โพรโทคอลการถ่ายโอนไฟล์ด้วยรหัสผ่านอย่างมั่นคงปลอดภัย ที่มีพื้นฐานจากเซสชันคีย์ที่ใช้เพียงครั้งเดียว

A Secure Password-based File Transfer Protocol Based on One-time Session Keys

เมฆินทร์ วรศาสตร์[1] สิรภูมิ เพ็ชรโต[2] ชาลี ธรรมรัตน์[3] และ ศุภกร กังพิศดาร[4]
คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร
140 ถนนเชื่อมสัมพันธ์ เขตหนองจอก กรุงเทพฯ 10530 โทรศัพท์ 02-988-3655 ต่อ 4111
[1]maykin@webmaster.in.th, [2]dimongimiss@gmail.com, [3]chalee23@hotmail.com, [4]supakorn@mut.ac.th

Abstract

Electronic transactions have been widely adopted around the globe. A number of file transfer applications have been developed without any concern about security of transactions. To secure file transfers, a number of security protocols were proposed, but they still lack of necessary security properties. This paper introduces a password-based secure file transfer protocol based on symmetric cryptography to provide data confidentiality, integrity and party authentication. Moreover, the proposed protocol applies an offline session key generation and distribution technique to enhance security of session keys.

Keywords: File transfers, Network protocols, Security protocol, Cryptography

บทคัดย่อ

การทำธุรกรรมอิเล็กทรอนิกส์ในปัจจุบันได้รับการยอมรับ และมีการใช้งานกันมากขึ้น ทำให้เกิดแอพลิเคชันที่ถูกออกแบบมาให้สามารถใช้ในการถ่ายโอนข้อมูลผ่านทางเครือข่ายขึ้นอย่างมากมาย หากแต่โพรโทคอลสำหรับการถ่ายโอนไฟล์ผ่านทางระบบเครือข่ายยังไม่มีความมั่นคงปลอดภัยที่เพียงพอ นอกจากนี้งานวิจัยที่มีอยู่ก็ยังไม่มีกลไกด้านความมั่นคงปลอดภัยในระดับที่น่าพอใจ บทความวิจัยฉบับนี้เสนอโพรโทคอลที่ใช้ในการถ่ายโอนไฟล์ข้อมูลซึ่งให้ความมั่นคงปลอดภัยที่สูงขึ้น โดยมีการประยุกต์ใช้งานใบรับรองดิจิทัลร่วมกับวิทยาการเข้ารหัสลับแบบสมมาตร เพื่อใช้ในการรักษาความลับของข้อมูล และการพิสูจน์ตัวจริงของข้อมูลได้ รวมถึงมีการประยุกต์ใช้เทคนิคการสร้างและกระจายเซสชันคีย์แบบออฟไลน์ เพื่อเพิ่มความมั่นคงปลอดภัยในการรับส่งข้อมูลยิ่งขึ้น

คำสำคัญ การถ่ายโอนไฟล์ข้อมูล, โพรโทคอลด้านเครือข่าย, โพรโทคอลความมั่นคงปลอดภัย, วิทยาการเข้ารหัสลับ

1. บทนำ

ปัจจุบันการทำธุรกรรมต่างๆ ผ่านระบบเครือข่ายได้รับการยอมรับและใช้งานกันอย่างแพร่หลาย โดยเฉพาะอย่างยิ่งแอพลิเคชันประเภทที่ใช้เพื่อถ่ายโอนแฟ้มข้อมูลผ่านเครือข่าย (File Transfer) โพรโทคอลที่ได้รับความนิยมสูงและใช้เป็นมาตรฐาน คือ FTP (File Transfer Protocol) ซึ่งมักจะถูกนำไปพัฒนาเป็นโปรแกรมประยุกต์มากมาย แต่โพรโทคอลนี้ กลับไม่ได้ถูกออกแบบมาให้มีความมั่นคงปลอดภัยในด้านการพิสูจน์ตัวจริง (Authentication) และการรับส่งข้อมูลทำโดยไม่มีการเข้ารหัสลับซึ่งเสี่ยงต่อการถูกดักจับ (Interception) เป็นอย่างมาก

