

การสร้างและการวิเคราะห์ Thai CAPTCHA

Thai CAPTCHA : Construction and Analysis

เกษรินทร์ ชาวเกวียน และ วรพล ลีลาเกียรติสกุล

คณะวิทยาการและเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีมหานคร

51 หมู่ 1 ถนนเชื่อมสัมพันธ์ แขวงกระทุ่มราย เขตหนองจอก กรุงเทพฯ 10530

Email: ketsarin48@hotmail.com , woraphon@gmail.com

บทคัดย่อ – เว็บไซต์ต่างๆ ในปัจจุบัน ที่เปิดโอกาสให้ผู้ใช้สามารถส่งข้อมูลไปยังเครื่องแม่ข่ายได้ เช่นเว็บไซต์ที่มีการสมัครสมาชิก หรือ เว็บไซต์ที่มีเว็บบอร์ด เป็นต้น เว็บไซต์เหล่านี้มักตกเป็นเป้าหมายของการโจมตีโดยการใช้บ็อต เพื่อสร้างข้อมูลที่คุณสมบัติบ็อต ต้องการไปยังเครื่องแม่ข่ายของเว็บไซต์ต่างๆ ซึ่งอาจจะเป็นการโพสต์ข้อความขายของ หรือ การสมัครสมาชิกปลอม เป็นต้น จากการสำรวจพบว่าบ็อตที่โจมตีตามเว็บไซต์ต่างๆ มาจากการภายนอกประเทศไทยเป็นหลัก ดังนั้นบทความนี้จึงนำเสนอเทคนิคของการสร้าง CAPTCHA โดยใช้ตัวอักษรภาษาไทย เพื่อที่จะแบ่งแยกผู้ใช้ที่เป็นคนไทยและผู้ใช้ที่เป็นบ็อต นอกจากนี้ยังมีการใส่การทำงานทางคณิตศาสตร์อย่างง่ายและใส่การรบกวนเพื่อเพิ่มประสิทธิภาพในการแบ่งแยกระหว่างผู้ใช้ที่เป็นคนและเครื่องคอมพิวเตอร์ และได้มีการวิเคราะห์ผลลัพธ์ของการใช้โปรแกรม OCR ที่ชื่อว่า Amthai เพื่อที่จะกำหนดคุณสมบัติของ CAPTCHA ที่นำเสนอให้มีประสิทธิภาพมากขึ้น

คำสำคัญ – แคปช่า, บ็อต

ABSTRACT – Currently, there are many websites that allow users to upload information to their servers such as member registration forms, webboard and etc. These websites are targeted by bots to create bogus information, i.e. sales or fake member registration, manipulated by bot-master. By surveying, we found that bots attacking on several websites always come from places outside Thailand. We therefore propose the new technique to construct CAPTCHA by using Thai characters to distinguish users that know Thai language from bot-machines. In addition, simple mathematic functions and background noise are inserted to enhance efficiency. We also analyze the results from Thai OCR which is named Amthai to define properties of proposed Thai CAPTCHA more efficiently.

KEYWORDS – CAPTCHA, Bot

1. บทนำ

ในปัจจุบัน เว็บไซต์ต่างๆ มีหน้าเว็บที่ต้องการให้ผู้ใช้กรอกข้อมูลเพื่อส่งไปยังเครื่องแม่ข่าย เพื่อประมวลผล หรือจัดเก็บต่อไป อย่างไรก็ตาม ได้มีความพยายามที่จะหาวิธีการที่ผู้ใช้ไม่ต้องทำการสร้างข้อมูล โดยอาศัยโปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์ ทำการสร้างข้อมูล แล้วทำการจัดส่งไปยังเครื่องแม่ข่ายโดยอัตโนมัติ โปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์เหล่านี้จะถูกคิดค้นอย่างตั้งใจหรือไม่ตั้งใจ (ไวรัสหรือ เวิร์ม) บนเครื่องคอมพิวเตอร์หนึ่งๆ และถูกควบคุมหรือสั่งการจากเครื่องคอมพิวเตอร์อีกเครื่องหนึ่งที่เป็นผู้สั่งการ เรามักเรียกเครื่องคอมพิวเตอร์ที่ถูกควบคุมการทำงานนี้ว่า บ็อต (Bot)

โดยปกติแล้ว บ็อต ถูกใช้อย่างแพร่หลายในหลายๆเรื่อง เช่นการช่วยค้นหาข้อมูลของการทำงานของเว็บไซต์กูเกิ้ล [1] การใช้บ็อตทำการเล่นเกมต่างๆ[2] แต่ปัญหาที่เกิดขึ้นส่วนใหญ่นั้น ก็คือการใช้บ็อตเพื่อปลอมตัวในการสมัครสมาชิกทางเว็บต่างๆ หรือ การใช้บ็อตสร้างข้อมูลบนเว็บบอร์ด หรือบล็อกเพื่อล่อลวงให้เชื่อมโยงไปยังเว็บไซต์ที่ผู้ควบคุมบ็อตต้องการ หรือการใช้บ็อตในการโจมตีแบบปฏิเสธการให้บริการ (Denial Of Service)

