ความรู้เบื้องต้นเกี่ยวกับการพิสูจน์ตัวตน

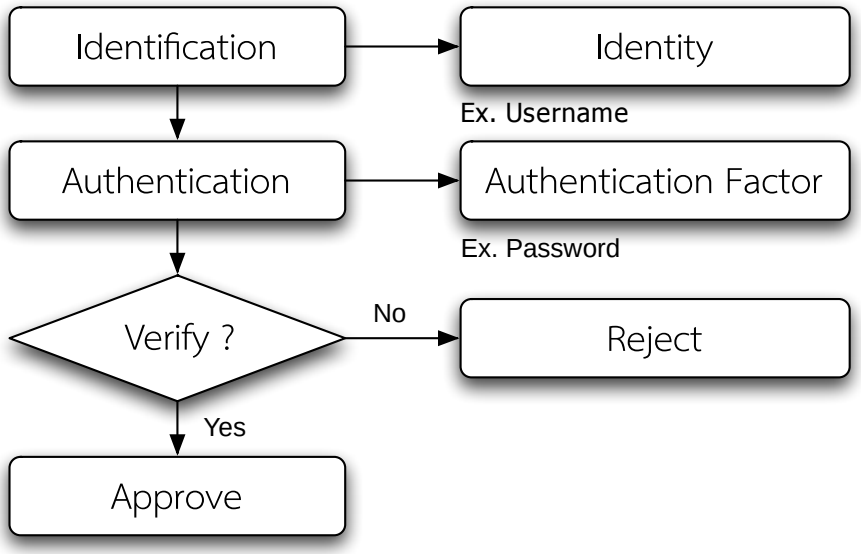
กล่าวนำ

การปกป้องความมั่นคงปลอดภัยของระบบและข้อมูลภายในองค์กรถือเป็นเรื่องสำคัญในปัจจุบัน ทั้งนี้เนื่องจากการถูกคุกคามโดยผู้ไม่ประสงค์ดีหรือจากโปรแกรมบางประเภทได้เพิ่มมากขึ้นและอาจนำมาซึ่งความเสียหายอย่างมากต่อองค์กร ดังนั้นถ้าภายในระบบมีการควบคุมความปลอดภัยที่ดีจะช่วยลดโอกาสเสี่ยงต่อการถูกคุกคามได้

การพิสูจน์ตัวตน (Authentication)

การพิสูจน์ตัวตน คือขั้นตอนการยืนยันความถูกต้องของหลักฐาน (Identity) ที่แสดงว่าเป็นบุคคลที่กล่าวอ้างจริง ในทางปฏิบัติจะแบ่งออกเป็น 2 ขั้นตอน คือ

- **การระบุตัวตน (Identification)** คือขั้นตอนที่ผู้ใช้แสดงหลักฐานว่าตนเองคือใครเช่น ชื่อผู้ใช้ (username)
- **การพิสูจน์ตัวตน (Authentication)** คือขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลที่กล่าวอ้างจริง



รูปที่ 2 แผนผังแสดงกระบวนการการพิสูจน์ตัวตน

จากแผนผังแสดงกระบวนการพิสูจน์ตัวตน ในขั้นแรกผู้ใช้จะทำการแสดงหลักฐานที่ใช้ในการพิสูจน์ตัวตนต่อระบบ ซึ่งในขั้นนี้คือการระบุตัวตน และในขั้นตอนต่อมาระบบจะทำการตรวจสอบหลักฐานที่ผู้ใช้นำมากล่าวอ้างซึ่งก็คือการพิสูจน์ตัวตน หลังจากระบบได้ทำการตรวจสอบหลักฐานเรียบร้อยแล้วถ้าหลักฐานที่นำมากล่าวอ้างถูกต้องจึงอนุญาตให้เข้าสู่ระบบได้ หากหลักฐานที่นำมากล่าวอ้างไม่ถูกต้องผู้ใช้จะถูกปฏิเสธจากระบบ

หลักฐานที่ผู้ใช้นำมากล่าวอ้างที่เกี่ยวกับเรื่องของความปลอดภัยนั้นสามารถจำแนกได้ 2 ชนิด

- **Actual identity** คือหลักฐานที่สามารถบ่งบอกได้ว่าในความเป็นจริงบุคคลที่กล่าวอ้างนั้นเป็นใคร
- **Electronic identity** คือหลักฐานทางอิเล็กทรอนิกส์ซึ่งสามารถบ่งบอกข้อมูลของบุคคลนั้นได้ แต่ละบุคคลอาจมีหลักฐานทางอิเล็กทรอนิกส์ได้มากกว่า 1 หลักฐาน ตัวอย่างเช่น บัญชีชื่อผู้ใช้

กลไกของการพิสูจน์ตัวตน (Authentication Mechanisms) สามารถแบ่งออกได้เป็น 3 คุณลักษณะคือ

- **สิ่งที่คุณมี** (Possession Factor) เช่น กุญแจหรือเครดิตการ์ด เป็นต้น
- **สิ่งที่คุณรู้** (Knowledge Factor) เช่น รหัสผ่าน (Passwords) หรือการใช้พิน (PINs) เป็นต้น
- **สิ่งที่คุณเป็น** (Biometric Factor) เช่น ลายนิ้วมือ รูปแบบเรตินา (Retinal Patterns) หรือใช้รูปแบบเสียง (Voice Patterns) เป็นต้น

กระบวนการพิสูจน์ตัวตนนั้นจะนำ 3 ลักษณะข้างต้นมาใช้ในการยืนยันหลักฐานที่นำมากล่าวอ้าง ทั้งนี้ขึ้นอยู่กับระบบ วิธีการที่นำมาใช้เพียงลักษณะอย่างใดอย่างหนึ่ง (Single-factor Authentication) นั้นมีข้อจำกัดในการใช้ ตัวอย่างเช่น สิ่งที่คุณมี (Possession Factor) นั้นอาจจะสูญหายหรือถูกขโมยได้ สิ่งที่คุณรู้ (Knowledge Factor) อาจจะถูกดักฟัง เดา หรือขโมยจากเครื่องคอมพิวเตอร์ สิ่งที่คุณเป็น (Biometric Factor) จัดได้ว่าเป็นวิธีที่มีความปลอดภัยสูงอย่างไรก็ตามการที่จะใช้เทคโนโลยีนี้ได้นั้นจำเป็นต้องมีการลงทุนที่สูง เป็นต้น

ดังนั้นจึงได้มีการนำแต่ละคุณลักษณะมาใช้ร่วมกัน (Multi-factor Authentication) ตัวอย่างเช่น ใช้สิ่งที่คุณมีกับสิ่งที่คุณรู้มาใช้ร่วมกัน เช่น การใช้ลายมือชื่อร่วมกับการใช้บัตรเครดิตหรือการใช้รหัสผ่านร่วมกับการใช้บัตร ATM เป็นต้น การนำแต่ละลักษณะของการพิสูจน์ตัวตนมาใช้ร่วมกันมากกว่า 1 ลักษณะ จะช่วยเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของข้อมูล

การกำหนดสิทธิ์ (Authorization)

การกำหนดสิทธิ์ คือขั้นตอนในการอนุญาตให้แต่ละบุคคลสามารถเข้าถึงข้อมูลหรือระบบใดได้บ้าง ก่อนอื่นต้องทราบก่อนว่าบุคคลที่กล่าวอ้างนั้นคือใครตามขั้นตอนการพิสูจน์ตัวตนและต้องให้แน่ใจด้วยว่าการพิสูจน์ตัวตนนั้นถูกต้อง

การเข้ารหัส (Encryption)

การเข้ารหัส คือการเก็บข้อมูลให้เป็นส่วนบุคคลจากบุคคลอื่นที่ไม่ได้รับอนุญาต ส่วนประกอบ 2 ส่วนที่สำคัญที่จะช่วยทำให้ข้อมูลนั้นเป็นความลับได้ก็คือ การกำหนดสิทธิ์และการพิสูจน์ตัวตนเพราะว่าก่อนการอนุญาตให้บุคคลที่กล่าวอ้างเข้าถึงข้อมูลหรือถอดรหัสข้อมูลนั้นต้องสามารถแน่ใจได้ว่าบุคคลที่กล่าวอ้างนั้นเป็นใครและได้รับอนุญาตให้สามารถเข้ามาดูข้อมูลได้หรือไม่ ในการเข้ารหัสนั้นวิธีการหนึ่งที่ทำได้คือการเข้ารหัสในรูปแบบของ

กุญแจลับ (Secret Key) ซึ่งในการใช้คีย์รูปแบบนี้ต้องเฉพาะผู้ที่มีกุญแจลับนี้เท่านั้นที่สามารถรับข้อมูลที่เข้ารหัสแล้วได้

การรักษาความสมบูรณ์ (Integrity)

การรักษาความสมบูรณ์ คือการรับรองว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงหรือทำลายไปจากต้นฉบับ (Source) ไม่ว่าจะเป็นโดยบังเอิญหรือดัดแปลงโดยเจตนาที่อาจส่งผลเสียต่อข้อมูล การคุกคามความสมบูรณ์ของข้อมูลคือการที่บุคคลที่ไม่ได้รับอนุญาตสามารถที่จะเข้าควบคุมการจัดการของข้อมูลได้

การตรวจสอบ (Audit)

การตรวจสอบ คือการตรวจสอบหลักฐานทางอิเล็กทรอนิกส์ ซึ่งสามารถใช้ในการติดตามการดำเนินการเพื่อตรวจสอบความถูกต้องและแม่นยำ ตัวอย่างเช่นการตรวจสอบบัญชีชื่อผู้ใช้ โดยผู้ตรวจบัญชี ซึ่งการตรวจสอบความถูกต้องของการดำเนินการเพื่อให้แน่ใจว่าหลักฐานทางอิเล็กทรอนิกส์นั้นได้ถูกสร้างและสั่งให้ทำงานโดยบุคคลที่ได้รับอนุญาต และในการเชื่อมต่อเหตุการณ์เข้ากับบุคคลจะต้องทำการตรวจสอบหลักฐานของบุคคลนั้นด้วย ซึ่งถือเป็นหลักการพื้นฐานของขั้นตอนการทำงานของการพิสูจน์ตัวตนด้วย

การพิสูจน์ตัวตนจัดเป็นการตรวจสอบหลักฐานขั้นพื้นฐานที่สำคัญที่สุดใน 5 ระดับชั้นของการควบคุมความปลอดภัย ดังนั้นการพิสูจน์ตัวตนดีจะช่วยเพิ่มความมั่นคงปลอดภัยขั้นพื้นฐานให้กับระบบมากยิ่งขึ้น

ประเภทของการพิสูจน์ตัวตน (Authentication type)

