จะเห็นว่าทั้งความเป็นส่วนตัวและการยืนยันข้อมูลสามารถเกิดขึ้นได้โดยปราศจากการเปิดเผยข้อมูลกุญแจส่วนตัว แต่ละคนจะใช้เพียงกุญแจสาธารณะของคนอื่นและกุญแจส่วนตัวของคนผู้นั้นเอง ทุกคนสามารถส่งข้อความที่เข้ารหัสลับแล้ว และสามารถตรวจสอบข้อความที่มีลายเซ็นกำกับไว้ แต่มีเพียงคนเดียวที่เป็นเจ้าของกุญแจส่วนตัวที่ถูกต้องที่สามารถถอดรหัสลับหรือเซ็นข้อความได้

2.2.3 ตัวอย่างการคำนวณ

สมมติให้ p = 5 และ q = 7 เราสามารถสร้างกุญแจได้โดย

- หาผลคูณ n = pq = 5x7 = 35
- คำนวณค่า (p-1)(q-1) = 4x6 = 24
- เลือก e ที่มีค่าน้อยกว่า 24 และไม่เป็นจำนวนเฉพาะสัมพัทธ์กับ 24 ในที่นี้เลือก 11
- หาค่า d ที่ทำให้ $1 = 11 \cdot d \bmod 24$ จะได้ d = 11

เมื่อต้องการเข้ารหัสลับข้อความ m = 2 จะได้ $c = m^e \bmod n = 2^{11} \bmod 35 = 18$

จากนั้น ทำการถอดรหัสลับข้อความ จะได้ $m = c^d \bmod n = 18^{11} \bmod 35 = 2$

2.3 การคำนวณการยกกำลังและหาเศษจากการหาร

ระบบการเข้ารหัสลับด้วยวิธี RSA นั้น ไม่ว่าจะเป็นการเซ็น การพิสูจน์ลายเซ็น การเข้ารหัสลับหรือการถอดรหัสลับ ล้วนมีการคำนวณที่สำคัญคือ การยกกำลังและหาเศษจากการหาร (Modular exponentiation) ในการคำนวณ เราจะไม่หาค่า $c = m^e \bmod n$ ด้วยการยกกำลัง m^e แล้วทำการหารเพื่อหาค่าเศษ $c = (m^e)\%\, n$ โดยตรง เนื่องจากค่า m^e เป็นค่าที่ใหญ่มาก ตัวอย่างเช่น ให้ m และ e แต่ละตัวมีขนาด 256 บิต เราจะต้องการที่ขนาด

$$\log_2(m^e) = e \log_2 m \approx 2^{256} \cdot 256 = 2^{264} \approx 10^{80}$$

บิต ในการเก็บค่า m^e ดังนั้น จึงต้องใช้ความสัมพันธ์

$$(xy) \bmod n = [(x \bmod n)(y \bmod n)] \bmod n \qquad (2\text{-}6)$$

เข้ามาช่วยในการคำนวณ โดยในที่นี้ จะกล่าวถึงการคำนวณการยกกำลังและหาเศษจากการหารด้วยวิธีไบนารี (Binary method) ซึ่งจะใช้ในการเขียนโปรแกรมภาษาแอสเซมบลีบนไมโครคอนโทรลเลอร์เพื่อถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งในหัวข้อ 4.6.3 การคำนวณประกอบด้วยขั้นตอนย่อย คือ การคูณ การยกกำลังสอง และการหาเศษจากการหาร โดยมีรายละเอียดดังนี้

2.3.1 การยกกำลังและหาเศษจากการหาร

การยกกำลังและหาเศษจากการหารด้วยวิธีไบนารีนั้น จะดูค่าบิตของตัวยกกำลังทีละบิตจากซ้ายไปขวาหรือขวาไปซ้ายก็ได้ โดยแต่ละบิตที่ผ่านไป จะมีการยกกำลังสองแล้วดูค่าบิตนั้นถ้าเป็น 1 ก็จะมีการคูณด้วย ทำไปเรื่อย ๆ จนครบทุกบิตของตัวยกกำลัง ในที่นี้จะขออธิบายวิธีไบนารีแบบซ้ายไปขวา [3] เนื่องจากแบบขวาไปซ้ายต้องใช้ตัวแปรเพิ่มขึ้นสำหรับเก็บผลการยกกำลังสองของ m ในแต่ละขั้น

ให้ k เป็นจำนวนบิตของ e คือ $k = 1 + \log_2 e$ และ e สามารถเขียนให้อยู่ในรูปเลขฐานสองได้เป็น

$$e = (e_{k-1} e_{k-2} \ldots e_1 e_0) = \sum_{i=0}^{k-1} e_i 2^i \qquad (2\text{-}7)$$

โดย $e_i \in \{0,1\}$ วิธีไบนารีจะคำนวณค่า $c = m^e \bmod n$ ตามขั้นตอนดังนี้

```
1:    if e_{k-1} = 1 then c = m else c = 1
2:    for i = k-2 down to 0
3:        c = c·c (mod n)
4:        if e_i = 1 then c = c·m (mod n)
5:    return c
```

ตัวอย่างการคำนวณ สมมติให้ e = 250 = (11111010) จะได้ k = 8 เริ่มแรกให้ค่า c = m เนื่องจาก $e_{k-1} = e_7 = 1$ จากนั้นจึงทำการคำนวณตามวิธีไบนารีได้ดังนี้

i	e_i	ขั้นที่ 3	ขั้นที่ 4
6	1	$(m)^2 = m^2$	$m^2 \cdot m = m^3$
5	1	$(m^3)^2 = m^6$	$m^6 \cdot m = m^7$
4	1	$(m^7)^2 = m^{14}$	$m^{14} \cdot m = m^{15}$
3	1	$(m^{15})^2 = m^{30}$	$m^{30} \cdot m = m^{31}$
2	0	$(m^{31})^2 = m^{62}$	m^{62}

i	e_i	ขั้นที่ 3	ขั้นที่ 4
1	1	$(m^{62})^2 = m^{124}$	$m^{124} \cdot m = m^{125}$
0	0	$(m^{125})^2 = m^{250}$	m^{250}

การยกกำลังและหาเศษจากการหารนี้ จะใช้ในการเขียนโปรแกรมภาษาแอสเซมบลีในหัวข้อ 4.6.3.2.1 โดยมีการเรียกใช้ขั้นตอนย่อยคือ การคูณ การยกกำลังสอง และการหาเศษจากการหาร ในหัวข้อ 2.3.2, 2.3.3 และ 2.3.4 ตามลำดับ

2.3.2 การคูณ

การคูณที่จะกล่าวถึงนี้ เป็นการคูณเลข 2 จำนวนที่มีความยาวไม่จำกัด [3] ซึ่งใช้ในการเขียนโปรแกรมภาษาแอสเซมบลีในหัวข้อ 4.6.3.2.4

ให้ a และ b เป็นเลขฐาน W มีความยาว s word ดังสมการ

$$a = (a_{s-1}a_{s-2}...a_0) = \sum_{j=0}^{s-1} a_j W^j$$
$$b = (b_{s-1}b_{s-2}...b_0) = \sum_{i=0}^{s-1} b_i W^i \qquad (2\text{-}8)$$

โดยที่แต่ละหลักของ a และ b อยู่ในช่วง [0, W-1] และ $W = 2^w$ ซึ่ง w คือ จำนวนบิตใน 1 word การคูณ a กับ b ตามปกติ (การคูณด้วยมือ) จะต้องหาผลคูณของตัวคูณ b 1 หลัก กับค่า a ทั้งหมดก่อน และเมื่อหาได้ครบทุกหลักของ b แล้ว จึงนำผลคูณที่ได้มาบวกกันเพื่อจะได้เป็นผลลัพธ์สุดท้าย t ขนาด 2s word ให้ t_{ij} เป็นคู่ของตัวทดและผลลัพธ์ที่ได้จากการคูณ a_j กับ b_i ตัวอย่างเช่น เมื่อ W = 10 และ $a_j = 7$, $b_i = 8$ จะได้ $t_{ij} = (5,6)$ ค่า t_{ij} สามารถเขียนได้ดังนี้

				a_3	a_2	a_1	a_0	(= a)
x				b_3	b_2	b_1	b_0	(= b)
				t_{03}	t_{02}	t_{01}	t_{00}	
			t_{13}	t_{12}	t_{11}	t_{10}		
		t_{23}	t_{22}	t_{21}	t_{20}			
+	t_{33}	t_{32}	t_{31}	t_{30}				
t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0	(= t)

ให้แถวสุดท้าย เป็นผลรวมของผลคูณของแต่ละหลัก และแทนผลลัพธ์ขนาด 2s word เพื่อเป็นการประหยัดเนื้อที่หน่วยความจำ การเก็บผลคูณของตัวคูณ b แต่ละหลักจะใช้ตัวแปรตัวเดียวคือ t ซึ่งเป็นตัวแปรในแถวสุดท้าย โดยเริ่มจากให้ค่าเริ่มต้นของ t เป็นศูนย์ แล้วนำแต่ละหลักของ b คูณกับ ค่า a ทุกหลัก แล้วจึงบวกผลลัพธ์ที่ได้เข้ากับ t ทำไปเรื่อย ๆ จนกระทั่งครบทุกหลักของ b จะได้ t เป็นผลคูณของ a กับ b ดังที่ได้แสดงต่อไปนี้

```
1:    Initially t_i = 0 for all i = 0, 1, 2, …, 2s-1
2:    for i = 0 to s-1
3:        C = 0
4:        for j = 0 to s-1
5:            (C,S) = t_{i+j} + a_j·b_i + C
6:            t_{i+j} = S
7:        t_{i+s} = C
8:    return (t_{2s-1}t_{2s-2}…t_0)
```

ตัวอย่างการคำนวณตามระเบียบวิธีข้างบน สมมติให้ a = 348, b = 857 (ฐาน 10) สามารถหาผลคูณได้ดังนี้

i	j	ขั้นตอน	(C,S)	ผลคูณ t
0	0	$t_0 + a_0b_0 + C$	(0, x)	000000
		$0 + 8\cdot7 + 0$	(5, 6)	000006
	1	$t_1 + a_1b_0 + C$		
		$0 + 4\cdot7 + 5$	(3, 3)	000036
	2	$t_2 + a_2b_0 + C$		
		$0 + 3\cdot7 + 3$	(2, 4)	000436
				002436
1	0	$t_1 + a_0b_1 + C$	(0, x)	
		$3 + 8\cdot5 + 0$	(4, 3)	002436
	1	$t_2 + a_1b_1 + C$		
		$4 + 4\cdot5 + 4$	(2, 8)	002836
	2	$t_3 + a_2b_1 + C$		
		$2 + 3\cdot5 + 2$	(1, 9)	009836
				019836
2	0	$t_2 + a_0b_2 + C$	(0, x)	
		$8 + 8\cdot8 + 0$	(7, 2)	019236
	1	$t_3 + a_1b_2 + C$		
		$9 + 4\cdot8 + 7$	(4, 8)	018236
	2	$t_4 + a_2b_2 + C$		
		$1 + 3\cdot8 + 4$	(2, 9)	098236
				298236

การทำงานในขั้น 5 ตามสมการ

$$(C,S) = t_{i+j} + a_j \cdot b_i + C \qquad (2\text{-}9)$$

ค่าตัวแปร t_{i+j}, a_j, b_i, C และ S มีขนาด 1 word หรือ w บิต เมื่อทำตามสมการ (2-9) แล้วจะได้ผลลัพธ์ขนาด 2 word หรือ 2w บิต เนื่องจากได้ค่ามากที่สุดเป็น

$$2^w - 1 + (2^w - 1)(2^w - 1) + 2^w - 1 = 2^{2w} - 1 \qquad (2\text{-}10)$$

2.3.3 การยกกำลังสอง

การยกกำลังสองที่จะกล่าวถึงนี้ เป็นการยกกำลังสองของเลขที่มีความยาวไม่จำกัด [3] ซึ่งใช้ในการเขียนโปรแกรมภาษาแอสเซมบลีในหัวข้อ 4.6.3.2.5

การยกกำลังสองมีลักษณะคล้ายกับการคูณแต่มีการทำงานที่ง่ายกว่า เพราะสามารถลดการคำนวณไปได้ครึ่งหนึ่ง เนื่องจาก $t_{ij} = a_i \cdot a_j = t_{ji}$

				a_3	a_2	a_1	a_0	(= a)
x				a_3	a_2	a_1	a_0	(= a)
				t_{03}	t_{02}	t_{01}	t_{00}	
			t_{13}	t_{12}	t_{11}	t_{01}		
		t_{23}	t_{22}	t_{12}	t_{02}			
+	t_{33}	t_{23}	t_{13}	t_{03}				
				$2t_{03}$	$2t_{02}$	$2t_{01}$	t_{00}	
			$2t_{13}$	$2t_{12}$	t_{11}			
		$2t_{23}$	t_{22}					
+	t_{33}							
t_7	t_6	t_5	t_4	t_3	t_2	t_1	t_0	(= t)

ดังนั้น เราจึงสามารถปรับเปลี่ยนขั้นตอนการคูณให้เป็นการยกกำลังสองได้ดังนี้

```
1:    Initially t_i = 0 for all i = 0, 1, 2, …, 2s-1
2:    for i = 0 to s-1
3         (C,S) = t_{i+i} + a_i·a_i
4:        t_{i+i} = S
5:        for j = i+1 to s-1
6:            (C,S) = t_{i+j} + 2·a_j·a_i + C
7:            t_{i+j} = S
8:        t_{i+s} = C
9:    return (t_{2s-1}t_{2s-2}…t_0)
```

อย่างไรก็ตาม การทำงานในขั้นที่ 6 ตามสมการ

$$(C,S) = t_{i+j} + 2 \cdot a_j \cdot a_i + C \qquad (2\text{-}11)$$

ผลลัพธ์ที่ได้จะมีขนาด 2w+1 บิต แทนที่จะเป็น 2w บิต เนื่องจาก

$$(2^w-1)+2(2^w-1)(2^w-1)+(2^w-1)=2^{2w+1}-2^{w+1} \quad (2\text{-}12)$$

และ

$$2^{2w}-1<2^{2w+1}-2^{w+1}<2^{2w+1}-1 \quad (2\text{-}13)$$

2.3.4 การหาเศษจากการหาร

ให้ t เป็นตัวตั้ง, n เป็นตัวหาร ผลหาร Q และเศษ R จะต้องเป็นไปตามสมการ

$$t=Q\cdot n+R \quad , \quad R<n \quad (2\text{-}14)$$

โดยถ้า t และ n เป็นค่าบวก จะได้ผลหาร Q และเศษ R เป็นค่าบวกด้วย สิ่งที่เราสนใจจากการหารก็คือเศษจากการหารเท่านั้น ซึ่งสามารถหาได้โดยวิธีการหารเป็นลำดับ (sequential division) และต่อไปนี้ก็เป็นวิธีการหารเป็นลำดับชนิดหนึ่งที่เรียกว่า ระเบียบวิธีการหารแบบคืนค่า [3] (Restoring division algorithm) ซึ่งใช้ในการเขียนโปรแกรมภาษาแอสเซมบลีในหัวข้อ 4.6.3.2.6

ให้ R_i เป็นเศษที่ได้ระหว่างการหารขั้นที่ i การทำงานเริ่มจากการจัดให้ด้านซ้ายของ t และ n ตรงกัน เนื่องจาก t เป็นเลขขนาด 2k บิต และ n เป็นเลขขนาด k บิต การจัดให้ซ้ายตรงกันจึงหมายถึง เลื่อน n ไปทางซ้าย k บิต นั่นคือ $2^k n$ จากนั้น ให้ค่าเริ่มต้นของ R เป็น t หรือ $R_0 = t$ แล้วเริ่มลบ n ที่ได้เลื่อนไปออกจาก t เพื่อให้ได้ R_1 ถ้า R_1 เป็นค่าบวกหรือศูนย์ ก็ทำขั้นต่อไป แต่ถ้า R_1 เป็นค่าลบ ก็จะใช้ค่าเดิมเป็นเศษแทน

```
1:    R0 = t
2:    n = 2^k n
3:    for i = 0 to k+1
4:        Ri = Ri-1 – n
5:        If Ri < 0 then Ri = Ri-1
6:        n = n / 2
7:    return Rk+1
```

ในขั้นที่ 5 เราจะดูเครื่องหมายของเศษ ถ้าเป็นลบ ก็นำค่าเศษเดิมมาใช้ คือเป็นกระบวนการคืนค่า (Restore) แต่ถ้าเศษ R_i เป็นบวก ก็ไม่ต้องคืนค่า ใช้ค่านั้นเป็นเศษต่อไป

ยกตัวอย่างการคำนวณตามระเบียบวิธีนี้ สมมติว่า ต้องการหาค่า 3019 mod 53 โดย $3019 = (101111001011)_2$ และ $53 = (110101)_2$ ผลลัพธ์คือ $51 = (110011)_2$

i				
	R_0	101111	001011	t
	n	110101		ลบ
1		000110		เศษค่าลบ
	R_1	101111	001011	คืนค่าเดิม
	n / 2	11010	1	เลื่อนและลบ
2		10100	1	เศษค่าบวก
	R_2	10100	101011	ไม่คืนค่า
	n / 2	1101	01	เลื่อนและลบ
3		0111	01	เศษค่าบวก
	R_3	0111	011011	ไม่คืนค่า
	n / 2	110	101	เลื่อนและลบ
4		000	110	เศษค่าบวก
	R_4	000	110011	ไม่คืนค่า
	n / 2	11	0101	เลื่อน
5	n / 2	1	10101	เลื่อน
6	n / 2		110101	เลื่อนและลบ
7			000010	เศษค่าลบ
	R_7		110011	คืนค่าเดิม
	R		110011	เศษสุดท้าย

ก่อนการลบ เราอาจตรวจสอบค่าบิตที่มีนัยสำคัญมากสุด (MSB) ของเศษก่อน ถ้าเป็นศูนย์ ก็ไม่ต้องลบเนื่องจาก $n > R_i$ เพียงเลื่อน n ไปจนกว่าจะพบกับบิตที่มีนัยสำคัญมากสุดของ R_i ที่ไม่เป็นศูนย์ ด้วยวิธีนี้ เราจะสามารถลดขั้นตอนการลบ/เรียกคืนค่าได้ โดยเฉลี่ยแล้วจะมีการลบ k/2 ครั้ง

2.4 **ไมโครคอนโทรลเลอร์ตระกูล** 8051

ไมโครคอนโทรลเลอร์เป็นไมโครโปรเซสเซอร์ประเภทหนึ่งที่ได้รับการออกแบบมาเพื่อใช้งานกับระบบควบคุมที่มีขนาดเล็ก โดยภายในไอซีไมโครคอนโทรลเลอร์หนึ่งตัวจะประกอบด้วยหน่วยการทำงานหลักของระบบคอมพิวเตอร์ครบถ้วน เช่น หน่วยประมวลผลกลาง หน่วยความจำ พอร์ตในการติดต่อหรือควบคุมอุปกรณ์ต่าง ๆ เป็นต้น ซึ่งหากว่าเป็นการใช้งานไมโครโปรเซสเซอร์ทั่วไปก็จะต้องใช้ไอซีภายนอกมาประกอบเพื่อทำหน้าที่เหล่านี้ ดังนั้นจึงอาจกล่าวได้ว่า ไมโครคอนโทรลเลอร์เป็นระบบคอมพิวเตอร์เพื่องานควบคุมที่สมบูรณ์ โดยบรรจุอยู่ภายในตัวไอซีเพียงหนึ่งตัวเท่านั้น ในบางครั้งจึงอาจพบว่ามีการเรียกไมโครคอนโทรลเลอร์ว่าเป็น ระบบไมโครคอมพิวเตอร์ชิปเดี่ยว

ในบรรดาไมโครคอนโทรลเลอร์ที่มีการผลิตจากบริษัทต่าง ๆ ไมโครคอนโทรลเลอร์ของบริษัทอินเทล ตระกูล MCS-51 ได้มีการนำไปใช้งานกันแพร่หลายมาก นับตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา และในระยะที่ผ่านมาได้มีการนำไปผลิตจำหน่ายโดยบริษัทอื่น ๆ เช่น บริษัทฟิลิปส์ หรือ ซีเมนส์ เป็นต้น พร้อมกับการเพิ่มประสิทธิภาพและหน่วยการทำงานต่าง ๆ มากขึ้น ทำให้มีไมโครคอนโทรลเลอร์จากผู้ผลิตต่าง ๆ ที่มีพื้นฐานมาจากไมโครคอนโทรลเลอร์ MCS-51 ของบริษัทอินเทลเป็นจำนวนมาก

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วยไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหรือจำนวนของหน่วยทำงานภายในที่แตกต่างกันออกไป เพื่อความเหมาะสมในงานประยุกต์ต่าง ๆ ตามความต้องการ ในที่นี้จะเรียกรวม ๆ ว่าไมโครคอนโทรลเลอร์ตระกูล 8051 [4 และ 5]

ไมโครคอนโทรลเลอร์นี้ จะใช้ในการถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งในหัวข้อ 4.5.1

2.4.1 **คุณลักษณะพื้นฐานของ** 8051

ไมโครคอนโทรลเลอร์ตระกูล 8051 มีคุณสมบัติพื้นฐานต่าง ๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EEPROM (เบอร์ 8751) หรือแบบ ROM (เบอร์ 8051)
- หน่วยความจำแบบ RAM ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบ Full duplex
- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรออสซิลเลเตอร์ภายใน

2.4.2 **เวลาของการประมวลผลชุดคำสั่ง**

8051 มีวงจรออสซิลเลเตอร์อยู่ภายในสำหรับการสร้างพัลส์ของสัญญาณนาฬิกา ซึ่งจะนำไปเป็นฐานเวลา หรือการกำหนดจังหวะการทำงานของหน่วยการทำงานทั้งหมดให้สอดคล้องกัน

เวลาในการประมวลผลคำสั่งหนึ่งจนเสร็จสิ้นของ 8051 นั้น จะนับเป็นหน่วยของแมชชีนไซเคิล (Machine cycle) โดย 1 แมชชีนไซเคิลจะใช้เวลา 12 คาบเวลาของออสซิลเลอร์ ค่าของแมชชีนไซเคิลนี้จัดว่าเป็นช่วงเวลาที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ซึ่งหากว่าเป็นคำสั่งที่ซับซ้อนมาก ก็จะต้องใช้เวลานาน 2-3 แมชชีนไซเคิล โดยคำสั่งใดใช้เวลาประมวลผลนานเท่าใดนั้นสามารถดูได้จากภาคผนวก ก

การคำนวณหาว่าเวลาที่ใช้ในการทำคำสั่งใดจนเสร็จสิ้น จะต้องดูว่าคำสั่งนั้นใช้จำนวนแมชชีนไซเคิลเป็นเท่าไรในการประมวลผล เวลาที่ใช้จะคำนวณตามสูตร

$$T = \frac{C \times 12}{\text{Crystal Frequency}}$$

โดย C เป็นจำนวนแมชชีนไซเคิลของคำสั่ง และ Crystal Frequency เป็นความถี่ของคริสตอลที่ใช้กับ 8051

ตัวอย่างเช่น คำสั่ง ADD A,R1 ใช้เวลา 3 แมชชีนไซเคิล เมื่อใช้คริสตอล 16 เมกะเฮิรตซ์ จะใช้เวลา 0.75 ไมโครวินาที แต่ถ้าใช้คริสตอล 12 เมกะเฮิรตซ์ จะใช้เวลา 1 ไมโครวินาที แต่โดยปกติแล้วจะใช้ค่าความถี่ของคริสตอลเป็น 11.059 เมกะเฮิรตซ์ เนื่องจากสามารถนำค่าความถี่นี้ไปใช้เป็นฐานเวลาสำหรับการสร้างความถี่ในการรับส่งข้อมูลอนุกรมซึ่งเป็นหน่วยการทำงานหนึ่งภายใน 8051 เอง โดยจะได้ค่าที่ใกล้เคียงกับค่ามาตรฐานคือ 19200, 9600, 4800, 2400, 1200 และ 300 บิต/วินาที

2.4.3 **หน่วยความจำของ** 8051

ไมโครคอนโทรลเลอร์ตระกูล 8051 แยกการจัดการหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน คือ หน่วยความจำโปรแกรม และหน่วยความจำข้อมูล หน่วยความจำทั้งสองนี้มีหน้าที่แตกต่างไปจากกัน และใช้วิธีการอ้างตำแหน่ง สัญญาณการติดต่อแยกออกจากกัน