ดังนั้น CAPTCHA [3] ได้ถูกพัฒนาขึ้นเพื่อป้องกันการทำงานของบ็อต ในการส่งข้อมูลไปยังเว็บไซต์ต่างๆ เช่น Google, Ebay เป็นต้น อย่างไรก็ตามถึงงานวิจัย [4] ได้แสดงถึงจุดอ่อนของเทคนิคต่างบนเว็บที่ มีการใช้งานอยู่ในขณะนี้ อย่างไรก็ตามเมื่อสำรวจการทำงานของเว็บไซต์ต่างๆของประเทศไทยที่มีผู้ใช้เป็นคนไทย โดยส่วนใหญ่จะนำ CAPTCHA ที่สร้างจากต่างประเทศ ซึ่งมีความเป็นไปได้ที่บ็อต จะสามารถแก้ปัญหาได้ตามที่ได้กล่าวถึงใน [4] ดังนั้นบทความนี้จะนำเสนอการสร้างและวิเคราะห์ CAPTCHA ที่ใช้ตัวอักษรภาษาไทยเพื่อป้องกันการโจมตีจากบ็อต ที่มาจากต่างประเทศ

การจัดเรียงเนื้อหาของบทความเป็นดังต่อไปนี้ หัวข้อที่ 2 กล่าวถึงการทำงานพื้นฐานของ CAPTCHA และงานวิจัยที่เกี่ยวข้อง หัวข้อที่ 3 อธิบายแนวคิดและโครงสร้างของวิธีการที่นำเสนอ หัวข้อที่ 4 อธิบายถึงวิธีการทดลอง หัวข้อที่ 5 เสนอผลการทดลองและวิเคราะห์ หัวข้อที่ 6 เป็นบทสรุปและงานวิจัยในอนาคต

2. พื้นฐานและงานวิจัยที่เกี่ยวข้อง

2.1 CAPTCHA [3]

CAPTCHA มาจากคำว่า Completely Automated Public Turing Test to tell Computers and Humans Apart ซึ่งเป็นเทคนิคที่ใช้แยกผู้ใช้ที่เป็นมนุษย์ออกจากผู้ใช้ที่เป็นเครื่องคอมพิวเตอร์ โดยใช้วิธีการทดสอบแบบทัวริงย้อนกลับ (Reverse Turing) มีหลักการพื้นฐานคือ การทดสอบต้องเป็นการทดสอบที่มนุษย์สามารถผ่านการทดสอบได้ง่าย แต่ยากสำหรับสิ่งที่ไม่ใช่มนุษย์ ตัวอย่างของการใช้ CAPTCHA เช่นการอ่านข้อมูลตัวอักษรต่างๆที่อยู่ในลักษณะบิดเบี้ยวและมีการรบกวน ดังรูปที่ 1



รูปที่ 1. ตัวอย่างของโปรแกรม reCAPTCHA [5]

ตัวอย่างประโยชน์ที่ได้รับจาก CAPTCHA พอสรุปได้ดังต่อไปนี้ดังนี้ เช่น

- ป้องกันการเขียนความคิดเห็นบน Blog จาก SPAM โดยก่อนที่จะมีการยอมรับความคิดเห็นจะต้องผ่านการทดสอบก่อน
- ป้องกันการลงทะเบียนบนเว็บไซต์ โดยก่อนที่จะมีการลงทะเบียนได้จะต้องผ่านการทดสอบก่อน
- ป้องกันอีเมลแอดเดรส จากการค้นหา โดยผู้ที่เป็น Spammer จะทำการค้นหาอีเมลแอดเดรสที่มีการโพสต์บนเว็บโดยไม่มีการเข้ารหัสไว้ CAPTCHA สามารถป้องกันได้โดยผู้ที่อ่านข้อมูลอีเมลแอดเดรสได้จะต้องผ่านการทดสอบก่อน
- ป้องกันการทำ Online Polls ในการลงคะแนน แต่ละครั้งผู้ที่ผ่านการทดสอบโดย CAPTCHA จึงจะสามารถลงคะแนนได้

