การสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดีขึ้น โดยสามารถใช้กุญแจตัวที่สองซ้ำได้

STRONGLY UNFORGEABLE SIGNATURE SCHEME WITH SECURE SECONDARY KEY REPETITION

วัชรี ตันติกิตติพิสุทธิ์

WATCHAREE TANTIKITTIPISUT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2551

KMITL-2009-IT-M-001-003

STRONGLY UNFORGEABLE SIGNATURE SCHEME WITH SECURE SECONDARY KEY REPETITION

WATCHAREE TANTIKITTIPISUT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

KMITL-2009-IT-M-001-003

หัวข้อวิทยานิพนธ์	การสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดีขึ้นโดยสามารถใช้กุญแจตัวที่สองซ้ำได้
นักศึกษา	นางสาววัชรี ตันติกิตติพิสุทธิ์
รหัสนักศึกษา	50066629
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	การจัดการเทคโนโลยีสารสนเทศ
พ.ศ.	2551
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ดร.นล เปรมัษเฐียร

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เสนอวิธีการสร้างลายมือชื่อซึ่งพัฒนามาจากวิธีเข้ารหัสพื้นฐานที่ยังคงความปลอดภัย 2 วิธีคือ โปรโตคอลอาร์เอสเอและโปรโตคอลดิฟฟี-เฮลล์แมน โดยใช้กุญแจสาธารณะ 2 ชุด ทำให้การสร้างลายมือชื่อวิธีนี้มีคุณสมบัติที่สามารถป้องกันการปลอมแปลงได้ดีขึ้น (Strong Unforgeability) มีความปลอดภัยต่อการโจมตีแบบเลือกชุดข้อมูลได้ และสามารถใช้กุญแจชุดที่สองซ้ำได้โดยไม่ส่งผลกระทบต่อความปลอดภัยของลายมือชื่อ ช่วยลดภาระในการจัดการกุญแจชุดที่สองซึ่งการสร้างลายมือชื่อวิธีอื่นจำเป็นต้องใช้กุญแจชุดที่สองในลักษณะใช้ครั้งเดียวเพื่อให้สามารถป้องกันการปลอมแปลงได้ดีขึ้นนี้

Thesis Title	Strongly Unforgeable Signature Scheme with Secure Secondary Key Repetition
Student	Ms. Watcharee Tantikittipisut
Student ID.	50066629
Degree	Master of Science
Programme	Information Technology Management
Year	2008
Thesis Advisor	Dr. Nol Premasathian

ABSTRACT

This thesis introduces a signature scheme that is constructed from two secure primitives, RSA and Diffie-Hellman, and makes use of two public keys. The scheme is strongly unforgeable, secure against chosen-message attack, and the secondary key is securely reusable thus simplifies the key management. Unlike existing schemes, its strong unforgeability does not rely on any one-time key.

กิตติกรรมประกาศ

ขอขอบพระคุณ ผศ.ดร.เขมะฑัต วิภาตะวนิช, ผศ.ดร.จันทร์บูรณ์ สถิตวิริยวงศ์, รศ.ดร.นพพร โชติกกำธร , รศ.ดร.โชติพัชร์ ภรณวลัย และดร.นลเปรมัษเฐียร คณะกรรมการสอบวิทยานิพนธ์ที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณบัณฑิตศึกษา, บัณฑิตวิทยาลัย และห้องสมุดประจำคณะเทคโนโลยีสารสนเทศที่ให้ความช่วยเหลือในเรื่องต่างๆ

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

วัชรี ตันติกิตติพิสุทธิ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

ภาพที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ความก้าวหน้าทางด้านเทคโนโลยีสารสนเทศที่รองรับการใช้งานในรูปแบบต่างๆ มากขึ้น สนับสนุนการเติบโตของยุคข้อมูลข่าวสาร (Information Age) ซึ่งข้อมูลและองค์ความรู้เป็นตัวแปรสำคัญทางสังคมและเศรษฐกิจ ทั้งในระดับบุคคลและองค์กร ด้วยเทคโนโลยีสารสนเทศทำให้ข้อมูลต่างๆ ถูกจัดเก็บอยู่ในรูปแบบข้อมูลอิเล็กทรอนิกส์เพื่อสะดวกแก่การใช้งานไม่ว่าจะเป็นการสร้าง ส่ง รับ เก็บรักษา การเข้าถึง หรือประมวลผล แต่การสื่อสารและทำธุรกรรมใดๆด้วยข้อมูลอิเล็กทรอนิกส์ซึ่งสามารถสร้างและแก้ไขได้ง่ายย่อมทำให้ผู้รับข้อมูลไม่มั่นใจได้ว่าตนได้รับข้อมูลที่ถูกต้องหรือไม่ และข้อมูลดังกล่าวมาจากแหล่งที่เชื่อถือได้หรือไม่ จึงมีการนำลายมือชื่อดิจิทัลมาใช้ซึ่งมีลักษณะการใช้งานเทียบได้กับการลงลายมือชื่อในระบบเอกสารทั่วไป

วัตถุประสงค์การใช้งานลายมือชื่อดิจิทัลหรือเรียกโดยย่อว่าลายมือชื่อคือ การสร้างความมั่นใจให้กับผู้รับข้อมูลในสภาวะที่ช่องทางการสื่อสารไม่มีความปลอดภัย เช่น การสื่อสารผ่านช่องทางสาธารณะอย่างอินเตอร์เน็ต ซึ่งมีโอกาสที่ผู้ไม่เกี่ยวข้องรวมทั้งผู้เป็นปฏิปักษ์ (Adversary) จะสามารถล่วงรู้ข้อมูลดังกล่าว ลายมือชื่อที่ดีจะป้องกันไม่ให้ผู้เป็นปฏิปักษ์สามารถใช้ประโยชน์จากข้อมูลที่ได้ไปในการปลอมแปลงลายมือชื่อ

ปัญหาการปลอมแปลงลายมือชื่อแบ่งได้ 2 ลักษณะ คือ

(1) ผู้เป็นปฏิปักษ์นำลายมือชื่อที่ถูกต้องมาใช้ซ้ำด้วยการเปลี่ยนแปลงข้อมูลที่เจ้าของกุญแจลับทำการสร้างลายมือชื่อกำกับไว้ ผู้รับจะยังสามารถถอดรหัสลายมือชื่อและตรวจสอบได้เสมือนว่าข้อมูลที่ถูกแก้ไขเป็นข้อมูลที่ถูกต้องโดยยังถูกกำกับไว้ด้วยลายมือชื่อเดิม

(2) ผู้เป็นปฏิปักษ์ทำการสร้างลายมือชื่อใหม่ที่ถูกต้องให้กับข้อมูลใดๆ ที่ต้องการได้

ทั้งนี้ปัญหาการปลอมแปลงลายมือชื่อขึ้นอยู่กับความแข็งแกร่งของวิธีเข้ารหัสเพื่อสร้างลายมือชื่อ หรือเรียกว่าการสร้างลายมือชื่อ ซึ่งพบว่ามีวิธีการสร้างลายมือชื่อจำนวนมากที่ได้รับการวิเคราะห์แล้วว่าไม่มีความปลอดภัยหากผู้เป็นปฏิปักษ์สามารถล่วงรู้ชุดข้อมูล (ข้อความ, ลายมือชื่อ) ได้มากพอ มีความเป็นไปได้ที่ความลับในการสร้างลายมือชื่อนั้นจะถูกเปิดเผย เป็นเหตุให้ลายมือชื่อถูกปลอมแปลงและหมดความน่าเชื่อถือ ดังนั้นจึงมีการพัฒนาการสร้างลายมือชื่อที่มีความแข็งแกร่งสามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้นเพื่อลดข้อบกพร่องนี้

1.2 วัตถุประสงค์ของการศึกษา

วัตถุประสงค์ของวิทยานิพนธ์ฉบับนี้เพื่อนำเสนอขั้นตอนการออกแบบ, พัฒนาและวิเคราะห์การสร้างลายมือชื่อที่มีคุณสมบัติป้องกันการปลอมแปลงได้ดีขึ้นจากวิธีเข้ารหัสพื้นฐานที่ยังคงความแข็งแกร่ง 2 วิธี โดยเลือกใช้กุญแจอสมมาตร 2 ชุดสำหรับการเข้ารหัส 2 ชั้น(Two-tier encryption) เพื่อเพิ่มความสามารถป้องกันการโจมตีแบบเลือกข้อมูลได้ (Chosen-message Attack) และต้องสามารถใช้กุญแจชุดที่สองซ้ำได้โดยยังคงความปลอดภัยของการเข้ารหัส

1.3 สมมติฐานของการศึกษา

งานวิจัยฉบับนี้นำเสนอการสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดีขึ้น ดังนั้นลายมือชื่อที่สร้างขึ้นควรมีคุณสมบัติที่สามารถป้องกันการโจมตีแบบเลือกข้อมูลได้ เพื่อป้องกันไม่ให้ผู้อื่นสามารถใช้ประโยชน์จากชุดข้อมูล (ข้อความ, ลายมือชื่อ) ไปในการเปิดเผยความลับของการสร้างลายมือชื่อ แล้วทำการปลอมแปลงลายมือชื่อได้

1.4 ทฤษฎีและแนวคิดที่ใช้ในการวิจัย

การสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดีขึ้นที่นำเสนอในวิทยานิพนธ์ฉบับนี้ได้พัฒนามาจากแนวคิดของการสร้างวิธีเข้ารหัสใหม่จากวิธีเข้ารหัสพื้นฐานที่มีอยู่เดิมซึ่งเป็นวิธีที่ยังคงแข็งแกร่ง 2 วิธี และใช้แนวคิดการเข้ารหัส 2 ชั้น การนำวิธีเข้ารหัสพื้นฐานที่แตกต่างกัน 2 วิธีมาใช้ร่วมกันจำเป็นต้องศึกษาถึงจุดอ่อนและจุดแข็งของแต่ละวิธีเพื่อใช้ในการออกแบบจัดวางโครงสร้างวิธีเข้ารหัสที่เหมาะสม ซึ่งจะสามารถลดจุดอ่อนที่มีอยู่และได้วิธีการใหม่ที่แข็งแกร่งขึ้น ส่วนการเข้ารหัส 2 ชั้นด้วยกุญแจอสมมาตร 2 ชุดจะทำให้การปลอมแปลงลายมือชื่อทำได้ยากยิ่งขึ้น สนับสนุนให้เกิดคุณสมบัติที่สามารถป้องกันการปลอมแปลงได้ดีขึ้นของการสร้างลายมือชื่อ

1.5 ขอบเขตการวิจัย

การพัฒนาเพื่อสร้างลายมือชื่อในวิทยานิพนธ์ฉบับนี้มีขอบเขตการวิจัยดังนี้

1. การใช้งานครอบคลุมข้อมูลที่เป็น binary string ทุกประเภท
2. โครงสร้างของวิธีใช้การเข้ารหัสแบบกุญแจอสมมาตร ซึ่งประยุกต์มาจากวิธีเข้ารหัสพื้นฐานที่เลือกใช้
3. การออกแบบและพัฒนาการสร้างลายมือชื่อพิจารณาอยู่ภายใต้โมเดลมาตรฐาน
4. ใช้โมเดลมาตรฐานในการวิเคราะห์และพิสูจน์ความปลอดภัย

1.6 ขั้นตอนของการศึกษา

แผนการดำเนินงานวิจัยมีดังนี้

1. กำหนดหัวข้อ เป้าหมาย จุดประสงค์ และขอบเขตของการทำวิทยานิพนธ์
2. ศึกษาทฤษฎีและหลักการพื้นฐานที่ใช้ในการวิจัย
3. ศึกษาเทคนิคต่างๆ ที่มีอยู่ถึงแนวคิด หลักการ ข้อดี และข้อบกพร่องของแต่ละเทคนิค
4. ออกแบบและพัฒนาวิธีการสร้างลายมือชื่อ
5. พิสูจน์ความปลอดภัยของการสร้างลายมือชื่อที่พัฒนาขึ้น
6. จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 ทฤษฎีพื้นฐานและแนวคิดที่ใช้ในการวิจัย

บทที่ 3 การออกแบบและพัฒนาวิธีการสร้างลายมือชื่อ

บทที่ 4 การวิเคราะห์และพิสูจน์ประสิทธิภาพด้านความปลอดภัย

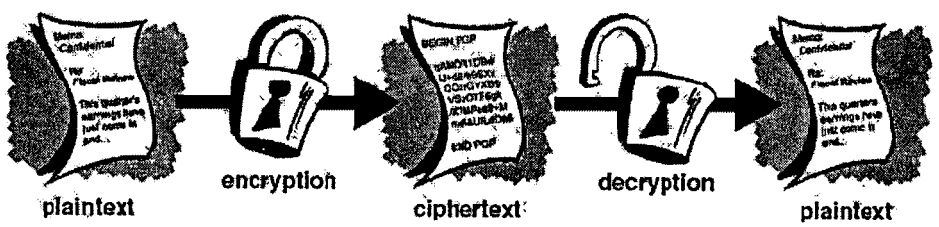
บทที่ 5 บทสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2
ทฤษฎีพื้นฐานและแนวคิดที่ใช้ในการวิจัย

ในบทนี้จะกล่าวถึงความรู้เบื้องต้นของวิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะ (Public-Key Cryptography) ลายมือชื่อและการสร้างลายมือชื่อ ทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้อง รวมทั้งงานวิจัยการสร้างลายมือชื่อวิธีอื่นที่ใช้เป็นแนวคิดของการวิจัยนี้

2.1 วิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะ

วิทยาการเข้ารหัสลับ (Cryptography) เป็นกระบวนการที่อาศัยหลักการคำนวณทางคณิตศาสตร์มาใช้ในการแปลงข้อความเพื่อซ่อนเนื้อหาของข้อความที่ต้องการสื่อสารอย่างเป็นความลับ



ภาพที่ 2.1 การเข้ารหัส และการถอดรหัส

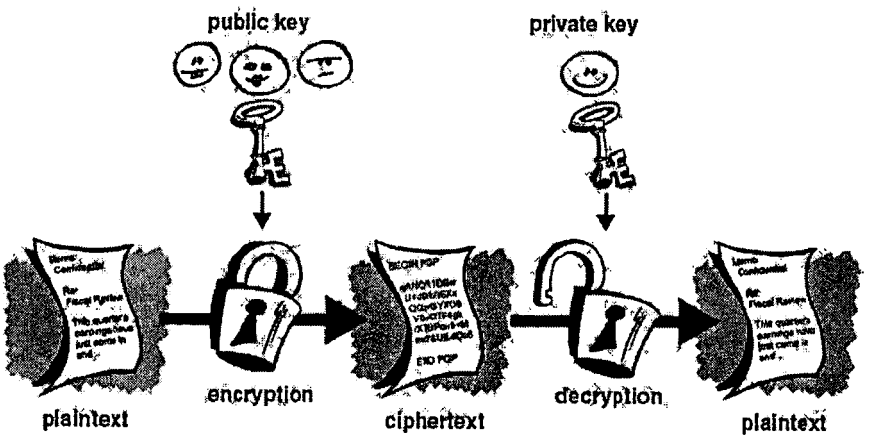
วิทยาการเข้ารหัสลับจะประกอบไปด้วย 2 กระบวนการคือ การเข้ารหัส (Encryption) และการถอดรหัส (Decryption) การเข้ารหัสเป็นการแปลงข้อความจากเดิมซึ่งเป็นข้อมูลที่มีความหมายไปเป็นข้อมูลที่ไม่มีความหมาย ส่วนการถอดรหัสจะเป็นการแปลงข้อความให้กลับมาเป็นข้อมูลที่มีความหมายดังเดิม ข้อความก่อนเข้ารหัสจะถูกเรียกว่า ข้อความปกติ (Plaintext) และเรียกผลลัพธ์ที่ได้จากการเข้ารหัสว่า ข้อความเข้ารหัส (Ciphertext)

ในยุคก่อนปี 1980 วัตถุประสงค์การใช้งานวิทยาการเข้ารหัสลับคือใช้เพื่อการสื่อสารอย่างเป็นความลับ ซึ่งเรียกวิทยาการเข้ารหัสลับในยุคนั้นว่าวิทยาการเข้ารหัสลับยุคเก่า (Classical Cryptography) โดยความลับของวิธีเข้ารหัสอยู่ที่ขั้นตอนในการเข้ารหัส การออกแบบขั้นตอนเข้ารหัสเพื่อให้ได้วิธีเข้ารหัสที่มีความปลอดภัยจึงเป็นสิ่งสำคัญ ในการใช้งานผู้ใช้แต่ละกลุ่มจำเป็นต้องสร้างวิธีเข้ารหัสสำหรับใช้เฉพาะภายในกลุ่มและสมาชิกทุกคนต้องรักษาขั้นตอนที่ใช้ในการเข้ารหัสไว้เป็นความลับอย่างเข้มงวด การเปลี่ยนแปลงสมาชิกภายในกลุ่มหมายถึงการเปลี่ยนแปลงวิธีเข้ารหัสใหม่ทุกครั้ง ส่งผลถึงความยุ่งยากในการจัดการข้อมูลวิธีเข้ารหัสเพื่อให้

สมาชิกภายในกลุ่มทุกคนสามารถใช้วิธีเข้ารหัสที่ถูกต้องได้อย่างเหมาะสม ด้วยข้อจำกัดซึ่งผูกติดอยู่กับวิธีเข้ารหัสทำให้ไม่สะดวกต่อการใช้งานและมีความน่าเชื่อถือด้านความปลอดภัยต่ำ

ในยุคหลังปี 1980 ได้มีการนำระบบคำนวณมาใช้กับวิทยาการเข้ารหัสลับ เรียกวิทยาการเข้ารหัสลับยุคนี้ว่าวิทยาการเข้ารหัสลับยุคใหม่ (Modern Cryptography) โดยมีการกำหนดใช้กุญแจเป็นข้อมูลรับเข้าอีกหนึ่งข้อมูลสำหรับกระบวนการเข้ารหัสและถอดรหัส ซึ่งความลับของการเข้ารหัสจะอยู่ที่กุญแจ ทำให้ผู้ใช้สามารถเปิดเผยวิธีเข้ารหัสลับที่ใช้และสามารถใช้วิธีเข้ารหัสร่วมกันได้อย่างปลอดภัย จากรูปแบบการใช้งานที่เปลี่ยนไปส่งผลให้วัตถุประสงค์ในการใช้งานเปลี่ยนแปลงไปด้วย จากเดิมเป็นเพียงเพื่อการสื่อสารที่เป็นความลับกลายมาเป็นส่วนหนึ่งในการสร้างระบบที่มีความปลอดภัย สนับสนุนให้วัตถุประสงค์และรูปแบบการใช้งานมีความหลากหลายมากขึ้น

วิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะ หรือที่เรียกว่าวิทยาการเข้ารหัสลับแบบอสมมาตร (Asymmetric Cryptography) เนื่องมาจากกุญแจที่ใช้ในการเข้ารหัสแตกต่างจากกุญแจสำหรับการถอดรหัส แต่กุญแจทั้งสองมีความเชื่อมโยงกันในการคำนวณตามขั้นตอนเข้ารหัสและถอดรหัส โดยกุญแจสาธารณะ (Public Key) ใช้ในการเข้ารหัส จะถูกเปิดเผยให้สมาชิกทุกคนทราบเพื่อใช้สำหรับสื่อสารกับผู้รับซึ่งจะใช้กุญแจลับ (Private Key) ในการถอดรหัส กุญแจลับจะรู้เฉพาะเจ้าของกุญแจเท่านั้นจึงมีเพียงเจ้าของกุญแจผู้เดียวเท่านั้นที่สามารถถอดรหัสได้



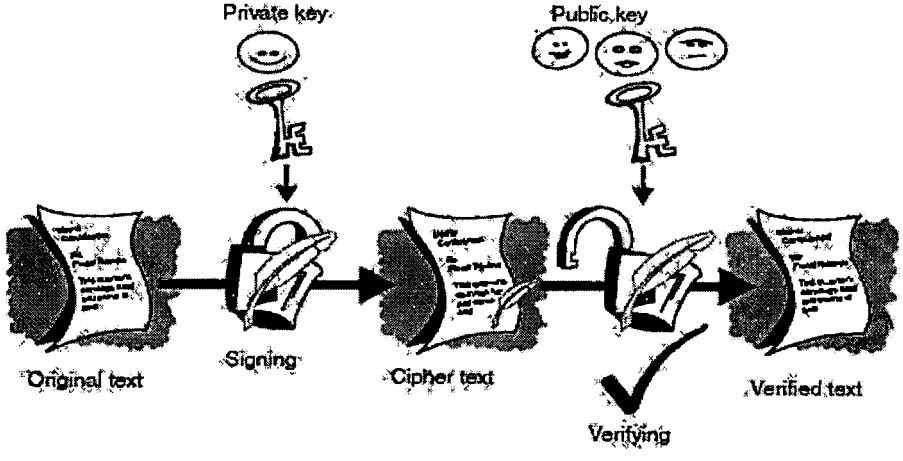
ภาพที่ 2.2 วิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะ

นอกจากนี้ยังสามารถเพิ่มประโยชน์ในการใช้งานวิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะนี้ด้วยการสลับหน้าที่ของกุญแจลับและกุญแจสาธารณะ เพื่อใช้ในการส่งสารจากเจ้าของกุญแจลับไปยังสมาชิกผู้ที่รู้กุญแจสาธารณะได้ โดยเจ้าของกุญแจลับทำการเข้ารหัสข้อความที่ต้องการจะส่งด้วยกุญแจลับ แล้วสมาชิกจะใช้กุญแจสาธารณะในการถอดรหัส ด้วยเหตุผลที่มีเพียง

