การเปรียบเทียบวัตถุโดยใช้พันธสัญญาแบบคลุมเครือ

OBJECT MATCHING USING FUZZY COMMITMENT

อรรคพล เสถียรจารุรัตน์
AKKAPON SATIENJARURAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2552

KMITL-2009-IT-M-001-005

การเปรียบเทียบวัตถุโดยใช้พันธสัญญาแบบคลุมเครือ

OBJECT MATCHING USING FUZZY COMMITMENT

อรรคพล เสถียนจารุรัตน์

AKKAPON SATIENJARURAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2552

KMITL-2009-IT–M–001-005

OBJECT MATCHING USING FUZZY COMMITMENT

AKKAPON SATIENJARURAT

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

2009

KMITL-2009-IT–M–001-005

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การเปรียบเทียบวัตถุโดยใช้พันธสัญญาแบบคลุมเครือ

Object Matching Using Fuzzy Commitment

นักศึกษา นายอรรคพล เสถียนจารุรัตน์

รหัสประจำตัว 50066542

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.นล เปรมัษเฐียร

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ ดร.ยงยุทธ เพิ่มพูนธนลาภ	
รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์	
รองศาสตราจารย์ ดร.บุญธีร์ เครือตราชู	
รองศาสตราจารย์ ดร.อาริต ธรรมโน	
ดร.นล เปรมัษเฐียร	

วัน / เดือน/ ปี ที่สอบ วันศุกร์ที่ 2 ตุลาคม 2552 เวลา 10.00 น.

สถานที่สอบ ณ ห้อง 328 (ชั้น 3) คณะเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศรับรองแล้ว

(รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์)

คณบดีคณะเทคโนโลยีสารสนเทศ

วันที่.....19.....**เดือน**.....ตุลาคม.....**พ.ศ**.....2552.....

หัวข้อวิทยานิพนธ์	การเปรียบเทียบวัตถุโดยใช้พันธสัญญาแบบคลุมเครือ
นักศึกษา	นาย อรรคพล เสถียนจารุรัตน์
รหัสนักศึกษา	50066542
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2552
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ดร.นล เปรมัษเฐียร

บทคัดย่อ

เทคโนโลยีสารสนเทศได้เข้ามามีส่วนกับการใช้ชีวิตประจำวันของมนุษย์ในปัจจุบัน แต่การนำเทคโนโลยีสารสนเทศเข้ามาใช้ สิ่งที่ต้องคำนึงถึงอย่างมากคือ ความปลอดภัยของการใช้ระบบ การยืนยันตัวตนของผู้ใช้งานเป็นส่วนหนึ่งที่ทำให้ระบบที่มีความสำคัญ เทคโนโลยีชีวภาพ เป็นเทคโนโลยีที่นำเข้ามาใช้ในการระบุตัวตนของผู้ใช้งานที่ให้ความถูกต้องสูง และยากต่อการปลอมแปลง แต่ การจัดเก็บตัวอย่างเทคโนโลยีชีวภาพของผู้ใช้งาน และการนำข้อมูลมาเปรียบเทียบให้ถูกต้อง และปลอดภัย นั้นเป็นสิ่งที่สำคัญมาก เนื่องจากข้อจำกัดของเทคโนโลยีชีวภาพ การรับข้อมูลที่เข้ามาในระบบ อาจจะเกิดความคลาดเคลื่อนของข้อมูล ทำให้การเปรียบเทียบข้อมูลของเทคโนโลยีชีวภาพ เป็นการเปรียบเทียบและยอมรับข้อมูลที่ต่างกันเล็กน้อย ในปริมาณที่ระบบจะยอมรับได้ จึงมีผู้เสนอแนวคิดการ จัดเก็บและเปรียบเทียบข้อมูลชีวภาพ

วิทยานิพนธ์ฉบับนี้นำเสนอเทคนิคการเปรียบเทียบข้อมูลแบบคลุมเครืออีกวิธีหนึ่ง ที่ใช้เทคนิคของ พันธสัญญาแบบคลุมเครือ ซึ่ง เป็นการรูปแบบการเปรียบเทียบข้อมูลแบบหนึ่ง ที่ใช้การเก็บ ข้อมูลไว้สำหรับเปรียบเทียบกับข้อมูลที่ได้รับมาใหม่ ด้วยการใช้เทคนิคของการแก้ไขข้อผิดพลาด ช่วยในการปรับข้อมูลที่ต่างกันเล็กน้อยให้สามารถเปรียบเทียบกันได้ ซึ่งสามารถแก้ปัญหาที่กล่าวมาข้างต้นได้ แต่ยังมีข้อจำกัดในการใช้งานในความปลอดภัยที่ง่ายต่อการถูกโจมตีด้วยข้อมูลทั้งหมดและสามารถทำการเปรียบเทียบข้อมูลที่เป็นจุดว่าเป็นจุดเดียวกันหรือไม่เท่านั้น มาพัฒนาให้มีความปลอดภัยในการทำงานที่มากขึ้นสามารถทำงานได้กับข้อมูลที่มิติ เช่น เส้น หรือรูปภาพ ถึงแม้หลังจากการพัฒนาความเร็วในการทำงานจะลดลงก็ตาม

Thesis Title	Object Matching using Fuzzy Commitment
Student	Mr. Akkapon Satienjarurat
Student ID.	50066542
Degree	Master of Science
Program	Information Technology
Year	2009
Thesis Advisor	Dr. Nol Premasathian

ABSTRACT

Today, as a member of technology driven society, there are many security and privacy related issues, used one which is reliable user authentication. Biometric is a high performance technology used in authentication system but it's data template and matching data need extra caution due to the limitation of the biometric technology. When the data is received, it may differ from the one being stored in the system. When the data are compared, they do not exactly match each other.

Fuzzy commitment is a technique that combines cryptography with error correction in this scheme. The challenge string can differ from the commitment string by some certain number of bits. Since the order of symbols in the string affects the mechanism of the fuzzy commitment scheme, this thesis proposes a matching algorithm that applies the technique of the Fuzzy commitment so that order-invariant fuzzy matching can be achieved in greater dimensions. The examples of using Fuzzy matching in application are given in this thesis.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ขอกราบพระคุณคณาจารย์ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

อรรคพล เสถียนจารุรัตน์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เทคโนโลยีสารสนเทศได้เข้ามามีส่วนร่วมกับการใช้ชีวิตประจำวันของมนุษย์ในปัจจุบันจนไม่สามารถแยกออกได้ เทคโนโลยีสารสนเทศช่วยอำนวยความสะดวกในชีวิตประจำวันแต่การนำเทคโนโลยีสารสนเทศเข้ามาใช้ สิ่งที่ต้องคำนึงถึงอย่างมากคือ ความปลอดภัยของการใช้ระบบการยืนยันตัวตนของผู้ใช้งานเป็นส่วนหนึ่งที่ทำให้ระบบที่มีความสำคัญ เช่นระบบตรวจคนเข้าเมืองระบบการเข้าถึงข้อมูล การเข้าใช้งานระบบทางการเงินหรือการเข้าออกสถานที่สำคัญ

เทคโนโลยีชีวภาพเป็นเทคโนโลยีที่นำเข้ามาใช้ในการระบุตัวตนของผู้ใช้งานที่ให้ความถูกต้องสูงและยากต่อการปลอมแปลง แต่การจัดเก็บตัวอย่างเทคโนโลยีชีวภาพของผู้ใช้งาน และการนำข้อมูลมาเปรียบเทียบให้ถูกต้องและปลอดภัยนั้นเป็นสิ่งที่สำคัญมาก เนื่องจากข้อจำกัดของเทคโนโลยีชีวภาพ ส่งผลให้การรับข้อมูลที่เข้ามาในระบบอาจจะเกิดความคลาดเคลื่อนของข้อมูล ทำให้การเปรียบเทียบข้อมูลของเทคโนโลยีชีวภาพ (Biometric) เป็นการเปรียบเทียบและยอมรับข้อมูลที่ต่างกันเล็กน้อยในปริมาณที่ระบบจะยอมรับได้ จึงมีผู้เสนอแนวคิดการจัดเก็บและเปรียบเทียบข้อมูลชีวภาพ

เทคนิคพันธสัญญาแบบคลุมเครือ เป็นการรูปแบบการเปรียบเทียบข้อมูลแบบหนึ่ง ที่ใช้การสร้างพันธสัญญาบนข้อมูลและเก็บไว้สำหรับเปรียบเทียบกับข้อมูลที่ได้รับมาใหม่ ด้วยการใช้เทคนิคของการแก้ไขข้อผิดพลาดช่วยในการปรับข้อมูลที่ต่างกันเล็กน้อยให้สามารถเปรียบเทียบกันได้ ซึ่งสามารถแก้ปัญหาที่กล่าวมาข้างต้นได้

เทคนิคพันธสัญญาแบบคลุมเครือ (Fuzzy Commitment) ยังมีข้อจำกัดในการใช้งานอยู่สองประการคือ เรื่องความปลอดภัยในการทำงาน อัลกอริทึมมีจุดอ่อนสามารถถูกโจมตีด้วยการใช้ข้อมูลสมาชิกในชุดรหัสสำหรับแก้ไขข้อผิดพลาด (Codeword) ทั้งหมดที่เป็นไปได้ทดสอบเพื่อหาข้อมูลที่เป็นความลับ และการใช้งานสามารถทำการเปรียบเทียบข้อมูลที่เป็นจุดว่าเป็นจุดเดียวกันหรือไม่เท่านั้นไม่สามารถทำงานกับข้อมูลที่เป็นพื้นที่หรือรูปภาพ วิทยานิพนธ์ฉบับนี้จะนำเสนอเทคนิคการเปรียบเทียบข้อมูลแบบคลุมเครืออีกวิธีหนึ่ง ที่ใช้เทคนิคพันธสัญญาแบบคลุมเครือมาพัฒนาเป็นการเปรียบเทียบข้อมูลแบบคลุมเครือที่มีความปลอดภัยในการใช้งานที่มากขึ้น ด้วยการแบ่งข้อมูลความลับออกเป็นส่วนๆ (Secret Splitting) เพื่อเก็บข้อมูลจากนั้นจึงใช้ทฤษฎีเศษเหลือแบบจีน(Chinese Remainder Theorem) หรือค่าคะแนนส่วนใหญ่(Majority Score) ในการตรวจสอบข้อมูลที่เป็นความลับ นอกจากนั้นแล้วจะแสดงการพัฒนาวิธีการให้สามารถ

ทำงานได้กับข้อมูลที่มิติ เช่น เส้น หรือ รูปภาพ รวมถึงตัวอย่างการประยุกต์ ใช้เทคนิคการเปรียบเทียบข้อมูลแบบคลุมเครือในการทำงานกับรูปทรงที่มีความสำคัญไม่เท่ากัน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษา เทคนิคพันธสัญญาแบบคลุมเครือ เพื่อวิเคราะห์หาข้อจำกัดการทำงานของเทคนิคดังกล่าวนำมาพัฒนาและปรับปรุง การเปรียบเทียบข้อมูลแบบคลุมเครือให้สามารถเปรียบเทียบข้อมูลที่ลักษณะสองข้อมูลต่างกันเพียงเล็กน้อยในปริมาณที่ระบบที่ใช้งานจะสามารถยอมรับได้ ความปลอดภัยในการใช้งานจะต้องมีมากเพียงพอ ข้อมูลต้นแบบนั้นสามารถเป็นข้อมูลที่เป็น เส้น รูปภาพ 2 มิติ หรือ 3 มิติ รูปทรงที่ไม่ใช่เรขาคณิต และจะต้องถูกเข้ารหัสเป็นความลับด้วยกระบวนการทางเดียว ที่มีความปลอดภัยไม่สามารถทำการย้อนกลับมาเป็นข้อมูลตั้งต้นได้ รวมถึงการประยุกต์ใช้งานกับระบบตามความเหมาะสม

1.3 สมมติฐานของการศึกษา

ข้อด้อยของระบบเทคโนโลยีชีวภาพคือ การจัดเก็บตัวอย่างของผู้ใช้งานต้องมีการจัดเก็บเป็นความลับ ซึ่งหากเป็นข้อมูลที่เป็นรหัสผ่านเราจะสามารถนำข้อมูลมาผ่านกระบวนการไม่สามารถย้อนกลับ (Hash Function) ได้เพื่อเก็บไว้เปรียบเทียบค่ากับที่ได้รับมาใหม่ได้อย่างถูกต้องในทันที และ ข้อมูลที่จัดเก็บจะไม่สื่อถึงข้อมูลตัวอย่างแต่เทคโนโลยีชีวภาพการรับข้อมูลเข้ามาในระบบในแต่ละครั้งจะเกิดความผิดพลาดซึ่งทำให้ข้อมูลแตกต่างจากข้อมูลตัวอย่างเล็กน้อย แต่เมื่อผ่านกระบวนการไม่สามารถย้อนกลับได้ จะทำให้เกิดความแตกต่างกันอย่างมากไม่สามารถเปรียบเทียบกันได้

การแก้ปัญหาข้างต้นนี้สามารถจะใช้เทคนิคพันธสัญญาแบบคลุมเครือ ซึ่งใช้การใช้เทคนิคของการแก้ไขข้อผิดพลาดของข้อมูลช่วยในการปรับข้อมูลที่ต่างกันเล็กน้อย ให้สามารถเปรียบเทียบกันได้ แต่การใช้งานเทคนิคพันธสัญญาแบบคลุมเครือยังมีจุดอ่อนการใช้งานที่สำคัญเนื่องจากค่าความลับที่ใช้เก็บเพื่อเปรียบเทียบเป็นข้อมูลค่าแฮชของรหัสที่ใช้ในการแก้ไขข้อผิดพลาดของข้อมูล ซึ่งจำนวนสมาชิกของชุดรหัสแก้ไขข้อผิดพลาดมีจำนวนที่คงที่ทำให้ง่ายต่อการถูกโจมตีโดยการใช้ข้อมูลในชุดรหัสแก้ไขข้อผิดพลาดทั้งหมดมาเข้าฟังก์ชันแฮช เพื่อหาค่าที่ถูกต้องและนำมาเข้าสมการปกติเพื่อหาค่าความลับที่ถูกต้องได้ วิทยานิพนธ์นี้จะปรับปรุงพัฒนาวิธีการของเทคนิคพันธสัญญาแบบคลุมเครือ ให้มีความปลอดภัยในการทำงานซึ่งทำให้ผู้ประสงค์ร้ายต้องใช้เวลาที่มากขึ้นจนเป็นไปได้ยากในการหาความลับและพัฒนาเทคนิคการเปรียบเทียบข้อมูลแบบคลุมเครือให้สามารถทำงานได้กับ ข้อมูล 1 มิติและ 2 มิติ

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

วิธีการเปรียบเทียบข้อมูลที่ใช้ในวิทยานิพนธ์นี้ จะมีการแบ่งข้อมูลที่เป็นความลับออกเป็นส่วนๆ ด้วยการสร้างชุดจำนวนเฉพาะสัมพัทธ์เมื่อนำมาหารเอาเศษ จะทำให้มีข้อมูลความลับจำนวนมากขึ้น แต่ละความลับจะถูกเข้ารหัสด้วยข้อมูลในชุดรหัสแก้ไขข้อผิดพลาด เมื่อมีการเปรียบเทียบข้อมูลผู้ใช้งานจะต้องสามารถหาความลับได้จำนวนมากพอที่จะใช้ทฤษฎีเศษเหลือแบบจีนหาค่าความลับที่ถูกต้องได้ และการพัฒนาเทคนิคให้สามารถทำงานได้ในข้อมูลที่เป็นรูปภาพ สามารถทำได้ด้วยการกำหนดจุดของรูปภาพ เป็นจุดอ้างอิงที่ใช้ในการเปรียบเทียบจากนั้นนำข้อมูลนั้นที่ได้มาแปลง ให้อยู่ในลักษณะที่เหมาะสม เพื่อที่จะสามารถใช้เทคนิคของ การแก้ไขข้อผิดพลาดปรับปรุงจุดอ้างอิงที่มีความแตกต่างกันเล็กน้อยให้เหมือนกันและสามารถเปรียบเทียบกันได้ หากมีรูปภาพหรือเส้นมากกว่า 1 ชิ้นขึ้นไประบบจะต้องมีการนำข้อมูลไปเปรียบเทียบกับข้อมูลตัวอย่างทุกชิ้น

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

วิธีการเทคนิคการเปรียบเทียบข้อมูลแบบคลุมเครือ มีการพัฒนาให้มีความปลอดภัยต่อการถูกโจมตีด้วยการใช้สมาชิกของชุดรหัสแก้ไขข้อผิดพลาดที่ใช้สำหรับการแก้ไขข้อผิดพลาดที่เป็นไปได้ทั้งหมดด้วยการที่ในแต่ข้อมูลความลับมีการใช้รหัสในชุดรหัสแก้ไขข้อผิดพลาดหลายค่าและผู้โจมตีต้องหาให้ครบทุกค่า ซึ่งหากเป็นการใช้พันธสัญญาแบบคลุมเครือ ผู้ประสงค์ร้ายสามารถทำการโจมตีได้โดยการนำข้อมูลรหัสในชุดรหัสแก้ไขข้อผิดพลาดที่เป็นไปได้ทั้งหมดมาเข้า ฟังชันแฮช เพื่อหาค่ารหัสในชุดรหัสแก้ไขข้อผิดพลาดที่ถูกต้องที่และสามารถเปรียบเทียบข้อมูลที่มีลักษณะเป็นเส้นหรือรูปภาพได้จากเดิมที่วิธีการพื้นฐานจะสามารถเปรียบเทียบข้อมูลที่เป็นตัวเลขเท่านั้น

1.6 ขอบเขตการวิจัย

ในวิทยานิพนธ์นี้ได้นำเสนอวิธีการเปรียบเทียบข้อมูลที่มีลักษณะแตกต่างกันในปริมาณที่ระบบยอมรับได้ ด้วยการปรับปรุงระบบให้มีความแข็งแกร่งทนต่อการโจมตีได้มากขึ้น พัฒนาอัลกอริทึมให้สามมารถทำงานกับระบบที่สร้างรูปขึ้นมา 2 รูปหากมีส่วนใดส่วนหนึ่งของรูปที่ซ้อนทับกันจะถือว่าข้อมูลที่นับมาเปรียบเทียบกันนั้นถูกต้องการนำเสนอวิธีการจะมีการนำเสนออัลกอริทึมที่ใช้ในการเปรียบเทียบ เส้นตรง สี่เหลี่ยม และการประยุกต์ใช้ในการกำหนดค่าให้ สี่เหลี่ยมแต่ละรูปมีน้ำหนักความสำคัญไม่เท่ากัน

1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย ซึ่งประกอบด้วย เทคนิคการใช้แฮช ฟังก์ชัน (Hash Function) เทคนิคการตรวจจับและแก้ไขความผิดพลาด (Error Correction) ทฤษฎี บทเศษเหลือแบบจีน (Chinese Remainder Theorem) เทคนิคพันธสัญญาแบบคลุมเครือ (Fuzzy Commitment) และ เทคนิคการเก็บข้อมูลแบบคลุมเครือ (Fuzzy Vault)

บทที่ 3 กล่าวถึงการเปรียบเทียบโดยใช้การเปรียบเทียบแบบคลุมเครือ ด้วยการนำทฤษฎี เศษเหลือแบบจีนหรือค่าคะแนนส่วนใหญ่ มาใช้ในการเพิ่มความปลอดภัย รวมถึงการประยุกต์ สำหรับใช้งานกับวัตถุที่เป็นเส้น รูปภาพ 2 มิติโดยในรูปแบบต่างๆ

บทที่ 4 กล่างถึงการวิเคราะห์และพิสูจน์ความปลอดภัย กล่าวถึงการวิเคราะห์การทำงาน ในด้าน ประสิทธิภาพการทำงาน และ ความปลอดภัยในด้านความแข็งแกร่งต่อการถูกโจมตีของ อัลกอริทึม ของการเปรียบเทียบข้อมูลแบบคลุมเครือและพันธสัญญาแบบคลุมเครือเพื่อแสดงให้ เห็นความปลอดภัยที่เพิ่มขึ้นเมื่อมีการใช้การเปรียบเทียบข้อมูลแบบคลุมเครือ

บทที่ 5 บทสรุปผลการวิจัยและข้อเสนอแนะสำหรับการทำวิจัยต่อไปในอนาคต

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในการวิจัย

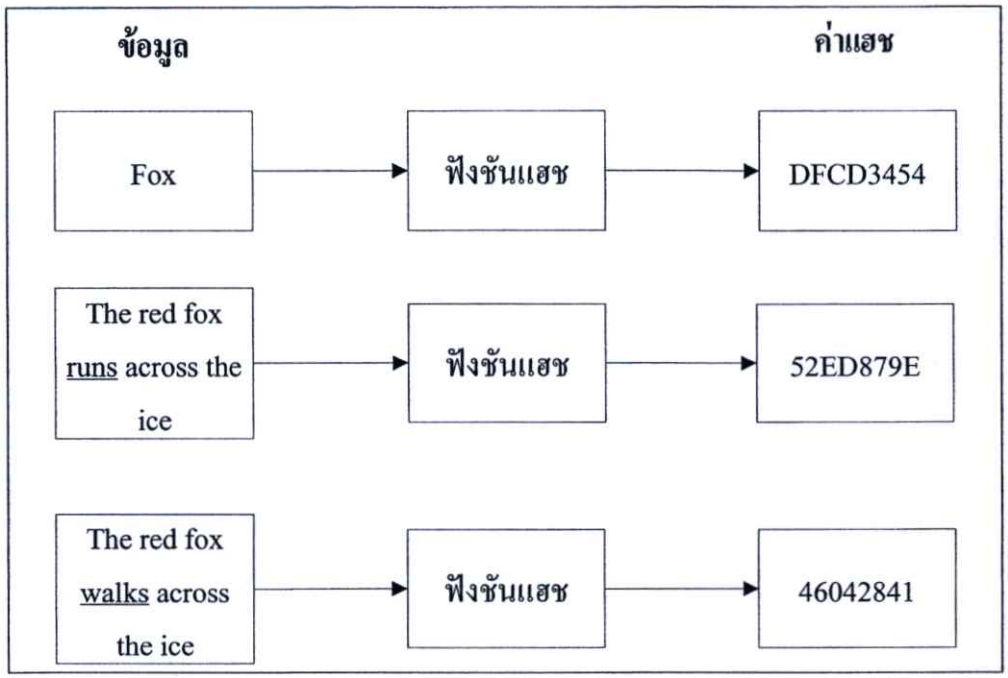
ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆที่เกี่ยวข้องในการวิจัย ซึ่งเนื้อหาในบทนี้จะกล่าวถึงเทคนิคการใช้ฟังก์ชันแฮช (Hash Function) เทคนิคการตรวจจับและแก้ไขข้อผิดพลาด (Error Correction) ทฤษฎีบทเศษเหลือแบบจีน (Chinese Remainder Theorem) อัลกอริทึมพันธสัญญาแบบคลุมเครือ และ อัลกอริทึมการเก็บข้อมูลแบบคลุมเครือ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษาการปรับปรุง การพิสูจน์ความปลอดภัย และประยุกต์ใช้งาน

