zkCAPTCHA: Zero-Knowledge Bot Detection for Mobile Devices

ABSTRACT

CAPTCHA systems have been widely deployed to identify and block fraudulent bot traffic. However, current solutions, such as Google reCAPTCHA, often require additional user actions and need to send the attestation data back to the server, raising privacy concerns. To address these problems, in this paper we present the first zero-knowledge proof based CAPTCHA system, zkCAPTCHA, for mobile devices. zkCAPTCHA is invisible to users and does not reveal any sensitive sensor data to the server. We demonstrate that bot traffic can be accurately discovered by studying the motion sensor outputs during touch events and propose several models to accomplish this task. For each model, we elaborate on the design of the zero-knowledge proof system and make the implementation open-source. Using public datasets and data collected from 10 participants, we evaluate the accuracy and time consumption of each model and choose the ideal model for mobile devices. Then, we integrate zkCAPTCHA in Brave browser on both iOS and Android platforms and further test the usability. In addition, we demonstrate that zkCAPTCHA can be further extended to protect against click farming. We show that zkCAPTCHA does not require trust in operation system, app integrity, and is effective against replay attacks.

1 INTRODUCTION

Aim. We design a novel sensor-based privacy-preserving bot detection system for mobile devices, zkCAPTCHA, using zero-knowledge proofs. zkCAPTCHA achieves the following properties:

- Does not reveal any sensitive information to the server
- Does not introduce additional operations from the user
- Does not require trust on the app code
- Does not rely on TEE (Trusted Execution Environment)

Design.

Contributions. We make the following contributions:

- We propose a novel context-aware bot detection system that requires no additional user action.
- We are the first to design a ZKP based invisible CAPTCHA solution for both mobile apps and websites.
- We are the first to implement ZKP for several machine learning models and make it open source.
- We integrate our system into mobile Brave and benchmark the performance of several bot detection models.

2 BACKGROUND

Background

3 SYSTEM DESIGN

3.1 Threat Model

We assume there is a TPM (Trusted Platform Module) in the mobile device that collects and signs sensor data.

The goal of an attacker is to fraudulently get more ad rewards via bot operations. The attacker has the following capabilities:

- Compromise the OS
- Modify the app code
- Run app in simulators
- Fake sensor outputs (but they cannot sign the data using TPM's key)
- Know the client-side defence

3.2 Bot Detection

Bot Detection

3.3 Zero-Knowledge Proof

Zero-Knowledge Proof

4 zkCAPTCHA EXTENSIONS

4.1 Proof of Movement

4.2 Proof of Walk

5 EVALUATION

Evaluation

6 DISCUSSION

Discussion

7 RELATED WORK

Related Work

7.1 Privacy Implications of Motion Sensor Data

On both iOS and Android, the access to motion sensors does not require explicit user permission; the accelerometer and gyroscope can also be accessed from a mobile website via JavaScript. Previous studies have shown that this data could expose sensitive information about a user. In particular, TouchLogger [3], TapLogger [18], and ACCessory [12] can infer user inputs on a touch screen an steal user passwords based on the device acceleration data during touch events. Mehrnezhad et al. demonstrated that similar attacks can also be launched via Javascript [11]. In addition, extensive studies have proven that user activity can be accurately tracked from the motion data [13, 15]. Other researchers have also shown that personal user information, such as gender, age, weight, and height can be leaked from the sensory data [5, 10]. Most recently, Zhang et al. [20] revealed that a globally unique device fingerprint can be generated from the motion sensor data. These studies strongly motivate us to design a privacy-preserving CAPTCHA scheme that does not reveal any sensitive sensor data to the server.

7.2 Bot Detection

To prevent automated programs, or bots, from abusing online services, the widely adopted solution is to deploy a CAPTCHA system. The early form of CAPTCHA typically requires users to identify text from a distorted image. For Google reCAPTCHA, the most popular CAPTCHA service, user inputs are also collected to help Google

