AIX Version 7.2

4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4

AIX Version 7.2

4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4

หมายเหตุ

ก่อนใช้ข้อมูลนี้ รวมถึงผลิตภัณฑ์ที่สนับสนุน โปรดอ่าน ข้อมูลใน “คำประกาศ” ในหน้า 73

This edition applies to AIX Version 7.2 and to all subsequent releases and modifications until otherwise indicated in new editions.

สารบัญ

เกี่ยวกับเอกสารนี้

ข้อมูลการติดตั้งอธิบาย Release 4.4 ของ IBM® Common Cryptographic Architecture (CCA) Support Program (อ้างถึงเป็น Support Program) สำหรับ IBM 4765 PCIe Cryptographic Coprocessor ส่วนสนับสนุนโปรแกรม ประกอบด้วยไดร์เวอร์ อุปกรณ์ ยูทิลิตี และโค้ดตัวประมวลผลร่วม

ใช้ข้อมูลนี้เพื่อให้ความช่วยเหลือกับภารกิจต่อไปนี้:

- ขอรับส่วนสนับสนุนโปรแกรมผ่านอินเตอร์เน็ต
- โหลดซอฟต์แวร์ไปยังโฮสต์คอมพิวเตอร์และไปยังตัวประมวลผลร่วม
- ใช้ยูทิลิตีเพื่อจัดหาส่วนสนับสนุนโปรแกรม:
  - โหลดตัวประมวลผลร่วม function-control vector (FCV)
  - เตรียมข้อมูลเบื้องต้นให้กับตัวประมวลผลร่วมตั้งแต่หนึ่งตัวขึ้นไป
  - สร้างและจัดการกับการควบคุมการเข้าถึงข้อมูล
  - สร้างคีย์หลักและ key-encrypting keys (KEKs) หลัก
  - จัดการกับหน่วยเก็บคีย์ที่โหนดที่เข้ารหัสลับ
  - สร้างไฟล์การกำหนดค่าเริ่มต้นให้กับโหนดเพื่อติดตั้ง และตั้งค่าโหนดที่เข้ารหัสลับ
- ลิงก์แอ็พพลิเคชันซอฟต์แวร์กับไลบรารี CCA
- ขอรับคำแนะนำเกี่ยวกับข้อควรพิจารณาด้านความปลอดภัยในการพัฒนาแอ็พพลิเคชัน และการฝึกปฏิบัติเกี่ยวกับการดำเนินการ

ผู้เข้าชม

ผู้เข้าชมเอกสารคู่มือนี้ประกอบด้วย:

- ผู้ดูแลระบบซึ่งเป็นผู้ติดตั้งซอฟต์แวร์
- เจ้าหน้าที่รักษาความปลอดภัยที่รับผิดชอบต่อระบบการควบคุมสิทธิในการเข้าถึง ตัวประมวลผลร่วม
- โปรแกรมเมอร์ระบบและโปรแกรมเมอร์แอ็พพลิเคชันผู้ที่กำหนดวิธีการใช้ ซอฟต์แวร์

การไฮไลต์

ระเบียบการไฮไลต์ต่อไปนี้ถูกใช้ในเอกสารนี้:

ตัวหนา	ระบุคำสั่ง รูทีนย่อย คีย์เวิร์ด ไฟล์ โครงสร้าง ไดเร็กทอรี และไอเท็มอื่นๆ ที่มีชื่อถูกกำหนดไว้ล่วงหน้าโดยระบบ และยังระบุอ็อบเจ็กต์รูปภาพ เช่น ปุ่ม เลเบล และไอคอนที่ผู้ใช้เลือก
Italics	ระบุพารามิเตอร์ซึ่งผู้ใช้จะเป็นผู้ระบุชื่อจริง หรือค่า
โมโนสเปซ	ระบุตัวอย่างค่าข้อมูลที่ระบุ, ตัวอย่างข้อความที่คล้ายกับที่คุณจะเห็นเมื่อถูกแสดง, ตัวอย่าง ของส่วนของโค้ดโปรแกรมที่คล้ายกับที่คุณอาจเขียนในฐานะที่เป็นโปรแกรมเมอร์, ข้อความจากระบบ หรือข้อมูลที่คุณควรพิมพ์

การคำนึงถึงขนาดตัวพิมพ์ใน AIX®

ทุกสิ่งในระบบปฏิบัติการ AIX เป็นแบบตรงตาม ตัวพิมพ์ ซึ่งหมายความว่ามีการแยกแยะความแตกต่างระหว่างตัวอักษรพิมพ์ใหญ่ และพิมพ์เล็ก ตัวอย่างเช่น คุณสามารถใช้คำสั่ง **ls** เพื่อ แสดงรายชื่อไฟล์ ถ้าคุณพิมพ์ `LS` ระบบ ตอบสนองว่าคำสั่งนี้ `not found` เช่นเดียวกับ `FILEA`, `FiLea` และ `filea` ถือเป็นชื่อไฟล์ต่างกันสามชื่อ แม้ว่า ไฟล์เหล่านี้จะอยู่ในไดเร็กทอรีเดียวกัน

เพื่อหลีกเลี่ยงการทำการดำเนินการที่ไม่ต้องการ ตรวจสอบให้แน่ใจว่าคุณใช้ตัวพิมพ์ที่ถูกต้องเสมอ

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

งานพิมพ์ที่เกี่ยวข้อง

คู่มือสำหรับ PCIe Cryptographic Coprocessor และแอ็พพลิเคชัน การเข้ารหัสเชิงพาณิชย์ทั่วไปให้ติดตามที่:

คู่มือฮาร์ดแวร์การเข้ารหัสมีอยู่ที่เว็บไซต์ *CryptoCards* ที่ http://www.ibm.com/security/cryptocards:

- *IBM CCA Basic Services Reference และคู่มือสำหรับ IBM 4765 PCIe และ IBM 4764 PCI-X Cryptographic Coprocessors*

4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4

หากต้องการใช้ข้อมูลนี้อย่างมีประสิทธิผล คุณต้องทำความคุ้นเคยกับคำสั่ง การเรียกของระบบ รูทีนย่อย รูปแบบไฟล์ และไฟล์พิเศษต่างๆ

มีอะไรใหม่ใน 4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4

อ่านเกี่ยวกับข้อมูลใหม่หรือข้อมูลที่ถูกเปลี่ยนแปลงอย่างมากสำหรับชุดหัวข้อ 4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4

วิธีดูสิ่งใหม่ หรือที่เปลี่ยนแปลง

ในไฟล์ PDF นี้ คุณอาจเห็นแถบการแก้ไข (|) ในขอบด้านซ้าย เพื่อระบุข้อมูลใหม่ และที่เปลี่ยนแปลง

ธันวาคม 2015

ต่อไปนี้เป็นข้อมูลสรุปของอัพเดตที่มีในชุดของหัวข้อนี้:

- IBM PCIe Cryptographic Coprocessor ถูกอัพเดตเป็น IBM PCIe Cryptographic Coprocessor Version 4.4.55 ใน หัวข้อต่อไปนี้:
  - "4765 PCIe Cryptographic Coprocessor AIX CCA Support Program Installation 4.4"
  - "การติดตั้งส่วนสนับสนุนโปรแกรมพื้นฐานรีลีส 4.4" ในหน้า 4
  - "การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม" ในหน้า 2

  คุณสามารถดาวน์โหลดซอฟต์แวร์ตัวประมวลผลร่วมจากเว็บไซต์ IBM PCIe Cryptographic Coprocessor http://www-03.ibm.com/security/cryptocards/pciecc/release4455.shtml

ภาพรวมกระบวนการการติดตั้ง Support Program

ภาพรวม AIX CCA นี้อธิบายขั้นตอนในการติดตั้ง และดำเนินการ IBM Cryptographic Coprocessor Support Program บนโฮสต์คอมพิวเตอร์

ข้อมูลที่เกี่ยวข้อง:

"การติดตั้งส่วนสนับสนุนโปรแกรม" ในหน้า 2

ขั้นตอนในการติดตั้ง IBM Common Cryptographic Architecture (CCA) Support Program บนคอมพิวเตอร์โฮสต์ตัวประมวลผลร่วม

การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม

ข้อมูลเกี่ยวกับการเลือก การติดตั้ง และการสั่งซื้อ ฮาร์ดแวร์ตัวประมวลผลร่วม และเพื่อดาวน์โหลดซอฟต์แวร์

ส่วนต่อไปนี้อธิบายถึงวิธีการดังต่อไปนี้:

- การสั่งซื้อตัวประมวลผลร่วม
- การเปิดใบสั่งซื้อตัวประมวลผลร่วม IBM 4765
- การติดตั้งฮาร์ดแวร์ IBM 4765
- การรับซอฟต์แวร์ตัวประมวลผลร่วม

การสั่งซื้อตัวประมวลผลร่วม

IBM 4765-001 ถูก สั่งซื้อจาก IBM ตามชนิดเครื่องและโมเดล ตัวประมวลผลร่วม ต้องการสล็อต PCIe ที่รองรับอะแด็ปเตอร์ PCIe ความยาว 2/3

ซอฟต์แวร์ สนับสนุนมากถึงแปดตัวประมวลผลร่วมต่อระบบ ขึ้นกับจำนวนของสล็อต PCIe ที่มี

การเปิดใบสั่งซื้อตัวประมวลผลร่วม IBM 4765

หากต้องการเปิดการสั่งซื้อฮาร์แวร์ตัวประมวลผลร่วม โปรดติดต่อตัวแทน IBM ของคุณ หรือพาร์ตเนอร์ทางธุรกิจของ IBM และสั่งซื้อโมเดล และฟีเจอร์ที่คุณเลือก

ลูกค้าในประเทศสหรัฐอเมริกายังสามารถติดต่อ IBM Direct ได้ที่ 1-800-IBM-CALL โดยเฉพาะ *IBM 4765* ที่คำสั่ง ซื้อของคุณถูกส่งไปที่กลุ่มที่ดำเนินการใบสั่งซื้อ IBM 4765

การติดตั้งฮาร์ดแวร์ IBM 4765

IBM 4765 ถูก ติดตั้งในแบบเดียวกับอะแด็ปเตอร์ PCIe อื่น ทำตามกระบวนการที่อธิบายไว้ใน *การติดตั้ง IBM PCIe Cryptographic Coprocessor* สำหรับข้อมูล โดยละเอียด

การรับ ซอฟต์แวร์ตัวประมวลผลร่วม

ซอฟต์แวร์รับได้โดยการดาวน์โหลด จากเว็บไซต์: http://www.ibm.com/security/cryptocards/pciecc/ordersoftware.shtml

การติดตั้งส่วนสนับสนุนโปรแกรม

ขั้นตอนในการติดตั้ง IBM Common Cryptographic Architecture (CCA) Support Program บนคอมพิวเตอร์โฮสต์ตัวประมวลผลร่วม

IBM Common Cryptographic Architecture (CCA) Support Program ประกอบด้วยหลายคอมโพเนนต์ รวมถึง:

- ไดร์เวอร์อุปกรณ์และระบบปฏิบัติการสำหรับฮาร์ดแวร์ตัวประมวลผลร่วม การเข้ารหัสลับ PCIe
- สนับสนุน IBM Common Cryptographic Architecture (CCA) application program interface (API)

- function-control vector (FCV)

  หมายเหตุ: FCV คือค่าที่ลงนามซึ่งจัดเตรียมไว้โดย IBM ซึ่งเปิดใช้งาน แอ็พพลิเคชัน CCA ภายในตัวประมวลผลร่วมกับผลผลิตในระดับของเซอร์วิสการเข้ารหัสลับที่สอดคล้องกับกฎข้อบังคับ ในการนำการอิมพอร์ตและเอ็กซ์พอร์ตการเข้ารหัสลับที่สามารถเรียกใช้ได้ไปใช้งาน
- แอ็พพลิเคชันยูทิลีตีที่ตัวประมวลผลร่วมต้องถูกติดตั้งไว้ ซึ่งรัน บนเครื่องโฮสต์

เมื่อต้องการติดตั้งและตั้งค่า IBM Common Cryptographic Architecture (CCA) Support Program ให้ทำขั้นตอนเหล่านี้ให้สมบูรณ์:

1. เลือกแพ็กเกจการสนับสนุนแพลตฟอร์มที่เหมาะสมกับการเซ็ตอัพ ของคุณ:

   AIX 6.1 หรือใหม่กว่า

   โปรดดู “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2สำหรับรายละเอียด
2. สั่งซื้อฮาร์ดแวร์ที่มี IBM หรือ IBM Business Partner ของคุณดูที่ “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2อธิบายวิธีสั่งซื้อและรับ ฮาร์ดแวร์ตัวประมวลผลร่วมจาก IBM
3. ดาวน์โหลดส่วนสนับสนุนโปรแกรมสำหรับ ระบบปฏิบัติการของคุณ ดูที่ “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2อธิบายวิธีติดตั้งระบบปฏิบัติการแบบฝังตัว และโปรแกรมแอ็พพลิเคชัน CCA ลงใน PCIe Cryptographic Coprocessor
4. ติดตั้งส่วนสนับสนุนโปรแกรมบนโฮสต์คอมพิวเตอร์ ตัวประมวลผลร่วม
5. ติดตั้งฮาร์ดแวร์ตัวประมวลผลร่วม โปรดดู “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2สำหรับรายละเอียด
6. โหลดซอฟต์แวร์ตัวประมวลผลร่วม โปรดดู “การโหลดและการยกเลิกการโหลดซอฟต์แวร์เข้าสู่ตัวประมวลผลร่วม” ในหน้า 8 สำหรับ รายละเอียด
7. ติดตั้งโหนดการทดสอบ CCA คุณสามารถสร้างโหนดการเข้ารหัส CCA โดยใช้ยูทิลีตีที่จัดเตรียมด้วย Support Program หรือ ลิงก์โปรแกรมแอ็พพลิเคชันของคุณกับ CCA API และตรวจสอบการควบคุมการเข้าถึง และข้อกำหนดการเซ็ตอัพอื่นที่กำหนดโดยแอ็พพลิเคชันซอฟต์แวร์ ที่คุณวางแผนจะใช้กับ IBM 4765 ยูทิลิตี CCA Node Management (CNM) กล่าวถึงใน “การจัดการโหนดที่เข้ารหัสโดยใช้ยูทิลีตี CNM และ CNI” ในหน้า 18 ประกอบด้วยการติดตั้งและฟังก์ชันการจัดการที่จำเป็นต้อง:
   - โหลด FCV
   - สร้างและแก้ไขข้อมูลการควบคุมสิทธิในการเข้าถึง
   - จัดการกับคีย์หลักตัวประมวลผลร่วม
   - จัดการ key encrypting keys (KEKs) หลัก
   - จัดการกับหน่วยเก็บคีย์ข้อมูล
   - สร้างรายการ (สคริปต์) สำหรับยูทิลิตี CCA Node Initialization (CNI)
8. รันโปรแกรมทดสอบที่นำไลบรารี CCA ไปใช้งาน โปรดดู “การ Build แอ็พพลิเคชันเพื่อใช้กับ CCA API” ในหน้า 42 สำหรับรายละเอียด

ข้อมูลที่เกี่ยวข้อง:

“การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2
ข้อมูลเกี่ยวกับการเลือก การติดตั้ง และการสั่งซื้อ ฮาร์ดแวร์ตัวประมวลผลร่วม และเพื่อดาวน์โหลดซอฟต์แวร์

“การโหลดและการยกเลิกการโหลดซอฟต์แวร์เข้าสู่ตัวประมวลผลร่วม” ในหน้า 8
หลังจากการติดตั้ง IBM Common Cryptographic Architecture (CCA) Support Program บนโฮสต์คอมพิวเตอร์ให้ใช้ Coprocessor Load Utility (CLU) เพื่อโหลดระบบปฏิบัติการของตัวประมวลผลร่วมและแอ็พพลิเคชัน CCA เข้าสู่ตัวประมวลผลร่วม

“การจัดการโหนดที่เข้ารหัสโดยใช้ยูทิลีตี CNM และ CNI” ในหน้า 18
คอมพิวเตอร์ที่จัดเตรียมเซอร์วิสการเข้ารหัสลับ เช่น การสร้างคีย์ และส่วนสนับสนุนลายเซ็นแบบดิจิทัลที่ได้นิยามไว้ที่นี่เป็น *โหนด การเข้ารหัสลับ*

การติดตั้งส่วนสนับสนุนโปรแกรมพื้นฐานรีลีส 4.4

คำแนะนำสำหรับการติดตั้ง Support Program บน โฮสต์คอมพิวเตอร์ตัวประมวลผลร่วม

สิ่งที่จำเป็นต้องมีก่อน

ก่อนคุณเริ่มการติดตั้ง ให้เลือกแพ็กเกจการสนับสนุนแพลตฟอร์มที่เหมาะสมกับการเซ็ตอัพ ของคุณ ดูที่ “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2 สำหรับรายละเอียด เกี่ยวกับข้อกำหนดซอฟต์แวร์และฮาร์ดแวร์สำหรับ AIX

หมายเหตุ: หากคุณไม่ได้ติดตั้งโปรแกรมในครั้งแรก ให้สำรองไฟล์หน่วยเก็บคีย์ของคุณก่อน

หากต้องการติดตั้งส่วนสนับสนุนโปรแกรม:

1. ป้อนคำสั่ง **smitty install_all**
2. ป้อนตำแหน่งของอิมเมจการติดตั้งที่คุณได้รับ โดยใช้ขั้นตอนที่อธิบายในส่วน การรับซอฟต์แวร์ตัวประมวลผลร่วม ใต้ “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2 กด Enter
3. ป้อน `csufx.4765.cca` `csufx.4765.man` ในฟิลด์ **SOFTWARE install** หรือกด F4 (Display) เพื่อเลือก จากรายการ ตรวจสอบว่า **ติดตั้งซอฟต์แวร์ที่จำเป็นโดยอัตโนมัติ** ถูก ตั้งค่าเป็น ใช่ และ **ยอมรับข้อตกลงการอนุญาตใช้สิทธิ์ใหม่** ถูกตั้งค่าเป็น ใช่ ใช้ปุ่มตั้งระยะเพื่อสลับ หรือคีย์ F4 (Display) เพื่อแสดงรายการ กด Enter และกด Enter อีกครั้งเพื่อดำเนินการต่อ เมื่อได้รับพร้อมต์ `ARE YOU SURE`
4. ออกจาก **smitty** โดยใช้คีย์ F10 (Exit)
5. อ่านไฟล์ `/usr/lpp/csufx.4765/README` ไฟล์นี้มีข้อมูลล่าสุดเกี่ยวกับผลิตภัณฑ์ ส่วนสนับสนุนโปรแกรม
6. ใช้ยูทิลิตีคอนฟิกูเรชันเพื่อตั้งค่าซอฟต์แวร์ตามที่กล่าวไว้ใน “การตั้งค่าส่วนสนับสนุนโปรแกรม”

การตั้งค่าส่วนสนับสนุนโปรแกรม

ส่วนนี้อธิบายยูทิลีตีและคำสั่งระบบที่ใช้ ตั้งค่าซอฟต์แวร์ CCA Cryptographic Coprocessor Support Program

csufadmin

ระบุสิทธิ์ในการเข้าถึงระบบที่เชื่อมโยงกับ ยูทิลีตี csufkeys, csufappl, csufclu (Coprocessor Load Utility), csufcnm (Cryptographic Node Management) และ csufcni (Cryptographic Node Initialization)

สิทธิ์ดีฟอลต์จำกัด การใช้ยูทิลีตีเหล่านี้ให้กับผู้ใช้รูทเท่านั้นและให้กับผู้ใช้ในกลุ่ม ระบบ ใช้ยูทิลีตี csufadmin เพื่อแก้ไขสิทธิ์เหล่านี้

csufappl

ระบุสิทธิในการเข้าถึงระบบที่เชื่อมโยง กับไลบรารี CCA

สิทธิที่เป็นค่าดีฟอลต์จำกัด การใช้ไลบรารี CCA กับผู้ใช้รูทและสมาชิกของกลุ่ม ระบบ ใช้ยูทิลีตี csufappl เพื่ออนุญาตให้กลุ่มอื่น ใช้เซอร์วิสที่มีให้โดย CCA API

csufkeys

สร้างและระบุไฟล์และชื่อไดเร็กทอรีของตำแหน่ง ซึ่งอยู่ภายในคีย์การเข้ารหัสลับและรายการคีย์ที่เก็บไว้โปรแกรมการติดตั้งนิยามไดเร็กทอรีที่เป็นค่าดีฟอลต์ต่อไปนี้ใน AIX object data manager (ODM):

- ไดเร็กทอรี AES key-record-list: `/usr/lpp/csufx.4765/csufkeys/aeslist`
- ไฟล์ที่เก็บคีย์ AES: `/usr/lpp/csufx.4765/csufkeys/aes.keys`
- ไดเร็กทอรี DES key-record-list: `/usr/lpp/csufx.4765/csufkeys/deslist`
- ไฟล์ที่เก็บคีย์ DES: `/usr/lpp/csufx.4765/csufkeys/des.keys`
- ไดเร็กทอรี PKA key-record-list: `/usr/lpp/csufx.4765/csufkeys/pkalist`
- ไฟล์ที่เก็บคีย์ PKA: `/usr/lpp/csufx.4765/csufkeys/pka.keys`

ใช้ยูทิลีตี csufkeys เพื่อเปลี่ยนแปลงตำแหน่ง ที่จัดเก็บ

หมายเหตุ: เมื่อคุณเตรียมข้อมูลเบื้องต้นให้กับหน่วยเก็บคีย์โดยใช้ ยุทิลิตี Cryptographic Node Management ตรวจสอบให้แน่ใจว่า คุณระบุ ไดเร็กทอรี ODM ที่นิยามไว้โดยยูทิลิตี

odmget ตรวจสอบชื่อไฟล์หน่วยเก็บคีย์ด้วยคำสั่งระบบ **odmget** คุณสามารถตรวจสอบชื่อหน่วยเก็บคีย์ได้โดยใช้ส่วนสนับสนุนโปรแกรม CCA โดยป้อนคำสั่ง **odmget csufodm** แอ็ตทริบิวต์ parameter name สี่ตัวระบุค่าต่อไปนี้:

- **csuaesds**: ไฟล์ที่มี AES key-records
- **csuaesld**: ไดเร็กทอรีที่มีไฟล์ AES key-record-list
- **csudesds**: ไฟล์ที่มี DES key-records
- **csudesld**: ไดเร็กทอรีที่มีไฟล์ DES key-record-list
- **csupkads**: ไฟล์ที่มี PKA key-records
- **csupkald**: ไดเร็กทอรีที่มีไฟล์ PKA key-record-list

เมื่อเตรียมข้อมูลเบื้องต้นให้กับหน่วยเก็บคีย์ CCA ด้วยยูทิลิตี CNM หรือด้วย csnbksi CCA verb คุณต้องใช้ชื่อไฟล์ที่ ส่งคืนจาก ODM ใช้ยูทิลีตี csufkeys เพื่อเปลี่ยนแปลง ชื่อไฟล์เหล่านี้

DES_Key_Record_List verb, PKA_Key_Record_List verb และ AES_Key_Record_List verb สร้างไฟล์รายการในไดเร็กทอรี `/usr/lpp/csufx.4765/csufkeys/deslist`, `/usr/lpp/csufx.4765/csufkeys/pkalist` และ `/usr/lpp/csufx.4765/csufkeys/aeslist` ตามลำดับ ซึ่งมีชื่อไดเร็กทอรีที่เป็นค่าดีฟอลต์ คุณสามารถแก้ไข ชื่อไดเร็กทอรีเมื่อคุณติดตั้งซอฟต์แวร์ไฟล์รายการ ถูกสร้างขึ้นภายใต้ความเป็นเจ้าของของคุณ หากคุณร้องขอบริการรายการ ให้ตรวจสอบว่าไฟล์ถูก

สร้างขึ้นภายใต้ ID กลุ่มที่จำเป็นต่อ การติดตั้ง ซึ่งยังสามารถบรรลุเป้าหมายได้โดยตั้งค่าบิต set-group-id-on-execution บนไดเร็กทอรีทั้งสามเหล่านี้ โปรดดูแฟล็ก g+s ในคำสั่ง **chmod** สำหรับข้อมูลเพิ่มเติม หากไม่ได้ทำตามโพรซีเดอร์นี้ ข้อผิดพลาด จะถูกส่งกลับไปยัง key-record-list verbs

เมื่อต้องการระบุตัวประมวลผลร่วม CCA ดีฟอลต์ ให้ใช้คำสั่ง EXPORT เพื่อตั้งค่าตัวแปรสภาวะแวดล้อม `CSU_DEFAULT_ADAPTER` เป็น `CRP0`*n* โดย *n* = 1, 2, 3, 4, 5, 6, 7 หรือ 8 ขึ้นอยู่ กับตัวประมวลผลร่วม CCA ที่ติดตั้งที่คุณต้องการเป็นค่าดีฟอลต์ หากตัวแปรสภาวะแวดล้อมไม่ได้ถูกตั้งค่าไว้ เมื่อ CCA verb แรกของกระบวนการถูกเรียก ซอฟต์แวร์ CCA จะใช้ตัวประมวลผลร่วม CRP01 เป็นค่าดีฟอลต์ หากตัวแปรสภาวะแวดล้อมถูกตั้งค่าซึ่งเป็นค่าที่ไม่ถูกต้อง คุณจะได้รับข้อผิดพลาดจนกระทั่งตัวแปรสภาวะแวดล้อมถูกตั้งค่า ให้มีค่าที่ถูกต้อง

ข้อมูลที่เกี่ยวข้อง:

"การสร้างเลเบลของคีย์" ในหน้า 40

ส่วนสนับสนุนโปรแกรม CCA และสิทธิในการใช้ไฟล์ AIX

ส่วนสนับสนุน CCA อ้างอิงตามสิทธิในการใช้ไฟล์ที่ระดับของกลุ่ม กับฟังก์ชันอย่างถูกต้อง

ผู้ใช้และผู้ดูแลระบบของส่วนสนับสนุนโปรแกรมต้องมี สิทธิในการใช้ไฟล์กลุ่มอย่างถูกต้องบนไลบรารี CCA ที่แบ่งใช้ ยูทิลิตีไฟล์หน่วยเก็บคีย์และไดเร็กทอรีที่ต้องการให้ทำงานอย่างสมบูรณ์ และรันโดยไม่มีข้อผิดพลาด

หมายเหตุ: ไฟล์ที่เก็บคีย์ และไดเร็กทอรีถูกกำหนดเป็นไฟล์และ ไดเร็กทอรีที่มีอยู่ในไดเร็กทอรีที่เก็บคีย์ ไดเร็กทอรี นี้ประกอบด้วยไดเร็กทอรีที่เก็บคีย์ระดับสูงสุด นั้นคือ ในคอนฟิกูเรชันดีฟอลต์ ไฟล์และไดเร็กทอรีทั้งหมดภายใต้ไดเร็กทอรี `/usr/lpp/csufx.4765/csufkeys/deslist` และไดเร็กทอรี `/usr/lpp/csufx.4765/csufkeys` เอง

เมื่อต้องการดำเนินการไฟล์หน่วยเก็บคีย์และไดเร็กทอรีต้องมี ID กลุ่มของกลุ่มผู้ใช้แอ็พพลิเคชัน นั่นคือ พารามิเตอร์ groupname ที่ถูกใช้เมื่อยูทิลิตี `csufappl` ถูกรัน

และตามกฎทั่วไป ไดเร็กทอรีหน่วยเก็บคีย์ทั้งหมดต้องมีสิทธิการใช้ไฟล์ 2770 (drwxrws---) และเป็นเจ้าของโดย root ไฟล์หน่วยเก็บคีย์ทั้งหมดต้องมีสิทธิการใช้ไฟล์ 660 (-rw-rw----)

ซอฟต์แวร์ 4765 CCA และที่เก็บคีย์ไม่สามารถมีอยู่พร้อมกับซอฟต์แวร์ 4764 CCA และที่เก็บคีย์ เนื่องจากมีข้อขัดแย้งกันในไลบรารีและ ฐานข้อมูล ODM

การตรวจทานข้อผิดพลาดฮาร์ดแวร์ตัวประมวลผลร่วม

ข้อผิดพลาดที่เกิดขึ้นในฮาร์ดแวร์ตัวประมวลผลร่วม IBM Power Systems™ ถูกบันทึกไว้ในบันทึกข้อผิดพลาด AIX

หากต้องการประมวลผลและดูบันทึกการทำงาน ให้ป้อนคำสั่งต่อไปนี้:

```
errpt -a -N Cryptn,libxcrypt.a | more
```

โดยที่ *n* คือ 0, 1, 2, 3, 4, 5, 6 หรือ 7 (ตัวอย่าง Crypt 0) ขึ้นกับบันทึก CCA Coprocessor ที่คุณต้องการดู

ข้อมูลที่เกี่ยวข้อง:

"การโหลดและการยกเลิกการโหลดซอฟต์แวร์เข้าสู่ตัวประมวลผลร่วม" ในหน้า 8

หลังจากการติดตั้ง IBM Common Cryptographic Architecture (CCA) Support Program บนโฮสต์คอมพิวเตอร์ให้ใช้ Coprocessor Load Utility (CLU) เพื่อโหลดระบบปฏิบัติการของตัวประมวลผลร่วมและแอ็พพลิเคชัน CCA เข้าสู่ตัวประมวล

ผลร่วม

การลบส่วนสนับสนุนโปรแกรม

หากไฟล์ที่เก็บคีย์ของคุณอยู่ในไดเร็กทอรีดีฟอลต์ ให้สำรองข้อมูลไฟล์ หรือบันทึกไฟล์เหล่านั้นก่อนคุณลบ IBM Cryptographic Coprocessor (CCA) Support Program การลบซอฟต์แวร์จะลบ ไฟล์ที่เก็บคีย์ในไดเร็กทอรีดีฟอลต์

เมื่อต้องการลบ IBM Cryptographic Coprocessor Support Program ทำตามขั้นตอนเหล่านี้:

1. ให้ล็อกออนเป็น `root`
2. ป้อนคำสั่ง **rmdev -dl Crypt0** ไดร์เวอร์อุปกรณ์ ตัวประมวลผลร่วมและข้อมูลที่เกี่ยวข้องอื่นๆ ถูกลบ คุณสามารถ ใช้คำสั่งนี้สำหรับตัวประมวลผลร่วม CCA แต่ละตัวที่คุณวางแผนลบหรือย้ายที่
3. ป้อนคำสั่ง **smitty install_remove**

   หมายเหตุ: เมื่อ พร้อมต์ ป้อน `csufx.4765.com` และ `devices.pciex.14107a0314107b03.rte` ชื่อ ผลิตภัณฑ์
4. ตรวจสอบว่าค่า **REMOVE dependent software** ถูกตั้งค่าเป็น NO รวมทั้งตรวจสอบว่าค่า **Preview Only** ถูกตั้งค่าเป็น NO
5. กดคีย์ **Enter**

ความต้องการฮาร์ดแวร์และซอฟต์แวร์ AIX

ข้อกำหนดเบื้องต้นที่จำเป็นในการติดตั้ง CCA

ฮาร์ดแวร์

ติดตั้งเซิร์ฟเวอร์ IBM Power Systems ที่มี ตัวประมวลผลร่วมเข้ารหัส 4765 PCIe ที่พร้อมใช้งาน

ในระหว่างการติดตั้งซอฟต์แวร์ ไดร์เวอร์จะโต้ตอบกับ ตัวประมวลผลร่วมเพื่อชี้ขาดถึงค่าติดตั้งเกี่ยวกับอินเตอร์รัปต์ ช่องสัญญาณ DMA และรีซอร์สของระบบอื่นๆ สำหรับ คำแนะนำการติดตั้งเกี่ยวกับฮาร์ดแวร์และไดร์เวอร์อุปกรณ์ ตัวประมวลผลร่วม ดูที่ “การขอรับฮาร์ดแวร์และซอฟต์แวร์ของตัวประมวลผลร่วม” ในหน้า 2

ซอฟต์แวร์

1. IBM AIX 6.1 และ สูงกว่า
2. Java Runtime Environment (JRE) 1.6.0 หรือใหม่กว่า ที่จำเป็นเพื่อรันยูทิลีตี CCA Node Management (CNM)
3. ซอฟต์แวร์แพ็กเกจ **csufx.4765** ต้องถูกดาวน์โหลดจาก เว็บไซต์ http://www.ibm.com/security/cryptocards/pcixcc/ordersoftware.shtml ซอฟต์แวร์แพ็กเกจ มีชุดไฟล์ต่อไปนี้:
   - **csufx.4765.com** – 4765 CCA Support Program
   - **csufx.4765.cca** – 4765 Support Program – Common Utilities
   - **csufx.4765.man** – Support Program man pages

สิทธิ์การใช้ไฟล์

ไฟล์สิทธิ์การใช้ไฟล์โดยใช้ยูทิลีตี CCA Node Management (CNM)

ยูทิลีตี CCA Node Management (CNM) จัดให้มี แนวทางในการจัดการจุดควบคุมการเข้าถึง หากต้องการให้ความช่วยเหลือในเรื่องของการปกป้องความล้มเหลวแบบตั้งใจหรือไม่เจตนาของไฟล์เรียกทำงานของยูทิลิตี CNM ให้ตั้งค่าสิทธิในการเข้าถึงไฟล์ `CNM.jar` เพื่ออ่าน และเรียกใช้งานเท่านั้น เช่นเดียวกัน เมื่อต้องการปกป้องไฟล์ข้อมูลของจุด การควบคุมการเข้าถึง ให้ตั้งค่าสิทธิ์การใช้ไฟล์ของไฟล์ `csuap.def` เป็นอ่านเท่านั้น

การโหลดและการยกเลิกการโหลดซอฟต์แวร์เข้าสู่ตัวประมวลผลร่วม

หลังจากการติดตั้ง IBM Common Cryptographic Architecture (CCA) Support Program บนโฮสต์คอมพิวเตอร์ให้ใช้ Coprocessor Load Utility (CLU) เพื่อโหลดระบบปฏิบัติการของตัวประมวลผลร่วมและแอ็พพลิเคชัน CCA เข้าสู่ตัวประมวลผลร่วม

หากคุณขอรับอัพเดตกับส่วนสนับสนุนโปรแกรม ให้ใช้ CLU เพื่อโหลดเซ็กเมนต์โปรแกรมที่จำเป็นอีกครั้ง คุณยังสามารถโหลดซอฟต์แวร์ของผู้จำหน่าย โดยใช้ CLU

ส่วนนี้ประกอบด้วย:

- คำสั่งสำหรับการใช้ CLU เพื่อทำความเข้าใจตัวประมวลผลร่วม ที่ติดตั้งและสถานะของตัวประมวลผลร่วมเหล่านั้น และเพื่อติดตั้ง และถอนการติดตั้งซอฟต์แวร์ที่รันอยู่ภายในตัวประมวลผลร่วม
- ส่วนการอ้างอิงที่อธิบาย:
  - เซ็กเมนต์หน่วยความจำตัวประมวลผลร่วม
  - การตรวจสอบความถูกต้องของสถานะตัวประมวลผลร่วม
  - ไวยากรณ์ถูกใช้เพื่อเริ่มต้นยูทิลิตี CLU
  - โค้ดส่งคืน CLU

สำหรับการทำความเข้าใจเกี่ยวกับการควบคุมการโหลดโค้ดที่ละเอียดมากกว่านี้ และข้อควรพิจารณาด้านความปลอดภัยที่นำมาใช้โดยตัวประมวลผลร่วม โปรดดู เอกสารการค้นคว้าวิจัยเกี่ยวกับ *การสร้างผลการทำงานในระดับสูงสำหรับ Programmable Secure Coprocessor* ซึ่งมีอยู่บนเพจไลบรารีของเว็บไซต์ผลิตภัณฑ์ที่ http://www.ibm.com/security/cryptocards

หมายเหตุ:

1. ตำแหน่งไฟล์ที่อ้างถึงในส่วนนี้เป็นพาธไดเร็กทอรี ดีฟอลต์
2. โค้ดระบุความผิดพลาดที่ส่งคืนโดยไดร์เวอร์อุปกรณ์ตัวประมวลผลร่วมถูก แสดงในรูปแบบของเลขฐานสิบหกเช่น X'8040xxxx' คุณอาจพบข้อผิดพลาด โดยเฉพาะอย่างยิ่ง เมื่อคุณเริ่มใช้ยูทิลิตี CLU และมีความคุ้นเคยกับผลิตภัณฑ์และขั้นตอนเหล่านี้น้อยกว่า
3. function-control vector (FCV) ของตัวประมวลผลร่วมถูกโหลดโดยยูทิลิตี CCA Node Management (CNM)

ข้อมูลที่เกี่ยวข้อง:

"โค้ดระบุความผิดพลาดของไดร์เวอร์อุปกรณ์" ในหน้า 51
ไดร์เวอร์อุปกรณ์สำหรับตัวประมวลผลร่วมจะมอนิเตอร์สถานะของการสื่อสาร กับตัวประมวลผลร่วมและการลงทะเบียนสถานะฮาร์ดแวร์ของตัวประมวลผลร่วม

"การจัดการโหนดที่เข้ารหัสโดยใช้ยูทิลีตี CNM และ CNI" ในหน้า 18
คอมพิวเตอร์ที่จัดเตรียมเซอร์วิสการเข้ารหัสลับ เช่น การสร้างคีย์ และส่วนสนับสนุนลายเซ็นแบบดิจิทัลที่ได้นิยามไว้ที่นี่เป็น *โหนด การเข้ารหัสลับ*

การโหลดซอฟต์แวร์ตัวประมวลผลร่วม

ค้นหาโพรซีเดอร์เพื่อโหลดซอฟต์แวร์ไปยังตัวประมวลผลร่วม ในส่วนนี้

ดูที่ไฟล์ README ที่มากับการแจกจ่ายซอฟต์แวร์ ที่คุณกำลังติดตั้งที่ชื่อไฟล์ `.clu` ที่เจาะจง ไฟล์ README ยังจัดเตรียมข้อมูลเพิ่มเติม ที่เพิ่มหรือแก้ไขโพรซีเดอร์ทั่วไปเหล่านี

ใช้หัวข้อย่อยต่อไปนี้ ทำตามลำดับภารกิจนี้:

1. ที่พร้อมต์คำสั่ง เปลี่ยนเป็นไดเร็กทอรีที่มีไฟล์ Coprocessor Load Utility (CLU) และรัน CLU
2. กำหนดซอฟต์แวร์ที่ปัจจุบันตั้งอยู่ภายในตัวประมวลผลร่วม
3. เปลี่ยนแปลงเนื้อหาของซอฟต์แวร์ในเซ็กเมนต์ 1, 2 และ 3 ตามความเหมาะสม
4. ตรวจสอบเนื้อหาล่าสุดของซอฟต์แวร์ในเซ็กเมนต์

การเปลี่ยนไดเร็กทอรีดีฟอลต์และการรัน CLU

เมื่อต้องการเปลี่ยนไดเร็กทอรีดีฟอลต์ คุณต้องวางไดเร็กทอรีที่มี ไฟล์โค้ดตัวประมวลผลร่วม (*.clu) และ Coprocessor Load Utility (CLU)

การเปลี่ยนไดเร็กทอรีดีฟอลต์

ที่พร้อมต์คำสั่ง ให้เปลี่ยนเป็นไดเร็กทอรีโค้ดตัวประมวลผลร่วมของไดเร็กทอรีดีฟอลต์ `/usr/lpp/csufx.4765/clu` เพื่อเข้าถึงไฟล์โค้ด ถ้า CLU ไม่อยู่ใน ไดเร็กทอรีดีฟอลต์ ให้ตรวจสอบว่าระบบปฏิบัติการของคุณสามารถค้นหา CLU

การรัน CLU

หมายเหตุ: เมื่อใช้ CLU แอ็พพลิเคชันที่ใช้ CCA ต้องไม่รันอยู่

เมื่อต้องการรันยูทิลีตี CLU ให้ป้อนชื่อโปรแกรม **csufclu** ที่พร้อมต์คำสั่ง

คุณสามารถจัดเตรียมพารามิเตอร์แบบโต้ตอบกับยูทิลิตี CLU ได้ หรือ คุณสามารถรวมพารามิเตอร์เหล่านี้ไว้บนบรรทัดรับคำสั่ง ในแต่ละครั้งที่คุณใช้ CLU คุณต้องระบุชื่อล็อกไฟล์ ซึ่งเป็นพารามิเตอร์แรก และสามารถรวมไว้ในบรรทัดรับคำสั่งได้โดยทั่วไป เมื่อทำงานกับตัวประมวลผลร่วมเฉพาะ เป็นวิธีการดีที่สุดในการใช้หมายเลขลำดับของตัวประมวลผลร่วม เป็นชื่อล็อกไฟล์ คุณสามารถขอรับหมายเลขลำดับได้จากเลเบล บนเครื่องหมายวงเล็บเหลี่ยมที่ส่วนท้ายของตัวประมวลผลร่วม

CLU จะต่อท้ายข้อมูลไปยังล็อกไฟล์สองไฟล์ หากไม่มีล็อกไฟล์อยู่ ไฟล์เหล่านั้นจะถูกสร้างขึ้น หนึ่งล็อกไฟล์จะมีข้อมูลที่เหมือนกัน ซึ่งจะแสดงอยู่บนคอนโซลของคุณ ล็อกไฟล์อื่นๆ ที่ CLU จะกำหนด MRL เป็นส่วนขยายของชื่อไฟล์จะมี บันทึกการทำงาน ที่เครื่องสามารถอ่านได้ไฟล์ MRL ถูกใช้กับยูทิลิตีการวิเคราะห์

หมายเหตุ: คำสั่งเครื่องลำดับถัดมาในส่วนนี้สมมติว่าคุณใช้ CLU แบบโต้ตอบ เปลี่ยนไปเป็นไดเร็กทอรีที่มีไฟล์โค้ด ตัวประมวลผลร่วม เริ่มต้น CLU ด้วยชื่อที่เหมาะสมกับ ระบบปฏิบัติการของคุณ ให้ตอบกลับพร้อมต์ตามคำร้องขอ