2.1 เทคนิคการใช้ฟังก์ชันแฮช (Cryptographic Hash) [4] [5] [6] [7]

ฟังก์ชันแฮชทางเดียว (One-way Hash Function) คือวิธีการอย่างหนึ่งซึ่งทำให้ข้อมูลส่วนหนึ่งหรือทั้งหมด ให้กลายเป็นจำนวนเล็กๆ อันหนึ่งอย่างมีปฏิสัมพันธ์ ซึ่งจำนวนดังกล่าวเปรียบได้ว่าเป็น "ลายนิ้วมือ" ของข้อมูล ฟังก์ชันแฮชทางเดียว เป็นที่รู้จักกันในชื่ออื่นด้วย เช่น Message Digest, Digital Fingerprint และ Compression Function คุณสมบัติสำคัญของ ฟังก์ชันแฮชทางเดียว มีดังนี้

1. การคำนวณหาค่าของรหัสแฮช H(M) จากข้อความ M ต้องมีขั้นตอนที่ชัดเจนไม่ซับซ้อน
2. การหาข้อความ M ที่ให้กำเนิดรหัส แฮชดังกล่าวจะต้องเป็นขั้นตอนที่ยากมาก
3. การหาข้อความ ใดๆที่จะทำให้มีค่า แฮชค่าเดียวกันกับ H(M) นั้นเป็นเรื่องที่ยากมาก
4. การหาค่า ข้อมูลสองค่าที่ไม่เท่ากัน M_1 และ M_2 ที่ให้ค่า $H(M_1) = H(M_2)$ เป็นไปได้ยาก

ฟังก์ชันแฮชสามารถนำมาประยุกต์ในงานได้อย่างหลากหลายยกตัวอย่างเช่นใช้ตรวจสอบบูรณภาพของข้อมูล (Integrity) หรือการลงลายมือชื่อดิจิตอลนั้นสามารถกระทำได้โดยตรงกับเอกสารหรือข้อความต้นฉบับ หรืออีกทางเลือกหนึ่งเราสามารถกระทำกับค่าแฮชที่ได้จากเอกสารต้นฉบับก็ได้ การทำโดยการเปรียบเทียบกับเอกสารโดยตรงมีข้อเสียคือระบบจะประมวลผลได้ช้ายิ่งข้อมูลมีมากก็ยิ่งต้องใช้เวลาในการเปรียบเทียบข้อมูลมากแต่หากกระทำกับค่าแฮชที่โดยทั่วไปมักจะมีขนาดเล็กกว่าเอกสารต้นฉบับจะทำให้สามารถประมวลผลได้เร็วขึ้น นอกจากนี้ในกรณีที่ต้องการนำเอกสารไปใช้งานอื่นที่ต้องเก็บเอกสารเป็นความลับ เช่น การบันทึกเวลาเราสามารถใช้ค่าแฮชแทนเอกสารต้นฉบับในกระบวนการที่ต้องการได้ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 การทำงานของฟังก์ชันแฮช

นอกจากนี้ยังมีการประยุกต์ใช้ในการจัดการรหัสผ่านของผู้ใช้งานระบบคอมพิวเตอร์ เนื่องจากในปัจจุบันระบบต่างๆ ในคอมพิวเตอร์จะต้องมีฐานข้อมูลเก็บรายละเอียดผู้ใช้งานระบบ รวมถึงรหัสผ่านของผู้ใช้งานระบบเพื่อใช้ในการตรวจสอบสิทธิการใช้งานระบบ เพราะฉะนั้น การเก็บข้อมูลรหัสผ่านเป็นอักษรในฐานข้อมูลเป็นสิ่งที่ไม่ปลอดภัย เนื่องจากสามารถถูกเปิดเผยรหัสผ่านได้ง่ายแต่หากมีการเป็นเป็นรหัสผ่านที่ผ่านการเข้ารหัสแล้ว ผู้มีกุญแจที่ถูกต้องเท่านั้นที่จะสามารถ ถอดรหัสเพื่อหารหัสผ่านที่ถูกต้องได้ก็จะทำให้ระบบมีความปลอดภัยมากขึ้นแต่ผู้ดูแลระบบยังเป็นผู้ที่สามารถถอดรหัสเพื่อดูรหัสผ่านได้ จึงมีการนำฟังก์ชันแฮชมาใช้ด้วยการเก็บค่าแฮชของรหัสผ่านแทนการเก็บรหัสผ่าน เมื่อผู้ใช้งานเข้าใช้งานระบบจะนำรหัสผ่านมาผ่านฟังก์ชันแฮชและนำค่าที่ได้ เปรียบเทียบกับค่าแฮชที่เก็บไว้ในฐานข้อมูล หากมีค่าแฮชที่ตรงกันก็จะอนุญาตให้ผู้ใช้งานเข้าใช้งานระบบ แต่หากค่าแฮชไม่ตรงกับที่เก็บไว้ในฐานข้อมูลก็จะไม่อนุญาตให้เข้าใช้งาน

ในอดีตจนถึงปัจจุบันได้มีการประดิษฐ์หรือพัฒนาอัลกอริทึมที่ให้ผลตรงตามคุณลักษณะของฟังก์ชันแฮชอยู่หลายวิธี ตัวอย่างของอัลกอริทึม ได้แก่ MD4, MD5 และ SHA-1

2.1.1 อัลกอริทึม MD5

อัลกอริทึม MD4 ได้รับการพัฒนาขึ้นในปี ค.ศ. 1990 โดย Ron Rivest อัลกอริทึมนี้รับข้อความที่มีขนาดแปรเปลี่ยนได้และให้ผลเป็นค่าแฮชขนาด 128 บิตเสมอ ต่อมาในปี ค.ศ.

1991 Ron Rivest ได้ปรับปรุงอัลกอริทึม MD4 ให้มีความปลอดภัยมากขึ้น ใช้ชื่อว่า MD5 ซึ่งให้ค่าแฮชที่มีขนาด 128 บิตเท่าเดิม ในปี ค.ศ. 1994 Van Oorschot and Wiener ได้พิจารณาการค้นหา Collision ของฟังก์ชันแฮชด้วยวิธี Brute-force และประมาณว่าเครื่องที่พัฒนาขึ้นสำหรับค้นหา Collision ที่ใช้อัลกอริทึม MD5 สามารถพบ Collision ได้ภายใน 24 วัน โดยเฉลี่ย การทดลองดังกล่าวทำให้เกิดความกังวลความปลอดภัยของผู้ใช้งาน ทำให้ความนิยมของอัลกอริทึมนี้ลดลง

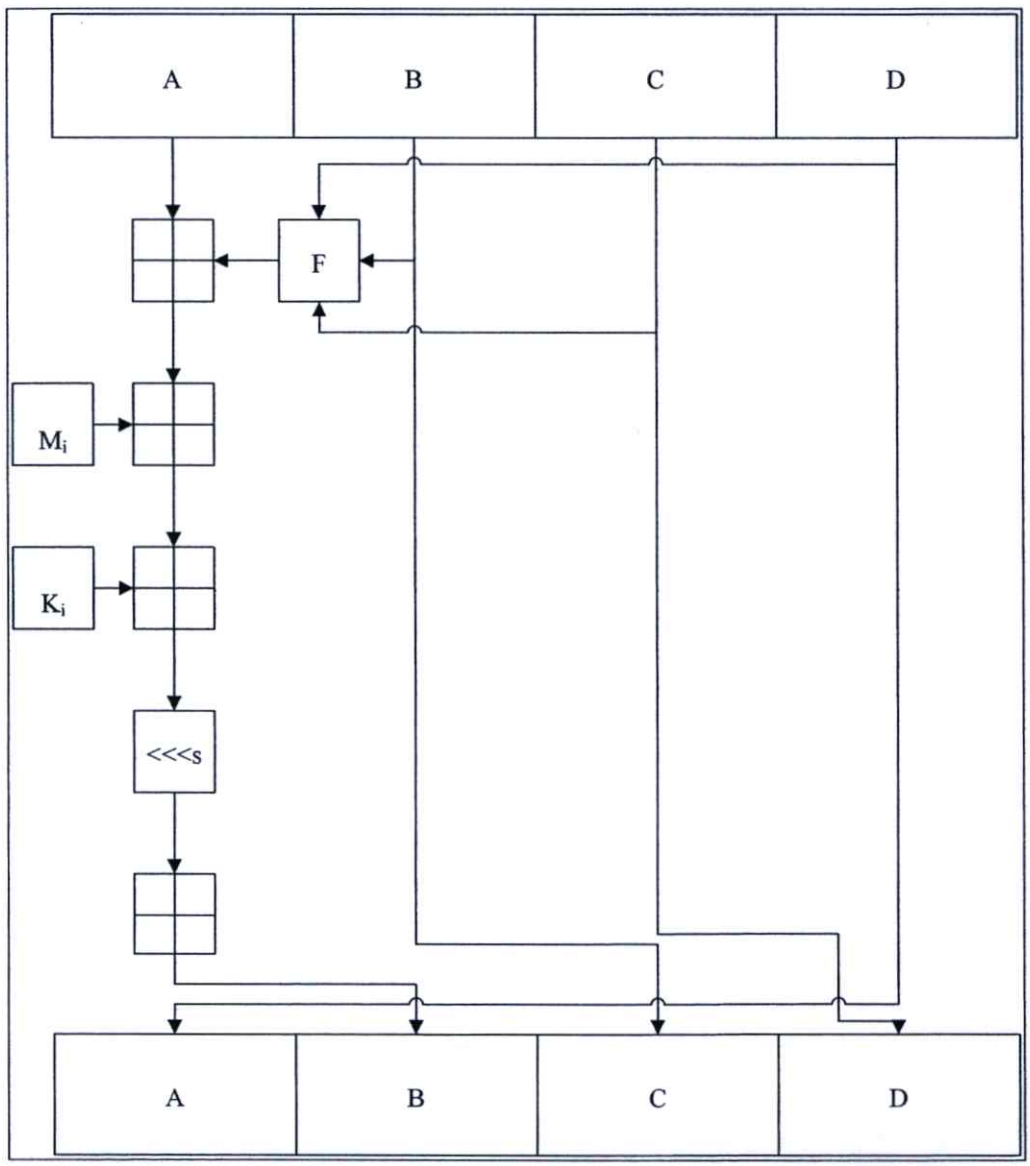
ขั้นตอนการทำงานของ MD5

ขั้นตอนที่ 1 การเพิ่มบิตหลังข้อความ (Padding)

ขั้นตอนที่ 2 การระบุขนาดของข้อความต้นฉบับ (Append Length)

ขั้นตอนที่ 3 แบ่งข้อความออกเป็นบล็อคละ 512 บิต

ขั้นตอนที่ 4 การดำเนินการกับบล็อค ตามรูป 2.2 จำนวน 4 รอบ



รูปที่ 2.2 แสดงการทำงานของแฮชฟังก์ชัน MD5

2.1.2 อัลกอริทึม SHA

อัลกอริทึม SHA ได้รับการพัฒนาโดย NIST (National Institute of Standards and Technology) และ NSA ในปี ค.ศ. 1993 โดยหลักการแล้วมีความปลอดภัยที่มากกว่า MD5 เพราะให้ค่าแฮชที่ยาวกว่าคือ 160 บิต ในการโจมตีด้วยวิธีการใช้ข้อมูลทั้งหมด จะต้องมีการทดสอบค่ามากถึง $2^{160} = 1.5x10^{48}$ กรณี ซึ่งต้องใช้เวลานานมาก ขนาดความยาวของข้อมูล ของอัลกอริทึมต่างต่างของ SHA แสดงตามตารางที่ 2.1 และขั้นตอนการทำงานของ SHA-1 แสดงดังรูป ที่2.3 และ 2.4

ตารางที่ 2.1 แสดงขนาดของข้อมูล Secure Hash Algorithm

อัลกอริทึม	ขนาด ผลลัพธ์ (บิต)	ขนาดเริ่มต้น (บิต)	บล๊อกข้อมูล (บิต)
SHA-0	160	160	512
SHA-1	160	160	512
SHA-256/224	256/224	256	512
SHA-512/384	512/384	512	1024

ในขั้นตอนที่ 1 เพิ่มจำนวนบิต (Padding) โดยจะเพิ่มเป็นจำนวนเท่ากับ 512ลบด้วยเศษที่ได้จากการหาร 512 แล้วลบออก 64 บิต เนื่องจากจะมีการเพิ่มความยาวอีก 64 บิตในขั้นตอนที่ 2 ดังนั้นแม้ว่าบล็อกข้อมูลที่หารด้วย 512 ลงตัว ก็จะต้องมีการเติมบิตด้วยเช่นกัน

ในขั้นตอนที่ 2 จะมีการเพิ่มข้อมูลความยาว 64 บิต โดยจะเป็นข้อมูลที่ระบุความยาวของข้อมูล ก่อนที่จะมีการเติมบิตเข้าไป โดยการเพิ่มข้อมูลความยาวบิตเท่ากับ 64 บิตนี้ จะทำให้ความยาวของข้อมูลที่รวมกับการเติมบิตและเพิ่มอีก 64 บิต จะมีความยาวที่หารด้วย 512 ลงตัวพอดี ซึ่งหมายความว่าจะแบ่งเป็นบล็อกละ 512 บิตได้อย่างลงตัว

ในขั้นตอนที่ 3 จะมีการกำหนดค่าเริ่มต้นของ MD Buffer โดยมีความยาวเท่ากับ 160 บิต โดยบัฟเฟอร์นี้จะเก็บค่าเริ่มต้นของ MD จะแทนด้วยรีจิสเตอร์จำนวน 5 ตัว ตัวละ 32 บิต โดยมีชื่อเป็น A, B, C, D และ E โดยมีค่าเริ่มต้นคงที่ดังนี้

A = 0X67452301

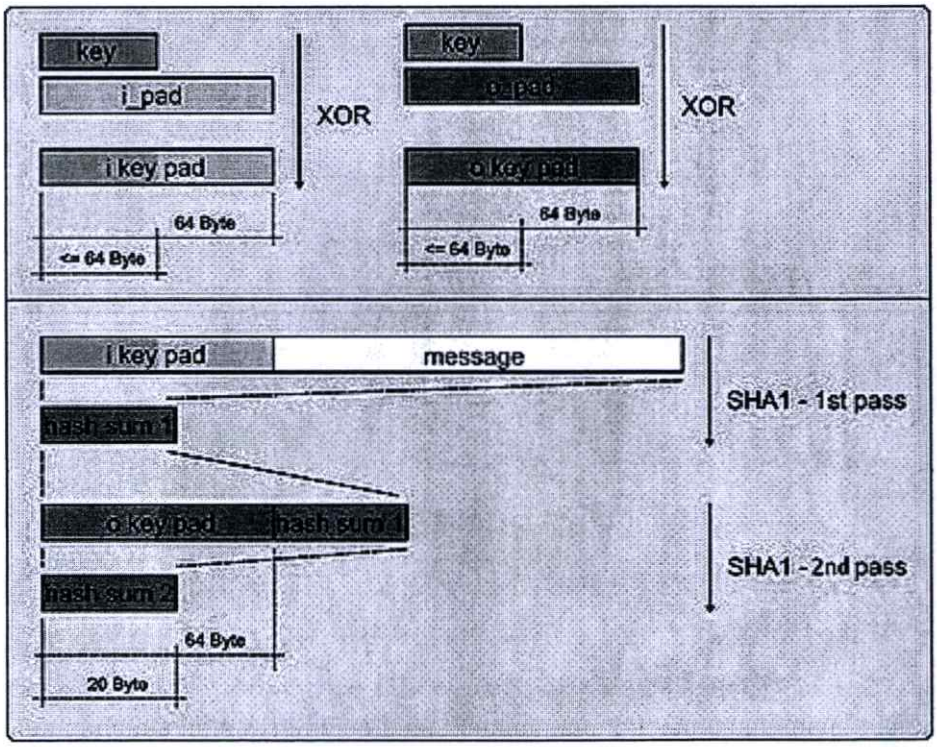
B = 0XEFCDAB89

C = 0X 98BADCFE

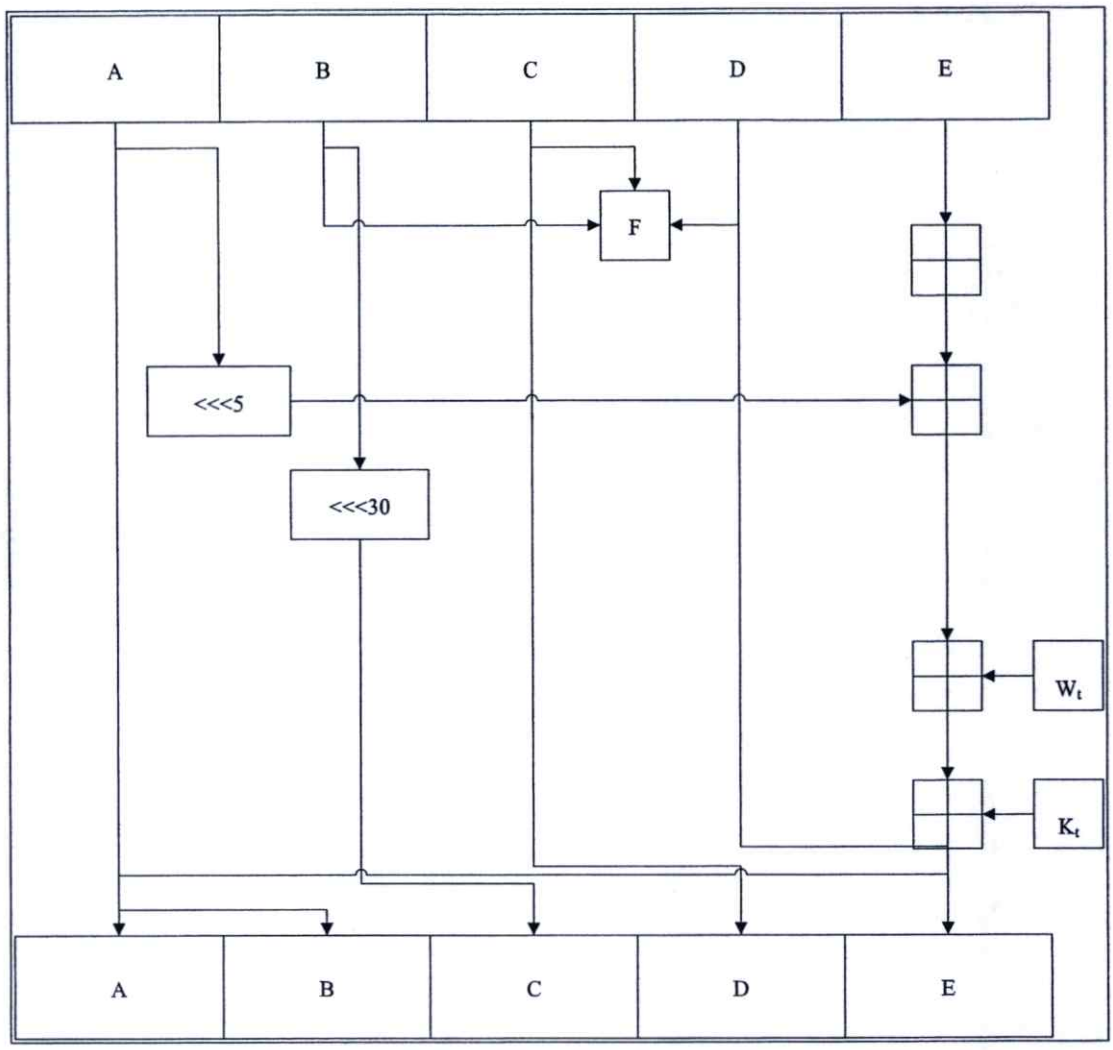
D = 0X 10325476

E = 0X C3D2E1F0

ในขั้นตอนที่ 4 ซึ่งถือเป็นหัวใจของการทำงานทั้งหมด โดยจะมีการประมวลผลข้อมูลเป็นบล็อก ครั้งละ 512 บิต โดยกระทำเป็นรอบ ๆ จนกว่าข้อมูลจะหมด โดยแสดงขั้นตอนการทำงานไว้ในรูปที่ 2.4 โดยจะมีการทำงานทั้งหมด 4 รอบ โดยในแต่ละรอบจะประกอบด้วย 20 ขั้นตอนย่อย จากนั้นเมื่อผลลัพธ์ของทั้ง 4 รอบออกมา ก็จะมีการนำไปบวกเข้ากับข้อมูล CV ที่เข้ามาอีกที ก็จะได้เป็น Message Digest ของบล็อกนั้นจากนั้น MD ก็จะใช้ในการประมวลผลข้อมูลในบล็อกถัดไปจนหมด ก็จะได้ MD สุดท้ายที่ความยาว 160 บิตออกมา



รูปที่ 2.3 แสดงการทำงานของ Secure Hash Algorithm



รูปที่ 2.4 แสดงการจัดการกับชุดของข้อมูลของ Secure Hash Algorithm

2.2 เทคนิคการตรวจจับและแก้ไขข้อผิดพลาด (Error Correction) [9]

ในการส่งข้อมูลหรือสื่อสารในระบบคอมพิวเตอร์ ผ่านการสื่อสารที่มีสัญญาณรบกวนจะส่งผลทำให้ข้อมูลที่ส่งมาเกิดความผิดพลาดไม่สามารถประมวลผลได้ถูกต้อง ดังแสดงในรูปที่ 2.5 เช่น

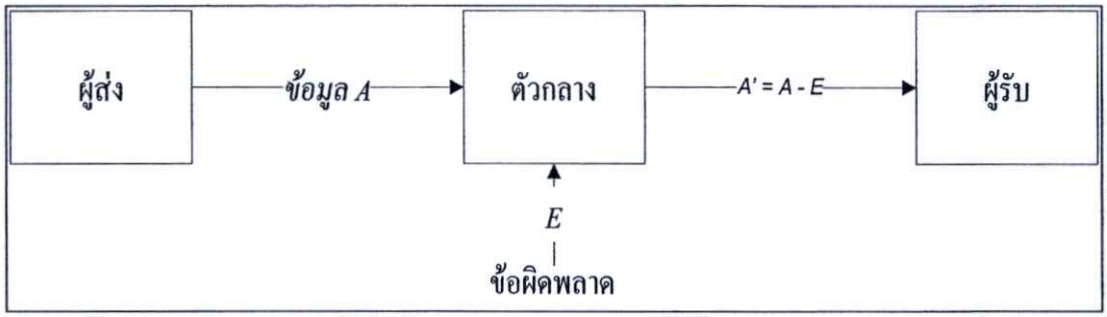
ข้อมูลที่ถูกต้อง A = 1011010

ข้อมูลผิดพลาด A' = 1101000

ข้อผิดพลาด$E = A\text{-}A'$ = 0110010

การตรวจจับหาข้อมูลที่เกิดผิดพลาดเพื่อทำการแก้ไขข้อมูลให้ถูกต้อง การแก้ไขดังกล่าวสามารถทำได้ ด้วยการผนวกข้อมูลเข้ากับชุดรหัสแก้ไขข้อผิดพลาดแล้วจึงทำการส่งเมื่อข้อมูลถูก

