

Designing Mobile Friendly CAPTCHAs: An Exploratory Study

Nan Jiang
Bournemouth University
Fern Barrow, Poole, BH12 5BB
njiang@bournemouth.ac.uk

Huseyin Dogan
Bournemouth University
Fern Barrow, BH12 5BB
hdogan@bournemouth.ac.uk

Feng Tian
Bournemouth University
Fern Barrow, BH12 5BB
ftian@bournemouth.ac.uk

CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart) are one of the most widely used authentication mechanisms that help to prevent online service abuse. With the advent of mobile computing, mobile devices such as smartphones and tablets have become the primary way people access the Internet. As a result, increasing attention has been paid to designing CAPTCHAs that are mobile friendly. Although such CAPTCHAs generally show their advantages over traditional ones, it is still unclear what the best practices are for designing a CAPTCHA scheme that is easy to use on mobile devices. In this paper, we present an exploratory study that focuses on developing a more holistic view of usability issues with interactive CAPTCHAs to inform design guidance. This is done through investigating the usability performance of seven mobile friendly CAPTCHA schemes representing five different CAPTCHA types.

CAPTCHAs. Smart Phones. Mobile Devices. Usability. Design Guidelines.

1. INTRODUCTION

CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart) are a kind of authentication mechanisms that utilises hard Artificial Intelligence (AI) problems to create tests to distinguish humans and computers (Von Ahn et al. 2004). CAPTCHAs have been widely used on webpages to protect online services from automatic attacks and abuses since the term was coined in 2003. The tests in a CAPTCHA scheme are often referred to as challenges or Human Interactive Proofs (HIP) where a popular type of test is based on character recognition (Chellapilla et al. 2005). For example, Google ReCAPTCHA (version 1) asks users to recognise two separated and distorted words and/or number plates to prove they are real humans (Figure 1).



Figure 1: Google ReCAPTCHA version 1 example.

With the advent of mobile computing, mobile devices such as smartphones and tablets have overtaken laptops and desktop computers as the most used devices to access the Internet and online services (Ofcom 2015; comScore 2015; Pew Research Center 2015). Unlike desktop and laptop

computers which often feature large displays with peripheral input devices (e.g., mouse/trackpad, keyboard), smartphones and tablets are usually equipped with small, touch-enabled displays. When CAPTCHAs are used on such a device, the challenge lies in not only the display size but also the shift of the main interaction methods from point and click + type to touch gestures (Wismer et al. 2012).

Various approaches have been taken to address this challenge. One approach is to use interaction techniques that are more 'compatible' with touch screens (e.g., using mouse point + click instead of keyboard input) (Chow et al. 2008; Desai & Patadia 2009; Chaudhari et al. 2011; Lin et al. 2011; Ye et al. 2013; Conti et al. 2016). The other approach is to purely rely on interaction methods that are native to mobile devices (Okada & Matsuyama 2012; Saxena et al. 2012; Jiang & Tian 2013; Jiang & Dogan 2015; Leiva & Alvaro 2015; Tsuruta et al. 2013). However, unlike character recognition based CAPTCHAs where there exist some general design guidelines for improving their usability (Yan et al. 2008; Bursztein et al. 2014), guidelines for designing mobile-friendly CAPTCHA schemes still remain underexplored.

In this paper, we present an exploratory study that aims to develop a more holistic view of usability issues in mobile friendly CAPTCHA design to inform design guidance. In the study, the usability performance of seven mobile friendly CAPTCHAs representing five different types of mobile friendly

CAPTCHA schemes is examined. The paper is organised as follows. The methodology is explained in Section 2 followed by the results and discussion in Section 3. The design implications are introduced in Section 4 and the conclusion is drawn in Section 5.

