การออกแบบโพรโทคอลการเข้ารหัส

Designing Crytographic Protocols

ศิรปัฐช์ บุญครอง*

บทคัดย่อ

โพรโทคอลการเข้ารหัส (Cryptographic Protocol) มีบทบาทสำคัญยิ่งในโลกแห่งการสื่อสารข้อมูลปัจจุบัน ดังนั้นผู้ออกแบบโพรโทคอลการเข้ารหัสจึงมีความจำเป็นที่จะต้องรู้ถึงหลักการหรือแนวทางที่ถูกต้อง หลักการที่แนะนำในบทความนี้จะชี้ให้ผู้ออกแบบโพรโทคอลการเข้ารหัสเห็นถึงความสำคัญของเรื่องต่างๆ เช่น ความหมายของข้อความที่ถูกรับส่งในโพรโทคอล ความจำเป็นและความถูกต้องในการใช้กระบวนการการเข้ารหัส (Encryption) และความจำเป็นที่ข้อความต้องมีความสดใหม่ (Freshness) หลักการต่างๆเหล่านี้เป็นสิ่งที่ผู้ออกแบบโพรโทคอลการเข้ารหัสพึงต้องรู้และเข้าใจ เพื่อที่จะได้มาซึ่งโพรโทคอลการเข้ารหัสที่มีช่องโหว่ในการโจมตีน้อยลง และมีความปลอดภัยมากยิ่งขึ้น

คำสำคัญ: โพรโทคอลการเข้ารหัส การออกแบบโพรโทคอลการเข้ารหัส

Abstract

Cryptographic protocols have become a very essential part in communications today. Therefore, it is necessary for cryptographic protocol designers to be aware of some design principles. These principles include such things as the meanings of the exchanged messages, the necessities and correctness of encryption and the freshness of the protocol messages. They are just some examples that the designers should learn and understand, so that better and, most importantly, more secure cryptographic protocols can be achieved.

Keyword: Cryptographic Protocol, Cryptographic Protocol, designers.

1. บทนำ

เทคโนโลยีสารสนเทศในปัจจุบันนั้นอำนวยความสะดวกให้แก่ผู้ใช้มากมาย ทั้งที่อยู่ในรูปแบบของคอมพิวเตอร์แบบเครื่องเดียว (Stand Alone Computer) หรือในรูปแบบของเครือข่าย (Network) การใช้งานระบบสารสนเทศนั้น รวมหลายๆ ด้านเข้าด้วยกัน ไม่ว่าจะเป็นการศึกษา บันเทิง รวมถึงการสื่อสาร ไม่ว่าจะเป็นการใช้งานในด้านใด สิ่งสำคัญที่ผู้ใช้งานจะต้องคำนึงถึงคือ เรื่องของความปลอดภัย

การรักษาความปลอดภัยของคอมพิวเตอร์และเครือข่ายได้รับความสนใจเป็นอย่างยิ่งทั้งในวงการการศึกษาและการพาณิชย์ เนื่องจากอันตรายในเทคโนโลยีสารสนเทศนั้นสามารถเกิดขึ้นได้ตลอดเวลา ทำให้เกิดการออกแบบและพัฒนา โพรโทคอลการเข้ารหัส (Cryptographic Protocol) ขึ้นมา เช่น [1], [2], [3] เพื่อป้องกันอันตรายจากการโจมตี หรือการบุกรุก ที่อาจจะเกิดขึ้นต่อเครื่องคอมพิวเตอร์หรือเครือข่าย

แต่ทว่าโพรโทคอลการเข้ารหัสนั้น ตัวของมันเองก็ตกเป็นเป้าโจมตีได้เช่นกัน [4], [5] ทั้งนี้เนื่องจาก โพรโทคอลเหล่านั้นไม่ได้รับการออกแบบมาอย่างรอบคอบ และไม่ได้รับการพิสูจน์ที่ครอบคลุมเพียงพอ บทความนี้จะนำเสนอหลักการออกแบบโพรโทคอลการเข้ารหัส [6], [7], [8] เพื่อที่ผู้ออกแบบจะได้นำไปเป็นแนวทางในการออกแบบโพรโทคอล และจะทำให้เกิดความมั่นใจในระดับหนึ่งว่า โพรโทคอลที่ถูกออกแบบมานั้นจะมีความปลอดภัยเพิ่มมากขึ้น และมีช่องโหว่ที่จะสามารถถูกโจมตีได้น้อยลง