ส่งมาถึงผู้รับ ผู้รับจะสามารถทำการตรวจสอบข้อมูลที่ได้รับมีความผิดพลาดจากการส่งหรือไม่ หากพบข้อผิดพลาด ผู้รับพบข้อผิดพลาดจะสามารถทำการแก้ไขข้อมูลที่ผิดพลาดได้



รูปที่ 2.5 แสดงการรับส่งข้อมูลที่ถูกรบกวนให้เกิดข้อผิดพลาด

ชุดรหัสที่ใช้ผนวกเข้าไปในข้อความจะเป็นชุดของข้อมูลที่มีลักษณะสามารถแก้ไขข้อผิดพลาดได้ โดยมีระยะห่างระหว่างรหัสแต่ละตัวน้อยที่สุด (Code Distance :d_{min}) มากกว่า 1 นั่นเอง

d_{min} ก็คือจำนวนบิตที่แตกต่างกันของรหัสแต่ละตัวอย่างน้อยกี่บิต ทั้งนี้จะต้องมีการเพิ่มจำนวนบิต หรือใช้จำนวนบิตมากกว่าปกติ เช่น การให้รหัสสำหรับค่า 4 ข้อมูลปกติใช้แค่ 2 บิตก็เพียงพอ (00,01,10,11) แต่จะทำให้ระยะห่างระหว่างรหัสเท่ากับ 1 ซึ่งเมื่อเกิดบิตผิดพลาดบิตใดบิตหนึ่งขึ้น ก็จะทำให้ไปเป็นรหัสของข้อความอีกตัวทันที ทางด้านรับจะไม่สามารถทราบได้ว่ามีบิตผิดพลาดเกิดขึ้น แต่ถ้าใช้ 3 บิตสำหรับแต่ละข้อความโดยการเลือกรหัสที่มีบิตต่างกัน 2 บิตขึ้นไป เมื่อเกิดบิตผิดพลาดขึ้นจะเป็นรหัสที่ไม่ได้กำหนดให้แก่ ข้อมูลใดทางฝั่งผู้รับก็จะทราบว่ามีการผิดพลาดเกิดขึ้น แต่ถ้าเกิดผิดพลาด 2 บิตขึ้นไปก็จะตรวจจับไม่ได้เช่นกัน การที่จะสามารถตรวจจับข้อผิดพลาดของข้อมูลได้นั้นจะสามารถตรวจสอบได้ตามสมการ

$$d = d_{min} - 1 \qquad (2.1)$$

เมื่อ

d คือจำนวนบิตที่ต้องการตรวจพบ

d_{min} คือระยะห่างระหว่างรหัส

หากระบบต้องมีการแก้ไขข้อผิดพลาดของข้อมูลระยะห่างของชุดรหัสแก้ไขข้อผิดพลาดต้องมีขนาดที่ยาวขึ้น ซึ่งระยะห่างของรหัสแก้ไขข้อผิดพลาด จำนวนบิตที่สามารถตรวจสอบได้ และจำนวนบิตที่สามารถแก้ไขข้อผิดพลาดได้มีความสัมพันธ์กันตามสมการดังนี้

$$d \geq t \tag{2.2}$$

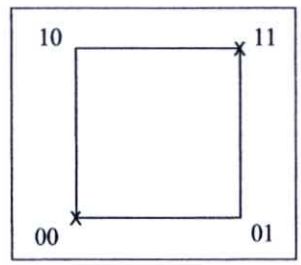
$$d_{\min} \geq 2t+1 \tag{2.3}$$

$$d_{\min} \geq t + d + 1 \tag{2.4}$$

d คือจำนวนบิตที่ต้องการตรวจพบ

$d_{\min}$ คือระยะห่างระหว่างรหัส

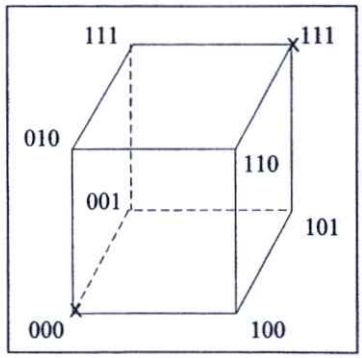
จากความสัมพันธ์ในสมการจะทำให้สามารถสร้างชุดแก้ไขข้อมูลที่มีระยะห่างที่เหมาะสมได้ ซึ่งรหัสทุกตัวสามารถเลือกที่จะออกแบบให้สามารถทำงาน ตรวจจับและแก้ไขข้อผิดพลาดได้ตามความต้องการของระบบตัวอย่างแสดงรหัสอย่างง่ายที่ใช้ในการจับงานที่ระยะห่างระหว่างรหัสดังแสดงต่อไปนี้



รูปที่ 2.6 ชุดรหัสแก้ไขข้อผิดพลาดอย่างง่าย ระยะห่าง 2 บิต {00,11}

รหัสระยะห่างเท่ากับ 2 บิต ชุดรหัสอย่างง่ายประกอบด้วย สมาชิก {00, 11} ดังแสดงในรูปที่ 2.6 หากเกิดข้อผิดพลาดเพียง 1 บิต จะสามารถตรวจสอบได้โดยรหัสแก้ไขข้อผิดพลาดจะเปลี่ยนเป็น 01 หรือ 10 แต่ไม่สามารถที่จะทำการแก้ไขข้อผิดพลาดนั้นได้ แต่หากมีข้อผิดพลาดมากเกิน บิต ชุดรหัสจะไม่สามารถทำการแก้ไขข้อผิดพลาดได้เนื่องจากรหัสที่ได้รับจะเป็นรหัสในชุดรหัสแก้ไขข้อผิดพลาดที่ถูกต้อง

รหัสระยะห่างเท่ากับ 3 บิต มีชุดรหัสอย่างง่าย คือ {000,111} ดังแสดงในรูปที่ 2.7 รหัสแก้ไขข้อผิดพลาดทั้งสองสามารถแสดงได้เป็นมุมของลูกบาศ หากเกิดข้อผิดพลาดขึ้น 1 บิตในข้อมูล ชุดรหัสแก้ไขข้อผิดพลาดจะสามารถทำการแก้ไขได้ด้วยการใช้ค่าคะแนนสูงสุดเช่น รหัสที่ส่งเป็น 000 ข้อมูลที่ผิดพลาดแสดง 001 เพราะฉะนั้น สามารถที่จะรู้ได้ว่าเกิดข้อผิดพลาดและสามารถแก้ไขได้เป็น 000 แต่หากเกิดข้อผิดพลาดเกินกว่า 1 บิต จะไม่สามารถแก้ไขได้เนื่องจากหากทำการแก้ไขแล้วข้อมูลได้จะเป็นรหัสคนละรหัสกับรหัสที่ถูกต้องเช่น รหัส 000 เกิดข้อผิดพลาด 2 บิต เป็น 011 การแก้ไขข้อผิดพลาดจะแก้ไขผิดเป็นรหัส 111 ทันที



รูปที่ 2.7 ชุดรหัสแก้ไขข้อผิดพลาดอย่างง่าย ระยะห่าง 3 บิต {000,111}

รหัสระยะห่างเท่ากับ 3 บิต มีชุดรหัสอย่างง่าย คือ {000,111} ดังแสดงในรูปที่ 2.7 รหัสแก้ไขข้อผิดพลาดทั้งสองสามารถแสดงได้เป็นมุมของลูกบาศ หากเกิดข้อผิดพลาดขึ้น 1 บิตในข้อมูล ชุดรหัสแก้ไขข้อผิดพลาดจะสามารถทำการแก้ไขได้ด้วยการใช้ค่าคะแนนสูงสุดเช่น รหัสที่ส่งเป็น 000 ข้อมูลที่ผิดพลาดแสดง 001 เพราะฉะนั้น สามารถที่จะรู้ได้ว่าเกิดข้อผิดพลาดและสามารถแก้ไขได้เป็น 000 แต่หากเกิดข้อผิดพลาดเกินกว่า 1 บิต จะไม่สามารถแก้ไขได้เนื่องจากหากทำการแก้ไขแล้วข้อมูลได้จะเป็นรหัสคนละรหัสกับรหัสที่ถูกต้องเช่น รหัส 000 เกิดข้อผิดพลาด 2 บิต เป็น 011 การแก้ไขข้อผิดพลาดจะแก้ไขผิดเป็นรหัส 111 ทันที

ตารางที่ 2.2 แสดงรหัสการตรวจจับและแก้ไขข้อผิดพลาดอย่างง่ายของรหัสระยะห่าง 4 บิต

รหัสแก้ไขข้อผิดพลาด	**0000**		**1111**
แก้ไข 1 บิต	1000		0111
	0100		1011
	0010		1101
	0001		1110
ตรวจสอบ 2 บิต		1100	
		1010	
		1001	
		0110	
		0101	
		0011	

รหัสระยะห่างเท่ากับ 4 บิต ชุดรหัสแก้ไขข้อผิดพลาดอย่างง่ายประกอบด้วย {0000, 1111} ปกติแล้ว รหัสระยะห่างเท่ากับ 4 บิต สามารถสามารถทำการตรวจสอบข้อผิดพลาดได้ 2

บิตและ แก้ไขข้อผิดพลาดได้ 1 บิต ดังแสดงในตัวอย่างตารางที่ 2.2 หากรหัสที่ได้มีการผิด 1 บิต จะสามารถทำการแก้ไขได้แต่หากมีการผิด 2 บิต รหัสที่ได้จะไม่สามารถทำการแก้ไขได้เนื่องจากมีรหัสที่เป็นไปได้ 2 ค่า

ตารางที่ 2.3 แสดงรูปแบบการทำงานที่สามารถทำงานได้ของรหัสระยะห่าง 6 บิต

เลือก	ตรวจสอบข้อผิดพลาดได้	แก้ไขข้อผิดพลาดได้
1	3	2
2	4	1
3	5	0

รหัสระยะห่างเท่ากับ 6 บิต ชุดรหัสแก้ไขข้อผิดพลาดอย่างง่ายประกอบด้วย {000000, 111111} ชุดรหัสสามารถเลือกรูปแบบการทำงานได้ 3 รูปแบบแสดงในตารางที่ 2.3 รูปแบบการทำงานทั้งหมดอยู่ในรูปของความสัมพันธ์ที่ จำนวนบิตที่สามารถตรวจสอบได้ต้องมีค่ามากกว่าจำนวนบิตที่สามารถแก้ไขได้เสมอ

2.3 ทฤษฎีบทเศษเหลือแบบจีน (Chinese Remainder Theorem)

ทฤษฎีบทเศษเหลือแบบจีน ถูกคิดค้นครั้งแรกโดยนักคณิตศาสตร์ชาวจีน และต่อมาได้รับการนำเสนอในหนังสือชื่อ Third-century AD book Sun Zi suanjing จากปัญหาการนับทหารในกองทัพ ด้วยการให้ทหารจัดแถวเป็นจำนวนต่างๆและนับเศษที่ครบแถวของทหารนำมาหาค่าของทหารทั้งหมด ปัญหาดังกล่าวสามารถหาคำตอบได้ด้วยการใช้ทฤษฎีบทดังนี้

ให้ $n_1, n_2, ..., n_k$ เป็นจำนวนเฉพาะที่มีความสัมพันธ์กันแล้ว $a_1, a_2, ..., a_k$ คือผลลัพธ์ที่ได้จากการหารเอาเศษของ x หารด้วย $n_1, n_2, ..., n_k$ จะได้ความสัมพันธ์

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\cdots \\ x &\equiv a_k \pmod{n_k} \end{aligned} \tag{2.5}$$

ค่า X ที่เป็นผลลัพธ์จะสามารถมีได้ค่าเดียว คือ

$$X = n_1.n_2...n_k. \tag{2.6}$$

สำหรับทุกค่าตั้งแต่ i ถึง k

หาจำนวนที่เท่ากับ 0 (mod n/n_i) และ a_i (mod n_i). โดย

- หาค่า

$$x_i = n/n_i \tag{2.7}$$

- หาค่า

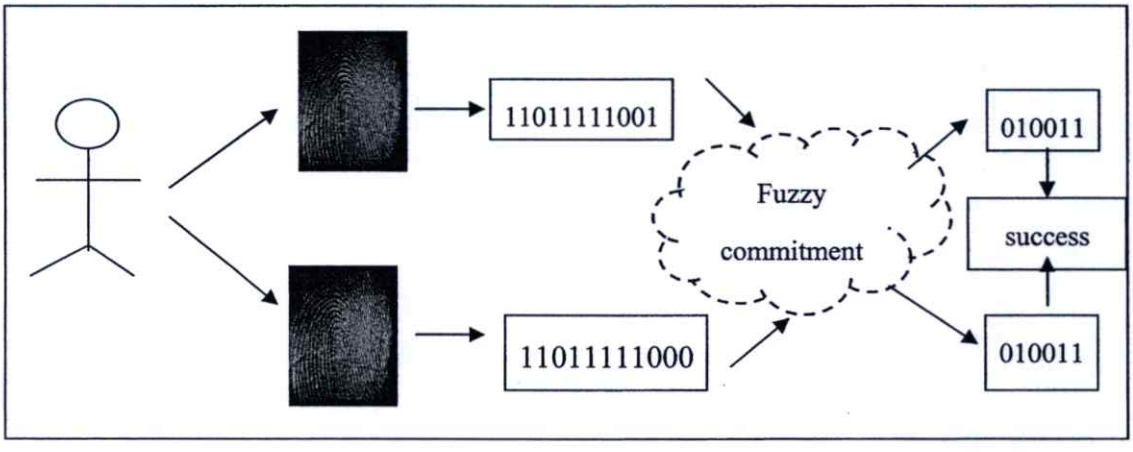
$$y_i = \text{The Inverse of } (n/n_i) \bmod (n_i) \tag{2.8}$$

$$z_i = x_i * y_i * a_i \tag{2.9}$$

หาค่า X ได้เท่ากับ

$$X = z_1 + z_2 + \ldots + z_k \bmod N \tag{2.10}$$

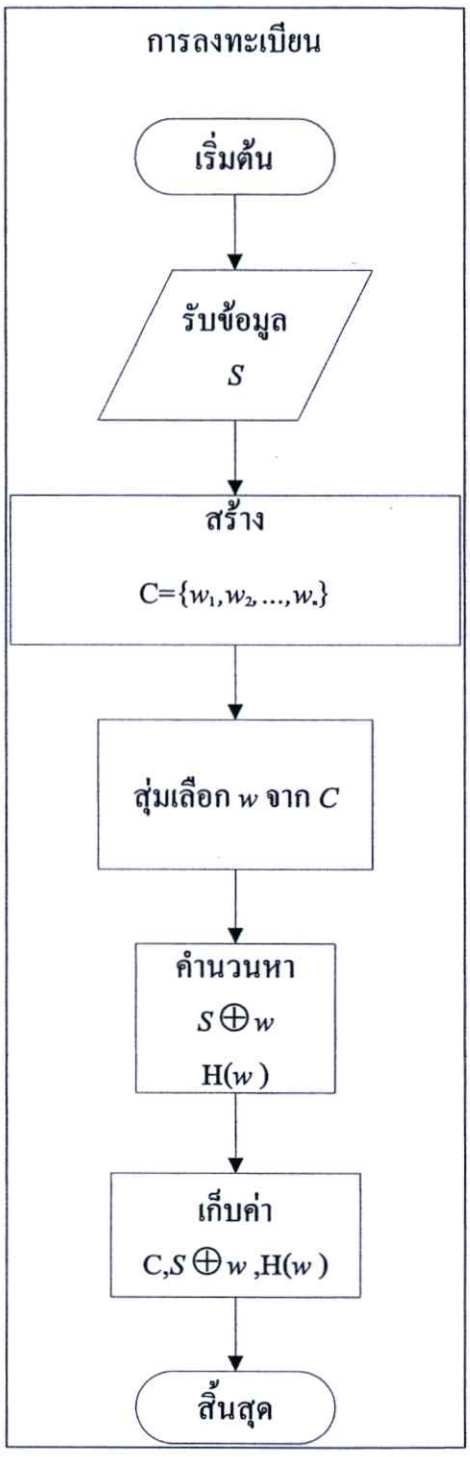
2.3 อัลกอริทึมพันธสัญญาแบบคลุมเครือ (Fuzzy Commitment) [2][10]



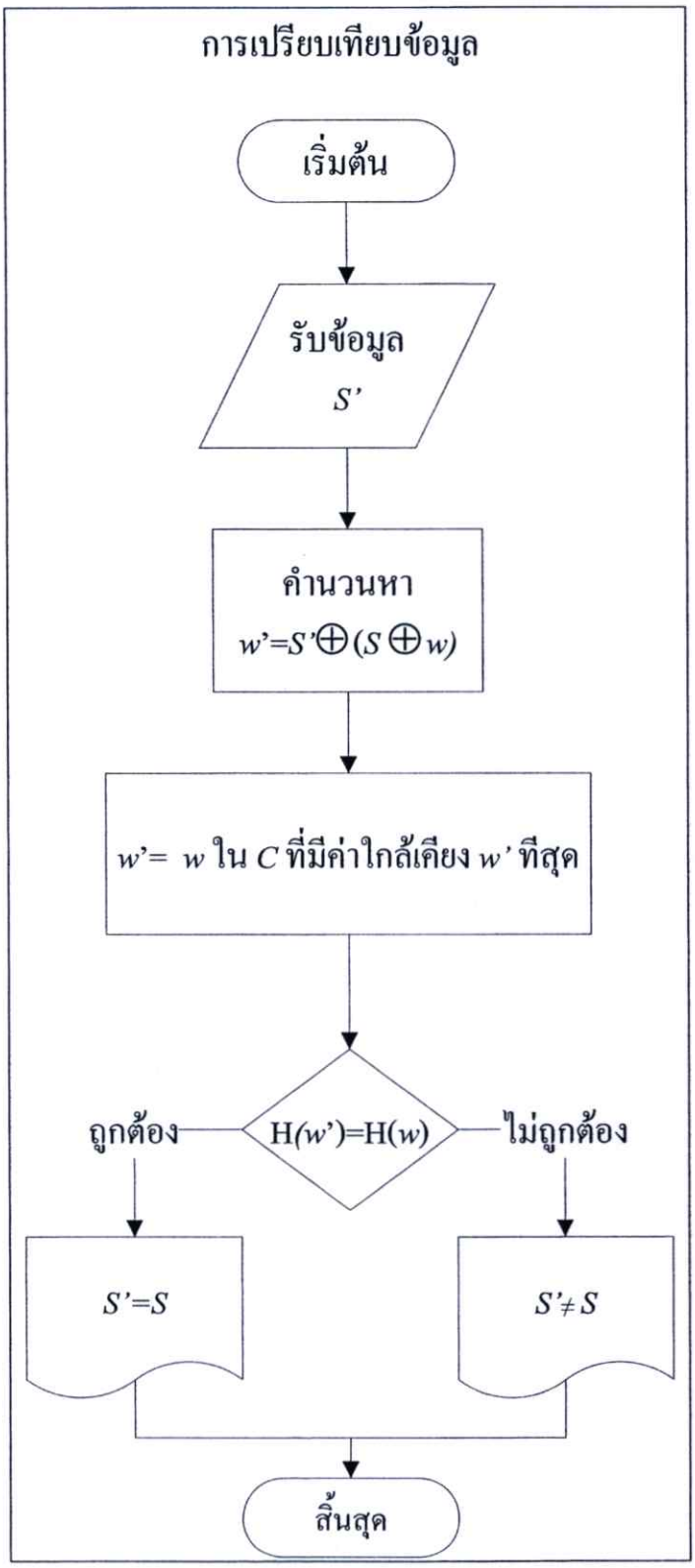
รูปที่ 2.8 แสดงขั้นตอนการพิสูจน์ตัวตนด้วยอัลกอริทึมพันธสัญญาแบบคลุมเครือ

อัลกอริทึมพันธสัญญาแบบคลุมเครือถูกเสนอโดย อริ จูเอล และ มาติน วอทเทนเบิร์ก ในปี ค.ศ. 1999 (A. Juels and M. Wattenberg: 1999) เป็นการนำเทคนิคการแก้ไขข้อผิดพลาดและเทคนิคการเข้ารหัสมาประยุกต์ใช้ ในการตรวจสอบข้อมูลที่มีความต่างการเล็กน้อย โดยไม่แสดงค่าจริงของข้อมูลในการเปรียบเทียบ หลักการของอัลกอริทึมคือ หากข้อมูล ต่างกันไม่เกินค่าที่จะ

สามารถทำการแก้ไขข้อผิดพลาดด้วย ชุดรหัสแก้ไขข้อผิดพลาดที่เลือกไว้จะยอมให้มีการแก้ไขข้อมูลเพื่อให้ได้ค่าที่ถูกต้องดังแสดงในรูปที่ 2.8 ขั้นตอนของอัลกอริทึมมีดังนี้



รูปที่ 2.9 แสดงขั้นตอนการลงทะเบียนของอัลกอริทึมพันธสัญญาแบบคลุมเครือ



รูปที่ 2.10 แสดงขั้นตอนการเปรียบเทียบข้อมูลของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

ขั้นตอนการลงทะเบียนของอัลกอริทึมพันธสัญญาแบบคลุมเครือดังแสดงในรูปที่ 2.9

1. สุ่มเลือกค่า w จาก ชุดรหัสแก้ไขข้อผิดพลาด C

2. นำค่า S ที่ได้รับจากการลงทะเบียนมาทำการ Exclusive-or กับค่า w ในข้อ 1.

$$S \oplus w = z \tag{2.11}$$

3. นำค่าw ที่ได้จากข้อ 1. เข้าฟังก์ชัน แฮช

4. ทำการเก็บค่า H (w) และ z

ขั้นตอนการลงทะเบียนของอัลกอริทึมพันธสัญญาแบบคลุมเครือดังแสดงในรูปที่ 2.10

1. ผู้ใช้งานนำค่าที่จะทดสอบเข้ามาในระบบเป็นค่า S'

2. ระบบทำการคำนวณค่า w' จาก

$$w' = z \oplus S' \tag{2.12}$$

หาก ค่า S' มีค่าไม่แตกต่างกับ S มาก ค่า w' ที่ได้จากสมการจะต้องมีค่าใกล้เคียงกับ w ทำให้เมื่อนำค่า w' มาแก้ไขข้อผิดพลาด จะต้องได้ค่า w และหาก

$$\mathrm{H}(c(w')) = \mathrm{H}(w) \tag{2.13}$$

จะถือว่าการเปรียบเทียบข้อมูลทั้งสองห่างกันเพียงเล็กน้อยสามารถยอมรับได้ แต่หากค่าที่ได้รับไม่สามารถทำให้สมการเป็นจริงข้อมูลทั้งสองไม่ใช่ข้อมูลเดียวกันและไม่มีความใกล้เคียงกัน

2.4 อัลกอริทึมการเก็บข้อมูลแบบคลุมเครือ (Fuzzy vault)[1][11]

