Removing Personally Identifiable Information from Shared Dataset for Keystroke Authentication Research

Jiaju Huang
Clarkson University
Potsdam, NY, USA 13699
jiajhua@clarkson.edu

Bryan Klee
Clarkson University
Potsdam, NY, USA 13699
kleebm@clarkson.edu

Daniel Schuckers
St. Lawrence University
Canton, NY, USA 13617
dpschuckers@gmail.com

Daqing Hou
Clarkson University
Potsdam, NY, USA 13699
dhou@clarkson.edu

Stephanie Schuckers
Clarkson University
Potsdam, NY, USA 13699
sschucke@clarkson.edu

Abstract

Research on keystroke dynamics has the good potential to offer continuous authentication that complements conventional authentication methods in combating insider threats and identity theft before more harm can be done to the genuine users. Unfortunately, the large amount of data required by free-text keystroke authentication often contain personally identifiable information, or PII, and personally sensitive information, such as a user's first name and last name, username and password for an account, bank card numbers, and social security numbers. As a result, there are privacy risks associated with keystroke data that must be mitigated before they are shared with other researchers. We conduct a systematic study to remove PII's from a recent large keystroke dataset. We find substantial amounts of PII's from the dataset, including names, usernames and passwords, social security numbers, and bank card numbers, which, if leaked, may lead to various harms to the user, including personal embarrassment, blackmails, financial loss, and identity theft. We thoroughly evaluate the effectiveness of our detection program for each kind of PII. We demonstrate that our PII detection program can achieve near perfect recall at the expense of losing some useful information (lower precision). Finally, we demonstrate that the removal of PII's from the original dataset has only negligible impact on the detection error tradeoff of the free-text authentication algorithm by Gunetti and Picardi. We hope that this experience report will be useful in informing the design of privacy removal in future keystroke dynamics based user authentication systems.

2019 IEEE 5^{th} International Conference on Identity, Security, and Behavior Analysis (ISBA)
978-1-7281-0532-1/19/$31.00 2019 IEEE

1. Introduction

Research on keystroke authentication has become increasingly active in the last decade or so [21] [8] [9] [10]. This line of research has the good potential to offer continuous authentication to complement conventional authentication methods that typically occur only once at the initial log-in time. Such continuous authentication are attractive because they can be effective in capturing insider threats and identity theft earlier, before more harm is done to the genuine users.

Keystroke authentication (free text, as opposed to fixed text) are commonly based on patterns of the timing between successive keystrokes, and the amount of time a key is pressed [21] [8] [9] [10]. As such, a large amount of keystroke data are often needed for this method to work [21] [8] [9] [10]. When keystroke data is collected through key loggers, the personally identifiable information and personally sensitive information such as the user's first and last names, username and password, bank card numbers, and social security numbers, will be recorded as well. As shown in this study, the large amount of keystroke data increases the likelihood of the presence of PII's. If a hacker gains access to such keystroke data, he/she might potentially be able to cause various damages such as personal embarrassments, blackmails, financial loss, and identity theft [12].

Unfortunately, the privacy of the keystroke data has not been investigated much in the literature [19]. Meanwhile, citizens and governments in modern society are increas-

ingly concerned about the protection of personal data, or PII [1]. For example, EU's General Data Protection Regulation (GDPR) has become effective since May 25, 2018 [5], further highlighting the importance of this problem. Therefore, it has become pressing to learn how to effectively remove sensitive PII's from keystroke data.

To this end, in this paper, we conduct a systematic study of removing PII's from a recent large keystroke dataset collected by Murphy *et al*. at Clarkson University [16]. We start off with identifying a list of PII's based on an analysis of their potential harms to the subjects. We implement sophisticated strategies, based on regular expressions, to detect and remove PII's from the natural language text in the dataset. To err on the side of caution, we opt to filter out all PII's in the expense of losing an affordable amount of useful keystroke information. Finally, we thoroughly evaluate the effectiveness of our detection program by comparing its output against a manually built ground truth and calculating precision and recall. We demonstrate that it has achieved near perfect recalls. Given the importance of such a large free keystroke dataset in advancing keystroke authentication research, we believe that our study represents an important contribution in mitigating the privacy risks associated with sharing such data with the research community.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 describes the data sanitization process. In Section 4, we evaluate the precision and recall of our PII detection program and demonstrate its effectiveness. Finally, Section 5 concludes the paper.