ตัวอย่างของโปรแกรมCAPTCHA ที่มีการใช้งานเช่น reCAPTCHA, SQUIGL-PIX, ESP-PIX

2.2 งานวิจัยที่เกี่ยวข้อง

Luis von Ahn และคณะ[3] นำเสนอวิธีการ ในการแบ่งแยกการทำงานระหว่างมนุษย์และเครื่องคอมพิวเตอร์โดยใช้วิธีการนำข้อมูลตัวอักษรมาทำให้เกิดการคิดเพี้ยนและมีการรบกวนเพื่อให้ผู้ใช้งานที่มีมนุษย์ไม่สามารถตีความจากข้อความเหล่านั้นได้ อย่างไรก็ตามก็มีงานวิจัย โดย Mori and Malik [6] ได้แสดงวิธีการแก้ปัญหาของCAPTCHAของโปรแกรม EZYGimpy สำเร็จ 92% และโปรแกรม Gimpy สำเร็จ 33% ต่อมา Kumar Chellapilla และคณะ [7] ได้พบว่า วิธีการที่ใช้ภาพตัวอักษรสามารถแก้ปัญหาได้ง่ายโดยการใช้กระบวนการเรียนรู้โดยเครื่องคอมพิวเตอร์ (Machine Learning) และได้แนะนำให้ CAPTCHA จะมีประสิทธิภาพที่ดีขึ้น ถ้าการเรียนรู้ด้วยเครื่องคอมพิวเตอร์ไม่สามารถแบ่งส่วนของตัวอักษรได้ (Segmentation) ซึ่งอาจทำได้โดยการสร้างการรบกวนโดยใช้เส้น หรือ จุด อย่างไรก็ตาม ตัวอักษรหรือข้อมูลที่นำมาแสดงก็อาจถูกรบกวนทำให้ผู้ใช้ที่เป็นมนุษย์อ่านได้ยากเช่นกัน ต่อมาได้มีวิธีการนำเสนอเทคนิคใหม่ๆของ CAPTCHA ที่ไม่ได้ใช้ภาพของตัวอักษรเช่น CAPTCHA ที่ใช้รูปบนหน้าจอ เป็นการตั้งคำถาม [8] หรือCAPTCHA แบบเสียง (Audio CAPTCHA) [9][10] ได้ถูกนำเสนอ เพื่อที่จะป้องกันสแปม (SPAM) บนระบบโทรศัพท์บนอินเทอร์เน็ต อย่างไรก็ตาม วิธีการ CAPTCHAที่กล่าวมาและมีการใช้งานอยู่ในปัจจุบันเป็นวิธีการที่ภาพตัวอักษรเป็นภาษาอังกฤษเป็นหลัก จากการสำรวจเบื้องต้นพบว่าการสร้างบ็อตเพื่อแก้ปัญหา CAPTCHA นั้นไม่ได้ถูกสร้างจากประเทศไทย ดังนั้นในงานวิจัยนี้แนะนำให้เสนอการสร้าง CAPTCHA เป็นภาษาไทยสำหรับการใช้งานที่มีจุดประสงค์สำหรับผู้ที่ใช้ที่เป็นคนไทย และเป็น การป้องกันการโจมตีจากบ็อตที่มาจากต่างประเทศอย่างได้ผล เนื่องจากผู้จัดทำบ็อตสำหรับการแก้ปัญหา CAPTCHAนั้นยังไม่ปรากฏว่ามีในประเทศไทย ซึ่งสามารถแน่ใจได้ว่าไม่มีความรู้ทางด้านภาษาไทย หรือตัวอักษรไทย

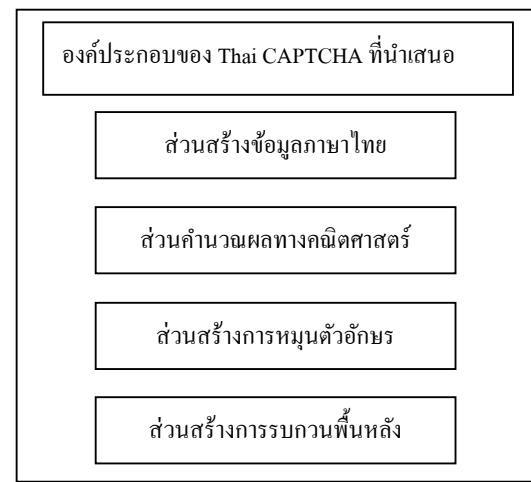
3. วิธีการที่นำเสนอ

เนื่องจากว่าการสร้างตัวอักษรภาษาไทยเพียงอย่างเดียวอาจจะสามารถใช้โปรแกรมโอซีอาร์ (OCR, Optical Character Recognition) ที่มีการเผยแพร่ [11] ผู้วิจัยได้มีการเพิ่มส่วนการ

รบกวนในโครงสร้างการทำงานของ CAPTCHA ที่นำเสนอตามข้อเสนอใน [7] นอกจากนั้นได้เพิ่มส่วนการคำนวณทางคณิตศาสตร์เพื่อให้การคาดเดาที่เกิดขึ้นสามารถทำได้โดยยาก

3.1 โครงสร้าง CAPTCHA ภาษาไทย

ประกอบด้วยการทำงาน 4 ส่วนคือ ส่วนของการสร้างข้อมูลเป็นภาษาไทย และส่วนของการคำนวณทางคณิตศาสตร์ คำตอบดังรูปที่ 2



รูปที่ 2. องค์ประกอบของ Thai CAPTCHA ที่นำเสนอ

องค์ประกอบของการทำงานประกอบไปด้วย 4 ส่วนดังต่อไปนี้

- **ส่วนสร้างข้อมูลภาษาไทย** ทำโดยการสุ่มตัวเลข จาก 1-10 จำนวน 2 ครั้งหลังจากนั้นก็แปลงเป็นตัวอักษรภาษาไทยโดยการแปลงตัวเลขให้เป็นตัวอักษรไทยนั้นจะมีรูปแบบของตัวอักษร 4 แบบที่แตกต่างกัน จะถูกเลือกเพื่อใช้งาน
- **ส่วนคำนวณผลทางคณิตศาสตร์** สามารถที่จะสุ่มวิธีการคำนวณขึ้นมาได้ แต่ในบทความนี้ได้แนะนำให้วิธีการคำนวณแบบบวก (+) การลบ (-) และการคูณ (*) เท่านั้น
- **ส่วนสร้างการหมุนตัวอักษร** ซึ่งเป็นส่วนสำคัญของการสร้าง CAPTCHA ให้มีการเปลี่ยนแปลงทิศทางของตัวอักษร โดยเราจะทำการสุ่ม องศาของการ

หมุนตัวอักษรดังนี้ (0,-15,15,-30,30,-45,45,-60,60,-75,75)

- ส่วนสร้างการรบกวนพื้นหลัง ซึ่งเป็นส่วนสำคัญของการสร้าง CAPTCHA สามารถที่จะสุ่มสร้างจุดและสุ่มสร้างเส้นภาพพื้นหลังขึ้นมาได้ หลังจากนั้นก็จะทำการสุ่มเส้นและจุดขึ้นมาตามที่กำหนด หลังจากนั้นก็จะทำการแปลงข้อมูลทั้งหมดเป็นรูปภาพ

