

A k-Nearest Neighbor Approach for User Authentication through Biometric Keystroke Dynamics

J.Hu, D. Gingrich
School of Computer Science and IT
RMIT University
Melbourne, Australia
{Jiankun, gingrich}@cs.rmit.edu.au

A. Sentosa
Editure Pty Ltd
Melbourne, Australia
andy.sentosa@gmail.com

Abstract— Keystroke dynamics exhibit people’s behavioral features which are similar to hand signatures. A major problem hindering the large scale deployment of this technology is its high FAR (False Acceptance Rate) and FRR (False Rejection Rate). A significant progress, in terms of improving the FAR and FRR performance, has been made by the work of Gunetti and Picardi (2005). However, their identification based authentication suffers a severe scalability issue as it needs to verify the input with every training sample of every user within the whole database. In this paper, a *k*-nearest neighbor approach has been proposed to classify users’ keystroke dynamics profiles. For authentication, an input will be checked against the profiles within the cluster which has greatly reduced the verification load. Experiment has demonstrated the same level of FAR and FRR as that of Gunetti and Picardi approach while as high as 66.7% improvement of the authentication speed has been achieved.

Keywords—biometric security; keystroke dynamics; access control; biometric authentication

I. INTRODUCTION

Access control is fundamentally important in network based applications. Authentication is the essential component of access control mechanisms. The most widely deployed authentication mechanism is based on token and/or password. Password or personal identification number (PIN) authentication is a knowledge-based mechanism relying on something you know. Token-based authentication is based on something you have, like a smartcard. These two types of authentication mechanisms have several flaws. First, tokens can be lost or stolen. Secondly, simple or meaningful PINs are easier to remember, but are vulnerable to attack. PINs that are complex and arbitrary are more secure, but are difficult to remember. Since users can only remember a limited number of PINs, they tend to write them down or will use similar or even identical PINs for different purposes. Passwords are prone to attacks such as brute-force or dictionary attacks.

Since passwords are limited to the characters found on keyboards, the number of combinations is limited and highly dependent upon on the length of the password itself. An unauthorised user may try all possible combinations until it finds a valid password. This is called as a brute-force attack.

Additionally, most users tend to choose something they can memorise easily such as the use of common words, things, events, or person. Consequently, attackers can create a list of meaningful words to try in breaking the system, this method is called dictionary attack.

The final but the worst problem using PIN/Token based authentication mechanisms is that we can only verify that the user has the right information but we have no idea whether the user is the genuine user or not [1]. This is a fundamental flaw in the existing security systems.

Biometric authentication has emerged as a promising solution in addressing the above issues and has now become more popular in civilian applications such as access control, financial security et al. [2]. Biometrics is the science of identifying individuals by a particular physiological, such as voice, fingerprint, face, iris etc or behavioral characteristic such as signature, keystroke dynamics et al. Fingerprints are arguably the most popular biometric currently in use [3][4]. Personal physiological or behavioral characteristics are not subject to lost, forgotten and hard to forge. Presentation of such characteristics will need the right person to be physically present. There are of course some attacks on biometric security mechanisms. For fingerprint, e.g., the attacker can manage to get the victim’s latent fingerprint and forge a copy of the fingerprint. However, there are also counter-attack measures such as body temperature sensing etc. Such attacks and counter-attacks are beyond the scope of this paper. In this paper, we focus on the issue of authentication in terms of user matching or verification.

Although physiological biometrics such as fingerprint, face etc are the most widely deployed for biometric authentication, keystroke dynamics based authentication has its own merit. The most important advantage is that it does not require any extra equipments or devices as a keyboard is universally available. The second advantage is that this technology is least intrusive which is also important for a large scale commercial deployment.

Keystroke dynamics, or typing rhythms, is based on the use of comprehensive timing information that describes the activity of keys when they are pressed or released as a person

This work was supported in part by the ARC (Australia Research Council) Linkage Grant under Grant LP0455324.

is typing on computer keyboard. Since keystroke dynamics is a behavioural biometry, it is often unstable and unreliable as it can vary from time to time, which has become its technological bottleneck. An on-going effort has been made in addressing this problem.

