Annals of Telecommunications
https://doi.org/10.1007/s12243-024-01039-z

Fixed-text keystroke dynamics authentication data set—collection and analysis

Halvor Nybø Risto[1] · Olaf Hallan Graven[1]

Received: 21 December 2023 / Accepted: 21 April 2024

Abstract
Keystroke dynamics authentication is a method of authenticating a user and could be an alternative or addition to one-time codes, with minimal user inconvenience. In this study, a new data set was collected for 6 unique passwords, adding to the limited available data sets for keystroke dynamics available for researchers. Data was collected by emulating legitimate users familiar with the passwords and a wider range of attackers with limited login attempts. The data set is analyzed with the use of various methods, and the effects of password length and complexity are investigated. Two algorithms were employed, one achieving an average equal error rate varying between 10.2 and 18.1% depending on the password, and the other method achieving an average true accept rate of 98% and true reject rate of 90.4% by comparing across multiple individuals in the data set. These results provide a benchmark for further studies on this data set.

Keywords Keystroke dynamics authentication · Data set · Password security · Data analysis · Usable security

1 Introduction

This paper is an extension of a paper presented at the CSNet 2023 conference, which detailed the collection and analysis of the data set and results using the correlation algorithm. This paper has the additions: further data analysis using K-Nearest Neighbour with Manhattan distance, T-SNE dimensional reduction for visualization of distinctive groups of data points, and additional results using K-NN to differentiate between the participants in the data set.

It is well known that a common security risk is the use of weak, reused, or compromised passwords, with compromised credentials causing 80% of breaches of web applications by external attackers [1]. While there have been efforts to raise awareness of these risks, many people are unconcerned for their password security, with 24% of people using a variation of the same 8 common passwords [2]. Furthermore, password reuse is very common, with half of IT professionals admitting to reusing passwords [3], the average employee reusing each password an average of 13 times [4], and half of all people using the same password for all their accounts [5]. To make matters even worse, in a study by Google [2], only 45% of respondents said they would change their password after it was discovering that accounts had been breached. It has for a long time been suggested that passwords are on their way out to be replaced by other methods of authentication [6, 7]; however, passwords still remain by far the most used method of personal authentication for account access control, often backed up by 2-factor authentication (2FA) on a mobile device giving a one-time code. While one-time codes are highly effective at preventing account breaches, a significant portion of people do not employ it on their accounts [8]. According to a Ponemon Institute report, roughly half of people report that one-time code 2FA is a cause of irritation and interruption of work flow [9]. Furthermore, 2FA is not a guarantee for security, as hackers can find ways to bypass it [10].

It has been shown in that the characteristics of a person's typing can be used to accurately differentiate between people [11, 12]. This can be used to improve the security of password authentication, by adding an additional authentication step. Multi-factor authentication consists of at least 2 factors, including "something you know", such as a password, "something you have", such as a device, and "something

✉ Halvor Nybø Risto
halvor.n.risto@usn.no

Olaf Hallan Graven
olaf.hallan.graven@usn.no

[1] Department of Science and Industry Systems, University of South-Eastern Norway, Hasbergs vei 36, Kongsberg 3616, Norway

Published online: 15 May 2024

you are", or personal biometrics. The first two of these are widely in common use; however, the use of physiological biometrics for authentication is uncommon due to the extra implementation costs, such as iris scanners or fingerprint readers, or potentially exploitable face-recognition [13]. However, authentication through behavioral keystroke dynamics requires no additional hardware solutions, nor any extra actions by the users, making it a promising method to unintrusively strengthen security.

There is a reported lack of keystroke dynamics data sets [12]. This study aims to add to the available data sets and give preliminary analysis of the data sets using simple statistical methods. The paper consists of sections Background, giving an overview of the literature and existing data sets, Method and Data collection, and finally Data analysis with results.

2 Background

In the 1980s, studies were done on the applicability of keystroke dynamics authentication (KDA) [14–16]. They found that it was a highly promising method to feasibly increase security. Since then, keystroke dynamics for profiling and authentication has been extensively studied for decades, and there is a wide collection of studies and literature on the topic. Some have investigated the effects of password length, password entropy, longitudinal effects [17], typing pressure, touch screens, with free-text typing for continuous authentication [18] as well as fixed-text authentication [11, 12, 19, 20].

2.1 Data sets

While multiple data sets have been made to study keystroke dynamics, they are often limited in size and variety of passwords, and few are publicly available. In [12], the openly available data sets for KDA were surveyed. They report a lack of available data sets for KDA and give a list of 6 KDA data sets, 4 of which included fixed text. The identified fixed-text data sets are ".tie5Roanl" [21], "try4-mbs" [22], "greyc laboratory" [23], and [24] consisting of "yesnomaybe", "bahaNe312!", and "ballzonecart". An issue with some of these data sets is that they use implausible passwords consisting of random characters, which goes against common password recommendations of memorable passphrases [25]. Another issue is the limited selection of passwords in these data sets, making it difficult to determine which features of a password are most beneficial for KDA, such as length, entropy, readability, and typing distance.

The data sets presented in this work consist of readable passwords of varying length and symbol replacements, and unlike other data sets also include an "attack set" of entries from individuals who are unfamiliar with the passwords to emulate an attacker, as well as the legitimate users of the familiarized password. The inclusion of the attack set allows a higher typing variance for KDA benchmarking to make up for a limited data set size.

2.2 Metrics

A list of metrics for typing characteristics have been studied [17]. Using a sequence of timing data of key actions of presses and releases on a keyboard, metrics such as timing between events can be extracted and used to categorize the password entry. The metrics used in this study are

- **Press-to-Press**: the time between when a key is pressed down and the subsequent key is pressed down.
- **Release-to-Press**: the time between when a key is released and the subsequent key is pressed down. This time may be negative.
- **Hold time**: also known as Press-to-Release, the duration of time a key is held down.

Figure 1 demonstrates these three metrics. Other metrics might also be used, such as release-to-release, typing speed, or measurements between more distant keys across the password. Such metrics can be derived from the three metrics used here.

