Cross-device tracking through identification of user typing behaviours

H. Yuan✉, C. Maple, C. Chen and T. Watson

A novel method of cross-device tracking based on user typing behaviours is presented. Compared with existing methods, typing behaviours can offer greater security and efficiency. When people type on their devices, a number of different factors may be considered to identify users, such as the angle and distance of contact point to the centre of the target character, the time elapsed between two typing actions and the physical force exerted on the device (which can be measured by an accelerometer). An experiment was conducted to validate the proposed model; those data are collected through an Android App developed for the purpose of this study. By collecting a reasonable amount of this type of data, it is shown that machine learning algorithms can be employed to first classify different users and subsequently authenticate users across devices.

Introduction: Cross-device tracking refers to technologies which enable tracking of users across multiple devices (such as tablets, phones, and PCs) [1]. It represents the assortment of methods used first for identifying users and subsequently to determine whether different devices are being used by the same person.

Currently, the main cross-device tracking methods are (i) username and password login systems, which allow users to log into their accounts from different devices through personal credentials [2]. (ii) Probabilistic cross-device tracking, which relies on a variety of information from multiple devices (e.g. a cookie [3], IP addresses [4], hardware identifier and device fingerprint [5]), and statistical models to infer whether those devices are used by the same person or group of people. Furthermore, in case the service provider is unable to directly implement cross-device user tracking, it is possible for third-party companies to do so by partnering with said providers. For example, websites could pass along identifying information during login to an outside tracking agency, allowing them to match user profiles on multiple devices [6].

Unfortunately, these cross-device tracking methods also create substantial privacy issues, as highly sensitive data is being collected on the user and very often shared with multiple parties. These practices are moreover extremely difficult for a consumer to control, especially when they are carried out without their own knowledge. This Letter, therefore, proposes a method of tracking users across different devices based on their typing behaviour.

As shown in Fig. 1, two users' behaviours data are collected by the sensors in two different devices, then those data are classified and modelled. After those two steps, the cross-device authentication service would identify whether those two users are the same person. If they are matching as the same person, then the cross-device based service is provided.

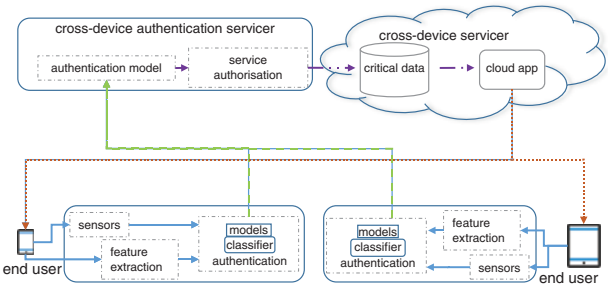

Fig. 1 *Cross-devices tracking model framework*

Experimental setup: To identify whether the proposed method produces accurate results, an experiment was set up by collecting data from devices which have a touchscreen and need a 6-digital personal identification number (PIN) code to unlock the device. While the users type the PIN code to unlock their devices, three features are detected: device movement through the accelerometer, typing time duration and tap location accuracy. All those data are collected through an Android App shown in Fig. 2*a*.

The data were collected from four different models of the Google Nexus, which include four different screen sizes (4.95, 5.96, 8.86 and 10.05 inches). For each model of the device, the test was completed by 50 different users (UEs), composed of 23 female and 27 male users, with ages that range from 21 to 53 years old. The tests were taken under normal environments and stress levels.

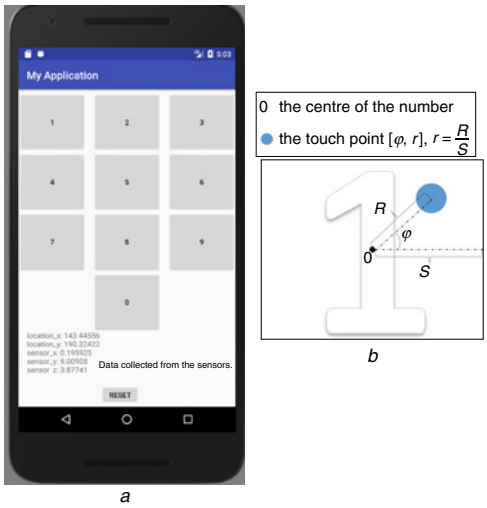

Fig. 2 *Experimental setups for data collection*
a Android platform App developed for data collection
b Parameters of the data when typing

Typing Location Accuracy: As shown in Fig. 2*b*, when a user is typing a number, e.g. the number '1' from a device keyboard layout, a point of contact with the screen is created, which has a certain distance from the centre of the target button. The parameters φ and r are used to label the touch point. Where r is defined as a ratio R/S, R is the distance between the centre of target button and the contact point, and S is the screen size. Which φ, is defined as the angle from the horizontal direction of the contact point, shown in Fig. 2*b*.

