ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่คงทนต่อการล่มของเอเจนต์

DECENTRALIZED MULTIPLE-AGENT KEY RECOVERY SYSTEM RESILIENT TO AGENT FAILURES

กนกวรรณ กันยะมี

KANOKWAN KANYAMEE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-IT-D-001-001

ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่คงทนต่อการล่มของเอเจนต์

DECENTRALIZED MULTIPLE-AGENT KEY RECOVERY SYSTEM RESILIENT TO AGENT FAILURES

กนกวรรณ กันยะมี

KANOKWAN KANYAMEE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-IT-D-001-001

DECENTRALIZED MULTIPLE-AGENT KEY RECOVERY SYSTEM RESILIENT TO AGENT FAILURES

KANOKWAN KANYAMEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2013

KMITL-2013-IT-D-001-001

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่คงทนต่อการล่มของเอเจนต์

Decentralized Multiple-Agent Key Recovery System Resilient to Agent Failures

นักศึกษา นางสาวกนกวรรณ กันยะมี

รหัสประจำตัว 48066302

ปริญญา ปรัชญาดุษฎีบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ ดร.นพพร โชติกกำธร ผู้ช่วยศาสตราจารย์ ดร.ทศพล สอตระกูล รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ ดร.สุเมธ ประภาวัต ดร.นล เปรมัษเฐียร	

วัน / เดือน/ ปี ที่สอบ วันจันทร์ที่ 20 พฤษภาคม 2556 เวลา 10.00 น.

สถานที่สอบ ณ ห้อง 334 (ชั้น 3) คณะเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศรับรองแล้ว

(รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์)

คณบดีคณะเทคโนโลยีสารสนเทศ

วันที่ 30 **เดือน** พฤษภาคม **พ.ศ.** 2556

หัวข้อวิทยานิพนธ์	ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่คงทนต่อการล่มของเอเจนต์
นักศึกษา	นางสาวกนกวรรณ กันยะมี
รหัสนักศึกษา	48066302
ปริญญา	ปรัชญาดุษฎีบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2556
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ. ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

บทคัดย่อ

เทคโนโลยีการเข้ารหัสลับจะช่วยเพิ่มความมั่นคงปลอดภัย และเพิ่มความเป็นส่วนตัวให้กับผู้ที่ใช้งานระบบเครือข่าย การเข้ารหัสลับแบบสมมาตรจะใช้กุญแจลับในการเข้ารหัสและถอดรหัสข้อมูล ในกรณีที่ผู้รับไม่สามารถใช้กุญแจในการถอดรหัสข้อมูลได้ หรือภาครัฐต้องการใช้สิทธิ์ในการเข้าถึงข้อมูลจะต้องขอใช้บริการกู้คืนกุญแจ วิทยานิพนธ์นี้นำเสนอระบบการกู้คืนกุญแจแบบหลายเอเจนต์ที่ทำงานโดยไม่ใช้ศูนย์กลางการกู้คืนกุญแจจำนวนสองระบบ คือ ระบบเอชเอดีเอ็ม-เคอาร์เอส และระบบเอสเอชเอดีเอ็ม-เคอาร์เอส ซึ่งเป็นระบบที่กุญแจลับมีความมั่นคงปลอดภัยสูง และมีความยืดหยุ่นในการจัดการจำนวนเอเจนต์ขั้นต่ำที่ใช้สำหรับการกู้คืนกุญแจ เพื่อให้สอดคล้องกับระดับของความมั่นคงปลอดภัยที่ต้องการได้ สามารถกู้คืนกุญแจลับได้แม้ในกรณีที่มีบางเอเจนต์ในกลุ่มการกู้คืนกุญแจล่ม โดยนำแนวคิดพื้นฐานทฤษฎีเรื่องการแชร์ความลับและเพาเวอร์เซตมาใช้ในการแบ่งและจัดสรรส่วนประกอบของกุญแจลับ ทำให้ระบบมีความพร้อมใช้งานและมีความน่าเชื่อถือสูง ตลอดจนผู้ใช้งานมีความเป็นส่วนตัวสูง ระบบสามารถพิสูจน์ตัวจริงของเอเจนต์ที่อยู่ในกลุ่มการกู้คืนกุญแจเดียวกัน ทั้งนี้ระบบยังให้การสนับสนุนการตรวจสอบข้อมูลโดยชอบด้วยกฎหมายและทำงานบนโครงสร้างพื้นฐานกุญแจสาธารณะหรือพีเคไอ

Thesis	Decentralized Multiple-Agent Key Recovery System Resilient to Agent Failures
Student	Miss Kanokwan Kanyamee
Student ID.	48066302
Degree	Doctor of Philosophy
Program	Information Technology
Year	2013
Thesis Advisor	Assoc. Prof. Dr. Chanboon Sathitwiriyawong

ABSTRACT

Cryptography technology will help to strengthen security and privacy of various network activities. Symmetric cryptography uses the same session key for message encryption and decryption. In case the session key is unavailable or legal investigation of transmitting messages is needed, an appropriate recovery mechanism is required. Thus, this research presents two novel decentralized multiple agent key recovery systems: High-Availability Decentralized Multiple-Agent Key Recovery System (HADM-KRS) and Simple High-Availability Decentralized Multiple-Agent Key Recovery System (SHADM-KRS) that have high secrecy of session key and high flexibility to manage the minimum number of key recovery agents for successful key recovery according to security policies and requirements. It can recover session key despite the failure of some key recovery agents. This feature is achieved by applying the basic concept of secret sharing and power set to distribute the session key to participating key recovery agents. It has high availability, high reliability, high privacy, and ability to detect attacks on group authentication. Finally, it also supports law enforcement and is based on security mechanism using well defined features of Public Key Infrastructure.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์ โดยได้รับความกรุณาเป็นอย่างดียิ่งจาก รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งและขอขอบพระคุณเป็นอย่างสูง ในความเมตตาของท่านที่ได้ให้คำแนะนำพร้อมทั้งข้อเสนอแนะที่เป็นประโยชน์ในการทำวิจัย จนกระทั่งงานวิจัยสำเร็จ

ขอขอบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้กับข้าพเจ้า

ขอขอบพระคุณมหาวิทยาลัยราชภัฏอุตรดิตถ์ ที่ได้ให้การอนุเคราะห์สนับสนุนทุนการศึกษาตลอดระยะเวลาของการศึกษาและการทำวิจัย

ขอขอบคุณผู้บริหาร เพื่อนร่วมงาน และเจ้าหน้าที่ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏอุตรดิตถ์ ที่ได้ให้กำลังใจ ให้โอกาสและเอื้อเฟื้อเวลาสำหรับการทำวิจัย

ขอขอบคุณพี่ ๆ เพื่อน ๆ น้อง ๆ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกคน ที่ได้ให้คำแนะนำต่าง ๆ และคอยให้กำลังใจเสมอมา

ขอขอบคุณครอบครัวที่เป็นกำลังใจที่ดีและอบอุ่น พร้อมทั้งสนับสนุนค่าใช้จ่ายตลอดเวลาที่ศึกษา จนส่งผลให้การทำวิจัยในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

กนกวรรณ กันยะมี

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันการติดต่อสื่อสารกันบนระบบเครือข่ายเกิดขึ้นอย่างกว้างขวางทั่วโลก สังเกตได้จากอัตราการเติบโตของผู้ใช้งานเครือข่ายที่เพิ่มขึ้นอย่างต่อเนื่อง โดยมีการนำมาประยุกต์ใช้กับการทำกิจกรรมที่หลากหลายในรูปแบบอิเล็กทรอนิกส์ อาทิเช่น E-Government, E-Commerce, E-Market, E-Banking, E-Education เป็นต้น เพื่อให้เกิดความแข็งแรงทางด้านความมั่นคงปลอดภัย (Security) และความเป็นส่วนตัว (Privacy) ในการใช้งานระบบเครือข่ายจึงมีการนำเทคโนโลยีวิทยาการเข้ารหัสลับข้อมูล (Cryptography Technology) มาใช้เพื่อรักษาความลับของข้อมูล ซึ่งข้อมูลจะถูกเปิดอ่านได้เฉพาะบุคคลที่ได้รับอนุญาตเท่านั้น

กระบวนการรักษาความลับของข้อมูลนั้น ข้อมูลจะถูกเข้ารหัสลับ (Encryption) โดยผู้ส่งซึ่งจะมีการแปลงข้อความต้นฉบับ (Plaintext) ให้เป็นข้อความอีกรูปแบบหนึ่งที่ไม่สามารถอ่านเข้าใจได้ (Ciphertext) แล้วส่งข้อความดังกล่าวไปยังคู่สื่อสารคือผู้รับ ทางฝั่งผู้รับจะใช้กระบวนการถอดรหัสลับ (Decryption) ข้อความที่ไม่สามารถอ่านได้ หรือข้อความที่ได้รับจากผู้ส่งข้างต้น ให้กลับเป็นข้อความต้นฉบับ การใช้เทคโนโลยีดังกล่าวมีเพียงผู้รับและผู้ส่งเท่านั้น ที่จะสามารถอ่านข้อความตั้งต้น และสื่อสารกันได้เข้าใจ

การเข้ารหัสลับข้อมูลกระทำการได้โดยอาศัยวิธีการใช้กุญแจ (Key) ในการเข้ารหัสลับข้อมูลและถอดรหัสลับข้อมูล วิธีการในการเข้ารหัสลับข้อมูลแบ่งออกเป็น 2 ประเภท [1] คือ (1) แบบสมมาตร (Symmetric Key Algorithms) (2) แบบอสมมาตร (Asymmetric Key Algorithms) ในการเข้ารหัสลับแบบสมมาตร จะใช้กุญแจเพียงตัวเดียวในการเข้ารหัสลับและถอดรหัสลับข้อมูล เรียกกุญแจนี้ว่า กุญแจลับ (Secret Key) หรือกุญแจเซสชั่น (Session Key) ส่วนแบบอสมมาตร จะใช้กุญแจสองตัวในการเข้ารหัสลับและถอดรหัสลับข้อมูล กุญแจนั้นคือกุญแจสาธารณะ (Public Key) และกุญแจส่วนตัว (Private Key) โดยผู้ส่งจะใช้กุญแจสาธารณะของผู้รับในการเข้ารหัสลับข้อมูล และผู้รับจะใช้กุญแจส่วนตัวของตนเองในการถอดรหัสลับข้อมูล

อย่างไรก็ตามการนำเทคโนโลยีเข้ารหัสลับมาใช้เพื่อการรักษาความลับของข้อมูลนั้น อาจมีการนำไปใช้ปกปิดหรือซ่อนเร้นข้อมูลที่เป็นภัยอันตรายต่อสังคมหรือบุคคลอื่น ด้วยเหตุดังกล่าวจึงเป็นจุดเริ่มต้นของการออกกฎหมายเพื่อตรวจสอบข้อมูลที่ต้องสงสัยบนระบบเครือข่ายและคุ้มครองผู้ใช้งานระบบเครือข่าย ด้วยการให้อำนาจสิทธิ์แก่ภาครัฐในการตรวจสอบ

เริ่มจากในปี 1993 ระบบฝากกุญแจ (Key Escrow System: KES) [2] ถูกพัฒนาขึ้นมาเพื่อสนับสนุนการทำงานของรัฐบาลในการเข้าถึงกุญแจลับเพื่อนำมาใช้ในการถอดรหัสข้อมูลที่มีการ

ส่งผ่านระบบเครือข่ายอย่างถูกต้องตามกฎหมาย ส่งเสริมความปลอดภัยของประชาชน และ ความมั่นคงของชาติ โดย KES เป็นระบบที่จัดเก็บสำเนากุญแจของผู้ใช้งานการเข้ารหัสข้อมูล โดยผู้ที่สามารถเข้าถึงกุญแจดังกล่าวได้คือหน่วยงานรัฐบาล ซึ่งจะช่วยให้รัฐบาลสามารถตรวจสอบข้อมูลที่ต้องสงสัย และสกัดกั้นการกระทำอันเป็นการบ่อนทำลายความมั่นคงทางสังคมได้ ทั้งนี้มีผลทำให้ความเป็นส่วนตัวของผู้ใช้งานลดน้อยลง

ต่อมาในปี 1994 รัฐบาลสหรัฐอเมริกาได้ประกาศมาตรฐานของระบบฝากกุญแจ เรียกว่า Escrow Encryption Standard (EES) [3,4] โดยใช้อัลกอริทึม SKIPJACK และใช้ฟิลด์ลีฟ (Law Enforcement Access Field : LEAF) สำหรับการเข้าถึงข้อมูลกุญแจ เพื่อใช้ในการถอดรหัสข้อมูลที่ต้องสงสัยได้อย่างถูกต้องตามกฎหมาย

ในปี 1996 รัฐบาลสหรัฐอเมริกาเกิดปัญหาที่เกี่ยวข้องกับ KES เนื่องจากระบบดังกล่าวเอื้อประโยชน์เฉพาะการเข้าถึงกุญแจเพื่อถอดรหัสข้อมูลโดยรัฐบาลเท่านั้น แต่ไม่ได้เอื้อประโยชน์ต่อผู้ใช้งานในกรณีกุญแจสูญหาย จึงเกิดการนำเสนอวิธีการกู้คืนกุญแจ (Commercial Key Recovery Schemes) [5] ซึ่งวิธีการนี้สามารถกู้คืนกุญแจที่หายได้ ในปีต่อมาก็ได้มีงานวิจัยที่นำเสนอระบบการกู้คืนกุญแจ (Cryptography Key Recovery System) [6] โดยไม่จำเป็นต้องอาศัยการฝากกุญแจ แต่จะอาศัย Key Recovery Entry (KRE) ในการจัดเก็บกุญแจ และใช้สำหรับการกู้คืนกุญแจเมื่อต้องการ คือ กุญแจสูญหายหรือกุญแจไม่สามารถใช้งานได้ งานวิจัยดังกล่าวได้แสดงให้เห็นถึงความแตกต่างของระบบฝากกุญแจกับระบบการกู้คืนกุญแจ

ต่อมาในปี 1998 สถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (National Institute of Standards and Technology: NIST) ได้ประกาศมาตรฐานของระบบการกู้คืนกุญแจ (Key Recovery System: KRS) [7] โดยได้ระบุข้อกำหนดที่สำคัญสำหรับการกู้คืนกุญแจที่จะนำมาใช้โดยหน่วยงานของรัฐบาลกลาง ระบบการกู้คืนกุญแจจะใช้สำหรับการกู้คืนกุญแจเมื่อหน่วยงานของรัฐบาลต้องการตรวจสอบข้อความที่ต้องสงสัย และการกู้คืนกุญแจของผู้ใช้งานที่สูญหายหรือไม่สามารถใช้งานได้

หลังจากนั้นการวิจัยทางด้านระบบการกู้คืนกุญแจก็มีการพัฒนาอย่างต่อเนื่องจนถึงปัจจุบัน โดยได้มีการพิจารณาเพิ่มเติมในประเด็นทางด้าน การบริหารจัดการกุญแจ [8] การพิสูจน์ตัวตน [9] การเข้าถึงข้อมูลโดยมีการคำนึงถึงเรื่องความเป็นส่วนตัว [10] รูปแบบการให้ความไว้วางใจเอเจนต์ในการกู้คืนกุญแจ [11, 12] การปรับปรุงเรื่องความมั่นคงปลอดภัยของระบบ [13,14] การกู้คืนกุญแจสำหรับการเข้ารหัสข้อมูล [15] การกู้คืนกุญแจสำหรับการเข้ารหัสเอกสาร [16] การทดลองการโจมตีการกู้คืนกุญแจบนโพรโทคอลเอชดีซีพี (High-Bandwidth Digital Content Protection Protocol: HDCP) [17] การกู้คืนกุญแจบน IPSec [18] และยังมีการจดสิทธิบัตรงานที่เกี่ยวข้องกับระบบการกู้คืนกุญแจอีกมากมาย ในปัจจุบัน NIST ได้ประกาศความสำคัญของระบบการกู้คืน

กุญแจไว้ในเอกสารกรอบการออกแบบระบบบริหารจัดการกุญแจเข้ารหัสลับ (A Framework for Designing Cryptographic Key Management System) [19]

ระบบการกู้คืนกุญแจ (Key Recovery System: KRS) มีสองรูปแบบ คือ ระบบการกู้คืนกุญแจแบบใช้เอเจนต์เดี่ยว (Single Key Recovery System: S-KRS) [2, 6, 10, 20] และระบบการกู้คืนกุญแจแบบใช้หลายเอเจนต์ (Multiple Key Recovery System: M-KRS) [13, 14, 21] โดยกระบวนการทำงานของ S-KRS จะใช้เอเจนต์เดี่ยว (Single Key Recovery Agent: S-KRA) ในการกู้คืนกุญแจ ส่วน M-KRS จะใช้หลายเอเจนต์ (Multiple Key Recovery Agent: M-KRA) ร่วมกันกู้คืนกุญแจ

การพัฒนาระบบในช่วงเริ่มแรกจะเป็นรูปแบบ S-KRS มีกระบวนการทำงานที่ไม่ซับซ้อน จึงทำให้เกิดภัยคุกคามต่อระบบได้ง่าย ระบบมีความมั่นคงปลอดภัยน้อยเมื่อเทียบกับภัยคุกคามที่มีหลากหลายรูปแบบ และมีความรุนแรงมากขึ้น [22] เช่น การปฏิเสธการให้บริการ (Denial of Service) [23, 24] เป็นต้น ต่อมาการพัฒนาระบบในช่วงหลัง จึงได้ออกแบบระบบโดยใช้รูปแบบ M-KRS ทั้งนี้เพื่อลดความเสี่ยงต่อความเสียหายที่เกิดขึ้นกับ S-KRS

กระบวนการทำงานของ M-KRS สามารถดำเนินการได้สองรูปแบบคือ (1) แบบอาศัยศูนย์กลางในการกู้คืนกุญแจ (Key Recovery Center: KRC) [13, 14, 25, 26] ในการทำหน้าที่ติดต่อประสานงานระหว่างเอเจนต์กู้คืนกุญแจ (Key Recovery Agent: KRA) ที่ทำหน้าที่ร่วมกันในการกู้คืนส่วนประกอบของกุญแจ (2) แบบไม่อาศัย KRC [21, 27] แต่จะอาศัยความร่วมมือกันระหว่าง KRA กับผู้ร้องขอการกู้คืนกุญแจ

เมื่อพิจารณาเปรียบเทียบข้อดีและข้อเสียของ M-KRS ทั้งสองรูปแบบแล้ว พบว่า M-KRS รูปแบบแรกมีข้อดีคือ ระบบทำงานไม่ซับซ้อน ข้อเสีย คือเมื่อ KRC เกิดข้อขัดข้องเสียหายจะส่งผลให้ระบบล้มเหลวไม่สามารถให้บริการได้ (Single Point of Failure: SPOF) และจำเป็นต้องใช้งบประมาณในการบริหารจัดการ KRC ส่วน M-KRS รูปแบบที่สองนั้นมีข้อดี คือระบบสามารถลดความเสี่ยงของการเกิดปัญหา SPOF และไม่มีต้นทุนในการบริหารจัดการ KRC แต่มีข้อเสีย คือการทำงานของระบบจะมีความซับซ้อนเพิ่มขึ้นเล็กน้อย

จากการศึกษางานวิจัยที่เกี่ยวข้องกับการพัฒนากระบวนการทำงานของระบบ M-KRS ที่ผ่านมายังปรากฏจุดอ่อนอยู่หลายด้าน ดังนี้

(1) Availability: ด้านความพร้อมใช้งานของระบบ กล่าวคือระบบไม่สามารถให้บริการกู้คืนกุญแจได้ ในกรณี KRC หรือบาง KRA ล่มไม่สามารถให้บริการได้ หรือเกิดปัญหา SPOF ที่ KRC

(2) Secrecy: ด้านความปลอดภัยของกุญแจลับ คือกุญแจลับไม่เป็นความลับเฉพาะผู้ส่งกับผู้รับเท่านั้น เพราะมีกระบวนการที่ให้ KRC หรือ KRA ทำหน้าที่กู้คืนกุญแจลับได้

(3) Flexibility: ด้านความยืดหยุ่นในการกำหนดจำนวน KRA ขั้นต่ำที่สามารถกู้คืนกุญแจลับได้สำเร็จ คือต้องให้ KRA ในกลุ่มการกู้คืนกุญแจทำการกู้คืนกุญแจครบทุกเอเจนต์ นั่นคือ หากมีบาง KRA ล่ม ระบบจะไม่สามารถกู้คืนกุญแจได้สำเร็จ

(4) Detection: ระบบยังขาดฟังก์ชันการตรวจสอบ KRA แปลกปลอม หรือ KRA ที่ไม่ได้อยู่ในกลุ่มการกู้คืนกุญแจ

ดังนั้นงานวิจัยนี้จะได้ศึกษาและวิจัยพัฒนากระบวนการทำงานของระบบ M-KRA ที่สามารถแก้ไขจุดอ่อนดังที่กล่าวมาแล้วข้างต้นทุกกรณี คือ

(1) ระบบสามารถให้บริการกู้คืนกุญแจได้โดยไม่มี KRC และสามารถให้บริการกู้คืนกุญแจได้เมื่อมีบาง KRA ล่ม

(2) กุญแจลับมีความปลอดภัย เนื่องจากความลับของกุญแจลับจะรู้เฉพาะผู้ส่งกับผู้รับเท่านั้น จึงเพิ่มความเป็นส่วนตัวให้กับผู้ใช้งานระบบ

(3) ระบบสามารถกำหนดจำนวน KRA ขั้นต่ำที่ใช้ในการกู้คืนกุญแจลับ

(4) ระบบมีฟังก์ชันการตรวจสอบ KRA แปลกปลอม หรือ KRA ที่ไม่ได้อยู่ในกลุ่มการกู้คืนกุญแจ

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อศึกษา วิจัย และพัฒนากระบวนการกู้คืนกุญแจลับแบบ M-KRS ที่มีความสามารถทางด้าน (1) ความพร้อมใช้งานหรือความสามารถในการให้บริการกู้คืนกุญแจได้แม้ในกรณีมีบาง KRA ล่ม (2) ความมั่นคงปลอดภัยของกุญแจลับและความเป็นส่วนตัวของผู้ใช้งาน (3) ความยืดหยุ่นในการกำหนดจำนวน KRA ขั้นต่ำที่ใช้ในการกู้คืนกุญแจ ทั้งนี้อาจขึ้นอยู่กับนโยบายและสภาพแวดล้อมของการนำระบบไปใช้งาน และ (4) ความสามารถในการตรวจสอบ KRA แปลกปลอม

1.3 สมมติฐานของการศึกษา

ระบบการกู้คืนกุญแจแบบ M-KRS จะสามารถกู้คืนกุญแจลับได้ในสองกรณี คือ กรณีที่กุญแจลับของผู้รับสูญหาย และกรณีที่รัฐบาลต้องการกู้คืนกุญแจลับเพื่อนำไปถอดรหัสลับข้อความที่ต้องสงสัย โดยระบบต้องมีความพร้อมใช้งานสูงและสามารถทำงานได้แม้ในกรณีที่มีบาง KRA ในกลุ่มการกู้คืนล่ม

1.4 แนวคิดที่ใช้ในการวิจัย

การออกแบบระบบการกู้คืนกุญแจแบบ M-KRS ต้องทำให้ระบบมีความพร้อมใช้งานสูง คือ ระบบสามารถกู้คืนกุญแจได้เมื่อมีบาง KRA ล่ม ในการกู้คืนกุญแจจะต้องอาศัยข้อมูลจากฟิลด์ที่ใช้ในการกู้คืนกุญแจ (Key Recovery Field : KRF) โดยจะถูกนำมาผ่านขั้นตอนและกระบวนการต่าง ๆ เช่น การคำนวณทางคณิตศาสตร์ กระบวนการพิสูจน์ตัวจริง การถอดรหัสลับข้อมูล เป็นต้น เพื่อให้ได้มาซึ่งกุญแจที่สามารถนำไปถอดรหัสลับได้ ดังนั้นจะต้องออกแบบกระบวนการกู้คืนกุญแจ และโครงสร้างของ KRF ให้มีความเหมาะสม ยืดหยุ่น โดยคำนึงถึงความลับของกุญแจลับและความเป็นส่วนตัวของผู้ใช้งาน และจะต้องสอดคล้องกับวิธีการในการกู้คืนกุญแจ

โครงสร้างของ KRF ถูกออกแบบให้จัดเก็บข้อมูลสำหรับการกู้คืนส่วนประกอบของกุญแจลับ โดยนำแนวคิดพื้นฐานเรื่องการแชร์ความลับ (Secret Sharing) [28] มาใช้ในการแบ่งและจัดสรรส่วนประกอบของกุญแจ และใช้ทฤษฎีพื้นฐานเรื่องเพาเวอร์เซต (Power Set) [29] มาใช้ในกระบวนการสำรองส่วนประกอบของกุญแจสำหรับการกู้คืนกุญแจในกรณีที่มีบาง KRA ล่ม และทำให้สามารถกำหนดระดับความมั่นคงปลอดภัยโดยการกำหนดจำนวน KRA ขั้นต่ำ

การทดสอบการทำงานของระบบการกู้คืนกุญแจแบบ M-KRS จะพิจารณาถึงเวลาที่ใช้ในกระบวนการกู้คืนกุญแจ คือ เวลาที่ใช้ในการสร้าง KRF ซึ่งจะเกิดขึ้นทุกครั้งที่มีการส่งข้อมูล เวลาที่ใช้ในการกู้คืนกุญแจลับ และเวลาที่ใช้ในการกู้คืนกุญแจลับเมื่อมีบาง KRA ล่ม

1.5 ขอบเขตการวิจัย

ในงานวิจัยนี้จะได้ศึกษาและพัฒนากระบวนการกู้คืนกุญแจแบบ M-KRS โดยทำการปรับปรุงกระบวนการทำงานและโครงสร้างของ KRF ซึ่งทั้งสองส่วนจะต้องมีความสอดคล้องและสัมพันธ์กัน สามารถกู้คืนกุญแจลับได้อย่างถูกต้อง มั่นคงปลอดภัย ภายในระยะเวลาที่รวดเร็ว และส่งเสริมเรื่องความเป็นส่วนตัวของผู้ใช้งาน ระบบสามารถให้บริการกู้คืนกุญแจได้แม้เกิดเหตุการณ์บาง KRA ในกลุ่มการกู้คืนล่มหรือไม่สามารถให้บริการได้ พร้อมทั้งออกแบบกระบวนการกู้คืนให้มีความยืดหยุ่นโดยระบบสามารถกำหนดจำนวน KRA ขั้นต่ำสำหรับการกู้คืนกุญแจได้ รวมทั้งให้มีฟังก์ชันสำหรับตรวจสอบ KRA แปลกปลอมได้ โดยกำหนดให้มีกระบวนการทำงานบนโครงสร้างพื้นฐานกุญแจสาธารณะ

1.6 ขั้นตอนของการศึกษา

งานวิจัยนี้มีรายละเอียดขั้นตอนการดำเนินการดังนี้

1.6.1 ศึกษาค้นคว้า เรื่องวิทยาการเข้ารหัสลับและเทคโนโลยีเกี่ยวกับการสร้างความมั่นคงปลอดภัยให้กับข้อมูล

1.6.2 ศึกษาค้นคว้าเทคนิควิธีการพื้นฐาน รวมทั้งองค์ประกอบของระบบการกู้คืนกุญแจในรูปแบบต่าง ๆ และความแตกต่างของแต่ละวิธีการ

1.6.3 จำลองกระบวนการกู้คืนกุญแจตามวิธีการที่ได้มีการนำเสนอไว้แล้ว เพื่อทำความเข้าใจ และทดสอบประสิทธิภาพของการกู้คืนกุญแจในแต่ละวิธีการ

1.6.4 วิเคราะห์และออกแบบกระบวนการกู้คืนกุญแจและโครงสร้างของ KRF โดยอาศัยเทคนิควิธีการพื้นฐานในการกู้คืนกุญแจ ตามที่ได้ศึกษามา

1.6.5 ทดลองและเก็บผลการจำลองการกู้คืนกุญแจลับตามกระบวนการที่ได้ออกแบบ

1.6.6 ประเมินผล สรุปผล และจัดทำเอกสาร

บทที่ 2
ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

ในบทนี้เป็นการอธิบายเกี่ยวกับทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับการวิจัยและพัฒนาระบบการกู้คืนกุญแจ แบ่งออกเป็นหัวข้อต่าง ๆ ได้แก่ นโยบายของรัฐบาลสหรัฐอเมริกาเกี่ยวกับระบบการกู้คืนกุญแจ ระบบการเข้ารหัสลับการกู้คืนกุญแจ องค์ประกอบของระบบการเข้ารหัสลับการกู้คืนกุญแจ วิธีการ/กระบวนการพื้นฐานสำหรับการกู้คืนกุญแจ รูปแบบการมอบความไว้วางใจของผู้ให้บริการออกใบรับรอง รูปแบบของบุคคลที่สามที่เกี่ยวข้องกับระบบการกู้คืนกุญแจ หน่วยงานที่ให้บริการกู้คืนกุญแจ การห่อหุ้มกุญแจ การแชร์ความลับ เพาเวอร์เซต ปัญหาความเสี่ยงและอุปสรรค ของระบบการกู้คืนกุญแจ และการสำรวจงานวิจัยที่เกี่ยวข้อง

2.1 นโยบายของรัฐบาลสหรัฐอเมริกาเกี่ยวกับระบบการกู้คืนกุญแจ

นโยบาย "การเข้าถึงข้อมูลอย่างถูกกฎหมายโดยหน่วยงานของรัฐบาล" (Lawful State Access) [30] ในบางสถานการณ์ เช่น เมื่อสงสัยว่าจะมีการก่อการร้าย หรือการละเมิดกฎหมายต่าง ๆ เช่น การฟอกเงิน การหลบเลี่ยงภาษี การฝ่าฝืนกฎหมายการแข่งขันอย่างเป็นธรรม เป็นต้น ตลอดจนนโยบายอื่นที่เกี่ยวกับการควบคุมการผลิต การใช้ประโยชน์ การส่งออก และการนำเข้าเทคโนโลยีการเข้ารหัสลับ สามารถเรียกรวม ๆ ว่า "นโยบายการเข้ารหัสลับ" (Encryption Policy)

นอกเหนือจากการควบคุมการส่งออก หรือการนำเข้าแล้ว วิธีการในการควบคุมเทคโนโลยีเข้ารหัสลับที่สำคัญอื่น ๆ คือ การจำกัดความยาวของกุญแจ (Key Length) [31] และการกำหนดให้มีระบบการเก็บหรือกู้คืนกุญแจ การจำกัดความยาวของกุญแจที่สามารถใช้ได้จะทำให้หน่วยงานของรัฐสามารถใช้เครื่องคอมพิวเตอร์ที่มีขีดความสามารถสูงในการถอดรหัสลับของข้อมูลที่ต้องการตรวจสอบได้ อย่างไรก็ตามการควบคุมโดยวิธีนี้ จะทำให้ระบบข้อมูลโดยรวมมีระดับความมั่นคงต่ำลง และเสี่ยงต่อการถูกผู้ประสงค์ร้ายลักลอบถอดรหัสลับด้วยวิธีการเดียวกัน ดังนั้นการอนุญาตให้ใช้เทคโนโลยีการเข้ารหัสลับที่มีความมั่นคงแข็งแรงสูง (Strong Encryption) แต่กำหนดให้หน่วยงานของรัฐที่เกี่ยวข้องสามารถเข้าถึงข้อมูลโดยชอบด้วยกฎหมาย โดยใช้ระบบการเก็บกุญแจหรือการกู้คืนกุญแจซึ่งเป็นวิธีการที่ดีกว่า

การเข้าถึงข้อมูลอย่างถูกกฎหมายโดยหน่วยงานของรัฐนั้น ในระยะแรกจะใช้ระบบเก็บกุญแจ (Key Archiving) หมายถึง การที่หน่วยงานของรัฐกำหนดให้ผู้ใช้เทคโนโลยีการเข้ารหัสลับข้อมูลต้องสำรองกุญแจลับ แล้วจัดเก็บไว้ที่ใดที่หนึ่ง ตัวอย่างที่รู้จักกันดีของระบบดังกล่าว คือ ระบบฝากกุญแจ (Key Escrow) [2] ซึ่งกำหนดให้เจ้าหน้าที่หรือบุคคลที่รัฐแต่งตั้ง (Authorized People) เช่น องค์กรออกใบรับรองกุญแจเป็นผู้เก็บกุญแจสำรองไว้เพื่อใช้ในการถอดรหัสลับ ใน

กรณีที่รัฐต้องการตรวจสอบ รัฐบาลสหรัฐอเมริกาพยายามนำระบบดังกล่าวมาใช้อย่างเต็มรูปแบบ แต่ไม่สามารถดำเนินการได้ เนื่องจากได้รับการต่อต้านอย่างรุนแรงจากองค์กรสิทธิมนุษยชนต่าง ๆ ในประเทศ ด้วยเหตุผลว่ามาตรการดังกล่าวเป็นการละเมิดสิทธิเสรีภาพในการแสดงออกและความเป็นส่วนตัวของประชาชน อีกทั้งระบบนี้เอื้อประโยชน์เฉพาะการดำเนินงานของภาครัฐเท่านั้น แต่ไม่สามารถเอื้อประโยชน์ให้กับผู้ใช้งานในกรณีที่พวกเขาทำกุญแจหายได้

ต่อมาจึงมีความพยายามจะแก้ปัญหาที่เกิดขึ้นข้างต้น รัฐบาลจึงได้ประกาศมาตรฐานของระบบการกู้คืนกุญแจ (Key Recovery System: KRS) [7] ซึ่งเป็นวิธีการที่ผู้ใช้เทคโนโลยีการเข้ารหัสลับไม่ต้องฝากกุญแจลับของตนไว้กับผู้อื่น แต่จะใช้กุญแจสาธารณะของหน่วยงานกู้คืนกุญแจ (Key Recovery Agent: KRA) เข้ารหัสลับกุญแจลับของตน แล้วผนวกเข้ากับข้อมูลที่ต้องการเข้ารหัสลับ เมื่อรัฐบาลต้องการตรวจสอบข้อมูลจะสามารถใช้กุญแจลับของ KRA ถอดรหัสกุญแจลับของผู้ใช้ และนำกุญแจลับนั้นมาถอดรหัสข้อมูลที่ต้องการตรวจสอบ ทั้งนี้ KRS จะใช้สำหรับการกู้คืนกุญแจเมื่อหน่วยงานของรัฐบาลต้องการตรวจสอบข้อความที่ต้องสงสัย และการกู้คืนกุญแจของผู้ใช้งานที่สูญหาย หรือกุญแจที่ไม่สามารถใช้งานได้

ในปัจจุบันสถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (National Institute of Standards and Technology: NIST) ได้ประกาศความสำคัญของ KRS ไว้ในเอกสารกรอบการออกแบบระบบบริหารจัดการกุญแจเข้ารหัสลับ (A Framework for Designing Cryptographic Key Management System) [19] โดยได้ระบุว่า การออกแบบระบบการจัดการกุญแจที่ใช้เข้ารหัสลับหรือซีเคเอ็มเอส (Cryptographic Key Management Systems : CKMS) จะต้องระบุนโยบายการกู้คืนกุญแจ และต้องมีวิธีการในการพัฒนาและบังคับใช้นโยบายดังกล่าว

2.2 ระบบการเข้ารหัสลับการกู้คืนกุญแจ

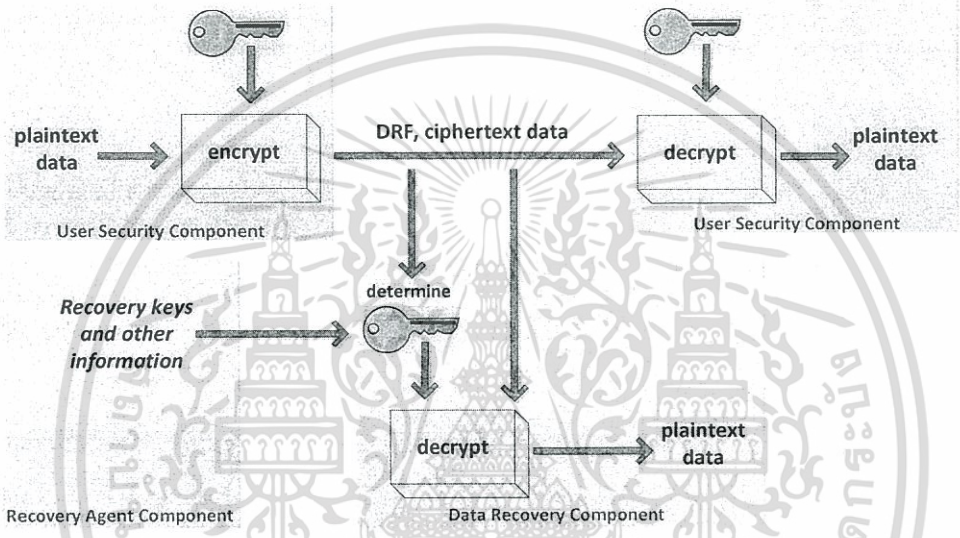
ระบบการเข้ารหัสลับการกู้คืนกุญแจ (A Key Recovery Encryption System) [32, 33] คือระบบที่ให้บริการกู้คืนกุญแจลับ ในกรณีที่กุญแจลับทางฝั่งผู้รับสูญหายหรือเสียหาย หรือเป็นการกู้คืนกุญแจลับเพื่อการเข้าถึงข้อมูลที่ต้องสงสัยโดยชอบด้วยกฎหมาย

ระบบการเข้ารหัสลับการกู้คืนกุญแจ เป็นระบบการเข้ารหัสลับที่อาศัยความสามารถของการสำรอง (Backup) ข้อมูลบางส่วนของกุญแจ เพื่อนำมาใช้ในการถอดรหัสลับ โดยอาศัยความไว้วางใจ (Trust) ยอมให้สิทธิ์ในการเข้าถึงข้อมูลดังกล่าวกับบุคคลที่สาม (Trusted Third Party: TTP) [34] เช่น หน่วยงานของรัฐบาลหรือองค์กรที่จัดตั้งขึ้นโดยถูกต้องตามกฎหมาย เป็นต้น ซึ่งการกู้คืนกุญแจจะต้องอยู่ภายใต้นโยบายหรือกฎหมายที่แน่นอน บุคคลที่สามารถร้องขอการกู้คืนกุญแจและมีสิทธิ์ในกุญแจนั้น คือผู้ที่เป็นเจ้าของกุญแจที่ต้องการกู้คืนหรือหน่วยงาน/บุคคลากรของ

รัฐบาลที่มีสิทธิ์ในการตรวจสอบข้อมูลที่ส่งผ่านระบบเครือข่ายซึ่งต้องการเข้าถึงข้อมูลกุญแจ และนำกุญแจไปถอดรหัสลับข้อมูลเพื่อตรวจสอบอีกครั้งหนึ่ง

2.3 องค์ประกอบของระบบการเข้ารหัสลับการกู้คืนกุญแจ

องค์ประกอบของระบบการเข้ารหัสลับการกู้คืนกุญแจ [32, 33] แบ่งออกเป็น 3 ส่วนหลัก ๆ ซึ่งจะมีการทำงานที่สัมพันธ์กัน ดังแสดงตามรูปที่ 2.1



รูปที่ 2.1 องค์ประกอบของระบบการเข้ารหัสลับการกู้คืนกุญแจ

2.3.1 ส่วนความมั่นคงของผู้ใช้งาน (User Security Component: USCn)

ทำหน้าที่ในการจัดการเรื่องความสามารถในการเข้ารหัสลับและถอดรหัสลับข้อมูล รวมทั้งสนับสนุนการกู้คืนกุญแจ รูปแบบการกู้คืนกุญแจซึ่งจะมีการแนบฟิลด์ที่ใช้ในการกู้คืนกุญแจ (Data Recovery Field: DRF หรือ Key Recovery Field: KRF) ไปกับข้อมูลที่ผ่านการเข้ารหัสลับแล้ว (Ciphertext)

2.3.2 ส่วนหน่วยงานกู้คืนกุญแจ (Recovery Agent Component: RACn)

อาจประกอบด้วยศูนย์กลางการกู้คืนกุญแจ (Key Recovery Center : KRC) และ/หรือ หน่วยงานกู้คืนกุญแจ (KRA) ซึ่งทำหน้าที่ในการบริหารจัดเก็บกุญแจ กู้คืนกุญแจ ใช้กุญแจในการกู้คืนข้อมูล รวมทั้งจัดเก็บข้อมูลอื่น ๆ ที่ช่วยให้การถอดรหัสลับทำได้โดยสะดวก ในกรณีเกิดปัญหากุญแจที่ใช้ในการถอดรหัสลับไม่สามารถใช้งานได้หรือสูญหาย และในส่วนของหน่วยงานนี้อาจมีการใช้ระบบการบริหารจัดการใบรับรองกุญแจสาธารณะ (Public Key Certificate) หรือใช้

โครงสร้างพื้นฐานการบริหารจัดการกุญแจทั่วไป (General Key Management Infrastructure) ร่วมด้วย

2.3.3 ส่วนการกู้คืนข้อมูล (Data Recovery Component: DRCn)

ประกอบไปด้วยอัลกอริทึม โพรโทคอล และกระบวนการขั้นตอนที่ทำให้ได้มาซึ่งกุญแจลับ โดยเอาจาก DRF หรือ KRF ที่แนบไปกับข้อมูลที่ผ่านการเข้ารหัสลับแล้ว เพื่อนำกุญแจที่ได้ไปถอดรหัสลับข้อมูลอีกครั้งหนึ่ง ในส่วนการกู้คืนข้อมูลหรือการกู้คืนกุญแจนี้ จะอนุญาตเฉพาะผู้ที่มีสิทธิ์ในการกู้คืนกุญแจเท่านั้น