อัลกอริทึมพันธสัญญาแบบคลุมเครือ ยังมีข้อจำกัดเรื่องลำดับของข้อมูลหากมีข้อมูลหลายๆ ค่าลำดับของข้อมูลจะต้องเรียงกันอย่างถูกต้อง อัลกอริทึม จึงจะสามารถทำงานได้อย่างถูกต้อง Juels และ Sudan ได้เสนออัลกอริทึมการเก็บข้อมูลแบบคลุมเครือ (Juels และ Sudan: 2002) ซึ่งสามารถแก้ไขปัญหาของ อัลกอริทึมพันธสัญญาแบบคลุมเครือ ได้ ซึ่งมีอัลกอริทึมดังนี้

1. ทำการสร้างชุดข้อมูล Reed-Solomon ที่ใช้สำหรับแสดงค่าของข้อมูล ซึ่งชุดข้อมูลได้จากการคำนวณ บน x-coordinates ที่ตรงกันกับสมาชิกในเซต

2. ทำการปกปิดข้อมูลด้วยการเพิ่มค่าจุด ที่ไม่มีอยู่จริงไปเป็นสมาชิกในเซต ขั้นตอนนี้เรียกว่า Lock Function ของ Fuzzy Vault

3. หากต้องการที่จะทำการ Unlock Function ของ Fuzzy Vault ผู้ใช้งานจะต้องให้ข้อมูลเข้ามาเพื่อทำการทดสอบ ซึ่งเซตนั้นจะต้องมีค่าตรงกันกับ เซตที่ได้จาก Lock Function ถ้าเซตไม่มีค่าที่มีอยู่จริงมากจนเกินไป Reed-Solomon Coding จะสามารถทำการแก้ไขข้อผิดพลาดเพื่อให้ได้รับข้อมูลที่ ถูกต้องออกมาได้

บทที่ 3

อัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

ข้อจำกัดของการใช้ อัลกอริทึมพันธสัญญาแบบคลุมเครือและอัลกอริทึมการเก็บข้อมูลแบบคลุมเครือคือ ข้อมูลที่รับเข้ามาใช้จะต้องเป็นข้อมูลเชิงเลขฐานสองแต่ละบิตในค่าเลขฐานสองมีความสำคัญไม่เท่ากันทำให้ในการเปรียบเทียบข้อมูลด้วยการใช้แก้ไขข้อผิดพลาดของข้อมูลที่ต่างกันไม่สามารถทำได้จึงต้องมีการปรับเปลี่ยนข้อมูลให้อยู่ในรูปเลขฐานหนึ่ง

การแก้ปัญหาข้างต้นนี้ จะสามารถแก้ไขได้ด้วยการพัฒนาปรับปรุงอัลกอริทึมพันธสัญญาแบบคลุมเครือให้สามารถทำงานกับข้อมูลชนิดดังกล่าวด้วยการกำหนดจุดที่จะใช้เป็นจุดอ้างอิงเพื่อทำการทดสอบและยอมให้จุดอ้างอิงคลาดเคลื่อนกันได้ในปริมาณหนึ่ง

การคำนวณในคอมพิวเตอร์ส่วนมาก แล้วจะทำงานในระบบเลขฐานสอง เนื่องจากเลขฐานสองมีสัญลักษณ์ คือ 0 และ 1 ข้อมูลอื่นๆสามารถสร้างได้จาก 0 และ 1 แต่ไม่ได้มีเพียงแค่เลขฐานสองเท่านั้นที่ใช้ 0 และ 1 อัลกอริทึม ใช้การแสดงข้อมูลแบบตัวเลขแบบ เลขฐานหนึ่ง ที่มีคุณสมบัติเหมาะสมในอัลกอริทึมการเปรียบเทียบข้อมูลแบบคลุมเครือเนื่องจากจำนวนบิตที่เปลี่ยนในการเพิ่มค่าแต่ละครั้งน้อยมากดังแสดงในตารางที่ 3.1 เปรียบเทียบจำนวนบิตที่ผิดพลาดเมื่อมีการผิดพลาดของข้อมูล 4 (00001111 ในรูปแบบเลขฐานหนึ่ง และ 100 ในรูปเลขฐานสอง)

ตารางที่ 3.1 จำนวนบิตที่ผิดพลาดของการแสดงข้อมูลแบบเลขฐานหนึ่งและเลขฐานสองสำหรับเลข 4

เลขฐานสิบ	เลขฐานหนึ่ง	จำนวนบิตที่ผิด	เลขฐานสอง	จำนวนบิตที่ผิด
0	00000000	4	000	1
1	00000001	3	001	2
2	00000011	2	010	2
3	00000111	1	011	3
4	00001111	0	100	0
5	00011111	1	101	1
6	00111111	2	110	1
7	01111111	3	111	2

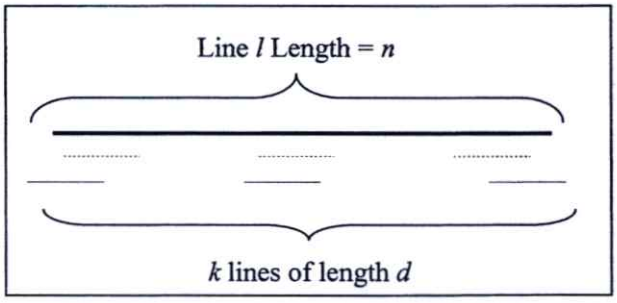
ข้อจำกัดที่สำคัญอีกประการหนึ่งของการใช้งานอัลกอริทึมพันธสัญญาแบบคลุมเครือที่ง่ายต่อการถูกโจมตีด้วยการใช้ข้อมูลทั้งหมดและเมื่อนำมาแก้ไขให้สามารถใช้งานกับข้อมูลที่เป็นเลขฐานหนึ่งแล้วทำให้ความง่ายต่อการถูกโจมตีมีมากขึ้นวิธีการแก้ไขสามารถทำได้โดยการพัฒนาให้มีความปลอดภัยมากขึ้นด้วยการแบ่งข้อมูลออกเป็นส่วนๆแล้วจึงทำการเข้ารหัสเก็บ

ข้อมูลไว้เมื่อต้องการเปรียบเทียบข้อมูลสามารถทำได้ด้วยการนำข้อมูลแต่ละส่วนกลับมาหาความลับอีกครั้งด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนเพื่อนำข้อมูลที่เป็นความลับกลับคืนมาเพื่อทำการตรวจสอบ

ในบทที่ 3 นี้จะกล่าวถึงการพัฒนาอัลกอริทึมพันธสัญญาแบบคลุมเครือ ให้สามารถทำงานการเปรียบเทียบข้อมูลเส้นในลักษณะ 1 มิติ และการเปรียบเทียบข้อมูลสี่เหลี่ยม 2 มิติ การพัฒนาการทำงานเพื่อให้มีความปลอดภัยที่มากขึ้นและสามารถรองรับต่อการถูกโจมตีด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน (Chinese Remainder Theorem)

3.1 การเปรียบเทียบข้อมูลเส้นตรง 1 มิติ

หัวข้อนี้จะแสดงอัลกอริทึม การเปรียบเทียบข้อมูลรูปภาพ 1 มิติ ด้วยการแสดงพื้นที่ที่มีเส้น l ที่มีความยาวเท่ากับ n มีการวางเส้นความยาว d จำนวน k เส้น และสามารถ เกิดข้อผิดพลาดได้มากที่สุด e การลงทะเบียนจะเรียบร้อยเมื่อ มีการวางเส้นยาว d จำนวน k เส้น ลงบนเส้น หลังจากนั้นผู้ใช้งานที่มองไม่เห็นเส้นทั้ง k เส้น จะทำการทดสอบโดยวางเส้น ความยาว d จำนวน k เส้น ลงบน เส้น l หาก เส้นทั้งสองมีการทับกันทั้งหมด การจับคู่จะเสร็จสมบูรณ์ ในรูปที่ 3.1 สามเส้นที่เป็นเส้นประแสดงข้อมูลที่เป็นการลงทะเบียน เส้นทึบสามเส้นแสดงข้อมูลที่นำมาเปรียบเทียบ จากรูปเส้นทั้ง 3 คู่มีทั้งหมดมีการซ้อนทับกัน ดังรูปที่3.1รายละเอียดการของอัลกอริทึมแสดงในกระบวนการลงทะเบียนแสดงในรูปที่ 3.2 และกระบวนการเปรียบเทียบข้อมูลแสดงในรูปที่ 3.3



รูปที่ 3.1 แสดงพื้นที่ของการเปรียบเทียบข้อมูล 1 มิติ

ให้

l คือ เส้นตรงที่เป็นความลับ

x คือ จุดเริ่มต้นของเส้นตรงที่เป็นความลับ

d คือ ความยาวของเส้นตรง

c คือ ชุดรหัสแก้ไขข้อผิดพลาด

o คือ เส้นตรงที่นำมาเปรียบเทียบ

s คือ จุดเริ่มต้นของเส้นตรงที่นำมาเปรียบเทียบ

3.1.1 กระบวนการลงทะเบียน

ขั้นตอนการลงทะเบียนของข้อมูล 1มิติที่เป็นเส้นตรงแสดงตามรูปที่ 3.2 มีขั้นตอนการทำงานดังนี้

1. เส้นตรงเส้นแรก l_1 ถูกวางลงในตำแหน่ง x_1 , เส้นตรงเส้นที่สอง l_2 ถูกวางลงในตำแหน่ง x_2 และวางลงเรื่อยๆจนถึงเส้นตรงเส้นสุดท้าย l_k ที่ตำแหน่ง x_k เมื่อเส้นที่ l_i ถูกวางลงบนตำแหน่ง x_i ซึ่งจะวางอยู่ในตำแหน่ง จาก x_i ถึง $x_i + d$-1 สำหรับเส้นสองเส้น l_i และ l_j ค่า $|x_i - x_j|$ ต้องมากกว่า d เพื่อความมั่นใจว่าไม่มีสองเส้นใดวางอยู่ใกล้กันจนเกินไป เนื่องจากหากมีการวางเส้นที่ใกล้จนเกินไปอาจจะทำให้หนึ่งเส้นสามารถทับข้อมูลสองเส้นได้ และตำแหน่งการวางเป็นความลับ
2. สำหรับทุกเส้นทำการเลือกค่า c_i จากชุดรหัสแก้ไขข้อผิดพลาด ขนาดความยาว n ที่สามารถแก้ไขข้อผิดพลาดได้ d ตำแหน่ง
3. นำค่า เลขฐานหนึ่ง ของ x_i และ$(x_i + d)$ exclusive-or ด้วย c_i และคำนวณค่าแฮชของ c_i
4. สร้างเซต L เก็บค่าที่ใช้ในการคำนวณ $(x_i \oplus c_i, (x_i + d) \oplus c_i, \mathrm{H}(c_i))$ ในเซต L

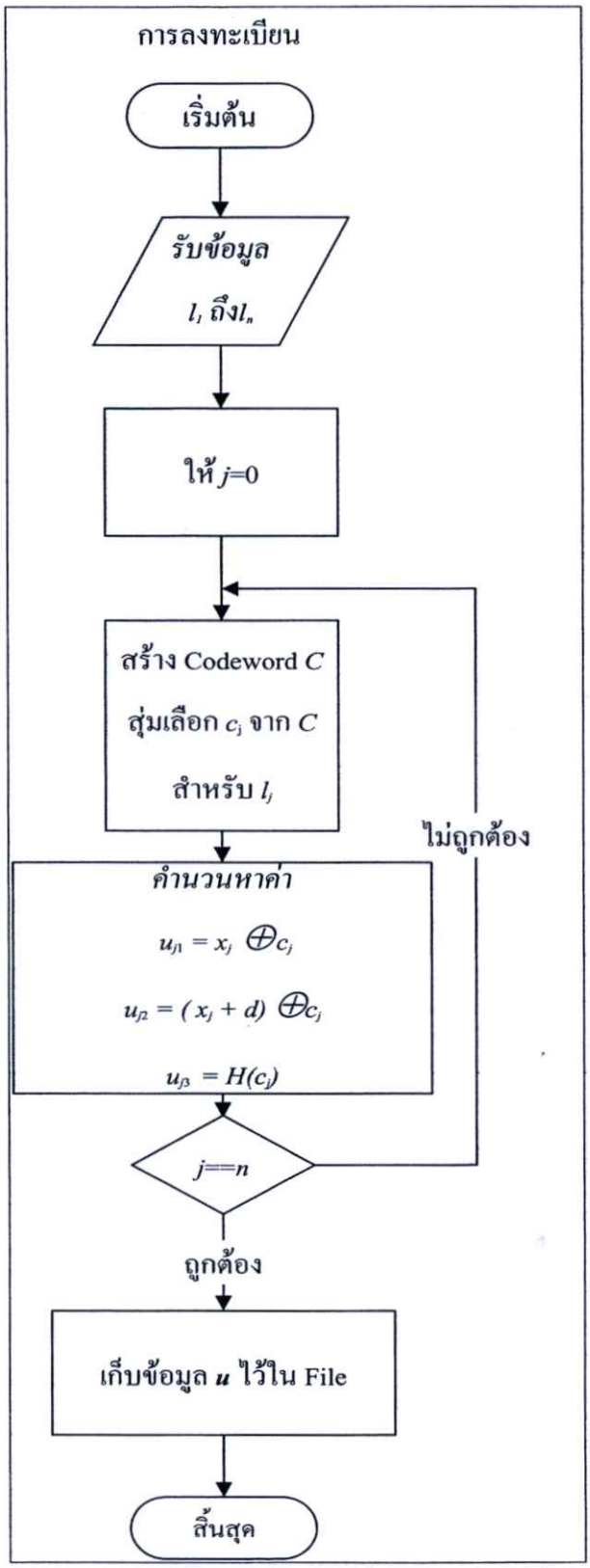
3.1.2 กระบวนการเปรียบเทียบข้อมูล

1. เส้นตรง k เส้น $O_1, .., O_n$, ทุกเส้นยาว d ถูกวางลงบนพื้นที่.
2. กำหนดให้จุดที่จะวางเส้นตรงแต่ละเส้นห่างกัน d เพื่อป้องกันการซ้อนทับกัน
3. สร้างเซต O นำข้อมูลตำแหน่งของทุกเส้น $(S_i, S_i + d)$ เก็บไว้ใน O.
4. สำหรับทุกค่า $(x_i \oplus c_i, (x_i + d) \oplus c_i, \mathrm{h}(c_i))$ ใน L และ $(S_j, S_j + d)$ ใน O นำเข้ามาในกระบวณการดังนี้
   a. คำนวณ หาค่า

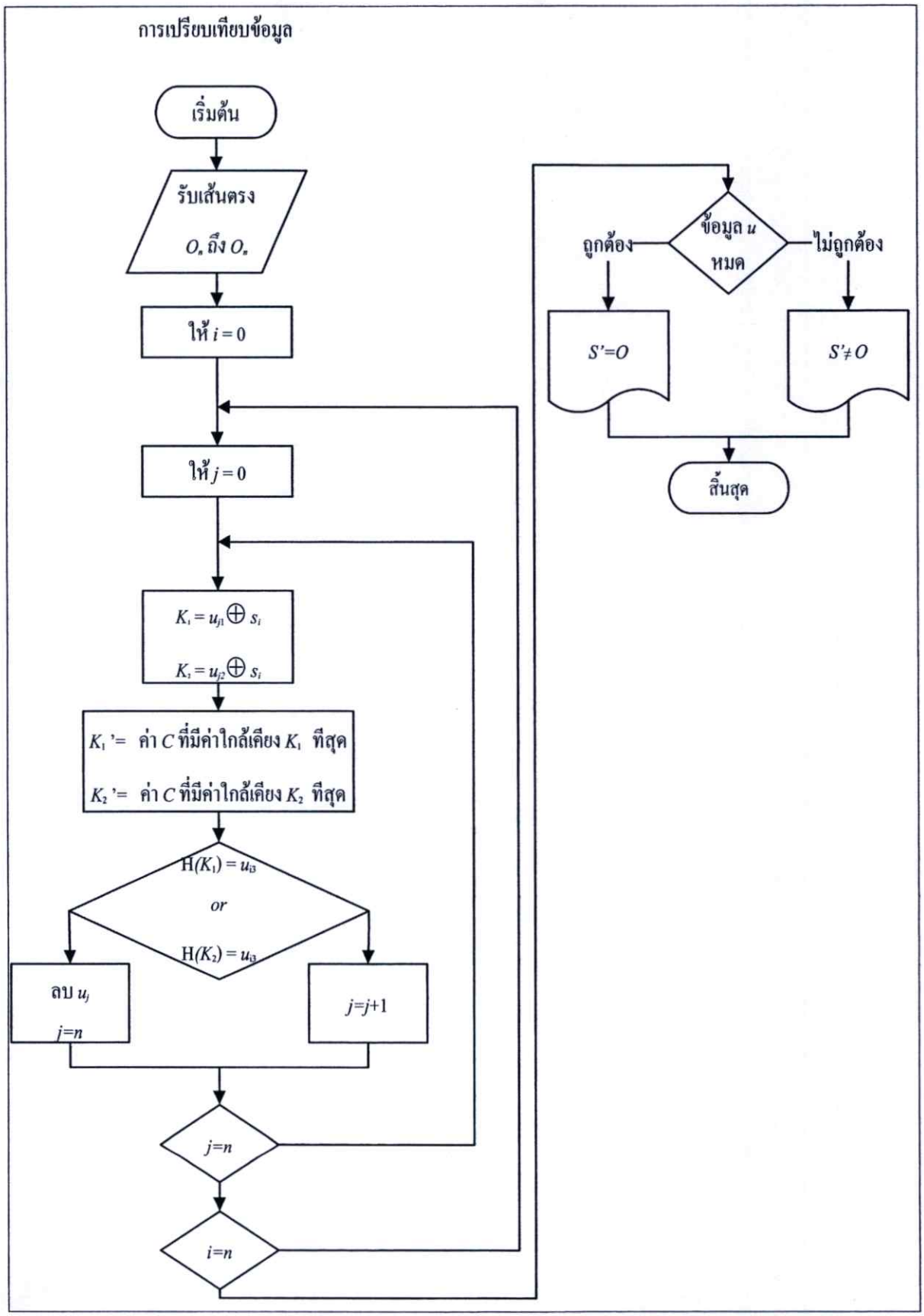
$$u_1 = S_j \oplus x_i \oplus c_i \tag{3.1}$$

   b. คำนวณ หาค่า

$$u_2 = S_j \oplus (x_i + d) \oplus c_i \tag{3.2}$$



รูปที่ 3.2 แสดงขั้นตอนการลงทะเบียนของการเปรียบเทียบข้อมูล 1 มิติ



รูปที่ 3.3 แสดงขั้นตอนการเปรียบเทียบของการเปรียบเทียบข้อมูล 1 มิติ

c. ให้ v_1 และ v_2 คือค่าที่ทำการแก้ไขข้อผิดพลาดแล้วของ u_1 และ u_2 ด้วยเทคนิคการแก้ไขข้อผิดพลาด ถ้า

$$H(v_1) = H(v_2) = H(c_i) \tag{3.3}$$

แล้วการเปรียบเทียบสำเร็จ

ลบ $(x_i \oplus c_i, (x_i + d) \oplus c_i, H(c_i))$ ออกจากเซต L และข้ามไปขั้นตอนที่ 5 นอกจากนั้นดำเนินการตามขั้นตอนต่อไป

d. ดำเนินการตามขั้นตอนในข้อ c อีกครั้งแต่แทนที่ S_j ด้วย $S_j + d$ หากการเปรียบเทียบสำเร็จลบ $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ ออกจากเซต L

5. ลบ $(S_i, S_i + d)$ ออกจาก O ดำเนินการซ้ำขั้นตอนที่ 4 จนกว่าสมาชิกในเซต O หมด
6. หากสมาชิกใน L ถูกลบจนหมดการเปรียบเทียบสำเร็จ นอกเหนือจากนี้ถือว่าการเปรียบเทียบล้มเหลว

3.2 การเปรียบเทียบข้อมูลรูปสี่เหลี่ยม 2 มิติ

ในหัวข้อนี้จะแสดงอัลกอริทึมการเปรียบเทียบข้อมูลรูปภาพ 2 มิติด้วยการแสดงพื้นที่ขนาด mxn มีการวางสี่เหลี่ยมขนาด dxd จำนวน k รูปดังแสดงในรูปที่ 3.4 และสามารถเกิดข้อผิดพลาดได้มากที่สุด d การลงทะเบียนจะเรียบร้อยเมื่อมีการวางสี่เหลี่ยมขนาด dxd จำนวน k รูปลงบนพื้นที่ หลังจากนั้นผู้ใช้งานที่มองไม่เห็นสี่เหลี่ยมทั้ง k รูป จะทำการทดสอบโดยวางสี่เหลี่ยม ขนาด dxd จำนวน k รูป ลงบนพื้นที่ หากสี่เหลี่ยมมีการทับกันการจับคู่จะเสร็จสมบูรณ์ ในรูปที่ 3.4รูปสี่เหลี่ยมสามรูปที่เป็นเส้นประแสดงข้อมูลที่เป็นการลงทะเบียนรูปสี่เหลี่ยมสามรูปที่เป็น เส้นทึบแสดงข้อมูลที่นำมาเปรียบเทียบ จากรูปสี่เหลี่ยมทั้ง 3 คู่มีสองคู่ที่มีการซ้อนทับกัน รายละเอียดการของอัลกอริทึมแสดงในกระบวนการลงทะเบียนและกระบวนการเปรียบเทียบข้อมูล