2. Related Work

Table 1 depicts chronologically eleven recent free text keystroke datasets in the literature. Among these, only four are publicly available for research, with the Clarkson dataset [16] being the largest in terms of the amount of keystrokes and the time span of data collection. That is why we choose it for this study. Unfortunately, there has been very little investigation of the privacy issues associated with keystroke dynamics datasets except that of Sun and Upadhyaya [19]. However, different from their study, this work involves a much larger number of subjects (103 versus 15) and more accurate pattern matching capabilities than those of Sun and Upadhyaya [19]. For example, all of Sun and Upadhyaya's PII matching rules require the presence of a certain special word before the PII, such as 'mail', 'password', and 'user' for an email address, a password, and a username, respectively. While this might be a reasonable assumption in their particular setting, it does not hold for cases such as the Clarkson dataset [16] where the data is completely natural language text. As a result, our detection strategies are designed to be more general and powerful than theirs, capable of handling even practical issues such as typos and auto-completion.

Aggregation of secondary information can be used to identify individuals with surprisingly high probability of success. It has been shown previously that a large portion of the US population can be re-identified using a combination of 5-digit ZIP code, gender, and date of birth [20]. This has motivated us to remove zip code, gender, and birthdates from the studied dataset.

Gross and Acquisti research about Facebook users' behavior in handling online privacy issues and their implications [7] [2]. Acquisti *et al*. present a survey of privacy and human behavior in the age of information [1].

3. Data Sanitization

3.1. The Dataset

We are able to analyze a copy of the Clarkson keystroke dataset [16] and extract and assess the impact of disclosing privacy revealing information. The dataset contains nearly 13 million keystrokes from 103 subjects. Of the 103 subjects in the Clarkson dataset, 62 are males and 41 females; 89 are undergraduate students, 6 graduate students, 5 staff, and 3 faculty. The average and maximum numbers of keystrokes per subject are 125K and 625K, respectively. The dataset is collected when the participants work in a completely uncontrolled and natural setting. A logger is loaded onto a user's laptop or on any computer in an open student lab. The logger passively records a participant's keystrokes without requiring or restricting them to do anything prescribed. The keystrokes are created when the participants work on tasks of their own choice. The average and maximum numbers of days when a subject contributes their keystrokes are 30 days and 2.5 years, respectively. So the dataset is sufficiently large to conduct a meaningful study of privacy and PII.

3.2. Process

Three coauthors are involved as a leader, a tester, and an implementer to detect and remove PII's from the dataset. The leader ensures that the team agrees upon a list of PII's to focus on as well as strategies to search for them in the dataset. The targeted sensitive data includes a participant's first and last names, phone numbers, addresses, campus ID, emails, usernames and passwords, social security numbers, and bank card numbers. Examples of each are shown in Table 2 along with its potential harm. Note that the first SSN of John David Sweeney, Jr., of Westchester County, New York, is used as an example for Social Security Number [2]. Overall, removing these PII's will free the subjects from potential harms such as personal embarrassment, identity theft, financial loss, or potential blackmails [12].

[1] https://en.wikipedia.org/wiki/Personally_identifiable_information: July 6, 2018

[2] https://www.wikitree.com/wiki/Sweeney-1072

Study	#User	Time Span	Logger Setting	#Keystrokes	Availability
Monrose and Rubin [14]	31	7 weeks	UI, set tasks	N/A	NO
Monrose and Rubin [15]	63	11 months	UI, set tasks	N/A	NO
Dowland and Furnell [4]	35	3 months	Uncontrolled, logger	3.4 M	NO
Gunetti and Picardi [8]	40	6 months	Web UI	400 K	YES
Messerman *et al*. [13]	55	12 months	Web UI (email)	293 K	NO
Ahmed and Traore [3]	53	5 months	Uncontrolled, logger	9.5 M	NO
Janakiraman and Sim [11]	22	2 weeks	Uncontrolled, logger	9.5 M	NO
Stewart *et al*. [17]	30	4 tests in 3 weeks	Web UI, short answers	720 K	NO
Vural *et al*. [22]	39	2 sessions	Web UI, tasks	840 K	YES
Sun *et al*. [18]	148	28 days	Web UI, tasks	2.14 M	YES
Murphy *et al*. [16]	103	2.5 years	Uncontrolled, logger	12.9 M	YES

Table 1. A set of free-text keystroke datasets in the literature. There are additional 165 imposters in Gunetti and Picardi's dataset [8].

