Towards Privacy-aware Keyboards

Krisztian Buza[1], Piroska B. Kis[2]

[1] Knowledge Discovery and Machine Learning,
Rheinische Friedrich-Wilhelms-Universität Bonn, Germany,
chrisbuza@yahoo.com
http://www.biointelligence.hu/typing.html
[2] University of Dunaujvaros, Hungary
piros@uniduna.hu

Abstract. As shown by various studies, the dynamics of typing on a keyboard is characteristic to persons. On the one hand, this may allow for person identification based on keystroke dynamics in various applications. On the other hand, in certain situations, such as chat-based anonymous helplines, web search for sensitive topics, etc., users may not want to reveal their identity. In general, there are various methods to increase the protection of personal data. In this paper, we propose the concept of *privacy-aware keyboard*, i.e., a keyboard which transmits keyboard events (such as pressing or releasing of a key) with small random delays in order to ensure that the identity of the user is difficult to be inferred from her typing dynamics. We use real-world keystroke dynamics data in order to simulate privacy-aware keyboards with uniformly random delay and Gaussian delay. The experimental results indicate that the proposed techniques may have an important contribution to keeping the anonymity of users.

Keywords: privacy, keystroke dynamics, machine learning, web search

1 Introduction

The dynamics of typing is known to be characteristic to persons [1–4]. While person identification based on keystroke dynamics may be desired in many applications, such as internet banking or online tax declaration [5], however, it may happen that the users want to remain anonymous.

As an example, let us consider the case of web search. The keywords used when searching the web, may reveal sensitive information about the users and their interests [6]. In particular, searching for particular diseases and symptoms may be an indication of health status, other keywords may allow to infer political or religious views of users, etc. We assume an attacker, who wants to gain access to such sensitive information and wants to link the pieces of information to persons. Obviously, IP-addresses linked to search queries may be highly informative, however, due to shared usage of computers and dynamic allocation of IP-addresses, they may not allow to identify users uniquely. Nevertheless, it may be possible to distinguish different users of the same computer (or the same IP-address, respectively) based on their keystroke dynamics. We note that keystroke dynamics may simply be captured by scripts running in the web browser, and the user is not likely to notice the activation of such scripts.

In order to contribute to the protection of the private information, we propose the concept of *privacy-aware keyboard*, by which we mean a keyboard that transmits keyboard events (such as pressing or releasing a key) with small random delays in order to ensure that the identity of the user is difficult to be inferred from her typing dynamics.

We perform experiments using real-world keystroke dynamics data in order to simulate privacy-aware keyboards choosing the delay from various random distributions such as uniform and Gaussian. The experimental results indicate that the identity of the users is much more difficult to be recognized in case of the privacy-aware keyboard, and therefore the proposed techniques may substantially contribute to the keeping the anonymity of users.

2 Problem Formulation

We assume an attacker whose goal is to identify the user who typed a particular text (such as keywords in the aforementioned web search scenario, or sentences in case of chat-based helplines). The attacker is able

- to run a script in the web browser that captures keystroke dynamics, and
- to use similarity-based models in order to compare keystroke dynamics.

On the one hand, we want to prevent the attacker from achieving his goal by the usage of a keyboard that transmits keyboard events with small random delays, causing the captured keystroke dynamics data to become corrupted. On the other hand, we want the corrupted keystroke dynamics data to look "natural", so that the attacker believes that he might be able to identify the user based on that data, and therefore he will *not* use other person identification techniques.

3 Our Privacy-aware Keyboard Models

Next, we define two privacy-aware keyboard models: the Uniformly Random Delay Keyboard and the Gaussian Keyboard.

After each keyboard event e (such as pressing or releasing of a key), both types of keyboards chose a random number d_e, and wait for d_e milliseconds before transmitting the signal indicating that event e occurred. The keyboards preserve the order of events: that is, in case if event e_1 occurs at time t_1 and event e_2 occurs at time t_2 and the random delay d_1 was generated for event e_1, then the random delay of d_2 to be generated for event e_2 must fulfill: $t_1+d_1 < t_2+d_2$. From which follows that $d_2 > t_1 - t_2 + d_1$.

The uniformly random delay keyboard chooses the random number d_e uniformly from the interval $[max(0, t_{pre} - t_e + d_{pre})\ ,\ d_{max}]$, where t_e and t_{pre} are the times (in milliseconds) when event e and the previous event happened, d_{pre} is the random number generated for the previous event and d_{max} is the maximal delay which is a parameter of this keyboard model.

