Exploring Intentional Behaviour Modifications for Password Typing on Mobile Touchscreen Devices

Lukas Mecke, *University of Applied Sciences Munich, Munich, Germany, and LMU Munich, Munich, Germany;* Daniel Buschek and Mathias Kiermeier, *LMU Munich, Munich, Germany;* Sarah Prange, *University of Applied Sciences Munich, Munich, Germany, and Bundeswehr University Munich, Munich, Germany, and LMU Munich, Munich, Germany;* Florian Alt, *Bundeswehr University Munich, Munich, Germany*

https://www.usenix.org/conference/soups2019/presentation/mecke-behaviour

This paper is included in the Proceedings of the Fifteenth Symposium on Usable Privacy and Security.

August 12–13, 2019 • Santa Clara, CA, USA

ISBN 978-1-939133-05-2

Open access to the Proceedings of the Fifteenth Symposium on Usable Privacy and Security is sponsored by USENIX.

Exploring Intentional Behaviour Modifications for Password Typing on Mobile Touchscreen Devices

Lukas Mecke[1,2†], Daniel Buschek[2†], Mathias Kiermeier[2‡], Sarah Prange[1,3,2†], Florian Alt[3]
[1]*University of Applied Sciences Munich, Munich, Germany, {firstname.lastname}@hm.edu*
[2]*LMU Munich, Munich, Germany, †{firstname.lastname}@ifi.lmu.de, ‡mathias.kiermeier@gmail.com*
[3]*Bundeswehr University Munich, Munich, Germany,{firstname.lastname}@unibw.de*

Abstract

Behavioural biometric systems are based on the premise that human behaviour is hard to intentionally change and imitate. So far, changing input behaviour has been studied with the goal of supporting mimicry attacks. Going beyond attacks, this paper presents the first study on understanding users' ability to modify their typing behaviour when entering passwords on smartphones. In a prestudy (N=114), we developed visual text annotations to communicate modifications of typing behaviour (for example, gap between letters indicates how fast to move between keys). In a lab study (N=24), participants entered given passwords with such modification instructions on a smartphone in two sessions a week apart. Our results show that users successfully control and modify typing features (flight time, hold time, touch area, touch-to-key offset), yet certain combinations are challenging. We discuss implications for usability and security of mobile passwords, such as informing behavioural biometrics for password entry, and extending the password space through explicit modifications.

1 Introduction

The way we type on physical and on-screen keyboards is remarkably individual: Many studies have shown that people can be identified based on their typing rhythm [36], finger placement [11], and other such features of typing and touch behaviour [8, 37, 44]. This approach can be used, for example, to block unwanted access to technical systems, accounts, and personal mobile devices: Even if attackers gain knowledge of a password, they also have to enter it with the same behaviour as the legitimate user. The underlying assumption of such behavioural biometric authentication systems is that humans differ *implicitly* in how they type.

We present the first systematic exploration of a fundamentally different view: We study how users *explicitly* modify commonly utilised biometric features of their typing behaviour. Our goal in this paper is not to design a new authentication system but to better understand users' fundamental ability to control their typing behaviour. Better understanding such an ability to intentionally modify interaction behaviour is important in the light of a growing number of biometric security systems, as illustrated with the following use-cases:

Extending the password space: Instead of only using different characters to compose a password, each character could be entered in a different manner. For instance, although both use the same eight characters, "password" is different from "pass[*hold long*]word", where the user keeps the second "s" pressed for longer than her usual behaviour.

Avoid leaking "natural" behaviour: As more and more systems process behaviour, it might be a viable strategy for users to intentionally modify behaviour for some. For example, a user might authenticate on a work laptop using a modified typing rhythm when giving a presentation, to not reveal her "natural" typing behaviour, which she uses in (other) biometric systems, to a potential attacker. This strategy might also be used for authentication on the web or filling in a form in an unsafe environment, e.g., when using an unknown device.

Recovering from a leak of behavioural data: A leak of behavioural information implies that this biometric can no longer be used if we assume that behaviour is unchangeable. However, this is worth challenging. As an analogue example, some people decide to intentionally change the way they write their signature. Similarly, it might be possible to intentionally change, for example, password typing behaviour features to recover from a leak to be able to continue using this biometric.

In all these examples, users have reasons to intentionally modify aspects of their behaviour which they do not need to control for the underlying input method (e.g., typing rhythm does not matter for entering an email). Prior work on inten-

USENIX Symposium on Usable Privacy and Security (SOUPS) 2019.
August 11–13, 2019, Santa Clara, CA, USA.

tional changes of typing behaviour has exclusively studied this ability for attackers with technical support [4, 23, 24] or for limited features in desktop settings without changes and learning over time [14, 21, 33]. Thus, it still remains unclear to what extent users can control and modify fundamental biometric features of their mobile touch typing behaviour.

We address this gap by contributing: (1) Visual text annotations to communicate typing behaviour modifications, developed in a prestudy (N=114). (2) A lab study (N=24) using this scheme to investigate intentional modifications for different features and their combinations, for password typing on smartphones in two sessions a week apart. Based on the results, we discuss implications for mimicry attacks, research on behavioural biometrics, and usable passwords with intentional modifications.

The paper is structured as follows: After discussing related work (2), we develop a visualisation of typing behaviour (3), followed by our study design (4) and results (5) on intentional behaviour modifications. We conclude with a discussion (6).

2 Related Work

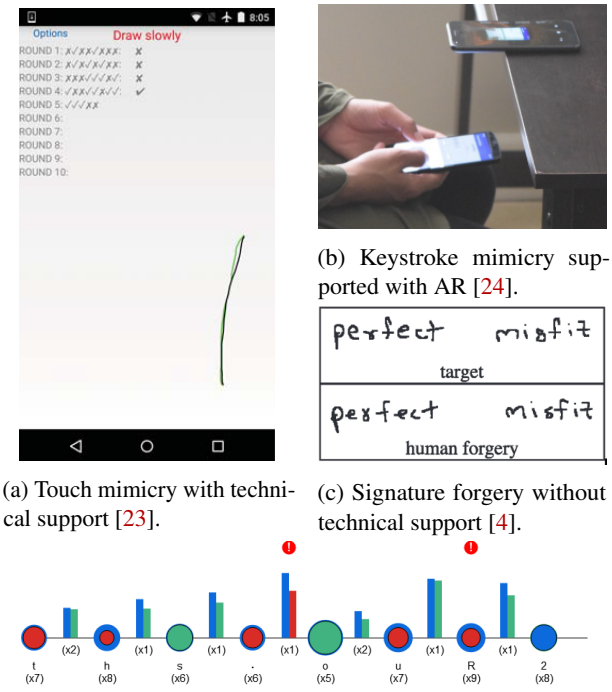
In this section, we relate our work to research on keystroke biometrics and mimicry attacks. These areas motivate our investigation of intentional modification of typing features and our choice of the specific features we studied.

2.1 Keystroke Biometrics

Our work is related to keystroke biometrics (or "keystroke dynamics"), which describe users' individual behavioural characteristics when entering text on a keyboard. This information can be used by the system to identify users, for example, to protect accounts, devices, and data. A rich body of related work examined this idea first for typing on physical desktop keyboards (for example, [29, 30]; survey [36]), then on early mobile phones with physical keys (for example, [7, 13, 15, 21, 22, 25, 46]). More recent work investigated keystroke biometrics for on-screen typing on smartphones (for example, [10, 11, 16, 44]; recent survey [37]), including keyboards operated via gestures instead of tapping [8].

For entering passwords in particular, recognising users based on *how* they enter the secret word provides an extra (implicit) layer of security [11], for example, to protect against cases in which the attacker got to know the password via shoulder surfing [32], smudge [2, 41] or thermal attacks [1].

Due to the origin of keystroke biometrics on physical desktop keyboards, the most commonly used typing behaviour features are temporal [36]: Users' typing is characterised by their typical *hold times* (i.e., time between key down and up event), and *flight times* (i.e., time between key up and down on the next key). Mobile touch devices offer further spatial features, such as touch area and offsets between touch locations and key centres. Offsets, in particular, showed higher



(a) Touch mimicry with technical support [23].