เจ้าของกุญแจลับเท่านั้นที่สามารถเข้ารหัสได้ สมาชิกจึงสามารถใช้การถอดรหัสในการตรวจสอบได้ว่าผู้ส่งข้อความคือใคร ตัวอย่างการใช้งานรูปแบบนี้ได้แก่การสร้างลายมือชื่อ

2.2 ลายมือชื่อและการสร้างลายมือชื่อ

การสร้างลายมือชื่อเป็นการประยุกต์ใช้วิทยาการเข้ารหัสลับแบบใช้กุญแจสาธารณะเพื่อประโยชน์ในการรับรองเอกสาร ผลลัพธ์ที่ได้จากการเข้ารหัสจะถูกเรียกว่าลายมือชื่อซึ่งใช้เพื่อช่วยยืนยันให้ผู้รับสามารถมั่นใจถึงแหล่งที่มาของข้อความและใช้ในการตรวจสอบความถูกต้องของข้อความที่ได้รับ



ภาพที่ 2.3 ลายมือชื่อดิจิทัล

การจัดการกุญแจทำได้ง่าย เพราะสามารถใช้กุญแจเพียงชุดเดียวกับผู้รับจำนวนมาก โดยเจ้าของลายมือชื่อเพียงเก็บกุญแจลับไว้ แล้วเปิดเผยกุญแจสาธารณะให้สมาชิกรู้ เมื่อต้องการส่งข้อความเจ้าของลายมือชื่อจะใช้กุญแจลับในการเข้ารหัสข้อความที่ต้องการส่งเพื่อสร้างลายมือชื่อ แล้วแนบลายมือชื่อกับข้อความที่ต้องการส่งเป็นชุดข้อมูล (ข้อความ, ลายมือชื่อ) ส่งไปยังสมาชิกผู้รับที่มีกุญแจสาธารณะของชุดกุญแจเดียวกันจะสามารถถอดรหัสลายมือชื่อเพื่อพิสูจน์ลายมือชื่อและตรวจสอบความถูกต้องตรงกันของข้อความที่ได้รับกับข้อมูลที่ถูกกำกับไว้ด้วยลายมือชื่อ

คุณสมบัติสำคัญของลายมือชื่อที่ทำให้การใช้งานเป็นที่ยอมรับ คือ

1. เป็นการระบุตัวผู้ส่ง (Authentication) เพราะชุดกุญแจที่ใช้ในแต่บุคคลจะมีความจำเพาะไม่ซ้ำกัน และผู้ที่สามารถเข้ารหัสข้อความเพื่อสร้างลายมือชื่อได้คือผู้ที่มีกุญแจลับเท่านั้น ดังนั้นหากผู้รับสามารถถอดรหัสลายมือชื่อได้ก็แสดงว่าลายมือชื่อนั้นถูกต้องและทราบได้ว่าผู้ส่งข้อความคือใคร

2. ไม่สามารถปลอมแปลงได้ (Unforgeable) เพราะลายมือชื่อได้มาจากการเข้ารหัสด้วยกุญแจลับของเจ้าของลายมือชื่อ ซึ่งมีเพียงเจ้าของกุญแจผู้เดียวเท่านั้นที่ทราบกุญแจลับ ดังนั้นผู้อื่นย่อมไม่สามารถปลอมแปลงลายมือชื่อเพื่อใช้รับรองเอกสารใดๆได้

3. ไม่สามารถใช้ซ้ำได้ (Not reusable) เพราะลายมือชื่อเป็นผลลัพธ์ที่ได้จากการเข้ารหัสข้อความที่ต้องการรับรอง ข้อความที่ต่างกันจะให้ผลลัพธ์เป็นลายมือชื่อที่ต่างกัน ดังนั้นจึงไม่สามารถนำลายมือชื่อที่ใช้รับรองข้อความหนึ่งแล้วไปใช้ซ้ำเพื่อรับรองข้อความอื่นได้

4. สามารถใช้ในการตรวจสอบความถูกต้องสมบูรณ์ (Integrity) ของข้อความที่ได้รับ เพราะลายมือชื่อได้จากการเข้ารหัสข้อความที่ต้องการรับรอง ดังนั้นผลลัพธ์ที่ได้จากถอดรหัสลายมือชื่อย่อมถูกต้องตรงกันกับข้อความที่รับรองโดยสมบูรณ์

5. ผู้เป็นเจ้าของกุญแจลับไม่สามารถปฏิเสธ (Non-repudiate) ความรับผิดชอบต่อลายมือชื่อที่สร้างขึ้นได้ เพราะผู้ที่รู้กุญแจลับมีเพียงเจ้าของลายมือชื่อเท่านั้น ดังนั้นลายมือชื่อซึ่งเป็นผลลัพธ์จากการเข้ารหัสด้วยกุญแจลับจึงเป็นความรับผิดชอบของเจ้าของกุญแจลับนั้น

2.2.1 การสร้างลายมือชื่อ

มีกระบวนการที่เกี่ยวข้อง 3 ขั้นตอน ดังนี้

a. ขั้นตอนสร้างกุญแจ (Key-Generation Algorithm: *KeyGen*) เป็นกระบวนการสร้างชุดกุญแจสาธารณะจากค่าพารามิเตอร์ที่มีความปลอดภัยขนาด n บิต ผลลัพธ์ที่ได้จะเป็นชุดกุญแจซึ่งประกอบด้วยกุญแจลับและกุญแจสาธารณะ (pk, sk) ใช้สำหรับการสร้างลายมือชื่อ และการพิสูจน์ลายมือชื่อตามลำดับ เขียนแสดงกระบวนการได้ดังนี้

$$(pk, sk) \leftarrow KeyGen(1^n)$$

b. ขั้นตอนสร้างลายมือชื่อ (Signing Algorithm: *Sign*) เป็นกระบวนการสร้างลายมือชื่อ (s) โดยการเข้ารหัสข้อความ $(m \in \{0, 1\}^*)$ ที่ต้องการรับรองด้วยกุญแจลับ เขียนแสดงกระบวนการได้ดังนี้

$$s \leftarrow Sign_{sk}(m)$$

c. ขั้นตอนพิสูจน์ (Verification Algorithm: *Vrfy*) เป็นกระบวนการตรวจพิสูจน์ความถูกต้องของลายมือชื่อ โดยการถอดรหัสลายมือชื่อ s ด้วยกุญแจสาธารณะ แล้วพิสูจน์ผลลัพธ์ที่ได้กับข้อความ m ที่ลายมือชื่อนั้นกำกับไว้ ผลลัพธ์ที่ได้จากการพิสูจน์จะเป็นบิต 1 หากลายมือชื่อที่ตรวจพิสูจน์สามารถถอดรหัสและผลลัพธ์ที่ได้ถูกต้องตรงกันกับข้อความที่รับรองไว้ แต่จะเป็นบิต 0 หากลายมือชื่อนั้นไม่ใช่ลายมือชื่อที่ถูกต้อง เขียนแสดงกระบวนการได้ดังนี้

$$Decision \leftarrow Vrfy_{pk}(m, s)$$

สำหรับลายมือชื่อที่ถูกต้อง เมื่อนำขั้นตอนกาสร้างลายมือชื่อและขั้นตอนการพิสูจน์มาเขียนรวมกันจะได้สมการดังนี้

$$Vrfy_{pk}(m, Sign_{sk}(m)) = 1$$

2.2.2 ความปลอดภัยของลายมือชื่อ

พิจารณาจากความปลอดภัยต่อการโจมตีแบบเลือกชุดข้อมูลได้ (Security against Chosen-Message Attack) สามารถแบ่งคุณสมบัติความปลอดภัยของลายมือชื่อออกได้เป็น 2 ลักษณะ คือ

[1]. ความปลอดภัยแบบที่สามารถป้องกันการปลอมแปลงได้ (Existential Unforgeability) ลายมือชื่อที่มีความปลอดภัยลักษณะนี้จะสามารถป้องกันไม่ให้ผู้เป็นปฏิปักษ์ทำการปลอมแปลงลายมือชื่อเพื่อรับรองข้อความใหม่ได้ โดยสามารถพิสูจน์ความถูกต้องของลายมือชื่อที่ใช้ได้รายละเอียดการพิสูจน์ความปลอดภัยมีดังนี้

a. <u>ขั้นตอนสร้างกุญแจ</u> ผู้พิสูจน์ใช้อัลกอริทึมของการสร้างกุญแจเพื่อสร้างชุดกุญแจอสมมาตร ผู้พิสูจน์จะเก็บกุญแจลับไว้เป็นความลับ แล้วเปิดเผยกุญแจสาธารณะให้ผู้เป็นปฏิปักษ์ทราบ
b. <u>ขั้นตอนสร้างลายมือชื่อ</u> ผู้เป็นปฏิปักษ์สามารถกำหนดข้อความที่ต้องการให้ผู้พิสูจน์ทำการสร้างลายมือชื่อให้จำนวนทั้งหมด q ข้อความ ($M = \{m_1, m_2, \ldots, m_q\}$) ผู้พิสูจน์จะทำการสร้างลายมือชื่อให้กับข้อความเหล่านั้นทั้งหมด q ลายมือชื่อ ($S = \{s_1, s_2, \ldots, s_q\}$) แล้วส่งชุดข้อมูล ($m_i, s_i$) ทั้งหมดคืนให้กับผู้เป็นปฏิปักษ์
c. <u>ผลลัพธ์</u> ผู้เป็นปฏิปักษ์จะใช้ประโยชน์จากชุดข้อมูล (m_i, s_i) ที่ได้รับจากผู้พิสูจน์ในการปลอมแปลงลายมือชื่อ ซึ่งผู้เป็นปฏิปักษ์จะชนะการพิสูจน์นี้หากผู้เป็นปฏิปักษ์สามารถสร้างชุดข้อมูล (m, s_i) โดยที่ s_i สามารถพิสูจน์ได้ว่าเป็นลายมือชื่อที่ถูกต้องของผู้พิสูจน์ ($s_i \in S$) แต่ m เป็นข้อความใหม่ที่ผู้พิสูจน์ยังไม่เคยสร้างลายมือชื่อกำกับไว้ ($m \neq m_i$)

สามารถกำหนดนิยามสำหรับการสร้างลายมือชื่อที่ยังสามารถป้องกันการปลอมแปลงได้ดังนี้ มีความเป็นไปได้น้อยมาก (ε) ที่ผู้เป็นปฏิปักษ์จะสามารถใช้การโจมตีแบบเลือกชุดข้อมูลจำนวน q ชุด เพื่อทำการปลอมแปลงลายมือชื่อ (m, s_i) ได้ภายในเวลา t ที่กำหนดเพื่อที่จะชนะการพิสูจน์ข้างต้น

[2]. ความปลอดภัยแบบสามารถป้องกันการปลอมแปลงได้ดีขึ้น (Strongly Unforgeability) ลายมือชื่อที่มีความปลอดภัยลักษณะนี้จะต้องมีความปลอดภัยแบบที่สามารถ

ป้องกันการปลอมแปลงได้เป็นพื้นฐาน และต้องมีความปลอดภัยเพิ่มเติมจากการปลอมแปลงด้วยวิธีแก้ไขลายมือชื่อที่กำกับข้อความชุดเดียวกันได้ รายละเอียดการพิสูจน์ความปลอดภัยมีดังนี้

a. ขั้นตอนสร้างกุญแจ ผู้พิสูจน์ใช้อัลกอริทึมของการสร้างกุญแจเพื่อสร้างชุดกุญแจอสมมาตร ผู้พิสูจน์จะเก็บกุญแจลับไว้เป็นความลับ แล้วเปิดเผยกุญแจสาธารณะให้ผู้เป็นปฏิปักษ์ทราบ

b. ขั้นตอนสร้างลายมือชื่อ ผู้เป็นปฏิปักษ์สามารถกำหนดข้อความที่ต้องการให้ผู้พิสูจน์ทำการสร้างลายมือชื่อให้จำนวนทั้งหมด q ข้อความ ($M = \{m_1, m_2, \ldots, m_q\}$) ผู้พิสูจน์จะทำการสร้างลายมือชื่อให้กับข้อความเหล่านั้นทั้งหมด q ลายมือชื่อ ($S = \{s_1, s_2, \ldots, s_q\}$) แล้วส่งชุดข้อมูล (m_i, s_i) ทั้งหมดคืนให้กับผู้เป็นปฏิปักษ์

c. ผลลัพธ์ ผู้เป็นปฏิปักษ์จะใช้ประโยชน์จากชุดข้อมูล (m_i, s_i) ที่ได้รับจากผู้พิสูจน์ในการปลอมแปลงลายมือชื่อ ซึ่งผู้เป็นปฏิปักษ์จะชนะการพิสูจน์นี้หากผู้เป็นปฏิปักษ์สามารถสร้างชุดข้อมูล (m, s) โดยที่ (m, s) ไม่ตรงกับชุดข้อมูล (m_i, s_i) ใดๆ ที่ผู้พิสูจน์ได้สร้างลายมือชื่อกำกับไว้ โดย s เป็นลายมือชื่อใหม่ ($s \neq s_i$) ที่ผู้เป็นปฏิปักษ์ได้สร้างขึ้นเพื่อกำกับข้อความ m, โดยที่ลายมือชื่อนั้นสามารถถอดรหัสเพื่อพิสูจน์ความถูกต้องได้

สามารถกำหนดนิยามสำหรับการสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้นดังนี้ มีความเป็นไปได้น้อยมาก (ε) ที่ผู้เป็นปฏิปักษ์จะสามารถใช้การโจมตีแบบเลือกชุดข้อมูลจำนวน q ชุด เพื่อทำการปลอมแปลงลายมือชื่อ (m_i, s) ได้ภายในเวลา t ที่กำหนดเพื่อที่จะชนะการพิสูจน์ข้างต้นได้

ข้อแตกต่างระหว่างความปลอดภัยแบบที่สามารถป้องกันการปลอมแปลงได้กับแบบสามารถป้องกันการปลอมแปลงได้ดีขึ้นคือ กรณีที่ใช้เกณฑ์ในการตัดสินว่าปลอมแปลงได้สำเร็จ ซึ่งความปลอดภัยแบบสามารถป้องกันการปลอมแปลงได้ดีขึ้นจะมีกรณีที่เป็นไปได้สำหรับการปลอมแปลงที่หลากหลายมากกว่า ความปลอดภัยแบบที่สามารถป้องกันการปลอมแปลงได้จะระบุการปลอมแปลงไว้เพียงกรณีที่ผู้เป็นปฏิปักษ์ต้องสามารถสร้างลายมือชื่อที่ถูกต้องให้กับข้อความใหม่ที่ยังไม่เคยถูกรับรองด้วยลายมือชื่อใดมาก่อนได้เท่านั้น ส่วนการปลอมแปลงที่ใช้เป็นเกณฑ์พิจารณาความปลอดภัยแบบสามารถป้องกันการปลอมแปลงได้ดีขึ้นจะรวมไปถึงการที่ผู้เป็นปฏิปักษ์สามารถสร้างลายมือชื่อใหม่ที่ถูกต้องให้กับข้อความเก่าได้ และกรณีที่สามารถหาข้อความใหม่ที่แตกต่างจากข้อความเดิมแต่สามารถใช้ลายมือชื่อที่ถูกต้องร่วมกันได้

ดังนั้นคุณสมบัติสามารถป้องกันการปลอมแปลงได้ดีขึ้นจึงเป็นคุณสมบัติที่แข็งแกร่งกว่าคุณสมบัติแบบสามารถป้องกันการปลอมแปลงได้ เพราะมีประสิทธิภาพในการป้องกันการปลอมแปลงในกรณีที่หลากหลายมากกว่า

2.3 คณิตศาสตร์ที่ใช้เป็นพื้นฐานของงานวิจัย

เนื่องจากวิทยาการเข้ารหัสลับมีแนวคิดพื้นฐานมาจากการคำนวณ สมมติฐานทางคณิตศาสตร์จึงมีความสำคัญต่อความซับซ้อนของขั้นตอนและส่งผลต่อเวลาที่ต้องใช้ในการคำนวณ ซึ่งสมมติฐานที่เกี่ยวข้องกับการวิจัยมีดังนี้

2.3.1 ทฤษฎีจำนวน (Number Theory)

a. จำนวนเฉพาะ (Prime Number) คือจำนวนเต็มบวกที่มีค่ามากกว่า 1 และมีตัวประกอบร่วมเพียง 1 และตัวมันเองเท่านั้น ไม่มีจำนวนอื่นสามารถหารได้ลงตัว เช่น 227, 1021, 3449 เป็นต้น

b. คณิตมอดุลาร์ (Modular Arithmetic) เป็นระบบเลขคณิตที่มีรากฐานมาจากระบบจำนวนเต็ม เป็นการหาค่าสมภาค (Congruence) จากสมการ $a = b + kn$ ซึ่ง k แทนจำนวนเต็มใดๆ ค่าสมภาคที่ได้จะเขียนแทนด้วย $a \equiv b \pmod{n}$ โดยมี n เป็นค่ามอดุลัส (Modulus) หากกำหนดให้ a เป็นจำนวนเต็มบวกใดๆ แล้ว b จะมีค่าอยู่ในช่วง 0 ถึง n-1 ซึ่งเป็นเศษ (Residue) เหลือจากการหาร a ด้วย n หรือเรียกได้ว่า a สมภาคกับ b จากการมอดุโล (Modulo) ด้วย n ตัวอย่างเช่น $23 \equiv 11 \pmod{12}$ หรือ $23 \bmod 12 = 11 \bmod 12$ ในความหมายเดียวกัน

c. ส่วนตกค้างกำลังสอง (Quadratic Residue) เป็นการหาค่าสมภาคจากสมการ $x^2 \equiv a \pmod{p}$ สำหรับบางค่า x ซึ่ง a จะมีค่าอยู่ในช่วง 1 ถึง p-1 ตัวอย่างเช่น

กำหนดให้ p = 7

$$1^2 = 1 \equiv 1 \pmod{7}$$
$$2^2 = 4 \equiv 4 \pmod{7}$$
$$3^2 = 9 \equiv 2 \pmod{7}$$
$$4^2 = 16 \equiv 2 \pmod{7}$$
$$5^2 = 25 \equiv 4 \pmod{7}$$
$$6^2 = 36 \equiv 1 \pmod{7}$$

ส่วนตกค้างกำลังสอง ได้แก่ 1, 2 และ 4

d. กลุ่ม (Groups) กำหนดให้ $\mathbb{G}$ เป็นชุดของข้อมูล และมี การดำเนินการทวิภาค (Binary Operation) ระหว่างสมาชิกของชุดข้อมูล เขียนแทนด้วยสัญลักษณ์ $\circ$ หากกำหนดให้ g และ h เป็นสมาชิกของชุดข้อมูล ชุดข้อมูลนี้จะถือเป็นกลุ่มเมื่อมีคุณสมบัติดังนี้

d-1 การปิด (Closure) สำหรับทุกค่า g และ h ซึ่งเป็นสมาชิกของ $\mathbb{G}$ แล้วผลลัพธ์ของ $g \circ h$ ก็เป็นสมาชิกของ $\mathbb{G}$ ด้วย

d-2 เอกลักษณ์ (Identity) ในกลุ่มของ $\mathbb{G}$ ต้องมีค่า e หนึ่งค่าที่เป็นเอกลักษณ์ คือ $g \circ e = g = e \circ g$

d-3 ตัวผกผัน (Inverse) สำหรับค่า g ทุกค่าที่เป็นสมาชิกของ $\mathbb{G}$ ต้องมีค่า h ซึ่ง เป็นสมาชิกของ $\mathbb{G}$ หนึ่งค่าที่ทำให้ $g \circ h = e = h \circ g$ เรียก h ว่าเป็นตัวผกผันของ g

d-4 การเปลี่ยนหมู่ได้ (Associativity) สำหรับค่า g_1, g_2, g_3 ซึ่งเป็นสมาชิกใดๆของกลุ่ม $\mathbb{G}$ จะได้ว่า $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

d-5 การสลับที่ได้ (Commutativity) สำหรับทุกค่า g และ h ซึ่งเป็นสมาชิกของ $\mathbb{G}$ แล้ว $g \circ h = h \circ g$

e. กลุ่ม Z_N^* เป็นกลุ่มซึ่งมีตัวดำเนินการทวิภาคเป็นการคูณมอดุโล N สมาชิกภายในกลุ่มจะมีค่าอยู่ในช่วง 1 ถึง N-1 และสมาชิกทุกตัวไม่มีตัวประกอบร่วมกับค่า N เขียนสัญลักษณ์แทนนิยามของกลุ่มได้ดังนี้

$$Z_N^* \overset{def}{=} \{a = \{1,\ldots, N\text{-}1\} \mid \gcd(a, N) = 1\};$$

f. กลุ่มวัฏจักรและตัวก่อกำเนิด (Cyclic Group and Generators) กลุ่มวัฏจักรเป็นกลุ่มที่สมาชิกทุกตัวสามารถสร้างได้จากตัวก่อกำเนิดตัวใดตัวหนึ่งเพียงหนึ่งตัวและตัวก่อกำเนิดนั้นต้องเป็นสมาชิกภายในกลุ่มวัฏจักรนั้น