2.4 วิธีการ/กระบวนการพื้นฐานสำหรับการกู้คืนกุญแจ [11]

ฟิลด์ที่ใช้ในการกู้คืนกุญแจ หรือที่เรียกว่า เคอาร์เอฟ (KRF) จะถูกสร้างโดยผู้ส่ง และถูกแนบไปกับข้อความที่ผ่านการเข้ารหัสลับแล้ว โดย KRF ประกอบด้วย ส่วนประกอบของกุญแจที่ใช้ในการถอดรหัสลับข้อมูล และเพื่อความมั่นคงของกุญแจ กุญแจจะถูกเข้ารหัสลับ ผู้ที่สามารถถอดรหัสลับกุญแจได้คือ KRA ที่ผู้ส่งเลือกใช้บริการเท่านั้น

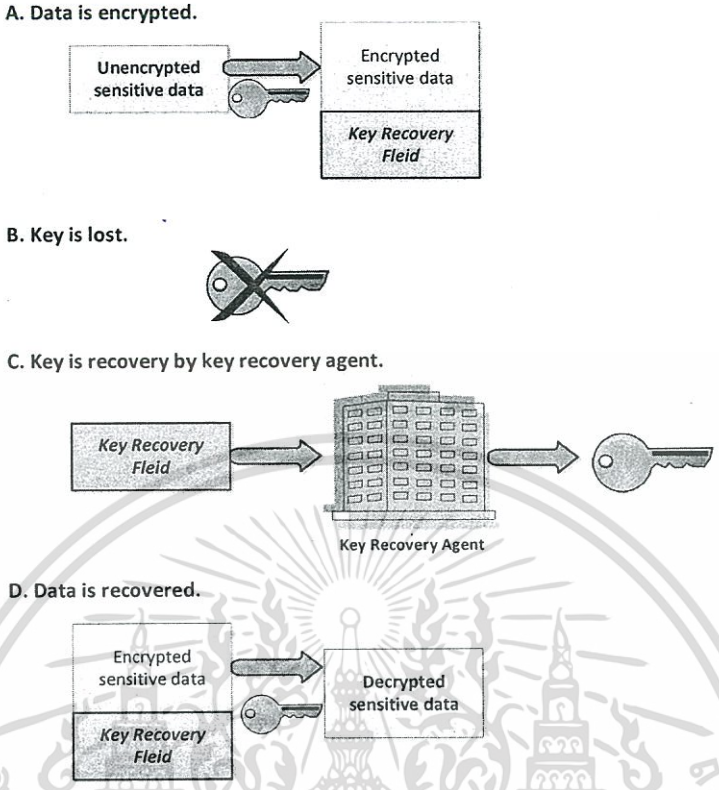
พื้นฐานของกระบวนการกู้คืนกุญแจประกอบไปด้วยขั้นตอนง่าย ๆ โดยได้แสดงตามรูปที่ 2.2 ดังนี้

ขั้นตอนที่ 1 (A) ข้อความต้นฉบับถูกเข้ารหัสลับ และผู้ส่งสร้าง KRF โดยข้อมูลทั้งสองส่วนถูกแนบส่งไปด้วยกัน

ขั้นตอนที่ 2 (B) เมื่อทางฝั่งผู้รับไม่มีกุญแจในการถอดรหัสลับข้อมูล อาจเนื่องมาจากกุญแจหายหรือกุญแจไม่สามารถใช้งานได้

ขั้นตอนที่ 3 (C) ผู้รับร้องขอบริการการกู้คืนกุญแจ โดยส่ง KRF ให้ KRA

ขั้นตอนที่ 4 (D) KRA ทำการกู้คืนกุญแจโดยอาศัยข้อมูลจาก KRF และผู้ส่งสามารถนำกุญแจที่ได้ไปใช้ในการถอดรหัสลับต่อไป



รูปที่ 2.2 กระบวนการพื้นฐานในการกู้คืนกุญแจ

2.5 รูปแบบความไว้วางใจ (Trust Model) ของผู้ให้บริการออกใบรับรอง (Certificate Authority : CA)

ผู้ให้บริการออกใบรับรองเป็นบุคคลที่สาม (TTP) ที่มีรูปแบบความไว้วางใจ [35, 36] ที่มีลำดับชั้นของการออกใบรับรองที่เข้มงวด (Strict Certification Hierarchy) กล่าวคือ ใบรับรองของบุคคลใด ๆ จะถูกสร้างโดยผู้ให้บริการออกใบรับรองเท่านั้น โดยผู้ให้บริการออกใบรับรองสามารถมีได้หลายลำดับชั้น ชั้นบนสุดเรียกว่า ผู้ให้บริการออกใบรับรองหลัก (Root CA) โดยที่ใบรับรองของผู้ให้บริการออกใบรับรองหลักเป็นแบบการรับรองตนเอง (Self-Signed) และเป็นผู้ออกใบรับรองสำหรับผู้ให้บริการออกใบรับรองในชั้นถัดลงมา (Intermediate CA) ที่อยู่ติดกันเป็นลำดับไปเรื่อย ๆ จนไปสิ้นสุดที่ผู้ให้บริการออกใบรับรองสำหรับผู้ใช้งาน

การมีผู้ให้บริการออกใบรับรองแบบหลายลำดับชั้นมีประโยชน์ในแง่ของการแบ่งกลุ่มผู้ใช้งานออกเป็นหลาย ๆ กลุ่ม เนื่องจากมีจำนวนผู้ใช้งานมาก ดังนั้นจึงมีการแบ่งกลุ่มผู้ใช้ตามหน่วยงานของผู้ใช้ตามความสามารถในการใช้งานใบรับรอง หรือตามระดับความรับผิดชอบของผู้ให้บริการออกใบรับรอง เป็นต้น

2.6 รูปแบบของ TTP ที่เกี่ยวข้องกับ KRS

KRS มีการทำงานร่วมกับ TTP ที่ได้รับความไว้วางใจ โดยแบ่งตามหน้าที่ได้เป็น 4 กรณี ดังนี้คือ

2.6.1 TTP ที่ทำหน้าที่ในการกู้คืนกุญแจ

การให้บริการส่วนนี้มีฟังก์ชั่นการทำงานหลากหลายรูปแบบ [5, 6 , 10-14] ขึ้นอยู่กับการให้บริการของแต่ละ TTP วิธีการที่นิยมในงานวิจัยส่วนใหญ่จะใช้วิธีการสร้างกุญแจคู่ของตัวเอง แล้วทำการส่งกุญแจสาธารณะให้กับผู้ใช้งาน (ผู้ส่งข้อมูล) เพื่อนำไปเข้ารหัสลับข้อมูลกุญแจที่ต้องการกู้คืน เมื่อผู้รับไม่สามารถใช้กุญแจถอดรหัสลับได้ TTP จะใช้กุญแจส่วนตัวในการกู้คืนกุญแจนั้น ตัวอย่างของ TTP ที่ทำหน้าที่ดังกล่าวได้แก่ บริการของศูนย์กลางให้การรับรอง Trusted Center (TC) ผู้ให้บริการออกใบรับรอง (Certificate Authority: CA) ศูนย์กลางการกู้คืนข้อมูล (Data Recovery Center: DRC) ศูนย์กลางการกู้คืนกุญแจ (KRC) และหน่วยงานกู้คืนกุญแจ (KRA)

2.6.2 TTP ที่ทำหน้าที่เป็นผู้ให้บริการออกใบรับรองผู้ใช้งาน

หน่วยงานนี้มีหน้าที่ออกใบรับรองผู้ใช้งานหรือใบรับรองดิจิทัล (Digital Certificate) [6, 8, 12-14] ซึ่งข้อมูลของผู้ใช้งานประกอบไปด้วย ข้อมูลส่วนบุคคลและข้อมูลกุญแจสาธารณะ รวมถึงข้อมูลผู้ให้บริการออกใบรับรอง (CA) ด้วย

ผู้ให้บริการออกใบรับรองที่น่าเชื่อถือ จำเป็นต้องมีระบบรักษาความมั่นคงของข้อมูลในระดับสูง เนื่องจากข้อมูลดังกล่าวมีผลต่อการยืนยันตัวบุคคลในการสื่อสารบนเครือข่ายและการประกอบธุรกรรมทางอิเล็กทรอนิกส์

2.6.3 TTP ที่ทำหน้าที่ในการออกใบรับรองหน่วยงานกู้คืนกุญแจ

การออกใบรับรองให้ KRA สามารถกระทำได้โดย CA [12] หน่วยงานผู้ให้บริการที่มีสิทธิ์ในการออกใบรับรอง KRA [11] หรือหน่วยงานออกใบรับรองการจัดเก็บกุญแจ (Certificate Escrow Authority: CEA) โดยเฉพาะก็ได้

2.6.4 TTP ที่ทำหน้าที่ในการให้บริการกุญแจ (Key Distribution)

การให้บริการกุญแจนี้ มีวัตถุประสงค์เพื่อให้ผู้ใช้งานนำกุญแจไปใช้ในการเข้ารหัสลับข้อมูลที่มีการสื่อสารกันในแต่ละช่วงเวลา (Session) เช่น บริการของ Kerberos [37] ศูนย์กลางการกระจายกุญแจ (Key Distribution Center: KDC) [38] เป็นต้น โดยทำการสุ่มค่าตัวเลขเพื่อสร้างกุญแจ และใช้วิธีการที่มั่นคงปลอดภัยในการส่งกุญแจผ่านระบบเครือข่าย

2.7 หน่วยงานกู้คืนกุญแจ (KRA)

KRA จัดเป็น TTP ที่มีหน้าที่ในการบริหารจัดเก็บกุญแจ ทดแทนกุญแจ กู้คืนกุญแจ และใช้กุญแจในการถอดรหัสลับเพื่อกู้คืนข้อมูล รวมทั้งจัดเก็บข้อมูลอื่น ๆ ที่ช่วยให้การถอดรหัสลับทำได้โดยสะดวก ในกรณีเกิดปัญหากุญแจที่ใช้ในการถอดรหัสลับไม่สามารถใช้งานได้หรือสูญหาย หรือในกรณีที่หน่วยงานของรัฐต้องการตรวจสอบข้อมูลต้องสงสัย ซึ่งถือว่าเป็นหน่วยงานที่มีบทบาทและมีหน้าที่ที่สำคัญ

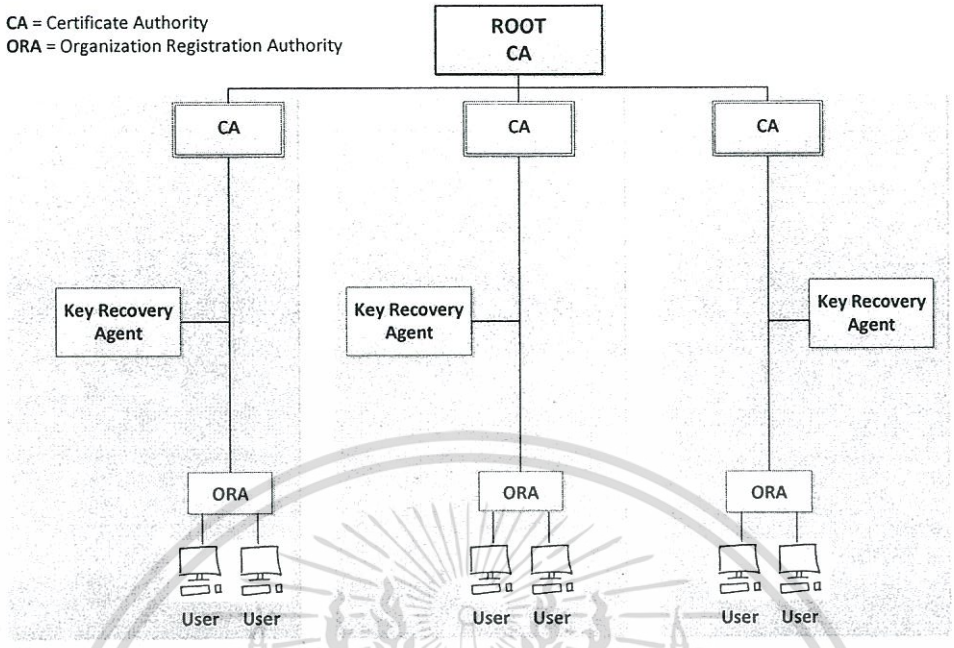
KRA จำเป็นที่จะต้องได้รับการรับรองและยืนยันความน่าเชื่อถือของหน่วยงานจาก TTP เพื่อเป็นการสร้างความไว้ใจ/เชื่อมั่นในเรื่องความมั่นคงและความเป็นส่วนตัวให้กับผู้ที่ใช้บริการ

2.7.1 ตัวอย่างของ KRA

KRA ที่มีการขอใบรับรองหน่วยงานจากหน่วยงานที่ให้บริการออกใบรับรองคือ องค์กรออกใบรับรอง (CA) และองค์กรผู้ได้รับอนุญาตในการออกใบรับรอง KRA มีดังนี้

2.7.1.1 หน่วยงานกู้คืนกุญแจที่ได้รับการรับรองจาก CA

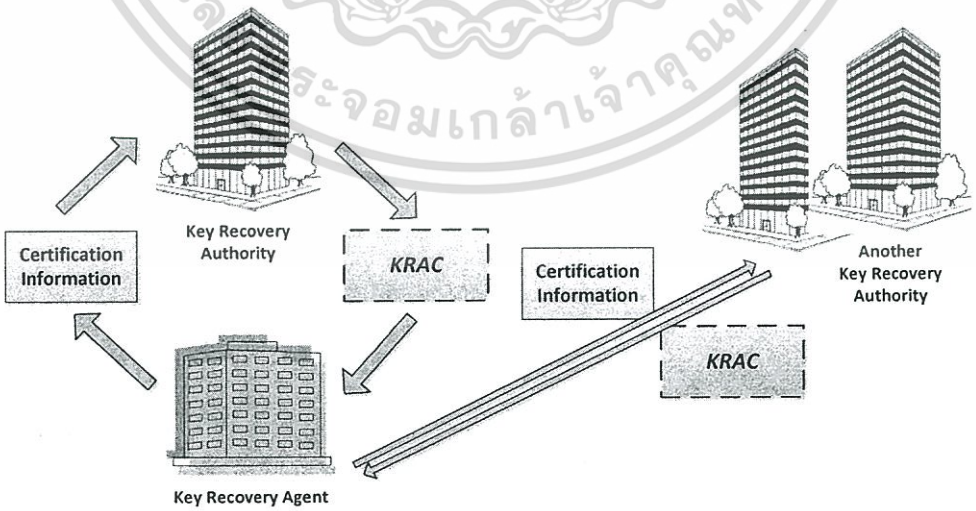
KRA จะทำการส่งข้อมูลส่วนตัวและข้อมูลกุญแจสาธารณะให้กับ CA เพื่อให้ CA ทำการรับรองข้อมูลส่วนตัวของหน่วยงาน และรับรองข้อมูลกุญแจด้วยการออกใบรับรองและประกาศการรับรองนั้นบนเครือข่าย โดย CA จะทำงานประสานกับหน่วยงานรับลงทะเบียน (Organization Registration Authorities: ORA) ซึ่งจะมีหน้าที่ตรวจสอบ/พิสูจน์ผู้ใช้งาน ตรวจสอบการร้องขอ และติดต่อกับ CA จากนั้น ORA จะร้องขอการกู้คืนกุญแจจาก KRA เพื่อให้ได้มาซึ่งกุญแจที่สามารถนำไปถอดรหัสลับข้อมูลต้นฉบับได้ ดังแสดงความสัมพันธ์ของการทำงานตามรูปที่ 2.3



รูปที่ 2.3 การให้การรับรองหน่วยงานที่ให้บริการกู้คืนกุญแจโดย CA

2.7.1.2 KRA ที่ได้รับการรับรองจาก Key Recovery Authority หรือ KRAu

KRA สามารถขอใบรับรองหน่วยงานได้จาก KRAu ดังแสดงตามรูปที่ 2.4 โดยส่งข้อมูลคำร้องและข้อมูลการขอใบรับรองไปยังหน่วยงานดังกล่าว จากนั้นก็จะได้รับใบรับรองหน่วยงานกู้คืนกุญแจ (Key Recovery Agent Certificate: KRAC) ที่มีการลงลายมือชื่อจาก KRAu เรียบร้อยแล้ว ทั้งนี้ KRA สามารถส่งคำร้องและข้อมูลการขอใบรับรองไปให้หลาย ๆ KRAu รับรองได้ด้วย



รูปที่ 2.4 กระบวนการขอใบรับรองจาก KRAu

2.7.2 การให้บริการของ KRA สามารถแบ่งตามรูปแบบการจัดการกุญแจได้ 4 รูปแบบ ดังแสดงตามรูปที่ 2.5

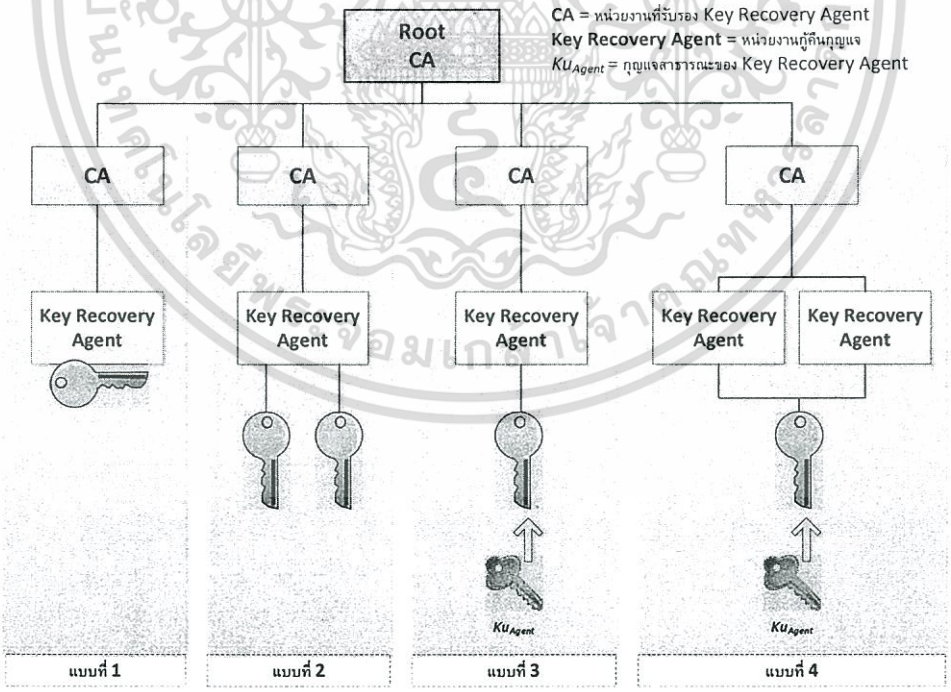
รูปแบบที่ 1 กุญแจต้นฉบับถูกจัดเก็บที่ KRA หรือหน่วยงานที่ทำหน้าที่ในการกู้คืนกุญแจ โดยหน่วยงานที่รับผิดชอบกุญแจจะสามารถเข้าถึงข้อมูลกุญแจได้โดยตรง

รูปแบบที่ 2 กุญแจถูกแยกส่วน และมีการจัดเก็บแยกกันตามตำแหน่งของการจัดเก็บ เมื่อต้องการกู้คืนกุญแจ KRA จะทำหน้าที่ในการนำส่วนที่แยกกันอยู่มารวมกันเป็นข้อมูลกุญแจ

รูปแบบที่ 3 KRA ไม่จัดเก็บกุญแจของผู้ใช้งาน แต่จะให้ผู้ใช้งานนำกุญแจสาธารณะของหน่วยงานไปเข้ารหัสลับข้อมูลกุญแจ ซึ่งจัดเก็บเป็นฟิลด์ที่ใช้ในการกู้คืนกุญแจ และเมื่อต้องการกู้คืนกุญแจ KRA ก็จะใช้กุญแจส่วนตัวในการถอดรหัสลับและทำการกู้คืนกุญแจให้ได้

รูปแบบที่ 4 ให้ KRA ทำงานร่วมกันมากกว่าหนึ่งหน่วยงาน แต่เป็นการนำกุญแจสาธารณะของหน่วยงานมาเข้ารหัสลับชิ้นส่วนกุญแจ ซึ่งจัดเก็บไว้ใน KRF เมื่อต้องการกู้คืนกุญแจ ต้องใช้กุญแจส่วนตัวของ KRA ถอดรหัสลับข้อมูลกุญแจเช่นเดียวกัน

การให้ KRA ทำงานร่วมกันมากกว่าหนึ่งหน่วยงาน เป็นการเพิ่มความมั่นคงในการกู้คืนกุญแจ ผู้ใช้งานมีความเป็นส่วนตัวมากขึ้น แต่อาจเป็นการเพิ่มความซับซ้อนและเพิ่มระยะเวลาในการกู้คืนกุญแจ



รูปที่ 2.5 รูปแบบการจัดการกุญแจของหน่วยงานกู้คืนกุญแจ

2.8 การห่อหุ้มกุญแจ (Key Encapsulation)

เมื่อนำกุญแจลับ (Secret Key: *Ks*) หรือส่วนประกอบของกุญแจลับมาผ่านกระบวนการเข้ารหัสลับ พร้อมด้วยข้อมูลสำหรับการกู้คืนกุญแจ จากนั้นถูกห่อหุ้ม (Capsule)ไว้ เรียกว่า Key Encapsulation ซึ่งมีลักษณะเป็นบล็อกหรือฟิลด์ที่ใช้สำหรับการกู้คืนกุญแจ (Key Recovery Block: KRB หรือ Key Recovery Field: KRF) ผู้ที่สามารถถอดรหัสลับบล็อกหรือฟิลด์นี้ได้ คือ หน่วยงานที่มีหน้าที่รับผิดชอบในการกู้คืนกุญแจ (KRC หรือ KRA) เมื่อผู้ใช้งานต้องการกู้คืนกุญแจ ฟิลด์ KRB หรือ KRF จะถูกส่งไปยังหน่วยงานกู้คืนกุญแจ เพื่อทำการกู้คืนกุญแจลับดังกล่าว แล้วส่งให้กับผู้ร้องขอการกู้คืนต่อไป

การใช้วิธีการห่อหุ้มกุญแจลับนี้ ช่วยลดความเสี่ยงในเรื่องของการคุกคามความเป็นส่วนตัวของเจ้าของกุญแจได้

2.9 การแชร์ความลับ (Secret Sharing)

การแชร์ความลับ หรือการแบ่งปันความลับ [28] หมายถึง วิธีการกระจายความลับไปยังสมาชิกของกลุ่มที่เกี่ยวข้องกัน เปรียบเสมือนการจัดสรรชิ้นส่วนของความลับให้กับผู้ที่เกี่ยวข้องในกลุ่ม การที่จะได้มาซึ่งความลับนั้นจะต้องนำชิ้นส่วนทุกชิ้นส่วนที่กระจายไปยังสมาชิกในกลุ่มมาประกอบกันให้ครบทุกชิ้นส่วน โดยจะมีการกำหนดให้สมาชิกในกลุ่ม เท่ากับ n

วิธีการแชร์ความลับขั้นพื้นฐานหรือการแบ่งปันความลับอย่างง่าย คือ การแบ่งและจัดสรรความลับ (S_C) ออกเป็น n ชิ้น เพื่อแจกให้กับสมาชิกในกลุ่ม n คน ซึ่งจะอาศัยการสุ่มค่าตัวเลข (Random Number) และการคำนวณทางคณิตศาสตร์ คือ Exclusive OR (XOR) เข้ามาช่วยในการแชร์ความลับ และการรวมความลับ โดยมีขั้นตอนอย่างง่ายดังนี้

1) การแชร์ความลับ

1.1) การแชร์ความลับทำได้ด้วยการสุ่มค่าตัวเลข (S_R) มาจำนวน n-1 ตัว คือ S_{R1} , S_{R2} , ..., $S_{R\,n-1}$ ดังนั้นจากขั้นตอนนี้จะสามารถแชร์ความลับให้กับสมาชิกในกลุ่ม n-1 เอเจนต์

1.2) นำตัวเลขที่ได้จากการสุ่มข้างต้นทุกตัวมา XOR กัน และ XOR กับ S_C เพื่อเป็นค่าความลับสำหรับสมาชิกเอเจนต์ที่ n ซึ่งสามารถแสดงได้ดังนี้

$$S_{Rn} = S_{R1} \oplus S_{R2} \oplus \ldots \oplus S_{R\,n-1} \oplus S_C$$

2) การรวมความลับ

2.1) การรวมความลับหรือการได้มาซึ่งความลับ เกิดจากการนำส่วนประกอบของความลับทั้งหมดตั้งแต่ S_{R1} ถึง S_{Rn} มาทำการ XOR กัน สามารถแสดงได้ดังนี้

$$S_C = S_{R1} \oplus S_{R2} \oplus \ldots \oplus S_{Rn}$$

งานวิจัยนี้ได้นำทฤษฎีการแชร์ความลับมาใช้ในการออกแบบกระบวนการจัดสรรส่วนประกอบของกุญแจ เพื่อจัดเก็บไว้ใน *KRF* ของแต่ละเอเจนต์ในกลุ่มการกู้คืน ทั้งนี้จะทำให้สะดวกและง่ายในกระบวนการกู้คืนกุญแจ

2.10 เพาเวอร์เซต (Power Set)

ทฤษฎีเพาเวอร์เซต [29] ในทางคณิตศาสตร์ หากกำหนดให้ S แทนเซต ดังนั้นสามารถเขียนเพาเวอร์เซตของเซต S ได้ดังนี้ $P(S)$ ซึ่งก็คือ สมาชิกทั้งหมดเป็นซับเซตของเซต S ใช้สัญลักษณ์ $P(S) = \{x \mid x \subset S\}$

ถ้า S เป็นเซตจำกัด

ถ้า $n(S) = k$ แล้ว

1) $n\,[P(S)] = 2^k$

2) $n\,[P(P(S))] = 2^{2^k}$

ทฤษฎีเกี่ยวกับเพาเวอร์เซต : ถ้า A และ B เป็นเซตจำกัดใด ๆ

1) สมาชิกทุกตัวของเพาเวอร์เซต ต้องเป็นเซต

2) $\phi \in P(A)$ และ $\phi \subset P(A)$ เสมอ

3) $A \in P(A)$ เสมอ แต่ A ไม่จำเป็นต้องเป็นสับเซตของ $P(A)$

4) เมื่อ $A \in P(A)$ ดังนั้น $P(A) \in P(P(A))$

5) เพาเวอร์เซต จะไม่มีทางเป็นเซตว่างได้เลยนั่นคือ $P(A) \neq \phi$

6) $P(\phi) = \{\phi\}$

7) $\{A\} \subset P(A)$ เสมอ ดังนั้น $\{P(A)\} \subset P(P(A))$

8) $P(A \cap B) = P(A) \cap P(B)$

9) ถ้า $A \subset B$ แล้ว $P(A) \subset P(B)$

ตัวอย่าง ถ้า $S = \{x, y, z\}$ แล้ว สมาชิกของซับเซต S คือ { } หรือ ϕ, $\{x\}$, $\{y\}$, $\{z\}$, $\{x, y\}$, $\{x, z\}$, $\{y, z\}$ และ $\{x, y, z\}$

ดังนั้น $P(S) = \{\{\}, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$

งานวิจัยนี้ได้นำทฤษฎีเพาเวอร์เซตมาใช้ในการออกแบบกระบวนการสำรองส่วนประกอบของกุญแจ สำหรับการกู้คืนกุญแจเมื่อมีบาง KRA ในกลุ่มการกู้คืนกุญแจล่มหรือไม่สามารถให้บริการการกู้คืนกุญแจได้

2.11 ปัญหา ความเสี่ยง และอุปสรรคของระบบการกู้คืนกุญแจ

ผลกระทบที่ปรากฏจากระบบกู้คืนกุญแจสามารถอธิบายได้ 3 ทิศทาง คือ ด้านความเสี่ยง (Risk) ความซับซ้อนของระบบ (Complexity) และต้นทุนทางเศรษฐศาสตร์ (Economic Cost)

ด้านความเสี่ยง : ระบบมีความเสี่ยงต่อความล้มเหลวในการกู้คืนกุญแจ ซึ่งสามารถส่งผลกระทบที่เป็นอันตรายต่อการจัดการระบบให้มีความเหมาะสม โดยต้องอยู่ภายใต้การดำเนินการที่ทำให้เป็นความลับ และอยู่บนหลักการของความมั่นคงของระบบการเข้ารหัสลับ รวมถึงความเสี่ยงต่อการเกิดภัยคุกคามที่นำไปสู่การเปิดเผยกุญแจ การขโมยข้อมูลกุญแจ หรือความล้มเหลวในการเข้าถึงข้อมูลกุญแจอย่างถูกต้องตามกฎหมาย

ความซับซ้อนของระบบ : ถึงแม้ว่ามีความเป็นไปได้ที่จะทำการกู้คืนกุญแจให้กับผู้ใช้งานได้โดยตรง กระบวนการทำงานพื้นฐานของระบบกู้คืนกุญแจถือว่าเป็นระบบที่พิเศษ ซึ่งมีความซับซ้อน และด้วยจำนวนเอนทิตี้ จำนวนกุญแจ จำนวนการร้องขอการกู้คืนกุญแจ รวมทั้งจำนวนความต้องการการตอบสนองในระบบที่เกิดขึ้นเป็นจำนวนมาก ส่งผลให้การกู้คืนกุญแจไม่สามารถพัฒนาวิธีการทำงานด้วยหลักการของการเข้ารหัสลับลับขั้นพื้นฐานได้ ซึ่งโดยลักษณะของระบบมีความซับซ้อนและยากในการพัฒนา

ต้นทุนทางเศรษฐศาสตร์ : ต้นแบบของระบบการกู้คืนกุญแจยังมีน้อยและไม่เหมาะสมกับการลงทุน อย่างไรก็ตามมีความเป็นไปได้ที่จะทำการวัดประเมินคุณภาพเกี่ยวกับองค์ประกอบพื้นฐานของระบบ วิธีการแชร์การกู้คืนกุญแจ เหล่านี้ล้วนมีผลกระทบต่อต้นทุนในการออกแบบพัฒนาระบบ รวมถึงการนำระบบไปใช้ และการบริหารจัดการระบบ

ผลกระทบทั้งสามทิศทางนี้ส่งผลในแง่ลบต่อการพัฒนาระบบการกู้คืนกุญแจ อย่างไรก็ตามมีการแสดงให้เห็นเป็นข้อมูลเชิงตัวเลขว่า ผลิตภัณฑ์ที่เกี่ยวข้องกับระบบการกู้คืนกุญแจมีมากถึงหนึ่งพันผลิตภัณฑ์ที่มีอยู่ในท้องตลาด มีหน่วยงานที่ทำหน้าที่จัดเก็บกุญแจที่ถูกสร้างขึ้นทั่วโลกกว่าหนึ่งพันหน่วยงาน มีกฎหมายหรือนโยบายของรัฐบาลหลายหมื่นฉบับที่ต้องการเข้าถึงข้อมูลอย่างถูกกฎหมาย มีผู้ใช้งานระบบที่มีการเข้ารหัสข้อมูลหลายล้านคน มีกุญแจในการเข้ารหัสลับเกิดขึ้นหลายสิบล้าน และมีกุญแจลับเกิดขึ้นหลายร้อยพันล้าน ทำให้โครงสร้างพื้นฐานที่มีอยู่ ยากที่จะรองรับการใช้งานที่เพิ่มมากขึ้นตามลำดับ

ในกรณีการเกิดจำนวนผู้ใช้งานที่เพิ่มมากขึ้นและเกิดจำนวนข้อมูลที่มากมายขึ้นตามไปด้วยนั้น ส่งผลให้การบริการของระบบการกู้คืนกุญแจเกิดปัญหาคอขวด และส่งผลต่อความมั่นคง

ปลอดภัย การมอบความไว้วางใจให้บุคคลที่สามที่จะต้องมีข้อจำกัดบางอย่าง ถ้าทุกข้อมูลถูกส่งและกระทำการผ่านหน่วยงานกู้คืนกุญแจ ย่อมส่งผลให้การทำงานของระบบและเครือข่ายช้าลง

หากพิจารณาถึงระบบกู้คืนกุญแจแบบหลายเอเจนต์ จะพบปัญหาที่เพิ่มเติมคือ เมื่อปรากฏมีเอเจนต์ใดเอเจนต์หนึ่งในกลุ่มการกู้คืนล่ม จะทำให้ระบบไม่สามารถให้บริการกู้คืนกุญแจได้ เนื่องจากกุญแจต้นฉบับจะได้มาจากการคำนวณส่วนประกอบของกุญแจ ซึ่งจะเป็นหน้าที่ของเอเจนต์ เมื่อมีเอเจนต์หนึ่งไม่สามารถคำนวณส่วนประกอบของกุญแจได้ จะส่งผลให้ไม่สามารถประกอบกุญแจต้นฉบับได้สำเร็จ และปัญหาเอเจนต์ปลอมที่พยายามเข้ามาเป็นสมาชิกของกลุ่มการกู้คืนกุญแจก็เป็นอีกหนึ่งปัญหาที่ต้องได้รับการพิจารณาแก้ไข

2.12 การสำรวจงานวิจัยที่เกี่ยวข้องกับระบบการกู้คืนกุญแจ

งานวิจัยและพัฒนาเกี่ยวกับระบบการเข้ารหัสลับการกู้คืนกุญแจ ได้มีผู้นำเสนอไว้มากมาย มีหลายงานวิจัยที่ได้นำเสนอวิธีการในการกู้คืนกุญแจที่น่าสนใจ ซึ่งจะเป็นพื้นฐานสำหรับการพัฒนา ปรับปรุง ค้นคว้าทำวิจัยการกู้คืนกุญแจต่อไป และมีรายละเอียดอย่างย่อของงานวิจัยดังนี้

2.12.1 Commercial Key Recovery

งานวิจัยนี้ได้นำเสนอวิธีการของระบบการจัดเก็บกุญแจและการกู้คืนกุญแจ [5] ซึ่งการวิจัยได้เน้นเรื่องการให้ความเป็นส่วนตัวจากหน่วยงานที่สามารถเข้าถึงข้อมูลได้อย่างถูกกฎหมายแก่ผู้ใช้งาน โดยได้กล่าวถึงการจัดเก็บกุญแจไว้กับซอฟต์แวร์ที่มีหน่วยงานของรัฐเป็นผู้รับผิดชอบ วิธีการนี้ทำให้ความเป็นส่วนตัวของผู้ใช้งานลดน้อยลง แต่สนับสนุนให้มีการเข้าถึงข้อมูลจากรัฐทำได้ง่ายขึ้น และวิธีการกู้คืนกุญแจจากฟิลด์ที่ใช้ในการกู้คืนกุญแจโดยไม่ต้องจัดเก็บกุญแจแต่อย่างใด

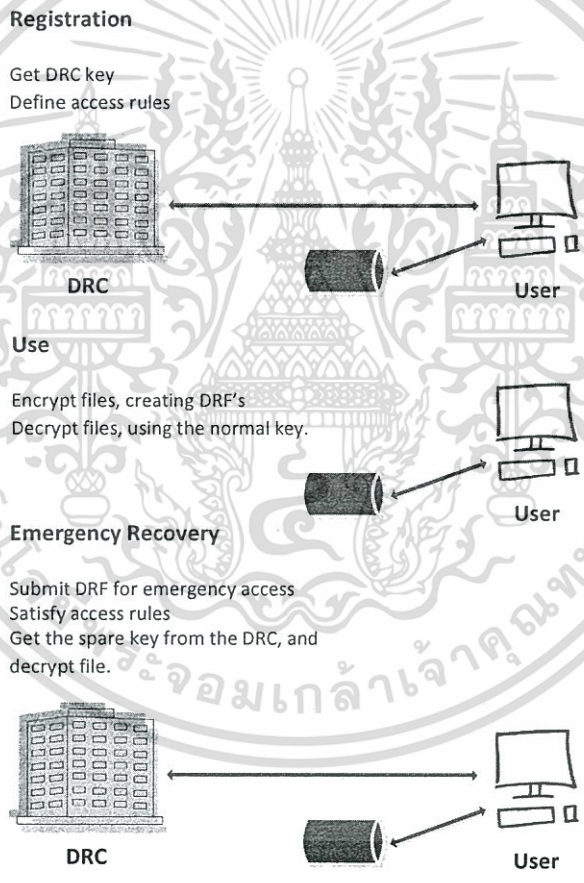
ดังนั้นงานวิจัยนี้จึงเสนอการออกแบบระบบเป็นสองขั้น คือ การจัดเก็บกุญแจ (Clipper Software Key Escrow : SKE) และการกู้คืนกุญแจ (Commercial Key Recovery : CKR)

ขั้นที่ 1 การจัดเก็บกุญแจ หรือ SKE เริ่มต้นด้วยการสร้างระบบ (โปรแกรม) SKE ภายใต้เงื่อนไขการเข้าถึงข้อมูลโดยชอบด้วยกฎหมาย ซึ่งถูกออกแบบมาให้ทำงานคู่ขนานไปกับนโยบายการจัดเก็บกุญแจโดยหน่วยงานของรัฐบาล (Government Key Escrow) จะใช้กุญแจคู่ในระบบการทำงาน โดยหน่วยงานของรัฐบาลจะเป็นผู้จัดเก็บส่วนประกอบของกุญแจส่วนตัวไว้กับโปรแกรม และนำกุญแจสาธารณะมาสร้าง LEAF (Law Enforcement Access Field) เพื่ออำนวยต่อการเข้าถึงข้อมูลกุญแจของผู้ใช้งาน เนื่องจากหน่วยงานของรัฐบาลสามารถเข้าถึงข้อมูลกุญแจได้จากฟิลด์นี้

ขั้นที่ 2 การกู้คืนกุญแจ (Commercial Key Recovery : CKR) เป็นโปรแกรมระบบที่ให้บริการแบบเร่งด่วนในการกู้คืนข้อมูลที่มีการเข้ารหัสลับลับ โดยใช้ฟิลด์ในการกู้คืนข้อมูล (Data Recovery Field : DRF) ดังแสดงขั้นตอนตามรูปที่ 2.6 ซึ่ง CKR จะต้องทำความร่วมมือกับศูนย์กลาง

การกู้คืนข้อมูล (Data Recovery Center : DRC) โดยผู้ใช้ต้องติดตั้งโปรแกรม RE (Recovery-Enabled : RE) เพื่อลงทะเบียนผ่านโปรแกรม ในการลงทะเบียนนั้นผู้ใช้จะต้องส่งข้อมูลที่จำเป็นในการยืนยันตัวตน เพื่อนำข้อมูลไปใช้ในการกู้คืนกุญแจ ระหว่างการลงทะเบียน DRC จะส่งค่าระบุตัวตนของผู้ใช้งาน พร้อมกับกุญแจสาธารณะและค่าระบุตัวตนของ DRC กลับไปยังผู้ใช้

โปรแกรม RE จะถูกใช้ในการเข้ารหัสลับข้อมูล หรือข้อความจากฝั่งผู้ส่ง โดยใช้กุญแจเซสชั่น ซึ่งจะมีการเพิ่มฟิลด์กู้คืนข้อมูล หรือ DRF ไปพร้อมกับข้อมูลนั้นด้วย ฟิลด์นี้จะถูกเข้ารหัสลับด้วยกุญแจสาธารณะของ DRC โดยวิธีการนี้จะไม่มีการจัดเก็บกุญแจไว้ในฐานข้อมูล เมื่อผู้รับต้องการกู้คืนข้อมูลก็ทำการส่ง DRF ไปให้ DRC ซึ่ง DRC จะใช้กุญแจส่วนตัวในการถอดรหัสลับ DRF เพื่อเข้าถึงข้อมูลกุญแจ ส่งให้กับผู้รับต่อไป



รูปที่ 2.6 การดำเนินการของ Commercial Key Recovery

2.12.2 NIST Key recovery

งานวิจัยนี้นำเสนอการกู้คืนกุญแจ [12] โดยเน้นการมอบความไว้วางใจให้กับผู้ให้บริการออกใบรับรองหรือ CA ที่มีการทำงานแบบลำดับชั้น และมีการทำงานร่วมกับ KRA และORA โดยแต่ละหน่วยงาน มีลักษณะการทำงานดังนี้

CA ทำการตรวจสอบและออกใบรับรองกุญแจสาธารณะแก่ผู้ใช้งานในระบบ

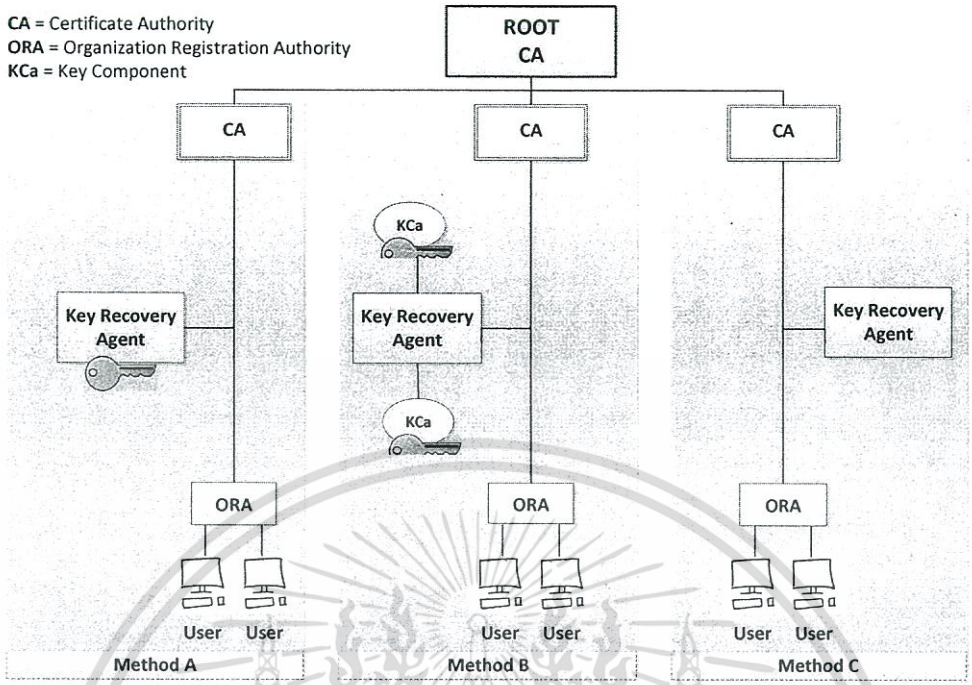
ORA ทำหน้าที่รับการร้องขอการกู้คืนกุญแจ ตรวจสอบ/พิสูจน์ตัวจริงของผู้ใช้งาน ตรวจสอบการร้องขอ และดำเนินการตามขั้นตอนกระบวนการร้องขอการกู้คืนกุญแจ เพื่อให้ได้มาซึ่งกุญแจที่สามารถนำไปถอดรหัสลับข้อมูลต้นฉบับได้อย่างถูกต้อง