2. โพรโทคอลการเข้ารหัส (Cryptographic Protocol)

ความหมายของคำว่า โพรโทคอล คือ กฎที่สร้างขึ้นมาเพื่อกำกับการแลกเปลี่ยนข้อมูล ระหว่างสองเอนติตี้ (Entity) ขึ้นไป คำว่าเอนติตี้ในที่นี้ จะรวมถึง ผู้ใช้ (User) กระบวนการ (Process) และเครื่องคอมพิวเตอร์ (Machines) ดังนั้นเมื่อเราพูดถึง โพรโทคอลการเข้ารหัส หรือ Cryptographic Protocol เรา

* ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

จะหมายความว่า ข้อมูลที่ได้รับการแลกเปลี่ยนนั้นจะถูกเข้ารหัส ซึ่งจะเข้ารหัสเป็นบางข้อมูล หรือทุกๆ ข้อมูลก็ได้

การเข้ารหัสข้อมูลนั้น เป็นกระบวนการที่เป็นได้ทั้งการเข้ารหัสและถอดรหัสข้อมูลแบบสมมาตร (Symmetric Cryptography) การเข้ารหัสและถอดรหัสข้อมูลแบบอสมมาตร (Asymmetric Cryptography) และ/หรือ การใช้ลายมือชื่ออิเล็กทรอนิกส์ (Digital Signature)

ประโยชน์ของโพรโทคอลการเข้ารหัสนั้นมี 3 ข้อหลัก ๆ ได้แก่

1) เพื่อปกปิดข้อมูลที่ถูกแลกเปลี่ยนให้เป็นความลับ (Confidentiality)

ตามนิยามของโพรโทคอลการเข้ารหัส ข้อมูลบางส่วนหรือทุกส่วนจะต้องมีการเข้ารหัส ซึ่งเป้าหมายหนึ่งของการเข้ารหัสข้อมูล ก็คือ การป้องกันไม่ให้ผู้ที่ไม่เกี่ยวข้องมาอ่านข้อมูลได้ หรือการปกปิดข้อมูลให้เป็นความลับนั่นเอง

2) เพื่อช่วยในการพิสูจน์ตัวตน (Authentication)

โพรโทคอลการเข้ารหัสสามารถนำมาใช้เพื่อการพิสูจน์ตัวตน กล่าวคือ คู่สนทนาต้องการจะแน่ใจว่า ผู้สนทนาอีกฝ่ายหนึ่งเป็นเอนติตี้นั้นๆ จริงๆ หรือข้อมูลที่ได้รับนั้นมาจากอีกฝ่ายหนึ่งจริงๆ ซึ่งการพิสูจน์ตัวตนนั้น ไม่ว่าจะเป็นการพิสูจน์ตัวตนแบบทางเดียว (One-Way Authentication) หรือ การพิสูจน์ตัวตนแบบสองทาง (Mutual Authentication) โพรโทคอลการเข้ารหัสสามารถช่วยได้ทั้งนั้น เช่น SSL [1] หรือ Kerberos [3] ต่างเป็นโพรโทคอลการเข้ารหัสที่ใช้เพื่อพิสูจน์ตัวตนทั้งสิ้น

3) เพื่อช่วยในการกระจายของกุญแจเข้ารหัส (Key Distribution)

ปัญหาหลักปัญหาหนึ่งของการเข้ารหัสข้อมูลแบบสมมาตรคือ การที่คู่สนทนาจะได้มาซึ่งกุญแจเข้ารหัสตัวเดียวกันนั้น จะทำได้อย่างไร ฝ่ายหนึ่งจะสร้างกุญแจเข้ารหัสขึ้นมาแล้วส่งไปให้อีกฝ่ายหนึ่งได้หรือไม่ หรือ ทั้งสองฝ่ายจะช่วยกันสร้างกุญแจเข้ารหัสขึ้นมาหนึ่งตัว ไม่ว่าจะกรณีใดก็แล้วแต่ คำถามคือ ทำอย่างไรจึงจะปลอดภัย และโพรโทคอลการเข้ารหัสนั่นเองที่เป็นคำตอบ และเป็นตัวช่วยให้การกระจายของกุญแจเข้ารหัสนั้น มีความปลอดภัยมากขึ้น

