On Designing Modern CAPTCHAs for Security and Usability

<<FULL NAME>>
<< STUDENT ID>>

British Columbia Institute of Technology

COMP 7036

<<DATE>>

On Designing Modern CAPTCHAs for Security and Usability

Preventing automated bots from abusing website features has long been a point of concern for web developers, and CAPTCHAs have long been accepted as the most reasonable means of addressing this risk. CAPTCHAs, or "completely automated public Turing tests for telling computers and humans apart", are simple challenge-response exercises which ask the user to perform some task that, while easy for a human to accomplish, is difficult for an automated bot. By completing the challenge, the user proves their humanity, hence the labelling of the exercise as a Turing test.

Designing these CAPTCHAs, however, is becoming more and more difficult by the day, due to the increasing capabilities of computer—and thus bots—to solve a variety of tasks. My research question, "What is a CAPTCHA design that is unsolvable by modern bots and viable for real-life use in websites and other applications?", essentially seeks to design such a test. To answer this question, I looked at the criteria for good CAPTCHAs and the various ways these criteria might be met.

A key criterion for an effective CAPTCHA is that it be easily solvable by humans, even if they are encountering a test of its type for the first time (Bulumulla, 2014). "Easily" is often broken down into three sub-evaluations: the accuracy of human responders in solving the CAPTCHA, which should be maximised; the time taken by human responders to solve the CAPTCHA, which should be minimized; and the usability of the CAPTCHA as perceived by human responders. Behshti, Das, Gao, and Xu all use such a breakdown in evaluating their CAPTCHA tests (Behshti, 2015; Das, 2016; Gao, 2017; Xu, 2014). The exact requirement for accuracy is not formally agreed upon, but Xu states that "the human success rate should be at least 90 percent" (Xu, 2014). The requirement for time taken is similarly vague, but Osadchy notes in her usability testing that responders that average above 40 seconds per test should be considered outliers, as most responders will take far less time than this for a typical CAPTCHA (Osadchy, 2017). Perceived usability can be determined in myriad ways, but a survey is often used. The standard System Usability Scale (SUS) used by Wickramasingha and Gao is often used, since it eliminates any doubt that the CAPTCHA designer may have skewed the survey questions to result in better responses (Gao, 2016; Wickramasingha, 2015). It should be noted that by the SUS, a score of 68 is considered "average usability" and can be used a benchmark (Wickramasingha, 2015). As Ahmad notes, CAPTCHA designers should also keep in mind special considerations for users with disabilities such as colour-blindness, which may impact their ability to solve certain types of CAPTCHAs (Ahmad, 2012) Based upon these measurement standards, I would argue that a CAPTCHA system which results in 90% or higher user success rate, takes less than 30 seconds to solve, and scores above 75 on the SUS should be considered to have good usability.

The second major criterion for CAPTCHAs is security, or the degree to which the CAPTCHA is difficult for a bot to solve. This is typically expressed as a percentage indicating how often a bot will give the right response. The exact target for this percentage is also not agreed upon, but can be anywhere from 0.01 percent (Xu, 2014) to 1 percent (Gao, 2013). As it is sometimes thought that the 0.01 percent goal is "too ambitious" (Gao, 2013), I will use the 1 percent goal for my research project.

Past researchers who have developed ways to break commonly used CAPTCHAs have suggested certain techniques to CAPTCHA developers for creating more robust CAPTCHA designs, often building on each other's research, and I will consider these suggestions in my research. Haichang Gao found that while basic text-based CAPTCHAs are quite easy for bots to solve, adding motion—such as in NuCAPTCHA—steeply increased the difficulty for bots (Gao, 2013). Xu built on Gao's research by developing an attack that could break NuCAPTCHA with over 70% accuracy (Xu, 2014). Xu proposed emergent-image

CAPTCHAs as a viable option to replace basic text tests (Xu, 2014). In response, Song Gao (not the same as the previously mentioned Gao) developed an attack that could break Xu's CAPTCHA by up to 89.2% (Gao, 2017). Gao criticizes the fact that Xu's CAPTCHA design uses 2D exclusively, which allowed him to rely on consistent camera projection of the text in beating the test (Gao, 2017). Based on this, Gao developed an emergent-image CAPTCHA using pseudo-3D projection to eliminate the weakness of 2D, although he concedes that his test has notably lower usability than Xu's (Gao, 2017). From the results of these studies, one can conclude that giving the user solution information over time rather than all at once steeply increases difficulty and solving time for bots while maintaining usability for humans.