KRA ทำหน้าที่ในการกู้คืนกุญแจตามลักษณะการจัดเก็บกุญแจ ซึ่งได้แบ่งการให้บริการตามการเข้าถึงข้อมูลกุญแจไว้ 3 วิธีการ ดังแสดงในรูปที่ 2.7 คือ

วิธีการที่ 1 (Method A) กุญแจถูกจัดเก็บและสามารถเข้าถึงโดยตรงโดย KRA เป็นวิธีการที่ผู้ใช้งานไม่มีความเป็นส่วนตัวและไม่มีมั่นคงปลอดภัยมากนัก แต่วิธีการในการกู้คืนกุญแจไม่ซับซ้อน ทำให้ง่ายในการกู้คืน

วิธีการที่ 2 (Method B) กุญแจถูกแยกส่วนและมีการจัดเก็บแยกกันตามตำแหน่งของการจัดเก็บ เมื่อต้องการกู้คืนกุญแจ KRA จะทำหน้าที่ในการนำส่วนที่แยกกันอยู่มารวมกันเป็นข้อมูลกุญแจ

วิธีการที่ 3 (Method C) วิธีการนี้ KRA ไม่จัดเก็บกุญแจของผู้ใช้งาน แต่ให้ผู้ใช้งานนำกุญแจสาธารณะของ KRA ไปเข้ารหัสลับกุญแจ และเมื่อต้องการกู้คืนกุญแจ KRA ก็จะใช้กุญแจส่วนตัวในการถอดรหัสลับ และทำการกู้คืนให้ได้ เป็นวิธีการที่ได้รับการยอมรับจากผู้ใช้การบริการกู้คืนกุญแจมากที่สุด



รูปที่ 2.7 ลักษณะการให้บริการของหน่วยงานกู้คืนกุญแจ

จากการศึกษางานวิจัยนี้ ผู้วิจัยได้ใช้หลักการของการให้บริการเข้าถึงข้อมูลกุญแจในรูปแบบที่ 3 ซึ่งเป็นวิธีการที่ได้รับการยอมรับจากผู้ใช้บริการ การกู้คืนกุญแจมากที่สุด มาประกอบการออกแบบกระบวนการกู้คืนกุญแจ

2.12.3 Cryptographic Key Recovery Entry

งานวิจัยนี้นำเสนอวิธีการกู้คืนกุญแจทั้งกุญแจลับและกุญแจส่วนตัว [6] โดยใช้ฟิลด์ในการกู้คืนกุญแจลับ ที่เรียกว่า Key Recovery Entry (KRE) ซึ่งเป็นฟิลด์ที่มีขนาดเล็ก โดยจะถูกสร้างและแนบไปกับข้อมูล เมื่อผู้รับต้องการใช้บริการระบบกู้คืนกุญแจจะใช้ฟิลด์นี้ส่งให้กับหน่วยงานกู้คืนกุญแจ เพื่อทำการกู้คืนกุญแจต่อไป ส่วนการกู้คืนกุญแจส่วนตัวจะใช้ฟิลด์ที่เรียกว่า Key Recovery Field (KRF) โดยมี CA ทำหน้าที่เป็นหน่วยงานกู้คืนกุญแจ ลักษณะของการพิสูจน์ตัวจริงของผู้ใช้งานจะใช้การถาม-ตอบ โดยเก็บคำถามและคำตอบที่ผ่านกระบวนการของฟังก์ชันแฮช (Hash Function) ของผู้ใช้งานไว้ในฐานข้อมูล

จากการศึกษางานวิจัยนี้ ผู้วิจัยได้นำหลักการของการใช้ฟังก์ชันแฮชมาประยุกต์ใช้สำหรับการพิสูจน์ตัวจริงของเอเจนต์ที่ทำหน้าที่กู้คืนกุญแจในเบื้องต้น

2.12.4 LEAF for Key Escrow System

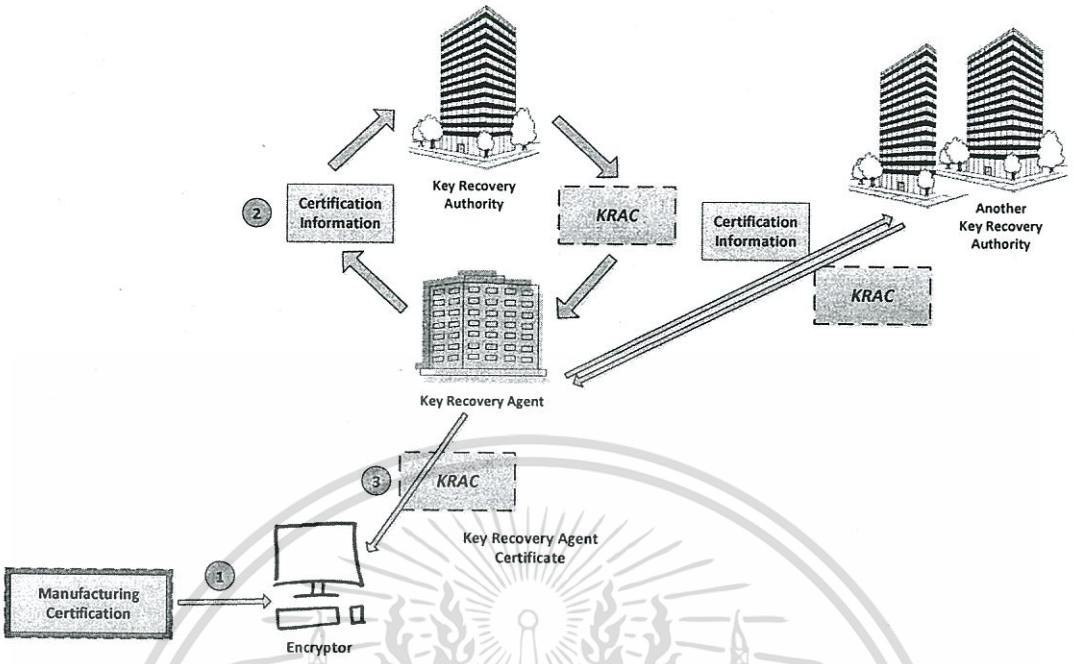
งานวิจัยนี้นำเสนอการจัดเก็บและการกู้คืนกุญแจ [10] โดยได้มุ่งประเด็นการกู้คืนกุญแจเมื่อกุญแจหาย และการตรวจสอบข้อมูลที่มีการเข้ารหัสลับและส่งผ่านระบบเครือข่ายอย่างถูกกฎหมาย โดยใช้ฟิลด์ในการกู้คืนกุญแจที่เรียกว่าลีฟ (Law Enforcement Access Field: LEAF) ซึ่งมีศูนย์กลางให้การรับรอง (Trusted Center: TC) ทำหน้าที่ในการกู้คืนกุญแจ จุดเด่นของงานวิจัยนี้คือ กุญแจที่ใช้ในการเข้ารหัสข้อมูล (M) ซึ่งจะมีการเข้ารหัสลับข้อมูลด้วยกุญแจ $KS \oplus [KS]PK_{TC}$ คือ มีการสร้างกุญแจสำหรับการเข้ารหัสลับข้อมูล จากการนำกุญแจลับ (*KS*) มา Exclusive-OR (XOR) กับ *KS* ที่ถูกเข้ารหัสลับด้วยกุญแจสาธารณะของ TC ($[KS]PK_{TC}$) ทั้งนี้จะช่วยเพิ่มความมั่นคงปลอดภัยให้กับตัวกุญแจและข้อมูล แต่จะทำให้ระบบมีกระบวนการทำงานที่ซับซ้อนเพิ่มขึ้น

2.12.5 Cylink's Key Recovery

งานวิจัยนี้ได้นำเสนอลักษณะการทำงานของระบบกู้คืนกุญแจ [11] โดยมีฟิลด์ที่ใช้ในการกู้คืนกุญแจที่เรียกว่า เคอาร์เอฟ (Key Recovery Field : KRF) ซึ่งผู้ใช้งานจะต้องมีการเลือกใช้บริการจากเอเจนต์ในการกู้คืนกุญแจ (Key Recovery Agent : KRA) โดยกุญแจจะถูกจัดเก็บและปกป้องไว้ใน KRF ซึ่งเป็นส่วนที่นำมาใช้ในการกู้คืนกุญแจ ผู้ที่ทำหน้าที่ในการกู้คืน คือ KRA เท่านั้น

Cylink's Key Recovery สนับสนุนการกู้คืนกุญแจสำหรับข้อมูลที่ถูกเข้ารหัสลับบนพื้นฐานของความสัมพันธ์แบบไว้วางใจ (Trusted Relationship) ระหว่างส่วนของผู้ใช้งาน ส่วนของ KRA และหน่วยงานผู้ให้บริการที่มีสิทธิ์ในการออกใบรับรองหน่วยงานกู้คืนกุญแจ (Key Recovery Authority: KRAu) ดังแสดงตามรูปที่ 2.8 และงานวิจัยนี้ได้เน้นถึงความต้องการพื้นฐานสำหรับระบบการกู้คืนกุญแจที่น่าสนใจ คือ ระบบต้องสามารถกู้คืนกุญแจที่ผู้ใช้งานทำหาย หรือเกิดกรณีที่กุญแจไม่สามารถใช้งานได้ ผู้ใช้งานจะต้องเลือกใช้บริการจาก KRA ระบบจะกู้คืนเฉพาะกุญแจเท่านั้น จะไม่มีการเปิดเผยข้อมูล และระบบจะต้องสอดคล้องกับกฎระเบียบและข้อบังคับของรัฐบาลที่ว่าด้วยเรื่องการนำเข้าและส่งออกผลิตภัณฑ์ที่เกี่ยวข้องกับเทคโนโลยีการเข้ารหัสลับ

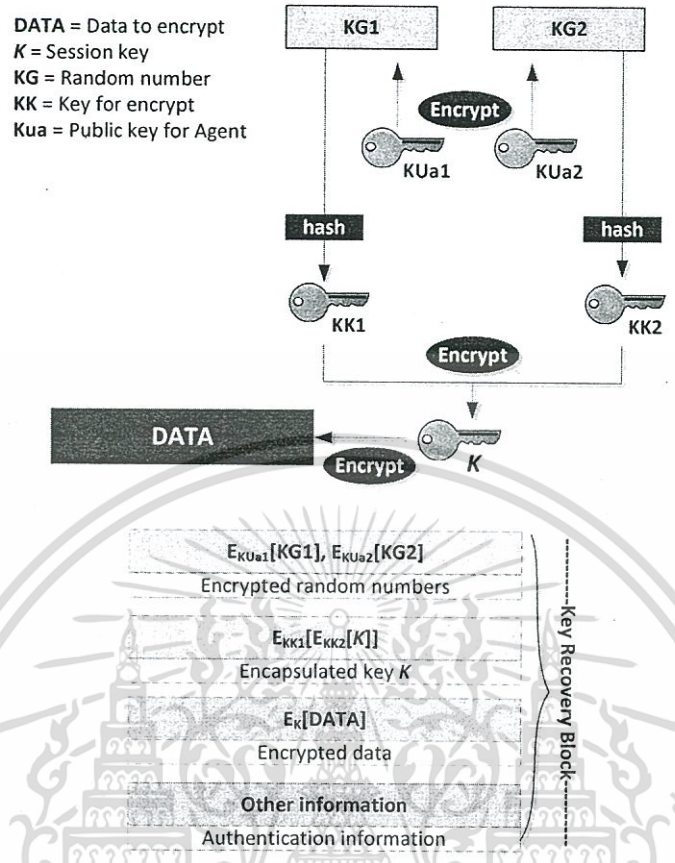
จากการศึกษางานวิจัยนี้ ผู้วิจัยได้นำแนวคิดและหลักการของระบบการกู้คืนกุญแจบางส่วนมาใช้ในการพัฒนากระบวนการกู้คืนกุญแจ อาทิเช่น หลักการที่ว่าระบบจะกู้คืนเฉพาะตัวกุญแจเท่านั้นแต่จะไม่เปิดเผยส่วนของข้อมูล เป็นต้น



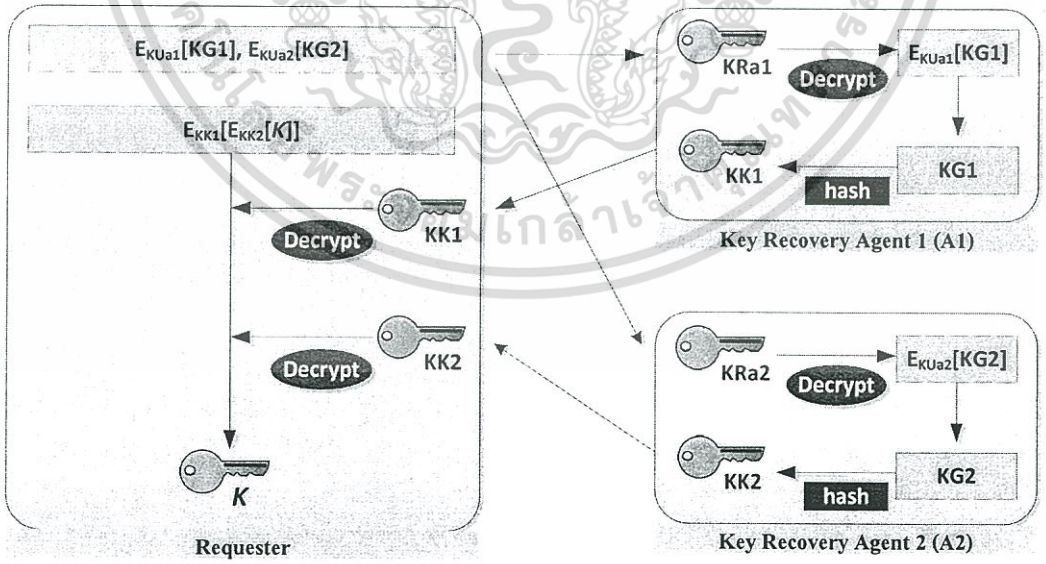
รูปที่ 2.8 การสร้างความสัมพันธ์แบบไว้วางใจระหว่างผู้ใช้งาน หน่วยงานกู้คืนกุญแจ และหน่วยงานผู้ให้บริการที่มีสิทธิ์ในการออกใบรับรองหน่วยงานกู้คืนกุญแจ

2.12.6 Internet Archiving Server with Key Recovery Function

งานวิจัยนี้ได้นำเสนอวิธีการกู้คืนกุญแจแบบ M-KRS โดยไม่อาศัย KRC [21] ใช้ฟิลด์ในการกู้คืนกุญแจที่เรียกว่า เคอาร์บี (Key Recovery Block: KRB) โดยมีวิธีการสร้าง KRB ดังแสดงในรูปที่ 2.9 และให้ KRA ทำหน้าที่ในการกู้คืนกุญแจ ดังแสดงในรูปที่ 2.10 ระบบจะสามารถกู้คืนกุญแจได้อย่างถูกต้องในกรณีที่ KRA ทำงานครบทุกเอเจนต์ หากมีเอเจนต์ใดเอเจนต์หนึ่งขัดข้อง คือ ไม่สามารถให้บริการกู้คืนกุญแจได้ จะส่งผลให้ระบบไม่สามารถกู้คืนกุญแจได้เช่นกัน



รูปที่ 2.9 การสร้างฟิลด์สำหรับการกู้คืนกุญแจ



รูปที่ 2.10 การกู้คืนกุญแจ

จากการศึกษางานวิจัยนี้ ผู้วิจัยได้แนวคิดเกี่ยวกับกระบวนการกู้คืนกุญแจแบบ M-KRS ที่ไม่ใช้ KRC และได้ค้นพบปัญหาสำหรับการกู้คืนกุญแจในกรณีที่ใช้หลายเอเจนต์ร่วมกันกู้คืนกุญแจ คือ หากมีเอเจนต์ใดเอเจนต์หนึ่งขัดข้อง จะไม่สามารถให้บริการกู้คืนกุญแจได้ ทั้งนี้ผู้วิจัยจะได้ศึกษากระบวนการกู้คืนกุญแจเพื่อแก้ปัญหาดังกล่าว

2.12.7 Modeling of Multiple Agent Based Cryptographic Key Recovery Protocol

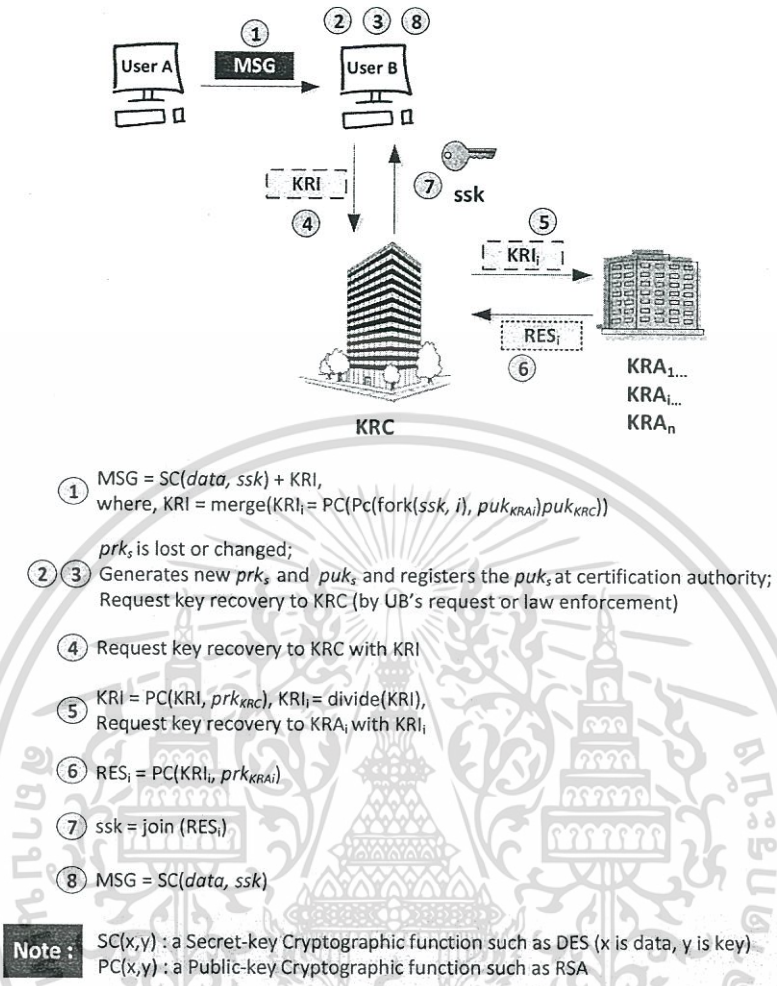
งานวิจัยนี้เน้นอธิบายโครงสร้างของสถาปัตยกรรมการทำงานของ KRS แบบ M-KRS [13, 14] ที่ผู้ใช้บริการกู้คืนกุญแจสามารถระบุจำนวนการขอใช้บริการ KRA ได้มากกว่าหนึ่งเอเจนต์ การกู้คืนกุญแจลับสามารถกู้คืนได้จากฟิลด์ข้อมูลกู้คืนกุญแจ ที่เรียกว่า เคอาร์ไอ (Key Recovery Information: KRI) โดยได้นำเสนอความต้องการพื้นฐานและสมมติฐานของระบบกู้คืนกุญแจที่น่าสนใจไว้ดังนี้ คือ การกู้คืนกุญแจจะกระทำสำเร็จได้ต้องอยู่ภายใต้การควบคุมของผู้ที่มีสิทธิ์หรือมีความมั่นคงภายใต้นโยบายหรือกฎหมาย การกู้คืนถูกจำกัดได้เฉพาะการกู้คืนกุญแจลับเท่านั้น (ไม่รวมถึงการกู้คืนข้อมูล) การกู้คืนกุญแจจะกระทำการอยู่บนโครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure: PKI) ดังนั้น ใบรับรองของแต่ละหน่วยงานกู้คืนกุญแจจะถูกกระจายหรือประกาศโดยผู้ให้บริการออกใบรับรอง (Certificate Authority: CA)

การกู้คืนกุญแจแบบใช้ KRA ได้มากกว่า 1 เอเจนต์หรือที่เรียกว่า M-KRS นั้น ได้แบ่งกระบวนการทำงานหลัก ๆ ออกเป็น 3 ขั้นตอน ดังแสดงตามรูปที่ 2.11 คือ

ขั้นตอนที่ 1 เริ่มต้นระบบ (Initialization) กุญแจสาธารณะของทุก ๆ KRA ถูกกระจายและประกาศ บนโครงสร้างพื้นฐานกุญแจสาธารณะ หลังจากที่ CA ออกใบรับรองให้กับ KRA แล้ว ในใบรับรองจะให้การรับรองกุญแจสาธารณะและข้อมูลของ KRA เพื่อการระบุตัวตน และไว้สำหรับการพิสูจน์ตัวจริง

ขั้นตอนที่ 2 การสื่อสารและการสร้างข้อมูลกู้คืนกุญแจ (Communication and Key Recovery Information Generation) ผู้ส่งทำการสุ่มเลือก KRA ว่าจะใช้กี่เอเจนต์ในการกู้คืนกุญแจ และทำการสร้าง KRI โดยใช้กุญแจสาธารณะของ KRC กับกุญแจสาธารณะของ KRA ในการเข้ารหัสลับ และทำการเลือก KRA ที่จะใช้บริการ ขั้นตอนนี้จะใช้กุญแจลับในการเข้ารหัสลับข้อมูลต้นฉบับ จากนั้นจะแนบข้อมูลต้นฉบับไปกับ KRI ส่งให้ผู้รับ

ขั้นตอนที่ 3 เมื่อผู้รับไม่สามารถทำการถอดรหัสลับได้ เนื่องจากกุญแจที่ใช้ในการถอดรหัสลับไม่สามารถใช้งานได้หรือสูญหาย หรือหน่วยงานที่มีอำนาจต้องการตรวจสอบ (กู้คืน) ข้อมูลจะต้องทำการสร้างกุญแจคู่ใหม่ และลงทะเบียนกุญแจสาธารณะไว้กับ CA เพื่อขอใบรับรองกุญแจสาธารณะ เมื่อได้รับใบรับรองแล้วจึงร้องขอการกู้คืนกุญแจต่อไป ด้วยการส่งใบรับรองของผู้ร้องขอความต้องการและข้อมูลในการกู้คืนกุญแจไปยัง KRC จากนั้นจะเป็นกระบวนการถอดรหัสลับข้อมูลตามลำดับ จนกระทั่งได้กุญแจลับเพื่อนำไปถอดรหัสลับข้อมูล



รูปที่ 2.11 กระบวนการทำงานของ Multiple Agent Based Cryptographic Key Recovery Protocol

อย่างไรก็ตามงานวิจัยนี้ได้เสนอระบบ M-KRS แบบอาศัย KRC โดยใช้เทคนิคการแชร์ความลับ (Secret Sharing) ในการแบ่งและจัดสรรส่วนประกอบของกุญแจลับไปไว้ใน KRI ของทุก KRA ที่อยู่ในกลุ่มการกู้คืนกุญแจ ซึ่งวิธีการดังกล่าวเป็นวิธีการที่ไม่ซับซ้อน ทำให้เกิดความรวดเร็วต่อกระบวนการกู้คืนกุญแจ ทั้งนี้สามารถอธิบายวิธีการแบ่งและจัดสรรกุญแจ และการรวมกุญแจได้ดังต่อไปนี้

การแบ่งและจัดสรรกุญแจ โดยผู้ส่ง

1) สุ่มตัวเลข (rk) จำนวน n-1 ตัว สำหรับ n เอเจนต์ (*Agent*) เช่น $rk_1, rk_2, ..., rk_{n-1}$, สำหรับ $Agent_1, Agent_2, ..., Agent_n$

2) รวมส่วนประกอบของกุญแจ ด้วยการใช้ XOR เพื่อนำค่าดังกล่าวไปเก็บไว้ใน KRF ดังนี้

ค่าใน KRF สำหรับ $Agent_1$ (ik_1) : ssk $\oplus$ rk_1

ค่าใน KRF สำหรับ $Agent_2$ (ik_2) : $rk_1 \oplus rk_2$

ค่าใน KRF สำหรับ $Agent_i$ (ik_i) : $rk_{i-1} \oplus rk_i$

ค่าใน KRF สำหรับ $Agent_n$ (ik_n) : rk_{n-1}

การรวมกุญแจ โดย KRC

1) นำค่าใน KRF มาทำการคำนวณกุญแจลับ ดังนี้

join (ik_1, ... ik_i, ... ik_n) = $ik_1 \oplus ik_2 \oplus$...., $\oplus$ ik_i ... $\oplus$...$\oplus$ ik_n

ดังนั้นกุญแจลับ (ssk) = (ssk $\oplus$ rk_1) $\oplus$ ($rk_1 \oplus rk_2$) $\oplus$..., ($rk_{i-1} \oplus rk_i$),...($rk_{n-2} \oplus rk_{n-1}$) $\oplus$ rk_{n-1}

โดยที่ ik_i ถูกเข้ารหัสลับที่ผู้ส่ง ด้วยกุญแจสาธารณะของ KRA และถอดรหัสลับที่ KRA ที่ i (KRA_i) ด้วยกุญแจส่วนตัวของ KRA_i

ทั้งนี้ระบบจะสามารถกู้คืนกุญแจได้อย่างถูกต้อง ในกรณีที่ KRA ทำงานครบทุกเอเจนต์ หากมี KRA ใด ๆ ที่ขัดข้อง คือไม่สามารถให้บริการกู้คืนกุญแจได้ ระบบจะไม่สามารถกู้คืนกุญแจได้อย่างถูกต้อง

จากการศึกษางานวิจัยนี้ ผู้วิจัยได้นำหลักการพื้นฐานของกระบวนการกู้คืนกุญแจมาใช้ประกอบการออกแบบกระบวนการกู้คืนกุญแจ และยังได้นำแนวคิดเรื่องการแบ่งและจัดสรรส่วนประกอบของกุญแจไปยัง KRA ที่อยู่ในกลุ่มการกู้คืนมาประยุกต์ใช้ในตอนต้นของกระบวนการกู้คืนกุญแจด้วย

2.13 สรุปการศึกษางานวิจัยที่เกี่ยวข้องกับระบบการกู้คืนกุญแจลับ

2.13.1 รูปแบบของ KRA

การจำแนกรูปแบบของ KRA ตามวิธีการจัดเก็บและการเข้าถึงกุญแจลับ เพื่อการกู้คืนสามารถจำแนกได้ 4 รูปแบบ ดังนี้

รูปแบบที่ 1 KRA ที่ทำหน้าที่จัดเก็บกุญแจลับ และสามารถเข้าถึงข้อมูลกุญแจลับนั้นเพื่อการกู้คืนกุญแจได้โดยตรง

ข้อดี เป็นวิธีการที่ไม่ซับซ้อน ง่ายต่อการกู้คืนกุญแจ

ข้อด้อย กุญแจมีความมั่นคงปลอดภัยต่ำ และผู้ใช้งานไม่มีความเป็นส่วนตัว รวมทั้งสิ้นเปลืองเนื้อที่สำหรับการจัดเก็บกุญแจลับ

รูปแบบที่ 2 KRA ที่ทำหน้าที่จัดเก็บกุญแจลับ แต่ลักษณะการจัดเก็บกุญแจจะเป็นการจัดเก็บแบบแยกส่วนกัน โดยเมื่อต้องการกู้คืนกุญแจลับ KRA จะดำเนินการรวบรวมกุญแจที่ถูกแยกส่วน เพื่อนำมารวมกันให้ได้เป็นกุญแจลับ

ข้อดี กุญแจมีความมั่นคงปลอดภัยมากขึ้นกว่าแบบที่ 1 เนื่องจากการเข้าถึงกุญแจทำได้ยากกว่า

ข้อด้อย KRA มีภาระเพิ่มมากขึ้น และผู้ใช้งานไม่มีความเป็นส่วนตัว รวมทั้งสิ้นเปลืองเนื้อที่สำหรับการจัดเก็บกุญแจลับ

รูปแบบที่ 3 KRA ไม่ได้ทำหน้าที่จัดเก็บกุญแจลับ โดยกุญแจลับจะถูกจัดเก็บไว้ในฟิลด์ในการกู้คืนกุญแจ (KRF) ซึ่งจะถูกแนบไปพร้อมกับข้อมูลต้นฉบับ และ KRA จะให้กุญแจสาธารณะไปเข้ารหัส KRF เมื่อต้องการกู้คืนกุญแจลับ KRA จะใช้กุญแจส่วนตัวถอดรหัส KRF วิธีนี้ได้รับการยอมรับจากผู้ใช้งาน

ข้อดี ผู้ใช้งานมีความเป็นส่วนตัวเพิ่มมากขึ้นกว่ารูปแบบที่ 1 และรูปแบบที่ 2

ข้อด้อย ระบบมีความซับซ้อนเพิ่มมากขึ้นกว่ารูปแบบที่ 1 และรูปแบบที่ 2

รูปแบบที่ 4 ใช้ KRA มากกว่า 1 เอเจนต์หรือใช้ KRA อย่างน้อย 2 เอเจนต์ขึ้นไป ในการกู้คืนกุญแจ ($n \geq 2$) โดย KRA ไม่ได้ทำหน้าที่จัดเก็บกุญแจลับ ในรูปแบบนี้กุญแจลับจะถูกแยกส่วนและจัดเก็บไว้ใน KRF โดย KRA จะให้กุญแจสาธารณะไปเข้ารหัส KRF เมื่อต้องการกู้คืนกุญแจลับ KRA จะใช้กุญแจส่วนตัวถอดรหัส KRF ส่วนการรวบรวมส่วนประกอบของกุญแจลับ อาจเป็นหน้าที่ของ KRA ที่ได้รับมอบหมายในกลุ่ม KRA หรือ KRC หรือผู้รับ

ข้อดี ระบบมีความมั่นคงเพิ่มมากขึ้น เนื่องจากการใช้ KRA มากกว่า 1 เอเจนต์ จะลดความเสี่ยงในด้าน SPOF และผู้ใช้งานมีความเป็นส่วนตัวสูง เนื่องจากมีการกระจายความลับของกุญแจลับไปยัง KRA มากกว่า 1 เอเจนต์

ข้อด้อย การทำงานมีความซับซ้อนมากขึ้น ต้องใช้เวลาในการประมวลผลเพิ่มขึ้น

ในงานวิจัยนี้ได้ใช้รูปแบบที่ 4 ในการออกแบบระบบการกู้คืนกุญแจ เนื่องจากเป็นรูปแบบที่เน้นเรื่องความมั่นคงปลอดภัยเป็นหลัก เพราะในปัจจุบันได้มีภัยคุกคามเกิดขึ้นหลากหลายรูปแบบ ส่วนประเด็นเรื่องความซับซ้อนของระบบที่มีผลต่อเวลาที่ใช้ในการประมวลผลนั้น ไม่ถือว่าเป็นข้อปัญหาเนื่องจากความก้าวหน้าของเทคโนโลยีคอมพิวเตอร์ จะช่วยเรื่องการประมวลผลงานที่มีความซับซ้อนให้ทำได้อย่างรวดเร็ว

2.13.2 สรุปผลการศึกษา

จากการศึกษางานวิจัยที่เกี่ยวข้องกับระบบการกู้คืนกุญแจลับ จะเห็นว่าการออกแบบระบบการกู้คืนกุญแจสามารถทำได้หลายรูปแบบ แต่ละรูปแบบได้นำเสนอเทคนิคและกระบวนการที่มีจุดเด่นในการพัฒนาแตกต่างกันออกไป ส่วนใหญ่จะเน้นในเรื่องการรักษาความลับของข้อมูล ความมั่นคงของกุญแจลับ และความเป็นส่วนตัวของผู้ใช้งาน โดยวัตถุประสงค์หลักของระบบการกู้คืนกุญแจมี 2 ประการ คือ (1) เพื่อให้บริการกู้คืนกุญแจในกรณีที่กุญแจชำรุดเสียหาย หรือกุญแจไม่

สามารถใช้งานได้ และ(2) เพื่อสนับสนุนภารกิจของรัฐบาลหรือหน่วยงานที่รัฐบาลมอบหมายในการตรวจสอบข้อมูลที่ต้องสงสัยที่มีการส่งผ่านระบบเครือข่าย

บุคคลที่มีสิทธิ์ในการร้องขอการกู้คืนกุญแจลับ คือผู้รับซึ่งเป็นผู้ที่มีสิทธิ์ในกุญแจที่ต้องการกู้คืน หรือหน่วยงาน/บุคคลของรัฐบาลที่มีอำนาจสิทธิ์ในการตรวจสอบข้อมูลต้องสงสัยที่ส่งผ่านระบบเครือข่าย และต้องการเข้าถึงข้อมูลต้นฉบับและกุญแจลับซึ่งจะต้องอยู่ภายใต้นโยบายหรือกฎหมายที่แน่นอน

ในปัจจุบันงานทางด้านระบบการกู้คืนกุญแจลับ ได้เปลี่ยนรูปแบบจากงานวิจัยเพื่อการตีพิมพ์เผยแพร่ไปเป็นการทำวิจัยเพื่อการค้าและมีการจดสิทธิบัตรงานดังกล่าว ผลิตภัณฑ์ต่าง ๆ ในท้องตลาดก็ได้มีการนำกระบวนการหรือฟังก์ชันของการกู้คืนกุญแจมาใช้จำนวนมากขึ้นตามลำดับ

บทที่ 3
ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจาย

ระบบการกู้คืนกุญแจลับ (KRS) มีวัตถุประสงค์เพื่อให้บริการกู้คืนกุญแจในกรณีที่กุญแจชำรุดเสียหายหรือไม่สามารถใช้งานได้ และสนับสนุนภารกิจของรัฐหรือหน่วยงานที่รัฐบาลมอบหมายในการตรวจสอบข้อมูลที่ต้องสงสัยที่ส่งผ่านระบบเครือข่าย ดังนั้นบุคคลที่สามารถร้องขอการกู้คืนกุญแจลับหรือมีสิทธิ์ในกุญแจลับนั้น คือผู้รับซึ่งเป็นเจ้าของกุญแจที่ต้องการกู้คืนหรือหน่วยงานของรัฐที่มีอำนาจตามกฎหมายในการเข้าถึงข้อมูลและกุญแจลับเพื่อตรวจสอบข้อมูลต้องสงสัยนั้น

งานวิจัยนี้นำเสนอระบบการกู้คืนกุญแจลับแบบหลายเอเจนต์ (M-KRS) สองรูปแบบ คือ (1) ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่มีความพร้อมใช้งานสูง (High-Availability Decentralized Multiple-Agent Key Recovery System: HADM-KRS) และ (2) ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่มีความพร้อมใช้งานสูงอย่างง่าย (Simple High-Availability Decentralized Multiple-Agent Key Recovery System: SHADM-KRS)

โดยทั้งสองรูปแบบมีจุดเด่นคือ (1) ระบบสามารถทำงานโดยไม่ใช้ศูนย์กลางการกู้คืนกุญแจ (Key Recovery Center: KRC) แต่เป็นการออกแบบให้ใช้เฉพาะกลุ่มของเอเจนต์ในการกู้คืนกุญแจ (Key Recovery Agent: KRA) โดยแต่ละเอเจนต์ทำการกู้คืนส่วนประกอบของกุญแจโดยอิสระจากกัน และ (2) ระบบมีความพร้อมใช้งานสูง กล่าวคือระบบสามารถทำงานได้แม้ในกรณีที่มีบางเอเจนต์ในกลุ่มการกู้คืนล่ม โดยเมื่อมีบางเอเจนต์ในกลุ่มการกู้คืนล่ม เอเจนต์อื่นที่อยู่ในกลุ่มการกู้คืนเดียวกันสามารถทำงานแทนกันได้ จุดเด่นทั้งสองประการนี้จะช่วยลดโอกาสการเกิดความล้มเหลวในการกู้คืนกุญแจ ทำให้ระบบมีความพร้อมใช้งานสูง

ความแตกต่างของระบบ HADM-KRS และSHADM-KRS คือความสามารถในการบริหารจัดการจำนวนเอเจนต์ที่ใช้ในการกู้คืนกุญแจ Ks กล่าวคือ ระบบ HADM-KRS สามารถกำหนดจำนวนเอเจนต์ขั้นต่ำที่ใช้ในการกู้คืนกุญแจได้ เช่น เมื่อใช้เอเจนต์จำนวน 6 เอเจนต์ในการกู้คืนกุญแจ ระบบ HADM-KRS สามารถกำหนดได้ว่าจะให้มีเอเจนต์ล่มได้อย่างมากที่สุดกี่เอเจนต์ หรืออย่างน้อยต้องเหลือเอเจนต์ที่สามารถให้บริการได้กี่เอเจนต์ จึงจะให้บริการกู้คืนกุญแจ Ks ได้สำเร็จ ส่วนระบบ SHADM-KRS ไม่มีฟังก์ชันนี้ แต่สามารถกู้คืนกุญแจได้เมื่อมีบางเอเจนต์ล่ม

ทั้งนี้ผู้วิจัยได้ออกแบบกระบวนการแชร์ความลับที่สามารถนำมาใช้ในขั้นตอนการแบ่งและจัดสรรส่วนประกอบของกุญแจ Ks ได้อย่างดีและมีประสิทธิภาพ โดยใช้พื้นฐานของทฤษฎีการแชร์ความลับ

3.1 การแบ่งและจัดสรรส่วนประกอบของกุญแจ *Ks*

การออกแบบกระบวนการแชร์ความลับ เพื่อนำมาใช้ในขั้นตอนการแบ่งและจัดสรรส่วนประกอบของกุญแจ *Ks* ทำให้การจัดเก็บค่าแอตทริบิวต์ในฟิลด์ *KRF* เป็นไปอย่างเหมาะสม ขั้นตอนการแชร์ความลับของกุญแจ *Ks* และการถอดความลับ สามารถแสดงได้ดังขั้นตอนในตารางที่ 3.1

ตารางที่ 3.1 ขั้นตอนการแชร์ความลับและการถอดความลับของกุญแจ *Ks*

การแชร์ความลับของ *Ks*	
❶ สุ่มตัวเลขจำนวน n-1 ตัว สำหรับ n-1 เอเจนต์	$S_1, S_2, ..., S_{n-1}$
❷ คำนวณ S_n สำหรับ $Agent_n$	$S_n = S_1 \oplus S_2 \oplus ... \oplus S_{n-1} \oplus Ks$
❸ สุ่มตัวเลข R สำหรับทุก ๆ เอเจนต์ $(R_i's)$	$R_1, R_2, ..., R_n$
❹ คำนวณหาค่าความลับ *SGN*	$SGN = R_1 \oplus R_2 \oplus ... \oplus R_n$
❺ คำนวณค่า *TT* สำหรับทุก ๆ เอเจนต์ $(TT_i's)$	$TT_i = S_i \oplus SGN$; เมื่อ $i = 1$ ถึง n
การถอดความลับของ *Ks*	S_i คือ ส่วนประกอบของกุญแจ *Ks*
แอตทริบิวต์สำหรับใช้ในการกู้คืนกุญแจ *Ks*	{ $Ks = S_1 \oplus S_2 \oplus ... \oplus S_n$ }
{ S_i, TT_i, *SGN* }	TT_i ใช้กู้คืน S_i เมื่อมีบางเอเจนต์ล่ม
	{ $TT_i = S_i \oplus SGN$ }
	SGN ใช้พิสูจน์ตัวจริงของเอเจนต์ในกลุ่มการกู้คืน

จากตารางที่ 3.1 จะได้นำไปใช้ในขั้นตอนการสร้างฟิลด์ *KRF* ของระบบ HADM-KRS และ SHADM-KRS ซึ่งจะมีการกล่าวถึงรายละเอียดของการนำไปใช้ในหัวข้อถัดไป

3.2 ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่มีความพร้อมใช้งานสูง (HADM-KRS)

HADM-KRS เป็นระบบการกู้คืนกุญแจที่มีกระบวนการทำงานเพื่อให้กุญแจลับมีความมั่นคงปลอดภัย ผู้ใช้งานมีความเป็นส่วนตัว และสามารถบริหารจัดการจำนวนเอเจนต์ในกลุ่มการกู้คืนให้สอดคล้องกับระดับของความมั่นคงปลอดภัยที่ต้องการได้ โดยระบบมีการทำงานดังต่อไปนี้

3.2.1 ผู้ที่มีส่วนร่วมในระบบ HADM-KRS

ระบบ HADM-KRS ประกอบด้วยผู้ที่มีส่วนร่วมในระบบดังแสดงตามรูปที่ 3.1 ดังนี้

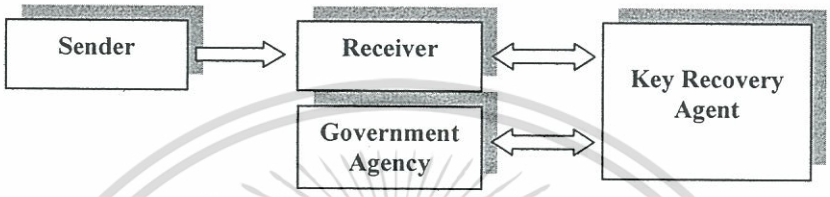
1) ผู้ส่ง (Sender)

