การวิเคราะห์ปัญหาและการทดสอบความมั่นคงของเทคโนโลยีรหัสผ่านแบบใช้ครั้งเดียว

Problem Analysis and Security Testing of One Time Password Technology

ประพจน์ ธรรมศิรารักษ์,[1] สมนึก พ่วงพรพิทักษ์[2]

Prapot Thumsiraruk,[1] Somnuk Puangpronpitag[2]

Received: 5 April 2014 ; Accepted: 14 August 2014

บทคัดย่อ

รหัสผ่านแบบใช้ครั้งเดียว (OTP: One Time Password) ถือเป็นองค์ประกอบที่สำคัญสำหรับระบบการยืนยันตัวตนทั้งหลาย โดยเฉพาะอย่างยิ่งสำหรับระบบธนาคารออนไลน์ ซึ่ง OTP มักจะถูกใช้ในการรักษาความมั่นคงเป็นชั้นที่สองเพื่อปกป้องระบบหากมีการรั่วไหลของรหัสผ่านหลัก อย่างไรก็ตาม OTP เองก็ยังมีช่องโหว่อยู่ เมื่อเร็วๆ นี้มีข่าวและรายงานเกี่ยวกับการโจมตีระบบธนาคารออนไลน์ออกมาเป็นจำนวนมาก แม้จะมีการใช้ OTP แล้วก็ตาม ดังนั้นในงานวิจัยนี้จึงได้ทำการวิเคราะห์ปัญหาของ OTP ชนิดต่างๆ ที่มีอยู่ในปัจจุบัน โดยแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ (1) วิเคราะห์ข้อดีและข้อเสียของ OTP แต่ละชนิดที่ถูกนำไปใช้งานจริง ได้แก่ Email OTP, SMS OTP, Token OTP แล Mobile OTP (2) วิเคราะห์จุดแข็งและจุดอ่อนของ Algorithm ที่ใช้ในการสร้างและจัดการ OTP ได้แก่ Counter-based OTP, Time-based OTP และ Challenge-Response OTP นอกจากนี้ ยังได้ทำการทดลองในระบบเครือข่ายสำหรับการทดสอบ เพื่อศึกษาการโจมตีที่อาจเกิดขึ้นกับ OTP และสุดท้ายนี้ งานวิจัยนี้ได้นำเสนอแนวความคิดการแก้ปัญหาและการปรับปรุงประสิทธิภาพของ OTP

คำสำคัญ: รหัสผ่านแบบใช้ครั้งเดียว การโจมตีระบบธนาคารออนไลน์ การยืนยันตัวตนแบบพหุปัจจัย

Abstract

One Time Password (OTP) is an important component in several authentication systems, particularly for online banking systems. It is generally deployed as the second security layer to protect a system in case the main password has been compromised. However, OTP itself has a few vulnerabilities. Recently, there have been several news reports of attacks on online banking systems, even with the OTPs. Hence, in this paper, we analyze the potential problems of various OTPs. The analysis focuses on: (1) the pros/cons of each OTP type (i.e., Email OTP, SMS OTP, Token OTP and Mobile OTP), (2) the strength and weakness of OTP algorithms (such as Counter-based OTP, Time-based OTP and Challenge-Response OTP). Furthermore, testbed experiments have been done to study the potential attacks of OTPs. Finally, we present our solutions to solve the problems, and how to improve the OTPs.

Keywords: One Time Password (OTP), Online Banking Attack, Multiple Factor Authentication (MFA)

[1] *นิสิตปริญญาโท,[2] อาจารย์, สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ, มหาวิทยาลัยมหาสารคาม อำเภอกันทรวิชัย จังหวัดมหาสารคาม 44150*

[1] *Master's degree student, [2] Lecturer, Department of Computer Science, Faculty of Informatics, Mahasarakham University, Kantarawichai District, Maha Sarakham 44150, Thailand.*

* *Corresponding author: Somnuk Puangpronpitag, Lecturer, Department of Computer Science, Faculty of Informatics, Mahasarakham University, Kantarawichai District, Maha Sarakham 44150, Thailand. somnuk.p@msu.ac.th*

บทนำ

รหัสผ่านแบบใช้ครั้งเดียว[1] (OTP: One Time Password) ถูกออกแบบมาเพื่อช่วยแก้ปัญหาการรั่วไหลของรหัสผ่านแบบ User Knowledge ที่มีจุดอ่อนอยู่หลายประการ เช่น รหัสผ่านรั่วไหลจากการถูกดักจับข้อมูลด้วยซอฟต์แวร์ดักจับข้อมูลในขณะที่ข้อมูลถูกส่งผ่านบนระบบเครือข่าย เช่น โปรแกรม Wireshark และ Cain & Abel รหัสผ่านรั่วไหลได้โดยการบอกต่อผู้อื่น รหัสผ่านรั่วไหลได้จากการตั้งรหัสผ่านที่ง่ายต่อการคาดเดา หรือจากการลืมรหัสผ่านของตัวผู้ใช้เอง ในปัจจุบัน OTP นิยมนำมาใช้เป็นปัจจัยที่สองร่วมกับรหัสผ่านแบบปกติ เพื่อเป็นการเพิ่มความมั่นคงในการยืนยันตัวตนเข้าใช้งานระบบ

ในระบบ Online Banking ได้นำ OTP ไปใช้งานโดยแบ่งออกเป็น 4 รูปแบบ ซึ่งล้วนแล้วแต่ยังมีปัญหาอยู่ ได้แก่ (1) Email OTP ปัจจุบันไม่ค่อยถูกนำมาใช้งานแล้ว เนื่องจากมีช่องโหว่มากมาย เช่น การถูกโจมตีด้วยวิธีการปลอมแปลง Email ไปหลอกลวงเหยื่อ (Email spoofing) เพื่อทำให้เหยื่อหลงเชื่อคลิกเข้าเว็บปลอม แล้วกรอกข้อมูล Username/Password ให้ ซึ่งเรียกเทคนิคนี้ว่า Phishing[2] (2) SMS OTP เป็นรูปแบบที่นิยมนำมาใช้งานมากที่สุดในปัจจุบัน อย่างไรก็ตาม ได้เริ่มมีข่าวตั้งแต่เดือนกุมภาพันธ์ พ.ศ. 2556 ที่ผ่านมาว่า SMS OTP โดนโจมตีด้วยวิธีการปลอมแปลง SMS ไปหลอกลวง (SMS spoofing) ให้เหยื่อคลิกลิงค์ที่ส่งมาให้ เพื่อติดตั้ง Malware[2] โดยแอบอ้างว่าเป็น SMS ที่ส่งมาจากธนาคาร เมื่อเหยื่อหลงเชื่อและติดตั้ง Malware ดังกล่าวแล้ว เหยื่อคนนั้นก็จะไม่ได้รับ SMS อีกเลย เนื่องจาก SMS จะถูกส่งต่อไปยังเครื่อง Hacker ทันที (3) Token OTP เป็นรูปแบบที่มีความมั่นคงที่สุดในขณะนี้ แต่ปัญหาคือการลงทุนที่สูงมากจากการสั่งซื้ออุปกรณ์มาใช้ ทำให้ธนาคารหรือบริษัทหลายแห่งยังไม่ค่อยกล้าเสี่ยงที่จะเปลี่ยนมาใช้รูปแบบนี้ (4) Mobile OTP เป็นรูปแบบใหม่ล่าสุด มีลักษณะคล้ายกับ Token OTP แต่ใช้ Smartphone เป็นอุปกรณ์ในการประมวลผล จึงมีความเสี่ยงในเรื่องของ Malware ไม่มากก็น้อย นอกจากนี้ OTP ส่วนใหญ่ที่นำมาใช้งานนั้นจะแสดงผลลัพธ์ออกมาเป็นตัวเลขอย่างเดียว จำนวน 6-8 หลักเท่านั้น ทำให้มีความเสี่ยงมากต่อการถูกโจมตีด้วยการสุ่มรหัสผ่านทีละตัวจนครบทุกข้อมูลที่เป็นไปได้ (Brute-force attack) ทั้งยังพบอีกว่า OTP Algorithm แต่ละรูปแบบที่มีในเอกสาร Request For Comment (RFC) ของ Internet Society (ISOC) ยังมีปัญหาหรือจุดอ่อนบางประการอยู่ เช่น ปัญหา Out of Synchronization, ผลลัพธ์ของ OTP ที่ยาวเกินไป เป็นต้น

จากปัญหาที่ได้กล่าวมา งานวิจัยนี้จึงสนใจทำการวิเคราะห์ปัญหาของเทคโนโลยี OTP ที่มีอยู่ในปัจจุบันและทำการทดลองภายใต้ระบบเครือข่ายสำหรับการทดสอบ (Testbed) เพื่อศึกษาการโจมตีที่อาจเกิดขึ้นกับ OTP โดยแบ่งออกเป็น 2 ส่วนใหญ่ๆ ได้แก่ (1) รูปแบบการนำ OTP ไปใช้งาน (2) OTP Algorithm ที่ใช้ในการสร้างและจัดการ OTP พร้อมทั้งนำเสนอแนวทางการแก้ปัญหาและปรับปรุง OTP ให้มีประสิทธิภาพและมีความมั่นคงมากขึ้น

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. Multiple Factor Authentication (MFA)

MFA คือ กระบวนการยืนยันตัวตน โดยใช้ปัจจัยมากกว่าหนึ่งปัจจัยร่วมกัน ปัจจัยการยืนยันตัวตนแบ่งออกเป็น 3 ส่วนหลักๆ ได้แก่ (1) User Knowledge สิ่งที่ผู้ใช้รู้และจดจำข้อมูลไว้ เช่น Username/Password, PIN Code (2) User Possession สิ่งที่ผู้ใช้ถือครอง เช่น Token OTP (3) User Attribute ลักษณะเฉพาะของตัวบุคคลนั้นๆ เช่น ลายนิ้วมือ ลายพิมพ์ม่านตา เป็นต้น

MFA เริ่มมีการนำมาใช้ในประเทศไทยตั้งแต่ปี พ.ศ. 2551 เนื่องจากการใช้ Username/Password ที่เป็น User Knowledge เพียงอย่างเดียวในการยืนยันตัวตนเข้าใช้งานระบบที่ต้องการความมั่นคงสูงอย่าง Online Banking กลายเป็นสิ่งที่ไม่น่าเชื่อถืออีกต่อไป เพราะรหัสผ่านรั่วไหลได้ ถูกดักจับได้ ถูก Phishing เป็นต้น จึงมีการนำ MFA มาใช้เพื่อแก้ปัญหาต่างๆ เหล่านี้ โดยระบบส่วนใหญ่จะใช้การยืนยันตัวตนแบบสองปัจจัยร่วมกัน (TFA: Two Factor Authentication) ซึ่งสองปัจจัยที่นิยมนำมาใช้งานร่วมกันคือ User Knowledge (เลือกใช้ Username/Password) และ User Possession (เลือกใช้ OTP)

2. One Time Password (OTP)

OTP หรือรหัสผ่านแบบใช้ครั้งเดียว คือ รหัสผ่านที่ใช้ได้เพียงครั้งเดียวเท่านั้น ใช้สำหรับยืนยันตัวตนเข้าสู่ระบบที่ต้องการความมั่นคงสูง ได้รับการออกแบบขึ้นมาเพื่อแก้ปัญหา Static Password ที่มีจุดอ่อน เช่น รั่วไหลได้ ถูกดักจับได้ใช้งานซ้ำ ผู้ใช้หลงลืมได้ เป็นต้น

OTP Algorithm แบ่งออกได้เป็น 3 รูปแบบดังนี้ (1) Event-based OTP/ Counter-based OTP คือการใช้ตัวนับ (Counter) ที่เพิ่มค่าขึ้นทุกครั้งเมื่อมีการร้องขอเพื่อยืนยันตัวตน มาเป็นตัวแปรในการสร้างรหัส OTP ซึ่งหาก Counter มีค่าตรงกันระหว่าง Client กับ Server และมีการประมวลผลด้วย Algorithm เดียวกัน ก็จะได้รหัส OTP ตัวเดียวกัน โดย OTP ที่ได้มาจะเป็นตัวเลขจำนวน 6-8 หลัก