ส่วนประกอบพื้นฐานของการพิสูจน์ตัวตนสมบูรณ์แบ่งได้เป็น 3 ส่วน คือ

- การพิสูจน์ตัวตน (Authentication) คือส่วนที่สำคัญที่สุดเพราะเป็นขั้นตอนแรกของการเข้าใช้ระบบ ผู้เข้าใช้ระบบต้องถูกยอมรับจากระบบว่าสามารถเข้าสู่ระบบได้ การพิสูจน์ตัวตนป็นการตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลนั้นจริง
- การกำหนดสิทธิ์ (Authorization) คือข้อจำกัดของบุคคลที่เข้ามาในระบบ ว่าบุคคลคนนั้นสามารถทำอะไรกับระบบได้บ้าง
- การบันทึกการใช้งาน (Accountability) คือการบันทึกรายละเอียดของการใช้ระบบและรวมถึงข้อมูลต่างๆที่ผู้ใช้กระทำลงไปในระบบ เพื่อผู้ตรวจสอบจะได้ตรวจสอบได้ว่า ผู้ใช้ที่เข้ามาใช้บริการได้เปลี่ยนแปลงหรือแก้ไขข้อมูลในส่วนใดบ้าง

จากที่ได้กล่าวไปข้างต้นว่าการพิสูจน์ตัวตนมีความสำคัญที่สุดกับการเข้าใช้ระบบ จึงแจกแจงชนิดของการพิสูจน์ตัวตนใช้กันอยู่ในปัจจุบันว่ามีอะไรบ้างและแต่ละชนิดมีลักษณะอย่างไร ดังนี้

A. ไม่มีการพิสูจน์ตัวตน (No Authentication)

ตามหลักการแล้วการพิสูจน์ตัวตนไม่มีความจำเป็น ถ้าเงื่อนไขต่อไปนี้เป็นจริง

- ข้อมูลเหล่านั้นเป็นข้อมูลสาธารณะ ที่อนุญาตให้ทุกคนเข้าใช้บริการและเปลี่ยนแปลงได้ **หรือ**
- ข้อมูลข่าวสารหรือแหล่งของข้อมูลนั้นๆ สามารถเข้าถึงได้เฉพาะบุคคลที่ได้รับอนุญาตเท่านั้น

B. การพิสูจน์ตัวตนโดยใช้รหัสผ่าน (Authentication by Passwords)

รหัสผ่านเป็นวิธีการที่ใช้มานานและนิยมใช้กันแพร่หลาย รหัสผ่านควรจำกัดให้เฉพาะผู้ใช้ที่มีสิทธิเท่านั้นที่ทราบ

แต่ว่าในปัจจุบันนี้ การใช้แค่รหัสผ่านไม่มีประสิทธิภาพมากพอที่จะรักษาความมั่นคงปลอดภัยให้กับระบบคอมพิวเตอร์หรือระบบเครือข่ายคอมพิวเตอร์ เนื่องจากการตั้งรหัสผ่านที่ง่ายเกินไป และวิทยาการและความรู้ที่ก้าวหน้าทำให้รหัสผ่านอาจจะถูกขโมยโดยระหว่างการสือสารผ่านเครือข่ายได้

C. การพิสูจน์ตัวตนโดยใช้ PIN (Authentication by PIN)

PIN (Personal Identification Number) เป็นรหัสลับส่วนบุคคลที่ใช้เป็นรหัสผ่านเพื่อเข้าสู่ระบบ ซึ่ง PIN ใช้อย่างแพร่หลายโดยเฉพาะการทำธุรกรรมทางด้านธนาคาร เช่นบัตร ATM และเครดิตการ์ดต่างๆ

การใช้ PIN ทำให้มีความปลอดภัยในการสื่อสารข้ามระบบเครือข่ายสาธารณะมากขึ้น เนื่องจาก PIN จะถูกเข้ารหัสเอาไว้และจำเป็นต้องมีเครื่องมือที่สามารถถอดรหัสนี้ออกมาได้ เช่นฮาร์ดแวร์ที่ออกแบบมาโดยเฉพาะและถูกติดตั้งไว้ในเครื่องของผู้รับและผู้ส่งเท่านั้น

D. การพิสูจน์ตัวตนโดยใช้ Password Authenticators หรือ Tokens (Authentication by Password Authenticators or Tokens)

Authenticator หรือ Token เป็นฮาร์ดแวร์พิเศษที่ใช้สร้าง "รหัสผ่านซึ่งเปลี่ยนแปลงได้ (Dynamic Password)" ในขณะที่กำลังเข้าสู่ระบบเครือข่าย มี 2 วิธี คือ ซิงโครนัส และ อะซิงโครนัส

- การพิสูจน์ตัวตนแบบซิงโครนัส แบ่งออกเป็น 2 ประเภทตามลักษณะของการใช้งาน คือ

  การพิสูจน์ตัวตนแบบซิงโครนัสโดยขึ้นอยู่กับสถานการณ์ (Event-synchronous authentication) เมื่อผู้ใช้ต้องการที่จะเข้าสู่ระบบ ผู้ใช้จะต้องกด Token เพื่อให้ Token สร้างรหัสผ่านให้ จากนั้นผู้ใช้นำรหัสผ่านที่แสดงหลังจากกด Token ใส่ลงในฟอร์ม เพื่อเข้าสู่ระบบ ระบบจะทำการตรวจสอบกับเซิร์ฟเวอร์ก่อน ว่ารหัสผ่านที่ใส่มีอยู่ในเซิร์ฟเวอร์จริง จึงจะยินยอมให้ผู้ใช้เข้าสู่ระบบ

  การพิสูจน์ตัวตนแบบซิงโครนัสโดยขึ้นอยู่กับเวลา (Time-synchronous authentication) เป็นวิธีการที่สร้างรหัสผ่านโดยมีการกำหนดช่วงระยะเวลาการใช้งาน โดยปกติแล้วรหัสผ่านจะถูกเปลี่ยนทุกๆ หนึ่งนาที การสร้างรหัสผ่านจะเป็นไปอย่างต่อเนื่อง ทำให้บางครั้งรหัสผ่านที่สร้างออกมาอาจจะซ้ำกันกับรหัสผ่านตัวอื่นที่เคยสร้างมาแล้วก็ได้ เมื่อผู้ใช้ต้องการเข้าสู่ระบบก็ใส่รหัสผ่านและเวลาที่รหัสผ่านตัวนั้นถูกสร้างขึ้นมา (รหัสผ่านจะถูกสร้างขึ้นมาจาก Token) ลงในฟอร์ม เพื่อเข้าสู่ระบบ ระบบจะทำการตรวจสอบเวลาและรหัสผ่านที่ผู้ใช้ใส่ลงไป กับเซิร์ฟเวอร์ว่ารหัสผ่านที่ใส่ตรงกับเวลาที่ Token สร้าง และมีอยู่ในเซิร์ฟเวอร์จริง จึงยินยอมให้ผู้ใช้เข้าสู่ระบบ

- การพิสูจน์ตัวตนแบบอะซิงโครนัส หรือเรียกอีกอย่างหนึ่งว่า "Challenge-Response" ถูกพัฒนาขึ้น เป็นลำดับแรกๆ ของระบบการใช้ "รหัสผ่านซึ่งเปลี่ยนแปลงได้" ซึ่งถือได้ว่าเป็นการป้องกันการจู่โจมที่ปลอดภัยที่สุด เพราะเนื่องจากว่าเมื่อผู้ใช้ต้องการจะเข้าสู่ระบบ ผู้ใช้จะต้องทำการร้องของไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์ก็จะส่ง Challenge String มาให้ผู้ใช้ เพื่อให้ผู้ใช้ใส่ลงใน Token ที่ผู้ใช้ถืออยู่ จากนั้น Token จะทำการคำนวณรหัสผ่านออกมาให้ผู้ใช้ ผู้ใช้จึงสามารถนำรหัสผ่านนั้นใส่ลงในฟอร์มเพื่อเข้าสู่ระบบได้

การพิสูจน์ตัวตนแบบซิงโครนัสทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะมีรหัสผ่านเก็บเอาไว้ แต่แบบอะซิงโครนัสไคลเอ็นต์จะต้องติดต่อเซิร์ฟเวอร์ก่อน ก่อนจะได้รับรหัสผ่านจริง ทำให้การพิสูจน์ตัวตนแบบอะซิงโครนัสมีขั้นตอนที่ซับซ้อนกว่าแบบซิงโคนัส

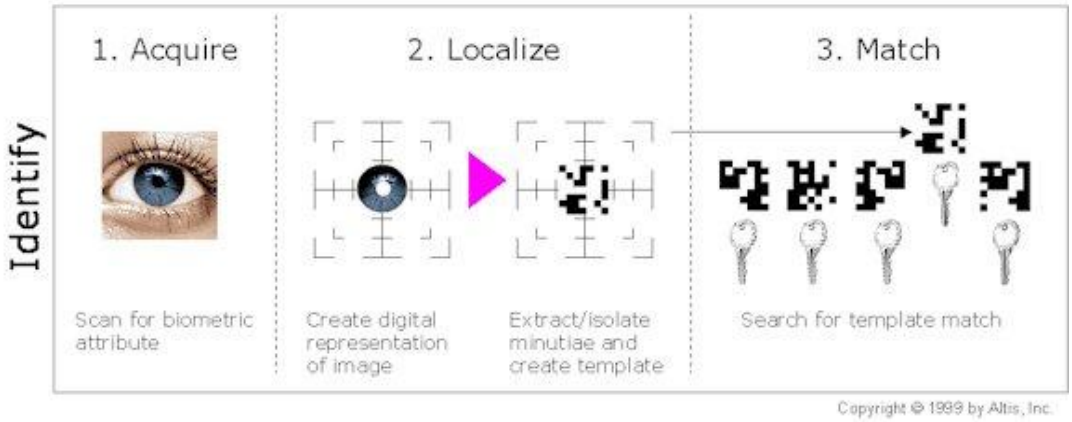
ตัวอย่างของฮาร์ดแวร์พิเศษที่ใช้ในการสร้างรหัสผ่านซึ่งเปลี่ยนแปลงได้ ของการพิสูจน์ตัวตนโดยใช้ Password Authenticator หรือ Token

รูปที่ 3 Authenticators หรือ Token ที่ใช้สำหรับสร้างรหัสผ่านซึ่งเปลี่ยนแปลงได้
ที่มา: http://www.securecomputing.com/index.cfm?sKey=665

E. การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล (Authentication by Biometric traits)

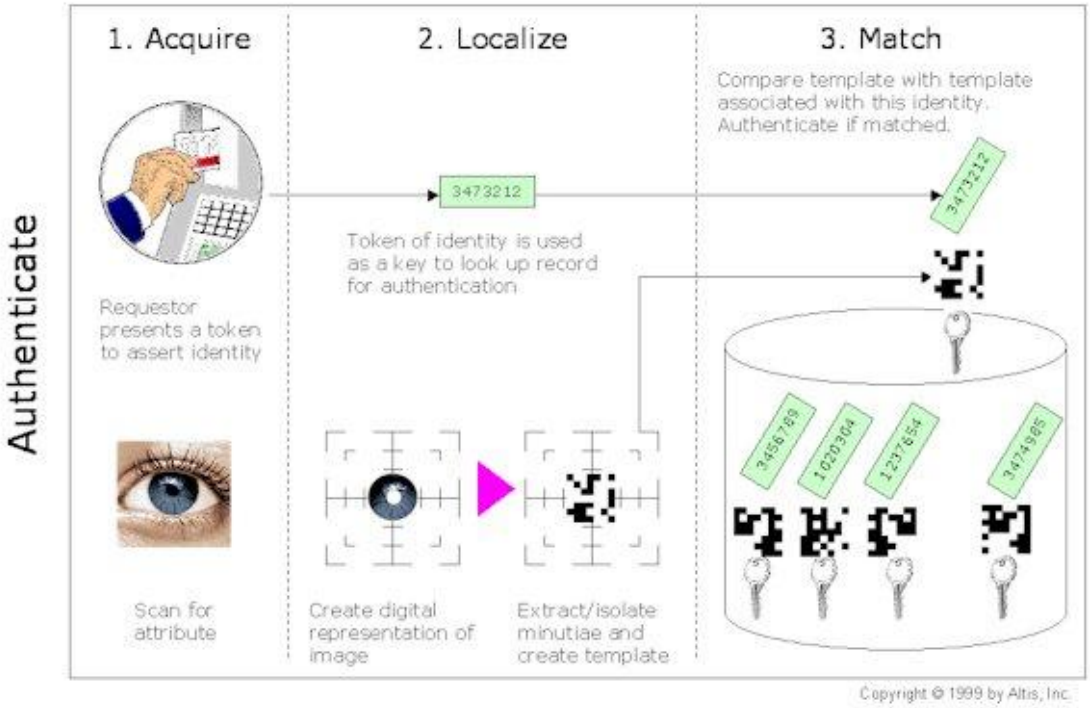
ลักษณะทางชีวภาพของแต่ละบุคคลเป็นลักษณะเฉพาะและลอกเลียนแบบกันไม่ได้ การนำมาใช้ในการพิสูจน์ตัวตนจะเพิ่มความน่าเชื่อถือได้มากขึ้นเช่นการใช้ลายนิ้วมือ เสียง ม่านตา เป็นต้น จึงมีการนำเทคโนโลยีนี้มาช่วยในการพิสูจน์ตัวตน เพื่อเพิ่มความปลอดภัยก่อนเข้าสู่ระบบ เช่นการใช้ควบคู่กับการใช้รหัสผ่าน

ตัวอย่างการใช้งานของการพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพร่วมกับการใช้ Token การ์ด หรือสมาร์ทการ์ด



รูปที่ 4 ขั้นตอนของการเก็บหลักฐานทางชีวภาพ
ที่มา: http://www.altisinc.com/Biometric/techniques.html

ในขั้นตอนของการเก็บหลักฐานทางชีวภาพ จากตัวอย่างของรูปที่ 4 ในขั้นแรกระบบจะทำการเก็บภาพของเรตินาจากบุคคลที่ถือ Token การ์ดหรือสมาร์ทการ์ด จากนั้นจะนำภาพเรตินาที่ได้มาแยกแยะเพื่อหาลักษณะเด่นของแต่ละบุคคลเพื่อไม่ให้ซ้ำกับบุคคลอื่น แล้วเก็บไว้เป็น Template ซึ่ง Template ที่ได้จะถูกบันทึกเป็นกุญแจคู่กับรหัสผ่านที่มีอยู่ใน Token การ์ด หรือสมาร์ทการ์ดของแต่ละบุคคล



รูปที่ 5 ขั้นตอนของการตรวจสอบหลักฐานทางชีวภาพ
ที่มา: http://www.altisinc.com/Biometric/techniques.html

ในขั้นตอนของการตรวจสอบหลักฐาน ผู้ใช้ที่ถือ Token การ์ด หรือสมาร์ทการ์ด จะนำบัตรมาผ่านเครื่องอ่านบัตรและแสดงเรตินาให้เครื่องเก็บภาพ เมื่อเครื่องอ่านบัตร อ่านค่าเลขที่ได้จากบัตรแล้ว ก็จะนำไปหากุญแจ ซึ่งในขณะเดียวกันภาพเรตินาที่เครื่องเก็บไว้ได้ ก็จะนำไปแยกแยะเพื่อหาลักษณะเด่น แล้วเก็บค่าไว้เป็น Template และนำ Template ที่ได้ไปตรวจสอบกับ Template ที่เก็บไว้เพื่อหากุญแจ และนำกุญแจที่ได้มาเปรียบเทียบกันว่าตรงกันหรือไม่ ถ้าตรงกันก็แสดงว่าผู้ที่ถือบัตรกับผู้ใช้เป็นคนเดียวกัน จึงอนุญาตให้เข้าสู่ระบบได้

F. การพิสูจน์ตัวตนโดยใช้รหัสผ่านที่ใช้เพียงครั้งเดียว (One-Time Password: OTP)

One-Time Password ถูกพัฒนาขึ้นเพื่อหลีกเลี่ยงปัญหาที่เกิดจากการใช้รหัสผ่านเพียงตัวเดียวซ้ำๆกัน OTP จะทำให้ระบบมีความปลอดภัยมากขึ้น เพราะรหัสผ่านจะถูกเปลี่ยนทุกครั้งก่อนที่ผู้ใช้จะเข้าสู่ระบบ

การทำงานของ OTP คือเมื่อผู้ใช้ต้องการจะเข้าใช้ระบบ ผู้ใช้จะทำการร้องขอไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะส่ง Challenge String กลับมาให้ผู้ใช้ จากนั้นผู้ใช้จะนำ Challenge String และรหัสลับที่มีอยู่กับตัวของผู้ใช้นำไปเข้าแฮชฟังก์ชันแล้วออกมาเป็นค่า Response ผู้ใช้ก็จะส่งค่านั้นกลับไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการตรวจสอบค่าที่ผู้ใช้ส่งมาเปรียบเทียบกับค่าที่เซิร์ฟเวอร์เองคำนวณได้ โดยเซิร์ฟเวอร์ก็ใช้วิธีการคำนวณเดียวกันกับผู้ใช้ เมื่อได้ค่าที่ตรงกันเซิร์ฟเวอร์ก็จะยอมรับให้ผู้ใช้เข้าสู่ระบบ

ศึกษาเพิ่มเติมได้ในเอกสารเผยแพร่เรื่องการติดตั้งและการใช้งาน OPIE

G. การพิสูจน์ตัวตนโดยการเข้ารหัสโดยใช้กุญแจสาธารณะ (Public-Key Cryptography)

เป็นการรักษาความปลอดภัยของข้อมูลระหว่างการส่งข้ามเครือข่ายวิธีหนึ่งที่นิยมใช้กันอยู่ในปัจจุบัน การเข้ารหัสแบบคู่รหัสกุญแจนี้จะมีความปลอดภัยมากกว่าการเข้ารหัสข้อมูลแบบธรรมดา แต่ก็ไม่ได้หมายความว่าการเข้ารหัสแบบคู่รหัสกุญแจนี้จะเป็นวิธีที่เหมาะสมที่สุดของวิธีการเข้ารหัส ทั้งนี้ขึ้นอยู่กับประเภทงานของแต่ละองค์กรหรือบุคคล

การเข้ารหัสโดยใช้กุญแจสาธารณะ ประกอบไปด้วยกุญแจ 2 ชนิด ที่ต้องใช้คู่กันเสมอในการเข้ารหัสและถอดรหัสคือ

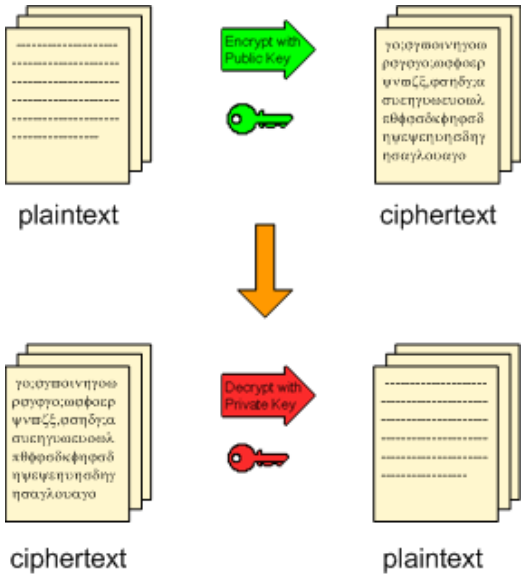
- **กุญแจสาธารณะ** (Public Key) เป็นกุญแจที่ผู้สร้างจะส่งออกไปให้ผู้ใช้อื่นๆ ทราบหรือเปิดเผยได้
- **กุญแจส่วนตัว** (Private Key) เป็นกุญแจที่ผู้สร้างจะเก็บไว้ โดยไม่เปิดเผยให้คนอื่นรู้

กระบวนการของการเข้ารหัสแบบคู่รหัสกุญแจ มีดังนี้

1. ผู้ใช้แต่ละคนจะสร้างคู่รหัสกุญแจของตัวเองขึ้นมา เพื่อใช้สำหรับการเข้ารหัสและการถอดรหัส
2. กุญแจสาธารณะจะถูกส่งออกไปยังผู้ใช้คนอื่นๆ แต่กุญแจส่วนตัวจะถูกเก็บที่ตนเอง
3. เมื่อจะส่งข้อมูลออกไปหาผู้ใช้คนใด ข้อมูลที่ส่งจะถูกเข้ารหัสด้วยกุญแจสาธารณะ ก่อนถูกส่งออกไป
4. เมื่อผู้รับได้รับข้อความแล้วจะใช้กุญแจส่วนตัวซึ่งเป็นคู่รหัสกันถอดรหัสออกมา

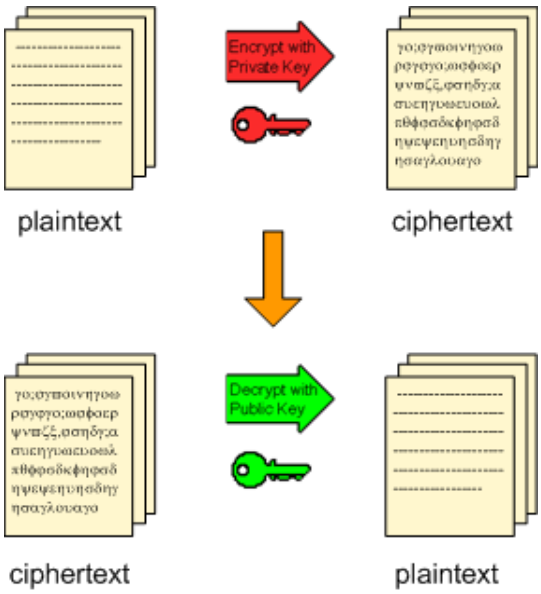
การเข้ารหัสโดยใช้กุญแจสาธารณะสามารถใช้ได้ทั้งในการเข้ารหัส (Encryption) และการพิสูจน์ตัวตน (Authentication)