2.3 Possible configurations of KDA

Figure 2 demonstrates two possible ways to combine KDA with one-time code. To achieve a more convenient multifactor user authentication, keystroke dynamics may be used in parallel with one-time codes. Only when the biometrics algorithm denies access, a one-time code from a 2FA app can be required to access the account, along with the correct password. This may reduce irritation and workflow interruption from 2FA, leading to higher adoption and resulting

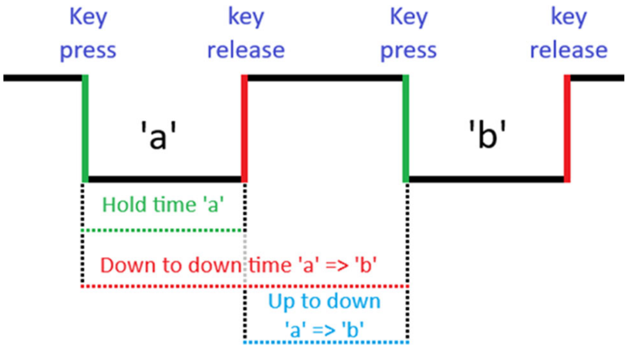
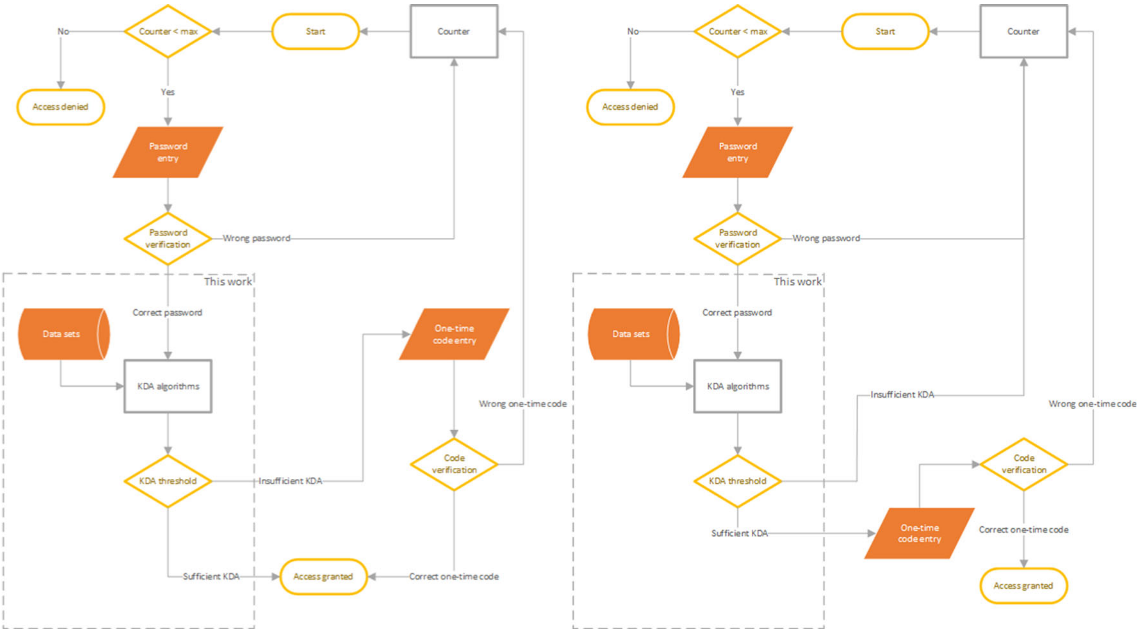


Fig. 1 A sequence of key actions showing the timing metrics



(a) One-time code in parallel with KDA

(b) One-time code in series with KDA

Fig. 2 Two possible configurations of a KDA algorithm

in stronger password security overall. However, the issue of accuracy is highly pertinent for this to be the case. The biometrics algorithm tuned to a user may correctly predict that the log in attempt is by the authorized user, or True Acceptance Rate (TAR), or by an unauthorized attacker, or True Rejection Rate (TRR). It may also erroneously deny access to the authorized user, or False Rejection Rate (FRR), or erroneously allow access to an unauthorized attacker, or False Acceptance Rate (FAR). In the event of a false rejection, the result is an irritation for the user having to go through an extra authentication step with the required one-time code. More importantly, a false acceptance would result in an account breach. Therefore, the applicability of keystroke dynamics to augment 2FA in parallel is strongly dependent on the false acceptance rate. Alternatively, keystroke dynamics can be implemented as an additional step in series with a one-time code for high-security scenarios. In this case, the FRR is the major factor for applicability. Using a distance metric between a password entry and the legitimate users recorded keystroke dynamics, a threshold can be used as a classifier. This threshold decides the strictness of the authentication and therefore affects both FRR and FAR. A lower FAR can be achieved at the cost of a higher FRR. Typically, studies give the Equal Error Rate (EER), which is the point where FAR and FRR meet. However, with the two proposed authentication flows, it is also useful to provide results of when the algorithm is optimized for lower false acceptances or false rejections. To achieve this, the confidence threshold to pass the KDA step can be adjusted. A higher threshold increases the FRR and decreases the FAR.

3 Method

The study is set up to gather data in a simulated real life situation. In the event of a widely compromised password, the number of login attempts per person will be limited, and the attackers will likely be unfamiliar with the password. One possible scenario is a group of students acquiring the password of a university faculty member, attempting to log into their account to gain access to exam material. To simulate this scenario, two data sets are needed. Firstly, a data set consisting of a large number of login attempts by a wide variety of participants at the university, here referred to as the “attack data set”. Secondly, a data set of the “legitimate” users of a password, consisting of a small set of individuals with a larger quantity of data per individual, referred to as the “defence data set”.

The study is designed to determine the following:

RQ1: How consistently do individuals type the collected passwords, and how is this affected by the typing proficiency (i.e., keystrokes per second)?

RQ2: How do features such as password length and complexity affect the consistency and distinctiveness of an individuals keystroke dynamics?

RQ3: How do realistic attackers differ from users, and can a data set of plausible attacker attempts augment the accuracy of a KDA algorithm?

3.1 Data collection

A system setup was developed to gather the data in the study, using a keylogger based on C++ which records every key press and release with millisecond timing. To maintain the data integrity and continuous data collection, measures were taken to ensure the participants could not stop, disturb, or sabotage the continuous data collection, by not allowing the participants to exit the data collection application, access the file system, or access other applications. The setup displays a word or phrase to type and only logs the successful typing attempts. Any unnecessary key presses are not logged. If there is no typing within a time limit, the program resets. Keyboard keys which can interfere with the data collection were disabled, and duplicate copies of the recorded data were made regularly. Many USB-connected keyboards have a polling frequency of less than 1000Hz, often 125Hz resulting in a 8 millisecond polling period. This is not an issue with PS/2 keyboards, so a PS/2 Norwegian keyboard was used to achieve millisecond time resolution in this experiment.

The data set collected for this study consists of two parts. The first is a set of 5 people typing the set of passwords 200 times and represents legitimate users of a password, and secondly a larger set of 100 people, each typing two passwords ten times, emulating a set of attackers. This was to provide a realistic comparison to compromised passwords being misused, where the small set of participants is equivalent to the legitimate users of accounts, while the larger set of participants giving 10 typing attempts per password represents malicious login attempts.