Counter-based OTP ที่ใช้อยู่ในขณะนี้ เป็นการนำเทคนิค Hash-based Message Authentication Code (HMAC)[3] มาประยุกต์ใช้โดยตรง จึงเรียกในอีกชื่อว่า HMAC-based OTP (HOTP)[4] (2) Time-based OTP (TOTP)[5] เป็นรูปแบบที่นิยมใช้งานมากที่สุดในขณะนี้ เกิดขึ้นมาหลัง HOTP โดยได้นำ HOTP มาใช้เป็นต้นแบบ เพียงแค่เปลี่ยนค่าของตัวแปรจากเดิมที่ใช้ค่าของ Counter มาเป็นค่าของเวลาที่อยู่บน Client กับ Server แทน ส่วนผล OTP ที่ได้ก็จะเป็นตัวเลข 6-8 หลักเช่นกัน (3) Challenge-Response OTP เป็นการใช้ข้อมูล Challenge ที่ส่งมาจาก Server ไปสร้างเป็นรหัส OTP อยู่ที่ Client จากนั้นก็ Response รหัส OTP ที่ได้กลับไปยัง Server เมื่อ Server ได้รับ OTP ก็จะทำการสร้าง OTP อีกชุดหนึ่ง (OTP') โดยใช้ Challenge เดียวกันกับที่ Client ได้รับ นำ OTP ทั้ง 2 มาเปรียบเทียบกัน (OTP==OTP') หากตรงกันแสดงว่าการยืนยันตัวตนถูกต้อง OTP ที่ได้เป็นตัวอักษรภาษาอังกฤษจำนวน 6-24 ตัว โดยมี S/Key OTP[6] เป็นระบบพื้นฐานของรูปแบบนี้

3. OTP กับ Online Banking

Online Banking หรือ Internet Banking หรือ E-banking เป็นระบบสารสนเทศที่ต้องการความมั่นคงสูง เนื่องจากเป็นการทำธุรกรรมผ่านอินเทอร์เน็ต ดังนั้น OTP จึงเป็นเครื่องมือและกลไกสำคัญที่ช่วยในเรื่องความมั่นคงสำหรับกระบวนการยืนยันตัวตนเพื่อเข้าใช้งานระบบ ซึ่งในการนำ OTP มาใช้งานจริงนั้น มีอยู่ด้วยกัน 4 รูปแบบ ดังนี้ (1) Email OTP คือ การส่ง OTP จาก Server ไปยัง Email address ของผู้ใช้ ผ่านทางอินเทอร์เน็ต (2) SMS OTP คือ การส่ง OTP จาก Server ไปยังโทรศัพท์มือถือของผู้ใช้ ผ่านทาง SMS ซึ่งเป็นรูปแบบที่นิยมนำมาใช้งานมากที่สุดในปัจจุบันนี้ (3) Token OTP คือ อุปกรณ์ที่ทำหน้าที่สร้าง OTP ขึ้นมา แล้วแสดงผลทางหน้าจอ ผู้ใช้จึงไม่จำเป็นต้องรอรับรหัส OTP จาก Server เช่น RSA SecurID เป็นต้น (4) Mobile OTP คือ การนำเอา Smartphone มาทำหน้าที่สร้างรหัส OTP เสมือนเป็นอุปกรณ์ Token OTP โดยการติดตั้ง Mobile app ที่พัฒนาขึ้นมาโดยเฉพาะ ทั้งนี้ขึ้นอยู่กับการออกแบบขององค์กรหรือบริษัทนั้นๆ เช่น Google Authenticator[7], Facebook Code Generator[8] พบว่ามีบางธนาคารในประเทศอินเดียได้นำรูปแบบนี้มาใช้ เช่น ธนาคารยูเนี่ยน (Union Bank of India) แต่ยังไม่มีรายงานข่าวว่ามีการนำมาใช้กับธนาคารในประเทศไทย

4. OTP กับประเทศไทย

สำหรับในประเทศไทยนั้น OTP ได้ถูกนำมาใช้เป็นเครื่องมือที่ช่วยเพิ่มความมั่นคงในการยืนยันตัวตนเข้าใช้งานระบบที่ต้องการความมั่นคงสูงเช่นเดียวกัน ที่เห็นได้ชัดคือระบบ Online Banking และ E-commerce ต่างๆ โดยในปี พ.ศ. 2551 ธนาคารแห่งประเทศไทยได้ประกาศให้ธนาคารทุกแห่งที่มีระบบการทำธุรกรรมออนไลน์ ต้องใช้การยืนยันตัวตนมากกว่าหนึ่งปัจจัยในการใช้งานระบบ[9] จึงเริ่มมีการนำระบบ OTP มาใช้ตั้งแต่นั้นมา ซึ่งในยุคแรกจะเป็นการใช้ Email OTP แต่ถูกยกเลิกไปในปี พ.ศ. 2552 เนื่องจากมีช่องโหว่จนสามารถถูกโจรกรรมเงินได้ จากนั้นธนาคารของประเทศไทยทุกแห่งจึงเปลี่ยนมาใช้ SMS OTP แทน และได้ใช้งานมาจนถึงทุกวันนี้ มีเพียงธนาคารต่างชาติบางแห่งที่ได้ใช้ Token OTP เช่น ธนาคาร HSBC สำหรับสถานการณ์ที่เกิดขึ้นในประเทศไทยตั้งแต่เดือนกุมภาพันธ์ปี พ.ศ. 2556 เป็นต้นมา พบว่ามีข่าวการโจรกรรมเงินใน Online Banking ออกมาอย่างต่อเนื่อง ถึงแม้ว่าได้ใช้ SMS OTP แล้วก็ตาม งานวิจัยนี้จึงได้ทำการทดลองและวิเคราะห์ปัญหาของระบบ OTP รวมทั้ง SMS OTP ว่าเหตุใดจึงยังถูก Hack ได้ และทางคนร้ายมีวิธีการอย่างไรในการโจรกรรมเงิน

5. Malware

Malware[2] มาจากคำว่า "Malicious Software" คือ ซอฟต์แวร์ที่มีลักษณะเป็นภัยคุกคามต่อคอมพิวเตอร์ ทำให้เกิดอันตรายต่อข้อมูลที่อยู่ในเครื่องคอมพิวเตอร์โดยทางตรงและทางอ้อม เช่น Virus, Worm, Trojan Horse, Spyware เป็นต้น ซึ่งพบว่า Trojan Horse หรือโปรแกรมที่ดูเหมือนจะมีประโยชน์แต่แท้จริงแล้วกลับมีการซ่อนโค้ดที่ได้ออกแบบมาเพื่อจุดประสงค์ร้าย เช่น ขโมยข้อมูล แก้ไขข้อมูล ทำลายระบบ เป็นต้น คือตัวการสำคัญที่ทำให้สามารถโจรกรรมเงินใน Online Banking ได้ สำหรับ Trojan ที่ขึ้นชื่อด้านนี้ ได้แก่ ZitMo[10], WUC's Conference.apk[11] และ Svpeng[12] ซึ่งได้กล่าวถึงรายละเอียดในหัวข้อผลการวิเคราะห์ Trojan Horse

6. งานวิจัยที่เกี่ยวข้อง

ปรัชญา ไชยเมือง และคณะ[13]ได้ทำการปรับปรุง S/Key OTP ที่มีปัญหาด้านความยาวของผลลัพธ์ OTP คือเป็นไปได้สูงสุดถึง 24 ตัวอักษร โดยใช้วิธีแปลงรหัสแบบ Base64 Encoding เพื่อลดจำนวนตัวอักษรลง จนเหลือ 12 ตัวอักษร แต่ยังคงรักษาคุณภาพของความมั่นคงไว้เท่าเดิม และทำการทดสอบประสิทธิภาพของ Base64 S/Key OTP พบว่าจำนวนครั้งของการกรอก OTP ที่ผิดพลาดของผู้ใช้ลดน้อยลงเมื่อเทียบกับ S/Key OTP ปกติ

Mulliner และคณะ[14] ทำการวิเคราะห์ความมั่นคงของ SMS OTP พบว่ามีวิธีการโจมตีหลัก ดังนี้ (1) Mobile Phone Trojans (2) Wireless Interception (3) Sim Swap พร้อมแสดงให้เห็นว่า SMS OTP ไม่มีความมั่นคงพอ และได้เสนอ

วิธีแก้ปัญหาคือ (1) SMS End-to-End Encryption คือ พัฒนา app ที่ใช้ในการเข้ารหัสและถอดรหัส SMS ทั้งต้นทางและปลายทาง (2) Virtual Dedicated Channel คือ สร้างช่องทางเสมือน โดยการพัฒนา OtpMessages app (เป็นแบบ Pre-installed app) เพื่อปกป้อง SMS OTP จาก Trojans โดย OtpMessage จะคอยทำหน้าที่รอรับ SMS จาก SMS port เช่นเดียวกันกับ SMS app ทั่วไป แต่จะคัดกรองเอาเฉพาะ SMS OTP โดยใช้ Keywords ตรวจสอบข้อความ เช่น OTP, mTAN, mobileTAN, securetoken เป็นต้น ถ้าไม่ใช่ SMS OTP จึงจะปล่อยไปยัง SMS app หรือปล่อยให้เข้าสู่ SMS inbox นั่นเอง

ภูกิจ บุรีภักดี และปราโมทย์ กั่วเจริญ[15] นำเสนอวิธีการรักษาความมั่นคงและการเพิ่มประสิทธิภาพในการส่ง SMS โดยในด้านการรักษาความมั่นคงได้ใช้วิธีการเข้ารหัสลับด้วยการใช้ Elliptic Curve Cryptography (ECC) ซึ่งเป็นแนวทางการเข้ารหัสลับแบบ Public Key และใช้ Key sizeเท่ากับ 160 bit โดยให้เหตุผลที่เลือกใช้ ECC แทนที่จะเลือกใช้ RSA ว่า ECC ใช้ Key size สั้นกว่า RSA ก็ยังให้ความมั่นคงที่เท่ากันได้ ดังนั้นหากใช้ Key size เท่ากันแล้ว ECC ย่อมให้ความมั่นคงที่มากกว่า ไม่เพียงเท่านี้ ECC ยังมีความสามารถในการคำนวณที่รวดเร็ว ทั้งยังใช้พลังงานต่ำและใช้หน่วยความจำเพียงเล็กน้อย จึงเหมาะสำหรับการนำมาใช้งานบนอุปกรณ์เคลื่อนที่ขนาดเล็ก เช่น Smartphone, PDA

จากการศึกษางานวิจัยที่เกี่ยวข้องกับ OTP พบว่าส่วนใหญ่เป็นงานวิจัยเชิงประยุกต์ใช้ OTP หรือเป็นการนำเสนอวิธีการปรับปรุงและแก้ไขปัญหาข้อบกพร่องของ OTP หรือเป็นงานวิจัยเชิงวิเคราะห์ปัญหา เป็นต้น ซึ่งงานวิจัยที่ได้กล่าวมานี้ล้วนแต่มุ่งเน้นอยู่เพียง OTP ชนิดใดชนิดหนึ่งเท่านั้น ยังไม่มีงานวิจัยใดทำการทดลองและวิเคราะห์ปัญหาของระบบ OTP อย่างครอบคลุมทุกด้านและหลากหลายมากพอ งานวิจัยนี้จึงสนใจทำงานชิ้นนี้ขึ้นมา เนื่องจาก OTP ถือว่าเป็นด่านสกัดที่สำคัญของระบบ Online Banking และเพื่อให้เป็นประโยชน์แก่ผู้ที่มีความสนใจทางด้านนี้ จะได้นำงานวิจัยนี้ไปเป็นแนวทางการปรับปรุงและเพิ่มความมั่นคงให้กับระบบ OTP มากยิ่งขึ้นต่อไป