การประยุกต์ใช้ในการเข้ารหัสข้อมูล (Encryption) เป็นการนำข้อมูลที่จะส่งไปยังผู้รับมาเข้ารหัสด้วยกุญแจสาธารณะของผู้รับ และเมื่อผู้รับได้รับข้อความนั้นแล้วจะถอดรหัสออกมาด้วยกุญแจส่วนตัว จึงจะเห็นได้ว่ามีเพียงผู้รับเท่านั้นที่จะสามารถถอดรหัสออกมาได้



รูปที่ 6 ระบบของการเข้ารหัสแบบใช้คู่รหัสกุญแจ

การประยุกต์ใช้ในการพิสูจน์ตัวตน (Authentication) เป็นการนำข้อมูลที่ผู้ส่งต้องการส่งมาเข้ารหัสด้วยกุญแจส่วนตัวของผู้ส่ง แล้วนำข้อมูลนั้นส่งไปยังผู้รับ ซึ่งผู้รับจะใช้กุญแจสาธารณะซึ่งเป็นคู่รหัสกันถอดรหัสออกมา ผู้รับก็สามารถรู้ได้ว่าข้อความนั้นถูกส่งมาจากผู้ส่งคนนั้นจริง ถ้าสามารถถอดรหัสข้อมูลได้อย่างถูกต้อง

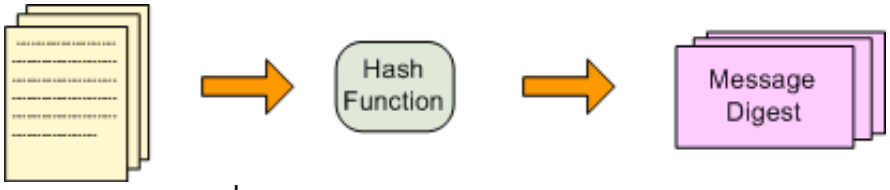


รูปที่ 7 ระบบของการเข้ารหัสแบบใช้คู่รหัสกุญแจเพื่อการพิสูจน์ตัวตน

H. การพิสูจน์ตัวตนโดยการใช้ลายเซ็นอิเล็กทรอนิคส์ (Digital Signature) เป็นการนำหลักการของ

การทำงานของระบบการเข้ารหัสแบบใช้คู่รหัสกุญแจเพื่อการพิสูจน์ตัวตนมาประยุกต์ใช้ ระบบของลายเซ็นดิจิตอลสามารถแบ่งเป็นขั้นตอนได้ดังนี้

1. เมื่อผู้ใช้ต้องการจะส่งข้อมูลไปยังผู้รับ ข้อมูลนั้นจะถูกนำไปเข้าฟังก์ชั่นทางคณิตศาสตร์ที่เรียกว่า "แฮชฟังก์ชัน" ได้เมสเซสไดเจสต์ (Message Digest) ออกมา

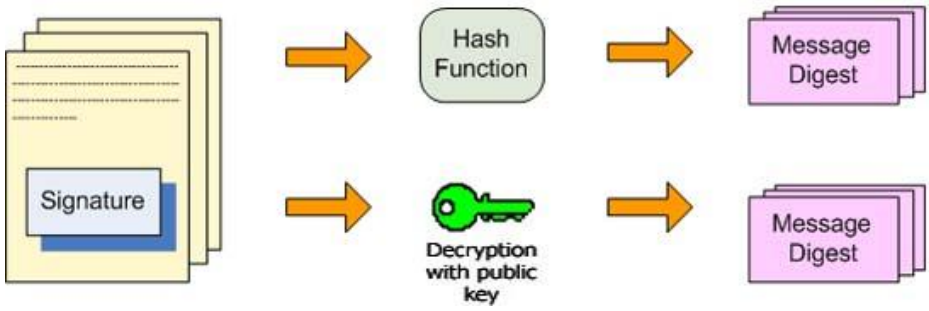


รูปที่ 8 การส่งข้อมูลเข้าไปใน Hash Function

2. การใช้กุญแจส่วนตัวเข้ารหัสข้อมูล หมายถึงว่าผู้ส่งได้ลงลายเซ็นดิจิตอล ยินยอมที่จะให้ผู้รับ สามารถทำการตรวจสอบด้วยกุญแจสาธารณะของผู้ส่งเพื่อพิสูจน์ตัวตนของผู้ส่งได้

รูปที่ 9 การเข้ารหัสเมสเซสไดเจสต์ด้วยกุญแจส่วนตัวเพื่อเป็นการลงลายเซ็น

3. การตรวจสอบข้อมูลว่าถูกส่งมาจากผู้ส่งคนนั้นจริงในด้านผู้รับ โดยการนำข้อมูลมาผ่านแฮชฟังก์ชันเพื่อคำนวณหาค่าเมสเซจไดเจสต์ และถอดรหัสลายเซ็นอิเล็กทรอนิคส์ด้วยกุญแจสาธารณะของผู้ส่ง ถ้าสามารถถอดได้อย่างถูกต้อง จะเป็นการยืนยันข้อมูลจากผู้ส่งคนนั้นจริง และถ้าข้อมูลเมสเซจไดเจสต์ที่ได้จากการถอดรหัสเท่ากันกับค่าเมสเซจไดเจสต์ในตอนต้นที่ทำการคำนวณได้ จะถือว่าข้อมูลดังกล่าวนั้นถูกต้อง



รูปที่ 10 ขั้นตอนการเปรียบเทียบความถูกต้อง

ลายเซ็นอิเล็กทรอนิคส์นิยมนำไปใช้ในระบบรักษาความปลอดภัยในการชำระเงินผ่านระบบอินเตอร์เนต ซึ่งในปัจจุบันนี้การทำธุรกรรมการเงินอิเล็กทรอนิกส์ได้รับความนิยมเป็นอย่างมาก

I. การพิสูจน์ตัวตนโดยใช้การถาม - ตอบ (Zero-Knowledge Proofs)

เป็นวิธีการพิสูจน์ตัวตนโดยใช้การถาม - ตอบ เมื่อผู้ใช้เข้ามาในระบบแล้ว ระบบจะแน่ใจได้อย่างไรว่าผู้ใช้คนนั้น เป็นคนที่ได้รับอนุญาตให้เข้ามาใช้ระบบได้จริง การใช้ชื่อผู้ใช้และรหัสผ่าน ในปัจจุบันนี้ไม่มีความปลอดภัยเพียงพอต่อการเข้าใช้ระบบ เนื่องจากความรู้และวิทยาการที่ก้าวหน้า ทำให้เกิดผู้ที่ต้องการจะเข้ามาละเมิดระบบต่างๆมีมากขึ้น ทำให้ชื่อผู้ใช้และรหัสผ่าน อาจจะถูกลักลอบดักข้อมูลระหว่างการสื่อสารกันได้

การที่จะทำให้ระบบมั่นใจได้ว่า ผู้ที่เข้าไปในระบบผู้นั้นเป็นผู้ที่ได้รับอนุญาตจริง นั่นก็คือ ระบบจะใช้การถาม - ตอบ ซึ่งคำถามและคำตอบเหล่านี้ ผู้ใช้จะเป็นคนสร้างคำถามและคำตอบขึ้นมาเอง จากนั้นจะส่งให้กับเซิร์ฟเวอร์ ซึ่งคำถาม - คำตอบที่ผู้ใช้สร้างขึ้นมา ผู้ใช้เท่านั้นจะเป็นคนที่ทราบคำตอบของแต่ละคำถามที่ถูกสร้างและเมื่อผู้ใช้คนนั้นๆเข้าสู่ระบบได้ ระบบจะถามสุ่มคำถามเหล่านั้นที่ผู้ใช้คนนั้นๆ สร้างขึ้นมา ถามผู้ใช้คนนั้นๆก่อนที่จะยอมให้เข้าใช้ระบบได้จริง การให้ใช้ระบบได้จริงจะได้รับการยินยอมก็ต่อเมื่อการตอบคำตอบที่ผู้ใช้ตอบนั้นสัมพันธ์กับคำตอบที่มีอยู่ในเซิร์ฟเวอร์ ยกตัวอย่างเช่น นาย ก. กับ นาย ข. รู้จักกันมานานละสนิทกัน นาย ก. และ นาย ข. ย่อมมีความสนิทกันเป็นส่วนตัวเมื่อนาย ก. และนาย ข. เล่น MSN กัน ต่างฝ่ายต่างจะแน่ใจได้อย่างไรว่า คนที่ตนคุยอยู่เป็นบุคคลเดียวกันกับที่ตนรู้จัก เพราะว่านาย ก. หรือ นาย ข. อาจจะทำการเข้าระบบทิ้งไว้ หรืออาจจะมีบุคคลอื่นสามารถดักจับหลักฐานและข้อมูลที่สามารถเข้าสู่ระบบของคนใดคนหนึ่งไว้ได้ แล้วทำการสวมรอยแทน นั่นก็คือการใช้คำถามและคำตอบที่มีเพียงนาย ก. และ นาย ข. เท่านั้นที่ทราบ

วิธีการพิสูจน์ตัวตนวิธีนี้ เป็นวิธีการที่ต้องใช้ความรู้ขั้นสูงในการนำมาใช้ เนื่องจากระบบจะใช้การเรียนรู้จากข้อมูลที่ได้รับ อาจจะเรียกระบบนี้ได้ว่าเป็นการนำความรู้ด้าน AI (Artificial Intelligence) มาใช้นั่นเอง

J. ตารางเปรียบเทียบข้อดีข้อเสียของการพิสูจน์ตัวตนแต่ละชนิด

การพิสูจน์ตัวตน	ข้อดี	ข้อเสีย
ไม่มีการพิสูจน์ตัวตนตน	ง่ายต่อการใช้งานและค่าใช้จ่ายต่ำ	ความปลอดภัยของข้อมูลจะขึ้นอยู่กับผู้ใช้ว่าจะนำข้อมูลเหล่านั้นไปใช้ในทางที่ควรหรือไม่
การพิสูจน์ตัวตนโดยใช้รหัสผ่าน	สามารถใช้ได้กับทุกระบบ	จะไม่ปลอดภัยเมื่อมีการส่งข้ามระบบเครือข่ายที่เป็นสาธารณะหรือไม่มีการเข้ารหัสข้อมูล
การพิสูจน์ตัวตนโดยใช้ PIN	- ง่ายต่อการจำและความปลอดภัยค่อนข้างดี (บัตร ATM)	- ต้องใช้ฮาร์ดแวร์เฉพาะในการอ่าน PIN