(b) Keystroke mimicry supported with AR [24].

(c) Signature forgery without technical support [4].

(d) Visualisation to support keystroke mimicry on a PC [38].

Figure 1: Several examples from related work for supporting mimicry attacks on (a) touch biometrics, (b, d) keystroke biometrics and (c) signatures. Images taken from cited papers.

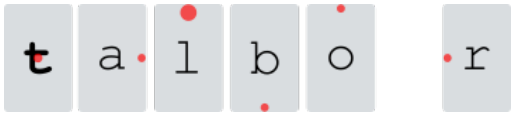
biometric value, that is, they facilitated more accurate distinction of users [10, 11]. Related work motivates our choice of features: hold time, flight time, offsets, and touch area.

In summary, related work on typing behavioural biometrics used features as they occur "naturally" as an *implicit* part of typing. Our work is fundamentally different: We examine these typing features as *explicit* and *actively controlled* by users, for example to increase the password space. In particular, we study how well users can indeed control these features when entering passwords on a smartphone.

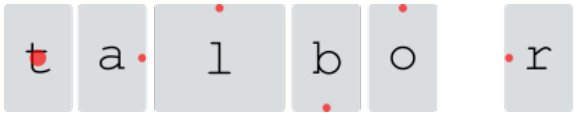
2.2 Mimicry Attacks

Attacks on keystroke biometric systems can be performed either automated or manually. Automated attacks use generative models to synthesise forgeries from observed data and were shown to be effective against handwritten signatures [4] and keystroke dynamics on a PC [28, 31, 34]. Some work also tested such attacks when proposing a new keystroke biometric system. For example, Stefan et al. found their system resistant against inputs generated from a first-level Markov model [35].

The most commonly considered attack on behavioural biometric systems is the so-called *mimicry* attack: Here, an impostor tries to manually reproduce (mimic) the (known) behaviour of a legitimate user to gain access.



(a) **'Bold Letter'** using bold font to indicate large touch area and circle size for hold time. Circle location shows offset, key gaps indicate flight time.



(b) **'Long Key'** using circle size for touch area and key width for hold time. Same as above: Circle location shows offset, key gaps indicate flight time.

Figure 2: Main design candidates for visualising target feature values for studying intentional behaviour modifications. Both were evaluated in our prestudy. Based on the results we decided to use the '*Long Key*' concept for our main study.

As a simple case, a *zero-effort attacker* model evaluates a biometric system against natural behaviour collected of other users who did not intend to actually bypass the system. While this model has been commonly used to evaluate vulnerability of behavioural biometric systems, related work found that it underestimates attack success [4, 31]. This calls for evaluations with means for more skilled and targeted attacks.

To support attackers in launching successful mimicry attacks they need to know the behaviour to imitate. In the case of handwritten text, for example, this could be a sample signature (cf. Figure 1–c). Researchers mounted successful mimicry attacks against touch input behaviour [23], keystroke dynamics on a PC [38], and keystroke dynamics on mobile phones [24].

Key to those attacks were systems which both visualise the target behaviour and provide the attacker with feedback on their attempts (cf. Figure 1). For example, Khan et al. [24] used augmented reality using a phone's camera to show visual cues on top of its view on another phone's keyboard. This guided correct timing and touch behaviour. In another approach they used audio stimuli to guide the timings.

In summary, prior work used representations and active modifications of typing behaviour to support mimicry attacks. In contrast, we aim to better understand the human ability to control mobile typing behaviour per se.

3 Prestudy: Developing a Visual Representation for Typing Behaviour Modifications

3.1 Selection of Features

There are a multitude of possible features that can be used for biometric authentication in the context of mobile touch interaction. An extensive list was compiled by related work [11] and covers 24 spatial, temporal and contact features. Khan et al. [24] found this extensive feature set hard to simultaneously control for their mimicry attack. They thus removed highly correlated features, resulting in a set of six: key hold time, flight time, down pressure, down area, down x, and down y.

We combine x and y together as touch offset. Furthermore, pressure and area were highly correlated on our test devices, since most Android phones[1] estimate pressure from area. We thus decided to omit pressure and used area directly.

To sum up, we decided to study a set of four features, namely *touch area*, *flight time*, *hold time* and *touch-to-key-offset* with the latter being two-dimensional (x, y).

3.2 Visualisation Design

We developed several designs that communicate modifications of the four features to instruct participants, for example, to perform a long key press for the second character in a password. We first tried simple markup (e.g., p– . **a̲s** . . sw̲–o̲**r** . d—) but found this representation to become cluttered quickly and to offer very limited expressiveness.

We thus chose a pictorial approach: We showed letters with a key metaphor to visualise behavioural changes (Figure 2). We explored a range of possible visual features, including offsetting the key or its label, writing bold or italic, and using underscores and coloured dots.

We narrowed the options down to two final designs (cf. Figure 2). Both used whitespace gaps between keys to indicate flight time and a red dot to indicate touch offset. One variant ('Bold Letter') visualised larger touch area by rendering the key in bold, and used the size of the offset dot to represent hold time. The other ('Long Key') used the size of the dot to visualise touch area, and key width to show longer hold time. While 'Bold Letter' resulted in a more compact format, 'Long Key' unified both temporal features on a shared axis (time flows from left to right). We conducted an online survey to determine our final design.

3.3 Online Survey

3.3.1 Survey Design and Procedure

To assess intuitiveness and readability of our designs, we created an online survey which showed example passwords with visualised modifications. Participants had to indicate which parts of the visualisation were used to encode which behavioural cues, without prior explanations. People did this for both designs in counterbalanced order. Afterwards, they were asked to rate on a 5-point Likert scale how intuitive and readable they found the two visualisations.

The survey was distributed over a university mailing list. It took 5 minutes to complete. Participants had a chance to win a €10 gift voucher.

[1] We used LG G6 phones in our study.

3.3.2 Results

A total of 114 participants answered our survey (56 % female; mean age 27 years, range 18 to 63 years). Both *offset* and *flight time* were correctly interpreted by 90 % of the participants for both designs. *Area* and *hold time* were correctly interpreted by 81 % and 82 % in the 'Long Key' condition, respectively. However, these two features were only correctly interpreted by 50 % and 51 % in the 'Bold Letter' condition. 'Long Key' was rated as more intuitive (median=agree, median_bold=neutral) but 'Bold Letter' was rated to be more readable (median=strongly agree, median_long=agree). When asked for their preferred method, 59 % of the participants reported the 'Long Key' notation while 39 % voted for the 'Bold Letter' visualisation. The rest had no preference.

3.4 Final Visual Representation

We decided to use the 'Long Key' visualisation: It has the advantage of encoding temporal features on a shared axis and all features allow for continuous representation of values (in contrast to the binary bold letter).

In conclusion, we used the following visual encoding shown in Figure 2–b: *Touch-to-key-offset* is marked by a red dot at the position where the key should be touched. *Flight time* is represented by a whitespace gap between two key rectangles that scales with duration. Analogously, *hold time* is represented by scaling the width of the key rectangle with duration. Finally *touch area* is visualised by the size of the red dot used for offset (larger size indicates larger area).

4 Main Study

4.1 Study Design

As our study design is quite complex, the following subsections each explain one main component. The most complex one is *task*, which is given both as an overview and in detail.

4.1.1 Passwords

In general, participants had to repeatedly enter given *passwords* ("football", "princess", "password"). While these three are obviously not great passwords in terms of security, we selected them since they have comparable properties and are common passwords[2]. Moreover, they do not require switching keyboard mode (e.g., between characters and symbols), which we wanted to avoid as a simplification for this first investigation into intentional typing behaviour modification. Similarly, we favoured simple passwords to ensure that task difficulty was mainly determined by behaviour variations and not affected by memorability or search time for rare symbols.

[2] https://www.teamsid.com/worst-passwords-2016/, last accessed 20.02.2019

4.1.2 Features

We studied intentional modification of four features: *touch-to-key-offset* (on five levels: centre/left/right/top/bottom), *flight time* and *hold time* (both on two levels: default/long), as well as *touch area* (on two levels: default/large).