2. METHOD

2.1 CAPTCHA selection and benchmarking

It is clear that the selection needs to focus on CAPTCHA schemes that are mobile friendly. In other words, it should look into whether a CAPTCHA can utilise input interaction methods that are native to mobile devices (i.e., touch gestures) rather than only being 'supported' on mobile devices (i.e., soft/virtual keyboards). Therefore, CAPTCHAs that require user to recognise characters in an image, audio clip and video clip, were excluded. Moreover, as selected CAPTCHAs need to be installed and configured in a testing environment for the experiment, their matureness and availability are also key considerations for the inclusion. As a result, Table 1 shows the list of CAPTCHAs selected for this study where their success criteria, interaction method and challenge type are also stated. Note except for TapCHA v2, all CAPTCHAs in the table are commercial solutions but they also represent some of the most common design approaches as seen in the research. For example, FunCaptcha uses a face recognition scheme (D'Souza et al. 2012; Kim et al. 2014;

Kalsoom et al. 2012) and an image orientation scheme (Gossweiler et al. 2009; Kim et al. 2010; Banday & Shah 2015). KeyCAPTCHA uses a puzzle scheme which is similar to (Gao et al. 2010). It should also be noted that Google ReCAPTCHA version 1, a character-recognition based CAPTCHA has been chosen as the benchmarking CAPTCHA for two main reasons. First, it is the most widely used 'traditional' type CAPTCHA so any mobile friendly CAPTCHA should achieve at least better usability performance when being compared with it on a mobile device. Second, in comparison with Google ReCAPTCHA version 1 which is the most widely used character-recognition based CAPTCHA solution, there lacks a universally recognised mobile CAPTCHA scheme.

2.2 Apparatus

CAPTCHAs are widely used to protect online services where a typical application area is website signup and log in. A dummy user login form was therefore created using Bootstrap, a responsive front-end framework to ensure the form would always be displayed properly on a mobile device. Note the user information was hardcoded in this form so participants were only required to complete the CAPTCHA challenge in order to submit the form. All 7 selected CAPTCHAs were integrated into this form individually so there were 7 forms prepared. An Apple iPhone 6, which features a 4.7-inch 1334x750 display, was used as the testing device to enable participants to access these online login forms.

Table 1: An overview of selected CAPTCHA schemes.

Name	Success criteria	Interaction method	Type
Confident CAPTCHA	Click the correct image for three times.	Tap (click)	Object recognition
FunCaptcha	1. Rotate an object making it the right way up. 2. Move the woman to the middle and select any object in the second screen.	1. Tap (click) 2. Drag and drop	3. Object orientation 4. Face recognition + gesture
KeyCAPTCHA	1. Assemble the image using puzzles provided. 2. Drag the images to the relevant categories	Drag and drop	Object recognition
ReCAPTCHA v2	Select all squares with specific object(s).	Tap (click)	Object recognition
sweetCaptcha	Move a specific object to touch another.	Drag and drop	Object recognition
TapCHA v2	Move a specific object to touch another.	Drag and drop	Object recognition + character recognition
visualCAPTCHA	Click the correct icon as stated.	Tap (click)	Object recognition

2.3 Participants

20 participants, aged between 18 and 55, were volunteered for the study. They were undergraduate and postgraduate students studying different subjects and professionals working in different industrial sectors. They were all active web and mobile users, owning at least one smartphone/tablet

and desktop/laptop computers. They were all familiar with CAPTCHAs.

2.4 Design

A within-subjects design was chosen for this study so that each participant had to complete all CAPTCHA tests on the mobile device. Moreover, for counter balancing, each participant was asked

to complete Google ReCAPTCHA v1 then a tap type CAPTCHA followed by a drag & drop type CAPTCHA as shown in Table 1 in an interleaving mode. The order remained the same but the actual CAPTCHA selected for each type was randomised using a script.

2.5 Procedure

Each participant was briefed on the purpose of the study and their consents were obtained before the actual experiment. They were then presented with all CAPTCHAs on the testing device and they were asked to try each of them several times to familiarise themselves with these schemes and the mobile device. After that, each participant was asked to complete 6 challenges for each CAPTCHA so a total of 120 tests per CAPTCHA were logged (20 x 6). Their performance was video recorded by an observer for further analysis. After completing all tests, they were asked to fill in a questionnaire and provide oral feedback about their experience. Before the smartphone was handed to the next participant, its screen was cleaned to remove finger smudges and the tests were reset.

