Advance Access publication on 29 July 2011
doi:10.1093/comjnl/bxr064

Correlation Keystroke Verification Scheme for User Access Control in Cloud Computing Environment

KAI XI[1], YAN TANG[2] AND JIANKUN HU[1,*]

[1]*School of Engineering and Information Technology, University of New South Wales at the Australian Defence Force Academy, Canberra, Australia*
[2]*School of Computer Science and IT, RMIT University, Melbourne, Australia*
[*]*Corresponding author: J.Hu@adfa.edu.au*

Cloud security is a major concern that may delay its widespread adoption. User access control (UAC) is the core component of security in cloud computing environment, aiming to ensure that stored data are allowed to be accessed only by authenticated/authorized users. As a typical behavioural biometrics, keystroke dynamics provides a promising UAC solution. The most challenging issue that hinders the wide deployment of keystroke is the high verification error rate. Gunetti *et al.* proposed a classical *n*-graph-based keystroke verification method (GP method), which can achieve a low False Acceptance Rate (FAR). However, the GP method suffers from a high False Rejection Rate (FRR) and a severe scalability issue. Thus, GP is not a feasible solution for computing cloud application where scalability is a big issue. In this paper, two keystroke verification approaches (*n*Gdv-V and *n*Gdv-C) are proposed to overcome GP's shortcomings. To reduce high FRR, we designed a new correlation measure using *n*-graph equivalent feature (*n*Gdv) that enables more accurate recognition for genuine users. Moreover, correlation-based hierarchical clustering is proposed to address the scalability issue. The experimental results show that the *n*Gdv-C can produce much lower FRR while achieving almost the same level of FAR as that of the GP method. Furthermore, 1250 times (when using *n*Gdv-V) and three times (when using *n*Gdv-C(17,4)) authentication speed gains have been achieved.

Keywords: keystroke dynamics; access control; biometric authentication; security; cloud computing

Received 3 February 2011; revised 14 April 2011
Handling editor: George Loukas

1. INTRODUCTION

Cloud computing has become an emerging technology for delivering computing and storage resources to customers on demand. It enables its users to store/backup local data to an online virtualized storage [1]. The benefit is that a user only needs to pay for the amount of storage the data actually consume. Most importantly, convenience is another major benefit since cloud data can be accessed anytime anywhere by the user when network is available. However, the security of cloud computing is still a contentious and unsolved issue that may hinder its wide deployment. As the core security component, user access control (UAC) plays a vital role in recent cloud computing systems. In past, the user data can be protected physically. For instance, the sensitive data, stored in a laptop, are unaccessible by other users except the owner, unless the laptop itself is lost or stolen. Unfortunately, cloud computing platform cannot offer such physical protection, and all protection relies significantly on the UAC mechanism such as user authentication.The worries about cloud security are not entirely theoretical. Recently, commercial cloud provider Google found a flaw in its SaaS (software-as-a-service) application that the stored private users' files become visible and accessible to some unauthorized users [2].

The most widely deployed user authentication mechanism is based upon password. Password or personal identification number (PIN) authentication belongs to a knowledge-based mechanism that relies on something the user knows. It is commonly known that all knowledge-based authentication mechanisms have a fundamental flaw in identifying genuine users [3]. Also user-friendly is another issue [3, 4]. Simple or meaningful passwords are easier to remember, however, are

vulnerable to attack. On the contrary, complex and arbitrary passwords are more secure, but are difficult to remember. Since most users are only able to remember a limited number of characters, they tend to write them down or use similar or even identical passwords for different purposes. Passwords are vulnerable 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 the length of the password itself. An attacker may try all possible combinations until finding a valid password, so-called brute-force attack. Besides, most users tend to choose something they can memorize easily such as names, favourite colours and date of birth. Consequently, attackers may create a list of meaningful words to break the system, which is called dictionary attack. The fundamental problem of knowledge-based authentication mechanisms is that they can only ensure that the person possesses the right information, however, cannot confirm that the person is a legal user [5]. This issue cannot be easily resolved in all existing knowledge-based security systems.

Biometric technique has emerged as a promising solution in addressing these issues and has now become more popular in civilian applications such as access control, financial security etc. [4]. Biometrics is the science of identifying a person by particular physiological features, such as fingerprint, face, iris etc. or behavioural characteristics such as signature, keystroke dynamics etc. Fingerprints are arguably the most widely deployed biometric [6].

Personal physiological or behavioural characteristics are not subject to lost, forgotten and hard to forge. Presentation of such characteristics requires the right person to be physically present. There are of course some attacks on biometric security mechanisms. For example, an attacker can manage to get the victim's latent fingerprint and forge a synthetic print [7]. 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 mainly focus on the issue of authentication in terms of user verification.

Although physiological biometrics such as fingerprint, face etc. are the most popular methods for biometric authentication, they may not be the best choice for cloud computing system. Firstly, physiological biometric authentication systems require professional and high-priced equipments such as fingerprint scanner or high-resolution digital camera, which increase the cost as well as difficulties in incorporating them to the existing cloud computing architecture. Secondly, physiological biometric normally is considered to be intrusive. For instance, a person may not be willing to store his/her facial image in an unknown place such as a third-party hosting server [8]. As a behavioural biometric, authentication via keystroke dynamics does not have the earlier-mentioned concerns. Keystroke sample capture only requires a keyboard, a universally available device, rather than extra scanner/camera.

Keystroke dynamics, or typing rhythms, refers to the use of comprehensive timing information that describes the activities of pressing keys and releasing keys when a person is typing on a keyboard [9]. However, as a behavioural biometrics, keystroke dynamics is often considered to be unstable, unreliable and vary from time to time. In general, behavioural biometrics' recognition performance is worse than that of the physiological biometrics. Hence, an on-going effort has been made in addressing this problem.

1.1. Related works

Monrose *et al.* [10] proposed to incorporate keystroke pattern recognition into password authentication. The proposed algorithm analysed keystroke patterns based on the password itself. The algorithm combined the password with the keystroke features (latency and duration) to generate the hardened password. This hardened password became the final outcome for authentication. The work has laid a foundation for the commercial software package called Biopassword [11]. There are some issues regarding this work. Firstly, 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 $\sim$40% of false rejection rate. Since password only represents a small amount of typing pattern, researchers acknowledge the need of longer text samples. Longer text samples can contain more typing features, yielding a higher probability to separate different users. Also, with more typing patterns, more comprehensive templates can be built.

Bergadano *et al.* [12] proposed a trigraph-based algorithm that did not rely directly on the exact timing information. Instead, they used the timing information to obtain the relative order of trigraphs. The training text contains 680 characters. Their method of analysis was called Degree of Disorder (DoD), which compares two different sets of sorted trigraphs and measures the difference in the ordering between them. It is a solution to reduce the effect of variations in the absolute timing data on the authentication mechanism. It is known that the keystroke typing durations usually vary for each typing sample. But the order of the timing will probably remain constant. It means that by using the order of trigraphs, the authentication results tend to be stable even if the samples of typing duration show variation.

Lau *et al.* [13] did further research that supports Bergadano's work. They discovered that there were large inconsistencies in keystroke 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 instability in the typing speed of particular keys and that DoD was an effective method in keystroke authentication.

Gunetti and Picardi [14] proposed the GP method, which is an extension of [12] incorporating all n-graph. The GP method is an identification-like authentication approach. They claimed, in the best-case scenario, 5% False Rejection Rate (FRR) with around 0.005% False Acceptance Rate (FAR) can be achieved. However, there are several issues. Firstly, the authors mistakenly calculate the FAR in their experiment. The total number of impostor attempts should be 30 000 instead of 450 000 as claimed. Subsequently, all FAR should be 15 times larger than the values claimed in the paper. Secondly, the proposed method is not practical for a large system due to its severe scalability problem. GP works in an identification way where a query sample will be compared with every training sample of every user in the system. The computational cost grows polynomially (n^2) with respect to the number of registered users in system, which makes it infeasible for large-scale systems, such as computational cloud.