The attack data sets were collected at the university campus reception. The data set contains the millisecond timing of every key press and key release, for a set of passwords being typed by a set of participants. The keylogger was left unattended, running on a publicly available PC on campus. The participants were asked to write 2 of 6 possible passwords 10 times each and were rewarded with a unique code which could be exchanged for a small chocolate bar. The PC was left unattended at the campus during data collection; however, to prevent participants from attempting to get multiple codes, only one chocolate was given per person in exchange for a code, and a sleep delay was added to the system after each participant had received their code. In the case that the participant gives up half way, the program will time out and reset, discarding the data from their attempt. For the defence set, the keylogger was altered to take a higher count of password entries. A group of 5 individuals were recruited to type all 6 passwords, 200 times each, on the same machine and keyboard as the attack set.

3.2 The passwords used

The passwords selected for this data set are shown in Table 1. They consist of varying lengths and complexity.

The number of keystrokes is how many keys need to be pressed to write the password, excluding "enter". Since 3 measurements are made per keystroke, the number of dimension is three times the keystrokes, plus the hold time of enter. The definition of entropy used here is the measure of possible configurations. Given a password length L and a pool of symbols R, there are R^L possibilities. E is the bits of entropy given that $2^E = R^L$, which gives $E = log_2(R^L)$. Four plausible passwords were made, consisting of one to four selected words from a dictionary. The shorter two passwords have two versions with special symbols, giving them a larger symbol pool L and thus higher entropy, while the longer pass-phrases are not appropriate for character substitutions as it would be too inconvenient to type. Since a common way of cracking passwords is a dictionary attack, it is common advice to not use dictionary words in passwords. Another common advice is to create memorable pass-phrases by chaining multiple words to achieve a high entropy [26, 27]. Nevertheless, using dictionary words in passwords is common and is therefore relevant to investigate.

3.3 Data quality and usability

The data consists of typing timing data of key presses and releases with 1 millisecond time resolution. Certain factors may affect the quality and usability of the data set.

Table 1 The selected passwords

Password	Keystrokes	Dimensions	Entropy
observer	8	25	37 bits
Ob$erv3r	11	34	49 bits
gigabit receiver	16	49	75 bits
Gigab!t R3ceiver	19	58	98 bits
flying automatic monster	24	73	112 bits
repetition learn machine thinker	32	97	150 bits

Time resolution The data sets have a time resolution of 1 millisecond. A PS/2 connected keyboard was used, since many USB-connected keyboards have a polling frequency lower than 1kHz. It has been shown in [28] that higher clock resolutions produce better results.

Size Since the defence data sets are limited in size and participants, attack sets were also collected. The attack data sets include 103 individuals typing two passwords ten times each, giving roughly 33 participants per password. This allows the attack sets to have a much higher variance than each defence set, which produces more realistic results than only comparing between the defence sets. While ten repetitions may not be sufficient for an algorithm to differentiate each individual, this data set is only intended to emulate an attack case. The "defence" data set consists of a much smaller number of individuals, however with a sufficiently high quantity of data per person, allowing an algorithm to recognize certain individuals typing certain passwords.

Demographic The data collection was performed at a university campus, and the participants consist mostly of students. There is a high variance in typing proficiency in the participation pool. The participants are a random sample of people at the university.

Repeating individuals While the data collection is stated to only permit one session per person of 20 password entries, some individuals may come more than once, since participation is anonymous and unsupervised. However, the reward handout for participation is done manually, and it is stated that only one reward is given per person. In the case that some individuals attempt to participate on multiple days, the data may still be useful, as each person may type slightly differently on different days, and is within a realistic attack scenario where the same attacker may attempt to login on different days.

Typing speed The typing speed of the attack data set is on average lower than the defence attack data set. This was expected, as the individuals in the defence set may improve their proficiency for the typed password, while the attack set is a set of attackers unfamiliar with the passwords. The set of attempts from the attack data set may be used to validate the algorithm.

Repetitive typing The act of typing the passwords many times in a row may affect data quality in the defence set. Other studies have investigated the longitudinal effects of typing over a longer time period; however, the data sets presented in this study were each collected in single sessions. There may be factors such as boredom and typing fatigue that manifest as detectable patterns in the data.

Realism The attack data set emulates a real-world scenario where many different attackers get a handful of attempts at logging in with a compromised password. This data is useful since it gives a look into how people generally type the passwords to compare with the typing patterns of specific individuals.

Plausibility of passwords The passwords used were selected from English dictionary words. Some of the pre-existing fixed-text data set passwords consist of a string of random symbols, while this data set aims to investigate the effect of password length and character substitutions.

3.4 Data format

The first step is to process the data into a more useful form. The data collection program produces two formats which can be derived from each other. One is a sequential list of key presses and releases with millisecond timing, while the other is a set of metrics of the timing between these key presses or key releases. This data could be further processed to show the timing relations between an arbitrary key to another key. Each row in the data sets is a single password entry, consisting of three metrics for each key of the corresponding password in sequential order.

4 Data analysis

4.1 Variance

The standard deviation for sampled data can be used to show typing consistency and is found by calculating the average deviation for each data column. This is shown in Table 2.

The participants have a variance of standard deviations ranging from 0.024 to 0.1916. This correlates with typing speed, and a lower variance is expected to correlate with

Table 2 Standard deviation

Participant	1	2	3	4	5	Attack	Average
”observer”	0.0240	0.0461	0.0417	0.0943	0.1153	0.1944	0.0859
”Ob$erv3r”	0.0431	0.1172	0.1567	0.2431	0.2180	0.3719	0.1916
”gigabit receiver”	0.0265	0.0540	0.0543	0.1254	0.1630	0.1542	0.0962
”Gigab!t R3ceiver”	0.0390	0.0799	0.0784	0.1746	0.1514	0.2579	0.1302
”flying automatic	0.0245	0.0420	0.0622	0.1308	0.1369	0.1228	0.0865
”repetition learn	0.0287	0.0589	0.0524	0.1123	0.1372	0.1463	0.0893