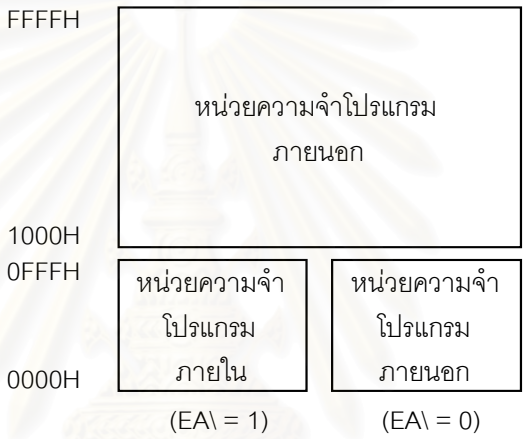
2.4.3.1 **หน่วยความจำโปรแกรม**

หน่วยความจำโปรแกรมของ 8051 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่าง ๆ โดย 8051 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น ดังรูปที่ 2.4 คือ

- หน่วยความจำโปรแกรมภายใน เป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง
- หน่วยความจำโปรแกรมภายนอก เป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

ขนาดของหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ภายในตระกูล 8051 จะแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานลักษณะต่าง ๆ กัน เช่น

- 8051 และ 8052 มีหน่วยความจำแบบ ROM ขนาด 4 และ 8 กิโลไบต์ ตามลำดับ
- มีหน่วยความจำแบบ EPROM ขนาด 4 กิโลไบต์ ข้อมูลที่จัดเก็บภายในนี้สามารถใช้แสงอัลตราไวโอเลตลบและนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง
- 8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซีเลย ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำโปรแกรมภายนอกเสมอ

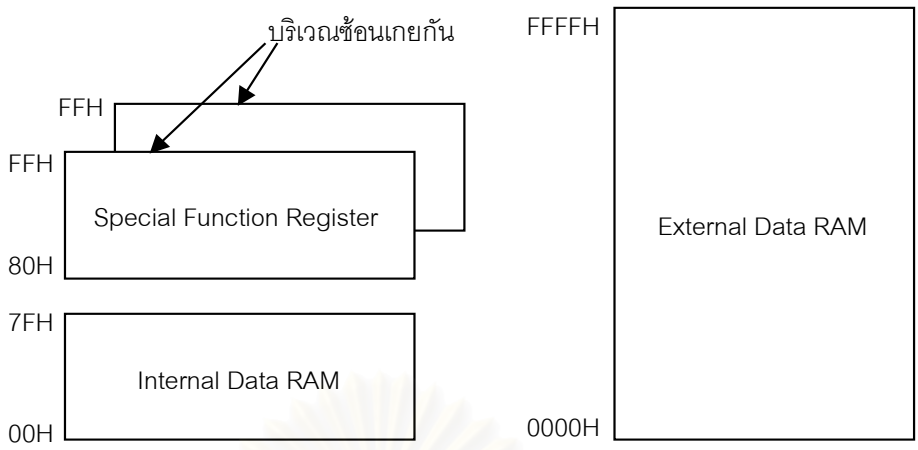


รูปที่ 2.4 หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล 8051

ไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ของตระกูล 8051 นี้ สามารถขยายให้ใช้งานหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้ว การอ้างตำแหน่งที่มีทั้งในหน่วยความจำโปรแกรมภายในและภายนอกนั้น จะต้องทำการควบคุมระดับสัญญาณ EA ในขณะนั้นด้วย

2.4.3.2 **หน่วยความจำข้อมูล**

โดยปกติแล้ว หน่วยความจำข้อมูลจะเป็นหน่วยความจำ RAM ซึ่งสามารถเขียนและอ่านข้อมูลได้ ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว เมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะทำให้ข้อมูลที่เก็บไว้ภายในหน่วยความจำนี้สูญหายไป หน่วยความจำข้อมูลของ 8051 นี้สามารถแบ่งออกได้เป็น 2 ลักษณะ ตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ดังรูปที่ 2.5 คือ



รูปที่ 2.5 หน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ตระกูล 8051

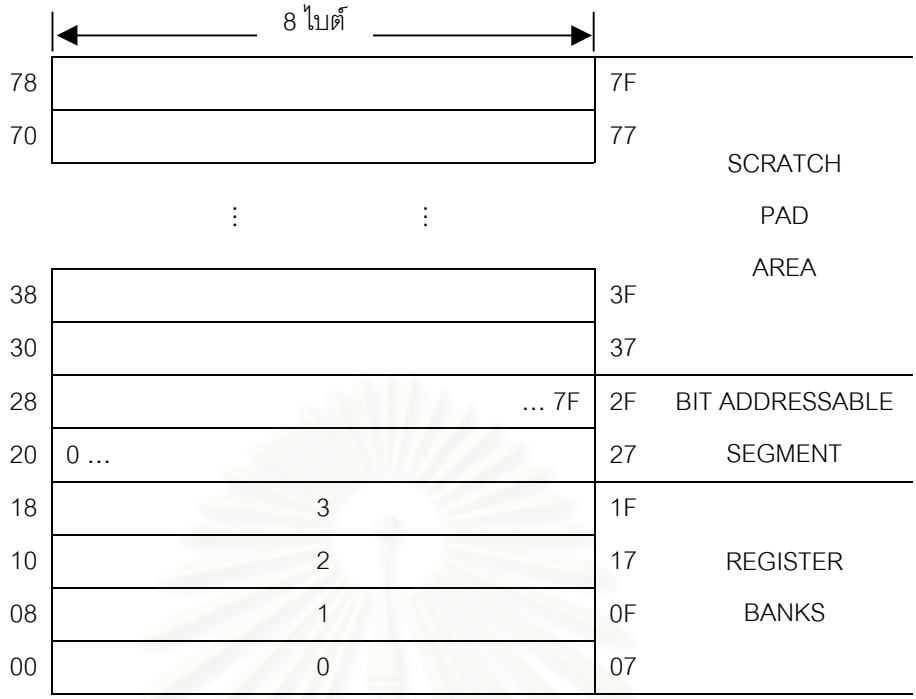
- *หน่วยความจำข้อมูลภายนอก* เป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเข้าไปในวงจร ลักษณะเดียวกันกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรม หน่วยความจำส่วนนี้มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และใช้คำสั่ง MOVX ในการเข้าถึงหน่วยความจำ (ชุดคำสั่งได้กล่าวไว้ในภาคผนวก ก)
- *หน่วยความจำข้อมูลภายใน* เป็น RAM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง ประกอบด้วยหน่วยความจำข้อมูลขนาดแตกต่างกันไปตามเบอร์ เช่น 256 ไบต์ สำหรับเบอร์ 8052 หรือขนาด 128 ไบต์ สำหรับเบอร์ 8051 และรีจิสเตอร์หน้าที่พิเศษ (SFR: Special Function Register)

2.4.3.2.1 **หน่วยความจำ 128 ไบต์แรก**

เป็นหน่วยความจำข้อมูลภายใน อยู่ที่ตำแหน่ง 00H – 7FH ดังรูปที่ 2.6 สามารถเข้าถึงได้โดยใช้การอ้างตำแหน่งทั้งแบบตรงและแบบอ้อม หน่วยความจำส่วนนี้ อาจแบ่งออกได้อีกเป็น 3 ส่วนคือ

- **กลุ่มรีจิสเตอร์** (Register Bank) 0-3

อยู่ที่ตำแหน่ง 00H – 1FH (32 ไบต์) เป็นกลุ่มรีจิสเตอร์ 4 กลุ่ม แต่ละกลุ่มมีรีจิสเตอร์ขนาด 1 ไบต์ จำนวน 8 ตัว ตั้งแต่ R0 ถึง R7 การเรียกใช้รีจิสเตอร์กลุ่มใดนั้น สามารถทำได้โดยการกำหนดค่าบิตภายในรีจิสเตอร์ PSW ดังตารางที่ 2.1



รูปที่ 2.6 หน่วยความจำข้อมูล 128 ไบต์แรกของไมโครคอนโทรลเลอร์ตระกูล 8051

ตารางที่ 2.1 การเลือกกลุ่มรีจิสเตอร์โดยใช้ค่า RS0 กับ RS1 ในรีจิสเตอร์ PSW

RS1	RS0	กลุ่มรีจิสเตอร์	ตำแหน่ง
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

เมื่อเริ่มต้นใช้งาน ค่าจะถูกตั้งไว้ที่กลุ่มรีจิสเตอร์ 0 และค่าตัวชี้สแต็ก (SP: Stack Pointer) จะอยู่ที่ 07H คือ จะเก็บค่าแรกลงในสแต็กที่ตำแหน่ง 08H ซึ่งตรงกับตำแหน่ง R0 ของกลุ่มรีจิสเตอร์ 1 ดังนั้น หากต้องการใช้งานกลุ่มรีจิสเตอร์มากกว่า 1 กลุ่ม จะต้องเปลี่ยนค่า SP ไปยังตำแหน่งอื่นที่ไม่ใช้ในการเก็บข้อมูล

โดยทั่วไป การใช้งานกลุ่มรีจิสเตอร์ จะใช้เฉพาะรีจิสเตอร์ R0 ถึง R7 ในกลุ่มรีจิสเตอร์ 0 เท่านั้น ดังนั้น ตำแหน่งของกลุ่มรีจิสเตอร์อื่นที่เหลือก็สามารถนำมาใช้ในการเก็บข้อมูลปกติโดยการอ้างถึงหมายเลขตำแหน่งนั้น ๆ โดยตรง

- **บริเวณที่สามารถอ้างตำแหน่งแบบบิตได้** (Bit Addressable Area)

อยู่ที่ตำแหน่ง 20H – 2FH (16 ไบต์) เป็นบริเวณที่สามารถอ้างถึงหน่วยความจำได้ทั้งแบบไบต์และแบบบิต โดยถ้าอ้างแบบบิต จะสามารถอ้างได้ตั้งแต่บิต 0 ถึง 7FH (128 บิต)

การอ้างตำแหน่งแบบบิต สามารถอ้างได้อีกแบบหนึ่งคือ อ้างถึงไบต์ 20H ถึง 2FH ก่อน แล้วตามด้วยบิต 0 ถึง 7 เช่น บิต 20.0 หมายถึง บิต 0, บิต 21.7 หมายถึง บิต 0FH

- **บริเวณใช้งานอิสระ** (Scratch Pad Area)

อยู่ที่ตำแหน่ง 30H – 7FH เป็นบริเวณที่สามารถนำไปใช้เก็บข้อมูลต่าง ๆ ได้อย่างอิสระ โดยสามารถอ้างถึงได้เฉพาะแบบไบต์เท่านั้น หากจะใช้บริเวณนี้เป็นสแต็ก จะต้องเว้นที่ไว้ให้เพียงพอเพื่อป้องกันการทำลายข้อมูลในสแต็ก

2.4.3.2.2 หน่วยความจำ 128 ไบต์ถัดไป

เป็นหน่วยความจำข้อมูลภายใน อยู่ที่ตำแหน่ง 80H – FFH สามารถเข้าถึงได้โดยใช้การอ้างตำแหน่งแบบอ้อมเท่านั้น หน่วยความจำส่วนนี้มีเฉพาะบางเบอร์เท่านั้น

สแต็กเป็นการทำงานอย่างหนึ่งที่ใช้การอ้างตำแหน่งแบบอ้อม ดังนั้น หน่วยความจำส่วนนี้อาจใช้เป็นสแต็กได้

2.4.3.2.3 รีจิสเตอร์หน้าที่พิเศษ (SFR: Special Function Register)

เป็นรีจิสเตอร์ที่อยู่ที่ตำแหน่ง 80H – FFH ซึ่งเป็นตำแหน่งเดียวกับหน่วยความจำข้อมูลปกติ แต่เป็นหน่วยความจำคนละบริเวณกัน และสามารถเข้าถึงได้โดยใช้การอ้างตำแหน่งแบบตรงหรือใช้ชื่อของรีจิสเตอร์นั้น ๆ ดัวอย่างเช่น คำสั่ง

```
MOV  80H,#0AAH
```

คือ การส่งค่า 0AAH ไปยังพอร์ต 0 ซึ่งเป็นหนึ่งใน SFR และคำสั่ง

```
MOV  R0,#80H
MOV  @R0,#0BBH
```

คือ การส่งค่า 0BBH ไปยังหน่วยความจำข้อมูลตำแหน่ง 80H

SFR ต่าง ๆ มีชื่อตามตารางที่ 2.2 และ SFR ที่สามารถอ้างถึงแบบบิตได้ จะอยู่ในสดมภ์แรกของรูปที่ 2.7 โดยมีรายละเอียดดังนี้

- *Accumulator (ACC)* เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับ accumulator ของหน่วยประมวลผลทั่วไป
- *B* เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณและหารตัวเลข ในกรณีที่ไม่ใช้ในการคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

ตารางที่ 2.2 รีจิสเตอร์หน้าที่พิเศษในไมโครคอนโทรลเลอร์ตระกูล 8051

Symbol	Name	Address
* ACC	Accumulator	0E0H
* B	B Register	0F0H
* PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
* P0	Port 0	80H
* P1	Port 1	90H
* P2	Port 2	0A0H
* P3	Port 3	0B0H
* IP	Interrupt Priority Control	0B8H
* IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
* TCON	Timer/Counter Control	88H
*+ T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+ TH2	Timer/Counter 2 High Byte	0CDH
+ TL2	Timer/Counter 2 Low Byte	0CCH
+ RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+ RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
* SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable + = 8052 only

- *ตัวนับคำสั่งโปรแกรม (PC: Program Counter)* เป็นรีจิสเตอร์ที่ใช้ชี้ตำแหน่งของหน่วยความจำโปรแกรมที่ต้องนำมาประมวลผลในลำดับถัดไป โดยค่าในรีจิสเตอร์นี้จะเพิ่มขึ้นโดยอัตโนมัติ
- *ตัวชี้สแต็ก (SP: Stack Pointer)* เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของสแต็กสำหรับเก็บข้อมูลต่าง ๆ
- *ตัวชี้ข้อมูล (DPTR: Data Pointer)* เป็นรีจิสเตอร์ขนาด 16 บิต สามารถแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัวคือ DPH กับ DPL เพื่อเก็บค่าตำแหน่งของหน่วยความจำหรืออุปกรณ์ที่ต้องการใช้งานภายในโปรแกรม การใช้

รีจิสเตอร์ DPTR ทำให้หน่วยประมวลผลกลางสามารถใช้การอ้างตำแหน่งแบบอ้อมได้

8 ไบต์

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Bit Addressable

รูปที่ 2.7 ตำแหน่งของรีจิสเตอร์หน้าที่พิเศษในไมโครคอนโทรลเลอร์ตระกูล 8051

- *Program Status Word (PSW)* เป็นรีจิสเตอร์ที่ใช้บอกสถานะการทำงานต่าง ๆ และมีบิตสำหรับการกำหนดเลือกกลุ่มของรีจิสเตอร์ด้วย ดังรูปที่ 2.8

CY	AC	F0	RS1	RS0	OV	—	P

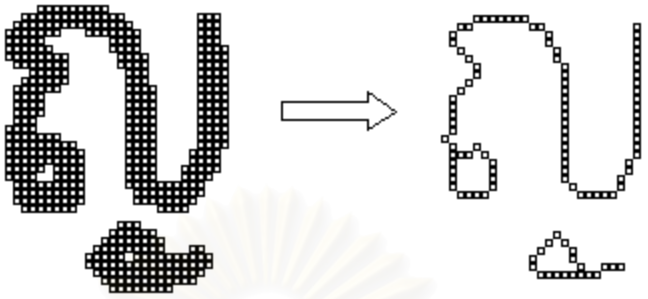
CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry Flag
F0	PSW.5	Flag 0 available to the user for general purpose
RS1	PSW.4	Register Bank selector bit 1
RS0	PSW.3	Register Bank selector bit 0
OV	PSW.2	Overflow Flag
—	PSW.1	User definable flag
P	PSW.0	Parity flag

รูปที่ 2.8 ค่าต่าง ๆ ในรีจิสเตอร์ PSW ในไมโครคอนโทรลเลอร์ตระกูล 8051

นอกจากนี้ ยังมีรีจิสเตอร์อื่น ๆ อีก ดังตารางที่ 2.2 ที่ไม่ได้กล่าวรายละเอียดในที่นี้

2.5 กระบวนการทำโครงร่างภาพ

กระบวนการทำโครงร่างภาพ เป็นการทำให้ความหนาของภาพต้นแบบเปลี่ยนแปลงไป แต่ยังคงเค้าโครงเดิมอยู่ โดยความหนาของภาพจะเหลือเพียงหนึ่งจุดภาพ ดังแสดงในรูปที่ 2.9



(ก) (ข)

รูปที่ 2.9 กระบวนการทำโครงร่างภาพ (ก) ภาพต้นแบบ (ข) ภาพที่ผ่านการทำโครงร่างภาพ

ภาพที่ผ่านกระบวนการทำโครงร่างภาพจะได้คุณลักษณะของจุดต่อในภาพโครงร่าง เช่น จุดปลาย จุดต่อเนื่อง จุดแยกสาม จุดแยกสี่ เป็นต้น

การทำโครงร่างภาพ จะใช้ในการเขียนโปรแกรมเพื่อหาจุดปลายของกลุ่มจุดดำในภาพ ซึ่งเป็นขั้นตอนหนึ่งในการดึงคุณลักษณะเฉพาะตัวของกระดาษในหัวข้อ 4.6.2.2.5

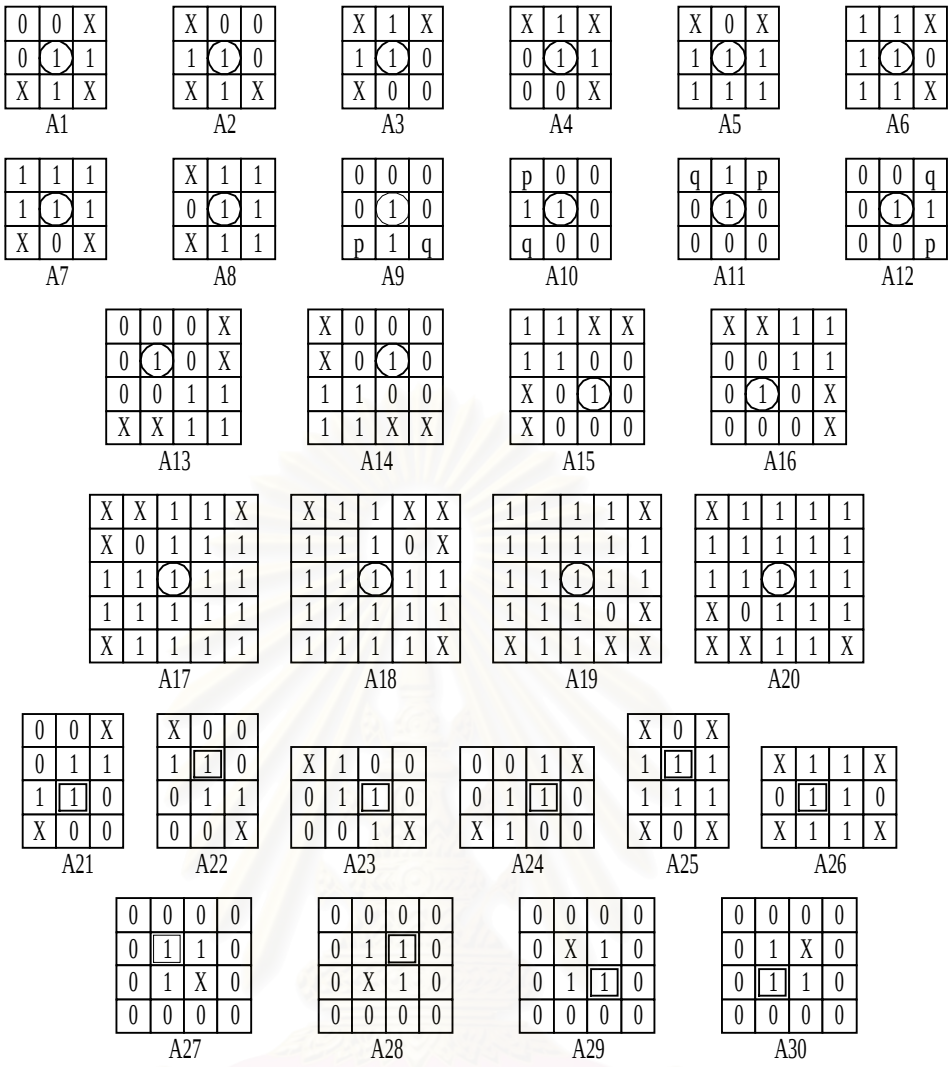
เนื่องจากการทำโครงร่างภาพมีอยู่ด้วยกันหลายวิธี วิธีที่ได้เลือกใช้คือวิธี One-Pass Parallel Thinning [6] ของ Ben K. Jang และ Roland T. Chin เพราะให้รายละเอียดคุณลักษณะของจุดต่อภาพได้ดีเพียงพอ โดยมีแม่แบบ (Template) ที่ใช้ในการทำโครงร่างภาพทั้งหมด 30 แบบ ซึ่งแม่แบบ A1 ถึง A20 เป็นแม่แบบการทำโครงร่างภาพ ในขณะที่ A21 ถึง A30 เป็นแม่แบบการเรียกกลับคืน (Restoring) p และ q เป็นตัวดำเนินการทางตรรกศาสตร์ : p or q = 1 และ X เป็นค่าที่ไม่สนใจ ดังแสดงในรูปที่ 2.10

ระเบียบวิธีการทำโครงร่างภาพ

1. จุดภาพจะถูกทำการตรวจสอบตามแม่แบบการทำโครงร่างภาพทั้งหมด

2. หากจุดภาพที่ถูกตรวจอยู่ในแม่แบบ A1 ถึง A20 จะถูกลบจุดภาพ ในขณะที่จุดภาพอยู่ในแม่แบบ A21 ถึง A30 จะถูกกู้กลับคืนมา

3. เมื่อภาพผ่านกระบวนการทำโครงร่างภาพจะเหลือความกว้างของภาพเพียง 1 จุดภาพ



รูปที่ 2.10 แม่แบบการทำโครงร่างภาพ

2.6 **รหัสแท่ง** Code 128

Code 128 [7 8 9 และ 10] เป็นรหัสแท่งสำหรับตัวอักษรและตัวเลขที่มีความหนาแน่นสูงมาก มีความยาวไม่จำกัด แล้วแต่ความยาวของข้อมูล สามารถเข้ารหัสตัวอักขระแอสกีได้ทั้ง 128 ตัว รหัสแท่งชนิดนี้จะใช้ในขั้นตอนการพิมพ์รหัสลับลงบนบัตรเลือกตั้งในหัวข้อ 4.4.2.4

2.6.1 **ลักษณะของรหัสแท่ง**

โครงสร้างของรหัสแท่ง Code 128 มีลักษณะดังรูปที่ 2.11 ประกอบด้วย

- เขตว่างเปล่า (Quiet zone) บริเวณด้านซ้ายของรหัสแท่ง
- ตัวอักขระเริ่มต้น (Start character)
- ข้อมูลที่เข้ารหัสไว้
- ตัวอักขระตรวจสอบ (Check character)

- ตัวอักขระหยุด (Stop character)
- เขตว่างเปล่า (Quiet zone) บริเวณด้านขวาของรหัสแท่ง

เขตว่างเปล่า (Quiet zone) ควรมีความกว้างอย่างน้อย 10 เท่าของความกว้างมอดูล

รูปที่ 2.11 ลักษณะของรหัสแท่ง Code 128

รหัสของตัวอักขระแต่ละตัวจะประกอบด้วยมอดูล (Module) สีขาวหรือดำรวม 11 มอดูล ยกเว้นตัวอักขระหยุด (Stop character) จะมี 13 มอดูล แถบดำ 3 แถบและช่องว่าง 3 ช่อง ซึ่งวางตัวสลับกันจะถูกสร้างขึ้นจากมอดูล 11 มอดูลนี้ โดยแถบดำแต่ละแถบและช่องว่างแต่ละช่องจะมีความกว้างได้ตั้งแต่ 1 ถึง 4 มอดูล ส่วนตัวอักขระหยุดจะมีแถบดำ 4 แถบและช่องว่าง 3 ช่อง เนื่องจากเป็นตัวสุดท้ายจึงมีแถบดำปิดท้ายอีก 1 แถบ

Code 128 มีชุดของตัวอักขระแตกต่างกัน 3 ชุด คือ ชุด A, ชุด B และ ชุด C แถบดำ 3 แถบและช่องว่าง 3 ช่อง จะถูกสร้างขึ้นให้มีลักษณะต่างกัน 106 แบบ แต่ละแบบจะถูกกำหนดให้กับตัวอักขระที่แตกต่างกันขึ้นกับชุดที่เลือกว่าเป็นชุด A, B หรือ C แต่ละชุดประกอบด้วยตัวอักขระที่แตกต่างกันดังนี้

