Accelerometer-based Device Fingerprinting for Multi-factor Mobile Authentication

Tom Van Goethem, Wout Scheepers, Davy Preuveneers, and Wouter Joosen

iMinds-DistriNet-KU Leuven
Leuven, Belgium
wout.scheepers@student.kuleuven.be,
{tom.vangoethem, davy.preuveneers, wouter.joosen}@cs.kuleuven.be

Abstract. Due to the numerous data breaches, often resulting in the disclosure of a substantial amount of user passwords, the classic authentication scheme where just a password is required to log in, has become inadequate. As a result, many popular web services now employ risk-based authentication systems where various bits of information are requested in order to determine the authenticity of the authentication request. In this risk assessment process, values consisting of geo-location, IP address and browser-fingerprint information, are typically used to detect anomalies in comparison with the user's regular behavior.
In this paper, we focus on risk-based authentication mechanisms in the setting of mobile devices, which are known to fall short of providing reliable device-related information that can be used in the risk analysis process. More specifically, we present a web-based and low-effort system that leverages accelerometer data generated by a mobile device for the purpose of device re-identification. Furthermore, we evaluate the performance of these techniques and assess the viability of embedding such a system as part of existing risk-based authentication processes.

1 Introduction

In September 2014, an attacker managed to access a privileged account on Bugzilla, a bugtracker software used by Mozilla, by simply using the credentials of the user that were leaked by a data breach on an unrelated website. As a result, the adversary was able to access security-sensitive information on flaws reported in the Firefox browser. By leveraging the obtained information, the attacker eventually tried to exploit unwitting visitors of a news website.

This is just one of the many stories where an attacker managed to gain access to a user's account, and along with the tendency of users to either re-use passwords between different websites or choose weak passwords [1], serves as a good example that passwords are no longer fitting for strong authentication. As a countermeasure to these threats, multi-factor authentication systems have recently gained in popularity. These multi-factor authentication schemes require a user to provide multiple elements that can be used to prove his identity. A popular choice is two-factor authentication, where a randomly generated passcode is

sent by SMS to the user's mobile phone. By requiring such a token, it becomes very challenging for an adversary to log in to a user's account, as this would require him to both know the password, as well as intercept the SMS message.

Although the additional factor substantially improves security, there exist several drawbacks, mainly with regards to deployment and costs, that prevent tokens over SMS from being widely adopted as a second factor in authentication mechanisms. Multi-factor authentication, which can be seen either as an alternative solution, or as a complement to the two-factor authentication system, attempts to tackle these drawbacks. In multi-factor authentication systems, various pieces of information on the authenticating agent are gathered during the authentication process. This information, which can consist of the user's IP address, behavioral and contextual information, or a fingerprint of the browser he is using to authenticate, is then compared against the user's typical behavior. In case some elements from this information deviate from what is expected, the user is either denied access, or is required to use a stronger authentication method for verification, e.g. by using the aforementioned two-factor authentication.

The strength of this type of multi-factor authentication system is strongly dependent on the trustworthiness of the acquired information. While it has been shown that fingerprint information gathered by a modern browser on a desktop computer typically yields high entropy [2], the fingerprints obtained from mobile devices carry a lot of similarity [3, 4], making them an unsuitable candidate in a multi-factor authentication system. In this paper, we focus on improving the reliability of multi-factor authentication in the web, and more specifically, on mobile devices. We show how current authentication systems can be augmented with a sensor-based fingerprint, in order to evaluate whether the authenticated user is using a trusted device. By computing an adaptive similarity score, the identity provider can determine the risk that the request is illicit, and take the appropriate actions. We exemplify this type of authentication system by developing a web-based system that leverages accelerometer data in combination with controlling the mobile phone's vibration motor. Finally, we evaluate a proof-of-concept implementation on the usability and feasibility of deploying such a system in real-world scenarios.

Our main contributions are:

- An accelerometer-based device fingerprinting mechanism with adaptive similarity scores as a suitable candidate for multi-factor mobile authentication
- The integration of such a web-based multi-factor authentication solution in a contemporary identity and access management (IAM) system
- A feasibility assessment of such a system in real-world scenarios

The rest of this paper is structured as follows: in Section 2, we sketch a brief background on fingerprinting and authentication. In Section 3, we motivate our approach and describe the implementation of a proof-of-concept application. In Section 4, we evaluate the application. In Section 5, we briefly reflect on related work, and finally, we conclude our work in Section 6.

2 Background

Fingerprinting browsers and devices has a variety of applications, both nefarious as well as beneficial. The majority of use cases require the unique identification of a specific user, whether it is used to prevent fraud, or to track a user over different sessions. This tracking is done by gathering information about the browser and system that is being used. Examples include identification of the browser version [2], canvas fingerprinting [5], and enumerating fonts and plugins that are installed on the browser or system [6]. In most desktop environments, these methods, or a combination thereof, provide a fingerprint that can uniquely identify a user. Consequently, it is not surprising that fingerprinting is a suitable technique for multi-factor authentication systems.