Table 3 Correlation analysis

Data sets / Participant	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Attack
”observer”						
Participant 1	**0.904**	0.183	0.528	0.166	0.239	0.338
Participant 2	0.183	**0.886**	0.274	0.726	0.646	0.553
Participant 3	0.528	0.274	**0.779**	0.408	0.195	0.398
Participant 4	0.166	0.726	0.408	**0.788**	0.564	0.544
Participant 5	0.239	0.646	0.195	0.564	**0.791**	0.493
Attack	0.338	0.553	0.398	0.544	0.493	**0.476**
”Ob$erv3r”						
Participant 1	**0.908**	0.724	0.647	0.525	0.611	0.520
Participant 2	0.724	**0.874**	0.746	0.708	0.797	0.713
Participant 3	0.647	0.746	**0.743**	0.572	0.720	0.641
Participant 4	0.525	0.708	0.572	**0.755**	0.656	0.642
Participant 5	0.611	0.797	0.720	0.656	**0.782**	0.706
Attack	0.520	0.713	0.641	0.642	0.706	**0.677**
”gigabit receiver”						
Participant 1	**0.856**	−0.052	0.536	−0.121	0.029	0.114
Participant 2	−0.052	**0.699**	0.173	0.383	0.374	0.311
Participant 3	0.536	0.173	**0.595**	0.122	0.194	0.279
Participant 4	−0.121	0.383	0.122	**0.650**	0.360	0.366
Participant 5	0.029	0.374	0.194	0.360	**0.413**	0.320
Attack	0.114	0.311	0.279	0.366	0.320	**0.358**
”Gigab!t R3ceiver”						
Participant 1	**0.814**	0.453	0.509	0.273	0.269	0.289
Participant 2	0.453	**0.796**	0.471	0.545	0.512	0.500
Participant 3	0.509	0.471	**0.613**	0.401	0.468	0.452
Participant 4	0.273	0.545	0.401	**0.679**	0.636	0.513
Participant 5	0.269	0.512	0.468	0.636	**0.684**	0.525
Attack	0.289	0.500	0.452	0.513	0.525	**0.502**
”flying automatic monster”						
Participant 1	**0.855**	0.155	0.605	−0.007	0.032	0.272
Participant 2	0.155	**0.767**	0.221	0.395	0.379	0.363
Participant 3	0.605	0.221	**0.611**	−0.032	0.082	0.269
Participant 4	−0.007	0.395	−0.032	**0.700**	0.423	0.289
Participant 5	0.032	0.379	0.082	0.423	**0.490**	0.280
Attack	0.272	0.363	0.269	0.289	0.280	**0.325**
”repetition learn machine thinker”						
Participant 1	**0.809**	0.059	0.546	−0.038	−0.016	0.204
Participant 2	0.059	**0.641**	0.191	0.286	0.255	0.288
Participant 3	0.546	0.191	**0.634**	0.102	0.072	0.252
Participant 4	−0.038	0.286	0.102	**0.573**	0.437	0.318
Participant 5	−0.016	0.255	0.072	0.437	**0.473**	0.277
Attack	0.204	0.288	0.252	0.318	0.277	**0.308**

Data in bold indicates the consistency of proportional timing between keys

KDA accuracy, as higher typing consistency is expected to produce more distinct classifications. The average standard deviation for a password can then indicate which password would perform the best with KDA. It can be seen that special characters result in a higher standard deviation. The password length does not appear to have a significant effect, which is expected since this is the average of the variances of each data measurement.

4.2 Correlation

Correlation is a measure of statistical relationships between data. The correlation between two entries X and Y is defined in Eq. 1, where $\bar{x}$ and $\bar{y}$ are the averages of the sets, and x and y are the members of the sets.

$$Correl(X, Y) = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \tag{1}$$

Each entry of each set is correlated to the entries of each other set, to get the average correlation between sets. Table 3 shows the average correlations of entries from one data set to entries of another data set. As each data set has some variance, the average correlation of entries to its own data set indicates the consistency of proportional timing between keys. Standard deviation showed average timing consistency of each key, while this is showing consistency of timing of each key relative to each other. As the average standard deviation of each feature was highest when using special characters, the self-correlation can also be seen to be lower in these cases. Additionally, it can be seen that increasing the length of the password has a detrimental effect on the self-correlation, i.e., consistency of typing between entries.

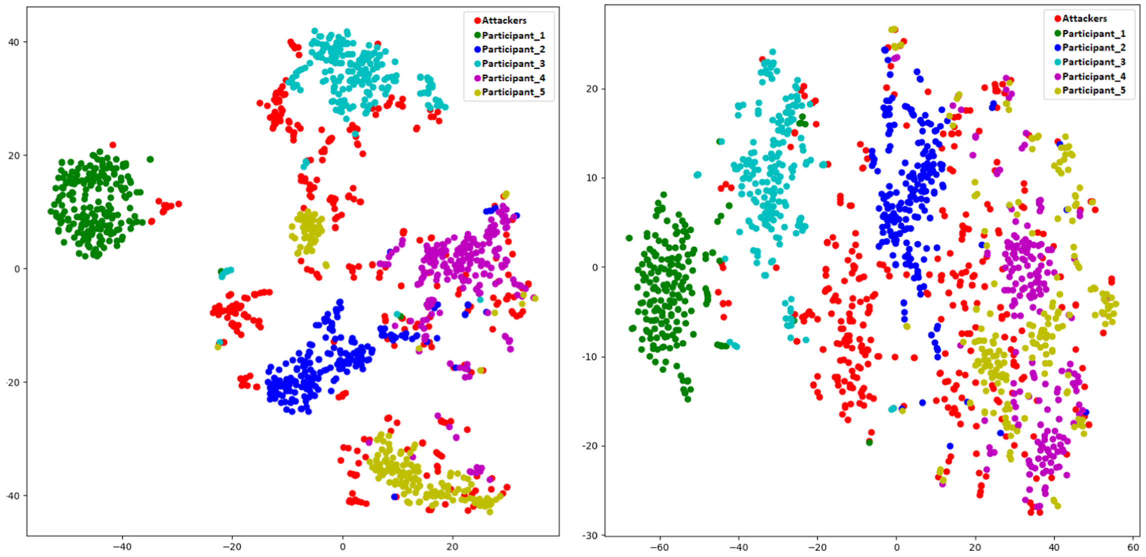
4.3 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique common for data visualization and can reveal patterns and structures within high-dimensional data by mapping them to a lower-dimensional space while preserving pairwise similarities. Figure 3a shows the t-SNE visualization of the “observer” data sets, reduced from 25 dimension. Figure 3b shows a visualization of the data sets for “Repetition Learn Machine Thinker”, reduced from 97 dimensions.

The attack set has a high variance. As the attack set consists of a mix of approximately 30 attackers per password, this is expected. The defence sets are well clustered, especially with the faster typers. Participants 4 and 5 are the slowest typers in the set and are less well-defined in this visualization. Comparing the two graphs, it can be observed that the shortest passwords result in significantly better clustering than the longer password.

