วิทยาการรหัสลับ

Cryptography

ผู้ช่วยศาสตราจารย์ ดร.กฤษณพงศ์ สมสุข

คณะเทคโนโลยี
มหาวิทยาลัยราชภัฏอุดรธานี
2562

วิทยาการรหัสลับ

(Cryptography)

ผู้ช่วยศาสตราจารย์ ดร.กฤษณพงศ์ สมสุข

ปร.ด. (วิศวกรรมคอมพิวเตอร์)

คณะเทคโนโลยี

มหาวิทยาลัยราชภัฏอุดรธานี

2562

คำนำ

วิทยาการรหัสลับ (Cryptography) เป็นศาสตร์ที่ใช้สำหรับการรักษาความลับข้อมูลข่าวสารที่มีการรับส่งผ่านระบบเครือข่ายซึ่งมีการพัฒนามาอย่างต่อเนื่องยาวนาน โดยจุดเปลี่ยนแปลงที่สำคัญเกิดขึ้นในปี ค.ศ. 1976 เมื่อวิธฟิลด์ ดิฟฟี่ และ มาร์ติน เฮลแมนได้นำเสนอวิทยาการรหัสลับแบบกุญแจสาธารณะที่สามารถแก้ปัญหาเรื่องการแลกเปลี่ยนกุญแจลับได้เป็นอย่างดี ในเวลาต่อมาจึงเกิดระบบวิทยาการรหัสลับแบบกุญแจสาธารณะประเภทอื่นที่เป็นที่นิยมและมีการใช้งานในปัจจุบันคือ วิทยาการรหัสลับอาร์เอสเอ และวิทยาการรหัสลับเส้นโค้งเชิงวงรี นอกเหนือจากการนำไปประยุกต์ใช้สำหรับการรักษาความลับแล้วยังสามารถนำระบบวิทยาการรหัสลับทั้งสองประเภทมาใช้สำหรับแผนวิธีลายเซ็นดิจิทัลซึ่งเพิ่มความสะดวกและรวดเร็วเนื่องจากเป็นการดำเนินการผ่านระบบคอมพิวเตอร์ ดังนั้นวิทยาการรหัสลับจึงเป็นศาสตร์ที่จำเป็นที่จะต้องศึกษาเพื่อให้ทันต่อการพัฒนาและการเปลี่ยนแปลงของโลกปัจจุบันและอนาคต

เนื้อหาภายในตำราเล่มนี้ประกอบไปด้วย คณิตศาสตร์ที่จำเป็นสำหรับวิทยาการรหัสลับ วิทยาการรหัสลับแบบสมมาตร วิทยาการรหัสลับดีอีเอส วิทยาการรหัสลับเออีเอส วิทยาการรหัสลับแบบกุญแจสาธารณะ วิทยาการการรหัสลับอาร์เอสเอ การแยกตัวประกอบ วิทยาการรหัสลับเส้นโค้งเชิงวงรี และแผนวิธีลายเซ็นดิจิทัล โดยมีการนำองค์ความรู้บางส่วนที่สกัดจากงานวิจัยที่ผู้เขียนได้ทำวิจัยมาอย่างต่อเนื่องบรรจุไว้ในตำราเล่มนี้ประกอบด้วยการเพิ่มความเร็วระบบถอดรหัสลับวิทยาการรหัสลับอาร์เอสเอซึ่งกล่าวไว้ในบทที่ 7 และในบทที่ 8 มีทั้งหมด 2 หัวข้อคือการประมาณค่าเริ่มต้นใหม่ของค่าพารามิเตอร์สำหรับขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์ และขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์ซึ่งทั้งสองขั้นตอนวิธีถูกนำมาใช้สำหรับการคำนวณหากุญแจส่วนตัว

ตำราเล่มนี้เหมาะสำหรับทั้งนักศึกษาสาขาวิชาทางคอมพิวเตอร์ และบุคคลทั่วไปที่สนใจในสายงานที่เกี่ยวข้องกับความปลอดภัยบนระบบเครือข่าย จากประสบการณ์สอนมาหลายปีพบว่านักศึกษามหาวิทยาลัยราชภัฏอุดรธานีส่วนมากนิยมอ่านตำราที่เป็นภาษาไทย และจากประสบการณ์ของผู้เขียนที่ศึกษาเกี่ยวกับวิทยาการรหัสลับมาหลายปีพบว่าปัจจุบันหนังสือที่เกี่ยวกับวิทยาการรหัสลับที่ถูกเขียนเป็นภาษาไทยมีจำนวนน้อย และเนื้อหาบางหัวข้อพบว่าอ่านเข้าใจยากโดยเฉพาะอย่างยิ่งเนื้อหาที่เกี่ยวกับการคำนวณซึ่งส่วนใหญ่จะไม่แสดงวิธีการคำนวณโดยละเอียด จึงไม่เหมาะสำหรับนักศึกษาสาขาวิชาวิศวกรรมคอมพิวเตอร์ และการสื่อสาร ซึ่งโดยส่วนมากมีทักษะการคำนวณไม่ดี ดังนั้นตำราเล่มนี้จึงมุ่งเน้นให้นักศึกษาที่มีพื้นฐานความรู้ทางด้านการคำนวณไม่มากสามารถอ่านเข้าใจ

ได้ และนักศึกษาที่มีพื้นฐานความรู้เดิมดีอยู่แล้วอ่านได้ความรู้มากยิ่งขึ้น โดยผู้เขียนพยายามใช้ภาษาที่ไม่ซับซ้อน อ่านทำความเข้าใจด้วยตนเองได้ง่าย โดยเฉพาะอย่างยิ่งเนื้อหาในส่วนการคำนวณที่เน้นการแสดงวิธีการคำนวณโดยละเอียด อย่างไรก็ตามผู้อ่านตำราเล่มนี้ควรมีความรู้พื้นฐานเกี่ยวกับรหัสคำสั่งพื้นฐานของภาษาโปรแกรมทางคอมพิวเตอร์ เนื่องจากถูกนำมาใช้เป็นส่วนประกอบของชุดคำสั่งเทียมสำหรับอธิบายการทำงานของขั้นตอนวิธี

นอกเหนือจากนั้นผู้เขียนได้เพิ่มรหัสคำสั่งโปรแกรมภาษาจาวาสำหรับบางขั้นตอนวิธีที่ถูกอธิบายเนื้อหาไว้ในบทเรียนเพื่อให้ผู้อ่านได้ฝึกทดลองและปฏิบัติจริงโดยบรรจุไว้ที่ภาคผนวก ก

ผู้เขียนหวังเป็นอย่างยิ่งว่าตำราเล่มนี้จะเป็นประโยชน์แก่นักศึกษาสาขาวิชาวิศวกรรมคอมพิวเตอร์ และการสื่อสาร คณะเทคโนโลยี มหาวิทยาลัยราชภัฏอุดรธานี และผู้สนใจที่จะอ่านทุกท่าน และพร้อมยินดีรับคำติชม ข้อเสนอแนะต่างๆ ที่เป็นประโยชน์เพื่อนำไปปรับปรุงในการจัดพิมพ์ประกอบการเรียนการสอนในครั้งต่อไป

ผู้ช่วยศาสตราจารย์ ดร.กฤษณพงศ์ สมสุข

ปรับปรุงล่าสุด ธันวาคม 2562

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญรูปภาพ

สารบัญตาราง

สารบัญตาราง (ต่อ)

บทที่ 1
บทนำวิทยาการรหัสลับ

วิทยาการรหัสลับ (Cryptography) [1] หมายถึง ศาสตร์ หรือ ศิลป์ ที่มีวัตถุประสงค์เพื่อใช้ในการรักษาข้อมูลข่าวสารให้เป็นความลับซึ่งทำให้บุคคลที่ไม่เกี่ยวข้องไม่สามารถอ่านหรือเข้าใจข้อมูลดังกล่าวได้ ในสมัยก่อนวิทยาการรหัสลับถูกนำมาใช้ในหน่วยงานราชการทางทหารเพื่อใช้ในการปกป้องหรือปิดบังข้อมูลที่เป็นความลับของตนเองเพื่อไม่ให้ฝ่ายข้าศึกสามารถรับรู้ถึงข้อมูลที่แท้จริงได้ ถึงแม้ว่าฝ่ายข้าศึกจะสามารถดักจับข้อมูลได้ก็ตาม (การดักจับข้อมูลหมายถึงทั้งการเห็นและการได้ยิน) เนื่องจากว่าข้อมูลดังกล่าวถูกเข้ารหัสไว้อยู่จึงทำให้การรับส่งข้อมูลข่าวสารถึงกันมีความปลอดภัยมากยิ่งขึ้น

ปัจจุบันการสื่อสารข้อมูลข่าวสารผ่านระบบเครือข่ายอินเทอร์เน็ตกำลังได้รับความนิยมเป็นอย่างมาก และมีการใช้งานกันอย่างแพร่หลายซึ่งปฏิเสธไม่ได้เลยว่าครอบครัวส่วนมากทั่วทั้งโลกมีการใช้งานเครือข่ายอินเทอร์เน็ต แต่เนื่องจากระบบเครือข่ายอินเทอร์เน็ตเป็นช่องทางติดต่อสื่อสารที่ไม่ปลอดภัยทำให้การดักจับข้อมูลโดยผู้ไม่ประสงค์ดีมีโอกาสเกิดขึ้นได้ ซึ่งหากข้อมูลข่าวสารข้างต้นเป็นข้อมูลที่เป็นความลับของหน่วยงานหรือองค์กรต่างๆ จะส่งผลให้เกิดความเสียหายแก่หน่วยงานหรือองค์กรดังกล่าวได้ โดยความเสียหายที่เกิดขึ้นนี้อาจจะส่งผลกระทบโดยตรงต่อตนเอง บุคคลอื่นที่เกี่ยวข้อง หน่วยงาน หรืออาจส่งผลกระทบต่อความมั่นคงของประเทศชาติก็เป็นได้ ดังนั้นการรักษาความปลอดภัยของข้อมูลข่าวสารที่มีการรับส่งผ่านระบบเครือข่ายจึงเป็นสิ่งจำเป็นที่ต้องให้ความสำคัญ ซึ่งวิทยาการรหัสลับเป็นกระบวนการรักษาความปลอดภัยที่เกี่ยวข้องกับระบบคอมพิวเตอร์ที่กำลังได้รับความนิยมเป็นอย่างมากวิธีหนึ่ง โดยหลักการของวิทยาการรหัสลับคือฝั่งผู้ส่งจะทำการเข้ารหัสข้อมูลข่าวสารก่อนที่จะมีการส่งผ่านช่องสัญญาณที่ไม่ปลอดภัย ในทางกลับกันฝั่งผู้รับจะทำการถอดรหัสข้อมูลที่ถูกเข้ารหัสโดยผู้ส่งและได้เป็นข้อมูลข่าวสารที่ถูกต้อง

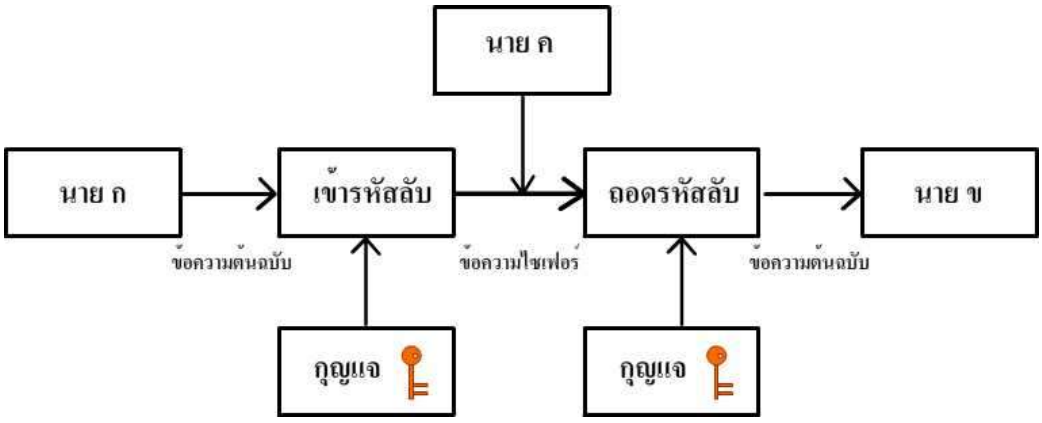
1. วิทยาการรหัสลับเบื้องต้น

วิทยาการรหัสลับมีกระบวนการหลักที่สำคัญอยู่ 2 กระบวนการ คือ การเข้ารหัสลับ (Encryption) คือการนำข้อความปกติ หรือ ข้อความต้นฉบับ (Plaintext) ซึ่งเป็นข้อความที่มนุษย์สามารถอ่านได้เข้าใจมาทำการเข้ารหัสด้วยกุญแจ (k_1) ได้เป็นข้อความไซเฟอร์ (Ciphertext) ซึ่งเป็นข้อความที่มนุษย์ไม่สามารถอ่านเข้าใจได้ อีกกระบวนการคือ การถอดรหัสลับ (Decryption) คือการนำข้อความไซเฟอร์มาทำการถอดรหัสด้วยกุญแจ k_1 สำหรับกรณีที่เป็นวิทยาการรหัสลับแบบสมมาตร

หรือกุญแจ k_2 สำหรับกรณีที่เป็นวิทยาการรหัสลับแบบอสมมาตร ซึ่ง k_2 จะมีความสัมพันธ์ทางคณิตศาสตร์กับ k_1 เพื่อให้ได้กลับมาซึ่งข้อความต้นฉบับ

ตัวอย่างต่อไปนี้อธิบายถึงสถานการณ์ที่มีการนำวิทยาการรหัสลับมาประยุกต์ใช้งานเพื่อใช้สำหรับการรักษาความปลอดภัยข้อมูล ข่าวสาร หรือสารสนเทศที่สำคัญ

จากรูปที่ 1.1 สมมติว่านาย ก และ นาย ข มีความประสงค์จะแลกเปลี่ยนข้อมูลข่าวสารที่เป็นความลับผ่านช่องสัญญาณที่ไม่ปลอดภัย หากมีการรับส่งข้อมูลลับโดยตรงจะมีความเป็นไปได้สูงที่นาย ค ซึ่งไม่ใช่บุคคลที่เกี่ยวข้องกับการสนทนาครั้งนี้ (ผู้ไม่ประสงค์ดี) สามารถดักฟังข้อมูลลับได้ซึ่งทำให้ทราบข้อมูลนั้นได้ในทันทีและส่งผลให้ข้อมูลดังกล่าวไม่เป็นความลับอีกต่อไป เพราะฉะนั้นจึงแก้ปัญหาโดยการนำวิทยาการรหัสลับมาประยุกต์ใช้งาน โดยมีหลักการคือหาก นาย ก จะส่งข้อมูลลับให้นาย ข การดำเนินการเริ่มจากนาย ก นำข้อมูลลับมาทำการเข้ารหัสลับด้วยกุญแจ k_1 ที่ซึ่งได้กำหนดไว้ก่อนหน้านี้และได้ผลลัพธ์เป็นข้อความไซเฟอร์ หลังจากนั้นส่งข้อความไซเฟอร์ผ่านช่องสัญญาณที่ไม่ปลอดภัยไปให้นาย ข เมื่อนาย ข ได้รับข้อความไซเฟอร์แล้วก็จะทำการถอดรหัสลับด้วยกุญแจ k_1 (หรือ k_2) ได้เป็นข้อความต้นฉบับกลับมาเช่นเดิม จากที่กล่าวมาข้างต้นถึงแม้ว่านาย ค สามารถดักฟังข้อมูลได้ แต่นาย ค ก็ไม่สามารถอ่านข้อความดังกล่าวได้ เนื่องจากว่าข้อมูลที่ได้มานั้นเป็นข้อความไซเฟอร์



รูปที่ 1.1 ตัวอย่างการเข้ารหัสลับและถอดรหัสลับผ่านช่องสัญญาณที่ไม่ปลอดภัย

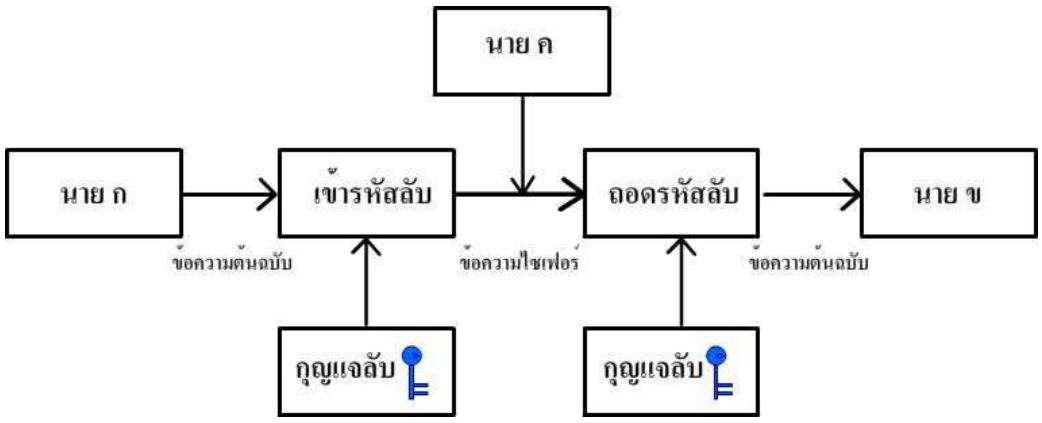
2. ประเภทวิทยาการรหัสลับ

วิทยาการรหัสลับสามารถแบ่งออกเป็น 2 ประเภทคือวิทยาการรหัสลับแบบสมมาตร และวิทยาการรหัสลับแบบอสมมาตรซึ่งแต่ละประเภทจะมีความแตกต่างกันที่ลักษณะของกุญแจและ

ขั้นตอนวิธีที่ใช้ในการเข้ารหัสลับและการถอดรหัสลับโดยแต่ละประเภทจะมีข้อดีและข้อเสียที่แตกต่างกันออกไป ดังนี้

2.1 วิทยาการรหัสลับแบบสมมาตร (Symmetric key Cryptography)

วิทยาการรหัสลับแบบสมมาตร หรือการเข้ารหัสลับด้วยกุญแจลับ (Secret key Encryption) คือการที่นาย ก และ นาย ข ใช้กุญแจดอกเดียวกันสำหรับการเข้ารหัสลับและถอดรหัสลับ เรียกกุญแจประเภทนี้ว่ากุญแจลับ (Secret key) ซึ่งหมายถึงกุญแจที่ผู้รับและผู้ส่งใช้สำหรับการเข้ารหัสลับและถอดรหัสลับและเป็นค่าที่ต้องเก็บเป็นความลับ ดังนั้นจึงมีเพียงนาย ก และ นาย ข หรือบุคคลอื่นที่เกี่ยวข้องเท่านั้นที่สามารถทราบกุญแจนี้ได้ เพราะหากมีผู้ไม่ประสงค์ดีทราบกุญแจลับแล้วจะสามารถถอดรหัสข้อมูลที่ดักฟังมาได้ในทันที ข้อดีของการเข้ารหัสลับด้วยกุญแจลับคือสามารถทำการเข้ารหัสลับ และถอดรหัสลับได้อย่างรวดเร็ว และมีความปลอดภัยสูงหากขั้นตอนวิธีที่เลือกใช้มีประสิทธิภาพเพียงพอ ข้อเสียคือเนื่องจากต้องใช้กุญแจดอกเดียวกันในการเข้ารหัสลับ และถอดรหัสลับเพราะฉะนั้นจึงเกิดปัญหาเกี่ยวกับการแลกเปลี่ยนกุญแจลับ หากส่งกุญแจลับผ่านช่องสัญญาณที่ไม่ปลอดภัยจะมีความเสี่ยงสูงที่นาย ค จะสามารถดักจับได้ ซึ่งนาย ก และ นาย ข อาจจะแก้ปัญหาโดยการนัดพบแบบตัวต่อตัวเพื่อทำการแลกเปลี่ยนกุญแจลับ แต่หากนาย ก และ นาย ข อยู่ไกลกันมาก เช่นอยู่กันคนละซีกโลก การนัดพบกันแบบตัวต่อตัวก็อาจจะไม่ใช่วิธีที่เหมาะสม โดยตัวอย่างการสนทนาระหว่างนาย ก และนาย ข โดยใช้วิทยาการรหัสลับแบบสมมาตร และมีนาย ค เป็นผู้เข้ามาแทรกแซงการสนทนาแสดงดังรูปที่ 1.2

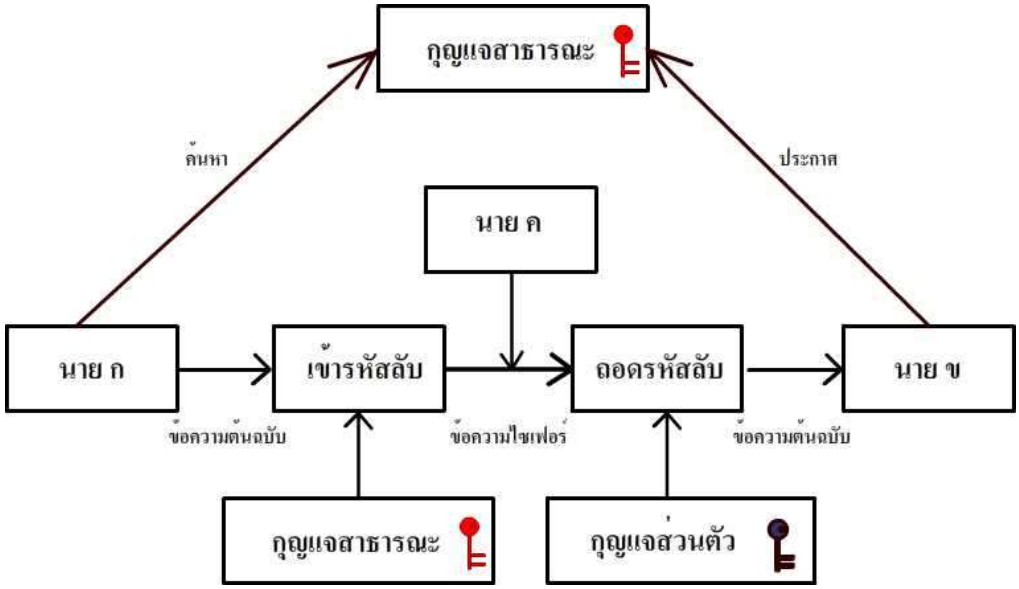


รูปที่ 1.2 ตัวอย่างการเข้ารหัสลับและการถอดรหัสลับโดยใช้วิทยาการรหัสลับแบบสมมาตร

2.2 วิทยาการรหัสลับแบบอสมมาตร (Asymmetric key Cryptography)

วิทยาการรหัสลับแบบอสมมาตร หรือการเข้ารหัสลับด้วยกุญแจสาธารณะ (Public key Encryption) คือการที่นาย ก และ นาย ข ใช้กุญแจคนละดอกในการเข้ารหัสลับและ ถอดรหัสลับ สมมติว่านาย ก ต้องการส่งข้อมูลลับไปให้นาย ข นาย ก จะต้องใช้กุญแจสาธารณะ (Public key) ของนาย ข สำหรับการเข้ารหัสข้อมูล โดยกุญแจสาธารณะหมายถึงค่ากุญแจที่ถูกประกาศเป็นสาธารณะ ดังนั้นไม่ว่าบุคคลใดก็สามารถทราบค่ากุญแจสาธารณะของนาย ข เพราะนาย ข ประกาศไว้อย่างเปิดเผย หลังจากนาย ก เข้ารหัสข้อมูลด้วยกุญแจสาธารณะของนาย ข แล้วจะได้ผลลัพธ์เป็นข้อความไซเฟอร์ และส่งข้อความไซเฟอร์นี้ไปให้นาย ข ผ่านช่องสัญญาณที่ไม่ปลอดภัย เมื่อนาย ข ได้รับข้อความไซเฟอร์ที่ส่งมาจากนาย ก แล้ว จะดำเนินการถอดรหัสข้อความดังกล่าวด้วยกุญแจส่วนตัว (Private key) ซึ่งหมายถึงค่ากุญแจที่ถูกเก็บไว้เป็นความลับเฉพาะผู้สร้างกุญแจและผู้เกี่ยวข้องเท่านั้น ดังนั้นข้อความไซเฟอร์ที่ถูกส่งมาจากนาย ก จะมีเพียงนาย ข เท่านั้นที่สามารถถอดรหัสลับได้ หลังจากถอดรหัสแล้วจะได้ข้อความต้นฉบับกลับมา โดยที่กุญแจสาธารณะ และกุญแจส่วนตัวจะต้องมีความสัมพันธ์ทางคณิตศาสตร์ซึ่งกันและกัน และมีความสัมพันธ์แบบหนึ่งต่อหนึ่งเท่านั้น เช่นหากใช้กุญแจสาธารณะ e ก็ต้องใช้กุญแจส่วนตัว d เท่านั้น ข้อดีของการเข้ารหัสลับด้วยกุญแจสาธารณะ คือ ถึงแม้นาย ค จะทราบกุญแจสาธารณะของนาย ข แต่นาย ค ไม่สามารถถอดรหัสข้อมูลที่นาย ก ส่งไปให้นาย ข ได้เนื่องจากต้องใช้กุญแจส่วนตัวสำหรับการถอดรหัสลับเท่านั้น และหากนาย ค ต้องการที่จะคำนวณหาค่ากุญแจส่วนตัวจากกุญแจสาธารณะที่ตนเองมีทำได้ยากมาก เพราะนาย ค จำเป็นต้องทำการแก้ปัญหาทางคณิตศาสตร์ของจำนวนเต็มที่มีขนาดใหญ่มหาศาล เช่น การแยกตัวประกอบ และการหาเลขยกกำลัง เป็นต้น ข้อเสียคือเนื่องจากเป็นขั้นตอนวิธีที่มีการคำนวณที่ซับซ้อนจึงส่งผลให้ใช้เวลานานในการเข้ารหัสลับ และถอดรหัสลับ อีกกรณีหนึ่งคือถึงแม้นาย ค จะไม่สามารถถอดรหัสข้อมูลของนาย ก ได้แต่นาย ค สามารถดำเนินการได้โดยวิธีอื่น เช่นการสวมรอยเป็นนาย ก โดยนาย ค ใช้กุญแจสาธารณะของนาย ข ในการเข้ารหัสลับข้อมูลของตนเองและส่งไปให้นาย ข เมื่อนาย ข ถอดรหัสข้อมูลออกมาจะได้รับข้อความต้นฉบับของนาย ค แทน โดยที่นาย ข อาจจะเข้าใจว่าข้อความดังกล่าวถูกส่งมาจากนาย ก เป็นต้น

รูปที่ 1.3 แสดงตัวอย่างการส่งข้อความลับจากนาย ก ไปยังนาย ข โดยใช้วิทยาการรหัสลับแบบอสมมาตร และมีนาย ค เป็นผู้เข้ามาแทรกแซง จากรูปนาย ข ได้ประกาศกุญแจสาธารณะไว้อย่างเปิดเผยและเก็บกุญแจส่วนตัวไว้เป็นความลับเพียงผู้เดียว นาย ก จึงสามารถนำข้อความลับที่ประสงค์จะส่งไปยังนาย ข มาเข้ารหัสลับโดยใช้กุญแจสาธารณะของนาย ข ได้เป็นข้อความไซเฟอร์ซึ่งเป็นส่วนที่จะส่งไปยังนาย ข ในระหว่างทางสมมตินาย ค สามารถดักจับข้อความไซเฟอร์นี้ได้ แต่อย่างไรก็ตามนาย ค ไม่สามารถรับรู้ความหมายของข้อความต้นฉบับเนื่องจากไม่มีกุญแจส่วนตัว



รูปที่ 1.3 ตัวอย่างการเข้ารหัสลับและการถอดรหัสลับโดยใช้วิทยาการรหัสลับแบบอสมมาตร

3. ประโยชน์วิทยาการรหัสลับ

ปัจจุบันวิทยาการรหัสลับไม่ได้ถูกนำมาใช้สำหรับการเข้ารหัสลับและถอดรหัสลับเพียงเท่านั้น แต่ยังถูกนำมาใช้เพื่อเพิ่มความปลอดภัยในด้านอื่นด้วยดังต่อไปนี้

3.1 การรักษาความลับ (Confidentiality) คือการเก็บรักษาข้อมูลส่วนตัวให้เป็นความลับ ไม่ให้บุคคลอื่นที่ไม่เกี่ยวข้องสามารถเข้าถึงข้อมูลดังกล่าวนี้ได้ เช่นข้อมูลด้านการเงิน ข้อมูลผลการเรียนของนักศึกษา เป็นต้น

3.2 การรักษาบูรณภาพของข้อมูล (Data integrity) คือ การรักษาข้อมูลไม่ให้ถูกเปลี่ยนแปลงแก้ไขโดยบุคคลอื่นที่ไม่เกี่ยวข้อง โดยหากข้อมูลดังกล่าวถูกเปลี่ยนแปลงแก้ไขไปจะต้องสามารถตรวจสอบได้ และปรับปรุงแก้ไขให้เป็นเช่นเดิมในทันที โดยที่การรักษาบูรณภาพของข้อมูลมีความแตกต่างกับการรักษาความลับตรงที่การรักษาบูรณภาพของข้อมูลไม่จำเป็นต้องเก็บเป็นความลับ อาจจะเผยแพร่สู่สาธารณะชนก็เป็นได้ เพียงแต่ต้องไม่ให้บุคคลอื่นที่ไม่เกี่ยวข้องสามารถเข้ามาลบแก้ไข หรือเปลี่ยนแปลงข้อมูลดังกล่าวนี้ได้ ตัวอย่างที่พบเห็นบ่อยๆคือ มีผู้ไม่ประสงค์ดีเจาะระบบมาเปลี่ยนแปลงข้อมูลที่หน้าเว็บเพจของผู้บริหารระดับสูงซึ่งทำให้บุคคลนั้นเกิดความอับอาย และเสื่อมเสียชื่อเสียง เป็นต้น

3.3 การพิสูจน์ตัวตน (Authentication) คือ การพิสูจน์ได้ว่าบุคคลที่กำลังติดต่ออยู่ด้วยนั้นเป็นบุคคลนั้นจริงๆ ไม่ใช่บุคคลอื่นที่สวมรอยเข้ามา ตัวอย่างของการพิสูจน์ตัวตนที่พบเห็นในชีวิตประจำวันคือ บัตรประชาชน สมมติว่านาย ก ไม่รู้จักกับนาย ข มาก่อน แต่นาย ก ต้องการทราบ

ว่าบุคคลที่นาย ก กำลังสนทนาอยู่ด้วยนั้นเป็นนาย ข จริง นาย ข สามารถพิสูจน์ตัวตนได้โดยแสดงบัตรประชาชนของตนเองให้นาย ก เห็น ซึ่งทำให้นาย ก มั่นใจว่าบุคคลที่ตนเองกำลังสนทนาด้วยนี้เป็นนาย ข จริง

3.4 การป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) คือการที่ผู้ส่ง หรือผู้รับไม่สามารถปฏิเสธได้ว่าตนเองไม่ได้เป็นผู้ส่ง หรือผู้รับข้อมูลดังกล่าว ตัวอย่างเช่น นาย ก ต้องการสั่งซื้อสินค้าของนาย ข นาย ก จึงโอนเงินไปให้นาย ข และเก็บใบเสร็จการโอนเงินไว้ เพราะฉะนั้นหากนาย ข ปฏิเสธว่าตนเองยังไม่เคยรับเงินจากนาย ก เลย นาย ก สามารถใช้ใบเสร็จการโอนเงินดังกล่าวนี้ในการยืนยันว่าตนเองได้โอนเงินไปเป็นที่เรียบร้อยแล้ว

4. การโจมตีระบบรหัสลับ

การโจมตีระบบรหัสลับ คือการที่ผู้ไม่ประสงค์ดีพยายามที่จะหาวิธีที่ใช้สำหรับการโจมตีระบบรหัสลับเพื่อให้ได้มาซึ่งข้อความต้นฉบับ กุญแจลับ หรือกุญแจส่วนตัว ซึ่งระบบรหัสลับหมายถึงขั้นตอนวิธีต่าง ๆ ทางวิทยาการรหัสลับซึ่งมีเป็นจำนวนมากโดยจะกล่าวถึงโครงสร้างของแต่ละขั้นตอนวิธีในบทถัด ๆ ไป สำหรับการโจมตีระบบรหัสลับแบ่งออกเป็นหลายวิธี แต่ละวิธีจะมีประสิทธิภาพที่แตกต่างกันออกไป ดังนี้

4.1 การโจมตีแบบตะลุย (Brute force Attack) หรือ เรียกอีกอย่างว่า **การค้นหาอย่างละเอียด (Exhaustive search)** คือการที่ผู้ไม่ประสงค์ดีพยายามที่จะโจมตีระบบรหัสลับโดยใช้กุญแจที่เป็นไปได้ทั้งหมด ซึ่งวิธีนี้จะมีประสิทธิภาพที่สูงมากหากใช้กับขั้นตอนวิธีที่ไม่แข็งแกร่ง ซึ่งโดยส่วนมากขั้นตอนวิธีลักษณะนี้จะเป็นขั้นตอนวิธีที่มีกุญแจขนาดไม่ใหญ่มาก เช่นรหัสซีซาร์ซึ่งมีกุญแจที่เป็นไปได้ทั้งหมดเพียง 26 ค่า เพราะฉะนั้นหากผู้ไม่ประสงค์ดีใช้วิธีการโจมตีแบบตะลุยโดยใช้กุญแจที่แตกต่างกันทั้งหมด 26 ค่ามาทดลองถอดรหัสข้อมูลทีละค่า จะทำให้ได้ข้อความต้นฉบับกลับมาได้ไม่ยาก แต่หากนำไปใช้กับขั้นตอนวิธีที่มีประสิทธิภาพสูงมากยิ่งขึ้น เช่น ขั้นตอนวิธีดีอีเอสซึ่งใช้กุญแจขนาด 56 บิต ดังนั้นจำนวนกุญแจที่เป็นไปได้ทั้งหมดคือ $2^{56} \approx 7.21 \times 10^{16}$ ซึ่งพบว่าเป็นจำนวนที่สูงมาก ฉะนั้นหากใช้วิธีการโจมตีแบบตะลุยจำเป็นต้องใช้เวลาอย่างมหาศาล และหากเครื่องมือที่ใช้ช่วยในการประมวลผลหรือคำนวณมีประสิทธิภาพไม่เพียงพออาจใช้เวลาหลายปีกว่าจะได้ข้อความต้นฉบับที่ถูกต้อง ตารางที่ 1.1 แสดงตัวอย่างการใช้การโจมตีแบบตะลุยที่ใช้กุญแจแต่ละขนาด และเวลาที่ใช้ในการถอดรหัสของแต่ละขั้นตอนวิธีซึ่งพบว่าเวลาที่ใช้สำหรับการประมวลผลเพิ่มสูงขึ้นในกรณีที่กุญแจมีขนาดที่ใหญ่มากขึ้น

ตารางที่ 1.1 ตัวอย่างของขนาดกุญแจและเวลาที่ใช้ในการถอดรหัสสำหรับการโจมตีแบบตะลุย

ขนาดกุญแจ	จำนวนกุญแจ	เวลาที่ใช้เมื่อกำหนดให้ในทุกๆ 1 นาทีสามารถถอดรหัสได้ 1, 10^2, 10^4 และ 10^6 ครั้ง			
		1 ครั้ง/นาที	**10^2 ครั้ง/นาที**	**10^4 ครั้ง/นาที**	**10^6 ครั้ง/นาที**
26	26	26 นาที	0.26 นาที	0.0026 นาที	0.000026 นาที
26!	4×10^{26}	7.61×10^{20} ปี	7.61×10^{18} ปี	7.61×10^{16} ปี	7.61×10^{14} ปี
8 บิต	256	4.27 ชั่วโมง	2.56 นาที	0.0256 นาที	2.56×10^{-4} นาที
16 บิต	65536	45 วัน	11 ชั่วโมง	6.55 นาที	3.93 วินาที
32 บิต	4.29×10^9	8162 ปี	81.62 ปี	298 วัน	3 ชั่วโมง
56 บิต	7.21×10^{16}	1.37×10^{11} ปี	1.37×10^9 ปี	1.37×10^7 ปี	1.37×10^5 ปี
112 บิต	5.19×10^{33}	9.87×10^{27} ปี	9.87×10^{25} ปี	9.87×10^{23} ปี	9.87×10^{21} ปี
128 บิต	3.4×10^{38}	6.46×10^{32} ปี	6.46×10^{30} ปี	6.46×10^{28} ปี	6.46×10^{26} ปี
168 บิต	3.74×10^{50}	7.12×10^{44} ปี	7.12×10^{42} ปี	7.12×10^{40} ปี	7.12×10^{38} ปี
256 บิต	1.16×10^{77}	2.21×10^{71} ปี	2.21×10^{69} ปี	2.21×10^{67} ปี	2.21×10^{65} ปี

4.2 การวิเคราะห์รหัสลับ (Cryptanalysis) คือการที่ผู้ไม่ประสงค์ดีพยายามที่จะค้นหาข้อความต้นฉบับ หรือกุญแจลับจากการวิเคราะห์ข้อความไซเฟอร์ที่ตนเองมีอยู่เพื่อให้ได้มาซึ่งข้อความต้นฉบับ หรือกุญแจลับที่ต้องการ หรือวิเคราะห์จากความสัมพันธ์ของข้อความต้นฉบับ และข้อความไซเฟอร์ ในกรณีที่ผู้ไม่ประสงค์ดีทราบของความต้นฉบับด้วย เพื่อให้ได้มาซึ่งกุญแจลับที่ต้องการ

ในการวิเคราะห์รหัสลับนี้ได้ตั้งสมมติฐานไว้ว่าผู้ไม่ประสงค์ดีทราบว่าโครงสร้างของระบบรหัสลับ (Cryptosystem) คืออะไร มิเช่นนั้นหากไม่ทราบถึงโครงสร้างของระบบรหัสลับ การวิเคราะห์รหัสลับจะทำได้ยากมาก สำหรับการวิเคราะห์ระบบสามารถดำเนินการได้โดยใช้หลักการของเคิร์คฮอฟร์ (Kerckhoffs) เรียกหลักการนี้ว่า Kerckhoffs' principle ซึ่งเป็นหลักการที่อธิบายถึงรูปแบบจำลองที่ใช้สำหรับการคุกคามหรือโจมตีขั้นตอนวิธีทางวิทยาการรหัสลับโดยรูปแบบจำลองดังกล่าวถูกเรียกว่าแบบจำลองการคุกคาม (Attack models) แบ่งออกเป็น 4 วิธีดังนี้

1. การโจมตีที่ทราบข้อความไซเฟอร์เท่านั้น (Ciphertext - only Attack) คือการที่ผู้ไม่ประสงค์ดีสามารถเข้าไปครอบครองข้อความไซเฟอร์ได้หลายชุด ซึ่งข้อความไซเฟอร์เหล่านี้ใช้ขั้นตอนวิธีตัวเดียวกันโดยเป้าหมายของการโจมตีประเภทนี้คือได้มาซึ่งข้อความต้นฉบับ และกุญแจลับ

2. การโจมตีที่ทราบข้อความต้นฉบับ (Known Plaintext Attack) คือการที่ผู้ไม่ประสงค์ดีสามารถเข้าไปครอบครองข้อความต้นฉบับ และข้อความไซเฟอร์ที่มีความสัมพันธ์กันได้หลายชุด โดยเป้าหมายของการโจมตีประเภทนี้คือได้มาซึ่งกุญแจลับ

3. การโจมตีแบบเลือกข้อความต้นฉบับได้ (Chosen Plaintext Attack) คือการที่ผู้ไม่ประสงค์ดีสามารถเข้าไปครอบครองเครื่องเข้ารหัสลับได้แบบชั่วคราว การโจมตีรูปแบบนี้ผู้ไม่ประสงค์ดีไม่เพียงแต่สามารถเข้าครอบครองข้อความต้นฉบับเท่านั้น แต่ผู้ไม่ประสงค์ดียังสามารถเลือกรูปแบบของข้อความต้นฉบับได้ และได้ข้อความไซเฟอร์ที่มีความสัมพันธ์กับข้อความต้นฉบับดังกล่าว ทำให้ผู้ไม่ประสงค์ดีสามารถศึกษารูปแบบของข้อความไซเฟอร์ได้ง่ายมากยิ่งขึ้น โดยเป้าหมายของการโจมตีประเภทนี้คือได้มาซึ่งกุญแจลับ

4. การโจมตีแบบเลือกข้อความไซเฟอร์ได้ (Chosen Ciphertext Attack) คือการที่ผู้ไม่ประสงค์ดีสามารถเข้าไปครอบครองเครื่องถอดรหัสลับแบบชั่วคราว การโจมตีรูปแบบนี้ผู้ไม่ประสงค์ดีสามารถเลือกข้อความไซเฟอร์ได้ และได้ข้อความต้นฉบับที่มีความสัมพันธ์กับข้อความไซเฟอร์ดังกล่าว ซึ่งการโจมตีลักษณะนี้เป็นการโจมตีที่มีประสิทธิภาพสูงสุด

5. คณิตศาสตร์เบื้องต้น

คณิตศาสตร์เป็นเครื่องมือที่สำคัญสำหรับวิทยาการรหัสลับ เนื่องจากขั้นตอนวิธีทางรหัสลับสำหรับกระบวนการเข้ารหัสลับและการถอดรหัสลับเกือบทั้งหมดจำเป็นต้องใช้สมการทางคณิตศาสตร์เพื่อแก้ปัญหา อย่างไรก็ตามความยากง่ายของสมการทางคณิตศาสตร์สำหรับแต่ละขั้นตอนวิธีมีความแตกต่างกันออกไป หัวข้อทั้งหมดหลังจากนี้อธิบายพื้นฐานและสมการทางคณิตศาสตร์ที่สำคัญที่จำเป็นต้องนำมาใช้สำหรับการแก้ปัญหาที่เกี่ยวกับวิทยาการรหัสลับ

6. เซต (Set)

เซตคือสิ่งที่ถูกนำมาใช้สำหรับบ่งบอกสมาชิกทั้งหมดภายในกลุ่ม โดยหากสิ่งที่ไม่อยู่ภายในกลุ่มของเซตจะไม่ใช่สมาชิกของเซต ยกตัวอย่างเช่น หากกล่าวถึงเซตของสัตว์แสดงว่า สุนัข แมว ยีราฟ นก เป็นสมาชิกของเซตนี้ ในทางกลับกันกล้วยไม้ไม่เป็นสมาชิกของเซตของสัตว์ เป็นต้น โดยเซตถูกแบ่งออกเป็น 2 ประเภทคือเซตจำกัด (Finite Set) คือเซตที่สามารถนับจำนวนสมาชิกทั้งหมดได้ซึ่งรวมไปถึงเซตว่าง (Empty Set) ซึ่งเป็นเซตที่ไม่มีสมาชิกอยู่เลยโดยทั่วไปสัญลักษณ์ที่ถูกนำมาใช้

แทนเซตว่างคือ “$\varnothing$” เซตประเภทที่สองคือเซตอนันต์ (Infinite Set) คือเซตที่ไม่สามารถนับสมาชิกทั้งหมดได้เนื่องจากมีจำนวนไม่จำกัดเช่น เซตของจำนวนเฉพาะ เซตของจำนวนเต็มบวก เป็นต้น

การเขียนเซตสามารถเขียนได้ 2 รูปแบบคือการเขียนเซตแบบแจกแจงสมาชิกซึ่งเป็นการเขียนสมาชิกของเซตทั้งหมดไว้ภายในเซต และการเขียนเซตแบบบอกเงื่อนไขคือการเขียนเซตโดยใช้เงื่อนไขสำหรับอธิบายถึงสมาชิกที่อยู่ในเซตโดยทั้ง 2 รูปแบบจะถูกกล่าวอีกครั้งในหัวข้อที่ 6.2 และหัวข้อที่ 6.3

เนื่องจากทฤษฎีเซตเป็นหัวข้อทางคณิตศาสตร์ที่มีเนื้อหาเป็นจำนวนมาก ดังนั้นในตำราเล่มนี้จึงกล่าวเพียงหัวข้อพื้นฐานที่สำคัญของเซตที่จำเป็นต้องนำมาใช้ร่วมกับวิทยาการรหัสลับเท่านั้น

6.1 สัญลักษณ์สำหรับเซต

โดยทั่วไปการเขียนเซตจะนิยมใช้ตัวอักษรภาษาอังกฤษตัวพิมพ์ใหญ่เพื่อแทนชื่อของเซต และใช้ตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กแทนตัวแปรที่เป็นสมาชิกของเซต นอกเหนือจากนั้นยังมีสัญลักษณ์ทางคณิตศาสตร์ที่สำคัญหลายตัวที่จำเป็นต้องนำมาใช้ในเซต ตารางที่ 1.2 แสดงตัวอย่างสัญลักษณ์ทางคณิตศาสตร์และคำอธิบายของแต่ละสัญลักษณ์ โดยตารางนี้จะแสดงเฉพาะสัญลักษณ์ที่จะถูกนำมาใช้สำหรับอธิบายเนื้อหาที่เกี่ยวกับวิทยาการรหัสลับในตำราเล่มนี้เท่านั้น

ตารางที่ 1.2 ตัวอย่างสัญลักษณ์ทางคณิตศาสตร์ที่ถูกนำมาใช้กับเซต

สัญลักษณ์	ความหมาย	ตัวอย่างการใช้	คำอธิบายตัวอย่าง
$\in$	เป็นสมาชิก	$a \in A$	a เป็นสมาชิกของ A
$\notin$	ไม่เป็นสมาชิก	$a \notin A$	a ไม่เป็นสมาชิกของ A
$\mathbb{Z}$	จำนวนเต็ม	$a \in \mathbb{Z}$	a เป็นสมาชิกของจำนวนเต็ม
$\mathbb{Z}^+$	จำนวนเต็มบวก	$a \in \mathbb{Z}^+$	a เป็นสมาชิกของจำนวนเต็มบวก
$\mathbb{N}$	จำนวนนับ	$a \in \mathbb{N}$	a เป็นสมาชิกของจำนวนนับ
$\mathbb{R}$	จำนวนจริง	$a \in \mathbb{R}$	a เป็นสมาชิกของจำนวนจริง

6.2 การเขียนเซตแบบแจกแจงสมาชิก

การเขียนเซตแบบแจกแจงสมาชิกเป็นวิธีที่ใช้สำหรับเขียนสมาชิกทุกตัวที่อยู่ภายในเซตซึ่งสมาชิกแต่ละตัวจะถูกเขียนไว้ภายในเครื่องหมายปีกกาและแต่ละตัวจะถูกคั่นด้วยเครื่องหมายจุลภาค

ตัวอย่างที่ 1.1 กำหนดให้ $A = \{1, 3, 9\}$ กล่าวได้ว่า A เป็นเซตจำกัดที่มีสมาชิกทั้งหมด 3 ตัวดังนี้

$1 \in A$ (ความหมายคือ 1 เป็นสมาชิกของ A)

$3 \in A$ (ความหมายคือ 3 เป็นสมาชิกของ A)

$9 \in A$ (ความหมายคือ 9 เป็นสมาชิกของ A)

โดยตัวเลขอื่นๆ ทั้งหมดที่นอกเหนือจากตัวเลขทั้ง 3 ตัวข้างต้นไม่เป็นสมาชิกของ A ดังตัวอย่างต่อไปนี้

$11 \notin A$ (ความหมายคือ 11 ไม่เป็นสมาชิกของ A)

สำหรับกรณีที่สมาชิกในเซตจำกัดมีปริมาณมากสามารถเขียนเซตแบบแจกแจงสมาชิกได้โดยเขียนสมาชิกชุดแรกอย่างน้อย 3 ตัวแล้วตามด้วยจุด 3 จุดและตามด้วยสมาชิกตัวสุดท้าย

ตัวอย่างที่ 1.2 กำหนดให้ B คือเซตของจำนวนเต็มคู่ที่มีค่าอยู่ระหว่าง 0 – 100 สามารถเขียนให้อยู่ในรูปแบบของเซตได้ดังนี้

$B = \{0, 2, 4, \cdots, 100\}$

แต่หากเป็นเซตไม่จำกัดการเขียนเซตแบบแจกแจงสมาชิกสามารถดำเนินการได้โดยเขียนสมาชิกชุดแรกอย่างน้อย 3 ตัวแล้วตามด้วยจุด 3 จุด

ตัวอย่างที่ 1.3 กำหนดให้ C คือเซตของจำนวนเต็มบวกสามารถเขียนให้อยู่ในรูปแบบของเซตได้ดังนี้

$C = \{1, 2, 3, \cdots\}$

6.3 การเขียนเซตแบบบอกเงื่อนไข

การเขียนเซตแบบบอกเงื่อนไขคือการกำหนดเงื่อนไขของสมาชิกที่อยู่ภายในเซต โดยที่รูปแบบการเขียนถูกแบ่งออกเป็น 3 ส่วนซึ่งจะถูกเขียนไว้ภายในปีกกามีรูปแบบเป็นดังนี้

ชื่อเซต = {ส่วนที่ 1 ส่วนที่ 2 ส่วนที่ 3}

โดยที่ ชื่อเซต คือชื่อของเซต

ส่วนที่ 1 คือชื่อตัวแปร

ส่วนที่ 2 คือ “|” แทนความหมายว่า “โดยที่”

ส่วนที่ 3 คือ เงื่อนไข

ตัวอย่างที่ 1.4 จงหาสมาชิกทั้งหมดของ $A = \{x \mid x = 10 \text{ and } x = 20\}$

วิธีทำ จากโจทย์ที่กำหนดมีความหมายเป็นดังนี้

"A คือ เซตของ x โดยที่ $x = 1$ และ $x = 2$"

ดังนั้น 1 และ 2 คือสมาชิกทั้งหมดของ A หรือสามารถเขียน A ในรูปแบบแจกแจงสมาชิกได้เป็นดังนี้ $A = \{1, 2\}$

ตัวอย่างที่ 1.5 จงหาสมาชิกทั้งหมดของ $B = \{y \in \mathbb{Z}^+ \mid 1 < y < 5\}$

วิธีทำ จากโจทย์ที่กำหนดมีความหมายเป็นดังนี้

"B คือ เซตของ y ที่เป็นจำนวนเต็มบวกโดยที่ y ต้องมีค่ามากกว่า 1 และมีค่าน้อยกว่า 5"

ดังนั้น 2, 3 และ 4 คือสมาชิกทั้งหมดของ B หรือสามารถเขียน B ในรูปแบบแจกแจงสมาชิกได้เป็นดังนี้ $B = \{2, 3, 4\}$

ตัวอย่างที่ 1.6 จงหาสมาชิกทั้งหมดของ $C = \{z \in \mathbb{Z}^+ \mid z + 2 = 7\}$

วิธีทำ จากโจทย์ที่กำหนดมีความหมายเป็นดังนี้

"C คือ เซตของ z ที่เป็นจำนวนเต็มบวกโดยที่ผลบวกระหว่าง z และ 2 ต้องมีค่าเป็น 7"

เนื่องจากผลเฉลยของสมการ $z + 2 = 7$ เมื่อ $z \in \mathbb{Z}^+$ มีเพียงผลเฉลยเดียวคือ $z = 5$ ดังนั้นสามารถเขียน C ในรูปแบบแจกแจงสมาชิกได้เป็นดังนี้ $C = \{5\}$

7. ระบบเลขฐานและการแปลงเลขฐาน

ขั้นตอนวิธีทางวิทยาการรหัสลับบางประเภทจำเป็นต้องมีกระบวนการแปลงเลขฐานเช่น วิทยาการรหัสลับดีอีเอส และวิทยาการรหัสลับเออีเอส ดังนั้นในหัวข้อนี้จะกล่าวถึงความหมายของระบบเลขฐาน และวิธีการแปลงเลขฐานที่จำเป็นต้องนำมาใช้งาน

7.1 ระบบเลขฐาน

ระบบเลขฐานคือ ระบบที่ใช้สำหรับการแสดงถึงจำนวนของสิ่งต่างๆ ซึ่งในแต่ละเลขฐานจะมีจำนวนสมาชิกที่แตกต่างกัน โดยจำนวนสมาชิกจะมีค่าตามชื่อของเลขฐาน

กำหนดให้ $n \in \mathbb{Z}^+$ ได้ว่าสมาชิกของเลขฐาน n จะประกอบด้วย 0, 1, 2, ⋯, $n - 1$ เช่น

เลขฐานสอง จะมีสมาชิกทั้งหมด 2 ตัวประกอบด้วย 0, 1

เลขฐานสิบ จะมีสมาชิกทั้งหมด 10 ตัวประกอบด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8 และ 9

เลขฐานสิบหก จะมีสมาชิกทั้งหมด 16 ตัวประกอบด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E และ F, เมื่อ A – B แทนค่า 10 – 15 ตามลำดับ

สำหรับเลขฐานใดๆ ที่ไม่ใช่เลขฐานสิบจะนิยมห้อยค่าเลขฐานดังกล่าวไว้ที่ส่วนท้ายของตัวเลขด้วยเพื่อช่วยให้ทราบว่าตัวเลขดังกล่าวเป็นเลขที่อยู่ในระบบฐานใด เช่น เลขฐานสองที่มีค่าเป็น 1101 สามารถถูกเขียนได้เป็น 1101_2

โดยการแปลงเลขฐาน ซึ่งหมายถึงการแปลงค่าตัวเลขที่อยู่ในระบบเลขฐานหนึ่งไปอยู่ในรูปแบบของเลขฐานอื่นที่ยังคงมีค่าคงเดิม เป็นวิธีที่จะช่วยให้สามารถหาค่าของตัวเลขที่อยู่ในระบบเลขฐานอื่นได้อย่างถูกต้องซึ่งการแปลงเลขฐานแบบปกติถูกแบ่งออกเป็น 2 วิธีคือการแปลงเลขฐานสิบเป็นเลขฐานใดๆ (ที่ไม่ใช่เลขฐานสิบ) และการแปลงจากเลขฐานใดๆ เป็นเลขฐานสิบ

7.2 การแปลงเลขฐานใดๆ เป็นเลขฐานสิบ

การแปลงเลขฐานใดๆ เป็นเลขฐานสิบคือวิธีการแปลงเลขฐานจากเลขฐานที่ไม่ใช่เลขฐานสิบให้ได้ผลลัพธ์เป็นเลขฐานสิบ ซึ่งสามารถดำเนินการได้โดยการนำเอาเลขแต่ละตำแหน่งของตัวเลขที่ต้องการแปลงเป็นเลขฐานสิบมาคูณด้วยค่าน้ำหนักของเลขฐานนั้น และนำผลลัพธ์ทั้งหมดมารวมกัน ผลลัพธ์ที่ได้คือคำตอบที่อยู่ในรูปของเลขฐานสิบ โดยหากแสดงให้อยู่ในรูปของสมการจะสามารถคำนวณได้จาก $\sum_{i=0}^{j} d_i r^i$ เมื่อ r คือระบบเลขฐานของตัวเลขที่ต้องการแปลง d_i คือตัวเลขในตำแหน่งต่างๆ i คือตำแหน่งของตัวเลขแต่ละตัวที่ต้องการแปลงเป็นเลขฐานสิบซึ่งตำแหน่งทางขวาสุดมีค่าเป็น 0 และเรียงเพิ่มขึ้นไปทางซ้ายสุดคือ j และ $\sum_{i=a}^{b}$ แทนผลบวกของแต่ละพจน์ระหว่างพจน์ที่ a ถึง b

ตัวอย่างที่ 1.7 จงแปลง 1010_2 เป็นเลขฐานสิบ
วิธีทำ เนื่องจากค่าตัวเลขและตัวคูณในแต่ละตำแหน่งเป็นดังนี้

ตัวคูณ (ค่าน้ำหนัก)	2^3	2^2	2^1	2^0
เลขประจำตำแหน่ง	1	0	1	0

ดังนั้น $1010_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

$= 8 + 0 + 2 + 0 = 10$

7.3 การแปลงเลขฐานสิบเป็นเลขฐานใดๆ

การแปลงเลขฐานสิบเป็นเลขฐานใดๆ คือวิธีการแปลงเลขฐานจากเลขฐานสิบให้ได้ผลลัพธ์เป็นเลขฐานอื่นที่ต้องการ ซึ่งสามารถดำเนินการได้โดยนำตัวเลขตั้งต้นซึ่งเป็นเลขฐานสิบมาหารด้วย

เลขฐานที่ต้องการแปลง และนำผลลัพธ์ที่ได้มาหารต่อด้วยตัวหารเดิมโดยจะดำเนินการซ้ำลักษณะเดิมจนกระทั่งผลลัพธ์ที่ได้มีค่าน้อยกว่าค่าเลขฐานที่ต้องการแปลง คำตอบที่ต้องการคือผลลัพธ์สุดท้ายและเศษที่ได้จากการหารในแต่ละรอบ โดยผลลัพธ์ที่ได้ตัวสุดท้ายจะมีนัยสำคัญสูงสุงและเรียงย้อนกลับไปยังเศษที่ได้จากการหารครั้งแรกซึ่งมีนัยสำคัญต่ำที่สุด

ตัวอย่างที่ 1.8 จงแปลง 10 เป็นเลขฐานสอง

วิธีทำ

$10 \div 2 = 5$ เศษ 0 ⟶ นัยสำคัญต่ำที่สุด

$5 \div 2 = 2$ เศษ 1

$2 \div 2 = 1$ เศษ 0

↓

นัยสำคัญสูงที่สุด

ดังนั้น $10 = 1010_2$

7.4 การแปลงระหว่างเลขฐานสองและเลขฐานสิบหก

การแปลงเลขฐานระหว่างเลขฐานใดๆ หมายถึงการแปลงค่าระหว่างเลขฐานโดยที่เลขฐานตั้งต้นและเลขฐานที่เป็นผลลัพธ์ไม่ใช่เลขฐานสิบ ซึ่งจำเป็นต้องดำเนินการ 2 ขั้นตอนคือต้องแปลงจากเลขฐานตั้งต้นเป็นเลขฐานสิบ หลังจากนั้นจึงแปลงเลขฐานสิบเป็นเลขฐานที่ต้องการ ยกตัวอย่างเช่น หากต้องการแปลงจากเลขฐานสองเป็นเลขฐานสิบหก การดำเนินการเริ่มจากการแปลงจากเลขฐานสองเป็นเลขฐานสิบก่อน แล้วจึงดำเนินการแปลงจากผลลัพธ์ที่เป็นเลขฐานสิบเป็นผลลัพธ์ที่อยู่ในรูปของเลขฐานสิบหก ซึ่งมีความยุ่งยากเนื่องจากจำเป็นต้องดำเนินการ 2 ขั้นตอน อย่างไรก็ตาม สำหรับกรณีการแปลงเลขฐานระหว่างเลขฐานสองและเลขฐานสิบหกสามารถดำเนินการได้โดยใช้อีกวิธีหนึ่งที่มีความเรียบง่ายกว่าเป็นอย่างมากโดยใช้ข้อมูลดังตารางที่ 1.3

จากตารางที่ 1.3 สังเกตได้ว่าตัวเลขฐานสิบหกจำนวน 1 ตัวจะถูกแทนค่าด้วยเลขฐานสองจำนวน 4 ตัว สำหรับการแปลงเลขฐานระหว่างเลขฐานสองและเลขฐานสิบหกโดยใช้ตารางแบ่งออกเป็น 2 กรณี

กรณีที่ 1 คือการแปลงจากเลขฐานสิบหกเป็นเลขฐานสอง วิธีการคือการแทนที่เลขฐานสิบหกแต่ละตัวด้วยเลขฐานสองที่มีค่าตรงกันดังตารางที่ 1.3

กรณีที่ 2 คือการแปลงจากเลขฐานสองเป็นเลขฐานสิบหก วิธีการคือแบ่งเลขฐานสองออกเป็นกลุ่มๆ ละ 4 บิตโดย เริ่มนับจากตำแหน่งทางขวาสุด (กำหนดให้เป็นกลุ่มที่ 1) เรียงไปจนกระทั่งถึง

ตำแหน่งซ้ายสุดสำหรับกรณีที่กลุ่มสุดท้ายซึ่งเป็นกลุ่มของตำแหน่งที่อยู่ทางซ้ายสุดมีสมาชิกไม่ครบ 4 ค่าให้ทำการเติมเลข 0 ที่ตำแหน่งด้านหน้าสุดเพื่อให้ครบ 4 ค่า หลังจากนั้นให้แทนค่าแต่ละกลุ่มด้วยเลขฐานสิบหกที่มีค่าตรงกันโดยพิจารณาจากตารางที่ 1.3 โดยกลุ่มที่ 1 จะอยู่ตำแหน่งนัยสำคัญต่ำที่สุดเรียงไปจนกระทั่งถึงกลุ่มสุดท้ายซึ่งอยู่ในตำแหน่งนัยสำคัญสูงที่สุด

ตารางที่ 1.3 ตารางความสัมพันธ์ระหว่างเลขฐานสองและเลขฐานสิบหก

เลขฐานสิบหก	เลขฐานสอง
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

ตัวอย่างที่ 1.9 จงแปลง $9A_{16}$ เป็นเลขฐานสอง

วิธีทำ จากตารางที่ 1.3 พบว่า

9 = 1001 และ A = 1010

ดังนั้น $9A_{16} = 10011010_2$

ตัวอย่างที่ 1.10 จงแปลง 10011010_2 เป็นเลขฐานสิบหก

วิธีทำ เริ่มจากแบ่งข้อมูลเป็นกลุ่มๆ ละ 4 ค่าได้ดังนี้

กลุ่มที่ 1: 1010 = A

กลุ่มที่ 2: 1001 = 9

ดังนั้น $10011010_2 = 9A_{16}$

8. การหาเศษที่ได้จากการหาร

การคำนวณหาเศษที่ได้จากการหารเป็นกระบวนการที่สำคัญมากสำหรับวิทยาการรหัสลับ เนื่องจากมีขั้นตอนวิธีสำหรับวิทยาการรหัสลับจำนวนมากทั้งวิทยาการรหัสลับแบบสมมาตร และวิทยาการรหัสลับแบบอสมมาตรที่จำเป็นต้องคำนวณหาเศษที่ได้จากการหาร ยกตัวอย่างเช่นรหัสลับซีซาร์ ซึ่งเป็นวิทยาการรหัสลับแบบสมมาตร และวิทยาการรหัสลับแบบอาร์เอสเอ ซึ่งเป็นวิทยาการรหัสลับแบบอสมมาตร เป็นต้น

กำหนดให้ $a, b \in \mathbb{Z}$ และ c คือ เศษที่เกิดจากการหาร a ด้วย b ได้สมการเป็นดังนี้

$$c = a \bmod b \tag{1.1}$$

ตัวอย่างที่ 1.11 จงคำนวณหาเศษที่เกิดจากการหาร 32 ด้วย 7

วิธีทำ จากโจทย์ $a = 32$ และ $b = 7$ ดังนั้นจากสมการ (1.1) ได้ว่า

$$\begin{aligned} c &= a \bmod b \\ &= 32 \bmod 7 \\ &= 4 \end{aligned}$$

ดังนั้น 4 คือเศษที่เกิดจากการหาร 32 ด้วย 7

อย่างไรก็ตามจากตัวอย่างที่ 1.11 นี้ สมมติ $a = -32$ ได้ว่า $c = -4$ ซึ่งสามารถปรับผลลัพธ์ให้มีค่าเป็นบวกได้โดยนำตัวหารมาบวกกับผลลัพธ์ที่เป็นลบ ซึ่งจากตัวอย่างดังกล่าวนี้จะได้ว่า $c = 7 + (-4) = 3$

บทนิยามที่ 1.1 กำหนดให้ $a, b \in \mathbb{Z}$ หาก a หาร b ลงตัวกล่าวได้ว่ามี $c \in \mathbb{Z}$ ที่ทำให้ $b = ac$ โดยจะใช้สัญลักษ์ $a|b$ แทนความหมายว่า a หาร b ลงตัว ในทางกลับกัน หาก a หาร b ไม่ลงตัว กล่าวได้ว่ามีจำนวนเต็ม $c, r \in \mathbb{Z}$ ที่ทำให้ $b = ac + r$ เมื่อ r คือเศษที่ได้จากการหารและมีค่าในช่วง $1 \leq r < a$ โดยใช้สัญลักษณ์ $a \nmid b$ แทนความหมายว่า a หาร b ไม่ลงตัว

ตัวอย่างที่ 1.12

1) $3 \mid 12$ เนื่องจาก $12 \div 3 = 4$ หรือ $12 = 3 \times 4$
2) $3 \nmid 11$ เนื่องจาก $11 \div 3 = 3$ เศษ 2 หรือ $11 = 3 \times 3 + 2$

9. สมภาค (Congruence)

สมภาคคือหลักการหารจำนวนเต็มหลายจำนวนด้วยตัวหารตัวเดียวกันและได้เศษที่เท่ากัน โดยที่หากเศษที่ได้จากการหารของจำนวนเต็มทั้งหมดมีค่าเท่ากันกล่าวได้ว่าจำนวนเต็มเหล่านั้นเป็นสมภาคต่อกัน

กำหนดให้ $a, b \in \mathbb{Z}$ และ $m \in \mathbb{Z}^+$ หากเศษที่ได้จากการหาร a ด้วย m มีค่าเท่ากับเศษที่ได้จากการหาร b ด้วย m กล่าวได้ว่า a สมภาคกับ b มอดูโล m หรือเขียนให้อยู่ในรูปของสมการได้ดังนี้

$$a \equiv b \pmod{m} \tag{1.2}$$

เนื่องจากเศษจากการหาร a ด้วย m มีค่าเท่ากับเศษที่ได้จากการหาร b ด้วย m ดังนั้นได้ว่า

$$r_1 = a \bmod m$$
$$= a - km$$

และ $$r_2 = b \bmod m$$
$$= b - lm$$

เนื่องจาก $r_1 = r_2$, ดังนั้น $$0 = a - km - (b - lm)$$
$$= (a - b) - (k - l)m$$
$$a - b = (k - l)m$$

หรือ $$m \mid (a - b)$$

จากสมการที่ (1.2) กล่าวอีกนัยหนึ่งได้ว่า b เป็นส่วนตกค้าง (Residue) ของ a มอดูโล m หรือ a เป็นส่วนตกค้างของ b มอดูโล m นอกเหนือจากนั้นเรียก r_1 และ r_2 ว่าส่วนตกค้างน้อยที่สุด เมื่อ r_1 คือเศษที่ได้จากการหาร a ด้วย m และ r_2 คือเศษที่ได้จากการหาร b ด้วย m

ตัวอย่างที่ 1.13 จงแสดงให้เห็นว่า 29 สมภาคกับ 71 มอดูโล 6

วิธีทำ เนื่องจาก

$$29 \bmod 6 = 5$$

$$71 \bmod 6 = 5$$

ดังนั้นสรุปได้ว่า $71 \equiv 29 \pmod 6$

อย่างไรก็ตามสมภาคมีทฤษฎีที่สำคัญ ดังนี้

ทฤษฎีบทที่ 1.2 กำหนดให้ $a \equiv b \bmod m$ และ $c \equiv d \bmod m$ แล้ว

1. $a + c \equiv b + d \pmod m$
2. $a - c \equiv b - d \pmod m$
3. $ac \equiv bd \pmod m$

พิสูจน์ข้อ 1.

เนื่องจาก $m \mid (a - b)$ และ $m \mid (c - d)$

ได้ว่า $m \mid ((a - b) + (c - d))$

จัดรูปใหม่ได้ $m \mid ((a + c) - (b + d))$

แสดงว่า $m \mid (a + c)$ และ $m \mid (b + d)$

ดังนั้น $a + c \equiv b + d \pmod m$

พิสูจน์ข้อ 2.

เนื่องจาก $m \mid (a - b)$ และ $m \mid (c - d)$

ได้ว่า $m \mid ((a - b) - (c - d))$

จัดรูปใหม่ได้ $m \mid ((a - c) - (b - d))$

แสดงว่า $m \mid (a - c)$ และ $m \mid (b - d)$

ดังนั้น $a - c \equiv b - d \pmod m$

พิสูจน์ข้อ 3.

เนื่องจาก $m \mid (a - b)$ และ $m \mid (c - d)$

ได้ว่า $m \mid (a - b)c$ และ $m \mid (c - d)b$

หรือ $m \mid ((a - b)c + (c - d)b)$

จัดรูปใหม่ได้ $m \mid (ac - bd)$

แสดงว่า $m \mid ac$ และ $m \mid bd$

ดังนั้น $ac \equiv bd \pmod{m}$ □

ตัวอย่างที่ 1.14 เนื่องจาก $17 \equiv 61 \pmod{11}$ และ $24 \equiv 46 \pmod{11}$ ดังนั้น

1. $17 + 24 \equiv 61 + 46 \pmod{11}$

   $41 \equiv 107 \pmod{11}$

   $8 \equiv 8 \pmod{11}$

2. $17 - 24 \equiv 61 - 46 \pmod{11}$

   $-7 \equiv 15 \pmod{11}$

   $4 \equiv 4 \pmod{11}$

3. $17 \times 24 \equiv 61 \times 46 \pmod{11}$

   $408 \equiv 2806 \pmod{11}$

   $1 \equiv 1 \pmod{11}$

10. ฟิลด์จำกัด (Finite Field)

ฟิลล์จำกัดคือ ขอบเขตของสมาชิกที่มีจำนวนจำกัด โดยหากนำสมาชิกในกลุ่มมาผ่านการดำเนินการทางคณิตศาสตร์ ผลลัพธ์ที่เกิดจากการคำนวณจะตกอยู่ภายในช่วงที่ไม่เกินจำนวนเต็มใดๆ ค่าหนึ่ง กำหนดให้ GF(z) คือฟิลด์จำกัดซึ่งมีสมาชิกประกอบไปด้วย {0, 1, 2, ⋯, z-1} และหากผลลัพธ์ที่ได้จากการคำนวณมีค่าเกิน z จะหักเฉพาะส่วนที่เกินมาเริ่มทำการนับใหม่ และดำเนินการลักษณะเดิมจนกระทั่งผลลัพธ์มีค่าน้อยกว่า z ซึ่งจากหลักการดังกล่าวเปรียบเสมือนการหาเศษที่ได้จากการหาร โดยที่จากสมการ (1.1) ผลลัพธ์ที่เกินฟิลด์จำกัด (GF(b)) ถูกแทนด้วย a และ c คือผลลัพธ์ใหม่ที่เกิดจากการปรับ a ให้ตกอยู่ภายในขอบเขตของฟิลด์จำกัด (GF(b))

ตัวอย่างที่ 1.15 สมมติผลลัพธ์ที่เกิดจากการคำนวณมีค่าเป็น 18 จงคำนวณหาว่าค่าผลลัพธ์ดังกล่าวนี้จะมีค่าเป็นเท่าใด ในกรณีที่ถูกพิจารณาภายใน GF(8)

วิธีทำ จากโจทย์ a = 18 และ b = 8 ดังนั้นจากสมการที่ (1.1)

$$c = a \bmod b$$
$$= 18 \bmod 8$$
$$= 2$$

ดังนั้น 18 มีค่าเท่ากับ 2 เหนือฟิลด์ GF(8)

กำหนดให้ a, b เป็นจำนวนเต็มใดๆ ที่อยู่เหนือฟิลด์ GF(z) ทฤษฎีการบวก และการคูณเหนือฟิลด์ GF(z) เป็นดังนี้

ทฤษฎีบทที่ 1.3 $(a + b) \bmod z = ((a \bmod z) + (b \bmod z)) \bmod z$

พิสูจน์

กำหนดให้ $r_1 = a \bmod z$ และ $r_2 = b \bmod z$ จึงสามารถเขียนอยู่ในรูป

$$a = kz + r_1$$

$$b = lz + r_2$$

ดังนั้น $$a + b = kz + r_1 + lz + r_2$$

$$= (l + k)z + (r_1 + r_2)$$

ได้ว่า $$(a + b) \bmod z = (r_1 + r_2) \bmod z$$

$$= ((a \bmod z) + (b \bmod z)) \bmod z$$ □

ทฤษฎีบทที่ 1.4 $ab \bmod z = ((a \bmod z)(b \bmod z)) \bmod z$

พิสูจน์

กำหนดให้ $r_1 = a \bmod z$ และ $r_2 = b \bmod z$ จึงสามารถเขียนอยู่ในรูป

$$a = kz + r_1$$

$$b = lz + r_2$$

ดังนั้น $$ab = (kz + r_1)(lz + r_2)$$

$$= (lkz + r_1l + r_2k)z + (r_1r_2)$$

ได้ว่า $$ab \bmod z = r_1r_2 \bmod z$$

$$= ((a \bmod z)(b \bmod z)) \bmod z$$ □

ตัวอย่างที่ 1.16 จงคำนวณหาผลลัพธ์ของ (28 + 76) mod 17

วิธีทำ เนื่องจาก 28 mod 17 = 11 และ 76 mod 17 = 8

ดังนั้น $$(28 + 76) \bmod 17 = ((28 \bmod 17) + (76 \bmod 17)) \bmod 17$$

$$= (11 + 8) \bmod 17$$

$$= 19 \bmod 17$$

$$= 2$$

ตัวอย่างที่ 1.17 จงคำนวณหาผลลัพธ์ของ 9 x 7 mod 5

วิธีทำ เนื่องจาก 9 mod 5 = 4 และ 7 mod 5 = 2

ดังนั้น $9 \times 7 \bmod 5 = ((9 \bmod 5)(7 \bmod 5)) \bmod 5$

$= (4 \times 2) \bmod 5$

$= 8 \bmod 5$

$= 3$

จากตัวอย่างที่ 1.16 และ 1.17 แสดงให้เห็นว่ากรณีที่จำนวนเต็มใดๆ ที่ไม่ได้เป็นสมาชิกเหนือฟิลด์จำกัดสามารถถูกนำมาปรับให้อยู่เหนือฟิลด์จำกัดก่อนจะนำมาดำเนินการบวก หรือคูณได้

11. หารร่วมมาก (Greatest Common Divisor)

หารร่วมมาก คือการคำนวณหาตัวหารร่วมที่มีค่ามากที่สุดระหว่างจำนวนเต็มตั้งแต่สองค่าขึ้นไป ยกตัวอย่างเช่นค่าหารร่วมมาก ของ 6 และ 12 คือ 6 เนื่องจาก 6 คือจำนวนเต็มที่มีค่ามากที่สุดที่สามารถหาร 6 และ 12 ลงตัว ถึงแม้ว่า 3 จะสามารถหารทั้ง 6 และ 12 ลงตัว แต่ 3 ไม่เป็นค่าหารร่วมมากของ 6 และ 12 เนื่องจาก 3 ไม่ใช่ค่าสูงสุดที่หารจำนวนเต็มทั้งสองค่าลงตัว

กำหนดให้ gcd(a, b) แทนการหาค่าหารร่วมมากระหว่าง a และ b ตัวอย่างที่ 1.18 แสดงวิธีการคำนวณ gcd(18, 42)

ตัวอย่างที่ 1.18 จงคำนวณหา gcd(18, 42)

วิธีทำ ตัวประกอบของ 18 คือ 1, 2, 3, **6**, 9, 18

ตัวประกอบของ 42 คือ 1, 2, 3, **6**, 7, 12, 21, 42

เนื่องจาก 6 คือตัวประกอบสูงสุดของ 18 และ 42 ดังนั้นจึงสรุปได้ว่า 6 คือ ค่าหารร่วมมากของ 18 และ 42

11.1 ขั้นตอนวิธียุคลิด (Euclidean Algorithm)

ขั้นตอนวิธียุคลิด คือวิธีการหาค่าหารร่วมมากระหว่างจำนวนเต็มสองจำนวนวิธีหนึ่งที่สามารถคำนวณหาค่าตัวประกอบร่วมสูงสุดได้อย่างรวดเร็ว หากเปรียบเทียบกับวิธีการหาค่าหารร่วมมากแบบปกติ โดยเฉพาะอย่างยิ่งหากจำนวนเต็มที่จะถูกนำมาคำนวณหาค่าหารร่วมมากมีขนาดใหญ่ กำหนดให้ $a, b \in \mathbb{Z}$ โดยที่ $a \geq b$ การหาค่าหารร่วมมากโดยขั้นตอนวิธีแบบยุคลิดถูกแบ่งออกเป็น 2 เงื่อนไข ดังนี้

1. กรณีที่ $b = 0$, gcd(a, b) = $|a|$
2. กรณีที่ $b \neq 0$, gcd(a, b) = gcd($|b|$, $a \bmod |b|$)

เมื่อสัญลักษณ์ “$|a|$” แทนความหมายของค่าสัมบูรณ์ (Absolute value) ของ a ที่มีผลลัพธ์เป็นดังนี้

$$|a| = a \text{ เมื่อ } a \geq 0 \text{ และ } |a| = -a \text{ เมื่อ } a < 0$$

จากทั้งสองเงื่อนไขข้างต้น หาก b ไม่เป็น 0 จะดำเนินการซ้ำเดิมในกรณีที่ 2 จนกระทั่งพบ b ที่มีค่าเป็น 0 จึงหยุดการดำเนินการ ดังนั้นได้ขั้นตอนวิธีแบบยุคลิดสำหรับคำนวณหาค่าหารร่วมมากระหว่างจำนวนเต็มทั้งสองค่าเป็นดังนี้

ขั้นตอนวิธีที่ 1.1 ขั้นตอนวิธียุคลิด

```
     INPUT: a, b
     OUTPUT: a
1:   a ← |a|
2:   b ← |b|
3:
     While(b ≠ 0) do
4:       r ← a mod b
5:       a ← b
6:       b ← r
7:   End While
```

ตัวอย่างที่ 1.19 จงคำนวณหา gcd(117, 52) ด้วยขั้นตอนวิธียุคลิด

วิธีทำ จากโจทย์ได้ $a = 117$, $b = 52$ จึงสามารถใช้ขั้นตอนวิธี 1.1 สำหรับคำนวณหา gcd(117, 52) ดังนี้

1. $a = |117| = 117$
2. $b = |52| = 52$

ขั้นตอนที่ 3 – 7 เป็นการดำเนินการภายในวงวน

เนื่องจาก $b \neq 0$ ดังนั้น

รอบที่ 1

4. $r = 117 \bmod 52 = 13$
5. $a = b = 52$
6. $b = r = 13$

รอบที่ 2

4. $r = 52 \bmod 13 = 0$

5. $a = b = 13$

6. $b = r = 0$

เนื่องจาก $b = 0$ ดังนั้นสรุปได้ว่า gcd(117, 52) คือ 13

อย่างไรก็ตามการคำนวณหาค่าหารร่วมมากด้วยขั้นตอนวิธีแบบยุคลิดดังตัวอย่างที่ 1.19 สามารถเก็บเพียงผลลัพธ์ และเศษที่ได้จากการหารในรอบสุดท้ายเท่านั้นซึ่งจะไม่สามารถเก็บผลลัพธ์และเศษที่คำนวณได้ในแต่ละรอบการทำงาน ดังนั้นการดำเนินการด้วยขั้นตอนวิธีแบบยุคลิดสามารถปรับเปลี่ยนเป็นอีกรูปแบบที่สามารถเก็บผลลัพธ์ และเศษที่ได้จากการหารในแต่ละรอบการคำนวณได้ดังต่อไปนี้

ขั้นตอนวิธีที่ 1.2 ขั้นตอนวิธียุคลิด (วิธีที่สอง)

```
     INPUT: a, b เมื่อ a > b
     OUTPUT: r_{k-1}
1:   r_0 ← |a|
2:   r_1 ← |b|
3:   k ← 1
4:   While(r_k ≠ 0) do
5:       q_k ← ⌊ r_{k-1} / r_k ⌋
6:       r_{k-1} ← q_k r_k + r_{k+1}
7:       k ← k + 1
8:   End While
```

จากขั้นตอนวิธีที่ 1.2 สัญลักษณ์ “$\lfloor a \rfloor$” แทนความหมายของฟังก์ชันพื้น (Floor Function) ของ a ที่มีความหมายคือจำนวนเต็มที่มากที่สุดที่มีค่าน้อยกว่าหรือเท่ากับ a ตัวอย่างเช่น $\lfloor 8.7 \rfloor = 8$

นอกเหนือจากนั้นยังมีอีกสัญลักษณ์หนึ่งที่มีความหมายตรงข้ามกับฟังก์ชันพื้นเรียกว่าฟังก์ชันเพดาน (Ceiling Function) ที่ใช้สัญลักษณ์เป็น “$\lceil a \rceil$” ที่มีความหมายคือจำนวนเต็มที่น้อยที่สุดที่มีค่ามากกว่าหรือเท่ากับ a ตัวอย่างเช่น $\lceil 8.7 \rceil = 9$

ตัวอย่างที่ 1.20 จงคำนวณหา gcd(117, 52) ด้วยขั้นตอนวิธียุคลิดวิธีที่สอง

วิธีทำ เนื่องจาก $a = 117$ และ $b = 52$ จึงสามารถใช้ขั้นตอนวิธี 1.2 สำหรับคำนวณหา gcd(117, 52) ดังนี้

1. $a = |117| = 117$

2. $b = |52| = 52$

3. $k = 1$

ขั้นตอนที่ 4 – 8 เป็นการดำเนินการภายในวงวน

รอบที่ 1 ($r_1 \neq 0$)

5. $q_1 = \left\lfloor \frac{r_0}{r_1} \right\rfloor$

$= \left\lfloor \frac{117}{52} \right\rfloor$

$= 2$

6. $r_0 = q_1 r_1 + r_2$

$117 = 2 \times 52 + r_2$

$r_2 = 13 \neq 0$

รอบที่ 2 ($r_2 \neq 0$)

5. $q_2 = \left\lfloor \frac{r_1}{r_2} \right\rfloor$

$= \left\lfloor \frac{52}{13} \right\rfloor$

$= 4$

6. $r_1 = q_2 r_2 + r_3$

$52 = 4 \times 13 + r_3$

$\mathbf{r_3 = 0}$

เนื่องจาก $r_3 = 0$ จึงสรุปได้ว่า gcd(117, 52) คือ r_2 ซึ่งมีค่าเป็น 13

11.2 ขั้นตอนวิธียุคลิดภาคขยาย (Extended Euclidean Algorithm)

ขั้นตอนวิธียุคลิดภาคขยาย [43] เป็นขั้นตอนวิธีที่ปรับปรุงจากขั้นตอนวิธียุคลิดที่สามารถนำมาใช้สำหรับแก้ปัญหาด้านอื่นๆ นอกเหนือจากการคำนวณหาค่าหารร่วมมากระหว่างจำนวนเต็มสองจำนวนได้ประกอบไปด้วยดังนี้ การนำขั้นตอนวิธียุคลิดภาคขยายมาแก้ปัญหาสมการเชิงเส้น และการนำขั้นตอนวิธียุคลิดภาคขยายมาใช้เพื่อหาค่าผกผันเหนือฟิลด์จำกัด สำหรับในหัวข้อนี้จะกล่าวถึงการนำขั้นตอนวิธียุคลิดภาคขยายมาแก้ปัญหาสมการเชิงเส้น

จากการคำนวณหา gcd(a, b) ด้วยขั้นตอนวิธียุคลิด แล้วมีจำนวนเต็มสองจำนวนกำหนดเป็น x และ y ที่ทำให้

$$ax + by = \gcd(a, b) \tag{1.3}$$

เรียกสมการที่ (1.3) ว่าสมการเชิงเส้น ซึ่งหมายถึงผลรวมของผลคูณระหว่างค่าคงที่และตัวแปรที่มีดีกรี (ผลรวมของเลขชี้กำลังของตัวแปรแต่ละพจน์) เป็น 0 หรือ 1 ซึ่งจากสมการที่ (1.3) เป็นสมการเชิงเส้นแบบ 2 ตัวแปรคือ x และ y

จากสมการ (1.3) สามารถใช้ขั้นตอนวิธียุคลิดภาคขยายเพื่อแก้ปัญหาสมการได้ดังนี้

กำหนดให้ $x_0 = 1$

$x_1 = 0$

$y_0 = 0$

$y_1 = 1$

$x_{k+1} = q_k x_k + x_{k-1}$

$y_{k+1} = q_k y_k + y_{k-1}$

โดยสามารถคำนวณหา x และ y ได้จากสมการต่อไปนี้

$$x = (-1)^l x_l \tag{1.4}$$

$$y = (-1)^{l+1} y_l \tag{1.5}$$

เมื่อ l คือตำแหน่งสุดท้ายที่ทำให้ $r_l \neq 0$

ตัวอย่างที่ 1.21 จงคำนวณหาค่า x และ y จาก $117x + 52y = 13$ ด้วยขั้นตอนวิธียุคลิดภาคขยาย

วิธีทำ จากสมการ (1.3) ได้ว่า $117x + 52y = \gcd(117, 52)$

จากตัวอย่างที่ 1.20 $\gcd(117, 52) = 13$

ดังนั้น $117x + 52y = 13$ จึงสามารถคำนวณหาค่า x และ y ด้วยขั้นตอนวิธียุคลิดภาคขยายได้ดังนี้

จากค่าเริ่มต้น $r_0, r_1, x_0, x_1, y_0, y_1$ และผลลัพธ์ r_{i+1} และ q_i เมื่อ $i = 1, 2, 3, \cdots, k'$ ที่ได้จากตัวอย่างที่ 1.20 สามารถคำนวณหา x และ y ได้ดังนี้

จากขั้นตอนวิธียุคลิดภาคขยาย (เพิ่มเติมขั้นตอนจากขั้นตอนวิธี 1.2)

$r_0 = 117, x_0 = 1, y_0 = 0, r_1 = 52, q_1 = 2, x_1 = 0, y_1 = 1$

รอบที่ 1

$r_2 = 13, q_2 = 4$

$x_2 = q_1 x_1 + x_0$

$= 2 \times 0 + 1 = 1$

$y_2 = q_1 y_1 + y_0$

$= 2 \times 1 + 0 = 2$

รอบที่ 2

เนื่องจาก $r_3 = 0$ สรุปได้ว่า $l = 2$ จึงสามารถคำนวณหา x และ y จากสมการ (1.4) และ (1.5) ได้ดังนี้

$$x = (-1)^2 x_2$$
$$= (-1)^2(1) = 1$$
$$y = (-1)^3 y_2$$
$$= (-1)^3(2) = -2$$

ทดสอบนำผลลัพธ์ของ x และ y ตรวจสอบความถูกต้อง ดังนี้

จาก $117x + 52y = 13$

แทนค่า $117 \times 1 + 52 \times -2 = 13$

$13 = 13$

11.3 การประยุกต์ขั้นตอนวิธียุคลิดภาคขยายสำหรับหาค่าผกผันเหนือฟิลด์จำกัด

ค่าผกผันเหนือฟิลด์จำกัด มีความหมายคือ หากกำหนดให้จำนวนเต็มสองจำนวนเป็นสมาชิกที่อยู่เหนือฟิลด์ GF(z) และเศษที่ได้จากการหารผลคูณระหว่างจำนวนเต็มทั้งสองค่าด้วย z มีค่าเป็น 1 กล่าวได้ว่าจำนวนเต็มทั้งสองค่าคือค่าผกผันต่อกันเหนือฟิลด์ GF(z)

กำหนดให้ a เป็นจำนวนเต็มที่เป็นสมาชิกเหนือฟิลด์ GF(b) ที่มีค่าผกผันแล้ว gcd(a, b) = 1 โดยเรียก a และ b ว่าเป็นจำนวนเฉพาะสัมพัทธ์ต่อกัน (Relatively Prime) ซึ่งหมายถึงจำนวนเต็มที่ไม่มีตัวประกอบร่วมกัน ยกเว้น 1 และ -1 ดังนั้น

$$ax + by = 1 \qquad (1.6)$$

และสามารถใช้ขั้นตอนวิธียุคลิดภาคขยายสำหรับคำนวณหา b^{-1} เหนือฟิลด์ GF(a) ได้ เนื่องจากสมการที่ (1.6) ได้ว่า $ax \bmod a = 0$ ดังนั้น $y = b^{-1} \bmod a$

ตัวอย่างที่ 1.22 จงคำนวณหา 7^{-1} เหนือฟิลด์ GF(29)

วิธีทำ จากโจทย์สามารถนำมาเขียนในรูปสมการ (1.6) ได้คือ $29x + 7y = 1$

จากขั้นตอนวิธียุคลิดภาคขยาย ($x_0 = 1$, $y_0 = 0$, $x_1 = 0$, $y_1 = 1$)

1. $r_0 = 29$
2. $r_1 = 7$
3. $k = 1$

รอบที่ 1

5. $q_1 = \left\lfloor \frac{r_0}{r_1} \right\rfloor$

$= \left\lfloor \frac{29}{7} \right\rfloor = 4$

6. $r_0 = q_1 r_1 + r_2$

$29 = 4 \times 7 + r_2$

$r_2 = 1 \neq 0$

และสามารถคำนวณหา x_2 และ y_2 ได้ดังนี้

$x_2 = q_1 x_1 + x_0$

$= 4 \times 0 + 1 = 1$

$y_2 = q_1 y_1 + y_0$

$= 4 \times 1 + 0 = 4$

รอบที่ 2

5. $q_2 = \left\lfloor \frac{r_1}{r_2} \right\rfloor$

$= \left\lfloor \frac{7}{1} \right\rfloor = 7$

6. $r_1 = q_2 r_2 + r_3$

$7 = 7 \times 1 + r_2$

$r_3 = 0$

เนื่องจาก $r_3 = 0$ สรุปได้ว่า $l = 2$ จึงสามารถคำนวณหา x และ y จากสมการ (1.4) และ (1.5) ได้ดังนี้

$x = (-1)^2 x_2$

$= (-1)^2(1) = 1$

$y = (-1)^3 y_2$

$= (-1)^3(4) = -4$

และเนื่องจาก -4 อยู่นอกฟิลด์ GF(29) จึงได้ว่า -4 mod 29 = 25

ทดสอบนำผลลัพธ์ของ x และ y ตรวจสอบความถูกต้อง ดังนี้

จาก $29x + 7y = 29 \times 1 + 7 \times 25$

เนื่องจากโจทย์เป็นการดำเนินการเหนือฟิลด์ GF(29)

ดังนั้น $29x + 7y = (29 \times 1 + 7 \times 25) \bmod 29$

$= 7 \times 25 \bmod 29$

$= 175 \bmod 29$

$= 1$

เนื่องจาก $7 \times 25 \bmod 29 = 1$ หมายความว่า $25 \equiv 7^{-1} \pmod{29}$

12. ฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

ฟังก์ชันพหุนาม (Polynomial Function) คือฟังก์ชันที่ถูกเขียนให้อยู่ในรูปของผลรวมของผลคูณระหว่างตัวแปรที่มีดีกรีเป็นจำนวนเต็มบวกหรือศูนย์และค่าคงที่ดังสมการ (1.7) ดังนั้นฟังก์ชันพหุนามเหนือฟิลด์จำกัด (Polynomial Function Over Finite Field) คือฟังก์ชันพหุนามที่ดำเนินการภายในขอบเขตที่จำกัดเป็นฟังก์ชันที่ถูกนำมาใช้สำหรับแก้ปัญหาวิทยาการรหัสลับทั้งแบบสมมาตร เช่นวิทยาการรหัสลับแบบเออีเอสซึ่งเนื้อหาโดยละเอียดอยู่ในบทที่ 4 และวิทยาการรหัสลับแบบอสมมาตร เช่นวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง ซึ่งเนื้อหาโดยละเอียดอยู่ในบทที่ 9

กำหนดให้ $a_i \in \mathbb{Z}$ เมื่อ $i = 0, 1, 2, \cdots, m-1$ และ $F(x)$ คือฟังก์ชันพหุนามที่มีดีกรีสูงสุดคือ m-1 ซึ่งถูกเขียนให้อยู่ในรูปแบบของสมการได้ดังนี้

$$F(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 \qquad (1.7)$$

อย่างไรก็ตามฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) จะมีรูปแบบที่แตกต่างจากฟังก์ชันพหุนามดังนี้ กำหนดให้ GF(2^m) แทนฟิลด์จำกัดของฟังก์ชันพหุนามที่มีดีกรีสูงสุดคือ m-1 ซึ่งถูกเขียนให้อยู่ในรูปแบบของสมการได้ดังนี้

$$f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 \qquad (1.8)$$

เมื่อ $a_i = \{0, 1\}$

สำหรับกรณีที่ผลลัพธ์ที่ได้จากการดำเนินการระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) ที่มีดีกรีสูงเกิน m-1 ซึ่งโดยทั่วไปเกิดจากการดำเนินการคูณระหว่างสองฟังก์ชันจำเป็นต้องปรับผลลัพธ์ดังกล่าวให้กลับมาอยู่ภายในฟิลด์โดยการลดดีกรีสูงสุดให้มีค่าน้อยกว่าหรือเท่ากับ m-1 ซึ่งสามารถดำเนินการได้โดยการใช้เศษที่เกิดจากการหารผลลัพธ์ตั้งต้นด้วยฟังก์ชันพหุนามไม่ลดรูป

(Irreducible Polynomial) ซึ่งเป็นฟังก์ชันพหุนามชนิดหนึ่งที่ไม่สามารถถูกลดรูปได้อีก โดยหลักการหาเศษที่ได้จากการหารระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) ใช้หลักการเดียวกับการหาผลหารของฟังก์ชันพหุนาม โดยในตำราเล่มนี้จะแสดงวิธีการหาเศษ และผลหารของฟังก์ชันพหุนามโดยใช้วิธีการหารยาว

12.1 การดำเนินการบวกระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

การดำเนินการบวกค่าระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) สามารถดำเนินการได้โดยนำสัมประสิทธิ์ (ค่าคงที่ที่เป็นตัวคูณของตัวแปรแต่ละตัว) ของตัวแปรแต่ละฟังก์ชันที่มีดีกรีเท่ากันมาบวกกัน แต่เนื่องจากสัมประสิทธิ์มีค่าเป็นไปได้เพียง 0, 1 ซึ่งผลลัพธ์ที่ได้จะต้องเป็นค่าที่อยู่ในช่วงดังกล่าวนี้เท่านั้น ดังนั้นผลลัพธ์ที่ได้จากการบวกแต่ละพจน์เป็นดังนี้

$$a_ix^j + b_ix^j = ((a_i + b_i) \bmod 2)x^j \tag{1.9}$$

ตัวอย่างที่ 1.23 กำหนดให้ $f(x) = x^3 + x + 1$ และ $g(x) = x^3 + x^2 + x + 1$ โดยที่ $f(x)$, $g(x)$ เป็นฟังก์ชันพหุนามเหนือฟิลด์ GF(2^4) จงคำนวณหา $f(x)+g(x)$

วิธีทำ

$$\begin{aligned} f(x) + g(x) &= (x^3 + x + 1) + (x^3 + x^2 + x + 1) \\ &= (1+1)x^3 + (0 + 1)x^2 + (1+1)x + (1+1) \\ &= 2x^3 + x^2 + 2x + 2 \end{aligned}$$

เนื่องจาก 2 mod 2 = 0, ดังนั้น $f(x) + g(x) = x^2$

12.2 การดำเนินการลบระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

การดำเนินการลบค่าระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) มีลักษณะเช่นเดียวกับการดำเนินการบวก แตกต่างกันเพียงเป็นการลบโดยรูปแบบการลบเป็นดังนี้

$$a_ix^j - b_ix^j = ((a_i - b_i) \bmod 2)x^j \tag{1.10}$$

ตัวอย่างที่ 1.24 จาก $f(x)$ และ $g(x)$ ที่กำหนดในตัวอย่างที่ 1.23 จงคำนวณหา $f(x) - g(x)$

วิธีทำ

$$\begin{aligned} f(x) - g(x) &= (x^3 + x + 1) - (x^3 + x^2 + x + 1) \\ &= (1 - 1)x^3 + (0 - 1)x^2 + (1-1)x + (1-1) \end{aligned}$$

$$= 0x^3 + (-1)x^2 + 0x + 0$$

เนื่องจาก -1 mod 2 = 1 ดังนั้น

$f(x) - g(x) = x^2$

12.3 การดำเนินการคูณระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

การหาผลคูณระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) สามารถทำได้โดยการคูณแต่ละพจน์ของฟังก์ชันหนึ่งกับทุกๆ พจน์ของอีกฟังก์ชันหนึ่ง โดยผลลัพธ์ของแต่ละพจน์คือนำดีกรีของตัวแปรมาบวกกัน (สัมประสิทธิ์โดยทั่วไปของแต่ละพจน์ของทั้งสองฟังก์ชันมีค่าเป็น 1 จึงไม่มีการเปลี่ยนแปลง) ขั้นตอนสุดท้ายนำผลลัพธ์ของแต่ละพจน์มาบวกกัน อย่างไรก็ตามดีกรีสูงสุดของผลลัพธ์ที่ได้จากการดำเนินการคูณอาจเกินขอบเขตของฟิลด์ ดังนั้นจึงจำเป็นต้องปรับผลลัพธ์ดังกล่าวโดยการลดดีกรีสูงสุดให้มีค่าน้อยกว่าหรือเท่ากับ m-1 ด้วยการคำนวณหาเศษที่เกิดจากการหารผลลัพธ์ด้วยฟังก์ชันพหุนามไม่ลดรูป

ตัวอย่างที่ 1.25 จาก $f(x)$ และ $g(x)$ ที่กำหนดในตัวอย่างที่ 1.23 และกำหนดให้ฟังก์ชันพหุนามไม่ลดรูป $h(x) = x^4 + x + 1$ จงคำนวณหา $f(x) \times g(x)$

วิธีทำ

$$\begin{aligned} f(x) \times g(x) &= (x^3 + x + 1)\cdot(x^3 + x^2 + x + 1) \\ &= x^{3+3} + x^{3+1} + x^3 + x^{3+2} + x^{2+1} + x^2 + x^{3+1} + x^{1+1} + x + x^3 + x + 1 \\ &= x^6 + x^4 + x^3 + x^5 + x^3 + x^2 + x^4 + x^2 + x + x^3 + x + 1 \\ &= x^6 + x^5 + x^3 + 1 \end{aligned}$$

เนื่องจากค่าผลลัพธ์ที่ได้อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$\begin{array}{r} x^2 + x \\ x^4 + x + 1 \overline{)\, x^6 + x^5 + x^3 + 1} \\ \underline{x^6 + x^3 + x^2} \\ x^5 - x^2 + 1 \\ \underline{x^5 + x^2 + x} \\ \underline{\underline{-x + 1}} \end{array}$$

อย่างไรก็ตามเนื่องจาก -1 mod 2 = 1 ได้ว่า $-x + 1 = x+1$

เนื่องจาก $(x^6 + x^5 + x^3 + 1) \div (x^4 + x + 1) = x^2 + x$ เศษ $x+1$

ดังนั้น $f(x) \times g(x) = x + 1$

12.4 การประยุกต์ขั้นตอนวิธียุคลิดสำหรับคำนวณหาค่าหารร่วมมากระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

เนื่องจากการนำวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองมาใช้สำหรับกระบวนการเข้ารหัสและถอดรหัสจำเป็นต้องมีการคำนวณหาค่าผกผันของฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) อย่างไรก็ตามก่อนคำนวณหาค่าผกผันจำเป็นต้องทราบผลลัพธ์ของการหารร่วมมากก่อน โดยสามารถเลือกใช้ขั้นตอนวิธียุคลิดทั้งแบบขั้นตอนวิธี 1.1 และ ขั้นตอนวิธี 1.2 หรือขั้นตอนวิธียุคลิดภาคขยายเพื่อคำนวณหาค่าหารร่วมมากได้

ตัวอย่างที่ 1.26 จาก $f(x)$ และ $g(x)$ ที่กำหนดในตัวอย่างที่ 1.23 จงคำนวณหา gcd($f(x)$, $g(x)$)

วิธีทำ ใช้ขั้นตอนวิธี 1.1 โดยมีขั้นตอนเป็นดังนี้

1. $a = f(x) = x^3 + x + 1$
2. $b = g(x) = x^3 + x^2 + x + 1$

ขั้นตอนที่ 3 – 7 เป็นการดำเนินการภายในวงวน

รอบที่ 1

4. $r = (x^3 + x + 1) \bmod (x^3 + x^2 + x + 1)$ โดยใช้วิธีหารยาว ดังนี้

$$
\begin{array}{r}
1 \\
x^3 + x^2 + x + 1 \overline{)\, x^3 + x + 1} \\
\underline{x^3 + x^2 + x + 1} \\
\underline{\underline{- x^2}}
\end{array}
$$

ดังนั้น $r = x^2$

5. $a = b = x^3 + x^2 + x + 1$
6. $b = r = x^2$

รอบที่ 2

4. $r = x^3 + x^2 + x + 1 \bmod x^2$ โดยใช้วิธีหารยาว ดังนี้

$$
\begin{array}{r}
x+1 \\
x^2\overline{)\,x^3+x^2+x+1} \\
\underline{x^3} \\
\quad x^2+x+1 \\
\quad \underline{x^2} \\
\quad\quad x+1 \\
\quad\quad \underline{\underline{x+1}}
\end{array}
$$

ดังนั้น $r(x) = x+1$

5. $a = b = x^2$

6. $b = r = x+1$

รอบที่ 3

4. $r = x^2 \bmod x+1$ โดยใช้วิธีหารยาว ดังนี้

$$
\begin{array}{r}
x-1 \\
x+1\overline{)\,x^2} \\
\underline{x^2+x} \\
\quad -x \\
\quad \underline{-x-1} \\
\quad\quad \underline{\underline{1}}
\end{array}
$$

ดังนั้น $r(x) = 1$

5. $a = b = x+1$

6. $b = r = 1$

รอบที่ 4

4. $r = x + 1 \bmod 1 = 0$

5. $a = b = 1$

6. $b = r = 0$

เนื่องจาก $b = 0$ ดังนั้นสรุปได้ว่า gcd($f(x)$, $g(x)$) คือ 1

12.5 การประยุกต์ขั้นตอนวิธียุคลิดภาคขยายสำหรับคำนวณหาค่าผกผันของฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m)

กำหนดให้ $f(x)$ และ $g(x)$ คือฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) โดยที่ gcd($f(x)$, $g(x)$) = 1 ได้ว่า $f(x)$ มีค่าผกผันเหนือฟิลด์จำกัดดังสมการ (1.11)

$$i(x) = f^{-1}(x) \bmod g(x) \tag{1.11}$$

ดังนั้นจึงสามารถนำขั้นตอนวิธียุคลิดภาคขยายมาประยุกต์สำหรับหาค่าผกผันของฟังก์ชันพหุนามได้

ตัวอย่างที่ 1.27 กำหนด $f(x) = x^2 + 1$ และ $g(x) = x^4 + x + 1$ จงคำนวณหา $f^{-1}(x)$

วิธีทำ จากโจทย์สามารถนำมาเขียนในรูปสมการ (1.6) ได้คือ $ax + by = 1$ (เมื่อ $a = g(x)$ และ $b = f(x)$) และใช้ขั้นตอนวิธีวิยุคลิดภาคขยายเพื่อหา $f^{-1}(x)$ ดังนี้

จากขั้นตอนวิธียุคลิดภาคขยาย ($x_0 = 1, y_0 = 0, x_1 = 0, y_1 = 1$)

1. $r_0 = x^4 + x + 1$
2. $r_1 = x^2 + 1$
3. $k = 1$

รอบที่ 1

5. $q_1 = \left\lfloor \frac{r_0}{r_1} \right\rfloor$

$$\begin{array}{r} x^2+1 \\ x^2+1 \overline{)\, x^4+x+1} \\ \underline{x^4+x^2} \\ x^2+x+1 \\ \underline{x^2+1} \\ x \end{array}$$

ดังนั้น $q_1 = x^2 + 1$

6. $r_2 = x$

และสามารถคำนวณหา x_2 และ y_2 ได้ดังนี้

$x_2 = q_1x_1 + x_0$

$= (x^2 + 1)(0) + 1 = 1$

$y_2 = q_1y_1 + y_0$

$= (x^2 + 1)(1) + 0 = x^2 + 1$

รอบที่ 2

5. $q_2 = \left\lfloor \frac{r_1}{r_2} \right\rfloor$

$$
\begin{array}{r}
x \\
x\overline{)\,x^2+1} \\
\underline{x^2} \\
1
\end{array}
$$

ดังนั้น $q_2 = x$

6. $r_3 = 1$

และสามารถคำนวณหา x_3 และ y_3 ได้ดังนี้

$x_3 = q_2x_2 + x_1$

$= (x)(1) + 0 = x$

$y_3 = q_2y_2 + y_1$

$= (x)(x^2 + 1) + 1 = x^3 + x + 1$

รอบที่ 3

5. $q_3 = \left\lfloor \frac{r_2}{r_3} \right\rfloor = x$

6. $r_4 = 0$

เนื่องจาก $r_4 = 0$ สรุปได้ว่า $l = 3$ จึงสามารถคำนวณหา x และ y จากสมการ (1.4) และ (1.5) ได้ดังนี้

$x = (-1)^3 x_3$

$= (-1)^3(x)$

$= x$

$y = (-1)^4 y_3$

$= (-1)^4(x^3 + x + 1)$

$= x^3 + x + 1$

ทดสอบนำผลลัพธ์ของ x และ y ตรวจสอบความถูกต้อง ดังนี้

จาก $ax + by = (x^4 + x + 1)x + (x^2 + 1)(x^3 + x + 1)$

เนื่องจากโจทย์เป็นการดำเนินการเหนือฟิลด์จำกัด ดังนั้น

$(x^4 + x + 1)x + (x^2 + 1)y = (x^4 + x + 1)x + (x^2 + 1)(x^3 + x + 1) \bmod (x^4 + x + 1)$

$= (x^2 + 1)(x^3 + x + 1) \bmod (x^4 + x + 1)$

$= x^5 + x^2 + x + 1 \bmod x^4 + x + 1$

$= 1$

เนื่องจาก $(x^2 + 1)(x^3 + x + 1) \bmod (x^4 + x + 1) = 1$ หมายความว่า

$$x^3 + x + 1 \equiv (x^2 + 1)^{-1} \bmod x^4 + x + 1$$

ดังนั้น $f^{-1}(x) = x^3 + x + 1$

12.6 การแปลงฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) ในรูปแบบเลขฐานสองหรือเลขฐานสิบหก

ฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) สามารถถูกนำมาเขียนให้อยู่ในรูปแบบของเลขฐานสองได้โดยใช้สัมประสิทธิของแต่ละพจน์ที่ถูกเขียนเรียงจากพจน์ที่มีเลขยกกำลังสูงสุดไปจนถึงพจน์ที่มีเลขยกกำลังต่ำที่สุดซึ่งต้องรวมถึงพจน์ที่มีสัมประสิทธิที่มีค่าเป็น 0 ไว้ด้วย นอกเหนือจากนั้นสามารถเขียนฟังก์ชันดังกล่าวนี้ให้อยู่ในรูปแบบเลขฐานสิบหกได้โดยใช้ตารางที่ 1.3

ตัวอย่างที่ 1.28 กำหนด $f(x) = x^3 + x + 1$ จงเขียนแบบเลขฐานสอง

วิธีทำ เนื่องจากการแปลงจากฟังก์ชันพหุนามเหนือฟิลด์จำกัด GF(2^m) ในรูปแบบเลขฐานสองจำเป็นต้องรวมสัมประสิทธิที่มีค่าเป็น 0 ด้วยจึงเขียน $f(x)$ ใหม่ได้ดังนี้

$$f(x) = x^3 + 0x^2 + x + 1$$

ดังนั้นจึงสามารถเขียน $f(x)$ ให้อยู่ในรูปแบบของเลขฐานสองได้ดังนี้

$$f(x) = 1011_2 = B_{16}$$

ในทางกลับกันหากฟังก์ชันพหุนามอยู่ในรูปของเลขฐานสองหรือฐานสิบหกสามารถแปลงกลับกับให้อยู่ในรูปแบบเดิมได้โดยใช้หลักการเดียวกัน

ตัวอย่างที่ 1.29 กำหนด $f(x) = 9C_{16}$ จงเขียนฟังก์ฟันพหุนามของค่าดังกล่าว

วิธีทำ เริ่มจากแปลง $9C_{16}$ ให้อยู่ในรูปแบบเลขฐานสองได้ดังนี้

$$9C_{16} = 10011100$$

ดังนั้น
$$f(x) = x^7 + 0x^6 + 0x^5 + x^4 + x^3 + x^2 + 0x + 0$$
$$= x^7 + x^4 + x^3 + x^2$$

13. บทสรุปสาระสำคัญ

วิทยาการรหัสลับ คือศาสตร์ที่ใช้สำหรับรักษาความปลอดภัยข้อมูลข่าวสารผ่านกระบวนการเข้ารหัสลับและการถอดรหัสลับ ซึ่งถูกแบ่งออกเป็นสองวิธีคือวิทยาการรหัสลับแบบสมมาตร และวิทยาการรหัสลับแบบอสมมาตรซึ่งมีข้อดีข้อเสียที่แตกต่างกัน จุดเด่นของวิทยาการรหัสลับแบบสมมาตรคือความเร็วในการประมวลผล อย่างไรก็ตามเกิดปัญหาเกี่ยวกับความยุ่งยากในการแลกเปลี่ยนกุญแจลับระหว่างผู้รับและผู้ส่ง โดยปัญหาดังกล่าวถูกแก้ไขโดยใช้วิทยาการรหัสลับแบบอสมมาตรซึ่งใช้กุญแจคู่ที่เรียกว่ากุญแจสาธารณะ และกุญแจส่วนตัวสำหรับการดำเนินการ เนื่องจากวิทยาการรหัสลับแบบอสมมาตรจำเป็นต้องใช้ทรัพยากรสำหรับการประมวลผลสูง การใช้งานจึงนิยมนำทั้งสองวิธีมาประยุกต์ร่วมกันโดยใช้วิทยาการรหัสลับแบบอสมมาตรสำหรับการแลกเปลี่ยนกุญแจลับ และใช้วิทยาการรหัสลับแบบสมมาตรสำหรับการรักษาความปลอดภัยข้อมูลข่าวสาร

แบบฝึกหัดท้ายบท
บทที่ 1

1. จงแปลงเลขฐานต่อไปนี้ 11001_2 เป็นเลขฐานสิบ
2. จงแปลงเลขฐานต่อไปนี้ $1A2_{16}$ เป็นเลขฐานสอง
3. กำหนดให้ A คือเซตของจำนวนเต็มบวกคู่ที่มีค่าน้อยกว่า 10 จงเขียนเซตของ A แบบแจกแจงสมาชิก
4. กำหนดให้ A คือเซตของจำนวนเต็มที่เป็นผลเฉลยของสมการต่อไปนี้ $x^2 + 2x - 35 = 0$ จงเขียนเซตของ A แบบบอกเงื่อนไข
5. จากคำถามข้อ 4 จงเขียนเซตของ A แบบแจกแจงสมาชิก
6. จากคำถามข้อ 4 หากต้องการผลเฉลยที่มีค่าเป็นจำนวนเต็มบวกเท่านั้นจงเขียนเซตของ A แบบบอกเงื่อนไขใหม่
7. จงคำนวณหาผลลัพธ์ของ 149 mod 13
8. จงคำนวณหาผลลัพธ์ของ 29 ที่อยู่ในฟิลด์ GF(11)
9. กำหนดให้ x mod 19 = 4 และ y mod 19 = 18 จงหา $(x + y)$ mod 19
10. กำหนดให้ x mod 19 = 4 และ y mod 19 = 18 จงหา xy mod 19
11. จงคำนวณหา gcd(287, 469)
12. จงคำนวณหาผลลัพธ์ของ 3^{-1} mod 23
13. จงคำนวณหาผลลัพธ์ของ 3^{-1} mod 53
14. กำหนดให้ $f(x) = x^4 + x^2 + 1$ และ $g(x) = x^3 + x^2$ จงหา $f(x) + g(x)$
15. กำหนดให้ $f(x) = A3_{16}$ จงเขียนฟังก์ฟันพหุนามของค่าดังกล่าว
16. กำหนด $f(x) = x^3 + x^2 + 1$ และ $g(x) = x^2 + x + 1$ และกำหนดให้ฟังก์ชันพหุนามไม่ลดรูป $h(x) = x^4 + x + 1$ จงคำนวณหา $f(x)$ x $g(x)$
17. กำหนด $f(x) = x + 1$ และ $g(x) = x^4 + x + 1$ จงคำนวณหา $f^{-1}(x)$

บทที่ 2
วิทยาการรหัสลับแบบสมมาตร

บทนี้จะกล่าวถึงวิทยาการรหัสลับแบบสมมาตรที่ได้เคยถูกนำมาใช้งานจริง [1], [2], [5] โดยวิทยาการรหัสลับที่นำเสนอในบทนี้มีความเรียบง่าย และไม่มีความซับซ้อนซึ่งจะช่วยให้ผู้อ่านสามารถศึกษาเพื่อให้มีความรู้ความเข้าใจเกี่ยวกับหลักการของวิทยาการรหัสลับได้ง่ายมากยิ่งขึ้น อย่างไรก็ตามรหัสลับส่วนมากในบทนี้เป็นกลุ่มที่ถูกโจมตีได้ง่ายจึงไม่ถูกนำมาใช้งานในปัจจุบัน

1. รหัสซีซาร์ (Caesar Cipher)

รหัสซีซาร์ [43] ถูกคิดค้นโดย จูเลียส ซีซาร์ (Julius Caesar) แม่ทัพทหารชาวโรมันเพื่อใช้สำหรับการสื่อสารข้อมูลราชการลับทางทหาร โดยแรกเริ่ม จูเลียส ซีซาร์ ใช้หลักการเข้ารหัสโดยการเลื่อนตัวอักษรภาษาอังกฤษแต่ละตัวอักษรในประโยค 3 ตำแหน่ง และการถอดรหัสสามารถทำได้โดยการเลื่อนตัวอักษรแต่ละตัวกลับมา 3 ตำแหน่ง พิจารณาตำแหน่งของตัวอักษรภาษาอังกฤษได้ทั้งหมดดังตารางที่ 2.1

ตารางที่ 2.1 แสดงตำแหน่งของตัวอักษรภาษาอังกฤษทั้งหมด 26 ตัว สมมติหากผู้ใช้งานต้องการเข้ารหัส “A” ซึ่งอยู่ตำแหน่งที่ 0 ผู้ใช้งานต้องทำการเลื่อนตัวอักษรไปอีก 3 ตำแหน่งซึ่งตกตำแหน่งที่ 3 จึงได้เป็นตัวอักษร “D” อย่างไรก็ตามตัวอักษร “X”, “Y” และ “Z” หากถูกเข้ารหัสแล้ว สังเกตได้ว่าการเลื่อน 3 ตำแหน่งจะเกินตำแหน่งที่ 25 ทั้งหมด ดังนั้นทั้ง 3 ตำแหน่งซึ่งเป็นตำแหน่งที่เริ่มเกินจะถูกวนไปที่ตำแหน่งที่ “0”, “1” และ “2” ตามลำดับ การเข้ารหัสตัวอักษร “X”, “Y” และ “Z” จึงได้เป็น “A”, “B” และ “C” ตามลำดับ

ตัวอย่างที่ 2.1 จงแสดงวิธีการเข้ารหัสข้อความ COMPUTER ด้วยรหัสซีซาร์แบบดั้งเดิม
วิธีทำ จากตารางที่ 2.1 สามารถหาข้อความไซเฟอร์ของข้อความ COMPUTER สำหรับแต่ละตัวอักษรได้ดังนี้

1. “C” อยู่ตำแหน่งที่ 2 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 5 ได้เป็นตัวอักษร “F”
2. “O” อยู่ตำแหน่งที่ 14 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 17 ได้เป็นตัวอักษร “R”
3. “M” อยู่ตำแหน่งที่ 12 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 15 ได้เป็นตัวอักษร “P”
4. “P” อยู่ตำแหน่งที่ 15 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 18 ได้เป็นตัวอักษร “S”
5. “U” อยู่ตำแหน่งที่ 20 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 23 ได้เป็นตัวอักษร “X”