CLU ขอรับจำนวนของตัวประมวลผลร่วมที่ติดตั้งไว้จาก ไดร์เวอร์อุปกรณ์ หากคุณมีมากกว่าหนึ่งตัวประมวลผลร่วมที่ติดตั้งไว้ CLU จะร้องขอจำนวนของตัวประมวลผลร่วมที่คุณตั้งใจจะโต้ตอบด้วย หมายเลข (coprocessor_*number*) มีค่าได้เป็น 0 – 2 เริ่มต้นด้วย 0 เพื่อให้หมายเลขเหล่านี้ สัมพันธ์กับตัวประมวลผลร่วม ให้ใช้ System Status (SS) เพื่อให้รู้ถึงจำนวนที่ตัวประมวลผลร่วม แต่ละตัวติดตั้งไว้ (ตัวอย่างของเอาต์พุตดูที่ รูปที่ 2 ในหน้า 18 ในหัวข้อคำสั่ง Coprocessor Load Utility)

หมายเหตุ: ยูทิลิตี CLU สามารถทำงานกับตัวประมวลผลร่วมได้ เมื่อขอรับการควบคุมเฉพาะ ของตัวประมวลผลร่วม หากแอ็พพลิเคชันอื่นเช่นเธรด กำลังรันอยู่ซึ่งได้ดำเนินการกับการเรียก CCA verb ตัวประมวลผลร่วมที่โหลดด้วย CCA จะ “ไม่ว่าง” และไม่สามารถใช้งานได้โดย CLU

ข้อมูลที่เกี่ยวข้อง:

“ไวยากรณ์สำหรับ Coprocessor Load Utility” ในหน้า 15

การกำหนดเนื้อหาเซ็กเมนต์ของซอฟต์แวร์ตัวประมวลผลร่วม

ตัวประมวลผลร่วมมีสามเซกเมนต์: segment 1, segment 2 และ segment 3 แต่ละเซกเมนต์มีสถานะ ถือซอฟต์แวร์และพับลิกคีย์ การตรวจสอบ และ identifier ของเจ้าของ (ยกเว้น segment 1)

ดูที่ ตารางที่ 1สำหรับข้อมูล เกี่ยวกับเซกเมนต์ของตัวประมวลผลร่วม

ตารางที่ 1. เนื้อหาเซ็กเมนต์ของซอฟต์แวร์

เซ็กเมนต์	เนื้อหา
1	Miniboot มีการวินิจฉัย และการควบคุมการโหลดโค้ด
2	โปรแกรมควบคุมแบบฝัง
3	CCA หรือแอ็พพลิเคชันอื่นๆ

คุณกำหนดเนื้อหาปัจจุบันและสถานะของเซ็กเมนต์ ตัวประมวลผลร่วมโดยใช้คำสั่ง **ST** รูปที่ 1 ในหน้า 11 แสดงการตอบกลับ ST ตามปกติ

```
====================================================================
CSUFCLU V4.1.1 st.log ST    begun Tue Sep 13 09:30:25 2011
********** Command ST started. ---- Tue Sep 13 09:30:25 2011

 *** VPD data;  PartNum = 45D5117
 *** VPD data;  EC Num = 0G43192
 *** VPD data;  Ser Num = 99000543
 *** VPD data;  Description = IBM 4765-001 PCI-e Cryptographic Coprocessor
 *** VPD data; Mfg. Loc. = 91
 *** ROM Status; POST0 Version 1, Release 27
 *** ROM Status; MiniBoot0 Version 1, Release 20
 *** ROM Status; INIT: INITIALIZED
 *** ROM Status; SEG2: RUNNABLE , OWNER2: 2
 *** ROM Status; SEG3: RUNNABLE , OWNER3: 2
 *** Page 1 Certified: YES
 *** Segment 1 Image: S0103 P1v0607 M1v011B P2v0706 F5180 201104151205401A000022000000000000
 *** Segment 1 Revision: 40105
 *** Segment 1 Hash: 177C AF13 C601 2276 90AA 8E20 D3BB BA58 79A6 7EBA 6C2A D68B 0A34 33E0 802C 4EA7
 *** Segment 1 Hash: 177C AF13
 *** Segment 2 Image: 4.1.7    y4_12-lnx-2011-03-04-16    201108111338401A000000000100010900
 *** Segment 2 Revision: 40107
 *** Segment 2 Hash: 698A 29DC EF8A 44D8 A025 3117 491B C552 45DA EC6F 0D0C 6671 BABE 7ABF 41E7 2FF5
 *** Segment 2 Hash: 698A 29DC
 *** Segment 3 Image: 4.1.7    CCA                        201108121155401A000000000000000000
 *** Segment 3 Revision: 40107
 *** Segment 3 Hash: EC02 B93A 309F 882A D859 031D 1F22 839D 2233 4D6A C58D D93C E43F 4A4C 1234 9F48
 *** Segment 3 Hash: EC02 B93A
 *** Query Adapter Status successful ***
Obtain Status ended successfully!
********** Command ST ended. ---- Tue Sep 13 09:31:26 2011

  ...finishing up...
********** Command ST exited. ---- Tue Sep 13 09:31:46 2011
```

รูปที่ 1. การตอบกลับสถานะ CLU แบบปกติ

นิยามของฟิลด์บนการตอบกลับ ST คือ:

ฟิลด์ คำอธิบาย

PartNum

หมายเลขชิ้นส่วน (P/N) ของตัวประมวลผลร่วม

EC Num

หมายเลขการเปลี่ยนแปลงทางวิศวกรรมของตัวประมวลผลร่วม

Ser Num

หมายเลขลำดับของผู้ผลิตตัวประมวลผลร่วม หมายเลขนี้ไม่ใช่หมายเลขลำดับการติดตาม IBM ที่ถูกใช้สำหรับการตรวจสอบการรับประกัน และดาวน์โหลดสิทธิ

คำอธิบาย

คำสั่งที่กล่าวถึงชนิดของตัวประมวลผลร่วมใน ข้อกำหนดทั่วไป ผู้ตรวจสอบต้องตรวจทานข้อมูลนี้และข้อมูลสถานะอื่นเพื่อยืนยัน ว่าตัวประมวลผลร่วมที่เหมาะสมถูกใช้อยู่

สถานะ ROM

ตัวประมวลผลร่วมต้องอยู่ในสถานะ INITIALIZED เสมอ หากสถานะคือ ZEROIZED ตัวประมวลผลจะตรวจพบเหตุการณ์ชักจูงที่อาจเกิดขึ้นได้ และอยู่ในสถานะที่ไม่สามารถกู้คืน และไม่มีการทำงาน(เหตุการณ์ชักจูงที่ เกิดขึ้นโดยบังเอิญถูกสร้างขึ้น หากตัวประมวลผลร่วมไม่ได้ถูกจัดการอย่างถูกต้อง ให้ถอด แบตเตอรี่ออกเฉพาะเมื่อคุณทำตามขั้นตอนที่แนะนำเพื่อเปลี่ยนแบตเตอรี่ ให้รักษา ตัวประมวลผลร่วมในช่วงอุณหภูมิที่ปลอดภัย และทำ ตามคำแนะนำ

ROM Status SEG2 / SEG3

เงื่อนไขสถานะต่างๆ สำหรับ Segment 2 และ Segment 3 จะมีอยู่ซึ่ง ประกอบด้วย:

- UNOWNED: ไม่ได้ใช้ในปัจจุบัน ไม่มีเนื้อหา
- RUNNABLE: มีโค้ด และอยู่ในสถานะที่สามารถใช้งานได้

เจ้าของ identifier ถูกแสดงดังนี้ ส่วนสนับสนุนโปรแกรม CCA มาตรฐานกำหนด identifier 2 สำหรับเซ็กเมนต์ 2 และเซ็กเมนต์ 3 **เจ้าของ identifier อื่น** ระบุว่า ซอฟต์แวร์ไม่ใช่โค้ดผลิตภัณฑ์ IBM CCA มาตรฐาน ในกรณีทั้งหมด ตรวจสอบว่า ซอฟต์แวร์ถูกโหลดลงในตัวประมวลผลร่วมของคุณ ซอฟต์แวร์ที่ไม่ได้รับอนุญาต หรือซอฟต์แวร์ที่ไม่รู้จักสามารถแสดงแทนค่าความเสี่ยงด้านความปลอดภัยให้กับการติดตั้งของคุณ

อิมเมจสำหรับเซ็กเมนต์ 1

ชื่อและคำอธิบายของเนื้อหาซอฟต์แวร์ของเซ็กเมนต์ 1 สำหรับตัวประมวลผลร่วมที่มาจากโรงงานชื่อจะมีคำว่า **Factory** อยู่ด้วย อิมเมจนี้และคีย์การตรวจสอบที่เชื่อมโยงกันต้องถูกเปลี่ยนแปลง

สำหรับ ตัวประมวลผลร่วมที่โหลดไว้ก่อนหน้านี้ชื่อ Segment 1 อาจประกอบด้วย CCA ตรวจสอบว่า คุณสังเกตเห็นระดับของการเปลี่ยนแปลง

อิมเมจสำหรับเซ็กเมนต์ 2 และเซ็กเมนต์ 3

หากเซ็กเมนต์เหล่านี้มีสถานะของตนเอง ให้สังเกตชื่ออิมเมจ และระดับของการเปลี่ยนแปลง IBM รวมกับ CCA ในชื่ออิมเมจ เพื่อบ่งชี้ว่า อิมเมจได้ถูกจัดเตรียมเป็นส่วนของส่วนสนับสนุนโปรแกรม CCA ตรวจสอบว่า ให้สังเกตระดับของการเปลี่ยนแปลง

ค่าการแฮชเซ็กเมนต์

ค่าการแฮชสำหรับแต่ละเซกเมนต์ต้องตรงกับค่าที่ถูก แสดงใน รูปที่ 1 ในหน้า 11

การเปลี่ยนเนื้อหาเซ็กเมนต์ของซอฟต์แวร์

ซอฟต์แวร์ภายในตัวประมวลผลร่วมต้องอยู่ที่ระดับของรีลีสเดียวกันกับ ซอฟต์แวร์ CCA ในระบบการสร้างโฮสต์

อย่าพยายามใช้ระดับรีลีสที่แตกต่างกันยกเว้นว่า จะได้รับคำแนะนำที่ระบุเฉพาะจาก IBM

เริ่มต้น Coprocessor Load Utility (CLU) และป้อนพารามิเตอร์ แบบโต้ตอบ สำหรับคำแนะนำดูที่ “การเปลี่ยนไดเร็กทอรีดีฟอลต์และการรัน CLU” ในหน้า 9

1. ป้อนชื่อล็อกไฟล์ (*nnnnnnnn*.**LOG** โดย *nnnnnnnn* คือเลขลำดับของตัวประมวลผลร่วม)
2. ป้อนคำสั่ง **PL**
3. ถ้าคุณไม่ตัวประมวลผลร่วมหลายตัว ให้ป้อนหมายเลขตัวประมวลผลร่วม
4. ป้อนชื่อไฟล์ CLU ตามที่ระบุในไฟล์ README

ทำซ้ำตามต้องการ เพื่อให้ซอฟต์แวร์ถูกโหลด สำหรับเซ็กเมนต์ 1, 2 และ 3

การตรวจสอบความถูกต้องของเนื้อหาเซ็กเมนต์ตัวประมวลผลร่วม

ขั้นตอนที่ต้องทำตามเพื่อตรวจสอบเนื้อหาของ เซกเมนต์ตัวประมวลผลร่วม

หลังจากที่คุณได้โหลดหรือแทนที่โค้ดในเซ็กเมนต์ 1, 2 และ 3 แล้ว ให้ใช้คำสั่ง **CLU VA** เพื่อยืนยันเนื้อหาเซ็กเมนต์และเพื่อตรวจสอบความถูกต้อง ของลายเซ็นแบบดิจิทัลบนการตอบกลับที่สร้างขึ้นโดยตัวประมวลผลร่วม

ขึ้นอยู่กับ IBM 4765 coprocessor (PartNum) ที่ใช้งานอยู่[1] เรียก คำสั่งต่อไปนี้ และแทนที่ชื่อไฟล์ใบรับรอง คีย์คลาสจาก ตารางที่ 2สำหรับชื่อไฟล์ ข้อมูล โปรดสังเกตว่าชื่อไฟล์ข้อมูล`v.clu` ถูกผนวกกับหมายเลข ชิ้นส่วนตัวประมวลผลร่วม ทั้งหมดเป็นตัวพิมพ์เล็ก

`csuxclu` *`nnnnnnnn`*`.log VA [coprocessor_`*`n`*`] `*`datafile`*

หมายเลขชิ้นส่วนสามารถขอรับได้โดยใช้คำสั่ง Coprocessor Load Utility (CLU) ST

ตารางที่ 2. ไฟล์ Class-key สำหรับใช้กับคำสั่ง CLU VA

PartNum	ไฟล์ใบรับรอง Class-key
12R8565	12r8565v.clu
41U0441	41u0441v.clu

พารามิเตอร์ [coprocessor_*n*] เป็น ตัวออกแบบทางเลือกสำหรับตัวประมวลผลร่วมเฉพาะและมีค่าดีฟอลต์ เป็นศูนย์

การยกเลิกการโหลดซอฟต์แวร์ตัวประมวลผลร่วมและ zeroize โหนด CCA

ขั้นตอนในการยกเลิกการโหลดซอฟต์แวร์ตัวประมวลผลร่วม และเพื่อล้างข้อมูล โหนด CCA เพื่อสละความเป็นเจ้าของเซ็กเมนต์ที่อธิบาย ที่นี่

เมื่อคุณใช้ Coprocessor Load Utility (CLU) เพื่อประมวลผลไฟล์ที่ สละความเป็นเจ้าของเซ็กเมนต์ 2 ทั้งเซ็กเมนต์ 2 และเซ็กเมนต์ 3 ที่เป็นส่วนย่อยจะถูกล้างค่า และลบโค้ดออก พับลิกคีย์การ ตรวจสอบความถูกต้องสำหรับเซ็กเมนต์ถูกล้างค่า รายการข้อมูลที่เกี่ยวกับความปลอดภัย ที่เก็บอยู่ภายในตัวประมวลผลร่วมสำหรับเซ็กเมนต์ถูกล้างข้อมูล ตัวบ่งชี้ เจ้าของถูกล้างค่า และสถานะของเซ็กเมนต์ถูกตั้งค่าเป็น UNOWNED

ดูที่ไฟล์ README ที่มากับการแจกจ่าย ซอฟต์แวร์ที่คุณกำลังใช้สำหรับชื่อไฟล์ `.clu` ที่ระบุ เฉพาะที่ใช้เพื่อสละความเป็นเจ้าของเซ็กเมนต์ 2 และ 3 ไฟล์ README ยังอาจระบุข้อมูลเพิ่มเติมที่ขยาย หรือแก้ไขโพรซีเดอร์ทั่วไปนี้

ดำเนินการกับการดำเนินการเหล่านี้:

- เปลี่ยนเป็นไดเร็กทอรีที่มีไฟล์ CLU
- เริ่มต้นยูทิลีตี CLU
- ตอบกลับพร้อมต์และใช้หมายเลขลำดับของตัวประมวลผลร่วม ในชื่อล็อกไฟล์
- ใช้คำสั่ง **PL** เพื่อปล่อยเซ็กเมนต์ 2 ตามที่กล่าวถึงในไฟล์ README สำหรับแพล็ตฟอร์มของคุณ

หมายเหตุ:

1. คุณยัง zeroize CCA โดยไม่ลบซอฟต์แวร์โดยใช้กระบวนการเตรียมข้อมูลเบื้องต้นให้กับ CCA อีกครั้ง

1. คุณสามารถอ้างถึงเว็บไซต์ผลิตภัณฑ์ IBM (http://www.ibm.com/security/cryptocards) ส่วนของ FAQ สำหรับโพรซีเดอร์เพื่อตรวจสอบความสมบูรณ์ของตัวประมวลผลร่วม หัวข้อนั้น จะมีรายการของไฟล์ใบรับรองคีย์คลาสปัจจุบันอยู่

2. IBM ไม่ได้พร้อมใช้งานกับไฟล์ เพื่อเรียกคืนเซ็กเมนต์ 1 จากโรงงานที่ตรวจสอบคีย์เพื่อวางตัวประมวลผลร่วม ลงในเงื่อนไขที่คล้ายกับผลิตภัณฑ์จากโรงงาน เซ็กเมนต์ 1 สามารถเปลี่ยนแปลง จำนวนครั้งที่จำกัดไว้ก่อนที่พื้นที่ในใบรับรองคีย์อุปกรณ์ที่มีอยู่จะถูกใช้ และ ตัวประมวลผลอาจเป็นไปได้ที่จะ render โดยที่ไม่สามารถ ใช้งานได้ ถ้าคุณต้องการความสามารถในการเรียกคืน คีย์การตรวจสอบของเซ็กเมนต์ 1 และต้องการแสดงตัวประมวลผลร่วมของคุณตามเงื่อนไขในการ ล็อก คุณสามารถขอรับไฟล์ที่จำเป็นต้องมีได้จาก IBM โดยการส่งเคียวรีที่ใช้แบบฟอร์มส่วนสนับสนุนบนเว็บไซต์ผลิตภัณฑ์ที่ http://www.ibm.com/security/cryptocards มีสิ่งสำคัญ ที่ต้องจดบันทึกไว้ว่า พื้นที่ในใบรับรองไม่ใช่รีซอร์สที่สามารถสร้างขึ้นใหม่ได้ หากนำมาใช้หมด พื้นที่นั้นจะไม่สามารถกู้คืนได้

ข้อมูลที่เกี่ยวข้อง:

“การเตรียมข้อมูลเบื้องต้นให้กับโหนด” ในหน้า 27
ขั้นตอนในการเตรียมข้อมูลเบื้องต้นโหนด CCA ให้กับสภาวะ เริ่มต้น

การอ้างถึง Coprocessor Load Utility (CLU)

เซ็กเมนต์หน่วยความจำตัวประมวลผลร่วมที่คุณโหลดซอฟต์แวร์ จะถูกอธิบายที่นี่ วิธีการที่ตัวประมวลผลร่วมใช้เพื่อตรวจสอบความถูกต้องของ ซอฟต์แวร์ที่โหลด ไวยากรณ์ที่ใช้เริ่มต้น CLU และโค้ดส่งคืน CLU

ถ้าคุณไม่ต้องการรายละเอียดในส่วนนี้ ให้ข้ามไปยัง “การจัดการโหนดที่เข้ารหัสโดยใช้ยูทิลีตี CNM และ CNI” ในหน้า 18

เซ็กเมนต์หน่วยความจำตัวประมวลผลร่วม

ตัวประมวลผลร่วมเซ็กเมนต์หน่วยความจำมี การจัดระเบียบเป็นกลุ่มต่าง ๆ

องค์ประกอบของเซกเมนต์หน่วยความจำ และฟังก์ชันดังต่อไปนี้:

ตารางที่ 3. การจัดการเซ็กเมนต์หน่วยความจำ

เซ็กเมนต์	คำอธิบาย
0	โค้ดระดับต้น โค้ดระดับต้นจัดการกับการกำหนดค่าเริ่มต้นตัวประมวลผลร่วม และอินเตอร์เฟสของฮาร์ดแวร์คอมโพเนนต์ โค้ดนี้ไม่สามารถ เปลี่ยนแปลงได้หลังจากที่ตัวประมวลผลร่วมออกจากโรงงานแล้ว
1	ซอฟต์แวร์การควบคุมดูแลและรูทีนการเข้ารหัสลับ ซอฟต์แวร์ในเซ็กเมนต์นี้: • ดูแลการแทนที่ซอฟต์แวร์ที่โหลดลงในเซ็กเมนต์ 1 • ดูแลการโหลดข้อมูลและซอฟต์แวร์ลงในเซ็กเมมนต์ 2 และ 3 • โหลดที่โรงงาน แต่สามารถแทนที่ได้โดยใช้ยูทิลิตี CLU
2	ระบบปฏิบัติการแบบฝัง ตัวประมวลผลร่วมสนับสนุน โปรแกรมประกอบด้วยระบบปฏิบัติการ ระบบปฏิบัติการสนับสนุนแอพพลิเคชัน ที่โหลดเข้าสู่ Segment 3 และ Segment 2 ว่างเปล่า เมื่อตัวประมวลผลร่วมถูกจัดส่งมาจากโรงงาน
3	แอ็พพลิเคชันซอฟต์แวร์ ส่วนสนับสนุนโปรแกรม ตัวประมวลผลร่วมประกอบด้วยแอ็พพลิเคชันโปรแกรม CCA ที่สามารถติดตั้งได้ในเซ็กเมนต์ 3 การทำงานของแอ็พพลิเคชันตามลำดับ IBM CCA และดำเนินการควบคุมสิทธิในการเข้าถึง การจัดการกับคีย์ และการดำเนินการเข้ารหัส เซ็กเมนต์ 3 ว่างเปล่าเมื่อตัวประมวลผลร่วมถูกจัดส่งมาพร้อมกับโรงงาน

การตรวจสอบความถูกต้องโหลดของซอฟต์แวร์ตัวประมวลผลร่วม

เมื่อตัวประมวลผลร่วมถูกจัดส่งจากโรงงาน ตัวประมวลผลร่วม จะอยู่ภายในพับลิกคีย์ที่จำเป็นต่อการตรวจสอบการแทนที่ซอฟต์แวร์ สำหรับเซ็กเมนต์ 1

เมื่อต้องการโหลดเข้าสู่ตัวประมวลผลร่วม Segment 2 และ Segment 3 สำหรับ แต่ละเซกเมนต์ให้ทำตามขั้นตอนเหล่านี้:

1. ระบุเจ้าของสำหรับเซ็กเมนต์โดยใช้คำสั่ง **สร้างเจ้าของ** identifier เจ้าของจะถูกยอมรับ หากลายเซ็นดิจิทัลที่เชื่อมโยงกับ identifier นี้ สามารถตรวจสอบความถูกต้องได้โดยพับลิกคีย์ที่ตั้งอยู่พร้อมกับเซ็กเมนต์ที่อยู่ต่ำกว่าโดยทันที หากสร้างขึ้นแล้ว ความเป็นเจ้าของจะยังคงมีผลบังคับใช้ จนกว่าคำสั่ง **Surrender Owner** ถูกประมวลผลโดยตัวประมวลผลร่วม
2. โหลดเซกเมนต์ไปที่โค้ด ซึ่งมีคำสั่งที่แตกต่างกัน สองคำสั่งที่พร้อมใช้งาน
   a. เริ่มต้นใช้คำสั่ง **Load** ข้อมูลคำสั่ง **Load** นใบรับรองพับลิกคีย์ที่ต้องถูกตรวจสอบโดย พับลิกคีย์ที่แสดงบนเซกเมนต์ที่ต่ำกว่าถัดไป ตัวประมวลผลร่วม ยอมรับโค้ดและคงพับลิกคีย์ที่ตรวจสอบแล้วสำหรับเซกเมนต์ ถ้าตรงกับหนึ่งในเงื่อนไข:
      - ใบรับรองถูกต้อง
      - ข้อมูลของ identifier เจ้าของในคำสั่ง **Load** ตรงกับความเป็นเจ้าของปัจจุบัน ที่ถืออยู่โดยตัวประมวลผลร่วม สำหรับ เซกเมนต์
      - ข้อมูลสมบูรณ์ในคำสั่ง **Load** สามารถถูก ตรวจสอบโดยพับลิกคีย์ในใบรับรองที่ถูกใช้สำหรับการตรวจสอบความถูกต้อง
   b. หากเซ็กเมนต์ยังคงมีพับลิกคีย์คำสั่ง **Reload** สามารถถูกใช้เพื่อแทนที่โค้ดในเซกเมนต์ การดำเนินการกับ ตัวประมวลผลร่วมเป็นการดำเนินการเดียวกันสำหรับคำสั่ง **Load** ยกเว้นว่า ใบรับรองที่รวมไว้ต้องถูกตรวจสอบโดยพับลิกคีย์ที่เชื่อมโยงกับเซ็กเมนต์เป้าหมายแทนคีย์ที่เชื่อมโยง กับเซ็กเมนต์ถัดไป

ระบบปฏิบัติการที่ฝังไว้ซึ่งทำงานกับฮาร์ดแวร์ตัวประมวลผลร่วม สามารถเก็บ security-relevant data items (SRDIs) ในฐานะเป็นตัวแทนของตนเอง และแอ็พพลิเคชันใน Segment 3. SRDIs ถูก zeroized ตามการปกป้อง การชักจูง การโหลดซอฟต์แวร์เซ็กเมนต์ หรือการประมวลผลคำสั่ง **Surrender Owner** ของเซ็กเมนต์ SRDIs สำหรับเซ็กเมนต์ไม่ถูก zeroized เมื่อคำสั่ง **Reload** ถูกใช้ แอ็พพลิเคชัน CCA เก็บคีย์หลัก, function control vector (FCV), ตารางการควบคุมการเข้าถึงและไพรเวตคีย์ RSA ที่เก็บไว้ เป็นข้อมูล SRDI ที่ถูกเชื่อมโยงกับ Segment 3

IBM ลงนามซอฟต์แวร์ของตนเอง ถ้าผู้จำหน่าย อื่นต้องการจัดหาซอฟต์แวร์สำหรับตัวประมวลผลร่วมคำสั่ง **Establish Owner** ของผู้จำหน่ายนั้น และใบรับรองพับลิกคีย์การลงนามโค้ด ต้อง ถูกลงนามโดย IBM ภายใต้สัญญาที่เหมาะสม ข้อจำกัดเหล่านี้ทำให้แน่ใจได้ว่าเป็นไปตามเงื่อนไขต่อไปนี้:

- เฉพาะโค้ดที่ได้รับสิทธิสามารถโหลดลงในตัวประมวลผลร่วม
- ข้อจำกัดในรัฐบาลจะตรงกับการนำการเข้ารหัสลับไปใช้งานสำหรับ การอิมพอร์ตและเอ็กซ์พอร์ต

ไวยากรณ์สำหรับ Coprocessor Load Utility

ไวยากรณ์ที่ใช้เริ่มต้น Coprocessor Load Utility (CLU) และฟังก์ชันของ ยูทิลีตีจะได้รับการอธิบาย

CLU ต้องถูกใช้สำหรับฟังก์ชันต่อไปนี้:

---

2. ในเอกสารนี้จะใช้คำว่า *load* และ *reload* เอกสารคู่มืออื่นๆ อาจอ้างถึงการดำเนินการเหล่านี้เป็น *emergency burn* (EmBurn) และ *regular burn* หรือ *remote burn* (RemBurn)

- ให้ตรวจสอบว่าตัวประมวลผลร่วมว่างอยู่โดยการสิ้นสุดแอ็พพลิเคชัน ที่ใช้งานตัวประมวลผลร่วม ตัวอย่างเช่น จบแอ็พพลิเคชันทั้งหมด ที่ใช้ CCA API
- ขอรับระดับของการรีลีสและสถานะของซอฟต์แวร์ที่ติดตั้งอยู่ ในเซ็กเมนต์หน่วยความจำของตัวประมวลผลร่วม
- ยืนยันความถูกต้องของข้อความที่ลงนามซึ่งส่งคืนโดย ตัวประมวลผลร่วม
- โหลดและรีโหลดส่วนของซอฟต์แวร์ตัวประมวลผลร่วม
- รีเซ็ตตัวประมวลผลร่วม

เมื่อต้องการเริ่มต้นยูทิลีตี ทำตามขั้นตอนเหล่านี้:

1. ล็อกออนตามที่ต้องการโดยระบบปฏิบัติการของคุณ
2. ที่บรรทัดรับคำสั่ง เปลี่ยนไดเร็กทอรีเป็นไดเร็กทอรี ที่มีไฟล์ CLU ไดเร็กทอรีดีฟอลต์คือ `/usr/lpp/csufx.4765/clu`
3. ป้อนชื่อยูทิลีตี **csufclu** ตามด้วยพารามิเตอร์ที่ ใช้ได้

ถ้าคุณไม่ระบุพารามิเตอร์ที่จำเป็น ยูทิลีตีจะพร้อมต์เมื่อต้องการ ข้อมูล พารามิเตอร์เผื่อเลือกถูกล้อมรอบอยู่ใน เครื่องหมายวงเล็บเหลี่ยม ไวยากรณ์สำหรับพารามิเตอร์ที่ต่อจาก ชื่อยูทิลีตีคือ

`[`*`log_filecmd`*`[`*`coprocessor _#`*`][`*`data_file`*`][`*`-Q`*`]]`

สถานที่:

log_file
: ระบุชื่อล็อกไฟล์ ยูทิลิตีต่อท้ายรายการกับไฟล์ข้อความ ASCII นี้ตามที่ดำเนินการตามที่ร้องขอ ล็อกไฟล์ที่เครื่องอ่านได้ที่สอง โดยมีชื่อไฟล์เป็น logfile_name MRL ถูกสร้างขึ้นด้วยไฟล์บันทึกการทำงานนี้ สามารถประมวลผลได้โดยโปรแกรมและมีการตอบกลับที่เข้ารหัสแบบไบนารีไว้ จากตัวประมวลผลร่วม

cmd
: ระบุตัวย่อสองตัวอักษรที่แสดงคำสั่งโหลดเดอร์ ที่ต้องรัน

coprocessor *_number*
: จัดเตรียมหมายเลขตัวประมวลผลร่วมตามที่สร้างขึ้นโดยไดร์เวอร์อุปกรณ์ พารามิเตอร์นี้มีดีฟอลต์เป็น 0 ตัวประมวลผลร่วมถูก กำหนดให้กับไดร์เวอร์อุปกรณ์เป็นหมายเลข 0, 1 และ 2 คุณสามารถใช้ข้อมูล หมายเลขลำดับที่คุณขอรับด้วยคำสั่ง **ST** หรือ **VA** และหมายเลขลำดับ ที่พิมพ์ไว้หลังเครื่องหมายวงเล็บเหลี่ยมของตัวประมวลผลร่วม เพื่อให้สัมพันธ์กับ ตัวประมวลผลร่วมเฉพาะกับ coprocessor *_number* ยูทิลีตี รอบรับตัวประมวลผลร่วมสูงสุดแปดตัวต่อหนึ่งระบบ

data_file
: ระบุไฟล์ข้อมูล (ไดร์ฟ ไดเร็กทอรี และชื่อไฟล์) ที่ ใช้สำหรับการดำเนินการที่ร้องขอ เมื่อต้องการระบุชื่อ *data_file* ให้ใช้หนึ่งในวิธีต่อไปนี้:
  - สำหรับซอฟต์แวร์ที่โหลดและรีโหลด ชื่อ *data_file* คือชื่อไฟล์ของอิมเมจซอฟต์แวร์ที่คุณกำลังโหลดเข้าสู่ ตัวประมวลผลร่วม ไฟล์ Support Program README ระบุชื่อ *data_file*
  - สำหรับตัวประมวลผลร่วม สถานะตัวประมวลผลร่วมขอรับจากคำสั่ง **VA** ชื่อ *data_file* คือชื่อไฟล์ใบรับรอง class-key ที่ใช้เพื่อตรวจสอบความถูกต้อง ของการตอบกลับของตัวประมวลผลร่วม ส่วน FAQ ของเว็บไซต์ผลิตภัณฑ์ (http://www.ibm.com/security/cryptocards) มีคำอธิบายของ ขั้นตอนสำหรับการตรวจสอบความถูกต้องตัวประมวลผลร่วมและโค้ดของตัวประมวลผลร่วม คำอธิบายนี้ยังมีรายการของ ชื่อไฟล์ใบรับรอง class-key ปัจจุบัน คุณสามารถดาวน์โหลดไฟล์ใบรับรองที่จำเป็นใดๆ ได้จากเว็บไซต์

-Q
: หยุด (quiets) โปรแกรม CLU ที่เอาต์พุตไปยังอุปกรณ์เอาต์พุตมาตรฐาน ข้อมูลสถานะยังคงต่อท้าย ไฟล์บันทึกการทำงาน

ตัวอย่างเช่น: หากต้องการขอรับสถานะของตัวประมวลผลร่วม และบันทึกผลลัพธ์ลงในไฟล์ บันทึกการทำงาน ให้ป้อน:

csufclu nnnnnnnn.log va *datafile_name*.**clu**

ขอแนะนำว่าคุณควรทำ *nnnnnnnn* หมายเลขลำดับ ของตัวประมวลผลร่วม ซึ่งไม่ใช่การบังคับให้ใช้หมายเลขลำดับ แต่ใช้เพื่อเก็บประวัติของการเปลี่ยนแปลงของซอฟต์แวร์ที่ทำกับ ตัวประมวลผลร่วมเฉพาะแต่ละตัว

ข้อมูลที่เกี่ยวข้อง:

"เนื้อหาของบันทึกการทำงานที่เครื่องสามารถอ่านได้" ในหน้า 50

ยูทิลิตี CLU สร้างล็อกไฟล์สองไฟล์ ไฟล์หนึ่งสำหรับการอ่าน และอีกไฟล์หนึ่งสำหรับอินพุตไปยังโปรแกรม

"คำสั่ง Coprocessor Load Utility"

Coprocessor Load Utility (CLU) สนับสนุนคำสั่ง loader หลายคำสั่ง

คำสั่ง Coprocessor Load Utility:

Coprocessor Load Utility (CLU) สนับสนุนคำสั่ง loader หลายคำสั่ง

คำสั่ง loader และฟังก์ชันที่ได้รับการสนับสนุนโดย CLU มีดังต่อไปนี้:

ตารางที่ 4. คำสั่ง CLU loader

คำสั่ง Loader	**คำอธิบาย**
PL: โหลดไมโครโค้ดลงในตัวประมวลผลร่วม คำสั่ง R1, E2, L2, R2, S2, E3, L3, R3 และ S3 ถูกอนุมานจากข้อมูล ที่มีอยู่ในไฟล์ข้อมูลที่คุณใช้กับคำสั่ง PL ไฟล์ "PL" เดี่ยว สามารถรวมข้อมูลเข้าด้วยกันสำหรับความเป็นเจ้าของและการโหลดคำสั่งจำนวนมาก	การประมวลผลชุดของคำสั่งโดยตรงจากเนื้อหา ของไฟล์ข้อมูลเพื่อสร้างความเป็นเจ้าของเซ็กเมนต์ และเพื่อโหลดหรือรีโหลดซอฟต์แวร์เซ็กเมนต์
RS: รีเซ็ตตัวประมวลผลร่วม	รีเซ็ตตัวประมวลผลร่วม โดยทั่วไป คุณจะไม่ใช้คำสั่งนี้ คำสั่งทำให้ตัวประมวลผลร่วมดำเนินการรีเซ็ต การเปิดเครื่อง คุณอาจพบว่าคำสั่งนี้มีประโยชน์ซึ่งตัวประมวลผลร่วม และซอฟต์แวร์ของระบบโฮสต์สูญเสียการซิงโครไนซ์ คุณควรจบการประมวลผล ซอฟต์แวร์ของระบบโฮสต์ที่กำลังทำงานอยู่พร้อมกับตัวประมวลผลร่วม ก่อนที่จะใช้คำสั่งนี้เพื่อเปิดใช้งานระบบย่อยการเข้ารหัสลับที่สมบูรณ์ในการขอรับสถานะการรีเซ็ต
SS: รับสถานะระบบ	ขอรับหมายเลขชิ้นส่วน หมายเลขลำดับ และส่วนของชื่ออิมเมจของซอฟต์แวร์เซ็กเมนต์ 3 สำหรับแต่ละตัวประมวลผลร่วม ที่ติดตั้งไว้ ซึ่งได้เตรียมการว่า ตัวประมวลผลร่วมเหล่านี้ไม่ได้ถูกใช้โดยแอ็พพลิเคชัน บางตัว เช่น CCA โปรดดู รูปที่ 2 ในหน้า 18
ST: ขอรับสถานะตัวประมวลผลร่วม	ขอรับสถานะของซอฟต์แวร์ที่โหลดแล้ว และระดับของรีลีสของคอมโพเนนต์อื่นๆ สถานะถูกต่อท้าย ล็อกไฟล์
VA: ตรวจสอบความถูกต้องของสถานะของตัวประมวลผลร่วม	ขอรับสถานะของซอฟต์แวร์ที่โหลดแล้ว และระดับของรีลีสของคอมโพเนนต์อื่นๆ ข้อมูลถูกส่งในข้อความที่ลงนามโดย คีย์อุปกรณ์ตัวประมวลผลร่วม จากนั้นจะถูกเก็บไว้ในล็อกไฟล์ ยูทิลิตี ยูทิลิตีใช้พับลิกคีย์ในตัวเพื่อตรวจสอบความถูกต้อง ของใบรับรอง class-key ตั้งแต่หนึ่งฉบับขึ้นไปที่มีอยู่ในพารามิเตอร์ชื่อ *data_file* หนึ่งในใบรับรองเหล่านี้ควรตรวจสอบความถูกต้องของพับลิกคีย์ หรือลูกโซ่ของพับลิกคีย์ ซึ่งขอรับจากตัวประมวลผลร่วม และยืนยันว่า ตัวประมวลผลร่วมไม่ได้ถูกเปลี่ยน

โดยปกติแล้ว ยูทิลิตีสามารถเรียกใช้งานได้โดยไฟล์สคริปต์ หรือไฟล์คำสั่ง เมื่อคุณสร้างไฟล์สคริปต์หรือไฟล์คำสั่ง เพื่อเริ่มต้นยูทิลีตีบนระบบที่ไม่เจาะจง ให้เพิ่มไวยากรณ์ "quiet" พารามิเตอร์ **-q** (หรือ **-Q, /q,** หรือ **/Q**) ให้กับ การร้องขอที่ไม่มีเอาต์พุตถูกส่งไปยังจอแสดงผล ตามค่าดีฟอลต์แล้ว ยูทิลิตีส่งคืน พร้อมต์และข้อความไปยังจอแสดงผล

ตัวอย่าง *การโต้ตอบสถานะระบบ CLU ปกติ* แสดง การโต้ตอบของระบบ CLU

```
======================================================================
CSUFCLU V4.00 ss.log SS    begun Tue Sep 28 10:49:36 2010
*********** Command SS started. ---- Tue Sep 28 10:49:36 2010

 Card #    P/N        S/N         Segment 3 Description
 ------    -------    --------    ------------------------------------
   0       45D6045    99000627    4.1.0    CCA
 *** Query System Status successful ***
System Status ended successfully!
*********** Command SS ended. ---- Tue Sep 28 10:50:37 2010

  ...finishing up...
*********** Command SS exited. ---- Tue Sep 28 10:50:57 2010
```

รูปที่ 2. การตอบกลับ สถานะของระบบ CLU แบบปกติ

โค้ดส่งคืน Coprocessor Load Utility

ส่วนนี้ระบุค่าโค้ดส่งคืนจาก CLU

เมื่อ CLU เสร็จสิ้นการประมวลผลแล้ว CLU จะส่งคืนค่าที่สามารถทดสอบได้ ในไฟล์สคริปต์หรือในไฟล์คำสั่ง แต่ละค่าที่ส่งคืนมีความหมาย

0 ตกลง นี่แสดงว่า CLU เสร็จสิ้นการประมวลผลอย่างถูกต้อง

1 พารามิเตอร์บรรทัดรับคำสั่งไม่ถูกต้อง

2 ไม่สามารถเข้าถึงตัวประมวลผลร่วมได้ ในกรณีนี้ให้ตรวจสอบให้แน่ใจว่า ตัวประมวลผลร่วม และไดร์เวอร์ได้ถูกติดตั้งไว้อย่างถูกต้อง

3 ตรวจสอบล็อกไฟล์ยูทิลิตีสำหรับรายงานเงื่อนไขผิดปกติ

4 ไม่มีการติดตั้งตัวประมวลผลร่วม ในกรณีนี้ให้ตรวจสอบให้แน่ใจว่า ตัวประมวลผลร่วม และไดร์เวอร์ได้ถูกติดตั้งไว้อย่างถูกต้อง

5 มีการระบุหมายเลขตัวประมวลผลร่วมที่ไม่ถูกต้อง

6 ไฟล์ข้อมูลจำเป็นต้องมีสำหรับคำสั่งนี้

7 ไฟล์ข้อมูลที่ระบุไว้ด้วยคำสั่งนี้ไม่ถูกต้องหรือใช้งานไม่ได้

การจัดการโหนดที่เข้ารหัสโดยใช้ยูทิลีตี CNM และ CNI

คอมพิวเตอร์ที่จัดเตรียมเซอร์วิสการเข้ารหัสลับ เช่น การสร้างคีย์ และส่วนสนับสนุนลายเซ็นแบบดิจิทัลที่ได้นิยามไว้ที่นี่เป็น *โหนด การเข้ารหัสลับ*

ยูทิลิตี CCA Node Management (CNM) และยูทิลิตี CCA Node Initialization (CNI) ที่จัดเตรียมไว้พร้อมกับส่วนสนับสนุนนี้ คือเครื่องมือที่ใช้ในการตั้งค่า และจัดการกับเซอร์วิสการเข้ารหัสลับของ CCA ที่จัดเตรียมไว้โดยโหนด

ส่วนนี้ประกอบด้วย:

- ยูทิลีตีและรายละเอียดเกี่ยวกับวิธีเริ่มใช้งาน
- สถานการณ์จำลองตัวอย่างสำหรับการใช้ยูทิลีตีที่คุณอาจต้องพิจารณา
- วิธีใช้ฟังก์ชันการทำงานของยูทิลีตี CNM: ตรวจทาน สื่อประกอบหลังจากที่ทำงานผ่านหัวข้อ“สถานการณ์จำลอง: การสร้างโหนดทดสอบ” ในหน้า 21
- วิธีสร้างและจัดการข้อมูลการควบคุมการเข้าถึง: อ่านรายละเอียดเกี่ยวกับ ส่วนการควบคุมการเข้าถึงของยูทิลีตี CNM
- วิธีการจัดการกับคีย์การเข้ารหัสลับ: การจัดการกับคีย์ ที่คุณสามารถทำให้บรรลุได้ด้วยยูทิลิตี CNM
- วิธีสร้างโหนดอื่นโดยใช้ยูทิลีตี CNI: คุณสามารถ ใช้ยูทิลีตี CNM แบบอัตโนมัติโยใช้โพรซีเดอร์ที่ห่อหุ้มไว้