- <u>ชุด A</u> ประกอบด้วยตัวอักษรตัวพิมพ์ใหญ่ เครื่องหมายวรรคตอน ตัวเลข (รหัสแอสกี 32 ถึง 95) และตัวอักขระควบคุม (Control character) เช่น return, tab (รหัสแอสกี 0 ถึง 31)
- <u>ชุด B</u> ประกอบด้วยตัวอักษรทั้งตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก เครื่องหมายวรรคตอน ตัวเลข (รหัสแอสกี 32 ถึง 127)
- <u>ชุด C</u> ประกอบด้วยตัวเลขเพียงอย่างเดียว โดยมีคู่ของตัวเลขตั้งแต่ 00 ถึง 99 รหัส 1 ตัว สามารถแทนตัวเลขได้ 1 คู่ จึงทำให้รหัสชุดนี้มีความหนาแน่นสูงขึ้นเป็น 2 เท่า

การเลือกชุดแต่ละชุดสามารถทำได้โดยใช้ตัวอักขระเริ่มต้นของชุดนั้น และภายในสัญลักษณ์ยังมีรหัส CODE และ SHIFT ซึ่งสามารถใช้เปลี่ยนชุดไปมาได้ โดยถ้าใช้รหัส CODE จะ

หมายถึงว่า รหัสที่ตามมาทั้งหมดนั้นจะเปลี่ยนชุดเป็นรหัสของชุดดังกล่าว ส่วนรหัส SHIFT นั้น จะเปลี่ยนรหัสของตัวอักขระถัดไปเพียง 1 ตัว ซึ่งจะเป็นการเปลี่ยนจากชุด A เป็นชุด B หรือชุด B เป็นชุด A เท่านั้น

นอกจากนี้ ยังมีรหัสฟังก์ชัน (Function code) FNC1 ถึง FNC4

- FNC1 ใช้สำหรับรหัสแท่ง EAN/UCC 128 โดยใช้ตัวอักขระเริ่มต้น Start C แล้วตามด้วย FNC1 แล้วต่อด้วยรหัสข้อมูลซึ่งใช้การเข้ารหัสเช่นเดียวกับ Code 128
- FNC2 ใช้เพื่อสั่งให้เครื่องอ่านรหัสแท่งรออ่านรหัสแท่งอันต่อไปแล้วนำข้อมูลมาต่อกับข้อมูลที่อ่านได้จากรหัสแท่งนี้
- FNC3 ใช้เพื่อสั่งให้เครื่องอ่านรหัสแท่ง reset
- FNC4 ยังไม่กำหนดให้ใช้กับสิ่งใดโดยเฉพาะ

2.6.2 **การหาค่าตัวอักขระตรวจสอบ**

รหัสของตัวอักขระแต่ละตัวจะมีค่ากำหนดไว้ตั้งแต่ 0 ถึง 105 ค่าเหล่านี้จะนำมาใช้ในการหาตัวอักขระตรวจสอบดังนี้

- นำค่าของรหัสของข้อมูลทุกตัวมาคูณกับตำแหน่งที่อยู่ในรหัสแท่ง โดยตำแหน่งของข้อมูลที่อยู่ซ้ายสุดมีค่าเป็น 1
- นำผลคูณที่ได้มาบวกกัน แล้วบวกกับค่าของตัวอักขระเริ่มต้น
- นำผลบวกที่ได้มาหาค่าเศษจากการหารด้วย 103 แล้วใช้เศษเป็นตัวอักขระตรวจสอบ

ตัวอย่างเช่น ข้อมูลคือ *Code 128* สามารถหาตัวอักขระตรวจสอบได้ดังนี้

รหัส	ตำแหน่ง	ค่า	ผลคูณ
C	1	35	35
o	2	79	158
d	3	68	204
e	4	69	276
5	0	0	
1	6	17	102
2	7	18	126
8	8	24	192
Start B			104
ผลรวม			1197
หารด้วย 103 เหลือเศษ			64

2.6.3 ขนาดของรหัสแท่ง

ความยาวของรหัสแท่งสามารถหาได้จากสมการ

$$L = (11 \cdot C + 35) \cdot X \qquad (2\text{-}15)$$

โดย L คือ ความยาวของรหัสแท่ง (ไม่รวมเขตว่างเปล่า)

C คือ จำนวนรหัสของตัวอักขระ ซึ่งรวมถึงรหัส SHIFT, CODE, FNC ด้วย

X คือ ความกว้างของมอดูล

ความสูงของรหัสแท่งต้องมีค่าอย่างน้อย 15 % ของความยาวของรหัสแท่ง แต่ต้องไม่น้อยกว่า 0.25 นิ้ว

ตารางที่ 2.3 แสดงถึงตัวอักขระในแต่ละชุดที่ถูกกำหนดให้กับรหัส และมีค่าของรหัสซึ่งใช้ในการคำนวณหาตัวอักขระตรวจสอบ นอกจากนี้ ในสดมภ์สุดท้ายมีลักษณะของแถบและช่องว่างของรหัสแต่ละตัวโดยตัวเลขในตารางคือจำนวนมอดูลของแถบดำหรือช่องว่าง

ตารางที่ 2.3 ตารางแสดงลักษณะแถบดำและช่องว่างของรหัสแท่ง Code 128

Value	Code Set A	Code Set B	Code Set C	Bar/Space Pattern B S B S B S
0	SP	SP	00	2 1 2 2 2 2
1	!	!	01	2 2 2 1 2 2
2	"	"	02	2 2 2 2 2 1
3	#	#	03	1 2 1 2 2 3
4	$	$	04	1 2 1 3 2 2
5	%	%	05	1 3 1 2 2 2
6	&	&	06	1 2 2 2 1 3
7	'	'	07	1 2 2 3 1 2
8	(	(	08	1 3 2 2 1 2
9	)	)	09	2 2 1 2 1 3
10	*	*	10	2 2 1 3 1 2
11	+	+	11	2 3 1 2 1 2
12	,	,	12	1 1 2 2 3 2
13	-	-	13	1 2 2 1 3 2
14	.	.	14	1 2 2 2 3 1
15	/	/	15	1 1 3 2 2 2
16	0	0	16	1 2 3 1 2 2
17	1	1	17	1 2 3 2 2 1
18	2	2	18	2 2 3 2 1 1

ตารางที่ 2.3 ตารางแสดงลักษณะแถบดำและช่องว่างของรหัสแท่ง Code 128 (ต่อ)

Value	Code Set A	Code Set B	Code Set C	Bar/Space Pattern B S B S B S
19	3	3	19	2 2 1 1 3 2
20	4	4	20	2 2 1 2 3 1
21	5	5	21	2 1 3 2 1 2
22	6	6	22	2 2 3 1 1 2
23	7	7	23	3 1 2 1 3 1
24	8	8	24	3 1 1 2 2 2
25	9	9	25	3 2 1 1 2 2
26	:	:	26	3 2 1 2 2 1
27	;	;	27	3 1 2 2 1 2
28	<	<	28	3 2 2 1 1 2
29	=	=	29	3 2 2 2 1 1
30	>	>	30	2 1 2 1 2 3
31	?	?	31	2 1 2 3 2 1
32	@	@	32	2 3 2 1 2 1
33	A	A	33	1 1 1 3 2 3
34	B	B	34	1 3 1 1 2 3
35	C	C	35	1 3 1 3 2 1
36	D	D	36	1 1 2 3 1 3
37	E	E	37	1 3 2 1 1 3
38	F	F	38	1 3 2 3 1 1
39	G	G	39	2 1 1 3 1 3
40	H	H	40	2 3 1 1 1 3
41	I	I	41	2 3 1 3 1 1
42	J	J	42	1 1 2 1 3 3
43	K	K	43	1 1 2 3 3 1
44	L	L	44	1 3 2 1 3 1
45	M	M	45	1 1 3 1 2 3
46	N	N	46	1 1 3 3 2 1
47	O	O	47	1 3 3 1 2 1
48	P	P	48	3 1 3 1 2 1
49	Q	Q	49	2 1 1 3 3 1
50	R	R	50	2 3 1 1 3 1
51	S	S	51	2 1 3 1 1 3
52	T	T	52	2 1 3 3 1 1

ตารางที่ 2.3 ตารางแสดงลักษณะแถบดำและช่องว่างของรหัสแท่ง Code 128 (ต่อ)

Value	Code Set A	Code Set B	Code Set C	Bar/Space Pattern B S B S B S
53	U	U	53	2 1 3 1 3 1
54	V	V	54	3 1 1 1 2 3
55	W	W	55	3 1 1 3 2 1
56	X	X	56	3 3 1 1 2 1
57	Y	Y	57	3 1 2 1 1 3
58	Z	Z	58	3 1 2 3 1 1
59	[	[	59	3 3 2 1 1 1
60	\	\	60	3 1 4 1 1 1
61	]	]	61	2 2 1 4 1 1
62	^	^	62	4 3 1 1 1 1
63	_	_	63	1 1 1 2 2 4
64	NUL	`	64	1 1 1 4 2 2
65	SOH	a	65	1 2 1 1 2 4
66	STX	b	66	1 2 1 4 2 1
67	ETX	c	67	1 4 1 1 2 2
68	EOT	d	68	1 4 1 2 2 1
69	ENQ	e	69	1 1 2 2 1 4
70	ACK	f	70	1 1 2 4 1 2
71	BEL	g	71	1 2 2 1 1 4
72	BS	h	72	1 2 2 4 1 1
73	HT	i	73	1 4 2 1 1 2
74	LF	j	74	1 4 2 2 1 1
75	VT	k	75	2 4 1 2 1 1
76	FF	l	76	2 2 1 1 1 4
77	CR	m	77	4 1 3 1 1 1
78	SO	n	78	2 4 1 1 1 2
79	SI	o	79	1 3 4 1 1 1
80	DLE	p	80	1 1 1 2 4 2
81	DC1	q	81	1 2 1 1 4 2
82	DC2	r	82	1 2 1 2 4 1
83	DC3	s	83	1 1 4 2 1 2
84	DC4	t	84	1 2 4 1 1 2
85	NAK	u	85	1 2 4 2 1 1
86	SYN	v	86	4 1 1 2 1 2

ตารางที่ 2.3 ตารางแสดงลักษณะแถบดำและช่องว่างของรหัสแท่ง Code 128 (ต่อ)

Value	Code Set A	Code Set B	Code Set C	Bar/Space Pattern B S B S B S
87	ETB	w	87	4 2 1 1 1 2
88	CAN	x	88	4 2 1 2 1 1
89	EM	y	89	2 1 2 1 4 1
90	SUB	z	90	2 1 4 1 2 1
91	ESC	{	91	4 1 2 1 2 1
92	FS	|	92	1 1 1 1 4 3
93	GS	}	93	1 1 1 3 4 1
94	RS	~	94	1 3 1 1 4 1
95	US	DEL	95	1 1 4 1 1 3
96	FNC 3	FNC 3	96	1 1 4 3 1 1
97	FNC 2	FNC 2	97	4 1 1 1 1 3
98	SHIFT	SHIFT	98	4 1 1 3 1 1
99	CODE C	CODE C	99	1 1 3 1 4 1
100	CODE B	FNC 4	CODE B	1 1 4 1 3 1
101	FNC 4	CODE A	CODE A	3 1 1 1 4 1
102	FNC 1	FNC 1	FNC 1	4 1 1 1 3 1
103	Start A	Start A	Start A	2 1 1 4 1 2
104	Start B	Start B	Start B	2 1 1 2 1 4
105	Start C	Start C	Start C	2 1 1 2 3 2
106	Stop	Stop	Stop	2 3 3 1 1 1 2

บทที่ 3
การออกแบบระบบเลือกตั้ง

เนื้อหาในบทนี้ ได้กล่าวถึงคุณสมบัติของระบบเลือกตั้งที่ต้องการ ลักษณะของบัตรเลือกตั้ง กุญแจสำหรับการเข้าและถอดรหัสลับบัตรเลือกตั้ง และการปรับเปลี่ยนขั้นตอนการทำงานตั้งแต่การพิมพ์บัตรเลือกตั้ง การลงคะแนนเสียงเลือกตั้ง จนถึงการตรวจสอบความถูกต้องของบัตรเลือกตั้ง และสุดท้ายคือ การปลอมแปลงและทุจริตต่างๆ ในกระบวนการเลือกตั้งที่สามารถตรวจพบได้ ดังมีรายละเอียดดังนี้

3.1 คุณสมบัติของระบบเลือกตั้ง

การเลือกตั้งที่ใช้กันอยู่ในปัจจุบันเป็นการเลือกตั้งโดยใช้บัตรกระดาษ ซึ่งมีปัญหาที่สำคัญ คือ สามารถปลอมแปลงได้ง่าย และเมื่อเกิดการปลอมแปลงขึ้นก็ไม่สามารถตรวจสอบแยกแยะได้ว่าบัตรใดเป็นบัตรจริง บัตรใดเป็นบัตรปลอม ดังนั้น จึงได้นำการเข้ารหัสลับมาใช้กับบัตรเลือกตั้ง และปรับเปลี่ยนกระบวนการเลือกตั้งต่าง ๆ ให้สอดคล้องกัน เพื่อที่จะทำให้บัตรปลอมแปลงได้ยากขึ้นในราคาที่ถูกและสามารถตรวจสอบได้ โดยมีคุณสมบัติที่ต้องการคือ

3.1.1 การป้องกันการปลอมบัตรเลือกตั้ง

เพื่อป้องกันการพิมพ์บัตรเลือกตั้งปลอม บัตรเลือกตั้งแต่ละใบจะมีรหัสลับอยู่บนบัตรซึ่งไม่ซ้ำกัน รหัสลับนี้จะถูกสร้างขึ้นโดยใช้กุญแจส่วนตัวซึ่งคณะกรรมการการเลือกตั้งจะเก็บไว้เป็นความลับ บุคคลอื่นจะไม่สามารถสร้างรหัสลับนี้ได้เพราะไม่รู้กุญแจส่วนตัว แต่จะสามารถตรวจสอบความถูกต้องของรหัสลับบนบัตรเลือกตั้งได้โดยใช้กุญแจสาธารณะที่คู่กันกับกุญแจส่วนตัวนั้น โดยคณะกรรมการการเลือกตั้งจะประกาศกุญแจสาธารณะให้รู้ทั่วกันสำหรับการเลือกตั้งแต่ละครั้ง

และเพื่อป้องกันการคัดลอกรหัสลับจากบัตรเลือกตั้งจริงไปยังบัตรเลือกตั้งปลอม รหัสลับบนบัตรเลือกตั้งแต่ละใบจะมีข้อมูลคุณลักษณะเฉพาะตัวของกระดาษใบที่ใช้พิมพ์บัตรนั้น ๆ ซึ่งบัตรแต่ละใบจะมีคุณลักษณะเฉพาะตัวของกระดาษแตกต่างกัน

3.1.2 การทราบที่มาของบัตรเลือกตั้ง

รหัสลับที่พิมพ์ลงบนบัตรเลือกตั้งแต่ละใบ นอกจากจะมีคุณลักษณะเฉพาะตัวของกระดาษแล้ว ยังจะมีข้อมูลที่มาของบัตรเลือกตั้งใบนั้นด้วย ทำให้ทราบถึงที่มาของบัตรได้ถ้ามีการตรวจสอบในภายหลัง

3.1.3 การตรวจสอบจำนวนบัตรเลือกตั้งที่ใช้ลงคะแนน

บัตรเลือกตั้งแต่ละใบจะมีต้นขั้วเพื่อใช้ในการตรวจสอบจำนวนบัตรลงคะแนนที่ถูกฉีกออกไป เพราะบัตรที่ถูกฉีกออกไปแล้วไม่สามารถนำมาต่อคืนกับต้นขั้วได้

3.1.4 การยืนยันตัวผู้มาใช้สิทธิ์เลือกตั้ง

เพื่อเป็นการยืนยันตัวผู้มาใช้สิทธิ์เลือกตั้ง ลายนิ้วมือของผู้มาใช้สิทธิ์จะถูกใช้เป็นรหัสเฉพาะตัวของบุคคลผู้มาใช้สิทธิ์ เนื่องจากสามารถทำปลอมให้มีจำนวนมากโดยไม่ซ้ำกันได้ยาก และถ้าเป็นลายนิ้วมือของผู้กระทำผิด ก็จะเป็นหลักฐานโยงถึงตัวได้ด้วย

3.1.5 ความสะดวกในการตรวจสอบบัตรเลือกตั้ง

การตรวจสอบบัตรเลือกตั้ง ไม่สามารถทำกับบัตรทุกใบได้ เนื่องจากการเลือกตั้งในแต่ละครั้ง มีบัตรเป็นจำนวนมาก และการปลอมบัตรเพียงไม่กี่ใบก็ไม่อาจทำให้ชนะการเลือกตั้งได้ จึงเป็นการเหมาะสมกว่าที่จะใช้การสุ่มบัตรเลือกตั้งมาตรวจสอบแทนพร้อมกับมีการบันทึกรายงานอย่างเป็นระบบในทุกระดับชั้น

นอกจากนี้ ก่อนการบันทึกรหัสลับลงบนบัตร รหัสลับจะถูกแปลงให้อยู่ในรูปแบบที่สะดวกต่อการตรวจสอบความถูกต้องด้วยอุปกรณ์ที่ใช้ในการตรวจสอบ เพื่อไม่ให้การตรวจสอบความถูกต้องเป็นไปด้วยความยากลำบาก ซับซ้อน หรือเสียเวลาจนเกินไป

3.2 ลักษณะของบัตรเลือกตั้ง

3.2.1 ส่วนประกอบของบัตรเลือกตั้ง

บัตรเลือกตั้งที่พิมพ์ออกมาแต่ละใบ จะประกอบด้วยส่วนสำคัญ 2 ส่วน คือ

- *ต้นขั้ว* ประกอบด้วยส่วนสำคัญ ๆ คือ
  - พื้นที่สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ
  - รหัสแท่งของข้อมูลประจำบัตรเลือกตั้งที่ผ่านการเข้ารหัสลับแล้ว
  - ช่องสำหรับพิมพ์ลายนิ้วมือ
- *บัตรลงคะแนน* ประกอบด้วยส่วนสำคัญ ๆ คือ
  - พื้นที่สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ
  - รหัสแท่งของข้อมูลประจำบัตรเลือกตั้งที่ผ่านการเข้ารหัสลับแล้ว
  - ช่องสำหรับลงคะแนน

3.2.2 ข้อมูลประจำบัตรเลือกตั้ง

บัตรเลือกตั้งแต่ละใบจะมีข้อมูลประจำบัตรที่ไม่ซ้ำกัน ประกอบด้วย

- *ข้อมูลที่มาของบัตร* เพื่อที่จะสามารถระบุได้ว่าบัตรใบนั้นมาจากที่ใด ถ้ามีการตรวจสอบในภายหลัง และเป็นการป้องกันการใช้บัตรข้ามพื้นที่ด้วย
- *คุณลักษณะเฉพาะตัวของกระดาษแผ่นที่ใช้พิมพ์บัตรเลือกตั้งใบนั้น ๆ* เพื่อป้องกันการคัดลอกชุดข้อมูลที่ผ่านการเข้ารหัสลับจากบัตรเลือกตั้งจริงไปยังบัตรเลือกตั้งปลอม ข้อมูลประจำบัตรเลือกตั้งแต่ละใบจึงต้องมีคุณลักษณะเฉพาะตัวของกระดาษใบนั้น ๆ โดยคุณลักษณะเฉพาะตัวของกระดาษแต่ละใบจะแตกต่างกัน ทำให้ไม่สามารถคัดลอกข้อมูลจากบัตรเลือกตั้งจริงไปยังบัตรเลือกตั้งปลอมได้

ข้อมูลเหล่านี้จะถูกนำมาเข้ารหัสลับ แล้วแปลงเป็นรหัสแท่ง (Barcode) ก่อนทำการพิมพ์ลงบนบัตรเลือกตั้งทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน

3.2.3 กระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

ในการผลิตกระดาษที่ใช้พิมพ์บัตรเลือกตั้งแต่ละใบนั้น จะมีการแทรกวัตถุชิ้นเล็ก ๆ เช่น เม็ดทราย หรือเส้นใยสี ลงไปในเนื้อกระดาษด้วย โดยกระดาษแต่ละใบจะมีการวางตัวของวัตถุเล็ก ๆ นี้ไม่เหมือนกัน ทำให้กระดาษแต่ละใบมีลักษณะไม่เหมือนกัน คือ สามารถระบุได้ว่า บัตรใบหนึ่งต่างจากบัตรอีกใบหนึ่งอย่างไร ด้วยลักษณะของเนื้อกระดาษที่แตกต่างกันนี้เอง ทำให้กระดาษแต่ละใบมีข้อมูลที่เป็นคุณลักษณะเฉพาะตัวของมัน ซึ่งต่างจากกระดาษใบอื่น ๆ โดยเราจะดึงข้อมูลที่เป็นคุณลักษณะเฉพาะตัวนี้ออกมาใช้เป็นส่วนหนึ่งของข้อมูลประจำบัตรเลือกตั้ง หากต้องการปลอมบัตรเลือกตั้งใบใด ก็ต้องสร้างกระดาษให้มีลักษณะเหมือนกับกระดาษของบัตรเลือกตั้งจริงใบนั้น ซึ่งทำได้ยาก เพราะการผลิตกระดาษโดยบังคับให้วัตถุเล็ก ๆ วางตัวอยู่ในตำแหน่งที่ต้องการทำได้ยาก และต้องปลอมในจำนวนที่มากพอที่จะชนะการเลือกตั้งได้

3.2.4 รหัสเฉพาะตัวสำหรับบุคคล

บนต้นขั้วบัตรเลือกตั้ง จะมีช่องสำหรับให้ผู้มาใช้สิทธิ์เลือกตั้งพิมพ์ลายนิ้วมือของตนเพื่อเป็นการยืนยันตัวผู้มาใช้สิทธิ์เลือกตั้ง เพราะลายนิ้วมือเป็นข้อมูลเฉพาะตัวสำหรับแต่ละคน สามารถปลอมให้มีจำนวนมากโดยไม่ซ้ำกันได้ยาก และถ้าเป็นลายนิ้วมือของผู้กระทำผิด ก็จะเป็นหลักฐานโยงถึงตัวได้ด้วย

ตำแหน่งสำหรับพิมพ์ลายนิ้วมือต้องอยู่บนต้นขั้วบัตรเลือกตั้ง เนื่องจาก

- เป็นการรับประกันว่าผู้ใช้สิทธิ์ได้พิมพ์ลายนิ้วมือลงบนกระดาษซึ่งเป็นแผ่นเดียวกันกับบัตรลงคะแนนก่อนจะฉีกออกจากกัน คือ หากบัตรลงคะแนนเป็นบัตรปลอม ต้นขั้วก็เป็นต้นขั้วปลอมด้วย แสดงว่าผู้ใช้สิทธิ์ได้ลงคะแนนบนบัตรลงคะแนนปลอมและพิมพ์ลายนิ้วมือลงบนต้นขั้วปลอม ซึ่งต้นขั้วปลอมนี้สามารถพบได้ในขั้นตอนการตรวจสอบ

บัตรเลือกตั้งที่เขตเลือกตั้งก่อนการนับคะแนน ทำให้ทราบว่ามีการทุจริตเกิดขึ้น หรือหากมีการสับเปลี่ยนต้นขั้ว นำต้นขั้วจริงมาให้เจ้าหน้าที่ที่เขตเลือกตั้งทำการตรวจสอบ ก็ต้องว่าจ้างคนมาพิมพ์ลายนิ้วมือลงบนต้นขั้วจริง ซึ่งหากเป็นดังที่กล่าวมานี้ ก็จะเป็นหลักฐานที่จะโยงไปสู่ผู้กระทำผิดได้

- ไม่สามารถตรวจสอบได้ว่าผู้ใช้สิทธิ์คนใดลงคะแนนให้กับใคร

3.2.5 ระบบต้นขั้ว

บัตรเลือกตั้งแต่ละใบ จะมีต้นขั้วเพื่อใช้ในการตรวจสอบจำนวนบัตรลงคะแนนที่ถูกฉีกออกไป โดยจำนวนบัตรลงคะแนนที่อยู่ในหีบเลือกตั้งจะต้องตรงกับจำนวนต้นขั้วที่บัตรลงคะแนนถูกฉีกออกไปแล้ว

ทั้งต้นขั้วและบัตรลงคะแนนจะมีรหัสลับของข้อมูลประจำบัตรเลือกตั้งพิมพ์ไว้ และเพื่อรับประกันว่าการเลือกตั้งเป็นความลับจริง รหัสลับบนต้นขั้วจะต้องไม่มีหมายเลขบัตรหรือข้อมูลที่สามารถโยงจับคู่กันกับบัตรลงคะแนนได้หลังจากฉีกออกจากกันแล้ว เพื่อให้ไม่สามารถโยงลายพิมพ์นิ้วมือบนต้นขั้วเข้ากับการเลือกบนบัตรลงคะแนนได้