However, for mobile devices, which often share the same hardware and do not allow as many customizations of the system, device fingerprints of devices that share the same brand often result in exactly the same identifier. Moreover, Spooren et al. found that the majority of fingerprinted properties are predictable [3], preventing these techniques from being used as part of authentication processes. In an attempt to overcome these shortcomings, researchers have evaluated alternative approaches that can be leveraged to collect identifying information.

Although mobile devices of the same type most likely share the same hardware, certain sensors may suffer from microscopic imperfections caused during the manufacturing process. Because the variations on the sensor data are most likely unique for each device, they become an interesting target for fingerprinting [7]. For example, Lukas et al. found that digital cameras expose a certain pattern noise, which can then be used for identification [8]. Similarly, Das et al. show that imperfections in device microphones and speakers induce anomalies in the sound that is produced and recorded [9]. Again, the variations among devices can be used for fingerprinting purposes.

Another interesting approach, is to analyze the data returned by a mobile device's accelerometer. In their research, Dey et al. describe how imperfections induced to accelerometer chips can be leveraged to create a unique fingerprint [10]. For the purpose of authentication in the context of the web, accelerometer data is particularly interesting because this data is exposed by the majority of mobile browsers, and does not require the user's consent.

3 Approach and implementation

The sensor used for fingerprinting in this work is the accelerometer. An accelerometer measures the acceleration force that is applied to the device along the three physical axes. This acceleration is expressed in meter per second squared (m/s^2). The reasons for using this sensor is that (1) nowadays every smartphone has an accelerometer, (2) accelerometer data is accessible through JavaScript in a mobile browser and (3) there is recent work about using accelerometer data for sensor fingerprinting [10].

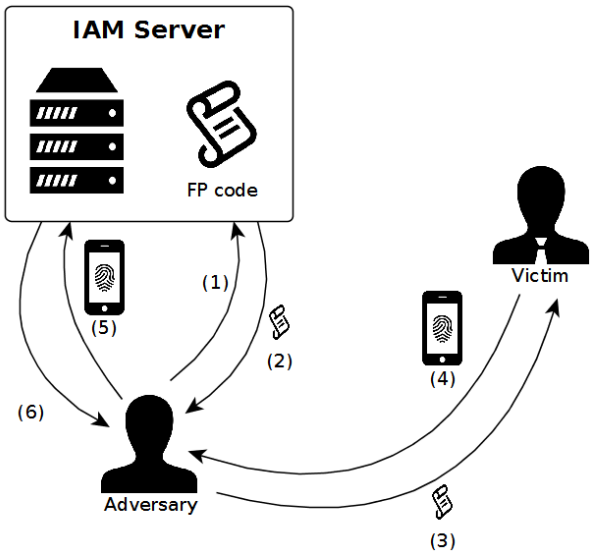

Fig. 1. Fingerprint phishing misuse scenario, enabling spoofing of the user agent.

3.1 Enduring the user agent and fingerprint spoofing threat

In contrast to prior work, where native APIs were used to obtain accelerometer data for tracking purposes, our research focuses on leveraging the accelerometer data, as exposed through browser APIs, for the purpose of improving the reliability of a multi-factor authentication system. Since the collection of fingerprinting data is done purely on the side of the client, an additional concern to be taken into consideration in our work is the mitigation against spoofing attacks where an adversary poses as the legitimate user. Consider the misuse scenario in Figure 1. Imagine an adversary who knows the credentials of a victim. He reads the fingerprinting code and sets up a phishing site to steal a user's device fingerprint. When prompted to log in, the adversary provides the stolen credentials along with the fingerprint and is authenticated to the system. It is clear that this scenario should not be possible, and thus a more clever approach is needed.

3.2 Mitigation against spoofing attacks

To counter the spoofing threat, our mitigation provides following mitigation. Upon registration, several *traces* are collected, each consisting of multiple *chunks*. A chunk contains a vibration part, i.e. a short period of time during which the device's vibration motor is enabled using the `navigator.vibrate()` API, and a non-vibration part. The length (in ms) of the vibration parts will increase for each chunk within a trace, as depicted in Figure 2. Upon registration, a trace is collected from the user's device and for each chunk, features are extracted. The assumption made here is that each chunk will have a different vibration

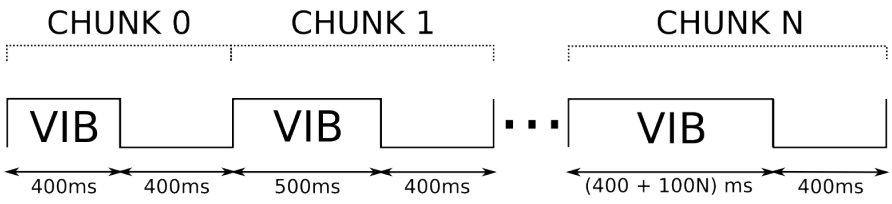

Fig. 2. A trace consisting of multiple chunks. Each chunk has a vibration part (increasing length) and a non-vibration part (fixed length).

behavior due to the difference in vibration length, and is sufficiently robust and distinguishable from other chunks.