6. “T” อยู่ตำแหน่งที่ 19 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 22 ได้เป็นตัวอักษร “W”
7. “E” อยู่ตำแหน่งที่ 4 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 7 ได้เป็นตัวอักษร “H”
8. “R” อยู่ตำแหน่งที่ 17 โดยหลังจากเลื่อนไป 3 ตำแหน่งจะตกตำแหน่งที่ 20 ได้เป็นตัวอักษร “U”

ดังนั้นหลังจากกระบวนการเข้ารหัสสิ้นสุดจะได้ข้อความไซเฟอร์ คือ FRPSXWHU

ตารางที่ 2.1 ตำแหน่งตัวอักษรภาษาอังกฤษสำหรับรหัสซีซาร์

ตำแหน่ง	ตำอักษร		ตำแหน่ง	ตำอักษร
0	A		13	N
1	B		14	O
2	C		15	P
3	D		16	Q
4	E		17	R
5	F		18	S
6	G		19	T
7	H		20	U
8	I		21	V
9	J		22	W
10	K		23	X
11	L		24	Y
12	M		25	Z

อย่างไรก็ตามการค้นหาข้อความต้นฉบับจากข้อความไซเฟอร์ที่ผ่านกระบวนการเข้ารหัสซีซาร์แบบดั้งเดิมสามารถดำเนินการได้ง่ายมาก เนื่องจากค่ากุญแจมีเพียงค่าเดียวคือค่า 3 ดังนั้นหากผู้บุกรุกทราบข้อความไซเฟอร์จะสามารถคำนวณหาข้อความต้นฉบับโดยการเลื่อนตัวอักษรแต่ละตัวกลับมา 3 ตำแหน่ง

ดังนั้น จึงมีการปรับปรุงรหัสซีซาร์ใหม่โดยการเพิ่มจำนวนของกุญแจให้มากขึ้นเป็น 26 ค่าซึ่งมีค่าอยู่ระหว่าง 0 – 25 และมีค่าเท่ากับจำนวนตัวอักษรภาษาอังกฤษ

กำหนดให้ k คือกุญแจลับโดยที่ $0 \leq k \leq 25$ ได้สมการเข้ารหัสลับและถอดรหัสลับเป็นดังนี้

การเข้ารหัส

$$c = (m + k) \bmod 26 \qquad (2.1)$$

การถอดรหัส

$$m = (c - k) \bmod 26 \qquad (2.2)$$

เมื่อ m คือ ข้อความต้นฉบับซึ่งเป็นค่าตำแหน่งของตัวอักษรภาษาอังกฤษจากตารางที่ 2.1 จำนวน 1 ตัวอักษร

c คือ ข้อความไซเฟอร์ซึ่งเป็นตัวอักษรภาษาอังกฤษที่แปลงจากค่าตำแหน่งที่ได้จากการคำนวณ

ตัวอย่างที่ 2.2 จงแสดงวิธีการเข้ารหัสข้อความ COMPUTER ด้วยรหัสซีซาร์แบบปรับปรุงโดยใช้ค่า $k = 18$

วิธีทำ

เนื่องจากโจทย์กำหนด $k = 18$ ดังนั้นจากสมการที่ (2.1) ได้ข้อความไซเฟอร์เป็นดังต่อไปนี้

1. เข้ารหัส “C”, ได้ $m = 2$, $c = (2 + 18) \bmod 26 = 20$, ดังนั้น ข้อความไซเฟอร์คือ “U”
2. เข้ารหัส “O”, ได้ $m = 14$, $c = (14 + 18) \bmod 26 = 6$, ดังนั้น ข้อความไซเฟอร์คือ “G”
3. เข้ารหัส “M”, ได้ $m = 12$, $c = (12 + 18) \bmod 26 = 4$, ดังนั้น ข้อความไซเฟอร์คือ “E”
4. เข้ารหัส “P”, ได้ $m = 15$, $c = (15 + 18) \bmod 26 = 7$, ดังนั้น ข้อความไซเฟอร์คือ “H”
5. เข้ารหัส “U”, ได้ $m = 20$, $c = (20 + 18) \bmod 26 = 12$, ดังนั้นข้อความไซเฟอร์คือ “M”
6. เข้ารหัส “T”, ได้ $m = 19$, $c = (19 + 18) \bmod 26 = 11$, ดังนั้น ข้อความไซเฟอร์คือ “L”
7. เข้ารหัส “E”, ได้ $m = 4$, $c = (4 + 18) \bmod 26 = 22$, ดังนั้น ข้อความไซเฟอร์คือ “W”
8. เข้ารหัส “R”, ได้ $m = 17$, $c = (17 + 18) \bmod 26 = 9$, ดังนั้น ข้อความไซเฟอร์คือ “J”

ดังนั้นหลังจากกระบวนการเข้ารหัสสิ้นสุดจะได้ข้อความไซเฟอร์ คือ UGEHMLWJ

อย่างไรก็ตามเนื่องจากขนาดกุญแจที่เป็นไปได้ทั้งหมดมีจำนวนน้อยมาก ส่งผลให้ผู้ไม่ประสงค์ดีสามารถค้นหาค่ากุญแจที่แท้จริงได้อย่างรวดเร็ว โดยใช้ค่ากุญแจที่เป็นไปได้ทั้งหมด สำหรับทดสอบถอดรหัสข้อความไซเฟอร์ ซึ่งหากข้อความที่เป็นผลลัพธ์จากกระบวนการถอดรหัสเป็นข้อความที่มีความหมายจะสามารถคาดการณ์ได้ว่าค่ากุญแจที่ทำให้ได้ข้อความดังกล่าวเป็นค่ากุญแจที่แท้จริง

ตารางที่ 2.2 การถอดรหัสข้อความ UGEHMLWJ ด้วยรหัสซีซาร์โดยใช้ค่ากุญแจที่เป็นไปได้ทั้งหมด

ค่ากุญแจ (k)	**ผลลัพธ์จากกระบวนการถอดรหัส**
0	UGEHMLWJ
1	TFDGLKVI
2	SECFKJUH
3	RDBEJITG
4	QCADIHSF
5	PBZCHGRE
6	OAYBGFQD
7	NZXAFEPC
8	MYWZEDOB
9	LXVYDCNA
10	KWUXCBMZ
11	JVTWBALY
12	IUSVAZKX
13	HTRUZYJW
14	GSQTYXIV
15	FRPSXWHU
16	EQORWVGT
17	DPNQVUFS
18	**COMPUTER**
19	BNLOTSDQ
20	AMKNSRCP
21	ZLJMRQBO
22	YKILQPAN
23	XJHKPOZM
24	WIGJONYL
25	VHFINMXK

ตารางที่ 2.2 แสดงการตะลุยหาค่ากุญแจที่เป็นไปได้ทั้งหมดสำหรับค้นหาข้อความต้นฉบับของ “UGEHMLWJ” (จากตัวอย่างที่ 2.1) พบว่าหลังกระบวนการถอดรหัสมีเพียงข้อความเดียวเท่านั้นที่มีความหมายคือข้อความ “COMPUTER” ซึ่งเกิดจากการใช้ค่า k = 18 ดังนั้นผู้ไม่ประสงค์ดีจึงสามารถคาดการณ์ได้ว่ารหัสลับนี้เกิดจากการใช้รหัสซีซาร์ด้วยค่ากุญแจดังกล่าว

ดังนั้นจำนวนรอบสูงสุดที่ใช้สำหรับการโจมตีรหัสซีซาร์คือ 26 รอบ ซึ่งเท่ากับจำนวนของค่ากุญแจลับที่เป็นไปได้ทั้งหมด

2. รหัสสับเปลี่ยน (Substitute Cipher)

จากปัญหาของรหัสซีซาร์ซึ่งกุญแจมีขนาดที่เล็กมาก ส่งผลให้การโจมตีสามารถทำได้อย่างรวดเร็ว ดังนั้นจึงได้มีการปรับปรุงรหัสลับซีซาร์ใหม่เพื่อขยายขนาดกุญแจให้ใหญ่มากขึ้นโดยได้รหัสลับใหม่ที่เรียกว่ารหัสสับเปลี่ยน [56] ที่มีกระบวนการเข้ารหัสแบบใช้การแทนที่ตัวอักษรภาษาอังกฤษตัวหนึ่ง ด้วยตัวอักษรภาษาอังกฤษตัวอื่นที่แตกต่างออกไป ยกตัวอย่างเช่นจากตารางที่ 2.3 การแทนตัวอักษร “A” ด้วยตัวอักษร “X” ความหมายคือ ตัวอักษร “A” ทุกตัวเมื่อถูกเข้ารหัสแล้วจะถูกแทนที่ด้วยตัวอักษร “X” อย่างไรก็ตามตัวอักษรที่ถูกนำมาใช้จับคู่กับตัวอักษรต้นฉบับแล้ว จะไม่สามารถถูกนำไปจับคู่กับตัวอักษรตัวอื่นได้ เนื่องจากต้องเป็นการจับคู่แบบหนึ่งต่อหนึ่ง

ตารางที่ 2.3 ตัวอย่างการสับเปลี่ยนตัวอักษรเพื่อใช้สำหรับรหัสสับเปลี่ยน

อักษรต้นฉบับ	A	B	C	D	E	F	G	H	I	J	K	L	M
อักษรแทนที่	X	N	H	T	Q	J	Y	V	B	Z	Q	W	E
อักษรต้นฉบับ	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
อักษรแทนที่	D	M	U	I	G	A	L	R	C	P	S	K	F

ตัวอย่างที่ 2.3 จงแสดงวิธีการเข้ารหัสข้อความ COMPUTER ด้วยรหัสสับเปลี่ยนโดยใช้กุญแจดังตารางที่ 2.3

วิธีทำ จากตารางที่ 2.3 การเข้ารหัสเป็นดังนี้

1. เข้ารหัส “C”, ตัวอักษรต้นฉบับ “C” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “H”
2. เข้ารหัส “O”, ตัวอักษรต้นฉบับ “O” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “M”
3. เข้ารหัส “M”, ตัวอักษรต้นฉบับ “M” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “E”
4. เข้ารหัส “P”, ตัวอักษรต้นฉบับ “P” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “U”
5. เข้ารหัส “U”, ตัวอักษรต้นฉบับ “U” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “R”
6. เข้ารหัส “T”, ตัวอักษรต้นฉบับ “T” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “L”
7. เข้ารหัส “E”, ตัวอักษรต้นฉบับ “E” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “O”
8. เข้ารหัส “R”, ตัวอักษรต้นฉบับ “R” จะถูกแทนที่ด้วย ตัวอักษรแทนที่ “G”

ดังนั้นหลังจากกระบวนการเข้ารหัสสิ้นสุดได้ข้อความไซเฟอร์ คือ HMEURLOG

สำหรับกระบวนการถอดรหัสสามารถดำเนินการได้โดยการพิจารณาตัวอักษรถูกแทนที่เป็นตัวอักษรต้นฉบับ และตัวอักษรต้นฉบับเป็นตัวอักษรถูกแทนที่แทน ยกตัวอย่างเช่น จากตารางที่ 2.3 การถอดรหัสตัวอักษร “T” จะได้เป็นตัวอักษร “D” เป็นต้น

เนื่องจากขนาดกุญแจที่เป็นไปได้ทั้งหมดสำหรับรหัสสับเปลี่ยนคือ 26! ซึ่งเป็นจำนวนที่ใหญ่มหาศาลมาก ดังนั้นจึงเป็นเรื่องที่ยากมากสำหรับการโจมตีแบบตะลุย อย่างไรก็ตามสามารถโจมตีรหัสสับเปลี่ยนได้โดยง่ายด้วยวิธีการวิเคราะห์ความถี่ของการเกิดตัวอักษรโดยใช้วิธีการวิเคราะห์ตัวอักษรของข้อความต้นฉบับได้จากบทความทั่วไป และวิเคราะห์ข้อความไซเฟอร์ที่ต้องการถอดรหัส โดยตัวอักษรที่มีความถี่สูงของทั้งสองส่วนจะถูกคาดการณ์เป็นตัวอักษรที่ถูกใช้สำหรับการแทนที่ซึ่งกันและกัน นอกเหนือจากนี้แล้วยังสามารถนำวิธีการคาดคะเนคำศัพท์เพื่อเป็นเครื่องมือช่วยสำหรับการวิเคราะห์รหัสสับเปลี่ยนร่วมกับการวิเคราะห์ความถี่การเกิดตัวอักษรได้

3. รหัสสัมพรรค (Affine Cipher)

รหัสสัมพรรค [43] เป็นรหัสลับอีกประเภทที่ปรับปรุงมาจากรหัสซีซาร์เพื่อขยายขนาดกุญแจให้ใหญ่มากยิ่งขึ้น

กำหนดให้ K คือ กุญแจลับสำหรับรหัสสัมพรรคโดยมีนิยาม ดังนี้

$K = \{(a, b) \in GF(26) \mid \gcd(a, 26) = 1\}$

ได้สมการเข้ารหัสเป็น ดังนี้

$$c = (am + b) \bmod 26 \quad (2.3)$$

และสมการถอดรหัสเป็นดังนี้

$$m = a^{-1}(c - b) \bmod 26 \quad (2.4)$$

เมื่อ m คือ ข้อความต้นฉบับซึ่งเป็นค่าตำแหน่งของตัวอักษรภาษาอังกฤษจากตารางที่ 2.1 จำนวน 1 ตัวอักษร

c คือ ข้อความไซเฟอร์ซึ่งเป็นตัวอักษรภาษาอังกฤษที่แปลงจากค่าตำแหน่งที่ได้จากการคำนวณสมการ (2.3)

อย่างไรตาม เนื่องจาก gcd(a, 26) = 1 ดังนั้นสำหรับค่า a บางค่าจะไม่สามารถถูกนำมาใช้เป็นกุญแจลับสำหรับรหัสสัมพรรคได้ ตารางที่ 2.4 แสดงค่า a ที่เป็นไปได้ทั้งหมดทั้งที่สามารถใช้งานได้และไม่สามารถใช้งานได้

ตารางที่ 2.4 ผลลัพธ์การหารร่วมมากระหว่าง a และ 26

a	gcd(a, 26)	a	gcd(a, 26)
0	26	13	13
1	**1**	14	2
2	2	15	**1**
3	**1**	16	2
4	2	17	**1**
5	**1**	18	2
6	2	19	**1**
7	**1**	20	2
8	2	21	**1**
9	**1**	22	2
10	2	23	**1**
11	**1**	24	2
12	2	25	**1**

จากตารางที่ 2.4 สังเกตได้ว่าค่า a ที่ส่งผลให้ gcd(a, 26) = 1 มีทั้งหมด 12 ค่าประกอบด้วย 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 และ 25 ซึ่งเป็นกลุ่มค่าที่สามารถนำมาใช้เป็นกุญแจลับสำหรับรหัสสัมพรรคได้

บทนิยามที่ 2.1 กำหนดให้ $m \in \mathbb{Z}^+$ และ a เป็นสมาชิกใน GF(m) ได้ว่า a มีตัวผกผันมอดูโล m ก็ต่อเมื่อ gcd(a, m) = 1 โดยที่จำนวนสมาชิกทั้งหมดที่มีตัวผกผันถูกเขียนแทนด้วย GF*(m) หรือ

$$GF^*(m) = \{a \in \mathbb{Z}^+ \mid \gcd(a, m) = 1\}$$

ตัวอย่างที่ 2.4 จากตารางที่ 2.4 ได้ว่าจำนวนสมาชิกของ a ที่มีตัวผกผันมอดูโล 26 และสามารถใช้งานได้สำหรับรหัสสัมพรรคมีจำนวน 12 ค่าคือ 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 และ 25

ตัวอย่างที่ 2.5 จงแสดงวิธีการเข้ารหัสข้อความ ANT ด้วยรหัสสัมพรรคโดยใช้กุญแจ a = 7 และ b = 14

วิธีทำ จากสมการที่ (2.3) และตารางที่ 2.1 ได้ผลลัพธ์เป็นดังนี้

1. เข้ารหัส “A”, m = 0, c = (0 x 7 + 14) mod 26 = 14, ดังนั้นข้อความไซเฟอร์คือ “O”
2. เข้ารหัส “N”, m = 13, c = (13 x 7 + 14) mod 26 = 1, ดังนั้นข้อความไซเฟอร์คือ “B”
3. เข้ารหัส “T”, m = 19, c = (19 x 7 + 14) mod 26 = 17, ดังนั้นข้อความไซเฟอร์คือ “R”

ดังนั้นหลังจากกระบวนการเข้ารหัสสิ้นสุดได้ข้อความไซเฟอร์ คือ OBR

ตัวอย่างที่ 2.6 จงถอดรหัสข้อความไซเฟอร์ที่ได้จากตัวอย่างที่ 2.5

วิธีทำ จากสมการ (2.4) จำเป็นต้องคำนวณหาค่า a^{-1} ใน GF*(26) โดยใช้ขั้นตอนวิธียุคลิดภาคขยาย ซึ่งได้เป็น 15 เนื่องมาจาก 15x7 mod 26 = 105 mod 26 = 1 ดังนั้น a^{-1} = 15, และได้ผลลัพธ์เป็นดังนี้

1. ถอดรหัส “O”, c = 14, m = 15(14 - 14) mod 26 = 0, ดังนั้น ข้อความต้นฉบับคือ “A”
2. ถอดรหัส “B”, c = 1, m = 15(1 - 14) mod 26 = 13, ดังนั้น ข้อความไซเฟอร์คือ “N”
3. ถอดรหัส “R”, c = 17, m = 15(17 - 14) mod 26 = 19, ดังนั้น ข้อความไซเฟอร์คือ “T”

ดังนั้นหลังจากกระบวนการถอดรหัสสิ้นสุดได้ข้อความต้นฉบับ คือ ANT

อย่างไรก็ตามหากเลือกค่า a ที่นอกเหนือจาก 12 ค่าดังตารางที่ 2.4 ส่งผลให้กระบวนการเข้ารหัสของตัวอักษรบางค่าที่แตกต่างกันอาจได้ผลลัพธ์เป็นข้อความไซเฟอร์ที่เป็นตัวอักษรเดียวกัน ซึ่งหากนำข้อความไซเฟอร์ตัวดังกล่าวไปผ่านกระบวนการถอดรหัสจะได้ค่าที่ไม่ถูกต้องอย่างแน่นอน ตัวอย่างเช่น หากต้องการเข้ารหัสตัวอักษร “I” ด้วยกุญแจ $a = 4$ และ $b = 5$ จะได้ข้อความไซเฟอร์คือตัวอักษร “L” ในทางกลับกันหากต้องการเข้ารหัสตัวอักษร “V” ด้วยกุญแจค่าเดียวกันจะได้ข้อความไซเฟอร์คือตัวอักษร “L” เช่นเดียวกัน เป็นต้น ซึ่งสังเกตได้ว่าตัวอักษรแตกต่างกัน 2 ค่า เมื่อผ่านกระบวนการเข้ารหัสแล้วได้ข้อความไซเฟอร์ที่เป็นตัวอักษรตัวเดียวกัน ซึ่งไม่เป็นแบบหนึ่งต่อหนึ่ง ดังนั้นหลังกระบวนการถอดรหัสสิ้นสุดจะไม่ได้ข้อความต้นฉบับที่ถูกต้องอย่างแน่นอน

เนื่องจากกุญแจที่เป็นไปได้สำหรับ a และ b มีทั้งหมด 12 และ 26 ค่า ตามลำดับ ดังนั้นกุญแจที่เป็นไปได้ทั้งหมดสำหรับรหัสสัมพรรคคือ $12 \times 26 = 312$ ค่า ถึงแม้ว่าจำนวนกุญแจจะเพิ่มขึ้นมาอย่างมากหากเปรียบเทียบกับรหัสซีซาร์แต่ยังคงสามารถใช้การโจมตีแบบตะลุยหากุญแจได้อย่างรวดเร็ว เนื่องจากจำนวนกุญแจยังคงมีจำนวนไม่มาก

4. รหัสวีเกเนร์ (Vigenere Cipher)

วิทยาการรหัสลับทั้ง 3 ชนิดที่กล่าวมาก่อนหน้านี้เป็นวิทยาการรหัสลับที่ใช้กุญแจลับในลักษณะตัวอักษรเดี่ยวเรียกการเข้ารหัสลับลักษณะดังกล่าวนี้ว่าวิทยาการรหัสลับอักขระเดี่ยว (Monoalphabetic cryptosystem) ซึ่งมีข้อเสียคือง่ายแก่การโจมตี จึงมีการนำเสนอรหัสวีเกเนร์ [57] ซึ่งเป็นวิทยาการรหัสลับที่ใช้ตัวอักษรมากกว่า 1 ตัวอักษรซึ่งมีขนาดที่ไม่ตายตัวเพื่อเรียงกันเป็นกุญแจลับ โดยการเข้ารหัสลับในแต่ละตำแหน่งพิจารณาได้จากตำแหน่งที่ตรงกันระหว่างตัวอักษรของกุญแจลับ และตัวอักษรของข้อความต้นฉบับ โดยใช้วิธีการเข้ารหัสลับ และถอดรหัสลับด้วยวิธีแบบซีซาร์ ยกตัวอย่างเช่น ข้อความต้นฉบับคือ BAT และ กุญแจคือ COM กระบวนการเข้ารหัสคือ จะต้องใช้ตัวอักษร “M” เพื่อเป็นกุญแจสำหรับกระบวนการเข้ารหัสตัวอักษร “T”, จะต้องใช้ตัวอักษร “O” เพื่อเป็นกุญแจสำหรับกระบวนการเข้ารหัสตัวอักษร “A” และ จะต้องใช้ตัวอักษร “C” เพื่อเป็นกุญแจสำหรับกระบวนการเข้ารหัสตัวอักษร “B” ตามลำดับ เป็นต้น ในทางกลับกันกระบวนการถอดรหัสสามารถดำเนินการในลักษณะเดียวกับกระบวนการเข้ารหัสคือ การถอดรหัสลับแต่ในละตำแหน่งพิจารณาได้จากตำแหน่งที่ตรงกันระหว่างตัวอักษรของกุญแจลับ และตัวอักษรของข้อความไซเฟอร์ อย่างไรก็ตามในกรณีที่จำนวนตัวอักษรของข้อความต้นฉบับ หรือข้อความไซเฟอร์ไม่เท่ากับจำนวนตัวอักษรของกุญแจลับจะต้องทำให้จำนวนตัวอักษรเท่ากันก่อน โดยใช้วิธีการนำชุดข้อมูลของกลุ่มตัวอักษรที่มีขนาดเล็กกว่ามาเรียงต่อกันเพื่อให้ได้ขนาดเท่ากับกลุ่มตัวอักษรที่มีขนาดที่ใหญ่กว่า

ตัวอย่างที่ 2.7 จงแสดงวิธีการเข้ารหัสข้อความ CRYPTOGRAPHY ด้วยรหัสวีเกเนร์ โดยใช้ชุดของกุญแจลับเป็นดังนี้ K = {2, 5, 7, 9}

วิธีทำ จากตารางที่ 2.1 สามารถแปลงข้อมูลเป็นค่าประจำตำแหน่ง ได้ดังนี้

ตำแหน่ง	11	10	9	8	7	6	5	4	3	2	1	0
ข้อความต้นฉบับ	C	R	Y	P	T	O	G	R	A	P	H	Y
ค่าประจำตำแหน่ง	2	17	24	15	19	14	6	17	0	15	7	24

หลังจากนั้นนำค่ากุญแจลับมาวางเรียงจากตำแหน่งต่ำที่สุดไปยังตำแหน่งสูงที่สุด เนื่องจากกุญแจลับมีขนาดเพียง 4 ค่า ซึ่งน้อยกว่าข้อความต้นฉบับที่มีจำนวนตัวอักษรทั้งหมด 12 ตัวอักษร จึงจำเป็นต้องขยายขนาดของกุญแจลับให้เท่ากับจำนวนตัวอักษรของข้อความต้นฉบับ โดยการนำค่ากุญแจชุดเดิมมาต่อในตำแหน่งที่ยังขาดเหลือ ดังนี้

ตำแหน่ง	11	10	9	8	7	6	5	4	3	2	1	0
ข้อความต้นฉบับ	C	R	Y	P	T	O	G	R	A	P	H	Y
ค่าประจำตำแหน่ง	2	17	24	15	19	14	6	17	0	15	7	24
ค่ากุญแจ	2	5	7	9	2	5	7	9	2	5	7	9

จากตารางข้างต้นสังเกตได้ว่ามีการนำกุญแจลับค่าเดิมกลับมาใช้ใหม่ 3 รอบเพื่อให้เท่ากับจำนวนตัวอักษรของข้อความต้นฉบับ

ขั้นตอนสุดท้าย ดำเนินการเข้ารหัสข้อมูลด้วยตัวอักษรของข้อความ และค่ากุญแจในตำแหน่งที่ตรงกันด้วยรหัสซีซาร์

ตำแหน่ง	11	10	9	8	7	6	5	4	3	2	1	0
ข้อความต้นฉบับ	C	R	Y	P	T	O	G	R	A	P	H	Y
ค่าประจำตำแหน่ง	2	17	24	15	19	14	6	17	0	15	7	24
ค่ากุญแจ	2	5	7	9	2	5	7	9	2	5	7	9
เข้ารหัสลับ	4	22	7	24	21	19	13	0	2	20	14	7
ข้อความไซเฟอร์	E	W	H	Y	V	T	U	A	C	Z	O	H

ดังนั้นหลังจากกระบวนการเข้ารหัสลับสิ้นสุดได้ข้อความไซเฟอร์คือ EWHYVTUACZOH

จากตารางผลลัพธ์ สังเกตว่าข้อความไซเฟอร์ในตำแหน่งที่ 4 และ ตำแหน่งที่ 10 มีค่าที่แตกต่างกันถึงแม้ว่าจะมีค่าข้อความต้นฉบับเป็นค่าเดียวกัน เนื่องมาจากค่าความแตกต่างของกุญแจลับ

อย่างไรก็ตาม ถึงแม้ว่ารหัสวีเกเนร์จะมีความแข็งแกร่งมากขึ้น ยากแก่การโจมตีแบบตะลุยแต่ยังคงสามารถโจมตีรหัสลับดังกล่าวนี้ได้ด้วยวิธีอื่น คือ ใช้ทฤษฎีความน่าจะเป็นมาใช้สำหรับการวิเคราะห์หาข้อความต้นฉบับ เป็นต้น

5. รหัสฮิลล์ (Hill Cipher)

รหัสฮิลล์คือวิทยาการรหัสลับแบบสมมาตรอีกรูปแบบหนึ่งที่ถูกคิดค้นโดยเลสเตอร์ ฮิลล์ (Hill Lester S.) [52] โดยนำหลักการของเมตริกซ์มาเป็นเครื่องมือที่ใช้สำหรับกระบวนการเข้ารหัส และถอดรหัสข้อมูล สำหรับข้อความต้นฉบับ กุญแจลับ หรือข้อความไซเฟอร์จะต้องเป็นสมาชิกเหนือฟิลด์ GF(26) โดยค่ากุญแจลับจะต้องอยู่ในรูปแบบของเมตริกซ์จัตุรัส ดังนี้

กำหนดให้ K คือกุญแจลับที่เป็นเมตริกซ์ ขนาด $m \times m$

x คือ ข้อความต้นฉบับ

หาก x มีจำนวนตัวอักษรมากกว่า m ตัวอักษร จะต้องแบ่งเป็นชุด ชุดละ m ตัวอักษรโดยเริ่มแบ่งจากตำแหน่งหลังสุดไปยังตำแหน่งที่อยู่หน้าสุด (จากตำแหน่งขวามือสุดไปยังตำแหน่งซ้ายมือสุด)

สมการเข้ารหัสเป็นดังนี้

$$y_i = x_i k \pmod{26} \tag{2.5}$$

และสมการถอดรหัสเป็นดังนี้

$$x_i = y_i k^{-1} \pmod{26} \tag{2.6}$$

เมื่อ x_i คือ = ข้อความต้นฉบับย่อยของ x ที่ถูกแบ่งให้มีจำนวน m ตัวอักษร

y_i คือ = ข้อความไซเฟอร์ย่อยที่ถูกแบ่งให้มีจำนวน m ตัวอักษร

ตัวอย่างที่ 2.8 จงแสดงวิธีการเข้ารหัสข้อความ deck ด้วยรหัสฮิลล์ โดยใช้ชุดของกุญแจลับขนาด 2x2 เป็นดังนี้ $k = \begin{bmatrix} 7 & 4 \\ 5 & 11 \end{bmatrix}$

วิธีทำ เนื่องจากขนาดของ k เป็น 2x2 ดังนั้น deck ต้องแบ่งเป็น 2 ส่วน ดังนี้

x_0 = ck = (2, 10) และ x_1 = de = (3, 4)

และการเข้ารหัสจึงจำเป็นต้องแบ่งเป็น 2 ส่วน ดังนี้

เข้ารหัส x_0: $y_0 = [2 \ \ 10] \times \begin{bmatrix} 7 & 4 \\ 5 & 11 \end{bmatrix} = [((2x7)+(10x5)) \quad ((2 \times 4)+(10 \times 11))]$

$= [64 \quad 118]$

อย่างไรก็ตามเนื่องจากการดำเนินการทั้งหมดอยู่เหนือฟิลด์ GF(26) ได้ว่า

$y_0 = [64 \ 118] \pmod{26}$

$= [12 \ 14]$ = mo

เข้ารหัส x_1: $y_1 = [3 \ \ 4] \times \begin{bmatrix} 7 & 4 \\ 5 & 11 \end{bmatrix} = [((3 \times 7)+(4 \times 5)) \ ((3 \times 4)+(4 \times 11))]$

$= [41 \ 66]$

อย่างไรก็ตามเนื่องจากการดำเนินการทั้งหมดอยู่เหนือฟิลด์ GF(26) ได้ว่า

$y_1 = [41 \ 66] \pmod{26}$

$= [15 \ 4]$ = pe

ดังนั้นได้ข้อความไซเฟอร์ เป็น pemo

ตัวอย่างที่ 2.9 จากตัวอย่างที่ 2.8 จงแสดงวิธีการถอดรหัสด้วยรหัสฮิลล์

วิธีทำ เนื่องจากสมการถอดรหัสจำเป็นต้องหา k^{-1} ซึ่งเป็นเมตริกซ์ผกผันเหนือฟิลด์ GF(26) ได้ค่าเป็นดังนี้

$$k^{-1} = \begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix}$$

เนื่องจาก

$$\begin{bmatrix} 7 & 4 \\ 5 & 11 \end{bmatrix}\begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix} \bmod 26 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathrm{I}$$

จาก y_0 = [12 14] และ y_1 = [15 4] ได้ผลลัพธ์การถอดรหัสเป็นดังนี้

ถอดรหัส y_0:

$$x_0 = [12\ \ 14]x\begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix} = [((12 \times 23) + (14 \times 25))\ ((12 \times 20) + (14 \times 17))]$$

$$= [626\ \ 478]$$

อย่างไรก็ตามเนื่องจากการดำเนินการทั้งหมดอยู่เหนือฟิลด์ GF(26) ได้ว่า

$$x_0 = [626\ 478] \bmod 26$$

$$= [2\ 10] = ck$$

ถอดรหัส y_1:

$$x_1 = [15\ \ 4]x\begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix} = [((15 \times 23) + (4\ \ \times 25))\ ((15 \times 20) + (4 \times 17))]$$

$$= [445\ 368]$$

อย่างไรก็ตามเนื่องจากการดำเนินการทั้งหมดอยู่เหนือฟิลด์ GF(26) ได้ว่า

$$x_1 = [445\ 368] \bmod 26$$

= [3 4] = de

ดังนั้นได้ข้อความไซเฟอร์ เป็น deck

โดยการโจมตีรหัสฮิลล์หากใช้วิธีการโจมตีในกรณีที่ทราบข้อความไซเฟอร์เท่านั้นสามารถดำเนินการได้ยากโดยเฉพาะอย่างยิ่งเมื่อกุญแจมีขนาดใหญ่ แต่อย่างไรก็ตามสมมติว่าผู้ไม่ประสงค์ดีสามารถเข้าไปครอบครองคู่ความสัมพันธ์ระหว่างข้อความไซเฟอร์ และข้อความต้นฉบับได้หลายคู่ ผู้ไม่ประสงค์ดีสามารถโจมตีรหัสฮิลล์ได้โดยวิธีการโจมตีที่ทราบข้อความต้นฉบับ โดยใช้คุณสมบัติทางเมตริกซ์ ดังนี้

เนื่องจาก $y = xk$

หาก x มีค่าผกผัน (x^{-1} สามารถหาค่าได้) จะได้ว่า

$$x^{-1}y = x^{-1}xk$$
$$= \mathrm{I}\,k$$
$$= k$$

สรุปคือสามารถคำนวณหาค่ากุญแจลับกลับคืนได้จาก

$$k = x^{-1}y \pmod{26} \qquad (2.7)$$

ตัวอย่างที่ 2.10 จงแสดงวิธีการคำนวณหาค่ากุญแจลับ k ขนาด 2x2 สำหรับรหัสฮิลล์ สมมติว่าข้อความต้นฉบับคือ left และข้อความไซเฟอร์คือ pcsh

วิธีทำ เนื่องจาก k มีขนาด 2x2 แต่ข้อความไซเฟอร์และข้อความต้นฉบับมีขนาด 4 ตัวอักษร แสดงว่าข้อมูลถูกแบ่งเป็น 2 ชุด

เนื่องจากค่าตำแหน่งของแต่ละตัวอักษรของ left คือ 11 4 5 19

และค่าตำแหน่งของแต่ละตัวอักษรของ pcsh คือ 15 2 18 7

ดังนั้นหากจำข้อความต้นฉบับ และข้อความไซเฟอร์ที่สัมพันธ์กันทั้งหมดมาเขียนลงเมตริกซ์จะได้เป็นดังนี้

$$X = \begin{bmatrix} 11 & 4 \\ 5 & 19 \end{bmatrix} \text{ และ } Y = \begin{bmatrix} 15 & 2 \\ 18 & 7 \end{bmatrix}$$

เมื่อ

X คือข้อความต้นฉบับที่ข้อมูลแต่ละกลุ่ม

Y คือข้อความไซเฟอร์ที่ข้อมูลในแต่ละแถวต้องสัมพันธ์กับข้อมูลต้นฉบับ

ก่อนการคำนวณหา k จำเป็นต้องหาค่า X^{-1} ก่อน ซึ่งมีค่าเป็นดังนี้

$$X^{-1} = \begin{bmatrix} 25 & 18 \\ 3 & 9 \end{bmatrix}$$

เนื่องจาก

$$\begin{bmatrix} 11 & 4 \\ 5 & 19 \end{bmatrix}\begin{bmatrix} 25 & 18 \\ 3 & 9 \end{bmatrix} \bmod 26 = \begin{bmatrix} 287 & 234 \\ 182 & 261 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \mathrm{I}$$

จาก $k = X^{-1}Y \pmod{26}$

$$= \begin{bmatrix} 25 & 18 \\ 3 & 9 \end{bmatrix}\begin{bmatrix} 15 & 2 \\ 18 & 7 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 699 & 176 \\ 207 & 69 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix}$$

ดังนั้นได้ว่าค่ากุญแจลับสำหรับรหัสฮิลล์ คือ $\begin{bmatrix} 23 & 20 \\ 25 & 17 \end{bmatrix}$

6. รหัส One Time Pad

รหัส One Time Pad (OTP) [3] เป็นวิทยาการรหัสลับแบบสมมาตรที่มีกระบวนการเข้ารหัสและถอดรหัสที่เรียบง่ายเนื่องจากเป็นการดำเนินการที่ผ่านเพียงตัวดำเนินการแบบเอ็กคลูซีฟออร์หรือดำเนินการแบบใช้วิธีของซีซาร์ แต่มีความปลอดภัยสูงมาก เนื่องจากกุญแจลับที่ใช้สำหรับกระบวนการเข้ารหัสและถอดรหัสถูกใช้เพียงครั้งเดียวเท่านั้น ซึ่งหากจำเป็นต้องมีกระบวนการเข้ารหัสและถอดรหัสในครั้งถัดไปจะต้องใช้กุญแจลับค่าใหม่เสมอ นอกเหนือจากนั้น OTP คือรหัสลับ

ที่ได้รับความยอมรับว่าเป็นความลับสมบูรณ์ (Perfect Secretly) โดยรหัส OTP ยังมีการใช้งานอยู่ในปัจจุบัน

การเข้ารหัสแบบ OTP สามารถดำเนินการได้ 2 วิธี คือการเข้ารหัสแบบบัสต่อบัส และการเข้ารหัสแบบระหว่างตัวอักษร สำหรับการเข้ารหัสแบบบัสต่อบัสจำเป็นต้องแปลงข้อความต้นฉบับและค่ากุญแจลับให้อยู่ในรูปแบบเลขฐานสองแล้วจึงดำเนินการเข้ารหัสแบบตำแหน่งต่อตำแหน่งผ่านตัวดำเนินการเอ็กคลูซีฟออร์ ผลลัพธ์ที่ได้คือข้อความไซเฟอร์ที่อยู่ในรูปแบบเลขฐานสองเช่นเดียวกัน ในทางกลับการกระบวนการถอดรหัสสามารถดำเนินการถอดรหัสแบบตำแหน่งต่อตำแหน่งระหว่างข้อความไซเฟอร์ และกุญแจลับ

ตารางที่ 2.5 ผลการดำเนินการแบบบิตผ่านตัวดำเนินการเอ็กคลูซีฟออร์

อินพุต		**เอาต์พุต**
A	**B**	**$Z = A \oplus B$**
0	0	0
0	1	1
1	0	1
1	1	0

ตารางที่ 2.5 แสดงผลลัพธ์การดำเนินการแบบบิตระหว่าง 2 อินพุตผ่านตัวดำเนินการเอ็กคลูซีฟออร์ แบ่งออกเป็น 4 กรณีดังนี้

กรณีที่ 1: A = 0, B = 0 ได้ผลลัพธ์ Z = 0
กรณีที่ 2: A = 0, B = 1 ได้ผลลัพธ์ Z = 1
กรณีที่ 3: A = 1, B = 0 ได้ผลลัพธ์ Z = 1
กรณีที่ 4: A = 1, B = 1 ได้ผลลัพธ์ Z = 0

สำหรับสมการเข้ารหัสแบบบิตสำหรับรหัส OTP เป็นดังนี้

$$c_i = m_i \oplus k_i \tag{2.8}$$

และสมการถอดรหัสเป็นดังนี้

$$m_i = c_i \oplus k_i \tag{2.9}$$

เมื่อ m_i คือ ข้อความต้นฉบับแบบบิตตำแหน่งที่ i

c_i คือ ข้อความไซเฟอร์แบบบิตตำแหน่งที่ i

k_i คือ ค่ากุญแจลับตำแหน่งที่ i

ตัวอย่างที่ 2.11 กำหนดให้ข้อความต้นฉบับ m = 1001 และ กุญแจลับ k = 1100 จงแสดงกระบวนการเข้ารหัสด้วยรหัส OTP

วิธีทำ เนื่องจากเป็นการดำเนินการแบบบิต จากโจทย์จำนวนบิตของข้อความต้นฉบับ และกุญแจลับมีขนาดที่เท่ากันคือ 4 บิต ดังนั้นกระบวนการเข้ารหัสแบบบิตด้วยรหัส OTP เป็นดังนี้

ตำแหน่ง	**3**	**2**	**1**	**0**
ข้อความต้นฉบับ	1	0	0	1
กุญแจลับ	1	1	0	0
ผลลัพธ์ (ข้อความไซเฟอร์)	0	1	0	1

จากตารางแสดงผลลัพธ์จากการเข้ารหัสด้วยรหัส OTP แบบบัสต่อบัสโดยจากสมการ (2.8) อธิบายผลลัพธ์แต่ละตำแหน่งดังนี้

ตำแหน่งที่ 0 (m_0 = 1, k_0 = 0): $c_0 = m_0 \oplus k_0$

$= 1 \oplus 0$

$= 1$

ตำแหน่งที่ 1 (m_1 = 0, k_1 = 0): $c_1 = m_1 \oplus k_1$

$= 0 \oplus 0$

$= 0$

ตำแหน่งที่ 2 (m_2 = 0, k_2 = 1): $c_2 = m_2 \oplus k_2$

$= 0 \oplus 1$

$= 1$

ตำแหน่งที่ 3 ($m_3 = 1, k_3 = 1$): $c_3 = m_3 \oplus k_3$

$= 1 \oplus 1$

$= 0$

จากผลลัพธ์ที่ได้แต่ละตำแหน่งจึงได้ข้อความไซเฟอร์ซึ่งเกิดจากการเรียงจากตำแหน่งสูงที่สุดไปยังตำแหน่งต่ำที่สุด (ตำแหน่งที่ 3 ไปยังตำแหน่งที่ 0) คือ 0101

อย่างไรก็ตามข้อเสียของการเข้ารหัสแบบบัสต่อบัสคือขนาดความยาวของข้อความและกุญแจลับที่นำมาใช้สำหรับกระบวนการเข้ารหัสและถอดรหัสมีค่าที่สูงมาก และต้องผ่านกระบวนการแปลงค่าไป-กลับระหว่างเลขฐานสองและตัวอักษร ดังนั้นการเข้ารหัสระหว่างตัวอักษรจึงได้รับความนิยมมากกว่า สำหรับกระบวนการเข้ารหัสและถอดรหัสจะใช้รหัสซีซาร์ โดยตัวอักษรในแต่ละตำแหน่งของกุญแจลับจะถูกนำมาใช้เข้ารหัส และถอดรหัสตัวอักษรในตำแหน่งที่ตรงกันกับข้อความต้นฉบับ หรือข้อความไซเฟอร์ ดังนั้นจำนวนตัวอักษรของข้อความและกุญแจจะต้องมีค่าเท่ากัน อย่างไรก็ตามหากจำนวนตัวอักษรไม่เท่ากัน จำเป็นต้องทำให้ตัวอักษรเท่ากันก่อนโดยวิธีเพิ่มหรือลดตัวอักษร

ตัวอย่างที่ 2.12 กำหนดให้ข้อความต้นฉบับ EASY และ กุญแจลับ TSWO จงแสดงกระบวนการเข้ารหัสด้วยรหัส OTP

วิธีทำ เนื่องจากเป็นการดำเนินการระหว่างตัวอักษร จากโจทย์จำนวนข้อความต้นฉบับ และกุญแจลับมีขนาดที่เท่ากันคือ 4 ตัวอักษร ดังนั้นจากสมการ (2.1) ได้ผลลัพธ์เป็นดังนี้

ตำแหน่ง	**3**	**2**	**1**	**0**
ข้อความต้นฉบับ	E	A	S	Y
กุญแจลับ	T	S	W	O
ผลลัพธ์ (ข้อความไซเฟอร์)	X	S	O	M

จากตารางแสดงผลลัพธ์จากการเข้ารหัสด้วยรหัส OTP แบบตัวอักษร ซึ่งอธิบายผลลัพธ์แต่ละตำแหน่งดังนี้

ตำแหน่งที่ 0 (m_0 = Y, k_0 = O): $c_0 = (m_0 + k_0) \bmod 26$
$= (24 + 14) \bmod 26$
$= 12 = M$

ตำแหน่งที่ 1 (m_1 = S, k_1 = W): $c_1 = (m_1 + k_1) \bmod 26$
$= (18 + 22) \bmod 26$
$= 14 = O$

ตำแหน่งที่ 2 (m_2 = A, k_2 = S): $c_2 = (m_2 + k_2) \bmod 26$
$= (0 + 18) \bmod 26$
$= 18 = S$

ตำแหน่งที่ 3 (m_3 = E, k_3 = T): $c_3 = (m_3 + k_3) \bmod 26$
$= (23 + 19) \bmod 26$
$= 23 = X$

จากผลลัพธ์ที่ได้แต่ละตำแหน่งจึงได้ข้อความไซเฟอร์ซึ่งเกิดจากการเรียงจากตำแหน่งสูงที่สุดไปยังตำแหน่งต่ำที่สุด (ตำแหน่งที่ 3 ไปยังตำแหน่งที่ 0) คือ XSOM

รหัสลับที่กล่าวมาทั้งหมดก่อนหน้านี้ใช้หลักการเข้ารหัสและการถอดรหัสด้วยวิธีการแทนที่ตัวอักษรหนึ่งด้วยตัวอักษรอีกตัวหนึ่งซึ่งมีค่าที่แตกต่าง เรียกรหัสลับลักษณะนี้ว่ารหัสลับแบบแทนที่ (Substitution Cipher) อย่างไรก็ตามยังมีรหัสลับอีกประเภทหนึ่งที่ใช้หลักการเข้ารหัสและถอดรหัสโดยการสลับตำแหน่งของตัวอักษรเรียกว่า รหัสลับแบบสลับตำแหน่ง (Transposition Cipher) สำหรับบทนี้จะแสดงรหัสลับแบบสลับตำแหน่งจำนวน 3 ประเภทประกอบด้วย รหัสแบบแนวรั้ว (Rail Fence Cipher) รหัสแบบสลับคอลัมน์ (Column Transposition Cipher) และรหัสแบบสับเปลี่ยน (Permutation Cipher)

7. รหัสแบบแนวรั้ว (Rail Fence Cipher)

หลักการของรหัสแบบแนวรั้ว [53] คือการนำข้อความต้นฉบับมาเขียนใหม่โดยการเรียงตัวอักษรแต่ละตัวอักษรลงมาตามแนวแถว และเขียนย้อนกลับขึ้นไปหลังจากเขียนตัวอักษรมาถึงแถวสุดท้าย และเขียนตัวอักษรตัวถัดไปลงมาตามแนวแถวเช่นเดิมหากตัวอักษรปัจจุบันอยู่แถวบนสุดดำเนินการซ้ำเดิมจนกระทั่งสิ้นสุดตัวอักษรตัวสุดท้าย โดยข้อความไซเฟอร์เกิดจากการอ่านตัวอักษรตามแนวคอลัมน์โดยเริ่มจากแถวที่ 1 ไปจนถึงตัวอักษรสุดท้ายซึ่งอยู่แถวสุดท้าย และค่ากุญแจลับคือจำนวนแถว

ตัวอย่างที่ 2.13 กำหนดให้ข้อความต้นฉบับคือ CRYPTOGRAPHY IS FUN จงหาข้อความไซเฟอร์โดยใช้รหัสแบบแนวรั้ว กำหนดให้กุญแจลับมีค่าเป็น 3

วิธีทำ ดำเนินการนำตัวอักษรแต่ละตัวมาเรียงขึ้นลงตามแถว (ตัดช่องว่างออกทั้งหมด) โดยเริ่มจากแถวที่ 1 ได้ดังนี้

1. นำตัวอักษรตัวที่ 1 “C” เขียนลงแถวที่ 1

แถวที่ 1	**C**																
แถวที่ 2																	
แถวที่ 3																	

2. นำตัวอักษรตัวที่ 2 “R” เขียนลงแถวที่ 2

แถวที่ 1	C																
แถวที่ 2		**R**															
แถวที่ 3																	

3. นำตัวอักษรตัวที่ 3 “Y” เขียนลงแถวที่ 3

แถวที่ 1	C																
แถวที่ 2		R															
แถวที่ 3			**Y**														

4. เนื่องจากตัวอักษรถูกเขียนครบ 3 แถวแล้ว ดังนั้นตัวอักษรตัวที่ 4 “P” จึงถูกใส่ย้อนกลับขึ้นไปยังแถวที่ 2

แถวที่ 1	C																
แถวที่ 2		R		**P**													
แถวที่ 3			Y														

5. ดำเนินการเขียนตัวอักษรที่เหลือให้ครบทั้งหมด ได้ตารางที่สมบูรณ์เป็น ดังนี้

แถวที่ 1	C				T				A				I				N
แถวที่ 2		R		P		O		R		P		Y		S		U	
แถวที่ 3			Y				G				H				F		

หลังจากตัวอักษรทั้งหมดถูกเขียนลงตารางเรียบร้อยแล้ว ข้อความไซเฟอร์สามารถพิจารณาได้โดยการอ่านเรียงตามแนวคอลัมน์ โดยเริ่มจากแถวที่ 1 ไปจนกระทั่งถึงแถวสุดท้าย โดยข้อความไซเฟอร์ที่ได้ในแต่ละแถวเป็นดังนี้

แถวที่ 1: CTAIN

แถวที่ 2: RPORPYSU

แถวที่ 3: YGHF

โดยข้อความไซเฟอร์เกิดจากการผลลัพธ์ของ แถวที่ 1+ แถวที่ 2 + แถวที่ 3 ซึ่งมีค่าเป็นดังนี้ CTAINRPORPYSUYGHF

สำหรับการถอดรหัสสามารถดำเนินการได้โดยนำข้อความไซเฟอร์มาเรียงตามแนวคอลัมน์ โดยเริ่มจากแนวคอลัมน์ของแถวที่ 1 จนกระทั่งแถวสุดท้าย

8. รหัสแบบสลับคอลัมน์ (Column Transposition Cipher)

รหัสลับแบบสลับคอลัมน์ [57] คือการกำหนดจำนวนคอลัมน์ตามค่าของกุญแจลับ และนำข้อความที่จะนำมาเข้ารหัสมาวางเรียงตามแนวคอลัมน์ๆ ละ 1 ตัวอักษร และขึ้นแถวใหม่เมื่อสิ้นสุดคอลัมน์สุดท้าย หลังจากการนำตัวอักษรแต่ละตัวมาวางเรียงตามแนวคอลัมน์เสร็จเรียบร้อย จะทำการสลับตำแหน่งหมายเลขของคอลัมน์ซึ่งคือค่ากุญแจลับ ขั้นตอนสุดท้ายคือการอ่านข้อความไซเฟอร์ โดยจะเริ่มอ่านจากคอลัมน์ที่อยู่ตำแหน่งที่ 1 (ที่เกิดจากการสลับตำแหน่งเรียบร้อยแล้ว) เรียงลงมาตามแนวแถว เมื่อสิ้นสุดแถวสุดท้ายจะดำเนินการอ่านข้อความลงมาตามแถวซึ่งอยู่คอลัมน์ตำแหน่งถัดไป ดำเนินการลักษณะเดิมจนกระทั่งสิ้นสุดคอลัมน์สุดท้าย

ตัวอย่างที่ 2.14 กำหนดให้ข้อความต้นฉบับคือ IT IS VERY IMPORTANT TO LEARN CRYPTO GRAPHY จงหาข้อความไซเฟอร์โดยใช้รหัสแบบสลับคอลัมน์ กำหนดให้กุญแจลับมีค่าเป็น 3 4 2 1
วิธีทำ เนื่องจากโจทย์กำหนดกุญแจลับคือ 3 4 2 1 ดังนั้นจึงมีจำนวนคอลัมน์ทั้งหมด 4 คอลัมน์ การเข้ารหัสเริ่มจากการดำเนินการนำตัวอักษรแต่ละตัวมาเรียงตามแนวคอลัมน์ (ตัดช่องว่างออกทั้งหมด) โดยเริ่มจากคอลัมน์ที่ 1 ได้ดังนี้

1. นำตัวอักษรตัวที่ 1 “I” เขียนลงคอลัมน์ที่ 1 ของแถวที่ 1

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**
1	**I**			

2. นำตัวอักษรตัวที่ 2 “T” เขียนลงคอลัมน์ที่ 2 ของแถวที่ 1

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**
1	I	**T**		

3. นำตัวอักษรตัวที่ 3 “I” เขียนลงคอลัมน์ที่ 3 ของแถวที่ 1

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**
1	I	T	**I**	

4. นำตัวอักษรตัวที่ 4 “S” เขียนลงคอลัมน์ที่ 4 ของแถวที่ 1

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**

1	I	T	I	**S**

5. เนื่องจากตัวอักษรถูกเขียนลงครบทั้ง 4 คอลัมน์ ดังนั้นตัวอักษรตัวถัดไป (ตัวอักษรที่ 5) “V” จึงเริ่มกลับมาเขียนที่คอลัมน์ที่ 1 ของแถวถัดไป (แถวที่ 2)

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**
1	I	T	I	S
2	**V**			

6. ดำเนินการเขียนตัวอักษรที่เหลือให้ครบทั้งหมด ได้ตารางที่สมบูรณ์เป็น ดังนี้

แถวที่	**ตำแหน่งคอลัมน์**			
	1	**2**	**3**	**4**
1	I	T	I	S
2	V	E	R	Y
3	I	M	P	O
4	R	T	A	N
5	T	T	O	L
6	E	A	R	N
7	C	R	Y	P
8	T	O	G	R
9	A	P	H	Y

หลังจากตัวอักษรทั้งหมดถูกเขียนลงตารางเรียบร้อยแล้ว สลับคอลัมน์ตามค่าของกุญแจลับ ดังนี้

แถวที่	ตำแหน่งคอลัมน์			
	3	4	2	1
1	I	T	I	S
2	V	E	R	Y
3	I	M	P	O
4	R	T	A	N
5	T	T	O	L
6	E	A	R	N
7	C	R	Y	P
8	T	O	G	R
9	A	P	H	Y

หลังจากตัวอักษรทั้งหมดถูกเขียนลงตารางเรียบร้อยแล้ว ข้อความไซเฟอร์สามารถพิจารณาได้โดยการอ่านเรียงตามแนวแถว โดยเริ่มจากคอลัมน์ตำแหน่งที่ 1 ไปจนกระทั่งถึงคอลัมน์ตำแหน่งสุดท้าย โดยข้อความไซเฟอร์ที่ได้ในแต่ละคอลัมน์เป็นดังนี้

คอลัมน์ตำแหน่งที่ 1: SYONLNPRY

คอลัมน์ตำแหน่งที่ 2: IRPAORYGH

คอลัมน์ตำแหน่งที่ 3: IVIRTECTA

คอลัมน์ตำแหน่งที่ 4: TEMTTAROP

โดยข้อความไซเฟอร์เกิดจาก คอลัมน์ตำแหน่งที่ 1+ คอลัมน์ตำแหน่งที่ 2 + คอลัมน์ตำแหน่งที่ 3 + คอลัมน์ตำแหน่งที่ 4 ได้เป็น SYONLNPRYIRPAORYGHIVIRTECTATEMTTAROP

สำหรับการถอดรหัสสามารถดำเนินการได้โดยนำข้อความไซเฟอร์มาเรียงตามแนวแถวของแต่ละคอลัมน์ตามค่าของกุญแจลับ

9. รหัสแบบสับเปลี่ยน (Permutation Cipher)

รหัสแบบสับเปลี่ยน [57] คือการแบ่งข้อความต้นฉบับออกเป็นกลุ่มโดยแต่ละกลุ่มจะมีจำนวนอักขระที่เท่ากันและใช้ตารางการสับเปลี่ยนตำแหน่งสำหรับเปลี่ยนตำแหน่งอักขระโดยจำนวนตำแหน่งขึ้นอยู่กับจำนวนอักขระในแต่ละกลุ่มซึ่งผู้ใช้งานสามารถเลือกขนาดได้อย่างอิสระ

ตารางที่ 2.6 ตัวอย่างตารางสับเปลี่ยนสำหรับอักขระ 6 ตัว

ตำแหน่งต้นทาง	1	2	3	4	5	6
ตำแหน่งปลายทาง	3	5	4	1	6	2

ตารางที่ 2.6 แสดงตัวอย่างการสับเปลี่ยนตำแหน่งของอักขระจำนวน 6 ตัวต่อ 1 กลุ่มโดยแถวที่ 1 แสดงตำแหน่งต้นทางหรือตำแหน่งเริ่มต้นของข้อความต้นฉบับที่อยู่ในกลุ่มที่ถูกแบ่ง และแถวที่ 2 แทนตำแหน่งการเคลื่อนย้ายของอักขระ เช่น อักขระตำแหน่งที่ 1 จะถูกสับเปลี่ยนไปอยู่ตำแหน่งที่ 3 และ อักขระตำแหน่งที่ 3 จะถูกสับเปลี่ยนไปอยู่ในตำแหน่งที่ 4 เป็นต้น หลังจากเสร็จสิ้นการสับเปลี่ยนในแต่ละกลุ่มแล้วจะนำข้อความไซเฟอร์ที่ได้มาเชื่อมต่อเป็นข้อความเดียว

สำหรับกระบวนการถอดรหัสจะใช้ตารางสับเปลี่ยนผกผันของตารางที่ใช้สำหรับกระบวนการเข้ารหัสโดยตำแหน่งปลายทางของตารางสับเปลี่ยนจะถูกเปลี่ยนเป็นตำแหน่งต้นทางของตารางสับเปลี่ยนผกผัน และพิจารณาตำแหน่งปลายทางจากความสัมพันธ์ของแต่ละคู่ของตารางสับเปลี่ยน

ตารางที่ 2.7 ตัวอย่างตารางสับเปลี่ยนผกผันของตารางที่ 2.6 สำหรับอักขระ 6 ตัว

ตำแหน่งต้นทาง	1	2	3	4	5	6
ตำแหน่งปลายทาง	4	6	1	3	2	5

ตารางที่ 2.7 แสดงตารางสับเปลี่ยนผกผันของตารางที่ 2.6 ซึ่งสามารถสร้างได้โดยนำตำแหน่งปลายทางของตารางสับเปลี่ยนมาเป็นตำแหน่งต้นทางของตารางสับเปลี่ยนผกผันโดยเรียงลำดับจากน้อยไปหามาก และตำแหน่งปลายทางของตารางสับเปลี่ยนผกผันสามารถหาได้โดยพิจารณาตำแหน่งต้นทางของตารางสับเปลี่ยนที่สัมพันธ์กัน เช่น อักขระตำแหน่งที่ 1 จะถูกสับเปลี่ยนไปอยู่ตำแหน่งที่ 4 เป็นต้น

ตัวอย่างที่ 2.15 กำหนดให้ข้อความต้นฉบับคือ PERMUTATION CIPHER IS ONE OF CRYPTO GRAPHY จงหาข้อความไซเฟอร์โดยใช้รหัสแบบสับเปลี่ยน เลือกขนาดอักขระเป็น 6 และใช้กุญแจดังตารางที่ 2.6 สำหรับการเข้ารหัสลับ

วิธีทำ จากวิธีการของรหัสแบบสับเปลี่ยน มีขั้นตอนเป็นดังนี้

1. นำข้อความต้นฉบับมาแบ่งเป็นกลุ่มๆ ละ 6 ตัวได้ดังนี้

PERMUT	ATIONC	IPHERI	SONEOF	CRYPTO	GRAPHY
กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4	กลุ่ม 5	กลุ่ม 6

2. นำข้อความต้นฉบับแต่ละกลุ่มมาสับเปลี่ยนตำแหน่งโดยใช้ตารางที่ 2.6 โดยในขั้นตอนนี้จะอธิบายเฉพาะการสับเปลี่ยนเฉพาะกลุ่มที่ 1 ดังนี้

ตำแหน่งต้นทาง 1 (P) จะถูกย้ายไปอยู่ตำแหน่งที่ 3 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 2 (E) จะถูกย้ายไปอยู่ตำแหน่งที่ 5 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 3 (R) จะถูกย้ายไปอยู่ตำแหน่งที่ 4 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 4 (M) จะถูกย้ายไปอยู่ตำแหน่งที่ 1 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 5 (U) จะถูกย้ายไปอยู่ตำแหน่งที่ 6 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 6 (T) จะถูกย้ายไปอยู่ตำแหน่งที่ 2 ในตำแหน่งปลายทาง

ดังนั้นได้ข้อความไซเฟอร์แต่ละตำแหน่งของกลุ่มที่ 1 เป็นดังนี้

ตำแหน่ง	1	2	3	4	5	6
อักขระ	M	T	P	R	E	U

3. ดำเนินการลักษณะเช่นเดียวกันกับขั้นตอนที่ 2 สำหรับทุกกลุ่มที่เหลือได้ผลลัพธ์เป็นดังนี้

MTPREU	OCAITN	EIIHPR	EFSNOO	POCYRT	PYGARH
กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4	กลุ่ม 5	กลุ่ม 6

4. ขั้นตอนสุดท้ายคือการนำข้อความไซเฟอร์ที่ได้ในแต่ละกลุ่มมาต่อเชื่อมกันได้เป็นดังนี้

MTPREUOCAITNEIIHPREFSNOOPOCYRTPYGARH

ตัวอย่างที่ 2.16 จากข้อความไซเฟอร์ตัวอย่างที่ 2.15 จงหาข้อความต้นฉบับโดยใช้ตารางสับเปลี่ยนผกผันดังตารางที่ 2.7 สำหรับการถอดรหัส

วิธีทำ จากวิธีการของรหัสแบบสับเปลี่ยน มีขั้นตอนเป็นดังนี้

1. นำข้อความไซเฟอร์มาแบ่งเป็นกลุ่มๆ ละ 6 ตัวได้ดังนี้

MTPREU	OCAITN	EIIHPR	EFSNOO	POCYRT	PYGARH
กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4	กลุ่ม 5	กลุ่ม 6

2. นำข้อความไซเฟอร์แต่ละกลุ่มมาสับเปลี่ยนตำแหน่งโดยใช้ตารางที่ 2.7 โดยในขั้นตอนนี้จะอธิบายเฉพาะการสับเปลี่ยนเฉพาะกลุ่มที่ 1 ดังนี้

ตำแหน่งต้นทาง 1 (M) จะถูกย้ายไปอยู่ตำแหน่งที่ 4 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 2 (T) จะถูกย้ายไปอยู่ตำแหน่งที่ 6 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 3 (P) จะถูกย้ายไปอยู่ตำแหน่งที่ 1 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 4 (R) จะถูกย้ายไปอยู่ตำแหน่งที่ 3 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 5 (E) จะถูกย้ายไปอยู่ตำแหน่งที่ 2 ในตำแหน่งปลายทาง
ตำแหน่งต้นทาง 6 (U) จะถูกย้ายไปอยู่ตำแหน่งที่ 5 ในตำแหน่งปลายทาง

ดังนั้นได้ข้อความไซเฟอร์แต่ละตำแหน่งของกลุ่มที่ 1 เป็นดังนี้

ตำแหน่ง	1	2	3	4	5	6
อักขระ	P	E	R	M	U	T

3. ดำเนินการลักษณะเช่นเดียวกันกับขั้นตอนที่ 2 สำหรับทุกกลุ่มที่เหลือได้ผลลัพธ์เป็นดังนี้

PERMUT	ATIONC	IPHERI	SONEOF	CRYPTO	GRAPHY
กลุ่ม 1	กลุ่ม 2	กลุ่ม 3	กลุ่ม 4	กลุ่ม 5	กลุ่ม 6

4. ขั้นตอนสุดท้ายคือการนำข้อความต้นฉบับที่ได้ในแต่ละกลุ่มมาต่อเชื่อมกันได้เป็นดังนี้

PERMUTATIONCIPHERISONEOFCRYPTOGRAPHY

10. บทสรุปสาระสำคัญ

ในบทนี้ได้อธิบายขั้นตอนวิธีในกลุ่มวิทยาการรหัสลับแบบสมมาตรซึ่งได้เลือกเฉพาะขั้นตอนวิธีที่มีความเรียบง่ายไม่ซับซ้อนเพื่อให้ผู้อ่านมีความเข้าใจเกี่ยวกับหลักการของวิทยาการรหัสลับได้โดยง่าย โดยวิทยาการรหัสลับแบบสมมาตรถูกแบ่งออกเป็นสองประเภทคือรหัสลับแบบสลับตำแหน่งคือตัวอักษรทั้งหมดที่อยู่ในข้อความต้นฉบับจะถูกสลับตำแหน่งตามคุณลักษณะของขั้นตอนวิธีที่แตกต่างกัน เช่นรหัสแบบแนวรั้ว รหัสแบบสลับคอลัมน์ และรหัสแบบสับเปลี่ยน วิทยาการรหัสลับแบบสมมาตรอีกประเภทหนึ่งคือรหัสลับแบบแทนที่ซึ่งตัวอักษรแต่ละตัวในข้อความต้นฉบับจะถูกแทนที่ด้วยตัวอักษรตัวอื่นๆ ซึ่งไม่จำเป็นต้องมีอยู่ในข้อความต้นฉบับดังนั้นตัวอักษรทั้งหมดในข้อความไซเฟอร์ที่ได้จากการเข้ารหัสลับด้วยขั้นตอนวิธีในกลุ่มรหัสแบบแทนที่อาจเป็นตัวอักษรที่ไม่เคยปรากฏอยู่ในข้อความต้นฉบับเลยก็เป็นได้

อย่างไรก็ตามวิทยาการรหัสลับแบบสมมาตรที่ได้กล่าวทั้งหมดในบทนี้ ยกเว้นรหัส OTP ไม่ถูกนำมาใช้งานแล้วในปัจจุบันเนื่องจากง่ายต่อการโจมตีโดยใช้เทคนิควิธีที่แตกต่างกันออกไป เช่นการโจมตีแบบตะลุย การวิเคราะห์ความถี่ของการเกิดตัวอักษร เป็นต้น โดยรหัส OTP เป็นเพียงวิทยาการรหัสลับชนิดเดียวในบทนี้ที่ยังคงถูกใช้งานอยู่ในปัจจุบัน ถึงแม้ว่าเป็นรหัสลับที่มีความเรียบง่าย แต่การโจมตีเกิดขึ้นได้ยากมากหรือแทบจะเป็นไปไม่ได้เนื่องจากกุญแจลับที่นำมาใช้งานจะถูกใช้งานเพียงครั้งเดียว และจะถูกสร้างใหม่ในทุกครั้งที่มีการใช้งานเกิดขึ้น

แบบฝึกหัดท้ายบท
บทที่ 2

1. จงเข้ารหัสข้อความ BABY ด้วยรหัสซีซาร์โดยกำหนดให้ใช้กุญแจลับที่มีค่าเป็น 11
2. ข้อความไซเฟอร์ที่กำหนดให้ต่อไปนี้ LJB เป็นข้อความที่ถูกเข้ารหัสด้วยรหัสซีซาร์ที่ใช้กุญแจลับมีค่าเป็น 9 จงคำนวณหาข้อความต้นฉบับ
3. จงหาข้อความต้นฉบับ และกุญแจลับ เมื่อกำหนดให้ข้อความไซเฟอร์ซึ่งถูกเข้ารหัสลับด้วยรหัสซีซาร์มีค่าเป็น VRJP
4. จงเข้ารหัสข้อความ EXPERT ด้วยรหัสสับเปลี่ยนโดยใช้กุญแจลับดังตารางที่ 2.3
5. ข้อความไซเฟอร์ที่กำหนดให้ต่อไปนี้ NBGT เป็นข้อความที่ถูกเข้ารหัสด้วยรหัสสับเปลี่ยนที่ใช้กุญแจลับดังตารางที่ 2.3 จงคำนวณหาข้อความต้นฉบับ
6. สามารถใช้ $a = 13$ เพื่อเป็นกุญแจลับสำหรับรหัสสัมพรรคได้หรือไม่เพราะเหตุใด
7. จงถอดรหัสข้อความไซเฟอร์ GADH ที่ถูกเข้ารหัสด้วยรหัสสัมพรรคที่มีกุญแจลับคือ $a = 7$ และ $b = 3$
8. กุญแจลับที่เป็นไปได้ทั้งหมดสำหรับรหัสสัมพรรคมีกี่ค่า
9. จงเข้ารหัสข้อความ TREE ด้วยรหัสวีเกเนร์โดยกำหนดให้ใช้กุญแจลับเป็น {2, 11, 3, 1}
10. จงเข้ารหัสข้อความ HE ด้วยรหัสฮิลล์โดยกำหนดให้ใช้กุญแจลับเป็น $k = \begin{bmatrix} 3 & 4 \\ 2 & 7 \end{bmatrix}$
11. จงเข้ารหัสข้อความ 110110 ด้วยรหัส OTP โดยกำหนดให้ใช้กุญแจลับเป็น 100011
12. จงเข้ารหัสข้อความ UDONTHANI RAJABHAT UNIVERSITY ด้วยรหัสแบบแนวรั้วกำหนดให้กุญแจลับคือ 5
13. จงหาข้อความต้นฉบับจากข้อความไซเฟอร์ที่กำหนดให้ต่อไปนี้ IEOLVMDGOY โดยข้อความดังกล่าวถูกเข้ารหัสด้วยรหัสแบบแนวรั้วและกุญแจลับมีค่าเท่ากับ 3
14. จงเข้ารหัสข้อความ UDONTHANI RAJABHAT UNIVERSITY ด้วยรหัสแบบสลับคอลัมน์กำหนดให้กุญแจลับคือ 2 3 1
15. จงหาข้อความต้นฉบับจากข้อความไซเฟอร์ที่กำหนดให้ต่อไปนี้ LFTLVYYIAOBLEDPYOAERA โดยข้อความดังกล่าวถูกเข้ารหัสด้วยรหัสแบบสลับคอลัมน์และกุญแจลับมีค่าเท่ากับ 2 3 1

16. จงเข้ารหัสข้อความ UDONTHANI RAJABHAT UNIVERSITY ด้วยรหัสแบบสับเปลี่ยนเมื่อกำหนดให้ตารางสับเปลี่ยนสำหรับอักขระ 3 ตัว เป็นดังนี้

ตำแหน่งต้นทาง	1	2	3
ตำแหน่งปลายทาง	2	3	1

17. จงหาข้อความต้นฉบับจากข้อความไซเฟอร์ที่กำหนดให้ต่อไปนี้ YCROPTAGRYPH โดยข้อความดังกล่าวถูกเข้ารหัสด้วยรหัสแบบสับเปลี่ยนและกุญแจลับมีค่าเดียวกับคำถามข้อ 16 (ต้องใช้ตารางสับเปลี่ยนผกผัน)

18. จงเข้ารหัสข้อความ HELLO ด้วยรหัส OTP โดยกำหนดให้ใช้กุญแจลับเป็น TXNRE

บทที่ 3
วิทยาการรหัสลับดีอีเอส

ในบทที่ 2 ได้กล่าวถึงวิทยาการรหัสลับแบบสมมาตรที่มีจุดเด่นคือใช้วิธีการคำนวณที่ไม่ยุ่งยากซับซ้อน อย่างไรก็ตามข้อเสียคือง่ายต่อการโจมตี สำหรับเนื้อหาในบทนี้จะกล่าวถึงขั้นตอนวิธีที่อยู่ในกลุ่มของวิทยาการรหัสลับแบบสมมาตรอีกวิธีหนึ่งที่มีความแข็งแกร่งมากยิ่งขึ้น

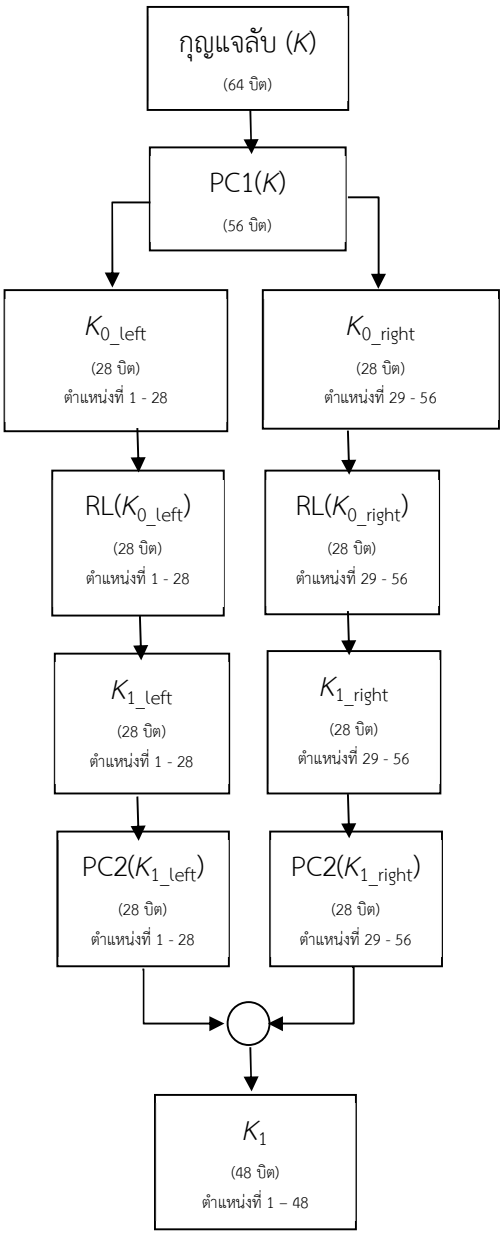
ดีอีเอส (Data Encryption Standard, DES) [28] ถูกพัฒนาครั้งแรกโดยสถาบันมาตรฐานและเทคโนโลยีแห่งสหรัฐอเมริกา (National Institute of Standards and Technology) ที่มีชื่อย่อคือ NIST ในปี ค.ศ. 1977 โดยได้รับการรับรองว่าเป็นวิทยาการรหัสลับแบบสมมาตรที่เป็นมาตรฐานและมีใช้งานอย่างแพร่หลายในประเทศสหรัฐอเมริกาในช่วงเวลานั้น อย่างไรก็ตามระบบดีอีเอสได้ถูกโจมตีลงแล้วจึงถูกยกเลิกการใช้งานในปัจจุบัน

สำหรับกระบวนการเข้ารหัส และถอดรหัสข้อมูลจะถูกแบ่งออกเป็น 16 รอบการทำงานโดยข้อความต้นฉบับ และกุญแจลับที่จะถูกนำมาใช้งานจะมีขนาด 64 บิต อย่างไรก็ตามกุญแจลับจะถูกเลือกนำมาเข้าสู่กระบวนการเพียง 56 บิต ในขณะที่อีก 8 บิตที่เหลือจะถูกนำมาใช้สำหรับการตรวจสอบพาริตี้ โดยรหัสดีอีเอสถูกแบ่งออกเป็น 3 กระบวนการหลักประกอบไปด้วยการจัดการกุญแจลับ กระบวนการเข้ารหัสลับ และกระบวนการถอดรหัสลับ อย่างไรก็ตามกระบวนการถอดรหัสลับจะมีรูปแบบการดำเนินการที่คล้ายคลึงกับกระบวนการเข้ารหัสลับ และใช้กุญแจลับชุดเดียวกันเพียงแต่ลำดับการทำงานจะแตกต่างกัน ซึ่งจะเริ่มจากรอบที่ 16 และย้อนกลับไปจนกระทั่งถึงรอบที่ 1 จึงจะได้ข้อความต้นฉบับ

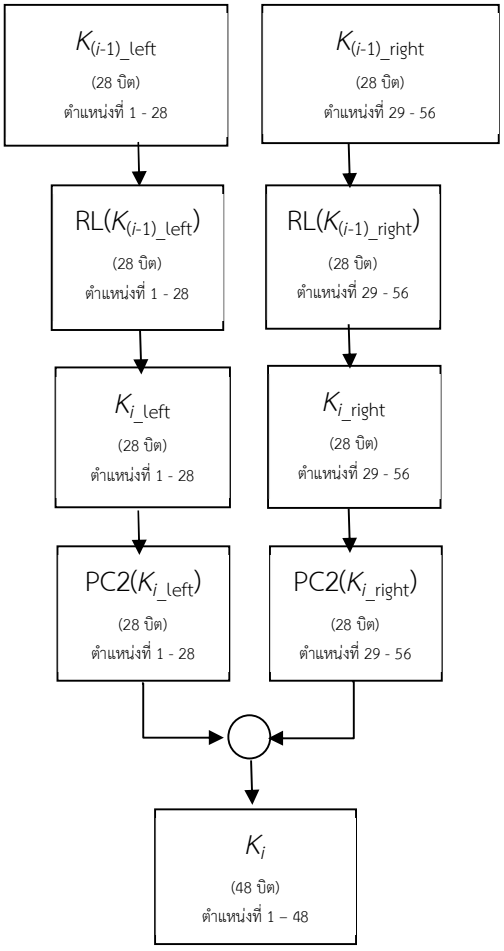
1. การจัดการกุญแจลับดีอีเอส

กุญแจลับที่ถูกนำมาใช้งานสำหรับรหัสดีอีเอสจะมีขนาด 64 บิต โดยจะถูกเลือกเข้าสู่กระบวนการเพียง 56 บิต เนื่องจากกระบวนการเข้ารหัสและถอดรหัสจะถูกดำเนินการทั้งหมด 16 รอบ ดังนั้นจึงมีการจัดการกุญแจลับเพื่อปรับเปลี่ยนค่าสำหรับแต่ละรอบการคำนวณทั้งหมด 16 ครั้ง

ขั้นตอนการจัดการกุญแจในรอบที่ 1 จะแตกต่างจากรอบที่ 2 – 16 เนื่องจากในรอบที่ 1 จำเป็นต้องนำกุญแจลับมาผ่านกล่องสลับลำดับ PC1 (Permuted Choice 1) ดังตารางที่ 3.1 ซึ่งจะดำเนินการในรอบแรกเพียงครั้งเดียว ในทางกลับกันการจัดการกุญแจในรอบอื่นๆ จะไม่ถูกนำมาผ่านกล่องสลับลำดับ PC1



รูปที่ 3.1 การจัดการกุญแจลับรอบที่ 1



รูปที่ 3.2 การจัดการกุญแจรอบที่ 2 - 16

ตารางที่ 3.1 กล่องสลับลำดับ PC1

ตำแหน่งเดิม	ตำแหน่งใหม่ (K_{0_left})		ตำแหน่งเดิม	ตำแหน่งใหม่ (K_{0_right})
57	1		63	29
49	2		55	30
41	3		47	31
33	4		39	32
25	5		31	33
17	6		23	34
9	7		15	35
1	8		7	36
58	9		62	37
50	10		54	38
42	11		46	39
34	12		38	40
26	13		30	41
18	14		22	42
10	15		14	43
2	16		6	44
59	17		61	45
51	18		53	46
43	19		45	47
35	20		37	48
27	21		29	49
19	22		21	50
11	23		13	51
3	24		5	52
60	25		28	53
52	26		20	54
44	27		12	55
36	28		4	56

ตารางที่ 3.2 กล่องสลับลำดับ PC2

ตำแหน่งเดิม	ตำแหน่งใหม่ (K_{i_left})		ตำแหน่งเดิม	ตำแหน่งใหม่ (K_{i_right})
14	1		41	25
17	2		52	26
11	3		31	27
24	4		37	28
1	5		47	29
5	6		55	30
3	7		30	31
28	8		40	32
15	9		51	33
6	10		45	34
21	11		33	35
10	12		48	36
23	13		44	37
19	14		49	38
12	15		39	39
4	16		56	40
26	17		34	41
8	18		53	42
16	19		46	43
7	20		42	44
27	21		50	45
20	22		36	46
13	23		29	47
2	24		32	48

ตารางที่ 3.3 จำนวนบิตที่ถูกหมุนแบบวนซ้ายแต่ละรอบ

รอบที่	จำนวนบิต		รอบที่	จำนวนบิต
1	1		9	1
2	1		10	2
3	2		11	2
4	2		12	2
5	2		13	2
6	2		14	2
7	2		15	2
8	2		16	1

รูปที่ 3.1 แสดงการจัดการกุญแจลับรอบที่ 1 ซึ่งหลังจากนำกุญแจลับ (K) ที่มีขนาด 64 บิต มาผ่านกล่องสลับลำดับ PC1 แล้ว ค่ากุญแจจะถูกแบ่งออกเป็น 2 ส่วนๆ ละ 28 บิตคือ K_{0_left} เก็บค่าที่อยู่ระหว่างตำแหน่งที่ 1 – 28 และ K_{0_right} เก็บค่าที่อยู่ระหว่างตำแหน่งที่ 29 – 56 และนำทั้งสองค่าไปผ่านการหมุนวนซ้าย (Rotate Left, RL) ซึ่งจำนวนบิตการหมุนขึ้นอยู่กับแต่ละรอบของการดำเนินการดังตารางที่ 3.3 สำหรับรอบแรกจะหมุน 1 บิตโดยกำหนดผลลัพธ์เป็น K_{1_left} และ K_{1_right} และขั้นตอนสุดท้าย นำผลลัพธ์ที่ได้ทั้งสองค่าไปผ่านกล่องสลับลำดับ PC2 (Permuted Choice 2) ดังตารางที่ 3.2 เพื่อให้ได้ค่ากุญแจในรอบที่ 1 (K_1) ซึ่งถูกลดขนาดลงเหลือ 48 บิต

หมายเหตุ: กำหนดให้การนับตำแหน่งเริ่มจากตำแหน่งซ้ายสุด (บิตที่มีค่านัยสำคัญสูงสุด) ไปยังตำแหน่งขวาสุด (บิตทีมีค่านัยสำคัญน้อยที่สุด)

รูปที่ 3.2 แสดงการจัดการกุญแจลับรอบที่ 2-16 ซึ่งจะแตกต่างจากรอบแรกเพียงแค่ไม่มีขั้นตอนการนำข้อมูลเข้าสู่กล่องสลับลำดับ PC1 โดยหลักการดำเนินงานเริ่มจากการนำค่า $K_{(i-1)\_left}$ และ $K_{(i-1)\_right}$ ที่ได้จากรอบก่อนหน้ามาผ่านการหมุนวนซ้าย และนำผลลัพธ์ที่ได้ทั้งสองค่าไปผ่านกล่องสลับลำดับ PC2 เพื่อให้ได้ K_i (ค่ากุญแจในรอบที่ i) ที่ถูกลดขนาดลงเหลือ 48 บิต

ตัวอย่างที่ 3.1 กำหนดให้กุญแจลับ K = A29BDE783CF21139 จงหา K_1 ขนาด 48 บิตซึ่งได้จากกระบวนการจัดการกุญแจลับในรอบที่ 1

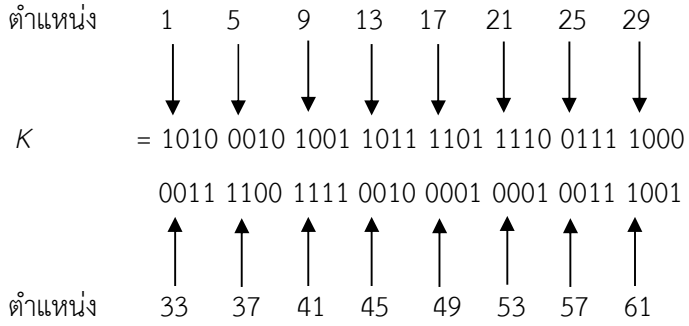
วิธีทำ จากโจทย์กำหนด K ที่อยู่ในรูปแบบเลขฐานสิบหกขนาด 64 บิต จึงสามารถแปลงให้อยู่ในรูปแบบเลขฐานสองได้ดังนี้

A 2 9 B D E 7 8

K = 1010 0010 1001 1011 1101 1110 0111 1000
0011 1100 1111 0010 0001 0001 0011 1001

3 C F 2 1 1 3 9

ซึ่งจากค่ากุญแจที่ได้กำหนดให้ได้ว่าตำแหน่งที่ 1 จะอยู่ทางซ้ายสุด และถูกเรียงไปจนถึงตำแหน่งสุดท้ายซึ่งคือตำแหน่งที่ 64 จึงได้แต่ละตำแหน่งของ K เป็นดังนี้



หลังจากทราบโครงสร้างของ K แล้วจะเริ่มเข้าสู่การจัดการกุญแจลับรอบที่ 1 ดังนี้

ขั้นตอนที่ 1 (ใช้กล่องสลับลำดับ PC1)

K_{0_left} = 0010 0111 0010 1100 1011 1001 1111

K_{0_right} = 0010 0111 0001 0100 1001 1110 1110

ขั้นตอนที่ 2 จากตารางที่ 3.3 จึงได้เป็นการหมุนซ้าย 1 บิต

K_{1_left} = 0100 1110 0101 1001 0111 0011 1110

K_{1_right} = 0100 1110 0010 1001 0011 1101 1100

ขั้นตอนที่ 3 (ใช้กล่องสลับลำดับ PC2)

K_1 = 0001 0100 0101 1110 1011 1111 1100 1010 0011 1110 1100 1000

ดังนั้นหลังเสร็จสิ้นกระบวนการจัดการกุญแจรอบที่ 1 ได้

K_1 = 0001 0100 0101 1110 1011 1111 1100 1010 0011 1110 1100 1000

ตัวอย่างที่ 3.2 จากกุญแจลับ และผลการจัดการกุญแจในรอบที่ 1 ดังตัวอย่างที่ 3.1 จงหากุญแจลับสำหรับรอบที่ 2 (K_2)

วิธีทำ จากตัวอย่างที่ 3.1

K_{1_left} = 0100 1110 0101 1001 0111 0011 1110

K_{1_right} = 0100 1110 0010 1001 0011 1101 1100

ดังนั้นการจัดการกุญแจเพื่อหา K_2 ตามรูปที่ 3.2 มีขั้นตอนเป็นดังนี้

ขั้นตอนที่ 1 จากตารางที่ 3.3 จึงได้เป็นการหมุนซ้าย 1 บิต

K_{2_left} = 1001 1100 1011 0010 1110 0111 1100

K_{2_right} = 1001 1100 0101 0010 0111 1011 1000

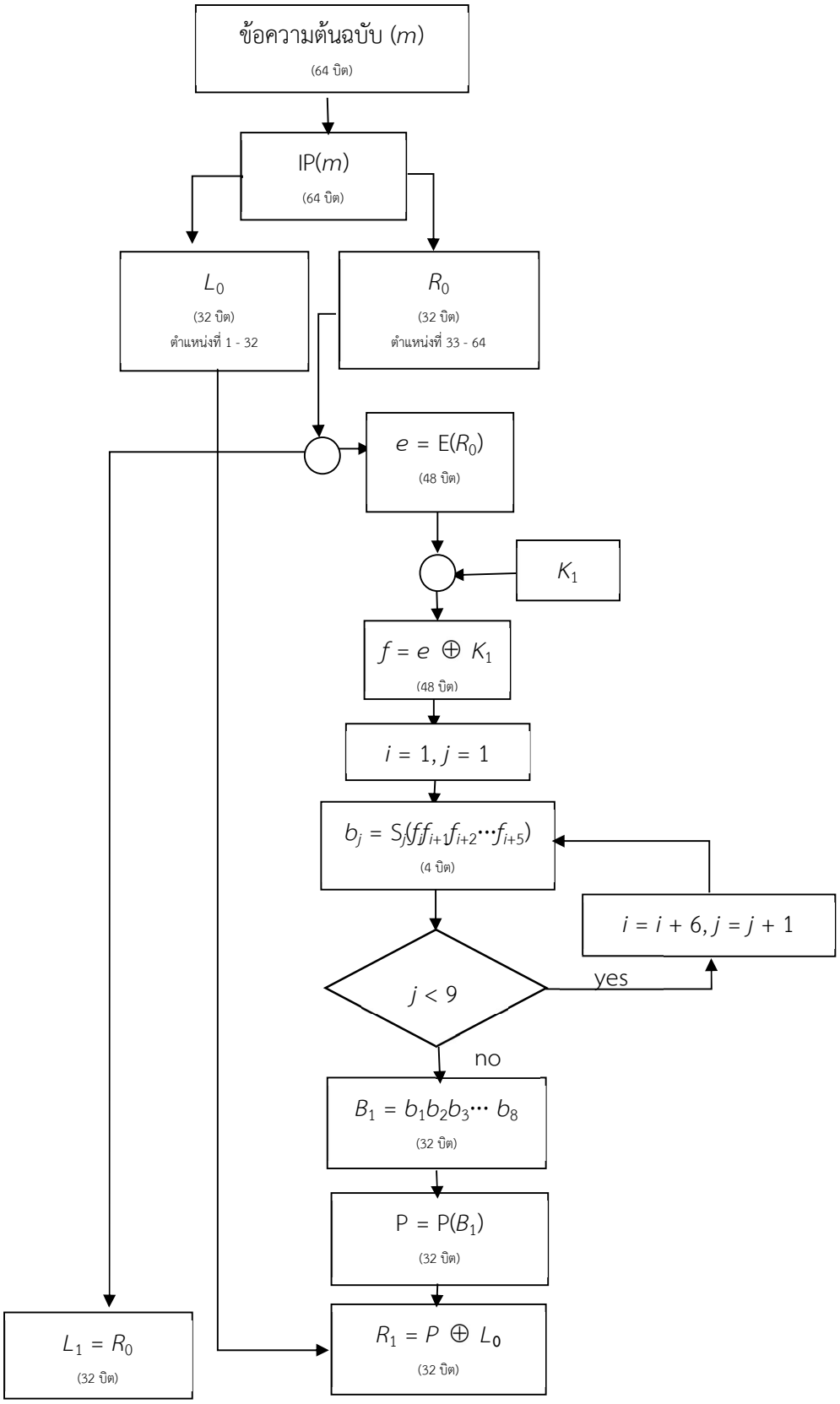
ขั้นตอนที่ 2 (ใช้กล่องสลับลำดับ PC2)

K_2 = 0111 1100 1100 1111 1000 0000 0100 1001 1011 0100 1110 0011

ดังนั้นหลังเสร็จสิ้นกระบวนการจัดการกุญแจรอบที่ 2 ได้

K_2 = 0111 1100 1100 1111 1000 0000 0100 1001 1011 0100 1110 0011

อย่างไรก็ตามการจัดการกุญแจในรอบที่ 3 – 16 สามารถดำเนินการได้เช่นเดียวกับรอบที่ 2 ซึ่งหลังจากดำเนินการครบทั้ง 16 รอบแล้ว ผู้สร้างกุญแจลับจะทราบค่า K_1, K_2, …, K_{16} เพื่อใช้สำหรับกระบวนการเข้ารหัส และกระบวนการถอดรหัสในแต่ละรอบ



รูปที่ 3.3 การเข้ารหัสลับดีอีเอสรอบแรก

ตารางที่ 3.4 กล่องสลับลำดับ IP

ตำแหน่งเดิม	ตำแหน่งใหม่ (L_0)		ตำแหน่งเดิม	ตำแหน่งใหม่ (R_0)
58	1		57	33
50	2		49	34
42	3		41	35
34	4		33	36
26	5		25	37
18	6		17	38
10	7		9	39
2	8		1	40
60	9		59	41
52	10		51	42
44	11		43	43
36	12		35	44
28	13		27	45
20	14		19	46
12	15		11	47
4	16		3	48
62	17		61	49
54	18		53	50
46	19		45	51
38	20		37	52
30	21		29	53
22	22		21	54
14	23		13	55
6	24		5	56
64	25		63	57
56	26		55	58
48	27		47	59
40	28		39	60
32	29		31	61
24	30		23	62
16	31		15	63
8	32		7	64

ตารางที่ 3.5 ฟังก์ชันขยายบิต (E)

ตำแหน่งขยาย	ตำแหน่งเดิม				ตำแหน่งขยาย
คอลัมน์ 1	**คอลัมน์ 2**	**คอลัมน์ 3**	**คอลัมน์ 4**	**คอลัมน์ 5**	**คอลัมน์ 6**
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

ตารางที่ 3.6 กล่องเอสที่ 1 (S_1)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถว	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

ตารางที่ 3.7 กล่องเอสที่ 2 (S_2)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

ตารางที่ 3.8 กล่องเอสที่ 3 (S_3)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

ตารางที่ 3.9 กล่องเอสที่ 4 (S_4)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

ตารางที่ 3.10 กล่องเอสที่ 5 (S_5)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

ตารางที่ 3.11 กล่องเอสที่ 6 (S_6)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

ตารางที่ 3.12 กล่องเอสที่ 7 (S_7)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถวที่	**0**	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

ตารางที่ 3.13 กล่องเอสที่ 8 (S_8)

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
แถว	**0**	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

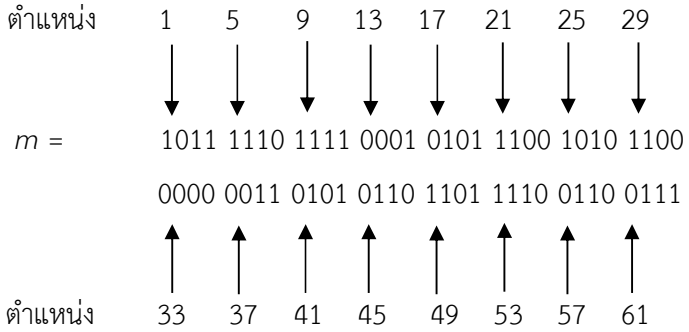
2. การเข้ารหัสลับดีอีเอส

การเข้ารหัสลับแบบดีอีเอสคือการนำข้อความต้นฉบับที่มีขนาด 64 บิตมาผ่านกระบวนการทั้งหมด 16 รอบ โดยที่แต่ละรอบจะใช้ค่ากุญแจลับขนาด 48 บิตที่มีความแตกต่างกันออกไปทั้งหมด 16 ค่าซึ่งได้กล่าวไว้แล้วในขั้นตอนการจัดการกุญแจลับ โดยการเข้ารหัสลับในรอบแรกจะมีความแตกต่างจากรอบที่ 2 – 16 เนื่องจากจำเป็นต้องนำข้อความต้นฉบับมาผ่านกล่องสลับลำดับ IP (Initial Permutation) ดังตารางที่ 3.4

รูปที่ 3.3 แสดงวิธีการเข้ารหัสดีอีเอสรอบแรกเพื่อหา L_1 และ R_1 โดยเริ่มจากนำข้อความต้นฉบับ (m) ที่มีขนาด 64 บิตมาผ่านกล่องสลับลำดับ IP และแบ่งออกเป็น 2 ส่วนๆ 32 บิตกำหนดเป็น L_0 (ตำแหน่งที่ 1 - 32) และ R_0 (ตำแหน่งที่ 33 – 64) และสามารถหา $L_1 = R_0$ ในทางกลับกันการคำนวณหา R_1 มีขั้นตอนต้องดำเนินการซึ่งจะกล่าวในลำดับถัดไป

ตัวอย่างที่ 3.3 กำหนดให้ m = 1011 1110 1111 0001 0101 1100 1010 1100 0000 0011 0101 0110 1101 1110 0110 0111 จงคำนวณหา L_0 และ R_0

วิธีทำ จากค่าที่กำหนดพิจารณาแต่ละตำแหน่งได้เป็นดังนี้

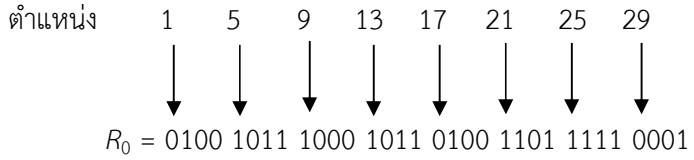


ใช้กล่องสลับลำดับ IP ได้ผลลัพธ์เป็นดังนี้

L_0 = 1110 0110 0110 0111 1110 1101 1001 0010
R_0 = 0100 1011 1000 1011 0100 1101 1111 0001

หลังจากทราบ R_0 ขั้นตอนต่อไปคือหา e โดยใช้ฟังก์ชันขยายบิต (e = E(R_0)) เพื่อขยายจำนวนบิตเป็น 48 บิต ดังตารางที่ 3.5 โดยหลักการคือนำค่า R_0 มาเรียงใหม่เป็น 8 แถวๆ ละ 4 คอลัมน์ โดยแต่ละแถวกำหนดให้สมาชิกตัวแรกอยู่คอลัมน์ที่ 2 และ ตัวสุดท้ายอยู่คอลัมน์ที่ 5 และมีการเพิ่มบิตแถวละ 2 บิตไว้ที่ตำแหน่งคอลัมน์ 1 และ คอลัมน์ที่ 6 โดยค่าที่เพิ่มจะขึ้นอยู่กับตำแหน่งเดิมของ E หลังจากเสร็จสิ้นการเพิ่มบิตแล้วจะได้ผลลัพธ์สุดท้ายทั้งหมด 48 บิตซึ่งเกิดจากการนำข้อมูลแต่จะแถวมาเรียงกันโดยเริ่มจากแถวที่ 1 จนถึงแถวสุดท้าย

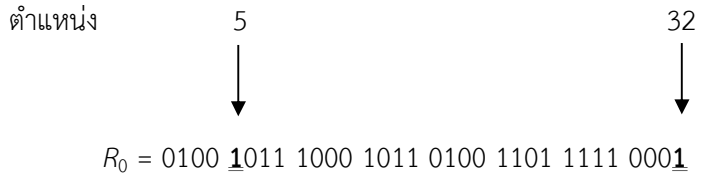
ตัวอย่างที่ 3.4 จาก R_0 ที่ได้จากตัวอย่างที่ 3.3 จงคำนวณหา e = E(R_0)
วิธีทำ จากค่าที่กำหนดพิจารณาแต่ละตำแหน่งได้เป็นดังนี้



นำค่าแต่ละตำแหน่งบรรจุลงตารางของฟังก์ชันขยายบิตได้ดังนี้

ตำแหน่งขยาย	**ตำแหน่งเดิม**				**ตำแหน่งขยาย**
คอลัมน์ 1	**คอลัมน์ 2**	**คอลัมน์ 3**	**คอลัมน์ 4**	**คอลัมน์ 5**	**คอลัมน์ 6**
	0	1	0	0	
	1	0	1	1	
	1	0	0	0	
	1	0	1	1	
	0	1	0	0	
	1	1	0	1	
	1	1	1	1	
	0	0	0	1	

หลังจากนั้นเพิ่มบิตอีกจำนวน 2 บิตสำหรับแต่ละแถวโดยเพิ่มที่ตำแหน่งคอลัมน์ 1 และคอลัมน์ที่ 6 ซึ่งพิจารณาจากตำแหน่งของ R_0 ตามตารางที่ 3.5 ยกตัวอย่างเช่นหากพิจารณาเฉพาะแถวที่ 1 สังเกตว่าคอลัมน์ 1 และ 6 จะพิจารณาตำแหน่งที่ 32 และ 5 ของ R_0 ตามลำดับ ซึ่งพบว่าตำแหน่งที่ 32 มีค่าเป็น 1 และตำแหน่งที่ 5 มีค่าเป็น 1 ดังภาพตัวอย่าง (ตำแหน่งที่เป็นตัวหนาและขีดเส้นใต้สองเส้น)



ตำแหน่งขยาย	**ตำแหน่งเดิม**				**ตำแหน่งขยาย**
คอลัมน์ 1	**คอลัมน์ 2**	**คอลัมน์ 3**	**คอลัมน์ 4**	**คอลัมน์ 5**	**คอลัมน์ 6**
1	0	1	0	0	1
	1	0	1	1	
	1	0	0	0	
	1	0	1	1	
	0	1	0	0	
	1	1	0	1	
	1	1	1	1	
	0	0	0	1	

และเพิ่มบิตส่วนอื่นที่เหลือทั้งหมด ได้ดังนี้

ตำแหน่งขยาย	ตำแหน่งเดิม				ตำแหน่งขยาย
คอลัมน์ 1	คอลัมน์ 2	คอลัมน์ 3	คอลัมน์ 4	คอลัมน์ 5	คอลัมน์ 6
1	0	1	0	0	1
0	1	0	1	1	1
1	1	0	0	0	1
0	1	0	1	1	0
1	0	1	0	0	1
0	1	1	0	1	1
1	1	1	1	1	0
1	0	0	0	1	0

ดังนั้น $e = E(R_0)$ = 101001 010111 110001 010110 101001 011011 111110 100010

ขั้นตอนต่อไปคือหา $f = e \oplus K_1$ เมื่อ K_1 คือค่ากุญแจที่ได้จากขั้นตอนการจัดการกุญแจลับรอบแรก

ตัวอย่างที่ 3.5 จาก K_1 และ e ดังตัวอย่างที่ 3.1 และ 3.4 จงหา f

วิธีทำ ดำเนินการคำนวณหา $f = e \oplus K_1$ แบบตำแหน่งต่อตำแหน่งได้เป็นดังนี้

$$\begin{aligned} K_1 &= 000101\ 000101\ 111010\ 111111\ 110010\ 100011\ 111011\ 001000 \\ e &= 101001\ 010111\ 110001\ 010110\ 101001\ 011011\ 111110\ 100010 \quad \oplus \\ f &= \underline{101100\ 010010\ 001011\ 101001\ 011011\ 111000\ 000101\ 101010} \end{aligned}$$

ดังนั้น f = 101100 010010 001011 101001 011011 111000 000101 101010

ขั้นตอนต่อไปคือคำนวณหา $b_1, b_2, b_3, \cdots, b_8$ โดยแต่ละค่าเกิดจากนำบิตจาก f มาครั้งละ 6 บิต เริ่มจากตำแหน่งที่ 1 – 6 เพื่อนำมาเป็นพารามิเตอร์สำหรับกล่องเอสซึ่งมีทั้งหมด 8 กล่องสำหรับทั้ง 8 ค่า ตารางที่ 3.3 – 3.13 แสดงกล่องเอสที่แตกต่างกันทั้งหมด 8 กล่อง โดยที่หากคำนวณหา b_i จะใช้กล่องเอสที่ i ดังนี้

$$b_1 = S_1(f_1f_2f_3f_4f_5f_6)$$
$$b_2 = S_2(f_7f_8f_9f_{10}f_{11}f_{12})$$
$$b_3 = S_3(f_{13}f_{14}f_{15}f_{16}f_{17}f_{18})$$
$$b_4 = S_4(f_{19}f_{20}f_{21}f_{22}f_{23}f_{24})$$
$$b_5 = S_5(f_{25}f_{26}f_{27}f_{28}f_{29}f_{30})$$
$$b_6 = S_6(f_{31}f_{32}f_{33}f_{34}f_{35}f_{36})$$
$$b_7 = S_7(f_{37}f_{38}f_{39}f_{40}f_{41}f_{42})$$
$$b_8 = S_8(f_{43}f_{44}f_{45}f_{46}f_{47}f_{48})$$

โดยผลลัพธ์ทั้ง 8 ค่าจะมีขนาดที่เท่ากันคือ 4 บิตซึ่งมีค่าอยู่ระหว่าง 0 – 15 โดยหากค่าตัวเลขใดที่ถูกแปลงให้อยู่ในรูปแบบเลขฐานสองแล้วพบว่ามีจำนวนบิตน้อยกว่า 4 ให้ดำเนินการเพิ่มบิตโดยการเติม 0 ที่ตำแหน่งหน้าสุดจนกระทั่งครบ 4 บิต

กำหนดให้

$$b_i = S_i(f_kf_{k+1}f_{k+2}f_{k+3}f_{k+4}f_{k+5}) \tag{3.1}$$

จากสมการ (3.1) ผลลัพธ์ b_i หาได้จากการอ่านค่าจากตำแหน่งแถว (r) ซึ่งเกิดจากการแปลงจากเลขฐานสองเป็นเลขฐานสิบของการนำ f_{k+5} มาต่อกับ f_k ความหมายคือ $r = (f_kf_{k+5})_2$ และตำแหน่งคอลัมน์ (c) ซึ่งเกิดจากการแปลงจากเลขฐานสองเป็นเลขฐานสิบของการนำตำแหน่งที่เหลือมาเรียงต่อกันหรือ $c = (f_{k+1}f_{k+2}f_{k+3}f_{k+4})_2$

ดังนั้น b_i คือค่าที่เกิดจากการพิจารณากล่องเอสที่ i ที่อยู่ตำแหน่งแถวที่ r และคอลัมน์ที่ c หรือ

$$b_i = S_i(r, c) \tag{3.2}$$

ตัวอย่างที่ 3.6 จาก f ดังตัวอย่างที่ 3.5 จงหา $b_1 - b_8$

วิธีทำ จากตัวอย่างที่ 3.5

f = 101100 010010 001011 101001 011011 111000 000101 101010

ดังนั้น

$b_1 = S_1(f_1f_2f_3f_4f_5f_6) = S_1(101100) = S_1((10)_2, (0110)_2) = S_1(2, 6) = 2 = 0010_2$

$b_2 = S_2(f_7f_8f_9f_{10}f_{11}f_{12}) = S_2(010010) = S_2((00)_2, (1001)_2) = S_2(0, 9) = 7 = 0111_2$

$b_3 = S_3(f_{13}f_{14}f_{15}f_{16}f_{17}f_{18}) = S_3(001011) = S_3((01)_2, (0101)_2) = S_3(1, 5) = 4 = 0100_2$

$b_4 = S_4(f_{19}f_{20}f_{21}f_{22}f_{23}f_{24}) = S_4(101001) = S_4((11)_2, (0100)_2) = S_4(3, 4) = 10 = 1010_2$

$b_5 = S_5(f_{25}f_{26}f_{27}f_{28}f_{29}f_{30}) = S_5(011011) = S_5((01)_2, (1101)_2) = S_5(1, 13) = 9 = 1001_2$

$b_6 = S_6(f_{31}f_{32}f_{33}f_{34}f_{35}f_{36}) = S_6(111000) = S_6((10)_2, (1100)_2) = S_6(2, 12) = 1 = 0001_2$

$b_7 = S_7(f_{37}f_{38}f_{39}f_{40}f_{41}f_{42}) = S_7(000101) = S_7((01)_2, (0010)_2) = S_7(1, 2) = 11 = 1011_2$

$b_8 = S_8(f_{43}f_{44}f_{45}f_{46}f_{47}f_{48}) = S_8(101010) = S_8((10)_2, (0101)_2) = S_8(2, 5) = 12 = 1100_2$

ขั้นตอนต่อไปคือการคำนวณหา B ซึ่งเกิดจากการนำ $b_1 - b_8$ มาเชื่อมต่อกันโดยเริ่มจาก b_1 และเรียงลำดับไปจนถึงตำแหน่งสุดท้ายคือ b_8 และนำเข้าสู่กล่องสลับลำดับ P (Permutation Function) ดังตารางที่ 3.14 ในลำดับถัดไป

ตารางที่ 3.14 กล่องสลับลำดับ P

ตำแหน่งเดิม	ตำแหน่งใหม่		ตำแหน่งเดิม	ตำแหน่งใหม่
16	1		2	17
7	2		8	18
20	3		24	19
21	4		14	20
29	5		32	21
12	6		27	22
28	7		3	23
17	8		9	24
1	9		19	25
15	10		13	26
23	11		30	27
26	12		6	28
5	13		22	29
18	14		11	30
31	15		4	31
10	16		25	32

ตัวอย่างที่ 3.7 จาก $b_1 - b_8$ ดังตัวอย่างที่ 3.6 จงหา B_1 และ P(B_1)

วิธีทำ จากตัวอย่างที่ 3.6

$b_1 = 0010_2$, $b_2 = 0111_2$, $b_3 = 0100_2$, $b_4 = 1010_2$, $b_5 = 1001_2$, $b_6 = 0001_2$, $b_7 = 1011_2$, $b_8 = 1100_2$

เนื่องจาก

$$B_1 = b_1b_2b_3b_4b_5b_6b_7b_8$$

ดังนั้น

ตำแหน่ง 1 5 9 13 17 21 25 29

B_1 = 0010 0111 0100 1010 1001 0001 1011 1100

และจากตารางที่ 3.14

$$P = \text{P}(B_1) = 0110\ 1011\ 0100\ 0001\ 0110\ 0110\ 0111\ 0001$$

ขั้นตอนสุดท้ายคือการคำนวณหา R_1 จาก $R_1 = P \oplus L_0$

ตัวอย่างที่ 3.8 จาก P และ L_0 ดังตัวอย่างที่ 3.3 และ 3.7 จงหา R_1

วิธีทำ ดำเนินการคำนวณหา $R_1 = P \oplus L_0$ แบบตำแหน่งต่อตำแหน่งได้เป็นดังนี้

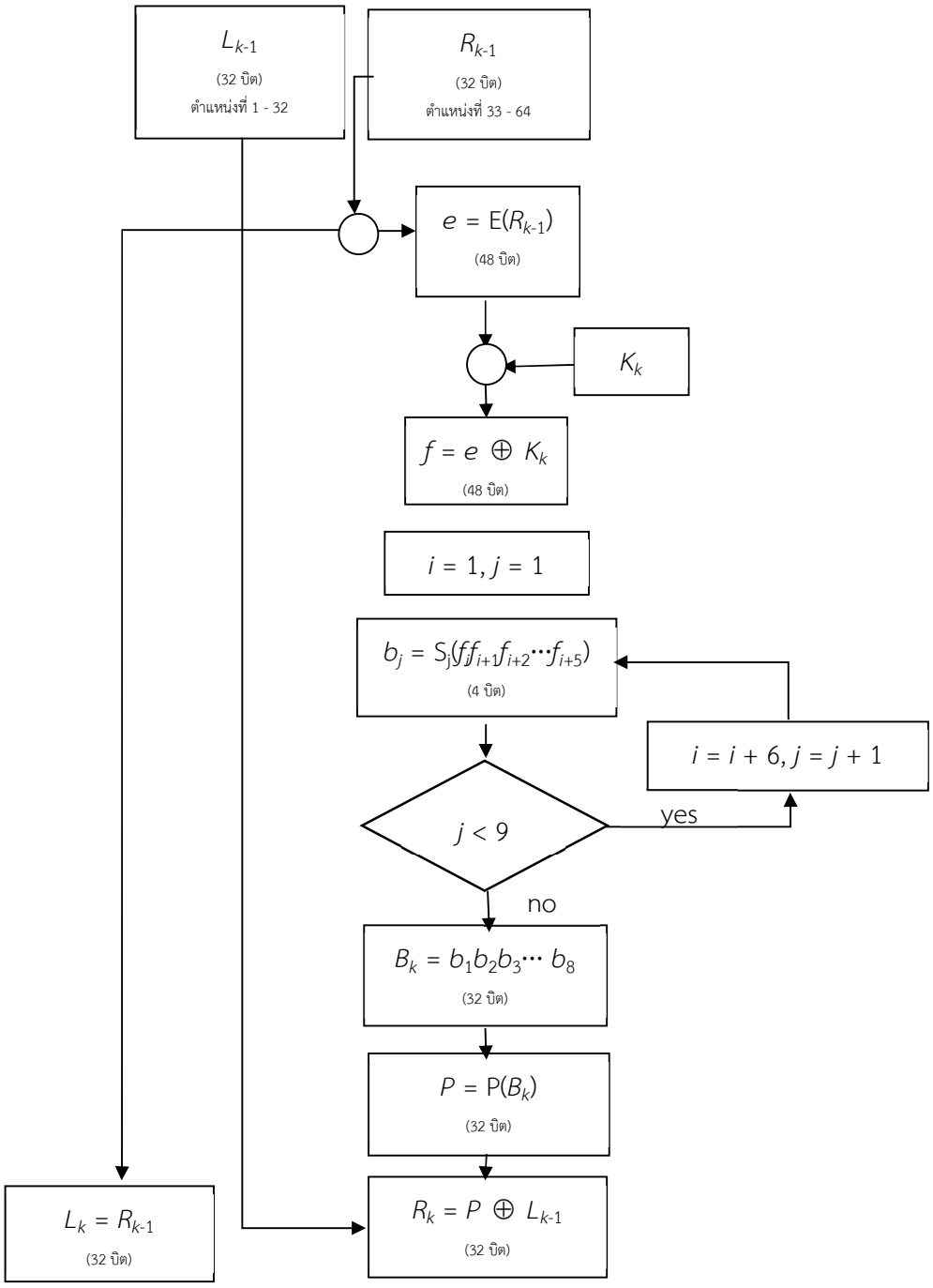
P = 0110 1011 0100 0001 0110 0110 0111 0001

L_0 = 1110 0110 0110 0111 1110 1101 1001 0010 $\oplus$

R_1 = 1000 1101 0010 0110 1000 1011 1110 0011

ดังนั้น R_1 = 1000 1101 0010 0110 1000 1011 1110 0011

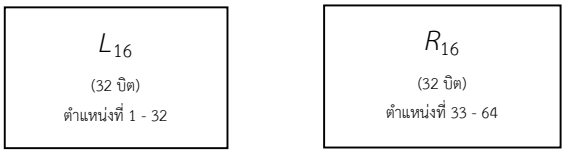
สำหรับการเข้ารหัสรอบที่ 2 – 16 ดังรูปที่ 3.4 จะแตกต่างจากรอบแรก คือไม่มีการใช้กล่องสลับลำดับ IP โดยในส่วนขั้นตอนอื่นจะเหมือนกันทุกประการ ดังนั้นจุดเริ่มต้นเข้ารหัสแต่ละรอบคือ L_{k-1} และ R_{k-1} เพื่อหา L_k และ R_k สำหรับแต่ละรอบการคำนวณ เมื่อ k แทนรอบการคำนวณระหว่างรอบที่ 2 – 16 อย่างไรก็ตามหลังจากได้ L_{16} และ R_{16} ซึ่งอยู่ที่รอบสุดท้ายของการคำนวณแล้ว ขั้นตอนสุดท้ายให้นำทั้งสองค่ามาผ่านกล่องสลับลำดับ IP^{-1} ดังตารางที่ 3.15 เพื่อหาข้อความไซเฟอร์ (C)

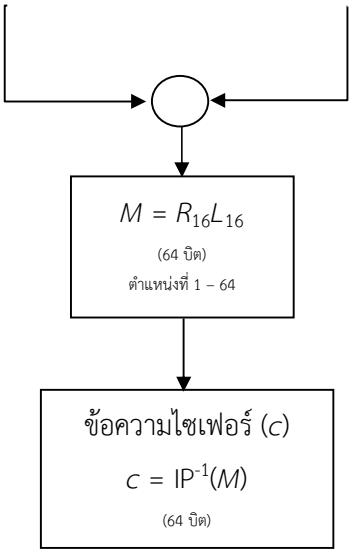


รูปที่ 3.4 การเข้ารหัสลับดีอีเอสรอบที่ 2 – 16

ตารางที่ 3.15 กล่องสลับลำดับ IP^{-1}

ตำแหน่งเดิม	ตำแหน่งใหม่		ตำแหน่งเดิม	ตำแหน่งใหม่
40	1		36	33
8	2		4	34
48	3		44	35
16	4		12	36
56	5		52	37
24	6		20	38
64	7		60	39
32	8		28	40
39	9		35	41
7	10		3	42
47	11		43	43
15	12		11	44
55	13		51	45
23	14		19	46
63	15		59	47
31	16		27	48
38	17		34	49
6	18		2	50
46	19		42	51
14	20		10	52
54	21		50	53
22	22		18	54
62	23		58	55
30	24		26	56
37	25		33	57
5	26		1	58
45	27		41	59
13	28		9	60
53	29		49	61
21	30		17	62
61	31		57	63
29	32		25	64





รูปที่ 3.5 การหาข้อความไซเฟอร์หลังสิ้นสุดการเข้ารหัสรอบที่ 16

รูปที่ 3.5 แสดงขั้นตอนสุดท้ายของการหาข้อความไซเฟอร์ (c) หลังเสร็จสิ้นกระบวนการเข้ารหัสทั้ง 16 รอบ โดยเริ่มจากหา M ซึ่งเกิดจากการนำ L_{16} มาเชื่อมต่อ R_{16} โดยให้ 32 บิตแรกคือค่าของ R_{16} และ 32 บิตสุดท้ายคือค่าของ L_{16} ดังนั้น M จึงมีขนาด 64 บิต และนำผลลัพธ์ที่ได้ไปผ่านกล่องสลับลำดับ IP^{-1} เพื่อให้ได้มาซึ่ง c

ตัวอย่างที่ 3.9 กำหนดให้

L_{16} = 1011 0110 0001 0100 1010 1101 1101 0011
R_{16} = 0001 0101 0010 1001 1100 1011 1000 1001

จงแสดงวิธีหา c
วิธีทำ เริ่มจากหา $M = R_{16}L_{16}$ ได้ดังนี้

M = 0001 0101 0010 1001 1100 1011 1000 1001 1011 0110 0001 0100 1010 1101 1101 0011

โดยมีค่าประจำตำแหน่งเป็นดังนี้

ตำแหน่ง 1 5 9 13 17 21 25 29

M = 0001 0101 0010 1001 1100 1011 1000 1001
1011 0110 0001 0100 1010 1101 1101 0011

ใช้กล่องสลับลำดับ IP^{-1} ได้ผลการเป็นดังนี้

c = 0101 1111 1000 0110 1110 1000 0001 1101 1110 0010 1001 1000 0000 0110 1000 1111