สาเหตุที่ต้นขั้วต้องมีการพิมพ์ข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ได้เข้ารหัสลับไว้ด้วย ก็เพื่อป้องกันการทำต้นขั้วปลอม เนื่องจากว่า หากต้นขั้วสามารถปลอมได้ คือไม่มีข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ เมื่อเจ้าหน้าที่นำบัตรเลือกตั้งปลอม ซึ่งหมายถึง ทั้งต้นขั้วและตัวบัตรลงคะแนนที่ยังติดกันอยู่เป็นบัตรปลอม มาให้ผู้มาใช้สิทธิ์เลือกตั้งลงคะแนนที่หน่วยเลือกตั้ง เมื่อเสร็จสิ้นการลงคะแนน จึงนำบัตรลงคะแนนปลอมที่ผู้มาใช้สิทธิ์ได้ลงคะแนนไว้ไปสับเปลี่ยนกับบัตรลงคะแนนจริงที่ได้ลงคะแนนให้กับผู้สมัครคนหนึ่งคนใดไว้ และก่อนการนับคะแนนที่เขตเลือกตั้ง เจ้าหน้าที่ที่เขตเลือกตั้งก็จะทำการตรวจสอบบัตรลงคะแนน พบว่าเป็นบัตรจริง และตรวจสอบต้นขั้ว พบว่า เป็นลายพิมพ์นิ้วมือที่ไม่มีลักษณะซ้ำกัน จึงสรุปว่า ไม่มีการทุจริตเกิดขึ้น ทั้งที่จริงแล้ว ได้มีการสับเปลี่ยนบัตรลงคะแนนเกิดขึ้น แต่หากต้นขั้วไม่สามารถปลอมได้ คือมีข้อมูลคุณลักษณะเฉพาะตัวของกระดาษอยู่ด้วย ก่อนการนับคะแนนที่เขตเลือกตั้ง เจ้าหน้าที่จะทำการตรวจสอบต้นขั้วก็จะทราบว่าเป็นต้นขั้วปลอม หรือหากมีการสับเปลี่ยนต้นขั้ว นำต้นขั้วจริงมาให้เจ้าหน้าที่ที่เขตเลือกตั้งทำการตรวจสอบ ก็ต้องว่าจ้างคนมาพิมพ์ลายนิ้วมือลงบนต้นขั้วจริง ซึ่งหากเป็นดังที่กล่าวมานี้ ก็จะเป็นหลักฐานที่จะโยงไปสู่ผู้กระทำผิดได้

3.3 กุญแจสำหรับการเข้าและถอดรหัสลับ

การเข้ารหัสลับที่ใช้กับบัตรเลือกตั้งเป็นการเข้ารหัสลับแบบกุญแจสาธารณะโดยวิธี RSA ซึ่งเป็นวิธีการเข้ารหัสลับที่ได้รับความนิยมอย่างแพร่หลาย

3.3.1 **การสร้างกุญแจ**

ก่อนที่จะทำการเข้าหรือถอดรหัสลับได้ จะต้องมีกุญแจที่ใช้ในการเข้าหรือถอดรหัสลับเสียก่อน โดยกุญแจที่สร้างขึ้นนี้จะใช้เฉพาะการเลือกตั้งครั้งหนึ่ง ๆ เพียงครั้งเดียวเท่านั้น

ความยาวของกุญแจที่ใช้ คือ 512 บิต ซึ่งเป็นความยาวที่เพียงพอต่อการตรวจสอบบัตรเลือกตั้งหลังจากการเลือกตั้งผ่านไปแล้วอย่างมากที่สุดประมาณ 6 เดือน

3.3.2 **ส่วนประกอบของกุญแจ**

กุญแจสำหรับการเข้าและถอดรหัสลับมีอยู่ 2 ชุด คือ กุญแจส่วนตัวกับกุญแจสาธารณะ

3.3.2.1 **กุญแจส่วนตัว**

คณะกรรมการการเลือกตั้งจะใช้กุญแจนี้สำหรับการเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง กุญแจส่วนตัว จะถูกแบ่งออกเป็นส่วน ๆ ตามจำนวนคณะกรรมการการเลือกตั้งที่มีอยู่ แล้วให้กรรมการแต่ละท่านเก็บรักษากุญแจของตนเองไว้เป็นความลับ

กุญแจส่วนตัวจะถูกเข้ารหัสลับไว้โดยใช้การเข้ารหัสลับแบบกุญแจลับ จึงต้องมีรหัสผ่านเมื่อต้องการนำกุญแจมาใช้ เพื่อป้องกันไม่ให้ผู้อื่นนำเอากุญแจส่วนตัวไปใช้ได้

3.3.2.2 **กุญแจสาธารณะ**

กุญแจสาธารณะ จะประกาศให้สาธารณชนรับรู้ได้ทั่วกันสำหรับการเลือกตั้งในแต่ละครั้ง เพื่อใช้ในการตรวจสอบความถูกต้องของบัตรเลือกตั้ง

3.4 **การพิมพ์รหัสลับบนบัตรเลือกตั้ง**

3.4.1 **ขั้นตอนการพิมพ์รหัสลับบนบัตรเลือกตั้ง**

ในการพิมพ์รหัสลับลงบนบัตรเลือกตั้ง จะมีขั้นตอนต่าง ๆ ดังนี้

- *การดึงคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง*

นำบัตรเลือกตั้งมาสแกนในบริเวณพื้นที่ใช้สำหรับดึงข้อมูล แล้วดึงข้อมูลคุณลักษณะของวัตถุชิ้นเล็ก ๆ ที่ฝังตัวอยู่ในกระดาษที่ใช้พิมพ์บัตรเลือกตั้งใบนั้น ทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน

- *การเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง*

เมื่อได้คุณสมบัติเฉพาะตัวของกระดาษ ก็จะนำมารวมกับข้อมูลที่มาของบัตร กลายเป็นข้อมูลประจำบัตรเลือกตั้งใบนั้น ๆ จากนั้น คณะกรรมการการเลือกตั้งจะนำข้อมูลประจำบัตรเลือกตั้งแต่ละใบมาเข้ารหัสลับด้วยวิธี RSA โดยใช้กุญแจส่วนตัวและรหัสผ่านของกรรมการทุกท่าน

- *การบันทึกรหัสลับลงบนบัตรเลือกตั้ง*

นำรหัสลับที่ได้จากขั้นตอนที่แล้วทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน มาเปลี่ยนเป็นรหัสแท่ง แล้วจึงพิมพ์รหัสแท่งนั้นลงบนบัตรเลือกตั้งแต่ละใบ เพื่อให้การตรวจสอบความถูกต้องของบัตรทำได้สะดวก โดยใช้เครื่องอ่านรหัสแท่ง ซึ่งอ่านได้แม่นยำและรวดเร็วกว่าการพิมพ์ชุดตัวเลขลงบนอุปกรณ์ตรวจสอบความถูกต้องเอง

3.4.2 การป้องกันการทุจริตในการพิมพ์บัตรเลือกตั้ง

จะเห็นว่า ในการพิมพ์รหัสลับลงบนบัตรเลือกตั้งแต่ละใบจะต้องนำกุญแจส่วนตัวแต่ละส่วนของกรรมการแต่ละท่านมาใช้ร่วมกันเพื่อเป็นการตรวจสอบกันเองของคณะกรรมการ หากขาดกุญแจไปเพียงส่วนใดส่วนหนึ่งก็จะไม่สามารถเข้ารหัสลับได้ และทุกคนสามารถตรวจสอบความถูกต้องของบัตรเลือกตั้งได้โดยใช้กุญแจสาธารณะที่คู่กันกับกุญแจส่วนตัวมาถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง

ดังนั้น จะมีเพียงคณะกรรมการชุดนี้เท่านั้นที่สามารถสร้างรหัสที่จะใช้พิมพ์บนบัตรเลือกตั้งแต่ละใบได้ เนื่องจากมีเพียงคณะกรรมการชุดดังกล่าวที่มีกุญแจส่วนตัวที่ใช้ในการเข้ารหัสลับ บุคคลอื่นจะไม่สามารถเดาสุ่มหรือสร้างชุดตัวเลขขึ้นมาให้เป็นรหัสที่ถูกต้องได้ โดยไม่รู้กุญแจส่วนตัวทั้งหมด และการรู้กุญแจสาธารณะจะไม่สามารถคำนวณหากุญแจส่วนตัวออกมาได้โดยง่าย นอกจากการทดลองเดาสุ่มไปเรื่อย ๆ ซึ่งใช้เวลานานมาก โดยจะใช้เวลานานเท่าไรนั้น ขึ้นกับความยาวของกุญแจที่ใช้ [11]

3.5 การลงคะแนนเสียงเลือกตั้ง

3.5.1 การพิมพ์ลายนิ้วมือ

ผู้มาใช้สิทธิ์เลือกตั้งทุกคนจะต้องพิมพ์ลายนิ้วมือลงบนต้นขั้วบัตรก่อนฉีกบัตรออกมาลงคะแนน เพื่อเป็นการยืนยันตัวผู้มาใช้สิทธิ์เลือกตั้ง

3.5.2 การรับบัตรลงคะแนนจากเจ้าหน้าที่

ในการออกเสียงเลือกตั้ง ห้ามผู้มาใช้สิทธิ์เลือกตั้งรับบัตรลงคะแนนที่ฉีกเอาไว้แล้วเด็ดขาด เนื่องจากอาจเป็นบัตรลงคะแนนที่ไม่ได้ติดมากับต้นขั้วที่ผู้มาใช้สิทธิ์ได้พิมพ์ลายนิ้วมือไว้ หรือเป็นบัตรลงคะแนนปลอมนั่นเอง ส่วนบัตรลงคะแนนจริงก็จะมีการลงคะแนนให้กับผู้สมัครคนใดคนหนึ่ง แล้วทำการสลับกับบัตรลงคะแนนปลอมก่อนการนับคะแนน ทำให้ไม่สามารถตรวจพบการทุจริตที่เกิดขึ้นได้

3.6 การตรวจสอบความถูกต้องของบัตรเลือกตั้ง

3.6.1 การสุ่มตรวจสอบบัตรเลือกตั้ง

เนื่องจากในการเลือกตั้งแต่ละครั้ง ต้องใช้บัตรเลือกตั้งเป็นจำนวนมาก การตรวจสอบความถูกต้องของบัตรเลือกตั้งทุกใบจึงยากที่จะกระทำได้ โดยเฉพาะอย่างยิ่ง การตรวจสอบความถูกต้องที่หน่วยเลือกตั้ง เพราะไม่มีอุปกรณ์การตรวจที่ดีเหมือนกับที่เขตเลือกตั้งซึ่งเป็นที่นับคะแนน นอกจากนี้ การปลอมบัตรเลือกตั้งเพียงไม่กี่ใบ ก็ไม่อาจทำให้ชนะการเลือกตั้งได้ จึงเป็นการเหมาะสมกว่าที่จะใช้การสุ่มบัตรเลือกตั้งบางส่วนมาตรวจสอบความถูกต้องแทน พร้อมกับมีการบันทึกผลการสุ่มตรวจสอบเป็นลายลักษณ์อักษรสำหรับเจ้าหน้าที่ในระดับปฏิบัติการทั้งที่หน่วยเลือกตั้งและที่เขตเลือกตั้ง เพื่อให้สามารถตรวจสอบการทุจริตของเจ้าหน้าที่ดำเนินการที่เกี่ยวข้องได้ และขณะเดียวกันก็ให้การปกป้องเจ้าหน้าที่ที่ปฏิบัติหน้าที่โดยสุจริตตามขั้นตอนที่กำหนดด้วย โดยจำนวนบัตรที่จะสุ่มเพื่อตรวจนั้น ต้องไม่มากเกินไปจนทำให้การปฏิบัติงานเป็นไปด้วยความไม่สะดวก และไม่น้อยเกินไปจนทำให้คุ้มที่จะเสี่ยงต่อการถูกจับได้

การสุ่มบัตรเลือกตั้งเพื่อทำการตรวจสอบ จะต้องไม่รู้ล่วงหน้าว่าจะได้บัตรใบใด และต้องไม่สามารถบังคับให้สุ่มได้บัตรใบที่ต้องการได้ เพื่อป้องกันการจัดเตรียมบัตรเลือกตั้งจริงสำหรับการตรวจสอบไว้ และปลอมบัตรเลือกตั้งใบอื่นที่ไม่ต้องผ่านการตรวจสอบ ดังนั้น ในการกำหนดว่าจะตรวจสอบบัตรใบใด จะต้องทำในขณะทำการตรวจสอบบัตรเลือกตั้งนั้น โดยให้อุปกรณ์การตรวจสอบบัตรเลือกตั้งสร้างชุดตัวเลขขึ้นมาเพื่อกำหนดบัตรที่จะตรวจสอบ แล้วทำการตรวจสอบในทันที

3.6.2 การตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้ง

3.6.2.1 อุปกรณ์ที่ใช้ในการตรวจสอบบัตรเลือกตั้ง

ที่หน่วยเลือกตั้ง จะใช้เครื่องถอดรหัสลับขนาดเล็กที่พัฒนาขึ้นเพื่อใช้ในการถอดรหัสลับ เนื่องจากมีราคาไม่สูงเกินไป เพราะต้องแจกจ่ายอุปกรณ์นี้ไปยังทุกหน่วยเลือกตั้งเพื่อให้เจ้าหน้าที่ประจำหน่วยใช้ในการสุ่มตรวจสอบบัตรเลือกตั้งที่ตนเองได้รับ โดยจะต้องสามารถอ่านรหัสแท่งบนบัตรเลือกตั้งได้ด้วย

3.6.2.2 ขั้นตอนการตรวจสอบบัตรเลือกตั้ง

ในการตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้ง จะมีขั้นตอนต่าง ๆ ดังนี้

- *การสุ่มบัตรเลือกตั้ง*

ก่อนการลงคะแนนเสียงเลือกตั้ง เจ้าหน้าที่ประจำหน่วยเลือกตั้งจะทำการสุ่มบัตรเลือกตั้งที่ได้รับเพื่อทำการตรวจสอบ โดยเครื่องถอดรหัสลับขนาดเล็กจะเป็นตัวกำหนดว่า เจ้าหน้าที่จะต้องตรวจสอบบัตรใบใด

- *การอ่านรหัสลับบนบัตรเลือกตั้ง*

นำบัตรเลือกตั้งที่สุ่มได้มาอ่านรหัสแท่งในส่วนของต้นขั้ว โดยใช้เครื่องถอดรหัสลับขนาดเล็กซึ่งสามารถอ่านรหัสแท่งได้ด้วย จะได้รหัสลับของต้นขั้วบัตรเลือกตั้งใบนั้น

- *การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง*

นำรหัสลับที่ได้มาถอดรหัสลับโดยวิธี RSA ด้วยกุญแจสาธารณะที่ได้ประกาศไว้สำหรับการเลือกตั้งครั้งนั้น โดยใช้เครื่องถอดรหัสลับขนาดเล็ก จะได้ข้อมูลประจำบัตรเลือกตั้งของต้นขั้วบัตรใบนั้น

- *การบันทึกผลการสุ่มตรวจ*

เมื่อตรวจสอบเสร็จ 1 ใบ ก็จะบันทึกข้อมูลที่มาของบัตรและค่าแฮช (Hash value) ที่อ่านได้จากเครื่องลงในบันทึกผลการสุ่มตรวจและเซ็นชื่อกำกับไว้ เพื่อเป็นการยืนยันว่าเจ้าหน้าที่ได้ทำการตรวจสอบจริง โดยสามารถตรวจสอบต้นขั้วบัตรดังกล่าวได้ในภายหลัง

- *กรณีที่พบความผิดปกติ*

หากพบความผิดปกติ คือ การถอดรหัสลับไม่ถูกต้อง หรือข้อมูลที่มาของบัตรไม่ถูกต้อง ให้ตรวจสอบบัตรทุกใบที่ได้รับ และทำบันทึกไว้เป็นหลักฐาน

จะเห็นว่าการตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้งไม่มีการตรวจสอบคุณลักษณะของกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง เนื่องจาก

- เจ้าหน้าที่จะเสียเวลาในการตรวจสอบ เพราะไม่มีเครื่องมือที่ดีพอ เช่น สแกนเนอร์ ไมโครคอมพิวเตอร์ จึงต้องใช้การตรวจสอบด้วยตาเปล่า แล้วใช้วิจารณญาณของเจ้าหน้าที่ในการตัดสิน
- ที่เขตเลือกตั้ง ก่อนการนับคะแนน จะมีการตรวจสอบบัตรเลือกตั้งอีกครั้ง ซึ่งจะตรวจสอบคุณลักษณะของกระดาษทั้งในส่วนของต้นขั้วและบัตรลงคะแนน

3.6.3 การตรวจสอบบัตรเลือกตั้งที่เขตเลือกตั้งสำหรับนับคะแนนรวม

3.6.3.1 ขั้นตอนการตรวจสอบบัตรเลือกตั้ง

ในการตรวจสอบบัตรเลือกตั้งที่เขตเลือกตั้ง จะมีขั้นตอนต่าง ๆ ดังนี้

- *การตรวจสอบจำนวนบัตร*

เมื่อนำบัตรเลือกตั้งจากหน่วยเลือกตั้งมาถึงเขตเลือกตั้ง ก่อนการเทรวมกับบัตรเลือกตั้งในหน่วยอื่น เจ้าหน้าที่ประจำเขตเลือกตั้งจะตรวจสอบจำนวนต้นขั้วกับจำนวนบัตรลงคะแนนในหีบเลือกตั้งว่าเท่ากันหรือไม่

- *การตรวจสอบลายพิมพ์นิ้วมือของผู้มาใช้สิทธิ์*

เป็นการตรวจเฉพาะที่ต้นขั้วของบัตรเท่านั้น โดยตรวจสอบว่าลายพิมพ์นิ้วมือมีอยู่บนต้นขั้วบัตรทุกใบหรือไม่ และลายพิมพ์นิ้วมือมีลักษณะซ้ำกันบนบัตรนับสิบ ๆ ใบจนสังเกตเห็นได้หรือไม่

- *การสุ่มบัตรเลือกตั้ง*

ทำการสุ่มต้นขั้วและบัตรลงคะแนนของหน่วยเลือกตั้งนั้นเพื่อมาตรวจ

- *การอ่านรหัสลับบนบัตรเลือกตั้ง*

นำบัตรเลือกตั้ง (หมายถึงทั้งต้นขั้วและบัตรลงคะแนน) มาอ่านรหัสแท่งโดยใช้เครื่องอ่านรหัสแท่ง จะได้รหัสลับของบัตรเลือกตั้งใบนั้น

- *การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง*

นำรหัสลับที่ได้มาถอดรหัสลับโดยวิธี RSA ด้วยกุญแจสาธารณะที่ได้ประกาศไว้สำหรับการเลือกตั้งครั้งนั้น จะได้ข้อมูลประจำบัตรเลือกตั้งใบนั้น ซึ่งประกอบด้วย ข้อมูลที่มาของบัตร ทำให้ทราบที่มาของบัตรได้ และข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ ซึ่งจะใช้ตรวจสอบกระดาษที่ใช้พิมพ์บัตรในขั้นตอนถัดไป เมื่อได้ข้อมูลที่มาของบัตร ก็จะทำการตรวจสอบว่าบัตรลงคะแนนกับต้นขั้วมาจากหน่วยเลือกตั้งเดียวกันหรือไม่

- *การตรวจสอบคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง*

ดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษใบนั้น เช่นเดียวกับขั้นตอนหนึ่งในหัวข้อ 3.4.1 แล้วนำมาเปรียบเทียบกับข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ได้จากขั้นตอนที่แล้ว

- *การบันทึกผลการสุ่มตรวจ*

เมื่อตรวจสอบเสร็จ 1 ใบ ก็จะบันทึกข้อมูลที่มาของบัตรและค่าแฮช (Hash value) ที่ได้จากเครื่องลงในฐานข้อมูลผ่านทางเครื่องไมโครคอมพิวเตอร์ เพื่อเป็นการยืนยันว่าเจ้าหน้าที่ได้ทำการตรวจสอบจริง โดยสามารถตรวจสอบความถูกต้องได้ในภายหลัง

- *กรณีที่พบความผิดปกติ*

หากพบความผิดปกติเกิดขึ้น ให้ตรวจสอบบัตรทุกใบที่ได้รับในหน่วยเลือกตั้งนั้นและทำบันทึกไว้เป็นหลักฐาน

3.7 การปลอมแปลงและทุจริตในกระบวนการเลือกตั้งที่สามารถตรวจพบได้

1. **การนำบัตรเลือกตั้งปลอมมาส่งที่หน่วยเลือกตั้งโดยเจ้าหน้าที่หน่วยไม่รู้เห็น**
   - ปลอมต้นขั้วโดยการสร้างรหัสลับปลอม ตรวจสอบพบได้ที่หน่วยเลือกตั้ง
   - ปลอมต้นขั้วโดยการคัดลอกรหัสลับจากบัตรจริง ตรวจสอบพบได้เฉพาะที่เขตเลือกตั้ง เพราะที่หน่วยเลือกตั้งไม่มีการตรวจสอบคุณลักษณะเฉพาะตัวของกระดาษ โดยเมื่อเสร็จสิ้นการลงคะแนนแล้ว
     - สับเปลี่ยนบัตรลงคะแนนปลอมที่ผู้มาใช้สิทธิ์ได้ลงคะแนนไว้กับบัตรลงคะแนนจริงที่ได้ลงคะแนนให้กับผู้สมัครคนใดคนหนึ่ง แล้วส่งไปยังเขตเลือกตั้ง ตรวจพบต้นขั้วปลอมที่เขตเลือกตั้ง
     - สับเปลี่ยนบัตรลงคะแนนปลอมที่ผู้มาใช้สิทธิ์ได้ลงคะแนนไว้กับบัตรลงคะแนนจริงที่ได้ลงคะแนนให้กับผู้สมัครคนใดคนหนึ่ง และต้นขั้วปลอมที่มีลายพิมพ์นิ้วมือของผู้มาใช้สิทธิ์กับต้นขั้วจริงที่ได้ว่าจ้างคนมาพิมพ์ลายนิ้วมือลงไป แล้วส่งไปยังเขตเลือกตั้ง เมื่อตรวจสอบที่เขตเลือกตั้งพบว่าต้นขั้วมีลายพิมพ์นิ้วมือที่มีลักษณะซ้ำกัน เพราะไม่สามารถสร้างลายพิมพ์นิ้วมือให้แตกต่างกันโดยการหาคนจำนวนมากมาพิมพ์ลายนิ้วมือในเวลาที่จำกัดได้ ลายพิมพ์นิ้วมือนี้จะเป็นหลักฐานที่จะโยงไปสู่ผู้กระทำผิด
2. **การนำบัตรเลือกตั้งปลอมมาให้ผู้ใช้สิทธิ์ลงคะแนนที่หน่วยเลือกตั้งโดยเจ้าหน้าที่หน่วยร่วมทุจริตด้วย**

เป็นการปลอมโดยการสร้างรหัสลับปลอม ซึ่งสามารถตรวจพบได้ที่หน่วยเลือกตั้ง แต่เจ้าหน้าที่หน่วยร่วมทุจริตโดยนำบัตรปลอมมาให้ผู้มาใช้สิทธิ์ลงคะแนน และเมื่อเสร็จสิ้นการลงคะแนนก็สับเปลี่ยนบัตรลงคะแนน หรือสับเปลี่ยนทั้งต้นขั้วและบัตรลงคะแนนเหมือนกับข้อ 1 ซึ่งสามารถตรวจพบการทุจริตได้ที่เขตเลือกตั้งก่อนการนับคะแนน และสามารถตรวจพบการทุจริตของเจ้าหน้าที่ประจำหน่วยเลือกตั้งได้โดยตรวจสอบบัตรเลือกตั้งที่ได้ทำการตรวจที่หน่วยเลือกตั้งแล้ว จากนั้นนำผลมาเปรียบเทียบกับบันทึกผลการสุ่มตรวจบัตรเลือกตั้งที่หน่วยเลือกตั้ง จะพบว่าค่าแฮชที่ได้ไม่ตรงกัน

3. **การเปลี่ยนบัตรลงคะแนนก่อนถึงเขตเลือกตั้ง**

มีการสับเปลี่ยนบัตรลงคะแนนจริงที่ผู้มาใช้สิทธิ์ได้ลงคะแนนไว้กับบัตรลงคะแนนที่ได้ลงคะแนนให้กับผู้สมัครคนใดคนหนึ่ง ซึ่งอาจมาจาก

- นำบัตรลงคะแนนจริงมาจากหน่วยเลือกตั้งอื่น แล้วเปลี่ยนกับบัตรลงคะแนนในหีบเลือกตั้ง เมื่อตรวจสอบที่เขตเลือกตั้ง จะพบว่าต้นขั้วกับบัตรลงคะแนนไม่ได้มาจากหน่วยเลือกตั้งเดียวกัน

- นำบัตรลงคะแนนปลอมมาเปลี่ยนกับบัตรลงคะแนนในหีบเลือกตั้ง ซึ่งอาจเป็นการปลอมโดยการคัดลอกรหัสลับจากบัตรจริง หรือปลอมโดยการสร้างรหัสลับปลอม ก็สามารถตรวจพบบัตรลงคะแนนปลอมได้ที่เขตเลือกตั้ง