When a user wants to log in, the user's device is asked to provide sensor data from chunks with random lengths. Upon receiving the sensor data for each requested chunk, the IAM system extracts the same features and compares them to the features from the corresponding chunk at registration. This approach makes a spoofing attempt more difficult because either the attacker needs to obtain sufficient information about a large amount of chunks, or trick the victim in providing data for the requested chunks. We consider the former attack to be unlikely, as this would require the attacker to collect accelerometer data during a considerable amount of time, something that would easily alert the user of the wrongdoing. Although the second attack, where the adversary just collects information on the requested chunks, may be viable under certain circumstances, it should be noted that the proposed mechanism is not a stand-alone authentication system. This means that an adversary not only needs to "forge" the accelerometer-based fingerprint, he also needs to know the user's credentials as well as uncover the expected values for the other aspects that are required by the multi-factor authentication system.

3.3 Sensor data collection and fingerprint extraction

For the collection of accelerometer data, we developed an HTML web application using the jQuery Mobile framework[1], collecting chunks as illustrated in Figure 2. The chunks are numbered from 0 until N. The first chunk has a vibration part of 400ms. For each subsequent chunk, the vibration part increases with 100ms. The length of the non-vibration part remains fixed at 400ms. This length was defined experimentally; 400ms without vibration is long enough for the device to return to a motionless state. When this non-vibration length is less, it is hard to distinguish between chunks because of noise the accelerometer still registers due to the momentum of the device. This results in chunk 0 having a length of 800ms, chunk 1 having a length of 900ms and so on. Chunk N has length $(400 + 100N)$ms. In this paper, we only consider accelerometer data that was

[1] `https://jquerymobile.com/`

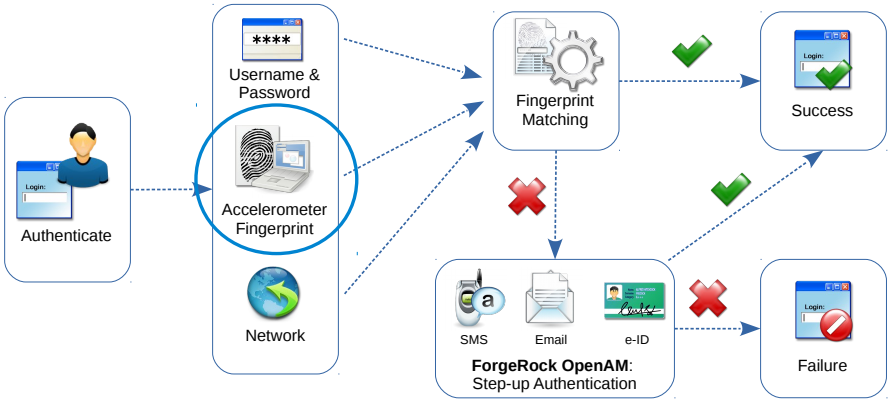

Fig. 3. Integrating accelerometer-based device fingerprinting in contemporary identity and access management systems.

collected when the device was placed on a hard surface, e.g. a table top, because we found it to result in more robust data, and is still viable in the context of authentication. This also makes potential attack scenarios more difficult, since this requires an adversary to either trick the user in placing his phone on a hard surface until a sufficient amount of accelerometer data has been obtained, or estimate the expected accelerometer values from the values that originate from vibrating on an unknown surface. While not completely unfeasible, this approach significantly constraints the viability and plausibility of attack scenarios.

The extraction of fingerprintable features is done in Matlab. After loading the data, the timestamps are normalized. Next, for each datapoint, the Root Sum Squared (RSS) is taken of the values on the three axes. According to the chunk-stamps taken at collection, the trace is split into the corresponding chunks. For each chunk, 8 time-domain features are extracted. These features are the *mean*, *standard deviation*, *average deviation*, *skewness*, *kurtosis*, *RMS amplitude*, *minimum* and *maximum*. These extracted features will be evaluated on their robustness and distinguishability among devices.

3.4 Integration in identity and access management systems

We have integrated our solution in OpenAM 12, a contemporary identity and access management system. OpenAM offers device fingerprinting and matching capabilities using client-side and server-side JavaScript technology. As shown in our previous work [3], the built-in fingerprinting code is not well suited for mobile devices. In this work, we adapted the JavaScript code to call our service to process accelerometer traces and chunks. The additional benefit of this integration is that OpenAM and our solution can be independently scaled out.