Monrose *et al.* (1999) [5] has proposed to incorporate keystroke pattern recognition into password authentication. The proposed algorithm analysed keystroke patterns based on a short static string which was the password itself. They proposed an algorithm that combined the password with the keystroke latency and duration to generate the hardened password. This hardened password became the final outcome for authentication. This work has laid a foundation of the commercial software package called Biopassword [7]. There are several issues regarding this work. First, their experiment was limited to a single password string for all users. Difficulties emerged because different people could have different familiarity with the password, which makes it unlikely to display their normal typing behaviour. Secondly, typing errors did occur during experiment and some forms of error correction were required. Without any error correction, the experiment undertaken performed poorly resulting in approximately 40% of false negative rate. Since password only represents a small amount of typing pattern, researchers acknowledge the need of longer text samples. Longer text samples will generate more typing patterns. Therefore, it is easier to recognise typing behaviour. Also, with more typing patterns, more discriminative profiles can be built.

Bergadano *et al.* (2002) [6] used around 680 characters in the training text. They proposed an algorithm that did not rely directly on the exact time information. Instead, they used the timing information to obtain the relative order of trigraphs. Their method of analysis was called Degree of Disorder. It was used to compare between two different sets of sorted trigraphs and to measure the difference in the ordering between them. This was seen as a solution to reduce the effect of variations in the absolute timing data on the authentication mechanism. It is known that the keystroke timings can vary for each authentication attempt. However, the order of the timing will probably remain constant. The authentication process will not be affected if the keystroke timings were different, as long as the order is the same.

Lau *et al.* (2004) [9] had done further research that supports this claim. They discovered that there were large inconsistencies in data that prevented the use of standard statistical methods such as mean and standard deviation in performing accurate comparisons of typing patterns for different users. They concluded that different typing samples exhibited some degree of stability in the typing speed of particular keys and that Degree of Disorder was an effective method in keystroke authentication.

Gunetti and Picardi (2005) [10] further extended this idea by incorporating all n -graphys. They also proposed an identification based approach for the authentication. In their experiment, an authentication performance of less than 5% FAR and less than 0.005% FRR has been achieved. However this approach is not practical for a large database due to its scalability problem. The proposed identification approach

needs to compare an input with every training sample of every user profile. The verification effort grows exponentially with the size of the database.

In this paper, a k -nearest neighbor classification based authentication is proposed. With this proposed scheme, an input needs only to be verified against limited user profiles within a cluster which effectively reduces the verification load significantly. Experiment shows that our proposed authentication can achieve the same level of FAR and FRR performance while has improved the authentication efficiency up to 66.7% compared with the Gunetti and Picardi (PG) approach. The remaining sections of this paper are organized as follows. Section II describes the proposed authentication scheme and performance evaluation. Section III is devoted to the conclusions.

II. K-NEAREST NEIGHBOUR CLASSIFICATION BASED KEYSTROKE DYNAMICS AUTHENTICATION

A. Preliminary

For convenience, we adopt similar notations used in [10] unless stated otherwise.

Degree of Disorder (trigraphs) [6]: The duration of trigraphs refers to the elapsed time between the first key pressed and the third key pressed. Two arrays of trigraphs will be then sorted according to the duration of trigraphs and hence a distance in terms of degree of disorder between these two arrays is counted as the sum of the number of disorder of each trigraph in an array. The dimension N of each array could be different. For normalization, the distance calculated above is divided by $N^2 / 2$ if N is even or $(N^2 - 1) / 2$ if N is odd.

Suppose that we have two samples of trigraph arrays in typing the word “computer” as follows

S1: com:190;omp205;mpu:160;put:192;ute:201;ter:243,
S2: com:200;omp196;mpu:180;put:202;ute:207;ter:201,

Then we have the following result,

Sorted S1			Sorted S2	
MPU	160 ms	d=0	MPU	180 ms
COM	190 ms	d=1	OMP	196 ms
PUT	192 ms	d=1	COM	200 ms
UTE	200 ms	d=1	PUT	202 ms
OMP	205 ms	d=3	UTE	207 ms
TER	243 ms	d=0	TER	220 ms

Fig. 1 Example of degree of disorder for S1 and S2 [8]

Therefore, the normalized distance is given by

$$d(S1, S2) = 7 / (36 / 2) = 0.389 . \quad (1)$$

Without loss of generality, distance is referred to a normalized disturbance in the remaining sections of this paper unless stated otherwise.