วิธีดำเนินการวิจัย

เป้าหมายของการวิเคราะห์ปัญหาของ OTP ที่มีอยู่ในปัจจุบัน เพื่อแสดงให้เห็นถึงจุดแข็งและจุดอ่อนของ OTP แต่ละชนิด โดยในภาพรวมสามารถแบ่งออกเป็น 2 ส่วนด้วยกันคือ (1) รูปแบบของการนำ OTP ไปใช้งาน ได้แก่ Email OTP, SMS OTP, Token OTP และ Mobile OTP (2) OTP Algorithm ได้แก่ Counter-based OTP, Time-based OTP และ Challenge-Response OTP

ในกระบวนการวิเคราะห์ได้มีการทดลองโจมตีกับเทคโนโลยีที่เกี่ยวข้อง ซึ่งเป็นการจำลองการโจมตีอยู่ภายใต้ระบบเครือข่ายสำหรับการทดสอบ (Testbed) โดยได้แบ่งการทดลองและ/หรือการวิเคราะห์ ออกเป็น 5 รายการ ดังนี้

1. การทดลองและวิเคราะห์ปัญหาของ Email OTP

Email OTP คือ รูปแบบที่ถูกนำมาใช้กับ Online Banking เมื่อในอดีต ซึ่งธนาคารของประเทศไทยได้ยกเลิกใช้ในปี พ.ศ. 2552 แต่ยังมีบางประเทศในกลุ่มประชาคมเศรษฐกิจอาเซียน (AEC) ที่ยังคงใช้รูปแบบนี้อยู่ ข้อดีของ Email OTP คือสามารถเข้าถึงได้จากทุกที่ที่มีอินเทอร์เน็ต และทางธนาคารก็ไม่ต้องเสียค่าใช้จ่ายในการส่ง OTP แต่ Email OTP นั้นยังมีช่องโหว่อยู่มาก ดังจะแสดงให้เห็นจากการทดลองและวิเคราะห์ต่อไปนี้

1.1 ทดลอง Email spoofing

เพื่อแสดงให้เห็นว่า สามารถปลอมแปลง Email เพื่อใช้ในการหลอกลวงเหยื่อได้ เครื่องมือที่ใช้คือ เว็บไซต์ Emkei's Instant Mailer[16] (http://emkei.cz) เป็นเว็บไซต์ที่ให้บริการส่ง Email ปลอม

1.2 วิเคราะห์ Email sniffing

เพื่อแสดงให้เห็นถึงกระบวนการที่ Hacker ใช้ในการดักจับข้อมูล Email ของเหยื่อ ไม่ว่าจะเป็น Password หรือเป็นเนื้อหาภายใน Email โดยใช้โปรแกรมดักจับข้อมูล เช่น Cain & Abel, Wireshark, TCPDump

2. การทดลองและวิเคราะห์ปัญหาของ SMS OTP

SMS OTP เป็นรูปแบบที่ถูกนำมาใช้งานมากที่สุดในขณะนี้ ไม่ว่าจะเป็น Online Banking หรือ E-commerce ธนาคารในประเทศไทยแทบทุกแห่งใช้ SMS OTP เนื่องจากในอดีตคิดว่ามีความปลอดภัยมาก ทั้งยังใช้งานง่าย สะดวกต่อผู้ใช้ และทางธนาคารก็ใช้เงินลงทุนน้อยมากหากเทียบกับการใช้อุปกรณ์ Token OTP แต่ในปัจจุบันไม่เป็นเช่นนั้นแล้ว เนื่องจากมีรายงานข่าวการโจรกรรมเงินใน Online Banking ออกมาอย่างต่อเนื่องตั้งแต่เดือนกุมภาพันธ์ปี พ.ศ. 2556 ที่ผ่านมา ดังนั้นในปัจจุบัน (ปี พ.ศ. 2557) การใช้ SMS OTP จึงถือว่ามีความเสี่ยงสูงมาก โดยสามารถศึกษาข้อมูลความเสี่ยงได้จากทดลองและวิเคราะห์ SMS OTP ต่อไปนี้

2.1 ทดลอง SMS spoofing

เพื่อแสดงให้เห็นว่า SMS สามารถถูกปลอมแปลงชื่อและเบอร์โทรศัพท์ของผู้ส่งได้ จึงเป็นสาเหตุให้คนร้ายนำไปใช้หลอกลวงเหยื่อ เช่น การแอบอ้างว่าส่งมาจากธนาคาร เครื่องมือที่ใช้คือ (1) Mobile app ชื่อ Fake Sms Sender[17]

ซึ่งมีให้ Download บน Play Store (2) เว็บไซต์ FakeText[18] (http://www.faketext.net) เป็นเว็บไซต์ที่ให้บริการส่ง SMS ปลอม อุปกรณ์ที่ใช้ทำการทดลอง ได้แก่ Smartphone ที่ใช้ระบบปฏิบัติการ Android จำนวน 2 เครื่อง และคอมพิวเตอร์ Notebook จำนวน 1 เครื่อง

2.2 วิเคราะห์ Trojan Horse (Sniff and Forward)

เพื่อศึกษากระบวนการทำงานของ Trojan Horse ที่เป็นต้นเหตุสำคัญของการโจรกรรมเงินใน Online Banking งานวิจัยนี้ได้วิเคราะห์ถึงวิธีการที่เหล่า Hacker ใช้ในการส่ง Trojan มาฝังอยู่บนเครื่องเหยื่อ และได้อธิบายถึงความเสี่ยงต่างๆ ที่อาจเกิดขึ้นหลังจากที่ Smartphone ติด Trojan แล้ว พร้อมทั้งนำเสนอแนวทางการแก้ปัญหาและป้องกัน

2.3 วิเคราะห์ Social Engineering

วิศวกรรมสังคม (Social Engineering)คือการโจมตีรูปแบบหนึ่งที่ใช้เทคนิคทางจิตวิทยา ซึ่งคดีการโจรกรรมเงินใน Online Banking ในช่วงหลังๆ พบว่าคนร้ายมักจะใช้วิธีนี้เป็นส่วนใหญ่ เช่น การใช้หลักฐานปลอมเพื่อขอ SIM ใหม่ การสร้างหลักฐานปลอมเพื่อขอเปิดบัญชีธนาคารที่มีชื่อบัญชีเดียวกันกับเหยื่อ เป็นต้น งานวิจัยนี้จึงได้ทำการศึกษาและวิเคราะห์ข้อมูลจากคดีต่างๆ ที่เกี่ยวข้องและได้เกิดขึ้นจริงในประเทศไทย

3. การวิเคราะห์ปัญหาของ Token OTP

เป็นการวิเคราะห์เพื่อนำเสนอข้อดีและข้อเสียของอุปกรณ์ Token OTP โดยได้วิเคราะห์ข้อมูลจากการศึกษาสำรวจ และสอบถามข้อมูลจากผู้รู้และผู้ที่ใช้งานจริง

4. การทดลองและการวิเคราะห์ปัญหาของ Mobile OTP

Mobile OTP ถือเป็นรูปแบบใหม่ล่าสุดของ OTP คาดว่าถูกสร้างขึ้นมาเพื่อต้องการแก้ปัญหา OTP รูปแบบอื่นๆ ก่อนหน้านี้ ซึ่งในอนาคตอาจถูกนำมาใช้แทนที่ตัวอื่น การทดลองนี้เป็นการทดลองใช้งาน Google Authenticator ที่เป็นหนึ่งในตัวอย่างของ Mobile OTP เพื่อนำไปวิเคราะห์หาจุดแข็งและจุดอ่อนต่อไป

5. การวิเคราะห์ปัญหาของ OTP Algorithm

OTP Algorithm นับว่าเป็นหัวใจสำคัญของ OTP เนื่องจากเป็นขั้นตอนวิธีในการสร้างรหัส OTP ขึ้นมา หากได้รับการออกแบบที่ดีมีคุณภาพ ย่อมส่งผลถึงความมั่นคงของ OTP ปัจจุบันมี 3 Algorithm ได้แก่ (1) Event-based OTP หรือ Counter-based OTP (2) Time-based OTP (3) Challenge-Response OTP ซึ่งพบว่าแต่ละ Algorithm ยังคงมีปัญหาอยู่ งานวิจัยนี้จึงได้ทำการศึกษาและวิเคราะห์ปัญหาของ OTP Algorithm ดังกล่าว

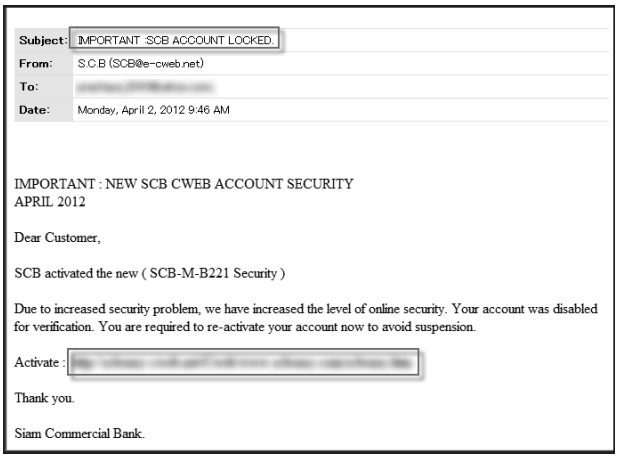
ผลการวิจัย

จากการทดลองและ/หรือการวิเคราะห์ปัญหาของ OTP ทั้ง 5 รายการ ได้ผลดังนี้

1. ผลการทดลองและวิเคราะห์ปัญหาของ Email OTP

1.1 ผลการทดลอง Email spoofing

Email สามารถถูกปลอมแปลงได้ ซึ่ง Hacker จะใช้ช่องโหว่ตรงนี้ในการนำไปหลอกลวงเหยื่อให้หลงกลได้ เช่น ส่ง Email ปลอมไปหลอกลวงเหยื่อว่าบัญชีธนาคารมีปัญหาจะถูกระงับใช้งาน ต้องรีบดำเนินการ Activate ด่วน โดย Hacker ใช้วิธีแอบอ้างว่าส่งมาจากธนาคาร ด้วยการปลอมแปลงชื่อผู้ส่ง ดัง Figure 1 เมื่อเหยื่อหลงกลกดคลิกลิงค์ที่แนบมาให้ ลิงค์ดังกล่าวจะเชื่อมโยงไปยังเว็บไซต์ปลอมที่ Hacker ได้เตรียมไว้ ซึ่งมีช่องว่างให้เหยื่อกรอกข้อมูลสำคัญ เช่น Username/Password ดัง Figure 2


Subject: IMPORTANT :SCB ACCOUNT LOCKED.
From: S.C.B (SCB@e-cweb.net)
To:
Date: Monday, April 2, 2012 9:46 AM

IMPORTANT : NEW SCB CWEB ACCOUNT SECURITY
APRIL 2012

Dear Customer,

SCB activated the new (SCB-M-B221 Security)

Due to increased security problem, we have increased the level of online security. Your account was disabled for verification. You are required to re-activate your account now to avoid suspension.

Activate :

Thank you.

Siam Commercial Bank.