Category	Benefit	Prototype
Security	Resilient-to-Physical Observation	o
	Resilient-to-Targeted-Impersonation	o
	Resilient-to-Throttled-Guessing	x
	Resilient-to-Unthrottled-Guessing	o
	Resilient-to-Internal-Observation	x
	Resilient-to-Leaks-from-Other-Verifiers	x
	Resilient-to-Phishing	o
	Resilient-to-Theft	o
	No-Trusted-Third-Party	x
	Requiring-Explicit-Consent	x
Privacy	Unlinkable	x
Usability	Memorywise-Effortless	
	Scalable-for-Users	?
	Nothing-to-Carry	o
	Physically-Effortless	
	Easy-to-Learn	x
	Efficient-to-Use	x
	Infrequent-Errors	x
	Easy-Recovery-from-Loss	x

Table 1. Summarizing table with the security, privacy and usability benefits. (x: offers the benefit; o: almost offers the benefit; ? further investigation is needed)

4 Evaluation

4.1 Qualitative evaluation

The prototype is evaluated thoroughly based on the framework proposed by Stajano et al. [11]. This framework provides an evaluation methodology and benchmark for web authentication proposals. For evaluation purposes it makes use of a taxonomy of 19 security, privacy and usability benefits. Table 1 shows a summary of all evaluated benefits. From the 19 benefits, the prototype offers 10 benefits completely. There are 6 benefits that are almost offered by the prototype. One benefit (*scalable-for-users*) could not be evaluated, and is left as future work.

4.2 Quantitative evaluation

Feature analysis For the design of the feature matching algorithm, the extracted features from the chunks are evaluated. All features are evaluated against three criteria: (1) the distinguishability among chunks, (2) the distinguishability among devices and (3) the robustness.

As we investigate whether the device's sensor fingerprint can be used for authentication, the features trivially have to be distinguishable among devices. If this would not be the case, a fingerprint from any random device could be used

to log in. The features also need to be robust, i.e. they cannot deviate too much from each other when new data is collected from the same device. In case this would not hold, the features would have little meaning, as their values deviate too much for every feature extraction of raw data.

We analyzed the criteria above and depicted them in Figure 4. This figure depicts 8 accelerometer-based features (mean, standard deviation, average deviation, skewness, Kurtosis, root mean square, minimum and maximum) on 3 different mobile devices. The length of each box plot represents the robustness of a given feature on a particular device. This gives an idea of how consistently the device produces the extracted features for different measurements. As the robustness of the features is device dependent, this value can be used as a risk measure for a device-specific adaptive scoring function. The distinguishability corresponds to the extent to which the box plots overlap. The more they overlap, the more the values lie in the same range and are hereby harder to distinguish from each other.

The most important conclusion of this analysis is that the short chunks contain more entropy than the long chunks. This is mainly due to the exponential behavior of the mean, skewness and RMS amplitude features. The minimum feature is bad for distinguishing among chunks and devices and hereby useless.

Matching algorithm Now that the behavior of the features is known, it is possible to design and implement a matching algorithm that checks whether a login trace will be accepted or not.

Upon registration, a series of registration traces is collected. This number of traces will be defined empirically in the next section. Using the data from the multiple registration trace, we calculate an interval, which consists of a lower- and upper-bound percentile of the observed values. During the login process, a single trace, consisting of a randomized subset of chunks, is requested. For each chunk in this login trace, all features are compared to those that were extracted during the registration process. A certain feature, for a certain chunk is marked as accepted when it falls within the boundaries of the registered values. Subsequently, a score is computed for each chunk, which is based on the accepted feature-values. Because the robustness of each feature is different, we attributed each feature a certain weight, based on their distinguishability. The final score of a specific chunk is determined as the sum of the weights of all accepted features. When this sum exceeds a certain threshold, the chunk is marked as accepted.

For a successful authentication attempt, a ratio, which can be user-defined, of all probed chunks should be classified as accepted. This ratio is directly related to the difficulty of passing a login attempt: if this ratio is set to a high value, and all chunks must match the values from the registration process, it becomes more likely that an legitimate login trace will fail due to momentary measurement inaccuracies. In our evaluation, we found that a ratio of $\frac{3}{4}$ provides a balanced end-result.

