Shared Dataset on Natural Human-Computer Interaction to Support Continuous Authentication Research

Christopher Murphy, Jiaju Huang, Daqing Hou, and Stephanie Schuckers
Electrical and Computer Engineering Department
Clarkson University, Potsdam NY USA 13699
murphycd, jiajhua, dhou, sschucke@clarkson.edu

Abstract

Conventional one-stop authentication of a computer terminal takes place at a user's initial sign-on. In contrast, continuous authentication protects against the case where an intruder takes over an authenticated terminal or simply has access to sign-on credentials. Behavioral biometrics has had some success in providing continuous authentication without requiring additional hardware. However, further advancement requires benchmarking existing algorithms against large, shared datasets. To this end, we provide a novel large dataset that captures not only keystrokes, but also mouse events and active programs. Our dataset is collected using passive logging software to monitor user interactions with the mouse, keyboard, and software programs. Data was collected from 103 users in a completely uncontrolled, natural setting, over a time span of 2.5 years. We apply Gunetti & Picardi's algorithm, a state-of-the-art algorithm in free text keystroke dynamics, as an initial benchmark for the new dataset.

1. Introduction

Traditionally authentication of a computer terminal takes place at the initial sign-on, thus offering security protection at only a single point of time. This one-time authentication cannot protect against the situation where an intruder gains access to an already-authenticated terminal, or simply authenticates using someone else's sign-on credentials. Continuous authentication is able to mitigate this problem by detecting when there is a mismatch between the intended user and the current user. Since the most common actions performed by users when working with computers are keystrokes and mouse movements, research in continuous authentication has made use of this data and shown individuality for behavioral biometrics [16], including typing rhythms [3] and mouse movements [2].

While multiple algorithms [7, 11, 4, 15] and numerous studies [3] have been developed to explore continuous authentication using behavioral biometrics, a primary constraint to further advancement in this field is the availability of large, shared datasets, needed to enable independent algorithm benchmarking, particularly during unconstrained natural behavior. Up to date, published algorithms have been tested only on small datasets, some only local to individual researchers. As a result, accurate comparisons across different algorithms have not been possible and, thus, published performance remains questionable. To this end, in this paper, we aim to provide a large, publicly available dataset to enable replicable research.

Our keystroke dataset is captured in a completely uncontrolled setting. This natural setting contrasts the fixed-text captured in some datasets where the participants are required to type a certain passphrase [10, 6, 15, 14, 13]. Unlike other laboratory-based free-text datasets [15, 14], which are limited to a few hours of typing on prescribed tasks, ours is collected over a 2.5 year time span, when a participant is doing work of their choice, on their own computer, and in their own environment. Moreover, unlike the Torino free text dataset [7], ours does not restrict a participant to use an HTML form but allows for any application of their choice.

Our dataset also expands upon the number of data dimensions. In addition to keystrokes data, we also capture mouse movements, mouse clicks, and data on the software background and foreground processes. In particular, our ability to capture the active foreground process provides a unique context to the rest of the dataset, as keystrokes and mouse clicks can be temporally matched to a specific task being performed by the user. As a whole, the data collected by the logger represents a user's free and unrestricted interaction with their computer. It provides a more complete picture of a user's interaction with their computer, but without requiring additional monitoring hardware such as facial cameras or fingerprint sensors.

To our best knowledge, our dataset is the largest collection of keystrokes captured in an uncontrolled setting. In addition, it also contains data about mouse clicks, mouse

movements, and active programs. The combination of these makes our dataset a novel contribution. Our dataset will be made freely available upon request for research.

The remainder of this paper is organized as follows: Section 2 surveys related datasets in literature. Section 3 describes our data collection process and design. In Section 4, we perform a preliminary comparison between our dataset and three others, using Gunetti & Picardi's keystroke dynamics algorithm [7]. Lastly, Section 5 concludes the paper.

2. Related Work

Table 1 depicts a set of keystroke datasets published in literature. Note that we have included only one fixed text dataset [12] as example. This is because our focus in this paper is on free text datasets. Including ours, four of these datasets are publicly available.