1.2. Our contributions

In the research, we worked on addressing two major issues. The first issue is authentication accuracy. Existing schemes such as the GP method do not have a high Genuine Acceptance Rate (GAR) though the False Acceptance Rate (FAR) is very low. We aimed to enhance GAR without compromising FAR. The second issue is the scalability and computational efficiency.The desirable method should feature high computational efficiency and low complexity. We intended to employ n-graph keystroke feature as well as DoD as part of our new algorithms.

Correlation pattern recognition (CPR) is a direct, sophisticated and universal method that has been widely used for recognizing biometric traits such as fingerprint and face [15]. Most importantly, the verification speed of CPR is very fast. However, it seems extremely hard to incorporate the correlation method in keystroke dynamics analysis. Sentosa [16] conducted preliminary tests and obtained poor results; so a conclusion is drawn that correlation is not suitable for keystroke. The poor performance is due to the inappropriate utilization of keystroke feature rather than the mechanism of CPR algorithm.

In our work, we introduced a novel way of transforming n-graph DoD to a correlation-oriented feature, so-called n-graph disorder vector (nGdv). The nGdv is exactly equivalent to n-graph DoD with no information loss, which successfully minimizes the negative impact on the system performance brought about by the inappropriate choice of keystroke features. Consequently, CPR is effective for keystroke analysis. We developed two ways of using CPR for keystroke pattern recognition. One way is applying CPR for verifying a person directly, and the other way is using CPR as classification/clustering, associated with DoD measure.

The contributions of this paper are concluded as:

(i) Having applied correlation pattern recognition (CPR) to keystroke pattern analysis for the first time.
(ii) Proposed an equivalent representation of n-graph DoD (nGdv) that retains all critical typing information.
(iii) Proposed a CPR-based keystroke verification scheme, nGdv-V, using proposed nGdv feature. nGdv-V is over 1000 times faster than GP method.
(iv) Proposed a keystroke verification scheme nGdv-C, which uses CPR-based hierarchical clustering with nGdv feature. The FRR and computational efficiency of nGdv-C outperforms that of the GP method.

The remaining sections of this paper are organized as follows. Section 2 provides preliminary knowledge of DoD measure, GP method and CPR. Section 3 describes the proposed nGdv feature and two CPR-based keystroke verification methods (nGdv-V and nGdv-C) using nGdv. Section 4 demonstrates the experimental results and analysis. Section 5 is devoted to the conclusions.

2. PRELIMINARY

2.1. Degree of disorder [12]

Given two vectors V and V' (each with N elements), as

DoD between V and V' can be calculated as the summation of the distances, denoted as d_{Idx}, between the position of each element in V and the position of the same element in V'. In this example, d_{Idx} of each corresponding element is computed as

$$\begin{aligned} d_{\mathrm{Idx}}(A) &= |\mathrm{Idx}(A^V) - \mathrm{Idx}(A^{V'})| = |1-3| = 2, \\ d_{\mathrm{Idx}}(E) &= |\mathrm{Idx}(E^V) - \mathrm{Idx}(E^{V'})| = |2-2| = 0, \\ d_{\mathrm{Idx}}(I) &= |\mathrm{Idx}(I^V) - \mathrm{Idx}(I^{V'})| = |3-5| = 2, \\ d_{\mathrm{Idx}}(O) &= |\mathrm{Idx}(O^V) - \mathrm{Idx}(O^{V'})| = |4-1| = 3, \\ d_{\mathrm{Idx}}(U) &= |\mathrm{Idx}(U^V) - \mathrm{Idx}(U^{V'})| = |5-4| = 1, \end{aligned} \tag{1}$$

where $d_{\mathrm{Idx}}(x)$ can be calculated as the absolute difference between the index numbers of an element x in V and V'. DoD between V and V' is computed as

$$\begin{aligned} \mathrm{DoD}(V, V') &= \sum_{x \in V} d_{\mathrm{Idx}}(x) \\ &= d_{\mathrm{Idx}}(A) + d_{\mathrm{Idx}}(E) + d_{\mathrm{Idx}}(I) \\ &\quad + d_{\mathrm{Idx}}(O) + d_{\mathrm{Idx}}(U) \\ &= 2 + 0 + 2 + 3 + 1 = 8. \end{aligned} \tag{2}$$

Apparently, DoD increases in direct proportion to the vector size N and thus a normalization is needed. It is convenient to

normalize DoD via dividing it by the value of the maximum possible DoD value for a given N. The maximum DoD is:

$$\begin{cases} \frac{N^2}{2} & \text{for } N \text{ is even} \\ \frac{N^2-1}{2} & \text{for } N \text{ is odd} \end{cases}. \quad (3)$$

After normalization, the value of DoD falls between 0 (when V and V' are exactly the same) and 1 (when V' is in the reverse order of V). In this example, the normalized DoD between V and V' is $8/((5^2-1)/2) = 8/12 = 0.667$. Unless otherwise stated, all DoD appearing in following sections are referred to the normalized DoD.

2.2. Similarity of two typing samples

A discriminative, stable and robust feature plays a vital role in measuring the similarity of two typing samples. Methods in the literature use the two basic features: (i) the duration of hold on one key (ii) the latency between two consecutively typed keys—the elapsed time between the release of the first key and pressing of the second key. In Gunetti and Picardi [14], the authors chose to use the n-graph feature. The experimental results further proved that n-graph outperformed other features. The duration of an n-graph refers to the elapsed time between the first key pressed and the nth key pressed. For instance, two keys typed one after the other are called a *digraph*. The duration of a *digraph* is the time between the first key pressed and the second key pressed. Similarly, three consecutively typed keys are called a *trigraph*. The duration of a *trigraph* is the time between the first key pressed and the third key pressed.

Suppose that we have two samples of digraph vectors in typing the word 'computer' as follows:

s1 : co 30, om 20, mp 10, pu 70, ut 50, te 60, er 40,

s2 : co 30, om 20, mp 45, pu 48, ut 50, te 60, er 40,

where each digraph is followed by its duration in millisecond (ms).

The sorting index of each digraph is listed in Table 1. DoD (normalized distance of disorder) between s_1 and s_2 is computed as

$$\text{DoD}(s1, s2) = (1+1+3+2+1+1+1)/((7^2-1)/2) = 0.4167. \quad (4)$$

2.3. GP method

Gunetti and Picardi [14] proposed the method (GP method) that employs DoD as the similarity measure, shown as follows.

Given a user u_i with m typing samples $S_i = \{s_{ij}\}_{j=1}^{m}$. S_i is called the typing profile/template of u_i. Define the internal mean DoD of S_i, denoted by M_d^{in}. For u_i, $M_d^{\text{in}}(S_i)$ is computed by taking the average of the DoD between each sample pair in a profile, as

$$M_d^{\text{in}}(S_i) = \frac{1}{n} \sum_{p,q \in [1,m]}^{p \neq q} \text{DoD}(s_{ip}, s_{iq}), \quad (5)$$

where s_{ip} and s_{iq} represent the pth and qth samples randomly selected from the profile of the user i. n is the total number of sample pairs in S_i yielding $n = C_m^2$.

Given a query sample s_x. Define the external mean DoD, denoted by M_d^{ex}. $M_d^{\text{ex}}(S_i, s_x)$ is used to measure the similarity between the template S_i and the query sample s_x, computed as

$$M_d^{\text{ex}}(S_i, s_x) = \frac{1}{m} \sum_{p \in S_i} \text{DoD}(s_{ip}, s_x). \quad (6)$$

Note all DoD is referred to normalized DoD.