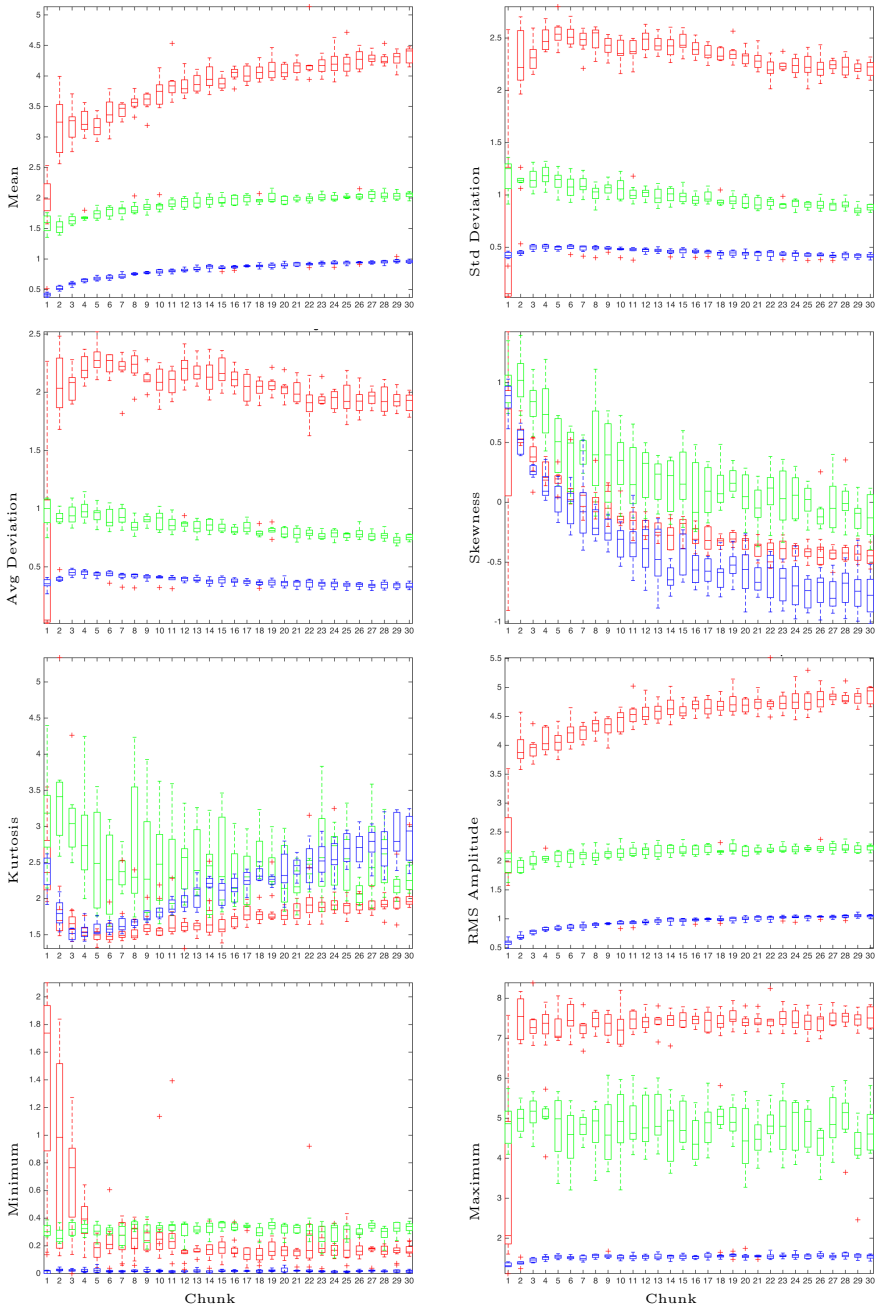

Fig. 4. Distinguishability and robustness of 8 accelerometer-based features of different mobiles devices (red: Motorola Moto G, green: Huawei P8 Lite, blue: Google Nexus 4). The length of each box plot is a measure for the device-specific robustness of the feature.

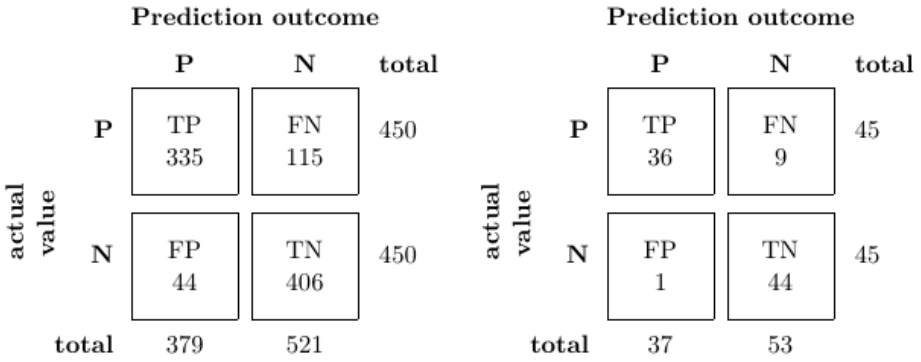

Fig. 5. Confusion matrices for all 15 devices. Left: chunks, right: traces.

4.3 Experiment setup

For conducting experiments, 10 registration traces consisting of 10 chunks were gathered together with 3 login traces. The dataset consists of data from 15 devices: 6 Google Nexus 5's, 3 Google Nexus 4's, 3 OnePlus One's, a Samsung Galaxy S4 mini, a Samsung Galaxy S5 and an LG G3.

Performance metrics For measuring classification performance, each login trace of the users device is checked by the matching algorithm. From this, the amount of *true positives* TP_i and *false negatives* FN_i is calculated for each chunk i. A true positive occurs when a chunk collected by the registered device is accepted, i.e. a chunk that needed to be classified as accepted is accepted. A false negative occurs when a chunk collected by the registered device is rejected, i.e. a chunk that needed to be classified as accepted is rejected.