The Torino dataset collected by Gunetti & Picardi [7] is in Italian. Participants are asked to open an HTML form in their browsers to fill in whatever they feel comfortable to type. Each session consists of about 800 keystrokes, and a participant may type no more than one session each day. The dataset has 400K keystrokes in total, of 40 participants. The data set also contains 165 users who have contributed only one session, to be used as imposter attacks. This dataset has only key press time, but without key release time. So it is impossible to calculate the dwell time for individual keys.

The Clarkson dataset [15] contains two sessions for each of 39 participants. In each session, the participants are invited to log their keystrokes in a lab setting where they all use the same desktop computer to conduct a set of prescribed tasks by running the same browser. The tasks create mostly free text keystrokes by answering survey questions that are carefully designed around the interest areas of the subject population to ensure fluent response. The dataset contains 840K keystrokes in total, of 39 participants [15].

The Buffalo dataset [14] contains a mix of (long) fixed text and free text keystrokes. The keystrokes are also collected in a laboratory. To study the effect of different keyboards, participants are required to use different kinds of keyboards during different sessions. Four different kinds of keyboards are used. There are 148 participants in this dataset, with a total of 2.14M keystrokes. Most users have contributed 3 sessions of data. The average user has 14.4K keystrokes, with the maximum of 18.3K.

Our dataset, unlike the three above, is collected when the participants work in a completely uncontrolled and natural setting. The logger is loaded onto the user's laptop or on any computer in an open student lab. The logger passively records participants' keystrokes without requiring or restricting them to do anything prescribed. The keystrokes are created when the participants work on tasks of their own choice. There are 103 participants in this dataset, with a total of 12.9M keystrokes. To our knowledge, our dataset is the largest, where an average user has 125K keystrokes, with the maximum of 625K.

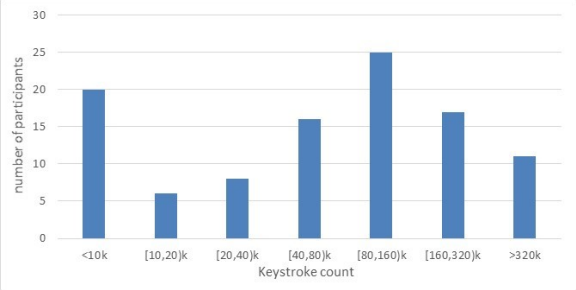


Figure 1: Distribution of total keystrokes per user over the 103 volunteer users. The average participant contributes 125 thousand keystrokes.

3. Data Collection

Our data collection utilizes a logger that is a stand-alone executable developed in C# for running on Windows machines. Once installed on a participant's computer, the logger passively records and sends keystroke, mouse, and process events to a remote database server. Over a course of 2.5 years, 103 participants have volunteered to use the logger for collecting keystroke, mouse, and application data. Since our study involved human subjects, we were required to thoroughly document our data collection process and seek formal approval. We received formal approval before data collection was launched. Monetary incentives were awarded to participants when their amounts of contributed keystrokes reached set milestones.

Figure 1 depicts a histogram for the amounts of keystrokes contributed by the 103 participants. The average participant contributes 125 thousand keystrokes, with the maximum at 625 thousands. However, the variation between participants is also fairly large. Our dataset is freely available to researchers upon request.

3.1. Logger

The logger executable is a Windows application written in C# to capture events via the Windows API. On startup of the application, users must acknowledge that data will now be collected as shown in Figure 2a. The logger runs in the background and contains a minimalist user interface, as shown in Figure 2b, which is accessible through the application's icon in the system tray. The logging operation may be paused indefinitely by the user via the interface menu, which is useful for entering sensitive data such as banking information or important passwords. Once paused, logging will not resume until the user requests to do so from the interaction menu. There is also an option to pause logging for 10 minutes after which the logger will resume automatically. At the discretion of the user, the application can be