ให้

l คือ สี่เหลี่ยมความลับ

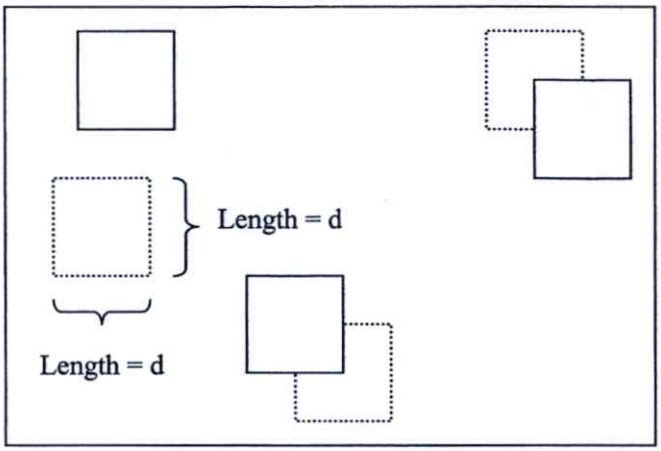
d คือ ความยาวด้านของสี่เหลี่ยม

(x,y) คือ จุดมุมล่างซ้ายของสี่เหลี่ยมความลับ

c คือ ชุดรหัสแก้ไขข้อผิดพลาด

O คือ สี่เหลี่ยมที่นำมาเปรียบเทียบ

(s,t) คือ จุดมุมล่างซ้ายของสี่เหลี่ยมที่นำมาเปรียบเทียบ



รูปที่ 3.4 แสดงพื้นที่ของการเปรียบเทียบข้อมูล 2 มิติ

3.2.1 กระบวนการลงทะเบียน

1. สี่เหลี่ยมรูปที่หนึ่ง l_1 ถูกวางลงบนตำแหน่ง (x_1, y_1)ซึ่งเป็นมุมล่างซ้ายของสี่เหลี่ยม สี่เหลี่ยมรูปที่สอง l_2 ถูกวางลงบนตำแหน่ง (x_2, y_2), และ จนถึงสี่เหลี่ยมรูปสุดท้ายถูกวางบนตำแหน่ง (x_k, y_k) ซึ่งสี่เหลี่ยม l_i ที่ถูกวางไว้บนตำแน่ง (x_i, y_i), สี่เหลี่ยมจะอยู่ในพื้นที่ของจุดสี่จุดคือ (x_i, y_i), $(x_i, y_i + d)$, $(x_i + d, y_i)$, $(x_i + d, y_i + d)$ สำหรับเป็นที่สี่เหลี่ยม l_i and l_j, $|x_i - x_j|$ และค่า $|y_i - y_j|$ ต้องมีค่ามากกว่าสองเท่าของ d เพื่อให้เป็นข้อยืนยันว่าไม่มีสี่เหลี่ยมใดซ้อนทับกันเอง และจุดที่วางสี่เหลี่ยมถือเป็นความลับ
2. สำหรับทุกสี่เหลี่ยมทำการเลือกค่า c_i จากชุดรหัสแก้ไขข้อผิดพลาด ขนาดความยาว n ที่สามารถแก้ไขข้อผิดพลาดได้ d ตำแหน่ง
3. ใช้ เลขฐานหนึ่งแสดงค่าของ x_i, $x_i + d$, y_i, $y_i + d$, และ exclusive-or ด้วย c_i คำนวณค่าแฮชของ c_i
4. สร้างเซต L นำค่าที่ได้เก็บไว้ใน L $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i,$ $H(c_i))$

3.1.2 กระบวนการเปรียบเทียบข้อมูล

1. สี่เหลี่ยม $O_1, .., O_n$, ทุกรูปมีขนาด dxd, วางอยู่บนพื้นที่.
2. กำหนดให้ตำแหน่งที่จะวางสี่เหลี่ยมแต่ละรูปห่างกัน dxd เพื่อป้องกันการซ้อนทับกันของสี่เหลี่ยม

3. สร้างเซต O เก็บตำแหน่งมุม แต่ละด้าน $(S_j, S_i + d, t_j, t_i + d)$ ลงในเซต O
4. สำหรับทุกค่า $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i, H(c_i))$ ที่เก็บไว้ใน L และ $(S_j, S_i + d, t_j, t_i + d)$ ที่เก็บไว้ในเซต O นำมาเข้าขั้นตอนดังนี้

a. คำนวณหาค่า

$$u_1 = S_j \oplus x_i \oplus c_i \tag{3.4}$$

b. คำนวณหาค่า

$$u_2 = S_j \oplus (x_i + d) \oplus c_i \tag{3.5}$$

c. คำนวณหาค่า

$$u_3 = t_j \oplus y_i \oplus c_i \tag{3.6}$$

d. คำนวณหาค่า

$$u_4 = t_j \oplus (y_i + d) \oplus c_i \tag{3.7}$$

e. ให้ v_1, v_2, v_3 และ v_4 คือค่าที่แก้ไขข้อผิดพลาดของ u_1, u_2, u_3 และ u_4 ด้วยการใช้วิธีการของ การแก้ไขข้อผิดพลาด

ถ้า

$$H(v_1) = H(v_2) = H(v_3) = H(v_4) = H(c_i) \tag{3.8}$$

แล้วการเปรียบเทียบพบคู่ของสี่เหลี่ยม

ให้ลบ $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i, h(c_i))$ ออกจาก L และดำเนินการขั้นตอนที่ 5. หากเปรียบเทียบไม่สำเร็จดำเนินการตามขั้นตอนต่อไป

f. ดำเนินการตามขั้นตอน e , แต่เปลี่ยน S_j ด้วย $S_j + d$ ถ้าเปรียบเทียบสำเร็จ ลบ $(x_i \oplus c_i, (x_i + d) \oplus c_i, H(c_i))$ ออกจาก เซต L หากเปรียบเทียบไม่สำเร็จดำเนินการตามขั้นตอนต่อไป

g. ดำเนินการตามขั้นตอน e แต่เปลี่ยน t_j ด้วย $t_j + d$ ถ้าเปรียบเทียบสำเร็จลบ ($x_i \oplus c_i$, $(x_i + d) \oplus c_i$, H(c_i)) ออกจาก เซต L หากเปรียบเทียบไม่สำเร็จดำเนินการตามขั้นตอนต่อไป

h. ดำเนินการตามขั้นตอน e , แต่เปลี่ยน S_j ด้วย $S_j + d$ และ t_j ด้วย $t_j + d$ ถ้าเปรียบเทียบสำเร็จลบ ($x_i \oplus c_i$, $(x_i + d) \oplus c_i$, $H(c_i)$) ออกจาก เซต L หากเปรียบเทียบไม่สำเร็จดำเนินการตามขั้นตอนต่อไป

5. ลบ ($S_i, S_i + d, t_i, t_i + d$) ออกจากเซต O ดำเนินการซ้ำขั้นตอนที่ 4 จนกว่าสมาชิกในเซต O หมด
6. หากสมาชิกใน L ถูกลบจนหมดการเปรียบเทียบสำเร็จ นอกเหนือจากนี้ถือว่าการเปรียบเทียบล้มเหลว

3.3 การเพิ่มความปลอดภัยให้กับอัลกอริทึมด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

เนื่องจากการใช้ข้อมูลเลขฐานหนึ่งในอัลกอริทึมทำให้เกิดช่องทางในการโจมตีข้อมูลด้วยการทดลองข้อมูลความลับทั้งหมดที่เป็นไปได้เพื่อทำการหาข้อมูลซึ่งใช้เวลาในการทดสอบหาข้อมูลด้วยการใช้เวลาเป็นสมการเส้นตรงและสามารถทำการโจมตีด้วยการนำ รหัสแก้ไขข้อผิดพลาดทั้งหมดที่เป็นไปได้มาทำการทดสอบหาค่าแฮชที่เป็นไปได้เพื่อหารหัสที่ถูกต้อง จึงจำเป็นต้องมีการปรับปรุงเพื่อให้อัลกอริทึมมีความปลอดภัยและยากต่อการถูกถอดรหัสมากขึ้นซึ่งสามารถแก้ไขปัญหานี้ได้ด้วยการแบ่งข้อมูลความลับออกเป็นส่วนๆ ด้วยการนำชุดจำนวนเฉพาะสัมพัทธ์มาหารแบบเอาเศษแล้วจึงนำไปเข้ารหัส เก็บไว้ เมื่อต้องมีการเปรียบเทียบข้อมูลเป็นด้วยการนำทฤษฎีเศษเหลือแบบจีนมาใช้ในการปรับปรุงข้อมูลที่เป็นเลขฐานหนึ่งให้มีลักษณะเป็นข้อมูลในเลขฐานสอง หรือ ใช้การตรวจสอบหาจำนวนความคลาดเคลื่อนที่เกิดขึ้นบ่อยครั้งที่สุดมาเป็นค่าคะแนนส่วนใหญ่เพื่อนำมาปรับปรุง ข้อมูลที่นำมาเปรียบเทียบ ให้มีค่าถูกต้องตรงกับค่าข้อมูลความลับและนำเข้ากระบวนการเปรียบเทียบกับค่าความลับที่เก็บไว้ในรูปแฮชฟังก์ชัน

3.3.1 การเลือกชุดจำนวนเฉพาะสัมพัทธ์และการสร้างชุดรหัสแก้ไขข้อผิดพลาดที่เหมาะสม

เนื่องจากการเลือกชุดจำนวนเฉพาะสัมพัทธ์เพื่อนำมาใช้หารแบบได้ผลลัพธ์เป็นเศษเหลือเพื่อนำมาใช้ในทฤษฎีบทเศษเหลือแบบจีน มีความสัมพันธ์กับค่าข้อมูลความลับและการสร้างชุดรหัสแก้ไขข้อผิดพลาด ทำให้ส่งผลกระทบต่อความปลอดภัยและประสิทธิภาพของการทำงาน ดังนั้นการเลือกชุดของข้อมูลที่จะนำมาใช้งานต้องมีคุณสมบัติดังนี้

1. ทุกค่าในชุดจำนวนเฉพาะสัมพัทธ์จะต้องมีค่าน้อยกว่าข้อมูลความลับ

$$S > a \tag{3.9}$$

หากสมาชิกในชุดจำนวนเฉพาะสัมพัทธ์ที่นำมาใช้มีขนาดใหญ่กว่าข้อมูลความลับเมื่อมีการนำมาหารเอาเศษผลลัพธ์ที่ได้จะคือข้อมูลความลับ เมื่อนำข้อมูลเหล่านั้นเก็บไว้จะส่งผลต่อความปลอดภัยของข้อมูลเมื่อถูกเปิดเผยข้อมูลที่เก็บไว้ จะทำให้ผู้ประสงค์ร้ายสามารถรู้ความลับได้จากค่าข้อมูลที่เก็บค่าเดียวกันซ้ำๆหลายครั้ง

2. เมื่อนำค่าความลับมาหารเอาเศษด้วยทุกค่าในชุดจำนวนเฉพาะสัมพัทธ์ ค่าที่ได้ต้องมีค่ามากกว่าความคลาดเคลื่อนที่ยอมรับได้และน้อยกว่าค่าจำนวนเฉพาะสัมพัทธ์ลบด้วยความคลาดเคลื่อนที่ยอมรับได้

$$e<(S \bmod a) < (a-e) \tag{3.10}$$

e คือค่าความคลาดเคลื่อนที่ยอมรับได้

หากมีการหารเอาเศษแล้วทำให้ได้ผลลัพธ์ข้อมูลที่ได้มีค่าน้อยกว่า e จะทำให้เมื่อมีการแก้ไขข้อผิดพลาดค่ารหัสแก้ไขข้อผิดพลาด ที่เลือกใช้จะมีการเลือกใช้ที่ผิดค่าซึ่งจะเป็นค่าที่อยู่ก่อนหน้าค่ารหัสที่ถูกต้อง หากค่าที่ได้มีค่ามากกว่า $a-e$ เมื่อมีไขความผิดพลาดเกิดขึ้นในช่วงดังกล่าวจะส่งผลให้เกิดการเลือกใช้ รหัสผิดค่าไปใช้ค่ารหัสที่อยู่ถัดไปจากรหัสที่ถูกต้อง ทั้งสองกรณีจะส่งผลให้เกิดการทำงานที่ผิดพลาด

3. การเลือกค่าของชุดจำนวนเฉพาะสัมพัทธ์จะส่งผลต่อการเลือกความยาวของชุดข้อมูลแก้ไขข้อผิดพลาดที่เหมาะสม เพราะจำนวนบิตของชุดข้อมูลแก้ไขข้อผิดพลาดจะประกอบไปด้วยสองส่วนคือส่วนที่เป็นข้อมูลและส่วนที่ใช้สำหรับตรวจสอบเพราะฉะนั้นการเลือกค่า

$$2^p \geq \sum_{i=0}^{j} \binom{n}{j} \tag{3.11}$$

เมื่อ n คือ จำนวนหลักทั้งหมดของชุดแก้ไขข้อมูล

p คือ จำนวนหลักที่ใช้สำหรับแก้ไขข้อผิดพลาด

j คือ ปริมาณบิตที่สามารถแก้ไขข้อผิดพลาดได้

4. จำนวนชุดจำนวนเฉพาะความสัมพัทธ์ต้องมีจำนวนมากเพียงพอที่จะใช้ทฤษฎีบทเศษเหลือแบบจีนทำการหาค่าข้อมูลความลับกลับคืนมาได้ซึ่งจำนวนของจำนวนเฉพาะ

สัมพัทธ์จะมีต้องมีจำนวนอย่างน้อยที่สัมพันธ์กับข้อมูลความลับจึงจะสามารถหาค่าความลับกลับมาได้ตามสมการ

$$N \geq \log_k S \tag{3.12}$$

เมื่อ

N คือ จำนวนในชุดจำนวนเฉพาะสัมพัทธ์กัน
k คือ ค่าของ จำนวนเฉพาะสัมพัทธ์ ที่มีค่าน้อยที่สุด
S คือ ข้อมูลความลับ

3.3.2 กระบวนการลงทะเบียน

1. สร้างชุดจำนวนเฉพาะสัมพัทธ์กัน n ค่า $a_1, a_2, ..., a_n$ ซึ่ง ค่าที่สร้างขึ้นมาต้องมีคุณสมบัติตามหัวข้อ 3.3.1
3. คำนวนหาค่า

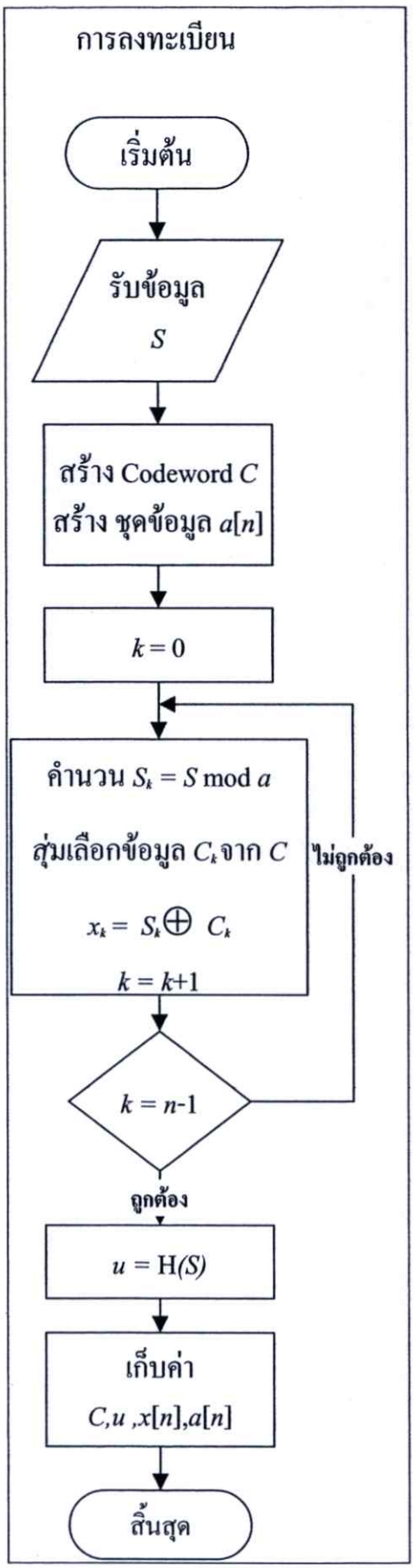
$$\begin{aligned} S_1 &= S \bmod a_1 \\ S_2 &= S \bmod a_2 \\ &\cdots \\ S_n &= S \bmod a_n \end{aligned} \tag{3.13}$$

4. สร้างค่าชุดรหัสแก้ไขข้อผิดพลาด สำหรับแก้ไขข้อผิดพลาด n บิต $x_1, x_2, ..., x_n$
5. คำนวนหาค่า

$$\begin{aligned} c_1 &= S_1 \oplus x_1 \\ c_2 &= S_2 \oplus x_2 \\ &\cdots \\ c_n &= S_n \oplus x_n \end{aligned} \tag{3.14}$$

ข้อมูลทั้งหมดอยู่ในรูปของเลขฐานหนึ่ง

6. คำนวนหาค่า H(S) ในรูปแบบเลขฐานสอง
7. เก็บค่า H(S),$\{ a_1, a_2, ..., a_n \}$, $\{ c_1, c_2, ..., c_n \}$ และ $\{ x_1, x_2, ..., x_n \}$



รูปที่ 3.5 แสดงขั้นตอนการลงทะเบียนด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

3.3.3 กระบวนการเปรียบเทียบข้อมูลด้วยการใช้ ทฤษฎีเศษเหลือแบบจีน

1. ผู้ใช้งานส่งข้อมูลที่ใช้ในการ O
2. คำนวนหาค่า

$$\begin{aligned} O_1 &= O \bmod a_1 \\ O_2 &= O \bmod a_2 \\ &\cdots \\ O_n &= O \bmod a_n \end{aligned} \tag{3.15}$$

3. คำนวนหาค่า

$$\begin{aligned} y_1 &= c_1 \oplus O_1 \\ y_2 &= c_2 \oplus O_2 \\ &\cdots \\ y_n &= c_n \oplus O_n \end{aligned} \tag{3.16}$$

ซึ่งมีการเปลี่ยนแปลงค่าทั้งหมดอยู่ในรูป เลขฐานหนึ่ง

4. แก้ไขข้อผิดพลาด $(y_1, y_2, \ldots, y_n)$ ด้วย ชุดรหัสแก้ไขข้อผิดพลาด $x_1, x_2, \ldots, x_n$
5. คำนวนหาค่า

$$\begin{aligned} u_1 &= c_1 \oplus y_1 \\ u_2 &= c_2 \oplus y_2 \\ &\cdots \\ u_n &= c_n \oplus y_n \end{aligned} \tag{3.17}$$

6. ใช้ ทฤษฎีบทเศษเหลือแบบจีน เพื่อหาค่า u
7. เปรียบเทียบค่า H(u) กับค่า H(S)

หาก

$$\mathrm{H}(u) = \mathrm{H}(S) \tag{3.18}$$

แล้ว

$$O = S \tag{3.19}$$

ข้อมูลทั้งสองข้อมูลเป็นข้อมูลเดียวกัน

หาก

$$H(u) \neq H(S) \tag{3.20}$$

แล้ว

$$O \neq S \tag{3.21}$$

ข้อมูลทั้งสองค่าไม่ใช่ข้อมูลเดียวกัน

3.3.4 กระบวนการเปรียบเทียบข้อมูลด้วยการใช้ค่าคะแนนส่วนใหญ่

ในการเปรียบเทียบข้อมูลหากนอกเหนือจากการใช้ทฤษฎีบทเศษเหลือแบบจีนในการตรวจสอบค่าความลับแล้วสามารถใช้ค่าคะแนนส่วนใหญ่ในการหาความลับได้ด้วยวิธีการดังนี้

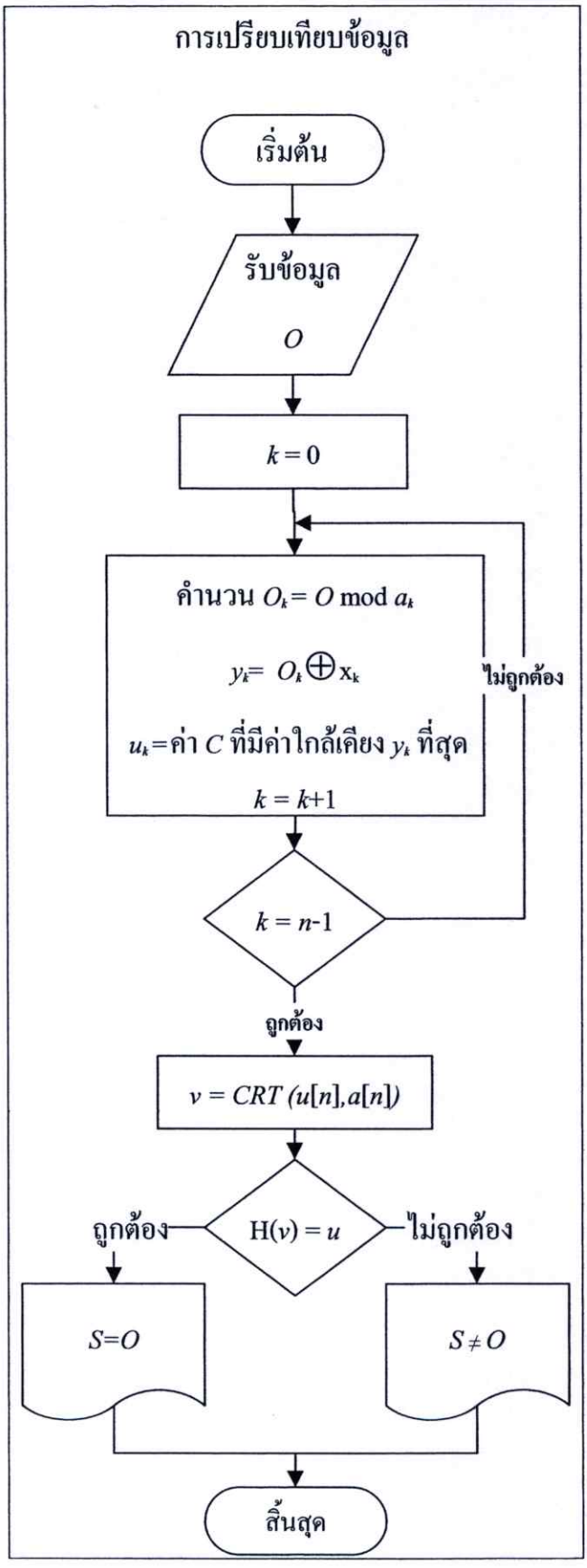
1. ผู้ใช้งานส่งข้อมูลที่ใช้ในการ O
2. คำนวนหาค่า

$$\begin{aligned} O_1 &= O \bmod a_1 \\ O_2 &= O \bmod a_2 \\ &\cdots \\ O_n &= O \bmod a_n \end{aligned} \tag{3.22}$$

3. คำนวนหาค่า

$$\begin{aligned} y_1 &= c_1 \oplus O_1 \\ y_2 &= c_2 \oplus O_2 \\ &\cdots \\ y_n &= c_n \oplus_n \end{aligned} \tag{3.23}$$

ซึ่งมีการเปลี่ยนแปลงค่าทั้งหมดอยู่ในรูป เลขฐานหนึ่ง



รูปที่ 3.6 แสดงขั้นตอนการเปรียบเทียบข้อมูลด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

4. หาค่าจำนวนบิตที่เกิดความผิดพลาดของ $(y_1, y_2, ..., y_n)$ ด้วยชุดรหัสแก้ไขข้อผิดพลาด $x_1, x_2, ..., x_n$
5. ตรวจสอบข้อมูลจำนวนบิตที่เกิดความผิดพลาดหาค่า จำนวนบิตที่มีการผิดพลาดบ่อยครั้งที่สุด
6. ปรับปรุงค่า O เท่ากับจำนวนบิตที่มีการผิดพลาดบ่อยครั้งที่สุด เป็นค่า v
7. เปรียบเทียบค่า H(v) กับค่า H(S)