	- สามารถสื่อสารข้ามเครือข่ายสาธารณะได้อย่างปลอดภัย	- ไม่สามารถใช้กับต่างระบบกันได้ - ราคาแพง
การพิสูจน์ตัวตนโดยใช้ password authenticators หรือ tokens แบบซิงโครนัส	- มีความปลอดภัยมากกว่าการใช้การจำรหัสผ่านแบบธรรมดา - ไม่ต้องใช้เครื่องอ่านการ์ด - ผู้ที่ละเมิดเข้ามาไม่สามารถจะเข้ามาจู่โจมได้	- การใช้งานยุ่งยากกว่าแบบจำรหัสผ่าน - authenticator เป็นวัตถุจึงง่ายต่อการสูญหายและการถูกขโมยได้
การพิสูจน์ตัวตนตนโดยใช้ Password Authenticators หรือ Tokens แบบอะซิงโครนัส	- มีความปลอดภัยมากกว่าการใช้การจำรหัสผ่าน แบบธรรมดา - ไม่ต้องใช้เครื่องอ่านการ์ด - เป็นวิธีการป้องกันที่ดีที่สุดเมื่อเปรียบเทียบกับ วิธีการใช้การพิสูจน์ตัวตนโดยใช้ Password Authenticators หรือ Tokens	- การใช้งานยุ่งยากกว่าแบบจำรหัสผ่าน - Authenticator เป็นวัตถุจึงง่ายต่อการสูญหาย และการถูกขโมยได้ไม่สามารถป้องกันผู้ที่ละเมิดเข้ามาในระบบได้ - การใช้งานค่อนข้างยุ่งยากกว่าวิธีการใช้ "รหัสผ่านซึ่งเปลี่ยนแปลงได้ (Dynamic Password)" วิธีอื่นๆ
การพิสูจน์ตัวตนโดยใช้ลักษณะเฉพาะทางชีวภาพของแต่ละบุคคล	มีความปลอดภัยสูงเพราะเลียนแบบกันได้ยาก	- ระบบมีความซับซ้อนสูง - ยังไม่ได้รับความนิยมกันอย่างแพร่หลาย - ค่าใช้จ่ายสูง
การพิสูจน์ตัวตนโดยวิธี One-Time Password	ทำให้การเดาหรือขโมยรหัสผ่านเป็นไปได้ยาก	- ไม่สะดวกต่อการใช้งาน เพราะผู้ใช้ต้องจำรหัสผ่านหลายตัว - ถ้าผู้ใช้จำรหัสผ่านไม่ได้ หรือ ทำรหัสผ่านสูญหาย ก็ไม่สามารถเข้าใช้ระบบได้
การพิสูจน์ตัวตนโดยการเข้ารหัสแบบคู่รหัสกุญแจ	- การจัดการกุญแจทำได้ปลอดภัย เพราะ ใช้กุญแจในการเข้ารหัส และถอดรหัสต่างกัน - สามารถระบุผู้ใช้โดยการใช้ร่วมกับลายมือชื่อ อิเล็กทรอนิกส์	- ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การคำนวณอย่างมาก - ต้องใช้ระบบที่สนับสนุนการทำงาน
การพิสูจน์ตัวตนโดยการใช้ลายเซ็นดิจิตอล	- สามารถระบุตัวผู้ส่งได้ชัดเจน - ป้องกันข้อมูลถูกแก้ไขระหว่างการส่งได้	ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การ

	หรือสามาถตรวจสอบข้อมูลได้ว่าผ่านการแก้ไขมาหรือไม่	คำนวณอย่างมาก
การพิสูจน์ตัวตนโดยวิธี Zero-Knowledge Proofs	ความปลอดภัยค่อนข้างสูง เพราะคำถามและคำตอบจะมีเพียงผู้ใช้ และเซิร์ฟเวอร์เท่านั้นที่ทราบ	ความซับซ้อนของระบบเพิ่มขึ้นตามความฉลาดของระบบ

โพรโตคอลในการพิสูจน์ตัวตน(Authentication Protocol)

ในระบบเครือข่ายแบบเปิดหรืออินเตอร์เน็ต การพิสูจน์ตัวตนถือได้ว่าเป็นกระบวนการเริ่มต้นและมีความสำคัญที่สุดในการปกป้องเครือข่ายให้ปลอดภัย โพรโตคอลในการพิสูจน์ตัวตน คือโพรโตคอลการสื่อสารที่มีกระบวนการพิสูจน์ตัวตนรวมอยู่ในชุดโพรโตคอล

โพรโตคอลการพิสูจน์ตัวตนที่กล่าวถึงในเอกสารฉบับนี้ เน้นเฉพาะโพรโตคอลหลักที่นิยมใช้อย่างแพร่หลายบนอินเตอร์เน็ตในปัจจุบัน ประกอบไปด้วย

- Secure Socket Layer (SSL)
- Secure Shell (SSH)
- Internet Security (IPSEC)
- Kerberos

A. Secure Socket Layer (SSL)

Secure Sockets Layer (SSL) เริ่มพัฒนาโดย Netscape Communications เพื่อใช้ในโพรโตคอลระดับแอพพลิเคชันคือ Hypertext Transfer Protocol (HTTP) ซึ่งเป็นการสื่อสารผ่านเว็บให้ปลอดภัย พัฒนาในช่วงต้นของยุคการค้าอิเล็กทรอนิคส์กำลังได้รับความนิยมในโลกอินเตอร์เน็ต

SSL ทำให้เกิดการสื่อสารอย่างปลอดภัยระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ โดยการอนุญาตให้มีกระบวนการพิสูจน์ตัวตนวมกับการใช้งานลายเซ็นดิจิตอลสำหรับการรักษาความถูกต้องของข้อมูลและการเข้ารหัสข้อมูลเพื่อป้องกันความเป็นส่วนตัวระหว่างการสื่อสารข้อมูล

โพรโตคอล SSL อนุญาตให้สามารถเลือกวิธีการในการเข้ารหัส วิธีสร้างไดเจสต์ [*1] และลายเซ็นดิจิตอลได้อย่างอิสระก่อนการสื่อสารจะเริ่มต้นขึ้น ตามความต้องการของทั้งเว็บเซิร์ฟเวอร์และบราวเซอร์ ทั้งนี้เพื่อเพิ่มความยืดหยุ่นในการใช้งาน เปิดโอกาสให้ทดลองใช้วิธีการในการเข้ารหัสวิธีใหม่ รวมถึงลดปัญหาการส่งออกวิธีการเข้ารหัสไปประเทศที่ไม่อนุญาต

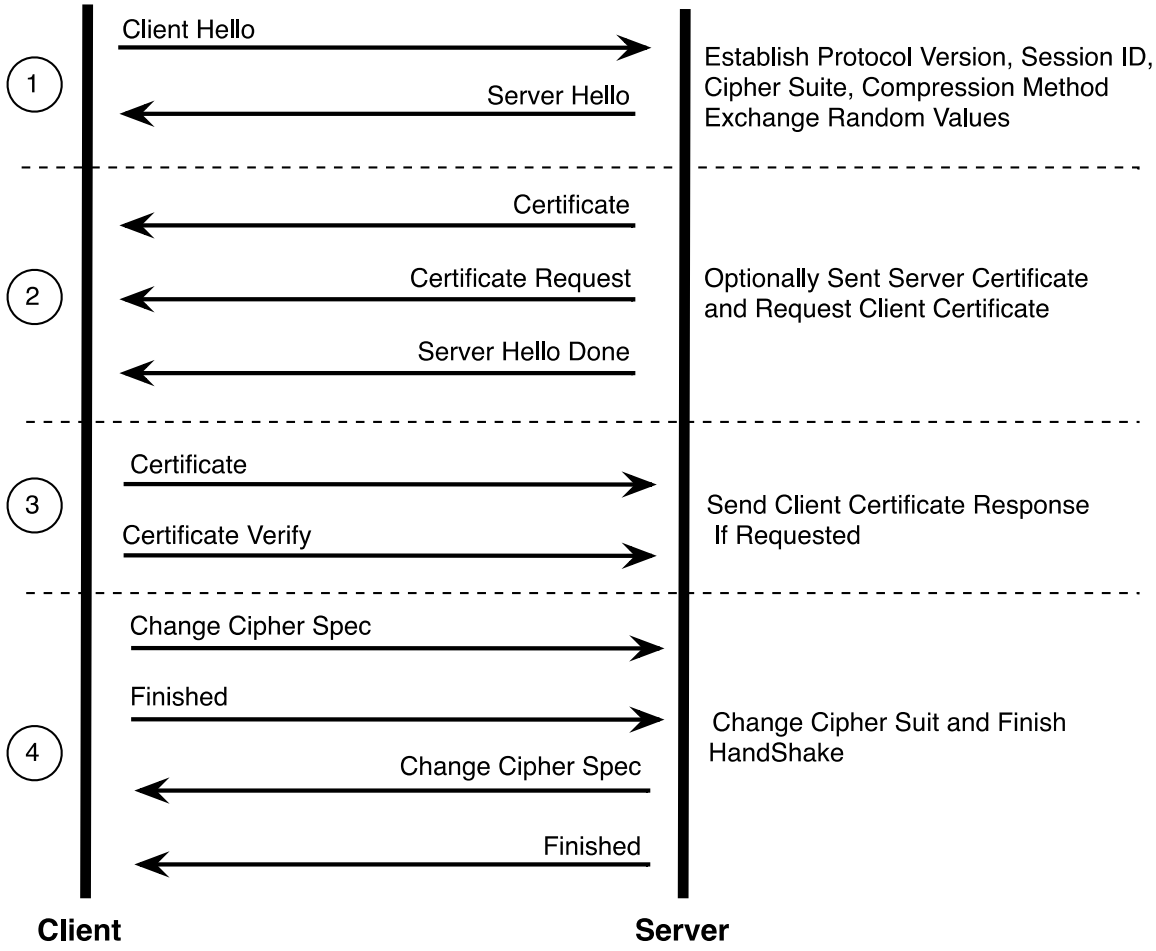
Netscape เริ่มพัฒนา SSL เวอร์ชันแรกคือเวอร์ชัน 2.0 และเวอร์ชันถัดมาเป็น 3.0 ซึ่งสนับสนุนความสามารถด้านความปลอดภัยมากขึ้น และเป็นเวอร์ชันสุดท้ายก่อนที่จะเป็นมาตรฐานกลางของโพรโตคอลบนอินเตอร์เน็ต โดยเปลี่ยนชื่อเป็น Transport Layer Security หรือ TLS ซึ่งดูแลมาตรฐานโดย Internet Engineering Task Force (IETF) อธิบายเวอร์ชันของ SSL และผู้พัฒนาได้ตามตาราง

