CREDENTIAL HARDENING BY USING TOUCHSTROKE DYNAMICS

PIN SHEN TEH
School Of Computer Science, University Of Lincoln,
Ln6 7ts, United Kingdom
email: anakinteh@gmail.com

ANDREW BENG JIN TEOH
School Of Electrical And Electronic Engineering, Yonsei University, Seoul 120-749,
South Korea.
&
Predictive Intelligence Research Cluster, Sunway University, Bandar Sunway,
47500, Selangor, Malaysia.
email: bjteoh@yonsei.ac.kr

SHIGANG YUE
School Of Computer Science, University Of Lincoln,
Ln6 7ts, United Kingdom
email: syue@lincoln.ac.uk

ABSTRACT

Today, reliance on digital devices for daily routines has been shifted towards portable mobile devices. Therefore, the need for security enhancements within this platform is imminent. Numerous research works have been performed on strengthening password authentication by using keystroke dynamics biometrics, which involve computer keyboards and cellular phones as input devices. Nevertheless, experiments performed specifically on touch screen devices are relatively lacking. This paper describes a novel technique to strengthen security authentication systems on touch screen devices via a new sub variant behavioural biometrics called *touchstroke dynamics*. We capitalize on the high resolution timing latency and the pressure information on touch screen panel as feature data. Following this a light weight algorithm is introduced to calculate the similarity between feature vectors. In addition, a fusion approach is proposed to enhance the overall performance of the system to an equal error rate of 7.71% (short input) and 6.27% (long input).

Keywords: touchstroke dynamics; authentication; keystroke dynamics; biometrics.

INTRODUCTION

Background

Improvements in internet technology have led to an increased dependence on computers and internet enabled communication devices in our daily lives. Consequently, the amount of sensitive data or information transmitted over the internet between different forms of digital devices has been greater than ever. As a result there is a need for a strong authentication mechanism to act as a shield to protect the confidentiality of valuable information from others attempting to obtain malicious access.

For more than three decades, a vast amount of research has been conducted to enhance the security of computer systems via keystroke dynamics (wang, guo, & ma, 2012). Due to the rapid evolution of portable mobile devices, new ways of securing authentication on these devices are essential to ensure that data and information are well protected. Furthermore, due to the replacement of a physical keypad and keyboard with touch screen panels on portable mobile devices, conventional feature data extracted from physical keyboards might become obsolete. In this paper we utilized the richness of feature data extracted from a high-end mobile tablet to enhance security authentication on touch screen devices.

Motivation And Contribution

The majority of research into keystroke dynamics has focused on workstation (bleha, slivinsky, & hussien, 1990)(obaidat, 1995) or a web-based environment (s. Cho, han, han, & kim, 2000)(stewart, monaco, cha, & tappert, 2011), with recent years seeing increases on the mobile platform (clarke & furnell, 2007a)(mcloughlin & naidu, 2009)(cunha urtiga & moreno, 2011). Despite the increases on the mobile platform, the research has focused on keystroke dynamics on mobile devices with a physical keyboard or keypad. Research addressing keystroke dynamics with high-end touch screen devices, using a screen keyboard as the input mechanism, is lacking.

The use of information such as typing pressure (saevanee & bhatarakosol, 2008) as part of the keystroke feature to increase recognition accuracy has been criticized for its impracticability. This is due to the need for additional customization of existing acquisition devices, which contradicts the major advantage of keystroke dynamics biometrics; the ability to utilize existing hardware for cost effectiveness. However, as high-end mobile portable gadgets become more widely available, this issue may no longer be a deterrence for exploiting pressure feature data. The increased sensitivity level of modern devices may be able to provide better quality and unique feature representations, which may in turn boost recognition performance.

Public datasets exist on keystroke dynamics biometrics however data acquisition was performed on a normal computer keyboard (r. Giot, el-abed, & rosenberger, 2009)(allen, 2010). It appears that there is no benchmark dataset available for data extracted from a touch screen device. The vast amount of time and resources required for the creation and collection of live data may account for the lack of open datasets (r. Giot et al., 2009). Since smart devices are experiencing rapid growth and increased usage, it is likely that more research will be carried out on this platform in the near future. It is therefore preferable to have a common dataset in order for comparisons to be made between different research algorithms on the same input data and experimental settings.

The objectives and contributions of this paper can be summarized as follows:

1. Propose a novel touchstroke dynamics authentication system.
2. Exploit timing and pressure feature of a touch screen device as unique feature data.
3. Create a benchmark dataset using a touch screen device to be shared.
4. Propose a simple fusion approach that is able to enhance recognition performance.
5. Compare the performance differences between short and long numeric input data.

Related Work

Gaines et al. (gaines, lisowski, press, & shapiro, 1980) was among the earliest research conducted in the area of keystroke dynamics. The study recorded the times between successive keystrokes of seven professional secretaries typing a given paragraph of prose. Following this, later research used distance measure (garcia, 1986) (young & hammon, 1989) and minimal distance (bleha et al., 1990) as classifiers. The research in this area has gained more attention with the employment of probabilistic approaches, for example, bayes classifier (t.-h. Cho, 2006), hidden markov model (montalvao, almeida, & freire, 2006), and gaussian model (p. Teh, teoh, tee, & ong, 2011). Statistical methods like k-nearest neighbour (saevanee & bhatarakosol, 2008) and degree of disorder (gunetti, picardi, & ruffo, 2005) have also received much attention among keystroke dynamics researchers, generally, the primary rationale for the popularity of the statistical method is the algorithm's simplicity, which correlates with low computational requirements and minimal system overheads.