2) ผู้รับ (Receiver)

3) หน่วยงานของรัฐ (Government Agency)

4) หน่วยงานหรือเอเจนต์ในการกู้คืนกุญแจ (KRA)

ตอนเริ่มต้นระบบทุกส่วนจะต้องมีกุญแจคู่ของตัวเอง คือ กุญแจส่วนตัว (*Kr*) และกุญแจสาธารณะ (*Ku*) ที่มีใบรับรองจากหน่วยงานออกใบรับรอง (Certificate Authority: CA) ระบบทำงานภายใต้โครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure: PKI)



รูปที่ 3.1 ผู้ที่มีส่วนร่วมในระบบ HADM-KRS

3.2.2 หน้าที่ของผู้ที่มีส่วนร่วมในระบบ HADM-KRS

หน้าที่ของผู้ที่มีส่วนร่วมในระบบสามารถแสดงได้ดังรูปที่ 3.2 ดังนี้

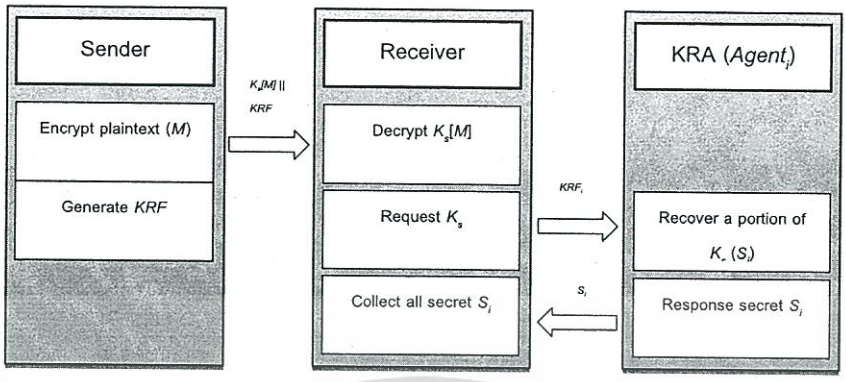
1) ผู้ส่ง (Sender) ทำหน้าที่สร้างฟิลด์ที่ใช้ในการกู้คืนกุญแจ (*KRF*) และแนบฟิลด์ *KRF* เข้ากับข้อความต้นฉบับ ส่งไปยังผู้รับทุกครั้งที่มีการส่งข้อความต้นฉบับ

2) ผู้รับ (Receiver) ทำหน้าที่ร้องขอการกู้คืนส่วนประกอบของกุญแจไปยัง KRA เมื่อกุญแจลับสูญหาย หรือไม่สามารถใช้ถอดรหัสข้อความต้นฉบับได้ และทำหน้าที่คำนวณข้อมูลกุญแจลับเมื่อได้รับข้อมูลส่วนประกอบของกุญแจครบถ้วนจาก KRA

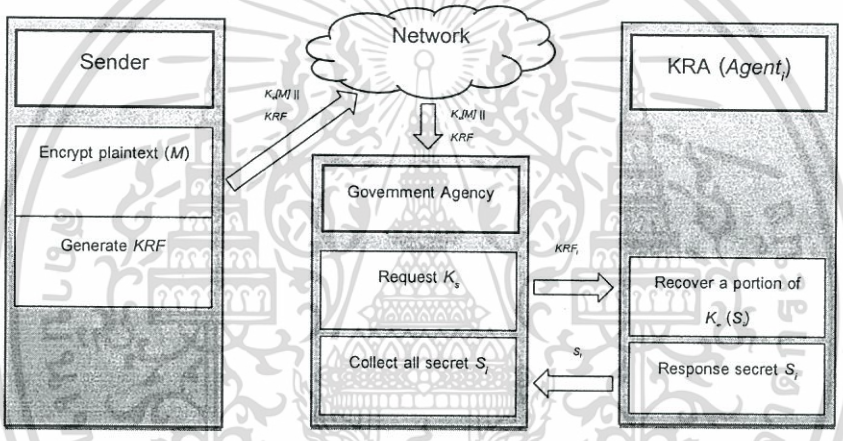
3) หน่วยงานของรัฐ (Government Agency) ทำการร้องขอการกู้คืนส่วนประกอบของกุญแจลับจาก KRA และคำนวณข้อมูลกุญแจลับเมื่อได้รับข้อมูลส่วนประกอบของกุญแจครบถ้วนจาก KRA และนำกุญแจไปใช้ถอดรหัสข้อมูลที่ต้องสงสัย เพื่อการตรวจสอบข้อมูลตามกฎหมาย

4) KRA ทำหน้าที่กู้คืนส่วนประกอบของกุญแจลับเมื่อได้รับการร้องขอการกู้คืนส่วนประกอบของกุญแจจากผู้ร้องขอ (ผู้รับหรือหน่วยงานของรัฐ) และส่งส่วนประกอบของกุญแจไปให้ผู้ร้องขอ

ทั้งนี้การส่ง $KRF_i's$ ไปยังแต่ละ KRA จะอาศัยข้อมูลจากใบรับรองกุญแจสาธารณะของ KRA นั้นๆ เพื่อบอกตำแหน่งที่อยู่ของ KRA และจะมีกระบวนการพิสูจน์ตัวตนระหว่างผู้ร้องขอและ KRA ทุกครั้งโดยใช้ข้อมูล other information ใน $KRF_i's$ ซึ่งกระบวนการที่ทำให้เกิดความมั่นคงระหว่างคู่สื่อสารจะกระทำบนพื้นฐานของ PKI



(1) หน้าที่ของ ผู้ส่ง ผู้รับ และ KRA



(2) หน้าที่ของ ผู้ส่ง หน่วยงานของรัฐ และ KRA

รูปที่ 3.2 หน้าที่ของผู้ที่มีส่วนร่วมในระบบ HADM-KRS

3.2.3 กระบวนการทำงานของระบบ HADM-KRS

กระบวนการทำงานในช่วงเริ่มต้นของระบบ ผู้ส่งร้องขอกุญแจลับ (*Ks*) จากหน่วยงานให้บริการกุญแจ เช่น KDC [38] เป็นต้น เพื่อใช้ในการเข้ารหัสข้อมูลต้นฉบับ (M) ที่ต้องการส่งไปให้ผู้รับ และจะทำการแชร์กุญแจลับกันระหว่างผู้รับกับผู้ส่ง

กระบวนการทำงานหลักของระบบ HADM-KRS แบ่งออกเป็นสองขั้นตอน คือ (1) การสร้างฟิลด์ *KRF* โดยผู้ส่ง และ (2) การกู้คืนส่วนประกอบของกุญแจ *Ks* และการกู้คืนกุญแจ *Ks* โดย KRA และผู้รับหรือหน่วยงานของรัฐ ดังมีรายละเอียดดังต่อไปนี้

3.2.3.1 การสร้างฟิลด์ในการกู้คืนกุญแจ (*KRF*)

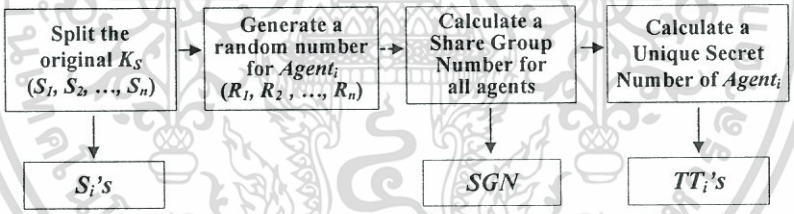
การสร้างฟิลด์ *KRF* จะกระทำโดยผู้ส่ง ซึ่งฟิลด์ *KRF* จะถูกแนบไปกับข้อมูลต้นฉบับ (*M*) ที่ผ่านการเข้ารหัสลับด้วยกุญแจลับ (*Ks*[*M*]) เรียบร้อยแล้วไปยังผู้รับ หากผู้รับต้องการร้องขอการกู้คืนกุญแจ หรือหน่วยงานของรัฐต้องการตรวจสอบข้อมูล จะส่งคำร้องขอการกู้คืนกุญแจพร้อมกับฟิลด์ *KRF* ไปยัง KRA

ฟิลด์ *KRF* ประกอบด้วยข้อมูลต่อไปนี้ (1) ส่วนประกอบของกุญแจ *Ks* (2) แอตทริบิวต์สำหรับการระบุตัวตนของ KRA ในกลุ่มการกู้คืน (3) แอตทริบิวต์สำหรับการกู้คืนส่วนประกอบของกุญแจ *Ks* ในกรณี KRA ที่อยู่ในกลุ่มการกู้คืนล่ม และ(4) ข้อมูลที่จำเป็นสำหรับการระบุตัวตนของผู้ที่มีส่วนร่วมในระบบ

การสร้างฟิลด์ *KRF* แบ่งออกเป็นส่วนย่อย ๆ ได้สองขั้นตอน คือ (1) การสร้างส่วนประกอบของฟิลด์ *KRF* และ(2) การประกอบฟิลด์ *KRF*

1) การสร้างส่วนประกอบของฟิลด์ *KRF*

ฟิลด์ *KRF* ถูกสร้างขึ้นเพื่อใช้ในการกู้คืนส่วนประกอบของกุญแจลับ (S_i) และกุญแจลับ *Ks* ขั้นตอนการสร้างส่วนประกอบของฟิลด์ *KRF* สามารถอธิบายได้ดังรูปที่ 3.3 ซึ่งมีกระบวนการดังต่อไปนี้



รูปที่ 3.3 กระบวนการสร้างส่วนประกอบของฟิลด์ *KRF*

1.1) แบ่ง *Ks* ออกเป็น *n* ชิ้น เมื่อ *n* คือ จำนวน KRA ทั้งหมดที่ใช้ในการกู้คืนกุญแจ โดยได้ใช้แนวคิดพื้นฐานเรื่องการแชร์ความลับ (Secret Sharing) [28]

กระบวนการแบ่งกุญแจลับสามารถทำได้ดังต่อไปนี้

1.1.1) สุ่มตัวเลข จำนวน *n*-1 ตัว สำหรับ *n*-1 เอเจนต์ (*Agent*) เช่น $S_1, S_2, \ldots, S_{n-1}$ สำหรับ $Agent_1, Agent_2, \ldots, Agent_{n-1}$ ตามลำดับ

1.1.2) คำนวณ S_n สำหรับ $Agent_n$ ด้วยการ XOR ค่า S_i และกุญแจ *Ks*

$$S_n = S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus Ks$$

1.2) สุ่มตัวเลข *R* สำหรับทุก ๆ เอเจนต์ (R_i's) เพื่อนำไปคำนวณหาค่าความลับของกลุ่มการกู้คืนกุญแจ (*SGN*) สำหรับการยืนยันตัวตนของ KRA

1.3) คำนวณหาค่าความลับ *SGN* ด้วยการ XOR ค่าของแอตทริบิวต์ R_i ทั้งหมด

$$SGN = R_1 \oplus R_2 \oplus ... \oplus R_n$$

1.4) คำนวณหาค่าของแอตทริบิวต์พิเศษ (*TT*) สำหรับทุก ๆ เอเจนต์ (TT_i's) เพื่อใช้ในการกู้คืน S_i ในกรณีที่มีบาง KRA ในกลุ่มการกู้คืนล่ม โดยสามารถคำนวณได้จาก

$$TT_i = S_i \oplus SGN$$

2) การประกอบฟิลด์ *KRF*

ฟิลด์ *KRF* ของ HADM-KRS ประกอบด้วยฟิลด์ *KRF* ย่อย ๆ (KRF_i's) ของเอเจนต์ในกลุ่มการกู้คืนกุญแจ *(KRA)*

ฟิลด์ KRF_i ประกอบด้วย S_i, *SGN*, TT_i และ *other information* แต่ละส่วนประกอบมีความสำคัญดังนี้

S_i คือส่วนประกอบของกุญแจ *Ks*

SGN เป็นค่าสำหรับการยืนยันตัวตนของ KRA ที่อยู่ในกลุ่มการกู้คืน

TT_i เป็นค่าสำหรับการกู้คืน S_i ในกรณี KRA ที่อยู่ในกลุ่มการกู้คืนล่ม

other information เก็บค่าอื่น ๆ ที่จำเป็นสำหรับการพิสูจน์ตัวจริงหรือการตรวจสอบเพื่อเพิ่มความมั่นคงปลอดภัยให้กับระบบ เช่น ใบรับรองกุญแจสาธารณะของเอเจนต์ และของผู้ที่มีสิทธิ์ในการร้องขอการกู้คืนกุญแจ เป็นต้น

โครงสร้างฟิลด์ *KRF* ของ HADM-KRS

ฟิลด์ *KRF* ของ HADM-KRS ประกอบด้วยฟิลด์ KRF_i จำนวน *n* ชิ้น ดังนี้

$$KRF = \{Ku_{agi}[\ KRF_i\text{'s}\ ]\}$$

ฟิลด์ KRF_i ประกอบด้วย S_i, *SGN*, TT_i และ *other information* ดังนี้

$$KRF_i = Ku_{agi}[\ S_i \,||\, SGN \,||\, TT_i\text{'s} \,||\, other\ information\ ]$$

โดยที่ฟิลด์ KRF_i จะถูกเข้ารหัสด้วยกุญแจสาธารณะของ KRA (Ku_{agi}) ดังนั้น KRA_i เท่านั้นที่สามารถถอดรหัส KRF_i ได้ โดย KRA_i จะถอดรหัสฟิลด์ KRF_i เพื่อส่ง S_i และ *SGN* ไปยังผู้ร้องขอการกู้คืนส่วนประกอบของกุญแจ เพื่อจะได้ทำการคำนวณกุญแจลับ *Ks* ต่อไป

HADM-KRS ถูกออกแบบให้ระบบมีความพร้อมใช้งานสูง กล่าวคือระบบสามารถทำงานได้แม้ในกรณีที่มีบาง KRA ล่ม โดยใช้ค่าของแอตทริบิวต์ TT_i ซึ่งจัดเก็บในฟิลด์ KRF_i สำหรับคำนวณ S_i ของ KRA_i ที่ล่ม

การจัดสรรค่าของแอตทริบิวต์ TT_i ไว้ในฟิลด์ KRF_i จะใช้ทฤษฎีเพาเวอร์เซต (Power Set) [29] โดยมีขั้นตอนการแบ่งและจัดสรรดังนี้

1) ระบุจำนวน KRA ที่จะใช้ในการกู้คืนกุญแจ (n)

2) ระบุจำนวน KRA ขั้นต่ำสำหรับการกู้คืนกุญแจ (mr) โดยที่ $mr \geq 2$

3) คำนวณหาจำนวนของ KRA สำหรับการกระจายค่า TT_i (t) ว่าจะกระจายค่า TT_i ไปให้กับ KRA กี่เอเจนต์ ได้จากสูตร

$$t = n - mr \text{ เมื่อ } t \leq mr$$

4) กระจายค่า TT_i ของ KRA_i (A_i) ไปยังเอเจนต์ที่อยู่ถัดไปในลักษณะของการหมุนตามเข็มนาฬิกา จำนวน t เอเจนต์ เมื่อ i=1 ถึง n ตามฟังก์ชันต่อไปนี้

$$A_i \rightarrow \begin{cases} A_{i+1}, A_{i+2}, \ldots, A_{i+t} & \text{where } i<mr \text{ and } i=mr \\ A_{i+1}, A_{i+2}, \ldots, A_m, A_1, A_2, \ldots, A_j & \text{where } i>mr \text{ and } j=i-mr \\ A_1, A_2, \ldots, A_t & \text{where } i=n \end{cases}$$

ทั้งนี้ KRA จะใช้ค่า TT_i สำหรับการกู้คืน S_i ได้ไม่เกิน t-1

ตัวอย่าง การกระจายและการจัดเก็บค่าของแอตทริบิวต์ TT_i ในฟิลด์ KRF_i

สมมุติให้ใช้ KRA สำหรับการกู้คืนกุญแจจำนวน 6 เอเจนต์ (n = 6) และให้ใช้ KRA ในการกู้คืนขั้นต่ำจำนวน 3 เอเจนต์ (mr = 3)

ดังนั้น สามารถคำนวณหาการกระจายค่าของแอตทริบิวต์ TT_i ไปยัง KRA ถัดไปในลักษณะของการหมุนตามเข็มนาฬิกา ดังนี้

แทนค่าเพื่อคำนวณตัวเลขสำหรับการกระจายค่าของแอตทริบิวต์ TT_i ว่าต้องทำการกระจายค่าของแอตทริบิวต์ TT_i ไปยัง KRA กี่เอเจนต์ ($t = n - mr$) ดังนี้

t = 6 - 3 ดังนั้น t = 3

ฉะนั้นแสดงว่าต้องกระจาย TT_i ไปยัง KRA ถัดไปในลักษณะของการหมุนตามเข็มนาฬิกา จำนวน 3 เอเจนต์ โดยสามารถแสดงการจัดเก็บค่าของแอตทริบิวต์ TT_i ในฟิลด์ KRF_i ได้ดังตารางที่ 3.2

ตารางที่ 3.2 การกระจายและการสำรองค่า TT_i ในฟิลด์ KRF_i ของ KRA

ค่าเริ่มต้น		การสำรองค่า TT_i ใน KRF_i ของ KRA					
KRF		TT_1	TT_2	TT_3	TT_4	TT_5	TT_6
KRA_1	$KRF_1 : TT_1$				TT_4	TT_5	TT_6
KRA_2	$KRF_2 : TT_2$	TT_1				TT_5	TT_6
KRA_3	$KRF_3 : TT_3$	TT_1	TT_2				TT_6
KRA_4	$KRF_4 : TT_4$	TT_1	TT_2	TT_3			
KRA_5	$KRF_5 : TT_5$		TT_2	TT_3	TT_4		
KRA_6	$KRF_6 : TT_6$			TT_3	TT_4	TT_5	

จากตารางที่ 3.2 สามารถสรุปว่าแต่ละฟิลด์ KRF_i บรรจุค่า TT_i ดังนี้

KRF_1 ของ KRA_1 บรรจุ TT_4, TT_5, TT_6

KRF_2 ของ KRA_2 บรรจุ TT_1, TT_5, TT_6

KRF_3 ของ KRA_3 บรรจุ TT_1, TT_2, TT_6

KRF_4 ของ KRA_4 บรรจุ TT_1, TT_2, TT_3

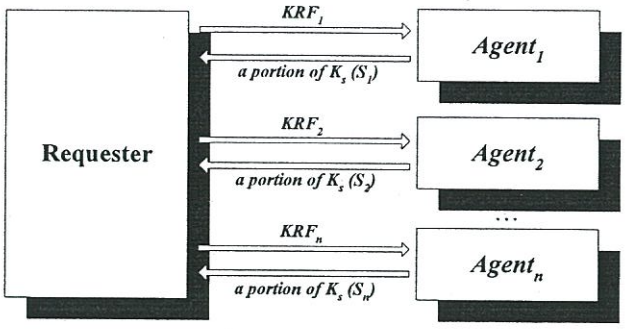
KRF_5 ของ KRA_5 บรรจุ TT_2, TT_3, TT_4

KRF_6 ของ KRA_6 บรรจุ TT_3, TT_4, TT_5

3.2.3.2 การกู้คืนกุญแจลับ *Ks*

เมื่อผู้รับได้รับ *Ks[M]* และฟิลด์ *KRF* จะทำการถอดรหัสข้อมูลด้วยกุญแจลับที่มีการแชร์กันในตอนต้นของการสื่อสาร

หากกุญแจ *Ks* ของผู้รับสูญหายหรือไม่สามารถใช้ในการถอดรหัสข้อมูลได้ ผู้รับจะทำการร้องขอการกู้คืนกุญแจ *Ks* ไปยัง KRA หรือหากหน่วยงานรัฐต้องการตรวจสอบข้อมูลที่ต้องสงสัยจะมีการร้องขอการกู้คืนกุญแจไปยัง KRA เช่นเดียวกัน ภาพรวมของการติดต่อสื่อสารระหว่างผู้ร้องขอการกู้คืนกุญแจ *Ks* กับ KRA สามารถแสดงได้ตามรูปที่ 3.4 และในขั้นตอนการกู้คืนกุญแจ *Ks* สามารถแบ่งออกเป็นส่วนย่อย ๆ ได้สองขั้นตอน คือ (1) การกู้คืนส่วนประกอบของกุญแจ *Ks* (S_i's) และ (2) การกู้คืนกุญแจ *Ks*



รูปที่ 3.4 การติดต่อสื่อสารระหว่างผู้ร้องขอการกู้คืนกุญแจ *Ks* กับ KRA ของ HADM-KRS

1) การกู้คืนส่วนประกอบของกุญแจ *Ks* (S_i's)

ในขั้นตอนนี้จะเป็นการกู้คืน S_i ซึ่งเป็นหน้าที่ของ KRA เมื่อ KRA ได้รับการร้องขอการกู้คืนกุญแจจากผู้ร้องขอ ซึ่งก็คือผู้รับหรือหน่วยงานของรัฐ

กระบวนการกู้คืน S_i แบ่งออกเป็นสองส่วน คือ (1) ส่วนของผู้ร้องขอการกู้คืนกุญแจ (Requester) และ(2) ส่วนของ KRA โดยมีรายละเอียดดังต่อไปนี้

1.1) ส่วนของผู้ร้องขอการกู้คืนกุญแจ *Ks*

1.1.1) ผู้ร้องขอการกู้คืนกุญแจถอดรหัสฟิลด์ *KRF* จะได้ฟิลด์ KRF_i's

1.1.2) ผู้ร้องขอการกู้คืนกุญแจส่งฟิลด์ KRF_i ให้กับ $Agent_i$ เมื่อ $i = 1$ ถึง n

1.2) ส่วนของ KRA

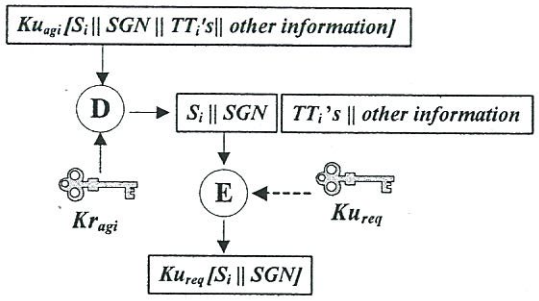
ขั้นตอนในการกู้คืน S_i สามารถแสดงได้ดังรูปที่ 3.5 โดยมีขั้นตอนดังต่อไปนี้

1.2.1) $Agent_i$ ถอดรหัสฟิลด์ KRF_i ด้วยกุญแจส่วนตัว (Kr_{agi}) จะได้ S_i, *SGN*, TT_i's, และ *other information*

1.2.2) $Agent_i$ ตรวจสอบ *other information* เช่น ใบรับรองกุญแจสาธารณะ (Public Key Certificate) ของผู้ร้องขอ เป็นต้น

1.2.3) $Agent_i$ เข้ารหัส S_i และ *SGN* ด้วยกุญแจสาธารณะของผู้ร้องขอ (Ku_{req})

1.2.4) $Agent_i$ ส่ง $Ku_{req}[S_i \,||\, SGN]$ ให้กับผู้ร้องขอ เพื่อให้ผู้ร้องขอนำไปคำนวณหากุญแจ *Ks* ต่อไป



รูปที่ 3.5 การกู้คืนส่วนประกอบของกุญแจ (S_i) โดย KRA ของ HADM-KRS

1.3) กรณีที่มีบาง KRA ล่ม

ในกรณีนี้ส่งผลให้ผู้ร้องขอการกู้คืนกุญแจ (ผู้รับ) รวบรวม $Ku_{req}[S_i]$ ได้ไม่ครบ ดังนั้นผู้ร้องขอจะต้องทำการร้องขอการกู้คืน S_i ที่หายไปหรือที่ยังไม่ได้รับกับ KRA ที่สามารถให้บริการได้ที่อยู่ในตำแหน่งถัดไป โดยกำหนดให้ KRA กู้คืน S_i ได้ไม่เกิน t-1 โดยมีกระบวนการดังต่อไปนี้

1.3.1) ผู้ร้องขอตรวจสอบ S_i ว่ายังไม่ได้รับมาจาก $Agent_i$ ใด

1.3.2) ผู้ร้องขอเข้ารหัสและส่งคำร้องขอการกู้คืนส่วนประกอบของกุญแจที่ยังไม่ได้รับ ($req\text{-}S_i$) และ SGN ด้วยกุญแจสาธารณะของเอเจนต์ถัดไป ($Agent_{nxt}$) ที่จะให้ทำการกู้คืนส่วนประกอบของกุญแจ (Ku_{nxt} of $Agent_{nxt}$) จะได้ $Ku_{nxt}[req\text{-}S_i \| SGN \| other\ information]$

1.3.3) $Agent_{nxt}$ ถอดรหัส $Ku_{nxt}[req\text{-}S_i \| SGN \| other\ information]$ ด้วยกุญแจส่วนตัว (Kr_{nxt})

1.3.4) $Agent_{nxt}$ ตรวจสอบ SGN พร้อมทั้งใบรับรองกุญแจสาธารณะของผู้ส่ง และคำนวณ S_i ดังนี้

$$S_i = TT_i \oplus SGN$$

1.3.5) $Agent_{nxt}$ เข้ารหัส S_i และ SGN ด้วย Ku_{req} จะได้ $Ku_{req}[S_i, SGN]$

1.3.6) $Agent_{nxt}$ ส่ง $Ku_{req}[S_i, SGN]$ ไปยังผู้ร้องขอ

ตัวอย่าง การร้องขอและการกู้คืน S_i ที่ยังไม่ได้รับจาก $Agent_i$

สมมุติให้ใช้ KRA สำหรับการกู้คืนกุญแจ 6 เอเจนต์ คือ $Agent_1$, $Agent_2$, $Agent_3$, $Agent_4$, $Agent_5$ และ $Agent_6$ ตามลำดับ และมี KRA ล่มไม่สามารถให้บริการกู้คืนกุญแจได้ 3 เอเจนต์ คือ $Agent_2$, $Agent_4$ และ $Agent_5$ ส่งผลให้ระบบสามารถใช้ KRA เพื่อการกู้คืนกุญแจลับได้เพียง 3 เอเจนต์ คือ $Agent_1$, $Agent_3$ และ $Agent_6$ ดังแสดงรายละเอียดดังตารางที่ 3.3

ตารางที่ 3.3 แสดงเอเจนต์ที่ใช้ในการกู้คืนกุญแจในกรณีมีเอเจนต์ล่ม และการสำรองค่า TT_i ในฟิลด์ KRF_i

KRF		TT_1	TT_2	TT_3	TT_4	TT_5	TT_6
KRA_1	$KRF_1 : TT_1$				TT_4	TT_5	TT_6
KRA_2	*Unavailable*						
KRA_3	$KRF_3 : TT_3$	TT_1	TT_2				TT_6
KRA_4	*Unavailable*	TT_1	TT_2	TT_3			
KRA_5	*Unavailable*		TT_2	TT_3	TT_4		
KRA_6	$KRF_6 : TT_6$			TT_3	TT_4	TT_5	

จากตารางที่ 3.3 สามารถสรุปว่า KRA ที่ล่มคือ $Agent_2$, $Agent_4$ และ $Agent_5$ ส่งผลให้ไม่สามารถหาค่า TT_2 , TT_4 และ TT_5 ได้ ดังนั้นจะต้องมีการร้องขอค่าของแอตทริบิวต์ TT_i ที่หายไปกับ KRA ที่อยู่ถัดไป และต้องเป็น KRA ที่ยังสามารถให้บริการได้ สามารถแสดงขั้นตอนได้ดังต่อไปนี้

1) ร้องขอค่าแอตทริบิวต์

TT_2 จาก KRA_3

TT_4 และ TT_5 จาก KRA_6

2) เมื่อได้ค่าของแอตทริบิวต์ TT_i's ครบจำนวนแล้ว จะสามารถนำ TT_i ไปคำนวณหา S_i และกุญแจ Ks ได้

2) การกู้คืนกุญแจ *Ks*

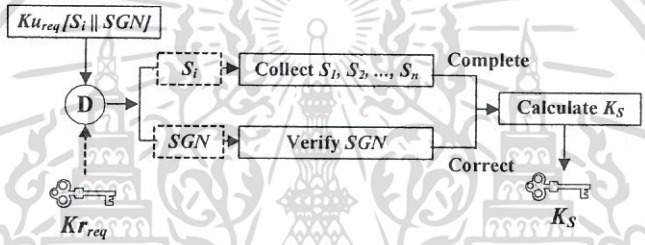
ส่วนนี้เป็นหน้าที่ของผู้รับที่จะทำการคำนวณเพื่อกู้คืนกุญแจ *Ks* ซึ่งวิธีดังกล่าวเป็นการช่วยลดภาระของ KRA และทำให้กุญแจ *Ks* มีความมั่นคง เป็นความลับ กล่าวคือกุญแจ *Ks* จะรู้เฉพาะผู้รับและผู้ส่ง หรือคู่สนทนาเท่านั้น แต่ยังคงให้การสนับสนุนการตรวจสอบข้อมูลโดยชอบด้วยกฎหมาย

การคำนวณการกู้คืนกุญแจ *Ks* สามารถแสดงได้ดังรูปที่ 3.6 และมีขั้นตอนดังต่อไปนี้

1) ผู้ร้องขอถอดรหัส $Ku_{req}[S_i \,||\, SGN]$ ด้วย Kr_{req} จะได้ S_i และ SGN

2) ผู้ร้องขอตรวจสอบ S_i ของ $Agent_i$ โดยการเปรียบเทียบ SGN

3) ผู้ร้องขอคำนวณกุญแจ *Ks* ด้วย $S_1 \oplus S_2 \oplus \ldots \oplus S_n$ และนำกุญแจ *Ks* ไปใช้ในการถอดรหัสต่อไป



รูปที่ 3.6 การกู้คืนกุญแจ *Ks* โดยผู้ร้องขอการกู้คืนกุญแจของ HADM-KRS

งานวิจัยระบบการกู้คืนกุญแจ HADM-KRS ที่ผู้วิจัยได้นำเสนอนี้ สามารถแก้ไขจุดอ่อนของระบบ KRS ที่กล่าวในตอนต้นได้ครบทุกกรณี

3.3 ระบบการกู้คืนกุญแจหลายเอเจนต์แบบกระจายที่มีความพร้อมใช้งานสูงอย่างง่าย (SHADM-KRS)

ระบบ SHADM-KRS เป็นระบบการกู้คืนกุญแจที่มีกระบวนการทำงานที่ทำให้กุญแจลับมีความมั่นคงสูง และผู้ใช้งานมีความเป็นส่วนตัว SHADM-KRS สามารถกู้คืนกุญแจ *Ks* ได้ในกรณีที่มีบาง KRA ในกลุ่มการกู้คืนล่ม แต่ไม่สามารถกำหนดเงื่อนไขว่าต้องมีจำนวน KRA คงเหลือที่สามารถให้บริการได้จำนวนเท่าไรจึงจะสามารถกู้คืนได้ การทำงานของ SHADM-KRS สามารถแสดงได้ดังต่อไปนี้

ผู้ที่มีส่วนร่วมในระบบและหน้าที่ของผู้ที่มีส่วนร่วมในระบบเหมือนกับระบบ HADM-KRS ที่ได้นำเสนอไว้ในหัวข้อ 3.2.1 และ 3.2.2 ตามลำดับ

3.3.1 กระบวนการทำงานของระบบ SHADM-KRS

กระบวนการทำงานหลักของระบบ SHADM-KRS แบ่งออกเป็นสองขั้นตอน คือ (1) การสร้างฟิลด์ในการกู้คืนกุญแจ (*KRF*)โดยผู้ส่ง และ(2) การกู้คืนกุญแจ *Ks* โดย KRA และ ผู้รับหรือหน่วยงานรัฐ โดยมีรายละเอียดดังต่อไปนี้

3.3.1.1 การสร้างฟิลด์ในการกู้คืนกุญแจ (*KRF*)

การสร้างฟิลด์ *KRF* จะกระทำโดยผู้ส่ง และสามารถแบ่งออกเป็นส่วนย่อย ๆ ได้สองขั้นตอน คือ (1) การสร้างส่วนประกอบของฟิลด์ *KRF* และ (2) การประกอบฟิลด์ *KRF*

1) การสร้างส่วนประกอบของฟิลด์ *KRF*

ในขั้นตอนนี้มีวิธีการเหมือนกับระบบการกู้คืนกุญแจ HADM-KRS ดังแสดงไว้ในหัวข้อที่ 3.2.3.1 หัวข้อย่อยที่ 1.1) – 1.4) แต่ในตอนท้ายได้เพิ่มกระบวนการ คือนำค่าความลับ *SGN* มาผ่านฟังก์ชันแฮช (Hash Function) จะได้ *h(SGN)* โดยค่าดังกล่าวใช้สำหรับการยืนยันตัวตนของ KRA ที่อยู่ในกลุ่มการกู้คืน

2) การประกอบฟิลด์ *KRF*

ฟิลด์ *KRF* ของ SHADM-KRS ประกอบด้วย ฟิลด์ *KRF* ย่อย ๆ (KRF_i's) ของ KRA ในกลุ่มการกู้คืนกุญแจ ค่า *h(SGN)* และค่าแอตทริบิวต์ TT_i

ฟิลด์ KRF_i ประกอบด้วย แอตทริบิวต์ S_i, *SGN* และ *other information* แต่ละส่วนประกอบมีความสำคัญดังนี้

S_i คือส่วนประกอบของกุญแจ *Ks*

SGN เป็นค่าสำหรับการยืนยันตัวตนของ KRA ที่อยู่ในกลุ่มการกู้คืน

h(SGN) เป็นค่าสำหรับการยืนยันตัวตนของ KRA ที่อยู่ในกลุ่มการกู้คืน

TT_i เป็นค่าสำหรับการกู้คืน S_i ในกรณี KRA ที่อยู่ในกลุ่มการกู้คืนล่ม

other information เก็บค่าอื่น ๆ ที่จำเป็นสำหรับการพิสูจน์ตัวจริงหรือการตรวจสอบเพื่อเพิ่มความมั่นคงปลอดภัยให้กับระบบ เช่น ใบรับรองกุญแจสาธารณะของเอเจนต์ และของผู้ที่มีสิทธิ์ในการร้องขอการกู้คืนกุญแจ เป็นต้น

โครงสร้างฟิลด์ *KRF* ของ SHADM-KRS

ฟิลด์ *KRF* ของ SHADM-KRS ประกอบด้วยฟิลด์ KRF_i จำนวน n ชิ้น ค่า *h(SGN)* และค่า TT_i จำนวน n ตัว ดังนี้

$$KRF = \{Ku_{agi}[\ KRF_i\text{'s}] \parallel h(SGN) \parallel TT_i\text{'s}\}$$

ฟิลด์ KRF_i ประกอบด้วย S_i, SGN และ *other information* โดยฟิลด์ KRF_i จะถูกเข้ารหัสด้วย Ku_{agi} ดังนี้

$$KRF_i = Ku_{agi}[\, S_i \,||\, SGN \,||\, other\ information\,]$$

ในส่วนนี้ฟิลด์ *KRF* จะเก็บค่า TT_i ของทุกเอเจนต์สำหรับการกู้คืน S_i ในกรณีที่มีบาง KRA ล่ม

3.3.1.2 การกู้คืนกุญแจ *Ks*

ขั้นตอนการกู้คืนกุญแจ *Ks* สามารถแบ่งออกเป็นส่วนย่อย ๆ ได้สองขั้นตอน คือ (1) การกู้คืนส่วนประกอบของกุญแจ และ(2) การกู้คืนกุญแจ *Ks*

1) การกู้คืนส่วนประกอบของกุญแจ (S_i)

การกู้คืนส่วนประกอบของกุญแจ แบ่งออกเป็นสองส่วน คือ (1) ส่วนของผู้ร้องขอการกู้คืนกุญแจ (Requester) และ(2) ส่วนของ KRA โดยมีรายละเอียดดังต่อไปนี้

1.1) ส่วนของผู้ร้องขอการกู้คืนกุญแจ (Requester)

ในส่วนนี้จะมีกระบวนการเหมือนกับระบบ HADM-KRS ดังแสดงในหัวข้อที่ 3.2.3.2

1.2) ส่วนของ KRA

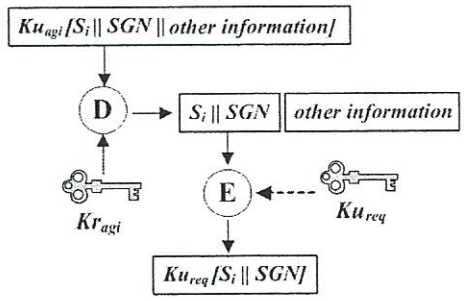
การกู้คืนส่วนประกอบของกุญแจ ของแต่ละ KRA (KRA_i) สามารถแสดงได้ดังรูปที่ 3.7 และมีขั้นตอนดังต่อไปนี้

1.2.1) $Agent_i$ ถอดรหัสฟิลด์ KRF_i ด้วย Kr_{agi} จะได้ S_i, SGN และ *other information*

1.2.2) $Agent_i$ ตรวจสอบ *other information* เช่น ใบรับรองกุญแจสาธารณะ (Public Key Certificate) ของผู้ร้องขอ เป็นต้น

1.2.3) $Agent_i$ เข้ารหัส S_i และ SGN ด้วย Ku_{req}

1.2.4) $Agent_i$ ส่ง $Ku_{req}[S_i \,||\, SGN]$ ให้กับผู้ร้องขอ เพื่อให้ผู้ร้องขอนำไปคำนวณหากุญแจ *Ks* ต่อไป



รูปที่ 3.7 การกู้คืนส่วนประกอบของกุญแจ (S_i) โดย KRA ของ SHADM-KRS

1.3) กรณีที่มีบาง KRA ล่ม

ในกรณีนี้ส่งผลให้ผู้ร้องขอการกู้คืนกุญแจ *Ks* ได้รับแอตทริบิวต์ TT_i ไม่ครบถ้วน ทั้งนี้ผู้ร้องขอสามารถใช้ TT_i ที่เก็บไว้ในฟิลด์ *KRF* มาคำนวณหา S_i ได้จาก

$$S_i = TT_i \oplus SGN$$

เมื่อมีการร้องขอการกู้คืน S_i เอเจนต์ต้องส่งค่า S_i พร้อมกับค่า *SGN* มายังผู้ร้องขอ เมื่อผู้ร้องขอคำนวณค่า S_i ได้ครบแล้ว จะสามารถคำนวณหากุญแจ *Ks* เพื่อนำไปใช้ในการถอดรหัสได้

การตรวจสอบ KRA ว่าเป็นเอเจนต์ที่อยู่ในกลุ่มการกู้คืนหรือไม่ ระบบ SHADM-KRS จะนำค่า *SGN* มาผ่านฟังก์ชันแฮช *h(SGN)* แล้วเปรียบเทียบกับค่า *h(SGN)* ที่อยู่ในฟิลด์ *KRF* หากมีค่าเหมือนกันแสดงว่าเป็นเอเจนต์ที่อยู่ในกลุ่มการกู้คืนเดียวกัน

2) การกู้คืนกุญแจ *Ks*

ขั้นตอนนี้มีกระบวนการเหมือนกับระบบ HADM-KRS ดังแสดงในหัวข้อที่ 3.1.3.2 และมีการพิสูจน์ตัวจริงของ KRA ตามวิธีที่ได้กล่าวมาแล้วข้างต้น

บทที่ 4
การประเมินความมั่นคงและสมรรถนะของระบบ

ในบทนี้เป็นการแสดงผลการเปรียบเทียบคุณสมบัติของระบบการกู้คืนกุญแจแบบ M-KRS และแสดงผลการประเมินกระบวนการทำงานของ M-KRS 3 ระบบ คือ (1) Typical M-KRS (2) HADM-KRS และ (3) SHADM-KRS พร้อมทั้งผลการประเมินความน่าเชื่อถือ (Reliability) และความพร้อมใช้งาน (Availability) ของระบบ M-KRS ที่รองรับการล่มของบางเอเจนต์และไม่รองรับการล่มของเอเจนต์

ทั้งนี้ในการประเมินกระบวนการทำงานของระบบ จะได้พิจารณาประเด็นหลัก ๆ ดังนี้คือ เวลาที่ใช้ในกระบวนการกู้คืนกุญแจ ซึ่งประกอบด้วย เวลาที่ใช้ในการสร้างฟิลด์ *KRF* เวลาที่ใช้ในการกู้คืนกุญแจ *Ks* และเวลาที่ใช้ในการกู้คืนส่วนประกอบของกุญแจ *Ks* (S_i) ในกรณีที่มีบางเอเจนต์ในกลุ่มการกู้คืนล่ม

4.1 การเปรียบเทียบคุณสมบัติของระบบ M-KRS

การเปรียบเทียบคุณสมบัติของระบบได้พิจารณาจากประเด็นสำคัญ 6 ประเด็น ซึ่งจะทำให้ระบบ M-KRS สามารถทำงานได้อย่างมีประสิทธิภาพ ดังนี้

1) ระบบสามารถทำงานได้โดยไม่ใช้ศูนย์กลางการกู้คืนกุญแจ (KRC)

2) ด้านความลับของกุญแจ *Ks* คือกุญแจ *Ks* มีการจัดเก็บอย่างมั่นคงปลอดภัย และมีสิทธิ์รู้ได้เฉพาะคู่สื่อสาร (ผู้รับและผู้ส่ง) เท่านั้น โดยที่หน่วยงานที่สามในที่นี้หมายถึง KRA จะไม่สามารถล่วงรู้กุญแจ *Ks* ได้