4. **การเพิ่มจำนวนบัตรลงคะแนนในหีบเลือกตั้งก่อนถึงเขตเลือกตั้ง**

มีการใส่บัตรลงคะแนนที่ได้ลงคะแนนให้กับผู้สมัครคนใดคนหนึ่งเพิ่มเข้าไปในหีบเลือกตั้ง โดยบัตรลงคะแนน อาจมาจาก

- นำบัตรลงคะแนนจริงมาจากหน่วยเลือกตั้งอื่น แล้วใส่ลงในหีบเลือกตั้ง เมื่อตรวจสอบที่เขตเลือกตั้ง จะพบว่าต้นขั้วกับบัตรลงคะแนนไม่ได้มาจากหน่วยเลือกตั้งเดียวกัน
- นำบัตรลงคะแนนปลอมใส่ในหีบเลือกตั้ง เมื่อตรวจสอบที่เขตเลือกตั้งพบบัตรลงคะแนนปลอมและจำนวนต้นขั้วไม่ตรงกับจำนวนบัตรลงคะแนนในหีบเลือกตั้ง
- ฉีกบัตรลงคะแนนจริงออกจากต้นขั้วแล้วใส่ในหีบเลือกตั้งและแก้ไขจำนวนผู้มาใช้สิทธิ์เลือกตั้ง ตรวจสอบที่เขตเลือกตั้งพบว่าต้นขั้วบางใบไม่มีลายพิมพ์นิ้วมือ
- ฉีกบัตรลงคะแนนจริงออกจากต้นขั้วแล้วใส่ในหีบเลือกตั้งและพิมพ์ลายนิ้วมือบนต้นขั้วพร้อมแก้ไขจำนวนผู้มาใช้สิทธิ์เลือกตั้ง ตรวจสอบที่เขตเลือกตั้งพบว่าต้นขั้วมีลายพิมพ์นิ้วมือที่มีลักษณะซ้ำกัน เพราะไม่สามารถสร้างลายพิมพ์นิ้วมือให้แตกต่างกันโดยการหาคนจำนวนมากมาพิมพ์ลายนิ้วมือในเวลาที่จำกัดได้

5. **การลดจำนวนบัตรลงคะแนนในหีบเลือกตั้งก่อนถึงเขตเลือกตั้ง**

มีการนำบัตรลงคะแนนบางส่วนที่ผู้มาใช้สิทธิ์ได้ลงคะแนนไว้ออกมาจากหีบเลือกตั้ง ตรวจสอบที่เขตเลือกตั้งพบว่าจำนวนต้นขั้วไม่ตรงกับจำนวนบัตรลงคะแนนในหีบเลือกตั้ง เนื่องจากบัตรลงคะแนนที่ฉีกออกไปแล้วไม่สามารถนำมาต่อกลับคืนกับต้นขั้วได้ และต้นขั้วมีลายพิมพ์นิ้วมือของผู้มาใช้สิทธิ์พิมพ์อยู่

6. **การส่งบัตรของหน่วยเลือกตั้งอื่น หรือบัตรของการเลือกตั้งครั้งอื่น**

ตรวจพบได้ที่หน่วยเลือกตั้ง

บทที่ 4
ต้นแบบระบบเลือกตั้ง

ในบทนี้ ได้กล่าวถึงต้นแบบของระบบเลือกตั้งตามที่ได้ออกแบบไว้ในบทที่ 3 ซึ่งประกอบด้วย ลักษณะของบัตรเลือกตั้ง กุญแจในการเข้าและถอดรหัสลับบัตรเลือกตั้ง ขั้นตอนการทำงานในการพิมพ์รหัสลับบนบัตรเลือกตั้ง การตรวจสอบบัตรเลือกตั้ง และโปรแกรมที่ได้พัฒนาขึ้นเพื่อใช้กับต้นแบบระบบเลือกตั้ง ซึ่งมีรายละเอียดดังนี้

4.1 ต้นแบบระบบเลือกตั้ง

1. ลักษณะของบัตรเลือกตั้ง

บัตรเลือกตั้งที่ใช้เป็นบัตรกระดาษคล้ายกับบัตรเลือกตั้งที่ใช้กันอยู่ในปัจจุบัน แต่มีลักษณะที่แตกต่างกันคือ มีการเพิ่มพื้นที่ที่ใช้สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้งทั้งต้นขั้วและบัตรลงคะแนน มีการพิมพ์รหัสลับลงบนบัตรในรูปของรหัสแท่ง และกระดาษที่ใช้พิมพ์บัตรจะมีวัตถุลักษณะเป็นเส้นฝังอยู่

2. กุญแจสำหรับการเข้าและถอดรหัสลับ

การสร้างกุญแจที่ใช้เข้าและถอดรหัสลับด้วยวิธี RSA สามารถทำได้โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ และเก็บเป็นแฟ้มข้อมูล

3. การพิมพ์รหัสลับบนบัตรเลือกตั้ง

ในการพิมพ์บัตรเลือกตั้ง จะมีการพิมพ์รหัสลับลงบนบัตรเลือกตั้งโดยใช้สแกนเนอร์เพื่อสแกนกระดาษที่ใช้พิมพ์บัตร แล้วใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์เพื่อดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษ นำมารวมกับข้อมูลที่มาของบัตรเลือกตั้ง แล้วนำไปเข้ารหัสลับโดยใช้กุญแจที่ได้สร้างขึ้น จากนั้นจึงนำไปเปลี่ยนเป็นรหัสแท่งและพิมพ์ลงบนบัตรเลือกตั้ง

4. การสุ่มตรวจสอบความถูกต้องของบัตรเลือกตั้ง

ที่หน่วยเลือกตั้ง ซึ่งเป็นที่ลงคะแนน จะใช้โปรแกรมที่เขียนขึ้นบนไมโครคอนโทรลเลอร์ตระกูล 8051 เพื่อถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง แล้วตรวจสอบที่มาของบัตร

ที่เขตเลือกตั้ง ซึ่งเป็นที่นับคะแนน จะใช้สแกนเนอร์เพื่อสแกนบัตรที่ต้องการตรวจสอบ แล้วใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์เพื่อถอดรหัสลับ ตรวจสอบที่มาของบัตรและข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ นอกจากนี้ยังมีการตรวจสอบลายพิมพ์นิ้วมือและจำนวนต้นขั้วด้วย

4.2 ลักษณะของบัตรเลือกตั้ง

4.2.1 ส่วนประกอบของบัตรเลือกตั้ง

บัตรเลือกตั้งที่พิมพ์ออกมาจะมีลักษณะเป็นเล่ม ในแต่ละเล่มมีบัตรเลือกตั้งจำนวน 100 ใบ แต่ละใบ จะมีขนาด A3 ลักษณะดังรูปที่ 4.1 และตัวอย่างบัตรเลือกตั้งซึ่งพิมพ์ลงบนตัวอย่างกระดาษที่สามารถใช้ในการพิมพ์บัตรเลือกตั้งได้ ได้แสดงไว้ในภาคผนวก ข

4.2.2 ข้อมูลประจำบัตรเลือกตั้ง

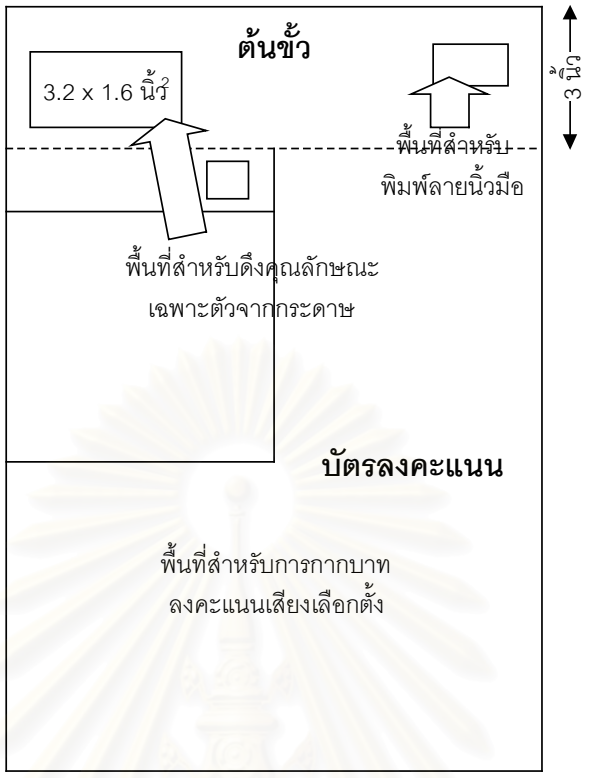
ชุดข้อมูลประจำบัตรเลือกตั้งแต่ละใบก่อนเข้ารหัสลับ ประกอบด้วย

- *ข้อมูลที่มาของบัตร* เป็นเลข BCD ขนาด 11 ไบต์ ประกอบด้วย
  - *วันเลือกตั้ง* ขนาด 4 ไบต์ ประกอบด้วย วันที่ ขนาด 1 ไบต์ เดือน ขนาด 1 ไบต์ และ ปี ขนาด 2 ไบต์ เช่น 25092001 หมายถึง วันที่ 25 กันยายน ค.ศ.2001
  - *รหัสเขตเลือกตั้ง* ขนาด 2 ไบต์
  - *หมายเลขบัตรเลือกตั้ง* ขนาด 4 ไบต์
  - *ไบต์สำรอง* ขนาด 1 ไบต์
- *ข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง* เป็นเลขฐาน 16 มีขนาดไม่เกิน 50 ไบต์

ข้อมูลเหล่านี้จะถูกนำมาเข้ารหัสลับ แล้วแปลงเป็นรหัสแท่ง (Barcode) ก่อนทำการพิมพ์ลงบนบัตรเลือกตั้งทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน โดยข้อมูลประจำบัตรเลือกตั้งของทั้งสองส่วนจะมีข้อแตกต่างกันเล็กน้อย คือ หมายเลขบัตรเลือกตั้งของต้นขั้วจะมีลักษณะเรียงกันไป ส่วนหมายเลขบัตรเลือกตั้งไบต์สุดท้ายของตัวบัตรลงคะแนนจะมีลักษณะสุ่ม เพื่อให้ไม่สามารถโยงจับคู่ต้นขั้วกับบัตรลงคะแนนได้หลังจากฉีกออกจากกันแล้ว

4.2.3 กระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

กระดาษที่ใช้พิมพ์บัตรเลือกตั้งแต่ละใบ จะมีวัตถุชิ้นเล็ก ๆ แทรกตัวอยู่ โดยกระดาษแต่ละใบจะมีการวางตัวของวัตถุเหล่านี้ไม่เหมือนกัน สำหรับงานวิจัยนี้ ได้เลือกกระดาษที่มีวัตถุลักษณะเป็นเส้นฝังอยู่ มีความยาวไม่มากนัก โดยสามารถหาซื้อได้ตามร้านค้าทั่วไป ดังรูปที่ 4.2



(ก)

(ข)

รูปที่ 4.1 ตัวอย่างบัตรเลือกตั้ง (ก) ด้านหน้า (ข) ด้านหลัง

รูปที่ 4.2 ตัวอย่างกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

4.3 กุญแจสำหรับการเข้าและถอดรหัสลับ

การเข้ารหัสลับที่ใช้กับบัตรเลือกตั้งเป็นการเข้ารหัสลับแบบกุญแจสาธารณะโดยวิธี RSA

4.3.1 การสร้างกุญแจ

กุญแจจะถูกสร้างขึ้นด้วยเครื่องไมโครคอมพิวเตอร์ ซึ่งจะได้กุญแจมา 2 ชุด คือ กุญแจส่วนตัวและกุญแจสาธารณะ กุญแจที่จะสร้างมีความยาว 512 บิต

กุญแจจะถูกสร้างขึ้นเพื่อใช้เฉพาะการเลือกตั้งครั้งนั้นเพียงครั้งเดียวเท่านั้น โดยกุญแจส่วนตัวจะให้กรรมการการเลือกตั้งเก็บไว้ ส่วนกุญแจสาธารณะจะประกาศให้ทราบโดยทั่วไป

4.3.2 การบันทึกแฟ้มข้อมูลกุญแจสาธารณะ

กุญแจที่สร้างขึ้นสามารถบันทึกเป็นแฟ้มข้อมูลบนเครื่องไมโครคอมพิวเตอร์ได้ โดยกุญแจสาธารณะจะมีนามสกุลเป็น pub

4.3.3 การบันทึกแฟ้มข้อมูลกุญแจส่วนตัว

กุญแจส่วนตัวจะมีนามสกุลเป็น prv โดยกุญแจส่วนตัวจะมีลักษณะพิเศษคือ สามารถถูกแบ่งออกเป็นส่วน ๆ ได้ตามจำนวนคณะกรรมการการเลือกตั้งที่มีอยู่ นอกจากนี้ กุญแจส่วนตัวยังต้องมีรหัสผ่าน เพื่อป้องกันไม่ให้ผู้อื่นนำเอากุญแจส่วนตัวไปใช้ได้

กุญแจส่วนตัวที่บันทึกจะมีองค์ประกอบสาธารณะรวมอยู่ด้วย ดังนั้น จึงสามารถนำกุญแจส่วนตัวมาใช้ในการถอดรหัสลับได้ด้วย

4.4 การพิมพ์รหัสลับบนบัตรเลือกตั้ง

กระดาษที่ใช้พิมพ์บัตรเลือกตั้งจะมีวัตถุลักษณะเป็นเส้นฝังอยู่ ดังรูปที่ 4.2 และมีการพิมพ์กรอบไว้ทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน เพื่อใช้เป็นพื้นที่สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษ โดยกรอบที่ใช้มีขนาด 3.2x1.6 ตารางนิ้ว

4.4.1 อุปกรณ์ที่ใช้ในการพิมพ์รหัสลับบนบัตรเลือกตั้ง

- *สแกนเนอร์* สำหรับการสแกนบัตรเลือกตั้ง
- *ไมโครคอมพิวเตอร์* สำหรับการดึงคุณลักษณะเฉพาะตัวจากกระดาษ การเข้ารหัสลับ การเปลี่ยนรหัสลับเป็นรหัสแท่ง
- *เครื่องพิมพ์* สำหรับการพิมพ์รหัสแท่งลงบนบัตรเลือกตั้ง

4.4.2 ขั้นตอนการพิมพ์รหัสลับบนบัตรเลือกตั้ง

ในการพิมพ์รหัสลับลงบนบัตรเลือกตั้ง จะเริ่มด้วยการสแกนบัตรเลือกตั้งเก็บเป็นแฟ้มข้อมูล แล้วจึงเปิดแฟ้มข้อมูลเพื่อดึงคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง จากนั้นจึงนำคุณลักษณะที่ได้มารวมกับข้อมูลที่มาของบัตร นำไปเข้ารหัสลับ แล้วจึงบันทึกรหัสลับที่ได้ลงบนบัตรเลือกตั้ง ซึ่งมีรายละเอียดดังนี้

4.4.2.1 การเก็บภาพกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

ขั้นตอนการทำงานมีดังนี้

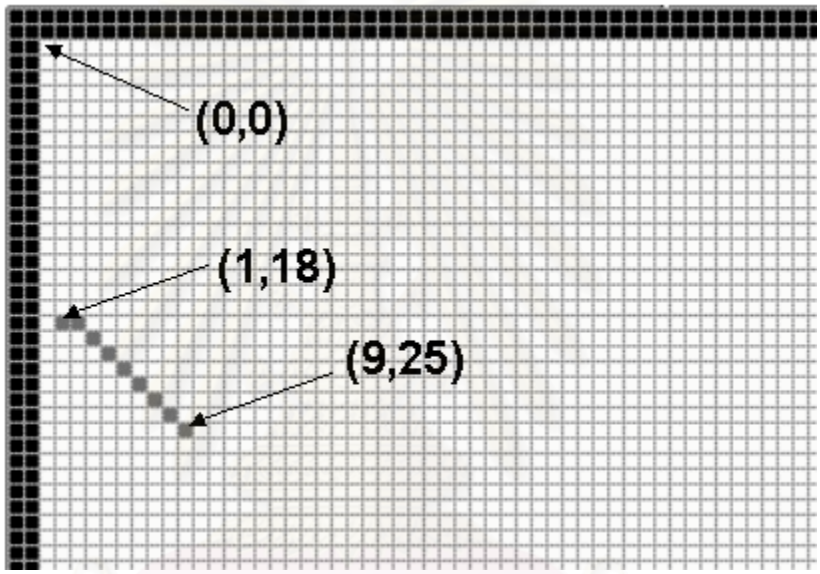
- สแกนบัตรเลือกตั้งเฉพาะในกรอบที่ทำเครื่องหมายไว้ให้เป็นพื้นที่สำหรับดึงข้อมูลจากกระดาษโดยใช้สแกนเนอร์
- เก็บภาพที่สแกนได้เป็นแฟ้มเข้าเครื่องคอมพิวเตอร์ในรูปแบบ bitmap (bmp) แบบสเกลสีเทา (gray-scale) โดยไม่ใช้การเข้ารหัสแบบ RLE ความละเอียด 100 จุดต่อนิ้ว (dpi) ดังรูปที่ 4.3

รูปที่ 4.3 ภาพที่เก็บเข้าเครื่องคอมพิวเตอร์เพื่อใช้ในการดึงคุณลักษณะเฉพาะตัว

4.4.2.2 การดึงคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

คุณลักษณะเฉพาะตัวของกระดาษที่ใช้ในงานวิจัยนี้ คือ ค่าพิกัดของจุดปลายของวัตถุที่ฝังอยู่ในเนื้อกระดาษ ดังรูปที่ 4.4 การดึงคุณลักษณะเฉพาะตัวจากกระดาษ สามารถทำได้โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ โดยมีขั้นตอนการทำงานดังนี้

- อ่านภาพกระดาษที่ใช้พิมพ์บัตรเลือกตั้งจากแฟ้มข้อมูลที่เก็บไว้
- หาค่าพิกัดของจุดปลายของวัตถุที่ฝังอยู่ในเนื้อกระดาษจากภาพ เพื่อใช้เป็นคุณลักษณะเฉพาะตัวของกระดาษ ข้อมูลที่ได้จะเป็นเลขฐาน 16 ความยาวไม่เกิน 50 ไบต์



รูปที่ 4.4 พิกัดของจุดปลายของวัตถุที่ฝังอยู่ในกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง

4.4.2.3 การเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง

เมื่อได้คุณสมบัติเฉพาะตัวของกระดาษ ก็จะนำมาต่อท้ายข้อมูลที่มาของบัตร กลายเป็นข้อมูลประจำบัตรเลือกตั้งใบนั้น ๆ ขนาดไม่เกิน 61 ไบต์ จากนั้น คณะกรรมการการเลือกตั้งจะนำข้อมูลประจำบัตรเลือกตั้งแต่ละใบมาเข้ารหัสลับโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ ซึ่งมีขั้นตอนดังนี้

- อ่านแฟ้มข้อมูลกุญแจส่วนตัวของกรรมการแต่ละท่านจากแฟ้มข้อมูลนามสกุล prv ซึ่งต้องใช้รหัสผ่านของกรรมการแต่ละท่านด้วย
- เมื่อได้ข้อมูลกุญแจส่วนตัวจากกรรมการครบทุกท่านแล้ว จึงนำกุญแจส่วนตัวที่ได้ไปเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้งด้วยวิธี RSA จะได้รหัสลับที่ต้องการซึ่งอยู่ในรูปเลขฐาน 16 ขนาด 64 ไบต์

4.4.2.4 การบันทึกรหัสลับลงบนบัตรเลือกตั้ง

นำรหัสลับที่ได้ทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนนมาเปลี่ยนเป็นรหัสแท่ง แล้วพิมพ์ลงบนบัตรเลือกตั้งโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ รหัสแท่งที่ใช้เป็น Code 128 ชุด B ดังที่ได้กล่าวไว้ในหัวข้อ 2.6

4.5 การตรวจสอบความถูกต้องของบัตรเลือกตั้ง

4.5.1 การสุ่มตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้ง

4.5.1.1 อุปกรณ์ที่ใช้ในการตรวจสอบบัตรเลือกตั้ง

ที่หน่วยเลือกตั้ง จะใช้เครื่องถอดรหัสลับขนาดเล็กที่พัฒนาขึ้นด้วยไมโครคอนโทรลเลอร์ตระกูล 8051 ในการถอดรหัสลับ โดยสามารถอ่านรหัสแท่งบนบัตรเลือกตั้งได้ด้วย

4.5.1.2 ขั้นตอนการสุ่มตรวจสอบบัตรเลือกตั้ง

ในการสุ่มตรวจสอบบัตรเลือกตั้งที่หน่วยเลือกตั้ง จะมีขั้นตอนต่าง ๆ ดังนี้

4.5.1.2.1 การสุ่มบัตรเลือกตั้ง

ก่อนการลงคะแนนเสียงเลือกตั้ง เจ้าหน้าที่ประจำหน่วยเลือกตั้งจะทำการสุ่มบัตรเลือกตั้งที่ได้รับเพื่อทำการตรวจสอบ โดยสุ่มมา 1 ใบ จากบัตรเลือกตั้ง 1 เล่ม หรือ 100 ใบ

4.5.1.2.2 การอ่านรหัสลับบนบัตรเลือกตั้ง

นำบัตรเลือกตั้งมาอ่านรหัสแท่งในส่วนของต้นขั้ว โดยใช้เครื่องถอดรหัสลับขนาดเล็กซึ่งสามารถอ่านรหัสแท่งได้ด้วย จะได้รหัสลับของต้นขั้วบัตรเลือกตั้งใบนั้นขนาด 64 ไบต์

4.5.1.2.3 การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง

นำรหัสลับที่ได้มาถอดรหัสลับตามขั้นตอนต่อไปนี้

- อ่านข้อมูลกุญแจสาธารณะที่ได้ประกาศไว้สำหรับการเลือกตั้งครั้งนั้น โดยอ่านมาจากรหัสแท่ง ด้วยเครื่องถอดรหัสลับขนาดเล็ก
- นำกุญแจสาธารณะที่ได้มาถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งด้วยวิธี RSA โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องถอดรหัสลับขนาดเล็ก จะได้ข้อมูลประจำบัตรเลือกตั้งของต้นขั้วใบนั้น ขนาดไม่เกิน 61 ไบต์

4.5.1.2.4 **การบันทึกผลการสุ่มตรวจ**

เมื่อตรวจสอบเสร็จ 1 ใบ ก็จะบันทึกข้อมูลที่มาของบัตรและค่าแฮช (Hash value) ที่อ่านได้จากเครื่องลงในบันทึกผลการสุ่มตรวจและเซ็นชื่อกำกับไว้

4.5.1.2.5 **กรณีที่พบความผิดปกติ**

หากพบความผิดปกติคือ การถอดรหัสลับไม่ถูกต้องหรือข้อมูลที่มาของบัตรไม่ถูกต้อง ให้ตรวจสอบบัตรทุกใบที่ได้รับ และทำบันทึกไว้เป็นหลักฐาน

4.5.2 **การสุ่มตรวจสอบบัตรเลือกตั้งที่เขตเลือกตั้งสำหรับนับคะแนนรวม**

4.5.2.1 **อุปกรณ์ที่ใช้ในการตรวจสอบบัตรเลือกตั้ง**

ที่เขตเลือกตั้งซึ่งเป็นที่นับคะแนน จะใช้อุปกรณ์ต่อไปนี้ในการตรวจสอบบัตรเลือกตั้ง

- *สแกนเนอร์* สำหรับการสแกนบัตรเลือกตั้ง
- *เครื่องอ่านรหัสแท่ง* สำหรับการอ่านรหัสแท่งบนบัตรเลือกตั้ง
- *ไมโครคอมพิวเตอร์* สำหรับการถอดรหัสลับและการตรวจสอบคุณลักษณะเฉพาะตัวของกระดาษ

4.5.2.2 **ขั้นตอนการสุ่มตรวจสอบบัตรเลือกตั้ง**

ในการสุ่มตรวจสอบบัตรเลือกตั้งที่เขตเลือกตั้ง จะมีขั้นตอนต่าง ๆ ดังนี้

4.5.2.2.1 **การตรวจสอบจำนวนบัตร**

เมื่อนำบัตรเลือกตั้งจากหน่วยเลือกตั้งมาถึงเขตเลือกตั้ง ก่อนการเทรวมกับบัตรเลือกตั้งในหน่วยอื่น เจ้าหน้าที่ประจำเขตเลือกตั้งจะตรวจสอบจำนวนต้นขั้วกับจำนวนบัตรลงคะแนนในหีบเลือกตั้งว่าเท่ากันหรือไม่

4.5.2.2.2 **การตรวจสอบลายพิมพ์นิ้วมือของผู้มาใช้สิทธิ์**

เป็นการตรวจเฉพาะที่ต้นขั้วของบัตรเท่านั้น โดยตรวจสอบว่าลายพิมพ์นิ้วมือมีอยู่บนต้นขั้วบัตรทุกใบหรือไม่ และลายพิมพ์นิ้วมือมีลักษณะซ้ำกันบนบัตรนับสิบ ๆ ใบจนสังเกตเห็นได้หรือไม่