ตัวอย่าง กลุ่ม $\mathbb{G} = \{g^0, g^1, g^2, g^3, g^4, g^5, g^6, g^7, g^8, g^9, g^{10}\}$ มีสมาชิกทั้งหมด 11 ตัว ซึ่ง g เป็นสมาชิกตัวหนึ่งของ $\mathbb{G}$ และ $g^{11} = g^0$ แล้ว ดังนั้น $\mathbb{G}$ เป็นกลุ่มวัฏจักรโดยมี g เป็นตัวก่อกำเนิด

ตัวอย่างการคำนวณ

ให้ p = 11 แล้ว 2 จะเป็นตัวก่อกำเนิดของกลุ่มมอดุโล 11 ดังนี้

$2^{10} = 1024 \equiv 1 \pmod{11}$

$2^{1} = 2 \equiv 2 \pmod{11}$

$2^{8} = 256 \equiv 3 \pmod{11}$

$2^{2} = 4 \equiv 4 \pmod{11}$

$2^{4} = 16 \equiv 5 \pmod{11}$

$2^{9} = 512 \equiv 6 \pmod{11}$

$2^{7} = 128 \equiv 7 \pmod{11}$

$2^{3} = 8 \equiv 8 \pmod{11}$

$2^{6} = 64 \equiv 9 \pmod{11}$

$2^{5} = 32 \equiv 10 \pmod{11}$

2.3.2 การหาจำนวนเฉพาะ (Prime Number Generation) ในปัจจุบันพบว่ามีจำนวนเฉพาะอยู่เป็นจำนวนมาก แต่ยังไม่ทราบแน่ชัดว่ามีจำนวนเท่าไหร่? และมีจำนวนใดบ้าง? และปัญหาที่สำคัญคือการตรวจสอบจำนวนเฉพาะที่มีขนาดใหญ่ (Large Prime) n บิตว่าเป็นจำนวนเฉพาะที่แท้จริงหรือไม่? เนื่องจากมีวิธีเข้ารหัสจำนวนมากที่อาศัยคุณสมบัติของจำนวนเฉพาะในการสร้างความแข็งแกร่งให้กับการเข้ารหัส หากจำนวนที่เลือกใช้ไม่ใช่จำนวนเฉพาะ หรือเป็นจำนวนเฉพาะที่มีขนาดเล็กจะทำให้วิธีเข้ารหัสนั้นมีจุดบกพร่องซึ่งง่ายต่อการโจมตี

2.3.3 การแยกตัวประกอบ (Factoring) เป็นปัญหาที่เก่าที่สุดปัญหาหนึ่งในทางทฤษฎีจำนวน ความซับซ้อนอยู่ที่ระยะเวลาที่ต้องใช้ในการแยกตัวประกอบจนได้ผลลัพธ์สุดท้ายเป็นจำนวนเฉพาะทั้งหมด โดยเฉพาะจำนวนที่ประกอบด้วยจำนวนเฉพาะขนาดใหญ่เพียง 2 ค่าจะทำให้การหาตัวประกอบทำได้ยากยิ่งขึ้น

2.3.4 ดีสครีตลอการิทึม (Discrete Logarithm) เป็นปัญหาที่เกิดจากความยากในการคำนวณย้อนกลับด้วยฟังค์ชันลอการิทึม เพื่อหาค่าเลขกำลังจากผลลัพธ์ที่ได้จากการมอดุลาร์สมการเลขยกกำลัง (Modular Exponentiation) ซึ่งเป็นการง่ายที่จะหาคำตอบของสมการ $x^a \equiv b \pmod{N}$ แต่ยากที่จะหาค่า a จากผลลัพธ์ b ด้วยสมการ $\log_x b = a$

2.4 วิธีพื้นฐานสำหรับพัฒนาการสร้างลายมือชื่อ

2.4.1 โปรโตคอลดิฟฟี-เฮลล์แมน (Diffie-Hellman protocol) เป็นวิธีการเข้ารหัสแบบใช้กุญแจอสมมาตรวิธีแรกที่ถูกพัฒนาขึ้นโดย Whitfield Diffie และ Martin Hellman เผยแพร่ในปี 1976 ซึ่งความปลอดภัยของวิธีอาศัยความยากจากปัญหาดีสครีตลอการิทึม โปรโตคอลนี้ใช้สำหรับการสร้างกุญแจลับร่วมกันระหว่างสมาชิกตั้งแต่ 2 คนผ่านช่องทางการสื่อสารที่ไม่มีความปลอดภัย โดยเริ่มจากการที่สมาชิกทุกคนตกลงร่วมกันในการเลือกใช้จำนวนเฉพาะ 2 ค่าคือ n และ g โดยที่ g เป็นเลขฐานสำหรับการยกกำลัง แล้วทำการคำนวณหาค่าเศษจากการหาร (Modular) ด้วยค่า n สมาชิกแต่ละรายจะทำการสุ่มเลือกค่าเพื่อใช้เป็นเลขกำลังซึ่งค่าที่เลือกจะถูกเก็บเป็นความลับ หลังจากนั้นสมาชิกแต่ละรายทำการเข้ารหัสโดยนำเลขฐาน g มายกกำลังด้วยค่าที่สมาชิกสุ่มเลือกแล้วหารเพื่อหาค่าเศษด้วย n ผลลัพธ์ที่ได้จะถูกส่งต่อไปยังสมาชิกลำดับถัดไป เมื่อการส่งต่อกุญแจวนครบจำนวนสมาชิก ผลลัพธ์สุดท้ายที่สมาชิกทุกคนได้รับจะมีค่าตรงกัน ดังรายละเอียดที่แสดงไว้ด้านล่าง

[1]. แก้วและขนุนตกลงใช้จำนวนเฉพาะ n และ g ร่วมกัน

[2]. แก้วสุ่มเลือกค่า x ซึ่งเป็นจำนวนเต็มบวกที่มีขนาดใหญ่ แล้วทำการคำนวณ $X = g^x \bmod n$ ส่งผลลัพธ์ X ไปให้ขนุน

[3]. ขนุนสุ่มเลือกค่า y ซึ่งเป็นจำนวนเต็มบวกที่มีขนาดใหญ่เช่นกัน แล้วทำการคำนวณ $Y = g^y \bmod n$ ส่งผลลัพธ์ Y ไปให้แก้ว

[4]. แก้วนำผลลัพธ์ที่ได้จากขนุนมาคำนวณอีกครั้งได้ $K = Y^x \bmod n$

[5]. ขนุนนำผลลัพธ์ที่ได้รับจากแก้วมาคำนวณเช่นกันจะได้ $K' = X^y \bmod n$

ค่า K และ K' ที่ได้จะมีค่าเท่ากันคือเท่ากับ $g^{xy} \bmod n$ ซึ่งแม้ว่าจะมีผู้เป็นปฏิปักษ์สามารถดักรับข้อมูลที่ถูกรับส่งในระหว่างการสื่อสารได้แก่ n, g, X และ Y ทั้งหมดก็ไม่สามารถจะคำนวณหาผลลัพธ์สุดท้ายที่เป็นกุญแจลับได้เนื่องจากไม่ทราบค่า x และ y

2.4.2 โปรโตคอลอาร์เอสเอ (RSA) พัฒนาขึ้นโดย Ron Rivest, Adi Shamir และ Leonard Adleman เผยแพร่ในปี 1977 เป็นวิธีเข้ารหัสแบบกุญแจอสมมาตรวิธีแรกที่ถูกนำมาประยุกต์ใช้ในการสร้างลายมือชื่อได้อย่างเหมาะสม ความปลอดภัยของการเข้ารหัสด้วยวิธีนี้ได้มาจากความยากของการหาจำนวนเฉพาะ 2 ค่าที่เป็นตัวประกอบของค่า n ซึ่งเป็นมอดุลัส วิธีเข้ารหัสแบบอาร์เอสเอมีกระบวนการทั้งหมด 3 ขั้นตอน ได้แก่

[1]. ขั้นตอนสร้างกุญแจ

a) ทำการสุ่มเลือกจำนวนเฉพาะ 2 ค่าคือ p และ q แล้วคำนวณหา $n = pq$

b) สุ่มเลือกจำนวนเฉพาะอีกค่าคือ e เพื่อใช้ e เป็นกุญแจในการเข้ารหัส โดยที่ค่า e จะต้องไม่ตัวประกอบร่วมกับค่า $(p\text{-}1)(q\text{-}1)$ หรือเรียกว่าเป็น Relatively prime ต่อกัน

c) สร้างกุญแจสาธารณะ d โดยคำนวณจาก $ed \equiv 1 \bmod (p\text{-}1)(q\text{-}1)$ หรือเท่ากับ $d = e^{-1} \bmod (p\text{-}1)(q\text{-}1)$ ค่า d ที่ได้จะต้องไม่มีตัวประกอบร่วมกับ $(p\text{-}1)(q\text{-}1)$ เช่นเดียวกัน

[2]. ขั้นตอนเข้ารหัส

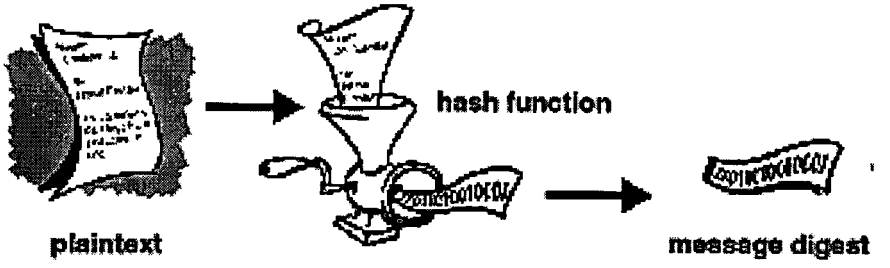
a) ข้อมูล m ที่จะเข้ารหัสควรมีค่าไม่เกิน n (2 ยกกำลังด้วยจำนวนบิตแล้วผลลัพธ์มีค่าน้อยกว่า n) หากมีค่ามากกว่าให้ทำการแบ่งข้อมูลออกเป็นชุดย่อยๆ แต่ละชุดมีค่าน้อยกว่า n

b) ทำการเข้ารหัสโดยคำนวณ $c = m^e \bmod n$ ผลลัพธ์ข้อความเข้ารหัส c

[3]. ขั้นตอนถอดรหัส คำนวณได้จาก $m = c^d \bmod n$

ความปลอดภัยของโปรโตคอลนี้อยู่ที่ความยากในการหาองค์ประกอบ p และ q ของค่า n ซึ่งเป็นส่วนสำคัญในการสร้างกุญแจ ซึ่งหากไม่ทราบค่า e ก็ไม่สามารถเข้ารหัสเพื่อสร้างค่า c ได้

2.4.3 ฟังค์ชันแฮช (Hash function) เป็นฟังค์ชันทางคณิตศาสตร์ที่ใช้เพื่อเปลี่ยนแปลงชุดข้อมูลจากเดิมที่มีขนาดความยาวไม่จำกัดไปเป็นชุดข้อมูลที่มีขนาดความยาวจำกัด และโดยทั่วไปความยาวของชุดข้อมูลที่ได้จะมีขนาดสั้นลง ซึ่งเรียกชุดข้อมูลที่ได้เป็นผลลัพธ์จากฟังค์ชันแฮชว่า ค่าแฮช (Hash value) หรือเรียกอีกอย่างหนึ่งว่า เมสเซสไดเจส (Message digest) หรือ เรียกสั้นๆ ว่า ไดเจส (Digest)



ภาพที่ 2.4 ฟังค์ชันแฮช

คุณสมบัติที่สำคัญของฟังค์ชันแฮชคือ

- ฟังค์ชันแฮชมีลักษณะเป็นฟังค์ชันแบบทิศทางเดียว (One-way function) คือ สามารถจะคำนวณจากข้อความตั้งต้นเพื่อหาค่าแฮชได้ แต่เป็นไปไม่ได้ที่จะคำนวณย้อนกลับจากค่าแฮชเพื่อหาข้อความตั้งต้น
- ทุกบิตของค่าแฮชจะขึ้นอยู่กับทุกบิตของข้อความตั้งต้น ดังนั้นจากข้อความตั้งต้นที่เหมือนกันจะต้องได้ค่าแฮชเดียวกันทุกครั้ง เรียกคุณสมบัตินี้ว่า การตรวจสอบได้ (Deterministic) นั่นคือหากมีการเปลี่ยนแปลงแม้เพียงบิตใดบิตหนึ่งของข้อความตั้งต้นจะส่งผลให้ค่าแฮชเปลี่ยนแปลงไป ดังนั้นโอกาสที่ข้อความตั้งต้นที่แตกต่างกัน 2 ข้อความใดๆ จะได้ค่าแฮชเดียวกัน (Collision) มีความเป็นไปได้น้อย

2.5 งานวิจัยที่เกี่ยวข้อง

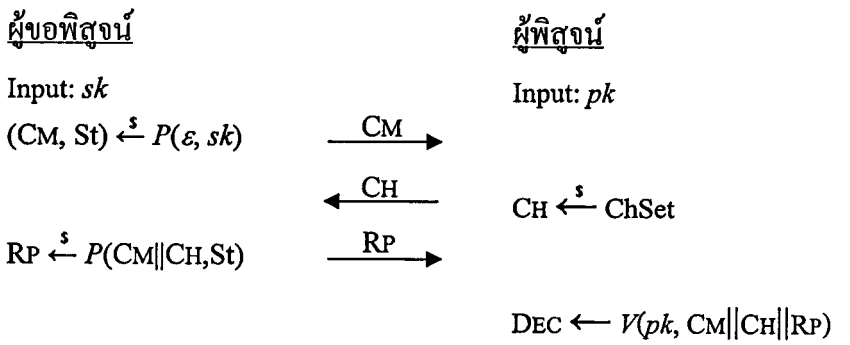
งานวิจัยที่จะนำเสนอต่อไปนี้เป็นงานศึกษาที่เกี่ยวข้องกับรายละเอียด ดังต่อไปนี้

1. การสร้างลายมือชื่อที่ไม่สามารถปลอมแปลงได้ยิ่งขึ้น
2. การประยุกต์ใช้วิธีพื้นฐานเพื่อพัฒนาการสร้างลายมือชื่อวิธีใหม่
3. การพัฒนาการสร้างลายมือชื่อที่มีความแข็งแกร่งขึ้น

2.5.1 Two-Tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. [6] เป็นงานวิจัยของ Mihir Bellare และ Sarah Shoup ในปี 2007 ซึ่งได้นำเสนอการสร้างลายมือชื่อแบบเข้ารหัส 2 ชั้นโดยใช้รูปแบบวิธีประยุกต์ของ Fiat-Shamir (Fiat-Shamir transform)

เดิมทีวิธีของ Fiat-Shamir เป็นวิธีสำหรับการพิสูจน์ตัวตนแบบที่ผู้พิสูจน์ไม่ทราบกุญแจลับของผู้ขอพิสูจน์ (Zero-Knowledge Identification protocol) ที่มีขั้นตอนการสื่อสาร 3 ขั้นตอน แล้วด้วยการประยุกต์ใช้ฟังค์ชันแฮชช่วยให้ขั้นตอนลดลงเหลือ 2 ขั้นตอนเพื่อนำไปใช้ในการสร้างลายมือชื่อ ดังรูปแบบต่อไปนี้

<u>วิธีเดิม</u>



ภาพที่ 2.5 วิธีพิสูจน์ตัวตนของ Fiat-Shamir

จากภาพเป็นการสื่อสารแบบ 2 ทิศทางระหว่างผู้ขอพิสูจน์กับผู้พิสูจน์ โดยเริ่มจากการที่ผู้ขอพิสูจน์ทำการสุ่มเลือกค่า ε แล้วทำการเข้ารหัสด้วย sk จะได้ผลลัพธ์เป็น (CM, St) แล้วส่งไปให้ผู้พิสูจน์ โดย CM ซึ่งเป็นข้อความเข้ารหัสจะถูกใช้เป็นข้อตกลงร่วมกัน (Commitment) ระหว่างผู้ขอพิสูจน์และผู้พิสูจน์สำหรับใช้ในการพิสูจน์ ผู้พิสูจน์จะทำการสุ่มเลือกค่า CH แล้วส่งคืนไปยังผู้ขอพิสูจน์ เพื่อให้ผู้ขอพิสูจน์ทำการเข้ารหัสอีกครั้ง ผลลัพธ์ RP ที่ได้จะถูกส่งกลับไปให้ผู้พิสูจน์เพื่อใช้ตรวจสอบความถูกต้องของค่าที่ได้หลังการถอดรหัส การพิสูจน์จะทำซ้ำทั้งหมด k รอบเพื่อสร้างความเชื่อมั่นในผลการพิสูจน์

การสร้างลายมือชื่อซึ่งประยุกต์มาจากวิธีพิสูจน์ตัวตนของ Fiat-Shamir

Algorithm skg(*ppk*, *psk*)	**Algorithm** sgn(*psk*, *ssk*, *m*)	**Algorithm** vf(*ppk*, *spk*, *m*, *s*)
Parse *ppk* as $K\|\|pk$	Parse *psk* as $K\|\|sk$	Parse *ppk* as $K\|\|pk$
Parse *psk* as $K\|\|sk$	Parse *ssk* as CM||St	$CM \leftarrow spk$
$(CM, St) \xleftarrow{\$} P(\varepsilon, sk)$	$CH \leftarrow H(K, CM\|\|m)$	$CH \leftarrow H(K, CM\|\|m)$
$spk \leftarrow CM$	$RP \xleftarrow{\$} P(CM\|\|CH, St)$	$RP \leftarrow s$
$ssk \leftarrow CM\|\|St$	$s \leftarrow RP$	$DEC \leftarrow V(pk, CM\|\|CH\|\|RP)$
Return (*spk*, *ssk*)	Return *s*	Return DEC

ภาพที่ 2.6 การสร้างลายมือชื่อซึ่งประยุกต์มาจากวิธีพิสูจน์ตัวตนของ Fiat-Shamir

จากภาพที่ 2.6 ขั้นตอน skg เป็นขั้นตอนการสร้างชุดกุญแจสำหรับใช้ในการสร้างและพิสูจน์ลายมือชื่อ การสร้างลายมือชื่อวิธีนี้ใช้กุญแจสาธารณะ 2 ชุดได้แก่ $K||pk$ เป็นกุญแจสาธารณะที่ 1 โดย $K||sk$ เป็นกุญแจลับที่ 1, CM เป็นกุญแจสาธารณะที่ 2 และ CM||St เป็นกุญแจลับที่ 2 โดยใช้ K เป็นกุญแจสำหรับฟังค์ชันแฮช, pk เป็นกุญแจสำหรับการถอดรหัสและ sk เป็นกุญแจสำหรับการเข้ารหัส ส่วนกุญแจชุดที่ 2 ซึ่งได้แก่ CM และ CM||St เป็นกุญแจสาธารณะ และกุญแจลับตามลำดับ จะใช้สำหรับการตรวจพิสูจน์ความถูกต้อง

ในขั้นตอน sgn เจ้าของลายมือชื่อจะทำการสร้าง CH ซึ่งใช้สำหรับการตรวจสอบความถูกต้องของเอกสารด้วยตัวเองโดยอาศัยฟังค์ชันแฮช มาช่วยแทนการรับค่าจากผู้พิสูจน์ ค่า CH เป็นค่าแฮชที่ได้จาก CM||m ซึ่ง m เป็นข้อความที่ต้องการรับรอง หลังจากนั้นจะทำการเข้ารหัสค่า CH และ CM||St ซึ่งเป็นกุญแจลับชุดที่ 2 ด้วย sk กุญแจลับชุดที่ 1 จะได้ลายมือชื่อเป็นผลลัพธ์

ขั้นตอน vf ผู้พิสูจน์จะใช้ข้อมูลข้อความ m, กุญแจสาธารณะที่ 1 และที่ 2 ในการพิสูจน์ลายมือชื่อ โดยการหาค่าแฮช CH และถอดรหัสลายมือชื่อ แล้วเปรียบเทียบว่าผลลัพธ์ที่ได้ถูกต้องตรงกับค่า CH และกุญแจสาธารณะที่ 2 หรือไม่ หากผลพิสูจน์ลายมือชื่อถูกต้องค่า DEC จะให้ผลลัพธ์เป็นบิต 1 แต่หากไม่ถูกต้องจะให้ค่าเป็นบิต 0

จากวิธีประยุกต์ข้างต้น ได้ถูกนำไปใช้กับการเข้ารหัสเพื่อพิสูจน์ตัวตนวิธีอื่น เช่น วิธีของ Schnorr โดยการเพิ่มฟังค์ชันแฮชแบบป้องกันการชนของผลลัพธ์ และกุญแจสาธารณะชุดที่ 2 เข้าไปในขั้นตอนการสร้างลายมือชื่อ ซึ่งรายละเอียดมีดังนี้

วิธีเดิม

Algorithm κ	**ผู้ขอพิสูจน์**		**ผู้พิสูจน์**
$g \xleftarrow{\$} G^*$	Input: $sk = (g, x)$		Input: $pk = (g, X)$
$x \xleftarrow{\$} Z_p$	$y \xleftarrow{\$} Zp$		
$X \leftarrow g^x$	$Y \leftarrow g^y$	$\xrightarrow{Y}$	
$pk \leftarrow (g, X)$		$\xleftarrow{c}$	$c \xleftarrow{\$} Z_p$
$sk \leftarrow (g, x)$	$z \leftarrow y+cx \bmod p\text{-}1$	$\xrightarrow{z}$	
Return (pk, sk)			ถ้า $g^z = YX^c$ แล้ว ค่า DEC เป็นบิต 1 แต่ถ้าไม่เท่ากัน ค่า DEC เป็นบิต 0