The verification process of the sample s_x is illustrated in Fig. 1, following the steps:

(i) Compute M_d^{in} for all existing users, using Equation (5).
(ii) Compute M_d^{ex} between s_x and each existing user in the system, using Equation (6).
(ii) Suppose s_x is claimed to be from the user u_a. The statement is true only if the following two authentication rules are satisfied:
  (a) $M_d^{\text{ex}}(S_a, s_x)$ is the smallest one among all $M_d^{\text{ex}}(\cdot, s_x)$;
  (b) $M_d^{\text{ex}}(S_a, s_x)$ is smaller than $M_d^{\text{in}}(S_a)$
  or
  $M_d^{\text{ex}}(S_a, s_x)$ is closer to $M_d^{\text{in}}(S_a)$ than to any other $M_d^{\text{ex}}(\cdot, s_x)$. The following inequality should be satisfied:

$$M_d^{\text{ex}}(S_a, s_x) < M_d^{\text{in}}(S_a) + 0.5(M_d^{\text{ex}}(\cdot, s_x) - M_d^{\text{in}}(S_a)). \quad (7)$$

For example, there are three users u_a, u_b, u_c in the system. Each user's template consists of three typing samples, which are $\{s_{a1}, s_{a2}, s_{a3}\}$, $\{s_{b1}, s_{b2}, s_{b3}\}$, $\{s_{c1}, s_{c2}, s_{c3}\}$. The query sample s_x claims to come from u_a.

TABLE 1. Sorting Index Table of **s1** and **s2**.

S1		*S2*	
Digraph x	Idx y	Digraph x	Idx y
co	3	co	2
om	2	om	1
mp	1	mp	4
pu	7	pu	5
ut	5	ut	6
te	6	te	7
er	4	er	3

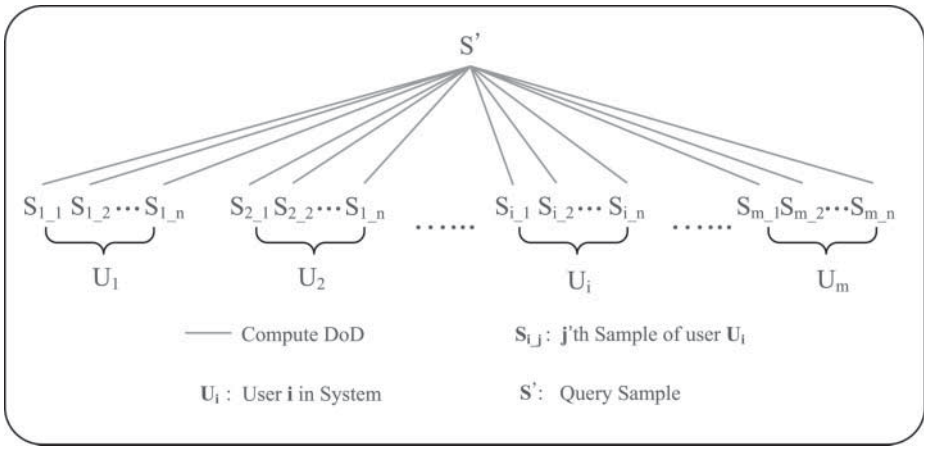

FIGURE 1. GP method.

Suppose the three corresponding internal mean DoD are: $M_d^{\text{in}}(S_a) = 0.314$, $M_d^{\text{in}}(S_b) = 0.324$, $M_d^{\text{in}}(S_c) = 0.457$. The external mean DoD are: $M_d^{\text{ex}}(S_a, s_x) = 0.329$, $M_d^{\text{ex}}(S_b, s_x) = 0.541$, $M_d^{\text{ex}}(S_c, s_x) = 0.532$.

Apparently, $M_d^{\text{ex}}(S_a, s_x) = 0.329$ is the smallest one; thus the authentication rule (i) is satisfied, and so does Equation (7). Hence s_x is from u_a and authentication is successful.

2.4. Correlation pattern recognition

Correlation (filter) pattern recognition (CPR) was originally designed for automatic target recognition (ATR) applications [15]. Afterward, it was widely applied in biometric authentication.

In signal processing, cross-correlation is a method of measuring the similarity between a reference pattern $r(x)$ and a test pattern $t(x)$. In the case that there exists relative shifts between $r(x)$ and $t(x)$, it is applicable to compute the cross-correlation between the two patterns for various possible shifts τ as in Equation (8). Then, the maximum value is selected as the similarity between the two patterns.

$$c(x) = \int t(\tau) r(\tau + x)\, d\tau. \tag{8}$$

In biometric verification application, often, a synthetic filter is built as a template using a number of training samples [15]. During matching stage, a cross-correlation between a filter (template) and a query sample will produce an array of numbers, named the correlation output, which is referred to the inner products of different shifted versions of the template with the query. When a query perfectly matches with the filter, a well-defined peak will appear in the output correlation plane. Otherwise, a relatively flat correlation output is expected to be observed.

Correlation and the relative operations can be performed extremely fast in the frequency domain. Often, a biometric feature, represented as a spatial domain vector or a matrix, can be transferred to the frequency domain using the Fourier Transform (FT). During enrolment, a correlation filter is obtained and represented in frequency-domain format. During verification, a query feature will be transformed to the frequency domain using FT and then be multiplied by the pre-stored correlation filter of the claimed person. The inverse FT (IFT) of the products is taken to obtain the final correlation output [15].

3. PROPOSED APPROACH

3.1. Design a CPR-oriented equivalent of *n*-graph disorder

CPR has been well developed and widely recognized to be an effective method for biometric authentication. However, the selection of typing features will significantly affect the authentication performance. Once a pattern recognition method is given, seeking for a good feature becomes the most critical task.

3.1.1. An instinctive attempt

In a previous work [12], directly using CPR methods with the absolute values of keystroke durations tended to be unsuccessful (unacceptable performance compared with that of the GP method). The main reason is because the performance of keystroke durations, in terms of uniqueness, discrimination, reliability and robustness, falls below the *n*-graph DoD which is adopted by the GP method. The best solution is applying *n*-graph DoD to CPR methods. However, it does not seems to be an easy task.

The instinctive and direct way is to use *n*-graph sorting indices but it is not working. Take the example using in Section 2.2:

s1 : co 30, om 20, mp 10, pu 70, ut 50, te 60, er 40,

s2 : co 30, om 20, mp 45, pu 48, ut 50, te 60, er 40.

s1 and **s2** are similar samples since five out of all seven digraphs are identical and only two are different. The example demonstrates a quite ideal scenario, and real-world situations may be even a lot worse. In our experimental data set, normally two samples only have less than 30% 'close' digraphs and no identical digraph can be found.

s1 and **s2** can be represented directly by a set of points, which are

s1 : {(co, 3), (om, 2), (mp, 1), (pu, 7), (ut, 5), (te, 6), (er, 4)},
s2 : {(co, 2), (om, 1), (mp, 4), (pu, 5), (ut, 6), (te, 7), (er, 3)},

where for each point, x-axis value is the n-graph notation, and y-axis value represents the corresponding sorting index number.

When plotting the two point sets in the Cartesian coordinate system, it is clear that **s1** and **s2** does not show any similarities from signal processing's point of view (Fig. 2). The signal **s2**, apparently, is not a linear transformation of **s1**. Besides, **s1** and **s2** does not have any overlapping points. That means correlation-based approach will not work with n-graph indices directly. Note that in **s2**, two distorted digraphs, mp and pu, change the indices of all the other digraphs. This will not affect the DoD metric obviously because the distance between each corresponding digraph will be averaged at the end. Nevertheless, CPR methods are not capable of tolerating such a kind of distortion in which all points of two signals are markedly different.

3.1.2. *The equivalent of n-graph disorder—nGDV*

This work is inspired from the CPR algorithms used for binary image-based biometric recognition such as face and fingerprint [17]. The pixel of a binary image can be represented by a single binary bit (0 and 1) and hence, the pattern of an image is represented by a set of 0 and 1 (Fig. 3). CPR-based methods not only take account of the y-axis values of pixels but also rely on the sequence/relation between each pixel. Note that in a binary image, the y values of any pixel are either 0 or 1; so the pixel sequence significantly determines the pattern. For example, (1, 1, 1, 0, 0, 0) and (1, 0, 1, 0, 1, 0) are different patterns. Inspired by this, we attempted to extract binary information from a n-graph typing sample, in order to replace the n-graph sorting indices.