4.4 K-nearest neighbor

Using K-nearest neighbor, an entry can be classified as one of the participants. This is done by measuring the distance from an entry, e.g., a data point, to the data points in a labeled data set and classifying the data point as the same class as the nearest neighbors. A constant K is used for the number of neighbors used to decide the class. To measure the distance between two data points, Manhattan distance is used, shown in Eq. 2, where a and b are the coordinate elements of data points A and B, respectively. Other distance metrics such



(a) ”observer” (b) ”repetition learn machine thinker”

Fig. 3 t-SNE of **a** the 25-dimensional data set and **b** the 97-dimensional data set

Table 4 KNN results, K = 3

Dataset	True accept / False reject	True reject / False accept
”observer”		
Participant 1	0.99 / 0.01	0.96 / 0.04
Participant 2	0.985 / 0.015	0.91 / 0.09
Participant 3	1 / 0	0.89 / 0.11
Participant 4	0.975 / 0.025	0.895 / 0.105
Participant 5	0.965 / 0.035	0.835 / 0.165
”Ob$erv3r”		
Participant 1	0.99 / 0.01	1 / 0
Participant 2	0.94 / 0.06	0.935 / 0.065
Participant 3	0.91 / 0.09	0.91 / 0.09
Participant 4	0.975 / 0.025	0.895 / 0.105
Participant 5	0.875 / 0.125	0.69 / 0.31
”gigabit receiver”		
Participant 1	1 / 0	0.99 / 0.01
Participant 2	0.995 / 0.005	0.91 / 0.09
Participant 3	0.99 / 0.01	0.81 / 0.19
Participant 4	0.99 / 0.01	0.875 / 0.125
Participant 5	0.98 / 0.02	0.85 / 0.15
”Gigab!t R3ceiver”		
Participant 1	0.995 / 0.005	0.995 / 0.005
Participant 2	0.98 / 0.02	0.925 / 0.075
Participant 3	0.995 / 0.005	0.885 / 0.115
Participant 4	0.97 / 0.03	0.915 / 0.085
Participant 5	0.95 / 0.05	0.82 / 0.18
”flying automatic monster”		
Participant 1	1 / 0	0.945 / 0.055
Participant 2	1 / 0	0.795 / 0.205
Participant 3	0.995 / 0.005	0.915 / 0.085
Participant 4	0.995 / 0.005	0.995 / 0.005
Participant 5	0.97 / 0.03	0.965 / 0.035
”repetition learn machine thinker”		
Participant 1	1 / 0	0.985 / 0.015
Participant 2	1 / 0	0.875 / 0.125
Participant 3	1 / 0	0.935 / 0.065
Participant 4	0.995 / 0.005	0.915 / 0.085
Participant 5	1 / 0	0.905 / 0.095

as Euclidean distance were also considered; however, Manhattan distance was found to be better for high-dimensional data.

The issue with implementing this method in a real scenario is the requirement of having a data set of multiple people typing the same password. However, it may be possible to synthesize these data.

$$ManhattanDistance(A, B) = \sum |a - b| \qquad (2)$$

To calculate the true accept and false reject of the K-nearest neighbor method for a data set, each data point is evaluated with the data set, with the evaluated data point removed from it, and the resulting classification is compared with the original classification. The value of K was selected based on the resulting accuracy. To find the true reject/false accept, the data points from the attack data set are classified as either the target class (false accept) or another class (true reject). A choice must be made here if a data point may be

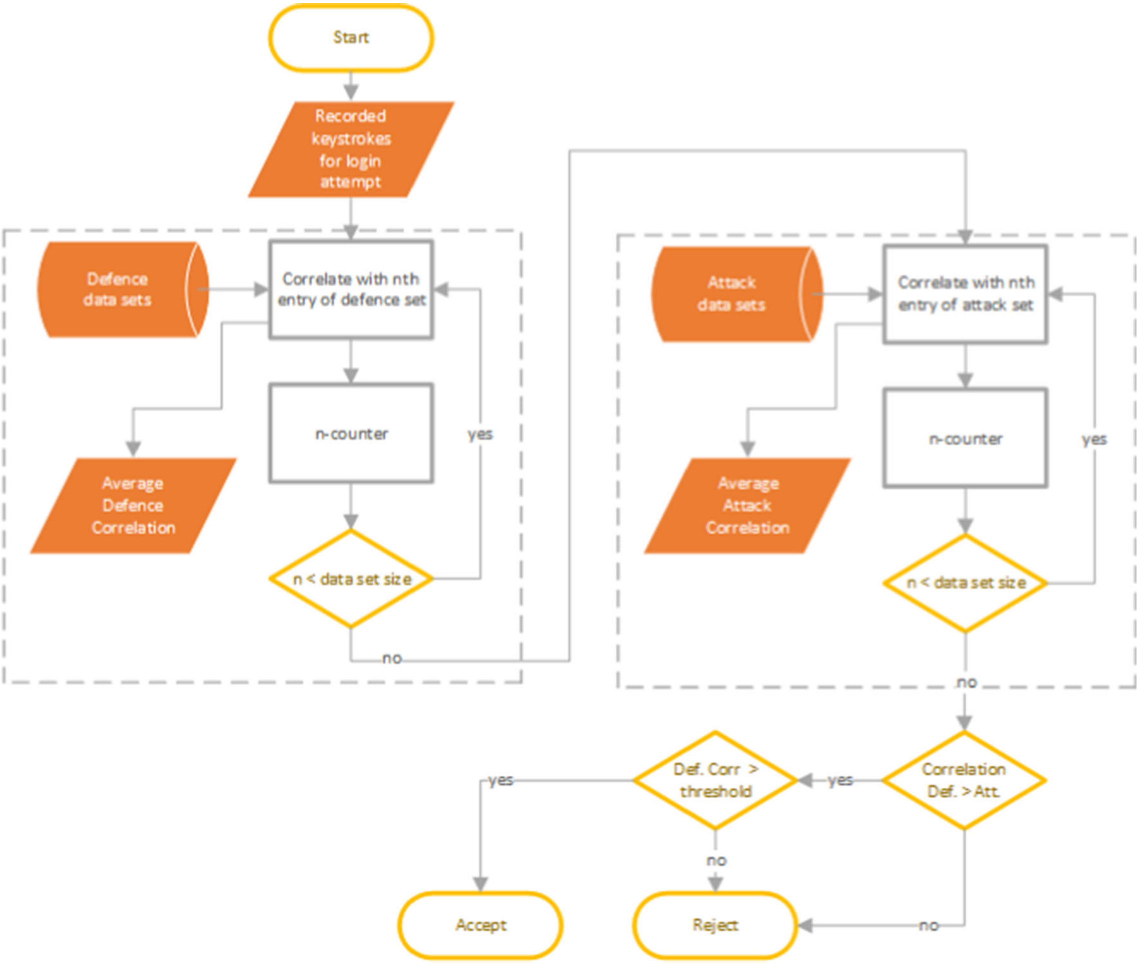

Start
Recorded keystrokes for login attempt
Defence data sets
Correlate with nth entry of defence set
n-counter
yes
Average Defence Correlation
n < data set size
no
Attack data sets
Correlate with nth entry of attack set
n-counter
yes
Average Attack Correlation
n < data set size
no
Correlation Def. > Att.
yes
Def. Corr > threshold
yes
no
no
Accept
Reject