ภาพที่ 2.7 การพิสูจน์ตัวตนของ Schnorr

การประยุกต์วิธีพิสูจน์ตัวตนของ Schnorr

Alg. pkg	**Alg.** skg(ppk, psk)	**Alg.** sgn(psk, ssk, m)	**Alg.** vf(ppk, spk, m, s)
$K \xleftarrow{\$} \{0, 1\}^k$	$(K, g, X) \leftarrow ppk$	$(K, g, x) \leftarrow psk$	$(K, g, X) \leftarrow ppk$
$g \xleftarrow{\$} G^*$	$(K, g, x) \leftarrow psk$	$Y\|y \leftarrow ssk$	$Y \leftarrow spk$
$x \leftarrow Zp$	$y \xleftarrow{\$} Z_p$	$c \leftarrow H(K, Y\|m)$	$c \leftarrow H(K, Y\|m)$
$X \leftarrow g^x$	$Y \leftarrow g^y$	$z \leftarrow y+cx \bmod p\text{-}1$	$z \leftarrow s$
$ppk \leftarrow (K, g, X)$	$spk \leftarrow Y$	$s \leftarrow z$	ถ้า $g^z = YX^c$ แล้ว ค่า DEC เป็นบิต 1
$psk \leftarrow (K, g, x)$	$ssk \leftarrow Y\|y$	Return s	แต่ถ้าไม่เท่ากัน ค่า DEC เป็นบิต 0
Return (ppk, psk)	Return (spk, ssk)		

ภาพที่ 2.8 การสร้างลายมือชื่อจากวิธีพิสูจน์ตัวตนของ Schnorr

การสร้างลายมือชื่อด้วยวิธีนี้จะใช้กุญแจสาธารณะ 2 ชุด โดยเลือกใช้ประโยชน์ของปัญหาดีสครีตลอการิทึมสำหรับการสร้างกุญแจ ในขั้นตอน skg กุญแจชุดที่ 2 (Y, $Y\|y$) จะถูกสร้างขึ้นใหม่ทุกครั้งของการสร้างลายมือชื่อเพื่อป้องกันการโจมตีการคำนวณ $z \leftarrow y+cx \bmod p$ ในขั้นตอน sgn การใช้ฟังค์ชันแฮชแบบป้องกันการชนของผลลัพธ์เพื่อหาค่าแฮชของข้อความที่เชื่อมต่อกับกุญแจสาธารณะที่ 2 ($Y\|m$) ช่วยป้องกันการแก้ไขข้อความเพื่อปลอมแปลงลายมือชื่อได้ เพราะค่าแฮชในแต่ละครั้งจะมีความจำเพาะกับกุญแจสาธารณะที่ 2 ซึ่งกุญแจสาธารณะชุดที่ 2 เป็นกุญแจแบบใช้

ครั้งเดียว จึงมีความเป็นไปได้น้อยมากที่จะแก้ไขข้อความ m ซึ่ง $m_i \neq m$ แต่ $H(K, Y\|m_i) = H(K, Y\|m)$ ได้

จากการใช้ฟังค์ชันแฮชแบบป้องกันการชนของผลลัพธ์ ร่วมกับการใช้กุญแจชุดที่ 2 แบบใช้ครั้งเดียว สนับสนุนให้การสร้างลายมือชื่อวิธีนี้มีคุณสมบัติป้องกันการปลอมแปลงได้ดียิ่งขึ้น

2.5.2 Strongly Unforgeable Signature Based on Computational Diffie-Hellman. [1]

พัฒนาขึ้นโดย Dan Boneh, Emily Shen และ Brent Waters และเผยแพร่ในปี 2006 งานวิจัยนี้นำเสนอการสร้างลายมือชื่อที่สามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้นโดยอาศัยมาตรฐานของปัญหาการคำนวณสมการดิฟฟี-เฮลล์แมนในกลุ่มข้อมูลเชิงเส้นคู่ (Bilinear group) ในการสร้างความปลอดภัยให้กับระบบ

การสร้างลายมือชื่อวิธีนี้ได้จากการแปลงการสร้างลายมือชื่อที่มีคุณสมบัติยังสามารถป้องกันการปลอมแปลงได้ให้มีคุณสมบัติป้องกันการปลอมแปลงได้ดียิ่งขึ้น ซึ่งวิธีพื้นฐานที่จะสามารถนำมาประยุกต์ได้ต้องมีโครงสร้างของวิธีที่สามารถแบ่งส่วน (Partition) การคำนวณได้

วิธีที่สามารถแบ่งส่วนการคำนวณได้มีคุณสมบัติดังนี้

[1]. ขั้นตอนการสร้างลายมือชื่อ สามารถแบ่งเป็นขั้นตอนย่อยได้ 2 ขั้นตอน คือ F_1 และ F_2 การเข้ารหัสเพื่อสร้างลายมือชื่อสำหรับข้อความ m ด้วยกุญแจลับ SK สามารถทำได้ดังนี้

a. สุ่มเลือกค่า r ซึ่งเป็นสมาชิกของกลุ่ม R

b. กำหนดให้ $s_1 \leftarrow F_1(m, r, SK)$ และ $s_2 \leftarrow F_2(r, SK)$

c. ลายมือชื่อที่ได้คือ $s \leftarrow (s_1, s_2)$

[2]. หากให้ข้อมูล m และ s_2 แล้วจะมีเพียง s_1 ที่ถูกต้องเท่านั้นที่จะสามารถถอดรหัสด้วย PK เพื่อตรวจสอบความถูกต้องของลายมือชื่อที่กำกับข้อความ m

การสร้างลายมือชื่อในงานวิจัยนี้จะใช้รูปแบบการประยุกต์โครงสร้างตามการแบ่งส่วนการคำนวณข้างต้น มีรายละเอียดวิธีดังนี้

a. ขั้นตอนการสร้างกุญแจ

a-1 สุ่มเลือกค่าตัวก่อกำเนิด g และ h และสุ่มเลือกกุญแจแฮช k

a-2 ใช้โปรแกรม *KeyGen* สำหรับสร้างชุดกุญแจ (SK, PK) โดยกุญแจสาธารณะ และกุญแจลับที่ใช้สำหรับระบบนี้ได้แก่

$$PK' = (PK, g, h)$$

และ $$SK' = (SK)$$

b. ขั้นตอนการสร้างลายมือชื่อ

b-1 สุ่มเลือกค่ากำลัง $x \in Zp$ และค่า $r \in R$

b-2 กำหนดให้ $s_2 \leftarrow F_2(r, SK)$

b-3 คำนวณ $t \leftarrow H_k(M||s_2)$ ซึ่ง $t \in \{0, 1\}^n$ และต้องเป็นสมาชิกของ Z_p

b-4 คำนวณ $m \leftarrow g^t h^x$ ซึ่ง $m \in G$

b-5 คำนวณ $s_1 \leftarrow F_1(m, r, SK)$

b-6 ประกาศลายมือชื่อ $s \leftarrow (s_1, s_2, x)$

c. ขั้นตอนการพิสูจน์

c-1 คำนวณ $t' \leftarrow H_k(M||s_2)$ ซึ่ง t' ต้องเป็นสมาชิกของ Z_p

c-2 คำนวณ $m' \leftarrow g^{t'} h^x$

c-3 ถอดรหัส s_1 และ s_2 ด้วย PK แล้วตรวจสอบความถูกต้องของ m' กับ m

ความปลอดภัยของการสร้างลายมือชื่อวิธีนี้อยู่ที่การแบ่งส่วนการคำนวณ แล้วสามารถใช้ผลลัพธ์ที่ได้ในการตรวจสอบความถูกต้องซึ่งกัน การโจมตีโครงสร้างที่แข็งแกร่งของขั้นตอนการสร้างลายมือชื่อทำได้ยากด้วยเหตุผลดังนี้

- ค่า r ที่สุ่มเลือกจะถูกใช้สำหรับการสร้างลายมือชื่อ s_1 และ s_2
- การใช้ฟังชันแฮชแบบใช้กุญแจในการหาค่าแฮชของ $M||s_2$
- การใช้ค่าแฮช $M||s_2$ ที่ได้ในการคำนวณหาค่า m ด้วยวิธีแฮชแบบ chameleon
- ต้องใช้ค่า x ที่ได้จากการสุ่มเลือกและเป็นองค์ประกอบหนึ่งของชุดลายมือชื่อ s ในการคำนวณค่าแฮชแบบ chameleon

ด้วยเหตุผลดังกล่าวทำให้กาสร้างลายมือชื่อวิธีนี้มีคุณสมบัติสามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้น

2.5.3 Generic Transformation from Weakly to Strongly Unforgeable Signature. [9] งานวิจัยนี้เป็นการแปลงวิธีสร้างลายมือชื่อที่มีอยู่เดิมแต่ไม่มีความปลอดภัยให้เปลี่ยนเป็นวิธีที่สามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้น เป็นผลงานวิจัยของ Qiong Haung, Duncan S. Wong, Jin Li และ Yi-Ming Zhao เผยแพร่ในปี 2008

งานวิจัยนี้เลือกใช้เทคนิคการสร้างลายมือชื่อแบบใช้ครั้งเดียวที่แข็งแกร่ง (Strong One-Time Signature Scheme) เป็นพื้นฐานของโครงสร้างการแปลง การแปลงจะไม่ใช่การเปลี่ยนโครงสร้างของวิธีเข้ารหัสเดิม แต่เป็นการเพิ่มบางขั้นตอนในกระบวนการสร้างลายมือชื่อและการพิสูจน์ รายละเอียดการแปลงมีดังนี้

<u>การสร้างลายมือชื่อวิธีเดิม</u>

a. ขั้นตอนการสร้างกุญแจ ใช้โปรแกรม *KeyGen* สำหรับสร้างชุดกุญแจ (pk, sk) โดยกุญแจสาธารณะคือ pk ส่วน sk เป็นกุญแจลับ

b. ขั้นตอนการสร้างลายมือชื่อ ทำการคำนวณ

$$s \leftarrow \text{Sign}(sk, m)$$

ประกาศ s เป็นลายมือชื่อสำหรับข้อความ m

c. ขั้นตอนการพิสูจน์ คำนวณ

$$m' \leftarrow \text{Vrfy}(pk, s)$$

แล้วเปรียบเทียบค่า m กับ m'

<u>วิธีที่ได้หลังการแปลง</u>

a. ขั้นตอนการสร้างกุญแจ ใช้โปรแกรม *KeyGen* สำหรับสร้างชุดกุญแจ (pk, sk) โดยกุญแจสาธารณะคือ pk ส่วน sk เป็นกุญแจลับ

b. ขั้นตอนการสร้างลายมือชื่อ

b-1 ใช้โปรแกรม $KeyGen_{OT}$ สร้างชุดกุญแจ (pk_{OT}, sk_{OT}) เป็นกุญแจแบบใช้ครั้งเดียว

b-2 เข้ารหัสกุญแจสาธารณะแบบใช้ครั้งเดียว pk_{OT} ด้วยกุญแจลับ sk

$$s_1 \leftarrow \text{Sign}\ (sk, pk_{OT})$$

b-3 เชื่อมข้อความ m กับลายมือชื่อ s_1 แล้วคำนวณ

$$s_2 \leftarrow \text{Sign}_{OT}\ (sk_{OT}, m\|s_1)$$

b-4 ประกาศชุดลายมือชื่อ $s \leftarrow (s_1, s_2, pk_{OT})$

c. จากค่านำเข้าได้แก่ ข้อความ m, กุญแจสาธารณะ pk และลายมือชื่อ $s \leftarrow (s_1, s_2, pk_{OT})$

c-1 ทำการถอดรหัส s_1 ด้วยกุญแจสาธารณะ pk เพื่อคืนค่ากุญแจสาธารณะแบบใช้ครั้งเดียว pk_{OT} แล้วเปรียบเทียบผลลัพธ์ที่ได้กับ pk_{OT} ในลายมือชื่อ s

$$b_1 \leftarrow \text{Vrfy}\ (pk, s_1, pk_{OT})$$

หากผลลัพธ์ที่ได้ถูกต้อง ค่า b_1จะเป็นบิต 1 แต่หากไม่ถูกต้องก็จะเป็นบิต 0

c-2 ใช้กุญแจสาธารณะแบบใช้ครั้งเดียว pk_{OT} ถอดรหัส s_2 เพื่อคืนค่า $m\|s_1$ ตรวจสอบผลลัพธ์ที่ได้จากการถอดรหัสกับข้อความ m ที่เชื่อมต่อกับลายมือชื่อ s_1

$$b_2 \leftarrow \text{Vrfy}\ (pk_{OT}, s_2, m\|s_1)$$

หากผลลัพธ์ที่ได้ถูกต้อง ค่า b_2จะเป็นบิต 1 แต่หากไม่ถูกต้องก็จะเป็นบิต 0

c-3 หากผลลัพธ์ b_1 และ b_2 เป็นบิต 1 ทั้ง 2 ค่าแสดงว่าลายมือชื่อนั้นถูกต้อง หากไม่แล้วลายมือชื่อนั้นจะไม่ใช่ลายมือชื่อที่ถูกต้อง

ความปลอดภัยของการสร้างลายมือชื่อที่ได้จากการแปลงวิธีตามงานวิจัยนี้เกิดจากปัจจัยดังต่อไปนี้

- การใช้กุญแจชุดที่ 2 เป็นกุญแจแบบใช้ครั้งเดียว
- โครงสร้างของการเข้ารหัส s_1 และ s_2 มีการจัดวางตัวแปร pk_{OT} และ s_1 ในรูปแบบซึ่งยากแก่การโจมตี เพราะการเปลี่ยนแปลงที่ค่าใดค่าหนึ่งจะส่งผลกระทบต่อตัวแปรที่เหลืออย่างหลีกเลี่ยงไม่ได้
- ความถูกต้องของลายมือชื่อจำเพาะที่กรณีที่ผลลัพธ์ b_1 และ b_2 เป็นบิต 1 ทั้ง 2 ค่าเท่านั้น

ด้วยเหตุผลดังกล่าวทำให้การสร้างลายมือชื่อวิธีที่ได้จากการแปลงโครงสร้างตามงานวิจัยนี้มีคุณสมบัติสามารถป้องกันการปลอมแปลงได้ดียิ่งขึ้น

2.5.4 Signature Schemes Based on The Strong RSA Assumption. [11] เป็นการเข้ารหัสวิธีใหม่เพื่อใช้ในการสร้างลายมือชื่อ โดยพัฒนามาจากสมมติฐานอาร์เอสเอที่แข็งแกร่ง เป็นผลงานของ Ronald Cramer และ Victor Shoup เผยแพร่ในปี 2000

แนวคิดสมมติฐานของอาร์เอสเอที่แข็งแกร่งเป็นความยากในการแก้ปัญหาดังนี้ จากรูปสมการของอาร์เอสเอ $y^r = z \bmod n$ หากทราบเพียงค่ามอดุลัส n และผลลัพธ์ z ซึ่ง $z \in Z_n^*$ แล้วเป็นการยากที่จะหาค่า y ซึ่ง $y \in Z_n^*$ และค่า $r > 1$ ที่จะทำให้สมการเป็นจริงได้ จะเห็นว่าสมมติฐานนี้แตกต่างจากสมมติฐานของอาร์เอสเอปกติ โดยทั่วไปค่า r จะถูกเลือกได้อย่างอิสระ แต่สำหรับสมมติฐานของอาร์เอสเอที่แข็งแกร่งขึ้นนั้นค่า r จะขึ้นอยู่กับค่า z

การสร้างลายมือชื่อจากสมมติฐานของอาร์เอสเอที่แข็งแกร่งขึ้นมีกระบวนการดังนี้

<u>ขั้นตอนสร้างกุญแจ</u>

a. สุ่มเลือกค่า p และ q ซึ่ง $p = 2p' + 1$ และ $q = 2q' + 1$, โดยที่ค่า p, p', q และ q' เป็นจำนวนเฉพาะทั้งหมด แล้วคำนวณ $n = pq$
b. สุ่มเลือกค่า h และ r ซึ่งทั้ง $h, r \in \mathrm{QR}_n$
c. เลือกจำนวนเฉพาะ e' ซึ่งมีขนาด $l + 1$ บิต
d. กำหนดกุญแจสาธารณะได้แก่ (n, h, x, e') และกุญแจลับได้แก่ (p, q)

<u>ขั้นตอนสร้างลายมือชื่อ</u>

a. สุ่มเลือกจำนวนเฉพาะ $e \neq e'$ ขนาด $l + 1$ บิต และสุ่มเลือกค่า $y' \in \mathrm{QR}_n$
b. จากสมการ

$$(y')^{e'} = x' h^{H(m)}$$

คำนวณหาค่า x'

c. คำนวณหาค่า y จากสมการ

$$y^e = xh^{H(x')}$$

d. กำหนดข้อมูลลายมือชื่อได้แก่ (e, y, y')

<u>ขั้นตอนพิสูจน์</u>

a. ตรวจสอบค่า e และ e' ว่าเป็นจำนวนคี่ขนาด $l + 1$ บิตหรือไม่ และ $e \neq e'$

b. แทนค่า y' ในสมการ

$$x' = (y')^{e'} h^{-H(m)}$$

คำนวณหาค่า x'

c. แทนค่า x' ที่ได้ในสมการ

$$x = y^e h^{-H(x')}$$

คำนวณหาค่า y

d. เปรียบเทียบผลลัพธ์ที่ได้กับค่า y จากลายมือชื่อว่าถูกต้องตรงกันหรือไม่

ความปลอดภัยของการเข้ารหัสวิธีนี้ ยังคงอาศัยสมมติฐานของปัญหาการแยกตัวประกอบของค่ามอดุลัสเป็นพื้นฐาน และใช้คุณสมบัติของฟังค์ชันแฮชซึ่งป้องกันการชนกันของค่าแฮช เพื่อให้ได้คุณสมบัติสามารถป้องกันการปลอมแปลงลายมือชื่อได้

2.5.5 Cryptanalysis of a Verifiably Committed Signature Scheme Based on GPS and RSA. [2] เป็นการสร้างลายมือชื่อแบบมีข้อความผูกมัดที่พัฒนามาจากวิธีพื้นฐาน 2 วิธีคือ จีพีเอส (GPS) และอาร์เอสเอ เป็นผลงานวิจัยของ Julien Cathalo, Benoit Libert และ Jean-Jacques Quisquater ตีพิมพ์ในปี 2004 ซึ่งเป็นการให้รายละเอียดจุดบกพร่องของวิธีหลังการเผยแพร่ผลงานครั้งแรกในปี 2001

- การสร้างลายมือชื่อแบบจีพีเอส

<u>ขั้นตอนสร้างกุญแจ</u>

a. สุ่มเลือกค่า p และ q ซึ่ง $p = 2p' + 1$ และ $q = 2q' + 1$, โดยที่ค่า p, p', q และ q' เป็นจำนวนเฉพาะทั้งหมด แล้วคำนวณ $n = pq$

b. กำหนดค่า A, B และ S ซึ่ง $A >> BS$

c. เลือกฟังก์ชันแฮช h ซึ่งค่าแฮชมีค่าอยู่ในช่วง $[0, B)$

d. เลือกค่า α เพื่อเป็นค่าฐานสำหรับการยกกำลังแล้วมอดุโล n โดยสามารถให้ผลลัพธ์จากการยกกำลังได้หลากหลายที่สุด

e. เลือกค่า x แล้วคำนวณ $y = \alpha^x \bmod n$

ได้กุญแจสาธารณะเป็น (n, α, h, y) และกุญแจลับเป็น (x)

<u>ขั้นตอนสร้างลายมือชื่อ</u>

a. สุ่มเลือกค่า r ซึ่งมีค่าอยู่ในช่วง $[0, A[$ แล้วคำนวณ $t = \alpha^r \bmod n$

b. สร้างลายมือชื่อให้กับข้อความ m โดยคำนวณ $z = r + h(t, m)x$ ค่า z ที่ได้จะมีค่าอยู่ในช่วง $[0, A + (B\text{-}1)(S\text{-}1)[$

ได้ลายมือชื่อเป็น (t, z) และกุญแจลับ (r)

<u>ขั้นตอนพิสูจน์ลายมือชื่อ</u>

a. ตรวจสอบค่าแฮชที่ได้จาก $h(t, m)$ มีค่าอยู่ในช่วง $[0, B[$

b. ตรวจสอบ $\alpha^z \equiv ty^{h(t, m)} \pmod n$

การสร้างลายมือชื่อแบบจีพีเอสได้รับการพิสูจน์แล้วว่ามีคุณสมบัติสามารถป้องกันการปลอมแปลงจากการโจมตีแบบเลือกข้อมูลได้

- การสร้างลายมือชื่อที่พัฒนาจากการสร้างลายมือชื่อแบบจีพีเอสร่วมกับโปรโตคอลอาร์-เอสเอมีรายละเอียดดังนี้

<u>ขั้นตอนสร้างกุญแจ</u>

บุคคลที่ 1

1) สุ่มเลือกค่า p และ q ซึ่ง $p = 2p' + 1$ และ $q = 2q' + 1$, โดยที่ค่า p, p', q และ q' เป็นจำนวนเฉพาะทั้งหมด แล้วคำนวณ $n = pq$
2) เลือกค่า c ซึ่งเป็นจำนวนเต็มที่ไม่มีตัวประกอบร่วมกับ $p'q'$ แล้วคำนวณหาค่า $d = c^{-1} \bmod p'q'$
3) เลือกค่า α เพื่อเป็นค่าฐานสำหรับการยกกำลัง
4) ได้กุญแจสาธารณะเป็น (n, α, h, c) และกุญแจลับเป็น (d)