Figure 1 an Example of Email spoofing

Figure 2 an Example of phishing website

จากกรณีดังกล่าว งานวิจัยจึงได้ทำการทดลองส่ง Email ปลอม โดยกระทำผ่านเว็บไซต์ชื่อ Emkei's Instant Mailer (http://emkei.cz) ซึ่งได้ทดลองปลอมแปลงชื่อผู้ส่งเป็น BualuangiBankingAlert (ลอกเลียนมาจากชื่อ Email ฉบับจริงของธนาคารกรุงเทพ) รวมทั้งปลอมแปลง Email address และข้อมูลอื่นๆ ดัง Figure 3 และผลลัพธ์จากการส่ง Email ปลอมฉบับนี้ไปยังปลายทางแสดงออกมาดัง Figure 4 จากการทดลองนี้ยังได้พบว่า Webmail Server ที่มีความมั่นคงสูงสามารถกรอง Email ขยะ (Spam mail) ได้เป็นอย่างดี อย่างไรก็ตามผู้ใช้ก็ควรตรวจทานอีกครั้ง

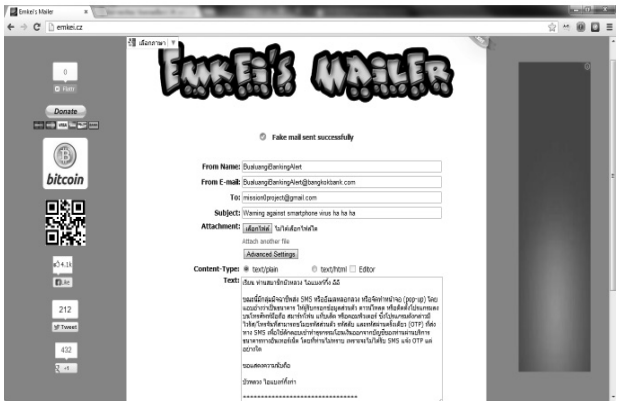


Figure 3 Email spoofing Test via http://emkei.cz

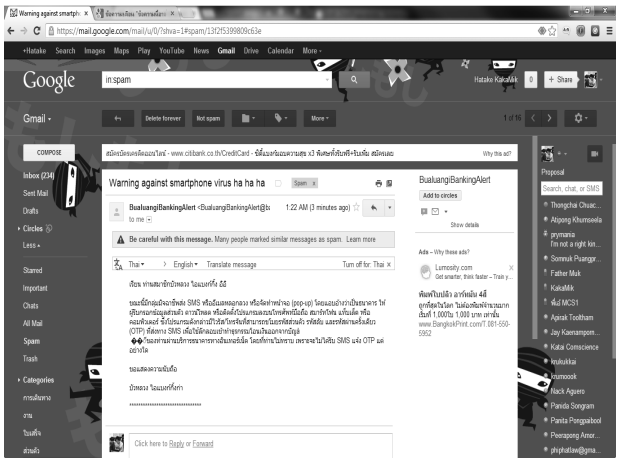


Figure 4 Result of the Email spoofing Test

1.2 ผลการวิเคราะห์ Email sniffing

เนื่องจาก Email ทำงานอยู่บนระบบเครือข่าย จึง มีความเสี่ยงที่จะถูกดักจับข้อมูลระหว่างการสื่อสาร วิธีการที่ Hacker นิยมใช้โจมตีคือ Man-in-the-Middle (MitM)[2] หมายถึงการแทรกกลางในระหว่างการสื่อสารของคน 2 คน ดัง Figure 5 ทำให้ Hacker สามารถดักจับข้อมูลของเหยื่อได้ ทั้งยังสามารถเปลี่ยนแปลงข้อมูลระหว่างทางได้ โปรแกรมที่ใช้ดักจับข้อมูลบนระบบเครือข่าย ได้แก่ Cain & Abel (ดัง Figure 6), Wireshark, TCPDump เป็นต้น

Figure 5 Man-in-the-Middle attack

หาก Hacker สามารถโจมตี Email OTP ด้วยวิธี MitM ได้สำเร็จ ไม่ว่าด้วยวิธีการ ARP spoof, DNS spoof, SSL Strip หรือใช้หลายวิธีการประกอบกัน ถ้าเหยื่อ Login เข้าใช้ Email เหยื่อก็จะถูกดักจับ Username/Password ได้ และในการเปิดอ่าน Email (เช่น อ่านค่า OTP) Hacker ย่อมได้รับข้อมูลจาก Webmail Server ก่อนเหยื่อ ดังนั้น Hacker จึงสามารถเปลี่ยนแปลงข้อมูลระหว่างทางได้ หรือไม่ก็จัดการลบ Email ฉบับนั้นทิ้งเลย จะเห็นได้ว่า เหยื่อมีความเสี่ยงที่จะเกิดความเสียหายได้อย่างแน่นอน ถึงแม้ว่าในปัจจุบัน Webmail Server ส่วนใหญ่ได้เพิ่มระบบรักษาความมั่นคงแล้วก็ตาม ตัวอย่างเช่น Gmail และ Hotmail โดย Gmail ได้ปรับทั้งเว็บไซต์ให้มีการเข้ารหัสในระหว่างการสื่อสารโดยใช้ Hypertext Transfer Protocol Secure (HTTPS) ส่วน Hotmail ผู้ใช้จะต้องเข้าไปตั้งค่าเพื่อเปิดใช้ HTTPS ด้วยตัวเอง นอกจากนี้ทั้งสองเว็บไซต์ยังได้เพิ่มระบบยืนยันตัวตนแบบสองขั้นตอน (2-Step Verification) เข้าไป โดยใช้รหัส OTP สำหรับการยืนยันตัวตนในขั้นตอนที่สอง แต่ถึงอย่างนั้นผู้ใช้ส่วนใหญ่ก็ไม่ได้เปิดใช้งานกัน

การทดลองและวิเคราะห์ปัญหาของ Email OTP ในข้างต้น แสดงให้เห็นว่าการใช้ Email OTP มีความเสี่ยงต่อการถูกโจรกรรมข้อมูลได้ง่าย การที่ธนาคารในประเทศไทยได้ยกเลิกใช้งานนั้นถูกต้องแล้ว ซึ่งบางประเทศในกลุ่ม AEC ที่ยังคงใช้งานอยู่ ก็ควรจะยกเลิกใช้งานเช่นเดียวกัน

Figure 6 Email sniffing by Cain & Abel

2. ผลการทดลองและวิเคราะห์ปัญหาของ SMS OTP

2.1 ผลการทดลอง SMS spoofing

ในปัจจุบันพบว่า มีวิธีการที่สามารถปลอมแปลงเลขหมายและชื่อของผู้ส่ง SMS ได้ จึงเป็นช่องทางให้เหล่าคนร้ายใช้ในการหลอกลวงเหยื่อ เช่น หลอกว่า SMS ส่งมาจากธนาคาร (ใช้เบอร์โทรศัพท์หรือชื่อของธนาคารนั้นๆ) ตัวอย่างกรณีการโจรกรรมข้อมูลด้วยการส่ง SMS ปลอม[19] คือ คนร้ายจะส่งลิงค์แนบมากับ SMS หลอกให้เหยื่อติดตั้ง app ของธนาคาร (app ปลอม ฝัง Trojan ที่มีประสงค์ร้าย เช่น ขโมยข้อมูลสำคัญของเหยื่อ) ดัง Figure 7

Figure 7 An Example of SMS Spoofing

จาก Figure 7 เหยื่อได้รับ SMS ปลอมจากคนร้ายที่ปลอมแปลงหมายเลขของผู้ส่งเป็นเบอร์ 027777777 แล้ว (เบอร์โทรศัพท์ของธนาคารไทยพาณิชย์) เนื้อหาภายในเป็นข้อความบอกให้เหยื่อกด Download app จากลิงค์ที่ส่งมาให้ถ้าหากเหยื่อหลงเชื่อกด Download และติดตั้ง app ดังกล่าว แล้วเปิดโปรแกรมขึ้นมา เหยื่อจะพบกับหน้า app ที่มีโลโก้ของธนาคารและมีช่องให้กรอก Username/ Password ของ SCB Easy Net ซึ่งหากเหยื่อหลงเชื่อกรอกข้อมูลจริง แล้วกดปุ่ม “ต่อ” ข้อมูลบัญชี SCB Easy Net ของเหยื่อก็จะตกเป็นของ Hacker ทันที ไม่เพียงเท่านี้ Hacker ยังได้แอบฝัง Trojan ไว้ใน app ด้วย โดยมันจะคอยทำหน้าที่ดักจับ SMS ที่เข้ามายังโทรศัพท์มือถือของเหยื่อ แล้วส่งต่อไปยัง Hacker ทันที เป็นสาเหตุที่ทำให้เหยื่อถูกโจรกรรมเงินไปโดยไม่รู้ตัว ขั้นตอนโดยสรุปแสดงออกมาเป็นแผนภาพ ดัง Figure 8 ส่งผลให้ธนาคารในประเทศไทยตื่นตัวและได้ออกประกาศแจ้งเตือนผู้ใช้งาน Online Banking ทุกคนให้ระมัดระวังภัยเกี่ยวกับกรณีดังกล่าวผ่านทางหน้าเว็บไซต์ธนาคาร ตัวอย่างประกาศแจ้งเตือนของธนาคารกรุงเทพ แสดงดัง Figure 9

Figure 8 The Steps of Hacking online-bank

Figure 9 An Annoucement of Bangkok Bank

งานวิจัยนี้ได้ทำการทดลองส่ง SMS ปลอมโดยใช้บริการของเว็บไซต์ FakeText และ Mobile app ที่มีชื่อว่า Fake Sms Sender แม้ว่าการสำรวจใน Play Store จะพบ app ที่เกี่ยวข้องเป็นจำนวนมาก แต่จากการทดลองใช้กลับพบว่ามีเพียงบาง app ที่สามารถส่ง SMS ปลอมได้จริง

Figure 10 SMS spoofing Test via www.faketext.net

จาก Figure 10 แสดงการทดลองส่ง SMS ปลอม ผ่านเว็บไซต์ FakeText (http://www.faketext.net) งานวิจัยทำการทดลองดังนี้ (1) กรอกเบอร์โทรศัพท์ปลายทางเป็น

เบอร์โทรศัพท์ของผู้วิจัยเอง (2) กรอกข้อมูลในช่องผู้ส่งว่า “JCSSE2013” (ชื่อหรือเบอร์โทรศัพท์ก็ได้) (3) ส่งข้อความ SMS ว่า “Welcome to JCSSE2013” ซึ่งผลลัพธ์ปรากฏดัง Figure 10 (ขวา) จะเห็นว่า Header และข้อความ SMS เป็นไปตามที่เราต้องการ ส่วนข้อความที่เกินมาว่า “(Sent by www.faketext.net)” เนื่องจากในครั้งนี้เป็นการทดลองส่งฟรี ซึ่งส่งได้เพียงครั้งเดียวเท่านั้น ถ้าหากไม่ต้องการให้ปรากฏข้อความดังกล่าว หรือต้องการส่ง SMS ปลอมมากกว่า 1 ฉบับ ผู้ใช้จำเป็นต้องเสียค่าบริการ

Figure 11 SMS spoofing Test via Mobile app

จาก Figure 11 แสดงการทดลองส่ง SMS ปลอม ผ่านทาง Mobile app ชื่อว่า Fake Sms Sender งานวิจัยทำการทดลองดังนี้ (1) เลือกเบอร์โทรศัพท์ปลายทางเป็นเบอร์บิดาของผู้วิจัย (สามารถเลือกได้จากรายชื่อที่มีอยู่ในโทรศัพท์) จึงปรากฏชื่อว่า “Father Muk” (2) กรอกข้อมูลเบอร์โทรศัพท์ของผู้ส่งเป็นเบอร์บ้านของผู้วิจัย ซึ่งแท้จริงแล้วโทรศัพท์บ้านไม่สามารถส่ง SMS ได้) (3) ส่งข้อความว่า “Good Afternoon Father. This is SMS which send from Home phone 555+” ดัง Figure 11 (กลาง) และผลที่ได้ปรากฏดัง Figure 11 (ขวา) จะสังเกตเห็นว่า Header แสดงเป็น “Ban Thumsiraruk” ซึ่งก็หมายความว่า เบอร์โทรศัพท์ของผู้ส่งที่ได้กรอกลงไปนั้นได้ไปเชื่อมโยงกับ Contact list ที่มีอยู่ในโทรศัพท์ของเหยื่อโดยอัตโนมัติ