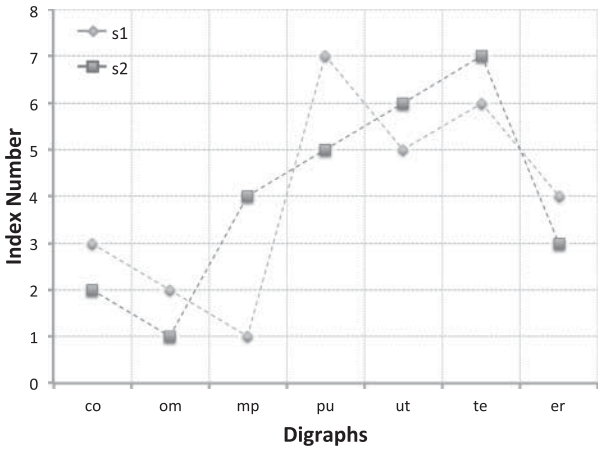

FIGURE 2. Correlation using digraph sorting index.

Suppose there are three digraphs:

$$\text{om } 30,\ \text{mp } 10,\ \text{pu } 70.$$

And the corresponding sorting index table is

Digraph x	Idx y
om	2
mp	1
pu	3

We extract three inequalities from Equation (9), which are

$$\begin{aligned} \text{mp} &< \text{om} \\ \text{mp} &< \text{pu} \\ \text{om} &< \text{pu}, \end{aligned} \tag{9}$$

where these three inequalities are the sufficient and necessary condition of the digraph's sequence shown in the index table mentioned earlier. It is definitely possible to reconstruct the sorting index table from Equation (9), and vice versa. So we say that they are exactly equivalent. The inequalities can be rewritten in the following list:

NO. x	Inequality	True/false	Encoding y
1	om > mp	True	2
2	om > pu	True	2
3	mp > pu	False	1

Each digraph pair will be compared. If the corresponding inequality holds, we encode it with '2', otherwise, with '1'. The three inequalities om $>$ mp, om $>$ pu, mp $>$ pu can be described by three 2D points (1,2), (2,2), (3,1), respectively. We put the numbers 2, 2, 1 in a column vector and name it as *n-graph Disorder Vector* (*n*Gdv). A typing sample containing N digraphs can be described by $N \cdot (N-1)/2$ inequalities and hence its *n*GDV consists of $N \cdot (N-1)/2$ elements. Note that $N \cdot (N-1)/2$ is the minimum number that is required to describe a typing sample. Using less than $N \cdot (N-1)/2$ inequalities may not be able to record all n-graph relations (information loss) while using more than $N \cdot (N-1)/2$ inequalities will incur redundant information.

Thus, **s1** and **s2** can be represented using digraph inequalities, as:

NO.	Inequality	S1		S2	
1	co > om	True	2	True	2
2	co > mp	True	2	False	1
3	co > pu	False	1	False	1
4	co > ut	False	1	False	1
5	co > te	False	1	False	1
6	co > er	False	1	False	1
7	om > mp	True	2	False	1
8	om > pu	False	1	False	1
9	om > ut	False	1	False	1
10	om > te	False	1	False	1
11	om > er	False	1	False	1
12	mp > pu	False	1	False	1
13	mp > ut	False	1	False	1
14	mp > te	False	1	False	1
15	mp > er	False	1	True	2
16	pu > ut	True	2	False	1
17	pu > te	True	2	False	1
18	pu > er	True	2	True	2
19	ut > te	False	1	False	1
20	ut > er	True	2	True	2
21	te > er	True	2	True	2

And,

$$\mathbf{nGdv}_{s1} = [2\ 2\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 1\ 2\ 2]^{\mathrm{T}},$$

$$\mathbf{nGdv}_{s2} = [2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 2\ 1\ 2\ 2]^{\mathrm{T}}.$$

Now plot the $n\mathrm{Gdv}_{s1}$ and $n\mathrm{Gdv}_{s2}$ in the Cartesian coordinate system and yield Fig. 4. It can be seen that most (16 out of 21) points of **s1** and **s2** are overlapping. In comparison with the n-graph sorting index, nGdv possesses the following advantages:

(1) Discernible Pattern
The usage of binary bit makes nGdv close to the vector representation of a binary image, shown in Fig. 3. It means nGdv can perfectly fit CPR methods.

(2) Reliable and Robust
The distortion of a single n-graph will no longer bring strong impact on the other n-graphs. In the example, the distortion of digraph mp and pu only change the results of 24% inequalities, i.e. co > mp, om > mp, mp > er, pu > ut, pu > te. The rest of 76% inequalities do not change.

One may notice that the earlier-mentioned nGdv contains only two values '1' and '2', not '0'/'1' in binary image CPR methods. In the proposed scheme, '2' indicates the inequality holds while '1' has the opposite meaning. '0' is used to represent the situation that an inequality exists in a template but not in the query sample. Figures 5 and 6 illustrate the correlation output of a genuine test and an impostor test, respectively. Note that Figures 5 and 6 are the graphs of cross-correlation function. For a test sample f (a collection of inequalities) and a template r, the cross-correlation (discrete) is defined as

$$y[x] = \sum_{p=1}^{2N-1} f[p] \cdot r[p+x],$$

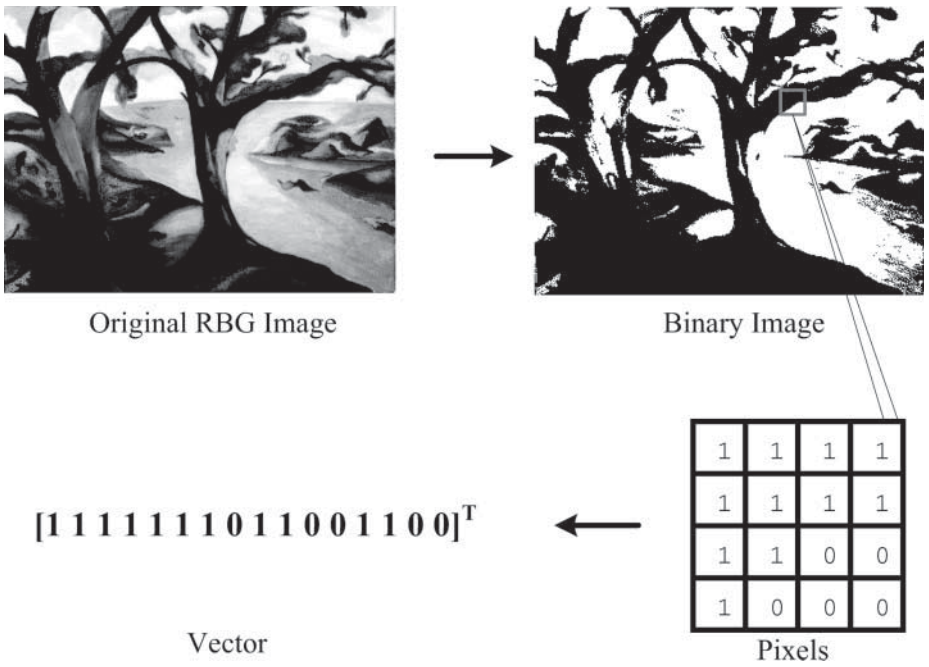

FIGURE 3. Binary image in CPR methods.