Fig. 4 The KDA correlation algorithm

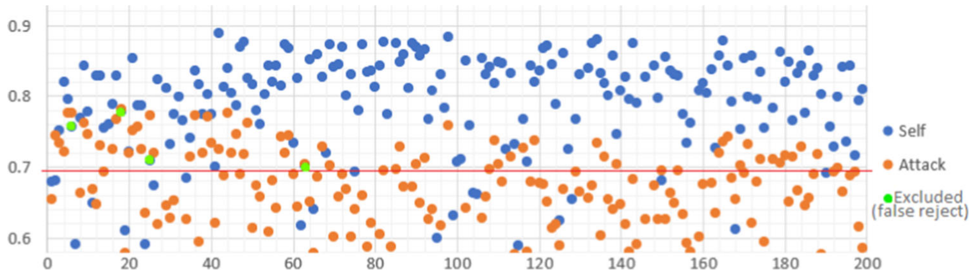


Fig. 5 Participant 4, ”Ob$erv3r”, correlation of each entry of the defence set with its own data set (blue) and with the attack set (orange)

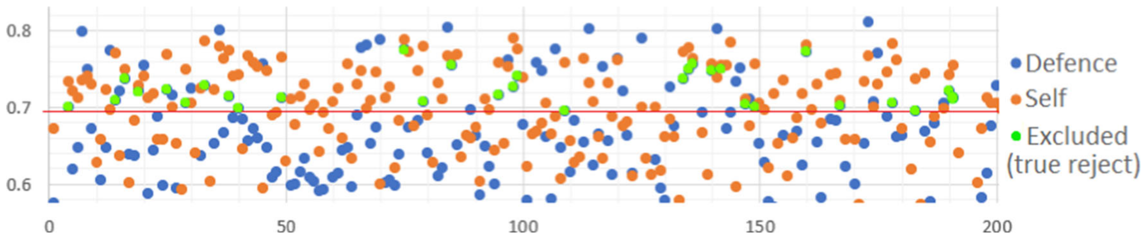


Fig. 6 Participant 4, ”Ob$erv3r”, correlation of each entry of the attack set with its own data set and with the defence set

Table 5 Correlation results

	Avg. correlation	EER	FRR → 0	FAR = 0	Typing
Data set	Self / Atk	FRR / FAR	FRR / FAR	FRR	Speed
”observer”					
Participant 1	0.915 / 0.338	2.5% / 2.57%	0.5% / 25.14%	3%	8.69 keys/s
Participant 2	0.791 / 0.553	7.5% / 7.71%	0% / 63.42%	25%	4.73 keys/s
Participant 3	0.818 / 0.397	11.5% / 11.42%	3% / 39.42%	28%	7.59 keys/s
Participant 4	0.813 / 0.543	13.5% / 13.42%	0% / 78%	40.5%	3.82 keys/s
Participant 5	0.821 / 0.493	16.0% / 16.0%	1.5% / 42%	31.5%	3.89 keys/s
Attack	0.476				5.01 keys/s
”Ob$erv3r”					
Participant 1	0.917 / 0.520	4.00% / 3.82%	0.5% / 10.29%	7%	8.18 keys/s
Participant 2	0.892 / 0.712	16% / 16.47%	2.5% / 69.41%	25%	4.07 keys/s
Participant 3	0.787 / 0.640	24% / 24.11%	16% / 35%	37.5%	3.88 keys/s
Participant 4	0.776 / 0.642	17% / 16.7%	8.5% / 26.47%	41.5%	2.50 keys/s
Participant 5	0.805 / 0.706	29.50% / 29.70%	4% / 72.94%	51%	2.76 keys/s
Attack	0.676				2.29 keys/s
”gigabit receiver”					
Participant 1	0.865 / 0.113	3.00% / 3.42%	0% / 9.42%	8%	10.85 keys/s
Participant 2	0.716 / 0.310	7.00% / 6.85%	0.5% / 35.42%	25%	4.86 keys/s
Participant 3	0.636 / 0.278	17.00% / 17.14%	3% / 24.57%	37%	8.15 keys/s
Participant 4	0.675 / 0.366	12.00% / 11.71%	2% / 55.14%	35%	3.37 keys/s
Participant 5	0.448 / 0.319	24.5% / 24.57%	11.5% / 35.42%	46%	3.64 keys/s
Attack	0.355				4.22 keys/s
”Gigab!t R3ceiver”					
Participant 1	0.825 / 0.289	4% / 4.11%	0.5% / 8.82%	18.5%	8.79 keys/s
Participant 2	0.809 / 0.500	10% / 10%	1.5% / 48.52%	22%	4.30 keys/s
Participant 3	0.638 / 0.451	19.5% / 19.70%	1.5% / 28.52%	50.5%	6.05 keys/s
Participant 4	0.705 / 0.513	17.5% / 17.35%	1% / 56.47%	38.5%	2.66 keys/s
Participant 5	0.720 / 0.525	19% / 19.11%	0.5% / 64.41%	44%	2.81 keys/s
Attack	0.500				2.77 keys/s
”flying automatic monster”					
Participant 1	0.874 / 0.272	7% / 7.05%	0% / 31.76%	14%	10.86 keys/s
Participant 2	0.778 / 0.362	6% / 5.58%	0% / 66.76%	11%	5.82 keys/s
Participant 3	0.651 / 0.268	16.5% / 16.76%	1.5% / 35.29%	38.5%	8.54 keys/s
Participant 4	0.722 / 0.288	9% / 8.82%	0% / 28.82%	21%	3.01 keys/s
Participant 5	0.529 / 0.280	17.5% / 17.35%	1% / 36.47%	38%	3.67 keys/s
Attack	0.325				5.40 keys/s
”repetition learn machine thinker”					
Participant 1	0.825 / 0.203	6% / 6.17%	0% / 28.82%	17.5%	11.75 keys/s
Participant 2	0.669 / 0.288	9.5% / 9.41%	0.5% / 43.23%	20.5%	5.34 keys/s
Participant 3	0.649 / 0.252	11.5% / 11.47%	0% / 33.52%	32.5%	9.02 keys/s
Participant 4	0.593 / 0.318	14% / 14.11%	0% / 55.29%	46%	3.53 keys/s
Participant 5	0.497 / 0.277	20% / 19.70%	1% / 42.35%	47.5%	3.38 keys/s
Attack	0.307				5.01 keys/s

classified as belonging to the attack set. If not, then the attack data points must be classified as one of the 5 users, guaranteeing an average of 20% false accept rate. If the attacks data points can be classified as belonging to the attack set, the data points from specific individual from the attack set which a data point is taken as an attempted attack must be excluded, since the attack set is only meant to represent plausible attackers and not include the specific attacker attempting to log in. Note that although attack participants were instructed not to participate twice, and the data collection process was only partially supervised, it is possible some have participated more than once, which would skew the false positive rate using this method. For the results listed in Table 4, it is assumed this is not the case.