หาก

$$H(v) = H(S) \tag{3.24}$$

แล้ว

$$O = S \tag{3.25}$$

ข้อมูลทั้งสองข้อมูลเป็นข้อมูลเดียวกัน

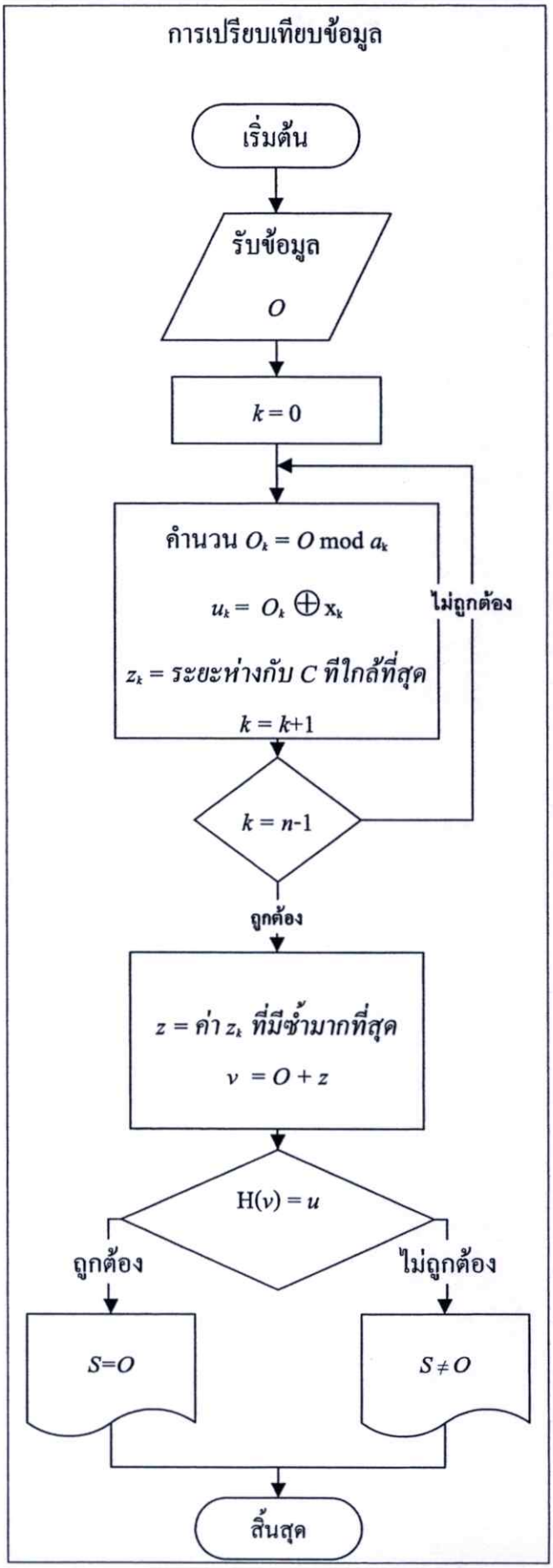
หาก

$$H(u) \neq H(S) \tag{3.26}$$

แล้ว

$$O \neq S \tag{3.27}$$

ข้อมูลทั้งสองค่าไม่ใช่ข้อมูลเดียวกัน



รูปที่ 3.7 แสดงขั้นตอนการเปรียบเทียบข้อมูลด้วยการใช้ค่าคะแนนส่วนใหญ่

3.3.5 ตัวอย่างการลงทะเบียนด้วยพันธสัญญาแบบคลุมเครือด้วยทฤษฎีบทเศษเหลือแบบจีน

ข้อมูลความลับ S= 23

ชุดรหัสแก้ไขข้อผิดพลาด x = {00000, 10110, 01011, 11101} ซึ่งสามารถแก้ไขข้อผิดพลาด 1 บิต

ชุดจำนวนเฉพาะสัมพัทธ์ a = {5, 7}

คำนวนหา

$S \bmod a_1 = 23 \bmod 5 = 3$

$S \bmod a_2 = 23 \bmod 7 = 2$

เปลี่ยนให้อยู่ในเลขฐานหนึ่ง

$S_1 = 00111$

$S_2 = 00011$

คำนวนหา

$c_1 = S_1 \oplus x_3 = 00111 \oplus 01011 = 01100$

$c_2 = S_2 \oplus x_2 = 00011 \oplus 10110 = 10101$

เก็บค่า {c = {01100, 10101}, a = {5, 7}, x = {00000, 10110, 01011, 11101}, H(S)} เพื่อใช้สำหรับเปรียบเทียบข้อมูล

3.3.6 ตัวอย่างการเปรียบเทียบข้อมูลด้วยพันธสัญญาแบบคลุมเครือด้วยทฤษฎีบทเศษเหลือแบบจีน

ข้อมูลสำหรับเปรียบเทียบ O = 22 ซึ่งมีค่าใกล้เคียงกับข้อมูลความลับ

ชุดรหัสแก้ไขข้อผิดพลาด x = {00000, 10110, 01011, 11101}

ชุดจำนวนเฉพาะสัมพัทธ์ a = {5, 7}

c = {01100, 10101}

คำนวนหา

$O \bmod a_1 = 22 \bmod 5 = 2$

$O \bmod a_2 = 22 \bmod 7 = 1$

เปลี่ยนให้อยู่ในเลขฐานหนึ่ง

$o_1 = 00011$

$o_2 = 00001$

คำนวนหา

$y_1 = o_1 \oplus c_1 = 00011 \oplus 01100 = 01111$

$y_2 = o_2 \oplus c_2 = 00001 \oplus 10110 = 10111$

$y_1 = \text{correct}(y_1) = \text{correct}(01111) = 01011$

$y_2 = \text{correct}(y_2) = \text{correct}(10111) = 10110$

$u_1 = y_1 \oplus c_1 = 01011 \oplus 01100 = 00111 = 3$

$u_2 = y_2 \oplus c_2 = 10110 \oplus 10101 = 00011 = 2$

ใช้ทฤษฎีบทเศษเหลือแบบจีนหาผลลัพธ์ของ

$u \bmod 5 = 3$

$u \bmod 7 = 2$

หาค่า x_1, x_2

$x_1 \bmod 5 = 3$

$x_1 \bmod 7 = 0$

$x_2 \bmod 5 = 0$

$x_2 \bmod 7 = 2$

$x_1 \bmod 5 = 3$

$x_1 \bmod 7 = 0$

$n = 5x7 = 35$

$n_1 = 35/5 = 7$

หา Inverse ของ 7 mod 5

คำตอบเท่ากับ 3

$7x3 \bmod 5 = 1$

$a_1 = 3$

$7x3x3 \bmod 5 = 3$

$x_2 \bmod 5 = 0$

$x_2 \bmod 7 = 2$

$n = 5x7 = 35$

n2 = 35/7 = 5

หา Inverse ของ 5 mod 7

คำตอบเท่ากับ 3

5x3 mod 7 = 1

$a_2 = 2$

5x3x2 mod 7 = 2

คำตอบคือ 7x3x3 + 5x3x2 mod 35

= 63 + 30 mod 35

= 23

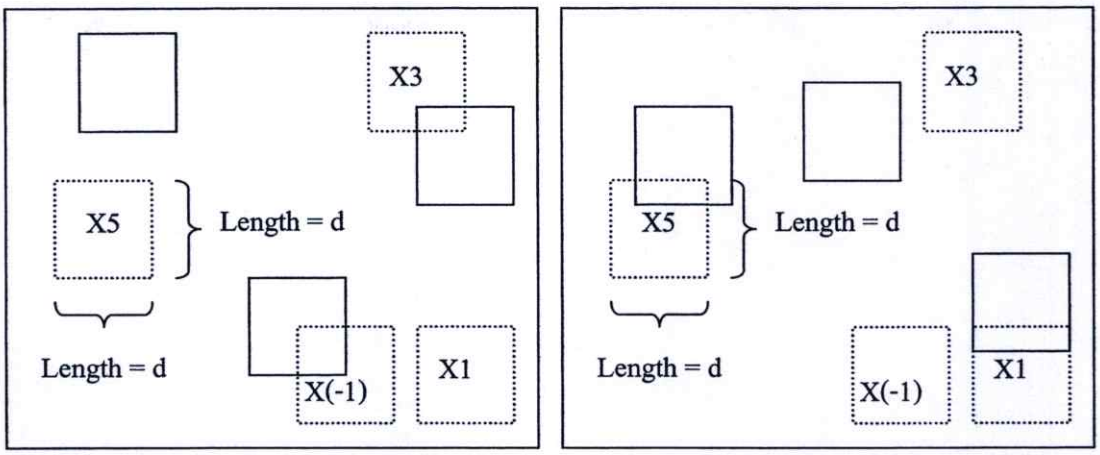
เมื่อค่า $u = 23$

นำค่า u ผ่านฟังก์ชันแฮช และนำเปรียบเทียบกับค่าแฮชของข้อมูลความลับที่เก็บไว้จะได้

$H(u) = H(S)$

3.4 การประยุกต์ใช้อัลกอริทึมการเปรียบเทียบวัตถุโดยพันธสัญญาแบบคลุมเครือ

อัลกอริทึมนี้สามารถที่จะพัฒนาให้สามารถเพิ่มค่าน้ำหนักความสำคัญของรูปแต่ละรูปที่ใช้เปรียบเทียบได้ เช่น ข้อมูลในรูปที่ 3.8 แสดงการเปรียบเทียบข้อมูลสองมิติ ทุกๆสี่เหลี่ยมสามารถมีน้ำหนักความสำคัญของการเปรียบเทียบที่ไม่เท่ากัน ซึ่งค่าน้ำหนักความสำคัญอาจจะมีค่าที่ติดลบได้ คือ {5, 3, 1,-1} เมื่อมีการวางรูปสี่เหลี่ยมเพื่อทำการทดสอบ จะมีการคูณค่าน้ำหนักความสำคัญของสี่เหลี่ยมแต่ละรูปเข้าไปด้วยจากรูปที่ 3.8 มีการเปรียบเทียบข้อมูลทั้งหมด 2 ครั้ง ในครั้งแรกได้รับคะแนนจากการเปรียบเทียบ 2 คะแนน แต่ในครั้งที่ 2 ได้คะแนนจากการเปรียบเทียบ 6 คะแนนพบว่าครั้งที่สองมีการเปรียบเทียบที่ถูกต้องมากกว่าครั้งแรก



รูปที่ 3.8 การกำหนดน้ำหนักของข้อมูลรูปสี่เหลี่ยมสองมิติ

บทที่ 4
การวิเคราะห์และพิสูจน์ความปลอดภัย

บทนี้กล่าวถึงการพิสูจน์ความปลอดภัยของอัลกอริทึมเมื่อเปรียบเทียบกับการใช้พันธสัญญาแบบคลุมเครือซึ่งเป็นการเปรียบเทียบจะทำการเปรียบเทียบในด้านประสิทธิภาพของอัลกอริทึมในด้านเวลาในการทำงานและความปลอดภัยของอัลกอริทึมเมื่อถูกโจมตีด้วยวิธีการนำสมาชิกข้อมูลที่เป็นไปได้ทั้งหมดมาเข้าทดสอบในสมการเพื่อหาค่าความลับที่ถูกต้องเพื่อจะแสดงให้เห็นถึงจุดอ่อนของการใช้งานพันธสัญญาแบบคลุมเครือและผลลัพธ์ของการปรับปรุงอัลกอริทึมให้มีความปลอดภัยมากขึ้นผลของการปรับปรุงอัลกอริทึมดังแสดงในตารางที่ 4.1

4.1 การวิเคราะห์และเปรียบเทียบประสิทธิภาพการทำงานระหว่าง พันธสัญญาแบบคลุมเครือและพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

การเปรียบเทียบประสิทธิภาพการทำงานระหว่างสองอัลกอริทึมคือ พันธสัญญาแบบคลุมเครือและพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนสามารถทำได้ด้วยการนำอัลกอริทึมทั้งสองมาแสดงขั้นตอนการทำงานเป็นรหัสเทียมนำมาวิเคราะห์เวลาการทำงานเพื่อใช้ในการพยากรณ์เวลาที่ต้องใช้ในการทำงานของอัลกอริทึม การพยากรณ์เวลาการทำงานของทั้งสองอัลกอริทึมมีรายละเอียดดังนี้

4.1.1 การวิเคราะห์เวลาการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

จากการขั้นตอนของอัลกอริทึมพันธสัญญาแบบคลุมเครือมาเขียนเป็นรหัสเทียมทำให้ได้รหัสเทียมสำหรับการลงทะเบียนและการเปรียบเทียบข้อมูลดังรูป 4.1 และ รูป 4.2 ตามลำดับ

ในการการวิเคราะห์จะกำหนดให้ คำสั่งพื้นฐานเช่น รับข้อมูล สุ่มเลือก เปรียบเทียบข้อมูลใช้เวลาการทำงานเท่ากับ a เมื่อนำรหัสเทียมมาของการลงทะเบียนมาวิเคราะห์เวลาการทำงานพบว่าสามารถเขียนเป็นเวลาการทำงานของอัลกอริทึมได้ตามสมการ

เวลาการทำงาน $t = 6a$

เมื่อ a คือเวลาการทำงานของคำสั่งพื้นฐาน

ตารางที่ 4.1 แสดงการเปรียบเทียบคุณสมบัติของอัลกอริทึมแบบต่างๆ

อัลกอริทึม	ข้อมูลที่เก็บ	เวลาการทำงาน	ข้อจำกัด	ข้อดี	วิธีการโจมตีหาความลับ	เวลาที่ใช้ในการโจมตี
พันธสัญญาแบบคลุมเครือ[2]	$H(c)$, $S \oplus c$ C	O(c)	1. เปรียบเทียบข้อมูลที่เป็นบิตเท่านั้น 2. ลำดับของข้อมูลต้องเรียงถูกต้องเท่านั้น 3. สามารถถูกโจมตีด้วย ชุดรหัสแก้ไขข้อผิดพลาดทั้งหมด	1. สามารถทำงานได้อย่างรวดเร็ว	ใช้ข้อมูลของรหัสแก้ไขข้อผิดพลาดทั้งหมด	$O(x)$
พันธสัญญาแบบคลุมเครือด้วยการใช้เลขฐานหนึ่ง	$H(c)$, $S \oplus c,C$	O(c)	1. สามารถถูกโจมตีด้วยการใช้ข้อมูลเลขฐานหนึ่งทั้งหมดได้	1. สามารถทำการเปรียบเทียบข้อมูลที่มีลักษณะเป็นช่วงข้อมูลได้	ใช้ข้อมูลเลขฐานหนึ่งทั้งหมด	$O(\log_2 n)$
พันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีน	$H(s)$, C, $x[n]$, $a[n]$	O(n)	1. ใช้ระยะเวลาในการทำงานที่มากขึ้น	1. มีความปลอดภัยจากการโจมตีด้วยชุดรหัสแก้ไขข้อผิดพลาดที่มากขึ้น	ใช้ข้อมูลชุดจำนวนสัมพัทธ์และรหัสแก้ไขข้อผิดพลาดทั้งหมด	$O(x^n)$
				2. สามารถทำการเปรียบเทียบข้อมูลที่มีลักษณะเป็นช่วงข้อมูลได้	ให้ค่าข้อมูลความลับที่เป็นไปได้ทั้งหมด	$O(n)$

เมื่อ n คือ จำนวนข้อมูลความลับที่เป็นไปได้, x คือ จำนวนรหัสในชุดรหัสแก้ไขข้อผิดพลาด, a คือ จำนวนข้อมูลในชุดจำนวนเฉพาะสัมพัทธ์

```
Procdure COMMITMENT ( )
var codeword[x],W,Secret,Encrypdata,Hashdata
        INPUT Secret
        Create codeword[x]
        Random Select W from codeword
        encrypdata = w xor Secret
                        Hashdata = hash(w)
        return (codeword[x],hashdata,encrypdata)
```

รูปที่ 4.1 แสดงรหัสเทียมการลงทะเบียนของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

```
Procdure DECOMMITMENT (codeword[R],hashdata,encrypdata)
var Challenge,w
        Input Challenge
        w =  encrypdata xor Challenge
        w = nearest w in codeword[x]
        w = hash(w)
if w==hashdata then
        return success
else    return failure
```

รูปที่ 4.2 แสดงรหัสเทียมการเปรียบเทียบข้อมูลของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

การวิเคราะห์เวลาการทำงานรหัสเทียมของเปรียบเทียบข้อมูลแล้วพบว่าเวลาการทำงานของการเปรียบเทียบข้อมูลอยู่ในรูปสมการ

$$\text{เวลาการทำงาน} = 5a$$

เมื่อ a คือเวลาการทำงานของคำสั่งพื้นฐาน

เมื่อทำการพิจารณาพฤติกรรมการเติบโตทางเวลาทั้งสองการทำงานของอัลกอริทึมให้อยู่ในรูปฟังก์ชันทางคณิตศาสตร์พบว่าสามารถจัดให้อยู่ในรูป

$$f(n) = 6a \tag{4.1}$$

แล้ว

$$6a \leq 6a * 1 \tag{4.2}$$

$$f(n) \leq 6a * 1 \tag{4.3}$$

ดังนั้นสามารถสรุปได้ว่า

$$f(n) = O(c) \tag{4.4}$$

จากการวิเคราะห์อัลกอริทึมพันธสัญญาแบบคลุมเครือสามารถสรุปได้ว่าเวลาที่ใช้ในการทำงานของอัลกอรึทึมทั้งการลงทะเบียนและการเปรียบเทียบ อยู่ในค่า O (c)

4.1.2 การวิเคราะห์เวลาการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือ ด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

จากการขั้นตอนของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีนมาเขียนเป็นรหัสเทียมทำให้ได้รหัสเทียมสำหรับการลงทะเบียนและการเปรียบเทียบข้อมูลดังรูป 4.3 และ รูป 4.4 ตามลำดับ

ในการการวิเคราะห์จะกำหนดให้ คำสั่งพื้นฐานเช่น รับข้อมูล สุ่มเลือก เปรียบเทียบข้อมูล ใช้เวลาการทำงานเท่ากับ a เมื่อนำรหัสเทียมตามรูปที่ 4.3 ของการลงทะเบียนมาวิเคราะห์เวลาการทำงานพบว่าสามารถเขียนเป็นเวลาการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนได้ตามสมการ

$$f(n) = (3a)n+5a \tag{4.5}$$

แล้ว

$$(3a)n+5a \leq (4a)n \tag{4.6}$$

$$f(n) \leq (4a)n \tag{4.7}$$

ดังนั้นสามารถสรุปได้ว่า

$$f(n) = O(n) \tag{4.8}$$

```
Procdure COMMITMENT ( )
var a[n],Secret,hash_of_Secret,S[n],Encrypdata[n]
Create codeword[x]
hash_of_Secret = hash(Secret)
create a[n]
        for i=0 to n-1
                s[i] = Secret mod a[i]
                w = Random Select codeword[x]
                Encrypdata[i] = s[i] xor w
        end for
```

รูปที่ 4.3 แสดงรหัสเทียมการลงทะเบียนของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

เมื่อนำรหัสเทียมตามรูปที่ 4.4 ของการเปรียบเทียบข้อมูลมาวิเคราะห์เวลาการทำงานพบว่าสามารถเขียนเป็นเวลาการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนได้ตามสมการ

$$f(n) = (4a)n+7a \tag{4.9}$$

แล้ว

$$(4a)n+7a \leq (5a)n \tag{4.10}$$

$$f(n) \leq (5a)n \tag{4.11}$$

ดังนั้นสามารถสรุปได้ว่า

$$f(n) = O(n) \tag{4.12}$$

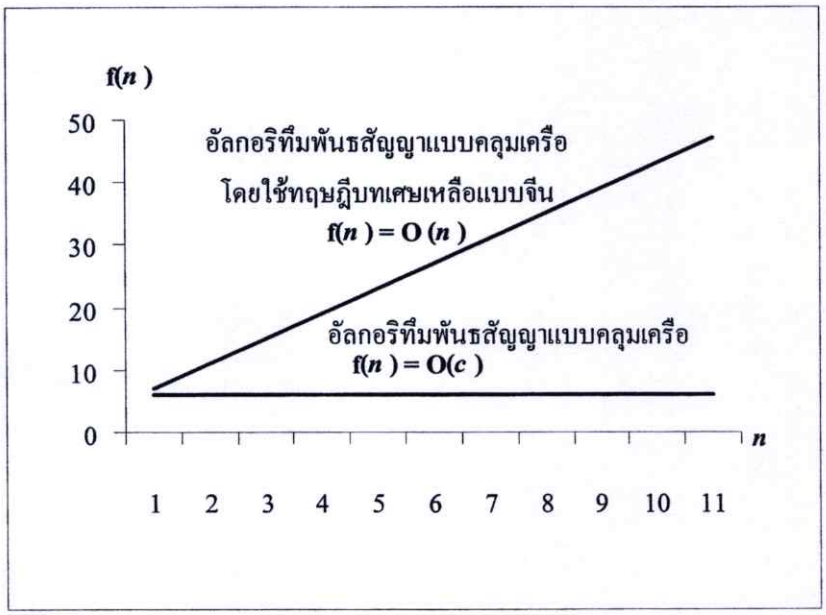
```
Procdure DECOMMITMENT
(a[n],Encrypdata[n],codeword[x],hash_of_Secret)
var Challenge,w[n],o[n],Secret
        Input Challenge
        for i=0 to n-1
                o[i] = Challenge mod a[i]
                w[i] = o[i] xor Encrypdata[i]
                w[i] = nearest w in codeword[x]
                o[i] = w[i] xor Encrypdata[i]
        end for
        Challenge = Chainese (o[n],a[n])
        hash_of_Challenge = hash (Challenge)
        if hash_of_Challenge == hash_of_Secret then
                return success
        else    return failure
```

รูปที่ 4.4 แสดงรหัสเทียมการเปรียบเทียบข้อมูลของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

จากข้อมูลขั้นต้นสรุปได้ว่าเวลาที่ใช้ในการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน ใช้เวลาในการทำงานของขั้นตอนการลงทะเบียนและขั้นตอนการเปรียบเทียบข้อมูลอยู่ในรูปของฟังก์ชัน $f(n) = O(n)$

4.1.3 การเปรียบเทียบอัลกอริทึมพันธสัญญาแบบคลุมเครือและพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

เมื่อทำการวิเคราะห์เวลาการทำงานของทั้งสองอัลกอริทึมแล้ว พบว่าอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนใช้เวลาในการทำงาน $f(n) = O(n)$ ซึ่งมากกว่าอัลกอริทึมพันธสัญญาแบบคลุมเครือที่ใช้เวลา $f(n) = O(c)$ ดังแสดงในภาพแสดงการเปรียบเทียบในรูปที่ 4.5



รูปที่ 4.5 แสดงการเปรียบเทียบเวลาการทำงานของอัลกอริทึมพันธสัญญาแบบคลุมเครือและอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

4.2 การวิเคราะห์และเปรียบเทียบความปลอดภัย

ในการเปรียบเทียบประสิทธิภาพความปลอดภัยของอัลกอริทึมนั้นคือการทดสอบการปลอดภัยจากการถูกโจมตีเพื่อหาข้อมูลความลับวิธีการที่นิยมใช้ในการวิเคราะห์ความปลอดภัยคือหาจำนวนครั้งที่ต้องใช้ในการนำข้อมูลที่เป็นไปได้มาทดสอบเพื่อหาความลับของข้อมูล หากข้อมูลที่เป็นไปได้มีจำนวนค่ามาก จำนวนครั้งในการทดลองหาข้อมูลจะมีจำนวนครั้งมากอัลกอริทึมจะมีความปลอดภัยเนื่องจากต้องใช้เวลาและทรัพยากรในการทดสอบเพื่อหาความลับสูง จะทำให้อัลกอริทึมมีความปลอดภัย หากข้อมูลที่เป็นไปได้มีจำนวนน้อยในการทดสอบเพื่อหาความลับความปลอดภัยจะน้อย