2.6 Measurement

Common usability metrics were used in the measurement including: *completion time* (efficiency), *completion rate* (effectiveness) and *errors* (effectiveness). In addition to that, System Usability Scale (SUS) was used as the questionnaire method to obtain participants' subject assessment on the CAPTCHA schemes they have tried.

3. RESULTS AND DISCUSSION

3.1 Completion time

The completion time was defined as the average time participants spent on completing a CAPTCHA test. It was recorded in seconds starting from the time as soon as the CAPTCHA was fully displayed in the login form and stopping at the time when the submit button in the form was tapped. The results are shown in Figure 2 (min = 4.95; max = 8.8; SD = 1.53; Mean = 6.71) where the horizontal line across all columns is the benchmark (Google ReCAPTCHA v1: 8.55 seconds). All CAPTCHAs, except for KeyCAPTCHA, were completed quicker than Google ReCAPTCHA v1 where four CAPTCHAs reported less than 6 seconds average completion time ($p < 0.05$, one-way ANOVA).

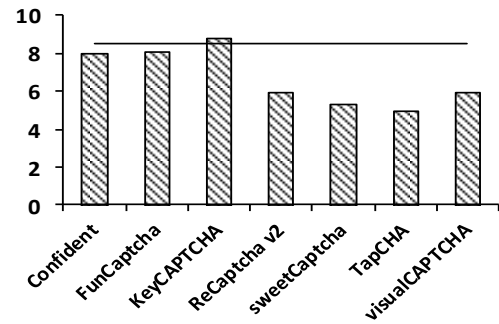


Figure 2: Average completion time of CAPTCHAs.

It was noticed that there were two obvious lags when participants were completing a KeyCAPTCHA test. The first lag was related to the position of all puzzle pieces as many participants tended to separate them from the stack first before deciding where to place them. This happened more often when it was not easy to distinguish puzzle pieces with similar patterns and colours (Figure 3). Second, many participants tried to place the puzzle to the exact location as if they were completing a real jigsaw puzzle. This has caused some lags as this kind of control requires precision whilst a participant's finger is usually larger than a puzzle on the screen (i.e., "fat finger problem" (Vogel et al. 2007; Roudaut et al. 2008)).



Figure 3: KeyCaptcha puzzle test example.

3.2 Completion rate

The completion rate was defined as the percentage of CAPTCHAs that were completed successfully by participants. Note as a CAPTCHA challenge is considered as solved only when all success criteria in the test were met, there was no partial success considered in the measurement. The results are shown in Figure 4 (min: 81.6%, max: 97.5%, SD: 6.32%, Mean: 91.86%). Only FunCaptcha (81.6%) and Google ReCAPTCHA v2 (84.1%) reported slightly lower success rate than the benchmarking Google ReCAPTCHA v1 (88.3%). There was a significant difference between CAPTCHAs (one-way ANOVA, $p < 0.05$).

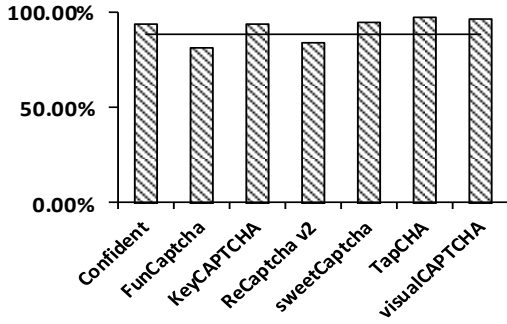


Figure 4: Average completion rate of CAPTCHAs.

It was noticed that failures recorded with FunCaptcha were relevant to both types of challenges it used. For the “Roll the ball” challenge, the main issue was to do with the progress indication when a sub test was completed as a tick would always appear no matter whether the test was completely successfully or not. For the “Move the woman in the middle” challenge, as the other 7 images were created using one contrasting face in different angles and distances, it sometimes confused participants. For Google ReCAPTCHA v2, most failures were related to one particular type of tests where participants were asked to select all images containing a specific object (e.g., store front). As the images used in these tests were very small, when different level of detail was presented, sometimes participants could easily miss some relevant images. Moreover, as there was no undo mechanism after selecting an image, any wrong selection by accident would lead to the whole test failure.