4.1.3 Tasks

Participants solved 37 *tasks*, each using one of the three passwords. The tasks differed in various aspects, described below. While the design is complex, the overall goal was to cover six aspects, namely (1) different *passwords* with (2) different *feature modifications* at (3) different *locations* within each word. We also include (4) different *combinations* of features that are modified in the same password, either (5) at the same character/keypress (we call this *co-located*) or (6) *distributed* across several characters/keypresses within the word.

We iterated the task design several times by means of prestudy runs with two to three people in each version. We gradually narrowed the tasks down to an acceptable study duration of one hour. In full detail, the tasks used in the main study were structured and designed as follows (Figure 3):

Natural tasks (1–3): The first three tasks simply asked people to enter each password six times without presenting any intentional behaviour modifications.

Modifying a single feature (tasks 4–15): In each of these tasks participants had to modify one feature (e.g., hold time). There were three such tasks per feature, namely one per password (i.e., 4 features $\times$ 3 passwords = 12 tasks). Across the three tasks per feature, all feature levels occurred at least once, while covering different locations: The first task per feature modified the 2nd character of the password, the second task modified the 2nd and 7th characters, and the last task modified 2nd, 4th, and 7th characters. The assignment of passwords across these tasks was counter-balanced, such that modifications overall occurred in all passwords at all locations.

Modifying two features (tasks 16–27): In each of these 12 tasks people modified two features (for example, hold time *and* flight time). There were two tasks per combination of two features: The first had one modification on the 2nd character and the other on the 3rd (i.e., *distributed*). The second task had both modifications on the 7th character (i.e., *co-located*).

Modifying three features (tasks 28–35): In these eight tasks, participants had to modify three features, with two tasks per combination of three features: The first had modifications on the 2nd, 4th, and 7th character (*distributed*). The second one had all three modifications on the 5th character (*co-located*).

Modifying four features (tasks 36 and 37): Finally, participants had to modify four features: The first one had modifications on the 2nd, 4th, 6th, and 8th character (*distributed*), the last had all modifications on the 5th character (*co-located*).

The task order was not randomised, in favour of gradually increasing the number of modified features per password, which we suspected to have an influence on task difficulty.

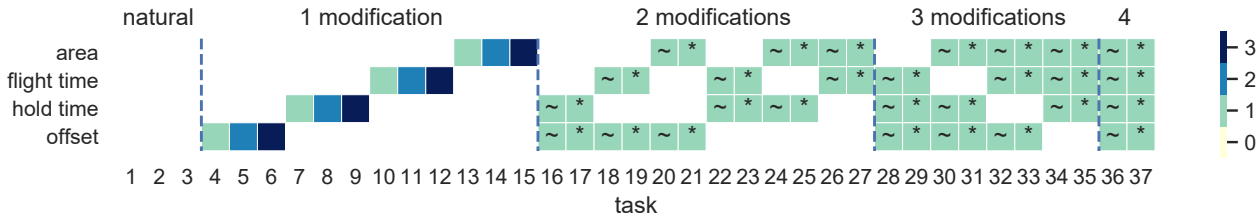


Figure 3: Overview over the tasks in each session. In the beginning (task 1–3) participants were asked to enter the passwords naturally, afterwards (task 4–15) a single feature had to be modified with increasing number of occurrences (colour of the cell). Thereafter, two (task 16–27), three (task 28–35) or four (tasks 36, 37) features had to be modified at once. All possible feature combinations were tested and features were either *distributed* (~) over the password or *co-located* (*) on a single key.

4.1.4 Sessions

The whole procedure was repeated two times, in two sessions about a week apart. In this way, we observed the typing behaviour of each participant at two points in time.

4.1.5 Summary

For the following report of our data analyses and results, it is useful to think of our study design as follows:

Tasks 1–3 are used to analyse natural (i.e., unmodified) behaviour, while the other tasks are used to analyse user behaviour when modifying the four behaviour features.

Note that from task 16 onward (i.e., all tasks with feature combinations), our study is a typical repeated measures design with: *number* of modifications (2, 3, 4) × *distributed* multiple modifications (distributed, co-located) × *session* (1st, 2nd). We use this for typical ANOVAs to study in particular the impact of modification of multiple features.

4.2 Apparatus

We developed an Android app that controlled the study process (e.g., counterbalancing, task progression, explanations).

The values used for scaling our visualisations (e.g., default flight time for default key gap) were informed by prestudy experiments and related work [10] (flight time 260 ms normal, 1000 ms long; hold time 80 ms normal, 300 ms long; area 0.2 normal, 0.4 large, unitless as reported by the Android API; offset x ±40 px, offset y ±70 px). To avoid visual clutter, we limited the scaling with minimum and maximum threshold values, beyond which the visualisation did not change.

We integrated a modified version of the Android open source project LatinIME[3] keyboard. This enabled us to log all typing events and touch features. To reduce distraction, we disabled the context menu for special characters shown on long press. In addition, our study app logged the expected key and behaviour modifications, as well as the current user and task for each keystroke.

[3] https://android.googlesource.com/platform/packages/inputmethods/LatinIME/, last accessed: 22.02.2019

4.3 Procedure

Upon arrival, participants were introduced to the goal of the study and were asked to sign a consent form to permit use of the collected data. After an initial demographics questionnaire they performed the tasks (cf. Figure 3) as described in section 4.1.3 on our test device. We asked participants to enter passwords with their right thumb to keep results comparable.

When first confronted with a new type of modification, participants got a short explanation of what to do and prior to every task they had the option to train entering the password. Except for the tasks without modifications (natural tasks) they were provided with real time feedback, using our visualisation, to show their behaviour next to the expected one. Every task had to be completed successfully six times and without feedback. The number of attempts was not limited.

Each task was followed by a short Likert questionnaire containing the statements: (1) *"I was able to adjust to the specified behaviour."*, (2) *"I was successful in completing the task."*, and (3) *"The task was difficult for me."*.

After completing all tasks, participants were asked to come up with a modified password on their own and could take notes to remember it. The same process was repeated in the second session, excluding the initial demographics questionnaire. Creating a custom password was replaced with recalling and performing the password from the previous session. After the second session we conducted a short interview. Sessions were scheduled one week apart.

4.4 Participants

Study invitations were distributed over a mailing list of our local university. Requirements were right-handedness and familiarity with typing on mobile phones. We recruited a total of 24 participants (14 female; mean age 27 years, range 14 to 54 years). Half of participants were in their twenties. 58 % were students, 30 % were employed, and the remaining ones were in school. Participants were compensated with €20 for completing the whole study.

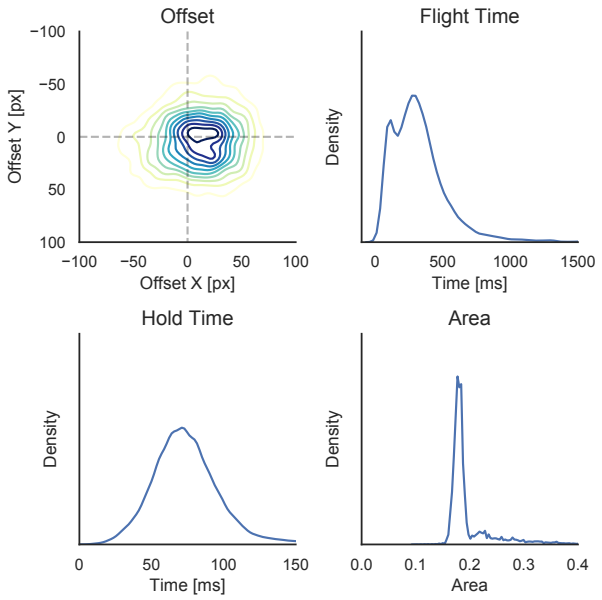


Figure 4: Overview of participants' natural typing behaviour (i.e., typing without being presented with any modifications), as measured in the first three tasks of each session.

5 Results

Significance tests were conducted using ANOVA with Greenhouse-Geisser correction and Bonferoni corrected post-hoc tests (significance at alpha level $p < 0.05$). If not reported otherwise, data for analyses is aggregated for both sessions.