จากประโยชน์ดังกล่าวข้างต้น จะเห็นได้ว่า การทำงานของโพรโทคอลการเข้ารหัสนั้นมีการรับส่งข้อมูลที่เกี่ยวกับความปลอดภัย เช่น ข้อมูลกุญแจเข้ารหัส หรือ ข้อมูลการพิสูจน์ตัวตน ดังนั้นการออกแบบโพรโทคอลการเข้ารหัส จึงจำเป็นต้องทำด้วยความรอบคอบ ถูกต้อง และมีแบบแผนหรือหลักการที่ชัดเจน

3. หลักการการออกแบบโพรโทคอลการเข้ารหัส (Design Principles)

หัวข้อนี้จะอธิบายถึงหลักการง่ายๆ ที่ผู้ออกแบบโพรโทคอลควรคำนึงถึงเวลาออกแบบโพรโทคอลการเข้ารหัส เพื่อที่จะได้มาซึ่งโพรโทคอลการเข้ารหัสที่มีความถูกต้อง และมีช่องโหว่ในการถูกโจมตีให้น้อยที่สุด หลักการการออกแบบโพรโทคอลการเข้ารหัสที่สำคัญมีดังต่อไปนี้ [6], [7], [8]

หลักการที่ 1 ข้อความทุกข้อความต้องสื่อความหมายให้ถูกต้องตามความต้องการจริงๆ

การแปลความหมายของแต่ละข้อความนั้น ควรจะมาจากเนื้อหาของข้อความนั้นๆเท่านั้น จริงๆ แล้วการแปลความหมายของแต่ละข้อความ ก็คือการอธิบายว่า แต่ละข้อความประกอบไปด้วยอะไรบ้าง ผู้ออกแบบโพรโทคอลสามารถอธิบายโดยใช้การเขียนบรรยาย หรือเขียนโดยใช้สัญลักษณ์ในเชิงตรรกะก็ได้เช่นกัน

ยกตัวอย่างเช่น เครื่องแม่ข่าย S ส่งข้อความ ซึ่งสามารถแปลความหมายออกมาได้ว่า “เครื่องแม่ข่าย S ส่งกุญแจเข้ารหัส K ไปให้เอนติตี้ A เพื่อที่จะใช้เข้ารหัสข้อมูลในการสื่อสารกับเอนติตี้ B”

หลักการนี้แนะนำว่า ส่วนประกอบใดๆ (Element) ที่จำเป็นสำหรับการสื่อความหมายของข้อความนี้ จะต้อง ถูกรวมเข้าไปในข้อความนี้ เพื่อที่ผู้รับข้อความจะสามารถเข้าใจความหมายได้จากการอ่านข้อความนี้ โดยไม่ต้องรอส่วนประกอบจากข้อความต่อๆ ไป

Boyd และ Mao [9] กล่าวว่า ความน่าเชื่อถือหรือความเชื่อมั่นของแต่ละข้อความที่ได้รับนั้น ขึ้นอยู่กับข้อมูลที่เป็นส่วนหนึ่งของข้อความนั้นๆ และข้อมูลที่ผู้รับข้อความนั้นได้มีอยู่แล้วเท่านั้น ดังนั้นทุกข้อความที่ส่งควรมีข้อมูลที่ถูกต้อง ครบถ้วน และเพียงพอต่อการที่จะสื่อความหมายที่ต้องการ

หลักการที่ 2 เงื่อนไขที่ทำให้ข้อความส่งผลต่อโพรโทคอลการเข้ารหัส จะต้องมีการชี้แจงอย่างชัดเจน

การที่ข้อความจะส่งผลต่อโพรโทคอลการเข้ารหัสนั้น ความเข้าใจในความหมายอย่างเดียวนั้นไม่เพียงพอ เงื่อนไขต่างๆ ที่จำเป็นต่อการทำงานของโพรโทคอล ต้องถูกต้องด้วย

การที่จะรู้ว่า เงื่อนไขที่จำเป็นนั้นถูกต้องหรือไม่ ผู้ออกแบบโพรโทคอลต้องทำการชี้แจงให้ชัดเจนว่า ข้อความนี้จะส่งผลต่อโพรโทคอลการเข้ารหัส ก็ต่อเมื่อมีเงื่อนไขนี้เท่านั้นเป็นต้น