The delay d_e associated with event e in case of the Gaussian Keyboard is $max(\mathcal{G}(\mu, \sigma)\ ,\ t_{pre} - t_e + d_{pre})$, where $\mathcal{G}(\mu, \sigma)$ is a random number generated from a Gaussian distribution with mean μ and standard deviation σ, which are parameters of this keyboard model. Symbols t_e, t_{pre} and d_{pre} denote the same as in case of the Uniformly Random Delay Keyboard.

4 Experiments

We begin this section by the introduction of our keystroke dynamics data which we used to simulate privacy-aware keyboards. This is followed by the description of the experimental settings and results.

4.1 Typing Dynamics Data

We collected keystroke dynamics data, or *typing patterns* for short, from 12 different users over several months, resulting in a collection of 548 typing patterns in total. In each of the typing sessions[3], the users were asked to type the following short text based on the English Wikipedia page about Neil Armstrong:

That's one small step for a man, one giant leap for mankind. Armstrong prepared his famous epigram on his own. In a post-flight press conference, he said that he decided on the words just prior to leaving the lunar module.

In each typing session, we measured the duration of each keystroke, i.e., the time between pressing and releasing a key. We used a self-made JavaScript application and a PHP script to capture the aforementioned time series and to save the data. We mention that the length of typing patterns varies slightly from session to session due to typing errors.

4.2 Experimental Settings

The primary goal of our experiments was to show that privacy-aware keyboards indeed make person identification difficult. In order to do that, we simulate privacy-aware keyboard by adding noise to the data according to the privacy-aware keyboard models. Specifically, we will show that the accuracy of person identification decreases dramatically in case of privacy-aware keyboards, concretely, the accuracy in case of privacy-aware keyboards is close to the accuracy

[3] The number of typing sessions was approximately the same for each user. Despite the fact that the data is balanced, the recognition of the user based on typing dynamics could lead to an imbalanced classification task, for example in case if binary classifiers are used according to the one-vs-rest schema.

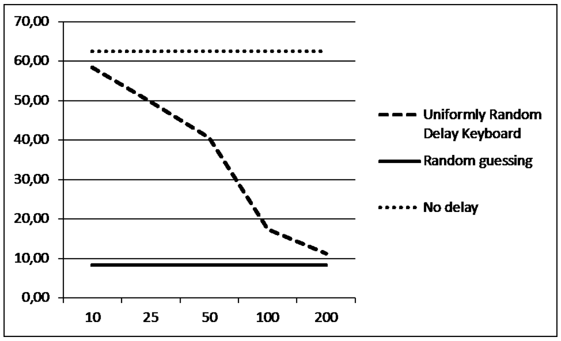

Fig. 1. Accuracy (in %) of person identification in case of Uniformly Random Delay Keyboard as function of the maximal delay d_{max}. The accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing are shown for comparison.

of random guessing. Furthermore, in order to allow to draw more general conclusions that are valid to *any* similarity-based algorithm, we analyze the similarities between the typing patterns both in case of the same user, as well as in case of different users, both for the original data, and the data "corrupted" by the privacy-aware keyboard models.

In principle, one could measure the accuracy of person identification in context of various classifiers, such as neural networks [7, 8], Hidden Markov Models [9], ensembles [10–13], or classifiers designed for imbalanced data [14, 15], see also [16] for a survey on data stream mining. However, we decided to use nearest neighbor classifiers in our experiments, because keystroke dynamics data are time series and, in case of time series data, the 1-nearest neighbor classifier (1NN) with dynamic time warping (DTW) as distance measure was shown to be competitive with complex models, such as neural networks, Hidden Markov Models or "super-kernel fusion scheme"[17, 18]. These empirical results are justified by theoretical analysis as well [19, 20]. Thus, 1NN with DTW can be considered as a representative of time-series classifiers. Furthermore, taking into account that 1NN is popular and simple to implement, we can assume that an attacker is likely to use this classifiers for person identification.