In an attempt to strive for higher authentication accuracy, researchers have used more sophisticated methods such as machine learning approach. A popular method being used is artificial neural network, which is able to approximate parameters under different conditions with no exact knowledge of all the contributing variables (pavaday & soyjaudah, 2007). Although neural network algorithms are more often than not computationally expensive (alexandre, 1997), several studies have been reported to achieve superior results (obaidat, 1995)(de lima e silva filho & roisenberg, 2006). As compared by statistical techniques, it has even been argued that neural networks are capable of producing a superior performance. (crawford, 2010). A major limitation of this methodology, however, has been the deprivation to the systems response time and power consumption, which is crucial on low powered mobile devices. Other techniques studied include decision tree (maxion & killourhy, 2010), fuzzy logic (de ru & eloff, 1997), support vector machine (romain giot, el-abed, hemery, & rosenberger, 2011) and evolutional computing (karnan & akila, 2009).

Since research began in the field of keystroke dynamics biometrics, studies have been primarily carried out on computer terminals where the common input acquisition device is the conventional qwerty keyboard. Keystroke timing latency is the immediate feature data available through the keyboard. A major advantage of keystroke dynamics biometrics is that feature data can be obtained entirely through software implementation without the addition of costly hardware.

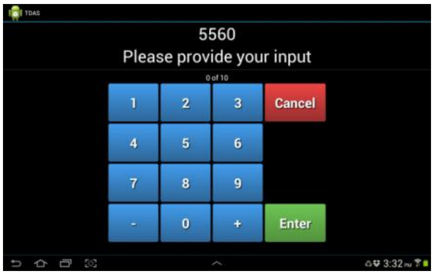
In an effort to enhance the richness of keystroke dynamics, typing pressure has been introduced as an additional feature (nonaka & kurihara, 2004). Although encouraging results have been reported, with the integration of pressure sensitive receivers, a number of the experiments required specific customization to existing input devices. Consequently, the inability to apply the devices to real world scenarios might explain the lack of research in this area. On the other hand, research has been carried out introducing artificial rhythms and cues in an attempt to improve the uniqueness of each user's typing pattern without using a method that requires modification to an acquisition device (s. Cho & hwang, 2005). However, it reduces the usability of the system as it increases the user's burden for memorizing extra information separate to the usual secret word.

It was not until the early 21st century, when the development of mobile devices rapidly increased, that researchers started to explore the possibility of implementing keystroke dynamics on the mobile platform. Clarke and furnell (clarke & furnell, 2007b) were among the first researchers to explore the possibility of integrating keystroke dynamics in a modified nokia 5110 cellular phone. By using feed forward multi-layered perceptron promising results were obtained with numeric inputs. Although the performance of mobile headsets might suffer from the extra computational power required by neural network algorithms, the result inspired more research on cellular phones (mcloughlin & naidu, 2009)(hwang, cho, & park, 2009)(campisi, maiorana, lo bosco, & neri, 2009). Despite research on early generation smartphones (nauman & ali, 2010), the results reported are preliminary and limited. Furthermore, to date, no research has been carried out on high-end touch screen devices.

METHODOLOGY

Acquisition Device And Development Platform

The input device used in this study was the samsung galaxy tab 10.1 (gt-p7510), a sophisticated digital tablet with a 1ghz dual-core processor running under android platform 4.0.4 (ice cream sandwich). The data collection prototype, as shown in
Figure 1, was developed using android api level 15. Each touch event (finger touching down and releasing from the touch screen panel) timestamp was recorded by calling the api function *system.nanotime()*, which is capable of returning the most precise timer available on the device's system. Since it is unlikely that a human's tapping speed can reach such a high pace, timing latency was normalized to the desired resolution upon feature extraction (to be discussed). We also utilized the api function under the motionevent class *getsize()*, to retrieve the value of the circumference in which the screen was being pressed. This function returns a normalized decimal value within the range of [0 1].


5560

Please provide your input

0 of 10

Figure 1. Screenshot of data collection prototype.

Each completed touch event on a key generated two timestamps (t_{down} and t_{up}) and a pressure size (*ps*) event as depicted in Figure 2. These three events played a major part in the template generation process. Raw touchstroke data inclusive of each key down and key up (k_{down} and k_{up}) of every input repetition (r) were recorded and stored in a separate file for each user with the format given below:

$$\{[r], [k_{down}], [t_{down}], [k_{up}], [t_{up}], [ps]\} \quad (1)$$

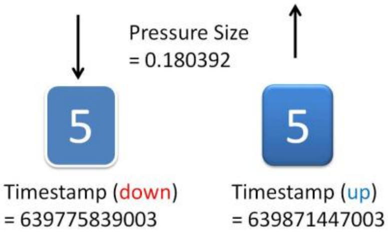


Figure 2. Sample timestamps and pressure size from the touch event on a key.

Input String

Users were required to provide two types of input string during the enrolment process. Firstly, a short 4 digit string (“5560”) that resembles a typical pin number used at most atm and debit or credit card payment machines. Secondly, a longer 16 digit string (“1379666624680852”) that emulates a conventional credit or debit card number used during online transactions. To diversify the variety of the input string, the fixed numerical strings, rather than being selected at random, were carefully designed to include different key positioning combinations. A brief explanation of the four different combinations is as follows:

- Apart – keys that are separated by at least one key apart.
- Repetition – consecutive reoccurrence of an identical key.
- Adjacent – keys that are located diagonally to each other.
- Sequence – keys that are situated horizontally or vertically next to each other.

Imposing a universal and predetermined input string for every user offers a significant advantage of increasing the total number of imposter samples available for comparison, during the testing phase, without collecting this data explicitly.

Device Freedom and Experimental Control

The entire data collection process was performed on a digital tablet. The justification of using a predefined device instead of user specific device was to remove uncontrolled variables such as device familiarity, program compatibility and functionality differences. As such, the result obtained reflects the discriminative power of touchstroke feature data and the classification algorithm used.

During the enrolment process, users were required to repeat each input string ten consecutive times. Any input error made was automatically discarded and the user was prompted to repeat that particular input. Users were neither strictly monitored nor were they required to perform the data collection process within the constraints of a laboratory environment. The aim was to provide users with flexibility and comfort during the experiment in an attempt to extract their natural touchstrokes.