3. ความปลอดภัยรหัสลับดีอีเอส

เนื่องจากกุญแจลับที่ถูกนำมาใช้งานมีขนาด 64 บิตโดยถูกนำไปใช้สำหรับการตรวจสอบพาริตี้อีกจำนวน 8 บิต จึงเหลือใช้สำหรับกระบวนการเข้ารหัสและถอดรหัสเพียง 56 บิต ส่งผลให้จำนวนกุญแจที่เป็นไปได้ทั้งหมดคือ 2^{56} หรือประมาณ 7.21×10^{16} ค่า ซึ่งเป็นขนาดที่ใหญ่มหาศาลเพียงพอที่จะหลีกเลี่ยงการโจมตีแบบตะลุยได้ในช่วงเวลาแรกที่ถูกนำเสนอ อย่างไรก็ตามเนื่องจากเทคโนโลยีมีความก้าวหน้าอย่างรวดเร็ว ในปี ค.ศ.1997 รหัสลับดีอีเอสถูกโจมตีสำเร็จโดยวิธีการประมวลผลแบบขนานของคอมพิวเตอร์จำนวน 3,500 เครื่อง [1] ดังนั้นวิทยาการรหัสลับดีอีเอสจึงไม่ปลอดภัยอีกต่อไป และความนิยมลดลงเนื่องจากมีการนำเสนอวิทยาการรหัสลับแบบสมมาตรที่มีความปลอดภัยสูงขึ้นคือวิทยาการรหัสลับเออีเอสซึ่งจะถูกกล่าวถึงอีกครั้งในบทถัดไป

4. การเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง

เนื่องจากปัญหาด้านความปลอดภัยสำหรับรหัสลับดีอีเอส และยังไม่มีวิทยาการรหัสลับที่มีประสิทธิภาพที่สูงกว่ารหัสดีอีเอสในช่วงเวลานั้น นักวิจัยจึงค้นหาวิธีการปรับปรุงรหัสลับดีอีเอสเพื่อให้มีความปลอดภัยสูงมากยิ่งขึ้น โดยพบว่าการนำข้อความต้นฉบับมาผ่านกระบวนการเข้ารหัสโดยใช้รหัสลับดีอีเอสจำนวน 2 ครั้งที่ใช้ร่วมกับกุญแจลับ 2 ค่าที่แตกต่างกันจะช่วยเพิ่มความปลอดภัยให้สูงมากยิ่งขึ้น เนื่องมาจากขนาดของกุญแจลับมีขนาดที่สูงขึ้นเป็น 2x56 = 112 บิต จึงจำเป็นต้องใช้เวลาสำหรับการประมวลผลอย่างมหาศาลหากใช้วิธีการโจมตีแบบตะลุย

กำหนดให้ m คือข้อความต้นฉบับ k_1 และ k_2 คือกุญแจขนาด 56 บิต (ไม่นับรวมกับบิตที่ใช้สำหรับตรวจสอบพาริตี้) $f(m, k)$ แทนฟังก์ชันการเข้ารหัสข้อความต้นฉบับ m ด้วยกุญแจลับ k และ

$f^{-1}(m, k)$ แทนฟังก์ชันการถอดรหัสข้อความไซเฟอร์ c ด้วยกุญแจลับ k การเข้ารหัสลับโดยรหัสลับดีอีเอส 2 ครั้งสามารถดำเนินการได้ดังนี้

การเข้ารหัสครั้งที่ 1:

$$t = f(m, k_1) \tag{3.3}$$

การเข้ารหัสครั้งที่ 2:

$$c = f(t, k_2) \tag{3.4}$$

การถอดรหัสครั้งที่ 1:

$$t = f^{-1}(c, k_2) \tag{3.5}$$

การถอดรหัสครั้งที่ 2:

$$m = f^{-1}(t, k_1) \tag{3.6}$$

การเลือกใช้กุญแจลับสำหรับขั้นตอนการถอดรหัสจะใช้ค่ากุญแจที่ถูกใช้เข้ารหัสครั้งที่ 2 สำหรับการถอดรหัสครั้งแรก และใช้กุญแจที่ถูกใช้สำหรับการเข้ารหัสครั้งแรกสำหรับการถอดรหัสครั้งที่ 2 ดังนั้นจากสมการที่ (3.3) ถึง (3.6) เนื่องจาก k_1, k_2 ถูกใช้สำหรับการเข้ารหัสครั้งที่ 1 และ 2 ตามลำดับ จึงได้ว่า k_2, k_1 จะถูกใช้สำหรับการถอดรหัสตามลำดับ

ถึงแม้ว่าการโจมตีแบบตะลุยจะไม่มีประสิทธิภาพหากนำมาใช้กับการเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง สมมติผู้ไม่ประสงค์ดีทราบข้อความต้นฉบับ m และข้อความไซเฟอร์ c ที่ถูกการเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง ผู้ไม่ประสงค์ดีจะสามารถนำวิธีการโจมตีแบบพบกันครึ่งทาง (meet-in-the-middle attack) มาประยุกต์ใช้แทนได้ ด้วยเหตุผลคือจากสมการที่ (3.3) ถึง (3.6) สังเกตได้ว่าการเข้ารหัส m ครั้งแรกด้วยกุญแจลับ k_1 จะได้ข้อความไซเฟอร์คือ t ในทางกลับกันการถอดรหัส c ครั้งแรกโดยใช้กุญแจลับ k_2 จะได้ข้อความไซเฟอร์คือ t เช่นเดียวกัน ดังนั้นสามารถนำวิธีการโจมตีแบบพบกันครึ่งทางมาประยุกต์ใช้งานได้โดยการเข้ารหัส m ครั้งแรก (โดยไม่จำเป็นต้องเข้ารหัสครั้งที่ 2) จะใช้กุญแจลับที่เป็นไปได้ทั้งหมดซึ่งมีจำนวน 2^{56} ค่าและเก็บผลลัพธ์ไว้ ในทางกลับกันผู้ไม่ประสงค์ดีดำเนินการถอดรหัส c ครั้งแรก (โดยไม่จำเป็นต้องถอดรหัสครั้งที่ 2) โดยใช้กุญแจลับที่เป็นไปได้ทั้งหมดซึ่งมีจำนวน 2^{56} ค่าและเก็บผลลัพธ์ไว้ หลังจากนั้นนำผลลัพธ์ที่เก็บไว้ทั้งสองฝั่งมาเปรียบเทียบกันโดยหาตำแหน่งที่ได้ข้อความไซเฟอร์ที่ได้จากการเข้ารหัสครั้งแรกที่มีผลลัพธ์ตรงกับข้อความไซ

เฟอร์ที่ได้จากการถอดรหัสครั้งแรก จึงได้ค่ากุญแจที่ใช้สำหรับเข้ารหัสและถอดรหัสที่อยู่ตำแหน่งที่ได้ผลลัพธ์เท่ากันคือ k_1 และ k_2

จากการนำวิธีการโจมตีแบบพบกันครึ่งทางมาประยุกต์ใช้งานสำหรับโจมตีการเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง จำเป็นต้องใช้กุญแจสำหรับการคำนวณทั้งหมด $2x2^{56} = 2^{57}$ ค่าซึ่งพบว่าใช้เวลาลดลงเป็นอย่างมาก

5. การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้ง

จากปัญหาของการใช้วิธีการเข้ารหัสด้วยรหัสลับดีอีเอสจำนวน 2 ครั้งด้วยค่ากุญแจลับที่มีความแตกต่างกันจำนวน 2 ชุด ที่ถูกโจมตีได้ด้วยวิธีการโจมตีแบบพบกันครึ่งทาง ส่งผลให้ผู้ไม่ประสงค์ดีไม่จำเป็นต้องค้นหากุญแจที่มีขนาดใหญ่ถึง 2^{112} ค่า แต่ใช้เพียง 2^{57} ค่าซึ่งพบว่าขนาดของกุญแจลับที่เป็นไปได้ทั้งหมดเพิ่มขึ้นเพียง 2 เท่า หากเปรียบเทียบกับการเข้ารหัสเพียง 1 ครั้ง ดังนั้นเพื่อเพิ่มความปลอดภัยให้สูงขึ้นจึงได้มีการพัฒนาการเข้ารหัสดีอีเอสแบบ 3 ครั้ง โดยแบ่งออกเป็น 2 วิธี คือวิธีที่ใช้กุญแจลับที่มีค่าแตกต่างกัน 3 ชุด และการใช้กุญแจลับที่มีค่าแตกต่างกันเพียง 2 ชุด

5.1 การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 3 ชุด

การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 3 ชุด คือวิธีการเข้ารหัสแบบดีอีเอสที่มีความปลอดภัยสูงที่สุด โดยขนาดกุญแจลับที่ใช้คือ 3x56 = 168 บิต ซึ่งเป็นขนาดที่ใหญ่มหาศาล ดังนั้นหากผู้ไม่ประสงค์ดีต้องการค้นหากุญแจลับที่ถูกต้องจำเป็นต้องค้นหากุญแจลับที่ถูกต้องเพียงค่าเดียวจากจำนวนที่เป็นไปได้ทั้งหมดคือ 2^{168} ค่า

กำหนดให้ m แทนข้อความต้นฉบับบ k_1, k_2 และ k_3 คือกุญแจขนาด 56 บิต (ไม่นับรวมกับบิตที่ใช้สำหรับตรวจสอบพาริตี้) การเข้ารหัสลับและการถอดรหัสลับโดยรหัสลับดีอีเอส 3 ครั้ง โดยใช้กุญแจลับ 3 ชุด สามารถดำเนินการได้ดังนี้

การเข้ารหัสครั้งที่ 1:

$$t = f(m, k_1) \tag{3.7}$$

การเข้ารหัสครั้งที่ 2:

$$h = f(t, k_2) \tag{3.8}$$

การเข้ารหัสครั้งที่ 3:

$$c = f(h, k_3) \quad (3.9)$$

การถอดรหัสครั้งที่ 1:

$$h = f^{1}(c, k_3) \quad (3.10)$$

การถอดรหัสครั้งที่ 2:

$$t = f^{1}(h, k_2) \quad (3.11)$$

การถอดรหัสครั้งที่ 3:

$$c= = f^{1}(t, k_1) \quad (3.12)$$

5.2 การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 2 ชุด

การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 2 ชุด จะมีความแตกต่างกับวิธีการที่ใช้กุญแจลับ 3 ชุดเพียงเล็กน้อยคือ วิธีการนี้จะใช้กุญแจลับเพียง 2 ชุด ซึ่งเทียบเท่ากับการเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง แต่มีความปลอดภัยที่สูงกว่าเป็นอย่างมาก ข้อดีคือความสะดวกในการใช้งาน และมีความปลอดภัยสูง

กำหนดให้ m แทนข้อความต้นฉบับบ k_1, k_2 คือกุญแจขนาด 56 บิต (ไม่นับรวมกับบิตที่ใช้สำหรับตรวจสอบพาริตี้) การเข้ารหัสลับและการถอดรหัสลับโดยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 2 ชุดสามารถดำเนินการได้ดังนี้

การเข้ารหัสครั้งที่ 1:

$$t = f(m, k_1) \quad (3.13)$$

การเข้ารหัสครั้งที่ 2:

$$h = f(t, k_2) \quad (3.14)$$

การเข้ารหัสครั้งที่ 3:

$$c = f(h, k_1) \quad (3.15)$$

การถอดรหัสครั้งที่ 1:

$$h = f^{-1}(c, k_1) \tag{3.16}$$

การถอดรหัสครั้งที่ 2:

$$t = f^{-1}(h, k_2) \tag{3.17}$$

การถอดรหัสครั้งที่ 3:

$$c = f^{-1}(t, k_1) \tag{3.18}$$

การเข้ารหัสลับด้วยรหัสลับดีอีเอสแบบ 3 ครั้งเป็นวิธีการที่ได้รับการยอมรับว่ามีความปลอดภัยสูงที่สุดเมื่อเปรียบเทียบกับการนำรหัสลับดีอีเอสมาประยุกต์ใช้ในรูปแบบอื่น

6. บทสรุปสาระสำคัญ

วิทยาการรหัสลับดีอีเอสคือวิทยาการรหัสลับแบบสมมาตรที่ถูกพัฒนาครั้งแรก ในปี ค.ศ. 1977 โดยสถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (National Institute of Standards and Technology) มีชื่อย่อคือ NIST ซึ่งกุญแจลับจะต้องมีขนาด 64 บิตแต่จะถูกนำเข้าสู่กระบวนการเข้ารหัสลับและกระบวนการถอดรหัสลับเพียง 56 บิต โดยหลังจากนำกุญแจลับเข้าสู่กระบวนการจัดการกุญแจลับแล้วจะได้กุญแจลับที่มีค่าแตกต่างกันทั้งหมด 16 ค่าเพื่อใช้สำหรับกระบวนการเข้ารหัสลับและกระบวนการถอดรหัสลับที่มีจำนวนรอบเท่ากันคือ 16 รอบ โดยลำดับขั้นตอนของกระบวนการถอดรหัสลับจะเป็นการดำเนินการย้อนกลับของกระบวนการเข้ารหัสลับเช่นขั้นตอนวิธีสำหรับการเข้ารหัสลับในรอบที่ 16 จะเทียบเท่ากับขั้นตอนวิธีการถอดรหัสลับในรอบแรก ขั้นตอนวิธีการถอดรหัสลับในรอบที่ 2 จะมีขั้นตอนตรงกับขั้นตอนวิธีการเข้ารหัสลับในรอบที่ 15 จนกระทั่งถึงขั้นตอนวิธีการถอดรหัสลับในรอบสุดท้ายซึ่งจะเทียบเท่ากับขั้นตอนวิธีการเข้ารหัสลับในรอบแรก

อย่างไรก็ตามเนื่องจากประสิทธิภาพของทรัพยากรที่สูงขึ้นตามกาลเวลาส่งผลให้วิทยาการรหัสลับดีอีเอสถูกโจมตีครั้งแรกในปี ค.ศ.1997 โดยวิธีการประมวลผลแบบขนานของคอมพิวเตอร์จำนวน 3,500 เครื่อง ดังนั้นจึงจำเป็นต้องปรับปรุงวิทยาการรหัสลับดีอีเอสให้มีประสิทธิภาพที่สูงขึ้นโดยการปรับปรุงครั้งแรกได้เสนอการเข้ารหัสด้วยรหัสลับดีอีเอส 2 ครั้ง แต่กลับถูกโจมตีได้โดยใช้เทคนิควิธีการพบกันครึ่งทาง จึงได้มีการพัฒนาอย่างต่อเนื่องและเกิดเป็นการเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งซึ่งเป็นเทคนิควิธีที่มีประสิทธิภาพสูงโดยแบ่งเป็น 2 วิธีคือการเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งที่ใช้กุญแจลับแตกต่างกัน 2 ชุด และการเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งที่ใช้กุญแจลับแตกต่าง

กัน 3 ชุดซึ่งเป็นเทคนิควิธีที่มีประสิทธิภาพสูงกว่าวิธีคือการเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งที่ใช้กุญแจลับแตกต่างกัน 2 ชุด

แบบฝึกหัดท้ายบท
บทที่ 3

1. รหัสดีอีเอสถูกพัฒนาครั้งแรกโดยหน่วยงานที่ชื่อว่าอะไร
2. กุญแจลับสำหรับรหัสดีอีเอสมีขนาดกี่บิต
3. กระบวนการเข้ารหัสแบบดีอีเอสต้องดำเนินการทั้งหมดกี่รอบ
4. กระบวนการถอดรหัสแบบดีอีเอสต้องดำเนินการทั้งหมดกี่รอบ
5. ค่ากุญแจลับที่ถูกใช้สำหรับการเข้ารหัสแบบดีอีเอสในรอบแรกจะถูกนำไปใช้สำหรับการถอดรหัสในรอบที่เท่าไร
6. ค่ากุญแจลับที่ถูกใช้สำหรับการเข้ารหัสแบบดีอีเอสในรอบที่ 4 จะถูกนำไปใช้สำหรับการถอดรหัสในรอบที่เท่าไร
7. กระบวนการเข้ารหัสในรอบที่ 3 จำเป็นต้องมีการหมุนวนซ้ายทั้งหมดกี่บิต
8. ผลลัพธ์หลังจากผ่านฟังก์ชันขยายบิตจะมีจำนวนกี่บิต
9. กำหนดให้กุญแจลับ K = C92183E1C55E789D จงหาผลลัพธ์หลังจากนำค่าดังกล่าวไปผ่านกล่องสลับลำดับ PC1
10. กำหนดให้ K_{3_left} = 1011 0110 1100 1101 0001 1010 0101 จงหา K_{4_left}
11. กำหนดให้ R_0 = 1011 0100 0101 1100 0011 1001 0001 1011 จงหา E(R_0)
12. จงหาผลลัพธ์ของ S_1(110010)
13. จงหาผลลัพธ์ของ S_3(011011)
14. กำหนดให้ข้อความต้นฉบับ m = 1001 1000 0101 1000 1100 1110 1110 1010 0110 1001 0001 0010 0100 1110 0010 1101 จงหา L_0
15. การเข้ารหัสด้วยรหัสลับดีอีเอส 3 ครั้งโดยใช้กุญแจลับ 3 ชุดมีค่ากุญแจลับที่เป็นไปได้ทั้งหมดเท่าไร
16. กำหนดให้

    L_{16} = 0001 1110 0000 1100 0100 0001 0110 1011

    R_{16} = 1010 0001 0110 1100 1001 0001 1010 0010

    จงคำนวณหา M
17. จากคำตอบข้อ 16 จงหาข้อความไซเฟอร์

บทที่ 4
วิทยาการรหัสลับเออีเอส

ความก้าวหน้าทางเทคโนโลยีสารสนเทศอย่างก้าวกระโดดส่งผลให้อุปกรณ์ต่างๆ ทางคอมพิวเตอร์มีสมรรถนะที่สูงขึ้นโดยเฉพาะอย่างยิ่งหน่วยประมวลผลที่ทำให้การประมวลผลงานต่างๆ ใช้เวลาลดลง ด้วยเหตุผลดังกล่าวนี้จึงทำให้ความปลอดภัยของวิทยาการรหัสลับแบบดีอีเอสซึ่งใช้กุญแจลับที่มีขนาดเพียง 56 บิตไม่เพียงพอต่อการหลีกเลี่ยงการโจมตีจากผู้ไม่ประสงค์ดี ดังนั้นในปี ค.ศ. 2001 ได้มีการนำเสนอวิทยาการรหัสลับแบบสมมาตรที่มีความปลอดภัยที่สูงขึ้นโดยนักออกแบบรหัสลับชาวเบลเยี่ยมชื่อ เจาน์ แดเมน (Joan Daemen) และ วินเซ็นท์ ไรจ์เมน (Vincent Rijmen) ที่มีชื่อเรียกว่า เออีเอส (Advanced Encryption Standard, AES) [29] ซึ่งได้รับการยอมรับให้เป็นมาตรฐานใหม่โดยสถาบันมาตรฐานและเทคโนโลยีแห่งสหรัฐอเมริกา

โครงสร้างของรหัสเออีเอสคือรับข้อความต้นฉบับที่มีขนาดคงที่คือ 128 บิต โดยมีกุญแจลับให้เลือกใช้งาน 3 ขนาดดังนี้ 128, 192 และ 256 บิต โดยแต่ละขนาดจะส่งผลให้จำนวนรอบของกระบวนการเข้ารหัส และถอดรหัสมีความแตกต่างกันพิจารณาดังตารางที่ 4.1

เนื่องจากการดำเนินการหลักของรหัสเออีเอส คือการบวก ลบ คูณ หาร ระหว่างฟังก์ชันพหุนามเหนือฟิลด์จำกัด ดังนั้นจึงจำเป็นต้องกำหนดฟังก์ชันพหุนามไม่ลดรูปซึ่งจากมาตรฐานเออีเอสได้กำหนดให้ฟังก์ชันพหุนามไม่ลดรูปมีค่าคือ $f(x) = x^8 + x^4 + x^3 + x + 1$

ตารางที่ 4.1 จำนวนรอบการคำนวณการเข้ารหัสและถอดรหัสสำหรับแต่ละขนาดของกุญแจลับ

ขนาดกุญแจลับ (บิต)	**จำนวนรอบการคำนวณ**
128	10
192	12
256	14

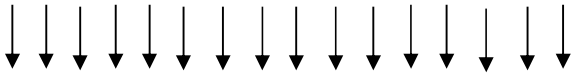
จากตารางที่ 4.1 จำนวนรอบของการเข้ารหัส และถอดรหัสสำหรับแต่ละขนาดจะมีค่าเท่ากัน เช่น กรณีเลือกใช้งานกุญแจลับขนาด 128 บิต จำเป็นต้องใช้จำนวนรอบการคำนวณกระบวนการเข้ารหัสและถอดรหัสข้อมูลจำนวน 10 รอบ เป็นต้น

1. การจัดเตรียมข้อความต้นฉบับและข้อความไซเฟอร์

ข้อความต้นฉบับที่นำมาผ่านกระบวนการเข้ารหัส และข้อความไซเฟอร์ที่นำมาผ่านกระบวนการถอดรหัสมีขนาดเท่ากันคือ 128 บิต โดยก่อนการนำข้อความทั้ง 2 ชนิดนี้เข้าสู่กระบวนการเข้ารหัส หรือกระบวนการถอดรหัสจำเป็นต้องนำมาบรรจุไว้ในเมตริกซ์ขนาด 4x4 โดยแต่ละตำแหน่งบรรจุข้อความขนาด 8 บิต (1 ไบต์) ดังนั้นข้อความต้นฉบับ และข้อความไซเฟอร์จึงถูกแบ่งออกเป็น 16 ส่วน โดยข้อความที่ถูกตัดมาจะถูกจัดเรียงลงตามแนวแถวซึ่งเริ่มจากคอลัมน์แรก (คอลัมน์ที่ 1) และวนกลับไปบรรจุแถวแรก (แถวที่ 1) ที่อยู่ในคอลัมน์ถัดไปจนครบทั้ง 16 ส่วน

ตัวอย่างที่ 4.1 จงแสดงผลลัพธ์จากการนำข้อความต้นฉบับขนาด 16 ไบต์ (128 บิต) มาบรรจุลงเมตริกซ์ขนาด 4x4 เพื่อจัดเตรียมไว้สำหรับกระบวนการเข้ารหัสลับเออีเอส

ตำแหน่ง 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



m = 1e 2f 31 48 5a bc 79 1a de f1 23 55 ab c7 19 2c

วิธีทำ

นำข้อความต้นฉบับ 4 ตำแหน่งแรก (ตำแหน่งที่ 1 - 4) บรรจุลงเมตริกซ์ตำแหน่งแถวที่ 1 – 4 ตามลำดับที่คอลัมน์ที่ 1 ได้ดังนี้

1e			
2f			
31			
48			

นำข้อความต้นฉบับ 4 ตำแหน่งถัดไป (ตำแหน่งที่ 5 - 8) บรรจุลงเมตริกซ์ตำแหน่งแถวที่ 1 – 4 ตามลำดับที่คอลัมน์ที่ 2 ได้ดังนี้

1e	5a		
2f	bc		
31	79		
48	1a		

นำข้อความต้นฉบับ 4 ตำแหน่งถัดไป (ตำแหน่งที่ 9 - 12) บรรจุลงเมตริกซ์ตำแหน่งแถวที่ 1 – 4 ตามลำดับที่คอลัมน์ที่ 3 ได้ดังนี้

1e	5a	de	
2f	bc	f1	
31	79	23	
48	1a	55	

และดำเนินการเช่นเดิมกับส่วนที่เหลือทั้งหมดได้เป็น ดังนี้

1e	5a	de	ab
2f	bc	f1	c7
31	79	23	19
48	1a	55	2c

2. การจัดการกุญแจลับเออีเอส

กุญแจลับที่สามารถนำมาใช้สำหรับวิทยาการรหัสลับเออีเอสแบ่งเป็นทั้งหมด 3 ขนาดคือ 128, 192 และ 256 บิต ซึ่งแต่ละขนาดจะใช้รอบการคำนวณสำหรับกระบวนการเข้ารหัสลับ และกระบวนการถอดรหัสลับที่แตกต่างกัน โดยจำเป็นต้องนำกุญแจลับมาดำเนินการขยายค่าสำหรับการนำไปประยุกต์ใช้งานแต่ละรอบของการดำเนินการ อย่างไรก็ตามขั้นตอนวิธีการดำเนินการขยายขนาดกุญแจลับทั้ง 3 ขนาดจะเหมือนกันทั้งหมด สำหรับการสร้างกุญแจลับที่ใช้ในการคำนวณแต่ละรอบเกิดจากการนำกุญแจย่อยที่มีขนาด 4 ไบต์ทั้งหมด 4 ค่ามาสร้างเป็นเมตริกซ์ที่มีขนาด 4x4 ดังนั้นหากกุญแจลับมีขนาด 128 บิต ซึ่งมีการคำนวณทั้งหมด 10 รอบจำเป็นต้องสร้างกุญแจย่อยทั้งหมด 4 ค่า (สำหรับกุญแจลับ 1 ค่า) x11 รอบ (10 รอบการคำนวณในวงวน และ 1 รอบก่อนเข้าสู่วงวนการคำนวณ) = 44 ค่า หากกุญแจลับมีขนาด 196 บิต ซึ่งมีการคำนวณทั้งหมด 12 รอบจำเป็นต้องสร้าง

กุญแจย่อยทั้งหมด 4 ค่า (สำหรับกุญแจลับ 1 ค่า) x13 รอบ (12 รอบการคำนวณในวงวน และ 1 รอบก่อนเข้าสู่วงวนการคำนวณ) = 52 ค่า และ หากกุญแจลับมีขนาด 256 บิต ซึ่งมีการคำนวณทั้งหมด 14 รอบจำเป็นต้องสร้างกุญแจย่อยทั้งหมด 4 ค่า (สำหรับกุญแจลับ 1 ค่า) x15 รอบ (14 รอบการคำนวณในวงวน และ 1 รอบก่อนเข้าสู่วงวนการคำนวณ) = 60 ค่า หัวข้อถัดไปจะกล่าวถึงกระบวนการทั้งหมดที่จำเป็นต้องใช้สำหรับการขยายขนาดกุญแจ

2.1 การดำเนินการ Rotate Word

การดำเนินการ Rotate Word คือคำสั่งที่ใช้สำหรับหมุนกุญแจย่อยที่มีขนาด 4 ไบต์ไปทางซ้ายจำนวน 1 ครั้ง โดยที่เป็นการหมุนแบบครั้งละ 1 ไบต์

ตัวอย่างที่ 4.2 กำหนดให้กุญแจย่อย *w*[0] = 123e4f76 จงหาผลลัพธ์หลังจากผ่าน Rotate Word
วิธีทำ หากพิจารณาเป็นตำแหน่งๆ ละ 1 ไบต์ โดยกำหนดให้ตำแหน่งที่ 1 อยู่ทางซ้ายสุด โดยเรียงไปจนถึงตำแหน่งที่ 4 ซึ่งเป็นตำแหน่งสุดท้ายได้ดังนี้

ค่าตั้งต้น	**ตำแหน่งที่**			
	1	**2**	**3**	**4**
w[0]	12	3e	4f	76

หากนำค่า *w*[0] มาผ่าน Rotate Word ดังนั้นตำแหน่งที่ 1 ของ *w*[0] จะย้ายไปตำแหน่งที่ 4 ของผลลัพธ์ และตำแหน่งที่ 2, 3 และ 4 ของ *w*[0] จะย้ายไปอยู่ตำแหน่งที่ 1, 2 และ 3 ของผลลัพธ์ตามลำดับ

กำหนดให้ *x* คือผลลัพธ์หลังจากนำ *w*[0] มาผ่านกระบวนการ Rotate Word ดังนั้น

ผลลัพธ์	**ตำแหน่งที่**			
	1	**2**	**3**	**4**
x	3e	4f	76	12

ดังนั้นผลลัพธ์ที่ได้คือ 3e4f7612

2.2 การดำเนินการ Substitute Word

การดำเนินการ Substitute Word คือการใช้กล่องเอส ดังตารางที่ 4.2 เพื่อใช้สำหรับแทนที่ไบต์ข้อมูลเดิม โดยใช้วิธีนำ 4 บิตที่มีนัยสำคัญสูงสุด มาพิจารณาเป็นข้อมูลตำแหน่งของแถวในกล่องเอส และนำ 4 บิตสุดท้ายที่มีนัยสำคัญต่ำที่สุดมาพิจารณาเป็นข้อมูลตำแหน่งของคอลัมน์ในกล่องเอส โดยทั่วไปการดำเนินการ Substitute Word คือการแทนที่กุญแจย่อยที่มีขนาด 4 ไบต์

ตารางที่ 4.2 กล่องเอสสำหรับการดำเนินการ Substitute Word และ กระบวนการ Substitute Byte

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
แถว	**0**	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

ตัวอย่างที่ 4.3 จงแสดงวิธีการใช้กล่องเอสเพื่อแทนที่ไบต์ข้อมูลที่มีค่าเป็น 3e

วิธีทำ จากไบต์ข้อมูลที่โจทย์กำหนด ให้พิจารณาตำแหน่งที่อยู่ในกล่องเอสดังนี้

4 บิตที่มีนัยสำคัญสูงสุดคือ “3” ดังนั้นให้พิจารณาแถวที่ 3

4 บิตที่มีนัยสำคัญต่ำสุดคือ “e” ดังนั้นให้พิจารณาคอลัมน์ที่ e

และเนื่องจากไบต์ข้อมูลที่อยู่ในแถวที่ 3 คอลัมน์ที่ e ในกล่องมีค่าเป็น b2

ดังนั้นสรุปได้ว่า b2 คือไบต์ข้อมูลที่แทนที่ 3e โดยใช้กล่องเอส

2.3 การดำเนินการ Rcon

Rcon คือตัวแปรแบบอาร์เรย์ที่เก็บค่าเลขฐานสิบหก โดยหลังจากได้ผลลัพธ์จากการคำนวณแล้ว จะมีไบต์ข้อมูลที่มีค่าเป็น 0 ทั้งหมดจำนวน 3 ไบต์ตามหลังดังนี้

$$Rcon[i] = 2^{i-1} + \text{“000000”} \qquad (4.1)$$

เมื่อ *i* คือตำแหน่งอ้างอิงโดยมีค่าเริ่มต้นที่ 1

โดยตารางที่ 4.3 แสดงตัวอย่างผลลัพธ์จำนวน 10 ค่าของ *Rcon[i]* โดยเริ่มจากตำแหน่งอ้างอิงจากตำแหน่งที่ 1 จนกระทั่งถึงตำแหน่งที่ 10

ตารางที่ 4.3 ผลลัพธ์ *Rcon[i]* จำนวน 10 ค่าโดยเริ่มจากตำแหน่งอ้างอิงที่ 1

ตำแหน่งอ้างอิง (*i*)	*Rcon[i]*
1	01000000
2	02000000
3	04000000
4	08000000
5	10000000
6	20000000
7	40000000
8	80000000
9	1b000000
10	36000000

จากตารางที่ 4.3 อธิบายผลลัพธ์ที่ได้จากแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1:

$$Rcon[1] = 2^{1-1} + \text{“000000”}$$
$$= 2^{0} + \text{“000000”} = 01000000$$

ตำแหน่งที่ 2:

$$Rcon[2] = 2^{2-1} + \text{“000000”}$$
$$= 2^{1} + \text{“000000”} = 02000000$$

ตำแหน่งที่ 3:

$$Rcon[3] = 2^{3-1} + \text{“000000”}$$
$$= 2^{2} + \text{“000000”} = 04000000$$

ตำแหน่งที่ 4:

$$Rcon[4] = 2^{4-1} + \text{“000000”}$$
$$= 2^{3} + \text{“000000”} = 08000000$$

ตำแหน่งที่ 5:

$$Rcon[5] = 2^{5-1} + \text{“000000”}$$
$$= 2^{4} + \text{“000000”} = 10000000, (2^{4} = 16_{10} = 10_{16})$$

ตำแหน่งที่ 6:

$$Rcon[6] = 2^{6-1} + \text{“000000”}$$
$$= 2^{5} + \text{“000000”} = 20000000, (2^{5} = 32_{10} = 20_{16})$$

ตำแหน่งที่ 7:

$$Rcon[7] = 2^{7-1} + \text{“000000”}$$
$$= 2^{6} + \text{“000000”} = 40000000, (2^{6} = 64_{10} = 40_{16})$$

ตำแหน่งที่ 8:

$$Rcon[8] = 2^{8-1} + \text{“000000”}$$
$$= 2^{7} + \text{“000000”} = 80000000, (2^{7} = 128_{10} = 80_{16})$$

ตำแหน่งที่ 9:

$$Rcon[9] = 2^{9-1} + \text{“000000”}$$
$$= 2^{8} + \text{“000000”}$$
$$= 100 + \text{“000000”}, \quad (2^{8} = 256_{10} = 100_{16})$$

เนื่องจาก $100_{16} = 100000000_2 = x^8$ มีบิตข้อมูลจำนวนทั้งหมด 9 บิต ซึ่งเกินขอบเขตฟิลด์จำกัด ดังนั้นจึงจำเป็นต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปดังนี้

$$x^8 \bmod x^8 + x^4 + x^3 + x + 1 = x^4 + x^3 + x + 1$$
$$= 00011011_2$$
$$= 1b_{16}$$

ดังนั้น $Rcon[9] = 1b + \text{“000000”} = 1b000000$

ตำแหน่งที่ 10:

$$Rcon[10] = 2^{10-1} + \text{“000000”}$$
$$= 2^9 + \text{“000000”}$$
$$= 100 + \text{“000000”}, \quad (2^9 = 512_{10} = 200_{16})$$

เนื่องจาก $200_{16} = 1000000000_2 = x^9$ มีบิตข้อมูลจำนวนทั้งหมด 10 บิต ซึ่งเกินขอบเขตฟิลด์จำกัด ดังนั้นจึงจำเป็นต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปดังนี้

$$x^9 \bmod x^8 + x^4 + x^3 + x + 1 = x^5 + x^4 + x^2 + x$$
$$= 00110110_2 = 36_{16}$$

ดังนั้น $Rcon[10] = 36 + \text{“000000”} = 36000000$

2.4 การก่อกำเนิดกุญแจลับเออีเอส

ขั้นตอนวิธีการก่อกำเนิดกุญแจลับสำหรับวิทยาการรหัสลับเออีเอสเกิดจากการนำการดำเนินการทั้ง 3 วิธีที่กล่าวไว้ในหัวข้อที่ 2.1 ถึง 2.3 มาประยุกต์ใช้งานร่วมกัน กำหนดให้ตำแหน่งทางซ้ายสุดของกุญแจลับแทนตำแหน่งไบต์ที่ 1 และเรียงไปจนกระทั่งถึงตำแหน่งทางขวาสุดซึ่งคือตำแหน่งไบต์สุดท้าย

จากขั้นตอนวิธี 4.1 กำหนดให้ n แทนข้อมูลที่บรรจุภายในเมตริกต์ขนาด 4x4 อธิบายการดำเนินการแต่ละฟังก์ชันได้ดังนี้

SubWord(n) คือการนำ n มาผ่านการดำเนินการ Substitute Word

RotWord(n) คือการนำ n มาผ่านการดำเนินการ Rotate Word

n_k มีค่าเป็น 4, 6 หรือ 8 เมื่อกุญแจลับมีขนาด 128, 192 หรือ 256 ตามลำดับ

n_r คือจำนวนรอบการคำนวณซึ่งมีค่าเป็นดังตารางที่ 4.1

กุญแจที่ใช้สำหรับการคำนวณในแต่ละรอบเกิดจากการ $w[j]$, $w[j+1]$, $w[j+2]$ และ $w[j+3]$ มาบรรจุลงตามแนวแถวในคอลัมน์ที่ 1, 2, 3 และ 4 ของตารางขนาด 4x4 ตามลำดับ โดยแต่ละแถวจะใช้ข้อมูลจำนวน 1 ไบต์ซึ่งเริ่มจากตำแหน่งแถวที่ 1 เรียงไปจนกระทั่งถึงแถวที่ 4 เมื่อ j คือตำแหน่งล่าสุดที่ยังไม่ถูกนำไปสร้างกุญแจลับ

ขั้นตอนวิธีที่ 4.1 การก่อกำเนิดกุญแจ

```
     INPUT: nk, nr, k
     OUTPUT: w[0], w[1], ⋯,w[4*(nr+1)-1]
1:   i ← 0
2:   While (i < nk) do
3:       w[i] ← นำตำแหน่งไบต์ที่ 4*i+1, 4*i+2, 4*i+3 และ 4*i+4 ของ k มาเชื่อมต่อกัน
4:       i ← i + 1
5:   End While
6:   While (i < 4*(nr + 1)) do
7:       c ← w[i−1]
8:       IF (i mod nk == 0) then
9:           a ← RotWord(c)
10:          b ← SubWord(a)
11:          c ← b ⊕ Rcon[i/nk]
12:      Else IF (nk > 6 and i mod nk == 4) then
13:          c ← SubWord(c)
14:      End IF
15:      w[i] ← w[i-nk] ⊕ c
16:      i ← i + 1
17:  End While
```

ตัวอย่างที่ 4.4 จากกุญแจลับ (*K*) ขนาด 128 บิตที่กำหนดให้ต่อไปนี้

K = ef 74 2a b3 19 26 d4 ea 18 07 41 38 91 44 af e3

จงคำนวณหากุญแจลับที่ใช้สำหรับกระบวนการเข้ารหัส 2 ครั้งแรก

วิธีทำ เนื่องจากโจทย์ถามหากุญแจลับที่ใช้สำหรับกระบวนการเข้ารหัส 2 รอบแรก ความหมายคือต้องใช้กุญแจลับ 2 ค่าสำหรับการเข้ารหัสลับแต่ละครั้ง ดังนั้นจึงต้องคำนวณหากุญแจย่อยจำนวนทั้งหมด 8 ค่าประกอบด้วย *w*[0], *w*[1], *w*[2], ⋯ , *w*[7] ซึ่งสามารถคำนวณหาได้โดยขั้นตอนวิธีที่ 4.1 ดังนี้

1. $i = 0$

ขั้นตอนที่ 2 – 5 จะเป็นการดำเนินการภายในวงวนที่ 1 ดังนี้

เนื่องจาก $i < n_k$ ($i = 0$, $n_k = 4$) ดังนั้น

รอบที่ 1

3. *w*[0] = ef742ab3

4. $i = 1$

รอบที่ 2

3. *w*[1] = 1926d4ea

4. $i = 2$

รอบที่ 3

3. $w[2]$ = 18074138

4. i = 3

รอบที่ 4

3. $w[3]$ = 9144afe3

4. i = 4

ขั้นตอนที่ 6 – 17 จะเป็นการดำเนินการภายในวงวนที่ 2 ดังนี้

เนื่องจาก $i < 4(n_r+1)$ (i = 4, n_r = 10) ดังนั้น

รอบที่ 1

7. $c = w[3]$ = 9144afe3

8. เนื่องจาก $i \bmod n_k = 0$ ดังนั้น

9. a = RotWord(c) = 44afe391

10. b = SubWord(a) = 1b791181

11. $c = b \oplus Rcon[4/4] = b \oplus Rcon[1]$ = 1a791181

12 – 14: ไม่ทำงานในส่วนเงื่อนไขนี้

15. $w[4] = w[0] \oplus c$ = f50d3b32

16. i = 5

รอบที่ 2

7. $c = w[4]$ = f50d3b32

8 – 14: ไม่ทำงานในทั้งสองเงื่อนไขนี้

15. $w[5] = w[1] \oplus c$ = ec2befd8

16. i = 6

รอบที่ 3

7. $c = w[5]$ = ec2befd8

8 – 14: ไม่ทำงานในทั้งสองเงื่อนไขนี้

15. $w[6] = w[2] \oplus c$ = f42caee0

16. i = 7

รอบที่ 4

7. c = w[6] = f42caee0

8 – 14: ไม่ทำงานในทั้งสองเงื่อนไขนี้

15. $w[7] = w[3] \oplus c = 65680103$

16. $i = 8$

สำหรับกรณีที่กุญแจลับมีขนาด 128 บิต จำเป็นต้องสร้างกุญแจลับสำหรับกระบวนการเข้ารหัสและถอดรหัสทั้งหมด 11 ค่า (สร้างกุญแจย่อยทั้งหมด 44 ค่าคือ $w[0]$, $w[1]$, $w[2]$ …,$w[43]$) อย่างไรก็ตามจากตัวอย่างนี้กำหนดให้หากุญแจลับเพียงแค่ 2 ค่าแรก ซึ่งสามารถสร้างได้จากกุญแจย่อยเพียง 8 ค่าคือ $w[0]$, $w[1]$, $w[2]$, …,$w[7]$ ดังนั้นจึงหยุดการดำเนินการไว้ที่รอบที่ 4 ในวงวนที่ 2 ของขั้นตอนวิธีที่ 4.1 และสามารถสร้างกุญแจลับ 2 ค่าแรก (กำหนดให้เป็น K_0 และ K_1) ได้ดังนี้

กุญแจลับค่าที่ 1 (สร้างจาก $w[0]$, $w[1]$, $w[2]$ และ $w[3]$)

K_0 =

ef	19	18	91
74	26	07	43
2a	d4	41	c1
b3	ea	38	e3

กุญแจลับค่าที่ 2 (สร้างจาก $w[4]$, $w[5]$, $w[6]$ และ $w[7]$)

K_1 =

f5	ec	f4	65
0d	2b	2c	68
3b	ef	ae	01
32	d8	e0	03

3. การเข้ารหัสลับเออีเอส

การเข้ารหัสลับแบบเออีเอสคือการนำข้อความต้นฉบับที่มีขนาด 128 บิตที่บรรจุอยู่ภายในเมตริกซ์ขนาด 4x4 มาผ่านกระบวนการทั้งหมด 4 กระบวนการประกอบไปด้วย Substitute Byte, Shift Row, Mix Column และ Add Round Key ซึ่งจะกล่าวถึงแต่ละกระบวนการโดยละเอียดในหัวข้อถัดไป

3.1 การดำเนินการ Substitute Byte

การดำเนินการ Substitute Byte คือการแทนที่ไบต์ข้อมูลเดิมด้วยไบต์ข้อมูลใหม่โดยใช้กล่องเอสกล่องเดียวกันกับที่ใช้สำหรับการดำเนินการ Substitute Word อย่างไรก็ตามการดำเนินการ Substitute Byte คือการแทนที่ข้อมูลขนาด 16 ไบต์ (128 บิต) ซึ่งจะแตกต่างกับการดำเนินการ Substitute Word ซึ่งเป็นการแทนที่กุญแจย่อยที่มีขนาด 4 ไบต์

ตัวอย่างที่ 4.5 จากข้อมูลขนาด 128 บิตที่กำหนดให้ต่อไปนี้

m = 4a 36 a7 91 08 39 7e 43 26 64 35 c2 10 58 25 37

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Substitute Byte

วิธีทำ เริ่มจากนำข้อมูลที่กำหนดมาบรรจุใส่เมตริกซ์ขนาด 4x4 ได้ดังนี้

4a	08	26	10
36	39	64	58
a7	7e	35	25
91	43	c2	37

เมื่อนำข้อมูลข้างต้นมาผ่านกระบวนการ Substitute Byte ผลลัพธ์ทั้งหมดที่ได้ในแต่ละช่องเป็นดังนี้

d6	30	f7	ca
05	12	43	6a
5c	f3	96	3f
81	1a	25	9a

3.2 การดำเนินการ Shift Row

การดำเนินการ Shift Row สำหรับวิทยาการรหัสลับเออีเอสคือการหมุนไบต์ข้อมูลภายในเมตริกซ์ขนาด 4x4 ไปทางซ้าย โดยจำนวนครั้งของการหมุนไบต์ข้อมูลในแต่ละแถวมีความแตกต่างกันดังนี้

แถวที่ 1: ไม่มีการหมุน

แถวที่ 2: หมุนทางซ้าย 1 ตำแหน่ง (1 ไบต์)

แถวที่ 3: หมุนทางซ้าย 2 ตำแหน่ง (2 ไบต์)

แถวที่ 4: หมุนทางซ้าย 3 ตำแหน่ง (3 ไบต์)

ตัวอย่างที่ 4.6 จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

d6	30	f7	ca
05	12	43	6a
5c	f3	96	3f
81	1a	25	9a

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Shift Row

วิธีทำ ดำเนินการหมุนข้อมูลในแต่ละแถวไปทางซ้ายโดย แถวที่ 1 ไม่มีการหมุน แถวที่ 2 หมุนทางซ้าย 1 ตำแหน่ง แถวที่ 3 หมุนทางซ้าย 2 ตำแหน่ง และ แถวที่ 4 หมุนทางซ้าย 3 ตำแหน่ง ได้ผลลัพธ์เป็นดังนี้

d6	30	f7	ca
12	43	6a	05
96	3f	5c	f3
9a	81	1a	25

3.3 การดำเนินการ Mix Column

การดำเนินการ Mix Column คือกระบวนการนำข้อมูลที่ถูกบรรจุภายในเมตริกซ์ขนาด 4x4 มาปรับเปลี่ยนค่าใหม่ โดยนำข้อมูลมาคละกับเมตริกซ์ข้อมูลขนาดเดียวกันซึ่งเป็นค่าคงที่ที่ถูกนำเสนอโดยทีมพัฒนาขั้นตอนวิธีเออีเอสโดยมีค่าเป็นดังนี้

h =

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

กำหนดให้ c แทนเมตริกซ์ผลลัพธ์ที่มีขนาด 4x4 และ c_{ij} แทนผลลัพธ์ซึ่งบรรจุภายในตำแหน่งแถวที่ i และคอลัมน์ที่ j โดยข้อมูลที่ถูกบรรจุอยู่ในแต่ละแถว และคอลัมน์เกิดจากการดำเนินการระหว่างข้อมูล (m) และ h ดังนี้

c_{11} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h และ คอลัมน์ที่ 1 ของ m
c_{21} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 2 ของ h และ คอลัมน์ที่ 1 ของ m
c_{31} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 3 ของ h และ คอลัมน์ที่ 1 ของ m
c_{41} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 4 ของ h และ คอลัมน์ที่ 1 ของ m
c_{12} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h และ คอลัมน์ที่ 2 ของ m
c_{22} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 2 ของ h และ คอลัมน์ที่ 2 ของ m
c_{32} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 3 ของ h และ คอลัมน์ที่ 2 ของ m
c_{42} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 4 ของ h และ คอลัมน์ที่ 2 ของ m
c_{13} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h และ คอลัมน์ที่ 3 ของ m
c_{23} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 2 ของ h และ คอลัมน์ที่ 3 ของ m
c_{33} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 3 ของ h และ คอลัมน์ที่ 3 ของ m
c_{43} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 4 ของ h และ คอลัมน์ที่ 3 ของ m
c_{14} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h และ คอลัมน์ที่ 4 ของ m
c_{24} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 2 ของ h และ คอลัมน์ที่ 4 ของ m
c_{34} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 3 ของ h และ คอลัมน์ที่ 4 ของ m
c_{44} = การคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 4 ของ h และ คอลัมน์ที่ 4 ของ m

โดยตำแหน่งของการนำไบต์ข้อมูลทั้ง 16 ค่ามาบรรจุไว้ในเมตริกซ์ขนาด 4x4 เป็นดังนี้

c =

c_{11}	c_{12}	c_{13}	c_{14}
c_{21}	c_{22}	c_{23}	c_{24}
c_{31}	c_{32}	c_{33}	c_{34}
c_{41}	c_{42}	c_{43}	c_{44}

ตัวอย่างที่ 4.7 จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

m =

d6	30	f7	ca
12	43	6a	05
96	3f	5c	f3
9a	81	1a	25

จงหาผลลัพธ์ c_{11} และ c_{22} ที่เกิดจากกระบวนการ Mix Column

วิธีทำ

1. คำนวณหา c_{11}: พิจารณาเฉพาะกรอบสีดำซึ่งคือการคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h และ คอลัมน์ที่ 1 ของ m ดังนี้

$$c_{11} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} d6 & 30 & f7 & ca \\ 12 & 43 & 6a & 05 \\ 96 & 3f & 5c & f3 \\ 9a & 81 & 1a & 25 \end{bmatrix}$$

c_{11} = (02 x d6) + (03 x 12) + (01 x 96) + (01 x 9a)

$= ((x)\cdot(x^7 + x^6 + x^4 + x^2 + x)) + ((x+1)\cdot(x^4 + x)) + (x^7 + x^4 + x^2 + x) + (x^7 + x^4 + x^3 + x)$

$= (x^8 + x^7 + x^5 + x^3 + x^2) + (x^5 + x^4 + x^2 + x) + x^3 + x^2$

$= x^8 + x^7 + x^2 + x^4 + x$

เนื่องจาก $x^8 + x^7 + x^4 + x^2 + x \bmod x^8 + x^4 + x^3 + x + 1 = x^7 + x^3 + x^2 + 1$

หากนำผลลัพธ์ซึ่งอยู่ในรูปของฟังก์ชันพหุนามเหนือฟิลด์จำกัดข้างต้นมาเขียนรูปแบบโดยย่อได้ดังนี้ $c_{11} = 10001101_2 = 8d$

2. คำนวณหา c_{22}: พิจารณาเฉพาะกรอบสีดำซึ่งคือการคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 2 ของ h และ คอลัมน์ที่ 2 ของ m ดังนี้

$$c_{22} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} d6 & 30 & f7 & ca \\ 12 & 43 & 6a & 05 \\ 96 & 3f & 5c & f3 \\ 9a & 81 & 1a & 25 \end{bmatrix}$$

c_{22} = (01 x 30) + (02 x 43) + (03 x 3f) + (01 x 81)

= $(x^5 + x^4) + ((x)\cdot(x^6 + x + 1)) + ((x + 1)(x^5 + x^4 + x^3 + x^2 + x + 1)) + (x^7 + 1)$

= $x^7 + x^5 + x^4 + 1 + (x^7 + x^2 + x) + (x^6 + x^5 + x^4 + x^3 + x^2 + x + x^5 + x^4 + x^3 + x^2 + x + 1)$

= $x^6 + x^5 + x^4 + x^2 + x$

หากนำผลลัพธ์ซึ่งอยู่ในรูปของฟังก์ชันพหุนามเหนือฟิลด์จำกัดข้างต้นมาเขียนรูปแบบโดยย่อได้ดังนี้ $c_{22} = 01110110_2 = 76$

3.4 การดำเนินการ Add Round Key

การดำเนินการ Add Round Key คือการนำข้อมูล และกุญแจลับสำหรับแต่ละรอบการคำนวณที่บรรจุในเมตริกซ์ขนาด 4x4 และอยู่ตำแหน่งแถว และคอลัมน์ที่ตรงกันมาผ่านกระบวนการเอ็กคลูซีฟออร์ระหว่างตำแหน่งบิตที่ตรงกัน ดังนี้

$$T = K_n \oplus m \tag{4.2}$$

เมื่อ m คือข้อมูลขนาด 128 บิตที่ถูกบรรจุในเมตริกซ์ขนาด 4x4

K_n คือกุญแจลับที่ได้จากขั้นตอนการก่อกำเนิดกุญแจ เพื่อใช้สำหรับกระบวนการเข้ารหัสลับรอบที่ n

T คือผลลัพธ์ที่มีขนาด 128 บิต

กำหนดให้ k_{ij} คือค่าที่อยู่ตำแหน่งแถวที่ i และ คอลัมน์ที่ j ของ K_n

m_{ij} คือค่าที่อยู่ตำแหน่งแถวที่ i และ คอลัมน์ที่ j ของ m

ดังนั้น

$$t_{ij} = k_{ij} \oplus m_{ij} \tag{4.3}$$

เมื่อ t_{ij} คือผลลัพธ์หลังผ่านขั้นตอน Add Round Key ตำแหน่งแถวที่ i และ คอลัมน์ที่ j ของ T

ตัวอย่างที่ 4.8 จากข้อมูล (m) ที่จะนำเข้าสู่ขั้นตอน Add Round Key ดังนี้

m =

4a	54	49	25
28	28	f8	47
32	72	47	86
46	d6	36	23

และกุญแจลับ (k) ดังนี้

k =

55	fc	90	69
44	86	9a	27
33	47	6c	14
22	11	72	88

จงหาผลลัพธ์ t_{33}

วิธีทำ เนื่องจาก m_{33} = 47 = 01000111_2 และ k_{33} = 6c = 01101100_2 ดังนั้น จาก

$$t_{ij} = k_{ij} \oplus m_{ij}$$

$$\begin{array}{ll} (k_{33}) & 01101100 \\ & \qquad\qquad \oplus \\ (m_{33}) & \underline{01000111} \\ (t_{33}) & \underline{00101011} \end{array}$$

หากนำผลลัพธ์ซึ่งอยู่ในรูปของฟังก์ชันพหุนามเหนือฟิลด์จำกัดข้างต้นมาเขียนรูปแบบโดยย่อได้ดังนี้ t_{33} = 00101011_2 = 2b ซึ่งถูกบรรจุในเมตริกซ์ผลลัพธ์ตำแหน่งแถวที่ 3 และคอลัมน์ที่ 3 ดังรูป

T =

		2b	

3.5 ขั้นตอนวิธีการเข้ารหัสลับเออีเอส

ขั้นตอนวิธีการเข้ารหัสลับด้วยวิทยาการรหัสลับเออีเอสเกิดจากการนำการดำเนินการทั้ง 4 วิธีที่กล่าวไว้ในหัวข้อ 3.1 ถึง 3.4 มาประยุกต์ใช้งานร่วมกัน โดยจำนวนรอบการดำเนินการจะขึ้นอยู่กับขนาดของกุญแจที่ใช้งานซึ่งมีทั้งหมด 3 ขนาด กำหนดให้การนับรอบการคำนวณในแต่ละรอบเริ่มนับหลังจากเสร็จสิ้นการดำเนินการ Add Round Key ขั้นตอนวิธีการเข้ารหัสแบบเออีเอสเป็นดังนี้

ขั้นตอนวิธีที่ 4.2 การเข้ารหัสลับเออีเอส

INPUT: ข้อความต้นฉบับขนาด 128 บิต (m), n_r (จำนวนรอบการคำนวณซึ่งมีค่าเป็นดังตารางที่ 4.1) และกุญแจลับที่ได้จากกระบวนการก่อกำเนิดกุญแจดังขั้นตอนวิธี 4.1 (K_0, K_1, K_2, ⋯)

OUTPUT: ข้อความไซเฟอร์ (c)

```
1:  i ← 1
2:  T ← m
3:  T ← T ⊕ K0 // Add Round Key รอบที่ 1
4:  While (i < nr) do
5:      T ← SubByte(T)
6:      T ← ShiftRows(T)
7:      T ← MixColumns(T)
8:      T ← T ⊕ Ki // Add Round Key รอบที่ i+1
9:      i ← i + 1
10: End While
11: T ← SubByte(T)
12: T ← ShiftRows(T)
13: T ← T ⊕ Ki // Add Round Key รอบสุดท้าย
14: c ← T
```

โดยจากขั้นตอนวิธี 4.2 อธิบายความหมายแต่ละฟังก์ชันได้ดังนี้

SubByte(*T*) คือการนำ *T* มาผ่านการดำเนินการ Substitute Byte

ShiftRows(*T*) คือการนำ *T* มาผ่านการดำเนินการ Shift Row

MixColumns(*T*) คือการนำ *T* มาผ่านการดำเนินการ Mix Column

ตัวอย่างที่ 4.9 จากข้อความต้นฉบับ (m) ต่อไปนี้

m =

01	30	31	10
11	04	90	20
10	17	a0	70
21	02	0f	06

จงแสดงวิธีการเข้ารหัสด้วยวิทยาการรหัสลับเออีเอสจำนวน 2 รอบแรกโดยใช้กุญแจลับขนาด 128 บิตจากตัวอย่างที่ 4.4

วิธีทำ เนื่องจากโจทย์ต้องการกระบวนการเข้ารหัสโดยวิทยาการรหัสลับเออีเอสจำนวน 2 รอบแรก ความหมายคือต้องใช้กุญแจลับ 2 ค่า โดยกำหนดให้ใช้กุญแจลับจากตัวอย่างที่ 4.4 ดังนั้นกุญแจลับที่จะถูกนำมาใช้สำหรับกระบวนการเข้ารหัส 2 รอบแรกคือ K_0 และ K_1 ตามลำดับ

1. $i = 1$
2. $T = m$ ดังนั้น

T =

01	30	31	10
11	04	90	20
10	17	a0	70
21	02	0f	06

3. $T = T \oplus K_0$ (รอบที่ 1) ดังนั้น

T =

ef	29	29	81
65	22	97	63
3a	c3	e1	b1
92	e8	37	e5

ขั้นตอนที่ 4 – 10 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก $i < n_r$ ($i = 1$, $n_r = 10$) ดังนั้น

รอบที่ 1

5. T = SubByte(T) ดังนั้น

T =

df	a5	a5	0c
4d	93	88	fb
80	2e	f8	c8
4f	9b	9a	d9

6. T = ShiftRows(T) ดังนั้น

T =

df	a5	a5	0c
93	88	fb	4d
f8	c8	80	2e
d9	4f	9b	9a

7. = MixColumns(T) ดังนั้น

T =

2a	55	5c	7b
28	a2	4b	7e
d7	77	f3	a8
b8	2a	a2	58

8. $T = T \oplus K_1$ (รอบที่ 2) ดังนั้น

$T =$

df	b9	a8	1e
25	89	67	16
ec	98	5d	a9
8a	f2	42	5b

เนื่องจากตัวอย่างที่ 4.9 กำหนดให้แสดงกระบวนการเข้ารหัสโดยใช้รหัสลับเออีเอสจำนวน 2 รอบดังนั้น สรุปได้ว่าข้อความไซเฟอร์เมื่อเสร็จสิ้นรอบที่ 2 มีค่าเป็นดังนี้

df	b9	a8	1e
25	89	67	16
ec	98	5d	a9
8a	f2	42	5b

4. การถอดรหัสลับเออีเอส

การถอดรหัสลับสำหรับวิทยาการรหัสลับเออีเอสมีรูปแบบการดำเนินการคล้ายคลึงกับการเข้ารหัสเป็นอย่างมาก โดยความแตกต่างเป็นเพียงการสลับลำดับการทำงานเนื่องจากเป็นการดำเนินการแบบย้อนกลับโดยกระบวนการที่จำเป็นต้องใช้สำหรับการถอดรหัสยังคงมีทั้งหมด 4 กระบวนการเช่นเดิมดังนี้

4.1 การดำเนินการ Inverse Substitute Byte

การดำเนินการ Inverse Substitute Byte คือกระบวนที่เป็นส่วนผกผันของกระบวนการ Substitute Byte โดยที่วิธีการดำเนินการจะเหมือนกับกระบวนการ Substitute Byte เพียงแต่ใช้กล่องเอสที่แตกต่างกันโดยที่การดำเนินการ Inverse Substitute Byte ใช้กล่องเอสดังตารางที่ 4.4

ตารางที่ 4.4 กล่องเอสสำหรับการดำเนินการ Inverse Substitute Byte

		คอลัมน์ที่															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
แถวที่	**0**	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

ตัวอย่างที่ 4.10 จากข้อมูลขนาด 128 บิตที่กำหนดให้ต่อไปนี้

87 f4 26 42 91 39 9f 2b 72 33 41 95 11 27 83 ab

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Inverse Substitute Byte

วิธีทำ เริ่มจากนำข้อมูลที่กำหนดมาบรรจุใส่เมตริกซ์ขนาด 4x4 ได้ดังนี้

87	91	72	11
f4	39	33	27
26	9f	41	83
42	2b	95	ab

เมื่อนำข้อมูลข้างต้นมาผ่าน กระบวนการ Inverse Substitute Byte ได้ผลการดำเนินการเป็นดังนี้

ea	ac	1e	e3
ba	5b	66	3d
23	6e	f8	41
f6	0b	ad	0e

4.2 การดำเนินการ Inverse Shift Row

การดำเนินการ Inverse Shift Row สำหรับรหัสลับเออีเอสคือ การดำเนินการผกผันของการดำเนินการ Shift Row โดยที่จำนวนครั้งของการหมุนไบต์ข้อมูลภายในเมตริกซ์ขนาด 4x4 จะมีความแตกต่างกับการดำเนินการ Shift Row ดังนี้

แถวที่ 1: ไม่มีการหมุน

แถวที่ 2: หมุนทางซ้าย 3 ตำแหน่ง (3 ไบต์)

แถวที่ 3: หมุนทางซ้าย 2 ตำแหน่ง (2 ไบต์)

แถวที่ 4: หมุนทางซ้าย 1 ตำแหน่ง (1 ไบต์)

ตัวอย่างที่ 4.11 จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

ea	ac	1e	e3
ba	5b	66	3d
23	6e	f8	41
f6	0b	ad	0e

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Inverse Shift Row

วิธีทำ ดำเนินการหมุนข้อมูลในแต่ละแถวไปทางซ้ายโดย แถวที่ 1 ไม่มีการหมุน แถวที่ 2 หมุนทางซ้าย 3 ตำแหน่ง แถวที่ 3 หมุนทางซ้าย 2 ตำแหน่ง และ แถวที่ 4 หมุนทางซ้าย 1 ตำแหน่ง ได้ผลลัพธ์เป็นดังนี้

ea	ac	1e	e3
3d	ba	5b	66
f8	41	23	6e
0b	ad	0e	f6

4.3 การดำเนินการ Inverse Mix Column

การดำเนินการ Inverse Mix Column คือการนำข้อมูลมาคละกับเมตริกซ์ข้อมูลขนาด 4x4 ซึ่งเป็นค่าคงที่ที่ถูกนำเสนอโดยทีมพัฒนาขั้นตอนวิธีเออีเอสโดยมีค่าเป็นดังนี้

h^{-1}=

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

โดยขั้นตอนการดำเนินการคละข้อมูลจะเหมือนกับการดำเนินการ Mix Column เพียงแต่การดำเนินการ Inverse Mix Column จะใช้ h^{-1} เป็นเมตริกซ์ค่าคงที่แทน h

ตัวอย่างที่ 4.12 จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

ea	ac	1e	e3
3d	ba	5b	66
f8	41	23	6e
0b	ad	0e	f6

จงหาผลลัพธ์ r_{11} (เมื่อ r_{11} แทนผลลัพธ์ที่เกิดจากการดำเนินการ Inverse Mix Column ในตำแหน่งแถวที่ 1 และคอลัมน์ที่ 1)

วิธีทำ

1. คำนวณหา r_{11}: พิจารณาเฉพาะกรอบสีดำซึ่งคือการคูณเหนือฟิลด์จำกัดระหว่างแถวที่ 1 ของ h^{-1} และ คอลัมน์ที่ 1 ของข้อความ ดังนี้

$$c_{11} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \times \begin{bmatrix} ea & ac & 1e & e3 \\ 3d & ba & 5b & 66 \\ f8 & 41 & 23 & 6e \\ 0b & ad & 0e & f6 \end{bmatrix}$$

c_{11} = (0e x ea) + (0b x 3d) + (0d x f8) + (09 x 0b)

= $((x^3 + x^2 + x + 1)\cdot(x^7 + x^6 + x^5 + x^3 + x)) + ((x^3 + x + 1)\cdot(x^5 + x^4 + x^3 + x^2 + 1)) +$
$((x^3 + x^2 + 1)\cdot(x^7 + x^6 + x^5 + x^4 + x^3)) + ((x^3 + 1)\cdot(x^3 + x + 1))$

= $(x^{10} + x^8 + x^7 + x^6 + x^2 + x) + (x^8 + x^7 + x^5 + x^3 + x^2 + x + 1) +$
$(x^{10} + x^7 + x^6 + x^4 + x^3) + (x^6 + x^4 + x + 1)$

= $x^7 + x^6 + x^5 + x$

หากนำผลลัพธ์ซึ่งอยู่ในรูปของฟังก์ชันพหุนามเหนือฟิลด์จำกัดข้างต้นมาเขียนรูปแบบโดยย่อได้ดังนี้ $c_{11} = 11100010_2 = e2$

4.4 การดำเนินการ Add Round Key

สำหรับการดำเนินการ Add Round Key จะมีขั้นตอนการดำเนินการที่มีลักษณะเดียวกันกับการเข้ารหัสลับ โดยการนำข้อมูล และกุญแจลับสำหรับแต่ละรอบการคำนวณที่บรรจุในเมตริกซ์ขนาด 4x4 และอยู่ตำแหน่งแถว และคอลัมน์ที่ตรงกันมาผ่านกระบวนการเอ็กคลูซีฟออร์ระหว่างตำแหน่งบิตที่ตรงกัน

4.5 ขั้นตอนวิธีการถอดรหัสลับเออีเอส

ขั้นตอนวิธีการถอดรหัสลับสำหรับวิทยาการรหัสลับเออีเอสเกิดจากการนำการดำเนินการทั้ง 4 วิธีทีกล่าวไว้ในหัวข้อ 4.1 ถึง 4.4 มาประยุกต์ใช้งานร่วมกัน โดยจำนวนรอบการดำเนินการจะขึ้นอยู่กับขนาดของกุญแจที่ใช้งานซึ่งมีทั้งหมด 3 ขนาดเช่นเดียวกันกับขั้นตอนวิธีการเข้ารหัสลับ อย่างไรก็ตามลำดับการทำงานของการดำเนินการแต่ละชนิดจะแตกต่างกับการเข้ารหัสลับ กำหนดให้การนับรอบการคำนวณในแต่ละรอบเริ่มนับหลังจากเสร็จสิ้นการดำเนินการ Add Round Key ขั้นตอนวิธีการถอดรหัสลับแบบเออีเอสเป็นดังนี้

ขั้นตอนวิธีที่ 4.3 การถอดรหัสลับเออีเอส

```
INPUT: ข้อความต้นไซเฟอร์ขนาด 128 บิต (c), nr (จำนวนรอบการคำนวณซึ่งมีค่าเป็นดังตารางที่ 4.1)
และ กุญแจลับที่ได้จากกระบวนการก่อกำเนิดกุญแจดังขั้นตอนวิธี 4.1 (K0, K1, K2, ⋯)
OUTPUT: ข้อความต้นฉบับ (T)
1:  i ← nr
2:  T ← c
3:  T ← T ⊕ Ki // Add Round Key รอบที่ 1
4:  i ← i - 1
5:  While (i > 0) do
6:      T ← InvShiftRows(T)
7:      T ← InvSubByte(T)
8:      T ← T ⊕ Ki // Add Round Key รอบที่ i+1
9:      T ← InvMixColumns(T)
10:     i ← i - 1
11: End While
12: T ← InvShiftRows(T)
13: T ← InvSubByte(T)
14: T ← T ⊕ Ki // Add Round Key รอบสุดท้าย
15: m ← T
```

โดยจากขั้นตอนวิธี 4.3 อธิบายความหมายของ *T* และแต่ละฟังก์ชันได้ดังนี้

T คือ ค่าข้อมูลขนาด 128 บิตที่ถูกบรรจุไว้ภายในเมตริกซ์ขนาด 4x4

InvSubByte(*T*) คือการนำ *T* มาผ่านการดำเนินการ Inverse Substitute Byte

ShiftRows(*T*) คือการนำ *T* มาผ่านการดำเนินการ Inverse Shift Row

MixColumns(*T*) คือการนำ *T* มาผ่านการดำเนินการ Inverse Mix Column

5. บทสรุปสาระสำคัญ

เนื่องจากเทคโนโลยีที่มีความเจริญก้าวหน้าอย่างรวดเร็วส่งผลให้วิทยาการรหัสลับดีอีเอสซึ่งใช้กุญแจลับที่มีขนาดเพียง 64 บิตไม่มีความปลอดภัยอีกต่อไป ถึงแม้ว่าจะมีการเพิ่มความแข็งแกร่งโดยใช้กระบวนการเข้ารหัสแบบหลายชั้นแต่โครงสร้างของขั้นตอนวิธียังคงเป็นเช่นเดิม การพัฒนาต่อยอดเพื่อให้วิทยาการรหัสลับดีอีเอสมีประสิทธิภาพที่สูงขึ้นจึงเป็นไปได้ยาก ต่อมาในปี ค.ศ. 2001 วิทยาการรหัสลับเออีเอสจึงถูกเสนอเพื่อแทนที่วิทยาการรหัสลับดีอีเอส โดยสามารถประยุกต์ใช้ได้กับกุญแจลับที่มีทั้งหมด 3 ขนาดให้เลือกใช้งานคือ 128, 192 และ 256 บิต ซึ่งมีขนาดที่ใหญ่มหาศาลมากหากเปรียบเทียบกับกุญแจลับสำหรับวิทยาการรหัสลับดีอีเอส ดังนั้นวิทยาการรหัสลับแบบเออีเอสจึงมีประสิทธิภาพที่สูงกว่าวิทยาการรหัสลับดีอีเอส และยังคงมีการใช้งานอยู่ในปัจจุบัน อย่างไรก็ตามขนาดกุญแจลับที่แตกต่างกันส่งผลให้จำนวนรอบของกระบวนการเข้ารหัสลับ และการถอดรหัสลับมีความแตกต่างกันด้วยเช่นกัน

สำหรับข้อความต้นฉบับที่จะนำเข้าสู่กระบวนการเข้ารหัสลับ และข้อความไซเฟอร์ที่จะนำเข้าสู่กระบวนการถอดรหัสลับต้องมีขนาดคงที่คือ 128 บิต โดยหากจำนวนบิตมีขนาดที่น้อยกว่า 128 บิต จะต้องทำการเติมบิต 0 ที่ตำแหน่งด้านหน้าซึ่งไม่ทำให้ค่าเกิดการเปลี่ยนแปลงจนกระทั่งครบตามจำนวนที่กำหนด ในทางกลับกันหากจำนวนบิตเกินกว่าที่กำหนดจำเป็นต้องตัดแบ่งข้อความต้นฉบับหรือข้อความไซเฟอร์ออกเป็นกลุ่มๆ ละ 128 บิตและนำแต่ละกลุ่มไปผ่านกระบวนการเข้ารหัสลับหรือกระบวนการถอดรหัสลับแล้วจึงนำผลลัพธ์ทั้งหมดมาประกอบกัน

แบบฝึกหัดท้ายบท
บทที่ 4

1. ข้อความต้นฉบับที่จะนำมาเข้ารหัสลับด้วยรหัสเออีเอสมีขนาดกี่บิต
2. กุญแจลับสำหรับรหัสลับเออีเอสมีทั้งหมดกี่ขนาด และมีขนาดเท่าไร
3. ฟังก์ชันพหุนามไม่ลดรูปสำหรับมาตรฐานเออีเอสมีค่าเท่าไร
4. หากเลือกใช้งานรหัสเออีเอสโดยใช้กุญแจลับขนาด 196 บิต จำเป็นต้องดำเนินการเข้ารหัสลับทั้งหมดเป็นจำนวนกี่รอบ
5. ข้อมูล 7e หากนำไปผ่านกล่องเอสสำหรับการดำเนินการ Substitute Word จะมีค่าเป็นเท่าไร
6. กำหนดให้กุญแจย่อย $w[0]$ = 48269143 จงหาผลลัพธ์หลังจากผ่าน Rotate Word
7. จากขั้นตอนวิธีที่ 4.1 กำหนดให้ b = 227a8391 และ i = 8 จงคำนวณหา c
8. จากข้อความต้นฉบับขนาด 128 บิตต่อไปนี้

a1	d8	58	2f
2f	31	6c	51
36	5b	7e	85
49	42	3a	97

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Substitute Byte

9. จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

54	27	55	14
3e	83	17	51
61	62	1a	26
4f	41	3b	49

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Shift Row

10. จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ดังตัวอย่างที่ 4.7 จงหา c_{23}

11. จากข้อมูล m และ k ดังตัวอย่างที่ 4.8 จงหา t_{21}

12. ข้อมูล 28 หากนำไปผ่านกล่องเอสสำหรับการดำเนินการ Inverse Substitute Word จะมีค่าเป็นเท่าไร

13. จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

25	27	93	76
38	62	a3	63
a4	81	47	e7
59	d1	11	54

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Inverse Substitute Byte

14. จากข้อมูลภายในเมตริกซ์ขนาด 4x4 ที่กำหนดให้ต่อไปนี้

26	68	42	87
f1	54	91	65
23	11	22	4c
71	33	67	28

จงหาผลลัพธ์หลังจากผ่านกระบวนการ Inverse Shift Row

15. การดำเนินการถอดรหัสลับด้วยรหัสเออีเอสจำนวน 14 รอบ แสดงว่ากุญแจลับมีขนาดเท่าไร

บทที่ 5
ทฤษฎีจำนวน และขั้นตอนวิธีสำหรับวิทยาการรหัสลับ

ในบทนี้จะกล่าวถึงทฤษฎีจำนวน [4] และขั้นตอนวิธีที่สำคัญที่จำเป็นต้องนำมาใช้สำหรับแก้ปัญหาวิทยาการรหัสลับทั้งแบบสมมาตรและแบบอสมมาตร อย่างไรก็ตามทฤษฎีจำนวนที่เคยถูกนำมาใช้สำหรับแก้ปัญหาวิทยาการรหัสลับแบบสมมาตรดังที่ได้กล่าวแล้วในบทที่ 1 เช่น เลขคณิตมอดูลาร์ สมภาค หรือ ขั้นตอนวิธียุคลิค เป็นต้น ยังคงจำเป็นต้องถูกนำมาใช้ร่วมกับวิทยาการรหัสลับแบบอสมมาตรด้วยเช่นกัน

1. จำนวนเฉพาะ (Prime Number)

จำนวนเฉพาะคือจำนวนเต็มใดๆ ที่มีตัวเลขที่เป็นจำนวนเต็มบวกเพียง 2 จำนวนเท่านั้นที่สามารถนำมาหารจำนวนดังกล่าวได้ลงตัว ประกอบด้วย 1 และค่าของตัวเอง ยกตัวอย่างเช่น 2 เป็นจำนวนเฉพาะเนื่องจากมีเพียง 1 และ 2 ที่สามารถนำมาหาร 2 ได้ลงตัว หรือ 7 เป็นจำนวนเฉพาะเนื่องจากมีเพียง 1 และ 7 ที่สามารถนำมาหาร 7 ได้ลงตัว เป็นต้น ในทางกลับกันจำนวนเต็มอื่นๆ ที่ไม่ตรงกลับคุณสมบัติดังกล่าวเรียกว่าจำนวนประกอบ (Composite Number) โดยจำนวนเฉพาะจะถูกนำมาใช้สำหรับขั้นตอนการก่อกำหนดกุญแจสำหรับขั้นตอนวิธีบางกลุ่มที่ถูกจัดอยู่ในกลุ่มของวิทยาการรหัสลับแบบอสมมาตร

อย่างไรก็ตาม 2 คือจำนวนเฉพาะเพียงค่าเดียวที่เป็นจำนวนเต็มบวกคู่ สำหรับจำนวนเฉพาะที่เป็นจำนวนเต็มบวกค่าอื่นๆ เป็นจำนวนเต็มบวกคี่ทั้งหมด

2. การตรวจสอบจำนวนเฉพาะ

การพิจารณาจำนวนเต็มใดๆ ว่าเป็นจำนวนเฉพาะ หรือจำนวนประกอบสามารถดำเนินการได้ง่ายหากตัวเลขดังกล่าวมีขนาดเล็ก แต่หากจำนวนเต็มที่ถูกนำมาพิจารณามีขนาดใหญ่ส่งผลให้ใช้เวลามหาศาลเมื่อดำเนินการพิจารณาโดยตรง อย่างไรก็ตามมีขั้นตอนวิธีที่ใช้สำหรับการตรวจสอบจำนวนเฉพาะถูกนำเสนอมาหลายวิธี โดยแต่ละขั้นตอนวิธีจะมีประสิทธิภาพแตกต่างกันออกไป ดังนี้

2.1 ขั้นตอนวิธีทดลองหาร (Trail Division Algorithm)

ขั้นตอนวิธีการทดลองหาร [43] เป็นขั้นตอนวิธีที่ได้รับความนิยมมากสำหรับการตรวจสอบจำนวนเฉพาะ หรือจำนวนประกอบ เนื่องจากเป็นวิธีที่ง่าย และมีการใช้งานตั้งแต่การเรียนใน

ระดับพื้นฐาน กำหนดให้ n คือจำนวนเต็มบวกคี่ใด ๆ ที่มีค่ามากกว่า 3 และเป็นจำนวนที่จะถูกนำมาตรวจสอบว่าเป็นจำนวนเฉพาะหรือจำนวนประกอบด้วยขั้นตอนวิธีทดลองหาร หลักการคือนำจำนวนเต็มบวกคี่ที่มีค่าน้อยที่สุดคือ 3 (ไม่พิจารณา 1 เนื่องจากเป็นจำนวนที่สามารถหารจำนวนเต็มทุกค่าได้ลงตัว) มาทดลองหาร n ซึ่งหากผลหารลงตัวจะสรุปได้ว่า n เป็นจำนวนประกอบทันที ในทางกลับกันหากผลหารไม่ลงตัวจะต้องเพิ่มค่าตัวหารขึ้นครั้งละ 2 ค่าเพื่อนำไปทดลองหารอีกครั้ง โดยจะดำเนินการเช่นนี้จนกระทั่งพบตัวหารที่สามารถหาร n ได้ลงตัวได้ว่า n เป็นจำนวนประกอบ หรือตัวหารมีค่ามากกว่า $\sqrt{n}$ (รากที่สองของ n) ได้ว่า n เป็นจำนวนเฉพาะจึงจะหยุดทำงาน

จากหลักการข้างต้นสังเกตได้ว่าตัวหารจะถูกเพิ่มค่าขึ้นครั้งละ 2 เนื่องจากขั้นตอนวิธีทดลองหารจะตัดตัวหารที่เป็นจำนวนเต็มคู่ออกจากการคำนวณเพื่อลดเวลาการคำนวณเพราะว่าจำนวนเฉพาะทั้งหมด (ยกเว้น 2) เป็นจำนวนเต็มคี่ ซึ่งหากตัวหารเป็นจำนวนเต็มคู่จะไม่สามารถหารตัวตั้งที่เป็นจำนวนเต็มคี่ได้ลงตัวอย่างแน่นอน ดังนั้นในกรณีที่ตัวหารปัจจุบันไม่สามารถนำไปหารตัวตั้งได้ลงตัวจะสามารถเพิ่มค่าได้ครั้งละ 2 ค่าเพื่อให้ตัวหารตัวถัดไปยังคงเป็นจำนวนเต็มบวกคี่ สำหรับขั้นตอนวิธีทดลองหารเป็นดังนี้

ขั้นตอนวิธีที่ 5.1 การทดลองหาร

```
     INPUT: n
     OUTPUT: ชนิดของ n (จำนวนเฉพาะหรือจำนวนประกอบ)
1:   x ← 3
2:   y ← n mod x
3:   While ((x < √n ) and (y ≠ 0)) do
4:       x ← x + 2
5:       y ← n mod x
6:   End While
7:
8:   IF (y ≠ 0) then
9:       n is prime number
10:  Else
11:      n is composite number
     End IF
```

ตัวอย่างที่ 5.1 จงตรวจสอบ 53 เป็นจำนวนเฉพาะ หรือจำนวนประกอบ

วิธีทำ จากขั้นตอนวิธีทดลองหารได้ว่า $n = 53$ และ $\sqrt{n} = 7.28$ โดยเริ่มกระบวนการดังนี้

1. $x = 3$
2. $y = 53 \bmod 3 = 2$

ขั้นตอนที่ 3 – 6 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก 3 < 7.28 (เป็นจริง) และ 2 ≠ 0 (เป็นจริง) จากเงื่อนไขทางตรรกศาสตร์ (จริง และ จริง ได้ผลลัพธ์เป็นจริง) จึงเข้าสู่รอบการทำงาน

รอบที่ 1:

4. $x = 3 + 2 = 5$

5. $y = 53 \bmod 5 = 3$

เนื่องจาก 5 < 7.28 (เป็นจริง) และ 3 ≠ 0 (เป็นจริง) ได้ว่า

รอบที่ 2:

4. $x = 5 + 2 = 7$

5. $y = 53 \bmod 7 = 4$

เนื่องจาก 7 < 7.28 (เป็นจริง) และ 4 ≠ 0 (เป็นจริง) ได้ว่า

รอบที่ 3:

4. $x = 7 + 2 = 9$

5. $y = 53 \bmod 7 = 8$

เนื่องจาก 9 < 7.28 (เป็นเท็จ) และ 8 ≠ 0 (เป็นจริง) จากเงื่อนไขทางตรรกศาสตร์ (เท็จ และ จริง ได้ผลลัพธ์เป็นเท็จ) จึงออกจากรอบการทำงาน

เงื่อนไขที่อยู่ในระหว่างขั้นตอนที่ 7 – 11

เนื่องจาก $y = 8 \neq 0$ จึงสรุปได้ว่า 53 **<u>เป็นจำนวนเฉพาะ</u>**

ตัวอย่างที่ 5.2 จงตรวจสอบ 15 เป็นจำนวนเฉพาะ หรือจำนวนประกอบ

วิธีทำ จากขั้นตอนวิธีทดลองหารได้ว่า $n = 15$ และ $\sqrt{n} = 3.87$ โดยเริ่มกระบวนการดังนี้

1. $x = 3$
2. $y = 15 \bmod 3 = 0$

ขั้นตอนที่ 3 – 6 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก 3 < 3.287 (เป็นจริง) และ 0 ≠ 0 (เป็นเท็จ) จากเงื่อนไขทางตรรกศาสตร์ (จริง และ เท็จ ได้ผลลัพธ์เป็นเท็จ) จึงไม่เข้าสู่รอบการทำงาน

เงื่อนไขที่อยู่ในระหว่างขั้นตอนที่ 7 – 11

เนื่องจาก $y = 0$ จึงสรุปได้ว่า 15 **<u>เป็นจำนวนประกอบ</u>**

เนื่องจากจำนวนเต็มที่นำมาใช้สำหรับการทดลองหารมีค่าเริ่มต้นที่ 3 และค่าที่เป็นไปได้สูงสุด คือ $\sqrt{n}$ จึงสรุปได้ว่าขั้นตอนวิธีทดลองหารใช้เวลาที่สูงสุดคือ $O(\sqrt{n})$ โดยกรณีที่จำนวนเต็มที่จะ

นำมาตรวจสอบเป็นจำนวนเฉพาะจะใช้เวลา $O(\sqrt{n})$ อย่างแน่นอน ในทางกลับกันหากจำนวนเต็มดังกล่าวเป็นจำนวนประกอบเวลาที่ใช้สำหรับการตรวจสอบขึ้นอยู่กับตัวประกอบที่มีขนาดเล็กที่สุดของจำนวนดังกล่าว โดยที่หากตัวประกอบค่าดังกล่าวนี้มีขนาดเล็กมากส่งผลให้เวลาที่ใช้สำหรับตรวจสอบต่ำ แต่หากตัวประกอบที่มีขนาดเล็กที่สุดมีค่าสูงส่งผลให้เวลาที่ใช้สำหรับตรวจสอบสูงขึ้นตามไปด้วย

2.2 ทฤษฎีบทเล็กของแฟร์มาต์ (Fermat's Little Theorem)

ปีแยร์ เดอร์ แฟร์มาต์ (Pierre de Fermat) [6] นักคณิตศาสตร์ชาวฝรั่งเศสผู้ซึ่งได้ค้นพบทฤษฎีทางคณิตศาสตร์ที่สำคัญมากมาย โดยในช่วงปี ค.ศ.1640 ได้ค้นพบทฤษฎีที่สำคัญเกี่ยวกับจำนวนเฉพาะ ดังนี้

บทตั้งที่ 5.1 กำหนดให้ p คือจำนวนเฉพาะและ $a \in \mathbb{Z}$ โดยที่ $\gcd(a, p) = 1$ แล้วได้ว่าส่วนตกค้างน้อยที่สุดของ $\{a, 2a, 3a, \cdots, (p-1)a\}$ มอดูโล p คือ $\{1, 2, 3, \cdots, p-1\}$

พิสูจน์ กำหนดให้ $1 \leq i \leq p-1$, การพิสูจน์เริ่มจากกำหนดให้ $ia \equiv 0 \bmod p$ ได้ว่า $p \mid a$ หรือ $p \mid i$ แต่เนื่องจาก $\gcd(p, a) = 1$ และ $p > i$ ดังนั้น $p \nmid a$ และ $p \nmid i$ ซึ่งเกิดข้อขัดแย้ง จึงสรุปได้ว่า $ia \neq 0 \bmod p$

ต่อมากำหนดให้ $ia \equiv ja \bmod p$ เมื่อ $1 \leq i, j \leq p-1$, แล้ว $p \mid (ia - ja)$ หรือ $p \mid a(i-j)$ แต่เนื่องจาก $\gcd(a, p) = 1$ ซึ่งความหมายคือ $p \nmid a$ ดังนั้น $p \mid (i-j)$ หรือ $i \equiv j \bmod p$ แต่เนื่องจาก $i, j < p$ ดังนั้นทั้งสองค่านี้เป็นส่วนตกค้างน้อยที่สุดจึงได้ว่า $i = j$ และสรุปได้ว่าไม่มีส่วนตกค้างน้อยที่สุดคู่ใดของ $\{a, 2a, 3a, \cdots, (p-1)a\}$ ที่สมภาคกันภายใต้การมอดูโล p หรือ ส่วนตกค้างน้อยที่สุดของ $\{a, 2a, 3a, \cdots, (p-1)a\}$ มอดูโล p คือ $\{1, 2, 3, \cdots, p-1\}$ □

ทฤษฎีบทที่ 5.2 กำหนดให้ p เป็นจำนวนเฉพาะ และ a เป็นจำนวนเต็มบวกใดๆ ที่หารด้วย p ไม่ลงตัว ได้ว่า

$$a^{p-1} \equiv 1 \bmod p$$

พิสูจน์ จากบทตั้งที่ 5.1 ได้ว่า

$$a \times 2a \times 3a \times \cdots \times (p-1)a \equiv 1 \times 2 \times 3 \times \cdots \times (p-1) \bmod p$$

ดังนั้น $1 \times 2 \times 3 \times \cdots \times (p-1) \times a^{p-1} \equiv 1 \times 2 \times 3 \times \cdots \times (p-1) \bmod p$

หรือ $a^{p-1} \equiv 1 \bmod p$ □

จากทฤษฎีบทที่ 5.2 หากผลลัพธ์ของ $a^{p-1} \bmod p$ มีค่าไม่เท่ากับ 1 เมื่อ $p \nmid a$ สรุปได้ทันทีว่าค่า p ดังกล่าวไม่ใช่จำนวนเฉพาะ ในทางกลับกันหากผลลัพธ์ที่ได้มีค่าเท่ากับ 1 สามารถสรุปได้ว่าค่า p เป็นจำนวนเฉพาะอย่างแน่นอน

ตัวอย่างที่ 5.3 จงแสดงให้เห็นว่า 9 ไม่ใช่จำนวนเฉพาะ

วิธีทำ จากทฤษฎีบทที่ 5.2 ได้ว่า

กำหนดให้ $p = 9$ และ เลือก $a = 2$ ($9 \nmid 2$) ดังนั้น

$$2^{9-1} \bmod 9 = 2^8 \bmod 9$$
$$= 256 \bmod 9$$
$$= 4$$

เนื่องจาก $2^8 \bmod 9 = 4$ จึงสรุปได้ว่า 9 ไม่ใช่จำนวนเฉพาะอย่างแน่นอน

ตัวอย่างที่ 5.4 จงแสดงให้เห็นว่า 11 เป็นจำนวนเฉพาะ

วิธีทำ จากทฤษฎีบทที่ 5.2 ได้ว่า

กำหนดให้ $p = 11$ และ เลือก $a = 3$ ($11 \nmid 3$) ดังนั้น

$$3^{11-1} \bmod 11 = 3^{10} \bmod 11$$
$$= 59049 \bmod 11$$
$$= 1$$

เนื่องจาก $3^{10} \bmod 11 = 1$ จึงสรุปได้ว่า 11 เป็นจำนวนเฉพาะอย่างแน่นอน

นอกเหนือจากการนำทฤษฎีบทเล็กของแฟร์มาต์มาใช้สำหรับการตรวจสอบความเป็นจำนวนเฉพาะหรือจำนวนประกอบแล้ว ยังสามารถนำทฤษฎีดังกล่าวมาเป็นเครื่องมือช่วยสำหรับการคำนวณโดยเฉพาะอย่างยิ่งเมื่อเลขยกกำลังมีขนาดใหญ่มหาศาลและตัวหารเป็นจำนวนเฉพาะ นอกเหนือจากนั้นยังเป็นทฤษฎีพื้นฐานสำหรับทฤษฎีบทของออยเลอร์ที่สามารถนำไปประยุกต์ใช้กับวิทยาการรหัสลับแบบอสมมาตร

ตัวอย่างที่ 5.5 จงคำนวณหาผลลัพธ์ของ $7^{3603} \bmod 37$ เมื่อ 37 คือจำนวนเฉพาะ

วิธีทำ เนื่องจากตัวอย่างนี้เลขยกกำลังมีขนาดใหญ่ ซึ่งหากคำนวณผลลัพธ์ของ 7^{3603} โดยตรงจะได้ตัวเลขที่มีขนาดใหญ่มหาศาล และใช้เวลาการคำนวณที่นานมาก อย่างไรก็ตามจากตัวอย่างนี้สามารถใช้ทฤษฎีบทเล็กของแฟร์มาต์มาช่วยพิจารณาหาคำตอบได้ ดังนี้

เนื่องจาก 37 เป็นจำนวนเฉพาะ และ $7 \nmid 37$ โดยจากทฤษฎีบทเล็กของแฟร์มาต์สรุปได้ว่า

$$7^{36} \bmod 37 = 1$$

ดังนั้น

$$7^{3603} \bmod 37 = 7^{3600}\, 7^3 \bmod 37$$

จากทฤษฎีบทที่ 1.4

$$\begin{aligned} &= [(7^{3600} \bmod 37)(7^3 \bmod 37)] \bmod 37 \\ &= [((7^{36})^{100} \bmod 37)(7^3 \bmod 37)] \bmod 37 \\ &= [((1)^{100} \bmod 37)(7^3 \bmod 37)] \bmod 37 \\ &= 1 \times (7^3 \bmod 37)) \bmod 37 \\ &= (343 \bmod 37) \bmod 37 \\ &= 10 \bmod 37 \\ &= 10 \end{aligned}$$

จากตัวอย่างที่ 5.5 สังเกตได้ว่าสามารถใช้ทฤษฎีบทเล็กของแฟร์มาต์สำหรับแก้ปัญหาเลขคณิตมอดูลาร์ที่จำเป็นต้องคำนวนเลขยกกำลังที่มีขนาดใหญ่มหาศาลได้ ซึ่งจากตัวอย่างไม่จำเป็นต้องหาผลลัพธ์ของ 7^{3603} โดยตรง แต่ใช้ทฤษฎีบทเล็กของแฟร์มาต์มาประยุกต์ใช้แทนซึ่งช่วยลดรอบการคำนวณและเวลาการคำนวณลงได้เป็นอย่างมาก อย่างไรก็ตามหากค่ามอดูลัสไม่ใช่จำนวนเฉพาะจะไม่สามารถใช้ทฤษฎีดังกล่าวนี้ได้

2.3 การทดสอบมิลเลอร์ - ราบิน (Miller – Rabin Test)

การทดสอบมิลเลอร์ – ราบิน [40], [41] เป็นอีกเทคนิคหนึ่งที่สามารถนำมาใช้ตรวจสอบการเป็นจำนวนเฉพาะ หรือจำนวนประกอบของจำนวนเต็มได้ ซึ่งเกิดจากการปรับปรุงทฤษฎีบทเล็กของแฟร์มาต์ โดยขั้นตอนวิธีการทดสอบมิลเลอร์ – ราบินเหมาะสำหรับนำมาใช้ตรวจสอบจำนวนเต็มที่มีขนาดใหญ่ ซึ่งหากจำนวนเต็มดังกล่าวมีขนาดเล็กการตรวจสอบด้วยขั้นตอนวิธีการทดลองหารจะมีประสิทธิภาพมากกว่า

กำหนดให้ $n \in \mathbb{Z}^+$ และ

$$s = \max\{r \in \mathbb{N} \mid 2^r \bmod n - 1 = 0\}$$

หรือกล่าวได้ว่า s คือจำนวนเต็มบวกที่มีค่ามากที่สุดที่ทำให้ $2^s \mid n - 1$

และหลังจากพบค่า s แล้วสามารถคำนวณหาค่า d ได้จาก

$$d = \frac{n-1}{2^s}$$

เนื่องจาก s คือค่าสูงสุดที่ทำให้ $2^s \mid n\text{-}1$ จึงได้ว่า d เป็นจำนวนเต็มคี่อย่างแน่นอน โดยค่า r, d และ s จะถูกนำมาใช้ในทฤษฎีบทที่ 5.3 เพื่อตรวจสอบความเป็นจำนวนเฉพาะ หรือจำนวนประกอบด้วยวิธีการทดสอบมิลเลอร์ – ราบิน

ทฤษฎีบทที่ 5.3 กำหนดให้ $a, n \in \mathbb{Z}^+$ เป็นจำนวนเฉพาะสัมพัทธ์ต่อกันได้ว่า n อาจจะเป็นจำนวนเฉพาะได้ก็ต่อเมื่อหนึ่งในสองสมการต่อไปนี้ เป็นจริง

$$a^d \equiv 1 \bmod n \tag{5.1}$$

$$a^{2^r d} \equiv -1 \bmod n \tag{5.2}$$

เมื่อ $r = \{0, 1, 2, \cdots, s - 1\}$

จากทฤษฎีบทที่ 5.3 ถ้าหาก n เป็นจำนวนเฉพาะ สมการ (5.1) หรือ (5.2) จะเป็นจริงสมการใดสมการหนึ่งอย่างแน่นอน แต่หากทั้งสองสมการเป็นเท็จ จะได้ว่าค่า n เป็นจำนวนประกอบอย่างแน่นอน ในทางกลับกันสมมติยังไม่ทราบว่าค่า n เป็นจำนวนเฉพาะ หรือจำนวนประกอบ และจากการตรวจสอบทั้งสองสมการพบว่าผลลัพธ์ที่เป็นจริงเกิดขึ้นในหนึ่งสมการข้างต้นแล้วจะยังไม่สามารถสรุปได้ว่าค่า n เป็นจำนวนเฉพาะ

อย่างไรก็ตามหากค่า a ที่เลือกมาพิจารณาไม่เป็นจำนวนเฉพาะสัมพัทธ์กับ n เมื่อ $1 < a < n\text{-}1$ สรุปได้ทันทีว่า n เป็นจำนวนประกอบเนื่องจาก $\gcd(a, n) > 1$ แต่หาก a และ n เป็นจำนวนเฉพาะสัมพัทธ์ต่อกันและพบคำตอบที่เป็นจริงในสมการใดสมการหนึ่ง ความเป็นไปได้ที่ n เป็นจำนวนประกอบคือ 0.25 ดังนั้นหากเลือก a ที่ไม่ซ้ำกันมา t ตัวที่เป็นจำนวนเฉพาะสัมพัทธ์กับ n และพบผลลัพธ์ที่เป็นจริงในสมการ (5.1) หรือ สมการที่ (5.2) สำหรับทุกค่า a ที่ถูกเลือก กล่าวได้ว่าความเป็นไปได้ที่ n เป็นจำนวนประกอบคือ 0.25^t

สมมติว่าเลือก a มาทั้งหมด 4 ค่า และทั้ง 4 ค่านี้มีคำตอบในสมการ (5.1) หรือ สมการที่ (5.2) กล่าวได้ว่าความเป็นไปได้ที่ n เป็นจำนวนประกอบคือ $0.25^4 = 0.00390625$ ซึ่งเป็นค่าที่น้อยมาก หรือกล่าวอีกมุมหนึ่งคือ ความน่าจะเป็นที่ n จะเป็นจำนวนเฉพาะสูงถึง 0.99609375

ดังนั้นเพื่อทำให้มั่นใจว่าค่า n ที่นำมาตรวจสอบเป็นจำนวนเฉพาะหรือไม่ ควรตรวจสอบร่วมกับค่า a ที่เป็นจำนวนเฉพาะสัมพัทธ์กับ n อย่างน้อย 4 ค่า

ตัวอย่างที่ 5.6 จงแสดงว่า $n = 77$ เป็นจำนวนประกอบ

วิธีทำ ดำเนินการตามขั้นตอนการตรวจสอบมิลเลอร์-ราบิน ได้ดังนี้

ขั้นตอนที่ 1: พิจารณาหาค่า s

เมื่อ $r = 0$, ได้ว่า $\frac{76}{2^0} = 76 \ , \ 2^0 \mid 76$

เมื่อ $r = 1$, ได้ว่า $\frac{76}{2^1} = 38 \ , \ 2^1 \mid 76$

เมื่อ $r = 2$, ได้ว่า $\frac{76}{2^2} = 19 \ , \ 2^2 \mid 76$

เมื่อ $r = 3$, ได้ว่า $\frac{76}{2^3} = 9.5 \ , \ 2^3 \nmid 76$

ดังนั้นได้ว่า $s = 2$

ขั้นตอนที่ 2: คำนวณหาค่า d ดังนี้

$$\text{จาก } d = \frac{(n-1)}{2^s}$$
$$= \frac{76}{2^2}$$
$$= 19$$

ขั้นตอนที่ 3: เลือก a เพื่อใช้ตรวจสอบ n โดยใช้สมการที่ (5.1) และ (5.2)

รอบที่ 1: เลือก $a = 3$ เนื่องจาก gcd(3, 77) = 1

ตรวจสอบสมการที่ 5.1:

$3^{19} \equiv 59 \bmod 77$ ซึ่งไม่เป็นจริง

ตรวจสอบสมการที่ 5.2:

เมื่อ $r = 0$, $3^{2^0 \times 19} = 3^{19} \equiv 59 \bmod 77$ ซึ่งไม่เป็นจริง

เมื่อ $r = 1$, $3^{2^1 \times 19} = 3^{38} \equiv 16 \bmod 77$ ซึ่งไม่เป็นจริง

เนื่องจากการเลือก $a = 3$ และส่งผลให้ไม่พบคำตอบสำหรับทั้งสองสมการสามารถสรุปได้ว่า 77 เป็นจำนวนประกอบ

ตัวอย่างที่ 5.7 จงแสดงว่า $n = 13$ เป็นจำนวนเฉพาะ

วิธีทำ ดำเนินการตามขั้นตอนการตรวจสอบมิลเลอร์-ราบิน ได้ดังนี้

ขั้นตอนที่ 1: พิจารณาหาค่า s

เมื่อ $r = 0$, ได้ว่า $\frac{12}{2^0} = 12, 2^0 \mid 12$

เมื่อ $r = 1$, ได้ว่า $\frac{12}{2^1} = 6, 2^1 \mid 12$

เมื่อ $r = 2$, ได้ว่า $\frac{12}{2^2} = 3, 2^2 \mid 12$

เมื่อ $r = 3$, ได้ว่า $\frac{12}{2^3} = 1.5, 2^3 \nmid 12$

ดังนั้นได้ว่า $s = 2$

ขั้นตอนที่ 2: คำนวนหาค่า d ดังนี้

$$\text{จาก } d = \frac{(n-1)}{2^s}$$
$$= \frac{12}{2^2}$$
$$= 3$$

ขั้นตอนที่ 3: เลือก a เพื่อใช้ตรวจสอบ n โดยใช้สมการที่ (5.1) และ (5.2)

รอบที่ 1: เลือก $a = 3$ เนื่องจาก gcd(3, 13) = 1

ตรวจสอบสมการที่ 5.1:

$3^3 \equiv 1 \bmod 13$ **เป็นจริง**

ดังนั้นจาก $a = 3$ ซึ่งทำให้สมการเป็นจริง ส่งผลให้มีโอกาสที่ n จะเป็นตัวประกอบด้วยความน่าจะเป็น 0.25

รอบที่ 2: เลือก $a = 4$ เนื่องจาก gcd(4, 13) = 1

ตรวจสอบสมการที่ 5.1:

$4^3 \equiv 12 \bmod 13$ ซึ่งไม่เป็นจริง

ตรวจสอบสมการที่ 5.2:

เมื่อ $r = 0$, $4^{2^0 \times 3} = 4^3 \equiv 12 \equiv -1 \bmod 13$ **เป็นจริง**

ดังนั้น a = 3 และ 4 ซึ่งทำให้สมการเป็นจริง ส่งผลให้มีโอกาสที่ n จะเป็นตัวประกอบด้วยความน่าจะเป็น $0.25^2 = 0.0625$

รอบที่ 3: เลือก a = 5 เนื่องจาก gcd(5, 13) = 1

ตรวจสอบสมการที่ 5.1:

$5^3 \equiv 8 \bmod 13$ ซึ่งไม่เป็นจริง

ตรวจสอบสมการที่ 5.2:

เมื่อ r = 0, $5^{2^0 \times 3} = 5^3 \equiv 8 \bmod 13$ ซึ่งไม่เป็นจริง

เมื่อ r = 1, $5^{2^1 \times 3} = 5^6 \equiv 12 \equiv -1 \bmod 13$ **เป็นจริง**

ดังนั้น a = 3, 4 และ 5 ซึ่งทำให้สมการเป็นจริง ส่งผลให้มีโอกาสที่ n จะเป็นตัวประกอบด้วยความน่าจะเป็น $0.25^3 = 0.015625$

รอบที่ 4: เลือก a = 6 เนื่องจาก gcd(6, 13) = 1

ตรวจสอบสมการที่ 5.1:

$6^3 \equiv 8 \bmod 13$ ซึ่งไม่เป็นจริง

ตรวจสอบสมการที่ 5.2:

เมื่อ r = 0, $6^{2^0 \times 3} = 6^3 \equiv 8 \bmod 13$ ซึ่งไม่เป็นจริง

เมื่อ r = 1, $6^{2^1 \times 3} = 6^6 \equiv 12 \equiv -1 \bmod 13$ **เป็นจริง**

ดังนั้น a = 3, 4, 5 และ 6 ซึ่งทำให้สมการเป็นจริง ส่งผลให้มีโอกาสที่ n จะเป็นตัวประกอบด้วยความน่าจะเป็น $0.25^4 = 0.00390625$ หรือกล่าวอีกนัยหนึ่งคือ n = 13 มีความน่าจะเป็นที่จะเป็นจำนวนเฉพาะสูงถึง 0.99609375 หรือ กล่าวได้ว่า 13 เป็นจำนวนเฉพาะ

3. การคำนวณสมการการยกกำลังมอดูลาร์ (Modular Exponentiation Equation Computing)

ขั้นตอนวิธีที่อยู่ในกลุ่มวิทยาการรหัสลับแบบอสมมาตรบางประเภทจำเป็นต้องมีการคำนวณสมการที่มีเลขยกกำลัง อย่างไรก็ตามหากเลขยกกำลังมีขนาดใหญ่การคำนวณโดยตรงจะสูญเสียเวลามหาศาล ในหัวข้อที่ 2 ได้กล่าวถึงการนำทฤษฎีบทเล็กของแฟร์มาต์มาใช้สำหรับแก้ปัญหาดังกล่าวแล้ว แต่ข้อเสียของวิธีดังกล่าวคือค่ามอดูลัสต้องเป็นจำนวนเฉพาะเท่านั้น ดังนั้นในหัวข้อนี้เสนอวิธีที่

ใช้สำหรับคำนวณสมการเลขยกกำลังมอดูลาร์โดยค่ามอดูลัสไม่จำเป็นต้องเป็นจำนวนเฉพาะซึ่งจะช่วยลดเวลาการคำนวณลงได้เป็นอย่างมาก

3.1 การประยุกต์การคูณมอดูลาร์สำหรับแก้ปัญหาการยกกำลังมอดูลาร์

การนำการคูณมอดูลาร์ (Modular Multiplication) มาประยุกต์ใช้สำหรับแก้ปัญหาการคำนวณเลขยกกำลังมอดูลาร์เพื่อลดขนาดของตัวเลขโดยใช้เพียงการคูณแทนการยกกำลัง ดังนี้

สมมติต้องการคำนวณ $m^i \bmod n$

เนื่องจาก

$$m^i = \underbrace{m \times m \times m \times m \times \cdots \times m}_{i \text{ ตัว}}$$

ดังนั้นสามารถนำหาผลลัพธ์โดยใช้เพียงเทคนิคการคูณได้ ดังนี้

รอบที่ 1: คำนวณ $m \bmod n = x_1$

รอบที่ 2: คำนวณ $m^2 \bmod n = x_1^2 \bmod n = x_2$

รอบที่ 3: คำนวณ $m^3 \bmod n = m^2 m \bmod n = x_2 x_1 \bmod n = x_3$

รอบที่ 4: คำนวณ $m^4 \bmod n = m^3 m \bmod n = x_3 x_1 \bmod n = x_4$

รอบที่ 5: คำนวณ $m^5 \bmod n = m^4 m \bmod n = x_4 x_1 \bmod n = x_5$

ดังนั้น

รอบที่ i: คำนวณ $m^i \bmod n = m^{i-1} m \bmod n = x_{i-1} x_1 \bmod n = x_i$

หลังจากเสร็จสิ้นรอบที่ i ได้คำตอบของ $m^i \bmod n$ คือ x_i ซึ่งสังเกตเห็นได้ว่าไม่จำเป็นต้องคำนวณเลขยกกำลังเพียงแต่คำนวณผลคูณเลขคณิตมอดูลาร์ระหว่างจำนวนเต็มสองค่าจำนวน i รอบ (หรือ $i - 1$ รอบหากไม่พิจารณารอบแรก) เมื่อ i คือค่าของเลขยกกำลัง

ตัวอย่างที่ 5.8 จงคำนวณผลลัพธ์ของ $4^{73} \bmod 100$ โดยใช้การคูณมอดูลาร์

วิธีทำ จากตัวอย่าง $m = 4$ และ $n = 100$

ครั้งที่ 1: คำนวณ $m \bmod n$, $4 \bmod 100 = 4$

ครั้งที่ 2: คำนวณ $m^2 \bmod n$, $4^2 \bmod 100 = 16$

ครั้งที่ 3: คำนวณ $m^3 \bmod n = m^2 m \bmod n$, $16 \times 4 \bmod 100 = 64$

ครั้งที่ 4: คำนวณ $m^4 \bmod n = m^3 m \bmod n$, $64 \times 4 \bmod 100 = 56$

ครั้งที่ 5: คำนวณ $m^5 \bmod n = m^4 m \bmod n$, 56 x 4 mod 100 = 24

.
.
.