3.3 Errors

The errors were defined as the total number of excessive, missing and/or wrong user actions in comparison to the minimal user actions required for completing a test successfully. Errors were captured in quantity and grouped in four categories as shown in Table 2. Note the number of errors was calculated as the difference between the total number of actual user actions and the total number of correct minimal actions.

Table 2: Error categories with definitions.

Category	Definition
Accidental (A)	Errors caused by accidental user actions (e.g., a user chose a wrong object by accident).
Unnecessary (U)	Errors caused by unnecessary user actions (e.g., a user clicked an object twice to confirm the selection).
Defective (D)	Errors caused by system defects (e.g., an object was not responsive to touch operation or CAPTCHA was not displayed properly in the screen so

	the user had to zoom in and move the screen).
Misinterpreted or misjudged (M)	Errors caused by incorrect user actions due to user’s misunderstanding of instructions or misjudgement of success criteria (e.g., a user clicked on an object which they thought it was relevant).

The detailed error breakdowns for each CAPTCHA are shown in Figure 5. In general, sweetCaptcha, TapCHA and visualCAPTCHA reported the fewest number of errors across all categories where TapCHA also reported no errors in Category U and M ($p < 0.05$).

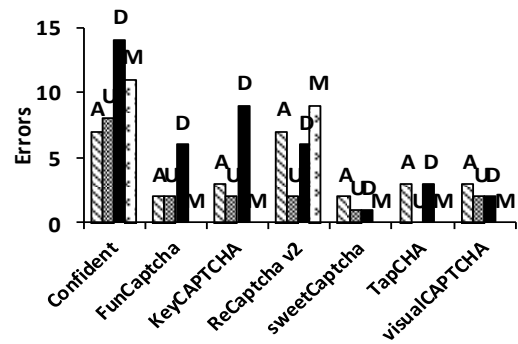


Figure 5: Error number and breakdowns for each CAPTCHA.

Errors in Category M were only found with Confident CAPTCHA and Google ReCAPTCHA v2 indicating the rest of CAPTCHAs had reasonably clear instruction and easily distinguishable objects. As the two CAPTCHAs were based on checking image relevancy, it was noted that sometimes participants over-interpreted what was required for the challenge. For example, when being asked to select an image where people are present in a Confident CAPTCHA test, some participants chose not to select the only relevant image where some LEGO men were present which led to a test failure. This also happened with Google ReCAPTCHA v2 where participants selected not only the Parking sign facing forward but also the sign showing the back on the image (Figure 6) when they were asked to choose all street signs.



Figure 6: Two street signs in Google ReCAPTCHA v2.

The two CAPTCHAs also reported high number of errors in Category A and D. As they both were

based on image recognition where images were displayed with different level of detail, it was observed that many participants had to zoom in an image to check its relevancy and zoom out to continue the completion (Category D). Moreover, when participants were zooming in and out to inspect the detail, they might have tapped on an image by accident as the images in such tests occupied the whole screen (Category A). Such issues suggest that image recognition based CAPTCHAs need to consider using clearer instruction to avoid confusion and choosing images that are more appropriate (e.g., clarity, representation, distinctiveness etc.) when being displayed on a mobile screen. Moreover, it should provide a mechanism to allow users to restart after making mistakes instead of carrying on the completion of the test.

There was also a specific issue to do with Google ReCAPTCHA v2 when a test was asking participants to select all images in two scenarios: (1) Click verify once there are none left and (2) if there are none, click skip. Some participants were unaware of the dynamics of the first scenario. That is, if they only chose all relevant images displayed initially and clicked verify without keeping choosing the new images displayed, it would lead to a failure.

Last but not least, it was found that most Category D errors reported with KeyCAPTCHA were due to the lack of a quick restart button. When participants moved puzzle pieces into the wrong places, it was sometimes not easy to locate those wrong ones as they were already blended into the image. In that case, many of them preferred to reload the form to start again.