Subject Demographic

Previous research has collected data from subjects in the area of academia (undergraduate, postgraduate, researcher and lecturer) to produce the datasets (kotani & horii, 2005)(lee & cho, 2007)(ngugi, tremaine, & tarasewich, 2011). The current study attempted to involve users from outside of academia from a range of professions and age distributions in order to reflect larger sample population diversity. A total of 50 subjects willingly participated in the experiment. Table 1 shows a summary of the information on the dataset that was collected.

Table 1. Summary of Experimental Dataset.

Property	Details
User Size	50
Input Type	“5560” (Short), “1379666624680852” (Long)
Repetition	10 sample for each input type
Gender	12 Male, 38 Female
Age	8(<20), 27(20-40), 15(>40)
Usage Frequency	16 Rare, 12 Average, 22 Often
Hand Preference	5 Left, 45 Right
Population	9 Academia, 41 Public

Feature Extraction

The key purpose of feature extraction is to extract vital information from the raw touchstroke data collected during enrolment stage for use later in the template generation process. In this study, we utilized touchstroke timing latency and pressure information as feature data. Timing feature data can easily be obtained by manipulating the high-resolution timestamp recorded during the enrolment stage.

The timing data extracted is similar to that of keystroke dynamics, whereby they can be sub divided into *Dwell Time* (DT) and *Flight Time* (FT) respectively. DT (also known as press time or hold time) is the time difference between touch events (press or release) of the same key. FT (also known as latency) is the time interval between two successive touch events of different keys. Further analysis found that FT can be separated into four subcategories as shown in Figure 3. The formula to calculate each type of timing feature data can be summarized as follows:

$$DT_k = R_k - P_k \qquad (2)$$

$$FT_{1,k} = P_{k+1} - R_k \tag{3}$$

$$FT_{2,k} = R_{k+1} - R_k \tag{4}$$

$$FT_{3,k} = P_{k+1} - P_k \tag{5}$$

$$FT_{4,k} = R_{k+1} - P_k \tag{6}$$

P and R refer to the timestamp of pressing down and releasing a key respectively, while k indicates the position of the intended feature data.

Note that in our experiment, we included the touch event for the “Enter” key as part of the input feature, as we believe that this key provides crucial touchstroke characteristics, which is distinct among different users. Therefore, the “Enter” key will form part of the input string length. Since it is impractical for user’s to press at such high resolutions (nanoseconds), the measurement of timing data was reduced to 0.1 milliseconds by multiplying each extracted feature data by the factor of 10^{-5}. This empirical value was based on a series of iterative tests depending on which multiplication factor returned the best recognition performance.

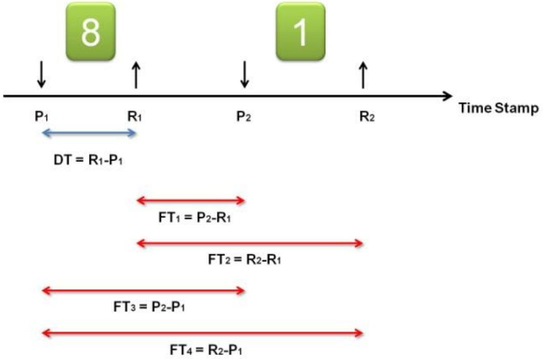


Figure 3. Different types of touchstroke timing feature data.

Whilst there was a possibility that FT_1 would result in a negative value (occurring when the successive key is pressed before releasing the previous) for the feature captured using a conventional keyboard (Sheng, Phoha, & Rovnyak, 2005), this was unlikely to happen on touch screen inputs. The reason for this is due to the difference in physical and geometrical size of on-screen input keys against computer keyboards. It was therefore not likely that a user would use all ten fingers whilst they provided their inputs. As a result, the chances of pressing the next key before releasing the previous was significantly reduced or in some cases eliminated.

The extraction of pressure feature data is a more direct approach in comparison to timing data. The value of pressure size (*PS*) is used directly from the return decimal value of an Android’s touch event API function without further manipulation. Since this measurement is used to quantify the area of depression of a key there will only be one pressure size value for each key interaction. Consequently, the total amount of possible vector elements generated for each type of feature data, DT (V_{DT}), FT (V_{FT}) and PS (V_{PS}) is dissimilar depending on the input length (n) as shown in the formula below:

$$V_{DT} = \{DT_1, DT_2, \cdots DT_n\} \quad (7)$$

$$V_{FT} = \{FT_1, FT_2, \cdots DT_{n-1}\} \quad (8)$$

$$V_{PS} = \{PS_1, PS_2, \cdots PS_n\} \quad (9)$$

Template Generation

In this stage, training samples of feature data extracted were combined and stored as template profiles. These templates, which uniquely represent each user, were later used in the authentication phase. Among the ten repeated input samples collected in the enrolment stage, seven were used for training and the remaining for testing. The order of distribution of training and testing samples were selected randomly. Based on a training sample set of *n* number of DT's, DT_1, DT_2, . . . , DT_n, the mean and standard deviation of the set are defined as below, respectively:

$$\mu = \frac{\sum_{i=1}^{n} DT_i}{n} \quad (10)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} DT_i^2 - \frac{\left(\sum_{i=1}^{n} DT_i\right)^2}{n}}{n-1}} \quad (11)$$

Consequently, a user's touchstroke template for the feature type DT, which consists of a total number of *k* feature vector element (where *k* depends on the length of input string as discussed in the prior section) was kept in the following format:

$$[\mu_{DT,1}, \cdots, \mu_{DT,k}], [\sigma_{DT,1}, \cdots, \sigma_{DT,k}] \quad (12)$$

A template data was generated for each of the six feature data types as mentioned in the previous section. The example above shows template generation for DT alone, subsequently, the same process and calculation was carried out on all the other five types of feature data.