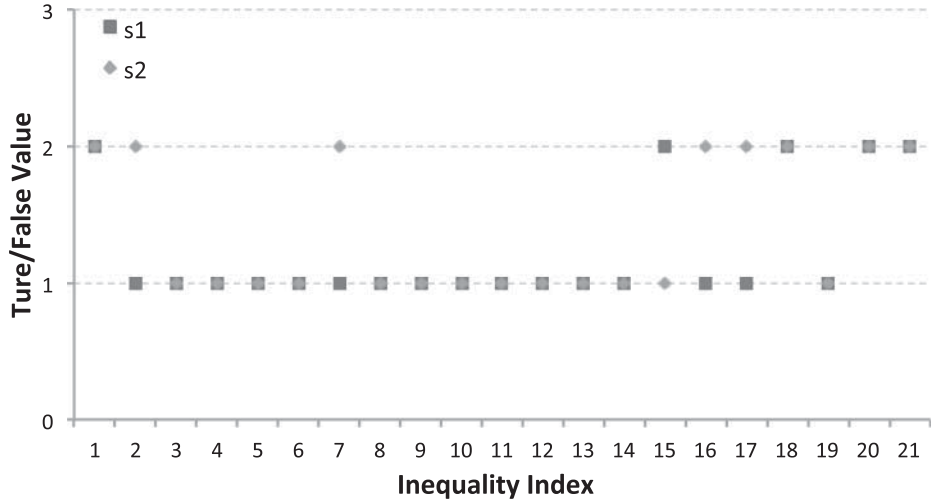

FIGURE 4. nGdv of s1 and s2.

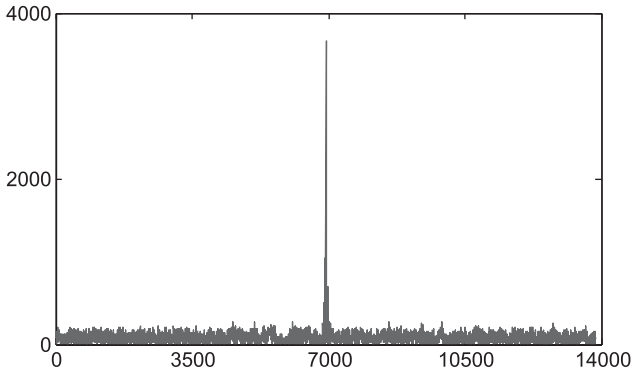

FIGURE 5. Correlation output of a genuine test using keystroke nGdv.

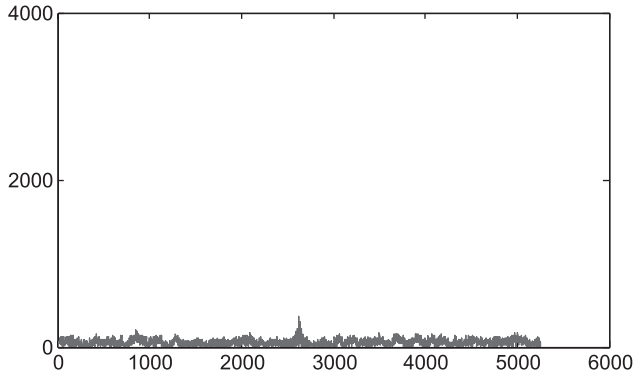

FIGURE 6. Correlation output of an imposter test using keystroke nGdv.

where N is the number of common inequalities, x is the offset between f and r, and y is the correlation output. For instance, given $f = [1\,2\,3\,4]$ and $r = [4\,6\,2\,7]$, $y[2] = 1 \cdot 2 + 2 \cdot 7 = 16$.

It is clear that a sharp correlation peak appears in the genuine test's figure. Therefore, genuine users and impostors can be distinguished via utilizing nGdv.

3.2. Correlation keystroke pattern recognition adopting nGdv

We propose two ways of utilizing correlation for authenticating a person through the keystroke dynamics. One is using correlation for verification directly and the other is for classification.

3.2.1. nGdv correlation verification (nGdv-V)

nGdv-V is a CPR-based matching scheme that verifies a person directly based on his typing features nGdv. When a user presents his input, the typing feature (a query sample) will be compared with the pre-stored feature (template) of the person he claimed. Usually there are two stages: enrolment stage and verification stage.

Enrolment: During Enrolment, the user (subject) u_i registers m typing samples for the purpose of template design. Each sample is denoted as s_{ij}, ($j \in [1, m]$). Either fixed text typing or free text typing can be employed, where the former means that the user should type a pre-designed text, and the latter offers the user freedom of typing any words and sentences he wants. n overlapping n-graphs are then extracted from the samples. Each n-graph should appear in every sample. In general, enrolment process is conducted offline. If choosing the free text manner, the user is expected to type as many characters as possible. The corresponding $n\text{Gdv}_{ij}$ is extracted from each sample s_{ij}, where $n\text{Gdv}_{ij}$ contains $n \cdot (n-1)/2$ elements (inequities). All $n\text{Gdv}_{ij}$ will then be combined to a single template, referring to correlation filter.

The design of the correlation filter vitally determines the verification performance. The simplest correlation filter is matched filter, which provides the maximum signal-to-noise

ratio (SNR) in detecting a known reference pattern in the presence of additive white noise. However, matched filters are extremely sensitive to distortion in template [15]. Therefore, each training sample should have one corresponding matched filter. During verification the query samples need to be cross-correlated with all matched filters, which tends to be computationally inefficient. To overcome the drawbacks, the synthetic discriminant function (SDF) filter, a linear combination of multiple training samples, was proposed. During verification the query samples only need to be cross-correlated with a single filter. However the performance of the SDF was not good since it produced correlation outputs where the sidelobes are larger than the controlled values at the origin [15].

The MACE filter [18] is designed to minimize the average correlation plane energy while constraining the certain pre-specified value at the origin. It suppresses the sidelobes of correlation plane such that a sharp correlation peak can be produced. MACE filter is capable of providing a good discrimination without the need for impostor training samples. The MACE filter is selected in our scheme.

Let $\mathbf{x}_{ij} = \mathbf{nGdv}_{ij}$, where $\mathbf{x}_{ij}$ has d elements ($d = n \cdot (n-1)/2$). A matrix $\mathbf{X_i}$ from m training samples is defined as

$$\mathbf{X_i} = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{im}]^{\mathrm{T}}. \tag{10}$$

The MACE filter obtained in the frequency domain is also ordered in a column vector $\mathbf{h}_i$. The jth correlation output at the origin is constrained to a pre-specified value u_{ij}, which can be represented as

$$c(0) = \mathbf{x}_{ij}^{+}\mathbf{h}_i = \mathbf{h_i}\mathbf{x}_{ij}^{+} = u_{ij}, \quad j \in [1, m], \tag{11}$$

where the superscript $+$ denotes a conjugate transpose. Note that $c(0)$ is also referred to the correlation output peak value.

Based on ParsevalÕs theorem, the average of the correlation plane energy, $\mathbf{E}_{\mathrm{avg}}$, can be obtained directly from the frequency domain by

$$\mathbf{E}_{ij} = \sum_{p=1}^{d} |c_{ij}(p)|^2 = \sum_{k=1}^{d} |\mathbf{h}_i(k)|^2 |\mathbf{x}_{ij}(k)|^2 = \mathbf{h}_i^{+}\mathbf{x}_{ij}\mathbf{x}_{ij}^{*}\mathbf{h}_i, \tag{12}$$

$$\mathbf{E}_{\mathrm{ave}} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{E}_{ij} = \mathbf{h}^{+}\left[\frac{1}{m}\sum_{j=1}^{m}\mathbf{x}_{ij}\mathbf{x}_{ij}^{*}\right]\mathbf{h}_i = \mathbf{h}_i^{+}\mathbf{D}\mathbf{h}_i, \tag{13}$$

where the superscript '$*$' denotes complex conjugation and $\mathbf{D}$ is a diagonal matrix of size $d \times d$ whose diagonal elements are the power spectrum of x_{ij}.