As a first overview, we report key descriptive measures: The grand mean task completion time across all tasks(i.e., completing all six successful password entries of a task) and participants was 38.3 seconds . For typing speed, the grand mean was 28.7 words per minute (WPM [43]). The grand mean of the number of incorrect entries per task was 1.74.

We report on participants' natural typing behaviour (5.1), their ability to modify it (5.2), and their accuracy in doing so (5.3). We analyse the effect of multiple simultaneous modifications (5.4) and the impact of modifications on individuality of behaviour (5.5). We conclude with details on technically detecting modifications (5.6) and participant feedback (5.7).

5.1 Natural Behaviour

We first report on "natural" behaviour – typing *without* any modification instructions (tasks 1–3). Figure 4 presents the results. They match our expectations based on related work:

Touch offsets are slightly shifted to the lower right, as typical for input with the right thumb [9]. Moreover, median flight time (290 ms) and hold time (72 ms) are in line with related work [10] and close to the ones we chose as defaults for scaling key width and gaps in our visualisation (flight time 260 ms, hold time 80 ms). Thus, our chosen values indeed matched people's natural behaviour.

Feature	Measure	target	session	target * session
Offset	absolute x	$.777^a$		
	absolute y	$.890^a$		$.015^c$
	relative (error)	$.082^b$		
Flight time	absolute	$.785^a$		$.010^c$
	relative (error)	$.332^a$	$.038^b$	
Hold time	absolute	$.848^a$		
	relative (error)	$.624^a$		
Touch area	absolute	$.737^a$		
	relative (error)	$.930^a$		

a: $p < .001$, *b*: $p < .005$, *c*: $p < .05$, empty cells not significant

Table 1: ANOVA results for ability (1) to modify behaviour (absolute, Section 5.2) and (2) to replicate target feature values (relative i.e., error, Section 5.3). The last three columns show the effect sizes (ω^2) for *target* value (i.e., the feature value communicated via our text annotation), *session*, and their interaction. See text for results from post-hoc tests.

Touch area significantly correlated with x location of the target key (r=-0.252, p<.001): Due to thumb stretching, typing keys on the left of the keyboard resulted in a flatter thumb posture and thus larger touch area. Flight time showed a main and secondary peak (Figure 4). The latter was caused by zero finger travel distance for "double letters" (e.g., pa*ss*word).

5.2 Ability to Modify Behaviour

Figures 5 and 6 visualise the distribution of the behavioural features for different *target values*, i.e., expected feature values shown by our visualisation. Next, we report on statistical tests comparing these distributions per feature (see Table 1). Here we report on the post-hoc tests and further details:

For all features, post-hoc tests showed that directions of differences were as expected (e.g., offset significantly further to the left for *left*, flight time significantly longer for *long*).

For vertical offset and flight time, the interactions of session and target were significant (see Table 1), yet the small effect sizes and visual inspection of descriptive plots indicated that this was too tiny to warrant meaningful interpretation.

In summary, the significant results of these statistical tests confirm the "big picture" visible in Figure 5 and Figure 6: For all features, people significantly modified their behaviour in the direction indicated by our visualisation.

5.3 Ability to Replicate Target Feature Values

The previous section investigated differences in absolute feature values. It is also interesting to analyse how *accurately* people were able to replicate modifications. To this end, Figure 7 visualises the distribution of participants' errors when reproducing the target values indicated by our visualisation for each feature. Table 1 summarises the ANOVA results.

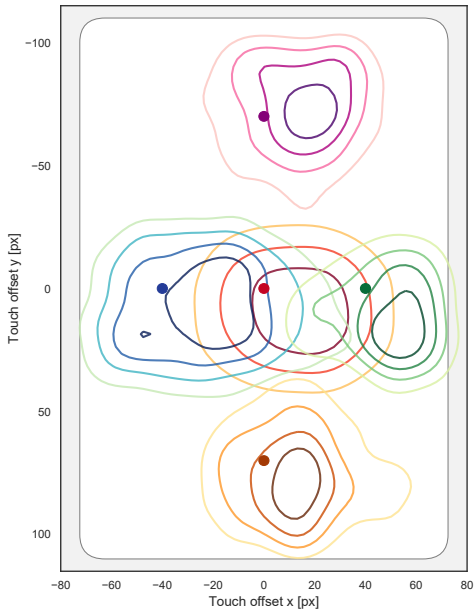


Figure 5: Overview of users' modified touch-to-key offsets: Provoking offset modifications resulted in clear differences in thumb placement. The rectangle indicates key borders.

For *offset*, post-hoc tests revealed errors to be significantly smaller for the target *right* compared to *left* (p=.010, d=-.773), *top* (p=.008, d=-.783), *bottom* (p=.011, d=-.765) and *default* offset (p=.027, d=-.685).

For *flight time*, we found errors to be significantly smaller for the *default* time than the *long* one (p<.001, d=-1.488), as well as for observations from the *second* session compared to the *first* (p=.004, d=-.645). The latter matches the observation that people typed slightly faster in the second session.

Regarding *hold time*, post-hoc tests showed errors to be significantly smaller for the *default* time compared to the *long* one (p<.001, d=-1.844). Finally, for *touch area*, we found errors to be significantly smaller for the *default* area size compared to the *large* one (p<.001, d=-4.470).

In summary, these results confirm that participants significantly modified their behaviour, namely towards the values indicated by our visualisation. In addition, people are more accurate in producing the default feature values compared to the more extreme ones, likely because the latter are further away from "natural" typing behaviour.

5.4 Impact of Modifying Multiple Features

Here we report on users' ability to modify multiple features in one password. Table 2 summarises the ANOVA results. Post-hoc tests and further details follow below.

5.4.1 Impact on Time, Speed, and Incorrect Entries

For *task completion time*, post-hoc tests revealed that three modifications resulted in significantly longer times compared to two (mean 40.70 s vs 36.36 s; p<.005, d=0.543); descriptively, this was also true for four modifications compared to two, yet not significantly so (p=.064). Moreover, distributed multiple modifications took significantly longer than co-located ones (mean 42.33 s vs 34.33 s; p<.01, d=1.397). People were also significantly slower in the first session than in the second one (mean 39.76 s vs 36.90 s; p<.05, d=0.444).

For *typing speed*, all pairwise comparisons of number of modifications were significant (all p<.001), with slower typing for higher numbers (mean 2: 30.18 WPM, 3: 27.33 WPM, 4: 25.15 WPM). Moreover, distributed multiple modifications were typed significantly slower compared to co-located ones (mean 26.91 WPM vs 30.46 WPM; p<.001, d=-2.445).

Finally, significantly more *incorrect password entries* occurred for distributed compared to co-located multiple feature modifications (mean 2.44 vs 1.45; p<.005, d=0.677).

These results show that users take significantly longer to enter passwords as the number of modified features increases, in particular if behaviour is modified for multiple features across different characters (i.e., *distributed*). In that case, people also produce significantly more incorrect password entries.

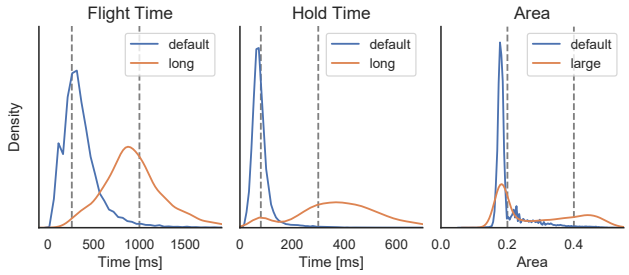


Figure 6: Overview of participants' modified typing behaviour across both sessions. Overall, this figure shows that presenting modifications via our visualisation provoked clear differences in the typing features (flight time, hold time, area; for touch offset see Figure 5). Vertical lines indicate the target values.

Measure	number of mod.	session	distributed	number * distributed	session * distributed
Offset error					
Flight time error	.93^a	.017^b	.109^a	.023^c	
Hold time error	.166^a		.178^a		
Touch area error	.039^a			.018^a	
Task compl. time	.032^c	.015^c	.224^a	.039^c	
Typing speed	.172^a		.232^a	.079^a	.002^c
Incorrect entries			.114^b		

a: p < .001, *b*: p < .005, *c*: p < .05, Empty cells not significant.