3.2 ขั้นตอนการทำงานของ CAPTCHA ที่นำเสนอ

การทำงานของCAPTCHAที่นำเสนอแสดงดังรูปที่ 3 โดยโปรแกรมจะทำงานดังต่อไปนี้

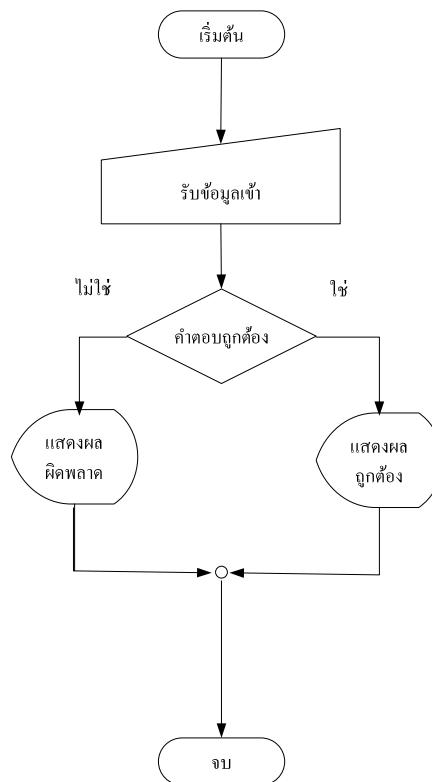
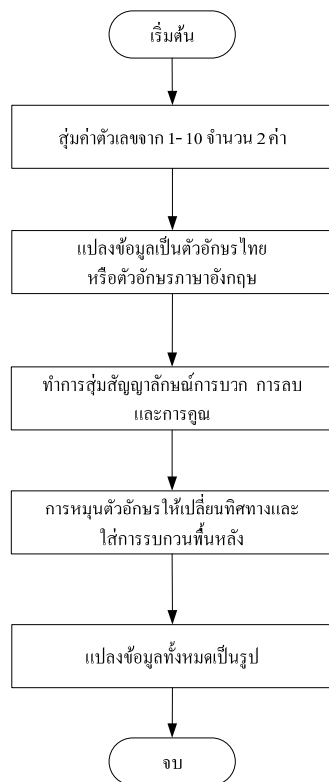
- ขั้นตอน 1. ทำการสุ่มค่าขึ้นมาเพื่อเลือกหมายเลข 1-10
จำนวน 2 ค่าคือตัวเลขที่เป็นตัวตั้งและตัวเลขที่เป็นตัวกระทำ (ตัวบวก,ตัวลบ,ตัวคูณ)
- ขั้นตอน 2. จากนั้นก็จะทำการแปลงข้อมูลตัวเลขให้เป็นภาษาไทยโดยการสุ่มตัวอักษรแต่ละแบบใน

บทความนี้มี 4 แบบ (ดูตัวอย่างตัวอักษรในหัวข้อที่ 4.2)

- ขั้นตอน 3. ทำการสุ่มสัญลักษณ์ทางคณิตศาสตร์ (+, -, *) หลังจากนั้นจะแปลงสัญลักษณ์ให้เป็นตัวหนังสือคือ การ ใส่คำว่า “บวก”, “ลบ”, “คูณ” ระหว่างตัวเลขที่สุ่มขึ้นมาทั้ง 2 ตัวเลข

- ขั้นตอน 4. ทำการหมุนตัวอักษรในทิศทางต่างและทำการใส่การรบกวนโดยการสุ่มเส้นพื้นหลังและจุดพื้นหลังตามที่ได้กำหนด

- ขั้นตอน 5. จากนั้นแปลงค่าตัวเลขเป็นตัวอักษรภาษาไทยตามที่ผู้ใช้เลือกใช้และเปลี่ยนข้อมูลทั้งหมดให้อยู่ในรูปแบบของไฟล์ภาพและเมื่อมีการส่งข้อมูลคำตอบก็จะทำการตรวจสอบ คำตอบที่ได้รับและเปรียบเทียบกับผลลัพธ์ที่เกิดจากการคำนวณ



รูปที่ 3. โฟลว์ชาร์ตการทำงานของ CAPTCHA ที่นำเสนอ

4. การทดลอง

4.1 สภาพแวดล้อมของการทดลอง

การทดสอบทำโดยการติดตั้งโปรแกรมสร้าง CAPTCHA บนเครื่องคอมพิวเตอร์ เพื่อสร้างไฟล์รูปภาพของตัวอักษร สร้างจากนั้นนำไฟล์รูปที่ได้ นำเข้าไปอ่านโดยโปรแกรม OCR ที่ชื่อว่า Amthai [11] ซึ่งเป็นโปรแกรมที่มีการเผยแพร่และอนุญาตให้ใช้งานได้

4.2 ข้อมูล พหุชนะ ตัวอักษรทั้งหมดที่ทำการทดลอง

1. รูปแบบของตัวอักษรในการทดลองใช้ 4 แบบ ดังต่อไปนี้

หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า สิบ

ตัวอักษรแบบที่ 1

หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า สิบ

ตัวอักษรแบบที่ 2

หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า สิบ

ตัวอักษรแบบที่ 3

หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า สิบ

ตัวอักษรแบบที่ 4

2. ข้อมูลตัวอักษรทั้งหมดที่จะทำการทดลอง

- พหุชนะไทย 44 ตัว คือ ก – ฮ
- สระในภาษาไทย 32 ตัว เช่น อะ อา เอะ เอ เป็นต้น
- ตัวเลขที่เป็นตัวหนังสือ คือ หนึ่ง - สิบ
- ตัวเลขไทย และ เลขอารบิก คือ ๐ – ๙, 0 -9