For the classification experiments, we used the *first* five typing patterns from each user as training data, and the remaining typing patterns were used as test data. This is consistent with the assumption that the attacker may have access to a few typing patterns from the *past*, while the attacker may not be able to observe the typing dynamics of a user for a very long time without being noticed. We note that the same data and train-test splits are used in the *Person Identification Challenge*.[4]

[4] http://biointelligence.hu/typing-challenge/

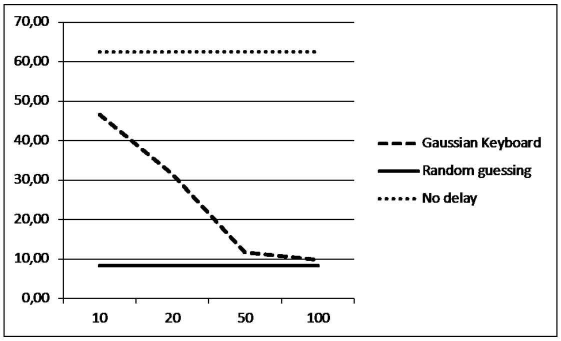

Fig. 2. Accuracy (in %) of person identification in case of Gaussian Keyboard as function of σ when $\mu = 100$ ms. The accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing are shown for comparison.

Both for regular and privacy-aware keyboards, we measured classification accuracy, i.e., the proportion of typing patterns for which the user was correctly recognized based on the dynamics of her typing.

Additionally to presenting results of 1NN classification, we examine the similarity of typing patterns as well. In particular we show the median, the 10th and 90th percentiles of DTW distances both in case of the original data, as well as in case of the data corrupted according to our privacy-aware keyboard models.

4.3 Experimental Results

Fig. 1 and Fig. 2 show the accuracy of person identification, i.e., the proportion of correctly identified users, for our privacy-aware keyboard models as function of maximal delay d_{max} (in case of the Uniformly Random Delay Keyboard) and σ (in case of the Gaussian Keyboard). For comparison, the accuracy in case of a regular keyboard (labeled as "No delay") as well as the accuracy of random guessing[5] are shown as well.

As expected, the accuracy of person identification decreases with increasing maximal delay in case of the Uniformly Random Delay Keyboard. Most importantly, already in case of a delay of 200ms, the accuracy is close to that of random guessing. Taking the speed of typing into account, the delay of 200ms seems to be acceptable for most of the users. Increasing σ in case of the Gaussian Keyboard has a similar effect.

[5] With *random guessing* we mean a naive classifier that works as follows: for each typing pattern x of the test data, is selects one of the users randomly (each user has an equal probability to be selected), and this randomly selected user, denoted as $y_x^{(rnd)}$, is the prediction of the classifier. That is: according to the "guess" of this naive classifier, the typing pattern x belongs to the randomly selected user $y_x^{(rnd)}$. As there are 12 users in our dataset, with a probability of 1/12 the randomly selected user will match the true user associated with the typing pattern, therefore, the accuracy of random guessing is 1/12.

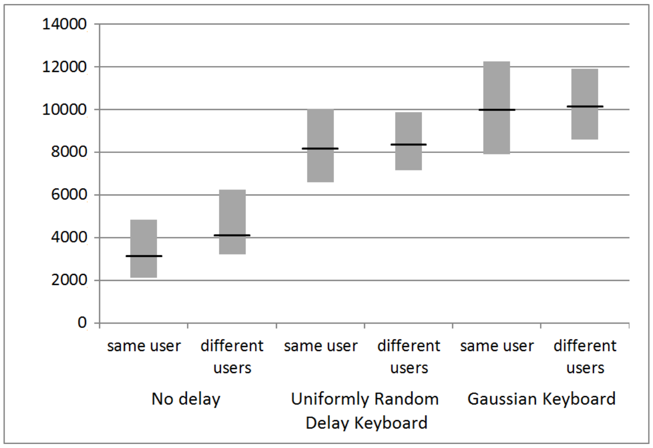

Fig. 3. Median, 10th percentile and 90th percentile of DTW-distances between typing patterns of the same user and different users in the following cases: regular keyboard (labeled as "No delay"), Uniformly Random Delay Keyboard and Gaussian Keyboard.

Fig. 3 shows the median, 10th percentile and 90th percentile of DTW-distances between typing patterns of the same user and different users in the following cases: (i) regular keyboard, (ii) Uniformly Random Delay Keyboard with $d_{max} = 200$ and Gaussian Keyboard with $\mu = 100$ and $\sigma = 100$. While the DTW-distances are generally larger in case of the privacy-aware keyboards, from the point of view of privacy-aware keyboards, the most important is that the range of distances between typing patterns of the *same* user almost perfectly overlap with the range of distances between the typing patterns of *different* users. This makes person identification difficult (if not impossible) for *any* model that is based on the distances (or similarities) between typing patterns.