Table 2: Overview of ANOVA results for the impact of modifying multiple features on performance measures (Section 5.4.1) and ability to replicate target feature values (i.e., error, Section 5.4.2). Columns show effect sizes (ω^2) for *number* of modifications, *session*, and *distributed* multiple feature modification, plus interactions. See text for details.

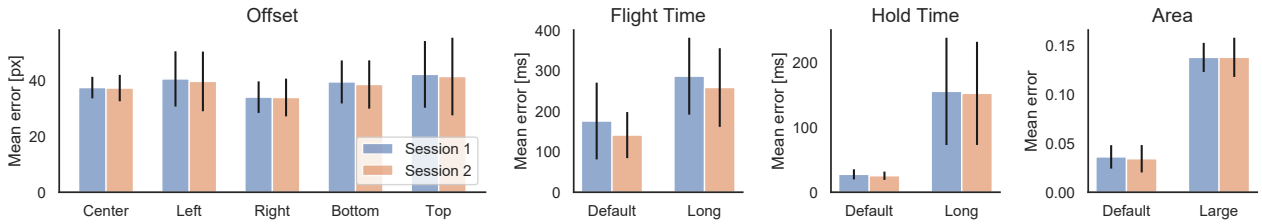


Figure 7: Observed derivation of participants' behaviour from the target values of the given modifications for both sessions. Participants were generally better at reaching the target value for the default level. For offsets, lowest error occurred for touches to the right, since this coincides with natural thumb offset [9]. In contrast to the other features, for flight time accuracy increased from the first to the second session, indicating a learning effect.

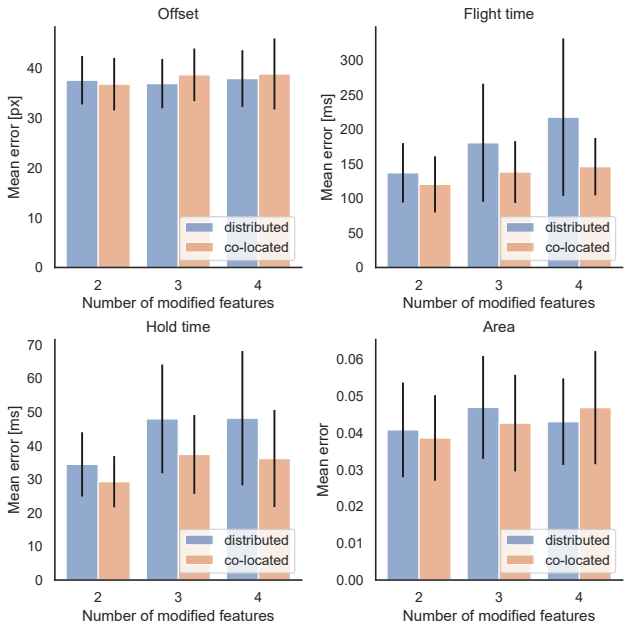


Figure 8: Participants' ability to replicate given behaviour depending on the number of features that had to be modified in one password and whether those features were co-located on a single key or distributed over the password.

5.4.2 Impact on Replicating Target Feature Values

Figure 8 shows participants' behaviour deviation from the given target behaviour (i.e., error), based on the *number* of features that had to be controlled within a single password and whether those features were *co-located* or *distributed*.

For *offset*, we found no significant effects (cf. very stable distribution of errors in Figure 8).

For *flight time*, errors were significantly lower for *co-located* modifications compared to *distributed* ones (p<.001, d=-.965), and for the second session compared to the first one (p=0.02, d=-.699). Regarding the number of modified features we observed significantly lower errors for two compared to three (p<.001, d=-1.149) and four (p<.001, d=-1.522), as well as for three compared to four modifications (p<.001, d=-.867).

Post-hoc tests for *hold time* revealed significantly lower errors for *co-located* features (p<.001, d=-1.004) and for two modified features compared to both three (p<.001, -1.479) and four (p<.001, d=-1.073) modifications.

Finally, for *touch area*, post-hoc tests showed significantly lower errors for two modified features compared to both three (p<.001, -1.565) and four (p<.001, d=-0.868) modifications.

Results are in line with the findings from the previous section. Participants generally performed better when features were *co-located* (i.e., not distributed over the password, Figure 8) and performance decreased for increasing *number* of modifications. Offset error was stable regarding all factors.

5.4.3 Impact on Subjective Rating

Participants answered three Likert items after each task: (1) *"I was able to adjust to the specified behaviour."*, (2) *"I was successful in completing the task."*, and (3) *"The task was difficult for me."* We compared users' ratings on these questions between tasks with co-located and distributed modifications: Wilcoxon signed-rank tests revealed significant differences for all three questions (Q1: Z=3.828, Q2: Z=4.074, Q3: Z= -3.765, all p<.001). Thus, participants subjectively perceived tasks with multiple feature modifications at the same character as significantly easier (i.e., better able to adjust behaviour, higher success, less difficult), compared to tasks with feature modifications distributed over several characters.

5.5 Impact of Modifications on Individuality

The previous analyses have shown behaviour differences *within users*, caused by modification instructions. Complementary, we now investigate how natural behaviour differences *between users* are influenced by modifications. This is interesting, for example, to inform behavioural biometric security layers. We will return to this in our discussion.

We thus compared the individuality (or "biometric value" [10]) of typing behaviour between natural and modified behaviour. To do so, we employed a user identification model [10, 12]. Note, that we do *not* intend to present this model as a practical biometric identification system. We rather

use it as an *analysis tool* to quantify the impact of explicit behaviour modifications on individuality. Thus, we are not interested in optimising identification accuracy, but in measuring the differences obtained on natural and modified behaviour.

5.5.1 Evaluation Scheme

We used the established Gaussian model for mobile touch typing, with a Gaussian distribution per feature per key [3, 19, 20, 45]. For touch location, for example, it defines the user's spread of touch points when aiming for that key. Thus, each user u is represented by a set of Gaussians (the model m_u), fitted to the touches from the training set for that user. We used the data from the first session to fit these models.

For each user u, we then fed the data from u's second session to this user's model m_u, which yields likelihoods for u (for an ideal model, these should be high). In particular, we computed the joint likelihood for all touches for each task t, that is, the likelihood that u is the one who typed the password in task t. Note that the features are per touch, not per password. Complementary, we fed the data from all other users $v \in U \backslash \{u\}$ to the model m_u as well (for an ideal model, these likelihoods should be lower). We repeated this for all pairs of users $u, v \in U$, such that we obtain 24 (user models) $\times$ 24 (user data) likelihoods per task. We repeated the whole analysis twice, once for natural and modified typing data.

On these likelihoods, we computed the standard measures for typing biometrics (e.g., see [10, 36]): receiver-operating-characteric (ROC) curve, area-under-curve (AUC), and equal error rate (EER). An EER of X% means that in X% of password entries the legitimate user would be incorrectly rejected while also X% of attacks would pass unnoticed.

5.5.2 ROC Analysis Results

Figure 9 shows ROC, AUC and EER. Compared to random guessing (dotted line, 0.5 AUC), both natural and modified typing clearly yield biometric information. The values are in line with related work using this model for password typing on smartphones with the right thumb in the lab [11]. The results also show that people retain aspects of their individual behaviour when asked to perform the same modifications.

The key observation is the *gap* between the curves in Figure 9. It quantifies the loss in individuality: To summarise, when measured using an established typing model, individuality of participants' typing behaviour was *reduced* by intentional behaviour modifications such that AUC dropped by .07 (relative -8.9 %) and EER increased by .06 (relative +20.7 %).

5.6 Detecting Modifications

Finally, we analysed how well behaviour modifications can be technically detected. This is important, for example, to build an authentication system that allows these modifications to be used as part of a password. For instance, to check a password like "pass[*hold long*]word", the system needs to be able to distinguish between normal and long hold times.