3.4 System Usability Scale (SUS)

The SUS scores for all CAPTCHAs are shown in Figure 7 (min: 65%, max: 85%, SD: 6.07%, Mean: 74.29%) where the benchmarking score is 62.5%. The highest SUS score was found with TapCHA which was 85%.

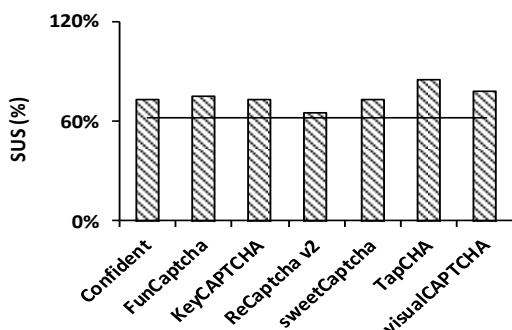


Figure 7: Average SUS score for each CAPTCHA

Apart from Google ReCAPTCHA v2 which achieved only 65%, all other CAPTCHAs achieved over 70% indicating participants' high acceptance. Note various studies have suggested that an aver-

age system should achieve at least 60 in SUS (Bangor et al. 2009; Lewis & Sauro 2009).

4. DESIGN IMPLICATIONS

Based on the discussions above, the following guidelines for designing mobile friendly CAPTCHA schemes are proposed.

Guideline 1: The CAPTCHA scheme must be designed in a way to only rely on native gesture interactions on mobile devices (e.g., tap and swipe). Our study shows that CAPTCHA schemes that are native to mobile devices such as sweetCaptcha, visualCAPTCHA and TapCHA performed generally better than the compatible ones such as Google ReCAPTCHA v2 and KeyCAPTCHA (e.g., click->tap, drag and drop -> swipe).

Guideline 2: The CAPTCHA scheme should utilise system feedback methods to help users monitor their actions constantly when solving the challenge. This is on the contrary of character-recognition based CAPTCHAs where such feedback is only needed at the challenge level (i.e., after typing all characters and submitting the answer). For example, when an object is tapped, a haptic feedback could be triggered and corresponding colour change could occur to confirm the selection with the user to help them identify potential issues.

Guideline 3: The CAPTCHA test and the objects presented in the test must be rendered properly on the mobile screen without the need for any additional user action such as pinching (zooming in/out) and swiping (scrolling up/down and/or left/right). For example, the CAPTCHA test needs to fit in one screen long to avoid excessive or unnecessary scrolling. Moreover, any object rendered and presented in the CAPTCHA test must be easily recognised and distinguished. If real-life image objects are used, the level of detail should be appropriate to minimise the risk of misinterpretation (e.g., Google ReCAPTCHA version 2). Our study shows that defective user errors were found with all CAPTCHA schemes tested and unnecessary errors were reported with 6 out of 7 where they are all object recognition based.

Guideline 4: The CAPTCHA scheme should be designed with clear instruction to help user understand what they are expected to do in a test. This is particularly important when the scheme uses real-life image objects. Our study shows that Google ReCAPTCHA v2 and Confident CAPTCHA, which use real-image objects in the tests, registered the highest number of misinterpreted and/or misjudged errors due to ambiguous instruction provided.

Guideline 5: when a number of sub tasks are used in a CAPTCHA test for added security measure (e.g., FunCaptcha, Google ReCAPTCHA version 2, Confident CAPTCHA), there must be a mechanism

to support redo and undo so as to help users recover from errors. It should be noted that supporting redo and undo does not mean the system should provide feedback on the outcome of each completion as it will weaken the security of a CAPTCHA. For example, restart/reset button can be provided to help users quickly restart the same test after making mistakes instead of carrying on the completion of the test.