Feature Matching

Unlike workstations with an unlimited power supply, mobile portable devices have limited battery power. Taking this into account the current study used the Gaussian Estimator Function (GEF) for feature matching. This method calculates the likeliness score of a test sample against the reference template with minimal computational overhead. As a result, the possibility of extensive power consumption, which leads to the reduction in battery operational time, was minimized. The mean (μ) and standard deviation (σ) from the reference template and the test sample value (d) is required to generate the similarity score (S) of a feature vector element of position (i), using the formula shown as follows:

$$S_i = e^{-\frac{(d_i-\mu_i)^2}{2\sigma_i^2}} \tag{13}$$

The output score produced by GEF will always be within the range of [0 1]. The further away a score from the value of 1, the lower the likelihood that the test sample belongs to the reference template. In other words, the possibility that the test sample belongs to an imposter is greater, and vice versa. Figure 4 illustrates the example of two different GEF scores (0.1353 and 0.6065) computed from two different test sample feature data elements (300 and 600) respectively. It is evident that the nearer the test data is to the mean (500) of the reference template (mid of graph), the higher the similarity score obtained, and vice versa.

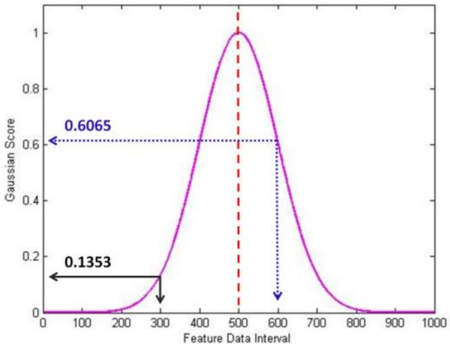


Figure 4. The graph illustration of GEF Score of two different feature data element values.

One by one, matching was performed on each individual element within the intended feature vector in order to obtain separate similarity scores for each feature element compared. These scores were then compared to an empirical threshold (thr) to establish a partial decision (pD_i) for feature data element of position (i) in that feature vector as shown below:

$$pD_i = \begin{cases} 0, & S_i \leq thr \\ 1, & S_i > thr \end{cases} \tag{14}$$

0 and 1 indicate rejection and acceptance respectively. A cumulative decision was then obtained to determine if a test sample belonged to the reference template via simple majority voting scheme, shown as follows:

$$D_{final} = \begin{cases} reject, & \frac{\sum_{i=1}^{k} pD_i}{k} < 0.5 \\ accept, & \frac{\sum_{i=1}^{k} pD_i}{k} \geq 0.5 \end{cases} \tag{15}$$

where k refers to the total elements in the feature vector considered, while D_{final} is the final acceptance or rejection decision of a given test sample.

Feature Data Fusion

As discussed in the previous section, the feature matching module allows the option of utilizing six possible types of feature data (five timing data and one pressure data) to calculate a user's touchstroke similarity. Using these features individually, PS recorded the best performance as compared to the other timing features, the results of which will be presented in subsequent sections of this study. It was found, rather than using each feature in isolation, by merging the decision made based upon different feature data types overall performance was improved.

This outcome was made possible as fusion utilizes information from more than one feature data source. Since different feature data types have unique properties in representing a user's touchstroke characteristic, the combination of feature data types complement one another**.** Therefore, the extra information generated facilitates in the discrimination between imposter and genuine user. A visual explanation of this concept can be seen in the graphical illustration in Figure 5.

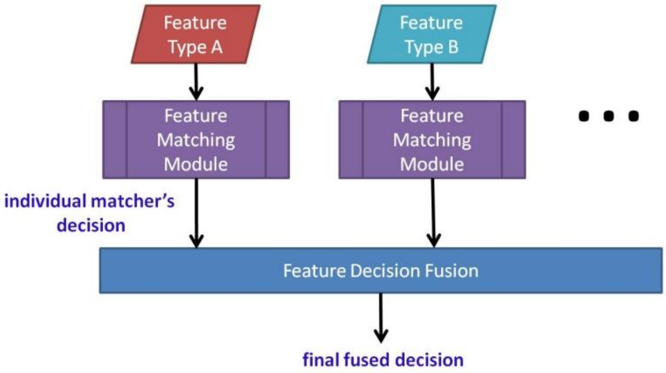


Figure 5. Diagram shows the process flow of the proposed feature data fusion approach.

A simple decision level fusion approach was adopted in our experiment, so that additional computational strain to the resource limited mobile device was minimized. Using AND voting rule, and individual decisions, made by independent feature matching module (tested against different feature data type) were merged together. The number of matching modules involved was based upon the amount of feature data types intended to be fused, ranging from two to six (all possible feature data type extracted). As a result, a given test sample was deemed genuine only if the decision of each feature matching modules remained true.

EXPERIMENTAL RESULTS AND DISCUSSIONS

Experimental Setting

In this experiment, seven out of the ten samples of input strings were used as training sets, while the remaining three were used for testing. The seven training samples were converted into a user template profile in the template generation phase as discussed previously. The sample selection order was randomized and the number of training and testing phases was based upon a sequence of iterative tests depending on which combination yielded the best recognition performance.

In the False Acceptance Rate (FAR) test, the first user touchstroke template was matched against all the other users' touchstroke testing samples. This process was reiterated for all users' touchstroke templates. Subsequently, the total number of impostor attempts conducted was $50 \times (50 - 1) \times 3 = 7350$.

As for the False Rejection Rate (FRR) test, a user's touchstroke template was matched against their own testing touchstroke samples respectively. After repeating the same matching process for all the users, the total number of genuine attempts stands at $50 \times 3 = 150$.

Input Length

In terms of input string length, researchers have often argued that using a longer string is key to performance improvements of keystroke dynamics authentication system. As discussed in the earlier section, two sequences of distinct numbers with the lengths of 4 and 16 digits were chosen as the representation of short and long input strings respectively. Referring to the experimental result plotted in Figure 6, it is apparent that, regardless of the feature data type in use, a longer input string produces better result compared to a shorter input string.