ยกตัวอย่างเช่น ในโพรโทคอลการเข้ารหัสหนึ่ง ผู้ออกแบบโพรโทคอลกำหนดเงื่อนไขว่า กุญแจเข้ารหัสที่ใช้ในการสื่อสารระหว่าง A และ B จะต้องถูกสร้างโดยเครื่องแม่ข่าย S เท่านั้น ดังนั้น ถ้าข้อความใดก็แล้วแต่ แสดงให้เห็นว่า กุญแจเข้ารหัสถูกสร้างโดยเอนติตี้อื่นๆ ที่ไม่ใช่ S ข้อความนี้ก็จะไม่มีผลต่อโพรโทคอลนี้ เนื่องจากว่า เงื่อนไขที่กำหนดไว้ไม่ได้ถูกทำตาม

หลักการที่ 3 ถ้าชื่อของเอนติตี้มีความสำคัญต่อความหมายของข้อความ หรือความหมายของโพรโทคอลการเข้ารหัสต้องระบุชื่อของเอนติตี้นั้นๆ ให้ชัดเจนในข้อความนั้นๆ

ในโพรโทคอลการเข้ารหัสนั้น การระบุชื่อของเอนติตี้ที่เป็นส่วนหนึ่งของโพรโทคอลเป็นสิ่งจำเป็นอย่างยิ่ง ในหลายๆ กรณีการไม่ได้ระบุชื่อเอนติตี้ที่เกี่ยวข้องในข้อความสามารถทำให้โพรโทคอลการเข้ารหัสนั้นเกิดช่องโหว่และถูกโจมตีได้ โดยเฉพาะการโจมตีแบบ Man-in-the-Middle ดังตัวอย่างโพรโทคอลการพิสูจน์ตัวตนในภาพที่ 1

A	->	*B:*	$I'm\ A,\ Random_A$
B	->	*A:*	$Random_B,\ \{Random_A\}K_{ab}$
A	->	*B:*	$\{Random_B\}K_{ab}$

ภาพที่ 1 *ตัวอย่างโพรโทคอลพิสูจน์ตัวตน*

วัตถุประสงค์ของโพรโทคอลนี้ คือการพิสูจน์ตัวตนแบบสองทางของ เอนติตี้ A และ เอนติตี้ B โดยการแสดงให้อีกฝ่ายเห็นว่าตนเองก็มีกุญแจเข้ารหัส Kab เช่นกัน โดยการเข้ารหัส RandomAและ RandomB ตามลำดับ จะเห็นได้ว่าโพรโทคอลนี้ไม่มีการระบุชื่อของเอนติตี้ในข้อความที่รับส่งเลย นั่นทำให้โพรโทคอลนี้เกิดจุดอ่อนและถูกโจมตีแบบ Man-in-the-Middle ได้ ดังแสดงในภาพที่ 2

Message 1:	*C*	->	*B:*	$I'm\ A,\ Random_A$
Message 2:	*B*	->	*C:*	$Random_B,\ \{Random_A\}K_{ab}$
Message 5:	*C*	->	*B:*	$\{Random_B\}K_{ab}$
Message 3:	*C*	->	*B:*	$I'm\ A,\ Random_B$
Message 4:	*B*	->	*C:*	$Random_B,\ \{Random_B\}K_{ab}$