เวอร์ชัน	ผู้พัฒนา	จุดเด่น	บราวเซอร์ที่สนับสนุน
SSL v2.0	Netscape Corp. [SSL2]	โพรโตคอล SSL รุ่นแรกที่พัฒนาบนบราวเซอร์	• NS Navigator 1.x/2.x • MS IE 3.x • Lynx/2.8 + OpenSSL
SSL v3.0	Netscape Corp. เป็น Internet Drafted รุ่นก่อนเป็นมาตรฐานกลาง [SSL3]	ปรับปรุงใหม่เพิ่มความปลอดภัยมากขึ้น สนับสนุนการใช้ non-RSA ciphers ในการเข้ารหัส และห่วงโซ่ Certificate[*2]	• NS Navigator 2.x/3.x/4.x • MS IE 3.x/4.x • Lynx/2.8 + OpenSSL
TLS v1.0	IETF กำลังเสนอให้เป็นมาตรฐานโพรโตคอลบนอินเตอร์เน็ต (Proposed Internet Standard)	ปรับปรุงจาก SSL v3.0 สนับสนุนการทำงานในชั้น MAC และ HMAC เพิ่ม Padding ชนิด Block และวิธีการจัดลำดับข้อมูล และเพิ่มระดับการแจ้งเตือน	• Lynx/2.8 + OpenSSL

- [*1] ไดเจสต์ (Digest) คือข้อความที่เกิดจากการเข้ารหัสข้อมูลด้วยฟังก์ชันแฮชเช่น MD5 หรือ SHA-1
- [*2] ห่วงโซ่ Certificate (Certificate Chain) คือการเพิ่มข้อมูล Certificate ที่เกี่ยวเนื่องกันเมื่อใช้ในขั้นตอนแลกเปลี่ยนข้อมูล ซึ่งจะช่วยลดเวลาในการค้นหา Certificate จากผู้ให้บริการ Certificate Authority (CA) ที่เกี่ยวเนื่องกันมากกว่า 1 ขึ้นไป

กระบวนการในการเริ่มต้นการสื่อสารผ่านชั้น SSL แบ่งเป็น 4 ขั้นตอนคือ

- ประกาศชุดวิธีการเข้ารหัส ไดเจสต์ และลายเซ็นดิจิตอลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์
- การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์
- การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์ ถ้าจำเป็น
- ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างไดเจสต์ และการใช้ลายเซ็นดิจิตอล



รูปที่ 11 กระบวนการเริ่มต้นการติดต่อสื่อสารของโพรโตคอล SSL

ขั้นตอน 1 : ประกาศชุดวิธีการเข้ารหัส ไดเจสต์ และลายเซ็นดิจิตอลที่สนับสนุนของทั้งไคลเอ็นต์และเซิร์ฟเวอร์

ไคลเอ็นต์และเซิร์ฟเวอร์ส่งข้อความเริ่มต้นการสื่อสาร (Hello Message) ซึ่งประกอบไปด้วยเวอร์ชันของโพรโตคอลที่ใช้ วิธีการเข้ารหัสที่เว็บเซิร์ฟเวอร์และไคลเอ็นต์สนับสนุน หมายเลขระบุการสื่อสาร (Session Identifier) รวมถึงวิธีการบีบอัดข้อมูลในการสื่อสารที่สนับสนุน

หมายเลขระบุการสื่อสารที่เกิดขึ้น ใช้สำหรับตรวจสอบการเชื่อมต่อระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ถ้ามีการเชื่อมต่อก่อนหน้านี้เกิดขึ้น แสดงว่าได้มีการตกลงวิธีการสื่อสารแล้ว สามารถเริ่มต้นส่งข้อมูลได้ทันที เป็นการลดเวลาติดต่อสื่อสารลง

ขั้นตอน 2 : การพิสูจน์ตัวตนของเซิร์ฟเวอร์ต่อไคลเอ็นต์

ถัดมาเว็บเซิร์ฟเวอร์ทำการส่ง Certificate หรือใบยืนยันความมีตัวตนของเซิร์ฟเวอร์ ไคลเอ็นต์จะทำการตรวจสอบ Certificate กับผู้ให้บริการ Certificate Authority ที่ได้ตั้งค่าไว้ เพื่อยืนยันความถูกต้องของ Certificate ของเซิร์ฟเวอร์

ขั้นตอน 3 : การพิสูจน์ตัวตนของไคลเอ็นต์ต่อเซิร์ฟเวอร์ ถ้าจำเป็น

เซิร์ฟเวอร์สามารถร้องขอ Certificate จากไคลเอ็นต์เพื่อตรวจสอบความถูกต้องของ Client ด้วยก็ได้ ใช้ในกรณีที่มีการจำกัดการใช้งานเฉพาะไคลเอ็นต์ที่ต้องการเท่านั้น ซึ่ง SSL สนับสนุนการตรวจสอบได้จากทั้งเซิร์ฟเวอร์และไคลเอ็นต์ ขึ้นอยู่กับการเลือกใช้งานในขณะติดต่อสื่อสารที่เกิดขึ้นนั้น

ขั้นตอน 4 : ไคลเอ็นต์และเซิร์ฟเวอร์ตกลงชุดวิธีการเข้ารหัส การสร้างไดเจสต์ และการใช้ลายเซ็นดิจิตอล

ขั้นตอนการตรวจสอบ Certificate ที่เซิร์ฟเวอร์ร้องขอจากไคลเอ็นต์จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับการตั้งค่าบนเซิร์ฟเวอร์ หลังจากขั้นตอนการตรวจสอบเสร็จสิ้น เซิร์ฟเวอร์และไคลเอ็นต์จะตกลงการใช้งานวิธีการเข้ารหัสระหว่างกันโดยใช้ค่าที่ได้จากการประกาศในขั้นตอนแรก

วิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส (Key exchange Method) คือการกำหนดกลไกการแลกเปลี่ยนกุญแจที่ใช้ในการเข้ารหัสระหว่างการสื่อสาร โดยทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะใช้กุญแจนี้ในการเข้ารหัสและถอดรหัสข้อมูล ใน SSL เวอร์ชัน 2.0 จะสนับสนุนวิธีการแลกเปลี่ยนกุญแจแบบ RSA ส่วน SSL เวอร์ชัน 3.0 ขึ้นไปจะสนับสนุนวิธีการอื่นๆ เพิ่มเติมเช่นการใช้ RSA ร่วมกับการใช้ Certificate หรือ Diffie-Hellman เป็นต้น

วิธีการเข้ารหัสในปัจจุบันแบ่งเป็นสองวิธีคือ การใช้กุญแจเดียวกันในการเข้ารหัสและถอดรหัส อาจเรียกกุญแจนี้ว่า Session key หรือ Secret key ส่วนอีกวิธีการคือ การใช้กุญแจคนละตัวในการเข้ารหัสและถอดรหัส ประกอบไปด้วยกุญแจสาธารณะและกุญแจส่วนตัวซึ่งเป็นคู่กันเสมอ เข้ารหัสด้วยกุญแจใด จะต้องถอดรหัสด้วยกุญแจที่คู่กันและตรงกันข้ามเท่านั้น มักใช้วิธีการเข้ารหัสด้วยกุญแจคนละตัวมาใช้ในการเข้ารหัส Session Key และส่งไปให้ฝั่งตรงข้ามก่อนการสื่อสารจะเกิดขึ้นรวมเรียกว่าวิธีการแลกเปลี่ยนกุญแจในการเข้ารหัส

SSL ใช้วิธีการเข้ารหัสด้วยกุญแจสมมาตร หรือกุญแจเดียวในการเข้ารหัสและถอดรหัส ตามที่กล่าวข้างต้น วิธีการเข้ารหัสคือ การเข้ารหัสด้วย DES และ 3DES (Data Encryption Standard), วิธีการเข้ารหัสด้วย IDEA ส่วน RC2 และ RC4 เป็นวิธีการเข้ารหัสของ RSA รวมถึงวิธีการเข้ารหัสแบบ Fortezza สำหรับความยาวของการเข้ารหัสที่ใช้คือ 40 บิต, 96 บิต และ 128 บิต

การสร้าง Message Authentication Code (MAC) เพื่อใช้สำหรับการยืนยันความถูกต้องของข้อมูลระหว่างการสื่อสารและป้องกันการปลอมข้อมูล ส่วนฟังก์ชันสร้างไดเจสต์ที่ SSL สนับสนุนและเลือกใช้ได้ในปัจจุบันคือ MD5 ขนาด 128 บิต และ SHA-1 (Secure Hash Algorithm) ขนาด 160 บิต

ซึ่งจะได้วิธีการที่ทั้งสองฝ่ายสนับสนุนและเหมาะสมซึ่งเป็นขั้นตอนสุดท้ายก่อนการสื่อสารที่มีการเข้ารหัสจะเริ่มต้นขึ้น

B. Secure Shell (SSH)

SSH เวอร์ชัน 1 พัฒนาขึ้นในปี 1995 โดย Tatu Ylonen ขณะที่เป็นนักวิจัยของมหาวิทยาลัยแห่งหนึ่งในฟินแลนด์ เพื่อแก้ปัญหาการดักจับรหัสผ่านที่เกิดขึ้นในระบบเครือข่าย และเผยแพร่ซอร์สโค้ดและเปิดให้ดาวโหลดไปใช้งานได้ฟรี ปลายปีเดียวกันได้จัดตั้งบริษัท SSH Communications Security, Ltd. (SCS) และเปิดตัว SSH เวอร์ชัน 2 ในต้นปี 1996 ในรูปของการค้า แต่ไม่สามารถทำงานร่วมกับ SSH เวอร์ชัน 1 ส่งผลให้มีการใช้งาน SSH เวอร์ชัน 1 แพร่หลายมากกว่าในเวลานั้น

เนื่องจากเหตุผลเรื่องลิขสิทธิ์ ทีมพัฒนาจากระบบปฏิบัติการ FreeBSD ได้ร่วมกันพัฒนา OpenSSH ซึ่งสนับสนุนการทำงานตามมาตรฐานของทั้ง SSH เวอร์ชัน 1 และ 2 ของ SCS และได้เปิดตัวครั้งแรกในเดือนธันวาคมปี 1999 ใน OpenSSH เวอร์ชัน 1.2.2 ซึ่งสนับสนุนเฉพาะ SSH เวอร์ชัน 1 และมาพร้อมกับระบบปฏิบัติการ OpenBSD เวอร์ชัน 2.6 และในเดือนมิถุนายนปี 2000 ได้เปิดตัว OpenSSH เวอร์ชัน 2.0 ซึ่งสนับสนุน SSH ทั้งสองเวอร์ชันและมาพร้อมกับ OpenBSD เวอร์ชัน 2.7

จากการนับสถิติการใช้งานโพรโตคอล SSH ในอินเตอร์เน็ตด้วยโปรแกรม ScanSSH ที่พัฒนาโดย Niels Provos ในเดือนเมษายนปี 2002 จากจำนวน 2.4 ล้านเครื่องในอินเตอร์เน็ตพบว่า

- มากกว่า 59% ใช้ OpenSSH เวอร์ชัน 1.99
- 17.9% ใช้ SSH เวอร์ชัน 1.5 (เป็น SSH ของบริษัท SCS)