3) ด้านความพร้อมใช้งานของระบบ คือ ระบบสามารถให้บริการกู้คืนกุญแจ *Ks* ได้ แม้ในกรณีที่มีบางเอเจนต์ในกลุ่มการกู้คืนล่ม

4) ความสามารถของการกำหนดจำนวนขั้นต่ำของ KRA สำหรับการกู้คืนกุญแจ *Ks* คือระบบมีฟังก์ชันในการระบุจำนวน KRA ขั้นต่ำสำหรับการกู้คืนกุญแจได้ เพื่อเป็นประโยชน์ในการสร้างระดับความมั่นคงของระบบ เช่นกำหนดให้ใช้ KRA ในการกู้คืนกุญแจทั้งหมด 6 เอเจนต์ และกำหนดเอเจนต์ขั้นต่ำในการกู้คืนกุญแจเท่ากับ 4 เอเจนต์ นั่นหมายความว่า อนุญาตให้มี KRA ล่มได้มากที่สุดเพียง 2 เอเจนต์ หากมี KRA ในกลุ่มการกู้คืนล่มมากกว่านั้น จะไม่สามารถกู้คืนกุญแจ *Ks* ได้สำเร็จ

5) ความสามารถในการพิสูจน์ตัวจริงของ KRA ที่ไม่ได้อยู่ในกลุ่มการกู้คืน คือระบบมีฟังก์ชันในการพิสูจน์ KRA ในกลุ่มการกู้คืนเดียวกันว่าเป็นตัวจริงหรือไม่

6) การใช้รูปแบบมาตรฐานของฟิลด์ *KRF* คือ รองรับการกู้คืนกุญแจ *Ks* สำหรับการตรวจสอบข้อมูลที่ต้องสงสัยอย่างถูกต้องตามกฎหมายโดยหน่วยงานของรัฐ

ผลการเปรียบเทียบสามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 การเปรียบเทียบคุณสมบัติของระบบ M-KRS

คุณสมบัติของระบบ	Typical M-KRS	HADM-KRS	SHADM-KRS
1) ศูนย์กลางการกู้คืนกุญแจ (KRC)	✓	–	–
2) ความลับของกุญแจ *Ks*	–	✓	✓
3) ความพร้อมใช้งานของระบบสูง	–	✓	✓
4) ความสามารถในการกำหนดจำนวน KRA ขั้นต่ำ	–	✓	–
5) ความสามารถในการพิสูจน์ตัวจริงของ KRA	–	✓	✓
6) การใช้รูปแบบมาตรฐานของฟิลด์ *KRF*	✓	✓	✓
หมายเหตุ : สัญลักษณ์ – หมายถึงไม่มีคุณสมบัติ สัญลักษณ์ ✓ หมายถึงมีคุณสมบัติ			

จากตารางที่ 4.1 จะเห็นได้ว่า HADM-KRS เป็นระบบที่มีคุณสมบัติครบทั้ง 5 ประเด็น ส่วน SHADM-KRS ไม่มีฟังก์ชั่นการกำหนดจำนวน KRA ขั้นต่ำ เนื่องจากถูกออกแบบให้ระบบมีความซับซ้อนน้อยกว่า HADM-KRS และTypical M-KRS มีคุณสมบัติตามประเด็นสำคัญเพียงข้อเดียว คือเรื่องของการใช้รูปแบบมาตรฐานของฟิลด์ *KRF* เท่านั้น ดังนั้นสามารถสรุปได้ว่า HADM-KRS และ SHADM-KRS เป็นระบบ M-KRS ที่มีคุณสมบัติที่จำเป็นต่อระบบการกู้คืนกุญแจที่มีความพร้อมใช้งานและความมั่นคงสูง

4.2 การประเมินกระบวนการทำงานของระบบ M-KRS

ผู้วิจัยได้ประเมินและเปรียบเทียบกระบวนการทำงานของระบบ M-KRS โดยมีประเด็นหัวข้อที่ได้ทำการประเมิน คือ เวลาที่ใช้ในกระบวนการต่อไปนี้ (1) กระบวนการสร้างฟิลด์ *KRF* (2) กระบวนการกู้คืน S_i (3) กระบวนการกู้คืนกุญแจ *Ks* และ (4) กระบวนการการกู้คืน S_i ในกรณีที่มีบาง KRA ล่ม โดยได้ทดลองบนเครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลกลาง Intel Core Duo T2400 1.83 GHz หน่วยความจำ SDRAM ความจุ 1 GB

การประเมินระบบ HADM-KRS และ SHADM-KRS ใช้แอตทริบิวต์ต่าง ๆ ดังนี้

S_i	มีขนาด 128 บิต
SGN	มีขนาด 128 บิต
h(SGN)	มีขนาด 256 บิต
TT_i	มีขนาด 128 บิต
other information	มีขนาด 1,024 บิต

Typical M-KRS ใช้แอตทริบิวต์ต่าง ๆ ดังนี้

rk_i	มีขนาด 128 บิต
ik_i	มีขนาด 128 บิต

4.2.1 กระบวนการสร้างฟิลด์ *KRF*

กำหนดให้มีการใช้จำนวน KRA (n) ในการประเมินจำนวน 2 4 6 8 และ 10 เอเจนต์

4.2.1.1 Typical M-KRS

กระบวนการสร้างฟิลด์ *KRF* ของ Typical M-KRS (ได้นำเสนอไว้ในบทที่ 2 หัวข้อ 2.11.6) โดยมีแอตทริบิวต์ที่เกี่ยวข้องในระบบ ดังนี้ $\{rk_i\}$ $\{ik_i\}$ และ {*other information*} โดยที่ *i* คือ KRA ใด ๆ

4.2.1.2 HADM-KRS

กระบวนการสร้างฟิลด์ *KRF* ของ HADM-KRS (ได้นำเสนอไว้ในบทที่ 3 หัวข้อที่ 3.1.3.1) โดยมีแอตทริบิวต์ที่เกี่ยวข้องกับระบบ ดังนี้ $\{S_i\}$ {*SGN*} $\{TT_i\}$ และ {*other information*} โดยที่ *i* คือ KRA ใด ๆ และกำหนดให้มีการสำรอง TT_i ในกรณีที่มี KRA ล่มจำนวน $n - 2$

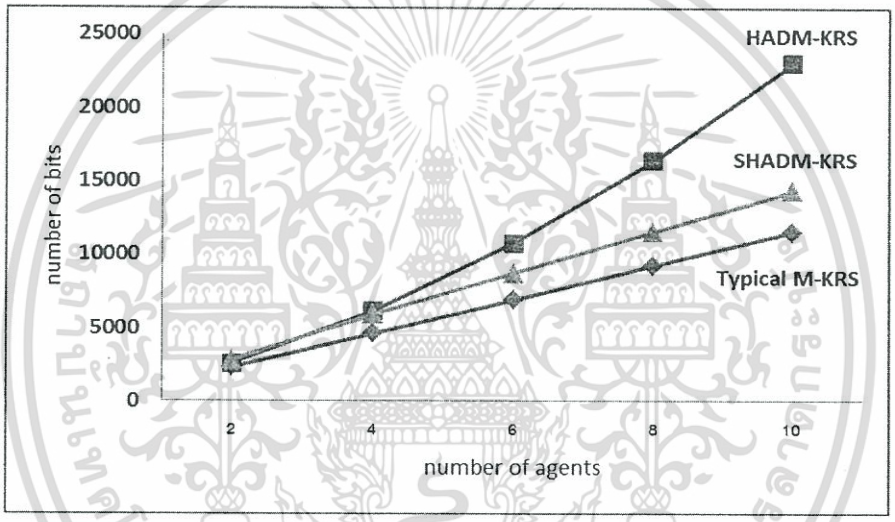
4.2.1.3 SHADM-KRS

กระบวนการสร้างฟิลด์ *KRF* ของ SHADM-KRS (ได้นำเสนอไว้ในบทที่ 3 หัวข้อ 3.2.1.1) โดยมีแอตทริบิวต์ที่เกี่ยวข้องในระบบ ดังนี้ $\{S_i\}$ {*SGN*} {*h(SGN)*} $\{TT_i\}$ และ {*other information*} โดยที่ *i* คือ KRA ใด ๆ

ขนาดของฟิลด์ *KRF* สามารถแสดงได้ดังตารางที่ 4.2 และรูปที่ 4.1

ตารางที่ 4.2 ขนาดของฟิลด์ *KRF* ของ Typical M-KRS HADM-KRS และ SHADM-KRS

จำนวน KRA	ขนาดของฟิลด์ ***KRF*** (บิต)		
	Typical M-KRS	HADM-KRS	SHADM-KRS
2	2,304	2,560	2,816
4	4,608	6,144	5,888
6	6,912	10,752	8,704
8	9,216	16,384	11,520
10	11,520	23,040	14,336



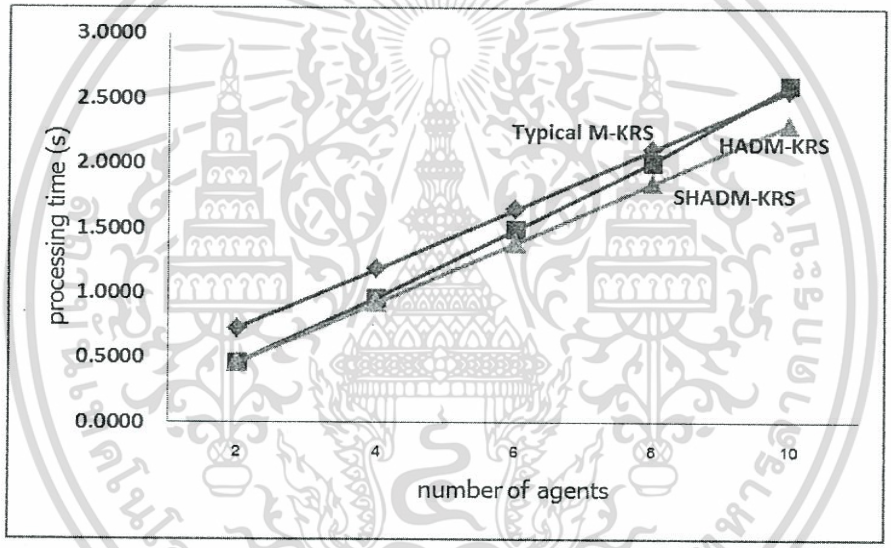
รูปที่ 4.1 การเปรียบเทียบขนาดของฟิลด์ *KRF*

จากตารางที่ 4.2 และรูปที่ 4.1 จะสังเกตเห็นว่าเมื่อใช้จำนวน KRA สำหรับการกู้คืนกุญแจจำนวน 2 เอเจนต์ ระบบ M-KRS จะมีจำนวนบิตที่บรรจุใน *KRF* ใกล้เคียงกัน เมื่อเพิ่มจำนวน KRA สำหรับการกู้คืนกุญแจ เป็น 4 6 8 และ 10 เอเจนต์ตามลำดับ จะใช้จำนวนบิตที่เพิ่มขึ้น โดยที่ระบบ HADM-KRS มีขนาดของฟิลด์ *KRF* ที่ใหญ่ที่สุด รองลงมาคือระบบ SHADM-KRS และ Typical M-KRS ตามลำดับ

เวลาที่ใช้ในกระบวนการสร้างฟิลด์ *KRF* ของ Typical M-KRS HADM-KRS และ SHADM-KRS แสดงดังตารางที่ 4.3 และรูปที่ 4.2

ตารางที่ 4.3 เวลาที่ใช้ในกระบวนการสร้างฟิลด์ *KRF* ของ Typical M-KRS HADM-KRS และ SHADM-KRS

จำนวน KRA	เวลาที่ใช้ในกระบวนการสร้างฟิลด์ ***KRF*** (วินาที)		
	Typical M-KRS	HADM-KRS	SHADM-KRS
2	0.7251	0.4615	0.4618
4	1.1851	0.9602	0.9224
6	1.6447	1.4872	1.3828
8	2.1048	2.0059	1.8434
10	2.5648	2.6069	2.3041



รูปที่ 4.2 การเปรียบเทียบเวลาที่ใช้ในกระบวนการสร้างฟิลด์ *KRF*

ผลการประเมินเวลาที่ใช้ในการสร้างฟิลด์ *KRF* พบว่าเมื่อใช้ KRA จำนวน 2 ถึง 8 เอเจนต์ ระบบ HADM-KRS และ SHADM-KRS ใช้เวลาในกระบวนการสร้างฟิลด์ *KRF* น้อยกว่า Typical M-KRS และเมื่อใช้จำนวน KRA เพิ่มขึ้นเป็น 10 เอเจนต์ ระบบ HADM-KRS ใช้เวลาในการสร้างฟิลด์ *KRF* มากที่สุดโดยมากกว่าทั้งสองระบบเพียงเล็กน้อยอย่างไม่มีนัยสำคัญ รองลงมาเป็น Typical M-KRS และ SHADM-KRS ตามลำดับ ทั้งนี้เนื่องจากระบบ HADM-KRS ต้องใช้เวลาในการจัดสรรแอตทริบิวต์ TT_i ซึ่งเป็นค่าแอตทริบิวต์ที่จะถูกนำมาใช้ในกระบวนการกู้คืนส่วนประกอบของกุญแจ ในกรณีที่มีบาง KRA ในกลุ่มการกู้คืนล่ม โดยที่กระบวนการจัดสรรค่า TT_i ของ HADM-KRS นั้นช่วยให้ระบบสามารถกำหนดเงื่อนไขว่า KRA ในกลุ่มการกู้คืนจะต้อง

สามารถให้บริการอยู่ในระบบจำนวนกี่เอเจนต์ ระบบจึงจะสามารถให้บริการการกู้คืนกุญแจได้ ส่วนระบบ Typical M-KRS นั้นไม่สามารถกู้คืนกุญแจได้เมื่อมีบาง KRA ในกลุ่มการกู้คืนล่ม และต้องใช้เวลาในการเข้ารหัส จะสังเกตเห็นว่าระบบ HADM-KRS และ SHADM-KRS ใช้เวลาในกระบวนการสร้างฟิลด์ *KRF* น้อยกว่า Typical M-KRS เมื่อใช้จำนวนเอเจนต์ไม่เกิน 8 เอเจนต์ และมีความสามารถที่เพิ่มขึ้นในด้านความพร้อมใช้งานที่สูงกว่าของระบบ

ทั้งนี้เวลาที่ใช้ในกระบวนการสร้างฟิลด์ *KRF* จะขึ้นอยู่กับจำนวนของ KRA ที่ใช้ในระบบการกู้คืนกุญแจ กล่าวคือหากใช้ KRA จำนวนมาก ก็จะใช้เวลาในกระบวนการสร้างฟิลด์ KRF_i มากขึ้นตามลำดับ เนื่องจากฟิลด์ KRF_i ได้ถูกสร้างขึ้นเท่ากับจำนวนของ KRA ที่ใช้ในระบบ

4.2.2 กระบวนการกู้คืนส่วนประกอบของกุญแจ (S_i)

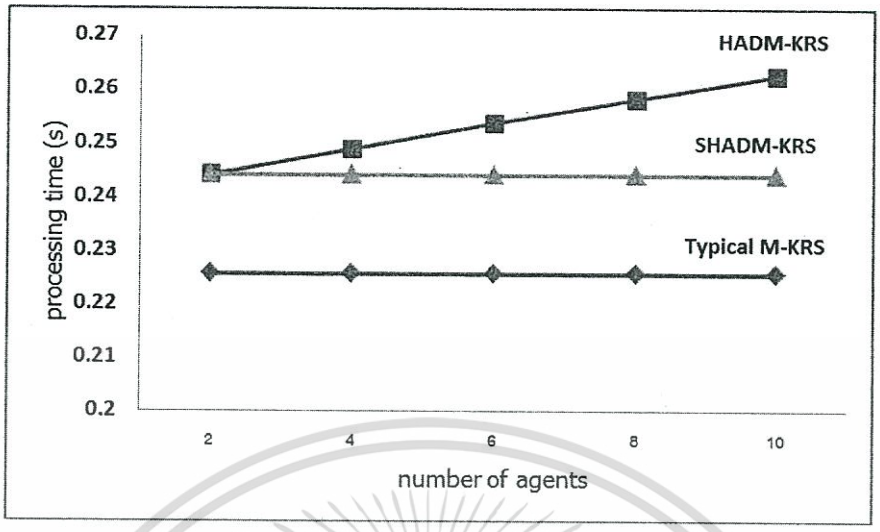
ในกระบวนการนี้การกู้คืน S_i อาศัยข้อมูลที่จัดเก็บอยู่ในฟิลด์ฟิลด์ *KRF* โดยที่ S_i ถูกนำมาคำนวณหากุญแจ *Ks*

การกู้คืน S_i เป็นหน้าที่ของ KRA ในกลุ่มการกู้คืน ซึ่งแต่ละ KRA ทำงานอย่างเป็นอิสระ ดังนั้นเวลาที่ใช้ในการกู้คืน S_i จึงไม่ขึ้นอยู่กับจำนวน KRA

เวลาที่ใช้ในกระบวนการกู้คืน S_i ของ Typical M-KRS HADM-KRS และ SHADM-KRS แสดงดังตารางที่ 4.4 และรูปที่ 4.3

ตารางที่ 4.4 เวลาที่ใช้ในกระบวนการกู้คืน S_i ของ Typical M-KRS HADM-KRS และ SHADM-KRS

จำนวน KRA	เวลาที่ใช้ในกระบวนการกู้คืน S_i (วินาที)		
	Typical M-KRS	HADM-KRS	SHADM-KRS
2	0.2256	0.2440	0.2440
4	0.2256	0.2488	0.2440
6	0.2256	0.2536	0.2440
8	0.2256	0.2581	0.2440
10	0.2256	0.2626	0.2440



รูปที่ 4.3 การเปรียบเทียบเวลาที่ใช้ในกระบวนการกู้คืนส่วนประกอบของกุญแจ (S_i)

ผลการประเมินเวลาที่ใช้ในกระบวนการกู้คืน S_i พบว่าระบบ HADM-KRS และ SHAM-KRS ใช้เวลาในการกู้คืน S_i มากกว่า Typical M-KRS เล็กน้อยอย่างไม่มีนัยสำคัญ เมื่อใช้จำนวน KRA ในการกู้คืนกุญแจจำนวน 2 เอเจนต์ ระบบ HADM-KRS และ SHAM-KRS ใช้เวลาในการกู้คืน S_i เท่ากัน เมื่อเพิ่มจำนวน KRA ขึ้นระบบ HADM-KRS ใช้เวลาในการกู้คืน S_i เพิ่มขึ้นตามลำดับเล็กน้อย และใช้เวลามากกว่า SHAM-KRS และ Typical M-KRS โดยเป็นผลมาจากขนาดของฟิลด์ KRF_i กล่าวคือระบบ HADM-KRS มีขนาดของฟิลด์ KRF_i ใหญ่กว่าอีกสองระบบ ทั้งนี้ระบบ SHAM-KRS และ Typical M-KRS ได้ถูกออกแบบให้บรรจุค่าของแอตทริบิวต์ในฟิลด์ KRF_i คงที่โดยไม่ขึ้นอยู่กับจำนวนของ KRA ส่งผลให้ไม่ว่าจะใช้ KRA จำนวนเท่าใดก็ตาม จะใช้เวลาในกระบวนการกู้คืน S_i เท่ากัน

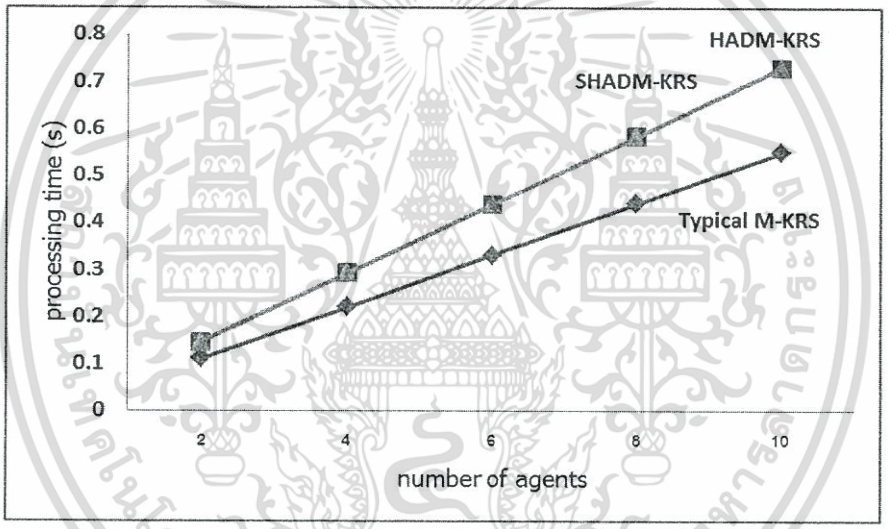
4.2.3 กระบวนการกู้คืนกุญแจ *Ks*

ระบบการกู้คืนกุญแจลับ HADM-KRS และ SHADM-KRS ได้ถูกออกแบบให้การกู้คืนกุญแจ *Ks* กระทำที่ผู้ร้องขอการกู้คืนกุญแจ โดยใช้ S_i ที่ได้รับจาก KRA ทั้งนี้เพื่อให้กุญแจ *Ks* มีความมั่นคงปลอดภัยสูงสุด กล่าวคือกุญแจ *Ks* จะรู้เฉพาะคู่สื่อสารซึ่งก็คือผู้รับและผู้ส่งเท่านั้น และมีกระบวนการพิสูจน์ตัวจริงของ KRA จากค่า *SGN* ทั้งนี้ยังคงให้การสนับสนุนการตรวจสอบข้อมูลที่ต้องสงสัยอย่างถูกต้องตามกฎหมาย ส่วนงานวิจัยระบบการกู้คืนกุญแจลับ Typical M-KRS ได้ถูกออกแบบให้การกู้คืนกุญแจ *Ks* กระทำที่ KRC

เวลาที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* ของ HADM-KRS SHADM-KRS และ Typical M-KRS แสดงดังตารางที่ 4.5 และรูปที่ 4.4

ตารางที่ 4.5 เวลาที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* ของ Typical M-KRS HADM-KRS และ SHADM-KRS

จำนวน KRA	เวลาที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* (วินาที)		
	Typical M-KRS	HADM-KRS	SHADM-KRS
2	0.1099	0.1459	0.1459
4	0.2197	0.2917	0.2917
6	0.3295	0.4375	0.4375
8	0.4394	0.5834	0.5834
10	0.5492	0.7292	0.7292



รูปที่ 4.4 การเปรียบเทียบเวลาที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks*

ผลการประเมินเวลาที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* พบว่าระบบ HADM-KRS และ SHAM-KRS ใช้เวลาในการกู้คืนกุญแจ *Ks* เท่ากัน เนื่องจากมีกระบวนการในการกู้คืนกุญแจเหมือนกัน และทั้งสองระบบใช้เวลามากกว่า Typical M-KRS เล็กน้อยอย่างไม่มีนัยสำคัญ

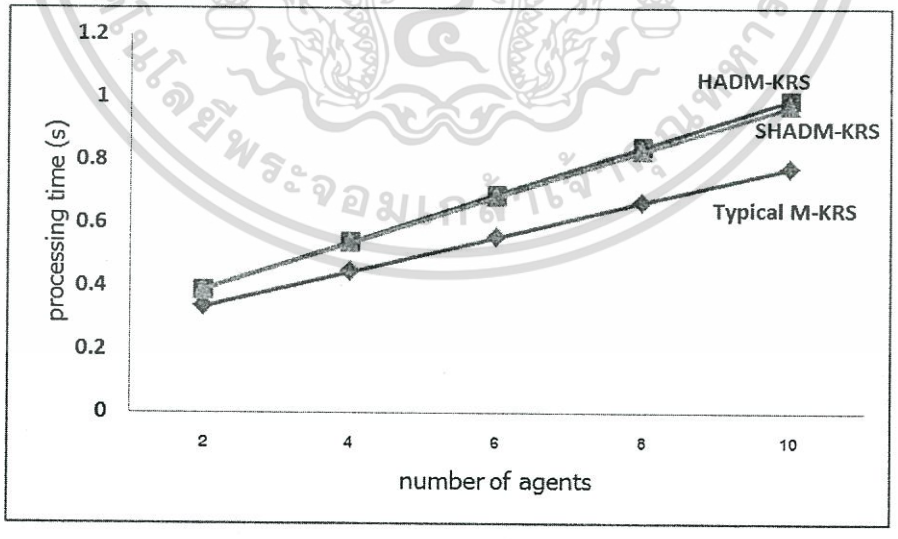
4.2.4 เวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks*

เวลาทั้งหมดที่ใช้ในการกู้คืนกุญแจ *Ks* ของระบบการกู้คืนกุญแจลับ HADM-KRS และ SHADM-KRS ถูกประเมินจากผลรวมของเวลาที่ใช้ในกู้คืนส่วนประกอบของกุญแจทั้งหมด (S_i, i = 1 ถึง n) กับเวลาที่ใช้ในการคำนวณกุญแจ *Ks* ส่วน Typical M-KRS ได้ทดลองหาเวลาที่ใช้ในการกู้คืนกุญแจ *Ks* ที่ KRC

เวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* ของ HADM-KRS SHADM-KRS และ Typical M-KRS แสดงดังตารางที่ 4.6 และรูปที่ 4.5

ตารางที่ 4.6 เวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* ของ Typical M-KRS HADM-KRS และ SHADM-KRS

จำนวน KRA	เวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* (วินาที)		
	Typical M-KRS	HADM-KRS	SHADM-KRS
2	0.3355	0.3899	0.3899
4	0.4453	0.5405	0.5357
6	0.5551	0.6911	0.6815
8	0.665	0.8415	0.8274
10	0.7748	0.9918	0.9732



รูปที่ 4.5 การเปรียบเทียบเวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks*

ผลการประเมินเวลาทั้งหมดที่ใช้ในกระบวนการกู้คืนกุญแจ *Ks* พบว่าระบบ HADM-KRS และ SHAM-KRS ใช้เวลาทั้งหมดในการกู้คืนกุญแจ *Ks* มากกว่า Typical M-KRS เล็กน้อย และ HADM-KRS จะใช้เวลามากกว่า SHAM-KRS เล็กน้อยอย่างไม่มีนัยสำคัญ ทั้งนี้ระบบ HADM-KRS และSHAM-KRS เป็นระบบการกู้คืนกุญแจที่มีความพร้อมใช้งานสูงและสามารถทำงานโดยไม่อาศัย KRC จึงทำให้มีกระบวนการทำงานที่ซับซ้อนกว่า แต่มีประสิทธิภาพที่เหนือกว่าระบบ Typical M-KRS

4.2.4 กระบวนการกู้คืนส่วนประกอบของกุญแจ ในกรณีที่มีบาง KRA ล่ม

ระบบการกู้คืนกุญแจลับ HADM-KRS และSHADM-KRS ได้ถูกออกแบบให้ระบบมีความสามารถในการกู้คืนกุญแจลับได้สำเร็จ แม้มีบาง KRA ในกลุ่มการกู้คืนล่ม เป็นฟังก์ชันที่สนับสนุนให้ระบบมีความพร้อมใช้งานสูง

กำหนดให้ ใช้ KRA ในการประเมิน (n) จำนวน 4 6 8 และ 10 เอเจนต์

KRA ล่ม จำนวนเท่ากับ n – 2

KRA ล่ม เป็น KRA ที่มีหมายเลขเอเจนต์ที่อยู่ติดกัน เช่น

หากใช้ KRA ในการกู้คืนกุญแจจำนวน 4 เอเจนต์ คือ KRA_1 KRA_2 KRA_3 และ KRA_4 ให้ KRA ที่ล่มได้เป็น 2 เอเจนต์ ดังนี้ {KRA_1 และ KRA_2} หรือ {KRA_2 และ KRA_3} หรือ {KRA_3 และ KRA_4} หรือ {KRA_4 และ KRA_1} เป็นต้น

เวลาที่ใช้ในกระบวนการกู้คืน S_i ของ HADM-KRS และ SHADM-KRS สามารถแสดงได้ดังตารางที่ 4.7

ตารางที่ 4.7 เวลาที่ใช้ในกระบวนการกู้คืน S_i เมื่อเกิดกรณีมีบาง KRA ล่มของ HADM-KRS และ SHADM-KRS

จำนวน KRA	จำนวน KRA_i ล่ม	เวลาที่ใช้ในกระบวนการกู้คืน S_i เมื่อเกิดกรณีบาง KRA ล่ม (วินาที)	
		HADM-KRS	SHADM-KRS
4	2	0.6030729	0.0000729
6	4	0.7882179	0.0001179
8	6	0.7985519	0.0001519
10	8	0.8018795	0.0001795

ผลการประเมินเวลาที่ใช้ในกระบวนการกู้คืน S_i เมื่อเกิดกรณีมีบาง KRA ล่ม ระบบ HADM-KRS จะใช้เวลามากกว่า SHADM-KRS เนื่องจาก HADM-KRS ต้องส่งคำร้องขอการกู้คืน S_i ไปที่ KRA ในกลุ่มการกู้คืนที่ยังให้บริการได้ และรอรับ S_i จาก KRA ดังกล่าว โดยที่ระบบ SHADM-KRS นั้น ผู้รับสามารถกู้คืน S_i ที่หายไปได้โดยไม่ต้องใช้บริการ KRA

การประมวลผลที่เกิดขึ้นในระบบ M-KRS สามารถสรุปได้ดังตารางที่ 4.8

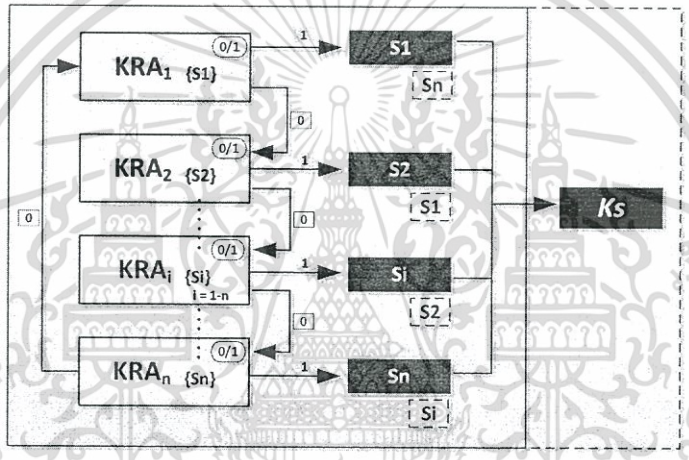
ตารางที่ 4.8 สรุปการประมวลผลในระบบ M-KRS

กระบวนการ	Typical-MKRS	HADM-KRS	SHADM-KRS
การสร้าง KRF	**ผู้รับ**		
เข้ารหัส	n + 1 ครั้ง	n ครั้ง	
ถอดรหัส	-	-	
Random number	n ตัว / 1 ครั้ง	ครั้ง 1: n-1 ตัว (S_i's) ครั้ง 2: n ตัว (R_i's)	
Exclusive OR	2 จำนวน / n รอบ	ครั้ง 1: n จำนวน / 1 รอบ (S_i's) ครั้ง 2: n จำนวน / 1 รอบ (SGN) ครั้ง 3: 2 จำนวน / n รอบ (TT_i's)	
		คำนวณการจัดสรร TT_i's	-
การกู้คืน S_i	**KRA_i**		
เข้ารหัส	1 ครั้ง	1 ครั้ง	
ถอดรหัส	1 ครั้ง	1 ครั้ง	
Exclusive OR	-	-	
การกู้คืน *Ks*	KRC	**ผู้รับ**	
เข้ารหัส	1 ครั้ง	-	
ถอดรหัส	n ครั้ง	n ครั้ง	
Exclusive OR	n จำนวน / 1 รอบ	n จำนวน / 1 รอบ	
การกู้คืน S_i เมื่อ KRA ล่ม	-	**KRA_i** / **ผู้รับ**	**ผู้รับ**
เข้ารหัส	-	1 ครั้ง / 1 ครั้ง	-
ถอดรหัส	-	1 ครั้ง / 1 ครั้ง	-
Exclusive OR	-	- / -	2 จำนวน / รอบเท่ากับ S_i ที่ขาด

4.3 การประเมินความน่าเชื่อถือและความพร้อมใช้งานของระบบ M-KRS

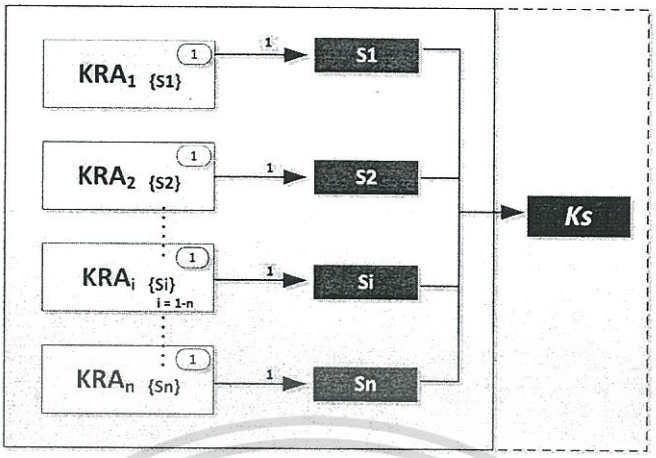
การประเมินความน่าเชื่อถือและความพร้อมใช้งานของระบบ MKRS ได้ทำการประเมินเปรียบเทียบระบบ M-KRS ที่ทำงานแตกต่างกัน 2 รูปแบบ คือ (1) M-KRS ที่รองรับการล่มของบาง KRA (Redundant KRA) กล่าวคือสามารถกู้คืนกุญแจ *Ks* ได้เมื่อมีบาง KRA ล่ม คือระบบ HADM-KRS และ SHADM-KRS และ(2) M-KRS ที่ไม่รองรับการล่มของ KRA (Non-redundant KRA) โดยระบบจะไม่สามารถกู้คืนกุญแจ *Ks* ได้เมื่อมี KRA ล่ม คือ Typical M-KRS

สถาปัตยกรรมของ KRA ในระบบ M-KRS ทั้งสองรูปแบบ สามารถแสดงได้ดังรูปที่ 4.6 และรูปที่ 4.7 ตามลำดับ



รูปที่ 4.6 สถาปัตยกรรมของ KRA ในระบบ M-KRS แบบที่มีการรองรับการล่มของบาง KRA

จากรูปที่ 4.6 ระบบ M-KRS ใช้ KRA มากกว่า 2 ตัว แต่ละตัวทำหน้าที่ในการกู้คืนส่วนประกอบของกุญแจ และทำงานเป็นอิสระต่อกัน หากมี KRA ตัวใดตัวหนึ่งล่ม (ให้ล่มได้ไม่เกินจำนวน n – 2 เมื่อ n คือจำนวน KRA ที่ใช้ทั้งหมด) KRA ที่อยู่ในตำแหน่งถัดไปจะทำหน้าที่ในการกู้คืนส่วนประกอบของกุญแจแทน ส่งผลให้ถึงแม้จะมี KRA บางเอเจนต์ล่ม ระบบก็ยังสามารถกู้คืนกุญแจลับได้



รูปที่ 4.7 สถาปัตยกรรมของ KRA ในระบบ M-KRS แบบที่ไม่มีการรองรับการล่มของบาง KRA

จากรูปที่ 4.7 ระบบ M-KRS ใช้ KRA มากกว่า 2 ตัว แต่ละตัวทำหน้าที่ในการกู้คืนส่วนประกอบของกุญแจ และทำงานเป็นอิสระต่อกัน หากมีตัวใดตัวหนึ่งล่ม จะทำให้กู้คืนกุญแจลับไม่สำเร็จ

4.3.1 ความน่าเชื่อถือของระบบ M-KRS

ความน่าเชื่อถือของระบบ M-KRS ถูกพิจารณาจากค่าความน่าจะเป็น (Probability: P) ที่ KRA ในกลุ่มการกู้คืนกุญแจเดียวกันล่มหรือไม่สามารถทำงานได้ โดยกำหนดให้ KRA แต่ละเอเจนต์มีความสามารถในการทำงานได้อย่างต่อเนื่องไม่ล้มเหลวที่ 95 เปอร์เซ็นต์ โดยในการหาค่าความน่าจะเป็นได้ใช้วิธีการแจกแจงแบบทวินาม (Binomial distribution) [41]

การประเมินนี้เป็นการประเมินเพื่อบ่งบอกถึงความสามารถในการคาดหวังว่าระบบจะไม่เกิดความผิดพลาด (Failure) หรือล้มเหลวในการให้บริการได้ดีเพียงใด

กำหนดให้ใช้ KRA ที่ทำหน้าที่ร่วมกันในการกู้คืนกุญแจ *Ks* จำนวน 8 กลุ่ม แต่ละกลุ่มมีสมาชิกคือจำนวน KRA (n) ตั้งแต่ 2, 3,… ถึง 6 เอเจนต์ ตามลำดับ

$$n = 2, 3, \ldots, 6$$

จำนวน KRA ที่ล่ม (x) จำนวน 0, 1, 2, 3, …., n เอเจนต์

$$x = 0, 1, 2, 3, \ldots, n$$

ความน่าจะเป็นที่แต่ละ KRA จะล่ม (t) เท่ากับ 5 เปอร์เซ็นต์

$$t = 0.05$$

KRA สามารถทำงานได้อย่างต่อเนื่องไม่ล้มเหลว (q) เท่ากับ 95 เปอร์เซ็นต์

$$q = 0.95$$

ในการประเมินตามจำนวน n เอเจนต์ จะมีเหตุการณ์ที่เอเจนต์ล่ม x เอเจนต์ สามารถเขียนฟังก์ชันแจกแจงแบบทวินามได้ดังนี้

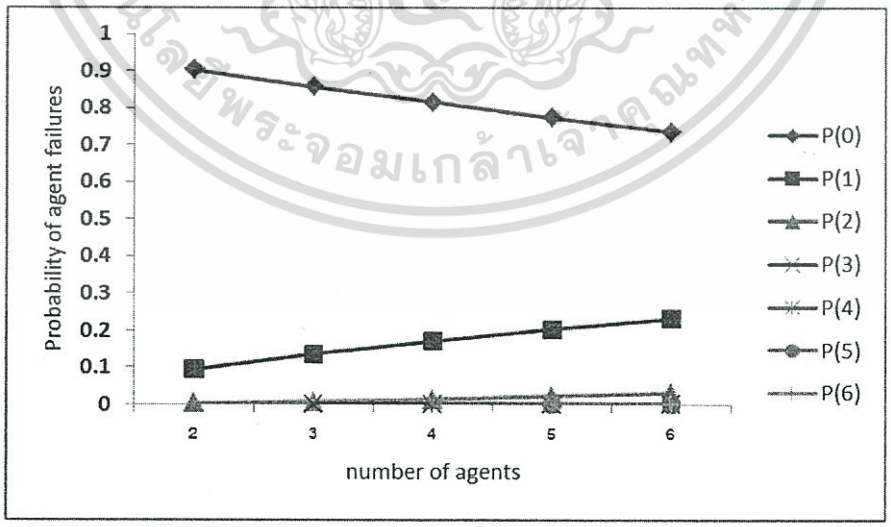
$$P(x) = {}^{n}C_{x}\, t^{x} q^{n-x}$$

$$P(x) = \frac{n!}{x!\,(n-x)!}\, t^{x} q^{n-x} \qquad (4.1)$$

การประเมินความน่าจะเป็นของระบบที่มี KRA ในกลุ่มการกู้คืนล่ม ตามสมการที่ (4.1) สามารถแสดงได้ดังตารางที่ 4.9 และรูปที่ 4.8

ตารางที่ 4.9 ความน่าจะเป็นของระบบที่มี KRA ในกลุ่มการกู้คืนล่ม

จำนวน KRA (n)	ความน่าจะเป็นของระบบที่มี KRA ในกลุ่มการกู้คืนล่ม						
	P(0)	P(1)	P(2)	P(3)	P(4)	P(5)	P(6)
2	0.90250000	0.09500000	0.00250000				
3	0.85737500	0.13537500	0.00712500	0.00012500			
4	0.81450625	0.17147500	0.01353750	0.00047500	0.00000625		
5	0.77378094	0.20362656	0.02143438	0.00112813	0.00002969	0.00000031	
6	0.73509189	0.23213428	0.03054398	0.00214344	0.00008461	0.00000178	0.00000002



รูปที่ 4.8 ความน่าจะเป็นของระบบที่มี KRA ในกลุ่มการกู้คืนล่ม

จากรูปที่ 4.6 ความน่าจะเป็นของ KRA ที่จะไม่ล่ม (P(0)) หรือ KRA ที่สามารถให้บริการได้ มีค่าลดลงเมื่อใช้จำนวน KRA ในการกู้คืนกุญแจเพิ่มขึ้น และค่าความน่าจะเป็นของจำนวน KRA ที่ล่มจะมีค่าเพิ่มขึ้นเมื่อมีการใช้จำนวน KRA เพิ่มขึ้น ดังนั้นจึงสรุปได้ว่า เมื่อใช้จำนวน KRA มากขึ้น จะส่งผลให้มีความน่าจะเป็นของ KRA ล่มมากขึ้นตามลำดับ

การประเมินความน่าเชื่อถือของระบบ M-KRS เปรียบเทียบระหว่างระบบที่รองรับการล่มของบาง KRA และระบบที่ไม่รองรับการล่มของบาง KRA ตามสมการที่ (4.2) และ (4.3) ตามลำดับ สามารถแสดงได้ดังตารางที่ 4.10 และรูปที่ 4.9

ในการประเมินความน่าเชื่อถือของระบบที่รองรับการล่มของบาง KRA กำหนดให้มีจำนวน KRA ที่ล่มเท่ากับ n – 2 เอเจนต์ ส่วนระบบที่ไม่รองรับการล่มของบาง KRA นั้น มีจำนวน KRA ที่ล่มเท่ากับ 0 เอเจนต์