4.5 Correlation as a KDA algorithm

By correlating a password entry to the data sets, a correlation threshold can be used to determine if the entry belongs to the legitimate user or an unknown attacker.

When a password entry is inputted to the KDA algorithm to determine if it is the legitimate user or an attacker, it is

- Correlated with every entry of the defence data set, to find the average correlation with the data set for that user.
- Correlated with every entry in the attack data set, to find the average correlation with attackers.
- Evaluated based on average correlations to defence and attack data sets, compared to each other and a correlation threshold.

The correlation value may be used to differentiate between user and attacker. One method is to assume the highest correlation is always the match. Another method is requiring correlation above a certain threshold. A third method is requiring that the difference between self-correlation and attack-correlation is high enough. The method used here is a combination of the two first.

Two types of measures can be made here. A common performance metric is the Equal Error Rate (EER), the point at which the FAR is equal to the false rejection rate. The confidence threshold is configured for each individual and password. The alternative measure is one where either a low FAR or FRR is highly prioritized, by measuring the accuracy at the point where the other hits 0.

Through the KDA process in Fig. 4, a data entry is correlated with both the defence set and attack set to determine if it belongs to the correct user or an attacker. To achieve an accept, the correlation coefficient to the defence set needs to be higher than the set threshold, but also higher than the correlation to the attack set. When correlating the entries of the defence set, this may lead to a false reject as demonstrated in Fig. 5. However, when correlating the entries of the attack set to the defence set, attack entries that have a correlation coefficient the attack set, leading to true rejects above the threshold, demonstrated in Fig. 6, allowing the threshold at the EER to be lowered, which in turn lowers the FRR.

When correlating the defence entries to the defence set, the specific entry in question is excluded from the set, and when attack entries are correlated to the attack set, every entry of the corresponding participant in the attack set is excluded. By comparing the entries marked as green in Figs. 5 and 6, it is clear that this method of correlating to an attack set in addition to the defence set can augment the accuracy of the algorithm, as the additional true rejects outnumber the additional false rejects. An issue with implementing this

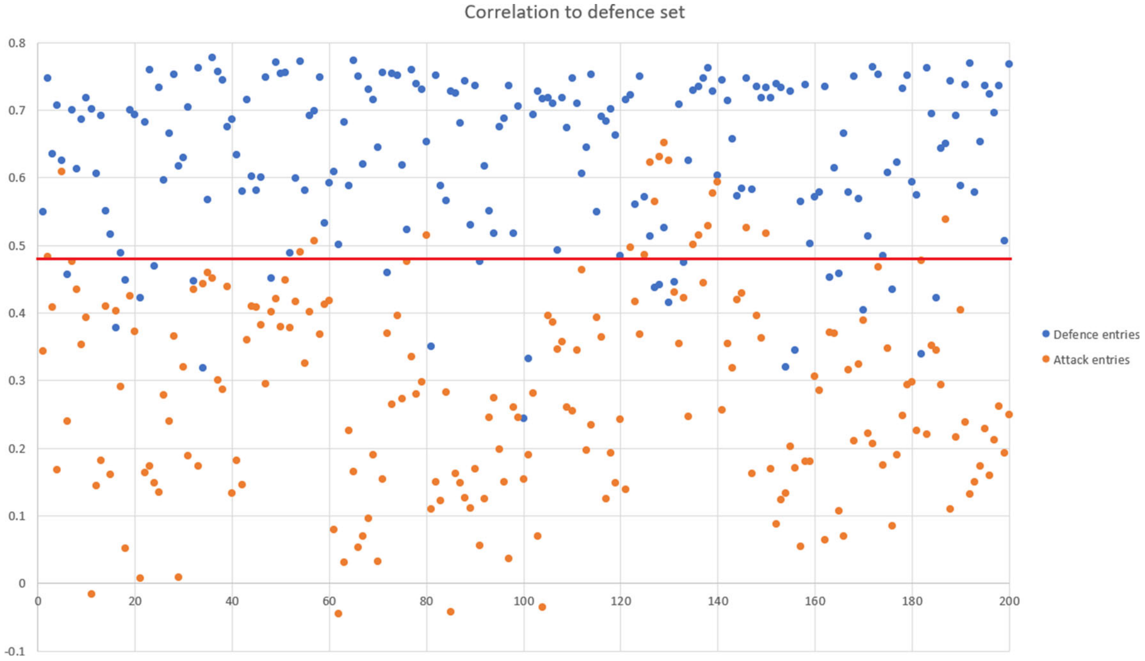


Fig. 7 Defence entries and attack entries with a threshold

method is that reducing false rejects to an absolute minimum might be a priority for the sake of usability, depending on whether a low false reject rate, a high true accept rate, or a low EER is prioritized.

To find the EER of this process for each defence set, each entry of the defence set is evaluated to find the FRR and TAR, and each entry of the attack set is evaluated to find the TRR and FAR. The threshold is then adjusted so that FAR is equal to FRR. Additionally, the threshold can be adjusted so that either FRR or FAR is equal to zero. Note that since excluded entries may lead to false rejects, a FRR of 0 can not always be achieved by adjusting the threshold. Table 5 presents these results for the data sets collected. Figure 7 shows these correlations for one of the participants typing one of the passwords as an example.

5 Conclusion and future work

The main contributions from this work are the public data set for keystroke dynamics research, an analysis of the data set using various statistical methods with investigation into the effects of password features, KNN, and correlation algorithm with results as a benchmark for further research.

The consistency of the data sets was analyzed using various methods, such as standard deviation of each data column, average correlation of each entry of a set with its own set as well as the other sets, and t-SNE to visualize the distinct clustering of each participant. It was found that typing speed was correlated with accuracy.

Two algorithms were used to produce benchmark results with the collected data sets. The KNN method is dependent on sets of multiple people typing the same password; however, these data could potentially be synthesized for an arbitrary password by simulating a human typist. The correlation or distance threshold method can be used on single individuals typing a unique password and is validated using a set of realistic attackers. Both methods can be augmented with a broad set of realistic attack attempts. Password length had a positive effect with the KNN method and a negative effect with the distance threshold method. Password complexity had a detrimental effect with both methods.