4.3 วิธีการทดลอง

การทดลองในงานวิจัยนี้จะแบ่งเป็น 2 ลักษณะ กล่าวคือ

- ทำโดยการสร้างข้อมูลรูปตัวอักษรแบบต่างๆ จากนั้นทำการเพิ่มค่าการรบกวนโดยใช้จุด หรือ การใส่เส้นที่พื้นหลัง แล้วทำการวัดความแม่นยำของการอ่านข้อมูลของ OCR โดยใช้โปรแกรม Amthai เพื่อที่จะได้ทราบถึงผลกระทบที่เกิดจาก

การรบกวนแบบต่างๆว่ามีผลอย่างไรกับการอ่านค่าของ OCR

- ทำโดยการสร้าง CAPTCHA ที่มีการทำงานที่สมบูรณ์แล้ว ทดลองป้อนค่าเพื่อทดสอบ CAPTCHA

5. ผลการทดลอง

5.1 ไม่มีการรบกวนที่พื้นหลัง

ผลการทดลองเป็นดังต่อไปนี้

- ตัวอักษรแบบที่ 1 สามารถอ่านตัวอักษรภาษาไทย ถูกต้อง 97%
- ตัวอักษรแบบที่ 2 สามารถอ่านตัวอักษรภาษาไทย ถูกต้อง 18%
- ตัวอักษรแบบที่ 3 สามารถอ่านตัวอักษรภาษาไทย ถูกต้อง 5%
- ตัวอักษรแบบที่ 4 สามารถอ่านตัวอักษรภาษาไทย ถูกต้อง 27%

จากผลการทดลองจึงพอสรุปได้ว่า รูปแบบของตัวอักษรมีผลกระทบต่อความถูกต้องของการอ่านจาก OCR และตัวอักษรในแบบที่ 2-4 เป็นแบบที่ควรเลือกในการใช้งานสำหรับ CAPTCHA ต่อไป

5.2 การรบกวนที่พื้นหลังแบบจุดและเส้น

จากตารางที่ 1 ผลการทดลองพอสรุปได้ดังนี้

- ตัวอักษรแบบที่ 1 ที่เป็นพหุชนะ สระ และตัวเลขมีแนวโน้มการอ่านค่าถูกต้องลดลงเมื่อมีการใส่การรบกวนแบบเส้นที่พื้นหลังเพิ่มขึ้น
- ตัวอักษรแบบที่ 2 ที่การรบกวนแบบจุด 5% และแบบเส้น 5% ตัวอักษรทุกลักษณะนั้นมีความถูกต้องในการอ่านค่าน้อยมาก (2%-3%) ยกเว้นตัวเลข (35%) และเมื่อเพิ่มการรบกวนแบบเส้นไปเป็น 15 % ขึ้นไปทำให้การอ่านค่าเกือบจะไม่ถูกต้องเลขทุกแบบของตัวอักษร
- ตัวอักษรแบบที่ 3 และแบบที่ 4 ให้ผลลัพธ์ที่ใกล้เคียงกัน กล่าวคือ การอ่านค่าแทบจะไม่สามารถอ่านค่าได้เลยในทุกๆแบบของการรบกวน

ตารางที่ 1. ผลการอ่านข้อมูลตัวอักษรโดยมีการบวกรวมที่พื้น
หลัง แบบจุด 5 % และแบบเส้นช่วง 5%-20%

การทดลอง	ตัวอักษร แบบที่ 1 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 2 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 3 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 4 อ่าน ถูกต้อง %	เฉลี่ย รวมที่ อ่าน ถูกต้อง %
1. การสุ่มจุด 5 % และสุ่มเส้น 5% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	70	2	1	1	18.50
ตัวสระ	25	3	1	1	7.38
ตัวเลขหนังสือ	20	3	0	0	5.75
ตัวเลขไทย+อาราบิก	80	35	4	3	30.44
2. การสุ่มจุด 5 % และเส้น 10% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	60	2	2	0	16.07
ตัวสระ	22	2	1	0	6.19
ตัวเลขหนังสือ	20	0	0	0	5.00
ตัวเลขไทย+อาราบิก	15	17	6	4	10.46
3. การสุ่มจุด 5 % และเส้น 15% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	54	1	1	0	14.09
ตัวสระ	21	0	0	0	5.25
ตัวเลขหนังสือ	20	0	0	0	5.00
ตัวเลขไทย+อาราบิก	80	5	3	1	22.32
4. การสุ่มจุด 5 % และเส้น 20% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	52	2	0	0	13.46
ตัวสระ	25	0	0	0	6.25
ตัวเลขหนังสือ	10	0	0	0	2.50
ตัวเลขไทย+อาราบิก	65	1	0	0	16.51

ตารางที่ 2. ผลการอ่านข้อมูลตัวอักษรโดยมีการบวกรวมที่พื้น
หลัง แบบจุด 10 % และแบบเส้นช่วง 5%-20%