ความน่าเชื่อถือของระบบที่รองรับการล่มของบาง KRA (R') หาได้จาก

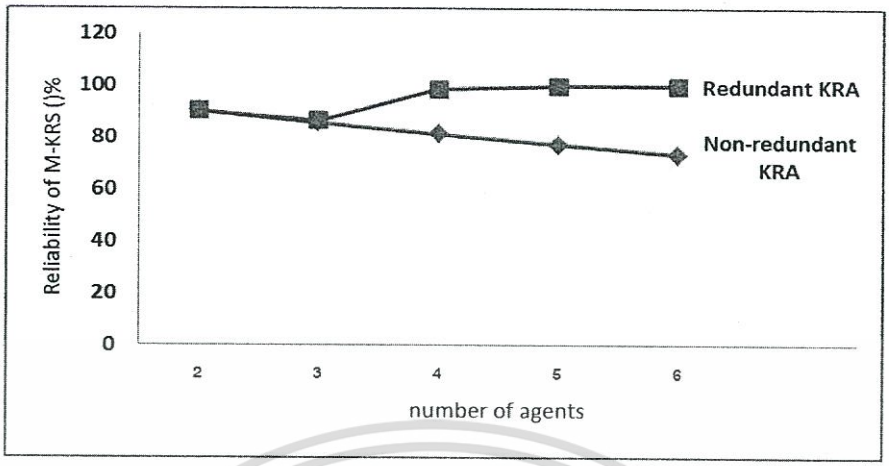
$$R' = \{1- P(x = n-2)\} * 100 \quad (4.2)$$

ความน่าเชื่อถือของระบบที่ไม่รองรับการล่มของบาง KRA (R) หาได้จาก

$$R = P(0) * 100 \quad (4.3)$$

ตารางที่ 4.10 การเปรียบเทียบความน่าเชื่อถือของระบบ M-KRS ระหว่างรูปแบบที่มีกับไม่มีการรองรับการล่มของ KRA

จำนวน KRA (n)	ความน่าเชื่อถือของ M-KRS (%)	
	ระบบที่รองรับการล่มของบาง KRA	ระบบที่ไม่รองรับการล่มของบาง KRA
2	90.25	90.25
3	86.46	85.74
4	98.65	81.45
5	99.89	77.38
6	99.99	73.51



รูปที่ 4.9 การเปรียบเทียบความน่าเชื่อถือของระบบ M-KRS

จากการเปรียบเทียบความน่าเชื่อถือของระบบ M-KRS ระหว่างรูปแบบที่มีกับไม่มีการรองรับการล่มของ KRA พบว่าการใช้จำนวน KRA ในการกู้คืนจำนวนมากขึ้นมีผลต่อความน่าเชื่อถือของระบบ M-KRS ทั้งสองรูปแบบ โดยระบบที่มีการรองรับการล่มของบาง KRA จะมีค่าความน่าเชื่อถือของระบบเพิ่มขึ้นตามลำดับเมื่อใช้จำนวน KRA ในการกู้คืนกุญแจเพิ่มขึ้น เพราะโอกาสของการเกิดเหตุการณ์ที่ KRA ล่มมีน้อยลง และเมื่อใช้จำนวน KRA จำนวน 6 เอเจนต์ จะมีค่าความน่าเชื่อถือมากจนมีค่าเข้าใกล้ 100 % ในทางตรงกันข้ามระบบที่ไม่มีการรองรับการล่มของบาง KRA จะมีค่าความน่าเชื่อถือของระบบลดลงอย่างมีนัยสำคัญ เมื่อใช้จำนวน KRA ในการกู้คืนจำนวนเพิ่มขึ้น เพราะโอกาสในการเกิดเหตุการณ์ที่ KRA ล่มมีมากขึ้น

อย่างไรก็ตามเมื่อใช้ KRA จำนวน 2 เอเจนต์ ระบบ M-KRS ทั้งสองรูปแบบมีค่าความน่าเชื่อถือของระบบเท่ากัน เพราะไม่มีการสำรอง KRA ไว้ในกรณีที่มีบางเอเจนต์ล่ม และเมื่อใช้ KRA 3 เอเจนต์ ระบบ M-KRS ทั้งสองรูปแบบมีค่าความน่าเชื่อถือของระบบใกล้เคียงกัน โดยระบบที่รองรับการล่มของบาง KRA มีค่าความน่าเชื่อถือของระบบสูงกว่าเล็กน้อย

4.3.2 ความพร้อมใช้งานของระบบ M-KRS

ความพร้อมใช้งานของระบบถูกพิจารณาจากค่าความน่าจะเป็นของเอเจนต์ล่ม ประกอบกับค่าเฉลี่ยของอัตราการซ่อมแซมสูงสุดที่ยอมรับได้ โดยในที่นี้กำหนดให้มีค่าเท่ากับ 5 เปอร์เซ็นต์

การประเมินความพร้อมใช้งานของระบบ M-KRS เปรียบเทียบระหว่างระบบที่รองรับการล่มของบาง KRA และระบบที่ไม่รองรับการล่มของบาง KRA ตามสมการที่ (4.4) และ (4.5) ตามลำดับ สามารถแสดงได้ดังตารางที่ 4.11 และรูปที่ 4.10

กำหนดให้ ค่าเฉลี่ยของอัตราการซ่อมแซมสูงสุดที่ยอมรับได้ (p) เท่ากับ 5 เปอร์เซ็นต์

$p = 0.05$

ความพร้อมใช้งานของระบบที่รองรับการล่มของบาง KRA (A') หาได้จาก

อัตราที่ KRA จะล่มเท่ากับ n - 2 ในหนึ่งหน่วยเวลา (F')

$$F' = \frac{1}{P(n-2)}$$

$$A' = \frac{F'}{F' + (\frac{1}{p})} \qquad (4.4)$$

ความพร้อมใช้งานของระบบที่ไม่รองรับการล่มของบาง KRA (A) หาได้จาก

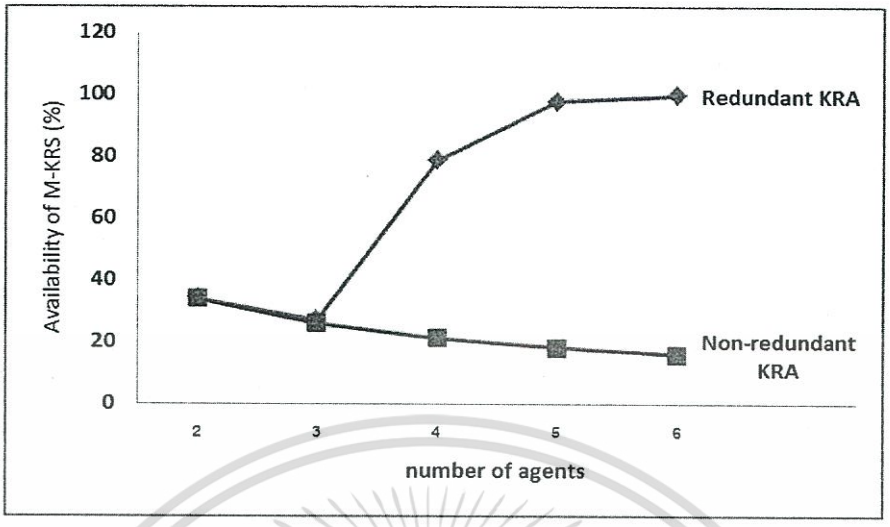
อัตราที่ KRA จะล่มเท่ากับ 1 - P(0) ในหนึ่งหน่วยเวลา (F)

$$F = \frac{1}{P(1-P(0))}$$

$$A = \frac{F}{F + (\frac{1}{p})} \qquad (4.5)$$

ตารางที่ 4.11 การเปรียบเทียบความพร้อมใช้งานของระบบ M-KRS ระหว่างรูปแบบที่มีกับไม่มีการรองรับการทำงานของบาง KRA

จำนวน KRA (n)	ความพร้อมใช้งานของ M-KRS (%)	
	ระบบที่รองรับการล่มของบาง KRA	ระบบที่ไม่รองรับการล่มของบาง KRA
2	33.90	33.90
3	26.97	25.96
4	78.69	21.23
5	97.79	18.10
6	99.83	15.88



รูปที่ 4.10 การเปรียบเทียบความพร้อมใช้งานของระบบ M-KRS

การประเมินความพร้อมใช้งานของระบบ M-KRS พบว่าระบบที่มีการรองรับการล่มของบาง KRA จะมีค่าความพร้อมใช้งานของระบบเพิ่มขึ้นเมื่อใช้ KRA จำนวนมากขึ้น และเมื่อใช้จำนวน KRA จำนวน 6 เอเจนต์ จะมีค่าความพร้อมใช้งานมากจนมีค่าเข้าใกล้ 100% แต่ระบบที่ไม่มีการรองรับการล่มของบาง KRA จะเกิดความเสี่ยงของการมี KRA ที่อยู่ในกลุ่มการกู้คืนล่ม เมื่อมีการใช้ KRA จำนวนมากขึ้น จึงมีค่าความพร้อมใช้งานของระบบลดลงไปตามลำดับอย่างมีนัยสำคัญ เมื่อใช้ KRA เพิ่มขึ้น ดังนั้นสามารถสรุปได้ว่า ระบบที่รองรับการล่มของบาง KRA เป็นระบบที่ความพร้อมใช้งานสูง

อย่างไรก็ตามเมื่อใช้ KRA จำนวน 2 เอเจนต์ ระบบ M-KRS ทั้งสองรูปแบบมีค่าความพร้อมใช้งานของระบบเท่ากัน และเมื่อใช้ KRA 3 เอเจนต์ ระบบ M-KRS ทั้งสองรูปแบบมีค่าความพร้อมใช้งานของระบบใกล้เคียงกัน โดยระบบที่รองรับการล่มของบาง KRA มีค่าความน่าเชื่อถือของระบบสูงกว่าเล็กน้อย

4.4 การวิเคราะห์เพื่อหาความเหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือ (Security and reliability trade-off) ของระบบที่รองรับการล่มของบาง KRA

กำหนดให้จำนวน KRA ที่ใช้ในการกู้คืนกุญแจ 5 เอเจนต์

$n = 5$

จำนวน KRA ที่ล่ม (x) จำนวน 0, 1, 2, 3 เอเจนต์

$x = 0, 1, 2, 3$ (เหลือ KRA ที่ใช้ในการกู้คืนกุญแจ 2, 3, 4, 5 เอเจนต์)

การกำหนดระดับความมั่นคง

ใช้ 5 เอเจนต์ในการกู้คืนกุญแจ ระดับความมั่นคง 100 เปอร์เซ็นต์

ใช้ 4 เอเจนต์ในการกู้คืนกุญแจ ระดับความมั่นคง 80 เปอร์เซ็นต์

ใช้ 3 เอเจนต์ในการกู้คืนกุญแจ ระดับความมั่นคง 60 เปอร์เซ็นต์

ใช้ 2 เอเจนต์ในการกู้คืนกุญแจ ระดับความมั่นคง 40 เปอร์เซ็นต์

ความน่าจะเป็นที่แต่ละ KRA จะล่ม (t) เท่ากับ 1, 5 และ 10 เปอร์เซ็นต์

$t = 0.01, 0.05$ และ 0.1

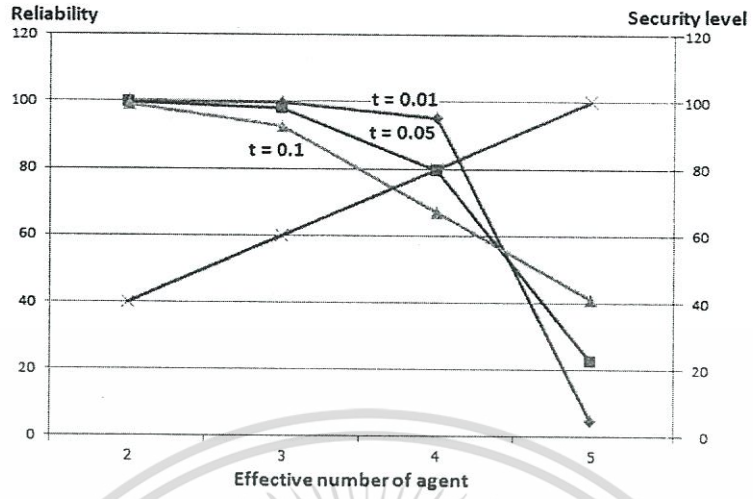
KRA สามารถทำงานได้อย่างต่อเนื่องไม่ล้มเหลว (q) เท่ากับ 99, 95 และ 90 เปอร์เซ็นต์

$q = 0.99, 0.95$ และ 0.9

ค่าความมั่นคงและความน่าเชื่อถือของระบบ สามารถแสดงได้ดังตารางที่ 4.12 และรูปที่ 4.11

ตารางที่ 4.12 ความมั่นคงและความน่าเชื่อถือของระบบ

จำนวน KRA	KRA ล่ม	KRA ที่ให้บริการ	ความมั่นคง (%)	ความน่าเชื่อถือ (%)		
				ที่ t = 0.01	ที่ t = 0.05	ที่ t = 0.1
5	3	2	40	99.99	99.89	99.09
5	2	3	60	99.91	97.86	91.43
5	1	4	80	95.93	79.64	59.27
5	0	5	100	22.62	22.62	22.62



รูปที่ 4.11 ความเหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือของระบบ

การหาจุดที่เหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือของระบบ (จุดตัดของเส้น Reliability กับเส้น Security level) จากกราฟรูปที่ 4.11 ใช้สูตรสมการเส้นตรง

$$y - y_0 = m(x - x_0)$$

เมื่อ (x, y) เป็นจุดใด ๆ บนเส้นตรง

(x_0, y_0) เป็นจุดผ่านของเส้นตรง

m เป็นความชันของเส้นตรง โดยที่

$$m = \frac{y_1 - y_2}{x_1 - x_2} = \frac{y_2 - y_1}{x_2 - x_1}$$

เมื่อ (x_1, y_1) และ (x_2, y_2) เป็นจุดที่เส้นตรงผ่าน 2 จุด

ค่าของจุดตัดความเหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือของระบบสามารถแสดงได้ดังตารางที่ 4.13

ตารางที่ 4.13 ความเหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือของระบบ

ความน่าจะเป็นที่แต่ละ KRA จะล่ม	Effective number of agents	Security level and Reliability (%)
t = 0.01	4.137	82.75
t = 0.05	3.995	79.90
t = 0.10	3.718	74.37

จากรูปที่ 4.11 เมื่อใช้จำนวน KRA ในการกู้คืนกุญแจครบทั้ง 5 เอเจนต์ระบบมีความมั่นคงสูงแต่มีความน่าเชื่อถือต่ำ เนื่องจากระบบมีความเสี่ยงสูงต่อการไม่สามารถให้บริการได้ครบทุกเอเจนต์ เมื่อลดจำนวน KRA ลง กล่าวคือไม่ใช้ KRA ครบทุกเอเจนต์ในการกู้คืนกุญแจ พบว่าระบบมีความมั่นคงลดลงแต่มีความน่าเชื่อถือสูงขึ้น โดยจุดที่เหมาะสมระหว่างความมั่นคงและความน่าเชื่อถือของระบบ ดังแสดงในตารางที่ 4.13 เป็นดังนี้

เมื่อให้ t เท่ากับ 0.01 จุดที่เหมาะสมของความน่าเชื่อถือและความมั่นคง เทียบกับจำนวนเอเจนต์ที่ต้องใช้ในการกู้คืนกุญแจ คือ ต้องใช้ KRA ในการกู้คืนกุญแจจำนวน 4.13 เอเจนต์ หรือประมาณ 4 เอเจนต์ โดยมีระดับความน่าเชื่อถือและระดับความมั่นคงที่ 82.76 เปอร์เซ็นต์

เมื่อให้ t เท่ากับ 0.05 จุดที่เหมาะสมของความน่าเชื่อถือและความมั่นคง เทียบกับจำนวนเอเจนต์ที่ต้องใช้ในการกู้คืนกุญแจ คือ ต้องใช้ KRA ในการกู้คืนกุญแจจำนวน 3.99 เอเจนต์ หรือประมาณ 4 เอเจนต์ โดยมีระดับความน่าเชื่อถือและระดับความมั่นคงที่ 79.9 เปอร์เซ็นต์

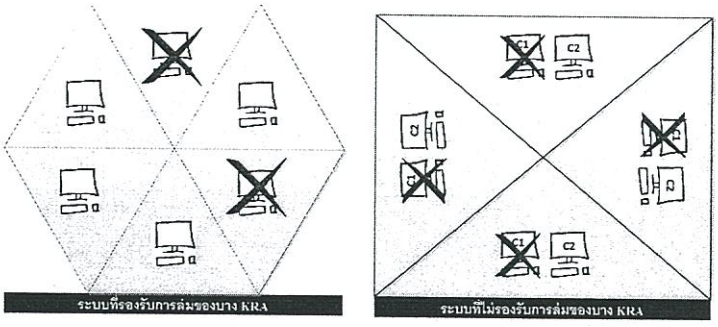
เมื่อให้ t เท่ากับ 0.1 จุดที่เหมาะสมของความน่าเชื่อถือและความมั่นคง เทียบกับจำนวนเอเจนต์ที่ต้องใช้ในการกู้คืนกุญแจ คือ ต้องใช้ KRA ในการกู้คืนกุญแจจำนวน 3.71 เอเจนต์ หรือประมาณ 4 เอเจนต์ โดยมีระดับความน่าเชื่อถือและระดับความมั่นคงที่ 74.37 เปอร์เซ็นต์

4.5 การอภิปรายด้านความต้องการใช้ KRA และความน่าเชื่อถือของระบบเมื่อมีการสำรองการทำงานของ KRA

เมื่อพิจารณาความต้องการใช้จำนวน KRA สำหรับการกู้คืนกุญแจ ระบบที่มีการรองรับการล่มของบาง KRA ไม่จำเป็นต้องมี KRA สำรองเมื่อเอเจนต์ล่ม ส่วนระบบที่ไม่มีการรองรับการล่มของบาง KRA สมมุติให้มีการสำรองเครื่องให้บริการกู้คืนกุญแจอีก 1 เครื่อง (สำรอง 1 ต่อ 1 เครื่อง) เพื่อให้สามารถทำงานแทนได้เมื่อ KRA หลักล่ม นั่นหมายความว่าแต่ละ KRA จะลดความเสี่ยงของความล้มเหลวในการกู้คืนส่วนประกอบของกุญแจลง กล่าวคือเมื่อมีเครื่องให้บริการของ KRA ล่ม 1 เครื่อง ส่วนที่สำรองไว้จะทำงานแทน

ดังนั้นระบบที่ไม่มีการรองรับการล่มของบาง KRA ต้องใช้จำนวน KRA มากกว่าระบบที่มีการรองรับการล่มของบาง KRA ดังแสดงตัวอย่างในรูปที่ 4.11 ซึ่งสมมุติให้ระบบที่ไม่รองรับการล่มของบาง KRA ใช้เอเจนต์ในการกู้คืนกุญแจจำนวน 4 เอเจนต์ แต่ละเอเจนต์มีการสำรอง 1 ต่อ 1 เครื่อง ดังนั้นเพื่อให้ระบบที่รองรับการล่มของบาง KRA มีการใช้จำนวนเอเจนต์ในการกู้คืนเท่ากันจะได้กำหนดให้ใช้จำนวนเอเจนต์ 6 เอเจนต์ แต่ล่ม 2 เอเจนต์ จึงเหลือเอเจนต์ที่ให้บริการจำนวน 4 เอเจนต์ แต่ระบบยังสามารถให้บริการกู้คืนกุญแจได้

ผลการประเมินความต้องการใช้จำนวน KRA สำหรับการกู้คืนกุญแจ และความน่าเชื่อถือของระบบ แสดงได้ดังตารางที่ 4.14



รูปที่ 4.12 การเปรียบเทียบการใช้เครื่องให้บริการการกู้คืนกุญแจของระบบ M-KRS

ตารางที่ 4.14 ความต้องการใช้จำนวน KRA สำหรับการกู้คืนกุญแจ และความน่าเชื่อถือของระบบ

จำนวน KRA (n) ในการกู้คืนกุญแจ	ระบบที่รองรับการล่มของบาง KRA		ระบบที่ไม่รองรับการล่มของบาง KRA	
	จำนวนที่ใช้/ล่ม	ความน่าเชื่อถือ	จำนวนที่ใช้/ล่ม	ความน่าเชื่อถือ
2	4 / 2	98.64	4 / 2	90.50
3	5 / 2	97.85	6 / 3	90.50
4	6 / 2	96.94	8 / 4	90.50
5	7 / 2	95.93	10 / 5	90.50

จากตารางที่ 4.14 เมื่อระบบที่ไม่รองรับการล่มของบาง KRA มีการสำรอง KRA เพิ่มขึ้นอีก 1 ต่อ 1 เอเจนต์ เพื่อให้ทำงานแทนในกรณี KRA หลักล่ม (ให้ล่มได้ 1 ตัว) ตามตารางที่ 4.10 หากนำมาวิเคราะห์ค่าความน่าเชื่อถือพบว่าระบบจะมีความน่าเชื่อถือถึง 90.50 เปอร์เซ็นต์ ซึ่งเพิ่มขึ้น 0.25 เปอร์เซ็นต์

เมื่อทำการเปรียบเทียบกับความน่าเชื่อถือกับระบบที่รองรับการล่มของบาง KRA ซึ่งให้ KRA ล่มได้เท่ากับ n - 2 และให้เหลือจำนวน KRA ที่ใช้ในการกู้คืนกุญแจเท่ากันทั้ง 2 รูปแบบ พบว่าเมื่อใช้ KRA จำนวน 2 ถึง 5 เอเจนต์ ระบบที่รองรับการล่มของบาง KRA มีความน่าเชื่อถือสูงกว่าระบบที่ไม่รองรับการล่มของบาง KRA และใช้จำนวนเครื่องที่ให้บริการน้อยกว่า อย่างไรก็ตาม ความน่าเชื่อถือของระบบที่รองรับการล่มของบาง KRA มีแนวโน้มลดลงเมื่อใช้จำนวน KRA เพิ่มขึ้น

4.6 การอภิปรายด้านความมั่นคงปลอดภัย

งานวิจัยนี้ได้ออกแบบระบบ HADM-KRS และระบบ SHADM-KRS ให้มีกระบวนการจัดการกุญแจ *Ks* ที่เป็นความลับ และผู้ใช้งานมีความเป็นส่วนตัว โดยทำงานอยู่บนโครงสร้างพื้นฐานกุญแจสาธารณะหรือพีเคไอ (Public Key Infrastructure) ดังนั้นการรับ-ส่งข้อมูลระหว่างคู่สื่อสารต้องมีการเข้ารหัสลับข้อมูลด้วยกุญแจสาธารณะของผู้รับ และผู้รับจะถอดรหัสลับข้อมูลด้วยกุญแจส่วนตัวของตนเอง

4.6.1 ความมั่นคงปลอดภัยของฟิลด์ *KRF*

ฟิลด์ *KRF* ถือว่าเป็นส่วนของข้อมูลที่สำคัญและต้องการความมั่นคงสูง การออกแบบกระบวนการทำงานของระบบและโครงสร้างฟิลด์ *KRF* ยังมีผลถึงด้านความเป็นส่วนตัวของผู้ใช้งานอีกด้วย

ภายในฟิลด์ *KRF* จะบรรจุ n x KRF_i ซึ่งจัดเก็บข้อมูลส่วนประกอบของกุญแจ Ks (n x S_i) และแอตทริบิวต์ที่จำเป็นสำหรับการกู้คืนกุญแจ โดยที่แต่ละ KRF_i จะถูกเข้ารหัสด้วยกุญแจสาธารณะ (Ku) ของ KRA_i ($Ku_{agi}[KRF_i]$) เมื่อต้องการกู้คืนกุญแจ *Ks* ผู้ร้องขอจะส่ง $Ku_{agi}[KRF_i]$ ให้กับ KRA_i ผู้ที่จะเปิด $Ku_{agi}[KRF_i]$ นี้ได้ คือ KRA_i เท่านั้น โดยจะถอดรหัส $Ku_{agi}[KRF_i]$ ดังกล่าวด้วยกุญแจส่วนตัว (Kr) ของ KRA_i สำหรับการกู้คืนกุญแจ *Ks* จะต้องอาศัย S_i ทุกตัวที่ได้รับมาจาก KRA_i โดยผู้ที่ทำหน้าที่นี้คือ ผู้ร้องขอการกู้คืนนั่นเอง อย่างไรก็ตามจะสังเกตเห็นว่างานวิจัยนี้ไม่ได้มีการจัดเก็บกุญแจ *Ks* ไว้ใน *KRF* ทำให้ผู้ใช้งานมีความมั่นใจได้ว่ากุญแจจะเป็นความลับและไม่ถูกล่วงรู้โดยบุคคลที่สาม ทำให้ผู้ใช้งานมีความเป็นส่วนตัวสูง

หากผู้ประสงค์ร้ายมีฟิลด์ *KRF* ก็ยากที่จะทำการถอดรหัสเพื่อเอา S_i มาคำนวณหากุญแจ *Ks* ได้ เนื่องจากผู้ประสงค์ร้ายต้องมีกุญแจส่วนตัวของ KRA (Kr_{agi}) ทุกเอเจนต์ที่อยู่ในกลุ่มการกู้คืนกุญแจ และในส่วนของ KRA ซึ่งเป็นผู้ให้บริการกู้คืน S_i ย่อมมีวิธีการจัดการกุญแจ Kr_{agi} ของตนเองให้มีความมั่นคงปลอดภัยสูงสุด

4.6.2 ความมั่นคงปลอดภัยของกุญแจ *Ks*

ความลับของกุญแจ *Ks* จะรู้เฉพาะคู่สนทนาหรือผู้ส่งกับผู้รับเท่านั้น บุคคลที่สามจะไม่สามารถล่วงรู้กุญแจนี้ได้ กล่าวคือกระบวนการคำนวณกุญแจ *Ks* จะเป็นหน้าที่ของผู้ร้องขอการกู้คืนกุญแจ ในที่นี้จะหมายถึงผู้รับหรือหน่วยงานของรัฐที่ต้องการตรวจสอบข้อมูลที่ต้องสงสัยอย่างถูกต้องตามกฎหมาย และกุญแจ *Ks* ไม่ได้ถูกจัดเก็บไว้ในฟิลด์ *KRF* ทำให้กุญแจ *Ks* มีความมั่นคงสูง

ระบบ HADM-KRS และ SHADM-KRS เป็นระบบที่มีการทำงานแบบใช้หลายเอเจนต์กู้คืนกุญแจ ทำให้กุญแจ Ks มีความมั่นคงเพิ่มมากขึ้น โดยจะช่วยลดความเสี่ยงในด้านการผูกขาดกุญแจ การลักลอบขโมยกุญแจ หรือแม้กระทั่งการถอดรหัสลับกุญแจก็ทำได้ยาก

4.6.3 ความมั่นคงปลอดภัยของ S_i

ส่วนประกอบของกุญแจ Ks (S_i) ถูกจัดเก็บอยู่ใน KRF_i ที่มีการเข้ารหัสด้วยกุญแจ Ku ของ KRA_i เมื่อผู้ใช้งานระบบต้องการกู้คืนกุญแจ Ks จึงจะร้องขอการกู้คืน S_i ไปยัง KRA_i ในกระบวนการที่ KRA_i ส่ง S_i มาให้ผู้ร้องขอนั้น จะมีการเข้ารหัสลับ S_i และแอตทริบิวต์สำหรับการพิสูจน์ตัวจริงของ KRA (SGN) ด้วยกุญแจ Ku ของผู้ร้องขอ ($Ku_{req}[S_i \,||\, SGN]$) เมื่อผู้ร้องขอต้องการเปิดฟิลด์ที่บรรจุ S_i จะต้องใช้กุญแจ Kr ของตนเองถอดรหัส

หากผู้ประสงค์ร้ายต้องการ S_i จะต้องถอดรหัส KRF_i ด้วย Kr_{agi} ซึ่งในความเป็นจริง KRA ที่เป็นผู้ให้บริการกู้คืน S_i จะต้องมีวิธีการจัดการกุญแจ Kr_{agi} ของตนเองให้มีความมั่นคงปลอดภัยสูงสุด หรืออีกนัยหนึ่งผู้ประสงค์ร้ายจะต้องถอดรหัส $Ku_{req}[S_i \,||\, SGN]$ ด้วยกุญแจส่วนตัวของผู้ร้องขอการกู้คืนกุญแจเท่านั้น ซึ่งโดยปกติกุญแจ Kr จะถูกจัดเก็บอย่างมั่นคงปลอดภัยโดยเจ้าของกุญแจ

จะสังเกตได้ว่าการที่ระบบทำงานอยู่บนโครงสร้างพื้นฐานกุญแจสาธารณะนั้น จะทำให้เกิดความมั่นคงปลอดภัยต่อกุญแจ Ks และข้อมูล นอกจากนี้ยังทำให้ผู้ใช้งานระบบมีความเป็นส่วนตัวอีกด้วย

4.7 การอภิปรายด้านการโจมตีเอเจนต์และการสมรู้ร่วมคิดกันของเอเจนต์

ระบบการกู้คืนกุญแจแบบรองรับการล่มของบาง KRA มีการสำรองค่าความลับเพื่อการกู้คืนกุญแจในกรณีที่มีบาง KRA ล่ม ซึ่งมีจุดเด่นคือระบบมีความยืดหยุ่นในการกำหนดจำนวนเอเจนต์ขั้นต่ำในการกู้คืนกุญแจ แต่ก็มีจุดอ่อนคือผู้ใช้งานมีความเป็นส่วนตัวลดลงหากมีการโจมตีเอเจนต์และมีการสมรู้ร่วมคิดกันของเอเจนต์ เพื่อให้ได้กุญแจลับ

ในกระบวนการแบ่งและจัดสรรค่า TT_i ของระบบ HADM-KRS เปรียบเสมือนการกระจายความลับ สามารถนำมาอภิปรายถึงจำนวน KRA ที่ต้องโจมตีหรือสมรู้ร่วมคิดเพื่อให้ได้กุญแจลับจากสมการ

$$AF = \frac{\mathrm{n}}{\mathrm{fn}+1}$$

เมื่อ AF คือ จำนวนเอเจนต์ที่ต้องโจมตีหรือสมรู้ร่วมคิดเพื่อให้ได้กุญแจลับ

n คือ จำนวน KRA ที่ใช้ในการกู้คืนกุญแจในกลุ่มการกู้คืน

fn คือ จำนวน KRA ที่ให้ล่มได้

ทั้งนี้ค่า AF หากเป็นจุดทศนิยมให้ปัดขึ้นเป็นจำนวนเต็ม

ตารางที่ 4.15 แสดงการอภิปรายจำนวน KRA ที่ต้องโจมตีหรือสมรู้ร่วมคิดเพื่อให้ได้กุญแจลับ

สมมุติให้ใช้ KRA จำนวน 6 เอเจนต์ และให้มี KRA ล่มได้จำนวน 1 ถึง 4 เอเจนต์ตามลำดับ

ตารางที่ 4.15 การอภิปรายจำนวน KRA ที่ต้องโจมตีหรือสมรู้ร่วมคิดเพื่อให้ได้กุญแจลับ

จำนวน KRA (n)	จำนวน KRA ล่ม (fn)	จำนวน KRA ที่ใช้ได้ (gn)	AF	อภิปราย
6	1	5	3	หากต้องการกุญแจลับต้องโจมตี / สมรู้ร่วมคิดกัน 3 เอเจนต์
6	2	4	2	หากต้องการกุญแจลับต้องโจมตี / สมรู้ร่วมคิดกัน 2 เอเจนต์
6	3	3	1.5	หากต้องการกุญแจลับต้องโจมตี / สมรู้ร่วมคิดกัน 2 เอเจนต์
6	4	2	1.2	หากต้องการกุญแจลับต้องโจมตี / สมรู้ร่วมคิดกัน 2 เอเจนต์

เมื่อเทียบกับระบบการกู้คืนกุญแจแบบที่ไม่รองรับการล่มของบาง KRA เมื่อใช้ KRA จำนวน 6 เอเจนต์ในการกู้คืนกุญแจ ระบบ HADM-KRS ต้องใช้ KRA จำนวน 11 เอเจนต์โดยให้กู้คืนได้ 10 เอเจนต์ นั่นคือสามารถล่มได้ 1 เอเจนต์ การโจมตีหรือการสมรู้ร่วมคิดจึงจะใช้ 6 เอเจนต์เท่ากัน

จากตารางที่ 4.15 จะเห็นว่า เมื่อกำหนดให้มีจำนวน KRA ที่สามารถล่มได้ (fn) ในกลุ่มการกู้คืนจำนวนมากขึ้น มีผลให้การโจมตีและการสมรู้ร่วมคิดกันทำได้ง่ายขึ้น เพราะจะใช้จำนวน KRA น้อยลงก็สามารถกู้คืนกุญแจลับได้ ทั้งนี้การเพิ่มจำนวน fn ทำให้ระบบ HADM-KRS ต้องสำรอง TT_i's มากขึ้นนั่นเอง

เมื่อกล่าวในประเด็นเรื่องความเป็นส่วนตัวของผู้ใช้งาน เมื่อการโจมตีหรือการสมรู้ร่วมคิดกันของ KRA เพื่อให้ได้กุญแจลับทำได้ง่ายโดยอาศัยจำนวน KRA ไม่มาก ทำให้ความเป็นส่วนตัวของผู้ใช้งานน้อยลง

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ระบบการกู้คืนกุญแจแบบหลายเอเจนต์ที่ได้นำเสนอใหม่ในงานวิจัยนี้มีวัตถุประสงค์เพื่อทำการกู้คืนกุญแจ *Ks* ในกรณีที่ผู้รับไม่สามารถใช้งานกุญแจนั้นได้ และสนับสนุนการปฏิบัติตามกฎหมายที่รัฐมีอำนาจในการตรวจสอบข้อมูลที่ต้องสงสัยที่ส่งผ่านระบบเครือข่าย

ระบบการกู้คืนกุญแจทั้งสองแบบที่ได้นำเสนอในงานวิจัยนี้คือ HADM-KRS และ SHADM-KRS เป็นระบบการกู้คืนกุญแจ *Ks* แบบหลายเอเจนต์ที่มีจุดเด่นอยู่ที่กระบวนการทำงานที่ไม่ใช้ศูนย์กลางการกู้คืนกุญแจ (KRC) และระบบมีความพร้อมใช้งานสูง (High Availability M-KRS) กล่าวคือ ระบบสามารถกู้คืนกุญแจได้แม้ในกรณีที่มีบาง KRA ในกลุ่มการกู้คืนล่ม โดยผู้วิจัยได้ออกแบบกระบวนการแชร์ความลับของกุญแจ *Ks* โดยใช้พื้นฐานของทฤษฎีการแชร์ความลับเพื่อนำมาใช้ในขั้นตอนการแบ่งและจัดสรรส่วนประกอบของกุญแจได้อย่างดีและมีประสิทธิภาพ และได้ใช้พื้นฐานทฤษฎีเพาเวอร์เซตมาใช้ในกระบวนการสำรองส่วนประกอบของกุญแจ ทำให้ระบบ HADM-KRS มีความสามารถในการบริหารจัดการจำนวน KRA ที่ใช้ในการกู้คืนกุญแจ *Ks* ให้สอดคล้องกับระดับความมั่นคงปลอดภัยที่ต้องการได้ นอกจากนี้ยังมีคุณสมบัติเด่นอื่น ๆ อีกอาทิ เช่น มีความลับ (Secrecy) ของกุญแจ *Ks* สูง มีฟังก์ชั่นการพิสูจน์ตัวจริงของ KRA ที่อยู่ในกลุ่มการกู้คืน เป็นต้น และระบบยังได้ถูกออกแบบให้มีกระบวนการทำงานที่รองรับการล่มของบาง KRA ในกลุ่มการกู้คืนกุญแจ ทำให้ระบบมีความน่าเชื่อถือสูง (High Reliability M-KRS) อีกด้วย

จากการประเมินสมรรถนะของระบบ HADM-KRS และSHADM-KRS ในด้านกระบวนการทำงาน ความน่าเชื่อถือ และความพร้อมใช้งานของระบบ พบว่า

1) กระบวนการทำงานของระบบ

ระบบ HADM-KRS ใช้เวลาในกระบวนการสร้างฟิลด์ *KRF* และการกู้คืนกุญแจ *Ks* มากกว่า SHADM-KRS เล็กน้อย ทั้งนี้เนื่องจากระบบ HADM-KRS มีคุณสมบัติที่พิเศษกว่า SHADM-KRS กล่าวคือ มีฟังก์ชั่นในการกำหนดจำนวน KRA ขั้นต่ำในการกู้คืนกุญแจ ซึ่งช่วยให้สามารถกำหนดระดับความมั่นคงปลอดภัยของการกู้คืนกุญแจได้ จึงทำให้ HADM-KRS ต้องใช้เวลาในกระบวนการทำงานมากกว่า

เวลาที่ใช้ในการประมวลผลของระบบ M-KRS ขึ้นอยู่กับขนาดของฟิลด์ *KRF* และความซับซ้อนของกระบวนการกู้คืนกุญแจ

2) ความน่าเชื่อถือของระบบ

ระบบ HADM-KRS และSHADM-KRS มีความน่าเชื่อถือของระบบเพิ่มขึ้นตามลำดับอย่างมีนัยสำคัญ เมื่อใช้จำนวน KRA ในการกู้คืนกุญแจมากขึ้น ดังเช่น จากการประเมินที่เมื่อใช้จำนวน KRA 3 เอเจนต์ในการกู้คืนกุญแจ ระบบมีความน่าเชื่อถือ 86.46 % เมื่อเทียบกับการใช้ KRA 6 เอเจนต์ในการกู้คืนกุญแจ ระบบมีความน่าเชื่อถือเพิ่มขึ้นถึง 99.99 % เนื่องจากระบบมีฟังก์ชันสำรองการทำงานของเอเจนต์ ทำให้โอกาสที่ระบบจะล่มลดน้อยลงเมื่อใช้จำนวนเอเจนต์เพิ่มขึ้น

3) ความพร้อมใช้งานของระบบ

ระบบ HADM-KRS และSHADM-KRS มีความพร้อมใช้งานของระบบเพิ่มขึ้นตามลำดับอย่างมีนัยสำคัญ เมื่อใช้จำนวน KRA ในการกู้คืนกุญแจมากขึ้น ดังเช่นจากการประเมินเมื่อใช้จำนวน KRA 3 เอเจนต์ในการกู้คืนกุญแจ ระบบมีความพร้อมใช้งาน 26.97 % เมื่อเทียบกับการใช้ KRA 6 เอเจนต์ในการกู้คืนกุญแจ ระบบมีความความพร้อมใช้งาน 99.83% เนื่องจากระบบมีฟังก์ชันสำรองการทำงานของเอเจนต์ เมื่อใช้จำนวนเอเจนต์เพิ่มขึ้นทำให้เกิดโอกาสที่ระบบจะไม่พร้อมใช้งานลดน้อยลง

5.2 ข้อเสนอแนะ

1) กระบวนการแบ่งและจัดสรรส่วนประกอบของกุญแจ อาจมีการจัดสรรให้แต่ละเอเจนต์กู้คืนส่วนประกอบของกุญแจได้มากกว่าหนึ่งส่วนประกอบ เพื่อเป็นการลดจำนวนเอเจนต์ที่ใช้ในการกู้คืนกุญแจ

2) จำนวนบิตที่อยู่ในฟิลด์ *KRF* มีผลต่อเวลาที่ใช้ในการประมวลผล หากมีการพัฒนาให้ใช้จำนวนบิตที่ลดลง จะทำให้เวลาที่ใช้ในการประมวลผลลดน้อยลงด้วย

3) การแบ่งและการจัดสรรค่า TT_i's ไปไว้ใน KRF_i's เพื่อกู้คืนกุญแจในกรณีที่มีบางเอเจนต์ล่ม มีจุดอ่อนคือเมื่อมีการ โจมตีหรือการสมรู้ร่วมคิดกันของ KRA จะทำให้กุญแจลับถูกเปิดเผยได้ ทำให้ความเป็นส่วนตัวของผู้ใช้งานน้อยลง

4) ควรมีการวิเคราะห์ความน่าเชื่อถือของระบบ HADM-KRS เมื่อเกิดการโจมตีหรือการสมรู้ร่วมคิดกันของ KRA เทียบกับระบบ Typical M-KRS เพื่อทราบจุดอ่อนของการแบ่งและจัดสรรค่าส่วนประกอบของกุญแจลับ ตัวอย่างเช่น

จากตารางที่ 4.15 ระบบ HADM-KRS ใช้ KRA ในการกู้คืนกุญแจ 6 เอเจนต์ สามารถล่มได้ 1 เอเจนต์ จึงมี KRA ที่ทำงานได้ 5 เอเจนต์ และเมื่อเกิดการโจมตีหรือสมรู้ร่วมคิดกันของ KRA ระบบ HADM-KRS จะกู้คืนกุญแจได้โดยใช้ KRA จำนวน 3 เอเจนต์ เปรียบเทียบกับระบบ Typical M-KRS ที่มีระดับความปลอดภัยหรือระดับความเป็นส่วนตัวเท่ากัน คือใช้ KRA จำนวน 3 เอเจนต์ โดยที่ Typical M-KRS สามารถสำรอง KRA ได้อีกเอเจนต์ละ 1 เครื่อง และเครื่องที่สำรองทั้ง 3

เครื่องสามารถล่มได้พร้อมกัน ระบบก็ยังคงให้บริการกู้คืนกุญแจได้ ในขณะที่ระบบ HADM-KRS สามารถล่มได้เพียง 1 เครื่องหรือ 1 เอเจนต์ เมื่อทำการพิจารณาเปรียบเทียบประเด็นด้านความน่าเชื่อถือของทั้ง 2 ระบบ ที่ระดับความเป็นส่วนตัวและความปลอดภัยเท่ากัน คือใช้ KRA จำนวน 3 เอเจนต์ พบว่าระบบ Typical-MKRS สามารถมีจำนวนเครื่องที่ล่มได้มากกว่าระบบ HADM-KRS จึงทำให้ระบบ Typical-MKRS มีความน่าเชื่อถือมากกว่า ทั้งนี้ควรมีการวิเคราะห์เปรียบเทียบให้เห็นในเชิงตัวเลข

5) กระบวนการแบ่งและจัดสรรส่วนประกอบของกุญแจ สามารถนำไปประยุกต์ใช้ในการกู้คืนความลับในระบบอื่น ๆ ได้ ไม่จำกัดเพียงแต่ระบบการกู้คืนกุญแจเท่านั้น

5.3 แนวทางการทำวิจัยในอนาคต

1) พิจารณาประเด็นการโจมตีและการสมรู้ร่วมคิดกันของ KRA เพื่อออกแบบกระบวนการกู้คืนกุญแจและ *KRF* ไม่ให้สามารถกู้คืนกุญแจลับได้เมื่อเกิดกรณีดังกล่าว

บรรณานุกรม

[1] William Stallings. 2011. **Cryptography and Network Security.** 5th ed. Pearson.