4.5.2.2.3 **การสุ่มบัตรเลือกตั้ง**

ทำการสุ่มตรวจทั้งตัวบัตรลงคะแนนและต้นขั้วของหน่วยเลือกตั้งนั้น โดยต้นขั้วจะสุ่มตรวจ 1 ใบ ใน 1 เล่ม ส่วนตัวบัตรลงคะแนนจะสุ่มมา 1 ใบ จาก 100 ใบ

4.5.2.2.4 การอ่านรหัสลับบนบัตรเลือกตั้ง

นำบัตรเลือกตั้งมาอ่านรหัสแท่งโดยใช้เครื่องอ่านรหัสแท่ง จะได้รหัสลับของบัตรเลือกตั้งใบนั้นขนาด 64 ไบต์

4.5.2.2.5 การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง

นำรหัสลับที่ได้มาถอดรหัสลับตามขั้นตอนต่อไปนี้

- อ่านข้อมูลกุญแจสาธารณะที่ได้ประกาศไว้สำหรับการเลือกตั้งครั้งนั้น ซึ่งอาจอ่านมาจากแฟ้มข้อมูลหรือรหัสแท่งก็ได้
- นำกุญแจสาธารณะที่ได้ไปถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งด้วยวิธี RSA โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ จะได้ข้อมูลประจำบัตรเลือกตั้งขนาดไม่เกิน 61 ไบต์ ซึ่งประกอบด้วย ข้อมูลที่มาของบัตร และข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ

เมื่อได้ข้อมูลประจำบัตรเลือกตั้ง จึงทำการตรวจสอบวันเลือกตั้ง รหัสเขตเลือกตั้งจากข้อมูลที่มาของบัตรซึ่งอยู่ในข้อมูลประจำบัตรเลือกตั้งว่าถูกต้องหรือไม่ นอกจากนี้ ในส่วนของบัตรลงคะแนน ยังตรวจเล่มที่ของบัตรซึ่งเป็นข้อมูลที่อยู่ในหมายเลขบัตรเลือกตั้งว่าตรงกับต้นขั้วที่ส่งมาหรือไม่

4.5.2.2.6 การตรวจสอบคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตร

นำข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ได้จากหัวข้อ 4.5.2.2.5 มาทำการตรวจสอบตามขั้นตอนดังนี้

- สแกนกระดาษและดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษใบนั้น ตามหัวข้อ 4.4.2.1 และ 4.4.2.2
- เปรียบเทียบพิกัดของจุดปลายซึ่งเป็นข้อมูลคุณลักษณะเฉพาะตัวที่ได้จากหัวข้อ 4.5.2.2.5 กับพิกัดของจุดปลายที่ได้จากหัวข้อ 4.4.2.2 โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์

4.5.2.2.7 การบันทึกผลการสุ่มตรวจ

เมื่อตรวจสอบเสร็จ 1 ใบ ก็จะบันทึกข้อมูลที่มาของบัตรและค่าแฮช (Hash value) ที่อ่านได้จากเครื่องลงในบันทึกผลการสุ่มตรวจและเซ็นชื่อกำกับไว้

4.5.2.2.8 กรณีที่พบความผิดปกติ

หากพบความผิดปกติเกิดขึ้น ให้ตรวจสอบบัตรทุกใบที่ได้รับในหน่วยเลือกตั้งนั้น และทำบันทึกไว้เป็นหลักฐาน

4.6 โปรแกรมที่พัฒนาขึ้นเพื่อใช้กับต้นแบบระบบเลือกตั้ง

4.6.1 โปรแกรมการสร้างและบันทึกกุญแจ

การเข้ารหัสลับที่ใช้กับบัตรเลือกตั้งเป็นการเข้ารหัสลับแบบกุญแจสาธารณะโดยวิธี RSA ซึ่งกุญแจที่ใช้สามารถสร้างและบันทึกได้โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ด้วยภาษา C++ ใน class ชื่อ CCrypt

4.6.1.1 การสร้างกุญแจ

กุญแจที่จะสร้างมีความยาว 512 บิต โดยโปรแกรมที่ใช้ในการสร้างกุญแจมีขั้นตอนการทำงานดังนี้

- เรียกใช้ library “cl32.dll” [12] ซึ่งเป็นซอฟต์แวร์สำหรับการเข้ารหัสลับของ Peter Gutmann เพื่อสร้างกุญแจ ซึ่งจะได้กุญแจมา 2 ตัว คือ กุญแจส่วนตัวและกุญแจสาธารณะ
- สุ่มค่าขึ้นมาค่าหนึ่งเพื่อใช้เป็นหมายเลขของกุญแจที่ได้สร้างขึ้น

กุญแจที่สร้างขึ้นจะประกอบไปด้วยส่วนต่าง ๆ ดังนี้

- *องค์ประกอบสาธารณะ*
  - ค่า n (Modulus) - ความยาวของ n (หน่วยเป็นบิต)
  - ค่า e - ความยาวของ e (หน่วยเป็นบิต)
- *องค์ประกอบส่วนตัว*
  - ค่า d - ความยาวของ d (หน่วยเป็นบิต)
  - ค่า p - ความยาวของ p (หน่วยเป็นบิต)
  - ค่า q - ความยาวของ q (หน่วยเป็นบิต)
  - ค่า u - ความยาวของ u (หน่วยเป็นบิต)
  - ค่า e1 - ความยาวของ e1 (หน่วยเป็นบิต)
  - ค่า e2 - ความยาวของ e2 (หน่วยเป็นบิต)

4.6.1.2 การบันทึกแฟ้มข้อมูลกุญแจสาธารณะ

กุญแจที่สร้างขึ้นสามารถบันทึกเป็นแฟ้มข้อมูลบนเครื่องไมโครคอมพิวเตอร์ได้ โดยกุญแจสาธารณะจะมีนามสกุลเป็น pub ซึ่งมีรูปแบบของแฟ้มข้อมูลตามตารางที่ 4.1

ตารางที่ 4.1 รูปแบบแฟ้มข้อมูลกุญแจสาธารณะ

ส่วนประกอบ	ความยาว (ไบต์)
- หมายเลขกุญแจ	2
- ความยาวของ n (หน่วยเป็นบิต)	4
- ค่า n	ตามค่าที่เก็บไว้
- ความยาวของ e (หน่วยเป็นบิต)	4
- ค่า e	ตามค่าที่เก็บไว้

4.6.1.3 การบันทึกแฟ้มข้อมูลกุญแจส่วนตัว

กุญแจส่วนตัวจะมีนามสกุลเป็น prv โดยกุญแจส่วนตัวจะมีลักษณะพิเศษคือสามารถถูกแบ่งออกเป็นส่วน ๆ ได้ตามความต้องการ ซึ่งโดยปกติแล้วก็จะแบ่งให้เท่ากับจำนวนคณะกรรมการการเลือกตั้งที่มีอยู่ นอกจากนี้ กุญแจส่วนตัวยังต้องมีรหัสผ่าน เพื่อป้องกันไม่ให้ผู้อื่นนำเอากุญแจส่วนตัวไปใช้ได้

ก่อนการบันทึกข้อมูล จะต้องทำการแบ่งกุญแจส่วนตัวเป็นส่วน ๆ ตามขั้นตอนดังนี้

- นำองค์ประกอบส่วนตัวของกุญแจที่สร้างขึ้นมาเรียงต่อกัน แล้วแบ่งออกเป็นส่วน ๆ ตามจำนวนคณะกรรมการโดยการวางสลับ (Interleave)
- เก็บค่าความยาวขององค์ประกอบส่วนตัวแต่ละส่วนที่ถูกแบ่งออกมา
- แทรก “CAST” ที่ส่วนหัวขององค์ประกอบส่วนตัวแต่ละส่วน เพื่อใช้เป็น header ในการตรวจสอบการถอดรหัสลับแบบกุญแจลับ
- เติมศูนย์ต่อท้ายองค์ประกอบส่วนตัวแต่ละส่วนเพื่อให้จำนวนไบต์ขององค์ประกอบส่วนตัวแต่ละส่วนหาร 8 ได้ลงตัว
- เรียกใช้ library “cl32.dll” [12] เพื่อเข้ารหัสลับองค์ประกอบส่วนตัวแต่ละส่วนด้วยวิธี CAST ในโหมด CBC (Cipher block chaining) ซึ่งเป็นวิธีการเข้ารหัสลับแบบกุญแจลับ โดยกุญแจที่ใช้คำนวณมาจากรหัสผ่านของกรรมการแต่ละท่าน
- อ่านค่าเวกเตอร์เริ่มต้น (Initialization vector) ที่ใช้ในการเข้ารหัสลับ

หลังจากนั้นจึงทำการบันทึกข้อมูลลงแฟ้ม โดยรูปแบบแฟ้มข้อมูลของกุญแจส่วนตัวแต่ละส่วนเป็นไปตามตารางที่ 4.2

ตารางที่ 4.2 รูปแบบแฟ้มข้อมูลกุญแจส่วนตัวแต่ละส่วน

ส่วนประกอบ	ความยาว (ไบต์)
- หมายเลขกุญแจ	2
- ความยาวของ n (หน่วยเป็นบิต)	4
- ค่า n	ตามค่าที่เก็บไว้
- ความยาวของ e (หน่วยเป็นบิต)	4
- ค่า e	ตามค่าที่เก็บไว้
- หมายเลขส่วน (ส่วนที่เท่าไร)	1
- จำนวนส่วนทั้งหมด	1
- เวกเตอร์เริ่มต้น (Initialization vector)	8
- ความยาวขององค์ประกอบส่วนตัวก่อนเข้ารหัสลับ (หน่วยเป็นไบต์)	2
- องค์ประกอบส่วนตัวที่เข้ารหัสลับแล้ว	ตามค่าที่เก็บไว้

จะเห็นว่ากุญแจส่วนตัวจะมีองค์ประกอบสาธารณะรวมอยู่ด้วย ดังนั้น จึงสามารถนำกุญแจส่วนตัวมาใช้ในการถอดรหัสลับได้ด้วย

4.6.2 โปรแกรมการพิมพ์รหัสลับบนบัตรเลือกตั้ง

ในการพิมพ์รหัสลับลงบนบัตรเลือกตั้ง จะเริ่มด้วยการดึงคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง จากนั้นจึงนำคุณลักษณะที่ได้มารวมกับข้อมูลที่มาของบัตร นำไปเข้ารหัสลับ แล้วจึงบันทึกรหัสลับที่ได้ลงบนบัตรเลือกตั้ง ซึ่งมีรายละเอียดดังนี้

4.6.2.1 ข้อกำหนดเบื้องต้น

- กระดาษที่ใช้พิมพ์บัตรเลือกตั้งจะมีวัตถุลักษณะเป็นเส้นฝังอยู่ ดังรูปที่ 4.2 และมีการพิมพ์กรอบไว้ทั้งในส่วนของต้นขั้วและตัวบัตรลงคะแนน เพื่อใช้เป็นพื้นที่สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษ กรอบที่ใช้มีขนาด 3.2x1.6 ตารางนิ้ว
- บัตรเลือกตั้งจะถูกสแกนในบริเวณกรอบที่กำหนดแล้วบันทึกเป็นแฟ้มข้อมูลในรูปแบบ bitmap (bmp) แบบสเกลสีเทา ไม่ใช้การเข้ารหัสแบบ RLE ความละเอียด 100 จุดต่อนิ้ว ดังรูปที่ 4.3

4.6.2.2 **การดึงคุณลักษณะเฉพาะตัวจากกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง**

คุณลักษณะเฉพาะตัวของกระดาษที่ใช้ในงานวิจัยนี้ คือ ค่าพิกัดของจุดปลายของวัตถุที่ฝังอยู่ในเนื้อกระดาษ ดังรูปที่ 4.4 การดึงคุณลักษณะเฉพาะตัวจากกระดาษ สามารถทำได้โดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ ด้วยภาษา C++ อยู่ใน class ชื่อ CExtract ซึ่งมีขั้นตอนดังนี้

4.6.2.2.1 **การอ่านภาพกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง**

เรียกใช้ class “cdib” เพื่ออ่านภาพจากแฟ้มข้อมูลที่เก็บไว้

4.6.2.2.2 **การหากรอบภาพ** (Frame detection)

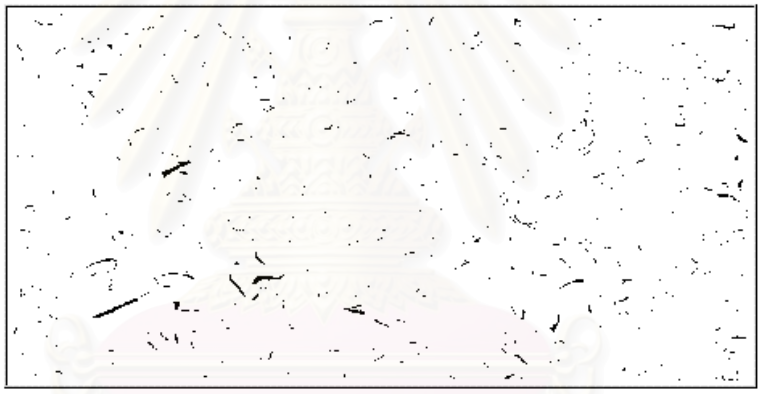
การหากรอบของภาพที่จะใช้ดึงคุณลักษณะเฉพาะตัว มีขั้นตอนดังนี้

- กำหนดให้จุดมุมบนด้านซ้ายเป็นจุดกำเนิด (มีพิกัด (0,0)) แกน +X ชี้ไปทางขวา ส่วนแกน +Y ชี้ลงด้านล่าง
- คัดลอกภาพเพื่อใช้ในการหากรอบภาพ
- แปลงภาพที่ได้คัดลอกมาให้เป็นภาพสองระดับ
- หากลุ่มจุดดำที่มีพื้นที่มากที่สุด ซึ่งน่าจะเป็นกรอบของภาพ
- ลบกลุ่มจุดดำอื่น ๆ ที่ไม่เกี่ยวข้องออกให้หมด
- สุ่มหาจุดซึ่งอยู่บนขอบด้านในของกรอบแต่ละด้านของภาพ ทั้ง 4 ด้าน
- หาสมการเส้นตรงของขอบด้านในของกรอบแต่ละด้านจากจุดที่หาได้ โดยใช้การถดถอยเชิงเส้น (Linear regression)
- หาจุดตัดของสมการเส้นตรงที่หาได้ ซึ่งจะเป็นมุมทั้ง 4 ของขอบด้านในของกรอบ
- เมื่อสามารถหากรอบได้แล้ว จึงนำภาพต้นฉบับมาลบกรอบและข้อมูลที่ไม่เกี่ยวข้องซึ่งอยู่ภายนอกกรอบออก จะได้ภาพดังรูปที่ 4.5
- เนื่องจากภาพที่สแกนได้อาจเอียงและขอบด้านในของกรอบด้านมุมบนซ้ายไม่อยู่ตรงจุดกำเนิด จึงต้องหาค่ามุมเอียงและค่าพิกัดที่เลื่อนไปจากจุดกำเนิดเพื่อใช้ในขั้นตอนการเลื่อนและหมุนภาพต่อไป

รูปที่ 4.5 ภาพที่ผ่านการหากรอบภาพในกระบวนการดึงคุณลักษณะเฉพาะตัว

4.6.2.2.3 การแปลงเป็นภาพสองระดับ (Binarization)

นำภาพที่ผ่านการหากรอบภาพมาแปลงจากภาพชนิดสเกลสีเทา ให้เป็นภาพสองระดับ จะได้ภาพดังรูปที่ 4.6



รูปที่ 4.6 ภาพที่ผ่านการแปลงเป็นภาพสองระดับในกระบวนการดึงคุณลักษณะเฉพาะตัว

4.6.2.2.4 การหากลุ่มจุดดำบนภาพ

เมื่อได้ภาพสองระดับ จึงนำภาพมาหากลุ่มจุดดำที่จะใช้ในการหาคุณลักษณะในกระบวนการอื่นภายหลัง โดยมีขั้นตอนดังนี้

- หาพื้นที่ของกลุ่มจุดดำบนภาพ ถ้ามีพื้นที่น้อยกว่า 3 จุด ให้ลบทิ้ง ถ้ามีพื้นที่ตั้งแต่ 3 จุดขึ้นไป ให้เก็บไว้
- เรียงลำดับกลุ่มจุดดำที่เก็บได้ จากกลุ่มที่มีพื้นที่มากไปพื้นที่น้อย ซึ่งทำให้การเก็บคุณลักษณะจะได้คุณลักษณะของกลุ่มจุดดำที่มีพื้นที่มากก่อน
- ลบกลุ่มจุดดำส่วนที่เกินจากความต้องการซึ่งมีพื้นที่น้อยออกไป จะได้ภาพดังรูปที่ 4.7

รูปที่ 4.7 ภาพที่ผ่านการหากลุ่มจุดดำบนภาพในกระบวนการดึงคุณลักษณะเฉพาะตัว

4.6.2.2.5 การทำโครงร่างภาพ (Thinning)

นำกลุ่มจุดดำที่ได้มาทำโครงร่างภาพ เพื่อเตรียมภาพสำหรับการหาจุดปลายของกลุ่มจุดดำในกระบวนการถัดไป โดยการทำโครงร่างาภาพที่ใช้เป็นการทำโครงร่างภาพโดยวิธี One-Pass Parallel Thinning [6] ของ Ben K. Jang และ Roland T. Chin เนื่องจากให้รายละเอียดคุณลักษณะของจุดต่อภาพได้ดีเพียงพอ โดยได้นำฟังก์ชันการทำโครงร่างภาพของนายประเสริฐ ฉอเรืองวิวัฒน์ ในวิทยานิพนธ์เรื่อง “การรู้จำตัวอักษรเขียนภาษาไทยโดยใช้การวิเคราะห์ลักษณะบ่งความต่าง” [13] มาดัดแปลงใช้ กลุ่มจุดดำที่ผ่านการทำโครงร่างภาพจะเหลือความกว้างเพียง 1 จุดภาพ ดังรูปที่ 4.8

รูปที่ 4.8 ภาพที่ผ่านการทำโครงร่างภาพในกระบวนการดึงคุณลักษณะเฉพาะตัว

4.6.2.2.6 การหาจุดปลายของกลุ่มจุดดำ

หลังจากผ่านการทำโครงร่างภาพ ก็จะทำการจัดแบ่งประเภทของจุดดำบนภาพโดยนับจำนวนครั้งของการเปลี่ยนค่าของจุดที่อยู่ล้อมรอบจุดดำนั้น โดยอาจนับในทิศทางทวนเข็มนาฬิกาหรือตามเข็มนาฬิกาก็ได้ จะได้ประเภทของจุดดำบนภาพ ซึ่งประกอบด้วย

- จุดเดี่ยว คือ จุดดำที่ไม่มีการเปลี่ยนค่าของจุดที่อยู่ล้อมรอบจุดดำนั้น
- จุดปลาย คือ จุดดำที่มีจำนวนครั้งของการเปลี่ยนค่าของจุดที่อยู่ล้อมรอบจุดดำนั้น เท่ากับ “สอง”
- จุดต่อเนื่อง คือ จุดดำที่มีจำนวนครั้งของการเปลี่ยนค่าของจุดที่อยู่ล้อมรอบจุดดำนั้น เท่ากับ “สี่”
- จุดแยก คือ จุดดำที่มีจำนวนครั้งของการเปลี่ยนค่าของจุดที่อยู่ล้อมรอบจุดดำนั้น ตั้งแต่ “หก” ขึ้นไป

เมื่อได้ประเภทของจุดดำแล้ว จะเก็บค่าพิกัดของจุดดำที่เป็น “จุดปลาย” หรือ “จุดเดี่ยว” ของกลุ่มจุดดำนั้น ซึ่งเป็นคุณลักษณะที่ต้องการ โดยจะเรียกรวมกันว่า จุดปลาย

4.6.2.2.7 **การเลื่อนและหมุนภาพ** (Translation and rotation)

นำค่าพิกัดที่เก็บได้ในกระบวนการที่แล้วมาเลื่อนและหมุน โดยใช้ค่ามุมเอียงและค่าพิกัดที่เลื่อนไปจากจุดกำเนิดที่คำนวณได้ในกระบวนการหากรอบภาพ

4.6.2.2.8 **การลดความละเอียดของภาพ** (Subsampling)

เนื่องจากพิกัดที่ได้มีความละเอียดมากเกินไป จึงทำการลดความละเอียดของภาพโดยหารค่าพิกัดด้วยค่าคงที่ค่าหนึ่ง (ในงานวิจัยนี้ใช้ค่าคงที่ดังกล่าวเป็น 4) ทำให้พิกัดมีความละเอียดลดลง แต่ก็เกิดปัญหาคือ พิกัดของจุดอาจมีค่าเท่ากันได้ จึงต้องลบจุดนั้นทิ้งถ้าพิกัดของจุดนั้นซ้ำกับจุดอื่นในกลุ่มจุดดำเดียวกัน เมื่อผ่านขั้นตอนนี้จะได้พิกัดของจุดปลายดังรูปที่ 4.9

รูปที่ 4.9 พิกัดของจุดปลายที่ได้จากกระบวนการดึงคุณลักษณะเฉพาะตัว

4.6.2.2.9 **การจัดรูปแบบข้อมูลคุณลักษณะที่ได้**

ข้อมูลคุณลักษณะที่ได้จะเป็นเลขฐาน 16 ความยาวไม่เกิน 50 ไบต์ แบ่งออกเป็นกลุ่มตามจำนวนจุดในกลุ่มจุดดำ เรียงต่อกันไปเรื่อย ๆ โดยในแต่ละกลุ่มจะมีรูปแบบข้อมูลดังนี้

ส่วนประกอบ	ความยาว (ไบต์)
- ชนิดซึ่งแบ่งตามจำนวนจุด เช่น 01 หมายถึง มีจุด 1 จุด ในกลุ่มจุดดำนั้น	1
- ความยาวของจุดทั้งหมดที่อยู่ในกลุ่มจุดดำชนิดเดียวกัน (หน่วยเป็นไบต์)	1
- พิกัดของจุดที่อยู่ในกลุ่มจุดดำชนิดเดียวกัน	ตามค่าที่เก็บไว้

4.6.2.3 **การเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง**

เมื่อได้คุณสมบัติเฉพาะตัวของกระดาษ ก็จะนำมาต่อท้ายข้อมูลที่มาของบัตร กลายเป็นข้อมูลประจำบัตรเลือกตั้งใบนั้น ๆ ขนาดไม่เกิน 61 ไบต์ ตามตารางที่ 4.3 จากนั้น คณะกรรมการการเลือกตั้งจะนำข้อมูลประจำบัตรเลือกตั้งแต่ละใบมาเข้ารหัสลับโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ ด้วยภาษา C++ อยู่ใน class ชื่อ CCrypt ซึ่งมีขั้นตอนดังนี้

ตารางที่ 4.3 รูปแบบข้อมูลประจำบัตรเลือกตั้ง

ส่วนประกอบ	ความยาว (ไบต์)
- วันเลือกตั้ง	4
- รหัสเขตเลือกตั้ง	2
- หมายเลขบัตรเลือกตั้ง	4
- ไบต์สำรอง	1
- คุณลักษณะเฉพาะตัวของกระดาษ	ไม่เกิน 50

4.6.2.3.1 **การอ่านแฟ้มข้อมูลกุญแจส่วนตัว**

กุญแจส่วนตัวแต่ละส่วนจะถูกอ่านมาจากแฟ้มข้อมูลของกรรมการแต่ละท่านซึ่งมีนามสกุล prv ซึ่งจะได้ข้อมูลต่าง ๆ ตามตารางที่ 4.2 แล้วนำมาผ่านขั้นตอนต่าง ๆ ดังนี้

- เรียกใช้ library “cl32.dll” [12] เพื่อถอดรหัสลับองค์ประกอบส่วนตัวแต่ละส่วนที่เข้ารหัสลับไว้ด้วยวิธี CAST ในโหมด CBC (Cipher block chaining) โดยกุญแจที่ใช้คำนวณมาจากรหัสผ่านของกรรมการแต่ละท่าน และใช้เวกเตอร์เริ่มต้นที่อ่านได้จากแฟ้มข้อมูล
- ตรวจสอบหมายเลขกุญแจ ค่าจำนวนส่วนทั้งหมด องค์ประกอบสาธารณะของกุญแจส่วนตัวแต่ละส่วนว่าตรงกันหรือไม่
- ตรวจสอบหมายเลขส่วน ว่ามีครบทุกส่วนหรือไม่