การทดลอง	ตัวอักษร แบบที่ 1 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 2 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 3 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 4 อ่าน ถูกต้อง %	เฉลี่ย รวมที่ อ่าน ถูกต้อง %
1. การสุ่มจุด 10 % และสุ่มเส้น 5% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	40	2	1	1	11.00
ตัวสระ	20	3	1	0	5.88
ตัวเลขหนังสือ	20	3	0	0	5.75
ตัวเลขไทย+อาราบิก	50	35	4	3	22.94
2. การสุ่มจุด 10 % และสุ่มเส้น 10% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	27	2	2	0	7.82
ตัวสระ	20	2	1	0	5.72
ตัวเลขหนังสือ	10	0	0	0	2.50
ตัวเลขไทย+อาราบิก	75	17	5	3	24.96
3. การสุ่มจุด 10 % และสุ่มเส้น 15% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	56	1	1	0	14.59
ตัวสระ	21	0	0	0	5.25
ตัวเลขหนังสือ	0	0	0	0	0.00
ตัวเลขไทย+อาราบิก	95	5	3	2	26.32
4. การสุ่มจุด 10 % และสุ่มเส้น 20% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	34	2	0	0	8.96
ตัวสระ	20	0	0	0	5.00
ตัวเลขหนังสือ	0	0	0	0	0.00
ตัวเลขไทย+อาราบิก	90	1	1	0	23.01

จากตารางที่ 2 -4 ผลการทดลองพอสรุปได้ดังนี้

- ตัวอักษรแบบที่ 1 ที่เป็นพยัญชนะ สระ และตัวเลขมีแนวโน้มการอ่านค่าถูกต้องลดลงเมื่อมีการใส่การบวกรวมแบบเส้นที่พื้นหลังเพิ่มขึ้น และเมื่อเทียบกับการบวกรวมแบบจุด 5% ในตารางที่ 1 เห็นได้ว่าความถูกต้องของการอ่านค่าลดลง
- ตัวอักษรแบบที่ 2 ให้ผลลักษณะเดียวกับตารางที่ 1 กล่าวคือเกือบจะไม่สามารถอ่านค่าตัวอักษรได้ถูกต้องเลย ยกเว้นตัวเลข และเมื่อเพิ่มค่าการบวกรวมแบบเส้นเป็น 20% ความถูกต้องเกือบเท่ากับศูนย์ (1%-2%)
- ตัวอักษรแบบที่ 3 และแบบที่ 4 ให้ผลลัพธ์ที่ใกล้เคียงกัน กล่าวคือ การอ่านค่าแทบจะไม่สามารถอ่านค่าได้เลยในทุกๆแบบของการบวกรวมและสอดคล้องกับตารางที่ 1

ตารางที่ 3. ผลการอ่านข้อมูลตัวอักษรโดยมีการบวกรวมที่พื้น
หลัง แบบจุด 15 % และแบบเส้นช่วง 5%-20%

การทดลอง	ตัวอักษร แบบที่ 1 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 2 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 3 อ่าน ถูกต้อง %	ตัวอักษร แบบที่ 4 อ่าน ถูกต้อง %	เฉลี่ย รวมที่ อ่าน ถูกต้อง %
1. การสุ่มจุด 15 % และสุ่มเส้น 5 % พื้นหลังของพื้นที่					
ตัวพยัญชนะ	54	2	1	0	14.25
ตัวสระ	25	3	1	0	7.13
ตัวเลขหนังสือ	0	3	0	0	0.75
ตัวเลขไทย+อาราบิก	75	35	4	0	28.44
2. การสุ่มจุด 15 % และเส้น 10% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	50	2	2	0	13.57
ตัวสระ	21	2	1	0	5.97
ตัวเลขหนังสือ	20	0	0	0	5.00
ตัวเลขไทย+อาราบิก	70	17	0	0	21.71
3. การสุ่มจุด 15 % และเส้น 15% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	55	1	1	0	14.34
ตัวสระ	20	0	0	0	5.00
ตัวเลขหนังสือ	20	0	0	0	5.00
ตัวเลขไทย+อาราบิก	90	5	3	0	24.57
4. การสุ่มจุด 15 % และเส้น 20% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	30	2	0	0	7.96
ตัวสระ	10	0	0	0	2.50
ตัวเลขหนังสือ	0	0	0	0	0.00
ตัวเลขไทย+อาราบิก	50	1	0	0	12.76

ตารางที่ 4. ผลการอ่านข้อมูลตัวอักษรโดยมีการรบกวนที่พื้นหลัง แบบจุด 20 % และแบบเส้นช่วง 5%-20%

การทดลอง	ตัวอักษรแบบที่ 1 อ่าน ถูกต้อง	ตัวอักษรแบบที่ 2 อ่าน ถูกต้อง	ตัวอักษรแบบที่ 3 อ่าน ถูกต้อง	ตัวอักษรแบบที่ 4 อ่าน ถูกต้อง	เฉลี่ย รวมที่ อ่าน ถูกต้อง %
1. การสุ่มจุด 20 % และเส้น 5% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	55	2	1	0	14.50
ตัวสระ	25	1	1	0	6.75
ตัวเลขหนังสือ	20	1	0	0	5.25
ตัวเลขไทย+อาราบิก	90	5	1	0	24.00
2. การสุ่มจุด 20 % และเส้น 10% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	43	1	0	0	11.00
ตัวสระ	12	2	0	0	3.47
ตัวเลขหนังสือ	20	0	0	0	5.00
ตัวเลขไทย+อาราบิก	70	0	0	0	17.50
3. การสุ่มจุด 20 % และเส้น 15% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	50	0	0	0	12.50
ตัวสระ	25	0	0	0	6.25
ตัวเลขหนังสือ	10	0	0	0	2.50
ตัวเลขไทย+อาราบิก	65	0	0	0	16.25
4. การสุ่มจุด 20 % และเส้น 20% พื้นหลังของพื้นที่					
ตัวพยัญชนะ	32	0	0	0	8.00
ตัวสระ	10	0	0	0	2.50
ตัวเลขหนังสือ	0	0	0	0	0.00
ตัวเลขไทย+อาราบิก	30	0	0	0	7.50