[2] D.E. Denning. 1994. "The US Key Escrow Encryption Technology." **Computer Communications**, pp. 453-457.

[3] D.E. Denning and M. Smid. 1994. "Key Escrowing Today." **IEEE Communications Magazine**, pp. 58-68.

[4] National Institute of Standards and Technology. "Escrowed Encryption Standard." **Federal Information Processing Standards Publication (FIPS PUB) 185**, 1994.

[5] S.T. Walker, S.B. Lipner, C.M. Ellison and D.M. Balenson. 1996. "Commercial Key Recovery." **Communications of the ACM**, pp.41-47.

[6] Y.Y. Al-Salqan., "Cryptographic Key Recovery." in **Proceedings of the Computer Society Workshop on Future Trends of Distributed Computing Systems**, pp. 34-37, October 1997.

[7] National Institute of Standards and Technology. "Requirements for Key Recovery Products." **Final Report, Federal Information Processing Standard for Federal Key Management Infrastructure**, 1998.

[8] B.W. McConnell and E.J. Appel. "Enabling Privacy, Commerce, Security and Public Safety in the Global Information Infrastructure." **Report of the Interagency Working Group on Cryptography Policy**, 1996.

[9] Tsuyoshi Nishiokaa, Kanta Matsuuraa, Yuliang Zhengb, and Hideki Imaib. "A Proposal for Authenticated Key Recovery System." in **Proceedings of the Joint Workshop on Information Security and Cryptology**, pp. 1-8, 1997.

[10] Yung-Cheng Lee and Chi-Sung Laih. "On the Key Recovery of the Key Escrow System." in **Proceedings of the Annual Computer Security Applications Conference**, pp. 216-220, 1997.

[11] Cylink. **CyKeyTM: Cylink's Key Recovery Solution.** [Online].Available: http://www.csm.ornl.gov/~dunigan/cykey.pdf, accessed 13 April 2013.

[12] Computer Security Resource Center, National Institute of Standards and Technology. **Key Recovery Examples.** [Online]. Available: http://csrc.nist.gov/krdp/exa.html, accessed 13 April 2013.

[13] Shinyoung Lim, Sangseung Kang and Joochan Sohn. "Modeling of Multiple Agent Based Cryptographic Key Recovery Protocol." in **Proceedings of the Annual Computer Security Applications Conference**, pp. 119-128, December 2003.

[14] Shin-Young Lim, Ho-Sang Hani, Myoung-Jun Kim and Tai-Yun Kim. "In Design of Key Recovery System Using Multiple Agent Technology for Electronic Commerce." in **Proceedings of the Industrial Electronics**, pp. 1351-1356, 2001.

[15] Global Information Assurance Certification. "Encryption Key Recovery." **GSEC Certification Practical Assignment V.1.4b**, 2004.

[16] Eric K. Wang, Joe C.K. Yau, Lucas C.K. Hui, Zoe L. Jiang and S.M. Yiu. "A Key-Recovery System for Long-term Encrypted Documents." in **Proceedings of the International Enterprise Distributed Object Computing Conference Workshops**, 2006.

[17] Johnson R., Rubnich M., and DelaCruz A. "Implementing a Key Recovery Attack on the High-Bandwidth Digital Content Protection Protocol." in **Proceedings of the IEEE Consumer Communications and Networking Conference**, pp. 313-317, 2011.

[18] Su R., Che X., Fu S., Li L., and Zhou L. "Protocol-Based Hidden Key Recovery: IBE Approach and IPSec Case." in **Proceedings of the Conference on Networks Security, Wireless Communications and Trusted Computing**, pp. 719-723, 2009.

[19] National Institute of Standards and Technology. "A Framework for Designing Cryptographic Key Management Systems." **Draft Special Publication 800-130**, 2010.

[20] Kesterson, H.L., III. "Key Recovery and Confidentiality Oops, Where Did I Put That Key?." in **Proceedings of the IEEE Aerospace Conference**, Vol.4, pp. 313-318, Mar 1998.

[21] Numao M. and Nakayama Y. "Internet Archiving Server with Key Recovery Function." in **Proceedings of the Symposium on Cryptography and Information Security**, 1998.

[22] Kundu A., Ghosh N., Chokshi I., and Ghosh S.K. "Analysis of attack graph-based metrics for quantification of network security." in **Proceedings of the 2012 Annual IEEE India Conference**, pp. 530-535, 2012.

[23] Zhang Chao-yang. "DOS Attack Analysis and Study of New Measures to Prevent." in **Proceedings of the International Conference on Intelligence Science and Information Engineering**, pp. 426-429, 2011.

[24] Xiuzhen Chen, Shenghong Li, Jin Ma, and Jianhua Li. "Quantitative threat assessment of denial of serviceattacks on service availability," in **Proceedings of the IEEE International Conference on Computer Science and Automation Engineering**, pp. 220-224, 2011.

[25] Kanyamee K. and Sathitwiriyawong C. "A simple high-availability multiple-agent key recovery system." in **Proceedings of the International Conference for Internet Technology and Secured Transactions**, pp. 1-6, 2009.

[26] Kanyamee K. and Sathitwiriyawong C. "A secure multiple-agent cryptographic key recovery system." in **Proceedings of the IEEE International Conference on Information Reuse & Integration**, pp. 91-96, 2009.

[27] Kanyamee K. and Sathitwiriyawong C. "High-Availability Decentralized Multi-Agent Key Recovery System." in **Proceedings of the Eighth IEEE/ACIS International Conference Computer and Information Science**, pp. 290-294, 2009.

[28] Lv C., Jia X., Tiany L, Jing J., and Suny M. "Efficient Ideal Threshold Secret Sharing Schemes Based on EXCLUSIVE-OR Operations." in **Proceedings of the Fourth International Conference on Network and System Security**, pp. 136-143, 2010.

[29] Jech T. 2006. **Set Theory.** New York: Springer-Verlag.

[30] Future of Privacy Forum. **Encryption, Lawful Access, and Globalization.** [Online]. Available: http://www.futureofprivacy.org/issues/encryption-lawful-access, accessed 13 April 2013.

[31] Kesterson, H.L., III. "Key Recovery and Confidentiality Oops, Where Did I Put That Key?." in **Proceedings of the IEEE Aerospace Conference**, Vol.4, pp. 313-318, Mar 1998.

[32] Dorothy E. Denning and Dennis K. Branstad. 1996. "A Taxonomy for Key Escrow Encryption Systems." **Communications of the ACM**, pp. 34-40.

[33] Dorothy E. Denning and Dennis K. Branstad. 1997. "A Taxonomy for Key Recovery Encryption Systems." **Internet besieged: countering cyberspace scofflaws**, pp. 357-371.

[34] Wikipedia. **Trusted Third Party.** [Online]. Available : http://en.wikipedia.org/wiki/Trusted_third_party, accessed 13 April 2013.

[35] Perlman R. 1999. "An Overview of PKI Trust Models." **IEEE Network**, Vol. 13, issue 6, pp. 38-43.

[36] Guo Z., Okuyama. T and Finley M.R. "A New Trust Model for PKI Interoperability." in **Proceedings of the International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services,** October 2005.

[37] Neuman, B.C. and Ts'o, T. 1994. "Kerberos: an Authentication Service for Computer Networks." **Communications Magazine of the IEEE**, Vol. 32, pp. 32-38.

[38] Paolo D'Arco. "On the Distribution of a Key Distribution Center." in **Proceedings of the Italian Conference on Theoretical Computer Science**, Vol. 2202, pp. 357-369, 2001.

[39] Hunt R. "PKI and Digital Certification Infrastructure." in **Proceedings of the IEEE International Conference on Networks**, October 2001.

[40] Nash A. and et al, 2001. **PKI : Implementing and Managing E-Security.** New York: McGraw-Hill.

[41] Charles Henry Brase and Corrinne Pellillo Brase, 2009. **Understandable statistics: concepts and methods.** 9th ed. U.S.A.: Houghton Mifflin.

ภาคผนวก ก
ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Kanyamee K. and Sathitwiriyawong C. "High-Availability Decentralized Multi-Agent Key Recovery System." in **Proceedings of the Eight IEEE/ACIS International Conference Computer and Information Science**, Shanghai, China, pp. 290-294, 2009.
2. Kanyamee K. and Sathitwiriyawong C. "A secure multiple-agent cryptographic key recovery system." in **Proceedings of the IEEE International Conference on Information Reuse & Integration**, Las Vagus, pp. 91-96, 2009.
3. Kanyamee K. and Sathitwiriyawong C. "A simple high-availability multiple-agent key recovery system." in **Proceedings of the International Conference for Internet Technology and Secured Transactions**, London, pp. 1-6, 2009.
4. Kanyamee K. and Sathitwiriyawong C. January 2014. "High-Availability Decentralized Cryptographic Multi-Agent Key Recovery." To be published in **the International Arab Journal of Information Technology (IAJIT)**, Vol.11, No.1.

2009 Eigth IEEE/ACIS International Conference on Computer and Information Science

High-Availability Decentralized Multi-Agent Key Recovery System

Kanokwan Kanyamee
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
e-mail: kikuit@hotmail.com

Chanboon Sathitwiriyawong
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
e-mail: chanboon@it.kmitl.ac.th

***Abstract*—In symmetric cryptography, any two communicating parties share the secret session key. In case it is unavailable or legal investigation of transmitting messages is needed, there should be a mechanism to recover it. The recovery of session key is typically provided by trusted key recovery agents (KRAs). They will recover the session key after receiving the request from those who have the right to use the key. Key recovery can be achieved by either single agent (S-KRA) or multiple agents (M-KRA). M-KRA enhances the security of S-KRA by reducing any risk of falsification and counterfeiting. This paper proposed a high-availability decentralized multi-agent key recovery system without the need of key recovery center (KRC), called HADM-KRS. The proposed method uses simple and flexible principles of secure session key management with appropriated design of key recovery function and the new format of key recovery field (KRF). The system has high availability, ability to detect attacks on group authentication, and can recover session key despite the failure of some KRAs, without the need of KRC. Therefore, the problem of single point of failure of KRC can be avoided. System administrators also have flexibility to manage and choose the number of KRAs to meet security requirements. The system also supports law enforcement and is based on security mechanism using well defined features of Public Key Infrastructure (PKI).**

Keywords-key recovery; key recovery agent; session key; key recovery field; secret sharing

I. INTRODUCTION

Cryptographic function is an essential component for implementing security and privacy requirements of most network activities. However, the dishonest ones can use cryptosystems to conceal their illegal activities. Therefore, it is necessary to develop a cryptosystem that will meet the requirements of social security while maintaining the protection of user privacy.

Starting in 1993, the US government announced a new encryption technology called Key Escrow System (KES) [1]. The method provides for user privacy in addition to legitimate investigation of any suspected message by government authorities.

In February 1994, the US government announced a standard for Key Escrow System called the Escrow Encryption Standard (EES) [2, 3]. The security of EES depends on the physical protection of tamper-free chips. It supports message accessibility of government authorities but does not emphasize on all aspects of security or privacy.

Later in 1995, Trusted Information System (TIS) presented research on Key Recovery System (KRS) [4] which is developed from KES. KRS can recover the requested session keys without escrowing them. The system recovers the session key by key recovery agent (KRA) [5, 6]. The session key is encapsulated in the key recovery field (KRF) by the sender for later session key recovery as needed. Thus, it can ensure the protection of user privacy.

Since then, there has been continuous improvement of key recovery methods in various areas such as trust [7, 8], authentication [9], key management [10], legal access of data but still of user privacy [11] and enhancement of system security [12, 13]. For instance, most single agent key recovery systems (S-KRS) [4, 7, 9, 11] aim to recover session keys that are lost by users, support law enforcement for message investigation, and consider personal rights of privacy. Furthermore, the system is based on Public Key Infrastructures (PKI) [14, 15, 16] which facilitate more communication security.

As the time passes, S-KRS can be easily attacked. Therefore, many researchers resort to designing multiple agent key recovery system (M-KRS) [12, 13, 17] that can solve various attacks such as brute-force attack and collusion of key recovery agents.

The research of multiple agent model [13] presented fork and join function for key recovery. The collaboration of at least two KRAs is required to recover the session key. Key recovery center (KRC) will act as the coordinating center for all KRAs within the group. A KRF is created for all KRAs. It contains portions of the session key for later key recovery. The sender chooses one or more KRAs among a pool of KRAs and generates a KRF. When the session key recovery service is needed, the KRF is sent to the KRC. Finally, the KRC joins all portions of the session key to obtain the session key.

The M-KRS can provide service by the collaboration of participating KRAs with or without the need of KRC. The latter can reduce the cost of the system and the risk of system unavailability.

A conventional M-KRS called Key Recovery Function [17] provides key recovery service by participating KRAs without the need of KRC. The user must decide which KRAs to use. Then a key recovery block (KRB) is generated from random numbers by using a one-way hash function. KRB contains portions of the session key stored together with the encrypted data. The user can recover the session

978-0-7695-3641-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIS.2009.187

key from KRB by sending KRB to those KRAs. However, some weaknesses persist as follows.

- Third parties have knowledge of the session key.
- The risk of single point of failure from the unavailability of KRC or KRAs.
- All KRAs have to participate in session key recovery since there is no feature of setting the minimum number of KRAs.
- It lacks the function to detect attacks on group authentication of KRAs.
- Finally, the cost of management and maintenance of KRC is high.

This paper presents a high-availability decentralized M-KRS called HADM-KRS. It retains the need of law enforcement and can resolve the problems incurred in the previous M-KRS. The proposed system has the following properties.

a) Secrecy of session key. The session key is known only to the two communicating parties.

b) High system availability. The problem of single point of failure and system bottleneck can be avoided due to decentralized approach. Also, the failure or the absent of some KRAs does not effect the system service.

c) Flexibility. The number of participating KRAs can be specified according to security policies and requirements.

d) Detection of attacks on group authentication. Any KRA that is not in the group can be detected.

e) Cost effective. The management and maintenance cost of KRC can be eliminated.

f) Standard KRF. The standard format of KRF is retained to support law enforcement.

II. TERMINOLOGY

The definitions of the terms used in this paper are as follows.

Key Recovery refers to the means of recovering session keys when it is unavailable for the recipient or government authorities want to inspect the transmitting messages.

Key Recovery Agent (KRA) refers to an agency that is responsible for providing session key recovery service when requested.

Single Key Recovery Agent (S-KRA) refers to a key recovery system that uses only one KRA to recover the session key.

Multiple Key Recovery Agents (M-KRA) refers to a key recovery system that requires the collaboration of at least two KRAs to recover the session key. The number of KRAs is equal to n, where $n \geq 2$.

Key Recovery Field (KRF) refers to a data unit that is used for session key recovery. KRF is encapsulated at the sender's side.

III. PROPOSED M-KRS (HADM-KRS)

This paper presents a high-availability decentralized session key recovery system called HADM-KRS. The certificates for communicating parties and all KRAs are issued by the certificate authority (CA) under the PKI environment. It contains a public key (Ku) and information for identification and authentication. Each KRA can use a trust model based on Gateway CA's (GWCA) [18] that is designed to allow certification to other different kinds of CA located anywhere in the global trust network.

Fig.1 shows the entire processes of HADM-KRS. The processed are described as follows.

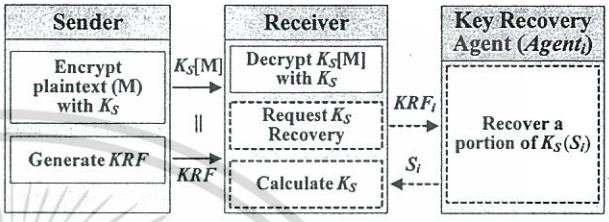


Figure 1. HADM-KRS processes

A. Processes Performed by the Sender

1) Initialization : A session key (K_S) is requested from the Key Distribution Center (KDC) [19] or Kerberos [20].

2) Message encryption : The original message (M) is encrypted with K_S, resulting $K_S[M]$.

3) Preparation to generate KRF : In this process, the components of the KRF is created for the generation and formation of KRF in the successive process. A KRF comprises portions of session key (S_i's), a Share Group Number (SGN), Unique Secret Number of every KRA (TT_i's), and *other information*. S_i's are used to recover K_S. SGN is used to verify the identity of a certain group of KRA. TT_i is used to recover lost S_i. *Other information* comprises public key certificates of the receiver and government authorities to verify their identities. These public keys are used to encrypt S_i.

The processes to create the components of the KRF are shown in Fig. 2, and can be described as follows.

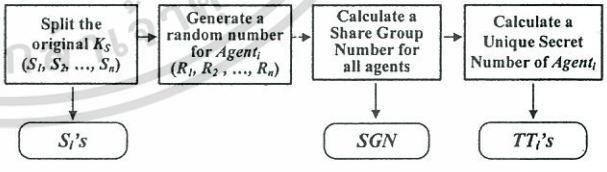


Figure 2. KRF's components preparation

a) The concept of secret sharing [21] is used to split K_S into a number of S_i's equaling the number of participating KRAs, and articulate them by the Exclusive-OR (XOR) operation. This scheme can strengthen the security of the system since keys (and trust) are distributed among several KRAs. The procedures are described as follows.

- Generate n-1 random strings, $S_1, S_2, \ldots, S_{n-1}$, for $Agent_1$ to $Agent_{n-1}$, respectively.
- Calculate S_n for $Agent_n$ by $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus K_S$

b) A random number (R_i) for $Agent_i$, for $i=1$ to n, is generated, giving $R_1, R_2, \ldots, R_n$.

c) A *SGN* is obtained by the XOR operation of all R_i's that is recognized by a particular group of agent. The benefit of creating *SGN* is to detect non-participating agents coming in the group and secure K_S since the *SGN* is unique for a certain key recovery group. *SGN* is calculated as follows.

$$SGN = R_1 \oplus R_2 \oplus \ldots \oplus R_n$$

d) A Unique Secret Number of $Agent_i$ (TT_i) will be used for the recovery of S_i in case $Agent_i$ fails or cannot provide the service. TT_i is calculated as follows.

$$TT_i = S_i \oplus SGN$$

4) Generation and formation of KRF. The proposed system focuses on the security of the session key, user privacy, and the ability to recover the session key correctly and completely despite the failure of some KRAs. These processes can be described as follows.

a) Collect a Unique Secret Number of every neighbor agent of $Agent_i$, for all agents. This method applies the basic mathematical concept of power set and can be defined as follows.

- Choose the number of KRAs in the group (n) to perform session key recovery.
- Choose the minimum number of KRAs (mr) required for successful session key recovery, where mr is not less than two KRAs.
- Calculate the number of KRAs (t) required to distribute TT_i as follows.

$$t = n - mr \quad ; \text{where } t \leq mr$$

- Distribute TT_i to t consecutive nearest neighbor agents of $Agent_i$ or A_i ($TTnag_i$), for $i=1$ to n, as follows.

$$A_i \rightarrow \begin{cases} A_{i+1}, A_{i+2}, \ldots, A_{i+t} & \text{where } i<mr \text{ and } i=mr \\ A_{i+1}, A_{i+2}, \ldots, A_n, A_1, A_2, \ldots, A_j & \text{where } i>mr \text{ and } j=i-mr \\ A_1, A_2, \ldots, A_t & \text{where } i=n \end{cases}$$

b) Every KRF_i for $Agent_i$ is formed and encrypted with the public key of $Agent_i$ (Ku_{agi}) as follows.

$$KRF_i = Ku_{agi}[S_i \,||\, SGN \,||\, TTnag_i \,||\, other\ information]$$

c) *KRF* is then formed by joining all KRF_i's and encrypting them with the public key of the requester (Ku_{req}) as follows.

$$KRF = Ku_{req}[KRF_i\text{'s}]$$

Then, $K_S[M]$ and *KRF* is sent to the receiver (requester).

B. Processes Performed by the Receiver

Upon obtaining $K_S[M]$ and *KRF*, the receiver decrypts $K_S[M]$ with K_S to get the original message and ignores the *KRF*.

In case K_S is unavailable, the receiver requests the session key recovery service as follows.

- Requester decrypts *KRF* with it private key (Kr_{req}) to get all KRF_i's.
- Requester sends KRF_i : $Ku_{agi}[S_i \,||\, SGN \,||\, TTnag_i \,||\, other\ information]$ to $Agent_i$, for $i=1$ to n.

C. Processes Performed by each Key Recovery Agent

The *KRF* is used to recover the required K_S. The session key recovery requires the collaboration of participating KRAs without the need of KRC. The perspective of HADM-KRS is shown in Fig. 3. A partial KRF (KRF_i) is recovered by each KRA ($Agent_i$). This work appropriately designs the functions of HADM-KRS that can enhance user privacy, reduce system response time, and eliminate cost of KRC administration and management. The processes of session key portions recovery are shown in Fig. 4 to 5.

1) When the recovery of session key is requested. A partial session key recovery at each KRA is performed as follows.

- $Agent_i$ decrypts KRF_i with its private key (Kr_{agi}) to obtain S_i, *SGN*, $TTnag_i$, and *other information*.
- $Agent_i$ verifies *SGN* and public key certificate of the requester.
- $Agent_i$ encrypts S_i and *SGN* with public key of the requester (Ku_{req}).
- $Agent_i$ sends $Ku_{req}[S_i \,||\, SGN]$ to the requester for the compilation and construction of K_S.

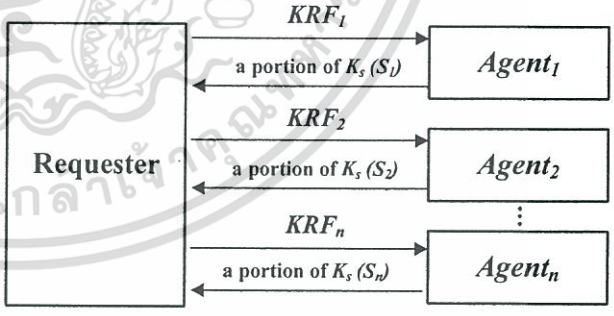


Figure 3. The perspective of HADM-KRS

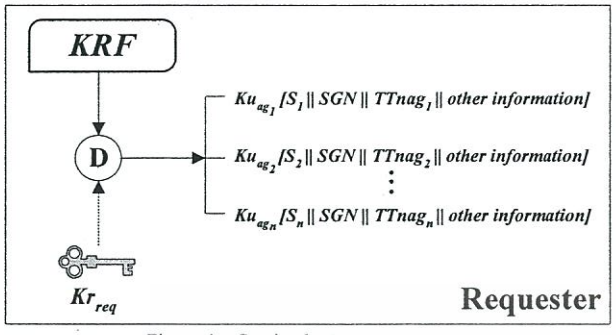


Figure 4. Session key recovery request

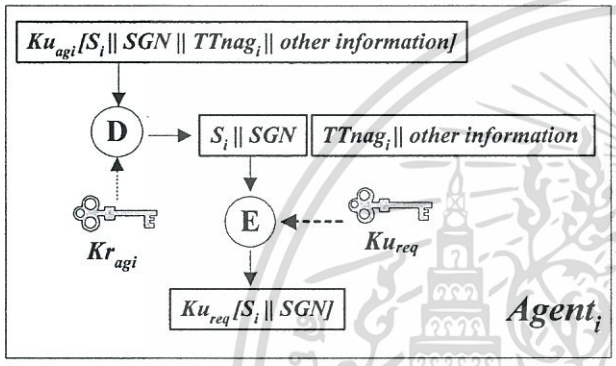


Figure 5. Partial session key recovery process by each KRA

2) In case some KRAs are out of service. The requester cannot collect all S_i. In this case, the requester manages the process as follows.

- Requester checks S_i that is not received from $Agent_i$.
- Requester encrypts the request ($req\text{-}S_i$) and SGN with the public key of next neighbor KRA (Ku_{nxt} of $Agent_{nxt}$) that can provide key recovery service to obtain $Ku_{nxt}[req\text{-}S_i \| SGN \| other\ information]$.
- Requester sends $Ku_{nxt}[req\text{-}S_i \| SGN]$ to $Agent_{nxt}$ that can provide key recovery service.
- $Agent_{nxt}$ decrypts $Ku_{nxt}[req\text{-}S_i \| SGN \| other\ information]$ with its private key (Kr_{nxt}).
- $Agent_{nxt}$ verifies SGN, public key certificate of the requester, and calculates S_i as follows.

$$S_i = TT_i \oplus SGN$$

- $Agent_{nxt}$ encrypts S_i and SGN with Ku_{req} to obtain $Ku_{req}[S_i, SGN]$.
- $Agent_{nxt}$ forwards $Ku_{req}[S_i, SGN]$ to the requester.

D. Session Key Construction Performed by the Requester

In this phase, K_S is constructed by the requester. This method shares the work of $Agent_i$ and makes sure that K_S is secure and private. The construction of K_S is shown in Fig. 6 and defined as follows.

- Requester decrypts $Ku_{req}[S_i \| SGN]$ with its private key (Kr_{req}) to obtain S_i and SGN.
- S_i of $Agent_i$ is verified using SGN.
- K_S is calculated by $S_1 \oplus S_2 \oplus .... \oplus S_n$.

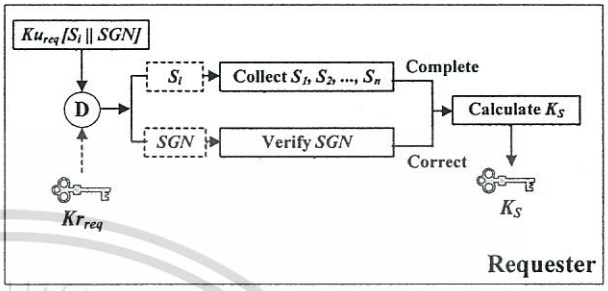


Figure 6. Session key recovery process by the requester

IV. Comparative Evaluation of Various M-KRS's

The results of a preliminary comparative evaluation of various M-KRS's without KRC are shown in Table 1.

Compared to the conventional Key Recovery Function [18], HADM-KRS has all aspects of capabilities including high secrecy of the session key, ability to avoid single point of failure, ability to detect attacks on group authentication, and flexibility to manage the number of KRAs. The summarization is presented as follows.

- HADM-KRS has an appropriate function of high secrecy of the session key and support law enforcement by using the suitable structure of new KRF.
- The system can recover K_S even the failure of some KRAs by using the concept of secret sharing and power set and it can also avoid the problem of single point of failure that cannot be solved by the Key Recovery Function.
- HADM-KRS has strong authentication function for detection and verification of non-participating KRAs by using SGN.
- Finally, HADM-KRS has the flexibility to manage the number of participating KRAs according to security requirements and policies much better than the Key Recovery Function.

TABLE I. Preliminary Comparison of HADM-KRS with Conventional Key Recovery Function

Capabilities	Key recovery function	HADM-KRS
High secrecy of K_S	Yes	Yes
K_S can be recovered despite the failure of some KRAs	No	Yes
Attack detection on group authentication	Medium	High
Flexibility to manage the number of KRAs	Low	High

V. ALGORITHM PERFORMANCE ANALYSIS

The efficiency of the proposed algorithm is considered by examining time complexity. Time complexity is the time (T) of the process running that has the number of inputs (n) equaling to $T(n)$. Thus, it means the required time to process the algorithm depends on the number of inputs that is equal to the number of KRAs. This can be measured with big O or the order of magnitude of this algorithm is $O(n)$.

Therefore, in creating the KRF, when the input those among the number of KRAs are bigger, the T value increases in relation to n. Therefore, it can be concluded that $T(n)=O(n)$.

VI. CONCLUSIONS

This paper proposes a high-availability decentralized session key recovery system called HADM-KRS. The system considers all aspects of security including confidentiality, integrity, and availability. The principal of session key recovery relies on the collaboration of KRAs, the flexibility to manage the number of KRAs, the high secrecy of session key, and the ability to detect KRAs that are not within the same group. Furthermore, HADM-KRS has appropriate function and confidentiality on the distribution of session key portions for participating KRAs without the need of KRC. Therefore, the management cost of HADM-KRS can be reduced. When a KRA is unable to recover a portion of the session key, its next neighbor KRA is assigned to recover the session key instead. Consequently, the system will be constantly available. The structure of KRF emphasizes the security of session key, the privacy of user, as well as the provision of law enforcement.

REFERENCES

[1] D.E. Denning. "The US Key Escrow Encryption Technology," Computer Communications, Vol. 17, No. 7, pp. 453-457, July 1994.

[2] D.E. Denning and M. Smid. "Key Escrowing Today," IEEE Communications Magazine, Vol. 32, Issue 9, pp. 58-68, September 1994.

[3] National Institute of Standards and Technology, "Escrowed Encryption Standard," Federal Information Processing Standards Publication (FIPS PUB) 185, February 1994.

[4] S.T. Walker, S.B. Lipner, C.M. Ellison and D.M. Balenson. "Commercial Key Recovery," Communications of the ACM, Vol. 39, No. 3, pp. 41-47, March 1996.

[5] D.E. Denning and D.K. Branstad. "A Taxonomy for Key Recovery Encryption Systems," Internet Besieged: Countering Cyberspace Scofflaws, pp. 357-371, 1997.

[6] N. Jefferies, C. Mitchell and M. Walker. "A Proposed Architecture for Trusted Third Party Services," Proceedings of the International Conference on Cryptography: Policy and Algorithms, Vol. 1029, pp. 98-104, 1995.

[7] Cylink. CyKey™: Cylink's Key Recovery Solution. [Online]. Available: http://www.csm.ornl.gov/~dunigan/cykey.pdf, March 1997.

[8] Computer Security Resource Center, National Institute of Standards and Technology. Key Recovery Examples. [Online]. Available: http://csrc.nist.gov/krdp/exa.html, 1996.

[9] Y.Y. Al-Salqan. "Cryptographic Key Recovery," Proceedings of the 6th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems (FTDCS' 97), pp. 34-37, October 1997.

[10] B.W. McConnell and E.J. Appel. Enabling Privacy, Commerce, Security and Public Safety in the Global Information Infrastructure. [Online]. Available: http://www.cdt.org/crypto/clipper_III/clipper_III_draft.html, May 1996.

[11] Yung-Cheng Lee and Chi-Sung Laih. "On the Key Recovery of the Key Escrow System," Proceedings of the 13th Annual Computer Security Applications Conference (ACSAC '97), pp. 216-220, 1997.

[12] Shinyoung Lim, Sangseung Kang and Joochan Sohn. "Modeling of Multiple Agent Based Cryptographic Key Recovery Protocol," Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003), pp. 119-128, December 2003.

[13] Shin-Young Lim, Ho-Sang Hani, Myoung-Jun Kim and Tai-Yun Kim. "Design of Key Recovery System Using Multiple Agent Technology for Electronic Commerce," Proceedings of 2001 IEEE Industrial Electronics, Vol. 2, pp. 1351-1356, 2001.

[14] R. Perlman. "An Overview of PKI Trust Models," IEEE Network, Vol. 13, Issue 6, pp. 38-43, November 1999.

[15] A. Nash, W. Duane, C. Joseph and D. Brink. PKI: Implementing and Managing E-Security, RSA Press, McGraw-Hill, Inc. 2001.

[16] R. Hunt. "PKI and Digital Certification Infrastructure," Proceedings of the 9th IEEE International Conference on Networks, pp. 234-239, October 2001.

[17] M. Numao and Y. Nakayama. "Internet Archiving Server with Key Recovery Function," 1998 Symposium on Cryptography and Information Security, 1998.

[18] Z. Guo, T. Okuyama and M. Finley. "A New Trust Model for PKI Interoperability," Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS 2005), October 2005.

[19] Paolo D'Arco. "On the Distribution of a Key Distribution Center," Proceedings of the 7th Italian Conference on Theoretical Computer Science, Vol. 2202, pp. 357-369, 2001.

[20] B.C. Neuman and T. Ts'o. "Kerberos: An Authentication Service for Computer Networks," IEEE Communications Magazine, Vol. 32, Issue 9, pp. 32-38, September 1994.

[21] A. Shamir. "How to Share a Secret." Communications of the ACM, Vol. 22, No. 11, pp. 612-613, November 1979.

A Secure Multiple-Agent Cryptographic Key Recovery System

Kanokwan Kanyamee and Chanboon Sathitwiriyawong
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
E-mail: kikuit@hotmail.com, chanboon@it.kmitl.ac.th

Abstract

Symmetric cryptography uses the same session key for message encryption and decryption. Without having it, the encrypted message will never be revealed. In case the session key is unavailable or government authorities need to inspect suspect messages, there should be a mechanism to recover it. The recovery of session key is usually provided by a trusted key recovery center as a coordinator among key recovery agents (KRAs). The session key will be recovered on receiving the request from those who are legitimate to view the message. Key recovery can be achieved by a single agent or multiple agents. The latter can enhance the security of the former by mitigating the risks of fabrication and collusion. This paper presents a secure multiple-agent cryptographic key recovery system (SEM-KRS) that uses the simple and flexible principles of secure session key management with appropriated design of key recovery function and the new format of key recovery field. The proposed system has high availability, ability to detect attacks on group authentication, and can recover session key despite the failure of some KRAs. Therefore, the problem of single point of failure can be avoided. System administrators also have flexibility to manage and choose the number of KRAs to meet security requirements. The system also supports law enforcement, and is based on Public Key Infrastructure to provide trusted and authenticated key distribution infrastructure.

Keywords: Key Recovery, Session Key, Key Recovery Agent, Key Recovery Center, Secret Sharing

1. Introduction

The number of Internet users has been growing rapidly over the past many years. Network communications are applied in various activities such as E-Government, E-Commerce, and E-Education. Cryptography technology will help to strengthen security and privacy of these network activities. However, the dishonest ones can also use cryptosystems to conceal their illegal activities, which will endanger the social security. Hence it is necessary to develop a cryptosystem that meets the requirements of social security while maintaining user privacy.

Starting in 1993, the US government announced a new encryption technology called Key Escrow System (KES) [1]. This method provides for the privacy of users in addition to legal investigation of any suspected message by government authorities.

In 1994, the US government announced a standard for Key Escrow System called the Escrow Encryption Standard (EES) [2] [3]. The security of EES depends on the physical protection of tamper-free chips. It supports message accessibility of government authorities and does not emphasize on every aspect of security and privacy.

Later in 1995, Trusted Information System (TIS) presented research on TIS Key Recovery System [4] which was developed from KES. Key Recovery System (KRS) can recover requested session keys without escrowing them. The system recovers the session key by Key Recovery Agent (KRA) [5] [6]. The session key is encapsulated in the Key Recovery Field (*KRF*) by the sender for later session key recovery as needed. Thus, it ensures the protection of user privacy.

Since then, there has been continuous improvement of key recovery methods by adding and improving various areas, such as trust [7] [8], authentication [9], key management [10], legal access of data but still of users' privacy [11], and system security enhancement [12] [13]. For instance, most single-agent key recovery systems (S-KRS) [4] [7] [9] [11] aim to recover lost session keys, support law enforcement for message investigation, and consider personal rights of privacy. Furthermore, the system is based on Public Key Infrastructure (PKI) [14] [15] [16] to provide trusted and secured key distribution infrastructure.

As the time passes, S-KRS can be easily attacked. Therefore, many researchers resort to designing multiple-agent key recovery system (M-KRS) [12] [13] [17] that can resist various threats such as brute-force attack and collusion of key recovery agents.

The conventional M-KRS called Key Recovery Function [17] provides a key recovery service by participating KRAs. The user must decide which KRAs to use and generates key recovery block (KRB) from random numbers using a one-way hash function. KRB contains all portions of the session key stored with the encrypted data. When the session key is lost or damaged,

IEEE IRI 2009, July 10-12, 2009, Las Vegas, Nevada, USA
978-1-4244-4116-7/09/$25.00 ©2009 IEEE

the user can recover the session key from KRB by sending KRB to those KRAs.

Later, the research of multiple agent model [13] presented fork and join function for key recovery. The collaboration of at least two KRAs is required to recover the session key. Key recovery center (KRC) will act as the coordinating center for all KRAs in the group. *KRF* is created for all KRAs. It contains portions of the session key for later key recovery. The sender chooses one or more KRAs among a pool of KRAs and generates *KRF*. When session key recovery is needed, *KRF* is sent to KRC. Finally, KRC joins portions of the session key to obtain the session key.

However, many weaknesses persist as follows. (1) Third parties have knowledge of the session key. (2) The risk of single point of failure from the unavailability of some KRAs since every KRA has to participate in the recovery of session key. (3) There is no feature to set the minimum number of KRAs for successful key recovery. (4) Finally, the absence of attack detection functions on group authentication of KRAs.

This paper presents a secure multiple-agent cryptographic key recovery system called SEM-KRS. It retains the need of law enforcement and can resolve the problems incurred in the previous M-KRS. The proposed system has the following properties. (1) *Secrecy of session key*. The session key is known only to the two communicating parties. (2) *High system availability*. The problem of single point of failure or the absent of some KRAs does not effect the system service. (3) *Flexibility*. The number of participating KRAs can be chosen to meet security policies and requirements. (4) *Detection of attacks on group authentication*. KRA that is not in the group can be detected. (5) *Standard KRF*. The standard format of *KRF* is retained to support law enforcement.

2. Terminology

The following are the definitions used in this paper.

Key Recovery refers to the means of recovering session keys when the key of the recipient is lost or damaged and government authorities want to verify the suspicious transmitting messages.

Key Recovery Agent (KRA) refers to an agency that is responsible for providing session key recovery service when requested.

Single-Agent Key Recovery System (S-KRS) refers to a key recovery system that uses only one KRA.

Multiple-Agent Key Recovery System (M-KRS) refers to a key recovery system that requires the collaboration of at least two KRAs.

Key Recovery Field (KRF) refers to an encapsulated data field at the sender side for session key recovery.

3. Proposed M-KRS (SEM-KRS)

This paper presents a SEM-KRS. The certificates for communicating parties, KRC, and all KRAs are issued by the certificate authority (CA) under the PKI environment. It contains a public key (*Ku*) and information for identification and authentication.

Fig.1 shows the entire processes of SEM-KRS that can be described as follows.

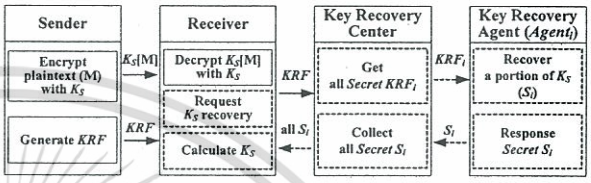


Figure 1. SEM-KRS processes

3.1. Processes performed by the sender

3.1.1. Initialization phase. A session key (K_S) is requested from the Key Distribution Center (KDC) [18] or Kerberos [19].

3.1.2. Message encryption phase. The original message (M) is encrypted with K_S, which is shared between the sender and the receiver, resulting $K_S[M]$.

3.1.3. KRF's components preparation phase. In this phase, the components of a *KRF* are generated. The number of participating KRAs is equal to n. The *KRF* comprises all portions of K_S (S_i's), a Share Group Number (*SGN*), a Unique Secret Number of every KRA (TT_i's), and *other information*. All S_i's are used to recover K_S. *SGN* is used to verify the identity of a certain group of KRA. Each TT_i is used to recover S_i due to the failure of *Agent*$_i$. *Other information* is used for identification and authentication.

The preparation procedures to create the components of a *KRF* are shown in Fig. 2, and are described in details as follows.

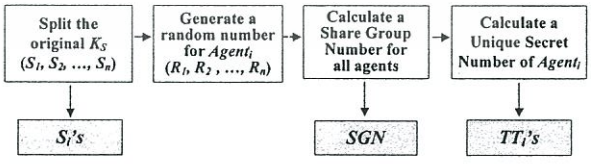


Figure 2. Preparation of KRF's components

1) The secret sharing scheme [20] is used by splitting K_S into a number of partial K_S's (S_i's) equaling the number of participating KRAs. They are articulated with

Exclusive-OR (XOR) operation. This scheme can strengthen the security of the system since keys (and trust) are distributed among several KRAs. The procedures are described as follows.

- Generate *n-1* random strings, S_1, S_2, ..., S_{n-1}, for $Agent_1$ to $Agent_{n-1}$, respectively.
- Calculate S_n for $Agent_n$ by $S_1 \oplus S_2 \oplus \cdots \oplus S_{n-1} \oplus K_S$

Since S_i is a portion of K_S owned by $Agent_i$, K_S cannot be recovered when any S_i is lost or damaged.

2) A random number (R_i) for $Agent_i$, for $i=1$ to n, is generated, giving $R_1, R_2, ..., R_n$.

3) A *SGN* is obtained by the XOR operation of all R_i's. A *SGN* is unique to a particular group of agent and is calculated as follows.

$$SGN = R_1 \oplus R_2 \oplus \cdots \oplus R_n$$

The benefit of creating a *SGN* is to detect non-participating agents coming to the group to secure K_S since *SGN* is unique to a particular key recovery group. When KRC detects an unknown *SGN*, the system is regarded as compromised.