ในการวิเคราะห์ความปลอดภัยของอัลกอริทึมจะต้องทำการเปรียบเทียบกับการโจมตีโดยใช้ข้อมูลความลับทั้งหมดซึ่งจะมีจำนวนครั้งในการโจมตีเท่ากับจำนวนข้อมูลความลับที่เป็นไปได้ หรือ $f(n) = O(n)$ หากอัลกอริทึมสามารถโจมตีด้วยวิธีการที่ใช้จำนวนครั้งน้อยกว่าจะทำให้ความปลอดภัยของข้อมูลความลับลดลง แต่หากอัลกอริทึมไม่สามารถใช้วิธีการอื่นในการโจมตีที่ให้เวลาที่ดีกว่าการใช้ข้อมูลความลับทั้งหมดที่เป็นไปได้แล้วผู้โจมตีจะใช้เวลาในการโจมตีมากที่สุดเป็น $f(n) = O(n)$

4.2.1 การวิเคราะห์ความปลอดภัยของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

อัลกอริทึมพันธสัญญาแบบคลุมเครือมีการเก็บข้อมูลที่ใช้สำหรับการเปรียบเทียบประกอบไปด้วย

- ชุดรหัสแก้ไขข้อผิดพลาด
- ค่าแฮชของรหัสแก้ไขข้อผิดพลาดที่ถูกเลือก
- ค่าผลรวมระหว่าง ความลับและรหัสแก้ไขข้อผิดพลาด ($S \oplus w$)

จากข้อมูลที่มีการเก็บไว้ทำให้สามารถวิเคราะห์ได้ว่าหากต้องการหาข้อมูลความลับจำเป็นต้องหาข้อมูลรหัสแก้ไขข้อผิดพลาดที่ถูกต้องซึ่งรหัสแก้ไขข้อผิดพลาดที่ถูกต้องนั้นอยู่ในชุดรหัสแก้ไขข้อผิดพลาด ขั้นตอนการโจมตีจึงสามารถทำได้ด้วยการนำชุดรหัสแก้ไขข้อผิดพลาดทุกค่ามาทำการหาค่าแฮช และนำมาเปรียบเทียบกับค่า ค่าแฮช ของรหัสแก้ไขข้อผิดพลาดที่ถูกเลือก เมื่อพบค่า รหัสแก้ไขข้อผิดพลาดที่ให้ค่าแฮชเท่ากับ ค่าแฮช ของรหัสแก้ไขข้อผิดพลาดที่ถูกเลือก รหัสแก้ไขข้อผิดพลาดนั้นลบออกจาก ค่า ผลรวมระหว่าง ความลับและรหัสแก้ไขข้อผิดพลาด จะทำให้ได้ความลับที่ถูกต้องรายละเอียดตามรูป 4.6

จากขั้นตอนการโจมตีจะทำให้ได้ข้อมูลจำนวนครั้งมากที่สุดในการโจมตีทั้งหมดเท่ากับจำนวนของสมาชิกของชุดข้อมูลแก้ไขข้อผิดพลาดซึ่งเขียนเป็นสมการได้

$$f(n) = x \tag{4.13}$$

แล้ว

$$f(n) = O(x) \tag{4.14}$$

เมื่อ $f(n)$ คือ จำนวนครั้งที่ต้องทำการทดสอบ

x คือ จำนวนสมาชิกชุดรหัสแก้ไขข้อผิดพลาด

```
Procedure BRUTEFORCECOMMITMENT (codeword[n],hashdata,encrypdata)
var i,Secret ,
for i = 0 to n
   w = hash (codeword[i])
        if w == hashdata then
                Secret = codeword[i] xor encrypdata
                i = n
                return (Secret)
        end if
end for
```

รูปที่ 4.6 แสดงรหัสเทียมการโจมตีด้วยการใช้ข้อมูลทั้งหมดของอัลกอริทึมพันธสัญญาแบบคลุมเครือ

ความปลอดภัยของอัลกอริทึมพันธสัญญาแบบคลุมเครือจึงขึ้นอยู่กับจำนวนของสมาชิกในชุดรหัสแก้ไขข้อผิดพลาดหากจำนวนสมาชิกมีปริมาณมากจะทำให้ความปลอดภัยของอัลกอริทึมมีมากขึ้น แต่จำนวนบิตของชุดรหัสแก้ไขข้อผิดพลาดจะต้องมีจำนวนที่เท่ากับจำนวนบิตของข้อมูลความลับซึ่งส่งผลให้จำนวนชุดรหัสแก้ไขข้อผิดพลาดมีจำนวนน้อยกว่าจำนวนข้อมูลความลับ

$$x < n \tag{4.15}$$

ดังนั้นเมื่อมีการเปรียบเทียบความปลอดภัยกับการใช้ข้อมูลทั้งหมดจะได้ดังสมการ

$$O(2^x) < O(2^n) \tag{4.16}$$

ดังนั้นผู้ที่ทำการโจมตีหาข้อมูลความลับจะเลือกการโจมตีด้วยการใช้ข้อมูลชุดรหัสแก้ไขข้อผิดพลาดทั้งหมดที่เป็นไปได้ในการโจมตี ($f(n) = O(x)$) เพื่อสามารถโจมตีได้รวดเร็วที่สุด

4.2.2 การวิเคราะห์ความปลอดภัยของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

อัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนมีการเก็บข้อมูลที่ใช้สำหรับการเปรียบเทียบประกอบไปด้วย

- ชุดรหัสแก้ไขข้อผิดพลาด
- ค่าแฮช ของข้อมูลความลับ
- ชุดจำนวนเฉพาะสัมพัทธ์
- ชุด ผลรวมระหว่างความลับย่อยทุกตัวและรหัสแก้ไขข้อผิดพลาดที่ถูกเลือก ($S_i \oplus w_i$)

จากข้อมูลที่มีการเก็บไว้ทำให้สามารถวิเคราะห์ได้ว่าหากต้องการหาข้อมูลความลับสามารถทำการโจมตี ได้สองวิธีคือ หาข้อมูลความลับทั้งหมดเพื่อที่จะสามารถใช้ทฤษฎีบทเศษเหลือแบบจีนได้ซึ่งการหาความลับทั้งหมดนั้นจะต้องมีการจับคู่ ชุดรหัสแก้ไขข้อผิดพลาดเพื่อแทนลงใน ชุดผลรวมระหว่างความลับย่อยทุกตัวและรหัสแก้ไขข้อผิดพลาดที่ถูกเลือก ให้ถูกต้องทุกค่า และ อีกวิธีการหนึ่งคือโจมตีที่ข้อมูลที่เป็นความลับทั้งหมด

จากวิธีการโจมตีอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนทำให้สามารถหาจำนวนข้อมูลที่เป็นไปได้ทั้งหมดที่จะต้องนำมาใช้ทดสอบเพื่อหาข้อมูลความลับอยู่ในรูปสมการ

$$f(n) = x^a \tag{4.17}$$

แล้ว

$$f(n) = O(x^a) \tag{4.18}$$

เมื่อ $f(n)$ คือ จำนวนครั้งที่ต้องทำการทดสอบ

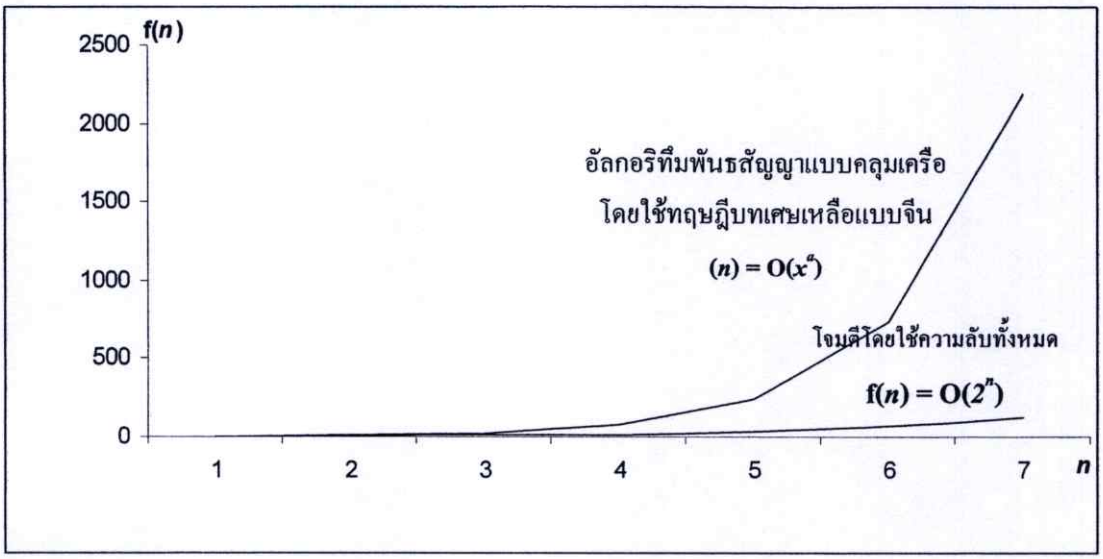
x คือ จำนวนสมาชิกชุดรหัสแก้ไขข้อผิดพลาด

a คือ จำนวนสมาชิกในชุดจำนวนเฉพาะสัมพัทธ์

ความปลอดภัยของอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนเมื่อนำมาเปรียบเทียบกับการโจมตีด้วยการใช้ข้อมูลความลับทั้งหมดพบว่าการโจมตีจะให้เวลามากกว่าการโจมตีด้วยการใช้ความลับทั้งหมด

$$O(x^a) > O(n) \tag{4.19}$$

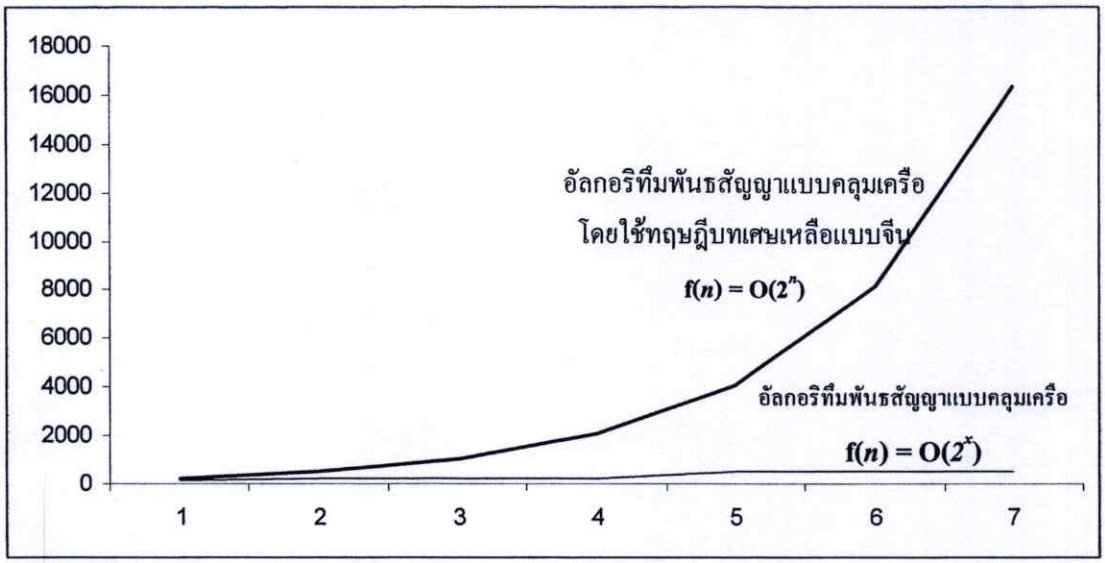
ดังนั้นหากผู้โจมตีจะทำการโจมตีหาข้อมูลความลับจะเลือกวิธีการใช้ข้อมูลความลับที่เป็นไปได้ทั้งหมดในการโจมตี ซึ่งใช้เวลาในการโจมตีทั้งหมด $f(n) = O(2^n)$ ดังแสดงในรูป 4.7



รูปที่ 4.7 แสดงการเปรียบเทียบจำนวนข้อมูลที่ใช้ในการโจมตีโดยใช้ข้อมูลความลับทั้งหมดและอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

4.2.3 การเปรียบเทียบอัลกอริทึมพันธสัญญาแบบคลุมเครือและพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีน

เมื่อทำการวิเคราะห์ความปลอดภัยของทั้งสองอัลกอริทึมแล้วพบว่า อัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีบทเศษเหลือแบบจีนมีความปลอดภัยมากกว่าอัลกอริทึมพันธสัญญาแบบคลุมเครือ เนื่องจากการใช้อัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยทฤษฎีบทเศษเหลือแบบจีนนั้นผู้โจมตีจะใช้วิธีการหาข้อมูลความลับที่เป็นไปได้ทั้งหมดในการโจมตี แต่อัลกอริทึมพันธสัญญาแบบคลุมเครือผู้โจมตีจะสามารถโจมตีโดยการใช้ค่าชุดรหัสแก้ไข้อผิดพลาดทั้งหมดที่เป็นไปได้ซึ่งใช้จำนวนครั้งในการโจมตีที่น้อยกว่าดังแสดงในสมการที่ 4.19 และ รูป 4.8



รูปที่ 4.8 แสดงการเปรียบเทียบจำนวนข้อมูลที่ใช้ในการโจมตีของอัลกอริทึมพันธสัญญาแบบคลุมเครือและอัลกอริทึมพันธสัญญาแบบคลุมเครือด้วยการใช้ทฤษฎีเศษเหลือแบบจีน

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ปัจจุบันความปลอดภัยของข้อมูลเป็นเรื่องที่มีความสำคัญมาก ข้อมูลที่สำคัญต้องได้รับการดูแลและจัดเก็บในรูปแบบที่เหมาะสมยากต่อการที่จะถูกโจรกรรมหรือขโมยได้ การนำข้อมูลที่ใช้สำหรับพิสูจน์ความถูกต้องจะเก็บไว้ในรูปแฮชฟังก์ชันเพื่อการรักษาความปลอดภัยของข้อมูลนั้น เช่น การเก็บรักษารหัสผ่าน หรือการจับคู่ข้อมูลที่เหมือนกัน แต่ในบางครั้งระบบข้อมูลที่นำมาตรวจสอบทั้งสองข้อมูลอาจจะไม่เหมือนกันทั้งหมดโดยยอมให้มีความผิดพลาดในขนาดที่ยอมรับได้ ซึ่งต้องมีการปรับปรุงวิธีการจัดเก็บให้สามารถทำงานในลักษณะดังกล่าวได้

อัลกอริทึมพันธสัญญาแบบคลุมเครือเป็นวิธีการจัดเก็บข้อมูลเพื่อใช้ในการเปรียบเทียบวิธีการหนึ่งซึ่งยอมให้สามารถมีข้อผิดพลาดได้ด้วยการใช้วิธีการของการแก้ไขข้อผิดพลาดช่วยในการทำงาน หากข้อมูลที่นำมาเปรียบเทียบมีความผิดพลาดในปริมาณที่สามารถทำการแก้ไขข้อผิดพลาดได้ จะมีการแก้ไขข้อผิดพลาดก่อนการนำข้อมูลไปเปรียบเทียบ ถึงแม้ว่าอัลกอริทึมพันธสัญญาแบบคลุมเครือจะสามารถทำงานได้เร็วโดยใช้เวลาในการทำงาน $f(n)= O(c)$ แต่อัลกอริทึมนี้ ยังมีจุดอ่อนที่ผู้ประสงค์ร้ายสามารถโจมตีได้ด้วยการนำชุดรหัสแก้ไขข้อผิดพลาดทั้งหมดมาหาค่าแฮชเพื่อหาข้อมูลที่ถูกต้อง ซึ่งใช้เวลาในการโจมตีอยู่ในรูป $f(n) = O(n)$ ข้อจำกัดอีกประการหนึ่งของอัลกอริทึมพันธสัญญาแบบคลุมเครือคือ การทำงานสามารถทำงานกับข้อมูลที่เป็นตัวเลขเท่านั้นไม่สามารถทำงานกับข้อมูลที่เป็นเส้นหรือรูปทรงวัตถุได้

วิทยานิพนธ์นี้ได้นำเสนอเทคนิคการเปรียบเทียบวัตถุด้วยพันธสัญญาแบบคลุมเครือที่จะสามารถใช้ในการตรวจสอบข้อมูลที่มีลักษณะต่างกันเล็กน้อยว่าเป็นข้อมูลเดียวกันหรือไม่ซึ่งได้มีการพัฒนาให้สามารถทำงานกับข้อมูลที่มีลักษณะ เป็นข้อมูลสองมิติและสามมิติ ด้วยการกำหนดหาจุดที่เป็นจุดอ้างอิงที่ใช้สำหรับการตรวจสอบและแปลงข้อมูลจากตัวเลขเป็นข้อมูลเลขฐานหนึ่งเพื่อให้สามารถทำงานกับข้อมูลได้แต่การทำงานกับข้อมูลที่เป็นเลขฐานหนึ่งยังมีปัญหาเรื่องความปลอดภัยของการใช้ข้อมูลทั้งหมดในการหาข้อมูลที่ถูกต้อง จึงได้มีการปรับปรุงอัลกอริทึมพันธสัญญาแบบคลุมเครือให้มีความปลอดภัยของการเก็บข้อมูลที่มากขึ้น ด้วยการนำทฤษฎีบทเศษเหลือแบบจีนมาใช้เพื่อการแบ่งแยกข้อมูลความลับออกเป็นส่วนๆ แต่ละส่วนมีการเลือกใช้ชุดรหัสแก้ไขข้อผิดพลาดคนละตำแหน่งกัน ทำให้เมื่อผู้ประสงค์ร้ายจะทำการจู่โจมเพื่อหาข้อมูลที่เป็นความลับจำเป็นจะต้องจับคู่ข้อมูลย่อยและรหัสแก้ไขข้อผิดพลาดให้ถูกต้องทุกคู่จึงจะสามารถทำการหาข้อมูลความมลับได้ ซึ่งต้องใช้เวลาในการหาอยู่ในรูปฟังก์ชัน $f(n) = O(n^a)$ เมื่อ a คือจำนวนข้อมูลความลับย่อย แต่การปรับปรุงข้อมูลด้วยการแบ่งข้อมูลความลับจะทำให้เวลาที่ใช้ใน

การทำงานของอัลกอริทึมเพิ่มขึ้นเป็น f(n)=O(n) ซึ่งใช้เวลามากกว่าการใช้อัลกอริทึมพันธสัญญาแบบคลุมเครือปกติ

วิธีการที่นำเสนอในวิทยานิพนธ์นี้จะช่วยปรับปรุงอัลกอริทึมของการเปรียบเทียบข้อมูลให้มีความปลอดภัยในการใช้งานที่มากขึ้น ถึงแม้ว่าจะต้องใช้เวลาในการทำงานที่มากขึ้นก็ตาม และสามารถทำงานได้กับข้อมูลที่เป็นเส้นหรือรูปภาพเรขาคณิต ได้ในอนาคตยังสามารถทำการพัฒนาให้ประสิทธิภาพการทำงานที่เร็วขึ้นจากเดิม และมีความสามารถในการเปรียบเทียบข้อมูลที่เป็นรูปภาพได้ สำหรับการนำอัลกอริทึมนี้ไปประยุกต์ใช้งานสามารถนำไปใช้งานได้กับระบบที่มีลักษณะของการเปรียบเทียบข้อมูลที่สามารถเกิดความคลาดเคลื่อนในการเปรียบเทียบได้เช่นการตรวจสอบข้อมูลชีวภาพ การจำคู่ข้อมูลระหว่างบุคคล หรือการเปรียบเทียบรูปภาพที่มีการเปลี่ยนแปลงไปเล็กน้อย

บรรณานุกรม

[1] A. Juels and M. Sudan, **"A Fuzzy Vault Scheme,"** in Proceedings of The IEEE International Symposium on Information Theory, p.408, Piscataway, NJ, USA, June-July 2002.

[2] A. Juels and M. Wattenberg, **"A Fuzzy Commitment Scheme,"** in Proceedings of 6th ACM Conference on Computer and Communications Security (ACM CCS '99), pp. 28–36,Singapore,November 1999.

[3] A. K. Jain, K. Nandakumar, and A. Nagar, **"Review Article Biometric Template Security,"** in EURASIP Journal on Advances in Signal Processing, Volume 2008, Michigan, USA, December 2007

[4] Bruce Schneier, **Applied Cryptography, Second Edition: Protocols, Algorithms and Source Code**. NEW YORK: John Wiley & Sons, Inc, 1996.

[5] J. L. Carter and M. N. Wegman, **"Universal Classes of Hash Functions,"** in Journal of Computer and System Sciences, vol. 18,no. 2, pp. 143–154, 1979.

[6] Katz, Jonathan, and Lindel, Yehuda. 2008. **Introduction to Modern Cryptography**. Boca Raton, FL: Chapman & Hall/CRC

[7] Mao, Wenbo. 2004. **Modern Cryptography: Theory & Practice**. Upper Saddle River, NJ: Prentice-Hall.

[8] S. Yang and I. M. Verbauwhede, **"Secure Fuzzy Vault based Fingerprint Verification System,"** in Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on, vol. 1, pp. 577-581, Nov 2004.

[9] T.R.N Rao andE. Fujiwara, **Error-control Coding for Computer Systems.** New Jersey: Prentice-Hall, Inc.1989.

[10] U. Uludang, S. Pankanti, S.Prabhakar and A. K.Jain. **"Biometric Cryptosystems: Issues and Challenges,"** in Proceedings of The IEEE, vol. 92, no. 6, pp.948-960, June 2004.