ดังนั้นจากผลการทดลองตามตารางที่ 1-4 พอสรุปเป็นแนวทางคุณสมบัติของการพัฒนา Thai CAPTCHA ดังต่อไปนี้

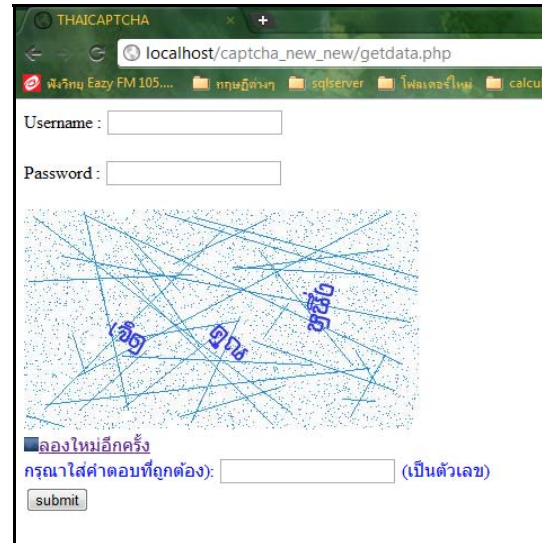
- การเลือกแบบตัวอักษรควรเลือกแบบที่เป็นแบบ 2-4 ซึ่งเป็นแบบตัวอักษรที่มีลักษณะพิเศษ กว่าแบบที่ 1 ซึ่งเป็นแบบมาตรฐานทั่วไป
- การใส่จุดพื้นหลังประมาณ 20 % แม้ว่าใส่การรบกวนจากเส้นน้อยก็ทำให้ไม่สามารถอ่านตัวอักษรแบบที่ 2 ได้ ส่วนตัวอักษรแบบที่ 3 และ 4 ใส่จุดพื้นหลังแค่ 5% และใส่การรบกวนแบบเส้น 20% ก็ไม่สามารถอ่านข้อมูลได้ถูกต้องเลย อย่างไรก็ตามการใส่การรบกวนมากเกินไปก็อาจทำให้ผู้ใช้อ่านค่าได้ยาก ดังนั้นต้องมีการพิจารณาการใส่ข้อมูลรบกวนไม่ให้ทำให้ผู้อ่านที่เป็นมนุษย์สับสน

5.3 การทดลองทำงาน Thai CAPTCHA

จากแนวทางการกำหนดคุณสมบัติในหัวข้อที่ 5.2 ได้ทำการทดลองสร้าง Thai CAPCHA โดยเลือกใช้คุณสมบัติการสร้างตัวอักษรดังต่อไปนี้

- ใช้ตัวอักษรแบบที่ 3
- ใช้การรบกวนแบบจุด 10%

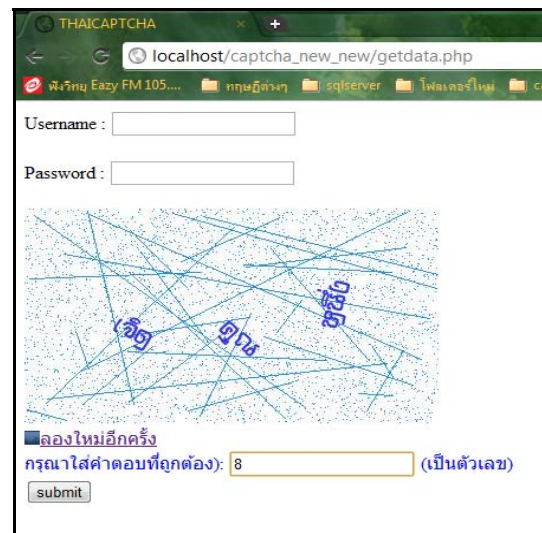
- รบกวนแบบเส้น 15%
- สร้างฟังก์ชันการคำนวณทางคณิตศาสตร์ ด้วยวิธีการที่นำเสนอในหัวข้อที่ 2 ผลการทำงานเป็นดังรูปที่ 4



รูปที่ 4. หน้าจอแสดงผล Thai CAPTCHA ที่นำเสนอ

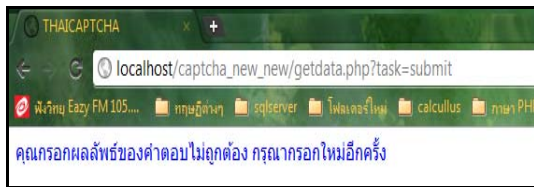
5.3.1 การทดสอบผลการทำงานโดยการป้อนผลลัพธ์ที่ผิดพลาด

จากรูปที่ 5 ทำการป้อนค่าข้อมูลผลลัพธ์ที่ผิด ซึ่งจาก Thai CAPTCHA ผลลัพธ์ที่ถูกต้องคือค่า 7 แต่ทำการป้อนค่าเป็น 8