Study	#User	Time Span	Collection Setting	#Keystrokes	Data Availability
Dowland and Furnell [5]	35	3 months	Uncontrolled	3.4 M	NO
Gunetti and Picardi [7]	40+165	6 months	Browser, anywhere	400 K	YES
Messerman et al. [11]	55	12 months	Predefined tasks	293 K	NO
Monaco et al. [12]	30	1-3 sessions	Fixed text	280 K	NO
Ahmed and Traore [1]	53	5 months	Uncontrolled	9.5 M	NO
Vural et al. [15]	39	2 sessions	Predefined tasks, in lab, browser	840 K	YES
Sun, Ceker, Upadhyaya [14]	148	1-3 sessions	Predefined tasks, in lab, desktop, 4 keyboards	2.14 M	YES
Our dataset	103	2.5 years	Uncontrolled	12.9 M	YES

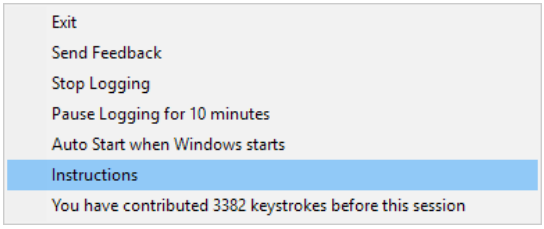
Table 1: Comparison of keystroke datasets in literature, in order of years of publication.

made to start automatically when the computer is booted up. In any case, users must acknowledge the startup prompt before data is collected.

Events are timestamped using the `System.DateTime.Now` property, from which we obtain the number of ticks (100-nanosecond intervals) of elapsed time since 12:00:00 midnight, January 1, 0001. When working with closely-spaced timings, it is important to note that the property's value is only as accurate as the system clock tick which has a resolution of approximately $10 - 16$ milliseconds. Event data captured by the program is streamed to a remote MySQL database for later analysis.

In this study, the idea of a unique user is tied to the physical machine they are using. Users were encouraged to remain with the same computer or they would be counted as an additional participant. Finally, a user's progress in terms of the number of keystrokes contributed is listed in the interaction menu. Participants who contribute 100,000 keystrokes or more were compensated for their participation in the study.

(a) Startup Prompt



(b) Interaction Menu

Figure 2: Users must acknowledge the startup prompt in (a) before any logging takes place. While running, the logger's interaction menu, pictured in (b), is accessible through its icon in the system tray. Notable options include pause and resume functions as well as an indication of how many keystrokes have been contributed.

3.2. Data Composition

Our data consists of three categories of temporal data: keystrokes (Table 2), mouse clicks and mouse coordinates (Table 3), and running processes, including process creation and termination, and the active (focused) process (Table 4).

Table 2 shows a few rows of keystroke data. Timings are recorded whenever a user presses or releases a key. Using this timing data, the dwell time of individual keys and duration of digraphs can be calculated for use in authentication.

As depicted in Table 3, for mouse input, the current position of the mouse cursor is recorded for every mouse movement event. Our logger also captures each left and right mouse button click via the Windows API. Mouse movement entries include virtual screen coordinates which contain the position of the mouse according to Windows operating system rather than corresponding to physical screen coordinates. Further, there is no distinction between pressing and releasing mouse buttons as the mouse click event captured by the logger is at the initial time of the button press.

Table 4 depicts the third type of event, running processes, which can be subdivided into three categories: process creation, process termination, and the name of the active system process (if any) resulting from the last left mouse click. Data on running processes gives additional context to a user's keystroke and mouse data. The "Info" column contains several key pieces of information delimited by a vertical bar. These begin with the event type, which may read "Process Terminated," "New Process," or "Current active program," followed by a description.

For the Current active program, the description is the window title, from which the application names can be ex-

User ID	Time Stamp (ticks)	Action Type	Key Name
4302075	636172286538589004 (2016-12-13 12:24:13)	'KeyDown'	'A'
4302075	636172286539669002 (2016-12-13 12:24:13)	'KeyDown'	'Space'
4302075	636172286541684820 (2016-12-13 12:24:14)	'KeyUp'	'A'

Table 2: Keystroke data includes separate events for key press and release. The sample data shows the "A" key being pressed for approximately 310 milliseconds, calculated by taking the difference in 100-nanosecond ticks from each event.