ครั้งที่ 73: คำนวณ $m^{73} \bmod n = m^{72}\, m \bmod n$, 16 x 4 mod 100 = 64

จากตัวอย่างที่ 5.8 นี้สังเกตได้ว่าการคำนวณผลลัพธ์จาก 4^{73} mod 100 สามารถดำเนินการได้โดยใช้การคูณมอดูลาร์จำนวนทั้งสิ้น 72 ครั้ง

ดังนั้นการใช้การคูณมอดูลาร์สำหรับคำนวณหาผลลัพธ์ของเลขยกกำลังมอดูลาร์จำเป็นต้องมีการคำนวณการคูณมอดูลาร์เป็นจำนวน $i - 1$ ครั้ง

3.2 เลขยกกำลังแบบเร็ว (Fast Exponentiation)

เลขยกกำลังแบบเร็ว [5] ใช้หลักการแปลงเลขยกกำลังจากเลขฐานสิบเป็นเลขฐานสอง และนำค่าประจำหลักของเลขฐานสองเฉพาะที่มีค่าเป็น 1 มาใช้สำหรับการคำนวณ

กำหนดให้ b คือเลขยกกำลังที่เป็นเลขฐานสิบ สามารถแปลงเป็นเลขฐานสองได้ ดังนี้

$$b = b_i 2^i + b_{i-1} 2^{i-1} + b_{i-2} 2^{i-2} + \cdots + b_0 2^0 \tag{5.3}$$

เมื่อ $b_i, b_{i-1}, b_{i-2}, \cdots, b_0$ คือ ค่าเลขฐานสองในตำแหน่งต่างๆ ซึ่งมีค่าเป็น 0 หรือ 1

จากสมการ (5.3) สามารถเขียนในรูปแบบสมการผลรวมได้ดังนี้

$$b = \sum_{j=0}^{i} b_j 2^j \tag{5.4}$$

โดยหลักการคำนวณคือจะแยกส่วนของ b ออกเป็นสมการเลขฐานสองดังสมการที่ (5.3) หรือ (5.4) และดำเนินการคำนวณสมการเลขยกกำลังโดยใช้สมการย่อยทั้งหมดที่ถูกแยกส่วนออกมาเป็นเลขยกกำลัง ขั้นตอนสุดท้ายคือนำผลลัพธ์ทั้งหมดกลับมาคูณกัน อย่างไรก็ตามสามารถตัดสมการย่อยบางตัวออกได้ในกรณีที่ b_j มีค่าเป็น 0

ตัวอย่างที่ 5.9 จงคำนวณผลลัพธ์ของ 4^{73} mod 100 โดยใช้เลขยกกำลังแบบเร็ว

วิธีทำ ดำเนินการเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 73 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$$73 = 1x2^6 + 0x2^5 + 0x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0$$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$$73 = 2^6 + 2^3 + 2^0$$

หรือ $$73 = 2^6 + 2^3 + 1$$

จากขั้นตอนที่ 2 ได้ว่า

$$4^{73} = 4^{2^6+2^3+2^1}$$

$$= 4^{2^6} \times 4^{2^3} \times 4^1$$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $4^2 \bmod 100$, $4^2 \bmod 100 = 16$

ครั้งที่ 2: คำนวณ $4^{2^2} \bmod 100$, เนื่องจาก $4^2 \bmod 100 = 16$ ดังนั้น $4^{2^2} \equiv 16^2 \bmod 100$

ดังนั้น $16^2 \bmod 100 = 56$

ครั้งที่ 3: คำนวณ $4^{2^3} \bmod 100$, เนื่องจาก $4^{2^3} \equiv 4^{2^2} \times 4^{2^2} \bmod 100 = 56^2 \bmod 100$

ดังนั้น $56^2 \bmod 100 =$ **36** → **ค่าที่ต้องใช้**

ครั้งที่ 4: คำนวณ $4^{2^4} \bmod 100$, เนื่องจาก $4^{2^4} \equiv 4^{2^3} \times 4^{2^3} \bmod 100 = 36^2 \bmod 100$

ดังนั้น $36^2 \bmod 100 = 96$

ครั้งที่ 5: คำนวณ $4^{2^5} \bmod 100$, เนื่องจาก $4^{2^5} \equiv 4^{2^4} \times 4^{2^4} \bmod 100 = 96^2 \bmod 100$

ดังนั้น $96^2 \bmod 100 = 16$

ครั้งที่ 6: คำนวณ $4^{2^6} \bmod 100$, เนื่องจาก $4^{2^6} \equiv 4^{2^5} \times 4^{2^5} \bmod 100 = 16^2 \bmod 100$

ดังนั้น $16^2 \bmod 100 =$ **56** → **ค่าที่ต้องใช้**

ดังนั้น $4^{73} \bmod 100 = 4^{2^6} \times 4^{2^3} \times 4^1 \bmod 100$

$$= 56 \times 36 \times 4 \bmod 100$$

$$= 64$$

จากตัวอย่างที่ 5.9 พบว่าหากนำเลขยกกำลังแบบเร็วมาใช้สำหรับแก้ปัญหาจะมีการใช้การคูณมอดูลาร์เพียง 3 ครั้ง และการใช้การยกกำลังสองมอดูลาร์ 6 ครั้ง ซึ่งลดการคำนวณลงเป็นอย่างมากเมื่อเปรียบเทียบกับการแก้ปัญหาโดยวิธีการคูณมอดูลาร์ซึ่งใช้การคูณมอดูลาร์สูงถึง 72 ครั้ง

3.3 ขั้นตอนวิธียกกำลังสองและการคูณ (Square-and-Multiply Algorithm)

ขั้นตอนวิธียกกำลังสองและการคูณ [48] เป็นขั้นตอนวิธีที่ช่วยลดจำนวนครั้งสำหรับการคำนวณการคูณมอดูลาร์ หลักการคือแปลงเลขฐานของเลขยกกำลังจากเลขฐานสิบเป็นเลขฐานสอง กำหนดให้หลังจากเสร็จสิ้นการแปลงเลขฐานแล้วเลขยกกำลังซึ่งเป็นเลขฐานสองมีจำนวนทั้งสิ้น m

บิต โดยที่ตำแหน่งบิตที่มีนัยสำคัญสูงที่สุดคือ m-1 และเรียงลงมาจนถึงตำแหน่งบิตที่มีนัยสำคัญต่ำที่สุดคือ 0 กำหนดให้ค่าเริ่มต้นของคำตอบคือ z มีค่าเท่ากับเลขฐานและอยู่ตำแหน่งที่ m-1 (ตำแหน่งนี้มีค่าเป็น 1 เสมอ) ดังนั้นการแก้ปัญหาเลขยกกำลังมอดูลาร์ด้วยขั้นตอนวิธียกกำลังสองและการคูณจะเริ่มจาก z และย้อนกลับมาพิจารณาบิตของเลขยกกำลังในตำแหน่งที่อยู่ติดกันครั้งละ 1 บิตโดยหากบิตดังกล่าวมีค่าเป็น “0” จะคำนวณเพียงกำลังสองของ z ปัจจุบัน แต่หากบิตในตำแหน่งที่กำลังพิจารณาอยู่มีค่าเป็น “1” จำเป็นต้องคำนวณการคูณมอดูลาร์อีกครั้งด้วย หลังจากได้ผลลัพธ์ของการคำนวณกำลังสองมอดูลาร์ โดยดำเนินการซ้ำเดิมจนกระทั่งการพิจารณามาถึงตำแหน่งที่ 0 ซึ่งเป็นตำแหน่งสุดท้ายจึงเสร็จสิ้นกระบวนการ และผลลัพธ์สุดท้ายคือคำตอบ

ตัวอย่างที่ 5.10 จงคำนวณผลลัพธ์ของ $4^{73} \bmod 100$ โดยใช้ขั้นตอนวิธียกกำลังสองและการคูณ

วิธีทำ ดำเนินการเป็นขั้นตอนได้ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง (73) ให้อยู่ในรูปแบบเลขฐานสอง ได้ดังนี้

ตำแหน่งบิต	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ตัวเลขประจำตำแหน่ง	1	0	0	1	0	0	1

จากขั้นตอนที่ 1 เมื่อแปลงเลขยกกำลังเป็นเลขฐานสองแล้วพบว่าผลลัพธ์มีขนาด 7 บิต

ขั้นตอนที่ 2: เริ่มแก้ปัญหาด้วยขั้นตอนวิธียกกำลังสองและการคูณ

ครั้งที่ 1: ตำแหน่งที่ 6, กำหนดให้ $z = 4$

ครั้งที่ 2: ตำแหน่งที่ 5, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 0 ดังนั้น

$z = 4 \times 4 \bmod 100 = 4^2 \bmod 100 = 4^{10_2} \bmod 100 = 16$

ครั้งที่ 3: ตำแหน่งที่ 4, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 0 ดังนั้น

$z = 4^2 \times 4^2 \bmod 100 = 16 \times 16 \bmod 100 = 4^4 \bmod 100 = 4^{100_2} \bmod 100 = 56$

ครั้งที่ 4: ตำแหน่งที่ 3, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 1 ดังนั้น

$z = 4^4 \times 4^4 \bmod 100 = 56 \times 56 \bmod 100 = 4^8 \bmod 100 = 4^{1000_2} \bmod 100 = 36$

$z = 4^8 \times 4 \bmod 100 = 36 \times 4 \bmod 100 = 4^9 \bmod 100 = 4^{1001_2} \bmod 100 = 44$

ครั้งที่ 5: ตำแหน่งที่ 2, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 0 ดังนั้น

$z = 4^9 \times 4^9 \bmod 100 = 44 \times 44 \bmod 100 = 4^{18} \bmod 100 = 4^{10010_2} \bmod 100 = 36$

ครั้งที่ 6: ตำแหน่งที่ 1, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 0 ดังนั้น

$z = 4^{18} \times 4^{18} \bmod 100 = 36 \times 36 \bmod 100 = 4^{36} \bmod 100 = 4^{100100_2} \bmod 100 = 96$

ครั้งที่ 7: ตำแหน่งที่ 0, เนื่องจากตัวเลขประจำตำแหน่งมีค่าเป็น 1 ดังนั้น

$z = 4^{36} \times 4^{36} \bmod 100 = 96 \times 96 \bmod 100 = 4^{72} \bmod 100 = 4^{100100_2} \bmod 100 = 16$

$z = 4^{72} \times 4 \bmod 100 = 16 \times 4 \bmod 100 = 4^{73} \bmod 100 = 4^{1001001_2} \bmod 100 = \mathbf{64}$

ดังนั้น การคำนวณหาผลลัพธ์ของ $4^{73} \bmod 100$ โดยใช้ขั้นตอนวิธียกกำลังสองและการคูณได้คำตอบคือ 64

อย่างไรก็ตามจากตัวอย่างที่ 5.10 มีการคำนวณการคูณมอดูลาร์เพียง 2 ครั้งและการยกกำลังสองมอดูลาร์อีก 6 ครั้ง โดยการคูณเกิดขึ้นจากกรณีที่ตำแหน่งของบิตที่ถูกพิจารณามีค่าเป็น 1 ดังนั้นหากเลขยกกำลังมีบิตที่มีค่า 1 เป็นจำนวนมากจะส่งผลให้มีกระบวนการคูณมอดูลาร์เกิดมากขึ้นตามไปด้วย โดยเรียกจำนวนของบิตที่มีค่าเป็น 1 ทั้งหมดว่าค่าน้ำหนักแฮมมิ่ง (Hamming Weight) ยกตัวอย่างเช่น $23 = 10111_2$ มีค่าน้ำหนักแฮมมิ่งคือ 4 หรือ $37 = 100101_2$ มีค่าน้ำหนักแฮมมิ่งเป็น 3 เป็นต้น

4. ทฤษฎีเศษเหลือจีน (Chinese Remainder Theorem)

ทฤษฎีเศษเหลือจีน [30] คือระบบสมภาคเชิงเส้นแบบหลายชั้นที่มีแนวคิดเริ่มต้นมาจากนักคณิตศาสตร์ชาวจีนชื่อ ซัน ซู (Sun Tzu) ที่ว่ามีจำนวนเต็มค่าหนึ่งซึ่งหากนำมาหารด้วย 3 จะได้เศษคือ 2 หากนำมาหารด้วย 5 จะได้เศษคือ 3 และหากนำมาหารด้วย 7 เศษที่ได้คือ 2 คำถามคือจำนวนเต็มดังกล่าวมีค่าเท่ากับเท่าไร สมมติว่า x คือจำนวนเต็มปริศนาของ ซัน ซู สามารถเขียนเป็นสมการได้ ดังนี้

$$x \equiv 2 \bmod 3$$
$$x \equiv 3 \bmod 5$$
$$x \equiv 2 \bmod 7$$

ซึ่งคำตอบที่ได้คือ $x = 23$

หลังจากนั้นจึงเกิดทฤษฎีเศษเหลือจีนขึ้น ดังนี้

ทฤษฎีบทที่ 5.4 กำหนดให้ $m_1, m_2, m_3, \cdots, m_n$ เป็นจำนวนเฉพาะสัมพัทธ์ตรงกัน และ $a_1, a_2, a_3 \cdots, a_n$ เป็นจำนวนเต็มใดๆ และมีจำนวนเต็ม x ที่สมมูลกับสมการทั้งหมดต่อไปนี้

$$x \equiv a_1 \bmod m_1$$
$$x \equiv a_2 \bmod m_2$$
$$x \equiv a_3 \bmod m_3$$
$$\vdots$$
$$x \equiv a_n \bmod m_n$$

สามารถคำนวณหา x ได้จาก

$$x = (\sum_{i=1}^{n} a_i y_i M_i) \bmod m$$

เมื่อ $$m = \prod_{i=1}^{n} m_i$$

และ $$M_i = \frac{m}{m_i}$$

โดยที่สัญลักษณ์ “ $\prod_{i=a}^{b}$ ” แทนผลคูณของแต่ละพจน์ระหว่างพจน์ที่ a ถึง b

พิสูจน์

เนื่องจาก m_i และ M_i เป็นจำนวนเฉพาะสัมพัทธ์ตรงกัน จึงกล่าวได้ว่ามีค่า y_i ซึ่งเป็นค่าผกผันของ M_i ภายใต้การโมดูโล m_i

$$y_i M_i \equiv 1 \bmod m_i$$

ดังนั้น $$a_i y_i M_i \equiv a_i \bmod m_i$$

เนื่องจาก m_i เป็นตัวประกอบของ M_j เมื่อ $i \neq j$ กล่าวได้ว่า

$$a_j y_j M_j \equiv 0 \bmod m_i$$

จากเหตุผลข้างต้น สามารถเขียน x ให้อยู่ในรูปแบบของสมการต่อไปนี้

$$x \equiv a_i y_i M_i + \sum_{j=1, j\neq i}^{n} a_j y_j M_j \equiv a_i \bmod m_i$$

หรือสามารถคำนวณหา x ได้จากสมการต่อไปนี้

$$x = (\sum_{i=1}^{n} a_i y_i M_i) \bmod m$$ □

โดยทฤษฎีเศษเหลือจีนจะถูกนำไปประยุกต์ใช้สำหรับการเร่งความเร็วกระบวนการถอดรหัสสำหรับวิทยาการรหัสลับแบบอสมมาตร เช่น วิทยาการรหัสลับอาร์เอสเอ

ตัวอย่างที่ 5.11 จงใช้ทฤษฎีเศษเหลือจีนเพื่อหาค่า x จากระบบสมการต่อไปนี้

$$x \equiv 2 \bmod 3$$
$$x \equiv 3 \bmod 5$$
$$x \equiv 2 \bmod 7$$

วิธีทำ

จากระบบสมการกำหนดให้ $a_1 = 2$, $m_1 = 3$, $a_2 = 3$, $m_2 = 5$, $a_3 = 2$ และ $m_3 = 7$

เริ่มจากคำนวณหาค่า $m = m_1m_2m_3 = 3 \times 5 \times 7 = 105$

คำนวณหา M_1, M_2 และ M_3 ดังนี้

$$M_1 = \frac{m}{m_1} = \frac{105}{3} = 35$$
$$M_2 = \frac{m}{m_2} = \frac{105}{5} = 21$$
$$M_3 = \frac{m}{m_3} = \frac{105}{7} = 15$$

และใช้ขั้นตอนวิธียุคลิดภาคขยายเพื่อหาค่า y_1, y_2 และ y_3 ได้ว่า

$y_1 = 2$ เนื่องจาก $2 \times 35 = 70 = 1 \bmod 3$

$y_2 = 1$ เนื่องจาก $1 \times 21 = 21 = 1 \bmod 5$

$y_3 = 1$ เนื่องจาก $1 \times 15 = 15 = 1 \bmod 7$

ดังนั้น

$$x = (a_1y_1M_1 + a_2y_2M_2 + a_3y_3M_3) \bmod m$$
$$= (2 \times 2 \times 35 + 3 \times 1 \times 21 + 2 \times 1 \times 15) \bmod 105$$
$$= (140 + 63 + 30) \bmod 105$$
$$= 233 \bmod 105$$
$$= 23$$

5. ฟังก์ชันออยเลอร์ (Euler's Function)

ฟังก์ชันออยเลอร์ [54] มีชื่อเต็มคือฟังก์ชันทอเทียนต์ออยเลอร์คือฟังก์ชันสำหรับนำมาประยุกต์ใช้กับวิทยาการรหัสลับแบบอสมมาตรบางกลุ่ม เช่นวิทยาการรหัสลับอาร์เอสเอ

กำหนดให้ $n \in \mathbb{N}$ นิยามฟังก์ชันออยเลอร์ เป็นดังนี้

$\Phi(n)$ คือ จำนวนสมาชิกจำนวนเต็มบวกที่มีเงื่อนไขเป็นดังนี้ $0 < a < n$ และ $\gcd(a, n) = 1$

ตัวอย่างที่ 5.12 จงหา $\Phi(4)$

วิธีทำ เนื่องจากตัวเลขที่ต้องใช้สำหรับการตรวจสอบมีทั้งหมด 4 ตัวดังนี้ 1, 2, 3 และ 4 ผลการหาค่าหารร่วมมากระหว่างตัวเลขทั้ง 4 ตัวและ 4 เป็นดังตารางที่ 5.1

ตารางที่ 5.1 การหาค่า $\Phi(4)$

a	1	2	3	4
gcd(a, 4)	1	2	1	4

จากตารางที่ 5.1 สังเกตได้ว่ามีสมาชิกจำนวนสองค่าที่ค่าหารร่วมมากระหว่างค่าดังกล่าวและ 4 มีค่าเป็น 1 จึงสรุปได้ว่า $\Phi(4) = 2$

ตัวอย่างที่ 5.13 จงหา $\Phi(7)$

วิธีทำ เนื่องจากตัวเลขที่ต้องใช้สำหรับการตรวจสอบมีทั้งหมด 7 ตัวดังนี้ 1, 2, 3, 4, 5, 6 และ 7 ผลการหาค่าหารร่วมมากระหว่างตัวเลขทั้ง 7 ตัวและ 7 เป็นดังตารางที่ 5.2

ตารางที่ 5.2 การหาค่า $\Phi(7)$

a	1	2	3	4	5	6	7
gcd(a, 7)	1	1	1	1	1	1	7

จากตารางที่ 5.2 สังเกตได้ว่ามีสมาชิกจำนวนหกค่าที่ค่าหารร่วมมากระหว่างค่าดังกล่าวและ 7 มีค่าเป็น 1 จึงสรุปได้ว่า $\Phi(7) = 6$

หมายเหตุ: $\Phi(p) = p - 1$ เมื่อ p เป็นจำนวนเฉพาะเนื่องจากมีเพียงแค่ 1 และ p เท่านั้นที่หาร p ได้ลงตัว

ทฤษฎีบทที่ 5.5 กำหนดให้ p เป็นจำนวนเฉพาะ และ $n \in \mathbb{N}$ ได้ว่า

$$\Phi(p^n) = p^n - p^{n-1}$$

พิสูจน์ จำนวนเต็มบวกทั้งหมดที่ไม่เป็นจำนวนเฉพาะสัมพัทธ์ตรงกันกับ p^n (ค่าหารร่วมมากระหว่างจำนวนเต็มดังกล่าวและ p มีค่าไม่เท่ากับ 1) มีดังต่อไปนี้

$$\underbrace{p, 2p, 3p, \cdots, (p-1)p, \cdots, p^{n-2}p, p^{n-1}p}_{\textbf{p}^{n-1}\textbf{ ตัว}}$$

และเนื่องจากจำนวนเต็มบวกตั้งแต่ 1 ถึง p^{n-1} มีจำนวนทั้งหมด p^{n-1} ตัวดังนั้น

$$\Phi(p^n) = p^n - p^{n-1}$$ □

ตัวอย่างที่ 5.14 จงหา $\Phi(7^3)$

วิธีทำ จากทฤษฎีบทที่ 5.5 ,

$$\begin{aligned}\Phi(7^3) &= 7^3 - 7^{3-1} \\ &= 7^3 - 7^2 \\ &= 7^2 \times (7-1) \\ &= 49 \times 6 \\ &= 294\end{aligned}$$

บทตั้งที่ 5.6 กำหนดให้ $a, b, c, d \in \mathbb{N}$ โดยที่ $\gcd(a, b) = \gcd(b, d) = \gcd(a, c) = 1$ จะได้ว่า $\gcd(ad + bc, ab) = 1$

พิสูจน์ สมมติให้ p เป็นจำนวนเฉพาะ โดยที่ $p \mid ab$ และ $p \mid (ad + bc)$ ดังนั้นจากเงื่อนไขที่ 1 ผลลัพธ์ที่ได้จะถูกแบ่งออกเป็นสองกรณี คือ $p \mid a$ หรือ $p \mid b$

กรณีที่ 1 ($p \mid a$), เนื่องจาก $p \mid (ad + bc)$ และ $p \nmid b$ ดังนั้นสรุปได้ว่า $p \mid c$ แต่อย่างไรก็ตามพบว่ามีข้อขัดแย้งเกิดขึ้น เนื่องจาก $\gcd(a, c) = 1$

กรณีที่ 2 ($p \mid b$), เนื่องจาก $p \mid (ad + bc)$ และ $p \nmid a$ ดังนั้นสรุปได้ว่า $p \mid d$ แต่อย่างไรก็ตามพบว่ามีข้อขัดแย้งเกิดขึ้น เนื่องจาก $\gcd(b, d) = 1$

จากทั้งสองกรณีกล่าวได้ว่าไม่มีจำนวนเฉพาะใดๆ ที่หาร $ad + bc$ และ ab ลงตัว จึงสรุปได้ว่า $\gcd(ad + bc, ab) = 1$ □

ทฤษฎีบทที่ 5.7 กำหนดให้ $a, b \in \mathbb{N}$ ได้ว่า $\Phi(ab) = \Phi(a)\Phi(b)$

พิสูจน์ กำหนดให้

$$X = \{x \mid 1 \leq x < a \text{ และ } \gcd(x, a) = 1\}$$

$$Y = \{y \mid 1 \leq y < b \text{ และ } \gcd(y, b) = 1\}$$

$$Z = \{z \mid 1 \leq z < ab \text{ และ } \gcd(z, ab) = 1\}$$

การพิสูจน์แบ่งเป็นสองกรณี ดังนี้

กรณีที่ 1: เนื่องจาก $\gcd(z, ab) = 1$ กล่าวได้ว่า $\gcd(z, a) = 1$ และ $\gcd(z, b) = 1$ ดังนั้นจึงมี $q_1, r_1, q_2, r_2 \in \mathbb{Z}$ ที่ทำให้สมการต่อไปนี้เป็นจริงเสมอ

$$z = q_1a + r_1, \qquad 1 \leq r_1 < a$$

และ $$z = q_2b + r_2, \qquad 1 \leq r_2 < b$$

และจากทั้งสองสมการพบว่า $\gcd(r_1, a) = 1$ และ $\gcd(r_2, b) = 1$ ซึ่งสังเกตได้ว่าสมาชิกทั้งหมดของ $\Phi(ab)$ ล้วนเป็นสมาชิกของ $\Phi(a)\Phi(b)$ เช่นกัน ดังนั้นจาก $r_1 \in X$ และ $r_2 \in Y$ ได้ว่า

$$\Phi(ab) \leq \Phi(a)\Phi(b)$$

กรณีที่ 2: กำหนดให้ $c \in X$ และ $d \in Y$ ได้ว่า $\gcd(a, c) = 1$ และ $\gcd(b, d) = 1$ ดังนั้นจากบทตั้งที่ 5.6 ได้ว่า $\gcd(ad + bc, ab) = 1$ ซึ่งมี $q, r \in \mathbb{Z}$ ที่ทำให้สมการต่อไปนี้เป็นจริงเสมอ

$$ad + bc = (ab)q + r, \quad 1 \leq r < ab$$

และจากสมการข้างต้นพบว่า $\gcd(r, ab) = 1$ ซึ่งสังเกตได้ว่าสมาชิกทั้งหมดของ $\Phi(a)\Phi(b)$ ล้วนเป็นสมาชิกของ $\Phi(ab)$ เช่นกัน จึงได้ว่า

$$\Phi(ab) \geq \Phi(a)\Phi(b)$$

ดังนั้นจากทั้งสองกรณีสรุปได้ว่า

$$\Phi(ab) = \Phi(a)\Phi(b)$$ □

ดังนั้นจากทฤษฎีบทที่ 5.7 สมมติ a และ b เป็นจำนวนเฉพาะจะได้ว่า

$$\Phi(ab) = (a-1)(b-1)$$

ทฤษฎีบทที่ 5.8 กำหนดให้ $n \in \mathbb{Z}$ โดยที่

$$n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$$

เมื่อ $p_1 < p_2 < p_3 < \cdots < p_k$ เป็นจำนวนเฉพาะ $a_i \in \mathbb{N}$ โดยที่ $i = 1, 2, 3, \cdots, k$ และ $k > 1$ ได้ว่า

$$\Phi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1}) \cdots (p_k^{a_k} - p_k^{a_k-1})$$

สำหรับการพิสูจน์สามารถดำเนินการได้โดยใช้อุปนัยทางคณิตศาสตร์ซึ่งมีขั้นตอนการดำเนินการเป็นดังนี้

ขั้นตอนที่ 1: พิสูจน์ว่าสมการเป็นจริงเมื่อ $k = 1$

ขั้นตอนที่ 2: สมมติให้สมการเป็นจริงเมื่อ $k = i$

ขั้นตอนที่ 3: พิสูจน์ว่าสมการเป็นจริงเมื่อ $k = i + 1$

พิสูจน์

ขั้นตอนที่ 1: สมมติ $k = 1$ ดังนั้น $n = p_1^{a_1}$ และจากทฤษฎีบทที่ 5.5 ได้ว่า

$\Phi(n) = \Phi(p_1^{a_1}) = (p_1^{a_1} - p_1^{a_1-1})$ ซึ่งทำให้ทฤษฎีนี้เป็นจริง

ขั้นตอนที่ 2: สมมติว่าทฤษฎีข้างต้นเป็นจริงเมื่อ $k = i$ ดังนั้นได้ว่า

$$n = p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i}$$

และ $$\Phi(n) = \Phi(p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i}) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1}) \cdots (p_i^{a_i} - p_i^{a_i-1})$$

ขั้นตอนที่ 3: กำหนดให้ $k = i + 1$ ดังนั้น

$$n = p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i} p_{i+1}^{a_{i+1}}$$

จากทฤษฎีบทที่ 5.7 ได้ว่า

$$\Phi(n) = \Phi(p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i} p_{i+1}^{a_{i+1}}) = \Phi(p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i}) \Phi(p_{i+1}^{a_{i+1}})$$

จากขั้นตอนที่ 2 ได้ว่า

$$\Phi(p_1^{a_1} p_2^{a_2} \cdots p_i^{a_i}) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1}) \cdots (p_i^{a_i} - p_i^{a_i-1})$$

ดังนั้น

$$\Phi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1})\cdots(p_i^{a_i} - p_i^{a_i-1})\Phi(p_{i+1}^{a_{i+1}})$$

และจาก $\Phi(p_{i+1}^{a_{i+1}}) = (p_{i+1}^{a_{i+1}} - p_{i+1}^{a_{i+1}-1})$

ดังนั้น

$\Phi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1})\cdots(p_i^{a_i} - p_i^{a_i-1})(p_{i+1}^{a_{i+1}} - p_{i+1}^{a_{i+1}-1})$ ซึ่งทำให้ทฤษฎีนี้เป็นจริง

ดังนั้นจึงสรุปได้ว่า

$$\Phi(n) = (p_1^{a_1} - p_1^{a_1-1})(p_2^{a_2} - p_2^{a_2-1})\cdots(p_k^{a_k} - p_k^{a_k-1}) \qquad \square$$

ตัวอย่างที่ 5.15 จงหาค่า $\Phi(2^3 \times 5^2)$

วิธีทำ จากทฤษฎีบทที่ 5.8 ได้ว่า

$$\begin{aligned}\Phi(2^3 \times 5^2) &= (2^3 - 2^2)(5^2 - 5)\\ &= (8 - 4)(25 - 5)\\ &= 4 \times 20\\ &= 80\end{aligned}$$

6. ทฤษฎีบทของออยเลอร์

ทฤษฎีบทของออยเลอร์คือทฤษฎีที่พัฒนาต่อจากทฤษฎีบทเล็กของแฟร์มาต์ซึ่งเป็นทฤษฎีที่สำคัญที่นำไปประยุกต์ใช้กับวิทยาการรหัสลับแบบอสมมาตรบางกลุ่มเช่น วิทยาการรหัสลับอาร์เอสเอ

บทตั้งที่ 5.9

กำหนดให้ $n \in \mathbb{N}$, $a \in \mathbb{Z}$ โดยที่ $\gcd(a, n) = 1$ และ $r_1, r_2, r_3, \cdots, r_{\Phi(n)} \in \mathbb{Z}^+$ มีค่าน้อยกว่าหรือเท่ากับ n โดยที่ $\gcd(r_i, n) = 1$ เมื่อ $i = 1, 2, 3, \cdots, \Phi(n)$ แล้วได้ว่าส่วนตกค้างที่น้อยที่สุดของ $\{ar_1, ar_2, ar_3, \cdots, ar_{\Phi(n)}\}$ ภายใต้การมอดูโล n คือ $r_1, r_2, r_3, \cdots, r_{\Phi(n)}$

พิสูจน์ การพิสูจน์เริ่มจากกำหนดให้ $\gcd(ar_i, n) > 1$ โดยให้ $\gcd(ar_i, n) = p$ โดยที่ p เป็นจำนวนเฉพาะ ได้ว่า $p \mid n$ และ $p \mid ar_i$ ดังนั้น $p \mid a$ หรือ $p \mid r_i$ สมมติ $p \mid r_i$ ได้ว่า $\gcd(r_i, n) \neq 1$ ซึ่งเกิดข้อ

ขัดแย้งขึ้น ดังนั้นหากเปลี่ยนเป็น $p \mid a$ ได้ว่า gcd(a, n) $\neq$ 1 ซึ่งเกิดข้อขัดแย้งขึ้นเช่นกัน ดังนั้น gcd(ar_i, n) = 1 เสมอ

ต่อมากำหนดให้ $ar_i \equiv ar_j \bmod n$ ได้ว่า $n \mid (ar_i - ar_j)$ หรือ $n \mid a(r_i - r_j)$ เนื่องจาก gcd(a, n) = 1 จึงได้ $n \mid (r_i - r_j)$ หรือ $r_i \equiv r_j \bmod n$ อย่างไรก็ตามเนื่องจาก r_i และ r_j คือส่วนตกค้างน้อยที่สุดภายใต้การมอดูโล n หมายความว่า $r_i = r_j$ ซึ่งหาก $r_i \neq r_j \bmod n$ แล้วจะได้ว่า $ar_i \neq ar_j \bmod n$ ดังนั้นสามารถสรุปได้ว่าส่วนตกค้างน้อยที่สุดของ $\{ar_1, ar_2, ar_3, \cdots, ar_{\Phi(n)}\}$ ภายใต้การมอดูโล n คือ $r_1, r_2, r_3, \cdots, r_{\Phi(n)}$ □

ทฤษฎีบทที่ 5.10 กำหนดให้ $n \in \mathbb{N}$, $a \in \mathbb{Z}$ โดยที่ gcd(a, n) = 1 ได้ว่า $a^{\Phi(n)} \bmod n = 1$

พิสูจน์ กำหนดให้ $r_1, r_2, r_3, \cdots, r_{\Phi(n)}$ คือส่วนตกค้างน้อยที่สุดภายใต้การมอดูโล n ดังนั้นจากบทตั้งที่ 5.9 ได้ว่า

$$ar_1 \times ar_2 \times ar_3 \times \cdots \times ar_{\Phi(n)} \equiv r_1 \times r_2 \times r_3 \times \cdots \times r_{\Phi(n)} \bmod n$$

ดังนั้น $r_1 \times r_2 \times r_3 \times \cdots \times r_{\Phi(n)} \times a^{\Phi(n)} \equiv r_1 \times r_2 \times r_3 \times \cdots \times r_{\Phi(n)} \bmod n$

เนื่องจาก gcd(r_i, n) = 1 และ gcd($r_1, r_2, r_3, \cdots, r_{\Phi(n)}$) = 1

ดังนั้น $a^{\Phi(n)} \equiv 1 \bmod n$ □

ตัวอย่างที่ 5.16 จงหาค่า $3^{60} \bmod 77$

วิธีทำ จากตัวอย่างได้ว่า $a = 3$, $n = 77 = 11 \times 7$

เนื่องจาก gcd(3, 77) = 1 และ $\Phi(n) = 10 \times 6 = 60$

ดังนั้นจากทฤษฎีบทที่ 5.10 ได้ว่า $3^{60} \bmod 77 = 1$

ตรวจสอบโดยใช้วิธีเลขยกกำลังแบบเร็ว

ขั้นตอนที่ 1: แปลงยกกำลัง (60) ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$$60 = 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 0x2^0$$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$$60 = 2^5 + 2^4 + 2^3 + 2^2$$

จากขั้นตอนที่ 2 ได้ว่า

$$3^{60} = 3^{2^5+2^4+2^3+2^2}$$

$$= 3^{2^5} \times 3^{2^4} \times 3^{2^3} \times 3^{2^2}$$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ 3^2, $3^2 = 9$

ครั้งที่ 2: คำนวณ 3^{2^2} , เนื่องจาก $3^2 = 9$ ดังนั้น $3^{2^2} = 9^2 \bmod 77$

ดังนั้น $9^2 \bmod 77 = \mathbf{4}$ → **ค่าที่ต้องใช้**

ครั้งที่ 3: คำนวณ 3^{2^3} , เนื่องจาก $3^{2^3} = 3^{2^2} \times 3^{2^2} = 4^2 \bmod 77$

ดังนั้น $4^2 \bmod 77 = \mathbf{16}$ → **ค่าที่ต้องใช้**

ครั้งที่ 4: คำนวณ 3^{2^4} , เนื่องจาก $3^{2^4} = 3^{2^3} \times 3^{2^3} = 16^2 \bmod 77$

$16^2 \bmod 77 = \mathbf{25}$ → **ค่าที่ต้องใช้**

ครั้งที่ 5: คำนวณ 3^{2^5} , เนื่องจาก $3^{2^5} = 3^{2^4} \times 3^{2^4} = 25^2 \bmod 77$

ดังนั้น $25^2 \bmod 77 = \mathbf{9}$ → **ค่าที่ต้องใช้**

ดังนั้น $3^{60} \bmod 77 = 3^{2^5} \times 3^{2^4} \times 3^{2^3} \times 3^{2^2} \bmod 77$

$= 9 \times 25 \times 16 \times 4 \bmod 100$

$= 1$

7. รากปฐมฐาน (Primitive root)

กำหนดให้ $g \in \mathbb{Z}^+$ และ p เป็นจำนวนเฉพาะแล้วเรียก g ว่ารากปฐมฐานก็ต่อเมื่อผลลัพธ์ของ $g^1, g^2, g^3, \cdots, g^{p-1}$ ภายใต้มอดูโล p เกิดสมาชิกทุกค่าในฟิลด์ GF*(p) หรือ

$$GF^*(p) = \{g^1, g^2, g^3, \cdots, g^{p-1}\}$$

ตัวอย่างที่ 5.17 จงตรวจสอบว่า 3 เป็นรากปฐมฐานใน GF*(7) หรือไม่

วิธีทำ เนื่องจาก GF*(7) = {1, 2, 3, 4, 5, 6}

ดังนั้นหาก 3 เป็นรากปฐมฐานแล้ว 3^1, 3^2, 3^3, 3^4, 3^5 และ 3^6 ภายใต้การมอดูโล 7 จะต้องมีผลลัพธ์ทั้งหมด 7 ค่าประกอบด้วย 1, 2, 3, 4, 5 และ 6 โดยผลลัพธ์ไม่จำเป็นต้องเรียงตามลำดับ

1. $3^1 \bmod 7 = 3$
2. $3^2 \bmod 7 = 2$
3. $3^3 \bmod 7 = 6$
4. $3^4 \bmod 7 = 4$

5. $3^5 \bmod 7 = 5$
6. $3^6 \bmod 7 = 1$

เนื่องจากเกิดผลลัพธ์ครบทั้ง 6 ค่าจึงสรุปได้ว่า 3 เป็นรากปฐมฐาน

8. เศษส่วนต่อเนื่อง (Continued Fraction)

กำหนดให้ $q_0 \in \mathbb{Z}$ และ $q_1, q_2, q_3, \cdots, q_n \in \mathbb{Z}^+$ เศษส่วนต่อเนื่องคือนิพจน์ที่ถูกเขียนอยู่ในรูปต่อไปนี้

$$x = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots + \cfrac{1}{q_n}}}} \tag{5.5}$$

จากสมการ (5.5) สามารถเขียนรูปแบบย่อได้คือ $x = [q_0; q_1, q_2, \cdots, q_n]$

ตัวอย่างที่ 5.18 เศษส่วนต่อเนื่องของ $\frac{33}{7}$ คือ [4; 1, 2, 2] เนื่องจาก

วิธีทำ

$$\begin{aligned}\frac{33}{7} &= 4 + \frac{5}{7} \\ &= 4 + \frac{1}{\frac{7}{5}} = 4 + \frac{1}{1 + \frac{2}{5}} \\ &= 4 + \frac{1}{1 + \frac{1}{\frac{5}{2}}} = 4 + \frac{1}{1 + \frac{1}{2 + \frac{1}{2}}}\end{aligned}$$

ดังนั้น $\frac{33}{7} = 4 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{2}}} = [4; 1, 2, 2]$

โดยสามารถใช้ขั้นตอนวิธียุคลิดสำหรับหาค่า $q_0, q_1, q_2, \cdots, q_n$ ได้ อย่างไรก็ตามเนื่องจาก q_0 อาจเป็นจำนวนเต็มลบซึ่งแตกต่างจากผลหารทุกค่าจากขั้นตอนวิธียุคลิดที่กล่าวไว้ในบทที่ 2 ที่เป็นจำนวนเต็มบวกเสมอ จึงต้องปรับแก้ขั้นตอนวิธียุคลิดเล็กน้อยเพื่อแก้ปัญหาเศษส่วนต่อเนื่อง กำหนดให้ $\frac{r_0}{r_1}$ คือผลลัพธ์ของเศษส่วนต่อเนื่อง การใช้ขั้นตอนวิธียุคลิดสำหรับหาผลหาร $q_0, q_1, q_2, \cdots, q_n$ ทั้งหมดเป็นดังนี้

$$r_0 = r_1q_0 + r_2$$
$$r_1 = r_2q_1 + r_3$$
$$r_2 = r_3q_2 + r_4$$
$$\vdots$$
$$r_{n-1} = r_nq_{n-1} + r_{n+1}$$
$$r_n = r_{n+1}q_n$$

จากขั้นตอนวิธียุคลิดข้างต้น $r_{n+1} = 1$ เสมอ เมื่อ $\gcd(r_0, r_1) = 1$

ตัวอย่างที่ 5.19 จงใช้ขั้นตอนวิธียุคลิดเพื่อหาเศษส่วนต่อเนื่องของ $\frac{33}{7}$

วิธีทำ จากโจทย์ได้ $r_0 = 33$ และ $r_1 = 7$ ดังนั้นใช้ขั้นตอนวิธียุคลิดได้ว่า

$33 = 7\text{x}4 + 5$ -> $q_0 = 4$

$7 = 5\text{x}1 + 2$ -> $q_1 = 1$

$5 = 2\text{x}2 + 1$ -> $q_2 = 2$

$2 = 1\text{x}2$ -> $q_3 = 2$

ดังนั้น $\frac{33}{7} = [4; 1, 2, 2]$

ตัวอย่างที่ 5.20 จงใช้ขั้นตอนวิธียุคลิดเพื่อหาเศษส่วนต่อเนื่องของ $\frac{-149}{31}$

วิธีทำ จากโจทย์ได้ $r_0 = -149$ และ $r_1 = 31$ ดังนั้นใช้ขั้นตอนวิธียุคลิดได้ว่า

$-149 = 31 \times (-5) + 6$ -> $q_0 = -5$

$31 = 6 \times 5 + 1$ -> $q_1 = 5$

$6 = 1 \times 6$ -> $q_2 = 6$

ดังนั้น $\frac{-149}{31}$ = [-5; 5, 6]

9. ตัวเบนเข้าของเศษส่วนต่อเนื่อง (Convergent of Continued Fraction)

กำหนดให้เศษส่วนต่อเนื่องคือ $x = [q_0; q_1, q_2, \cdots, q_n]$ แล้วตัวเบนเข้าของเศษส่วนต่อเนื่องคือลำดับย่อยของ x เช่น $[q_0]$, $[q_0; q_1]$, $[q_0; q_1, q_2], \cdots, [q_0; q_1, q_2, \cdots, q_{n-2}, q_{n-1}]$ โดยที่ตัวเบนเข้าจะมีค่าเข้าใกล้ x สูงมากในกรณีที่สมาชิกในลำดับย่อยของ x ถูกพิจารณาจำนวนมากขึ้น

ตัวอย่างที่ 5.21 จงพิจารณาหาตัวเบนเข้าของ $\frac{33}{7}$ เมื่อพิจารณาลำดับย่อย 3 ตำแหน่ง

วิธีทำ จากตัวอย่างที่ 5.19 ทราบว่าเศษส่วนต่อเนื่องของ $\frac{33}{7}$ คือ [4; 1, 2, 2]

ดังนั้นตัวเบนเข้าของ $\frac{33}{7}$ เมื่อพิจารณาลำดับย่อย 3 ตำแหน่งจะมีค่าเป็น [4; 1, 2]

และคำนวณหาในรูปแบบของเศษส่วนได้เป็นดังนี้

$$4 + \frac{1}{1 + \frac{1}{2}} = 4 + \frac{1}{\frac{3}{2}}$$

$$= 4 + \frac{2}{3}$$

$$= \frac{14}{3}$$

ข้อสังเกต: $\frac{14}{3}$ ซึ่งเป็นตัวเบนเข้าของ $\frac{33}{7}$ จะมีค่าที่ใกล้เคียงกันเป็นอย่างมาก

10. การประมาณค่าเศษส่วนต่อเนื่อง

การหาผลลัพธ์ของเศษส่วนต่อเนื่องจะหยุดเมื่อพบเศษที่มีค่าเป็น 1 อย่างไรก็ตามสามารถดำเนินการหาเศษส่วนต่อเนื่องต่อได้อีกโดยใช้วิธีการประมาณค่า

กำหนดให้ $\frac{h_t}{k_t}$ คือเศษส่วนต่อเนื่องที่เกิดจากการประมาณค่าจาก $\frac{1}{n}$ ในรอบที่ t เมื่อ $n \in \mathbb{R}$ โดยที่ในแต่ละรอบสามารถคำนวณหาได้โดยสมการที่ (5.6) และ สมการที่ (5.7)

$$h_i = a_i h_{i-1} + h_{i-2} \tag{5.6}$$

$$k_i = a_i k_{i-1} + k_{i-2} \tag{5.7}$$

เมื่อกำหนดให้ $h_{(-2)} = 1$, $h_{(-1)} = 0$, $k_{(-2)} = 0$ และ $k_{(-1)} = 1$

สำหรับการคำนวณหาค่า h_i และ k_i (เมื่อ i คือลำดับรอบของการคำนวณ) ในแต่ละรอบเป็นดังนี้

รอบที่ 1 (คำนวณ h_0 และ k_0): ดำเนินการตามลำดับดังต่อไปนี้

ลำดับที่ 1: $a_0 = \lfloor n \rfloor$

ลำดับที่ 2: $x = n - a_0$

ลำดับที่ 3: $y = \frac{1}{x}$

ลำดับที่ 4: $a_1 = \lfloor y \rfloor$

รอบที่ 2 เป็นต้นไป (คำนวณ h_{i-1} และ k_{i-1} เมื่อ i แทนผลลัพธ์รอบที่ i): ดำเนินการตามลำดับดังต่อไปนี้

ลำดับที่ 1: $x = y - a_{i-1}$

ลำดับที่ 2: $y = \frac{1}{x}$

ลำดับที่ 3: $a_i = \lfloor y \rfloor$

ตัวอย่างที่ 5.22 จงหาค่าประมาณเศษส่วนต่อเนื่องของ $\frac{1}{172.212347}$ จำนวน 3 ค่า

วิธีทำ จากขั้นตอนวิธีการประมาณค่าเศษส่วนต่อเนื่องได้ว่า

รอบที่ 1:

ลำดับที่ 1: $a_0 = \lfloor 172.212347 \rfloor = 172$

ลำดับที่ 2: $x = 172.212347 - 172 = 0.212347$

ลำดับที่ 3: $y = \frac{1}{0.212347} \approx 4.709273029522432622076130107795 3$

ลำดับที่ 4: $a_1 = \lfloor 4.709273029522432622076130107795 3 \rfloor = 4$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_0 = a_0 h_{(-1)} + h_{(-2)}$$
$$= 172x0 + 1 = 1$$
$$k_0 = a_0 k_{(-1)} + k_{(-2)}$$
$$= 172x1 + 0 = 172$$

รอบที่ 2:

ลำดับที่ 1: $x = y - a_1$

$$=4.709273029522432622076130107795 3 - 4$$

$$= 0.709273029522432622076130107795 3$$

ลำดับที่ 2: $y = \dfrac{1}{0.709273029522432622076130107795 3}$

$$\approx 1.409894297931107747058667304066$$

ลำดับที่ 3: $a_2 = \left\lfloor 1.409894297931107747058667304066 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_1 = a_1h_0 + h_{(-1)}$$

$$= 4x1 + 0 = 4$$

$$k_1 = a_1k_0 + k_{(-1)}$$

$$= 4x172 + 1 = 689$$

รอบที่ 3:

ลำดับที่ 1: $x = y - a_2$

$$= 1.409894297931107747058667304066 - 1$$

$$= 0.409894297931107747058667304066$$

ลำดับที่ 2: $y = \dfrac{1}{0.409894297931107747058667304066}$

$$\approx 2.439653357090791285332469425771 9$$

ลำดับที่ 3: $a_3 = \left\lfloor 2.439653357090791285332469425771 9 \right\rfloor = 2$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_2 = a_2h_1 + h_0$$

$$= 1x4 + 1 = 5$$

$$k_2 = a_2k_1 + k_0$$

$$= 1x689 + 172 = 861$$

ดังนั้น $\dfrac{1}{172.212347} \approx \dfrac{1}{172} \approx \dfrac{4}{689} \approx \dfrac{5}{861}$

11. บทสรุปสาระสำคัญ

เนื้อหาในบทนี้กล่าวถึงการนำนิยาม ทฤษฎีทางคณิตศาสตร์ และขั้นตอนวิธีที่สำคัญเพื่อนำมาประยุกต์ใช้กับวิทยาการรหัสลับอสมมาตรทั้งในส่วนของการเพิ่มความปลอดภัย เพิ่มความเร็ว และการโจมตี ประกอบไปด้วยดังต่อไปนี้ การก่อกำเนิดจำนวนเฉพาะซึ่งเป็นค่าที่สำคัญสำหรับการก่อกำเนิดกุญแจคู่ ทฤษฎีเศษเหลือจีนสามารถถูกนำมาใช้สำหรับการเพิ่มความเร็วในการคำนวณสมการยกกำลังมอดูลาร์เนื่องจากจะแบ่งเลขยกกำลังที่มีขนาดใหญ่ออกเป็นหลายๆ ค่าโดยที่แต่ละค่ามีขนาดเล็กลงเป็นอย่างมาก ฟังก์ชันออยเลอร์เป็นอีกปัจจัยหนึ่งที่สำคัญสำหรับการก่อกำเนิดกุญแจคู่สำหรับวิทยาการรหัสลับอาร์เอสเอซึ่งเป็นวิทยาการรหัสลับแบบอสมมาตรที่ได้รับความนิยมมากที่สุดในปัจจุบัน รากปฐมฐานคือนิยามที่สำคัญที่ใช้สำหรับขั้นตอนวิธีดิฟฟี่เฮลแมนซึ่งเป็นขั้นตอนวิธีที่ใช้สำหรับกระบวนการแลกเปลี่ยนกุญแจลับสำหรับวิทยาการรหัสลับแบบสมมาตร รวมไปถึงขั้นตอนวิธีในกลุ่มวิทยาการรหัสลับแบบอสมมาตร เช่นวิทยาการรหัสลับเส้นโค้งเชิงวงรี และเศษส่วนต่อเนื่องซึ่งเป็นเทคนิคที่สำคัญสำหรับขั้นตอนวิธีบางประเภทที่ถูกนำมาใช้สำหรับการโจมตีวิทยาการรหัสลับอาร์เอสเอ

แบบฝึกหัดท้ายบท
บทที่ 5

1. จงตรวจสอบว่า 29 เป็นจำนวนเฉพาะหรือจำนวนประกอบโดยใช้ขั้นตอนวิธีการทดลองหาร

2. จงตรวจสอบว่า 111 เป็นจำนวนเฉพาะหรือจำนวนประกอบโดยใช้ขั้นตอนวิธีการทดลองหาร

3. สมมติว่าผลลัพธ์จากการตรวจสอบจำนวนเต็มด้วยวิธีการทดสอบมิลเลอร์ – ราบินจำนวนสองรอบพบคำตอบในสมการที่ (5.1) หรือ (5.2) ทั้งสองรอบ จำนวนเต็มดังกล่าวมีโอกาสเป็นจำนวนเฉพาะด้วยความน่าจะเป็นที่เท่าไร

4. กำหนดให้ 227 คือจำนวนเฉพาะจงคำนวณหา $7^{230} \bmod 227$ โดยใช้ทฤษฎีบทเล็กของแฟร์มาต์

5. จงคำนวณหา $37^{4471} \bmod 2301$ โดยใช้วิธีเลขยกกำลังแบบเร็ว

6. จงคำนวณหา $37^{4471} \bmod 2301$ โดยใช้ขั้นตอนวิธียกกำลังสองและการคูณ

7. จงใช้ทฤษฎีเศษเหลือจีนเพื่อหาค่า x จากระบบสมการต่อไปนี้

$$x \equiv 3 \bmod 11$$
$$x \equiv 1 \bmod 23$$
$$x \equiv 18 \bmod 29$$

8. จงคำนวณหา $\Phi(23)$

9. กำหนดให้ 809 และ 997 เป็นจำนวนเฉพาะจงคำนวณหา $\Phi(809 \times 907)$

10. กำหนดให้ $p_1 = 3$, $p_2 = 13$ จงคำนวณหาค่า $5^{26} \bmod p_1p_2$ โดยใช้ทฤษฎีบทของออยเลอร์เครื่องเครื่องมือช่วย

11. กำหนดให้ 17 เป็นจำนวนเฉพาะจงคำนวณหา $\Phi(17^3)$ โดยใช้ทฤษฎีบทที่ 5.5 เป็นเครื่องเครื่องมือช่วย

12. กำหนดให้ 7 และ 11 เป็นจำนวนเฉพาะจงคำนวณหา $\Phi(7^4 \times 11^2)$ โดยใช้ทฤษฎีบทที่ 5.8 เป็นเครื่องเครื่องมือช่วย

13. จงคำนวณหา $7^{804768} \bmod 806573$

14. จากคำถามข้อ 1 และข้อ 2 จงตรวจสอบตัวเลขทั้งสองอีกครั้งโดยใช้ทฤษฎีบทเล็กของแฟร์มาต์สำหรับการตรวจสอบสถานะของผลลัพธ์และใช้เลขยกกำลังแบบเร็วเพื่อคำนวณหาผลลัพธ์ เมื่อกำหนดให้ $a = 11$ (เนื่องจาก gcd(7, 29) = 1 และ gcd(11, 111) = 1)

15. จงตรวจสอบว่า 4 เป็นรากปฐมฐานใน GF*(7) หรือไม่เพราะเหตุใด

16. จงตรวจสอบว่า 7 เป็นรากปฐมฐานใน GF*(11) หรือไม่เพราะเหตุใด

17. จงหาเศษส่วนต่อเนื่องของ $\frac{61}{13}$

18. จงพิจารณาหาตัวเบนเข้าของ $\frac{61}{13}$ เมื่อพิจารณาลำดับย่อย 3 ตำแหน่ง

บทที่ 6
วิทยาการรหัสลับแบบกุญแจสาธารณะ

ข้อดีของวิทยาการรหัสลับแบบสมมาตรคือความเรียบง่าย และรวดเร็ว อย่างไรก็ตามข้อเสียคือปัญหาของการแลกเปลี่ยนกุญแจลับระหว่างผู้รับ และผู้ส่งซึ่งอยู่ห่างไกลกัน จากปัญหาดังกล่าวจึงทำให้เกิดวิทยาการรหัสลับอีกชนิดหนึ่งเรียกว่า วิทยาการรหัสลับแบบอสมมาตรหรือที่นิยมถูกเรียกว่าวิทยาการรหัสลับแบบกุญแจสาธารณะ [7] ซึ่งมีแนวคิดคือมีกุญแจสองดอกที่ถูกนำมาใช้สำหรับกระบวนการเข้ารหัสลับ และกระบวนการถอดรหัสลับโดยกุญแจดอกหนึ่งจะถูกประกาศเป็นสาธารณะเรียกว่ากุญแจสาธารณะ และกุญแจอีกดอกหนึ่งจะถูกเก็บเป็นความลับโดยผู้สร้างกุญแจเรียกว่ากุญแจส่วนตัว โดยกุญแจทั้งสองค่านี้จะมีความสัมพันธ์ทางคณิตศาสตร์ซึ่งกันและกัน ดังนั้นหากนำข้อความลับมาเข้ารหัสลับด้วยกุญแจดอกหนึ่ง การถอดรหัสลับเพื่อให้ได้ข้อความต้นฉบับกลับคืนมาจำเป็นต้องใช้กุญแจอีกดอกหนึ่งที่เข้าคู่กันเท่านั้น การนำวิทยาการรหัสลับมาประยุกต์ใช้สำหรับการสื่อสารสามารถทำได้ดังนี้ ก่อนเริ่มสนทนาคู่สนทนาทั้งสองจำเป็นต้องประกาศกุญแจสาธารณะของตนเองไว้เป็นสาธารณะ โดยความเป็นจริงอาจมีแหล่งศูนย์กลางที่ใช้สำหรับเก็บกุญแจสาธารณะของทุกคน (ดังตัวอย่างใน ตารางที่ 6.1) หลังจากนั้นสมมติผู้ส่งต้องการส่งข้อความลับไปยังผู้รับ ผู้ส่งจะต้องไปร้องขอกุญแจสาธารณะของผู้รับจากแหล่งศูนย์กลางที่ใช้จัดเก็บกุญแจสาธารณะเพื่อนำมาใช้สำหรับการเข้ารหัสลับ หลังจากเสร็จสิ้นกระบวนการเข้ารหัสลับผู้ส่งจะได้ข้อความไซเฟอร์ซึ่งเป็นข้อความที่จะถูกส่งไปยังผู้รับ ทางฝั่งผู้รับหลังจากได้รับข้อความไซเฟอร์ผู้รับจะใช้กุญแจส่วนตัวสำหรับถอดรหัสลับออกมาซึ่งจะได้เป็นข้อความต้นฉบับ

สมมติ นาย ก ต้องการส่งข้อความลับ “secret” ไปยัง นาย ข ตัวอย่างการสนทนาโดยใช้วิทยาการรหัสลับแบบกุญแจสาธารณะเป็นดังนี้

ขั้นตอนที่ 1: นาย ก ร้องขอกุญแจสาธารณะของ นาย ข จากแหล่งศูนย์กลางที่ใช้สำหรับเก็บกุญแจสาธารณะดังตัวอย่างที่ถูกเก็บในตารางที่ 6.1

หลังจากนั้นแหล่งศูนย์กลางที่ใช้สำหรับเก็บกุญแจสาธารณะจะส่งค่ากุญแจสาธารณะของนาย ข ซึ่งมีค่าเป็น 345 กลับมายัง นาย ก

ขั้นตอนที่ 2: นาย ก นำค่า 345 มาใช้สำหรับเข้ารหัสลับข้อความ “secret” สมมติหลังจากเสร็จสิ้นกระบวนการเข้ารหัสลับได้ข้อความไซเฟอร์เป็น “axtz”

ขั้นตอนที่ 3: นาย ก ส่ง “axtz”ไปยัง นาย ข

ขั้นตอนที่ 4: หลังจากที่ นาย ข ได้รับ “axtz” แล้ว นาย ข จะใช้กุญแจส่วนตัวของตนเองที่มีความสัมพันธ์ทางคณิตศาสตร์กับค่า 345 และมีเพียงค่าเดียวเท่านั้น (สมมติค่ากุญแจส่วนตัวของ นาย ข คือ 731) สำหรับกระบวนการถอดรหัสลับ หลังจากเสร็จสิ้นกระบวนการถอดรหัสลับ นาย ข จะได้รับข้อความต้นฉบับที่ A ส่งมาให้คือ “secret”

ในทางกลับกันหาก นาย ข ต้องการส่งข้อความลับกลับไปหา นาย ก ก่อนเริ่มการสนทนา นาย ข จำเป็นต้องร้องขอกุญแจสาธารณะของ นาย ก เพื่อดำเนินการต่อ ซึ่งกระบวนการเป็นดังขั้นตอนที่ 1 ถึง 4 ข้างต้น

สมมติ นาย ค ซึ่งเป็นบุคคลที่ 3 สามารถดักจับข้อมูลที่ นาย ก ส่งมาให้ นาย ข ได้ ส่งผลให้ นาย ค ทราบข้อความไซเฟอร์ จากนาย ก คือ “axtz” อย่างไรก็ตาม นาย ค ไม่สามารถทราบข้อความต้นฉบับจาก นาย ก ได้เนื่องจาก นาย ค ไม่ทราบค่ากุญแจส่วนตัวที่ นาย ข เก็บไว้เป็นความลับแต่เพียงผู้เดียว

ตารางที่ 6.1 ตัวอย่างแหล่งศูนย์กลางที่ใช้สำหรับเก็บกุญแจสาธารณะ

เจ้าของกุญแจ	กุญแจสาธารณะ
นาย ก	123
นาย ข	345
นาย ค	567
นาย ง	789

อย่างไรก็ตามเนื่องจากกระบวนการเข้ารหัสลับแบบกุญแจสาธารณะมีความซับซ้อนค่อนข้างสูง หากเปรียบเทียบกับวิทยาการรหัสลับแบบสมมาตร ดังนั้นในทางปฏิบัติจึงนิยมนำวิทยาการรหัสลับแบบกุญแจสาธารณะมาใช้เป็นกระบวนการแลกเปลี่ยนกุญแจลับสำหรับวิทยาการรหัสลับแบบสมมาตร และใช้วิทยาการรหัสลับแบบสมมาตรสำหรับการสื่อสารระหว่างผู้รับ และ ผู้ส่ง

สำหรับบทนี้จะกล่าววิทยาการรหัสลับแบบกุญแจสาธารณะสองประเภทคือ กระบวนการแลกเปลี่ยนกุญแจที่นำเสนอโดยวิธฟิลด์ ดิฟฟี่ (Whitfield Diffie) และ มาร์ติน เฮลแมน (Martin Hellman) ซึ่งเป็นขั้นตอนวิธีแรกของวิทยาการรหัสลับแบบกุญแจสาธารณะ และวิทยาการรหัสลับเอ็ลกามอล (Elgamal Cryptography) ซึ่งเป็นวิทยาการรหัสลับแบบกุญแจสาธารณะที่ปรับปรุงจากขั้นตอนวิธีที่นำเสนอโดยดิฟฟี่และเฮลแมน

1. ขั้นตอนวิธีดิฟฟี่เฮลแมนสำหรับการแลกเปลี่ยนกุญแจ

ในปี ค.ศ. 1976 วิธฟิลด์ ดิฟฟี่ และ มาร์ติน เฮลแมน [7] ได้นำเสนอขั้นตอนวิธีแรกสำหรับวิทยาการรหัสลับแบบกุญแจสาธารณะ โดยนำเสนอผลงานในรูปแบบของบทความในงานประชุมวิชาการ National Computer Conference และได้รับการพิจารณาตีพิมพ์ลงวารสาร IEEE Transaction on Information Theory ในเวลาต่อมาและเรียกขั้นตอนวิธีดังกล่าวนี้ว่าขั้นตอนวิธีวิธีดิฟฟี่เฮลแมน อย่างไรก็ตามขั้นตอนวิธีดิฟฟี่เฮลแมนไม่สามารถนำมาใช้สำหรับกระบวนการเข้ารหัสลับ และถอดรหัสลับได้ แต่สามารถนำมาใช้สำหรับกระบวนแลกเปลี่ยนกุญแจลับสำหรับวิทยาการรหัสลับแบบสมมาตรเท่านั้น

สมมติ นาย ก และ นาย ข ประสงค์ที่จะแลกเปลี่ยนกุญแจลับซึ่งกันและกันด้วยขั้นตอนวิธีดิฟฟี่เฮลแมน มีขั้นตอนเป็นดังนี้

ขั้นตอนที่ 1: นาย ก และ นาย ข ตกลงเลือกใช้จำนวนเฉพาะ p และเลือกรากปฐมฐาน g ภายใต้การมอดูโล p

ขั้นตอนที่ 2: นาย ก สุ่มเลือกจำนวนเต็ม $a \in \{0, 1, 2, \cdots, p-2\}$ และคำนวณ A จาก

$$A = g^a \bmod p$$

โดยนาย ก จะเก็บ a ไว้เป็นความลับ แต่จะส่ง A ไปยังนาย ข

ขั้นตอนที่ 3: นาย ข สุ่มเลือกจำนวนเต็ม $b \in \{0, 1, 2, \cdots, p-2\}$ และคำนวณ B จาก

$$B = g^b \bmod p$$

โดยนาย ข จะเก็บ b ไว้เป็นความลับ แต่จะส่ง B ไปยังนาย ก

ขั้นตอนที่ 4: นาย ก นำค่า B ที่รับมาจากนาย ข และคำนวณ $K_ก$ จาก

$$K_ก = B^a \bmod p \tag{6.1}$$

ขั้นตอนที่ 5: นาย ข นำค่า A ที่รับมาจากนาย ก และคำนวณ $K_ข$ จาก

$$K_ข = A^b \bmod p \tag{6.2}$$

จากสมการที่ (6.1) และ (6.2) กล่าวได้ว่า $K_ก = K_ข = K$ เนื่องจาก

จากสมการ (6.1), $K_ก = B^a \bmod p = (g^b)^a \bmod p = g^{ab} \bmod p$

จากสมการ (6.2), $K_ข = A^b \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p$

ดังนั้นสรุปได้ว่า K คือกุญแจลับสำหรับการการสนทนาระหว่าง นาย ก และ นาย ข

ตัวอย่างที่ 6.1 กระบวนการแลกเปลี่ยนกุญแจด้วยขั้นตอนวิธีดิฟฟี่เฮลแมน

วิธีทำ

ขั้นตอนที่ 1: นาย ก และ นาย ข ตกลงใช้ค่า $p = 29$ และ $g = 3$ ร่วมกัน

ขั้นตอนที่ 2: นาย ก เลือก $a = 8$ และ คำนวณ $A = g^a \bmod p = 3^8 \bmod 29 = 7$ และส่งไปยังนาย ข

ขั้นตอนที่ 3: นาย ข เลือก $b = 13$ และ คำนวณ $B = g^b \bmod p = 3^{13} \bmod 29 = 19$ และส่งไปยัง นาย ก

ขั้นตอนที่ 4: นาย ก คำนวณ $K_ก = B^a \bmod p = 19^8 \bmod 29 = 25$

ขั้นตอนที่ 5: นาย ข คำนวณ $K_ข = A^b \bmod p = 7^{13} \bmod 29 = 25$

ดังนั้นสรุปได้ว่า $K = 25$ คือกุญแจลับที่ใช้ร่วมกันระหว่างนาย ก และ นาย ข

ความปลอดภัยของขั้นตอนวิธีดิฟฟี่เฮลแมนขึ้นอยู่กับความยากของการค้นหา a หรือ b เพื่อคำนวณหาค่า K ตัวอย่างเช่น สมมติผู้ไม่ประสงค์ดีทราบค่า a จะสามารถคำนวณหาค่า K ได้จาก $K = B^a \bmod p$ (เนื่องจากผู้ไม่ประสงค์ดีทราบค่า B และ p) หรือหากผู้ไม่ประสงค์ดีทราบค่า b จะสามารถคำนวณหาค่า K ได้จาก $K = A^b \bmod p$ (เนื่องจากผู้ไม่ประสงค์ดีทราบค่า A และ p) อย่างไรก็ตาม การค้นหา a หรือ b จำเป็นต้องใช้เวลานานมหาศาลโดยเฉพาะอย่างยิ่งหาก p มีขนาดใหญ่เรียกปัญหาดังกล่าวว่าปัญหาดิฟฟี่เฮลแมน (Diffie - Hellman Problem: DHP) หรือ เรียก a (หรือ b) ว่าเป็นปัญหาวิยุตลอการิทึม (Discrete Logarithm Problem, DLP) ของ A (หรือ B) ฐาน g

อย่างไรก็ตามยังมีการโจมตีอีกวิธีหนึ่งเรียกว่า มนุษย์คุกคามระหว่างกลาง (man in the middle attack) ซึ่งผู้ไม่ประสงค์ดีไม่จำเป็นต้องทราบค่ากุญแจลับของคู่สนทนา แต่จะเข้าไปแทรกแซงการสนทนา สมมตินาย ค เป็นผู้ไม่ประสงค์ดีที่ต้องการทราบบทสนทนาระหว่างนาย ก และนาย ข หลักการคือ นาย ค สวมรอยเป็น นาย ข เพื่อแลกเปลี่ยนกุญแจลับร่วมกับ นาย ก (โดยที่นาย ก เข้าใจว่ากำลังแลกเปลี่ยนกุญแจลับกับนาย ข) กำหนดเป็น $K_{กค}$ ในทางกลับกันนาย ค สวมรอยเป็นนาย ก เพื่อแลกเปลี่ยนกุญแจลับร่วมกับ นาย ข (โดยที่นาย ข เข้าใจว่ากำลังแลกเปลี่ยนกุญแจลับกับนาย ก) กำหนดเป็น $K_{ขค}$ ดังนั้นหลังจากเสร็จสิ้นกระบวนการนาย ค จะถือกุญแจลับ 2 ค่า สมมตินาย ก ประสงค์จะส่งข้อความลับ m_1 ไปยังนาย ข จึงนำกุญแจ $K_{กค}$ เพื่อเข้ารหัส m_1 ได้เป็น c_1 เพื่อส่งไปยังนาย ข แต่นาย ค ได้ดักจับ c_1 ระหว่างกลางก่อนที่ข้อมูลจะถูกส่งไปยงนาย ข และใช้กุญแจ $K_{กค}$ เพื่อถอดรหัสทำให้นาย ค ทราบว่านาย ก กำลังส่ง m_1 ไปยังนาย ข หลังจากนั้นนาย ค เข้ารหัส m_1 ด้วยกุญแจ $K_{ขค}$ ได้เป็น c_2 และส่งค่าดังกล่าวไปยังนาย ข เมื่อได้รับข้อมูลนาย ข สามารถถอดรหัสได้

เป็น m_1 กลับมา จากวิธีดังกล่าวบทสนทนาระหว่างนาย ก และนาย ข ยังคงเดิมเพียงแต่คู่สนทนาไม่ทราบว่ามีบุคคลที่สามซึ่งคือนาย ค สามารถรับรู้ข้อความลับระหว่างคู่สนทนาด้วยเช่นกัน

การป้องกันการคุกคามจากผู้ไม่ประสงค์ดีที่ใช้วิธีมนุษย์คุกคามระหว่างกลางสามารถทำได้โดยใช้ลายเซ็นดิจิทัลโดยจะกล่าวถึงในบทที่ 10

2. วิทยาการรหัสลับเอ็ลกามอล (Elgamal Cryptography)

วิทยาการรหัสลับเอ็ลกามอล [24] เป็นวิทยาการรหัสลับแบบกุญแจสาธารณะที่เสนอโดย ทาเฮอร์ เอ็ลกามอล (Taher Elgamal) นักวิทยาการรหัสลับชาวอียิป ใน ค.ศ. 1985 โดยมีการปรับปรุงเพิ่มจากขั้นตอนวิธีดิฟฟี่เฮลแมนเพื่อให้สามารถนำไปใช้สำหรับกระบวนการเข้ารหัส และถอดรหัสได้

รหัสลับเอ็ลกามอลถูกแบ่งออกเป็น 3 กระบวนการ ดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจ หรือผู้รอรับข้อความไซเฟอร์ (กำหนดเป็น ผู้รับ) มีลำดับการทำงานเป็นดังนี้

1. เลือกจำนวนเฉพาะ p และรากปฐมฐาน g ภายใต้การมอดูโล p
2. เลือก $a \in \{0, 1, 2, \cdots, p-2\}$
3. คำนวณ A จาก $A = g^a \bmod p$

กุญแจสาธารณะคือ $\{p, g, A\}$

กุญแจส่วนตัวคือ $\{a\}$

กระบวนการที่ 2 การเข้ารหัสลับ: เป็นกระบวนการที่ถูกดำเนินการโดยบุคคล (กำหนดเป็นผู้ส่ง) ผู้ซึ่งต้องการส่งข้อความลับ (สมมติข้อความลับคือ m) ไปยังผู้รับมีลำดับการทำงานเป็นดังนี้

1. เลือก $b \in \{0, 1, 2, \cdots, p-2\}$
2. คำนวณ B จาก
$$B = g^b \bmod p$$
3. คำนวณข้อความไซเฟอร์ c จาก
$$c = A^b m \bmod p$$

หลังเสร็จสิ้นกระบวนการในขั้นตอนที่ 3 ผู้ส่งจะส่ง $\{c, B\}$ ไปยังผู้รับและเก็บ b ไว้เป็นความลับ

กระบวนการที่ 3 การถอดรหัสลับ: หลังจากที่ผู้รับได้รับ $\{c, B\}$ จากผู้ส่งจะสามารถคำนวณหา m ได้โดยสมการต่อไปนี้

$$m = (B^{-1})^a c \bmod p$$

หลังจากการถอดรหัสด้วยวิทยาการรหัสลับเอ็ลกามอลจะได้ค่า m กลับคืนเสมอ เนื่องจาก

$$\begin{aligned}(B^{-1})^a c \bmod p &= B^{-a} c \bmod p \\ &= (g^b)^{-a} c \bmod p \\ &= (g^{-ab}) c \bmod p \\ &= (g^{-ab})(A^b m) \bmod p \\ &= (g^{-ab})(g^{ab} m) \bmod p \\ &= (g^{-ab})(g^{ab})\, m \bmod p \\ &= m\end{aligned}$$

ตัวอย่างที่ 6.2 การประยุกต์ใช้วิทยาการรหัสลับเอ็ลกามอล

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

1. เลือกจำนวนเฉพาะ $p = 37$ และรากปฐมฐาน $g = 13$
2. เลือก $a = 7$
3. คำนวณ $A = 13^7 \bmod 37$
   โดยใช้วิธีเลขยกกำลังแบบเร็ว ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 7 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$$7 = 1x2^2 + 1x2^1 + 1x2^0$$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$$7 = 1x2^2 + 1x2^1 + 1x2^0$$

หรือ $7 = 2^2 + 2^1 + 1$

จากขั้นตอนที่ 2 ได้ว่า

$$\begin{aligned}13^7 &= 13^{2^2+2^1+2^0} \\ &= 13^{2^2} \times 13^{2^1} \times 13^1\end{aligned}$$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $13^2 \bmod 37 = 21$

ครั้งที่ 2: คำนวณ $13^{2^2} \bmod 37$, เนื่องจาก $13^{2^2} \equiv 21^2 \bmod 37$

ดังนั้น $21^2 \bmod 37 = 34$

ดังนั้น $13^7 \bmod 37 = 13^{2^2} \times 13^{2^1} \times 13^1 \bmod 37$

$= 34 \times 21 \times 13 \bmod 37$

$= 32$

กุญแจสาธารณะคือ {$p = 37$, $g = 13$, $A = 32$}

กุญแจส่วนตัวคือ {$a = 7$}

กระบวนการเข้ารหัส

สมมติผู้ส่งต้องการส่ง $m = 13$

1. เลือก $b = 15$
2. คำนวณ $B = 13^{15} \bmod 37$

   โดยใช้วิธีเลขยกกำลังแบบเร็ว ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 15 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$15 = 1x2^3\ 1x2^2 + 1x2^1 + 1x2^0$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$15 = 1x2^3\ 1x2^2 + 1x2^1 + 1x2^0$

หรือ $15 = 2^3 + 2^2 + 2^1 + 2^0$

จากขั้นตอนที่ 2 ได้ว่า

$13^{15} = 13^{2^3+2^2+2^1+2^0}$

$= 13^{2^3} \times 13^{2^2} \times 13^{2^1} \times 13^1$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $13^2 \bmod 37 = 21$

ครั้งที่ 2: คำนวณ $13^{2^2} \mod 37$, เนื่องจาก $13^{2^2} \equiv 21^2 \mod 37$

ดังนั้น $21^2 \mod 37 = 34$

ครั้งที่ 3: คำนวณ $13^{2^3} \mod 37$, เนื่องจาก $13^{2^3} \equiv 34^2 \mod 37$

ดังนั้น $34^2 \mod 37 = 9$

ดังนั้น $13^{15} \mod 37 = 13^{2^3} \times 13^{2^2} \times 13^{2^1} \times 13^1 \mod 37$

$= 9 \times 34 \times 21 \times 13 \mod 37$

$= 29$

3. คำนวณ $c = 32^{15} \times 13 \mod 37$

โดยใช้วิธีเลขยกกำลังแบบเร็วเพื่อคำนวณ $32^{15} \mod 37$ ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 15 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$15 = 1x2^3\ 1x2^2 + 1x2^1 + 1x2^0$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$15 = 1x2^3\ 1x2^2 + 1x2^1 + 1x2^0$

หรือ $15 = 2^3 + 2^2 + 2^1 + 2^0$

จากขั้นตอนที่ 2 ได้ว่า

$32^{15} = 32^{2^3+2^2+2^1+2^0}$

$= 32^{2^3} \times 32^{2^2} \times 32^{2^1} \times 32^1$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $32^2 \mod 37 = 25$

ครั้งที่ 2: คำนวณ $32^{2^2} \mod 37$, เนื่องจาก $32^{2^2} \equiv 25^2 \mod 37$

ดังนั้น $25^2 \mod 37 = 33$

ครั้งที่ 3: คำนวณ $32^{2^3} \mod 37$, เนื่องจาก $32^{2^3} \equiv 33^2 \mod 37$

ดังนั้น $33^2 \mod 37 = 16$

ดังนั้น $32^{15} \bmod 37 = 32^{2^3} \times 32^{2^2} \times 32^{2^1} \times 32^1 \bmod 37$

$= 16 \times 33 \times 25 \times 32 \bmod 37$

$= 8$

ดังนั้น $c = 32^{15} \times 13 \bmod 37$

$= 8 \times 13 \bmod 37$

$= 30$

ส่ง $\{c = 30, B = 29\}$ ไปยังผู้รับ

กระบวนการถอดรหัส

เมื่อรับข้อความลับ $\{c = 30, B = 29\}$ จากผู้ส่งแล้วผู้รับต้องคำนวณหา B^{-1} ก่อนโดยใช้ขั้นตอนวิธียุคลิดภาคขยาย ได้ว่า $B^{-1} = 23$ เนื่องจาก

$$23 \times 29 \bmod 37 = 1$$

หลังจากทราบค่า B^{-1} ผู้รับสามารถคำนวณหา m โดยใช้สมการการถอดรหัสดังนี้

$$(B^{-1})^a c \bmod p = 23^7 \times 30 \bmod 37 = 13$$

ใช้วิธีเลขยกกำลังแบบเร็วเพื่อคำนวณ $23^7 \bmod 37$ ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 7 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$7 = 1x2^2 + 1x2^1 + 1x2^0$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$7 = 1x2^2 + 1x2^1 + 1x2^0$

หรือ $7 = 2^2 + 2^1 + 1$

จากขั้นตอนที่ 2 ได้ว่า

$23^7 = 23^{2^2+2^1+2^0}$

$= 23^{2^2} \times 23^{2^1} \times 23^1$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $23^2 \bmod 37 = 11$

ครั้งที่ 2: คำนวณ $23^{2^2} \bmod 37$, เนื่องจาก $23^{2^2} \equiv 11^2 \bmod 37$

ดังนั้น $11^2 \bmod 37 = 10$

ดังนั้น $23^7 \bmod 37 = 23^{2^2} \times 23^{2^1} \times 23^1 \bmod 37$

$= 10 \times 11 \times 23 \bmod 37$

$= 14$

ดังนั้น $m = 23^7 \times 30 \bmod 37$

$= 14 \times 30 \bmod 37$

$= 13$

จากกระบวนการถอดรหัสสังเกตได้ว่าผู้รับจำเป็นต้องคำนวณหา B^{-1} ซึ่งคือค่าผกผันของ B มอดูโล p อย่างไรก็ตามผู้รับสามารถหลีกเลี่ยงการใช้ตัวผกผันสำหรับสมการถอดรหัสลับ โดยใช้สมการดังต่อไปนี้

$$m = B^x c \bmod p \tag{6.3}$$

เมื่อ $x = p - 1 - a$

จากสมการที่ (6.3) สามารถพิสูจน์ได้ดังต่อไปนี้

$$B^x c \bmod p = B^{(p-1-a)} c \bmod p$$

$$= g^{b(p-1-a)} c \bmod p$$

$$= g^{b(p-1)} g^{-ab} c \bmod p$$

จากทฤษฎีบทเล็กของแฟร์มาต์ได้ว่า

$$B^x c \bmod p = g^{-ab} c \bmod p$$

แทนค่า $c = A^b m$, $= g^{-ab} A^b m \bmod p$

$$= g^{-ab} g^{ab} m \bmod p$$

$$= m \bmod p$$

ตัวอย่างที่ 6.3 ทดสอบถอดรหัสข้อความไซเฟอร์จากตัวอย่างที่ 6.2 โดยใช้สมการ (6.3)

วิธีทำ เริ่มจากคำนวณ $x = p - 1 - a = 37 - 1 - 7 = 29$

จากสมการ (6.1) ได้ว่า

$$m = B^x c \bmod p$$
$$= 29^{29} \times 30 \bmod 37$$

ใช้วิธีเลขยกกำลังแบบเร็วเพื่อคำนวณ $29^{29} \bmod 37$ ดังนี้

ขั้นตอนที่ 1: แปลงยกกำลัง 29 ให้อยู่ในรูปแบบสมการเลขฐานสอง ดังนี้

$$29 = 1x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$$

ขั้นตอนที่ 2: ตัดเลขฐานสองที่มีค่าเป็น 0 ออกจากการคำนวณ

$$29 = 1x2^4 + 1x2^3 + 1x2^2 + 1x2^0$$

หรือ $29 = 2^4 + 2^3 + 2^2 + 2^0$

จากขั้นตอนที่ 2 ได้ว่า

$$29^{29} = 29^{2^4+2^3+2^2+2^0}$$
$$= 29^{2^4} \times 29^{2^3} \times 29^{2^2} \times 29^{2^0}$$

ขั้นตอนที่ 3: คำนวณหาผลลัพธ์ย่อยในแต่ละส่วน ดังนี้

ครั้งที่ 1: คำนวณ $29^2 \bmod 37 = 27$

ครั้งที่ 2: คำนวณ $29^{2^2} \bmod 37$, เนื่องจาก $29^{2^2} \equiv 27^2 \bmod 37$

ดังนั้น $27^2 \bmod 37 = 26$

ครั้งที่ 3: คำนวณ $29^{2^3} \bmod 37$, เนื่องจาก $29^{2^3} \equiv 26^2 \bmod 37$

ดังนั้น $26^2 \bmod 37 = 10$

ครั้งที่ 4: คำนวณ $29^{2^4} \bmod 37$, เนื่องจาก $29^{2^4} \equiv 10^2 \bmod 37$

ดังนั้น $10^2 \bmod 37 = 26$

ดังนั้น $29^{29} \bmod 37 = 29^{2^4} \times 29^{2^3} \times 29^{2^2} \times 29^{2^0} \bmod 37$

$$= 26 \times 10 \times 26 \times 29 \bmod 37$$
$$= 14$$

ดังนั้น $m = 29^{29} \times 30 \bmod 37$

$= 14 \times 30 \bmod 37$

$= 13$

3. ความปลอดภัยของวิทยาการรหัสลับเอ็ลกามอลและขั้นตอนวิธีดิฟฟี่เฮลแมน

ความปลอดภัยของวิทยาการรหัสลับเอ็ลกามอลและขั้นตอนวิธีดิฟฟี่เฮลแมนมีลักษณะที่คล้ายคลึงกันกล่าวคือหากผู้ไม่ประสงค์ดีทราบค่า a ของ A ฐาน g ส่งผลให้ผู้ไม่ประสงค์ดีสามารถใช้สมการถอดรหัสเพื่อใช้สำหรับการถอดรหัส m กลับมาได้ อย่างไรก็ตามค่าของ p ที่ถูกนำมาใช้จริงมีขนาดใหญ่มหาศาล จึงเป็นเรื่องยากมากสำหรับผู้ไม่ประสงค์ดีที่คำนวณหาค่า a จากค่า A, g และ p ที่ถูกเปิดเผยเป็นสาธารณะ

ในหัวข้อนี้จะกล่าวถึงขั้นตอนวิธีสำหรับแก้ปัญหาวิยุตลอการิทึมที่สามารถโจมตีได้ทั้งขั้นตอนวิธีดิฟฟี่เฮลแมนและวิทยาการรหัสลับเอ็ลกามอลในทางทฤษฎี แต่อย่างไรก็ตามในทางปฏิบัติยังไม่พบขั้นตอนวิธีใดที่สามารถโจมตีทั้งสองขั้นตอนวิธีนี้ได้หากค่า p มีขนาดใหญ่และมีความแข็งแกร่ง

กำหนดให้ $A = g^a \bmod p$ เมื่อ A, g และ p คือพารามิเตอร์ที่ถูกเปิดเผยเป้าหมายของขั้นตอนวิธีสำหรับโจมตีทั้งสองขั้นตอนวิธีคือค่า a ซึ่งสามารถดำเนินการค้นหาได้โดยใช้ขั้นตอนวิธีที่ใช้สำหรับแก้ปัญหาวิยุตลอการิทึมดังตัวอย่างที่จะนำเสนอต่อไปนี้

3.1 การโจมตีแบบตะลุย (Brute Force Attack)

การโจมตีแบบตะลุยคือขั้นตอนวิธีที่มีขั้นตอนการดำเนินงานที่เรียบง่ายไม่ซับซ้อนโดยเริ่มจากการกำหนดค่า x ที่เป็นไปได้และมีค่าน้อยที่สุด ($x = 2$) และคำนวณหาค่า $A' = g^x \bmod p$ โดยหากผลลัพธ์ของ A' มีค่าเท่ากับ A แสดงว่า x คือค่า a อย่างไรก็ตามหากผลลัพธ์ที่ได้มีค่าไม่เท่ากันจำเป็นต้องเพิ่มค่า x ขึ้นเพื่อคำนวณหาค่า A' ใหม่ ซึ่งจะดำเนินการลักษณะเดิมจนกระทั่งผลลัพธ์ของ A' และ A มีค่าเท่ากันถูกตรวจพบจึงจะหยุดการทำงาน โดยมีขั้นตอนวิธีเป็นดังนี้

ขั้นตอนวิธีที่ 6.1 การโจมตีแบบตะลุย

```
     INPUT: A, g, p
     OUTPUT: a
1:   x ← 2
2:   A' ← g^x mod p
3:   While (A' is not equal to A) do
4:       x ← x + 1
5:       A' ← A'*g mod p
6:   End While
7:   a ← x
```

ตัวอย่างที่ 6.4 กำหนดให้ $A = 12$, $p = 23$ และ $g = 3$ จงคำนวณหา a โดยใช้ขั้นตอนวิธีการโจมตีแบบตะลุย

วิธีทำ จากขั้นตอนวิธีการโจมตีแบบตะลุยได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

1. $x = 2$
2. $A' = 3^2 \bmod 23 = 9$

ขั้นตอนที่ 3 – 6 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก $9 \neq 12$ ดังนั้น

รอบที่ 1:

4. $x = 2 + 1 = 3$
5. $A' = 9 \times 3 \bmod 23 = 4$

เนื่องจาก $4 \neq 12$ ดังนั้น

รอบที่ 2:

4. $x = 3 + 1 = 4$
5. $A' = 4 \times 3 \bmod 23 = 12$

เนื่องจาก $12 = 12$ ดังนั้นสรุปได้ว่า $a = 4$

ข้อดีของการโจมตีแบบตะลุยคือสามารถตะลุยหา a ได้อย่างรวดเร็วในกรณีที่ a มีขนาดเล็ก อย่างไรก็ตามวิธีนี้มีประสิทธิภาพต่ำมากในกรณีที่ a มีค่าที่ใหญ่มหาศาล สำหรับการประยุกต์ใช้จริง ขนาดของ p จะไม่น้อยกว่า 128 บิต ส่งผลให้การตะลุยหาค่า a โดยเฉลี่ยจะมีค่าเป็น $\frac{p-1}{2}$ ซึ่งใช้เวลาในการประมวลผลมหาศาล

3.2 ขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ (Baby-Step Giant-Step)

กำหนดให้ $m = \left\lceil \sqrt{p} \right\rceil$ และ $a = sm + t$ เมื่อ $0 \leq s, t < m$ ดังนั้น

จาก $A = g^a \bmod p$

ได้ว่า $= g^{sm+t} \bmod p$

$= g^{sm} g^t \bmod p$

ดังนั้น $g^{-sm} A = g^{-sm} g^{sm} g^t \bmod p$

$= g^t \bmod p$

จากสมการข้างต้นสามารถคำนวณหา a ได้โดยการสร้างตารางเก็บข้อมูลจำนวน 2 ตาราง เพื่อเก็บค่าที่เป็นไปได้ทั้งหมดของ $g^t \bmod p$ และ $g^{-sm}A \bmod p$ โดยค่า a สามารถคำนวณได้จากการพิจารณาผลลัพธ์ของแถวใดๆ จากทั้ง 2 ตารางที่มีค่าตรงกัน

ตารางที่ 6.2 การหา $g^t \bmod p$ เมื่อ $t = 0, 1, 2, \cdots, m$

t	$g^t \bmod p$
0	$g^0 \bmod p$
1	$g^1 \bmod p$
2	$g^2 \bmod p$
⋮	⋮
m	$g^m \bmod p$

ตารางที่ 6.3 การหา $g^{-sm}A \bmod p$ เมื่อ $s = 0, 1, 2, \cdots, m$

s	$g^{-sm}A \bmod p$
0	$A \bmod p$
1	$g^{-m}A \bmod p$
2	$g^{-2m}A \bmod p$
⋮	⋮
m	$g^{-m^2}A \bmod p$

หากพิจารณาตารางที่ 6.2 สังเกตได้ว่าเลขยกกำลังถูกเพิ่มขึ้นเพียงรอบละ 1 ค่าซึ่งเป็นการเพิ่มทีละเล็กน้อยจึงเรียกว่า เบบี้สเต็พ (Baby-Step) ในทางกลับกันหากพิจารณาตารางที่ 6.3 อัตราการเพิ่มของเลขยกกำลังในแต่ละแถวมีค่าเป็น $-m$ ซึ่งเป็นการเพิ่มขึ้นอย่างมากจึงเรียกว่า ไจแอ็นสเต็พ (Giant-Step) โดยทั้งสองตารางจำเป็นต้องถูกพิจารณาร่วมกันจึงเรียกวิธีดังกล่าวนี้ว่าเบบี้สเต็พไจแอ็นสเต็พ [59]

ผลลัพธ์ดังตารางที่ 6.2 จำเป็นต้องถูกสร้างเสมอเพื่อเก็บค่าที่เป็นไปได้ทั้งหมดของ $g^t \bmod p$ แต่ในทางกลับกันการคำนวณหา $g^{-sm}A \bmod p$ ไม่จำเป็นต้องคำนวณหาค่าที่เป็นไปได้ทั้งหมด โดย

หากพบผลลัพธ์ที่มีค่าตรงกับค่าใดค่าหนึ่งในตารางที่ 6.2 จะหยุดการคำนวณหาผลลัพธ์ของตารางที่ 6.3 และสามารถคำนวณหา a ได้ โดยจากหลักการของการพบกันตรงครึ่งทางซึ่งความหมายคือค้นหาผลลัพธ์จากตารางที่สองครึ่งหนึ่งของทั้งหมดแล้วพบคำตอบจะได้ว่าจำนวนแถวของทั้งสองตารางที่จำเป็นต้องถูกคำนวณทั้งหมดมีค่าเป็น m (จำนวนแถวของตารางที่ 6.2) + $\frac{m}{2}$ (จำนวนแถวของตารางที่ 6.3)

สำหรับตารางที่ 6.3 จำเป็นต้องคำนวณหาค่าของ $g^{-1} \bmod p$ เพื่อใช้ในการคำนวณหาค่าของ $g^{-m} \bmod p$ ซึ่งเป็นค่าที่ใช้เป็นตัวคูณสำหรับแต่ละแถว

ตัวอย่างที่ 6.5 กำหนดให้ $A = 21$, $p = 29$ และ $g = 11$ จงคำนวณหา a โดยใช้ขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ

วิธีทำ เนื่องจาก $m = \lceil \sqrt{29} \rceil = 6$ ดังนั้นจากขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ ลำดับแรกสร้างตารางเพื่อหา $g^t \bmod p$ เมื่อ $0 \leq t < 6$

ใช้วิธีการคูณมอดูลาร์ ดังนี้

ครั้งที่ 1: คำนวณ $g \bmod p$, $11 \bmod 29 = 11$

ครั้งที่ 2: คำนวณ $g^2 \bmod p$, $11^2 \bmod 29 = 5$

ครั้งที่ 3: คำนวณ $g^3 \bmod p = g^2 g \bmod p$, $5 \times 11 \bmod 29 = 26$

ครั้งที่ 4: คำนวณ $g^4 \bmod p = g^3 g \bmod p$, $26 \times 11 \bmod 29 = 25$

ครั้งที่ 5: คำนวณ $g^5 \bmod p = g^4 g \bmod p$, $25 \times 11 \bmod 29 = 14$

ครั้งที่ 6: คำนวณ $g^6 \bmod p = g^5 g \bmod p$, $14 \times 11 \bmod 29 = 9$

จึงได้ตารางสำหรับ $g^t \bmod p$ เมื่อ $0 \leq t < 6$ เป็นดังนี้

t	$g^t \bmod p$
0	1
1	11
2	5
3	26
4	25
5	14
6	9

ขั้นตอนถัดไปสร้างตารางเพื่อหาค่าของ $g^{-sm}A \bmod p$ เมื่อ $0 \leq s < 6$ อย่างไรก็ตามขั้นตอนแรกจำเป็นต้องคำนวณหา $g^{-1} \bmod p$ ซึ่งมีค่าเป็น 8 เนื่องจาก 8 x 11 mod 29 = 1 และสามารถคำนวณหา $g^{-m} \bmod p$ ได้จาก $g^{-m} = (g^{-1})^m = (g^{-1})^6 = 8^6 \bmod 29 = 13$ ที่จะถูกใช้เป็นตัวคูณสำหรับแต่ละแถวได้ดังนี้

ใช้วิธีการคูณมอดูลาร์ ดังนี้

ครั้งที่ 1: คำนวณ $g^{-m}A \bmod p$, 13 x 21 mod 29 = 12

ครั้งที่ 2: คำนวณ $g^{-2m}A \bmod p = g^{-m}g^{-m}A \bmod p$, 13 x 12 mod 29 = 11

ครั้งที่ 3: คำนวณ $g^{-3m}A \bmod p = g^{-m}g^{-2m}A \bmod p$, 13 x 11 mod 29 = 27

ครั้งที่ 4: คำนวณ $g^{-4m}A \bmod p = g^{-m}g^{-3m}A \bmod p$, 13 x 27 mod 29 = 3

ครั้งที่ 5: คำนวณ $g^{-5m}A \bmod p = g^{-m}g^{-4m}A \bmod p$, 13 x 3 mod 29 = 10

ครั้งที่ 6: คำนวณ $g^{-6m}A \bmod p = g^{-m}g^{-5m}A \bmod p$, 13 x 10 mod 29 = 14

จึงได้ตารางสำหรับ $g^{-sm}A \bmod p$ เมื่อ $0 \leq s < 6$ เป็นดังนี้

s	$g^{-sm}A \bmod p$
0	21
1	12
2	11
3	27
4	3
5	10
6	14

จากทั้งสองตารางข้างต้นพบว่าทั้งสองสมการมีค่าเท่ากันตรงตำแหน่งที่ t= 1 และ s = 2 ดังนั้น

$$g^1 = g^{-(2\times6)} A \bmod 29$$
$$= g^{-12} A \bmod 29$$
$$g^{12}g^1 = g^{12}g^{-12} A \bmod 29$$
$$g^{13} = A \bmod 29$$

ดังนั้น a = 13

อย่างไรก็ตามขนาด p ที่ใช้งานจริงมีค่าไม่น้อยกว่า 128 บิตส่งผลให้ตารางการคำนวณ $g^t \bmod p$ มีจำนวนแถวอยู่ที่ประมาณ $\sqrt{2^{128}} = 2^{64}$ ซึ่งเป็นขนาดที่ใหญ่มหาศาล ดังนั้นขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ จึงยังไม่มีประสิทธิภาพเพียงพอที่จะนำมาใช้สำหรับโจมตีวิทยาการรหัสลับเอ็ลกามอลและขั้นตอนวิธีดิฟฟี่เฮลแมนได้ภายในระยะเวลาสั้น

3.3 ขั้นตอนวิธีโพลิกเฮลแมน (Pohlig-Hellman Algorithm)

ขั้นตอนวิธีโพลิกเฮลแมนเป็นอีกขั้นตอนวิธีหนึ่งที่ถูกนำมาใช้แก้ปัญหาวิยุตลอการิทึมที่ถูกค้นพบโดยโรแลนด์ ซิลเวอร์ (Roland Silver) และถูกเผยแพร่ครั้งแรกโดยสตีเฟรน โพลิก (Stephen Pohlig) และมาติน เฮลแมน จึงเรียกขั้นตอนวิธีนี้ว่า ขั้นตอนวิธีโพลิกเฮลแมน [58]

กำหนดให้

$$
\begin{aligned}
p - 1 &= p_1^{d_1} p_2^{d_2} \cdots p_j^{d_j} \\
a_1 &= c_{0\_p1} + c_{1\_p1} p_1 + c_{2\_p1} p_1^2 + \cdots + c_{(d1-1)\_p1} p_1^{d_1 - 1} \\
a_2 &= c_{0\_p2} + c_{1\_p2} p_2 + c_{2\_p2} p_2^2 + \cdots + c_{(d2-1)\_p2} p_2^{d_2 - 1} \\
&\vdots \\
a_j &= c_{0\_pj} + c_{1\_pj} p_j + c_{2\_pj} p_j^2 + \cdots + c_{(dj-1)\_pj} p_j^{d_j - 1}
\end{aligned}
\tag{6.4}
$$

และ $A^{\frac{p-1}{p_s^{d}}} \equiv (g^{\frac{p-1}{p_s^{d}}})^{a_s} \bmod p$ (6.5)

เมื่อ $c_{i\_pi} \in \{0, 1, 2, \cdots, p_i - 1\}$ และ $p_s \in \{p_1, p_2, \cdots, p_j\}$

จากขั้นตอนวิธีโพลิกเฮลแมน หากทราบค่า $a_1, a_2, a_3, \cdots, a_j$ แล้วสามารถคำนวณหา a โดยใช้ทฤษฎีเศษเหลือจีนดังนี้

$$
\begin{aligned}
a_1 &\equiv a \bmod p_1^{d_1} \\
a_2 &\equiv a \bmod p_2^{d_2} \\
&\vdots \\
a_j &\equiv a \bmod p_j^{d_j}
\end{aligned}
$$

ตัวอย่างที่ 6.6 จาก A, p และ g ที่กำหนดให้ดังตัวอย่างที่ 6.5 จงคำนวณหา a โดยใช้ขั้นตอนวิธีโพลิกเฮลแมน

วิธีทำ เนื่องจาก $p - 1 = 2^2 \times 7$ ดังนั้นจากสมการ (6.4) ได้ว่า

$$a_1 = c_{0\_2} + c_{1\_2}p_1, \text{ เมื่อ } p_1 = 2$$

$$a_2 = c_{0\_7}$$

โดยที่การคำนวณหา a_1 และ a_2 สามารถแยกพิจารณาได้ดังนี้

การคำนวณหา a_1

จากสมการที่ (6.5), $$A^{\frac{p-1}{p_1^2}} \equiv (g^{\frac{p-1}{p_1^2}})^{a_1} \text{ mop } p$$

$$\equiv (g^{\frac{p-1}{p_1^2}})^{(c_{0\_2}+c_{1\_2}p_1)} \text{ mop } p$$

ยกกำลังด้วย p_1 ทั้ง 2 ข้างได้ว่า

$$(A^{\frac{p-1}{p_1^2}})^{p_1} \equiv (g^{\frac{p-1}{p_1^2}})^{p_1(c_{0\_2}+c_{1\_2}p_1)} \text{ mop } p$$

$$\equiv (g^{\frac{p-1}{p_1^2}})^{p_1(c_{0\_2})} (g^{\frac{p-1}{p_1^2}})^{c_{1\_2}p_1^2} \text{ mop } p$$

$$\equiv (g^{\frac{p-1}{p_1^2}})^{p_1(c_{0\_2})} (g^{p-1})^{c_{1\_2}} \text{ mop } p$$

จากทฤษฎีบทที่ 5.2 ได้ว่า $g^{p-1} \bmod p = 1$ ดังนั้น

$$(A^{\frac{p-1}{p_1^2}})^{p_1} \equiv (g^{\frac{p-1}{p_1^2}})^{p_1(c_{0\_2})} \text{ mop } p$$

พิจารณาสมการทางซ้ายมือ

$$(A^{p_1^{\frac{p-1}{2}}})^{p_1} \bmod p = (21^{2^{\frac{29-1}{2}}})^2 \bmod 29$$

$$= (21^{4^{\frac{28}{}}})^2 \bmod 29$$

$$= (21^7)^2 \bmod 29$$

$$= 21^{14} \bmod 29$$

$$= 28 \bmod 29$$

พิจารณาสมการทางขวามือ

$$(g^{p_1^{\frac{p-1}{2}}})^{p_1(c_{0\_2})} \bmod p = (11^{2^{\frac{29-1}{2}}})^{2(c_{0\_2})} \bmod 29$$

$$= (11^{4^{\frac{28}{}}})^{2(c_{0\_2})} \bmod 29$$

$$= (11^{2^{\frac{28}{}}})^{(c_{0\_2})} \bmod 29$$

$$= (11^{14})^{(c_{0\_2})} \bmod 29$$

$$= (28)^{(c_{0\_2})} \bmod 29$$

เพราะฉะนั้นจากสมการทั้งสองข้างได้ว่า $c_{0\_2} = 1$

แทนค่า c_{0_2} ลงในสมการตั้งต้นเพื่อหาค่า c_{1_2} ได้ว่า

$$(A^{p_1^{\frac{p-1}{2}}}) \equiv (g^{p_1^{\frac{p-1}{2}}})^{(c_{0\_2}+c_{1\_2}p_1)} \bmod p$$

$$21^{2^{\frac{29-1}{2}}} \equiv 11^{2^{\frac{29-1}{2}}(1+2c_{1\_2})} \bmod 29$$

$$21^7 \equiv 11^7 \times 11^{7(2c_{1\_2})} \bmod 29$$

$$12 \equiv 12 \times 11^{7(2c_{1\_2})} \bmod 29$$

$$1 \equiv 11^{7(2c_{1\_2})} \bmod 29$$

เพราะฉะนั้นจากสมการทั้งสองข้างได้ว่า $c_{1\_2} = 0$

แทนค่า c_{0_2} และ c_{1_2} เพื่อหา a_1,

$$\begin{aligned} a_1 &= c_{0\_2} + c_{1\_2}p_1 \\ &= 1 + 0x2 \\ &= 1 \end{aligned}$$

การคำนวณหา a_2

จากสมการที่ (6.5), $A^{\frac{p-1}{p_2}} \equiv (g^{\frac{p-1}{p_2}})^{a_2} \bmod p$

$$\equiv (g^{\frac{p-1}{p_2}})^{c_{0\_7}} \bmod p$$

พิจารณาสมการทางซ้ายมือ

$$\begin{aligned} A^{\frac{p-1}{p_2}} \bmod p &= 21^{\frac{29-1}{7}} \bmod 29 \\ &= 21^4 \bmod 29 \\ &= 7 \end{aligned}$$

พิจารณาสมการทางขวามือ

$$\begin{aligned} (g^{\frac{p-1}{p_2}})^{c_{0\_7}} \bmod p &= (11^{\frac{29-1}{7}})^{c_{0\_7}} \bmod 29 \\ &= (11^4)^{c_{0\_7}} \bmod 29 \\ &= 25^{c_{0\_7}} \end{aligned}$$

เนื่องจาก $25^6 \bmod 29 = 7$ ดังนั้น $c_{0\_7} = 6$

ได้ว่า $a_2 = 6$

หลังจากทราบค่า a_1 และ a_2 จึงสามารถคำนวณหา a โดยใช้ทฤษฎีเศษเหลือจีนได้จาก

$$a \equiv 1 \bmod 4$$
$$a \equiv 6 \bmod 7$$

เมื่อ $m = 28$, $m_1 = 4$, $m_2 = 7$, $M_1 = 7$ และ $M_2 = 4$
และใช้ขั้นตอนวิธียุคลิดภาคขยายเพื่อหาค่า y_1 และ y_2 ได้ว่า

$y_1 = 3$ เนื่องจาก $3 \times 7 = 21 = 1 \bmod 4$

$y_2 = 2$ เนื่องจาก $2 \times 4 = 8 = 1 \bmod 7$

ดังนั้น $a = (a_1y_1M_1 + a_2y_2M_2) \bmod m$
$= ((1)(3)(7) + (6)(2)(4)) \bmod 28$
$= 69 \bmod 28$
$= 13$

กำหนดให้ p_t แทนจำนวนเฉพาะที่มีค่าสูงสุดที่เป็นตัวประกอบค่าหนึ่งของ $p - 1$ ความปลอดภัยของขั้นตอนวิธีโพลิกเฮลแมนจะขึ้นอยู่กับ $\sqrt{p_t}$ ดังนั้นเพื่อให้สามารถหลีกเลี่ยงการโจมตีด้วยขั้นตอนวิธีโพลิกเฮลแมนจำเป็นต้องเลือก p_t ให้มีขนาด 256 บิตเป็นอย่างน้อย

3.4 ขั้นตอนวิธีดรรชนีแคลคูลัส (Index Calculus Algorithm)

ขั้นตอนวิธีดรรชนีแคลคูลัส [48] เป็นอีกขั้นตอนวิธีที่มีประสิทธิภาพสูงสำหรับแก้ปัญหาวิยุตลอการิทึม

กำหนดให้ $S(B)$ คือฐานตัวประกอบโดยที่ $S(B) = \{p_1, p_2, p_3, \cdots, p_j\}$ เมื่อ $p_1, p_2, p_3, \cdots, p_j < B$ แล้วเรียกจำนวนเต็ม n ว่า B-ปรับเรียบก็ต่อเมื่อตัวประกอบเฉพาะทุกตัวของ n เป็นสมาชิกอยู่ใน $S(B)$

ตัวอย่างที่ 6.7 กำหนดให้ $B = 8$ และ $S(8) = \{2, 3, 5, 7\}$ จงแสดงว่า 216 เป็น 8-ปรับเรียบใช่หรือไม่

วิธีทำ เนื่องจาก $216 = 2 \times 2 \times 2 \times 3 \times 3 \times 3$ หรือ $2^3 \times 2^3$
ซึ่งสังเกตุได้ว่าตัวประกอบทั้งหมดของ 216 เป็นสมาชิกใน $S(8)$
ดังนั้นสรุปได้ว่า 216 เป็น 8-ปรับเรียบ

ตัวอย่างที่ 6.8 จาก B และ $S(B)$ ดังตัวอย่างที่ 6.7 แสดงว่า 198 เป็น 8-ปรับเรียบใช่หรือไม่

วิธีทำ เนื่องจาก $198 = 2 \times 3 \times 3 \times 11$ หรือ $2 \times 3^2 \times 11$

ซึ่งสังเกตุได้ว่า 11 เป็นตัวประกอบของ 198 แต่ไม่เป็นสมาชิกใน $S(8)$

ดังนั้นสรุปได้ว่า 198 ไม่เป็น 8-ปรับเรียบ

สำหรับหลักการของการนำขั้นตอนวิธีดรรชนีแคลคูลัสมาประยุกต์เพื่อคำนวณหา a เริ่มจากแก้สมการเพื่อคำนวณหา $l_1, l_2, l_3, \cdots, l_j$ เมื่อ $l, j \in \mathbb{Z}$ ดังนี้

$$\begin{aligned} p_1 &= g^{l_1} \bmod p \\ p_2 &= g^{l_2} \bmod p \\ &\vdots \\ p_j &= g^{l_j} \bmod p \end{aligned} \tag{6.6}$$

จากสมการข้างต้นสามารถแก้ปัญหาได้โดยสุ่มเลือกค่า k โดยที่ $1 \leq k \leq p-2$ เพื่อคำนวณหาค่า $g^k \bmod p$ ได้ดังนี้

$$g^k \equiv x_k \bmod p \tag{6.7}$$

อย่างไรก็ตาม $x_k \bmod p$ สามารถเขียนให้อยู่ให้รูปของ $(g^{l_1})^{m_1} (g^{l_2})^{m_2} (g^{l_3})^{m_3} \cdots (g^{l_j})^{m_j} \bmod p$ ดังนั้น

$$g^k \bmod p = (g^{l_1})^{m_1} (g^{l_2})^{m_2} (g^{l_3})^{m_3} \cdots (g^{l_j})^{m_j} \bmod p \tag{6.8}$$

ดังนั้น

$$k = m_1 l_1 + m_2 l_2 + m_3 l_3 + \cdots + m_j l_j \bmod p - 1 \tag{6.9}$$

โดยที่ทุกตัวต้องเป็นสมาชิกใน $S(B)$ หากบางค่าไม่เป็นสมาชิกของ $S(B)$ จะไม่สามารถใช้ค่า k ที่เลือกได้ และจำเป็นต้องเลือกค่าใหม่

นอกเหนือจากนั้น จากสมการ (6.9) พบว่าหากเลือกค่า k เพียงค่าเดียวจะไม่สามารถคำนวณหา l_i ได้ทุกค่า เมื่อ $1 \leq i \leq j$ ดังนั้นจึงจำเป็นต้องสุ่มหา k จำนวนหลายค่าที่ตรงตามเงื่อนไขที่กำหนดข้างต้นเพื่อให้สามารถคำนวนหา l_i ทุกค่าได้

หลังจากทราบ l_i ครบทุกค่าแล้ว ขั้นตอนสุดท้ายคือการสุ่มหาค่า k อีก 1 ค่า และคำนวณ $Ag^k \bmod p$ ดังนี้

$$Ag^k \equiv y_k \bmod p \tag{6.10}$$

อย่างไรก็ตาม $y_k \bmod p$ สามารถเขียนให้อยู่ให้รูปของ $(g^{l_1})^{n_1} (g^{l_2})^{n_2} (g^{l_3})^{n_3} \cdots (g^{l_j})^{n_j} \bmod p$ ดังนั้น

$$\begin{aligned} Ag^k \bmod p &= (g^{l_1})^{n_1} (g^{l_2})^{n_2} (g^{l_3})^{n_3} \cdots (g^{l_j})^{n_j} \bmod p \\ g^a g^k \bmod p &= (g^{l_1})^{n_1} (g^{l_2})^{n_2} (g^{l_3})^{n_3} \cdots (g^{l_j})^{n_j} \bmod p \\ g^{a+k} \bmod p &= (g^{l_1})^{n_1} (g^{l_2})^{n_2} (g^{l_3})^{n_3} \cdots (g^{l_j})^{n_j} \bmod p \end{aligned} \tag{6.11}$$

ดังนั้น

$$a + k = n_1 l_1 + n_2 l_2 + n_3 l_3 + \cdots + n_j l_j \bmod p - 1 \tag{6.12}$$

ตัวอย่างที่ 6.9 กำหนดให้ $B = 8$, $S(B) = \{2, 3, 5, 7\}$, $A = 43$, $p = 113$ และ $g = 3$ จงคำนวณหา a โดยใช้ขั้นตอนวิธีดรรชนีแคลคูลัส

วิธีทำ จากสมการ (6.6) ได้ว่า

$$\begin{aligned} 2 &= 3^{l_1} \bmod 113 \\ 3 &= 3^{l_2} \bmod 113 \\ 5 &= 3^{l_3} \bmod 113 \\ 7 &= 3^{l_4} \bmod 113 \end{aligned}$$

ขั้นตอนถัดไปคือการเลือก k เพื่อใช้สำหรับแก้สมการหา l_1, l_2, l_3 และ l_4 โดยพิจารณาสมการ (6.7)

1) เลือก $k = 1$,

$$3^1 \bmod 113 = 3$$

เนื่องจาก 3 เป็นสมาชิกใน $S(8)$ กล่าวได้ว่า $3^{l_2} = 3$ ดังนั้นจากสมการ (6.9) ได้ว่า

$$1 = 0 \times l_1 + 1 \times l_2 + 0 \times l_3 + 0 \times l_4 \bmod 112$$

หรือ

$$l_2 = 1 \bmod 112 \qquad (1)$$

2) เลือก $k = 5$,

$$3^5 \bmod 313 = 17 \bmod 113$$

เนื่องจาก 17 ไม่เป็นสมาชิกใน $S(8)$ ดังนั้นไม่สามารถเลือก $k = 5$ มาใช้ได้

3) เลือก $k = 7$,

$$3^7 \bmod 313 = 40 \bmod 113$$
$$= 2^3 \times 5 \bmod 113$$

เนื่องจาก 2 และ 5 เป็นสมาชิกใน $S(8)$ กล่าวได้ว่า $3^{l_1} = 2$ และ $3^{l_3} = 5$ ดังนั้นจากสมการ (6.9) ได้ว่า

$$7 = 3 \times l_1 + 0 \times l_2 + 1 \times l_3 + 0 \times l_4 \bmod 112$$

หรือ

$$3l_1 + l_3 = 7 \bmod 112 \qquad (2)$$

4) เลือก $k = 8$,

$$3^8 \bmod 113 = 7 \bmod 113$$

เนื่องจาก 7 เป็นสมาชิกใน $S(8)$ กล่าวได้ว่าและ $3^{l_4} = 7$ ดังนั้นจากสมการ (6.9) ได้ว่า

$$8 = 0 \times l_1 + 0 \times l_2 + 0 \times l_3 + 1 \times l_4 \bmod 112$$

$$l_4 = 8 \bmod 112 \qquad (3)$$

5) เลือก $k = 12$,

$$3^{12} \bmod 113 = 2 \bmod 113$$

เนื่องจาก 2 เป็นสมาชิกใน $S(8)$ กล่าวได้ว่า $3^{l_1} = 2$ ดังนั้นจากสมการ (6.9) ได้ว่า

$$12 = 1 \times l_1 + 0 \times l_2 + 0 \times l_3 + 0 \times l_4 \bmod 112$$

หรือ $l_1 = 12 \bmod 112$ (4)

แทน l_1 ใน (2) $3l_1 + l_3 = 7 \bmod 112$

$3 \times 12 + l_3 = 7 \bmod 112$

$l_3 = 83 \bmod 112$

ดังนั้นจาก (1), (2), (3), (4) ได้

$l_1 = 12 \bmod 113$

$l_2 = 1 \bmod 113$

$l_3 = 83 \bmod 113$

$l_4 = 8 \bmod 113$

หลังจากทราบค่า l_1, l_2, l_3 และ l_4 แล้วขั้นตอนสุดท้ายคือการเลือกค่า k อีกหนึ่งค่าเพื่อใช้สำหรับคำนวณหาค่า a ดังนี้

เลือก $k = 19$,

จาก $Ag^k \bmod p = y_k \bmod p$

$43 \times 3^{19} \bmod 113 = 43 \times 80 \bmod 113$

$= 3440 \bmod 113$

$= 50 \bmod 113$

$= 2 \times 5^2 \bmod 113$

เนื่องจาก 2 และ 5 เป็นสมาชิกใน S(8) กล่าวได้ว่า $7^{l_1} = 2$ และ $7^{l_3} = 3$ ดังนั้นจาก

$a + k = n_1l_1 + n_2l_2 + n_3l_3 + n_4l_4 \bmod p - 1$

$a + 19 = 1 \times 12 + 0 \times 1 + 2 \times 83 + 0 \times 1 \bmod 112$

$a + 19 = 178 \bmod 112$

$a + 19 = 66 \bmod 112$

$a = 47 \bmod 112$

ขั้นตอนวิธีดรรชนีแคลคูลัสได้รับการยอมรับว่าเป็นขั้นตอนวิธีที่มีประสิทธิภาพสูงที่สุดสำหรับแก้ปัญหาวิยุตลอการิทึม ดังนั้นเพื่อหลีกเลี่ยงการโจมตีด้วยขั้นตอนวิธีดังกล่าวนี้จำเป็นต้องเลือกใช้งาน B, g และ p ที่มีขนาด 1024 บิตเป็นอย่างน้อย และ p ต้องเป็นจำนวนเฉพาะที่แข็งแกร่งซึ่งยากแก่การโจมตี

3.5 การโจมตีวิทยาการรหัสลับเอ็ลกามอลโดยพิจารณาคู่ความสัมพันธ์ระหว่างข้อความต้นฉบับและข้อความไซเฟอร์

การโจมตีรหัสลับเอ็ลกามอลอีกวิธีหนึ่งคือ สมมติผู้ไม่ประสงค์ดีทราบข้อความต้นฉบับ m_1 และ c_1 ซึ่งเป็นข้อความไซเฟอร์ของ m_1 และทราบข้อความไซเฟอร์ c_2 ซึ่งเป็นข้อความไซเฟอร์ของ m_2 โดยที่ทั้ง c_1 และ c_2 เกิดจากการใช้กุญแจส่วนตัว และกุญแจสาธารณะชุดเดียวกันทั้งหมด ผู้ไม่ประสงค์ดีสามารถคำนวณหา m_2 ได้จากความสัมพันธ์ดังนี้

จาก $c_1 = A^b m_1 \bmod p$

ดังนั้น $c_1^{-1} = (A^b m_1)^{-1} \bmod p$

$= A^{-b} m_1^{-1} \bmod p$

จาก $c_2 = A^b m_2 \bmod p$

$c_2 c_1^{-1} = A^{-b} m_1^{-1} A^b m_2 \bmod p$

$= m_1^{-1} m_2 \bmod p$

ดังนั้น

$$m_2 = m_1 c_2 c_1^{-1} \bmod p \qquad (6.13)$$

ตัวอย่างที่ 6.10 กำหนดให้ $p = 37$, $g = 13$, $A = 32$ และผู้ไม่ประสงค์ดีทราบคู่ความสัมพันธ์ระหว่างข้อความต้นฉบับและข้อความไซเฟอร์คือ 13 และ 30 ตามลำดับ และทราบข้อความไซเฟอร์อีกค่าหนึ่งที่มีค่าเป็น 9 จงคำนวณหาข้อความต้นฉบับของข้อความไซเฟอร์ดังกล่าว

วิธีทำ จากโจทย์ได้ว่า $m_1 = 13$, $c_1 = 30$ และ $c_2 = 9$

จากสมการ (6.13)

$$m_2 = m_1 c_2 c_1^{-1} \bmod p$$

และเนื่องจาก $30 \times 21 \bmod 37 = 1$ ดังนั้น $c^{-1} = 21 \bmod 37$

ดังนั้น $m_2 = 13 \times 9 \times 21 \bmod 37$

$= 15$

ดังนั้นเพื่อหลีกเลี่ยงการโจมตีโดยพิจารณาคู่ความสัมพันธ์ระหว่างข้อความต้นฉบับและข้อความไซเฟอร์ จะต้องป้องกันไม่ให้ผู้ไม่ประสงค์ดีทราบข้อความต้นฉบับทั้งหมด

4. บทสรุปสาระสำคัญ

วิทยาการรหัสลับแบบกุญแจสาธารณะคือจุดเปลี่ยนแปลงที่สำคัญของศาสตร์ด้านวิทยาการรหัสลับเนื่องมาจากขั้นตอนวิธีทั้งหมดในกลุ่มนี้ใช้กุญแจคู่สำหรับกระบวนการเข้ารหัสลับและถอดรหัสลับที่ประกอบด้วยกุญแจสาธารณะและกุญแจส่วนตัว โดยที่กุญแจสาธารณะจะถูกประกาศอย่างเปิดเผยในทางกลับกันกุญแจส่วนตัวจะถูกเก็บไว้เป็นความลับ ขั้นตอนวิธีดิฟฟี่เฮลแมนคือขั้นตอนวิธีแรกที่ถูกนำเสนอที่สามารถนำมาใช้ได้กับกระบวนการแลกเปลี่ยนกุญแจลับเพียงเท่านั้น ในเวลาต่อมาจึงได้มีการเสนอขั้นตอนวิธีอื่นที่สามารถนำมาประยุกต์ใช้แก้ปัญหาที่สำคัญนอกเหนือจากการแลกเปลี่ยนกุญแจลับคือ กระบวนการเข้ารหัสข้อมูล และลายเซ็นดิจิทัล โดยในบทนี้ได้กล่าวถึงวิทยาการรหัสลับเอ็ลกามอลซึ่งเป็นขั้นตอนวิธีที่ปรับปรุงมาจากขั้นตอนวิธีดิฟฟี่เฮลแมนเพื่อให้สามารถนำมาใช้สำหรับการเข้ารหัสลับข้อมูล และลายเซ็นดิจิทัลได้ อย่างไรก็ตามความปลอดภัยของทั้งสองวิธีจะขึ้นอยู่กับปัญหาวิยุตลอการิทึมคือความยากของการคำนวณหาเลขยกกำลังมอดูลาร์จากเลขฐานและคำตอบที่มีขนาดใหญ่มหาศาล ถึงแม้ว่าจะมีขั้นตอนวิธีที่ถูกนำเสนอขึ้นมาเพื่อแก้ปัญหาวิยุตลอการิทึมออกมาเป็นจำนวนมาก แต่หากการเลือกใช้กุญแจที่มีขนาดใหญ่มหาศาลและมีความแข็งแกร่ง การคำนวณหาผลลัพธ์ของเลขยกกำลังจากปัญหาดังกล่าวภายในระยะเวลาสั้นเป็นไปได้ยากมาก

แบบฝึกหัดท้ายบท
บทที่ 6

1. วิทยาการรหัสลับแบบอสมมาตรนิยมถูกเรียกอีกชื่อว่าอะไร
2. จำนวนกุญแจที่จำเป็นต้องใช้สำหรับวิทยาการรหัสลับแบบอสมมาตรมีทั้งหมดเท่าไร
3. กุญแจสาธารณะคืออะไร
4. ขั้นตอนวิธีดิฟฟี่เฮลแมนนิยมนำมาใช้สำหรับดำเนินการอะไร
5. หากผู้ไม่ประสงค์ดีประสงค์จะโจมตีขั้นตอนวิธีดิฟฟี่เฮลแมน หรือวิทยาการรหัสลับเอ็ลกามอล จะต้องแก้ปัญหาที่เรียกว่าอะไร
6. กำหนดให้นาย ก และ นาย ข ตกลงใช้ค่ามอดูลัส และรากปฐมฐานเป็น 13 และ 7 ตามลำดับ โดย นาย ก เลือกกุญแจส่วนตัวเป็น 4 และ นาย ข เลือกกุญแจส่วนตัวเป็น 8 จงคำนวณหากุญแจลับระหว่างนาย ก และนาย ข โดยใช้ขั้นตอนวิธีดิฟฟี่เฮลแมน
7. กำหนดให้กุญแจสาธารณะ และค่ามอดูลัสมีค่าเป็นดังนี้ $\{g = 13, A = 615, p = 937\}$ และ $b = 6$ จงเข้ารหัสลับ $m = 37$ โดยใช้วิทยาการรหัสลับเอ็ลกามอล
8. จงถอดรหัสข้อความไซเฟอร์ที่ได้จากคำถามข้อ 7 กำหนดให้ $a = 51$
9. จงคำนวณหา a จากสมการ $112 = 23^a \bmod 131$ โดยใช้วิธีโจมตีแบบตะลุย
10. จากคำถามข้อ 9 จงคำนวณหา a โดยใช้วิธีเบบี้สเต็พไจแอ็นสเต็พ
11. จากคำถามข้อ 9 จงคำนวณหา a โดยใช้วิธีโพลิกเฮลแมน
12. กำหนดให้ $B = 22$ จงหา $S(B)$
13. กำหนดให้ $B = 12$ และ $S(12) = \{2, 3, 5, 7\}$ จงแสดงว่า 84 เป็น 12-ปรับเรียบใช่หรือไม่
14. จงแสดงว่า 85 เป็น 12-ปรับเรียบใช่หรือไม่
15. การคำนวณหาเลขยกกำลังโดยใช้วิธีเบบี้สเต็พไจแอ็นสเต็พไม่จำเป็นต้องคำนวณหาผลลัพธ์ทั้งหมดในตาราง แต่สามารถค้นหาข้อมูลเพียงบางส่วนแล้วพบคำตอบโดยใช้หลักการที่เรียกว่าอะไร
16. กำหนดให้ $p = 53$ และ $g = 11$ สมมติผู้ไม่ประสงค์ดีทราบคู่ความสัมพันธ์ระหว่างข้อความต้นฉบับและข้อความไซเฟอร์คือ 17 และ 28 ตามลำดับ และทราบข้อความไซเฟอร์อีกค่าหนึ่งที่มีค่าเป็น 22 จงคำนวณหาข้อความต้นฉบับของข้อความไซเฟอร์ดังกล่าว

บทที่ 7
วิทยาการรหัสลับอาร์เอสเอ

วิทยาการรหัสลับอาร์เอเอส (RSA Cryptography) [8] เป็นวิทยาการรหัสลับแบบกุญแจสาธารณะที่ได้รับความนิยมสูงมากในปัจจุบันที่ถูกนำเสนอโดย รอน ริเวสต์ (Ron Rivest) อาดี ชามีร์ (Adi Shamir) และ เล็น เอเดิลแมน (Len Adleman) ในช่วง ค.ศ. 1978 ซึ่งเกิดก่อนวิทยาการรหัสลับเอ็ลกามอล โดยคำว่า “อาร์เอสเอ” มาจากตัวอักษรตัวแรกของนามสกุลของทั้ง 3 ท่าน วิทยาการรหัสลับอาร์เอสเอเกิดหลังจากที่ขั้นตอนวิธีดิฟฟี่เฮลแมนถูกตีพิมพ์เผยแพร่ประมาณ 2 ปี แต่เนื่องจากขั้นตอนวิธีดิฟฟี่เฮลแมนถูกนำมาใช้สำหรับกระบวนการแลกเปลี่ยนกุญแจลับเพียงเท่านั้น จึงกล่าวได้ว่าวิทยาการรหัสลับอาร์เอสเอเป็นวิทยาการรหัสลับแบบกุญแจสาธารณะขั้นตอนวิธีแรกที่สามารถนำมาใช้สำหรับการรักษาความปลอดภัยข้อมูลข่าวสารผ่านกระบวนการเข้ารหัสลับและถอดรหัสลับ

วิทยาการรหัสลับอาร์เอสเอเป็นขั้นตอนวิธีที่มีความปลอดภัยสูง เมื่อถูกนำมาใช้ร่วมกับค่ามอดูลัสที่มีขนาดอย่างน้อย 1024 บิต [42] เนื่องจากมีหัวใจที่สำคัญคือการคำนวณหาจำนวนเฉพาะขนาดใหญ่จำนวน 2 ค่าสามารถดำเนินการได้อย่างรวดเร็ว โดยที่จำนวนเฉพาะทั้ง 2 ค่านี้ถูกนำมาใช้สำหรับกระบวนการก่อกำเนิดกุญแจ ในทางกลับกันการแยกตัวประกอบค่ามอดูลัสเพื่อให้ได้กลับมาซึ่งจำนวนเฉพาะทั้ง 2 ค่าสามารถดำเนินการได้ยากมากเนื่องจากต้องใช้เวลาสำหรับการประมวลผลมหาศาล สำหรับเนื้อหาในบทนี้จะแนะนำวิทยาการรหัสลับอาร์เอสเอ ความปลอดภัย และเทคนิคต่างๆ ที่เป็นประโยชน์สำหรับวิทยาการรหัสลับอาร์เอสเอ