ภาพที่ 2 *การโจมตีแบบ Man-in-the-Middle*

การโจมตีข้างบนแสดงให้เห็นว่า ผู้โจมตีหรือ เอนติตี้ C สามารถโจมตีแบบ Man-in-the-Middle ในการโจมตีลักษณะนี้ สองข้อความแรกเป็นการทำตามขั้นตอนของกระบวนการการพิสูจน์ตัวตนตามปกติ เมื่อเอนติตี้ C ได้รับข้อความที่สอง (Message 2) แน่นอนว่าเอนติตี้ C ไม่มีกุญแจเข้ารหัส K_{ab} ที่จะเข้ารหัส RandomB ทำให้เอนติตี้ C ต้องสร้างเซสชั่น (Session) ใหม่ขึ้นมาในข้อความที่สาม (Message 3) แต่ครั้งนี้เอนติตี้ C จะส่ง RandomB ที่เค้าได้รับในข้อความที่สอง ไปให้เอนติตี้ B เข้ารหัสในข้อความที่สาม และเอนติตี้ B ก็ตอบกลับมาในข้อความที่สี่ (Message 4) ณ เวลานี้ผู้โจมตีหรือเอนติตี้ C ได้รับ RandomB ที่ถูกเข้ารหัสเรียบร้อยแล้ว ซึ่งเค้าสามารถนำไปส่งต่อได้ในข้อความที่ห้า (Message 5) ทำให้เอนติตี้ B เชื่อว่าเอนติตี้ C มีกุญแจเข้ารหัส K_{ab} ซึ่งเป็นตัวเดียวกันกับที่เค้าถืออยู่ ด้วยเหตุนี้เอนติตี้ B จึงเชื่อว่าเอนติตี้ C นั้นคือเอนติตี้ A โดยใช้จุดอ่อนของการที่ไม่ได้ระบุชื่อของเอนติตี้ที่เกี่ยวข้องในโพรโทคอลการพิสูจน์ตนนี้เป็นช่องในการโจมตี สำหรับการอุดช่องโหว่นี้ ก็คือการทำตามหลักการข้อนี้ นั่นคือการระบุชื่อของเอนติตี้ให้ชัดเจนในข้อความที่จำเป็น ดังแสดงในภาพที่ 3

A	->	*B:*	$I'm\ A,\ Random_A$
B	->	*A:*	$Random_B,\ \{\text{“}B\text{”},\ Random_A\}K_{ab}$
A	->	*B:*	$\{\text{“}B\text{”},\ Random_B\}K_{ab}$

ภาพที่ 3 *การระบุชื่อของเอนติตี้ในข้อความ*

เมื่อมีการระบุชื่อของเอนติตี้ คือ “A” และ “B” ว่าข้อความที่ถูกส่งนี้มาจากใครแล้ว เอนติตี้ C หรือผู้โจมตี ก็ไม่สามารถจะโจมตีแบบ Man-in-the-Middle ได้โดยการส่งข้อความที่ห้าในรูปแบบการโจมตีข้างบนได้อีกต่อไป เนื่องจากชื่อของผู้ส่ง (ผู้โจมตี) และชื่อที่อยู่ในข้อความนั้นไม่ตรงกัน

หลักการที่ 4 ต้องมีเหตุผลที่ชัดเจนว่าการเข้ารหัสข้อมูล (Encryption) นั้นใช้ด้วยเหตุผลอะไร

การเข้ารหัสข้อมูลนั้นมีจุดประสงค์หลายประการ ผู้ออกแบบโพรโทคอลการเข้ารหัสจะต้องแสดงเหตุผลที่ชัดเจนว่าการเข้ารหัสข้อมูลในแต่ละครั้งนั้นทำไปเพื่ออะไร จุดประสงค์ของการเข้ารหัสข้อมูลนั้น ยกตัวอย่างเช่น

4.1) เข้ารหัสข้อมูลเพื่อปกปิดข้อมูลให้เป็นความลับ ในกรณีนี้ แน่นอนว่าผู้รับข้อมูลหรือข้อความนี้จะต้องมีกุญแจถอดรหัสที่ถูกต้องจึงจะสามารถอ่านข้อมูลหรือข้อความได้

4.2) เข้ารหัสข้อมูลเพื่อพิสูจน์ว่าข้อมูลนี้เป็นข้อมูลที่เชื่อถือ

ได้ ในกรณีนี้ข้อมูลหรือข้อความอาจจะไม่จำเป็นต้องเป็นความลับก็ได้ แต่ผู้ส่งข้อความนั้นจะเข้ารหัสเพื่อพิสูจน์ให้ผู้รับรู้ว่า ข้อมูลนี้ถูกส่งมาจากเอนติตี้ที่เชื่อถือได้ กล่าวคือ ในกรณีการเข้ารหัสแบบสมมาตร กุญแจที่ใช้เข้ารหัสจะเป็นตัวเดียวกันกับที่ผู้รับใช้ถอดรหัส หรือ ในกรณีการเข้ารหัสแบบอสมมาตร ผู้ส่งสามารถเข้ารหัสโดยการใช้กุญแจส่วนตัว (Private Key) ซึ่งในที่นี้ก็คือการทำ ลายมือชื่ออิเล็กทรอนิกส์นั่นเอง