Minimizing the average correlation energy $\mathbf{E}_{\mathrm{avg}}$ subjecting to the constraints placed in Equation (11) leads to the MACE filter solution

$$h_i = \mathbf{D}^{-1}\mathbf{X}_i(\mathbf{X_i}\mathbf{D}^{-1}\mathbf{X}_i)^{-1}u, \tag{14}$$

where $u = [u_{i1}, u_{i2}, \ldots, u_{im}]^{\mathrm{T}}$. h_i, represented in a frequency-domain form, is the typing template stored in the system for verification.

Verification: At the verification stage, a user $u_{t'}$ provides his typing feature, as a query, to the system in either fixed text or free text manner. The system will verify him via comparing the query with the template of the person he claims, say u_t. If choosing free text, the n-graphs that exist in the stored template will be extracted from the query. The query's n-graph disorder vector, $n\mathrm{Gdv}_{t'}$, is obtained from the query sample. In terms of the vector size and the sequence of inequalities, $n\mathrm{Gdv}_{t'}$ should be coincident with the template vector $n\mathrm{Gdv}_t$, which is generated from the training sample of u_t. In case that an inequality relationship that exists in the templates u_t but does not appear in the query sample, we still include this relationship in $n\mathrm{Gdv}_t$. In this case, this relationship is assigned a '0' for the purpose of padding.

$n\mathrm{Gdv}_{t'}$ is transformed to the frequency-domain by using FT and multiplied pair-wise by the stored filter. The resulting product vector is input to an inverse FT to produce a correlation output. In theory, the correlation output should demonstrate a sharp peak if $u_{t'}$ is u_t and no obvious peak otherwise.

Generally, the sharpness of a correlation peak is measured by the performance metric called peak-to-correlation energy (PCE) [19]. The larger the PCE, the sharper the peak. PCE is defined as the ratio of the correlation peak and the energy in the correlation plane, defined as

$$\mathrm{PCE} = \frac{|c(0,0)|^2}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}|c(x,y)|^2\,dx\,dy}. \tag{15}$$

If PCE is above a pre-defined threshold, a successful authentication is declared and a failed authentication is declared otherwise. By adjusting the threshold, the system performance, indicted by FAR/FRR, can be different. The verification process is depicted in Fig. 7.

3.2.2. *nGdv correlation classification (nGdv-C)*

Classification/clustering-based verification approach is often deployed for the problem of biometric identification within

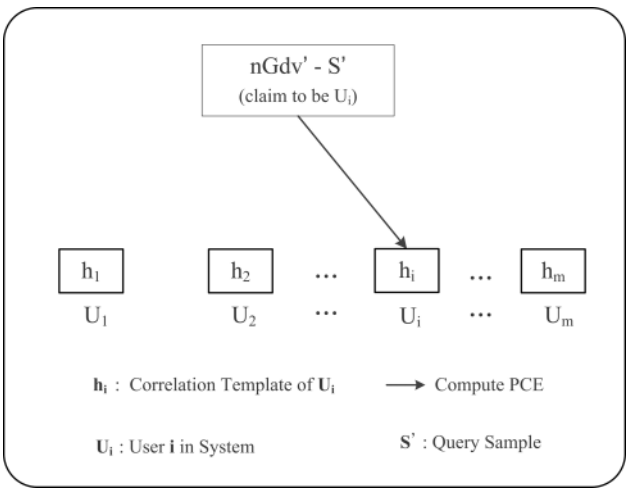

FIGURE 7. User authentication process of nGdv-V verification algorithm.

a large database where the input is unknown [6]. Due to the high computational efficiency and low complexity, CPR can be employed as a classification/clustering approach, incorporated with DoD scheme. The proposed nGdv Correlation Classification (nGdv-C) algorithm consists of the following two stages: (1) Enrolment and Clustering (2) Verification.

Enrolment and clustering: At this stage, user i registers m typing samples s_{ij} in the system. The MACE filter h_i is generated from s_{ij}, following the same procedure described in Section 3.2.1. Both s_{ij} and h_i are stored in the system where s_{ij} serve as template and h_i is used for classification rather than verification. In terms of classification/clustering, two situations should be considered:

(1) *Cluster Building*: No cluster exists in the system. New clusters will be built for all existing users, including the user i.
(2) *Cluster Updating*: All existing users have been put in clusters already. The user i should be inserted into a proper cluster.

For **situation 1**, firstly calculate the similarities between every two users, where the similarity is measured by average PCE (denoted as $\overline{\text{PCE}}$). Given two users u_a and u_b, u_a has m_a typing samples $\{s_{aj}\}_{j=1}^{m_a}$ and the correlation template h_a. u_b has m_b typing samples $\{s_{bk}\}_{k=1}^{m_b}$ and the correlation template h_b. The average PCE is computed as

$$\overline{\text{PCE}_{a\_b}} = \frac{1}{m_a + m_b}\left(\sum_{j=1}^{m_a}\text{PCE}(n\text{Gdv}_{aj}, h_b) + \sum_{k=1}^{m_b}\text{PCE}(n\text{Gdv}_{bk}, h_a)\right) \quad (16)$$

The bigger the $\overline{\text{PCE}}$ value, the higher the similarity and the shorter the distance. Agglomerative hierarchical clustering algorithm is utilized for establishing clusters, the process of which is based on the union between the two closest clusters. In initial status, each user is considered as a single cluster. After a few iterations, it converges to the final clusters wanted. Basically it is a bottom-up process. Average-linkage clustering criterion is selected, in which the distance d between cluster A and cluster B is equal to the average distance from any member of A to any member of B, as

$$d_{A\to B} = \text{average}(\overline{\text{PCE}_{i\_j}}), \quad (\forall i \in A,\ \forall j \in B). \quad (17)$$

The final number of clusters p is equal to the number of existing users in the system, n, minus the times that clustering iterations operated.

Each cluster C_i ($i \in [1, p]$) should have a representative rp, which is selected using the criteria as

$$\forall k \in C_i \setminus rp : \sum_{t\in C_i\setminus rp}\overline{\text{PCE}_{rp\_t}} \geq \sum_{t\in C_i\setminus k}\overline{\text{PCE}_{k\_t}}. \quad (18)$$

These criteria ensures that rp is close to all the other users within the cluster.

For **situation 2**, i is first compared with the representative of each cluster in order to find the closest cluster. Then i is added into the cluster. If necessary, reselect the representative on the basis of Equation (18).

Verification: When a user t' attempts to log in, his sample $n\text{Gdv}_{i'}$ will first be compared with each cluster representative in the system using correlation. The process of comparison, actually, is looking for the closest representative that produces the maximum PCE value. Given p clusters, one user query only needs to compute PCE p times. The cluster, to which the closest representative belongs, is selected as the candidate cluster. Afterwards, GP method will be applied to compare the DoD between t' with all users that in the candidate cluster. Note that it is allowed to select more than one candidate cluster, to reduce the clustering error.

The verification process is depicted in Fig. 8.