For a general distance involving all n -graphs, the following formula is given [10]

$$d_{n,m,p}(S1, S2) = d_n(S1, S2) + d_m(S1, S2) \frac{M}{N} + d_p(S1, S2) \frac{P}{N} , \quad (2)$$

where n, m , and p represent different n -graphs and N, M , and P represent the number of shared n -graphs with $N > M, P$.

To improve the performance of the above measures, an "A" measure is also introduced [10]. "A" measure deals directly to the timing or duration of n -graphs. It calculates the number of n -graphs that have similar duration between two typing data. In "A" measures, duration 1 (D1) and duration 2 (D2) are said to be similar if the following condition is satisfied:

$$1 < \frac{\max(D1, D2)}{\min(D1, D2)} \leq 1.25 . \quad (3)$$

For two samples S1 and S2, a normalized "A" measure is given by

$$A_n(S1, S2) < 1 - \frac{(\text{no. of similar } n\text{-graphs shared})}{\text{total no. of } n\text{-graphs shared}} . \quad (4)$$

The normalized "A" measures $A_n(S1, S2)$ will be added to the right side of the eq.(2) accordingly to refine the distance measure [8][10]. For convenience, distance measure will be referred to this adjusted distance metric in the rest of this paper unless stated otherwise.

B. Proposed Classification-Based Authentication Method

Classification based verification approach is often deployed for the problem of biometric identification within a large database where the input is unknown [4]. The main benefit of clustering or classification is that it can significantly increase the matching efficiency. Nearest neighbour is a simple classification method based on distance measurement. It works by applying a distance measurement between two data and then calculate the value. If the distance value is within a chosen value, k , then both data will be considered as a neighbour. There is no general optimum value for k and it is usually found by using trial and error approach.

The k -Nearest neighbour approach is a very simple classifier that can easily apply any distance measurement into

the classification mechanism. There is a nice fit between the characteristics of this classification and distance based authentication problem in this paper. Therefore, we choose the k -nearest neighbour approach for our clustering purpose.

Clustering Based Keystroke Authentication Algorithm (CKAA):

Step 1. Building representative user profiles: (i) Each legitimate user needs to provide several training samples. Each training sample will contain many n -graphs. A sorted n -graphs vector is formed by averaging corresponding groups of the graphs. For instance, an element "AB" is formed by averaging all "AB" digraphs within this sample. (ii) A representative user profile is built by averaging all such vectors from all training samples provided. For convenience, we use its representative mean $m(A)$ to present the mean of the user A.

Step 2. Clustering process: The k -nearest neighbour method is applied to cluster the representative profiles based on the distance measure. A threshold is set to control the size of the clusters. The optimal threshold is experimentally determined. For a user profile, it associates with a cluster and we call all other profiles associated with this profile the List of this profile.

Step 3. Authentication Process: When a keystroke dynamics sample X is claiming as user A , a positive authentication is confirmed if following conditions are satisfied: (i) A must be within the cluster that is closest to X , and (ii) $md(A, X)$ value must be the closest to $m(A)$ within its cluster.

Discussions

The major difference between the proposed CKAA algorithm and the method of Gunetti and Picardi (GP method) [10] is that the verification process of the CKAA is within a cluster while the GP method needs to go through the entire database. Also for a user profile A, CKAA uses only its representative profile in the authentication process while the GP method needs to compare with every sample of each user profile. The extra overhead of the CKAA algorithm is the need to cluster the database. However, this clustering work can be done offline without affecting the authentication process. If a new user joins into the existing systems, we may either include it in an existing cluster if it satisfies the predefined cluster rules or establish a new cluster if it can not be included into any other existing cluster.

C. Performance Evaluation

Like in many other biometric authentication applications, the specifications of performance of the system will be evaluated by measuring the number of correct authentication, false authentication, and computational efficiency. Following metrics are used.

False Acceptance Rate (FAR)—the percentage of an impostor that managed to login to the system

False Rejection Rate (FRR) — the percentage of a valid user that is being denied an authentication

Time efficiency — the time taken for a user to perform a single authentication.