5 Conclusions and Outlook

In this paper, we proposed the concept of privacy-aware keyboards, and we discussed two privacy-aware keyboard models. In our experiments, we simulated privacy-aware keyboards by adding noise to real-world keystroke dynamics data. The analysis shows that privacy-aware keyboards indeed make different users' typing patterns to appear much more similar to each other and therefore it becomes difficult for *any* similarity-based algorithm to distinguish users when privacy-aware keyboards are used.

We note that typing dynamics may be characterized by various features. For example, instead of measuring the duration of each keystroke (i.e., the time between pressing and releasing each key), one may measure the time between consecutive keystrokes (dwell times). In fact, we performed similar experiments in case of such data as well, and our observations are in accordance with the results reported in Section 4.3 for keystroke duration data.

Furthermore, it has to be pointed out that increasing usage of smartphones and tablets underline the importance of the protection of personal information. We argue that a combination of various techniques may be necessary: for example, if the user forbids an app to read unique identifiers of the device, the app may still try to identify the user based on heuristics, such as the keystroke dynamics, on which we focused in this paper.

While a detailed study of the realization of privacy-aware keyboards are out of scope of this paper, we note that most devices with Android and iOS systems have touch screens which measure pressure as well [1]. Therefore, on such systems, one should pay attention to add noise to the pressure information as well. Furthermore, when the privacy-aware keyboard is realized in a software, it is necessary that it captures keyboard events (or touch screen events, respectively) *before* any other application of the system. As this may be difficult to ensure, we believe that in cases of laptops and desktop computers, it may be more safe to realize a privacy-aware keyboard within the hardware, i.e., in the actual keyboard of the computer.

References

1. Antal, M., Szabó, L. Z., László, I.: Keystroke dynamics on Android platform. Procedia Technology 19, 820–826 (2015)
2. Monrose, F., Rubin, A.D.: Keystroke dynamics as a biometric for authentication. Future Generation Computer Systems 16(4), 351–359 (2000)
3. Doroz, R., Porwik, P., Safaverdi, H.: The New Multilayer Ensemble Classifier for Verifying Users Based on Keystroke Dynamics. Computational Collective Intelligence, Lecture Notes in Computer Science 9330, 598–605 (2015)
4. Buza, K., Neubrandt, D.: How You Type Is Who You Are. 11th IEEE Int'l Symposium on Applied Computational Intelligence and Informatics, 453–456 (2016)
5. Kozierkiewicz-Hetmanska, A., Marciniak, A., Pietranik, A.: Data Evolution Method in the Procedure of User Authentication Using Keystroke Dynamics. Computational Collective Intelligence, Lecture Notes in Computer Science 9875, 379–387 (2016)
6. Korolova, A., Kenthapadi, K., Mishra, N., Ntoulas, A.: Releasing search queries and clicks privately. Proceedings of the 18th International Conference on World Wide Web, 171–180 (2009)
7. Wong, F.W.M.H., Supian, A.S.M., Ismail, A.F., Kin, L.W., Soon, O.C.: Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm. 35th IEEE Asilomar Conference on Signals, Systems and Computers, vol. 2, 911–915 (2001)
8. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time-series data. International Journal of Computer Research, 10(3), 49–61 (2001)
9. Kim, S., Smyth, P., Luther S.: Modeling waveform shapes with random effects segmental Hidden Markov Models. In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, 309–316 (2004)
10. Woźniak, M., Jackowski, K.: Fusers Based on Classifier Response and Discriminant Function – Comparative Study. Lecture Notes in Computer Science 5271, 361–368 (2008)

11. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: A survey. Information Fusion 37, 132–156 (2017)
12. Buza, K., Nanopoulos, A., Horváth, T., Schmidt-Thieme, L.: GRAMOFON: General model-selection framework based on networks. Neurocomputing 75(1), 163-170 (2012)
13. Buza, K.: Fusion methods for time-series classification. Peter Lang Verlag (2011)
14. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in AI 5(4), 221-232 (2016)
15. Saez, J.A., Krawczyk, B., Wozniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recognition 57, 164-178 (2016)
16. Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., Herrera, F.: A survey on Data Preprocessing for Data Stream Mining: Current status and future directions. Neurocomputing (2017)
17. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. Proceedings of the 23rd ACM International Conference on Machine Learning, 1033–1040 (2006)
18. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment 1(2), 1542–1552 (2008)
19. Chen, G.H., Nikolov, S., Shah, D.: A latent source model for nonparametric time series classification. Advances in Neural Information Processing Systems 26, 1088–1096 (2013)
20. Devroye, L., Györfi, L., Lugosi, G.: A probabilistic theory of pattern recognition, Springer Science & Business Media (1996)