รูปที่ 5. หน้าจอการป้อนข้อมูลที่ผิดพลาด

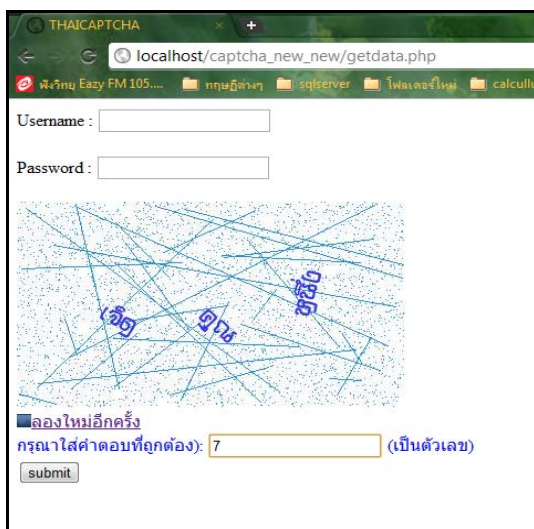
ผลลัพธ์ที่ได้ในกรณีที่การป้อนคำตอบผิดพลาดจะแสดง
ดังรูปที่ 6



รูปที่ 6. หน้าจอผลลัพธ์การป้อนข้อมูลผิดพลาด

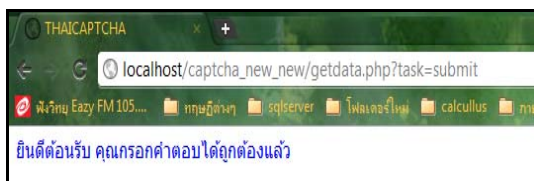
5.3.2 การทดสอบผลการทำงานโดยการป้อนผลลัพธ์ที่ถูกต้อง

ทำการป้อนค่าข้อมูลโดยป้อนผลลัพธ์ที่ถูกต้อง จากตัวอย่างการทำงานดังรูปที่ 7 ผลลัพธ์ที่ถูกต้องคือค่า 7



รูปที่ 7. หน้าจอการป้อนข้อมูลที่ต้องการ

ผลลัพธ์ที่ได้ในกรณีที่การป้อนคำตอบที่ถูกต้องจะแสดง
ดังรูปที่ 8



รูปที่ 8. หน้าจอผลลัพธ์การป้อนข้อมูลที่ต้องการ

6. บทสรุป

จากการทดลองโปรแกรม Thai CAPTCHA ที่มีการรบกวนในแบบจุดประมาณ 10% และแบบเส้นประมาณ 15% ก็เพียงพอที่จะทำให้โปรแกรม OCR ที่มีอยู่ในปัจจุบัน คือโปรแกรม Amthai ไม่สามารถที่จะอ่านค่าข้อมูลได้ถูกต้อง และก็ไม่รบกวนการตีความหมายของมนุษย์

ผลลัพธ์ของการใช้คณิตศาสตร์อย่างง่ายช่วยให้ตัวอักษรที่มีความหมายซึ่งผู้ใช้ที่ไม่ใช่มนุษย์สามารถคาดเดาคำตอบได้ยากขึ้น นอกจากนั้นยังเพิ่มการสุ่มเลือกการกระทำทางคณิตศาสตร์จาก 3 แบบ คือ บวก ลบ และคูณ ทำให้ลักษณะคำตอบเป็นรูปแบบที่ไม่แน่นอนทำให้ประสิทธิภาพของ Thai CAPTCHA ดีขึ้น

อย่างไรก็ตามการทดสอบจากโปรแกรม OCR นั้นใช้โปรแกรม Amthai เป็นหลักเนื่องจากเป็นโปรแกรมที่ใช้แพร่หลายในประเทศไทย ดังนั้นแนวทางในอนาคตจะเพิ่มการทดสอบโดยโปรแกรม OCR อื่นๆ และ พัฒนาตัวอักษรให้มีลักษณะบิดเบี้ยว หรือนำวิธีการทางด้านเสียงภาษาไทย หรือการใช้ความรู้ท้องถิ่นด้านอื่นๆ มาเพื่อเพิ่มรูปแบบในการป้องกันอีกทางหนึ่ง

7. เอกสารอ้างอิง

- [1] James C. Foste "Automating Google Search" Hacking for Penetration Testers, 2005, Pages 363-415
- [2] <http://members.cox.net/randar/botfaq.html>
- [3] Von Ahn L, Blum M, and Langford J, The CAPTCHA Project. <http://www.captcha.net>
- [4] Carlos Javier Hernandez-Castro, Arturo Ribagorda, "Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study", computers & security journal 2010, Pages 141 – 157
- [5] <http://www.google.com/recaptcha>
- [6] Mori G, Malik J (2003), "Recognizing Objects in Adversarial Clutter: Breaking a VisualCAPTCHA," *Proc. of the Computer Vision and Pattern Recognition (CVPR) Conference*, IEEE Computer Society, vol.1, pages:I-134 - I-141, June 18-20, 2003
- [7] Kumar Chellapilla and Patrice Y. Simard. [Using Machine Learning to Break Human Interaction Proofs \(HIPs\)](#). In *NIPS*.

- [8] M. Hassan Shirali-Shahreza, Mohammad Shirali-Shahreza, "Localized CAPTCHA for Illiterate People", International Conference on Intelligent and Advanced Systems 2007, Pages 675-679
- [9] MSN Audio CAPTCHA, <https://signup.live.com/>
- [10] Recaptcha Audio CAPTCHA, <http://recaptcha.net/learnmore.html>
- [11] http://www.nectec.or.th/img/index.php?option=com_content&task=view&id=25&Itemid=39&lang=en

Copyright © 2012 by the Journal of Information Science and Technology.