PII	Potential Harm If Not Removed	Examples
Name	Directly identifying a person	Warren Buffett
Phone	linked to a person	AAA8540644 / AAA-244-7116 / (AAA)343-0960
Home Address	linked to a person; needed to verify identity by third parties, identity theft	46 lake street, city, state, zip / 530 south state street / 101 old falls street
Campus ID	linked to a person	0XXXXX
Email Address	linked to a person; impact level varied, but since they are commonly being used as usernames for important accounts, harm can be serious	jim.carlock58@gmail.com / laura1610@twcny.rr.com / steveqzx@yahoo.com
Username & Password	linkable; impact level varied from low to high	sorat1* / Hk14Cu18ae* / fireball1*
Social Security Number	linked to a person; identity theft	055-09-0001, John David Sweeney, Jr., of Westchester County, New York
Bank Card	linkable; potentially financial loss	4847745571680 / 5515001450003000 / 4485386955130025

Table 2. PII, potential harm if not removed, and examples. Examples are altered to prevent identification of the true subjects in the dataset.

Based on these, our tester reads through the text of the entire dataset and manually marks up all the PII's from it. For example, due to unintelligible handwriting in a subject's paper record, the tester is able to confirm some user's names by recognizing their signatures in emails. This process takes the tester about 200 hours of work. The implementer implements a PII detection program in Python utilizing its regular expression facilities. The marked up PII's are used to test the implemented Python PII detection program, where the tester manually identifies all the false positives and false negatives from the output of the detection program.

The detection and removal of PII's is implemented in Python using regular expressions, with an overall process as follows. For each user's keystroke data, we first look for and remove patterns of campus username and names from each user's keystroke data. We look for possible passwords by finding common substrings that follow email addresses, Clarkson usernames, and URL's for common social networks, including Google.com, Facebook.com, and Twitter.com, but see Section 3.4 for more details about this step. After removing usernames and passwords, we sequentially look for and remove bank cards, phone numbers, social security numbers, street addresses, and campus ID's from the user's data. Lastly, we filter out special words that can be used to identify specific individuals within an organization, such as a department chair or someone with a special title within a certain office.

The data structures that the detection program works on involve two arrays of the same length that represent each user's keystroke data. The first array stores the pressed key names and their associated timestamps. The second stores just the keys. The first array is suitable for representing the keystroke dynamics data for authentication, while the second suitable for pattern matching by Python. A match in the second array is identified by a pair of indices, from which a start time and an end time of the match is obtained from the first array as a timestamp pair. Once all PII's have been identified, we remove in one pass from the raw keystroke

data all the keystrokes that fall within the range of any timestamp pair of that user. We then store the remaining content of the raw data into a new file for sharing.

3.3. PII Detection

The convention for writing phone numbers varies among different countries. The number of digits, grouping, and separators may all differ [3]. Therefore, we consider only the case of United States. A USA phone number starts with an optional 1, followed by an area code of three digits, and then two more groups of digits of length 3 and 4, respectively. To improve accuracy, we also require that the area code must match one of the existing US area codes, which can be found online [4].

```
phone_number = re.compile('\D( (?:\+\d{1,2}\s)? \(? (?:\%s) \)? -? \d{3} -? d{4,} )(?:$|\D)'\%'|'.join(area_code))
#variable area_code contains all US phone area codes
```

For a street address, we look for any string that starts with a number, and followed shortly afterwards by words or abbreviations that indicate a street, such as road, Rd, avenue, Ave, street, St, lane, Lane, and drive. After that, we look for an optional state abbreviation and zip code.

```
street = re.compile('([0-9]+\s.{2,15} (?:avenue|Avenue|Ave\.? |street|Street|St \.? |st \.? |road|Road|rd|Rd|way|Way|lane|Lane|pkwy|Pkwy|Drive)(?:.{1,20} (?:NY|Ny|ny|NC|Nc|nc|VT|Vt|vt)? \d{5}(?:$|\D))?) ')
```

The format for an email address is "username@domain.name." To make matching more accurate, we only look for common domains, including .com, .edu, .gov, .org, .net and .mil. Since email addresses are also often used as usernames, followed by passwords, to log in to important sites such as moodle, peoplesoft, bank accounts, and social networks such as Facebook, we remove both such email addresses and the passwords that follow them (see Section 3.4 for details). Lastly, due to the existence of typos and auto-completion, we also regard a string such as "abc2clarkson.edu", "abc@gmail", as an email address, as long as it contains a "@" sign or certain website domains, including "clarkson", "gmail", "hotmail", and "yahoo".

```
email =re.compile('([a-zA-Z0-9_\.\^]{2,63}@(?:[a-zA-Z0-9-\^]+ (?:\.[a-zA-Z0-9-\^]+)*? \.(?:com|edu|gov|org|net|mil)|clarkson|gmail|hotmail|yahoo|aol))')
```