3.2.3. *Discussion*

In theory, both nGdv-V and nGdv-C should be much faster than GP method. Note that GP method is an identification-like scheme in which each incoming query sample need to go through the entire system. As a real-world cloud computing environment, Microsoft Azure claimed to have over 10 000

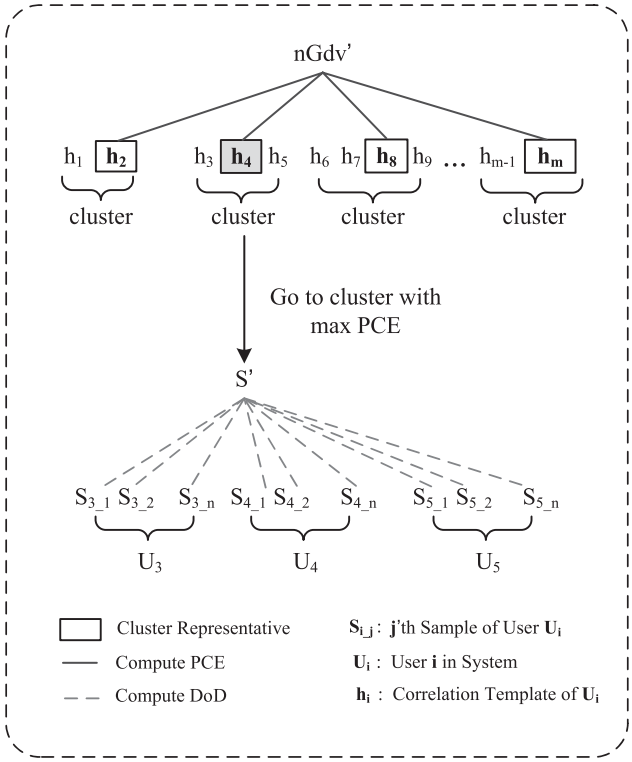

FIGURE 8. Authentication process of nGdv-C.

users [20]. Suppose 1) each user has 15 samples (templates), and 2) comparing (compute DoD) one query sample with one template sample takes 0.1 s (based on our experiment), then the total time of authenticating one user is $10\,000 \times 15 \times 0.1 = 15\,000\,\text{s} \approx 4.16\,\text{h}$, which is definitely unacceptable! If using *n*Gdv-V, a query sample only need to compare with the claimed user. The authentication time is the duration of calculating PCE once, normally 0.002 second (plus some negligible overhead time of extracting *n*Gdv and generating inequalities). However in terms of accuracy, *n*Gdv-V may not be as good as GP because *n*Gdv-V uses a pre-defined threshold. As a combination of *n*Gdv-V and GP method, *n*Gdv-C inherits the advantages from both, i.e. *n*Gdv-V's high computation efficiency and GP's accuracy. Taking Microsoft Azure [20] for example, suppose 10 000 users is clustered into 500 clusters and each cluster has 20 users. The total authentication time is the summation of classification time plus DoD computation time, i.e. $500 \times 0.002 + 20 \times 15 \times 0.1 = 31\,\text{s}$, which still seems to be more feasible than GP method.

4. EXPERIMENT

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. The following metrics are used:

(1) *False Acceptance Rate (FAR)*
The percentage of an impostor that managed to login to the system, defined as

$$\text{FAR} = \frac{\text{Number of Accepted Impostor Test}}{\text{Number of Total Impostor Test}}.$$

(2) *False Rejection Rate (FRR)*
The percentage of a valid user that is being denied an authentication, defined as

$$\text{FRR} = \frac{\text{Number of Rejected Genuine Test}}{\text{Number of Total Genuine Test}}.$$

(3) *Time Efficiency*
The time consumption for a user to perform a single authentication. It refers to the elapsed time between a user submitting the typing sample and the system providing the authentication result as output.

FAR and FRR are aimed to be as low as possible, ideally 0% for both. However nearly all biometry applications can never reach this limit. In the biometric systems that using matching threshold, FAR and FRR are usually inversely proportional to each other. Reducing FAR will result in an increase of FRR, and vice versa. Although people sometimes allocate more preference on one of these rates, a tradeoff 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. Normally, a balanced threshold value will be explored through trial and error technique.

Time metric is added to measure the computational efficiency. We recorded the time taken to perform the entire genuine test and then divide it by the number of legitimate authentications made. By doing this, the average time for each authentication was obtained.

4.1. Experiment setting

Keystroke samples adopted in our experiment was the database used by [14]. Free text keystroke samples were gathered, i.e. the volunteers were allowed to type any text they wanted instead of a pre-designed text, during the sample collection process. Typing samples were written in Italian, provided by staff or students (all are native Italian speakers), from the Department of Information, University of Torino. Typing the same word and phrase repeatedly is not allowed; however, typing errors can be corrected. The typing samples' collection process were conducted on different platforms (Windows or UNIX) using either Internet Explorer or Netscape. A JavaScript in a simple HTML form was utilized to collect keystroke data [14]. The format of the collected sample can be expressed as the time (in μs) when a key is pressed, followed by the ASCII value (numbers in bold in the following example) of the key. For example, a segment of a typing sample is:

$$\{76405\ \mathbf{65}\quad 76584\ \mathbf{78}\quad 76914\ \mathbf{67}\quad 99909\ \mathbf{19}\}.$$

In this example, it can be seen that key A represented by ASCII 65 is pressed at time 76 405. The original database consists of 40 internal users (subjects) with 15 typing samples per subject. Another 165 external users, each with only one typing sample, act as impostors to the system. Due to the privacy issue, the database owner published only the keystroke samples of 21 internal subjects and 165 impostors.

A Matlab program is used to implement the GP method for authentication performance evaluation. Gunetti and Picardi [14] proposed two methods 'R' measure and 'A' measure, where the former uses *n*-graphÕs relative timing information (DoD) and the latter uses absolute timing values of *n*-graph. Since 'R' provides high authentication performance than 'A', we only investigated 'R' measure in this paper. Furthermore, only digraph is adopted as keystroke timing patterns.

In the scenario of free text testing, it is important that two typing samples under comparison must have the same *n*-graphs. However, in the database some pair of samples only share very limited digraphs, e.g. s_{11} only has four overlapping digraphs with s_{21}. Thus a pre-defined threshold T_{ovlp} on the number of overlapping digraph N_{ovlp} is defined. For each sample pair, if $N_{\text{ovlp}} < T_{\text{ovlp}}$, samples are directly considered to be unmatched, and otherwise continue verification process using 'R' method.

4.2. Performance evaluation

Re-implement and test GP method: As mentioned in Section 1.1, Gunetti *et al.* incorrectly calculate the error rate during the evaluation of GP method. The paper [14] states '... Hence, on the whole, the system is tested with 600 legal connection attempts and with 450 000 impostor attacks brought by 40+165 individuals ($600\times39\times15+165\times40\times15 = 450\,000$)'. Apparently, a query can only claim that it belongs to one user/subject, rather than to a certain training sample of the user. Therefore, 40 subjects (each contributes 15 attempts and each attempt can claim as any one of the 39 subjects) can only generate ($40 \times 15 \times 39 = 23,400$ impostor attempts). In the same way, 165 external users can generate ($165 \times 40 = 6600$ impostor $attempts$). The total impostor attacks should be $23\,400 + 6600 = 30\,000$ instead of 450 000 as claimed in the paper. Hence, the correct value of FAR should be 15 times larger than the value claimed by the authors, i.e. FAR $= 0.125\% \times 15 = 1.875\%$ when applying 'R' measure to digraph.

We reimplemented GP method and tested on the dataset of 21 internal subjects and 165 external subjects. The FAR is 1.34% and FRR is 11.22%. The average time consumption for authenticating one query is 3.532 s. Note the FAR, which is of the same order of magnitude as 1.875%.

Evaluation of nGdv-V: Firstly we evaluated the *n*Gdv-V, the experimental result of which is shown in Table 2. Here five FAR-FRR pairs are listed. Although *n*Gdv-V is not as accurate as GP method, the average time consumption of verifying a query only took 0.0028 s, which is 1000 times faster than GP! We investigated the experimental result and discovered that the high error rate is due to two major reasons:

(1) *n*Gdv-V is a fixed threshold algorithm, while GP is an identification-like method. GP computes DoD between a query and all users, and always considers the smallest DoD comes from the right person. The case that two queries (from different users) both successfully match with the same template will never happen ('smallest' implicitly means 'one'). However, for *n*Gdv-V, it is common that two queries both pass or fail the authentication. In other words, the matching process of GP is restricted to 1:1 matching, while *n*Gdv-V is a M:N matching.

TABLE 2. Performance of *n*Gdv-V.