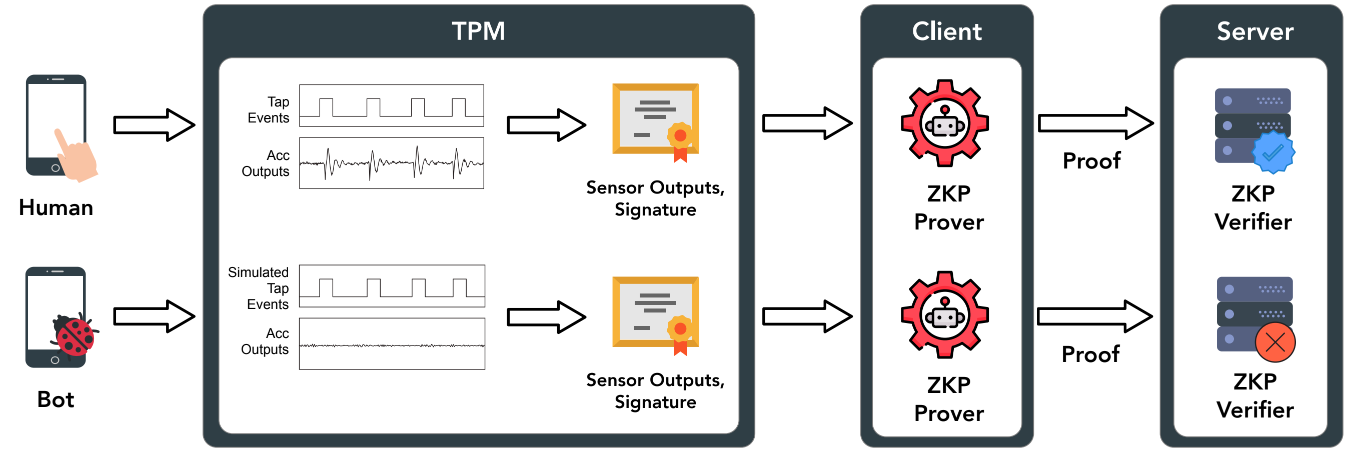

Figure 1: zkCAPTCHA schema

digitalise printed documents [17]. However, text-based CAPTCHA schemes have been proven to be insecure as machines achieved 99.8% success rate in identifying distorted text [4, 19?]. Therefore, CAPTCHA service providers, such as Google, started to test image-based CAPTCHA schemes, which require users to select images that match given description [?]. In the same time, reCAPTCHA users also help Google label images for free. Nevertheless, Sivakorn et al. demonstrated that more than 70% of image-based Google reCAPTCHA and Facebook image CAPTCHA can be efficiently solved using deep learning [? ?]. In addition, these CAPTCHA systems also deliver bad user experience, especially when running on mobile devices [14]. To counter this, Google reCAPTCHA v2 use a risk analysis engine to avoid interrupting users unnecessarily [?]. This engine collects and analyses relevant data during click events to attest the humanness of the user. The latest reCAPTCHA v3 no longer requires users to click a button. Instead, it studies user interactions within a webpage and gives a score that represents the likelihood that a user is a human [?]. Although these CAPTCHA schemes are invisible to users, a plethora of sensitive data, including cookies, browser plugins, and all JavaScript objects, is collected [?]. Needless to say, these wide data collection activity raise privacy concerns. To conform to data protection acts, such as the California Online Privacy Protection Act (CalOPPA) and the EU General Data Protection Regulation (GDPR), Google requires every website using reCAPTCHA to include a privacy policy to give consent to the data collection to use the service [?].

7.3 Privacy Preserving and Verifiable Machine Learning

Privacy preserving evaluation of machine learning models has become of particular interest given the changes in regulations (maybe cite GDPR) or events increasing the general public awareness on how private data is used to track users (cite something). [1] Several approaches are present in current literature. On the one hand, we have Homomorphic Encryption based schemes [2, 7, 8], where the user encrypts the data over which the model has to be evaluated and sends it to the server. Then the server evaluates over the encrypted data and sends back the result to the user. This method is both private and verifiable, as the server never gets to see the plain user data, but is the evaluating the model, and hence is convinced of the validity of the output of the computation. However, such schemes centralise the evaluation of ML models, which can become problematic when a high number of requests are received. [2] Moreover, evaluating ML models over encrypted data gives more restrictions than tha ZK case, as non-linear and non-polynomial functions cannot be computed (limiting like that the application of several models as random forests and forcing an approximation to linear functions in many other such as logistic regression or (D)NN).

[3]

[1] IQ: Maybe this could be motivated in the introduction

[2] IQ: Read the evaluations on the referenced papers to try and make a point here.

[3] IQ: Here would be cool some references and numbers on how this affects accuracy.

Another approach to offer privacy preserving machine learning is to evaluate the model locally, avoiding data to be sent to the server. However, if, unlike zkCAPTCHA, such approach is taken without proving correct evaluation of the model [1, 9, 16], verification is lost. In these papers the model is evaluated for targeted advertising, which can be argued that users are interested in evaluating the latter correctly, removing like that the need of verifying the correct evaluation. However, in other cases (such as bot detection) the user's interest might be of faking the evaluation model, and therefore such limits open the gap for user attacks.