งานวิจัยจำนวนมากได้นำเสนอขึ้น เพื่อแก้ปัญหาความมั่นคงปลอดภัยในการรับส่งข้อมูล [2, 3, 4] หนึ่งในงานวิจัยที่น่าสนใจได้แก่งานของ Xia *et al.* [4] ซึ่งได้นำเสนอโพรโทคอลสำหรับการพิสูจน์ตัวจริงและโพรโทคอลการรับส่งข้อมูลด้วยความมั่งคงปลอดภัยและได้นำเสนอการป้องกันการโจมตีต่างๆ เช่น การโจมตีแบบ Replay Attacks และการโจมตีแบบปลอมตัว (Impersonation Attacks) หรือแม้แต่การเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต เป็นต้น แต่ถึงอย่างไรก็ตาม ในงานวิจัยดังกล่าวยังคงมีข้อบกพร่องและข้อจำกัดอยู่ คือ การใช้งานรหัสผ่าน (Password) ที่เหมือนเดิม ไม่มีการเปลี่ยนแปลง และข้อมูลที่เก็บในสมาร์ทการ์ด (Smart Card) ก็ไม่มีการเข้ารหัสลับ ส่วน K_{CS} ที่ใช้ในการเข้ารหัสลับระหว่างการรับส่งข้อมูลของไคลเอนท์และเซิร์ฟเวอร์นั้น ใช้ตัวเดิมตลอดเวลา ทำให้เกิดความไม่มั่นคงปลอดภัยจากการโจมตีแบบ Replay Attacks ได้

งานวิจัยฉบับนี้เสนอโพรโทคอลสำหรับการรับส่งข้อมูลอย่างมั่นคงปลอดภัย ซึ่งสามารถแก้ไขปัญหาและข้อจำกัดของงานวิจัยที่มีอยู่ [4] โดยประยุกต์เอาเทคนิคการสร้างและกระจายเซสชันคีย์ (Session key) แบบออฟไลน์ (Offline) ของ Kungpisdan *et al.* [9] มาใช้เพื่อเพิ่มความมั่นคงปลอดภัยและลดขั้นตอนในการพิสูจน์ตัวจริงโดยใช้พับบลิคคีย์ นอกจากนี้วิธีการที่นำเสนอโดยใช้รหัสผ่านในการพิสูจน์ตัวจริงยังสามารถลดการใช้งานใบรับรองดิจิทัลได้อีกด้วย

2. ทฤษฎีที่เกี่ยวข้อง

ในบทนี้เป็นการอภิปรายถึงงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่นำเสนอ โดยมีรายละเอียดดังต่อไปนี้

2.1 โพรโทคอลของ Xia *et al.*

Xia *et al*. [4] ได้นำเสนอโพรโทคอลเพื่อใช้ในการสื่อสารกันระหว่างไคลเอนท์ (Client หรือ C) และเซิร์ฟเวอร์ (Server หรือ S) มีคุณสมบัติทางด้านความมั่นคงปลอดภัย ซึ่งแบ่งการออกแบบออกเป็น 2 ส่วนด้วยกัน คือ โพรโทคอลการพิสูจน์ตัวจริง (Authentication Protocol) และโพรโทคอลการรับส่งข้อมูล (Transfer Protocol)

ในการเริ่มใช้งานระบบนั้น C จะต้องทำการลงทะเบียนกับ Authority Center เพื่อขอ $\{ID, H(PW), K_S\}$ โดยที่ ID คือ ข้อมูลของผู้ใช้ PW คือ รหัสผ่าน และ K_S เป็นคีย์ที่ใช้ร่วมกันระหว่าง C และ S ซึ่งค่าดังกล่าวถูกเก็บไว้ในบัตรสมาร์ทการ์ด (Smart Card)

2.1.1 โพรโทคอลการพิสูจน์ตัวจริง

เมื่อต้องการเริ่มใช้งานนั้น C จะต้องนำบัตรสมาร์ทการ์ดเชื่อมต่อกับเครื่องอ่าน (Smartcard Reader) โดยมีขั้นตอนการพิสูจน์ตัวจริง ดังนี้

C นำชื่อผู้ใช้ที่ได้มาผ่านฟังก์ชันแฮช (Hash function: H) และนำผลลัพธ์มาเปรียบเทียบกับค่าแฮช $H(PW)$ ที่ถูกเก็บอยู่ในบัตรสมาร์ทการ์ด หากค่าที่ได้ตรงกันแสดงว่า C เป็นผู้ที่ได้รับอนุญาตในการเข้าใช้ระบบ