The probable reason for this result is that as more input strings are involved, feature data samples within each input string, and the number of different chunking combinations (breaking up longer inputs into smaller subsets for easier memorization) grows simultaneously. As a result, the ability to better represent a user's touchstroke cadence also increases. In addition, the amount of imposter feature data samples required to match that of a genuine reference template is also significantly increased. Therefore, such a stringent condition contributes to a better recognition performance by reducing the imposter pass rate.

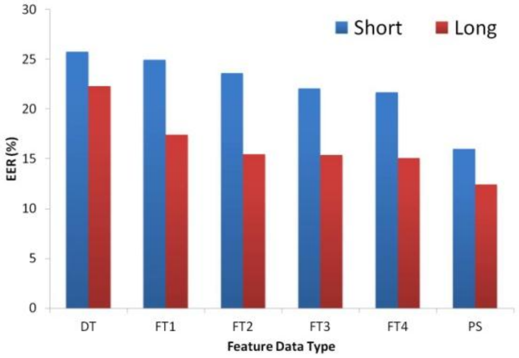


Figure 6. Graph illustrates the performance comparison between short and long numeric input string.

Feature Data

In this paper, two different kinds of feature data were extracted to represent a user's touchstroke dynamics; 1) timing, which can be sub-divided into DT and four variants of FT, and 2) pressure information. Experiments were conducted on each and every feature data type separately in order to identify which is better at discriminating a user's touchstroke pattern. The individual performances of each feature data type tested for both short and long input strings are summarized in Table 2.

It can be clearly seen from Table 2 that PS performs the best, followed by FT and then DT. A possible explanation for the performance of PS may be due to the way pressure size is extracted. Providing input on a touch screen panel, including the user's choice and arrangement of fingers along with the amount of force asserted on the screen, establishes the value of pressure size. The combination of these properties forms a pattern that is found to be unique for different users. It is also noticed from Figure 6 that the performance difference between short and long input strings is minimal when PS is selected as the feature data. This provides evidence in favour of pressure feature in comparison to timing feature data.

Table 2. Result Comparison among Individual Touchstroke Feature Data Type.

	Short	**Long**
Feature	**EER (%)**	
DT	25.78	22.29
FT_1	24.93	17.42
FT_2	23.61	15.45
FT_3	22.08	15.40
FT_4	21.70	15.07
PS	**15.98**	**12.44**

In terms of timing feature data, the results of all four kinds of flight time are better than dwell time. This result indicates that the amount of time a user's finger travels between different digits contains more discriminative power than the duration of each key hold. This argument can be further supported by the different ways in which users construct chunking patterns while providing their input string. Figure 7 demonstrates the three different versions of the many possible chunking patterns of a given sample input. It has been observed that a user does not usually complete the entire input string in one go, instead, inputs are broken into small subsets (chunks), which results in a minor pause or delay in between. The information that is contained within the natural short pauses between chunks of input can be associated with and increase the uniqueness of the flight time of a given string. It is particularly effective for longer string lengths, due to the increased number of possible grouping permutations. Therefore, it can be seen in Table 2, that regardless of the FT tested, 16 digit inputs always achieve better results than 4 digit inputs.

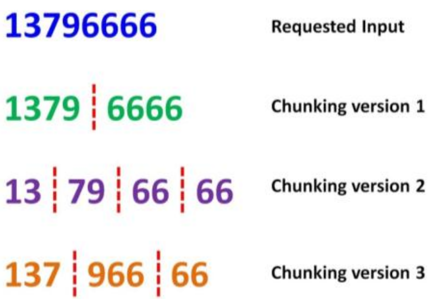


Figure 7. Diagram illustrating several different versions of chunking combinations.

Due to the absence of physical buttons and the limited geometrical size, it is normally more difficult for a user to type on an on-screen keyboard. Therefore, the variation of dwell time amongst different users is smaller than that of a computer keyboard. Thus, the results are less distinctive when dwell time is used instead of flight time, as reflected in the performance reported in Table 2.

Feature Combination

In the search for better authentication results we proposed a simple fusion scheme which accumulates individual's decisions made by GEF tested on different feature data types to produce a more precise authentication conclusion. To start with two feature data types with the best individual performance (PS and FT_4) are combined, and then the number of feature data type gradually increases one by one, according to the next best result.

The performance improvement after the fusion of different number of feature combinations is summarized in Table 3. For the sake of a simple comparison, the results shown for non-fusion (only one feature data type used) is based only on the EER of PS, where it obtained the best result when used independently among all six feature data types. As predicted an increase in performance can be seen as more features are combined. Remarkably, the best result was achieved when all feature data types are used at an EER of 7.71% and 6.27% for both short and long inputs respectively, which converts into a significant increment of approximately 50% in comparison to the use of PS alone (best result among single feature data type).

Table 3. The Performance Gained after Fusion of Different Number of Feature Data Type.

	Short		**Long**	
# of Feature	**EER (%)**	**±(%)***	**EER (%)**	**±(%)***
1	15.98	-	12.44	-
2	10.14	36.53	8.12	34.74
3	9.71	39.25	7.86	36.76
4	9.82	38.53	7.71	38.02
5	9.52	40.40	8.05	35.23

6	**7.71**	**51.77**	**6.27**	**49.56**

*indicates performance differences against result of non-fusion.

As discussed earlier, in terms of performance, DT is the worst feature data type when used independently. However, when used in conjunction with the other five feature data types, it boosts authentication performance further, as shown in the final data point of Figure 8. By observing Figure 8, it is notable that authentication performance is at all times significantly better with fusion. This indicates that as more feature information is used in decision-making, the ability to accurately distinguish between a genuine and an imposter user is improved.