4) A Unique Secret Number of $Agent_i$ (TT_i) will be used for the recovery of S_i in case $Agent_i$ fails or cannot provide the service. TT_i is calculated as follows.

$$TT_i = S_i \oplus SGN$$

3.1.4. Generation and formation of a *KRF* phase. The proposed system focuses on the security of the session key, privacy of users and the ability to recover the session key correctly and completely despite the failure of some KRAs. These processes are described as follows.

1) Collect a Unique Secret Number of every neighbor agent of $Agent_i$, for all agents. This method applies the basic mathematical concept of power set. The processes are described as follows.

- Choose the number of KRAs in the group (n) to perform session key recovery.
- Choose the minimum number of KRAs (mr) required for successful session key recovery, where mr is not less than two KRAs.
- Calculate the number of KRAs required to distribute TT_i (t) as follows.

$$t = n - mr \; ; \quad where \; t \leq mr$$

- Distribute TT_i to t consecutive nearest neighbor agents of $Agent_i$ (A_i), for $i=1$ to n, as follows.

$$A_i \rightarrow \begin{cases} A_{i+1}, A_{i+2}, ..., A_{i+t} & \text{where } i<mr \text{ and } i=mr \\ A_{i+1}, A_{i+2}, ..., A_n, A_1, A_2, ..., A_j & \text{where } i>mr \text{ and } j=i-mr \\ A_1, A_2, ..., A_t & \text{where } i=n \end{cases}$$

2) Every KRF_i for $Agent_i$ is formed and encrypted with the public key of $Agent_i$ (Ku_{agi}) as follows.

$$KRF_i = Ku_{agi} [S_i \,||\, SGN \,||\, TT_{neighbor\ agents\ of\ Agent_i}]$$

3) *KRF* is then formed by encrypting all KRF_i's and *other information* with the public key of KRC (Ku_{KRC}) as follows.

$$KRF = Ku_{KRC} [KRF_i's \,||\, other\ information]$$

Then, $K_S[M]$ and *KRF* is sent to the receiver.

3.2. Processes performed by the receiver

On receiving $K_S[M]$ and *KRF*, the receiver can decrypt $K_S[M]$ with K_S, while the *KRF* is neglected. The *KRF* will only be used in case K_S is lost.

In case K_S is unavailable, the receiver requests the session key recovery service by sending the *KRF* to KRC.

3.3. Processes performed by KRC and all KRAs

This process requires the collaboration of KRC and all participating KRAs. The function of KRC is appropriately designed to enhance user privacy and reduce system response time. The KRC has been designed to collect only S_i's, but not K_S. The overview illustration of SEM-KRS is shown in Fig. 3. Fig. 4 depicts the preliminary session key recovery process performed by KRC. The subsequent partial session key recovery process performed by each agent is shown in Fig. 5. The entire processes can be described as follows.

1) KRC decrypts *KRF* with its private key (Kr_{KRC}) to obtain n portions of *KRF* (Ku_{agi} [KRF_i]'s).

2) KRC forwards $Ku_{agi}[S_i \,||\, SGN \,||\, TT_{neighbor\ agents\ of\ Agent_i}]$ to $Agent_i$.

3) $Agent_i$ decrypts it with Kr_{agi} to obtain S_i, *SGN* and $TT_{neighbor\ agents\ of\ Agent_i}$.

4) $Agent_i$ encrypts S_i with the public key of requester (Ku_{req}), attaches *SGN*, and encrypts them with Ku_{KRC}.

5) $Agent_i$ forwards $Ku_{KRC}[Ku_{req}[S_i] \,||\, SGN]$ to KRC.

6) KRC decrypts it with Kr_{KRC} to obtain $Ku_{req}[S_i]$ and *SGN*.

7) KRC verifies *SGN*.

8) Finally, when KRC completes the gathering of all $Ku_{req}[S_i]$, they are forwarded to the requester for the assembly of K_S.

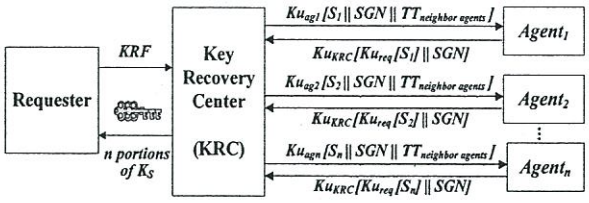


Figure 3. Overview illustration of SEM-KRS

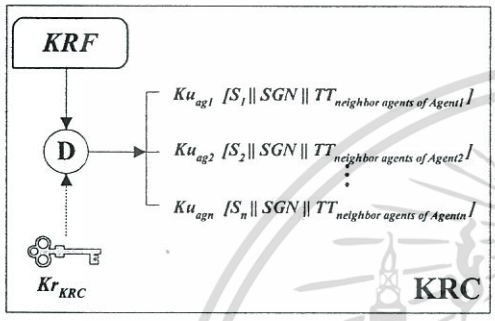


Figure 4. Preliminary session key recovery process performed by KRC

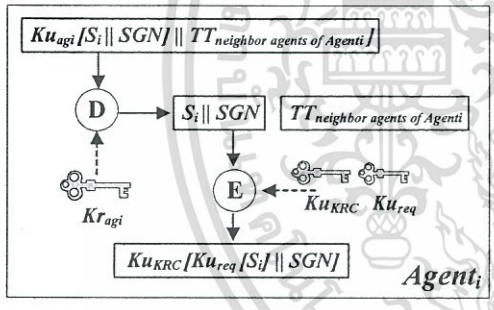


Figure 5. Partial session key recovery process by $Agent_i$

In case of the failure of some KRAs, KRC cannot collect all $Ku_{req}[S_i]$. In this case, KRC manages the process as follows.

1) KRC checks S_i that is not received from $Agent_i$.

2) KRC encrypts the request (req_S_i) and SGN by the public key of next neighbor KRA (Ku_{nxt} of $Agent_{nxt}$) which can provide key recovery service to obtain $Ku_{nxt}[req\_S_i \| SGN]$.

3) KRC sends $Ku_{nxt}[req\_S_i \| SGN]$ to $Agent_{nxt}$ that can provide key recovery service.

4) $Agent_{nxt}$ decrypts $Ku_{nxt}[req\_S_i \| SGN]$ with its private key (Kr_{nxt}).

5) $Agent_{nxt}$ verifies SGN and calculates S_i as follows.

$$S_i = TT_i \oplus SGN$$

6) $Agent_{nxt}$ encrypts S_i and SGN by Ku_{req} and encrypts them again by Ku_{KRC} to obtain $Ku_{KRC}[Ku_{req}[S_i] \| SGN]$.

7) Finally, $Agent_{nxt}$ forwards $Ku_{KRC}[Ku_{req}[S_i] \| SGN]$ to KRC to get $Ku_{req}[S_i]$.

3.4. Processes performed by the requester

Session key recovery phase. In this phase, K_S is constructed by the requester. This method shares the work of KRC and makes sure that K_S is secure and private. The construction of K_S is shown in Fig. 6 and defined as follows.

1) Requester decrypts $Ku_{req}[S_1]$, $Ku_{req}[S_2]$, ..., $Ku_{req}[S_n]$ with its private key (Kr_{req}).

2) K_S is calculated by $S_1 \oplus S_2 \oplus .... \oplus S_n$.

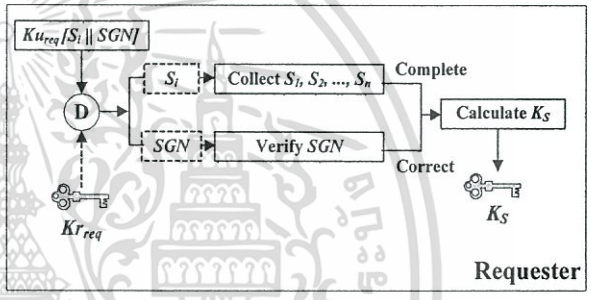


Figure 6. Session key recovery process by requester

4. Preliminary Comparison of M-KRS's

The results of a preliminary comparative evaluation of various M-KRS's are shown in Table 1.

Compared to the conventional Key Recovery Function [17] and Multiple Agent Model [13], SEM-KRS has all aspects of capabilities including high secrecy of the session key, secret sharing of K_S, ability to avoid single point of failure, ability to detect attacks on group authentication, flexibility to manage the number of KRAs, while maintaining law enforcement support. The summarization is presented as follows.

1) SEM-KRS has an appropriate function of high secrecy of K_S and support law enforcement by using the suitable structure of new *KRF*.

2) The system can recover K_S even the failure of some agents by using the concept of secret sharing and power set. It can also avoid the problem of single point of failure that cannot be solved by other methods.

3) SEM-KRS has strong authentication function for detection and verification of non-participating KRAs by using *SGN*.

4) Finally, SEM-KRS has the flexibility to manage the number of participating KRAs to meet security policies and requirements much better than other methods.

Table 1. Preliminary comparison of various M-KRS's

Capabilities	Key Recovery Function	Multiple Agent Model	SEM-KRS
High secrecy of K_S	Yes	No	Yes
Secret sharing of K_S	No	Yes	Yes
Ability to recover K_S despite the failure of some KRAs	No	No	Yes
Attack detection on group authentication	No	No	Yes
Flexibility to manage the number of KRAs	Low	Low	High
Law enforcement support	Medium	High	High

The capability of SEM-KRS is not only higher than other M-KRS's, but also more robust than other methods.

5. Performance Evaluation of SEM-KRS

Two measurement experiments were conducted to determine the processing time (in seconds) during the generation of a *KRF* and the recovery of K_S, in terms of the number of agents. The processing time was measured when using a Genuine Intel® CPU, 794 MHz with 1 GB of RAM. The results are shown in Fig. 7 and Fig. 8, respectively. For the generation of *KRF*, the processing time increases more rapidly when the number of agents is more than 10. It demonstrates that the additional processing time of SEM-KRS does not incur numerous processing time, especially when the number of agents is less than 10. For the recovery of K_S, the processing time is almost constant when the number of agents is greater than 4.

SEM-KRS requires more processing time than Multiple Agent Model because of the distribution of unique secret numbers for session key recovery in case of the failure of some KRAs. The extra time added is not significant for many applications, knowing that SEM-KRS is more robust than Multiple Agent Model in terms of secrecy, availability and flexibility. However, the difference of processing time between SEM-KRS and Multiple Agent Model is negligible.

The last measurement experiment was conducted to determine the processing time required for the recovery of lost S_i's due to the failure of $Agent_i$'s.

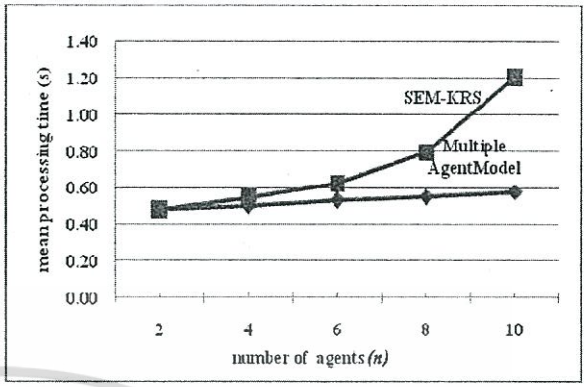


Figure 7. Performance of KRF generation

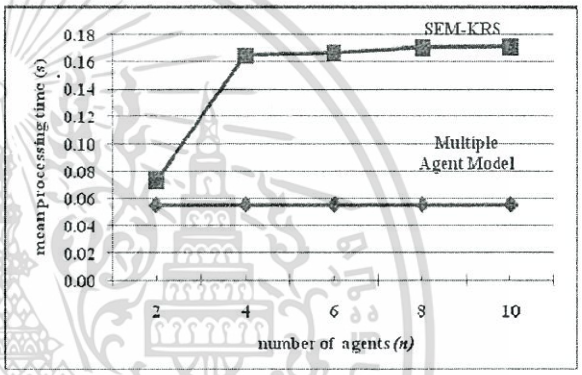


Figure 8. Performance of session key recovery

Table 2 shows the results of S_i's recovery times (in microseconds) when one or more agents fail to provide their S_i's for session key recovery service. In the measurement experiment, the number of participating agents was set to 3, 4, and 5, respectively. Since at least two agents are required to perform session key recovery in the multiple agent key recovery system, the number of failed agents was set to at most 1, 2, and 3, respectively. The result shows that the processing time increases slowly in relation to the number of failed agents.

Table 2. Performance of S_i's recovery

Number of agents (*n*)	Number of failed agents	Recovery time of lost S_i's (μs)
3	1	79.83
4	1	79.83
	2	81.12
5	1	79.83
	2	81.12
	3	89.56

6. Conclusions

This paper proposes a secure multiple-agent cryptographic key recovery system (SEM-KRS) that considers all aspects of security including confidentiality, integrity and availability. The proposed system can solve the single point of failure problem effectively. The principal of session key recovery relies on the collaboration of KRAs, the flexibility to manage the number of KRAs, the high secrecy of K_S and the ability to detect KRAs that are not legitimate participants. Furthermore, KRC has an appropriate function and confidentiality on the distribution of session key portions to all participating KRAs. When a KRA is defective or unable to recover a portion of the session key, its next neighbor KRA is assigned to recover the session key. The functions of KRC can enhance user privacy and reduce system response time. Consequently, the system will be constantly available. The structure of *KRF* emphasizes the security of session key, privacy of user and system, as well as the provision of law enforcement. For the future research work, the quality of SEM-KRS in terms of both performance and reliability will be thoroughly evaluated in comparison to the existing multiple agent model.

The preliminary measurement experiments prove that the proposed SEM-KRS do not incur numerous processing time as compared to the existing multiple agent model, with a reasonable number of agents of less than ten agents.

7. References

[1] D.E. Denning, "The US Key Escrow Encryption Technology", *Computer Communications*, Vol. 17, No. 7, July 1994, pp. 453-457.

[2] D.E. Denning and M. Smid, "Key Escrowing Today", *IEEE Communications Magazine*, Vol. 32, Issue 9, September 1994, pp. 58-68.

[3] National Institute of Standards and Technology, "Escrowed Encryption Standard", *Federal Information Processing Standards Publication (FIPS PUB) 185*, February 1994.

[4] S.T. Walker, S.B. Lipner, C.M. Ellison and D.M. Balenson, "Commercial Key Recovery", *Communications of the ACM*, Vol. 39, No. 3, March 1996, pp. 41-47.

[5] D.E. Denning and D.K. Branstad, "A Taxonomy for Key Recovery Encryption Systems", *Internet Besieged: Countering Cyberspace Scofflaws*, 1997, pp. 357-371.

[6] N. Jefferies, C. Mitchell and M. Walker, "A Proposed Architecture for Trusted Third Party Services", *Proceedings of the International Conference on Cryptography: Policy and Algorithms*, Vol. 1029, 1995, pp. 98-104.

[7] Cylink, CyKey: Cylink's Key Recovery Solution. [Online]. Available : http://www.csm.ornl.gov/~dunigan/cykey.pdf, March 1997.

[8] Computer Security Resource Center, National Institute of Standards and Technology, Key Recovery Examples. [Online]. Available : http://csrc.nist.gov/krdp/exa.html, 1996.

[9] Y.Y. Al-Salqan, "Cryptographic Key Recovery", *Proceedings of the 6th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems (FTDCS' 97)*, October 1997, pp. 34-37.

[10] B.W. McConnell and E.J. Appel, Enabling Privacy, Commerce, Security and Public Safety in the Global Information Infrastructure. [Online]. Available : http://www.cdt.org/crypto/clipper_III/clipper_III_draft.html, May 1996.

[11] Yung-Cheng Lee and Chi-Sung Laih, "On the Key Recovery of the Key Escrow System", *Proceedings of the 13th Annual Computer Security Applications Conference (ACSAC '97)*, 1997, pp. 216-220.

[12] S. Lim, S. Kang and J. Sohn, "Modeling of Multiple Agent Based Cryptographic Key Recovery Protocol", *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003)*, Las Vegas, December 2003, pp. 119-128.

[13] Shin-Young Lim, Ho-Sang Hani, Myoung-Jun Kim and Tai-Yun Kim, "Design of Key Recovery System Using Multiple Agent Technology for Electronic Commerce", *Proceedings of 2001 IEEE Industrial Electronics*, Vol. 2, 2001, pp. 1351-1356.

[14] R. Perlman, "An Overview of PKI Trust Models", *IEEE Network*, Vol. 13, Issue 6, November 1999, pp. 38-43.

[15] A. Nash, W. Duane, C. Joseph and D. Brink, *PKI: Implementing and Managing E-Security*, RSA Press, McGraw-Hill, Inc., 2001.

[16] R. Hunt, "PKI and Digital Certification Infrastructure", *Proceedings of the 9th IEEE International Conference on Networks*, October 2001, pp. 234-239.

[17] M. Numao and Y. Nakayama, "Internet Archiving Server with Key Recovery Function", *1998 Symposium on Cryptography and Information Security*, 1998.

[18] Paolo D'Arco, "On the Distribution of a Key Distribution Center", *Proceedings of the 7th Italian Conference on Theoretical Computer Science*, Vol. 2202, 2001, pp. 357-369.

[19] B.C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks", *IEEE Communications Magazine*, Vol. 32, Issue 9, September 1994, pp. 33-38.

[20] A. Shamir, "How to Share a Secret", *Communications of the ACM*, Vol. 22, No. 11, November 1979, pp. 612-613.

A Simple High-Availability Multiple-Agent Key Recovery System

Kanokwan Kanyamee, Chanboon Sathitwiriyawong
Faculty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang, Thailand
E-mail: kikuit@hotmail.com, chanboon@it.kmitl.ac.th

Abstract

In symmetric cryptosystem, a session key is used for both message encryption and decryption. In case the session key is unavailable or legal investigation of transmitting messages is needed, an appropriate recovery mechanism is required. Key recovery can be achieved by a single agent or multiple agents. The latter can primarily enhance the security of the former by mitigating the risks of fabrication and collusion. The recovery of session key is usually provided by a trusted key recovery center (KRC) as a coordinator between key recovery agents (KRAs). This paper presents a key recovery system (SHAM-KRS) that has high availability. Since the session key can be recovered despite the failure of some key recovery agents, the problem of single point of failure can be avoided. It also has a feature to detect attacks on group authentication. Finally, it supports law enforcement needs and is based on the Public Key Infrastructure.

1. Introduction

Cryptography can ensure the security and privacy of network communications. However, the dishonest ones can also use the cryptosystems to conceal their criminal communications, which will endanger the social security. Therefore, it is necessary to develop a cryptosystem that meets the requirements of social security while maintaining user privacy.

Starting in 1993, the US government announced a new encryption technology called Key Escrow System (KES) [1]. This method provides for the privacy of users in addition to legal investigation of any suspected message by government authorities.

In February 1994, the US government announced a standard for Key Escrow System called the Escrow Encryption Standard (EES) [2, 3]. The security of EES depends on the physical protection of tamper-free chips. It supports message accessibility of government authorities and does not emphasize on all aspects of security and privacy.

Later in 1995, Trusted Information System (TIS) presented research on TIS Key Recovery System [4], which is developed from KES. Key Recovery System (KRS) can recover requested session keys without escrowing them. The system recovers the session key by Key Recovery Agent (KRA) [5, 6]. The session key is encapsulated in the Key Recovery Field (KRF) by the sender for later session key recovery as needed. Thus, it ensures the protection of user privacy.

Since then there has been continuous improvement of key recovery methods, such as trust [7, 8], authentication [9], key management [10], legal data access that user privacy is maintained [11], and enhancement of system security [12, 13]. For instance, most single-agent key recovery systems (S-KRS) [4, 7, 9, 11] aim to recover session keys that are lost by users, support law enforcement for message investigation, and consider personal rights of privacy. Furthermore, the system is based on Public Key Infrastructure (PKI) [14, 15, 16] to facilitate communication security services.

As the time passes, S-KRS can be easily attacked. Therefore, many researchers resort to designing multiple-agent key recovery system (M-KRS) [12, 13, 17] that can resist various threats such as brute-force attack and collusion of key recovery agents. M-KRS requires the collaboration of at least two KRAs.

The conventional M-KRS called Key Recovery Function [17] provides key recovery service by all participating KRAs. The user must decide which KRAs to use and generates key recovery block (KRB) from random numbers by using a one-way hash function. KRB contains portions of the session key stored with the encrypted data. When the session key is lost or damaged, the user can recover the session key from KRB by sending KRB to those KRAs.

Later, the research of multiple agent model [13] presented fork and join function for key recovery. The collaboration of at least two KRAs is required to recover the session key. Key recovery center (KRC) will act as the coordinating center for all KRAs

within the same group. KRF is created for all KRAs. It contains portions of the session key for later key recovery. The sender chooses one or more KRAs among a pool of KRAs and generates KRF. When session key recovery is needed, KRF is sent to KRC. Finally, KRC joins the portions of the session key to obtain the session key. However, some weakness persists in many M-KRS's as follows:

1) The risk of single point of failure from the unavailability of some KRAs since every KRA has to participate in the recovery of session key.

2) The absence of an attack detection function on group authentication of KRAs.

This paper presents a simple high-availability multiple-agent key recovery system (SHAM-KRS). The functions of the proposed system are as follows:

1) *High system availability.* The problem of single point of failure or the absent of some KRAs does not effect the system service.

2) *Detection of attacks on group authentication.* KRA that is not in the group can be detected.

The system also retains the standard format of KRF to support law enforcement and is based on the well-established PKI.

2. Proposed SHAM-KRS

The certificates for communicating parties, KRC, and all KRAs, are issued by the certificate authority (CA) in the PKI environment. It contains a public key (Ku) and other information for identification and authentication. Figure 1 depicts the processes of the proposed SHAM-KRS. These processes can be described as follows.

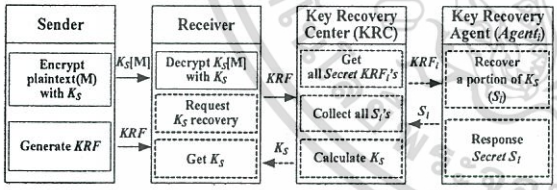


Figure 1. SHAM-KRS processes

2.1. Processes performed by the sender

The session key recovery process begins with the preparation of a key recovery field (*KRF*) by the sender. The encrypted message is sent together with the *KRF*. The encryption process is carried out at the sender as normal, while the generation of *KRF* is specific to the proposed KRS. The entire processes performed by the sender can be described as follows:

2.1.1. Initialization phase. A session key (K_S) is requested from the Key Distribution Center (KDC) [18] or Kerberos [19].

2.1.2. Message encryption phase. The original message (M) is encrypted with K_S, which is shared between the sender and the receiver, resulting $K_S[M]$.

2.1.3. Initial provision of KRF components phase. A *KRF* is created by the sender during normal encryption operation for the readiness of future key recovery. It is sent together with the standard encrypted message. It comprises portions of *KRF* (KRF_i's), Unique Secret Numbers (TT_i's), a hash value of the Share Group Number ($h(SGN)$), and *Other Information*.

Each KRF_i comprises a portion of the session key (S_i) and a Share Group Number (SGN). The S_i is used to recover K_S, while the SGN is used for group authentication. The SGN is specific to a certain group of KRA at a particular time. The TT_i's are reserved to recover K_S in case some participating agents fail or cannot provide the service. *Other Information* is the identification and authentication information stored in the certificate.

The initial procedures for the construction of a *KRF* are shown in Figure 2, and can be described as follows:

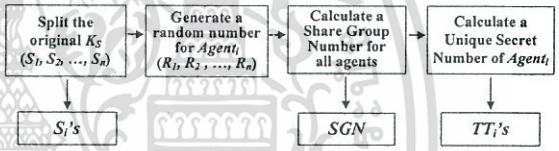


Figure 2. Initial procedures for *KRF* construction

1) The concept of secret sharing [20] is used to split K_S into a number of S_i's equaling the number of participating KRAs (n). The K_S can then be recovered by articulating all S_i's by the exclusive-OR (XOR) operation. This step can strengthen the security of the system. The procedures are described as follows:

- Generate $n-1$ random strings, $S_1, S_2, \ldots, S_{n-1}$, for $Agent_1, Agent_2, \ldots, Agent_{n-1}$, respectively.
- Calculate S_n for $Agent_n$ by $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus K_S$

Since S_i is a portion of K_S owned by $Agent_i$, K_S cannot be recovered when any S_i is lost or damaged.

2) A random number (R_i) for $Agent_i$ is generated. All R_i's are then distributed to every KRA ($Agent_1, Agent_2, \ldots, Agent_n$).

3) A Share Group Number (SGN) is obtained by the XOR operation of all R_i's as follows:

$$SGN = R_1 \oplus R_2 \oplus \cdots \oplus R_n$$

The benefit of creating *SGN* is to detect non-participating agents coming in the group to secure K_S. When KRC detects an unknown *SGN*, the system is compromised.

4) A Unique Secret Number (TT_i) will be used for the recovery of S_i in case $Agent_i$ fails or cannot provide the service. TT_i is calculated as follows:

$$TT_i = S_i \oplus SGN$$

5) A hash value of the Share Group Number (*h(SGN)*) is obtained from the calculation of *SGN* value by one-way hash function. The benefit of creating *h(SGN)* is to detect non-participating agents coming in the group and secure K_S since the *h(SGN)* is equal to *h(SGN)* of $Agent_i$. When KRC detects an unequal *h(SGN)*, the system is compromised.

2.1.4. Generation and formation of a KRF phase. The proposed system focuses on the security of the session key, the privacy of users and the ability to recover the session key correctly and completely despite the failure of some KRAs. These processes are described as follows:

- S_i and *SGN* are attached to KRF_i. Every KRF_i for $Agent_i$ is formed and encrypted with the public key of $Agent_i$ (Ku_{agi}) as follows:

$$KRF_i = Ku_{agi}[S_i, SGN]$$

- KRF_i , TT_i , *h(SGN)*, and *Other Information* are attached to *KRF* as follows:

$$KRF = Ku_{KRC}[KRF_i\text{'s} \,||\, TT_1, TT_2, \ldots, TT_n \,||\, h(SGN) \,||\, Other\ Information]$$

Then, the *KRF* is sent together with $K_S[M]$ to the receiver as usual.

2.2. Processes performed by the receiver

There are not many works done at the receiver, except for the reception and keeping of *KRF* every communication session and the launch of key recovery request. The entire processes performed by the receiver can be described as follows:

2.2.1 Decryption phase. Upon obtaining $K_S[M]$ || *KRF*, the receiver decrypts $K_S[M]$ with shared K_S from the Initialization phase. The *KRF* will only be used in case K_S is lost.

2.2.2 Session key recovery request phase. In case K_S is lost or damaged, or legal investigation of transmitting messages is needed, the KRC will be requested for key recovery service.

2.3. Processes of KRC and participating KRAs

All key recovery tasks take place at the KRC and all participating KRAs. The entire processes can be described as follows:

Session key recovery phase. K_S is constructed by the KRC. The session key recovery requires the collaboration of KRC and participating KRAs. The overview illustration of SHAM-KRS is shown in Figure 3.

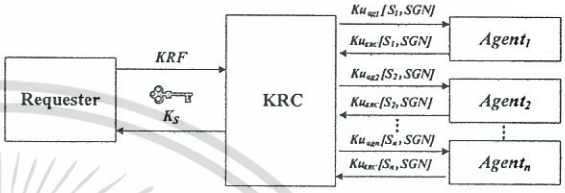


Figure 3. Overview illustration of the SHAM-KRS

The processes of session key recovery are shown in Figure 4 to Figure 6. Figure 4 depicts the preliminary session key recovery process by KRC. The subsequent partial session key recovery process performed by each agent is shown in Figure 5. The construction of K_S is shown in Figure 6. The entire processes can be described as the following steps.

1) Requester sends *KRF* to KRC.

2) KRC decrypts *KRF* with Kr_{KRC} to obtain KRF_i's.

3) KRC forwards $KRF_i = Ku_{agi}[S_i, SGN]$ to $Agent_i$.

4) $Agent_i$ decrypts $Ku_{agi}[S_i, SGN]$ with Kr_{agi} to obtain S_i and *SGN*.

5) $Agent_i$ encrypts S_i and *SGN* with Ku_{KRC}.

6) $Agent_i$ forwards $Ku_{KRC}[S_i, SGN]$ to KRC.

7) KRC decrypts $Ku_{KRC}[S_i, SGN]$ with Kr_{KRC} to obtain S_i and *SGN*.

8) KRC verifies *SGN* and collects S_i. The SGN is verified as follows:

- Calculate the hash value of the received *SGN* (*h(SGN)*).
- Compare *h(SGN)* with its own *h(SGN)*.

If they are equal, $Agent_i$ is a member of the key recovery group. Otherwise, a fabrication attack is detected.

9) Upon completing the collection of all S_i's, K_S is calculated by $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus S_n$.

10) K_S is encrypted with the public key of the requester (Ku_{req}), resulting $Ku_{req}[K_S]$.

11) Finally, $Ku_{req}[K_S]$ is forwarded to the requester for the decryption of $K_S[M]$ to get the original message.

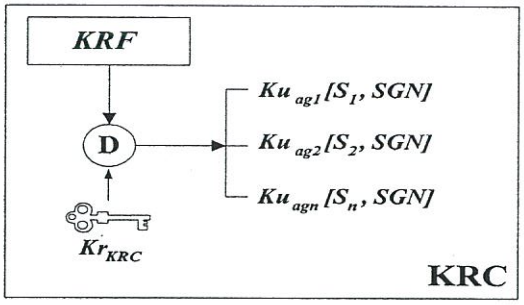


Figure 4. Preliminary session key recovery process by KRC

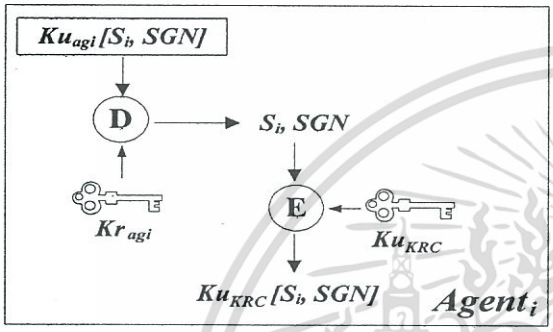


Figure 5. Partial session key recovery process by each KRA

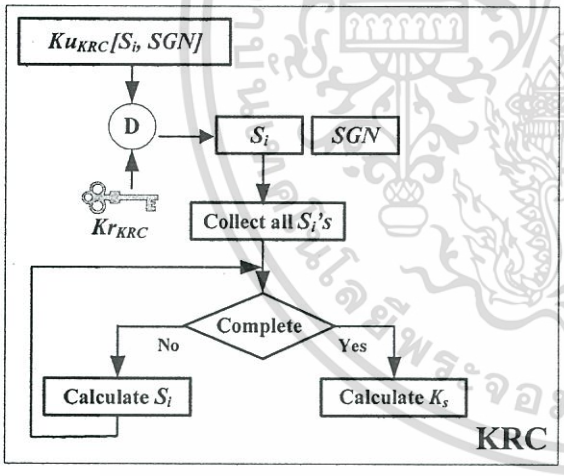


Figure 6. Session key recovery process by KRC

In case any KRA fails to provide its S_i, KRC cannot complete its task of gathering all S_i's. In this case, KRC manages the process as follows:

1) KRC checks for S_i that is not received from $Agent_i$.

2) KRC calculates S_i. KRC adopts TT_i of $Agent_i$ that cannot deliver S_i to KRC. S_i is calculated as follows:

$$S_i = TT_i \oplus SGN$$

3) Upon completing the collection of S_i, KRC can calculate K_S by $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus S_n$.

3. Feature analysis of various M-KRS's

A preliminary comparison of various M-KRS's shows differences in session key recovery system capabilities. The analyses of security features of various M-KRS's are shown in Table 1.

Compared to the conventional Key Recovery Function [17] and Multiple Agent Model [13], the proposed SHAM-KRS is able to avoid the single point of failure of KRAs and detect attacks on group authentication of KRAs, while maintaining law enforcement support. The summarization is presented as follows:

1) SHAM-KRS has an appropriate function of high secrecy of K_S and support law enforcement by using the suitable structure of new *KRF*.

2) SHAM-KRS can always securely recover K_S through the concept of secret sharing to circumvent the collusion of unfaithful agents and reservation of unique secret numbers to handle the failure of some agents that cannot be solved by other methods.

3) SHAM-KRS has the group authentication function using *h(SGN)* to detect non-participating KRAs coming in the group of KRAs to secure K_S.

Table 1. The analyses of security features of various M-KRS's

Capabilities	Conventional Key Recovery Function	Multiple-Agent Model	Proposed System (SHAM-KRS)
Secret sharing of K_S	No	Yes	Yes
Ability to recover K_S despite the failure of some KRAs	No	No	Yes
Group authentication of KRAs	No	No	Yes
Law enforcement support	Medium	High	High

The capabilities of SHAM-KRS are not only higher than other M-KRS's, but also more robust than other methods.

4. Performance study of SHAM-KRS

The two performance measurement experiments were conducted to determine the processing time (in milliseconds) during the generation of a *KRF* and the recovery of K_S, in terms of the number of agents.

The processing time was measured when using a Genuine Intel® CPU, 794 MHz with 1 GB of RAM. The results are shown in Figure 7 and Figure 8, respectively.

For the generation of *KRF*, SHAM-KRS requires more processing time than Multiple Agent Model because of the reservation of unique secret numbers for session key recovery in case of the failure of some KRAs. The processing time increases slowly in relation to the number of agents. It demonstrates that the additional processing time of SHAM-KRS does not incur numerous processing time, especially when the number of agents is less than 10. The extra time added is not significant for many applications, knowing that SHAM-KRS is more robust than Multiple Agent Model in terms of secrecy and availability. However, the difference of processing time between SHAM-KRS and Multiple Agent Model is negligible.

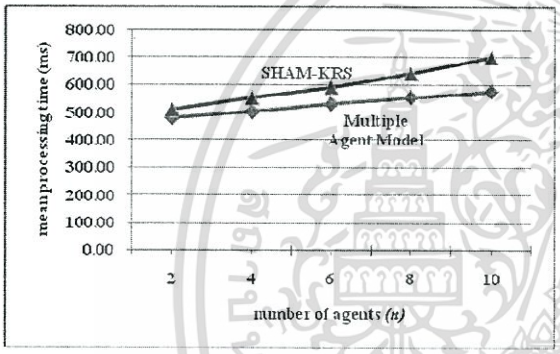


Figure 7. Performance of *KRF* generation

For the recovery of K_S, the results of processing time show the advantage of SHAM-KRS over Multiple Agent Model because of the less complicated structure of *KRF*. The processing time increases slowly with the number of agents.

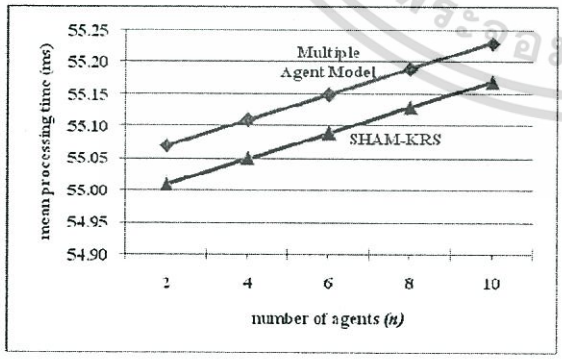


Figure 8. Performance of session key recovery

The last measurement experiment was conducted to determine the processing time required for the recovery of lost S_i's due to the failure of $Agent_i$'s.

Table 2 shows the results of S_i's recovery times (in microseconds) when one or more agents fail to provide their S_i's for key recovery service. It is noted that the number of maximum failed agents allowed is equal to the number of participating agents minus two. The processing time increases slowly in relation to the number of failed agents.

Table 2. Performance of S_i's recovery

Number of agents *(n)*	Number of failed agents	Recovery times of lost S_i's (μs)
3	1	79.32
4	1	79.32
	2	84.34
5	1	79.32
	2	84.34
	3	93.99

5. Conclusions

This paper proposes a simple high-availability multiple-agent key recovery system. Every aspect of security including confidentiality, integrity and availability, is considered. The proposed system can avoid the problem of single point of failure of key recovery agents since it can work despite the failure of some key recovery agents. The KRC provides an appropriate function and confidentiality in specifying the Unique Secret Number for participating agents. When any agent is unable to recover its portion of the session key, the KRC can still provide the key recovery service. Consequently, the system will be constantly available. The system is also able to detect attacks on group authentication. Only a predefined group of KRAs can involve in the key recovery operation. The structure of *KRF* emphasizes the security of session key and the privacy of users. Future work will be based on the function of SHAM-KRS to manage the minimum number of KRAs for key recovery according to security policies and requirements.

6. References

[1] D.E. Denning, "The US Key Escrow Encryption Technology", *Computer Communications*, Vol. 17, No. 7, July 1994, pp. 453-457.

[2] D.E. Denning and M. Smid, "Key Escrowing Today", *IEEE Communications Magazine*, Vol. 32, Issue 9, September 1994, pp. 58-68.

[3] National Institute of Standards and Technology, "Escrowed Encryption Standard", *Federal Information Processing Standards Publication (FIPS PUB) 185*, February 1994.

[4] S.T. Walker, S.B. Lipner, C.M. Ellison and D.M. Balenson, "Commercial Key Recovery", *Communications of the ACM*, Vol. 39, No. 3, March 1996, pp. 41-47.

[5] D.E. Denning and D.K. Branstad, "A Taxonomy for Key Recovery Encryption Systems", *Internet Besieged: Countering Cyberspace Scofflaws*, 1997, pp. 357-371.

[6] N. Jefferies, C. Mitchell and M. Walker, "A Proposed Architecture for Trusted Third Party Services", *Proceedings of the International Conference on Cryptography: Policy and Algorithms*, Vol. 1029, 1995, pp. 98-104.

[7] Cylink, CyKey: Cylink's Key Recovery Solution. [Online]. Available : http://www.csm.ornl.gov/~dunigan /cykey.pdf, March 1997.

[8] Computer Security Resource Center, National Institute of Standards and Technology, Key Recovery Examples. [Online]. Available : http://csrc.nist.gov/krdp/exa.html, 1996.

[9] Y.Y. Al-Salqan, "Cryptographic Key Recovery", *Proceedings of the 6th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems (FTDCS' 97)*, October 1997, pp. 34-37.

[10] B.W. McConnell and E.J. Appel, Enabling Privacy, Commerce, Security and Public Safety in the Global Information Infrastructure. [Online]. Available : http:// www.cdt.org/crypto/clipper_III/clipper_III_draft.html, May 1996.

[11] Yung-Cheng Lee and Chi-Sung Laih, "On the Key Recovery of the Key Escrow System," *Proceedings of the 13th Annual Computer Security Applications Conference*, December 1997, pp. 216-220.

[12] S. Lim, S. Kang and J. Sohn, "Modeling of Multiple Agent Based Cryptographic Key Recovery Protocol", *Proceedings of the 19th Annual Computer Security Applications Conference*, Las Vegas, December 2003, pp. 119-128.

[13] Shin-Young Lim, Ho-Sang Hani, Myoung-Jun Kim and Tai-Yun Kim, "Design of Key Recovery System Using Multiple Agent Technology for Electronic Commerce", *Proceedings of 2001 IEEE Industrial Electronics*, Vol. 2, 2001, pp. 1351-1356.

[14] R. Perlman, "An Overview of PKI Trust Models", *IEEE Network*, Vol. 13, Issue 6, November 1999, pp. 38-43.

[15] A. Nash, W. Duane, C. Joseph and D. Brink, *PKI: Implementing and Managing E-Security*, RSA Press, McGraw-Hill, Inc., 2001.

[16] R. Hunt, "PKI and Digital Certification Infrastructure", *Proceedings of the 9th IEEE International Conference on Networks*, October 2001, pp. 234-239.

[17] M. Numao and Y. Nakayama, "Internet Archiving Server with Key Recovery Function", *1998 Symposium on Cryptography and Information Security*, 1998.

[18] Paolo D'Arco, "On the Distribution of a Key Distribution Center", *Proceedings of the 7th Italian Conference on Theoretical Computer Science*, Vol. 2202, 2001, pp. 357-369.

[19] B.C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks", *IEEE Communications Magazine*, Vol. 32, Issue 9, September 1994, pp. 33-38.

[20] A. Shamir, "How to Share a Secret", *Communications of the ACM*, Vol. 22, No. 11, November 1979, pp. 612-613.

The International Arab Journal of Information Technology

No: 916/1317

Date: 12/3/2013

Dear Dr. Kanokwan Kanyamee,

We have completed the review of your paper submitted to *International Arab Journal of Information Technology (IAJIT)*.

Paper Number: 4749
Paper Title: High-Availability Decentralized Cryptographic Multi-Agent Key Recovery

Author(s): **Kanokwan Kanyamee** and Chanboon Sathitwiriyawong

I am happy to inform you that your paper mentioned above has been accepted for publication in the *International Arab Journal of Information Technology (IAJIT)*. We are planning tentatively to publish your paper in Volume 11, No. 1, January 2014.

Thank you for submitting a paper to IAJIT. We wish you the best and hope to receive more submissions from you in the future.

Sincerely yours,

Dr. Hamed Al-Fawareh
Editor-in-Chief

ประวัติผู้เขียน

ชื่อ-นามสกุล นางสาวกนกวรรณ กันยะมี

วัน เดือน ปีเกิด 16 เมษายน พ.ศ. 2521 ที่จังหวัดน่าน

ที่อยู่ปัจจุบัน 27 ถ.อินใจมี ต.ท่าอิฐ อ.เมือง จ.อุตรดิตถ์ 53000

ประวัติการศึกษา

ระดับปริญญาตรี วท.บ. สาขาวิทยาการคอมพิวเตอร์ สถาบันราชภัฏอุตรดิตถ์

ระดับปริญญาโท วท.ม. สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยนเรศวร

การทำงาน

ตำแหน่งอาจารย์ประจำหลักสูตรเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏอุตรดิตถ์