บทที่ 8 การรักษาความปลอดภัยบนระบบเครือข่าย (Network Security)

ในช่วงอายุราวห้าสิบปีของการเกิดระบบเครือข่ายคอมพิวเตอร์ งานส่วนใหญ่ได้ถูกนำมาใช้ โดยผู้ค้นคว้าวิจัยตามสถาบันการศึกษาในการรับและส่งอีเมลล์ (e-mail) และถูกใช้โดยพนักงานขององค์กรต่างๆ ในการใช้งานเครื่องพิมพ์ร่วมกัน ภายใต้เงื่อนไขการใช้งานที่กล่าวมานี้การรักษาความปลอดภัยจึงไม่ใช่เรื่องใหญ่ที่จะต้องนำมาพิจารณาอย่างเป็นจริงเป็นจัง อย่างไรก็ตาม ในปัจจุบันมีประชาชนมากมายที่นำระบบเครือข่ายคอมพิวเตอร์มาใช้งานในเรื่องเกี่ยวกับการธนาคาร การเลือกซื้อสินค้าและการป้อนข้อมูลเกี่ยวกับการเสียภาษีเงินได้ให้แก่รัฐ การรักษาความปลอดภัยบนระบบเครือข่ายฯ จึงได้ทวีความสำคัญขึ้นเป็นอย่างมากเพราะอาจเป็นส่วนที่สร้างปัญหาอย่างใหญ่หลวงขึ้นได้ ในบทนี้จะได้กล่าวถึงการรักษาความปลอดภัยบนระบบเครือข่ายคอมพิวเตอร์ในหลายแง่มุม แสดงให้เห็นถึงความเข้าใจผิดที่เกิดขึ้น และอธิบายอัลกอริทึมและโพรโตคอลมากมายที่ถูกนำมาใช้ในการรักษาความปลอดภัย

ปัญหาเกี่ยวกับความปลอดภัยมักจะเกิดขึ้นโดยความตั้งใจของผู้ประสงค์ร้ายที่พยายามหาประโยชน์ใส่ตนเอง เรียกร้องความสนใจ หรือต้องการที่จะทำร้ายผู้ใดผู้หนึ่ง รูป 8.1 แสดงผู้กระทำผิดส่วนหนึ่งที่มักจะพบได้บ่อยๆ

8.1 หนังสือที่เขียนด้วยอักษรลับ

หนังสือที่เขียนด้วยอักษรลับ (cryptography) มาจากคำในภาษากรีกว่า "secret writing" ในหัวข้อนี้จะได้กล่าวถึงพื้นฐานที่สำคัญบางประการ คำสองคำแรกที่ควรรู้จักคือ "ซิเฟอร์ (ciphers)" และ "โค้ด (codes)" คำว่า ซิเฟอร์ หมายถึง การแทนที่ตัวอักษรด้วยตัวอักษร หรือ บิตด้วยบิต โดยไม่คำนึงถึงโครงสร้างทางภาษาของข่าวสารนั้นๆ ในทางกลับกัน คำว่า โค้ด หมายถึงการแทนที่ "คำ" ด้วย "คำ" หรือ "สัญลักษณ์" อื่น ในปัจจุบันโค้ดไม่ได้ถูกนำมาใช้อีกต่อไปแม้ว่าจะมีประวัติการใช้งานอันน่ายกย่อง ยาวนานมาแต่ในอดีต โค้ดถูกนำมาใช้งานมากที่สุดในระหว่างสงครามโลกครั้งที่สองโดยกองทัพอเมริกันประจำภาคพื้นแปซิฟิก โค้ดที่นำมาใช้เป็นภาษาอินเดียนแดงเรียกว่า "Navajo" ซึ่งเป็นภาษาพูดที่ไม่มีภาษาเขียนเป็นของตนเองและมีความซับซ้อนมาก ที่สำคัญที่สุดคือศัตรูของสหรัฐอเมริกาในเวลานั้นคือประเทศญี่ปุ่นซึ่งไม่มีความรู้เกี่ยวกับภาษานี้เลย

8.1.1 แนะนำการเขียนหนังสือด้วยอักษรลับ

ตามประวัติความเป็นมาในอดีต บุคคลสี่กลุ่มที่ได้รับการยกย่องว่ามีความเกี่ยวพันกับศิลปะของการเขียนหนังสือด้วยรหัสลับคือ ทหาร นักการฑูต นักบันทึกเหตุการณ์ และนักรัก ทหารนับว่าเป็นกลุ่มบุคคลที่มีความสำคัญมากที่สุดและมีความเกี่ยวข้องมามากกว่าหนึ่งศตวรรษภายในโครงสร้างองค์กรทาง

รูป 8-1
ตัวอย่างผู้ที่ก่อปัญหาให้กับระบบรักษาความปลอดภัยและเหตุผลที่ทำ

Adversary	Goal
Student	To have fun snooping on people's e-mail
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by e-mail
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

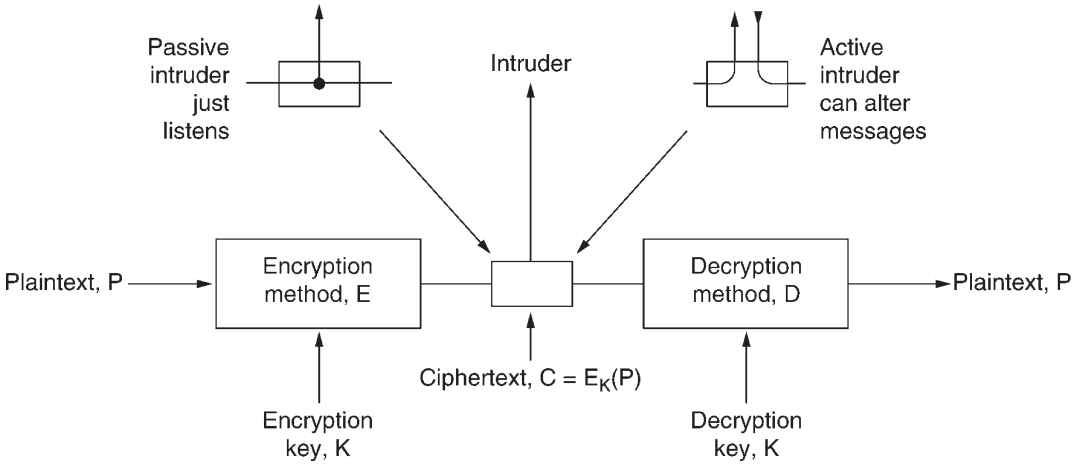
ทหาร ข่าวสารส่วนใหญ่ที่ถูกนำมาเข้ารหัสนั้นถูกส่งให้แก่ทหารขั้นผู้น้อยที่มีเงินเดือนต่ำทำการเข้ารหัสและส่งข่าวสารนั้นออกไป ทั้งนี้เนื่องจากปริมาณข่าวสารที่มีอยู่มากมายทำให้การเข้ารหัสไม่ได้กระทำโดยผู้เชี่ยวชาญซึ่งมีอยู่เพียงจำนวนน้อย

ก่อนที่จะนำคอมพิวเตอร์เข้ามาใช้งาน เงื่อนไขที่สำคัญในการเขียนข้อความด้วยรหัสลับนั้นคือความสามารถของเจ้าหน้าที่ในการเปลี่ยนแปลงรูปแบบของข่าวสารซึ่งมีเครื่องมือช่วยเพียงเล็กน้อยเท่านั้น นอกจากนี้ยังมีความยุ่งยากเกี่ยวกับการเปลี่ยนวิธีการเข้ารหัสจากแบบหนึ่งเป็นอีกแบบหนึ่งอย่างรวดเร็ว เนื่องจากอันตรายจากการที่เจ้าหน้าที่สื่อสารจะถูกจับตัวไปโดยข้าศึกทำให้มีความสำคัญที่จะต้องสามารถเปลี่ยนวิธีการเข้ารหัสได้อย่างรวดเร็ว ความขัดแย้งในความต้องการนี้แสดงให้เห็นในรูปที่ 8-2

ข่าวสารที่ถูกนำไปเข้ารหัสเรียกว่า "plaintext" จะถูกเปลี่ยนแปลงรูปแบบไปโดยถูกควบคุมด้วย "กุญแจ หรือคีย์ (key)" ผลลัพธ์ที่ได้คือข่าวสารที่ถูกเข้ารหัสแล้ว เรียกว่า "ciphertext" ซึ่งจะถูกส่งออกไปโดยเจ้าหน้าที่ส่งข่าวสารหรือสัญญาณวิทยุ ข่าวสารที่ถูกส่งออกไปจะถูกสมมุติว่าถูกตรวจพบโดยข้าศึก (enemy or intruder) อย่างไรก็ตาม ข้าศึกซึ่งไม่มีกุญแจสำหรับการถอดรหัส (decryption key) จะไม่สามารถถอดรหัสจากข้อความ ciphertext ได้ ศิลปของการถอดข้อความรหัส cipher เรียกว่า "cryptanalysis" และศิลปของการประดิษฐ์การเข้ารหัสข้อความเรียกว่า "cryptology"

ความสัมพันธ์ระหว่าง plaintext (P), ciphertext (C), และ คีย์(K) สามารถกำหนดได้ดังนี้ C = Ek(P) ซึ่งอธิบายว่าการเข้ารหัสข้อความธรรมดา P โดยการใช้คีย์ K จะได้ผลลัพธ์ออกมาเป็นข้อความที่เข้ารหัสแล้วคือ C และในทำนองเดียวกันความสัมพันธ์ P = Dk(C) อธิบายถึงการถอดรหัสข้อความ C ด้วยคีย์ K จะได้ออกมาเป็นข้อความปกติ P ส่วนความสัมพันธ์ P = Dk(Ek(P)) นั้นบอกให้ทราบว่า E และ D เป็นเพียงฟังก์ชันทางคณิตศาสตร์ที่ถูกนำมาใช้เท่านั้น

กฎพื้นฐานของการเข้ารหัสข้อความคือ ทุกคนจะต้องคิดเสมือนหนึ่งว่าผู้ที่ทำงานเกี่ยวกับการเข้าและถอดรหัสข้อความนั้นรู้จักวิธีที่ทำการเข้ารหัสและถอดรหัสข้อความ นั่นคือผู้ที่ทำงานเกี่ยวกับการเข้าและถอดรหัสข้อความรู้จักรายละเอียดของวิธีการเข้ารหัส E และวิธีการถอดรหัส D ดังที่แสดงในรูป 8.2 เป็นอย่างดี ความพยายามในการสร้างวิธีการใหม่ๆ ขึ้นมา ทำการทดสอบ และติดตั้งใช้งานในทุกครั้งที่วิธีการเดิมถูกเปิดเผยออกไปแล้วนั้นเป็นเรื่องที่เป็นไปไม่ได้ในทางปฏิบัติ การคิดว่าวิธีการที่ตนเองนำ



รูปที่ 8-2
โมเดลของการเขียนหนังสือด้วยรหัสลับ

มาใช้นั้นเป็นวิธีการลับที่ไม่มีผู้ใดทราบนั้นมักจะทำให้เกิดผลร้ายมากกว่าผลดีเสมอ

จุดนี้เองที่ทำให้คีย์เข้ามามีบทบาท คีย์หรือกุญแจรหัสนั้นประกอบด้วยสายอักขระสั้นๆ ที่ถูกนำมาใช้ในการเลือกวิธีการเข้ารหัส โดยทั่วไปแล้ว การเลือกวิธีการเข้ารหัสและถอดรหัสโดยทั่วไปนั้นจะมีการเปลี่ยนแปลงเกิดขึ้นในทุกสองถึงสามปีแต่การเลือกคีย์นั้นสามารถที่จะเปลี่ยนแปลงได้บ่อยเท่าที่ต้องการ ดังนั้นรูปแบบพื้นฐานของระบบการเข้ารหัสข้อความคือการเลือกใช้วิธีการเข้ารหัสถอดรหัสที่เป็นที่รู้จักกันโดยทั่วไปซึ่งจะถูกควบคุมการทำงานโดยคีย์ที่สามารถแก้ไขเปลี่ยนแปลงได้โดยง่าย กฎพื้นฐานนี้เรียกว่า "Kerckhoffs principle" ซึ่งสรุปสั้นๆ ได้ว่า อัลกอริทึมที่นำมาใช้ในการเข้าและถอดรหัสข้อความนั้นเป็นอัลกอริทึมที่รู้จักกันโดยทั่วไป มีแต่คีย์เท่านั้นที่เป็นความลับ

การทำให้วิธีการเข้าและถอดรหัสเป็นที่รู้จักกันโดยทั่วไปนั้นทำให้เกิดผลดีประการหนึ่งคือ ผู้ที่ออกแบบอัลกอริทึมนั้นคล้ายกับว่าจะได้รับคำปรึกษาฟรีจากผู้ที่มีความอยากรู้อยากเห็นและอยากที่จะแก้รหัสข้อความนั้นให้ได้ จึงพยายามทดลองแก้ไขข้อความรหัสด้วยวิธีการต่างๆ กัน โดยทั่วไปแล้วอัลกอริทึมที่ไม่มีผู้ใดสามารถถอดรหัสข้อความออกโดยไม่ใช้คีย์ได้ภายในระยะเวลา 5 ปี ถือว่าอัลกอริทึมนั้นมีความน่าเชื่อถือได้

เนื่องจากความลับทั้งหมดนั้นขึ้นอยู่กับคีย์ ขนาดความยาวของคีย์จึงเป็นประเด็นที่สำคัญในการออกแบบคีย์ขึ้นมาใช้งาน ดูตัวอย่างจากกุญแจที่ล็อคด้วยตัวเลข หลักการพื้นฐานคือการพยายามใส่ตัวเลขที่ถูกต้องเข้าไปเพื่อเปิดล็อคกุญแจนั้น ในที่นี้หมายเลขล็อคจึงหมายถึงคีย์ หากหมายเลขล็อคเป็นตัวเลขขนาด 2 หลัก แสดงว่ามีคีย์ที่เป็นไปได้จำนวน 100 รหัส (00, 01, 02,..., 99) ถ้าความยาวของเลขล็อคเป็น 3 หลัก ก็จะมีคีย์ทั้งหมด 1000 ตัว ถ้าความยาวของเลขล็อคเป็น 6 หลัก ก็จะมีคีย์ทั้งหมด 1 ล้านตัว จะเห็นได้ว่าถ้าคีย์มีความยาวมากขึ้นเท่าใดก็จะทำให้ผู้ที่พยายามถอดรหัสนั้นมีงานต้องทำ (เรียกว่า work factor) มากขึ้น ปริมาณงาน work factor สำหรับความเป็นไปได้ในการถอดรหัสจะมีความสัมพันธ์เป็นแบบเอ็กซ์โปเนนเชียลกับความยาวของคีย์ ความลับของกระบวนการเข้าและถอดรหัสจึงมาจากการใช้อัลกอริทึมที่มีผู้รู้จักทั่วไปและมีคีย์ที่มีความยาว สำหรับการปกป้องอีเมลล์จากคนในครอบครัวเดียวกัน การใช้คีย์ขนาด 64 บิตถือว่าเพียงพอแล้ว สำหรับงานขององค์กรธุรกิจทั่วไปอาจใช้คีย์ขนาด 128 บิต และสำหรับการปกป้องเอกสารสำคัญของรัฐบาลอาจต้องใช้คีย์ขนาด

สำหรับกระบวนการเข้าและถอดรหัสข้อความ ได้แบ่งปัญหาความพยายามในการถอดรหัส (โดยไม่ใช้คีย์) ไว้เป็น 3 กลุ่มคือ กลุ่มแรกเรียกว่า ciphertext-only problem ซึ่งหมายถึงผู้ที่พยายามจะถอดรหัสข้อความนั้นมีเพียงแต่ข้อความที่ถูกเข้ารหัสไว้แล้วเท่านั้น กลุ่มที่สองเรียกว่า Know plaintext problem ซึ่งหมายถึงการที่มีข้อความที่เข้ารหัสแล้วและข้อความปกติของข้อความนั้นอยู่ด้วยเพื่อนำมาใช้เป็นข้อมูลสำหรับการถอดข้อความรหัสอื่น และกลุ่มที่สามเรียกว่า chosen plaintext problem คือการที่สามารถเข้ารหัสข้อความปกติใดๆ ให้เป็นข้อความเข้ารหัสได้ด้วยตนเอง

ระบบการรักษาความปลอดภัยขององค์กรทางธุรกิจโดยทั่วไปนั้นตั้งอยู่บนสมมุติฐานว่า ถ้า ciphertext ของตนเองสามารถปลอดภัยจากการโจมตีในกลุ่ม ciphertext-only ได้ ถือว่าระบบการรักษาความปลอดภัยนั้นสามารถนำมาใช้งานได้ ข้อสมมุติฐานนี้ค่อนข้างที่จะตื้นเกินไปเพราะในหลายกรณีโดยเฉพาะสำหรับผู้บุกรุกที่มีประสบการณ์แล้ว การเดาอย่างมีเหตุผลก็อาจสามารถถอดรหัสของระบบนี้ได้ ถ้าผู้บุกรุกมีทั้งข้อความปกติและข้อความที่เข้ารหัสแล้วของข้อความนั้นอยู่ในมือ การถอดรหัสข้อความอื่นก็เป็นเรื่องที่ง่ายยิ่งขึ้น ดังนั้นระบบที่ถือว่ามีความปลอดภัยมากนั้น ระบบนั้นจะต้องสามารถป้องกันผู้บุกรุกจากในกรณีนี้ได้หรือแม้กระทั่งในกลุ่มสุดท้ายคือผู้บุกรุกสามารถเข้ารหัสข้อความใดๆ ได้ด้วยตนเองนั้น ระบบที่มีความปลอดภัยสูงก็จะต้องยังคงปลอดภัยคือผู้บุกรุกไม่สามารถถอดรหัสข้อความที่เข้ารหัสแล้วได้

8.1.2 วิธีแทนที่ซิเฟอร์

วิธีแทนที่ซิเฟอร์ (substitution cipher) จะแทนตัวอักษรตัวหนึ่งหรือกลุ่มหนึ่งด้วยตัวอักษรอีกตัวหนึ่งหรืออีกกลุ่มหนึ่งเพื่อปิดบังค่าที่แท้จริง วิธีการแบบนี้ที่เก่าแก่ที่สุดเรียกว่า Caesar cipher ซึ่งเกิดขึ้นในสมัยของจูเลียดซีซ่าร์ (Julius Caesar) ตัวอย่างเช่น แทนตัวอักษร a ด้วย D, b ด้วย E, c ด้วย F, ... ดังนั้นคำว่า "attack" จะถูกแปลงให้ไปอยู่ในรูป "DWWDFN" เป็นต้น การดัดแปลงวิธีการนี้เพียงเล็กน้อยจะทำให้เกิดการเลื่อนตัวอักษรได้ k ตัวแทนที่จะเป็นเพียงแค่ 3 ตัวเสมอ ดังนั้นค่าของ k จึงกลายมาเป็นคีย์สำหรับวิธีการเลื่อนตัวอักษรนี้

การปรับปรุงในขั้นต่อไปคือการแทนที่ตัวอักษรแต่ละตัวด้วยตัวอักษรอื่นที่ไม่ได้อยู่เรียงตามลำดับกัน เช่น

Plaintext: a b c d e f g h i j k l m n o p q r s t u v w x y z

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

วิธีการแทนที่ตัวอักษรตัวหนึ่งด้วยตัวอักษรอีกตัวหนึ่งเรียกว่า monoalphabetic substitution โดยมีคีย์เป็นสายอักขระยาว 26 ตัวอักษรซึ่งสอดคล้องกับตำแหน่งของตัวอักษรแต่ละตัว ดังนั้น คำว่า "attack" จะถูกแปลงเป็นคำว่า "QZZQEA" เป็นต้น

วิธีการนี้เมื่อมองดูผิวเผินแล้วอาจเห็นว่าเป็นวิธีการที่ดีเนื่องจากผู้ที่พยายามจะถอดรหัสจะต้องพยายามเลือกรหัสที่ถูกต้องเพียง 1 ชุดจากที่เป็นไปได้ทั้งหมด $26! \approx 4 \times 10^{26}$ วิธี ถ้าหากว่าการพยายามทดลองวิธีการที่เป็นไปได้ 1 วิธีใช้เวลาในการทดลอง 1 นาโนวินาที (ส่วนพันล้านวินาที) การ

ทดลองทุกวิธีที่เป็นไปได้ทั้งหมดนี้จะต้องใช้เวลานานถึงประมาณ 10^{10} ปีเลยทีเดียว

อย่างไรก็ตาม วิธีการนี้สามารถถูกถอดรหัสออกได้ไม่ยากเย็นนัก วิธีการที่นำมาใช้ในการถอดรหัส คือ การสังเกตุความถี่ในการใช้ตัวอักษรในภาษาอังกฤษจะพบว่า ตัวอักษร e จะถูกใช้มากที่สุด ตามด้วย t, o, a, n, i, ... และตัวอักษรที่มักจะอยู่คู่กันบ่อยๆ (เรียกว่า digram) ได้แก่ th, in, er, re, และ an ส่วนตัวอักษรที่มักจะอยู่ติดกัน 3 ตัวบ่อยๆ (เรียกว่า trigram) ได้แก่ the, ing, and, และ ion นักถอดรหัสที่พยายามจะถอดรหัสแบบ monoalphabetic cipher นี้จะเริ่มต้นสังเกตุดูตัวอักษรที่ใช้มากที่สุดใน ciphertext ซึ่งอาจจะเริ่มต้นด้วยการสมมุติให้ตัวอักษรที่ปรากฏอยู่มากที่สุดเป็นตัว e และตัวที่มีความถี่รองลงมาเป็นตัว t จากนั้นอาจจะสังเกตุเห็นคำ trigram เช่น tXe ปรากฏขึ้น ซึ่งมีโอกาสเป็นไปได้สูงที่ตัว X จะหมายถึงตัว h โดยการสังเกตุในทำนองเดียวกัน คำว่า thYt อาจชี้แนะว่าตัว Y นั้นคือตัว a หรือคำว่า aZW อาจเป็นคำว่า and อาศัยการสังเกตุและทดลองไปเรื่อยๆ ในลักษณะเช่นนี้ ผู้ถอดรหัสอาจสามารถถอดรหัส ciphertext ได้เร็วกว่า 10^{10} ปีมากมายนัก

อีกวิธีการหนึ่งที่สามารถนำมาใช้ในการถอดรหัสได้คือการเดาคำ (guessing) เช่น สมมุติว่ามี ciphertext ที่นำมาจากสถาบันการเงินแห่งหนึ่ง (เมื่อเขียนโดยจับกลุ่มทีละ 5 ตัวอักษร) จะปรากฏดังนี้

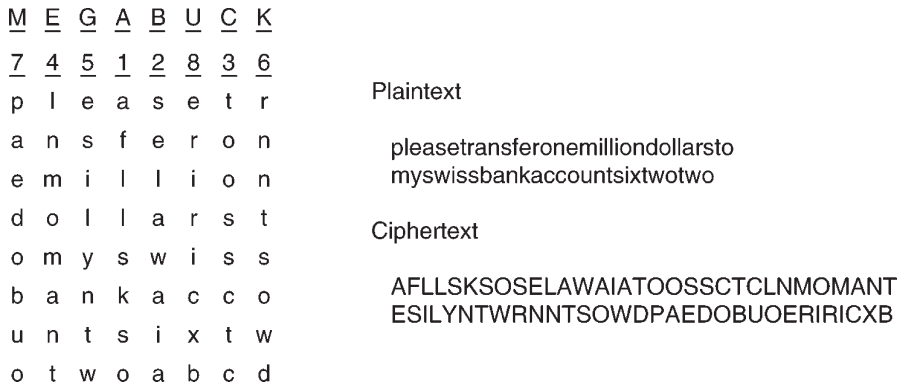
```
CTBMN  BYCTC  BTJDS  QXBNS  GSTJC  BTSWX  CTQTZ  CQVUJ
QJSGS  TJQZZ  MNQJS  VLNSX  VSZJU  JDSTS  JQUUS  JUBXJ
DSKSU  JSNTK  BGAQJ  ZBGYQ  TLCTZ  BNYBN  QJSW
```

คำที่มักจะปรากฏอยู่ในข้อความที่มาจากสถาบันการเงินได้แก่คำว่า "financial" ถ้าคำนี้มีอยู่ใน ciphertext ข้างบนนี้แล้วจะต้องมีตัวอักษรจำนวน 4 ตัว ("nanc") คั่นอยู่ระหว่างตัว "i" คู่หนึ่ง เมื่อพิจารณาจาก ciphertext ข้างบนจะพบว่ามีรูปแบบนี้ปรากฏอยู่ทั้งหมด 12 แห่งด้วยกัน เช่น ที่ตำแหน่งที่ 6 ที่บรรทัดบนสุดมีคำว่า "BYCTCB" และที่ตำแหน่ง 15, 27, 31, 42, 48, 56, 66, 70, 71, 76, และ 82 อย่างไรก็ตาม ถ้าข้อสมมุติแรกถูกต้อง ในคำที่พบทั้งหมดนี้จะต้องมีคำว่า "nan" อยู่ภายในด้วยซึ่งมีเพียงคำที่ตำแหน่งที่ 31 และ 42 เท่านั้นที่น่าจะถูกต้อง ท้ายที่สุด คำที่ตำแหน่ง 31 น่าจะเป็นคำที่ถูกต้องแต่เพียงคำเดียวเนื่องจากตัวอักษร "a" นั้นอยู่ในตำแหน่งที่ถูกต้อง ด้วยวิธีการในทำนองเดียวกันนี้ก็จะสามารถถอดรหัส ciphertext ที่ต้องการได้ในที่สุด

8.1.3 ซิเฟอร์ที่ทำการสับเปลี่ยนตำแหน่งตัวอักษร

ซิเฟอร์แบบแทนที่ตัวอักษรยังรักษาลำดับเดิมของทุกตัวอักษรในข้อความต้นฉบับเอาไว้แต่จัดการสับเปลี่ยนตัวอักษรเหล่านั้นเป็นตัวอักษรอื่นทั้งหมด ซิเฟอร์แบบที่ทำการสับเปลี่ยนตำแหน่งตัวอักษร (Transposition ciphers) จะจัดลำดับตัวอักษรเสียใหม่แต่ยังคงตัวอักษรเดิมเอาไว้ รูป 8-3 แสดงตัวอย่างการทำงานของซิเฟอร์แบบที่ทำการสับเปลี่ยนตำแหน่งตัวอักษรแบบหนึ่งที่ทำการสับตำแหน่งคอลัมน์ ข้อความซิเฟอร์จะถูกกำหนดคีย์ให้เป็นคำหรือวลีที่ไม่มีตัวอักษรซ้ำกันเลยซึ่งในที่นี้คือคำว่า MEGABUCK วัตถุประสงค์ของการกำหนดคีย์ก็คือต้องการนำมาใช้ในการจัดลำดับคอลัมน์โดยกำหนดให้เรียงจากตัวอักษรที่ใกล้ "A" มากที่สุดเป็นลำดับที่ 1 ไปจนถึงตัวอักษรที่ใกล้ "Z" มากที่สุดเป็นลำดับสุดท้าย ในที่นี้ A=1, B=2, C=3, E=4, G=5, K=6, M=7, และ U=8 ตามลำดับ ข้อความต้นฉบับหรือ plaintext จะถูกนำมาเขียนเรียงตามลำดับปกติ (ตามแนวนอน) ถ้าในแถวสุดท้ายมีช่องว่างก็อาจหาตัวอักษรใดๆ มาเติมให้เต็มก็ได้ ข้อความจะถูกเข้ารหัสด้วยการเขียนข้อความต้นฉบับเสียใหม่โดยให้

รูปที่ 8-3
การเข้ารหัสด้วยการเปลี่ยนตำแหน่งข้อมูล



เรียงตามลำดับคอลัมภ์ที่ถูกกำหนดด้วยคีย์ เริ่มต้นจากคอลัมน์ที่ 1 ทั้งหมด ตามด้วยคอลัมน์ที่ 2 และเรียงตามลำดับจนครบทุกคอลัมน์ดังที่แสดงในรูป

ในการพยายามถอดรหัสโดยไม่ใช้คีย์ ผู้ทำการถอดรหัสจะต้องทราบว่าข้อความนี้เกิดขึ้นจากวิธีการสับเปลี่ยนตำแหน่งตัวอักษร จากการสังเกตความถี่ในการใช้ตัวอักษรต่างๆ เช่น E, T, A, O, I, N, ... ก็จะสามารถสังเกตุได้ว่าความถี่ของตัวอักษรเหล่านี้ไม่มีการเปลี่ยนแปลง เนื่องจากวิธีการสับเปลี่ยนตำแหน่งยังคงใช้ตัวอักษรชุดเดิมทั้งหมด ความถี่ในการใช้ตัวอักษรจึงยังคงเหมือนเดิม

ขั้นต่อไปคือการเดาว่าข้อความนี้ถูกแบ่งออกเป็นกี่คอลัมน์ โดยทั่วไปคีย์ที่นำมาใช้จะสามารถเดาได้จากเนื้อหาของข้อความที่เข้ารหัสนั้น เช่น ถ้าสงสัยว่าจะมีคำว่า milliondollars ปรากฏอยู่ในข้อความนี้ก็อาจจะมีตัวอักษรคู่ MO, IL, LL, LA, IR, และ OS ปรากฏอยู่ ถ้าระยะห่างของตัวอักษรเท่ากับ 9 ตำแหน่ง โดยข้อเท็จจริงแล้ว สำหรับแต่ละระยะห่างจะปรากฏข้อความที่แตกต่างกันไปใน ciphertext ซึ่งสามารถเดาได้ไม่ยากนัก ดังนั้น เพียงใช้ความพยายามในการเดานี้สักระยะหนึ่งผู้ถอดรหัสก็จะทราบระยะห่างที่ถูกนำมาใช้ได้ ขั้นตอนที่เหลือก็คือการจัดลำดับคอลัมน์เสียใหม่ก็จะได้ข้อความต้นฉบับที่ต้องการ

8.1.4 รหัสที่ใช้งานเพียงครั้งเดียว

การสร้าง ciphertext ที่ไม่สามารถถอดรหัส (โดยผู้ที่ไม่ได้รับอนุญาต) ได้นั้นอันที่จริงไม่ยากนัก ซึ่งเป็นวิธีการที่รู้กันมานานนับสิบปีมาแล้ว ขั้นตอนแรกให้เลือกสายอักขระบิต (bit string) ขึ้นมาแบบสุ่มเลือก จากนั้นจึงเปลี่ยน plaintext ให้เป็นสายอักขระบิต เช่น การใช้รหัสข้อมูล ASCII จากนั้นให้ทำการ exclusive-or สายอักขระทั้งสองเข้าด้วยกันทีละบิต ผลที่ได้รับจะเป็น ciphertext ที่ไม่มีผู้ใดจะสามารถถอดรหัสได้ วิธีการนี้เรียกว่าการใช้รหัสที่ใช้งานเพียงครั้งเดียว (One-time pad) ซึ่งสามารถป้องกันการแอบถอดรหัสได้ไม่ว่าผู้บุกรุกนั้นจะมีเครื่องคอมพิวเตอร์ที่มีขีดความสามารถสูงเพียงใดก็ตาม

รูป 8-4 แสดงตัวอย่างการนำรหัสที่ใช้งานครั้งเดียวมาใช้งาน ข้อความ plaintext ที่นำมาใช้คือ "I love you" จะถูกเปลี่ยนให้เป็นรหัสแทนข้อมูล ASCII ขนาด 7 บิตต่อหนึ่งตัวอักษร จากนั้นนำรหัส pad-1 มาทำการ exclusive-or เข้ากับข้อความนี้กลายเป็น ciphertext

รหัสแบบที่ใช้งานเพียงครั้งเดียวนี้แม้ว่าจะดีมากในทางทฤษฎี แต่ก็มีข้อบกพร่องในการนำไปใช้งาน

```
Message 1:  1001001 0100000 1101100 1101111 1110110 1100101 0100000 1111001 1101111 1110101 0101110
Pad 1:      1010010 1001011 1110010 1010101 1010010 1100011 0001011 0101010 1010111 1100110 0101011
Ciphertext: 0011011 1101011 0011110 0111010 0100100 0000110 0101011 1010011 0111000 0010011 0000101

Pad 2:      1011110 0000111 1101000 1010011 1010111 0100110 1000111 0111010 1001110 1110110 1110110
Plaintext 2: 1000101 1101100 1110110 1101001 1110011 0100000 1101100 1101001 1110110 1100101 1110011
```

รูปที่ 8-4 การเข้ารหัสข้อมูลโดยการนำรหัสที่ใช้งานเพียงครั้งเดียวมาใช้

ปัญหาประการแรกคือคีย์ที่นำมาใช้นั้นไม่มีผู้ใดจะสามารถจดจำได้หมด ทำให้ทั้งผู้ส่งและผู้รับข่าวสารจำเป็นจะต้องจดบันทึกไว้ ถ้าฝ่ายใดฝ่ายหนึ่งมีความเสี่ยงที่จะถูกจับโดยข้าศึกแล้วการมีคีย์ติดตัวจึงเป็นเรื่องที่ไม่พึงประสงค์ ประการต่อมา ปริมาณของข้อมูลที่จะสามารถส่งออกไปได้นั้นขึ้นอยู่กับขนาดของคีย์ที่ใช้ ผู้ส่งข่าวสารที่มีปริมาณข่าวสารยาวมากจะไม่สามารถนำคีย์นั้นมาใช้ในการส่งข่าวสารได้ ปัญหาอีกประการหนึ่งคือความอ่อนไหวต่อการสูญหายหรือการเพิ่มขึ้นของข้อมูลโดยไม่ได้ตั้งใจ ถ้าผู้ส่งและผู้รับข่าวสารไม่สามารถจัดลำดับของคีย์ทั้งสองฝ่ายให้ตรงตำแหน่งกันทุกตำแหน่งแล้ว ผลที่ได้รับก็จะเป็นเพียงข่าวสารที่ไม่อาจใช้การได้

การเข้ารหัสแบบ Quantum

เป็นเรื่องที่น่าสนใจมากที่การแก้ปัญหาการส่ง one-time pad ผ่านระบบเครือข่ายนั้นสามารถทำได้โดยใช้วิธี quantum mechanics ซึ่งแม้ว่าจะอยู่ในขั้นตอนการค้นคว้าวิจัยแต่ผลการวิจัยพบว่ามีทางเป็นไปได้ในการนำมาใช้งานจริง เมื่อใดที่สามารถนำวิธีการนี้มาใช้งานได้จริงก็จะทำให้การเข้ารหัสข้อมูลนั้นหันมาใช้ one-time pad กันเป็นส่วนมากเนื่องจากเป็นวิธีที่ค่อนข้างจะปลอดภัยมาก

การเข้ารหัสแบบ quantum นี้ใช้กับการสื่อสารสัญญาณแสงผ่านสายใยแก้วนำแสง ลำแสงจะถูกส่งออกมาเป็นแพ็กเก็ตเรียกว่า "โฟตอน (photons)" ซึ่งมีคุณสมบัติเฉพาะของตนเอง ยิ่งกว่านี้ลำแสงอาจถูกทำให้เกิดขั้วโดยการส่องลำแสงผ่านฟิลเตอร์ชนิด polarizing filter ซึ่งเป็นฟิลเตอร์ชนิดเดียวกันกับแว่นกันแดดแบบโพรารอยด์ที่ผู้คนทั่วไปใช้ หรือที่นำมาใช้เป็นฟิลเตอร์กรองแสงสำหรับกล้องถ่ายรูป ถ้าลำแสง (หรือกระแสของโฟตอน) ผ่านฟิลเตอร์ออกมาลำแสงนั้นจะถูกจัดขั้ว (polar) หรือทิศทางให้ตรงกับขั้วของฟิลเตอร์นั้น (เช่น ขั้วตามแนวตั้งหรือแนวนอน) ถ้าลำแสงนี้ถูกส่งผ่านฟิลเตอร์ตัวที่สองจะทำให้ความเข้มของลำแสงที่ผ่านออกมานั้นมีค่าเป็น $\cos(\alpha)$ เมื่อ α คือมุมระหว่างขั้วทั้งสองของฟิลเตอร์ ซึ่งถ้าขั้วทั้งสองตั้งฉากกันก็จะไม่มีลำแสงลอดออกมาเลย ค่าของมุม α นี้คือสิ่งที่จะนำมาใช้

สมมุติว่า จินตรา ต้องการจัดส่ง one-time pad ไปให้ ธงไชย เพื่อจัดการสื่อสารที่ปลอดภัยระหว่างคนทั้งสอง จินตราจะต้องใช้ฟิลเตอร์สองชุด ชุดละสองตัว ชุดแรกประกอบด้วยฟิลเตอร์ตามแนวนอนและฟิลเตอร์ตามแนวตั้ง เรียกว่า rectilinear basis ซึ่งจะถูกนำมาใช้เป็นวิธีการบอกตำแหน่ง (coordinate system) ฟิลเตอร์ชุดที่สองก็มีลักษณะแบบเดียวกันและทำมุม 45 องศากับชุดแรก เรียกว่า diagonal basis ทำให้จินตรามีฐานอ้างอิงสองฐาน ในความเป็นจริงจะใช้ฟิลเตอร์เพียงตัวเดียวที่สามารถเปลี่ยนแนวขั้วได้โดยใช้สัญญาณไฟฟ้าควบคุม ทำให้สามารถสลับเปลี่ยนขั้วฟิลเตอร์ไปเป็นทิศทางใดก็ได้ (หนึ่งในสี่ทิศทาง) ได้อย่างรวดเร็ว ธงไชยก็จะมีฟิลเตอร์อย่างเดียวกับที่จินตรามี

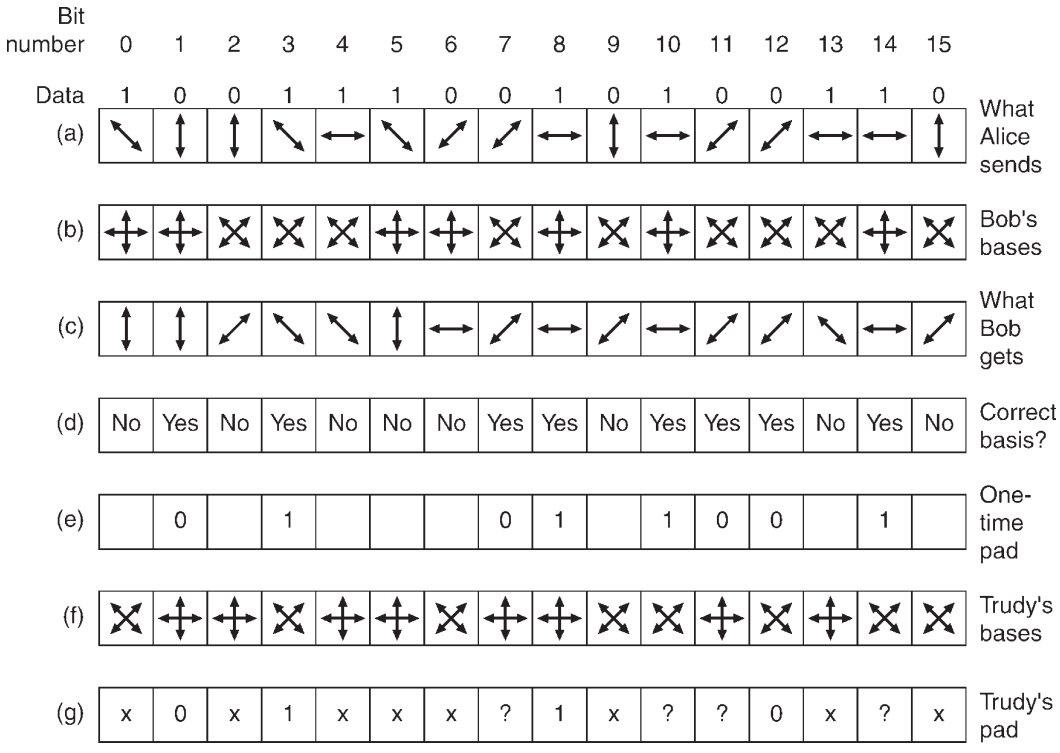
จินตราจะกำหนดให้ทิศทางหนึ่งสำหรับแต่ละฐานให้มีค่าเป็น "0" และอีกทิศทางหนึ่งเป็น "1"

เช่นตัวอย่างต่อไปนี้กำหนดให้ทิศทางแนวตั้งเป็น "0" และทิศทางแนวนอนเป็น "1" ในทำนองเดียวกันทิศทางเฉียงจากมุมล่างซ้ายไปทางบนขวามีค่าเป็น "0" และมุมล่างขวาไปทางมุมบนซ้ายเป็น "1" จินตราส่งข้อมูลนี้ให้แก่ธงไชยในรูปแบบของข้อความที่ไม่ได้เข้ารหัส (plaintext)

ต่อไป จินตราเลือก one-time pad ขึ้นมาหนึ่งชุด เช่น ชุดที่สร้างขึ้นมาจากการสุ่มตัวเลข (ซึ่งมีความซับซ้อนอยู่ในตัวเอง) จินตราจะส่งข้อมูลนี้ไปยังธงไชยทีละบิตโดยการเลือกฐานแบบสุ่มจากสองฐานที่กำหนด ในการส่งบิตนั้น ต้นกำเนิดแสงจะกำหนดขั้วจากฐานที่เลือกแล้วส่งลำแสงออกไป ดังแสดงในรูป 8-5(a) จินตราอาจเลือกฐานเป็น diagonal, rectilinear, rectilinear, diagonal, rectilinear,... ในการส่ง one-time pad "1001110010100110" ไปยังธงไชย บิตแต่ละบิตที่ส่งออกไปนี่เรียกว่า qubits ซึ่งการเลือก one-time pad และลำดับของฐานที่ใช้จะสามารถใช้กำหนดเอกลักษณ์ของข้อมูลแต่ละบิตได้

เนื่องจากธงไชยไม่ทราบว่าจินตราเลือกใช้ฐานในลำดับใด ธงไชยจึงเลือกฐานที่ใช้สำหรับข้อมูลแต่ละบิตเองดังแสดงในรูป 8-5(b) ถ้าธงไชยเลือกฐานที่ถูกต้อง เขาก็จะได้บิตที่ถูกต้องไป แต่ถ้าเลือกฐานผิดเขาจะได้รับบิตข้อมูลมาอย่างเดาสุ่ม เนื่องจากถ้าโฟตอนผ่านฟิลเตอร์ที่มุม 45 องศาจากมุมฐานของตนเองโฟตอนจะเปลี่ยนทิศทางไปยังแกนใดแกนหนึ่งของฐานด้วยความน่าจะเป็นเท่า ๆ กัน ซึ่งเป็นคุณสมบัติพื้นฐานของ quantum mechanics บิตบางส่วนก็อาจถูกต้องและส่วนที่เหลือก็จะผิดโดยที่ธงไชยไม่สามารถทราบได้ ดังแสดงในรูป 8-5(c)

ธงไชยจะส่งข้อความ plaintext กลับไปบอกจินตราว่าตนเองใช้ฐานในลำดับใดซึ่งจินตราก็จะตอบกลับมาด้วย plaintext ให้ทราบว่าฐานใดถูกฐานใดผิด ดังแสดงในรูป 8-5(d) ด้วยข้อมูลนี้จะทำให้ทั้งสองฝ่ายสามารถสร้างกระแสบิต (bit string) ของข้อมูลการเดาที่ถูกต้อง ดังแสดงในรูป 8-5(e) โดย



รูปที่ 8-5 ตัวอย่างการทำงานของการเข้ารหัสแบบ quantum

ทั่วไปแล้ว กระแสบิตชุดที่ได้รับนี้จะมีความยาวประมาณครึ่งหนึ่งของกระแสบิตต้นฉบับที่ถูกต้อง เนื่องจากทั้งสองฝ่ายต่างก็ทราบกระแสบิตชุดใหม่นี้ จึงสามารถนำมาใช้เป็น one-time pad ระหว่างกันได้ เพื่อให้ได้ one-time pad ที่มีความยาวเท่ากับความยาวที่ต้องการ จินตราก็เพียงแต่ส่งข้อมูลกระแสบิตเริ่มต้นให้มีความยาวประมาณสองเท่าของความยาวที่ต้องการ

สมมุติต่อไปว่า สุวนันท์ ทำการตัดต่อสายใยแก้วนำแสงที่ส่งออกมาจาก จินตราไปยังธงไชยเพื่อต้องการขโมยข้อมูล เนื่องจากว่า สุวนันท์ก็ไม่ทราบลำดับของฐานที่จินตราใช้ จึงต้องเดาสุ่มเช่นเดียวกับที่ธงไชยกระทำขึ้นมาใช้งานเอง ดังแสดงในรูป 8-5(f) เมื่อจับสัญญาณตอบกลับของจินตราที่ส่งกลับไปได้ (8-5(d)) สุวนันท์ก็จะได้กระแสบิตขึ้นมาชุดหนึ่งคือบิตที่ 1, 3, 7, 8, 10, 11, 12, และ 14 แต่จะเห็นได้ว่ามีเพียงบิตที่ 1, 3, 8, และ 12 เท่านั้นที่เป็นบิตที่ถูกต้อง ดังแสดงในรูป 8-5(g)

8.1.5 กฎพื้นฐานสองข้อสำหรับการใช้อักษรลับ

ต่อไปนี้จะกล่าวถึงกฎพื้นฐานที่สำคัญสองประการของการเข้ารหัสข้อมูล

กฎพื้นฐานข้อแรก: ความซ้ำซ้อน

กฎพื้นฐานที่สำคัญข้อแรกคือข้อความที่ถูกเข้ารหัสนั้นจะต้องประกอบด้วยข้อมูลซ้ำซ้อน (redundancy) นั่นคือ ข้อมูลที่ไม่มีความจำเป็นจะต้องใช้ในการทำความเข้าใจข่าวสารที่ส่งออกไป ตัวอย่างเช่น บริษัทที่ขายสินค้าทางไปรษณีย์แห่งหนึ่งมีสินค้าอยู่ 60,000 ชนิด โปรแกรมเมอร์ของบริษัทนี้ออกแบบระบบการสั่งซื้อสินค้าโดยกำหนดให้ชื่อของลูกค้ามีความยาว 16 ไบท์ ตามด้วยข้อมูลอีก 3 ไบท์ (1 ไบท์สำหรับการแจ้งปริมาณที่ต้องการและอีก 2 ไบท์สำหรับบอกหมายเลขของสินค้าที่ต้องการ) ข้อมูล 3 ไบท์สุดท้ายถูกเข้ารหัสด้วยคีย์ที่มีความยาวมากที่รู้เฉพาะลูกค้าและบริษัทแห่งนี้เท่านั้น

ดูเหมือนว่าเป็นวิธีการส่งข้อมูลที่มีความปลอดภัยมากวิธีหนึ่ง เนื่องจากผู้บุกรุก (passive intruder) จะไม่สามารถถอดรหัสข้อมูลนี้ได้ อย่างไรก็ตาม วิธีการนี้ก็มีข้อบกพร่องที่ทำให้กลายเป็นวิธีการที่ใช้ไม่ได้ สมมุติว่ามีพนักงานคนหนึ่งที่ถูกไล่ออกจากบริษัทฯ ซึ่งต้องการแก้แค้น จึงได้นำรายชื่อลูกค้าติดตัวไปด้วย คนผู้นี้จึงสร้างรายการสั่งซื้อสินค้าปลอมขึ้นมาโดยใช้หมายเลขลูกค้าจริง แต่เนื่องจากเขาไม่มีรายการคีย์ของลูกค้าจึงใช้วิธีการใส่เลขเดาสุ่มเข้าไปที่ 3 ไบท์สุดท้าย แล้วจัดการส่งรายการสั่งซื้อเหล่านี้ไปยังบริษัทฯ

เมื่อรายการสั่งซื้อเข้ามายังบริษัทฯ โปรแกรมของบริษัทจึงใช้หมายเลขลูกค้าในการค้นหาคีย์ของลูกค้าและพยายามถอดรหัสข้อมูลซึ่งรหัสบางส่วนอาจจะสามารถถอดรหัสออกมาได้แม้ว่าจะกลายเป็นตัวเลขที่ไม่สมเหตุสมผล เช่น ลูกค้าคนหนึ่งสั่งซื้อกล่องทราย 837 กล่อง อีกคนหนึ่งสั่งซื้อม้าหมุน 540 ชุด เป็นต้น แต่คอมพิวเตอร์ก็ไม่สนใจและสร้างรายการสั่งผลิตสินค้าขึ้นมา ด้วยวิธีการนี้ผู้บุกรุก (active intruder) คืออดีตพนักงานสามารถสร้างความเสียหายให้แก่บริษัทฯ ได้มากมีเดียว

ปัญหานี้สามารถแก้ไขได้โดยการใส่ข้อมูลซ้ำซ้อนเข้าไปในข้อความ เช่น รายการสั่งซื้อสินค้าอาจถูกขยายไปเป็น 12 ไบท์โดยที่ 9 ไบท์แรกอาจเป็น "0" ทั้งหมด ด้วยวิธีการนี้ผู้บุกรุกจะไม่สามารถเดารหัสที่ถูกต้องของข้อมูลทั้ง 12 ไบท์ได้ ตัวอย่างนี้สรุปได้ว่าข้อมูลจริงควรมีการเพิ่มเติมข้อมูลซ้ำซ้อนเข้าไปด้วยเพื่อป้องกันการเดาสุ่มแล้วกลายเป็นข้อมูลที่ถูกต้องตามเงื่อนไข อย่างไรก็ตาม การเพิ่มข้อมูลซ้ำซ้อนเปิดโอกาสให้นักถอดรหัสสามารถทำงานได้ง่ายยิ่งขึ้น ดังนั้นจะได้กฎพื้นฐานเกี่ยวกับการ

เข้ารหัสข้อมูลคือ ข้อความจะต้องประกอบด้วยข้อมูลซ้ำซ้อนบางส่วน

กฎพื้นฐานข้อที่สอง: ความสดใหม่ของข้อความ

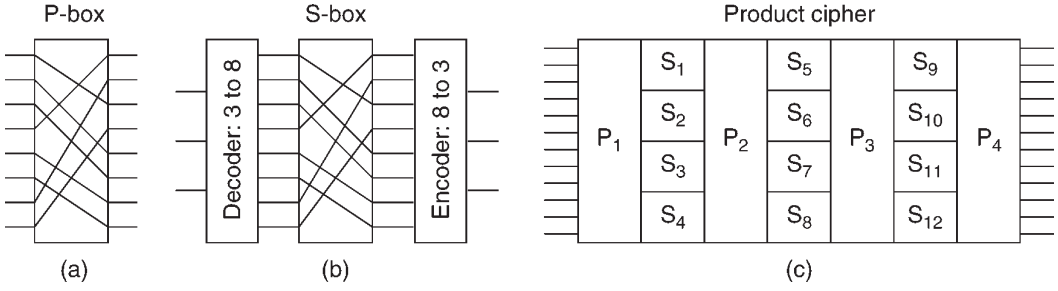
กฎพื้นฐานข้อที่สองกล่าวว่า จะต้องมีมาตรการบางอย่างเพื่อให้แน่ใจได้ว่าข้อความที่ได้รับมาในแต่ละครั้งนั้นสามารถถูกตรวจสอบได้ว่าเป็นข้อความที่สดใหม่ (freshness) ที่พึ่งส่งออกมา วิธีการนี้เป็นการป้องกันไม่ให้ผู้บุกรุกนำข้อมูลเก่าส่งกลับเข้ามาในระบบใหม่ เช่น พนักงานเก่าของบริษัทที่ยกตัวอย่างมานั้นอาจกระทำเพียงแค่ดักจับสัญญาณข้อความที่ลูกค้าส่งเข้ามาสู่บริษัทฯ จริง แต่ได้นำข้อความเหล่านั้นส่งเข้าสู่บริษัทฯ ซ้ำหลายๆ ครั้ง

8.2 อัลกอริทึมที่ใช้คีย์แบบสมมาตร

การเข้ารหัสข้อมูลในยุคใหม่ยังคงใช้แนวความคิดพื้นฐานของการเข้ารหัสข้อมูลแบบเก่าอยู่ (เช่น การเปลี่ยนตำแหน่ง หรือการแทนที่) แต่ได้ให้ความสำคัญในจุดที่แตกต่างออกไป โดยปกติผู้ทำการเข้ารหัสข้อมูลจะใช้อัลกอริทึมแบบง่าย แต่ในปัจจุบันได้ใช้อัลกอริทึมที่มีความสลับซับซ้อนมากเพื่อให้แน่ใจว่าแม้ว่าผู้บุกรุกจะได้รับข้อมูลเป็นจำนวนมากก็จะไม่สามารถถอดรหัสได้โดยไม่มีคีย์

วิธีการเข้ารหัสข้อมูลแบบแรกที่จะกล่าวถึงต่อไปเรียกว่า อัลกอริทึมแบบคีย์สมมาตร (symmetric-key algorithm) เนื่องจากเป็นวิธีการที่ใช้คีย์ตัวเดียวกันในการเข้ารหัสและถอดรหัส รูป 8-2 แสดงการใช้อัลกอริทึมแบบคีย์สมมาตร ในที่นี้จะใช้วิธีเรียกว่า block ciphers ซึ่งจะนำ plaintext มาทีละบล็อค (ขนาด n-bit) มาเปลี่ยนรูปแบบโดยใช้คีย์กลายเป็น ciphertext หนึ่งบล็อค (ขนาด n-bit เท่ากัน)

อัลกอริทึมสำหรับการเข้ารหัสข้อมูลสามารถสร้างขึ้นมาใช้งานได้โดยใช้ฮาร์ดแวร์ (เพื่อวัตถุประสงค์ในด้านความเร็วในการทำงาน) หรือซอฟต์แวร์ (เพื่อผลทางด้านความอ่อนตัว) แม้ว่าการอธิบายอัลกอริทึมและโพรโตคอลในหนังสือเล่มนี้จะเป็นอิสระจากการสร้างขึ้นมาใช้งานจริงแต่ก็จะกล่าวถึงการสร้างใช้งานบ้างเล็กน้อย การเปลี่ยนตำแหน่งและการแทนที่ตัวอักษรสามารถสร้างขึ้นมาได้โดยการใช้วงจรไฟฟ้าแบบง่ายๆ รูป 8-6(a) แสดงภาพอุปกรณ์เรียกว่า P-box (P นั้นย่อมาจากคำว่า Permutation) ซึ่งนำมาใช้ในการเปลี่ยนตำแหน่งของข้อมูล (Transposition) ที่ส่งเข้ามาขนาด 8 บิต ถ้ากำหนดหมายเลขให้แก่ข้อมูลนำเข้าขนาด 8 บิตนี้เป็น 01234567 แล้ว P-box นี้จะเปลี่ยนตำแหน่งข้อมูลเป็น 36071245 การสร้างแผงวงจรไฟฟ้าภายในอย่างเหมาะสม จะทำให้ P-box นี้สามารถเปลี่ยนตำแหน่งบิตข้อมูลได้ด้วยความเร็วสูงสุด (ประมาณความเร็วแสง) เนื่องจากไม่มีการคำนวณใดๆ เกิดขึ้นเลย



รูปที่ 8-6 อุปกรณ์เข้ารหัสข้อมูลแบบพื้นฐาน (a) P-box (b) S-box (c) Product

อุปกรณ์เรียกว่า S-box นำมาใช้ในการแทนที่ข้อมูล (substitution) ดังแสดงในรูป 8-6(b) ในตัวอย่างนี้ ข้อมูล plaintext ขนาด 3 บิตถูกส่งเข้าสู่อุปกรณ์ซึ่งจะส่งข้อมูล cipherttext ขนาด 3 บิตออกมา ข้อมูล 3 บิตที่ถูกส่งเข้าไปจะเลือกสายสัญญาณหนึ่งในแปดเส้นเป็นทางออกจากการทำงานขั้นแรกซึ่งจะกำหนดให้เป็นบิต "1" ส่วนที่เหลือจะเป็นบิต "0" ในขั้นตอนที่สองจะเป็น P-box ตัวหนึ่งซึ่งจะทำหน้าที่เปลี่ยนตำแหน่งข้อมูล และในขั้นตอนที่สามจะเป็นการเข้าโค้ดออกไปเป็นสายสัญญาณ 3 เส้น เท่ากับทางขาเข้า

สิ่งที่แสดงให้เห็นถึงพลังอำนาจของอุปกรณ์พื้นฐานเหล่านี้เกิดขึ้นเมื่อนำอุปกรณ์หลายตัวมาเชื่อมต่อเข้าด้วยกันเป็นอุปกรณ์เรียกว่า product cipher ดังแสดงในรูป 8-6(c) ในตัวอย่างนี้ ข้อมูลขาเข้าเป็นสายสัญญาณ 12 เส้นจะถูกเปลี่ยนตำแหน่งโดยอุปกรณ์ในขั้นตอนแรก (P1) ในทางทฤษฎีมีความเป็นไปได้ที่จะมีอุปกรณ์ S-box หลายตัวประกอบกันเป็นขั้นตอนที่สองซึ่งจะแทนที่ข้อมูลทั้ง 12 บิต ด้วยตัวเลขขนาด 12 บิต ซึ่งจะถูกแบ่งออกเป็นตัวเลขกลุ่มละ 3 บิตจำนวน 4 กลุ่ม และด้วยการเพิ่มจำนวนขั้นตอนในอุปกรณ์ตัวนี้ให้มีหลายขั้นตอน ก็จะทำให้อุปกรณ์ตัวนี้มีความซับซ้อนเป็นอย่างมาก

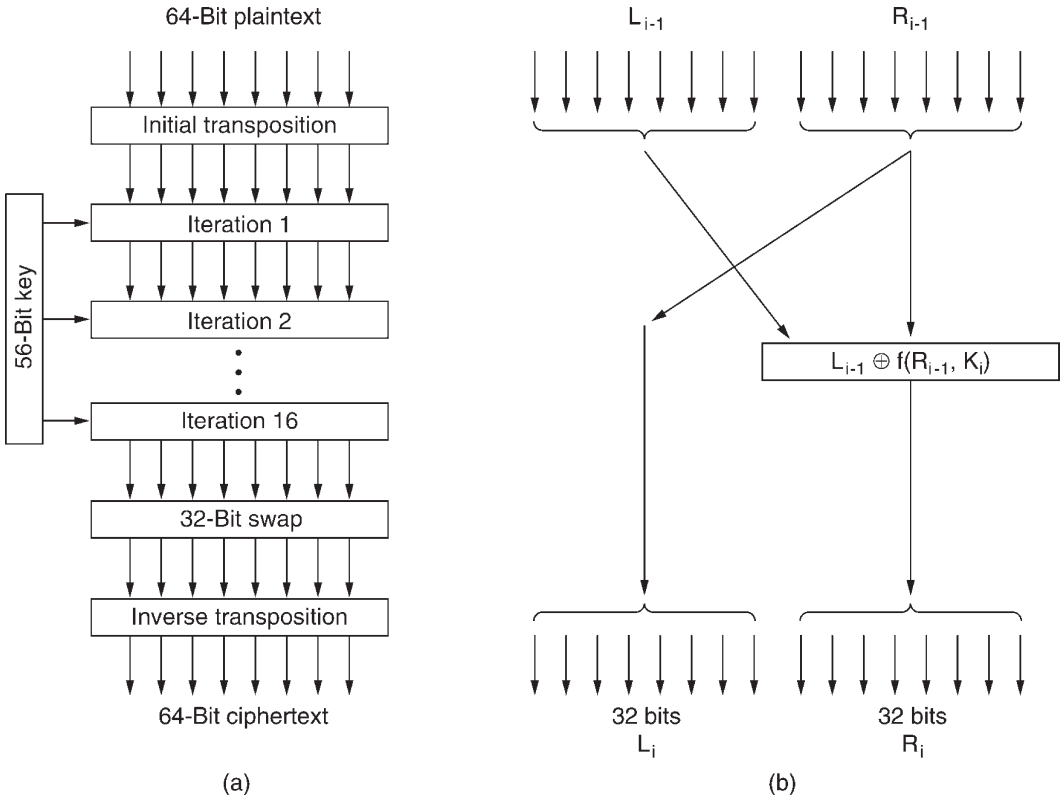
โดยทั่วไป product cipher จะมีสายสัญญาณขาเข้าและขาออกเป็นจำนวนข้างละ 64 ถึง 256 เส้น ตัวอุปกรณ์เองจะแบ่งออกเป็นประมาณ 18 ขั้นตอนแทนที่จะเป็นเพียง 7 ขั้นตอนตามที่แสดงในรูป 8-6(c) การสร้างด้วยซอฟต์แวร์สามารถทำได้โดยการสร้างโปรแกรมแบบวนซ้ำจำนวน 8 รอบเป็นอย่างน้อยโดยแต่ละรอบจะทำหน้าที่เสมือน S-box หนึ่งตัวซึ่งจะรอรับข้อมูลนำเข้าเป็นบล็อคๆ ละ 64 ถึง 256 บิต

8.2.1 มาตรฐานการเข้ารหัสข้อมูลแบบ DES

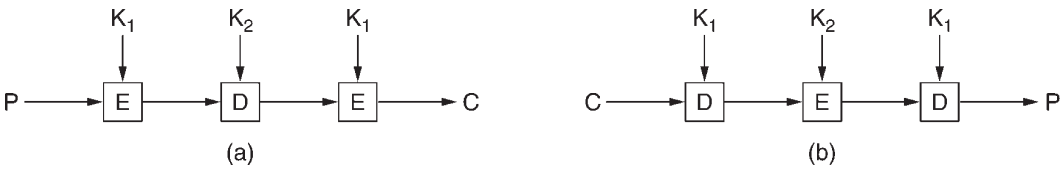
ในราวเดือนมกราคม พ.ศ. 2520 รัฐบาลของประเทศสหรัฐอเมริกาได้นำ product cipher ซึ่งได้รับการพัฒนาโดยบริษัทไอบีเอ็มไปใช้เป็นมาตรฐานในการเข้ารหัสข้อมูลประเภทไม่มีระดับความสำคัญ (Unclassified) มาตรฐานนี้เรียกว่า DES (Data Encryption Standard) ยังได้ถูกนำไปใช้งานอย่างกว้างขวางในวงการอุตสาหกรรมสำหรับข้อมูลที่ต้องการรักษาความลับ แม้ว่าตัววิธีการต้นแบบจะไม่สามารถรักษาความลับของข้อมูลได้อีกต่อไป แต่วิธีการที่ได้รับการปรับแต่งแล้วก็ยังคงสามารถนำมาใช้งานได้อย่างดี

รูป 8-7(a) แสดงภาพโครงสร้างการทำงานของวิธี DES ข้อมูล plaintext จะถูกเข้ารหัสเป็นบล็อคขนาด 64 บิตซึ่งจะสร้าง ciphertext ขนาด 64 บิตขึ้นมา อัลกอริทึมนี้ถูกควบคุมโดยข้อมูลคีย์ขนาด 56 บิต และแบ่งการทำงานออกเป็น 19 ขั้นตอน ขั้นตอนแรกจะทำการเปลี่ยนตำแหน่งข้อมูล 64 บิต (ทำงานเป็นอิสระจากคีย์) ซึ่งถูกเปลี่ยนตำแหน่งย้อนกลับในขั้นตอนสุดท้าย ขั้นตอนก่อนสุดท้ายจะสลับข้อมูล 32 บิตแรกกับ 32 บิตหลัง ขั้นตอนที่เหลืออีก 16 ขั้นตอนทำหน้าที่เหมือนกันหมดแต่ถูกควบคุมให้มีความแตกต่างกันด้วยคีย์ อัลกอริทึมนี้ได้รับการออกแบบให้ทำการถอดรหัสโดยใช้คีย์ตัวเดียวกับที่ใช้ในการเข้ารหัสแต่ใช้ขั้นตอนที่ย้อนกลับกัน

รูป 8-7(b) แสดงรายละเอียดการทำงานในหนึ่งวงรอบการทำงานของแต่ละขั้นตอน (ในกลุ่ม 16 ขั้นตอนตรงกลาง) ข้อมูลที่ส่งเข้ามาและส่งออกไป จะถูกแบ่งออกเป็นสองกลุ่ม กลุ่มละ 32 บิต ข้อมูลกลุ่ม 32 บิตที่ส่งออกทางด้านซ้ายเป็นสำเนาของข้อมูลกลุ่ม 32 บิตทางขวา ส่วนข้อมูลกลุ่ม 32 บิตที่ส่ง



รูปที่ 8-7
การเข้ารหัสข้อมูลแบบ DES
(a) โครงสร้างทั่วไป
(b) รายละเอียดของหนึ่งวงรอบการทำงาน



รูปที่ 8-8
(a) Triple encryption โดยใช้ DES
(b) การถอดรหัส

ออกทางด้านขวาเกิดจากการนำข้อมูลนำเข้าของกลุ่ม 32 บิตด้านซ้ายมาทำการ exclusive-OR เข้ากับข้อมูลนำเข้าของกลุ่ม 32 บิตด้านขวาและคีย์ของขั้นตอนนั้น (Ki) ซึ่งความซับซ้อนของอัลกอริทึมนี้ขึ้นอยู่กับการนำคีย์เข้ามากระทำกับข้อมูลในกลุ่มนี้

ในช่วงต้น พ.ศ. 2522 บริษัทไอบีเอ็มได้ตระหนักว่าคีย์ขนาด 56 บิตนั้นอาจสั้นเกินไปจึงได้คิดค้นวิธีการเพิ่มขนาดของคีย์อย่างมีประสิทธิภาพโดยการใช้เทคนิคเรียกว่า triple encryption ดังแสดงในรูป 8-8 วิธีการนี้แบ่งออกเป็นสามขั้นตอนและใช้คีย์สองตัว ขั้นตอนแรก plaintext จะถูกเข้ารหัสโดยการใช้วิธี DES แบบปกติซึ่งจะนำคีย์ K1 ไปใช้ ในขั้นตอนที่สอง DES จะทำงานในลักษณะของการถอดรหัสแต่ใช้คีย์ K2 และขั้นตอนสุดท้ายจะนำวิธี DES มาใช้อีกครั้งหนึ่งโดยใช้คีย์ K1

8.2.2 มาตรฐานการเข้ารหัสข้อมูลขั้นสูงแบบ AES

มาตรฐานการเข้ารหัสข้อมูลขั้นสูงแบบ AES (Advanced Encryption Standard) ได้รับการกระตุ้นให้เกิดการพัฒนาขึ้นมาโดยหน่วยงาน NIST (National Institute of Standards and Technology) ของกระทรวงพาณิชย์สหรัฐอเมริกาเพื่อนำมาใช้ในการเข้ารหัสข้อมูลประเภทไม่มีชั้นความลับแทนวิธีเข้ารหัสแบบ DES โดยได้จัดให้มีการแข่งขันในการนำเสนอวิธีการเข้ารหัสแบบใหม่ขึ้นในเดือนมกราคม พ.ศ. 2540 จากนักวิจัยทั่วโลก กติกาการแข่งขันได้กำหนดให้วิธีการที่นำเสนอมีข้อกำหนดดังนี้

1. ใช้วิธีการเข้ารหัส block cipher แบบสมมาตร

2. รายละเอียดของวิธีการที่นำเสนอจะต้องประกาศให้เป็นวิธีการแบบเปิดเผย

3. สนับสนุนการใช้คีย์ความยาว 128, 192, และ 256 บิต

4. จะต้องสามารถสร้างขึ้นใช้งานได้ทั้งวิธีซอฟต์แวร์และฮาร์ดแวร์

5. อัลกอริทึมที่ใช้จะต้องเปิดเผยสู่สาธารณะหรือมีใบอนุญาตให้ใช้งานโดยไม่มีการตั้งข้อรังเกียจ

ในราวเดือนสิงหาคม พ.ศ. 2541 NIST ได้คัดเลือกวิธีการ 5 แบบที่คิดว่าดีที่สุดในแง่ของความปลอดภัย ประสิทธิภาพ ความง่าย ความอ่อนตัว และความต้องการใช้หน่วยความจำในขณะทำงาน ในเดือนตุลาคม พ.ศ. 2544 NIST ได้ประกาศให้วิธีการของ Rijndael (คิดค้นโดย Rijmen และ Daemen ชาวเบลเยี่ยม) เป็นมาตรฐานการเข้ารหัสข้อมูลของรัฐบาลสหรัฐอเมริกา FIPS 197 (Federal Information Processing Standard)

Rijndael (อ่านว่า ไรน์-โดล) สนับสนุนคีย์ที่มีความยาว 128 ถึง 256 บิต (เพิ่มขึ้นครั้งละ 32 บิต) ความยาวของคีย์และขนาดของบล็อคเป็นอิสระแก่กันและกัน แต่โดยทั่วไปจะกำหนดให้มีสองแบบคือ ใช้บล็อคขนาด 128 บิตและคีย์ขนาด 128 บิต และใช้บล็อคขนาด 128 บิตและคีย์ขนาด 256 บิต ในกรณีที่ใช้คีย์ขนาด 128 บิตจะทำให้มีความเป็นไปได้ของคีย์เป็นจำนวน 2^{128} หรือประมาณ $3 * 10^{38}$ แบบ ซึ่งถ้าใช้เครื่องคอมพิวเตอร์ที่มีซีพียูจำนวน 1 พันล้านตัวที่ทำงานแบบขนานและการคำนวณแต่ละครั้งใช้เวลา 1 ส่วนล้านล้านวินาทีแล้ว จะต้องใช้เวลาในการคำนวณหาคีย์ทั้งสิ้น 10^{10} ปี

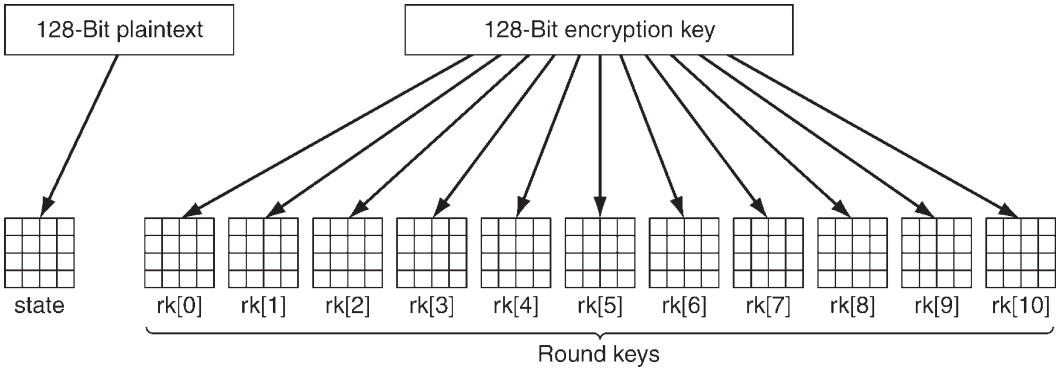
วิธีการแบบ Rijndael

วิธีการแบบ Rijndael สร้างขึ้นมาบนพื้นฐานของทฤษฎี Galois field theory และใช้ทั้งการแทน

```
#define LENGTH 16                       /* # bytes in data block or key */
#define NROWS 4                         /* number of rows in state */
#define NCOLS 4                         /* number of columns in state */
#define ROUNDS 10                       /* number of iterations */
typedef unsigned char byte;             /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
  int r;                                /* loop index */
  byte state[NROWS][NCOLS];             /* current state */
  struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];     /* round keys */

  expand_key(key, rk);                  /* construct the round keys */
  copy_plaintext_to_state(state, plaintext);    /* init current state */
  xor_roundkey_into_state(state, rk[0]);        /* XOR key into state */

  for (r = 1; r <= ROUNDS; r++) {
      substitute(state);                /* apply S-box to each byte */
      rotate_rows(state);               /* rotate row i by i bytes */
      if (r < ROUNDS) mix_columns(state);       /* mix function */
      xor_roundkey_into_state(state, rk[r]);    /* XOR key into state */
  }
  copy_state_to_ciphertext(ciphertext, state);  /* return result */
}
```

รูปที่ 8-9
โครงสร้างของโค้ด
สำหรับวิธี Rijndael

รูปที่ 8-10 การสร้างอาเรย์ state และอาเรย์ rk[i]



ตัวอักษรและการสลับตำแหน่งข้อมูลแบบ permutation รวมทั้งการทำงานแบบวนซ้ำที่จะวนเป็นจำนวน 10 รอบสำหรับบล็อคขนาด 128 บิตและคีย์ขนาด 128 บิต ขึ้นไปจนถึง 14 รอบสำหรับบล็อคที่มีขนาดใหญ่ที่สุดหรือใช้คีย์ขนาด 256 บิต รูป 8-9 แสดงโครงสร้างของโค้ดสำหรับวิธี Rijndael

ฟังก์ชัน Rijndael ประกอบด้วยพารามิเตอร์สามตัว plaintext หมายถึงอาเรย์ขนาด 16 ไบต์ที่ใช้เก็บข้อมูลที่จะทำการเข้ารหัส ciphertext อาเรย์ขนาด 16 ไบต์ที่ใช้เก็บข้อมูลที่ผ่านการเข้ารหัสแล้วและส่งออกเป็นผลลัพธ์ และ Key หมายถึงคีย์ขนาด 16 ไบต์ (128 บิต)

ตัวแปร state เป็นอาเรย์ที่กำหนดค่าเริ่มต้นให้เป็น plaintext แล้วจึงถูกเปลี่ยนค่าไปเรื่อยๆ ในทุกขั้นตอนการคำนวณ ในบางขั้นตอนอาจเกิดการแทนที่ครั้งละหนึ่งไบท์ ส่วนในขั้นตอนอื่นๆ อาจมีการสลับตำแหน่งข้อมูลเกิดขึ้น เมื่อผ่านขั้นตอนสุดท้าย ค่าในตัวแปร state นี้ก็คือข้อมูลที่ถูกเข้ารหัสเรียบร้อยแล้ว (ciphertext) รูป 8-10 แสดงให้เห็นการแบ่งข้อมูลออกเป็นส่วนเล็กๆ ซึ่งเป็นส่วนที่เกิดขึ้นในขั้นตอนแรก

เนื่องจากการทำงานในทุกขั้นตอนนั้นสามารถทำงานย้อนกลับได้ ดังนั้นในการถอดรหัสข้อมูลจึงนำอัลกอริทึมนี้มาทำงานย้อนกลับ

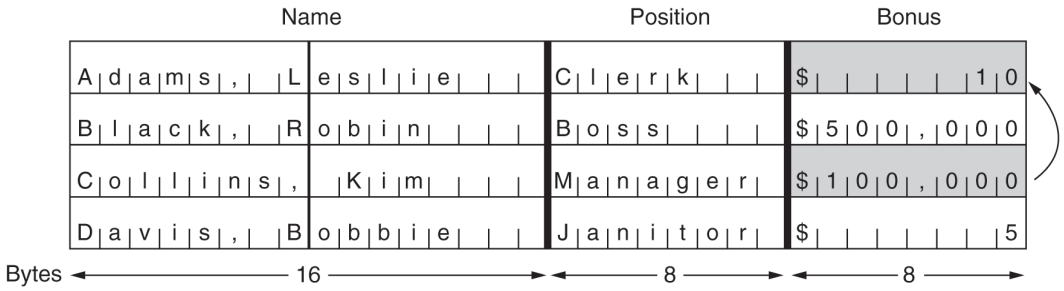
อัลกอริทึมนี้ได้รับการออกแบบมาเป็นอย่างดีจึงไม่เพียงแต่ให้ความปลอดภัยเป็นอย่างดีเท่านั้น แต่ยังสามารถทำงานได้อย่างรวดเร็ว ด้วยเครื่องคอมพิวเตอร์ที่ทำงานด้วยความเร็วอย่างน้อย 2 GHz จะสามารถเข้ารหัสข้อมูลได้ด้วยความเร็วประมาณ 700 Mbps ซึ่งเร็วพอที่จะเข้ารหัสข้อมูลวิดีทัศน์ MPEG-2 จำนวน 100 เรื่องได้แบบ real-time และถ้าสร้างวิธีนี้ด้วยฮาร์ดแวร์แล้วก็จะยิ่งเพิ่มความเร็วขึ้นไปอีก

8.2.3 การใช้ซิเฟอร์ในโหมดต่างๆ

ถ้าไม่พิจารณาถึงความซับซ้อนที่เกิดขึ้นแล้ว ทั้งวิธี AES และ DES นั้นมีการทำงานพื้นฐานเป็นการแทนที่ตัวอักษรเดี่ยว (monoaphabetic substitution) โดยใช้ตัวอักษรขนาดใหญ่ (128 บิตสำหรับ AES และ 64 บิตสำหรับ DES) เมื่อใส่ plaintext เป็นข้อมูลนำเข้าตัวเดิม ก็จะได้รับ ciphertext เป็นตัวเดิมเสมอ (และใช้คีย์ตัวเดิม) ดังนั้นผู้บุกรุกอาจใช้ข้อจำกัดนี้ในการพยายามถอดรหัสได้

Electronic Code Book Mode

ต่อไปจะแสดงให้เห็นว่าคุณสมบัติการแทนที่ตัวอักษรเดี่ยวสามารถถูกแก้ไขได้เป็นบางส่วนดังนี้ ในที่นี้จะใช้ triple DES เพราะเป็นการง่ายที่จะเดาบล็อคขนาด 64 บิตได้ง่ายกว่าบล็อคขนาด 128 บิต แต่ AES ก็มีปัญหาในลักษณะเดียวกัน วิธีการแบบตรงไปตรงมาในการใช้ DES เข้ารหัส plaintext ขนาดใหญ่คือการแบ่งข้อความนั้นออกเป็นบล็อคขนาด 8 ไบท์ (64 บิต) เรียงติดต่อกันและทำการเข้ารหัสที



รูปที่ 8-11 plaintext ของแฟ้มข้อมูลที่ถูกแบ่งออกเป็น DES บล็อคจำนวน 16 บล็อค

ละบล็อคเรียงตามลำดับโดยใช้คีย์ตัวเดียวกัน บล็อคสุดท้ายอาจมีขนาดน้อยกว่า 8 ไบท์ก็จะถูกเติมให้เต็มบล็อค (เช่น เติมด้วยตัวอักษรว่างหรือ space) วิธีการเช่นนี้เรียกว่า ECB mode (Electronic Code Book Mode)

ในรูป 8-11 ได้แสดงให้เห็นแฟ้มข้อมูลซึ่งเป็นรายการแจกเงินโบนัสประจำปีให้แก่พนักงาน แฟ้มข้อมูลนี้ประกอบด้วยระเบียนขนาด 32 ไบต์ (ต่อพนักงาน 1 คน) ซึ่งอยู่ในรูปแบบที่แสดงให้เห็นนั่นคือ ชื่อมีความยาว 16 ไบท์ ตำแหน่ง ขนาด 8 ไบต์ และเงินโบนัสอีก 8 ไบต์ แต่ละส่วน (จากทั้งหมดซึ่งเป็นข้อมูลขนาด 8 ไบต์จำนวน 16 ชิ้น ที่มีหมายเลข 0 ถึง 15) จะถูกนำมาเข้ารหัสด้วยวิธีการ tripple DES

สมมุติว่า Leslie (จากรูป 8-11) พึ่งจะมีข้อขัดแย้งกับเจ้านายจึงไม่คาดหวังว่าจะได้รับโบนัสมากนัก ในขณะที่ Kim เป็นที่โปรดปรานของเจ้านายซึ่งพนักงานทุกคนต่างก็ทราบดี Leslie สามารถเข้าไปดูแฟ้มข้อมูล (ที่ถูกเข้ารหัสแล้ว) นี้ได้ก่อนที่จะถูกส่งไปยังธนาคาร ปัญหาก็คือว่า Leslie จะสามารถแก้ไขแฟ้มข้อมูลนี้ได้หรือไม่

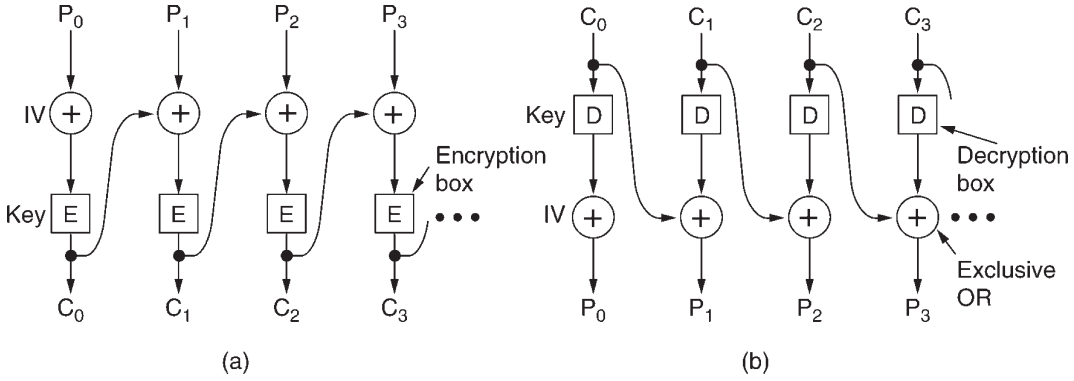
คำตอบก็คือได้แน่นอน Leslie เพียงสร้างสำเนาของบล็อคที่ 12 ใน ciphertext (ซึ่งเป็นข้อมูลตัวเลขโบนัสของ Kim) แล้วนำไปใส่แทนที่ข้อมูลบล็อคที่ 4 (ซึ่งเป็นข้อมูลตัวเลขโบนัสของ Leslie) เท่านั้น แม้ว่า Leslie จะไม่ทราบว่าข้อมูลที่ตนนำไปแทนที่นั้นเป็นอะไรแต่ก็สามารถคาดเดาได้ว่าคงเป็นตัวเลขโบนัสที่สูงกว่าที่ตนเองควรจะได้รับแน่นอน

Cipher Block Chaining Mode

เพื่อเป็นการป้องกันการแก้ไขข้อมูลในลักษณะที่กล่าวมานี้ ก็จะนำข้อมูลซิเฟอร์แต่ละบล็อคมาโยงเข้ากับข้อมูลในบล็อคต่อไปในหลายลักษณะ ดังนั้นถ้ามีการทำสำเนาข้อมูลในลักษณะที่ Leslie ทำ เมื่อถอดรหัสออกมาก็จะได้ข้อมูลที่ผิดพลาด หนทางหนึ่งในการโยงข้อมูลเข้าด้วยกันนี้เรียกว่า cipher block chaining ดังที่แสดงในรูป 8-12 ข้อมูล plaintext ในบล็อคหนึ่งจะถูกนำมาทำ exclusive-OR เข้ากับข้อมูล ciphertext ในบล็อคที่อยู่ก่อนหน้านั้นก่อนที่จะถูกนำไปเข้ารหัส ผลที่เกิดขึ้นก็คือข้อมูลใน plaintext บล็อคนั้นจะไม่ถูกแปลงเป็น ciphertext บล็อคอันเดิมและการเข้ารหัสก็จะไม่ใช่การแทนที่ตัวอักษรเดี่ยวเหมือนอย่างเดิมอีกต่อไป สำหรับข้อมูลในบล็อคแรกจะถูกทำ exclusive-OR กับข้อมูลในบล็อคที่ถูกเลือกขึ้นมาอย่างสุ่ม เรียกว่า Initialization Vector (IV) ซึ่งจะถูกส่งไปในรูป plaintext พร้อมกับข้อมูล ciphertext

รูป 8-12 แสดงตัวอย่างการทำงานของ cipher block chaining mode เริ่มต้นด้วยการคำนวณ C0 = E(P0 XOR IV) จากนั้นจึงคำนวณ C1 = E(P1 XOR C0) และบล็อคต่อๆ ไป การถอดรหัสก็

รูปที่ 8-12
Cipher block chaining
(a) การเข้ารหัส
(b) การถอดรหัส



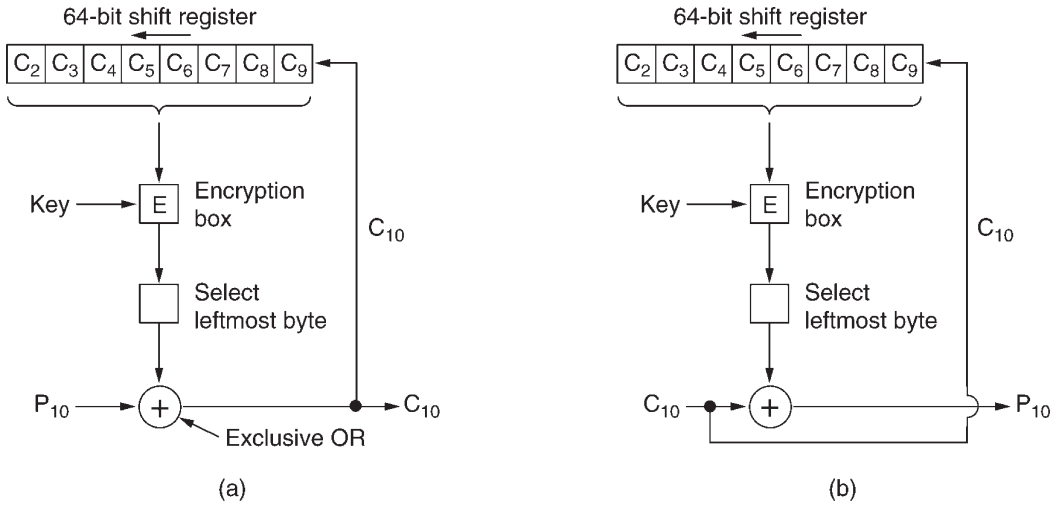
ทำได้โดยการใช้ฟังก์ชัน exclusive-OR ในลำดับที่สลับกัน นั่นคือ P0 = IV XOR D(C0) , P1 = C0 XOR D(C1) เป็นต้น สังเกตุว่าการเข้ารหัสของบล็อคที่ n จะได้มาจาก plaintext ในบล็อคที่ 0 ถึง บล็อคที่ n-1 ดังนั้นข้อมูล plaintext ที่แม้ว่าจะซ้ำกันก็จะถูกเข้ารหัสเป็น ciphertext ที่แตกต่างกัน ขึ้นอยู่กับตำแหน่งของ plaintext นั้นๆ การสร้างข้อมูลปลอมด้วยวิธีของ leslie นั้นจะทำให้เกิดข้อมูล ที่ไม่มีความหมายขึ้นสองบล็อค คือบล็อคที่เป็นโบนัสของ leslie และบล็อคที่อยู่ตามมา ซึ่งสำหรับ เจ้าหน้าที่รักษาความปลอดภัยแล้วนี่คือจุดเริ่มต้นที่ดีของการสอบสวนหาผู้กระทำผิด

Cipher block chaining มีข้อได้เปรียบที่ข้อความใน plaintext ที่เหมือนกันจะถูกเข้ารหัสแล้ว เป็นข้อความ ciphertext ที่ไม่เหมือนกันทำให้ความพยายามในการแก้ไขข้อมูล ciphertext ทำได้ยาก ขึ้นซึ่งก็เป็นเหตุผลหลักของการเลือกใช้วิธีนี้

Cipher Feedback mode

อย่างไรก็ตาม cipher block chaining มีข้อเสียที่ว่าจำเป็นต้องใช้ข้อมูลทั้งบล็อค (64 บิต) ที่ ถูกส่งมาก่อนหน้านั้นในการเริ่มต้นการถอดรหัส ทำให้การทำงานกับเทอร์มินอลที่มีการโต้ตอบแบบ ทันทีทันใดที่ผู้ใช้อาจพิมพ์ตัวอักษรเข้าไปสั้นกว่า 8 ตัวอักษรแล้วหยุดพิมพ์เพื่อรอรับการตอบสนองนั้น เป็นไปไม่ได้ สำหรับการเข้ารหัสที่ละไบต์จึงต้องหันมาใช้วิธีการ cipher feedback mode ร่วมกับ triple DES แทน ดังที่แสดงในรูป 8-13 สำหรับ AES ก็ใช้หลักการในทำนองเดียวกันเพียงแต่เปลี่ยนไป ใช้ 128-bit shift register แทน ในรูปนี้แสดงให้เห็นสถานะของเครื่องที่ทำการเข้ารหัสหลังจากที่ไบต์ที่ 0 ถึงไบต์ที่ 9 ได้ถูกเข้ารหัสและส่งออกไป เมื่อ plaintext ไบต์ที่ 10 มาถึงดังในรูป 8-13(a) อัลกอริทึม DES จะทำงานกับ 64-bit shift register เพื่อสร้าง 64-bit ciphertext ไบต์ที่อยู่ทางด้านซ้ายสุดของ ciphertext ถูกถอดรหัสมาจาก P10 ซึ่งไบต์นี้จะถูกส่งออกมาทางสายสัญญาณ นอกจากนี้ shift register จะถูกเลื่อนค่าไปทางซ้าย 8 บิต ทำให้ C2 หายไปทางด้านซ้ายและ C10 ถูกใส่เข้ามาในตำแหน่งที่ พึ่งจะว่างลงซึ่งอยู่ทางขวาของ C9 สังเกตุว่าค่าของ shift register จะขึ้นอยู่กับข้อมูลเก่าทั้งหมดของ plaintext ดังนั้น รูปแบบข้อมูลที่เกิดซ้ำซ้อนหลายครั้งจะถูกเข้ารหัสเป็น ciphertext ที่แตกต่างกัน เหมือนกับวิธี cipher block chaining ทั้งนี้ ยังคงมีความต้องการใช้ Initialization vector ในตอน เริ่มต้นการทำงาน

การถอดรหัสด้วย cipher feedback mode นั้นทำเช่นเดียวกับการเข้ารหัส นั่นคือ ค่าของ shift register จะถูกเข้ารหัสไม่ใช่ถูกถอดรหัส ดังนั้น ไบต์ที่ถูกเลือกมาทำ exclusive-OR กับ C10 เพื่อให้ได้



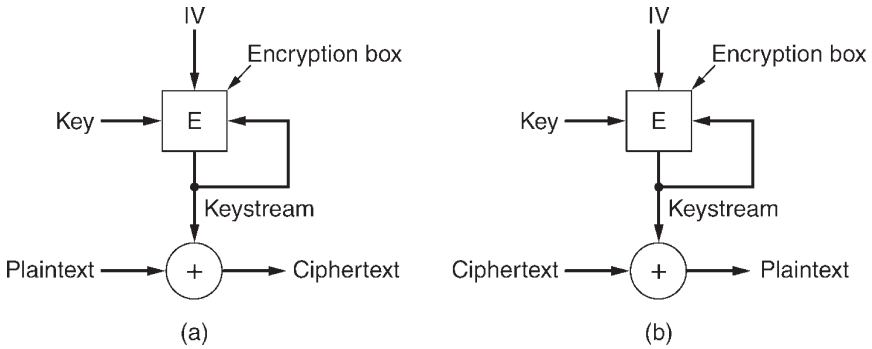
รูปที่ 8-13
Cipher feedback mode
(a) การเข้ารหัส
(b) การถอดรหัส

P10 นั้นเป็นอันเดียวกับที่ถูกทำ exclusive-OR กับ P10 เพื่อให้ได้ C10 ในตอนเริ่มต้น ตราบเท่าที่ shift register ทั้งสองตัวมีค่าเท่ากัน การถอดรหัสจะสามารถกระทำได้อย่างถูกต้อง ดังแสดงในรูป 8-13(b)

ปัญหาที่เกิดขึ้นกับ cipher feedback mode คือ ถ้าข้อมูลเพียง 1 บิตของ ciphertext นั้นถูกเปลี่ยนไปโดยอุบัติเหตุ ข้อมูลทั้ง 8 ไบต์ที่ถูกถอดรหัสโดยมีข้อมูลเสียอยู่ใน shift register นั้นจะกลายเป็นข้อมูลเสียทั้งหมด เมื่อไบต์ที่เสียถูกเลื่อนออกไปพ้น shift register แล้วจึงจะสามารถสร้าง plaintext ที่ถูกต้องได้ต่อไป ดังนั้น ผลกระทบของบิตที่เสียเพียง 1 บิตจะมีผลกระทบในวงจำกัดซึ่งจะไม่ส่งผลเสีย ไปยังข้อมูลส่วนอื่น อย่างไรก็ตาม ถ้า shift register มีขนาดใหญ่ขึ้นผลเสียก็จะเพิ่มขนาดขึ้นด้วย

Stream Cipher Mode

ไม่ว่าจะอย่างไรก็ตามโปรแกรมประยุกต์ที่มีบิตเสีย 1 บิตแล้วส่งผลให้ข้อมูลอื่นอีก 64 บิตเสียหายไปด้วยนั้นอาจไม่สามารถยอมรับได้ สำหรับโปรแกรมประยุกต์ประเภทนี้จึงต้องใช้วิธีการที่ 4 เรียกว่า stream cipher mode ซึ่งทำงานโดยการเข้ารหัส initialization vector ด้วยการใช้ key ในการดึงบล็อคที่ต้องการออกมา จากนั้น Output block จึงจะถูกนำไปเข้ารหัสด้วยการใช้ key เพื่อให้ได้ output บล็อคที่สองออกมา บล็อคนี้ก็จะถูกนำไปเข้ารหัสและนำไปใช้สร้างบล็อคที่สาม และวนต่อไปเรื่อยๆ ลำดับของ output block (ซึ่งมีขนาดใหญ่) เรียกว่า keystream ถูกนำมาใช้ในลักษณะเดียวกับ one-time pad และทำการ exclusive-OR เข้ากับ plaintext เพื่อให้ได้ ciphertext ดังแสดงในรูป 8-14(a)



รูปที่ 8-14
A stream cipher
(a) การเข้ารหัส
(b) การถอดรหัส

สังเกตว่า Initialization vector ถูกนำมาใช้ในขั้นตอนแรกเท่านั้น หลังจากนั้น output จะถูกเข้ารหัส และสังเกตว่า keystream นั้นเป็นอิสระจากข้อมูล ดังนั้น จึงสามารถทำการคำนวณล่วงหน้าได้ถ้าต้องการ และเป็นอิสระจากข้อผิดพลาดของข้อมูลที่อาจเกิดขึ้นในระหว่างการนำส่ง รูป 8-14(b) แสดงการถอดรหัส

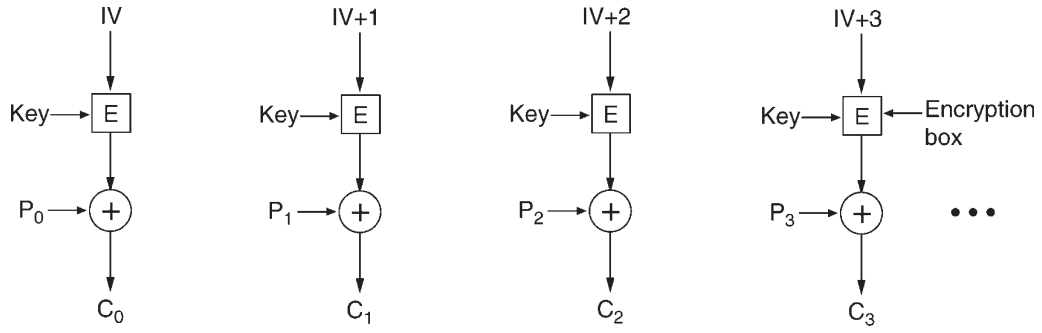
การถอดรหัสเกิดขึ้นโดยการสร้าง keystream ขึ้นด้วยวิธีเดียวกันทางด้านผู้รับข้อมูล เนื่องจาก keystream ขึ้นอยู่กับ Initialization vector และ key เท่านั้น จึงไม่มีผลกระทบเกิดขึ้นเนื่องจากความผิดพลาดในระหว่างการนำส่งข้อมูล ciphertext ดังนั้นปัญหาข้อมูลผิดพลาด 1 บิตจึงส่งผลกระทบต่อบิตนั้นเพียงบิตเดียว

ข้อสังเกตประการหนึ่งคือไม่ควรใช้ key และ IV คู่เดียวกันสองครั้งในการส่ง stream cipher เพราะการทำดังกล่าวจะสร้าง keystream ขึ้นมาเหมือนกันทั้งสองครั้ง ซึ่งจะเป็นการทำให้เกิดจุดอ่อนต่อการพยายามถอดรหัสโดยผู้บุกรุกได้ซึ่งเรียกปัญหานี้ว่า keystream reuse attack ลองนึกดูว่า plaintext บล็อก P0 ถูกเข้ารหัสด้วย keystream เพื่อให้ได้ P0 XOR K0 ต่อมา plaintext บล็อคที่สอง Q0 ก็ถูกเข้ารหัสด้วย keystream อันเดียวกันเพื่อให้ได้ Q0 XOR K0 ผู้บุกรุกที่สามารถสำเนาข้อมูลทั้งสองบล็อค นี้ไปได้ก็เพียงแต่นำข้อมูลทั้งสองมาทำ exclusive-OR กันก็จะได้ P0 XOR Q0 ซึ่งจะกำจัด key ออกไปได้ ผู้บุกรุกก็จะมีข้อมูล plaintext สองบล็อคที่ทำ exclusive-OR กันอยู่ ถ้าสามารถทราบหรือเดาได้ว่าหนึ่งในสองนั้นคืออะไรก็จะสามารถทราบข้อความที่สองได้ทันที ในเหตุการณ์ใดก็แล้วแต่ข้อความสองข้อความที่ถูกทำ exclusive-OR กันอยู่จะสามารถใช้คุณสมบัติทางด้านสถิติเข้ามาสืบหาข้อความทั้งสองนั้นได้ ตัวอย่างเช่น สำหรับข้อความภาษาอังกฤษ ตัวอักษรที่อาจจะเกิดการทำ exclusive-OR กันมากที่สุดก็คือ ตัวอักษรว่าง (space) จำนวน 2 ตัว หรือลำดับต่อมาก็คือตัวอักษร "e" กับตัวอักษรว่าง เป็นต้น

Counter mode

ปัญหาหนึ่งที่ mode ทั้งหลายยกเว้น electronic code book mode ประสบก็คือไม่สามารถทำการเข้ารหัสข้อมูลแบบสุ่มตำแหน่งได้ ตัวอย่างเช่น สมมุติว่ามีการส่งแฟ้มข้อมูลผ่านระบบเครือข่ายและถูกส่งมาเก็บไว้ในดิสก์ในรูปแบบของการเข้ารหัส ซึ่งเป็นวิธีการที่เหมาะสมที่จะนำมาใช้โดยเฉพาะเครื่องโน้ตบุ๊คที่อาจถูกขโมยไปได้โดยง่าย เพราะการจัดเก็บข้อมูลในรูปแบบการเข้ารหัสช่วยลดความเสียหายที่อาจจะเกิดขึ้นเนื่องจากข้อมูลที่เป็นความลับถูกขโมย

อย่างไรก็ตาม แฟ้มข้อมูลในดิสก์นั้นมักจะถูกนำไปใช้งานในลำดับที่ข้อมูลไม่ได้เรียงลำดับต่อกัน



รูปที่ 8-15 การเข้ารหัสด้วยวิธี counter mode

โดยเฉพาะข้อมูลในระบบฐานข้อมูล ด้วยแฟ้มข้อมูลที่เข้ารหัสด้วยวิธี cipher block chaining นั้น การถอดรหัสข้อมูลในบล็อคใดก็ตามจะเกิดขึ้นได้ก็ต่อเมื่อบล็อคที่อยู่ก่อนหน้านั้นทั้งหมดถูกถอดรหัสออกหมดแล้วเท่านั้น ซึ่งเป็นวิธีการที่ไม่เหมาะสมในการนำมาใช้งานกับฐานข้อมูล ด้วยเหตุผลนี้จึงมีวิธีการเข้ารหัสอีกวิธีหนึ่งเรียกว่า counter mode ดังแสดงในรูป 8-15 ด้วยวิธีการนี้ข้อมูล plaintext ไม่ได้ถูกเข้ารหัสโดยตรง นั่นคือ initialization vector และ ค่าคงที่ค่าหนึ่งจะถูกเข้ารหัส แล้วจึงนำ ciphertect มาทำ exclusive-OR กับ plaintext โดยการเพิ่มค่า initialization vector ทีละ 1 สำหรับแต่ละบล็อค ก็จะเป็นการง่ายในการถอดรหัสบล็อคใดก็ได้ในแฟ้มข้อมูลโดยไม่ต้องถอดรหัสบล็อคที่อยู่ก่อนหน้านั้นเลย

แม้ว่า counter mode จะเป็นวิธีการที่มีประโยชน์แต่ก็มีจุดอ่อนที่จะต้องนำมากล่าวถึง สมมุติว่า key ตัวเดียวกัน (K) ถูกนำมาใช้อีกครั้งหนึ่งในอนาคต (ใช้ plaintext ต่างกันแต่มี Initialization Vector ตัวเดียวกัน) และผู้บุกรุกได้ขโมยข้อมูลทั้งสองครั้งไปได้ ในกรณีนี้ค่าของ keystream จะเหมือนกันทั้งสองกรณีซึ่งเป็นจุดอ่อนสำหรับ keystream reuse attack แบบเดียวกับที่ได้อธิบายไปแล้ว นั่นคือผู้บุกรุกเพียงแต่นำข้อความ ciphertext ทั้งสองครั้งมาทำ exclusive-OR เข้าด้วยกันก็จะสามารถกำจัด key ที่ใช้ออกไปได้เหลืออยู่แต่เพียงข้อความ plaintext เท่านั้น จุดอ่อนนี้ไม่ได้ทำให้วิธีการ counter mode หมดความหมายไปเพียงแต่เป็นการเตือนให้เลือกใช้ initialization vector และ key ต่างกัน และควรจะเป็นการเลือกแบบสุ่ม แม้ว่า key ตัวเดียวกันอาจถูกเลือกใช้สองครั้งโดยบังเอิญแต่ถ้า initialization vector ต่างกันก็ยังสามารถรักษาความปลอดภัยไว้ได้

8.2.4 การใช้ซิเฟอร์ในแบบอื่นๆ

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

รูปที่ 8-16 อัลกอริทึมการเข้ารหัสแบบใช้คีย์สมดุลย์แบบต่างๆ

วิธีการแบบ DES และ Rijndael เป็นวิธีการเข้ารหัสสองแบบที่ใช้คีย์สมดุลย์ที่ได้รับความนิยมในการนำมาใช้งานมากที่สุด อย่างไรก็ตาม ยังมีวิธีการอีกมากมายที่ได้รับการพัฒนาขึ้นมาใช้งาน วิธีการเหล่านี้บางส่วนก็ได้รับการผสมผสานการใช้งานเข้ากับโปรแกรมประยุกต์ต่างๆ รูป 8-16 แสดงรายการวิธีที่น่าสนใจแบบอื่นๆ

8.2.5 การวิเคราะห์เอกสารที่เขียนด้วยอักษรลับ

หัวข้อนี้จะได้กล่าวถึงวิธีการ 4 วิธีที่นำมาใช้ในการวิเคราะห์เอกสารที่เขียนด้วยรหัสลับ วิธีการแรกเรียกว่า differential cryptanalysis วิธีการนี้สามารถนำมาใช้ในการถอดรหัสแบบ block cipher ได้ทุกชนิด ทำงานโดยเริ่มต้นด้วยการนำ plaintext บล็อคจำนวน 2 บล็อคที่มีความแตกต่างกันน้อยมากและเฝ้าสังเกตอย่างระมัดระวังว่าจะเกิดอะไรขึ้นเมื่อวงรอบในการเข้ารหัสเกิดขึ้น ในหลายกรณีรูปแบบของบิตบางส่วนจะมีลักษณะที่เหมือนกันซึ่งจะนำไปสู่วิธีการถอดรหัสแบบprobabilistic attack ได้

การพัฒนาวิธีการที่สองเรียกว่า linear cryptanalysis ทำงานโดยทำ exclusive-OR กับบิตบางบิตใน plaintext กับ ciphertext เข้าด้วยกันและทำการตรวจสอบรูปแบบของผลลัพธ์ที่ได้ เมื่อทำดังนี้ซ้ำแล้วซ้ำเล่าบิตจำนวนครึ่งหนึ่งควรจะเป็น "0" และอีกครึ่งหนึ่งเป็น "1" อย่างไรก็ตามซิเฟอร์จะมีค่า bias ไปสู่ทิศทางใดทิศทางหนึ่ง และถ้าค่า bias นี้เป็นค่าขนาดเล็กก็จะสามารถนำมาใช้ประโยชน์โดยการลดขนาดของงานที่ต้องทำให้ลดลง

วิธีการที่สามเป็นการใช้การวิเคราะห์ปริมาณไฟฟ้าที่ถูกใช้ไปเพื่อค้นหา secret key คอมพิวเตอร์มักใช้ไฟฟ้าขนาด 3 โวลท์เพื่อแทนบิต "1" และ 0 โวลท์เพื่อแทนบิต "0" ดังนั้นการประมวลผลข้อมูลที่เป็น "1" จึงใช้พลังงานไฟฟ้ามากกว่าการประมวลผลบิต "0" ถ้าอัลกอริทึมการเข้ารหัสประกอบด้วยการทำงานวนซ้ำที่มี key บิตทำการประมวลผลเป็นลำดับ ผู้บุกรุกสามารถเปลี่ยนสัญญาณนาฬิกาเป็นสัญญาณที่ช้าลง เช่น 100 Hz และทำการตรวจวัดปริมาณไฟฟ้าที่ซีพียูใช้ในการประมวลผลแต่ละคำสั่งได้ จากข้อมูลที่ได้นี้จะสามารถแปลงกลับมาเป็น key ได้อย่างง่ายดาย การป้องกันวิธีการถอดรหัสแบบนี้สามารถทำได้โดยการเขียนโปรแกรมอย่างระมัดระวังเพื่อให้แน่ใจว่าปริมาณไฟฟ้าที่ใช้นั้นเป็นอิสระจากค่าของ key

วิธีการที่สี่คือการวิเคราะห์ timing อัลกอริทึมเข้ารหัสข้อมูลนั้นเต็มไปด้วยประโยคคำสั่ง "if" ที่ทำการตรวจสอบค่าบิตใน key ถ้าส่วนคำสั่ง "then" และ "else" นั้นใช้เวลาในการทำงานไม่เท่ากันและใช้วิธีการทำให้สัญญาณนาฬิกาของซีพียูช้าลงเพื่อสังเกตระยะเวลาที่ใช้ในแต่ละขั้นตอนก็มีความเป็นไปได้ที่จะค้นพบ key ที่ต้องการ การวิเคราะห์ปริมาณพลังงานและระยะเวลาที่ใช้ในระหว่างการประมวลผล อาจนำมาใช้ร่วมกันเพื่อให้มีประสิทธิภาพมากยิ่งขึ้น แม้ว่าวิธีการนี้ดูเหมือนเป็นวิธีการที่ไม่มีผู้ใดนำมาใช้ แต่ในความเป็นจริงนั้นเป็นวิธีการที่มีประสิทธิภาพมากซึ่งสามารถถอดรหัส cipher ใดๆ ก็ได้ที่ไม่ได้รับการออกแบบมาให้ต้านทานต่อวิธีการนี้

8.3 อัลกอริทึมที่ใช้คีย์สาธารณะ

เท่าที่ผ่านมาในอดีต การแจกจ่ายคีย์นั้นเป็นจุดอ่อนที่สุดในระบบการเข้ารหัสทุกชนิด ไม่ว่าระบบการเข้ารหัสจะดีเพียงใดก็ตามถ้าผู้บุกรุกสามารถขโมยคีย์ไปได้แล้วระบบการเข้ารหัสนั้นก็ถือว่าล้มเหลว นักวิจัยในเรื่องเทคนิคการเข้ารหัสมักจะถือเสมอว่าคีย์สำหรับการเข้ารหัสจะต้องเป็นตัวเดียวกันที่นำมาใช้ในการถอดรหัส (หรือจะต้องสามารถสร้างมาจากอีกอันหนึ่งได้โดยง่าย) แต่คีย์นั้นจะต้องถูกแจกจ่ายไปยังทุกๆ คนที่เกี่ยวข้องกับการใช้ข้อมูลนั้นๆ ดังนั้น จึงดูเหมือนว่าเป็นปัญหาที่เกิดขึ้นมาพร้อมกับการใช้คีย์ คีย์จะต้องได้รับการป้องกันแต่ในขณะเดียวกันก็จะต้องถูกแจกจ่าย ดังนั้น จึงไม่สามารถเก็บคีย์ไว้ในตู้เซฟได้

ในปี ค.ศ. 1976 นักวิจัยสองท่านแห่งมหาวิทยาลัย Stanford University ชื่อ Diffie และ Hellman

ได้นำเสนอวิธีการเข้ารหัสข้อมูลแบบใหม่ที่แตกต่างไปจากเดิมโดยสิ้นเชิง นั่นคือ วิธีการเข้ารหัสที่ใช้คีย์ในการเข้ารหัสและถอดรหัสแตกต่างกัน และคีย์ที่ใช้ถอดรหัสนั้นไม่สามารถสร้างขึ้นมาจากคีย์ที่ใช้เข้ารหัสได้ ในวิธีการที่นำเสนอนั้น การเข้ารหัส (E) และถอดรหัส (D) จะต้องอยู่ในเงื่อนไขสามประการคือ

1. D(E(P)) = P
2. เป็นการยากมากที่จะสร้าง D ขึ้นมาจาก E
3. E จะต้องทนทานต่อวิธีการถอดรหัสแบบ chosen plaintext attack

ความต้องการประการแรกกล่าวว่าเมื่อใช้อัลกอริทึม D กับข้อมูลที่ถูกเข้ารหัส E(P) แล้วจะต้องได้ข้อความดั้งเดิม plaintext กลับออกมา ถ้าปราศจากคุณสมบัติข้อนี้ผู้รับที่มีสิทธิอันชอบธรรมจะไม่สามารถถอดรหัสข้อมูลที่ถูกต้องได้ ความต้องการประการที่สองอธิบายได้ในตัวเอง ความต้องการประการที่สามนั้นมีความจำเป็นเนื่องจากผู้บุกรุกอาจทำการทดลองใส่รหัสข้อความที่สร้างขึ้นมาเอง ซึ่งจะได้เห็นในโอกาสต่อไป ด้วยความต้องการทั้งสามข้อนี้จึงไม่มีเหตุผลใดๆ ที่จะต้องเก็บคีย์ที่ใช้ในการเข้ารหัสไว้เป็นความลับ

อัลกอริทึมนี้ทำงานดังนี้ คนคนหนึ่งให้ชื่อว่า อลิส ต้องการรับข้อความลับจึงสร้างอัลกอริทึมขึ้นมาสองอันที่ตรงตามความต้องการของคุณสมบัติสามข้อข้างต้น อัลกอริทึมที่ใช้เข้ารหัสข้อมูลและคีย์ของอลิสได้รับการแจกจ่ายทั่วไปจึงเรียกว่า การเข้ารหัสโดยใช้คีย์สาธารณะ (public-key cryptography) เช่น อลิสอาจใส่คีย์ของเธอไว้ในเว็บเพจของตนเองก็ได้ ทั้งนี้ ให้ใช้สัญลักษณ์เป็น EA หมายถึงอัลกอริทึมที่ใช้ในการเข้ารหัสพร้อมทั้งคีย์สาธารณะของอลิส ในทำนองเดียวกัน ให้ใช้สัญลักษณ์ DA หมายถึงอัลกอริทึมที่ใช้ในการถอดรหัสพร้อมทั้งคีย์ลับของอลิส บ๊อบก็ทำแบบเดียวกับอลิสคือแจกจ่าย EB แต่เก็บ DB ไว้เป็นความลับ

ต่อไปจะแสดงให้เห็นว่าทั้งบ๊อบและอลิสสามารถสร้างช่องสื่อสารลับระหว่างบุคคลทั้งสองขึ้นมาได้อย่างไร โดยที่คนทั้งสองไม่เคยพบปะกันมาก่อนเลย คีย์สำหรับการเข้ารหัสของทั้งอลิส (EA) และบ๊อบ (EB) ได้รับการเปิดเผยในที่สาธารณะ ต่อไปอลิสนำข้อความแผ่นแรก (P) มาทำการคำนวณ EB(P) แล้วจัดส่งไปยังบ๊อบ บ๊อบสามารถถอดรหัสได้โดยการใช้คีย์ลับของเขา DB ทำการคำนวณ DB(EB(P)) = P จะไม่มีผู้ใดสามารถอ่านข้อความที่เข้ารหัสแล้ว EB(P) ได้เพราะว่าวิธีการเข้ารหัสนั้นดีมากและเป็นการยากมากที่จะคำนวณหาค่า DB มาจากค่า EB ที่ทราบ ในการส่งข้อความตอบกลับไปยังอลิส บ๊อบก็เพียงแค่ส่งข้อความ EA(R) กลับไปเท่านั้น ทั้งอลิสและบ๊อบก็จะสามารถสื่อสารระหว่างกันได้โดยปลอดภัย

การเข้ารหัสแบบ public-key cryptography นั้น ผู้ใช้แต่ละคนจะต้องมีคีย์สองตัวคือ คีย์สาธารณะที่สามารถประกาศให้ทราบได้ทั่วไปสำหรับใช้ในการเข้ารหัสข้อมูล และคีย์ส่วนตัวซึ่งผู้ใช้ต้องมีไว้ใช้ในการถอดรหัสข้อความที่ได้รับมา ซึ่งจะเรียกว่า คีย์สาธารณะ (public key) และคีย์ส่วนตัว (private key) ทั้งนี้ จะไม่ใช้คำว่าคีย์ลับ (secret key) ซึ่งเป็นคำที่ใช้ในวิธีการเข้ารหัสแบบ symmetric-key cryptography

8.3.1 อัลกอริทึม RSA

เนื่องจากข้อได้เปรียบของการเข้ารหัสข้อมูลโดยใช้คีย์สาธารณะทำให้นักวิจัยจำนวนมากพยายามคิดค้นหาอัลแกอริทึมที่สามารถใช้งานจริงขึ้นมา คณะวิจัยที่ประสบผลสำเร็จจากมหาวิทยาลัย MIT โดย

Rivest, Shamir, และ Adleman ได้นำเสนออัลกิริทึมที่สอดคล้องกับคุณสมบัติทั้งสามข้อเรียกว่า RSA ปรากฏว่าในช่วง 25 ปีที่ผ่านมาไม่มีผู้ใดสามารถถอดรหัสวิธีการนี้ได้เลย (โดยที่ไม่ทราบคีย์) จึงถือได้ว่าเป็นวิธีการที่ดีมากวิธีหนึ่ง และทำให้วิธีการเข้ารหัสส่วนใหญ่นำวิธีการนี้ไปใช้ ข้อเสียของวิธีการนี้คือต้องใช้คีย์ขนาดไม่น้อยกว่า 1024 บิตจึงจะเป็นวิธีที่ปลอดภัย (เปรียบเทียบกับคีย์ขนาดเพียง 128 บิตสำหรับวิธี symmetric-key algorithm) ซึ่งทำให้ทำงานได้ช้า

วิธีการแบบ RSA นั้นนำพื้นฐานของทฤษฎีตัวเลขมาใช้ ขั้นตอนในการทำงานมีดังนี้

1. เลือกตัวเลขที่เป็น prime number (เลขที่ไม่มีเลขใดหารได้ลงตัวนอกจาก 1 กับตัวเอง) p และ q

2. คำนวณ n = p x q และ z = (p - 1) x (q - 1)

3. เลือกตัวเลขที่เป็น prime number เมื่อเทียบกับ z เรียกว่า d

4. หาค่า e ซึ่งมีค่าเท่ากับ e x d = 1 mod z

ด้วยตัวพารามิเตอร์เหล่านี้สามารถทำการคำนวณได้ล่วงหน้า (ก่อนทำการเข้ารหัส) จึงพร้อมที่จะทำการเข้ารหัสข้อมูลดังนี้ แบ่ง plaintext (ซึ่งในขณะนี้คือกระแสของบิต) ออกเป็นบล็อคขนาด k บิต โดยที่ k คือค่าเลขจำนวนเต็มที่มีค่ามากที่สุดที่ทำให้ 2k < n เมื่อ n คือขนาดของ plaintext P ที่นำ มาเข้ารหัส

เมื่อต้องการเข้ารหัสข้อความ P ให้ทำการคำนวณ C = Pe (mod n) เมื่อต้องการถอดรหัส C ให้คำนวณ P = Cd (mod n) สามารถพิสูจน์ให้เห็นได้ว่าสำหรับทุกๆ ค่าของ P ภายในช่วงที่กำหนด การเข้ารหัสและการถอดรหัสเป็นกระบวนการที่สวนทางกัน ในการเข้ารหัส ผู้ใช้ต้องการทราบค่าของ e และ n ส่วนการถอดรหัสจะต้องทราบค่า d และ n ดังนั้น คีย์สาธารณะจึงประกอบด้วย (e, n) และคีย์ส่วนตัวประกอบด้วย (d, n)

ความปลอดภัยของวิธีการนี้ขึ้นอยู่กับความยากของการหาค่าตัวประกอบร่วมของเลขจำนวนหนึ่ง ถ้าผู้บุกรุกสามารถหาค่าตัวประกอบของ n ได้ ก็จะสามารถหาค่า p และ q ได้จากค่า z ถ้าทราบค่า z, และ e จะสามารถคำนวณหาค่า d ได้จาก Euclids algorithm อย่างไรก็ตาม นักคณิตศาสตร์ได้พยายามหาวิธีหาค่าตัวประกอบมาเป็นระยะเวลานานกว่า 300 ปีมาแล้วและยังไม่สามารถหาวิธีทำได้

ตามการกล่าวอ้างของผู้คิดวิธี RSA การหาค่าประกอบของตัวเลขขนาด 500 หลักจะต้องใช้เวลาประมาณ 10^{25} ปีทำงานบนเครื่องคอมพิวเตอร์ที่สามารถคำนวณ 1 คำสั่งได้ในเวลา 1 ไมโครวินาที แม้ว่าเครื่องคอมพิวเตอร์จะได้รับการพัฒนาความเร็วขึ้นในระยะเวลาที่สั้นมาก แต่คงจะต้องรออีกเป็นระยะเวลากว่าหนึ่งศตวรรษที่การแยกตัวประกอบขนาด 500 หลักจะเป็นเรื่องที่เป็นไปได้ เมื่อถึงเวลานั้นสิ่งที่ต้องทำก็คือการเพิ่มขนาดของ p และ q ให้ใหญ่ขึ้นไปกว่าเดิม

รูป 8-17 แสดงตัวอย่างวิธีการทำงานของวิธี RSA ในตัวอย่างนี้กำหนดให้ p = 3 และ q = 11 ทำให้ค่า n = 33 และ z = 20 ค่าที่เหมาะสมของ d คือ d = 7 เนื่องจาก 7 และ 20 ไม่มีตัวหารร่วม จากนั้นทำการคำนวณหาค่า e ได้จาก 7e = 1 mod (20) ซึ่งจะได้ว่า e = 3 การเข้ารหัสทำได้จาก

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
		Sender's computation		Receiver's computation		

รูปที่ 8-17
ตัวอย่างการทำงาน
อัลกอริทึม RSA

การคำนวณ C = P3 (mod 33) ส่วนการถอดรหัสทำได้โดยารคำนวณ P = C7 (mod 33) รูปดังกล่าวแสดงการเข้ารหัสคำว่า "SUZANNE"

เนื่องจากการเลือกค่าตัวเลข prime number ในตัวอย่างนี้เป็นค่าน้อยมาก จึงทำให้ P ต้องมีค่าน้อยกว่า 33 แต่ละ plaintext บล็อคจึงมีค่าเพียงตัวอักษรตัวเดียวเท่านั้น ผลที่ได้จึงคล้ายกับวิธีการแทนที่ตัวอักษรตัวเดียว (mono-alphabetic substitution) ซึ่งไม่น่าประทับใจนัก แต่ถ้าเลือกค่า p และ q ประมาณ 2512 ก็จะได้ค่า n ประมาณ 21024 จะทำให้แต่ละบล็อคมีขนาด 1024 บิตหรือ 128 ตัวอักษรเมื่อเปรียบเทียบกับบล็อคละ 8 ตัวอักษรของ DES และบล็อคละ 16 ตัวอักษรสำหรับ AES

วิธีการแบบ RSA นั้นคล้ายกับวิธีการแบบ symmetric key algorithm เมื่อใช้ ECB mode ข้อมูลที่ทำการเข้ารหัสมีขนาดเท่ากับข้อมูลที่ถูกเข้ารหัสแล้ว ดังนั้น จะต้องนำวิธี chaining มาใช้ในการเข้ารหัสข้อมูลด้วย อย่างไรก็ตาม ในทางปฏิบัติวิธีการที่นำ RSA ไปใช้ส่วนใหญ่จะใช้การเข้ารหัสที่ใช้คีย์สาธารณะเพียงครั้งเดียวสำหรับการแจกจ่าย symmetric key algorithm เช่น AES หรือ triple DES เนื่องจาก RSA นั้นทำงานได้ช้ามากสำหรับการเข้ารหัสข้อมูลปริมาณมากๆ แต่ได้รับความนิยมในการนำมาใช้เป็นวิธีแจกจ่ายคีย์สำหรับใช้กับวิธีอื่น

8.3.2 อัลกอริทึมใช้คีย์สาธารณะแบบอื่น

แม้ว่าวิธี RSA ได้รับการนำไปใช้งานอย่างกว้างขวาง แต่ก็ไม่ใช่วิธีการเดียวที่มีใช้งานอยู่ในปัจจุบัน วิธีการแบบ public-key algorithm แบบแรกเรียกว่า knapsack algorithm (พัฒนาโดย Merkel และ Hellman ในปี ค.ศ. 1978) แนวความคิดเป็นดังนี้ คนผู้หนึ่งเป็นเจ้าของวัตถุจำนวนมากซึ่งวัตถุแต่ละชิ้นมีน้ำหนักที่แตกต่างกัน เจ้าของทำการเข้ารหัสข่าวสารโดยการเลือกวัตถุมาจำนวนหนึ่งจากที่มีอยู่ (อย่างเป็นความลับ) และใส่เข้าไปในถุง น้ำหนักโดยรวมของถุงนั้นและรายการของวัตถุทั้งหมดสามารถประกาศให้ทราบได้โดยทั่วไป แต่รายการวัตถุในถุงนั้นจะต้องเก็บรักษาเป็นความลับ เมื่อนำมารวมกับข้อจำกัดบางอย่าง ปัญหาการเดารายการวัตถุที่มีอยู่ในถุงนั้นพร้อมทั้งน้ำหนักที่ระบุเป็นเรื่องที่ไม่สามารถคำนวณหาคำตอบได้

ผู้พัฒนาวิธีการนี้มีความมั่นใจว่าอัลกอริทึมนี้จะไม่สามารถมีผู้ใดแก้ไขได้ จึงเสนอเงินรางวัล

จำนวนหนึ่งให้แก่ผู้ที่สามารถแก้ปัญหานี้ได้ ปรากฏว่า Adi Shamir (หนึ่งในผู้คิดค้นวิธี RSA, "A") สามารถแก้ปัญหานี้ได้ในทันทีและได้รับเงินรางวัลไป Merkle ผู้ไม่ยอมแพ้ ได้พัฒนาวิธีการนี้ให้ดียิ่งขึ้นไปอีกและเสนอเพิ่มเงินรางวัลให้มากกว่าเดิม ปรากฏว่า Ronald Rivest (หนึ่งในผู้คิดค้นวิธี RSA, "R") สามารถแก้ปัญหาใหม่นี้และได้รับเงินรางวัลไปอีก อย่างไรก็ตาม วิธีการนี้จัดว่าเป็นวิธีการที่ไม่ปลอดภัยและไม่ได้รับการนำไปใช้งาน

8.4 ลายเซ็นต์ดิจิตอล

การพิสูจน์สิทธิผู้ใช้ในหลายองค์กรกระทำได้โดยการปรากฏหรือไม่ปรากฏลายเซ็นต์ ซึ่งเป็นลายมือชื่อของผู้ที่มีสิทธิในเรื่องนั้น ๆ แต่ไม่ยอมรับเอกสารที่สร้างขึ้นมาจากการทำสำเนาเอกสาร สำหรับระบบข้อความทางด้านคอมพิวเตอร์ที่จะมาแทนที่เอกสารที่สร้างขึ้นจากกระดาษและหมึกพิมพ์ ก็จำเป็นที่จะต้องหาวิธีการเพื่อให้สามารถเซ็นต์ได้เช่นเดียวกับการเซ็นต์เอกสารปกติ

ปัญหาในการคิดค้นหาวิธีที่จะมาแทนลายเซ็นต์ที่ได้จากการเซ็นต์นั้นค่อนข้างยาก โดยพื้นฐานแล้วสิ่งที่ต้องการคือระบบที่ผู้ใช้งานสามารถเซ็นต์ข้อความออกไปยังผู้รับข่าวสารโดยมีเงื่อนไขดังนี้

1. ผู้รับสามารถตรวจสอบความเป็นตัวตน (identity) ของผู้ส่งได้
2. ผู้ส่งจะไม่สามารถปฏิเสธความรับผิดชอบในเอกสารที่ส่งไปแล้วได้ในภายหลัง
3. ผู้รับจะต้องไม่สามารถแก้ไขข้อความในเอกสารนั้นได้

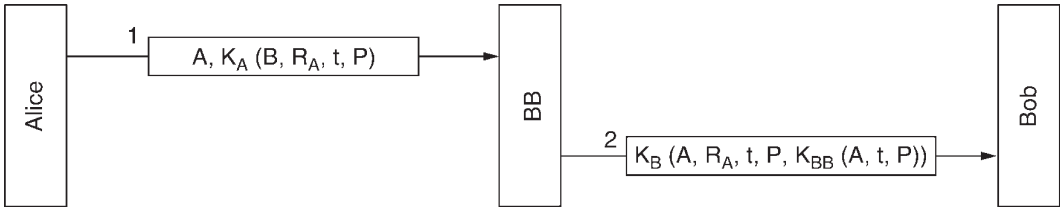
คุณสมบัติข้อแรกนั้นมีความต้องการเป็นอย่างยิ่ง เช่น ในระบบการเงินการธนาคาร เมื่อเครื่องคอมพิวเตอร์ของลูกค้าสั่งให้เครื่องคอมพิวเตอร์ของทางธนาคารชำระเงินในการซื้อทองคำสักหนึ่งพันกิโลกรัม เครื่องคอมพิวเตอร์ของธนาคารจะต้องสามารถทำให้แน่ใจได้ว่าเครื่องคอมพิวเตอร์ที่เป็นผู้ออกคำสั่งนั้นเป็นเครื่องของลูกค้าจริง หรืออีกนัยหนึ่งก็คือ ธนาคารจะต้องสามารถตรวจสอบผู้ใช้ (authenticate) (และผู้ใช้ก็จะต้องตรวจสอบธนาคารได้ด้วย)

คุณสมบัติประการที่สองนั้นมีไว้เพื่อป้องกันการโกง (fraud) สมมุติว่าทางธนาคารได้จัดการชำระค่าสินค้าให้ตามที่ได้รับการร้องขอมา และหลังจากนั้นราคาทองคำตกลงอย่างรวดเร็ว ลูกค้าที่ไม่ซื่ออาจจะทำการฟ้องร้องหาว่าทางธนาคารจัดการชำระเงินเองโดยที่เขาไม่ได้เป็นผู้ออกคำสั่งซื้อทองคำจำนวนนั้น เมื่อทางธนาคารนำหลักฐานการสั่งซื้อไปแสดงต่อศาล ลูกค้าผู้นั้นก็ปฏิเสธว่าไม่ได้เป็นผู้ส่งคำสั่งดังกล่าว ลายเซ็นต์อิเล็กทรอนิกส์ที่กำลังจะกล่าวถึงนี้จะสามารถนำมาใช้ในการแสดงหลักฐานมัดลูกค้าว่าเขาเองเป็นผู้ส่งคำสั่งซื้อนั้นมายังธนาคาร

คุณสมบัติข้อที่สาม มีความจำเป็นในการปกป้องลูกค้า เช่น ในกรณีที่ราคาทองคำได้เพิ่มขึ้นอย่างมาก ทางธนาคารจึงพยายามหากำไรเข้าสู่ตัวเองด้วยแก้ไขคำสั่งซื้อของลูกค้าให้กลายเป็นเพียงการซื้อทองคำเพียงบาทเดียวแทนที่จะเป็นหนึ่งตัน (ธนาคารเก็บกำไรไว้กับตัว)

8.4.1 ลายเซ็นต์แบบใช้คีย์สมมาตร

แนวทางหนึ่งของการใช้ลายเซ็นต์อิเล็กทรอนิกส์คือการจัดให้มีผู้มีอำนาจส่วนกลางที่รู้ทุกอย่างว่าใครสามารถเชื่อถือได้ สมมุติให้ชื่อว่า BB (Big Brother) ผู้ใช้แต่ละคนจะเลือกคีย์ลับเฉพาะของตนเองขึ้นมาแล้วนำไปฝากไว้ที่สำนักงานของ BB เช่น จะมีแต่อลิสและ BB เท่านั้นที่ทราบคีย์ลับของอลิส (KA)



รูปที่ 8-18 ลายเซ็นอิเล็กทรอนิกส์ที่มี Big Brother เป็นคนกลาง

เป็นต้น

เมื่ออลิสต้องการส่งข้อความ plaintext (P) และลงชื่ออิเล็กทรอนิกส์กับข้อความนั้น ไปยังผู้แทนธนาคารของเธอคือ บ็อบ เธอก็จะสร้างข้อมูล KA(B, RA, t, P) ขึ้นมา โดยที่ B คือชื่อทางอิเล็กทรอนิกส์ของบ็อบ RA คือ ตัวเลขที่สุ่มขึ้นมาโดยอลิส และ t คือการลงเวลาที่ทำการสร้างข้อความนี้ขึ้นมา อลิสจะส่งข้อความดังกล่าวไปยังบ็อบดังแสดงในรูป 8-18 เมื่อ BB ได้รับข้อความที่ส่งมาโดยอลิสก็จะทำการถอดรหัสและส่งข้อความนั้นไปยังบ็อบดังแสดงในรูป ข้อความที่ถูกส่งไปยังบ็อบประกอบด้วยข้อความ plaintext ของอลิสและการลงชื่อรับรอง KBB(A, t, P) ณ ตอนนี้บ็อบก็สามารถดำเนินการตามที่อลิสร้องขอมาในข้อความนั้น

อะไรจะเกิดขึ้นถ้าอลิสปฏิเสธว่าไม่ได้ส่งข้อความดังกล่าวไป เมื่อเรื่องดำเนินการไปถึงการสอบสวนในชั้นศาล ผู้พิพากษาจะถามบ็อบว่าเขาแน่ใจได้อย่างไรว่าข้อความที่ได้รับนั้นมาจากอลิสไม่ใช่คนอื่น บ็อบตอบว่า ประการแรก BB จะไม่รับข้อความจากอลิสนอกเสียจากว่าข้อความนั้นจะถูกเข้ารหัสมาด้วย KA จึงเป็นไปไม่ได้ว่าคนผู้อื่นจะส่งข่าวสารมายัง BB ในชื่อของอลิสโดยที่ BB ไม่รู้

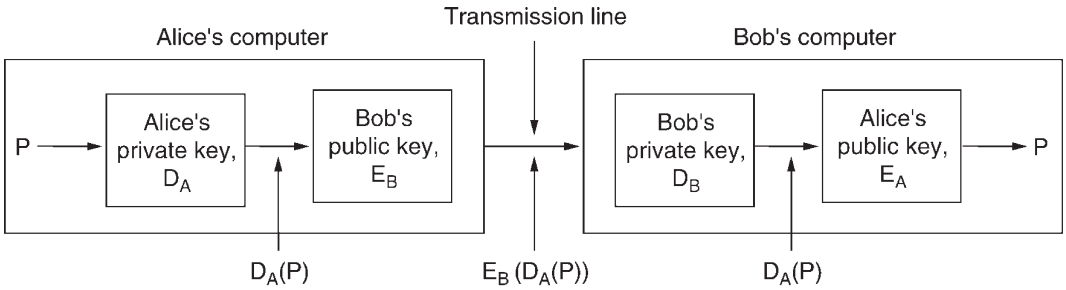
จากนั้นบ็อบจะสร้าง Exhibit A ขึ้นมาจาก KBB(A, t, P) บ็อบกล่าวว่านี่คือข้อความที่ลงชื่ออิเล็กทรอนิกส์กำกับไว้โดย BB ซึ่งเป็นการพิสูจน์ว่าอลิสส่งข้อความ P มายังบ็อบจริง จากนั้นผู้พิพากษาอาจจะให้ BB (คนที่ทุกคนไว้ใจ) ทำการถอดรหัสข้อความ Exhibit A ซึ่งก็จะพบว่าเป็นข้อความตรงกับที่บ็อบมีอยู่จริง

ปัญหาประการหนึ่งเกี่ยวกับวิธีการที่ใช้ในรูป 8-18 คือ อาจมีบุคคลอื่นทำการส่งข้อความ (ที่ถูกต้อง) ซ้ำ เพื่อเป็นการขจัดปัญหานี้จึงได้มีการบันทึกเวลาที่สร้างข้อความนั้นขึ้นมา ยิ่งกว่านี้ บ็อบสามารถตรวจสอบข้อความล่าสุดทั้งหมดเพื่อดูว่า RA ถูกนำไปใช้อีกหรือไม่ สำหรับข่าวสารที่มีค่า RA ซ้ำเดิมก็จะถูกลบทิ้งทั้งหมด สังเกตว่าการลงเวลาที่สร้างข่าวสารขึ้นมานั้น บ็อบจะลบข่าวสารเก่าทิ้งเพื่อป้องกันปัญหาการบุกรุกด้วยข้อความซ้ำซ้อนในทันทีทันใด (instant replay attack) บ็อบจะตรวจสอบค่า RA สำหรับทุกข่าวสารที่รับเข้ามาว่าเป็นข่าวสารที่ได้รับเข้ามาแล้วหรือไม่ ถ้าไม่ใช่ก็สามารถรับข่าวสารนั้นได้

8.4.2 ลายเซ็นต์แบบใช้คีย์สาธารณะ

ปัญหาทางโครงสร้างเกี่ยวกับการเข้ารหัสแบบใช้คีย์สมมาตรกับลายเซ็นต์อิเล็กทรอนิกส์ก็คือ ทุกคนจะต้องตกลงใจที่จะให้ความเชื่อถือ Big Brother ยิ่งกว่านี้ Big Brother ยังสามารถอ่านข้อความได้ทุกข้อความ กลุ่มคนที่สามารถทำตัวเป็น Big Brother ได้ได้แก่ รัฐบาล ธนาคาร นักบัญชี และนักกฎหมาย อย่างไรก็ตาม ไม่มีผู้ใดในองค์กรเหล่านี้ที่เป็นผู้ที่ไว้วางใจได้สำหรับประชาชนทุกคน ดังนั้นจะเป็นการดีกว่าถ้าเอกสารที่ลงชื่ออิเล็อกทรอนิกส์จะไม่ต้องอาศัยผู้แทนที่ต้องเชื่อถือได้

รูปที่ 8-19
ลายเซ็นอิเล็กทรอนิกส์โดยใช้การเข้ารหัสด้วยคีย์สาธารณะ



การเข้ารหัสโดยใช้กุญแจสาธารณะ (public-key cryptography) สามารถนำมาใช้แก้ปัญหานี้ได้ สมมุติว่าอัลกอริทึมที่นำมาใช้ในการเข้ารหัส (E) และถอดรหัส (D) มีคุณสมบัติ E(D(P)) = P และ D(E(P)) = P อลิสต้องการส่งข้อความที่ลงชื่อกำกับ (P) ไปยังบ็อบจึงส่งข้อความ EB(DA(P)) ซึ่งสามารถทำได้เนื่องจากเธอทราบคีย์ส่วนตัวของเธอเอง (DA) และทราบคีย์สาธารณะของบ็อบ (EB) อลิสจึงสามารถสร้างข่าวสารชิ้นนี้ขึ้นมาได้

เมื่อบ็อบได้รับข่าวสารชิ้นนี้ เขาก็สามารถเปลี่ยนแปลงให้กลับมาอยู่ในรูปแบบปกติได้โดยใช้คีย์ส่วนตัวของเขาและคีย์สาธารณะของอลิสดังแสดงในรูป 8-19

เพื่อเป็นการพิสูจน์ว่าวิธีการนี้ใช้ได้ผล สมมุติว่าอลิสปฏิเสธว่าไม่ได้ส่งข่าวสาร P มายังบ็อบ เมื่อเริ่มขึ้นสู่ศาล บ็อบจะสามารถสร้างข้อความได้ทั้ง P และ DA(P) ผู้พิพากษาก็จะสามารถพิสูจน์ความจริง ได้โดยทำการถอดรหัส DA(P) โดยใช้อัลกอริทึมสำหรับการเข้ารหัส EA ของอลิสซึ่งจะได้ข้อความ P ที่เป็นข้อความเดียวกันกับที่บ็อบมีอยู่ เนื่องจากบ็อบไม่ทราบคีย์ส่วนตัวของอลิสจึงเป็นการพิสูจน์ว่าอลิสต้องเป็นผู้ส่งข่าวสารชิ้นนี้มายังบ็อบ

แม้ว่าการนำวิธีการเข้ารหัสโดยใช้กุญแจสาธารณะมาใช้กับลายเซ็นต์อิเล็กทรอนิกส์เป็นวิธีการที่ดีอย่างหนึ่ง แต่ก็ยังอาจเกิดปัญหาขึ้นได้เนื่องจากสิ่งแวดล้อมที่นำวิธีการนี้ไปใช้ไม่ใช่ปัญหาที่ตัววิธีการนี้ กล่าวคือ บ็อบจะสามารถพิสูจน์ได้ว่าข้อความได้ถูกส่งมาจากอลิสก็ต่อเมื่อข้อความ DA ยังคงถูกเก็บเป็นความลับ ถ้าอลิสเปิดเผยคีย์ส่วนตัวของเธอออกมาเมื่อใดการพิสูจน์นี้ก็ใช้ไม่ได้เนื่องจากผู้ใดก็สามารถส่งข้อความรหัส DA ได้ รวมทั้งตัวบ็อบเอง

อีกปัญหาหนึ่งที่อาจเกิดขึ้นคือ อะไรจะเกิดขึ้นถ้าอลิสตัดสินใจที่จะเปลี่ยนรหัสส่วนตัวของเธอเอง การเปลี่ยนรหัสส่วนตัวเป็นเรื่องที่ถูกกฎหมายแต่จะเป็นการดีถ้าทำการเปลี่ยนรหัสส่วนตัวเป็นระยะ ๆ ถ้าการพิสูจน์ในชั้นศาลเกิดขึ้นภายหลังจากที่อลิสได้เปลี่ยนรหัสส่วนตัวของเธอไปแล้ว ผู้พิพากษาก็จะไม่สามารถพิสูจน์ได้ว่าอลิส (พร้อมคีย์ใหม่) เป็นผู้สร้างข้อความนั้นขึ้นมา

ในทางปฏิบัติ การเข้ารหัสโดยใช้กุญแจสาธารณะสามารถนำมาใช้กับลายเซ็นต์อิเล็กทรอนิกส์ได้ มาตรฐานที่เป็นที่ยอมรับกันโดยทั่วไปคือการนำวิธี RSA มาใช้ สินค้าเกี่ยวกับความปลอดภัยหลายชนิดต่างก็ใช้วิธีการนี้ อย่างไรก็ตาม ในปี ค.ศ. 1991 องค์กร NIST ได้นำเสนอให้ใช้วิธี El Gamal public-key algorithm สำหรับเป็นมาตรฐานใหม่ของลายเซ็นต์อิเล็กทรอนิกส์ (Digital Signature Standard; DSS)

อย่างไรก็ตาม เมื่อองค์กรของรัฐพยายามที่จะเข้ามากำหนดบทบาทมาตรฐานการเข้ารหัสก็มักจะได้รับการต่อต้าน DSS ได้รับการต่อต้านด้วยเหตุผลหลัก 4 ประการคือ

1. เป็นความลับมากเกินไป (องค์กร NSA เป็นผู้พัฒนาวิธีการนี้)

2. ทำงานช้าเกินไป (ทำงานช้ากว่าวิธี RSA 10 ถึง 40 เท่า)

3. ใหม่เกินไป (El Gamal ยังไม่ได้รับการศึกษาวิเคราะห์อย่างถ่องแท้)

4. ไม่ปลอดภัย (ใช้คีย์คงที่ขนาด 512 บิต)

8.4.3 เทคนิคการย่อยข่าวสาร

การติเตียนวิธีการลายเซ็นต์อิเล็กทรอนิกส์อย่างหนึ่งคือเป็นวิธีการที่รวมเอางานสองอย่างเข้าไว้ด้วยกันคือ การตรวจสอบผู้ใช้ (authentication) และการรักษาความลับ (secrecy) โดยทั่วไปจะพบว่าการตรวจสอบผู้ใช้นั้นเป็นสิ่งที่ต้องการแต่การรักษาความลับนั้นไม่จำเป็นต้องทำ เช่น การขออนุญาตส่งผลิตภัณฑ์เป็นสินค้าส่งออก (ในประเทศสหรัฐอเมริกา) นั้นจะทำได้ง่ายขึ้นถ้าสินค้า (ซอฟต์แวร์) นั้นทำหน้าที่เพียงแค่ตรวจสอบผู้ใช้แต่ไม่มีความลับ (ทางการค้า-คือความลับที่ใช้ในการผลิตซอฟต์แวร์นั้น) เข้ามาเกี่ยวข้องในหัวข้อนี้จะได้กล่าวถึงวิธีการตรวจสอบผู้ใช้โดยไม่จำเป็นต้องเข้ารหัสข้อความทั้งหมด

วิธีการนี้มีพื้นฐานมาจากการใช้วิธี one-way hash function ที่นำข้อความที่ยาวมากของ plaintext มาทำการคำนวณหากระแสบิตที่มีความยาวคงที่ Hash function นี้เรียกว่า MD หรือ Message Digest ซึ่งมีคุณสมบัติที่สำคัญ 4 ประการคือ

1. กำหนดให้ P, จะสามารถคำนวณ MD(P) ได้อย่างง่ายดาย

2. กำหนดให้ MD(P), จะไม่สามารถนำไปสร้างเป็น P ได้

3. กำหนดให้ P, จะไม่มีผู้ใดสามารถค้นหา P ที่มีคุณสมบัติ MD(P) = MD(P)

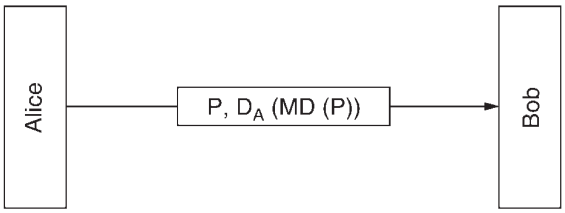
4. การเปลี่ยนแปลงแม้เพียง 1 บิตที่เกิดขึ้นใน P จะสร้างผลลัพธ์ที่แตกต่างไปจากเดิมมาก

เพื่อให้เป็นไปตามคุณสมบัติข้อ3 hash function จะต้องมีความยาวไม่น้อยกว่า 128 บิตหรือมากกว่านี้ เพื่อให้เป็นไปตามคุณสมบัติข้อ 4 hash function จะต้องตรวจสอบบิตข้อมูลทุกบิตโดยละเอียดซึ่งไม่เหมือนกับวิธีการเข้ารหัสแบบใช้คีย์สมมาตรที่ได้กล่าวมาแล้ว

การคำนวณ message digest จากข้อความชิ้นหนึ่งนั้นทำได้เร็วกว่าการเข้ารหัส plaintext ที่ใช้คีย์สาธารณะมาก ดังนั้น message digest จึงสามารถนำมาใช้เพิ่มความเร็วในการลงชื่ออิเล็กทรอนิกส์ให้เร็วขึ้นได้ ลองพิจารณารูป 8-18 อีกครั้งหนึ่ง แทนที่จะลงชื่อกำกับข้อความ P ด้วย KBB(A, t, P) BB จะหันมาคำนวณ message digest ซึ่งจะได้ MD(P) จากนั้น BB ก็จะใส่ KBB(A, t, MD(P)) เป็นข้อมูลตัวที่ 5 ในรายการข้อมูลที่เข้ารหัสด้วย KB ซึ่งจะถูกส่งไปให้บ๊อบแทนที่จะเป็น KBB(A, t, P)

ถ้ามีการโต้เถียงเกิดขึ้น บ๊อบจะสามารถสร้างข้อความทั้ง P และ KBB(A, t, MD(P)) ขึ้นมาได้ หลังจากนั้น Big Brother ก็จะสามารถถอดรหัสข้อความที่เข้ารหัสไว้ให้แก่ผู้พิพากษาได้ ส่วนบ๊อบก็ยังมี MD(P) ซึ่งได้มาจากข้อความ P ของอลิสไว้เป็นการพิสูจน์อีกทางหนึ่งด้วย อย่างไรก็ตาม เนื่องจากเป็นไปไม่ได้ที่บ๊อบจะสามารถหาข้อความอื่นที่จะให้ผลเช่นเดียวกันกับข้อความเดิม ก็จะเป็นการพิสูจน์ว่า บ๊อบนั้นพูดความจริง การใช้ message digest ในที่นี้ช่วยทั้งการประหยัดเวลาในการเข้ารหัสและ

รูปที่ 8-20
ลายเซ็นต์อิเล็กทรอนิกส์
โดยใช้ message
digest



ค่าสื่อสารในการส่งรหัสข้อความนั้นด้วย

Message digest ทำงานร่วมกับการเข้ารหัสแบบใช้คีย์สาธารณะดังแสดงในรูป 8-20 ในที่นี้อลิสเริ่มต้นด้วยการคำนวณ message digest สำหรับข้อความ palintext ของเธอ จากนั้นจัดการลงชื่ออิเล็กทรอนิกส์แล้วจัดการส่งข้อความดังกล่าวมายังบ็อบ ถ้ามีผู้ใดแก้ไขเปลี่ยนแปลงข้อความในระหว่างการนำส่ง บ็อบจะสามารถทราบได้ทันทีด้วยการคำนวณหา MD(P) ด้วยตนเอง

MD5

วิธีการ message digest ได้รับการพัฒนาขึ้นมาหลากหลายวิธี ซึ่งวิธีที่ได้รับการนำไปใช้งานมากที่สุดคือ MD5 และ SHA-1 วิธี MD5 เป็นวิธีการที่ 5 ที่ได้รับการพัฒนาขึ้นมาโดย Ronald Rivest ในปีค.ศ. 1992 มีหลักการทำงานคือจะทำการตรวจสอบข้อมูลโดยละเอียดด้วยวิธีการที่ซับซ้อนมากจนสามารถรับประกันได้ว่าผลลัพธ์ที่ได้รับนั้นจะมีค่าเปลี่ยนไปจากเดิมเสมอถ้าข้อมูลที่นำเข้ามานั้นเปลี่ยนแปลงไปเพียงบิตเดียวก็ตาม วิธีการนี้เริ่มต้นโดยการเพิ่มเติมข้อมูลให้มีขนาดเป็น 448 บิต จากนั้นต่อข้อมูลเดิมเข้าไปอีก 64 บิตเพื่อให้ข้อมูลทุกบล็อคมีขนาดเป็นจำนวนเท่าของ 512 บิต ขั้นตอนสุดท้ายในการเตรียมการจะกำหนดค่าเริ่มต้นให้กับบัฟเฟอร์ขนาด 128 บิตให้เป็นค่าคงที่ค่าหนึ่ง

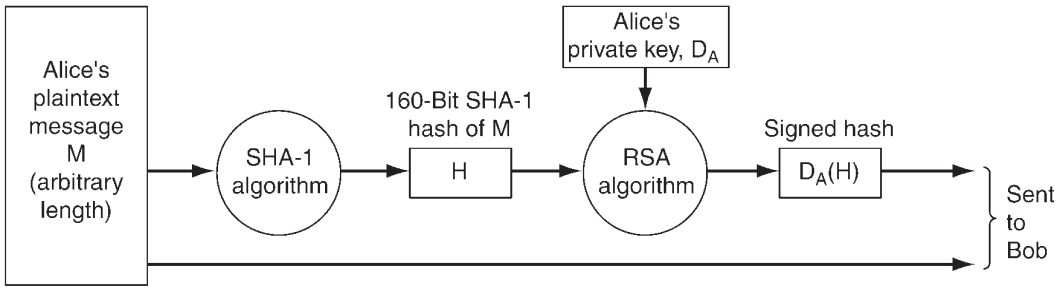
ต่อไปเริ่มทำการคำนวณในแต่ละรอบ ให้นำข้อมูลจำนวน 512 บิตมาทำการผสมกันอย่างทั่วถึงในบัฟเฟอร์ขนาด 128 บิตที่เตรียมไว้ ข้อมูลจะถูกนำมาผสมกัน 4 รอบ กระบวนการนี้จะเกิดขึ้นสำหรับข้อมูลทุกบล็อคจนหมด ค่าที่คงค้างอยู่ในบัฟเฟอร์ขนาด 128 บิตนี้คือ message digest

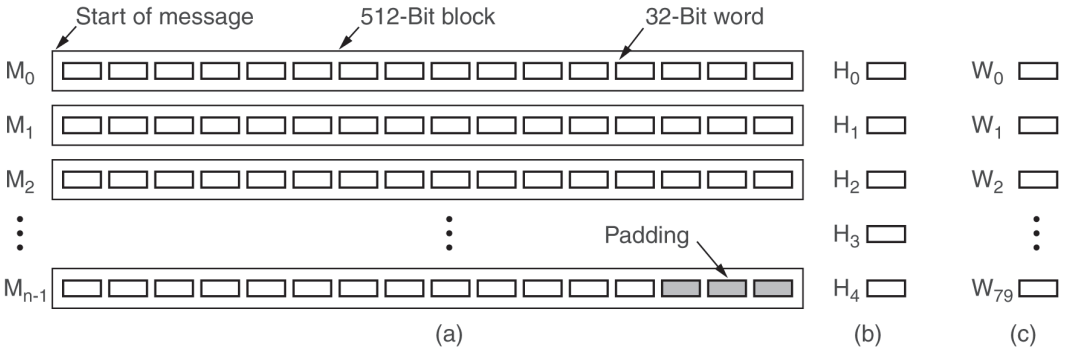
วิธี MD5 ได้รับการคิดค้นและนำมาใช้งานมากกว่า 5 ทศวรรษมาแล้วและได้เคยถูกลองพยายามถอดรหัสมาแล้วหลายครั้ง พบว่ามีจุดที่เป็นจุดที่เปราะบางแต่ขั้นตอนการทำงานวนซ้ำภายในช่วยไม่ให้สามารถถอดรหัสได้

SHA-1

วิธีการ message digest อีกวิธีหนึ่งเรียกว่า SHA-1 (secure Hash Algorithm 1) ได้รับการพัฒนาขึ้นมาจากองค์กร NSA (National Security Agency) วิธีการนี้ทำการประมวลผลข้อมูลบล็อค

รูปที่ 8-21
การใช้ SHA-1
ร่วมกับ RSA ในการ
ลงชื่อรับรองข่าวสาร
ที่ไม่เป็นความลับ





รูปที่ 8-22
(a) การเพิ่มเติมข้อมูลเข้าไปเพื่อให้ครบ 512 บิต
(b) พลลัพธ์ variable ที่ได้
(c) the word array

ละ 512 บิต ซึ่งจะสร้าง message digest ขึ้นมาขนาด 160 บิต รูป 8-21 แสดงวิธีการที่อลิสใช้ในการส่งข้อความที่ไม่เป็นความลับแต่ต้องการลงชื่อรับรองมายังบ็อบ ข้อความของเธอได้ถูกส่งผ่านอัลกอริทึม SHA-1 ซึ่งได้สร้าง message digest ขนาด 160 บิตออกมา จากนั้นจะทำการลงชื่อรับรองด้วยคีย์ส่วนตัวแบบ RSA ของอลิส แล้วส่งข้อความพร้อมกับ message digest ที่ลงชื่อรับรองแล้วไปยังบ็อบ

หลังจากที่ได้รับข้อความแล้ว บ็อบจะทำการคำนวณ SHA-1 hash ด้วยตัวเขาเองและใช้คีย์สาธารณะของอลิสในการลงชื่อรับรองเพื่อให้ได้เป็น hash H ถ้าหากว่าทั้ง hash H และ hash ที่อลิสส่งมานั้นเหมือนกันก็แสดงว่าข้อความที่ส่งมานั้นถูกต้อง เนื่องจากเป็นไปไม่ได้ที่จะมีบุคคลอื่นมาทำการแก้ไข plaintext ที่อลิสส่งมาแล้วทำให้ค่า hash H ที่บ็อบคำนวณได้จะมีค่าเหมือนกับค่า hash ที่อลิสส่งมาด้วย สำหรับข้อความที่เน้นความสำคัญของความถูกต้องของข้อมูลแต่ตัวข้อความนั้นไม่เป็นความลับมักจะนำวิธีการที่แสดงในรูป 8-21 ไปใช้เนื่องจากใช้เวลาในการคำนวณน้อยและสามารถรับประกันได้ว่าการเปลี่ยนแปลงที่เกิดขึ้นกับ plaintext แม้เพียงเล็กน้อยก็จะสามารถถูกตรวจจับได้ด้วยความเป็นไปได้สูงมาก

ต่อไปพิจารณาว่ากระบวนการ SHA-1 นั้นทำงานอย่างไร เริ่มต้นด้วยการเติมข้อความที่จะส่งด้วยบิต "1" ที่จุดสิ้นสุดของข้อความเพียงบิตเดียว แล้วตามด้วยบิต "0" มากเท่าที่จะทำให้ข้อความนั้นมีความยาว 512 บิต (หรือเป็นจำนวนเท่าของ 512) จากนั้นนำตัวเลขขนาด 64 บิตมาตัวหนึ่งที่บอกความยาวของข้อความที่จะส่ง (ก่อนเติมบิตพ่วงท้ายเข้าไป) มาทำการ OR กับ 64 บิตสุดท้ายของข้อความนั้น รูป 8-22 แสดงข้อความที่เติมบิตพ่วงท้ายไว้ทางด้านขวามือเพราะข้อความภาษาอังกฤษนั้นอ่านจากซ้ายไปขวาและจากบนลงล่าง สำหรับคอมพิวเตอร์แล้วการวางตัวอักษรในลักษณะเช่นนี้เรียกว่า big-endian

ในระหว่างการคำนวณ SHA-1 จะรักษาตัวแปรขนาด 32 บิตจำนวน 5 ตัวคือ H0, H1, H2, H3, และ H4 เอาไว้เพื่อเป็นตัวสะสมค่าของ hash ที่คำนวณได้ ดังแสดงในรูป 8-22(b) ค่าของตัวแปรทั้ง 5 ตัวนี้จะถูกกำหนดค่าเริ่มต้นไว้ตามที่มาตรฐานกำหนด

ข้อความแต่ละบล็อค คือ M0 ถึง Mn-1 จะถูกนำมาคำนวณหมุนเวียนกัน สำหรับแต่ละบล็อคข้อความ 16 คำแรกจะถูกสร้างสำเนาเข้าไปเก็บไว้ในอาเรย์ขนาด 80 คำ (W) ดังแสดงในรูป 8-22(c)

จากนั้นอีก 64 คำที่เหลือในอาเรย์จะถูกเติมเข้าไปโดยใช้สูตรการคำนวณคือ

$$W_i = S^1(W_{i-3} \text{ XOR } W_{i-8} \text{ XOR } W_{i-14} \text{ XOR } W_{i-16} \quad (16 \le i \le 79)$$

โดยที่ Sb(W) คือการหมุนคำ W ขนาด 32 บิตนั้นไปทางด้านซ้าย (left circular rotation) เป็นจำนวน b บิต จากนั้นตัวแปรอิสระอีก 5 ตัวคือ A, B, C, D, และ E จะถูกกำหนดค่าเริ่มต้นให้เป็น H0 ถึง H4 ตามลำดับ

การคำนวณดังกล่าวสามารถเขียนแทนด้วยโค้ดจำลองได้ดังนี้

```
for (i = 0; i < 80; i++) {
    temp = S^5(A) + f_i (B, C, D) + E + W_i + K_i;
    E = D;  D = C;  C = S^30(B);  B = A;  A = temp;
}
```

โดยที่ Ki คือค่าคงที่ที่กำหนดไว้ในมาตรฐานของ SHA-1 ค่าของ fi กำหนดไว้ดังนี้

$$f_i(B,C,D) = (B \text{ AND } C) \text{ OR } (\text{NOT } B \text{ AND } D) \qquad (0 \le i \le 19)$$
$$f_i(B,C,D) = B \text{ XOR } C \text{ XOR } D \qquad (20 \le i \le 39)$$
$$f_i(B,C,D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \qquad (40 \le i \le 59)$$
$$f_i(B,C,D) = B \text{ XOR } C \text{ XOR } D \qquad (60 \le i \le 79)$$

เมื่อทำการคำนวณวนซ้ำครบ 80 รอบแล้ว ให้นำค่า A ถึง E ไปบวกเข้ากับ H0 ถึง H4 ตามลำดับ

การคำนวณ 512 บิตแรกเสร็จเรียบร้อย ก็ให้เริ่มคำนวณ 512 บิตถัดมา อาเรย์ W จะต้องถูกกำหนดค่าเริ่มต้นใหม่เมื่อเริ่มทำงานกับบล็อคใหม่แต่ค่าของ H นั้นให้คงไว้อย่างที่เป็นอยู่ เมื่อคำนวณบล็อคนี้เสร็จก็เริ่มบล็อคใหม่ต่อไปเรื่อยๆ จนครบทุกบล็อค เมื่อคำนวณเสร็จแล้ว ค่าที่ปรากฏอยู่ในอาเรย์ H ทั้ง 5 ค่านั้น (H0 ถึง H4) จะถูกนำมาเรียงต่อกันกลายเป็นข้อมูลขนาด 160 บิต hash ที่ต้องการ

8.4.4 Birthday Attack

ในโลกของการเข้ารหัสข้อมูลไม่มีอะไรที่เป็นไปตามความคาดหวัง เช่น อาจคิดว่าน่าจะมีการทำงานเกิดขึ้น 2m ครั้งในการคำนวณหาค่า message digest ขนาด m บิต อันที่จริงแล้วการคำนวณเพียง 2m/2 ครั้งก็น่าจะเพียงพอแล้วโดยใช้วิธีการเรียกว่า birthday attack ซึ่งเป็นวิธีการที่พัฒนาขึ้นโดย Yuval ในปี ค.ศ. 1979

แนวความคิดในการโจมตีแบบนี้นำมาจากเทคนิคทางด้านวิชาความน่าจะเป็น คำถามก็คือ จะต้องมีจำนวนนักศึกษาในชั้นเรียนเป็นจำนวนเท่าใดจึงจะทำให้ความน่าจะเป็นที่นักศึกษาอย่างน้อยสองคนมีวันคล้ายวันเกิดเป็นวันเดียวกันนั้นมีค่ามากกว่า 1/2 นักศึกษาส่วนใหญ่มักจะคาดหวังว่าคำตอบคงเป็นจำนวนกว่า 100 คนขึ้นไป ซึ่งอันที่จริงตามทฤษฎีความน่าจะเป็นแล้วคำตอบที่ได้คือมีจำนวนนักศึกษาตั้งแต่ 23 คนขึ้นไปเท่านั้น การพิสูจน์แบบง่ายๆ คือ จากจำนวนนักศึกษา 23 คน สามารถจับคู่ได้เป็น (23x22)/2 = 253 คู่ แต่ละคู่มีความน่าจะเป็นที่จะมีวันคล้ายวันเกิดกันเป็น 1/356 ดังนั้นความน่าจะเป็นที่นักศึกษาจะมีวันคล้ายวันเกิดเป็นวันเดียวกันคือ 253/356 ซึ่งมีค่ามากกว่า 1/2

ในรูปแบบทั่วไป ถ้ามีการจับคู่ระหว่าง input กับ output โดยมีจำนวน input เป็น n จำนวน (ผู้คน ข่าวสาร ฯลฯ) และความเป็นไปได้ของ output เป็น k (วันเกิด, message digest, ฯลฯ) แล้ว จะสามารถจับคู่ได้เป็น n(n-1)/2 คู่ ถ้า n(n-1)/2 > k จะทำให้โอกาสที่จะมีอย่างน้อย 1 คู่ที่จะได้ตรงตามคุณสมบัติที่ต้องการนั้นเป็นไปได้สูงมาก ดังนั้น ค่าของการได้การจับคู่ที่จะประสบความสำเร็จจึงได้ประมาณ n>k หมายความว่าข้อความ message digest ขนาด 64 บิตมีความน่าจะเป็นที่จะสามารถถูกถอดรหัสออกได้โดยการสร้างข้อความขึ้นมา 232 ข้อความและมองหาข้อความที่มีค่าเหมือนกัน

ลองพิจารณาตัวอย่างที่สามารถเกิดขึ้นได้จริงดังนี้ ภาควิชาคอมพิวเตอร์ที่มหาวิทยาลัย State University มีตำแหน่งว่างหนึ่งตำแหน่งสำหรับสมาชิกของคณะอาจารย์และมีผู้เข้าแข่งขันสองท่านคือ Tom และ Dick Tom นั้นได้เข้ามาทำงานก่อน Dick 2 ปี จึงได้รับการพิจารณาเป็นลำดับแรก ถ้าผ่านการพิจารณา Dick ก็จะหมดโอกาสในทันที Tom รู้จักเป็นการส่วนตัวกับหัวหน้าคณะวิชาคอมพิวเตอร์ชื่อ Marilyn ซึ่งมีทัศนคติที่ดีต่อเขา เขาจึงขอร้องให้เธอเขียนจดหมายรับรองไปยัง Dean ซึ่งเป็นผู้ที่จะ

Dear Dean Smith,

This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a candidate* | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Prof. Wilson for [*about* | *almost*] six years. He is an [*outstanding* | *excellent*] researcher of great [*talent* | *ability*] known [*worldwide* | *internationally*] for his [*brilliant* | *creative*] insights into [*many* | *a wide variety of*] [*difficult* | *challenging*] problems.

He is also a [*highly* | *greatly*] [*respected* | *admired*] [*teacher* | *educator*]. His students give his [*classes* | *courses*] [*rave* | *spectacular*] reviews. He is [*our* | *the Department's*] [*most popular* | *best-loved*] [*teacher* | *instructor*].

[*In addition* | *Additionally*] Prof. Wilson is a [*gifted* | *effective*] fund raiser. His [*grants* | *contracts*] have brought a [*large* | *substantial*] amount of money into [*the* | *our*] Department. [*This money has* | *These funds have*] [*enabled* | *permitted*] us to [*pursue* | *carry out*] many [*special* | *important*] programs, [*such as* | *for example*] your State 2000 program. Without these funds we would [*be unable* | *not be able*] to continue this program, which is so [*important* | *essential*] to both of us. I strongly urge you to grant him tenure.

ทำการตัดสิน ในกรณีของ Tom เมื่อส่งจดหมายไปแล้ว จดหมายจะถูกเก็บเป็นความลับ

Dear Dean Smith,

This [*letter* | *message*] is to give my [*honest* | *frank*] opinion of Prof. Tom Wilson, who is [*a candidate* | *up*] for tenure [*now* | *this year*]. I have [*known* | *worked with*] Tom for [*about* | *almost*] six years. He is a [*poor* | *weak*] researcher not well known in his [*field* | *area*]. His research [*hardly ever* | *rarely*] shows [*insight in* | *understanding of*] the [*key* | *major*] problems of [*the* | *our*] day.

Furthermore, he is not a [*respected* | *admired*] [*teacher* | *educator*]. His students give his [*classes* | *courses*] [*poor* | *bad*] reviews. He is [*our* | *the Department's*] least popular [*teacher* | *instructor*], known [*mostly* | *primarily*] within [*the* | *our*] Department for his [*tendency* | *propensity*] to [*ridicule* | *embarrass*] students [*foolish* | *imprudent*] enough to ask questions in his classes.

[*In addition* | *Additionally*] Tom is a [*poor* | *marginal*] fund raiser. His [*grants* | *contracts*] have brought only a [*meager* | *insignificant*] amount of money into [*the* | *our*] Department. Unless new [*money is* | *funds are*] quickly located, we may have to cancel some essential programs, such as your State 2000 program. Unfortunately, under these [*conditions* | *circumstances*] I cannot in good [*conscience* | *faith*] recommend him to you for [*tenure* | *a permanent position*].

Marilyn บอกให้ Ellen ซึ่งเป็นเลขาของเธอเขียนจดหมายไปยัง Dean โดยที่เธอได้เขียนหัวข้อสั้นๆ ที่เธอต้องการเขียนในจดหมายนั้นให้ เมื่อจดหมายเสร็จเรียบร้อยแล้ว Marilyn จะดูจดหมายนั้นอีกครั้งหนึ่ง ทำการคำนวณ และลงชื่อกำกับด้วยวิธี 64 บิต digest ก่อนที่จะส่งไปยัง Dean ส่วน Ellen สามารถส่งจดหมายนั้นได้ในภายหลังโดยการใช้ e-mail

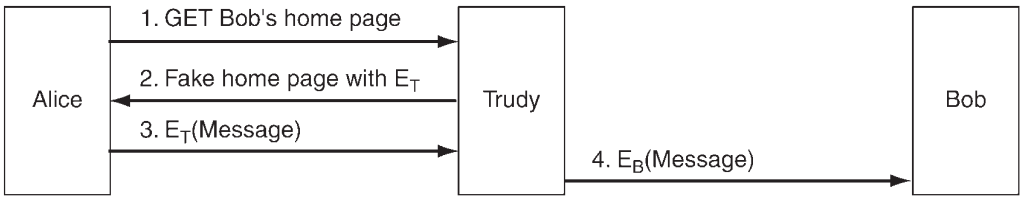
โชคไม่ดีสำหรับ Tom ที่ Ellen นั้นมีความสัมพันธ์ที่ลึกซึ้งกับ Dick และต้องการช่วยให้เขาได้รับตำแหน่งแทน Tom เธอจึงได้เขียนจดหมายที่มีตัวเลือกจำนวน 32 แห่งดังแสดงในข้อความข้างล่างนี้

โชคไม่ดีสำหรับ Tom ในทันทีที่ Ellen เขียนร่างจดหมายฉบับนี้จบ เธอก็เขียนร่างจดหมายขึ้นมาอีกฉบับหนึ่ง มีข้อความดังนี้

ขั้นต่อไป Ellen ได้โปรแกรมให้เครื่องคอมพิวเตอร์ของเธอทำการคำนวณ message digest ขึ้นมาเป็นจำนวน 232 ข้อความสำหรับจดหมายแต่ละฉบับ โอกาสที่เป็นไปได้ก็คือ หนึ่งใน message digest ของจดหมายฉบับแรกจะเป็นข้อความที่เหมือนกับ message digest ของจดหมายฉบับที่สอง ถ้าไม่พบข้อความที่เหมือนกัน เธอก็สามารถปรับค่าตัวแปรเพียงเล็กน้อยแล้วทำการสร้าง message digest ขึ้นมาใหม่ เมื่อพบข้อความที่เหมือนกัน ให้เรียกข้อความสำหรับจดหมายฉบับแรกว่าเป็น Good letter A และเรียกข้อความสำหรับจดหมายฉบับที่สองว่า Bad letter B

ขั้นต่อไป Ellen จะส่ง Good letter A ไปยัง Marilyn เพื่อให้ลงชื่อรับรอง ในขณะที่เก็บจดหมายฉบับที่สองไว้เป็นความลับ Marilyn ก็จะรับรองจดหมายนั้นแล้วสร้าง message digest ขนาด 64 บิตขึ้นมา ทำการลงชื่อรับรอง message digest นั้นแล้วส่งไปให้ Dean Smith ในเวลาเดียวกัน Ellen ก็จะ e-mail จอหมายลับของเธอไปให้ Dean Smith

หลังจากที่ Dean Smith ได้รับจดหมายและ message digest แล้วก็จะทำการคำนวณ message digest กับจดหมายฉบับ Bad letter B ซึ่งจะได้รับผลที่สอดคล้องกับสิ่งที่ Marilyn ส่งมา ซึ่งก็คงจะไล่ Tom ออกจากงานแทนที่จะแต่งตั้งเขาเข้าตำแหน่งใหม่ Dean Smith ไม่ทราบเลยว่า Ellen ได้จัดการสร้างจดหมายขึ้นมาสองฉบับที่มี message digest เหมือนกันและจัดการส่งจดหมายฉบับ Bad letter B มาแทนที่ Good letter A ซึ่งเป็นฉบับที่ Marilyn เห็นและลงนามรับรอง วิธีการ birthday attack เมื่อนำมาใช้กับ MD5 จะมีความยุ่งยากกว่า เนื่องจากแม้ว่าจะสามารถสร้าง message digest ได้ 1 พันล้านข้อความต่อวินาทีก็จะต้องใช้เวลานานถึง 500 ปีเพื่อที่จะได้สามารถสร้างข้อความได้ครบ 264 ข้อความสำหรับจดหมายทั้งสองฉบับ และแม้ว่าจะสร้างข้อความได้ครบก็ยังไม่ได้รับประกันว่าจะพบข้อความที่ตรงกัน อย่างไรก็ตาม ถ้านำเครื่องคอมพิวเตอร์ 5000 เครื่องมาทำการประมวลผลพร้อมกัน ระยะเวลาจาก 500 ปีก็จะลดลงเหลือเพียง 5 สัปดาห์ ดังนั้นวิธี SHA-1 ก็จะดีกว่าเพราะใช้ข้อความที่ยาวกว่า



รูปที่ 8-23
วิธีการที่ทรูดี้ใช้ในการเปลี่ยนคีย์สาธารณะ

8.5 การบริหารคีย์สาธารณะ

การเข้ารหัสแบบใช้คีย์สาธารณะช่วยให้กลุ่มคนที่ไม่ได้ใช้คีย์ร่วมกันสามารถสื่อสารถึงกันได้อย่างปลอดภัย และยังเป็นวิธีที่ทำให้การลงชื่อรับรองข้อความเป็นไปได้โดยไม่ต้องอาศัยบุคคลที่สามที่ต้องไว้ใจได้เข้ามาเกี่ยวข้อง ประการสุดท้ายการลงชื่อรับรอง message digest ยังช่วยให้สามารถตรวจสอบความถูกต้องของเอกสารได้โดยง่ายดาย

อย่างไรก็ตาม มีปัญหาประการหนึ่งที่ยังไม่สามารถหาคำตอบที่ชัดเจนได้ คือ ถ้า อลิส และ บ๊อบ ไม่รู้จักกันมาก่อนแล้วทั้งสองคนจะค้นหาคีย์สาธารณะของอีกคนหนึ่งได้อย่างไรเพื่อนำมาใช้ในการสื่อสาร คำตอบของปัญหานี้คือให้ใส่คีย์สาธารณะของแต่ละคนไว้ในเว็บไซต์ของตนเอง แต่ก็ไม่สามารถนำมาใช้งานได้จริงเนื่องจากเหตุผลดังนี้ สมมุติว่า อลิสต้องการดูคีย์สาธารณะของบ๊อบจากเว็บไซต์ของเขา เธอจะทำได้อย่างไร เธออาจเริ่มต้นด้วยการป้อนที่อยู่ URL ของบ๊อบเข้าไปในเว็บบราวเซอร์ของเธอ เว็บบราวเซอร์ก็จะค้นหาใน DNS เพื่อหาที่อยู่ของบ๊อบด้วยการส่ง GET request ดังแสดงในรูป 8-23 แต่โชคไม่ดีที่ ทรูดี้ สามารถดักรับข้อความนี้ได้จึงตอบกลับไปด้วยที่อยู่โฮมเพจปลอมซึ่งอาจจะเป็นโฮมเพจที่เหมือนกับโฮมเพจของบ๊อบทุกอย่าง ยกเว้นได้ใส่คีย์สาธารณะของทรูดี้ (ET) เข้าไปแทนที่คีย์สาธารณะของบ๊อบ (EB) อลิสจึงเริ่มเข้ารหัสข่าวสารโดยใช้ ET ซึ่งทรูดี้สามารถถอดรหัสได้ อ่านข้อความนั้นได้ และจัดการเข้ารหัสข้อความนั้นด้วยคีย์สาธารณะของบ๊อบ (EB) แล้วจึงส่งไปให้บ๊อบซึ่งจะไม่มีทางทราบได้เลยว่าข้อความนั้นถูกทรูดี้อ่านไปแล้ว ยิ่งกว่านั้น ทรูดี้อาจทำการแก้ไขข้อความก่อนที่จะเข้ารหัสและส่งไปให้บ๊อบ จึงเห็นได้ชัดเจนว่าจะต้องมีวิธีการที่ปลอดภัยในการแลกเปลี่ยนคีย์สาธารณะ

8.5.1 การใช้ใบรับรอง

ความพยายามแรกที่นำมาใช้ในการส่งคีย์สาธารณะได้อย่างปลอดภัย คือ การจัดตั้งศูนย์กระจายคีย์สาธารณะ (public key distribution center) ที่เปิดทำการตลอด 24 ชั่วโมงเพื่อจัดส่งคีย์สาธารณะให้แก่ผู้ใช้ได้ตามต้องการ ปัญหาของการใช้ศูนย์แจกจ่ายคีย์สาธารณะก็คือไม่สามารถขยายตัวได้เมื่อมีผู้ใช้บริการมากขึ้น และศูนย์ฯ จะกลายเป็นจุดคอขวดของระบบสื่อสารอย่างรวดเร็ว นอกจากนี้ถ้าศูนย์มีเหตุให้ต้องปิดการให้บริการ (อาจจะเป็นการชั่วคราว) จะทำให้ระบบการรักษาความปลอดภัยในระบบอินเตอร์เน็ตทั้งหมดต้องหยุดไปด้วย

ด้วยเหตุผลเหล่านี้จึงได้มีการพัฒนาวิธีการอื่นขึ้นมาแทนที่ ซึ่งเป็นวิธีการหนึ่งที่ไม่ต้องใช้ศูนย์กระจายข่าวที่ต้องทำงานอยู่ตลอด 24 ชั่วโมง อันที่จริงวิธีการนี้ไม่มีความจำเป็นต้องทำงานแบบออนไลน์ (on-line) เลย นั่นคือใช้วิธีการออกใบรับรองคีย์สาธารณะให้แก่บุคคลทั่วไป บริษัทเอกชน และองค์กรทั่วไป องค์กรที่ทำหน้าที่ให้การรับรองคีย์สาธารณะนี้เรียกว่า CA (Certification Authority)

รูปที่ 8-24
ใบรับรองคีย์สาธารณะ
และ SHA-1 hash
ที่ลงชื่อรับรองโดย CA

ตัวอย่างเช่น สมมุติว่าบ็อบต้องการให้อลิสและคนอื่นๆ สามารถสื่อสารกับเขาได้อย่างปลอดภัย เขาก็จะไปที่ CA พร้อมกับคีย์สาธารณะและเอกสารแสดงตนเพื่อขอให้องค์กรรับรองคีย์ของเขา CA จะให้ใบรับรองซึ่งอาจจะมีลักษณะคล้ายกับที่แสดงในรูป 8-24 และลงชื่อรับรองด้วย SHA-1 hash ด้วยคีย์ส่วนตัวของ CA เอง บ็อบก็จะชำระค่าธรรมเนียมและได้รับใบรับรองพร้อมทั้งการลงชื่อรับรองด้วย SHA-1 hash ของ CA

งานพื้นฐานของใบรับรองคือการให้ข้อมูลคีย์สาธารณะควบคู่กับชื่อสาธารณะ เช่น ชื่อบริษัทหรือชื่อขององค์กร ตัวใบรับรองเองนั้นไม่ได้เป็นความลับ คือสามารถแจกจ่ายได้ บ็อบสามารถสร้างเว็บไซต์ส่วนตัวเพื่อใช้บอกคีย์สาธารณะได้เหมือนเดิมพร้อมทั้งมีเพจที่ใช้แสดงใบรับรอง CA นี้กำกับอยู่ด้วย

สมมุติว่ากลับไปสู่เหตุการณ์ที่เกิดขึ้นในรูป 8-23 อีกครั้งหนึ่ง เมื่อทรูดี้สามารถดักรับข้อความ GET request ของอลิสได้ คราวนี้เธอจะทำอย่างไร เธออาจจะทำอย่างเดิมคือการใส่คีย์สาธารณะของตนเองรวมทั้งใบรับรองของ CA เข้าไปแทนที่ของบ็อบ แต่ในทันทีที่อลิสอ่านใบรับรอง CA ก็จะทราบได้ทันทีว่าคีย์ที่ได้รับนั้นไม่ใช่ของบ็อบเพราะไม่มีชื่อของบ็อบอยู่ด้วย ทรูดี้อาจทำการเปลี่ยนแปลงชื่อของตนเองในใบรับรองให้กลายเป็นชื่อของบ็อบ (แต่ยังเป็นคีย์สาธารณะของทรูดี้อยู่) อย่างไรก็ตามเมื่ออลิสทำการคำนวณอัลกอริทึม SHA-1 โดยใช้คีย์สาธารณะของ CA ที่เป็นที่รู้จักกันโดยทั่วไปกับข้อมูลที่อยู่ในใบรับรองนั้น เธอจะได้รับค่า hash ที่แตกต่างไปจากค่า hash ที่มาพร้อมกับใบรับรอง เนื่องจากทรูดี้ไม่มีคีย์ส่วนตัวของ CA เธอจึงไม่มีทางที่จะแก้ไขค่า SHA-1 hash ให้กลายเป็นชื่อ บ็อบ แต่มีคีย์สาธารณะของทรูดี้ ด้วยวิธีการนี้ อลิสสามารถแน่ใจได้ว่าคีย์สาธารณะที่ได้รับนั้นจะเป็นของบ็อบอย่างแน่นอน วิธีการนี้ยังไม่ต้องการให้ CA จะต้องทำงานอยู่ตลอดเวลาจึงสามารถลดปัญหาคอขวดในระบบเครือข่ายลงไปได้ด้วย

ในขณะที่หน้าที่มาตรฐานของใบรับรองคือการจับคู่ชื่อองค์กรเข้ากับคีย์สาธารณะ แต่ใบรับรองสามารถนำมาใช้ในการจับคู่คีย์สาธารณะเข้ากับคุณลักษณะอื่นๆ (attribute) ได้ด้วย ตัวอย่างเช่น ใบรับรองอาจจะระบุว่า ใบรับรองนี้ให้ไว้แก่บุคคลที่มีอายุเกิน 18 ปี หรืออาจนำมาใช้ในการพิสูจน์ว่าเจ้าของคีย์ส่วนตัวนั้นไม่ใช่เยาวชนจึงสามารถเข้าไปดูเอกสารที่ไม่เหมาะสมกับเด็กได้ ทั้งนี้ โดยไม่จำเป็นต้องเปิดเผยตัวตนของเจ้าของคีย์นั้นเลย โดยทั่วไป คนที่จะถือใบรับรองนี้จะส่งใบรับรองไปยังเว็บไซต์ผู้รับผิดชอบ หรือกระบวนการที่อ่อนไหวต่อการให้บริการที่มีอายุเป็นเกณฑ์ ผู้รับผิดชอบหรือกระบวนการ

Field	Meaning
Version	Which version of X.509
Serial number	This number plus the CA's name uniquely identifies the certificate
Signature algorithm	The algorithm used to sign the certificate
Issuer	X.500 name of the CA
Validity period	The starting and ending times of the validity period
Subject name	The entity whose key is being certified
Public key	The subject's public key and the ID of the algorithm using it
Issuer ID	An optional ID uniquely identifying the certificate's issuer
Subject ID	An optional ID uniquely identifying the certificate's subject
Extensions	Many extensions have been defined
Signature	The certificate's signature (signed by the CA's private key)

รูปที่ 8-25
เขตข้อมูลหลักของ
ใบรับรอง X.509

นั้นอาจจะสร้างเลขสุ่มขึ้นมาหมายเลขหนึ่งและจัดการเข้ารหัสด้วยคีย์สาธารณะที่ปรากฏอยู่ในใบรับรองนั้น ซึ่งถ้าผู้รับข้อความสามารถถอดรหัสและส่งข้อความตอบกลับมาได้ก็แสดงว่าผู้รับนั้นเป็นผู้ที่มีคุณสมบัติตามที่ระบุไว้ในใบรับรอง

8.5.2 มาตรฐาน X.509

ถ้าทุกคนต้องการอะไรบางอย่างให้ได้รับการรับรองผ่าน CA ด้วยการรับรองแบบอื่นๆ จะทำให้การบริหารจัดการใบรับรองยุ่งยากขึ้น เพื่อแก้ปัญหานี้ มาตรฐานการออกใบรับรองจึงได้รับการพัฒนาขึ้นมาใช้งานซึ่งได้รับการรับรองโดยองค์การ ITU (International Telecommunication Union) เรียกว่ามาตรฐาน X.509 ซึ่งได้รับการนำมาใช้งานอย่างกว้างขวางในระบบอินเตอร์เน็ต ในปัจจุบันมาตรฐานนี้ได้รับการแก้ไขจนถึงรุ่นที่สามแล้วนับตั้งแต่ได้สร้างขึ้นใช้งานในปี ค.ศ. 1988

มาตรฐาน X.509 V.3 เป็นวิธีการอธิบายการออกใบรับรอง รูป 8-25 แสดงเขตข้อมูลหลักที่มีใช้ในใบรับรอง คำอธิบายที่มีให้นี้สามารถบอกให้ทราบว่าเขตข้อมูลแต่ละอันนั้นมีไว้ใช้ทำอะไร

ตัวอย่างเช่น ถ้าบ็อบทำงานอยู่ในแผนกให้กู้ยืมของธนาคาร Money Bank ที่อยู่ X.509 ของเขาอาจเป็นลักษณะเช่นนี้

/C=US/O=MoneyBank/OU=Loan/CN=Bob/

โดยที่ C = country, O = Organization, OU = Organization unit, และ CN = Common name การออกใบรับรองให้กับคุณสมบัติข้ออื่นๆ ก็ทำได้ในทำนองเดียวกัน ปัญหาที่สำคัญของ X.509 คือ ถ้าอลิสพยายามที่จะติดต่อกับ bob@moneybank.com และได้รับใบรับรองพร้อมด้วยชื่อ X.509 อาจเป็นไปได้ว่าในใบรับรองนั้นไม่ได้หมายความถึง บ็อบ คนที่เธอต้องการติดต่อด้วย โชคดีที่มาตรฐาน X.509 ในรุ่นที่ 3 นี้อนุญาตให้ใช้ชื่อใน DNS แทนที่จะเป็นชื่อ X.509 จึงทำให้ปัญหานี้หมดไป

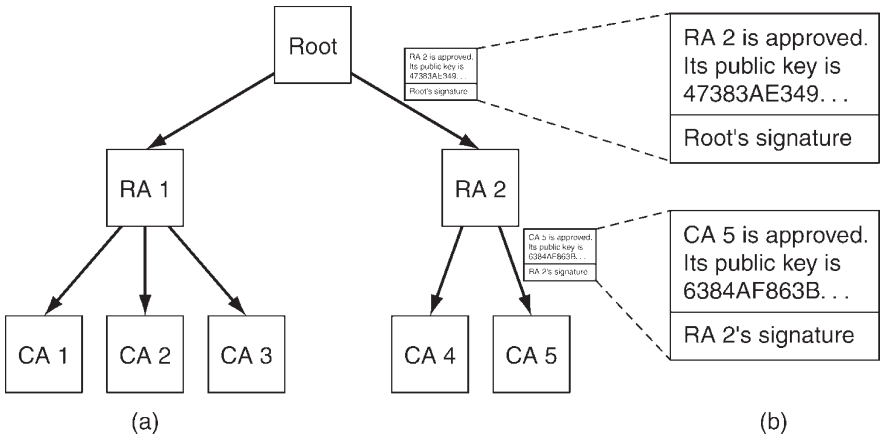
ใบรับรองได้รับการเข้ารหัสโดยใช้มาตรฐาน ASN.1 (Abstract Syntax Notation 1) ซึ่งสามารถเปรียบเทียบได้กับโครงสร้างข้อมูลแบบ "struc" ที่มีใช้ในภาษาซี (ภาษาสำหรับเขียนโปรแกรมคอมพิวเตอร์อย่างหนึ่ง)

8.5.3 โครงสร้างภายในคีย์สาธารณะ

การใช้ใบรับรอง CA เพียงแบบเดียวสำหรับการออกในรับรองทั่วโลกนั้นคงจะเป็นไปไม่ได้ เพราะคงจะไม่สามารถรองรับปริมาณงานจำนวนมหาศาลได้และคงจะล้มเหลวในที่สุด หนทางที่เป็นไปได้คือการมีศูนย์ให้บริการใบรับรองอยู่จำนวนหนึ่งซึ่งทั้งหมดอยู่ภายใต้การบริหารขององค์กรเดียวกันและทั้งหมดใช้คีย์ส่วนตัวอันเดียวกันในการลงชื่อในใบรับรอง แม้ว่าวิธีการนี้จะช่วยแก้ปัญหาปริมาณงานและความล้มเหลวที่อาจจะเกิดขึ้นได้แต่ในเวลาเดียวกันก็ได้ทำให้เกิดปัญหาใหม่ขึ้นคือปัญหาการรั่วไหลของคีย์ส่วนตัว ถ้ามีเซิบเวอร์จำนวนหลายสิบเครื่องกระจายกันอยู่ทั่วโลกซึ่งทั้งหมดมีคีย์ส่วนตัวอันเดียวกัน ก็มีโอกาสเป็นไปได้ที่คีย์ส่วนตัวของ CA อาจจะถูกขโมยไปได้ การมี CA เพียงแห่งเดียวในโลกเป็นเรื่องที่มีความเสี่ยงมากเนื่องจากปัญหาการรั่วไหลของคีย์ส่วนตัวซึ่งจะทำลายโครงสร้างการรักษาความปลอดภัยทางอิเล็กทรอนิส์

นอกจากนี้ จะให้องค์กรใดทำหน้าที่เป็น CA นั้นเป็นเรื่องที่ละเอียดอ่อนมาก เพราะเป็นการยากมากที่จะหาองค์กรหนึ่งที่จะได้รับการยอมรับอย่างกว้างขวางทั่วโลกอย่างเป็นทางการและสามารถให้ความเชื่อถือได้ ในบางประเทศประชาชนอาจให้ความไว้วางใจแก่หน่วยงานของรัฐบาลในขณะที่ประเทศอื่นอาจต้องการให้เป็นองค์กรเอกชนที่ไม่ได้เป็นของรัฐ

ด้วยเหตุผลเหล่านี้ จึงมีวิธีการต่างๆ ในการรับรองคีย์สาธารณะซึ่งใช้ชื่อว่า PKI (Public Key Infrastructure) ระบบ PKI ประกอบด้วยส่วนประกอบหลายส่วนได้แก่ ผู้ใช้, CA, ใบรับรอง, และ directories ระบบ PKI เป็นการกำหนดแนวทางในการกำหนดความสัมพันธ์ขององค์ประกอบเหล่านี้และกำหนดมาตรฐานสำหรับเอกสารและโพรโตคอลต่างๆ ตัวอย่างรูปแบบของ PKI ได้แก่โครงสร้างลำดับชั้นของ CA ดังที่แสดงในรูป 8-26 ในตัวอย่างนี้โครงสร้างแบ่งออกเป็น 3 ระดับ ซึ่งในทางปฏิบัติอาจมีมากหรือน้อยกว่านี้ก็ได้ CA ในระดับบนสุด (หรือ root) เป็นผู้ให้การรับรอง CA ที่อยู่ในระดับที่สอง ซึ่งเรียกว่า RA (Regional Authorities) เพราะว่าเป็น CA ที่อยู่ในเขตพื้นที่การให้บริการในเขตภูมิภาคต่างๆ เช่น ทวีป หรือประเทศ คำเรียกต่างๆ นี้ไม่มีมาตรฐานกำหนดไว้อย่างแน่นอน อันที่จริงแล้วไม่มีคำใดที่ถูกกำหนดให้เป็นมาตรฐานในระดับใดๆ เลย ในที่นี้ RA จะเป็นผู้ให้การรับรอง CA ซึ่งจะเป็นผู้ออกใบรับรอง X.509 ให้แก่องค์กรและบุคคลต่างๆ เมื่อ root ได้ให้การรับรองแก่ RA ใหม่ root จะสร้างใบรับรอง X.509 ซึ่งกล่าวถึงการรับรอง RA ใหม่นี้รวมทั้งส่งคีย์สาธารณะของ RA ตัวใหม่นี้ไป



รูปที่ 8-26
(a) โครงสร้างลำดับชั้นของ PKI
(b) ห่วงโซ่ของการออกใบรับรอง

ให้แก่ RA ที่มีอยู่แต่เดิมทั้งหมด ในทำนองเดียวกันเมื่อ RA ได้ให้การรับรอง CA ใหม่ก็จะสร้าง X.509 เพื่อรับรอง CA ใหม่และบอกคีย์สาธารณะแล้วกระจายข่าวไปบอกยัง CA อื่นทุกแห่ง

ระบบ PKI ทำงานดังนี้ สมมุติว่าอลิสต้องการคีย์สาธารณะของบ๊อบเพื่อจะได้สื่อสารถึงกันได้ เธอจึงค้นหาใบรับรองของบ๊อบที่ให้การรับรองโดย CA5 แต่อลิสไม่รู้จัก CA5 จึงติดต่อไปที่ CA5 เพื่อให้เขาพิสูจน์ความเป็น CA ดังนั้น CA5 จึงตอบกลับมาด้วยใบรับรองที่ตนเองได้รับมาจาก RA2 ซึ่งมีคีย์สาธารณะของ CA5 อยู่ อลิสจึงใช้คีย์สาธารณะของ CA5 ที่ได้รับมานี้ในการตรวจสอบใบรับรองของบ๊อบซึ่งให้การรับรองโดย CA5 ดังนั้นจึงสามารถเชื่อถือคีย์สาธารณะของบ๊อบได้

อาจเป็นไปได้ว่าอลิสก็ไม่รู้จัก RA2 จึงต้องส่งข้อความไปสอบถาม RA2 ซึ่งก็จะได้รับคำตอบเป็นใบรับรองที่ออกให้โดย root เธอจึงเชื่อใจใน RA2 และ CA5 ซึ่งก็จะทำให้เกิดความเชื่อถือในคีย์สาธารณะของบ๊อบในที่สุด

ปัญหาที่เกิดขึ้นคือ อลิสจะทราบคีย์สาธารณะของ root ได้อย่างไร คำตอบก็คืออลิสและทุกคนจะต้องทราบคีย์สาธารณะของ root ด้วยวิธีใดก็ตาม เช่น ทราบตอนที่สมัครเข้าเป็นสมาชิกของ PKI กลุ่มนี้ หรือการใช้เว็บบราวเซอร์ที่มีคีย์สาธารณะของ root ที่สร้างไว้ในตัวเองแล้ว

สมมุติว่าบ๊อบเป็นคนที่มีจิตใจกว้างขวางจึงไม่ต้องการให้อลิสต้องทำงานมากนัก เขาทราบว่าอลิสจะต้องตรวจสอบ CA5 และ RA2 ดังนั้นจึงได้ช่วยให้งานของเธอง่ายขึ้นด้วยการส่งใบรับรองทั้งสองใบนี้ส่งไปให้อลิสด้วยเลย อลิสจึงสามารถใช้ข้อมูลที่เธอทราบคือคีย์สาธารณะของ root ในการตรวจสอบใบรับรองทั้งสองใบ ด้วยวิธีการนี้อลิสก็ไม่จำเป็นจะต้องติดต่อใครในการพิสูจน์ตัวตนของบ๊อบเลย ใบรับรองทุกใบมีการลงชื่อกำกับเสมอ อลิสจึงสามารถตรวจสอบการแก้ไขเปลี่ยนแปลงเอกสารได้ทุกฉบับ การตรวจสอบย้อนกลับไปที่ root นี้เรียกว่า chain of trust หรือ certification path ซึ่งเป็นเทคนิคที่นำไปใช้งานอย่างกว้างขวาง

แม้ว่ายังเป็นปัญหาว่าใครจะทำหน้าที่เป็น root แต่หนทางปฏิบัติที่เป็นไปได้คือการมีหลาย root โดยที่แต่ละ root ก็จะมี RA และ CA เป็นของตนเอง อันที่จริงแล้วเว็บบราวเซอร์สมัยใหม่จะทำการติดตั้งคีย์สาธารณะของ root ที่เป็นที่รู้จักกันทั่วไปมากกว่า 100 แห่งมาด้วยแล้วซึ่งเรียก root เหล่านี้ว่า trust anchors ด้วยวิธีการนี้สามารถหลีกเลี่ยงการมี root เพียงแห่งเดียวในโลกได้

ไดเร็กทอรี่ (Directories)

ประเด็นที่น่าสนใจอีกประเด็นหนึ่งของระบบ PKI คือจะจัดเก็บใบรับรองต่างๆ รวมทั้งการใช้ใบรับรองเพื่อย้อนกลับไปตรวจสอบ root ที่เป็น trust anchors ไว้ที่ใด หนทางหนึ่งที่เป็นไปได้ก็คือให้ผู้ใช้แต่ละคนจัดเก็บใบรับรองของตนเอง แม้ว่าการกระทำเช่นนี้จะเป็นวิธีที่ปลอดภัย คือไม่มีทางเป็นไปได้ที่ผู้ใช้จะได้รับเอกสารที่ถูกแก้ไขโดยไม่มีทางตรวจสอบได้ แต่ก็เป็นวิธีที่ไม่สะดวกในการทำงาน หนทางเลือกทางหนึ่งที่ได้รับการนำเสนอคือการใช้ DNS (Domain Name Server) ทำหน้าที่เป็นผู้จัดเก็บ certificate directory เนื่องจากก่อนที่อลิสจะติดต่อกับบ๊อบ อลิสจะต้องใช้หมายเลข IP ของบ๊อบซึ่งได้รับมาจาก DNS ดังนั้น จึงควรให้ DNS จัดการส่งข้อมูลใบรับรองทั้งหมดของบ๊อบมาพร้อมกับหมายเลข IP

บางคนคิดว่านี่คือวิธีการที่สะดวกในการทำงาน ในขณะที่คนบางกลุ่มกลับต้องการให้มีเซิบเวอร์ที่จัดเก็บไดเร็กทอรี่นี้และมีหน้าที่ในการจัดการบริหารใบรับรอง X.509 โดยตรง ไดเร็กทอรี่ดังกล่าวจะช่วยในการค้นหาบริการต่างๆ ที่ใช้คุณสมบัติของ X.509 names ตัวอย่างเช่น ในทางทฤษฎีไดเร็กทอรี่ดังกล่าวสามารถนำมาใช้ตอบคำถามได้หลายอย่าง เช่น ขอให้แสดงรายชื่อของผู้ใช้ที่ชื่ออลิสที่มีอยู่ทั้งหมดที่ทำงานในแผนกฝ่ายขายที่ใดๆ ก็ตามในเขตสหรัฐและแคนนาดา

การยกเลิกการรับรอง (Revocation)

ในโลกของความเป็นจริงนั้นเต็มไปด้วยใบรับรอง เช่น พาสปอร์ต ใบขับขี่ บัตรประชาชน และอื่นๆ บางครั้งใบรับรองเหล่านี้อาจถูกยกเลิก (revoked) ก็ได้ เช่น บุคคลที่เปลี่ยนสัญชาติก็จะถูกยกเลิกบัตรประชาชนของประเทศเดิม ปัญหาเดียวกันนี้ก็อาจเกิดขึ้นในโลกของดิจิตอลได้เช่นกัน กล่าวคือองค์กรที่เป็นผู้ออกใบรับรองอาจต้องการยกเลิกการรับรองบุคคลบางคน (ที่เคยให้การรับรอง) หรืออาจต้องการยกเลิกการรับรองเนื่องจากคีย์ส่วนตัวของสมาชิกถูกเปิดเผยหรือถูกขโมยไป หรือแย่ยิ่งกว่านั้นคือคีย์ส่วนตัวของ CA ถูกเปิดเผย ดังนั้น PKI จำเป็นจะต้องมีมาตรการรองรับการยกเลิกการให้การรับรอง

ขั้นตอนแรกของกระบวนการนี้คือ จะต้องให้ CA แต่ละแห่งส่งความต้องการ CRL (Certificate Revocation List) เพื่อให้บอกหมายเลขลำดับของใบรับรองทุกใบที่ถูกยกเลิกการรับรอง เนื่องจากใบรับรองนี้ใช้บอกวันที่หมดอายุการรับรอง การตอบสนองต่อคำร้องขอ CRL จึงต้องบรรจุหมายเลขลำดับของผู้ใช้ที่ยังคงให้การรับรองอยู่ในปัจจุบัน เมื่อหมดอายุการรับรองแล้วใบรับรองนั้นก็จะใช้งานไม่ได้อีกต่อไป ดังนั้น จึงไม่มีความจำเป็นจะต้องแยกความแตกต่างระหว่างใบรับรองที่หมดอายุกับผู้ที่ถูกยกเลิกการรับรองออกจากกัน ซึ่งทั้งสองกรณี ผู้ใช้จะไม่ได้รับการรับรองอีกต่อไป

อย่างไรก็ตาม การแนะนำ CRL ขึ้นมาใช้งานทำให้ผู้ที่กำลังจะใช้ใบรับรองจะต้องร้องขอ CRL เพื่อตรวจสอบดูว่าใบรับรองนั้นถูกยกเลิกหรือไม่ ซึ่งถ้าถูกยกเลิกไปแล้วก็จะไม่สามารถใช้ใบรับรองนั้นได้อีกต่อไป อย่างไรก็ตาม แม้ว่าใบรับรองนั้นจะปรากฏอยู่ในรายการ CRL ก็อาจเป็นไปได้ว่าใบรับรองนั้นถูกยกเลิกภายหลังจากที่ใบรายการนั้นได้ถูกสร้างขึ้นมาแล้ว ดังนั้น วิธีการเดียวที่จะแน่ใจได้ก็คือการถามไปที่ CA โดยตรง และการใช้งานใบรับรองใบเดียวกันในครั้งต่อไปก็จะต้องสอบถาม CA อีกเสมอ เนื่องจากใบรับรองนั้นอาจถูกยกเลิกไปแล้ว

ความซับซ้อนอีกประการหนึ่งก็คือ ใบรับรองที่ถูกยกเลิกไปนั้นอาจได้รับการรับรองใหม่ เช่น การยกเลิกเกิดขึ้นเนื่องจากลูกค้าผู้นั้นขาดการชำระเงิน ซึ่งเมื่อลูกค้าได้ชำระเงินแล้วก็จะสามารถใช้ใบรับรองนั้นได้ต่อไป การที่จะต้องจัดการแก้ไขปัญหาการยกเลิกและการรับรองใหม่ของใบรับรองนั้นได้ทำลายคุณสมบัติที่สำคัญของการใช้ใบรับรองไป คือการที่สามารถนำใบรับรองมาใช้งานได้โดยไม่ต้องติดต่อกับ CA

CRL ควรที่จะถูกจัดเก็บไว้ที่ใด ที่ที่ดีที่สุดอาจหมายถึงที่เดียวกันกับที่ใช้ในการเก็บใบรับรอง แนวทางหนึ่งที่ใช้ได้คือการให้ CA คอยจัดส่ง CRL ออกไปอยู่อย่างสม่ำเสมอและมีไดเร็กทอรี่ที่จัดการลบใบรับรองที่ถูกยกเลิกออกไป ถ้าไดเร็กทอรี่ไม่ได้ใช้ในการจัดเก็บใบรับรอง CRL ก็อาจถูกจัดเก็บไว้ในที่ที่เหมาะสมแห่งอื่นในระบบเครือข่าย เนื่องจาก CRL เองก็เป็นเอกสารที่ได้รับการลงชื่อกำกับ ดัง

นั้นถ้าถูกแก้ไขก็จะสามารถตรวจสอบได้

8.6 การรักษาความปลอดภัยในการสื่อสาร

หลังจากที่ได้กล่าวถึงเทคนิค โพรโตคอล และเครื่องมือต่างๆ ที่นำมาใช้ในการรักษาความปลอดภัยข้อมูลแล้ว ในหัวข้อนี้และหัวข้อที่เหลือในบทนี้จะได้กล่างถึงการนำเทคนิคเหล่านี้ไปประยุกต์ใช้งาน รวมทั้งแนวความคิดในการรักษาความปลอดภัยทางด้านสังคมในตอนท้ายของบทด้วย

8.6.1 การใช้ IPsec

องค์กร IETF (Internet Engineering Task Force) ได้รับทราบมาเป็นระยะเวลานานแล้วว่าระบบอินเตอร์เน็ตนั้นมีจุดบกพร่องในเรื่องการรักษาความปลอดภัย แต่การเพิ่มการรักษาความปลอดภัยนั้นก็ไม่ใช่เรื่องที่จะทำได้ง่ายๆ เนื่องจากมีปัญหาว่าจะใส่ระบบนี้ไว้ในส่วนใด ผู้เชี่ยวชาญในเรื่องการรักษาความปลอดภัยส่วนใหญ่มีความเชื่อว่าถ้าจะให้มีความปลอดภัยจริงๆ แล้วการเข้ารหัสข้อมูลและการตรวจสอบความถูกต้องของข้อมูลจะต้องเป็นการทำงานในระดับผู้ใช้ถึงผู้ใช้ (end-to-end) เช่น ในโปรแกรมในชั้นสื่อสารโปรแกรมประยุกต์ นั่นคือ โปรเซสที่เป็นผู้ผลิตข้อมูลจะทำการเข้ารหัสและทำการป้องกันความถูกต้องของข้อมูลและส่งข้อมูลนั้นไปยังโปรเซสผู้รับซึ่งจะทำการถอดรหัสข้อมูลและทำการตรวจสอบข้อมูล การแก้ไขใดๆ ที่เกิดขึ้นในระหว่างสองโปรเซสนี้รวมทั้งที่เกิดขึ้นในระบบปฏิบัติการก็จะสามารถถูกตรวจพบได้ ปัญหาของวิธีการนี้ก็คือ จำเป็นจะต้องแก้ไขเปลี่ยนแปลงโปรแกรมประยุกต์ทั้งหมดเพื่อให้สามารถจัดการกับการรักษาความปลอดภัยข้อมูลได้ ในมุมมองนี้ สิ่งที่ดีที่สุดน่าจะเป็นการใส่การเข้ารหัสข้อมูลไว้ในชั้นสื่อสารนำส่งข้อมูลหรือในชั้นสื่อสารใหม่ที่ทำงานอยู่ระหว่างชั้นสื่อสารโปรแกรมประยุกต์และชั้นนำส่งข้อมูลทำให้ยังคงเป็นการทำงานในระดับผู้ใช้ถึงผู้ใช้แต่ไม่จำเป็นต้องแก้ไขโปรแกรมประยุกต์

ในมุมมองตรงกันข้ามก็คือ ผู้ใช้ไม่มีความเข้าใจในเรื่องการรักษาความปลอดภัยข้อมูลและไม่มีความสามารถในการรักษาความปลอดภัยได้อย่างถูกต้อง และไม่มีผู้ใดต้องการแก้ไขโปรแกรมประยุกต์ที่มีอยู่แล้วไม่ว่าเพื่ออะไรก็ตาม ดังนั้น ชั้นสื่อสารควบคุมเครือข่ายควรจะเป็นผู้ทำการตรวจสอบและเข้ารหัสข้อมูลในระดับแพ็กเก็ตโดยไม่ต้องให้ผู้ใช้เข้ามาเกี่ยวข้อง ภายหลังจากการต่อสู้ในด้านความคิดกันมานานหลายปี มุมมองนี้ก็ประสบชัยชนะเพราะได้รับการสนับสนุนมากพอที่จะกำหนดมาตรฐานในการรักษาความปลอดภัยข้อมูลให้แก่ชั้นสื่อสารควบคุมเครือข่าย เหตุผลในบางส่วนที่น่าสนใจก็คือการเข้ารหัสข้อมูลในชั้นสื่อสารควบคุมเครือข่ายนี้ไม่ได้กันให้ผู้ใช้ที่มีความสนใจในการรักษาความปลอดภัยข้อมูลจากการที่จะทำให้เกิดความปลอดภัยขึ้นอย่างถูกต้อง ในขณะเดียวกันก็ช่วยรักษาความปลอดภัยให้แก่ผู้ใช้ที่ไม่มีความสนใจในเรื่องนี้

ผลลัพธ์ที่ได้จากการโต้เถียงในเรื่องนี้คือการออกแบบที่เรียกว่า IPsec (IP security) ซึ่งได้รับการอธิบายไว้ในมาตรฐาน RFC 2401, 2402, และ 2406 ผู้ใช้บางกลุ่มก็ไม่มีความต้องการในเรื่องการรักษาความปลอดภัย เนื่องจากต้องเสียเวลาในการคำนวณมาก แต่แทนที่จะทำให้เรื่องนี้เป็นทางเลือกผู้รับผิดชอบได้ตัดสินใจให้การรักษาความปลอดภัยเป็นข้อบังคับที่จะต้องเกิดขึ้นตลอดเวลาแต่ก็ได้สร้างหนทางแก้ไขเวลาที่จะต้องเสียไปด้วยการอนุญาตให้สร้าง null algorithm (อัลกอริทึมซึ่งไม่มีการ

ทำงานใดๆ เกิดขึ้นเลย) (RFC 2410)

IPsec ที่ได้รับการออกแบบไว้อย่างสมบูรณ์เป็นโครงสร้างสำหรับการให้บริการ, อัลกอริทึม, และการทำงานในส่วนละเอียดหลายอย่าง เหตุผลที่มีบริการหลายอย่างคือไม่ใช่ทุกคนที่ต้องการจะเสียเวลาไปกับการใช้บริการทุกอย่างอยู่ตลอดเวลา ดังนั้น บริการจึงมีให้เลือกใช้ได้ตามความต้องการ บริการหลักที่มีให้ได้แก่ การรักษาความลับข้อมูล การรักษาความถูกต้องของข้อมูล และการป้องกันปัญหาการส่งข่าวสารเดิมซ้ำซาก (replay attack) บริการทั้งหมดนี้ใช้วิธีการเข้ารหัสข้อมูลแบบคีย์สมดุลย์เนื่องจากต้องการเน้นในเรื่องประสิทธิภาพในการทำงาน

เหตุผลที่ต้องมีอัลกอริทึมหลายอย่างก็เนื่องจากอัลกอริทึมที่คิดว่าปลอดภัยที่สุดในปัจจุบันอาจจะถูกถอดรหัสได้ในอนาคต เนื่องจาก IPsec นั้นสามารถเลือกใช้อัลกอริทึมได้หลายอย่างตามที่ต้องการ จึงทำให้โครงสร้างนี้ปลอดภัยแม้ว่าอัลกอริทึมบางอย่างจะไม่ปลอดภัยในเวลาต่อมาก็ตาม

เหตุผลที่มีการทำงานในส่วนรายละเอียดในหลายระดับก็คือ ทำให้สามารถเลือกให้การรักษาความปลอดภัยในระหว่างการเชื่อมต่อ TCP เพียงคู่เดียว (single TCP connection), การเชื่อมต่อทั้งหมดที่เกิดขึ้นระหว่างโฮส (traffic between a pair of hosts), หรือการเชื่อมต่อระหว่างเราเตอร์ที่ต้องการความปลอดภัย (traffic between a pair of secure routers) เป็นต้น

สิ่งที่น่าแปลกใจประการหนึ่งของ IPsec ก็คือแม้ว่าจะเป็นกระบวนการที่เกิดขึ้นในชั้นสื่อสาร IP แต่ก็เป็นการเชื่อมต่อแบบต่อเนื่อง (connection oriented) โดยแท้จริงแล้วนี่ไม่ใช่สิ่งที่น่าแปลกใจ เนื่องจากการที่จะทำให้การรักษาความปลอดภัยเกิดขึ้นได้นั้น กุญแจที่สำคัญคือการจัดตั้งการเชื่อมต่อและใช้ช่วงเวลาในการสื่อสารเข้ามาเกี่ยวข้อง ซึ่งก็คือการเชื่อมต่อแบบต่อเนื่องอย่างหนึ่งนั่นเอง การเชื่อมต่อในมุมมองของ IPsec นั้นเรียกว่า SA (security association) ซึ่งเป็นการเชื่อมต่อแบบทางเดียวระหว่างจุดที่มีการสื่อสารถึงกันสองจุด และมีการกำหนดการรักษาความปลอดภัยขึ้นมาใช้งาน ถ้าต้องการรักษาความปลอดภัยให้แก่การสื่อสารทั้งสองทาง (จากผู้ส่งไปยังผู้รับและจากผู้รับกลับไปยังผู้ส่ง) ก็จะต้องสร้าง SA ขึ้นมาคู่หนึ่ง ตัวกำหนดการรักษาความปลอดภัย (security identifier) จะถูกใส่เข้าไปในแพ็กเก็ตที่เดินทางผ่านช่องทางสื่อสารที่มีการรักษาความปลอดภัยและถูกใช้สำหรับการค้นหาคีย์และข่าวสารอื่นๆ ที่สำคัญเมื่อแพ็กเก็ตเดินทางมาถึงผู้รับ

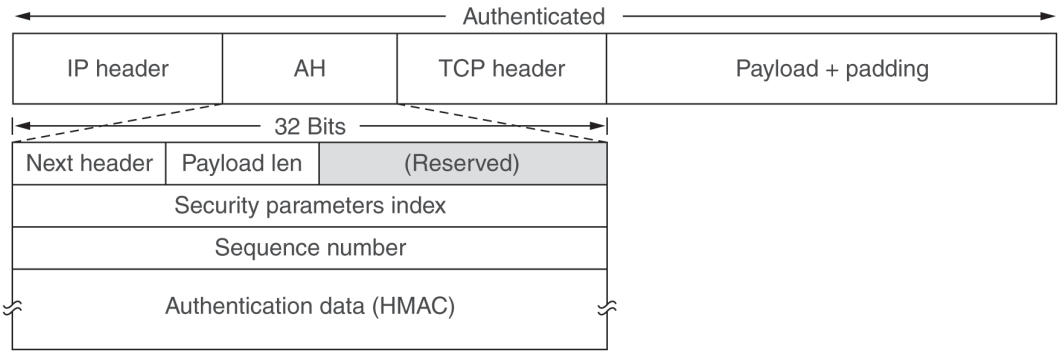
ในทางเทคนิคแล้ว IPsec ประกอบด้วยสองส่วน ส่วนแรกคือส่วนที่อธิบายข้อมูลส่วนหัวที่เพิ่มเติมเข้าไปกับแพ็กเก็ตเพื่อใช้ในการเก็บรักษาตัวกำหนดการรักษาความปลอดภัย (security identifier), ข้อมูลสำหรับการตรวจสอบความถูกต้องของข้อมูล (integrity control), และข่าวสารที่จำเป็นอื่นๆ ส่วนที่สองคือ ISAKMP (Internet Security Association and Key Management Protocol) ซึ่งจะเป็นตัวจัดการเกี่ยวกับการจัดตั้งคีย์

IPsec สามารถทำงานได้ในสองโหมด ในโหมด transport mode ข้อมูลส่วนหัวของ IPsec จะถูกใส่เข้าไปในแพ็กเก็ตหลังจากตำแหน่งของข้อมูลส่วนหัวของ IP (IP header) เขตข้อมูล Protocol ใน IP header จะถูกเปลี่ยนเพื่อบอกให้ทราบว่าข้อมูลส่วนที่ตามหลังมาคือ IPsec header (อยู่ในตำแหน่งก่อนหน้า TCP header) IPsec header นั้นมีข้อมูลเกี่ยวกับการรักษาความปลอดภัย, SA identifier, หมายเลขลำดับ (ที่กำหนดขึ้นมาใหม่), และข้อมูลสำหรับการตรวจสอบความถูกต้องของข้อมูล

ในโหมด tunnel mode ข้อมูลทั้งแพ็กเก็ตรวมทั้งข้อมูลส่วนหัวด้วยจะถูกห่อหุ้มไว้ในแพ็กเก็ต IP ใหม่ที่มีข้อมูลส่วนหัวใหม่ด้วย โหมดนี้มีประโยชน์ในเมื่ออุโมงค์สื่อสาร (tunnel) ไปจบสิ้นที่อื่นนอกเหนือไปจากที่อยู่ของผู้รับข้อมูล ในบางกรณีจุดสิ้นสุดของอุโมงค์สื่อสารคือเครื่องเกตเวย์ที่ทำหน้าที่ในการรักษาความปลอดภัย เช่น ระบบไฟร์วอลขององค์กร ในโหมดนี้ ไฟร์วอลจะห่อหุ้มและดึงเอาแพ็กเก็ตข้อมูลออกมาเมื่อแพ็กเก็ตเดินทางผ่านไฟร์วอล และการที่กำหนดให้อุโมงค์สื่อสารไปจบสิ้นที่เครื่องรักษาความปลอดภัยนี้ ทำให้เครื่องคอมพิวเตอร์ในระบบ LAN ขององค์กรไม่จำเป็นต้องรู้สึกถึงการมีอยู่ของ IPsec นอกจากเครื่องไฟร์วอลเท่านั้น

Tunnel mode มีประโยชน์เมื่อมีการเชื่อมต่อ TCP เกิดขึ้นจำนวนหนึ่งที่ถูกนำมารวมเข้าด้วยกันและจัดการเสมือนกระแสข้อมูลเพียงกระแสเดียวที่ถูกนำมาเข้ารหัส เพราะสามารถป้องกันผู้บุกรุกไม่ให้มองเห็นว่าใครกำลังส่งแพ็กเก็ตไปยังผู้ใดในจำนวนมากน้อยเท่าใด ในบางครั้งการรู้เพียงแค่ปริมาณข้อมูลที่สื่อสารถึงกันและสื่อสารถึงใครนั้นก็กลายเป็นข้อมูลที่มีค่า (สำหรับผู้บุกรุก) ได้ ตัวอย่างเช่น ในระหว่างการปฏิบัติงานทางทหารที่สำคัญ ปริมาณข้อมูลที่ไหลระหว่าง Pentagon (ที่ตั้งกระทรวงกลาโหมสหรัฐอเมริกา) และ White House (ที่ตั้งทำเนียบประธานาธิบดี) นั้นตกลงไปอย่างมาก แต่ในขณะเดียวกันปริมาณข้อมูลระหว่าง Pentagon กับที่ตั้งทางทหารแห่งหนึ่งกลับสูงขึ้นอย่างมากอาจทำให้ข้าศึกหรือผู้บุกรุกสามารถเดาได้ว่าประธานาธิบดีคงจะย้ายออกจาก White House ไปอยู่ที่ที่ตั้งทหารแห่งนั้นแล้วก็ได้ การศึกษาปริมาณการไหลของแพ็กเก็ตข้อมูลแม้ว่าจะเป็นข้อมูลที่ถูกเข้ารหัสแล้วก็ตาม เรียกว่า traffic analysis และการสื่อสารในโหมด tunnel mode ช่วยป้องกันการศึกษาข้อมูลในแนวทางนี้ได้ในระดับหนึ่ง ข้อเสียของ tunnel mode ก็คือเป็นการเพิ่ม IP header พิเศษให้แก่ข้อมูลทำให้แพ็กเก็ตข้อมูลมีขนาดใหญ่มากขึ้นเป็นอย่างมาก ในทางกลับกัน transport mode จะไม่ส่งผลให้ขนาดของแพ็กเก็ตใหญ่ขึ้น

ข้อมูลส่วนหัวตัวใหม่คือ AH (Authentication Header) เป็นการสนับสนุนการตรวจสอบความถูกต้องของข้อมูลและป้องกันปัญหาการส่งข้อมูลซ้ำซ้อน แต่ไม่เกี่ยวข้องกับการรักษาความปลอดภัยข้อมูล เช่น ไม่มีการเข้ารหัสข้อมูล การนำ AH ไปใช้งานใน transport mode นั้นได้แสดงให้เห็นในรูป 8-27 ใน IPv4 ข้อมูล AH จะอยู่ระหว่าง IP header (รวมทั้งตัวเลือกอื่นๆ) และ TCP header ใน IPv6 ส่วนของ AH กลายมาเป็นข้อมูลส่วนขยายของ header อันที่จริง รูปแบบนี้ใกล้เคียงกับรูปแบบข้อมูลหัวส่วนขยาย (extension header) ของมาตรฐาน IPv6 ข้อมูลจริงที่ใส่เข้ามาในแพ็กเก็ตจะถูกขยาย



รูปที่ 8-27 ข้อมูล IPsec ที่ใช้ตรวจสอบผู้ใช้ใน transport mode ในข้อมูลส่วนหัวสำหรับ IPv4

ออกให้มีความยาวในระดับหนึ่งเพื่อประโยชน์ในการตรวจสอบผู้ใช้ดังแสดงในรูป 8-27

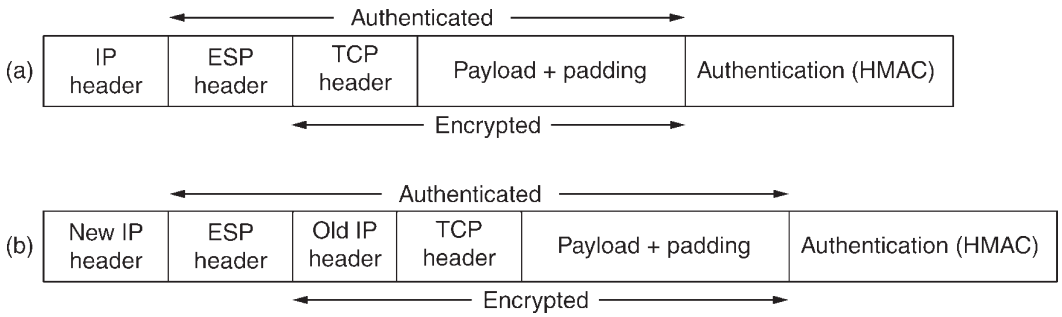
ต่อไปทำการสำรวจ AH header เขตข้อมูล Next header field ใช้ในการจัดเก็บค่าเดิมที่อยู่ในเขตข้อมูล IP protocol ก่อนที่จะถูกแทนที่ด้วยค่า "51" เพื่อเป็นการบอกให้ทราบว่า AH header นั้นอยู่ตามหลัง IP header ส่วนใหญ่โค้ดสำหรับ TCP จะถูกเก็บไว้ที่นี่ เขตข้อมูล Payload length เป็นตัวเลขขนาด 32 บิตที่บอกขนาดของข้อมูลจริง (ต้องลบออกด้วย 2 เสมอ) ที่อยู่ในแพ็กเก็ต

เขตข้อมูล Security parameters index คือ ตัวกำหนดค่าการเชื่อมต่อ (connection identifier) ถูกใส่เข้าไปโดยผู้ส่งข้อมูลเพื่อบอกให้ทราบถึงระเบียนข้อมูลในฐานข้อมูลของผู้รับที่ต้องการ ระเบียนข้อมูลนี้ประกอบด้วยคีย์ที่ใช้งานร่วมกันในระหว่างการเชื่อมต่อครั้งนี้และข่าวสารที่สำคัญอื่น ๆเกี่ยวกับการเชื่อมต่อ ถ้าโพรโตคอลนี้ถูกสร้างขึ้นมาโดย ITU แทนที่จะเป็น IETF แล้วเขตข้อมูลนี้ก็จะถูกเรียกว่า Virtual circuit number

เขตข้อมูล Sequence number ถูกใช้สำหรับนับจำนวนแพ็กเก็ตที่ถูกส่งออกมาทาง SA ข้อมูลทุกแพ็กเก็ตจะมีหมายเลขเฉพาะของตนเองแม้ว่าจะเป็นการส่งซ้ำก็ตาม หรืออีกนัยหนึ่งก็คือการจัดส่งแพ็กเก็ตซ้ำจะได้รับการกำหนดหมายเลขให้ใหม่ที่แตกต่างไปจากหมายเลขเดิม (แม้ว่าหมายเลข TCP จะยังคงเป็นหมายเลขเดิม) วัตถุประสงค์ของเขตข้อมูลนี้ก็เพื่อจะตรวจหาแพ็กเก็ตที่ถูกผู้บุกรุกดักจับแพ็กเก็ตแล้วจัดการจัดส่งซ้ำ หมายเลขนี้อาจจะไม่ถูกวนกลับมาใช้ใหม่ ถ้าตัวเลขทั้ง 232 หมายเลขถูกนำมาใช้หมดแล้วจะต้องสร้าง SA ขึ้นมาใหม่เพื่อจะได้สามารถสื่อสารได้ต่อไป

ประการสุดท้าย เขตข้อมูล Authentication data ซึ่งมีความยาวไม่คงที่จะใช้ในการบรรจุลายเซ็นต์อิเล็กทรอนิกส์ของข้อมูลจริง เมื่อได้จัดตั้ง SA ขึ้นมาแล้ว ทั้งสองฝ่ายจะทำการต่อรองการเลือกใช้อัลกอริทึมสำหรับทำลายเซ็นต์อิเล็กทรอนิกส์ โดยปกติการเข้ารหัสและคีย์สาธารณะจะไม่ถูกเลือกมาใช้เนื่องจากแพ็กเก็ตจะต้องได้รับการประมวลผลอย่างรวดเร็ว แต่อัลกอริทึมแบบคีย์สาธารณะที่รู้จักกันโดยทั่วไปนั้นทำงานได้ช้ามาก เนื่องจาก IPsec นั้นทำการเข้ารหัสด้วยการใช้คีย์สมมาตร ซึ่งทั้งผู้รับและผู้ส่งข้อมูลจะต้องตกลงการใช้คีย์ร่วมกันก่อนที่จะจัดตั้ง SA ขึ้นใช้งาน คีย์ร่วมจึงถูกนำมาใช้ในการทำลายเซ็นต์อิเล็กทรอนิกส์ด้วย วิธีการที่ง่ายวิธีหนึ่งคือการคำนวณค่า hash ของแพ็กเก็ตโดยใช้คีย์ที่ใช้ร่วมกัน และคีย์ร่วมนั้นก็จะไม่ถูกส่งออกไปพร้อมกับข้อมูลอย่างแน่นอน วิธีการทำงานเช่นนี้เรียกว่า HMAC (hashed Message Authentication Code) ซึ่งเป็นวิธีที่ทำงานได้เร็วมากกว่าการคำนวณ SHA-1 ร่วมกับ RSA ที่กล่าวถึงมาแล้ว

AH header จะไม่ทำการเข้ารหัสข้อมูล ดังนั้น จึงเป็นวิธีการที่มีประสิทธิภาพมากเมื่อต้องการ



รูปที่ 8-28 (a) ESP ใน transport mode (b) ESP ใน tunnel mode

การตรวจสอบความถูกต้องของข้อมูลแต่ไม่ต้องการการรักษาความปลอดภัย คุณสมบัติที่สำคัญประการหนึ่งของ AH คือการตรวจสอบความถูกต้องของข้อมูลนั้นจะครอบคลุมข้อมูลบางส่วนใน IP header ด้วย ข้อมูลดังกล่าวจะเป็นส่วนที่ไม่มีการเปลี่ยนแปลงเมื่อแพ็กเก็ตถูกส่งจากเราเตอร์ตัวหนึ่งไปยังเราเตอร์ อีกตัวหนึ่ง เช่น Time-to-live นั้นจะเปลี่ยนแปลงค่าไปเรื่อยๆ ดังนั้นจะไม่ถูกนำมาตรวจสอบด้วย แต่ IP source address จะถูกรวมอยู่ในการตรวจสอบซึ่งทำให้ผู้บุกรุกไม่สามารถแก้ไขข้อมูลที่มีความสำคัญได้

ข้อมูล IPsec header อีกทางเลือกหนึ่งคือ ESP (Encapsulating Security Payload) ซึ่งมีประโยชน์มากสำหรับทั้ง transport mode และ tunnel mode แสดงให้เห็นในรูป 8-28

ESP header ประกอบด้วยข้อมูลขนาด 32 บิตจำนวน 2 words ซึ่งถูกเรียกว่าเขตข้อมูล Security parameter index และ Sequence number ดังที่ได้กล่าวถึงใน AH ไปแล้ว ข้อมูล word ที่สามที่มักจะอยู่ตามหลัง (ในทางเทคนิคแล้วไม่ถือว่าเป็นส่วนหนึ่งของ header) เรียกว่า Initialization vector ที่นำมาใช้สำหรับการเข้ารหัสข้อมูล ยกเว้นในกรณีที่ใช้ null algorithm (ไม่มีการเข้ารหัสข้อมูล) จึงจะไม่มีการบันทึกลงไป

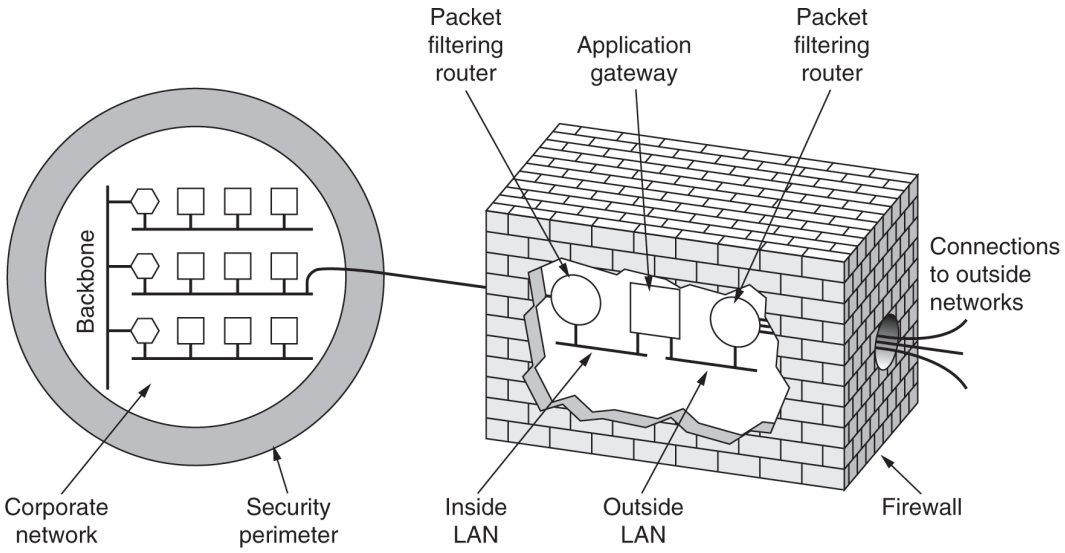
ESP ยังสนับสนุนการตรวจสอบความถูกต้องของข้อมูลแบบ HMAC เหมือนกับที่ AH ให้การสนับสนุน แต่แทนที่จะรวมเข้าไว้ใน header ก็จะใส่ไว้ในส่วนข้อมูลจริง (payload) ดังที่แสดงในรูป 8-28 การใส่ HMAC ไว้ที่ส่วนสุดท้ายทำให้เกิดข้อได้เปรียบในการสร้างฮาร์ดแวร์ขึ้นใช้งาน HMAC อาจถูกคำนวณในขณะที่บิตข้อมูลกำลังถูกส่งออกไปในสายสื่อสารและส่งเพิ่มเติมเป็นข้อมูลต่อท้ายได้ นี่คือเหตุผลที่ทำไมระบบ Ethernet และระบบ LAN แบบอื่นๆ จึงใส่ข้อมูล CRC เข้าไว้ที่ส่วนท้ายแพ็กเก็ตแทนที่จะเป็น header ด้วยวิธี AH แพ็กเก็ตจะถูกเก็บไว้ในบัฟเฟอร์และทำการคำนวณลายเซ็นต์อิเล็กทรอนิกส์ก่อนที่จะส่งแพ็กเก็ตนั้นออกไป ซึ่งจะสามารถช่วยลดจำนวนแพ็กเก็ตที่จะต้องส่งออกไปได้

เนื่องจาก ESP สามารถทำได้ทุกอย่างที่ AH สามารถทำได้และสามารถทำได้มากกว่าด้วย ปัญหาจึงเกิดขึ้นว่าเหตุใดจึงต้องมี AH ไว้ใช้งาน คำตอบอยู่ที่ประวัติที่ผ่านมา ในตอนเริ่มต้น AH สามารถจัดการได้เฉพาะการตรวจสอบความถูกต้องของข้อมูลในขณะที่ ESP ทำการรักษาความปลอดภัยให้แก่ข้อมูล ต่อมาการตรวจสอบความถูกต้องข้อมูลได้ถูกเพิ่มเติมเข้าไปใน ESP แต่ผู้ที่คิดค้น AH ก็ไม่ต้องการเห็น AH ถูกยกเลิกไป อย่างไรก็ตาม สิ่งเดียวที่ยังถกเถียงกันอยู่คือการที่ AH ตรวจสอบข้อมูลบางส่วนของ IP header ในขณะที่ ESP ไม่ได้ทำซึ่งเป็นข้อถกเถียงที่ฟังไม่ขึ้น ข้อถกเถียงอีกประการหนึ่งคือการที่ AH ไม่ได้เข้าไปยุ่งเกี่ยวกับการรักษาความปลอดภัยจึงทำให้ผลิตภัณฑ์ AH สามารถผลิตเพื่อการส่งออกได้ อย่างไรก็ตาม AH คงจะถูกยกเลิกไปในที่สุด

8.6.2 การใช้ไฟร์วอล

ความสามารถในการเชื่อมต่อเครื่องคอมพิวเตอร์ใดๆ เข้ากับคอมพิวเตอร์เครื่องอื่นได้ในทุกที่ที่ต้องการเป็นความสามารถที่มีทั้งผลดีและผลเสีย สำหรับผู้ใช้งานตามบ้านการได้ท่องไปยังส่วนต่างๆ ในระบบอินเตอร์เน็ตเป็นสิ่งที่สนุกสนาน แต่สำหรับผู้รักษาความปลอดภัยในองค์กรแล้วนั่นคือฝันร้ายเลยทีเดียว องค์กรส่วนใหญ่มีข้อมูลปริมาณมากที่มีการใช้งานแบบออนไลน์ (on-line) ซึ่งเป็นความลับ

รูปที่ 8-29
ไฟร์วอลที่ประกอบด้วย packet filter จำนวน 2 ตัวและ application gateway



ทางการค้า แผนการผลิตสินค้าใหม่ กลยุทธทางการตลาด การวิเคราะห์ทางการเงิน และอื่นๆ การเปิดเผยข้อมูลเหล่านี้ออกไปให้คู่แข่งทางการค้าทราบเท่ากับเป็นการฆ่าตัวตายเลยทีเดียว

นอกเหนือจากอันตรายจากการรั่วไหลของข้อมูลแล้ว ยังมีอันตรายที่มากับข้อมูลที่ถูกใส่เข้ามาในระบบด้วย นั่นก็คือ ไวรัส วอร์ม และอันตรายจากโปรแกรมอันตรายต่างๆ ที่เข้ามาทำลายการรักษาความปลอดภัย ทำอันตรายกับข้อมูลที่มีค่า และทำให้สิ้นเปลืองเวลาอันมีค่ายิ่งของผู้ดูแลระบบที่พยายามที่จะทำให้ระบบอยู่ในสภาพที่ดี โดยทั่วไปโปรแกรมอันตรายเหล่านี้มักจะเกิดขึ้นจากพนักงานที่ไม่มีความระมัดระวังในการใช้งานหรือพนักงานที่พยายามจะสร้างสถานะการณ์เลวร้ายให้เกิดขึ้นในองค์กร

ผลที่ตามมาคือ จำเป็นจะต้องมีกลไกที่จัดการเก็บรักษาข้อมูลที่ดีเอาไว้ในขณะที่กำจัดข้อมูลที่ไม่ดีออกไปจากระบบคอมพิวเตอร์ วิธีการที่ดีอันหนึ่งคือการใช้ IPsec ดังที่ได้กล่าวไปแล้ว วิธีการนี้จะช่วยป้องกันข้อมูลในการส่งออกไปยังที่อื่นที่ปลอดภัย อย่างไรก็ตาม IPsec ไม่ได้ทำอะไรเลยในการป้องกันไม่ให้โปรแกรมอันตรายทั้งหลายเข้าสู่ระบบ LAN ขององค์กร วิธีการที่ดีกว่าในการจัดการปัญหาเหล่านี้คือการใช้ไฟร์วอล

ไฟร์วอล (firewalls) เป็นการปรับตัวของวิธีการรักษาความปลอดภัยแบบเก่า ไฟร์วอลบังคับให้ทุกคนเข้าหรือออกจากองค์กรผ่านประตูหรือเส้นทางที่จัดเตรียมไว้ให้เท่านั้น ซึ่งจะทำให้ระบบสามารถตรวจสอบข้อมูลที่จะส่งเข้าหรือออกจากองค์กรได้ ดังที่แสดงในรูป 8-29

ไฟร์วอลในลักษณะนี้ประกอบด้วยสองส่วนคือ เราเตอร์สองตัวที่ทำการตรวจสอบแพ็กเก็ตข้อมูล packet filtering และ application gateway โครงสร้างแบบที่ง่ายกว่านี้ก็อาจเป็นไปได้แต่ข้อได้เปรียบของโครงสร้างที่นำเสนอนี้คือข้อมูลทุกแพ็กเก็ตจะต้องถูกส่งผ่านเราเตอร์สองตัวซึ่งทำหน้าที่ในการกรองข้อมูลและจะต้องผ่าน application gateway เพื่อที่จะได้สามารถส่งข้อมูลออกไปนอกระบบหรือเข้ามาในระบบได้ โดยไม่มีเส้นทางอื่นให้เลือก

Packet filter แต่ละตัวคือเราเตอร์มาตรฐานทั่วไปที่ติดตั้งหน้าที่พิเศษเข้าไปด้วย นั่นคือการ

ตรวจสอบข้อมูลที่ถูกส่งเข้ามาหรือส่งออกไป แพ็กเก็ตที่มีลักษณะถูกต้องตามข้อกำหนดจะถูกส่งต่อไปได้ตามปกติ ส่วนที่ไม่ผ่านข้อกำหนดก็จะถูกทำลายทิ้งไป

รูป 8-29 เราเตอร์ที่อยู่ด้านในของระบบ LAN จะทำหน้าที่กรองข้อมูลที่จะถูกส่งออกไป ส่วนเราเตอร์ตัวที่อยู่นอกระบบ LAN จะทำหน้าที่ตรวจสอบแพ็กเก็ตที่จะถูกส่งเข้ามาในระบบ LAN แพ็กเก็ตที่ผ่านการตรวจขั้นต้นจากเราเตอร์ ทั้งขาเข้าและขาออก จะถูกส่งไปตรวจสอบอีกขั้นตอนหนึ่งที่ application gateway การที่ต้องติดตั้งเราเตอร์ไว้สองตัวก็เพื่อให้แน่ใจได้ว่าจะไม่มีแพ็กเก็ตใดสามารถเดินทางเข้าหรือออกจากระบบได้โดยไม่ถูกตรวจสอบจาก application gateway เนื่องจากไม่มีเส้นทางอื่นให้เลือก

Packet filter โดยทั่วไปจะทำงานโดยอาศัยการกำหนดเงื่อนไขผ่านตารางข้อมูลที่ผู้ดูแลระบบเป็นผู้กำหนดขึ้นใช้งาน ตารางข้อมูลเหล่านี้ประกอบด้วยข้อมูลสามชนิดคือ (1) จะแสดงที่อยู่ของที่มาของข้อมูลและเป้าหมายที่จะถูกส่งออกไปซึ่งเป็นสถานที่ที่ได้รับการตรวจสอบและอนุญาตให้ใช้งานได้ (2) รวมทั้งที่อยู่ของผู้ส่งและผู้รับที่ไม่อนุญาตให้ใช้งานผ่านระบบเครือข่ายขององค์กร และ (3) กฎเกณฑ์ที่จะนำมาใช้ว่าจะต้องทำอย่างไรกับแพ็กเก็ตข้อมูลที่จะส่งออกนอกระบบหรือส่งเข้ามาในระบบ

ในกรณีทั่วไปที่สร้างขึ้นมาสำหรับระบบ TCP/IP ที่อยู่ของผู้ส่งและที่อยู่ของผู้รับประกอบด้วย IP address และหมายเลขพอร์ตที่ใช้ (เป็นตัวกำหนดบริการที่เรียกใช้จากโพรโตคอล TCP/IP เช่น พอร์ต 23 คือ telnet พอร์ต 79 คือ finger และพอร์ต 119 คือ USENET เป็นต้น) องค์กรสามารถกั้นแพ็กเก็ตที่เดินทางเข้ามาสำหรับทุก IP address ที่ต้องการใช้บริการในพอร์ตใดก็ได้ ด้วยวิธีการนี้บุคคลที่อยู่นอกองค์กรจะไม่สามารถ log-in เข้ามาใช้งานในระบบได้โดยการเรียกใช้บริการ telnet จากพอร์ต 23 ได้ ยิ่งกว่านี้ องค์กรยังสามารถป้องกันไม่ให้พนักงานในองค์กรมัวแต่อ่านข่าวใน USENET ทั้งวันโดยไม่ทำงานได้โดยการปิดพอร์ต 119

การกั้นไม่ให้ส่งข้อมูลออกไปนั้นเป็นการทำงานที่ซับซ้อนกว่าเพราะว่าไซต์ส่วนใหญ่จะใช้พอร์ตมาตรฐาน ยิ่งกว่านี้บริการที่สำคัญบางอย่าง เช่น FTP (file transfer protocol) หมายเลขพอร์ตจะถูกกำหนดให้ใช้ด้วยการเปลี่ยนแปลงหมายเลขไปเรื่อยๆ นอกจากนี้การกั้นการเชื่อมต่อ TCP นั้นยากยิ่งขึ้นไปอีก เช่น การกั้นแพ็กเก็ต UDP นั้นยากมากเพราะไม่ทราบว่าว่าแพ็กเก็ตนั้นจะทำอะไร Packet filter ส่วนใหญ่จึงไม่อนุญาตให้ใช้ UDP เลย

ส่วนที่สองของไฟร์วอลคือ application gateway แทนที่จะมองดูที่ข้อมูลดิบที่ส่งมาในแพ็กเก็ตแต่เพียงอย่างเดียวเกตเวย์จะทำงานในระดับชั้นสื่อสารโปรแกรมประยุกต์ ตัวอย่างเช่น เมลล์เกตเวย์ (mail gateway) สามารถที่จะจัดตั้งขึ้นมาเพื่อตรวจสอบแต่ละข่าวสารที่อยู่ในเมลล์ทุกฉบับทั้งขาเข้าและขาออก เมลล์แต่ละฉบับจะถูกตรวจสอบจากข้อมูลที่ปรากฏอยู่ใน header field หรือขนาดของข่าวสาร หรือแม้กระทั่งเนื้อหาที่อยู่ในเมลล์ เช่น ในหน่วยที่ตั้งทางทหารถ้าหากตรวจพบว่ามีคำว่า "นิวเคลียร์" "ระเบิด" หรืออื่นๆ เกตเวย์ก็สามารถที่จะแจ้งเตือนไปยังผู้รับผิดชอบได้เพื่อหาทางจัดการกับเมลล์ฉบับนั้น

การติดตั้ง application gateway สามารถทำได้ทั้งที่มีโปรแกรมเดียวหรือมีหลายโปรแกรมทำงานพร้อมกัน แต่ในบางสถานการณ์องค์กรอาจไม่อนุญาตให้รับหรือส่งอีเมลล์ได้ หรืออาจจะอนุญาตให้ใช้ World Wide Web ได้แต่ตัดบริการอย่างอื่นทิ้งทั้งหมด เมื่อนำวิธีการนี้มารวมกับการ

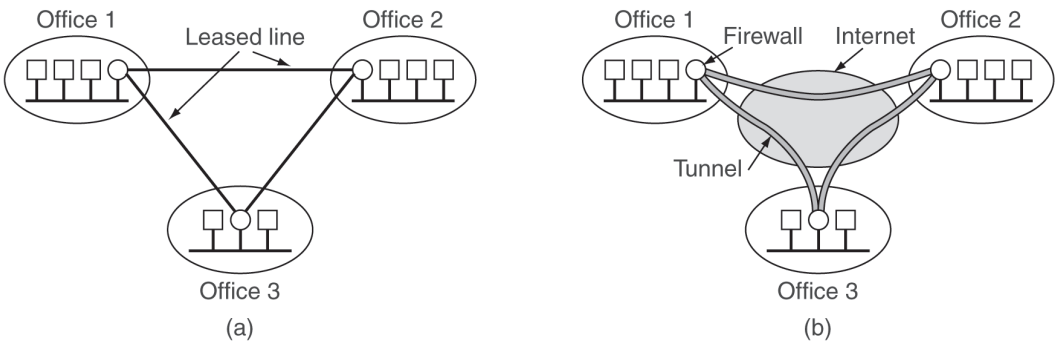
เข้ารหัสข้อมูลและ packet filtering แล้วจะช่วยให้มีความปลอดภัยมากขึ้นแต่ก็จะมีความสะดวกสบายในการทำงานลดลง

แม้ว่าระบบไฟร์วอลจะได้รับการติดตั้งอย่างสมบูรณ์ แต่ก็ยังคงมีปัญหาในด้านการรักษาความปลอดภัยอยู่อีกพอสมควร ตัวอย่างเช่น ถ้าระบบไฟร์วอลถูกกำหนดให้ยินยอมให้รับเฉพาะแพ็กเก็ตที่ส่งมาจากระบบเครือข่ายบางแห่ง (เช่นจากโรงงานอีกแห่งหนึ่งของบริษัทนั้น) ผู้บุกรุกอาจใช้วิธีการเปลี่ยนแปลงหมายเลขที่อยู่ของ source address ในแพ็กเก็ตที่ส่งเข้ามาเพื่อให้สามารถผ่านการตรวจสอบนี้ไปได้ ถ้าผู้บุกรุกต้องการส่งเอกสารลับที่ใส่ไว้ในแพ็กเก็ตเพื่อส่งออกไปนอกองค์กร เขาอาจใช้วิธีการเข้ารหัสข้อความหรือใช้วิธีการเปลี่ยนรูปแบบเอกสารให้กลายเป็นภาพกราฟฟิก เช่น JPEG เพื่อให้สามารถหลบรอดการกรองข้อความของ application gateway ไปได้ นอกจากนี้จากข้อเท็จจริงพบว่า 70% ของการบุกรุกหรือการโจมตีระบบเกิดขึ้นจากคนภายในองค์กรเอง

นอกจากนี้ยังมีวิธีการโจมตีอย่างอื่นอีกหลายอย่างที่ระบบไฟร์วอลไม่สามารถช่วยได้ แนวความคิดพื้นฐานของระบบไฟร์วอลคือการป้องกันผู้บุกรุกไม่ให้เข้าสู่ระบบและป้องกันไม่ให้ส่งข้อมูลอันเป็นความลับขององค์กรถูกส่งออกไปนอกระบบ อย่างไรก็ตาม ยังมีคนจำนวนหนึ่งที่มีความต้องการแปลก ๆ คือต้องการให้ระบบของผู้อื่นล้มเหลว เขาสามารถทำได้ง่าย ๆ ด้วยการสร้างแพ็กเก็ตที่ถูกต้องตามเงื่อนไขขึ้นมาแล้วจัดส่งแพ็กเก็ตประเภทนี้จำนวนมหาศาลเพื่อส่งเข้ามายังระบบหนึ่ง ระบบนั้นเมื่อได้รับแพ็กเก็ตจำนวนมาก (แม้ว่าจะทำเพียงแค่การลบทิ้งเท่านั้น) เมื่อถึงจุดหนึ่งก็จะไม่สามารถรับงานได้อีกต่อไปและทำให้ระบบล้มเหลวในที่สุด ตัวอย่างเช่น ถ้าต้องการโจมตีเว็บไซต์แห่งหนึ่งผู้บุกรุกสามารถส่ง TCP SYN แพ็กเก็ตมาเพื่อขอจัดตั้งการเชื่อมต่อ เว็บไซต์นั้นก็จะจัดเนื้อที่ในตารางข้อมูลให้เพื่อการเชื่อมต่อและจัดการส่งแพ็กเก็ต SYN และ ACK ตอบกลับมา ถ้าผู้บุกรุกไม่โต้ตอบกลับไปเนื้อที่ในตารางข้อมูลนั้นจะถูกสำรองเอาไว้ชั่วระยะเวลาหนึ่ง (สองถึงสามวินาที) ก่อนที่จะยกเลิกการสำรองเนื้อที่นั้น ถ้าหากว่าผู้บุกรุกส่งแพ็กเก็ตเพื่อขอการเชื่อมต่อเข้ามาทีเดียวหลายพันรายการพร้อม ๆ กัน เนื้อที่ในตารางข้อมูลก็จะถูกจองจนเต็มทั้งหมดทำให้ไม่สามารถให้บริการแก่ผู้อื่นได้อีกต่อไป วิธีการโจมตีแบบนี้มีวัตถุประสงค์ที่จะปิดการให้บริการของเป้าหมายแทนที่จะขโมยข้อมูล เรียกการโจมตีเช่นนี้ว่า DoS (Denial of Service) attack โดยปกติแพ็กเก็ตพวกนี้จะใช้ที่อยู่ปลอมใส่ไว้ในแพ็กเก็ตจึงทำให้ไม่สามารถทราบแหล่งที่มาได้

ในกรณีที่รุนแรงกว่านี้อาจเกิดขึ้นเมื่อผู้บุกรุกสามารถแทรกตัวเข้าไปสู่ระบบคอมพิวเตอร์นับร้อยแห่งจากทั่วโลกได้แล้ว จากนั้นจึงส่งคำสั่งจากระบบคอมพิวเตอร์ทั้งหมดให้พุ่งมาที่ระบบเป้าหมายเพียงระบบเดียวในเวลาเดียวกัน วิธีการนี้นอกจากจะเพิ่มขีดความสามารถในการโจมตีให้แก่ผู้บุกรุกแล้วยังช่วยลดโอกาสในการตามหาตัวผู้กระทำได้อีกด้วยเนื่องจากแพ็กเก็ตถูกส่งมาจากเครื่องคอมพิวเตอร์จำนวนมากซึ่งเป็นของระบบที่ไม่ใช่ผู้ต้องสงสัย การโจมตีในลักษณะนี้เรียกว่า DDoS (Distributed Denial of Service) attack ซึ่งเป็นการโจมตีที่ยากแก่การป้องกันเป็นอย่างยิ่ง แม้ว่าเครื่องที่ถูกโจมตีจะทราบว่าเป็นแพ็กเก็ตปลอมแต่ก็ต้องใช้ระยะเวลาหนึ่งในการประมวลผลและกำจัดแพ็กเก็ตปลอมนั้นทิ้งไป ถ้ามีแพ็กเก็ตประเภทนี้ถูกส่งเข้ามาในจำนวนที่มากพอและอย่างต่อเนื่องแล้ว เครื่องที่ถูกโจมตีก็จะต้องใช้เวลาของ CPU ทั้งหมดในการจัดการแก้ปัญหานี้และไม่สามารถมีเวลาไปทำงานอย่างอื่นได้

8.6.3 Virtual Private Network



รูปที่ 8-30
(a) ระบบเครือข่ายส่วนตัวที่ใช้สายเช่า
(b) ระบบเครือข่ายส่วนตัวเสมือน

องค์กรจำนวนมากมีสำนักงานและโรงงานผลิตหลายแห่งที่กระจายตัวกันอยู่ในพื้นที่ต่าง ๆ ซึ่งอาจอยู่คนละเมืองหรือแม้แต่อยู่คนละประเทศ ในสมัยก่อนที่จะมีระบบเครือข่ายข้อมูลสาธารณะเกิดขึ้นองค์กรต่าง ๆ จะใช้วิธีการเช่าใช้สายสื่อสาร (leased line) จากบริษัทผู้ให้บริการโทรศัพท์จากสำนักงานแห่งหนึ่งไปยังสำนักงานอีกแห่งหนึ่ง ในปัจจุบันบางองค์กรก็ยังคงใช้วิธีนี้อยู่ ระบบเครือข่ายที่สร้างขึ้นระหว่างเครื่องคอมพิวเตอร์ขององค์กรเดียวกันและสายเช่าเหล่านี้ประกอบกันเป็น ระบบเครือข่ายส่วนตัว (Private network) รูป 8-30(a) แสดงตัวอย่างระบบเครือข่ายส่วนตัวที่เชื่อมต่อสถานที่สามแห่งเข้าด้วยกัน

ระบบเครือข่ายส่วนตัวสามารถใช้งานได้ดีและมีความปลอดภัยสูง ถ้าสายสื่อสารที่ใช้เป็นสายเช่าก็จะแน่ใจได้ว่าจะไม่มีข้อมูลรั่วไหลออกไปภายนอกองค์กร และผู้บุกรุกจำเป็นจะต้องสร้างสายเชื่อมต่อเข้ากับสายเช่าเพื่อที่จะได้สามารถเข้าสู่ระบบเครือข่ายได้ซึ่งเป็นวิธีการที่ทำได้ยาก ปัญหาของการใช้สายเช่าก็คือ การใช้สายเช่าอย่างเช่นสาย T1 นั้นมีค่าใช้จ่ายสูง และสายเช่า T3 ยิ่งมีราคาสูงขึ้นไปอีกมาก ต่อมาเมื่อระบบเครือข่ายข้อมูลสาธารณะและระบบอินเตอร์เน็ตได้รับการพัฒนาขึ้นมาใช้งาน องค์กรจำนวนมากต้องการเคลื่อนย้ายข้อมูลของตนเองผ่านระบบเครือข่ายสาธารณะแต่ก็ยังคงต้องการการรักษาความปลอดภัยให้อยู่ในระดับเดิม

ความต้องการดังกล่าวได้นำมาสู่การพัฒนาระบบเครือข่ายส่วนตัวเสมือน (Virtual Private Networks; VPN) ซึ่งหมายถึงระบบเครือข่ายที่ซ้อนทับอยู่บนระบบเครือข่ายสาธารณะซึ่งมีคุณสมบัติเช่นเดียวกันกับระบบเครือข่ายส่วนตัว ที่เรียกว่า "เสมือน" นั้นก็เนื่องมาจากเป็นระบบเครือข่ายที่สร้างขึ้นมาจากวงจรเสมือน (virtual circuit) ซึ่งไม่ใช่วงจรจริงและใช้หน่วยความจำเสมือนซึ่งไม่ใช่หน่วยความจำจริง

แม้ว่า VPN จะสามารถสร้างขึ้นมาบนระบบเครือข่าย ATM หรือ frame relay ได้แต่แนวโน้มหรือความนิยมในปัจจุบันคือการสร้าง VPN บนระบบเครือข่ายอินเตอร์เน็ต การออกแบบโดยทั่วไปนั้นจะติดตั้งแต่ละสำนักงานด้วยไฟร์วอลและสร้างอุโมงค์สื่อสาร (tunnel) ขึ้นระหว่างสำนักงานต่าง ๆ ดังแสดงในรูป 8-30(b) ถ้านำระบบ IPsec มาใช้เป็นตัวสร้างอุโมงค์สื่อสารก็จะสามารถรวบรวมปริมาณข้อมูลในอุโมงค์สื่อสารทั้งหมดที่มีการรับและส่งระหว่างสำนักงานต่าง ๆ เข้าเป็นระบบ SA ซึ่งมีการเข้ารหัสและตรวจสอบผู้ใช้เพียงระบบเดียว จึงช่วยในการตรวจสอบและควบคุมความถูกต้องของข้อมูล การรักษาความลับ และยังปลอดภัยจากการโจมตีด้วยวิธีการวิเคราะห์ปริมาณข้อมูล (traffic analysis attack)

เมื่อระบบถูกสร้างขึ้นมาใช้งาน ไฟร์วอลแต่ละคู่จะต้องต่อรองค่าพารามิเตอร์ที่จะนำมาใช้ในระบบ SA รวมทั้ง service mode, อัลกอริทึม, และคีย์ร่วม ระบบไฟร์วอลจำนวนมากมีขีดความสามารถในการสร้าง VPN แม้ว่าเราเตอร์ธรรมดาก็สามารถนำมาสร้าง VPN ได้เช่นกัน แต่เนื่องจากไฟร์วอลเป็นตัวหลักในระบบรักษาความปลอดภัยของหน่วยงานธุรกิจ การสร้างอุโมงค์สื่อสารจึงมักจะให้เริ่มต้นที่ไฟร์วอลตัวหนึ่งและไปสิ้นสุดที่ไฟร์วอลอีกตัวหนึ่งซึ่งจะเป็นวิธีการที่แยกระบบเครือข่ายสื่อสารระหว่างระบบเครือข่ายขององค์กรกับระบบอินเตอร์เน็ตออกจากกันโดยเด็ดขาด ดังนั้น ไฟร์วอล VPN และ IPsec ที่ทำงานใน ESP tunnel mode คือองค์ประกอบหลักของระบบ VPN ที่ได้รับการนำไปใช้งานอย่างกว้างขวาง

เมื่อระบบ SA ได้รับการจัดตั้งขึ้นแล้ว ก็จะสามารถเริ่มต้นการส่งข้อมูลได้ แพ็กเก็ตที่เดินทางไปตามอุโมงค์สื่อสาร VPN tunnel ก็คือแพ็กเก็ตธรรมดาที่เดินทางผ่านเราเตอร์ในระบบอินเตอร์เน็ต สิ่งเดียวที่แตกต่างไปจากแพ็กเก็ตทั่วไปก็คือ เป็นแพ็กเก็ตที่มี IPsec header อยู่ตามหลัง IP header แต่เนื่องจากส่วนที่เพิ่มเติมเข้ามานี้ไม่มีผลต่อกระบวนการจัดส่งแพ็กเก็ต ดังนั้น เราเตอร์จึงไม่ได้ให้ความสนใจกับ header ส่วนที่เพิ่มเข้ามาด้วยนี้

ข้อเด่นของการจัดโครงสร้าง VPN แบบนี้ก็คือเป็นระบบที่โปร่งใสจากการทำงานของโปรแกรมประยุกต์โดยสิ้นเชิง ไฟร์วอลจะทำหน้าที่จัดตั้งและบริหาร SA บุคคลผู้เดียวที่จะต้องเข้ามาเกี่ยวพันในที่นี้คือผู้บริหารระบบเครือข่ายซึ่งเป็นผู้ที่จะต้องปรับค่าพารามิเตอร์ให้แก่ไฟร์วอล สำหรับผู้อื่นแล้วก็คล้ายกับว่ากลับไปใช้ระบบเครือข่ายส่วนตัวแทนที่จะเป็นระบบอินเตอร์เน็ต

8.6.4 การรักษาความปลอดภัยในระบบสื่อสารไร้สาย

เป็นที่น่าแปลกใจว่าการออกแบบระบบที่มีความปลอดภัยสูงโดยการใช้ VPN และไฟร์วอลนั้นสามารถทำได้โดยง่ายแต่ข้อเท็จจริงก็คือการรั่วไหลนั้นมีอยู่เต็มไปหมด สถานการณ์เช่นนี้อาจเกิดขึ้นได้ถ้าเครื่องจำนวนหนึ่งทำการสื่อสารโดยใช้ระบบไร้สายโดยใช้คลื่นวิทยุในการสื่อสารระหว่างกัน ซึ่งสามารถเดินทางข้ามไฟร์วอลไปได้ทั้งสองทิศทาง คือทั้งขาเข้าและขาออก ขอบเขตการสื่อสารตามมาตรฐานระบบเครือข่าย 802.11 นั้นอยู่ที่หลายร้อยเมตร ดังนั้น ใครก็ตามที่ต้องการจะเข้ามาล้วงความลับขององค์กรก็เพียงแค่ขับรถเข้ามาจอดไว้ที่ลานจอดรถขององค์กรในตอนเช้า และเปิดเครื่องโน้ตบุ๊คที่มีช่องสื่อสาร 802.11 เอาไว้ในรถยนต์แล้วทำการบันทึกข้อมูลทั้งหมดที่สามารถรับได้ จากนั้นก็ขับรถออกไปในตอนเย็น ในช่วงบ่ายของวันนั้นฮาร์ดดิสก์ของเครื่องโน้ตบุ๊คก็จะเต็มไปด้วยข่าวสารที่มีค่าจำนวนมาก โดยทาง ทฤษฎีแล้วการรั่วไหลของข่าวสารในทางนี้ไม่น่าที่จะเกิดขึ้น เช่นเดียวกันในทางทฤษฎีแล้วก็ไม่ควรที่จะมีการปล้นธนาคารเกิดขึ้น (แต่มันก็เกิดขึ้นจนได้)

ปัญหาเกี่ยวกับการรักษาความปลอดภัยสามารถมองย้อนกลับไปที่ผู้ผลิตอุปกรณ์สื่อสารไร้สายที่ต้องการผลิตอุปกรณ์ที่ง่ายต่อการใช้งาน โดยปกติผู้ใช้เพียงแค่นำอุปกรณ์ออกจากกล่อง เสียบปลั๊กอุปกรณ์นั้นก็จะสามารถทำงานได้ในทันทีซึ่งแทบจะไม่มีเรื่องการรักษาความปลอดภัยเข้ามาเกี่ยวข้องด้วยเลย นั่นคือ ปลดปล่อยความลับไปให้กับทุกคนที่อยู่ภายในขอบเขตการกระจายคลื่นของอุปกรณ์นั้น ถ้านำอุปกรณ์นี้มาเสียบต่อเข้ากับ Ethernet ข้อมูลทั้งหมดที่ไหลเวียนอยู่ในระบบ Ethernet ก็จะถูกส่งออกไปที่ลานจอดรถ (ที่มีผู้ดักฟังสัญญาณอยู่) ด้วยเช่นกัน ระบบสื่อสารไร้สายจึงเป็นเสมือนความฝันที่กลายเป็นความจริง นั่นคือ ได้ข้อมูลมาฟรีโดยไม่ต้องออกแรงเลย ดังนั้นการรักษาความปลอดภัยใน

ระบบสื่อสารไร้สายจึงเป็นเรื่องที่สำคัญมากกว่าการรักษาความปลอดภัยให้กับระบบเครือข่ายแบบที่ใช้สายสื่อสาร

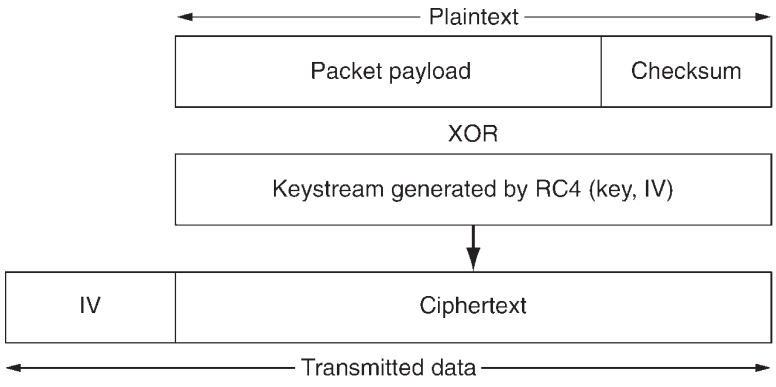
ความปลอดภัยบนระบบ 802.11

มาตรฐาน 802.11 ได้อธิบายถึงการรักษาความปลอดภัยในระดับชั้นสื่อสารเชื่อมต่อข้อมูลไว้ เรียกว่า WEP (Wired Equivalent Privacy) ซึ่งได้รับการออกแบบมาเพื่อสร้างความปลอดภัยให้แก่ระบบเครือข่าย LAN แบบไร้สายให้ดีเท่าๆ กับระบบ LAN แบบใช้สายสื่อสาร เนื่องจากระบบ LAN แบบใช้สายสื่อสารนั้นไม่มีการรักษาความปลอดภัยอยู่ด้วย ดังนั้นเป้าหมายของความปลอดภัยบนระบบ LAN ไร้สายจึงสามารถทำให้สำเร็จได้ง่าย

เมื่อมีการกำหนดใช้การรักษาความปลอดภัยบนระบบสื่อสารไร้สาย 802.11 แต่ละสถานี (station) จะมีคีย์ลับที่ใช้งานร่วมกับสถานีฐาน (base station) แต่มาตรฐานไม่ได้กำหนดวิธีการกระจายคีย์ลับเอาไว้ซึ่งอาจจะทำได้โดยการใส่เข้าไปในตัวอุปกรณ์โดยผู้ผลิตอุปกรณ์นั้นๆ คีย์ลับอาจมีการแลกเปลี่ยนกันล่วงหน้าผ่านระบบเครือข่ายแบบใช้สาย ประการสุดท้าย สถานีฐานหรือสถานีผู้ใช้อาจใช้วิธีการเลือกคีย์แบบสุ่มเลือกขึ้นมาและส่งไปให้อีกฝ่ายหนึ่งผ่านการเข้ารหัสโดยใช้คีย์สาธารณะของอีกฝ่ายหนึ่งก็ได้ เมื่อสามารถกำหนดคีย์ลับได้แล้ว คีย์นี้มักจะถูกนำไปใช้งานเป็นระยะเวลานานหลายเดือนหรือหลายปีเลยทีเดียว

การเข้ารหัส WEP จะใช้ stream cipher ที่ทำงานร่วมกับอัลกอริทึมเข้ารหัสแบบ RC4 ซึ่งได้รับการพัฒนาขึ้นมาโดย Ronald Rivest และได้รับการรักษาความลับไว้จนกระทั่งเกิดการรั่วไหลออกมาและนำไปประกาศไว้บนระบบอินเตอร์เน็ตในปี ค.ศ. 1994 ใน WEP วิธี RC4 ถูกนำมาสร้าง keystream แล้วนำไป exclusive-OR กับ plaintext เพื่อสร้างเป็น ciphertext ออกมา

ข้อมูลจริงในแต่ละแพ็กเก็ตจะถูกเข้ารหัสโดยใช้วิธีการที่แสดงในรูป 8-31 แรกทีเดียว ข้อมูลจริงจะถูกสร้างข้อมูลสำหรับการตรวจสอบ checksum โดยใช้ CRC-32 polynomial และข้อมูลดังกล่าวจะถูกใส่ต่อท้ายเข้าไปกับข้อมูลจริงเพื่อสร้างเป็น plaintext สำหรับอัลกอริทึมเข้ารหัสข้อมูล จากนั้น plaintext จะถูกนำมาทำ exclusive-OR กับส่วนของ keystream ที่มีขนาดเท่ากัน ผลที่ได้คือ ciphertext ข้อมูล IV (initialization vector) ที่ใช้ในการเริ่มต้นการทำงานของ RC4 จะถูกส่งมาพร้อมกับ ciphertext เมื่อผู้รับได้รับแพ็กเก็ตนี้แล้ว ก็จะดึงข้อมูลจริงที่ถูกเข้ารหัสไว้ออกมา ทำการสร้าง keystream จากคีย์ลับที่ใช้งานร่วมกันพร้อมกับ IV ที่ถูกส่งมาด้วยกันและทำการ exclusive-OR keystream เข้ากับข้อมูล



รูปที่ 8-31
การเข้ารหัสข้อมูลด้วยวิธี WEP

ที่ถูกเข้ารหัสเพื่อให้ได้เป็น plaintext ออกมา จากนั้นจึงทำการตรวจสอบ checksum เพื่อดูว่ามีความเสียหายเกิดขึ้นกับข้อมูลจริงหรือไม่

แนวทางนี้ดูเหมือนว่าจะเป็นแนวทางที่ดีแต่วิธีการถอดรหัสโดยไม่ต้องใช้คีย์ลับได้ถูกตีพิพม์เผยแพร่ในปี ค.ศ. 2001 แล้ว ซึ่งมีแนวทางดังนี้ ประการแรกอุปกรณ์จำนวนมากใช้คีย์ลับตัวเดียวกันสำหรับผู้ใช้ทุกคน ในกรณีนี้ทำให้ผู้ใช้คนหนึ่งสามารถอ่านข่าวสารของผู้ใช้คนอื่นได้ซึ่งเป็นสิ่งที่เกิดขึ้นในระบบ Ethernet จึงเป็นวิธีการที่ไม่ปลอดภัยนัก

สมมุติว่าผู้ใช้แต่ละคนใช้คีย์ลับคนละตัวกัน WEP ก็ยังคงถูกโจมตีได้ เนื่องจากคีย์จะถูกนำมาใช้เป็นระยะเวลานานมาก มาตรฐาน WEP จึงได้แนะนำ (แต่ไม่ได้บังคับ) ให้เปลี่ยนค่า IV สำหรับทุกแพ็กเก็ตเพื่อป้องกันปัญหา keystream reuse attack ดังที่ได้อธิบายในหัวข้อ 8.2.3 อย่างไรก็ตาม อุปกรณ์ 802.11 จำนวนมากสำหรับเครื่องโน้ตบุ๊คจะเปลี่ยนค่าให้ IV มีค่าเป็น 0 เมื่อมีการเสียบอุปกรณ์เข้าไปในเครื่องฯ และเพิ่มค่าขึ้นทีละ 1 ในทุกแพ็กเก็ตที่ถูกส่งออกมา เนื่องจากผู้ใช้มักจะถอดและเสียบอุปกรณ์นี้เข้ากับเครื่องฯ อยู่เสมอทำให้มีแพ็กเก็ตเป็นจำนวนมากที่มีค่า IV เป็นเลขจำนวนต่ำ ถ้าผู้บุกรุกสามารถรวบรวมแพ็กเก็ตที่ส่งจากผู้ใช้คนหนึ่งได้เป็นจำนวนมากพอ (ที่มีค่า IV เท่ากัน) เขาก็จะสามารถคำนวณค่า exclusive-OR ของ plaintext 2 ค่าและอาจจะสามารถถอดรหัส ciphertext ได้ในที่สุด

แม้ว่าจะให้อุปกรณ์ 802.11 เลือกใช้ค่า IV มาอย่างสุ่มสำหรับทุกแพ็กเก็ต ค่าของ IV เป็นตัวเลขขนาด 24 บิต ดังนั้น เมื่อทำการส่งแพ็กเก็ตออกมา 224 แพ็กเก็ตแล้วตัวเลขชุดนี้ก็จะต้องถูกนำกลับมาใช้ใหม่ ยิ่งกว่านั้น การเลือกค่า IV อย่างสุ่มทำให้ค่าคาดหวังของจำนวนแพ็กเก็ตที่ถูกส่งออกไปก่อนที่จะใช้หมายเลขเดิมสองครั้งมีค่าประมาณ 5000 เนื่องจากคุณลักษณะของ birthday attack ดังที่อธิบายไว้ในหัวข้อ 8.4.4 ดังนั้น ถ้าผู้บุกรุกดักฟังข่าวสารเป็นเวลาหลายนาทีก็เป็นที่แน่นอนว่าผู้บุกรุกจะสามารถดักจับแพ็กเก็ตอย่างน้อย 2 แพ็กเก็ตที่มีค่า IV และคีย์ตัวเดียวกัน จากการนำ ciphertext ไปทำ exclusive-OR ผู้บุกรุกก็จะได้ค่า exclusive-OR ของ plaintext ลำดับของบิตนี้อาจถูกแก้ไขได้หลายวิธีเพื่อทำให้ได้ plaintext ออกมา และถ้าทำต่อไปอีกก็จะสามารถค้นหา keystream สำหรับ IV นั้นออกมาได้ เมื่อผู้บุกรุกยังคงทำงานในลักษณะนี้อีกต่อไปก็จะสามารถค้นหา keystream สำหรับค่า IV ต่างๆ ออกมาได้ เมื่อสามารถค้นหาค่า IV ได้แล้วแพ็กเก็ตที่ถูกส่งออกมาในอนาคตด้วย ค่า IV ที่ค้นพบนี้ก็จะสามารถถูกถอดรหัสได้ทั้งหมด

ยิ่งกว่านั้น เนื่องจาก IV ถูกนำมาใช้อย่างสุ่ม เมื่อผู้บุกรุกสามารถค้นหาค่า IV และ keystream ได้เขาก็จะสามารถนำมาสร้างแพ็กเก็ตที่ต้องการเพื่อส่งเข้าไปรบกวนการสื่อสารได้ ในทางทฤษฎี ผู้รับข้อมูลจะสามารถสังเกตได้ว่ามีปริมาณแพ็กเก็ตจำนวนมากที่มีค่า IV เดียวกัน แต่ระบบ WEP เองก็อาจทำให้เกิดเหตุการณ์เช่นนี้ได้และไม่มีผู้ใดสนใจที่จะตรวจสอบอยู่ดี

ประการสุดท้าย การใช้ CRC นั้นไม่มีผลมากนักเนื่องจากมีความเป็นไปได้ที่ผู้บุกรุกจะทำการเปลี่ยนข้อมูลจริงและแก้ไขค่า CRC ให้ถูกต้องได้โดยไม่มีความจำเป็นที่จะต้องถอดรหัสข้อความออกมาก่อนเลย โดยภาพรวมแล้วการถอดรหัสข้อความนั้นเป็นกระบวนการที่ทำได้อย่างตรงไปตรงมาซึ่งยังมีจุดบกพร่องอยู่อีกมากที่ไม่ได้กล่าวถึง

การรักษาความปลอดภัยบน Bluetooth

ระบบ Bluetooth มีขอบเขตการแพร่กระจายคลื่นสั้นกว่ามาตรฐาน 802.11 จึงไม่สามารถที่จะถูกขโมยข้อมูลได้ง่านนัก เช่น การจอดรถดักฟังสัญญาณอยู่ในลานจอดรถแต่เรื่องการรักษาความปลอดภัยก็ยังคงเป็นประเด็นที่จะต้องกล่าวถึง ตัวอย่างเช่น เครื่องคอมพิวเตอร์ของอลิสติดตั้งแป้นพิมพ์ไร้สาย (wireless keyboard) ที่รับส่งข้อมูลแบบ bluetooth เอาไว้ ถ้าไม่มีการรักษาความปลอดภัยเลย ผู้บุกรุกที่นั่งอยู่ในห้องทำงานติดกันจะสามารถดักจับสัญญาณซึ่งเป็นตัวพิพม์บนแป้นพิมพ์ที่อลิสใช้ได้ทั้งหมดซึ่งรวมถึงอีเมลล์ที่อลิสส่งไปยังผู้อื่น ผู้บุกรุกยังสามารถดักจับสัญญาณข้อมูลทั้งหมดที่ Bluetooth printer ทำการพิมพ์ออกมา แต่ยังโชคดีที่ระบบ Bluetooth มีกลไกในการรักษาความปลอดภัยอยู่ในตัวเองดังนี้

Bluetooth แบ่งการรักษาความปลอดภัยออกเป็น 3 mode ตั้งแต่ระดับที่ไม่มีการรักษาความปลอดภัยเลยไปจนถึงระดับที่มีการเข้ารหัสข้อมูลและตรวจความถูกต้องของข้อมูล โดยปกติผู้ใช้มักจะเลือกที่จะไม่ใช้ระบบรักษาความปลอดภัยจนกว่าจะเกิดเรื่องร้ายขึ้นจึงเริ่มที่จะหันมาใช้ระบบรักษาความปลอดภัย หรือเรียกว่า วัวหายแล้วจึงล้อมคอก

Bluetooth มีการรักษาความปลอดภัยในหลายระดับชั้นสื่อสาร ในชั้นสื่อสารกายภาพจะเลือกใช้เทคนิคการเปลี่ยนความถี่สัญญาณแบบ frequency hopping ซึ่งช่วยรักษาความปลอดภัยได้ในระดับหนึ่ง แต่เนื่องจากอุปกรณ์ bluetooth ได้หันไปใช้ piconet จึงจำเป็นต้องบอกลำดับ frequency hopping sequence ล่วงหน้า ซึ่งลำดับฯ นี้ก็ไม่เป็นความลับอย่างแน่นอน ความปลอดภัยเริ่มเกิดขึ้นเมื่ออุปกรณ์ slave ร้องขอช่องสื่อสารมายัง master อุปกรณ์ทั้งสองจะใช้คีย์ลับร่วมกันซึ่งได้รับการกำหนดใช้งานล่วงหน้า ในบางกรณี อุปกรณ์ทั้งสองจะถูกกำหนดคีย์ลับมาจากโรงงานผู้ผลิต ในบางกรณี อุปกรณ์ตัวหนึ่งจะได้รับการกำหนดคีย์ลับมาจากโรงงานและผู้ใช้จะต้องป้อนคีย์ลับให้แก่อุปกรณ์อีกตัวหนึ่ง คีย์ที่ใช้ร่วมกันนี้เรียกว่า passkeys

ในการจัดตั้งช่องสื่อสารอุปกรณ์ทั้ง slave และ master จะตรวจสอบดูว่าอุปกรณ์อีกตัวหนึ่งนั้นใช้ passkey ตัวเดียวกัน จากนั้นจึงเริ่มเจรจาว่าจะใช้ช่องสื่อสารที่เข้ารหัส และมีการตรวจสอบความถูกต้องของข้อมูลหรือไม่ อย่างไร จากนั้นจึงเลือกหมายเลขมาแบบสุ่มเป็นเลขขนาด 128 บิต (session key)

การเข้ารหัสจะใช้ stream cipher เรียกว่า E0: ส่วนการตรวจสอบความถูกต้องของข้อมูลจะใช้ SAFER+ ทั้งสองวิธีคือวิธีการเข้ารหัสข้อมูลแบบคีย์สมมาตร การเข้ารหัสโดยใช้ stream cipher แสดงในรูป 8-14 เมื่อนำ plaintext มาทำการ exclusive-OR เข้ากับ keystream ก็จะได้ ciphertext อย่างไรก็ตาม E0 เองก็อาจมีจุดอ่อนอยู่ในตัวเองซึ่งแม้ว่าในขณะนี้จะยังไม่มีวิธีการถอดรหัสแบบนี้โดยไม่ใช้คีย์ลับก็ตาม โดยทั่วไปการแข่งขันระหว่างผู้เข้ารหัสกับผู้ที่พยายามถอดรหัสมักจะเกิดขึ้นอยู่ตลอดเวลาโดยที่ผู้ที่พยายามถอดรหัสมักจะเป็นฝ่ายชนะเสมอ

ปัญหาในเรื่องการรักษาความปลอดภัยอีกประการหนึ่งก็คือ ระบบ bluetooth ทำการตรวจสอบ

สิทธิการใช้งานของอุปกรณ์ไม่ใช้ผู้ที่ใช้อุปกรณ์นั้น ดังนั้น ผู้ที่สามารถขโมยอุปกรณ์ bluetooth ไปได้ก็จะสามารถขโมยข้อมูลอันเป็นความลับไปได้ด้วย อย่างไรก็ตาม bluetooth ก็มีการสร้างระบบรักษาความปลอดภัยในชั้นสื่อสารระดับสูง ดังนั้น แม้ว่าอุปกรณ์อาจจะถูกขโมยไปได้แต่การรักษาความปลอดภัยโดยเฉพาะกับโปรแกรมประยุกต์ก็ยังมีอยู่ คือจะต้องให้ผู้ใช้ระบุหมายเลข PIN code จึงจะสามารถทำให้โปรแกรมนั้นทำงานได้ตามปกติ เป็นต้น

การรักษาความปลอดภัยบนระบบ WAP 2.0

คณะกรรมการ WAP forum ได้เรียนรู้ข้อผิดพลาดมาจากการที่ไม่มีมาตรฐานกำหนดไว้ใน WAP 1.0 ดังนั้น WAP 2.0 จึงใช้โพรโตคอลมาตรฐานในทุกชั้นสื่อสารซึ่งรวมถึงการรักษาความปลอดภัยด้วย เนื่องจากเป็นระบบที่ใช้กับ IP ดังนั้นจึงสนับสนุนการใช้มาตรฐาน IPsec ในชั้นสื่อสารควบคุมเครือข่าย การเชื่อมต่อ TCP ในชั้นสื่อสารนำส่งข้อมูลก็สามารถได้รับการป้องกันโดย TSL และมาตรฐานของ IEFT ซึ่งจะได้กล่าวถึงในตอนท้ายของบทนี้ในชั้นสื่อสารระดับสูงขึ้นมาก็ใช้วิธีการตรวจสอบผู้ใช้ผ่านระบบ HTTP client authentication ดังที่อธิบายไว้ในมาตรฐาน RFC 2617 ในชั้นสื่อสารโปรแกรมประยุกต์ก็ใช้การรักษาความปลอดภัยแบบ crypto libraries โดยภาพรวมแล้ว WAP 2.0 นั้นทำงานโดยใช้มาตรฐานที่เป็นที่รู้จักกันโดยทั่วไปจึงมีความเป็นส่วนตัว มีการตรวจสอบผู้ใช้ มีการตรวจสอบความถูกต้องของข้อมูล และมีการปฏิเสธการใช้งานที่ดีกว่าระบบ 802.11 และ bluetooth

8.7 โพรโตคอลสำหรับการตรวจสอบผู้ใช้

การตรวจสอบผู้ใช้ (authentication) เป็นกระบวนการตรวจสอบ (verify) ผู้ที่กำลังสื่อสารด้วยนั้นว่าเป็นบุคคลรู้จักหรือเป็นบุคคลที่คาดว่าจะเป็นหรือว่าเป็นตัวปลอมกันแน่ การตรวจสอบความเป็นตัวตน (identity) ของโปรเซสที่อยู่ภายนอกระบบว่าเป็นโปรเซสที่เข้ามาเพื่อทำลายล้างหรือเป็นผู้บุกรุกที่ต้องการเข้ามาขโมยข้อมูลเป็นเรื่องที่ทำได้ยากมากและจำเป็นต้องใช้โพรโตคอลที่มีความสลับซับซ้อนที่ใช้การเข้ารหัสข้อมูล ในหัวข้อนี้จะได้กล่าวถึงโพรโตคอลจำนวนหนึ่งที่ใช้ในการตรวจสอบผู้ใช้ที่นำมาใช้ในระบบเครือข่ายที่ไม่มีความปลอดภัย

คนส่วนหนึ่งมีความเข้าใจสับสนระหว่างการให้อำนาจ (authorization) กับการตรวจสอบผู้ใช้ (authentication) การตรวจสอบผู้ใช้เกี่ยวข้องกับคำถามว่าท่านคือบุคคลตัวจริงที่กำลังสื่อสารอยู่กับโปรเซสหนึ่งหรือไม่ ส่วนการให้อำนาจนั้นเกี่ยวข้องกับการอนุญาตให้โปรเซสทำงานอะไรได้บ้าง ตัวอย่างเช่น โปรเซสของผู้ใช้ทำการติดต่อกับไฟล์เซิบเวอร์และพูดว่า "ฉันคือโปรเซสของสก็อตและต้องการที่จะลบแฟ้มข้อมูลที่ชื่อ cookbook.old" ในแง่มุมของไฟล์เซิบเวอร์ มีคำถามสองคำถามที่จะต้องได้รับคำตอบก่อนที่จะทำงานต่อไป นั่นคือ

1. นี่คือโปรเซสของสก็อตใช่หรือไม่ (การตรวจสอบผู้ใช้)

2. สก็อตได้รับอนุญาตให้ลบแฟ้มข้อมูล "cookbook.old" หรือไม่ (การตรวจสอบอำนาจของผู้ใช้)

เมื่อได้รับคำตอบอย่างชัดเจนและสอดคล้องกับข้อมูลที่ระบบมีอยู่สำหรับคำถามทั้งสองข้อนี้แล้ว ระบบจึงจะทำตามที่คำขอนี้ต้องการ คำถามแรกเป็นคำถามที่มีความสำคัญเป็นอย่างมาก เมื่อไฟล์เซิบเวอร์ทราบว่ากำลังพูดอยู่กับใครแล้ว การตรวจสอบอำนาจผู้ใช้ก็เป็นเพียงการดูข้อมูลในตารางเท่านั้น ในหัวข้อนี้จึงมุ่งความสนใจไปที่การตรวจสอบผู้ใช้

รูปแบบทั่วไปของการตรวจสอบผู้ใช้เป็นดังนี้ อลิสเริ่มต้นด้วยการส่งข่าวสารไปยังบ๊อบหรือ KDC (Key Distribution Center) ซึ่งจะต้องเป็นบุคคลหรือองค์กรที่มีความซื่อสัตย์ อาจมีข่าวสารอีกหลายอย่างที่มีการส่งแลกเปลี่ยนกัน ข่าวสารที่ส่งไปนี้อาจถูกผู้บุกรุกคือทรูดี้ดักจับได้ ทำการแก้ไข และส่งต่อไปเพื่อหลอกลวงอลิสและบ๊อบหรือเพียงแค่ขัดขวางการทำงานของคนทั้งสองเท่านั้น

อย่างไรก็ตาม เมื่อโพรโตคอลทำงานเสร็จ อลิสจะแน่ใจว่ากำลังคุยอยู่กับบ๊อบและบ๊อบก็แน่ใจว่ากำลังคุยอยู่กับอลิส ยิ่งกว่านั้น ในโพรโตคอลส่วนใหญ่ทั้งสองคนนี้จะต้องร่วมกันสร้าง session key ที่เป็นความลับสำหรับนำมาใช้ในการสื่อสารที่จะตามมา ในทางปฏิบัติ เพื่อเหตุผลทางด้านประสิทธิภาพการทำงาน ข้อมูลทั้งหมดจะถูกนำมาเข้ารหัสด้วยวิธีการแบบคีย์สมมาตร (อาจจะเป็น AES หรือ triple DES) แม้ว่าการเข้ารหัสแบบใช้คีย์สาธารณะจะถูกนำมาใช้อย่างกว้างขวางสำหรับโพรโตคอลการตรวจสอบผู้ใช้และในการสร้าง session key ขึ้นมา

ประเด็นของการใช้ session key ตัวใหม่ที่ถูกเลือกขึ้นมาแบบสุ่มสำหรับการเชื่อมต่อที่พึ่งจะเกิดขึ้นนั้นก็เพื่อลดปริมาณข้อมูลที่จะถูกส่งไปพร้อมกับคีย์ลับของผู้ใช้หรือคีย์สาธารณะ เพื่อลดปริมาณ ciphertext ที่ผู้บุกรุกอาจได้ไป และเพื่อลดความเสียหายที่อาจเกิดขึ้นถ้ากระบวนการนี้เกิดล้มเหลวลงกลางคัน คีย์ตัวเดียวที่เหลืออยู่ควรจะเป็น session key คีย์ที่เหลือทั้งหมดควรถูกขจัดออกไปภายหลังจากที่การสื่อสาร (session) ได้เริ่มต้นขึ้น

8.7.1 การตรวจสอบผู้ใช้ด้วยการใช้คีย์ลับร่วมกัน

สำหรับโพรโตคอลแรกที่นำมาใช้ในการตรวจสอบผู้ใช้ สมมุติว่าอลิสและบ๊อบนั้นใช้คีย์ลับร่วมกัน คือ KAB คีย์ร่วมนี้อาจจะได้รับการตกลงร่วมกันทางโทรศัพท์หรือติดต่อกันเป็นการส่วนตัว หรือผ่านระบบเครือข่ายที่ปลอดภัย

โพรโตคอลนี้มีพื้นฐานเดียวกันกับที่ใช้ในโพรโตคอลอื่นๆ คือ ผู้ส่งจะส่งหมายเลขสุ่มไปยังผู้รับซึ่งจะทำการเปลี่ยนแปลงรูปแบบในลักษณะพิเศษแล้วส่งผลลัพธ์นั้นกลับมา โพรโตคอลนี้เรียกว่า challenge-response protocol สัญลักษณ์ต่างๆ ที่ใช้จะเป็นดังนี้

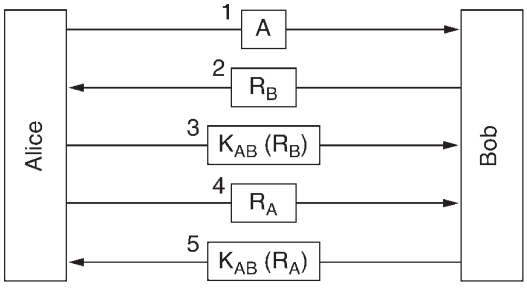
A, B คือตัวตนของอลิสและบ๊อบ

Ri คือ challenge โดยที่ตัวห้อยคือผู้ที่ส่งข้อความนี้

Ki คือคีย์ โดยที่ตัวห้อยคือเจ้าของคีย์

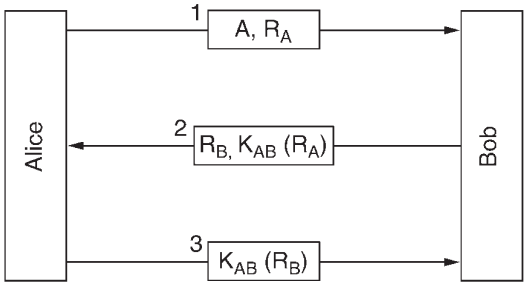
Ks คือ session key

รูป 8-32 แสดงลำดับการส่งข่าวสารที่เกิดขึ้นในโพรโตคอล shared-key authentication protocol



รูปที่ 8-32 Two-way authentication โดยการใช้ challenge-response protocol

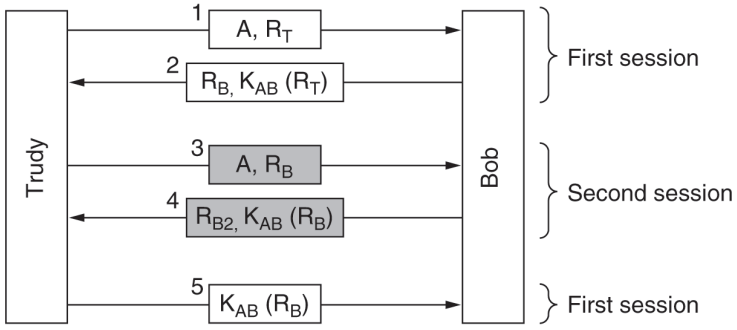
รูปที่ 8-33
Challenge-response protocol ฉบับย่อ



ในข่าวสารที่ 1 อลิสส่งตัวตนของเธอคือ A ไปยังบ๊อบในรูปแบบที่บ๊อบสามารถเข้าใจได้ บ๊อบซึ่งไม่มีทางทราบว่าข่าวสารนี้ถูกส่งมาจากอลิสจริงหรือผู้อื่น (ทรูดี้) เขาจึงเลือกที่จะท้าทาย (challenge) ด้วยการส่งหมายเลขสุ่มขนาดใหญ่ RB กลับไปยังอลิสในรูปแบบของ plaintext หมายเลขนี้คือส่วนหนึ่งของ challenge-response protocol ซึ่งเรียกว่า nonces อลิสจะทำการเข้ารหัสหมายเลขนี้ด้วยคีย์ที่ใช้ร่วมกับบ๊อบและจัดการส่ง ciphertext KAB(RB) เป็นข่าวสารชิ้นที่ 3 เมื่อบ๊อบได้รับข่าวสารชิ้นนี้แล้ว เขาก็จะทราบในทันทีว่าข่าวสารนี้มาจากอลิสเนื่องจากทรูดี้ไม่ทราบ KAB จึงไม่มีทางที่จะสร้างข่าวสารชิ้นที่ 3 นี้ขึ้นมาได้อย่างถูกต้อง ยิ่งกว่านั้น เนื่องจาก RB ซึ่งเป็นตัวเลขขนาดใหญ่ (128 บิต) ถูกเลือกขึ้นมาอย่างสุ่มจึงมีความเป็นไปได้น้อยมากที่ทรูดี้จะเคยเห็นเลขนี้มาก่อนและทราบว่าจะต้องตอบกลับมาด้วยตัวเลขใดจากการสื่อสารในครั้งก่อนหน้านี้ และยังมีความเป็นไปได้น้อยมากที่ทรูดี้จะสามารถเดาตัวเลขตอบรับได้อย่างถูกต้อง

ณ จุดนี้ บ๊อบสามารถแน่ใจได้ว่าเขากำลังติดต่ออยู่กับอลิส แต่อลิสยังไม่แน่ใจว่ากำลังติดต่ออยู่กับใคร ดังนั้นเธอจึงเลือกตัวเลขมาแบบสุ่มหนึ่งจำนวนคือ RA และส่งไปยังบ๊อบในรูปแบบของ plaintext ในข่าวสารชิ้นที่ 4 เมื่อบ๊อบตอบกลับมาด้วย KAB(RA) อลิสก็จะทราบว่ากำลังติดต่ออยู่กับบ๊อบ ตอนนี้ทั้งสองคนสามารถกำหนด session key ขึ้นมาใช้งานได้แล้วโดยที่อลิสจะเลือก session key Ks แล้วส่งไปให้บ๊อบในรูปแบบ KAB(KS)

โพรโตคอลในรูป 8-32 ประกอบด้วยข่าวสารจำนวน 5 ข่าวสาร ต่อไปลองมาดูว่าจะสามารถลดจำนวนข่าวสารลงไปได้มากน้อยเพียงใด แนวทางหนึ่งได้แสดงให้เห็นในรูป 8-33 ในที่นี้อลิสจะเริ่มต้นกระบวนการ challenge-response เองแทนที่จะรอให้บ๊อบเป็นฝ่ายเริ่ม ในทำนองเดียวกันเมื่อบ๊อบตอบ challenge ของอลิสเขาก็จะส่ง challenge ของตนเองมาพร้อมกันเลย โพรโตคอลทั้งหมดจึงจบลง



รูปที่ 8-34
The reflection attack

ด้วยการสื่อสารเพียง 3 ข่าวสารเท่านั้น

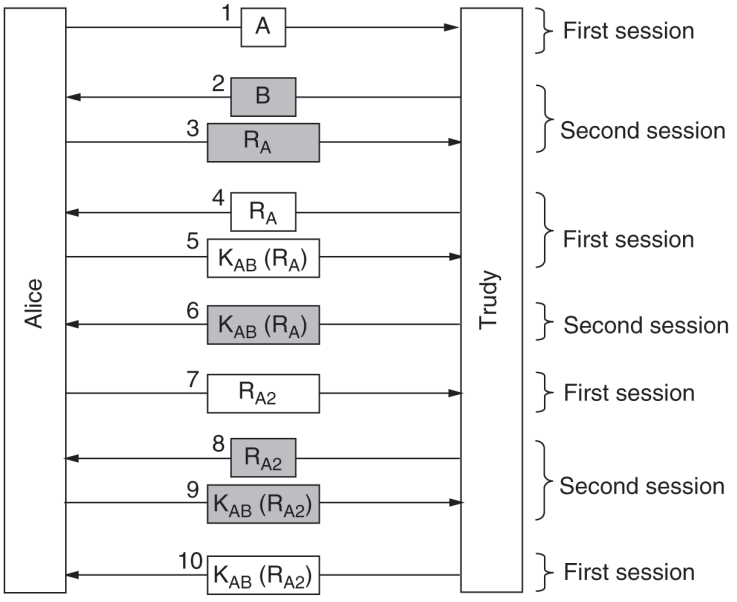
แม้ว่า challenge-response ฉบับย่อจะดูเหมือนว่าเป็นการปรับปรุงโพรโตคอลเดิมให้กระทัดรัดมากยิ่งขึ้น แต่ก็เป็นวิธีการที่ผิด ภายใต้สถานะการณ์หนึ่งที่อาจเกิดขึ้น ทรูดี้สามารถแก้ไขโพรโตคอลนี้ได้โดยใช้วิธีการเรียกว่า reflection attack กล่าวคือ ทรูดี้จะแก้ไขโพรโตคอลนี้ได้ถ้าเขาสามารถเปิดช่องการสื่อสารขึ้นได้หลายช่องทางไปยังบ็อบได้พร้อมกัน ซึ่งก็อาจเป็นไปได้ ตัวอย่างเช่น บ็อบหมายถึงคอมพิวเตอร์ของทางธนาคารที่ต้องติดต่อกับเครื่อง ATM จำนวนมากในเวลาเดียวกัน

รูป 8-34 แสดงกระบวนการ reflection attack ของทรูดี้ เริ่มต้นด้วยการที่ทรูดี้แอบอ้างตัวว่าเป็นอลิสและจัดการส่ง RT ไปยังบ็อบ บ็อบก็จะตอบสนองตามปกติและส่ง challenge RB ของเขามาด้วย ตอนนี้ทรูดี้จะมีปัญหาเพราะไม่สามารถตอบสนอง challenge ของบ็อบได้ (ไม่รู้วิธีการสร้าง KAB(RB))

ทรูดี้จึงเปิดการสื่อสารช่องใหม่ขึ้นมาด้วยการส่งข่าวสารชิ้นที่ 3 ไปยังบ็อบและจัดการส่ง RB ที่นำมาจากข่าวสารชิ้นที่ 2 ไปด้วยในฐานะที่เป็น challenge ของเธอเอง บ็อบจึงตอบกลับมาด้วย KAB(RB) ในข่าวสารชิ้นที่ 4 โดยที่ไม่ทราบว่าถูกทรูดี้เล่นกลเสียแล้วในขณะนี้ทรูดี้มีข้อมูลที่จะสามารถตอบสนองต่อ challenge ของบ็อบ (ใน session แรก) ได้แล้ว บ็อบก็จะถูกทำให้เชื่อว่าทรูดี้คืออลิส ทรูดี้ก็จะสามารถได้รับข้อมูลทุกอย่างของอลิสตามที่เข้าต้องการ

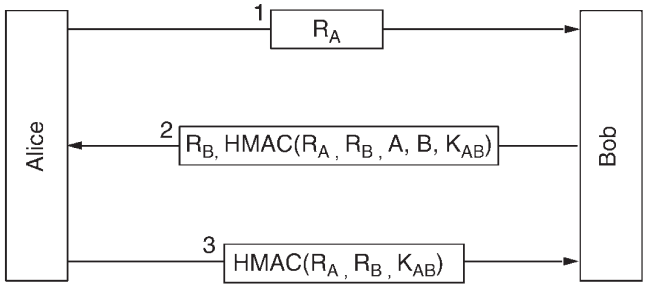
กฎเกณฑ์ทั่วไปที่ควรนำมาใช้ 4 ข้อเป็นดังนี้

1. ให้ผู้ที่ติดต่อเข้ามาทำการพิสูจน์ตนเองก่อนเสมอ ในกรณีนี้บ็อบได้ส่งข่าวสารที่มีค่ายิ่งออกไปก่อนที่ทรูดี้จะให้ข่าวสารว่าเธอเป็นใครกันแน่
2. ให้ผู้ที่ติดต่อเข้ามาและผู้ที่รับการติดต่อใช้คีย์ลับคนละตัวกัน
3. ให้ผู้ที่ติดต่อเข้ามาและผู้ที่รับการติดต่อใช้ challenge ที่เป็นข้อมูลจากคนละเซ็ทกัน



รูปที่ 8-35
A reflection attack ที่นำมาใช้กับโพรโตคอลในรูป 8-32

รูปที่ 8-36
Authentication using HMAC



4. สร้างโพรโตคอลให้สามารถต่อต้านการโจมตีแบบ reflection attack ได้

ถ้ากฎเพียงหนึ่งในสี่ข้อนี้ถูกละเมิดก็อาจทำให้โพรโตคอลนี้หยุดการทำงานแล้วเริ่มต้นใหม่ ในตัวอย่างนี้ กฎทั้งสี่ข้อถูกละเมิดพร้อมกัน

ต่อมาให้ย้อนกลับไปดูรูป 8-32 ถามว่าโพรโตคอลนี้ปลอดภัยจากการโจมตีแบบ reflection attack ได้จริงหรือ ปรากฏว่าทรูดี้ยังคงสามารถเล่นกลกับบ็อบหรือใครก็ตามให้เขาตอบคำถามที่เข้าเป็นผู้ถามเองได้อย่างเดิม รูป 8-35 แสดงวิธีการที่ทรูดี้ใช้ ในที่นี่อลิสเป็นตัวละครซึ่งเป็นเครื่องคอมพิวเตอร์ที่มีขีดความสามารในการสร้าง session ขึ้นได้หลาย session พร้อมกัน อลิสเริ่มต้นด้วยการส่ง identity ของตนเองออกมาในข้อความ 1 (ให้ถือว่าเป็น session 1) ทรูดี้สามารถดักจับข้อความนี้ ได้จึงเริ่ม session 2 ด้วยการส่งข้อความประกาศ identity ของตนเองออกไปในข่าวสารที่ 2 โดยอ้างว่าตนเองคือบ็อบ อลิสตอบสนองข่าวสารที่ 2 ด้วยการส่งคำถามเช่น "ถ้าคุณคิดว่าคุณคือบ็อบ ก็ให้พิสูจน์ให้เห็น" ในข่าวสารชิ้นที่ 3 ตอนนี้ทรูดี้ต้องหยุดเพราะไม่ทราบว่าจะพิสูจน์ว่าเป็นบ็อบได้อย่างไร

ทรูดี้จึงกลับไปที่ session 1 ซึ่งเป็นคราวที่เธอจะต้องส่ง challenge เธอจึงส่ง RA ที่ได้รับมาจากข่าวสารที่ 3 ไปให้อลิส อลิสตอบ KAB(RA) กลับมาในข่าวสารที่ 5 ซึ่งช่วยให้ทรูดี้รู้คำตอบใน session 2 จึงตอบกลับไปในข่าวสารที่ 6 ณ จุดนี้ทรูดี้ก็สามารถเข้าสู่ระบบของอลิสได้แล้วเพราะเธอสามารถตองคำถามของอลิส (session 2) ได้แล้ว เธอจึงสามารถยกเลิก session 1 แล้วติดต่อกับอลิสโดยใช้ session 2 แต่เพียงอย่างเดียว

ทรูดี้อาจดำเนินการต่อไปดังที่แสดงในรูป 8-35 เพื่อให้สามารถติดต่อไปทั้งสอง session ในเวลาเดียวกันโดยแทนที่จะส่งหมายเลขเก่ากลับไปยังอลิสเพื่อให้ session 2 เสร็จสิ้นโดยสมบูรณ์ เธอจะรอจนกว่าอลิสจะส่งข่าวสารที่ 7 มายังเธอซึ่งจะเป็น challenge ของอลิสใน session 1 ทรูดี้จึงนำตัวเลขนี้ไปใช้เป็น challenge สำหรับ session 2 คือส่งกลับไปที่อลิสซึ่งก็จะตอบกลับมาด้วย KAB(RA2) ในข่าวสารที่ 9 ท้ายที่สุดทรูดี้ก็นำคำตอบจากอลิสไปตอบเป็นข่าวสารที่ 10 ทำให้กระบวนการพิสูจน์ตัวคนผู้ใช้สำเร็จทั้งสอง session

การโจมตีครั้งนี้แตกต่างไปจากการโจมตีที่แสดงในรูป 8-34 ในครั้งนี้ทรูดี้สามารถมีช่องทางการสื่อสาร (Session) กับอลิสได้ถึงสองช่องทางในเวลาเดียวกัน ในขณะที่การโจมตีครั้งแรกนั้นจะได้ช่องสื่อสารเพียงช่องทางเดียวเท่านั้น แต่ถ้านำกฎเกณฑ์ที่เสนอไปทั้ง 4 ข้อมาใช้ก็จะสามารถป้องกันการโจมตีแบบนี้ได้

รูป 8-36 แสดง Authentication โพรโตคอลอีกแบบหนึ่ง วิธีการนี้ได้นำวิธี HMAC ที่ได้กล่าวถึงไปแล้วมาใช้งาน อลิสเริ่มต้นด้วยการส่งข่าวสาร nonce (RA) มายังบ็อบเป็นข่าวสารที่ 1 บ็อบตอบ

รับด้วยการเลือก nonce ของเขาเอง (RB) และส่งกลับไปพร้อมด้วย HMAC ในที่นี้ HMAC ถูกนำมาใช้สร้างโครงสร้างข้อมูลที่ประกอบด้วย Alice's nonce, Bob's nonce, identity ของทั้งสองคน และคีย์ร่วม KAB โครงสร้างข้อมูลนี้จะถูกนำไป hash เข้ากับ HMAC เช่นการใช้วิธี SHA-1 เมื่ออลิสได้รับข่าวสาร 2 เธอก็จะมี RA ที่เลือกขึ้นมาเอง RB ที่มาพร้อมกับข่าวสาร 2 identity ของทั้งสองคน และคีย์ลับร่วม KAB ซึ่งทราบดีอยู่แล้ว เธอจะสามารถคำนวณ HMAC ด้วยตัวของเธอเองซึ่งถ้าตรงกับที่ส่งมานั้นก็แสดงว่าเธอกำลังคุยอยู่กับบ๊อบเนื่องจากทรูดี้ไม่รู้จัก KAB จึงไม่สามารถสร้าง HMAC ขึ้นมาได้ อลิสจะส่งข่าวสารที่ 3 กลับไปยังบ๊อบซึ่งมีเพียง nonce ของทั้งสองคนและ HMAC

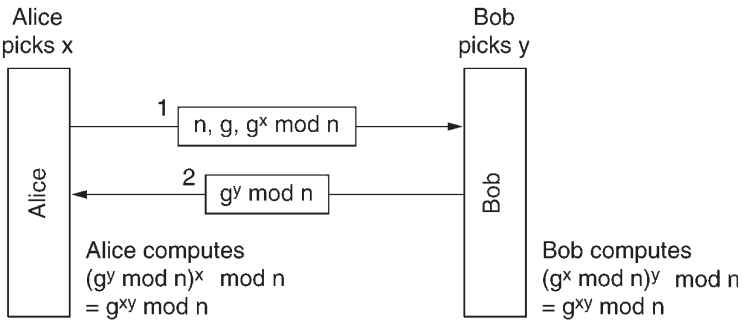
ทรูดี้จะสามารถโจมตีโพรโตคอลนี้ได้หรือไม่ คำตอบก็คือเป็นไปไม่ได้เพราะเธอไม่สามารถบังคับให้ฝ่ายใดฝ่ายหนึ่งทำการเข้ารหัสข้อมูลหรือคำนวณหาค่า hash ตามที่เธอต้องการได้ อีกประการหนึ่งคือ HMAC ทั้งสองอันนี้มีค่าเลขสุ่มที่เลือกขึ้นมาโดยแต่ละฝ่ายเองซึ่งเป็นสิ่งที่ทรูดี้ไม่สามารถเลือกได้

8.7.2 การจัดตั้งคีย์ร่วมโดยวิธี Diffie-Hellman Key Exchange

ที่กล่าวมาแล้วนั้น มีสมมุติฐานว่าทั้งอลิสและบ๊อบมีคีย์ลับร่วมกันอยู่แล้ว แต่ถ้าเขาไม่มีคีย์ลับร่วมกันแล้วจะทำอย่างไร วิธีการหนึ่งที่เป็นไปได้คือการให้อลิสและบ๊อบแลกเปลี่ยนหมายเลขโทรศัพท์ซึ่งกันและกันแล้วแลกเปลี่ยนคีย์ลับกันทางโทรศัพท์ แต่การสนทนาก็อาจเริ่มต้นด้วยการที่บ๊อบถามว่าเขาจะทราบได้อย่างไรว่ากำลังคุยอยู่กับอลิส เขาทั้งสองคนอาจจะนัดพบกันที่ใดที่หนึ่งเพื่อที่แต่ละฝ่ายจะได้นำบัตรแสดงตน เช่น บัตรประชาชนมาเพื่อให้แน่ใจได้ว่าบ๊อบคือบ๊อบและอลิสคืออลิส แต่ถ้าทั้งสองคนต่างก็มีธุระยุ่งมากหรืออยู่ห่างจากกันไกลมาก ก็อาจทำให้การนัดหมายนั้นต้องล่าช้าออกไปเป็นเดือนได้ โชคดีที่มีวิธีการหนึ่งที่ช่วยให้คนแปลกหน้าสองคนสามารถจัดตั้งคีย์ลับร่วมกันได้แม้ว่าจะมีผู้อื่นอย่าง ทรูดี้กำลังจับตามองอยู่ก็ตาม

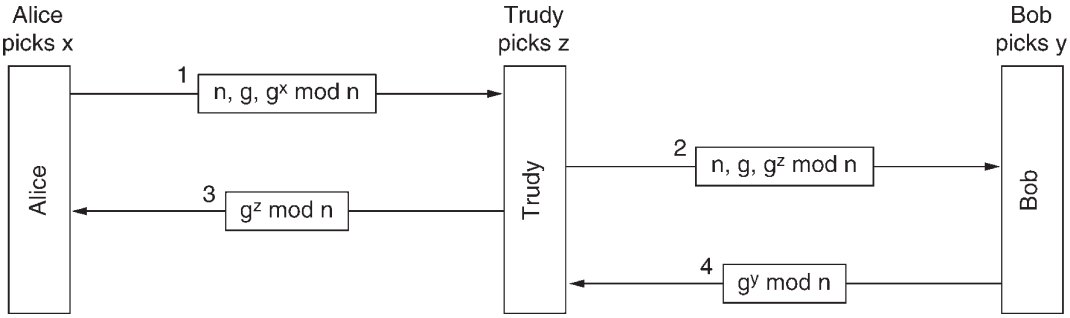
โพรโตคอลนี้เรียกว่า Diffie-Hellman key exchange ทำงานดังนี้ อลิสและบ๊อบจะตกลงเลือกหมายเลขขึ้นมาสองหมายเลข (n และ g) ที่มีขนาดใหญ่มากและเป็นเลขจำนวนเฉพาะ (prime number) เลขทั้งสองนี้ไม่จำเป็นจะต้องเป็นความลับ คือทั้งสองคนสามารถเลือกตัวเลขขึ้นมาได้โดยอิสระ ขั้นต่อไปอลิสเลือกตัวเลขขึ้นมาอีกตัวหนึ่ง (x) ซึ่งเป็นตัวเลขขนาดใหญ่ (512 บิต) และเก็บไว้เป็นความลับ ในเวลาเดียวกัน บ๊อบก็จะเลือกหมายเลขลับ (y) ของตนเองขึ้นมาหนึ่งตัว

อลิสเริ่มต้นกระบวนการแลกเปลี่ยนหมายเลขด้วยการส่งข่าวสารไปยังบ๊อบซึ่งประกอบด้วย (n, g, g^x mod n) ดังที่แสดงในรูป 8-37 บ๊อบตอบอลิสมาด้วยหมายเลข (g^y mod n) ต่อไปอลิสทำการคำนวณ $(g^y \bmod n)^x \bmod n$ ในขณะที่บ๊อบก็ทำการคำนวณเลข $(g^x \bmod n)^y \bmod n$ ตามกฎของ



รูปที่ 8-37
The Diffie-Hellman key exchange

รูปที่ 8-38
The bucket brigade หรือ man-in-the-middle attack



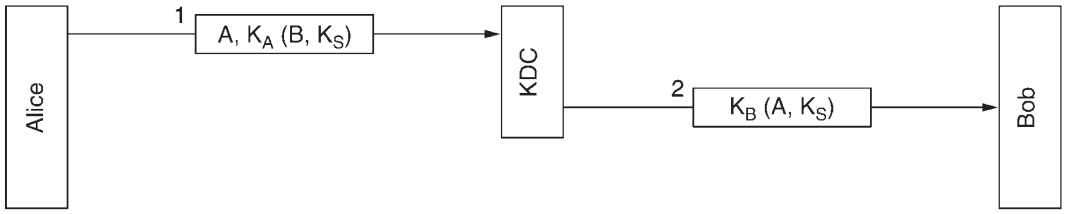
การใช้ modulation ผลการคำนวณของทั้งสองคนจะมีค่าเท่ากับ g^{xy} mod n ทำให้ทั้งบ๊อบและอลิสมีคีย์ลับร่วมกันตามต้องการ

สมมุติว่าทรูดี้สามารถดักรับข้อมูลได้ทั้งหมด ทำให้เธอทราบค่า g และ n จากข่าวสารที่ 1 ถ้าทรูดี้สามารถคำนวณหาค่า x และ y ได้ก็จะทราบค่าคีย์ลับระหว่างคนทั้งสองนี้ ปัญหาก็คือด้วยข้อมูลที่มีอยู่คือ g^x mod n ทรูดี้จะไม่สามารถหาค่า x ได้ และในทำนองเดียวกันเธอก็จะไม่สามารถหาค่า y จาก g^y mod n ได้เช่นกัน ทั้งนี้ในปัจจุบันไม่มีอัลกอริทึมใดที่จะสามารถคำนวณหาค่า modulo ของเลขจำนวนเฉพาะขนาดใหญ่ได้

เพื่อให้เห็นภาพที่ชัดเจนจะยกตัวอย่างการคำนวณ (ที่ไม่มีใครนำไปใช้งานจริง) ดังนี้ ให้ n = 47 และ g = 3 อลิสเลือกตัวเลข x = 8 และบ๊อบเลือกตัวเลข y = 10 อลิสจะส่งข่าวสารไปยังบ๊อบเป็น (47, 3, 28) เพราะว่า 3^8 mod 47 = 28 และข่าวสารที่บ๊อบส่งไปให้อลิสคือ 17 (3^{10} mod 47 = 17) อลิสทำการคำนวณ 17^8 mod 47 = 4 ในเวลาเดียวกันบ๊อบคำนวณ 28^{10} mod 47 = 4 ทั้งอลิสและบ๊อบต่างก็มีวิธีการของตนเองในการคำนวณหาค่าคีย์ลับร่วมกันซึ่งในที่นี้ก็คือ 4 นั่นเอง ทรูดี้จำต้องทำการคำนวณ 3^x mod 47 = 28 ซึ่งด้วยตัวเลขที่มีขนาดเล็กมากนี้ทรูดี้ก็อาจโชคดีสามารถคำนวณหาค่า x ได้ แต่สำหรับตัวเลขที่ใช้งานจริงซึ่งมีขนาดหลายร้อยบิตนั้นจะไม่สามารถเป็นไปได้เลยที่จะหาค่า x แม้ว่าจะใช้วิธีการคำนวณแบบขนาน (parallel computing) เข้ามาช่วยก็ตาม

แม้ว่าจะมีวิธีการที่สวยหรูอย่างเช่น Diffie-Hellman ก็ตามแต่ก็ยังอาจมีปัญหาขึ้นมาได้นั่นคือ เมื่อบ๊อบได้รับข่าวสาร (47, 3, 28) นั้นบ๊อบจะทราบได้อย่างไรว่าเป็นตัวเลขที่ส่งมาจากอลิส คำตอบก็คือไม่มีทางเป็นไปได้เลย รูป 8-38 แสดงให้เห็นว่าทรูดี้สามารถใช้ประโยชน์จากสถานะการณ์นี้ได้อย่างไรในที่นี้ ในขณะที่อลิสและบ๊อบกำลังเลือกตัวเลข x และ y ของตนเองอยู่นั้น ทรูดี้ก็จะเลือกเลขสุ่มของตนเองขึ้นมาหนึ่งหมายเลข (z) อลิสส่งข่าวสารที่ 1 มาเพื่อให้กับบ๊อบ แต่ทรูดี้สามารถดักเอาไว้ได้และจัดการส่งข่าวสารที่ 2 มาให้บ๊อบแทนโดยใช้เลข g และ n ตัวเดิมแต่ใส่เลข z เข้าไปแทนเลข x ของอลิส ทรูดี้ก็ส่งข่าวสาร 3 กลับไปให้อลิส ต่อมาเมื่อบ๊อบส่งข่าวสาร 4 มาทรูดี้ก็สามารถดักรับไว้ได้และเก็บไว้ที่ตนเอง

ขั้นต่อไปทุกคนทำการคำนวณเลขของตนเอง อลิสจะคำนวณเลขคีย์ลับร่วมออกมาเป็น g^{xz} mod n ซึ่งเหมือนกับ ทรูดี้ (ใช้สำหรับแลกเปลี่ยนข่าวสารกับอลิส) ส่วนบ๊อบก็จะคำนวณ g^{yz} mod n ซึ่งเหมือนกับทรูดี้ในส่วนที่ใช้ติดต่อกับบ๊อบ อลิสคิดว่าตนเองกำลังติดต่ออยู่กับบ๊อบจึงสร้าง session key



รูปที่ 8-39
An authentication protocol ที่ใช้ KDC

ขึ้นมา ส่วนบ็อบก็ทำในทำนองเดียวกัน ทุกข่าวสารที่อลิสส่งมาใน session ที่มีการเข้ารหัสจะถูกจับได้โดยทรูดี้ซึ่งจะสามารถถอดรหัสได้ จับเก็บไว้ แก้ไขตามที่ต้องการ และอาจจะส่งมาให้บ็อบอีกต่อหนึ่ง ส่วนในทางกลับกันก็มีลักษณะเช่นเดียวกัน ทรูดี้สามารถมองเห็นทุกอย่างและแก้ไขข่าวสารได้ตามที่ต้องการในขณะที่ทั้งอลิสและบ็อบต่างก็ตกอยู่ในภาพลวงตาว่าตนเองนั้นกำลังอยู่ในช่องทางสื่อสารที่ปลอดภัย วิธีการโจมตีเช่นนี้เรียกว่า bucket brigade attack หรือ man-in-the-middle attack

8.7.3 การตรวจสอบผู้ใช้โดยการใช้ศูนย์แจกจ่ายคีย์

การจัดทำคีย์ลับเมื่อต้องการติดต่อกับคนแปลกหน้านั้นอาจเป็นวิธีการที่นำมาใช้ได้ แต่ในทางกลับกันอาจเป็นสิ่งที่ควรหลีกเลี่ยงก็ได้ ในการสื่อสารกับคน n คน จำเป็นจะต้องใช้คีย์เป็นจำนวน n คีย์ สำหรับคนที่มีผู้ต้องการติดต่อด้วยเป็นจำนวนมากก็คงจะต้องหาวิธีการบริหารจัดการคีย์ทั้งหลายที่มีอยู่

วิธีการที่แตกต่างออกไปได้แก่การจัดตั้งศูนย์แจกจ่ายคีย์ (Key Distribution Center; KDC) ที่ไว้วางใจได้ขึ้นมาซักแห่งหนึ่ง ในรูปแบบนี้ผู้ใช้แต่ละคนจะมีคีย์ตัวหนึ่งที่ใช้งานร่วมกับ KDC การตรวจสอบผู้ใช้และการกำหนด session key ก็จะกลายเป็นหน้าที่ของ KDC รูป 8-39 แสดงวิธีการใช้ KDC แบบที่ง่ายที่สุดวิธีหนึ่งในการตรวจสอบผู้ใช้ซึ่งเกี่ยวข้องกับผู้ใช้สองคนกับ KDC ที่ไว้วางใจได้แห่งหนึ่ง

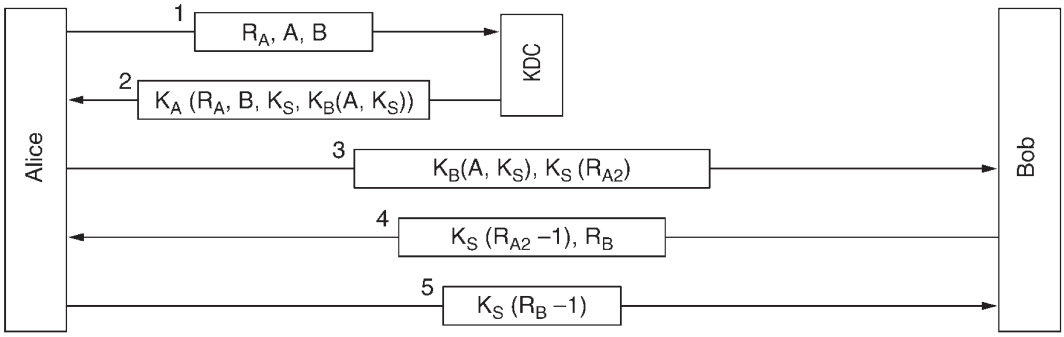
แนวความคิดของโพรโตคอลนี้เป็นดังนี้ อลิสทำการเลือก session key (KS) และบอกให้ KDC ทราบว่าเธอต้องการสื่อสารกับบ็อบโดยใช้ KS ข่าวสารนี้ถูกเข้ารหัสโดยใช้คีย์ลับ (KA) ที่อลิสใช้งานร่วมกับ KDC ต่อไป KDC ทำการถอดรหัสข่าวสารนี้และดึงข้อมูล identity ของบ็อบและ session key ออกมา จากนั้นจึงสร้างข่าวสารใหม่ที่มี identity ของอลิสและ session key แล้วส่งข่าวสารนี้ไปยังบ็อบ การเข้ารหัสนี้ทำโดยใช้คีย์ KB ซึ่งเป็นคีย์ลับร่วมระหว่างบ็อบกับ KDC เมื่อบ็อบถอดรหัสข่าวสารนี้เขาก็จะทราบว่าอลิสต้องการสื่อสารกับเขาและต้องการใช้ session key ตามที่ส่งมาให้

การตรวจสอบผู้ใช้นี้เกิดขึ้นโดยไม่ต้องเสียค่าใช้จ่าย KDC ทราบว่าข่าวสาร 1 นั้นจะต้องมาจากอลิส เนื่องจากไม่มีผู้ใดทราบวิธีที่จะเข้ารหัสข้อมูลนอกจากอลิส ทำนองเดียวกัน ข่าวสาร 2 ต้องมาจาก KDC ซึ่งเป็นคนเดียวที่บ็อบไว้ใจและได้มอบคีย์ลับร่วมไปให้

โชคไม่ดีที่โพรโตคอลนี้มีข้อเสียที่ร้ายแรงมาก ทรูดี้ต้องการเงินจำนวนหนึ่งเธอจึงคิดหาวิธีที่จะให้บริการแก่อลิส ด้วยการนำเสนอข้อแลกเปลี่ยนที่ดี และทำให้ได้งานนั้นมา หลังจากที่ทำงานให้แก่อลิสทรูดี้ก็ขอร้องให้อลิสจ่ายเงินค่าตอบแทนให้เธอผ่านทางบัญชีธนาคาร อลิสจึงจัดตั้งช่องสื่อสารและ session key กับทางธนาคาร ในที่นี้ก็คือบ็อบ จากนั้นเธอจึงส่งข่าวสารไปยังบ็อบแจ้งว่าเธอต้องการให้โอนเงินให้ทรูดี้

ในเวลาเดียวกันทรูดี้ก็ใช้วิธีการเดิมคือแอบเข้าไปขโมยข้อมูลในระบบเครือข่าย เธอทำการสำเนา

รูปที่ 8-40
The Needham-Schroeder authentication protocol



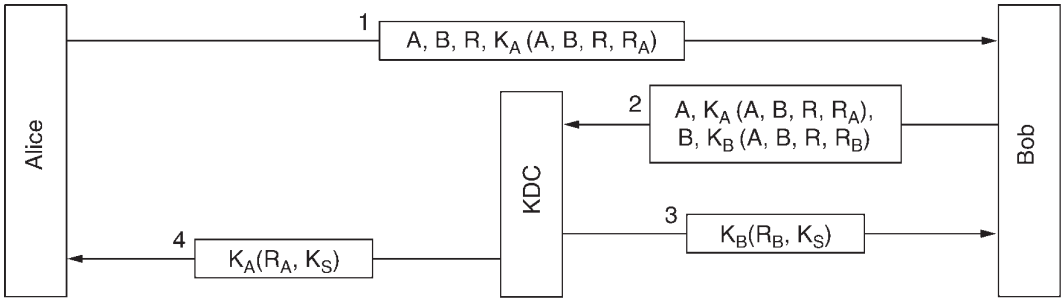
ข้อมูลในข่าวสาร 2 ในรูป 8-39 และคำสั่งให้โอนเงินซึ่งถูกส่งตามมา ต่อมาทรูดี้จัดการส่งสำเนาข่าวสารทั้งสองนี้กลับไปยังบ็อบ บ็อบอาจคิดว่าอลิสได้ว่าจ้างทรูดี้อีกครั้งหนึ่งจึงไปจ่ายเงินให้แก่ทรูดี้ตามความต้องการ (ครั้งที่สอง) ของอลิส เหตุการณ์เช่นนี้อาจเกิดขึ้นซ้ำแล้วซ้ำเล่าตราบเท่าที่อลิสยังไม่รู้ว่าเงินตนเองหายไป วิธีการโจมตีในลักษณะนี้เรียกว่า replay attack

มีหลายหนทางที่ได้รับการพัฒนาขึ้นมาเพื่อแก้ปัญหา replay attack วิธีการแรกคือการใส่การบันทึกเวลา (time stamp) เข้าไปในแต่ละข่าวสาร ดังนั้นใครก็ตามที่ได้รับข่าวสารที่เก่าแล้วก็จะสามารถลบข่าวสารนั้นทิ้งไปได้ ปัญหาของวิธีการนี้ก็คือ ไม่มีทางที่จะตั้งเวลานาฬิกาของเครื่องคอมพิวเตอร์ในระบบเครือข่ายให้ตรงกันได้เลย ดังนั้นจึงต้องมีช่วงเวลาหนึ่งที่อนุญาตให้เวลาที่บันทึกลงไปนั้นยังคงใช้ได้แม้ว่าอาจเป็นเวลาที่ช้าหรือเร็วไปกว่าเวลาของนาฬิกาผู้รับข่าวสาร ทำให้ทรูดี้ยังคงสามารถใช้วิธีการเดิมและได้รับเงินไปใช้ฟรี

วิธีการแก้ปัญหาวิธีที่สองคือ ใส่ข้อมูล nonce เข้าไปในทุกข่าวสาร ผู้ร่วมการสื่อสารแต่ละคนก็จะต้องจดจำ nonce ของผู้ที่ตนสื่อสารด้วยและกำจัดข่าวสารที่ใช้ nonce ที่ซ้ำกับของเก่าออกไป แต่ nonce จะต้องถูกจดจำไว้ตลอดไปมิฉะนั้นทรูดี้อาจทำการส่งข่าวสารเก่าที่มีอายุ 5 ปีมาแล้วอีกก็ได้ นอกจากนี้ถ้าคอมพิวเตอร์เกิดการ crash ก็จะทำให้สูญเสีย nonce ที่บันทึกไว้ทั้งหมดก็จะทำให้คอมพิวเตอร์เครื่องนั้นตกเป็นเป้าหมายการโจมตีแบบ replay attack อีกครั้งหนึ่ง การบันทึกเวลาและ nonce อาจถูกนำมาใช้งานร่วมกันเพื่อจำกัดปริมาณ nonce ที่จะต้องจดจำ แต่โพรโตคอลนี้ก็จะต้องมีการทำงานที่ซับซ้อนมากขึ้นกว่าเดิม

วิธีการที่ซับซ้อนกว่าที่ทำให้การตรวจสอบผู้ใช้น่าเชื่อมถือมากขึ้นเรียกว่า multiway challenge-response protocol ซึ่งได้แก่ Needham-Schroeder authentication protocol ดังแสดงในรูป 8-40

โพรโตคอลนี้เริ่มต้นด้วยการที่อลิสแจ้งแก่ KDC ว่าเธอต้องการสื่อสารกับบ็อบ ข่าวสารนี้ประกอบด้วยตัวเลขที่เลือกมาแบบสุ่มขนาดใหญ่ RA กำหนดให้เป็น nonce KDC จะส่งข่าวสารตอบกลับมาประกอบด้วยเลข RA, session key, และ ticket ที่อลิสจะต้องส่งไปให้บ็อบ การส่งเลข RA นั้นก็เพื่อรับประกันว่าข่าวสารนั้นเป็นข่าวใหม่ นอกจากนี้ยังมี identity ของบ็อบส่งมาด้วยเพื่อป้องกันไม่ให้ทรูดี้ทำการเขียน identity ของเธอเองเข้ามาแทนที่ B ในข่าวสาร 1 ซึ่งจะทำให้ KDC ทำการเข้ารหัส ticket ที่ตอนท้ายของข่าวสารที่ 2 ด้วย KT แทนที่จะเป็น KB ข้อความ ticket จะถูกเข้ารหัสด้วย KB ซึ่งถูก



รูปที่ 8-41
The Otway-Rees authentication protocol

ใส่เข้าไว้ภายในข่าวสารที่ถูกเข้ารหัสอีกชั้นหนึ่งเพื่อป้องกันไม่ให้ทรูดี้แก้ไขข่าวสารด้วยข้อมูลอื่นในระหว่างการส่งไปยังอลิส

ต่อไปอลิสจะส่ง ticket ไปยังบ๊อบพรอ้มตัวตัวเลขสุ่มตัวใหม่ RA2 ซึ่งจะถูกเข้ารหัสด้วย session key KS ในข่าวสารที่ 4 บ๊อบจะส่งข้อมูล KS(RA2 -1) กลับมาเพื่อพิสูจน์ให้อลิสแน่ใจว่าเธอกำลังสื่อสารอยู่กับบ๊อบตัวจริง การส่ง KS(RA2) นั้นจะไม่มีประโยชน์เพราะว่าทรูดี้อาจขโมยเลขนี้ไปจากข่าวสารที่ 3

ภายหลังจากที่ได้รับข่าวสารที่ 4 อลิสก็จะแน่ใจว่าเธอกำลังสื่อสารอยู่กับบ๊อบจนถึงขณะนี้ยังไม่มีการใช้ข่าวสาร replay เกิดขึ้นเลย อย่างไรก็ตามอลิสพึ่งจะสร้าง RA2 เมื่อไม่กี่มิลลิวินาทีที่ผ่านมานี้เอง วัตถุประสงค์ของการส่งข่าวสารที่ 5 ก็เพื่อทำให้บ๊อบแน่ใจว่าเขากำลังสื่อสารอยู่กับอลิสและไม่มีการใช้ข่าวสาร replay เกิดขึ้นเลย การที่ให้แต่ละฝ่ายนั้นสร้าง challenge และ response ขึ้นมาเพียงครั้งเดียวนั้นเป็นการกำจัดปัญหา replay attack ออกไป

แม้ว่าโพรโตคอลนี้ดูเหมือนว่ามีความเข็มแข็งมาก แต่ก็ยังมีจุดบกพร่องอยู่บ้าง ถ้าทรูดี้สามารถหา session key ตัวเก่าที่เคยใช้งานมาได้ในรูปของ plaintext เธอก็จะสามารถเริ่มการติดต่อ session ใหม่กับบ๊อบได้โดยการ replay ข่าวสารที่ 3 ซึ่งเรียกว่าเป็น compromised key และอาจทำให้บ๊อบเชื่อว่าเธอคืออลิสได้

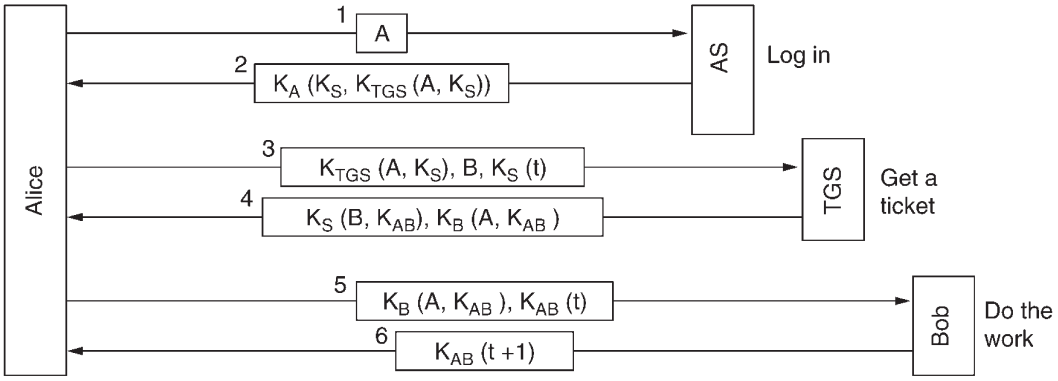
Needham และ Schroeder ได้แก้ไขโพรโตคอลนี้ให้สามารถแก้ปัญหาที่เกิดขึ้นได้ ต่อมา Otway และ Rees ได้สร้างโพรโตคอลที่สั้นกว่าดังแสดงในรูป 8-41

8.7.4 การตรวจสอบผู้ใช้โดยใช้ Kerberos

Authentication protocol อันหนึ่งที่ถูกนำไปใช้ในระบบงานจริงหลายระบบ (รวมทั้ง Windows 2000) คือ Kerberos ซึ่งเป็นวิธีที่พัฒนาต่อมาจากวิธีการของ Needham-Schroeder Kerberos ได้รับการพัฒนาโดยมหาวิทยาลัย M.I.T. เพื่อช่วยให้เครื่องผู้ใช้สามารถติดต่อใช้ทรัพยากรของระบบเครือข่ายได้อย่างปลอดภัย สิ่งที่แตกต่างเป็นอย่างมากจากวิธีของ Needham-Schroeder คือข้อสมมุติฐานที่ว่านาฬิกาของเครื่องคอมพิวเตอร์บนระบบเครือข่ายนั้นตั้งไว้ตรงกัน โพรโตคอลรุ่นที่ 4 (V 4) เป็นรุ่นที่ได้รับความนิยมนำมาใช้งานมากที่สุด

ในระบบ Kerberos จะต้องประกอบด้วยเซิบเวอร์จำนวนสามเครื่องคือ

รูปที่ 8-42
การทำงานของ
Kerveros รุ่น 4



1. Authentication Server (AS): ทำการตรวจสอบผู้ใช้ในระหว่างการ log-in

2. Ticket-Granting Server (TGS): เป็นผู้ออกใบรับรอง "Proof of identity tickets"

3. Bob the server: เป็นเซิบเวอร์ที่ทำงานตามที่อลิสต้องการ

AS นั้นทำหน้าที่เหมือน KDC คือจะจัดเก็บคีย์ลับที่ใช้งานร่วมกับผู้ใช้แต่ละคน TGS ทำหน้าที่ในการออก ticket ที่สามารถนำมาใช้บอกเครื่องเซิบเวอร์ว่าผู้ที่ถือ ticket อยู่นั้นเป็นใคร

ในการจัดตั้ง session การสื่อสาร อลิสจะนั่งทำงานอยู่ที่เครื่องคอมพิวเตอร์เครื่องหนึ่งในบริษัท และพิมพ์ชื่อของเธอเข้าไป คอมพิวเตอร์เครื่องนั้นจะส่งชื่อเธอไปที่ AS ในลักษณะของ plaintext ดังที่แสดงในรูป 8-42 สิ่งที่ได้รับตอบกลับมาคือ session key KS และ ticket KTGS(A, KS) จาก TGS ข้อมูลสองตัวนี้จะถูกใส่เข้ามาในแพ็กเก็ตเดียวกันและถูกเข้ารหัสโดยใช้คีย์ลับร่วมของอลิสเพื่อให้อลิสเท่านั้นที่จะสามารถถอดรหัสออกมาได้ เมื่อข่าวสาร 2 เดินทางมาถึงเครื่องคอมพิวเตอร์จะสอบถามรหัสผ่านจากอลิส รหัสผ่านจะถูกนำไปใช้ในการสร้าง KA เพื่อจะได้ถอดรหัสข่าวสาร 2 และได้รับ session key และ TGS ticket จากนั้นเครื่องคอมพิวเตอร์ของอลิสจะลบรหัสผ่านของอลิสทิ้งไปเพื่อเป็นการรักษาความลับเอาไว้ ถ้าทรูดี้พยายามที่จะ log-in โดยการแอบอ้างเป็นอลิส เมื่อเธอพิมพ์รหัสผ่าน (ที่ไม่ถูกต้อง) เข้าไปเครื่องคอมพิวเตอร์จะสามารถทราบได้ทันทีเพราะส่วนที่เป็นข้อความมาตรฐานในข่าวสาร 2 จะไม่ถูกต้อง

หลังจากที่อลิส log-in เข้าไปแล้วเธออาจจะบอกเครื่องคอมพิวเตอร์ที่เธอใช้งานอยู่ว่าเธอต้องการติดต่อกับบ็อบซึ่งทำหน้าที่เป็นไฟล์เซิบเวอร์ เครื่องคอมพิวเตอร์จะส่งข่าวสาร 3 ไปยัง TGS เพื่อขอ ticket ที่จะสามารถใช้งานร่วมกับบ็อบได้ ข้อความที่สำคัญในข่าวสารนี้คือ KTGS(A, KS) ซึ่งจะถูกเข้ารหัสโดยใช้คีย์ลับของ TGS และใช้ในการพิสูจน์ว่าผู้ส่งที่แท้จริงคืออลิส TGS จะตอบสนองโดยการสร้าง session key KAB เพื่อให้อลิสไว้ใช้งานร่วมกับบ็อบ คีย์นี้จะถูกส่งกลับมาใน 2 รูปแบบโดยรูปแบบแรกนั้นจะเข้ารหัสโดยใช้ session key KS ของอลิสเพื่อให้อลิสสามารถอ่านได้ และรูปแบบที่สองใช้คีย์ลับร่วม KB ของบ็อบ เพื่อให้บ็อบสามารถอ่านได้

ทรูดี้อาจจะสามารถดักจับข่าวสาร 3 ได้และพยายามนำมาใช้อีก แต่ก็จะต้องพบกับปัญหาการบันทึกเวลาทำงาน (time stamp) t ที่เข้ารหัสเอาไว้และถูกส่งมาพร้องกับข่าวสารนั้น ทรูดี้จะไม่สามารถเปลี่ยนเวลาที่บันทึกให้เป็นเวลาในปัจจุบันได้เนื่องจากเธอไม่ทราบ session key KS ซึ่งอลิสใช้

ในการสื่อสารกับ TGS แม้ว่าทรูดี้จะสามารถส่งข่าวสาร 3 ซ้ำ (replay) ได้อย่างรวดเร็วมาก แต่สิ่งที่เธอจะได้รับตอบกลับมาก็คือสำเนาของข่าวสาร 4 ที่เธอก็ไม่สามารถอ่านออกได้

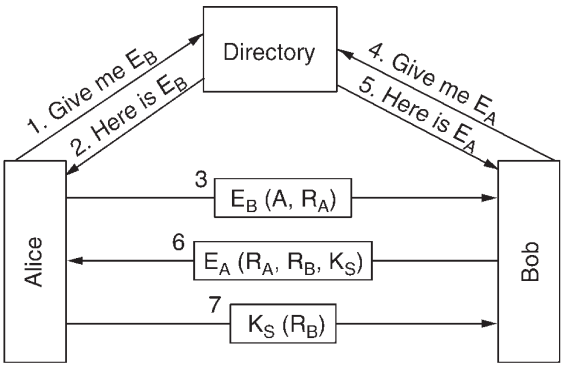
ขั้นต่อไปอลิสสามารถส่ง KAB ไปยังบ็อบเพื่อเริ่มต้น session การสื่อสารระหว่างกัน ข่าวสารที่ส่งไปนี้ก็มีการบันทึกเวลาเอาไว้ด้วย ข่าวสารที่ตอบกลับมาจากบ็อบจะเป็นสิ่งยืนยันว่าเธอกำลังสื่อสารอยู่กับบ็อบตัวจริง

ภายหลังจากการแลกเปลี่ยนข่าวสารชุดนี้จบลง อลิสจะสามารถสื่อสารกับบ็อบได้โดยใช้ session key KAB ต่อมาถ้าอลิสต้องการติดต่อกับเซิบเวอร์อีกเครื่องหนึ่งคือ คาร์โรล เธอก็เพียงส่งข่าวสารหมายเลข 3 ไปยัง TGS โดยเปลี่ยน identity จากของบ็อบ (B) ไปเป็นของคาร์โรล (C) TGS จะตอบสนองโดยการส่ง ticket KC กลับมาเพื่อให้อลิสส่งต่อไปให้กับคาร์โรลซึ่งคาร์โรลก็จะเชื่อและจะสามารถสื่อสารกับอลิสได้

กระบวนการทำงานนี้จะช่วยให้อลิสสามารถติดต่อกับเซิบเวอร์ในระบบเครือข่ายได้ทุกตัวได้อย่างปลอดภัยและรหัสผ่านของเธอก็ไม่ได้ถูกส่งออกไปจากเครื่องคอมพิวเตอร์ที่เธอใช้งานอยู่เลย อันที่จริงรหัสผ่านนั้นจะอยู่ในเครื่องคอมพิวเตอร์เพียงไม่กี่มิลลิวินาทีเท่านั้นก็จะถูกทำลายทิ้งไป อย่างไรก็ตามเซิบเวอร์แต่ละตัวจะทำการตรวจสอบผู้ใช้ด้วยตนเอง เมื่ออลิสส่ง ticket ไปยังบ็อบนั้นถือว่าเป็นการพิสูจน์ในขั้นต้นว่าบ็อบกำลังสื่อสารอยู่กับใคร การที่อลิสจะสามารถทำอะไรได้บ้างนั้นขึ้นอยู่กับว่าบ็อบจะอนุญาตให้ทำได้หรือไม่

เนื่องจากผู้ออกแบบ Kerberos ไม่ได้คาดหวังว่าทั่วทั้งโลกจะเชื่อใจ AS เพียงตัวเดียว จึงได้ออกแบบให้มีการแบ่งเขตเป็น อาณาเขต (realm) หลายแห่ง แต่ละแห่งจะมี AS และ TGS เป็นของตนเอง ในการขอ ticket จากเซิบเวอร์ในอาณาจักรอื่นอลิสจะต้องขอให้ TGS ของเธอออก ticket ที่สามารถติดต่อกับ TGS ในอาณาจักรนั้นๆ ได้ ถ้า TGS ของอาณาจักรทั้งสองต่างก็ลงทะเบียนเอาไว้ซึ่งกันและกัน (วิธีเดียวกับที่ local server ทำเพื่อให้สามารถไว้ใจซึ่งกันและกันได้) TGS ของอลิสจะสามารถออก ticket ที่เอาไปใช้กับ TGS ที่อลิสต้องการติดต่อด้วยได้ ซึ่งเธอก็จะสามารถติดต่อขอ ticket เพื่อสื่อสารกับเซิบเวอร์ใดๆ ในอาณาจักรนั้นได้ด้วยวิธีการเดียวกับที่เธอติดต่อกับเซิบเวอร์ต่างๆ ในอาณาจักรของตนเอง

Kerberos V.5 นั้นมีขีดความสามารถสูงกว่ารุ่นที่ 4 มีค่าต้นทุนในการดำเนินงาน (overhead)



รูปที่ 8-43 การตรวจสอบผู้ใช้ซึ่งกันและกันโดยใช้การเข้ารหัสแบบคีย์สาธารณะ

สูงกว่าด้วย และยังใช้ OSI ASN.1 สำหรับการอธิบายโครงสร้างข้อมูลและมีการเปลี่ยนแปลงโพรโตคอลเล็กน้อย ยิ่งกว่านั้นในรุ่น 5 ยังกำหนดอายุของ ticket ให้ยาวนานกว่าเดิม ยินยอมให้มีการขยายอายุของ ticket ได้ และในทางทฤษฎีรุ่น 5 ไม่ได้ใช้การเข้ารหัสแบบ DES เหมือนในรุ่น 4 แต่สามารถเลือกใช้ก็ได้ และมีการสนับสนุนการติดต่อกับหลายอาณาจักรด้วยการสร้าง ticket ที่สามารถติดต่อกับ TGS ได้หลายตัว

8.7.5 การตรวจสอบผู้ใช้โดยใช้การเข้ารหัสลับด้วยคีย์สาธารณะ

การตรวจสอบซึ่งกันและกันสามารถทำได้โดยใช้การเข้ารหัสแบบคีย์สาธารณะ อลิสเริ่มต้นด้วยการค้นหาคีย์สาธารณะของบ็อบ ถ้ามีการใช้ PKI พร้อมกับไดเร็กทอรี่เซิบเวอร์ที่จัดส่งใบรับรองสำหรับคีย์สาธารณะ อลิสก็จะสามารถหาคีย์สาธารณะที่ถูกต้องของบ็อบได้ดังแสดงในรูป 8-43 (ข่าวสารที่ 1) คำตอบที่ได้ในข่าวสาร 2 คือ ใบรับรอง X.509 ที่มีคีย์สาธารณะของบ็อบอยู่ด้วย เมื่ออลิสตรวจสอบว่าการลงชื่อรับรองนั้นถูกต้อง เธอก็สามารถส่งข่าวสาร 3 ไปยังบ็อบซึ่งมี identity และ nonce ของอลิสอยู่

เมื่อบ็อบได้รับข่าวสารนี้ เขาจะไม่ทราบว่าข่าวสารถูกส่งมาจากอลิสหรือทรูดี้หรือใครก็ตาม แต่เข้าก็จะทำงานต่อไปโดยสอบถามไปยังไดเร็กทอรี่เซิบเวอร์เพื่อขอคีย์สาธารณะของอลิส (ข่าวสารที่ 4) ซึ่งก็จะได้รับคำตอบกลับมาเป็นข่าวสาร 5 จากนั้นเขาจึงส่งข่าวสารไปยังอลิสโดยใช้ nonce ของอลิส (RA), nonce ของเขาเอง (RB) และ session key KS ดังแสดงในข่าวสาร 6

เมื่ออลิสได้รับข่าวสาร 6 เธอจะทำการถอดรหัสโดยใช้คีย์ส่วนตัวของเธอเองซึ่งก็จะเห็น nonce ของตนเอง (RA) ข่าวสารนี้จะต้องมาจากบ็อบแน่ๆ เพราะทรูดี้ไม่มีทางทราบ nonce ของอลิส ยิ่งกว่านั้นยังต้องเป็นข่าวสารที่สดไม่ใช่ข่าวสารซ้ำ (replay) เนื่องจากเธอพึ่งจะส่ง RA ไปยังบ็อบ อลิสตกลงที่จะใช้ session key ด้วยการส่งข่าวสาร 7 กลับไปหาบ็อบ เมื่อบ็อบมองเห็น RB ที่เข้ารหัสด้วย session key ที่เขาเป็นคนสร้างขึ้นมาเอง เขาก็จะทราบว่าอลิสได้รับข่าวสาร 6 และเขาก็จะทำการตรวจสอบความถูกต้องของ RA

ทรูดี้จะสามารถทำอย่างไรได้บ้าง ทรูดี้อาจจะดักจับข่าวสารที่ 3 และพยายามหลอกบ็อบว่าเธอคืออลิส แต่เมื่ออลิสมองเห็น RA ซึ่งเป็น nonce ที่เธอไม่ได้สร้างขึ้นมา อลิสก็จะยกเลิกการติดต่อ (อาจพยายามใหม่ทีหลัง) ส่วนทรูดี้ก็จะไม่สามารถปลอมข่าวสาร 7 ที่จะต้องส่งไปให้บ็อบได้เพราะเธอไม่ทราบ RB และ KS เนื่องจากจำเป็นต้องใช้คีย์ส่วนตัวของอลิสในการถอดรหัสนี้ ความพยายามของทรูดี้จึงล้มเหลว

8.8 การรักษาความปลอดภัยให้กับอีเมลล์

เมื่อข่าวสารประเภทอีเมลล์ถูกส่งระหว่างสถานที่สองแห่งที่อยู่ห่างจากกันข่าวสารนี้มักจะถูกส่งผ่านเครื่องคอมพิวเตอร์หรืออุปกรณ์สื่อสารหลายสิบเครื่องในระหว่างทางที่ส่งไป อุปกรณ์ใดๆ ที่อยู่ระหว่างทางนี้จะสามารถอ่านและบันทึกข่าวสารในอีเมลล์เพื่อประโยชน์ในอนาคตได้ ในทางปฏิบัติความเป็นส่วนตัวนั้นไม่มีอยู่ไม่ว่าคนจำนวนมากจะคิดอย่างไร ถึงกระนั้นก็ตามคนจำนวนมากก็ยังมีความต้องการที่จะส่งอีเมลล์ไปยังผู้รับโดยมีผู้รับเท่านั้นที่จะสามารถอ่านข้อความในอีเมลล์ได้ ความต้องการนี้ได้เป็นตัวกระตุ้นให้คนจำนวนหนึ่งทำการประยุกต์การเข้ารหัสข้อมูลที่ได้กล่าวมาข้างต้นเข้าไปกับอีเมลล์เพื่อให้เกิดเป็นอีเมลล์ที่มีความปลอดภัยขึ้นมา

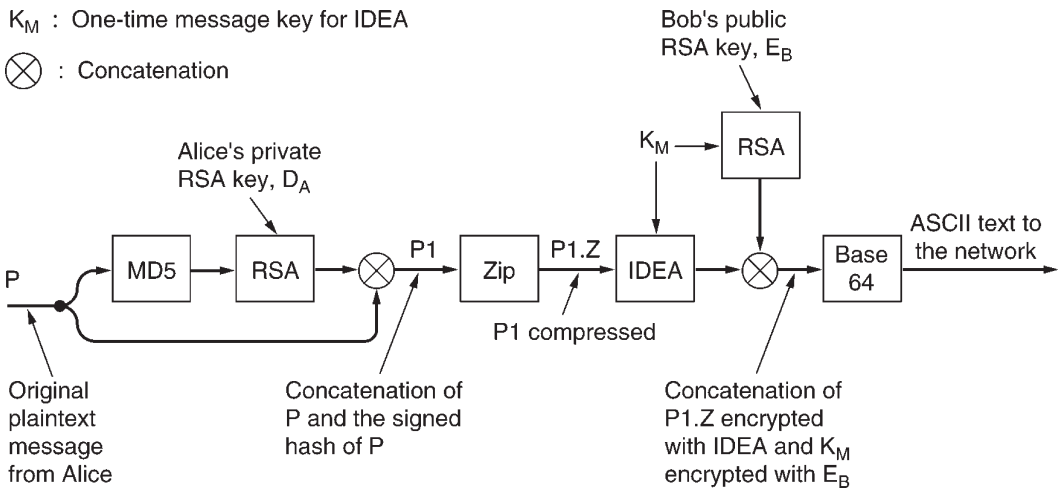
8.8.1 เทคนิค PGP-Pretty Good Privacy

ตัวอย่างแรกของการเข้ารหัสข้อมูลในอีเมลล์คือวิธีการเรียกว่า PGP (Pretty Good Privacy) ได้รับการพัฒนาขึ้นมาโดย Phill Zimmermann ในปี ค.ศ. 1991 วิธี PGP มีความสามารถในการรักษาความปลอดภัย การตรวจสอบผู้ใช้ ลายเซ็นอิเล็กทรอนิกส์ และการบีบอัดข้อมูล อยู่ในตัวเดียวกันในรูปแบบที่ง่ายต่อการใช้งาน ยิ่งกว่านั้นวิธีการนี้ฉบับสมบูรณ์รวมทั้งโปรแกรมต้นฉบับได้แจกจ่ายให้แก่ผู้สนใจทั่วไปฟรีผ่านทางระบบอินเตอร์เน็ต เนื่องจากเป็นวิธีการที่ดี ไม่มีต้นทุนในการจัดหามาใช้ (ฟรี) และสามารถใช้งานได้ทั้งในระบบ UNIX, Linux, Windows, และ MAC OS ทำให้กลายเป็นวิธีการที่นำมาใช้อย่างแพร่หลายในปัจจุบัน

PGP ทำการเข้ารหัสข้อมูลโดยการใช้ block cipher เรียกว่า IDEA (International Data Encryption Algorithm) ซึ่งใช้คีย์ขนาด 128 บิต เป็นวิธีที่ได้รับการพัฒนาขึ้นมาในประเทศสวิสเซอร์แลนด์ในช่วงเวลาเดียวกันกับที่ได้คิดค้นวิธีการแบบ DES แต่วิธี AES ยังไม่เกิดขึ้น โดยหลักการแล้ววิธี IDEA นั้นคล้ายคลึงกับวิธี DES และ AES คือทำการผสมบิตเป็นจำนวนหลายวงรอบการทำงาน แต่รายละเอียดของการผสมบิตนั้นแกต่างไปจาก DES และ AES การบริหารคีย์ใช้วิธี RSA และการตรวจสอบความถูกต้องของข้อมูลเป็นแบบ MD5 ซึ่งเป็นวิธีการที่ได้กล่าวถึงไปแล้ว

PGP เข้าไปเกี่ยวพันกับความยุ่งยากและการโต้เถียงนับตั้งแต่วันแรกที่ได้เปิดตัวออกมา เนื่องจาก Zimmerman ไม่ได้ห้ามคนอื่นๆ ในการนำวิธี PGP นี้เข้าไปเผยแพร่บนอินเตอร์เน็ตซึ่งเป็นที่ที่ใครก็ได้ในโลกจะสามารถค้นหาวิธีการนี้ไปใช้ รัฐบาลของสหรัฐอเมริกาได้กล่าวหาว่าเขาละเมิดกฎหมายการห้ามเผยแพร่ความลับทางทหารออกไปนอกประเทศจึงทำการสอบสวนเข้าเป็นระยะเวลานานกว่า 5 ปี แต่ก็ยกเลิกการสวบสวนไปในที่สุด

อีกปัญหาหนึ่งคือ PGP เข้าไปเกี่ยวข้องกับปัญหาการละเมิดสิทธิบัตร องค์กรที่เป็นเจ้าของสิทธิบัตร RSA คือ RSA Security Inc. ได้กล่าวหาว่า การใช้เทคนิค RSA ของ PGP นั้นเป็นการละเมิดสิทธิบัตรของ RSA แต่ปัญหาดังกล่าวก็ได้รับการตกลงยอมความกันในภายหลัง ยิ่งกว่านั้น PGP ยังได้ใช้



รูปที่ 8-44 กระบวนการส่งข่าวสารในระบบ PGP

สิทธิบัตรของอีกวิธีการหนึ่งคือ IDEA ซึ่งก็ได้ทำให้เกิดปัญหาขึ้นเช่นกัน

เนื่องจาก PGP เป็นวิธีการที่เปิดเผยต่อสาธารณะจึงทำให้มีผู้สนใจจำนวนมากทำการค้นคว้าและปรับปรุงวิธีการนี้กลายเป็น PGP รุ่นใหม่หลายรุ่น บางรุ่นได้รับการออกแบบให้สามารถหลีกเลี่ยงกฎหมายการห้ามเผยแพร่ความลับทางทหาร ในขณะที่รุ่นอื่นๆ ได้หลีกเลี่ยงการใช้อัลกอริทึมที่มีการจดสิทธิบัตรเอาไว้และก็มีอีกไม่น้อยที่พยายามจะเปลี่ยนให้วิธีการนี้กลายเป็นความลับเพื่อหวังผลทางการค้า แม้ว่ากฎหมายฯดังกล่าวจะได้รับการแก้ไขผ่อนปรนลงไปบ้างแล้ว และสิทธิบัตรของ RSA ก็ได้หมดอายุลงไปตั้งแต่เดือนกันยายน ค.ศ. 2000 แต่จากประวัติความเป็นมาทำให้เกิดเป็นวิธี PGP ขึ้นหลายรุ่นในหลายชื่อเรียก ในที่นี้จะกล่าวถึงรุ่นดั้งเดิมซึ่งเป็นรุ่นที่มีโครงสร้างง่ายที่สุด

PGP มีความตั้งใจที่จะใช้วิธีการเข้ารหัสเดิมที่มีอยู่แล้วแทนที่จะสร้างขึ้นมาใหม่และเลือกใช้อัลกอริทึมที่สามารถต้านทานความพยายามในการโจมตีได้ดีและยังได้รับการวิเคราะห์มาจากคนหลายกลุ่มแล้ว นอกจากนี้ต้องเป็นวิธีที่ไม่ได้อยู่ภายใต้อิทธิพลของรัฐบาลในการที่จะเข้ามาชี้แนะหรือทำให้กลายเป็นวิธีการที่อ่อนแอลงไป

PGP สนับสนุนการบีบอัดข้อมูล การรักษาความลับ การใช้ลายเซ็นต์อิเล็กทรอนิกส์ และยังสนับสนุนการบริหารจัดการคีย์ แต่ไม่ได้สนับสนุนสิ่งอำนวยความสะดวกสำหรับอีเมลล์ เป็นวิธีการที่ใช้การคำนวณล่วงหน้าที่ใช้ plaintext เป็น input และจัดการสร้าง ciphertext ที่มีการลงชื่อรับรองเป็น output ซึ่งจะถูกส่งไปทางอีเมลล์ต่อไป

รูป 8-44 แสดงตัวอย่างการทำงานของวิธี PGP ในที่นี้อลิสต้องการส่ง plaintext ที่มีการลงชื่อรับรอง (P) ไปยังบ๊อบด้วยวิธีการลับ ทั้งอลิสและบ๊อบมีคีย์ส่วนตัว DX และ คีย์สาธารณะ EX RSA keys สมมุติว่าแต่ละคนต่างก็รู้จักคีย์สาธารณะของอีกฝ่ายหนึ่งแล้ว

อลิสเริ่มต้นกระบวนการด้วยการเรียกใช้โปรแกรม PGP บนเครื่องคอมพิวเตอร์ของเธอเอง PGP จะทำการ hash ข้อความของอลิส (P) โดยใช้วิธี MD5 จากนั้นก็ทำการเข้ารหัสผลลัพธ์ hash ที่ได้โดยใช้คีย์ RSA ส่วนตัวของอลิส (DA) เมื่อบ๊อบได้รับข่าวสารนี้ก็จะสามารถถอดรหัส hash ออกมาได้โดยใช้คีย์สาธารณะของอลิสและทำการตรวจสอบความถูกต้องของข้อความนั้น ในขั้นตอนนี้แม้ว่าจะมีใครก็ตาม เช่น ทรูดี้จะสามารถดักรับข้อความและถอดรหัสโดยใช้คีย์สาธารณะของอลิสได้ก็ตาม ความเข็มแข็งของ MD5 จะช่วยรับประกันว่าจะไม่สามารถสร้างข้อความอื่นใดมาแล้วทำให้ได้ค่า hash ที่ตรงกับข้อความจริงนั้น

ข้อความ hash ที่ได้รับการเข้ารหัสแล้วและข้อความจริง (original message) จะถูกนำมาเรียงต่อกันเป็นข้อความเดียว (P1) และทำการบีบอัดด้วยโปรแกรม ZIP (Ziv-Lempel algorithm) เรียกผลที่ได้ในขั้นตอนนี้ว่า P1.Z

ขั้นต่อไปโปรแกรม PGP จะขอข้อมูลซึ่งเป็นเลขสุ่มจากอลิส ทั้งตัวเลขที่ป้อนเข้ามาและความเร็วในการพิมพ์จะถูกนำมาใช้ในการสร้าง IDEA message key ขนาด 128 บิต (KM) (ในวิธี PGP จะเรียกข้อมูลนี้ว่า session key) ต่อไป KM จะถูกนำมาใช้ในการเข้ารหัส P1.Z โดยใช้ IDEA ใน cipher feedback mode นอกจากนี้ KM ยังถูกนำไปเข้ารหัสด้วยคีย์สาธารณะของบ๊อบ (EB) ข้อมูลทั้งสอง

ส่วนนี้จะถูกนำมาเรียงต่อกันและเปลี่ยนไปอยู่ในรูปแบบ Base64 ข้อความสุดท้ายที่ได้รับจะประกอบด้วยตัวอักษร ตัวเลข และสัญลักษณ์ +, /, และ = ซึ่งสามารถที่จะใส่เข้าไปใน RFC 822 และถูกส่งต่อไปยังผู้รับโดยไม่มีการเปลี่ยนแปลงใดๆ เกิดขึ้น

เมื่อบ๊อบได้รับข้อความนี้ เขาก็จะกลับตัวเลข Base64 และทำการถอดรหัส IDEA key โดยใช้คีย์ RSA ส่วนตัวของเขาซึ่งก็จะได้ออกมาเป็น P1.Z จากนั้นทำการขยายข้อมูลออกแล้วแยก plaintext ออกจาก hash ที่ถูกเข้ารหัสออกจากกัน แล้วทำการถอดรหัส hash โดยใช้คีย์สาธารณะของอลิส ถ้า plaintext hash มีข้อความตรงกับข้อความที่ได้จากการคำนวณ MD5 ก็แสดงว่าข้อความ P ที่ได้รับนั้นถูกต้องและเป็นข้อความที่มาจากอลิสจริง

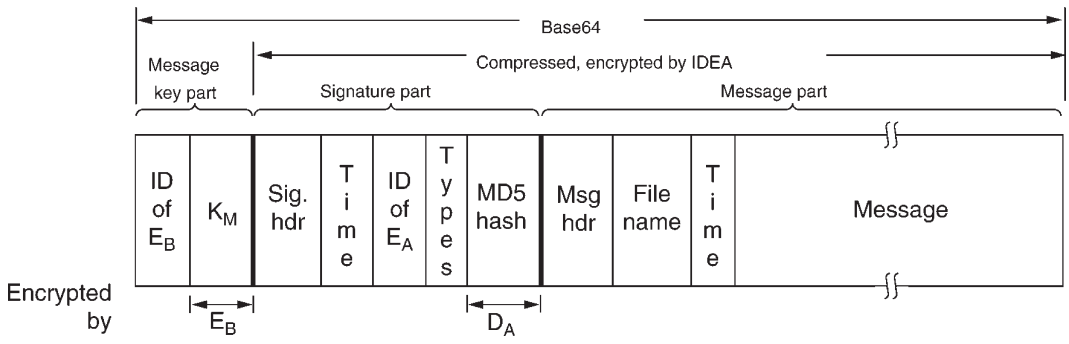
ในที่นี้ RSA ถูกนำมาใช้สองครั้ง ครั้งแรกในการเข้ารหัส MD5 hash ขนาด 128 บิต และครั้งที่สองใช้ในการเข้ารหัส IDEA key ขนาด 128 บิตเช่นกัน แม้ว่าวิธี RSA จะทำงานช้ามากแต่ก็ถูกนำมาใช้เข้ารหัสข้อมูลเพียง 256 บิตเท่านั้นไม่ใช่ข้อความทั้งข้อความ การเข้ารหัสข้อมูลปริมาณมากนั้นเป็นหน้าที่ของ IDEA ซึ่งมีความเร็วในการทำงานสูงกว่า RSA มาก ดังนั้น PGP จึงเป็นวิธีที่ให้ทั้งความปลอดภัย ทำการบีบอัดข้อมูล และมีลายเซ็นอิเล็กทรอนิกส์

PGP สนับสนุนการใช้คีย์ RSA ที่มีความยาว 4 ขนาด ขึ้นอยู่กับผู้ใช้จะเป็นผู้เลือกแบบที่คิดว่าเหมาะสมที่สุด นั่นคือ

1. Casual (384 บิต)-ในปัจจุบันสามารถถูกถอดรหัสได้โดยง่าย
2. Commercial (512 บิต)-แม้ว่าจะยุ่งยากขึ้นแต่ก็ยังสามารถถอดรหัสได้
3. Military (1024 บิต)-ไม่สามารถถูกถอดรหัสได้
4. Alien (2048 บิต)-ไม่สามารถถูกถอดรหัสได้เช่นกัน

เนื่องจาก RSA ถูกนำมาใช้งานกับข้อมูลขนาดสั้นมากเพียงสองครั้ง ดังนั้น ทุกคนจึงสามารถเลือกใช้ความปลอดภัยระดับ Alien ได้ตลอดเวลา

รูปแบบของ PGP แบบดั้งเดิมแสดงให้เห็นในรูป 8-45 รูปแบบอื่นๆ ก็มีใช้งานอย่างกว้างขวางเช่นกัน ข้อความถูกแบ่งออกเป็นสามส่วนประกอบด้วยส่วนที่เก็บ IDEA คีย์, ส่วนที่เก็บการลงชื่อรับรอง, และส่วนที่เก็บข้อความที่ต้องการส่ง ส่วนที่เก็บ IDEA คีย์นั้นยังเก็บ key identifier เอาไว้ด้วย เนื่องจากผู้ใช้ได้รับอนุญาตให้คีย์สาธารณะได้หลายตัว



รูปที่ 8-45 โครงสร้างข้อความ PGP

ส่วนที่เก็บการลงชื่อรับรองประกอบด้วย header ตามด้วยการลงเวลา (timestamp) identifier สำหรับคีย์สาธารณะของผู้ส่ง ที่จะถูกนำไปใช้ในการถอดรหัส signature hash ตามด้วยข้อมูลที่ระบุอัลกอริทึมที่นำมาใช้ (เผื่อไว้สำหรับ MD6 หรือ RSA2 เมื่อได้รับการคิดค้นขึ้นมาใช้งาน) และปิดท้ายด้วย hash ที่ถูกเข้ารหัสแล้ว

ส่วนที่เก็บข้อความที่ต้องการส่งมี header เป็นของตนเอง ตามด้วยชื่อสำรองของแฟ้มข้อมูลซึ่งจะนำมาใช้ในกรณีที่ผู้รับต้องการบันทึกข้อมูลนี้ลงในแฟ้มข้อมูลในดิสก์ ตามด้วยการลงบันทึกเวลาที่สร้างข้อความนี้ขึ้นมา และปิดท้ายด้วยข้อความที่ต้องการส่ง

การบริหารคีย์ (key management) ได้รับความสนใจเป็นอย่างมากสำหรับวิธี PGP ซึ่งทำงานดังนี้ ผู้ใช้แต่ละคนจะบริหารจัดการโครงสร้างข้อมูลสองแบบด้วยตนเอง คือ private key ring และ public key ring ส่วนที่เป็น private key ring ประกอบด้วยคีย์ส่วนตัวและคีย์สาธารณะที่เป็นคู่ของกันและกันอย่างน้อยหนึ่งคู่ เหตุผลที่สนับสนุนให้มีได้หลายคู่เนื่องจากเปิดโอกาสให้ผู้ใช้สามารถเปลี่ยนคีย์สาธารณะได้อยู่เสมอ หรือเมื่อคิดว่าคีย์ที่กำลังใช้อยู่นั้นถูกขโมยไปได้ ทั้งนี้โดยที่ไม่ต้องยกเลิกการส่งข้อความปัจจุบันที่อยู่ในขั้นตอนการเตรียมส่ง คีย์แต่ละคู่จะมี identity เฉพาะเพื่อใช้บอกให้ผู้รับทราบว่าใช้คีย์ตัวใดในการเข้ารหัส ส่วน message identifier จะประกอบด้วย 64 บิตแรก (low-order) ของคีย์สาธารณะ ผู้ใช้มีความรับผิดชอบในการหลีกเลี่ยงความขัดแย้งที่อาจเกิดขึ้นกับ identifier ของคีย์สาธารณะ คีย์ส่วนตัวที่เก็บอยู่ในดิสก์จะใช้ระหัสผ่านพิเศษที่มีความยาวมากในการปกป้องจากการแอบขโมย

Public key ring ประกอบด้วยคีย์สาธารณะของผู้ที่กำลังโต้ตอบอยู่กับผู้ใช้ คีย์สาธารณะนี้ถูกนำมาใช้ในการเข้ารหัส message key ที่เกี่ยวข้องกับแต่ละข้อความ ข้อมูลแต่ละตัวประกอบด้วยคีย์สาธารณะและ identifier ขนาด 64 บิต และตัวชี้บอกระดับความเชื่อมั่นของผู้ใช้ที่มีต่อคีย์แต่ละตัว

ปัญหาที่ทำให้วิธีการบริหารคีย์ถูกโจมตีเป็นดังนี้ สมมุติว่าคีย์สาธารณะได้รับการดูแลอยู่บนกระดานข่าวสาร (Bulletin boards) วิธีหนึ่งที่ทรูดี้สามารถใช้ในการอ่านอีเมลล์ลับของบ็อบคือการโจมตีที่กระดานข่าวแล้วเปลี่ยนคีย์สาธารณะของบ็อบด้วยคีย์อะไรก็ได้ที่ต้องการ ต่อมาเมื่ออลิสนำข้อมูลที่ถูกแก้ไขนี้ไปใช้ ทรูดี้ก็จะสามารถใช้วิธี the-man-in-the-middle attack กับบ็อบได้

เพื่อป้องกับการโจมตีที่อาจเกิดขึ้นนี้ หรือลดผลกระทบที่อาจเกิดขึ้นได้ อลิสจะต้องทราบระดับความเชื่อมั่นต่อคีย์ของบ็อบที่ปรากฏอยู่ใน public key ring ของเธอ ถ้าเธอทราบว่าบ็อบจัดการส่งดิสก์ที่มีคีย์อยู่ข้างในด้วยตนเอง เธอก็อาจกำหนดให้ระดับความเชื่อมั่นเป็นระดับสูงสุด การควบคุมการบริหารคีย์สาธารณะโดยผู้ใช้แต่ละคนนี้เองที่ทำให้วิธี PGP มีความแตกต่างจากวิธี PKI ซึ่งเป็นการควบคุมที่ศูนย์กลางแต่เพียงแห่งเดียว

อย่างไรก็ตาม คนส่วนหนึ่งได้ให้ความไว้วางใจกับคีย์เซิบเวอร์ในการร้องของคีย์สาธารณะของคนที่ต้องการจะติดต่อด้วย ด้วยเหตุผลนี้ทำให้ PGP สนับสนุนการใช้ทั้ง X.509 และการบริหารคีย์สาธารณะ public key ring ของ PGP เอง

8.8.2 เทคนิค PEM-Privacy Enhanced Mail

เทคนิค PEM (Privacy Enhanced Mail) ได้รับการพัฒนาขึ้นมาในช่วงปลายทศวรรษที่ 1980 ได้

รับการประกาศให้เป็นมาตรฐานในการส่งอีเมลล์ทางอินเตอร์เน็ต (RFC 1421-1424) โดยภาพรวมแล้ว PEM มีขีดความสามารถเหมือนกับ PGP คือ ในเรื่องความลับ และการตรวจสอบผู้ใช้เพื่อนำมาใช้ร่วมกับระบบอีเมลล์มาตรฐาน RFC 822

ข้อความที่ถูกส่งผ่านระบบ PEM จะเริ่มต้นด้วยการถูกเปลี่ยนให้ไปอยู่ในรูป canonical form เพื่อให้ทั้งข้อความให้ความหมายของ "ช่องว่าง" หรือ "white space" ตรงกัน จากนั้นข้อความจะถูกสร้าง hash ขึ้นมาด้วยวิธี MD2 หรือ MD5 ซึ่งจะนำไปเรียงต่อท้ายข้อความที่จะส่ง แล้วข้อความทั้งหมดจะถูกเข้ารหัสด้วยวิธี DES ข้อความที่ถูกเข้ารหัสแล้ว จะนำไปเข้ารหัสกับ Base64 แล้วส่งต่อไปให้ผู้รับ

ข้อความจะถูกเข้ารหัสด้วยคีย์ที่ใช้งานเพียงครั้งเดียวเหมือนกับที่ใช้ใน PGP ซึ่งคีย์ที่ใช้จะถูกส่งไปพร้อมกับข้อความที่เข้ารหัสแล้ว คีย์อาจได้รับการป้องกันด้วยวิธี RSA หรือ triple DES

การบริการคีย์นั้นมีลักษณะที่เป็นโครงสร้างที่ดีกว่า PGP คีย์จะได้รับการรับรองด้วย X.509 ที่ส่งมาโดย CA ซึ่งจะถูกนำมาใส่โครงสร้างแบบลำดับชั้นที่เริ่มต้นจาก root ข้อได้เปรียบของวิธีการนี้คือการยกเลิกการรับรองนั้นสามารถทำได้โดยให้ root ส่งข้อความ CRL ออกไปเป็นระยะ ๆ

ปัญหาเดียวที่มีอยู่ของวิธีการแบบ PEM ก็คือไม่มีใครเคยใช้วิธีนี้ในการปฏิบัติงานจริงซึ่งเป็นปัญหาทางด้านการเมืองมากกว่าอย่างอื่น นั่นคือใครจะเป็นผู้ควบคุมการทำงานของ root และในสภาวะอย่างไร มีผู้เสนอตัวเข้ามามากมายแต่ก็ไม่มีผู้ใดที่ต้องการมอบความเชื่อถือให้แก่องค์กรเพียงแห่งเดียวเข้ามาควบคุมรักษาความปลอดภัยให้แก่ระบบทั้งระบบ องค์กรที่น่าจะได้รับความไว้วางใจมากที่สุดคือ RSA Security Inc. ต้องการคิดค่าบริการต่อจำนวนครั้งที่มีการออกใบรับรองให้แก่ลูกค้า แต่องค์กรจำนวนหนึ่งก็ต่อต้านแนวความคิดนี้โดยเฉพาะรัฐบาลของสหรัฐอเมริกาที่ให้บริการการจดสิทธิบัตรฟรีและองค์กรต่าง ๆ ทั่วโลกโดยเฉพาะองค์กรทางการค้าก็มีความคุ้นเคยกับการใช้อัลกอริทึม RSA โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ ในที่สุดก็ไม่มีผู้ใดสามารถทำหน้าที่เป็น root ได้ ทำให้เทคนิค PEM ต้องถูกยกเลิกไปโดยปริยาย

8.8.3 เทคนิค S/MIME

องค์กร IETF ได้ให้การรับรองวิธีการเข้ารหัสข้อความในอีเมลล์แบบที่เรียกว่า S/MINE (Secure/MINE) ซึ่งได้กำหนดเป็นมาตรฐาน RFC 2632 ถึง 2643 วิธีการนี้ก็คล้ายกับ PEM คือให้บริการการตรวจสอบผู้ใช้ การตรวจสอบความถูกต้องของข้อมูล การรักษาความปลอดภัย และการปฏิเสธคำรับรอง อีกทั้งยังเป็นวิธีที่มีความอ่อนตัวสูง สนับสนุนอัลกอริทึมในการเข้ารหัสข้อมูลหลายแบบ ทำให้สามารถปกป้องข้อความได้ทุกชนิด และได้มีการพัฒนา header สำหรับ MINE ขึ้นมาหลายแบบเพื่อสนับสนุนงานด้านต่าง ๆ เช่น การลงชื่ออิเล็กทรอนิกส์ เป็นต้น

องค์กร IETF ได้เรียนรู้จากประสบการณ์ที่เกิดขึ้นกับ PEM นั่นคือ S/MINE ไม่มีโครงสร้างการรับรอง certificate อย่างเป็นลำดับชั้นที่เข้มงวด โดยอนุญาตให้ผู้ใช้สามารถให้ความไว้วางใจต่อ anchor ได้หลายแห่ง ตราบเท่าที่ใบรับรองถูกตรวจสอบย้อนกลับไปยัง anchor ที่ผู้ใช้ไว้ใจหรือผู้ใช้ให้ความเชื่อถือก็ถือว่าการรับรองนั้นใช้ได้

8.9 การรักษาความปลอดภัยบนเว็บ

การกล่าวถึงการรักษาความปลอดภัยในการสื่อสารและในอีเมลล์นั้นถือว่าเป็นจุดเริ่มต้นที่ดีของเรื่องที่มีความสำคัญมากกว่า นั่นคือการรักษาความปลอดภัยบนเว็บ (Web security) เว็บเป็นสถานที่ที่ผู้ร้ายอย่างเช่นทรูดี้ทำงานอยู่ ในหัวข้อต่อไปนี้จะได้กล่าวถึงปัญหาบางส่วนที่เกี่ยวข้องกับการรักษาความปลอดภัยบนเว็บ

การรักษาความปลอดภัยบนเว็บสามารถแบ่งออกได้เป็นสามส่วน ส่วนแรก จะตั้งชื่ออ๊อบเจ็คและทรัพยากรต่างๆ ให้มีความปลอดภัยได้อย่างไร ส่วนที่สอง จะสามารถจัดตั้งการเชื่อมต่อที่ปลอดภัยและสามารถตรวจสอบผู้ใช้ได้อย่างไร และส่วนที่สาม จะทำอย่างไรเมื่อเว็บไซท์ส่งโปรแกรมที่สามารถประมวลผลได้ไปยังผู้ใช้

8.9.1 ภัยคุกคาม

ลองพิจารณาถึงเรื่องต่างๆ ที่ได้เกิดขึ้นแล้ว ประการแรก โฮมเพจขององค์กรจำนวนมากได้ถูกโจมตีและถูกแทนที่ด้วยโฮมเพจแห่งใหม่ที่ผู้โจมตีเลือกให้ (คำที่สื่อสารมวลชนเรียกพวกโจมตีประเภทนี้คือ hackers แต่ในขณะเดียวกันสำหรับโปรแกรมเมอร์มืออาชีพแล้วจะใช้คำนี้ในการเรียกโปรแกรมเมอร์ที่มีความสามารถสูงกว่าโปรแกรมเมอร์ทั่วไป ในที่นี้จึงใช้คำเรียกอย่างเป็นกลางว่า crackers) เว็บไซต์ยอดนิยมทั้งหลายต่างก็ถูกพวก cracker นี้โจมตีมาแล้วทั้งนั้น เช่น เว็บไซต์ของ Yahoo, CIA, NASA และอื่นๆ ส่วนมากพวก cracker จะใส่ถ้อยคำหรือภาพตลกเข้าไปในเว็บไซต์เหล่านี้ซึ่งก็จะถูกแก้ไขได้ในไม่กี่ชั่วโมง

ต่อไปลองพิจารณากรณีที่ส่งผลร้ายแรงกว่านี้ เว็บไซต์จำนวนมากถูกทำลายลงด้วยการโจมตีแบบ Denial-of-service attack ซึ่ง cracker จะโจมตีด้วยการส่งข่าวสารจำนวนมหาศาลมายังเว็บไซต์นั้นทำให้ไม่สามารถให้บริการได้ตามปกติอีกต่อไปและอาจร้ายแรงถึงขั้นต้องระงับการให้บริการไปชั่วขณะ โดยทั่วไปการโจมตีจะเกิดขึ้นจากการรุมส่งข่าวสารมาจากเว็บไซต์จำนวนมากที่ cracker ได้เข้าไปยึดไว้ได้แล้ว (DDoS attack) การโจมตีชนิดนี้มีมากจนกระทั่งถือเป็นเรื่องธรรมดาจนไม่ปรากฏเป็นข่าวอีกต่อไป การโจมตีแต่ละครั้งทำให้เว็บไซท์ที่ถูกโจมตีนั้นสูญเสียเงินไปเป็นจำนวนมาก

ในปี ค.ศ. 1999 cracker ชาวสวีเดนได้บุกเข้าไปในเว็บไซต์ Microsoft's Hotmail และได้เข้าไปสร้าง mirror site ที่อนุญาตให้ผู้ใดก็ได้สามารถพิมพ์ชื่อของผู้ใช้ Hotmail แล้วสามารถเข้าไปอ่านข้อความอีเมลล์ของผู้นั้นได้ราวกับเป็นเจ้าของเสียเอง

ในอีกกรณีหนึ่ง cracker ที่เป็นเด็กอายุ 19 ปีชาวรัสเซียชื่อว่า Maxim ได้บุกเข้าไปในเว็บไซต์ e-commerce แห่งหนึ่งและขโมยหมายเลขเครดิตการ์ดไปกว่า 300,000 ใบ จากนั้นจึงได้เข้าไปในเว็บไซต์ของเจ้าของเว็บไซต์ดังกล่าวเพื่อเรียกร้องเงินเป็นจำนวน 100,000 เหรียญเป็นค่าไถ่สำหรับเครดิตการ์ดที่เขาขโมยไปได้ เจ้าของเว็บไซต์ปฏิเสธที่จะจ่ายเงินให้ทำให้ Maxim โกรธและนำหมายเลขเครดิตการ์ดทั้ง 300,000 ใบออกประกาศทั่วอินเตอร์เน็ตทำให้ผู้เคราะห์ร้ายคือเจ้าของบัตรเครดิตต้องสูญเสียเงินไปเป็นจำนวนมาก

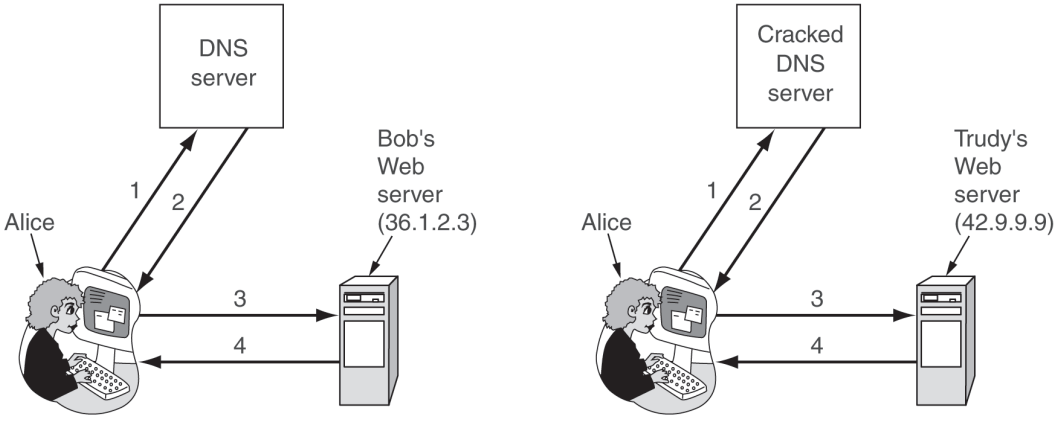
ในอีกเหตุการณ์หนึ่ง นักศึกษาหนุ่มชาวแคลิฟอเนียอายุ 23 ปีได้ส่งอีเมลล์ที่เป็นการแถลงข่าวไป

ยังสำนักข่าวแห่งหนึ่งกล่าวว่า Emulex Corporation กำลังจะนำเสนอรายงานการขาดทุนการประกอบการในไตรมาศที่สามเป็นเงินจำนวนมากซึ่งมีผลทำให้ประธานกรรมการบริหารต้องลาออกในทันที ภายในไม่กี่ชั่วโมงหลังจากนั้นหุ้นของบริษัทได้ตกลงกว่า 60% ทำให้ผู้ถือหุ้นต้องขาดทุนไปมากกว่าสองพันล้านเหรียญสหรัฐ ส่วนผู้ส่งข่าวเองได้จัดการขายหุ้นไปได้เงินมากว่า 250,000 เหรียญก่อนที่จะทำการประกาศข่าวนั้น ในกรณีนี้ไม่เกี่ยวข้องกับการบุกรุกเข้าไปในเว็บไซต์ แต่ก็ได้แสดงให้เห็นว่าการประกาศในลักษณะนี้ในโฮมเพจของกิจการขนาดใหญ่อาจทำให้เกิดผลเสียหายได้เช่นกัน

8.9.2 การตั้งชื่ออย่างปลอดภัย

เริ่มต้นด้วยการทำงานที่ง่ายที่สุด อลิสต้องการเข้าไปยังเว็บไซต์ของบ๊อบ เธอจึงพิมพ์ชื่อ URL ของบ๊อบเข้าไปในบราวเซอร์ซึ่งต่อมาไม่นานเว็บเพจของบ๊อบก็จะปรากฏขึ้น แต่ปัญหาก็คือเป็นเว็บเพจของบ๊อบหรือเปล่า หรือว่าเป็นแผนการชั่วร้ายของทรูดี้อีกหรือเปล่า ทรูดี้อาจจะดักจับข่าวสารทั้งหมดที่ส่งออกมาจากอลิสและทำการตรวจสอบดู ถ้าพบว่าเป็น HTTP GET request ที่มีเป้าหมายไปยังเว็บไซต์ของบ๊อบ ทรูดี้อาจจะไปที่เว็บไซต์ของบ๊อบทำการคัดลอกสำเนาทุกอย่างและแก้ไขสิ่งที่ต้องการจากนั้นจึงส่งกลับมาให้อลิส อลิสจะไม่มีทางทราบได้เลย ยิ่งกว่านั้นทรูดี้อาจจะแก้ไขราคาสินค้าในร้าน e-store ของบ๊อบเพื่อให้ดูน่าสนใจยิ่งขึ้นไปอีก จากนั้นก็อาจจะหลอกให้อลิสสั่งซื้อสินค้าในร้านของบ๊อบ (ตัวปลอม) เพื่อให้อลิสส่งหมายเลขบัตรเครดิตของเธอมาให้

สิ่งหนึ่งที่เป็นจุดด้อยของวิธีการ man-in-the-middle attack ในลักษณะนี้ก็คือ ทรูดี้จะต้องอยู่ในตำแหน่งที่จะสามารถดักจับข่าวสารที่ส่งออกไปจากอลิสได้และส่งข่าวสารปลอมกลับมาให้เธอ ในทางปฏิบัติ ทรูดี้จะต้องสามารถเชื่อมต่อ (tap) สายโทรศัพท์ของอลิสหรือของบ๊อบได้ซึ่งถ้าเป็นสายใยแก้วนำแสงแล้วก็เป็นเรื่องที่ทำได้ยากมาก แม้ว่าทรูดี้จะเป็นคนฉลาดแต่ก็ขี้เกียจที่จะไปทำการเชื่อมสายด้วยตนเอง ดังนั้นเธอจึงหาวิธีการอื่นที่ง่ายกว่านี้



รูปที่ 8-46 (a) เหตุการณ์ปกติ (b) เมื่อทรูดี้สามารถบุกรุกเข้าไปใน DNS server ได้

การหลอกว่าเป็น DNS

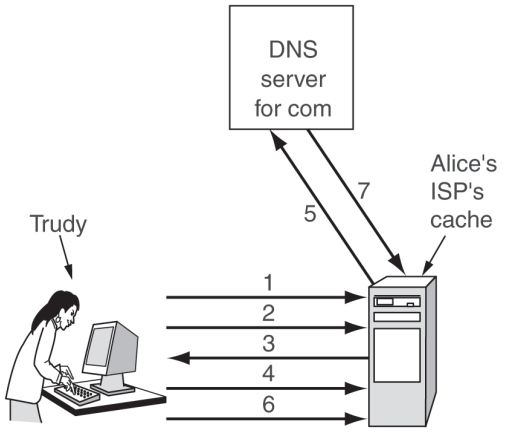
ตัวอย่างเช่น สมมุติว่าทรูดี้สามารถบุกเข้าไปในระบบ DNS ได้ซึ่งบางครั้งอาจเป็นเพียง DNS cache ของ ISP ของอลิสและจัดการแทนที่หมายเลข IP ของบ๊อบ (สมมุติว่าเป็น 36.1.2.3) ด้วยหมายเลข IP ของเธอ (เช่น 42.9.9.9) การกระทำเช่นนี้ทำให้ทรูดี้สามารถทำในสิ่งต่อไปนี้ได้ ดังแสดงในรูป 8-46(a) นั่นคือ (1) อลิสร้องขอหมายเลข IP ของบ๊อบไปยัง DNS (2) ได้รับหมายเลข IP ที่ต้องการ (3) ติดต่อไปยังโฮมเพจของบ๊อบ และ (4) ได้รับโฮมเพจของบ๊อบตามต้องการ ภายหลังจากที่ทรูดี้ได้จัดการแก้ไขข้อมูลของบ๊อบใน DNS แล้วด้วยการใส่หมายเลข IP ของเธอลงไปแทนที่ IP ของบ๊อบ จะเกิดเป็นเหตุการณ์ดังแสดงในรูป 8-46(b) ในที่นี้เมื่ออลิสมองหาหมายเลข IP ของบ๊อบ เธอจะได้รับหมายเลข IP ของทรูดี้มาแทน ทำให้ข่าวสารทั้งหมดที่อลิสต้องการส่งไปยังบ๊อบจะถูกส่งไปที่ทรูดี้ทั้งหมด จากนี้ไปทรูดี้สามารถทำตัวเป็น man-in-the-middle attack โดยที่ไม่ต้องไปจัดการเชื่อมต่อสายให้ยุ่งยากเลย ด้วยการบุกเข้าไปใน DNS server และจัดการแก้ไขข้อมูลเพียงระเบียนเดียวเท่านั้น

ทรูดี้จะหลอก DNS server ได้อย่างไร กลับกลายเป็นเรื่องที่ง่ายอย่างคาดไม่ถึงเลยทีเดียว ในที่นี้ทรูดี้ต้องการหลอก DNS server ที่ ISP ของอลิสให้เปลี่ยนหมายเลข IP ของบ๊อบให้เป็นหมายเลขอื่นตามที่ต้องการ โชคไม่ดีนักเนื่องจาก DNS ใช้ UDP ทำให้ DNS server ไม่มีทางที่จะตรวจสอบว่าใครเป็นผู้ให้คำตอบ (สำหรับคำถามของอลิส) ทรูดี้สามารถใช้ประโยชน์จากข้อเท็จจริงนี้ได้โดยการหลอกด้วยการใส่หมายเลข IP ปลอมเข้าไปใน DNS cache เพื่อความง่ายจึงสมมุติว่า ISP ของอลิสไม่มีข้อมูลเว็บไซต์ของบ๊อบ (bob.com) อยู่ตั้งแต่แรก แต่ถ้ามีอยู่ ทรูดี้ก็จะต้องรอจนกว่าข้อมูลนี้จะหมดอายุการใช้งานแล้วลองพยายามใหม่ (หรือใช้เทคนิคอื่น)

ทรูดี้เริ่มต้นการโจมตีด้วยการส่งคำถามไปยัง ISP ของอลิสเพื่อถามหาเว็บไซต์ของบ๊อบ แต่เนื่องจากไม่มีข้อมูลนี้อยู่ใน DNS ทำให้ cache server จำเป็นต้องถามหาข้อมูลนี้มาจากเครื่องเซิบเวอร์ที่อยู่ระดับบนขึ้นไป (ในที่นี้คือ ".com" domain) ในตอนนี้เองที่ทรูดี้รีบชิงการส่งคำตอบมายัง DNS server ด้วยคำตอบปลอม เช่น "bob.com is 42.9.9.9" ซึ่งเป็นชื่อเว็บไซต์ของบ๊อบแต่เป็นหมายเลข IP ของเธอ ถ้าคำตอบปลอมของทรูดี้ถูกส่งไปถึง ISP ของอลิสก่อนคำตอบนี้ก็จะถูกใส่ไว้ใน cache และคำตอบที่แท้จริงซึ่งมาถึงทีหลังก็จะถูกลบทิ้งไป (ในฐานะที่เป็นคำตอบสำหรับคำถามที่ได้รับคำตอบไปแล้ว หรือคำตอบที่ไม่มีใครถาม) การหลอกลวง DNS server ในลักษณะนี้เรียกว่า DNS Spoofing ส่วน cache ที่บันทึกข้อมูลหลอกลวงเอาไว้เรียกว่า poisoned cache

อันที่จริงแล้ว เหตุการณ์ไม่ได้ง่ายอย่างที่ได้อธิบายไว้ ประการแรก ISP ของอลิสจะตรวจสอบดูว่าคำตอบที่ได้รับนั้นมีหมายเลข IP ของ source address ตรงกับหมายเลข IP ของเซิบเวอร์ที่อยู่ระดับเหนือขึ้นไปหรือไม่ แต่เนื่องจากทรูดี้สามารถใส่ข้อมูลอะไรก็ได้เข้าไปใน source address เธอจึงสามารถผ่านการตรวจสอบในขั้นตอนนี้ไปได้โดยง่าย

ประการที่สอง เพื่อที่จะให้ DNS server สามารถบอกได้ว่าคำตอบใดนั้นคู่กับคำถามใด คำถามทุกคำถามจะต้องมีหมายเลขลำดับ ในการหลอก ISP ของอลิสทรูดี้จะต้องทราบหมายเลขลำดับที่



1. Look up foobar.trudy-the-intruder.com
   (to force it into the ISP's cache)
2. Look up www.trudy-the-intruder.com
   (to get the ISP's next sequence number)
3. Request for www.trudy-the-intruder.com
   (Carrying the ISP's next sequence number, n)
4. Quick like a bunny, look up bob.com
   (to force the ISP to query the com server in step 5)
5. Legitimate query for bob.com with seq = n+1
6. Trudy's forged answer: Bob is 42.9.9.9, seq = n+1
7. Real answer (rejected, too late)

รูปที่ 8-47
วิธีการที่ทรูดี้หลอก ISP ของอลิส

กำลังใช้งานอยู่ในปัจจุบัน วิธีการที่ง่ายที่สุดในการเรียนรู้หมายเลขลำดับนี้ก็คือทรูดี้จะต้องทำการลงทะเบียนใน domain ด้วยตัวเอง เช่น "trudy-the-intruder.com" สมมุติว่าใช้หมายเลข IP เดียวกันคือ 42.9.9.9 และเธอจะต้องสร้าง DNS server สำหรับ domain ที่พึ่งลงทะเบียนใหม่คือ "dns.trudy-the-intruder.com" ซึ่งก็ใช้หมายเลข IP ของทรูดี้ด้วย (ทรูดี้อาจใช้คอมพิวเตอร์เพียงเครื่องเดียว) ขั้นต่อไปเธอจะต้องทำให้ ISP ของอลิสรู้จัก DNS server ของเธอซึ่งก็เป็นเรื่องที่ง่ายมากคือการส่งคำถามไปยัง ISP ของอลิสเพื่อหาหมายเลข IP ของ "foobar.trudy-the-intruder.com" ซึ่งจะทำให้ ISP ของอลิสต้องส่งคำถามไปยัง ".com" server

เมื่อ "dns.trudy-the-intruder.com" อยู่ใน cache ของ ISP ของอลิสเรียบร้อยแล้ว การโจมตีที่แท้จริงก็จะเริ่มขึ้น ทรูดี้จะส่งคำถามไปยัง ISP ของอลิสเพื่อถามหา "www.trudy-the-intruder.com" ซึ่ง ISP ก็จะส่งคำถามต่อมายัง DNS server ของทรูดี้ตามปกติแล้ว คำถามนี้จะมีหมายเลขลำดับที่ทรูดี้กำลังมองหาอยู่ ในทันทีนั้นทรูดี้ก็จะให้ ISP ของอลิสหาข้อมูลของบ๊อบ ซึ่งเธอก็จะจัดการตอบคำถามของตัวเธอเองด้วยการส่งคำตอบปลอมกลับไปยัง ISP ของอลิสโดยการอ้างว่าเป็นคำตอบที่มาจาก ".com" server โดยระบุว่า "bob.bom is 42.9.9.9" คำตอบปลอมนี้จะใช้หมายเลขลำดับที่มีค่ามากกว่าหมายเลขที่คำถามของเธออยู่ 1 ในจังหวะเดียวกันนี้ทรูดี้สามารถส่งคำตอบปลอมๆ ออกมาได้มากมายโดยเพิ่มค่าของหมายเลขลำดับขึ้นมาทีละ 1 เสมอ หนึ่งในจำนวนนี้คงจะเป็นหมายเลขที่ถูกต้องอย่างแน่นอน ส่วนที่เหลือก็จะถูกเครื่องเซิบเวอร์ลบทิ้งไป เมื่อคำตอบปลอมเดินทางมาถึง ISP ของอลิสก็จะถูกเก็บไว้ใน cache เมื่อคำตอบที่แท้จริงเดินทางมาถึงก็จะถูกลบทิ้งไป (เพราะไม่มีคำถามค้างอยู่)

ทีนี่เมื่ออลิสมองหาหมายเลข IP ของบ๊อบ "bob.com" เธอก็จะได้รับคำตอบให้ไปใช้หมายเลข 42.9.9.9 ซึ่งเป็นที่อยู่ของทรูดี้ ทรูดี้ก็จะประสบความสำเร็จในการทำตัวเป็น man-in-the-middle attack ได้จากสถานที่ที่เธอนั่งอยู่ในขณะนั้น ขั้นตอนต่างๆ ของการโจมตีด้วยวิธีการนี้แสดงไว้ในรูป 8-47 โปรดเข้าใจไว้ว่านี่ไม่ใช่วิธีการเดียวที่จะหลอก DNS ได้เพราะยังมีอีกมากมายหลายวิธี

DNS ที่ปลอดภัย

การโจมตีด้วยวิธีการที่กล่าวถึงนี้สามารถป้องกันได้โดยการให้ DNS server ใช้ตัวเลขแบบสุ่ม (random ID) ในคำถามที่ส่งไปยัง ".com" แทนที่จะใช้หมายเลขลำดับที่เรียงต่อกัน แต่ก็ดูเสมือนว่า

เมื่อได้ทำการอุดรูรั่วหนึ่งไปได้แล้วก็จะมีรูรั่วใหม่เกิดขึ้น ปัญหาที่แท้จริงนั้นคือ DNS ได้ถูกออกแบบมาในระยะเวลาที่ระบบอินเตอร์เน็ตเป็นเพียงระบบทดลองที่มีเพียงผู้ใช้เป็นเพียงนักศึกษาในมหาวิทยาลัยไม่กี่ร้อยแห่งซึ่งผู้ใช้ทั่วไป เช่น อลิส บ็อบ และทรูดี้ยังไม่มีสิทธิเข้าไปใช้งานได้เลย ในช่วงเวลานั้นการรักษาความปลอดภัยยังเป็นเรื่องที่ไม่มีความจำเป็น ความสำคัญเร่งด่วนในเวลานั้นคือการทำให้ระบบอินเตอร์เน็ตสามารถทำงานได้ สิ่งแวดล้อมได้เปลี่ยนแปลงไปเป็นอย่างมากในช่วงเวลาหลายปีที่ผ่านมา ดังนั้นในปี ค.ศ. 1994 องค์กร IETF จึงได้จัดตั้งคณะทำงานเพื่อค้นหาวิธีที่จะทำให้ DNS มีความปลอดภัย โครงการนี้มีชื่อว่า DNSsec (DNS security) ซึ่งได้ผลลัทธ์ออกมาเป็นมาตรฐาน RFC 2535 อย่างไรก็ตาม DNSsec ยังไม่ได้รับการนำไปใช้งานอย่างเต็มที่ทำให้ DNS จำนวนหนึ่งยังคงเปราะบางต่อการโจมตี

DNSsec นั้นเป็นแนวทางที่ง่ายมาก กล่าวคือเป็นวิธีการที่นำการเข้ารหัสแบบคีย์สาธารณะมาใช้ ในทุกพื้นที่การให้บริการของ DNS (DNS zone) จะมีคีย์สาธารณะร่วมกับคีย์ส่วนตัวใช้งานอยู่ ข่าวสารทั้งหมดที่ถูกส่งไปจาก DNS server จะถูกลงชื่อด้วยการใช้คีย์ส่วนตัวของเขตการให้บริการ (zone) ที่เป็นเจ้าของคำถามทำให้ผู้รับสามารถตรวจสอบเจ้าของคำถามนั้นได้

DNSsec ได้นำเสนอการให้บริการพื้นฐานสามอย่างคือ

1. การพิสูจน์ว่าข้อมูลนั้นถูกส่งมาจากผู้ใด
2. การแจกจ่ายคีย์สาธารณะ
3. การตรวจสอบรายการทำงานและคำร้องขอ

บริการหลักที่มีให้คือบริการอันแรกซึ่งทำการตรวจสอบว่าข้อมูลที่ถูกส่งกลับไปนั้นได้รับการรับรองโดย DNS เจ้าของพื้นที่การให้บริการนั้น บริการลำดับที่สอง มีไว้เพื่อการจัดเก็บและการให้ได้มาซึ่งคีย์สาธารณะอย่างปลอดภัย ส่วนบริการที่สามนั้นมีความจำเป็นในการป้องกันต่อการโจมตีแบบ playback attack และ spoofing attack สังเกตุว่าการรักษาความลับไม่ได้ถูกจัดให้เป็นหนึ่งในบริการที่นำเสนอ เนื่องจากข่าวสารที่อยู่ใน DNS นั้นถือว่าเป็นข่าวสารสาธารณะ เนื่องจากขั้นตอนในการนำ DNSsec มาใช้งานนั้นคาดว่าจะใช้เวลาหลายปี ความสามารถของเซิบเวอร์ที่คำนึงถึงเรื่องความปลอดภัยที่จะทำงานร่วมกับเซิบเวอร์ที่ไม่คำนึงถึงเรื่องความปลอดภัยนั้นจึงเป็นเรื่องที่สำคัญมากซึ่งสามารถกล่าวได้ว่าโพรโตคอลที่นำมาใช้นั้นจะต้องไม่มีการเปลี่ยนแปลง

ข้อมูลใน DNS จะถูกจัดรวมเป็นกลุ่มเรียกว่า RRSets (Resource Record Sets) โดยให้ทุกระเบียนที่มีชื่อเรียก ชื่อคลาส และชนิดอย่างเดียวกันจะถูกนำมารวมอยู่ในเซ็ทเดียวกัน ตัวอย่างเช่น ใน RRSet อาจมีระเบียน A หลายระเบียนถ้าชื่อ DNS นั้นโยงไปหาหมายเลข IP เดียวกันและอาจจะมีหมายเลข IP ลำดับที่สองซึ่งก็ต้องเหมือนกันด้วย RRSet ถูกเพิ่มเติมด้วยระเบียนชนิดใหม่หลายชนิด (ดังจะอธิบายในลำดับต่อไป) ข้อมูลแต่ละเซ็ทใน RRSet จะถูกเข้ารหัสด้วยวิธีการ hash เช่น SHA-1 หรือ MD5 ค่าขอ hash จะถูกลงชื่อกำกับด้วยคีย์ส่วนตัวของผู้ให้บริการในเขตนั้น (เช่นการใช้ RSA) ข้อมูลส่วนที่จะส่งไปยังผู้ใช้นั้นคือส่วน RRSet ที่ได้รับการลงชื่อกำกับแล้ว เมื่อผู้ใช้ได้รับข้อความนี้ก็จะทำการตรวจสอบว่าเป็นข้อความที่ถูกลงชื่อกำกับโดยใช้คีย์ส่วนตัวของผู้ให้บริการเขตนั้นๆ (Zone) ถ้าข้อมูลตรงกันข้อความที่ถูกส่งมาก็จะถูกนำไปใช้งานต่อไป เนื่องจากแต่ละ RRSet จะมีการลงชื่อกำกับในแบบของตนเองทำให้ RRSet สามารถถูกใส่ไว้ใน cache ที่ใดก็ได้ แม้แต่ภายใน cache ของผู้ใช้ที่

Domain name	Time to live	Class	Type	Value
bob.com.	86400	IN	A	36.1.2.3
bob.com.	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com.	86400	IN	SIG	86947503A8B848F5272E53930C...

รูปที่ 8-48
ตัวอย่าง RRSet
ของ "bob.com"

ไม่สามารถไว้วางใจได้โดยไม่ทำอันตรายต่อการรักษาความปลอดภัยเลย

DNSsec ได้นำชนิดของข้อมูลแบบใหม่มาใช้หลายชนิด ชนิดแรกคือ KEY record ซึ่งใช้ในการบันทึกคีย์สาธารณะของเขตการให้บริการ ผู้ใช้ โฮส อัลกอริทึมที่ใช้ในการเข้ารหัสข้อมูล โพรโตคอลที่ใช้ในการส่งข้อมูล และบิตควบคุมอีกจำนวนหนึ่ง คีย์สาธารณะจะถูกเก็บไว้ในสภาพ plaintext ใบรับรอง X.509 ไม่ได้ถูกนำมาใช้เนื่องจากขนาดอันใหญ่โตของมัน เขตข้อมูลอัลกอริทึมจะแสดงค่า "1" สำหรับวิธี MD5/RSA เขตข้อมูลโพรโตคอลจะแสดงว่าใช้ IPsec หรือโพรโตคอลแบบอื่น

ชนิดของระเบียนแบบที่สองคือ SIG record ซึ่งจะจัดเก็บค่า hash ที่ถูกลงชื่อกำกับด้วยอัลกอริทึมที่ระบุไว้ใน KEY record และยังทำการบันทึกเวลาที่เริ่มการรับรองและเวลาที่หยุดการรับรอง รวมทั้งชื่อของผู้ที่ลงชื่อรับรองและข้อมูลอื่นอีกบางส่วน

การออกแบบ DNSsec ช่วยให้คีย์ส่วนตัวของเขตผู้ให้บริการสามารถเก็บไว้ในลักษณะ off-line ได้ นั่นคือสิ่งที่เก็บอยู่ในฐานข้อมูลของเขตผู้ให้บริการ (Zone) จะถูกนำไปส่งด้วยมือ (ผ่าน CD-ROM) เพื่อทำการปรับปรุงฐานข้อมูลของเครื่องคอมพิวเตอร์ที่ไม่ได้เชื่อมต่ออยู่และเป็นที่เก็บคีย์ส่วนตัว ประมาณหนึ่งหรือสองครั้งต่อวัน RRSet จะได้รับการลงชื่อกำกับได้ที่นั่นและทำการสร้าง SIG record ขึ้นมาด้วย ซึ่งทั้งหมดนี้จะถูกนำกลับไปยังเครื่องเซิบเวอร์ของเขตผู้ให้บริการผ่านทางแผ่น CD-ROM ได้อีกเช่นกัน ภายหลังจากการลงชื่อกำกับเรียบร้อยแล้วคีย์ทั้งหมดจะถูกลบไปจากหน่วยความจำและแผ่นดิสก์และ CD-ROM ทั้งหมดก็จะถูกนำไปเก็บไว้ในที่ที่ปลอดภัย ขั้นตอนนี้เป็นการลดการรักษาความปลอดภัยทางอิเล็กทรอนิกส์ลงไปโดยแทนที่ด้วยมาตรการรักษาความปลอดภัยทางกายภาพที่คุ้นเคย

วิธีการลงชื่อกำกับใน RRSet ล่วงหน้าช่วยเพิ่มความเร็วในกระบวนการการตอบคำถามขึ้นได้เป็นอย่างมาก เนื่องจากไม่มีความจำเป็นจะต้องมีการถอดรหัสข้อมูลเกิดขึ้นเลย แต่สิ่งที่ต้องแลกเปลี่ยนก็คือจะต้องเสียเนื้อที่ในดิสก์เป็นจำนวนมากในการจัดเก็บคีย์ทั้งหมดและการลงชื่อกำกับไว้ในฐานข้อมูลของ DNS ข้อมูลบางส่วนอาจมีขนาดใหญ่ขึ้นสิบเท่าเนื่องจากต้องมีการลงชื่อรับรอง

เมื่อโปรเซสของผู้ใช้ ได้รับ RRSet ที่ได้รับการลงชื่อกำกับ ก็จะต้องใช้คีย์สาธารณะของเจ้าของพื้นที่ ให้บริการนั้นในการถอดรหัส hash จากนั้นทำการคำนวณค่า hash ด้วยตัวเองและเปรียบเทียบค่าของ hash ทั้งสองซึ่งจะต้องตรงกัน จึงจะทำให้เชื่อถือข้อมูลที่ส่งมาได้ อย่างไรก็ตามวิธีการนี้ได้ทำให้เกิดปัญหาขึ้นมาว่าผู้ใช้จะสามารถได้รับคีย์สาธารณะของเจ้าของเขตพื้นที่ให้บริการได้อย่างไร ซึ่งวิธีการหนึ่ง ที่นำมาแก้ปัญหานี้คือการสอบถามจากเซิบเวอร์ที่ไว้ใจได้ ผ่านช่องสื่อสารที่ปลอดภัย เช่นการใช้ IPsec

อย่างไรก็ตาม ในทางปฏิบัติแล้วเป็นที่คาดหวังว่าผู้ใช้จะได้รับทราบล่วงหน้าถึงค่าคีย์สาธารณะของ domain ในระดับบนทั้งหมด ถ้าอลิสต้องการเข้าไปยังเว็บไซต์ของบ็อบเธอก็จะสามารถถาม DNS

เพื่อให้ได้ค่า RRSet ของ "bob.com" ซึ่งจะบรรจุหมายเลข IP ของเว็บไซต์นี้และ KEY record ที่บรรจุคีย์สาธารณะของบ็อบ RRSet จะถูกลงชื่อกำกับโดย ".com" domain ทำให้อลิสสามารถตรวจสอบความถูกต้องได้โดยง่าย รูป 8-48 แสดงตัวอย่างของ RRSet

เมื่อมีการปกป้องด้วยข้อมูลที่สามารถตรวจสอบได้ด้วยคีย์สาธารณะของบ็อบ อลิสสามารถที่จะขอหมายเลข IP ของ www.bob.com จาก DNS server ของบ็อบได้ RRSet จะถูกลงชื่อกำกับโดยใช้คีย์ส่วนตัวของบ็อบ ทำให้อลิสสามารถตรวจสอบการลงชื่อกำกับใน RRSet ที่บ็อบส่งกลับมาได้ ถ้าทรูดี้สามารถที่จะส่ง RRSet ปลอมเข้ามาแทนที่ไว้ใน cache (เหมือนดังที่ได้กล่าวในข้างต้น) อลิส ก็จะสามารถทราบได้ในทันทีเพราะค่าใน SIG record จะเป็นค่าที่ไม่ถูกต้อง

อย่างไรก็ตาม DNSsec ยังสนับสนุนกลไกในการเข้ารหัสข้อมูลเพื่อให้ในการสร้างคำตอบสำหรับคำถามบางประเภท เพื่อเป็นการป้องกันการหลอกลวงโดนทรูดี้ดังที่แสดงในรูป 8-47 มาตรการป้องกันการหลอกลวง (ที่เป็นทางเลือก) นี้เป็นการเพิ่ม hash ให้กับข่าวสารที่เป็นคำตอบด้วยคีย์ส่วนตัวของผู้ที่ถาม เนื่องจากทรูดี้ไม่ทราบค่าคีย์ส่วนตัวของ ".com" server เธอจึงไม่สามารถปลอมคำตอบที่ ISP ของอลิสส่งไปถามได้ ซึ่งแม้ว่าเธอจะสามารถจัดการดักเอาคำตอบมาได้ก่อนแต่ก็ไม่สามารถที่จะปลอมคำตอบได้เนื่องจากค่า hash ในคำตอบนั้นจะผิดเพี้ยนไป

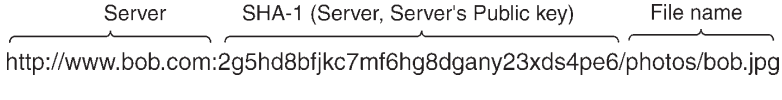
รูปที่ 8-49 URL ที่สามารถตรวจสอบตัวเองได้

DNSsec ยังสนับสนุน record type แบบอื่นๆ เช่น CERT record ซึ่งนำมาใช้ในการจัดเรียงลำดับใบรับรองของ X.509 ข้อมูลชนิดนี้ถูกนำมาใช้โดยคนบางกลุ่มที่ต้องการเปลี่ยน DNS ให้กลายเป็น PKI เป็นต้น

ชื่อที่สามารถตรวจสอบได้ด้วยตนเอง

Secure DNS ไม่ได้วิธีการรักษาความปลอดภัยให้แก่ชื่อแต่เพียงอย่างเดียว อีกแนวทางหนึ่งที่แตกต่างไปจากเดิมถูกนำมาใช้ในระบบ Secure File System ในโครงการนี้ผู้วิจัยได้ออกแบบระบบที่มีความปลอดภัย สามารถขยายขนาดได้ และเป็นระบบที่ใช้งานได้ทั่วโลก โดยไม่ได้แก้ไขมาตรฐาน DNS และไม่ได้ใช้ใบรับรองหรือการตั้งสมมุติฐานว่าจะต้องมี PKI แต่อย่างใด แม้ว่าวิธีการนี้จะสามารถนำไปประยุกต์ใช้กับเว็บเพื่อให้เกิดความปลอดภัยในระดับสูงได้แต่ในปัจจุบันก็ยังคงไม่ได้รับการนำไปใช้งาน เนื่องจากจำเป็นจะต้องมีการเปลี่ยนแปลงซอฟต์แวร์อย่างมากก่อนที่จะนำไปใช้งานได้จริง

เริ่มด้วยการตั้งสมมุติฐานว่าเว็บเซิบเวอร์แต่ละตัวมีคีย์ส่วนตัวและคีย์สาธารณะใช้งาน สิ่งที่สำคัญที่สุดของแนวความคิดนี้ก็คือ แต่ละ URL จะต้องมีการเข้ารหัส hash ของชื่อเซิบเวอร์และคีย์สาธารณะให้เป็นส่วนหนึ่งของ URL ตัวอย่างเช่น รูป 8-49 แสดงให้เห็น URL ของไฟล์รูปถ่ายของบ็อบ ซึ่งเริ่มต้นด้วย http ตามปกติตามด้วยชื่อ DNS ของเซิบเวอร์ (www.bb.com) ตามด้วยเครื่องหมายโคลอน ":" และตัวอักษรที่เป็นค่า hash จำนวน 32 ตัว และปิดท้ายด้วยชื่อไฟล์ตามปกติ นอกเหนือจาก hash แล้วนี่ก็คือ URL ที่ใช้กันตามปกตินั่นเอง แต่เมื่อรวม hash เข้าไปด้วยแล้วนี่ก็คือชื่อ URL ที่สามารถตรวจสอบตัวเองได้ หรือ self-certifying URL



คำถามที่น่าสนใจก็คือ ค่าของ hash นั้นเข้ามาทำอะไร ค่า hash ถูกคำนวณโดยการต่อท้ายชื่อ DNS ของเซิบเวอร์ด้วยคีย์สาธารณะและใช้วิธีการ SHA-1 ในการคำนวณค่าออกมาเป็น hash ขนาด 160 บิต ในวิธีการนี้ค่า hash ถูกเขียนให้อยู่ในรูปแบบของตัวอักษร (lower case letter) จำนวน 32 ตัวโดยเลือกที่จะไม่ใช้ตัวอักษร "l" และ"o" และตัวเลข "1" และ "0" เพื่อหลีกเลี่ยงความสับสนในการใช้งาน ซึ่งจะทำให้เหลือตัวอักษรและตัวเลขจำนวน 32 ตัว ด้วยตัวอักษรและตัวเลขแต่ละตัวในจำนวน 32 ตัวนี้จะถูกกำหนดค่าให้เป็นบิตจำวน 5 บิตทำให้ string 32 ตัวนี้มีค่าเท่ากับ 160 บิตซึ่งก็เป็นค่าของ hash นั่นเอง แม้ว่าในความจริงไม่จำเป็นจะต้องใช้ค่าของ hash เลย นั่นคือ ตัวคีย์เองก็สามารถนำมาใช้งานได้แล้ว แต่ข้อได้เปรียบของการใช้ hash คือสามารถลดขนาดของชื่อให้สั้นลงได้

วิธีการที่ง่ายที่สุด (แต่ไม่สะดวกเลย) ในการดูเว็บไซต์ที่มีรูปถ่ายของบ็อบก็คือ อลิสจะต้องพิมพ์ชื่อดังที่ปรากฏในรูป 8-49 ลงไปในบราวเซอร์ของเธอ บราวเซอร์จะส่งข่าวสารไปยังเว็บไซต์ของบ็อบเพื่อขอทราบคีย์สาธารณะ เมื่อได้รับคีย์แล้วบราวเซอร์จะนำชื่อเซิบเวอร์และคีย์สาธารณะมาต่อเข้าด้วยกันและคำนวณหาค่า hash ถ้าผลที่ออกมานั้นมีค่าตรงกับตัวอักษร 32 ตัวที่อยู่ในชื่อของเว็บไซต์ก็จะทำให้เชื่อถือได้ว่านั่นคือคีย์สาธารณะของบ็อบ อย่างไรก็ตาม เนื่องจากคุณสมบัติของวิธี SHA-1 แม้ว่าทรูดี้จะสามารถดักจับข่าวสารนี้ได้และส่งคำตอบปลอมกลับมาเธอก็จะไม่สามารถหาคีย์สาธารณะตัวอื่นที่จะให้ค่าตรงกับ hash ที่มีอยู่ การแก้ไขใดๆ ที่ทรูดี้กระทำก็จะสามารถตรวจพบได้และคีย์สาธารณะของบ็อบก็จะสามารถถูกเก็บไว้ใน cache เพื่อการใช้งานในอนาคตได้

ขั้นต่อไปอลิสจะต้องตรวจสอบให้ได้ว่าบ็อบมีคีย์ส่วนตัวที่คู่กับคีย์สาธารณะที่เธอมีอยู่ เธอจะสร้างข่าวสารที่ประกอบด้วย AES session key, nonce, และการบันทึกเวลา (timestamp) จากนั้นจะทำการเข้ารหัสข้อมูลด้วยคีย์สาธารณะของบ็อบแล้วส่งไปให้เขา เนื่องจากบ็อบ (ตัวจริง) เท่านั้นที่จะมีคีย์ส่วนตัวที่ตรงกับคีย์สาธารณะที่ใช้ในการเข้ารหัสข้อมูล บ็อบจึงจะสามารถถอดรหัสข้อมูลได้และส่งข้อความ nonce ที่เข้ารหัสโดยใช้ AES session key กลับมายังอลิส เมื่ออลิสได้รับและถอดข้อความออกมาเป็น nonce ที่ถูกต้องเธอก็จะทราบได้ทันทีว่ากำลังสื่อสารอยู่กับบ็อบตัวจริง และในเวลานี้ทั้งบ็อบและอลิสก็จะมี AES session key สำหรับใช้ในการเข้าและถอดรหัสในการแลกเปลี่ยนข่าวสารอื่นต่อไป

เมื่ออลิสได้รับรูปของบ็อบ (หรือเว็บเพจใดๆ) เธอก็จะสามารถสร้าง bookmark เพื่อการใช้งานในอนาคตได้โดยไม่ต้องพิมพ์ชื่อ (อันแสนจะยาว) ของเว็บไซท์บ็อบอีกต่อไป ยิ่งกว่านั้น URL ที่ถูกเก็บไว้ในเว็บเพจยังสามารถตรวจสอบตนเองได้ ทำให้การใช้งานครั้งต่อไปนั้นง่ายขึ้นกว่าเดิมและยังมีความปลอดภัยมากขึ้น วิธีการอื่นที่อาจนำมาใช้ในการขอชื่อเว็บไซต์ (ที่ยาวเหยียด) นี่คือการใช้การเชื่อมต่อที่ปลอดภัยไปยังเซิบเวอร์ที่ไว้ใจได้หรือการจัดส่งมาโดยใช้ใบรับรอง X.509 โดย CA แห่งหนึ่ง

อีกวิธีการหนึ่งที่นำมาใช้ในการให้ได้มาซึ่งชื่อ URL ที่มีการตรวจสอบตัวเองได้คือการเชื่อมต่อไปยัง search engine ที่ไว้ใจได้ด้วยการพิมพ์ชื่อ URL ที่ตรวจสอบตัวเองได้ของ search engine นั้นเข้าไป (ในครั้งแรก) และทำงานผ่านโพรโตคอลที่กล่าวถึงข้างต้น ซึ่งจะนำไปสู่การเชื่อมต่อที่ปลอดภัยเข้ากับ search engine ที่ไว้ใจได้ และจะนำไปสู่การค้นหาเว็บไซต์ที่ต้องการได้โดยมีชื่อ URL ที่ตรวจสอบตัวเองได้อย่างพร้อมมูลซึ่งจะเป็นวิธีการที่สะดวกต่อผู้ใช้เป็นอย่างมาก

ต่อไปมาดูว่าวิธีการนี้จะสามารถป้องกันการหลอกลวงจากทรูดี้ (DNS spoofing attack) ได้หรือไม่ ถ้าทรูดี้สามารถแก้ไขข้อมูลใน cache ใน ISP ของอลิสได้ดังที่กล่าวมาแล้ว คำถามของอลิสก็จะถูกส่งไปที่ทรูดี้แทนที่จะเป็นบ็อบ แต่โพรโตคอลกำหนดให้ว่าผู้ที่รับข่าวสารนี้ (ในครั้งแรก) จะต้องสามารถส่งคีย์สาธารณะที่สอดคล้องกับค่า hash ที่มีอยู่ได้ ถ้าทรูดี้ส่งคีย์สาธารณะของเธอเองกลับมาอลิสก็จะสามารถตรวจพบได้ในทันทีเพราะค่า hash ที่คำนวณได้จะไม่ตรงกับค่า hash ที่มีมาพร้อมกับชื่อ URL ถ้าทรูดี้ส่งคีย์สาธารณะของบ็อบกลับมา อลิสจะไม่สามารถตรวจพบว่ากำลังคุยอยู่กับทรูดี้ได้ แต่อลิสจะเข้ารหัสข่าวสารต่อไปโดยใช้คีย์ของบ็อบ ทรูดี้จะได้รับข่าวสารนี้แต่ก็ไม่มีทางที่จะถอดรหัส AES session key และ nonce ออกมาได้ ไม่ว่าจะอย่างไรก็ตามผลการหลอกลวง DNS ก็จะกลายเป็นเพียงการโจมตีแบบ Denial-of-service attack

8.9.3 เทคนิค SSL-The Secure Sockets Layer

การใช้ชื่อที่ปลอดภัยนับเป็นการเริ่มต้นที่ดี แต่ก็ยังมีเรื่องอีกมากมายที่เกี่ยวข้องกับการรักษาความปลอดภัยบนเว็บ ขั้นต่อไปคือการสร้างการเชื่อมต่อที่มีความปลอดภัยซึ่งจะได้กล่าวถึงดังต่อไปนี้

เมื่อเว็บเริ่มได้รับความสนใจจากสาธารณะชนอย่างแพร่หลายนั้น เป็นการนำเสนอข้อมูลที่ไม่มีการเปลี่ยนแปลงเสียเป็นส่วนมาก อย่างไรก็ตาม บางองค์กรก็เริ่มมีแนวความคิดที่จะนำเว็บมาใช้ในการทำธุรกรรมทางด้านการเงิน เช่น การซื้อสินค้าโดยใช้บัตรเครดิต การติดต่อธนาคารแบบออนไลน์ (on-line banking) และการซื้อขายหุ้นทางอิเล็กทรอนิกส์ งานประยุกต์เหล่านี้ได้เป็นตัวกระตุ้นให้เกิดความต้องการการเชื่อมต่อที่มีความปลอดภัย (secure connection) ในปี 1995 บริษัท Netscape Communication Corp. ซึ่งเป็นผู้นำตลาดของการใช้เว็บบราวเซอร์อยู่ในขณะนั้นได้ตอบสนองต่อความต้องการนี้ด้วยการสร้างสิ่งที่เรียกว่า SSL (Secure Sockets Layer) ขึ้นมา ซอฟต์แวร์และโพรโตคอลนี้ได้รับการนำไปใช้งานอย่างแพร่หลายซึ่งรวมถึง Internet Explorer ของบริษัทไมโครซอฟต์ด้วย

SSL สร้างการเชื่อมต่อที่ปลอดภัยระหว่างซ็อคเก็ต (socket) คู่หนึ่งรวมทั้ง

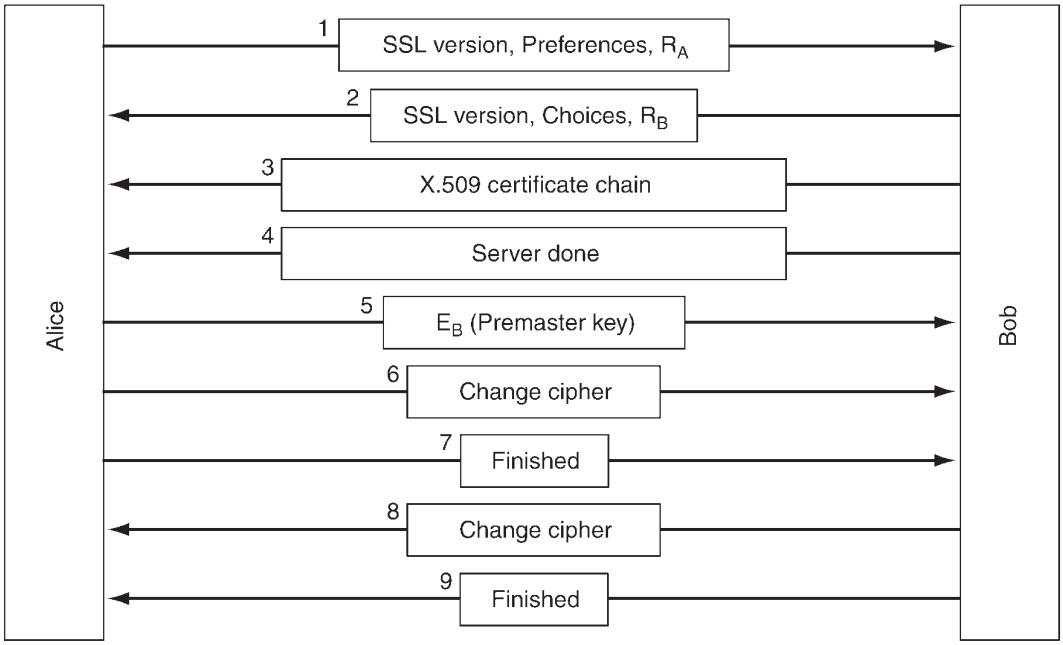
1. การต่อรองค่าพารามิเตอร์ระหว่างผู้ใช้บริการและผู้ให้บริการ
2. ทำการตรวจสอบผู้ใช้ทั้งสองฝ่ายสำหรับทั้งผู้ใช้และผู้ให้บริการ
3. การสื่อสารที่เป็นความลับ
4. การปกป้องความถูกต้องของข้อมูล

ซึ่งคุณสมบัติทั้งสี่ประการนี้ได้ถูกกล่าวถึงไปแล้วจึงจะไม่กล่าวถึงอีก

ตำแหน่งของ SSL ในโพรโตคอลสแต็กนั้นแสดงให้เห็นในรูป 8-50 SSL เป็นชั้นสื่อสารใหม่ที่ทำงาน

Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

รูปที่ 8-50
โพรโตคอลสแต็ก



รูปที่ 8-51
โพรโตคอลย่อย
สำหรับการจัดตั้งการ
เชื่อมต่อ SSL

อยู่ระหว่างชั้นสื่อสารโปรแกรมประยุกต์และชั้นสื่อสารนำส่งข้อมูลที่มีประสิทธิภาพในการรับคำร้องจากบราวเซอร์แล้วจัดการส่งผ่านการสื่อสาร TCP ไปยังเซิบเวอร์ เมื่อการเชื่อมต่อที่ปลอดภัยเกิดขึ้นแล้ว หน้าที่หลักของ SSL คือการจัดการการบีบอัดข้อมูลและการถอดรหัสข้อมูล เมื่อ HTTP ถูกใช้งานผ่าน SSL จะถูกเรียกว่าเป็น HTTPS (Secure HTTP) แม้ว่าจะยังคงเป็น HTTP แบบมาตรฐานก็ตาม บางครั้ง SSL ก็มีให้ใช้งานผ่านพอร์ต 443 แทนที่จะเป็นพอร์ตมาตรฐาน 80 นอกจากนี้ SSL ยังไม่ได้จำกัดการใช้งานอยู่แต่เพียงเว็บบราวเซอร์เท่านั้นแต่ยังสามารถนำไปใช้งานร่วมกับโปรแกรมประยุกต์ทั่วไปได้ด้วย

โพรโตคอล SSL ได้รับการพัฒนามาแล้วหลายรุ่น ในที่นี้จะพิจารณารุ่นที่ 3 ซึ่งเป็นรุ่นที่ได้รับความนิยมในการนำมาใช้งานมากที่สุด SSL สนับสนุนการทำงานร่วมกับอัลกอริทึมและทางเลือกจำนวนมากทางเลือกเหล่านี้รวมทั้งการมีอยู่หรือไม่มีอยู่ของการบีบอัดข้อมูล การเข้ารหัสข้อมูล และมาตรการอื่นๆที่อาจเกี่ยวข้องกับการส่งออกเทคโนโลยีทางด้านการเข้ารหัสข้อมูลซึ่งเป็นข้อจำกัดของรัฐบาลสหรัฐอเมริกา นอกจากนี้ยังทำให้แน่ใจว่าการเข้ารหัสข้อมูลที่เข้มแข็งถูกนำมาใช้ในทั้งสองด้านของการเชื่อมต่อที่เกิดขึ้นในสหรัฐอเมริกา ในด้านอื่นๆ คีย์จะถูกจำกัดขนาดไว้ที่ 40 บิตซึ่งในกลุ่มผู้วิจัยเกี่ยวกับเรื่องการเข้ารหัสข้อมูลไม่อาจยอมรับได้ (เพราะเป็นขนาดที่เล็กมากเกินไป) Netscape ได้ถูกบังคับให้ใช้ข้อกำหนดเหล่านี้เพื่อที่จะได้สามารถผลิตเป็นสินค้าส่งออกนอกสหรัฐอมเริกาได้

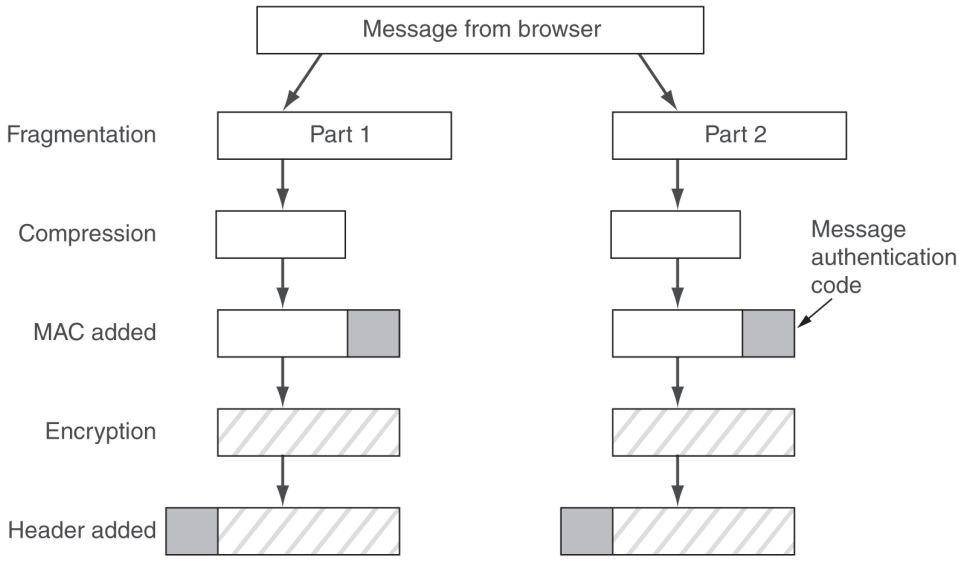
SSL ประกอบด้วยโพรโตคอลย่อยสองโพรโตคอล อันหนึ่งใช้สำหรับการสร้างการเชื่อมต่อที่ปลอดภัย ส่วนอีกอันหนึ่งเป็นตัวที่ใช้งานการเชื่อมต่อนี้ ขอเริ่มต้นการอธิบายโดยการชี้ให้เห็นว่าการเชื่อมต่อที่ปลอดภัยถูกสร้างขึ้นมาได้อย่างไร รูป 8-51 แสดงโพรโตคอลย่อยสำหรับการสร้างการเชื่อมต่อ โพรโตคอลเริ่มต้นที่อลิสส่งข่าวสาร 1 ไปยังบ็อบเพื่อขอจัดตั้งการเชื่อมต่อขึ้นมา คำขอนั้นจะระบุรุ่นของ SSL ที่อลิสใช้และความต้องการของเธอที่เกี่ยวกับการบีบอัดข้อมูลและการเข้ารหัสข้อมูล นอกจากนี้ยังมี nonce (RA) ซึ่งจะถูกนำมาใช้ในภายหลัง

ต่อไปเป็นคราวที่บ็อบจะต้องส่งข่าวสารบ้าง ในข่าวสารที่ 2 บ็อบได้จัดการเลือกอัลกอริทึมต่างๆ ที่อลิสสามารถสนับสนุนได้และจัดการส่ง nonce (RB) ของเขามา จากนั้นข่าวสารที่ 3 บ็อบจะส่งใบรับรองที่มีคีย์สาธารณะของเขามาด้วย ถ้าใบรับรองนี้ไม่ได้รับการลงชื่อรับรองโดยคนที่อลิสไว้ใจ เขาก็จะต้องส่งใบรับรองของผู้ที่รับรอง (chain of certificates) คีย์สาธารณะของเขามาด้วย บราวเซอร์ในปัจจุบันจะมีคีย์สาธารณะของ CA ที่เป็นที่รู้จักกันทั่วไปมากกว่า 100 แห่งติดมาด้วยแล้ว ทำให้บ็อบสามารถระบุผู้รับรองที่เป็นหนึ่งในจำนวนนี้ได้ และอลิสก็จะสามารถตรวจสอบคีย์สาธารณะของบ็อบได้ ในระหว่างนี้บ็อบสามารถที่จะส่งข่าวสารอื่นๆ เช่น คำร้องขอใบรับรองคีย์สาธารณะของอลิส มาได้ เมื่อบ็อบส่งข่าวสารเสร็จแล้วก็จะส่งข่าวสารที่ 4 มายังอลิส

อลิสตอบสนองด้วยการส่งเลขสุ่มขนาด 384 บิตเรียกว่า premaster key มายังบ็อบซึ่งจะต้องจัดการเข้ารหัสด้วยคีย์สาธารณะของบ็อบ (ข่าวสารที่ 5) session key ที่จะนำมาใช้งานจริงนั้นจะถูกสร้างขึ้นมาจาก premaster key รวมกับ nonce จากทั้งสองคนด้วยวิธีการที่ซับซ้อน หลังจากได้รับข่าวสาร 5 แล้วทั้งอลิสและบ็อบจะสามารถคำนวณหา session key ได้ ด้วยเหตุผลนี้เองอลิสจะบอกบ็อบให้เปลี่ยนไปใช้การเข้ารหัสแบบใหม่ (ข่าวสารที่ 6) และก็เป็นการจบสิ้นการทำงานของโพรโตคอลย่อยนี้ บ็อบตอบรับด้วยข่าวสารที่ 8 และที่ 9

อย่างไรก็ตาม แม้ว่าอลิสจะทราบว่าคนที่กำลังสื่อสารด้วยนั้นคือบ็อบ แต่บ็อบไม่ทราบว่ากำลังสื่อสารอยู่กับใคร (นอกจากอลิสจะมีคีย์สาธารณะและใบรับรองคีย์นั้นซึ่งเป็นเหตุการณ์ที่ไม่ปกติสำหรับผู้ใช้งานส่วนตัวทั่วไป) ดังนั้น ข่าวสารแรกของบ็อบอาจเป็นการขอให้อลิส log-in โดยการใช้ user name และ password (ซึ่งอาจได้มาจากการติดต่อครั้งก่อนหน้า) แต่โพรโตคอลที่ใช้ในการ log-in ไม่เกี่ยวข้องกับขอบเขตงานของ SSL เมื่อได้ทำงานเสร็จสิ้นแล้ว (จะด้วยวิธีการใดก็ตาม) การแลกเปลี่ยนข้อมูลก็จะเกิดขึ้นได้

ดังที่ได้กล่าวข้างต้น SSL สนับสนุนการใช้อลักอริทึมในการเข้ารหัสข้อมูลหลายแบบ วิธีการที่ เข้มแข็งที่สุดคือแบบ triple DES ซึ่งใช้คีย์แยกจากกันสามตัวในการเข้ารหัส SHA-1 เพื่อให้เกิดความมั่นคงของข่าวสารที่จะส่ง การรวมวิธีการหลายแบบเข้าด้วยกันนี้ทำให้เกิดความล่าช้าในการใช้งานจึงเป็นวิธี



รูปที่ 8-52 การนำส่งข้อมูลด้วย SSL

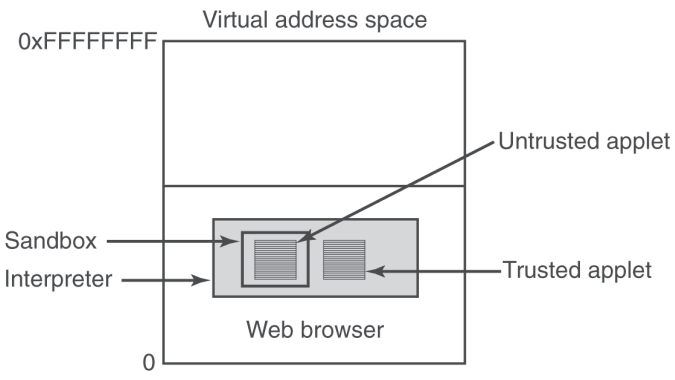
ที่นำไปใช้กับการติดต่อกับธนาคารหรือการติดต่อที่ต้องการการรักษาความปลอดภัยในระดับสูง สำหรับการใช้งาน e-commerce ทั่วไปนิยมใช้วิธี RC4 ร่วมกับคีย์ขนาด 128 บิตในการเข้ารหัสข้อมูลและใช้วิธี MD5 สำหรับการตรวจสอบผู้ใช้ วิธี RC4 นำคีย์ขนาด 128 บิตมาใช้เป็นฐานและทำการขยายออกไปเป็นตัวเลขที่มีขนาดใหญ่ขึ้นสำหรับการทำงานในขั้นตอนต่อไป จากนั้นจึงสร้าง keystream แล้วนำมา exclusive-OR กับ plaintext เพื่อให้ได้เป็น stream cipher ดังที่เห็นในรูป 8-14

สำหรับการนำส่งข้อมูลที่เกิดขึ้นจริงจะเกิดขึ้นโดยใช้โพรโตคอลย่อยอันที่สอง ดังที่แสดงในรูป 8-52 ข่าวสารจากบราวเซอร์จะถูกแบ่งออกเป็นส่วนย่อยขนาดส่วนละไม่เกิน 16 KB ถ้ามีการเลือกใช้การบีบอัดข้อมูลแต่ละส่วนย่อยจะถูกบีบอัดแยกจากกัน หลังจากนั้น คีย์ลับที่สร้างขึ้นมาจาก nonce ทั้งสองและ premaster key จะถูกนำมาเรียงต่อจากข้อความและทำการสร้าง hash ขึ้นมาด้วยอัลกอริทึมที่ตกลงกันระหว่างผู้ใช้และผู้ให้บริการ (โดยปกติคือ MD5) ค่าของ hash จะถูกเขียนต่อกันกับข้อมูลแต่ละส่วน เรียกว่า MAC ข้อมูลส่วนที่ถูกบีบอัดแล้วและ MAC จะถูกเข้ารหัสด้วยอัลกอริทึมแบบคีย์สมมาตร (โดยปกติคือการทำ exclusive-OR กับ keystream ที่ได้จาก RC4) ขั้นตอนสุดท้าย header สำหรับข้อมูลแต่ละส่วนจะถูกใส่เข้าไปแล้วข้อมูลทั้งหมดจึงถูกนำส่งผ่านการเชื่อมต่อ TCP

เนื่องจากได้แสดงให้เห็นแล้วว่าวิธีการแบบ RC4 มีจุดอ่อนบางประการที่อาจทำให้ถูกถอดรหัสได้ การรักษาความปลอดภัย SSL โดยใช้ RC4 จึงไม่มีอนาคตที่แจ่มใสนัก สำหรับบราวเซอร์ที่อนุญาตให้ผู้ใช้กำหนดค่าตัวเลือกได้เองควรหันมาใช้วิธี triple DES พร้อมกับคีย์ขนาด 168 บิตร่วมกับ SHA-1 ตลอดเวลา แม้ว่าจะเป็นทางเลือกที่ทำงานได้ช้ากว่าการใช้วิธี RC4 ร่วมกับ MD5 ก็ตาม

ในปี ค.ศ. 1996 Netscape Communication Corp. ได้ส่ง SSL ให้แก่องค์กร IETF เพื่อทำการกำหนดให้เป็นมาตรฐาน ซึ่งผลที่ได้รับคือมาตรฐานที่เรียกว่า TLS (Transport Layer Security) ซึ่งได้อธิบายไว้ในมาตรฐาน RFC 2246

การแก้ไขเปลี่ยนแปลงที่เกิดขึ้นกับ SSLนั้นเกิดขึ้นน้อยมาก แต่ก็มากพอที่จะทำให้ SSL รุ่น 3 ไม่สามารถทำงานร่วมกับ TLS ได้ ตัวอย่างเช่น วิธีการสร้าง session key ขึ้นมาจาก nonce และ premaster key ได้ถูกเปลี่ยนไปใช้วิธีการอื่นเพื่อทำให้คีย์มีความเข้มแข็งมากขึ้นกว่าเดิม TLS ได้กลายมาเป็น SSL รุ่น 3.1 ซึ่งได้รับการสร้างขึ้นมาใช้งานเป็นครั้งแรกในปี 1999 แต่ก็ยังไม่เป็นที่ชัดเจนว่า TLS จะเข้ามาแทนที่ SSL ในการใช้งานทั่วไปหรือไม่แม้ว่าจะเป็นวิธีการที่เข้มแข็งมากขึ้นก็ตาม



รูปที่ 8-53 Applets ที่สามารถประมวลผลได้บนบราวเซอร์

8.9.4 การใช้โค้ดที่ปลอดภัยสำหรับโมบาย

การตั้งชื่อและการเชื่อมต่อคือสองประเด็นที่สำคัญที่เกี่ยวข้องกับการรักษาความปลอดภัยบนเว็บ แต่ก็ยังมีเรื่องอื่นที่จะต้องพิจารณาอีก ในยุคต้นๆของการใช้เว็บ เมื่อเว็บเพจยังคงเป็นเพียงหน้าต่างข้อมูลแบบ static HTML ไฟล์ ซึ่งไม่มีโค้ดที่สามารถประมวลผลได้ร่วมอยู่ด้วย ในปัจจุบันเว็บเพจมักจะมีส่วนประกอบที่เป็นโปรแกรมขนาดเล็ก เช่น Java applets, ActiveX controls, และ Java scripts การ download และการประมวลผล โค้ดโมไบล์ (mobile code) เหล่านี้เห็นได้ชัดเจนว่ามีความเกี่ยวพันกับความเสี่ยงในการรักษาความปลอดภัยเป็นอย่างมาก ทำให้มีการคิดค้นวิธีการหลากหลายวิธีเพื่อลดปัญหาที่โค้ดเหล่านี้อาจสร้างขึ้นได้

การรักษาความปลอดภัย Java applets

Java applets คือโปรแกรมจาวาขนาดเล็กที่ได้รับการออกแบบให้สามารถทำงานบน JVM (Java Virtual Machine) โปรแกรมนี้อาจถูกใส่ไว้ในเว็บเพจสำหรับการ download ไปพร้อมๆ กับข้อความในเว็บเพจนั้น ภายหลังจากเพจนั้นได้รับการ load เรียบร้อยแล้ว applets จะถูกส่งไปประมวลผลแบบ Interpreted บน JVM ที่ติดตั้งไว้ภายในบราวเซอร์ ดังที่แสดงในรูป 8-53

ข้อได้เปรียบของการประมวลผลแบบนี้คือคำสั่งทุกคำสั่งจะถูกตรวจสอบก่อนที่จะเกิดการประมวลผลจริง ซึ่งเป็นการเปิดโอกาสให้ Interpreter อย่างเช่น JVM ทำการตรวจสอบความถูกต้องของคำสั่งนั้นๆ ได้ก่อน นอกจากนี้การเรียกใช้ system call ยังจะได้รับการตรวจจับและทำการแปลความหมาย วิธีการตรวจสอบ system call นั้นเป็นเรื่องของกระบวนการรักษาความปลอดภัย ตัวอย่างเช่น ถ้า applet นั้นได้รับความไว้วางใจ system call ที่ถูกเรียกใช้ก็จะได้รับการประมวลผลได้ในทันที อย่างไรก็ตาม ถ้าเป็น applet ที่ไม่ได้รับความไว้วางใจแล้วก็จะถูกห่อหุ้มด้วยโค้ดที่ทำหน้าที่เสมือน sandbox ที่คอยจำกัดพฤติกรรมและหยุดยั้งความพยายามที่จะให้ทรัพยากรของระบบ (ซึ่งอาจเป็นไปในทางที่ไม่ปลอดภัยก็ได้)

เมื่อ applet พยายามที่จะใช้ทรัพยากรของระบบคอมพิวเตอร์ การเรียกใช้นั้นจะถูกส่งต่อไปยังผู้ตรวจการรักษาความปลอดภัย (security monitor) เพื่อให้ทำการอนุญาตเสียก่อน ผู้ตรวจฯ จะตรวจสอบการเรียกใช้ทรพัยากรนั้นว่าเป็นไปตามนโยบายการรักษาความปลอดภัยหรือไม่จากนั้นจะทำการตัดสินใจอนุญาตหรือไม่อนุญาตต่อไป ด้วยวิธีการนี้ จึงมีความเป็นไปได้ที่จะควบคุมการใช้งานทรัพยากรให้เป็นไปตามที่ต้องการได้ แต่ในความเป็นจริงแล้ว การตรวจการรักษาความปลอดภัยไม่สามารถทำงานได้ตามที่ต้องการ

ActiveX

ActiveX controls หมายถึงโปรแกรมที่สามารถใส่ไว้ภายในเว็บเพจได้ เมื่อคอมพิวเตอร์ตรวจพบโปรแกรมประเภทนี้ก็จะทำการตรวจสอบดูว่าสมควรที่จะทำการประมวลผลหรือไม่ซึ่งถ้าผ่านการทดสอบก็จะทำการประมวลผล โปรแกรมประเภทนี้จะไม่ถูก interpret หรือห่อหุ้มด้วย sandbox ดังนั้นจึงมีความสามารถในการทำงานได้มากเท่ากับโปรแกรมของผู้ใช้เองและมีแนวโน้มที่จะทำอันตรายได้เป็นอย่างมาก ดังนั้น การรักษาความปลอดภัยจึงอยู่ในขั้นตอนการตัดสินใจว่าจะอนุญาตให้ประมวลผลโปรแกรมนี้หรือไม่

วิธีการที่บริษัทไมโครซอฟต์ซึ่งเป็นผู้คิดค้น ActiveX control ตั้วอยู่บนแนวคิดที่เรียกว่า "code signing" ActiveX control แต่ละตัวจะถูกเขียนกำกับด้วยลายเซ็นต์อิเล็กทรอนิกส์ซึ่งหมายถึง hash code ที่ลงชื่อกำกับโดยผู้ที่เขียนโปรแกรมนั้นๆ ขึ้นมาโดยใช้เทคนิคการเข้ารหัสแบบคีย์สาธารณะ เมื่อบราวเซอร์ค้นหาว่ามี ActiveX control อยู่ในเว็บเพจ บราวเซอร์จะทำการตรวจสอบลายเซ็นต์ของโปรแกรมนั้นว่าเป็นโค้ดที่ไม่ได้ถูกดัดแปลงไปแต่อย่างใด ถ้าลายเซ็นต์นั้นถูกต้อง บราวเซอร์ก็จะทำการตรวจสอบตาราง ข้อมูลภายในว่าผู้ที่เขียนโปรแกรมนี้ขึ้นมาเป็นผู้ที่เชื่อถือได้หรือเปล่า ถ้าเป็นผู้ที่เชื่อถือได้ก็จะทำการประมวลผลโปรแกรมนั้นมิฉะนั้นก็จะไม่อนุญาตให้ทำการประมวลผล กระบวนการตรวจสอบ ActiveX control นี้เรียกว่า Authenticode

เมื่อทำการเปรียบเทียบแนวทางของ Java applet กับ ActiveX control จะพบว่า Java applet ไม่มีการพยายามที่จะตรวจสอบดูว่าใครคือผู้ที่เขียนโปรแกรมนั้นๆ ขึ้นมา แต่ run-time interpreter จะทำหน้าที่ในการควบคุมการทำงานของ applet ไม่ให้ไปทำในสิ่งที่ไม่ได้รับอนุญาต ในวิธีการของ code signing จะไม่มีการตรวจสอบการทำงานของโค้ดแต่อย่างใด ถ้าโค้ดนั้นมาจากแหล่งกำเนิดที่ไว้ใจได้และไม่ได้ถูกแก้ไขเปลี่ยนแปลงในระหว่างการนำส่งแล้ว ก็จะอนุญาตให้ทำการประมวลผลโค้ดนั้นได้ โดยไม่มีการตรวจสอบว่าโค้ดนั้นถูกเขียนขึ้นมาเพื่อทำอะไร เช่น ถ้าโปรแกรมเมอร์ผู้เขียนโค้ดนั้นตั้งใจที่จะให้โค้ดทำการ format ฮาร์ดดิสก์ จากนั้นลบข้อมูลใน flash ROM ทิ้งไปทำให้คอมพิวเตอร์เครื่องนั้นไม่สามารถที่จะเปิดใช้งานได้อีกต่อไปและถ้าโปรแกรมเมอร์ได้รับการรับรองว่าเป็นผู้ที่ไว้ใจได้แล้ว โค้ดนั้นก็จะสามารถถูกประมวลผลได้และทำลายคอมพิวเตอร์เครื่องนั้น

JavaScript

JavaScript ไม่มีรูปแบบการรักษาความปลอดภัย แต่มีประวัติอันยาวนานเกี่ยวกับการสร้างขึ้นมาใช้งานที่มีรูรั่วมากมาย แต่ละบริษัทซอฟต์แวร์มีวิธีการรักษาความปลอดภัยหลายวิธีแตกต่างกัน ตัวอย่างเช่น Netscape Navigator รุ่น 2 ใช้รูปแบบการรักษาความปลอดภัยที่มีความคล้ายคลึงกับ Java แต่ในรุ่นที่ 4 ได้กลับหันมาใช้วิธี code signing แทน

พื้นฐานของปัญหาอยู่ที่การที่จะอนุญาตให้โค้ดที่มาจากที่อื่นสามารถถูกประมวลผลบนเครื่องของตนเองได้นั้นเป็นเรื่องที่มีความเสี่ยงสูง จากในมุมมองของการรักษาความปลอดภัยนั้นได้เปรียบเทียบการทำงานในลักษณะนี้ว่าเหมือนเป็นการเชื้อเชิญให้ขโมยเข้ามาในบ้านของตนเองแล้วพยายามที่จะคอยจับตาไม่ให้เขาหนีจากห้องครัวไปยังห้องนั่งเล่นได้ ถ้ามีสิ่งไม่คาดฝันเกิดขึ้นทำให้เจ้าของบ้านหันเหความสนใจไปทางอื่นแล้ว สิ่งที่เลวร้ายก็อาจจะเกิดขึ้นได้ ความตึงเครียดในที่นี้ก็คือผู้ใช้โมบายทั้งหลายต้องการภาพกราฟฟิกที่น่าตื่นตาตื่นใจและมีการโต้ตอบที่รวดเร็ว ในขณะที่ผู้ออกแบบเว็บไซต์ต่างก็คิดว่าสิ่งที่น่าตื่นตาตื่นใจนี้มีความสำคัญมากกว่าการรักษาความปลอดภัยโดยเฉพาะอย่างยิ่งเมื่อความเสี่ยงต่อความไม่ปลอดภัยนั้นเกิดขึ้นที่เครื่องคอมพิวเตอร์ของผู้อื่น

Viruses

ไว้รัส (Virus) เป็นอีกรูปแบบหนึ่งของโมบายโค้ด (mobile code) สิ่งที่แตกต่างไปจากตัวอย่างที่กล่าวข้างต้นก็คือ ไวรัสเป็นโปรแกรมที่ไม่ได้รับการเชื้อเชิญเลยแม้แต่น้อย ความแตกต่างระหว่างโมบายโค้ดกับไวรัสนั้นคือไวรัสถูกเขียนขึ้นมาเพื่อสร้างตัวเองขึ้นมาใหม่ เมื่อไวรัสมาถึงไม่ว่าจะด้วยวิธีผ่านมาทางเว็บเพจหรือติดมากับสิ่งที่ส่งมาพร้อมกับอีเมลล์ หรือด้วยวิธีใดก็ตาม มันก็จะเริ่มทำงาน

ด้วยการแพร่กระจายตัวเองไปยังไฟล์ที่สามารถประมวลผลได้ที่เก็บอยู่ในดิสก์ในทันที เมื่อหนึ่งในโปรแกรมเหล่านี้ถูกประมวลผล การควบคุมการทำงานของโปรแกรมจะถูกโอนไปให้โปรแกรมไวรัสซึ่งโดยปกติก็จะพยายามกระจายตัวเองออกไปยังคอมพิวเตอร์เครื่องอื่น เช่น การกระจายผ่านอีเมลล์โดยนำที่อยู่ผู้รับมาจาก e-mail address book ของเครื่องที่ติดไวรัสนั้น ไวรัสบางชนิดจะกระจายไปยัง boot sector ของฮาร์ดดิสก์ทำให้โปรแกรมไวรัสเริ่มทำงานทันทีที่เปิดเครื่องคอมพิวเตอร์ ไวรัสได้กลายเป็นปัญหาขนาดยักษ์ในระบบอินเตอร์เน็ตซึ่งในแต่ละปีได้สร้างความเสียหายขึ้นมามีมูลค่าสูงหลายพันล้านเหรียญสหรัฐ โชคไม่ดีที่ไม่มีหนทางแก้ปัญหานี้ได้อย่างเด็ดขาด

8.10 บทสรุปท้ายบท

การเข้ารหัสข้อมูลเป็นเครื่องมือที่สามารถนำมาใช้ในการเก็บรักษาข่าวสารไว้เป็นความลับและทำให้แน่ใจในความถูกต้องของข้อมูลและการตรวจสอบผู้ใช้ได้อย่างถูกต้อง เทคนิคการเข้ารหัสข้อมูลสมัยใหม่นั้นพัฒนาขึ้นมาจากแนวความคิดของ Kerckhoff ที่กำหนดให้มีอัลกอริทึมที่เป็นที่รู้จักกันโดยทั่วไปและคีย์ลับ อัลกอริทึมเข้ารหัสข้อมูลจำนวนมากใช้วิธีการเปลี่ยนแปลงรูปแบบข้อมูลที่ซับซ้อนที่เกี่ยวข้องกับการแทนที่และการเปลี่ยนแปลงในรูปแบบต่างๆ ในการแปลงข้อความธรรมดาที่เรียกว่า plaintext ไปสู่ข้อความที่เข้ารหัสแล้ว เรียกว่า ciphertext อย่างไรก็ตามถ้าการเข้ารหัสข้อมูลปริมาณมากสามารถใช้งานได้จริง การใช้โค้ดรหัสที่ใช้งานเพียงครั้งเดียวจะสนับสนุนให้เกิดวิธีการเข้ารหัสที่ป้องกันการถูกแก้ไขได้อย่างแท้จริง

อัลกอริทึมสำหรับการเข้ารหัสข้อมูลสามารถแบ่งออกได้เป็นการเข้ารหัสโดยใช้คีย์สมมาตร และการเข้ารหัสโดยใช้คีย์สาธารณะ การเข้ารหัสโดยใช้คีย์สมมาตรทำการเปลี่ยนแปลงบิตข้อมูลด้วยการทำงานวนซ้ำหลายรอบและเปลี่ยนค่าพารามิเตอร์ไปเรื่อยๆ โดยการใช้คีย์เป็นตัวกำหนดเพื่อเปลี่ยน plaintext ให้กลายเป็น ciphertext วิธี Triple DES และ Rijndael (AES) เป็นวิธีการที่ได้รับความนิยมนำมาใช้งานเป็นอย่างมากในปัจจุบัน อัลกอริทึมทั้งสองนี้สามารถนำมาใช้ได้ใน electronic code book mode, cipher block chaining mode, stream cipher mode, counter mode, และอื่นๆ

การเข้ารหัสโดยใช้คีย์สาธารณะมีคุณสมบัติที่ใช้คีย์ต่างกันในการเข้ารหัสและถอดรหัสข้อมูลและคีย์ทั้งสองไม่อาจที่จะนำไปใช้สร้างคีย์อีกตัวหนึ่งขึ้นมาได้ คุณสมบัติเหล่านี้ทำให้มีความเป็นไปได้ที่จะแจกจ่ายคีย์สาธารณะไปยังที่ต่างๆ ได้ วิธีการที่สำคัญได้แก่ RSA ซึ่งมีความเข้มแข็งมากเนื่องจากข้อเท็จจริงที่ว่าเป็นการยากมากหรือเป็นไปไม่ได้ที่จะแยกตัวประกอบของเลขจำนวนที่มีขนาดใหญ่มากได้

เอกสารที่มีผลตามกฎหมาย เอกสารทางการค้า และเอกสารอื่นๆ มีความจำเป็นที่จะต้องมีการลงชื่อรับรอง ได้มีการคิดค้นวิธีการต่างๆ ขึ้นมาสำหรับการลงชื่อทางดิจิตอลทั้งการใช้การเข้ารหัสแบบคีย์สมมาตรและคีย์สาธารณะ โดยทั่วไปข่าวสารที่จะได้รับการลงชื่อจะถูกสร้างเป็นข้อมูล hash โดยใช้วิธีการเช่น MD5 หรือ SHA-1 จากนั้นจึงทำการลงชื่อที่ hash แทนที่จะเป็นการลงชื่อที่ข้อความในตัวเอกสารเอง

การบริการจัดการคีย์สาธารณะสามารถทำได้โดยการใช้ใบรับรองซึ่งหมายถึงเอกสารที่บอกให้ทราบถึงบุคคลผู้เป็นเจ้าของและคีย์สาธารณะของเขา ใบรับรองจะได้รับการลงชื่อรับรองโดยผู้มีอำนาจในการรับรองที่เป็นที่เชื่อถือได้หรืออาจเป็นการรับรองโดยผู้ที่ได้รับการรับรองอีกทอดหนึ่งหรือหลายทอดก็ได้

ต้นตอของการรับรองจะต้องเป็นที่รู้จักเป็นการล่วงหน้าซึ่งโดยปกติแล้วบราวเซอร์จะรู้จักต้นตอของผู้รับรองเป็นจำนวนมาก

เครื่องมือในการเข้ารหัสข้อมูลนี้สามารถนำมาใช้ในการรักษาความปลอดภัยให้กับข้อมูลที่ส่งผ่านระบบเครือข่าย IPsec ทำงานในระดับชั้นสื่อสารควบคุมเครือข่าย ทำการเข้ารหัสแพ็กเก็ตข้อมูลที่ไหลจากโฮสตัวหนึ่งไปยังโฮสอีกตัวหนึ่ง ไฟร์วอลสามารถตรวจกรองข้อมูลที่เดินทางเข้าหรือออกจากองค์กรได้ โดยทั่วไปจะใช้โพรโตคอลและพอร์ตที่ใช้งานเป็นตัวจัดการ ระบบเครือข่ายเสมือนสามารถจำลองรูปแบบเดิมของการสร้างระบบเครือข่ายโดยใช้สายสื่อสารที่เช่ามาใช้งานในการสนับสนุนความต้องการในเรื่องการรักษาความปลอดภัย ท้ายที่สุด ระบบเครือข่ายไร้สายจำเป็นจะต้องได้รับการรักษาความปลอดภัยที่ดี ซึ่งระบบ WEP ในมาตรฐาน 802.11 ไม่สามารถสนับสนุนได้ และเป็นที่คาดหวังว่ามาตรฐาน 802.11i จะสามารถสนับสนุนการรักษาความปลอดภัยในระบบเครือข่ายไร้สายได้

เมื่อบุคคลสองคนต้องการจัดตั้งช่องการสื่อสารระหว่างกันขึ้น บุคคลทั้งสองจำเป็นจะต้องมีการตรวจสอบซึ่งกันและกันและอาจจะต้องมีการสร้าง session key ซึ่งจะถูกนำมาใช้เป็นคีย์ที่ใช้เข้ารหัสข้อมูลในระหว่างการสื่อสารนั้น มีการคิดค้นโพรโตคอลสำหรับการตรวจสอบผู้ใช้ขึ้นมามากมาย รวมทั้งวิธีการที่ให้ความเชื่อถือกับบุคคลที่สามเช่น Diffie-Hellman, Kerberos, และการเข้ารหัสโดยใช้คีย์สาธารณะ

การรักษาความปลอดภัยอีเมลล์สามารถทำได้โดยใช้วิธีการผสมหลายอย่าง เช่น PGP ทำการบีบอัดข้อมูล จากนั้นก็เข้ารหัสโดยใช้ IDEA ส่วน IDEA คีย์จะถูกเข้ารหัสอีกชั้นหนึ่งโดยใช้คีย์สาธารณะของผู้รับข่าวสาร นอกจากนี้ยังทำการสร้างข้อมูล hash ของข้อความที่จะส่งเพื่อเป็นการตรวจสอบความถูกต้องของข้อมูล

การรักษาความปลอดภัยบนเว็บเป็นอีกเรื่องหนึ่งที่มีความสำคัญมาก เริ่มต้นด้วยการตั้งชื่อที่มีความปลอดภัย DNSsec และการใช้ชื่อที่สามารถตรวจสอบตนเองได้ สนับสนุนวิธีการที่ป้องกันการหลอกลวง DNS เว็บไซต์ e-commerce ส่วนมากใช้เทคนิค SSL ในการจัดตั้งช่องสื่อสารที่ปลอดภัย ทำการตรวจสอบผู้ใช้ระหว่างผู้ให้บริการและผู้ใช้บริการ มีเทคนิคต่างๆ ที่สามารถนำมาใช้กับโมบายโค้ดได้โดยเฉพาะอย่างยิ่ง sandbox และ code signing