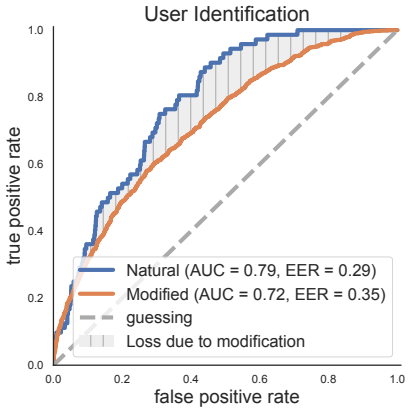


Figure 9: Impact of behaviour modification on individuality of typing behaviour, quantified by measuring the difference (shaded area) between ROC curves for user identification on natural (blue) vs modified (orange) typing. Typing behaviour becomes less individual through performing modifications. Clear individual characteristics remain, as evident from the modified (orange) line well above chance (dashed line).

We employed Random Forest classifiers with 100 trees and default parameters[4]. We used all typing features as input (hold time, flight time, area, offset x, y) and trained one model per modification (e.g., to classify normal vs long hold times).

We used leave-one-user-out evaluation across sessions: For each user u, we trained the classifier on the first session's data of all users except for u. We tested this model on u's data from session two. Thus, the model could be shipped pre-trained and would not require data collection during enrolment.

We report mean (std) classification accuracy over all users: hold time 97.9 % (1.36 %), flight time 96.14 % (1.84 %), area 94.71 % (1.16 %), and offset 94.29 % (0.96 %). Note, that the remaining error includes user errors (e.g., user accidentally performed normal instead of long hold time). For these user errors, the model has to give an incorrect classification.

These results demonstrate that modifications can be reliably detected. It is thus technically feasible to implement an authentication system that allows users to use these modifications as part of their password. We provide the model code and trained model as part of the material for this paper (see Section 8) to facilitate implementations and further research on such password systems.

5.7 User Feedback

After the study we conducted short interviews: Half of the participants (12) stated to be interested in using passwords

[4] https://scikit-learn.org/stable/modules/ensemble.html#forest, *last accessed 20.02.2019*

with behavioural modifications and four were strictly against it. The other eight had concerns (e.g., security, being able to reproduce their behaviour under different circumstances or technical feasibility of such a system), but stated they would be interested in using a system utilising intentional modifications if those concerns could be addressed.

Many participants said they struggled with offset modifications as they would often hit the wrong key. Some also had difficulties distinguishing large area and long hold time.

When creating passwords, users often first observed their natural behaviour to then emphasise it. For example, P20 stated: "*When I created the password I first typed it and observed what I automatically did. For example I typed a 'g' rather to the left, entered a 'b' rather [long]; That's what I adjusted [the password] to.*". Another common strategy was putting modifications at salient positions, such as at the beginning of words or syllables.

6 Discussion

6.1 Controlling Password Typing Behaviour

As a key insight, we revealed that people are able to significantly modify temporal and spatial features of their mobile typing behaviour in given directions. It is also possible to train a model that distinguishes between these features levels (e.g., default vs long press) with high accuracy (Section 5.6).

People were more accurate (i.e., deviated less from target feature values) in reproducing default values rather than extreme ones. We thus conclude that people are better at replicating behaviour that is close to their natural behaviour.

For flight time, accuracy was higher in the second week. We attribute this to people getting accustomed to our device, modifications, and tasks, indicating a learning effect.

In some cases participants performed default behaviour when expected to show a modification (see secondary peaks in distributions in Figure 6), likely due to the cognitive load of actively controlling their actions, especially when modifying multiple features. Controlling touch area is partly affected by the usage of the right thumb, which naturally leads to larger areas towards the left of the screen, due to stretching.

6.2 Modifying Multiple Behaviour Features

Overall, modifying an increasing number of behaviour features in a password becomes significantly more difficult to control. A possible explanation is the likely higher cognitive demand for intentionally modifying several aspects of typing behaviour, as supported by participants' comments and a higher number of incorrect inputs.

Specifically, modifying multiple features at different characters within one password ("distributed modification") is significantly more difficult than modifying multiple features at the same character ("co-located modification"). This conclusion is supported by all quantitative measures (task completion time, typing speed, incorrect entries, error measures), as well as participants' subjective Likert ratings.

Control of temporal features particularly suffers when other modifications are present, likely since focusing on those others distracts users from keeping the timing for the temporal modifications. Controlling spatial features is more robust.

In summary, our findings show that multiple features are harder to control when spread over multiple different characters; in particular, if temporal modifications are involved.

6.3 Methodology

We developed a visual text annotation scheme (Figure 2) to communicate target behaviour modifications. We chose this approach to be able to use text entry research's most common and established transcription task (i.e., enter given text) with our new concept of intentional behaviour modifications.

An alternative would have been to visualise desired feature values directly on the keyboard (e.g., show cross-hair on the key for offset modifications). However, this would have turned the task into a *reaction* exercise (i.e., hitting such cross-hairs), which likely leads to different behaviour. This approach also borrows heavily from the technical support work on mimicry attacks. Yet we were interested in users' ability to modify behaviour without such scaffolding. With our task, we thus gave clear instructions while participants were left to implement those modifications as they saw fit.

Future work could compare the two approaches. For example, work on systems for mimicry attacks could use our results here as a baseline for unsupported modification ability.

6.4 Deployment

As shown in Section 5.6, it is possible to reliably detect behaviour modifications, which enables building authentication systems that utilise them as part of a password. With backends that store passwords as hashes of strings, this could be easily integrated by inserting a special symbol depending on the preceding character's modification (e.g., "pass$holdlong$word" where $ stands for any character not allowed to be used directly for passwords in the system). Therefore, this technique can potentially be used in any context that passwords are currently used in – given that client software and hardware are capable of detecting modifications. For non-touch keyboards, only temporal features would be available.

Moreover, our visualisation (Section 3) could give users feedback on their typing, analogous to revealing entered characters in a password field on demand.

Finally, it is not clear how different devices and keyboard layouts influence behaviour and control, which could be investigated in future work.

6.5 Implications for Usable Passwords with Intentional Behaviour Modifications

Intentional behaviour modifications increase the space of possible passwords. We focused on the fundamental ability of users to control behaviour features. Our results offer plenty of opportunities for future work, e.g., investigating observability and memorability.We summarise practical recommendations for usable passwords with behaviour modifications:

Flight time, hold time, and touch-to-key offset present suitable behaviour features for intentional modification for password typing on smartphones. Modifications of touch area for thumb input should be avoided. Area is harder to control since it is partly determined by stretching of the thumb.

Flight time and hold time can be controlled on two levels (normal vs long). Offsets can be controlled on five levels though they were the most difficult modification for participants. We see several options to improve this for future work. This includes tolerance for miss-typing (i.e., accepting input that hits a neighbouring key in the direction of the executed modification) and using offset modifications only with larger keys (e.g., on tablets or for PINs). Modifying offsets may also be easier when typing with a different finger that allows for more precision (e.g., index). Modifying behaviour for one character in multiple ways should be favoured over distributing feature modifications across several characters. Combinations of feature modifications across multiple characters in particular for temporal modifications should be avoided.

Based on user feedback after creating own passwords, a promising creation strategy is to observe one's own natural behaviour and add emphasising modifications.

6.6 Implications for Mimicry Attacks

Related work [10, 11] found that spatial features (particularly offsets) have higher biometric value, that is, they lead to more accurate user identification, compared to temporal features. Our results show that it is difficult to intentionally modify multiple temporal features, or temporal features combined with others. In contrast, for modifying offsets, users are not inherently under time-pressure when controlling them.

We thus revealed a novel trade-off: Spatial features have higher biometric value than temporal ones in the literature, yet they might be easier for informed attackers to modify. Future work can investigate such mimicry attacks: In particular, our results suggest 1) to compare mimicry attacks on biometric systems that use either spatial or temporal features; and 2) to compare such attacks for "victims" that do or do not intentionally control these features as part of their passwords.

In contrast to most previous work on mimicry attacks, these new study ideas do not focus on technical support for attackers or specific protection methods, but rather on better understanding the fundamental human capabilities for copying and controlling otherwise uncontrolled input behaviour details.