Values for FAR and FRR were aimed to be as low as possible. Ideally the FAR and FRR should be 0%, but nearly all biometry applications have never produced this value. As described by European Standard for Access Control (EN 50133-1), an acceptable error rate for commercial biometrics is 1% and 0.001% of FRR and FAR respectively [11].

FAR and FRR are usually inversely proportional to each other. So, reducing FAR will result in an increase of FRR, and vice versa. Although people sometimes allocate more preference on one of these rates, a balance result between FAR and FRR is required for a better authentication system. A threshold value is used to investigate the best possible combination of FAR and FRR. The effect of threshold value will be different depending on the implementation of the experiment. In our case, a low threshold value refers to a strict authentication process, which will produce a low FAR, but at the same time a high FRR. In contrast, by increasing the threshold value, the system would be more tolerant to typing variation and would result in low FRR, but high FAR. A balance threshold value will be explored through trial and error technique.

Time metric is added in order to measure the authentication efficiency. We measured the time taken to perform the full experiment then divided by the number of legitimate authentications made. By doing this, the average time for each authentication was obtained. The time was measured by using Linux command ``time" (user time + system time) which provided precision up to one millisecond.

Experimental Setting

The sample collection process was designed to use *thick* client mechanism in order to avoid any obstacles that might occur in collecting keystroke data. *Thick* client means that all data collection is performed in the client computer and then these data are transferred to the server. All typing samples in this experiment were obtained from a JavaScript in simple HTML form. Two pieces of information were collected from each keystroke event; they were the character typed and the time stamp when the key was pressed. Time stamp was recorded with accuracy of one millisecond. The experiment was carried out on a PC with AMD 1GHz processor and 640 MB of internal memory running on Linux Ubuntu 6.10 operating system. All programs for this experiment were created by using the Perl Programming Language.

Sample Collection

We provided a JavaScript in a simple HTML form with the same information gathered (keystroke character and keystroke time stamp). We collected typing samples on static text. This text was seen as a more extensive way to detect error in false acceptance rate because the typing text was the same for all users and impostors. With the same text, the chance of impostor passing rate would potentially be higher. This text was prepared by taking into account some common digraphs and trigraphs that occurred in English language. Top 20 digraphs and trigraphs from

```
\url{http://home.ccil.org/~cowan/trigrams} and  
\url{http://www.cs.chalmers.se/Cs/Grundutb/Kurser/krypto/en  
_stat.html}
```

were formed into words and sentences to create this static text. It was designed to be complex and did not represent any flow. The reason was to make every participant unfamiliar with the text so that the typing rhythms would be neutral to all users. 19 individuals participated in this experiment with each of them provided five typing data. These data acted as legitimate users. All participants were university students or graduates and had some experience in typing. Although, we did not put any time frame for users to provide samples, each user returned with the samples in about two days. With two people provided samples everyday. Another 17 people provided 27 typing samples which were used as impostor data.

Typing environments were not controlled in this data collection. So people could type whatever they want such as performing correction. This way volunteers could provide typing samples in their natural way. All samples above were provided on different platforms (Windows and UNIX) using both Internet Explorer and Netscape browsers. In static text samples, nineteen individuals provided five typing samples for each user. This would make the total number of legal attempt to be $5 \times 19 = 95$. However, there was one void sample, hence legal attempt were subtracted by one to 94. These 94 legal attempts were used to generate FRR by authenticating each legal attempt with the correct user's profile. During each legal attempt, the sample, which was used as the authentication, was removed from the profile temporarily. This was to enable the system to consider the attempt as a new sample. By doing this, we generated a true behaviour in real life application since each authentication would be new to the system.

On the other hand, to generate FAR, we used 27 typing data that were provided by impostors along with samples from other's profile in the database. Therefore, for each profile, it was attacked $27 + 18 \times 5 = 117$. For all profiles, we could accumulate $117 \times 19 = 2223$ impostor attempts.

Experimental Results

In the experiment, the clustering threshold parameter *Th* is represented as the percentage of the maximum distance among

all user profiles. A cluster size is the size d of the corresponding user profile plus the threshold which is $d+Th$. Expanding the threshold will increase the size of a cluster. The experimental results are shown in TABLE I and TABLE II where authentication time in the table is an averaged value.