2.2 ผลการวิเคราะห์ Trojan Horse

จากการทดลอง SMS spoofing หรือการส่ง SMS ปลอมในหัวข้อที่แล้วนั้น จะเห็นว่า เป็นการใช้ช่องโหว่เพื่อต้องการจะส่ง Trojan Horse เข้าไปฝังตัวใน Smartphone ของเหยื่อผ่านการติดตั้ง app จากลิงค์ที่แนบไปในข้อความ ดังนั้นตัวการสำคัญที่เป็นต้นเหตุของการโจรกรรมเงินใน Online Banking ก็คือ Trojan Horse นี่เอง สำหรับหน้าที่หลักของ Trojan Horse กรณี SMS OTP ก็คือ คอยดักจับ SMS ที่เข้ามาในเครื่องของเหยื่อและส่งต่อไปยัง Hacker

นอกจากวิธีข้างต้นแล้วยังมีอีกหลายวิธีที่ Hacker ใช้ในการปล่อย Trojan Horse มาฝังใน Smartphone ของเหยื่อ เช่น ทำการติดตั้งโดยตรงในขณะที่เหยื่อกำลังเผลอ หลอกให้ติดตั้ง Mobile app ผ่าน Email ปลอม หรือ สร้าง app ปลอมมาปล่อยให้ Download บน Play Store เป็นต้น ซึ่งสาเหตุที่ทำให้ Smartphone ติด Trojan ได้ง่ายขึ้นเป็นเพราะผู้ใช้ Smartphone ส่วนใหญ่ไม่ได้ติดตั้ง Anti Virus เนื่องด้วยเหตุผลหลักๆ คือ จะทำให้เครื่องทำงานได้ช้าลง และต้องเสียเงินในการใช้งาน app

Trojan Horse ที่มีชื่อเสียงทางด้านการโจรกรรมข้อมูลเกี่ยวกับ Online Banking มีดังนี้ (1) Zeus-in-the-Mobile หรือ ZitMo[10] เป็น Trojan ที่ถูกสร้างมาเพื่อตรวจสอบและขโมยข้อมูล SMS OTP โดยเฉพาะ พบเมื่อเดือนกันยายนปี พ.ศ. 2553 โดย Kaspersky Lab ดังนั้น Zeus จึงเป็น Trojan ที่ถูกพบมาก โดยเริ่มแรกพบอยู่บนระบบปฏิบัติการ Symbian และ BlackBerry แต่ปัจจุบันถูกตรวจพบมากขึ้นไม่ว่าจะเป็น Windows Phone หรือ Android โดยหน้าที่ของมันคือคอยตรวจสอบเวลาเหยื่อทำธุรกรรมต่างๆ เพื่อ Redirect page ไปยังหน้าที่ได้เตรียมไว้ และยังคอยขโมย OTP จากบริการ E-banking หรือที่เรียกว่า mTANs (mobile Transaction Authentication Numbers) แล้วยังคอยรับคำสั่งจาก Hacker อยู่ตลอดเวลาเพื่อเริ่มกระบวนการโจรกรรมข้อมูลต่างๆ (2) WUC’s Conference.apk[11] เป็น Trojan ที่ถูกส่งมากับ Email โดยเป็นไฟล์ที่แนบมาในลักษณะของ Mobile app เพื่อหลอกให้เหยื่อติดตั้ง ตรวจพบเมื่อเดือนมีนาคมปี พ.ศ. 2556 โดย Kaspersky Lab และได้เรียก Trojan นี้ว่า Backdoor.AndroidOS.Chuli.a ผลกระทบของ app นี้คือจะคอยขโมยข้อมูล Contact list, SMS, Call logs, Phone data (เช่น Phone number, OS version, Model, SDK version) เป็นต้น ซึ่งจะเข้ารหัสด้วย Base64 แล้วส่งต่อไปยัง Hacker ทันที (3) Svpeng[12] เป็น Trojan ประเภทหนึ่งที่คอยขโมยข้อมูลบัญชีธนาคารและอื่นๆ โดยแอบแฝงมากับ SMS spam วิธีการคือ Hacker จะสั่งให้เหยื่อส่ง SMS ไปยังเบอร์ที่มีใน Contact list ทำให้คนรู้จักของเหยื่อที่ได้รับ SMS หลงเชื่อและติดตั้งโปรแกรมลงบนเครื่อง ซึ่งส่งผลให้คนๆ นั้นกลายเป็นเหยื่อรายต่อไปและก็กระจาย Trojan นี้ต่อไปเรื่อยๆ อีก สำหรับ Svpeng หรือ Trojan-SMS.AndroidOS.Svpeng ได้ถูกตรวจพบครั้งแรกที่ประเทศรัสเซีย บน Smartphone ของผู้ใช้งาน app ของธนาคารแห่งหนึ่ง มันคอยขโมยข้อมูล Login Account แล้วก็โจรกรรมเงินในบัญชีของเหยื่อไป โดยใช้เทคนิคการ Phishing ตอนเหยื่อทำธุรกรรมออนไลน์ผ่าน app กล่าวคือ มันจะเปิดหน้า Login หลอกขึ้นมา ถ้าตรวจพบว่ามีการทำธุรกรรมใดๆ เกิดขึ้น จากนั้นก็การขโมยข้อมูลดังกล่าวและอ่านข้อมูลบัญชีต่างๆ ของเหยื่อ เพื่อโอนย้ายเงินต่อไป

Figure 12 SMS attacking by Mobile Trojans

จาก Figure 12 เป็นการทดลองจากงานวิจัยของ Mulliner และคณะ[14] ที่ได้ทำการวิเคราะห์ความมั่นคงของ SMS OTP และเสนอแนวทางป้องกันไว้ ในการทดลองนี้มีจุดประสงค์เพื่อทดสอบ Virtual Dedicated Channel หรือ Virtual SMS Channel ซึ่งเป็นหนึ่งในแนวทางการป้องกันที่ได้เสนอไป Figure12 (ซ้าย) เป็นการทดลองส่ง SMS ธรรมดาเข้าไป พบว่า SMS app ได้รับข้อความตามปกติ และ PoC Trojan ก็ได้รับข้อความนี้เช่นกัน (ดักจับ SMS ได้สำเร็จ) แต่จากการทดลองส่ง SMS OTP เข้าไป จะพบว่า SMS app และ PoC Trojan จะไม่ได้รับข้อความนี้เลย มีเพียง Otp Message app ที่ได้รับดัง Figure12 (ขวา) แสดงให้เห็นว่าวิธีนี้ช่วยป้องกันการดักจับ SMS OTP ได้

2.3 ผลการวิเคราะห์ Social Engineering

Social Engineering[2] หรือที่เรียกว่า การต้มตุ๋นหลอกลวงโดยใช้เทคนิคทางจิตวิทยา และวิธีนี้ก็มักจะได้ผลเสมอ รวมถึงคดีการโจรกรรม Online Banking ที่เริ่มมีข่าวออกมาเรื่อยๆ ตั้งแต่ช่วงกลางปี พ.ศ. 2556 ที่ผ่านมา ว่าคนร้ายได้ใช้เทคนิคนี้ในการโจรกรรมดังกล่าว ซึ่ง Social Engineering ไม่ได้มีวิธีการที่ตายตัว วิธีการที่พบเห็นบ่อยเช่น การโทรศัพท์มาหลอกลวงเพื่อให้เปิดเผยข้อมูลสำคัญ การหลอกลวงผ่านอินเทอร์เน็ต (เว็บไซต์ Email หรือแชท) การค้นข้อมูลจากเว็บไซต์ เอกสาร สิ่งของ หรือจากถังขยะของบุคคลหรือองค์กรเป้าหมาย การแอบสังเกตขณะเหยื่อกำลังป้อนข้อมูล เป็นต้น ส่วนวิธีการที่เป็นข่าวดังที่ถูกเผยแพร่บนโลกออนไลน์นั้นก็คือ การปลอมตัวเป็นคนอื่นหรือการสวมรอยเป็นเหยื่อนั่นเอง ด้วยการสร้างหลักฐานปลอม โดยคนร้ายจะนำเอาหลักฐานปลอมนี้ไปใช้ในการขอ SIM ใหม่และเปิดบัญชีใหม่ในธนาคารเดียวกันกับเหยื่อ ซึ่งการออก SIM ใหม่นี้ นอกจากจะทำให้เหยื่อโทรออกไม่ได้แล้ว คนร้ายยังได้รับ SMS OTP ที่ธนาคารส่งมาให้อีกข่าวของคดีที่เกิดขึ้นจริงนี้[20] ปรากฏดัง Figure 13 และ Figure 14

Figure 13 News on Bank Hacking by using Social Engineering

วิธีการสวมรอยเป็นคนอื่น (เหยื่อ) โดยใช้การสร้างหลักฐานปลอมขึ้นมานั้น ถือเป็นวิธีล่าสุดที่คนร้ายใช้ในการโจรกรรมเงินใน Online Banking ถึงแม้ว่าเหยื่อจะป้องกันดีแล้วก็ตาม ยังมีสิทธิ์โดนได้ เนื่องจากขึ้นอยู่กับหลายปัจจัย โดยเฉพาะพนักงานธนาคารและเจ้าหน้าที่ต่างๆ ที่เกี่ยวข้องล้วนจำเป็นต้องตรวจสอบหลักฐานของลูกค้าให้ดีเสียก่อนที่จะดำเนินการในขั้นตอนต่อไป จากคดีที่เกิดขึ้นจริง คนร้ายได้สร้างหลักฐานปลอมที่เป็นเอกสารราชการ ดัง Figure 14 ตอนนี้คนร้ายได้ถูกจับกุมตัวดำเนินคดีแล้ว ข้อมูลที่ได้มาคือ คนร้ายได้ก่อเหตุมาแล้ว 4 ครั้ง ทำงานกัน 2 คน วิธีการที่ใช้คือ (1) เก็บสลิปถอนเงินตามตู้ ATM ของธนาคารต่างๆ ดูว่าสลิปใดมีเงินเหลือมาก (2) หารายชื่อเจ้าของบัญชีจากเลขที่บัญชีที่ปรากฏในสลิปจากอินเทอร์เน็ต หรือจากบัญชีของคนรู้จักที่เคยมีการโอนเงินมา (3) ปลอมบัตรข้าราชการเป็นชื่อเจ้าของบัญชี แล้วไปขอข้อมูลทะเบียนราษฎร์ เพื่อทราบวันเดือนปีเกิด (4) นำบัตรข้าราชการปลอมไปเปิดบัญชีใหม่กับธนาคารเดียวกันกับเจ้าของบัญชีแต่ต่างสาขา (5) เมื่อเปิดบัญชีได้สำเร็จก็สมัคร Online Banking โดยระบุขอรวมบัญชีของเจ้าของบัญชีตัวจริงเข้ามาใช้บริการด้วย (6) เมื่อขอรวมบัญชีได้แล้วก็ทำการโอนเงินจากบัญชีตัวจริงของเหยื่อมายังบัญชีใหม่ผ่าน Online Banking จากนั้นก็ใช้บัตร ATM เบิกถอนเป็นเงินสดออกไป กรณีนี้คนร้ายได้เงินไป 4 แสนบาท

Figure 14 an Example of fake evidence document

จากการทดลองและวิเคราะห์ปัญหาของ SMS OTP ซึ่งเป็นรูปแบบที่ธนาคารในประเทศไทยส่วนใหญ่ใช้ พบว่าปัจจุบันมีความเสี่ยงสูงมาก มีข่าวคดีโจรกรรมออกมาอย่างต่อเนื่องตั้งแต่ปี พ.ศ. 2556 ซึ่งหากยังคงใช้รูปแบบนี้ต่อไปในอนาคตจึงจำเป็นอย่างยิ่งที่ต้องเพิ่มระบบการรักษาความมั่นคงให้มากขึ้น ดังตัวอย่างงานวิจัยที่ได้กล่าวไว้ในข้างต้น