The authors will proceed to implement machine learning algorithms such as LSTM in order to improve the results and investigate the effects of password features and the usage of an attack set with such algorithms. The data sets are publicly available online on USN Figshare and [29], adding to the limited selection of data sets for KDA research, and include a variance of passwords, typed by multiple people with a variance of typing speeds.

Acknowledgements This research has been supported by the research council of Norway and Indra Navia as part of an industrial Ph.D. with University of South-Eastern Norway. The authors thank the anonymous participants.

Author Contributions H.N.R. wrote the main manuscript. O.H.G. provided guidance with planning, experiment design, and manuscript review and corrections. H.N.R is a PhD student, and O.H.G. is the PhD supervisor. All authors reviewed the manuscript.

Funding Open access funding provided by University Of South-Eastern Norway. This research is funded by the Research Council of Norway, as well as Indra Navia as part of an industrial Ph.D with project number 323117. The Ph.D. supervisor is funded by the University of South-Eastern Norway.

Data Availability Keystroke dynamics data set is available at https://doi.org/10.23642/usn.23790858.

Declarations

Ethics approval No ethics approval is needed, as all participants are anonymous and no personal information is collected.

Consent to participate All participants participated knowingly and voluntarily in the study.

Consent for publication All authors consent to publication.

Conflict of interest The authors declare no competing interests.

References

1. Verizon (2022) Data breach investigations report. https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf. (Accessed on 23 Mar 2023)
2. The United States of P@ssw0rd$ (2019). https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/PasswordCheckup-HarrisPoll-InfographicFINAL.pdf. (Accessed on 23 Mar 2023)
3. 15 password statistics that will change your attitude toward them - Cybersecurity ASEE (2022) . https://cybersecurity.asee.co/blog/password-statistics-that-will-change-your-attitude/. (Accessed on 23 May 2023)
4. Online Security Survey, Google / Harris Poll (2019). https://services.google.com/fh/files/blogs/google_security_infographic.pdf. (Accessed on 23 Mar 2023)
5. LMI0828a-IAM-LastPass-State-of-the-Password-Report.pdf (2020). https://lp-cdn.lastpass.com/lporcamedia/document-

library/lastpass/pdf/en/LMI0828a-IAM-LastPass-State-of-the-Password-Report.pdf. (Accessed on 23 Mar 2023)

6. Gates predicts death of the password (2004). https://www.cnet.com/news/privacy/gates-predicts-death-of-the-password/. (Accessed on 23 Mar 2023)
7. IDG-Passwordless-The-Future-of-User-Authentication.pdf (2020). https://www.okta.com/sites/default/files/2021-03/IDG-Passwordless-The-Future-of-User-Authentication.pdf. (Accessed on 23 Mar 2023)
8. Microsoft: 99.9% of compromised accounts did not use multi-factor authentication | ZDNET (2020). https://www.zdnet.com/article/microsoft-99-9-of-compromised-accounts-did-not-use-multi-factor-authentication/. (Accessed on 23 May 2023)
9. 25+ Password statistics that may change your password habits. https://www.comparitech.com/blog/information-security/password-statistics/. (Accessed on 05/23/2023)
10. Multi-factor authentication is (not) 99 percent effective (2023). https://cybersecurityventures.com/multi-factor-authentication-is-not-99-percent-effective/. (Accessed on 05/23/2023)
11. Teh PS, Teoh A, Yue S (2013) A survey of keystroke dynamics biometrics. Sci World J 2013:408280. https://doi.org/10.1155/2013/408280
12. Sadikan SFN, Ramli TAA, Md Fudzee MF (2019) A survey paper on keystroke dynamics authentication for current applications. 2173:020010. https://doi.org/10.1063/1.5133925
13. Galbally J, Marcel S, Fierrez J (2014) Biometric antispoofing methods: a survey in face recognition. IEEE Access 2:1530–1552. https://doi.org/10.1109/ACCESS.2014.2381273
14. Gaines RS, Lisowski W, Press SJ, Shapiro N (1980) Authentication by keystroke timing: some preliminary results. Technical report, Rand Corp Santa Monica CA
15. Leggett J, Williams G (1988) Verifying identity via keystroke characterstics. Int J Man-Mach Stud 28(1):67–76. https://doi.org/10.1016/S0020-7373(88)80053-1
16. Umphress D, Williams G (1985) Identity verification through keyboard characteristics. Int J Man-Mach Stud 23(3):263–273
17. Parkinson S, Khan S, Badea A, Crampton A, Liu N, Xu Q (2022) An empirical analysis of keystroke dynamics in passwords: a longitudinal study. IET Biom. https://doi.org/10.1049/bme2.12087
18. Kim J, Kim H, Kang P (2018) Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection. Appl Soft Comput 62:1077–1087
19. Giot R, El-Abed M, Rosenberger C (2011). Keystroke dynamics overview. https://doi.org/10.5772/17064
20. Banerjee SP, Woodard DL (2012) Biometric authentication and identification using keystroke dynamics: a survey. J Pattern Recognit Res 7(1):116–139
21. Killourhy K, Maxion R (2010) Why did my detector do that. In: Recent advances in intrusion detection, pp 256–276. Springer
22. Typing Biometrics Keystroke Dataset (2014). https://personal.ie.cuhk.edu.hk/~ccloy/downloads_keystroke100.html. (Accessed on 23 May 2023)
23. Giot R, El-Abed M, Rosenberger C (2009) Greyc keystroke: a benchmark for keystroke dynamics biometric systems. In: 2009 IEEE 3rd International conference on biometrics: theory, applications, and systems, pp 1–6. https://doi.org/10.1109/BTAS.2009.5339051
24. Vural E, Huang J, Hou D, Schuckers S (2014) Shared research dataset to support development of keystroke authentication. IJCB 2014 - 2014 IEEE/IAPR International joint conference on biometrics. https://doi.org/10.1109/BTAS.2014.6996259
25. NIST special publication 800-63B (2020). https://pages.nist.gov/800-63-3/sp800-63b.html. (Accessed on 27 July 2023)
26. Password vs. passphrase: differences & which is better? — Okta (2023). https://www.okta.com/identity-101/password-vs-passphrase/. (Accessed on 16 Dec 2023)
27. Strong passwords - IS&T contributions - Hermes (2021). https://kb.mit.edu/confluence/display/istcontrib/Strong+Passwords. (Accessed on 16 Dec 2023)
28. Killourhy K, Maxion RA (2008) The effect of clock resolution on keystroke dynamics, 331–350. https://doi.org/10.1007/978-3-540-87403-4_18
29. Risto HN (2023) Keystroke dynamics data set. https://doi.org/10.23642/usn.23790858

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.