Although my research will focus on CAPTCHAs for desktop users, it is worth mentioning that other platforms, such as mobile, offer their own unique possibilities for new CAPTCHA designs. Bulumulla's LineCAPTCHA design, for example, makes use of mobile device touch screens in its line-drawing–based CAPTCHA test (Bulumulla, 2014). Similarly, Wickramasingha's RotateCAPTCHA capitalizes on the accelerometers commonly present in mobile devices to present a test that requires users to rotate their device to match the alignment of image segments (Wickramasingha, 2015). These considerations are becoming even more important as desktop and laptop devices begin to adopt more of the technologies which have often been thought of as mobile-specific, such as touch screens.

Lastly, I will make a note on human-assisted CAPTCHA attacks. In this type of attack, a bot program will attempt to abuse a website or other application that makes use of CAPTCHAs. Whenever a CAPTCHA is encountered, control is handed off to a human assistant, who will solve the CAPTCHA and then return control to the bot for the continuation of the attack. As Xu notes, this approach is sometimes considered by attackers when the cost—either temporal of financial—of having a bot break the CAPTCHA is too great (Xu, 2014). Human-assisted attacks are highly problematic because they would appear to be impossible to defend against by definition; CAPTCHAs, after all, are deliberately designed to be human-solvable. Thus, for the purposes of this research, I will consider human-assisted attacks to be part of a separate problem domain and not concern myself with this threat. Instead, I will focus on developing a CAPTCHA design that is robust against bots operating without the intervention of humans.

I have discussed the past work of various researchers in determining what makes CAPTCHAs usable and secure, as well as how to break many CAPTCHA schemes typically thought of as secure. Some of these researchers have suggested new models for CAPTCHA tests to improve security. However, all of the models suggested have been either broken, or have not been widely implemented due to usability concerns. Thus, my research will aim to develop a CAPTCHA test design that meets the previously described goals in usability and security.

References

Ahmad, A., Yan, J., & Ng, W.-Y. (2012). CAPTCHA Design: Color, Usability, and Security. *IEEE Internet Computing*, 44–50.

Behshti, S. M. R. S., & Liatsis, P. (2015). CAPTCHA Usability and Performance, How to Measure the Usability Level of Human Interactive Applications Quantitatively and Qualitatively? *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, 131–136.

Bulumulla, C. B., & Ragel, R. G. (2014). LineCAPTCHA mobile: A user friendly replacement for unfriendly reverse turing tests for mobile devices. *7th International Conference on Information and Automation for Sustainability*, 1–6.

Das, M., Naresh, A., Narang, A., Narayana, A., & Jayashree, R. (2016). Automated CAPTCHA Generation from Annotated Images Using Encoder Decoder Architecture. *2016 International Conference on Information Technology (ICIT)*, 45–50.

Gao, H., Wang, W., Qi, J., Wang, X., & Liu, X. (2013). The Robustness of Hollow CAPTCHAs. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.*

Gao, H., Wang, X., Cao, F., Zhang, Z., Lei, L., Qi, J., & Liu, X. (2016). Robustness of text-based completely automated public turing test to tell computers and humans apart. *IET Information Security, 10*(1), 45–52.

Gao, S., Mohamed, M., Saxena, N., & Zhang, C. (2017). Emerging-Image Motion CAPTCHAs: Vulnerabilities of Existing Designs, and Countermeasures. *IEEE Transactions on Dependable and Secure Computing*.

Osadchy, M., Hernandez-Castro, J., Gibson, S., Dunkelman, O., & Pérez-Cabo, D. (2017). No Bot Expects the DeepCAPTCHA! Introducing Immutable Adversarial Examples. *IEEE Transactions on Information Forensics and Security, 12*(11), 2640–2653.

Wickramasingha, N. R. W. W. M. R. D., Keerawella, H. B. R. A. K. R. A. M., Samarasinghe, S. A. N., & Ragel, R. G. (2015). RotateCAPTCHA: A novel interactive CAPTCHA design targeting mobile devices. *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, 49–54.

Xu, Y., Reynaga, G., Chiasson, S., Frahm, J.-M., Monrose, F., & van Oorschot, P. C. (2014). Security Analysis and Related Usability of Motion-Based CAPTCHAs: Decoding Codewords in Motion. *IEEE Transactions on Dependable and Secure Computing, 11*(5), 480–493.