1. วิทยาการรหัสลับอาร์เอสเอ (RSA Cryptography)

วิทยาการรหัสลับอาร์เอสเอ ถูกแบ่งออกเป็น 3 กระบวนการ ดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจ หรือผู้รอรับข้อความไซเฟอร์ (กำหนดเป็น ผู้รับ) มีลำดับการทำงานเป็นดังนี้

4. เลือกจำนวนเฉพาะจำนวน 2 ค่า กำหนดเป็น p และ q
5. คำนวณค่ามอดูลัส (n) จาก $n = pq$
6. คำนวณค่าออยเลอร์ ($\Phi(n)$) จาก $\Phi(n) = (p - 1)(q - 1)$
7. เลือกค่ากุญแจสาธารณะ (e) ที่มีเงื่อนไขว่า $1 < e < \Phi(n)$ และ $\gcd(e, \Phi(n)) = 1$

8. คำนวณหาค่ากุญแจส่วนตัว (d) จาก $d = e^{-1} \bmod \Phi(n)$

กุญแจสาธารณะคือ {e, n}

กุญแจส่วนตัวคือ {d}

กระบวนการที่ 2 การเข้ารหัสลับ: กำหนดให้ข้อความต้นฉบับคือ m โดยที่ $1 < m < n$ ผู้ส่งใช้ (e, n) ของผู้รับสำหรับเข้ารหัส m ด้วยสมการเข้ารหัสด้วยอาร์เอสเอ ดังนี้

$$c = m^e \bmod n \tag{7.1}$$

เมื่อ c คือข้อความไซเฟอร์

หลังเสร็จสิ้นกระบวนการเข้ารหัสลับแล้ว ผู้ส่งจะส่ง c ไปยังผู้รับ

กระบวนการที่ 3 การถอดรหัสลับ: หลังจากที่ผู้รับได้รับ c จากผู้ส่งจะสามารถคำนวณหา m ได้โดยสมการต่อไปนี้

$$m = c^d \bmod n \tag{7.2}$$

หลังจากเสร็จสิ้นกระบวนการถอดรหัสลับโดยใช้วิทยาการรหัสลับแล้ว จะได้ค่า m กลับคืนเสมอ เนื่องจาก

$$\begin{aligned} c^d \bmod n &= (m^e)^d \bmod n \\ &= m^{ed} \bmod n \\ &= m^{1+k\Phi(n)} \bmod n \\ &= m\, m^{k\Phi(n)} \bmod n \\ &= m\,(m^{\Phi(n)})^k \bmod n \end{aligned}$$

จากทฤษฎีของออยเลอร์ หาก gcd(m, n) = 1 ได้ว่า

$$\begin{aligned} &= m\,(1^k) \bmod n \\ &= m \end{aligned}$$

ตัวอย่างที่ 7.1 การประยุกต์ใช้วิทยาการรหัสลับอาร์เอสเอ

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

4. เลือกจำนวนเฉพาะ $p = 67$ และ $q = 79$
5. คำนวณ $n = 67 \times 79 = 5293$
6. คำนวณ $\Phi(n) = 66 \times 78 = 5148$
7. เลือก $e = 919$ เนื่องจาก $1 < 919 < 5148$ และ gcd(919, 5148) = 1
8. คำนวณ $d = 919^{-1} \bmod 5148 = 1927$

กุญแจสาธารณะคือ {$e = 919$, $n = 5293$}

กุญแจส่วนตัวคือ {$d = 1927$}

กระบวนการเข้ารหัส

สมมติผู้ส่งต้องการเข้ารหัส $m = 141$

คำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$c = 141^{919} \bmod 5293$$
$$= 891$$

ส่ง $c = 891$ ไปยังผู้รับ

กระบวนการถอดรหัส

เมื่อรับข้อความไซเฟอร์ $c = 891$ จากผู้ส่ง ผู้รับสามารถคำนวณหา m โดยใช้สมการการถอดรหัสสำหรับวิทยาการรหัสลับอาร์เอสเอ โดยคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$m = 891^{1927} \bmod 5293$$
$$= 141$$

2. การแปลงค่าระหว่างตัวอักขระและตัวเลข

ข้อความต้นฉบับ และข้อความไซเฟอร์ที่ถูกนำมาใช้สำหรับวิทยาการรหัสลับอาร์เอสเอเป็นตัวเลขทั้งหมด แต่ในทางปฏิบัติแล้วข้อความลับส่วนใหญ่เป็นตัวอักขระ (ตัวอักษร) ดังนั้นหัวข้อนี้จะกล่าวถึงกระบวนการแปลงค่าระหว่างตัวอักขระแบบกลุ่ม (มีจำนวนอักขระมากกว่า 1 ตัว) และตัวเลข [43]

กำหนดให้ N คือจำนวนตัวอักขระทั้งหมด

2.1 การแปลงจากอักขระแบบกลุ่มเป็นตัวเลข

การแปลงจากอักขระแบบกลุ่มเป็นตัวเลขคือกระบวนการแปลงข้อความต้นฉบับที่เป็นกลุ่มตัวอักขระไปเป็นตัวเลขเพื่อนำไปเข้าสู่กระบวนการเข้ารหัส กำหนดให้ k คือจำนวนอักขระที่เป็นไปได้ทั้งหมดต่อกระบวนการเข้ารหัส 1 ครั้ง สามารถคำนวณหาได้ ดังนี้

$$k = \left\lfloor \log_N{}^n \right\rfloor \tag{7.3}$$

โดยการแปลงจากตัวอักขระแบบกลุ่มเป็นตัวเลขสามารถดำเนินการได้โดยใช้สมการ (7.4)

$$m = \sum_{i=0}^{k-1} m_i N^i \tag{7.4}$$

เมื่อ m คือผลลัพธ์ที่เป็นเลขฐานสิบ

m_i ค่าเลขฐาน N ในตำแหน่งที่ i

ตัวอย่างที่ 7.2 สมมติว่ามีอักขระ 5 ตัวและมีค่าประจำตัวดังตารางต่อไปนี้

ตารางที่ 7.1 การแทนค่าระหว่างอักขระและตัวเลขจำนวน 5 ตัว

A	B	C	D	E
0	1	2	3	4

จากตารางได้ $N = 5$ และจากสมการ (7.3) และค่ามอดูลัสจากตัวอย่างที่ 7.1 ได้ว่า

$$k = \left\lfloor \log_5{}^{5293} \right\rfloor = 5$$

ดังนั้นสรุปได้ว่า กรณีที่มีตัวอักขระทั้งหมด 5 ตัว หากใช้ชุดกุญแจคู่จากตัวอย่างที่ 7.1 จะสามารถเข้ารหัสได้ครั้งละไม่เกิน 5 อักขระ

หมายเหตุ: จำนวนตัวอักขระที่นำมาใช้สำหรับกระบวนการเข้ารหัส 1 ครั้งจะมีค่าลดลงเมื่อ N มีค่าที่สูงขึ้น

สมมติผู้ส่งต้องการส่งข้อความ “BED” ไปยังผู้รับ ก่อนเข้าสู่กระบวนการเข้ารหัสผู้ส่งจำเป็นต้องแปลงข้อความดังกล่าวเป็นตัวเลขเสียก่อน ดังนี้

จากตารางที่ 7.1 พิจารณาหาค่า m_i เมื่อ i มีค่าเป็น 0 ถึง 2 ได้ดังนี้

m_0 = ‘D’ = 3

m_1 = ‘E’ = 4

m_2 = ‘B’ = 1

และคำนวณหา m จากสมการที่ (7.4) เป็นดังนี้

จาก $m = m_2N^2 + m_1N^1 + m_0N^0$

$= 1x5^2 + 4x5^1 + 3x5^0$

$= 25 + 20 + 3$

$= 48$

เมื่อทราบค่า m แล้วจึงสามารถคำนวณ c โดยใช้สมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$c = 48^{919} \bmod 5293 = 4814$

2.2 การแปลงจากตัวเลขเป็นอักขระแบบกลุ่ม

การแปลงจากตัวเลขเป็นอักขระแบบกลุ่มจะอยู่ในขั้นตอนหลังจากเสร็จสิ้นกระบวนเข้ารหัส ซึ่งจะได้ผลลัพธ์เป็นตัวเลข อย่างไรก็ตามข้อความไซเฟอร์โดยส่วนมากจะถูกแปลงเป็นอักขระแบบกลุ่มก่อนที่จะถูกส่งไปยังผู้รับ

สำหรับกระบวนการแปลงจากตัวเลขเป็นอักขระแบบกลุ่มสามารถดำเนินการได้โดยการนำตัวเลขที่ต้องการแปลงมาหารด้วย N โดยเศษที่ได้จากการหารจะมีค่าอยู่ระหว่าง 0 ถึง N-1 ซึ่งให้ทำการแปลงเป็นอักขระที่ตรงกัน และนำผลลัพธ์ที่ได้จากการหารมาหารด้วย N และเก็บเศษในลักษณะเดียวกับการดำเนินการครั้งแรก โดยจะดำเนินการลักษณะนี้จนกระทั่งผลลัพธ์สุดท้ายมีค่าน้อยกว่า N คำตอบสุดท้ายคือนำผลลัพธ์สุดท้ายและเศษที่คำนวณได้ทั้งหมดมาเขียนเรียงต่อกันจากซ้ายไปขวา โดยผลลัพธ์สุดท้ายจะถูกจัดอยู่ตำแหน่งซ้ายสุด และตามด้วยเศษที่คำนวณได้ครั้งล่าสุดเรียงไปจนกระทั่งถึงเศษที่ถูกคำนวณได้ในครั้งแรกที่จะถูกจัดไว้ตำแหน่งขวาสุดซึ่งจากหลักการดังกล่าวจะเป็นลักษณะเช่นเดียวกันกับการแปลงเลขฐานจากเลขฐานสิบเป็นเลขฐานใดๆ

ตัวอย่างที่ 7.3 นำผลลัพธ์ที่ได้จากตัวอย่างที่ 7.2 มาแปลงเป็นอักขระแบบกลุ่ม

วิธีทำ ผลลัพธ์จากตัวอย่างที่ 7.2 คือ c = 4814

ได้ว่า

4814 ÷ 5 = 962 เศษ 4 (4 = ‘E’)

962 ÷ 5 = 192 เศษ 2 (2 = ‘C’)

192 ÷ 5 = 38 เศษ 2 (2 = ‘C’)

38 ÷ 5 = 7 เศษ 3 (3 = ‘D’)

7 ÷ 5 = 1 เศษ 2 (2 = ‘C’)

1 ÷ 5 = 0 เศษ 1 (1 = ‘B’)

ดังนั้น ข้อความไซเฟอร์ (อักขระ) คือ “BCDCCE”

เนื่องจากอักขระภาษาอังกฤษซึ่งมีจำนวน 26 ตัว ดังนั้นหากต้องการใช้อักขระภาษาอังกฤษทุกตัว และช่องว่างอีก 1 ช่องจำเป็นต้องใช้ค่า $N = 27$ ซึ่งหากนำมาประยุกต์ใช้กับค่า n จากตัวอย่างที่ 7.2 และ 7.3 ซึ่งมีค่าเพียง 5293 จะส่งผลให้ $k = \left\lfloor \log_{27}{}^{5293} \right\rfloor = 2$ ดังนั้นหากต้องการแทนอักขระแบบกลุ่มให้ได้จำนวนสูงขึ้นจะต้องใช้ค่า n ที่สูงมากยิ่งขึ้น อย่างไรก็ตามค่ามอดูลัสที่ถูกนำมาใช้จริงสำหรับขั้นตอนวิธีอาร์เอสเอมีขนาด 1024 บิตเป็นอย่างน้อย ซึ่งเป็นขนาดใหญ่มากพอสำหรับการแทนค่าอักขระแบบกลุ่มที่มีจำนวนอักขระจำนวนมากด้วยตัวเลข

ตารางที่ 7.2 การแทนค่าระหว่างอักขระและตัวเลขจำนวน 27 ตัว

ตำแหน่ง	ตำอักษร		ตำแหน่ง	ตำอักษร
0	A		14	O
1	B		15	P
2	C		16	Q
3	D		17	R
4	E		18	S
5	F		19	T
6	G		20	U
7	H		21	V
8	I		22	W
9	J		23	X
10	K		24	Y
11	L		25	Z
12	M		26	Space
13	N			

ตัวอย่างที่ 7.4 กำหนดให้ n = 17138213 (3373 x 5081), $\Phi(n)$ = 17129760 และ e = 3037 จงคำนวณหาข้อความไซเฟอร์เมื่อข้อความต้นฉบับคือ “SECRET KEY” โดยใช้ตารางที่ 7.2 สำหรับแปลงค่าระหว่างอักขระและตัวเลข

วิธีทำ จากสมการ (7.3)

$$k = \left\lfloor \log_{27}{17138213} \right\rfloor = 5$$

ดังนั้นสรุปได้ว่า กรณีที่มีตัวอักขระทั้งหมด 27 ตัว หากใช้ชุดกุญแจคู่จากตัวอย่างที่ 7.4 จะสามารถเข้ารหัสได้ครั้งละไม่เกิน 5 อักขระ

แต่เนื่องจากข้อความไซเฟอร์คือ “SECRET KEY” มีอักษระจำนวนทั้งหมด 10 ตัว ดังนั้นจึงจำเป็นต้องแบ่งออกเป็น 2 กลุ่ม สำหรับการเข้ารหัสแต่ละครั้ง

กำหนดให้

m_1 (อักขระ) = “SECRE”

m_2 (อักขระ) = “T KEY”

การเข้ารหัสจึงแบ่งออกเป็น 2 ครั้ง โดยกำหนดให้ c_1 คือข้อความไซเฟอร์ของ m_1 และ c_2 คือข้อความไซเฟอร์ของ m_2

การแปลงอักขระกลุ่ม m_1 เป็นตัวเลข

จากตารางที่ 7.2 พิจารณาหาค่า m_{1_i} เมื่อ $i \in \mathbb{Z}$ ที่มีค่าระหว่าง 0 ถึง 5 ได้ดังนี้

m_{1_0} = ‘E’ = 4

m_{1_1} = ‘R’ = 17

m_{1_2} = ‘C’ = 2

m_{1_3} = ‘E’ = 4

m_{1_4} = ‘S’ = 18

และคำนวณหา m_1 จากสมการที่ (7.4) เป็นดังนี้ (N = 27)

จาก $m_1 = m_{1\_4}N^4 + m_{1\_3}N^3 + m_{1\_2}N^2 + m_{1\_1}N^1 + m_{1\_0}N^0$

$= 18\text{x}27^4 + 4\text{x}27^3 + 2\text{x}27^2 + 17\text{x}27^1 + 4\text{x}27^0$

$= 9565938 + 78732 + 1458 + 459 + 4$

$= 9646591$

การแปลงอักขระกลุ่ม m_2 เป็นตัวเลข

พิจารณาหาค่า m_{2_i} ได้ดังนี้

m_{2_0} = ‘Y’ = 24

m_{2_1} = ‘E’ = 4

m_{2_2} = ‘K’ = 10

m_{2_3} = ช่องว่าง = 26

m_{2_4} = ‘T’ = 19

และคำนวณหา m_2 จากสมการที่ (7.2) เป็นดังนี้ (N = 27)

จาก $m_2 = m_{2\_4}\cdot N^4 + m_{2\_3}\cdot N^3 + m_{2\_2}\cdot N^2 + m_{2\_1}\cdot N^1 + m_{2\_0}\cdot N^0$

$= 19x27^4 + 26x27^3 + 10x27^2 + 4x27^1 + 24x27^0$

$= 10097379 + 511758 + 7290 + 24$

$= 10616451$

เมื่อทราบค่า m_1 และ m_2 แล้วจึงสามารถคำนวณ c_1 และ c_2 โดยใช้การคำนวณการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้ได้ดังนี้

$c_1 = 9646591^{3037} \bmod 17138213 = 1279414$

และ

$c_2 = 10616451^{3037} \bmod 17138213 = 9166346$

การแปลง c_1 เป็นอักขระแบบกลุ่ม

1279414 ÷ 27 = 47385 เศษ 19 (19 = ‘T’)

47385 ÷ 27 = 1755 เศษ 0 (0 = ‘A’)

1755 ÷ 27 = 65 เศษ 0 (0 = ‘A’)

65 ÷ 27 = 2 เศษ 11 (11 = ‘L’)

ผลลัพธ์สุดท้ายคือ 2 (2 = ‘C’)

ดังนั้น c_1 (อักขระ) คือ “CLAAT”

การแปลง c_2 เป็นอักขระแบบกลุ่ม

9166346 ÷ 27 = 339494 เศษ 8 (8 = ‘I’)

339494 ÷ 27 = 12573 เศษ 23 (23 = ‘X’)

12573 ÷ 27 = 465 เศษ 18 (18 = ‘S’)

465 ÷ 27 = 17 เศษ 6 (6 = ‘G’)

ผลลัพธ์สุดท้ายคือ 17 (17 = ‘R’)

ดังนั้น c_2 (อักขระ) คือ “RGSXI”

หลังจากนั้นผู้ส่งจะส่ง c_1 (อักขระ) และ c_2 (อักขระ) ไปยังผู้รับ และเมื่อผู้รับเสร็จสิ้นกระบวนการถอดรหัสผู้รับจะทราบค่า m_1 และ m_2 และสามารถหาข้อความต้นฉบับได้จากการนำ m_2 มาเรียงต่อจาก m_1 (m (อักขระ) = m_1 (อักขระ) + m_2 (อักขระ))

โดยการแปลงค่าระหว่างตัวอักขระและตัวเลขสามารถนำไปประยุกต์ใช้กับขั้นตอนวิธีอื่นได้ เช่นวิทยาการรหัสลับเอ็ลกามอล ซึ่งได้กล่าวไว้แล้วในบทที่ 6

3. การเพิ่มความเร็วกระบวนการถอดรหัสอาร์เอสเอ

โดยทั่วไปเพื่อเพิ่มความเร็วสำหรับกระบวนการเข้ารหัส ผู้ก่อกำเนิดกุญแจ (ผู้รับ) นิยมเลือกกำหนดค่า e ให้มีขนาดเล็ก ซึ่งสามารถช่วยลดเวลาการคำนวณสมการการยกกำลังมอดูลาร์ลงได้เป็นอย่างมาก อย่างไรก็ตามการเลือกค่า e ที่มีขนาดเล็กอาจส่งผลให้ค่า d มีค่าสูง ซึ่งการคำนวณสมการการยกกำลังมอดูลาร์ใช้เวลาประมวลผลที่สูงมาก ถึงแม้จะเลือกใช้ขั้นตอนวิธีที่มีประสิทธิภาพ เช่น เลขยกกำลังแบบเร็ว หรือ ขั้นตอนวิธียกกำลังสองและการคูณ อย่างไรก็ตามหากผู้ใช้งานเลือกใช้ d ที่มีค่าต่ำจะส่งผลให้การโจมตีทำได้ง่ายมากขึ้นโดยใช้ขั้นตอนวิธีบางกลุ่มซึ่งจะกล่าวอีกครั้งในหัวข้อความปลอดภัยของวิทยาการรหัสลับอาร์เอสเอ

ดังนั้นในหัวข้อนี้จะกล่าวถึงเทคนิควิธีที่สามารถนำมาช่วยลดเวลาการคำนวณสมการถอดรหัสด้วยขั้นตอนวิธีอาร์เอสเอโดยไม่จำเป็นต้องลดขนาดของ d ดังนั้นความปลอดภัยยังคงเดิมในขณะเดียวกันกระบวนการถอดรหัสสามารถประมวลผลได้อย่างรวดเร็วมากยิ่งขึ้น

3.1 การประยุกต์ใช้ทฤษฎีเศษเหลือจีนสำหรับวิทยาการรหัสลับอาร์เอสเอ

การนำทฤษฎีเศษเหลือจีน [44] มาประยุกต์ใช้กับกระบวนถอดรหัสอาร์เอสเอสามารถช่วยลดเวลาการประมวลผล เนื่องจากค่า d ซึ่งถูกใช้เป็นเลขยกกำลังจะถูกแบ่งออกเป็นส่วนย่อยๆ โดยที่แต่ละส่วนมีขนาดที่เล็กลงเป็นอย่างมาก ซึ่งการคำนวณหา m ด้วยทฤษฎีเศษเหลือจีนสามารถดำเนินได้ด้วยการปรับสมการการถอดรหัสใหม่เป็นดังนี้

$$m = (m_p y_q q + m_q y_p p) \bmod n \qquad (7.5)$$

เมื่อ

$$m_p = c^{d \bmod p - 1} \bmod p \qquad (7.6)$$

$$m_q = c^{d \bmod q - 1} \bmod q \qquad (7.7)$$

และ $y_p p + y_q q = 1$ (7.8)

หรือ $y_p = p^{-1} \bmod q$ และ $y_q = q^{-1} \bmod p$

โดยที่ m จะมีความสัมพันธ์กับ m_p และ m_q ดังนี้

จากสมการ (7.6)

$$\begin{aligned} m_p &= (m^e)^{d \bmod (p-1)} \bmod p \\ &= m^{ed \bmod (p-1)} \bmod p \\ &= m^{(1+k\Phi(n) \bmod (p-1))} \bmod p \\ &= m^{(1+k((p-1)(q-1)) \bmod p-1)} \bmod p \\ &= m \bmod p \end{aligned}$$

จากสมการ (7.7)

$$\begin{aligned} m_q &= (m^e)^{d \bmod (q-1)} \bmod q \\ &= m^{ed \bmod (q-1)} \bmod q \\ &= m^{(1+k\Phi(n) \bmod (q-1))} \bmod q \\ &= m^{(1+k((p-1)(q-1)) \bmod q-1)} \bmod q \\ &= m \bmod q \end{aligned}$$

ดังนั้นจึงสามารถใช้ทฤษฎีเศษเหลือจีนสำหรับคำนวณหา m ได้ดังสมการที่ (7.5) และจากสมการที่ (7.8) สามารถคำนวณหา y_p และ y_q ได้โดยใช้ขั้นตอนวิธียุคลิดภาคขยาย

ตัวอย่างที่ 7.5 จากตัวอย่างที่ 7.1 จงถอดรหัสค่า $c = 891$ โดยวิธีการประยุกต์ใช้ทฤษฎีเศษเหลือจีน

วิธีทำ จากตัวอย่างที่ 7.1 ทราบว่า $p = 67$, $q = 79$, $d = 1927$ และ $n = 5293$ ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

จาก $m_p = 891^{1927 \bmod 66} \bmod 67$

$= 891^{13} \bmod 67$

$= 7$

และ $m_q = 891^{1927 \bmod 78} \bmod 79$

$= 891^{55} \bmod 79$

$= 62$

และจากสมการที่ (7.6)

$$67y_p + 79y_q = 1$$

ใช้ขั้นตอนวิธียุคลิดภาคขยายสำหรับหาค่า y_p และ y_q ได้ว่า

$y_p = -33$ (หรือ $y_p = 79 - 33 = 46$) และ $y_q = 28$

ดังนั้นจากสมการที่ (7.5) ได้ว่า

$m = (7x28x79 + 62 \times (-33) \times 67) \bmod 5293$

$= (15484 - 137082) \bmod 5293$

$= -121598 \bmod 5293$

$= 141$

3.2 การปรับสมการถอดรหัสใหม่

ในปี พ.ศ. 2561 ผู้เขียนได้เสนอผลงานวิจัยและได้รับการตอบรับเพื่อตีพิมพ์ลงหนังสือประมวลบทความในงานประชุมวิชาการ International Computer Science and Engineering Conference ครั้งที่ 17 [39] โดยผลงานที่นำเสนอเกี่ยวกับการปรับปรุงสมการการถอดรหัสสำหรับวิทยาการรหัสลับอาร์เอสเอใหม่ซึ่งสมการที่นำเสนอนี้มีประสิทธิภาพสูงมากหาก d มีค่าสูง ในทางกลับกันสมการดังกล่าวมีประสิทธิภาพต่ำหากนำไปประยุกต์ใช้กับ d ที่มีค่าต่ำ โดยสมการถอดรหัสที่ถูกปรับปรุงใหม่นี้ผ่านการพิสูจน์โดยใช้ทฤษฎีดังต่อไปนี้

ทฤษฎีบทที่ 7.1 กำหนดให้ c และ n คือจำนวนเฉพาะสัมพัทธ์ตรงกัน โดยที่ c^{-1} คือค่าผกผันธ์ของ c มอดูโลร์ n และผลลัพธ์ของ $x + d = \Phi(n)$ สมการการถอดรหัสลับอาร์เอสเอสามารถคำนวณได้จาก

$$m \equiv (c^{-1})^x \bmod n$$

พิสูจน์

เนื่องจากทฤษฎีบทข้างต้นกำหนดไว้ว่า c และ n คือจำนวนเฉพาะสัมพัทธ์ตรงกัน ดังนั้นจึงมี c^{-1} จากสมการดังต่อไปนี้เสมอ

$$c^{-1}c = 1 \bmod n$$

และจากสมการที่ (7.2)

$$m = c^d \bmod n$$

ได้ว่า

$$= (c^{-1}c \bmod n) \times (c^d \bmod n)$$

$$= \underbrace{(c^{-1}c \bmod n) \times \cdots \times (c^{-1}c \bmod n)}_{x \text{ ครั้ง}} \times (c^d \bmod n)$$

$$= ((c^{-1})^x c^x \bmod n) \times (c^d \bmod n)$$

$$= ((c^{-1})^x c^x c^d) \bmod n$$

$$= ((c^{-1})^x c^{x+d}) \bmod n$$

$$= ((c^{-1})^x \ c^{\Phi(n)}\ ) \bmod n$$

$$= ((c^{-1})^x \bmod n) \times (c^{\Phi(n)} \bmod n) \bmod n$$

จากทฤษฎีบทของออยเลอร์ได้ว่า $c^{\Phi(n)} \bmod n = 1$

ดังนั้น $m = (c^{-1})^x \bmod n$ □

จากทฤษฎีบทที่ 7.1 กล่าวได้ว่าสมการถอดรหัสอาร์เอสเอที่ถูกปรับปรุงใหม่นี้จะมีประสิทธิภาพสูงกรณีที่ d มีค่าสูงมากเนื่องจาก

จาก $x + d = \Phi(n)$

หรือ $x = \Phi(n) - d$ (7.9)

ดังนั้น x จึงมีค่าต่ำ และเนื่องจาก x ถูกนำมาใช้เป็นเลขยกกำลังแทน d จึงสรุปได้ว่าสามารถใช้สมการดังกล่าวเพื่อถอดรหัสข้อมูลได้อย่างรวดเร็ว

ในทางกลับกัน จากสมการที่ (7.9) สมการถอดรหัสที่ปรับปรุงใหม่นี้จะมีประสิทธิภาพต่ำหาก d มีค่าต่ำมาก ซึ่งจะส่งผลให้ x มีค่าสูง ซึ่งหากกรณีดังกล่าวนี้เกิดขึ้นควรใช้สมการถอดรหัสดั้งเดิมจะมีประสิทธิภาพที่สูงมากกว่า

ตัวอย่างที่ 7.6 แสดงวิธีการถอดรหัสสำหรับวิทยาการรหัสลับอาร์เอสเอในกรณีที่ d มีค่าสูง

วิธีทำ

กำหนดให้ $p = 929$ และ $q = 647$

ดังนั้น $n = 929 \times 647 = 601063$

และ $\Phi(n) = 928 \times 646 = 599488$

สมมติผู้สร้างกุญแจ เลือก $e = 362067$ ได้ว่า $d = 599387$ เนื่องจาก

$$362067 \times 599387 \bmod 599488 = 1$$

จากสมการ (7.9)

$$x = \Phi(n) - d$$
$$= 599488 - 599387 = 101$$

กำหนดให้ c คือข้อความไซเฟอร์ ได้ว่าสมการถอดรหัสอาร์เอสเอที่ปรับปรุงใหม่คือ

$$m = (c^{-1})^{101} \bmod 601063$$

ซึ่งหากเทียบสมการดังกล่าวกับสมการถอดรหัสอาร์เอสเอแบบดั้งเดิม ($m = c^{599387} \bmod 601063$) พบว่าเลขยกกำลังของสมการถอดรหัสที่ปรับปรุงใหม่มีค่าต่ำกว่ามาก จึงส่งผลให้ใช้เวลาสำหรับการประมวลผลลดลง

กำหนดให้ $m_1 = 721$ และ $m_2 = 1087$ แทนข้อความต้นฉบับ ดังนั้นสามารถคำนวณหาข้อความไซเฟอร์ของทั้งสองค่าด้วยการเข้ารหัสโดยดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$c_1 = 721^{362067} \bmod 601063 = 531908$$

และ $$c_2 = 1087^{362067} \bmod 601063 = 109813$$

ขั้นตอนถัดไปทดสอบถอดรหัส c_1 และ c_2 โดยใช้สมการถอดรหัสสำหรับวิทยาการรหัสลับอาร์เอสเอที่ปรับปรุงใหม่ได้ดังนี้

1. ถอดรหัส c_1 = 415058

ขั้นแรกจำเป็นต้องคำนวณหาของ c_1^{-1} ภายใต้การมอดูโล n ได้ว่า c_1^{-1} = 517294

เนื่องจาก

$$517294 \times 531908 \bmod 601063 = 1$$

ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$(c^{-1})^x \bmod n = 517294^{101} \bmod 601063$$
$$= 721$$
$$= m_1$$

2. ถอดรหัส c_2 = 96509

เช่นเดียวกับการถอดรหัส c_1 ขั้นแรกจำเป็นต้องคำนวณหาของ c_2^{-1} ภายใต้การมอดูโล n ได้ว่า c_2^{-1} = 96509

เนื่องจาก

$$96509 \times 109813 \bmod 601063 = 1$$

ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$(c^{-1})^x \bmod n = 96509^{101} \bmod 601063$$
$$= 1087$$
$$= m_2$$

จากตัวอย่างที่ 7.6 สรุปได้ว่าสามารถใช้สมการถอดรหัสที่ปรับปรุงใหม่จากทฤษฎีบทที่ 7.1 เพื่อใช้สำหรับคำนวณหาข้อความต้นฉบับได้ โดยสมการดังกล่าวเหมาะที่จะถูกนำมาประยุกต์ใช้งานในกรณีที่ d มีค่าสูง ซึ่งโดยทั่วไปการนำวิทยาการรหัสลับอาร์เอสเอมาประยุกต์ใช้งานจริงนั้นนิยมเลือกใช้งานค่า d ที่มีค่าสูงเพื่อเพิ่มความปลอดภัย ดังนั้นสมการดังกล่าวจึงเป็นประโยชน์อย่างมากที่จะนำมาลดเวลาสำหรับการประมวลผล

เนื่องจากก่อนที่จะคำนวณหาข้อความต้นฉบับโดยใช้สมการถอดรหัสแบบปรับปรุงใหม่นี้ จำเป็นต้องคำนวณหาค่าผกผันของข้อความไซเฟอร์ก่อนที่จะคำนวณโดยใช้สมการถอดรหัส ซึ่งจากบทนิยามที่ 2.1 ทราบว่าหากมีค่าผกผันของ c มอดูโล n แล้ว gcd(c, n) ต้องมีค่าเท่ากับ 1 ดังนั้นสรุป

ได้ว่าไม่สามารถใช้สมการถอดรหัสสำหรับวิทยาการรหัสลับอาร์เอสแบบปรับปรุงได้ก็ต่อเมื่อผลลัพธ์ของ gcd(c, n) มีค่าไม่เท่ากับ 1

อย่างไรก็ตามจำนวนเต็มที่ไม่มีตัวตัวผกผันภายใต้การมอดูโลร์ n คือค่าที่เกิดจากผลคูณระหว่างจำนวนเต็มและตัวประกอบตัวใดตัวหนึ่งของ n (p หรือ q) เช่น p, $2p$, $3p$, …, ip, q, $2q$, $3q$, …jq เมื่อ ip และ jq มีค่าน้อยกว่า n เนื่องมาจากค่าหารร่วมมากระหว่างค่ากลุ่มดังกล่าวและ n ไม่เท่ากับ 1

4. การเพิ่มความปลอดภัยวิทยาการรหัสลับอาร์เอสเอ

ความปลอดภัยของวิทยาการรหัสลับอาร์เอสเอขึ้นอยู่กับความยากของการแยกตัวประกอบ n เนื่องจากหากผู้ไม่ประสงค์ดีสามารถแยกตัวประกอบ n ได้ จะทำให้ทราบค่า p และ q และสามารถคำนวณค่า $\Phi(n)$ ได้เพื่อที่จะใช้สำหรับคำนวณหา d ในท้ายที่สุด ดังนั้นการเพิ่มความปลอดภัยแก่วิทยาการรหัสลับอาร์เอสเอจึงเป็นสิ่งที่สำคัญ หลายปีที่ผ่านมามีนักวิจัยได้ทำการเผยแพร่ผลงานวิจัยลงทั้งวารสาร และงานประชุมวิชาการเกี่ยวกับหัวข้อดังกล่าวนี้เป็นจำนวนมาก อย่างไรก็ตามในหัวข้อนี้จะกล่าวเทคนิคที่ใช้สำหรับการเพิ่มความปลอดภัยวิทยาการรหัสลับแบบอาร์เอสเอโดยใช้ตัวประกอบที่มากกว่า 2 ค่า [45] ซึ่งการประยุกต์ใช้หลักการดังกล่าวจะช่วยให้กระบวนการแยกตัวประกอบใช้เวลาสูงมากขึ้น เนื่องจากจำเป็นต้องแยกตัวประกอบมากกว่า 1 ครั้งเพื่อให้ได้มาซึ่งจำนวนประกอบที่เป็นจำนวนเฉพาะทั้งหมด โดยหลักการดังกล่าวถูกแบ่งออกเป็น 3 กระบวนการ ดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ:

1. เลือกจำนวนเฉพาะจำนวน i ค่า กำหนดเป็น p_1, p_2, p_3, …, p_i
2. คำนวณค่ามอดูลัสจาก $n = \prod_{j=1}^{i} p_j$
3. คำนวณค่าออยเลอร์จาก $\Phi(n) = \prod_{j=1}^{i} (p_j - 1)$
4. เลือกค่ากุญแจสาธารณะโดยมีเงื่อนไขว่า $1 < e < \Phi(n)$ และ gcd(e, $\Phi(n)$) = 1
5. คำนวณหาค่ากุญแจส่วนตัวจาก $d = e^{-1} \bmod \Phi(n)$

กุญแจสาธารณะคือ {e, n}

กุญแจส่วนตัวคือ {d}

กระบวนการที่ 2 การเข้ารหัสลับ: กระบวนการเข้ารหัสยังคงใช้สมการเช่นเดียวกันกับสมการดั้งเดิม

กระบวนการที่ 3 การถอดรหัสลับ: กระบวนการถอดรหัสสามารถเลือกใช้ได้ทั้งสมการดั้งเดิม หรือสมการปรับปรุงจากทฤษฎีบทที่ 7.1 โดยตัวอย่างประกอบการอธิบายหัวข้อนี้จะกล่าวถึงทั้ง 2 วิธี

ตัวอย่างที่ 7.7 การใช้วิทยาการรหัสลับอาร์เอสเอที่มีตัวประกอบมากกว่า 2 ค่า

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

1. เลือกจำนวนเฉพาะ $p_1 = 641$, $p_2 = 619$ และ $p_3 = 881$
2. คำนวณ $n = 641 \times 619 \times 881 = 349562299$
3. คำนวณ $\Phi(n) = 640 \times 618 \times 880 = 348057600$
4. เลือก $e = 195512057$ เนื่องจาก $1 < 195512057 < 348057600$ และ gcd(195512057, 348057600) = 1
5. คำนวณ $d = 195512057^{-1} \bmod 348057600 = 348056393$

กุญแจสาธารณะคือ {e = 195512057, n = 349562299}

กุญแจส่วนตัวคือ {d = 348056393}

กระบวนการเข้ารหัส

สมมติผู้ส่งต้องการเข้ารหัส m = 111587 ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

จาก $c = 111587^{195512057} \bmod 349562299$

$= 36904516$

และส่ง c = 36904516 ไปยังผู้รับ

กระบวนการถอดรหัส (โดยใช้สมการดั้งเดิม)

สมมติผู้รับต้องการถอดรหัส c = 36904516 ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

จาก $m = 36904516^{348056393} \bmod 349562299$

$= 111587$

กระบวนการถอดรหัส (โดยใช้สมการปรับปรุง)

สมมติผู้รับต้องการถอดรหัส c = 36904516 ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

เริ่มจากคำนวณหา c^{-1} ได้ว่า c^{-1} = 238479586

เนื่องจาก 238479586 x 36904516 mod 349562299 = 1

และคำนวณหา x จาก

$$x = \Phi(n) - d$$
$$= 348057600 - 348056393$$
$$= 1207$$

จาก $m = 238479586^{1207} \bmod 349562299$

$= 111587$

5. การประยุกต์ใช้ทฤษฎีเศษเหลือจีนสำหรับวิทยาการรหัสลับอาร์เอสเอที่มีตัวประกอบมากกว่า 2 ค่า

หัวข้อนี้จะกล่าวถึงการนำทฤษฎีเศษเหลือจีนมาประยุกต์ใช้สำหรับการเพิ่มความเร็วสมการถอดรหัสอาร์เอสเอในกรณีที่มีตัวประกอบมากกว่า 2 ค่า

กำหนดให้ $n = \prod_{j=1}^{i} p_j$ การคำนวณหา m ด้วยการนำทฤษฎีเศษเหลือจีนสามารถดำเนินได้โดยการปรับสมการการถอดรหัสใหม่เป็นดังนี้

$$m = \sum_{j=1}^{i} m_{pj} Y_{pj} T_{pj} \bmod n \qquad (7.10)$$

เมื่อ

$$m_{pj} = c^{d \bmod pj - 1} \bmod p_j \qquad (7.11)$$

$$T_{pj} = \prod_{k=1, k\neq j}^{i} p_k \qquad (7.12)$$

และ $$Y_{pj} = T_{pj}^{-1} \bmod p_j \qquad (7.13)$$

ตัวอย่างที่ 7.8 จงถอดรหัสค่า c = 36904516 จากตัวอย่างที่ 7.7 โดยวิธีการประยุกต์ใช้ทฤษฎีเศษเหลือจีน

วิธีทำ จากตัวอย่างที่ 7.7 ทราบว่า p_1 = 641, p_2 = 619 และ p_3 = 881, d = 348056393 และ n = 349562299 ดำเนินการคำนวณสมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

ลำดับที่ 1: หา m_{pi}, $m_{p1} = 36904516^{348056393 \bmod 640} \bmod 641$

$= 36904516^{73} \bmod 641$

$= 53$

$m_{p2} = 36904516^{348056393 \bmod 618} \bmod 619$

$= 36904516^{29} \bmod 619$

$= 167$

และ $m_{p3} = 36904516^{348056393 \bmod 880} \bmod 881$

$= 36904516^{553} \bmod 881$

$= 581$

ลำดับที่ 2: หา T_{pi}, $T_{p1} = 619 \times 881 = 545339$

$T_{p2} = 641 \times 881 = 564721$

$T_{p3} = 641 \times 619 = 396779$

ลำดับที่ 3: หา Y_{pi}, $Y_{p1} = 545339^{-1} \bmod 641$

$= 544$

$Y_{p2} = 564721^{-1} \bmod 619$

$= 263$

$Y_{p3} = 396779^{-1} \bmod 881$

$= 640$

ดังนั้น

$m = 53 \times 545339 \times 544 + 167 \times 564721 \times 263 + 581 \times 396779 \times 640 \bmod 349562299$

$= 15723214048 + 24803111041 + 147538303360 \bmod 349562299$

$= 111587$

จากตัวอย่างที่ 7.8 สังเกตได้ว่าเลขยกกำลังจากเดิมคือค่า d ถูกแบ่งออกเป็น 3 ค่า คือ 73, 29 และ 553 ซึ่งมีขนาดที่เล็กลงเป็นอย่างมาก ดังนั้นจึงสามารถนำทฤษฎีเศษเหลือจีนมาประยุกต์ใช้งานเพื่อลดเวลาสำหรับการถอดรหัสลงได้

6. การแยกตัวประกอบ

จากหลักการของวิทยาการรหัสลับอาร์เอสเอพบว่าค่า d และ $\Phi(n)$ คือค่าที่ผู้ไม่ประสงค์ดีพึงประสงค์เนื่องจากหากผู้ไม่ประสงค์ดีสามารถคำนวณหาค่า d กลับคืนได้จะส่งผลให้วิทยาการรหัสลับอาร์เอสเอถูกโจมตีได้โดยตรง โดยวิธีการแยกตัวประกอบเป็นวิธีหนึ่งที่สามารถนำมาใช้สำหรับ

โจมตีรหัสลับอาร์เอสเอได้ [26], [46] เนื่องมาจากหากผู้ไม่ประสงค์ดีสามารถแยกตัวประกอบค่า n ได้ จะทราบค่า p และ q ส่งผลให้สามารถคำนวณหา $\Phi(n)$ เพื่อใช้สำหรับคำนวณหาค่า d ได้ อย่างไรก็ตามหาก n มีขนาดอย่างน้อย 1024 บิต และตัวประกอบของ n ทุกค่าเป็นจำนวนเฉพาะที่แข็งแกร่ง (ความหมายของจำนวนเฉพาะที่แข็งแกร่งคือจำนวนเฉพาะที่ยากแก่การถูกโจมตี) ยังไม่พบขั้นตอนวิธีการแยกตัวประกอบใดที่สามารถแยกตัวประกอบ n ได้ในเวลาอันสั้น ดังนั้นวิทยาการรหัสลับอาร์เอสเอยังคงมีความปลอดภัยและถูกใช้งานในปัจจุบัน

อย่างไรก็ตามยังคงมีการพัฒนาขั้นตอนวิธีการแยกตัวประกอบทั้งในรูปแบบของการปรับปรุงขั้นตอนวิธีเดิม และการพัฒนาขึ้นใหม่เพื่อเพิ่มประสิทธิภาพให้สูงมากขึ้น ซึ่งหากงานวิจัยทางด้านนี้ยังคงพัฒนาอย่างต่อเนื่องอาจส่งผลให้วิทยาการรหัสลับอาร์เอสเอที่มีขนาด 1024 บิตมีความปลอดภัยไม่เพียงพออีกต่อไป โดยขั้นตอนวิธีการแยกตัวประกอบแต่ละประเภทจะมีประสิทธิภาพที่แตกต่างกันออกไปขึ้นอยู่กับคุณลักษณะและขนาดของตัวประกอบของค่า n

6.1 ขั้นตอนวิธีทดลองหาร

นอกจากการใช้ขั้นตอนวิธีทดลองหารสำหรับตรวจสอบความเป็นจำนวนเฉพาะแล้วยังสามารถใช้ขั้นตอนวิธีนี้สำหรับแยกตัวประกอบ n ได้เช่นกัน [23] โดยจากขั้นตอนวิธีทดลองหารกรณีที่ n คือจำนวนประกอบและ x คือตัวประกอบของ n ได้ว่า $\frac{n}{x}$ คือตัวประกอบของ n เช่นกัน

ตัวอย่างที่ 7.9 จงแสดงวิธีการแยกตัวประกอบ $n = 77$ โดยใช้ขั้นตอนวิธีทดลองหาร

วิธีทำ จากขั้นตอนวิธีทดลองหารได้ว่า $n = 77$ และ $\sqrt{n} = 8.77$ โดยเริ่มกระบวนการดังนี้

3. $x = 3$
4. $y = 77 \bmod 3 = 2$

ขั้นตอนที่ 3 – 6 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก $3 < 8.77$ (เป็นจริง) และ $2 \neq 0$ (เป็นจริง) จากเงื่อนไขทางตรรกศาสตร์ (จริง และ จริง ได้ผลลัพธ์เป็นจริง) จึงเข้าสู่วงวนการทำงาน

รอบที่ 1:

$x = 3 + 2 = 5$

$y = 77 \bmod 5 = 2$

เนื่องจาก 5 < 8.77 (เป็นจริง) และ 2 ≠ 0 (เป็นจริง) ได้ว่า

รอบที่ 2:

$x = 5 + 2 = 7$

$y = 77 \bmod 7 = 0$

เนื่องจาก 7 < 7.28 (เป็นจริง) และ $0 \neq 0$ (เป็นเท็จ) จึงออกจากรอบการทำงาน เงื่อนไขที่อยู่ในระหว่างขั้นตอนที่ 7 – 11

เนื่องจาก $y = 0$ จึงสรุปได้ว่า 77 **<u>เป็นจำนวนประกอบ</u>** โดยที่ $x = 7$ และ $\frac{n}{x} = 11$ คือตัวประกอบของ n

ขั้นตอนวิธีทดลองหารจะมีประสิทธิภาพที่สูงมากสำหรับกรณีที่ตัวประกอบอย่างน้อย 1 ค่ามีขนาดเล็กเนื่องจากตัวเลขสำหรับทดลองหารจะเริ่มจาก 3 ซึ่งเป็นค่าที่มีขนาดเล็กที่สุด และถูกเพิ่มขึ้นจนกระทั่งพบจำนวนเฉพาะที่เป็นตัวประกอบของ n

อย่างไรก็ตามขั้นตอนวิธีทดลองหารยังถูกปรับปรุงประสิทธิภาพเพื่อลดเวลาการทำงานดังนี้ จากขั้นตอนวิธีที่นำเสนอสังเกตได้ว่าตัวเลขที่จะนำมาใช้สำหรับการทดลองหารซึ่งเป็นเลขคี่เสมอมีโอกาสที่จะมีเลขหลักหน่วยที่มีค่าเท่ากับ 5 โดยที่กลุ่มตัวเลขลักษณะนี้เป็นจำนวนประกอบเสมอ (ยกเว้น 5) เนื่องจากสามารถนำ 5 ไปหารได้ลงตัว ดังนั้นหากตัวเลขที่ใช้สำหรับทดลองหารมีเลขหลักหน่วยเท่ากับ 5 ไม่ต้องนำค่าดังกล่าวมาทดลองหารเนื่องจากไม่เป็นจำนวนประกอบของ n อย่างแน่นอนซึ่งจะสามารถลดจำนวนรอบการคำนวณลงได้และส่งผลให้เวลาที่ใช้สำหรับการประมวลผลลดลง

การปรับปรุงขั้นตอนวิธีทดลองหารอีกวิธีหนึ่งคือการเลือกใช้เฉพาะตัวเลขที่เป็นจำนวนเฉพาะมาใช้สำหรับการทดลองหารซึ่งจะช่วยลดจำนวนรอบการคำนวณได้มากยิ่งขึ้น แต่อย่างไรก็ตามข้อเสียของวิธีนี้คือจำเป็นต้องใช้หน่วยความจำสำหรับจัดเก็บจำนวนเฉพาะที่นำมาใช้สำหรับทดลองหาร

เนื่องจากค่า p จะมีขนาดสูงสุดไม่เกิน $\lfloor \sqrt{n} \rfloor$ และหากค่า p และ q มีขนาดใหญ่และมีขนาดใกล้เคียงกันส่งผลให้ขั้นตอนวิธีทดลองหารมีประสิทธิภาพที่ต่ำมากจึงสามารถปรับเปลี่ยนขั้นตอนวิธีทดลองหารโดยกำหนดให้ตัวเลขที่จะนำมาใช้สำหรับการทดลองหารมีค่าเริ่มต้นที่ $\lfloor \sqrt{n} \rfloor$ และเปลี่ยนจากการเพิ่มค่าขึ้นเป็นการลดค่าลงในกรณีที่ผลหารไม่ลงตัวซึ่งจะช่วยสามารถแยกตัวประกอบได้อย่างรวดเร็ว เพราะค่า p จะมีค่าที่ใกล้เคียงกับ $\lfloor \sqrt{n} \rfloor$ เป็นอย่างมาก สำหรับขั้นตอนวิธีสามารถนำขั้นตอนวิธีที่ 5.1 มาปรับเปลี่ยนได้ดังนี้

ขั้นตอนวิธีที่ 7.1 การทดลองหาร (แบบปรับค่าลง)

```
     INPUT: n
     OUTPUT: ชนิดของ n (จำนวนเฉพาะหรือจำนวนประกอบ)
1:   x ← ⌊√n⌋
2:   If x is even then
3:       x ← x - 1
4:   y ← n mod x
5:
6:   While ((x > 0) and (y ≠ 0)) do
7:       x ← x - 2
8:       y ← n mod x
9:   End While
10:
11:  IF (y ≠ 0) then
12:      n is prime number
13:  Else
         n is composite number
     End IF
```

ตัวอย่างที่ 7.10 จงแสดงวิธีการแยกตัวประกอบ n = 1267146373 โดยใช้ขั้นตอนวิธีทดลองหาร (แบบปรับค่าลง)

วิธีทำ จากขั้นตอนวิธีทดลองหารได้ว่า n = 1267146373 และ $\lfloor \sqrt{n} \rfloor$ = 35596 โดยเริ่มกระบวนการดังนี้

1. x = 35596

2 – 3. เนื่องจาก x เป็นจำนวนเต็มคู่ดังนั้น $x = x - 1$ = 35595

4. y = 1267146373 mod 35595 = 35563

ขั้นตอนที่ 5 – 8 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก 35595 > 0 (เป็นจริง) และ 35563 ≠ 0 (เป็นจริง) จากเงื่อนไขทางตรรกศาสตร์ (จริง และ จริง ได้ผลลัพธ์เป็นจริง) จึงเข้าสู่รอบการทำงาน

รอบที่ 1:

x = 35595 - 2 = 35593

y = 1267146373 mod 35593 = 35573

เนื่องจาก 35593 > 0 (เป็นจริง) และ 35573 ≠ 0 (เป็นจริง) ได้ว่า

รอบที่ 2:

x = 35593 - 2 = 35591

y = 1267146373 mod 35591 = 0

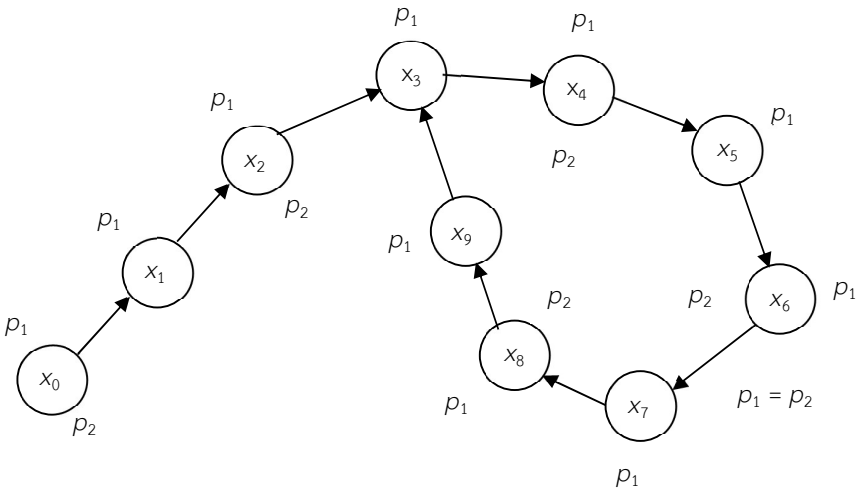
เงื่อนไขที่อยู่ในระหว่างขั้นตอนที่ 7 – 11

เนื่องจาก y = 0 จึงสรุปได้ว่า 35591 **<u>เป็นจำนวนประกอบ</u>** โดยที่ x = 35591 และ $\frac{n}{x}$ = 35603 คือตัวประกอบของ n

จากตัวอย่างข้างต้นนี้สังเกตว่าถึงแม้ว่า n จะมีขนาดใหญ่ แต่ขั้นตอนวิธีทดลองหาร (แบบปรับค่าลง) สามารถแยกตัวประกอบได้อย่างรวดเร็ว เนื่องจาก p มีขนาดที่ใกล้เคียงกับ $\lfloor \sqrt{n} \rfloor$ ในทางกลับกันหากนำขั้นตอนวิธีทดลองหารแบบดั้งเดิมมาใช้สำหรับแยกตัวประกอบค่าดังกล่าวนี้จะใช้รอบการคำนวณที่สูง อย่างไรก็ตามในความเป็นจริงนั้นเนื่องจากมีเพียง n ที่ถูกประกาศเป็นสาธารณะเท่านั้นจึงไม่ทราบลักษณะของค่า p และ q ส่งผลให้ไม่สามารถเลือกขั้นตอนวิธีที่เหมาะสมได้

6.2 ขั้นตอนวิธีการแยกตัวประกอบโรห์ของโพลลาร์ด

โรห์ของโพลลาร์ด (Pollard's Rho) [11] คือ วิธีการแยกตัวประกอบที่ถูกเสนอโดย จอห์น โพลลาร์ด (John Pollard) โดยมีแนวคิดพื้นฐานมาจากการเกิดวงวนของฟลอยน์ (Floyd's cycle Finding) ซึ่งมีหลักการคือกำหนดให้ p_1 และ p_2 มีจุดเริ่มต้นเดียวกันแต่มีจังหวะการเคลื่อนที่ๆ แตกต่างกัน หากทั้งสองค่านี้เคลื่อนที่เข้าสู่วงวนแล้วจะพบกันได้ในบางจุดภายในวงวนซึ่งพิจารณาได้ดังรูปที่ 7.1



รูปที่ 7.1 ตัวอย่างการทำงานโรห์ของโพลลาร์ด

จากรูปที่ 7.1 กำหนดให้ p_1 และ p_2 เริ่มต้นที่จุด x_0 โดยที่ p_1 มีจังหวะการเคลื่อนที่ครั้งละ 1 จังหวะ แต่ในการกลับกัน p_2 มีจังหวะการเคลื่อนที่ครั้งละ 2 จังหวะเมื่อทั้งสองเข้าสู่วงวนจะมีโอกาสที่จะพบกันได้ในบางตำแหน่งซึ่งจากตัวอย่างทั้งสองค่าพบกันที่ตำแหน่ง x_6 โดยสามารถนำหลักการนี้มาประยุกต์ใช้กับการแยกตัวประกอบ n ได้โดยการหาตัวเลข 2 ค่า (กำหนดเป็น x และ y) ที่เป็น สมภาคกันภายใต้การมอดูโล p จึงได้ว่า $p \mid (x - y)$ ดังนั้นหากทราบ x และ y จะสามารถคำนวณหา p ได้จาก gcd($x - y$, n) สำหรับขั้นตอนวิธีโรห์ของโพลลาร์ดมีลำดับขั้นตอนเป็นดังนี้

ขั้นตอนวิธีที่ 7.2 โรห์ของโพลลาร์ด

```
     INPUT: n
     OUTPUT: p, q
1:   i ← 2
2:   สุ่มหาค่า m_0
3:   m_1 ← m_0^2
4:   m_2 ← (m_1^2 + 1) mod n
5:   While (gcd(m_i-m_i/2, n) == 1) do
6:       m_i+1 ← (m_i^2 + 1) mod n
7:
8:       m_i+2 ← (m_i+1^2 + 1) mod n
9:       i ← i + 2
10:  End While
11:  p ← gcd(m_i-m_i/2, n)
12:  q ← n / p
```

ตัวอย่างที่ 7.11 จงแสดงวิธีการแยกตัวประกอบ $n = 221$ โดยใช้ขั้นตอนวิธีโรห์ของโพลลาร์ด

วิธีทำ จากขั้นตอนวิธีโรห์ของโพลลาร์ดได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

1. $i = 2$
2. $m_0 = 3$ (หากดำเนินการใหม่อาจไม่ได้ค่านี้ เนื่องจากเป็นการสุ่ม)
3. $m_1 = 3^2 \bmod 221 = 9$
4. $m_2 = (9^2 + 1) \bmod 221 = 82$

ขั้นตอนที่ 5 – 9 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก gcd(73, 221) = 1 ดังนั้น

รอบที่ 1

6. $m_3 = (82^2 + 1) \bmod 221 = 95$

7. $m_4 = (95^2 + 1) \bmod 221 = 186$

8. $i = 4$

เนื่องจาก gcd(91, 221) = 13 $\neq$ 1 ดังนั้น

10. $p = 13$

11 $q = \frac{221}{13} = 17$

อย่างไรก็ตามขั้นตอนวิธีโรห์ของโพลลาร์ดจะมีประสิทธิภาพสูงเพียงเฉพาะในกรณีที่ n มีขนาดเล็กซึ่งควรมีขนาดไม่เกิน 10^{15}

6.3 ขั้นตอนวิธีการแยกตัวประกอบ $p - 1$ ของโพลลาร์ด

นอกเหนือจากโรห์ของโพลลาร์ดแล้วโพลลาร์ดยังได้เสนอวิธีการแยกตัวประกอบอีกวิธีหนึ่งซึ่งเป็นขั้นตอนวิธีที่เกิดก่อนขั้นตอนวิธีโรห์ของโพลลาร์ดเรียกว่าขั้นตอนวิธีการแยกตัวประกอบ $p - 1$ ของโพลลาร์ด [14] ซึ่งเป็นวิธีที่มีประสิทธิภาพที่สูงมากในกรณีที่ตัวประกอบทั้งหมดของ $p - 1$ หรือ q - 1 มีขนาดเล็ก

กำหนดให้ $k \in \mathbb{Z}$ โดยที่ $k! = (p - 1)q$ และจากทฤษฎีบทที่ 5.2 ได้ว่า

$$\begin{aligned} a^{k!} &= a^{(p-1)q} \bmod p \\ &= (a^{(p-1)})^q \bmod p \\ &= (1)^q \bmod p \\ &= 1 \bmod p \end{aligned}$$

ดังนั้นกล่าวได้ว่า $p \mid (a^{k!} - 1)$

เนื่องจาก $p \mid (a^{k!} - 1)$ และ $p \mid n$ จึงสรุปได้ว่า $\gcd((a^{k!} - 1, n) = p$

สำหรับวิธีการคำนวณหา p สามารถดำเนินการได้โดยสุ่มหาค่า a และกำหนดค่าเริ่มต้นของ $k = 1$ คำนวณหาผลลัพธ์ด้วยสมการข้างต้นคือ $a^k \bmod p$ และหาค่า $\gcd((a^{k!} - 1, n)$ โดยหากผลลัพธ์มีค่าเป็น 1 จะคำนวณหาผลลัพธ์ใหม่โดยเปลี่ยนค่า $k = k(k+1)$ และคำนวณ $(a^k)^{k+1} \bmod p$ ซึ่งจะดำเนินการลักษณะเช่นนี้ไปจนกระทั่งผลลัพธ์ของค่าหารร่วมมากมีค่าไม่เท่ากับ 1 จึงจะหยุดการทำงานและสรุปว่าผลลัพธ์ดังกล่าวคือค่า p โดยมีขั้นตอนวิธีเป็นดังนี้

ขั้นตอนวิธีที่ 7.3 p – 1 ของโพลลาร์ด

```
     INPUT: n
     OUTPUT: p, q
1:   สุ่มหาค่า b
2:   i ← 2
3:   r1 ← b mod n
4:   r2 ← r1^2 mod n
5:   While (gcd(ri-1, n) == 1) do
6:       i ← i + 1
7:       ri ← r(i-1)^i mod n
8:   End While
9:   p ← gcd(ri-1, n)
10:  q ← n / p
```

ตัวอย่างที่ 7.12 จงแสดงวิธีการแยกตัวประกอบ n = 32541217399 โดยใช้ขั้นตอนวิธี $p - 1$ ของโพลลาร์ด

วิธีทำ จากขั้นตอนวิธี $p - 1$ ของโพลลาร์ดได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

1. $b = 3$
2. $i = 2$
3. $r_1 = 3 \bmod 32541217399 = 3$
4. $r_2 = 3^2 \bmod 32541217399 = 9$

ขั้นตอนที่ 5 – 8 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก gcd(8, 32541217399) = 1 ดังนั้น

รอบที่ 1

6. $i = 3$
7. $r_3 = 9^3 \bmod 32541217399 = 729$

เนื่องจาก gcd(728, 32541217399) = 1 ดังนั้น

รอบที่ 2

6. $i = 4$
7. $r_4 = 729^4 \bmod 32541217399 = 22099797289$

เนื่องจาก gcd(22099797288, 32541217399) = 1 ดังนั้น

รอบที่ 3

6. $i = 5$

7. $r_5 = 22099797289^5 \bmod 32541217399 = 8234274967$

เนื่องจาก gcd(8234274966, 32541217399) = 1 ดังนั้น

รอบที่ 4

6. $i = 6$

7. $r_6 = 8234274967^6 \bmod 32541217399 = 836783318$

เนื่องจาก gcd(836783317, 32541217399) = 1 ดังนั้น

รอบที่ 5

6. $i = 7$

7. $r_7 = 836783318^7 \bmod 32541217399 = 18702279589$

เนื่องจาก gcd(18702279588, 32541217399) = 1 ดังนั้น

รอบที่ 6

6. $i = 8$

7. $r_8 = 18702279589^8 \bmod 32541217399 = 26968920166$

เนื่องจาก gcd(26968920165, 32541217399) = 1 ดังนั้น

รอบที่ 7

6. $i = 9$

7. $r_9 = 26968920166^9 \bmod 32541217399 = 5117164259$

เนื่องจาก gcd(5117164258, 32541217399) = 1 ดังนั้น

รอบที่ 8

6. $i = 10$

7. $r_{10} = 5117164259^{10} \bmod 32541217399 = 6132547255$

เนื่องจาก gcd(6132547254, 32541217399) = 1 ดังนั้น

รอบที่ 9

6. $i = 11$

7. $r_{11} = 6132547255^{11} \bmod 32541217399 = 18649463035$

เนื่องจาก gcd(18649463034, 32541217399) = 1 ดังนั้น

รอบที่ 10

6. $i = 12$

7. $r_{12} = 18649463035^{12} \bmod 32541217399 = 4988139815$

เนื่องจาก gcd(4988139814, 32541217399) = 1 ดังนั้น

รอบที่ 11

6. $i = 13$

7. $r_{13} = 4988139815^{13} \bmod 32541217399 = 14736794452$

เนื่องจาก gcd(14736794451, 32541217399) = 1 ดังนั้น

รอบที่ 12

6. $i = 14$

7. $r_{14} = 14736794452^{14} \bmod 32541217399 = 29737374882$

เนื่องจาก gcd(29737374881, 32541217399) = 1 ดังนั้น

รอบที่ 13

6. $i = 15$

7. $r_{15} = 29737374882^{15} \bmod 32541217399 = 24097923132$

เนื่องจาก gcd(24097923131, 32541217399) = 1 ดังนั้น

รอบที่ 14

6. $i = 16$

7. $r_{16} = 24097923132^{16} \bmod 32541217399 = 4135504334$

เนื่องจาก gcd(4135504333, 32541217399) = 1 ดังนั้น

รอบที่ 15

6. $i = 17$

7. $r_{17} = 4135504334^{17} \bmod 32541217399 = 10400242199$

เนื่องจาก gcd(10400242198, 32541217399) = 1 ดังนั้น

รอบที่ 16

6. $i = 18$

7. $r_{18} = 10400242199^{18} \bmod 32541217399 = 29951431956$

เนื่องจาก gcd(29951431955, 32541217399) = 1 ดังนั้น

รอบที่ 17

6. $i = 19$

7. $r_{19} = 29951431956^{19} \bmod 32541217399 = 17435950345$

เนื่องจาก gcd(17435950344, 32541217399) = 1 ดังนั้น

รอบที่ 18

6. $i = 20$

7. $r_{20} = 17435950345^{20} \bmod 32541217399 = 7101901710$

เนื่องจาก gcd(7101901709, 32541217399) = 1 ดังนั้น

รอบที่ 19

6. $i = 21$

7. $r_{21} = 7101901710^{21} \bmod 32541217399 = 19833085028$

เนื่องจาก gcd(19833085027, 32541217399) = 1 ดังนั้น

รอบที่ 20

6. $i = 22$

7. $r_{22} = 19833085028^{22} \bmod 32541217399 = 21384318021$

เนื่องจาก gcd(21384318020, 32541217399) = 284593 ดังนั้น

9. $p = 284593$

10. $q = \dfrac{32541217399}{284593} = 114343$

จากตัวอย่างที่ 7.12 ถึงแม้ว่า n จะมีขนาดใหญ่แต่เนื่องจากตัวประกอบของ $p - 1$ มีขนาดเล็กดังนี้

$$p - 1 = 284593 - 1 = 284592 = 2^4 \times 3 \times 7^2 \times 11^2$$

ดังนั้นการใช้ขั้นตอนวิธี $p - 1$ ของโพลลาร์ดสำหรับแยกตัวประกอบตัวอย่างนี้จึงมีประสิทธิภาพที่สูงซึ่งมีการดำเนินการเพียง 20 รอบ

6.4 ขั้นตอนวิธีการแยกตัวประกอบทดลองหารแบบทั่วไป

ในปี ค.ศ. 2011 มุรัส ซาริน (Murat Sahin) [12] ได้เสนอการปรับปรุงขั้นตอนวิธีการทดลองหารโดยใช้ชื่อว่าการทดลองหารแบบทั่วไป (Generalized Trial Division) โดยใช้วิธีการทดลองหาค่าหารร่วมมากระหว่างตัวเลขที่เลือกและค่า n แทนที่วิธีการทดลองหาร

กำหนดให้ $a = kp$ และ $b = jp$ เมื่อ $k, j < q$ ดังนั้นได้ว่า

$$\gcd(a, n) = p$$

และ

$$\gcd(b, n) = p$$

กำหนดให้ค่าเริ่มต้นของ $a = \left\lfloor \sqrt{n} \right\rfloor + 1$ และ $b = \left\lfloor \sqrt{n} \right\rfloor - 1$ ดังนั้นหากผลลัพธ์ของหารร่วมมากระหว่างค่าใดค่าหนึ่งและ n มีค่าไม่เท่ากับ 1 จะหยุดการทำงานและได้ว่าผลลัพธ์นี้คือตัวประกอบของ n ในทางกลับกันหากผลลัพธ์ทั้งสองค่ามีค่าเป็น 1 จำเป็นต้องเพิ่มค่า a ขึ้นอีก 1 ค่าและลดค่า b ลงอีก 1 ค่าเพื่อหาผลลัพธ์ใหม่จนกระทั่งพบผลลัพธ์ที่มีค่าไม่เป็น 1

จากหลักการข้างต้นกล่าวได้ว่าขั้นตอนวิธีทดลองหารแบบทั่วไปจะมีประสิทธิภาพที่สูงมากในกรณีที่ ip มีค่าที่ใกล้เคียง $\left\lfloor \sqrt{n} \right\rfloor$ เมื่อ $i \in \mathbb{Z}^+$ โดยมีขั้นตอนวิธีเป็นดังนี้

ขั้นตอนวิธีที่ 7.4 การทดลองหารแบบทั่วไป

```
     INPUT: n
     OUTPUT: p, q
1:   i ← ⌊√n⌋
2:   a ← i + 1
3:   b ← i - 1
4:   A ← gcd(a, n)
5:   B ← gcd(b, n)
6:   While ((A == 1) and (B == 1)) do
7:       a ← a + 1
8:       b ← b + 1
9:       A ← gcd(a, n)
10:      B ← gcd(b, n)
11:  End While
12:
13:  IF (A ≠ 1) then
14:      p ← A
15:  Else
16:      P ← B
     End IF
17:
     q ← n/p
```

ตัวอย่างที่ 7.13 จงแยกตัวประกอบ n = 3570739 โดยใช้ขั้นตอนวิธีการทดลองหารแบบทั่วไป

วิธีทำ จากขั้นตอนวิธีการทดลองหารแบบทั่วไปได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

1. $i = \left\lfloor \sqrt{3570739} \right\rfloor = 1889$
2. $a = 1890$
3. $b = 1888$
4. A = gcd(1890, 3570739) = 1
5. B = gcd(1888, 3570739) = 59

ขั้นตอนที่ 6 – 11 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก 1 = 1 (จริง) และ 59 = 1 (เท็จ) จึงได้ผลลัพธ์เป็นเท็จ ดังนั้นจึงไม่มีการดำเนินการภายในวงวน

ขั้นตอนที่ 12 – 16 คือคำสั่งเงื่อนไข ดังนี้

เนื่องจาก $B = 59 \neq 1$ ดังนั้น p =59

17. $q = \dfrac{3570739}{59} = 60521$

จากตัวอย่างที่ 7.13 ถึงแม้ว่า n จะมีขนาดใหญ่แต่เนื่องจาก $32p$ = 1888 มีค่าใกล้เคียง $\lfloor \sqrt{n} \rfloor$ = 1889 เป็นอย่างมาก ดังนั้นการใช้ขั้นตอนวิธีการทดลองหารแบบทั่วไปสำหรับแยกตัวประกอบตัวอย่างนี้จึงมีประสิทธิภาพที่สูงมากซึ่งจากตัวอย่างมีการดำเนินการเพียงแค่รอบเดียว

6.5 ขั้นตอนวิธีการแยกตัวประกอบวีแฟกเตอร์

ขั้นตอนวิธีการแยกตัวประกอบวีแฟกเตอร์ (VFactor) [13] เป็นผลงานวิจัยที่ถูกนำเสนอโดย ปราชานท์ ชาร์มา (Prashant Sharma) และคณะในช่วงปี ค.ศ. 2012 ที่ถูกเผยแพร่ในหนังสือประมวลบทความในการประชุมทางวิชาการของ IEEE Explore ซึ่งใช้หลักการแยกตัวประกอบที่เรียบง่ายคือกำหนดจำนวนเต็มบวกคี่จำนวน 2 ค่าซึ่งค่าที่หนึ่งจะเป็นค่าที่น้อยที่สุดที่มีค่ามากกว่า $\lfloor \sqrt{n} \rfloor$ ส่วนอีกค่าหนึ่งเป็นค่าที่มากที่สุดที่มีค่าน้อยกว่า $\lfloor \sqrt{n} \rfloor$ โดยจะนำทั้งสองค่านี้มาทดสอบคูณกันซึ่งหากผลลัพธ์ที่ได้มีค่าเท่ากับ n สรุปได้ว่าจำนวนเต็มทั้ง 2 ค่านี้คือตัวประกอบของ n ในทางกลับกันหากผลลัพธ์ที่ได้ไม่เท่ากับ n จะเกิดเงื่อนไขเป็น 2 กรณีดังนี้ กรณีที่ 1 ผลลัพธ์ที่ได้มีค่าน้อยกว่า n จะต้องเพิ่มค่าตัวเลขที่มีค่ามากกว่าขึ้นอีก 2 ค่า แต่หากผลลัพธ์ที่ได้มีค่ามากกว่า n จะต้องลดค่าตัวเลขที่มีค่าน้อยกว่าลงอีก 2 ค่า โดยหลังจากเพิ่มหรือลดค่าใดค่าหนึ่งเรียบร้อยแล้วจะนำตัวเลขทั้งสองค่ามาทดสอบคูณกันอีกครั้งซึ่งขั้นตอนการดำเนินการจะเป็นลักษณะวนซ้ำเช่นนี้จนกระทั่งผลลัพธ์ที่มีค่าเท่ากับ n ถูกตรวจพบ โดยมีขั้นตอนวิธีเป็นดังนี้

ขั้นตอนวิธีที่ 7.5 วีแฟกเตอร์

```
     INPUT: n
     OUTPUT: p, q
1:   i ← ⌊√n⌋
2:   IF (i mod 2 == 0) then
3:       i ← i – 1
4:   End IF
5:   y ← i
6:   x ← i + 2
7:   t ← x*y
8:
9:   While (t ≠ n) do
10:      IF (t < n) then
11:          x ← x + 2
12:      Else
13:          y ← y – 2
14:      End IF
15:      t ← x*y
16:  End While
17:  p ← x
     q ← y
```

ตัวอย่างที่ 7.14 จงแยกตัวประกอบ n = 589106906053 โดยใช้ขั้นตอนวิธีวีแฟกเตอร์

วิธีทำ จากขั้นตอนวิธีวีแฟกเตอร์ได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

1. $i = \left\lfloor \sqrt{589106906053} \right\rfloor = 767532$

เงื่อนไขที่อยู่ในระหว่างขั้นตอนที่ 2 – 4, เนื่องจาก 767532 mod 2 = 0 ดังนั้น

i = 767532 – 1 = 767531

5. y = 767531

6. x = 767531 + 2 = 767533

7. t = 767531 x 767533 = 589105371023

ขั้นตอนที่ 8 – 15 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก $t \neq$ 589106906053 ดังนั้น

รอบที่ 1:

9 – 13 เนื่องจาก $t <$ 589106906053 ดังนั้น

x = 767533 + 2 = 767535

14. t = 767535 x 767531 = 589106906085

เนื่องจาก $t \neq$ 589106906053 ดังนั้น

รอบที่ 2:

9 – 13 เนื่องจาก $t >$ 589106906053 ดังนั้น

y = 767531 - 2 = 767529

14. t = 767535 x 767529 = 589105371015

เนื่องจาก $t \neq$ 589106906053 ดังนั้น

รอบที่ 3:

9 – 13 เนื่องจาก $t <$ 589106906053 ดังนั้น

x = 767535 + 2 = 767537

14. t = 767537 x 767529 = 589106906073

เนื่องจาก $t \neq 589106906053$ ดังนั้น

รอบที่ 4:

9 – 13 เนื่องจาก $t > 589106906053$ ดังนั้น

$y = 767529 - 2 = 767527$

14. $t = 767537 \times 767527 = 589105370999$

เนื่องจาก $t \neq 589106906053$ ดังนั้น

รอบที่ 5:

9 – 13 เนื่องจาก $t < 589106906053$ ดังนั้น

$x = 767537 + 2 = 767539$

14. $t = 767539 \times 767527 = 589106906053$

เนื่องจาก $t = 589106906053$ ดังนั้น

16. $p = x = 767539$

17. $q = y = 767527$

จากตัวอย่างนี้ สังเกตได้ว่าขั้นตอนวิธีวีแฟกเตอร์สามารถแยกตัวประกอบค่า $n = 589106906053$ ซึ่งมีขนาดใหญ่ด้วยรอบการคำนวณเพียงแค่ 5 รอบ เนื่องจากขนาดของ p และ q มีค่าใกล้เคียงกัน อย่างไรก็ตามหากนำขั้นตอนวิธีวีแฟกเตอร์ไปประยุกต์ใช้แยกตัวประกอบที่เกิดจากตัวประกอบมีขนาดที่แตกต่างกันมากจะมีประสิทธิภาพต่ำเนื่องจากใช้จำนวนรอบการคำนวณสูง

6.6 ขั้นตอนวิธีการแยกตัวประกอบของแฟร์มาต์

ขั้นตอนวิธีการแยกตัวประกอบของแฟร์มาต์ (Fermat's Factorization Algorithm) [17] เป็นขั้นตอนวิธีการแยกตัวประกอบอีกวิธีหนึ่งที่มีประสิทธิภาพสูงมากในกรณีที่ p และ q มีขนาดที่ใกล้เคียงกัน ถึงแม้ว่าทั้งสองค่านี้จะมีขนาดที่ใหญ่มหาศาล ขั้นตอนวิธีนี้ถูกค้นพบโดย ปิแอร์ เดอร์แฟร์มาต์ ในช่วงปี ค.ศ. 1600 โดยแฟร์มาต์พบว่าหากจำนวนประกอบเป็นจำนวนเต็มที่เกิดจากการคูณกันระหว่างจำนวนเฉพาะสองค่าแล้วจะสามารถถูกเขียนให้อยู่ในรูปของสมการผลต่างยกกำลังสองได้ดังนี้

$$n = x^2 - y^2 \tag{7.14}$$

เมื่อ $x = \frac{p+q}{2}$ และ $y = \frac{p-q}{2}$

โดยหากแทนค่า x และ y ในสมการ (7.14) ได้ว่า

$$\begin{aligned} n &= (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2 \\ &= (\frac{p^2+2pq+q^2}{4}) - (\frac{p^2-2pq+q^2}{4}) \\ &= \frac{4pq}{4} \\ &= pq \end{aligned}$$

กระบวนการแยกตัวประกอบโดยใช้ขั้นตอนวิธีของแฟร์มาต์เริ่มจากกำหนดค่าเริ่มต้น $x = \lceil \sqrt{n} \rceil$ และคำนวณหา $y = \sqrt{x^2 - n}$ ซึ่งเกิดจากการย้ายข้างของสมการที่ (7.14) โดยหากผลลัพธ์ของ y เป็นจำนวนเต็มจะสรุปได้ว่า $p = x + y$ และ $q = x - y$ คือตัวประกอบของ n ในทางกลับกันหากผลลัพธ์ไม่เป็นจำนวนเต็มจะต้องดำเนินการเพิ่มค่า x ขึ้นเพื่อคำนวนหาค่า y ใหม่จนกระทั่ง y ที่เป็นจำนวนเต็มถูกพบโดยขั้นตอนวิธีการแยกตัวประกอบของแฟร์มาต์ [15] เป็นดังนี้

ขั้นตอนวิธีที่ 7.6 วิธีของแฟร์มาต์

```
     INPUT: n
     OUTPUT: p, q
1:   x ← ⌈√n⌉
2:   y ← √(x² - n)
3:   While (y is not an integer) do
4:       x ← x + 1
5:       y ← √(x² - n)
6:   End While
7:   p ← x + y
8:   q ← x - y
```

ตัวอย่างที่ 7.15 จงแสดงวิธีการแยกตัวประกอบ $n = 344381$ โดยใช้ขั้นตอนวิธีของแฟร์มาต์

วิธีทำ จากขั้นตอนวิธีของแฟร์มาต์ได้ลำดับขั้นตอนการดำเนินการเป็นดังนี้

3. $x = \lceil \sqrt{344381} \rceil = 587$

4. $y = \sqrt{587^2 - 344381} = 13.71$

ขั้นตอนที่ 3 – 6 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก y ไม่เป็นจำนวนเต็มดังนั้น

รอบที่ 1:

4. $x = 587 + 1 = 588$

5. $y = \sqrt{588^2 - 344381} = 36.92$

เนื่องจาก y ไม่เป็นจำนวนเต็มดังนั้น

รอบที่ 2:

4. $x = 588 + 1 = 589$

5. $y = \sqrt{589^2 - 344381} = 50.39$

เนื่องจาก y ไม่เป็นจำนวนเต็มดังนั้น

รอบที่ 3:

4. $x = 589 + 1 = 590$

5. $y = \sqrt{590^2 - 344381} = 60.98$

เนื่องจาก y ไม่เป็นจำนวนเต็มดังนั้น

รอบที่ 4:

4. $x = 590 + 1 = 591$

5. $y = \sqrt{591^2 - 344381} = 70$

เนื่องจาก y เป็นจำนวนเต็มแล้วดังนั้นได้ว่า

$p = 591 + 70 = 661$ และ $q = 591 - 70 = 521$

อย่างไรก็ตามในแต่ละรอบของการคำนวณจำเป็นต้องคำนวณค่ารากที่สองเพื่อหาจำนวนเต็มของค่า y ซึ่งโดยทั่วไปการคำนวณหาค่ารากที่สองโดยเฉพาะอย่างยิ่งของค่าตัวเลขที่มีขนาดใหญ่

จำเป็นต้องใช้เวลาการประมวลผลสูง หากเปรียบเทียบกับกระบวนการบวก ลบ และคูณ ดังนั้นเพื่อหลีกเลี่ยงการคำนวณหารากที่สองในแต่ละรอบของการคำนวณจึงปรับสมการ (7.14) ดังนี้

จาก $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$

ได้ว่า $4n = (p+q)^2 - (p-q)^2$

กำหนดให้ $u = p + q$ และ $v = p - q$ ดังนั้น

$$4n = u^2 - v^2$$

หรือ
$$u^2 - v^2 - 4n = 0 \qquad (7.15)$$

กำหนดให้ค่าเริ่มต้นของ $u = 2\lceil \sqrt{n} \rceil$ และ $v = 0$ และหากแทนทั้งสองค่าในสมการที่ (7.15) แล้วทำให้สมการเป็นจริงกล่าวได้ว่า $p = \frac{u+v}{2}$ และ $q = \frac{u-v}{2}$ คือตัวประกอบของ n ในทางกลับกันหากผลลัพธ์ส่งผลให้สมการเป็นเท็จจะแบ่งผลลัพธ์ออกเป็น 2 กรณีดังนี้

กำหนดให้ $r = u^2 - v^2 - 4n$ ดังนั้น

กรณีที่ 1: $r > 0$

ความหมายคือค่า v ปัจจุบันมีค่าน้อยเกินความเป็นจริงจึงจำเป็นต้องเพิ่มค่า v ขึ้น อย่างไรก็ตามเนื่องจาก p และ q เป็นจำนวนเฉพาะที่เป็นจำนวนเต็มบวกคี่เสมอ ส่งผลให้ผลลัพธ์ของ $v = p - q$ เป็นจำนวนเต็มบวกคู่ ดังนั้นการเพิ่มค่าของ v จึงสามารถเพิ่มขึ้นครั้งละ 2 ค่าได้เพื่อปรับให้ v เป็นจำนวนเต็มบวกคู่ โดยหลังจากปรับค่า v ใหม่แล้วจึงต้องปรับค่า r ด้วยพิจารณาได้ดังนี้

กำหนดให้, $r_n = u^2 - v_n^2 - 4n$
เมื่อ v_n คือค่า v ที่ปรับใหม่ซึ่งมีค่าเป็น $v_n = v + 2$
r_n คือค่า r ที่ปรับใหม่

ดังนั้น
$$\begin{aligned} r_n &= u^2 - (v+2)^2 - 4n \\ &= u^2 - (v^2 + 4v + 4) - 4n \\ &= u^2 - v^2 - 4n - (4v + 4) \\ &= r - (4v + 4) \end{aligned}$$

ดังนั้นหากผลลัพธ์ของ $r > 0$ จำเป็นต้องปรับ r และ v ใหม่ดังนี้

$$r = r - (4v + 4) \tag{7.16}$$

$$v = v + 2 \tag{7.17}$$

กรณีที่ 2: $r < 0$

ความหมายคือค่า u ปัจจุบันมีค่าน้อยเกินความเป็นจริงจึงจำเป็นต้องเพิ่มค่า u ขึ้น อย่างไรก็ตามเนื่องจาก p และ q เป็นจำนวนเฉพาะที่เป็นจำนวนเต็มบวกคี่เสมอ ส่งผลให้ผลลัพธ์ของ $u = p + q$ เป็นจำนวนเต็มบวกคู่ ดังนั้นการเพิ่มค่าของ u จึงสามารถเพิ่มขึ้นครั้งละ 2 ค่าได้เพื่อปรับให้ u เป็นจำนวนเต็มบวกคู่ โดยหลังจากปรับค่า u ใหม่แล้วจึงต้องปรับค่า r ด้วยพิจารณาได้ดังนี้

กำหนดให้, $r_n = u_n^2 - v^2 - 4n$

เมื่อ u_n คือค่า u ที่ปรับใหม่ซึ่งมีค่าเป็น $u_n = u + 2$

r_n คือค่า r ที่ปรับใหม่

ดังนั้น

$$\begin{aligned} r_n &= (u + 2)^2 - v^2 - 4n \\ &= (u^2 + 4u + 4) - v^2 - 4n \\ &= u^2 - v^2 - 4n + (4u + 4) \\ &= r + (4u + 4) \end{aligned}$$

ดังนั้นหากผลลัพธ์ของ $r < 0$ จำเป็นต้องปรับ r และ u ใหม่ดังนี้

$$r = r + (4u + 4) \tag{7.18}$$

$$u = u + 2 \tag{7.19}$$

จากทั้งสองเงื่อนไขข้างต้นจึงได้ขั้นตอนวิธีวิธีของแฟร์มาต์แบบไม่มีการคำนวณหาค่ารากที่สอง [16] เป็นดังนี้

ขั้นตอนวิธีที่ 7.7 วิธีของแฟร์มาต์แบบไม่มีการคำนวณค่ารากที่สอง

```
     INPUT: n
     OUTPUT: p, q
1:   u ← 2⌈√n⌉
2:   v ← 0
3:   r ← u² – v² – 4n
4:   While (r != 0) do
5:       IF (r > 0) then
6:           r ← r – (4v + 4)
7:           v ← v + 2
8:       Else
9:           r ← r + (4u + 4)
10:          u ← u + 2
11:      End IF
12:  End While
13:  p ← (u+v)/2
14:  q ← (u-v)/2
```

ถึงแม้ว่าวิธีของแฟร์มาต์แบบไม่มีการคำนวณค่ารากที่สองจะไม่มีการคำนวนหาค่ารากที่สอง แต่รอบการคำนวณเพิ่มมากขึ้นเนื่องมาจากการคำนวณจะถูกแบ่งออกเป็น 2 ส่วนคือรอบการคำนวณในส่วนของ u ซึ่งมีค่าเท่ากับรอบการคำนวณของวิธีของแฟร์มาต์แบบคำนวณหาค่ารากที่สอง และรอบการคำนวณของ v ดังนี้

$$t_u = \frac{p+q}{2} - \lceil \sqrt{n} \rceil \tag{7.20}$$

$$t_v = \frac{p-q}{2} \tag{7.21}$$

เมื่อ t_u คือรอบการคำนวณของ u

t_v คือรอบการคำนวณของ v

ดังนั้นรอบการคำนวณทั้งหมดของวิธีแฟร์มาต์แบบไม่มีการคำนวณค่ารากที่สองคือ $t_v + t_u$

7. การโจมตีของไวเนอร์ (Wiener's attack)

การโจมตีของไวเนอร์ (Wiener's attack) [27] คือขั้นตอนวิธีหนึ่งที่สามารถใช้ในการคำนวณหา d ได้ โดยจะมีความแตกต่างจากขั้นตอนวิธีที่ถูกกล่าวก่อนหน้าทั้งหมด เนื่องจากไม่

จำเป็นต้องมีการแยกตัวประกอบแต่อย่างไร โดยจะใช้เศษส่วนต่อเนื่องเป็นเครื่องมือในการค้นหาค่า d กำหนดให้อุปกรณ์สื่อสารของผู้รับมีประสิทธิภาพต่ำ ดังนั้นแนวทางการแก้ปัญหาวิธีหนึ่งคือผู้รับเลือกใช้ d ที่มีขนาดเล็กซึ่งจะช่วยลดเวลาการคำนวณลงได้เป็นอย่างมาก อย่างไรก็ตามในช่วงปี ค.ศ. 1990 มิชาเอล ไวเนอร์ (Michael J. Wiener) ได้พิสูจน์ให้เห็นว่าหาก d มีค่าน้อยกว่า $\frac{1}{3}n^{\frac{1}{4}}$ แล้วจะสามารถคำนวณหาค่าดังกล่าวนี้กลับมาได้อย่างรวดเร็วโดยใช้ตัวเบนเข้าของเศษส่วนต่อเนื่องเรียกหลักการคำนวณหา d โดยใช้วิธีดังกล่าวนี้ว่าการโจมตีของไวเนอร์

ทฤษฎีบทที่ 7.2 กำหนดให้ $a, b, c, d \in \mathbb{Z}$ โดยที่ $1 \leq d < b$ และ $\gcd(a, b) = \gcd(c, d) = 1$ กล่าวได้ว่า $\frac{a}{b}$ คือตัวเบนเข้าของ $\frac{u}{v}$ ก็ต่อเมื่อ

$$\left|\frac{a}{b} - \frac{c}{d}\right| < \frac{1}{2d^2}$$

หลักการพื้นฐานของการโจมตีของไวเนอร์ คือ สมมติ n และ $\Phi(n)$ มีค่าใกล้เคียงกันเป็นอย่างมากแล้ว $\frac{e}{n}$ และ $\frac{e}{\Phi(n)}$ จะมีตัวเบนเข้าเป็นค่าเดียวกัน ดังนั้นหากผู้ไม่ประสงค์ดีสามารถคำนวณหาตัวเบนเข้าของ $\frac{e}{n}$ ได้จะสามารถคำนวณหา $\Phi(n)$ ได้เช่นเดียวกัน สำหรับตัวเบนเข้าของ $\frac{e}{n}$ สามารถคำนวณหาได้โดยพิจารณาจากทฤษฎีที่ 7.3 ซึ่งเป็นทฤษฎีของไวเนอร์

ทฤษฎีบทที่ 7.3 กำหนดให้ e, d คือกุญแจสาธารณะ และกุญแจส่วนตัวของ $n = pq$ โดยที่ $p < q < 2p$ และ $d < \frac{1}{3}n^{\frac{1}{4}}$ แล้ว $\frac{k}{d}$ คือตัวเบนเข้าของ $\frac{e}{n}$ เมื่อ $k \in \mathbb{Z}$ ที่คำนวณได้จาก $ed = k\Phi(n) + 1$

พิสูจน์ เนื่องจาก

$$\left|\frac{e}{n} - \frac{k}{d}\right| = \left|\frac{ed - kn}{dn}\right|$$

$$= \left|\frac{1 + k\Phi(n) - kn}{dn}\right|$$

$$= \left| \frac{1 + k(\Phi(n) - n)}{dn} \right|$$

และจาก $\Phi(n) = (p - 1)(q - 1) = pq - p - q + 1$

ได้ว่า $|n - \Phi(n)| = |n - pq + p + q - 1| = p + q - 1$

เนื่องจาก $n > \Phi(n)$ ดังนั้น $|n - \Phi(n)| > 0$

และเนื่องจาก $2p > q$, ดังนั้น

$$p + q - 1 < p + 2p - 1 = 3p - 1 < 3p$$

อย่างไรก็ตามเนื่องจาก $p < \sqrt{n}$ เสมอ ดังนั้นสรุปได้ว่า

$$|n - \Phi(n)| < 3\sqrt{n}$$

จึงได้ว่า

$$\left| \frac{e}{n} - \frac{k}{d} \right| < \left| \frac{1 + 3k\sqrt{n}}{dn} \right|$$

$$\leq \left| \frac{3k\sqrt{n}}{dn} \right|$$

เนื่องจาก $k, d, n > 0$ เสมอ

$$\left| \frac{e}{n} - \frac{k}{d} \right| \leq \frac{3k\sqrt{n}}{dn}$$

เนื่องจาก $d < \frac{1}{3} n^{\frac{1}{4}}$, หรือ $3d < n^{\frac{1}{4}}$ และ $k < d$ ดังนั้น $3k < 3d$

ได้ว่า

$$\frac{3k\sqrt{n}}{dn} < \frac{3d\sqrt{n}}{dn}$$

$$< \frac{n^{\frac{1}{4}}\sqrt{n}}{dn}$$

$$= \frac{1}{dn^{\frac{1}{4}}}$$

และเนื่องจาก

$$\frac{1}{dn^{\frac{1}{4}}} < \frac{1}{d(3d)} = \frac{1}{3d^2}$$

จึงได้ว่า $$\left|\frac{e}{n}-\frac{k}{d}\right| < \frac{1}{3d^2}$$

และจากทฤษฎีบทที่ 7.2 กล่าวได้ว่า $\frac{k}{d}$ คือตัวเบนเข้าของ $\frac{e}{n}$ □

เนื่องจาก

$$(x-p)(x-q) = x^2-(p+q)x+pq$$
$$= x^2-(p+q)x+n$$

และ $$n - \Phi(n) + 1 = n-(n-p-q+1)+1$$
$$= p+q$$

ดังนั้น $$(x-p)(x-q) = x^2-(n-\Phi(n)+1)x+n \qquad (7.22)$$

กำหนดให้ตัวเบนเข้าของ $\frac{e}{n}$ ที่เรียงจากการพิจารณาจากจำนวนสมาชิกน้อยสุดไปยังจำนวนสมาชิกสูงสุด คือ $\frac{k_0}{d_0},\frac{k_1}{d_1},\frac{k_2}{d_2},\cdots,\frac{k_r}{d_r}$ และ $M_i = \frac{ed_i - 1}{k_i}$ เมื่อ $i = 0, 1, 2, \cdots, r$ โดยการคำนวณเริ่มจากการคำนวณที่ $i = 0$ และหากแทนค่า $\Phi(n) = M_i$ แล้วผลลัพธ์ที่ได้จากสมการที่ (7.22) เป็นจำนวนเต็มจะสรุปได้ว่าผลลัพธ์ที่คำนวณได้คือตัวประกอบของ n ในทางกลับกันหากผลลัพธ์ที่ได้ไม่ใช่จำนวนเต็ม จำเป็นต้องพิจารณา i ที่ตำแหน่งถัดไป ($i = i + 1$) เพื่อคำนวณหา M_i ใหม่อีกครั้งจนกระทั่งพบผลลัพธ์จากสมการ (7.22) ที่เป็นจำนวนเต็ม

ตัวอย่างที่ 7.16 กำหนดให้ $n = 24797$ และ $e = 20983$ จงหา d, p และ q โดยใช้ขั้นตอนวิธีการโจมตีของไวเนอร์

วิธีทำ เนื่องจากเศษส่วนต่อเนื่องของ $\frac{20983}{24797}$ มีค่าเป็น

$$0+\cfrac{1}{1+\cfrac{1}{5+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{158+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2}}}}}}}}$$

หรือ $\frac{20983}{24797} = [0; 1, 5, 1, 1, 158, 2, 2, 2]$

ดังนั้นการทดสอบในแต่ละรอบเป็นดังนี้

รอบที่ 1: พิจารณาตัวเบนเข้าเป็น [0; 1] ซึ่งผลลัพธ์ที่ได้คือ 1 จึงได้ว่า $k = 1$ และ $d = 1$ ซึ่งโดยปกติ d จะต้องมีค่ามากกว่า 1 เสมอจึงไม่ใช่ค่าที่ถูกต้อง

รอบที่ 2: พิจารณาตัวเบนเข้าเป็น [0; 1, 5] ซึ่งผลลัพธ์ที่ได้คือ

$$0+\frac{1}{1+\frac{1}{5}} = 0+\frac{1}{\frac{6}{5}}$$

$$= \frac{5}{6}$$

จึงได้ว่า $k = 5$ และ $d = 6$ ซึ่งโดยปกติ d จะต้องเป็นจำนวนเต็มคี่เสมอจึงไม่ใช่ค่าที่ถูกต้อง

รอบที่ 3: พิจารณาตัวเบนเข้าเป็น [0; 1, 5, 1] ซึ่งผลลัพธ์ที่ได้คือ

$$0+\cfrac{1}{1+\cfrac{1}{5+\cfrac{1}{1}}} = 0+\frac{1}{1+\frac{1}{6}}$$

$$= 0+\frac{1}{\frac{7}{6}}$$

$$= \frac{6}{7}$$

$$M_i = \frac{20983x7 - 1}{6} = 24480$$

จากสมการ (7.22)

$$x^2 - (24797 - 24480 + 1)x + 24797 = x^2 - 318x + 24797$$

แก้สมการหาค่า $x = \frac{-(-318) \pm \sqrt{(-318)^2 - 4(1)(24797)}}{2(1)}$

$$= \frac{318 \pm \sqrt{1936}}{2}$$

$$= \frac{318 \pm 44}{2}$$

ได้ว่า $x = \frac{318 + 44}{2} = 181$ และ $x = \frac{318 - 44}{2} = 137$

ดังนั้น $d = 7$, $p = 181$ และ $q = 137$

จากทฤษฎีบทที่ 7.3 แสดงให้เห็นว่าการโจมตีของไวเนอร์มีประสิทธิภาพที่สูงมากในกรณีที่ d มีขนาดเล็ก ดังนั้นการเลือกใช้งานจริง d ควรมีขนาดใหญ่ ($d > \frac{1}{3}n^{\frac{1}{4}}$) เพื่อหลีกเลี่ยงการโจมตีของไวเนอร์ นอกเหนือจากนั้นในปี ค.ศ. 1999 ดัน บอเนอช์ (Dan Boneh) และ จเล็นน์ เดอร์ฟรี (Glenn Durfee) [50] แสดงให้เห็นว่าถึงแม้ d มีขนาดที่สูงกว่า $n^{\frac{1}{4}}$ (หรือ $n^{0.25}$) แต่หากมีขนาดเล็กกว่า $n^{0.292}$ ค่าดังกล่าวยังคงถูกคำนวณกลับได้ไม่ยาก ดังนั้นเพื่อเพิ่มความปลอดภัยวิทยาการรหัสลับอาร์เอสเอ d ต้องมีขนาดใหญ่กว่า $n^{0.292}$

8. บทสรุปสาระสำคัญ

วิทยาการรหัสลับอาร์เอเอสเป็นวิทยาการรหัสลับแบบกุญแจสาธารณะที่ได้รับความนิยมสูงมากในปัจจุบัน โดยทั่วไปเพื่อเพิ่มความแข็งแกร่งให้กับวิทยาการรหัสลับอาร์เอเอสค่ากุญแจส่วนตัวควรมีขนาดที่ใหญ่มหาศาลจึงส่งผลให้การประมวลผลการถอดรหัสลับซึ่งใช้ค่ากุญแจส่วนตัวเป็นเลขยกกำลังใช้เวลาสูงมาก จากปัญหาข้างต้นการเพิ่มความเร็วแก่กระบวนการถอดรหัสลับจึงเป็นเรื่องจำเป็นและควรให้ความสำคัญ โดยงานวิจัยที่เกี่ยวกับการลดเวลากระบวนการถอดรหัสลับสำหรับ

วิทยาการรหัสลับอาร์เอสเอถูกนำเสนอเป็นจำนวนมาก ทฤษฎีเศษเหลือจีนเป็นอีกวิธีหนึ่งที่ได้รับความนิยมเป็นอย่างมากที่สามารถนำไปประยุกต์ใช้สำหรับการลดเวลาการประมวลผลโดยจะแบ่งค่ากุญแจส่วนตัวออกเป็นหลายส่วน โดยที่แต่ละส่วนจะมีขนาดเล็กลงเป็นอย่างมาก จึงส่งผลให้การถอดรหัสลับโดยใช้กุญแจส่วนตัวที่ถูกแบ่งย่อยเป็นค่าเล็กๆ เป็นเลขยกกำลังใช้ทรัพยากรสำหรับการประมวลผลลดลงเช่นกัน นอกเหนือจากนั้นผู้เขียนได้นำเสนอขั้นตอนวิธีใหม่ที่ถูกนำมาใช้สำหรับการลดเวลากระบวนการถอดรหัสลับที่ใช้กุญแจส่วนตัวเป็นเลขยกกำลังซึ่งจากสมการที่นำเสนอใหม่พบว่ามีความเหมาะสมที่จะนำมาใช้ร่วมกับค่ากุญแจส่วนตัวที่มีขนาดใหญ่มหาศาลโดยเฉพาะอย่างยิ่งเมื่อกุญแจส่วนตัวมีค่าใกล้เคียงกับค่าออยเลอร์ ในทางกลับกันขั้นตอนวิธีใหม่ที่นำเสนอมีประสิทธิภาพต่ำมากในกรณีที่กุญแจส่วนตัวมีขนาดเล็ก ดังนั้นหากค่ากุญแจดังกล่าวมีขนาดเล็กผู้เขียนได้เสนอให้ใช้สมการดั้งเดิมซึ่งมีประสิทธิภาพที่สูงกว่ามาก อย่างไรก็ตามค่ากุญแจส่วนตัวที่ถูกนำมาใช้จริงมักจะมีขนาดที่ใหญ่มหาศาลเพื่อหลีกเลี่ยงการโจมตีโดยขั้นตอนวิธีที่ถูกเสนอโดยวีเนอร์ ดังนั้นขั้นตอนวิธีที่ผู้เขียนได้นำเสนอจึงมีประโยชน์เป็นอย่างมาก นอกเหนือจากนั้นยังสามารถนำขั้นตอนวิธีดังกล่าวไปประยุกต์ใช้งานร่วมกับทฤษฎีเศษเหลือจีนเพื่อลดเวลาลงได้อีก

ความปลอดภัยของวิทยาการรหัสลับอาร์เอเอสขึ้นอยู่กับความยากของการแยกตัวประกอบค่ามอดูลัสซึ่งหากผู้ไม่ประสงค์ดีสามารถแยกตัวประกอบได้จะสามารถคำนวณกุญแจส่วนตัวกลับคืนได้ ดังนั้นเพื่อเพิ่มความปลอดภัยจึงได้มีการนำเสนอค่ามอดูลัสซึ่งเกิดจากการคูณกันของจำนวนเฉพาะมากกว่า 2 ค่า แต่อย่างไรก็ตามความปลอดภัยที่เพิ่มมากขึ้นกับส่งผลให้เวลาที่ใช้สำหรับการประมวลผลเพิ่มมากขึ้นตามไปด้วย

ถึงแม้ว่าในปัจจุบันขั้นตอนวิธีที่ถูกนำมาใช้สำหรับแยกตัวประกอบได้ถูกนำเสนอออกมาเป็นจำนวนมาก แต่ยังไม่พบว่ามีขั้นตอนวิธีใดที่สามารถแยกตัวประกอบค่ามอดูลัสที่มีขนาดอย่างน้อย 1024 บิตและ ตัวประกอบทั้งหมดมีความแข็งแกร่งได้ภายในระยะเวลาอันสั้น จึงเป็นเหตุผลที่วิทยาการรหัสลับอาร์เอสเอยังคงได้รับความนิยมสูงมากในปัจจุบัน แต่อย่างไรก็ตามขั้นตอนวิธีการแยกตัวประกอบยังคงถูกพัฒนาอย่างต่อเนื่อง และมีประสิทธิภาพที่สูงขึ้น ดังนั้นในอนาคตวิทยาการรหัสลับอาร์เอสอาจถูกโจมตีได้หากขั้นตอนวิธีการแยกตัวประกอบประสิทธิภาพสูงได้ถูกกำเนิดขึ้น

แบบฝึกหัดท้ายบท
บทที่ 7

1. วิทยาการรหัสลับอาร์เอสเอถูกนำเสนอโดยใคร
2. เพื่อความปลอดภัยค่ามอดูลัสสำหรับวิทยาการรหัสลับอาร์เอสเอควรมีขนาดอย่างน้อยสุดเท่าไร
3. กำหนดให้ $p = 149$ และ $q = 229$ จงคำนวณหา n และ $\Phi(n)$
4. จากคำถามข้อ 3 สมมติเลือก $e = 3$ ได้หรือไม่เพราะเหตุใด
5. จากคำถามข้อ 3 จงคำนวณหา d หากเลือก $e = 23$
6. จากคำถามข้อ 3 สมมติตัวอักษรที่จะใช้สำหรับการสื่อสารมีทั้งหมด 6 ตัว จงหาจำนวนตัวอักษรสูงสุดที่สามารถนำมาเข้ารหัสลับได้ใน 1 ครั้ง
7. กำหนดให้ตัวอักษรทั้งหมด และค่าประจำตำแหน่งเป็นดังตารางต่อไปนี้

A	B	C	D	E	F
0	1	2	3	4	5

จงแปลงข้อความ “FACE” เป็นตัวเลขฐานสิบเพื่อใช้สำหรับการเข้ารหัสลับ

8. จงเข้ารหัสข้อความ “FACE” โดยใช้ n, $\Phi(n)$ และ e จากคำถามที่ 5 (คำตอบที่เป็นตัวเลข)
9. จากผลลัพธ์ข้อ 8 จงหาข้อความไซเฟอร์ก่อนที่จะถูกส่งไปยังผู้รับ (คำตอบเป็นตัวอักษร)
10. จาก d ที่ได้จากคำถามข้อ 5 จงหาเลขยกกำลังใหม่ทั้งสองค่าที่จำเป็นต้องใช้สำหรับทฤษฎีเศษเหลือจีน
11. กำหนดให้ $p_1 = 13$, $p_2 = 23$, $p_3 = 43$, $e = 19$ และ $n = p_1p_2p_3$ จงหา d
12. กำหนดให้ $\Phi(n) = 21672$ ($n = pq$) และ $d = 21001$ จงเปรียบเทียบประสิทธิภาพของการถอดรหัสลับระหว่างสมการถอดรหัสแบบดั้งเดิมและสมการถอดรหัสลับจากทฤษฎีบทที่ 7.1
13. จงแยกตัวประกอบ $n = 8129$ โดยใช้ขั้นตอนวิธีทดลองหารทั้งแบบปรับค่าขึ้น และแบบปรับค่าลงพร้อมทั้งอภิปรายผลว่าขั้นตอนวิธีใดมีประสิทธิภาพที่สูงกว่าเพราะเหตุใด
14. กำหนดให้ $n = 34909860103$ และ $e = 25388702051$ จงคำนวนหา d โดยใช้ขั้นตอนวิธีการโจมตีของไวเนอร์
15. กำหนดให้ $n = 9907577$ และ $d = 401399$ การใช้วิธีการโจมตีของไวเนอร์เหมาะสมสำหรับคำนวณหาค่า d ดังกล่าวนี้หรือไม่เพราะเหตุใด

16. กำหนดให้ n = 53483002077180661257 คือผลคูณของ p = 17827667359060220419 และ q = 3 (โดยค่า p และ q ในสถานการณ์จริงยังไม่ถูกเปิดเผย) แล้วหากเลือกขั้นตอนวิธีการแยกตัวประกอบ 3 วิธีประกอบด้วยขั้นตอนวิธีการทดลองหาร ขั้นตอนวิธี $p - 1$ ของโพลลาร์ด และขั้นตอนวิธีวีแฟกเตอร์ จงตอบคำถามว่าขั้นตอนวิธีใดมีประสิทธิภาพสูงสุดสำหรับแก้ปัญหาในข้อนี้

บทที่ 8
การปรับปรุงขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์

จากที่ได้กล่าวในบทที่แล้วว่าความปลอดภัยของวิทยาการรหัสลับอาร์เอสเอขึ้นอยู่กับความยากของการแยกตัวประกอบค่ามอดูลัส ซึ่งหากสามารถแยกตัวประกอบสำเร็จจะส่งผลให้ทราบค่าตัวประกอบทั้งสองค่าและสามารถคำนวณหากุญแจส่วนตัวคืนได้ โดยหากเลือกใช้ค่ามอดูลัสที่มีขนาดใหญ่ และมีตัวประกอบที่แข็งแกร่งที่ไม่สนับสนุนต่อการแยกตัวประกอบโดยใช้ขั้นตอนวิธีใดวิธีหนึ่งแล้วพบว่ายังไม่มีขั้นตอนวิธีใดที่สามารถแยกตัวประกอบค่ามอดูลัสได้ภายในระยะเวลาสั้น จึงส่งผลให้วิทยาการรหัสลับอาร์เอสเอยังคงมีความปลอดภัยและถูกใช้งานอยู่ในปัจจุบัน แต่อย่างไรก็ตามหากเลือกใช้จำนวนเฉพาะไม่เหมาะสม อาจส่งผลให้ขั้นตอนวิธีการแยกตัวประกอบบางชนิดสามารถแยกตัวประกอบได้อย่างรวดเร็ว โดยขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์มีประสิทธิภาพสูงมากกรณีที่ตัวประกอบทั้งสองค่ามีขนาดใกล้เคียงกัน ในบทนี้จะกล่าวถึงการปรับปรุงขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์ซึ่งมีวิธีการที่ถูกนำเสนอออกมาเป็นจำนวนมาก และส่งผลให้เวลาที่ใช้ในการแยกตัวประกอบลดลงเป็นอย่างมาก ดังนั้นขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์จึงมีประสิทธิภาพที่สูงขึ้นและสามารถใช้แยกตัวประกอบค่ามอดูลัสในระยะเวลาลดลง ถึงแม้ว่าจำนวนเฉพาะทั้งสองมีค่าห่างกันมากยิ่งขึ้น ดังนั้นประโยชน์ของการศึกษาบทเรียนนี้คือหากผู้อ่านมีความพึงประสงค์ที่จะเลือกใช้งานวิทยาการรหัสลับอาร์เอสเอจำเป็นต้องระมัดระวังการเลือกใช้งานจำนวนเฉพาะที่จะใช้งานเพื่อเป็นตัวประกอบของค่ามอดูลัสมากขึ้นเพื่อหลีกเลี่ยงความเสี่ยงจากการโจมตีโดยผู้ไม่ประสงค์ดี

1. การพิจารณาเลขหลักหน่วยของกำลังสองสมบูรณ์

เนื่องจากค่ากำลังสองสมบูรณ์ของจำนวนเต็มที่มีเลขหลักหน่วยเป็น 0, 1, 2, 3, 4, 5, 6, 7, 8 และ 9 จะได้ผลลัพธ์ที่มีเลขหลักหน่วยเป็น 0, 1, 4, 9, 6, 5, 6, 9, 4 และ 1 ตามลำดับ [19], [20] ซึ่งสังเกตได้ว่ากำลังสองสมบูรณ์ทุกค่าจะมีเลขหลักหน่วยเป็น 0, 1, 4, 5, 6 และ 9 เพียงเท่านั้น หรือกล่าวอีกนัยหนึ่งคือเป็นไปไม่ได้ที่เลขหลักหน่วยของกำลังสองสมบูรณ์จะมีค่าเป็น 2, 3, 7 หรือ 8

ดังนั้นหากนำหลักการดังกล่าวนี้มาประยุกต์ใช้งานร่วมกับขั้นตอนวิธีที่ 7.6 คือหากเลขหลักหน่วยของ $x^2 - n$ มีค่าเป็น 2, 3, 7 หรือ 8 จะไม่จำเป็นต้องเสียเวลาในการคำนวณหาค่ารากที่สองของ y^2 เนื่องจากผลลัพธ์ที่ได้ไม่เป็นจำนวนเต็มอย่างแน่นอน

2. การพิจารณาเศษจากการหารเลขกำลังสองสมบูรณ์ด้วย 20

การพิจารณาเศษที่ได้จากการหารเลขกำลังสองสมบูรณ์ด้วย 20 จะสามารถลดจำนวนรอบการคำนวณได้มากยิ่งขึ้นหากเปรียบเทียบกับการพิจารณาเลขหลักหน่วยของกำลังสองสมบูรณ์ เนื่องจากการพิจารณาเลขหลักหน่วยสามารถทำได้โดยพิจารณาเศษที่ได้จากการหารค่าดังกล่าวด้วย 10 ซึ่งเศษที่เป็นไปได้ทั้งหมดคือ 0 – 9 อย่างไรก็ตามหากพิจารณาเศษที่ได้จากการหารจำนวนเต็มใดๆ ด้วย 20 พบว่าผลลัพธ์ที่เป็นไปได้ทั้งหมดคือ 0 – 19 ซึ่งสังเกตได้ว่าขอบเขตของการพิจารณาจะกว้างกว่าการพิจารณาเลขหลักหน่วย ดังนั้นจึงสามารถตัดผลลัพธ์ของ $x^2 - n$ ที่หากคำนวณหาค่ารากที่สองแล้วไม่ใช่จำนวนเต็มออกได้มากยิ่งขึ้น โดยเศษที่ได้จากการหารกำลังสองสมบูรณ์ด้วย 20 จะต้องมีค่าเป็น 0, 1, 4, 5, 9 หรือ 16 เสมอ [21] ดังนั้นหากผลลัพธ์ที่ได้จากการหาร $x^2 - n$ ด้วย 20 มีค่าเป็น 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18 หรือ 19 จะไม่จำเป็นต้องคำนวณหาค่ารากที่สอง เนื่องจากผลลัพธ์ที่ได้ไม่เป็นจำนวนเต็มอย่างแน่นอน

3. การหารูปแบบของ x โดยพิจารณาจากเศษจากการหารเลขกำลังสองสมบูรณ์ด้วย 20

เนื่องจากปัญหาของการพิจารณาเศษที่ได้จากการหารเลขกำลังสองสมบูรณ์ด้วย 20 คือต้องเสียเวลาในการคำนวณหาผลลัพธ์ของ $(x^2 - n) \bmod 20$ เสมอก่อนที่จะพิจารณาคำนวณหาค่ารากที่สอง ดังนั้นจึงมีการเสนอแนวคิดใหม่ซึ่งสามารถคาดการณ์หาผลลัพธ์ของ $(x^2 - n) \bmod 20$ ได้ล่วงหน้าโดยไม่จำเป็นต้องมีการคำนวณใดๆ [21] ซึ่งพิจารณาได้จากตารางที่ 8.1

เนื่องจาก n เป็นจำนวนเต็มบวกคี่ที่เลขหลักหน่วยมีค่าไม่เท่ากับ 5 (เนื่องจากหากเลขหลักหน่วยเป็น 5 จะทราบทันทีว่า 5 เป็นหนึ่งในตัวประกอบของ n) ดังนั้นเศษที่ได้จากการหาร n ด้วย 20 จะมีค่าเป็น 1, 3, 7, 9, 11, 13, 17 หรือ 19 เท่านั้น

กำหนดให้ LSG(z) คือสัญลักษณ์ที่แทนเลขหลักหน่วยของ z เมื่อ $z \in \mathbb{Z}^+$ ตารางที่ 8.1 แทนผลลัพธ์ที่เป็นไปได้ทั้งหมดของ $(x^2 - n) \bmod 20$ ที่พิจารณาจากความสัมพันธ์ระหว่าง LSG(x) และเศษที่ได้จากการหาร n ด้วย 20 โดยปราศจากการคำนวณซึ่งวิธีการอ่านค่าจากตารางเป็นดังนี้ ตำแหน่งแถวแทนเลขหลักหน่วยของ x และแนวคอลัมน์แทนเศษที่ได้จากการหาร n ด้วย 20 โดยผลลัพธ์ของ $(x^2 - n) \bmod 20$ จะอยู่ตำแหน่งแถวที่ตรงกับเลขหลักหน่วยของ x และตำแหน่งคอลัมน์ของเศษที่ได้จากการหาร n ด้วย 20

ตารางที่ 8.1 ผลลัพธ์ที่เป็นไปได้ทั้งหมดของ ($x^2 - n$) mod 20

LSG(x)	n mod 20							
	1	**3**	**7**	**9**	**11**	**13**	**17**	**19**
0	19	17	13	11	**9**	7	3	**1**
1	**0**	18	14	12	10	8	**4**	2
2	3	**1**	17	15	13	11	7	**5**
3	8	6	2	**0**	18	**16**	12	10
4	15	13	**9**	7	**5**	3	9	17
5	**4**	2	18	**16**	14	12	8	6
6	15	13	**9**	7	**5**	3	19	17
7	8	6	2	**0**	18	**16**	12	10
8	3	**1**	17	15	13	11	7	**5**
9	**0**	18	14	12	10	8	**4**	2

จากตารารงที่ 8.1 กำหนดให้แถวที่ 1 ถึง 10 แทน LSG(x) = 0 ถึง 9 ตามลำดับ และกำหนดให้ให้คอลัมน์ที่ 1 ถึง 8 แทนผลลัพธ์ของ n mod 20 = 1, 3, 7, 9, 11, 13, 17 และ 19 ตามลำดับ ตัวอย่างการพิจารณาและการพิสูจน์บางตำแหน่งเป็นดังนี้

1. พิจารณาแถวที่ 1, คอลัมน์ที่ 1: ความหมายคือ หากเลขหลักหน่วยของ x มีค่าเป็น 0 และเศษที่ได้จากการหาร n ด้วย 20 มีค่าเป็น 1 ผลลัพธ์ของ $x^2 - n$ mod 20 มีค่าเป็น 19 เสมอ

พิสูจน์

เนื่องจากเลขหลักหน่วยของ x มีค่าเป็น 0 ดังนั้นเศษที่ได้จากการหาร x^2 ด้วย 20 คือ 0 เสมอ

จาก $y^2 = x^2 - n$

ดังนั้น $y^2 \bmod 20 = (x^2 - n) \bmod 20$

$= (x^2 \bmod 20 - n \bmod 20) \bmod 20$

$= (0 - 1) \bmod 20$

$= -1 \bmod 20$

$= 19$

2. พิจารณาแถวที่ 7, คอลัมน์ที่ 8: ความหมายคือ หากเลขหลักหน่วยของ x มีค่าเป็น 6 และเศษที่ได้จากการหาร n ด้วย 20 มีค่าเป็น 19 ผลลัพธ์ของ $x^2 - n \bmod 20$ มีค่าเป็น 17 เสมอ

พิสูจน์

เนื่องจากเลขหลักหน่วยของ x มีค่าเป็น 6 ดังนั้นเศษที่ได้จากการหาร x^2 ด้วย 20 มีค่าเป็น 16 เสมอ