C สร้างการเชื่อมต่อไปยัง S และ S ทำการสร้างค่านอนซ์ (nonce) NS_0 ส่งกลับมายัง C ดังนี้

C → S : *Connection Request*

S → C : NS_0

C คำนวณหา $f(NS_0) = (NS_0)H(PW)$ และ C จะสร้าง NC_0 แล้วคำนวณหาค่า $H_C = H(NC_0)$ พร้อมทั้งสร้างค่า K_{CS} และนำมาเข้ารหัสลับด้วย K_S แล้วจึงส่งให้ S อนึ่ง K_{CS} เป็นคีย์ที่แลกเปลี่ยนกันระหว่าง C และ S

C → S : $\{ID, f(NS_0), H_C, K_{CS}\}_{Ks}$

S คำนวณค่า $f(NS_0)$ ที่ได้มาจาก C เพื่อใช้ถอดรหัสลับด้วย K_S หลังจากนั้นนำค่า ID และ $f(NS_0)$ มาเปรียบเทียบกับค่า CS จากนั้น S นำค่า H_C มาเข้ารหัสลับด้วย K_{SC} แล้วส่งให้ C

S → C : $\{H_C\}_{Kcs}$

C นำค่า H_C มาถอดรหัสลับด้วย K_{SC} แล้วนำ H_C ที่ได้จาก S และ C มาตรวจสอบโดยใช้การเปรียบเทียบกัน หากค่าที่ได้ตรงกันแสดงว่าการพิสูจน์ตัวจริงเสร็จสมบูรณ์

2.1.2 โพรโทคอลการรับส่งข้อมูล

หลังจากที่ C ทำการพิสูจน์ตัวจริงสำเร็จแล้ว ขั้นตอนของการถ่ายโอนไฟล์มีดังนี้

A → B : $\{Z(M, H(M))\}_{Kcs}$

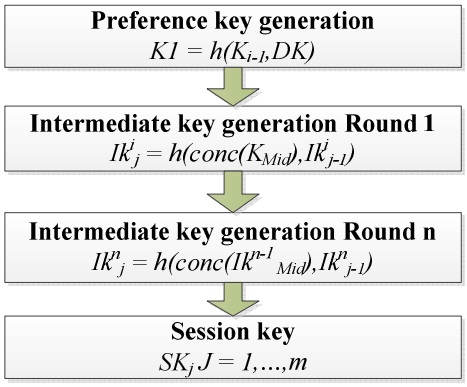
โดยที่ *{A, B}* คือ เซ็ตของผู้ที่เกี่ยวข้อง ซึ่งอาจจะเป็น C หรือ S จากข้อความข้างต้นนี้ C หรือ S นำข้อความ *M* มาผ่านฟังก์ชันแฮช ได้เป็น *H(M)* แล้วจึงนำค่า *M และ H(M)* มาทำการบีบอัด (Zip) ได้เป็น *Z(M,H(M))* และต่อจากนั้นจึงนำมาเข้ารหัสลับด้วย K_{CS} เมื่อ S ได้รับก็จะนำไปถอดรหัสลับด้วย K_{CS} แล้วมาคลายการบีบอัด (Unzip) แล้วนำค่า *M* มาผ่านฟังก์ชันแฮช แล้วนำค่าแฮชของ C มาเปรียบเทียบกัน

โพรโทคอลนี้มีข้อดี คือ สามารถป้องกันการ Replay Attack เนื่องจากการสร้างค่า N_{Si} และ N_{Ci} ในแต่ละครั้งจะไม่ซ้ำกัน นอกจากนั้นยังสามารถป้องกัน Impersonation Attack เนื่องจาก K_S นั้นถูกสร้างขึ้นในกระบวนการลงทะเบียนกับ Authority Center จึงยากต่อการปลอมเป็นเซิร์ฟเวอร์ แต่โพรโทคอลนี้ก็ยังคงมีปัญหาและข้อจำกัดอยู่ คือ *PW* และ K_S ที่เก็บอยู่ในสมาร์ทการ์ดเป็นค่าเดิมไม่มีการเปลี่ยนแปลงจึงง่ายต่อการ Brute force Attack รวมถึงการใช้ค่า K_{CS} ที่เป็นค่าเดิมตลอดการรับส่งข้อมูลนั้นง่ายต่อการถูก Replay Attack