A Social Security Number (SSN) consists of 9 digits, optionally separated by a "-" into three groups, as XXX-XX-XXXX. Some restrictions apply to SSN: the first group cannot be 666, 000, or start with a 9; the second group cannot be 00, and the last group cannot be 0000 [5]. Note that in the Python code below, (?!(?:000|666|9))\d{3} specifies the first group of SSN. The "?!" means that a match of a digital string is possible only if the string does not begin with 000, 666, or 9. The "?:" inside a pair of parentheses means that the pattern defined by the pair is non-capturing. Lastly, "\d{3}" means 3 digits.

```
ssn = re.compile('(?:^|\D)((?!(?:000|666|9))\d{3}-?(?!00)\d{2}-?(?!0000)\d{4})(?:$|\D)')
```

Formal definitions for the format of bank card numbers can be found online [6]. Each payment card organization requires a different format of card numbers. For example, as shown below, a Visa card starts with a digit 4 followed by 12 or 15 digits, and a MasterCard starts with 51, 52, 53, 54, 55, 222100-272099 and contains 16 digits. We support the format of Visa, Master, Discover, American Express, JCB, Maestro, and DCI. These seven kinds of cards cover the vast majority of the cases, and are adequate as far as USA is concerned. As shown in the last line below, a string of digits must be surrounded by a pair of non-digit letters and match one of the 7 card formats to be recognized as a bank card number.

```
visa = ''4 [0-9]{12} (?:[0-9]{3})?''
master = ''(?:5[1-5] | 2[2-7]) [0-9]{14}''
discover = ''6 (?:011|5[0-9][0-9]|4[4-9][0-9]) [0-9]{12}''
ae = ''3[47] [0-9]{13}''
jcb = ''(?:2131|1800|35\d{3}) \d{11}''
maestro = ''(?:5[06-8]|6[0-9]) \d{14}''
dci = ''3 (?:0[0-5]|[68][0-9]) [0-9]{11}''
bank_card = re.compile('\D(%s|%s|%s|%s|%s|%s)(?:\D|$)'
%(visa, master, discover, ae, dci, maestro, jcb))
```

Lastly, we filter out a participant's first and last names. We gather these from the participant's records. When the paper record for a participant is incomplete or with unintelligible handwriting, we discover their names by reading their keystrokes. We also look for any Clarkson campus ID, which consists of 7 digits starting with a '0'.

3.4. Removing Usernames and Passwords

We look for possible combinations of usernames and passwords by finding common substrings that follow email addresses, Clarkson usernames, and URL's of common social networks such as Google.com, Facebook.com, and Twitter.com. If a string follows them often enough (for example, at least appearing twice and more than 10% of the time), we regard it as a possible password. We then search

[3] https://en.wikipedia.org/wiki/National_conventions_for_writing_telephone_numbers

[4] https://www.allareacodes.com/area_code_listings_by_state.htm

[5] https://en.wikipedia.org/wiki/Social_Security_number

[6] https://en.wikipedia.org/wiki/Payment_card_number

	#Users	#Instances	Min by user	Max by user	Average	#keys
Name	75	3,765	1	375	50.2	23,077
Phone Number	42	146	1	20	3.5	1,700
Address	38	149	1	15	3.9	3,029
Campus ID	34	65	1	7	1.9	556
Email	99	2,130	1	107	21.5	49,353
Username	77	1,445	1	120	18.8	9,450
Password	73	4,014	1	207	55.0	41,498
SSN	45	134	1	12	3.0	1,380
Bank Card	11	17	1	4	1.6	290

Table 3. Statistics of detected PII per user

	#Instances	#False Positives	#False Negatives	Precision	Recall
Name	3,765	0	22	100%	99.4%
Phone Number	146	24	0	83.4%	100%
Address	149	5	1	96.6%	99.3%
Campus ID	65	-	0	-	100%
Email	2,130	0	0	100%	100%
Username	1,445	0	0	100%	100%
Password	4,014	68	12	98.3%	99.7%
SSN	134	-	0	-	100%
Bank Card	17	-	0	-	100%

Table 4. Precision and recall of PII detection and removal

the entire data file to remove all occurrences of such passwords as well as the associated email address, usernames, and URL's. Email addresses are considered because nowadays it is a common practice to use email addresses as usernames for a web account. Clarkson usernames are considered because they are used to access web systems such as Peoplesoft and Moodle on their campus network, or remotely log into a server machine. Our subjects often type their usernames and passwords to log into popular social network systems.