บุคคลที่ 2

1) สุ่มเลือกจำนวนเต็ม x แล้วคำนวณ $y = \alpha^x \bmod n$

ได้กุญแจสาธารณะเป็น (y) และกุญแจลับเป็น (x)

<u>ขั้นตอนสร้างลายมือชื่อย่อย</u>

บุคคลที่ 2

1) สุ่มเลือกค่า r เพื่อคำนวณ $t = \alpha^{cr} \bmod n$
2) คำนวณ $z = cr + h(t, m)x$

ได้ข้อมูลลายมือชื่อย่อยเป็น (t, z)

ขั้นตอนพิสูจน์ลายมือชื่อย่อย

บุคคลที่ 3

1) ตรวจสอบว่า $\alpha^z \equiv ty^{h(t, m)} \pmod{n}$

ขั้นตอนสร้างลายมือชื่อหลัก

บุคคลที่ 2

1) ใช้ค่า r จากขั้นตอนสร้างลายมือชื่อย่อย นำมาคำนวณ $t' = \alpha^r \bmod n$
2) และนำค่า z ที่ได้จากการคำนวณในขั้นตอนสร้างลายมือชื่อย่อยมาใช้ ได้ชุดข้อมูลลายมือชื่อหลักเป็น (t', z)

ขั้นตอนพิสูจน์ลายมือชื่อหลัก

บุคคลที่ 3

1) ตรวจสอบว่า $\alpha^z \equiv t'^c y^{h(t'^c \bmod n, m)} \pmod{n}$

ในกรณีที่บุคคลที่ 2 ปฏิเสธที่จะให้ข้อมูลลายมือชื่อแก่บุคคลที่ 3 บุคคลที่ 1 สามารถใช้ชุดข้อมูลลายมือชื่อย่อย (t, z) และกุญแจลับ d ในการคำนวณหาค่า (t', z) ได้จาก $t' = t^d \bmod n$

จุดบกพร่องที่พบ คือ จากการที่ขั้นตอนสร้าง และพิสูจน์ลายมือชื่อทั้งหมดต้องใช้ค่า z ในการคำนวณ โดย $z = cr + h(t, m)x$ หากนำสมการนี้มามอดุโลด้วย c ซึ่ง (n, α, h, c) เป็นส่วนกุญแจสาธารณะที่ได้จากบุคคลที่ 1 แล้วจะได้ว่า $z \bmod c = h(t, m)x$ ทำให้สามารถคำนวณหาค่า x ซึ่งเป็นกุญแจลับของบุคคลที่ 2 ได้ ส่งผลให้การสร้างลายมือชื่อวิธีนี้ไม่มีความปลอดภัยสำหรับการใช้งาน

บทที่ 3
การออกแบบและพัฒนาการสร้างลายมือชื่อ

การสร้างลายมือชื่อมีความสำคัญต่อการพัฒนาระบบการเข้ารหัสที่ปลอดภัยต่อการโจมตีแบบเลือกชุดข้อมูลด้วยข้อความเข้ารหัสได้ (Chosen-Ciphertext Attack Secure Encryption System) เพราะลักษณะการใช้งานลายมือชื่อซึ่งเป็นข้อความเข้ารหัสของข้อความที่ต้องการรับรองมีความจำเพาะกับข้อความที่กำกับไว้เท่านั้นไม่สามารถนำไปใช้เพื่อรับรองข้อความอื่นได้ แต่การสร้างลายมือชื่อโดยมากเมื่อผู้ใช้ใช้เพื่อเข้ารหัสข้อความไประยะหนึ่งจะมีชุดข้อมูล (ข้อความ, ลายมือชื่อ) จำนวนมากพอที่ผู้เป็นปฏิปักษ์จะเลือกใช้การโจมตีแบบเลือกข้อมูลเพื่อเปิดเผยความลับของการสร้างลายมือชื่อและนำไปสู่การปลอมแปลงลายมือชื่อได้

การสร้างลายมือชื่อที่มีความแข็งแกร่งจะสามารถให้ผลลัพธ์เป็นลายมือชื่อที่มีคุณสมบัติป้องกันการโจมตีแบบเลือกข้อมูลได้ นั่นคือแม้ว่าในระบบจะมีชุดข้อมูล (ข้อความ, ลายมือชื่อ) เป็นจำนวนมากแต่ก็ไม่ช่วยให้ผู้เป็นปฏิปักษ์สามารถปลอมแปลงลายมือชื่อได้ ซึ่งเรียกลายมือชื่อที่มีคุณสมบัตินี้ว่าลายมือชื่อที่ปลอมแปลงได้ยาก และเรียกวิธีสร้างลายมือชื่อที่มีคุณสมบัตินี้ว่าการสร้างลายมือชื่อที่ปลอมแปลงได้ยาก

งานวิจัยฉบับนี้จึงมีวัตถุประสงค์เพื่อพัฒนาการสร้างลายมือชื่อที่มีคุณสมบัติปลอมแปลงได้ยาก โดยมีขั้นตอนการออกแบบและพัฒนาวิธีการดังนี้

3.1 การออกแบบการสร้างลายมือชื่อ

จากแนวคิดและทฤษฎีในบทที่ 2 นำมาวิเคราะห์เพื่อการออกแบบการสร้างลายมือชื่อที่มีคุณสมบัติปลอมแปลงได้ยากขึ้น โดยมีลำดับขั้นตอนดังนี้

3.1.1 แนวคิดจากงานวิจัยอื่น

จากการศึกษางานวิจัยอื่นๆ ในบทที่ 2 พบว่าการสร้างลายมือชื่อในงานวิจัย Two-Tier signatures, Strongly Unforgeable Signatures, and Fiat-Shamir without Random Oracles. [6] มีโครงสร้างที่ไม่ซับซ้อน วิธีพื้นฐานที่ใช้ในการพัฒนาต้องสร้างมาจากสมมติฐานปัญหาดีสครีตลอการิทึม เช่น โปรโตคอลของ Schnorr เป็นต้น การปรับปรุงโปรโตคอลทำโดยเพิ่มชุดกุญแจที่ต้องใช้ในขั้นตอนการเข้ารหัสจากเดิมใช้กุญแจในการเข้ารหัสเพียง 1 ชุดเปลี่ยนเป็น 2 ชุดดังนี้

- วิธีเดิม $z = xc \bmod (p\text{-}1)$ (1)

  กุญแจที่ใช้ได้แก่ $(x,\ X)$ ซึ่ง $X = g^x \bmod p$ โดย z เป็นผลลัพธ์ที่ได้จากการเข้ารหัสข้อความ c ด้วยกุญแจลับ x

- หลังการปรับปรุงแล้ว $z = y + xc \bmod (p\text{-}1)$ (2)

กุญแจที่ใช้ได้แก่ (x, X) และ (y, Y) ซึ่ง $X = g^x \bmod p$ และ $Y = g^y \bmod p$

โครงสร้างการเข้ารหัสที่ได้จากการใช้กุญแจซึ่งสร้างจากปัญหาดีสครีตลอการิทึม 2 ชุด พบว่ามีข้อจำกัดบางอย่างเกิดขึ้น ความแข็งแกร่งของการเข้ารหัสขึ้นอยู่กับการใช้งานกุญแจชุดที่ 2 (y, Y) การใช้กุญแจชุดที่ 2 ซ้ำจะส่งผลให้ความลับของกุญแจชุดที่ 1 ซึ่งเป็นชุดกุญแจหลักถูกเปิดเผยได้ดังนี้

การเข้ารหัสครั้งที่ 1 $z_1 = x + yc_1 \bmod (p\text{-}1)$ (3)

การเข้ารหัสครั้งที่ 2 $z_2 = x + yc_2 \bmod (p\text{-}1)$ (4)

$z_2 - z_1 = y(c_2 - c_1) \bmod (p\text{-}1)$ (5)

หาก $c_2 - c_1 = \pm 1$ (6)

จะได้ $y = z_2 - z_1$ (7)

เมื่อแทนค่า y ในสมการเข้ารหัสจะสามารถหาค่ากุญแจ x ได้

ด้วยข้อจำกัดนี้ผู้ใช้จำเป็นต้องมีระบบจัดการข้อมูลการใช้กุญแจที่ดีเพื่อป้องกันการสุ่มใช้กุญแจชุดที่ 2 ชุดเดิมซ้ำ

แนวคิดที่เป็นประโยชน์อีกประการหนึ่งคือการใช้ฟังก์ชันแฮชแบบป้องกันการชนของผลลัพธ์ โดยค่าแฮชในแต่ละครั้งมีความจำเพาะกับกุญแจสาธารณะที่ 2 ซึ่งไม่มีการใช้ซ้ำ ทำให้การแก้ไขข้อความเพื่อปลอมแปลงลายมือชื่อได้มีความเป็นไปได้น้อยมาก

การสร้างลายมือชื่อแบบตรวจสอบข้อตกลงได้ในงานวิจัย Cryptanalysis of A Verifiably Committed Signature Scheme Based on GPS and RSA. [2] พัฒนามาจากวิธีเข้ารหัสแบบอาร์เอสเอและจีพีเอส เป็นตัวอย่างของการปรับปรุงวิธีเข้ารหัสใหม่ให้มีความปลอดภัยกว่าวิธีเดิมที่ประสบความล้มเหลว ด้วยเหตุผลที่เดิมโปรโตคอลจีพีเอสพัฒนามาจากวิธีเข้ารหัสอาร์เอสเอ โดยการใช้ปัญหาดีสครีตลอการิทึมในสมการที่มีค่ามอดุลัสเป็นผลคูณของจำนวนเฉพาะ 2 ค่า ดังนี้

- ค่ามอดุลัส $n = pq$ (8)
- กุญแจชุดที่ 1 (x, y) ซึ่ง $y = \alpha^x \bmod n$ (9)
- กุญแจชุดที่ 2 (r, t) ซึ่ง $t = \alpha^r \bmod n$ (10)
- การเข้ารหัสเพื่อสร้างลายมือชื่อ $z = r + h(t, m)x$ (11)
- การถอดรหัสจะนำค่า α มายกกำลังด้วยสมการเข้ารหัสดังนี้

$$\alpha^z \bmod n = \alpha^{r+h(t,m)x} \bmod n \quad (12)$$

$$= ty^{h(t,m)} \bmod n \quad (13)$$

ด้วยข้อจำกัดของการใช้ปัญหาดีสครีตลอการิทึมกับกลุ่ม Z_n^* ซึ่งเป็นกลุ่มที่มีตัวดำเนินการทวิภาคเป็นการคูณมอดุโล n (ดูหัวข้อ 2.3.1 ในบทที่ 2 ประกอบ) จึงไม่มีการคำนวณมอดุเลชั่นในสมการ (11) ทำให้ผลลัพธ์ z ที่ได้จะมีขนาดใหญ่โดยขึ้นอยู่กับขนาดของ $x \in [0, S[$, $r \in [0, A[$ และ $h(t, m) \in [0, B[$ ดังนั้นขนาดของ $z \in [0, A + (B-1)(S-1)[$

โปรโตคอลจีพีเอสมีคุณสมบัติสามารถป้องกันการปลอมแปลงได้ในลักษณะเดียวกับสมการ (2) ของงานวิจัย [6] คือ การเลือกกุญแจลับ r ใหม่ทุกครั้งเมื่อต้องการสร้างลายมือชื่อ แต่จากการไม่ได้ใช้มอดุเลชันส่งผลให้โอกาสที่กุญแจ x ถูกเปิดเผยเมื่อมีการใช้กุญแจชุดที่สองซ้ำมีความเป็นไปได้มากกว่างานวิจัย [6] ผู้วิจัยจึงเลือกใช้การเข้ารหัสอาร์เอสเอในการปรับปรุงโครงสร้างการเข้ารหัสเพื่อเพิ่มความปลอดภัยให้กับโปรโตคอลจีพีเอส แต่ชุดกุญแจอาร์เอสเอที่นำมาใช้ประกอบสมการไม่ใช่กุญแจส่วนตัวของผู้เป็นเจ้าของกุญแจลับจีพีเอสแต่เป็นของผู้ที่ทำหน้าที่เป็นคนกลางระหว่างเจ้าของกุญแจจีพีเอสกับผู้พิสูจน์ โดยใช้กุญแจสาธารณะ c ของผู้เป็นคนกลางคูณกับกุญแจลับที่สอง r ก่อนดำเนินการบวกกับผลคูณของค่าแฮชกับกุญแจลับที่หนึ่งตามสมการ

$$z = cr + h(t, m)x \tag{14}$$

การเลือกใช้กุญแจสาธารณะ c เพื่อเพิ่มความแข็งแกร่งให้สมการแทนการใช้กุญแจลับ d ทำให้จุดแข็งของสมการ (11) ที่มีอยู่หมดไป โดยการนำสมการ (14) มามอดุโลด้วยกุญแจสาธารณะ c ซึ่งเป็นตัวแปรหนึ่งในสมการทำให้ได้

$$z \bmod c = (cr + h(t, m)x) \bmod c \tag{15}$$

$$= h(t, m)x \bmod c \tag{16}$$

ด้วยผลลัพธ์ z และค่าแฮช $h(t, m)$ เป็นข้อมูลที่ถูกเปิดเผย ดังนั้นการคำนวณตามสมการ (16) เพื่อหากุญแจลับ x จึงสามารถทำได้ไม่ยาก ซึ่งสาเหตุหลักของจุดอ่อนในกรณีนี้เกิดจากตัวแปรที่เลือกใช้เป็นองค์ประกอบรวมทั้งการจัดวางรูปแบบสมการซึ่งมองเห็นจุดบกพร่องได้อย่างชัดเจนเป็นเหตุให้การสร้างลายมือชื่อวิธีนี้ถูกโจมตีและสามารถเปิดเผยกุญแจลับได้โดยง่าย ดังนั้นในการกำหนดโครงสร้างของวิธีควรคำนึงถึงข้อบกพร่องที่อาจเกิดขึ้นจากการวางรูปแบบสมการและข้อจำกัดของวิธีพื้นฐานที่เลือกใช้

งานวิจัย Strongly Unforgeable Signature Based on Computational Diffie-Hellman. [1] ให้ความสำคัญกับการเตรียมค่า m ซึ่งเป็นผลลัพธ์ที่ได้จากการคำนวณข้อความ M ด้วยฟังก์ชันแฮชแบบ Chameleon คือ $m = g^{H_k(M \| s_2)} h^x$ ความยากของการปลอมแปลงลายมือชื่อขึ้นอยู่กับปัญหาดีสครีตลอการิทึมและคุณสมบัติป้องกันการชนของฟังก์ชันแฮช

การสร้างลายมือชื่อในงานวิจัย Signature Schemes Based on The Strong RSA Assumption. [11] แม้ว่าจะมีรูปแบบของการเข้ารหัสแบบอาร์เอสเอ แต่ไม่มีการใช้กุญแจเข้ารหัสและถอดรหัสของอาร์เอสเอ แนวคิดที่สำคัญของการสร้างลายมือชื่อวิธีนี้คือ สมมติฐานของปัญหา $y^r = z \bmod n$ ซึ่งมีลักษณะผกผันกับปัญหาดีสครีตลอการิทึมที่ต้องการหาค่ากำลัง r ที่ทำให้สมการเป็นจริง แต่อาร์เอสเอที่แข็งแกร่งต้องการหาค่าตัวก่อกำเนิด y ที่สามารถทำให้สมการเป็นจริง ความยากของปัญหาอย่างน้อยยังคงเทียบเท่าปัญหาอาร์เอสเอ แต่มีรูปแบบที่ต่างไป

3.1.2 วิธีพื้นฐานที่เลือกใช้สำหรับพัฒนาการสร้างลายมือชื่อ

ในงานวิจัยนี้ได้เลือกใช้วิธีเข้ารหัสพื้นฐาน 2 วิธี ได้แก่วิธีเข้ารหัสแบบอาร์เอสเอและวิธีเข้ารหัสแบบดิฟฟี-เฮลล์แมนซึ่งเป็นวิธีเข้ารหัสแบบใช้กุญแจสาธารณะทั้งสองวิธี แต่มีคุณสมบัติที่ต่างกัน ดังนี้

ตารางที่ 3.1 เปรียบเทียบคุณสมบัติของการเข้ารหัสแบบอาร์เอสเอกับแบบดิฟฟี-เฮลล์แมน

	อาร์เอสเอ	**ดิฟฟี่-เฮลล์แมน**
สมมติฐาน	ปัญหาของการแยกตัวประกอบ	ปัญหาดีสครีตลอการิทึม
การใช้งาน	• ใช้จำนวนเฉพาะ 2 ค่า คือ p และ q • กำหนด d เป็นกุญแจลับและ e เป็นกุญแจสาธารณะ • ในการเข้ารหัสจะยกกำลัง m (ข้อความ) ด้วยกุญแจลับ d แล้วมอดุโลด้วย n ได้ผลลัพธ์เป็นลายมือชื่อ (s) • เมื่อถอดรหัสจะยกกำลัง s (ลายมือชื่อ) ด้วยกุญแจสาธารณะ e แล้วมอดุโลด้วย n จะได้ผลลัพธ์เป็นข้อความ m กลับคืนมา	• ใช้จำนวนเฉพาะเพียง 1 ค่า คือ p • กำหนด x เป็นกุญแจลับและ X เป็นกุญแจสาธารณะ • ลายมือชื่อ (s) เป็นผลลัพธ์ที่ได้จากการถูณข้อความ m ด้วยกุญแจลับ x แล้วมอดุโลด้วย (p-1) • เมื่อนำกุญแจสาธารณะ X มายกกำลังด้วยข้อความ m แล้วมอดุโล p และยกกำลังค่าก่อกำเนิด g ด้วยลายมือชื่อ s แล้วมอดุโล p ผลลัพธ์ที่ได้ควรมีค่าเท่ากัน
สมการ • **การสร้างกุญแจ** • **การเข้ารหัส** • **การพิสูจน์**	$n = pq$ $d = e^{-1} \bmod (p\text{-}1)(q\text{-}1)$ $s = m^d \bmod n$ $s^e = m \bmod n$	$X = g^x \bmod p$ $s = mx \bmod (p\text{–}1)$ $X^m = g^s \bmod p$
ข้อจำกัด	• จำนวนเฉพาะ p และ q ควรมีขนาดใหญ่มากพอ (ดู [8] ประกอบ) • ค่าของกุญแจสาธารณะ e และกุญแจลับ d ต้องไม่มีตัวประกอบร่วมกับ (p-1)(q-1) • ข้อความ m ต้องมีค่าไม่เกินค่ามอดุลัส n • ลายมือชื่อ s ที่ได้จะเป็นสมาชิกของ Z_n เสมอ	• จำนวนเฉพาะ p ควรมีขนาดใหญ่มากพอ (ดู [8] ประกอบ) และควรมีจำนวนเฉพาะ q ซึ่งมีขนาดใหญ่เป็นตัวประกอบของ (p–1) • ค่าตัวก่อกำเนิด g ต้องเป็นสมาชิกของ Z_p และ $g^q \equiv 1 \pmod p$ • กุญแจลับ x ต้องมีค่าน้อยกว่า q

3.1.3 โครงสร้างวิธีการเข้ารหัส

จากคุณสมบัติที่ต่างกันของทั้ง 2 วิธี ในการกำหนดโครงสร้างหลักของการสร้างลายมือชื่อจะใช้วิธีเข้ารหัสแบบ 2 ชั้น โดยชั้นแรกจะเป็นการเข้ารหัสด้วยวิธีดีฟฟี-เฮลล์แมนและชั้นที่สองจะเป็นการเข้ารหัสวิธีอาร์เอสเอ ดังนั้นในการถอดรหัสจะกระทำย้อนกลับจากวิธีอาร์เอสเอแล้วถอดรหัสสุดท้ายด้วยวิธีดีฟฟี-เฮลล์แมน กุญแจสาธารณะที่ต้องใช้จึงมี 2 ชุด และเลือกใช้ค่ามอดุลัสเป็น $n = pq$ โดย p และ q เป็นจำนวนเฉพาะขนาดใหญ่ที่มีคุณสมบัติตามสมการ $p = 2p' + 1$ และ $q = 2q' + 1$ ซึ่งค่า p' และ q' เป็นจำนวนเฉพาะ

การเลือกวิธีเข้ารหัสแบบดีฟฟี-เฮลล์แมนเป็นชั้นแรกเนื่องจากข้อจำกัดของตัวก่อกำเนิดที่ใช้ยกกำลังแล้วมอดุโลด้วย n เพราะในกรณีที่ค่ามอดุลัสเป็น p ตัวก่อกำเนิดจะสามารถให้ผลลัพธ์เป็นค่าระหว่าง 1 ถึง $p - 1$ ได้ แต่ในกรณีที่มอดุลัสเป็น n ไม่มีตัวก่อกำเนิดค่าใดมีคุณสมบัตินี้ แต่ตัวก่อกำเนิดที่ดีจะสามารถให้ผลลัพธ์ได้หลากหลายที่สุด ดังนั้นในการถอดรหัสชั้นสุดท้ายด้วยวิธีดีฟฟี-เฮลล์แมนจึงมีวัตถุประสงค์เพื่อเปรียบเทียบผลลัพธ์ที่ได้จากการคำนวณ $X^m = g^s \bmod n$ เท่านั้น

ในการสร้างลายมือชื่อจะใช้ข้อมูลลับ 4 ค่า ได้แก่ กุญแจลับ x สำหรับการเข้ารหัสวิธี ดีฟฟี-เฮลล์แมน, กุญแจลับ d สำหรับการเข้ารหัสวิธีอาร์เอสเอ, p และ q ซึ่งเป็นตัวประกอบของมอดุลัส n แล้วเปิดเผยกุญแจสาธารณะเป็น X สำหรับการถอดรหัสวิธีดีฟฟี-เฮลล์แมน, e สำหรับการถอดรหัสวิธีอาร์เอสเอ และค่ามอดุลัส n