2.2 การสร้างและกระจายเซสชันคีย์แบบออฟไลน์

จากบทที่ผ่านมาพบว่าวิทยาการเข้ารหัสลับแบบสมมาตรมีการใช้งานคีย์ร่วมกันระหว่างผู้ส่งและผู้รับ ซึ่งคีย์ดังกล่าวถูกใช้ในการเข้ารหัสลับข้อความที่สื่อสารกัน ซึ่งงานวิจัยที่มีอยู่ [4] นั้นไม่มีการเปลี่ยนค่าคีย์ทำให้ง่ายต่อการ Cryptanalysis เพื่อเป็นการแก้ไขข้อจำกัดดังกล่าว ควรมีการนำเอาเทคนิคการสร้างและกระจายคีย์ที่มีความมั่นคงปลอดภัยมาใช้งาน ในบทนี้จะกล่าวถึงเทคนิคที่จะถูกนำมาใช้ในงานวิจัยฉบับนี้

Kungpisdan *et al*. [9] ได้นำเสนอวิธีการสร้างและกระจายคีย์แบบออฟไลน์ ซึ่งมีการสร้างและกระจายเซสชันคีย์โดยที่ไม่ต้องส่งคีย์ดังกล่าวผ่านเครือข่าย [9] ซึ่งมีจุดเด่นเหนือกว่าเทคนิคการกระจายคีย์แบบออนไลน์ รายละเอียดของวิธีการดังกล่าวมีดังนี้



รูปที่ 1 Session Key Generation

การสร้างเซสชันคีย์ (Session Key Generation)

ก่อนทำการสร้างเซสชันคีย์นั้น จะต้องมีการแลกเปลี่ยนค่า $\{K_{AB}, DK, m\}$ สมมติว่า เป็นการแลกเปลี่ยนกันโดยอลิสและบ๊อบ ซึ่งต่อไปนี้จะกำหนดให้ K_{AB} เรียกว่า Long-term key, *DK* เรียกว่า Distribution key และ *m* คือค่าสุ่มที่ใช้ระบุจำนวนของคีย์ที่ต้องการสร้างขึ้น โดยขั้นตอนการสร้างเซสชันคีย์สามารถอธิบายได้ดังรูปที่ 1

หลังจากมีแลกเปลี่ยนค่า $\{K_{AB}, DK, m\}$ อลิสและบ๊อบ จะสร้างเซ็ตของ Preference keys K_i, $i = 1,2,..., m$ ดังสมการ

$$K_i = h(K_{i-1}, DK) \qquad (2.1)$$

จากนั้น ทั้งคู่สร้างเซ็ตของ Intermediate Keys (*IK*) ซึ่งเป็นการเพิ่มความยากในการวิเคราะห์การถอดรหัสลับ เพิ่มความยากในการสืบค้นของ Preference key โดยมีรูปแบบดังนี้

$$IK^x_j = h(conc(IK^{x-1}_{mid}), IK^x_{j-1}) \qquad (2.2)$$

โดย *x* เป็นจำนวนรอบ, *j* เป็นจำนวน Intermediate key ที่ถูกสร้างขึ้น, IK^{x-1}_{mid} เป็นค่าของ $\{ IK^{x-1}_{mid1}, IK^{x-1}_{mid2}, IK^{x-1}_{mid3}\}$, $conc(IK^{x-1}_{mid})$ จะเป็นการเชื่อมต่อค่า $\{IK^{x-1}_{mid1}, IK^{x-1}_{mid2}, IK^{x-1}_{mid3}\}$ ตามลำดับ การหาค่า $IK^x_{mid1} = mid(IK^x_1, IK^x_{rm})$ โดยที่ *rm* คือ จำนวนของ Intermediate key ที่ยังเหลืออยู่ในชุดข้อมูลของ IK^x_j, $IK^x_{mid2} = mid(IK^x_{mid1}, IK^x_{rm})$, $IK^x_{mid3} = mid(IK^x_1, IK^x_{mid2})$, $IK^1_{mid1} = K_{mid1}$, $IK^1_{mid2} = K_{mid2}$, $IK^1_{mid3} = K_{mid3}$ และ $IK^x_{j-1} = \phi$