ยูทิลีตีเหล่านี้ถูกเขียนใน Java™ และ ต้องการใช้ Java runtime environment (JRE) คุณยังสามารถใช้ Java Development Kit (JDK)

ภาพรวม CNM และ CNI

ผู้ใช้ทั่วไปของยูทิลีตี CCA Node Management (CNM) และยูทิลีตี CCA Node Initialization (CNI) คือบุคคลผู้ดูแลระบบความปลอดภัย ผู้พัฒนาแอ็พพลิเคชัน ผู้ดูแลระบบและในบางกรณี ผู้ดำเนินการ กับโหมดที่ใช้งานจริง

หมายเหตุ:

1. ยูทิลิตี CNM ตกแต่งชุดของเซอร์วิส CCA API ที่จำกัด หลังจากที่คุ้นเคยกับยูทิลิตีแล้ว คุณสามารถกำหนดให้ตรงกับความต้องการของคุณ หรือคุณอาจต้องมีแอ็พพลิเคชันแบบกำหนดเองเพื่อบรรลุ การควบคุมที่ครอบคลุมเพิ่มเติมและการจัดการกับคีย์
2. ไฟล์ที่คุณสร้างผ่านการใช้ยูทิลิตี CNM อาจขึ้นอยู่กับ รีลีสของสภาพแวดล้อมแบบรันไทม์ของ Java Runtime Environment (JRE) หากคุณเปลี่ยนรีลีสของสภาพแวดล้อมแบบรันไทม์ของ Java Runtime Environment (JRE) ที่คุณใช้ไฟล์ที่คุณได้สร้างขึ้นด้วยยูทิลิตี CNM อาจทำงานไม่ถูกต้องกับรีลีสใหม่นี้
3. ยูทิลิตี CNM ได้ถูกออกแบบมาเพื่อใช้กับเมาส์ใช้ เมาส์แทนคีย์ **Enter** สำหรับผลลัพธ์ที่สอดคล้องกัน
4. ไม่มีพาเนลวิธีใช้ที่จัดเตรียมไว้สำหรับส่วนของการโคลนคีย์หลัก ของยูทิลิตี
5. ยูทิลิตีเหล่านี้ใช้ IBM Common Cryptographic Architecture (CCA) Support Program API เพื่อร้องขอเซอร์วิสจากตัวประมวลผลร่วม คู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* มีรายชื่อที่ครอบคลุมของ verbs (ซึ่งรู้จักกันว่าเซอร์วิสที่เรียกได้หรือการเรียกโพรซีเดอร์) จัดเตรียมโดย CCA API โปรดอ้างถึงหนังสือนี้และเซอร์วิสแต่ละตัว ที่กล่าวไว้ เพื่อทำความเข้าใจถึงคำสั่งที่อาจต้องการการพิสูจน์ตัวตน ในบทบาทต่างๆ ที่คุณจะนิยามไว้โดยใช้โพรซีเดอร์ที่กล่าวถึง ในส่วนนี้

ภาพรวมยูทิลีตีการจัดการโหนด CCA

ยูทิลิตี CCA Node Management คือแอ็พพลิเคชัน Java ที่จัดเตรียมอินเตอร์เฟสแบบกราฟิกเพื่อใช้ในการติดตั้งและคอนฟิกูเรชันของโหนด IBM 4765 CCA cryptographic ยูทิลิตี ทำหน้าที่หลักในการติดตั้งโหนด สร้างและจัดการกับข้อมูลการควบคุมสิทธิในการเข้าถึง และจัดการกับคีย์หลัก CCA ที่จำเป็นต่อการดูแลโหนด การเข้ารหัสลับ

คุณสามารถโหลดข้อมูลอ็อบเจ็กต์ได้โดยตรงไปยังตัวประมวลผลร่วม หรือบันทึกลงในดิสก์ อ็อบเจ็กต์ข้อมูลมีประโยชน์สำหรับโหนด IBM 4765 CCA อื่นๆ ที่ใช้ระบบปฏิบัติการเดียวกัน และระดับของการทำงานร่วมกันได้ของแอ็พพลิเคชัน Java

หมายเหตุ: การเริ่มยูทิลีตี CCA Node Management: เมื่อต้องการเริ่มยูทิลีตี CCA Node Management ให้ป้อน คำสั่ง **csufcnm** จากนั้นโลโก้ยูทิลีตี CNM และหน้าต่างหลักถูกแสดง

ภาพรวมยูทิลีตีการกำหนดค่าเริ่มต้นโหนด CCA

ยูทิลิตี CCA Node Initialization รันสคริปต์ที่คุณสร้างขึ้น โดยใช้ *เอดิเตอร์ CNI* ภายในยูทิลิตี CNM สคริปต์เหล่านี้ รู้จักกันในนามของ *รายการ CNI* ยูทิลิตี CNI สามารถรันฟังก์ชันยูทิลิตี CNM ที่จำเป็นในการตั้งค่าโหนด ตัวอย่างเช่น ยูทิลิตีสามารถใช้เพื่อโหลดบทบาทและโปรไฟล์ การควบคุมการเข้าถึง

เนื่องจากคุณสร้างรายการ CNI คุณระบุตำแหน่งดิสก์ของอ็อบเจ็กต์ข้อมูล ที่ยูทิลิตี CNI จะโหลดลงในโหนดเป้าหมาย หลังจากที่สร้างรายการ CNI แล้ว คุณสามารถแจกจ่ายรายการ CNI และการประกอบขึ้นเป็นไฟล์ข้อมูลใดๆ (สำหรับบทบาท โปรไฟล์ และอื่นๆ) ไปยังยูทิลิตี CNI ที่จะใช้สำหรับการติดตั้ง แบบอัตโนมัติ โหนดปลายทางและโหนดทั้งหมดที่รันรายการ CNI แบบกระจาย ต้องใช้ระบบปฏิบัติการเดียวกันและระดับของความเข้ากันได้ของแอ็พพลิเคชัน Java

หมายเหตุ: การเริ่มยูทิลีตี CCA Node Management: เมื่อต้องการเริ่มยูทิลีตี CCA Node Management ให้ป้อน คำสั่ง **csufcnm** จากนั้นโลโก้ยูทิลีตี CNM และหน้าต่างหลักถูกแสดง

ข้อมูลที่เกี่ยวข้อง:

"สถานการณ์จำลอง: การโคลนคีย์หลัก DES หรือ PKA" ในหน้า 24
ขั้นตอนในการโคลนคีย์หลัก data encryption standard (DES) หรือ public key algorithm (PKA) จากตัวประมวลผลร่วมหนึ่งไปอีกตัวหนึ่ง

"การสร้างโหนดอื่นโดยใช้ยูทิลีตี CNI" ในหน้า 41
การสร้างรายการ CNI สำหรับยูทิลีตี CCA Node Initialization (CNI) ทำให้คุณสามารถโหลดคีย์และข้อมูลการควบคุมสิทธิในการเข้าถึงที่เก็บอยู่บนดิสก์ไปยัง โหนดการเข้ารหัสลับโดยไม่รันยูทิลีตี CNM บน โหนดเป้าหมายใดๆ

สถานการณ์จำลอง: การใช้ยูทิลีตี CNM และ CNI

ส่วนนี้อธิบายการใช้ยูทิลีตี CCA Node Management (CNM) และยูทิลีตี CCA Node Initialization (CNI) เพื่อสร้างโหนดและโคลนไปที่ตัวประมวลผลร่วมอื่น

การใช้งานยูทิลีตีถูกแสดงในสถานการณ์จำลอง ซึ่ง ประกอบด้วย:

1. สร้างโหนดการทดสอบที่ต้องถูกใช้เพื่อพัฒนาแอ็พพลิเคชัน หรือสร้างโพรซีเดอร์สำหรับการใช้ยูทิลิตี CNM *ผู้ใช้งานในครั้งแรกควรทำตาม โพรซีเดอร์นี้เพื่อเริ่มต้นการทดสอบด้วยยูทิลิตีและตัวประมวลผลร่วม*
2. สร้างโหนดต่างๆ สำหรับสภาพแวดล้อมที่ใช้งานจริงโดยใช้ส่วนของคีย์ สถานการณ์ จำลองนี้ใช้รายการ CNI เพื่อทำการสร้างโหนดเป้าหมาย ที่ใช้งานจริงแบบอัตโนมัติ
3. โคลนคีย์หลักจากหนึ่งตัวประมวลผลร่วมไปเป็นอีกหนึ่งตัวประมวลผลร่วม นี่คือโพรซีเดอร์ที่น่าสนใจสำหรับการติดตั้งด้วยความปลอดภัยระดับสูง ซึ่งใช้ตัวประมวลผลร่วมจำนวนมาก

วัตถุประสงค์ของสถานการณ์จำลองคือ การแสดงให้เห็นถึงวิธีการใช้โพรซีเดอร์ที่กล่าวถึงในที่นี้ ในจุดที่เหมาะสม สถานการณ์จำลองอ้างถึง ส่วนอื่นของชุดหัวข้อนี้ โดยมีข้อมูลรายละเอียดเพิ่มเติม

หากคุณไม่คุ้นเคยกับระบบการควบคุมสิทธิในกาเรข้าถึง CCA ของตัวประมวลผลร่วม ดูที่ “ภาพรวมการควบคุมการเข้าถึง” ในหน้า 30 และ “สถานะเริ่มต้นของระบบการควบคุมสิทธิในการเข้าถึง” ในหน้า 30 ที่นี่คุณสามารถ พบคำอธิบายของเทอม เช่น *บทบาท บทบาทดีฟอลต์ เริ่มต้น* และ *โปรไฟล์ผู้ใช้* สถานการณ์จะสมมติว่า ระบบการควบคุมสิทธิในการเข้าถึงอยู่ใน สถานะ เริ่มต้น

หมายเหตุ: สถานการณ์จำลองเหล่านี้จะเป็นสถานการณ์จำลองในรูปของการให้คำแนะนำเท่านั้น คุณต้องสนับสนุน การกำหนดโพรซีเดอร์ที่เหมาะสมกับสภาพแวดล้อมที่ระบุเฉพาะของคุณมากที่สุด อ้างถึงภาคผนวกเกี่ยวกับการดำเนินการที่ปลอดภัยใน IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe และ 4764 PCI-X Cryptographic Coprocessors*

สถานการณ์จำลอง: การสร้างโหนดทดสอบ

ในสถานการณ์จำลองนี้ โปรแกรมเมอร์เดี่ยวจะติดตั้งโหนดเพื่ออนุญาตให้เข้าถึง เซอร์วิสการเข้ารหัสลับแบบไม่มีขีดจำกัด

สิ่งสำคัญ: ผลลัพธ์ของโหนดการเข้ารหัสลับต้องไม่นำมาพิจารณา ความปลอดภัย เนื่องจากภายใต้สถานการณ์จำลองนี้จะใช้คำสั่งที่สำคัญ จำนวนมากและอนุญาตให้ใช้โดยไม่มีข้อจำกัด

สิ่งที่จำเป็นต้องมีก่อน: คุณต้องติดตั้ง ระดับที่เหมาะสมของ Java Runtime Environment (JRE) หรือ Java Development Kit (JDK) ไว้แล้ว

เมื่อต้องการสร้างโหนดทดสอบ ดำเนินขั้นตอนต่อไปนี้:

1. ติดตั้งตัวประมวลผลร่วมและ IBM Cryptographic Coprocessor Support Program ตามที่กล่าวไว้ใน การติดตั้ง Support Program
2. เริ่มต้นยูทิลีตี CCA Node Management โดยป้อนคำสั่ง **csufcnm** โลโก้ยูทิลีตี CNM และพาเนลหลักแสดง
3. หากคุณมีตัวประมวลผลร่วมมากกว่าหนึ่งซึ่งได้ติดตั้ง CCA ไว้ ให้ระบุยูทิลีตี CNM ที่มีตัวประมวลผลร่วมที่คุณต้องการใช้ จากเมนู **Crypto Node** ให้เลือก **เลือกอะแด็ปเตอร์** รายการของหมายเลข อะแด็ปเตอร์ (1 - 8) ที่ใช้ได้ถูกแสดง เลือกอะแด็ปเตอร์ (ตัวประมวลผลร่วม) จาก รายการ ถ้าคุณไม่ได้ใช้รายการ **เลือกอะแด็ปเตอร์** เพื่อ เลือกอะแด็ปเตอร์ จะใช้อะแด็ปเตอร์ (ตัวประมวลผลร่วม) ดีฟอลต์
4. ซิงโครไนซ์นาฬิกาภายในตัวประมวลผลร่วม และโฮสต์คอมพิวเตอร์ จากเมนู **Crypto Node** คลิก **Time**จากเมนูย่อย ที่แสดงให้คลิก **เซ็ต** นาฬิกาถูกซิงโครไนซ์
5. ใช้ยูทิลีตี CNM เพื่ออนุญาตให้ใช้คำสั่งทั้งหมดในบทบาท DEFAULT
   a. จากเมนู **การควบคุมการเข้าถึง** คลิก **บทบาท**
   b. ไฮไลต์รายการ **DEFAULT** และเลือก **แก้ไข** หน้าต่าง แสดงคำสั่งที่เปิดใช้งาน และที่ไม่เปิดใช้งาน โดยบทบาท DEFAULT
   c. คลิก **อนุญาตทั้งหมด**
   d. โหลดบทบาทที่แก้ไขกลับเข้าในตัวประมวลผลร่วมโดยการคลิก **โหลด** เลือก **ตกลง**
   e. บันทึกสำเนาของบทบาทโดยการคลิกปุ่ม **บันทึก** และ ตั้งชื่อบทบาท
6. โหลด function-control vector (FCV) ลงในตัวประมวลผลร่วม จากเมนู **Crypto Node** คลิก **การอนุญาต** จากเมนูย่อยผลลัพธ์ คลิก **โหลด** เพื่อระบุและโหลด FCV

   ไฟล์ FCV คือไฟล์ที่ถูกวางบนเซิร์ฟเวอร์ของคุณระหว่างกระบวน การติดตั้ง โดยปกติแล้ว FCV จะมีชื่อไฟล์ เช่น `fcv_td4kECC521.crt` และถูกค้นหาได้โดยใช้ยูทิลีตีการค้นหาไฟล์ที่มีอยู่ ในระบบปฏิบัติการของคุณ

7. ติดตั้งคีย์หลักจากเมนู **คีย์หลัก** คลิก **คีย์หลัก DES / PKA** หรือ **คีย์หลัก AES** และคลิก **ใช่** ตัวประมวลผลร่วมสร้างและตั้งค่าคีย์หลักสุ่ม

   คีย์หลัก ที่ถูกติดตั้งด้วยอ็อพชัน **ติดตั้งอัตโนมัติ** จะมีการ ส่งผ่านไปยังหน่วยความจำหลักของตัวประมวลผลระบบของคุณเป็นส่วนคีย์ สำหรับวัตถุประสงค์ของการใช้งาน ใช้เมธอดด้วยความปลอดภัย ในการสร้างคีย์หลัก เช่น การสร้างแบบสุ่ม หรือการติดตั้งส่วนของคีย์ที่รู้จัก ซึ่งป้อนโดยบุคคลตั้งแต่สองรายขึ้นไป อ็อพชันเหล่านี้ยังเข้าถึงได้จาก เมนูที่กล่าวถึงข้างต้น
8. เตรียมข้อมูลเบื้องต้นไฟล์หน่วยเก็บข้อมูลคีย์ สำหรับข้อมูลเกี่ยวกับการเตรียมข้อมูลเบื้องต้น ไฟล์หน่วยเก็บข้อมูลคีย์ ดูที่ “การสร้างหรือการเตรียมข้อมูลเบื้องต้นหน่วยเก็บคีย์” ในหน้า 39

   หน่วยเก็บคีย์ คือเงื่อนไข CCA ที่กล่าวถึงตำแหน่งที่ส่วนสนับสนุนโปรแกรมสามารถเก็บคีย์ การเข้ารหัสลับ Data Encryption Standard (DES), Rivest–Shamir–Adleman algorithm (RSA) และ Advanced Encryption Standard (AES) ภายใต้ชื่อที่คุณ (หรือแอ็พพลิเคชันของคุณ) นิยามไว้ หากคุณตั้งใจที่จะใช้ หน่วยเก็บคีย์ คุณต้องกำหนดค่าเริ่มต้นให้กับไฟล์หน่วยเก็บคีย์ หรือไฟล์ที่สอดคล้องกับชนิดของคีย์ที่คุณกำลังใช้: DES, RSA (PKA) หรือ AES ตัวอย่างเช่น หากคุณตั้งใจที่จะใช้เฉพาะคีย์ DES คุณต้องเตรียมข้อมูลเบื้องต้นให้กับไฟล์หน่วยเก็บคีย์ DES แต่ไม่ใช่ไฟล์อื่น หากคุณตั้งใจที่จะใช้คีย์ DES และ PKA คุณต้องเตรียมข้อมูลเบื้องต้นให้กับไฟล์หน่วยเก็บคีย์ DES และ PKA แต่ไม่ใช่ไฟล์หน่วยเก็บคีย์ AES หากคุณตั้งใจที่จะใช้ทั้งสามไฟล์นี้ คุณต้องเตรียมข้อมูลเบื้องต้นให้กับทั้งสามไฟล์

ลิงก์ที่เกี่ยวข้อง: “การสร้างบทบาท” ในหน้า 31

“การโหลดคีย์หลักแบบอัตโนมัติ” ในหน้า 37

สถานการณ์จำลอง: การสร้างโหนดในสภาพแวดล้อมที่ใช้งานจริง

ในสถานการณ์จำลองนี้ ความรับผิดชอบต่อการสร้างโหนดที่เข้ารหัสลับ ถูกแบ่งออกเป็นสามกลุ่ม หนึ่งกลุ่มสำหรับผู้ดูแลระบบควบคุมสิทธิในการเข้าถึง และพนักงานผู้จัดการคีย์สองกลุ่ม

ผู้ดูแลระบบติดตั้งโหนดและระบบการควบคุมการเข้าถึง จากนั้น เจ้าหน้าที่ทีจัดการคีย์โหลดคีย์หลัก และ key encrypting keys (KEKs) ที่จำเป็น KEKs สามารถนำมาใช้เป็นคีย์การส่งข้อมูล เพื่อถ่ายทอดคีย์ระหว่างโหนด

สถานการณ์จำลองนี้จะมุ่งเน้นเกี่ยวกับการติดตั้งคีย์หลัก และ data encryption standard (DES) KEKs ระหว่างโหนดระดับสูง จาก *ส่วนคีย์* การนำ CCA มาใช้สนับสนุนตัวเลือกเทคนิคเกี่ยวกับส่วนของคีย์ เช่นการสร้างคีย์หลักแบบสุ่ม และการกระจายคีย์ DES โดยใช้เทคนิคที่อ้างอิงเทคโนโลยีพับลิกคีย์ Rivest–Shamir–Adleman (RSA) *เทคนิคเกี่ยวกับส่วนของคีย์* จะ สมมติว่ามี *พนักงานผู้จัดการคีย์* สองคนที่สามารถให้ความไว้วางใจ เพื่อดำเนินการกับภารกิจต่างๆ และไม่แบ่งใช้ข้อมูลส่วนของคีย์ใดๆ เทคโนโลยีนี้นำนโยบาย *แบ่งความรู้* มาใช้ ระบบ การควบคุมสิทธิในการเข้าถึงถูกตั้งค่าไว้เพื่อบังคับใช้ *การควบคุมแบบคู่* โดยแบ่งแยกภารกิจของเจ้าหน้าที่รายแรกและรายที่สอง

ในสถานการณ์จำลองนี้ ผู้ดูแลระบบการควบคุมการเข้าถึงใช้ยูทิลีตี cryptographic node management (CNM) เพื่อจัดเตรียมรายการ coprocessor node initialization (CNI) สำหรับโหนดปลายทาง รายการ CNI ทำขั้นตอนของการใช้ยูทิลิตี CNM แบบอัตโนมัติที่โหนดปลายทาง ผู้ดูแลระบบจัดเตรียม รายการ CNI สำหรับภารกิจที่ดำเนินการโดยผู้ดูแลระบบการควบคุมการเข้าถึง โหนดปลายทาง และพนักงานผู้จัดการคีย์สองคน ผู้ดูแลระบบ ต้องรู้คำสั่งที่ต้องการสิทธิในโหนดเป้าหมาย ภายใต้เงื่อนไขอื่นๆ ซึ่งประกอบด้วย:

- การดำเนินการปกติแบบจำกัด (เมื่อใช้บทบาทดีฟอลต์)
- เมื่อรันงานของผู้ดูแลระบบการควบคุมการเข้าถึง
- เมื่อรันงานของพนักงานผู้จัดการคีย์แต่ละคน

- ภายใต้สถานการณ์พิเศษอื่นๆ ที่ใช้บทบาทและโปรไฟล์ ที่เพิ่มเติม

หมายเหตุ: ยูทิลีตี CNM และ CNI คือเครื่องที่ใช้เพื่อติดตั้ง และจัดการบริการที่เข้ารหัส CCA ที่จัดเตรียมไว้โดยโหนด

ผู้ดูแลระบบได้รับสิทธิให้ใช้คำสั่งในบทบาทต่างๆ เพื่อตรวจสอบว่า คำสั่งที่ต้องการถูกเปิดใช้งานแล้ว คำสั่งที่สำคัญมาก เช่น การโหลดส่วนของคีย์แรกหรือการโหลดส่วนของคีย์ลำดับถัดไป ถูกเปิดใช้งานในบทบาทสำหรับผู้ใช้ที่มีความรับผิดชอบและสิทธิในการ ใช้คำสั่งเหล่านั้น ซึ่งเป็นสิ่งสำคัญในการแบ่งแยกความรับผิดชอบ ดังนั้น นโยบายต่างๆ เช่น แบ่งแยกความรู้ และการควบคุมแบบคู่ ถูกบังคับให้ใช้โดยระบบการควบคุมสิทธิในการเข้าถึงของตัวประมวลผลร่วม

ข้อมูลที่เกี่ยวข้อง:

"การสร้างและการจัดการข้อมูลการควบคุมการเข้าถึง" ในหน้า 29

สถานการณ์จำลอง: การจัดเตรียมรายการ CNI สำหรับโหนดปลายทาง: ในงานนี้ ผู้ดูแลระบบการควบคุมการเข้าถึงใช้ยูทิลีตี CCA Node Management (CNM) เพื่อจัดเตรียมรายการ CCA Node Initialization (CNI) สำหรับโหนดปลายทาง

เมื่อต้องการตั้งค่าโหนด และสร้างข้อมูลการควบคุมการเข้าถึง ผู้ดูแลระบบ การควบคุมการเข้าถึงสามารถ:

1. บนโหนดที่สร้างขึ้น ให้เริ่มต้นยูทิลีตี CNM
2. สร้างและบันทึกข้อมูลการควบคุมการเข้าถึงไปยังดิสก์สำหรับ โหนดปลายทาง ซึ่งประกอบด้วย:
   - บทบาท Supervisory และโปรไฟล์ผู้ใช้สำหรับผู้ดูแลระบบการควบคุมสิทธิในการเข้าถึง และเจ้าหน้าที่ผู้จัดการคีย์
   - บทบาทดีฟอลต์เพื่อแทนที่บทบาทดีฟอลต์เริ่มต้น
   a. เมื่อต้องการสร้างรายการ CNI เพื่อซิงโครไนซ์นาฬิกาและ ปฏิทินภายในตัวประมวลผลร่วม และโฮสต์คอมพิวเตอร์.
      1) โหลดข้อมูลการควบคุมการเข้าถึง
      2) ล็อกออนเป็นผู้ดูแลระบบการควบคุมการเข้าถึง
      3) โหลดการแทนที่บทบาทดีฟอลต์
      4) โหลด Function Control Vector (FCV)
      5) ล็อกออฟ
   b. สร้างรายการ CNI สำหรับเจ้าหน้าที่ผู้จัดการคีย์:
      1) ล็อกออนในฐานะเจ้าหน้าที่ผู้จัดการคีย์
      2) โหลดคีย์หลักแรกของส่วนคีย์
      3) โหลดข้อมูลคีย์การเข้ารหัสคีย์ส่วนแรก
      4) ล็อกออฟ
   c. สร้างรายการ CNI สำหรับเจ้าหน้าที่ผู้จัดการ คีย์อันดับที่สอง:
      1) ล็อกออนในฐานะเจ้าหน้าที่ผู้จัดการคีย์อันดับที่สอง
      2) โหลดคีย์หลักที่สองของส่วนคีย์
      3) โหลดข้อมูลคีย์การเข้ารหัสคีย์ส่วนที่สอง
      4) ล็อกออฟ
3. ติดตั้งตัวประมวลผลร่วมของ IBM Common Cryptographic Architecture (CCA) Support Program บนโหนดปลายทาง
4. ส่งข้อมูลการควบคุมสิทธิในการเข้าถึงไปยังโหนดเป้าหมาย และ FCV ที่ระบุอยู่ในรายการ CNI

5. ด้วยความเกี่ยวข้องกันของพนักงานผู้จัดการคีย์ บนโหนดเป้าหมายแต่ละโหนด ให้รันรายการ CNI ที่คุณสร้างขึ้นในขั้นตอน 2a ในหน้า 23, 2b ในหน้า 23และ 2c ในหน้า 23

โหนดเป้าหมายจะพร้อมสำหรับการจัดการกับเซอร์วิสการเข้ารหัสลับ

ข้อมูลที่เกี่ยวข้อง:

"การสร้างและการจัดการข้อมูลการควบคุมการเข้าถึง" ในหน้า 29
"การสร้างโหนดอื่นโดยใช้ยูทิลีตี CNI" ในหน้า 41

การสร้างรายการ CNI สำหรับยูทิลิตี CCA Node Initialization (CNI) ทำให้คุณสามารถโหลดคีย์และข้อมูลการควบคุมสิทธิในการเข้าถึงที่เก็บอยู่บนดิสก์ไปยังโหนดการเข้ารหัสลับโดยไม่รันยูทิลิตี CNM บน โหนดเป้าหมายใดๆ

สถานการณ์จำลอง: การจัดเตรียมและการโหลดส่วนคีย์:

ส่วนนี้อธิบายถึงโพรซีเดอร์ในการจัดเตรียม โหลด และการส่งผ่านส่วนของคีย์

เจ้าหน้าที่การจัดการคีย์จัดเตรียมส่วนคีย์สำหรับใช้ ที่โหนดปลายทาง และโหลดส่วนคีย์ที่โหนดปลายทาง

ติดสินใจเลือกวิธีที่จะส่งส่วนคีย์จากจุด ที่ทำการสร้างไปยังจุดของการติดตั้ง ต่อไปนี้คือความเป็นไปได้บางส่วน:

- สร้างส่วนของคีย์ที่ตำแหน่งกลางและถ่ายโอนส่วนหล่านี้ บนดิสเก็ต
- สร้างส่วนของคีย์ที่ตำแหน่งกลางและถ่ายโอนส่วนเหล่านี้ในรูปแบบกระดาษ
- สร้างส่วนของคีย์ที่จุดและเวลาของการติดตั้ง (ในครั้งแรก) หากส่วนของคีย์จำเป็นหลังการติดตั้ง เพื่อรีโหลด หรือ แบ่งใช้กับโหนดอื่นๆ ดังนั้นคุณต้องตัดสินใจถึงวิธีการจัดส่ง ส่วนคีย์

ตรวจทานความสามารถที่เฉพาะของยูทิลีตี CNM โดยการทำงาน กับยูทิลีตี จากนั้น ตรวจสอบวิธีการเฉพาะที่คุณเลือก และทดสอบยูทิลีตี CCA Node Initialization (CNI) ได้จัดเตรียม ร่วมกับผู้ดูแลระบบการควบคุมสิทธิในการเข้าถึง

สถานการณ์จำลอง: การโคลนคีย์หลัก DES หรือ PKA

ขั้นตอนในการโคลนคีย์หลัก data encryption standard (DES) หรือ public key algorithm (PKA) จากตัวประมวลผลร่วมหนึ่งไปอีกตัวหนึ่ง

คำว่า *การโคลน* ถูกใช้มากกว่าการคัดลอกเนื่อง จากคีย์หลักถูกแบ่งออกเป็นการแบ่งใช้สำหรับกรส่งผ่านระหว่าง ตัวประมวลผลร่วม เทคนิคถูกอธิบายไว้ใต้หัวข้อ "การทำความเข้าใจและการจัดการ กับคีย์หลัก" ในคู่มือ IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* ส่วน "การโคลนคีย์หลัก" ในหน้า 52 จัดเตรียม ขั้นตอน ทีละขั้นที่คุณสามารถทำตามได้ ข้อมูลเบื้องหลัง ที่อนุญาตให้คุณปรับโพรซีเดอร์ถูกอธิบายไว้ในส่วนนี้

หมายเหตุ: การโคลน คีย์หลัก AES ไม่ได้รับการสนับสนุน

การโคลนคีย์หลักเกี่ยวข้องกับโหนดสองหรือสามโหนดขึ้นไป:

- โหนดต้นทางคีย์หลัก
- โหนดปลายทางคีย์หลัก
- โหนด share administration (SA) โหนด SA เป็นได้ทั้ง โหนดต้นทางหรือโหนดปลายทาง

ยูทิลิตี CNM สามารถเก็บหน่วยข้อมูลที่หลากหลายซึ่งเกี่ยวข้องกันในกระบวนการนี้ ในฐานข้อมูลที่คุณสามารถถือ (ดิสเก็ต) หรือถ่ายโอน (FTP) ระหว่าง โหนดที่แตกต่างกันได้ ฐานข้อมูลหนึ่งคือ `sa.db` ซึ่ง เป็นดีฟอลต์ และมีข้อมูลเกี่ยวกับคีย์ SA และคีย์ที่ถูกรับรอง โหนดปลายทางที่คีย์ถูกโคลนยังมีฐานข้อมูล ที่รู้จักกันตามค่าดีฟอลต์ว่าคือ `csr.db`

คุณสามารถบรรลุภารกิจเหล่านั้นโดยใช้ยูทิลิตี CNM:

1. เริ่มต้นยูทิลีตี CCA Node Management โดยป้อนคำสั่ง **csufcnm** โลโก้ยูทิลิตี CNM และหน้าต่างหลักจะแสดงขึ้น
2. ตั้งค่าโหนดด้วยวิธีที่ปลอดภัยด้วยบทบาทการควบคุม สิทธิในการเข้าถึง และโปรไฟล์ผู้ใช้และคีย์หลัก

   คุณต้องได้รับหนึ่งบทบาทหรือหนึ่งโปรไฟล์ผู้ใช้ หรือมากกว่านั้น ที่โหนดต้นทางและปลายทางสำหรับผู้ใช้แต่ละคนซึ่งได้รับหรือเก็บการแบ่งใช้ การประมวลผลการแบ่งใช้ถูกดำเนินการโดยคำสั่งที่แยกกันดังนั้น หากคุณต้องการให้บทบาทของคุณสามารถป้องกันบุคคลแต่ละรายที่ เกี่ยวข้องกับการขอรับและการติดตั้งการแบ่งใช้ที่แตกต่างกัน

   พิจารณา การใช้การสร้างคีย์หลักและบทบาทที่บังคับใช้นโยบายการรักษาความปลอดภัย การควบคุมสองแบบ ตัวอย่างอนุญาตให้บุคคลหนึ่งหรือบทบาท รีจิสเตอร์การแฮชและอีกคนหนึ่งหรือบทบาท รีจิสเตอร์พับลิกคีย์ เลือกบุคคลหรือบทบาทอื่น สำหรับการขอรับและการติดตั้งบุคคลที่แบ่งใช้คีย์หลัก

   ดูที่ส่วน คำแนะนำในคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* สำหรับคำอธิบายของ Master_Key_Process และ Master_Key_Distribute verbs
3. ติดตั้ง 1 – 16 byte environment ID (EID) เฉพาะ ของตัวเลือกของคุณในแต่ละโหนด

   จากเมนู **Crypto Node** คลิก Set Environment ID ป้อน ตัวบ่งชี้และคลิก โหลด ใช้เฉพาะอักขระเหล่านี้ใน EID: A – Z, a – z, 0 – 9 และ @, (X'40'), อักขระเว้นวรรค (X'20'), &, (X'26') และ = (X'3D')

   คุณต้องป้อนตัวบ่งชี้ 16-อักขระเต็ม สำหรับตัวบ่งชี้แบบสั้นให้กรอกข้อมูลในรายการด้วยอักขระเว้นวรรค
4. เตรียมข้อมูลเบื้องต้นการแบ่งใช้คีย์หลักค่า m และ n ในโหนดต้นทาง และปลายทาง ค่าเหล่านี้ต้องเป็นค่าเดียวกันในโหนดต้นทางและโหนดปลายทาง ค่า n คือจำนวนสูงสุดของการแบ่งใช้ ขณะที่ m คือจำนวนต่ำสุดของการแบ่งใช้ที่ต้องถูกติดตั้งไว้ เพื่อสร้างคีย์หลักขึ้นใหม่ในโหนดปลายทาง

   จากเมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **กำหนดจำนวนการแบ่งใช้** ป้อน ค่าและคลิก โหลด
5. ที่โหนดอื่นๆ ให้สร้างคีย์เหล่านี้และ รับแต่ละพับลิกคีย์ ที่ถูกรับรองโดยคีย์ SA คุณสามารถใช้ฐานข้อมูล `sa.db` ของยูทิลิตีเพื่อส่งผ่านข้อมูลคีย์และใบรับรอง

   การควบคุมดูแลการแบ่งใช้ (SA)

   คีย์นี้ถูกใช้เพื่อรับรองคีย์และคีย์ที่ตามมา คุณต้องลงทะเบียนการแฮช ของพับลิกคีย์ SA และพับลิกคีย์เองใน SA โหนดต้นทางและ โหนดปลายทาง

   หลังจากคีย์ SA ถูกสร้างยูทิลีตีจะระบุค่าอักขระ 8 ไบต์หรือ 16-hexadecimal ที่เป็นส่วนของการแฮชคีย์ SA*ตรวจสอบให้แน่ใจ เพื่อเก็บสำเนาค่านี้* คุณต้องการค่านี้เพื่อยืนยัน ค่าการแฮชที่ถูกบันทึกในฐานข้อมูลเพื่อรีจิสเตอร์พับลิกคีย์ SA ที่โหนดต้นทาง และปลายทาง

   การลงนามการแบ่งใช้ตัวประมวลผลร่วม (CSS)

   คีย์นี้ถูกใช้เพื่อลงนามการแบ่งใช้ที่ถูกแจกจ่าย จากโหนดต้นทางคีย์ส่วนตัวถูกเก็บอยู่ภายในโหนดต้นทาง

   การรับการแบ่งใช้ตัวประมวลผลร่วม (CSR)

   คีย์นี้ถูกใช้เพื่อรับการแบ่งใช้คีย์ที่เข้ารหัสในโหนด ปลายทาง พับลิกคีย์ CSR ที่รับรอง SA ถูกใช้ที่โหนดต้นทางเพื่อตัด (เข้ารหัส) การแบ่งใช้คีย์การเข้ารหัสลับที่ไม่ซ้ำกันสำหรับการแบ่งใช้แต่ละครั้ง คีย์ส่วนตัวถูกเก็บอยู่ภายในโหนดปลายทาง

สร้างคู่ของคีย์: SA, CSS และ CSR

จากเมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **สร้างคีย์** คลิก **คีย์การควบคุมดูแล การแบ่งใช้, คีย์ CSS** หรือ **คีย์ CSR** คลิก **สร้าง**

คุณต้องระบุเลเบลคีย์สำหรับคีย์ CSS และ CSR ที่ถูกเก็บ ไว้ในโหนดต้นทางและปลายทาง ตัวอย่าง `IBM4765.CLONING.CSS.KEY` และ `IBM4765.CLONING.CSR.KEY` เลเบลที่คุณใช้ต้องไม่ชนกับเลเบล ของคีย์อื่นที่ถูกใช้ในแอ็พพลิเคชันของคุณ

เมื่อสร้างคีย์ CSR ที่แบ่งใช้การรับโหนด คุณต้องขอรับหมายเลขลำดับของตัวประมวลผลร่วม จาก **Crypto Node** คลิก **สถานะ** คุณ ต้องป้อนค่าหมายเลขลำดับเพื่อรับรองคีย์ CSR

6. ลงทะเบียนพับลิกคีย์ SA ในตัวประมวลผลร่วมที่ SA โหนดต้นทาง และโหนดปลายทาง ขั้นตอนนี้เป็นขั้นตอนแบบสองขั้นตอนที่ต้องทำ ภายใต้นโยบายความปลอดภัยในการควบคุมแบบคู่

   หนึ่งขั้นตอนติดตั้ง การแฮชพับลิกคีย์ SA จากเมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **การควบคุมดูแล การแบ่งใช้รีจิสเตอร์** และคลิก การแฮชคีย์ SA คุณต้องป้อน ค่าแฮช ที่ได้รับในระหว่างการสร้างคีย์ SA

   ขั้นตอนอื่นติดตั้ง พับลิกคีย์ SA จริง จากเมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **การควบคุมดูแล การแบ่งใช้รีจิสเตอร์** และคลิก คีย์ SA โดยดีฟอลต์ข้อมูล พับลิกคีย์อยู่ใน ใน `sa.db`

7. ใช้คีย์ CSS และคีย์ CSR กับโหนด SA และใช้คีย์ที่ ได้รับการรับรอง

   จากเมนูดร็อปดาวน์ **Crypto Node** เลือก **คีย์การควบคุมดูแลการแบ่งใช้, รับรองคีย์คีย์ CSS** หรือ **คีย์ CSR**

   สำหรับคีย์ CSR คุณต้องจัดหาหมายเลขลำดับของตัวประมวลผลร่วมเป้าหมาย เป็นการตรวจสอบเชิงโพรซีเดอร์ที่รับรองคีย์ที่เหมาะสม โพรซีเดอร์ของคุณต้องรวมการสื่อสารกับข้อมูลนี้ ในวิธีการที่เชื่อถือได้

8. ที่โหนดต้นทางผู้ที่ได้รับอนุญาตต้องเข้าสู่ระบบกับบทบาท ซึ่งอนุญาตให้บุคคลใดๆ ขอรับการแบ่งใช้ อย่างน้อยต้องได้รับ การแบ่งใช้ การแบ่งใช้เหล่านี้เป็นคีย์หลักปัจจุบัน

   จาก เมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **รับการแบ่งใช้** และป้อนหมายเลขการแบ่งใช้ที่จะได้รับ **ให้สังเกตหมายเลขลำดับและ ตัวบ่งชี้ฐานข้อมูล** เมื่อการแบ่งใช้เหล่านี้อยู่ในข้อตกลง ให้คลิก ขอรับ การแบ่งใช้ ข้อมูลการแบ่งใช้ต้องถูกกำหนดโดยดีฟอลต์ลงในไฟล์ `csr.db` และรับใบรับรองคีย์ CSR โดยดีฟอลต์ จากไฟล์ `sa.db`

   ขอรับข้อมูลการตรวจสอบความถูกต้องของคีย์หลักปัจจุบันสำหรับ ใช้ในภายหลังที่โหนดปลายทาง จากเมนู **คีย์หลัก** คลิก **คีย์หลัก DES/PKA** > **ตรวจสอบ**คลิก **ปัจจุบัน**

9. ที่โหนดปลายทาง บุคคลที่ได้รับสิทธิ์ต้องเข้าสู่ระบบเป็นบทบาทที่อนุญาต ให้แต่ละบุคคลติดตั้งการแบ่งใช้ของตน อย่างน้อย ที่สุดการแบ่งใช้ m ต้องถูกติดตั้งเพื่อสร้างคีย์หลักอีกครั้งในการลงทะเบียน คีย์หลักใหม่

   จากเมนู **Crypto Node** คลิก **การควบคุมดูแลการแบ่งใช้** > **โหลดการแบ่งใช้** และป้อนหมายเลขการแบ่งใช้ที่จะถูกติดตั้ง ตรวจสอบว่า หมายเลขลำดับและตัวบ่งชี้ฐานข้อมูลถูกต้อง จากนั้นคลิก **สังเกต หมายเลขลำดับและตัวบ่งชี้ฐานข้อมูล**เมื่อการแบ่งใช้เหล่านี้ได้รับการยอมรับว่าถูกต้อง ให้คลิก ขอรับ การแบ่งใช้ ที่โหนดปลายทาง บุคคลที่ได้รับสิทธิ์ต้องเข้าสู่ระบบเป็นบทบาทที่อนุญาต ให้บุคคลติดตั้งการแบ่งใชของตน ข้อมูลการแบ่งใช้ถูกขอรับตามค่าดีฟอลต์ จากไฟล์ `csr.db` และใบรับรองคีย์ CSS ขอรับโดยดีฟอลต์จาก ไฟล์ `sa.db` หากเซิร์ฟเวอร์ของคุณมีตัวประมวลผลร่วมการเข้ารหัสจำนวนมาก ที่ถูกโหลดด้วย CCA ตัวประมวลผลร่วมต้องติดตั้งคีย์หลักเฉพาะ สำหรับการทำหน้าที่ของหน่วยเก็บคีย์

   เมื่อโหลดการ แบ่งใช้ให้ตรวจสอบว่าคีย์ในส่วนลงทะเบียนคีย์หลักใหม่เหมือนกับ คีย์หลักปัจจุบันในโหนดต้นทางเมื่อมีการจัดหาการแบ่งใช้ บนโหนดเป้าหมาย จากเมนู **คีย์หลัก** คลิก **คีย์หลัก DES/PKA** > **สร้าง**

10. เมื่อยืนยันผ่านการตรวจสอบคีย์หลักที่คีย์หลักได้ถูกโคลน บุคคลที่ได้รับสิทธิสามารถ ตั้งค่า คีย์หลักได้ การดำเนินการนี้ลบคีย์หลักเก่า และย้ายคีย์หลักปัจจุบัน ไปที่รีจิสเตอร์คีย์หลักเก่า แอ็พพลิเคชันโปรแกรมที่ใช้คีย์ ที่เข้ารหัสลับโดยคีย์หลักอาจได้รับผลกระทบโดยการเปลี่ยนแปลงนี้ ดังนั้น ให้ตรวจสอบว่า การตั้งค่าของคีย์หลักประสานงานกับความต้องการ ของแอ็พพลิเคชันโปรแกรมของคุณ
11. จากเมนู **คีย์หลัก** คลิก **คีย์หลัก DES/PKA** > **เซ็ต**

การใช้ยูทิลีตีฟังก์ชัน CNM

ส่วนนี้อธิบายถึงโพรซีเดอร์ที่ใช้ฟังก์ชันต่างๆ ของยูทิลีตี CNM

การระบุตัวประมวลผลร่วมที่ระบุเฉพาะ

โพรซีเดอร์ในการเลือกตัวประมวลผลร่วมจากหลาย ตัวประมวลผลร่วมที่ใช้ได้บนระบบ

หากระบบของคุณมีตัวประมวลผลร่วมจำนวนมากที่โหลดด้วยโค้ด CCA คุณจำเป็นต้องเลือกตัวประมวลผลที่ระบุเฉพาะเพื่อทำงาน หากคุณไม่ได้เลือกไว้ คุณจะทำงานกับตัวประมวลผลร่วมที่เป็นค่าดีฟอลต์แทน หลังจากคุณเลือกตัวประมวลผลร่วมแล้ว การเลือกนั้นจะยังคงมีผลบังคับใช้ สำหรับเซสชันยูทิลิตีปัจจุบัน หรือจนกว่าคุณจะทำการเลือกอีกครั้ง ภายในเซสชันยูทิลิตี

เมื่อต้องการเลือกตัวประมวลผลร่วม คลิก **เลือกอะแด็ปเตอร์** จากเมนู **Crypto Node** ถ้าคุณไม่เลือกอะแด็ปเตอร์ จะใช้อะแด็ปเตอร์ ดีฟอลต์แทน

หมายเหตุ:

1. เมื่อใช้ยูทิลิตี CLU ตัวประมวลผลร่วมจะถูกอ้างอิงเป็นค่า 0, 1 และ 2 ตัวประมวลผลเฉพาะอาจหรืออาจไม่ได้ติดตั้งแอ็พพลิเคชัน CCA ไว้ ด้วยยูทิลิตี CNM (และแอ็พพลิเคชันอื่นๆ ที่ใช้ CCA API) ตัวประมวลผลร่วมที่โหลดด้วยแอ็พพลิเคชัน CCA จะถูกกำหนดค่าเป็น 1, 2 และ 3 ตัวบ่งชี้ใหม่เหล่านี้จะถูกกำหนดโดย CCA ขณะสแกน ตัวประมวลผลร่วมที่ติดตั้งไว้ทั้งหมดสำหรับตัวประมวลผลร่วมที่ โหลดด้วยแอ็พพลิเคชัน CCA
2. เมื่อโค้ดแอ็พพลิเคชัน CCA คีย์เวิร์ด **CRP01**, **CRP02** และ **CRP03** ถูกใช้เพื่อจัดสรรตัวประมวลผลร่วม ตัวประมวลผลร่วมเหล่านี้ สอดคล้องกับหมายเลข 1, 2 และ 3 ที่ใช้ในเมนูยูทิลีตี CNM

การเตรียมข้อมูลเบื้องต้นให้กับโหนด

ขั้นตอนในการเตรียมข้อมูลเบื้องต้นโหนด CCA ให้กับสภาวะ เริ่มต้น

คุณสามารถเรียกคืนโหนด CCA กลับสู่สถานะเริ่มต้น ซึ่งจัดเตรียม บทบาทที่คุณกำลังทำงานอยู่ภายใต้ (บทบาทดีฟอลต์หรือบทบาทที่ล็อกออนอยู่) การให้สิทธิในการใช้คำสั่ง **กำหนดค่าเริ่มต้นให้กับอุปกรณ์** (ออฟเซ็ต X'0111')

ใช้คำสั่ง **Reinitialize Device** ทำให้การดำเนินการ ต่อไปนี้เกิดขึ้น:

- เคลียร์รีจิสเตอร์คีย์หลัก
- เคลียร์ Public Key Algorithm (PKA) และพับลิกคีย์ PKA ที่รีจิสเตอร์ ที่เก็บไว้
- เคลียร์บทบาทและโปรไฟล์ และการเรียกคืน การควบคุมสิทธิในการเข้าถึงสถานะเริ่มต้น

หากต้องการเตรียมข้อมูลเบื้องต้นให้กับโหนด CCA ให้เลือก **กำหนดค่าเริ่มต้น** จากเมนู Crypto Node คุณจะได้รับคำถามเพื่อให้ยืนยันการดำเนินการของคุณ

ข้อมูลที่เกี่ยวข้อง:

“สถานะเริ่มต้นของระบบการควบคุมสิทธิในการเข้าถึง” ในหน้า 30
สถานะเริ่มต้นมีบทบาทดีฟอลต์เริ่มต้น

การล็อกออนและล็อกออฟโหนด

ผู้ใช้ต้องล็อกออนเข้าสู่ตัวประมวลผลร่วมเพื่อเรียกทำงาน โปรไฟล์ผู้ใช้และบทบาทที่เชื่อมโยง นี้เป็นวิธีเดียวในการใ บทบาทที่ไม่ใช่บทบาทดีฟอลต์

เมื่อต้องการล็อกออน เลือก **Passphrase Logon** จากเมนู **ไฟล์**

เมื่อต้องการล็อกออฟ เลือก **ล็อกออฟ** จากเมนู **ไฟล์**

หมายเหตุ: ด้วยข้อยกเว้นของบทบาท DEFAULT การเข้าถึงตัวประมวลผลร่วม ถูกจำกัดโดยการพิสูจน์ตัวตน passphrase

การโหลด function-control vector

ขั้นตอนในการโหลด FCV ตัวประมวลผลร่วม

function-control vector (FCV) คือค่าที่ลงนามแล้วซึ่งจัดเตรียมไว้โดย IBM เพื่อเปิดใช้งานแอ็พพลิเคชันในตัวประมวลผลร่วมในการจัดเตรียมระดับของเซอร์วิส การเข้ารหัสลับที่สอดคล้องกับกฎข้อบังคับ ในการอิมพอร์ตและเอ็กซ์พอร์ต ภายใต้กฎข้อบังคับปัจจุบัน ผู้ใช้ทุกรายถูกกำหนดสิทธิ ให้มีระดับของการทำงานสำหรับการเข้ารหัสลับระดับเดียวกัน ดังนั้นในตอนนี้ IBM รองรับ FCV เดี่ยวกับ IBM Common Cryptographic Architecture (CCA) Support Program

คุณใช้ยูทิลิตี CNM เพื่อโหลด FCV ลงในตัวประมวลผลร่วม ไฟล์ FCV มีชื่อว่า `fcv_td4kECC521.crt`

หากต้องการโหลด FCV:

1. จากเมนู **Crypto Node** เลือก **Authorization**
2. จากเมนูย่อยที่แสดงคลิก **Load** เพื่อระบุไฟล์ FCV บนดิสก์ ระบุชื่อไฟล์และคลิก **อัพเดต** ยูทิลิตี โหลด FCV
3. คลิก **OK**

การตั้งค่ายูทิลิตี CCA Node Management

โพรซีเดอร์เพื่อตั้งค่า ค่าดีฟอลต์สำหรับยูทิลิตี CNM

พาเนลคอนฟิกูเรชันของยูทิลิตี CNM อนุญาตให้คุณระบุพาธไดเร็กทอรี สำหรับไฟล์ต่างๆ ที่คุณสร้างด้วยยูทิลิตี อย่างไรก็ตาม ยูทิลิตีไม่ใช้พาธที่คุณเก็บอยู่ใน พาเนลคอนฟิกูเรชัน แต่ พาธดีฟอลต์ถูกเก็บ ในตัวแปรสภาวะแวดล้อม Windows แทน คุณอาจค้นหา พาเนลคอนฟิกูเรชันในตำแหน่งที่มีประโยชน์เพื่อบันทึกตำแหน่งที่คุณ ตั้งใจจะเก็บคลาสต่างๆ ของหน่วยข้อมูล

การซิงโครไนซ์นาฬิกาและปฏิทิน

ขั้นตอนในการซิงโครไนซ์นาฬิกาและปฏิทินใน ตัวประมวลผลร่วมและโฮสต์คอมพิวเตอร์

ตัวประมวลผลร่วมใช้นาฬิกาและปฏิทินของตัวเองเพื่อบันทึกเวลา และวันที่aเพื่อป้องกันการโจมตีแบบเล่นซ้ำในการพิสูจน์ตัวตนโปรไฟล์ passphrase หลังจากที่ติดตั้งตัวประมวลผลร่วมแล้ว ให้ซิงโครไนซ์นาฬิกาและปฏิทิน ด้วยระบบโฮสต์นั้น

หากต้องการซิงโครไนซ์นาฬิกาและปฏิทิน:

1. จากเมนู **Crypto Node** คลิกm **Time**
2. จากเมนูย่อยที่แสดงให้คลิก **เซ็ต**

3. พิมพ์ **Yes** เพื่อซิงโครไนซ์นาฬิกาและปฏิทินกับ โฮสต์
4. คลิก **OK**

การรับข้อมูลสถานะของแอ็พพลิเคชัน CCA

คุณสามารถใช้ยูทิลิตีตัวประมวลผลร่วม CNM เพื่อขอรับสถานะ ของแอ็พพลิเคชัน CCA

พาเนลสถานะที่สนับสนุนบนตัวประมวลผลร่วมยูทิลีตี CNM คือ:

แอ็พพลิเคชัน CCA:
แสดงเวอร์ชันและวันที่บิลด์ของแอ็พพลิเคชัน และยังแสดงสถานะของ รีจิสเตอร์คีย์หลัก

อะแด็ปเตอร์:
แสดงหมายเลขลำดับของตัวประมวลผลร่วม ID และระดับของฮาร์ดแวร์

ประวัติคำสั่ง:
แสดงคำสั่งล่าสุดห้าคำสั่งและคำสั่งย่อยที่ ส่งไปยังตัวประมวลผล

วินิจฉัย:
บ่งชี้ว่า เซนเซอร์ที่ชักจูงตัวประมวลผลร่วมใดๆ ได้ถูกทริกเกอร์แล้ว ไม่ว่าจะมีข้อผิดพลาดใดๆ ถูกบันทึกไว้ และมีผลต่อสถานะของแบตเตอรี่ของตัวประมวลผลร่วม

เอ็กซ์พอร์ตการควบคุม:
การควบคุมการเอ็กซ์พอร์ต: แสดงข้อดีของคีย์การเข้ารหัสลับ ที่ใช้โดยโหนด ตามที่ได้นิยามไว้โดย function-control vector (FCV) ที่ฝังไว้ภายในตัวประมวลผลร่วม

หากต้องการดูพาเนลสถานะ:

1. จากเมนู **Crypto Node** คลิก **สถานะ** สถานะของแอ็พพลิเคชัน CCA จะแสดงขึ้น
2. หากต้องการเลือกข้อมูลสถานะอื่นๆ ให้ใช้ปุ่มที่อยู่ด้านล่าง
3. คลิก **ยกเลิก**

ข้อมูลที่เกี่ยวข้อง:

"การจัดการกับคีย์หลัก" ในหน้า 36
คีย์หลักถูกใช้เพื่อเข้ารหัสคีย์การทำงานสำหรับโหนดโลคัล ขณะที่เก็บอยู่ภายนอกตัวประมวลผลร่วม

การสร้างและการจัดการข้อมูลการควบคุมการเข้าถึง

ระบบการควบคุมการเข้าถึงของ IBM CCA Cryptographic Coprocessor Support Program กำหนดสถานการณ์ ภายใต้ตัวประมวลผลร่วมที่สามารถใช้งานได้ ซึ่งจะดำเนินการโดยจำกัดการใช้ คำสั่ง CCA

สำหรับรายชื่อของคำสั่ง CCA เหล่านี้ดูที่ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* และ โปรดดูส่วนของ "คำสั่งที่จำเป็น" ที่ส่วนท้ายของ คำอธิบาย verb แต่ละตัว

ผู้ดูแลระบบสามารถกำหนดผู้ใช้ให้มีสิทธิที่แตกต่างกัน ดังนั้น ผู้ใช้บางรายสามารถใช้เซอร์วิส CCA ที่ไม่พร้อมใช้งานกับผู้ใช้รายอื่น ส่วนนี้ประกอบด้วย ภาพรวมของระบบการควบคุมสิทธิในการเข้าถึงและวิธีการสำหรับการจัดการ กับข้อมูลการควบคุมสิทธิในการเข้าถึง คุณจำเป็นต้องทราบคำสั่งที่จำเป็น และอยู่ภายใต้สถานการณ์ต่างๆ พิจารณาว่าบางคำสั่งควรให้สิทธิเฉพาะบุคคลที่ไว้วางใจได้ หรือโปรแกรมบางโปรแกรมที่ ทำงานภายใต้เวลาที่ระบุไว้โดยทั่วไปแล้ว คุณให้สิทธิเฉพาะคำสั่งที่จำเป็นเหล่านั้น ดังนั้น จึงไม่สามารถเปิดใช้งานความสามารถ ที่ใช้เพื่อลดระดับความปลอดภัยของการติดตั้งของคุณ

คุณจะขอรับข้อมูลเกี่ยวกับคำสั่งที่ใช้จาก เอกสารคู่มือสำหรับแอ็พพลิเคชันที่คุณตั้งใจให้มีการสนับสนุน สำหรับคำแนะนำเพิ่มเติม ดูที่ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors*

ภาพรวมการควบคุมการเข้าถึง

ระบบควบคุมสิทธิในการเข้าถึงจำกัดหรืออนุญาตให้ใช้คำสั่งต่างๆ ตามบทบาทและโปรไฟล์ผู้ใช้

ใช้ยูทิลิตี CNM เพื่อสร้างบทบาท ที่สอดคล้องกับความต้องการและสิทธิพิเศษของผู้ใช้ที่ได้กำหนดไว้

หากต้องการเข้าถึงสิทธิพิเศษที่กำหนดให้กับบทบาท ที่ไม่ได้รับอนุญาตสำหรับบทบาทดีฟอลต์ ผู้ใช้ต้องล็อกออนเข้าสู่ตัวประมวลผลร่วม โดยใช้โปรไฟล์ผู้ใช้อื่นที่ไม่ซ้ำกัน แต่ละโปรไฟล์ผู้ใช้ถูกเชื่อมโยงกับบทบาท และหลายโปรไฟล์สามารถใช้บทบาทเดียวกัน ตัวประมวลผลร่วม จะพิสูจน์ตัวตนการล็อกออนโดยใช้ passphrase ที่ถูกเชื่อมโยงกับโปรไฟล์ ที่ได้ระบุผู้ใช้ไว้

หมายเหตุ: เงื่อนไข *ผู้ใช้* นำมาใช้กับทั้งคนและโปรแกรม

ตัวประมวลผลร่วมยังมีอย่างน้อยหนึ่งบทบาท บทบาทดีฟอลต์ การใช้บทบาทดีฟอลต์ไม่ได้ต้องการโปรไฟล์ผู้ใช้ ผู้ใช้ใดๆ สามารถใช้เซอร์วิสที่อนุญาตให้ใช้โดยบทบาทดีฟอลต์โดยไม่ได้ล็อกออนหรือพิสูจน์ตัวตนโดย ตัวประมวลผลร่วม

ตัวอย่างเช่น ระบบพื้นฐานอาจประกอบด้วยบทบาทต่อไปนี้:

- **ผู้ดูแลระบบการควบคุมสิทธิในการเข้าถึง**: สามารถสร้างโปรไฟล์ผู้ใช้ใหม่ และแก้ไขสิทธิในการเข้าถึงของผู้ใช้ปัจจุบันได้
- **พนักงานผู้จัดการคีย์**: สามารถเปลี่ยนคีย์การเข้ารหัสลับได้ ความรับผิดชอบนี้เป็นการดีที่สุดที่จะแบ่งใช้โดยผู้ใช้มากกว่าสองราย ที่ใช้สิทธิในการป้อนส่วนของคีย์ อันดับแรกหรืออันดับถัดมา
- **ผู้ใช้ทั่วไป**: สามารถใช้เซอร์วิสในการเข้ารหัสลับ เพื่อปกป้องการทำงานของพวกเขา แต่ไม่มีสิทธิพิเศษในการดูแลระบบ หากแผนงานความปลอดภัยของคุณ ไม่ต้องการพิสูจน์ตัวตนการล็อกออนสำหรับผู้ใช้ทั่วไป ให้กำหนดความต้องการในบทบาทดีฟอลต์

หมายเหตุ: ผู้ใช้บางรายจะถูกกำหนดบทบาทเจ้าหน้าที่ผู้จัดการคีย์ หรือผู้ดูแลระบบการควบคุมสิทธิในการเข้าถึง โดยทั่วไป ผู้ที่ได้สิทธิที่มากกว่า จะไม่ล็อกออน และจะมีสิทธิที่ได้รับในบทบาท ดีฟอลต์

สถานะเริ่มต้นของระบบการควบคุมสิทธิในการเข้าถึง

สถานะเริ่มต้นมีบทบาทดีฟอลต์เริ่มต้น

หลังจากที่คุณโหลดส่วนสนับสนุนซอฟต์แวร์ CCA ลงในเซ็กเมนต์ 3 ของตัวประมวลผลร่วมแล้ว หรือหลังจากที่เริ่มต้นระบบ การควบคุมสิทธิในการเข้าถึง ไม่มีข้อมูลการควบคุมสิทธิในการเข้าถึงอยู่ยกเว้นสำหรับบทบาทดีฟอลต์เริ่มต้น ซึ่งอนุญาตให้ผู้ใช้ที่ไม่ได้พิสูจน์ตัวตนสร้างและโหลดข้อมูลการควบคุมสิทธิในการเข้าถึง

หลังจากที่สร้างบทบาทและโปรไฟล์ที่จำเป็นสำหรับสภาพแวดล้อมของคุณแล้ว ซึ่งประกอบด้วยบทบาทของหัวหน้างานที่จำเป็นต่อการโหลดข้อมูลการควบคุมสิทธิในการเข้าถึง และเพื่อจัดการกับคีย์การเข้ารหัสลับ ลบสิทธิทั้งหมดที่กำหนดให้กับบทบาท ดีฟอลต์ จากนั้น เพิ่มเฉพาะสิทธิเหล่านั้นที่คุณต้องการให้สิทธิกับผู้ใช้ ที่ไม่ได้พิสูจน์ตัวตน

สิ่งสำคัญ: โหนดการเข้ารหัสลับและข้อมูลที่โหนดนั้นปกป้อง ไม่ปลอดภัยขณะที่บทบาทดีฟอลต์ถูกให้สิทธิในการโหลดข้อมูลการควบคุมสิทธิในการเข้าถึง

ข้อมูลที่เกี่ยวข้อง:

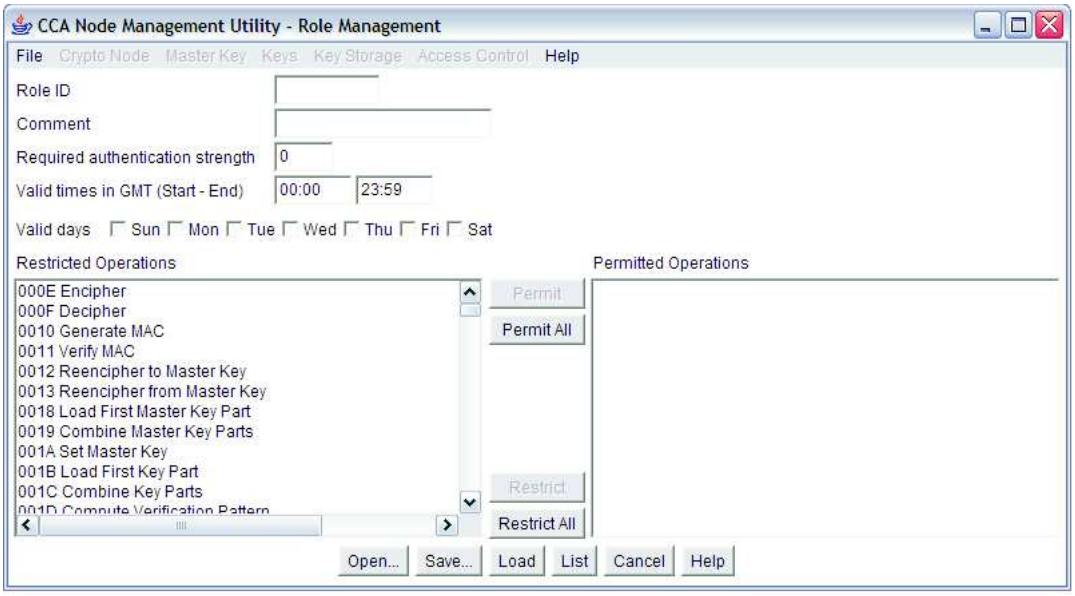
“คำสั่งบทบาทดีฟอลต์เริ่มต้น” ในหน้า 50
คุณลักษณะของบทบาทดีฟอลต์หลังจากตัวประมวลผลร่วมถูก กำหนดค่าเริ่มต้นและเมื่อไม่มีข้อมูลการควบคุมการเข้าถึงอื่นอยู่ ถูกอธิบายไว้ และ คำสั่งการควบคุมการเข้าถึงที่เปิดใช้งานถูกแสดงไว้

การสร้างบทบาท

บทบาทนิยามสิทธิและคุณสมบัติอื่นๆ ของผู้ใช้ ที่กำหนดให้กับบทบาทนั้น

เมื่อต้องการสร้างบทบาท ให้ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **บทบาท** รายการ ของบทบาทที่กำหนดไว้ในขณะนี้ถูกแสดง
2. เลือก **สร้าง** เพื่อแสดงหน้าต่าง การจัดการกับบทบาท ทุกๆ ครั้ง ในกระบวนการ ให้คลิก **รายการ** เพื่อส่งคืนรายการของบทบาท ที่นิยามไว้ในปัจจุบัน



รูปที่ 3. หน้าต่างการจัดการบทบาท

3. กำหนดบทบาทโดยใช้พารามิเตอร์ต่อไปนี้:

ID บทบาท

สตริงอักขระที่นิยามชื่อของบทบาท ชื่อนี้ มีอยู่ในโปรไฟล์ผู้ใช้แต่ละโปรไฟล์ที่ถูกเชื่อมโยงกับบทบาท

ข้อคิดเห็น

สตริงอักขระเผื่อเลือกเพื่อกล่าวถึงบทบาท

ข้อดีของการพิสูจน์ตัวตนที่จำเป็นต้องมี

เมื่อผู้ใช้ล็อกออน ข้อดีของการพิสูจน์ตัวตนที่จัดเตรียมไว้ ถูกเปรียบเทียบระดับของข้อดีที่จำเป็นสำหรับบทบาท หากข้อดีของการพิสูจน์ตัวตน น้อยกว่าที่ต้องการ ผู้ใช้จะไม่สามารถล็อกออนได้ ณ ปัจจุบัน เฉพาะวิธีการพิสูจน์ตัวตนของ passphrase ได้รับการสนับสนุนเท่านั้น ใช้ความแข็งแรง ที่ 50

เวลาที่ถูกต้องและวันที่ถูกต้อง

เมื่อผู้ใช้สามารถล็อกออน โปรดสังเกตว่า เวลาเหล่านี้คือ Coordinated Universal Time หากคุณไม่คุ้นเคยกับ

ระบบ การควบคุมการเข้าถึง ให้ดูที่บทเกี่ยวกับระบบการควบคุมการเข้าถึงของคู่มือ IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe และ 4764 PCI-X Cryptographic Coprocessors*

การดำเนินการที่จำกัดและการดำเนินการที่ได้รับอนุญาต

รายการที่นิยามคำสั่งที่ อนุญาตให้ใช้บทบาท

CCA API verb แต่ละตัวอาจต้องการ คำสั่งหนึ่งคำสั่งหรือมากกว่าเพื่อขอรับเซอร์วิสจากตัวประมวลผล ผู้ใช้ที่ร้องขอเซอร์วิส ต้องถูกกำหนดให้กับบทบาทที่อนุญาตให้ใช้คำสั่งเหล่านี้จำเป็นต้อง รัน verb

สำหรับ ข้อมูลเพิ่มเติมเกี่ยวกับการเรียกและคำสั่ง CCA verb โปรดอ้างถึงคู่มือ IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors*

4. คลิก **บันทึก** เพื่อบันทึกบทบาทกับดิสก์
5. คลิก **โหลด** เพื่อโหลดบทบาทลงในตัวประมวลผล

การแก้ไขบทบาทที่มีอยู่

คุณสามารถใช้ยูทิลีตี CNM เพื่อแก้ไขบทบาท disk stored และ coprocessor stored role และลบบทบาท coprocessor stored

หมายเหตุ: บทบาทที่มีอยู่ใดๆ สามารถใช้เป็นเท็มเพลต เพื่อสร้างบทบาทใหม่ได้ เมื่อคุณเปิดบทบาทที่บันทึกไว้ ข้อมูลที่มีอยู่จะถูกแสดงอยู่ในหน้าต่างนิยามบทบาท คุณจำเป็นต้องแก้ไขหรือป้อนข้อมูล ที่ระบุเฉพาะกับบทบาทใหม่เท่านั้น กำหนด ID บทบาทใหม่และโหลดหรือ บันทึก

การแก้ไขบทบาทการเก็บดิสก์:

ส่วนนี้อธิบายโพรซีเดอร์ที่แก้ไขบทบาทที่มีอยู่ ที่จัดเก็บบนดิสก์

เมื่อต้องการแก้ไขบทบาทที่จัดเก็บบนดิสก์ ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **บทบาท** รายการ ของบทบาทที่กำหนดไว้ในขณะนี้ถูกแสดง
2. คลิก **เปิด** คุณจะได้รับพร้อมต์ให้ป้อนไฟล์
3. เปิดแฟ้ม ข้อมูลถูกแสดงในหน้าต่าง นิยามบทบาท
4. แก้ไขบทบาท
5. คลิก **บันทึก** เพื่อบันทึกบทบาทกับดิสก์
6. ทางเลือก: คลิก **โหลด** เพื่อโหลดบทบาทไปยังตัวประมวลผลร่วม

การแก้ไขบทบาทการเก็บตัวประมวลผลร่วม:

ส่วนนี้อธิบายโพรซีเดอร์ที่แก้ไขบทบาทที่จัดเก็บ ในตัวประมวลผลร่วม CCA

เมื่อต้องการแก้ไขบทบาทที่เก็บในตัวประมวลผลร่วมให้ดำเนินขั้นตอน ต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **บทบาท** รายการ ของบทบาทที่กำหนดไว้ในขณะนี้ถูกแสดง
2. ไฮไลต์บทบาทที่คุณต้องการแก้ไข
3. คลิก **แก้ไข** ข้อมูลในหน้าต่างย่อยนิยามบทบาทจะแสดง
4. แก้ไขบทบาท
5. คลิก **บันทึก** เมื่อต้องการบันทึกบทบาทลงในดิสก์

6. ทางเลือก: คลิก **โหลด** เมื่อต้องการโหลดบทบาทไปยังตัวประมวลผลร่วม

การลบบทบาทการเก็บตัวประมวลผลร่วม:

ส่วนนี้อธิบายถึงโพรซีเดอร์ที่ใช้ลบบทบาท จากตัวประมวลผลร่วม CCA

สิ่งสำคัญ: เมื่อคุณลบบทบาททิ้ง ยูทิลิตี CNM ไม่ได้ลบหรือกำหนดโปรไฟล์ผู้ใช้ที่เชื่อมโยงกับบทบาทนั้น โดยอัตโนมัติ คุณต้องลบหรือกำหนดโปรไฟล์ผู้ใช้ที่ถูก เชื่อมโยงกับบทบาทอีกครั้งก่อนที่คุณจะลบบทบาท

เมื่อต้องการลบบทบาทที่เก็บในตัวประมวลผลร่วมให้ดำเนินขั้นตอน ต่อไปนี้:

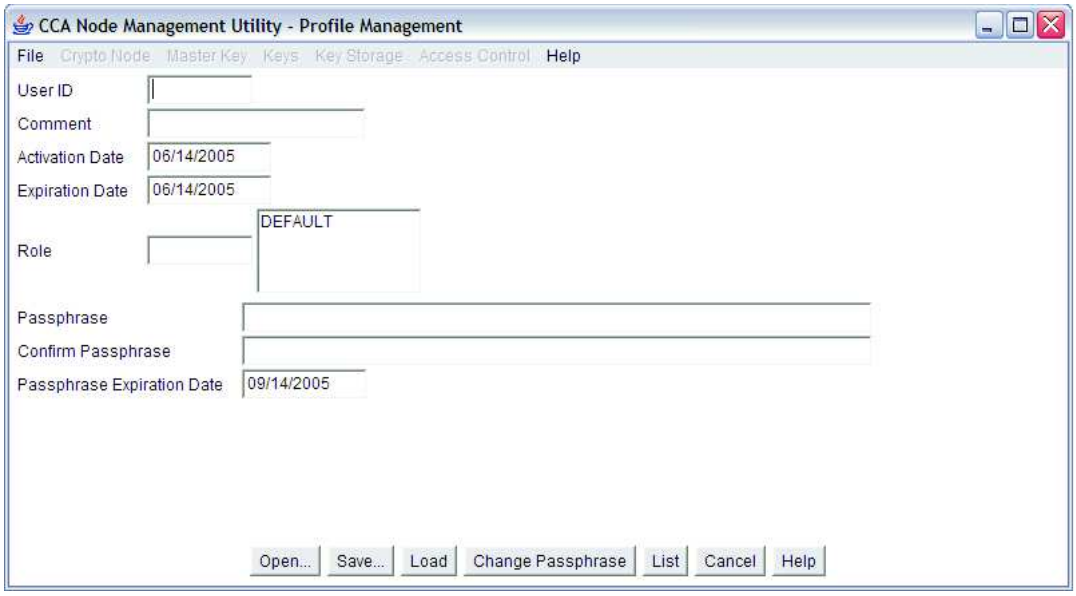
1. จากเมนู **การควบคุมการเข้าถึง** คลิก **บทบาท** รายการ ของบทบาทที่กำหนดไว้ในขณะนี้ถูกแสดง
2. ไฮไลต์บทบาทที่คุณต้องการลบ
3. คลิก **ลบ** บทบาทถูกลบ

การสร้างโปรไฟล์ผู้ใช้

โปรไฟล์ผู้ใช้ระบุผู้ใช้เฉพาะกับตัวประมวลผลร่วม

เมื่อต้องการสร้างโปรไฟล์ผู้ใช้ ให้ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **โปรไฟล์** รายการ ของโปรไฟล์ที่กำหนดไว้ในขณะนี้ถูกแสดง
2. เลือก **สร้าง** เพื่อแสดงหน้าต่างการจัดการโปรไฟล์ดูที่ รูปที่ 4เพื่อดูฟิลด์ของหน้าต่าง การจัดการโปรไฟล์



รูปที่ 4. พาเนล การจัดการกับโปรไฟล์

3. กำหนดโปรไฟล์ผู้ใช้

   ฟิลด์ของโปรไฟล์ผู้ใช้มีดังนี้:

   ID ผู้ใช้ ชื่อที่กำหนดให้กับโปรไฟล์ผู้ใช้ของตัวประมวลผลร่วมการเข้ารหัสลับ

ข้อคิดเห็น

สตริงอักขระเผื่อเลือกเพื่ออธิบายถึงโปรไฟล์ผู้ใช้

วันที่เรียกใช้และวันที่หมดอายุ

วันที่เริ่มแรกและวันที่สุดท้ายที่ผู้ใช้สามารถล็อกออนเข้า โปรไฟล์ผู้ใช้

บทบาท

ชื่อของบทบาทที่นิยามสิทธิที่ให้แก่โปรไฟล์ ผู้ใช้

Passphrase และยืนยัน Passphrase

สตริงอักขระที่ผู้ใช้ต้องป้อนเพื่อขอรับสิทธิในการเข้าถึง โหนดการเข้ารหัสลับ

วันที่หมดอายุของ Passphrase

วันที่หมดอายุสำหรับ passphrase ยูทิลิตีจะตั้งค่านี้ ตามค่าดีฟอลต์คือ 90 วันจากวันที่ปัจจุบัน คุณสามารถเปลี่ยนวันที่หมดอายุได้ passphrase ทุกตัวจะมีวันที่หมดอายุ ซึ่งนิยามช่วงอายุการทำงานของ passphrase นั้น ซึ่งจะแตกต่างจาก วันที่หมดอายุของโปรไฟล์เอง

4. คลิก **บันทึก** เพื่อบันทึกโปรไฟล์ไปที่ดิสก์
5. ทางเลือก: คลิก **โหลด** เพื่อโหลดโปรไฟล์ลงในตัวประมวลผลร่วม

การแก้ไขโปรไฟล์ที่มีอยู่

คุณสามารถใช้ยูทิลีตี CNM เพื่อแก้ไขโปรไฟล์ disk stored และ coprocessor stored และลบโปรไฟล์ coprocessor stored

หมายเหตุ: โปรไฟล์ที่มีอยู่ใดๆ สามารถใช้เป็นเท็มเพลต เพื่อสร้างโปรไฟล์ใหม่ได้ เมื่อคุณเปิดโปรไฟล์ที่บันทึกไว้ ข้อมูลที่มีอยู่ จะถูกแสดงอยู่ในหน้าต่างนิยามโปรไฟล์ คุณจำเป็นต้องแก้ไขหรือป้อนข้อมูล ที่ระบุเฉพาะกับโปรไฟล์ใหม่เท่านั้น กำหนด ID โปรไฟล์ใหม่และโหลดหรือ บันทึก

การแก้ไขโปรไฟล์ผู้ใช้การเก็บดิสก์:

ส่วนนี้อธิบายโพรซีเดอร์ที่แก้ไขโปรไฟล์ผู้ใช้ ที่จัดเก็บบนดิสก์

เมื่อต้องการแก้ไขโปรไฟล์ผู้ใช้ที่จัดเก็บบนดิสก์ ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** เลือก **โปรไฟล์** รายการ ของโปรไฟล์ที่กำหนดไว้ในขณะนี้ถูกแสดง
2. คลิก **เปิด** คุณจะได้รับพร้อมต์ให้ป้อนไฟล์
3. เปิดแฟ้ม ข้อมูลถูกแสดงในหน้าต่าง นิยามโปรไฟล์ ผู้ใช้
4. แก้ไขโปรไฟล์
5. คลิก **บันทึก** เพื่อบันทึกโปรไฟล์ไปที่ดิสก์
6. ทางเลือก: คลิก **โหลด** เพื่อโหลดโปรไฟล์ลงในตัวประมวลผลร่วม

การแก้ไขโปรไฟล์ผู้ใช้ที่จัดเก็บในตัวประมวลผลร่วม:

ส่วนนี้อธิบายโพรซีเดอร์ที่แก้ไขโปรไฟล์ผู้ใช้ในตัวประมวลผลร่วม CCA

เมื่อต้องการแก้ไขโปรไฟล์ผู้ใช้ที่จัดเก็บในตัวประมวลผลร่วม ให้ดำเนิน ขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **โปรไฟล์** รายการ ของโปรไฟล์ที่กำหนดไว้ในขณะนี้ถูกแสดง
2. ไฮไลต์โปรไฟล์ผู้ใช้ที่คุณต้องการแก้ไข

3. คลิก **แก้ไข** ข้อมูลในหน้าต่างนิยามโปรไฟล์จะแสดง
4. แก้ไขโปรไฟล์ผู้ใช้
5. คลิก **บันทึก** เมื่อต้องการบันทึกโปรไฟล์ลงดิสก์
6. ทางเลือก: คลิก **โหลด** เมื่อต้องการโหลดโปรไฟล์ไปยังตัวประมวลผลร่วม

การลบโปรไฟล์ผู้ใช้การเก็บตัวประมวลผลร่วม:

ส่วนนี้อธิบายถึงโพรซีเดอร์ที่ใช้ลบโปรไฟล์ผู้ใช้ ที่ถูกเก็บในตัวประมวลผลร่วม CCA

เมื่อต้องการลบโปรไฟล์ที่เก็บในตัวประมวลผลร่วมให้ดำเนินขั้นตอน ต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **โปรไฟล์** รายการ ของโปรไฟล์ผู้ใช้ที่กำหนดไว้ในขณะนี้ถูกแสดง
2. ไฮไลต์โปรไฟล์ผู้ใช้ที่คุณต้องการลบ
3. คลิก **ลบ** โปรไฟล์ผู้ใช้ถูกลบ

การรีเซ็ตจำนวนความล้มเหลวของโปรไฟล์ผู้ใช้: หากต้องการปกป้องการล็อกออนที่ไม่ได้รับการพิสูจน์ตัวตน ระบบการควบคุมสิทธิในการเข้าถึง จะรักษาจำนวนของความล้มเหลวในการความพยายามล็อกออนสำหรับโปรไฟล์ผู้ใช้แต่ละโปรไฟล์ หากจำนวนของความพยายาม ที่ล้มเหลวสำหรับโปรไฟล์ผู้ใช้มีค่าเกินกว่าค่าที่จำกัดไว้ซึ่งนิยามไว้ในโปรไฟล์ โปรไฟล์จะปิดใช้งาน

เมื่อต้องการรีเซ็ตจำนวนความล้มเหลว ให้ทำตามขั้นตอนต่อไปนี้:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **โปรไฟล์** รายการ ของโปรไฟล์ผู้ใช้ที่กำหนดไว้ในขณะนี้ถูกแสดง
2. ไฮไลต์โปรไฟล์ผู้ใช้
3. คลิก **รีเซ็ต FC** หน้าต่างยืนยันถูกแสดง
4. คลิก **ใช่** เพื่อยืนยัน จำนวนความล้มเหลวการพยายามล็อกออนถูก ตั้งค่าเป็น 0

การเตรียมข้อมูลเบื้องต้นของระบบควบคุมการเข้าถึง

เมื่อคุณเตรียมข้อมูลเบื้องต้นของระบบควบคุมการเข้าถึง ยูทิลีตี CNM จะเคลียร์ข้อมูลการควบคุมการเข้าถึงในตัวประมวลผลร่วม และปรับแต่ง บทบาทดีฟอลต์ที่มีคำสั่งที่จำเป็นเพื่อโหลดข้อมูลการควบคุมการเข้าถึง

สำคัญ: โหนดการเข้ารหัสลับและข้อมูลที่โหนดนั้นปกป้อง ไม่ปลอดภัยขณะที่บทบาทดีฟอลต์ถูกให้สิทธิในการโหลดข้อมูลการควบคุมในการเข้าถึง

การดำเนินการที่เป็นผลสำเร็จจะลบการควบคุมสิทธิในการเข้าถึง และคีย์ ดังนั้น จึงเป็นการดำเนินการที่สำคัญที่สามารถ render โหนดที่ไม่สามารถทำงานได้ของคุณสำหรับสภาพแวดล้อมที่ใช้งานจริง การติดตั้งบางส่วนอาจ จะเลือกเพื่อถอดสิทธิสำหรับฟังก์ชันนี้ออกจากบทบาทของ ตัวประมวลผลร่วม ในเหตุการณ์นี้ หากคุณต้องการเตรียมข้อมูลเบื้องต้นให้กับโหนด CCA cryptographic คุณต้องถอนซอฟต์แวร์ CCA ออกจากตัวประมวลผลร่วมและติดตั้งซอฟต์แวร์ CCA อีกครั้ง

หากต้องการเตรียมข้อมูลเบื้องต้นให้กับระบบการควบคุมสิทธิในการเข้าถึง:

1. จากเมนู **การควบคุมการเข้าถึง** คลิก **เตรียมข้อมูลเบื้องต้น** หน้าต่างยืนยันถูกแสดง
2. เลือก **ใช่** เพื่อยืนยัน ยูทิลีตีเตรียมข้อมูลเบื้องต้นให้กับระบบ การควบคุมการเข้าถึง

หมายเหตุ: หากต้องการเริ่มต้น CCA Node Management Utility ให้ป้อนคำสั่ง **csufcnm** โลโก้ยูทิลิตี CNM และหน้าต่างหลักจะแสดงขึ้น

การจัดการกับคีย์การเข้ารหัสลับ

คุณสามารถใช้ยูทิลิตี cnm เพื่อจัดการกับคีย์หลัก เพื่อจัดการ กับคีย์การเข้ารหัสคีย์หลัก (keks) รีเซ็ต และจัดการข้อมูลมาตรฐาน การเข้ารหัส (DES) อัลกอริทึมพับลิกคีย์ (PKA) และ ที่เก็บคีย์ advanced encryption standard (AES)ชนิดของคีย์ถูกนิยามไว้ดังต่อไปนี้:

คีย์หลัก คือ KEK พิเศษ ที่เก็บไว้ในแบบข้อความปกติ (ไม่ได้เข้ารหัส) และเก็บอยู่ภายใน โมดูลความปลอดภัยของตัวประมวลผลร่วม ซึ่งคีย์หลักที่สนับสนุนมีอยู่ด้วยกันสามชนิดคือ: DES, PKA และ AES ทั้งสามชนิดนี้ถูกใช้เพื่อตัดคีย์อื่น ดังนั้น คีย์เหล่านั้นสามารถเก็บไว้ภายนอก โมดูลความปลอดภัยได้ คีย์หลัก DES และ PKA คือคีย์ขนาด 168 บิต ซึ่งมีรูปแบบมาจากคีย์ DES ขนาด 56 บิตจำนวนสามคีย์ คีย์หลัก AES คือคีย์ขนาด 256 บิต

KEKs หลัก คือคีย์ DES ที่แบ่งใช้โดยโหนดการเข้ารหัสลับ และในบางครั้ง อ้างอิงถึงคีย์การส่งข้อมูล ซึ่งจะถูกใช้เพื่อเปลี่ยนรหัสคีย์อื่น ที่แบ่งใช้โดยโหนด KEKs หลัก เช่น คีย์หลัก ถูกติดตั้งจากส่วนของคีย์ ความรู้ของส่วนของคีย์สามารถแบ่งใช้ในส่วน โดยบุคคลสองคนเพื่อให้มีผลต่อการแบ่งแยกความรู้ นั่นคือ นโยบายความปลอดภัยในการควบคุมแบบคู่

คีย์ DES, คีย์ PKA และคีย์ AES อื่น ถูกเข้ารหัส คีย์ที่ถูกใช้เพื่อเตรียมเซอร์วิสการเข้ารหัส เช่นคีย์ media access control (MAC) คีย์ DATA และคีย์ PKA ไพรเวต

หมายเหตุ: เมื่อแลกเปลี่ยนการล้างข้อมูลส่วนของคีย์ ให้ตรวจสอบว่า แต่ละฝ่าย เข้าใจถึงวิธีการแลกเปลี่ยนข้อมูลที่ต้องการใช้ เนื่องจากการจัดการส่วนของคีย์ จะแตกต่างกันท่ามกลางผู้ผลิตที่แตกต่างกันและผลิตภัณฑ์ เข้ารหัสลับที่แตกต่างกันด้วยเช่นกัน

การจัดการกับคีย์หลัก

คีย์หลักถูกใช้เพื่อเข้ารหัสคีย์การทำงานสำรหับโหนดโลคัล ขณะที่เก็บอยู่ภายนอกตัวประมวลผลร่วม

CCA นิยามการลงทะเบียนคีย์หลัก สามรายการ:

- **การลงทะเบียนคีย์หลักปัจจุบัน** เก็บคีย์หลักปัจจุบันไว้โดยใช้ตัวประมวลผลร่วมเพื่อเข้ารหัสและถอดรหัสคีย์โลคัล
- **การลงทะเบียนคีย์หลักเก่า** จะเก็บคีย์หลักก่อนหน้านี้ และถูกใช้เพื่อถอดรหัสคีย์ที่เปลี่ยนรหัสโดยคีย์หลักนั้น
- **การลงทะเบียนคีย์หลักใหม่** คือ ตำแหน่งกลางที่ถูกใช้เก็บข้อมูลคีย์หลักตามที่ สะสมเป็นรูปแบบของคีย์หลักใหม่

IBM Common Cryptographic Architecture (CCA) Support Program ใช้สามเซ็ตของรีจิสเตอร์คีย์หลัก หนึ่งเซ็ตสำหรับการเข้ารหัสคีย์ DES (สมมาตร) หนึ่งเซ็ตสำหรับการเข้ารหัสคีย์ PKA private (อสมมาตร) และหนึ่งเซ็ตสำหรับการเข้ารหัสคีย์ AES (สมมาตร)

หมายเหตุ:

1. Master_Key_Distribution master-key-administration verb ไม่ได้สนับสนุนคีย์หลัก AES โปรแกรมที่ใช้ CCA Master_Key_Process และ Master_Key_Distribution นั้น master-key-administration verbs สามารถใช้คีย์เวิร์ด ASYM-MK เพื่อนำทางการดำเนินการกับการลงทะเบียนคีย์หลัก PKA แบบไม่สมมาตร คีย์เวิร์ด SYM-MK เพื่อนำทางไปยังการลงทะเบียนคีย์หลัก DES แบบสมมาตร หรือทั้งชุดของการลงทะเบียนคีย์หลัก DES แบบสมมาตรและ PKA แบบไม่สมมาตร ยูทิลิตี CNM ใช้อ็อพชัน **BOTH** หากคุณใช้โปรแกรมอื่นๆ เพื่อโหลดคีย์หลัก และหากโปรแกรมนี้ทำงานบนการลงทะเบียนคีย์หลักแบบ SYM-MK หรือ ASYM-MK อย่างใดอย่างหนึ่ง โดยทั่วไป คุณจะไม่สามารถใช้ยูทิลิตี CNM เพื่อดูแลคีย์เหล่านี้ อีกต่อไป โปรดสังเกตว่า คีย์หลัก AES ทำงานเป็นอิสระจากคีย์หลัก DES และ PKA

2. หากการติดตั้งของคุณมีตัวประมวลผลร่วมจำนวนมากอยู่ให้โหลดด้วย CCA คุณจำเป็นต้องดูแลคีย์หลักอย่างเป็นอิสระในแต่ละตัวประมวลผลร่วม
3. หากการติดตั้งของคุณมีเซิร์ฟเวอร์ที่มีตัวประมวลผลร่วมการเข้ารหัสลับจำนวนมาก ที่โหลดด้วย CCA ตัวประมวลผลเหล่านั้นอาจต้องติดตั้งไว้พร้อมกับ คีย์หลักเฉพาะ

ข้อมูลที่เกี่ยวข้อง:

"การรับข้อมูลสถานะของแอ็พพลิเคชัน CCA" ในหน้า 29

คุณสามารถใช้ยูทิลิตีตัวประมวลผลร่วม CNM เพื่อขอรับสถานะ ของแอ็พพลิเคชัน CCA

การตรวจสอบคีย์หลักที่มีอยู่:

ยูทิลิตี CNM สร้างหมายเลขการตรวจสอบสำหรับคีย์หลักแต่ละคีย์ ที่ถูกเก็บอยู่ในการลงทะเบียนคีย์หลัก หมายเลขนี้ระบุคีย์แต่ไม่มีข้อมูลที่เกี่ยวข้องเปิดเผยเกี่ยวกับค่าของคีย์จริง

เมื่อต้องการดูหมายเลขการตรวจสอบคีย์หลัก ให้ทำตามขั้นตอนเหล่านี้:

1. จากหน้าต่างโหลดคีย์หลักคลิก **คีย์หลัก**
2. จากเมนู **คีย์หลัก** ให้เลือก **คีย์หลัก DES/PKA** หรือ **คีย์หลัก AES** จากนั้น เลือก **ตรวจสอบ** เมนูย่อยจะถูกแสดง
3. จากเมนูย่อยที่แสดง เลือกรีจิสเตอร์คีย์หลัก การ ตรวจสอบความถูกต้องสำหรับคีย์ที่เก็บอยู่ในการลงทะเบียนจะถูกแสดง

การโหลดคีย์หลักแบบอัตโนมัติ:

ยูทิลีตี CNM สามารถตั้งค่าคีย์หลักโดยอัตโนมัติใน ตัวประมวลผลร่วม ค่าคีย์หลักไม่สามารถดูได้จากยูทิลิตี

สิ่งสำคัญ: หากคีย์หลักของค่าที่ไม่รู้จักหายไป คุณไม่สามารถถอดรหัสคีย์ที่แนบมาได้

เมื่อต้องการโหลดคีย์หลักโดยอัตโนมัติ ให้ทำตามขั้นตอนเหล่านี้:

1. จากหน้าต่างโหลดคีย์หลักคลิก **คีย์หลัก**
2. จากเมนู **คีย์หลัก** เลือก **คีย์หลัก DES/PKA** หรือ **คีย์หลัก AES**
3. เลือก **เซ็ตอัตโนมัติ** หรือ **สุ่ม** คุณจะได้รับพร้อมต์ เพื่อตรวจสอบคำสั่ง
4. คลิก **Yes** ตัวประมวลผลร่วมสร้างและตั้งค่า คีย์หลัก

หมายเหตุ:

1. อ็อพชัน **สุ่ม** เหมาะสมกว่าเนื่องจากอ็อพชัน **เซ็ตอัตโนมัติ** ส่งส่วนคีย์เคลียร์ผ่านหน่วยความจำระบบโฮสต์
2. เมื่อคุณตั้งค่าหรือตั้งค่าอัตโนมัติคีย์หลัก คุณต้องเข้ารหัส คีย์ที่ถูกเข้ารหัสภายใต้คีย์แบบเดิม ทั้งหมดอีกครั้ง

ข้อมูลที่เกี่ยวข้อง:

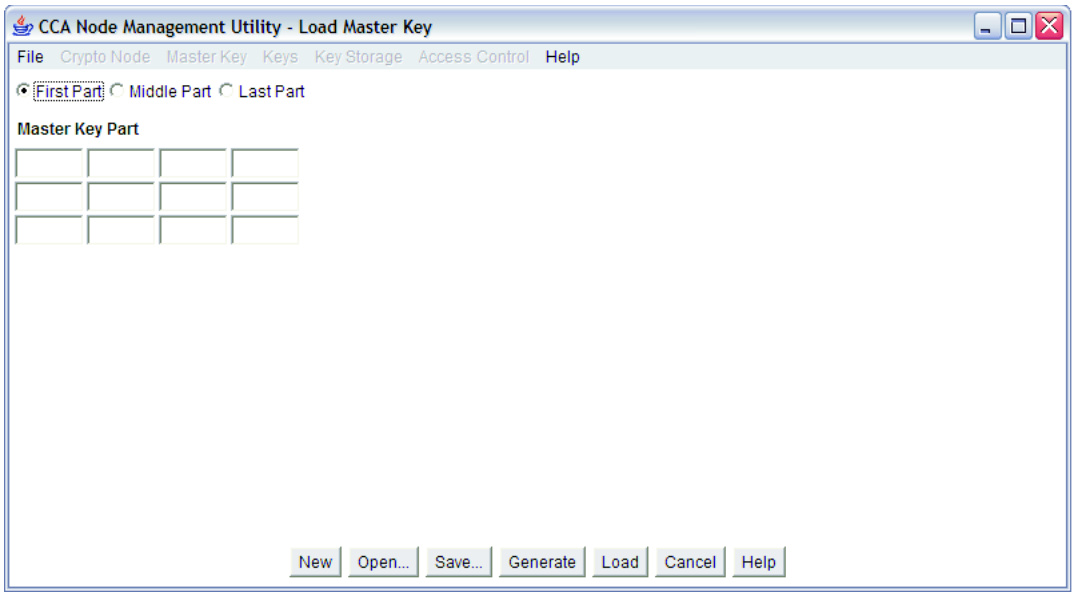
"การเปลี่ยนรหัสคีย์ที่เก็บไว้อีกครั้ง" ในหน้า 39

การโหลดคีย์หลักใหม่จากส่วนของคีย์:

หากต้องการตั้งค่าคีย์หลักใหม่ลงในตัวประมวลผลร่วม ให้ป้อนส่วน ของคีย์ใดๆ ลงในการลงทะเบียนคีย์หลัก และตั้งค่าคีย์หลัก

เมื่อต้องการั้งค่าคีย์หลักใหม่ ทำตามขั้นตอนเหล่านี้:

1. จากเมนู **คีย์หลัก** ให้เลือก **คีย์หลัก DES/PKA** หรือ **คีย์หลัก AES** จากนั้นคลิก **ส่วน** หน้าต่างโหลดคีย์หลักแสดงขึ้น ดังแสดงใน รูปที่ 5



รูปที่ 5. หน้าต่างโหลด คีย์หลัก

2. เลือกปุ่มวิทยุสำหรับส่วนคีย์ที่คุณกำลังแก้ไข (**ส่วน แรก, ส่วนกลาง** หรือ **ส่วน สุดท้าย**)
3. ป้อนข้อมูลด้วยหนึ่งในการดำเนินการต่อไปนี้:
   - คลิก **สร้าง** เพื่อล้างข้อมูลที่ป้อนด้วยความผิดพลาด
   - คลิก **เปิด** เพื่อดึงข้อมูลที่มีอยู่ก่อนหน้านี้
   - คลิก **สร้าง** เพื่อกรอกข้อมูลลงในฟิลด์ที่มีหมายเลขแบบสุ่ม ซึ่งสร้างขึ้นโดยตัวประมวลผล
   - ป้อนข้อมูลลงในฟิลด์ **ส่วนคีย์หลัก** ด้วยตนเอง แต่ละฟิลด์จะรับค่าเลขฐานสิบหก 4 หลัก
4. คลิก **โหลด** เพื่อโหลดส่วนคีย์ลงในส่วนลงทะเบียนคีย์หลัก ใหม่
5. คลิก **บันทึก** เพื่อบันทึกส่วนคีย์ ลงดิสก์

   สำคัญ: ส่วนคีย์ที่บันทึกลงดิสก์ไม่ถูกเข้ารหัส ให้พิจารณาเก็บดิสก์ ด้วยส่วนของคีย์ตามที่เก็บไว้ในที่ที่ปลอดภัยหรือที่เก็บ

   หมายเหตุ: เมื่อ คุณสร้างคีย์จากส่วนต่างๆ คุณต้องมีทั้งส่วนแรกและส่วน สุดท้าย ส่วนกลาง คือส่วนที่สามารถเลือกได้
6. ขั้นตอนขั้นตอนก่อนหน้าเพื่อโหลดส่วนคีย์ที่เลือกไปยัง ส่วนลงทะเบียนคีย์หลักใหม่

   หมายเหตุ: สำหรับการแบ่งแยกความรู้เกี่ยวกับนโยบายความปลอดภัย บุคคลอื่นๆ ต้องป้อนส่วนของคีย์ที่แยกออกจากกัน หากต้องการบังคับใช้ การควบคุมแบบคู่ของนโยบายความปลอดภัย ระบบการควบคุมสิทธิในการเข้าถึงต้อง กำหนดสิทธิ์เพื่อป้อนคีย์แรกลงในหนึ่งบทบาท และสิทธิ์ในการป้อนส่วนของคีย์ถัดมา ลงในบทบาทอื่น จากนั้น ผู้ใช้ที่ได้รับสิทธิ์สามารถล็อกออน เข้าสู่ส่วนของคีย์ตามลำดับ
7. จากเมนู **คีย์หลัก** เลือก **คีย์หลัก DES/PKA** หรือ **คีย์หลัก AES**
8. คลิก **ตั้งค่า** สำหรับยูทิลิตีเพื่อถ่ายโอนข้อมูล:

a. จากส่วนลงทะเบียนคีย์หลักปัจจุบันไปยังส่วนลงทะเบียนคีย์หลักเก่า และเพื่อลบคีย์หลักเก่า
b. จากส่วนลงทะเบียนคีย์หลักใหม่ไปยังส่วนลงทะเบียนคีย์หลักปัจจุบัน

หลังจากการตั้งค่าคีย์หลักใหม่ ให้เปลี่ยนรหัสคีย์อีกครั้ง ซึ่งอยู่ในหน่วยเก็บปัจจุบัน

ลิงก์ที่เกี่ยวข้อง: "การเปลี่ยนรหัสคีย์ที่เก็บไว้อีกครั้ง"

การจัดการกับหน่วยเก็บคีย์

ยูทิลีตี CNM เปิดใช้งานฟังก์ชันการจัดการหน่วยเก็บคีย์พื้นฐาน สำหรับคีย์ ฟังก์ชันยูทิลิตีเหล่านี้ไม่ได้อยู่ในรูปแบบระบบการจัดการคีย์ที่ครอบคลุม

แอ็พพลิเคชันโปรแกรมคือโปรแกรมที่ดีกว่าซึ่งเหมาะสมกับการดำเนินการทำซ้ำ ภารกิจการจัดการกับคีย์

หน่วยเก็บคีย์คือที่เก็บของคีย์ที่คุณเข้าถึงได้โดยเลเบลของคีย์ ซึ่งใช้เลเบลที่คุณหรือแอ็พพลิเคชันของคุณที่นิยาม คีย์ Data Encryption Standard (DES), คีย์ Public Key Algorithm (PKA) Rivest-Shamir-Adleman (RSA) และคีย์ Advanced Encryption Standard (AES) ถูกพักอยู่ใน ระบบหน่วยเก็บที่แยกออกต่างหาก และหน่วยเก็บคีย์มีหน่วยเก็บข้อมูลภายใน ที่จำกัดสำหรับคีย์ PKA คีย์ที่เก็บตัวประมวลผลร่วม ไม่ได้ถูกพิจารณาเป็นส่วนของหน่วยเก็บคีย์ในการอภิปรายนี้

หมายเหตุ:

1. หากเซิร์ฟเวอร์ของคุณมีตัวประมวลผลร่วมที่เข้ารหัสไว้ ซึ่งโหลดด้วย CCA ตัวประมวลผลร่วมเหล่านั้นต้องมีคีย์หลักเฉพาะ ที่ติดตั้งอยู่สำหรับให้หน่วยเก็บทำงานได้อย่างถูกต้อง
2. ยูทิลิตี CNM แสดงจำนวนสูงสุด 1,000 เลเบลของคีย์ หากคุณมีมากกว่า 1,000 เลเบลคีย์ในหน่วยเก็บคีย์ ให้ใช้แอ็พพลิเคชันโปรแกรม เพื่อจัดการกับเลเบลคีย์เหล่านั้น

การสร้างหรือการเตรียมข้อมูลเบื้องต้นหน่วยเก็บคีย์: เมื่อต้องการสร้างหรือเตรียมข้อมูลเบื้องต้นหน่วยเก็บคีย์สำหรับคีย์ Data Encryption Standard (DES), คีย์ Public-Key Algorithm (PKA) หรือ Advanced Encryption Standard (AES) ของคุณ ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **หน่วยเก็บคีย์** เลือก **หน่วยเก็บคีย์ DES, หน่วยเก็บคีย์ PKA** หรือ **หน่วยเก็บคีย์ AES**
2. จากเมนูย่อยผลลัพธ์ คลิก **เตรียมข้อมูลเบื้องต้น** หน้าต่าง เตรียมข้อมูลเบื้องต้น หน่วยเก็บคีย์ DES, เตรียมข้อมูลเบื้องต้นหน่วยเก็บคีย์ PKA หรือเตรียมข้อมูลเบื้องต้น หน่วยเก็บ AES ถูกแสดง
3. ป้อนรายละเอียดสำหรับไฟล์ หน่วยเก็บคีย์
4. คลิก **เตรียมข้อมูลเบื้องต้น** คุณได้รับพร้อมต์เพื่อป้อนชื่อสำหรับ ชุดข้อมูลหน่วยเก็บคีย์
5. ป้อนชื่อสำหรับไฟล์และบันทึกไว้ ไฟล์หน่วยเก็บคีย์ ถูกสร้างขึ้นบนโฮสต์

   หมายเหตุ: หากมีไฟล์ที่มีชื่อเดียวกัน คุณ จะได้รับพร้อมต์เพื่อตรวจสอบตัวเลือกของคุณ เนื่องจาก การเตรียมข้อมูลเบื้องต้นให้กับหน่วยเก็บคีย์จะแก้ไขไฟล์ ดังนั้นหากมีคีย์ใดๆ อยู่ คีย์เหล่านั้นจะถูกลบทิ้ง

การเปลี่ยนรหัสคีย์ที่เก็บไว้อีกครั้ง: หากต้องการเปลี่ยนรหัสคีย์ที่อยู่ในหน่วยเก็บภายใต้คีย์หลักใหม่: ให้ดำเนินขั้นตอนต่อไปนี้ให้สมบูรณ์

1. จากเมนู **หน่วยเก็บคีย์** เลือก **หน่วยเก็บคีย์ DES, หน่วยเก็บคีย์ PKA** หรือ **หน่วยเก็บคีย์ AES**
2. จากเมนูย่อยที่แสดงคลิก **จัดการ** หน้าต่าง การจัดการกับหน่วยเก็บคีย์ DES การจัดการกับหน่วยเก็บคีย์ PKA หรือการจัดการกับหน่วยเก็บคีย์ AES จะแสดงขึ้น พาเนลหน้าต่างนี้แสดงเลเบลของคีย์ใน หน่วยเก็บข้อมูล
3. คลิก **เปลี่ยนรหัส** คีย์จะถูกเปลี่ยนรหัสภายใต้คีย์ ในการลงทะเบียนคีย์หลักปัจจุบัน

การลบคีย์ที่เก็บไว้: เมื่อต้องการลบคีย์ที่เก็บไว้ให้ดำเนินขั้นตอนต่อไปนี้ให้สมบูรณ์:

1. จาก **หน่วยเก็บคีย์** คลิก **หน่วยเก็บคีย์ DES, หน่วยเก็บคีย์ PKA** หรือ **หน่วยเก็บคีย์ AES**
2. จากเมนูย่อยที่แสดง ให้คลิก **จัดการ** หน้าต่างการจัดการหน่วยเก็บคีย์ DES,การจัดการหน่วยเก็บคีย์ PKA หรือการจัดการหน่วยเก็บคีย์ AES ถูกแสดง หน้าต่างนี้แสดงเลเบลของคีย์ในหน่วยเก็บข้อมูล
   คุณ สามารถตั้งค่าเงื่อนไขการกรองเพื่อแสดงเซ็ตย่อยของคีย์ภายในหน่วยเก็บ ตัวอย่าง ถ้าคุณป้อน `*.mac` เป็น เงื่อนไขตัวกรองและรีเฟรชรายการ เซ็ตย่อยถูกจำกัดกับ คีย์ที่มีเลเบลที่ลงท้ายด้วย .mac (เครื่องหมายดอกจัน คืออักขระ wildcard)
3. ไฮไลต์เลเบลของคีย์สำหรับคีย์ที่ต้องการลบ
4. คลิก **ลบ** ข้อความยืนยันถูกแสดง
5. คลิก **Yes** เพื่อยืนยันว่าคีย์ที่เก็บถูกลบ

การสร้างเลเบลของคีย์: เมื่อต้องการสร้างเลเบลคีย์ ให้ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **หน่วยเก็บคีย์** คลิก **หน่วยเก็บคีย์ DES, หน่วยเก็บคีย์ PKA** หรือ **หน่วยเก็บคีย์ AES**
2. จากเมนูย่อยที่แสดง ให้คลิก **จัดการ** หน้าต่างการจัดการหน่วยเก็บคีย์ DES,การจัดการหน่วยเก็บคีย์ PKA หรือการจัดการหน่วยเก็บคีย์ AES ถูกแสดง หน้าต่างนี้แสดงเลเบลของคีย์ในหน่วยเก็บข้อมูล
   คุณ สามารถตั้งค่าเงื่อนไขการกรองเพื่อแสดงเซ็ตย่อยของคีย์ภายในหน่วยเก็บ ตัวอย่าง ถ้าคุณป้อน `*.mac` เป็น เงื่อนไขตัวกรองและรีเฟรชรายการ เซ็ตย่อยถูกจำกัดกับ คีย์ที่มีเลเบลที่ลงท้ายด้วย .mac (เครื่องหมายดอกจัน คืออักขระ wildcard)
3. คลิก **New** คุณจะได้รับพร้อมต์ให้ป้อนเลเบลคีย์
4. คลิก **โหลด** เลเบลคีย์ถูกโหลดลงในหน่วยเก็บข้อมูล

การสร้างและการจัดเก็บ DES KEKs หลัก

Key encrypting keys (KEKs) ถูกเข้ารหัสภายใต้คีย์หลัก Data Encryption Standard (DES) และจัดเก็บในหน่วยเก็บข้อมูลคีย์ DES สำรหับการใช้บนโลคัล

ส่วนของคีย์ที่ใช้เพื่อสร้าง KEK สามารถสร้างหรือป้อนแบบสุ่ม ตามการล้างข้อมูล ส่วนต่างๆ ยังสามารถบันทึกลงในดิสก์หรือดิสเก็ตที่ล้างข้อมูลเพื่อส่งโหนดอื่นๆ หรือเพื่อสร้าง KEK โลคัลอีกครั้ง

หมายเหตุ: ยูทิลิตี Cryptographic Node Management (CNM) สนับสนุนเฉพาะ DES KEKs สำหรับการส่งผ่านคีย์ระหว่างโหนด แอ็พพลิเคชันสามารถใช้ CCA API เพื่อตกแต่งเซอร์วิสที่จำเป็นสำหรับการกระจายคีย์ที่อ้างอิงพับลิก คีย์หรือ Advanced Encryption Standard (AES)

เมื่อต้องการสร้างและจัดเก็บ DES KEK หลัก (หรือคีย์การดำเนินการ ที่มีความยาวเป็นสองเท่า) ดำเนินขั้นตอนต่อไปนี้:

1. จากเมนู **คีย์** คลิก **คีย์การเข้ารหัสคีย์ DES หลัก** หน้าต่างคีย์การเข้ารหัสคีย์ DES หลักถูกแสดง
   ทุกๆ ครั้ง ที่คุณคลิก **สร้าง** เพื่อล้างฟิลด์ข้อมูลทั้งหมด และรีเซ็ตปุ่มวิทยุทั้งหมดกับค่าติดตั้งดีฟอลต์
2. เลือก radio button สำหรับ ส่วนของคีย์ที่ต้องการ ป้อน: **ส่วนแรก ส่วนกลาง** หรือ **ส่วนท้าย**
3. ป้อนข้อมูลในฟิลด์ **ส่วนของคีย์** โดยใช้หนึ่งใน แอ็คชันต่อไปนี้:
   - คลิก **เปิด** เพื่อดึงข้อมูล **ส่วนคีย์, การควบคุม เวกเตอร์** และ **เลเบลคีย์** ที่มีอยู่แล้ว ที่ถูกจัดเก็บลงดิสก์ก่อนหน้านี้โดยใช้คำสั่ง **บันทึก**
   - คลิก **สร้าง** เพื่อกรอกข้อมูลลงในฟิลด์ **ส่วนของคีย์** ด้วยหมายเลขแบบสุ่มที่สร้างโดยตัวประมวลผล

- ป้อนข้อมูลลงในฟิลด์ **ส่วนคีย์** ด้วยตนเอง แต่ละฟิลด์ **ส่วนคีย์** จะรับค่าเลขฐานสิบหก 4 หลัก

4. เลือกการควบคุมเวกเตอร์สำหรับคีย์:
   - หากต้องการใช้เวกเตอร์การควบคุม KEK แบบดีฟอลต์ ให้เลือก radio button **ตัวอิมพอร์ตดีฟอลต์** หรือ **ตัวเอ็กซ์พอร์ตดีฟอลต์** ที่เหมาะสม
   - หากต้องการใช้การควบคุมเวกเตอร์แบบกำหนดเอง ให้เลือก radio button**กำหนดเอง** ในฟิลด์ **เวกเตอร์การควบคุม** ป้อนครึ่งซ้ายหรือขวา ของเวกเตอร์การควบคุมสำหรับคีย์ที่มีความยาวเป็นสองเท่า โปรดสังเกตว่า บิตส่วนของคีย์ (บิต 44) ต้องเปิดอยู่ และแต่ละไบต์ของ เวกเตอร์การควบคุมต้องมีพาริตีคู่

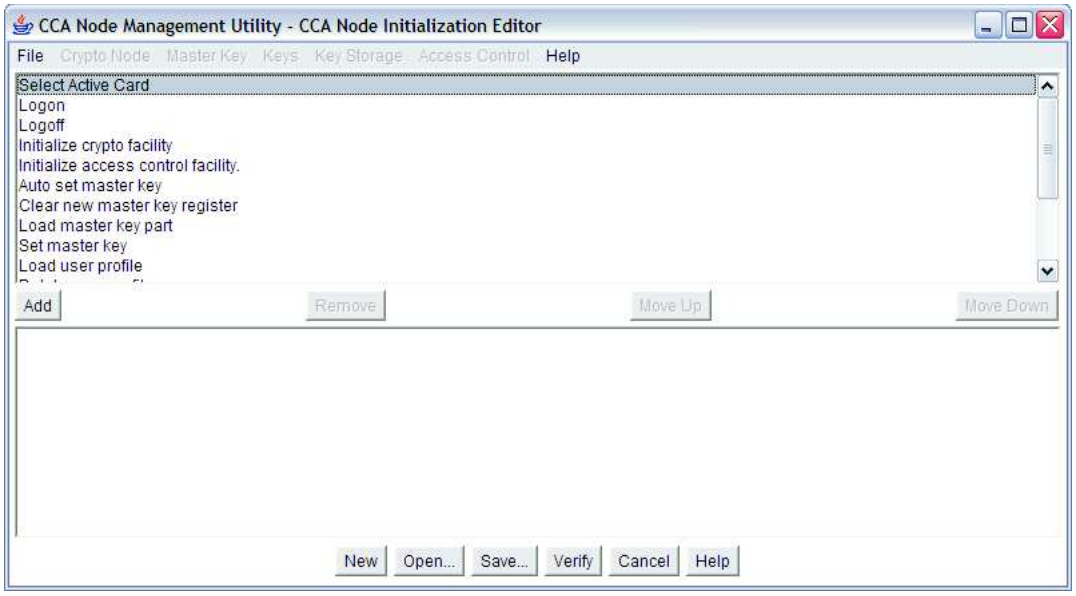
     สำหรับข้อมูลโดยละเอียด เกี่ยวกับ control vectors ดูที่คู่มือ IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors*
5. ป้อนเลเบลของคีย์เพื่อระบุโทเค็นคีย์ในหน่วยเก็บคีย์
6. คลิก **โหลด** เพื่อโหลดส่วนคีย์ไปยังตัวประมวลผลร่วมและ จัดเก็บโทเค็นคีย์ผลลัพธ์ลงในหน่วยเก็บข้อมูลคีย์
7. คลิก **บันทึก** เพื่อบันทึก ส่วนคีย์ที่ไม่ได้เข้ารหัส และเวกเตอร์การควบคุมที่สัมพันธ์กัน รวมถึงค่ เลเบลคีย์ไปยังดิสก์
8. **บันทึก** ลงในดิสก์ หรือ **โหลด** ลงในหน่วยเก็บคีย์ ข้อมูลส่วน คีย์ที่เหลือโดยขั้นตอนต่อไปนี้ 2 ในหน้า 40- 7ตรวจสอบให้แน่ใจว่า คุณได้ใช้เลเบลของคีย์ เดียวกันสำหรับคีย์เดี่ยวแต่ละส่วน

การสร้างโหนดอื่นโดยใช้ยูทิลีตี CNI

การสร้างรายการ CNI สำหรับยูทิลิตี CCA Node Initialization (CNI) ทำให้คุณสามารถโหลดคีย์และข้อมูลการควบคุมสิทธิในการเข้าถึงที่เก็บอยู่บนดิสก์ไปยัง โหนดการเข้ารหัสลับโดยไม่รันยูทิลิตี CNM บน โหนดเป้าหมายใดๆ

เมื่อต้องการเซ็ตอัพโหนดโดยใช้ยูทิลีตี CNI ให้ดำเนินขั้นตอนต่อไปนี้ให้สมบูรณ์:

1. เริ่มต้นยูทิลีตี CCA Node Management โดยป้อนคำสั่ง **csufcnm** โลโก้ยูทิลีตี CNM และพาเนลหลักแสดง
2. บันทึกไปที่โฮสต์หรือสื่อบันทึกที่เคลื่อนย้ายได้ เช่นดิสเก็ต ข้อมูล การควบคุมการเข้าถึงและคีย์ที่คุณต้องการติดตั้งบนโหนดอื่น เมื่อคุณรันยูทิลิตี CNI บนโหนดเป้าหมาย ยูทิลิตีจะค้นหาพาธไดเร็กทอรีเฉพาะ สำหรับแต่ละไฟล์ ตัวอย่าง เช่น:
   - หากคุณบันทึกโปรไฟล์ผู้ใช้ไปยังโหนดไดเร็กทอรี `c:\IBM4764\profiles` ยูทิลิตี CNI ค้นหาโหนดไดเร็กทอรีเป้าหมาย `c:\IBM4764\profiles`
   - หากคุณบันทึกโปรไฟล์ผู้ใช้ลงในดิสเก็ตไดเร็กทอรี `a:\profiles` ยูทิลิตี CNI จะค้นหาโหนดไดเร็กทอรีเป้าหมาย `a:\profiles`
3. จากเมนู **ไฟล์** คลิก **CNI Editor** หน้าต่าง CCA Node Initialization Editor แสดงตามที่ปรากฏใน รูปที่ 6 ในหน้า 42



รูปที่ 6. หน้าต่าง CCA Node Initialization Editor

รายการในหน้าต่างย่อยของหน้าต่างด้านบนแสดงฟังก์ชัน ที่สามารถถูกเพิ่มให้กับรายการ CNI หน้าต่างย่อยล่างแสดงฟังก์ชัน ที่รวมไว้ในรายการ CNI ปัจจุบัน การอ้างอิงถึงคีย์หลักใน รายการอ้างอิงกับคีย์หลัก DES และ PKA

4. เพิ่มฟังก์ชันที่คุณต้องการ หากต้องการเพิ่มฟังก์ชันให้กับรายการ CNI:
   a. ไฮไลต์ฟังก์ชัน
   b. คลิก **Add** ฟังก์ชันถูกเพิ่มให้กับรายการ CNI

   หมายเหตุ: หากฟังก์ชัน ที่คุณเลือกโหลดอ็อบเจ็กต์ข้อมูล เช่น ส่วนของคีย์ ไฟล์หน่วยเก็บคีย์ โปรไฟล์ผู้ใช้ หรือบทบาท คุณจะได้รับพร้อมต์เพื่อป้อนชื่อไฟล์ หรือ ID ของอ็อบเจ็กต์ที่ต้องถูกโหลด
5. การใช้ปุ่ม **เลื่อนขึ้น** และ **เลื่อนลง** จัดการกับฟังก์ชันเพื่อให้มีผลต่อลำดับเดียวกันของคุณที่ทำตาม เมื่อใช้ยูทิลิตี CNM ตัวอย่าง ถ้าคุณกำลังโหลดข้อมูลการควบคุมการเข้าถึง
6. คลิก **ตรวจสอบ** เพื่อยืนยันว่า อ็อบเจ็กต์ได้ถูกสร้างขึ้น อย่างถูกต้อง
7. คลิก **บันทึก** คุณจะได้รับพร้อมต์ให้เลือกชื่อและตำแหน่งไดเร็กทอรีไดเร็กทอรี สำหรับไฟล์รายการ CNI
8. บันทึกไฟล์รายการ CNI ไฟล์รายการไม่มีอ็อบเจ็กต์ข้อมูล ที่ระบุในรายการ CNI
9. คัดลอกไฟล์ที่จำเป็นต่อยูทิลิตี CNI ไปยังตำแหน่งโฮสต์ไดเร็กทอรีเป้าหมาย ที่มิเรอร์ตำแหน่งบนโฮสต์ปลายทาง หากคุณได้บันทึกไฟล์ ลงในสื่อบันทึกที่ถอดออกได้ ให้แทรกสื่อบันทึกลงในโหนดเป้าหมาย
10. จากโหนดเป้าหมาย ให้รันรายการที่ใช้ยูทิลิตี CNI โดยป้อนคำสั่ง **csufcni**

    หากรายการ CNI รวมการล็อกออนไว้ให้ป้อน **csulcni** หรือ **csuncni** บนบรรทัดรับคำสั่ง (โดยไม่ระบุชื่อไฟล์) ข้อมูลวิธีใช้ยูทิลีตี CNI อธิบายถึงไวยากรณ์สำหรับการป้อน ID และ passphrase

    ยูทิลิตี CNI โหลดไฟล์ไปยังตัวประมวลผลร่วมจากโฮสต์หรือสื่อบันทึกแบบถอดออกได้ ตามที่ระบุไว้โดยรายการ CNI

การ Build แอ็พพลิเคชันเพื่อใช้กับ CCA API

แอ็พพลิเคชันสามารถถูกสร้างขึ้นได้ ซึ่งสามารถถูกใช้กับ Common Cryptographic Architecture (CCA) API

ซอร์สโค้ดสำหรับตัวอย่างรูทีนที่ถูกรวมมาพร้อมกับซอฟต์แวร์นี้ คุณสามารถใช้ตัวอย่างที่รวมไว้ให้เพื่อทดสอบตัวประมวลผลร่วมและส่วนสนับสนุนโปรแกรม

หมายเหตุ: ตำแหน่งไฟล์ที่อ้างถึงในส่วนนี้เป็นพาธไดเร็กทอรี ดีฟอลต์

ภาพรวม CCA verbs

แอ็พพลิเคชนโปรแกรมและยูทิลิตีใช้คำร้องขอเซอร์วิสกับตัวประมวลผลร่วมการเข้ารหัส โดยเรียก CCA verbs คำว่า *verb* หมายความถึงการดำเนินการที่แอ็พพลิเคชันโปรแกรมสามารถเริ่มต้นได้ โค้ดของระบบปฏิบัติการจะเปลี่ยนมาเรียกตัวประมวลผลร่วมไดร์เวอร์อุปกรณ์ แบบฟิสิคัล (PDD) ฮาร์ดแวร์และซอฟต์แวร์ที่เข้าถึงผ่าน API คือระบบย่อยรวม

การเรียก Verb ถูกเขียนลงในไวยากรณ์มาตรฐานของภาษาโปรแกรม C และสอดแทรก entry-point พารามิเตอร์, verb และตัวแปร สำหรับพารามิเตอร์เหล่านั้น

สำหรับการแสดงรายละเอียดของ verbs ตัวแปร และพารามิเตอร์ คุณสามารถใช้เมื่อโปรแกรมมิง สำหรับ application programming interface (API) ของการรักษาความปลอดภัย CCA ดูที่คู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors*

การเรียก CCA verbs ในไวยากรณ์โปรแกรมภาษา C

ในสภาพแวดล้อมของระบบปฏิบัติการ คุณสามารถโค้ดการเรียก CCA API verb โดยใช้ไวยากรณ์ภาษาโปรแกรม C มาตรฐาน

ต้นแบบการเรียกฟังก์ชันสำหรับคำกริยา CCA security API ทั้งหมดอยู่ใน ไฟล์ส่วนหัว ไฟล์และตำแหน่งการกระจาย ที่เป็นค่าดีฟอลต์คือ:

AIX `/usr/include/`

หากต้องการสอดแทรกการประกาศ verb เหล่านี้ ให้ใช้คำสั่งคอมไพล์ต่อไปนี้ในโปรแกรมของคุณ:

AIX
```
#include "csufincl.h"
```

เมื่อต้องการเรียกไปที่คำกริยา CCA security API ให้โค้ดชื่อ verb entry-point เป็นอักขระตัวพิมพ์ใหญ่ ให้คั่น identifier พารามิเตอร์ด้วยคอมมา และครอบ identifier เหล่านั้น ให้อยู่ในเครื่องหมายวงเล็บ จบการเรียกด้วยอักขระเครื่องหมายเซมิโคลอน ตัวอย่างเช่น:

```
CSNBCKI (&return_code,
         &reason_code,
         &exit_data_length,    /* exit_data_length */
         exit_data,            /* exit_data        */
         clear_key,
         key_token);
```

หมายเหตุ: พารามิเตอร์ตัวที่สามและสี่ของการเรียก CCA นั่นคือ *exit_data_length* และ *exit_data* ขณะนี้ไม่ได้รับการสนับสนุนโดย CCA Cryptographic Coprocessor Support Program แม้ว่าจะสามารถอนุญาตให้โค้ดตัวชี้แอดเดรสที่มีค่า null สำหรับพารามิเตอร์เหล่านั้นได้ก็ตามแต่ก็มีข้อแนะนำว่า คุณควร ระบุค่าเลขจำนวนเต็มแบบ long ให้มีค่า 0 ด้วยพารามิเตอร์ *exit_data_length*

การคอมไพล์และการลิงก์โปรแกรมแอ็พพลิเคชัน CCA

CCA Cryptographic Coprocessor Support Program มีซอร์สโค้ดภาษา C และ makefile สำหรับโปรแกรมตัวอย่าง

ไฟล์และตำแหน่งการกระจายแบบดีฟอลต์ต่อไปนี้:

AIX `/usr/lpp/csufx.4765/samples/c`

คอมไพล์แอ็พพลิเคชันที่ใช้ CCA และลิงก์โปรแกรมที่คอมไพล์แล้ว กับไลบรารี CCA ไลบรารีและตำแหน่งการกระจายดีฟอลต์ ต่อไปนี้:

AIX `/usr/lib/libcsufcca.a.`

รูทีน C ตัวอย่าง: การสร้าง MAC

หากต้องการแสดงภาพของการฝึกใช้แอ็พพลิเคชัน ของการเรียก CCA verb หัวข้อนี้อธิบายตัวอย่างรูทีนภาษาโปรแกรม C ตัวอย่าง ที่มาพร้อมกับ CCA Cryptographic Coprocessor Support Program

มีตัวอย่างโปรแกรมบนเว็บไซต์ผลิตภัณฑ์ ตัวอย่างโปรแกรมนั้น สามารถช่วยคุณทำความเข้าใจผลการทำงานของการนำไปใช้งานของ CCA

ตัวอย่างรูทีนสร้างโค้ดการพิสูจน์ตัวตนของข้อความ (MAC) บนสตริงข้อความ จากนั้นตรวจสอบ MAC เมื่อต้องการสร้างและตรวจสอบ MAC รูทีน:

1. เรียกคำกริยา **Key_Generate** (CSNBKGN) เพื่อสร้างคู่คีย์ MAC และ MACVER
2. เรียก **MAC_Generate** (CSNBMGN) verb เพื่อสร้าง MAC บนสตริงข้อความด้วยคีย์ MAC
3. เรียก **MAC_Verify** (CSNBMVR) verb เพื่อตรวจสอบสตริงข้อความ MAC ด้วยคีย์ MACVER

ตัวอย่างรูทีนถูกแสดงอยู่ใน รูปที่ 7 ในหน้า 45 ดูที่คู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* สำหรับ รายละเอียดของคำกริยาและพารามิเตอร์ คำกริยาเหล่านี้ถูกแสดงใน ตารางต่อไปนี้

ตารางที่ 5. Verbs ถูกเรียกโดย ตัวอย่างรูทีน

Verb	**ชื่อ Entry-point**
Key_Generate	CSNBKGN
MAC_Generate	CSNBMGN
MAC_Verify	CSNBMVR

รูปที่ 7. รูทีน C ตัวอย่าง: การสร้าง MAC

```
/*****************************************************************/
/*                                                               */
/* Module Name: mac.c                                            */
/*                                                               */
/* DESCRIPTIVE NAME: Cryptographic Coprocessor Support Program   */
/*                   C language source code example              */
/*                                                               */
/*---------------------------------------------------------------*/
/*                                                               */
/* Licensed Materials - Property of IBM                          */
/*                                                               */
/* (C) Copyright IBM Corp. 1997-2010 All Rights Reserved         */
/*                                                               */
/* US Government Users Restricted Rights - Use duplication or    */
/* disclosure restricted by GSA ADP Schedule Contract with IBM Corp. */
/*                                                               */
/*---------------------------------------------------------------*/
/*                                                               */
/*        NOTICE TO USERS OF THE SOURCE CODE EXAMPLES            */
/*                                                               */
/* The source code examples provided by IBM are only intended to */
/* assist in the development of a working software program. The   */
/* source code examples do not function as written: additional   */
/* code is required. In addition, the source code examples may   */
/* not compile and/or bind successfully as written.              */
/*                                                               */
/* International Business Machines Corporation provides the source */
/* code examples, both individually and as one or more groups,   */
/* "as is" without warranty of any kind, either expressed or     */
/* implied, including, but not limited to the implied warranties of */
/* merchantability and fitness for a particular purpose. The entire */
/* risk as to the quality and performance of the source code     */
/* examples, both individually and as one or more groups, is with */
/* you. Should any part of the source code examples prove defective, */
/* you (and not IBM or an authorized dealer) assume the entire cost */
/* of all necessary servicing, repair or correction.             */
/*                                                               */
/* IBM does not warrant that the contents of the source code     */
/* examples, whether individually or as one or more groups, will */
/* meet your requirements or that the source code examples are   */
/* error-free.                                                   */
/*                                                               */
/* IBM may make improvements and/or changes in the source code   */
/* examples at any time.                                         */
/*                                                               */
/* Changes may be made periodically to the information in the     */
/* source code examples; these changes may be reported, for the  */
/* sample code included herein, in new editions of the examples. */
/*                                                               */
/* References in the source code examples to IBM products, programs, */
/* or services do not imply that IBM intends to make these       */
/* available in all countries in which IBM operates. Any reference */
```

```
/* to the IBM licensed program in the source code examples is not    */
/* intended to state or imply that IBM's licensed program must be    */
/* used. Any functionally equivalent program may be used.            */
/*                                                                   */
/*-------------------------------------------------------------------*/
/*                                                                   */
/* This example program:                                             */
/*                                                                   */
/* 1) Calls the Key_Generate verb (CSNBKGN) to create a MAC (message */
/*    authentication code) key token and a MACVER key token.         */
/*                                                                   */
/* 2) Calls the MAC_Generate verb (CSNBMGN) using the MAC key token  */
/*    from step 1 to generate a MAC on the supplied text string      */
/*    (INPUT_TEXT).                                                  */
/*                                                                   */
/* 3) Calls the MAC_Verify verb (CSNBMVR) to verify the MAC for the  */
/*    same text string, using the MACVER key token created in        */
/*    step 1.                                                        */
/*                                                                   */
/*********************************************************************/
#include <stdio.h>
#include <string.h>

#ifdef _AIX
  #include <csufincl.h>
#elif __WINDOWS__
  #include "csunincl.h"
#else
  #include "csulincl.h"     /* else linux */
#endif

/* Defines */
#define KEY_FORM            "OPOP"
#define KEY_LENGTH          "SINGLE  "
#define KEY_TYPE_1          "MAC     "
#define KEY_TYPE_2          "MACVER  "
#define INPUT_TEXT          "abcdefghijklmn0987654321"
#define MAC_PROCESSING_RULE "X9.9-1  "
#define SEGMENT_FLAG        "ONLY    "
#define MAC_LENGTH          "HEX-9   "
#define MAC_BUFFER_LENGTH   10

void main()
{
  static long          return_code;
  static long          reason_code;
  static unsigned char key_form[4];
  static unsigned char key_length[8];
  static unsigned char mac_key_type[8];
  static unsigned char macver_key_type[8];
  static unsigned char kek_key_id_1[64];
  static unsigned char kek_key_id_2[64];
  static unsigned char mac_key_id[64];
  static unsigned char macver_key_id[64];
  static long          text_length;
```

```
static unsigned char text[26];
static long          rule_array_count;
static unsigned char rule_array[3][8];       /* Max 3 rule array elements  */
static unsigned char chaining_vector[18];
static unsigned char mac_value[MAC_BUFFER_LENGTH];

/* Print a banner */
printf("Cryptographic Coprocessor Support Program example program.\n");

/* Set up initial values for Key_Generate call */
return_code = 0;
reason_code = 0;
memcpy (key_form,        KEY_FORM,   4);     /* OPOP key pair              */
memcpy (key_length,      KEY_LENGTH, 8);     /* Single-length keys         */
memcpy (mac_key_type,    KEY_TYPE_1, 8);     /* 1st token, MAC key type    */
memcpy (macver_key_type, KEY_TYPE_2, 8);     /* 2nd token, MACVER key type */
memset (kek_key_id_1,  0x00, sizeof(kek_key_id_1));  /* 1st KEK not used   */
memset (kek_key_id_2,  0x00, sizeof(kek_key_id_2));  /* 2nd KEK not used   */
memset (mac_key_id,    0x00, sizeof(mac_key_id));    /* Init 1st key token */
memset (macver_key_id, 0x00, sizeof(macver_key_id)); /* Init 2nd key token */

/* Generate a MAC/MACVER operational key pair */
CSNBKGN(&return_code,
        &reason_code,
        NULL,                                /* exit_data_length           */
        NULL,                                /* exit_data                  */
        key_form,
        key_length,
        mac_key_type,
        macver_key_type,
        kek_key_id_1,
        kek_key_id_2,
        mac_key_id,
        macver_key_id);

/* Check the return/reason codes. Terminate if there is an error.         */
if (return_code != 0 || reason_code != 0) {
  printf ("Key_Generate failed: ");             /* Print failing verb     */
  printf ("return_code = %ld, ",  return_code); /* Print return code      */
  printf ("reason_code = %ld.\n", reason_code); /* Print reason code      */
  return;
}
else
  printf ("Key_Generate successful.\n");

/* Set up initial values for MAC_Generate call */
return_code = 0;
reason_code = 0;
text_length = sizeof (INPUT_TEXT) - 1;          /* Length of MAC text     */
memcpy (text, INPUT_TEXT, text_length);         /* Define MAC input text  */
rule_array_count = 3;                           /* 3 rule array elements  */
memset (rule_array, ' ', sizeof(rule_array));   /* Clear rule array       */
memcpy (rule_array[0], MAC_PROCESSING_RULE, 8); /* 1st rule array element */
memcpy (rule_array[1], SEGMENT_FLAG,        8); /* 2nd rule array element */
memcpy (rule_array[2], MAC_LENGTH,          8); /* 3rd rule array element */
memset (chaining_vector, 0x00, 18);             /* Clear chaining vector  */
```

```
memset (mac_value, 0x00, sizeof(mac_value));    /* Clear MAC value        */

/* Generate a MAC based on input text */
CSNBMGN ( &return_code,
         &reason_code,
         NULL,                              /* exit_data_length          */
         NULL,                              /* exit_data                 */
         mac_key_id,                        /* Output from Key_Generate  */
         &text_length,
         text,
         &rule_array_count,
         &rule_array[0][0],
         chaining_vector,
         mac_value);

/* Check the return/reason codes. Terminate if there is an error.         */
if (return_code != 0 || reason_code != 0) {
  printf ("MAC Generate Failed: ");             /* Print failing verb     */
  printf ("return_code = %ld, ",  return_code); /* Print return code      */
  printf ("reason_code = %ld.\n", reason_code); /* Print reason code      */
  return;
}
else {
  printf ("MAC_Generate successful.\n");
  printf ("MAC_value = %s\n", mac_value);    /* Print MAC value (HEX-9)    */
}
/* Set up initial values for MAC_Verify call */
return_code = 0;
reason_code = 0;
rule_array_count = 1;                       /* 1 rule array element       */
memset (rule_array, ' ', sizeof(rule_array));/* Clear rule array          */
memcpy (rule_array[0], MAC_LENGTH, 8);      /* Rule array element         */
                                            /*  (use default Ciphering    */
                                            /*  Method and Segmenting     */
                                            /*  Control)                  */
memset (chaining_vector, 0x00, 18);         /* Clear the chaining vector  */

/* Verify MAC value */
CSNBMVR (&return_code,
         &reason_code,
         NULL,                              /* exit_data_length          */
         NULL,                              /* exit_data                 */
         macver_key_id,                     /* Output from Key_Generate  */
         &text_length,                      /* Same as for MAC_Generate  */
         text,                              /* Same as for MAC_Generate  */
         &rule_array_count,
         &rule_array[0][0],
         chaining_vector,
         mac_value);                        /* Output from MAC_Generate  */

/* Check the return/reason codes. Terminate if there is an error.         */
if (return_code != 0 || reason_code != 0) {
  printf ("MAC_Verify failed: ");               /* Print failing verb     */
  printf ("return_code = %ld, ", return_code);  /* Print return code      */
  printf ("reason_code = %ld.\n", reason_code); /* Print reason code      */
```

```
    return;
  }
  else                                  /* No error occurred          */
    printf ("MAC_Verify successful.\n");
}
```

การปรับปรุงทรูพุตด้วย CCA และตัวประมวลผลร่วม

เมื่อคุณใช้ CCA API คุณสมบัติของโฮสต์ของคุณ จะมีผลต่อผลการทำงานและทรูพุตของ 4765 สำหรับผลการดำเนินงานที่ดีที่สุดบนตัวประมวลผลร่วม 4765 ประเมินและออกแบบแอ็พพลิเคชันของคุณ จากการทำมัลติเธรดและมัลติโพรเซสซิง และจากการแคชคีย์ Data Encryption Standard (DES), Public-Key Algorithm (PKA) และ Advanced Encryption Standard (AES)

มัลติเธรดและการประมวลผลจำนวนมาก

แอ็พพลิเคชัน CCA ที่รันอยู่ภายใน 4765 สามารถประมวลผลคำร้องขอ CCA จำนวนมากได้อย่างพร้อมเพียงกัน ตัวประมวลผลร่วมมีองค์ประกอบของฮาร์ดแวร์ที่เป็นอิสระ ซึ่งรวมถึงเอ็นจิน Rivest-Shamir-Adleman algorithm (RSA), เอ็นจิน Data Encryption Standard (DES), CPU, ตัวสร้างหมายเลขแบบสุ่มและ อินเตอร์เฟสการสื่อสาร Peripheral Component Interconnect-X (PCI-X)อิลิเมนต์เหล่านี้สามารถทำงานร่วมกันได้ในเวลาเดียวกัน ประมวลผลผลส่วนต่างๆ ของ CCA verbs ต่างๆ ด้วยการทำงานบน verbs หลายๆ ตัว ในเวลาเดียวกัน ตัวประมวลผลร่วมสามารถเก็บอิลิเมนต์ฮาร์ดแวร์ทั้งหมดที่ไม่ว่างเพิ่มทรูพุต ของระบบโดยภายรวมให้มากขึ้น

หากต้องการใช้ประโยชน์ของความสามารถนี้ ระบบโฮสต์ของคุณ ต้องส่งคำร้องขอ CCA จำนวนมากไปยังตัวประมวลผลร่วมโดยไม่ต้องรอให้ดำเนินการคำร้องขอแต่ละรายการ ให้เสร็จสิ้นก่อนที่จะส่งไปยังคำร้องขอถัดไป วิธีที่ดีที่สุดในการส่งการร้องขอหลายรายการคือ ออกแบบแอ็พพลิเคชันโปรแกรมแบบมัลติเธรด ซึ่งแต่ละเธรดสามารถส่งคำร้องขอ CCA ไปยังตัวประมวลผลร่วมโดยแยกออกจากกัน ตัวอย่างเช่น เว็บเซิร์ฟเวอร์สามารถเริ่มต้น เธรดใหม่สำหรับแต่ละคำร้องขอที่ได้รับผ่านเน็ตเวิร์ก แต่ละเธรด จะส่งคำร้องขอการเข้ารหัสที่จำเป็นต่อตัวประมวลผลร่วมไปยังตัวประมวลผลร่วม ซึ่งเป็นอิสระจากที่เธรดอื่นๆ กะลังทำ โมเดลมัลติเธรดการันตี ว่าตัวประมวลผลร่วมจะไม่ถูกใช้อยู่ อ็อพชันอื่นๆ คือ มีแอ็พพลิเคชันโปรแกรมที่เป็นอิสระจำนวนมากที่ใช้ตัวประมวลผลร่วม ในเวลาเดียวกัน

การสร้างแคชสำหรับคีย์ DES, PKA และ AES

ซอฟต์แวร์ CCA สำหรับ 4765 เก็บสำเนาของ DES, PKA และคีย์ AES ที่เข้ารหัสไว้ (ไม่ใช้ข้อความปกติ) ล่าสุดใน แคชภายในโมดูลการรักษาความปลอดภัยคีย์ถูกเก็บอยู่ในรูปแบบที่ได้ถอดรหัส ตรวจสอบความถูกต้อง และพร้อมใช้งาน สำหรับการใช้หากคีย์เดียวกันนี้ถูกนำกลับมาใช้ในคำร้องขอ CCA ในภายหลัง 4765 สามารถใช้ สำเนาที่แคชแล้วและหลีกเลี่ยงค่าใช้จ่ายเพิ่มเติมกับการถอดรหัส และการตรวจสอบความถูกต้องของโทเค็นคีย์ นอกจากนี้ สำหรับคีย์ PKA แคชจะกำจัดค่าใช้จ่ายของการเรียกคืนคีย์จากแฟลชภายในหน่วยความจำ Erasable Programmable Read Only Memory (EPROM)

ตามผลลัพธ์แล้ว แอ็พพลิเคชันที่นำชุดของคีย์ทั่วไปกลับมาใช้สามารถรันได้เร็วกว่า การใช้คีย์อื่นสำหรับการดำเนินรายการแต่ละรายการ แอ็พพลิเคชันทั่วไปใช้ชุดของคีย์ DES ทั่วไป คีย์ PKA ส่วนบุคคล และคีย์ AES ที่เข้ารหัสไว้ และการสร้างแคชที่มีประสิทธิผลในการปรับปรุงทรูพุต พับลิกคีย์ PKA และคีย์ AES ที่ล้างข้อมูลแล้ว ซึ่งมีค่าใช้จ่ายในการประมวลผลเพียงน้อยจะไม่ถูกสร้างแคช

คำสั่งบทบาทดีฟอลต์เริ่มต้น

คุณลักษณะของบทบาทดีฟอลต์หลังจากตัวประมวลผลร่วมถูก กำหนดค่าเริ่มต้นและเมื่อไม่มีข้อมูลการควบคุมการเข้าถึงอื่นอยู่ ถูกอธิบายไว้ และ คำสั่งการควบคุมการเข้าถึงที่เปิดใช้งานถูกแสดงไว้

สำหรับคำสั่งบทบาทดีฟอลต์เริ่มต้น role ID เป็นค่าดีฟอลต์ และความแข็งแกร่งการพิสูจน์ตัวตนเป็นศูนย์ บทบาทดีฟอลต์ใช้ได้ในทุกๆ ครั้งของวัน และทุกๆ วันของสัปดาห์ เฉพาะฟังก์ชัน ที่อนุญาตเท่านั้นที่จำเป็นต้องโหลดข้อมูลการควบคุมสิทธิในการเข้าถึง

สำคัญ: โหมดการเข้ารหัสไม่ปลอดภัยเมื่อ ผู้ใช้ที่ไม่ได้พิสูจน์ตัวตนสามารถโหลดข้อมูลการควบคุมการเข้าถึงโดยใช้บทบาทดีฟอลต์ จำกัดคำสั่งเหล่านี้เพื่อเลือกบทบาท ของหัวหน้างาน

ตารางที่ 6แสดงคำสั่งการควบคุมการเข้าถึง ที่ถูกเปิดใช้งานในบทบาทดีฟอลต์เมื่อซอฟต์แวร์ CCA ถูกโหลดเริ่มต้นและเมื่อโหนด CCA ถูกกำหนดค่าเริ่มต้น

ตารางที่ 6. คำสั่งบทบาทดีฟอลต์เริ่มต้น

โค้ด	ชื่อคำสั่ง
X'0107'	การแฮชวิธีหนึ่ง นั่นคือ SHA-1
X'0110'	ตั้งค่านาฬิกา
X'0111'	กำหนดค่าเริ่มต้นให้กับอุปกรณ์อีกครั้ง
X'0112'	กำหนดค่าเริ่มต้นระบบการควบคุมการเข้าถึง
X'0113'	เปลี่ยนวันที่หมดอายุของโปรไฟล์ผู้ใช้
X'0114'	เปลี่ยนแปลงข้อมูลการพิสูจน์ตัวตนสำหรับโปรไฟล์ผู้ใช้
X'0115'	รีเซ็ตจำนวนความล้มเหลวของความพยายามในการล็อกออนสำหรับโปรไฟล์ผู้ใช้
X'0116'	อ่านข้อมูลการควบคุมสิทธิในการเข้าถึงแบบพับลิก
X'0117'	ลบโปรไฟล์ผู้ใช้
X'0118'	ลบบทบาท
X'0119'	โหนด Function-Control Vector
X'011A'	ล้างข้อมูล Function-Control Vector

เนื้อหาของบันทึกการทำงานที่เครื่องสามารถอ่านได้

ยูทิลิตี CLU สร้างล็อกไฟล์สองไฟล์ ไฟล์หนึ่งสำหรับการอ่าน และอีกไฟล์หนึ่งสำหรับอินพุตไปยังโปรแกรม

ล็อกไฟล์ที่เครื่องสามารถอ่านได้ (MRL) มีเอาต์พุตไบนารีจากตัวประมวลผลร่วมในการตอบกลับ ไปยังคำสั่งต่างๆ ที่ส่งให้กับตัวประมวลผลร่วม

ข้อมูลโดยละเอียดเกี่ยวกับเนื้อหาของ MRL มีอยู่ที่การพัฒนา IBM 4764 และ IBM 4765ติดต่อ IBM โดยใช้ แท็บการสนับสนุนและดาวน์โหลดในเว็บไซต์ผลิตภัณฑ์ IBM ที่ *http://www.ibm.com/security/cryptocards*

โค้ดระบุความผิดพลาดของไดร์เวอร์อุปกรณ์

ไดร์เวอร์อุปกรณ์สำหรับตัวประมวลผลร่วมจะมอนิเตอร์สถานะของการสื่อสาร กับตัวประมวลผลร่วมและการลงทะเบียนสถานะฮาร์ดแวร์ของตัวประมวลผลร่วม

แต่ละครั้งที่รีเซ็ตตัวประมวลผลร่วม และการรีเซ็ตไม่ได้เป็นสาเหตุ ทำให้เกิดความผิดพลาดหรือเปลี่ยนแปลงเหตุการณ์ ตัวประมวลผลร่วมจะรันผ่าน miniboot, power-on self-test (POST) การโหลดโค้ด และรูทีนสถานะ ในระหว่างกระบวนการนี้ ตัวประมวลผลร่วมจะพยายามประสานงานกับ ไดร์เวอร์อุปกรณ์ของระบบโฮสต์ การรีเซ็ตตัวประมวลผล ร่วมสามารถเกิดขึ้นได้เนื่องจากการเปิดซึ่งเป็น คำสั่ง **reset** ที่ส่งจากไดร์เวอร์อุปกรณ์ หรือ อาจเป็นเพราะกิจกรรมภายในตัวประมวลผลร่วม เช่น ความสมบูรณ์ของอัพเดตโค้ด

ความผิดพลาดของตัวประมวลผลร่วมหรือการเปลี่ยนวงจรตรวจสอบยังสามารถ รีเซ็ตตัวประมวลผลร่วมได้

โปรแกรม เช่น Coprocessor Load Utility (CLU) และ CCA Support Program สามารถรับสถานะที่ไม่ปกติได้ในรูปแบบโค้ดส่งคืนขนาด 4 ไบต์ จากไดร์เวอร์อุปกรณ์

โค้ด 4-ไบต์ที่เป็นไปได้ อยู่ในรูปแบบ X'8xxxxxxx' โค้ดที่ได้รับ บ่อยครั้ง ถูกอธิบายไว้ใน ตารางที่ 7หากคุณพบโค้ดในรูปแบบ XX'8340xxxx' หรือ X'8440xxxx' และโค้ดไม่อยู่ในตารางให้ติดต่อทีม IBM cryptographic ผ่านทางอีเมลจากหน้า Support บนเว็บไซต์ ผลิตภัณฑ์ IBM ที่ *http://www.ibm.com/security/cryptocards*>

ตารางที่ 7. โค้ดระบุความผิดพลาดไดร์เวอร์ Device-class ในคลาส X'8xxxxxxx'

โค้ดส่งคืน ขนาด 4 ไบต์ (ฐานหก)	เหตุผล	คำอธิบาย
8040FFBF	การบุกรุกจากภายนอก	การบุกรุกเพิ่มขึ้นเนื่องจากการเชื่อมต่อไฟฟ้า ทางเลือกกับตัวประมวลผลร่วม เงื่อนไขนี้สามารถรีเซ็ตได้
8040FFDA	แบตเตอรี่ไม่ทำงาน	แบตเตอรี่ได้รับอนุญาตให้ทำงานโดยไม่มีกำลังที่เพียงพอ หรือได้ถูกถอดออกแล้ว ตัวประมวลผลร่วมถูก zeroize และไม่ทำงานอีกต่อไป
8040FFDB	เปลี่ยน X-ray หรือแบตเตอรี่ไม่ทำงาน	ตัวประมวลผลร่วมถูก zeroized และไม่ทำงานอีกต่อไป
8040FFDF	X-ray หรือแบตเตอรี่ไม่ทำงาน	ตัวประมวลผลร่วมถูก zeroized และไม่ทำงานอีกต่อไป
8040FFEB	อุณหภูมิเปลี่ยน	ข้อจำกัดด้านอุณหภูมิสูงหรือต่ำเกินไป ตัวประมวลผลร่วมถูก zeroized และไม่ทำงานอีกต่อไป
8040FFF3	แรงดันไฟเปลี่ยน	ตัวประมวลผลร่วมถูก zeroized และไม่ทำงานอีกต่อไป
V8040FFF9	กับดักการเปลี่ยน	ตัวประมวลผลร่วมถูก zeroized และไม่ทำงานอีกต่อไป
8040FFFB	การรีเซ็ตบิตเปียู่	ตรวจพบแรงดันไฟล์ต่ำ อุณหภูมิการทำงานภายในของตัวประมวลผลเกินกว่าขีดจำกัด หรือไดร์เวอร์โฮสต์ส่งคำสั่งรีเซ็ต ลองย้ายหรือใส่ตัวประมวลผลร่วม ลงใน PCI-X bus
8040FFFE	คำเตือนแบตเตอรี่	กำลังไฟของแบตเตอรี่ไม่สำคัญ สำหรับขั้นตอนที่ต้องปฏิบัติตามเพื่อแทนที่ แบตเตอรี่ ดูที่คู่มือการติดตั้ง *IBM 4764 PCI-X Cryptographic Coprocessor*

ตารางที่ 7. โค้ดระบุความผิดพลาดไดร์เวอร์ Device-class ในคลาส X'8xxxxxx' (ต่อ)

โค้ดส่งคืน ขนาด 4 ไบต์ (ฐานหก)	เหตุผล	คำอธิบาย
804xxxxx (ตัวอย่างเช่น **80400005**)	ปัญหาด้านการสื่อสารโดยทั่วไป	ยกเว้นสำหรับโค้ด X'8040xxxx' ก่อนหน้า มีภาวะเพิ่มเติมเกิดขึ้น ในการสื่อสารระหว่างโฮสต์กับตัวประมวลร่วม ให้กำหนดว่า ระบบโฮสต์มีตัวประมวลผลร่วมจริง ลองย้าย หรือใส่ตัวประมวลผลร่วมลงใน PCI-X รันคำสั่งสถานะ CLU (ST) ถ้าปัญหายังอยู่ ให้ติดต่อทีม IBM cryptographic ผ่านทางอีเมลจากหน้า Support บนเว็บไซต์ผลิตภัณฑ์ IBM ที่ *http://www.ibm.com/security/cryptocards*
8340xxxx	โค้ด Miniboot-0	คลาสนี้ของโค้ดส่งคืนเกิดขึ้นจากระดับที่ต่ำสุด ของการทดสอบการรีเซ็ต ถ้าโค้ดในคลาสนี้เกิดขึ้นให้ติดต่อทีม IBM cryptographic ผ่านทาง อีเมลจากหน้า Support บนเว็บไซต์ผลิตภัณฑ์ IBM ที่ *http://www.ibm.com/security/cryptocards*
8340038F	ข้อผิดพลาดในการสร้างหมายเลขแบบสุ่ม	ให้มอนิเตอร์ตัวสร้างหมายเลขแบบสุ่มต่อ ซึ่งตรวจพบปัญหาที่อาจเกิดขึ้นได้ มีความน่าจะเป็นเชิงสถิติขนาดเล็ก ของเหตุการณ์นี้ที่เกิดขึ้นโดยไม่ได้ระบุปัญหาต่อเนื่อง รันคำสั่ง CLU status (ST) อย่างน้อยสองครั้งเพื่อระบุว่า สภาวะสามารถถูกเคลียร์ได้หรือไม่
8440xxxx	โค้ด Miniboot-1	คลาสนี้ของโค้ดการส่งคืนเกิดขึ้นจากการเปลี่ยน POST และโค้ดของการโหลดโค้ด
844006B2	การลงนามที่ไม่ถูกต้อง	การลงนามบนข้อมูลที่ส่งจากยูทิลีตี CLU ไปที่ miniboot ไม่สามารถถูก ตรวจสอบได้โดย miniboot โปรดมั่นใจว่า คุณกำลังใช้ไฟล์ที่เหมาะสม (ตัวอย่างเช่น **CR1**xxxxx.clu กับ **CE1**xxxxx.clu) ถ้าปัญหายังอยู่ รับเอาต์พุตของรายงานสถานะ CLU และส่งต่อรายงานพร้อมกับรายละเอียดของงานที่คุณต้องการทำไปที่ ทีม IBM cryptographic ผ่าน ทางอีเมลจากหน้า Support บนเว็บไซต์ผลิตภัณฑ์ IBM ที่ *http://www.ibm.com/security/cryptocards*

การโคลนคีย์หลัก

ส่วนนี้ให้คำแนะนำสำหรับการโคลนคีย์หลัก และให้ข้อควรพิจารณาการควบคุมการเข้าถึงขณะทำการโคลน

ภาพรวมการโคลนคีย์หลัก

ขั้นตอนการโคลนแสดงกรอบถึงวิธีการโคลนคีย์หลักจากหนึ่งตัวประมวลผลร่วม ไปยังตัวประมวลผลร่วมอื่นๆ โดยใช้ยูทิลิตี Cryptographic Node Management (CNM)

หมายเหตุ: ตรวจสอบว่า ยูทิลิตีที่อยู่ในระดับเดียวกันบนระบบทั้งหมดที่เกี่ยวข้องในขั้นตอนของการโคลน

ขั้นตอนการโคลนคีย์หลักจะไม่กำหนดข้อสรุปเกี่ยวกับ เซิร์ฟเวอร์ที่มีตัวประมวลผลร่วมที่ใช้สำหรับ:

- Share administration (**โหนด SA**)
- ต้นทางคีย์หลัก (โหนด **CSS** coprocessor share-signing)
- ปลายทางคีย์หลัก (โหนด **CSR** coprocessor share-receiving)

หมายเหตุ: การโคลนคีย์หลัก AES ไม่ได้รับการสนับสนุน

คีย์ SA สามารถตั้งอยู่ในตัวประมวลผลร่วมตัวเดียวกันกับคีย์ CSS หรือคีย์ CSR หรือสามารถตั้งอยู่ในโหนดตัวประมวลผลร่วมที่แยกจากกัน ตัวประมวลผลร่วมใดๆ สามารถตั้งอยู่ด้วยกันในเซิร์ฟเวอร์เครื่องเดียวกันได้ หากตัวประมวลผลร่วมจำนวนมาก ที่มี CCA พร้อมใช้งาน

โพรซีเดอร์จะละเว้นการดำเนินการของตัวดำเนินการเพื่อล็อกออนและล็อกออฟ เนื่องจากขั้นตอนเหล่านี้ขึ้นอยู่กับบทบาทที่ระบุเฉพาะที่ใช้งานอยู่ที่การติดตั้งของคุณ คุณสามารถสับเปลี่ยนตัวประมวลผลร่วมได้ เมื่อคุณกำลังใช้มากกว่า หนึ่งตัวประมวลผลร่วมภายในเซิร์ฟเวอร์

โพรซีเดอร์นี้ถูกแบ่งออกเป็นหลายๆ เฟสตามกรอบที่อยู่ใน ตารางที่ 8

ตารางที่ 8. ภาพรวมของเฟสเกี่ยวกับขั้นตอน ของการโคลนคีย์หลัก

เฟส	โหนด	ภารกิจ
1	SA	สร้างโหนดการควบคุมดูแลการแบ่งใช้ สร้าง ฐานข้อมูล SA สร้างคีย์ SA และเก็บพับลิกคีย์ และแฮชภายในฐานข้อมูล SA
2a	Source	สร้างโหลดต้นทาง สร้างคีย์ CSS และเพิ่มพับลิกคีย์ให้กับฐานข้อมูล SA ติดตั้งพับลิกคีย์ SA
2b	SA	รับรองคีย์ CSS และเก็บใบรับรอง ในฐานข้อมูล SA
สำหรับแต่ละโหนดเป้าหมาย ให้ทำซ้ำขั้นตอนทั้ง 3 ขั้นตอน		
3a	Target	สร้างโหนดปลายทาง สร้างฐานข้อมูล CSR สร้างคีย์ CSR และเพิ่มพับลิกคีย์ให้กับฐานข้อมูล CSR สำหรับ โหนดนี้ ติดตั้งพับลิกคีย์ SA
3b	SA	รับรองคีย์ CSR และเก็บใบรับรอง ภายในฐานข้อมูล CSR สำหรับโหนดเป้าหมาย
3c	Source	รับข้อมูลการตรวจสอบการแบ่งใช้และ คีย์หลักปัจจุบัน
3d	Target	ติดตั้งการแบ่งใช้และยืนยันคีย์หลักใหม่ ตั้งค่าคีย์หลัก

ก่อนการเริ่มต้นขั้นตอนการโคลนคีย์หลัก ขอแนะนำให้ คุณกรอกแบบฟอร์มที่พบในตาราง ตารางที่ 9และภาพa รูปที่ 8 ในหน้า 55 ให้สมบูรณ์

ตารางที่ 9. ความรับผิดชอบในการโคลน โปรไฟล์ และบทบาท

ภารกิจ	โหนด	โปรไฟล์	บทบาท	ความรับผิดชอบ
การควบคุมสิทธิในการเข้าถึงการตรวจสอบระบบ	SA			
สร้างคีย์ SA	SA			
ลงทะเบียนการแฮชคีย์ SA	SA			
ลงทะเบียนคีย์ SA	SA			
การควบคุมสิทธิในการเข้าถึงการตรวจสอบระบบ	CSS			
สร้างคีย์ CSS	CSS			
ขอรับคีย์หลัก CSS	CSS			
ลงทะเบียนการแฮชคีย์ SA	CSS			

ตารางที่ 9. ความรับผิดชอบในการโคลน โปรไฟล์ และบทบาท (ต่อ)

ภารกิจ	โหนด	โปรไฟล์	บทบาท	ความรับผิดชอบ
ลงทะเบียนคีย์ SA	CSS			
รับรองคีย์ CSS	SA			
การควบคุมสิทธิในการเข้าถึงการตรวจสอบระบบ	CSR1			
สร้างคีย์ CSR	CSR1			
ลงทะเบียนการแฮชคีย์ SA	CSR1			
ลงทะเบียนคีย์ SA	CSR1			
รับรองคีย์ CSR1	SA			
ขอรับการแบ่งใช้	CSS			
ติดตั้งการแบ่งใช้	CSR1			
ตรวจสอบ CSR ใหม่	CSR1			
ตั้งค่าคีย์หลัก CSR	CSR1			
การควบคุมสิทธิในการเข้าถึงการตรวจสอบระบบ	CSR2			
สร้างคีย์ CSR	CSR2			
ลงทะเบียนการแฮชคีย์ SA	CSR2			
ลงทะเบียนคีย์ SA	CSR2			
รับรองคีย์ CSR2	SA			
ขอรับการแบ่งใช้	CSS			
ติดตั้งการแบ่งใช้	CSR2			
ตรวจสอบ CSR ใหม่	CSR2			
ตั้งค่าคีย์หลัก CSR	CSR2			

NODE INFORMATION	Node	Machine	Selector Number	Coprocessor Serial Number	Data Base Path and Name
	SA Node Control				(sa.db)
	CSS Node Source				(sa.db)
	CSR Node Target 1				(csr1.db)
	CSR Node Target 2				(csr2.db)

SA-KEY HASH				

NUMBER OF SHARES	Minimum: “m”	Maximum: “n”

SHARES DISTRIBUTION	Obtained from:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Installed into CSR-1:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

	Obtained from:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Installed into CSR-2:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

รูปที่ 8. เวิร์กชีตข้อมูลการโคลน

ขั้นที่ 1 สำหรับการโคลนคีย์หลัก: การสร้างโหนดการควบคุมดูแล การแบ่งใช้

เมื่อต้องการใช้ตัวประมวลผลร่วมเป็นโหนด share administration (SA) ให้ทำตาม ขั้นตอนจากการโคลนคีย์หลักที่กล่าวไว้ในตารางที่ 10 ในหน้า 56 ตัวประมวลผลร่วมนี้ยังสามารถทำหน้าที่ เป็นโหนดต้นทางคีย์หลัก หรือโหนดปลายทางคีย์หลัก

สิ่งที่จำเป็นต้องมีก่อน: ก่อนการรันขั้นตอน นี้ให้คุณทำความคุ้นเคยกับขั้นตอนที่อธิบายในส่วน “สถานการณ์จำลอง: การโคลนคีย์หลัก DES หรือ PKA” ในหน้า 24 และบทเกี่ยวกับการทำความเข้าใจ และการจัดการคีย์ในคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors*

เมื่อต้องการสร้างโหนด SA ให้ทำตามขั้นตอนในตารางต่อไปนี้ให้สมบูรณ์:

ตารางที่ 10. การโคลนโพรซีเดอร์คีย์หลัก: การสร้างโหนด SA

เฟส	ภารกิจ	✓
1.1	ตรวจสอบความเหมาะสมของการควบคุมสิทธิในการเข้าถึง	
1.2	ดำเนินการซิงโครไนซ์เวลาและตรวจสอบว่า การพิสูจน์ตัวตน (`fcv_td4kECC521.crt`) ถูกติดตั้งแล้ว	
1.3	ยืนยัน (หรือติดตั้ง) คีย์หลัก	
1.4	การใช้สิ่งอำนวยความสะดวกของระบบปฏิบัติการของคุณ ให้ลบฐานข้อมูล SA ก่อนหน้านี้ออกจากสื่อบันทึก ฐานข้อมูล SA	
1.5	หากไม่ได้สร้างไว้ให้ป้อน environment ID (EID) โดยทำขั้นตอนต่อไปนี้ให้สมบูรณ์: • คลิก **Crypto Node** > **ตั้งค่า environment ID** • ป้อน EID คลิก **คลิก**	
1.6	สร้างคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **สร้าง คีย์** > **คีย์การควบคุมดูแลการแบ่งใช้** • ยอมรับพับลิกคีย์ SA ที่เป็นค่าดีฟอลต์และเลเบลไพรเวตคีย์ และป้อนตำแหน่งและชื่อของฐานข้อมูล SA (`sa.db`) • คลิก **สร้าง** • บันทึกค่าการแฮชคีย์ SA สำหรับใช้ในภายหลังในโพรซีเดอร์	
1.7	รีจิสเตอร์การแฮชพับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **สร้าง คีย์** > **คีย์การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์คีย์การควบคุมดูแล การแบ่งใช้** > **การแฮช SA-Key** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) • ป้อนการแฮช SA-key คลิก **รีจิสเตอร์**	
1.8	รีจิสเตอร์พับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **สร้าง คีย์** > **คีย์การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์คีย์การควบคุมดูแล การแบ่งใช้** > **การแฮช SA-Key** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) คลิก รีจิสเตอร์	

ขั้นที่สอง 2 สำหรับการโคลนคีย์หลัก: การสร้างโหนดต้นทาง

การใช้ตัวประมวลผลร่วมถูกกำหนดเป็นโหนดต้นทางสำหรับคีย์หลัก ให้ทำตามขั้นตอนสำหรับการโคลนคีย์หลักที่กล่าวไว้ในตารางที่ 11ตัวประมวลผลร่วมนี้ ยังสามารถใช้เป็นโหนด SA

ตารางที่ 11. การโคลนคีย์หลัก: การสร้าง โหนด (CSS) ต้นทาง

เฟส	ภารกิจ	✓
2a.1	ตรวจสอบความเหมาะสมของการควบคุมสิทธิในการเข้าถึง	
2a.2	ดำเนินการซิงโครไนซ์เวลาและตรวจสอบว่า การพิสูจน์ตัวตน `fcv_td4kECC521.crt` ถูกติดตั้งแล้ว	

ตารางที่ 11. การโคลนคีย์หลัก: การสร้าง โหนด (CSS) ต้นทาง (ต่อ)

เฟส	ภารกิจ	✓
2a.3	ยืนยันหมายเลขลำดับของตัวประมวลผลร่วม: • คลิก **Crypto Node** > **สถานะ** • คลิก **อะแด็ปเตอร์** • จดบันทึกหมายเลขลำดับของตัวประมวลผลร่วม คลิก **ยกเลิก**	
2a.4	ยืนยัน (หรือติดตั้ง) คีย์หลัก	
2a.5	รับข้อมูลการตรวจสอบคีย์หลักปัจจุบัน: • คลิก **คีย์หลัก** > **ตรวจสอบ** > **ปัจจุบัน** • คลิก **บันทึก** กับสื่อบันทึกขนย้าย คลิก **ยกเลิก**	
2a.6	หากไม่ได้สร้างไว้ให้ป้อน environment ID (EID): • คลิก **Crypto Node** > **ตั้งค่า environment ID** • ป้อน EID คลิก **คลิก**	
2a.7	ถ้ายังไม่ได้สร้าง ให้ตั้งค่าหมายเลขค่าการแบ่งใช้ m และ n: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **ตั้งค่า จำนวนการแบ่งใช้** • ตั้งค่าจำนวนสูงสุดและต่ำสุดของการแบ่งใช้ที่จำเป็น คลิก **โหลด**	
2a.8	สร้างคีย์ CSS: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **สร้าง คีย์** > **คีย์ CSS** • ป้อนเลเบลของคีย์ CSS (ตัวอย่าง CSS.KEY) • ยืนยันหมายเลขลำดับของตัวประมวลผลร่วม • ยืนยันหรือป้อนชื่อฐานข้อมูล SA และตำแหน่ง • คลิก **สร้าง**	
2a.9	ลงทะเบียนการแฮชพับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์ คีย์การควบคุมดูแลการแบ่งใช้** > **การแฮช SA-Key** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) • ป้อนการแฮชคีย์ SA คลิก **รีจิสเตอร์**	
2a.10	ลงทะเบียนพับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์ คีย์การควบคุมดูแลการแบ่งใช้** > **คีย์ SA** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) คลิก**รีจิสเตอร์**	

ขั้นที่ 3 สำหรับการโคลนคีย์หลัก: การสร้างโหนดปลายทาง และการโคลนคีย์หลัก

การใช้โหนดที่กำหนดไว้ สร้างโหนดปลายทางและโคลน คีย์หลัก ทำตามขั้นตอนสำหรับการโคลนคีย์หลักที่กล่าวไว้ใน ตาราง ที่ 12 ในหน้า 58 ตัวประมวลผลร่วมนี้ ยังสามารถใช้เป็นโหนด SA

ตารางที่ 12. การโคลนคีย์หลัก: การสร้าง โหนด CSR และการโคลนคีย์หลัก

เฟส	โหนด	ภารกิจ	✓
ที่โหนดปลายทาง			
3a.1	เป้าหมาย	ตรวจสอบความเหมาะสมของการควบคุมสิทธิในการเข้าถึง	
3a.2	เป้าหมาย	ดำเนินการซิงโครไนซ์เวลาและตรวจสอบว่า การพิสูจน์ตัวตน `fcv_td2k.crt` ถูกติดตั้งแล้ว	
3a.3	เป้าหมาย	ยืนยันหมายเลขลำดับของตัวประมวลผลร่วม: • คลิก **Crypto Node** > **สถานะ** • คลิก **อะแด็ปเตอร์** • จดบันทึกหมายเลขลำดับของตัวประมวลผลร่วม คลิก **ยกเลิก**	
3a.4	เป้าหมาย	ตรวจสอบการมีอยู่ของคีย์หลัก (แบบชั่วคราว)	
3a.5	เป้าหมาย	หากไม่ได้สร้างไว้ให้ป้อน environment ID (EID): • คลิก **Crypto Node** > **ตั้งค่า environment ID** > **Crypto Node** • ป้อน EID (ตัวอย่างเช่น CSR1 NODE และส่วนขยายด้วยช่องว่าง จนถึง 16 ตัวอักษรที่ป้อน) • คลิก **โหลด**	
3a.6	เป้าหมาย	ถ้ายังไม่ได้สร้าง ให้ตั้งค่าหมายเลขค่าการแบ่งใช้ *m* และ *n* : • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **ตั้งค่า จำนวนการแบ่งใช้** • ตั้งค่าจำนวนสูงสุดและต่ำสุดของการแบ่งใช้ที่จำเป็น • คลิก **โหลด**	
3a.7	เป้าหมาย	การใช้สิ่งอำนวยความสะดวกของระบบปฏิบัติการของคุณ ให้ลบไฟล์ข้อมูล `csr.db`	
3a.8	เป้าหมาย	สร้างคีย์ CSR: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **สร้าง คีย์** > **คีย์ CSR** • ป้อนเลเบลของคีย์ CSR (ตัวอย่างเช่น CSR1.KEY) • ยืนยันหมายเลขลำดับของตัวประมวลผลร่วม • เลือกขนาดของคีย์ • จัดเตรียมชื่อฐานข้อมูล CSR และตำแหน่ง (ตัวอย่างเช่น CSR1.DB) • คลิก **สร้าง**	
3a.9	เป้าหมาย	ลงทะเบียนการแฮชพับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์ การควบคุมดูแลการแบ่งใช้** > **การแฮช SA-Key** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) • ป้อนการแฮชคีย์ SA คลิก **รีจิสเตอร์**	
3a.10	เป้าหมาย	ลงทะเบียนพับลิกคีย์ SA: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รีจิสเตอร์ การควบคุมดูแลการแบ่งใช้** > **คีย์ SA** • ป้อนชื่อไฟล์และตำแหน่งฐานข้อมูล SA คลิก **ถัดไป** • ป้อนเลเบลพับลิกคีย์ SA (หรือยอมรับค่าดีฟอลต์) คลิก**รีจิสเตอร์**	

ตารางที่ 12. การโคลนคีย์หลัก: การสร้าง โหนด CSR และการโคลนคีย์หลัก (ต่อ)

เฟส	โหนด	ภารกิจ	✓
ที่โหนด SA			
3b.1	SA	รับรองคีย์ CSS (ตามต้องการ): • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รับรอง คีย์** > **คีย์ CSS** • ป้อนชื่อและพาธสำหรับฐานข้อมูล SA คลิก **ถัดไป** • ยืนยันเลเบลของคีย์ CSS หมายเลขลำดับของตัวประมวลผล และ ID สภาพแวดล้อมของ SA • คลิก **รับรอง**	
3b.2	SA	รับรองคีย์ CSR: • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รับรอง คีย์** > **คีย์ CSS** • ป้อนชื่อและพาธสำหรับฐานข้อมูล SA และ CSR คลิก **ถัดไป** • ยืนยันเลเบลของคีย์ SA เลเบลของคีย์ CSR และ ID สภาพแวดล้อม SA • ป้อนหมายเลขลำดับ CSR • คลิก **รับรอง**	
ที่โหนด ต้นทาง			
3c.1	ต้นทาง	รับจำนวนขั้นต่ำของการแบ่งใช้ m และ n ดำเนินการขั้นตอนย่อยต่อไปนี้สำหรับแต่ละการแบ่งใช้โปรดสังเกตว่า ล็อกออนและล็อกออฟ อาจจำเป็นต้องการขอรับการแบ่งใช้แต่ละครั้ง • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **รับ การแบ่งใช้** • เลือกการแบ่งใช้โปรดสังเกตว่า หากคุณกำลังขอรับชุดของการแบ่งใช้เพิ่มเติม ข้อความที่แจกจ่ายแล้วอาจไม่มีความหมาย • ป้อนชื่อและพาธสำหรับฐานข้อมูล SA และ CSR คลิก **ถัดไป** • ยืนยันเลเบลของคีย์ CSS หมายเลขลำดับของตัวประมวลผลร่วม CSS และหมายเลขลำดับของตัวประมวลผลร่วม CSR • คลิก **รับการแบ่งใช้** ทำซ้ำตามความต้องการ	
ที่โหนด ปลายทาง			

ตารางที่ 12. การโคลนคีย์หลัก: การสร้าง โหนด CSR และการโคลนคีย์หลัก (ต่อ)

เฟส	โหนด	ภารกิจ	✓
3d.1	เป้าหมาย	ติดตั้งจำนวนการแบ่งใช้ m และ n ให้ดำเนินการ กับการแบ่งใช้แต่ละครั้งและการสังเกตถึงการตอบกลับ การตอบกลับ บ่งชี้ว่า เมื่อมีการแบ่งใช้ที่เพียงพอถูกติดตั้งในรูปของ คีย์หลักใหม่ หมายเหตุ การล็อกออน และการล็อกออฟอาจต้องการติดตั้งการแบ่งใช้และครั้ง • คลิก **Crypto Node** > **การควบคุมดูแลการแบ่งใช้** > **โหลด การแบ่งใช้** • เลือกการแบ่งใช้ • ป้อนชื่อและพาธสำหรับฐานข้อมูล CSR และ SA **ถัดไป** • ยืนยันเลเบลคีย์ CSS หมายเลขลำดับตัวประมวลผลร่วมของ CSS และหมายเลขลำดับตัวประมวลผลร่วมของ CSR • คลิก **โหลดการแบ่งใช้** สังเกตถึงการตอบกลับ การโหลดการแบ่งใช้ทีเพียงพอ จะทำให้คีย์หลักใหม่เสร็จสิ้น ทำซ้ำตามความต้องการ	
3d.2	เป้าหมาย	ยืนยันคีย์หลักใหม่: • คลิก **คีย์หลัก** > **ตรวจสอบ** > **สร้าง** • คลิก **เปรียบเทียบ** หรือ **เลือกไฟล์** หรือคลิก **ตกลง** หรือ **คลิก ยกเลิก**	
3d.3	เป้าหมาย	ลบไฟล์ข้อมูล `csr.db` นี่ไม่ใช่ปัญหาด้านการรักษาความปลอดภัย แต่หลีกเลี่ยงความซับซ้อนขณะที่ดำเนินการโคลน คีย์หลัก	
3d.4	เป้าหมาย	ตั้งค่าคีย์หลักตามความเหมาะสม: • คลิก **คีย์หลัก** > **ตั้งค่า** • คลิก **OK**	

ข้อควรพิจารณาในการควบคุมการเข้าถึงเมื่อโคลน

มีคลาสของบทบาทอยู่สามคลาสในการพิจารณาสำหรับการดำเนินการโคลน

- บทบาทที่โหนด share administration (SA)
- บทบาทที่โหนดต้นทาง: โหนด coprocessor share signing (CSS)
- บทบาทที่โหนดปลายทาง: โหนด coprocessor share signing (CSS)

นโยบายความปลอดภัยของคุณ ต้องนิยามผู้ที่จะมีสิทธิ:

- สร้างคีย์หลักสุ่มที่โหนดต้นทาง
- ตั้งค่าคีย์หลัก การดำเนินการที่นำคีย์หลักไปสู่ การดำเนินการ เมื่อคีย์หลักเปลี่ยนแปลงไป คีย์ที่เปลี่ยนรหัสโดยคีย์หลักต้องถูกอัพเดต
- สร้างคีย์ Rivest-Shamir-Adleman (RSA) ที่เก็บไว้เพื่อรับรองพับลิกคีย์ของโหนดต้นทาง และโหนดปลายทาง (คีย์ SA) และเพื่อสร้างคีย์ที่เก็บไว้ที่โหนดต้นทาง (CSS) และโหนดปลายทาง (CSR)
- รีจิสเตอร์คีย์ SA และการแฮช และระบุว่าจะถูกแยก ความรับผิดชอบหรือไม่

นอกจากนั้น คุณต้องตัดสินใจถึงจำนวนโหนดที่ต้องทำงานร่วมกันเพื่อ โคลนคีย์หลัก ซึ่งต้องถูกเลือกเพื่อหลีกเลี่ยงความขัดแย้ง

ในการตัดสินใจเลือกค่า m และ n คุณต้องพิจารณา เมื่อการโคลนเข้าแทนที่และคุณจำเป็นต้องสร้างคีย์หลักขึ้นใหม่ จากจำนวนของการแบ่งใช้ที่น้อยกว่าจำนวนทั้งหมดที่ขอรับจากโหนดต้นทางหรือไม่ (เนื่องจากความล้มเหลวในการแบ่งใช้หรือความไม่พร้อมใช้งานของบุคคลตั้งแต่ หนึ่งรายขึ้นไปซึ่งสามารถขอรับหรือติดตั้งการแบ่งใช้ได้)

หมายเหตุ: ยูทิลิตี cryptographic node management (CNM) วางการแบ่งใช้ทั้งหมดจากโหนดใน ไฟล์ `csr.db` แต่ละการแบ่งใช้ถูกเข้ารหัสภายใต้คีย์ data encryption standard (DES) เฉพาะที่มีความยาวเป็นสามเท่า ซึ่งเข้ารหัสไว้โดยพับลิกคีย์ CSR ของโหนดเป้าหมาย

ตารางที่ 13จัดเตรียมคำแนะนำสำหรับการเลือก สิทธิในการเรียกใช้งานกับบทบาทที่ถูกเชื่อมโยงกับการโคลน

ตารางที่ 13. คำสั่ง CCA ที่เกี่ยวข้องกับการโคลน คีย์หลัก

โค้ด	ชื่อคำสั่ง	ชื่อ Verb	ข้อควรพิจารณา
X'001A'	ตั้งค่าคีย์หลัก	**Master_Key_Process**	วิกฤต บทบาทนี้ต้องรับรู้เนื้อหา ของรีจิสเตอร์คีย์หลักใหม่และนัยยะของ การเปลี่ยนแปลงคีย์หลัก
X'001D'	คำนวณรูปแบบการตรวจสอบความถูกต้อง	**จำนวนมาก**	ทั้งหมด
X'0020'	สร้างคีย์หลักแบบสุ่ม	**Master_Key_Process**	ไม่มีความรุนแรง ยกเว้นจะกรอกรายละเอียดของรีจิสเตอร์ คีย์หลักใหม่
X'0032'	ล้างข้อมูลการลงทะเบียนคีย์หลักใหม่	**Master_Key_Process**	บทบาทนี้กำหนดให้กับบทบาทที่สามารถตั้งค่าคีย์หลักได้ บทบาทสามารถแทนที่การแบ่งใช้ที่รวบรวม ซึ่งต้องเป็นการทำงานร่วมกันเฉพาะกับคำสั่ง สร้างคีย์หลักแบบสุ่ม
X'0033'	ล้างข้อมูลการลงทะเบียนคีย์หลักเก่า	**Master_Key_Process**	ตามปกติแล้ว ไม่ได้ใช้
X'008E'	สร้างคีย์	**Key_Generate Random_Number_Generate**	ทั้งหมด
X'0090'	เปลี่ยนรหัสอีกครั้งในคีย์หลักปัจจุบัน	**Key_Token_Change**	บทบาทนี้ขึ้นกับ ผู้ที่จะอัพเดตคีย์การทำงานที่เข้ารหัสโดยคีย์หลัก
X'0100'	PKA96 การสร้างลายเซ็นแบบดิจิทัล	**Digital_Signature_Generate**	บทบาทนี้รับรองคีย์ SA, CSS และ CSR
X'0101'	PKA96 การตรวจสอบลายเซ็นแบบดิจิทัล	**Digital_Signature_Verify**	ทั้งหมด
X'0102'	PKA96 การเปลี่ยนโทเค็นคีย์	**PKA_Key_Token_Change**	บทบาทนี้ขึ้นกับ ผู้ที่จะอัพเดตคีย์การทำงานที่เข้ารหัสโดยคีย์หลัก
X'0103'	PKA96 PKA การสร้างคีย์	**PKA_Key_Generate**	บทบาทนี้จำเป็นในการสร้างคีย์ SA, CSS, และ CSR
X'0107'	การแฮชวิธีหนึ่ง นั่นคือ SHA-1	**One_Way_Hash**	ทั้งหมด
X'0114'	เปลี่ยนแปลงข้อมูลการพิสูจน์ตัวตนสำหรับโปรไฟล์ผู้ใช้	**Access_Control_Initialization**	บทบาทนี้อนุญาตให้เปลี่ยน passphrase ใน โปรไฟล์ทั้งหมด โปรดใช้ด้วยความระมัดระวัง
X'0116'	อ่านข้อมูลการควบคุมการเข้าถึงพับลิก	**Access_Control_Maintenance**	ทั้งหมด

ตารางที่ 13. คำสั่ง CCA ที่เกี่ยวข้องกับการโคลน คีย์หลัก (ต่อ)

โค้ด	ชื่อคำสั่ง	ชื่อ Verb	ข้อควรพิจารณา
X'011C'	ตั้งค่า EID	**Cryptographic_Facility_Control**	บทบาทนี้จำเป็นในการเซ็ตอัพโหนด CSS และ CSR
X'011D'	เริ่มต้นการโคลนคีย์หลัก	**Cryptographic_Facility_Control**	บทบาทนี้จำเป็นในการเซ็ตอัพค่า m จาก n ที่โหนด CSS และ CSR
X'0200'	PKA การลงทะเบียนการแฮชพับลิกคีย์	**PKA_Public_Key_Hash_Register**	บทบาทนี้ต้องถูกใช้ที่โหนด CSS และ CSR เพื่อตรวจสอบว่า คีย์ SA เป็นที่รู้จักแยกความรับผิดชอบกับ X'0201'
X'0201'	PKA การลงทะเบียนพับลิกคีย์	**PKA_Public_Key_Register**	บทบาทนี้ต้องถูกใช้ที่โหนด CSS และ CSR เพื่อตรวจสอบว่า คีย์ SA เป็นที่รู้จักแยกความรับผิดชอบกับ X'0200'
X'0203'	ลบคีย์ที่มีอยู่ทิ้ง	**Retained_Key_Delete**	บทบาทนี้ถูกใช้เพื่อลบคีย์ SA, CSS, และ CSR ที่หมดอายุ โปรดระวังเกี่ยวกับ denial of service
X'0204'	PKA การสร้างการโคลนคีย์	**PKA_Key_Generate**	บทบาทนี้จำเป็นในการสร้างคีย์ CSS และ CSR
X'0211' - X'021F'	ขอรับข้อมูลการโคลน (แบ่งใช้)	**Master_Key_Distribution**	บทบาทนี้ถูกกำหนดโปรไฟล์และบทบาทสำหรับการแบ่งใช้แต่ละครั้ง เพื่อบังคับให้แยกความรับผิดชอบ
X'0221' - X'022F'	ติดตั้งข้อมูลการโคลน (แบ่งใช้)	**Master_Key_Distribution**	บทบาทนี้ถูกกำหนดโปรไฟล์และบทบาทสำหรับการแบ่งใช้แต่ละครั้ง เพื่อบังคับให้แยกความรับผิดชอบ
X'0230'	แสดงคีย์ที่เก็บไว้	**Retained_Key_List**	ทั้งหมด

ข้อควรพิจารณาเกี่ยวกับการคุกคามสำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล

พิจารณาถึงการคุกคามต่างๆ เมื่อคุณปรับใช้ IBM 4765 กับ IBM Common Cryptographic Architecture (CCA) Support Program ในแอ็พพลิเคชันที่มีการลงนาม แบบดิจิทัล มีการอภิปรายจำนวนมากที่เรียกใช้งานในสภาพแวดล้อมอื่น ซึ่งคุณอาจนำตัวประมวลผลร่วมมาใช้

องค์กรที่มี certification authority (CA), registration authority (RA), Online Certificate Status Protocol (OCSP) responder หรือเซอร์วิสการประทับเวลาภายในการดำเนินการจำเป็นต้องพิจารณาถึง วิธีการติดตั้งที่จะกำหนดการคุกคามที่หลากหลาย ตารางที่ 14 ในหน้า 63 แสดงรายการคุกคามที่สำคัญและแสดงการออกแบบผลิตภัณฑ์และโซลูชันการนำไปใช้งาน กับการคุกคามต่างๆ เหล่านี้ หมายเหตุอธิบายขั้นตอนที่คุณจำเป็นต้องพิจารณา เพื่อถ่ายโอนปัญหาเพิ่มเติม

ดูที่คู่มือ IBM CCA Basic Services Reference and Guide for the *IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* อธิบาย การดำเนินการที่คุณใช้ได้ในการปรับใช้ตัวประมวลผลร่วม นโยบายที่ต้องพิจารณา ฟังก์ชันแอ็พพลิเคชันที่จะนำมารวมไว้

อ่านเนื้อหา ตารางที่ 14 ในหน้า 63 หลังจากที่คุณได้ทำการตัดสินใจในขั้นแรกเกี่ยวกับการติดตั้งของคุณ

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล

การอภิปรายเกี่ยวกับการคุกคาม	**การถ่ายโอนการคุกคาม**
การคุกคามที่เชื่อมโยงกับ การจู่โจมทางฟิสิคัลบนตัวประมวลผลร่วม	
การโพรบแบบฟิสิคัลของตัวประมวลผลร่วม ฝ่ายตรงข้าม อาจดำเนินการโพรบแบบฟิสิคัลของตัวประมวลผลร่วม เพื่อแสดงข้อมูลการออกแบบและเนื้อหาของการดำเนินการ การโพรบบางส่วนอาจประกอบด้วย การทำงานกับระบบไฟฟ้า แต่ถูกอ้างถึงไว้ที่นี่เป็นฟิสิคัล เนื่องจากต้องการติดต่อ โดยตรงกับฟังก์ชันภายในตัวประมวลผลร่วม การโพรบแบบฟิสิคัลอาจนำมาซึ่ง การอ่านข้อมูลจากตัวประมวลผลร่วมผ่านเทคนิคที่ใช้ในการวิเคราะห์ความล้มเหลว IC โดยทั่วไปและการส่งเสริมวิศวกรรมย้อนทาง ของ IC เป้าหมายของฝ่ายตรงข้าม คือ ระบุรายละเอียดของการออกแบบเป็นกลไกความปลอดภัยของฮาร์ดแวร์ กลไกการควบคุมสิทธิในการเข้าถึง ระบบการพิสูจน์ตัวตน ระบบการปกป้องข้อมูล การแบ่งพาร์ติชันหน่วยความจำ หรือโปรแกรมการเข้ารหัสลับ การกำหนด การออกแบบซอฟต์แวร์ซึ่งประกอบด้วยข้อมูลการกำหนดค่าเริ่มต้น รหัสผ่าน PIN หรือคีย์การเข้ารหัสลับอาจยังเป็นเป้าหมาย	ตัวประมวลผลอิเล็กทรอนิกส์รวมเข้ากับชุดของเซนเซอร์ การตรวจพบการชักจูงที่แอ็คทีฟที่มีความซับซ้อน หรือกลไกการตอบกลับ อุณหภูมิ สูงและต่ำ ระดับของแรงดันไฟ และการจัดลำดับ การแผ่รังสี และเซนเซอร์ การโจมตีแบบฟิสิคัลถูกออกแบบเพื่อปกป้องสถานการณ์เชิงสภาพแวดล้อม ที่ผิดปกติ ข้อมูลอิเล็กทรอนิกส์ที่สำคัญทั้งหมดถูกล้อมรอบอยู่ใน แพ็กเกจที่ถูกห่อหุ้มแบบฟิสิคัล ขึ้นอยู่กับการตรวจพบเหตุการณ์การชักจูงที่อาจเป็นไปได้ ตัวประมวลผลร่วมลางข้อมูลหน่วยความจำ RAM ภายในทั้งหมดโดยทันที ซึ่งยัง zeroize คีย์ที่ถูกใช้เพื่อกู้คืนข้อมูลที่สำคัญซึ่งเป็นข้อมูลที่มีอยู่จาก หน่วยความจำแฟลช เครื่องควบคุมอิสระยังถูกรีเซ็ต ซึ่งบ่งชี้ว่า ตัวประมวลผลร่วมไม่มีอยู่ในเงื่อนไขที่ได้รับการรับรอง จากโรงงาน เซนเซอร์ การชักจูงต่างๆ เกิดขึ้นจากเวลาของผู้ผลิตตัวประมวลผลร่วม ผ่านจุดสิ้นสุดของช่วงอายุการใช้งานของตัวประมวลผลร่วม ตัวประมวลผลร่วมลงนามการตอบกลับเคียวรีแบบดิจิทัลซึ่งคุณสามารถตรวจสอบเพื่อยืนยันว่าตัวประมวลผลร่วมนั้นเป็นตัวประมวลผลรวมจริงและไม่ได้ถูกชักจูง เกือบทั้งหมด ของซอฟต์แวร์ที่รันอยู่บนตัวประมวลผลหลักภายในตัวประมวลผลร่วม จะมีอยู่บนเว็บและเกี่ยวข้องกับวิศวกรรมย้อนทาง อย่างไรก็ตาม ตัวประมวลผลร่วมตรวจสอบความถูกต้องของลายเซ็นแบบดิจิทัลบนโค้ด ซึ่งร้องขอให้ยอมรับโค้ดที่แก้ไขโดยฝ่ายตรงข้าม ที่ไม่สามารถโหลดลงในตัวประมวลผลร่วม พับลิกคีย์ถูกใช้เพื่อตรวจสอบว่า โค้ดที่นำเสนอนั้นถูกทำลายลงเมื่อเหตุการณ์ที่ชักจูงถูกจดจำไว้ การออกแบบ และการนำไปปฏิบัติถูกประเมินผลอย่างเป็นอิสระ และรับรองโดย USA NIST ภายใต้ FIPS PUB 140-2 ระดับ 4 แบบมาตรฐาน **หมายเหตุ:** คุณ ต้องตรวจสอบความถูกต้องของเงื่อนไขของตัวประมวลผลร่วมและเนื้อหาโค้ด
การแก้ไขแบบฟิสิคัลของตัวประมวลผลร่วม ฝ่ายตรงข้าม อาจแก้ไขตัวประมวลผลร่วมแบบฟิสิคัลเพื่อแสดงข้อมูล การออกแบบหรือข้อมูลที่เกี่ยวข้องกับความปลอดภัย การแก้ไขนี้ อาจบรรลุได้โดยผ่านเทคนิคทั่วไปที่ใช้ในการวิเคราะห์ความขัดข้องของฮาร์ดแวร์ และการสนับสนุนวิศวกรรมย้อนทาง เป้าหมายคือ ระบุรายละเอียดของการออกแบบตามกลไกด้านความปลอดภัยของฮาร์ดแวร์ กลไกการควบคุมสิทธิในการเข้าถึง ระบบการพิสูจน์ตัวตน ระบบการปกป้องข้อมูล การแบ่งพาร์ติชัน หรือโปรแกรมการเข้ารหัสลับ การกำหนดการออกแบบซอฟต์แวร์ ซึ่งประกอบด้วยข้อมูลการกำหนดค่าเริ่มต้น รหัสผ่าน หรือ คีย์การเข้ารหัสลับอาจยังคงเป็นเป้าหมายอยู่	ข้อมูลอิเล็กทรอนิกส์ที่สำคัญถูกทำเป็นแพ็กเกจไว้ทั้งหมด ภายในแพ็กเกจการตอบกลับที่ชักจูงซึ่งประกอบเข้ากับตัวประมวลผลร่วม ในกระบวนการของการเลือกข้อมูลอิเล็กทรอนิกส์ที่สำคัญ ใบรับรองตัวประมวลผลร่วมจากโรงงาน จะทำลายการแสดงอุปกรณ์ที่ไม่ได้ใช้งาน **หมายเหตุ:** ให้ยืนยันว่าตัวประมวลผลร่วมเฉพาะ ที่กำหนดหมายเลขลำดับแล้วใช้งานอยู่และตรวจสอบการตอบกลับสถานะของเคียวรีเพื่อยืนยันว่ายังคงมีตัวประมวลผลร่วม IBM ที่ไม่มีการเปลี่ยนแปลงซึ่งโหลดด้วยซอฟต์แวร์ที่เหมาะสม

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	**การถ่ายโอนการคุกคาม**
การจัดการเชิงสภาพแวดล้อมของตัวประมวลผลร่วม ฝ่ายตรงข้ามอาจใช้ประโยชน์จากเงื่อนไขเชิงสภาพแวดล้อมที่อยู่ใกล้กับ ข้อกำหนดคุณสมบัติตัวประมวลผลร่วมเหล่านี้ เพื่อขอรับหรือแก้ไขข้อมูลหรือโฟลว์ของโปรแกรมสำหรับการใช้ตัวประมวลผลร่วมที่หลอกลวง การแก้ไขนี้อาจ ประกอบด้วยการจัดการกับสายไฟ อัตราสัญญาณนาฬิกา หรือเปิดอุณหภูมิสูง และต่ำ และการแผ่รังสี ดังนั้น ตัวประมวลผลร่วม อาจขอรับสถานการณ์ซึ่งเป็นคำสั่งที่ไม่ได้เรียกใช้อย่างถูกต้อง ตามผลลัพธ์ที่ได้ ข้อมูลความปลอดภัยที่สำคัญอาจขอรับการแก้ไขหรือการเปิดเผย การโต้แย้งกับข้อกำหนดด้านความปลอดภัยสำหรับตัวประมวลผลร่วม	ตัวประมวลผลร่วมมีเซนเซอร์เพื่อตรวจพบแรงผลักดัน เชิงสภาพแวดล้อมที่อาจชักนำให้ดำเนินการด้วยความผิดพลาด เงื่อนไขที่ผิดปกติ สามารถเป็นสาเหตุทำให้หน่วยเป็น zeroize
กระบวนการที่เข้ามาแทนที่ คำร้องขอ และการตอบกลับ ตัวประมวลผลอาจถูกสั่งไปยังการนำตัวเลือกไปใช้งาน เพื่ออนุญาตให้ฝ่ายตรงข้ามมีอิทธิผลต่อผลลัพธ์ การนำไปใช้งานสำรอง อาจถูกแทนที่ด้วยคุณลักษณะความปลอดภัยที่แตกต่างกัน ตัวอย่าง เช่น การสร้างคีย์ส่วนตัวและลายเซ็นดิจิทัลที่ใช้งานจริง อาจถูกดำเนินการในการนำไปใช้งานสำรองซึ่งจะอนุญาตให้เปิดเผย คีย์ส่วนตัว	**หมายเหตุ:** 1. ผู้ตรวจสอบจำเป็นต้องทำกระบวนการต่างๆ ให้เสร็จสมบูรณ์ที่กล่าวถึงไว้ เพื่อตรวจสอบการลงนามคีย์โดยละเอียดที่มีอยู่ภายใน ตัวประมวลผลร่วมที่เหมาะสม 2. การเข้าถึงระบบโฮสต์ควรถูกดูแล เพื่อให้การวัดความปลอดภัยของระบบโฮสต์ และการดำเนินการที่ถูกต้องสามารถเชื่อถือได้
การคุกคามที่เชื่อมโยงกับ การจู่โจมแบบโลจิคัลบนตัวประมวลผลร่วม	
การแทรกความผิดพลาด ฝ่ายตรงข้ามอาจกำหนดข้อมูล ความปลอดภัยที่สำคัญผ่านการสังเกตผลลัพธ์ของการแทรกการทำซ้ำของ ข้อมูลที่เลือกไว้ การแทรกอินพุตที่เลือกไว้ตามด้วยการมอนิเตอร์เอาต์พุตสำหรับการเปลี่ยนแปลงคือวิธีการจู่โจมที่รู้จักกันดี สำหรับอุปกรณ์การเข้ารหัสลับ ความต้องการ คือ การนิยาม ข้อมูลแบบอิงวิธีการตอบกลับ ตัวประมวลผลร่วมไปยังอินพุตที่เลือกไว้ การคุกคามนี้ถูกแบ่งแยก โดยการแลกเปลี่ยนความคิดเห็นและตัวเลือกของการทำซ้ำ และการจัดการของข้อมูลอินพุตซึ่งตรงข้ามกับการเลือกแบบสุ่มหรือการจัดการของ คุณลักษณะแบบฟิสิคัลที่เกี่ยวข้องกันในการดำเนินการอินพุตหรือเอาต์พุต	การออกแบบเชิงอิเล็กทอรนิกส์ของตัวประมวลผลที่ render วิธีการแบบคลาสลิคกับการจู่โจมสมาร์ทการ์ดที่ไม่สามารถปฏิบัติได้ **หมายเหตุ:** การดูแล ของระบบโฮสต์และการควบคุมการเข้าถึงระบบ ทั้งแบบโลจิคัล และแบบฟิสิคัลมีขั้นตอนความปลอดภัยที่สำคัญเพื่อใช้โดยองค์กร
การรีเซ็ตแบบบังคับใช้ ฝ่ายตรงข้ามอาจบังคับใช้ตัวประมวลผลร่วม ในสถานะที่ไม่มีความปลอดภัยผ่านการยกเลิกที่ไม่เหมาะสม ของการดำเนินการที่เลือกไว้ ความพยายามในการสร้างสถานะแบบไม่ปลอดภัยใน ตัวประมวลผลร่วมอาจทำผ่านการยกเลิกรายการดำเนินการก่อนกำหนด หรือการสื่อสารระหว่างตัวประมวลผลร่วมและโฮสต์โดยการแทรกของอินเตอร์รัปต์ หรือโดยการใช้ฟังก์ชันอินเตอร์เฟสที่ไม่เหมาะสม	ตัวประมวลผลร่วมถูกออกแบบมาเพื่อรันผ่าน ลำดับการเปิดเริ่มต้นในเหตุการณ์ของแทร็บและเงื่อนไขการรีเซ็ต คำร้องขอแต่ละระดับของแอ็พพลิเคชันถูกใช้เป็นหน่วยงานที่แบ่งแยก และประมวลผลจากชุดของเงื่อนไขเริ่มต้นที่นิยามไว้
อินพุตที่ไม่ถูกต้อง ฝ่ายตรงข้ามหรือผู้ที่ได้รับสิทธิ ของตัวประมวลผลร่วมอาจประนีประนอมคุณลักษณะความปลอดภัยของตัวประมวลผลร่วม ผ่านคำแนะนำของอินพุตที่ไม่ถูกต้อง อินพุตที่ไม่ถูกต้องอาจใช้ รูปแบบของการดำเนินการที่ไม่ได้อยู่ในรูปแบบที่ถูกตอง คำร้องขอสำหรับขอมูล ที่อยู่ใกล้ข้อจำกัดของการลงทะเบียน หรือความพยายามในการค้นหา และการเรียกใช้งานคำสั่งที่ไม่ได้ทำเป็นเอกสารไว้ ผลลัพธ์ของการจู่โจมอาจถูกประนีประนอม ในฟังก์ชันความปลอดภัย การสร้างข้อผิดพลาดที่สามารถนำมาใช้ประโยชน์ในการดำเนินการหรือรีลีสข้อมูลที่ปกป้องไว้	คำร้องขอการดำเนินรายการใช้ข้อมูลการพิสูจน์ตัวตน ที่ใช้ในโดเมนของตัวเรียกและการตรวจสอบความถูกต้องโดยตัวประมวลผลร่วม แต่ละคำร้องขอถูกประมวลผลจากสถานที่รู้จักซึ่งเป็นสถานะเดี่ยวด้วยเงื่อนไขที่กำหนดไว้ ลวงหน้า ซอฟต์แวร์ตัวประมวลผลร่วมตรวจสอบคุณลักษณะของคำร้องขอแต่ละรายการ เพื่อแสดงสถานการณ์จำลองที่ใช้งานผิด

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	**การถ่ายโอนการคุกคาม**
การโหลดข้อมูลที่ทำงานผิดหน้าที่ ฝ่ายตรงข้ามอาจสร้าง ข้อผิดพลาดที่มีความประสงค์ร้ายในการติดตั้งข้อมูลเพื่อประนีประนอมกับฟังก์ชันความปลอดภัย ของตัวประมวลผลร่วม ในระหว่างขั้นตอนของการเตรียมตัวประมวลผลร่วม ซึ่งเกี่ยวข้องกับการโหลดตัวประมวลผลร่วมด้วยคีย์พิเศษ identification ของบทบาท และอื่นๆ ข้อมูลอาจถูกเปลี่ยนแปลงจากข้อมูลที่มีเจตนา หรืออาจล้มเหลว เหตุการณ์สามารถพยายาม สอดแทรกเข้าสู่ฟังก์ชันความปลอดภัยของตัวประมวลผลร่วม หรือเปิดเผยความปลอดภัยด้วยวิธีการที่ไม่ได้รับสิทธิ	**หมายเหตุ:** เนื่องจากโครงร่างในโพรซีเดอร์ของผู้ตรวจสอบ การตั้งค่าการควบคุมสิทธิในการเข้าถึงควรถูกตรวจสอบพร้อมกับการยืนยัน ที่ติดตั้งซอฟต์แวร์ตัวประมวลผลร่วม
การโหลดโปรแกรมที่ไม่ได้รับสิทธิ ฝ่ายตรงข้าม อาจใช้ประโยชน์จากโปรแกรมที่ไม่ได้รับสิทธิเพื่อสอดแทรกหรือแก้ไขฟังก์ชันความปลอดภัย ของตัวประมวลผลร่วม โปรแกรมที่ไม่ได้ให้สิทธิอาจรวมถึงการเรียกใช้โปรแกรม ที่ถูกต้องแต่ไม่อนุญาตให้ใช้งานระหว่างการทำงานปกติ หรือการโหลดที่ไม่ได้รับอนุญาตของโปรแกรมที่เป็นเป้าหมายพิเศษ เมื่อสอดแทรก หรือแก้ไขฟังก์ชันความปลอดภัย	ตัวประมวลผลร่วมยอมรับซอฟต์แวร์ที่ลงนาม แบบดิจิทัลหลังจากตรวจสอบลายเซ็นแล้ว การประเมินผลที่เป็นอิสระของ ซอฟต์แวร์ของ IBM จะ build และลงนามโพรซีเดอร์และการออกแบบตัวประมวลผลร่วม จะยืนยันความไว้วางใจที่สามารถวางอยู่ในซอฟต์แวร์ที่โหลด ซึ่งเป็นเอกลักษณ์ **หมายเหตุ:** ผู้ตรวจสอบควรทำตามโพรซีเดอร์เพื่อยืนยันว่า ซอฟต์แวร์ที่ระบุเฉพาะใช้งานอยู่
การคุกคามที่เชื่อมโยงกับ การควบคุมสิทธิในการเข้าถึง	
การเข้าถึงที่ไม่ถูกต้อง ผู้ใช้หรือฝ่ายตรงข้าม ของตัวประมวลผลร่วมอาจเข้าถึงข้อมูลหรือเซอร์วิสที่ไม่มีสิทธิ ตามนิยามในโปรไฟล์บทบาท แต่ละบทบาทมีนิยามที่มีสิทธิพิเศษซึ่งอนุญาตให้เข้าถึงได้เฉพาะกับเซอร์วิสที่เลือกไว้ของตัวประมวลผลร่วม การเข้าถึง ที่อยู่ใกล้กับเซอร์วิสที่ระบุเฉพาะเหล่านี้สามารถส่งผลทำให้เกิดการเปิดเผยของ ข้อมูลที่มีความปลอดภัย	ผู้ตรวจสอบสามารถยืนยันสิทธิที่ได้รับในบทบาทที่สร้างขึ้น และชุดของโปรไฟล์ผู้ใช้ ที่เชื่อมโยงกับ บทบาทแต่ละบทบาท การประเมินผลที่เป็นอิสระของการนำไปใช้งานของซอฟต์แวร์ตัวประมวลผลร่วม และการทดสอบที่ได้ตรวจทานความสมบูรณ์ของการนำการควบคุมสิทธิในการเข้าถึง ไปปฏิบัติ
การหลอกลวงสำหรับการใช้ในครั้งแรก ฝ่ายตรงข้าม อาจได้รับการเข้าถึงข้อมูลตัวประมวลผลร่วมโดยใช้การเข้าถึงใหม่ที่ไม่ได้รับอนุญาต และตัวประมวลผลร่วมที่ติดตั้งไว้แล้ว ฝ่ายตรงข้ามอาจลองขอรับการเข้าถึง ตัวประมวลผลร่วมในระหว่างหรือหลังจากกระบวนการผลิต และโหลดซอฟต์แวร์ที่มีการหลอกลวงในตัวประมวลผลร่วมหรือแก้ไขข้อมูลที่สำคัญ ซึ่งเก็บอยู่ภายในตัวประมวลผลร่วมในระหว่างการผลิตและกระบวนการกำหนดค่าเริ่มต้นก่อนที่จะจัดส่งไปยังลูกค้า	การผลิตและการฝึกปฏิบัติในการแจกจ่ายของ IBM เพื่อตรวจสอบว่า ก่อนหน้าที่ใบรับรองจากโรงงานที่ผู้ใช้ขั้นปลายของตัวประมวลผลร่วม ไม่เป็นที่รู้จักและไม่ถูกกำหนดไว้ ซอฟต์แวร์ที่ติดตั้งจากโรงงาน ถูกตรวจสอบผ่านการตรวจสอบของลายเซ็นแบบดิจิทัล **หมายเหตุ:** 1. การติดตั้งแบบมาตรฐานที่สร้างกระบวนการแทนที่ ซอฟต์แวร์ตัวประมวลผลร่วมแบบรันไทม์ทั้งหมด 2. คุณควรตรวจสอบว่า เซ็กเมนต์ที่ 2 และ 3 ไม่ได้เป็นเจ้าของ ก่อนที่จะโหลดซอฟต์แวร์ตัวประมวลผลร่วมสำหรับการใช้งานจริง การดำเนินการนี้ทำให้แน่ใจว่า ข้อมูลที่เหลืออยู่นำมาสู่การดำเนินการตามลำดับขั้นตอน

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	การถ่ายโอนการคุกคาม
การปลอมตัว ฝ่ายตรงข้ามอาจได้รับการเข้าถึง ข้อมูลตัวประมวลผลร่วมหรือเซอร์วิสโดยปลอมตัวผู้ใช้ที่ได้รับสิทธิ ของตัวประมวลผลร่วม ตัวประมวลผลร่วมจำเป็นต่อการนิยามบทบาท รวมถึงกลไกการพิสูจน์ตัวตนที่ต้องการและเซอร์วิส ของบทบาทที่อนุญาตให้ใช้งาน ฝ่ายตรงข้ามอาจพยายามหลอกลวง ผู้ใช้ที่ได้รับสิทธิ เพื่อทำงานภายในบทบาทที่นิยามไว้ เพื่อขอรับสิทธิในการเข้าถึง ข้อมูลหรือดำเนินการกับเซอร์วิสที่อนุญาตไว้สำหรับผู้ใช้ที่ได้รับสิทธิ	สองคลาสผู้ใช้ได้แก่: 1. (IBM) ตัวลงนามโค้ดตัวประมวลผลร่วม: การประเมินค่าของ โพรซีเดอร์ของ IBM อย่างเป็นอิสระสำหรับการสร้างและการลงนามโค้ด ทำให้แน่ใจได้ว่า โค้ดที่ถูกต้องสามารถระบุได้โดยผู้ตรวจสอบ ของผู้ใช้ 2. การออกแบบการควบคุมสิทธิในการเข้าถึง CCA จะปกป้องความสมบูรณ์และ การรักษาความลับของ passphrase การควบคุมสิทธิในการเข้าถึงของผู้ใช้จากโดเมนของ กระบวนการสำหรับผู้ใช้ในตัวประมวลผลร่วม passphrase และ identification โปรไฟล์ที่ถูกต้องให้สิทธิในการใช้บทบาท **หมายเหตุ:** ความปลอดภัยของระบบโฮสต์ การออกแบบแอ็พพลิเคชันของระบบโฮสต์ และนโยบายการควบคุมดูแลจำเป็นต้องมีเพื่อให้แน่ใจว่า passphrase ของผู้ใช้ที่ได้กำหนดไว้มีความปลอดภัย
การคุกคามที่เชื่อมโยงกับ การโต้ตอบที่ไม่ได้คาดการณ์ไว้	
การใช้ฟังก์ชันของแอ็พพลิเคชันที่ไม่ได้รับอนุญาต ฝ่ายตรงข้าม อาจหาใช้ประโยชน์จากการโต้ตอบระหว่างแอ็พพลิเคชัน เพื่อแสดงตัวประมวลผลร่วมที่สำคัญหรือข้อมูลผู้ใช้ การโต้ตอบอาจรวมถึงการเรียกใช้คำสั่ง ที่ไม่ต้องการหรืออนุญาตให้ใช้ในแอ็พพลิเคชันที่ระบุเฉพาะซึ่งกำลังดำเนินการอยู่ ตัวอย่างประกอบด้วยการใช้ฟังก์ชันที่เกี่ยวของกับการจัดการกับคีย์หลัก หรือฟังก์ชันที่เกี่ยวข้องกับการเข้ารหัสแบบสมมาตรหรือเซอร์วิสทางด้านการเงิน ฟังก์ชันเหล่านี้ไม่มีผลกระทบทางด้านลบ บนฟังก์ชันของตัวประมวลผลร่วม ที่จำเป็นสำหรับ แอ็พพลิเคชันการลงนามแบบดิจิทัล	การออกแบบตัวประมวลผลร่วมต้องการให้คุณตั้งค่า การติดตั้งการควบคุมสิทธิในการเข้าถึง ซอฟต์แวร์ CCA ได้ถูกตรวจสอบเพื่อทำให้มั่นใจว่าฟังก์ชันไม่ได้รับอนุญาตให้ใช้เมื่อไม่ได้เปิดใช้งานคำสั่งที่จำเป็นต้องมี **หมายเหตุ:** 1. คอนฟิกูเรชันการควบคุมสิทธิในการเข้าถึงของคุณควรทำตามหลักการที่กล่าวอยู่ในภาคผนวก H ของคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* Redbooks ดังนั้นเฉพาะฟังก์ชันที่จำเป็นสำหรับขั้นตอนดำเนินการ ที่สามารถเรียกได้ในขั้นตอนนี้ 2. สำหรับแอ็พพลิเคชันการลงนามแบบดิจิทัล ให้สร้างคำแนะนำสำหรับชุดของบทบาทที่มีความสามารถที่จำกัด และลำดับการติดตั้งที่จำกัดการทำงานตัวประมวลผลร่วมที่จำเป็นที่สุดสำหรับ การลงนามแบบดิจิทัล ในบางการติดตั้ง อาจต้องการวิธีการที่แตกต่างกันกับบทบาท หรือพิจารณาฟังก์ชันของแอ็พพลิเคชันเพิ่มเติม หรือทั้งสองอย่าง ในกรณีเหล่านี้ ตรวจสอบว่า คุณตรวจทานคำแนะนำและการสังเกตในภาคผนวก H ของคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe และ 4764 PCI-X Cryptographic Coprocessors* Redbooks สำหรับความสามารถในการใช้งานกับสถานการณ์ของคุณ
การคุกคามที่เกี่ยวข้องกับ ฟังก์ชันการเข้ารหัสลับ	