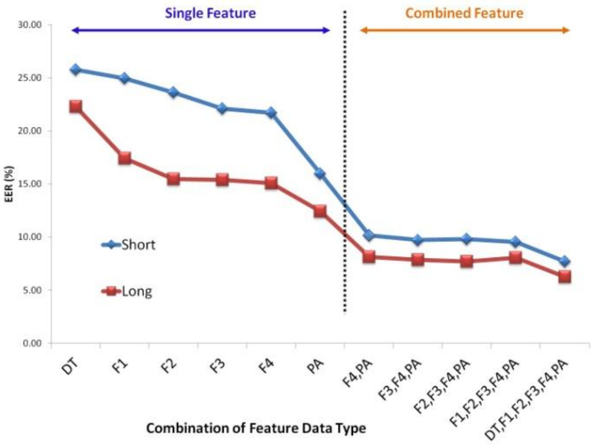


Figure 8. Result comparison between individual vs. combination of more than one feature data type.

Comparison with Other Techniques

In this section, the discussion is focused on the research that has the most similar platform and experimental settings as the study within this paper. They are research works conducted on mobile devices and experiments involving numerical input data that extracts pressure vector as feature data.

Unlike this study, which uses a more recent high-end digital tablet, the majority of the studies in keystroke dynamics use a cellular phone with a built-in numerical keypad as the common acquisition device. Such devices are normally equipped with low processing power, and the only extractable feature data that does not require modification to the devices hardware is typically timing information.

Although research works by (Clarke & Furnell, 2007b) and (Nauman, Ali, & Rauf, 2011) attempted to use a more sophisticated neural network algorithm to classify a user’s keystroke pattern, the performance reported was not much better in comparison to simpler methods, such as statistical technique and distance measure, as summarized in Table 4. Therefore, not only do these algorithms result in reduced performance improvements, they impose a more intensive computational requirement and may degrade authentication speed, especially within the resource limited mobile platform.

Table 4. Comparison with Existing Research Work on Mobile Platform.

Study	**Subjects**	**Input**	**Feature**	**Method**	**Device**	**FAR (%)**	**FRR (%)**	**EER (%)**
(Clarke & Furnell, 2007b)	32	11 digit	DT,FT$_1$	Neural Network	Nokia 5110	-	-	12.8
(McLoughlin & Naidu, 2009)	3	6 digit	DT,FT$_3$	Statistical	Renesas H8S-2377			90*
(Hwang et al., 2009)	25	4 digit	DT,FT$_1$	-	SAMSUNG SCH-V740	-	-	4
(Zahid, Shahzad, Khayam, & Farooq, 2009)	25	250 char	DT,FT$_1$	Fuzzy Logic	Nokia (Symbian)	2	0	-
(Campisi et al., 2009)	30	10 char	DT,FT$_1$,FT$_2$,FT$_3$	Statistical	Nokia 6608	-	-	13
(Cunha Urtiga & Moreno, 2011)	15	8 digit	DT	Distance	Nokia 5200	12.97	2.25	-
(Maiorana, Campisi, González-Carballo, & Neri, 2011)	40	10 char	DT,FT$_1$,FT$_2$,FT$_3$	Distance	Nokia 6680	-	-	14.74
(Nauman et al., 2011)	-	password	DT,FT$_1$	Neural Network	HTC G1	-	-	86.1*
This Paper	**50**	**4 digit**	**DT,FT$_1$,FT$_2$,FT$_3$,FT$_4$,PS**	**Statistical**	**Samsung Galaxy Tab 10.1**	**10.08**	**5.33**	**7.71**
		16 digit	**DT,FT$_1$,FT$_2$,FT$_3$,FT$_4$,PS**			**7.21**	**5.33**	**6.27**

*performance in terms of accuracy, similar but inverse to EER where value closer to 100% indicates better performance.

It is evident from both Table 4 and Table 5 that researchers are looking at numerical base keystroke dynamics authentication addresses either short or long digit strings, however does not compare both simultaneously within the same experimental setting. Based on the lack of research, we designed our data collection protocol to contain both 4 and 16 digit input strings. A clearer performance comparison can be deduced by assessing both short and long input strings simultaneously. It was found within the current study that longer inputs have a greater discriminatory power.

Despite (Saevanee & Bhattarakosol, 2009) achieving extraordinary performances of EER 1%, only 10 subjects were involved in the experiment. The small amount of subjects makes it difficult to draw any concrete conclusion (Alexandre, 1997). To date, the dataset within the present study involved more subjects than any other research in the area of keystroke dynamics on a mobile platform.

Recognizing the potentially distinct properties that reside within pressure information a number of research works have attempted to extract pressure information as a form of feature data. The majority of the acquisition devices used in these experiments, however, required additional modification to existing equipments as shown in Table 5. Such modifications pose restrictions to the larger scale applicability in the physical world. Since the input device used in the current experiment is a well known and widely available digital tablet, such a limitation is less likely to be significant.

Table 5. Comparison of Research Works which Incorporates Pressure as Feature Data on Numerical Input.

Study	**Subjects**	**Digit**	**Timing Feature**	**Method**	**Modification**	**FAR (%)**	**FRR (%)**	**EER (%)**
(Kotani & Horii, 2005)	9	4	DT,FT$_1$	Statistical	Yes	-	-	2.4
(Martono, Ali, & Salami, 2007)	5	6	FT$_1$	SVM	Yes	0.95	5.6	-
(Saevanee & Bhatarakosol, 2008)	10	10	DT,FT$_1$	Clustering	No	-	-	1
(Leberknight, Widmeyer, & Recce, 2008)	-	4	DT	Distance	Yes	-	-	-
(Grabham & White, 2008)	30	4	DT,FT$_1$	Statistical	Yes	15	0	10
(Saevanee & Bhattarakosol, 2009)	10	10	DT,FT$_1$	Neural Network	No	-	-	1
This Paper	**50**	**4**	**DT,FT$_1$,FT$_2$,FT$_3$,FT$_4$**	**Statistical**	**No**	**10.08**	**5.33**	**7.71**

		16	**DT,FT$_1$,FT$_2$,FT$_3$,FT$_4$**			**7.21**	**5.33**	**6.27**

CONCLUSION AND FUTURE WORKS

In this paper, we have attempted to use touchstroke dynamics to enhance numerical authentication on a touch screen device. This is a sub variant behavioural biometrics similar to that of keystroke dynamics, which has yet to be explored on the smartphone or tablet platform. As shown in the experimental results, pressure feature data has proven to be the most effective in uniquely representing user touchstroke cadence.