To the best of our knowledge, the only paper that aims at solving this problem with provable machine learning evaluated locally is MoRePriv [6]. However, this paper is outdated in the proofs they use and the models applied to the data. In zkCAPTCHA we prove that more complex models can be used without a high impact on running time.

8 CONCLUSION

Conclusion

REFERENCES

[1] Mikhail Bilenko and Matthew Richardson. 2011. Predictive Client-side Profiles for Personalized Advertising. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 413–421. https://doi.org/10.1145/2020408.2020475

[2] Joppe Bos, Kristin Lauter, and Michael Naehrig. 2013. *Private Predictive Analysis on Encrypted Medical Data.* Technical Report MSR-TR-2013-81. https://www.microsoft.com/en-us/research/publication/private-predictive-analysis-on-encrypted-medical-data/

[3] Liang Cai and Hao Chen. 2011. TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion. In *Proceedings of the 6th USENIX Conference on Hot Topics in Security (HotSec'11)*. USENIX Association, Berkeley, CA, USA, 9. http://dl.acm.org/citation.cfm?id=2028040.2028049

[4] A A Chandavale, A M Sapkal, and R M Jalnekar. 2009. Algorithm to Break Visual CAPTCHA. In *2009 Second International Conference on Emerging Trends in Engineering Technology*. 258–262. https://doi.org/10.1109/ICETET.2009.24

[5] Erhan Davarci, Betul Soysal, Imran Erguler, Sabri Orhun Aydin, Onur Dincer, and Emin Anarim. 2017. Age group detection using smartphone motion sensors. In *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2201–2205.

[6] Drew Davidson, Matt Fredrikson, and Benjamin Livshits. 2014. MoRePriv: Mobile OS Support for Application Personalization and Privacy. In *Proceedings of the 30th Annual Computer Security Applications Conference (ACSAC '14)*. ACM, New York, NY, USA, 236–245. https://doi.org/10.1145/2664243.2664266

[7] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 201–210. http://dl.acm.org/citation.cfm?id=3045390.3045413

[8] Thore Graepel, Kristin Lauter, and Michael Naehrig. 2012. ML Confidential: Machine Learning on Encrypted Data. In *Lecture notes in computer science*, Vol. 7839. 1–21. https://doi.org/10.1007/978-3-642-37682-5{_}1

[9] Saikat Guha, Bin Cheng, and Paul Francis. 2011. Privad: Practical Privacy in Online Advertising. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. USENIX Association, Berkeley, CA, USA, 169–182. http://dl.acm.org/citation.cfm?id=1972457.1972475

[10] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. 2018. Protecting Sensory Data Against Sensitive Inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems (W-P2DS'18)*. ACM, New York, NY, USA, 2:1–2:6. https://doi.org/10.1145/3195258.3195260

[11] Maryam Mehrnezhad, Ehsan Toreini, Siamak F Shahandashti, and Feng Hao. 2016. Touchsignatures: identification of user touch actions and pins based on mobile sensor data via javascript. *Journal of Information Security and Applications* 26 (2016), 23–38.

[12] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. 2012. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 9.

[13] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171 (2016), 754–767. https://doi.org/10.1016/j.neucom.2015.07.085

[14] Gerardo Reynaga and Sonia Chiasson. 2013. The usability of CAPTCHAs on smartphones. In *2013 International Conference on Security and Cryptography (SECRYPT)*. IEEE, 1–8.

[15] Rubén San-Segundo, Henrik Blunck, José Moreno-Pimentel, Allan Stisen, and Manuel Gil-Martín. 2018. Robust Human Activity Recognition using smartwatches and smartphones. *Engineering Applications of Artificial Intelligence* 72 (2018), 190–202. https://doi.org/10.1016/j.engappai.2018.04.002

[16] Theja Tulabandhula, Shailesh Vaya, and Aritra Dhar. 2017. Privacy-preserving Targeted Advertising. *CoRR* abs/1710.0 (2017). http://arxiv.org/abs/1710.03275

[17] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. recaptcha: Human-based character recognition via web security measures. *Science* 321, 5895 (2008), 1465–1468.

[18] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 113–124.

[19] Jeff Yan and Ahmad Salah El Ahmad. 2008. A Low-cost Attack on a Microsoft CAPTCHA. In *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 543–554.

[20] Jiexin Zhang, Alastair R Beresford, and Ian Sheret. 2019. SensorID: Sensor Calibration Fingerprinting for Smartphones. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*. IEEE.