4.3) เข้ารหัสข้อมูลเพื่อสร้างตัวเลขแบบสุ่ม (Random Number) ขึ้นมา โพรโทคอลการเข้ารหัสบางโพรโทคอล โดยเฉพาะในแบบที่เรียกว่า Challenge-and-Response จำเป็นต้องสร้างตัวเลขแบบสุ่มขึ้นมา และการเข้ารหัสข้อมูลก็เป็นวิธีหนึ่งที่สามารถสร้างตัวเลขแบบสุ่มนี้ได้

ด้วยเหตุผลที่ว่า การเข้ารหัสข้อมูลนั้นมีจุดประสงค์หลายประการ และการคำนวณการเข้ารหัสข้อมูลนั้นเป็นกระบวนการที่กินทรัพยากร ผู้ออกแบบโพรโทคอลการเข้ารหัสต้องเข้าใจว่า การเข้ารหัสข้อมูลนั้นจำเป็นหรือไม่ และการเข้ารหัสข้อมูลนั้นต้องอธิบายให้ชัดเจนว่าทำไปเพื่อจุดประสงค์อะไร

มีคำกล่าวว่า “การเข้ารหัสข้อมูลนั้นไม่ได้หมายความว่าความปลอดภัย” การใช้การเข้ารหัสโดยจำเป็นหรือไม่ถูกวิธีนั้น อาจทำให้เกิดข้อผิดพลาดหรือช่องโหว่ได้เช่นกัน

หลักการที่ 5 เมื่อผู้ส่งข้อความลงลายมือชื่ออิเล็กทรอนิกส์บนข้อมูลที่ถูกเข้ารหัสมาแล้ว ผู้รับข้อความนี้ไม่ควรจะสรุปว่าผู้ส่งรู้ว่าข้อมูลที่ถูกเข้ารหัสนั้นคืออะไร

หากผู้ออกแบบโพรโทคอลการเข้ารหัสไม่ทำตามหลักการนี้แล้ว ข้อผิดพลาดหรือช่องโหว่จะเกิดขึ้นได้ ดังแสดงในภาพที่ 4

$$A \rightarrow B : A, \quad \{\{X\}_{K_{bc}}\}_{K_a^{-1}}$$

ภาพที่ 4 *ตัวอย่างข้อผิดพลาด*

ในตัวอย่างนี้ เอนติตี้ *A* ส่งข้อความไปให้เอนติตี้ B โดยข้อความมีส่วนประกอบคือ ส่วนแรก A เป็นการระบุชื่อของผู้ส่ง ส่วนที่สอง X คือข้อมูลซึ่งถูกเข้ารหัสด้วยกุญแจ K_{bc} ที่แชร์กันระหว่างเอนติตี้ B และ C และข้อมูลที่ถูกเข้ารหัสนี้ถูกลงลายมือชื่ออิเล็กทรอนิกส์โดยใช้กุญแจส่วนตัวของเอนติตี้ A ถึงแม้ว่า X เป็นข้อมูลที่ถูกลงลายมือชื่ออิเล็กทรอนิกส์โดยเอนติตี้ A ตรงนี้ก็ไม่ได้มีหลักฐานมาแสดงว่า เอนติตี้ A นั้นรู้ว่า X คืออะไร

ถ้าถามว่า ความหมายโดยนัยของหลักการนี้ หรือตัวอย่างนี้คืออะไร อยากให้ลองใช้ตัวอย่างเดียวกัน แล้วนึกถึงในกรณีที่ ข้อความนี้ถูกดักโดยผู้โจมตี ผู้โจมตีสามารถปลดลายมือชื่ออิเล็กทรอนิกส์ของเอนติตี้ A ออกได้อย่างง่าย โดยใช้กุญแจสาธารณะ (Pubic Key) ของเอนติตี้ A จากนั้นผู้โจมตีสามารถลงลายมือชื่ออิเล็กทรอนิกส์ของเขาเองแทนที่

ปัญหานี้จริง ๆ แล้วแก้ได้ง่าย โดยการลงลายมือชื่ออิเล็กทรอนิกส์บนตัวข้อมูลก่อน แล้วจึงมาเข้ารหัสอีกครั้งเพื่อปกปิดข้อมูลนี้ให้เป็นความลับ