Although newly developed mobile gadgets possess much greater processing abilities compared to that of their predecessors, battery consumption rate by application remains to be an issue. As a result of this, the computational intensity of feature matching algorithms applied on this platform should be kept minimal. Nevertheless, our proposed GEF, if compared against neural network algorithms, imposes lower computational strain, offers faster authentication speed and consumes less battery.

Based on previous studies, it is reported that as more information is combined, higher authentication accuracy can be obtained (Hai-Rong Lv & Wen-Yuan Wang, 2006)(P. S. Teh, Yue, & Teoh, 2012). This can be seen in our experimental results, whereby the combination of more than one kind of feature data results in an improved performance. Nevertheless, in this paper, only one method (GEF) is employed for feature matching. Therefore, we plan to incorporate fusion approach along with several other classifiers, in an attempt to improve performance and validate this claim.

Compared to previous keystroke dynamics research works conducted on the mobile platform, the current experiment involves the largest number of subjects. In addition to this, the dataset is based on a diverse group of subjects, of different ages, with a range of device usage frequencies. The performance difference between short and long numerical inputs was also thoroughly examined. It was found that longer input strings had a more superior recognition performance.

At present, we are expanding our dataset sample size with the hope of increasing the existing amount to at least 100 users and more. The targeted user group will continue to be from outside the university population, from a broad age distribution and a diverse touch screen usage frequency. We are also committed to sharing the collected resources with the research community in the near future.

REFERENCE

Alexandre, T. J. (1997). Biometrics on smart cards: An approach to keyboard behavioral signature. Future Generation Computer Systems, 13(1), 19–26. doi:10.1016/S0167-739X(97)00005-8

Allen, J. D. (2010, July). An analysis of pressure-based keystroke dynamics algorithms. SOUTHERN METHODIST UNIVERSITY. Retrieved from http://gradworks.umi.com/14/77/1477849.html

Bleha, S., Slivinsky, C., & Hussien, B. (1990). Computer-access security systems using keystroke dynamics. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 12(12), 1217–1222.

Campisi, P., Maiorana, E., Lo Bosco, M., & Neri, A. (2009). User authentication using keystroke dynamics for cellular phones. Iet Signal Processing, 3(4), 333–341. doi:10.1049/iet-spr.2008.0171

Cho, S., Han, C., Han, D. H., & Kim, H.-I. (2000). Web-Based Keystroke Dynamics Identity Verification Using Neural Network. Journal of Organizational Computing and Electronic Commerce, 10(4), 295–307.

Cho, S., & Hwang, S. (2005). Artificial Rhythms and Cues for Keystroke Dynamics Based Authentication. In D. Zhang & A. Jain (Eds.), Advances in Biometrics (Vol. 3832, pp. 626–632). Springer Berlin / Heidelberg. Retrieved from http://www.springerlink.com/content/y3051wg5625j1631/abstract/

Cho, T.-H. (2006). Pattern Classification Methods for Keystroke Analysis. In SICE-ICASE, 2006. International Joint Conference (pp. 3812 –3815). doi:10.1109/SICE.2006.314667

Clarke, N. L., & Furnell, S. M. (2007a). Advanced user authentication for mobile devices. Computers & Security, 26(2), 109–119. doi:10.1016/j.cose.2006.08.008

Clarke, N. L., & Furnell, S. M. (2007b). Authenticating mobile phone users using keystroke analysis. International Journal of Information Security, 6(1), 1–14. doi:10.1007/s10207-006-0006-6

Crawford, H. (2010). Keystroke dynamics: Characteristics and opportunities. In Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on (pp. 205–212).

Cunha Urtiga, E. V., & Moreno, E. D. (2011). Keystroke-Based Biometric Authentication in Mobile Devices. Latin America Transactions, IEEE (Revista IEEE America Latina), 9(3), 368–375.

De Lima e Silva Filho, S., & Roisenberg, M. (2006). Continuous Authentication by Keystroke Dynamics Using Committee Machines. Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/11760146_90

De Ru, W. G., & Eloff, J. H. P. (1997). Enhanced password authentication through fuzzy logic. IEEE Expert, 12(6), 38–45.

Gaines, R. S., Lisowski, W., Press, S. J., & Shapiro, N. (1980). Authentication by keystroke timing: Some preliminary results (No. R-2526-NSF). Santa Monica, CA: Rand Corporation.

Garcia, J. D. (1986, November 4). Personal identification apparatus. Retrieved from http://www.google.com/patents/US4621334

Giot, R., El-Abed, M., Hemery, B., & Rosenberger, C. (2011). Unconstrained keystroke dynamics authentication with shared secret. Computers & Security, 30(6–7), 427–445. doi:10.1016/j.cose.2011.03.004

Giot, R., El-Abed, M., & Rosenberger, C. (2009). GREYC keystroke: A benchmark for keystroke dynamics biometric systems. In Biometrics: Theory, Applications, and Systems, 2009. BTAS ’09. IEEE 3rd International Conference on (pp. 1 –6). doi:10.1109/BTAS.2009.5339051

Grabham, N. J., & White, N. M. (2008). Use of a Novel Keypad Biometric for Enhanced User Identity Verification. In Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE (pp. 12–16).