A possible password is determined as follows. Once an email address, a clarkson username, or a URL is found in a user's text, we extract the first alphanumerical string of at most 12 characters long that follow it. We then generate all substrings of such alphanumerical strings and call them *password candidates*. For a candidate to be considered as a password, we require it to have a minimal length of 6 characters, as a string that is short than 6 characters cannot be a password. We allow a password to have a maximal length of 12. If a password is longer than 12 characters, we only use the first 12 and consider it safe if the first 12 characters of a password is removed. A user may type the same password multiple times. It is also possible that the user uses similar but different passwords for different websites. This strategy will be able to capture the common substrings that are shared by multiple password candidates. Florencio and Herley find that "the average user has 6.5 passwords, each of which is shared across 3.9 different sites. Each user has about 25 accounts that require passwords, and types an average of 8 passwords per day." [6] Therefore, we consider the top 10 substrings that have the highest counts as passwords; we remove these stings from the user's keystroke data. This will generate some false positives but nearly none false negatives for the studied dataset.

Overall, since an attacker needs to know all three pieces of information (URL, username, and password) in order to hack into a subject's account, we believe that our strategy of removing usernames and passwords should be effective as it will remove at least one of the three with a high probability.

4. Evaluation

Table 3 depicts the statistics for the nine PII's that we are interested in detecting from the dataset [16]. For each PII, Table 3 lists the number of users who have typed it in their keystrokes, the total numbers of occurrences in the entire dataset, the min/max/average number of instances by user, and the total amount of keys involved and removed as a result. About 1.33% of the total keystrokes are filtered out due to PII removal.

As noted in Section 3.2, we create the PII ground truth by reading through each subject's keystrokes and manually marking up the PII's. We compare the output of the PII detection program with the created ground truth to count

the numbers of false positives and false negatives as well as calculating precision and recall. Table 4 depicts the results of our evaluation.

We are able to justify the false positives and false negatives in Table 4 as follows.

First of all, we are surprised to see 22 false negatives for first and last names as they are supposed to be straightforward to detect. Upon closer inspection of the actual keystrokes from the user involved, we find out that one user appears to have shared their account with someone else. The 22 false negatives on names belong to the other person who shares the account with the main user.

The 24 false positives for phone numbers are actually numbers used in mathematical computation, random typing, or a mix of digital strings such as dates with some other digits. The one false negative of address ("25 swan") contains only house number and street name ('swan') but without any of the word street, city, or state. We don't believe this will be meaningful to any third party that is geographically far away from Clarkson University. The five false positives include "ON your way", "1 in the way", "7 either way", "4 of the way", and "l, o pulled up road", as an address.

Not surprisingly, the 68 false positives for passwords reflect the empirical nature of our username/password detection mechanism described in Section 3.4, but we are happy with its performance with a precision of 98.3%. On the other hand, the 12 false negatives are ones that lack the necessary context for us to ascertain that they are indeed passwords, where either a URL for the web system or a username is missing. So given that the data has been anonymized, we conclude that all twelve of them are benign. Lastly, due to the time-consuming nature of evaluating the precision for campus ID, SSN, and bank cards, we choose to evaluate only their recalls instead. As shown, our detection program is able to achieve perfect recalls.

The filtered PII keystrokes constitute about 1.33% of the total keystrokes in the original dataset. To assess the impact of removing PII's on keystroke authentication, we compare the detection error rates of Gunetti & Picardi [8] before and after PII removal. As shown in Figure 1, the filtered dataset is only marginally worse than the original in terms of EER. This is as expected, as the users tend to be more familiar with the removed PII's, and thus the PII keystrokes are easier to pass the authentication than less familiar free keystrokes. However, the performance degradation due to PII removal, appears to be small and acceptable.

5. Conclusion

The keystroke data used for authentication often contains sensitive personally identifiable information that if leaked, may lead to various possible harms, including personal embarrassment, financial loss, blackmails, and identify theft. Our PII detection program is able to achieve near perfect recall at the acceptable expense of losing some useful data. The difference of detection error tradeoff before and after removing PII's from the original dataset is negligible. We find that there are indeed significant amounts of PII's in the studied dataset that must be removed before sharing with the research community.