การใช้ Intermediate Keys ในทุกๆ รอบ จะทำการลบค่าออกจากระบบ ส่วนที่เหลือของ Intermediate Keys ในรอบอื่นๆ สามารถเขียนได้ดังนี้

$$\{K_1, K_2, \ldots, K_{rm}\},$$
$$\{K^1_1, K^1_2, \ldots, K^1_{rm}\},$$
$$\{K^2_1, K^2_2, \ldots, K^2_{rm}\}, \ldots,$$
$$\{K^n_1, K^n_2, \ldots, K^n_{rm}\}$$

ผลที่ได้ในรอบสุดท้ายของ Intermediate Keys ที่ได้จะถูกใช้เป็นเซสชันคีย์ (Session key) (SK_j) โดยที่ $j = 1, \ldots, m$ คือ

$$IK^n_1 = SK_1, IK^n_2 = SK_2, \ldots, IK^n_m = SK_j \qquad (2.3)$$

จากรูปแบบข้างต้นนี้ ทั้งอลิสและบ๊อบสามารถใช้ค่า SK_j ในการเข้ารหัสลับข้อมูลเพื่อเพิ่มความปลอดภัยในการสื่อสารได้ จากเทคนิคนี้จะเห็นว่าจะมีการส่งเพียงค่าตัวแปรที่เป็นค่าเริ่มต้นเพื่อนำไปใช้ในการสร้างคีย์ และคีย์ที่ถูกสร้างขึ้น ถูกนำมาใช้เพียงครั้งเดียวไม่มีการนำมาใช้ซ้ำ ทำให้ลดช่องโหว่ในการถูกโจมตีลง รวมถึงไม่สามารถเดาคีย์ตัวถัดไปจากการล่วงรู้คีย์ตัวปัจจุบันหรือตัวก่อนหน้าได้

3. การออกแบบระบบ

เพื่อแก้ไขปัญหาและข้อจำกัดของงานวิจัยที่มีอยู่ งานวิจัยฉบับนี้นำเสนอโพรโทคอลใหม่ สำหรับการยืนยันตัวจริงและการส่งข้อมูลที่มีความมั่นคงปลอดภัย โดยใช้การสร้างและการกระจายคีย์แบบออฟไลน์

3.1 สมมติฐานเบื้องต้น (Initial Assumptions)

S คือ Server มีหน้าที่ให้บริการรับส่งไฟล์, C คือ Client มีหน้าที่ในการร้องขอบริการรับส่งไฟล์, $\{M\}_K$ คือ การเข้ารหัสลับแบบสมมาตรของข้อความ M ด้วย คีย์ K ค่า $H(M)$ คือ ค่าแฮช (Hash) ของข้อความ M, $H(M, K)$ คือ ค่า MAC ของข้อความ M ด้วยคีย์ K

3.2 Registration Protocol

โพรโทคอลสำหรับการลงทะเบียนมีการทำงานดังนี้ เริ่มโดยที่ C สร้างเซ็ตของ $\{PW_C, K_{CS}, DK_{CS}, M_{CS}\}$ โดยที่ K_{CS} คือ Long-term key, DK_{CS} คือ Distribution key, M_{CS} คือ ค่าสุ่มที่เป็นการระบุจำนวนของคีย์ที่ต้องการสร้างขึ้นเป็นพารามิเตอร์ของเทคนิคการสร้างและกระจายเซสชันคีย์แบบออฟไลน์ [9] และ PW_c คือ ค่ารหัสผ่านของ C จากนั้นจึงนำค่าทั้งหมดส่งผ่าน SSL Tunnel ดังแสดงข้างล่าง

C → S : $PW_C, K_{CS}, DK_{CS}, M_{CS}$

โดยที่มีการตั้งสมมติฐานว่าการสื่อสารด้วย SSL นั้น เป็นการสื่อสารอย่างปลอดภัยตราบใดที่คีย์ที่ถูกสร้างจาก SSL นั้นไม่ถูกล่วงรู้โดยผู้บุกรุก จากนั้น ทั้ง C และ S สร้างเซสชันคีย์ $\{SK_{CS1}, SK_{CS2}, \ldots, SK_{CSM}\}$ ตามรายละเอียดที่ได้กล่าวไปแล้วในหัวข้อ 2.2 เรียบร้อยแล้ว