Gunetti, D., Picardi, C., & Ruffo, G. (2005). Dealing with different languages and old profiles in keystroke analysis of free text. In Proceedings of the 9th conference on Advances in Artificial Intelligence (Vol. Milan, Italy, pp. 347–358). Berlin, Heidelberg: Springer-Verlag. doi:http://dx.doi.org/10.1007/11558590_36

Hai-Rong Lv, & Wen-Yuan Wang. (2006). Biologic verification based on pressure sensor keyboards and classifier fusion techniques. Consumer Electronics, IEEE Transactions on, 52(3), 1057–1063.

Hwang, S. S., Cho, S., & Park, S. (2009). Keystroke dynamics-based authentication for mobile devices. Computers & Security, 28(1-2), 85–93. doi:10.1016/j.cose.2008.10.002

Karnan, M., & Akila, M. (2009). Identity authentication based on keystroke dynamics using genetic algorithm and particle Swarm Optimization. In Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on (pp. 203–207).

Kotani, K., & Horii, K. (2005). Evaluation on a keystroke authentication system by keying force incorporated with temporal characteristics of keystroke dynamics. Behaviour & Information Technology, 24(4), 289–302. doi:10.1080/01449290512331321884

Leberknight, C. S., Widmeyer, G. R., & Recce, M. L. (2008). An Investigation into the Efficacy of Keystroke Analysis for Perimeter Defense and Facility Access. In Technologies for Homeland Security, 2008 IEEE Conference on (pp. 345 –350). doi:10.1109/THS.2008.4534475

Lee, H., & Cho, S. (2007). Retraining a keystroke dynamics-based authenticator with impostor patterns. Computers & Security, 26(4), 300–310. doi:10.1016/j.cose.2006.11.006

Maiorana, E., Campisi, P., González-Carballo, N., & Neri, A. (2011). Keystroke dynamics authentication for mobile phones. In Proceedings of the 2011 ACM Symposium on Applied Computing (pp. 21–26). New York, NY, USA: ACM. doi:10.1145/1982185.1982190

Martono, W., Ali, H., & Salami, M. J. E. (2007). Keystroke pressure-based typing biometrics authentication system using support vector machines. In Proceedings of the 2007 international conference on Computational science and Its applications - Volume Part II (pp. 85–93). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1802954.1802964

Maxion, R. A., & Killourhy, K. S. (2010). Keystroke biometrics with number-pad input. In Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on (pp. 201–210).

McLoughlin, I. V., & Naidu, N. (2009). Keypress biometrics for user validation in mobile consumer devices. In Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on (pp. 280–284).

Montalvao, J., Almeida, C. A. S., & Freire, E. O. (2006). Equalization of keystroke timing histograms for improved identification performance. In Telecommunications Symposium, 2006 International (pp. 560–565).

Nauman, M., & Ali, T. (2010). TOKEN: Trustable Keystroke-Based Authentication for Web-Based Applications on Smartphones. In ISA (Vol. 76, pp. 286–297). Springer. Retrieved from http://dblp.uni-trier.de/db/conf/sersc-isa/isa2010c.html#NaumanA10

Nauman, M., Ali, T., & Rauf, A. (2011). Using trusted computing for privacy preserving keystroke-based authentication in smartphones. Telecommunication Systems, 1–13. doi:10.1007/s11235-011-9538-9

Ngugi, B., Tremaine, M., & Tarasewich, P. (2011). Biometric keypads: Improving accuracy through optimal PIN selection. Decision Support Systems, 50(4), 769–776.

Nonaka, H., & Kurihara, M. (2004). Sensing Pressure for Authentication System Using Keystroke Dynamics. In International Conference on Computational Intelligence (pp. 19–22).

Obaidat, M. S. (1995). A verification methodology for computer systems users. In Proceedings of the 1995 ACM symposium on Applied computing (Vol. Nashville, Tennessee, United States, pp. 258–262). ACM. doi:http://doi.acm.org/10.1145/315891.315976

Pavaday, N., & Soyjaudah, K. M. S. (2007). Investigating performance of neural networks in authentication using keystroke dynamics. In AFRICON 2007 (pp. 1–8).

Saevanee, H., & Bhatarakosol, P. (2008). User Authentication Using Combination of Behavioral Biometrics over the Touchpad Acting Like Touch Screen of Mobile Device. In Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on (pp. 82–86).

Saevanee, H., & Bhattarakosol, P. (2009). Authenticating User Using Keystroke Dynamics and Finger Pressure. In Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE (pp. 1 –2). doi:10.1109/CCNC.2009.4784783

Sheng, Y., Phoha, V. V., & Rovnyak, S. M. (2005). A parallel decision tree-based method for user authentication-based on keystroke patterns. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics, 35(4), 826–833. doi:10.1109/tsmcb.2005.846648

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

Teh, P. S., Yue, S., & Teoh, A. B. J. (2012). Improving keystroke dynamics authentication system via multiple feature fusion scheme. In 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec) (pp. 277 –282). doi:10.1109/CyberSec.2012.6246096

Teh, P., Teoh, A., Tee, C., & Ong, T. (2011). A multiple layer fusion approach on keystroke dynamics. Pattern Analysis & Applications, 14(1), 23–36. doi:10.1007/s10044-009-0167-9

Wang, X., Guo, F., & Ma, J. (2012). User authentication via keystroke dynamics based on difference subspace and slope correlation degree. Digital Signal Processing, 22(5), 707–712. doi:10.1016/j.dsp.2012.04.012

Young, J. R., & Hammon, R. W. (1989, February 14). Method and apparatus for verifying an individual’s identity. Retrieved from http://www.google.com/patents/US4805222

Zahid, S., Shahzad, M., Khayam, S. A., & Farooq, M. (2009). Keystroke-Based User Identification on Smart Phones. In Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection (Vol. Saint-Malo, France, pp. 224–243). Berlin, Heidelberg: Springer-Verlag. doi:http://dx.doi.org/10.1007/978-3-642-04342-0_12