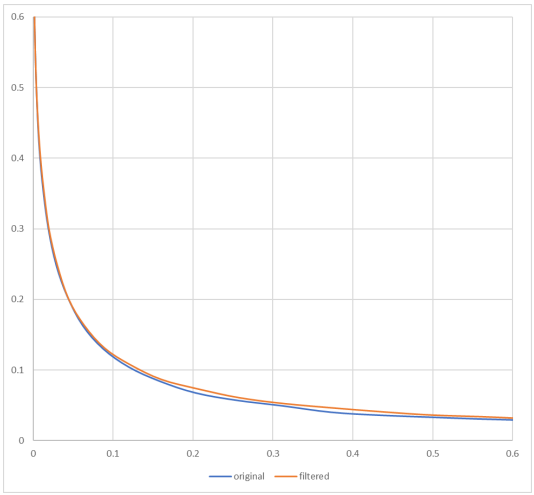

Figure 1. Detection Error Tradeoff (DET) for datasets with/without privacy filtering. X axis: FAR (false accepts) and Y axis: FRR (false rejects). Filtering privacy information slightly raises FRR.

Note that our main contribution in this paper is an experience report on how we remove PII's from a research dataset before sharing, not necessarily any algorithms that can be directly applicable for general enrollment of a person into some keystroke based biometric identification system. The proposed algorithm is run only once before releasing keystroke database into the public domain, and is not applicable for subsequent usage of developed keystroke biometric system, neither person enrollment nor identification. While the utility of the proposed algorithms may be limited, we hope that our experience in removing PII's can be useful to inform the future design of enrollment and identification in keystroke dynamics systems.

Acknowledgment

This work was funded partially by US NSF Award CNS-1314792.

References

[1] A. Acquisti, L. Brandimarte, and G. Loewenstein. Privacy and human behavior in the age of information. *Science*, 347(6221):509–514, 2015. 2

[2] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *IN 6TH WORKSHOP ON PRIVACY ENHANCING TECHNOLOGIES*, pages 36–58, 2006. 2

[3] A. A. Ahmed and I. Traore. Biometric recognition based on free-text keystroke dynamics. *IEEE Transactions on Cybernetics*, 44(4):458–472, April 2014. 3

[4] 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. 3

[5] European Union. EU General Data Protection Regulation (EU-GDPR). http://http://www.privacy-regulation.eu/en/index.htm. Last accessed: 07/05/2018. 2

[6] D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 657–666, New York, NY, USA, 2007. ACM. 5

[7] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM. 2

[8] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, Aug. 2005. 1, 3, 6

[9] J. Hu, D. Gingrich, and A. Sentosa. A k-nearest neighbor approach for user authentication through biometric keystroke dynamics. In *2008 IEEE International Conference on Communications*, pages 1556–1560, May 2008. 1

[10] J. Huang, D. Hou, S. Schuckers, T. Law, and A. Sherwin. Benchmarking keystroke authentication algorithms. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, Dec 2017. 1

[11] R. Janakiraman and T. Sim. Keystroke dynamics in a general setting. *Advances in Biometrics*, pages 584–593, 2007. 3

[12] E. McCallister, T. Grance, and K. Scarfone. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) . Technical Report SP 80-122, NIST, April 2010. 1, 2

[13] A. Messerman, T. Mustafić, 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. IEEE, 2011. 3

[14] F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM, 1997. 3

[15] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359, 2000. 3

[16] C. Murphy, J. Huang, D. Hou, and S. Schuckers. Shared dataset on natural human-computer interaction to support continuous authentication research. In *Biometrics (IJCB), 2017 International Joint Conference on*, pages 1–6. IEEE, 2017. 2, 3, 5

[17] J. C. Stewart, J. V. Monaco, S.-H. Cha, and C. C. Tappert. An investigation of keystroke and stylometry traits for authenticating online test takers. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–7. IEEE, 2011. 3

[18] 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. 3

[19] Y. L. Sun and S. J. Upadhyaya. Secure and privacy preserving data processing support for active authentication. *Information Systems Frontiers*, 17:1007–1015, 2015. 1, 2

[20] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002. 2

[21] P. S. Teh, A. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013:408280, 11 2013. 1

[22] 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. 3