3.3 Authentication Protocol

โพรโทคอลสำหรับการพิสูจน์ตัวจริงมีการทำงานดังนี้ เริ่มโดยที่ C สร้างข้อความ *Request, n* และค่าแฮชของ *Request, ID_C, PW_C, n, SK_{CSj}* โดยที่ n คือค่าสุ่ม (nonce) ID_C คือ ID ของผู้ใช้ PW_C คือ รหัสผ่าน และ SK_{CSj} คือ เซสชันคีย์ แล้วส่งไปยัง S ดังแสดงข้างล่าง

C → S : $Request, n, ID_C, H(Request, ID_C, PW_C, n, SK_{CSj})$

S → C : $Success, H(Success, SK_{CSj}, n)$

หลังจาก S ได้รับข้อความมาแล้ว S จะนำ *Request, ID_C, PW_C, n, SK_{CSj}* มาผ่านฟังก์ชันแฮช แล้วนำไปเปรียบเทียบกับข้อความที่ได้รับจาก C ถ้าตรงกัน S ก็จะส่งข้อความ *Success* ไปยัง C เป็นการตอบรับ

3.4 Transfer Protocol

Transfer Protocol ถูกแบ่งออกเป็น 2 ส่วนคือ โพรโทคอลที่ใช้สำหรับการรับไฟล์จากเซิร์ฟเวอร์ และโพรโทคอลที่ใช้สำหรับการส่งไฟล์ให้แก่เซิร์ฟเวอร์ ดังรายละเอียดต่อไปนี้

3.4.1 โพรโทคอลสำหรับการรับไฟล์

เมื่อ C ต้องการไฟล์จาก S ทาง C นำชื่อไฟล์ที่ต้องการรับมาเข้ารหัสลับด้วย SK_{csi+1} พร้อมทั้งนำชื่อไฟล์และ SK_{csi+1} ผ่านฟังก์ชันแฮชแล้วส่งให้แก่ S ดังนี้

C → S : $\{FileName\}_{Skcsi+1}, H(FileName, SK_{csi+1})$

จากนั้น S นำไฟล์ที่ C ร้องขอมาเข้ารหัสลับด้วย SK_{csi+1} พร้อมทั้งนำไฟล์และคีย์ SK_{csi+1} ผ่านฟังก์ชันแฮชแล้วส่งให้แก่ C ดังนี้

S → C : $\{File\}_{SKsci+1}$, $H(File, SK_{sci+1})$

3.4.2 โพรโทคอลสำหรับการส่งไฟล์

เมื่อ C ต้องการส่งไฟล์ไปยัง S นั้น C จะต้องนำไฟล์ที่ต้องการส่ง มาเข้ารหัสลับด้วย SK_{csi+1} พร้อมทั้งนำรหัสพิสูจน์ตัวจริงข้อความ ส่งให้แก่ S ดังแสดงข้างล่าง

C → S: $\{File\}_{SKcsi+1}$, $H(File, SK_{sci+1})$

หลังจากที่ S ได้รับไฟล์จาก C แล้วนั้น หาก S ได้รับไฟล์สมบูรณ์จะส่งข้อความว่า *Success* กลับไปให้ C ซึ่งหากการส่งไม่สมบูรณ์ก็จะส่งข้อความว่า *Fail* กลับไปให้ C ข้อความที่จะส่งกลับมาให้ C ถูกเข้ารหัสลับด้วยคีย์ SK_{csi+1} พร้อมทั้งนำข้อความและ SK_{csi+1} ซึ่งผ่านฟังก์ชันแฮชแล้วส่งให้แก่ C ดังแสดงข้างล่าง

S → C: $\{Success/Fail\}_{SKcsi+1}$,$H(Success/Fail\ ,SK_{sci+1})$

4. ผลการทดลอง

โพรโทคอลที่นำเสนอมีรายละเอียดของอัลกอริทึมในการเข้ารหัสลับชนิดต่างๆ ที่ถูกเลือกใช้ในงานวิจัยฉบับนี้มีดังนี้

4.1 Cryptographic Hash Function

อัลกอริทึมแบบ SHA-1 [7] ซึ่งมีขนาด 160 บิต ถูกเลือกใช้เป็นฟังก์ชันแฮชในงานวิจัยนี้ เนื่องจากมีความยาวของค่าแฮชที่มากกว่าอัลกอริทึมแบบ MD5

4.2 Message Authentication Code

ในการสร้างค่า MAC เลือกใช้ SHA-1 [7] เนื่องจากเป็นอัลกอริทึมที่ใช้กันอย่างแพร่หลายเพราะมีความทนทานสูงกว่า MD5