Accelerometer on tap: When a tapping event takes place, the device experiences minute physical movements which can be detected by the accelerometer. Data is therefore collected on the device's movement and accelerations along a three-dimensional (3D) space represented through the three axes (X, Y, Z) are captured by the accelerometer. To measure the volume of the overall movement detected, the equation $A_{\text{tap}} = \sqrt{a_x^2 + a_y^2 + a_z^2}$ was used, where a_x, a_y, a_z represents the vector containing the device's acceleration measurements along each of the 3D axes.

Fig. 3 illustrates a device's movements when a tapping event occurs while the user is sitting still, with the circled segments representing such events. As the users are bound to make minute movements unrelated to tapping events, the represented accelerometer data contains a visible amount of noise and errors. Therefore, the duration between two tapping events is considered as well for greater classification accuracy.

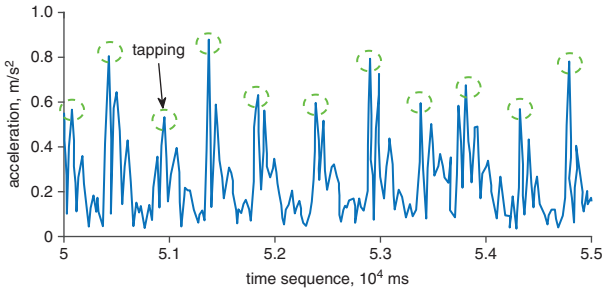

Fig. 3 *Accelerometer data when typing*

Tapping duration: As shown in Fig. 3, the distance between two circles on the x-axis represents the time spent between two screen taps. Not only is this parameter dependent on the screen size of the device being used, but it may also be dependent on the user's mood. As such, all users were requested to undergo the experiment under normal stress levels in order to minimise its impact on the considered parameters.

Data analysis model: This section presents the data analysis carried out on the information collected from the aforementioned sample of 50 different users. For the classification method, an Adaptive Boosting (AdaBoost) algorithm was used, the output of which is determined by the weighted sum of the outputs of many different weak classification methods [7]. The full methodology is shown in Algorithm 1.

Algorithm 1: AdaBoost classification algorithm

1: **function** AdaBoost($D, f(\boldsymbol{x})$)
2: $w_1(\boldsymbol{x}) = 1/n$.
3: **for** $t = 1, 2, 3, ..., T$. **do**;
4: $h_t = f(W, W_t)$; $\varepsilon_t = P_{\boldsymbol{x}\sim W_t}(h_t \neq f(\boldsymbol{x}))$
5: **if** $\varepsilon_t > 0.5$; **then**;
6: **Break**;
7: $$w_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \quad (1)$$
8: **if** $h_t = f(\boldsymbol{x})$ **then**;
9: $$w_{t+1}(\boldsymbol{x}) = \frac{w_t(\boldsymbol{x})}{Z_t} \times \exp(-w_t(\boldsymbol{x})) \quad (2)$$
10: **else** $h_t \neq f(\boldsymbol{x})$;
11: $$w_{t+1}(\boldsymbol{x}) = \frac{w_t(\boldsymbol{x})}{Z_t} \times \exp(w_t(\boldsymbol{x})) \quad (3)$$
12: $$H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} w_t h_t(\boldsymbol{x})\right)$$

Being an iterative process, the AdaBoost algorithm initialises its weak classifier's weights to be all the same $(1/n)$ and progressively changes them according to how well each method is able to classify the data. In fact, at a stage h_t if a weak classifier were to correctly identify each instance, its weights would be increased for the next iteration, thus increasing its contribution to the overall classification. Otherwise, the inverse process takes place, with a weak classifier being dropped from the algorithm if its percentage of correct classifications falls below 50%. The algorithm is composed of a set amount of iterations T.

Performance: For the purpose of this Letter, the AdaBoost algorithm was implemented multiple times according to both the number of weak classifiers utilised (from 1 to 16) and the type of data (Accelerometer data, Tap Location data, Typing duration and a combination of all three). To determine the effectiveness of the considered classification methods, Fig. 4 presents both the false acceptance ratio (FAR), defined as the ratio of the number of false positives divided by the overall observations and the false rejection ratio (FRR), determined by the number of false rejections divided by the sample size, of each implementation.