User ID	Time Stamp	Info
4302075	636172286538589004 (2016-12-13 12:24:13)	Click event: Left
4302075	636172286539669002 (2016-12-13 12:24:13)	Click event: Right
3461752	636275108577494505 (2017-04-11 12:34:17)	Location: (817, 375)
3461752	636275108577564436 (2017-04-11 12:34:17)	Location: (816, 375)

Table 3: Mouse events include button click and location events. Click events record left and right button presses while location events record when the mouse is moved at least one virtual screen coordinate. The location of any click may be inferred from the most recent location event.

tracted. For process creation and termination events, the description includes the following parameters: name of the executable, process handle, priority, session ID, path, and command-line call. Of these parameters, the process handle is the only unique yet arbitrary value maintained by the operating system for the duration of the process lifecycle. Note that the active system process, or focused window, is only checked at the time of a left mouse click. Occasionally, a window may request focus from the operating system on its own, or users may focus a window via hotkey rather than by a left mouse click. We believe that the method of focusing a window via a left mouse click is by far the most common one and thus capturing it would provide sufficient data needed for research.

3.3. Data sanitation

Since our data is collected from a participant's natural interaction with a computer, as opposed to artificial, prescribed tasks, the data must be sanitized to remove accidental private or sensitive information, before released to other researchers. We have taken several measures to accomplish this. First, as described in Section 3.1, our logger is designed to allow a participant to stop logging whenever they believe that they are going to type sensitive data. Our participants are advised to pause the logger when they need the keystrokes to go unlogged. If they unfortunately forget to do so, they can request their keystrokes over a certain period of time to be deleted. Second, we assign randomized IDs to the participants, and our database only record these IDs. Third, we have utilized a set of regular expression patterns to detect and remove from our data a set of common sensitive information such as bank card numbers, social security numbers, and phone numbers. Lastly, we performed manual checking on a random subset of participants' data to ensure that the remaining dataset is 'clean.'

4. Initial Evaluation

To gain more confidence in our dataset, we performed an initial evaluation by applying Gunetti & Picardi's algorithm [7], a state-of-the-art for free text keystroke dynamics [11, 1]. As comparison, we also apply the same algorithm to three other public datasets: the Torino dataset [7], the Clarkson dataset [15], and the Buffalo dataset [14].

Informed by prior studies [7, 11, 8], we use 1,000 consecutive keystrokes to form a sample. Specifically, each user's data is divided into samples of 1,000 keystrokes. The reference profile contains the first 10 samples in a user's data, i.e., 10,000 keystroke are used to form the user profile. The remaining samples are used as genuine test samples. Each user uses samples from all other users as imposter attacks. Performance metrics (FFR: False Reject Rate; FAR: False Accept Rate) are calculated for the users and reported in the form of DET (Detection Error Tradeoff).

4.1. Gunetti & Picardi's algorithm

We reimplement Gunetti & Picardi's verification algorithm [7]. Briefly, given a test sample x and a user A, the algorithm verifies whether X comes from A, as follows:

User ID	Time stamp	Info
1449549	635894165235792504 (2016-01-26:14:48:43)	New Process | chrome.exe | 5712 | Normal | 1 | *Execution Path* | *Command Line*
1449549	635894172341470811 (2016-01-26 15:00:34)	Current active program | Google - Google Chrome
1449549	635894176710573820 (2016-01-26 15:07:51)	Process Terminated | chrome.exe | 11864 | Normal | 1 | *Execution Path* | *Command Line*

Table 4: Three types of process-related events: New Processes, Current Active (Focused) Programs, and Process Termination. Process creation and termination events also list the name of the executable, unique process handle, priority, session ID, execution path, and command line call. The Current Active Program captures the title of the associated window. The Execution Path and Command Line fields have been omitted from the above sample data.

1. For each legal user U in the system, calculate two metrics $m(U)$ and $md(U, X)$ defined as follows: $m(U)$: the average distance between all the samples in Us profile. $md(U, X)$: the average distance between all the samples in Us profile and the test sample X.

2. If there exists any other legal user B of the system, to whom the test sample X is closer than to the verified user A, that is, $md(B, X) <= md(A, X)$, then reject X for user A. Otherwise, X is closest to A in the system; conduct a further test in the next step to ensure that X is sufficiently close to As profile samples.