TABLE I. SPEED GAIN OVER THE PG METHOD

Method	Th(%)	FRR %	FAR %	Auth. time(s)	Speed Gain. Over PG method (%)
PG		0	0.045	22.006	
CKAA	32.5	0	79.622	2.484	88.7
	35.0	0	26.541	3.873	82.4
	37.5	0	0.045	7.323	66.7
	40.0	0	0.045	13.16	40.2
	42.5	0	0.045	16.908	23.2
	45.0	0	0.045	19.731	10.3

TABLE II. AUTHENTICATION TIME ALONG WITH SIZES OF THE DATABASE

No. Profiles in the database	Th (%)	5 Profiles	10 Profiles	15 Profiles
Method				
Auth. Time based on PG		21.852	29.155	37.959
Auth. Time based on CKAA	32.5	14.305	13.239	13.150
	35.0	14.130	14.653	14.612
	37.5	16.028	18.192	19.301
	40.0	18.709	22.671	27.068
	42.5	20.211	24.602	31.303
	45.0	20.199	25.892	34.355

Table I shows that the proposed CKAA algorithm can achieve the same level of FAR and FRR performance while having 66.7% gain in the average authentication speed. The optimal threshold for clustering is at 37.5% value. As shown in the Table II, the average authentication time for the PG method increases exponentially with the size of the database (22s to 37.9s) while for the proposed CKAA algorithm the average authentication time increases from 16s to 19.3s which is very scalable.

III. CONCLUSIONS

In this paper, an efficient keystroke dynamic authentication algorithm has been introduced. The proposed clustering based keystroke authentication algorithm (CKAA) has solved the scalability problem suffered by the PG method while can achieve the same good performance in terms of FAR and FRR. Online experiments have been conducted which has validated the proposed scheme. Keystroke dynamic authentication has its unique advantages such as inexpensive and universal available etc. Our result can further help to advance this technology towards practical applications.

REFERENCES

- [1] D. Gollman, Computer security, John Wiley & Sons, 1999.
- [2] A. K. Jain, L. Hong and r. Bolle, "On-line fingerprint verification," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol.19, no.4, pp.302-314, 1997.
- [3] R. S. Germain, A. Califano, and S. Colville, Fingerprint matching using transformation parameter clustering, *IEEE Computational Science and Eng.*, vol.4, pp.42-49, 1997.
- [4] Y. Wang, J.Hu and D. Philip, "A fingerprint orientation model based on 2D Fourier Expansion (FOMFE) and its application to singular-point detection and fingerprint Indexing," *Special Issue on Biometrics: Progress and Directions, IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.573-585, April 2007.
- [5] F. Monrose, M.K. Reiter and S. Wetzel, "Password hardening based on keystroke dynamics," *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 73-82, New York, NY, USA, ACM Press, 1999.
- [6] F. Bergadano, D.Gunetti and C. Picardi, "User authentication through keystroke dynamics," *ACM Trans. Info. Syst. Secur.*, 5(4), 2002, pp.367-397.
- [7] Biopassword. Biopassword authentication software homepage. <http://www.biopassword.com>. Retrieved on 20-May-2007.
- [8] Andy Sentosa, User authentication through keystroke dynamics, Honors Thesis, RMIT University, June 2007.
- [9] E. Lau, X. Liu, C. Xiao and x. Yu, "Enhanced user authentication through keystroke biometrics." Technical Report, MIT 2004. <http://web.mit.edu/edmond/edmond-space/projects/6.857/Keystroke%20Biometrics.pdf>. Retrieved on 20 May, 2007.
- [10] D. Gunetti, and C. Picardi, "Keystroke analysis of free text," *ACM Trans. Info. Syst. Secur.*, 8(3), 2005, pp.312-347.
- [11] D. Polemi, "Biometric techniques: review and evaluation of biometric techniques for identification and wuthentication, including an appraisal of the areas where they are most applicable." Reported prepared for the European Commision DG XIIC.4 on the Information Society Technologies. <http://cordis.europa.eu/infosec/src/stud5.fr.htm>. Retrieved on September 24, 2007.