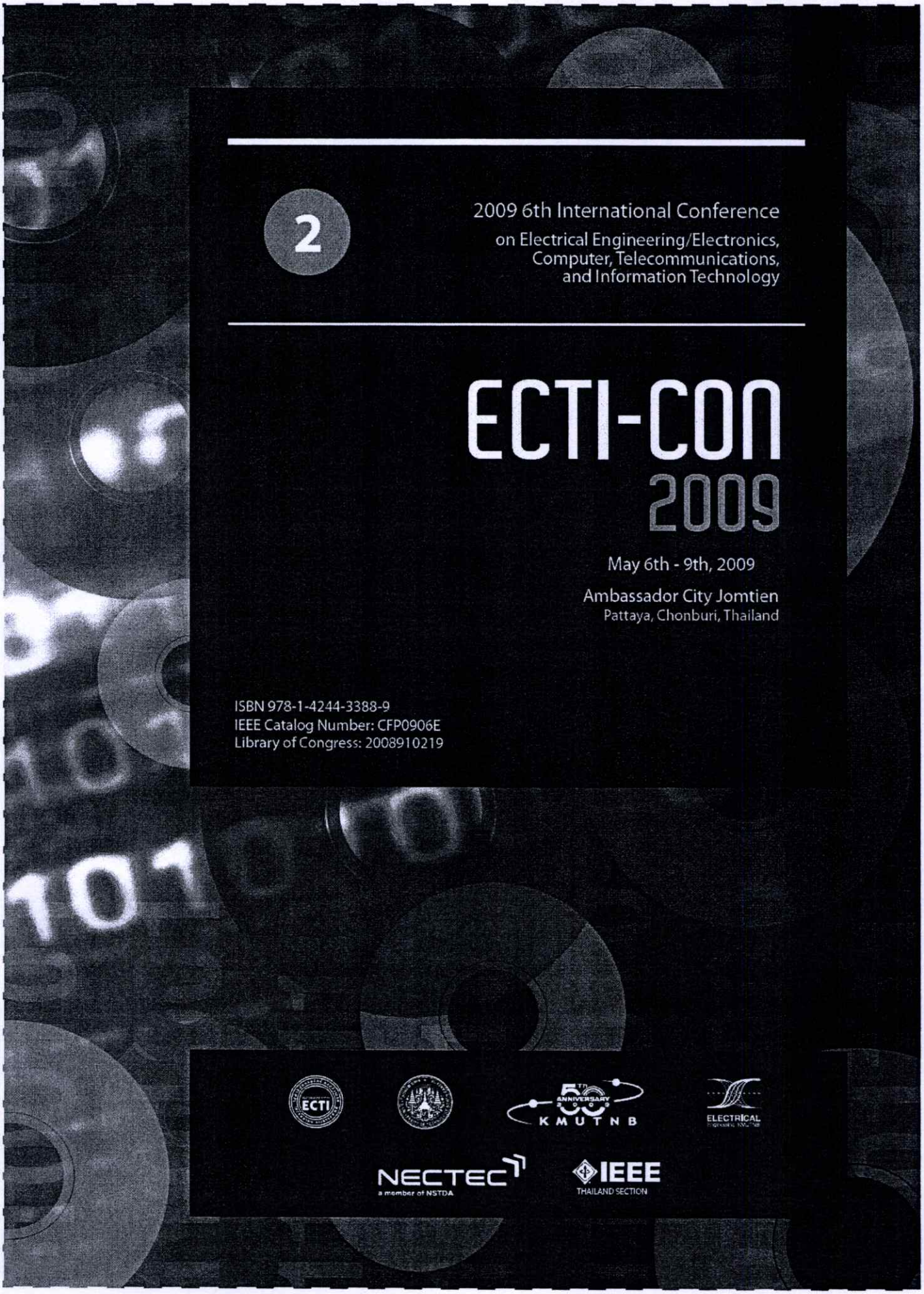
[11] U. Uludang, S. Pankanti, S.Prabhakar and A. K.Jain. **" Fuzzy Vault for Fingerprints "** in Proceedings of the workshop Biometrics: Challenges Arising from Theory and Practice, pp. 13-16, Cambridge UK, August 2004.

ภาคผนวก

ภาคผนวก ก.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. A. Satienjarurat and N. Premasathian **"Fuzzy Matching of Objects using Fuzzy Commitment,"** International Conference on Electrical Engineering/Elctronics, Computer, Telecommunications, and Nnformation Technology (ECTI-CON'2009), pp. 622-625, Pattaya, Thailand, May 6th-9th, 2009.


2
2009 6th International Conference
on Electrical Engineering/Electronics,
Computer, Telecommunications,
and Information Technology
ECTI-CON
2009
May 6th - 9th, 2009
Ambassador City Jomtien
Pattaya, Chonburi, Thailand
ISBN 978-1-4244-3388-9
IEEE Catalog Number: CFP0906E
Library of Congress: 2008910219
ECTI
5TH ANNIVERSARY
2004 2009
KMUTNB
ELECTRICAL
NECTEC
a member of NSTDA
IEEE
THAILAND SECTION

Fuzzy Matching of Objects using Fuzzy Commitment

A. Satienjarurat and N. Premasathian
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang.
Bangkok, Thailand 10520
Email: Marcus_sut@hotmail.com
Nol.p@kmitl.ac.th

Abstract- **This paper studies some fuzzy schemes that can be applied in matching algorithms. A fuzzy matching algorithm is constructed by applying the fuzzy commitment technique on unary numerical representation of object attributes. This enables the scheme to match objects based on their attributes in the order-invariant manner. In this paper, examples are given using sizes and locations of objects as attributes. The one dimensional and two dimensional examples of the scheme are given.**

I. INTRODUCTION

A commitment scheme is a protocol that the first user commits to some bits, called the commitment string, and the second user tries to determine it by providing another set of bits called the challenge string. The determination is successful if the two strings are exactly matched [1]. However, in some applications, the commitment scheme considers the determinations successful when they contain some limited amount of error such as biometric template [2]. Juels and Wattenburg introduced the fuzzy commitment scheme, a technique that combines cryptography with error correction that the challenge string can differ from the commitment string by some certain number of bits [3]. Since ordering of symbols in the string affects the mechanism of the fuzzy commitment scheme, there is another scheme developed by Juels and Sudan called fuzzy vault scheme [4]. This scheme allows the symbols in both strings to be order-invariant. Since the operation fuzzy vault scheme is in the level of points, each point to point matching must be exact. This can considerably reduce the accuracy of matching in some application [5]. This paper introduces a matching algorithm that applies the technique of the fuzzy commitment so that order-invariant fuzzy matching can be achieved in greater dimensions. The examples of matching in one dimension and two dimensions are given in this paper.

II. RELATED WORKS

There are two related schemes mentioned in the introduction, the first one is the fuzzy commitment and the second one is the fuzzy vault. The first scheme allows some error in the matching. It works as follows.

1. A codeword w is randomly selected from an error-correcting code C.
2. Exclusive-or the commitment s with w to get s⊕w.
3. Hash [6] w to get h(w).
4. The user is given s⊕w and h(w).
5. The user enters s' as a challenge.
6. The system computes s'⊕s⊕ w to get w'. If s' is close to s, then w' is also close to w and can be decoded to obtain the nearest codeword w. Its hash is compared with the stored h(*w*) for verification.

In the first scheme, all symbols are ordered. The permutation of symbols in any string causes inaccuracy in the matching. The second scheme differs from the first one as the committed information can be order-invariant. It works as follows.

1. Create a Reed-Solomon codeword [7] to represent the commitment. The codeword is computed over some x-coordinates corresponding to elements in a set.
2. The concealment is achieved by adding some irrelevant points to the set. This function is called the Lock function of the fuzzy vault.
3. To unlock the vault, one must provide an acceptable challenge. That is a set of points corresponding to the locked values. If the set does not contain more irrelevant points than the predefined limit, it can be corrected by the error correction of the Reed-Solomon coding scheme and the vault is unlocked.

As described above, not all points in the fuzzy vault scheme must be matched but each matching must be exact. Therefore we propose another scheme that is able to match points when the locations of the points can differ to some limited extent. The fuzzy commitment scheme is used in our work to create a fuzzy matching scheme in greater dimensions. Our scheme maintains the simplicity of the fuzzy commitment allows the information to be reordered as in the fuzzy vault but provides greater flexibility. The work is explained in details in the next section.

III. OUR SCHEME

In this section, we present our scheme, which can provide fuzzy matching in any dimension. We give the description of the scheme by two simplified examples of the matching in one dimension and two dimensions.

Computing machines mostly operate binary data in their basic operations. As binary data consists of two symbols, 0 and 1, our code can consist of 0 and 1 no matter if the numeral representation is a binary one or not. Our scheme uses the unary numerical representation in encoding the

978-1-4244-3388-9/09/$25.00 ©2009 IEEE

attribute values. Although the encoding results in a longer code, its characteristic enables it to be integrated into the error correcting part in the fuzzy commitment. That is, the number of distorted bits of the value in the attribute in unary numerical format reflects the amount of error occurred. Table 1 compares the number of error bits of the number 4 (00001111 in unary and 100 in binary) when unary and binary numerical representations are used. This advantage of the error correcting capability is valid for all number and can be easily verified.

Table 1 Number of error bits of the unary and binary numerical representations for number 4.

Decimal	Unary	Error in unary (bits)	Binary	Error in binary (bits)
0	00000000	4	000	1
1	00000001	3	001	2
2	00000011	2	010	2
3	00000111	1	011	3
4	00001111	0	100	0
5	00011111	1	101	1
6	00111111	2	110	1
7	01111111	3	111	2

In our one dimensional matching scheme, the platform consists of a line of length n, placements of k lines of length d, and the maximal match error e. The commitment is done by placing k lines of length d on it. After that, a user, without seeing the location of the placements of the first k lines, can place a challenge by putting other k lines of length d on the same line (of length n). If more than k-e pairs of lines overlap, the match is successful. An example of the platform of the scheme is depicted in figure 1. In this figure, the three lines as the dashed lines represent the commitment while the three solid lines represent the challenge. There are two pairs of lines that are matched. The details of scheme are described in the following procedures, the commitment and the matching.

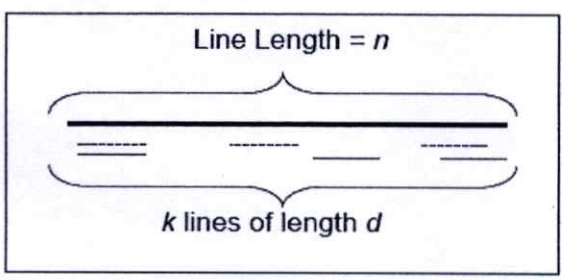


Figure 1. The platform of the one-dimensional fuzzy matching scheme.

In the commitment procedure,

The first line l_1 is placed at the position x_1, the second line l_2 is placed at the position x_2, and so on until the last line is placed at the position x_k. Since the line Since the scheme encodes bit in unary, it is vulnerable to a brute force attack such that an attacker can try all possible input in linear time. The scheme can be adjusted to remedy such problem as follow.

In the commitment procedure

1. The first line l_1 is placed at the position x_i, it covers the positions from x_i to $x_i + d$ -1. For any two lines l_i and l_j, $| x_i - x_j |$ must be greater than $2d$ to ensure that no two lines are placed closer than d positions from each other. The positions of all placements must be kept secret.
2. For a line l_i, select a codeword c_i from an error correcting code of length n that can correct up to d errors.
3. The unary numerical representations of the value x_i and $x_i + d$ are exclusive-ored with c_i. The hash of c_i is also computed.
4. Create a new set L. Put each concealment $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ in L.

In the matching procedure,

1. k lines, $o_1, .., o_n$, each of length d, are placed on the platform.
2. We denote the position of the line o_i as s_i. Similarly, the placement must ensure that no two lines are placed closer than d positions from each other.
3. Create a new set O. Put the end positions of each line $(s_i, s_i + d)$ in O.
4. For each element $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ in L and $(s_j, s_j + d)$ in O, perform the following steps.
   a. Compute $u_1 = s_j \oplus x_i \oplus c_i$
   b. Compute $u_2 = s_j \oplus (x_i + d) \oplus c_i$
   c. Let v_1 and v_2 be the corrected u_1 and u_2, according to the error correction scheme, if $h(v_1) = h(v_2) = h(c_i)$, then the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ from L and go to step 5. Otherwise proceed to the next step.
   d. Perform the operations as in step c, but replace s_j with $s_j + d$. If the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ from L.
5. Remove $(s_i, s_i + d)$ from O. Repeat step 4 until O is empty.
6. If e or fewer elements remain in L, the matching succeeds. Otherwise the matching fails.

The platform of the two-dimensional scheme consists of a rectangle of the size mxn, placements of k squares of the size dxd, and the maximal match error e. The commitment is done by placing k squares of the size dxd on it. After that, a user, without seeing the location of the placements of the first k squares, can place a challenge by putting other k squares of the size dxd on the platform. If more than k-e pairs of squares overlap, the match is successful. An example of the platform of the scheme is depicted in figure 2. The

details of scheme are described in the following procedures, the commitment and the matching.

In the commitment procedure,

1. The first square l_1 is placed at the position (x_1, y_1), the second square l_2 is placed at the position (x_2, y_2), and so on until the last square is placed at the position (x_k, y_k). Since the square l_i is placed at the position (x_i, y_i), it covers the area bounded by (x_i, y_i), $(x_i, y_i + d)$, $(x_i + d, y_i)$, $(x_i + d, y_i + d)$. For any two squares l_i and l_j, $| x_i - x_j |$ as well as $| y_i - y_j |$ must be greater than $2d$ to ensure that no two squares are placed closer than d positions from each other. The positions of all placements must be kept secret.
2. For a square l_i, select a codeword c_i from an error correcting code of length n that can correct up to d errors.
3. The unary numerical representations of the value x_i, $x_i + d$, y_i, $y_i + d$, are exclusive-ored with c_i. The hash of c_i is also computed.
4. Create a new set L. Put each concealment $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i, h(c_i))$ in L.

Similar to the one-dimensional fuzzy matching scheme, in the matching procedure of the two-dimensional fuzzy matching procedure,

1. k squares, $o_1, .., o_n$, each of the size dxd, are placed on the platform.
2. We denote the position of the line o_i as (s_i, t_i). Similarly, the placement must ensure that no two squares are placed closer than d positions from each other.
3. Create a new set O. Put the corner positions of each square $(s_i, s_i + d, t_i, t_i + d)$ in O.
4. For each element $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i, h(c_i))$ in L and $(s_i, s_i + d, t_i, t_i + d)$ in O, perform the following steps.
   a. Compute $u_1 = s_j \oplus x_i \oplus c_i$
   b. Compute $u_2 = s_j \oplus (x_i + d) \oplus c_i$
   c. Compute $u_3 = t_j \oplus y_i \oplus c_i$
   d. Compute $u_4 = t_j \oplus (y_i + d) \oplus c_i$
   e. Let v_1, v_2, v_3, and v_4 be the corrected u_1, u_2, u_3, and u_4, according to the error correction scheme, if $h(v_1) = h(v_2) = h(v_3) = h(v_4) = h(c_i)$, then the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, y_i \oplus c_i, (y_i + d) \oplus c_i, h(c_i))$ from L and go to step 5. Otherwise proceed to the next step.
   f. Perform the operations as in step e, but replace s_j with $s_j + d$. If the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ from L. Otherwise proceed to the next step.
   g. Perform the operations as in step e, but replace t_j with $t_j + d$. If the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ from L. Otherwise proceed to the next step.
   h. Perform the operations as in step e, but replace s_j with $s_j + d$ and t_j with $t_j + d$. If the match is successful, remove $(x_i \oplus c_i, (x_i + d) \oplus c_i, h(c_i))$ from L. Otherwise proceed to the next step.
5. Remove $(s_i, s_i + d, t_i, t_i + d)$ from O. Repeat step 4 until O is empty.

If e or fewer elements remain in L, the matching succeeds. Otherwise the matching fails.

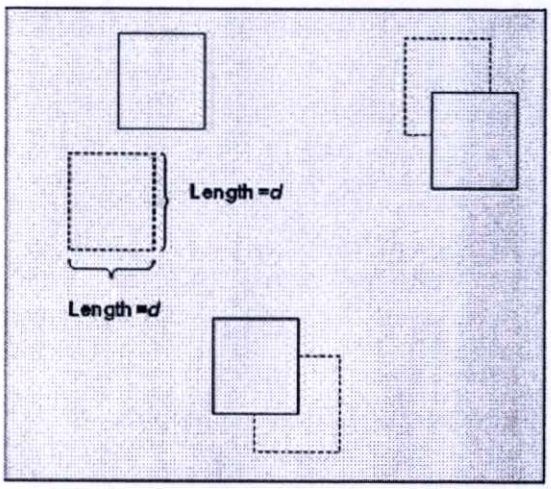


Figure 2. The platform of the two-dimensional fuzzy matching scheme.

The scheme can be adjusted for the use of any number of attributes. Each attribute can be subject to different error possibility. This can be achieved by using different codeword with different error correcting capability for each attribute. Therefore the matching scheme for three dimensional objects without the restriction that the width, the length, and the height must be the same, i.e. it needs not be a cube, can be constructed.

The scheme can be enhanced furthermore by adding some coefficients or weights to the matching. For example, in a two dimensional matching scheme, each rectangle is given a coefficient or a weight. A coefficient can be negative indicating the undesirable match. Moreover, it can be set to the value of negative infinity to rule out the object from being successfully matched at all when it happens to match some extremely undesirable object. The higher score the matching, the more likely it will succeed. A threshold value can be set to determine the successful ones. This increases the flexibility of the scheme and enables it to be adapted for more extensive uses. Figures 3 and 4 show a two-dimensional matching scheme with coefficients. There are four rectangles in the scheme with the coefficients 5, 3, 1 and -1. The matching in figure 3 scores less than the one in figure 4 as it matches one rectangle with a positive coefficient and another rectangle with a negative coefficient

while the one in figure 5 matches two rectangles with positive coefficients and the combined matched coefficient is greater than the one in figure 4.

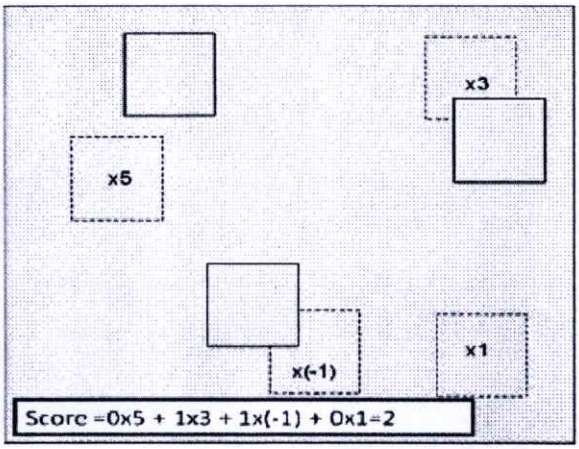


Figure 3. The platform of the two dimensional fuzzy matching scheme with score = 2.

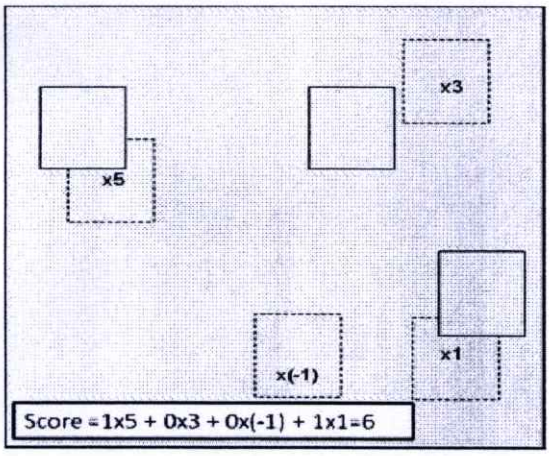


Figure 4. The platform of the two dimensional fuzzy matching scheme with score = 6.

Since the scheme encodes bit in unary, it is vulnerable to a brute force attack such that an attacker can try all possible input in linear time. The scheme can be adjusted to remedy such problem as follow.

In the commitment procedure

1. Generate n pair wise prime number $a_1, a_2, \ldots, a_n$ such that $a_1, a_2, \ldots, a_n >$ the secret s.
2. Calculate $s_1 = s \bmod a_1$, $s_2 = s \bmod a_2, \ldots, s_n = s \bmod a_n$.
3. Randomly generate n error correcting code $x_1, x_2, \ldots, x_n$.
4. Calculate $c_1 = s_1 \oplus x_1$, $c_2 = s_2 \oplus x_2, \ldots, c_n = s_n \oplus x_n$, all in unary.
5. Calculate $h(s)$ in binary.

In matching process

1. User input data for verification o
2. Calculate $o_1 = o \bmod a_1$, $o_2 = o \bmod a_2, \ldots, o_n = o \bmod a_n$.
3. Calculate $u_1 = c_1 \oplus o_1$, $u_2 = c_2 \oplus o_2, \ldots, u_n = c_n \oplus o_n$, all in unary.
4. Correct $(u_1, u_2, \ldots, u_n)$.
5. Use Chinese reminder theorem to find u.
6. Match $h(u)$ with $h(s)$.

IV.CONCLUSIONS

This paper introduces a fuzzy matching scheme that can match objects according to their attributes. The scheme applies the fuzzy commitment technique with the attribute of object in unary numerical format. It can be adjusted by putting some weight to each matching. The weight can be negative to indicate undesirable match. It can be enhanced for practical use in the template protection of secure biometric systems such as fingerprint, iris and face recognition.

REFERENCES

[1] Bruce Schneier. 1996. Applied Cryptography, Second Edition: Protocols, Algorithms and Source Code. NEWYORK : John Wiley & Sons, Inc.

[2] U.Uludang S. Pankanti, S.Prabhakar and A. K.Jain. "Biometric Cryptosystems: Issues and Challenges" *in PROCEEDINGS OF THE IEEE*, vol. 92, no. 6, June 2004. pp.948-960.

[3] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," *in Proceedings of 6th ACM Conference on Computer and Communications Security (ACM CCS '99)*, pp. 28–36,Singapore,November 1999.

[4] A. Juels and M. Sudan, "A fuzzy vault scheme," *in Proceedings of the IEEE International Symposium on Information Theory*, p.408, Piscataway, NJ, USA, June-July 2002.

[5] A. K. Jain, K. Nandakumar, and A. Nagar, "Review Article Biometric Template Security" *in EURASIP Journal on Advances in Signal Processing*, Volume 2008, Michigan, USA, December 2007

[6] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *in Journal of Computer and System Sciences*, vol. 18,no. 2, pp. 143–154, 1979.

[7] V. Guruswami and M. Sudan. "Improved decoding of reed-solomon and algebraic-geometric codes." *In FOCS '98*, pages 28{39. IEEE Computer Society, 1998.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายอรรคพล เสถียนจารุรัตน์
วัน/เดือน/ปีเกิด	1 ตุลาคม พ.ศ.2523 ที่จังหวัดนครราชสีมา
ที่อยู่ปัจจุบัน	241/1 ถนน ผดุงพานิช ตำบล ในเมืองร้อยเอ็ด อำเภอ เมืองร้อยเอ็ด จังหวัดร้อยเอ็ด 45000
ประวัติการศึกษา	2546 วิศวกรรมศาสตร์บัณฑิต สาขาวิชา วิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
	2548 บริหารธุรกิจมหาบัณฑิต สาขาวิชา บริการการตลาด มหาวิทยาลัยราชภัฏสวนดุสิต
ความชำนาญเฉพาะด้าน	1.) ระบบเครือข่ายคอมพิวเตอร์
	2.) ระบบความปลอดภัยข้อมูลคอมพิวเตอร์
ประสบการณ์ทำงาน	
พ.ศ.2546-2548	ตำแหน่งผู้ดูแลระบบ บริษัทโคราชโพทรีย์ จำกัด
พ.ศ.2548-2549	ตำแหนงวิศวกรระบบ บริษัท เน็ตแบรนคอนเซาลติ้ง จำกัด
พ.ศ.2549-ปัจจุบัน	ตำแหน่งผู้ช่วยหัวหน้าทีม บริษัทโปรเกรสซอฟท์แวร์ จำกัด