3. Furthermore, if test sample X is closer to the core of user As profile samples than the average distance between themselves ($m(A)$). That is, if $md(A, X) <= m(A)$ is true, then accept X. Otherwise, $md(A, X) > m(A)$ holds, and check the next condition.

4. Finally, accept X, if $md(A, X)$ is closer to $m(A)$ than to any other $md(B, X)$ computed by the system, i.e., $md(A, X) - m(A) < md(B, X) - md(A, X)$.

The distance between typing samples S1 and S2 is measured in terms of n-graphs (n consecutive keystrokes typed by a user), using a measurement called A measure (for "*A*bsolute": using the difference in average ngraph latencies to determine similarity) and R measure (for "*R*elative": using the relative ordering of average ngraph latencies to determine similarity) .

The A measure between $S1$ and $S2$ is defined in terms of the n-graphs they share and a constant value t:

$A_{t,n}(S1, S2) = 1 - \#similar/\#shared$

where $\#similar$ represents the number of similar n-graphs shared by $S1$ and $S2$, and $\#shared$ the total number of n-graphs shared.

The constant t is needed for defining n-graph similarity. Specifically, let $G_{S1,d1}$ and $G_{S2,d2}$ be the same n-graph occurring in typing samples $S1$ and $S2$, with duration $d1$ and $d2$, respectively. We say that $G_{S1,d1}$ and $G_{S2,d2}$ are similar if $1 <= max(d1, d2)/min(d1, d2) <= t$, where t is a constant greater than 1 (we used $t = 1.25$).

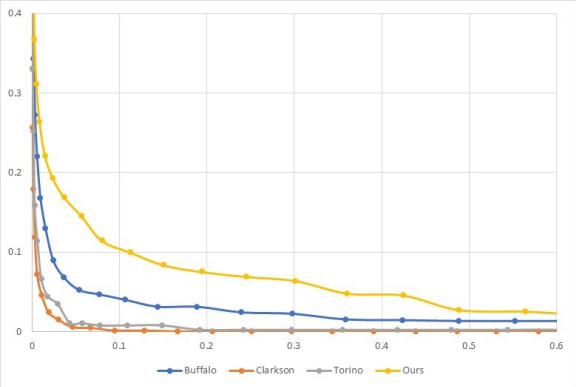


Figure 3: DET curves of benchmarking Gunetti & Picardi's algorithm against four public datasets (Torino dataset [7]; EER: 3.11%, Clarkson dataset [15]; EER: 2.17%, Buffalo dataset [14]; EER: 3.75%, and our dataset; EER: 10.36%, in temporal order of dataset creation).

4.2. Performance Result

We apply Gunetti & Picardi's algorithm [7] to our new dataset. To highlight the value of having multiple datasets, we also apply the same algorithm to three other public free text datasets: the Torino dataset [7], the Clarkson dataset [15], and the Buffalo dataset [14]. Figure 3 depicts the four DET (Detection Error Tradeoff) curves [1] obtained from our experiment.

As shown in Figure 3, the EER (Equal Error Rate) for Torino, Clarkson, Buffalo, and our own dataset are 3.11%, 2.17%, 3.75%, and 10.36%, respectively. Most importantly, we can see that our dataset produces a much worse EER (10.36%) than the other three datasets. The Clarkson dataset has the best performance (2.17%), a little better than the second best, the Torino dataset (3.11%). The Buffalo dataset performs the third best (3.75%), but it is much closer to the first two than to our dataset.

These different performance can be explained in terms

[1] https://en.wikipedia.org/wiki/Detection_error_tradeoff Last accessed: 04/30/2017.

of the differences in the design of data collection protocols. The Clarkson dataset [15] is the most constrained: all participants must perform the same set of prescribed tasks, using the same HTML form, and on the same desktop computer within a university laboratory. The Buffalo dataset [14] adopts a similar design as Clarkson except that they purposefully introduce four different kinds of keyboards. In addition, unlike Clarkson's, their logger is a desktop application instead of a browser based form. So the difference in performance between the Clarkson and Buffalo datasets can be attributed to the different keyboards as well as the desktop-based logger used in the Buffalo dataset.