3.2 การพัฒนาวิธีสร้างลายมือชื่อ

การสร้างลายมือชื่อวิธีนี้พัฒนาขึ้นโดยอาศัยปัญหาพื้นฐานทางคณิตศาสตร์ 2 ประการคือ ความยากในการหาจำนวนเฉพาะที่เป็นองค์ประกอบของจำนวนเต็มบวกค่าหนึ่ง และความยากของปัญหาดีสครีตลอการิทึม ซึ่งมีสมมติฐานของปัญหาดังนี้

<u>สมมติฐานของปัญหาที่ 1</u> กำหนดให้ N เป็นผลคูณของจำนวนเฉพาะขนาดใหญ่ 2 ค่า หากไม่ทราบจำนวนเฉพาะที่เป็นตัวประกอบนั้นแล้ว เป็นไปไม่ได้ที่จะทำการคำนวณเพื่อหาจำนวนเฉพาะใดๆ ซึ่งสามารถหาร N ได้ลงตัว

<u>สมมติฐานของปัญหาที่ 2</u> กำหนดให้ g เป็นตัวก่อกำเนิดซึ่งเป็นสมาชิกในกลุ่ม Z_p หากให้ค่า X ซึ่ง $g^x \bmod p = X \bmod p$ แล้ว เป็นไปไม่ได้ที่จะคำนวณย้อนกลับเพื่อหาค่า x ซึ่ง x เป็นจำนวนเฉพาะที่มีขนาดใหญ่

นอกจากนี้ในขั้นตอนการเข้ารหัสได้เลือกใช้ฟังก์ชันแฮชแบบป้องกันการชนของผลลัพธ์เพื่อเพิ่มคุณสมบัติการตรวจสอบได้ ดังสมมติฐานการคำนวณคือ

<u>สมมติฐานของปัญหาที่ 3</u> กำหนดให้ $\mathcal{H}$ เป็นแฟมิลี่ของฟังก์ชันแฮชที่มีคุณสมบัติป้องกันการชนของค่าแฮชแบบ (t, ε) ซึ่งภายในระยะเวลา t ไม่มีผู้เป็นปฏิปักษ์รายใดสามารถใช้โอกาส (Advantage) ε ในการทำลายคุณสมบัติป้องกันการชนกันของ $\mathcal{H}$ ได้

ในการสร้างลายมือชื่อวิธีนี้กำหนดให้ $\mathcal{H} = \{H_k\}$ เป็นแฟมิลีของฟังก์ชันแฮชที่มีคุณสมบัติป้องกันการชนของค่าแฮช โดยกำหนด $k \in \mathcal{K}$ เป็นกุญแจสำหรับฟังก์ชัน $H_k : \{0, 1\}^* \rightarrow \{0, 1\}^h$ ค่าแฮชที่เป็นไปได้ทั้งหมดมีจำนวน 2^h ซึ่ง $\phi(n) \geq 2^h$ ดังนั้นค่าแฮชที่ได้จะมีค่าอยู่ในกลุ่ม $Z_{\phi(n)}$ โดยมีค่าอยู่ในช่วง $[0, \phi(n))$

กำหนดใช้พารามิเตอร์ h, ℓ และ ℓ' ซึ่ง $\ell+1 < \ell'$ โดย ℓ และ ℓ' เป็นค่าพารามิเตอร์ที่ปลอดภัย

การสร้างลายมือชื่อที่พัฒนาในงานวิจัยนี้มีรายละเอียดกระบวนการดังต่อไปนี้

[1] <u>การสร้างกุญแจ</u>

a. สุ่มเลือกจำนวนเฉพาะขนาด ℓ' บิต 2 ค่าคือ p และ q ซึ่ง

$$p \leftarrow 2p' + 1$$
$$q \leftarrow 2q' + 1$$

โดยที่ p' และ q' ก็เป็นจำนวนเฉพาะที่มีขนาดใหญ่

b. คำนวณ $n \leftarrow pq$

c. เลือกจำนวนฉพาะเพื่อกำหนดเป็นค่า g โดยที่ g จะต้องไม่อยู่ในกลุ่มส่วนตกค้างกำลังสอง (ดูหัวข้อ 2.3.1 บทที่ 2 ประกอบ) ของค่ามอดุลัส n และเป็นตัวก่อกำเนิดที่ให้ผลลัพธ์เป็นค่าของสมาชิกในกลุ่ม n ได้หลากหลายที่สุด

d. สุ่มเลือกค่ากุญแจแฮช $k \in \mathcal{K}$

e. เลือกจำนวนเต็มคี่เป็นค่า e กำหนดเป็นกุญแจสาธารณะที่ 1 โดยที่ค่า e จะต้องไม่มีตัวประกอบร่วมกับค่า $p'q'$ แล้วคำนวณ

$$d \leftarrow e^{-1} \bmod \phi(n)$$

ผลลัพธ์ d ที่ได้จะต้องไม่มีตัวประกอบร่วมกับ $\phi(n)$ เพื่อกำหนดใช้เป็นกุญแจลับที่ 1

f. กำหนดชุดกุญแจสาธารณะ (pk) เป็น (n, g, k, e) และชุดกุญแจลับ (sk) เป็น (p, q, d, k)

[2] <u>การสร้างลายมือชื่อ</u> (sk, m)

เป็นการสร้างลายมือชื่อเพื่อรับรองข้อความ m ด้วยกุญแจลับ

a. $(p, q, d, k) \leftarrow$ sk

b. สุ่มเลือกจำนวนเต็ม 1 ค่าเป็นค่า x ซึ่ง $x \in Z_{\phi(n)}$ และต้องไม่มีตัวประกอบร่วมกับ $\phi(n)$ กำหนดเป็นกุญแจลับที่ 2 แล้วคำนวณ

$$X \leftarrow g^x \bmod n$$

ใช้ค่า X เป็นกุญแจสาธารณะที่ 2

c. คำนวณ

$$C \leftarrow H_k(m\|e) \in \{0, 1\}^h$$

และ $$C' \leftarrow H_k(m\|X) \in \{0, 1\}^h$$

จะได้ C และ C' มีค่าอยู่ในกลุ่ม $Z_{\phi(n)}$

d. คำนวณ

$$x(C + C')) \bmod \phi(n)$$

ผลลัพธ์ที่ได้ต้องอยู่ในกลุ่ม Z_n^* แต่หากผลลัพธ์ไม่มีคุณสมบัติดังกล่าวให้ทำการเลือกค่า $x \in Z_{\phi(n)}$ ใหม่แล้วทำการคำนวณตามขั้นตอน c. และ d. อีกครั้ง

e. ทำการเข้ารหัสเพื่อสร้างลายมือชื่อ โดยคำนวณ

$$z \leftarrow ((x(C + C')) \bmod \phi(n))^d \bmod n$$

ลายมือชื่อ s ได้แก่ (X, z)

[3] <u>การพิสูจน์ลายมือชื่อ</u> (pk, m, s)

สามารถทำการพิสูจน์ลายมือชื่อ s ซึ่งใช้รับรองข้อความ m ได้ดังนี้

a. $(n, g, k, e) \leftarrow$ pk

b. $(X, z) \leftarrow s$

c. คำนวณ

$$C \leftarrow H_k(m\|e) \in \{0, 1\}^h$$

และ $$C' \leftarrow H_k(m\|X) \in \{0, 1\}^h$$

จะได้ C และ C' มีค่าอยู่ในกลุ่ม $Z_{\phi(n)}$

d. คำนวณ

$$y \leftarrow z^e \bmod n$$

e. ตรวจสอบว่า

$$g^{y} = X^{(C+C')} \bmod n$$

ข้อเสนอแนะในการใช้งาน

ในขั้นตอนที่ [2] – b ซึ่งเป็นการสุ่มเลือกค่า x ในกรณีใช้เป็นกุญแจชุดที่สองแบบใช้ครั้งเดียว มีข้อเสนอแนะว่าผู้เป็นเจ้าของกุญแจลับไม่จำเป็นต้องเลือกด้วยวิธีการสุ่มเท่านั้น แต่สามารถทำการเลือกได้ตามความต้องการ เพียงแต่ค่าที่ใช้แล้วห้ามไม่ให้ใช้ซ้ำอีก

อย่างไรก็ตามแม้ว่าค่า x ซึ่งกำหนดเป็นกุญแจแบบใช้ครั้งเดียวจะถูกใช้ซ้ำก็ไม่ได้ลดประสิทธิภาพด้านความปลอดภัยลงแต่อย่างใด แต่กลับช่วยลดภาระในการจัดการกุญแจแบบใช้ครั้งเดียวทำให้การใช้งานการสร้างลายมือชื่อวิธีนี้ทำได้สะดวกขึ้น

ดังนั้นการใช้งานกุญแจสาธารณะชุดที่ 2 (x, X) สามารถเลือกใช้แบบใช้ครั้งเดียว หรือจะกำหนดใช้เพียงชุดเดียวแล้วใช้ซ้ำก็ได้ขึ้นอยู่กับความเหมาะสมในการงานของผู้ใช้แต่ละราย

3.3 ทดสอบการคำนวณและผลการทดสอบ

ผู้วิจัยเลือกใช้โปรแกรม MATLAB สำหรับทดสอบการคำนวณ (ดูชุดคำสั่งที่ภาคผนวก ก. ประกอบ) เพื่อแสดงให้เห็นตัวอย่างการใช้งานโปรโตคอล การทดสอบจะเป็นการคำนวณในขั้นตอนสร้างกุญแจ, ขั้นตอนสร้างลายมือชื่อและขั้นตอนพิสูจน์ลายมือชื่อ โดยเริ่มจากการเลือกค่า p และ q เพื่อสร้างค่ามอดุลัส n และเลือกค่า e เพื่อสร้างชุดกุญแจสาธารณะชุดที่ 1 (e, d) แล้วเลือกค่า g สำหรับใช้เป็นตัวก่อกำเนิดในการสร้างกุญแจสาธารณะชุดที่ 2 โดยค่าที่เลือกไว้ข้างต้นจะกำหนดใช้เป็นค่าคงที่ แล้วมีตัวแปร x_i สำหรับคำนวณสร้างกุญแจสาธารณะชุดที่ 2 (x_i , X_i) และตัวแปร C_i (สมมติใช้แทนผลรวมที่ได้จาก $C + C'$) เป็นผลลัพธ์ที่ได้จากการหาค่าแฮชของข้อความ m_i เพื่อคำนวณหาผลลัพธ์ z_i ซึ่งเป็นลายมือชื่อ แล้วคำนวณเปรียบเทียบผลลัพธ์ที่ได้จากการถอดรหัสเพื่อพิสูจน์ลายมือชื่อ ลายมือชื่อ z_i ที่ถูกต้องย่อมให้ค่า $g^{y} = X_i^{C_i} \bmod n$

กำหนดค่าพารามิเตอร์และตัวแปรต่างๆ ดังนี้

- ค่าพารามิเตอร์สำหรับจำนวนเฉพาะ $\ell' = 20$ ($2^{20} = 1048576$)
- เลือกจำนวนเฉพาะ $p = 587$ ซึ่งมีจำนวนเฉพาะ $p' = 293$ เป็นตัวประกอบตามคุณสมบัติ $p = 2p' + 1$
- เลือกจำนวนเฉพาะ $q = 983$ ซึ่งมีจำนวนเฉพาะ $q' = 491$ เป็นตัวประกอบตามคุณสมบัติ $q = 2q' + 1$
- คำนวณหาค่า $n = 587 \times 983 = 577021$
- และได้ค่า $\phi(n) = (587-1) \times (983-1) = 575452$
- ค่าพารามิเตอร์สำหรับค่าแฮช $h = 19$ ($2^{19} = 524288 < \phi(n)$)

การสร้างกุญแจ

1. เลือกจำนวนเฉพาะ $g = 149$
2. สุ่มเลือกค่ากุญแจแฮช $k \in \mathcal{K}$
3. เลือกจำนวนคี่ขนาดไม่เกิน ℓ' บิต $e = 23$
4. คำนวณหา $d \leftarrow 23^{-1} \bmod 575452 = 75059$ ตรวจสอบผลลัพธ์ d ที่ได้ไม่มีตัวประกอบร่วมกับ $\phi(n)$
5. กำหนดชุดกุญแจสาธารณะ (pk) เป็น (n, g, k, e) และชุดกุญแจลับ (sk) เป็น (p, q, d, k)

การสร้างลายมือชื่อ (sk, m)

1. $(p, q, d, k) \leftarrow$ sk
2. เลือกค่า x ซึ่ง $x \in Z_{\phi(n)}$ กำหนดเป็นกุญแจลับที่ 2 แล้วคำนวณ $X \leftarrow g^x \bmod n$ กำหนดค่า X เป็นกุญแจสาธารณะที่ 2
3. สุ่มเลือกจำนวนเต็มจาก $Z_{\phi(n)}$ เพื่อกำหนดเป็นค่า C และ C ' แล้วทดสอบผลลัพธ์จากการคำนวณ $x(C + C')) \bmod \phi(n) \in Z_n^*$ หากผลทดสอบไม่อยู่ในกลุ่ม Z_n^* ให้ทำการเลือกค่า $x \in Z_{\phi(n)}$ ใหม่แล้วทดสอบซ้ำ
4. ทำการเข้ารหัสเพื่อสร้างลายมือชื่อ โดยคำนวณ $z \leftarrow ((x(C + C')) \bmod \phi(n))^d \bmod n$

การพิสูจน์ลายมือชื่อ (pk, m, s)

1. $(n, g, k, e) \leftarrow$ pk
2. คำนวณ $y \leftarrow z^e$
3. ตรวจสอบว่า $g^y = X^{(C+C')} \bmod n$

ตัวอย่างผลลัพธ์ที่ได้จากการทดสอบการใช้งานโปรโตคอลแสดงไว้ในตารางที่ 3.2

ตารางที่ 3.2 แสดงผลลัพธ์จากการทดสอบคำนวณตามโปรโตคอล

x	X	C	xC	z	y	g^y	$X^{(C+C')}$
76	241340	39446	120636	57566	120636	435220	435220
109	414530	522714	6078	274787	6078	334514	334514
584	358534	83	48472	17983	48472	279758	279758
4707	42687	183659	154009	5466	154009	317442	317442
7422	257608	568256	108324	366011	108324	69156	69156
10622	374276	901642	569140	403360	569140	201693	201693
29304	211779	897083	321968	120591	321968	442566	442566
60004	523202	1054581	74596	566449	74596	460330	460330

ตารางที่ 3.2 แสดงผลลัพธ์จากการทดสอบคำนวณตามโปรโตคอล (ต่อ)

x	X	C	xC	z	y	g^y	$X^{(C+C')}$
94511	391274	11303	218921	188598	218921	34084	34084
129903	456172	517913	57311	18601	57311	264254	264254
145662	5965	720759	8222	419598	8222	253772	253772
346718	392236	6581	83978	544259	83978	30126	30126
458711	419951	971858	441542	181803	441542	505116	505116

บทที่ 4
การวิเคราะห์และพิสูจน์ความปลอดภัย

ตามทฤษฎีสารสนเทศ (Information Theory) ในงานวิทยาการเข้ารหัสลับเชื่อว่าการเข้ารหัสด้วยวิธี One-time pad ซึ่งเป็นการเข้ารหัสแบบสมมาตรที่มีขนาดของกุญแจเท่ากับขนาดของข้อความและใช้กุญแจลับนั้นเพียงครั้งเดียวเป็นวิธีการเข้ารหัสที่มีความปลอดภัยมากที่สุด ส่วนการเข้ารหัสวิธีอื่นมักมีจุดอ่อนที่สามารถโจมตีเพื่อเปิดเผยข้อมูลลับได้หากผู้เป็นปฏิปักษ์มีจำนวนข้อมูลสำหรับใช้ในการวิเคราะห์ที่มากพอ

แม้ว่าการเข้ารหัสวิธีอื่นจะไม่สามารถให้ความปลอดภัยที่สมบูรณ์ได้ แต่เป้าหมายที่สำคัญของการเข้ารหัสคือ การที่ความลับของการเข้ารหัสยังคงเป็นความลับ โดยมีความเป็นไปได้น้อยมากที่ความลับจะถูกเปิดเผย (Reasonable Probability of Success) และการจะเปิดเผยความลับได้จำเป็นต้องใช้ระยะเวลาที่นานมากพอ (Reasonable Time)

ในบทนี้จะอธิบายการวิเคราะห์และพิสูจน์ความปลอดภัยของการสร้างลายมือชื่อที่พัฒนาขึ้นซึ่งมีรายละเอียดดังนี้

4.1 การวิเคราะห์ต้นทุน

เนื่องจากวิทยาการเข้ารหัสลับมีพื้นฐานมาจากการคำนวณทางคณิตศาสตร์ ความซับซ้อนของสมการที่ใช้จึงมีผลต่อต้นทุนในการคำนวณนั่นก็คือทรัพยากรที่ต้องใช้ในการประมวลผลได้แก่ เวลา (T: Time), หน่วยความจำ (S: Space) และพลังงาน (P: Power) ซึ่งตัวแปรทั้ง 3 ค่านี้จะแปรผันตามองค์ประกอบของสมการ เช่น การการบวกย่อมมีความซับซ้อนน้อยกว่าการคูณและการคูณย่อมซับซ้อนน้อยกว่าการยกกำลัง นอกจากนี้ขนาดของค่าที่รับเข้าเช่น ขนาดของกุญแจลับ และตัวแปรอื่นๆ ในสมการก็ส่งผลต่อต้นทุนในการคำนวณเช่นกัน

ในการวิเคราะห์ต้นทุนของการสร้างลายมือชื่อที่พัฒนาขึ้นนี้ (อ้างอิงวิธีวิเคราะห์มาจาก [6]) จะใช้วิธีเปรียบเทียบกับต้นทุนของการสร้างลายมือชื่อวิธีอื่นที่ได้ศึกษามา โดยเปรียบเทียบตั้งแต่สมมติฐานที่ใช้, ต้นทุนในขั้นตอนของการสร้างกุญแจสาธารณะชุดที่ 2, ต้นทุนในขั้นตอนสร้างลายมือชื่อ, ต้นทุนในขั้นตอนพิสูจน์ลายมือชื่อ, และขนาดของผลลัพธ์ที่ได้จากขั้นตอนสร้างกุญแจสาธารณะชุดที่ 2 และขั้นตอนสร้างลายมือชื่อ รายละเอียดการเปรียบเทียบแสดงไว้ในตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 แสดงการเปรียบเทียบต้นทุนในการสร้างลายมือชื่อ

วิธีสร้างลายมือชื่อ	สมมติฐาน	การสร้างกุญแจชุดที่ 2	การสร้างลายมือชื่อ	การพิสูจน์ลายมือชื่อ	ขนาดกุญแจชุดที่ 2	ขนาดลายมือชื่อ
วิธีที่พัฒนาขึ้น	อาร์เอสเอและ ดีสครีตลอการิทึม	ยกกำลัง 1 ครั้ง	การคูณ 2 ครั้ง และยกกำลัง 1 ครั้ง	การบวก 1 ครั้ง และยกกำลัง 3 ครั้ง	1 ชุด	1 ชุด $\in Z_n$
วิธี [7] โดยใช้วิธีของ Schnorr เป็นพื้นฐาน	ดีสครีตลอการิทึม	ยกกำลัง 1 ครั้ง	การคูณ 1 ครั้ง และการบวก 1 ครั้ง	ยกกำลัง 2 ครั้ง และการคูณ 1 ครั้ง	1 ชุด	1 ชุด $\in Z_p$
วิธี [7] โดยใช้วิธีของ Okamoto เป็นพื้นฐาน	ดีสครีตลอการิทึม	ยกกำลัง 2 ครั้ง	การคูณ 2 ครั้ง และการบวก 2 ครั้ง	ยกกำลัง 3 ครั้ง และการคูณ 2 ครั้ง	1 ชุด	2 ชุด $\in Z_p$
วิธี [2]	ดีสครีตลอการิทึม	สุ่มเลือก	การเตรียมค่า m สำหรับเข้ารหัสประกอบด้วย การยกกำลัง 2 ครั้ง และการคูณ 1 ครั้ง แต่ไม่ระบุวิธีเข้ารหัส	การเตรียมค่า m สำหรับการพิสูจน์ประกอบด้วย การยกกำลัง 2 ครั้ง และการคูณ 1 ครั้ง แต่ไม่ระบุวิธีเข้ารหัส	2 ชุด	2 ชุด $\in Z_p$
วิธี [3]	ไม่ระบุ	ไม่ระบุ	ไม่ระบุ	ไม่ระบุ	1 ชุด	2 ชุด
วิธี [4]	อาร์เอสเอ	สุ่มเลือก	การคูณ 2 ครั้ง และยกกำลัง 6 ครั้ง	การคูณ 2 ครั้ง และยกกำลัง 4 ครั้ง	2 ชุด	ไม่เกิน Z_n
วิธี [5]	อาร์เอสเอ	ยกกำลัง 1 ครั้ง	ยกกำลัง 1 ครั้ง คูณ 2 ครั้ง และ บวก 1 ครั้ง	ยกกำลัง 6 ครั้ง และ คูณ 2 ครั้ง	1 ชุด	1 ชุด $\in Z_n$ และ 1 ชุด ไม่จำกัดขนาด