To investigate how the algorithm behaves for login traces from different devices, three login traces are selected randomly. These traces will be referred to as *alien traces*. The results of the alien traces are used to calculate the *false positives* FP_i and *true negatives* TN_i for each chunk i. A false positive occurs when a chunk collected from an alien trace is classified as accepted, i.e. a chunk that needed to be classified as rejected is accepted. A true negative occurs when a chunk collected from an alien trace is classified as rejected, i.e. a chunk that needed to be classified as rejected is rejected.

To measure the performance of the matching algorithm (which acts as a binary classifier), the *true positive rate* (TPR) and *false positive rate* (FPR) are defined as follows (for each chunk i):

$$TPR_i = \frac{(TP_i)}{(TP_i + FN_i)}, FPR_i = \frac{(FP_i)}{(FP_i + TN_i)}$$

These metrics can be plotted as an ROC-curve, where the FPR is shown on the X-axis and the TPR on the Y-axis. It is obvious that in an authentication

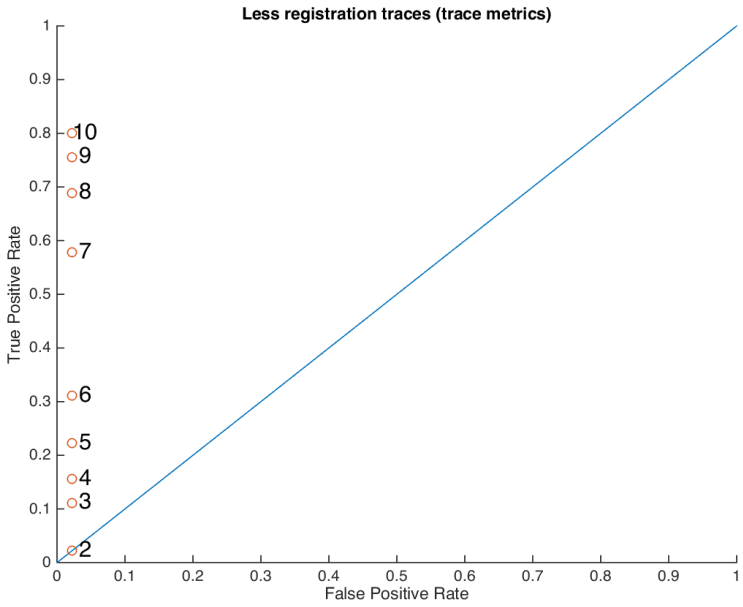

Fig. 6. ROC with the number of registration traces varying from 10 to 2. The threshold factor $t = \frac{1}{2}$.

system, the focus should be on minimizing the false positive rate. A trace of an alien device should not be marked as accepted by the matching algorithm. Otherwise logging in would be possible from any device, making accelerometer-based fingerprinting useless for authentication.

Results We investigate the performance of the matching algorithm on 10 registration traces and 10 probed chunks. The results are shown in a confusion matrix in Figure 5. As there are 15 devices that test 6 traces, the results show the classification of 900 chunks and 90 traces. For the chunk classification, the TPR and FPR are 0.7444 and 0.0978 respectively. Hereby, the classification is considered good, as the false positive rate is low. When the trace classification's TPR and FPR are calculated, they yield 0.8000 and 0.0222 respectively. The true positive rate is even higher and the false positive rate lower, indicating a good trace classification.

The false positives of the chunks can be considered insignificant, as they almost all are filtered by the trace acceptance (which depends on the ratio parameter). Only one alien trace was classified as accepted. As the user still needs to fill in a username and password, the classification algorithm is definitely sufficient.

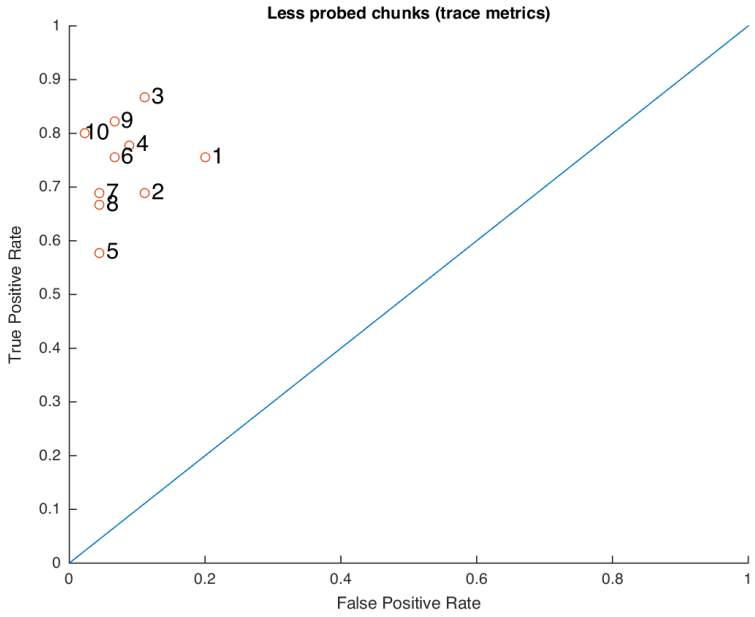

Fig. 7. ROC with number of probed chunks varying from 10 to 1.