หลักการที่ 6 ส่วนหนึ่งหรือทั้งหมดของข้อความที่รับส่งระหว่างเอนติตี้ต้องมีความสดใหม่ (Freshness)

ความสดใหม่นั้นมีความจำเป็นอย่างยิ่งในโพรโทคอลการเข้ารหัส ด้วยเหตุผลที่ว่า ความสดใหม่นั้นจะช่วยป้องกันการโจมตีแบบทำซ้ำ (Replay Attack) การโจมตีแบบทำซ้ำคือการนำเอาข้อความที่เคยถูกส่งไปแล้วก่อนหน้า มาส่งใหม่หรือส่งซ้ำ การโจมตีแบบทำซ้ำนั้นมีความเสี่ยงดังอธิบายในภาพที่ 5

A	-> *B:*	*I'm A*
B	-> *A:*	*Prove it*
A	-> *B:*	*My password is “aabbcc”*

ภาพที่ 5 *ตัวอย่างโพรโทคอลการพิสูจน์ตัวตนอย่างง่าย*

โพรโทคอลตัวอย่างในภาพที่ 5 เป็นโพรโทคอลการพิสูจน์ตัวตนอย่างง่าย กล่าวคือ เอนติตี้ A ต้องการพิสูจน์ให้เอนติตี้ B เห็นว่าตัวเค้าคือเอนติตี้ A จริง ๆ โดยการส่งรหัสผ่าน aabbcc ไปให้เอนติตี้ B ตรวจสอบ

หากมีผู้โจมตีหรือเอนติตี้ C ซึ่งต้องการจะปลอมแปลงตัวเองเป็นเอนติตี้ A คอยดักฟังข้อความเหล่านี้อยู่ เอนติตี้ C สามารถรับส่งข้อความชุดเดิมนี้กับเอนติตี้ B ได้เช่นเดียวกัน ปัญหาก็คือเอนติตี้ B ไม่สามารถรู้ได้เลยว่า ข้อความที่เค้าได้รับนั้นเป็นข้อความใหม่ หรือข้อความเก่า เนื่องจากไม่มีส่วนใดของข้อความเป็นตัวบ่งบอก นั่นหมายความว่าเอนติตี้ C จะสามารถปลอมแปลงตัวตนได้โดยง่าย ดังภาพที่ 6

C	-> *B:*	*I'm A*
B	-> *A:*	*Prove it*
C	-> *B:*	*My password is “aabbcc”*

ภาพที่ 6 *ตัวอย่างโพรโทคอลการพิสูจน์ตัวตนอย่างง่าย*

หลักการข้อนี้จึงเสนอวิธีป้องกันการโจมตีแบบทำซ้ำ โดยการรวมสิ่งที่สดใหม่ไปเป็นส่วนหนึ่งของข้อความที่รับส่งกันในโพรโทคอลการเข้ารหัส สิ่งที่สดใหม่นั้น ผู้ออกแบบ

โพรโทคอลการเข้ารหัสอาจเลือกใช้ Nonce (Number Used Once) หรือ Timestamp ก็ได้

Nonce คือตัวเลขที่ได้มาจากการสร้างแบบสุ่ม และจะเปลี่ยนแปลงทุกครั้งสำหรับทุกๆ ข้อความที่ถูกส่งออกไป Timestamp ก็คือเวลาที่ข้อความนั้นถูกสร้างขึ้นมา หรือถูกส่งออกไป และแน่นอนเวลาจะเปลี่ยนทุกครั้งที่มีการรับส่งข้อความเช่นกัน แต่การใช้ Timestamp นั้นมีข้อจำกัดที่ผู้ออกแบบโพรโทคอลการเข้ารหัสต้องคำนึงถึง นั่นคือเวลาของแต่ละเอนติตี้อาจจะไม่ตรงกัน ดังนั้นถ้าใช้ Timestamp เป็นตัวรับประกันความสดใหม่ ผู้ออกแบบโพรโทคอลการเข้ารหัสจะต้องกำหนดช่วงความแตกต่างของเวลา หรือกำหนดอายุของข้อความให้เหมาะสม