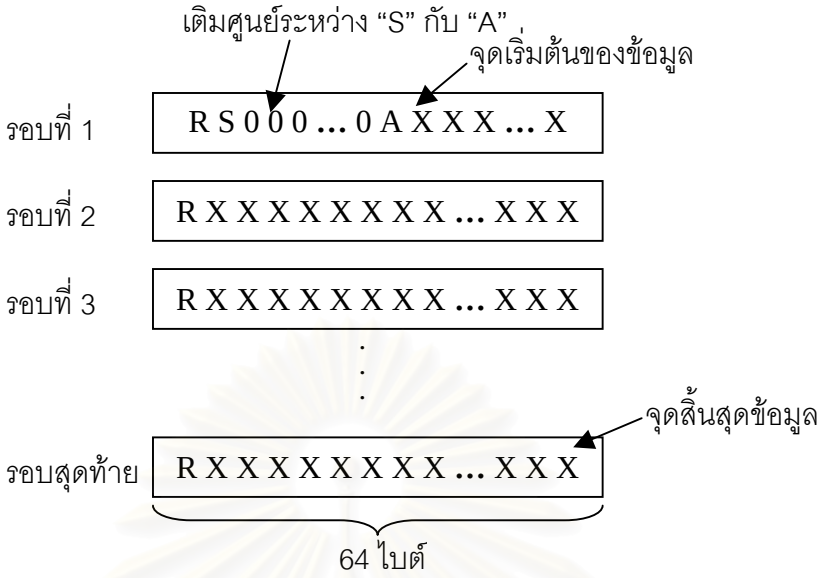
- เรียงลำดับกุญแจส่วนตัวแต่ละส่วนตามลำดับที่ถูกต้องโดยใช้หมายเลขส่วน จะได้องค์ประกอบส่วนตัวของกุญแจที่ต้องการ

4.6.2.3.2 การเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง

เมื่อได้กุญแจส่วนตัวแล้ว จึงนำกุญแจส่วนตัวมาเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้ง ตามขั้นตอนดังนี้

- จัดรูปแบบข้อมูลประจำบัตรเลือกตั้ง ดังรูปที่ 4.10 โดยมีรายละเอียดดังนี้
  - แทรก “R” ทางด้านหน้าของข้อมูลที่จะเข้ารหัสลับในแต่ละรอบ เพื่อให้แน่ใจว่าข้อมูลที่จะเข้ารหัสลับมีค่าไม่เกิน n (modulus) ทำให้ข้อมูลประจำบัตรเลือกตั้งที่จะเข้ารหัสลับในแต่ละรอบมีความยาว 63 ไบต์ ยกเว้นรอบแรก
  - ในรอบแรก แทรก “S” ต่อจาก “R” เพื่อใช้เป็น header ในการตรวจสอบการถอดรหัสลับ
  - แทรก “A” หน้าจุดเริ่มต้นของข้อมูล เพื่อใช้เป็นตัวบอกจุดเริ่มต้นของข้อมูล
  - เติมศูนย์ระหว่าง “S” กับ “A” ให้ได้ความยาวของข้อมูลทั้งหมดเป็นจำนวนเท่าของ 512 ไบต์

  หมายเหตุ เนื่องจากข้อมูลประจำบัตรเลือกตั้งที่ได้มีขนาดไม่เกิน 61 ไบต์ เมื่อรวมกับ “R”, “S”, “A” จะมีขนาดไม่เกิน 64 ไบต์ หรือ 512 บิต จึงใช้การเข้ารหัสลับเพียง 1 รอบเท่านั้น
- เรียกใช้ library “cl32.dll” [12] เพื่อเข้ารหัสลับข้อมูลประจำบัตรเลือกตั้งที่ผ่านการจัดรูปแบบแล้วด้วยวิธี RSA โดยใช้กุญแจที่ได้จากหัวข้อ 4.6.2.3.1 จะได้รหัสลับที่ต้องการซึ่งอยู่ในรูปเลขฐาน 16 ขนาด 64 ไบต์



รูปที่ 4.10 การจัดรูปแบบข้อมูลประจำบัตรเลือกตั้งก่อนการเข้ารหัสลับ

4.6.2.4 การบันทึกรหัสลับลงบนบัตรเลือกตั้ง

นำรหัสลับที่ได้มาเปลี่ยนเป็นรหัสแท่ง แล้วพิมพ์ลงบนบัตรเลือกตั้งโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ ด้วยภาษา C++ อยู่ใน class ชื่อ CBarcode ซึ่งมีขั้นตอนดังนี้

- เปลี่ยนรหัสลับซึ่งเป็นเลขฐาน 16 ให้อยู่ในรูปรหัสแอสกี ทำให้รหัสลับ 1 ไบต์ กลายเป็นรหัสแอสกี 2 ตัว จึงได้รหัสแอสกีทั้งหมด 128 ตัว
- นำรหัสแอสกีที่ได้มาเปลี่ยนเป็นรหัสแท่งโดยใช้ code 128 ชุด B
- คำนวณหาความกว้างของรหัสแท่งซึ่งขึ้นกับความละเอียดของเครื่องพิมพ์ด้วย เนื่องจากต้องให้ความกว้างของมอดูล (Module) ของรหัสแท่งเป็นจำนวนเท่าของจุดของเครื่องพิมพ์
- พิมพ์รหัสแท่งที่ได้ลงบนบัตรเลือกตั้ง

4.6.3 โปรแกรมการถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งที่หน่วยเลือกตั้ง

โปรแกรมนี้เขียนโดยใช้ภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051 เพื่อใช้ถอดรหัสลับที่อ่านได้จากบัตรเลือกตั้งในขั้นตอนการตรวจสอบความถูกต้องของบัตรเลือกตั้งที่หน่วยเลือกตั้ง ซึ่งมีรายละเอียดดังนี้

4.6.3.1 **ข้อกำหนดเบื้องต้น**

การเก็บค่าลงในหน่วยความจำ จะให้ค่านัยสำคัญน้อยสุด (LSB) อยู่ที่ตำแหน่งน้อยสุดของหน่วยความจำ แล้วเก็บค่าเรียงขึ้นไปเรื่อย ๆ จนถึงค่านัยสำคัญมากสุด (MSB) จะอยู่ไม่เกินตำแหน่งมากสุดของหน่วยความจำที่กำหนดไว้

ตัวแปรที่สำคัญต่าง ๆ จะเก็บอยู่ในตำแหน่งตามตารางที่ 4.4

ตารางที่ 4.4 ตำแหน่งหน่วยความจำของตัวแปรสำคัญในไมโครคอนโทรลเลอร์

ชนิด	ตำแหน่ง
c	0000H – 007FH
e	0080H – 00FFH
n	0100H – 017FH
n ที่ถูกเลื่อนไปทางขวา 1 บิต	0200H – 0280H
n ที่ถูกเลื่อนไปทางขวา 2 บิต	0300H – 0380H
n ที่ถูกเลื่อนไปทางขวา 3 บิต	0400H – 0480H
n ที่ถูกเลื่อนไปทางขวา 4 บิต	0500H – 0580H
n ที่ถูกเลื่อนไปทางขวา 5 บิต	0600H – 0680H
n ที่ถูกเลื่อนไปทางขวา 6 บิต	0700H – 0780H
n ที่ถูกเลื่อนไปทางขวา 7 บิต	0800H – 0880H
บัฟเฟอร์ 0	0900H – 09FFH
บัฟเฟอร์ 1	0A00H – 0AFFH

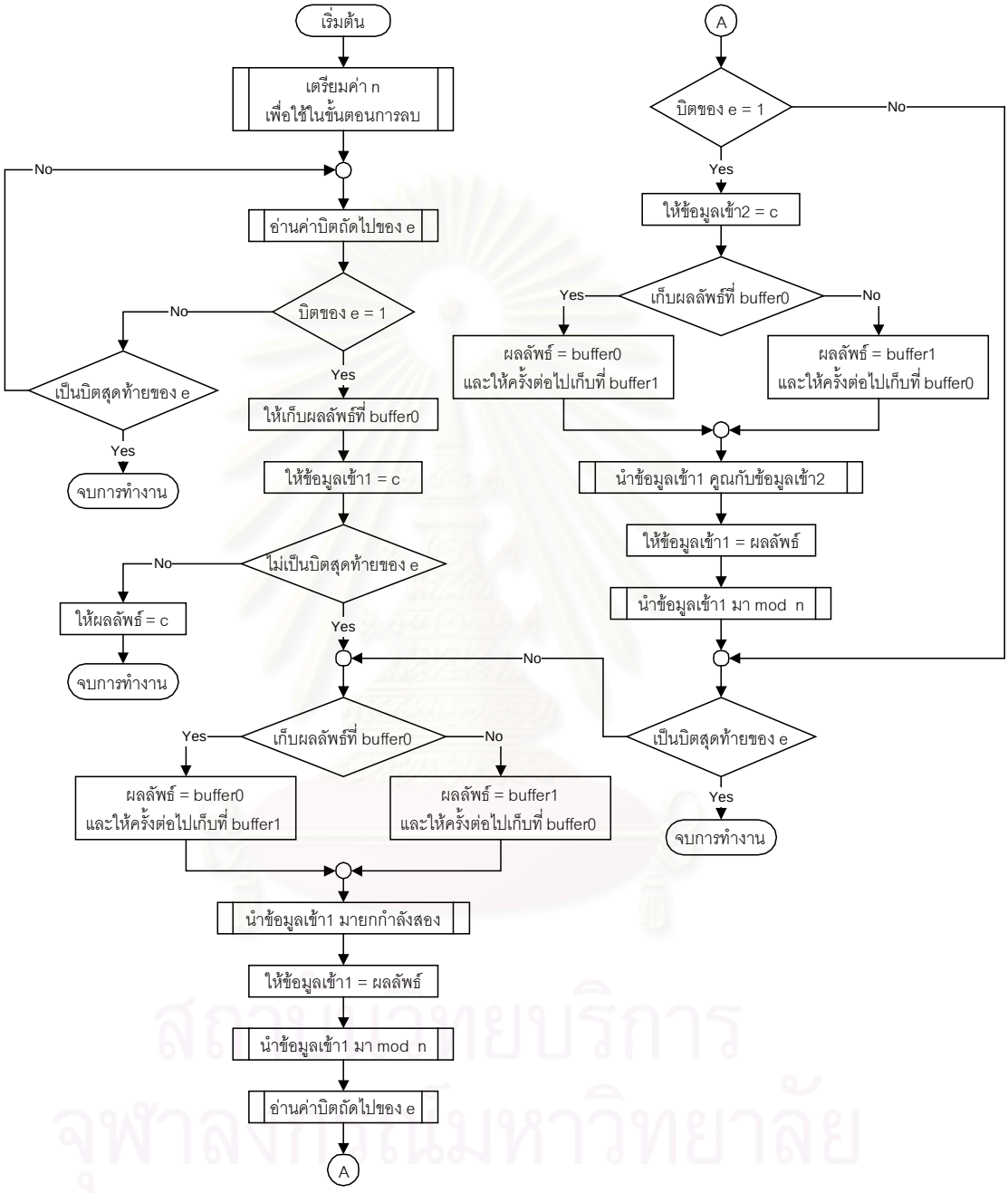
4.6.3.2 **การทำงานของโปรแกรม**

โปรแกรมที่เขียนขึ้น อาจแบ่งออกได้เป็น 8 ส่วน คือ การยกกำลังแล้วหาเศษจากการหาร เป็นโปรแกรมหลัก ซึ่งจะเรียกใช้โปรแกรมย่อยอื่น ๆ , การเตรียมค่า n, การอ่านค่าบิตถัดไปของ e, การคูณ, การยกกำลังสอง, การหาเศษจากการหาร, การเปรียบเทียบเศษกับตัวหาร และการลบค่าตัวหารออกจากเศษ ดังมีรายละเอียดดังนี้

4.6.3.2.1 **การยกกำลังแล้วหาเศษจากการหาร** (Modular Exponentiation)

ส่วนนี้เป็นส่วนหลักของโปรแกรม ใช้หลักการ Binary method ในการยกกำลังแล้วหาเศษจากการหาร ดังที่ได้กล่าวไว้ในหัวข้อ 2.3.1 การทำงานของโปรแกรมเป็นดังรูปที่ 4.11

โดยเรียกใช้โปรแกรมย่อยอื่นๆ และมีบัฟเฟอร์1 และบัฟเฟอร์2 เป็นตัวเก็บผลลัพธ์ (เศษจากการหาร) สลับกันไปในแต่ละรอบเพื่อให้ไม่เสียเวลาในการย้ายข้อมูล



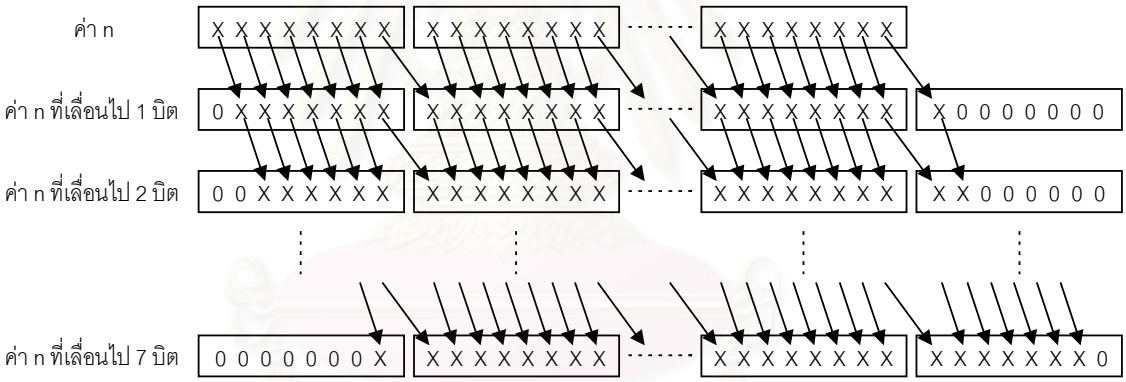
รูปที่ 4.11 แผนผังขั้นตอนการทำงานในการยกกำลังแล้วหาเศษจากการหาร

4.6.3.2.2 การเตรียมค่า n

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ (2-2) เพื่อเตรียมค่า n (Modulus) ไว้ใช้ในโปรแกรมการลบค่าตัวหารออกจากตัวตั้งซึ่งเป็นขั้นตอนหนึ่งในการหาเศษจากการหาร

เนื่องจากการลบค่าตัวหาร (n) ออกจากตัวตั้ง ต้องมีการเลื่อนค่า n ไปทางขวาครั้งละ 1 บิต (คือ n/2) แล้วนำมาลบออกจากตัวตั้ง การเลื่อนจะต้องทำไปเรื่อย ๆ จนกว่าจะได้เศษจากการหารที่ต้องการ จะสังเกตเห็นว่าเมื่อเลื่อนไปครบ 8 บิต หรือ 1 ไบต์ ก็จะมีค่าเหมือนกับค่า n เดิม แต่มีตำแหน่งเปลี่ยนไป 1 ไบต์ และเมื่อเลื่อนไป 9 บิต ก็จะมีค่าเหมือนกับ n เลื่อนไป 1 บิต แต่มีตำแหน่งเปลี่ยนไป 1 ไบต์ จึงเป็นการทำงานซ้ำซ้อน ทำให้เสียเวลาในการประมวลผล

โปรแกรมนี้จึงทำหน้าที่เลื่อนค่า n ไปทางขวา ตั้งแต่ 1 ไบต์ จนถึง 7 ไบต์ ดังรูปที่ 4.12 แล้วเก็บค่าไว้ในตัวแปรตามตำแหน่งในตารางที่ 4.4 เมื่อต้องการใช้ค่า n ค่าไหน ก็มาอ่านค่าจากตัวแปรที่เก็บไว้



รูปที่ 4.12 การเลื่อนค่า n ไปทางขวา ตั้งแต่ 1 ถึง 7 ไบต์

4.6.3.2.3 การอ่านค่าบิตถัดไปของ e

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ (2-2) เพื่ออ่านค่าบิตของ e ทีละบิต ตั้งแต่บิตที่มีนัยสำคัญมากสุด (MSB) ไปจนถึงบิตที่มีนัยสำคัญน้อยสุด (LSB)

4.6.3.2.4 การคูณ

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ (2-2) เพื่อคูณเลข 2 จำนวนที่ได้รับมา ตามวิธีการในหัวข้อ 2.3.2

4.6.3.2.5 การยกกำลังสอง

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ (2-2) เพื่อยกกำลังสองตัวเลขที่ได้รับมา ตามวิธีการในหัวข้อ 2.3.3

4.6.3.2.6 การหาเศษจากการหาร

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ (2-2) เพื่อนำตัวตั้งที่ได้รับมาหารด้วย n แล้วหาเศษจากการหารตามวิธีในหัวข้อ 2.3.4 แต่มีการปรับเปลี่ยนเล็กน้อย คือ วิธีในหัวข้อ 2.3.4 นั้น จะนำตัวหารมาลบออกจากเศษก่อน แล้วพิจารณาว่าเศษที่ได้เป็นลบหรือไม่ ถ้าเป็นลบก็จะกลับไปใช้ค่าเศษเดิมก่อนที่จะลบกัน ส่วนในโปรแกรมนี้ จะพิจารณาว่าตัวหารมีค่ามากกว่าเศษหรือไม่ก่อน ถ้าตัวหารมีค่าน้อยกว่าเศษจึงทำการลบตัวหารออกจากเศษ แต่ถ้าตัวหารมีค่ามากกว่าเศษ ก็เปลี่ยนตัวหารจนกว่าจะมีค่าน้อยกว่าเศษ จึงนำมาลบออกจากเศษ

การเปลี่ยนตัวหารเพื่อเปรียบเทียบกับเศษนั้น จะใช้ค่าตัวหาร n ที่ได้เตรียมไว้ตามขั้นตอนในหัวข้อ 4.6.3.2.2 แล้วส่งค่าไปยังโปรแกรมย่อยเพื่อเปรียบเทียบเศษกับตัวหาร และ/หรือลบค่าตัวหารออกจากเศษต่อไป

4.6.3.2.7 การเปรียบเทียบเศษกับตัวหาร

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ 4.6.3.2.6 เพื่อเปรียบเทียบค่าระหว่างเศษกับตัวหารที่ได้รับมา การเปรียบเทียบจะเปรียบเทียบไบต์ที่มีนัยสำคัญมากสุดก่อน ถ้ามีค่าเท่ากันก็จะเปรียบเทียบไบต์ที่มีค่านัยสำคัญต่ำกว่าไปเรื่อย ๆ จนกว่าจะพบไบต์ที่มีค่าไม่เท่ากัน หรือจนถึงไบต์ที่มีนัยสำคัญน้อยสุด

4.6.3.2.8 การลบค่าตัวหารออกจากเศษ

โปรแกรมย่อยนี้จะถูกเรียกใช้โดยโปรแกรมในข้อ 4.6.3.2.6 เพื่อลบค่าตัวหารออกจากเศษ โดยเริ่มลบจากไบต์ที่มีนัยสำคัญน้อยสุดก่อน ไปเรื่อย ๆ จนถึงไบต์ที่มีนัยสำคัญมากสุด

4.6.3.3 ผลการทดสอบโปรแกรม

โปรแกรมที่เขียนขึ้นถูกแปลงเป็นภาษาเครื่องโดยใช้ cross assembler สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ชื่อ ASEM-51 Version 1.2 [14] ของ W.W. Heinz ซึ่งทำงานบนระบบปฏิบัติการ MS-DOS จากนั้น ได้ลองใช้งานบนตัวจำลองแบบของไมโครคอนโทรลเลอร์ตระกูล 8051 ชื่อ TS Controls Emulator 8051 Evaluation Version 1.00 [15] ของ Tarvydas-Sanford Controls Inc. ซึ่งทำงานบนระบบปฏิบัติการ MS Windows 98 โดยใช้ค่าต่าง ๆ ดังนี้

- ความถี่ของคริสตัล 11.059 เมกะเฮิรตซ์

- กุญแจสาธารณะซึ่งเป็นกุญแจที่ใช้ในการถอดรหัสลับ มีค่า 10001H ความยาว 17 บิต
- ความยาวของกุญแจ (ความยาวของ n) คือ 512 บิต

ปรากฏว่าใช้ความเร็วในการถอดรหัสลับประมาณ 9 วินาที ส่วนทางด้านหน่วยความจำ ใช้หน่วยความจำภายในแบบไบต์ 22 ไบต์ แบบบิต 3 บิต ใช้รีจิสเตอร์ R0 ถึง R7 และใช้หน่วยความจำภายนอกตามตารางที่ 4.4

4.6.4 โปรแกรมการตรวจสอบความถูกต้องของบัตรเลือกตั้งที่เขตเลือกตั้ง

4.6.4.1 ข้อกำหนดเบื้องต้น

บัตรเลือกตั้งจะถูกสแกนในบริเวณกรอบสำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษแล้วบันทึกเป็นแฟ้มข้อมูลในรูปแบบเดียวกันกับขั้นตอนการพิมพ์รหัสลับบนบัตรเลือกตั้ง คือ เป็นรูปแบบ bitmap (bmp) แบบสเกลสีเทา ไม่ใช้การเข้ารหัสแบบ RLE ความละเอียด 100 จุดต่อนิ้ว

4.6.4.2 การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง

นำรหัสลับที่ได้มาถอดรหัสลับโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ด้วยภาษา C++ อยู่ใน class ชื่อ CCrypt ซึ่งมีขั้นตอนดังนี้

4.6.4.2.1 การอ่านกุญแจสาธารณะ

กุญแจสาธารณะสามารถอ่านมาได้ทั้งจากแฟ้มข้อมูลและจากรหัสแท่งของกุญแจสาธารณะ โดย

- ถ้าอ่านจากแฟ้มข้อมูล สามารถอ่านจากแฟ้มข้อมูลกุญแจสาธารณะที่มีนามสกุล pub หรือจากแฟ้มข้อมูลกุญแจส่วนตัวที่มีนามสกุล prv ก็ได้ ซึ่งจะได้ค่าต่าง ๆ ตามตารางที่ 4.1
- ถ้าอ่านจากรหัสแท่งของกุญแจสาธารณะ จะต้องอ่าน 4 ค่า คือ ค่า n, ความยาวของ n (บิต), ค่า e, ความยาวของ e (บิต) ซึ่งจะได้ข้อมูลเหมือนกับตารางที่ 4.1 เว้นแต่ว่าไม่มีหมายเลขกุญแจ

4.6.4.2.2 การถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง

เมื่อได้กุญแจสาธารณะ จึงนำกุญแจสาธารณะมาถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง ตามขั้นตอนต่อไปนี้

- ตรวจสอบความยาวของข้อมูลว่าเป็นจำนวนเท่าของความยาวกุญแจหรือไม่

- แบ่งข้อมูลออกเป็นส่วน ๆ ตามความยาวของกุญแจสำหรับการถอดรหัสลับในแต่ละรอบ แต่เนื่องจากข้อมูลยาว 64 ไบต์ จึงใช้การถอดรหัสเพียง 1 รอบ
- เรียกใช้ library “cl32.dll” [12] เพื่อถอดรหัสลับข้อมูลแต่ละรอบด้วยวิธี RSA โดยใช้กุญแจที่ได้จากหัวข้อ 4.6.4.2.1 จะได้ข้อมูลดังรูปที่ 4.10
- ตัดค่า header ต่าง ๆ ออกให้หมด จะได้ข้อมูลประจำบัตรเลือกตั้งที่ต้องการ

เมื่อได้ข้อมูลประจำบัตรเลือกตั้ง จึงทำการตรวจสอบวันเลือกตั้ง รหัสเขตเลือกตั้ง หมายเลขบัตรเลือกตั้ง จากข้อมูลที่มาของบัตรซึ่งอยู่ในข้อมูลประจำบัตรเลือกตั้งว่าถูกต้องหรือไม่ พร้อมจดบันทึกข้อมูลที่มาของบัตรและค่าแฮชที่อ่านได้จากเครื่องลงในบันทึกผลการสุ่มตรวจและเซ็นชื่อกำกับไว้

4.6.4.3 การตรวจสอบคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตร

นำข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ได้จากหัวข้อ 4.6.4.2 มาทำการตรวจสอบโดยใช้โปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์ ด้วยภาษา C++ อยู่ใน class ชื่อ CExtract ตามขั้นตอนดังนี้

- ทำการดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษใบนั้น ตามหัวข้อ 4.6.2.2
- เปรียบเทียบพิกัดของจุดปลายซึ่งเป็นข้อมูลคุณลักษณะเฉพาะตัวที่ได้จากหัวข้อ 4.6.4.2 กับพิกัดของจุดปลายที่ได้จากหัวข้อ 4.6.2.2
- ถ้ายังมีพิกัดของจุดปลายที่ไม่ผ่าน ให้ทำการดึงข้อมูลคุณลักษณะตามหัวข้อ 4.6.2.2 ใหม่ โดยเปลี่ยนค่าจุดเริ่มเปลี่ยน (Threshold) แล้วนำมาเปรียบเทียบกับพิกัดของจุดปลายในหัวข้อ 4.6.4.2 ที่ไม่ผ่านใหม่ จนกระทั่งพิกัดของจุดปลายผ่านหมดทุกจุด หรือใช้ค่าจุดเริ่มเปลี่ยนไปแล้ว 5 ค่า
- ตัดสินว่าคุณลักษณะเฉพาะตัวของกระดาษใบนี้ถูกต้องหรือไม่ โดยดูจากจำนวนพิกัดของจุดปลายที่ผ่านเทียบกับจำนวนพิกัดของจุดปลายทั้งหมด