For each device, the number of registration traces is varied from 10 to 2. The lower bound is 2 because this is the minimum amount of registration traces needed to calculate the intervals used for matching. The intuition behind lowering the amount of registration traces is that the calculated intervals will be less accurate. This inaccuracy will probably result in more chunk rejections. There will be more FPs and FNs as the number of registration traces is reduced.

The next experiment investigates what happens when the amount of registration traces is reduced. The amount of registration traces are varied from 10 to 2. The results in Figure 6 show the classification performance of the traces. As can be seen, the true positive rate declines when we reduce the amount of registration traces. This result confirms the intuition that the intervals used for matching become less accurate when the number of registration traces decreases. Notably, the false positive rate remains more or less constant. This means that a reduction in registration traces only induces more false negatives, and no false positives. In the results described above, the threshold factor t was set to $\frac{1}{2}$. It is clear that there is a trade-off between usability and security. The more registration traces are collected, the better matching and hereby the security. But taking many registration traces impact the usability by imposing a longer registration time.

Figure 7 shows what happens when the number of probed chunks decreases. All classifications yield approximately the same results. The intervals stay as

accurate as before because they stay the same. Because every chunk has a good classification performance, every possible subset of chunks could be used for logging in. This behavior is desirable, as it should be possible to collect any subset for enforcing the spoofing mitigation approach. When a small subset of chunks is probed, the probability of fingerprint phishing is lower because an attacker his a harder time collecting the data of every possible chunk-probe.

Discussion We note that sensor fingerprinting through the browser is only possible because browser vendors implement APIs for collecting sensor data. The purpose of this is to enrich the user's experience, like automatically rotating the display or using device motion in games. However, browser vendors could investigate to which extent the induction of bias or noise is possible to mitigate privacy concerns of tracking [12]. By doing this, they could distort device identification through sensor fingerprinting. In that case, the authentication scheme presented can not be used, and other authentication mechanism are needed. These methods could include the fingerprinting of user behavior, like for example capturing a user's search patterns or battery consumption.

Furthermore, since the security properties of the proposed mechanism are not absolute, the proposed method should not be considered a standalone authentication mechanism, but rather part of a multi-factor authentication mechanism where multiple features are used to determine the authenticity of an authentication request.

5 Related work

To counter the imminent threat of account compromise, multi-factor authentication is one of the most popular solutions. Motivated by their ubiquitousness, the mobile devices of users are often leveraged in a multi-factor authentication setup. For instance, mobile devices can be trusted to compute a One Time Password [13], or in authentication schemes, where they are used to scan a QR-code [14], communicate over Near Field Communication (NFC) technology [15], or emit and receive inaudible soundwaves [16]. Unfortunately, the majority of these solutions can only be used when the user attempts to authenticate on his desktop computer or laptop. Contrastingly, in this paper, we focus on improving the security of users who attempt to authenticate using just their mobile device [17].

For this purpose, we leverage sensor data generated by the accelerometer chip in the mobile device, which was found to be unique due to manufacturing imprecisions [10, 7]. Prior research has evaluated using the accelerometer in the context of authentication. In contrast to fingerprinting the imprecisions of the accelerometer chip, Wang et al. leveraged the accelerometer data to analyze a user's gestures, and perform authentication on the basis of the uniqueness of the gestures [18]. Similarly, Mayrhofer and Gellersen proposed a device-to-device authentication mechanism where two devices are shaken together, and

thus share very similar accelerometer data, in order to generate authenticated, secret keys [19].

In the context of analyzing sensor data to uniquely identify devices, researchers have evaluated various sensors for imperfections. Examples include pattern noise exposed the camera [8, 20, 21], imperfections of a device's acoustic components [9, 22], or using the gyroscope to recognize speech [23]. We can further generalize our solution by fusing behavioral information about how people interact with the world, exploiting user specific traits about app usage, location, stylometry, keystroke dynamics, phone calls, etc. [24–27].

6 Conclusion

In this work, our objective was to extend an authentication system where browser fingerprinting is used as an additional authentication factor. The scope of the system is limited to authentication with mobile devices, where security and privacy threats imposed by fingerprinting are taken into account. The contributions of this paper are (1) the extension of a state-of-practice authentication system architecture for fingerprinting, (2) a prototype implementation of this architecture using accelerometer sensor data and (3) a critical assessment of this prototype.

The main conclusion is that sensor fingerprinting can be used for authentication. However, the quantitative evaluation has shown that there is a trade-off between usability and security, depending on the amount of traces used for registration. The limitations of this work are the absence of a large scale sensor fingerprint assessment, the focus on only one sensor for fingerprinting and the use of a limited set of features for fingerprint extraction.