ย้อนกลับมาที่ตัวอย่าง ถ้าผู้ออกแบบโพรโทคอลทำตามหลักการข้อนี้ นั่นคือรวมสิ่งที่สดใหม่ในข้อความที่รับส่ง ผลลัพธ์ที่ได้จะเป็นดังภาพที่ 7

A	*-> B:*	*I'm A,* $Nonce_A$
B	*-> A:*	*Prove it,* $Nonce_B$
A	*-> B:*	*My password is "aabbcc",* $Nonce_{A+1}$

ภาพที่ 7 *ตัวอย่างโพรโทคอลที่ออกแบบตามหลักการที่ 6*

จะเห็นได้ว่า ผู้โจมตีจะไม่สามารถนำข้อความชุดนี้มาใช้ซ้ำได้ เนื่องจากเมื่อมีการใช้ Nonce ผู้รับข้อความจะรู้ได้ทันทีว่า ข้อความที่ได้รับนี้เป็นข้อความที่ถูกนำมาใช้ซ้ำหรือไม่ ถ้าเป็นข้อความซ้ำ ผู้รับสามารถปฏิเสธข้อความนั้นได้ ถ้าเป็นข้อความสดใหม่ ผู้รับก็สามารถดำเนินตามกระบวนการของโพรโทคอลได้ต่อไป

4. บทสรุป

โพรโทคอลการเข้ารหัสนั้นมีความจำเป็นและมีความสำคัญเป็นอย่างยิ่ง โดยเฉพาะในปัจจุบันที่เป็นโลกแห่งการสื่อสาร โพรโทคอลการเข้ารหัสเป็นโพรโทคอลที่มีประโยชน์หลายประการ ไม่ว่าจะเป็นการปิดข้อมูลให้เป็นความลับ การพิสูจน์ตัวตนของเอนติตี้ และการกระจายหรือการแจกกุญแจเข้ารหัส ดังนั้นการออกแบบโพรโทคอลการเข้ารหัสจำเป็นต้องทำด้วยความรอบคอบและมีหลักการ บทความนี้นำเสนอหลักการซึ่งผู้ออกแบบโพรโทคอลการเข้ารหัสสามารถนำไปเป็นแนวทางในการออกแบบโพรโทคอลการเข้ารหัส เพื่อที่จะทำให้โพรโทคอลที่ถูกออกแบบนั้นมีความปลอดภัยมากขึ้น และมีช่องโหว่น้อยลง ซึ่งมีตัวอย่างของโพรโทคอลที่ใช้งานจริงในปัจจุบัน ที่นำหลักการเหล่านี้ไปประยุกต์ใช้ เช่น Kerberos [3] และ X.509 [10] เป็นต้น

5. เอกสารอ้างอิง

[1] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol," *Netscape Communications Corp*., 18 Nov, 1996.

[2] T. Dierks, and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," *RFC 5246*, August, 2008.

[3] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," RFC 4120, July, 2005.

[4] Peter Burkholder, "SSL Man-in-the-Middle Attacks," SANS Institute InfoSec Reading Room, February, 2002.

[5] Bruce Schneier, "Man-in-the-Middle Attacks Against SSL," *Schneier on Security*, April, 2010.

[6] Martin Abadi, "Security Protocols and Specifications," *In Foundations of Software Science and Computation Structures: Second International Conference*, FOSSACS'99, volume 1578, pp. 1 - 13, Springer-Verlag, Berlin Germany, 1999.

[7] Martin Abadi and Roger Needham, "Prudent Engineering Practice for Cryptographic Protocols," *Technical Report 125*, Digital, Systems Research Center, 130 Lytton Avenue, Palo Alto, California 94301, June 1, 1994.

[8] Ling Dong, Kefei Chen, Mi Wen, and Yanfei Zheng, "Protocol Engineering Principles for Cryptographic Protocols Design," *In Proceedings of Eighth ACIS International Conference on Software Engineering*, Artificial Intelligence, Networking and Parallel/ Distributed Computing, pages 641-646, August, 2007.

[9] C. Boyd, and W. Mao, "On a Limitation of BAN Logic," *Advances in Cryptology: Eurocrypt '93*, Springer-Verlag, pp. 240-247, 1993.

[10] International Telecommunications Union (ITU), "X.509: Information technology-Open systems interconnection-The Directory: Public-key and attribute certificate frameworks," *ITU Recommendations*, October, 2009.