ดังนั้น $y^2 \bmod 20 = (16 - 19) \bmod 20$

$= -3 \bmod 20$

$= 17$

โดยการพิสูจน์กรณีอื่นๆ สามารถดำเนินการได้โดยใช้วิธีเดียวกันนี้ ซึ่งผลลัพธ์ที่ได้จะเป็นดังเช่นตารางที่ 8.1 เสมอ

4. การพิจารณาเศษจากการหารมอดูลัสด้วย 4

เนื่องจากตัวประกอบทั้งสองค่าของ n เป็นจำนวนเต็มบวกคี่เสมอ ดังนั้นหากนำตัวประกอบแต่ละค่าหารด้วย 4 พบว่าเศษที่ได้จากการหารมีค่าเป็น 1 หรือ 3 เท่านั้น [18], [32] ดังนั้นการพิจารณาหารูปแบบของการหาร n ด้วย 4 ถูกแบ่งเป็น 4 กรณีดังนี้

กรณีที่ 1: $p \bmod 4 = 1$ ($p = 4k + 1$) และ $q \bmod 4 = 1$ ($q = 4l + 1$)

จากสมการ, $n = pq$

แทนค่า, $= (4k + 1)(4l + 1)$

$= 4(4lk) + 4l + 4k + 1$

$= 4(4lk + l + k) + 1,$ ($n \bmod 4 = 1$)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(4k+1)+(4l+1)}{2}$

$= \frac{4k+4l+2}{2}$

$= 2(k + l) + 1,$ (จำนวนเต็มบวกคี่)

กรณีที่ 2: $p \bmod 4 = 3$ ($p = 4k + 3$) และ $q \bmod 4 = 3$ ($q = 4l + 3$)

จากสมการ, $n = pq$

แทนค่า, $= (4k + 3)(4l + 3)$

$= 4(4lk) + 4(3l) + 4(3k) + 9$

$= 4(4lk) + 4(3l) + 4(3k) + 4(2) + 1$

$= 4(4lk + 3l + 3k + 2) + 1,$ (n mod 4 = 1)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(4k+3)+(4l+3)}{2}$

$= \frac{4k+4l+6}{2}$

$= 2(k + l + 1) + 1,$ (จำนวนเต็มบวกคี่)

กรณีที่ 3: p mod 4 = 1 ($p = 4k + 1$) และ q mod 4 = 3 ($q = 4l + 3$)

จากสมการ, $n = pq$

แทนค่า, $= (4k + 1)(4l + 3)$

$= 4(4lk) + 4l + 4(3k) + 3$

$= 4(4lk + l + 3k) + 3,$ (n mod 4 = 3)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(4k+1)+(4l+3)}{2}$

$= \frac{4k+4l+4}{2}$

$= 2(k + l + 1),$ (จำนวนเต็มบวกคู่)

กรณีที่ 4: p mod 4 = 3 ($p = 4k + 3$) และ q mod 4 = 1 ($q = 4l + 1$)

จากกรณีที่ 4 สังเกตได้ว่ารูปแบบของ x และ n จะเหมือนกับผลลัพธ์ดังกรณีที่ 3 เนื่องจากรูปแบบของ p และ q มีลักษณะเหมือนกันเพียงแต่มีการสลับรูปแบบกันเท่านั้น

หากนำทั้ง 4 กรณีไปประยุกต์ใช้กับขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์จะมีขั้นตอนการดำเนินการดังนี้ เริ่มจากหาเศษที่ได้จากการหาร n ด้วย 4 ซึ่งผลลัพธ์จะแบ่งออกเป็น 2 แบบคือเศษมีค่าเป็น 1 (ตรงกับกรณีที่ 1 และ 2) จะสังเกตได้ว่า x จะเป็นจำนวนเต็มบวกคี่เสมอ ในทางกลับกันหากเศษที่ได้มีค่าเป็น 3 (ตรงกับกรณีที่ 3 และ 4) จะสังเกตได้ว่า x จะเป็นจำนวนเต็ม

บวกคู่เสมอ ดังนั้นการเพิ่มค่า x สามารถเพิ่มได้ครั้งละ 2 ค่า (ปกติจากขั้นตอนวิธีที่ 7.6 ค่าของ x จะเพิ่มได้เพียงครั้งละ 1 ค่าเท่านั้น)

นอกเหนือจากนั้นในขั้นตอนการพิจารณาหาเศษที่ได้จากการหาร n ด้วย 4 ไม่จำเป็นต้องนำค่า n ทั้งหมดมาทำการทดสอบ แต่สามารถนำเพียงเลข 2 หลักสุดท้ายของ n มาหารด้วย 4 ซึ่งเศษที่ได้จะมีค่าเท่ากับการใช้ตัวเลขทุกหลักของ n ดังนี้ กำหนดให้ $n = n_m n_{m-1}\, n_{m-2} \cdots n_0$ เมื่อ n_i แทนตัวเลขในแต่ละตำแหน่งของ n และ $i \in \mathbb{Z}^+$ ได้ว่า

จาก, $n = (n_m \times 10^m) + (n_{m-1} \times 10^{m-1}) + \cdots + (n_2 \times 10^2) + (n_1 \times 10) + n_0$

ดังนั้น,

$n \bmod 4 = ((n_m \times 10^m) + (n_{m-1} \times 10^{m-1}) + \cdots + (n_2 \times 10^2) + (n_1 \times 10) + n_0) \bmod 4$

เนื่องจาก $10^i \bmod 4 = 0$ เมื่อ $i > 1$ ดังนั้น

$n \bmod 4 = ((n_1 \times 10) + n_0) \bmod 4$

หรือ $n \bmod 4 = (n_1 n_0)_{10} \bmod 4$

ตัวอย่างที่ 8.1 จงหาเศษที่ได้จากการหาร 1987326719135714632157 ด้วย 4

วิธีทำ เนื่องจาก 1987326719135714632157 mod 4 = 57 mod 4

โดย 57 mod 4 = 1

ดังนั้น 1987326719135714632157 mod 4 = 1

5. การพิจารณาเศษจากการหารมอดูลัสด้วย 6

การพิจารณาเศษจากการหาร n ด้วย 6 [22] จะมีแนวคิดที่คล้ายคลึงกับการพิจารณาเศษจากการหาร n ด้วย 4 อย่างไรก็ตามการหารจำนวนเต็มด้วย 6 จะเกิดเศษที่เป็นไปได้ทั้งหมด 6 ค่าซึ่งแยกพิจารณาได้ดังนี้

กรณีที่ 1: $n \bmod 6 = 0$, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k = 2(3k)$ โดยพบว่า n จะเป็นจำนวนเต็มคู่เสมอ ซึ่งขัดแย้งกับความเป็นจริง ดังนั้นกรณีนี้จึงไม่เกิดขึ้น

กรณีที่ 2: $n \bmod 6 = 1$, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k + 1$

กรณีที่ 3: $n \bmod 6 = 2$, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k + 2 = 2(3k+1)$ โดยพบว่า n จะเป็นจำนวนเต็มคู่เสมอ ซึ่งขัดแย้งกับความเป็นจริง ดังนั้นกรณีนี้จึงไม่เกิดขึ้น

กรณีที่ 4: $n \bmod 6 = 3$, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k + 3 = 3(2k + 1)$ ซึ่งพบว่าตัวประกอบ 1 ค่าของ n จะต้องมีค่าเป็น 3 เสมอซึ่งในความเป็นจริงค่าดังกล่าวนี้

ไม่นิยมถูกนำมาใช้งานเป็นตัวประกอบของ n ดังนั้นหากกำหนดให้ตัวประกอบของ n มีค่าไม่เท่ากับ 3 กรณีนี้จะไม่เกิดขึ้นเช่นกัน

กรณีที่ 5: n mod 6 = 4, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k + 4 = 2(3k+2)$ โดยพบว่า n จะเป็นจำนวนเต็มคู่เสมอ ซึ่งขัดแย้งกับความเป็นจริง ดังนั้นกรณีนี้จึงไม่เกิดขึ้น

กรณีที่ 6: n mod 6 = 5, แสดงว่า n จะสามารถถูกเขียนให้อยู่ในรูปของ $n = 6k + 5$

จากทั้ง 6 กรณีสรุปได้ว่ากำหนดให้ 3 ไม่เป็นตัวประกอบของ n แล้วเศษที่ได้จากการหาร n ด้วย 6 จะมีเพียง 2 คำตอบเท่านั้นคือ 1 และ 5 จึงสามารถหารูปแบบของ n และ x ได้ 4 กรณีดังนี้

กรณีที่ 1: p mod 6 = 1 ($p = 6k + 1$) และ q mod 6 = 1 ($q = 6l + 1$)

จากสมการ, $n = pq$

แทนค่า, $= (6k + 1)(6l + 1)$

$= 6(6lk) + 6l + 6k + 1$

$= 6(6lk + l + k) + 1,$ (n mod 6 = 1)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(6k+1)+(6l+1)}{2}$

$= \frac{6k+6l+2}{2}$

$= 3(k + l) + 1,$ (จำนวนเต็มที่หาร 3 ไม่ลงตัว)

กรณีที่ 2: p mod 6 = 5 ($p = 6k + 5$) และ q mod 6 = 5 ($q = 6l + 5$)

จากสมการ, $n = pq$

แทนค่า, $= (6k + 5)(6l + 5)$

$= 6(6lk) + 6(5l) + 6(5k) + 25$

$= 6(6lk) + 6(5l) + 6(5k) + 6(4) + 1$

$= 6(6lk + 5l + 5k + 4) + 1,$ (n mod 6 = 1)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(6k+5)+(6l+5)}{2}$

$= \frac{6k+6l+10}{2}$

$= 3(k + l) + 5,$

$= 3(k + l + 1) + 2,$ (จำนวนเต็มที่หาร 3 ไม่ลงตัว)

กรณีที่ 3: $p \bmod 6 = 1$ ($p = 6k + 1$) และ $q \bmod 6 = 5$ ($q = 6l + 5$)

จากสมการ, $n = pq$

แทนค่า, $= (6k + 1)(6l + 5)$

$= 6(6lk) + 6l + 6(5k) + 5$

$= 6(6lk + l + 5k) + 5,$ ($n \bmod 6 = 5$)

จากสมการ, $x = \frac{p+q}{2}$

แทนค่า, $= \frac{(6k+1)+(6l+5)}{2}$

$= \frac{6k+6l+6}{2}$

$= 3(k + l + 1),$ (จำนวนเต็มที่หาร 3 ลงตัว)

กรณีที่ 4: $p \bmod 6 = 5$ ($p = 6k + 5$) และ $q \bmod 6 = 1$ ($q = 6l + 1$)

จากกรณีที่ 4 สังเกตได้ว่ารูปแบบของ x และ n จะเหมือนกับผลลัพธ์ดังกรณีที่ 3 เนื่องจากรูปแบบของ p และ q มีลักษณะเหมือนกันเพียงแต่มีการสลับรูปแบบกันเท่านั้น

หากนำทั้ง 4 กรณีไปประยุกต์ใช้กับขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์จะมีขั้นตอนการดำเนินการดังนี้ เริ่มจากหาเศษที่ได้จากการหาร n ด้วย 6 ซึ่งผลลัพธ์จะแบ่งออกเป็น 2 กรณีคือเศษมีค่าเป็น 1 (ตรงกับกรณีที่ 1 และ 2) จะสังเกตได้ว่า x จะเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัวเสมอ ในทางกลับกันหากเศษที่ได้มีค่าเป็น 5 (ตรงกับกรณีที่ 3 และ 4) จะสังเกตได้ว่า x จะเป็นจำนวนเต็มที่หาร 3 ลงตัวเสมอ ดังนั้นการเพิ่มค่า x สามารถเพิ่มค่าได้มากกว่า 1 ค่าสำหรับทุกๆ 1 รอบของการคำนวณ

6. การพิจารณาเศษจากการหารผลรวมระหว่างค่ามอดูลัสและ 1 ด้วย 8

จากงานวิจัย [31] พบว่าหากไม่เกิดเศษจากการหาร $n + 1$ ด้วย 8 แล้วจะได้ว่ารูปแบบของ u จะเป็นจำนวนเต็มบวกที่หาร 8 ลงตัวเสมอ ดังนั้นหากพบว่ารูปแบบของ n เป็นดังเงื่อนไขข้างต้นและเลือกใช้ขั้นตอนวิธีการแยกตัวประกอบของแฟร์มาต์แบบไม่คำนวณรากที่สองจะสามารถกำหนดค่าเริ่มต้นของ u เป็นจำนวนเต็มที่น้อยที่สุดที่มากกว่าหรือเท่ากับ $2\lceil \sqrt{n} \rceil$ และหาร 8 ได้ลงตัวเสมอ และสามารถเพิ่มค่าขึ้นได้รอบละ 8 ค่าในกรณีที่ u ยังไม่ใช่ค่าเป้าหมาย

7. การพิจารณาเศษจากการหารมอดูลัสด้วย 4, 6 และ 20

หัวข้อนี้จะกล่าวถึงการวิเคราะห์เศษที่ได้จากการหาร n ด้วย 4, 6 และ 20 มาพิจารณาร่วมกัน [49] เพื่อวิเคราะห์หารูปแบบของ x ที่ใกล้เคียงกับผลลัพธ์ให้มากที่สุดซึ่งจะช่วยให้สามารถตัดรูปแบบของ x ที่ไม่มีความเกี่ยวข้องกับผลลัพธ์ออกจากการคำนวณได้มากยิ่งขึ้น

เนื่องจากเศษที่ได้จากการหาร n ด้วย 4, 6, 20 มีค่าที่เป็นไปได้ทั้งหมด 2, 2 และ 8 ค่าตามลำดับ ดังนั้นหากนำทั้ง 3 กรณีมาพิจารณาพร้อมกันจะเกิดกรณีที่เป็นไปได้ทั้งหมด 2x2x8 = 32 กรณี ดังนี้

กรณีที่ 1: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 1

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 1 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 1, 5 หรือ 9 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 1, 5 หรือ 9 เสมอ

กรณีที่ 2: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 3

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 3 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 2 หรือ 8 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 3: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 7

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 7 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 4 หรือ 6 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 4: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 9

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 9 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 3, 5 หรือ 7 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 3, 5 หรือ 7 เสมอ

กรณีที่ 5: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 11

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 11 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 0, 4 หรือ 6 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 6: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 13

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 13 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 3 หรือ 7 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 3 หรือ 7 เสมอ

กรณีที่ 7: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 17

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 17 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 1 หรือ 9 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 1 หรือ 9 เสมอ

กรณีที่ 8: n mod 4 = 1, n mod 6 = 1 และ n mod 20 = 19

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 19 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 0, 2 หรือ 8 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 9: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 1

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 1 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 1, 5 หรือ 9 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 1, 5 หรือ 9 เสมอ

กรณีที่ 10: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 3

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 3 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 2 หรือ 8 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 11: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 7

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 7 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 4 หรือ 6 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 12: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 9

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 9 ได้ว่าเลขหลักหน่วยของ x

ต้องเป็น 3, 5 หรือ 7 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 3, 5 หรือ 7 เสมอ

กรณีที่ 13: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 11

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 11 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 0, 4 หรือ 6 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 14: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 13

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 13 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 3 หรือ 7 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 3 หรือ 7 เสมอ

กรณีที่ 15: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 17

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 17 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 1 หรือ 9 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคี่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 1 หรือ 9 เสมอ

กรณีที่ 16: n mod 4 = 1, n mod 6 = 5 และ n mod 20 = 19

เนื่องจาก n mod 4 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคี่เสมอ แต่ในทางกลับกัน n mod 20 = 19 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 0, 2 หรือ 8 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 1 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 17: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 1

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 1 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 1, 5 หรือ 9 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 18: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 3

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 3 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 2 หรือ 8 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 2 หรือ 8 เสมอ

กรณีที่ 19: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 7

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 7 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 4 หรือ 6 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 4 หรือ 6 เสมอ

กรณีที่ 20: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 9

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 9 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 3, 5 หรือ 7 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 21: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 11

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 11 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 0, 4 หรือ 6 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 0, 4 หรือ 6 เสมอ

กรณีที่ 22: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 13

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 13 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 3 หรือ 7 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 23: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 17

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 17 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 1 หรือ 9 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 24: n mod 4 = 3, n mod 6 = 1 และ n mod 20 = 19

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 1 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ไม่ลงตัว และ n mod 20 = 19 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 0, 2 หรือ 8 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ไม่ลงตัวและมีเลขหลักหน่วยเป็น 0, 2 หรือ 8 เสมอ

กรณีที่ 25: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 1

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 1 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 1, 5 หรือ 9 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 26: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 3

เนื่องจาก n mod 4 = 3 ได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 5 ได้ว่า x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 3 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 2 หรือ 8 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 2 หรือ 8 เสมอ

กรณีที่ 27: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 7

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 7 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 4 หรือ 6 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 4 หรือ 6 เสมอ

กรณีที่ 28: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 9

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 9 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 3, 5 หรือ 7 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 29: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 11

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 11 ได้ว่าเลขหลักหน่วยของ x ต้องเป็น 0, 4 หรือ 6 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 0, 4 หรือ 6 เสมอ

กรณีที่ 30: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 13

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 13 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 3 หรือ 7 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 31: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 17

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ แต่ในทางกลับกัน n mod 20 = 17 ซึ่งเลขหลักหน่วยของ x จะต้องเป็น 1 หรือ 9 เสมอ ซึ่งขัดแย้งกับเงื่อนไขของ x ที่พิจารณาจาก n mod 4 = 3 ดังนั้นกรณีนี้จึงเป็นไปไม่ได้ที่จะเกิดขึ้น

กรณีที่ 32: n mod 4 = 3, n mod 6 = 5 และ n mod 20 = 19

เนื่องจาก n mod 4 = 3 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มบวกคู่เสมอ n mod 6 = 5 ผลลัพธ์ของ x จะต้องเป็นจำนวนเต็มที่หาร 3 ลงตัว และ n mod 20 = 19 ได้ว่าเลขหลักหน่วยของ x

ต้องเป็น 0, 2 หรือ 8 เท่านั้น ดังนั้นหากพิจารณาทั้ง 3 ค่าพร้อมกันสรุปได้ว่า x จะต้องเป็นจำนวนเต็มบวกคู่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยเป็น 0, 2 หรือ 8 เสมอ

จากทั้ง 32 กรณีข้างต้นพบว่าปรากฏกรณีที่มีความเป็นไปได้ที่จะเกิดขึ้นเพียง 16 กรณี ประกอบไปด้วยกรณีที่ 1, 4, 6, 7, 9, 12, 14, 15, 18, 19, 21, 24, 26, 27, 29 และ 32 ซึ่งในแต่ละกรณีจะมีรูปแบบของ x ที่แตกต่างกัน อย่างไรก็ตามเนื่องจากการเลือกค่า x เพื่อใช้สำหรับการคำนวณหาจำนวนเต็ม y นั้นจะเลือกเพียงค่า x ที่มีรูปแบบตรงตามในแต่ละกรณีเท่านั้นจึงสามารถตัดค่าที่ไม่เกี่ยวข้องที่ไม่ตรงตามรูปแบบของ x ออกจากการคำนวณได้เป็นจำนวนมาก

ตัวอย่างที่ 8.2 จงแยกตัวประกอบ n = 1472129 โดยใช้หลักการพิจารณาเศษจากการหาร n ด้วย 4, 6 และ 20

วิธีทำ เนื่องจาก

$$29 \bmod 4 = 1$$
$$1472129 \bmod 6 = 5$$
$$1472129 \bmod 20 = 9$$

ดังนั้นรูปแบบของ x ซึ่งตรงกับกรณีที่ 12 คือจำนวนเต็มบวกคี่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยมีค่าเป็น 3 5 หรือ 7

จากขั้นตอนวิธีที่ 7.6 ที่เลือกใช้ค่า x ที่มีรูปแบบตรงกับกรณีที่ 12 จึงได้ว่า

5. $x = \left\lceil \sqrt{1472129} \right\rceil = 1214$

เนื่องจาก 1214 ไม่ตรงกับกรณีที่ 12 ดังนั้นจึงสามารถเพิ่มค่า x เพื่อให้มีรูปแบบที่ตรงกับกรณีดังกล่าวโดยไม่ต้องคำนวณหาค่ารากที่สองของค่าที่ไม่ตรงรูปแบบซึ่ง 1215 คือค่าที่ใกล้เคียงกับ 1214 มากที่สุดและตรงกับเงื่อนไขที่ 12

6. $y = \sqrt{1215^2 - 1472129} = 64$

เนื่องจาก y เป็นจำนวนเต็มจึงสรุปได้ว่า $p = 1215 + 64 = 1279$ และ $q = 1215 - 64 = 1151$ เป็นตัวประกอบของ n

จากตัวอย่างที่ 8.2 แสดงให้เห็นว่าไม่มีการคำนวณค่ารากที่สองเมื่อ x มีค่าเป็น 1214 ซึ่งสามารถลดเวลาการคำนวณลงได้ อย่างไรก็ตามจากตัวอย่างสังเกตว่าสามารถลดจำนวนรอบการคำนวณได้เพียง 1 รอบเนื่องจากตัวประกอบทั้งสองค่ามีค่าที่ใกล้เคียงกัน ในตัวอย่างถัดไปจะแสดงให้

เห็นว่ากรณีที่ตัวประกอบแตกต่างกันมากขึ้นจะส่งผลให้สามารถลดจำนวนรอบของการคำนวณได้มากยิ่งขึ้น

ตัวอย่างที่ 8.3 จงแยกตัวประกอบ n = 2415911 โดยใช้หลักการพิจารณาเศษเหลือจากการหาร n ด้วย 4, 6 และ 20

วิธีทำ เนื่องจาก $11 \bmod 4 = 3$

$2415911 \bmod 6 = 5$

$2415911 \bmod 20 = 11$

ดังนั้นรูปแบบของ x ซึ่งตรงกับกรณีที่ 29 คือจำนวนเต็มบวกคู่ที่หาร 3 ลงตัวและมีเลขหลักหน่วยมีค่าเป็น 0, 4 หรือ 6

จากขั้นตอนวิธีที่ 7.6 ที่เลือกใช้ค่า x ที่มีรูปแบบตรงกับกรณีที่ 29 จึงได้ว่า

1. $x = \left\lceil \sqrt{2415911} \right\rceil = 1555$

เนื่องจาก 1555 ไม่ตรงกับกรณีที่ 29 ดังนั้นจึงสามารถเพิ่มค่า x เพื่อให้มีรูปแบบที่ตรงกับกรณีดังกล่าวโดยไม่ต้องคำนวณหาค่ารากที่สองของค่าที่ไม่ตรงรูปแบบซึ่ง 1560 คือค่าที่ใกล้เคียงกับ 1555 มากที่สุดและตรงกับเงื่อนไขที่ 29

2. $y = \sqrt{1260^2 - 2415911} = 133$

เนื่องจาก y เป็นจำนวนเต็มจึงสรุปได้ว่า p = 1560 + 133 = 1693 และ q = 1560 – 133 = 1427เป็นตัวประกอบของ n

จากตัวอย่างที่ 8.3 แสดงให้เห็นว่าไม่มีการคำนวณค่ารากที่สองเมื่อ x มีค่าอยู่ระหว่าง 1555 ถึง 1559 ซึ่งสามารถลดเวลาการคำนวณลงได้ 5 รอบ

ข้อสังเกต 1556 จะไม่ถูกนำมาใช้สำหรับคำนวณหารากที่สองเนื่องจากหาร 3 ไม่ลงตัว

8. การพิจารณากลุ่มตัวเลขหลักสุดท้ายของมอดูลัส

การพิจารณากลุ่มตัวเลขหลักสุดท้ายของ n จะช่วยให้ทราบรูปแบบของ u และ v ที่เป็นตัวแปรสำคัญสำหรับการคำนวณหาตัวประกอบสำหรับขั้นตอนวิธีที่ 7.7 ได้อย่างละเอียดมากยิ่งขึ้น [34] โดยยิ่งหากพิจารณาตัวเลขจำนวนมากขึ้น ส่งผลให้ทราบรูปแบบของ u และ v ได้ละเอียดมากขึ้นและ

ช่วยให้กำจัดค่าที่ไม่เกี่ยวข้องออกจากการคำนวณมากยิ่งขึ้น ดังนั้นหัวใจสำคัญของวิธีการนี้คือต้องทราบรูปแบบที่เป็นไปได้ทั้งหมดของ u และ v เพื่อวิเคราะห์หาอัตราการเพิ่มค่าของทั้งสองค่าสำหรับแต่ละรอบของการคำนวณ

กำหนดให้ $LSG_m(z)$ แทนสัญลักษณ์ที่แทนเลข m หลักสุดท้ายของ z เมื่อ $z \in \mathbb{Z}^+$ บทตั้งที่ 8.1, 8.2 และ 8.3 แสดงกฎการคำนวณสำหรับเลข m หลักสุดท้าย

บทตั้งที่ 8.1 กำหนดให้ $a, b \in \mathbb{Z}^+$ โดยที่ $a > b$ ได้ว่า

1. $LSG_m(a + b) = LSG_m(LSG_m(a) + LSG_m(b))$
2. $LSG_m(a - b) = LSG_m(LSG_m(a) - LSG_m(b) + 10^m)$
3. $LSG_m(ab) = LSG_m(LSG_m(a) \times LSG_m(b))$

พิสูจน์

1. จากหลักการบวกเลขระหว่างจำนวนเต็ม 2 ค่าจะเริ่มบวกจากตำแหน่งตัวเลขที่มีค่านัยสำคัญต่ำที่สุด (เลขหลักหน่วย) และเรียงไปจนกระทั่งถึงตำแหน่งสุดท้ายซึ่งเป็นตำแหน่งที่มีค่านัยสำคัญสูงที่สุด และเนื่องจาก ผลลัพธ์ของ $LSG_m(a + b)$ คือกลุ่มตัวเลข m ตัวสุดท้ายของผลบวกระหว่าง a และ b ดังนั้นจึงไม่จำเป็นต้องบวกตัวเลขทุกตัวของ a และ b แต่สามารถเลือกเพียงเลข m ตัวสุดท้ายของ a และ b เพื่อมาบวกกันหรือการหาผลลัพธ์ของ $LSG_m(a) + LSG_m(b)$ แต่อย่างไรก็ตามผลลัพธ์ที่ได้อาจมีตัวทดไปยังตำแหน่งถัดไปซึ่งไม่ใช่ตำแหน่งที่ต้องการจึงสามารถถูกกรองได้โดยนำผลลัพธ์ที่ได้ไปใส่ไว้ในสัญลักษณ์ที่คืนค่าเป็นเลข m ตัวสุดท้ายอีกครั้งคือ $LSG_m(LSG_m(a) + LSG_m(b))$

2. จากหลักการลบเลขระหว่างจำนวนเต็ม 2 ค่าที่ตัวตั้งมีค่ามากกว่าตัวลบจะเริ่มลบจากตำแหน่งตัวเลขที่มีค่านัยสำคัญต่ำที่สุด (เลขหลักหน่วย) และเรียงไปจนกระทั่งถึงตำแหน่งสุดท้ายซึ่งเป็นตำแหน่งที่มีนัยสำคัญสูงที่สุด และเนื่องจาก ผลลัพธ์ของ $LSG_m(a - b)$ คือกลุ่มตัวเลข m ตัวสุดท้ายของผลลบระหว่าง a และ b ดังนั้นจึงไม่จำเป็นต้องลบตัวเลขทุกตัวของ a และ b แต่สามารถเลือกเพียงเลข m ตัวสุดท้ายของ a และ b เพื่อมาลบกันหรือการหาผลลัพธ์ของ $LSG_m(a) - LSG_m(b)$ อย่างไรก็ตามมีความเป็นไปได้ที่ $LSG_m(a)$ จะมีค่าน้อยกว่า $LSG_m(b)$ จึงจำเป็นต้องยืมค่าจากตำแหน่งที่ m ของ a ซึ่งค่าประจำตำแหน่งคือ 10^m ดังนั้นจึงได้ผลลัพธ์เป็น $LSG_m(a) - LSG_m(b) + 10^m$ ในทางกลับกันหาก $LSG_m(a)$ มีค่ามากกว่า $LSG_m(b)$ ผลลัพธ์ที่ได้จะมีตัวเลขทั้งหมด $m + 1$ ตัว (เนื่องจากมีการรวม 10^m เข้าไปด้วย) ดังนั้นจึงจำเป็นต้องกรองค่าตำแหน่งส่วนเกินออกโดยนำผลลัพธ์ที่ได้ไปใส่ไว้ในสัญลักษณ์ที่คืนค่าเป็นเลข m ตัวสุดท้ายอีกครั้งคือ $LSG_m(LSG_m(a) - LSG_m(b) + 10^m)$

3. จากหลักการคูณเลขระหว่างจำนวนเต็ม 2 ค่าจะเริ่มคูณจากคู่ตัวเลขที่มีค่านัยสำคัญต่ำที่สุด (เลขหลักหน่วย) และเรียงไปจนกระทั่งถึงตำแหน่งสุดท้ายซึ่งเป็นตำแหน่งที่มีนัยสำคัญสูงที่สุด และเนื่องจาก ผลลัพธ์ของ $LSG_m(ab)$ คือกลุ่มตัวเลข m ตัวสุดท้ายของผลคูณระหว่าง a และ b ดังนั้นจึงไม่จำเป็นต้องหาผลคูณตัวเลขทุกตัวของ a และ b แต่สามารถเลือกเพียงเลข m ตัวสุดท้ายของ a และ b เพื่อมาคูณกันหรือการหาผลลัพธ์ของ $LSG_m(a) \times LSG_m(b)$ แต่อย่างไรก็ตามผลลัพธ์ที่ได้อาจมีสมาชิกมากกว่า m จึงจำเป็นต้องกรองค่าตำแหน่งส่วนเกินออกโดยนำผลลัพธ์ที่ได้ไปใส่ไว้ในสัญลักษณ์ที่คืนค่าเป็นเลข m ตัวสุดท้ายอีกครั้งคือ $LSG_m(LSG_m(a) \times LSG_m(b))$ □

บทตั้งที่ 8.2 $LSG_m(a + b\text{x}10^m) = LSG_m(a)$

พิสูจน์ จากบทตั้งที่ 8.1 ข้อที่ 1 ได้ว่า

$$LSG_m(a + b\text{x}10^m) = LSG_m(LSG_m(a) + LSG_m(b\text{x}10^m))$$

เนื่องจาก $10^m = 1\underbrace{000\cdots000}_{m \text{ ตัว}}$

หรือกล่าวอีกนัยหนึ่งคือ 10^m จะมีเลข 0 ต่อท้ายเลข 1 เป็นจำนวน m ตัว จึงมีตัวเลขเป็นจำนวนทั้งหมด $m + 1$ ตัว ได้ว่า $LSG_m(b \times 10^m) = 0$

ดังนั้น
$$\begin{aligned} LSG_m(a + b\text{x}10^m) &= LSG_m(LSG_m(a) + 0)) \\ &= LSG_m(LSG_m(a)) \\ &= LSG_m(a) \end{aligned}$$ □

บทตั้งที่ 8.3 กำหนดให้ $a, b \in \mathbb{Z}^+$ โดยที่ $a = a_x a_{x-1} a_{x-2} \cdots a_0$ และ $b = b_y b_{y-1} b_{y-2} \cdots b_0$ และ $x > m$ ได้ว่าผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $LSG_m(a + b\text{x}10^{m-1})$ สามารถคำนวณได้จาก $(a_{m-1} + b_0) \bmod 10$

พิสูจน์ จากบทตั้งที่ 8.2 ได้ว่า

$$LSG_{m-1}(a + b\text{x}10^{m-1}) = LSG_{m-1}(a)$$

ความหมายคือหากพิจารณาเพียง $m - 1$ ตำแหน่ง (ตำแหน่งที่ 0 ถึงตำแหน่งที่ m-2) ผลบวกระหว่าง $a + b\text{x}10^{m-1}$ ที่จะมีค่าเท่ากับ a เสมอ เนื่องจากผลลัพธ์ของ 10^{m-1} จะมี 0 จำนวน $m - 1$ ตัว ดังนั้นจึงไม่เกิดตัวทดไปยังตำแหน่งที่ $m - 1$ และผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $b\text{x}10^{m-1}$ คือ b_0

หากพิจารณาตำแหน่งที่ $m - 1$ ของ $LSG_m(a + b\text{x}10^{m-1})$ หมายถึงการพิจารณาเพียงผลลัพธ์ที่อยู่ในตำแหน่งที่มีนัยสำคัญสูงสุดเท่านั้น และเนื่องจากผลลัพธ์ $m - 1$ ตำแหน่งแรกระหว่าง $a + b\text{x}10^{m-1}$ ไม่เกิดตัวทดมายังตำแหน่งที่ $m - 1$ จึงสามารถหาผลลัพธ์ได้จากการบวกกันระหว่าง

ประจำตำแหน่งที่ $m-1$ ของ a (a_{m-1}) และ $b\text{x}10^{m-1}$ (b_0) คือ $a_{m-1}+b_0$ อย่างไรก็ตามผลบวกระหว่างทั้งสองค่าอาจเป็นไปได้ที่จะมีค่าเกิน 10 ซึ่งจะเกิดตัวทดไปยังตำแหน่งถัดไป แต่จากบทตั้งนี้สนใจเพียงผลลัพธ์ในตำแหน่งที่ $m-1$ เท่านั้น ดังนั้นจึงสามารถกรองตัวทดออกได้โดยการคำนวณ $c_{m-1} = (a_{m-1} + b_0) \bmod 10$ □

8.1 การแทนค่าการคูณระหว่างจำนวนเต็มสองจำนวน

กำหนดให้ $a, b \in \mathbb{Z}^+$, $c = ab$ โดยที่ $a = a_x a_{x-1} a_{x-2} \cdots a_0$, $b = b_y b_{y-1} b_{y-2} \cdots b_0$ และ $c = c_z c_{z-1} c_{z-2} \cdots c_0$ แล้วผลลัพธ์ในแต่ละตำแหน่งของ c สามารถพิจารณาได้ดังนี้

ตำแหน่งที่ 0: ผลลัพธ์ในหลักหน่วย (ตำแหน่งที่ 0) เกิดจากการคูณกันระหว่าง a_0 และ b_0 อย่างไรก็ตามผลลัพธ์ที่ได้อาจมีค่าเกิน 1 ตำแหน่งจึงจำเป็นต้องกรองเพื่อเอาตัวทดออกได้ดังนี้

$$d_0 = a_0 b_0 \tag{8.1}$$

$$c_0 = d_0 \bmod 10 \tag{8.2}$$

ตำแหน่งที่ 1: ผลลัพธ์หลักสิบ (ตำแหน่งที่ 1) จะเกิดจากผลบวกของผลคูณของแต่ละคู่ดังต่อไปนี้ (a_0, b_1) และ (a_1, b_0) อย่างไรก็ตามจำเป็นต้องบวกกับตัวทดที่ได้จากผลลัพธ์หลักหน่วยซึ่งสามารถพิจารณาจาก $\left\lfloor \frac{d_0}{10} \right\rfloor$

$$d_1 = a_0 b_1 + a_1 b_0 + \left\lfloor \frac{d_0}{10} \right\rfloor \tag{8.3}$$

$$c_1 = d_1 \bmod 10 \tag{8.4}$$

ตำแหน่งที่ 2: ผลลัพธ์หลักร้อย (ตำแหน่งที่ 2) จะเกิดจากผลบวกของผลคูณของแต่ละคู่ดังต่อไปนี้ (a_0, b_2), (a_1, b_1) และ (a_2, b_0) อย่างไรก็ตามจำเป็นต้องบวกกับตัวทดที่ได้จากผลลัพธ์หลักหน่วยซึ่งสามารถพิจารณาจาก $\left\lfloor \frac{d_1}{10} \right\rfloor$

$$d_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 + \left\lfloor \frac{d_1}{10} \right\rfloor \tag{8.5}$$

$$c_2 = d_2 \bmod 10 \tag{8.6}$$

ตำแหน่งที่ *m*: จากผลลัพธ์ c_0, c_1 และ c_2 บอกเป็นนัยว่าผลลัพธ์ตำแหน่งที่ m จะเกิดจากผลบวกของผลคูณของแต่ละคู่ดังต่อไปนี้ (a_0, b_m), (a_1, b_{m-1}), (a_2, b_{m-2}), ..., (a_m, b_0) อย่างไรก็ตามอาจจำเป็นต้องบวกกับตัวทดที่ได้จากผลลัพธ์หลักหน่วยซึ่งสามารถพิจารณาจาก $\left\lfloor \frac{d_{m-1}}{10} \right\rfloor$

$$d_m = a_0b_m + a_1b_{m-1} + a_2b_{m-2} + \cdots + a_mb_0 + \left\lfloor \frac{d_{m-1}}{10} \right\rfloor \tag{8.7}$$

$$c_m = d_m \bmod 10 \tag{8.8}$$

ตัวอย่างที่ 8.4 พิจารณาผลลัพธ์ของ $LSG_3(187 \times 273)$ โดยใช้หลักการแทนค่าการคูณ

วิธีทำ กำหนดให้ $a = 187$ และ $b = 273$ ดังนั้นจากหลักการแทนค่าการคูณได้ว่า $a_0 = 7$, $a_1 = 8$, $a_2 = 1$, $b_0 = 3$, $b_1 = 7$ และ $b_2 = 2$

ตำแหน่งที่ 0:

$$d_0 = 7 \times 3 = 21$$

$$c_0 = 21 \bmod 10 = 1$$

ตำแหน่งที่ 1:

$$d_1 = 7x7 + 8x3 + \left\lfloor \frac{21}{10} \right\rfloor = 75$$

$$c_1 = 75 \bmod 10 = 5$$

ตำแหน่งที่ 2:

$$d_2 = 7x2 + 8x7 + 1x3 + \left\lfloor \frac{75}{10} \right\rfloor = 80$$

$$c_2 = 80 \bmod 10 = 0$$

ดังนั้น $LSG_3(187 \times 273) = c_2c_1c_0 = 051 = 51$

จากบทตั้งที่ 8.1, 8.2 และ 8.3 และหลักการแทนค่าการคูณระหว่างจำนวนเต็มสองจำนวน จึงได้มาซึ่งทฤษฎีบทที่ 8.4 และ 8.5 เพื่อใช้สำหรับพิจารณาหารูปแบบของ u และ v ดังนี้

ทฤษฎีบทที่ 8.4 กำหนดให้ a, b เป็นจำนวนเต็มบวกคี่ที่เลขหลักหน่วยมีค่าเท่ากัน และ $LSG_m(ab) = LSG_m(n)$ เมื่อ $m \geq 1$ แล้วได้ว่า

$$=LSG_m((a + 10^{m-1})(b + 9x10^{m-1})) = LSG_m((a + 9x10^{m-1})(b + 10^{m-1})) = LSG_m(n)$$

พิสูจน์

กำหนดให้ $LSG_m(a) = a_{m-1}a_{m-2}a_{m-3} \cdots a_0$

$LSG_m(b) = b_{m-1}b_{m-2}b_{m-3} \cdots b_0$

$LSG_m(ab) = LSG_m(c) = c_{m-1}c_{m-2}c_{m-3} \cdots c_0$

การพิสูจน์ถูกแบ่งออกเป็น 2 กรณี ดังนี้

กรณีที่ 1 $LSG_m((a + 10^{m-1})(b + 9x10^{m-1})) = LSG_m(n)$

จากบทตั้งที่ 8.2 ได้ว่า

$$LSG_{m-1}(a + 10^{m-1}) = LSG_{(m-1)}(a)$$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $a + 10^{m-1}$ และ a มีค่าเท่ากัน

และ $$LSG_{m-1}(b + 9x10^{m-1}) = LSG_{(m-1)}(b)$$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $b + 9x10^{m-1}$ และ b มีค่าเท่ากัน

และจากบทตั้งที่ 8.3 ได้ว่า

ผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $a + 10^{m-1}$ คือ $(a_{m-1} + 1) \bmod 10$ และ

ผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $b + 9x10^{m-1}$ คือ $(b_{m-1} + 9) \bmod 10$

จากสมการที่ (8.7) และ (8.8) ได้ว่าผลคูณตำแหน่งที่ m - 1 ระหว่าง a และ b คือ

$$c_{m-1} = (a_0b_{m-1} + a_1b_{m-2} + a_2b_{m-3} + \cdots + a_{m-1}b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

กำหนดให้ x แทนผลคูณตำแหน่งที่ m - 1 ระหว่าง $a + 10^{m-1}$ และ $b + 9x10^{m-1}$ ซึ่งมีค่าเป็นดังนี้

$$x = (a_0(b_{m-1}+9) + a_1b_{m-2} + \cdots + a_{m-2}b_1 + (a_{m-1}+1)b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + 9a_0 + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

เนื่องจาก $a_0 = b_0$ ดังนั้นจึงได้ว่า

$$x = (a_0b_{m-1} + 9a_0 + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 9a_0 + a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 10a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) + \bmod 10$$

เนื่องจาก $10a_0 \bmod 10 = 0$ ดังนั้น

$$x = (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 10a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10 = c_{m-1}$$

ดังนั้นสรุปได้ว่า $LSG_m((a + 10^{m-1})(b + 9x10^{m-1})) = LSG_m(ab) = LSG_m(n)$

กรณีที่ 2 $LSG_m((a + 9x10^{m-1})(b + 10^{m-1})) = LSG_m(n)$

จากบทตั้งที่ 8.2 ได้ว่า

$$LSG_{m-1}(a + 9x10^{m-1}) = LSG_{m-1}(a)$$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $a + 9x10^{m-1}$ และ a มีค่าเท่ากัน

และ $LSG_{m-1}(b + 10^{m-1}) = LSG_{m-1}(b)$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $b + 10^{m-1}$ และ b มีค่าเท่ากัน

และจากบทตั้งที่ 8.3 ได้ว่า

ผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $a + 9x10^{m-1}$ คือ $(a_{m-1} + 9) \bmod 10$ และ

ผลลัพธ์ตำแหน่งที่ $m - 1$ ของ $b + 10^{m-1}$ คือ $(b_{m-1} + 1) \bmod 10$

กำหนดให้ y แทนผลคูณตำแหน่งที่ m-1 ระหว่าง $a + 9x10^{m-1}$ และ $b + 10^{m-1}$ ซึ่งมีค่าเป็นดังนี้

$$y = (a_0(b_{m-1}+ 1) + a_1b_{m-2} + \cdots + a_{m-2}b_1 + (a_{m-1}+ 9)b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_0 + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 9b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

เนื่องจาก $a_0 = b_0$ ดังนั้นจึงได้ว่า

$$y = (a_0b_{m-1} + a_0 + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 9a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + a_0 + 9a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + 10a_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0b_{m-1} + a_1b_{m-2} + \cdots + a_{m-2}b_1 + a_{m-1}b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10 = c_{m-1}$$

ดังนั้นสรุปได้ว่า $LSG_m((a + 9x10^{m-1})(b + 10^{m-1})) = LSG_m(ab) = LSG_m(n)$
ดังนั้นจากทั้งสองกรณีได้ว่า
$LSG_m((a + 10^{m-1})(b + 9x10^{m-1})) = LSG_m((a + 9x10^{m-1})(b + 10^{m-1})) = LSG_m(n)$ □

ทฤษฎีบทที่ 8.5 กำหนดให้ a, b เป็นจำนวนเต็มบวกคี่ที่เลขหลักหน่วยมีค่าไม่เท่ากันโดยที่ $LSG_m(ab) = LSG_m(n)$ เมื่อ $m \geq 1$ และมีจำนวนเต็มบวกคี่ k_1 และ k_2 ที่มีค่าเป็น 1, 3, 7 หรือ 9 และทำให้ $LSG(a) \times k_2 + LSG(b) \times k_1 \bmod 10 = 0$ แล้วได้ว่า

$$LSG_m((a + k_1x10^{m-1})(b + k_2x10^{m-1})) = LSG_m(n)$$

พิสูจน์

กำหนดให้ $LSG_m(a) = a_{m-1}a_{m-2}a_{m-3}\cdots a_0$

$LSG_m(b) = b_{m-1}b_{m-2}b_{m-3}\cdots b_0$

$LSG_m(ab) = LSG_m(c) = c_{m-1}c_{m-2}c_{m-3}\cdots c_0$

จากบทตั้งที่ 8.2 ได้ว่า

$$LSG_{m-1}(a + k_1x10^{m-1}) = LSG_{m-1}(a)$$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $a + k_1x10^{m-1}$ และ a มีค่าเท่ากัน

และ $LSG_{m-1}(b + k_2x10^{m-1}) = LSG_{m-1}(b)$

ความหมายคือตัวเลขตำแหน่งที่ 0 ถึง $m - 2$ ของ $b + k_2x10^{m-1}$ และ b มีค่าเท่ากัน

และจากบทตั้งที่ 8.3 ได้ว่า

ผลลัพธ์ตำแหน่งที่ $m-1$ ของ $a + k_1 \times 10^{m-1}$ คือ $(a_{m-1} + k_1) \bmod 10$ และ

ผลลัพธ์ตำแหน่งที่ $m-1$ ของ $b + k_2 \times 10^{m-1}$ คือ $(b_{m-1} + k_2) \bmod 10$

กำหนดให้ z แทนผลคูณตำแหน่งที่ m - 1 ระหว่าง $a + k_1 \times 10^{m-1}$ และ $b + k_2 \times 10^{m-1}$ ซึ่งมีค่าเป็นดังนี้

$$z = (a_0(b_{m-1} + k_2) + a_1 b_{m-2} + \cdots + a_{m-2} b_1 + (a_{m-1} + k_1) b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0 b_{m-1} + k_2 a_0 + a_1 b_{m-2} + \cdots + a_{m-2} b_1 + a_{m-1} b_0 + k_1 b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

$$= (a_0 b_{m-1} + a_1 b_{m-2} + \cdots + a_{m-2} b_1 + a_{m-1} b_0 + k_2 a_0 + k_1 b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10$$

เนื่องจาก $(k_2 a_0 + k_1 b_0) \bmod 10 = 0$ ดังนั้น

$$z = (a_0 b_{m-1} + a_1 b_{m-2} + \cdots + a_{m-2} b_1 + a_{m-1} b_0 + \left\lfloor \frac{d_{m-2}}{10} \right\rfloor) \bmod 10 = c_{m-1}$$

ดังนั้นสรุปได้ว่าหาก $a_0 \neq b_0$ และ $(k_2 a_0 + k_1 b_0) \bmod 10 = 0$ แล้วได้ว่า

$$\mathrm{LSG}_m((a + k_1 \times 10^{m-1})(b + k_2 \times 10^{m-1})) = \mathrm{LSG}_m(n) \qquad \square$$

อย่างไรก็ตามทฤษฎีบทที่ 8.4 จำเป็นต้องหา k_1 และ k_2 ที่ทำให้ $(k_2 a_0 + k_1 b_0) \bmod 10 = 0$ ดังนั้นเพื่อช่วยลดเวลาการคำนวณขั้นตอนดังกล่าวนี้ จึงเสนอคู่ความสัมพันธ์ระหว่างทั้งสองค่าดังกล่าวนี้ที่เป็นไปได้ทั้งหมดไว้ดังตารางที่ 8.2

ตารางที่ 8.2 แสดงคู่ความสัมพันธ์ (k_1, k_2) ที่เป็นไปได้ทั้งหมดที่ทำให้ $(k_2 a_0 + k_1 b_0) \bmod 10 = 0$ เมื่อ $a_0 \neq b_0$ โดยแต่ละคู่ความสัมพันธ์ (a_0, b_0) จะเกิด (k_1, k_2) ที่เป็นไปได้ทั้งหมด 4 กรณี ซึ่งสามารถเลือกใช้กรณีใดกรณีหนึ่งได้ ตัวอย่างเช่นกำหนดให้ $(a_0, b_0) = (1, 3)$ สามารถเลือกใช้ (k_1, k_2) ได้ 4 กรณีคือ (1, 7), (3, 1), (7, 9) หรือ (9, 3) เป็นต้น

ตารางที่ 8.2 คู่ความสัมพันธ์ระหว่าง k_1 และ k_2 ที่เป็นไปได้ทั้งหมดที่ทำให้ $(k_2 a_0 + k_1 b_0) \bmod 10 = 0$ เมื่อ $a_0 \neq b_0$

LSG(n)	(LSG(a), LSG(b)) หรือ (a_0, b_0)	(k_1, k_2)			
		คู่ที่ 1	**คู่ที่ 2**	**คู่ที่ 3**	**คู่ที่ 4**
1	(3, 7)	(1, 1)	(3, 3)	(7, 7)	(9, 9)
3	(1, 3)	(1, 7)	(3, 1)	(7, 9)	(9, 3)
	(7, 9)	(1, 3)	(3, 9)	(7, 1)	(9, 7)
7	(1, 7)	(1, 3)	(3, 9)	(7, 1)	(9, 7)
	(3, 9)	(1, 7)	(3, 1)	(7, 9)	(9, 3)
9	(1, 9)	(1, 1)	(3, 3)	(7, 7)	(9, 9)

8.2 การหารูปแบบของ u และ v

โดยทั่วไปแล้วอัตราการเพิ่มค่าของ u และ v มีค่าเท่ากันคือ 2 อย่างไรก็ตามการพิจารณากลุ่มตัวเลขหลักสุดท้ายของ n จะช่วยให้ทราบรูปแบบกลุ่มเลขหลักสุดท้ายของ u และ v ได้ชัดเจนมากยิ่งขึ้น ดังนั้นจึงสามารถตัดค่าที่ไม่ตรงตามรูปแบบออกจากการคำนวณได้ส่งผลให้อัตราการเพิ่มค่าของ u และ v มีค่ามากกว่า 2 เสมอ โดยเฉพาะอย่างยิ่งหากจำนวนกลุ่มตัวเลขหลักสุดท้ายที่ถูกนำมาพิจารณามีปริมาณที่มากขึ้นแล้วมีความเป็นไปได้สูงที่อัตราการเพิ่มค่าของ u และ v จะเพิ่มมากยิ่งขึ้นตามไปด้วย โดยการหารูปแบบที่เป็นไปได้ทั้งหมดของ u และ v ที่พิจารณาจากกลุ่มตัวเลขหลักสุดท้ายของ n จำเป็นต้องใช้ทฤษฎีบทที่ 8.4 และ 8.5 เป็นเครื่องมือสำคัญสำหรับวิเคราะห์หารูปแบบที่เป็นไปได้ทั้งหมด

กำหนดให้เลข m - 1 ตัวสุดท้ายที่เป็นไปได้ทั้งหมดของคู่ความสัมพันธ์ของตัวประกอบของค่า n ถูกเปิดเผย ดังนั้นสามารถหาเลข m ตัวสุดท้ายที่เป็นไปได้ทั้งหมดที่จะได้มาซึ่งคู่ความสัมพันธ์ของตัวประกอบของค่า n ดังต่อไปนี้

ขั้นตอนที่ 1: แบ่งคู่ความสัมพันธ์ของตัวประกอบที่พิจารณาจากเลข m – 1 ตัวสุดท้ายออกเป็นกลุ่มๆละ 1 คู่

ขั้นตอนที่ 2: สำหรับแต่ละกลุ่มให้ใช้คู่ความสัมพันธ์จากขั้นตอนที่ 1 เพื่อพิจารณาหาเลข m ตัวสุดท้ายของทั้งสองค่าที่เป็นไปได้ที่จะเป็นตัวประกอบของมอดูลัสโดยหามาเพียง 1 คู่และตัดคู่ความสัมพันธ์คู่เก่าทิ้งไป

ขั้นตอนที่ 3: สำหรับแต่ละกลุ่มให้พิจารณาหาคู่ความสัมพันธ์ที่เหลือทั้งหมดโดยแบ่งออก 2 กรณีคือ

กรณีที่ 1 (เลขหลักหน่วยของคู่ความสัมพันธ์เท่ากัน): ใช้ทฤษฎีที่ 8.4 เพื่อวิเคราะห์หาคู่ความสัมพันธ์ที่เหลือ

กรณีที่ 2 (เลขหลักหน่วยของคู่ความสัมพันธ์ไม่เท่ากัน): ใช้ทฤษฎีที่ 8.5 เพื่อวิเคราะห์หาคู่ความสัมพันธ์ที่เหลือ

ตัวอย่างที่ 8.5 กำหนดให้เลขหลักสุดท้ายของคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่จะเป็นตัวประกอบของ n ที่มีเลขหลักหน่วยเป็น 7 คือ (1, 7) และ (3, 9) จงหาเลข 2 ตัวสุดท้ายคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่จะเป็นตัวประกอบของ n ที่มีเลขสองหลักสุดท้ายเป็น 97

วิธีทำ เนื่องจากเลขหลักสุดท้ายของคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่มีโอกาสที่จะเป็นตัวประกอบของ n ที่มีเลขหลักหน่วยเป็น 7 มี 2 คู่ เพราะฉะนั้นการพิจารณาหาเลข 2 หลักสุดท้ายของคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่มีโอกาสที่จะเป็นตัวประกอบของ n ที่มีเลข 2 หลักสุดท้ายเป็น 97 จะถูกแบ่งเป็น 2 กลุ่ม โดยมีลำดับขั้นตอนการดำเนินการเป็นดังนี้

ขั้นตอนที่ 1: แบ่งกลุ่มคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่พิจารณาจากเลขหลักสุดท้ายเพียง 1 หลัก

กลุ่มที่ 1: (1, 7)

กลุ่มที่ 2: (3, 9)

ขั้นตอนที่ 2: หาคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่พิจารณาจากเลข 2 หลักที่มีโอกาสที่จะเป็นตัวประกอบของ n ที่มีเลข 2 หลักสุดท้ายเป็น 97

พิจารณากลุ่มที่ 1 จะใช้ (1, 7) เพื่อพิจารณาหาตัวเลขที่อยู่ตำแหน่งด้านหน้าของแต่ละคู่ซึ่งสมมติค่าที่หาได้คือ (01, 97)

พิจารณากลุ่มที่ 2 จะใช้ (3, 9) เพื่อพิจารณาหาตัวเลขที่อยู่ตำแหน่งด้านหน้าของแต่ละคู่ซึ่งสมมติค่าที่หาได้คือ (03, 99)

ดังนันสมาชิกใหม่แต่ละกลุ่มเป็นดังนี้

กลุ่มที่ 1: (01, 97)

กลุ่มที่ 2: (03, 99)

ขั้นตอนที่ 3: ใช้ทฤษฎีที่ 8.4 และ 8.5 เพื่อหาคู่ความสัมพันธ์ที่เหลือ

พิจารณากลุ่มที่ 1 เนื่องจากเลขหลักหน่วยของคู่ความสัมพันธ์ในกลุ่มคือ (1, 7) ซึ่งมีค่าไม่เท่ากันจึงต้องใช้ทฤษฎีที่ 8.5 โดยจากตารางที่ 8.2 เลือก $(k_1, k_2) = (1, 3)$

จากทฤษฎีที่ 8.5 ได้ว่า

$LSG_2((1 + 1x10)(97 + 3x10)) = LSG_2(11 \times 127) = LSG_2(11 \times 27) = 97$

$LSG_2((11 + 1x10)(27 + 3x10)) = LSG_2(21 \times 57) = 97$

$LSG_2((21 + 1x10)(57 + 3x10)) = LSG_2(31 \times 87) = 97$

$LSG_2((31 + 1x10)(87 + 3x10)) = LSG_2(41 \times 117) = LSG_2(41 \times 17) = 97$

$LSG_2((41 + 1x10)(17 + 3x10)) = LSG_2(51 \times 47)$

$LSG_2((51 + 1x10)(47 + 3x10)) = LSG_2(61 \times 77)$

$LSG_2((61 + 1x10)(77 + 3x10)) = LSG_2(71 \times 107) = LSG_2(71 \times 7) = 97$

$LSG_2((71 + 1x10)(7 + 3x10)) = LSG_2(81 \times 37) = 97$

$LSG_2((81 + 1x10)(37 + 3x10)) = LSG_2(91 \times 67) = 97$

$LSG_2((91 + 1x10)(67 + 3x10)) = LSG_2(101 \times 97) = LSG_2(1 \times 97) = 97$ => ซ้ำรูปแบบแรก

พิจารณากลุ่มที่ 2 เนื่องจากเลขหลักหน่วยของคู่ความสัมพันธ์ในกลุ่มคือ (3, 9) ซึ่งมีค่าไม่เท่ากันจึงต้องใช้ทฤษฎีที่ 8.5 โดยจากตารางที่ 8.2 เลือก $(k_1, k_2) = (1, 7)$

จากทฤษฎีที่ 8.5 ได้ว่า

$LSG_2((3 + 1x10)(99 + 7x10)) = LSG_2(13 \times 169) = LSG_2(13 \times 69) = 97$

$LSG_2((13 + 1x10)(69 + 7x10)) = LSG_2(23 \times 139) = LSG_2(23 \times 39) = 97$

$LSG_2((23 + 1x10)(39 + 7x10)) = LSG_2(33 \times 109) = LSG_2(33 \times 9) = 97$

$LSG_2((33 + 1x10)(9 + 7x10)) = LSG_2(43 \times 79) = 97$

$LSG_2((43 + 1x10)(79 + 7x10)) = LSG_2(53 \times 149) = LSG_2(53 \times 49) = 97$

$LSG_2((53 + 1x10)(49 + 7x10)) = LSG_2(63 \times 119) = = LSG_2(63 \times 19) = 97$

$LSG_2((63 + 1x10)(19 + 7x10)) = LSG_2(73 \times 89) = 97$

$LSG_2((73 + 1x10)(89 + 7x10)) = LSG_2(83 \times 159) = LSG_2(83 \times 59) = 97$

$LSG_2((83 + 1x10)(59 + 7x10)) = LSG_2(93 \times 129) = LSG_2(93 \times 29) = 97$

$LSG_2((93 + 1x10)(29 + 7x10)) = LSG_2(103 \times 99) = LSG_2(3 \times 99) = 97$ => ซ้ำรูปแบบแรก

ดังนันจึงได้สมาชิกทั้งหมดของแต่ละกลุ่มเป็นดังนี้

กลุ่มที่ 1: (01, 97), (11, 27), (21, 57), (31, 87), (41, 17), (51, 47), (61, 77), (71, 7), (81, 37), (91, 67)

กลุ่มที่ 2: (03, 99), (13, 69), (23, 39), (33, 9), (43, 79), (53, 49), (63, 19), (73, 89), (83, 59), (93, 29)

หลังจากทราบเลข 2 หลักสุดท้ายของคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่จะเป็นตัวประกอบของ n แล้วจะสามารถหารูปแบบของเลข m หลักสุดท้ายของ u ได้จากผลบวกระหว่างคู่ความสัมพันธ์แต่ละคู่ทั้งหมด และหารูปแบบ m หลักสุดท้ายของ v ได้จากผลต่างระหว่างคู่ความสัมพันธ์แต่ละคู่

ทั้งหมดโดยจำเป็นต้องหารูปแบบที่เกิดจากทั้งการนำตัวหน้าลบออกด้วยตัวหลัง และการนำตัวหลังลบออกด้วยตัวหน้าเนื่องจากผลลัพธ์มีค่าไม่เท่ากัน และเพื่อให้ได้มาซึ่งรูปแบบที่เป็นไปได้ทั้งหมด อย่างไรก็ตามกรณีที่ผลลัพธ์ที่ได้มีค่าเป็นลบให้นำผลลัพธ์ดังกล่าวบวกด้วย 10^m (เปรียบเสมือนการยืมค่าจากตำแหน่งถัดไปที่มีค่านัยสำคัญที่สูงกว่า)

ตัวอย่างที่ 8.6 จากเลข 2 หลักสุดท้ายของคู่ความสัมพันธ์ที่เป็นไปได้ทั้งหมดที่มีโอกาสจะเป็นตัวประกอบของ n ที่มีเลข 2 หลักสุดท้ายเป็น 97 ที่หาได้จากตัวอย่างที่ 8.5 จงหารูปแบบเลข 2 หลักสุดท้ายของ u และ v ที่เป็นไปได้ทั้งหมด

วิธีทำ เริ่มพิจารณาเลข 2 หลักสุดท้ายของ u ที่เป็นไปได้ทั้งหมดดังนี้

กรณีที่เป็นไปได้ทั้งหมดของ $LSG_2(u)$ ที่พิจารณาจาก $LSG_2(n) = 97$	
$LSG_2(1 + 97) = 98$	$LSG_2(3 + 99) = 2$
$LSG_2(11 + 27) = 38$	$LSG_2(13 + 69) = 82$
$LSG_2(21 + 57) = 78$	$LSG_2(23 + 39) = 62$
$LSG_2(31 + 87) = 18$	$LSG_2(33 + 9) = 42$
$LSG_2(41 + 17) = 58$	$LSG_2(43 + 79) = 22$
$LSG_2(51 + 47) = 98$	$LSG_2(53 + 49) = 2$
$LSG_2(61 + 77) = 38$	$LSG_2(63 + 19) = 82$
$LSG_2(71 + 7) = 78$	$LSG_2(73 + 89) = 62$
$LSG_2(81 + 37) = 18$	$LSG_2(83 + 59) = 42$
$LSG_2(91 + 67) = 58$	$LSG_2(93 + 29) = 22$

จากตารางข้างต้นแสดงผลลัพธ์ที่เป็นได้ทั้งหมดที่มีโอกาสจะเป็นเลข 2 หลักสุดท้ายของ u โดยหากตัดค่าซ้ำออกและนำค่ามาเรียงจากน้อยไปมากจะได้เลข 2 หลักสุดท้ายของ u มีรูปแบบเป็นดังนี้เสมอ

$$LSG_2(u) = 2, 18, 22, 38, 42, 58, 62, 78, 82 \text{ และ } 98$$

ดังนั้นหากพิจารณาหาค่าของ u แล้วพบว่าเลข 2 หลักสุดท้ายไม่ตรงกับรูปแบบข้างต้นจะถูกตัดทิ้งออกจากการคำนวณ

ขั้นตอนถัดไปพิจารณาเลข 2 หลักสุดท้ายของ v ที่เป็นไปได้ทั้งหมดดังนี้

กรณีที่เป็นไปได้ทั้งหมดของ $LSG_2(v)$ ที่พิจารณาจาก $LSG_2(n) = 97$			
$LSG_2(1–97+10^2) = 4$	$LSG_2(97 – 1) = 96$	$LSG_2(3–99+10^2) = 4$	$LSG_2(99 - 3) = 96$
$LSG_2(11-27+10^2) = 84$	$LSG_2(27 - 11) = 16$	$LSG_2(13-69+10^2) = 44$	$LSG_2(69 - 13) = 56$
$LSG_2(21-57+10^2) = 64$	$LSG_2(57 - 21) = 36$	$LSG_2(23-39+10^2) = 84$	$LSG_2(39 - 23) = 16$
$LSG_2(31-87+10^2) = 44$	$LSG_2(87 - 31) = 56$	$LSG_2(33 - 9) = 24$	$LSG_2(9-33+10^2) = 76$
$LSG_2(41 - 17) = 24$	$LSG_2(17–41+10^2) = 76$	$LSG_2(43-79+10^2) = 64$	$LSG_2(79 - 43) = 36$
$LSG_2(51 - 47) = 4$	$LSG_2(47–51+10^2) = 96$	$LSG_2(53 - 49) = 4$	$LSG_2(49-53+10^2) = 96$
$LSG_2(61-77+10^2) = 84$	$LSG_2(77 - 61) = 16$	$LSG_2(63 - 19) = 44$	$LSG_2(19-63+10^2) = 56$
$LSG_2(71 - 7) = 64$	$LSG_2(7–71+10^2) = 36$	$LSG_2(73-89+10^2) = 84$	$LSG_2(89 - 73) = 16$
$LSG_2(81 - 37) = 44$	$LSG_2(37–81+10^2) = 56$	$LSG_2(83 - 59) = 24$	$LSG_2(59-83+10^2) = 76$
$LSG_2(91 - 67) = 24$	$LSG_2(67–91+10^2) = 76$	$LSG_2(93 - 29) = 64$	$LSG_2(29-93+10^2) = 36$

จากตารางข้างต้นแสดงผลลัพธ์ที่เป็นได้ทั้งหมดที่มีโอกาสจะเป็นเลข 2 หลักสุดท้ายของ v โดยหากตัดค่าซ้ำออกและนำค่ามาเรียงจากน้อยไปมากจะได้เลข 2 หลักสุดท้ายของ v มีรูปแบบเป็นดังนี้เสมอ

$LSG_2(v)$ = 4, 16, 24, 36, 44, 56, 64, 76, 84 และ 96

ดังนั้นหากพิจารณาหาค่าของ v แล้วพบว่าเลข 2 หลักสุดท้ายไม่ตรงกับรูปแบบข้างต้นจะถูกตัดทิ้งออกจากการคำนวณ

ตัวอย่างที่ 8.7 จงแสดงวิธีการแยกตัวประกอบ n = 3596597 โดยใช้ขั้นตอนวิธีของแฟร์มาต์แบบไม่มีการคำนวณรากที่สองที่ผสมผสานกับการพิจารณาเลข 2 หลักสุดท้ายของ n

วิธีทำ เนื่องจาก $LSG_2(3596597) = 97$ ดังนั้น u และ v ที่จะใช้สำหรับการคำนวณจะมีรูปแบบของ $LSG_2(u)$ และ $LSG_2(v)$ เป็นดังตัวอย่างที่ 8.6 เสมอ

เนื่องจากตัวอย่างนี้กำหนดให้ใช้ขั้นตอนวิธีของแฟร์มาต์แบบไม่มีการคำนวณรากที่สองผสมผสานกับการพิจารณาเลข 2 หลักสุดท้ายของ n ดังนั้นจึงต้องเลือกใช้ขั้นตอนวิธีที่ 7.7 ที่มีการปรับปรุงใหม่เล็กน้อยคือตัด u และ v ที่มีรูปแบบไม่ตรงตามเงื่อนไขทิ้ง

1. $u = 2\left\lceil \sqrt{3596597} \right\rceil = 3794$

เนื่องจาก $LSG_2(3794) = 94$ ไม่ตรงตามรูปแบบของ $LSG_2(p+q) = LSG_2(u)$ จึงสามารถเพิ่มค่าได้ โดยค่าที่น้อยที่สุดที่ตรงเงื่อนของ $LSG_2(u)$ และมีค่ามากกว่า 3794 คือ 3798

จากขั้นตอนวิธีที่ 7.7 สังเกตว่าค่าเริ่มต้นของ v คือ 0 เสมอ อย่างไรก็ตาม 0 ไม่ตรงตามรูปของ $LSG_2(p - q) = LSG_2(v)$ จึงสามารถเพิ่มค่าได้ โดยค่าที่น้อยที่สุดที่ตรงเงื่อนของ $LSG_2(v)$ และมีค่ามากกว่า 0 คือ 4 ดังนั้น

2. $v = 4$

3. $r = 3798^2 - 4^2 - 4x3596597 = 38400$

ขั้นตอนที่ 4 – 12 จะเป็นการดำเนินการภายในวงวนดังนี้

เนื่องจาก $r \neq 0$ ดังนั้น

รอบที่ 1:

เนื่องจาก $r > 0$ ดังนั้นจึงใช้ v ค่าใหม่ ($v = 16$)

6. $r = u^2 - (v + 12)^2 - 4n$, $v + 12$ มาจากการเพิ่มค่าจาก 4 เป็น 16 ซึ่งมีส่วนต่างคือ 12

$= u^2 - (v^2 + 24v + 144) - 4n$

$= u^2 - v^2 - 4n - 24v - 144$

$= 38400 + 24x4 - 144$

$= 38160$

7. $v = 16$ หรือ $v = v + 12$, (หากเทียบกับอัตราการเพิ่มค่าแบบเดิม $v = v+2$ สังเกตว่าตัดค่าอื่นๆ ออกได้เป็นจำนวนมาก)

เนื่องจาก $r \neq 0$ ดังนั้น

รอบที่ 2:

เนื่องจาก $r > 0$ ดังนั้นจึงใช้ v ค่าใหม่ ($v = 24$)

6. $r = u^2 - (v + 8)^2 - 4n$, $v + 8$ มาจากการเพิ่มค่าจาก 16 เป็น 24 ซึ่งมีส่วนต่างคือ 8

$= u^2 - (v^2 + 16v + 64) - 4n$

$= u^2 - v^2 - 4n - 16v - 64$

$= 38160 + 16x16 - 64$

$= 37840$

7. $v = 24$ หรือ $v = v + 8$, (หากเทียบกับอัตราการเพิ่มค่าแบบเดิม $v = v+2$ สังเกตว่าตัดค่าอื่นๆ ออกได้เป็นจำนวนมาก)

โดยหากดำเนินการตามขั้นตอนวิธีนี้จะพบคำตอบในรอบที่ 20 ซึ่งได้ $u = 3798$ และ $v = 196$ ดังนั้นจึงได้ $p = \frac{3798+196}{2} = 1997$ และ $q = \frac{3798-196}{2} = 1801$

หากพิจารณาขั้นตอนวิธีที่ 7.7 สามารถพิจารณารอบการคำนวณทั้งหมดได้ดังนี้

จากสมการ (7.18)

$$t_u = \frac{1997+1801}{2} - \left\lceil \sqrt{3596597} \right\rceil$$

$= 1899 - 1897$

$= 2$

และจากสมการ (7.19)

$$t_v = \frac{1997-1801}{2}$$

$= 98$

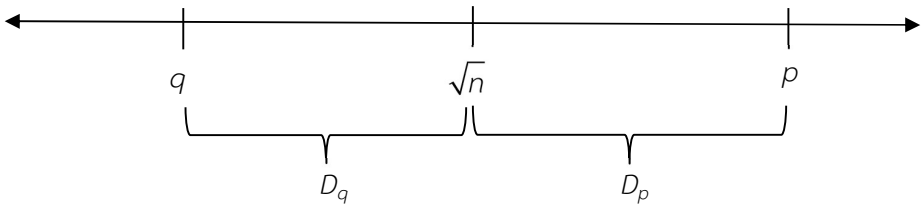
ดังนั้นรอบการคำนวณทั้งหมดคือ $2 + 98 = 100$ ซึ่งเป็นจำนวนที่สูงมากหากเปรียบเทียบกับการใช้ขั้นตอนวิธีของแฟร์มาต์แบบไม่มีการคำนวณรากที่สองที่ผสมผสานกับการพิจารณาเลข 2 หลักสุดท้ายของ n ซึ่งจากตัวอย่างที่ 8.7 ใช้รอบการคำนวณเพียง 20 รอบ

นอกเหนือจากนั้นสามารถใช้รูปแบบของ u และ v จากตัวอย่างที่ 8.6 เพื่อใช้เป็นเงื่อนไขแยกตัวประกอบ n ที่มีเลขสองหลักสุดท้ายมีค่าเป็น 97 ได้ในทุกกรณีโดยไม่จำเป็นต้องเสียเวลาในการพิจารณาใหม่อีก

จากตัวอย่างที่ 8.6 และ 8.7 บอกเป็นนัยได้ว่าหากพิจารณาเลขกลุ่มสุดท้ายของ n จำนวนมากขึ้นจะช่วยให้ทราบรูปแบบของ u และ v ได้ละเอียดมากยิ่งขึ้นซึ่งจะช่วยให้สามารถตัดค่าที่ไม่เกี่ยวข้องออกจากการคำนวณได้เป็นจำนวนมากยิ่งขึ้นตามไปด้วย

9. การประมาณค่าเริ่มต้นสำหรับตัวประกอบที่มีจำนวนบิตแตกต่างกัน

ในปี ค.ศ. 2014 กลุ่มนักวิจัยชาวไต้หวัน 3 ท่านประกอบด้วย มูเอ็น วู (Mu-En Wu), เรย์ลิน โทส (Raylin Tso) และ ฮุนมิน ซัน (Hung-Min Sun) [16] ได้เสนอวิธีการประมาณค่าเริ่มต้นของ u และ v ใหม่ เรียกว่า Estimated Prime Factor (EPF) ซึ่งโดยทั่วไปค่าเริ่มต้นของ u และ v มีค่าเป็น $2\left\lceil \sqrt{n} \right\rceil$ และ 0 ตามลำดับ แต่หากดำเนินการตามขั้นตอนวิธีที่ถูกนำเสนอโดยนักวิจัย 3 ท่านนี้แล้วค่าเริ่มต้นของทั้งสองค่าจะมีค่าที่สูงขึ้น ดังนั้นจึงมีค่าที่ใกล้เคียงค่าเป้าหมายมากขึ้นส่งผลให้รอบการคำนวณลดลง ถึงแม้ว่าวิธีดังกล่าวนี้สามารถใช้ได้กับ n ที่เกิดจากตัวประกอบที่มีจำนวนบิตที่แตกต่างกันได้ทั้งหมด แต่ใช้ได้กับตัวประกอบที่มีจำนวนบิตที่เท่ากันบางค่าซึ่งเป็นส่วนน้อยเท่านั้น



รูปที่ 8.1 เส้นจำนวนสำหรับพารามิเตอร์ที่เกี่ยวข้องกับมอดูลัส

รูปที่ 8.1 แสดงเส้นจำนวนสำหรับพารามิเตอร์ที่เกี่ยวข้องกับ n โดยกำหนดให้ $p = \sqrt{n} + D_p$ และ $q = \sqrt{n} - D_q$ จึงสามารถจัดรูปแบบสมการของค่า n ใหม่ได้ดังนี้

จาก
$$\begin{aligned} n &= pq \\ &= (\sqrt{n} + D_p)(\sqrt{n} - D_q) \\ &= n + D_p\sqrt{n} - D_q\sqrt{n} - D_pD_q \\ &= n + \sqrt{n}\,(D_p - D_q) - D_pD_q \\ 0 &= \sqrt{n}\,(D_p - D_q) - D_pD_q \\ D_pD_q &= \sqrt{n}\,(D_p - D_q) \end{aligned}$$

ดังนั้น
$$\frac{1}{\sqrt{n}} = \frac{D_p - D_q}{D_pD_q} \tag{8.9}$$

จากสมการที่ (8.9) ดำเนินการประมาณค่าเศษส่วนต่อเนื่องของ $\frac{1}{\sqrt{n}}$ ซึ่งในแต่ละรอบจะได้ผลลัพธ์เป็น $\frac{h_0}{k_0}, \frac{h_1}{k_1}, \frac{h_2}{k_2}, \cdots$ จนกระทั่งพบ $\frac{h_{t+1}}{k_{t+1}}$ ซึ่ง k_{t+1} เป็นค่าแรกที่มีค่ามากกว่า n จึงย้อนกลับไปเลือกใช้งานค่า $\frac{h_t}{k_t}$ ดังนั้นได้ว่า

$$\frac{1}{\sqrt{n}} = \frac{D_p - D_q}{D_p D_q} \approx \frac{h_t}{k_t} \tag{8.10}$$

ดังนั้น $h_t \approx D_p - D_q$ และ $k_t \approx D_p D_q$

จาก
$$\begin{aligned} u &= p + q \\ &= \sqrt{n} + D_p + \sqrt{n} - D_q \\ &= D_p - D_q + 2\sqrt{n} \end{aligned}$$

เนื่องจาก $u \in \mathbb{Z}^+$ และ $D_p - D_q > h_t$ ดังนั้น

$$u > h_t + 2\sqrt{n}$$

ดังนั้นสามารถกำหนดค่าเริ่มต้น, $u = \left\lceil h_t + 2\sqrt{n} \right\rceil$ (8.11)

และจาก
$$\begin{aligned} v &= p - q \\ &= \sqrt{n} + D_p - \sqrt{n} + D_q \\ &= D_p + D_q \end{aligned}$$

เนื่องจาก
$$\begin{aligned} (D_p + D_q)^2 &= D_p^2 + 2D_pD_q + D_q^2 \\ &= D_p^2 + 2D_pD_q + D_q^2 + 2D_pD_q - 2D_pD_q \\ &= D_p^2 - 2D_pD_q + D_q^2 + 2D_pD_q + 2D_pD_q \\ &= (D_p - D_q)^2 + 4D_pD_q \end{aligned}$$

$$\begin{aligned} D_p + D_q &= \sqrt{(D_p - D_q)^2 + 4D_pD_q} \\ &= \sqrt{h_t^2 + 4k_t} \end{aligned}$$

เนื่องจาก $v \in \mathbb{Z}^+$ ดังนั้นสามารถกำหนดค่าเริ่มต้นของ v ได้เป็น

$$v = \left\lfloor \sqrt{h_t^2 + 4k_t} \right\rfloor \tag{8.12}$$

จากสมการที่ (8.11) และ (8.12) ค่าเริ่มต้นของ u และ v มีค่าที่สูงขึ้น ดังนั้นจึงสามารถลดจำนวนรอบของการคำนวณได้มากยิ่งขึ้น

ตัวอย่างที่ 8.8 จงใช้ EPF ประมาณค่าเริ่มต้นของ n = 10276297

วิธีทำ เริ่มจากการประมาณค่าเศษส่วนต่อเนื่องของ $\frac{1}{\sqrt{10276297}} = \frac{1}{3205.666389379906}$ ดังนี้

รอบที่ 1:

ลำดับที่ 1: $a_0 = \left\lfloor 3205.666389379906 \right\rfloor = 3205$

ลำดับที่ 2: $x = 3205.666389379906 - 3205 = 0.666389379906$

ลำดับที่ 3: $y = \frac{1}{0.666389379906} \approx 1.500624154816301$

ลำดับที่ 4: $a_1 = \left\lfloor 1.500624154816301 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_0 = a_0 h_{(-1)} + h_{(-2)}$$
$$= 3205 \times 0 + 1 = 1$$
$$k_0 = a_0 k_{(-1)} + k_{(-2)}$$
$$= 3205 \times 1 + 0 = 3205$$

รอบที่ 2:

ลำดับที่ 1: $x = y - a_1$

$$= 1.500624154816301 - 1 = 0.500624154816301$$

ลำดับที่ 2: $y = \frac{1}{0.500624154816301}$

$$\approx 1.9975064934031$$

ลำดับที่ 3: $a_2 = \left\lfloor 1.9975064934031 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_1 = a_1 h_0 + h_{(-1)}$$
$$= 1 \times 1 + 0 = 1$$
$$k_1 = a_1 k_0 + k_{(-1)}$$
$$= 1 \times 3205 + 1 = 3206$$

รอบที่ 3:

ลำดับที่ 1: $x = y - a_2$

$= 1.9975064934031 - 1 = 0.9975064934031$

ลำดับที่ 2: $y = \dfrac{1}{0.9975064934031}$

$\approx 1.002499739714368$

ลำดับที่ 3: $a_3 = \left\lfloor 1.002499739714368 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_2 = a_2h_1 + h_0$

$= 1x1 + 1 = 2$

$k_2 = a_2k_1 + k_0$

$= 1x3206 + 3205 = 6411$

รอบที่ 4:

ลำดับที่ 1: $x = y - a_3$

$= 1.002499739714368 - 1 = 0.002499739714368$

ลำดับที่ 2: $y = \dfrac{1}{0.002499739714368}$

$\approx 400.0416500374825$

ลำดับที่ 3: $a_4 = \left\lfloor 400.0416500374825 \right\rfloor = 400$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_3 = a_3h_2 + h_1$

$= 1x2 + 1 = 3$

$k_3 = a_3k_2 + k_1$

$= 1x6411 + 3206 = 9617$

รอบที่ 5:

ลำดับที่ 1: $x = y - a_4$

$= 400.0416500374825 - 400 = 0.0416500374825$

ลำดับที่ 2: $y = \dfrac{1}{0.0416500374825}$

$\approx 24.00958223435376$

ลำดับที่ 3: $a_5 = \left\lfloor 24.00958223435376 \right\rfloor = 24$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_4 = a_4h_3 + h_2$

$= 400x3 + 2 = 1202$

$k_4 = a_4k_3 + k_2$

$= 400x9617 + 6411 = 3853211$

รอบที่ 6:

ลำดับที่ 1: $x = y - a_5$

$= 24.00958223435376 - 24 = 0.00958223435376$

ลำดับที่ 2: $y = \frac{1}{0.00958223435376}$

≈ 104.359793664158

ลำดับที่ 3: $a_6 = \left\lfloor 104.359793664158 \right\rfloor = 104$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_5 = a_5h_4 + h_3$

$= 24x1202 + 3 = 28851$

$k_5 = a_5k_4 + k_3$

$= 24x3853211 + 9617 = 92486681$

ดังนั้น $\frac{1}{3205.666389379906} \approx \frac{1}{3205} \approx \frac{1}{3206} \approx \frac{2}{6411} \approx \frac{3}{9617} \approx \frac{1202}{3853211} \approx \frac{28851}{92486681}$

เนื่องจาก $92486681 > 10276297$ ดังนั้น $\frac{h_t}{k_t} = \frac{1202}{3853211}$ หรือ

$h_t = 1202$ และ $k_t = 3853211$

จากสมการ (8.11) ได้ว่า

$$u = \left\lceil 1202 + 2\sqrt{10276297} \right\rceil$$
$$= 7614$$

และจากสมการ (8.12) ได้ว่า

$$v = \left\lfloor \sqrt{1202^2 + 4x43853211} \right\rfloor$$
$$= 4105$$

เนื่องจาก v เป็นจำนวนเต็มบวกคู่เสมอ ดังนั้นสามารถปรับ v เป็น 4106 ได้

เนื่องจาก $2\left\lceil \sqrt{10276297} \right\rceil = 6412$ สามารถลดจำนวนรอบการคำนวณของ u ได้ $\frac{7614 - 6412}{2} = 601$ รอบ และลดจำนวนรอบการคำนวณของ v ได้ $\frac{4106}{2} = 2053$ รอบ ดังนั้นสรุปได้ว่า EPF สามารถลดรอบการคำนวณค่า $n = 10276297 = 1069 \times 9613$ ได้สูงถึง 2053 + 601 = 2654 รอบ

โดยจากตัวอย่างที่ 8.8 นี้ ค่า u และ v ที่แท้จริงคือ 10682 และ 8544 ตามลำดับ

ข้อเสียของ EPF คือจะไม่สามารถถูกนำมาใช้ประมาณค่าเริ่มต้นของ u และ v ได้ ในกรณีที่จำนวนบิตของ p และ q เท่ากัน โดยเฉพาะอย่างยิ่งหากจำนวนเต็มทั้งสองมีค่าที่ใกล้เคียงกัน ซึ่งมีโอกาสเกิดข้อผิดพลาดที่สูงเนื่องจากค่าเริ่มต้นที่ถูกประมาณขึ้นมาใหม่มีค่าสูงกว่าค่าที่แท้จริง และจากขั้นตอนวิธีของแฟร์มาต์แบบไม่คำนวณค่ารากที่สองนั้นสังเกตได้ว่าไม่มีโอกาสที่ u และ v จะถูกลดค่า ดังนั้นจึงเป็นไปไม่ได้ที่จะพบค่าเป้าหมาย ดังตัวอย่างที่ 8.9

ตัวอย่างที่ 8.9 แสดงข้อผิดพลาดจากการ EPF ประมาณค่าเริ่มต้นของ $n = 3062797 = 2027 \times 1511$ (p และ q มีขนาดที่เท่ากันคือ 11 บิต)

วิธีทำ เริ่มจากการประมาณค่าเศษส่วนต่อเนื่องของ

$$\frac{1}{\sqrt{3062797}} = \frac{1}{1750.084855085604127747474719742411}$$ ดังนี้

รอบที่ 1:

ลำดับที่ 1: $a_0 = \left\lfloor 1750.084855085604127747474719742411 \right\rfloor = 1750$

ลำดับที่ 2: $x = 1750.084855085604127747474719742411 - 1750$

$$= 0.084855085604127747474719742411$$

ลำดับที่ 3: $y = \frac{1}{0.084855085604127747474719742411}$

$$\approx 11.784797491870720968846706983 82$$

ลำดับที่ 4: $a_1 = \left\lfloor 11.784797491870720968846706983 82 \right\rfloor = 11$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_0 = a_0 h_{(-1)} + h_{(-2)}$$
$$= 1750x0 + 1 = 1$$
$$k_0 = a_0 k_{(-1)} + k_{(-2)}$$
$$= 1750x1 + 0 = 1750$$

รอบที่ 2:

ลำดับที่ 1: $x = y - a_1$

$$= 11.784797491870720968846706983 82 - 11$$
$$= 0.784797491870720968846706983 82$$

ลำดับที่ 2: $y = \dfrac{1}{0.784797491870720968846706983 82}$

$$\approx 1.274214062045867444519294842 0022$$

ลำดับที่ 3: $a_2 = \left\lfloor 1.274214062045867444519294842 0022 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_1 = a_1 h_0 + h_{(-1)}$$
$$= 11x1 + 0 = 11$$
$$k_1 = a_1 k_0 + k_{(-1)}$$
$$= 11x1750 + 1 = 19521$$

รอบที่ 3:

ลำดับที่ 1: $x = y - a_2$

$$= 1.274214062045867444519294842 0022 - 1$$
$$= 0.274214062045867444519294842 0022$$

ลำดับที่ 2: $y = \dfrac{1}{0.274214062045867444519294842 0022}$

$$\approx 3.646785990985142278679885230 3065$$

ลำดับที่ 3: $a_3 = \left\lfloor 3.646785990985142278679885230 3065 \right\rfloor = 3$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_2 = a_2h_1 + h_0$

$= 1x11 + 1 = 12$

$k_2 = a_2k_1 + k_0$

$= 1x19521 + 1750 = 21271$

รอบที่ 4:

ลำดับที่ 1: $x = y - a_3$

$= 3.646785990985142278679885230306 5 - 3$

$= 0.646785990985142278679885230306 5$

ลำดับที่ 2: $y = \frac{1}{0.646785990985142278679885230306 5}$

$\approx 1.546106461701288794517484297458 3$

ลำดับที่ 3: $a_4 = \left\lfloor 1.546106461701288794517484297458 3 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_3 = a_3h_2 + h_1$

$= 3x12 + 11 = 47$

$k_3 = a_3k_2 + k_1$

$= 3x21271 + 19521 = 83334$

รอบที่ 5:

ลำดับที่ 1: $x = y - a_4$

$= 1.546106461701288794517484297458 3 - 1$

$= 0.546106461701288794517484297458 3$

ลำดับที่ 2: $y = \frac{1}{0.546106461701288794517484297458 3}$

$\approx 1.831144786100303399209813767346 2$

ลำดับที่ 3: $a_5 = \left\lfloor 1.831144786100303399209813767346 2 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_4 = a_4h_3 + h_2$

$= 1x47 + 12 = 59$

$k_4 = a_4k_3 + k_2$

$= 1x83334 + 21271 = 104605$

รอบที่ 6:

ลำดับที่ 1: $x = y - a_5$

$= 1.831144786100303399209813767346 2 - 1$

$= 0.831144786100303399209813767346 2$

ลำดับที่ 2: $y = \frac{1}{0.831144786100303399209813767346 2}$

$\approx 1.203159806478433447836838638689 3$

ลำดับที่ 3: $a_6 = \left\lfloor 1.203159806478433447836838638689 3 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_5 = a_5h_4 + h_3$

$= 1x59 + 47 = 106$

$k_5 = a_5k_4 + k_3$

$= 1x104605 + 83334 = 187939$

รอบที่ 7:

ลำดับที่ 1: $x = y - a_6$

$= 1.203159806478433447836838638689 3 - 1$

$= 0.203159806478433447836838638689 3$

ลำดับที่ 2: $y = \frac{1}{0.203159806478433447836838638689 3}$

$\approx 4.922233473903981125194708069555 1$

ลำดับที่ 3: $a_7 = \left\lfloor 4.922233473903981125194708069555 1 \right\rfloor = 4$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_6 = a_6h_5 + h_4$

$= 1x106 + 59 = 165$

$k_6 = a_6k_5 + k_4$

$= 1x187939 + 104605 = 292544$

รอบที่ 8:

ลำดับที่ 1: $x = y - a_7$

$= 4.92223347390398112519470806955551 - 4$

$= 0.92223347390398112519470806955551$

ลำดับที่ 2: $y = \dfrac{1}{0.92223347390398112519470806955551}$

$\approx 1.08432411997237435118732825361 32$

ลำดับที่ 3: $a_8 = \left\lfloor 1.08432411997237435118732825361 32 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_7 = a_7h_6 + h_5$

$= 4x165 + 106 = 766$

$k_7 = a_7k_6 + k_5$

$= 4x292544 + 187939= 1358115$

รอบที่ 9:

ลำดับที่ 1: $x = y - a_8$

$= 1.08432411997237435118732825361 32 - 1$

$= 0.08432411997237435118732825361 32$

ลำดับที่ 2: $y = \dfrac{1}{0.08432411997237435118732825361 32}$

$\approx 11.859003098136246839664596196231$

ลำดับที่ 3: $a_9 = \left\lfloor 11.859003098136246839664596196231 \right\rfloor = 11$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$h_8 = a_8h_7 + h_6$

$= 1x766 + 165 = 931$

$k_8 = a_8k_7 + k_6$

$= 1x1358115 + 292544= 1650659$

รอบที่ 10:

ลำดับที่ 1: $x = y - a_9$

$= 11.859003098136246839664596196231 - 11$

$$= 0.859003098136246839664596196231$$

ลำดับที่ 2: $y = \dfrac{1}{0.859003098136246839664596196231}$

$$\approx 1.164140155221407109435629552 4878$$

ลำดับที่ 3: $a_{10} = \left\lfloor 1.164140155221407109435629552 4878 \right\rfloor = 1$

จากสมการที่ (5.6) และ (5.7) ได้ว่า

$$h_9 = a_9h_8 + h_7$$

$$= 11x931 + 766 = 11007$$

$$k_9 = a_9k_8 + k_7$$

$$= 11x1650659 + 1358115= 19515364$$

ดังนั้น $\dfrac{1}{1750.084855085604127747 4719742411} \approx \dfrac{1}{1750} \approx \dfrac{11}{19521} \approx \dfrac{12}{21271} \approx \dfrac{47}{83334} \approx \dfrac{59}{104605} \approx \dfrac{106}{1879393} \approx \dfrac{165}{292544} \approx \dfrac{766}{1358115} \approx \dfrac{931}{1650659} \approx \dfrac{11007}{19515369}$

เนื่องจาก 19515369 > 3062797 ดังนั้น $\dfrac{h_t}{k_t} = \dfrac{931}{1650659}$ หรือ $h_t = 931$ และ $k_t = 1650659$

จากสมการ (8.11) ได้ว่า

$$u = \left\lceil 931 + 2\sqrt{3062797} \right\rceil$$

$$= 4432$$

แต่เนื่องจากค่า u ที่แท้จริงคือ $u = 2027 + 1511 = 3538$ ซึ่งสังเกตได้ว่าการใช้ EPF เพื่อประมาณค่าเริ่มต้นของ u มีค่ามากกว่าค่า u ที่แท้จริง ดังนั้นจึงไม่มีโอกาสที่จะพบค่าดังกล่าวได้อย่างแน่นอนส่งผลให้เกิดเป็นข้อผิดพลาด

นอกเหนือจากนั้นหากทดลองประมาณค่า v จากสมการ (8.12) ได้ว่า

$$v = \left\lfloor \sqrt{931^2 + 4x1650659} \right\rfloor$$

$$= 2733$$

แต่เนื่องจากค่า v ที่แท้จริงคือ $v = 2027 - 1511 = 516$ ซึ่งสังเกตได้ว่าการใช้ EPF เพื่อประมาณค่าเริ่มต้นซึ่งมีค่ามากกว่าค่าที่แท้จริง ดังนั้นจึงไม่มีโอกาสที่จะพบค่าดังกล่าวได้อย่างแน่นอน ส่งผลให้เกิดเป็นข้อผิดพลาดขึ้นเช่นกัน

10. การประมาณค่าเริ่มต้นรูปแบบใหม่ที่สามารถใช้ได้กับค่ามอดูลัสทุกกรณี

จากปัญหาของการประมาณค่าเริ่มต้นของ u และ v ด้วย EPF ซึ่งมีโอกาสสูงที่จะเกิดข้อผิดพลาดในกรณีที่ p และ q มีขนาดที่เท่ากัน ดังนั้นผู้เขียนจึงได้เสนอขั้นตอนวิธีที่ใช้สำหรับการกำหนดค่าเริ่มต้นของ u และ v ใหม่ที่สามารถใช้ได้กับค่า n ทุกค่าโดยไม่มีข้อผิดพลาด [35]

ทฤษฎีบทที่ 8.6 กำหนดให้ p และ q คือจำนวนเฉพาะที่เป็นตัวประกอบของ n โดยที่ $p > q$ แล้ว ผลลัพธ์ของ $\sqrt{p} + \sqrt{q} > 2\sqrt[4]{n}$

พิสูจน์ หาก $p = q = \sqrt{n}$ จะได้ว่า $p + q = 2\sqrt{n}$

จากทฤษฎีกำหนดให้ $p > q$, ดังนั้น $q < \sqrt{n} < p$ ซึ่งต้องกำหนดค่า p และ q ใหม่

กำหนดให้ $p = \sqrt{n} + x$ และ $q = \sqrt{n} - x$ เมื่อ $x \in \mathbb{Z}^+$ จะได้ว่า

$$\begin{aligned} pq &= (\sqrt{n} + x)(\sqrt{n} - x) \\ &= n + x\sqrt{n} - x\sqrt{n} - x^2 \\ &= n - x^2 \neq n \end{aligned}$$

ดังนั้นจึงไม่สามารถกำหนดค่าดังข้างต้นได้

กำหนดค่าใหม่ดังนี้ $p = \sqrt{n} + a$ และ $q = \sqrt{n} - b$ เมื่อ $a, b \in \mathbb{Z}^+$ จะได้ว่า

$$\begin{aligned} pq &= (\sqrt{n} + a)(\sqrt{n} - b) \\ &= n + a\sqrt{n} - b\sqrt{n} - ab \end{aligned}$$

จากสมการข้างต้นมีความเป็นไปได้ที่ $n = pq$ หาก $a\sqrt{n} - b\sqrt{n} - ab = 0$ ซึ่งสังเกตได้ว่า a จะต้องมีค่ามากกว่า b เสมอ

ดังนั้น
$$\begin{aligned} p + q &= (\sqrt{n} + a) + (\sqrt{n} - b) \\ &= 2\sqrt{n} + a - b \end{aligned}$$

เนื่องจาก $a > b$, $p + q > 2\sqrt{n}$ เสมอ

จาก
$$(\sqrt{p} + \sqrt{q})^2 = p + 2\sqrt{p}\sqrt{q} + q$$

$$= p + q + 2\sqrt{n}$$

$$\sqrt{p} + \sqrt{q} = \sqrt{p+q+2\sqrt{n}}$$

เนื่องจาก $p + q = 2\sqrt{n} + a - b$

ดังนั้น $$\sqrt{p} + \sqrt{q} = \sqrt{2\sqrt{n}+a-b+2\sqrt{n}}$$

$$= \sqrt{4\sqrt{n}+a-b}$$

เนื่องจาก $a > b$, ดังนั้น $\sqrt{p} + \sqrt{q} > \sqrt{4\sqrt{n}}$

หรือ $$\sqrt{p} + \sqrt{q} > 2\sqrt[4]{n}$$ □

ทฤษฎีบทที่ 8.7 กำหนดให้ p และ q คือจำนวนเฉพาะที่เป็นตัวประกอบของ n โดยที่ $p > q$, U แทนรูปแบบของสมาชิกที่เป็นไปได้ทั้งหมดของ $LSG_m(p + q)$ และ d แทนผลต่างระหว่าง $LSG_m(2\lceil\sqrt{n}\rceil)$ และสมาชิกค่าหนึ่งของ U ที่มีค่าน้อยที่สุดและมีค่ามากกว่า $LSG_m(2\lceil\sqrt{n}\rceil)$ หรือเป็นค่าที่น้อยที่สุดในกรณีที่ $LSG_m(2\lceil\sqrt{n}\rceil)$ มีค่ามากกว่าสมาชิกทุกค่าใน U แล้วสามารถประมาณค่าเริ่มต้นใหม่ของ v ได้เป็น $2\left\lceil\sqrt[4]{d^2n}\right\rceil$

พิสูจน์ เนื่องจาก $p > q$ ดังนั้น

$$(\sqrt{p} - \sqrt{q}) > 1$$

$$(\sqrt{p} - \sqrt{q})(\sqrt{p} + \sqrt{q}) > (\sqrt{p} + \sqrt{q})$$

หรือ $$\sqrt{p}^2 - \sqrt{q}^2 > (\sqrt{p} + \sqrt{q})$$

ดังนั้น $$p - q > (\sqrt{p} + \sqrt{q})$$

จากทฤษฎีบทที่ 8.6, $p - q > 2\sqrt[4]{n}$

เนื่องจากเป็นไปได้ที่ค่าเริ่มต้นของ $u = 2\lceil\sqrt{n}\rceil$ จะมีผลลัพธ์ของ $LSG_m(2\lceil\sqrt{n}\rceil)$ ไม่ตรงกับสมาชิกใดๆใน U จึงสามารถเพิ่มค่าขึ้นจนกระทั่งเลข m หลักสุดท้ายของผลลัพธ์ใหม่ตรงกับสมาชิกค่าหนึ่งใน U ซึ่งจากทฤษฎีกำหนดให้ระยะห่างระหว่างค่าดังกล่าวและ $LSG_m(2\lceil\sqrt{n}\rceil)$ คือ d อย่างไรก็ตามค่าเริ่มต้นค่าใหม่นี้มีค่าน้อยกว่าหรือเท่ากับ $p + q$ เสมอ จึงได้ว่า

$$p + q - (2\left\lceil \sqrt{n} \right\rceil + d) \geq 0$$

$$p + q - 2\left\lceil \sqrt{n} \right\rceil - d \geq 0$$

$$p + q - 2\left\lceil \sqrt{n} \right\rceil \geq d$$

ได้ว่า $p + q - 2\sqrt{n} > d$

$$(\sqrt{p} - \sqrt{q})^2 > d$$

ดังนั้น $\sqrt{p} - \sqrt{q} > \sqrt{d}$

$$(\sqrt{p} - \sqrt{q})(\sqrt{p} + \sqrt{q}) > \sqrt{d}$$

$$p - q > \sqrt{d}(\sqrt{p} + \sqrt{q})$$

$$> 2\sqrt{d}\sqrt[4]{n}$$

เนื่องจาก $v = p - q$ เป็นจำนวนเต็มเสมอ ดังนั้น

$$p - q \geq 2\left\lceil \sqrt[4]{d^2 n} \right\rceil$$

หรือกล่าวอีกนัยหนึ่งคือสามารถกำหนดค่าเริ่มต้นใหม่ให้ $v = 2\left\lceil \sqrt[4]{d^2 n} \right\rceil$ □

นอกเหนือจากนั้นหากผลลัพธ์ของ $LSG_m(2\left\lceil \sqrt[4]{d^2 n} \right\rceil)$ มีค่าไม่ตรงรูปแบบที่เป็นไปได้ทั้งหมดของ $LSG_m(p - q)$ สามารถเพิ่มค่าขึ้นได้จนกว่าจะพบค่าแรกที่ตรงตามรูปแบบ

ตัวอย่างที่ 8.10 จงประมาณค่าเริ่มต้นของ v จาก n = 10276297 โดยการพิจารณาเลข 2 หลักสุดท้ายของ n

วิธีทำ เริ่มจากคำนวณค่าเริ่มต้นของ $u = 2\left\lceil \sqrt{10276297} \right\rceil = 6412$

เนื่องจาก $LSG_2(6412) = 12$ ไม่ตรงตามรูปแบบของ $LSG_2(p + q) = LSG_2(u)$ จึงสามารถเพิ่มค่าได้ โดยค่าที่น้อยที่สุดที่ตรงเงื่อนของ $LSG_2(u)$ และมีค่ามากกว่าค่าเริ่มต้นของ u คือ 6418

จึงได้ว่า $d = 6418 - 6412 = 6$

จาก $v = 2\left\lceil \sqrt[4]{6^2 \text{x} 10276297} \right\rceil$

$= 278$

เนื่องจาก $LSG_2(278) = 78$ ไม่ตรงกับรูปแบบของ $LSG_2(p - q) = LSG_2(v)$ จึงสามารถเพิ่มค่าขึ้นซึ่งค่าแรกที่พบว่าตรงกับรูปแบบหลังจากเพิ่มค่าคือ 284

ดังนั้นจึงสามารถกำหนดค่าเริ่มต้นให้ v ใหม่ได้เป็น 284

หากนำไปเปรียบเทียบกับการประมาณค่า v จากตัวอย่างที่ 8.8 (ซึ่งประมาณจากค่า n เดียวกัน) พบว่า EPF สามารถประมาณค่าได้ใกล้เคียงมากกว่าขั้นตอนวิธีดังกล่าวนี้ แต่อย่างไรก็ตาม จุดเด่นของขั้นตอนวิธีที่ผู้เขียนนำเสนอนี้คือ นอกเหนือจากสามารถประมาณค่าได้ใหม่แล้ว ยังสามารถนำขั้นตอนวิธีนี้ไปประยุกต์ใช้ได้กับค่า n ในทุกกรณี ข้อเสียของขั้นตอนวิธีนี้คือหาก d มีค่าเป็น 0 จะส่งผลให้ $v = 0$ ซึ่งไม่ได้ช่วยให้ค่าเริ่มต้นของ v มีค่าใกล้เคียงค่าเป้าหมายได้มากขึ้น อย่างไรก็ตาม ยังคงสามารถคำนวณหา p และ q ได้โดยไม่มีข้อผิดพลาดเกิดขึ้นเพียงแต่ไม่สามารถลดรอบการคำนวณลงได้

ตัวอย่างที่ 8.11 จงประมาณค่าเริ่มต้นของ v จาก $n = 3062797$ โดยการพิจารณาเลข 2 หลักสุดท้ายของ n

วิธีทำ เริ่มจากคำนวณค่าเริ่มต้นของ $u = 2\left\lceil \sqrt{3062797} \right\rceil = 3502$

เนื่องจาก $LSG_2(3502) = 02$ ซึ่งตรงกับรูปแบบที่เป็นไปได้ของ $LSG_2(p + q) = LSG_2(u)$ ดังนั้นจึงไม่สามารถปรับค่าเริ่มต้นของ u ได้ ส่งผลให้ $d = 0$ ดังนั้น $v = \left\lceil \sqrt[4]{0^2 \text{x} 3062797} \right\rceil = 0$

อย่างไรก็ตามหากเปรียบเทียบผลลัพธ์ที่ได้นี้กับผลลัพธ์จากตัวอย่างที่ 8.9 ซึ่งเป็นการประมาณค่าโดยใช้ EPF พบว่าการใช้ EPF สำหรับประมาณค่าเริ่มต้นของ $n = 3062797$ จะเกิดข้อผิดพลาดอย่างแน่นอน แต่ขั้นตอนวิธีที่ผู้เขียนนำเสนอนี้จะยังคงดำเนินการหาค่าตัวประกอบต่อได้ถึงแม้ว่าไม่สามารถลดรอบการคำนวณลงได้

นอกเหนือจากการนำหลักการประมาณค่าเริ่มต้นใหม่ของ v นี้ไปประยุกต์ใช้กับขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์แล้วยังสามารถนำผลลัพธ์นี้ไปประยุกต์ใช้กับขั้นตอนวิธีการแยกตัวประกอบประเภทอื่น ๆ บางประเภทได้เช่นกัน โดยมีเป้าหมายเดียวกันคือเพื่อใช้สำหรับการประมาณค่าเริ่มต้นใหม่ที่มีค่าใกล้เคียงกับค่าเป้าหมาย โดยจะยกตัวอย่างการนำผลลัพธ์ที่เป็นค่าเริ่มต้นใหม่ของ v ไปประยุกต์ใช้กับขั้นตอนวิธีการทดลองหารแบบปรับค่าลง

โดยทั่วไปหากนำขั้นตอนวิธีการทดลองหารแบบปรับค่าลง (ขั้นตอนวิธีที่ 7.1) มาใช้สำหรับคำนวณหาจำนวนประกอบของจำนวนเต็ม ตัวหารเริ่มต้นจะมีค่าเท่ากับจำนวนเต็มบวกคี่มากที่สุดที่มีค่าน้อยกว่าหรือเท่ากับ $\left\lfloor \sqrt{n} \right\rfloor$ เสมอ

กำหนดให้ t_d แทนจำนวนรอบของการทดลองหารโดยใช้ขั้นตอนวิธีที่ 7.1

n_d แทนจำนวนเต็มบวกคี่มากที่สุดที่มีค่าน้อยกว่าหรือเท่ากับ $\left\lfloor \sqrt{n} \right\rfloor$

ได้ว่า

$$t_d = \frac{n_d - q}{2} \tag{8.13}$$

อย่างไรก็ตามหากนำผลลัพธ์ที่เป็นค่าเริ่มต้นใหม่ของ u และ v มาประยุกต์ใช้จะสามารถประมาณค่าของตัวหารเริ่มต้นใหม่สำหรับขั้นตอนวิธีทดลองหารแบบปรับค่าลงที่มีค่าใกล้เคียง q มากกว่า n_d [47] ดังนี้

กำหนดให้ u_i และ v_i คือค่าเริ่มต้นใหม่ของ u และ v ตามลำดับโดยทั้งสองค่านี้คำนวณได้จากทฤษฎีบทที่ 8.7 ได้ว่า $\frac{u_i+v_i}{2}$ มีค่าน้อยกว่าหรือเท่ากับ $p = \frac{u+v}{2}$ เสมอเนื่องจาก $u_i \leq u$ และ $v_i \leq v$ ดังนั้นหากกำหนดให้ $p_i = \frac{u_i+v_i}{2}$ จะได้ว่า $n_d \leq p_i \leq p$ หรือ $q \leq \frac{n}{p_i} \leq n_d$

กำหนดให้ q_i คือจำนวนเต็มบวกคี่มากที่สุดที่มีค่าน้อยกว่าหรือเท่ากับ $\left\lfloor \frac{n}{p_i} \right\rfloor$ จำนวนรอบของการทดลองหารแบบปรับค่าลงเมื่อใช้ q_i เป็นตัวหารเริ่มต้นมีค่าเป็นดังนี้

$$t_{d\_new} = \frac{q_i - q}{2} \tag{8.14}$$

เมื่อ t_{d_new} คือจำนวนรอบของการทดลองหารใหม่

ตัวอย่างที่ 8.12 จากผลลัพธ์ของ u_i และ v_i ที่ได้จากตัวอย่างที่ 8.10 จงหาตัวหารเริ่มต้นใหม่สำหรับขั้นตอนวิธีการทดลองหารแบบปรับค่าลงพร้อมเปรียบเทียบผลลัพธ์กับตัวหารเริ่มต้นเดิม

วิธีทำ จากตัวอย่างได้ 8.10 ได้ $u_i = 6418$ และ $v_i = 284$

ได้ว่า $$p_i = \frac{6418+284}{2}$$

$$= 3351$$

และ $$\left\lfloor \frac{10276297}{3351} \right\rfloor = 3066$$

ดังนั้น $q_i = 3065$ เนื่องจากจำนวนเฉพาะที่ไม่เท่ากับ 5 จะมีเลขหลักหน่วยไม่เท่ากับ 5 เสมอ จึงได้ว่า $q_i = 3063$ เป็นตัวหารเริ่มต้นใหม่สำหรับขั้นตอนวิธีการทดลองหารแบบปรับค่าลง

เนื่องจาก $\left\lfloor \sqrt{10276297} \right\rfloor$ = 3205 โดยที่จำนวนเฉพาะที่ไม่เท่ากับ 5 จะมีเลขหลักหน่วยไม่เท่ากับ 5 เสมอ จึงได้ว่า n_d = 3203 โดยค่าเป้าหมายคือ q = 1069 จึงสรุปได้ว่า q_i มีค่าเข้าใกล้ q มากกว่า n_d โดยหากพิจารณาจากจำนวนรอบของการทดลองหารทั้งหมดจะได้ว่า t_d = 1067 แต่ในทางกลับกัน t_{d_new} = 997 ซึ่งสังเกตได้ว่าจำนวนรอบการทดลองหารมีค่าลดลง

11. ขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์

ข้อเสียของขั้นตอนวิธีที่ 7.6 คือจำเป็นต้องมีการคำนวณหาค่ารากที่สองทุกรอบของการคำนวณเพื่อคำนวณหาผลลัพธ์ที่เป็นจำนวนเต็มโดยการคำนวณรากที่สองจำเป็นต้องใช้ทรัพยากรณ์การคำนวณที่สูง ถึงแม้ว่าขั้นตอนวิธีที่ 7.7 จะสามารถแก้ปัญหาดังกล่าวนี้ได้แต่กับเกิดปัญหาในเรื่องของรอบการคำนวณที่เพิ่มสูงขึ้นมากเมื่อเปรียบเทียบกับขั้นตอนวิธีที่ 7.6 ดังนั้นผู้เขียนจึงเสนอขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์ [51] ที่มีข้อดีคือการกำจัดข้อเสียของทั้งสองขั้นตอนวิธีออกโดยขั้นตอนวิธีที่นำเสนอใหม่นี้จะมีรอบการคำนวณเท่ากับขั้นตอนวิธีที่ 7.6 และไม่มีการคำนวณหารากที่สอง (การคำนวณรากที่สองเกิดขึ้นเพียงการคำนวณหาค่าเริ่มต้นของ $u = 2\left\lceil \sqrt{n} \right\rceil$ และการคำนวณหาผลลัพธ์ที่เป็นจำนวนเต็มเมื่อเงื่อนไขที่กำหนดเป็นจริงซึ่งความน่าจะเป็นที่เงื่อนไขจะเป็นจริงและได้ผลลัพธ์ที่ไม่เป็นจำนวนเต็มน้อยมาก เมื่อเปรียบเทียบกับความน่าจะเป็นที่เงื่อนไขเป็นเท็จ)

กำหนดให้ n = pq

ดังนั้น $\Phi(n) = (p-1)(q-1) = n-(p+q)+1$

จากทฤษฎีบทที่ 5.10, $a^{\Phi(n)} \equiv 1 \bmod n$

ดังนั้น
$$a^{n-(p+q)+1} \equiv 1 \bmod n \tag{8.15}$$

จากสมการ (7.20) พบว่ารอบการคำนวณทั้งหมดสำหรับการหาค่า u มีค่าเป็น $t_u = \frac{p+q}{2} - \left\lceil \sqrt{n} \right\rceil$ แต่เนื่องจากค่าเริ่มต้นของ u คือ $2\left\lceil \sqrt{n} \right\rceil$ และค่าสุดท้ายคือ $p+q$ หากพิจารณาขั้นตอนวิธีที่ 7.7 ที่เป็นแบบดั้งเดิม (ไม่พิจารณาขั้นตอนวิธีที่มีการปรับปรุง) พบว่า u จะเพิ่มขึ้นรอบละ 2 ค่า จึงสามารถปรับสมการที่ (7.20) ใหม่เป็นดังนี้

$$t_u = p + q - 2\left\lceil \sqrt{n} \right\rceil \tag{8.16}$$

หรือ $$p + q = t_u + 2\left\lceil \sqrt{n} \right\rceil \quad (8.17)$$

จากสมการ (8.15) ได้ว่า $$a^{n-(p+q)+1} \bmod n = a^{n-(t_u+2\left\lceil \sqrt{n} \right\rceil)+1} \bmod n$$

$$= a^{n-2\left\lceil \sqrt{n} \right\rceil+1-t_u} \bmod n$$

$$= a^{(n-2\left\lceil \sqrt{n} \right\rceil+1)-t_u} \bmod n$$

$$= a^{(n-2\left\lceil \sqrt{n} \right\rceil+1)} * a^{-t_u} \bmod n$$

$$= a^{(n-2\left\lceil \sqrt{n} \right\rceil+1)} * (a^{-1})^{t_u} \bmod n$$

หรือ $$a^{(n-2\left\lceil \sqrt{n} \right\rceil+1)} * (a^{-1})^{t_u} \equiv 1 \bmod n \quad (8.18)$$

กำหนดให้ $c = a^{-1} \bmod n$, $s = c^2 \bmod n$, $t_u = 0$, และ

$$t = a^{(n-2\left\lceil \sqrt{n} \right\rceil+1)} \bmod n \quad (8.19)$$

จากสมการ (8.19) แบ่งผลลัพธ์ของ t เป็น 2 กรณีดังนี้

กรณีที่ 1: $t = 1$, ตรวจสอบจำนวนเต็ม y จาก $y = \sqrt{x^2 - n}$ เมื่อ $x = \dfrac{2\left\lceil \sqrt{n} \right\rceil + t_u}{2}$ โดยกรณีที่ y เป็นจำนวนเต็มสามารถคำนวณหา p และ q ได้จาก $p = x + y$ และ $q = x - y$ ในทางกลับกันหาก y ไม่เป็นจำนวนเต็มให้ปรับค่า t และ t_u ใหม่ตามสมการที่ (8.20) และ (8.21) ตามลำดับ

กรณีที่ 2: $t \neq 1$, สามารถคาดการณ์ได้ว่า y ที่ได้จากการคำนวณไม่เป็นจำนวนเต็มอย่างแน่นอนจึงสามารถปรับค่า t และ t_u ใหม่ตามสมการที่ (8.18) และ (8.19) ตามลำดับ

$$t = t * s \bmod n \quad (8.20)$$

$$t_u = t_u + 2 \quad (8.21)$$

เนื่องจากค่าเริ่มต้นคือ $a^{(n-2\lceil\sqrt{n}\rceil+1)} \bmod n$ และค่าสุดท้ายที่เป็นคำตอบคือ $a^{(n-2\lceil\sqrt{n}\rceil+1)-t_u} \bmod n$ โดยการเพิ่มค่าจะเพิ่มรอบละ $(a^{-1})^2$ ดังเปรียบเสมือนว่า t_u ถูกปรับเพิ่มรอบละ 2 ค่าดังนั้นจึงกล่าวได้ว่ารอบการคำนวณของขั้นตอนวิธีที่นำเสนอใหม่นี้มีจำนวนเท่ากับรอบการคำนวณของขั้นตอนวิธีที่ 7.6 แต่การคำนวณหลักของขั้นตอนวิธีที่นำเสนอจะเป็นเพียงการบวกและการคูณมอดูลาร์เท่านั้นซึ่งใช้เวลาการประมวลผลน้อยมากเมื่อเปรียบเทียบกับการคำนวณหาค่ารากที่สอง

ตัวอย่างที่ 8.13 จงแสดงวิธีการแยกตัวประกอบ n = 344381 โดยใช้ขั้นตอนวิธีที่นำเสนอใหม่

วิธีทำ เริ่มจากเลือก a = 114794, $c = 114794^{-1} \bmod 344381 = 3$, $s = 3^2 \bmod 344381 = 9$ และ

$$t = 114794^{(344381-2\lceil\sqrt{344381}\rceil+1)} \bmod 344381 = 142088$$

เนื่องจาก $t \neq 1$ ดังนั้น

รอบที่ 1:

$t = 142088*9 \bmod 344381 = 245649$

$t_u = 0 + 2 = 2$

เนื่องจาก $t \neq 1$ ดังนั้น

รอบที่ 2:

$t = 245649*9 \bmod 344381 = 144555$

$t_u = 2 + 2 = 4$

เนื่องจาก $t \neq 1$ ดังนั้น

รอบที่ 3:

$t = 144555*9 \bmod 344381 = 267852$

$t_u = 4 + 2 = 6$

เนื่องจาก $t \neq 1$ ดังนั้น

รอบที่ 4:

$t = 267852*9 \bmod 344381 = 1$

$t_u = 6 + 2 = 8$

เนื่องจาก $t = 1$ ดังนั้น

$$x = \frac{2\left\lceil \sqrt{344381} \right\rceil + 8}{2} = 591$$

$$y = \sqrt{591^2 - 344381} = 70$$

เนื่องจาก y เป็นจำนวนเต็มแล้วดังนั้นได้ว่า

$p = 591 + 70 = 661$ และ $q = 591 - 70 = 521$

จากตัวอย่างข้างต้นสังเกตได้ว่ารอบการคำนวณมีค่าเท่ากับรอบการคำนวณจากตัวอย่างที่ 7.15 ซึ่งใช้ค่า n ค่าเดียวกัน อย่างไรก็ตามขั้นตอนวิธีที่นำเสนอจะมีการคำนวณค่ารากที่สองในกรณีที่ $t = 1$ เพียงเท่านั้นซึ่งความเป็นไปได้น้อยมากเมื่อเทียบกับผลลัพธ์ของ t ที่เป็นค่าอื่นๆ หรืออาจกล่าวได้ว่า $t = 1$ อาจเกิดขึ้นเพียงครั้งเดียวคือรอบที่ตรงกับผลลัพธ์