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	**การถ่ายโอนการคุกคาม**
การจู่โจมการเข้ารหัสลับ ฝ่ายตรงข้ามอาจพ่ายแพ้ ต่อฟังก์ชันความปลอดภัยผ่านการจู่โจมการเข้ารหัสกับอัลกอริธึม หรือผ่านการโจมตีแบบออกแรงทำงานเพียงอย่างเดียว การจู่โจมนี้อาจรวมถึง การสร้างลายเซ็นและฟังก์ชันการตรวจสอบหรือตัวสร้างหมายเลขแบบสุ่ม อย่างใดอย่างหนึ่ง	ตัวประมวลผลร่วมนำฟังก์ชันการเข้ารหัสลับ ที่สร้างขึ้นไว้และเป็นมาตรฐาน การนำการสร้างหมายเลขแบบสุ่มมาใช้งาน เกี่ยวข้องกับการประเมินผลที่ขยายเพิ่มภายใต้เงื่อนไขของการเผยแพร่โดย USA NIST และ German Information Security Agency (German Bundesamt fur f³r Sicherhert in der Informations Technik หรือ German BSI) ความลับที่สามารถหาได้ซึ่งมีคีย์อยู่เกี่ยวข้อง กับการประเมินผลที่เป็นอิสระ การออกแบบและขั้นตอนการนำมาใช้เหล่านี้ จัดเตรียมการรับประกันกับการจู่โจมการเข้ารหัสลับ **หมายเหตุ:** สำหรับ เซิร์ฟเวอร์การลายเซ็นดิจิทัล ดูที่คำแนะนำในภาคผนวก H ของคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* Redbooks
การคุกคามที่เกี่ยวข้องกับ ลายเซ็นแบบดิจิทัล	
การปลอมแปลงข้อมูลที่ลงนามแล้ว ฝ่ายตรงข้าม อาจแก้ไขข้อมูลที่ลงนามแล้วแบบดิจิทัลโดยตัวประมวลผลร่วม ดังนั้น การแก้ไขนี้ไม่สามารถตรวจพบได้โดยผู้ลงนามในสัญญาหรือกลุ่มบุคคลที่สาม การโจมตี อาจใช้จุดอ่อนในฟังก์ชันการแฮชที่ป้องกันความปลอดภัย จุดอ่อนในการเข้ารหัสการลงนาม หรือจุดอ่อนในอัลกอริธึมการเข้ารหัสที่ใช้ เพื่อสร้างการลงนามที่ถูกปลอมแปลง	ตัวประมวลผลร่วมนำฟังก์ชันการเข้ารหัสลับ ที่สร้างขึ้นไว้และเป็นมาตรฐาน **หมายเหตุ:** 1. ข้อควรระวังในการใช้ CCA ซึ่งควรนำมาพิจารณาตามที่จัดทำเอกสารไว้ในภาคผนวก H ของคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* Redbooks 2. ผู้ใช้ควรรักษาการรับรู้ของความอ่อนแอที่กล่าวถึง ในฟอรัม (เปิด) เกี่ยวข้องกับจุดแข็งของอัลกอริธึมการเข้ารหัสลับ และกระบวนการที่ใช้
การปลอมแปลงข้อมูลก่อนที่จะถูกลงนาม ฝ่ายตรงข้าม อาจแก้ไขข้อมูลที่ต้องถูกลงนามโดยตัวประมวลผลร่วม ก่อนลายเซ็นจะถูกสร้างภายในตัวประมวลผลร่วม การจู่โจมนี้อาจใช้จุดอ่อน ในการนำไปปฏิบัติที่อนุญาตให้ฝ่ายตรงข้ามเพื่อแก้ไข ข้อมูลที่ส่งผ่านลายเซ็นไปยังตัวประมวลผลร่วมก่อนที่ตัวประมวลผลร่วม คำนวณลายเซ็น	คำร้องขอจากผู้ใช้หน่วยความจำในการประมวลผลโฮสต์แอ็พพลิเคชัน ใช้ค่าการตรวจสอบความสมบูรณ์ที่ตัวประมวลผลร่วมยืนยันก่อนที่จะ รวมเข้าด้วยกันกับการแฮชในลายเซ็นแบบดิจิทัล **หมายเหตุ:** ผู้ใช้ ต้องตรวจสอบความปลอดภัยของโปรแกรมระบบโฮสต์และแอ็พพลิเคชันโฮสต์ เพื่อตรวจสอบให้มั่นใจว่า ค่าการแฮชที่พิสูจน์ตัวตนแล้วซึ่งรับลงในตัวประมวลผลร่วม ไม่ได้ถูกยินยอมและเป็นการแทนที่ข้อมูลที่ต้องการปกป้อง
การใช้ฟังก์ชันลายเซ็นที่ไม่ถูกต้อง ผู้ไม่หวังดี อาจใช้การสร้างลายเซ็นตัวประมวลผลร่วม เพื่อลงนามข้อมูล ที่ตัวประมวลผลร่วมไม่สนับสนุนให้ลงลายเซ็น ฝ่ายตรงข้ามอาจพยายามส่งข้อมูล ไปที่ตัวประมวลผลร่วมและขอรับการลงนาม โดยไม่ต้องส่งผ่านการตรวจสอบการพิสูจน์ตัวตน ของตัวประมวลผลร่วมซึ่งดำเนินการก่อนที่จะสร้าง ลายเซ็นแบบดิจิทัล เนื่องจากเป็นตัวสำรอง ฝ่ายตรงข้ามอาจพยายามแก้ไขข้อมูลภายในตัวประมวลผลร่วมผ่านการใช้ฟังก์ชัน ตัวประมวลผลร่วมหรือโดยพยายามมีอิทธิพลต่อตัวประมวลผลร่วม ดังนั้น ข้อมูลในตัวประมวลผลร่วมจะขอรับข้อมูลที่แก้ไขแล้ว	การตรวจทานที่เป็นอิสระของซอฟต์แวร์ตัวประมวลผลร่วม ถูกคาดการณ์เพื่อยืนยันว่า: • เซอร์วิสการสร้างลายเซ็นดิจิทัลต้องการสิทธิ ที่เหมาะสมในบทบาท • การประมวลผลคำร้องขอและความสมบูรณ์ของการออกแบบปกป้องการเปลี่ยนแปลงข้อมูล **หมายเหตุ:** 1. ความสมบูรณ์ของตัวประมวลผลร่วมและโค้ดต้องถูกยืนยันโดย ผู้ตรวจสอบซึ่งเป็นผู้ที่ตรวจทานเคียวรีสถานะของตัวประมวลผลร่วม 2. ผู้ตรวจสอบต้องยืนยันว่าบทบาทการควบคุมการเข้าถึงและโปรไฟล์ที่เหมาะสม ได้ถูกสร้างขึ้นซึ่งแยกผู้ใช้ที่ไม่ได้รับอนุญาต ไม่ให้ใช้ฟังก์ชันการสร้างลายเซ็นแบบดิจิทัล

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	การถ่ายโอนการคุกคาม
การปลอมแปลงฟังก์ชันการตรวจสอบลายเซ็น ฝ่ายตรงข้าม อาจแก้ไขฟังก์ชันสำหรับการตรวจสอบลายเซ็น เช่น การลงนามผิดถูกยอมรับว่าถูกต้อง การโจมตีนี้อาจพยายาม แก้ไขฟังก์ชันการตรวจสอบความถูกต้องของลายเซ็นหรือข้อมูลที่ลงนามเพื่อตรวจสอบ ตัวประมวลผลร่วมที่ส่งคืนข้อความสำเร็จ เมื่อลายเซ็นที่ผิดพลาดนี้ถูกแสดงไว้สำหรับการตรวจสอบความถูกต้อง	ฟังก์ชันการตรวจสอบลายเซ็นของความสนใจหลักในที่นี้ เกิดขึ้นในกระบวนการโหลดโค้ดของตัวประมวลผล (ใน Miniboot) ด้วยผลิตภัณฑ์นี้: • โค้ด Miniboot เช่น โปรแกรมการควบคุมและโค้ดแอ็พพลิเคชันโปรแกรม (CCA) ถูกยอมรับในตัวประมวลผลร่วมเมื่อตัวประมวลผลร่วม ตรวจสอบความถูกต้องบนโค้ดที่ลงนามแล้วเท่านั้น • โค้ด Miniboot เริ่มต้นที่โหลดจากโรงงานยังเกี่ยวข้องกับ การตรวจสอบการลงนามแบบดิจิทัล • กระบวนการเข้ารหัสลับแบบมาตรฐานถูกใช้ (SHA-1, RSA, ISO 9796) สำหรับลายเซ็น • การ build โค้ดและกระบวนการลงนามเกี่ยวข้องกับการตรวจทาน ที่เป็นอิสระ
การเปิดเผยของคีย์การลงนาม RSA ส่วนบุคคล ผู้ไม่ หวังดีอาจใช้ฟังก์ชันที่เปิดเผยคีย์ลายเซ็น RSA ส่วนตัว	การประเมินผลแบบอิสระถูกคาดการณ์เพื่อยืนยันว่า ส่วนสนับสนุนโปรแกรม CCA ไม่ได้มีฟังก์ชันใดๆ ที่ต้องเอาต์พุต หรือแสดงค่าของคีย์ส่วนตัวที่มีอยู่ การประเมินค่าใบรับรอง ถูกคาดการณ์เพื่อสาธิตให้เห็นว่าโปรแกรมการควบคุมไม่ได้เอาต์พุต ข้อมูลที่มีอยู่ในหน่วยเก็บที่มีตัวประมวลผลอยู่หรือไม่ได้อยู่ในฟังก์ชันที่มีระดับต่ำกว่า เพื่ออ่านหน่วยเก็บบางส่วน
การลบคีย์ลายเซ็น RSA ส่วนบุคคล ฝ่ายตรงข้าม อาจใช้ฟังก์ชันที่ลบคีย์การลงนาม RSA ส่วนบุคคลโดยไม่มี การพิสูจน์ตัวตนที่ต้องทำ และไม่มีการชักจูงด้วย ตัวประมวลผลร่วม	การประเมินผลแบบอิสระถูกคาดการณ์เพื่อยืนยันว่า คีย์ส่วนตัวที่มีอยู่ถูกลบทิ้งแล้วเท่านั้นในสถานการณ์ต่อไปนี้: 1. ภายใต้การควบคุม CCA ด้วย Retained_Key_Delete verb 2. โดยการโหลดซอฟต์แวร์ตัวประมวลผลร่วม CCA* 3. โดยการลบซอฟต์แวร์ตัวประมวลผลร่วม CCA 4. โดยเป็นต้นเหตุของเหตุการณ์การชักจูง **หมายเหตุ:** เพื่อระบุถึงช่องโหว่ เหล่านี้ให้ดำเนินการดังนี้: 1. เลือกที่จะเปิดใช้งานคำสั่ง ลบคีย์ที่มีอยู่ X'0203' 2. ใช้การควบคุมการเข้าถึงระบบโฮสต์เพื่อจัดการการใช้ CLU 3. จัดการกับการเข้าถึงแบบฟิสิคัลกับตัวประมวลผลร่วม *การรีโหลดซอฟต์แวร์ตัวประมวลผลร่วม ด้วยไฟล์ เช่น CEXxxxxx.clu ไม่ได้ zeroize เนื้อหาของหน่วยเก็บที่มีอยู่ ไฟล์ CNWxxxxx.clu จะ zeroize หน่วยเก็บที่มีอยู่ โปรดดู "การโหลดและการยกเลิกการโหลดซอฟต์แวร์เข้าสู่ตัวประมวลผลร่วม" ในหน้า 8
การคุกคามที่มอนิเตอร์ ข้อมูล	
การรั่วของข้อมูล ผู้ไม่หวังดีอาจใช้ข้อมูล ที่รั่วไหลมาจากตัวประมวลผลร่วมระหว่างการใช้งานตาม ปกติ การรั่วไหลของข้อมูลอาจเกิดขึ้นได้ผ่านจุดกำเนิด ซึ่งคือการเปลี่ยนแปลงการใช้กำลังไฟ คุณสมบัติของ I/O ความถี่ของสัญญาณนาฬิกา หรือการเปลี่ยนแปลงข้อกำหนด เกี่ยวกับเวลาในการประมวลผล การรั่วนี้อาจถูกตีความเป็นการแปลง ช่องสัญญาณการส่งผ่านข้อมูล แต่โดยส่วนใหญ่จะเกี่ยวข้องกับการวัดพารามิเตอร์การทำงาน ซึ่งอาจได้รับจากการวัดโดยตรง (การติดต่อ) หรือการวัดการส่งผ่าน และอาจเกี่ยวข้องกับการดำเนินการที่ระบุเฉพาะ ที่ต้องถูกดำเนินการ	การฝึกปฏิบัติหมายความถึงการตีความการรั่วของข้อมูล ที่เกี่ยวข้องกับการค้นหาในเชิงพาณิชย์และห้องปฏิบัติการวิจัยของรัฐบาล การคุ้มกันแบบลึกควรรวมถึงข้อจำกัดในการเข้าถึงสภาพแวดล้อมการเข้ารหัสลับ และข้อจำกัดเกี่ยวกับการใช้อุปกรณ์พิเศษ และ การอยู่ใกล้กับสภาพแวดล้อมการเข้ารหัสลับ

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	**การถ่ายโอนการคุกคาม**
การลิงก์ข้อสังเกตจำนวนมาก ฝ่ายตรงข้าม อาจสังเกตการใช้งานรีซอร์สหรือเซอร์วิสจำนวนมากและ โดยการลิงก์การสังเกตเหล่านี้ ซึ่งได้ข้อมูลสรุปที่จะแสดงข้อมูล ความปลอดภัยที่สำคัญ ชุดของข้อสังเกตที่อยู่เหนือชวงระยะเวลา ของการใช้ตัวประมวลผลร่วมจำนวนมาก หรือการรวมกันของความรู้ ที่ได้รับจากการสังเกตเห็นความแตกต่างในการดำเนินการอาจแสดงข้อมูล ที่อนุญาตให้ฝ่ายตรงข้ามเรียนรู้ข้อมูลได้โดยตรง หรือคำนวณการโจมตีที่สามารถแสดงข้อมูลเพิ่มเติมซึ่งตัวประมวลผลร่วม ต้องการเก็บไว้เป็นความลับ	**หมายเหตุ:** 1. การใช้งานของอุปกรณ์การเข้ารหัสลับควรถูกควบคุมไว้ ซึ่งประกอบด้วยคำแนะนำต่อไปนี้ในภาคผนวก H ของคู่มือ *IBM CCA Basic Services Reference and Guide for the IBM 4765 PCIe and 4764 PCI-X Cryptographic Coprocessors* Redbooks 2. ฝ่ายตรงข้ามอาจเข้าถึงข้อมูลและลายเซ็นที่ลงนามแล้ว ดังนั้น การควบคุมควรวางอยู่ในตำแหน่งที่จำกัดความสามารถของผู้ใช้ เพื่อส่งการลงนามคำร้องของโดยพละการ 3. การใช้โพรซีเดอร์การเข้ารหัสลับแบบมาตรฐานและการมอนิเตอร์ การทำความเข้าใจกับความอ่อนแอของกระบวนการเหล่านี้ของ community การเข้ารหัสร่วม (SHA-1, RSA, ISO 9796, X9.31, HMAC และ triple-DES) สามารถจัดเตรียมความเชื่อมั่นของการดำเนินการที่ได้รับความปลอดภัย
การคุกคาม อื่นๆ	
การโจมตีที่ถูกลิงก์ ฝ่ายตรงข้ามอาจ ดำเนินการโจมตีได้เป็นผลสำเร็จด้วยผลลัพธ์ที่ตัวประมวลผลร่วมมีสถานะไม่คงที่ หรือฟังก์ชันด้านความปลอดภัยลดระดับลง การโจมตีต่อไปนี้ อาจถูกเรียกใช้งานได้เป็นผลสำเร็จ การมอนิเตอร์เอาต์พุต ขณะที่จัดการกับอินพุตในสภาพแวดล้อมที่มีแรงผลักดัน คือตัวอย่างของการจู่โจมแบบลิงก์	**หมายเหตุ:** 1. การใช้ระบบการเข้ารหัสลับควรจำกัดสถานการณ์ที่ได้รับสิทธิ ซึ่งบังคับใช้ผ่านการควบคุมการเข้าถึงตัวประมวลผลร่วม และผ่านการใช้การควบคุมระบบโฮสต์ 2. การควบคุมระบบโฮสต์และนโยบายเชิงจัดการควรจำกัด การเข้าถึงระบบสำหรับการมอนิเตอร์และการส่งคำร้องขอ โดยพละการ
การโจมตีแบบซ้ำๆ ฝ่ายตรงข้ามอาจ ใช้ประโยชน์จากความพยายามที่ไม่ได้ปกป้องแบบซ้ำๆ ที่ปลอมแปลงเพื่อเปิดเผย เนื้อหาของหน่วยความจำหรือเปลี่ยนอิลิเมนต์ความปลอดภัยที่สำคัญในตัวประมวลผลร่วม ความพยายามในการทำซ้ำที่เกี่ยวข้องกับการคุกคามอื่นๆ ทั้งหมดที่กล่าวถึงในที่นี่ อาจใช้เพื่อพัฒนาการปลอมแปลงของความปลอดภัยของตัวประมวลผลร่วม ให้มีประสิทธิภาพ หากการโจมตีเหล่านี้สามารถคงความไม่ปกป้อง ในทุกกรณี จะไม่มีคำเตือนถึงความอ่อนแอที่เพิ่มขึ้น	**หมายเหตุ:** การใช้ระบบการเข้ารหัสลับควรจำกัดสถานการณ์ที่ได้รับสิทธิ ซึ่งบังคับใช้ผ่านการควบคุมการเข้าถึงตัวประมวลผลร่วม และผ่านการใช้การควบคุมระบบโฮสต์การควบคุมระบบโฮสต์และนโยบายเชิงจัดการควรจำกัดการเข้าถึงระบบสำหรับการมอนิเตอร์และการส่งคำร้องขอ โดยพละการ
การโคลน ฝ่ายตรงข้ามอาจโคลนส่วน หรือตัวประมวลผลร่วมเชิงฟังก์ชันทั้งหมดเพื่อพัฒนาการโจมตีเพิ่มเติม ข้อมูลที่จำเป็น ต่อการโคลนส่วนต่างๆ หรือตัวประมวลผลร่วมทั้งหมดได้เป็นผลสำเร็จ อาจได้รับมาจากการตรวจสอบโดยละเอียดของตัวประมวลผลร่วมเอง หรือจากข้อมูลการออกแบบที่เป็นของเถื่อน	**หมายเหตุ:** ผู้ตรวจสอบต้องยืนยันว่า คีย์ลายเซ็นดิจิทัล โค้ดที่เหมาะสม และเกณฑ์การควบคุมสิทธิในการเข้าถึงอยู่ใน ตัวประมวลผลร่วมที่ได้รับอนุญาต
การคุกคามที่แสดงโดย สภาพแวดล้อมการทำงาน	
การแก้ไขตัวประมวลผลร่วมและการนำกลับมาใช้ใหม่ ฝ่ายตรงข้าม อาจใช้ตัวประมวลผลร่วมที่แก้ไขแล้วเพื่อหลอกตัวประมวลผลร่วมเดิม ดังนั้น ข้อมูลสินทรัพย์สามารถเข้าถึงได้ การลบ การแก้ไข และการใส่ตัวประมวลผลร่วมอีกครั้งลงในระบบโฮสต์ สามารถนำมาใช้เพื่อส่งผ่านชุดข้อมูลที่เป็นต้นฉบับ ซึ่งอาจใช้ เพื่อเข้าถึงหรือเปลี่ยนคีย์ลายเซ็นส่วนบุคคล หรือข้อมูลความปลอดภัยที่สำคัญที่ต้องได้รับการปกป้อง	**หมายเหตุ:** 1. ผู้ตรวจสอบต้องยืนยันผ่านการตรวจสอบของการตอบกลับเคียวรีที่ลงนาม ตัวประมวลผลร่วมแล้ว ซึ่งอุปกรณ์นั้นต้องเป็นอุปกรณ์ของจริงและต้องโหลด โค้ดที่เหมาะสม 2. ผู้ตรวจสอบยังต้องยืนยันว่า คีย์ลายเซ็นดิจิทัล เป็นคีย์ที่เก็บไว้ในตัวประมวลผลร่วม

ตารางที่ 14. ข้อควรพิจารณาเกี่ยวกับการคุกคาม สำหรับเซิร์ฟเวอร์การลงนามแบบดิจิทัล (ต่อ)

การอภิปรายเกี่ยวกับการคุกคาม	การถ่ายโอนการคุกคาม
การใช้งานผิดโดยผู้ใช้ที่ได้รับสิทธิพิเศษ ผู้ดูแลระบบที่เล่นเล่อ จงใจ หรือละเลย หรือผู้ใช้ที่มีสิทธิพิเศษอื่นๆ อาจสร้างการประนีประนอมของสินทรัพย์ตัวประมวลผลร่วมผ่านการประมวลผลการดำเนินการที่เปิดเผยฟังก์ชันความปลอดภัยหรือข้อมูลที่ได้รับการปกป้อง ผู้ใช้ที่มีสิทธิพิเศษหรือผู้ดูแลระบบสามารถนำการโจมตีหรืออำนวยความสะดวก ในการโจมตีโดยอ้างอิงตามการคุกคามใดๆ ที่กล่าวถึงในที่นี้	**หมายเหตุ:** องค์กรต้องสร้าง บังคับใช้ และตรวจสอบนโยบายที่จำกัดการเข้าถึงที่บุคคลแต่ละราย เข้าสู่ระบบการเข้ารหัสลับ โพรซีเดอร์การติดตั้งต้องทำให้มั่นใจว่า ผู้ใช้เดี่ยวไม่มีโอกาสที่จะนำระบบที่ไม่เหมาะสม เข้าสู่ระบบที่ใช้งานจริง
การแก้ไขข้อมูล ข้อมูลที่ลงนามโดยตัวประมวลผลร่วม อาจถูกแก้ไขโดยฝ่ายตรงข้ามหรือตามค่าดีฟอลต์ในสภาพแวดล้อมการทำงาน ก่อนที่จะถูกอนุมัติโดยผู้ใช้ที่มีสิทธิ แต่ก่อนที่ข้อมูลจะถูกส่งไปยังตัวประมวลผลร่วมที่ต้องลงนาม ข้อมูลที่อนุมัติแล้วโดยผู้ใช้ที่มีสิทธิที่ต้องลงนาม อาจแก้ไขโดยฝ่ายตรงข้ามซึ่งผิดหรือเป็นความประสงค์ร้ายของโปรแกรม หรือเป็นข้อผิดพลาดทางสภาพแวดล้อม (เช่น ข้อผิดพลาดในการส่งข้อมูล) หลังจากที่ข้อมูลได้รับการอนุมัติแล้วโดยผู้ใช้ที่มีสิทธิ และก่อนที่ข้อมูลจะถูกส่งผ่านไปยังตัวประมวลผลที่ต้องลงนาม	**หมายเหตุ:** การป้องกันความปลอดภัยระบบโฮสต์และนโยบายการจัดการต้องถูกนิยาม บังคับใช้ และตรวจสอบเพื่อขัดขวางการจู่โจม
การตรวจสอบความถูกต้องของข้อมูล ข้อมูลที่ลงนามซึ่งต้องถูกตรวจสอบ โดยตัวประมวลผลร่วมอาจถูกแก้ไขโดยฝ่ายตรงข้ามหรือตามค่าดีฟอลต์ ในสภาพแวดล้อมการทำงานก่อนที่จะถูกส่งไปยังตัวประมวลผลร่วม เพื่อตรวจสอบลายเซ็น ดังนั้น การตอบกลับของตัวประมวลผลร่วม จะไม่มีผลต่อความถูกต้องของลายเซ็น ข้อมูลที่ลงนามแล้วซึ่งส่งโดยผู้ใช้ อาจถูกแก้ไขภายในสภาพแวดล้อมของตัวประมวลผลร่วมก่อนที่จะถูกส่งผ่านไปยังตัวประมวลผลร่วมเพื่อตรวจสอบความถูกต้อง ซึ่งอาจส่งผล ทำให้ตอบกลับจากตัวประมวลผลร่วมที่ไม่มีผลต่อความถูกต้องของลายเซ็นแบบดิจิทัลจริง ซึ่งควรถูกตรวจสอบ และยังมี ความเป็นไปได้ที่การตอบกลับของตัวประมวลผลร่วมถูกแก้ไข ในสภาพแวดล้อมของตัวประมวลผลร่วมก่อนที่จะถูกส่งผ่านไปยังผู้ใช้ ที่ร้องขอการตรวจสอบลายเซ็น	ตัวประมวลผลร่วมตรวจสอบลายเซ็นบนโค้ด และคำสั่งสำหรับการโหลดโค้ดบางคำสั่ง การประเมินผลอย่างเป็นอิสระ ถูกคาดการณ์ไว้เพื่อยืนยันว่า การประเมินผลนี้ไม่สามารถส่งผ่านได้ การออกแบบ CCA สนับสนุนการตรวจสอบความถูกต้องของความสมบูรณ์ของคำร้องขอและการตอบกลับระหว่าง ตัวประมวลผลร่วมและเลเยอร์บนสุดของโค้ด CCA ในระบบโฮสต์ **หมายเหตุ:** การวัดระดับของรักษาความปลอดภัยของระบบโฮสต์ ต้องกำหนดการบล็อกการแก้ไขคำร้องขออินพุต และเอาต์พุต

คำประกาศ IBM Cryptographic Coprocessor

IBM Cryptographic Coprocessor ประกอบด้วยคำประกาศ 3 ที่จัดเตรียมคำแนะนำสำหรับการทิ้งอุปกรณ์อิเล็กปลอดภัย

การรีไซเคิลและการทิ้งผลิตภัณฑ์

ยูนิตมีวัสดุ เช่น แผงวงจร สายเคเบิล ปะเก็นความเข้ากันได้กับแม่หลักไฟฟ้า และตัวเชื่อมต่อที่อาจมีตะกั่วและทองแดง/อัลลอยเบริลเลียม ที่ต้องการการจัดการพิเศษและทิ้งเมื่อหมดอายุ ก่อนที่จะทิ้งยูนิตนี้ วัสดุเหล่านี้ต้องถูกถอดออก และรีไซเคิลกหรือนำไปทิ้งตามกฎข้อบังคับที่บังคับใช้ IBM นำเสนอโปรแกรมรับคืนผลิตภัณฑ์ในหลายๆ ประเทศ ข้อมูลเกี่ยวกับการรีไซเคิลผลิตภัณฑ์ สามารถค้นหาได้ที่ไซต์อินเตอร์เน็ตของ IBM Internet ที่ http://www.ibm.com/ibm/environment/products/prp.shtml IBM ส่งเสริมให้เจ้าของอุปกรณ์เทคโนโลยีสารสนเทศ (IT) มีความรับผิดชอบต่อการรีไซเคิล อุปกรณ์ของตนเองเมื่อไม่ต้องการใช้งานอีกต่อไป IBM นำเสนอโปรแกรมและเซอร์วิสที่หลากหลายเพื่อช่วยให้เจ้าของอุปกรณ์รีไซเคิลผลิตภัณฑ์ IT ของตนเอง ข้อมูลเกี่ยวกับการรีไซเคิลผลิตภัณฑ์ สามารถค้นหาได้ที่ไซต์อินเตอร์เน็ตของ IBM:

http://www.ibm.com/ibm/environment/products/prp.shtml

คำประกาศ: เครื่องหมายนี้ใช้กับประเทศที่อยู่ในแถบยุโรป (EU) และนอร์เวย์เท่านั้น เครื่องมือได้รับการติดแถบป้ายตาม European Directive 2002/96/EC ซึ่งเกี่ยวข้องกับการกำจัดอุปกรณ์ไฟฟ้า และอุปกรณ์อิเล็กทอรนิกส์ (WEEE) แนวทางปฏิบัติจะกำหนดกรอบงานสำหรับ การรับคืนและรีไซเคิลเครื่องมือที่ใช้แล้ว ซึ่งใช้งานอยู่ในประเทศ แถบยุโรป แถบป้ายนี้นำมาใช้กับผลิตภัณฑ์ต่างๆ เพื่อบ่งชี้ว่า ไม่ควรทิ้งผลิตภัณฑ์ แต่ควรนำกลับมาเมื่อหมดอายุการใช้งานตาม Directive นี้

โปรแกรมการส่งคืนแบตเตอรี่

ผลิตภัณฑ์นี้อาจปนเปื้อนสารตะกั่ว นิกเกิลแคดเมียม นิกเกิลไฮดราย ลิเธียม หรือลิเธียมไอออนแบตเตอรี่ ศึกษาคู่มือการใช้งานหรือคู่มือการให้บริการของคุณเพื่อดูข้อมูลแบตเตอรี่เฉพาะ แบตเตอรี่ต้องถูกนำมารีไซเคิลหรือ ทิ้งอย่างถูกต้อง หน่วยงานการรีไซเคิลอาจไม่มีอยู่ในพื้นที่ของคุณ สำหรับข้อมูล เกี่ยวกับการทิ้งแบตเตอรี่นอกสหรัฐอเมริกา ให้ไปที่ http://www.ibm.com/ibm/environment/products/batteryrecycle.shtml หรือติดต่อหน่วยงานกำหนดขยะในท้องถิ่นของคุณ ในสหรัฐอเมริกา IBM จัดทำกระบวนการรับคืนสำหรับการนำกลับมาใช้ใหม่ การรีไซเคิล หรือการกำจัดแบตเตอรี่ของ IBM ที่มีสารตะกั่ว นิกเกิลแคดเมียม นิกเกิลไฮดราย หรือก้อนแบตเตอรี่อื่นๆ ที่ใช้แล้วจาก IBM Equipment สำหรับข้อมูลเกี่ยวกับการทิ้งแบตเตอรี่เหล่านี้อย่างถูกต้อง โปรดติดต่อ IBM ที่ 1-800-426-4333 โปรดเตรียมหมายเลขชิ้นส่วนของ IBM ที่แสดงอยู่บนแบตเตอรี่ก่อนที่จะ โทรหาเรา

สำหรับประเทศไต้หวัน: โปรดนำแบตเตอรี่ไปรีไซเคิล

โครงการรับคืนการ์ด IBM Cryptographic Coprocessor

เครื่องนี้อาจมีคุณลักษณะเผื่อเลือกเพิ่มเติม การ์ดตัวประมวลผลร่วมเข้ารหัสลับ ซึ่งประกอบด้วยวัสดุโพลียูรีเทน (polyurethane) ที่มีสารปรอท โปรดทำตามกฎ หรือข้อบังคับเกี่ยวกับการทิ้งการ์ดนี้ IBM ได้สร้างโครงการรับคืน สำหรับการ์ด IBM Cryptographic Coprocessor สำหรับ ข้อมูลเพิ่มเติม สามารถค้นหาได้ที่:

http:/www.ibm.com/ibm/environment/products/prp.shtml

คำประกาศ

ข้อมูลนี้ถูกพัฒนาขึ้นสำหรับผลิตภัณฑ์ และบริการที่นำเสนอในประเทศสหรัฐอเมริกา

IBM อาจไม่นำเสนอผลิตภัณฑ์ เซอร์วิส หรือคุณลักษณะที่อธิบายในเอกสารนี้ในประเทศอื่น โปรดปรึกษาตัวแทน IBM ในท้องถิ่นของคุณสำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์และการบริการที่มีอยู่ในพื้นที่ของคุณในปัจจุบัน การอ้างอิงใดๆ ถึงผลิตภัณฑ์ โปรแกรม หรือการบริการของ IBM ไม่ได้มีวัตถุประสงค์ที่จะระบุหรือตีความว่าสามารถใช้ได้เฉพาะผลิตภัณฑ์ โปรแกรม หรือการบริการของ IBM เพียงอย่างเดียวเท่านั้น ผลิตภัณฑ์ โปรแกรม หรือการบริการใดๆ ที่สามารถทำงานได้เท่าเทียมกัน และไม่ละเมิดสิทธิทรัพย์สินทางปัญญาของ IBM สามารถนำมาใช้แทนได้ อย่างไรก็ตาม เป็นความรับผิดชอบของผู้ใช้ ที่จะประเมินและตรวจสอบการดำเนินการของผลิตภัณฑ์ โปรแกรม หรือการบริการใดๆ ที่ไม่ใช่ของ IBM

IBM อาจมีสิทธิบัตร หรืออยู่ระหว่างดำเนินการขอสิทธิบัตร ที่ครอบคลุมถึงหัวข้อซึ่งอธิบายในเอกสารนี้ การตกแต่งเอกสารนี้ไม่ได้ให้สิทธิใช้งานใดๆ ในสิทธิบัตรเหล่านี้แก่คุณ คุณสามารถส่งการสอบถามเกี่ยวกับใบอนุญาตเป็นลายลักษณ์อักษร ไปที่:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

หากมีคำถามเกี่ยวกับข้อมูลไบต์คู่ (DBCS) โปรดติดต่อแผนกทรัพย์สินทางปัญญาของ IBM ในประเทศของคุณ หรือส่งคำถามเป็นลายลักษณ์อักษรไปที่ :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION จัดเตรียมเอกสาร "ตามสภาพที่เป็น" โดยไม่มีการรับประกันใดๆ ทั้งโดยชัดแจ้งหรือโดยนัย ซึ่งรวมถึง แต่ไม่จำกัดถึงการรับประกันโดยนัยที่ไม่ละเมิดความสามารถในการจัดจำหน่าย หรือตามความเหมาะสมสำหรับวัตถุประสงค์อย่างใดอย่างหนึ่ง เนื่องจากเขตอำนาจศาลบางเขตไม่อนุญาต ให้ปฏิเสธการรับประกันทางตรงหรือทางอ้อมในธุรกรรมบางอย่าง ดังนั้น ข้อความนี้จึงอาจจะไม่ใช้กับคุณ

ข้อมูลนี้อาจมีความไม่ถูกต้องทางเทคนิคหรือความผิดพลาด ทางการพิมพ์ การเปลี่ยนแปลงข้อมูลในนี้จะมีเป็นระยะๆ ซึ่งจะสอดคล้องกับ การตีพิมพ์ในครั้งใหม่ IBM อาจปรับปรุงและ/หรือเปลี่ยนแปลงในผลิตภัณฑ์และ/หรือโปรแกรมที่อธิบายไว้ในสิ่งพิมพ์นี้ได้ตลอดเวลาโดยไม่ต้องแจ้งให้ทราบ

การอ้างอิงใดๆ ในข้อมูลนี้ถึงเว็บไซต์ที่ไม่ใช่ของ IBM มีการนำเสนอเพื่อความสะดวกเท่านั้น และไม่ได้เป็นการสนับสนุนเว็บไซต์ดังกล่าวในลักษณะใดๆ เนื้อหาที่อยู่ในเว็บไซต์เหล่านั้นไม่ได้เป็น ส่วนหนึ่งของเนื้อหาสำหรับผลิตภัณฑ์ของ IBM นี้ และการใช้เว็บไซต์ดังกล่าวถือเป็นความเสี่ยงของคุณเอง

IBM อาจใช้ หรือแจกจ่ายข้อมูลใดๆ ที่คุณมอบให้ในวิธีใดๆ ซึ่ง IBM เชื่อว่าเหมาะสมโดยไม่ก่อให้เกิดข้อผูกมัดใดๆ กับ คุณ

ผู้รับใบอนุญาตของโปรแกรมนี้ที่ต้องการได้รับข้อมูลเกี่ยวกับโปรแกรมเพื่อเปิดใช้งาน: (i) การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่สร้างขึ้นอย่างอิสระและโปรแกรมอื่นๆ (รวมถึงโปรแกรมนี้) และ (ii) การใช้ข้อมูลที่มีการแลกเปลี่ยนร่วมกัน ควรติดต่อ:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

ข้อมูลดังกล่าวอาจพร้อมใช้งานภายใต้ระยะเวลาและเงื่อนไขที่เหมาะสม โดยมีการชำระค่าธรรมเนียมในบางกรณี

โปรแกรมที่ได้รับอนุญาตซึ่งอธิบายไว้ในเอกสารนี้และเอกสารประกอบที่ได้รับอนุญาตทั้งหมดที่มีอยู่มีการนำเสนอโดย IBM ภายใต้ระยะเวลาของข้อตกลงกับลูกค้าของ IBM, ข้อตกลงเกี่ยวกับใบอนุญาตโปรแกรมระหว่างประเทศของ IBM หรือข้อตกลงที่เท่าเทียมกันใดๆ ระหว่างเรา

ข้อมูลประสิทธิภาพและตัวอย่างลูกค้าที่ระบุมีการนำเสนอสำหรับวัตถุประสงค์การสาธิตเท่านั้น ผลลัพธ์ประสิทธิภาพจริงอาจแตกต่างไปขึ้นอยู่กับคอนฟิกูเรชัน และ เงื่อนไขการปฏิบัติการเฉพาะ

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM ได้มาจากผู้จัดจำหน่ายของผลิตภัณฑ์เหล่านั้น คำประกาศที่เผยแพร่หรือแหล่งข้อมูลที่เปิดเผยต่อ สาธารณะ IBM ไม่ได้ทดสอบผลิตภัณฑ์ดังกล่าว และไม่สามารถยืนยันความถูกต้องของประสิทธิภาพ ความเข้ากันได้ หรือการเรียกร้องอื่นใดที่เกี่ยวข้องกับผลิตภัณฑ์ที่ไม่ใช่ของ IBM หากมีคำถามเกี่ยวกับความสามารถของผลิตภัณฑ์ที่ไม่ใช่ของ IBM ควรสอบถามกับ ผู้จัดจำหน่ายของผลิตภัณฑ์ดังกล่าว

ข้อความเกี่ยวกับทิศทางในอนาคตหรือเจตจำนงค์ของ IBM อาจมีการเปลี่ยนแปลงหรือยกเลิก โดยมิได้มีการแจ้งให้ทราบล่วงหน้า และถือเป็นเพียงข้อมูลเป้าหมายและวัตถุประสงค์เท่านั้น

ราคาที่แสดงทั้งหมดของ IBM เป็นราคาขายปลีกที่แนะนำของ IBM ในปัจจุบัน และอาจเปลี่ยนแปลงได้โดยไม่ต้องแจ้งให้ทราบ ราคาของผู้แทนจำหน่ายอาจแตกต่างออกไป

ข้อมูลนี้ใช้สำหรับวัตถุประสงค์การวางแผนเท่านั้น ข้อมูลในเอกสารฉบับนี้อาจมีการเปลี่ยนแปลง ก่อนที่ผลิตภัณฑ์ที่กล่าวถึงจะมีจำหน่าย

ข้อมูลนี้ประกอบด้วยตัวอย่างข้อมูลและรายงานที่ใช้ในการดำเนินธุรกิจ ประจำวัน เพื่อแสดงให้เห็นอย่างสมบูรณ์ที่สุดเท่าที่จะเป็นไปได้ ตัวอย่างเหล่านี้จึงประกอบด้วย ชื่อของบุคคล บริษัท ตราสินค้า และผลิตภัณฑ์ ชื่อเหล่านี้ทั้งหมดเป็นชื่อสมมติความคล้ายคลึงกับบุคคล หรือเอ็นเตอร์ไพรซ์ทางธุรกิจจริงถือเป็นความบังเอิญ

ใบอนุญาตลิขสิทธิ์:

ข้อมูลนี้ประกอบด้วยโปรแกรมแอ็พพลิเคชันตัวอย่างในภาษาต้นฉบับ ซึ่งแสดงเทคนิคในการเขียนโปรแกรมบนแพล็ตฟอร์มปฏิบัติการที่หลากหลาย คุณสามารถคัดลอก ปรับเปลี่ยน และแจกจ่ายโปรแกรมตัวอย่างเหล่านี้ในรูปแบบต่างๆ ได้โดยไม่ต้องชำระเงินให้แก่ IBM เพื่อใช้สำหรับการพัฒนา การใช้งาน การตลาด หรือการแจกจ่ายโปรแกรมแอ็พพลิเคชันที่สอดคล้องกับอินเทอร์เฟซโปรแกรมแอ็พพลิเคชันของแพล็ตฟอร์มการดำเนินงานที่เขียนโปรแกรมตัวอย่าง ตัวอย่างเหล่านี้ยังไม่ได้ผ่านการ

ทดสอบในทุกสภาพ ดังนั้น IBM จึงไม่สามารถรับประกันหรือแจ้งถึงความน่าเชื่อถือ การให้บริการได้ หรือฟังก์ชันของโปรแกรมเหล่านี้ได้ โปรแกรมตัวอย่างมีการนำเสนอ "ตาม สภาพ" โดยไม่มีการรับประกันประเภทใดๆ IBM ไม่ต้องรับผิดชอบต่อความเสียหายใดๆ ที่เกิดขึ้นจากการใช้โปรแกรมตัวอย่างของคุณ

แต่ละสำเนาหรือส่วนใดๆ ของโปรแกรมตัวอย่างเหล่านี้ หรืองานที่สืบเนื่อง ต้องมีคำประกาศลิขสิทธิ์ดังนี้:

© (ชื่อบริษัทของคุณ) (ปี)

ส่วนต่างๆ ของรหัสนี้ได้มาจากโปรแกรมตัวอย่างของ IBM Corp.

© ลิขสิทธิ์ IBM Corp. _ป้อนปี_

สิ่งที่ต้องพิจารณาเกี่ยวกับนโยบายความเป็นส่วนตัว

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as the customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

เครื่องหมายการค้า

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows เป็นเครื่องหมายการค้า ของ Microsoft Corporation ในประเทศสหรัฐอเมริกา ประเทศอื่นหรือทั้งสอง

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

ดัชนี

C

F

K

S

ก

ข

ค

จ

ต

ท

บ

ป

ผ

ภ

ม

ย

ร

ล

ว

ส

ห

อ

พิมพ์ในสหรัฐอเมริกา