3. ผลการวิเคราะห์ปัญหาของ Token OTP

สำหรับธนาคารแล้ว การใช้อุปกรณ์ Token OTP ในการรับรหัส OTP นั้นถือได้ว่าเป็นรูปแบบที่มีความมั่นคงมากที่สุดในขณะนี้ เนื่องจากผู้ใช้ไม่ต้องรอรับ OTP ที่ส่งมาผ่านระบบเครือข่ายใดๆ อีกต่อไป เพียงแค่ใช้งานอุปกรณ์ Token OTP ที่ได้มา ส่งผลให้ Hacker ไม่สามารถดักจับ OTP ได้ และหากมีการนำ Trojan ไปฝังอยู่บนอุปกรณ์ได้สำเร็จ ก็ไม่เกิดผลอะไร เพราะอุปกรณ์นี้ไม่สามารถสื่อสารไปยังภายนอกได้ ส่วนวิธีการใช้งานและการแสดงผลของ OTP จะขึ้นอยู่กับอุปกรณ์นั้นๆ และ OTP Algorithm ที่ใช้

จากการวิเคราะห์ข้อมูลของอุปกรณ์ Token OTP จากการศึกษา สำรวจ และสอบถามจากผู้รู้และผู้ที่ใช้งานจริง พบว่ามีปัญหาหลักดังนี้ (1) Cost กล่าวคือ ราคาของอุปกรณ์ Token OTP มีราคาสูงมากเริ่มตั้งแต่หลักพันเป็นต้นไป ทำให้ธนาคารและบริษัทส่วนมากไม่กล้าเสี่ยงลงทุนซื้อมาใช้งาน และลูกค้าส่วนใหญ่ก็ไม่มีกำลังซื้อหรือไม่ต้องการที่จะเสียเงินเพิ่ม (2) Carry กล่าวคือ เป็นการเพิ่มภาระการพกพาอุปกรณ์ให้แก่ผู้ใช้ ถ้าหากลืมก็จะไม่สามารถเข้าใช้งานระบบได้ ต่างจากโทรศัพท์มือถือที่จำเป็นต้องพกพาติดตัวอยู่แล้ว ไม่ได้เป็นการเพิ่มภาระแต่อย่างใด

Figure 15 Various Token OTPs

จาก Figure 15 คือภาพของอุปกรณ์ Token OTP ในรูปแบบต่างๆ ที่พบในปัจจุบัน มีตั้งแต่อุปกรณ์ที่ไม่มีลูกเล่นอะไรเลยไปจนถึงเป็นแบบ Card แผ่นบางๆ บ้างก็ผลิตออกมาในรูปอุปกรณ์ Flash Drive เพื่อให้ได้ใช้ประโยชน์อีกทาง ซึ่งแต่ละแบบก็มีราคาที่แตกต่างกันไป เช่น RSA SecurID 900 (เป็นแบบ Card)ราคาอยู่ที่ 282 US Dollar หรือประมาณ 9,200 บาท ต่ออุปกรณ์ 5 ชิ้น หรือคิดราคาต่อชิ้นประมาณ 1,840 บาทดัง Figure 16

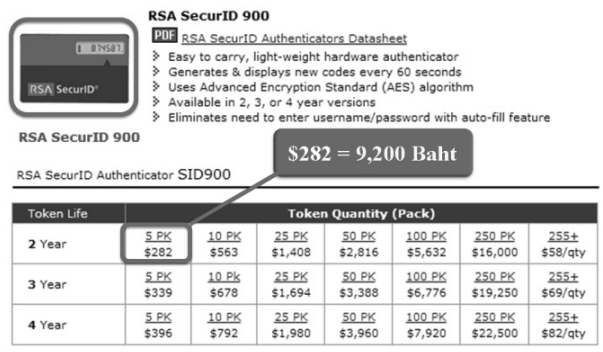


Figure 16 ราคาของอุปกรณ์ RSA SecurID 900

4. ผลการทดลองและวิเคราะห์ปัญหาของ Mobile OTP

Mobile OTP ถือว่าเป็นรูปแบบใหม่ล่าสุดสำหรับการใช้งาน OTP ทำหน้าที่เสมือนเป็นอุปกรณ์ Token OTP ตัวหนึ่ง โดยจะทำงานอยู่บน Smartphone ในลักษณะของ Mobile app อาจเรียกในอีกชื่อว่า Software Token OTP คาดว่าออกแบบมาเพื่อแก้ปัญหา OTP รูปแบบอื่นๆ เช่น ลดภาระค่าใช้จ่ายและการพกพาอุปกรณ์ Token OTP เพิ่ม ไม่ต้องรอรับรหัส OTP ที่ส่งมาผ่านระบบเครือข่าย ซึ่งเป็นการลดความเสี่ยงที่จะถูกดักจับข้อมูล เป็นต้น ในอนาคต Mobile OTP อาจถูกนำมาใช้งานแทนที่ตัวอื่นๆ ปัจจุบันมีธนาคารบางประเทศ

ได้นำมาใช้กับระบบ Online Banking เช่น ธนาคาร Union ในประเทศอินเดีย แต่ยังไม่พบการนำมาใช้งานกับธนาคารในประเทศไทย พบเพียงการนำมาใช้กับบริษัทดังๆ เช่น Google (Google Authenticator) ดัง Figure 17 และ Facebook (Code Generator) ดัง Figure 18 ที่มีจุดประสงค์คือเพิ่มความมั่นคงในการเข้าถึงบัญชีผู้ใช้และอำนวยความสะดวกแก่ผู้ใช้ในการรับรหัส OTP ทั้งยังสามารถป้องกันการดักจับข้อมูล OTP ได้ในระดับหนึ่ง งานวิจัยนี้จึงได้ทำการทดลองใช้งาน Google Authenticator เพื่อวิเคราะห์ปัญหา พร้อมทั้งพิจารณาจุดแข็งและจุดอ่อน

เนื่องจาก Mobile OTP ทำงานบน Smartphone ดังนั้นปัญหาสำคัญอย่างหนึ่งที่มีโอกาสจะได้พบเจอก็คือ ความเสี่ยงที่จะติด Malware อย่าง Trojan Horse นั่นเอง โดย Trojan ที่น่ากลัวมากที่สุดสำหรับ Mobile OTP ก็คือ Trojan ประเภทขโมยข้อมูลแล้วส่งต่อไปยัง Hacker ที่มีลักษณะใกล้เคียงกับ Trojan ของ SMS OTP ที่ได้กล่าวมาก่อนหน้านี้ แต่ Mobile OTP จะมีความเสี่ยงน้อยกว่า

Figure 17 the Google Authenticator of Google

Figure 18 the Code Generator of Facebook

Google ได้เปิดตัวระบบ 2-Step Verification เมื่อ20 กันยายน พ.ศ. 2553 ที่ผ่านมา[7] โดยนำระบบ OTP มาใช้ในการยืนยันตัวตนขั้นตอนที่สอง ซึ่งมีหลากหลายช่องทางในการขอรับรหัส OTP ให้ผู้ใช้เลือกได้แก่ SMS OTP, Voice Call และ Mobile OTP (Google Authenticator) ที่อำนวยความสะดวกแก่ผู้ใช้ได้มากที่เดียว จากการทดลองใช้ Google Authenticator พบว่า (1) OTP Algorithm ที่ Google Authenticator ใช้คือ Time-based OTP ผู้ใช้จึงจำเป็นต้องเปิดใช้งานอินเทอร์เน็ตในการเข้าใช้งานครั้งแรกเพื่อทำการตั้งค่า Account และเพื่อให้ app ทำการ Sync เวลาของ Client กับ Server ให้ตรงกันหรือเข้าจังหวะกัน (2) รหัส OTP จะเปลี่ยนใหม่ทุกๆ 30 วินาที และไม่ซ้ำกัน นั่นหมายความว่า app นี้สามารถสร้าง OTP ที่ไม่ซ้ำกันเลยภายในระยะเวลา 11 เดือนครึ่ง (3) มีรหัสผ่านสำรองเตรียมไว้ให้กรณีผู้ใช้ลืมโทรศัพท์มือถือ แบตหมด หรือว่าทำหายปรากฏอยู่บนเว็บไซต์ภายใต้บัญชีของผู้ใช้ มีให้ทั้งหมด 10รหัสผ่าน ใช้ได้หนึ่งครั้งต่อรหัส ผู้ใช้สามารถให้ระบบสร้างขึ้นใหม่ได้เรื่อยๆ เมื่อต้องการหรือเมื่อใช้จนครบ วิธีนี้ถือว่าเป็นวิธีแก้ปัญหาที่ดี แต่ไม่ควรใช้กับระบบ Online Banking (4) รหัส OTP จำนวน 9 รหัสล่าสุด (รวมรหัสที่กำลังแสดงบนหน้าจอของ app) สามารถใช้ Login เข้าระบบได้ทั้งหมด แต่เมื่อได้ใช้รหัสตัวใดตัวหนึ่งแล้ว รหัส OTP ตัวที่อยู่ลำดับถัดไปจากตัวที่ใช้ไปจะถูกปรับเป็นรหัสล่าสุดลำดับที่ 1 ในทันที หมายความว่ารหัส OTP ตัวที่อยู่ลำดับก่อนหน้าของตัวที่ใช้ไปจะใช้ไม่ได้อีกต่อไป เช่น รหัส OTP 9 รหัสล่าสุดคือ 000001-000009 และได้ใช้รหัส 000003 ไป ดังนั้นจึงเหลือรหัสล่าสุดที่ยังใช้ได้คือ 000004-000009 (เหลือเพียง 6 รหัสล่าสุด) ประโยชน์ของการใช้เทคนิควิธีนี้ก็คือ เพื่อแก้ปัญหาการเกิดเหตุการณ์ Out of Synchronization ซึ่งจะได้กล่าวถึงในหัวข้อผลการวิเคราะห์ OTP Algorithm ต่อไป

Mobile OTP ส่วนใหญ่นิยมใช้ Time-based OTPหรือไม่ก็ Counter-based OTP เป็น OTP Algorithm แต่มีจุดอ่อนคือจะถูก Brute-force attack ได้ง่าย เนื่องจากผลของ OTP ที่ได้เป็นเพียงแค่ตัวเลข 6-8 หลัก หากเทียบกับ Challenge-Response OTP ที่มีผลลัพธ์เป็นคำศัพท์ภาษาอังกฤษจำนวน 6 คำ (มีความยาวรวม 6-24 ตัวอักษร) ซึ่งถูก Brute-force attack ได้ยากกว่า จึงเป็น OTP Algorithm ที่เหมาะกับ Mobile OTP มากกว่า เนื่องจาก Mobile app สามารถประมวลผลการทำงานได้อย่างซับซ้อน นอกเหนือจากการแสดงผลบนหน้าจอเพียงอย่างเดียว

Table 1 Pros and Cons of OTPs

รูปแบบ OTP	ข้อดี	ข้อเสีย	สถานะการใช้งาน
Email OTP	- ฟรี - เข้าถึงได้จากทุกที่ที่มี Internet	- Email spoofing - Email sniffing - ใช้งานไม่สะดวก	ธนาคารทุกแห่งในประเทศไทยยกเลิกใช้งานแล้ว
SMS OTP	- สะดวก - ใช้งานง่าย - มือถือทุกรุ่นรองรับการทำงาน	- เสียค่าส่ง SMS - SMS spoofing - ปัญหา Trojans - วิศวกรรมสังคม	นิยมนำไปใช้งานมากที่สุด โดยเฉพาะกับธนาคารในประเทศไทย
Token OTP	- มีความมั่นคงมากที่สุดขณะนี้	- ราคาแพง - เพิ่มภาระพกพา	หลายประเทศนำมาใช้กับธนาคารมากขึ้น
Mobile OTP	- ไม่ต้องพกพาอุปกรณ์เพิ่ม - ใช้แทน TokenOTP	- ปัญหาTrojans - มือถือรุ่นเก่าไม่รองรับการทำงาน (Smartphone only)	เริ่มมีการนำมาใช้กับบริษัท (Google,Facebook) บางประเทศนำมาใช้กับธนาคารแล้ว