นอกเหนือจากนั้นยังสามารถนำเทคนิควิธีต่างๆที่ประยุกต์ใช้กับวิธีของแฟร์มาต์มาประยุกต์ใช้กับขั้นตอนวิธีที่นำเสนอนี้ได้เพื่อลดรอบและเวลาการคำนวณ เช่น การประมาณค่าเริ่มต้นใหม่ หรือการตัดค่าที่ไม่เกี่ยวข้องออกจากการคำนวณ เป็นต้น

ตารางที่ 8.3 เปรียบเทียบการดำเนินการหลักและรอบการคำนวณของขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์กับขั้นตอนวิธีที่ 7.6 และขั้นตอนวิธีที่ 7.7

ขั้นตอนวิธี	**รอบการคำนวณ**	**การดำเนินการหลัก**
ขั้นตอนวิธีแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์	ต่ำ	การคูณมอดูลาร์
ขั้นตอนวิธีที่ 7.6	ต่ำ	รากที่สอง
ขั้นตอนวิธีที่ 7.7	สูง	การคูณ

ตารางที่ 8.3 แสดงการเปรียบเทียบการดำเนินการหลักและรอบการคำนวณของขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์กับขั้นตอนวิธีที่ 7.6 และขั้นตอนวิธีที่ 7.7 ซึ่งทั้งสองวิธีนี้เป็นขั้นตอนวิธีแบบดั้งเดิมของแฟร์มาต์พบว่าขั้นตอนวิธีการแยกตัวประกอบใหม่ใช้จำนวนรอบการคำนวณต่ำซึ่งเทียบเท่ากับขั้นตอนวิธีที่ 7.6 อย่างไรก็ตามขั้นตอนวิธีใหม่จะคำนวณหาค่ารากที่สองในกรณีเพียงแค่เงื่อนไขของ t มีค่าเป็น 1 ซึ่งความน่าจะเป็นในการเกิดค่าดังกล่าวน้อย

มากหากเปรียบเทียบกับความน่าจะเป็นในการเกิดผลลัพธ์ของ t ที่ไม่เท่ากับ 1 ในทางกลับกันการคำนวณหารากที่สองเป็นการดำเนินการหลักสำหรับขั้นตอนวิธีที่ 7.6 ถึงแม้ว่าตัวดำเนินการหลักสำหรับขั้นตอนวิธีที่ 7.7 คือการคูณแต่รอบการคำนวณสูงมาก ดังนั้นหากเปรียบเทียบกันจากทั้ง 3 ขั้นตอนวิธีพบว่าขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์มีประสิทธิภาพสูงที่สุด นอกเหนือจากนั้นสามารถนำเทคนิควิธีต่างๆ ที่ใช้เพิ่มประสิทธิภาพสำหรับขั้นตอนวิธีที่ 7.6 หรือขั้นตอนวิธีที่ 7.7 มาประยุกต์ใช้กับขั้นตอนวิธีใหม่ได้เช่นกัน

12. บทสรุปสาระสำคัญ

ขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์เป็นขั้นตอนวิธีการแยกตัวประกอบที่มีประสิทธิภาพสูงมากในกรณีที่ตัวประกอบของมอดูลัสมีค่าที่ใกล้เคียงกัน หัวใจสำคัญคือการหาจำนวนเต็มสองค่าที่ผลต่างยกกำลังสองระหว่างจำนวนเต็มดังกล่าวนี้มีค่าเท่ากับมอดูลัส ในบทนี้ได้กล่าวถึงผลงานวิจัยที่เกี่ยวกับการปรับปรุงขั้นตอนวิธีการแยกตัวประกอบของแฟร์มาต์เพื่อลดจำนวนรอบการคำนวณซึ่งส่งผลให้เวลาที่ใช้สำหรับการประมวลผลลดลงด้วยเช่นกัน โดยงานวิจัยแบ่งออกเป็นสองกรณีคือ การตัดตัวเลขที่ไม่เกี่ยวข้องออกจากการคำนวณโดยใช้องค์ความรู้ทางทฤษฎีจำนวน เช่นการพิจารณาเลขหลักหน่วยจากการคำนวณหารากที่สองของจำนวนเต็ม การวิเคราะห์เลข m หลักสุดท้ายที่เป็นไปได้ทั้งหมดของคำตอบ เป็นต้น การปรับปรุงขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์อีกวิธีหนึ่งคือการประมาณค่าเริ่มต้นผลลัพธ์ใหม่ที่มีค่าใกล้เคียงกับผลลัพธ์มากยิ่งขึ้น งานวิจัยแรกของการประมาณค่าเริ่มต้นคือขั้นตอนวิธีที่เรียกว่า EPF ถูกนำเสนอโดยกลุ่มนักวิจัยชาวไต้หวัน 3 ท่าน ประกอบด้วย มูเอ็น วู เรย์ลิน โทส และ ฮุนมิน ซัน โดยใช้หลักการของการประมาณค่าเศษส่วนต่อเนื่องเพื่อหาเริ่มต้นของคำตอบใหม่ที่สามารถประมาณค่าได้ใกล้เคียงมากขึ้น อย่างไรก็ตามวิธีดังกล่าวนี้สามารถใช้ได้กับค่ามอดูลัสที่เกิดจากการคูณกันของจำนวนเฉพาะที่มีขนาดแตกต่างกันเท่านั้น หากเป็นจำนวนเฉพาะที่มีขนาดเท่ากันแล้วค่าเริ่มต้นที่ได้จะมีค่าเกินคำตอบ และเนื่องจากการคำนวณจะเป็นการเพิ่มค่าจากค่าเดิมในกรณีที่ผลลัพธ์ที่ได้ยังไม่ตรงกับคำตอบเสมอ ดังนั้นหากค่าเริ่มต้นที่ประมาณได้มีค่าเกินคำตอบแล้วจะไม่สามารถคำนวณหาคำตอบนี้ได้ จากปัญหาดังกล่าวนี้ผู้เขียนจึงนำเสนอสมการที่ใช้สำหรับการประมาณค่าเริ่มต้นใหม่ ซึ่งหากนำไปเปรียบเทียบกับ EPF แล้วมีข้อดีคือค่าเริ่มต้นที่เกิดจากการประมาณโดยขั้นตอนวิธีที่ผู้เขียนนำเสนอจะยังคงน้อยกว่าหรือเท่ากับค่าเป้าหมายเสมอ ดังนั้นจึงสามารถใช้ขั้นตอนวิธีนี้ได้กับค่ามอดูลัสทุกค่า อย่างไรก็ตาม EPF อาจประมาณค่าได้ใกล้เคียงกับคำตอบมากกว่าในกรณีที่ไม่มีข้อผิดพลาดเกิดขึ้น (จำนวนเฉพาะมีขนาดแตกต่างกัน)

นอกเหนือจากนั้นผู้เขียนได้นำเสนอขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์ที่มีจุดเด่นคือใช้รอบการคำนวณเท่ากับขั้นตอนวิธีที่ 7.6 แต่การคำนวณหาค่ารากที่สอง

เกิดขึ้นเพียงไม่กี่รอบ ซึ่งหากเปรียบเทียบกับขั้นตอนวิธีที่ 7.6 พบว่าการคำนวณหารากที่สองเกิดขึ้นทุกรอบของการคำนวณ โดยสามารถนำเทคนิควิธีต่างๆ ที่ใช้เพิ่มประสิทธิภาพสำหรับขั้นตอนวิธีที่ 7.6 หรือขั้นตอนวิธีที่ 7.7 มาประยุกต์ใช้กับขั้นตอนวิธีใหม่ได้เช่นกัน

โดยวัตถุประสงค์ของบทนี้คือเพื่อให้ผู้ที่ประสงค์ที่จะนำวิทยาการรหัสลับอาร์เอสเอไปประยุกต์ใช้งานพึงระมัดระวังในการเลือกใช้จำนวนเฉพาะให้มากขึ้น เนื่องจากขั้นตอนวิธีการแยกตัวประกอบยังคงถูกพัฒนาอย่างต่อเนื่อง สังเกตได้จากเดิมที่ขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์จะมีประสิทธิภาพเฉพาะในกรณีที่จำนวนเฉพาะมีค่าใกล้กัน แต่ในปัจจุบันที่ซึ่งขั้นตอนวิธีนี้ถูกปรับปรุงมาอย่างต่อเนื่องส่งผลให้ขั้นตอนวิธีดังกล่าวนี้ยังคงมีประสิทธิภาพสูงถึงแม้ว่าจำนวนเฉพาะทั้งสองค่าจะห่างกันมากขึ้น

แบบฝึกหัดท้ายบท
บทที่ 8

1. เลขหลักหน่วยของค่ากำลังสองสมบูรณ์เป็นเลขอะไรได้บ้าง

2. กำหนดให้ n = 5411611 และ x = 2403 หากพิจารณาตารางที่ 8.1 จำเป็นต้องคำนวณหาค่ารากที่สองหรือไม่ เพราะเหตุใด

3. จงหาผลลัพธ์ของ 171983214576324864153427199315 9 mod 4

4. จงพิจารณาหารูปแบบของ x สำหรับค่า n = 4682983 ที่พิจารณาหาจากเศษที่ได้จากการหาร n ด้วย 4

5. จงพิจารณาหารูปแบบของ x สำหรับค่า n = 4682983 ที่พิจารณาหาจากเศษที่ได้จากการหาร n ด้วย 6

6. จงพิจารณาหารูปแบบของ x สำหรับค่า n = 4682983 ที่พิจารณาหาจากเศษที่ได้จากการหาร n ด้วย 4, 6 และ 20

7. กำหนดให้ n = 512327 ((n + 1) mod 8 = 0) และสมมติค่าปัจจุบันของ u มีค่าเป็น 1436 แล้วค่า u ดังกล่าวมีโอกาสเป็นค่าที่แท้จริงหรือไม่เพราะเหตุใด

8. กำหนดให้ $a = a_m a_{m-1} a_{m-2} \cdots a_0$ เมื่อ $m > 1$ และ $b = b_1 b_0$ จงหาผลลัพธ์ของ $LSG_m(a + b \times 10^m)$

9. กำหนดให้ (51, 13) คือคู่ความสัมพันธ์คู่หนึ่งของ $LSG_2(n) = 63$ ที่ถูกเปิดเผย จงคำนวณหาคู่ความสัมพันธ์คู่อื่นที่เป็นไปได้ทั้งหมดในกลุ่มนี้

10. กำหนดให้ (17, 39) คือคู่ความสัมพันธ์คู่หนึ่งของ $LSG_2(n) = 63$ ที่ถูกเปิดเผย จงคำนวณหาคู่ความสัมพันธ์คู่อื่นที่เป็นไปได้ทั้งหมดในกลุ่มนี้

11. จงหาเลข 2 ตัวสุดท้ายของ u และ v ที่เป็นไปได้ทั้งหมดสำหรับ $LSG_2(n) = 63$

12. จากคำตอบที่ได้ข้อ 11 จงหาค่าเริ่มต้นของ u สำหรับแยกตัวประกอบ n = 6847963

13. จากคำตอบที่ได้ข้อ 11 และ 12 จงหาค่าเริ่มต้นของ v สำหรับแยกตัวประกอบ n = 6847963 โดยใช้วิธีการประมาณค่าเริ่มต้นรูปแบบใหม่ที่สามารถใช้ได้กับค่ามอดูลัสทุกกรณี

14. จงหาตัวประกอบของ n = 6847963 โดยใช้ขั้นตอนวิธีการแยกตัวประกอบด้วยวิธีของแฟร์มาต์แบบไม่คำนวณหารากที่สองโดยใช้ค่าเริ่มต้นของ u และ v ที่ได้จากคำถามข้อ 12 และ 13

15. จุดเด่นของขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์คืออะไร

16. กำหนดให้ $n = 738233$ และ $a = 101$ จงหาค่าเริ่มต้นของ t เพื่อใช้สำหรับการแยกตัวประกอบใหม่ที่มีรากฐานจากวิธีของแฟร์มาต์

17. จากค่าเริ่มต้นของ u และ v ที่คำนวณได้จากคำถามข้อ 12 และ 13 จงหาค่าเริ่มต้นใหม่ของตัวหารสำหรับขั้นตอนวิธีการทดลองหารแบบปรับค่าลงเพื่อแยกตัวประกอบ $n = 6847963$

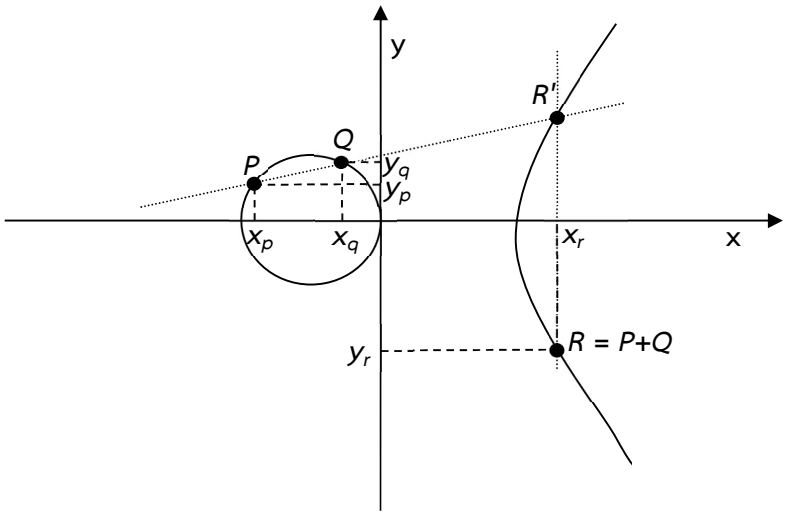
บทที่ 9
วิทยาการรหัสลับเส้นโค้งเชิงวงรี

ปัญหาของวิทยาการรหัสลับอาร์เอสเอคือค่ามอดูลัสต้องมีขนาดไม่น้อยกว่า 1024 บิต จึงจะส่งผลให้ข้อมูลข่าวสารมีความปลอดภัย อย่างไรก็ตามค่ามอดูลัสดังกล่าวนี้เป็นขนาดที่ใหญ่มหาศาลซึ่งผลเสียที่ตามมาคือจำเป็นต้องใช้กำลัง และเวลาสำหรับการประมวลผลสูง ดังนั้นวิทยาการรหัสลับอาร์เอสเอจึงไม่เหมาะสมที่จะนำมาประยุกต์ใช้งานกับคอมพิวเตอร์ที่มีหน่วยประมวลผลที่มีประสิทธิภาพต่ำ และอุปกรณ์อิเล็กทรอนิกส์ที่เป็นแบบสมองกลฝังตัว เช่น สมาร์ทโฟน แท็บเล็ต ซึ่งเป็นอุปกรณ์อิเล็กทรอนิกส์ที่ได้รับความนิยมสูงมากในปัจจุบัน

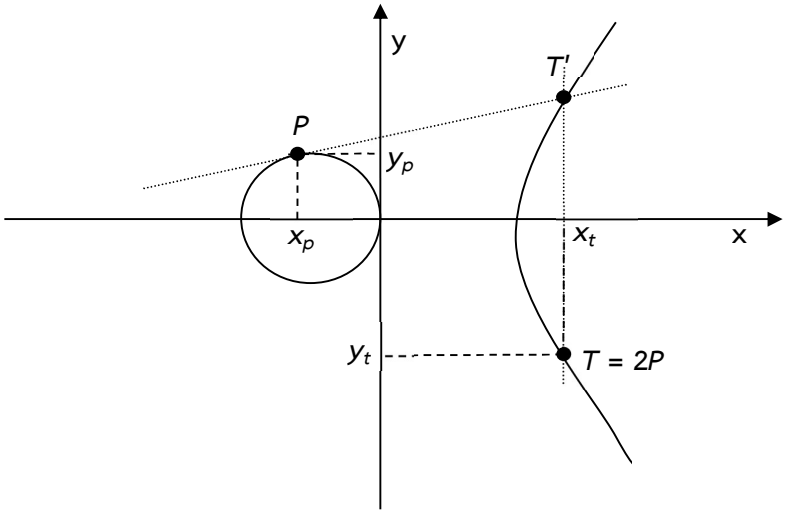
จากปัญหาข้างต้น นีล โคบลิทซ์ (Neal Koblitz) [9] และ วิคเตอร์ มิลเลอร์ (Victor Miller) [10] ได้เสนอวิทยาการรหัสลับแบบกุญแจสาธารณะอีกรูปแบบหนึ่งเรียกว่า วิทยาการรหัสลับเส้นโค้งเชิงวงรี (Elliptic Curve Cryptography, ECC) คือการนำสมการเส้นโค้งเชิงวงรีมาประยุกต์ใช้สำหรับการรักษาความปลอดภัยข้อมูลข่าวสารผ่านกระบวนการเข้ารหัสลับ และถอดรหัสลับในรูปแบบของกุญแจสาธารณะ จุดเด่นของวิทยาการรหัสลับเส้นโค้งเชิงวงรีคือเป็นขั้นตอนวิธีที่มีความปลอดภัยสูงมาก ถึงแม้ว่าค่ากุญแจจะมีขนาดเล็ก ยกตัวอย่างเช่นกุญแจลับขนาดประมาณ 160 บิต [36], [37] สำหรับวิทยาการรหัสลับเส้นโค้งเชิงวงรีมีความปลอดภัยเทียบเท่ากับกุญแจที่มีขนาด 1024 บิต สำหรับวิทยาการรหัสลับอาร์เอสเอ ดังนั้นวิทยาการรหัสลับเชิงโค้งเชิงวงรีจึงนิยมถูกนำมาประยุกต์ใช้งานร่วมกับคอมพิวเตอร์ที่มีหน่วยประมวลผลประสิทธิภาพต่ำ หรืออุปกรณ์อิเล็กทรอนิกส์ที่เป็นแบบสมองกลฝังตัว

วิทยาการรหัสลับเส้นโค้งเชิงวงรีสามารถนำมาประยุกต์ใช้งานได้กับ 3 งานหลักประกอบด้วย การแลกเปลี่ยนกุญแจ การรักษาความปลอดภัย และลายเซ็นดิจิทัล นอกเหนือจากนั้นวิทยาการรหัสลับชนิดดังกล่าวยังถูกแบ่งออกเป็นหลายประเภท เช่น วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ (Elliptic Curve Cryptography over Prime Field) วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง (Elliptic Curve Cryptography over Field of Characteristic Two) และ วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสาม (Elliptic Curve Cryptography over Field of Characteristic Three) อย่างไรก็ตาม สำหรับบทนี้จะกล่าวถึงเพียง วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และ วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง เท่านั้น

1. เส้นโค้งเชิงวงรี



รูปที่ 9.1 การจำลองวิธีการคำนวณ Point Addition บนเส้นโค้งเชิงวงรี



รูปที่ 9.2 การจำลองวิธีการคำนวณ Point Doubling บนเส้นโค้งเชิงวงรี

รูปที่ 9.1 แสดงตัวอย่างการคำนวณหาผลบวกระหว่างสองจุดบนเส้นโค้งเชิงวงรีโดยมีจุด $P = (x_p, y_p)$ และจุด $Q = (x_q, y_q)$, (เมื่อ $P \neq Q$) เป็นจุดพิกัดที่อยู่บนเส้นโค้งเชิงวงรีการคำนวณหาจุด $R = P + Q = (x_r, y_r)$ สามารถดำเนินการได้โดยลากเส้นตรงผ่านจุด P และ Q จนกระทั่งเส้นตรงดังกล่าวตัดจุดใหม่ (กำหนดเป็น R') และให้ลากเส้นตรงใหม่ที่เป็นแนวตั้งอีกเส้นจากจุด R' ผ่านแกน x จนกระทั่งเส้นตรงตัดจุดใหม่อีกจุดบนเส้นโค้ง จุดดังกล่าวคือคำตอบของ R โดยเรียกวิธีการนี้ว่า Point Addition

สำหรับการคำนวณอีกวิธีหนึ่งดังรูปที่ 9.2 คือการคำนวณหาจุด $T = 2P$ โดยกำหนดจุด $P = (x_p, y_p)$ และจุดผลลัพธ์ $T = (x_t, y_t)$ การคำนวณหาจุด T สามารถดำเนินการได้โดยลากเส้นตรงผ่านจุด P จนกระทั่งเส้นตรงดังกล่าวตัดจุดใหม่ (กำหนดเป็น T') และให้ลากเส้นตรงใหม่ที่เป็นแนวตั้งอีกเส้นจากจุด T' ผ่านแกน x จนกระทั่งเส้นตรงตัดจุดใหม่อีกจุดบนเส้นโค้ง จุดดังกล่าวคือคำตอบของ T โดยเรียกวิธีการนี้ว่า Point Doubling

สำหรับการบวกระหว่างจุดบนเส้นโค้งเชิงวงรีมีกฎที่สำคัญดังนี้

กำหนดให้ P, Q และ R คือจุดที่อยู่บนเส้นโค้งเชิงวงรี โดยที่ $P = (x_p, y_p)$ ได้ว่า

กฎข้อที่ 1: $P + Q = Q + P$ (กฎการสลับที่)

กฎข้อที่ 2: $(P + Q) + R = P + (Q + R)$ (กฎการเปลี่ยนกลุ่ม)

กฎข้อที่ 3: กรณีที่ความชันของเส้นตรงที่ลากผ่านจุดกำเนิดเป็น ∞ กล่าวได้ว่าจุดบนเส้นโค้งที่เป็นผลลัพธ์คือ ∞

กฎข้อที่ 4: $P + \infty = P$

กฎข้อที่ 5: $-P = (x_p, -y_p)$ สำหรับกรณีที่เป็นวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

$-P = (x_p, x_p + y_p)$ สำหรับกรณีที่เป็นวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง

กฎข้อที่ 6: $P + (-P) = \infty$

2. ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี (Elliptic Curve Discrete Logarithm Problem)

ความยากของการโจมตีวิทยาการรหัสลับเส้นโค้งเชิงวงรีขึ้นอยู่กับปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี (Elliptic Curve Discrete Logarithm Problem, ECDLP) กล่าวคือการคำนวณจุด $Q = kP$ เมื่อ P คือจุดกำเนิด และ $k \in \mathbb{Z}$ สามารถดำเนินการได้อย่างรวดเร็ว ในทางกลับกันการคำนวณหา k ทำได้ยากมากหรือไม่สามารถดำเนินการได้ในระยะเวลาที่สั้น ถึงแม้ว่าปัจจุบันจะมีขั้นตอนวิธีที่ใช้สำหรับการแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรีถูกนำเสนอออกมาเป็นจำนวนมากซึ่งโดยส่วนใหญ่

จะใช้ขั้นตอนวิธีเดียวกันกับการแก้ปัญหาวิยุตลอการิทึมเพียงแต่ปรับเปลี่ยนรูปแบบสมการของการคำนวณ เช่น การโจมตีแบบตะลุย [38] ขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ และ ขั้นตอนวิธีโพลิก เฮลแมน เป็นต้น จากหลักการดังกล่าวส่งผลให้เส้นโค้งเชิงวงรีจึงนิยมถูกนำมาประยุกต์ใช้สำหรับการรักษาความปลอดภัยข้อมูลข่าวสารผ่านกระบวนการเข้ารหัส และถอดรหัสข้อมูล โดยเฉพาะอย่างยิ่งการนำมาใช้งานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ที่เป็นสมองกลฝังตัวซึ่งมีหน่วยประมวลผลประสิทธิภาพต่ำ เนื่องจากกุญแจที่ถูกนำมาใช้งานมีขนาดเล็ก

3. วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

กำหนดให้ $p \in \mathbb{Z}^+$ และ $p > 3$ (แต่โดยส่วนใหญ่นิยมเลือก p ที่เป็นจำนวนเฉพาะ) สมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะคือคู่อันดับ (x, y) ที่เป็นไปได้ทั้งหมดที่ทำให้สมการต่อไปนี้เป็นจริง

$$y^2 \equiv x^3 + ax + b \bmod p \tag{9.1}$$

เมื่อ a, b คือจำนวนเต็มใดๆ ที่อยู่ในฟิลด์ GF(p)

และ

$$a^3 + 27b^2 \bmod p \neq 0 \tag{9.2}$$

3.1 การสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

หัวข้อนี้จะแนะนำวิธีการสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ ซึ่งหากเลือกวิธีที่ไม่มีประสิทธิภาพจะส่งผลให้ใช้เวลาการประมวลผลสูงมาก ตัวอย่างเช่นหากเลือกค่า x, a, b และ p ก่อน แล้วจึงพิจารณาหา y พบว่าอาจเป็นไปได้ที่จะต้องปรับเปลี่ยนค่าของ x, a, b หรือ p ใหม่สาเหตุเกิดจากจำเป็นต้องมีการคำนวณหากำลังสองสมบูรณ์ของ y และไม่พบคำตอบที่ทำให้สมการเป็นจริง ซึ่งวิธีดังกล่าวจะใช้เวลาและกำลังการประมวลผลสูง อย่างไรก็ตามวิธีที่เหมาะสมที่ใช้เวลา และกำลังสำหรับการประมวลผลต่ำมีขั้นตอนคือ เลือกคู่อันดับ (x, y), a และ p ก่อน แล้วจึงคำนวณหา b ที่ทำให้สมการเป็นจริง

ตัวอย่างที่ 9.1 การทดสอบสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

วิธีทำ

สมมติผู้ใช้งานเลือกพารามิเตอร์ทั้งหมดเป็นดังนี้

$(x, y) = (17, 3)$, $a = 23$ และ $p = 53$

ดังนั้นจากสมการ (9.1) ได้ว่า

$$3^2 = 17^3 + 23 \times 17 + b \bmod 53$$

$$9 = 4913 + 391 + b \bmod 53$$

$$9 = 37 + 20 + b \bmod 53$$

$$9 = 57 + b \bmod 53$$

$$9 = 4 + b \bmod 53$$

ดังนั้น

$$b = 5 \bmod 53$$

และจากสมการ (9.2) เนื่องจาก

$$4 \times 17^3 + 27 \times 5^2 \bmod 53 = 19652 + 675 \bmod 53$$

$$= 42 + 39 \bmod 53$$

$$= 81 \bmod 53$$

$$= 28 \bmod 53$$

$$\neq 0 \bmod 53$$

จึงได้สมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะเป็นดังนี้

$y^2 = x^3 + 23x + 5 \bmod 53$ ที่มี (17, 3) เป็นคู่อันดับหนึ่งที่อยู่บนเส้นโค้งนี้

3.2 การคำนวณหาผลบวกระหว่างจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

การคำนวณหา $R = P + Q$ เมื่อ $P = (x_p, y_p)$, $Q = (x_q, y_q)$, $R = (x_r, y_r)$ สามารถคำนวณหา x_r และ y_r ได้ ดังนี้

กรณีที่ $P \neq Q$, $x_r = m^2 - x_q - x_p \bmod p$ (9.3)

กรณีที่ $P = Q$, $x_r = m^2 - 2x_p \bmod p$ (9.4)

และ

$$y_r = m(x_p - x_r) - y_p \bmod p \quad (9.5)$$

เมื่อ m คือค่าความชันของเส้นตรงที่ลากผ่านจุดกำเนิด ซึ่งแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1 (Point Addition): $P \neq Q$

$$m = \frac{y_q - y_p}{x_q - x_p} \bmod p \qquad (9.6)$$

การคำนวณหาค่า m จากสมการ (9.6) พบว่าจะมีการคำนวณหาค่าผกผันเหนือฟิลด์จำนวนเฉพาะจำนวน 1 ครั้ง และการคูณเหนือฟิลด์จำนวนเฉพาะอีก 1 ครั้ง

และจากสมการที่ (9.3) และ (9.5) พบว่ามีการคำนวณหาค่ากำลังสองสมบูรณ์เหนือฟิลด์จำนวนเฉพาะจำนวน 1 ครั้ง และการคูณเหนือฟิลด์จำนวนเฉพาะอีก 1 ครั้ง ดังนั้นจึงสรุปได้ว่า Point Addition ใช้ทรัพยากรสำหรับการประมวลผลเป็นดังนี้ ค่าผกผันเหนือฟิลด์จำนวนเฉพาะจำนวน 1 ครั้ง การคูณเหนือฟิลด์จำนวนเฉพาะ 2 ครั้ง และกำลังสองสมบูรณ์ 1 ครั้ง

กรณีที่ 2 (Point Doubling): $P = Q$

$$m = \frac{3x_p^2 + a}{2y_p} \bmod p \qquad (9.7)$$

การคำนวณหาค่า m จากสมการ (9.7) พบว่าจะมีการคำนวณหาค่าผกผันเหนือฟิลด์จำนวนเฉพาะจำนวน 1 ครั้ง การคูณเหนือฟิลด์จำนวนเฉพาะอีก 1 ครั้ง และ กำลังสองสมบูรณ์ 1 ครั้ง

ดังนั้นหากพิจารณาร่วมกับการคำนวณหา x_r และ y_r จากสมการที่ (9.4) และ (9.5) สรุปได้ว่า Point Doubling ใช้ทรัพยากรสำหรับการประมวลผลเป็นดังนี้ ค่าผกผันเหนือฟิลด์จำนวนเฉพาะจำนวน 1 ครั้ง การคูณเหนือฟิลด์จำนวนเฉพาะ 2 ครั้ง และกำลังสองสมบูรณ์ 2 ครั้ง

ตัวอย่างที่ 9.2 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และจุด P ที่ได้จากตัวอย่างที่ 9.1 จงคำนวณหา $Q = 2P$

วิธีทำ

เนื่องจาก $Q = 2P = P + P$, $P = P$, ดังนั้นจึงต้องใช้ Point Doubling

เริ่มจากการคำนวณหาค่า m โดยใช้สมการที่ (9.7) ได้ดังนี้

$$m = \frac{3x_p^2 + a}{2y_p} \bmod p$$

$$= \frac{3\text{x}17^2 + 23}{2\text{x}3} \bmod 53$$

$$= \frac{890}{6} \mod 53$$

$$= 42 \times 6^{-1} \mod 53$$

$$= 42 \times 9 \mod 53$$

$$= 7$$

และจากสมการ (9.4) และ (9.5) ได้ว่า

$$x_q = m^2 - 2x_p \mod p$$

$$= 7^2 - 2 \times 17 \mod 53$$

$$= 15$$

และ $y_q = m(x_p - x_q) - y_p \mod p$

$$= 7 \times (17 - 15) - 3 \mod 53$$

$$= 11$$

ดังนั้นสรุปได้ว่า $Q = (15, 11)$

ตัวอย่างที่ 9.3 จากผลลัพธ์ตัวอย่างที่ 9.2 จงคำนวณหา $R = P + Q$

วิธีทำ

เนื่องจาก $R = P + Q$, $P \neq Q$, ดังนั้นจึงต้องใช้ Point Addition เริ่มจากการคำนวณหาค่า m โดยใช้สมการที่ (9.6) ได้ดังนี้

$$m = \frac{y_q - y_p}{x_q - x_p} \mod p$$

$$= \frac{11-3}{15-17} \mod 53$$

$$= \frac{8}{(-2)} \mod 53$$

$$= \frac{8}{51} \mod 53$$

$$= 8 \times 51^{-1} \mod 53$$

$$= 8 \times 26 \mod 53$$

$$= 49$$

และจากสมการ (9.3) และ (9.5) ได้ว่า

$$x_r = m^2 - x_q - x_p \mod p$$

$= 49^2 - 15 - 17 \bmod 53$

$= 37$

และ $y_r = m(x_p - x_r) - y_p \bmod p$

$= 49 \times (17 - 37) - 3 \bmod 53$

$= -29 \bmod 53$

$= 24$

ดังนั้นสรุปได้ว่า $R = (37, 24)$

3.3 การคำนวณหาผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

การคำนวณหาผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะเป็นการดำเนินการที่สำคัญที่ใช้สำหรับการคำนวณหาจุดผลลัพธ์บนเส้นโค้ง อย่างไรก็ตามไม่มีตัวดำเนินการคูณสำหรับเส้นโค้งเชิงวงรีโดยตรง แต่สามารถคำนวณหาได้โดยใช้การดำเนินการบวกทั้งสองประเภท (Point Addition และ Point Doubling) มาประยุกต์ใช้งานแทน

การคำนวณหา $Q = kP$ เมื่อ $k \in \mathbb{Z}^+$ สามารถดำเนินการได้โดยการนำการดำเนินการบวกมาประยุกต์ใช้งานได้ดังนี้

$$Q = \underbrace{P+P+P+ \cdots +P}_{k \text{ ตัว}} \tag{9.8}$$

อย่างไรก็ตามการเลือกตัวดำเนินการเป็นสิ่งสำคัญ ซึ่งหากเลือกได้ดีจะช่วยให้เวลา และทรัพยากรที่ใช้สำหรับการคำนวณลดลง ตัวอย่างที่ 9.4 แสดงการเปรียบเทียบวิธีการคำนวณหา $8P$ ระหว่างการใช้ Point Addition และ Point Doubling

ตัวอย่างที่ 9.4 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และจุด P ที่ได้จากตัวอย่างที่ 9.1 จงคำนวณหา $8P$ โดยใช้ Point Additon และ Point Doubling แบบเพิ่มครั้งละ 1 ลำดับ

วิธีทำ จากตัวอย่างที่กำหนด แบ่งวิธีการคำนวณออกเป็น 2 วิธี ดังนี้

วิธีที่ 1: Point Addition ซึ่งมีลำดับการดำเนินการเป็นดังนี้

ลำดับที่ 1: คำนวณหา $2P = P + P$ แต่อย่างไรก็ตามการคำนวณหา $2P$ ไม่สามารถดำเนินการได้โดย Point Addition จึงจำเป็นต้องใช้ Point Doubling ซึ่งจากตัวอย่างที่ 9.2 ได้ว่า $2P = (15, 11)$

ลำดับที่ 2: คำนวณหา $3P = 2P + P$ ซึ่งจากตัวอย่างที่ 9.3 ได้ว่า $3P = (37, 24)$

ลำดับที่ 3: คำนวณหา $4P = 3P + P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $$m = \frac{y_{3p} - y_p}{x_{3p} - x_p} \bmod p$$

$$= \frac{24 - 3}{37 - 17} \bmod 53$$

$$= \frac{21}{20} \bmod 53$$

$$= 21 \times 20^{-1} \bmod 53$$

$$= 21 \times 8 \bmod 53$$

$$= 9$$

$$x_{4p} = m^2 - x_{3p} - x_p \bmod p$$

$$= 9^2 - 37 - 17 \bmod 53$$

$$= 27$$

และ $$y_{4p} = m(x_p - x_{4p}) - y_p \bmod p$$

$$= 9 \times (17 - 27) - 3 \bmod 53$$

$$= 13$$

ดังนั้น $4P = (27, 13)$

ลำดับที่ 4: คำนวณหา $5P = 4P + P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $$m = \frac{y_{4p} - y_p}{x_{4p} - x_p} \bmod p$$

$$= \frac{13 - 3}{27 - 17} \bmod 53$$

$$= \frac{10}{10} \bmod 53$$

$$= 10 \times 10^{-1} \bmod 53$$

$$= 1$$

$$x_{5p} = m^2 - x_{4p} - x_p \bmod p$$

$$= 1^2 - 27 - 17 \bmod 53$$

$$= 10$$

และ $$y_{5p} = m(x_p - x_{5p}) - y_p \bmod p$$

$= 1 \times (17 - 10) - 3 \bmod 53$

$= 4$

ดังนั้น $5P = (10, 4)$

ลำดับที่ 5: คำนวณหา $6P = 5P + P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $m = \dfrac{y_{5p} - y_p}{x_{5p} - x_p} \bmod p$

$= \dfrac{4 - 3}{10 - 17} \bmod 53$

$= \dfrac{1}{46} \bmod 53$

$= 1 \times 46^{-1} \bmod 53$

$= 1 \times 15 \bmod 53$

$= 15$

$x_{6p} = m^2 - x_{5p} - x_p \bmod p$

$= 15^2 - 10 - 17 \bmod 53$

$= 39$

และ $y_{6p} = m(x_p - x_{6p}) - y_p \bmod p$

$= 15 \times (17 - 39) - 3 \bmod 53$

$= 38$

ดังนั้น $6P = (39, 38)$

ลำดับที่ 6: คำนวณหา $7P = 6P + P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $m = \dfrac{y_{6p} - y_p}{x_{6p} - x_p} \bmod p$

$= \dfrac{38 - 3}{39 - 17} \bmod 53$

$= \dfrac{35}{22} \bmod 53$

$= 35 \times 22^{-1} \bmod 53$

$= 35 \times 41 \bmod 53$

$= 4$

$$x_{7p} = m^2 - x_{6p} - x_p \bmod p$$
$$= 4^2 - 39 - 17 \bmod 53$$
$$= 13$$

และ $$y_{7p} = m(x_p - x_{7p}) - y_p \bmod p$$
$$= 4 \times (17 - 13) - 3 \bmod 53$$
$$= 13$$

ดังนั้น $7P = (13, 13)$

ลำดับที่ 7: คำนวณหา $8P = 7P + P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $$m = \frac{y_{7p} - y_p}{x_{7p} - x_p} \bmod p$$
$$= \frac{13 - 3}{13 - 17} \bmod 53$$
$$= \frac{10}{49} \bmod 53$$
$$= 10 \times 49^{-1} \bmod 53$$
$$= 10 \times 13 \bmod 53$$
$$= 24$$
$$x_{8p} = m^2 - x_{7p} - x_p \bmod p$$
$$= 24^2 - 13 - 17 \bmod 53$$
$$= 16$$

และ $$y_{8p} = m(x_p - x_{8p}) - y_p \bmod p$$
$$= 24 \times (17 - 16) - 3 \bmod 53$$
$$= 21$$

ดังนั้น $8P = (16, 21)$

วิธีที่ 2: Point Doubling ซึ่งมีลำดับการดำเนินการเป็นดังนี้

ลำดับที่ 1: คำนวณหา $2P = P + P$ ซึ่งจากตัวอย่างที่ 9.2 ได้ว่า $2P = (15, 11)$

ลำดับที่ 2: คำนวณหา $4P = 2P + 2P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $$m = \frac{3x_{2p}^2 + a}{2y_{2p}} \bmod p$$

$$= \frac{3x15^2 + 23}{2x11} \mod 53$$

$$= \frac{9}{22} \mod 53$$

$$= 9 \times 22^{-1} \mod 53$$

$$= 9 \times 41 \mod 53$$

$$= 51$$

$$x_{4p} = m^2 - 2x_{2p} \mod p$$

$$= 51^2 - 2 \times 15 \mod 53$$

$$= 27$$

และ $$y_{4p} = m(x_{2p} - x_{4p}) - y_{2p} \mod p$$

$$= 51 \times (15 - 27) - 11 \mod 53$$

$$= 13$$

ดังนั้น $4P = (27, 13)$

ลำดับที่ 3: คำนวณหา $8P = 4P + 4P$ ซึ่งสามารถคำนวณได้ดังนี้

จาก $$m = \frac{3x_{4p}^2 + a}{2y_{4p}} \mod p$$

$$= \frac{3x27^2 + 23}{2x13} \mod 53$$

$$= \frac{37}{26} \mod 53$$

$$= 37 \times 26^{-1} \mod 53$$

$$= 37 \times 51 \mod 53$$

$$= 32$$

$$x_{8p} = m^2 - 2x_{4p} \mod p$$

$$= 32^2 - 2 \times 27 \mod 53$$

$$=16$$

และ $$y_{8p} = m(x_{4p} - x_{8p}) - y_{4p} \mod p$$

$$= 32 \times (27 - 16) - 13 \mod 53$$

$$= 21$$

ดังนั้น $8P = (16, 21)$

จากตัวอย่างที่ 9.4 สรุปได้ดังนี้ การคำนวณ $8P$ แบบเพิ่มครั้งละ 1 ลำดับ พบว่าหากใช้วิธีการคำนวณแบบ Point Addition จำเป็นต้องใช้การคำนวณสูงถึง 7 ครั้ง ในทางกลับกันหากเลือกใช้วิธีการคำนวณแบบ Point Doubling จะมีการคำนวณเกิดขึ้นเพียง 3 ครั้ง

ดังนั้นการเลือกตัวดำเนินการเป็นปัจจัยสำคัญที่จะช่วยการใช้เวลาสำหรับการคำนวณและทรัพยากรลดลง หัวข้อถัดไปจะกล่าวถึงขั้นตอนวิธีทวิภาคซึ่งเป็นเทคนิควิธีการคำนวณผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้งเชิงวงรีที่ช่วยลดทั้งเวลาและทรัพยากรสำหรับการคำนวณลงได้

3.4 ขั้นตอนวิธีทวิภาค (Binary Method)

ขั้นตอนวิธีทวิภาค [48] คือเทคนิควิธีที่นำมาประยุกต์ใช้สำหรับการคำนวณหาผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้งเชิงวงรีเพื่อช่วยลดทั้งเวลาและทรัพยากรสำหรับการคำนวณ โดยมีหลักการคือจะแปลงเลขจำนวนเต็มซึ่งเป็นตัวคูณของจุดบนเส้นโค้งเชิงวงรีจากเลขฐานสิบเป็นเลขฐานสองเพื่อนำมาพิจารณาหาผลลัพธ์โดยใช้วิธีการตรวจสอบค่าเลขฐานสองแต่ละตำแหน่งซึ่งแบ่งออกเป็นสองวิธีคือการตรวจสอบจากตำแหน่งซ้ายสุดไปยังตำแหน่งขวาสุด (ตำแหน่งที่มีนัยสำคัญสูงสุดไปยังตำแหน่งที่มีนัยสำคัญต่ำสุด) และการตรวจสอบจากตำแหน่งขวาสุดไปยังตำแหน่งซ้ายสุด (ตำแหน่งที่มีนัยสำคัญต่ำสุดไปยังตำแหน่งที่มีนัยสำคัญสูงสุด) กำหนดให้ $Q = kP$ เมื่อ $k = (k_{m-1}k_{m-2}k_{m-3}\cdots k_0)_2$ ขั้นตอนวิธีทวิภาคจึงถูกแบ่งออกเป็น 2 วิธีดังต่อไปนี้

ขั้นตอนวิธีที่ 9.1 ทวิภาค (ตรวจสอบจากตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด)

```
     INPUT: k = (k_{m-1}k_{m-2}...k_2k_1k_0)_2, P, y^2 = x^3 + ax + b mod p
     OUTPUT: Q = kP
1:   Q ← P
2:   For i = m-2 to 0 do
3:             Q ← 2Q
4:             IF k_i == 1 then
5:                 Q ← Q + P
6:             End IF
7:   End For
```

ตัวอย่างที่ 9.5 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และจุด P ที่ได้จากตัวอย่างที่ 9.1 จงคำนวณหา $5P$ โดยใช้ขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด)

วิธีทำ เนื่องจาก $k = 5 = 101_2$ และจากขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด) มีลำดับขั้นตอนการดำเนินการเป็น ดังนี้

1. $Q = P = (17, 3)$

ขั้นตอนที่ 2 – 7 เป็นขั้นตอนที่มีการทำงานภายในวงวนดังนี้

รอบที่ 1: $i = 1, k_1 = 0$

3. $Q = 2Q = 2(17, 3) = (15, 11)$

ขั้นตอนที่ 4 – 6 ไม่มีการดำเนินการ (เนื่องจาก $k_2 = 0$)

รอบที่ 2: $i = 0, k_0 = 1$

3. $Q = 2Q = 2(15, 11) = (27, 13)$

ขั้นตอนที่ 4 – 6: $Q = Q + P = (27, 13) + (17, 3) = (10, 4)$

ดังนั้นสรุปได้ว่าการคำนวณหา $Q = 5P$ ด้วยขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด) ได้จุดผลลัพธ์บนเส้นโค้งคือ (10, 4) ซึ่งจากตัวอย่างมีการใช้ Point Doubling 2 ครั้ง ใช้ Point Addition 1 ครั้ง และมีการวนรอบการทำงาน 2 รอบ

ขั้นตอนวิธีที่ 9.2 ทวิภาค (ตรวจสอบจากตำแหน่งขวาสุดไปตำแหน่งซ้ายสุด)

```
     INPUT: k = (k_{m-1}k_{m-2}…k_2k_1k_0)_2, P, y^2 = x^3 + ax + b mod p
     OUTPUT: Q = kP
1:   Q ← 0
2:   S ← P
3:   For i = 0 to m-1 do
4:       IF k_i == 1 then
5:           IF Q == 0 then
6:               Q ← S
7:           Else
8:               Q ← Q + S
9:           End IF
10:      End IF
11:      S ← 2S
12:  End For
```

ตัวอย่างที่ 9.6 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และจุด P ที่ได้จากตัวอย่างที่ 9.1 จงคำนวณหา $5P$ โดยใช้ขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งขวาสุดไปตำแหน่งซ้ายสุด)

วิธีทำ เนื่องจาก $k = 5 = 101_2$ และจากขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งขวาสุดไปตำแหน่งซ้ายสุด) มีลำดับขั้นตอนการดำเนินการเป็น ดังนี้

1. $Q = 0$

2. $S = P = (17, 3)$

ขั้นตอนที่ 3 - 12 เป็นขั้นตอนที่มีการทำงานภายในวงวนดังนี้

รอบที่ 1: $i = 0, k_0 = 1$

ขั้นตอนที่ 4 – 10 เนื่องจาก $k_0 = 1$ และ $Q = 0$ ดังนั้น $Q = S = (17, 3)$

11. $S = 2S = 2(17, 3) = (15, 11)$

รอบที่ 2: $i = 1$, $k_1 = 0$

ขั้นตอนที่ 4 – 10 ไม่มีการดำเนินการ (เนื่องจาก $k_1 = 0$)

11. $S = 2S = 2(15, 11) = (27, 13)$

รอบที่ 3: $i = 2$, $k_2 = 1$

ขั้นตอนที่ 4 – 10 เนื่องจาก $k_2 = 1$ และ $Q \neq 0$ ดังนั้น

$$Q = Q + S = (17, 3) + (27, 13) = (10, 4)$$

11. $S = 2S = 2(27, 13) = (16, 21)$

ดังนั้นสรุปได้ว่าการคำนวณหา $Q = 5P$ ด้วยขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งขวาสุดไปตำแหน่งซ้ายสุด) ได้จุดผลลัพธ์บนเส้นโค้งคือ (10, 4) อย่างไรก็ตามจากตัวอย่างมีการใช้ Point Doubling 3 ครั้ง ใช้ Point Addition 1 ครั้ง และมีการวนรอบการทำงาน 3 รอบ ซึ่งพบว่าการคำนวณ Point Doubling และจำนวนรอบการทำงานของวิธีข้างต้นสูงกว่าการคำนวณด้วยขั้นตอนวิธีทวิภาค (ตรวจสอบจากตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด)

3.5 การคำนวณหา $2P+Q$ โดยการตัดการคำนวณพิกัด y ของจุดบนเส้นโค้ง

โดยทั่วไปการคำนวณหาผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะจำเป็นต้องมีการคำนวณ Point Addition และ Point Doubling จำนวนหลายครั้ง ส่งผลให้จำเป็นต้องคำนวณหาค่าผกผันเหนือฟิลด์จำนวนเฉพาะ การคูณเหนือฟิลด์จำนวนเฉพาะ และกำลังสองสมบูรณ์จำนวนหลายครั้งเช่นกัน สำหรับหัวข้อนี้จะกล่าวถึงวิธีการคำนวณหา $2P+Q$ โดยการตัดการคำนวณพิกัด y ซึ่งเป็นผลลัพธ์จากการคำนวณ Point Addition ออกจากการคำนวณ ซึ่งจะช่วยให้สามารถลดทรัพยากรสำหรับการคำนวณการคูณเหนือฟิลด์จำนวนเฉพาะลงได้

เนื่องจาก Point Addition ใช้ทรัพยากรสำหรับการคำนวณน้อยกว่า Point Doubling ดังนั้นการคำนวณหา $2P+Q$ จึงควรใช้ Point Addition จำนวน 2 ครั้งคือ $R = P+Q$ และ $S = R+P = P+Q+P = 2P+Q$ ซึ่งจะใช้ทรัพยากรเพียงดังต่อนี้ ค่าผกผันเหนือฟิลด์จำนวนเฉพาะจำนวน 2 ครั้ง การคูณเหนือฟิลด์จำนวนเฉพาะ 4 ครั้ง และกำลังสองสมบูรณ์ 2 ครั้ง

อย่างไรก็ตามการคำนวณโดยการตัดพิกัด y [33] ออกจากการคำนวณสามารถลดทรัพยากรสำหรับการคำนวณให้ลดลงได้อีก กำหนดให้ $P = (x_p, y_p)$ และ $Q = (x_q, y_q)$ การคำนวณหา $2P+Q$ เป็นดังนี้

เริ่มจากการคำนวณหา $R = P+Q$ ดังนี้

$$m_r = \frac{y_q - y_p}{x_q - x_p} \bmod p$$

$$x_r = m_r^2 - x_p - x_q \text{ mop } p$$

และคำนวณหา $S = 2P+Q$ ดังนี้

$$m_s = \frac{y_r - y_p}{x_r - x_p} \bmod p$$

เนื่องจาก y_r ไม่ถูกคำนวณ จึงต้องปรับเปลี่ยนสมการสำหรับการคำนวณหา m_s ใหม่ดังนี้

จาก $y_r = m_r(x_p - x_r) - y_p \bmod p$

ดังนั้น $$m_s = \frac{m_r(x_p - x_r) - y_p - y_p}{x_r - x_p} \bmod p$$

$$= -m_r - \frac{2y_p}{x_r - x_p} \bmod p \qquad (9.9)$$

จากหลักการดังกล่าวข้างต้นสังเกตได้ว่าสามารถลดขั้นตอนการคำนวณการคูณเหนือฟิลด์จำนวนเฉพาะลงได้ 1 ขั้นตอน

ตัวอย่างที่ 9.7 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ และจุด P ที่ได้จากตัวอย่างที่ 9.1 จงคำนวณหา $2P+Q$ เมื่อกำหนดให้ $P = (17, 3)$ และ $Q = (15, 11)$

วิธีทำ จากตัวอย่างที่ 9.3 ได้แสดงวิธีการคำนวณหา $R = P+Q$ เรียบร้อยแล้วโดย $m_r = 49$, $x_r = 37$ และ $y_r = 24$ อย่างไรก็ตามจากตัวอย่างนี้สมมติว่า y_r ไม่ถูกคำนวณออกมา (m_r และ x_r ยังคงมีค่าคงเดิมตามตัวอย่าง 9.3) ดังนั้น $S = 2P + Q$ สามารถคำนวณออกมาได้เป็นดังต่อไปนี้

จาก $$m_s = -m_r - \frac{2y_p}{x_r - x_p} \bmod p$$

$$= -49 - \frac{2 \times 3}{37-17} \bmod 53$$

$= 4 - 6 \times 20^{-1} \bmod 53$

$= 4 - 6 \times 8 \bmod 53$

$= 4 - 48 \bmod 53$

$= -44 \bmod 53$

$= 9 \bmod 53$

ดังนั้น $x_s = m_s^2 - x_r - x_p \bmod p$

$= 9^2 - 37 - 17 \bmod 53$

$= 27$

และ $y_s = m_s(x_p - x_s) - y_p \bmod p$

$= 9 \times (17 - 27) - 3 \bmod 53$

$= 13$

ดังนั้นจุดผลลัพธ์ของ $2P+Q$ บนเส้นโค้งเชิงวงรีคือ (27, 13)

อย่างไรก็ตามการคำนวณหา $2P+Q$ โดยการตัดการคำนวณพิกัด y ออกจากการคำนวณ 1 ครั้งดังกล่าวข้างต้นนี้เป็นเพียงวิธีหนึ่งที่สามารถนำมาใช้สำหรับการลดทรัพยากรสำหรับการคำนวณลงได้ โดยยังมีเทคนิควิธีการอื่นๆ อีกเป็นจำนวนมากที่สามารถนำมาใช้ลดทรัพยากรการคำนวณลงได้ ซึ่งเทคนิควิธีเหล่านี้ได้ถูกเผยแพร่ลงทั้งรูปแบบวารสาร และสิ่งตีพิมพ์จากงานประชุมวิชาการ

3.6 การวิเคราะห์หาจำนวนจุดบนเส้นโค้งเหนือฟิลด์จำนวนเฉพาะ

การวิเคราะห์หาจำนวนจุดที่เป็นไปได้ทั้งหมดจากสมการที่ (9.1) สามารถทำได้โดยการแทนค่า $x = 0, 1, 2, \cdots, p-1$ ในสมการ และคำนวณหาค่า y^2 ที่เป็นกำลังสองสมบูรณ์เหนือฟิลด์ GF(p) ซึ่งโดยทั่วไปจาก x ที่ถูกนำมาแทนค่าในสมการทั้งหมดนั้นจะพบค่า y^2 ที่เป็นกำลังสองสมบูรณ์อยู่ครึ่งหนึ่ง และเนื่องจาก y^2 ที่เป็นกำลังสองสมบูรณ์จะมีรากของสมการ 2 คำตอบคือ y และ $-y$ ดังนั้นจำนวนจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์ GF(p) เป็นไปได้สูงสุดมีค่าประมาณ $p+1$ จุด

ต่อไปจะกล่าวถึงทฤษฎีของ เฮลมัต แฮซซี (Helmut Hasse) [62] ซึ่งเกี่ยวกับจำนวนจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์ GF(p)

ทฤษฎีบทที่ 9.1 กำหนดให้สมการเส้นโค้งเชิงวงรี $y^2 = x^3 + ax + b \bmod p$ มีจำนวนจุดบนเส้นโค้งทั้งหมด N จุด ได้ว่า

$$\left|N - p - 1\right| < 2\sqrt{p}$$

ดังนั้นจึงสามารถใช้ทฤษฎีบทข้างต้นสำหรับประมาณจำนวนจุดที่เป็นไปได้ทั้งหมดบนเส้นโค้งเชิงวงรีได้ โดย N จะมีค่าอยู่ในช่วงดังต่อไปนี้

$$(p+1) - 2\sqrt{p} < N < (p+1) + 2\sqrt{p}$$

3.7 การประยุกต์เส้นโค้งเหนือฟิลด์จำนวนเฉพาะสำหรับกระบวนการเข้ารหัสข้อมูล

ในช่วงปี ค.ศ.1993 อัลเฟรด มีนีซีส (Alfred Menezes) และ สกอตต์ แวนสโทน (Scott Vanstone) [55] ได้นำเสนอการนำสมการเส้นโค้งเชิงวงรีเหนือฟิลด์ GF(p) มาประยุกต์ใช้สำหรับกระบวนการเข้ารหัสลับในรูปแบบกุญแจสาธารณะโดยแบ่งออกเป็น 3 กระบวนการดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจ หรือผู้รอรับข้อความไซเฟอร์ (กำหนดเป็น ผู้รับ) มีลำดับการทำงานเป็นดังนี้

1. เลือกสมการ $y^2 = x^3 + ax + b \bmod p$
2. เลือกจุดกำเนิดบนสมการเส้นโค้งจากขั้นตอนที่ 1 กำหนดเป็น $P = (x_p, y_p)$
3. เลือก $a \in \{2, 3, 4, \cdots, N - 1\}$
4. คำนวณจุด $Q = aP$

กุญแจสาธารณะคือ $\{P, Q\}$

กุญแจส่วนตัวคือ $\{a\}$

กระบวนการที่ 2 การเข้ารหัสลับ: เป็นกระบวนการที่ถูกดำเนินการโดยบุคคลผู้ซึ่งต้องการส่งข้อความลับ (สมมติข้อความลับคือ $m = (x_m, y_m)$) ไปยังผู้รับ โดยมีลำดับการทำงานเป็นดังนี้

1. เลือก $k \in \{2, 3, 4, \cdots, N - 1\}$
2. คำนวณ $R = kQ = (x_r, y_r)$
3. คำนวณ $y_0 = kP$
4. คำนวณ $y_1 = x_r x_m \bmod p$
5. คำนวณ $y_2 = y_r y_m \bmod p$

โดยข้อความไซเฟอร์ที่จะถูกส่งไปยังผู้รับคือ (y_0, y_1, y_2) โดย y_0 คือจุดบนเส้นโค้ง ส่วน y_1 และ y_2 คือจำนวนเต็ม

กระบวนการที่ 3 การถอดรหัสลับ: หลังจากที่ผู้รับได้รับ (y_0, y_1, y_2) จากผู้ส่งจะสามารถคำนวณหา m ได้โดยสมการต่อไปนี้

1. คำนวณ $(x_r, y_r) = ay_0$
2. คำนวณ $x_m = y_1 x_r^{-1} \bmod p$

3. คำนวณ $y_m = y_2 y_r^{-1} \bmod p$

ดังนั้นผู้รับทราบ $m = (x_m, y_m)$

อย่างไรก็ตามทั้ง 3 ขั้นตอนจากกระบวนการถอดรหัสพิสูจน์ได้ดังนี้

พิสูจน์ขั้นตอนที่ 1

จาก
$$\begin{aligned} R &= kQ \\ &= k(aP) \\ &= akP \\ &= ay_0 \end{aligned}$$
□

พิสูจน์ขั้นตอนที่ 2

$$\begin{aligned} y_1 x_r^{-1} \bmod p &= x_r x_m x_r^{-1} \bmod p \\ &= x_m x_r x_r^{-1} \bmod p \\ &= x_m \bmod p \end{aligned}$$
□

พิสูจน์ขั้นตอนที่ 3

$$\begin{aligned} y_2 y_r^{-1} \bmod p &= y_r y_m y_r^{-1} \bmod p \\ &= y_m y_r y_r^{-1} \bmod p \\ &= y_m \bmod p \end{aligned}$$
□

ตัวอย่างที่ 9.8 การประยุกต์ใช้วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะสำหรับกระบวนการเข้ารหัส

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

1. เลือกสมการ $y^2 = x^3 + 23x + 5 \bmod 53$
2. เลือกจุด $P = (17, 3)$
3. เลือก $a = 3$
4. คำนวณ $Q = 3P = (37, 24)$

กระบวนการเข้ารหัส

สมมติผู้ส่งต้องการเข้ารหัส m = (15, 11)

1. เลือก k = 4
2. คำนวณ $R = 4Q$ = (39, 15)
3. คำนวณ $y_0 = 4P$ = (27, 13)
4. คำนวณ

$$y_1 = x_r x_m \bmod p$$
$$= 39 \times 15 \bmod 53$$
$$= 2$$

5. คำนวณ

$$y_2 = y_r y_m \bmod p$$
$$= 15 \times 11 \bmod 53$$
$$= 6$$

ดังนั้นข้อความไซเฟอร์คือ ((27, 13), 2, 6)

กระบวนการถอดรหัส

1. คำนวณ $(x_r, y_r) = ay_0 = 3y_0$ = (39, 15)
2. คำนวณ

$$x_m = y_1 x_r^{-1} \bmod p$$
$$= 2 \times 39^{-1} \bmod 53$$
$$= 2 \times 34 \bmod 53$$
$$= 15$$

3. คำนวณ

$$y_m = y_2 y_r^{-1} \bmod p$$
$$= 6 \times 15^{-1} \bmod 53$$
$$= 6 \times 46 \bmod 53$$
$$= 11$$

จากตัวอย่างที่ 9.8 สังเกตได้ว่าทั้งกระบวนการเข้ารหัสลับ และถอดรหัสลับด้วยวิทยาการรหัสลับเชิงโค้งเชิงวงรีไม่มีการคำนวณเลขยกกำลัง แต่ใช้กระบวนการคูณระหว่างจำนวนเต็มและจุด

บนเส้นโค้งแทน โดยหากนำสมการเส้นโค้งเชิงวงรีไปประยุกต์ใช้งานกับกระบวนการเข้ารหัสและถอดรหัสข้อมูลจริง ขนาดกุญแจไม่ควรต่ำกว่า 160 บิต

3.8 การประยุกต์เส้นโค้งเหนือฟิลด์จำนวนเฉพาะสำหรับกระบวนการแลกเปลี่ยนกุญแจ

บทที่ 6 ได้กล่าวถึงการนำขั้นตอนวิธีดิฟฟี่เฮลแมนมาประยุกต์ใช้สำหรับกระบวนการแลกเปลี่ยนกุญแจซึ่งความปลอดภัยขึ้นอยู่กับปัญหาวิยุตลอการิทึม นอกเหนือจากนี้ยังสามารถนำขั้นตอนวิธีดังกล่าวมาประยุกต์ใช้กับวิทยาการรหัสลับเส้นโค้งเชิงวงรีได้

สมมติ นาย ก และ นาย ข ประสงค์ที่จะแลกเปลี่ยนกุญแจลับซึ่งกันและกันโดยใช้ขั้นตอนวิธีดิฟฟี่เฮลแมนด้วยเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะจะมีขั้นตอนเป็นดังนี้

ขั้นตอนที่ 1: นาย ก และ นาย ข ตกลงเลือกใช้สมการ $y^2 = x^3 + ax + b \bmod p$ และจุดบนเส้นโค้ง $P = (x_p, y_p)$

ขั้นตอนที่ 2: นาย ก สุ่มเลือกจำนวนเต็ม $a \in \{0, 1, 2, \cdots, N-1\}$ และคำนวณ K_A จาก

$$K_A = aP$$

โดยนาย ก จะเก็บ a ไว้เป็นความลับ แต่จะส่ง K_A ไปยังนาย ข

ขั้นตอนที่ 3: นาย ข สุ่มเลือกจำนวนเต็ม $b \in \{0, 1, 2, \cdots, N-1\}$ และคำนวณ K_B จาก

$$K_B = bP$$

โดยนาย ข จะเก็บ b ไว้เป็นความลับ แต่จะส่ง K_B ไปยังนาย ก

ขั้นตอนที่ 4: นาย ก นำค่า K_B ที่รับมาจากนาย ข และคำนวณ $K_ก$ จาก

$$K_ก = aK_B$$
$$= abP$$

ขั้นตอนที่ 5: นาย ข นำค่า K_A ที่รับมาจากนาย ก และคำนวณ $K_ข$ จาก

$$K_ข = bK_A$$
$$= baP$$
$$= abP$$

จากขั้นตอนที่ 4 และ 5 สรุปได้ว่า

$$K_ก = K_ข = K$$

ดังนั้น K คือกุญแจลับที่เกิดจากการแลกเปลี่ยนระหว่างนาย ก และนาย ข ได้อย่างปลอดภัย

จากกระบวนการข้างต้น หากผู้ไม่ประสงค์ดีต้องการทราบ K จำเป็นต้องคำนวณหาค่าใดค่าหนึ่งระหว่าง a หรือ b ซึ่งเป็นไปได้ยากมาก เนื่องจากจำเป็นต้องแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี

ตัวอย่างที่ 9.9 การประยุกต์ใช้วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะสำหรับกระบวนการแลกเปลี่ยนกุญแจด้วยขั้นตอนวิธีดิฟฟี่เฮลแมน

วิธีทำ

ขั้นตอนที่ 1: นาย ก และ นาย ข ตกลงใช้สมการ $y^2 = x^3 + 23x + 5 \bmod 53$ และจุด (17, 3) ร่วมกัน

ขั้นตอนที่ 2: นาย ก เลือก $a = 11$ และ คำนวณ

$$K_A = 11P$$
$$= (13, 40)$$

และส่ง K_A ไปยัง นาย ข

ขั้นตอนที่ 3: นาย ข เลือก $b = 4$ และ คำนวณ

$$K_B = 4P$$
$$= (27, 13)$$

และส่ง K_B ไปยัง นาย ก

ขั้นตอนที่ 4: นาย ก คำนวณ $K_ก = aK_B = 11K_B = (16, 21)$

ขั้นตอนที่ 5: นาย ข คำนวณ $K_ข = bK_A = 4K_A = (16, 21)$

ดังนั้นสรุปได้ว่า $K = (16, 21)$ คือกุญแจลับที่ใช้ร่วมกันระหว่างนาย ก และ นาย ข

จากตัวอย่างที่ 9.9 หากผู้ไม่ประสงค์ดีต้องการทราบค่า K จำเป็นต้องแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี โดยไม่จำเป็นต้องทราบทั้งค่า a และ b เนื่องจากผลลัพธ์ของทั้งสองสมการมีค่าเท่ากัน ดังนั้นจึงสามารถคำนวณหาเพียงค่าใดค่าหนึ่ง เช่นหากทราบค่า a จะสามารถคำนวณหา K ได้จาก $K = aK_B$ เป็นต้น อย่างไรก็ตามหากขนาดกุญแจที่ใช้งานมีขนาดใหญ่ (ไม่ควรต่ำกว่า 160 บิต) การแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรีเป็นเรื่องที่ยากมาก

4. วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง

เนื่องจากการวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะมีโครงสร้างพื้นฐานทางฮาร์ดแวร์ไม่เหมือนกับระบบคอมพิวเตอร์ซึ่งการประมวลผลจะมีเพียงรหัสตัวเลขที่มีสมาชิกคือ 0 หรือ 1 เท่านั้น ดังนั้นถึงแม้ว่าการนำวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะมาประยุกต์ใช้งานในระบบคอมพิวเตอร์จะมีความซับซ้อนไม่สูงมากและมีความปลอดภัยสูง แต่เวลาที่ใช้สำหรับการประมวลผลสูง ในหัวข้อนี้จะกล่าวถึงวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองซึ่งเป็นระบบวิธีที่มีโครงสร้างตรงกับระบบคอมพิวเตอร์ซึ่งจะช่วยลดความซับซ้อน

ของเวลาลงได้ อย่างไรก็ตามขั้นตอนวิธีดังกล่าวนี้มีความซับซ้อนทางการคำนวณสูง ส่งผลให้วิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองยังคงอยู่ในช่วงระหว่างการพัฒนา

สมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองคือคู่อันดับ (x, y) ที่เป็นไปได้ทั้งหมดที่ทำให้สมการต่อไปนี้เป็นจริง

$$y^2 + xy = x^3 + ax^2 + b \tag{9.10}$$

เมื่อ a, b อยู่ในฟิลด์ GF(2^m) และ $b \neq 0$

4.1 การสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง

ลำดับขั้นตอนการสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองจะคล้ายกับการสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะคือเริ่มจากเลือกคู่อันดับ (x, y), a และ ฟังก์ชันพหุนามไม่ลดรูปตามลำดับ แล้วจึงคำนวณหา b ที่ทำให้สมการเป็นจริง

ตัวอย่าง 9.10 การทดสอบสร้างสมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง

วิธีทำ

สมมติผู้ใช้งานเลือกพารามิเตอร์ทั้งหมดเป็นดังนี้

$P = (x_p, y_p) = (x^2 + x, x^3)$, $a = x+1$, GF(2^4) = $a_3x^3 + a_2x^2 + a_1x + a_0$ และ $f(x) = x^4 + x + 1$

ดังนั้นจากสมการ (9.11) ได้ว่า

$$y_p^2 + x_p y_p = x_p^3 + ax_p^2 + b$$

$$(x^3)^2 + (x^2+x)x^3 = (x^2 + x)^3 + (x+1)(x^2+x)^2 + b$$

$$x^6 + (x^5 + x^4) = (x^6 + x^5 + x^4 + x^3) + (x^5 + x^4 + x^3 + x^2) + b$$

$$x^6 + x^5 + x^4 = x^6 + x^5 + x^4 + x^3 + x^5 + x^4 + x^3 + x^2 + b$$

$$x^6 + x^5 + x^4 = x^6 + x^2 + b$$

ดังนั้น $$b = x^5 + x^4 - x^2$$

$$= x^5 + x^4 + x^2$$

เนื่องจากค่า b อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงจำเป็นต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$b = x^5 + x^4 + x^2 \bmod x^4 + x + 1$$

$= 1$

จึงได้สมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองเป็นดังนี้

$$y^2 + xy = x^3 + (x+1)x^2 + 1$$

และมี $(x^2 + x, x^3)$ เป็นคู่อันดับหนึ่งที่อยู่บนเส้นโค้งนี้

4.2 การคำนวณหาผลบวกระหว่างจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง

การคำนวณหา $R = P + Q$ เมื่อ $P = (x_p, y_p)$, $Q = (x_q, y_q)$, $R = (x_r, y_r)$ จะสามารถคำนวณหา x_r และ y_r ได้ ดังนี้

กรณีที่ $P \neq Q$, $x_r = m^2 + m + x_q + x_p + a$ (9.11)

และ $y_r = m(x_p + x_r) + x_r + y_p$ (9.12)

กรณีที่ $P = Q$, $x_r = m^2 + m + a$ (9.13)

และ $y_r = x_p^2 + (m+1)x_r$ (9.14)

เมื่อ m คือค่าความชันของเส้นตรงที่ลากผ่านจุดกำเนิด ซึ่งแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1 (Point Addition): $P \neq Q$

$$m = \frac{y_p + y_q}{x_p + x_q} \tag{9.15}$$

เมื่อ $(x_p + x_q)^{-1}$ คือค่าผกผันธ์เหนือฟิลด์ GF(2^m) ดังนั้น $(x_p + x_q)^{-1}(x_p + x_q) \bmod f(x) = 1$

กรณีที่ 2 (Point Doubling): $P = Q$

$$m = x_p + \frac{y_p}{x_p} \tag{9.16}$$

เมื่อ x_p^{-1} คือค่าผกผันธ์เหนือฟิลด์ GF(2^m) ดังนั้น $(x_p^{-1})(x_p) \bmod f(x) = 1$

ตัวอย่างที่ 9.11 จากสมการเส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสอง และจุด P ที่ได้จากตัวอย่างที่ 9.10 จงคำนวณหา $Q = 2P$

วิธีทำ

เนื่องจาก $Q = 2P$, ดังนั้นจึงต้องใช้ Point Doubling

เริ่มจากการคำนวณหาค่า m โดยใช้สมการที่ (9.16) ได้ดังนี้

$$m = x_p + \frac{y_p}{x_p}$$

$$= (x^2 + x) + \frac{x^3}{x^2 + x}$$

เนื่องจาก $(x^2 + x + 1) = (x^2 + x)^{-1} \bmod x^4 + x + 1$

ดังนั้น $m = (x^2 + x) + x^3(x^2 + x + 1)$

$= x^5 + x^4 + x^3 + x^2 + x$

เนื่องจากค่า m อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$m = x^5 + x^4 + x^3 + x^2 + x \bmod x^4 + x + 1$$

$$= x^3 + x + 1$$

และจากสมการ (9.13) และ (9.14) ได้ว่า

$$x_q = m^2 + m + a$$

$$= x^6 + x^2 + 1 + x^3 + x + 1 + x+1$$

$$= x^6 + x^3 + x^2 + 1$$

เนื่องจากค่า x_q อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$x_q = x^6 + x^3 + x^2 + 1 \bmod x^4 + x + 1$$

$$= 1$$

และ $y_q = x_p^2 + (m+1)x_q$

$$= (x^2 + x)^2 + ((x^3 + x + 1) + 1)(1)$$

$$= x^4 + x^3 + x^2 + x$$

เนื่องจากค่า y_q อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$y_q = x^4 + x^3 + x^2 + x \bmod x^4 + x + 1$$
$$= x^3 + x^2 + 1$$

ดังนั้นสรุปได้ว่า $Q = (1, x^3 + x^2 + 1)$

ตัวอย่างที่ 9.12 จากผลลัพธ์ตัวอย่างที่ 9.11 จงคำนวณหา $R = P + Q$

วิธีทำ

เนื่องจาก $R = P + Q$, $P \neq Q$, ดังนั้นจึงต้องใช้ Point Addition

เริ่มจากการคำนวณหาค่า m โดยใช้สมการที่ (9.15) ได้ดังนี้

$$m = \frac{y_p + y_q}{x_p + x_q}$$
$$= \frac{x^3 + x^3 + x^2 + 1}{x^2 + x + 1}$$
$$= \frac{x^2 + 1}{x^2 + x + 1}$$

เนื่องจาก $(x^2 + x) = (x^2 + x + 1)^{-1} \bmod x^4 + x + 1$

ดังนั้น $m = (x^2 + 1)(x^2 + x)$

$= x^4 + x^3 + x^2 + x$

เนื่องจากค่า m อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$m = x^4 + x^3 + x^2 + x \bmod x^4 + x + 1$$
$$= x^3 + x^2 + 1$$

และจากสมการ (9.11) และ (9.12) ได้ว่า

$$x_r = m^2 + m + x_q + x_p + a$$
$$= (x^3 + x^2 + 1)^2 + (x^3 + x^2 + 1) + 1 + (x^2 + x) + (x + 1)$$
$$= x^6 + x^4 + 1 + x^3 + x^2 + 1 + 1 + x^2 + x + x + 1$$
$$= x^6 + x^4 + x^3$$

เนื่องจากค่า x_r อยู่เกินขอบเขตของ GF(2^4) ดังนั้นจึงต้องลดรูปโดยใช้ฟังก์ชันพหุนามไม่ลดรูปได้ดังนี้

$$x_r = x^6 + x^4 + x^3 \bmod x^4 + x + 1$$
$$= x^2 + x + 1$$

และ
$$y_r = m(x_p + x_r) + x_r + y_p$$
$$= (x^3 + x^2 + 1)(x^2 + x + x^2 + x + 1) + x^2 + x + 1 + x^3$$
$$= x^3 + x^2 + 1 + x^2 + x + 1 + x^3$$
$$= x$$

ดังนั้นสรุปได้ว่า $R = (x^2 + x + 1 , x)$

สำหรับการคำนวณหาผลคูณระหว่างจำนวนเต็มและจุดบนเส้นโค้ง กระบวนการเข้ารหัส กระบวนการแลกเปลี่ยนกุญแจ และลายเซ็นดิจิทัลโดยใช้เส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองจะใช้หลักการเช่นเดียวกับการดำเนินการดังกล่าวด้วยเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ เพียงแต่การดำเนินการโดยใช้เส้นโค้งเชิงวงรีเหนือฟิลด์ลักษณะเฉพาะสองจะดำเนินการภายในขอบเขต GF(2^m) และใช้สมการ (9.12) – (9.17) สำหรับการคำนวณ Point Addition หรือ Point Doubling

5. การแยกตัวประกอบด้วยเส้นโค้งเชิงวงรี

นอกจากการนำเส้นโค้งเชิงวงรีไปประยุกต์ใช้เป็นวิทยาการรหัสลับแบบกุญแจสาธารณะประเภทหนึ่งแล้ว ยังสามารถนำเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะไปประยุกต์ใช้สำหรับการแยกตัวประกอบจำนวนเต็มได้ [25] ซึ่งหมายความว่าสามารถใช้เส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะสำหรับโจมตีวิทยาการรหัสลับอาร์เอสเอได้เช่นกัน สำหรับการแยกตัวประกอบด้วยเส้นโค้งเชิงวงรีจะใช้ n เป็นค่ามอดูลัสแทนจำนวนเฉพาะจึงได้สมการเส้นโค้งเชิงวงรีสำหรับการแยกตัวประกอบเป็นดังนี้

$$y^2 \equiv x^3 + ax + b \bmod n \qquad (9.17)$$

เมื่อ $n = pq$

การดำเนินการเริ่มจากเลือกสมการเส้นโค้งเชิงวงรีโดยจากสมการที่ (9.17) ควรเลือกจุดพิกัด (P) ที่อยู่บนเส้นโค้งก่อน และตามด้วยเลือกค่า a แล้วจึงคำนวณหา b หลังจากได้สมการและจุดพิกัดเรียบร้อยแล้วให้ดำเนินการคำนวณ $2P, 3P, 4P, \cdots$ จนกระทั่งพบจุด $iP = \infty$ ซึ่งจะทำให้ตัวประกอบของ n ถูกเปิดเผยเนื่องจากค่าหารร่วมมากระหว่างตัวหารของค่าความชัน และ n มีค่าไม่เท่ากับ 1 ดังนั้นผลลัพธ์ดังกล่าวจึงมีเป็นตัวประกอบค่าหนึ่งของ n

ตัวอย่างที่ 9.13 จงแยกตัวประกอบ $n = 21$ ด้วยเส้นโค้งเชิงวงรี

วิธีทำ จากสมการที่ (9.18) ได้ว่า

$$y^2 \equiv x^3 + ax + b \bmod 21$$

เลือกจุด $P = (2, 4) = (x_1, y_1)$ ได้ว่า

$$4^2 \equiv 2^3 + 2a + b \bmod 21$$

$$16 \equiv 8 + 2a + b \bmod 21$$

เลือก $a = 7$, $16 \equiv 8 + 2 \times 7 + b \bmod 21$

$$16 \equiv 22 + b \bmod 21$$

$$-6 \equiv b \bmod 21$$

หรือ $b \equiv 15 \bmod 21$

ดังนั้น $y^2 \equiv x^3 + 7x + 15 \bmod 21$ และจุด $P = (2, 4)$

คำนวณ $2P = P + P = (x_2, y_2)$, Point Doubling

$$m = \frac{3x^2 + a}{2y} \bmod p$$

$$= \frac{3x2^2 + 7}{2x4} \bmod 21$$

$$= \frac{19}{8} \bmod 21$$

$$= 19 \times 8^{-1} \bmod 21$$

$$= 19 \times 8 \bmod 21$$

$$= 5$$

จาก $x_2 = m^2 - 2x_1 \bmod p$

$$= 5^2 - 2 \times 2 \bmod 21$$

$$= 0$$

และ $y_2 = m(x_1 - x_2) - y_1 \bmod p$

$$= 5 \times (2 - 0) - 4 \bmod 21$$

$$= 6$$

ดังนั้น $2P = (0, 6)$

คำนวณ $3P = P + 2P = (x_3, y_3)$, Point Addition

$$m = \frac{y_2 - y_1}{x_2 - x_1} \bmod p$$

$$= \frac{6 - 4}{0 - 2} \bmod 21$$

$$= \frac{2}{(-2)} \bmod 21$$

$$= 2 \times 19^{-1} \bmod 21$$

$$= 2 \times 10 \bmod 21$$

$$= 20$$

$$x_3 = m^2 - x_2 - x_1 \bmod p$$

$$= 20^2 - 0 - 2 \bmod 21$$

$$= 20$$

และ $y_3 = m(x_1 - x_3) - y_1 \bmod p$

$$= 20 \times (2 - 20) - 4 \bmod 21$$

$$= 14$$

ดังนั้นสรุปได้ว่า $3P = (20, 14)$

คำนวณ $4P = P + 3P = (x_4, y_4)$, Point Addition

$$m = \frac{y_3 - y_1}{x_3 - x_1} \bmod p$$

$$= \frac{14 - 4}{20 - 2} \bmod 21$$

$$= \frac{11}{18} \bmod 21$$

$$= 11 \times 18^{-1} \bmod 21$$

อย่างไรก็ตามเนื่องจาก gcd(18, 21) = 3 ดังนั้นจึงได้ว่า 3 เป็นตัวประกอบค่าหนึ่งของ 21 และสามารถคำนวณหาตัวประกอบอีกค่าได้จาก $\frac{21}{3} = 7$

นอกเหนือจากนั้นจากผลลัพธ์ที่ได้นี้ได้ว่า $4P = \infty$

การเลือกจุดพิกัดเริ่มต้นเป็นอีกปัจจัยที่สำคัญสำหรับการแยกตัวประกอบด้วยเส้นโค้งเชิงวงรี เนื่องจากหากเลือกจุดได้เหมาะสม จะส่งผลให้สามารถแยกตัวประกอบได้เร็วขึ้น ยกตัวอย่างเช่นจากตัวอย่างที่ 9.13 หากเปลี่ยนจุดพิกัดเป็น $P' = (9, 3)$ ซึ่งเป็นจุดหนึ่งของสมการ $y^2 \equiv x^3 + 7x + 15 \bmod 21$ เนื่องจาก

$$3^2 \equiv 9^3 + 7 \times 9 + 15 \bmod 21$$
$$9 \equiv 729 + 63 + 15 \bmod 21$$
$$\equiv 807 \bmod 21$$
$$\equiv 9 \bmod 21$$

จะพบว่าสามารถพบตัวประกอบได้ตั้งแต่ครั้งแรกที่คำนวณ $2P'$ เนื่องจาก gcd($2y$, n) = gcd(6, 21) = 3

อย่างไรก็ตามการแยกตัวประกอบด้วยเส้นโค้งเชิงวงรีจะมีประสิทธิภาพที่สูงในกรณีเพียงแค่จำนวนตัวเลขของ n มีค่าอยู่ระหว่าง 40 – 50 ตัวเท่านั้น ซึ่งในทางปฏิบัติจริงค่า n มีขนาดอย่างน้อย 1024 บิตซึ่งมีจำนวนตัวเลขสูงกว่านี้เป็นอย่างมาก ดังนั้นจึงกล่าวได้ว่าเส้นโค้งเชิงวงรียังคงไม่สามารถนำมาใช้สำหรับการโจมตีวิทยาการรหัสลับอาร์เอสเอที่ใช้อยู่จริงได้เช่นกัน

6. บทสรุปสาระสำคัญ

จากปัญหาของวิทยาการรหัสอาร์เอสเอที่จำเป็นต้องใช้กุญแจขนาดใหญ่จึงไม่เหมาะที่จะนำมาประยุกต์ใช้งานร่วมกับอุปกรณ์สื่อสารที่ใช้หน่วยประมวลผลที่มีประสิทธิภาพต่ำ เช่นสมาร์ทโฟน อย่างไรก็ตามปัจจุบันการติดต่อสื่อสารด้วยสมาร์ทโฟนกำลังได้รับความนิยมสูงมากเนื่องจากมีขนาดเล็กและพกพาง่าย จึงเกิดวิทยาการรหัสลับเส้นโค้งเชิงวงรีที่เป็นวิทยาการรหัสลับแบบกุญแจสาธารณะอีกประเภทหนึ่งที่มีความปลอดภัยสูง จุดเด่นคือความปลอดภัยสูงมากเทียบเท่ากับวิทยาการรหัสลับอาร์เอสเอถึงแม้ว่ากุญแจที่นำมาใช้งานมีขนาดเล็กลงเป็นอย่างมาก ยกตัวอย่างเช่นการใช้งานวิทยาการรหัสสลับเส้นโค้งเชิงวงรีที่มีกุญแจขนาด 160 บิตมีความปลอดภัยเทียบเท่ากับกุญแจที่มีขนาด 1024 บิตสำหรับวิทยาการรหัสลับอาร์เอสเอ ส่งผลให้เวลาที่ใช้สำหรับการประมวลผลทั้งกระบวนการเข้ารหัสลับ และกระบวนการถอดรหัสลับลดลงเป็นอย่างมาก ดังนั้นวิทยาการรหัสลับเส้นโค้งเชิงวงรีจึงเหมาะสมที่จะถูกนำมาประยุกต์ใช้งานกับอุปกรณ์ประมวลผลที่มีประสิทธิภาพต่ำ

ความยากของการโจมตีวิทยาการรหัสลับเส้นโค้งเชิงวงรีขึ้นอยู่กับการแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี คือหากผู้ไม่ประสงค์ดีต้องการคำนวณหากุญแจส่วนตัวจำเป็นต้องคำนวณหาตัวคูณของจุดๆ หนึ่งที่อยู่บนเส้นโค้งเชิงวงรีซึ่งเป็นปัญหาที่ยากมาก ถึงแม้ว่าปัจจุบันจะมีขั้นตอนวิธีที่ใช้สำหรับการแก้ปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรีถูกนำเสนอออกมาเป็นจำนวนมาก ซึ่งโดยส่วนใหญ่จะใช้ขั้นตอนวิธีเดียวกันกับการแก้ปัญหาวิยุตลอการิทึมเพียงแต่ปรับเปลี่ยนรูปแบบสมการของการคำนวณ เช่น การโจมตีแบบตะลุย ขั้นตอนวิธีเบบี้สเต็พไจแอ็นสเต็พ และ ขั้นตอนวิธีโพลิกเฮลแมน เป็นต้น แต่ยังไม่พบขั้นตอนวิธีใดที่สามารถแก้ปัญหาดังกล่าวได้ในระยะเวลาอันสั้น

นอกจากการนำเส้นโค้งเชิงวงรีไปประยุกต์ใช้เป็นวิทยาการรหัสลับแบบกุญแจสาธารณะประเภทหนึ่งแล้ว ยังสามารถนำเส้นโค้งเชิงวงรีไปประยุกต์ใช้สำหรับการแยกตัวประกอบจำนวนเต็มได้ แต่อย่างไรก็ตามการแยกตัวประกอบด้วยเส้นโค้งเชิงวงรีจะมีประสิทธิภาพที่สูงในกรณีเพียงแค่จำนวนตัวเลขของมอดูลัสมีค่าอยู่ระหว่าง 40 – 50 ตัวเท่านั้น

แบบฝึกหัดท้ายบท
บทที่ 9

1. วิทยาการรหัสลับเส้นโค้งเชิงวงรีถูกนำเสนอโดยใคร
2. จุดเด่นของวิทยาการรหัสลับเส้นโค้งเชิงวงรีเหนือวิทยาการรหัสลับอาร์เอสเอคืออะไร
3. การดำเนินการบวกระหว่างจุดเหนือเส้นโค้งเชิงวงรีมีกี่วิธี และมีชื่อเรียกว่าอะไร
4. สามารถเลือกใช้สมการ $y^2 \equiv x^3 + 2x + 3 \bmod 251$ ได้หรือไม่ เพราะเหตุใด
5. กำหนดให้ (3, 7) คือจุดคำตอบของ $P + Q$ จงหาผลลัพธ์ของ $Q + P$
6. กำหนดให้ $P = (5, 12)$ คือจุดหนึ่งของเส้นโค้งเชิงวงรีเหนือฟิลด์ GF(31) จงหา $-P$
7. สมมติผู้ใช้งานเลือกพาราเตอร์ทั้งหมดเป็นดังนี้ $P = (x, y) = (2, 7)$, $a = 5$ และ $p = 59$ จงหา b
8. จากสมการเส้นโค้งเชิงวงรีข้อ 7 จงหา $2P$
9. จากสมการเส้นโค้งเชิงวงรีข้อ 7 และคำตอบของ $2P$ ข้อ 8 จงหา $3P$
10. จากสมการเส้นโค้งเชิงวงรีข้อ 7 จงหา $3P$ $(2P + P)$ โดยใช้วิธีการตัดการคำนวณพิกัด y ของจุดบนเส้นโค้ง
11. กำหนดให้ผู้ใช้เลือก $P = (x_p, y_p) = (x^2 + 1, x)$, $a = 1$, $GF(2^4) = a_3x^3 + a_2x^2 + a_1x + a_0$ และ $f(x) = x^4 + x + 1$ จงหา b
12. จงแยกตัวประกอบ $n = 6767441$ โดยใช้วิทยาการรหัสลับเส้นโค้งเชิงวงรี
13. จงแยกตัวประกอบ $n = 655217$ โดยใช้วิทยาการรหัสลับเส้นโค้งเชิงวงรี
14. การดำเนินการ Point Addition สำหรับขั้นตอนวิธีทวิภาคจะเกิดขึ้นช่วงใด
15. กำหนด P คือจุดบนเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะจุดหนึ่ง หากจำเป็นต้องคำนวณหาค่า $11P$ โดยใช้ขั้นตอนวิธีทวิภาค (ตรวจสอบตำแหน่งซ้ายสุดไปตำแหน่งขวาสุด) จำเป็นต้องมีการคำนวณ Point Addition จำนวนทั้งหมดกี่รอบเพราะเหตุใด
16. กำหนดให้สมการเส้นโค้งเชิงวงรีเป็นดังต่อไปนี้ $y^2 = x^3 + 17x + 38 \bmod 71$ จงใช้ทฤษฎีบทที่ 9.1 เพื่อประมาณจำนวนจุดที่เป็นไปได้ทั้งหมดบนเส้นโค้งเชิงวงรี

บทที่ 10
ฟังก์ชันแฮช และลายเซ็นดิจิทัล

ฟังก์ชันแฮช (Hash Function) หรือที่นิยมถูกเรียกอีกชื่อว่าแมสเสจไดเจสท์ (Message Digest) คือฟังก์ชันที่รับข้อมูลนำเข้าที่มีความยาวไม่แน่นอนสำหรับการประมวลผลและได้ผลลัพธ์เป็นข้อมูลที่มีความยาวคงที่ ซึ่งเรียกผลลัพธ์นี้ว่า ค่าแฮช (Hash Value) โดยฟังก์ชันแฮชเป็นฟังก์ชันแบบทิศทางเดียวกล่าวคือกระบวนการหาค่าแฮชของข้อความต้นฉบับสามารถดำเนินการได้อย่างง่ายดาย แต่ในทางกลับกันการคำนวณหาข้อความต้นฉบับจากค่าแฮชทำได้ยากมาก โดยฟังก์ชันแฮชที่ดีจะต้องมีคุณลักษณะปลอดการชน (Collision Free) กล่าวคือหากนำข้อความต้นฉบับจำนวนมากไปผ่านฟังก์ชันแฮชแล้วค่าแฮชของข้อความต้นฉบับแต่ละค่าต้องมีความแตกต่างกัน นอกเหนือจากนั้นถึงแม้ว่าข้อความต้นฉบับจะมีค่าที่ใกล้เคียงกัน ค่าแฮชของข้อความเหล่านั้นไม่จำเป็นต้องมีค่าใกล้เคียงกันดังเช่นข้อความต้นฉบับซึ่งเป็นคุณลักษณะเด่นอีกคุณลักษณะหนึ่งของฟังก์ชันแฮชที่สามารถช่วยหลีกเลี่ยงการโจมตีจากผู้ไม่ประสงค์ดีได้ ยกตัวอย่างเช่น ข้อความต้นฉบับคือ “tree” เมื่อนำไปผ่านฟังก์ชันแฮชจะได้ค่าแฮชเป็น “AABB11” ในทางกลับกันหากเติม “s” ต่อท้ายข้อความต้นฉบับซึ่งมีค่าใหม่เป็น “trees” หากนำค่าดังกล่าวนี้ไปผ่านฟังก์ชันแฮชจะได้ค่าแฮชมีค่าเป็น “FFEE33” ซึ่งสังเกตได้ว่าค่าแฮชของ “tree” และ “trees” มีความแตกต่างกันโดยสิ้นเชิง ถึงแม้ว่าข้อความต้นฉบับทั้งสองค่าจะมีความใกล้เคียงกัน

ฟังก์ชันแฮชได้ถูกพัฒนาออกมาหลายวิธี ยกตัวอย่างเช่น MD2 (Message Digest 2), MD4 (Message Digest 4), MD5 (Message Digest 5) และ SHA (Secure Hash Algorithm) โดยที่ MD2 คือฟังก์ชันแฮชที่ถูกนำเสนอโดย รอน ริเวสต์ ซึ่งค่าแฮชที่ได้จะมีขนาด 128 บิตจึงเป็นขั้นตอนวิธีที่มีความแข็งแกร่งมากในสมัยนั้น อย่างไรก็ตามปัญหาของ MD2 คือใช้เวลาในการประมวลผลที่สูง ในเวลาต่อมา รอน ริเวสต์ จึงปรับปรุง MD2 ใหม่เพื่อแก้ปัญหาในเรื่องของเวลาที่ใช้ในการประมวลผลได้เป็น MD4 ที่ได้ค่าแฮชขนาดคงเดิม ต่อมาพบว่า MD4 มีปัญหาในเรื่องของปัญหาการเกิดการชนกันบ่อยครั้งของค่าแฮชที่ได้ จึงปรับปรุงใหม่อีกครั้งเป็น MD5 และเนื่องจากขนาดของค่าแฮชยังคงเดิมคือ 128 บิตขั้นตอนวิธีดังกล่าวนี้จึงยังมีปัญหาเช่นเดียวกับ MD4 จึงได้มีการนำเสนออีกขั้นตอนวิธีคือ SHA ซึ่งเป็นขั้นตอนวิธีที่ปรับปรุงมาจาก MD4 โดยค่าแฮชมีขนาดสูงขึ้นเป็น 160 บิต วัตถุประสงค์สำคัญคือเพื่อใช้เป็นองค์ประกอบส่วนหนึ่งในการใช้งานร่วมกับการสร้างลายเซ็นดิจิทัล (Digital Signature) ให้เป็นไปตามมาตรฐานของลายเซ็นดิจิทัล (Digital Signature Standard, DSS) ต่อมา NIST จึงได้การปรับปรุง SHA ใหม่เล็กน้อยเพื่อเพิ่มประสิทธิภาพการใช้งานจึงได้ชื่อใหม่เป็น Secure

Hash Algorithm 1 (SHA-1) ถึงแม้ว่าปัจจุบัน SHA ได้ถูกปรับปรุงใหม่เพื่อเพิ่มประสิทธิภาพให้สูงขึ้น เช่น SHA-256 และ SHA-512 แต่การดำเนินการพื้นฐานทางคณิตศาสตร์ยังคงเดิมเช่น การดำเนินการแอนด์ การดำเนินการออร์ และการหมุน เป็นต้น ดังนั้นบทนี้จะนำเสนอขั้นตอนวิธี SHA-1 ซึ่งเป็นขั้นตอนวิธีหนึ่งที่มีความปลอดภัย เพื่อให้ผู้อ่านมีความรู้ความเข้าใจเกี่ยวกับฟังก์ชันแฮชและขั้นตอนวิธีของฟังก์ชันแฮชโดยนำเสนอเป็นส่วนแรก และส่วนที่สองจะนำเสนอลายเซ็นดิจิทัลซึ่งจำเป็นต้องนำฟังก์ชันแฮชมาประยุกต์ใช้งานด้วย

1. ขั้นตอนวิธี SHA-1

ขั้นตอนวิธี SHA-1 [60] คือฟังก์ชันแฮชรูปแบบหนึ่งที่มีความปลอดภัยสูงโดยข้อมูลนำเข้าจะมีขนาด 512 บิต และผลลัพธ์ที่ได้เป็นค่าแฮชที่มีขนาด 160 บิต ซึ่งหากข้อมูลมีขนาดเกินค่าที่กำหนดจำเป็นต้องแบ่งข้อมูลเป็นบล็อคที่มีขนาดบล็อคละ 512 บิตก่อนเพื่อนำข้อมูลเข้าสู่ SHA-1 ทีละ 1 บล็อค อย่างไรก็ตามกรณีที่ข้อมูลต้นฉบับมีขนาดไม่ถึง 512 บิตจำเป็นต้องเพิ่มบิตให้ครบตามกำหนดก่อนจึงจะสามารถนำข้อมูลเข้าสู่ SHA-1 ได้ สำหรับขั้นตอนวิธี SHA-1 ถูกแบ่งออกเป็น 3 ขั้นตอนดังนี้

ขั้นตอนที่ 1: การเพิ่มบิตเติมเต็ม

จากที่กล่าวข้างต้นแล้วว่าข้อมูลที่จะนำเข้า SHA-1 ต้องถูกแบ่งเป็นบล็อคๆ ละ 512 บิต โดยบล็อคที่ข้อมูลมีจำนวนบิตไม่ครบตามที่กำหนดจำเป็นต้องถูกเพิ่มให้ครบก่อน กำหนดให้ n แทนจำนวนบิตของข้อมูลต้นฉบับ การเพิ่มบิตเติมเต็มสามารถทำได้โดยนำบิตที่มีค่าเป็น 1 จำนวน 1 บิตมาต่อท้ายตำแหน่งบิตสุดท้ายของ n แล้วนำบิตที่มีค่าเป็น “0” จำนวน r บิตมาต่อท้ายและปิดท้ายด้วย t ที่มีขนาด 64 บิตโดยค่าของ t แสดงขนาดของ n เพื่อให้ผู้รับทราบว่าขนาดที่แท้จริงของ n คือจำนวนกี่บิต การคำนวณหาจำนวนบิตของ r สามารถคำนวณได้จาก

$$r = 512 - (n + 1 + 64) \tag{10.1}$$

ตัวอย่างที่ 10.1 จงเพิ่มบิตเติมเต็มของข้อความต้นฉบับต่อไปนี้ ก่อนที่จะนำเข้าสู่ขั้นตอนวิธี SHA-1

$$m = \text{F4C78D10A41}$$

วิธีทำ เนื่องจาก m ที่โจทย์กำหนดเป็นเลขฐานสิบหก ซึ่งตัวอักษรหรือตัวเลข 1 ตัวจะมีขนาด 4 บิต ดังนั้นขนาดของ m ซึ่งมีสมาชิกจำนวน 11 ตัว จะมีขนาด $n = 4 \times 11 = 44$ บิต

จากสมการ (10.1) $r = 512 - (n + 1 + 64)$

$= 512 - (44 + 1 + 64) = 403$ บิต

และหากรวมกับบิต “1” ตัวที่จะนำมาต่อท้ายบิตตัวสุดท้ายของ *m* ก่อนจะเติมบิต “0” จำนวน 403 บิตจะได้ว่าบิตเติมเต็มทั้งหมดที่ยังไม่รวม 64 บิตสุดท้ายมีทั้งหมด 404 บิต ดังนี้

1000 0000 0000 0000 0000 0000 ⋯ 0000

403 ตัว

จากตารางที่ 1.3 สามารถเขียนบิตเติมเต็มที่ยังไม่รวม 64 บิตสุดท้ายในนรูปของเลขฐานสิบหกได้ดังนี้

80000000 00000000 00000000 ⋯ 00000000

100 ตัว

และเนื่องจาก *m* มีขนาด 44 บิต ดังนั้น ค่าของ *t* ซึ่งแทน 64 บิตสุดท้ายจึงมีค่าเป็น 44 ซึ่งแสดงในรูปของเลขฐานสองได้ดังนี้

t = 0000 0000 0000 0000 0000 ⋯ 0010 1100

58 ตัว

ซึ่งเขียน *t* ในรูปเลขฐานสิบหกได้ดังนี้

t = 00000000 0000002C

ดังนั้นข้อความขนาด 512 บิตซึ่งเกิดจาก *m* (44 บิต) + “1” + “000 ⋯ 0” (403) บิต + *t* (64 บิต) มีค่าเป็น

ตำแหน่ง

1	9	17	25	33	41	49	57
F4C78D10	A4180000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	000000C2
65	72	81	89	97	105	113	121

ขั้นตอนที่ 2: การขยายบิต

ขั้นตอนนี้จะนำข้อความต้นฉบับที่ผ่านการเพิ่มบิตเติมเต็มแล้วมาขยายจำนวนบิตเพิ่มเป็น 2560 บิต โดยเริ่มจากนำข้อความต้นฉบับมาแบ่งกลุ่มๆ ละ 32 บิต และเริ่มนับจากตำแหน่งที่ 1 ไปจนกระทั่งถึงตำแหน่งสุดท้ายซึ่งจะได้ทั้งหมด 16 กลุ่มดังนี้

กลุ่มที่ 1: $W(0)$ = ตำแหน่งที่ 1 – 8

กลุ่มที่ 2: $W(1)$ = ตำแหน่งที่ 9 - 16

กลุ่มที่ 3: $W(2)$ = ตำแหน่งที่ 17 – 24

กลุ่มที่ 4: $W(3)$ = ตำแหน่งที่ 25 - 32

กลุ่มที่ 5: $W(4)$ = ตำแหน่งที่ 33 – 40

กลุ่มที่ 6: $W(5)$ = ตำแหน่งที่ 41 - 48

กลุ่มที่ 7: $W(6)$ = ตำแหน่งที่ 49 – 56

กลุ่มที่ 8: $W(7)$ = ตำแหน่งที่ 57 - 64

กลุ่มที่ 9: $W(8)$ = ตำแหน่งที่ 65 – 72

กลุ่มที่ 10: $W(9)$ = ตำแหน่งที่ 73 - 80

กลุ่มที่ 11: $W(10)$ = ตำแหน่งที่ 81 – 88

กลุ่มที่ 12: $W(11)$ = ตำแหน่งที่ 89 - 96

กลุ่มที่ 13: $W(12)$ = ตำแหน่งที่ 97 – 104

กลุ่มที่ 14: $W(13)$ = ตำแหน่งที่ 105 - 112

กลุ่มที่ 15: $W(14)$ = ตำแหน่งที่ 113 – 120

กลุ่มที่ 16: $W(15)$ = ตำแหน่งที่ 121 - 128

โดยอีก 64 กลุ่มที่เหลือซึ่งเป็นส่วนของการขยายประกอบไปด้วย $W(17)$, $W(18)$, $W(19)$, … , $W(80)$ สามารถคำนวณได้ดังนี้

$$W(t) = \mathrm{RL}(W(t\text{-}3) \oplus W(t\text{-}8) \oplus W(t - 14) \oplus W(t\text{ - }16), 1) \qquad (10.2)$$

เมื่อ t คือจำนวนเต็มที่มีค่าอยู่ระหว่าง 16 ถึง 79

$\mathrm{RL}(x, y)$ คือการหมุนค่าเลขฐานสองของ x ไปทางซ้ายจำนวน y ครั้ง

ตัวอย่างที่ 10.2 จากข้อความต้นฉบับที่ผ่านการเพิ่มเติมบิตดังตัวอย่างที่ 10.2 จงหา $W(1) – W(16)$

วิธีทำ จากผลลัพธ์ที่ได้จากตัวอย่างที่ 10.1 ได้ว่า

$W(0)$ = F4C78D10

$W(1)$ = A4180000

$W(2)$ = 00000000

$W(3)$ = 00000000

$W(4)$ = 00000000

$W(5)$ = 00000000

$W(6)$ = 00000000

$W(7)$ = 00000000

$W(8)$ = 00000000

$W(9)$ = 00000000

$W(10)$ = 00000000

$W(11)$ = 00000000

$W(12)$ = 00000000

$W(13)$ = 00000000

$W(14)$ = 00000000

$W(15)$ = 000000C2

ตัวอย่างที่ 10.3 จากผลลัพธ์ดังตัวอย่างที่ 10.1 และ 10.2 จงแสดงวิธีหา $W(16)$

วิธีทำ จากสมการที่ (10.2) ได้ว่า

$$W(16) = \mathrm{RL}(W(13) \oplus W(8) \oplus W(2) \oplus W(0), 1)$$

เริ่มจากคำนวณ $x = W(13) \oplus W(8)$

$$= 00000000_{16} \oplus 00000000_{16}$$

$$= 00000000_{16}$$

ต่อมาคำนวณ $x = x \oplus W(2)$ (คำนวณหา $W(13) \oplus W(8) \oplus W(2)$)

$$x = x \oplus W(2)$$

$$= 00000000_{16} \oplus 00000000_{16}$$

$$= 00000000_{16}$$

และคำนวณ $x = x \oplus W(0)$ (คำนวณหา $W(13) \oplus W(8) \oplus W(2) \oplus W(0)$)

$$x = x \oplus W(0)$$
$$= 00000000_{16} \oplus F4C78D10_{16}$$
$$= F4C78D10_{16}$$

หรือ

$$W(16) = RL(F4C78D10_{16}, 1)$$

เนื่องจาก $F4C78D10_{16} = 11110100110001111000110100010000_2$

ดังนั้น

$$W(16) = RL(11110100110001111000110100010000_2, 1)$$
$$= 11101001100011110001101000100001_2$$
$$= E98F1A21_{16}$$

ขั้นตอนที่ 3: การคละบิต

การคละบิตคือการดำเนินการตามขั้นตอนวิธีที่ 10.1 เพื่อสลับค่าข้อมูลแต่ละค่าสำหรับในแต่รอบโดยจะดำเนินการทั้งหมด 80 รอบ โดยผลลัพธ์สุดท้ายที่ได้คือค่าแฮชขนาด 160 บิต

ขั้นตอนวิธีที่ 10.1 การคละบิต

```
     INPUT: W(0), W(1), …,W(15), H0, H1, H2, H3, H4, Kt, B, p
     OUTPUT: h
1:   A ← H0, B ← H1, C ← H2, D ← H3, E ← H4
2:   For (t = 0 to 79) do
3:       tmp ← (RL(A, 5) + ft(B, C, D) + E + W(t) + Kt) mod 100000000₁₆
4:       E ← D, D ← C, C ← RL(B, 30), B ← A, A ← tmp
5:   End For
6:   H0 ← (H0 + A) mod 100000000₁₆
7:   H1 ← (H1 + B) mod 100000000₁₆
8:   H2 ← (H2 + C) mod 100000000₁₆
9:   H3 ← (H3 + D) mod 100000000₁₆
10:  H4 ← (H4 + E) mod 100000000₁₆
11:  h ← concatenation of H0, H1, H2,H3, H4
```

จากขั้นตอนวิธีที่ 10.1 เนื่องจากตัวแปรแต่ละตัวจะเก็บค่าข้อมูลขนาด 32 บิต ดังนั้นหากผลลัพธ์ที่ได้จากการดำเนินการบวกมีค่าเกิน $ffffffff_{16}$ จำเป็นต้องนำผลลัพธ์ดังกล่าวไปผ่านกระบวนการมอดูโลด้วยค่า 100000000_{16} ซึ่งมีค่าเท่ากับ 4294967296

สำหรับค่า H_0, H_1, H_2, H_3 และ H_4 จะมีค่าเริ่มต้นเป็นดังต่อไปนี้

$H_0 = 67452301_{16}$

$H_1 = EFCDAB89_{16}$

$H_2 = 98BADCFE_{16}$

$H_3 = 10325476_{16}$

$H_4 = C3D2E1F0_{16}$

ในส่วนของ K_t จะถูกแบ่งออกเป็นทั้งหมด 4 ค่า โดยการเลือกใช้งานในแต่ละค่าจะขึ้นอยู่กับรอบของการคำนวณดังนี้

1. เลือกใช้ $K_t = 5A827999_{16}$ กรณีที่ t มีค่าอยู่ระหว่าง 0 ถึง 19
2. เลือกใช้ $K_t = 6ED9EBA1_{16}$ กรณีที่ t มีค่าอยู่ระหว่าง 20 ถึง 39
3. เลือกใช้ $K_t = 8F1BBCDC_{16}$ กรณีที่ t มีค่าอยู่ระหว่าง 40 ถึง 59
4. เลือกใช้ $K_t = CA62C1D6_{16}$ กรณีที่ t มีค่าอยู่ระหว่าง 60 ถึง 79

และค่าสุดท้ายคือ $f_t(B, C, D)$ ซึ่งเป็นฟังก์ชันที่มีสมการขึ้นอยู่กับรอบของการคำนวณ สำหรับการเลือกใช้งานแต่ละสมการเป็นดังนี้

1. $f_t(B, C, D) = (B \wedge C) \vee ((\text{not } B) \wedge D)$ กรณีที่ t มีค่าอยู่ระหว่าง 0 ถึง 19
2. $f_t(B, C, D) = B \oplus C \oplus D$ กรณีที่ t มีค่าอยู่ระหว่าง 20 ถึง 39
3. $f_t(B, C, D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$ กรณีที่ t มีค่าอยู่ระหว่าง 40 ถึง 59
4. $f_t(B, C, D) = B \oplus C \oplus D$ กรณีที่ t มีค่าอยู่ระหว่าง 60 ถึง 79

เมื่อสัญลักษณ์ “$\wedge$” คือตัวดำเนินการแอนด์ (and) และ สัญลักษณ์ “$\vee$” คือตัวดำเนินการออร์ (or) ซึ่งผลลัพธ์ที่เกิดจากอินพุตขนาด 1 บิตจำนวน 2 ค่าเชื่อมกันด้วยเครื่องหมาย “$\wedge$” และเครื่องหมาย “$\vee$” และผลลัพธ์แสดงดังตารางที่ 10.1 และตารางที่ 10.2 ตามลำดับ

ตารางที่ 10.1 แสดงผลลัพธ์การดำเนินการแบบบิตระหว่าง 2 อินพุตผ่านตัวดำเนินการแอนด์ แบ่งออกเป็น 4 กรณีดังนี้

กรณีที่ 1: $A = 0$, $B = 0$ ได้ผลลัพธ์ $Z = 0$

กรณีที่ 2: $A = 0$, $B = 1$ ได้ผลลัพธ์ $Z = 0$

กรณีที่ 3: $A = 1$, $B = 0$ ได้ผลลัพธ์ $Z = 0$

กรณีที่ 4: $A = 1$, $B = 1$ ได้ผลลัพธ์ $Z = 1$

ตารางที่ 10.1 ผลการดำเนินการแบบบิตผ่านตัวดำเนินการแอนด์

อินพุต		**เอาต์พุต**
A	B	$Z = A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

ตารางที่ 10.2 ผลการดำเนินการแบบบิตผ่านตัวดำเนินการออร์

อินพุต		**เอาต์พุต**
A	B	$Z = A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

ตารางที่ 10.2 แสดงผลลัพธ์การดำเนินการแบบบิตระหว่าง 2 อินพุตผ่านตัวดำเนินการออร์ แบ่งออกเป็น 4 กรณีดังนี้

กรณีที่ 1: $A = 0, B = 0$ ได้ผลลัพธ์ $Z = 0$

กรณีที่ 2: $A = 0, B = 1$ ได้ผลลัพธ์ $Z = 1$

กรณีที่ 3: $A = 1, B = 0$ ได้ผลลัพธ์ $Z = 1$

กรณีที่ 4: $A = 1, B = 1$ ได้ผลลัพธ์ $Z = 1$

สำหรับวิธีการคำนวณการดำเนินการแอนด์ และการดำเนินการออร์จะมีลักษณะที่คล้ายกับการดำเนินการเอ็กคลูซีฟออร์ ซึ่งจะดำเนินการระหว่างบิตโดยใช้ตำแหน่งบิตที่มีนัยสำคัญต่ำที่สุดคือบิตที่ 0 และเรียงไปจนกระทั่งถึงบิตที่มีนัยสำคัญสูงที่สุดคือตำแหน่งสุดท้าย

ตัวอย่างที่ 10.4 กำหนดให้ $x = BC_{16}$ และ $y = 75_{16}$ จงหา $z = x \vee y$

วิธีทำ เนื่องจากเป็นการดำเนินการระหว่างบิตจึงจำเป็นต้องแปลง x และ y ให้อยู่ในรูปแบบของเลขฐานสองดังนี้

ตำแหน่ง	7	6	5	4	3	2	1	0
x	1	0	1	1	1	1	0	0
y	0	1	1	1	0	1	0	1
z	1	1	1	1	1	1	0	1

ดังนั้น $z = 11111101_2 = FD_{16}$

สำหรับตัวดำเนินการสุดท้ายคือนิเสธ “not” คือการดำเนินการกลับบิตของเลขฐานสองของแต่ละบิตให้อยู่ในสถานะตรงกันข้ามกับสถานะตั้งต้น

ตัวอย่างที่ 10.5 จากผลลัพธ์ดังตัวอย่างที่ 10.1 ถึง 10.3 จงหา A, B, C, D และ E ที่ได้หลังจากผ่านการคำนวณจากขั้นตอนวิธีการคละบิตรอบแรก

วิธีทำ จากขั้นตอนวิธีการคละบิต ได้ขั้นตอนการดำเนินงานเป็นดังนี้

1. A = 67452301

   B = EFCDAB89

   C = 98BADCFE

   D = 10325476

   E = C3D2E1F0

ขั้นตอนที่ 2 – 5 คือการวนรอบการดำเนินการทั้งหมด 80 รอบ ดังนี้

รอบที่ 1: (t = 0)

3. คำนวนค่า *tmp* ดังนี้

3.1 คำนวณหา RL(A, 5)

เนื่องจาก

A = 67452301 = 0110 0111 0100 0101 0010 0011 0000 0001

ดังนั้น RL(A, 5) = RL($01100111010001010010001100000001_2$, 5)

= 1110 1000 1010 0100 0110 0000 0010 1100

= E8A4602C

3.2 คำนวนหา $f_t(B, C, D)$

เนื่องจากเป็นการคำนวณรอบแรกจึงเลือกใช้ $f_t(B, C, D)$ ดังนี้

$f_t(B, C, D) = (B \wedge C) \vee ((\text{not } B) \wedge D)$

เนื่องจาก $B \wedge C$ = EFCDAB89 $\wedge$ 98BADCFE

= 88888888

not B = not (EFCDAB89)

= not (1110 1111 1100 1101 1010 1011 1000 1001)

= 0001 0000 0011 0010 0101 0100 0111 0110

= 10325476

(not B) $\wedge$ D = 10325476 $\wedge$10325476

= 10325476

ดังนั้น $f_t(B, C, D)$ = 88888888 $\vee$ 10325476

= 98BADCFE

3.3 เลือก k_t เนื่องจากเป็นการคำนวณรอบแรกจึงเลือกใช้ k_t = 5A827999 จาก

$tmp = (\text{RL}(A, 5) + f_t(B, C, D) + E + W(t) + K_t) \bmod 100000000$

= E8A4602C + 98BADCFE + C3D2E1F0 + F4C78D10 + 5A827999 mod 100000000

= 947C25C3

4. หาค่า A, B, C, D และ E ดังนี้

4.1 $E = D$ = 10325476

4.2 $D = C$ = 98BADCFE

4.3 C = RL(B, 30)

= RL(EFCDAB89, 30)

= RL(1110 1111 1100 1101 1010 1011 1000 1001, 30)

= 0111 1011 1111 0011 0110 1010 1110 0010

= 7BF36AE2

4.4 $B = A$ = 67452301

4.5 $A = tmp$ = 947C25C3

เนื่องจากโจทย์ถามหา หา *A*, *B*, *C*, *D* และ *E* ที่ได้หลังจากผ่านการคำนวณจากขั้นตอนวิธีการคละบิตในรอบแรกดังนั้นจึงได้ว่า *A* = 947C25C3, *B* = 67452301, *C* = 7BF36AE2, *D* = 98BADCFE และ *E* = 10325476

ตัวอย่างที่ 10.6 กำหนดให้หลังจากเสร็จสิ้นการคำนวณรอบสุดท้ายของการคละบิตได้ *A* = 11111111, *B* = 22222222, *C* = 33333333, *D* = 44444444 และ *E* = 55555555 จงคำนวณหาค่าแฮช

วิธีทำ จากขั้นตอนวิธีการคละบิตหลังจากเสร็จสิ้นการคำนวณรอบสุดท้าย จะต้องคำนวณหา H_0, H_1, H_2, H_3 และ H_4 ดังนี้

$H_0 = (H_0 + A) \bmod 100000000$

$= (67452301 + 11111111) \bmod 100000000$

$= 78563412$

$H_1 = (H_1 + B) \bmod 100000000$

$= (EFCDAB89 + 22222222) \bmod 100000000$

$= 11EFCDAB$

$H_2 = (H_2 + C) \bmod 100000000$

$= (98BADCFE + 33333333) \bmod 100000000$

$= CBEE1031$

$H_3 = (H_3 + D) \bmod 100000000$

$= (10325476 + 44444444) \bmod 100000000$

$= 547698BA$

$H_4 = (H_4 + E) \bmod 100000000$

$= (C3D2E1F0 + 55555555) \bmod 100000000$

$= 19283745$

ดังนั้นค่าแฮชคือ *h* = 78563412 11EFCDAB CBEE1031 547698BA 19283745

2. ตัวอย่างการนำฟังก์ชันแฮชไปประยุกต์ใช้งาน

ฟังก์ชันแฮชสามารถถูกนำไปประยุกต์ใช้งานได้ในหลายๆ ด้านเช่น ลายเซ็นดิจิทัล และการเก็บค่ารหัสผ่าน สำหรับการนำฟังก์ชันแฮชไปประยุกต์ใช้กับลายเซ็นดิจิทัลจะกล่าวอีกครั้งอย่าง

ละเอียดในหัวข้อที่ 5 สำหรับในหัวข้อนี้จะอธิบายตัวอย่างของการนำฟังก์ชันแฮชไปประยุกต์ใช้สำหรับเก็บรหัสผ่าน