4.3 Symmetric Encryption

ใช้อัลกอริทึม AES (Advanced Encryption Standard) [6] สำหรับการเข้ารหัสลับแบบสมมาตร ซึ่งได้รับความนิยมสูง เพราะเป็นอัลกอริทึมที่มีความซับซ้อนสูง แต่สามารถทำการคำนวณได้อย่างรวดเร็ว

ในรายละเอียดของการทดลอง ผู้วิจัยได้พัฒนาระบบต้นแบบที่ใช้โพรโทคอลที่นำเสนอ โดยนำซอฟต์แวร์ติดตั้งที่เครื่องไคลเอนท์และเซิร์ฟเวอร์ ใช้การเชื่อมต่อผ่านทาง ADSL ซึ่งตั้งเซิร์ฟเวอร์ไว้ที่ CAT Telecom

การทดลองที่ได้นำเสนอ ได้ทดสอบ Transfer Protocol โดยทำการทดสอบกับไฟล์ 2 แบบคือ ไฟล์ที่เข้ารหัสลับและไฟล์ที่ไม่เข้ารหัสลับ ไฟล์ที่ใช้ในการทดสอบนี้ มีขนาด 1, 5 และ 10 MB ตามลำดับ โดยทำการทดสอบจำนวน 20 ครั้ง ซึ่งจากผลการทดสอบตามตารางที่ 1 พบว่าการรับส่งไฟล์โดยใช้ระบบที่พัฒนาขึ้นใช้เวลามากกว่าการรับส่งไฟล์โดยไม่มีการเข้ารหัสลับ ซึ่งถือว่าสมเหตุสมผล เมื่อพิจารณาจากคุณสมบัติทางด้านความมั่นคงปลอดภัยที่เพิ่มขึ้น

ตารางที่ 1 **เวลาที่ใช้ใน** Transfer Protocol

ขนาดไฟล์ (MB)	ไฟล์ที่เข้ารหัสลับ		ไฟล์ที่ไม่เข้ารหัสลับ	
	เวลาเฉลี่ยของการรับ (ms)	เวลาเฉลี่ยของการส่ง (ms)	เวลาเฉลี่ยของการรับ (ms)	เวลาเฉลี่ยของการส่ง (ms)
1	6,431	43,441	3,649	17,817
5	52,445	214,981	13,527	83,782
10	111,429	429,252	25,082	168,100

5. การวิเคราะห์ความมั่นคงปลอดภัย

5.1 การรักษาความลับของข้อมูล

ในการเข้ารหัสลับด้วยเซสชันคีย์แต่ละครั้ง ผู้ที่มีเซสชันคีย์ที่ตรงกันเท่านั้นที่จะสามารถถอดรหัสลับข้อมูลได้ จึงทำให้ได้คุณสมบัติการรักษาความลับของข้อมูลที่ส่ง

5.2 การต้านทานต่อ Brute Force Attack

สำหรับการค้นหาเซสชันคีย์ที่ถูกต้องนั้น ในโพรโทคอลที่นำเสนอนี้จะทำได้ค่อนข้างยาก เนื่องจากมีการเปลี่ยนแปลงค่าของเซสชันคีย์ในทุกๆ ครั้งที่มีการส่งหรือตอบกลับคำร้องขอต่างๆ นอกจากนี้การค้นหาค่าตั้งต้นที่ใช้ในการสร้างชุดของเซสชันคีย์ทั้งหมดก็ทำได้ยากอีกเช่นกันเนื่องจากเมื่อทำการสร้างชุดของคีย์แล้ว ค่าเริ่มต้นดังกล่าวจะถูกลบทิ้งจากระบบทันที

5.3 การต้านทานต่อ Replay Attack

การโจมตีจะมีโอกาสสำเร็จได้น้อยมากเนื่องจากการเปลี่ยนเซสชันคีย์ทุกครั้งที่มีการส่งหรือตอบกลับคำร้องขอต่างๆ ฉะนั้น ถ้ามีการส่งคำร้องขอหรือตอบกลับเดิมมาอีกครั้งจะต้องเป็นข้อความที่เข้ารหัสลับด้วยเซสชันคีย์ตัวถัดไปแล้ว ซึ่งผู้โจมตีจะไม่รู้ค่าเซสชันคีย์ตัวถัดไปนั่นเอง