Like Clarkson, the Torino dataset [7] is also collected using a fixed HTML form, but participants are otherwise allowed to use any keyboard and work in their own natural environment rather than a laboratory. These may explain why the Clarkson dataset has a better EER than the Torino.

While the other three datasets are all controlled but to different degrees, our dataset is produced in a completely uncontrolled setting. Each participant contributes their data while using different computers and keyboards, different computer applications, such as office software, browser, and programming tools, for tasks of their own choice, and in their own natural environments. For example, some of our participants have typed a large amount of 'noisy' keystrokes, such as when playing computer games [9]. As shown in Figure 3, the performance of the algorithm degrades significantly when applied to such uncontrolled data.

5. Conclusion

Advancement in the field of continuous authentication necessitates the creation of shared datasets to enable replicable research. In this paper, we contribute a novel large dataset that combines free text keystrokes, mouse events, and applications. Unlike other free text keystroke datasets that are collected in a laboratory or otherwise restricted setting, our dataset is from completely uncontrolled, natural settings. Our initial evaluation shows that authentication performance degrades significantly when the data collection becomes less controlled.

Acknowledgment

This work was supported by NSF Award CNS-1314792.

References

[1] A. A. Ahmed and I. Traore. Biometric recognition based on free-text keystroke dynamics. *IEEE Transactions on Cybernetics*, 44(4):458–472, April 2014.

[2] A. A. E. Ahmed and I. Traore. A new biometric technology based on mouse dynamics. *IEEE Transactions on Dependable and Secure Computing*, 4:165–179, 2007.

[3] S. P. Banerjee and D. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1), 2012.

[4] H. Ceker and S. Upadhyaya. User authentication with keystroke dynamics in long-text data. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6, Sept 2016.

[5] P. S. Dowland and S. M. Furnell. A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In Y. Deswarte, F. Cuppens, S. Jajodia, and L. Wang, editors, *Security and Protection in Information Processing Systems: IFIP 18th World Computer Congress TC11 19th International Information Security Conference, Toulouse, France*, pages 275–289. Springer US, 2004.

[6] R. Giot, B. Dorizzi, and C. Rosenberger. Analysis of template update strategies for keystroke dynamics. In *2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, pages 21–28, April 2011.

[7] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, Aug. 2005.

[8] J. Huang, D. Hou, S. Schuckers, and Z. Hou. Effect of data size on performance of free-text keystroke authentication. In *Identity, Security and Behavior Analysis (ISBA), 2015 IEEE International Conference on*, pages 1–7, March 2015.

[9] J. Huang, D. Hou, S. Schuckers, and S. J. Upadhyaya. Effects of text filtering on authentication performance of keystroke biometrics. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Dec 2016.

[10] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, pages 125–134, June 2009.

[11] A. Messerman, T. Mustafi, S. A. Camtepe, and S. Albayrak. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–8, Oct 2011.

[12] J. V. Monaco, N. Bakelman, S. H. Cha, and C. C. Tappert. Developing a keystroke biometric system for continual authentication of computer users. In *European Intelligence and Security Informatics Conference (EISIC12)*, pages 210–216, 2012.

[13] A. Morales, J. Fierrez, R. Tolosana, J. Ortega-Garcia, J. Galbally, M. Gomez-Barrero, A. Anjos, and S. Marcel. Keystroke Biometrics Ongoing Competition. *IEEE Access*, 4:7736–7746, 2016.

[14] Y. Sun, H. Ceker, and S. Upadhyaya. Shared keystroke dataset for continuous authentication. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Dec 2016.

[15] E. Vural, J. Huang, D. Hou, and S. Schuckers. Shared research dataset to support development of keystroke authentication. In *International Joint Conference on Biometrics (IJCB), 2014 IEEE International Conference on*, 2014.

[16] R. V. Yampolskiy and V. Govindaraju. Behavioural biometrics: a survey and classification. *Int. J. Biometrics*, 1(1):81–113, June 2008.