โดยทั่วไปโปรแกรมที่การใช้งานจำเป็นต้องมีการผ่านระบบพิสูจน์ตัวตน ดังนั้นผู้ใช้งานจำเป็นต้องกรอกชื่อผู้ใช้งาน และรหัสผ่านก่อนเข้าใช้งานซึ่งข้อมูลเหล่านี้จะถูกเก็บไว้ในฐานข้อมูลของระบบ ดังนั้นจึงมีความเป็นไปได้ที่รหัสผ่านจะสามารถเข้าถึงได้โดยผู้ดูแลระบบ แต่เนื่องจากรหัสผ่านเป็นข้อมูลที่เป็นความลับที่ผู้ใช้งานไม่ต้องการให้บุคคลอื่นรับรู้ ดังนั้นจึงสามารถแก้ปัญหาดังกล่าวได้โดยการเก็บแฮชของรหัสผ่านไว้ที่ฐานข้อมูลแทน ซึ่งการใช้งานเมื่อผู้ใช้งานกรอกรหัสผ่านระบบจะนำรหัสผ่านไปผ่านฟังก์ชันแฮชและนำผลลัพธ์ที่ได้ไปตรวจสอบกับค่าที่เก็บไว้ที่ฐานข้อมูลซึ่งหากข้อมูลตรงกันจึงจะให้สิทธิการเข้าใช้งานระบบแก่ผู้ใช้งาน จากหลักการดังกล่าวนี้ผู้ดูแลระบบจะไม่สามารถทราบค่ารหัสผ่านที่แท้จริงของผู้ใช้งานได้เนื่องจากค่าที่เก็บไว้ในฐานข้อมูลคือค่าแฮชนั่นเอง

3. ความปลอดภัยของฟังก์ชันแฮช

ค่าแฮชที่ได้จาก SHA-1 มีขนาด 160 บิต จึงมีค่าที่เป็นไปได้ทั้งหมด 2^{160} ค่า เนื่องจากข้อความต้นฉบับซึ่งมีขนาด 512 บิต มีค่าที่เป็นไปได้ทั้งหมด 2^{512} ค่า ดังนั้นจึงมีความเป็นไปได้ที่ข้อความต้นฉบับที่แตกต่างกันจะได้ค่าแฮชที่มีค่าเท่ากัน โดยที่หากสร้างข้อความที่แตกต่างกันจำนวน $2^{160} + 1$ ข้อความจะเกิดค่าแฮชที่มีค่าเท่ากันอย่างน้อย 1 คู่อย่างแน่นอน

อย่างไรก็ตามไม่มีความจำเป็นที่ต้องสร้างข้อความจำนวนมากถึง $2^{160} + 1$ ข้อความก็ยังคงมีโอกาสที่จะพบข้อความที่มีค่าแฮชที่ซ้ำกันด้วยความน่าจะเป็นที่สูงซึ่งสามารถอธิบายได้โดยใช้วันเกิดผิดปกติ

4. วันเกิดผิดปกติ (Birthday paradox)

วันเกิดผิดปกติ [61] คือวิธีการที่ใช้สำหรับแก้ปัญหาโดยอาศัยหลักการชนกันของข้อมูลซึ่งมีแนวคิดดังนี้ใน 1 ปีมี 365 วัน ดังนั้นหากพิจารณาประชากรทั้งหมด 366 คน จะต้องมีประชากรที่เกิดในวันเดียวกันอย่างน้อย 1 คู่อย่างแน่นอน แต่ในความเป็นจริงโอกาสของวันเกิดของประชากรที่เกิดในวันเดียวกันมีความเป็นไปได้สูงโดยไม่จำเป็นต้องพิจารณาประชากรมากถึง 366 คน ดังนี้

พิจารณาประชากร 1 คน ความน่าจะเป็นที่จะไม่เกิดวันเดียวกันมีค่าเป็น $\frac{365}{365} = 1$

พิจารณาประชากร 2 คน ความน่าจะเป็นที่จะไม่เกิดวันเดียวกันมีค่าเป็น $\frac{365}{365} \times \frac{364}{365}$

พิจารณาประชากร 3 คน ความน่าจะเป็นที่จะไม่เกิดวันเดียวกันมีค่าเป็น $\frac{365}{365} \times \frac{364}{365} \times \frac{363}{365}$

ดังนั้นหากพิจารณาประชากร d คน ความน่าจะเป็นที่จะไม่เกิดวันเดียวกันมีค่าเป็น

$$p = \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{365-d+1}{365}$$

$$p = \frac{365!}{(365-d)! \times 365^d} \quad (10.3)$$

จากสมการที่ (10.3) คือความน่าจะเป็นของประชากรจำนวน d คนที่ไม่เกิดวันเดียวกัน ดังนั้นการหาความน่าจะเป็นของประชากรจำนวน d คนที่เกิดวันเดียวกันคือ

$$p' = 1 - \frac{365!}{(365-d)! \times 365^d} \quad (10.4)$$

จากสมการที่ (10.4) ทดลองแทน $d = 23$ จะได้ผลลัพธ์ของ p' มีค่าเป็น 0.507 ความหมายคือความเป็นไปได้ที่ประชากรจำนวน 23 คนจะเกิดวันเดียวกันมีค่าสูงถึง 0.507 โดยที่หากประชากรยิ่งสูงขึ้นจะส่งผลค่า p' มีค่าที่สูงมากยิ่งขึ้นตามไปด้วย

จากปัญหาวันเกิดผิดปกติ 365 คือค่าข้อมูลที่เป็นไปได้ทั้งหมด ดังนั้นหากนำไปประยุกต์ใช้งานเพื่อวิเคราะห์หาความน่าจะเป็นการชนกันของค่าแฮชสามารถทำได้โดยการปลี่ยนสมการเป็นดังนี้

$$p' = 1 - \frac{n!}{(n-d)! \times n^d} \quad (10.5)$$

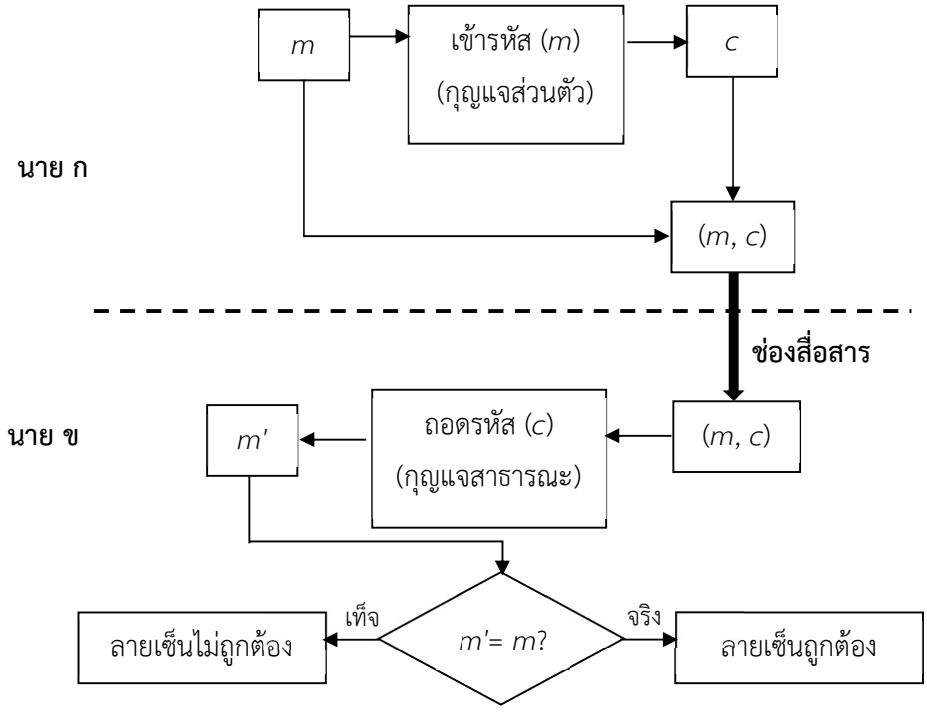
เมื่อ n คือค่าแฮชที่เป็นไปได้ทั้งหมด

5. ลายเซ็นดิจิทัล (Digital Signature)

สมมติอาจารย์ที่ปรึกษาท่านหนึ่งต้องการส่งคะแนนผลการเรียนของนักศึกษาให้ผู้ปกครองทราบ ปัญหาคือผู้ปกครองจะเชื่อถือได้อย่างไรว่าผลการเรียนนั้นถูกส่งมาจากอาจารย์ที่ปรึกษาบุตรของตนเอง ซึ่งการเซ็นชื่อของอาจารย์ที่ปรึกษากำกับไว้ที่เอกสารฉบับดังกล่าวคือการยืนยันให้ผู้ปกครองของนักศึกษามั่นใจได้ว่าเอกสารฉบับนี้ถูกส่งมาจากอาจารย์ที่ปรึกษาจริง

ลายเซ็นดิจิทัล คือวิธีการที่ใช้สำหรับยืนยันเอกสารอิเล็กทรอนิกส์เพื่อเป็นการยืนยันให้ผู้รับมั่นใจว่าเอกสารดังกล่าวถูกส่งมาจากผู้ส่งจริง จากความหมายของลายเซ็นดิจิทัลพบว่ามีหลักการคล้ายคลึงกับการเซ็นลายเซ็นกำกับที่เอกสารแบบปกติ ความแตกต่างคือลายเซ็นดิจิทัลจะถูกนำมาใช้กับเอกสารอิเล็กทรอนิกส์ โดยวิทยาการรหัสลับแบบกุญแจสาธารณะคือเครื่องมือหลักที่สำคัญ จาก

แนวคิดของวิทยาการรหัสลับแบบกุญแจสาธารณะที่นำมาใช้สำหรับการรักษาความปลอดภัยข้อมูลคือ หากนำกุญแจสาธารณะเพื่อเข้ารหัสข้อมูลจะต้องใช้กุญแจส่วนตัวที่มีความสัมพันธ์กับกุญแจสาธารณะสำหรับถอดรหัส ในทางกลับกันหากนำวิทยาการรหัสลับมาประยุกต์ใช้งานเป็นลายเซ็นดิจิทัลจะใช้กุญแจส่วนตัวเพื่อเข้ารหัสข้อมูล (ลายเซ็น) ของผู้ส่งและผู้รับสามารถตรวจสอบลายเซ็นได้โดยใช้กุญแจสาธารณะ

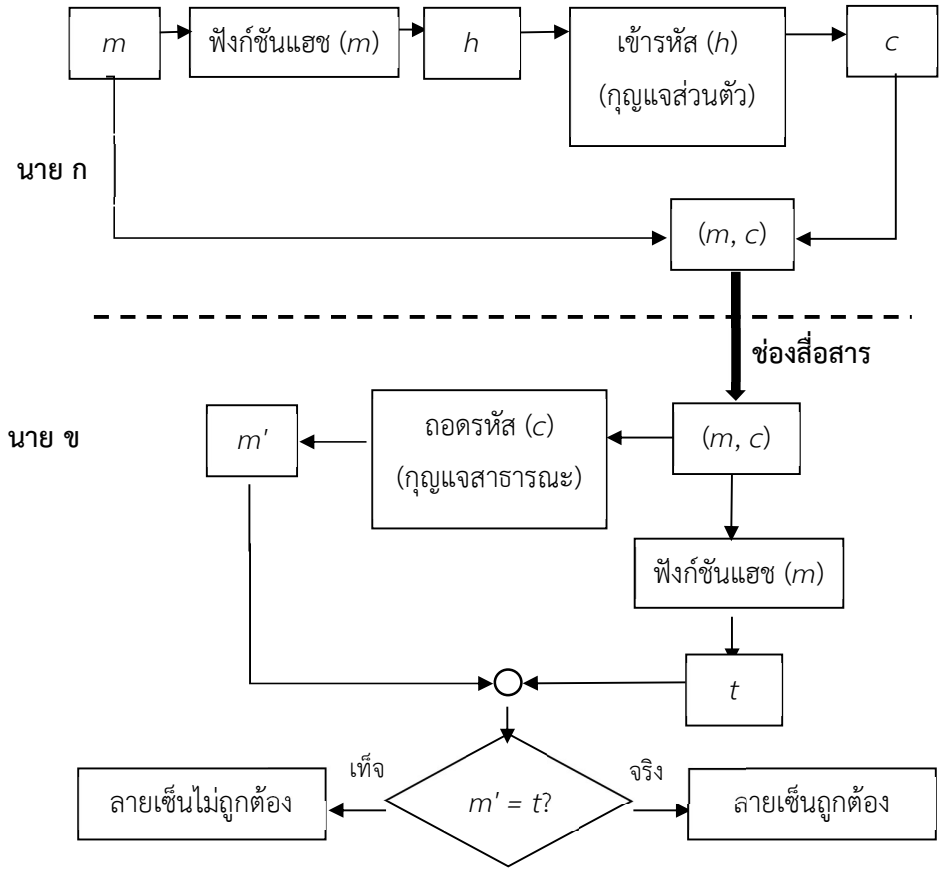


รูปที่ 10.1 ตัวอย่างลายเซ็นดิจิทัล

สมมตินาย ก ต้องการส่งเอกสารอิเล็กทรอนิกส์ไปยังนาย ข เพื่อเป็นการยืนยันให้นาย ข มั่นใจว่าเอกสารดังกล่าวนี้ถูกส่งมาจากนาย ก จริง นาย ก และนาย ข จึงตกลงใช้ลายเซ็นดิจิทัลดังรูปที่ 10.1 ซึ่งมีหลักการดังนี้ นาย ก ทำการเข้ารหัสลายเซ็นของตนเองโดยใช้กุญแจส่วนตัวซึ่งจากรูปกำหนดให้ m คือลายเซ็นของนาย ก และ c คือข้อความไซเฟอร์ที่เกิดจากการเข้ารหัส m ด้วยกุญแจส่วนตัว หลังจากเสร็จสิ้นกระบวนการเข้ารหัสนาย ก จะส่งทั้ง m และ c ไปยังนาย ข โดยหลังจากได้รับ m และ c นาย ข จะนำ c มาถอดรหัสโดยใช้กุญแจสาธารณะของนาย ก กำหนดให้ข้อความที่ได้จากการถอดรหัสคือ m' และขั้นตอนสุดท้ายนาย ข จะนำ m' และ m มาทำการเปรียบเทียบหากมีค่าเท่ากันแสดงว่าลายเซ็นดังกล่าวเป็นของนาย ก จริง ในทางกลับกันหาก m' และ m มีผลลัพธ์ที่ไม่เท่ากันสรุปได้ว่าลายเซ็นไม่ถูกต้อง

จากตัวอย่างข้างต้นที่ผู้เขียนกำหนดให้ m' เป็นข้อความที่ได้จากการถอดรหัสด้วยกุญแจสาธารณะเนื่องจากหาก c ไม่ได้เกิดจากการเข้ารหัส m ด้วยกุญแจส่วนตัวจะส่งผลให้ m' ที่ได้ออกมามีค่าไม่ตรงกับ m

เนื่องจากนาย ก คือบุคคลที่ถือค่ากุญแจส่วนตัวแต่เพียงผู้เดียว ดังนั้นจึงมีเพียงนาย ก เท่านั้นที่จะสามารถเข้ารหัสลายเซ็นด้วยกุญแจส่วนตัวได้ จึงสามารถนำวิทยาการรหัสลับแบบกุญแจสาธารณะมาประยุกต์ใช้เพื่อสร้างลายเซ็นดิจิทัลได้



รูปที่ 10.2 ตัวอย่างลายเซ็นดิจิทัลที่ใช้ค่าแฮชของลายเซ็น

อย่างไรก็ตามในการใช้งานจริงขนาดของลายเซ็นจะมีค่าที่ไม่แน่นอนซึ่งขึ้นอยู่กับชนิดของวิทยาการรหัสลับแบบกุญแจสาธารณะที่เลือกมาใช้งาน ยกตัวอย่างเช่นหากเลือกวิทยาการรหัสลับอาร์เอสเอซึ่งขนาดของค่าโมดูลัสไม่ควรน้อยกว่า 1024 บิต ดังนั้นค่าลายเซ็นที่สามารถเลือกใช้งานได้จะมีขนาดที่อยู่ระหว่าง 1 – 1024 บิต ซึ่งเป็นไปได้ที่ลายเซ็นที่เลือกใช้งานจะมีขนาดใหญ่มหาศาล ดังนั้นวิธีที่สะดวกกว่าคือนำค่าแฮชของลายเซ็นมาใช้สำหรับกระบวนการเข้ารหัสและถอดรหัสเพื่อใช้

สำหรับการตรวจสอบลายเซ็นแทน อย่างไรก็ตามค่าแฮชจะมีขนาดที่แน่นอนและมีขนาดที่ไม่ใหญ่หากเปรียบเทียบกับค่าที่เป็นไปได้ทั้งหมดของลายเซ็น เช่น สมมติเลือกขั้นตอนวิธี SHA-1 ค่าแฮชจะมีขนาด 160 เสมอ รูปที่ 10.2 แสดงตัวอย่างลายเซ็นดิจิทัลที่ใช้ค่าแฮชของลายเซ็นซึ่งมีหลักการเป็นดังนี้

นาย ก นำลายเซ็น (m) มาผ่านฟังก์ชันแฮช และทำการเข้ารหัสลายเซ็นโดยใช้กุญแจส่วนตัว ซึ่งจากรูป h คือค่าแฮชลายเซ็นของนาย ก และ c คือข้อความไซเฟอร์ที่เกิดจากการเข้ารหัส h ด้วยกุญแจส่วนตัว หลังจากเสร็จสิ้นกระบวนการเข้ารหัสนาย ก จะส่งทั้ง m และ c ไปยังนาย ข โดยหลังจากได้รับ m และ c นาย ข จะนำ c มาถอดรหัสโดยใช้กุญแจสาธารณะของนาย ก กำหนดให้ข้อความที่ได้จากการถอดรหัสคือ m' และนำ m มาผ่านฟังก์ชันแฮชกำหนดให้ค่าแฮชที่ฝั่งนาย ข คือ t ขั้นตอนสุดท้ายนาย ข จะนำ m' และ t มาทำการเปรียบเทียบหากมีค่าเท่ากันแสดงว่าลายเซ็นดังกล่าวเป็นของนาย ก จริง ในทางกลับกันหาก m' และ t มีผลลัพธ์ที่ไม่เท่ากันสรุปได้ว่าลายเซ็นไม่ถูกต้อง

ในบทนี้จะกล่าวถึงวิทยาการรหัสลับแบบกุญแจสาธารณะ 3 ประเภทประกอบไปด้วย วิทยาการรหัสลับเอ็ลกามอล วิทยาการรหัสลับอาร์เอสเอ และวิทยาการรหัสลับเส้นโค้งเชิงวงรีเพื่อนำไปสร้างเป็นลายเซ็นดิจิทัล

6. ลายเซ็นดิจิทัลเอ็ลกามอล (ElGamal Digital Signature)

ลายเซ็นดิจิทัลเอ็ลกามอล จะมีรูปแบบของสมการที่ใช้สำหรับการเซ็นลายเซ็น (การเข้ารหัส) และการตรวจสอบลายเซ็น (การถอดรหัส) แตกต่างจากวิทยาการรหัสลับเอ็ลกามอลที่ใช้สำหรับการรักษาความปลอดภัยข้อมูลข่าวสาร โดยแบ่งออกเป็น 3 กระบวนการดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจ หรือผู้เซ็นลายเซ็น (กำหนดเป็น ผู้ส่ง) มีลำดับการทำงานเป็นดังนี้

9. เลือกจำนวนเฉพาะ p และรากปฐมฐาน g มอดูโล p
10. เลือก $a \in \{0, 1, 2, \cdots, p-2\}$
11. คำนวณ A จาก $A = g^a \bmod p$

กุญแจสาธารณะคือ $\{p, g, A\}$

กุญแจส่วนตัวคือ $\{a\}$

กระบวนการที่ 2 การเซ็นลายเซ็นดิจิทัล (เข้ารหัสลับ): เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจผู้ซึ่งต้องการยืนยันลายเซ็นของตนเอง (สมมติลายเซ็นคือ m และค่าแฮชของ m คือ h เมื่อ $1 < m, h < p - 1$) โดยใช้กุญแจส่วนตัว มีลำดับการทำงานเป็นดังนี้

1. เลือก $b \in \{0, 1, 2, \cdots, p-2\}$ โดยที่ $\gcd(b, p-1) = 1$
2. คำนวณ c_1 จาก

$$c_1 = g^b \bmod p$$

3. คำนวณ c_2 จาก

$$c_2 = (h - ac_1)b^{-1} \bmod p - 1$$

หลังเสร็จสิ้นกระบวนการเซ็นลายเซ็นแล้วได้ลายเซ็นดิจิทัลที่ได้คือ $\{c_1, c_2\}$ โดยผู้ส่งจะส่ง $\{m, c_1, c_2\}$ ไปยังผู้รับ

กระบวนการที่ 3 การตรวจสอบลายเซ็นดิจิทัล (การถอดรหัสลับ): หลังจากที่ผู้รับได้รับ $\{m, c_1, c_2\}$ จากผู้ส่งจะสามารถตรวจสอบลายเซ็นได้โดยสมการต่อไปนี้

1. คำนวณ $r = A^{c_1}c_1^{c_2} \bmod p$
2. คำนวณค่าแฮชของ m กำหนดเป็น h'
3. คำนวณ $s = g^{h'} \bmod p$
4. เปรียบเทียบผลลัพธ์ระหว่าง r และ s หากผลลัพธ์มีค่าเท่ากันแสดงว่าการตรวจสอบลายเซ็นผ่านการอนุมัติ ในทางกลับกันหากผลลัพธ์ไม่เท่ากันจะไม่ผ่านขั้นตอนการตรวจสอบลายเซ็น

จากทั้ง 3 กระบวนการข้างต้นหากไม่มีการปลอมแปลงเอกสารเกิดขึ้น เมื่อเสร็จสิ้นกระบวนตรวจสอบลายเซ็นแล้วพบว่า r มีค่าเท่ากับ s เสมอ ดังนี้

$$\begin{aligned} r &= A^{c_1}c_1^{c_2} \bmod p \\ &= (g^a)^{c_1}(g^b)^{(h-ac_1)b^{-1} \bmod p-1} \bmod p \end{aligned}$$

เนื่องจากในกระบวนการที่ 2 ได้กำหนดให้ $\gcd(b, p-1) = 1$ ได้ว่า $bb^{-1} \bmod p - 1 = 1$ ดังนั้น

$$\begin{aligned} r &= (g^a)^{c_1} g^{(h-ac_1) \bmod p-1} \bmod p \\ &= g^{ac_1} g^{(h-ac_1) \bmod p-1} \bmod p \\ &= g^{ac_1} g^{-ac_1} g^{h \bmod p-1} \bmod p \end{aligned}$$

$$= g^{ac_1} g^{-ac_1} g^{h \bmod p-1} \bmod p$$

$$= g^{h \bmod p-1} \bmod p$$

เนื่องจาก $h < p - 1$ ดังนั้น

$$r = g^h \bmod p$$

หาก m ไม่ถูกเปลี่ยนแปลง h จะมีค่าเท่ากับ h' เสมอ ดังนั้น

$$r = s$$

ตัวอย่างที่ 10.7 การประยุกต์ใช้ลายเซ็นดิจิทัลเอ็ลกามอล

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

9. เลือกจำนวนเฉพาะ $p = 37$ และรากปฐมฐาน $g = 17$
10. เลือก $a = 7$
11. คำนวณ $A = 17^7 \bmod 37 = 15$

กุญแจสาธารณะคือ $\{p = 37, g = 13, A = 15\}$

กุญแจส่วนตัวคือ $\{a = 7\}$

กระบวนการเซ็นลายเซ็นดิจิทัล

สมมติผู้ส่งต้องการส่งลายเซ็นที่มีค่าแฮชคือ 15 (ค่าแฮชที่กำหนดให้คือค่าสมมติ)

4. เลือก $b = 13$ เนื่องจาก gcd(13, 36) = 1
5. คำนวณ c_1 โดยใช้สมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้ $c_1 = 17^{13} \bmod 37 = 35$
6. คำนวณ $c_2 = (15 - 7 \times 35) \times 13^{-1} \bmod 36$

$$= (-230) \times 13^{-1} \bmod 36$$

$$= 22 \times 13^{-1} \bmod 36$$

$$= 22 \times 25 \bmod 36$$

$$= 10$$

ส่ง $\{m, c_1 = 35, c_2 = 10\}$ ไปยังผู้รับ

กระบวนการตรวจสอบลายเซ็นดิจิทัล

เมื่อรับข้อมูล {m, c_1 = 35, c_2 = 10 } จากผู้ส่ง ผู้รับต้องคำนวณหา r และ s ดังนี้

1. คำนวณ r โดยใช้สมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ
   $r = 15^{35} \times 35^{10} \bmod 37 = 14$
2. คำนวณค่าแฮชของ m ได้เป็น 15
3. คำนวณ $s = 17^{15} \bmod 37 = 14$

เนื่องจาก $r = s$ ผู้รับจึงมั่นใจได้ว่าลายเซ็นถูกส่งมาจากผู้ส่งจริง

เนื่องจากความปลอดภัยของลายเซ็นดิจิทัลเอ็ลกามอลขึ้นอยู่กับปัญหาวิยุตลอการิทึม ดังนั้น ขนาดของ p ควรมีค่าไม่น้อยกว่า 1024 บิตเพื่อหลีกเลี่ยงการโจมตีด้วยขั้นตอนวิธีดรรชนีแคลคูลัส นอกเหนือจากนั้นจำนวนประกอบทั้งหมดของ $p - 1$ ควรมีขนาดใหญ่เพื่อหลีกเลี่ยงการโจมตีด้วยขั้นตอนวิธีโพลิกเฮลแมน

7. ลายเซ็นดิจิทัลอาร์เอสเอ (RSA Digital Signature)

ลายเซ็นดิจิทัลอาร์เอสเอ เป็นวิธีการที่มีขั้นตอนคล้ายคลึงกับวิทยาการรหัสลับอาร์เอสเอ ข้อแตกต่างมีเพียงแค่ขั้นตอนการเซ็นลายเซ็นดิจิทัลจะอยู่ที่ฝั่งผู้ก่อกำเนิดกุญแจซึ่งใช้กุญแจส่วนตัวสำหรับกระบวนการเข้ารหัส ในทางกลับการผู้ตรวจสอบลายเซ็นจะใช้กุญแจสาธารณะสำหรับการตรวจสอบลายเซ็น สำหรับกระบวนการก่อกำเนิดกุญแจยังคงเป็นเช่นเดิม แต่ขั้นตอนการเซ็นลายดิจิทัล และกระบวนการตรวจสอบลายเซ็นดิจิทัล เป็นดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: ขั้นตอนการดำเนินการเหมือนกับวิทยาการรหัสลับอาร์เอสเอ

กระบวนการที่ 2 การเซ็นลายเซ็นดิจิทัล: กำหนดให้ h แทนค่าแฮชของลายเซ็น m ซึ่ง $1 < m, h < n$ ผู้ส่งเซ็นลายเซ็นดิจิทัล ดังนี้

$$c = h^d \bmod n$$

หลังเสร็จสิ้นกระบวนการเซ็นลายเซ็นแล้วได้ลายเซ็นดิจิทัลที่ได้คือ {c} โดยผู้ส่งจะส่ง {m, c} ไปยังผู้รับ

กระบวนการที่ 3 การถอดรหัสลับ: หลังจากที่ผู้รับได้รับ $\{m, c\}$ จากผู้ส่งจะสามารถคำนวณหา s ได้โดยสมการต่อไปนี้

$$s = c^e \bmod n$$

และคำนวณหาค่าแฮชของ m ซึ่งกำหนดให้เป็น h' และเปรียบผลลัพธ์ระหว่าง h' และ s หากผลลัพธ์มีค่าเท่ากันแสดงว่าการตรวจสอบลายเซ็นผ่านการอนุมัติ ในทางกลับกันหากผลลัพธ์ไม่เท่ากันจะไม่ผ่านขั้นตอนการตรวจสอบลายเซ็น

จากทั้ง 3 กระบวนการข้างต้นหากไม่มีการปลอมแปลงเอกสารเกิดขึ้น เมื่อเสร็จสิ้นกระบวนตรวจสอบลายเซ็นแล้วพบว่า h' มีค่าเท่ากับ s เสมอ

ตัวอย่างที่ 10.8 การประยุกต์ใช้ลายเซ็นดิจิทัลอาร์เอสเอ

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

เลือกใช้กุญแจดังตัวอย่างที่ 7.1 ดังนี้

กุญแจสาธารณะคือ $\{e = 919, n = 5293\}$

กุญแจส่วนตัวคือ $\{d = 1927\}$

กระบวนการเซ็นลายเซ็นดิจิทัล

สมมติผู้ส่งต้องการส่งลายเซ็นที่มีค่าแฮชคือ 23 (ค่าแฮชที่กำหนดให้คือค่าสมมติ) จึงสามารถคำนวณหาค่าลายเซ็นโดยใช้สมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ได้ดังนี้

$$c = 23^{1927} \bmod 5293$$
$$= 2946$$

ส่ง $\{m, c = 2946\}$ ไปยังผู้รับ

กระบวนการตรวจสอบลายเซ็นดิจิทัล

เมื่อรับข้อมูล $\{m, c = 2946\}$ จากผู้ส่ง ผู้รับต้องคำนวณหา s และค่าแฮชของ m โดยใช้สมการการยกกำลังมอดูลาร์ด้วยวิธีเลขยกกำลังแบบเร็วหรือวิธีอื่นๆ ดังนี้

$$s = 2946^{919} \bmod 5293$$
$$= 23$$

และคำนวณ $h' = 23$

เนื่องจาก $h' = s$ ผู้รับจึงมั่นใจได้ว่าลายเซ็นถูกส่งมาจากผู้ส่งจริง

ความปลอดภัยของลายเซ็นดิจิทัลอาร์เอสเอขึ้นอยู่กับความยากของการแยกตัวประกอบ เช่นเดียวกันกับวิทยาการรหัสลับอาร์เอสเอ ดังนั้นขนาดของ n ควรมีค่าไม่น้อยกว่า 1024 บิต และจำนวนเฉพาะที่เป็นตัวประกอบควรเป็นจำนวนเฉพาะที่แข็งแกร่งที่ยากแก่การโจมตี

8. ลายเซ็นดิจิทัลเส้นโค้งเชิงวงรี (Elliptic Curve Digital Signature)

ลายเซ็นดิจิทัลเส้นโค้งเชิงวงรี จะมีรูปแบบของสมการที่ใช้สำหรับการเซ็นลายเซ็น (การเข้ารหัส) และการตรวจสอบลายเซ็น (การถอดรหัส) แตกต่างจากวิทยาการรหัสลับเส้นโค้งเชิงวงรีที่ใช้สำหรับการรักษาความปลอดภัยข้อมูลข่าวสาร โดยในหัวข้อนี้จะกล่าวถึงเพียงลายเซ็นดิจิทัลเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะแบ่งออกเป็น 3 กระบวนการดังนี้

กระบวนการที่ 1 การก่อกำเนิดกุญแจ: เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจ หรือผู้เซ็นลายเซ็น (กำหนดเป็น ผู้ส่ง) มีลำดับการทำงานเป็นดังนี้

1. เลือกสมการ $y^2 = x^3 + ax + b \bmod p$
2. เลือกจุดกำเนิดบนสมการเส้นโค้งจากขั้นตอนที่ 1 กำหนดเป็น $P = (x_p, y_p)$
3. เลือก $a \in \{2, 3, 4, \cdots, N - 1\}$
4. คำนวณจุด $Q = aP$

กุญแจสาธารณะคือ $\{N, Q, P, p\}$

กุญแจส่วนตัวคือ $\{a\}$

กระบวนการที่ 2 การเซ็นลายเซ็นดิจิทัล (เข้ารหัสลับ): เป็นกระบวนการที่ถูกดำเนินการโดยผู้ก่อกำเนิดกุญแจผู้ซึ่งต้องการยืนยันลายเซ็นของตนเอง (สมมติลายเซ็นคือ $M = (x_m, y_m)$) โดยใช้กุญแจส่วนตัว มีลำดับการทำงานเป็นดังนี้

1. เลือก $b \in \{2, 3, 4, \cdots, N - 1\}$
2. คำนวณ $R = bP = (x_r, y_r)$
3. คำนวณ $r = x_r \bmod N$ โดยที่หาก $r = 0$ จะต้องย้อนกลับไปขั้นตอนที่ 1 เพื่อเลือก b ค่าใหม่
4. คำนวณ $s = b^{-1}(x_m + ar) \bmod N$ โดยที่หาก $r = 0$ จะต้องย้อนกลับไปขั้นตอนที่ 1 เพื่อเลือก b ค่าใหม่

หลังเสร็จสิ้นกระบวนการเซ็นลายเซ็นแล้วได้ลายเซ็นดิจิทัลที่ได้คือ {*r*, *s*} ซึ่งจะถูกส่งไปยังผู้รับพร้อมกับ *M*

กระบวนการที่ 3 การตรวจสอบลายเซ็นดิจิทัล (การถอดรหัสลับ): หลังจากที่ผู้รับได้รับ {*r*, *s*} จากผู้ส่งจะสามารถตรวจสอบลายเซ็นได้โดยสมการต่อไปนี้

1. คำนวณ $c = s^{-1} \bmod N$
2. คำนวณ $d = x_m c \bmod N$
3. คำนวณ $e = rc \bmod N$
4. คำนวณ $F = dP + eQ = (x_f, y_f)$
5. คำนวณ $g = x_f \bmod N$

จากทั้ง 3 กระบวนการข้างต้นหากไม่มีการปลอมแปลงเอกสารเกิดขึ้น เมื่อเสร็จสิ้นกระบวนตรวจสอบลายเซ็นแล้วพบว่า *g* มีค่าเท่ากับ *r* เสมอ

ตัวอย่างที่ 10.9 การประยุกต์ใช้ลายเซ็นดิจิทัลเส้นโค้งเชิงวงรี

วิธีทำ

กระบวนการก่อกำเนิดกุญแจ

1. เลือกสมการ $y^2 = x^3 + x + 6 \bmod 11$, $N = 13$
2. เลือกจุดกำเนิดบนสมการเส้นโค้งจากขั้นตอนที่ 1 กำหนดเป็น $P = (3, 5)$
3. เลือก $a \in 3$
4. คำนวณจุด $Q = aP = 3P$

เริ่มจากการคำนวณหาค่า *2P*,

จาก $$m = \frac{3x_p^2 + a}{2y_p} \bmod p$$

$$= \frac{3x3^2 + 1}{2x5} \bmod 11$$

$$= \frac{28}{10} \bmod 11$$

$$= \frac{6}{10} \bmod 11$$

$$= 6 \times 10^{-1} \bmod 11$$

$$= 6 \times 10 \bmod 11$$

$= 60 \bmod 11$

$= 5$

ดังนั้น $x_{2p} = m^2 - 2x_p \bmod p$

$= 5^2 - 2 \times 3 \bmod 11$

$= 8$

และ $y_{2p} = m(x_p - x_{2p}) - y_p \bmod p$

$= 5 \times (3 - 8) - 5 \bmod 11$

$= 3$

และคำนวณหา $Q = 3P = 2P + P$

จาก $m = \dfrac{y_{2p} - y_p}{x_{2p} - x_p} \bmod p$

$= \dfrac{3-5}{8-3} \bmod 11$

$= \dfrac{-2}{5} \bmod 11$

$= \dfrac{9}{5} \bmod 11$

$= 9 \times 5^{-1} \bmod 11$

$= 9 \times 9 \bmod 11$

$= 4$

ดังนั้น $x_{3p} = m^2 - x_{2p} - x_p \bmod p$

$= 4^2 - 8 - 3 \bmod 11$

$= 5$

และ $y_{3p} = m(x_p - x_{3p}) - y_p \bmod p$

$= 4 \times (3 - 5) - 5 \bmod 11$

$= 9$

ดังนั้นได้ $Q = (5, 9)$

เพราะฉะนั้น, กุญแจสาธารณะคือ $\{N = 13, Q = (5, 9), P = (3, 5), p = 11\}$

กุญแจส่วนตัวคือ $\{a = 3\}$

กระบวนการเซ็นลายเซ็นดิจิทัล (เข้ารหัสลับ), กำหนดให้ $M = (4, 4)$

1. เลือก $b = 2$
2. คำนวณ $R = 2P$

เริ่มจากการคำนวณหาค่า $2P$ ซึ่งจากขั้นตอนการก่อกำเนิดกุญแจได้ $2P = (8, 3)$

หมายเหตุ: ในทางปฏิบัติต้องคำนวณหา $2P$ ด้วย เนื่องจากการคำนวณหา $2P$ ในขั้นตอนก่อกำเนิดกุญแจเป็นการคำนวณในฝั่งผู้รับ แต่การคำนวณ $2P$ ในขั้นตอนเซ็นลายเซ็นดิจิทัลเป็นการคำนวณในฝั่งผู้ส่ง ดังนั้นจึงจำเป็นต้องคำนวนใหม่

3. $r = x_r \bmod N$
   $= 8$ mop 13
   $= 8$
4. $s = 2^{-1}(x_m + ar) \bmod N$
   $= 2^{-1} \times (4 + 3 \times 8) \bmod 13$
   $= 2^{-1} \times 28 \bmod 13$
   $= 2^{-1} \times 2 \bmod 13$
   $= 1$

ส่ง $\{r = 8, s = 1\}$ และ $M = (4, 4)$ ไปยังผู้รับ

กระบวนการที่ 3 การตรวจสอบลายเซ็นดิจิทัล (การถอดรหัสลับ)

1. $c = 1^{-1} \bmod N$
   $= 1^{-1} \bmod 13$
   $= 1$
2. $d = x_m c \bmod N$
   $= 4 \times 1 \bmod 13$
   $= 4$
3. $e = rc \bmod N$
   $= 8 \times 1 \bmod 13$
   $= 8$
4. $F = dP + eQ = 4P + 8Q = (x_f, y_f)$

เริ่มจากการคำนวณหาค่า $2P$ ซึ่งจากขั้นตอนการก่อกำเนิดกุญแจได้ $2P = (8, 3)$

คำนวณหา $4P = 2P + 2P$,

จาก $$m = \frac{3x_{2p}^2 + a}{2y_{2p}} \bmod p$$

$$= \frac{3x8^2 + 1}{2x3} \bmod 11$$

$$= \frac{193}{6} \bmod 11$$

$$= \frac{6}{6} \bmod 11$$

$$= 6 \times 6^{-1} \bmod 11$$

$$= 1$$

ดังนั้น $$x_{4p} = m^2 - 2x_{2p} \bmod p$$

$$= 1^2 - 2 \times 8 \bmod 11$$

$$= 7$$

และ $$y_{4p} = m(x_{2p} - x_{4p}) - y_{2p} \bmod p$$

$$= 1 \times (8 - 7) - 3 \bmod 11$$

$$= 9$$

ดังนั้นได้ $4P = (7, 9)$

ขั้นตอนถัดไปคำนวณ $8Q$

เริ่มจากการคำนวณหาค่า $2Q = Q + Q$,

จาก $$m = \frac{3x_q^2 + a}{2y_q} \bmod p$$

$$= \frac{3x5^2 + 1}{2x9} \bmod 11$$

$$= \frac{76}{18} \bmod 11$$

$$= \frac{10}{7} \bmod 11$$

$$= 10 \times 7^{-1} \bmod 11$$

$$= 10 \times 8 \bmod 11$$

$$= 80 \bmod 11$$

$$= 3$$

ดังนั้น $$x_{2q} = m^2 - 2x_q \bmod p$$

$= 3^2 - 2 \times 5 \bmod 11$

$= 10$

และ $y_{2q} = m(x_q - x_{2q}) - y_q \bmod p$

$= 3 \times (5 - 10) - 9 \bmod 11$

$= 9$

คำนวณหาค่า $4Q = 2Q + 2Q$,

จาก $m = \dfrac{3x_{2q}^2 + a}{2y_{2q}} \bmod p$

$= \dfrac{3x10^2 + 1}{2x9} \bmod 11$

$= \dfrac{301}{18} \bmod 11$

$= \dfrac{4}{7} \bmod 11$

$= 4 \times 7^{-1} \bmod 11$

$= 4 \times 8 \bmod 11$

$= 10$

ดังนั้น $x_{4q} = m^2 - 2x_{2q} \bmod p$

$= 10^2 - 2 \times 10 \bmod 11$

$= 3$

และ $y_{4q} = m(x_{2q} - x_{4q}) - y_{2q} \bmod p$

$= 10 \times (10 - 3) - 9 \bmod 11$

$= 6$

คำนวณหาค่า $8Q = 4Q + 4Q$,

จาก $m = \dfrac{3x_{4q}^2 + a}{2y_{4q}} \bmod p$

$= \dfrac{3x3^2 + 1}{2x6} \bmod 11$

$= \dfrac{28}{12} \bmod 11$

$$= \frac{6}{1} \bmod 11$$

$$= 6 \times 1^{-1} \bmod 11$$

$$= 6$$

ดังนั้น $x_{8q} = m^2 - 2x_{4q} \bmod p$

$$= 6^2 - 2 \times 3 \bmod 11$$

$$= 8$$

และ $y_{8q} = m(x_{4q} - x_{8q}) - y_{4q} \bmod p$

$$= 6 \times (3 - 8) - 6 \bmod 11$$

$$= 8$$

ดังนั้นได้ $8Q = (8, 8)$

ดังนั้น $F = 4P + 8Q = (7, 9) + (8, 8)$

จาก $m = \dfrac{y_{4p} - y_{8q}}{x_{4p} - x_{8q}} \bmod p$

$$= \frac{9-8}{7-8} \bmod 11$$

$$= \frac{1}{-1} \bmod 11$$

$$= \frac{1}{10} \bmod 11$$

$$= 1 \times 10^{-1} \bmod 11$$

$$= 1 \times 10 \bmod 11$$

$$= 10$$

ดังนั้น $x_f = m^2 - x_{4p} - x_{8q} \bmod p$

$$=10^2 - 7 - 8 \bmod 11$$

$$= 8$$

และ $y_f = m(x_{8q} - x_f) - y_{8q} \bmod p$

$$= 10 \times (8 - 8) - 8 \bmod 11$$

$$= 3$$

ดังนั้นได้ $4P + 8Q = (8, 3)$

5. $g = x_f \bmod N$

$= 8 \bmod 13$

$= 8$

เนื่องจาก $g = r$ ผู้รับจึงมั่นใจได้ว่าลายเซ็นถูกส่งมาจากผู้ส่งจริง

เนื่องจากความปลอดภัยของลายเซ็นดิจิทัลเส้นโค้งเชิงวงรีขึ้นอยู่กับปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี เช่นเดียวกันกับวิทยาการรหัสลับเส้นโค้งเชิงวงรี ดังนั้นขนาดของ p ควรมีค่าไม่น้อยกว่า 160 บิตเพื่อหลีกเลี่ยงการโจมตีโดยผู้ไม่ประสงค์ดี

9. ความปลอดภัยของลายเซ็นดิจิทัล

ความปลอดภัยของลายเซ็นดิจิทัลแต่ละประเภทนั้นจะเหมือนกับวิทยาการรหัสลับแบบกุญแจสาธารณะของแต่ละประเภทดังนี้ ความปลอดภัยของลายเซ็นดิจิทัลเอ็ลกามอลจะเหมือนกับวิทยาการรหัสลับเอ็ลกามอลคือขึ้นอยู่กับความยากของปัญหาวิยุตลอการิทึม ความปลอดภัยของลายเซ็นดิจิทัลอาร์เอสเอจะเหมือนกับวิทยาการรหัสลับอาร์เอสเอคือขึ้นอยู่กับความยากของการแยกตัวประกอบ และความปลอดภัยของลายเซ็นดิจิทัลเส้นโค้งเชิงวงรีจะเหมือนกับวิทยาการรหัสลับเส้นโค้งเชิงวงรีคือขึ้นอยู่กับความยากของปัญหาวิยุตลอการิทึมเส้นโค้งเชิงวงรี เนื่องจากหากผู้ไม่ประสงค์ดีสามารถแก้ปัญหาเหล่านี้ได้ จะส่งผลให้ทราบค่ากุญแจส่วนตัวของขั้นตอนวิธีในแต่ละประเภท จึงสามารถใช้กุญแจส่วนตัวปลอมลายเซ็นของเจ้าของกุญแจได้ อย่างไรก็ตามยังมีวิธีอื่นๆ ที่สามารถนำมาใช้สำหรับคุกคามลายเซ็นดิจิทัลได้อีกหลายวิธี โดยมีตัวอย่างเป็นดังต่อไปนี้

9.1 การสวมรอยเป็นเจ้าของกุญแจ

การสวมรอยเป็นเจ้าของกุญแจคือ ผู้บุกรุกทำการสร้างกุญแจคู่สำหรับวิทยาการรหัสลับแบบกุญแจสาธารณะขึ้นมาเป็นของตนเอง และส่งกุญแจสาธารณะไปยังผู้รับและใช้อุบายเพื่อหลอกให้ผู้รับเชื่อว่ากุญแจสาธารณะดังกล่าวเป็นของผู้ส่ง ซึ่งหากผู้รับหลงเชื่อจะส่งผลให้ลายเซ็นดิจิทัลที่ผู้รับได้รับมาหลังจากใช้กุญแจสาธารณะนั้นเป็นลายเซ็นที่ถูกเซ็นโดยผู้บุกรุก

9.2 การปลอมลายเซ็นโดยใช้ค่าแฮชที่ตรงกัน

สมมติเลขานุการของบริษัทแห่งหนึ่งเป็นผู้ไม่ประสงค์ดี โดยปกติเลขานุการจะมีหน้าที่ร่างเอกสารให้ผู้บริหารเซ็นรับรองก่อนที่จะนำส่งเอกสารไปยังแผนกที่เกี่ยวข้อง สมมติเลขานุการต้องการหลอกลวงเพื่อให้บริษัทโอนเงินให้เป็นจำนวน 1 ล้านบาท จะสามารถดำเนินการได้โดยมีวิธีการดังนี้

ลำดับแรกเลขานุการสร้างเอกสารที่สามารถนำไปให้ผู้บริหารเซ็นได้ทั้งหมด $2^{\frac{m}{2}}$ ชุดเมื่อ m แทนจำนวนบิตของค่าแฮช โดยข้อความทั้งหมดที่สร้างจะมีความหมายที่คล้ายคลึงกันทั้งหมด เพียงแต่จะมีความแตกต่างเกี่ยวกับการใช้คำศัพท์ที่แตกต่างกันในบางคำ หรือการเว้นช่องว่างระหว่างคำ และคำนวณหาค่าแฮชของข้อความแต่ละชุดแสดงดังตารางที่ 10.3

ตารางที่ 10.3 ตัวอย่างข้อความที่จะถูกเลือกเพื่อให้ผู้บริหารเซ็นรับรอง

ลำดับที่	ข้อความ	ค่าแฮช
1	ขอให้พนักงานทุกท่านร่วมกันรักษาความสะอาด	1AE4287C
2	ขอ ให้พนักงานทุกท่านร่วมกันรักษาความสะอาด	286DBC91
3	เชิญชวนทุกท่านร่วมกันรณรงค์รักษาความสะอาด	3FE4278C
4	เชิญชวน ทุกท่านร่วมกันรณรงค์รักษาความสะอาด	57766C12
. . .	. . .	
$2^{\frac{m}{2}}$	เชิญชวนพนักงานทุกท่านช่วยกันรักษาความสะอาด	36486CC1

ตารางที่ 10.4 ตัวอย่างข้อความที่จะถูกเลือกแต่ไม่ถูกเปิดเผยต่อผู้บริหาร

ลำดับที่	ข้อความ	ค่าแฮช
1	เรียนฝ่ายการเงินกรุณาโอนเงินจำนวน 1 ล้านบาทให้นางสาว ข	11122233
2	เรียน ฝ่ายการเงินกรุณาโอนเงินจำนวน 1 ล้านบาทให้นางสาว ข	57766C12
3	แจ้งฝ่ายการเงินกรุณาโอนเงิน 1 ล้านบาทให้นางสาว ข	44455AAF
4	แจ้งการเงินกรุณาโอนเงินจำนวน 1 ล้านบาทให้นางสาว ข ด่วน	77668899
. . .	. . .	
$2^{\frac{m}{2}}$	เรียนหัวหน้าฝ่ายการเงินโอนเงินเข้าบัญชีนางสาว ข จำนวน 1 ล้านบาท	11223344

หมายเหตุ: ค่าแฮชดังตารางที่ 10.3 และ 10.4 คือค่าสมมติไม่ใช่ค่าจริง

หลังจากนั้นเลขานุการทำการสร้างเอกสารอีกทั้งหมด $2^{\frac{m}{2}}$ ชุด โดยข้อความทั้งหมดของกลุ่มนี้จะมีความหมายที่คล้ายคลึงกันทั้งหมด เพียงแต่จะมีความแตกต่างเกี่ยวกับการใช้คำศัพท์ที่แตกต่างกันในบางคำ หรือการเว้นช่องว่างระหว่างคำ และข้อความในกลุ่มนี้จะไม่ถูกเปิดเผยให้ผู้บริหารทราบ แสดงตัวอย่างให้เห็นดังตารางที่ 10.4

จากหลักการของวันเกิดผิดปกติได้ว่ามีความเป็นไปได้สูงที่จะเกิดค่าแฮชที่ตรงกันระหว่างแถวใดแถวหนึ่งจากตารางที่ 10.3 และตารางที่ 10.4 ซึ่งจากตัวอย่างพบว่าค่าแฮชของข้อความลำดับที่ 4 ของตารางที่ 10.3 มีค่าตรงกับค่าแฮชของข้อความลำดับที่ 2 ของตารางที่ 10.4

เมื่อพบผลลัพธ์ดังกล่าวเลขานุการจึงดำเนินการต่อดังนี้

เลือกข้อความลำดับที่ 4 จากตารางที่ 10.3 คือ “เชิญชวน ทุกท่านร่วมกันรณรงค์รักษาความสะอาด” เพื่อนำไปให้ผู้บริหารเซ็น และเมื่อผู้บริหารเห็นข้อความดังกล่าวแล้วพบว่าไม่เป็นข้อความอันตรายและเป็นประโยชน์ต่อหน่วยงานจึงยอมเซ็นค่าแฮชของเอกสารดังกล่าว ซึ่งเท่ากับว่าผู้บริหารเซ็นค่าแฮชของข้อความลำดับที่ 2 จากตารางที่ 10.4 คือ “เรียน ฝ่ายการเงินกรุณาโอนเงินจำนวน 1 ล้านบาทให้นางสาว ข” ด้วยเช่นกัน

หลังจากได้ลายเซ็นจากผู้บริหารแล้วเลขานุการนำข้อความที่ประสงค์ร้ายคือ “เรียน ฝ่ายการเงินกรุณาโอนเงินจำนวน 1 ล้านบาทให้นางสาว ข” พร้อมลายเซ็นผู้บริหารเสนอฝ่ายการเงิน

เมื่อฝ่ายการเงินทำการตรวจสอบลายเซ็นและค่าแฮชของข้อความที่ได้รับพบว่ามีค่าตรงกันจึงเชื่อได้ว่าเอกสารดังกล่าวถูกส่งมาจากผู้บริหารจริง ดังนั้นฝ่ายการเงินจึงยอมโอนเงินให้เลขานุการ

10. บทสรุป

ฟังก์ชันแฮชหรือที่นิยมถูกเรียกอีกชื่อว่าแมสเสจไดเจสท์คือฟังก์ชันที่รับข้อมูลนำเข้าที่มีความยาวไม่แน่นอนสำหรับการประมวลผลและได้ผลลัพธ์เป็นข้อมูลที่มีความยาวคงที่ ซึ่งเรียกผลลัพธ์นี้ว่าค่าแฮช โดยฟังก์ชันแฮชเป็นฟังก์ชันแบบทิศทางเดียวกล่าวคือกระบวนการหาค่าแฮชของข้อความต้นฉบับสามารถดำเนินการได้อย่างง่ายดาย แต่ในทางกลับกันการคำนวณหาข้อความต้นฉบับจากค่าแฮชทำได้ยากมาก ฟังก์ชันแฮชที่ดีจะต้องมีคุณลักษณะปลอดการชนกล่าวคือหากนำข้อความต้นฉบับจำนวนมากไปผ่านฟังก์ชันแฮชแล้วค่าแฮชของข้อความต้นฉบับแต่ละค่าต้องมีความแตกต่างกัน นอกเหนือจากนั้นถึงแม้ว่าข้อความต้นฉบับจะมีค่าที่ใกล้เคียงกัน ค่าแฮชของข้อความเหล่านั้นไม่จำเป็นต้องมีค่าใกล้เคียงกัน ฟังก์ชันแฮชได้ถูกพัฒนาออกมาหลายวิธีซึ่งขั้นตอนวิธี Secure Hash Algorithm 1 (SHA-1) เป็นขั้นตอนวิธีหนึ่งที่มีความปลอดภัย โดยข้อมูลนำเข้าจะมีขนาด 512 บิต และผลลัพธ์ที่ได้เป็นค่าแฮชมีขนาด 160 บิต ซึ่งหากข้อมูลมีขนาดเกินค่าที่กำหนดจำเป็นต้องแบ่งข้อมูลเป็นบล็อคที่มีขนาดบล็อคละ 512 บิตก่อนเพื่อนำข้อมูลเข้าสู่ SHA-1 ทีละ 1 บล็อค อย่างไรก็

ตามกรณีที่ข้อมูลต้นฉบับมีขนาดไม่ถึง 512 บิตจำเป็นต้องเพิ่มบิตให้ครบตามกำหนดก่อนจึงจะสามารถนำข้อมูลเข้าสู่ SHA-1 ได้ ถึงแม้ว่าปัจจุบันฟังก์ชันแฮชได้ถูกปรับปรุงให้มีประสิทธิภาพที่สูงขึ้น เช่น SHA-256 และ SHA-512 แต่พื้นฐานของการคำนวณทางคณิตศาสตร์จะมีความคล้ายคลึงกับ SHA-1 เพียงแต่มีขั้นตอน และรอบการคำนวณที่เพิ่มมากขึ้น ดังนั้นในบทนี้จึงอธิบายขั้นตอนวิธี SHA-1 ซึ่งผู้อ่านสามารถนำไปประยุกต์กับฟังก์ชันแฮชรุ่นใหม่ในปัจจุบันได้ ซึ่งฟังก์ชันแฮชถูกนำมาประยุกต์ใช้งานร่วมกับลายเซ็นดิจิทัลโดยการตรวจสอบความถูกต้องของค่าแฮชของลายเซ็นแทนที่การตรวจสอบลายเซ็นโดยตรง

ลายเซ็นดิจิทัล คือวิธีการที่ใช้สำหรับยืนยันเอกสารอิเล็กทรอนิกส์เพื่อเป็นการยืนยันให้ผู้รับมั่นใจว่าเอกสารดังกล่าวถูกส่งมาจากผู้ส่งจริง โดยสามารถนำวิทยาการรหัสลับแบบกุญแจสาธารณะ เช่น วิทยาการรหัสลับเอ็ลกามอล วิทยาการรหัสลับอาร์เอสเอ และวิทยาการรหัสลับเส้นโค้งเชิงวงรีมาประยุกต์ใช้งานเป็นลายเซ็นดิจิทัลได้ ซึ่งความปลอดภัยของลายเซ็นดิจิทัลแต่ละประเภทนั้นจะเหมือนกับวิทยาการรหัสลับแบบกุญแจสาธารณะของแต่ละประเภท อย่างไรก็ตามยังมีวิธีอื่นๆ ที่สามารถนำมาใช้สำหรับคุกคามลายเซ็นดิจิทัลได้อีกหลายวิธีดังนี้ การสวมรอยเป็นเจ้าของกุญแจคือ ผู้บุกรุกทำการสร้างกุญแจคู่สำหรับวิทยาการรหัสลับแบบกุญแจสาธารณะขึ้นมาเป็นของตนเอง และส่งกุญแจสาธารณะไปยังผู้รับและใช้อุบายเพื่อหลอกให้ผู้รับเชื่อว่ากุญแจสาธารณะดังกล่าวเป็นของผู้ส่ง และการปลอมลายเซ็นโดยใช้ค่าแฮชที่ตรงกัน คือเลือกข้อความปกติ และข้อความอันตรายที่มีค่าแฮชตรงกันแล้วนำข้อความปกติไปให้เจ้าของกุญแจเซ็นลายเซ็นดิจิทัล ดังนั้นการประยุกต์ใช้งานจริงจึงจำเป็นต้องเลือกใช้ค่าพารามิเตอร์ที่มีความแข็งแกร่ง และมีขนาดใหญ่เพื่อหลีกเลี่ยงการโจมตีด้วยขั้นตอนวิธีต่างๆ ให้ได้ทั้งหมด

แบบฝึกหัดท้ายบท
บทที่ 10

1. ฟังก์ชันแฮชมีชื่อเรียกอีกชื่อหนึ่งว่าอะไร
2. ฟังก์ชันแฮชที่มีประสิทธิภาพควรมีคุณลักษณะเป็นอย่างไร
3. กำหนดให้ h = “1AE4 73F5 BBCE F7C9 112A 4756 661F” จงวิเคราะห์ว่าค่า h ดังกล่าวเป็นไปได้หรือไม่ที่จะเป็นค่าแฮชจาก MD2
4. ค่าแฮชที่ได้จากขั้นตอนวิธี SHA-1 จะมีขนาดกี่บิต
5. ข้อมูลอินพุตสำหรับขั้นตอนวิธี SHA-1 จะมีขนาดกี่บิต
6. กำหนดให้ m = “FFAA4433 11FFDDED 61289A4E 551EB611 8899ACFE AABBCC11 213CBA17 133ECFA1 11122233 451897A1” จงหาบิตเติมเต็ม (ที่ไม่รวม 64 บิตสุดท้าย)
7. จากผลลัพธ์ข้อ 6 จงหา t
8. กำหนดให้ข้อความต้นฉบับที่ผ่านขั้นตอนการขยายบิตเป็นดังนี้
14576AF1 23480000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000000C2
จงหา W(0) และ W(1)
9. กำหนดให้ $A = D3_{16}$ และ $B = 4C_{16}$ จงหา $A \wedge B$
10. กำหนดให้ $A = D3_{16}$ และ $B = 4C_{16}$ จงหา $A \vee B$
11. สมมติมีการเลือกประชากรมาทั้งหมด 10 คนที่มีเงื่อนไขว่าต้องเกิดอยู่ระหว่างวันที่ 1 – 10 ของแต่ละเดือน จงหาความน่าจะเป็นที่สุ่มประชากรจากกลุ่มดังกล่าวมา 3 คนแล้วมีโอกาสที่จะมีวันเกิดที่แตกต่างกันทั้งหมด (ไม่สนใจเดือน และปีเกิด)
12. จากเงื่อนไขของประชากรที่เลือกมาทั้ง 10 คนที่อยู่ในคำถามข้อ 11 จงหาความน่าจะเป็นที่สุ่มประชากรจากกลุ่มดังกล่าวมา 3 คนแล้วมีโอกาสที่จะมีวันเกิดวันเดียวกันอย่างน้อย 1 คู่
13. จากเงื่อนไขของประชากรที่เลือกมาทั้ง 10 คนที่อยู่ในคำถามข้อ 11 จงหาจำนวนประชากรที่เลือกมาแล้วส่งผลให้ความน่าจะเป็นของวันเกิดที่ต่างกันทั้งหมด และความน่าจะเป็นของการเกิดวันเดียวกันอย่างน้อย 1 คู่มีค่าที่ใกล้เคียงกัน
14. จงอธิบายความแตกต่างของการนำวิทยาการรหัสลับมาประยุกต์ใช้กับการรักษาความปลอดภัยและลายเซ็นดิจิทัล

15. กำหนดให้ m = 18 และค่าแฮชของ m คือ 126 จงหาผลลัพธ์ที่ได้หลังจากการถอดรหัส (การตรวจสอบลายเซ็น) โดยใช้วิทยาการรหัสลับแบบกุญแจสาธารณะ

16. จงอธิบายสาเหตุที่ค่าแฮชของข้อมูลจากแถวที่ 4 ของตารางที่ 10.3 มีค่าเท่ากับข้อมูลจากแถวที่ 2 ของตารางที่ 10.4

17. กำหนดให้ n = 497039, e = 11 และ d = 180227 และค่าแฮชของข้อความต้นฉบับมีค่าเท่ากับ 29 จงหาลายเซ็นของค่าแฮชดังกล่าวโดยใช้วิทยาการรหัสลับอาร์เอสเอ

บรรณานุกรม

1. สัญจกร วุฒิสิทธิกุลกิจ, ธงชัย โรจน์กังสดาล, วรากร ศรีเชวงทรัพย์ และ นพดล พรมภักษร. (2548). **วิทยาการรหัสลับเบื้องต้น**. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
2. วรเศรษฐ สุวรรณิก. (2553). **วิทยาการรหัสลับ**. กรุงเทพฯ: สำนักพิมพ์วรรณิก.
3. พิศิษฏ์ โภคารัตน์กุล, รังสิพรรณ มฤคทัต, สุรทศ ไตรติลานันท์, ทรงพล องค์วัฒนกุล, ธนัสนี เพียรตระกุล, วศิน สุทธิฉายา และ ชาญยุทธ ดิษฐศิริ. (2558). **วิทยาการรหัสลับในระบบเทคโนโลยีสารสนเทศและโทรคมนาคม**. กรุงเทพฯ: บริษัทรับพิมพ์ จำกัด.
4. ดำรงค์ ทิพย์โยธา. (2556). **โลกทฤษฎีจำนวน**. กรุงเทพฯ: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
5. พิเชษฐ เชี่ยวธนะกุล. (2556). **วิทยาการรหัสลับเชิงคณิตศาสตร์**. ขอนแก่น: โรงพิมพ์มหาวิทยาลัยขอนแก่น.
6. Wiles, A. (1995). Modular elliptic curves and Fermat's last theorem. **Annals of Mathematics, 141**, 443–551.
7. Diffie, W. and Hellman, M. (1976). New directions in cryptography. **Journal of IEEE Transactions on Information Theory, 22**(6), 644-654.
8. Rivest, R.L., Shamir, A. and Adleman, L. (1978). A method for obtaining digital signatures and public key cryptosystems. **Journal of Communications of the ACM, 21**(2), 120–126.
9. Koblitz, N. (1987). Elliptic Curve Cryptosystems. **Mathematics of Computation**, **48**, 203–209.
10. Miller, V.S. (1986). Uses of elliptic curves in cryptography. **Lecture Notes in Computer Science, 218**, 417–428.
11. Pollard, J.M. (1978). Monte Carlo methods for index computation (mod p). **Journal of Mathematics of Computation, 32**, 918-924.

12. Murat, S. (2011). Generalized Trial Division. **International Journal of Contemporary Mathematical Science**, **6**(2), 59 – 64.

13. Sharma, P., Gupta, A.K. and Vijay, A. (2012). Modified Integer Factorization Algorithm using V-Factor Method. **Proceedings of the Second International Conference on Advanced Computing & Communication Technologies, Jan 7-8 2012.** (pp. 423 – 425). India: RG Education Society.

14. Pollard, J. M. (1974). Theorems of factorization and primality testing. **Mathematical Proceedings of the Cambridge Philosophical Society, 76**(3), 521–528.

15. Ambedkar, B.R., Gupta, A., Gautam, P. and Bedi, S.S. (2011). An Efficient Method to Factorize the RSA Public Key Encryption. **Proceedings of the International Conference on Communication Systems and Network Technologies, June 3-5 2011.** (pp. 108 – 111). Katra: MIR Labs India.

16. Wu, M.E., Tso, R. and Sun, H.M. (2014). On the improvement of Fermat factorization using a continued fraction technique. **Future Generation Computer Systems**, **30**(1), 162 – 168.

17. McKee, J. (1999). Speeding Fermat's factoring method. **Mathematics of Computation, 68**, 1729–1737.

18. Xiang, G. (2004). Fermat's Method of Factorization. **Applied Probability Trust**, **36**(2), 34 – 35.

19. Somsuk, K. and Kasemvilas, S. (2013). MFFV2 And MNQSV2 Improved Factorization Algorithms. **Proceedings of the fourth International Conference on Information Science and Applications, June 24 – 26 2013.** (pp. 1 - 3). Korea: IEEE.

20. Somsuk, K. and Kasemvilas, S. (2014). MFFV3 An Improved Integer Factorization Algorithm to Increase Computation Speed. **Advanced Material Research**, **931-932**, 1432 – 1436.

21. Somsuk, K. (2014). A New Modified Integer Factorization Algorithm Using Integer Mod 20's Technique. **Proceedings of the 18 International Computer Science and Engineering Conference, July 30 – August 1 2014.** (pp. 312 - 316). Thailand: IEEE.

22. Somsuk, K. and Kasemvilas, S. (2014). Possible Prime Modified Fermat Factorization New Improved Integer Factorization to Decrease Computation Time for Breaking RSA. **Proceedings of the 10 International Conference on Computing and Information Technology, May 6 – 8 2014.** (pp. 325 - 334). Thailand: DBLP.

23. Nidhi, L., Anurag, P. and Shishupal, K. (2014). Modified Trial Division Algorithm Using KNJ-Factorization Method To Factorize RSA Public Key Encryption. **Proceedings of the International Conference on Contemporary Computing and Informatics, November 27 – 29 2014.** (pp. 992 - 995). India: IEEE.

24. ElGamal, T. (1985). A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. **IEEE Transactions on Information Theory, 31**, 469 – 472.

25. Trappe, W. and Washington, L. (2005). **Introduction to Cryptography with Coding Theory**. New Jersey: Pearson Education.

26. Hans, S. (1994). **Prime Numbers and Computer Methods for Factorization.** Stockholm: Spinger Science Business Media.

27. Wiener, M. (1990). Cryptanalysis of short RSA secret exponents. **IEEE Transactions on Information Theory, 36**, 553-558.

28. U.S. Department of Commerce/National Institute of Standards and Technology, **Data Encryption Standard (DES)**. Federal Information Processing Standards Publication FIPS PUB 46-3, Reaffirmed 1999 October 25.

29. Daemen, J. and Rijmen, V. (2002). **The Design of Rijndael, AES - The Advanced Encryption Standard**. Germany: Springer-Verlag.

30. Thomos, H., Charles, E., Leiserson, R. and Clifford, S. (2009). **Introduction to Algorithms.** Unitate States: McGraw-Hill Book.

31. Hammad, Y.B., Carter, G. and Dawson, E. (2005). RAK factoring algorithm. **Australasian Journal of Combinatorics, 33**(1), 291 – 305.

32. Huang, Q., Li, Y.T., Zhang, Y. and Lu, C. (2007). A Modified Non-Sieving Quadratic Sieve For Factoring Simple Blur Integers. **Proceedings of the International Conference on Multimedia and Ubiquitous Engineering, April 26-28 2007.** (pp. 729 – 732). Korea: IEEE.

33. Eisentrager, K., Lauter, K. and Montgomery, P.L. (2003). Fast Elliptic curve arithmetic and improved Weil pairing evaluation. **In: Joye, M. (ed.) Lecture Notes in Computer Science, 2612**, 343–354.

34. Somsuk, K. and Tientanopajai, K. (2017). An Improvement of Fermat's Factorization by Considering the Last m Digits of Modulus to Decrease Computation Time. **International Journal of Network Security**, **19**(1), 99 – 111.

35. Somsuk, K. (2018). The improvement of initial value closer to the target for Fermat's factorization algorithm. **Journal of Discrete Mathematical Sciences and Cryptography**, **21**(7-8), 1573 – 1580.

36. Amara, M. and Siad, A. (2011). Elliptic curve cryptography and its application. **Proceedings of the International Workshop on Systems, Signal Processing and their Applications**, **May 9 – 11 2011**. (pp. 247 - 250). Tipaza, Algeria: IEEE.

37. Subhranil, S., Rana, M. and Sandip, D. (2017). Elliptic curve cryptography: a dynamic paradigm. **Proceedings of the International Conference on Infocom Technologies and Unmanned Systems, December 18 – 20 2017**. (pp. 427 - 431). Dubai, UAE: IEEE.

38. Somsuk, K. and Sanemueang, C. (2018). The New Modified Methodology to Solve ECDLP Based on Brute Force Attack. **Proceedings of the 14 International Conference on Computing and Information Technology**, **July 5 – 6 2018.** (pp. 255 - 264). Thailand: Springer International Publishing AG.

39. Somsuk, K. (2017). The new Equation for RSA's Decryption Process Appropriate with High Private Key Exponent. **Proceedings of the 21 International Computer Science and Engineering Conference, November 15 - 18 2017.** (pp. 45 - 48). Thailand: IEEE.

40. Miller, G.L. (1976). Riemann's Hypothesis and Tests for Primality. **Journal of Computer and System Sciences, 13**, 300–317.

41. Rabin, M.O. (1980). Probabilistic algorithm for testing primality. **Journal of Number Theory, 12**, 128–138.

42. Da Costa, C.A., Moreno, R.L., Carpinteiro, O.S.A. and Pimenta, T.C. (2015). Design of a 1024 bit RSA coprocessor with SPI slave interface. **Proceedings of the International Caribbean Conference on Devices, Circuits and Systems**, **April 2 – 4 2014.** (pp. 1-4). Mexico: IEEE.

43. Buchmann, J.A. (2004). **Introduction to Cryptography**. Germany: Springer-Verlag.

44. Kong, F., Zhou, D., Jiang, Y., Shang, J. and Yu, J. (2017). Fault Attack on an Improved CRT-RSA algorithm with the Modulus Chaining Method. **Proceedings of the IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing, July 21 – 24 2017**. (pp. 866 – 869). China: IEEE.

45. Gueron, G. and Drucker, N. (2018). Cryptosystems with a multi prime composite modulus. **Proceedings of the IEEE Annual Consumer Communications & Networking Conference, January 12 – 15 2018**. (pp. 1 – 7). United States: IEEE.

46. Zalaket, J. and Hajj Boutros, J. (2011). Prime factorization using square root approximation. **Computers & Mathematics with Applications**, **61**(9), 2463 – 2467.

47. Somsuk, K., Chiawchanwattana, T. and Sanemueang, C. (2019). Estimating the new Initial Value of Trial DivisionAlgorithm for Balanced Modulus to Decrease Computation Loops. **Proceedings of the 16 International Joint Conference on Computer Science and Software Engineering, July 10-12 2019**. (pp. 143-147). Thailand: IEEE.

48. Elbirt, A.J. (2009). **Understanding and Applying Cryptography and Data Security**. United State: Auerbach Publications.

49. Somsuk, K. and Tientanopajai, K. (2016). Improving fermat factorization algorithm by dividing modulus into three forms. **KKU Engineering Journal, 43**, 350 – 353.

50. Boneh, D. and Durfee, G. (1999). Cryptanalysis of RSA with Private Key d less than $N^{0.292}$. **Lecture Notes in Computer Science, 1592**, 1 – 11.

51. Somsuk, K. (2020). The new integer factorization algorithm based on Fermat's Factorization Algorithm and Euler's theorem. **International Journal of Electrical and Computer Engineering, 10**(2), 1469 – 1476.

52. Hill, L.S. (1929). Cryptography in an Algebraic Alphabet. **The American Mathematical Monthly, 36**, 306–312.

53. Pratt, F. (1939). **Secret and Urgent: The Story of Codes and Ciphers**. United States: Aegean Park Press.

54. Euler, L. (1763). Theoremata arithmetica nova methodo demonstrate, **Novi Commentarii academiae scientiarum Petropolitanae, 8**, 74 – 104.

55. Alfred, J.M. and Scott, A.V. (1993). Elliptic curve cryptosystems and their implementation, **Journal of Cryptology, 6**, 209–224.

56. Fred, A.S. (1973). A homophonic cipher for computational cryptography. **Proceedings of national computer conference and exposition, June 4-8 1973**. (pp. 565-568). United States: ACM.

57. Smith, L.D. (1955). **Cryptography: The Science of Secret Writing**. United States: Dover Publications, Inc.

58. Pohlig, S. and Hellman, M. (1978). An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. **IEEE Transactions on Information Theory, 24**, 106-110.

59. Shanks, D. (1971). Class Number, a Theory of Factorization and Genera. **Proceedings of Symposium of Pure Mathematics, 20**, 415-440.

60. National Institute of Standards and Technology, **Secure Hash Standard.** Federal Information Processing Standards Publication FIPS PUB 180-2, 2002 August 1.

61. Mckinney, E. H. (1966). Generalized Birthday Problem. **The American Mathematical Monthly, 73**(4), 385-387.

62. Hasse, H. (1936). Zur Theorie der abstrakten elliptischen Funktionenkörper I. **Journal für die reine und angewandte Mathematik, 175**, 55 - 62.

ดัชนี

ภาคผนวก

ภาคผนวก ก
ตัวอย่างการใช้งานคลาส BigInteger และโปรแกรมภาษาจาวาสำหรับวิทยาการรหัสลับ

เนื่องจากในการใช้งานจริงวิทยาการรหัสลับแบบกุญแจสาธารณะแต่ละประเภทนั้นจำเป็นต้องใช้กุญแจที่มีขนาดใหญ่ จึงไม่สามารถใช้ชนิดข้อมูลแบบทั่วไปเช่น int, long หรือ double สำหรับการดำเนินการได้เนื่องจากขอบเขตการใช้งานที่เล็กเกินไป

BigInterger คือคลาสชนิดหนึ่งที่เป็นมาตรฐานในโปรแกรมภาษาจาวาที่มีจุดเด่นคือสามารถนำมาใช้งานเป็นชนิดข้อมูลที่มีขนาดไม่จำกัดเนื่องจากเป็นคลาสที่ถูกพัฒนาด้วยสตริง และสามารถนำมาใช้คำนวณได้เหมือนชนิดข้อมูลที่คำนวณได้ทั่วไป ดังนั้นคลาส BigInteger จึงเหมาะที่จะถูกนำมาใช้สำหรับการสร้างขั้นตอนวิธีสำหรับวิทยาการรหัสลับแบบกุญแจสาธารณะ

ในภาคผนวกนี้จะกล่าวถึงคลาส BigInteger และตัวอย่างโปรแกรมภาษาจาวาสำหรับบางขั้นตอนวิธีที่กล่าวไว้ในบทเรียน โดยขั้นตอนวิธีสำหรับวิทยาการรหัสลับแบบสมมาตรบางประเภทไม่จำเป็นต้องใช้คลาส BigInteger เนื่องจากใช้งานกุญแจขนาดเล็ก เช่น รหัสซีซาร์ แต่ในทางกลับกันขั้นตอนวิธีในกลุ่มวิทยาการรหัสลับแบบกุญแจสาธารณะทั้งหมดจำเป็นต้องใช้คลาสดังกล่าวเนื่องจากขนาดกุญแจที่ต้องนำไปใช้งานจริงมีขนาดใหญ่

ก1. คลาส BigInteger

เนื่องจากคลาส BigInteger ถูกเก็บไว้ในไลบรารีชื่อ math.BigInteger ดังนั้นก่อนการสร้างออบเจ็คจำเป็นต้องเรียกใช้งานไลบรารี่ดังกล่าวนี้เสียก่อน

```
import java.math.BigInteger;
```

ก1.1 การสร้างออบเจ็ค

```
BigInteger objectname = new BigInteger(String BigInt);
```

โดยที่ objectname คือชื่อออบเจ็คสำหรับคลาส BigInteger

BigInt คือตัวเลขที่มีขนาดไม่จำกัด

ก1.2 ตัวอย่างเมธอดในคลาส BigInteger

หัวข้อนี้จะกล่าวถึงตัวอย่างเมธอดในคลาส BigInteger ที่จำเป็นต้องใช้งานสำหรับสร้างขั้นตอนวิธีในกลุ่มวิทยาการรหัสลับแบบกุญแจสาธารณะ

1. เมธอด add()

คือเมธอดที่ใช้สำหรับการบวกค่าระหว่างออบเจ็ค

```
Object1.add(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: Object1 + Object2

การคืนค่า: BigInteger

2. เมธอด substract()

คือเมธอดที่ใช้สำหรับการหาผลลบระหว่างออบเจ็ค

```
Object1.subtract(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: Object1 - Object2

การคืนค่า: BigInteger

3. เมธอด multiply()

คือเมธอดที่ใช้สำหรับการหาผลคูณระหว่างออบเจ็ค

```
Object1.multiply(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: Object1 x Object2

การคืนค่า: BigInteger

4. เมธอด divide()

คือเมธอดที่ใช้สำหรับการหาผลหารระหว่างออบเจ็คโดยไม่พิจารณาเศษ

```
Object1.divide(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: Object1 ÷ Object2

การคืนค่า: BigInteger

5. เมธอด mod()

คือเมธอดที่ใช้สำหรับการหาเศษที่ได้จากการหารระหว่างออบเจ็ค

```
Object1.mod(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: Object1 mod Object2

การคืนค่า: BigInteger

6. เมธอด gcd()

คือเมธอดที่ใช้สำหรับการหาค่าหารร่วมมากระหว่างออบเจ็ค

```
Object1.gcd(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: gcd(Object1, Object2)

การคืนค่า: BigInteger

7. เมธอด modInverse()

คือเมธอดที่ใช้สำหรับการหาค่าผกผันในฟิลด์จำกัด

```
Object1.modInverse(Object2)
```

โดยที่ Object1 และ Object2 คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: หาค่า $Object1^{-1}$ ใน GF(Object2)

การคืนค่า: BigInteger

8. เมธอด pow()

คือเมธอดที่ใช้สำหรับการหาค่าผลลัพธ์ของเลขยกกำลัง

```
Object.pow(exponent)
```

โดยที่ Object คือออบเจ็คสำหรับคลาส BigInteger

exponent คือตัวแปรที่มีชนิดข้อมูลเป็น int

การดำเนินการ: $Object^{exponent}$

การคืนค่า: BigInteger

9. เมธอด intValue()

คือเมธอดที่ใช้สำหรับแปลงชนิดข้อมูลของออบเจ็คเป็นจำนวนเต็มแบบ int

```
Object.intValue()
```

โดยที่ Object คือออบเจ็คสำหรับคลาส BigInteger

การดำเนินการ: แปลงชนิดข้อมูลของ Object จากเดิม BigInteger เป็น int

การคืนค่า: int

ก1.3 ตัวอย่างการก่อกำเนิดจำนวนเฉพาะขนาดใหญ่

การสุ่มหาจำนวนเฉพาะขนาดใหญ่จำเป็นต้องกำหนดออบเจ็คของคลาส SecureRandom เพื่อใช้เป็นพารามิเตอร์ตัวสุดท้ายของออบเจ็คที่เป็นจำนวนเฉพาะที่ถูกสร้างด้วยคลาส BigInteger

เนื่องจากคลาส SecureRandom ถูกเก็บไว้ในไลบรารีชื่อ security.SecureRandom ดังนั้นก่อนการสร้างออบเจ็คจำเป็นต้องเรียกใช้งานไลบรารี่ดังกล่าวนี้เสียก่อน

```
import java.security.SecureRandom;
```

โดยการสร้างจำนวนเฉพาะแบบสุ่มสามารถดำเนินการได้ดังนี้

```
BigInteger obj = new BigInteger(int bitlength, int certainly, Random rd);
```

โดยที่ obj คือชื่อออบเจ็คสำหรับคลาส BigInteger

bitlength คือจำนวนบิตสำหรับ obj

certainly คือ ค่าความเป็นไปได้ที่จะทำให้โอกาสของตัวเลขที่ได้ออกมาเป็นจำนวนเฉพาะซึ่งคำนวณจากสูตร $1-(0.5)^{centainly}$

rd คือออบเจ็คของคลาส SecureRandom

ก2. ตัวอย่างโปรแกรมภาษาจาวาสำหรับขั้นตอนวิธีทางวิทยาการรหัสลับ

ในหัวข้อนี้จะกล่าวถึงตัวอย่างโปรแกรมภาษาจาวาสำหรับขั้นตอนวิธีทางวิทยาการรหัสลับบางส่วนที่ได้กล่าวไว้ภายในบทเรียน

1) ขั้นตอนวิธีที่ 1.2 (ขั้นตอนวิธียุคลิด)

```
package gcd_test;
import java.util.Scanner;
public class Gcd_test {
    public static int gcd(int a, int b){
        int[] r = new int[a];
        int[] q = new int[a];
        r[0] = a;
        r[1] = b;
        int k = 1;
        while(r[k] != 0){
            q[k] = r[k-1]/r[k];
            r[k+1] = r[k-1] - q[k]*r[k];
            k = k+1;
        }
        return r[k-1];
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("a: ");
        int a = sc.nextInt();
        System.out.print("b: ");
        int b = sc.nextInt();
        int c = gcd(a, b);
        System.out.println("gcd("+a+", "+b+") = "+c);
    }
}
```

2) ขั้นตอนวิธียุคลิดภาคขยาย

```
package mod_inverse_test;
import java.util.Scanner;
public class Mod_inverse_test {
    public static int mod_inverse(int a, int b){
        int[] r = new int[a];
        int[] q = new int[a];
        int[] x = new int[a];
        int[] y = new int[a];
        r[0] = a; r[1] = b;
        x[0] = 1; y[0] = 0; x[1] = 0; y[1] = 1;
        int k = 1;
        while(r[k] != 0){
            q[k] = r[k-1]/r[k];
            r[k+1] = r[k-1] - q[k]*r[k];
            x[k+1] = q[k]*x[k] + x[k-1];
            y[k+1] = q[k]*y[k] + y[k-1];
            k = k+1;
        }
        if((k-1)%2 == 0){
            y[k-1] = a - y[k-1];
        }
        return y[k-1];
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("a: ");
        int a = sc.nextInt();
        System.out.print("b: ");
        int b = sc.nextInt();
        int c = mod_inverse(a, b);
        System.out.println("inverse of "+b +" modulo "+a+" is "+c);
    }
}
```

3) รหัสซีซาร์

```
package caesar_test;
import java.util.Scanner;
public class Caesar_test {
    public static int convert_StringtoInteger(String m){
        int x=0;
        switch(m){
            case "A": x = 0; break;
```

```
            case "B": x = 1; break;
            case "C": x = 2; break;
            case "D": x = 3; break;
            case "E": x = 4; break;
            case "F": x = 5; break;
            case "G": x = 6; break;
            case "H": x = 7; break;
            case "I": x = 8; break;
            case "J": x = 9; break;
            case "K": x = 10; break;
            case "L": x = 11; break;
            case "M": x = 12; break;
            case "N": x = 13; break;
            case "O": x = 14; break;
            case "P": x = 15; break;
            case "Q": x = 16; break;
            case "R": x = 17; break;
            case "S": x = 18; break;
            case "T": x = 19; break;
            case "U": x = 20; break;
            case "V": x = 21; break;
            case "W": x = 22; break;
            case "X": x = 23; break;
            case "Y": x = 24; break;
            case "Z": x = 25; break;
        }
        return x;
    }
        public static String convert_IntegertoString(int x){
        String m="";
        switch(x){
            case 0: m = "A"; break;
            case 1: m = "B"; break;
            case 2: m = "C"; break;
            case 3: m = "D"; break;
            case 4: m = "E"; break;
            case 5: m = "F"; break;
            case 6: m = "G"; break;
            case 7: m = "H"; break;
            case 8: m = "I"; break;
            case 9: m = "J"; break;
            case 10: m = "K"; break;
            case 11: m = "L"; break;
            case 12: m = "M"; break;
            case 13: m = "N"; break;
```

```
            case 14: m = "O"; break;
            case 15: m = "P"; break;
            case 16: m = "Q"; break;
            case 17: m = "R"; break;
            case 18: m = "S"; break;
            case 19: m = "T"; break;
            case 20: m = "U"; break;
            case 21: m = "V"; break;
            case 22: m = "W"; break;
            case 23: m = "X"; break;
            case 24: m = "Y"; break;
            case 25: m = "Z"; break;
        }
        return m;
    }
    public static String encrypt(String p, int k){
        String tmp;
        int m;
        int c;
        String cipher ="";
        for(int i=0;i<p.length();i++){
            tmp = p.substring(i,i+1);
            m = convert_StringtoInteger(tmp);
            c = (m+k)%26;
            cipher = cipher+convert_IntegertoString(c);
        }
        return cipher;
    }
        public static String decrypt(String c, int k){
        String tmp;
        int m;
        int p;
        String plaintext ="";
        for(int i=0;i<c.length();i++){
            tmp = c.substring(i,i+1);
            m = convert_StringtoInteger(tmp);
            p = (m-k)%26;
            if(p < 0){
                p = 26 + p;
            }
            plaintext = plaintext+convert_IntegertoString(p);
        }
        return plaintext;
    }
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        System.out.print("Plaintext: "); //ตัวพิมพ์ใหญ่
        String p = sc.nextLine();
        System.out.print("Key: ");
        int k = sc.nextInt();
////////////////////////////////////////////////Encrypt/////////////////////////////////////////////////////////////////////////
        String cipher = encrypt(p, k);
        System.out.println("Encrypted message is "+cipher);
////////////////////////////////////////////////Decrypt/////////////////////////////////////////////////////////////////////////
        String plaintext = decrypt(cipher, k);
        System.out.println("Decrypted message is "+plaintext);
    }
}
```

4) รหัสสัมพรรค

```
package affine_test;
import java.util.Scanner;
public class Affine_test {
    public static int convert_StringtoInteger(String m){
        //ใช้ code ชุดเดียวกันกับโปรแกรมรหัสซีซาร์
    }
    public static int mod_inverse(int a, int b){
        //ใช้ code ชุดเดียวกันกับโปรแกรมหาค่าผกผัน (Mod_inverse_test)
    }
        public static String convert_IntegertoString(int x){
            //ใช้ code ชุดเดียวกันกับโปรแกรมรหัสซีซาร์
    }
    public static String encrypt(String p, int a, int b){
        String tmp;
        int m;
        int c;
        String cipher ="";
        for(int i=0;i<p.length();i++){
            tmp = p.substring(i,i+1);
            m = convert_StringtoInteger(tmp);
            c = (a*m+b)%26;
            cipher = cipher+convert_IntegertoString(c);
        }
        return cipher;
    }
        public static String decrypt(String c, int a, int b){
        String tmp;
        int m;
        int p;
```

```
        String plaintext ="";
        int a_inv = mod_inverse(26, a);
        for(int i=0;i<c.length();i++){
            tmp = c.substring(i,i+1);
            m = convert_StringtoInteger(tmp);
            p = a_inv*(m-b)%26;
            if(p < 0){
                p = 26 + p;
            }
            plaintext = plaintext+convert_IntegertoString(p);
        }
        return plaintext;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Plaintext: "); //ตัวพิมพ์ใหญ่
        String p = sc.nextLine();
        System.out.print("Key a: ");
        int a = sc.nextInt();
        System.out.print("Key b: ");
        int b = sc.nextInt();
//////////////////////////Encrypt////////////////////////////////////
        String cipher = encrypt(p, a, b);
        System.out.println("Encrypted message is "+cipher);
//////////////////////////Decrypt////////////////////////////////////
        String plaintext = decrypt(cipher, a, b);
        System.out.println("Decrypted message is "+plaintext);
    }
}
```

5) ขั้นตอนวิธีที่ 5.1 การทดลองหาร (ตรวจสอบความเป็นจำนวนเฉพาะ)

```
package trial_division;
import java.math.BigInteger;
import java.util.Scanner;
public class Trial_division {
    static BigInteger SqRtN(BigInteger N)
    {
      BigInteger G = new BigInteger(String.valueOf((N.shiftRight((N.bitLength() + 1) / 2))));
      BigInteger LastG = null;
      BigInteger One = new BigInteger("1");
      while (true)
      {
        LastG = G;
        G = N.divide(G).add(G).shiftRight(1);
```

```
        int i = G.compareTo(LastG);
        if (i == 0) return G;
        if (i < 0)
        {
         if (LastG.subtract(G).compareTo(One) == 0)
            if (G.multiply(G).compareTo(N) < 0 && LastG.multiply(LastG).compareTo(N) > 0) return G;
        }
        else
        {
          if (G.subtract(LastG).compareTo(One) == 0)
            if (LastG.multiply(LastG).compareTo(N) < 0 && G.multiply(G).compareTo(N) > 0) return LastG;
        }
      }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Number for checking: ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger x = new BigInteger("3");
        BigInteger Zero = new BigInteger("0");
        BigInteger y = n.mod(x);
        BigInteger t = SqRtN(n);
        while((x.max(t).equals(t)) && !(y.equals(Zero))){
            x = x.add(new BigInteger("2"));
            y = n.mod(x);
        }
        if(!(y.equals(Zero))){
            System.out.println(n +" is "+ "a prime number");
        }
        else{
            System.out.println(n +" is "+ "a composite number");
        }
    }
}
```

6) การก่อกำเนิดกุญแจสำหรับขั้นตอนวิธีอาร์เอสเอ

```
package rsa_keygen;
import java.math.BigInteger;
import java.security.SecureRandom;
public class RSA_keyGen {
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        BigInteger p = new BigInteger(32, 4, sr); // p มีขนาด 32 บิต
```

```
        BigInteger q = new BigInteger(32,4,sr); // q มีขนาด 32 บิต
        BigInteger one = new BigInteger("1");
        BigInteger n = p.multiply(q);
        BigInteger euler = p.subtract(one).multiply(q.subtract(one));
        BigInteger e = new BigInteger(5, 1, sr);
        while(!e.gcd(euler).equals(one)){
            e = new BigInteger(5, 1, sr);
        }
        BigInteger d = e.modInverse(euler);
        System.out.println("p = "+p);
        System.out.println("q = "+q);
        System.out.println("n = "+n);
        System.out.println("euler = "+euler);
        System.out.println("e = "+e);
        System.out.println("d = "+d);
    }
}
```

7) การประยุกต์ทฤษฎีเศษเหลือจีนร่วมกับวิทยาการรหัสลับอาร์เอสเอ (การถอดรหัสลับ)

```
package crt_rsa;
import java.math.BigInteger;
import java.util.Scanner;
public class CRT_RSA {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Prime p: ");
        BigInteger p = new BigInteger(sc.nextLine());
        System.out.print("Prime q: ");
        BigInteger q = new BigInteger(sc.nextLine());
        System.out.print("Private key d: ");
        BigInteger n = p.multiply(q);
        BigInteger d = new BigInteger(sc.nextLine());
        System.out.print("Cipher: ");
        BigInteger c = new BigInteger(sc.nextLine());
        BigInteger dp = d.mod(p.subtract(new BigInteger("1")));
        BigInteger dq = d.mod(q.subtract(new BigInteger("1")));
        BigInteger mp = c.modPow(dp, p);
        BigInteger mq = c.modPow(dq, q);
        BigInteger yq = q.modInverse(p);
        BigInteger yp = p.modInverse(q);
        BigInteger m = mp.multiply(yq.multiply(q)).add(mq.multiply(yp.multiply(p))).mod(n);
        System.out.println("Recovered message is "+m);
```

```
    }
}
```

8) การประยุกต์ใช้สมการที่ผู้เขียนนำเสนอใหม่สำหรับการถอดรหัสลับวิทยาการรหัสลับอาร์เอสเอ

```
package javaapplication14;
import java.math.BigInteger;
import java.util.Scanner;
public class JavaApplication14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Prime p: ");
        BigInteger p = new BigInteger(sc.nextLine());
        System.out.print("Prime q: ");
        BigInteger q = new BigInteger(sc.nextLine());
        System.out.print("Private key d: ");
        BigInteger n = p.multiply(q);
        BigInteger d = new BigInteger(sc.nextLine());
        System.out.print("Cipher: ");
        BigInteger c = new BigInteger(sc.nextLine());
        BigInteger c_inv = c.modInverse(n);
        BigInteger euler = p.subtract(new BigInteger("1")).multiply(q.subtract(new BigInteger("1")));
        BigInteger x = euler.subtract(d);
        System.out.println("x = "+x);
        BigInteger m = c_inv.modPow(x, n);
        System.out.println("Recovered message is "+m);
    }
}
```

9) ขั้นตอนวิธีที่ 7.1 การทดลองหาร (แบบปรับค่าลง)

```
package trial_division;
import java.math.BigInteger;
import java.util.Scanner;
public class Trial_division {
    static BigInteger SqRtN(BigInteger N)
    {
   //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Number for checking: ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger x = SqRtN(n);
        if(x.mod(new BigInteger("2")).equals(new BigInteger("0"))){
```

```
            x = x.subtract(new BigInteger("1"));
        }
        BigInteger Zero = new BigInteger("0");
        BigInteger y = n.mod(x);
        BigInteger t = SqRtN(n);
        while((x.max(t).equals(t)) && !(y.equals(Zero))){
            x = x.subtract(new BigInteger("2"));
            y = n.mod(x);
        }
        if(!(y.equals(Zero))){
            System.out.println(n +" is "+ "prime number");
        }
        else{
            System.out.println(n +" is "+ "composite number");
            System.out.println("p = "+x);
            System.out.println("q = "+n.divide(x));
        }
    }
}
```

10) ขั้นตอนวิธีที่ 7.2 โรห์ของโพลลาร์ด

```
package pollard_rho;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Pollard_Rho {
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
        BigInteger n = new BigInteger(sc.nextLine());
        int len = 10000000;
        BigInteger[] m = new BigInteger[len];
        int i = 2;
        m[0] = new BigInteger(5,4,sr);
        m[1] = m[0].pow(2);
        m[2] = m[1].pow(2).add(BigInteger.ONE).mod(n);
        while(n.gcd(m[i].subtract(m[i/2])).equals(BigInteger.ONE)){
            m[i+1] = m[i].pow(2).add(BigInteger.ONE).mod(n);
            m[i+2] = m[i+1].pow(2).add(BigInteger.ONE).mod(n);
            i=i+2;
        }
        BigInteger p = n.gcd(m[i].subtract(m[i/2]));
        BigInteger q = n.divide(p);
```

```
        System.out.println("p = "+p);
        System.out.println("q = "+q);
    }
}
```

11) ขั้นตอนวิธีที่ 7.3 p - 1 ของโพลลาร์ด

```
package pollard_p_1;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Pollard_p_1 {
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
        BigInteger n = new BigInteger(sc.nextLine());
        int len = 10000000;
        BigInteger[] r = new BigInteger[len];
        int i = 2;
        BigInteger b = new BigInteger(5,1,sr);
        r[1] = b.mod(n);
        r[2] = r[1].pow(2).mod(n);
        while(n.gcd(r[i].subtract(BigInteger.ONE)).equals(BigInteger.ONE)){
            i = i+1;
            r[i] = r[i-1].modPow(new BigInteger(i+""), n);
            System.out.println("r["+i+"] = "+r[i]);
        }
        BigInteger p = n.gcd(r[i].subtract(BigInteger.ONE));
        BigInteger q = n.divide(p);
        System.out.println("p = "+p);
        System.out.println("q = "+q);
    }
}
```

12) ขั้นตอนวิธีที่ 7.5 วีแฟกเตอร์

```
package vfactor;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class VFactor {
static BigInteger SqRtN(BigInteger N)
    {
  //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
```

```
    }
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger i = SqRtN(n);
        BigInteger two = new BigInteger("2");
        if(i.mod(two).equals(BigInteger.ZERO)){
            i = i.subtract(BigInteger.ONE);
        }
        BigInteger y = i;
        BigInteger x = i.add(two);
        BigInteger t = x.multiply(y);
        while(!t.equals(n)){
            if(t.max(n).equals(n)){
                x = x.add(two);
            }
            else{
                y = y.subtract(two);
            }
            t = x.multiply(y);
        }
        System.out.println("p = "+x);
        System.out.println("q = "+y);
    }
}
```

13) ขั้นตอนวิธีที่ 7.6 วิธีของแฟร์มาต์

```
package fermat_squareroot;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Fermat_squareroot {
    static BigInteger SqRtN(BigInteger N)
    {
   //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
    }
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger x = SqRtN(n).add(BigInteger.ONE);
```

```
        BigInteger tmp = x.pow(2).subtract(n);
        BigInteger y = SqRtN(tmp);
        BigInteger y_2 = y.pow(2);
        while(!y_2.equals(tmp)){
            x = x.add(BigInteger.ONE);
            tmp = x.pow(2).subtract(n);
            y = SqRtN(tmp);
            y_2 = y.pow(2);
        }
        System.out.println("p = "+x.add(y));
        System.out.println("q = "+x.subtract(y));
    }
}
```

14) ขั้นตอนวิธีที่ 7.7 วิธีของแฟร์มาต์แบบไม่มีการคำนวณรากที่สอง

```
package fermat_nosquareroot;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Fermat_nosquareroot {
    static BigInteger SqRtN(BigInteger N)
    {
   //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
    }
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger two = new BigInteger("2");
        BigInteger four = new BigInteger("4");
        BigInteger u = two.multiply(SqRtN(n).add(BigInteger.ONE));
        BigInteger v = BigInteger.ZERO;
        BigInteger r = u.pow(2).subtract(v.pow(2)).subtract(n.multiply(four));
        while(!r.equals(BigInteger.ZERO)){
            if(r.max(BigInteger.ZERO).equals(r)){
                r = r.subtract(four.multiply(v).add(four));
                v = v.add(two);
            }
            else{
                r = r.add(four.multiply(u).add(four));
                u = u.add(two);
            }
        }
```

```
        System.out.println("p = "+u.add(v).divide(two));
        System.out.println("q = "+u.subtract(v).divide(two));
    }
}
```

15) การประมาณค่าเริ่มต้นของ *u* และ *v* ใหม่สำหรับวิธีของแฟร์มาต์แบบไม่มีการคำนวณรากที่สองโดยเลขสองหลักสุดท้ายของ *n* คือ 97 (หากเลข *m* หลักสุดท้ายของ *n* เป็นค่าอื่นจะต้องปรับขั้นตอนวิธีในโปรแกรมเนื่องจากรูปแบบ *m* ตัวสุดท้ายของ *u* และ *v* จะเกิดการเปลี่ยนแปลง)

```
package estimate_initial_value;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Estimate_initial_value {
    static BigInteger SqRtN(BigInteger N)
    {
   //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
    }
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus (last 2 digits must be only 97): ");
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger two = new BigInteger("2");
        BigInteger u = two.multiply(SqRtN(n).add(BigInteger.ONE));
        int i = u.mod(new BigInteger("100")).intValue();
        int i0 = i%10;
        int i1 = i/10;
        int t = 0;
        if(i1%2 == 0){
            if(i0 > 2){
                t = i - i0+18;
            }
            else{
                t = i - i0 + 2;
            }
        }
        else{
            if(i0 > 8){
                t = i - i0 + 12;
            }
            else{
                t = i - i0 + 8;
            }
```

```
        }
        BigInteger u_i = u.subtract(new BigInteger(i+"")).add(new BigInteger(t+""));
        BigInteger d = u_i.subtract(u);
        BigInteger v_i = two.multiply(SqRtN(SqRtN(d.pow(2).multiply(n))).add(BigInteger.ONE));
        int j = v_i.mod(new BigInteger("100")).intValue();
        int j0 = j%10;
        int j1 = j/10;
        int k = 0;
        if(j1%2 == 0){
            if(i0 > 4){
                k = j - j0+16;
            }
            else{
                k = j - j0 + 4;
            }
        }
        else{
            if(j0 > 6){
                k = j - j0 + 14;
            }
            else{
                k = j - j0 + 6;
            }
        }
        v_i = v_i.subtract(new BigInteger(j+"")).add(new BigInteger(k+""));
        System.out.println("Initial value of u is: "+u_i);
        System.out.println("Initial value of v is: "+v_i);
    }
}
```

16) ขั้นตอนวิธีการแยกตัวประกอบใหม่ที่มีรากฐานมาจากวิธีของแฟร์มาต์

```
package efa;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class EFA {
    static BigInteger SqRtN(BigInteger N)
    {
   //ใช้ code ชุดเดียวกันกับโปรแกรมทดลองหาร (ขั้นตอนวิธีที่ 5.1)
    }
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.print("Modulus: ");
```

```
        BigInteger n = new BigInteger(sc.nextLine());
        BigInteger two = new BigInteger("2");
        BigInteger u = two.multiply(SqRtN(n).add(BigInteger.ONE));
        BigInteger c = new BigInteger("2");
        BigInteger a = c.modInverse(n);
        BigInteger s = c.pow(2).mod(n);
        BigInteger t = a.modPow(n.subtract(u).add(BigInteger.ONE), n);
        BigInteger x = u.divide(two);
        BigInteger tmp = x.pow(2).subtract(n);
        BigInteger y = SqRtN(tmp);
        BigInteger y_2 = y.pow(2);
        if(t.equals(BigInteger.ONE)){
            if(!y_2.equals(tmp)){
                t = two;
            }
        }
        while(!t.equals(BigInteger.ONE)){
            t = t.multiply(s).mod(n);
            x = x.add(BigInteger.ONE);
            if(t.equals(BigInteger.ONE)){
                tmp = x.pow(2).subtract(n);
                y = SqRtN(tmp);
                y_2 = y.pow(2);
                if(!y_2.equals(tmp)){
                      t = two;
                }
            }
        }
        System.out.println("p = "+x.add(y));
        System.out.println("q = "+x.subtract(y));
    }
}
```

17) ขั้นตอนวิธีการคำนวณ point addition สำหรับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

```
package point_addition;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Point_addition {
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.println("Select a, b, p for the equation: y = x^3 + ax + b mod p");
        System.out.print("a: ");
```

```
        BigInteger a = new BigInteger(sc.nextLine());
        System.out.print("b: ");
        BigInteger b = new BigInteger(sc.nextLine());
        System.out.print("p: ");
        BigInteger p = new BigInteger(sc.nextLine());
        System.out.println("Select a point: (x1, y1)");
        System.out.print("x1: ");
        BigInteger xp = new BigInteger(sc.nextLine());
        System.out.print("y1: ");
        BigInteger yp = new BigInteger(sc.nextLine());
        System.out.println("Select a point: (x2, y2)");
        System.out.print("x2: ");
        BigInteger xq = new BigInteger(sc.nextLine());
        System.out.print("y2: ");
        BigInteger yq = new BigInteger(sc.nextLine());
        BigInteger X = xq.subtract(xp).mod(p);
        BigInteger Y = yq.subtract(yp).mod(p);
        if(X.max(BigInteger.ZERO).equals(BigInteger.ZERO)){
            X = p.add(X);
        }
        BigInteger X_inv = X.modInverse(p);
        BigInteger m = Y.multiply(X_inv).mod(p);
        BigInteger xr = m.pow(2).subtract(xq).subtract(xp).mod(p);
        BigInteger yr = m.multiply(xp.subtract(xr)).subtract(yp).mod(p);
        System.out.println("("+xp+","+yp+") + ("+xq+","+yq+") = ("+xr+","+yr+")");
    }
}
```

18) ขั้นตอนวิธีการคำนวณ point doubling สำหรับเส้นโค้งเชิงวงรีเหนือฟิลด์จำนวนเฉพาะ

```
package point_doubling;
import java.math.BigInteger;
import java.security.SecureRandom;
import java.util.Scanner;
public class Point_doubling {
    public static void main(String[] args) {
         SecureRandom sr = new SecureRandom();
        Scanner sc = new Scanner(System.in);
        System.out.println("Select a, b, p for the equation: y = x^3 + ax + b mod p");
        System.out.print("a: ");
        BigInteger a = new BigInteger(sc.nextLine());
        System.out.print("b: ");
        BigInteger b = new BigInteger(sc.nextLine());
        System.out.print("p: ");
        BigInteger p = new BigInteger(sc.nextLine());
```

```
        System.out.println("Select a point: (x1, y1)");
        System.out.print("x1: ");
        BigInteger xp = new BigInteger(sc.nextLine());
        System.out.print("y1: ");
        BigInteger yp = new BigInteger(sc.nextLine());
        BigInteger two = new BigInteger("2");
        BigInteger yp_2 = yp.multiply(two).mod(p);
        BigInteger yp_2_inv = yp_2.modInverse(p);
        BigInteger m = xp.pow(2).multiply(new BigInteger("3")).add(a).multiply(yp_2_inv).mod(p);
        BigInteger xr = m.pow(2).subtract(two.multiply(xp)).mod(p);
        BigInteger yr  = m.multiply(xp.subtract(xr)).subtract(yp).mod(p);
        System.out.println("2("+xp+","+yp+") = ("+xr+","+yr+")");
    }
}
```

ภาคผนวก ข
เฉลยคำถามท้ายบท

บทที่ 1

1. 25
2. 110100010_2
3. A = {2, 4, 6, 8}
4. A = $\{x \in \mathbb{Z} \mid x^2 + 2x - 35 = 0\}$
5. A = {-7, 5}
6. A = $\{x \in \mathbb{Z}^+ \mid x^2 + 2x - 35 = 0\}$
7. 6
8. 7
9. 3
10. 15
11. 7
12. 8
13. 18
14. $x^4 + x^3 + 1$
15. $f(x) = x^7 + x^5 + x + 1$
16. $x^2 + 1$
17. $x^3 + x^2 + 1$

บทที่ 2

1. MLMJ
2. CAT
3. ข้อความต้นฉบับคือ EASY และกุญแจลับคือ 17
4. OSUOGL
5. NBGT
6. ไม่ได้ เนื่องจาก gcd(a, 26) = 13

7. THAI

8. 312 ค่า

9. VCHF

10. DE

11. 010101

12. UAANIDHNJBUISTOTIAHTVRYNRAE

13. ILOVEMYDOG

14. OHIJHUVSYUNARAANEIDTNABTIRT

15. IPLAYFOOTBALLEVERYDAY

16. OUDHNTIANJRAHABUATVNISERYIT

17. CRYPTOGRAPHY

18. ABYCS

บทที่ 3

1. สถาบันมาตรฐานและเทคโนโลยีแห่งสหรัฐอเมริกา (National Institute of Standards and Technology) หรือ NIST

2. 64 บิต

3. 16 รอบ

4. 16 รอบ

5. รอบที่ 16

6. รอบที่ 13

7. 2 บิต

8. 48 บิต

9. K_{0_left} = 1001 1101 0111 1001 0100 1010 1110

   K_{0_right} = 0010 0100 1011 0000 1110 0001 0000

10. 1101 1011 0011 0100 0110 1001 0110

11. 110110 101000 001011 111000 000111 110010 100011 110111

12. 12

13. 11

14. 0101 1110 0010 0011 1100 0100 1001 0000

15. 2^{168}

16. 10100001011011001001000110100010000111100000110001000001011010 11

17. 01001110100001111011000010110010100001000101001100011010010001 01

บทที่ 4

1. 128 บิต

2. 3 ขนาด คือ 128, 192 และ 256 บิต

3. $x^8 + x^4 + x^3 + x + 1$

4. 12 รอบ

5. f3

6. 26914348

7. 207a8391

8.

32	61	6a	15
15	c7	50	d1
05	39	f3	97
3b	2c	80	88

9.

54	27	55	14
83	17	51	3e
1a	26	61	62
49	4f	41	3b

10. 0b

11. 6c

398

12. ee

13.

c2	3d	22	0f
76	ab	71	00
1d	91	16	b0
15	51	e3	fd

14.

26	68	42	87
65	f1	54	91
22	4c	23	11
33	67	28	71

15. 256 บิต

บทที่ 5

1. จำนวนเฉพาะ
2. จำนวนประกอบ
3. 0.9375
4. 131
5. 1198
6. 1198
7. 47
8. 23
9. 804768
10. 25
11. 4624
12. 226380

13. 1

14. 29 เป็นจำนวนเฉพาะ เพราะ $11^{28} \bmod 29 = 1$

    111 เป็นจำนวนประกอบ เพราะ $11^{110} \bmod 111 = 10$

15. ไม่เป็น เพราะ

    7. $4^1 \bmod 7 = 4$
    8. $4^2 \bmod 7 = 2$
    9. $4^3 \bmod 7 = 1$
    10. $4^4 \bmod 7 = 4$
    11. $4^5 \bmod 7 = 2$
    12. $4^6 \bmod 7 = 1$

16. เป็น เพราะ

    1. $7^1 \bmod 11 = 7$
    2. $7^2 \bmod 11 = 5$
    3. $7^3 \bmod 11 = 2$
    4. $7^4 \bmod 11 = 3$
    5. $7^5 \bmod 11 = 10$
    6. $7^6 \bmod 11 = 4$
    7. $7^7 \bmod 11 = 6$
    8. $7^8 \bmod 11 = 9$
    9. $7^9 \bmod 11 = 8$
    10. $7^{10} \bmod 11 = 1$

17. [4; 1, 2, 4]

18. $\frac{14}{3}$

บทที่ 6

1. วิทยาการรหัสลับแบบกุญแจสาธารณะ
2. 2 ค่า
3. กุญแจที่ถูกเปิดเผยได้
4. แลกเปลี่ยนกุญแจลับ
5. ปัญหาวิยุตลอการิทึม
6. 3
7. 707
8. 37
9. 10
10. 10
11. 10
12. {2, 3, 5, 7, 11, 13, 17, 19}
13. เป็น เพราะ $84 = 2^2 \times 3 \times 7$
14. ไม่เป็น เพราะ $85 = 5 \times 17$
15. พบกันตรงครึ่งทาง
16. 2

บทที่ 7

1. โดย รอน ริเวสต์ อาดี ชามีร์ และ เล็น เอเดิลแมน
2. 1024 บิต
3. $n = 34121$ และ $\Phi(n) = 33744$
4. ไม่ได้ เพราะ $\gcd(e, \Phi(n)) = 3$
5. 22007
6. 5 ตัว
7. 1096

8. 11145

9. BCDDD

10. 103 และ 119

11. 7003

12. หากเลือกใช้สมการถอดรหัสลับแบบดั้งเดิมพบว่าเลขยกกำลังมีขนาดที่สูงมาก (d = 21001) หากเปรียบเทียบกับเลขยกกำลังสำหรับสมการถอดรหัสลับจากทฤษฎีบทที่ 7.1 (x = 21762 – 21001 = 671) ซึ่งมีขนาดที่เล็กกว่ามาก ดังนั้นการเลือกใช้สมการถอดรหัสลับจากทฤษฎีบทที่ 7.1 จะมีเหมาะสมกว่า

13. 11 และ 739 โดยขั้นตอนวิธีการทดลองหารแบบปรับค่าขึ้นมีประสิทธิภาพที่สูงกว่าเนื่องจากตัวประกอบค่าหนึ่งมีขนาดเล็กมาก

14. 11

15. ไม่เหมาะสมเนื่องจาก $401399 > \frac{1}{3}n^{\frac{1}{4}}$ ($\frac{1}{3}n^{\frac{1}{4}} \approx 19$)

16. ขั้นตอนวิธีทดลองหารมีประสิทธิภาพสูงสุดเนื่องจากตัวประกอบค่าหนึ่งเป็นจำนวนเฉพาะที่มีขนาดเล็กที่สุด

บทที่ 8

1. 0, 1, 4, 5, 6 และ 9

2. ไม่จำเป็นเพราะ ($x^2 - n$) mod 16 = 18

3. 3

4. จำนวนเต็มคู่เสมอ

5. จำนวนเต็มที่หาร 3 ไม่ลงตัว

6. จำนวนเต็มคู่ที่หาร 3 ไม่ลงตัว มีเลขหลักหน่วยมีค่าเป็น 2 หรือ 8 เท่านั้น

7. เป็นไปไม่ได้ เนื่องจาก 1436 mod 8 = 4 โดยผลลัพธ์ของ u mod 8 ต้องมีค่าเท่ากับ 0

8. $a = a_m a_{m-1} a_{m-2} \cdots a_0$

9. (61, 83), (71, 53), (81, 23), (91, 93), (01, 63), (11, 33), (21, 03), (31, 73), (41, 43)

10. (27, 69), (37, 99), (47, 29), (57, 59), (67, 89), (77, 19), (87, 49), (97, 79), (07, 09)

11. $LSG_2(u)$ = 4, 16, 24, 36, 44, 56, 64, 76, 84 และ 96

$LSG_2(v)$ = 2, 18, 22, 38, 42, 58, 62, 78, 82 และ 98

12. 5236

13. 146

14. 1709, 4007

15. จำนวนรอบการคำนวณที่ลดลงเมื่อเปรียบเทียบกับขั้นตอนวิธีที่ 7.7 และจำนวนรอบการคำนวณหาค่ารากที่สองน้อยมากเมื่อเปรียบเทียบกับขั้นตอนวิธีที่ 7.6 ซึ่งต้องคำนวณทุกรอบ และสามารถนำเทคนิควิธีต่างๆ ที่ใช้สำหรับลดรอบการคำนวณของทั้งสองขั้นตอนวิธีมาประยุกต์ใช้ร่วมกับขั้นตอนวิธีนี้ได้

16. 396395

17. 2543

บทที่ 9

1. นีล โคบลิทซ์ และ วิคเตอร์ มิลเลอร์

2. จำนวนบิตที่ต่ำกว่า

3. 2 วิธี คือ point addition และ point doubling

4. ไม่ได้ เนื่องจาก $2^3 + 27x3^2 \bmod 251 = 0$

5. (3, 7)

6. (5, 19)

7. 31

8. (5, 2)

9. (22, 13)

10. (22, 13)

11. $y^2 + xy = x^3 + x^2 + x$

12. 2039, 3319

13. 643, 1019

14. เลขฐานสองทุกตำแหน่งของ k ที่มีค่าเป็น 1 (เมื่อ $Q = kP$)

15. 2 รอบ เนื่องจาก $11 = 1011_2$

16. $56 < N < 88$ เมื่อ N คือจำนวนจุดบนเส้นโค้งเชิงวงรี

บทที่ 10

1. แมสเสจไดเจสท์

2. มีความปลอดภัยสูง และปลอดการชน

3. เป็นไปไม่ได้ เนื่องจากจำนวนบิตของ h คือ 112 แต่ค่าแฮชที่ได้จาก MD2 ต้องมีขนาด 128 บิต เท่านั้น

4. 160 บิต

5. 512 บิต

6.

$$\underbrace{1000\ 0000\ 0000\ 0000\ 0000\ 0000 \cdots 0000}_{\text{127 ตัว}}$$

7.

$$\underbrace{0000\ 0000\ 0000\ 0000 \cdots 0001\ 0100\ 0000}_{\text{55 ตัว}}$$

8. W(0) = 14576AF1

   W(1) = 23480000

9. 40_{16}

10. DF_{16}

11. 0.72

12. 0.28

13. 4 คน

14. การรักษาความปลอดภัยจะเข้ารหัสข้อมูลด้วยกุญแจสาธารณะ และถอดรหัสข้อความไซเฟอร์ด้วยกุญแจส่วนตัว แต่ลายเซ็นดิจิทัลจะเข้ารหัสข้อมูลด้วยกุญแจส่วนตัว แต่ถอดรหัสข้อมูลด้วยกุญแจสาธารณะ

15. 126

16. ขอบเขตของข้อความต้นฉบับมีขนาดที่ใหญ่กว่าขอบคุณของค่าแฮชจึงเกิดปัญหาการชนกันของค่าแฮชที่เกิดจากข้อความต้นฉบับที่แตกต่างกัน

404

17. 445731

ประวัติผู้เขียน

นายกฤษณพงศ์ สมสุข

Mr. Kritsanapong Somsuk

ตำแหน่งทางวิชาการ

ผู้ช่วยศาสตราจารย์ สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถานที่ทำงาน

สาขาวิชาวิศวกรรมคอมพิวเตอร์ และการสื่อสาร คณะเทคโนโลยี มหาวิทยาลัยราชภัฏอุดรธานี (สามพร้าว)

ประวัติการศึกษา

ปริญญาเอก	ปรัชญาดุษฎีบัณฑิต (ปร.ด.) สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น, 2560
ปริญญาโท	วิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น, 2557
ปริญญาโท	วิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.) สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น, 2553
ปริญญาตรี	วิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น (เกียรตินิยมอันดับ 2), 2551

ประวัติการทำงาน

2553 – 2558	อาจารย์ประจำสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะเทคโนโลยี มหาวิทยาลัยราชภัฏอุดรธานี
2558	ได้รับแต่งตั้งให้ดำรงตำแหน่งผู้ช่วยศาสตราจารย์ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์
2558 – ปัจจุบัน	ผู้รับผิดชอบหลักสูตร และอาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ และการสื่อสาร คณะเทคโนโลยี มหาวิทยาลัยราชภัฏอุดรธานี