Future work could conduct a more extensive scalability assessment, with regard to the uniqueness of sensor fingerprints and the performance of the system on a large scale. As only the accelerometer sensor is used, it would also be interesting to investigate to which extent other sensors such as speakers, microphones and cameras can be fingerprinted for authentication purposes.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven, and by the MediaTrust and TRU-BLISS projects funded by iMinds.

References

1. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of the 16th international conference on World Wide Web, ACM (2007) 657–666
2. Eckersley, P.: How unique is your web browser? In: Privacy Enhancing Technologies, Springer (2010) 1–18

3. Spooren, J., Preuveneers, D., Joosen, W.: Mobile device fingerprinting considered harmful for risk-based authentication. In: Proceedings of the Eighth European Workshop on System Security, ACM (2015) 6
4. Hupperich, T., Maiorca, D., Kührer, M., Holz, T., Giacinto, G.: On the robustness of mobile device fingerprinting: Can mobile users escape modern web-tracking mechanisms? In: Proceedings of the 31st Annual Computer Security Applications Conference, ACM (2015) 191–200
5. Mowery, K., Shacham, H.: Pixel perfect: Fingerprinting canvas in html5. Proceedings of W2SP (2012)
6. Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., Preneel, B.: Fpdetective: Dusting the web for fingerprinters. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, ACM (2013) 1129–1140
7. Bojinov, H., Michalevsky, Y., Nakibly, G., Boneh, D.: Mobile device identification via sensor fingerprinting. arXiv preprint arXiv:1408.1416 (2014)
8. Lukas, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. Information Forensics and Security, IEEE Transactions on **1**(2) (2006) 205–214
9. Das, A., Borisov, N., Caesar, M.: Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM (2014) 441–452
10. Dey, S., Roy, N., Xu, W., Choudhury, R.R., Nelakuditi, S.: Accelprint: Imperfections of accelerometers make smartphones trackable. In: Proceedings of the Network and Distributed System Security Symposium (NDSS). (2014)
11. Bonneau, J., Herley, C., van Oorschot, P., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: Security and Privacy (SP), 2012 IEEE Symposium on. (May 2012) 553–567
12. Das, A., Borisov, N., Caesar, M.: Exploring ways to mitigate sensor-based smartphone fingerprinting. CoRR **abs/1503.01874** (2015)
13. Aloul, F., Zahidi, S., El-Hajj, W.: Two factor authentication using mobile phones. In: Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on, IEEE (2009) 641–644
14. Dodson, B., Sengupta, D., Boneh, D., Lam, M.S.: Secure, consumer-friendly web authentication and payments with a phone. In: Mobile Computing, Applications, and Services. Springer (2012) 17–38
15. Alpár, G., Batina, L., Verdult, R.: Using NFC phones for proving credentials. In: Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. Springer (2012) 317–330
16. Google: Slicklogin
17. Preuveneers, D., Joosen, W.: Smartauth: Dynamic context fingerprinting for continuous user authentication. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing. SAC '15, New York, NY, USA, ACM (2015) 2185–2191
18. Wang, H., Lymberopoulos, D., Liu, J.: Sensor-based user authentication. In: Wireless Sensor Networks. Springer (2015) 168–185
19. Mayrhofer, R., Gellersen, H.: Shake well before use: Authentication based on accelerometer data. In: Pervasive computing. Springer (2007) 144–161
20. Chen, M., Fridrich, J., Goljan, M., Lukáš, J.: Determining image origin and integrity using sensor noise. Information Forensics and Security, IEEE Transactions on **3**(1) (2008) 74–90

21. Bertini, F., Sharma, R., Iannì, A., Montesi, D.: Profile resolution across multilayer networks through smartphone camera fingerprint. In: Proceedings of the 19th International Database Engineering & Applications Symposium, ACM (2015) 23–32
22. Chen, D., Mao, X., Qin, Z., Wang, W., Li, X.Y., Qin, Z.: Wireless device authentication using acoustic hardware fingerprints. In: Big Data Computing and Communications. Springer (2015) 193–204
23. Michalevsky, Y., Boneh, D., Nakibly, G.: Gyrophone: Recognizing speech from gyroscope signals. In: Proc. 23rd USENIX Security Symposium (SEC'14), USENIX Association. (2014)
24. Fridman, L., Weber, S., Greenstadt, R., Kam, M.: Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. CoRR **abs/1503.08479** (2015)
25. Antal, M., Szabo, L.Z., Laszlo, I.: Keystroke dynamics on android platform. Procedia Technology **19** (2015) 820 – 826 8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9-10 October 2014, Tirgu Mures, Romania.
26. Li, F., Clarke, N.L., Papadaki, M., Dowland, P.: Active authentication for mobile devices utilising behaviour profiling. Int. J. Inf. Sec. **13**(3) (2014) 229–244
27. Shi, E., Niu, Y., Jakobsson, M., Chow, R.: Implicit authentication through learning user behavior. In: Proceedings of the 13th International Conference on Information Security. ISC'10, Berlin, Heidelberg, Springer-Verlag (2011) 99–113