สรุปรวมการใช้งานทั้งหมด

- รวมสรุปทุกเวอร์ชันพบว่าจำนวนการใช้งาน OpenSSH (รวมเวอร์ชัน 1.3, 1.5, 1.99 และ 2.0) ทั้งหมด 66.8%
- ส่วนจำนวนการใช้งาน SSH ทุกเวอร์ชัน (เวอร์ชัน 1.3, 1.5, 1.99 และ 2.0) มีทั้งสิ้น 28.1%

การใช้งาน SSH เป็นการติดต่อสื่อสารโดยใช้การพิสูจน์ตัวตนวมกับลายเซ็นดิจิตอล และมีการเข้ารหัสการสื่อสารตรงกันข้ามกับการสื่อสารแบบเก่าเช่น Telnet หรือ R Utilities เป็นต้นและสรุปวิธีการที่ SSH ทั้งเวอร์ชัน 1 และ 2 สนับสนุนได้ตามตาราง

	มาตรฐานตาม SSH เวอร์ชัน 1	มาตรฐานตาม SSH เวอร์ชัน 2
การเข้ารหัสแบบ Public key	RSA	DSA, DH
การสร้างไดเจสต์	MD5, CRC-32	MD5, SHA-1
การเข้ารหัสด้วยกุญแจสมมาตร	3DES, IDEA, ARCFOUR, DES	3DES, Blowfish, Twofish, CAST-128, IDEA, ARCFOUR
การบีบอีดข้อมูล	Zlib	zlib

การเข้ารหัสแบบกุญแจสาธารณะจะใช้ร่วมกับการใช้ฟังก์ชันแฮชในการสร้างไดเจสต์ สำหรับการแลกเปลี่ยน Secret key ก่อนการเข้ารหัสจะเริ่มต้นขึ้น

การเริ่มต้นการติดต่อสื่อสารตามโพรโตคอล SSH เป็นไปตามขั้นตอนสรุปได้เป็น

1. ไคลเอ็นต์เริ่มถามเวอร์ชันของโพรโตคอล SSH บนเซิร์ฟเวอร์ ถ้าใช้ SSH เวอร์ชันเดียวกันถือว่าสื่อสารกันได้
2. ไคลเอ็นต์จะประกาศวิธีการเข้ารหัส วิธีการสร้างไดเจสต์ และการแลกเปลี่ยนกุญแจในการเข้ารหัสที่สนับสนุน
3. เซิร์ฟเวอร์จะทำหน้าที่เลือกชุดวิธีการทั้งหมดที่ไคลเอ็นต์สนับสนุน
4. ไคลเอ็นต์และเซิร์ฟเวอร์เริ่มต้นแลกเปลี่ยนกุญแจในการเข้ารหัส ตามรูปแบบวิธีการแลกเปลี่ยนกุญแจด้วยวิธีการกุญแจสาธารณะเช่นการใช้วิธี Diffie-Hellman เป็นต้น
5. เมื่อแลกเปลี่ยนกุญแจสำหรับการเข้ารหัสด้วยวิธีการแลกเปลี่ยนกุญแจแล้ว ทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะสามารถเริ่มต้นติดต่อสื่อสารด้วยการเข้ารหัสด้วยกุญแจที่ได้จากการแลกเปลี่ยนกุญแจและสามารถใช้การบีบอัดข้อมูลร่วมได้

โพรโตคอล SSH ยังสนับสนุนการพิสูจน์ตัวตนของทั้งเซิร์ฟเวอร์และไคลเอ็นต์ในขั้นตอนการแลกเปลี่ยนกุญแจด้วย กล่าวคือในขั้นตอนการแลกเปลี่ยนกุญแจนั้น ทั้งไคลเอ็นต์และเซิร์ฟเวอร์จะสร้างคู่รหัสกุญแจ ประกอบไปด้วยกุญแจสาธารณะและกุญแจส่วนตัว ซึ่งกุญแจส่วนตัวของทั้งไคลเอ็นต์และเซิร์ฟเวอร์นี้เองที่ใช้ในการพิสูจน์ตัวตนได้ตามหลักการพิสูจน์ตัวตนด้วยวิธีการใช้กุญแจสาธารณะ ถ้าตรวจสอบได้ว่ามีการส่งข้อมูลด้วยกุญแจที่เปลี่ยนไปจากเดิม อาจจะแสดงได้ว่าการสื่อสารนี้ไม่ปลอดภัยแล้ว

ปัจจุบันมีซอฟต์แวร์ที่สนับสนุนการทำงานตามโพรโตคอล SSH ให้เลือกใช้มากอาทิเช่น OpenSSH จากผู้พัฒนา OpenBSD ในระบบปฏิบัติการตระกูลยูนิกซ์ ส่วนในตระกูลวินโดวส์เช่นโปรแกรม Putty ของ Simon Tatham หรือ Window SSH Secure Shell จาก www.ssh.com เป็นต้น

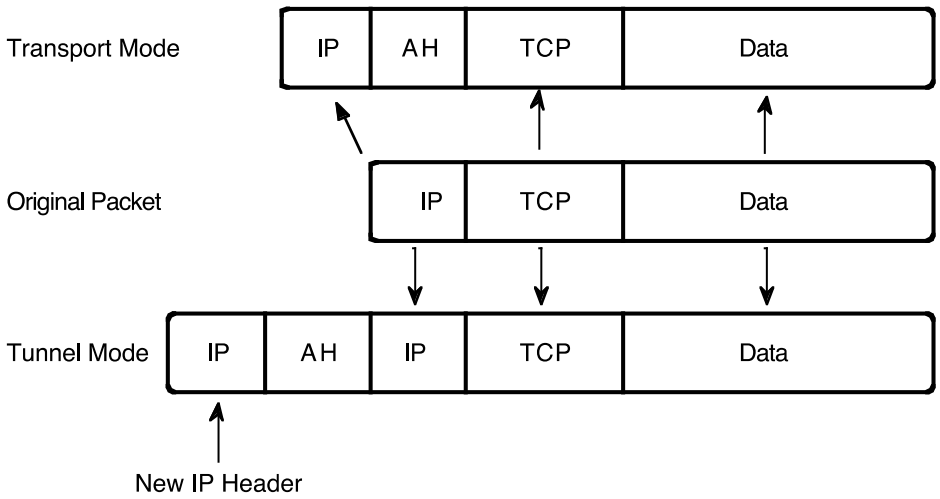
การสื่อสารด้วยโพรโตคอล SSH สนับสนุนการเข้ารหัสการสื่อสาร และการพิสูจน์ตัวตนดีในองค์กรคือการเปลี่ยนมาใช้การสื่อสารด้วย SSH แทนการสื่อสารแบบเดิมเช่นการใช้ R Utilities เช่น rlogin หรือ rcp บนตระกูลยูนิกซ์และการใช้งาน telnet และที่สำคัญคือการใช้งาน ftp ควรจะเปลี่ยนมาใช้งานโปรแกรม scp (Secure Copy) หรือ WinSCP แทนในการแลกเปลี่ยนไฟล์เป็นต้น

C. Internet Protocol Security (IPsec)

IPsec เป็นส่วนเพิ่มขยายของ Internet Protocol (IP) ในชุดโพรโตคอล TCP/IP พัฒนาเพื่อเป็นส่วนหนึ่งของมาตรฐานของ IPv6 ซึ่งเป็นโพรโตคอลที่พัฒนาเพื่อใช้แทน IPv4 ที่ใช้ในปัจจุบันและกำหนดหมายเลข RFC เป็น RFC2401

IPsec ใช้โพรโตคอล 2 ชุดคือ Authentication Header (AH) และ Encapsulated Security Payload (ESP) เพื่อรองรับการพิสูจน์ตัวตน(Authentication) การรักษาความถูกต้องของข้อมูล (Integrity) และการรักษาความลับ (Confidentiality) ในระดับชั้นของ IP

โดยการใช้งานสามารถเลือกใช้ได้สองรูปแบบตามรูป



รูปที่ 12 รูปแบบการใช้งาน IPsec

- Tunnel mode เป็นการนำส่วนแพ็กเก็ตเดิมทั้งหมดมาครอบด้วย IP โพรโตคอลชุดใหม่ที่เป็นไปตามชุดโพรโตคอล IPsec สังเกตได้จากมีการเพิ่มเฮดเดอร์ IP และ AH เข้าไปข้างหน้าแพ็กเก็ตชุดเดิม
- Transport mode นำเฉพาะข้อมูลของโพรโตคอล IP ซึ่งจะประกอบด้วยข้อมูลของชั้น Transport (TCP หรือ UDP) และชั้นแอพพลิเคชัน โดยเพิ่มโพรโตคอล AH และเพิ่มข้อมูลใน IP เดิมให้เหมาะสมตามมาตรฐาน IPsec

การรักษาความถูกต้องของข้อมูลของ IP ดาตาแกรม (IP Datagram) ในชุดโพรโตคอล IPsec ใช้ Hash Message Authentication Codes หรือ HMAC ด้วยฟังก์ชันแฮชเช่น MD5 หรือ SHA-1 ทุกครั้งที่มีการส่งแพ็กเก็ตจะมีการสร้าง HMAC และใช้การเข้ารหัสไปด้วยทุกครั้ง เพื่อให้ปลายทางสามารถตรวจสอบได้ตามหลักการลายเซ็นดิจิตอลว่าต้นทางเป็นผู้ส่งแพ็กเก็ตนั้นมาจริง

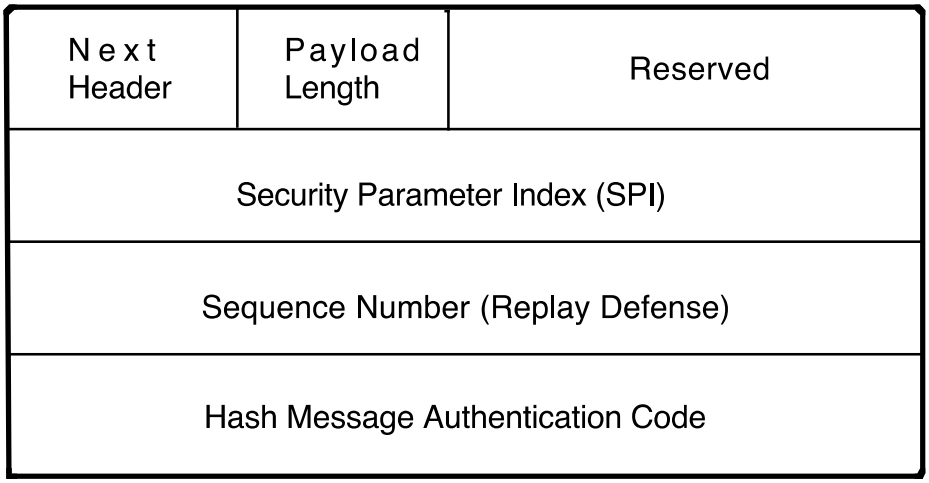
ส่วนการรักษาความลับของข้อมูลนั้น จะใช้การเข้ารหัส IP ดาตาแกรมด้วยวิธีการเข้ารหัสด้วยกุญแจสมมาตรด้วยวิธีการมาตรฐานที่เป็นรู้จักกันดีเช่น 3DES AES หรือ Blowfish เป็นต้น