5. CONCLUSION AND FUTURE WORK

In this paper, we examined the usability performance of 7 CAPTCHAs representing 5 different types of mobile friendly CAPTCHA schemes in a user study with 20 participants. The purpose of this study was to investigate the best design practices based on usability issues identified in these CAPTCHAs rather than comparing them to find out which one is more advantageous. The results show that most schemes have outperformed the benchmarking Google ReCAPTCHA v1 in all aspects indicating their suitability for mobile devices. However, we also noticed that these schemes all presented usability issues to certain extent in the test. Interestingly, some issues identified in our study such as instruction clarity and the lack of support of redo/undo are often seen as more general design issues for all computer systems (Nielsen 1994; Reynaga et al. 2015). This suggests that designing a mobile friendly CAPTCHA scheme should not only follow specific design guidelines but also need to check universal system design guidelines. The list of design guidelines are proposed based on the findings which need to be further validated by including more CAPTCHA schemes in the future. It should note that a well-designed CAPTCHA scheme should address both usability and security requirements to a high standard. This means the study also needs to be extended by looking into the security issues of interactive CAPTCHAs as over-emphasizing on the usability side may have an impact on the security side which is also essential to a CAPTCHA design.

6. REFERENCES

- Banday, M. T., Shah, N. A.: Image flip CAPTCHA. The ISC International Journal of Information Security, 1(2) (2015).
- Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of usability studies, 4(3), 114-123 (2009).
- Bursztein, E. Moscicki, A. Fabry, C. Bethard, S. Mitchell, J.C. Jurafsky, D. (2014) Easy does it: more usable CAPTCHAs. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, Toronto, Canada, 26 April 2014, pp. 2637-2646. ACM, New York.
- Brooke, J.: SUS - A quick and dirty usability scale. In: P.W., Thomas, B., Weerdmeester, B.A., McClelland, A.L. (eds.) Usability Evaluation in Industry, pp. 189 - 194. Taylor & Francis, London, UK (1996)
- Chaudhari, S. K., Deshpande, A. R., Bendale, S. B., Kotian, R. V.: 3D drag-n-drop CAPTCHA enhanced security through CAPTCHA. In: Proceedings of the International Conference & Workshop on Emerging Trends in Technology, pp. 598-601. ACM (2011)
- Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Designing Human Friendly Human Interaction Proofs (HIPs). In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 711-720. ACM, New York (2005)
- Chow, R., Golle, P., Jakobsson, M., Wang, L., Wang, X.: Making captchas clickable. In: Proceedings of the 9th workshop on Mobile computing systems and applications, pp. 91-94. ACM, New York (2008)
- comScore: 2015 U.S. Digital Future in Focus, <https://www.comscore.com/Insights/Presentation-s-and-Whitepapers/2015/2015-US-Digital-Future-in-Focus>
- Conti, M., Guarisco, C., Spolaor, R.: CAPTCHAstar! A novel CAPTCHA based on interactive shape discovery. In: International Conference on Applied Cryptography and Network Security, pp. 611-628. Springer International Publishing (2016)
- D'Souza, D., Polina, P. C., Yampolskiy, R. V.: Avatar captcha: Telling computers and humans apart via face classification. In: 2012 IEEE International Conference on Electro/Information Technology (EIT), pp. 1-6. IEEE (2012)
- Desai, A., Patadia, P.: Drag and drop: A better approach to CAPTCHA. In: 2009 Annual IEEE India Conference (INDICON), pp. 1-4. IEEE (2009)
- Gao, H., Yao, D., Liu, H., Liu, X., Wang, L.: A novel image based CAPTCHA using jig-saw puzzle. In: 2010 IEEE 13th International Conference on Computational Science and Engineering (CSE), pp. 351-356. IEEE (2010)
- Gossweiler, R., Kamvar, M., Baluja, S.: What's up CAPTCHA?: a CAPTCHA based on image orientation. In: Proceedings of the 18th International Conference on World wide web, pp. 841 - 850. ACM (2009)
- Jiang, N., Tian, F.: A novel gesture-based CAPTCHA design for smart devices. In: Proceedings of the 27th International BCS Human