Method	FAR (%)	FRR (%)	Computing time (sec.)
GP [14]	1.34	11.22	3.532
*n*Gdv-V	42.3	1.89	
	15.81	11.3	
	9.43	24.7	**0.0028**
	6.78	32.2	
	1.09	60.18	

(2) As a CPR method, *n*Gdv-V shows a great advantage of dealing with genuine test. When control the threshold properly, the query sample from a genuine user will not miss easily. Nevertheless compared with GP, *n*Gdv-V is not much capable of filtering impostors. Reducing the threshold can prevent the impostors; however, the genuine acceptance rate will deteriorate as well. On the contrary, GP can effectively stop the impostors but it cannot recognize the genuine user with a high successful rate.

The 1000 times speed gain of *n*Gdv-V is important and encouraging. In designing a large-scale system such as computing cloud, the concerns of computational efficiency and usability are given the same consideration as security.

Evaluation of nGdv-C: The performance of nGdv-C method varies as two parameters change: (1) total number of clusters that in a system, denoted as p (2) the number of candidate clusters using for succeeding verification, denoted as n. Hence in our experiment, we assigned different values to such p and n in order to investigate the performance upper bound as well as study how the p and n affect the system performance.

We use the notation *n*Gdv-C(p, n) for the situation that *n*Gdv-C method is given specific parameters. Table 3 demonstrates the performance of proposed *n*Gdv-C method. The most exciting performance result appears in *n*Gdv-C(17,4), in which all users are distributed into 17 clusters, and 4 closest clusters are used for verification. FAR = 1.65% and FRR = 2.75% are achieved. In comparison with GP method, *n*Gdv-C(17,4) presents a very close FAR but dramatically reduces FRR from 11.22 to 2.75%. In addition, *n*Gdv-C(17,4) reduces the authentication time from 3.53 to 1.08 s. The authentication speed gain can be another success.

Table 3 also shows that in a system when n increases, the authentication performance will be more accurate; however, authentication time will be longer. This sounds reasonable since the use of more clusters reduces the probability of clustering error (query sample drops into a wrong cluster during classification). The sacrifice is the computational time. More clusters, more time consumption.

We conducted the experiment in order to investigate the influence of number of clusters on the system performance, the result of which is shown in Table 4. In the experiment, n were fixed to 3 and p were set to 3, 7, 11, 16, 20. It shows an interesting result that FAR values are almost invariable but FRR and time gets reduced when p increases. Note that *n*Gdv-C(3,3) shows almost the same performance as GP method. It makes sense that *n*Gdv-C is almost equivalent to GP when $m = n$. In such situations, all clusters/users are involved in succeeding verification and thus clustering does not take effect. Another extreme scenario is that every single user forms a cluster. In such cases, Gdv-C is almost equivalent to *n*Gdv-V yielding the fastest authentication speed.

TABLE 3. Performance of *n*Gdv-C (same number of total clusters, different number of candidate clusters).

Method	Cluster (Total m)	Cluster (Top n)	FAR (%)	FRR (%)	Time (s)
GP [14]	–	–	1.34	11.22	3.532
*n*Gdv-C	17	1	2.58	12.64	**0.25**
	17	2	1.74	6.04	0.54
	17	3	1.69	3.30	0.81
	17	4	**1.65**	**2.75**	1.08

TABLE 4. Performance of *n*Gdv-C (same number of candidate cluster, different number of total clusters).

Method	Cluster (Total m)	Cluster (Top n)	FAR (%)	FRR (%)	Time (sec.)
GP [14]			1.34	11.22	3.532
*n*Gdv-C	3	3	1.34	11.22	3.78
	7	3	1.38	9.65	1.71
	11	3	1.42	6.12	1.27
	16	3	1.41	5.61	0.85
	20	3	1.37	4.66	0.58

Overall, the experimental results show that the *n*Gdv-C significantly improves FRR and authentication times with a very close FAR in comparison with GP method. The high accuracy as well as fast speed make *n*Gdv-C one of the most promising keystroke solution for UAC in clouding computing environment.

5. CONCLUSION

In this paper, we investigate the feasibility of adopting keystroke for UAC in cloud computing environment. A new n-graph equivalent keystroke feature *n*Gdv has been introduced. Two correlation-based keystroke verification schemes (*n*Gdv-V and *n*Gdv-C) are proposed. When compared with the classical GP method, *n*Gdv-V features extremely fast speed and the *n*Gdv-C outperforms in both accuracy (low FRR) and computational efficiency. The experimental results can further help in advancing keystroke technology towards practical applications.

FUNDING

This research is supported by ARC (Australia Research Council) Projects LP110100602, LP100200538, LP100100404 and DP0985838.

REFERENCES

[1] http://www.cloudtweaks.com/2010/08/the-top-10-cloud-computing-trends/.

[2] http://fcw.com/articles/2009/06/22/tech-cloud-security.aspx.

[3] Jain, A.K. and Maltoni, D. (2003) *Handbook of Fingerprint Recognition*. Springer, New York, Inc., Secaucus, NJ, USA.

[4] Xi, K., Ahmad, T., Han, F. and Hu, J. (2010) A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. *Secur. Comm. Netw.*, http://dx.doi.org/10.1002/sec.225.

[5] Jain, A., Hong, L. and Bolle, R. (1997) On-line fingerprint verification. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**, 302–314.

[6] Wang, Y., Hu, J. and Phillips, D. (2007) A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular-point detection and fingerprint indexing, *IEEE Trans. on Pattern Anal. and Mach. Intell.*, **29**, 573–585.

[7] Xi, K. and Hu, J. (2009) Biometric Mobile Template Protection: A Composite Feature Based Fingerprint Fuzzy Vault. In *ICC '09. IEEE Int. Conf. Communications*, Dresden, Germany, June 14–18, pp. 1–5. IEEE, Dresden, Germany.

[8] Middleton, L., Wagg, D.K., Bazin, A.I., Carter, J.N. and Nixon, M.S. (2006) Developing a Non-Intrusive Biometric Environment. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Beijing, China, October 11–13, pp. 723–728. IEEE, Beijing, China.

[9] Hu, J., Gingrich, D. and Sentosa, A. (2008) A k-Nearest Neighbor Approach for User Authentication Through Biometric Keystroke Dynamics. *IEEE Int. Conf. Communications*, Beijing, China, May, pp. 1556–1560. IEEE, China.

[10] Monrose, F., Reiter, M.K. and Wetzel, S. (1999) Password Hardening Based on Keystroke Dynamics. In *Proc. 6th ACM Conf. Computer and Communications Security*, Singapore, pp. 73–82. ACM, New York, NY, USA.

[11] Biopassword. http://www.biopassword.com.

[12] Bergadano, F., Gunetti, D. and Picardi, C. (2002) User authentication through keystroke dynamics. *ACMTrans. Inf. Syst. Secur.*, **5**, 367–397.

[13] Lau, E., Xiau, C., Liu, X. and Yu, X. (2004) Enhanced User Authentication Through Keystroke Biometrics. Technical Report. MIT.

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

[15] Kumar, B.V.K.V., Mahalanobis, A. and Juday, R.D. (2005) *Correlation Pattern Recognition*. Cambridge University Press, New York, NY, USA.

[16] Sentosa, A. (2007) User authentication through keystroke dynamics. PhD Thesis, RMIT University.

[17] Kumar, B.V.K.V. and Hassebrook, L. (1990) Performance measures for correlation filters. *Appl. Opt.*, **29**, 2997–3006.

[18] Mahalanobis, A., Kumar, B.V.K.V. and Casasent, D. (1987) Minimum average correlation energy filters. *Appl. Opt.*, **26**, 3633–3640.

[19] Wang, Y., Hu, J., Xi, K. and Bhagavatula, V. (2007) Investigating correlation-based fingerprint authentication schemes for mobile devices using the j2me technology. In *IEEE Workshop on Automatic Identification Advanced Technologies*, Alghero, Italy, June 7–8. IEEE, Piscataway, USA.

[20] http://www.zdnet.com/blog/microsoft/microsoft-passes-the-10000-customer-milestone-with-azure/6433.