6.7 Implications for Biometrics Research

We showed for the first time that when multiple people follow the *same* modification instructions, their mobile typing behaviour becomes less distinguishable (here relative +20.7 % equal error rate for user identification across sessions).

Earlier work on typing on desktop keyboards [14, 33] and phones with physical key pads [21] discussed "artificial rhythms" (e.g., inserting a pause), which *increased* biometric value, contradicting our results. This difference may be due to typing on touchscreens in our work and the fact that related work studied behaviour in one session only, ignoring changes over time. Moreover, users received "open" instructions to modify the rhythm as they liked and thus likely responded in more individual ways [33]. Typing biometrics for desktops can only utilise temporal features. In contrast, mobile touchscreens enable rich spatial features and it can be difficult to coordinate modifications of multiple features in one password entry. This might have caused less consistent behaviour across sessions, reducing accuracy of user identification.

On one hand, this suggests that authentication systems need to be careful with applying *both* behavioural biometrics (e.g., as an extra security layer) and intentional modifications (e.g., for extended password space). On the other hand, suggesting *different* modifications to different users could improve biometric value, as we find users able to follow modifications of the most important features in typing biometrics.

Other work examined related ideas that might be investigated in our context as well: (1) nudging users towards creating more diverse lock patterns via subtle visual cues [40]; and (2) facilitating user exploration of "original" behaviour [42].

Our results guide future work on the idea of provoking more diverse behaviour: For example, a future study could ask users to set up a password not only with composition instructions (e.g., minimum length), but also suggest (random) behaviour modifications for how to enter it. Based on our results, we expect to achieve higher biometric value in this way, compared to 1) suggesting no behaviour modifications, or 2) suggesting the same modification to all users.

6.8 Security Considerations

Using intentional behaviour modifications impacts password capture and guessing attacks [6]. *Capture attacks* like smudge attacks [2] may be deflected, as temporal features leave no marks. Video-based attacks like shoulder surfing [32] or thermal attacks [1] may still be possible, though potentially harder, as extracting exact timings may prove difficult and fingers occlude the concrete touch points as long as no feedback is given (compare 6.4). Phishing may only be successful if the interface can capture and transmit modifications.

Assuming random passwords and modifications, adding modifications makes both online and offline *guessing attacks* harder (Table 3). Including one modification adds up to about

password length	8	7	6	5
no modifications	**49.36**	43.19	37.02	30.85
1 modification	**55.14**	48.77	42.38	35.94
2 modifications	59.84	**53.27**	46.63	39.90
3 modifications	63.90	57.10	**50.20**	43.16

Table 3: Entropy (*bits*) of random passwords with and without (random) modifications on an alphabet of 72 characters (upper and lower case letters, numbers and 10 special characters).

5 bits of entropy (calculations in Appendix A). Thus, modifications may enable shorter passwords maintaining similar entropy. For instance, under the given assumptions, an eight character password can be reduced to six characters when using exactly 3 modifications. This is promising as passwords on mobile devices tend to be weaker and harder to enter [27].

Notice that these are upper bounds; there may be common patterns of choosing modifications, which reduce theoretical entropy in practice (e.g., participants reported to choose beginnings of words or syllables for modifications, cf. Section 5.7). Moreover, focusing modifications on a single key instead of spreading them out makes guessing easier. However, our calculations assume that the attacker knows the exact number of modifications, thus (slightly) underestimating entropy. While suggesting concrete modifications might solve some of those drawbacks it may introduce usability issues. We suggest practical security as an area for future work.

6.9 Limitations

We examined a limited set of typing features with a commonly used keyboard app (modified Google open source keyboard). We did not measure pressure or shape features from the full capacitive image (cf. [26]). Nevertheless, we covered the most commonly used temporal and spatial typing biometrics features (cf. [36, 37]), found to be the most important ones among a larger set for mobile password typing [11].

To avoid an impact of password complexity we chose a limited set of easy passwords for our study. Our findings may not generalise to more complex passwords.

To keep an acceptable study duration, we only observed one-handed use with the right thumb. This is one of the most considered postures in research [5, 17, 18, 45] and one of the most frequently used ones in daily life [10]. All participants were right-handed and used to this posture. Future studies could compare our results to typing with the index finger.

During analysis of the results we noticed that the target behaviour in task 34 contained an additional hold time modification instead of the intended flight time modification. Thus the combination of area, hold and flight time was not tested.

Our sample is biased towards younger people and might not represent the overall population. Finger precision and timing might change with age (cf. [39]). Future work could compare our results to samples with children and older adults.

7 Conclusion

Typing behaviour can be analysed to identify users based on features such as typing rhythm [36] and finger placement [11]. So far, research had studied these features as they occur "naturally" as an implicit, uncontrolled part of typing, or in the context of supporting mimicry attacks with technical means.

This paper addresses the gap in the literature with the first study on users' ability to intentionally modify their behaviour when typing passwords on smartphones: We developed a novel visual text annotation in a prestudy (N=114), before using it to study intentional modifications in the lab (N=24).

Overall, our results reveal that users can successfully modify the features most commonly used in typing biometrics systems for smartphones. This fundamental insight has several implications for users, threat models, and biometrics research. We conclude by outlining some of them here:

It is worth investigating further the idea of using intentional modifications as a part of passwords. This could extend the password space (e.g., "password" vs "pass[*hold long*]word") and possibly also reduce observability, as attackers would have to guess the modification, not just the entered word.

Our results also motivate novel research directions for touch and typing biometrics systems: These might suffer from "standardizing" typing behaviour across users with given modifications, as revealed in our study. However, nudging different users to use different modifications in turn promises to increase user identification accuracy (cf. [40]).

Related, threat models for evaluating such biometric systems need to take into account that some target behaviours are inherently more difficult to attack: In particular, our results strongly motivate comparing attacks that require modifying temporal vs spatial features to mimic the victim's behaviour.

Overall, we show the rich capabilities of users to intentionally control typical input behaviour features previously considered as an implicit "information byproduct" of interaction. With this work, we hope to spark new research and discussion regarding the use of behaviour-aware security systems that go beyond the view of a passively analysed user to take into account these human capabilities.

8 Project Resources

Material for this paper is available at: https://www.unibw.de/usable-security-and-privacy/downloads/datasets/intentional-behaviour-modifications

Acknowledgements

Work on this project was partially funded by the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B). This research was supported by the Deutsche Forschungsgemeinschaft (DFG), Grant No.: AL 1899/2-1.

References

[1] Yomna Abdelrahman, Mohamed Khamis, Stefan Schneegass, and Florian Alt. Stay cool! understanding thermal attacks on mobile-based user authentication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3751–3763. ACM, 2017.

[2] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[3] Tyler Baldwin and Joyce Chai. Towards online adaptation and personalization of key-target resizing for mobile devices. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, IUI '12, pages 11–20, New York, NY, USA, 2012. ACM.

[4] Lucas Ballard, Daniel Lopresti, and Fabian Monrose. Forgery quality and its implications for behavioral biometric security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1107–1118, 2007.

[5] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1991–2000, New York, NY, USA, 2014. ACM.

[6] Robert Biddle, Sonia Chiasson, and Paul C Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR)*, 44(4):19, 2012.

[7] A. Buchoux and N. L. Clarke. Deployment of Keystroke Analysis on a Smartphone. In *Australian Information Security Management Conference*, 2008.

[8] Ulrich Burgbacher and Klaus Hinrichs. An implicit author verification system for text messages based on gesture typing biometrics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2951–2954, New York, NY, USA, 2014. ACM.

[9] Daniel Buschek and Florian Alt. TouchML: A machine learning toolkit for modelling spatial touch targeting behaviour. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, New York, NY, USA, 2015. ACM.

[10] Daniel Buschek, Benjamin Bisinger, and Florian Alt. ResearchIME: A mobile keyboard application for studying free typing behaviour in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 255:1–255:14, New York, NY, USA, 2018. ACM.

[11] Daniel Buschek, Alexander De Luca, and Florian Alt. Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1393–1402, New York, NY, USA, 2015. ACM.