- Computer Interaction Conference, p. 49. British Computer Society (2013)
- Jiang, N., Dogan, H.: A gesture-based captcha design supporting mobile devices. In: Proceedings of the 2015 British HCI Conference, pp. 202-207. ACM, New York (2015)
- Kaloom, S., Ziauddin, S., Abbasi, A. R.: An image-based CAPTCHA scheme exploiting human appearance characteristics. *TIIS*, 6(2), 734-750 (2012)
- Kim, J., Kim, S., Yang, J., Ryu, J. H., Wohn, K.: FaceCAPTCHA: a CAPTCHA that identifies the gender of face images unrecognized by existing gender classifiers. *Multimedia tools and applications*, 72(2), pp. 1215-1237 (2014)
- Kim, J. W., Chung, W. K., Cho, H. G.: A new image-based CAPTCHA using the orientation of the polygonally cropped sub-images. *The Visual Computer*, 26(6), pp. 1135-1143 (2010).
- Leiva, L. A., Alvaro, F.: μ captcha: Human Interaction Proofs Tailored to Touch-Capable Devices via Math Handwriting. *International Journal of Human-Computer Interaction*, 31(7), 457-471 (2015).
- Lewis, J. R., Sauro, J.: The factor structure of the system usability scale. *International Conference on Human Centered Design*. Springer Berlin Heidelberg (2009)
- Lin, R., Huang, S. Y., Bell, G. B., Lee, Y. K.: A new CAPTCHA interface design for mobile devices. In: Proceedings of the Twelfth Australasian User Interface Conference, Volume 117, pp. 3-8. Australian Computer Society, Inc. (2011)
- Nielsen, J.: Enhancing the explanatory power of usability heuristics. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 152-158. ACM (1994)
- Ofcom. (2015) The Communications Market Report 2015. https://www.ofcom.org.uk/__data/assets/pdf_file/0022/20668/cmr_uk_2015.pdf (20 March 2017)
- Okada, M., Matsuyama, S.: New captcha for smartphones and tablet pc. In: Consumer Communications and Networking Conference (CCNC), pp. 34-35. IEEE (2012)
- Pew Research Center: U.S. Smartphone Use in 2015. Pew Research Center, <http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015>
- Reynaga, G., Chiasson, S., van Oorschot, P. C.: Heuristics for the evaluation of captchas on smartphones. In: Proceedings of the 2015 British HCI Conference, pp. 126-135. ACM (2015)
- Roudaut, A., Huot, S., Lecolinet, E.: TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In: Proceedings of the working conference on Advanced visual interfaces, pp. 146-153. ACM (2008)
- Saxena, A., Chauhan, N. S., Reddy, S. K., Vangal, A. S., Rodriguez, D. P.: A new scheme for mobile based CAPTCHA service on Cloud. In: 2012 IEEE International Conference on Cloud Computing in Emerging Markets (CEEM), pp. 1 – 6. IEEE (2012)
- Tsuruta, Y., Takaya, M., Yamamura, A.: CAPTCHA suitable for smartphones. In: Information and Communication Technology - EurAsia Conference. Springer Berlin Heidelberg (2013)
- Vogel, D., Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 657-666. ACM (2007)
- Von Ahn, L., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically. *Communications of the ACM*, 47(2), 56-60 (2004)
- Wismer, A. J., Madathil, K. C., Koikkara, R., Juang, K. A., Greenstein, J. S.: Evaluating the usability of CAPTCHAs on a mobile device with voice and touch input. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 56, No. 1, pp. 1228-1232. Sage CA: Los Angeles, CA (2012)
- Yan, J. El Ahmad, A. S. (2008) Usability of CAPTCHAs or usability issues in CAPTCHA design. In Proceedings of the 4th Symposium on Usable Privacy and Security, Pittsburgh, PA, July 2008, pp. 44-52. ACM, New York.
- Ye, Q. B., Wei, T. E., Jeng, A. B., Lee, H. M., Wu, K. P.: DDIM-CAPTCHA: A Novel Drag-n-Drop Interactive Masking CAPTCHA against the Third Party Human Attacks. In: 2013 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 158-163. IEEE (2013)