จาก Table 1 เป็นข้อมูลการเปรียบเทียบข้อดีและข้อเสียของ OTP รูปแบบต่างๆ ซึ่งได้แก่ Email OTP, SMS OTP, Token OTP และ Mobile OTP สามารถสรุปได้ดังนี้ (1) ปัจจุบันธนาคารทุกแห่งในประเทศไทยยกเลิกการใช้งาน Email OTP แล้ว เนื่องจากมีช่องโหว่ที่จะโดนโจมตีอยู่มากมาย (2) SMS OTP ถือว่าเป็นรูปแบบที่นิยมนำมาใช้งานมากที่สุดในขณะนี้ มีข้อดีหลายอย่าง ในอดีตนับว่ามีความมั่นคงสูงมาก แต่ปัจจุบันพบว่ามีความเสี่ยงอยู่ไม่น้อยเลยทีเดียว (3) อุปกรณ์ Token OTP ถึงแม้ว่ามีความมั่นคงมากที่สุดในขณะนี้ แต่เนื่องจากมีราคาที่สูงมากทั้งยังต้องพกพาอุปกรณ์เพิ่ม จึงไม่ได้รับความนิยมมากนัก (4) คาดว่าในอนาคต Mobile OTP จะถูกนำมาใช้งานแทน Token OTP และอาจมาแทน SMS OTP ด้วยเช่นกัน ทั้งนี้ผู้ใช้ Smartphone จำเป็นต้องระมัดระวังเรื่องของ Trojans และที่สำคัญคือผู้พัฒนา Mobile app ควรพัฒนาโปรแกรมให้มีประสิทธิภาพ มีความมั่นคงสูงและมีระบบ Verify App เพื่อป้องกันการปลอมแปลง app ส่วนข้อจำกัดที่กล่าวว่าใช้กับ Smartphone เท่านั้นคงไม่ใช่ปัญหาอีกต่อไป เพราะในปัจจุบัน Smartphone ได้ถูกผลิตออกมาเป็นจำนวนมากและมีให้เลือกมากมายหลายรุ่นหลายราคา เริ่มตั้งแต่ราคาไม่ถึงสามพันบาทไปจนถึงหลายหมื่นบาท จากผลสำรวจการใช้งาน Smartphone ทั่วโลก[21] พบว่า ส่วนใหญ่เลือกใช้ Smartphone ที่เป็นระบบปฏิบัติการ Android

5. ผลการวิเคราะห์ปัญหาของ OTP Algorithm

จาก OTP Algorithm 3 รูปแบบ ที่ได้กล่าวมาแล้วพบว่า Time-based OTP คือรูปแบบที่ได้รับความนิยมนำมาใช้งานมากที่สุด รองลงมาคือ Event-based OTP หรือ Counter-based OTP และที่ถูกนำมาใช้น้อยที่สุดก็คือแบบ Challenge-Response OTP ทั้งที่มีความมั่นคงมากที่สุดอาจเป็นเพราะเหตุผลว่าสองอันดับแรกมีผลลัพธ์ของ OTP เป็นตัวเลขจำนวน 6-8 หลัก ทำให้ง่ายต่อการจดจำและนำไปกรอกข้อมูล ช่วยลดโอกาสกรอกรหัส OTP ที่ผิดพลาดของผู้ใช้ เป็นต้น อย่างไรก็ตาม ทุก Algorithm ที่กล่าวมานี้ล้วนยังมีปัญหาหรือจุดอ่อนอยู่ งานวิจัยนี้จึงได้ลงมือศึกษาและทำการวิเคราะห์ปัญหาของ OTP Algorithm ดังกล่าว

5.1 Event-based OTP หรือ Counter-based OTP พบปัญหาดังนี้ (1) เกิดปัญหา Out of Synchronization หรือที่เรียกว่า การหลุดการทำงานที่ผสานงานกัน ในที่นี้คือค่าของ Counter ซึ่งหากมีค่าไม่ตรงกันระหว่างฝั่ง Client กับฝั่ง Server จะส่งผลให้ OTP ที่ได้มาจากการประมวลผลด้วย Algorithm เดียวกันมีผลลัพธ์ไม่ตรงกันจึงทำให้การยืนยันตัวตนเกิดความผิดพลาดได้ (2) มีผลลัพธ์ OTP เป็นตัวเลขเพียงแค่ 6-8 หลักเท่านั้น ทำให้ง่ายต่อการถูก Brute-forceat-tack และด้วยความเสี่ยงนี้เองจึงนำไปสู่ Trial and Error Check[22] ที่มีค่าน้อย ซึ่งก็คือจำนวนครั้งที่ผู้ใช้สามารถกรอก OTP ผิดได้จะลดน้อยตามไปด้วย ที่พบบ่อยคือ 3 ครั้ง หากผู้ใช้กรอก OTP ผิดเกินกว่าที่กำหนด ระบบจะมองว่ากำลังถูกโจมตีด้วยวิธี Brute-force attack จากนั้นระบบส่วนใหญ่จะใช้วิธีเพิ่มการตรวจสอบการกรอกรหัสผ่านว่ากระทำโดยมนุษย์จริงหรือไม่ โดยการใช้เทคนิคที่เรียกว่า CAPTCHA[23] จึงเป็นการเพิ่มภาระให้แก่ผู้ใช้ ส่งผลให้การยืนยันตัวตนเกิดความล่าช้าหรืออาจไม่สำเร็จได้

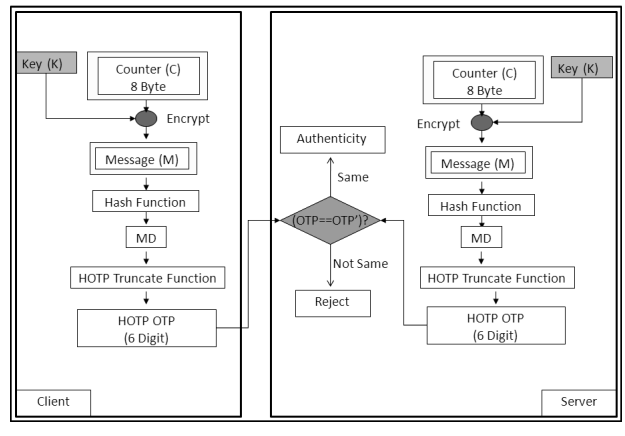


Figure 19 Procedures of HOTP

จาก Figure 19 คือกระบวนการทำงานของ HOTP หรือก็คือ Counter-based OTP นั่นเอง โดยมีขั้นตอนคร่าวๆ ดังต่อไปนี้ (1) Client จะเก็บกุญแจหลัก (Master Key) ไว้ เพื่อใช้ในขั้นตอนการเข้ารหัสแบบ AES[24] (2) Client ทำการสร้างตัวนับ (Counter) ขึ้นมา มีค่าเริ่มต้นเป็น 1 และ

จะเพิ่มขึ้นทุกครั้งที่มีการร้องขอ OTP (3) นำ Key ในขั้นตอนที่ 1 มาเข้ารหัสข้อมูล Counter ได้ Message (4) นำ Message ที่ได้เข้าสู่กระบวนการ Hash เช่น SHA-1[25] จะได้ Message Digest (MD) ที่มี output ขนาด 160 bits (5) นำ MD ที่ได้ เข้าสู่ฟังก์ชัน HOTP Truncate เพื่อลดขนาด output เหลือ 32 bits จากนั้นนำไปแปลงเป็นตัวเลข 6-8 หลักจะได้ OTP เพื่อใช้ส่งไปยัง Server (6) เมื่อ Server ได้รับ OTP แล้วก็จะนำ OTP ที่ได้มาเปรียบเทียบกับ OTP ที่ Server สร้างขึ้นจากกระบวนการเดียวกันกับ Client ถ้าผลการเปรียบเทียบออกมาตรงกันก็แสดงว่าผลการยืนยันตัวตนถูกต้อง ซึ่งจากกระบวนการดังกล่าว จะเห็นได้ว่าค่าของ Counter มีโอกาสเกิดความผิดพลาดได้ เช่น เมื่อผู้ใช้มีการร้องขอ OTP ใหม่ แต่ค่าของ Counter กลับไม่เพิ่มขึ้น ก็จะส่งผลให้เกิดปัญหา Out of Synchronization ตามมา

5.2 Time-based OTP พบปัญหาดังนี้ (1) ปัญหา Out of Synchronization เช่นเดียวกับ Counter-based OTP แต่กรณีนี้จะเป็นค่าของเวลาที่ไม่ตรงกันหรือไม่เข้าจังหวะกันระหว่างฝั่ง Client กับฝั่ง Server ส่งผลให้ OTP ที่ได้ออกมามีค่าไม่ตรงกันทำให้การยืนยันตัวตนเกิดความผิดพลาดได้ (2) ผลลัพธ์ของ OTP เป็นตัวเลขอย่างเดียว และมีเพียง 6-8 หลักเท่านั้นทำให้ง่ายต่อการถูก Brute-force attack จึงนำไปสู่ Trial and Error Check ที่มีค่าน้อย ดังที่ได้อธิบายไว้ในผลการวิเคราะห์ปัญหาของ Counter-based OTP

5.3 Challenge-Response OTP เช่น S/Key OTP พบปัญหาดังนี้ (1) ผลลัพธ์ OTP เป็นตัวอักษรภาษาอังกฤษที่มีโอกาสเรียงกันยาวสูงสุดถึง 24 ตัวอักษร ซึ่งเป็นการเพิ่มภาระให้แก่ผู้ใช้ ทั้งยังส่งผลให้มีโอกาสกรอก OTP ผิดพลาดได้ง่าย (2) กระบวนการที่ได้มาซึ่ง OTP นั้น ยุ่งยากซับซ้อนพอสมควร จำเป็นต้องให้ผู้ใช้ออกแรงช่วยจึงจะครบกระบวนการ เช่น ผู้ใช้ต้องนำเอาค่า Seed ที่ได้รับจาก Server ไปใช้ดำเนินการต่อเพื่อสร้างเป็นรหัส OTP ด้วย Algorithm ของ S/Key OTP แล้วจึงจะได้นำ OTP ไปใช้จริง แสดงให้เห็นว่าผู้ใช้ต้องออกแรงเพิ่มอย่างน้อยสองเท่าหากเทียบกับแบบอื่น ปัญหานี้สามารถแก้ได้ด้วยการพัฒนา app ที่ช่วยผ่อนแรงผู้ใช้ เช่น นำเทคโนโลยี QR Code มาช่วยสแกนค่า Seed

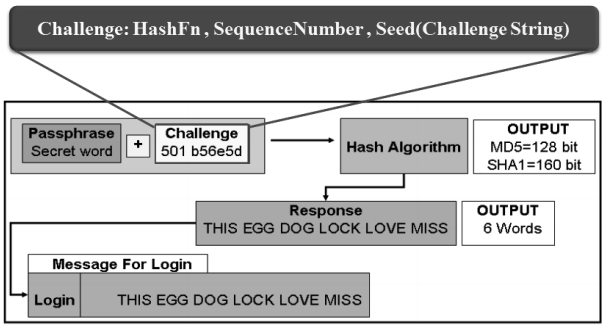


Figure 20 Procedures of S/Key OTP

จาก Figure 20 คือกระบวนการทำงานของ S/Key OTP แบบย่อ มีขั้นตอนวิธีดังนี้ (1) นำ Passphrase และ Challenge มาเชื่อมอักขระเข้าด้วยกันโดย Passphrase คือสิ่งที่ผู้ใช้กำหนดขึ้นมาเองหรือโปรแกรมกำหนดมาให้ มีลักษณะคล้ายกับ Password และต้องเก็บไว้เป็นความลับ ส่วน Challenge ประกอบไปด้วย Hash Function (MD4[26], MD5[27], SHA-1[25]), Sequence Number และ Seed (เช่น Challenge String, Challege Number) ซึ่ง Seed เป็นสิ่งที่ Server จะส่งมาให้ Client (2) นำข้อความจากขั้นตอนที่ 1 มาผ่านกระบวนการ Hash ตามที่กำหนดไว้ใน Challenge เช่น หากกำหนดเป็น SHA-1 ผลลัพธ์ที่ได้ก็จะเป็น MD ที่มีขนาด 160 bits (3) นำ MD ที่ได้ไปผ่านกระบวนการแปลงเป็นคำศัพท์ภาษาอังกฤษจำนวน 6 คำ โดยอ้างอิงจากฐานข้อมูลคำศัพท์ที่มีอยู่ทั้งหมด 2048 คำ แต่ละคำมีความยาวตั้งแต่ 1-4 ตัวอักษร จากนั้นจะได้ OTP ปรากฏอยู่ที่ Client (5) ผู้ใช้นำ OTP ที่ได้ไปกรอกเพื่อผ่านกระบวนการยืนยันตัวตนตัว อย่างผลลัพธ์ของ S/Key OTP เมื่อใช้ Challenge ที่แตกต่างกัน ดัง Table 2 แสดงให้เห็นว่าแม้มีเพียงหนึ่งค่าที่แตกต่างกัน ไม่ว่าจะเป็น Passphrase, Seed หรือ Count (Sequence Number) ก็ส่งผลทำให้ผลลัพธ์ของ OTP ที่ได้ออกมา มีความแตกต่างกันโดยสิ้นเชิง