หมายเหตุ: 1. ในเปรียบเทียบต้นทุนของการคำนวณนี้ไม่นับรวมต้นทุนในการทำงานของฟังก์ชันแฮชและการคำนวณมอดุเลชัน เนื่องจากต้นทุนส่วนใหญ่เกิดจากการเข้ารหัสและถอดรหัส

2. $Z_n = \{0, \ldots n\text{-}1\}$ คือ กลุ่มซึ่งมีตัวดำเนินการทวิภาคเป็นการบวกมอดุโล n

3. $Z_p = \{0, \ldots, p\text{-}1\}$ คือ กลุ่มซึ่งมีตัวดำเนินการทวิภาคเป็นการบวกมอดุโล p

4.2 การพิสูจน์ความปลอดภัย

กำหนดให้ Σ= (KeyGen, Sign, Verify) เป็นการสร้างลายมือชื่อที่พัฒนาขึ้น

ขั้นตอนสร้างลายมือชื่อ

$$C_i = H_k(m_i \| e) \quad (1.)$$

$$C'_i = H_k(m_i \| X_i) \quad (2.)$$

$$z_i = ((x_i C_i + x_i C'_i) \bmod \phi(n))^d \bmod n \quad (3.)$$

ขั้นตอนพิสูจน์ลายมือชื่อ

$$z_i^e = ((x_i C_i + x_i C'_i) \bmod \phi(n) \quad (4.)$$

$$g^{z_i^e} = X^{(C_i + C'_i)} \quad (5.)$$

Claim ที่ 1 การสร้างลายมือชื่อ Σ มีคุณสมบัติปลอมแปลงได้ยาก (t, q, ε) คือมีความเป็นไปได้เท่ากับ ε ที่ผู้เป็นปฏิปักษ์จะสามารถปลอมแปลงลายมือชื่อได้ภายในเวลา t โดยอาศัยข้อมูลจำนวน q เป็นข้อมูลเบื้องต้นสำหรับการปลอมแปลงลายมือชื่อ ซึ่งคุณสมบัตินี้ขึ้นอยู่กับปัจจัยดังต่อไปนี้

1. ความปลอดภัยที่สามารถป้องกันการปลอมแปลงได้ (t, q, ε)
2. คุณสมบัติป้องกันการชนกันของค่าแฮช (t, ε)
3. สมมติฐานอาร์เอสเอ (t, ε)
4. สมมติฐานอาร์เอสเอที่แข็งแกร่ง (t, ε)
5. สมมติฐานดีสคีตลอการิทึม (t, ε)

การพิสูจน์ Claim ที่ 1 กำหนดให้ผู้เป็นปฏิปักษ์โจมตีคุณสมบัติปลอมแปลงได้ยาก (t, q, ε) ของการสร้างลายมือชื่อ โดยผู้เป็นปฏิปักษ์จะได้รับชุดกุญแจสาธารณะ (n, g, k, e) ผู้เป็นปฏิปักษ์สามารถถามขอลายมือชื่อ (s) ของข้อความ (m)ได้ทั้งหมด q ครั้ง

ให้

$$C_i = H_k(m_i \| e)$$

$$C'_i = H_k(m_i \| X_i)$$

และ

$$z_i = ((x_i C_i + x_i C'_i) \bmod \phi(n))^d \bmod n$$

สำหรับ $i = 1, \ldots, q$

และให้ $s = (X, z)$ เป็นลายมือชื่อที่ผู้เป็นปฏิปักษ์ทำการปลอมแปลง โดย

$$C = H_k(m \| e)$$

$$C' = H_k(m \| X)$$

และ

$$z = ((xC + xC') \bmod \phi(n))^d \bmod n$$

แล้วทำการพิจารณาความปลอดภัยจากการปลอมแปลงใน 3 รูปแบบคือ

รูปแบบที่ 1 การปลอมแปลงที่ $m \neq m_i$ สำหรับทุกค่า $i \in \{1, \ldots, q\}$

รูปแบบที่ 2 การปลอมแปลงที่ $(C + C') = (C_i + C'_i)$ และ $z = z_i$

รูปแบบที่ 3 การปลอมแปลงที่ $(C + C') = (C_i + C'_i)$ แต่ $z \neq z_i$

ผู้เป็นปฏิปักษ์จะสามารถปลอมแปลงลายมือชื่อได้สำเร็จหากสามารถสร้างลายมือชื่อได้ในกรณีใดๆ จาก 3 รูปแบบข้างต้น โดยการปลอมแปลงรูปแบบที่ 1 เป็นการโจมตีความปลอดภัยที่สามารถป้องกันการปลอมแปลงได้ของการสร้างลายมือชื่อ ผู้เป็นปฏิปักษ์จะสามารถปลอมแปลงรูปแบบที่ 1 ได้ หากสามารถเปิดเผยความลับของสมการ (3.) ในขั้นตอนสร้างลายมือชื่อได้, การปลอมแปลงรูปแบบที่ 2 เป็นการปลอมแปลงลายมือชื่อโดยสามารถทำลายคุณสมบัติป้องกันการชนกันของค่าแฮชในสมการ (1.) และ(2.) ของขั้นตอนสร้างลายมือชื่อได้, การปลอมแปลงรูปแบบที่ 3 เป็นการปลอมแปลงที่สามารถแก้ปัญหาของสมมติฐานดีสครีตลอการิทึมและอาร์เอสเอที่แข็งแกร่งในสมการ (5.) ของขั้นตอนพิสูจน์ลายมือชื่อได้ เพื่อสามารถคำนวณหาค่า z^e ซึ่ง $g^{z^e} = X^{(C_1+C_2)}$ แล้วสามารถหา z' ซึ่ง $z'^e = z^e$ ได้ ซึ่งรายละเอียดการปลอมแปลงมีดังนี้

4.2.1 การปลอมแปลงรูปแบบที่ 1 กรณีที่ $m \neq m_i$ สำหรับทุกค่า $i \in \{1, \ldots, q\}$

<u>กำหนด</u> ให้ ผู้เป็นปฏิปักษ์ ก. เป็นผู้โจมตีคุณสมบัติปลอมแปลงได้ยาก (t, q, ε) ของการสร้างลายมือชื่อและให้ผู้เป็นปฏิปักษ์ ข. โจมตีคุณสมบัติที่สามารถป้องกันการปลอมแปลงได้ (t, q, ε) ข. จะได้รับกุญแจสาธารณะ (e) เป็นข้อมูลแล้วทำการกำหนดค่าตัวแปรที่เหลือสำหรับใช้ในการสร้างลายมือชื่อ ข. จะส่งข้อมูลทั้งหมดให้กับ ก. การปลอมแปลงลายมือชื่อของ ข. จะสำเร็จหาก ก. สามารถสร้างชุดข้อมูล (m, s) ซึ่งสามารถพิสูจน์ได้ว่า s เป็นลายมือชื่อที่ถูกต้องของ m และ m เป็นข้อความใหม่ที่เจ้าของกุญแจลับยังไม่เคยสร้างลายมือชื่อกำกับมาก่อน

$$\mathbf{Adv}^{suf}_{DS}(\text{ก.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}]$$

$$\mathbf{Adv}^{xuf}_{DS}(\text{ข.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}] - \Pr[\text{กรณีที่ } m = m_i]$$

กระบวนการพิสูจน์มีดังนี้

<u>การกำหนดค่า</u> ผู้เป็นปฏิปักษ์ ข. จะกำหนดค่าตัวแปรดังนี้

1. สุ่มเลือกค่า $g \in Z^*_n$ สำหรับใช้เป็นตัวก่อกำเนิด
2. สุ่มเลือกกุญแจแฮช $k \in \mathcal{K}$
3. ส่งชุดข้อมูลกุญแจสาธารณะ (e, g, k)ไปให้ ก.

<u>การซักถามลายมือชื่อ</u> ก. มีโอกาสที่จะซักถามขอลายมือชื่อสำหรับข้อความ m_i ได้ทั้งหมด q ครั้ง $i = \{1, 2, \ldots q\}$ โดย ข. จะทำหน้าที่ตอบข้อซักถามให้แก่ ก. ดังนี้

1. ก. ส่งข้อมูลของข้อความ m_i ที่ต้องการขอลายมือชื่อกำกับให้แก่ ข.
2. ข. สุ่มเลือกค่า x_i ซึ่ง $x_i \in Z_n$ แล้วคำนวณ $X_i \leftarrow g^{x_i} \bmod n$
3. ข. ทำการคำนวณ $C_i \leftarrow H_k(m_i \| e)$ และ $C'_i \leftarrow H_k(m_i \| X_i)$
4. คำนวณหา $x_i\,(C_i + C'_i)$
5. ข. ส่งผลลัพธ์ของ $x_i\ (C_i + C'_i)$ ไปให้เจ้าของกุญแจลับเพื่อขอลายมือชื่อ ลายมือชื่อที่ได้รับจะเป็น (z_i)
6. ข. ส่งข้อมูลลายมือชื่อ (X_i, z_i) ให้แก่ ก.

ผลลัพธ์จากการปลอมแปลง เมื่อ ก. สามารถปลอมแปลงลายมือชื่อได้ผลลัพธ์เป็น (m, X, z) แล้ว ข. สามารถอ้างอิงการปลอมแปลงได้ดังนี้

1. ข. ทำการคำนวณ $C \leftarrow H_k(m \| e)$ และ $C' \leftarrow H_k(m \| X)$
2. ผลลัพธ์จากการปลอมแปลงของ ข. เป็น (m, C, C', z)

ดังนั้นการปลอมแปลงของ ข. จะสำเร็จได้เมื่อ ก. สามารถปลอมแปลงได้สำเร็จ และข้อความ m ที่เข้ารหัสจะต้องเป็นข้อความใหม่ $m \notin \{m_1, \ldots, m_q\}$ ซึ่งโอกาสที่ ข. จะสามารถปลอมแปลงได้สำเร็จเท่ากับ

$$\mathbf{Adv}_{DS}^{suf}(\text{ข.}) \geq \mathbf{Adv}_{DS}^{suf}(\text{ก.}) \wedge \Pr[\text{กรณีที่ } m = m_i]$$

4.2.2 การปลอมแปลงรูปแบบที่ 2 กรณีที่ $(C + C') = (C_i + C'_i)$ และ $z = z_i$

กำหนด ให้ ก. เป็นผู้โจมตีคุณสมบัติปลอมแปลงได้ยาก $(t,\ q,\ \varepsilon)$ ของการสร้างลายมือชื่อ และให้ ข. โจมตีคุณสมบัติป้องกันการชนกันของค่าแฮช $(t,\ \varepsilon)$ ในเบื้องต้น ข. จะได้รับกุญแจแฮช (k) เป็นข้อมูล วัตถุประสงค์ของ ข. คือสามารถหาการชนกันของค่าแฮชได้ ซึ่ง ข. จะทำได้สำเร็จเมื่อ ก.สามารถสร้างลายมือชื่อ (m, z) ซึ่ง $(m, z) \notin \{(m_1, z_1), (m_2, z_2), \ldots, (m_q, z_q)\}$ แล้วมีข้อความ $m \neq m_i$ ที่ให้ผลลัพธ์เป็นลายมือชื่อ $z = z_i$ ได้

$$\mathbf{Adv}_{DS}^{suf}(\text{ก.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}]$$

$$\mathbf{Adv}_{H_k}(\text{ข.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}] \wedge [\text{กรณีที่ } z = z_i] - \Pr[\text{กรณีที่ } m = m_i]$$

การปลอมแปลงกรณีนี้สามารถแบ่งการพิจารณาได้เป็น 2 กรณีคือ

2.1 กรณีที่ $C = C_i$ และ $C' = C'_i$ แล้ว $(C + C') = (C_i + C'_i)$

2.2 กรณีที่ $C \neq C_i$ และ $C' \neq C'_i$ แล้ว $(C + C') = (C_i + C'_i)$

กระบวนการพิสูจน์มีดังนี้

กรณีที่ 2.1 เป็นการโจมตีคุณสมบัติป้องกันการชนกันของผลลัพธ์ของฟังก์ชันแฮช ผู้เป็นปฏิปักษ์จะสามารถปลอมแปลงได้สำเร็จหากสามารถหาข้อความ m_i ซึ่ง

$$m \| e \neq m_i \| e \text{ แต่ } H_k(m \| e) = H_k(m_i \| e)$$

และ $m||X \neq m_i||X$ แต่ $H_k(m||X) = H_k(m_i||X)$ ได้

$\mathbf{Adv}_{Hk}$(ข.) = Pr[ก. ปลอมแปลงได้สำเร็จ] $\wedge$ Pr[กรณีที่ $z = z_i$] – Pr[กรณีที่ $m = m_i$]

$\mathbf{Adv}^{col}_{Hk}$ (ข.) = Pr[ก.ปลอมแปลงได้สำเร็จ] $\wedge$ Pr[กรณีที่ $H_k(m||e) = H_k(m_i||e) \wedge$
$H_k(m||X) = H_k(m_i||X)$] – Pr[กรณีที่ $m = m_i$]

<u>การกำหนดค่า</u> ข. ได้รับ k' เป็นกุญแจแฮช แล้วทำการกำหนดค่าตัวแปรต่างๆ ดังนี้

1. สร้างชุดกุญแจ (e, d) ตามขั้นตอนการสร้างกุญแจ
2. สุ่มเลือกค่า $g \in Z^*_n$ สำหรับใช้เป็นตัวก่อกำเนิด
3. ข.สุ่มเลือกค่า x ซึ่ง $x \in Z_n$ แล้วคำนวณ $X \leftarrow g^x \mod n$
4. ส่งชุดข้อมูลกุญแจสาธารณะ (e, g, k, X)ไปให้ ก.

<u>การซักถามลายมือชื่อ</u> ก. สามารถซักถามลายมือชื่อสำหรับข้อความ m_i ได้ทั้งหมด q ครั้ง $i = \{1, 2, \ldots q\}$ โดย ข. จะทำหน้าที่ตอบข้อซักถามให้แก่ ก. ดังนี้

1. ก. ส่งข้อมูลของข้อความ m_i ที่ต้องการขอลายมือชื่อกำกับให้แก่ ข.
2. ข.คำนวณ $C_i \leftarrow H_k(m_i||e)$ และ $C'_i \leftarrow H_k(m_i||X)$
3. เข้ารหัสเพื่อสร้างลายมือชื่อ $z_i \leftarrow (x(C_i + C'_i) \bmod \phi(n))^d \bmod n$
4. ข. ส่งข้อมูลลายมือชื่อ (z_i) ให้แก่ ก.

<u>ผลลัพธ์จากการปลอมแปลง</u> หาก ก. สามารถปลอมแปลงลายมือชื่อได้ผลลัพธ์เป็น (m, z) ซึ่ง $(m, z) \notin \{(m_1, z_1), (m_2, z_2), \ldots, (m_q, z_q)\}$ ซึ่ง $C = C_i$ และ $C' = C_i'$ สำหรับบางค่า $i \in \{1, 2, \ldots, q\}$ แล้ว ข. สามารถอ้างอิงการปลอมแปลงได้ว่า มีข้อความ m_i และ m ที่ทำให้ฟังก์ชันแฮชซึ่งมีกุญแจ k สามารถเกิดการชนกันของค่าแฮชได้ โดยโอกาสที่ ข. จะสามารถปลอมแปลงได้สำเร็จเท่ากับ

$$\mathbf{Adv}^{col}_{Hk}(\text{ข.}) \geq \mathbf{Adv}^{suf}_{DS}(\text{ก.}) \wedge \Pr[\text{กรณีที่ } C = C_i \wedge C' = C_i'] \wedge \Pr[\text{กรณีที่ } m = m_i]$$

กรณีที่ 2.2 เป็นการโจมตีคุณสมบัติป้องกันการหาค่าตั้งต้น (Preimage Resistance) ของฟังก์ชันแฮช ผู้เป็นปฏิปักษ์จะสามารถปลอมแปลงได้สำเร็จหากสามารถหาข้อความ m_i ซึ่ง

$$m||e \neq m_i||e \text{ แล้ว } H_k(m||e) \neq H_k(m_i||e)$$

และ $m||X \neq m_i||X$ แล้ว $H_k(m||X) \neq H_k(m_i||X)$

แต่ $(H_k(m||e) + H_k(m||X)) = (H_k(m_i||e + H_k(m_i||X)$

$\mathbf{Adv}_{Hk}$ (ข.) = Pr[ก. ปลอมแปลงได้สำเร็จ] $\wedge$ Pr [กรณีที่ $z = z_i$] – Pr[กรณีที่ $m = m_i$]

$\mathbf{Adv}^{pre}_{Hk}$ (ข.) = Pr[ก.ปลอมแปลงได้สำเร็จ] $\wedge$ Pr [กรณีที่ $(H_k(m||e) + H_k(m||X)) =$
$(H_k(m_i||e + H_k(m_i||X))]$ – Pr[กรณีที่ $H_k(m||e) = H_k(m_i||e)$]

การกำหนดค่า ข. ได้รับ k เป็นกุญแจแฮช แล้วทำการกำหนดค่าตัวแปรต่างๆ ดังนี้

1. สร้างชุดกุญแจ (e, d) ตามขั้นตอนการสร้างกุญแจ
2. สุ่มเลือกค่า $g \in Z_n^*$ สำหรับใช้เป็นตัวก่อกำเนิด
3. ข.สุ่มเลือกค่า x ซึ่ง $x \in Z_n$ แล้วคำนวณ $X \leftarrow g^x \bmod n$
4. ส่งชุดข้อมูลกุญแจสาธารณะ (e, g, k, X)ไปให้ ก.

การซักถามลายมือชื่อ ก. สามารถซักถามลายมือชื่อสำหรับข้อความ m_i ได้ทั้งหมด q ครั้ง $i = \{1, 2, \ldots q\}$ โดย ข. จะทำหน้าที่ตอบข้อซักถามให้แก่ ก. ดังนี้

1. ก. ส่งข้อมูลของข้อความ m_i ที่ต้องการขอลายมือชื่อกำกับให้แก่ ข.
2. ข.คำนวณ $C_i \leftarrow H_k(m_i \| e)$ และ $C'_i \leftarrow H_k(m_i \| X)$
3. เข้ารหัสเพื่อสร้างลายมือชื่อ $z_i \leftarrow ((xC_i + xC'_i) \bmod \phi(n))^d \bmod n$
4. ข. ส่งข้อมูลลายมือชื่อ (z_i) ให้แก่ ก.

ผลลัพธ์จากการปลอมแปลง หาก ก. สามารถปลอมแปลงลายมือชื่อได้ผลลัพธ์เป็น (m, z) ซึ่ง $(m, z) \notin \{(m_1, z_1), (m_2, z_2), \ldots, (m_q, z_q)\}$ ซึ่ง $C \neq C_i$ และ $C' \neq C'_i$ แต่ $z = z_i$ สำหรับบางค่า $i \in \{1, 2, \ldots, q\}$ แล้ว ข. สามารถอ้างอิงการปลอมแปลงได้ว่า ข.สามารถหาข้อความ m_i ซึ่งเป็นค่าตั้งต้นของผลลัพธ์ที่ได้จากฟังก์ชันแฮชแบบใช้กุญแจ k ได้ โดยที่ $H_k(m\|e) \neq H_k(m_i\|e)$ และ $H_k(m\|X) \neq H_k(m_i\|X)$ แต่ m_i และ m มีผลรวมของค่าแฮชเท่ากัน โดยโอกาสที่ ข. จะสามารถปลอมแปลงได้สำเร็จเท่ากับ

$$\mathbf{Adv}_{Hk}^{col}(\text{ข.}) \geq \mathbf{Adv}_{DS}^{suf}(\text{ก.}) \wedge \Pr[\text{กรณีที่ } C \neq C_i] \wedge \Pr[\text{กรณีที่ } C' \neq C_i'] \wedge \Pr[\text{กรณีที่ } m = m_i]$$

4.2.3 การปลอมแปลงรูปแบบที่ 3 กรณีที่ $C = C_i$ แต่ $z \neq z_i$

กำหนด ให้ ผู้เป็นปฏิปักษ์ ก. เป็นผู้โจมตีคุณสมบัติปลอมแปลงได้ยาก (t, q, ε) ของการสร้างลายมือชื่อและให้ ข. เป็นผู้ไขปัญหาอาร์เอสเอที่แข็งแกร่งและดีสครีตลอการิทึมเพื่อใช้ในการปลอมแปลงลายมือชื่อ ในเบื้องต้น ข. จะได้รับค่าตัวก่อกำเนิด g ซึ่ง ข. จะทำได้สำเร็จเมื่อ ก. สามารถสร้างลายมือชื่อ (m, z) ซึ่ง $(m, z) \notin \{(m_1, z_1), \ldots, (m_q, z_q)\}$ แล้วมีข้อความ m ซึ่ง $m = m_i$ แต่ $z \neq z_i$

$$\mathbf{Adv}_{DS}^{suf}(\text{ก.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}]$$

$$\mathbf{Adv}_{Hk}(\text{ข.}) = \Pr[\text{ก. ปลอมแปลงได้สำเร็จ}] \wedge \Pr[\text{กรณีที่ } m = m_i] - \Pr[\text{กรณีที่ } z = z_i]$$

กระบวนการพิสูจน์มีดังนี้

การกำหนดค่า ข. ได้รับค่าตัวก่อกำเนิด g และกุญแจสาธารณะ e แล้วทำการกำหนดค่าตัวแปรต่างๆ ดังนี้

1. สร้างชุดกุญแจ (e, d) ตามขั้นตอนการสร้างกุญแจ

2. สุ่มเลือกกุญแจแฮช $k \in \mathcal{K}$

3. ข. สุ่มเลือกค่า x ซึ่ง $x \in Zn$ แล้วคำนวณ $X \leftarrow g^x \bmod n$

4. ส่งชุดข้อมูลกุญแจสาธารณะ (e, g, k, X)ไปให้ ก.