5.4 ความคงสภาพของข้อมูล

การใช้ Message Authentication Code (หรือ MAC) ทำให้สามารถตรวจสอบได้ว่าข้อมูลที่ส่งมานั้นไม่ถูกแก้ไขระหว่างการรับส่ง โดยบุคคลอื่นมาแล้วตรวจสอบไม่ได้ เพราะหากข้อความถูกแก้ไข เมื่อสร้างค่า MAC มาเปรียบเทียบจะมีค่าไม่ตรงกัน

5.5 การพิสูจน์ตัวจริงของผู้ส่งข้อความ

แต่ละข้อความมีการนำเอา MAC มาประยุกต์ใช้ ทำให้ไคลเอนท์มั่นใจได้ว่าเซิร์ฟเวอร์เป็นผู้ส่งข้อความมา รวมถึงเซิร์ฟเวอร์เองก็มั่นใจได้ด้วยเช่นกัน เพราะค่า MAC ต้องใช้คีย์ที่แชร์กันไว้ระหว่างผู้รับกับผู้ส่งมาใช้ร่วมด้วยนั่นเอง

6. บทสรุปและข้อเสนอแนะ

งานวิจัยฉบับนี้ได้นำเสนอรูปแบบการติดต่อสื่อสารเพื่อการรับส่งข้อมูลที่มีความมั่นคงปลอดภัย โดยพิจารณาปัญหาและข้อจำกัดของโพรโทคอลการรับส่งข้อมูลที่มีอยู่ นอกจากนี้ยังได้ทำการเลือกเอาอัลกอริทึมการเข้ารหัสลับที่เหมาะสมทั้งในด้านความมั่นคงปลอดภัยและความเร็วในการทำงานมาใช้เพื่อเพิ่มประสิทธิภาพของระบบ การนำเอาการพิสูจน์ตัวจริงด้วยรหัสผ่านโดยไม่จำเป็นต้องส่งรหัสผ่านซึ่งเป็นการเพิ่มความสะดวกให้แก่ผู้ใช้เนื่องจากผู้ใช้ไม่จำเป็นที่จะต้องเก็บพับบลิคคีย์หรือไพรเวทคีย์ไว้ภายในระบบ ผลที่ได้จากการทดลองแสดงให้เห็นว่า โพรโทคอลที่นำเสนอรวมทั้งระบบที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้ได้ในการใช้งานได้จริงในปัจจุบัน

สำหรับงานวิจัยที่จะดำเนินการพัฒนาต่อไปนั้น ผู้วิจัยมุ่งเน้นที่จะทำการพัฒนาเทคนิคที่จะทำให้การรับส่งข้อมูลมีความรวดเร็วขึ้น รวมถึงพัฒนาการกระจายคีย์ที่เหมาะสมกับสภาพแวดล้อมสำหรับอุปกรณ์ไร้สาย

เอกสารอ้างอิง

[1] J. Postel and J. Reynolds, "File Transfer Protocol (FTP)," RFC 959, Oct 1985.
Available: http://www.ietforg/rfc/

[2] Y. Ma, H. T. Liu, and B. Y. Cai, "Design and implementation of a secure FTP system," Applications and Software, Aug 2007, pp.175-176.

[3] W. C. He, Y. Y. Zhang, and P H. Liu, "Research and design of a computer encryption communication system based on secure FTP," Network Security Technology and Application, Jan 2007, pp.92-94.

[4] L. Xia, C. Feng, D. Yuan, C. Wang, “Design of Secure FTP System”, Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS) 2010, pp.270-273.

[5] S. Kungpisdan, “Accountability of Centralized Payment Systems: Formal Reasoning, Protocol Design, and Analysis,” IETE Technical Review, 2010.

[6] Announcing the Advanced Encryption Standard (AES), FIPS 197 November 26, 2001.
Available:
http://csrc.nist.gov/publications/fips/fips197/

[7] H. Krawczyk, M. Bellare, R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” RFC 2104, February 1997.

[8] RSA Laboratories, “PKCS #1 v2.1: RSA Cryptography Standard,” June 14, 2002.

[9] Kungpisadan *et al.* “A Secure Offline Key Generation With Protection Against Key Compromise”, Proceedings of the 13th World Multiconference on Systemics, Cybernetics, and Informatics 2009, Orlando, Florida, July, 2009, pp.63-67.