ปัญหาหนึ่งของ IPsec คือการส่งกุญแจที่ใช้ในการเข้ารหัสไปกับแพ็กเก็ต ซึ่งจัดว่าไม่ปลอดภัย นอกจากนี้การแลกเปลี่ยนกุญแจนำไปสู่ปัญหาของการดูแลระบบที่ใช้ IPsec เพราะทั้งระบบต้องสนับสนุนการใช้งานโพรโตคอล IPsec เดียวกัน จะทำอย่างไรให้สามารถส่งกุญแจในการเข้ารหัสไปกับแพ็กเก็ตถ้าไม่มีการเข้ารหัสแพ็กเก็ตแต่อย่างใด เพื่อแก้ปัญหาจึงได้พัฒนาโพรโตคอลในการแลกเปลี่ยนกุญแจหรือ Internet Key Exchange Protocol (IKE)

IKE จะทำการพิสูจน์ตัวตนของปลายทางก่อนการสื่อสาร ในขั้นตอนถัดมาจึงสามารถแลกเปลี่ยนและตกลง Security Association และกุญแจในการเข้ารหัสได้ด้วยวิธีการแลกเปลี่ยนกุญแจตามวิธีการแลกเปลี่ยนกุญแจด้วยการใช้กุญแจสาธารณะเช่น Diffie-Hellmann เป็นต้น ซึ่งชุดโพรโตคอล IKE จะตรวจสอบกุญแจที่ใช้ในการเข้ารหัสระหว่างการติดต่อสื่อสารเป็นระยะตลอดการสื่อสารข้อมูลที่เกิดขึ้นแต่ละครั้ง

ชุดโพรโตคอล IPsec ประกอบจาก 2 โพรโตคอลหลักสองโพรโตคอลคือ Authentication Header (AH) และ Encapsulated Security Payload (ESP)

AH หรือ Authentication Header ทำหน้าที่รักษาความถูกต้องของ IP ดาตาแกรม โดยการคำนวณ HMAC กับทุก IP ดาตาแกรมตามรูป

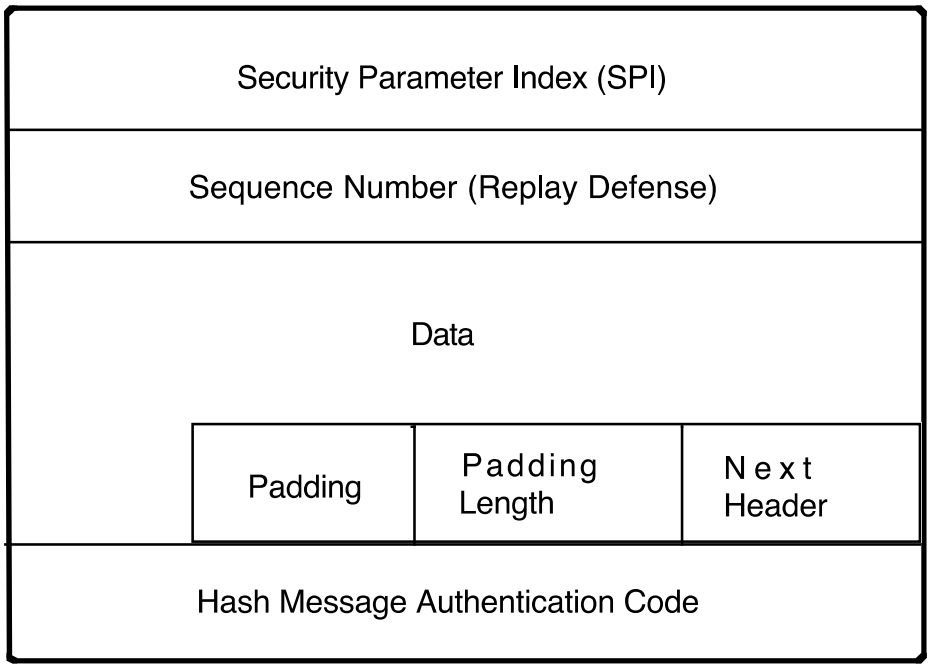


รูปที่ 13 Authentication Header

เฮดเดอร์ของ AH มีขนาด 24 ไบต์ อธิบายได้ดังนี้

- Next Header ใช้เพื่อบอกให้ทราบว่ากำลังใช้รูปแบบใดในการใช้งาน IPsec ระหว่าง Tunnel mode ค่าจะเป็น 4 ส่วน Transport mode ค่าจะเป็น 6
- Payload length บอกความยาวของข้อมูลที่ต่อท้ายเฮดเดอร์ ตามด้วย Reserved จำนวน 2 ไบต์
- Security Parameter Index (SPI) กำหนด Security Association สำหรับใช้ในการถอดรหัสแพ็กเก็ตเมื่อถึงปลายทาง
- Sequence Number ขนาด 32 บิตใช้บอกลำดับของแพ็กเก็ต
- Hash Message Authentication Code (HMAC) เป็นค่าที่เกิดจากฟังก์ชันแฮชเช่น MD5 หรือ SHA-1 เป็นต้น

ESP หรือ Encapsulated Security Payload ใช้สำหรับรักษาความถูกต้องของแพ็กเก็ตโดยใช้ HMAC และการเข้ารหัสร่วมด้วย



รูปที่ 14 Encapsulated Security Payload

- Security Parameter Index (SPI) กำหนด Security Association (SA) ระบุ ESP ที่สอดคล้องกัน
- Sequence Number ระบุลำดับของแพ็กเก็ต
- Initialization Vector (IV) ใช้ในกระบวนการเข้ารหัสข้อมูล ป้องกันไม่ให้สองแพ็กเก็ตมีการเข้ารหัสที่ซ้ำกันเกิดขึ้น
- Data คือข้อมูลที่เข้ารหัส
- Padding เป็นการเติม Data เพื่อให้ครบจำนวนไบต์ที่เข้ารหัสได้

- Padding Length บอกความยาวของ Padding ที่เพิ่ม
- Next Header กำหนดเฮดเดอร์ถัดไป
- HMAC ค่าที่เกิดจากฟังก์ชันแฮชขนาด 96 บิต

D. Kerberos

การพิสูจน์ตัวตนแบบ Kerberos พัฒนาขึ้นโดย Massachusetts Institute of Technology หรือ MIT

ระบบ Kerberos ประกอบขึ้นจากสองส่วนหลักคือ

- Ticket ใช้สำหรับการพิสูจน์ตัวตนของผู้ใช้ในระบบ และการเข้ารหัสข้อมูล
- Authenticator ใช้ในการตรวจสอบ Ticket ว่าเป็นผู้ใช้คนเดียวกันที่ใช้ Ticket เป็นใบเบิกทางเข้าสู่ระบบ และเป็นผู้ใช้ที่ระบบสร้างให้อย่างถูกต้อง

Kerberos เซิร์ฟเวอร์ มีสองส่วนบริการในการใช้งานคือ

- Authentication service (AS) สำหรับการพิสูจน์ตัวตนของผู้ใช้กับ Kerberos เซิร์ฟเวอร์ก่อนการเข้าใช้บริการ
- Ticket Granting Service (TGS) เป็นบริการที่ออก Ticket เพื่อให้ผู้ใช้นำไปใช้กับเซิร์ฟเวอร์ที่ต้องการ

กระบวนการใช้งานระบบ Kerberos มีลำดับดังนี้

1. ผู้ใช้จะทำการพิสูจน์ตัวตนครั้งแรกกับ Authentication service ของ Kerberos ซึ่งจะได้กุญแจสมมาตร ซึ่งจะใช้ในการเข้ารหัสข้อมูลในการติดต่อสื่อสาร
2. ก่อนผู้ใช้จะเข้าไปใช้บริการใด ๆ ในระบบได้ต้องมี Ticket ก่อน ด้วยการติดต่อไปที่ Ticket Granting Service เพื่อให้ออก Ticket ที่เหมาะสมกับการเข้าไปใช้บริการบนเซิร์ฟเวอร์ในระบบได้
3. ผู้ใช้นำ Ticket สำหรับไปใช้กับการร้องขอการติดต่อการบริการจากเซิร์ฟเวอร์ในระบบ

ปัญหาสำคัญของการใช้ระบบ Kerberos คือการขยายระบบเนื่องจากเซิร์ฟเวอร์ Kerberos ต้องเก็บกุญแจของผู้ใช้ทุกคนที่เข้ามาในระบบ ถ้าระบบใหญ่มากขึ้น มีการกระจายตัวมากกว่าหนึ่งจุด ย่อมส่งผลเสียต่อการใช้งานระบบโดยรวม แต่การนำระบบ Kerberos มาใช้จะเพิ่มความสะดวกในการพิสูจน์ตัวตนได้มากขึ้น มักเรียกการใช้งาน Kerberos ว่าเป็นระบบ Single Sign-On แบบหนึ่ง คือการเข้าถึงการใช้บริการของระบบทั้งหมดได้ด้วยการพิสูจน์ตัวตนยงครั้งเดียว

ศึกษาการพิสูจน์ตัวตนแบบ Kerberos เพิ่มเติมได้ในเอกสารเผยแพร่เรื่องการติดตั้งและใช้งาน Kerberos

บทสรุป

การรักษาความมั่นคงปลอดภัยของระบบคอมพิวเตอร์ หรือ ระบบเครือข่ายคอมพิวเตอร์เป็นสิ่งที่ควรตระหนักเป็นอย่างยิ่งในปัจจุบัน เพราะโลกในยุคปัจจุบันเป็นโลกแห่งข้อมูลข่าวสาร การเก็บรักษาข้อมูลให้ปลอดภัยจึงเป็นสิ่งสำคัญกับตัวบุคคลและองค์กร เพราะฉะนั้นการที่จะอนุญาตให้บุคคลใดบุคคลหนึ่งสามารถเข้าถึงข้อมูลจึงเป็นสิ่งที่ควรระมัดระวัง เพราะข้อมูลบางอย่างของบุคคลและองค์กรมีความสำคัญและไม่สามารถเปิดเผยต่อบุคคลภายนอกได้

การพิสูจน์ตัวตนจึงมีความสำคัญ เนื่องจากว่าการที่บุคคลใดบุคคลหนึ่งจะเข้าสู่ระบบได้ จะต้องได้รับการยอมรับว่าได้รับอนุญาตจริง การตรวจสอบหลักฐานจึงเป็นขั้นตอนแรกก่อนอนุญาตให้เข้าสู่ระบบ การยืนยันตัวตนยิ่งมีความซับซ้อนมาก นั่นก็หมายถึงว่าความปลอดภัยของข้อมูลก็มีมากขึ้นด้วย