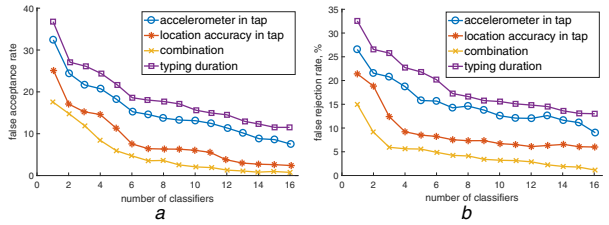

Fig. 4 *FAR and FRR by different actions and a different number of classifiers*
a FAR
b FRR

Both FAR and FRR exhibit a decreasing trend as the number of weak classifiers increases, with the classifier using a combination of all three features consistently displaying the lowest rates when comparing it to the single feature classifiers. Likewise, Fig. 5*a* shows that the three-feature classifier presents higher successful classification rates on data of same model devices than any of the one-feature classifiers, achieving with one weak classifier an accuracy rate of 68.72%, which progressively increases with each additional weak classifier included until it reaches 92.29% at 16 weak classifiers.

Similar remarks can be made by considering data from different device models, as shown in Fig. 5*b*, with the three-factor classifier once again outperforming the other one-factor classifiers, reaching an accuracy rate of 87.6% with 16 weak classifiers. It should be noted, however, that the accuracy of the classifiers used on data of different device models is consistently lower compared to those which consider same model data. This is most probably due to the fact that different screen sizes produce dissimilar typing behaviour: a bigger screen, in fact, would naturally generate larger virtual keyboards, thus affecting both typing accuracy and time spent between typing actions, while a difference in device weight is likely to influence the accelerometer readings.

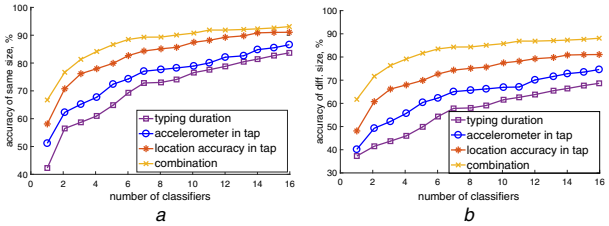

Fig. 5 *Success identification for cross devices tracking with the same and different size*
a Same sizes devices
b Different size devices

Conclusion: This Letter shows that typing behaviour can be used to implement cross-device user tracking without raising additional data privacy concerns, as the three types of data examined are to be considered less sensitive compared to user credentials or digital fingerprints. When a typing event happens, the tapping coordinate, device movement accelerate and duration can be obtained from the system. The behaviours of the same user on different devices are fairly identical. We found that combined those data can achieve a high probability (over 97%) to achieve cross-device tracking.

In the experiments described in this Letter, participants were asked to perform with a constant level of stress, and the differences in device sizes affects the accuracy. As further work, the relationship between stress level and typing behaviours could be studied, and size could be an extra domain of the behaviour classification model.

Acknowledgment: This work was supported by the EPSRC project PETRAS IoT Hub (EP/N02298X/1).

Submitted: *19 March 2018* E-first: *21 June 2018*
doi: 10.1049/el.2018.0893
One or more of the Figures in this Letter are available in colour online.

H. Yuan, C. Maple, C. Chen and T. Watson (*Cyber Security Centre, WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom*)
✉ E-mail: H.Yuan.4@warwick.ac.uk

References

1 Asplund, M., and Nadjm-Tehrani, S.: 'Attitudes and perceptions of iot security in critical societal services', *Access*, 2016, **4**, pp. 2130–2138
2 Brookman, J., Rouge, P., Alva, A., *et al.*: 'Cross-device tracking: measurement and disclosures', *Proc. Priv. Enhanc. Technol.*, 2017, **2017**, (2), pp. 133–148
3 Zimmeck, S., Li, J.S., Kim, H., *et al.*: 'A privacy analysis of cross-device tracking'. 26th {USENIX} Security Symp., Vancouver, BC, Canada, August 2017, pp. 1391–1408
4 Neufeld, E.: 'Cross-device and cross-channel identity measurement issues and guidelines', *J. Advert. Res.*, 2017, **57**, (1), pp. 109–117
5 Nikiforakis, N., Kapravelos, A., Joosen, W., *et al.*: 'Cookieless monster: exploring the ecosystem of web-based device fingerprinting'. The 2013 IEEE Symp. on Security and Privacy (SP), Berkeley, CA, USA, June 2013, pp. 541–555
6 Englehardt, S., Reisman, D., Eubank, C., *et al.*: 'Cookies that give you away: the surveillance implications of web tracking'. The 24th Int. Conf. on World Wide Web, Florence, Italy, May 2015, pp. 289–299
7 Bauer, E., and Kohavi, R.: 'An empirical comparison of voting classification algorithms: bagging, boosting, and variants', *Mach. Learn.*, 1999, **36**, (1), pp. 105–139