<u>การซักถามลายมือชื่อ</u> ก. สามารถซักถามลายมือชื่อสำหรับข้อความ m_i ได้ทั้งหมด q ครั้ง $i = \{1, 2, \ldots q\}$ โดย ข. จะทำหน้าที่ตอบข้อซักถามให้แก่ ก. ดังนี้

1. ก. ส่งข้อมูลของข้อความ m_i ที่ต้องการขอลายมือชื่อกำกับให้แก่ ข.

2. ข. ทำการคำนวณ $C_i \leftarrow H_k(m_i \| e)$ และ $C'_i \leftarrow H_k(m_i \| X)$

3. เข้ารหัสเพื่อสร้างลายมือชื่อ โดยคำนวณ

$$z \leftarrow ((xC_i + xC'_i) \bmod \phi(n))^d \bmod n$$

4. ข. ส่งข้อมูลลายมือชื่อ (z_i) ให้แก่ ก.

<u>ผลลัพธ์จากการปลอมแปลง</u> เมื่อ ก. สามารถปลอมแปลงลายมือชื่อได้ผลลัพธ์เป็น (m, z) ซึ่ง $(m, z) \notin \{(m_1, z_1), (m_2, z_2), \ldots, (m_q, z_q)\}$ แล้วมี $m = m_i$ แต่ $z \neq z_i$ สำหรับบางค่า $i \in \{1, 2, \ldots, q\}$ แล้ว ข. สามารถอ้างอิงการปลอมแปลงลายมือชื่อสำหรับข้อความ m ได้ว่า ข. สามารถแก้ปัญหาดีสครีตลอการิทึมได้โดยมีค่า y_i ซึ่งทำให้ $g^y = g^{y_i} \bmod n$ และสามารถแก้ปัญหาอาร์เอสเอที่แข็งแกร่งของกรณีเดียวกันได้โดยมีค่า z_i ซึ่งทำให้ค่า y_i เป็นจริง โดยโอกาสที่ ข. จะสามารถปลอมแปลงได้สำเร็จเท่ากับ

$$\mathbf{Adv}_{DL,\,sRSA}(\text{ข.}) \geq \mathbf{Adv}^{suf}_{DS}(\text{ก.}) \wedge \Pr[\text{กรณีที่ } m = m_i] \wedge \Pr[\text{กรณีที่ } z = z_i]$$

4.3 วิเคราะห์การใช้ฟังก์ชันแฮช

ในการพัฒนาวิธีสร้างลายมือชื่อในงานวิจัยนี้เลือกใช้ ฟังก์ชันแฮชแบบ Universal One-way (UOWHF) ซึ่งนำเสนอครั้งแรกโดย Naor และ Yung ในปี 1989 [7] เป็นชุดของฟังก์ชันแฮชที่มีการกำหนดใช้กุญแจ k เป็นดรรชนี เรียก $\mathcal{H}$ เป็นแฟมิลี่ของฟังก์ชันแฮช กล่าวคือ ฟังก์ชันแฮชที่เป็นสมาชิกอยู่ใน แฟมิลี่จะเป็นฟังก์ชันแฮชเดียวกันแต่มีดรรชนี k ซึ่งจะใช้เป็นกุญแจแฮชต่างกัน ในการใช้งาน UOWHF ค่ารับเข้าของฟังค์ชันจะประกอบด้วยกุญแจแฮช k และข้อความ m ที่ต้องการหาค่าแฮช โดยมีสมมติฐานคือ เมื่อการกำหนดให้ k เป็นกุญแจแฮชแล้ว เป็นการยากที่จะสามารถหาข้อความ x และ y ซึ่ง $x \neq y$ ที่ $H_k(x) = H_k(y)$ ได้

แม้ว่าในงานวิจัย [6] ได้พิสูจน์แล้วว่ามีความแข็งแกร่งที่ด้อยกว่าฟังก์ชันแฮชประเภทป้องกันการชนของผลลัพธ์ แบบอื่น แต่การเลือกใช้ฟังก์ชันแฮชนี้ เพื่อแสดงให้เห็นว่าการสร้างลายมือชื่อที่พัฒนาขึ้นนี้มีความแข็งแกร่งมากพอ สามารถป้องกันการโจมตีแบบเลือกข้อมูลได้โดยไม่จำเป็นต้องใช้ฟังก์ชันแฮชที่ดีที่สุดและความปลอดภัยไม่ได้ขึ้นอยู่กับโมเดลแรนดอมออราเคิล อย่างไรก็ตามผู้ใช้งานสามารถเลือกและปรับการใช้งานฟังก์ชันแฮชตามความเหมาะสมของงานได้

การสร้างลายมือชื่อที่พัฒนาขึ้นได้นำ UOWHF มาปรับใช้ในขั้นตอนการเตรียมข้อความต้นฉบับ สำหรับการเข้ารหัสเพื่อสร้างลายมือชื่อ ดังนี้

$$C \leftarrow H_k(m||e) \in \{0, 1\}^k \qquad \text{---------- (1)}$$

$$C' \leftarrow H_k(m||X) \in \{0, 1\}^k \qquad \text{---------- (2)}$$

ในสมการ (1) นำข้อความ m เชื่อมต่อกับกุญแจสาธารณะ e ได้เป็น $m||e$ แล้วใช้ UOWHF ที่มีกุญแจแฮช k เพื่อหาค่าแฮช $C \in \{0, 1\}^k$ ทำเช่นเดียวกันในสมการ (2) แต่เปลี่ยนแปลงค่าที่เชื่อมต่อกับข้อความ m จากกุญแจสาธารณะ e เป็นกุญแจสาธารณะ X แล้วหาผลลัพธ์ค่าแฮช $C' \in \{0, 1\}^k$ แล้วใช้ค่าแฮชที่ได้จากทั้งสองสมการเป็นข้อความต้นฉบับสำหรับการสร้างลายมือชื่อ ดังนั้นในขั้นตอนพิสูจน์ลายมือชื่อผู้พิสูจน์จึงจำเป็นต้องหาค่าแฮช C และ C' ก่อนพิสูจน์ด้วยสมการ

$$g^y = X^{(C+C')} \bmod n$$

ข้อดีของ UOWHF คือ เป็นฟังก์ชันแฮชที่มีต้นทุนในการคำนวณต่ำกว่าฟังก์ชันแฮชประเภทป้องกันการชนของผลลัพธ์แบบอื่น ทำให้ UOWHF มักถูกนำไปใช้เป็นองค์ประกอบหนึ่งของการเข้ารหัสวิธีต่างๆ

การนำ UOWHF มาใช้เป็นองค์ประกอบของการสร้างลายมือชื่อสามารถพิสูจน์ความปลอดภัยดังนี้

<u>Claim ที่ 2</u> การสร้างลายมือชื่อที่พัฒนาขึ้นนี้มีความปลอดภัยภายใต้สมมติฐานอาร์เอสเอ, สมมติฐานดีสครีตลอการิทึม และสมมติฐานของ UOWHF

<u>การพิสูจน์ Claim ที่ 2</u> การพิสูจน์ความปลอดภัยภายใต้สมมติฐานอาร์เอสเอและสมมติฐานดีสครีตลอการิทึมขออ้างอิงการพิสูจน์ใน Claim ที่ 1 ส่วนการพิสูจน์สมมติฐานของ UOWHF สามารถกำหนดรูปแบบการโจมตีได้เป็น 2 รูปแบบ

รูปแบบที่ 1 สำหรับ $1 \le i \le q$, $H(k, m||e) = H(k_i, m_i||e)$ และ $H(k, m||X) = H(k_i, m_i||X)$ แล้ว $(H(k, m||e) + H(k, m||X)) = (H(k_i, m_i||e) + H(k_i, m_i||X))$ และ $z = z_i$

รูปแบบที่ 2 สำหรับ $1 \le i \le q$, $H(k, m||e) \neq H(k_i, m_i||e)$ และ $H(k, m||X) \neq H(k_i, m_i||X)$ แล้ว $(H(k, m||e) + H(k, m||X)) = (H(k_i, m_i||e) + H(k_i, m_i||X))$ และ $z = z_i$

การโจมตีรูปแบบที่ 1 นั้นเป็นการโจมตีคุณสมบัติป้องกันการชนของผลลัพธ์ของฟังก์ชันแฮช หากกำหนดให้ $k = k_i$ แล้วผู้เป็นปฏิปักษ์จะต้องสามารถหาข้อความ m_i ที่สามารถให้ค่าแฮช $H(k_i, m_i||e)$ และ $H(k_i, m_i||X)$ ตรงกันกับค่าแฮช $H(k, m||e)$ และ $H(k, m||X)$ ของข้อความ m ได้ โดยที่ $m \neq m_i$ ซึ่งมีความเป็นไปได้น้อยมาก

ส่วนรูปแบบที่ 2 เป็นการโจมตีคุณสมบัติป้องกันการหาค่าตั้งต้นของ UOWHF ผู้เป็นปฏิปักษ์จะต้องสามารถหาข้อความ m_i ซึ่งเป็นค่าตั้งต้นของผลลัพธ์ $H(k_i , m_i\|e)$ และ $H(k_i , m_i\|X)$ ได้โดยผลรวมของ $(H(k, m\|e) + H(k, m\|X)) = (H(k_i , m_i\|e) + H(k_i , m_i\|X))$ แต่ $H(k, m\|e) \neq H(k_i , m_i\|e)$ โดยโอกาสที่ผู้เป็นปฏิปักษ์จะสามารถหา m_i ที่ให้ผลลัพธ์ $(H(k, m\|e) + H(k, m\|X)) = (H(k_i , m_i\|e) + H(k_i , m_i\|X))$ ซึ่งจะเป็นการทำลายคุณสมบัติป้องกันการหาค่าตั้งต้นได้มีความเป็นไปได้น้อยมาก

บทที่ 5
สรุปผลการวิจัย

งานวิจัยฉบับนี้เป็นการพัฒนาวิธีสร้างลายมือชื่อที่มีคุณสมบัติปลอมแปลงได้ยาก มีความปลอดภัยต่อการโจมตีแบบเลือกข้อมูลได้ โดยการพัฒนาอาศัยวิธีเข้ารหัสพื้นฐาน 2 วิธีได้แก่ การเข้ารหัสแบบอาร์เอสเอและการเข้ารหัสแบบดีสครีตลอการิทึม แล้วเสริมความปลอดภัยให้กับวิธีสร้างลายมือชื่อด้วยการใช้ฟังก์ชันแฮชเพื่อป้องกันการปลอมแปลงทำให้ลายมือชื่อมีคุณสมบัติบูรณภาพสร้างความมั่นใจให้กับผู้รับว่าเป็นข้อความที่ถูกต้องจากผู้ส่ง

จากการกำหนดโครงสร้างของการเข้ารหัสทำให้วิธีสร้างลายมือชื่อนี้มีความปลอดภัยโดยมีปัจจัยสนับสนุน 5 ประการได้แก่ คุณสมบัติความปลอดภัยที่สามารถป้องกันการปลอมแปลงได้ (t, q, ε), คุณสมบัติป้องกันการชนกันของค่าแฮช (t, ε), สมมติฐานอาร์เอสเอ (t, ε), สมมติฐานอาร์เอสเอที่แข็งแกร่ง (t, ε) และสมมติฐานดีสดีตลอการิทึม (t, ε) ด้วยความแข็งแกร่งของปัญหาการแยกตัวประกอบของสมมติฐานอาร์เอสเอสนับสนุนให้การใช้กุญแจสาธารณะชุดที่ 2 สามารถใช้ซ้ำได้อย่างปลอดภัย ในขณะที่งานวิจัยอื่นกำหนดการใช้กุญแจสาธารณะชุดที่ 2 ในลักษณะใช้ครั้งเดียว เพราะการใช้กุญแจสาธารณะชุดที่ 2 ซ้ำส่งผลถึงความปลอดภัยของการเข้ารหัส

การใช้ฟังก์ชันแฮชที่มีคุณสมบัติป้องกันการชนของผลลัพธ์กับชุดข้อมูล $m||e$ และ $m||X$ ซึ่งมาจากข้อความ m เดียวกันช่วยให้ความเกี่ยวพันระหว่างข้อความ m กับค่าแฮช $H_k(m||e)$ และ $H_k(m||X)$ มีความจำเพาะมากขึ้น สนับสนุนคุณสมบัติป้องกันการโจมตีแบบเลือกข้อมูลของการสร้างลายมือชื่อทำให้การโจมตีลายมือชื่อด้วยวิธีนี้ทำได้ยากยิ่งขึ้น

จากการทดสอบการใช้งานโดยคำนวณตามโปรโตคอลด้วยโปรแกรม MATLAB ผลลัพธ์ที่ได้จากการคำนวณในขั้นตอนสร้างลายมือชื่อสามารถคำนวณพิสูจน์ความถูกต้องของลายมือชื่อได้ ไม่พบปัญหาจากการคำนวณใดๆ ดังนั้นจึงสรุปได้ว่าสมการตามโปรโตคอลสามารถใช้งานจริง

จากการวิเคราะห์และพิสูจน์ความปลอดภัยของการสร้างลายมือชื่อพบว่าต้นทุนในการสร้างลายมือชื่อที่พัฒนาขึ้นนี้ไม่มีความแตกต่างจากการสร้างลายมือชื่อของงานวิจัยอื่น และสามารถพิสูจน์ได้ว่าการสร้างลายมือชื่อที่พัฒนาขึ้นนี้มีความปลอดภัยจากการปลอมแปลงใน 3 รูปแบบ คือ

(1) ความปลอดภัยจากการปลอมแปลงในกรณีที่ผู้เป็นปฏิปักษ์สามารถสร้างลายมือชื่อเพื่อรับรองให้กับข้อความใหม่ตามที่ต้องการได้

(2) ความปลอดภัยจากการปลอมแปลงในกรณีที่ผู้เป็นปฏิปักษ์สามารถนำลายมือชื่อที่พิสูจน์ความถูกต้องแล้วไปรับรองให้ข้อความใหม่ โดยสามารถพิสูจน์ความถูกต้องของลายมือชื่อนั้นกับข้อความใหม่ได้

46

(3) ความปลอดภัยจากการปลอมแปลงในกรณีที่ผู้เป็นปฏิปักษ์สามารถกำกับข้อความที่มีลายมือชื่อที่ถูกต้องอยู่แล้วด้วยลายมือชื่อใหม่ โดยสามารถพิสูจน์ความถูกต้องของข้อความนั้นกับลายมือชื่อใหม่ได้

จากการวิเคราะห์และพิสูจน์โปรโตคอลที่พัฒนาขึ้นทั้งด้านความปลอดภัยและการใช้งาน สามารถสรุปได้ว่าการสร้างลายมือชื่อที่พัฒนาขึ้นนี้มีคุณสมบัติปลอมแปลงลายมือชื่อได้ยาก สามารถป้องกันการโจมตีแบบเลือกข้อมูลได้ โดยมีจุดเด่นที่แตกต่างจากวิธีอื่นคือสามารถใช้กุญแจสาธารณะชุดที่ 2 ซ้ำได้โดยไม่ส่งผลเสียหายต่อความปลอดภัยของการเข้ารหัสดังที่ปรากฏในวิธีการอื่น ด้วยเหตุนี้จึงสามารถนำโปรโตคอลดังกล่าวมาประยุกต์ใช้ในการสร้างลายมือชื่อบนเอกสารจริง และใช้เพื่อการวิเคราะห์, พัฒนาและปรับปรุงการสร้างลายมือชื่อต่อไปได้อีกในอนาคต

บรรณานุกรม

[1] Boneh D., Shen E., and Waters B, **Strongly Unforgeable Signatures Based on Computational Diffie-Hellman,** proceedings of PKC '06, LNCS 3958, pp. 229-240, 2006

[2] Julien Cathalo, Beoit Libert, and Jean-Jacques Quisquater. **Cryptanlysis of a Verifiably Committed Signature Scheme Based on GPS and RSA**. ISC 2004, LNCS 3225, pp.52-60, 2004.

[3] Katz, Jonathan, and Lindel, Yehuda. 2008. **Introduction to modern cryptography**. Boca Raton, FL: Chapman & Hall/CRC.

[4] Mao, Wenbo. 2004. **Modern cryptography: theory & practice**. Upper Saddle River, NJ: Prentice-Hall.

[5] Mihir Bellare and Peter Rogaway. **Collision-resistant Hashing: Towards Making UOWHFs practical.** In Advances in Cryptology-Crypto'97, 1997.

[6] Mihir Bellare and Sarah Shoup. **Two-tier signature, strongly unforgeable signature, and Fiat-Shamir without random oracles**. In Proc. Public Key Cryptography, PKC 2007, Beijing, China, LNCS 4450, Springer-Verlag, 2007, pp. 201-216

[7] Moni Naor and Moti Yung. **Universal One-way hash Function and Their Cryptographic Applications**. In 21st Annual ACM Symposium on Theory of Computing, 1989.

[8] National Institute of Standards and Technology (NIST). 2007. **Key Management**. Information Technology Laboratory [Online]. Available : http://csrc.nist.gov/groups/ST/toolkit/key_management.html

[9] Qioung Haung, Duncan S. Wong, Jin Li, Yi-Ming Zhao. **Generic Transformation from weakly to strongly unforgeable signature**. Journal of Computer Science and Technology, 2008, 23(2): 240 – 252

[10] Ralph M., **A digital signature based on a conventional encryption function,** Lecture Notes In Computer Science; Vol. 293, pp 369 – 378, 1987

[11] Ronald Cramer and Victor Shoup. **Signature Scheme Based on the Strong RSA Assumption**. ACM TISSEC, 3(3):161-185, 2000. Extended abstract in Proc. 6th ACM CCS, 1999.

[12] Shneier, Bruce. 1996. **Applied Cryptography**. John Wiley & Son Inc.

ภาคผนวก

ภาคผนวก ก.

ชุดคำสั่งที่ใช้ทดสอบการคำนวณตามขั้นตอนการสร้างลายมือชื่อ

โปรแกรม MATLAB version 7.2.0.232 (R2006a)

```
%-- หาค่ามอดุลัส n และ ϕ(n) --%
>>  p = 587;
>> q = 983;
>> pp = (p-1)/2;
>> qq = (q-1)/2;
>> n = p*q;
>> ppqq = pp*qq;
>> fn = (p-1)*(q-1);
>> ffn = (pp-1)*(qq-1);
>> n , fn
n =
     577021
fn =
     575452

%-- กุญแจสาธารณะชุดที่ 1 --%
>> e = 23;
>> d = e;
>> for i = 1:(ffn-2)
        d  = mod((d*e),ppqq);
    end
>>d
d =
     75059

%-- กุญแจสาธารณะชุดที่ 2 --%
>> g = 149;
>> x = 145662;
>> X = g;
>> for i =1:(x-1)
        X = mod((X*g),n);
    end
```

```
>> X
X =
     5965

%-- การสร้างลายมือชื่อ --%
>> c = 720759;
>> xc = mod((x*c),fn);
>> z = xc;
>> for i = 1:(d-1)
        z = mod((z*xc),n);
    end
>> z
z =
     419598

%-- การพิสูจน์ลายมือชื่อ --%
>> y = z;
>> for i = 1:(e-1)
        y = mod((ze*z),n);
   end
>> gy = g;
>> for i = 1:(y-1)
       gy = mod((gy*g),n);
    end
>> Xc = X;
>> for i = 1:(c-1)
       Xc = mod((Xc*X),n);
   end
>> gy, Xc
gy =
     253772
Xc =
     253772
```

ภาคผนวก ช.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. **W. Tantikittipisut** and N. Premasathian, "**Strongly Unforgeable Signature Scheme without One-Time Key**," To be published in the proceeding of International Conference on Computer and Information Science (ICIS2009), Shanghai, China, June 1-3, 2009.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาววัชรี ตันติกิตติพิสุทธิ์
วัน เดือน ปีเกิด	11 มิถุนายน 2518
ที่อยู่	438 หมู่ 3 ถ.รามคำแหง 164 แขวงคลองสองต้นนุ่น
	เขตลาดกระบัง กรุงเทพฯ 10520
ประวัติการศึกษา	2540 วิทยาศาสตรบัณฑิต สาขาชีววิทยา มหาวิทยาลัยบูรพา
	2542 ศิลปศาสตรบัณฑิต สาขาภาษาอังกฤษ มหาวิทยาลัยรามคำแหง
ประสบการณ์ทำงานและผลงานวิจัย	
ปัจจุบัน	ตำแหน่งรองผู้จัดการทั่วไป บริษัท เองย่งฮง กรุ๊ป จำกัด
พ.ศ. 2547-2549	ตำแหน่งเจ้าหน้าที่การเงิน บริษัทเอเชี่ยนบิสซิเนสคอมมูนิเคชั่น จำกัด
พ.ศ. 2543-2546	ตำแหน่งพนักงานวิจัย ภาควิชาวิทยาภูมิคุ้มกัน
	คณะแพทยศาสตรศิริราชพยาบาล มหาวิทยาลัยมหิดล
พ.ศ. 2542 – 2543	ตำแหน่งพนักงานวิทยาศาสตร์ ภาควิชาจุลชีววิทยา คณะแพทยศาสตร์
	มหาวิทยาลัยศรีนครินทรวิโรฒประสานมิตร