Table 2 an Example of S/Key OTP

Passphrase	Seed	Count	Output (Hex)	Output (Six Word Format)
This is a test.	TeSt	0	BB9E 6AE1 979D 8FF4	MILT VARY MAST OK SEES WENT
This is a test	TeSt	1	63D9 3663 9734 385B	CART OTTO HIVE ODE VAT NUT
This is a test	TeSt	99	87FE C776 8B73 CCF9	GAFF WAIT SKID GIG SKY EYED
AbCdEfGhIjK	alpha1	0	AD85 F658 EBE3 83C9	LEST OR HEEL SCOT ROB SUIT
AbCdEfGhIjK	alpha1	1	D07C E229 B5CF 119B	RITE TAKE GELD COST TUNE RECK
AbCdEfGhIjK	alpha1	99	27BC 7103 5AAF 3DC6	MAY STAR TIN LYON VEDA STAN

Table 3 Comparison of S/Key OTP and Base 64 S/Key OTP

Input Data		OTP Output	
type	seed	S/key OTP	BASE64 S/Key OTP
Counter	0	HOB SIT POW NET NUT NINA	G0eQ05XC2Wk=
	1	KUDO VARY BEND BELL COLD BOTH	mHxpWKr2nrs=
	2	AREA GIG HUT NASH KERR BEN	TeLcc1kJYg0=
	3	OVER MEN DRUB THEY DOOR DUKE	umUF3G03VvA=
	4	MAP MOE QUIT OLGA WING REAL	J0Um69v+w3o=
Time	1970-01-01 00:00:59	SOAR DOSE SOAK SEA GAL AQUA	0S63RB2BXJs=
	2011-12-20 01:58:29	WANE ONTO MULE TOTE VINE BOLO	5xcewW8+ULY=
	2011-12-20 01:58:30	PAN MUTE NICE NODE GORY MY	MfYe0Np48FQ=
	2011-12-20 01:58:31	BIG ARMY EDGY YALE REEF	B57tOT0+6Xs=
	2011-12-20 01:58:32	BOMB ROOF GAD NAIR ALIA	W7g4VljUr+4=
Challenge	90552072	WHAM JO TRIO OUST JUNK KICK	6oQPf1z5Qyw=
	32546588	BURL OVA MAO WALL ACTA	YKYXyROeapE=
	12654858	LETS DING EEL SELL LETS	9XO1zYhskzs=
	56854254	WAVY FOWL BLUM CRAM RIFT	6JC72qz2w38=
	75568751	GLAD JUG GIRD JUNK SHOW	jEQmL33JQ5Y=

จาก Table 3 แสดงถึงการแก้ปัญหา S/Key OTP ด้วย Base64 Encoding โดยปรัชญา ไชยเมือง และคณะ[13] ได้ทำการปรับปรุง S/Key OTP ที่มีปัญหาด้านความยาวของ OTP ที่เป็นไปได้สูงสุดถึง 24 ตัวอักษร โดยใช้วิธีแปลงรหัสแบบ Base64 เพื่อลดจำนวนตัวอักษรลงเหลือ 12 ตัวอักษร แต่ยังคงรักษาคุณภาพความมั่นคงไว้เท่าเดิม และได้ทดสอบประสิทธิภาพของ Base64 S/Key OTP พบว่าจำนวนครั้งของการกรอกรหัส OTP ที่ผิดพลาดของผู้ใช้ลดน้อยลงเมื่อเทียบกับ S/Key OTP แบบปกติ

สรุปผลและข้อเสนอแนะ

One Time Password เป็นเครื่องมือและเป็นกลไกสำคัญที่ช่วยเพิ่มการรักษาความมั่นคงให้กับการเข้าใช้งานระบบสารสนเทศที่สำคัญ ๆ โดยเฉพาะ Online Banking และ E-commerce อย่างไรก็ตาม แม้ว่ามีระบบ OTP เข้ามาช่วย ก็ยังมีข่าวการโจรกรรมเงินใน Online Banking ออกมาอยู่เรื่อย ๆ งานวิจัยนี้จึงทำการวิเคราะห์หาจุดแข็งและจุดอ่อนของระบบ OTP เพื่อให้เห็นปัญหาที่พบใน OTP แต่ละชนิด ข้อควรระวังเกี่ยวกับการใช้งาน OTP และ Online Banking รวมถึงการนำเสนอแนวทางการแก้ปัญหาและปรับปรุงระบบ OTP ให้มีประสิทธิภาพและมีความมั่นคงมากขึ้น

แนวโน้มของเทคโนโลยี OTP ในอนาคต คาดว่า Mobile OTP น่าจะเป็นรูปแบบที่ถูกนำมาใช้งานแทนรูปแบบอื่นๆ เนื่องด้วยเหตุผลหลายประการ เช่น ผู้ใช้ไม่จำเป็นต้องพกพาอุปกรณ์เพิ่ม ไม่ต้องเสียเงินซื้ออุปกรณ์ Token OTP ราคาแพง ลดความเสี่ยงที่อาจเกิดขึ้นจาก SMS spoofing และ Trojans เป็นต้น ส่วน OTP Algorithm ที่เหมาะสมกับระบบ Online Banking ก็คือ Challenge-Response OTP เพราะมีความมั่นคงสูงกว่าแบบอื่น แต่ควรได้รับการแก้ไขและปรับปรุงในส่วนที่ยังมีปัญหาให้ดีขึ้นเสียก่อน

กิตติกรรมประกาศ

โครงการวิจัยนี้ ได้รับทุนอุดหนุนการวิจัยจากงบประมาณรายได้ประจำปีงบประมาณ 2557 มหาวิทยาลัยมหาสารคาม และทุนอุดหนุนการวิจัยจากสำนักงานคณะกรรมการวิจัยแห่งชาติ

เอกสารอ้างอิง

1. Haller N, Metz C, Nesser P, Straw M. A One-Time Password System. IETF, RFC 2289, February 1998.
2. จตุชัย แพงจันทร์. Master in Security 2nd Edition. พิมพ์ครั้งที่ 1. นนทบุรี: บริษัท ไอดีซี พรีเมียร์ จำกัด; 2553.
3. Krawczyk H, Bellare M, Canetti R. HMAC: Keyed-Hashing for Message Authentication. IETF, RFC 2104, Feb 1997.
4. Raihi D, Bellare M, Hoornaert F, Naccache D, Ranen O. HOTP: An HMAC-Based One-Time Password Algorithm. IETF, RFC 4226, December 2005.
5. Raihi D, Machani S, Pei M, Rydell J. TOTP: Time-Based One-Time Password Algorithm. IETF, RFC 6238, May 2011.
6. Haller N. The S/KEY One-Time Password System. IETF, RFC 1760, February 1995.
7. Feigenbaum E. A more secure cloud for millions of Google Apps users. [online]. September 2010 [cited 18 April 2014]; http://googleenterprise.blogspot.com/2010/09/more-secure-cloud-for-millions-of.html.
8. A Few Updates to Make Your Mobile Experience More Safe and Secure. [online]. 7 June 2012 [cited 17 April 2014]; https://www.facebook.com/notesfacebook-security/a-few-updates-to-make-your-mobile-experience-more-safe-and-secure/10150839779545766.
9. ธนาคารแห่งประเทศไทย. เพิ่มการรักษาความมั่นคงปลอดภัย ด้วยการใช้ Two-Factor Authentication. [online]. May 2008 [cited 3 May 2014]; http://www.bot.or.th/Thai/PaymentSystems/Publication/ps_pamphlet/Pages/Pamphlet_May2008.aspx.
10. Kaspersky Lab. ZeuS-in-the-Mobile – Facts and Theories. [online]. October 2011 [cited 24 March 2014]; https://www.securelist.com/en/analysis/204792194/ZeuS_in_the_Mobile_Facts_and_Theories.

11. Kaspersky Lab. Android Trojan Found in Targeted Attack. [online]. March 2013 [cited 24 March 2014]; https://www.securelist.com/en/blog/208194186.
12. Mimoso M. Android Banking Trojan Svpeng Goes Phishing. [online]. 2013 [cited 24 March 2014]; http://threatpost.com/android-banking-trojan-svpeng-goes-phishing/102822.
13. ปรัชญา ไชยเมือง, สุชาติ คุ้มมะณี, สมนึก พ่วงพรพิทักษ์. การปรับปรุงรหัสผ่านใช้ครั้งเดียวแบบเอสคีย์โดยใช้การแปลงรหัสข้อมูลแบบฐาน 64. วารสารเทคโนโลยีสารสนเทศ กรกฎาคม-ธันวาคม 2555; 8[2]: หน้า 1-7.
14. Mulliner C, Borgaonkar R, Stewin P, et al. SMS-Based One-Time Passwords: Attacks and Defense. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA); 17-19 July 2013; Berlin. Springer; pp. 150-159.
15. ภูกิจ บุรีภักดี, ปราโมทย์ กั่วเจริญ.การเพิ่มความปลอดภัยและประสิทธิภาพในการรับส่งข้อความ SMS. The National Conference on Computing and Information Technology (NCCIT); 9-10 พฤษภาคม 2555;
16. Emkei's Instant Mailer. [online]. 2013 [cited 9 June 2013]; http://emkei.cz.
17. Fake Sms Sender 1.10. [online]. 2013 [cited 19 May 2013]; http://www.androidaapps com/c-65- communications/detail-34780-fake-sms-sender-1-10.html.
18. Welcome to faketext.net. [online]. 2013 [cited 19 May 2013]; http://www.faketext.net.
19. ระวังภัยจารกรรมในรูปแบบ SMS. [online].2013 [cited 8 May 2013]; http://www.it24hrs.com/2013/sms-fake-app-fake-mobile-banking.
20. เตือนภัย Internet Banking รูปแบบใหม่ ปลอมเป็นคุณด้วยหลักฐานปลอม สวมรอยโอนเงินออก สูญหลายแสน. [online].August 2013 [cited 6 May 2014]; http://www.it24hrs.com/2013/hack-otp-banking-change-new-sim-card.
21. Worldwide Smartphone Sales to End Users by Operating System in 3Q13. [online]. Nov 2013 [cited 21 Dec 2013]; https://www.gartner.com/newsroom/id/2623415.
22. Nahari H, Krutz R. Web Commerce Security: Design and Development. 1st ed. Indianapolis: Wiley, Inc; 2011.
23. Kang L, Xiang J. CAPTCHA Phishing: A Practical Attack on Human Interaction Proofing. International Conference on Information Security and Cryptology (Inscrypt); Beijing, China. pp. 411-425.
24. Schaad J, Housley R. Advanced Encryption Standard (AES) Key Wrap Algorithm. IETF, RFC 3694, September 2002.
25. Eastlake D, Jones P. US Secure Hash Algorithm (SHA1). IETF, RFC 3147, September 2001.
26. Rivest R. The MD4 Message-Digest Algorithm. IETF, RFC 1320, April 1992.
27. Rivest R. The MD5 Message-Digest Algorithm. IETF, RFC 1321, April 1992.