[12] Daniel Buschek, Alexander De Luca, and Florian Alt. Evaluating the influence of targets and hand postures on touch-based behavioural biometrics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1349–1361, New York, NY, USA, 2016. ACM.

[13] P Campisi, E Maiorana, M Lo Bosco, and A Neri. User authentication using keystroke dynamics for cellular phones. *IET Signal Processing*, 3(4):333–341, 2009.

[14] Sungzoon Cho and Seongseob Hwang. Artificial rhythms and cues for keystroke dynamics based authentication. In David Zhang and Anil K. Jain, editors, *Advances in Biometrics*, pages 626–632, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[15] Nathan L Clarke and Steven M Furnell. Authenticating mobile phone users using keystroke analysis. *International journal of information security*, 6(1):1–14, 2007.

[16] Benjamin Draffin, Jiang Zhu, and Joy Zhang. Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction. In Gérard Memmi and Ulf Blanke, editors, *Mobile Computing, Applications, and Services*, pages 184–201, Cham, 2014. Springer International Publishing.

[17] Mayank Goel, Leah Findlater, and Jacob Wobbrock. Walktype: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2687–2696, New York, NY, USA, 2012. ACM.

[18] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. Contexttype: Using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2795–2798, New York, NY, USA, 2013. ACM.

[19] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. Language modeling for soft keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 194–195, New York, NY, USA, 2002. ACM.

[20] Asela Gunawardana, Tim Paek, and Christopher Meek. Usability guided key-target resizing for soft keyboards. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10, pages 111–118, New York, NY, USA, 2010. ACM.

[21] Seong-seob Hwang, Sungzoon Cho, and Sunghoon Park. Keystroke dynamics-based authentication for mobile devices. *Computers & Security*, 28(1–2):85–93, 2009.

[22] Sevasti Karatzouni and Nathan Clarke. Keystroke analysis for thumb-based keyboards on mobile devices. In Hein Venter, Mariki Eloff, Les Labuschagne, Jan Eloff, and Rossouw von Solms, editors, *New Approaches for Security, Privacy and Trust in Complex Environments*, pages 253–263, Boston, MA, 2007. Springer US.

[23] Hassan Khan, Urs Hengartner, and Daniel Vogel. Targeted mimicry attacks on touch input based implicit authentication schemes. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 387–398. ACM, 2016.

[24] Hassan Khan, Urs Hengartner, and Daniel Vogel. Augmented reality-based mimicry attacks on behaviour-based smartphone authentication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 41–53. ACM, 2018.

[25] Emanuele Maiorana, Patrizio Campisi, Noelia González-Carballo, and Alessandro Neri. Keystroke dynamics authentication for mobile phones. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pages 21–26, New York, NY, USA, 2011. ACM.

[26] Sven Mayer, Huy Viet Le, and Niels Henze. Estimating the finger orientation on capacitive touchscreens using convolutional neural networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, pages 220–229, New York, NY, USA, 2017. ACM.

[27] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Usability and security of text passwords on mobile devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 527–539. ACM, 2016.

[28] John V Monaco, Md Liakat Ali, and Charles C Tappert. Spoofing key-press latencies with a generative keystroke dynamics model. In *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2015.

[29] Fabian Monrose, Michael K. Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, pages 73–82, New York, NY, USA, 1999. ACM.

[30] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 48–56, New York, NY, USA, 1997. ACM.

[31] Khandaker A Rahman, Kiran S Balagani, and Vir V Phoha. Snoop-forge-replay attacks on continuous verification with keystrokes. *IEEE Transactions on Information Forensics and Security*, 8(3):528–541, 2013.

[32] Florian Schaub, Ruben Deyhle, and Michael Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 13:1–13:10, New York, NY, USA, 2012. ACM.

[33] Seong seob Hwang, Hyoung joo Lee, and Sungzoon Cho. Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication. *Expert Systems with Applications*, 36(7):10649 – 10656, 2009.

[34] Abdul Serwadda and Vir V Phoha. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM Transactions on Information and System Security (TISSEC)*, 16(2):8, 2013.

[35] Deian Stefan, Xiaokui Shu, and Danfeng (Daphne) Yao. Robustness of keystroke-dynamics based biometrics against synthetic forgeries. *Computers & Security*, 31(1):109–121, February 2012.

[36] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue. A Survey of Keystroke Dynamics Biometrics. *The Scientific World Journal*, 2013, 2013.

[37] Pin Shen Teh, Ning Zhang, Andrew Beng Jin Teoh, and Ke Chen. A survey on touch dynamics authentication in mobile devices. *Computers & Security*, 59(C):210–235, 2016.

[38] Chee Meng Tey, Payas Gupta, and Debin Gao. I can be you: Questioning the use of keystroke dynamics as biometrics. In *Annual Network and Distributed System Security Symposium 20th NDSS*, pages 1–6. Research Collection School Of Information Systems, 2013.

[39] Radu-Daniel Vatavu, Lisa Anthony, and Quincy Brown. Child or adult? inferring smartphone users' age group from touch measurements alone. In Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2015*, pages 1–9, Cham, 2015. Springer International Publishing.

[40] Emanuel von Zezschwitz, Malin Eiband, Daniel Buschek, Sascha Oberhuber, Alexander De Luca, Florian Alt, and Heinrich Hussmann. On quantifying the effective password space of grid-based unlock gestures. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, MUM '16, pages 201–212, New York, NY, USA, 2016. ACM.

[41] Emanuel von Zezschwitz, Anton Koslow, Alexander De Luca, and Heinrich Hussmann. Making graphic-based authentication secure against smudge attacks. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 277–286, New York, NY, USA, 2013. ACM.

[42] John Williamson and Roderick Murray-Smith. Rewarding the original: Explorations in joint user-sensor motion spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1717–1726, New York, NY, USA, 2012. ACM.

[43] Jacob O. Wobbrock. Measures of text entry performance. In *Text Entry Systems: Mobility, Accessibility, Universality*, chapter 3, pages 47 – 74. Morgan Kaufmann, 2010.

[44] Hui Xu, Yangfan Zhou, and Michael R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 187–198, Menlo Park, CA, July 2014. USENIX Association.

[45] Ying Yin, Tom Yu Ouyang, Kurt Partridge, and Shumin Zhai. Making touchscreen keyboards adaptive to keys, hand postures, and individuals: A hierarchical spatial backoff model approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2775–2784, New York, NY, USA, 2013. ACM.

[46] Saira Zahid, Muhammad Shahzad, Syed Ali Khayam, and Muddassar Farooq. Keystroke-Based User Identification on Smart Phones. In *LNCS*, volume 5758, pages 224–243, 2009.

A Calculating Entropy of Modified Passwords

For a random password with no modifications of length n on the alphabet Σ we calculate entropy E as:

$$E_0 = log_2(|\Sigma|^n)$$

For one modification we choose a password first and then add a single modification at a random location. There are 7 possible modifications (assuming that one manifestation of each feature would be the default (e.g., pressing keys in the centre). Finally we exclude the single case where a flight time would be applied to the first character (as it does not have a preceding character to measure flight time from). This yields:

$$E_1 = log_2(|\Sigma|^n \cdot (7n-1))$$

Analogous, we calculate the entropy for two modifications by choosing a password first and then either applying two modifications on one character (15 options) or two single modifications; again excluding cases where a flight time modification would be applied to the first character.

$$E_2 = log_2(|\Sigma|^n \cdot \underbrace{((15n-6)}_{\text{2 on one}} + \underbrace{(\frac{7n \cdot 7(n-1)}{2} - 7(n-1))))}_{\text{2 single}}$$

We calculate entropy for three modifications analogously, taking into account the possibility of three modifications on one character (line 1), two modifications on one character combined with a single modification (line 2) and three single modifications (line 3):

$$\begin{aligned} E_3 = & log_2(|\Sigma|^n \cdot ((13n-9) \\ & +(15n \cdot 7(n-1) - 57(n-1)) \\ & +(\frac{7n \cdot 7(n-1) \cdot 7(n-2)}{6} - \frac{7(n-1) \cdot 7(n-2)}{2})) \end{aligned}$$