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ ได้นำเสนอระบบเลือกตั้งที่ได้นำเอาการเข้ารหัสลับแบบกุญแจสาธารณะมาใช้กับบัตรเลือกตั้ง โดยได้มีการออกแบบระบบเลือกตั้งใหม่ พัฒนาต้นแบบของระบบเลือกตั้งตามที่ได้ออกแบบไว้ และได้พัฒนาโปรแกรมเพื่อใช้กับต้นแบบระบบเลือกตั้ง

5.1.1 การออกแบบระบบเลือกตั้ง

บัตรเลือกตั้งที่ใช้ในการเลือกตั้งจะมีข้อมูลประจำบัตรเลือกตั้ง ซึ่งประกอบด้วยข้อมูลที่มาของบัตรและข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์บัตรใบนั้น ข้อมูลประจำบัตรเลือกตั้งจะถูกนำมาเข้ารหัสลับด้วยวิธี RSA แล้วแปลงเป็นรหัสแท่ง ก่อนพิมพ์ลงบนบัตรเลือกตั้งเพื่อให้สะดวกต่อการตรวจสอบ มีระบบต้นขั้ว ลายพิมพ์นิ้วมือของผู้ใช้สิทธิ์เลือกตั้ง พร้อมกับมีการปรับกระบวนการเลือกตั้งให้สอดคล้องกัน เพื่อป้องกันการปลอมแปลงบัตรเลือกตั้งและการทุจริตที่อาจเกิดขึ้นในขั้นตอนต่าง ๆ โดยไม่ทำให้การดำเนินงานเป็นไปด้วยความยากลำบากหรือซับซ้อนจนเกินไปโดยเฉพาะอย่างยิ่งในระดับผู้ปฏิบัติงาน การใช้ระบบการตรวจสอบเอกสารต่าง ๆ ด้วยการสุ่มตรวจพร้อมการบันทึกรายงานอย่างเป็นระบบในทุกระดับชั้น จะทำให้สามารถตรวจสอบการทุจริตของเจ้าหน้าที่ดำเนินการที่เกี่ยวข้องได้ แต่ในขณะเดียวกัน ก็ให้การปกป้องเจ้าหน้าที่ที่ปฏิบัติหน้าที่โดยสุจริตตามขั้นตอนที่กำหนดได้ด้วย

5.1.2 ต้นแบบระบบเลือกตั้ง

1. ลักษณะของบัตรเลือกตั้ง

บัตรเลือกตั้งที่ใช้ในการเลือกตั้งจะใช้พื้นที่ส่วนหนึ่งของบัตรสำหรับดึงข้อมูลของกระดาษที่ใช้พิมพ์บัตรเลือกตั้งใบนั้น ๆ มาเข้ารหัสลับ พื้นที่ในส่วนนี้มีขนาดเล็กเพียง 3.2 x 1.6 ตารางนิ้ว กระดาษที่ใช้พิมพ์บัตรเลือกตั้งจะมีวัตถุลักษณะเป็นเส้นฝังอยู่ ซึ่งพิกัดของจุดปลายของเส้นเหล่านี้จะถูกใช้เป็นคุณลักษณะเฉพาะตัวของกระดาษ

2. การพิมพ์รหัสลับบนบัตรเลือกตั้ง

การพิมพ์รหัสลับบนบัตรเลือกตั้ง จะใช้สแกนเนอร์ในการสแกนบัตรเลือกตั้ง และใช้เครื่องไมโครคอมพิวเตอร์ในการดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษ การเข้ารหัสลับ และการพิมพ์รหัสแท่ง

3. **การตรวจสอบความถูกต้องของบัตรเลือกตั้ง**
   - ที่หน่วยเลือกตั้งซึ่งเป็นที่ลงคะแนน จะใช้เครื่องถอดรหัสลับขนาดเล็กที่พัฒนาขึ้นด้วยไมโครคอนโทรลเลอร์ตระกูล 8051 เพื่ออ่านรหัสแท่งและถอดรหัสลับ
   - ที่เขตเลือกตั้งซึ่งเป็นที่นับคะแนนจะใช้สแกนเนอร์ในการสแกนบัตรเลือกตั้ง ใช้เครื่องอ่านรหัสแท่งในการอ่านรหัสแท่ง และใช้เครื่องไมโครคอมพิวเตอร์ในการถอดรหัสลับ การดึงและตรวจสอบข้อมูลคุณลักษณะเฉพาะตัวของกระดาษ

5.1.3 การพัฒนาโปรแกรมเพื่อใช้กับต้นแบบระบบเลือกตั้ง

โปรแกรมที่พัฒนาขึ้นเพื่อใช้กับต้นแบบระบบเลือกตั้ง ประกอบด้วย

1. **โปรแกรมการสร้างและบันทึกกุญแจ**

เป็นโปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์เพื่อใช้ในการสร้างกุญแจที่ใช้ในการเข้าและถอดรหัสลับด้วยวิธี RSA โดยกุญแจที่สร้างสามารถบันทึกเก็บเป็นแฟ้มข้อมูลได้

2. **โปรแกรมการพิมพ์รหัสลับบนบัตรเลือกตั้ง**

เป็นโปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์เพื่อใช้ในขั้นตอนการพิมพ์รหัสลับบนบัตรเลือกตั้ง ประกอบด้วยการเก็บค่าพิกัดของจุดปลายของวัตถุที่ฝังอยู่ในกระดาษที่ใช้พิมพ์บัตรเลือกตั้ง เพื่อใช้เป็นคุณลักษณะเฉพาะตัวของกระดาษ การเข้ารหัสลับด้วยวิธี RSA การแปลงรหัสลับเป็นรหัสแท่ง การพิมพ์รหัสแท่งบนบัตรเลือกตั้ง

3. **โปรแกรมการถอดรหัสลับข้อมูลประจำบัตรเลือกตั้งที่หน่วยเลือกตั้ง**

เป็นโปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอนโทรลเลอร์ตระกูล 8051 เพื่อใช้ในขั้นตอนการตรวจสอบความถูกต้องของบัตรเลือกตั้งที่หน่วยเลือกตั้งซึ่งเป็นที่ลงคะแนน การถอดรหัสลับสามารถทำได้ในเวลาประมาณ 9 วินาที ซึ่งเป็นเวลาที่เร็วเพียงพอที่จะนำไปใช้งานจริง

4. **โปรแกรมการตรวจสอบความถูกต้องของบัตรเลือกตั้งที่เขตเลือกตั้ง**

เป็นโปรแกรมที่เขียนขึ้นบนเครื่องไมโครคอมพิวเตอร์เพื่อใช้ในขั้นตอนการตรวจสอบความถูกต้องของบัตรเลือกตั้งที่เขตเลือกตั้งซึ่งเป็นที่นับคะแนน ประกอบด้วยการถอดรหัสลับด้วยวิธี RSA การเก็บค่าพิกัดของจุดปลายของวัตถุเช่นเดียวกับโปรแกรมการพิมพ์รหัสลับบนบัตรเลือกตั้ง ซึ่งนำมาใช้ในการตรวจสอบกับข้อมูลคุณลักษณะที่ได้จากการถอดรหัสลับ การตรวจสอบบัตรสามารถทำได้อย่างถูกต้องทุกใบ

5.2 การประยุกต์ใช้

1. สลากกินแบ่งรัฐบาล

การเข้ารหัสลับสามารถนำไปใช้กับสลากกินแบ่งรัฐบาลได้ โดยดึงคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์สลากมารวมกับหมายเลขสลาก วันที่ออกรางวัล และข้อมูลสำคัญอื่น ๆ แล้วนำมาเข้ารหัสลับ และพิมพ์ลงบนสลาก ทำให้สลากไม่สามารถปลอมแปลงได้ สำหรับกระดาษที่ใช้พิมพ์สลากนั้น สามารถใช้กระดาษที่ใช้กันอยู่ในปัจจุบันได้ เนื่องจากมีวัตถุชิ้นเล็ก ๆ ฝังอยู่แล้ว

2. โฉนดที่ดิน

นอกจากสลากกินแบ่งรัฐบาลแล้ว การเข้ารหัสลับสามารถนำมาใช้กับโฉนดที่ดินได้ด้วย โดยดึงเอาข้อมูลคุณลักษณะเฉพาะตัวของกระดาษที่ใช้พิมพ์โฉนด มารวมกับข้อมูลสำคัญของโฉนดที่ดินใบนั้น เช่น หมายเลขโฉนด ขนาดที่ดิน ตำแหน่งที่ตั้ง จากนั้นนำมาเข้ารหัสลับ แล้วพิมพ์ลงบนโฉนดที่ดินใบนั้น วิธีนี้จะทำให้การตรวจสอบโฉนดทำได้โดยง่าย ไม่จำเป็นต้องใช้ต้นขั้ว

5.3 ข้อเสนอแนะ

1. ระบบเลือกตั้งที่ได้นำเสนอนี้ ต้องใช้สแกนเนอร์เพื่อสแกนกระดาษที่ใช้พิมพ์บัตรเลือกตั้งทั้งในขั้นตอนการพิมพ์รหัสลับบนบัตรเลือกตั้ง และขั้นตอนการตรวจสอบบัตรเลือกตั้ง ทำให้ต้องเสียเวลาในการสแกนและต้องมีอุปกรณ์ในการสแกนคือสแกนเนอร์ ดังนั้น จึงอาจเปลี่ยนวิธีการพิมพ์รหัสลับบนบัตรเลือกตั้งหรือการตรวจสอบบัตรเลือกตั้งใหม่โดยไม่ต้องใช้สแกนเนอร์ เช่น ในการตรวจสอบบัตรเลือกตั้ง อาจใช้การสร้างภาพกระดาษที่ใช้พิมพ์บัตรเลือกตั้งจากข้อมูลคุณลักษณะที่ได้จากการถอดรหัสลับข้อมูลประจำบัตรเลือกตั้ง แล้วให้เจ้าหน้าที่เปรียบเทียบลักษณะภาพที่สร้างขึ้นกับกระดาษจริง

2. เครื่องถอดรหัสลับที่หน่วยเลือกตั้ง ซึ่งเป็นเครื่องถอดรหัสลับที่พัฒนาขึ้นโดยใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 ควรมีจอแสดงผลที่ใหญ่เพียงพอเพื่อให้สะดวกในการอ่านค่าขณะทำการตรวจสอบ โดยอาจมีลักษณะคล้ายกับจอแสดงผลของวิทยุติดตามตัว

3. ในการพิมพ์รหัสลับบนบัตรเลือกตั้ง ต้องมีพื้นที่สำหรับดึงข้อมูลคุณลักษณะเฉพาะตัวของกระดาษเพื่อใช้เป็นข้อมูลประจำบัตรเลือกตั้งใบนั้น พื้นที่ดังกล่าวต้องมีการพิมพ์กรอบเตรียมไว้สำหรับใช้ดึงข้อมูล ทำให้ต้องเสียเวลาในการพิมพ์ หากสามารถดึงข้อมูลคุณลักษณะเฉพาะตัวจากกระดาษได้โดยไม่จำเป็นต้องพิมพ์กรอบก่อน ก็จะช่วยลดเวลาในการพิมพ์ลงได้

รายการอ้างอิง

1. RSA Laboratories. Frequently Asked Questions About Today's Cryptography version 4.1 [Online]. RSA Data Security, 1995. Available from: http://www.rsasecurity.com/rsalabs/faq/index.html
2. Denning, D. E. Cryptography and data security. United States of America : Addison-Wesley Publishing, 1982.
3. Koc, C. K. High-Speed RSA Implementation [Computer file]. Technical Report TR-201 version 2.0, RSA Laboratories, 1994. Available from: http://www.rsasecurity.com/rsalabs/technotes/
4. สุนทร วิทูสุรพจน์. การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2537.
5. Intel Corporation. MCS 51 Microcontroller Family User's Manual [Computer file]. Intel Corporation, 1994. Available from: http://developer.intel.com/design/mcs51/manuals/272383.htm
6. Jang, B. K., and Chin, R. T. One-Pass Parallel Thinning: Analysis, Properties, and Quantitative Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 14 No. 11 (November 1992) : 1129-1140.
7. Thym Infoware. Code 128 Barcode Details [Online]. Thym Infoware, (n.d.). Available from: http://www.in-barcode.com/sym_c128.html
8. IDAutomation.com. Code 128 / USS Code-128 Barcode FAQ [Online]. IDAutomation.com, (n.d.). Available from: http://www.idautomation.com/code128faq.html
9. Altek Instruments. Code 128 Barcode Specification [Online]. Altek Instruments, 2001. Available from: http://www.barcodeman.com/info/c128.php3
10. Adams, R. Bar Code 1 Code 128 Specification Page [Online]. Adams Communications, 2001. Available from: http://www.barcode-1.net/pub/russadam/128code.html
11. Schneier, B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed. the United Stated of America : John Wiley & Sons, 1996.

12. Gutmann, P. Cryptlib Security Toolkit Version 2.1 final beta [Computer Software]. Peter Gutmann, 1999. Available from: http://www.cs.auckland.ac.nz/~pgut001/cryptlib/

13. ประเสริฐ ฉอเรืองวิวัฒน์. การรู้จำตัวอักษรเขียนภาษาไทยโดยการวิเคราะห์ลักษณะบ่งความต่าง. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2541.

14. Heinz, W. W. MCS-51 Microcontroller Family Macro Assembler ASEM-51 Version 1.2 [Computer program]. W. W. Heinz, 1996. Available from: http://www.muenchen.roses.de/~allinger/asem-51/home.htm

15. Tarvydas-Sanford Controls. TS Controls Emulator 8051 Evaluation Version 1.00 [Computer program]. Tarvydas-Sanford Controls, 1997. Available from: http://www.tscontrols.com

บรรณานุกรม

ภาษาไทย

นิรุธ อำนวยศิลป์. คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0. กรุงเทพมหานคร : ซัคเซส มีเดีย, 2542.

วันสุระ ศรีใสดี. ประยุกต์/อินเตอร์เฟส ไมโครคอนโทรลเลอร์ MCS-51 ภาคปฏิบัติ. กรุงเทพมหานคร : ดวงกมล, 2542.

สุนทร วิทูสุรพจน์. การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2537.

ภาษาอังกฤษ

Bates, J. and Tompkins, T. Using Visual C++ 6. United States of America : Que, 1998.

Garfinkel, S. PGP : Pretty Good Privacy. 1st ed. United States of America : O'Reilly & Associates, 1995.

Gilbert, S. D. and McCarty, B. Visual C++ 6 Programming Blue Book. United States of America : Coriolis Group, 1999.

Gonzalez, R. C. and Woods, R. E. Digital Image Processing. Massachusetts : Addison-Wesley, 1993.

Gregory, K. Special Edition Using Visual C++ 6. United States of America : Que, 1998.

Nelson, P. A. GNU bc version 1.02 [Computer program]. Free Software Foundation, 1992.

TAL Technologies. TAL Bar Code ActiveX Control [Computer Software]. TAL Technologies, 2000. Available from: http://www.taltech.com

ภาคผนวก

ภาคผนวก ก

ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51 ใน MCS 51 Microcontroller Family User's Manual จาก http://developer.intel.com/design/mcs51/manuals/272383.htm

MCS®-51 INSTRUCTION SET

Table 10. 8051 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C,bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

(1)Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Note on instruction set and addressing modes:

Rn — Register R7–R0 of the currently selected Register Bank.

direct — 8-bit internal data location's address. This could be an Internal Data RAM location (0–127) or a SFR [i.e., I/O port, control register, status register, etc. (128–255)].

@Ri — 8-bit internal data RAM location (0–255) addressed indirectly through register R1 or R0.

#data — 8-bit constant included in instruction.

#data 16 — 16-bit constant included in instruction.

addr 16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte Program Memory address space.

addr 11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.

rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to +127 bytes relative to first byte of the following instruction.

bit — Direct Addressed bit in Internal Data RAM or Special Function Register.

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (Continued)				
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12

Mnemonic		Description	Byte	Oscillator Period
LOGICAL OPERATIONS (Continued)				
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (Continued)				
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move External RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24

Table 10. 8051 Instruction Set Summary (Continued)

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (Continued)				
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

Table 11. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A, #data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr, code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A, #data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr, #data
44	2	ORL	A, #data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr, #data
54	2	ANL	A, #data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr, #data
64	2	XRL	A, #data
65	2	XRL	A,data addr

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 11. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

ภาคผนวก ข

ตัวอย่างบัตรเลือกตั้งทั้งด้านหน้าและด้านหลังซึ่งย่อขนาดจาก A3 พิมพ์ลงบนตัวอย่างกระดาษที่สามารถใช้ในการพิมพ์บัตรเลือกตั้งได้

(ต้นขั้วบัตรเลือกตั้ง)

เขตที่ .....

หมายเลขบัตร .....

(ลงชื่อ)

(.....)

กรรมการประจำหน่วยเลือกตั้ง

พิมพ์ลายนิ้วหัวแม่มือขวา
ของผู้มีสิทธิเลือกตั้ง

(เขียนตัวบรรจง) ชื่อ .....

(ปรุจีก)

ไม่ประสงค์จะลงคะแนนให้แก่ผู้สมัครใดเลย
ให้ทำเครื่องหมาย **กากบาท** (เช่น X)
ใน **"ช่องไม่ลงคะแนน"** นี้

ให้ทำเครื่องหมาย **กากบาท** (เช่น X)

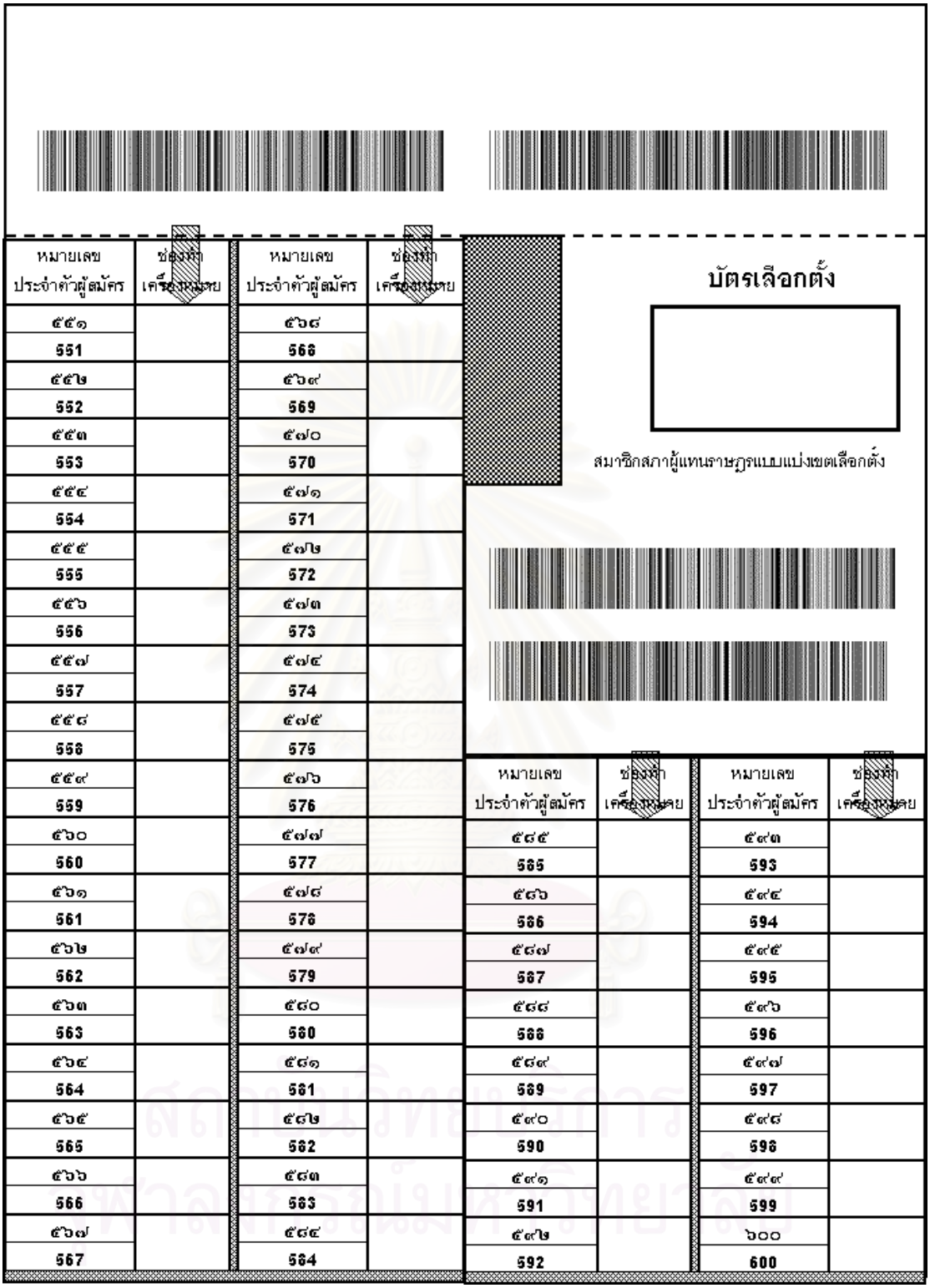
จำนวนเครื่องหมายเดียว

ใน **"ช่องทำเครื่องหมาย"** นี้

หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย	หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย
๕๐๑		๕๐๙	
501		509	
๕๐๒		๕๑๐	
502		510	
๕๐๓		๕๑๑	
503		511	
๕๐๔		๕๑๒	
504		512	
๕๐๕		๕๑๓	
505		513	
๕๐๖		๕๑๔	
506		514	
๕๐๗		๕๑๕	
507		515	
๕๐๘		๕๑๖	
508		516	

หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย	หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย
๕๑๗		๕๓๔	
517		534	
๕๑๘		๕๓๕	
518		535	
๕๑๙		๕๓๖	
519		536	
๕๒๐		๕๓๗	
520		537	
๕๒๑		๕๓๘	
521		538	
๕๒๒		๕๓๙	
522		539	
๕๒๓		๕๔๐	
523		540	
๕๒๔		๕๔๑	
524		541	
๕๒๕		๕๔๒	
525		542	
๕๒๖		๕๔๓	
526		543	
๕๒๗		๕๔๔	
527		544	
๕๒๘		๕๔๕	
528		545	
๕๒๙		๕๔๖	
529		546	
๕๓๐		๕๔๗	
530		547	
๕๓๑		๕๔๘	
531		548	
๕๓๒		๕๔๙	
532		549	
๕๓๓		๕๕๐	
533		550	

ตัวอย่างบัตรเลือกตั้งด้านหน้า



หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย	หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย
๕๕๑		๕๖๘	
551		568	
๕๕๒		๕๖๙	
552		569	
๕๕๓		๕๗๐	
553		570	
๕๕๔		๕๗๑	
554		571	
๕๕๕		๕๗๒	
555		572	
๕๕๖		๕๗๓	
556		573	
๕๕๗		๕๗๔	
557		574	
๕๕๘		๕๗๕	
558		575	
๕๕๙		๕๗๖	
559		576	
๕๖๐		๕๗๗	
560		577	
๕๖๑		๕๗๘	
561		578	
๕๖๒		๕๗๙	
562		579	
๕๖๓		๕๘๐	
563		580	
๕๖๔		๕๘๑	
564		581	
๕๖๕		๕๘๒	
565		582	
๕๖๖		๕๘๓	
566		583	
๕๖๗		๕๘๔	
567		584	

บัตรเลือกตั้ง

สมาชิกสภาผู้แทนราษฎรแบบแบ่งเขตเลือกตั้ง

หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย	หมายเลขประจำตัวผู้สมัคร	ช่องทำเครื่องหมาย
๕๘๕		๕๙๓	
585		593	
๕๘๖		๕๙๔	
586		594	
๕๘๗		๕๙๕	
587		595	
๕๘๘		๕๙๖	
588		596	
๕๘๙		๕๙๗	
589		597	
๕๙๐		๕๙๘	
590		598	
๕๙๑		๕๙๙	
591		599	
๕๙๒		๖๐๐	
592		600	

ตัวอย่างบัตรเลือกตั้งด้านหลัง

ประวัติผู้เขียน

นายศิริพงษ์ ประยูรหงษ์ เกิดเมื่อวันที่ 25 สิงหาคม พ.ศ.2520 ที่จังหวัดกรุงเทพมหานคร เข้าศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต เกียรตินิยมอันดับสอง ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2540 จากนั้น ได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ห้องปฏิบัติการไฟฟ้าสื่อสาร สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2541