Utilizing Keystroke Dynamics as an Additional Security Measure to Password Security in Computer Web-based Applications - A Case Study of UEW

Osei Boakye Michael
Dept. Computer Science
KNUST-KUMASI

Yaw Marfo Missah
Dept. Computer Science
KNUST-KUMASI

ABSTRACT
Keystroke Dynamics is one of the well-known and economical behavioral biometric developments that attempt to recognize the genuineness of a client when the client invokes his keystrokes from a computer keyboard. The keystrokes pattern helps to determine the typing behaviour of users of the system thus serves as the benchmark for identity verification. This paper displays the use of biometrics to augment traditional passwords security in computer web-based application systems. The system was evaluated base on these criteria; authenticating legitimate user against imposter user and a guess imposter user of the system. Also, evaluation of character timings was performed to know the best combination of strings to use in setting passwords in systems where keystroke dynamics would be applied in order to achieve high efficiency. This study concludes that the use of keystrokes dynamics in augmenting password security in computer web-based applications should be embraced.

General Terms
Keystroke dynamics web-based application.

Keywords
Keystroke dynamics, password, biometrics, web-based application, Terminus time, Source time.

1. INTRODUCTION
The universally accepted use of information system and computers has made humanity to carry out their daily activities with less effort. Advancement in information technologies for few decades now has made possible improvements in network performance, reliability and accessibility as well as reducing operating costs by adapting to these efficient technologies [1]. However, the ubiquitous nature of computer technologies and other accessibility has unveiled new threats to internet security.

Efficient security measures are being sought now to secure computer resources against unauthorized access through fraudulent and masquerading in web applications. Online based applications are developed to allow individuals who have access to internet and web browser to access the content of the particular web applications without any platform dependent problems. Users of a system enter their credentials to log into a system in order to have authorized access to whatever content stored. The authentication credentials need to be secured properly to reduce system vulnerabilities which could be exploited by imposters.

There are various methods currently used in authenticating system users. The most commonly used is the password. Passwords are appropriate as they are easily employed in software and require no dedicated hardware. Users are also at ease with their use. On the other hand, passwords also come with many flaws. Users frequently share passwords, fail to recall passwords, and select weak passwords that may be easily cracked.

Authenticating users to web application systems becomes a problem due to these threats to computer and internet security [5]. The traditional measures used to safeguard credentials of legitimate users of a system such as passwords and pins are no longer much reliable as an efficient authentication approach to some extent. Hence a new secured methodology that must be probed into is biometric techniques. As compared to password and pin methods, biometrics technologies make use of physiological and behavioural traits to identify legitimate users to a system. Biometric authentication system is the appropriate and most secured authentication method, because it is practically impossible to be borrowed, stolen, and forged. Biometrics, specifically keystroke dynamics is not in to eliminate the usefulness of passwords and pins but to augment their efficiency as security methods. The aim of this paper is to determine the effectiveness of biometric keystroke dynamics that identifies unique patterns in the typing behavior of users used together with passwords as authentication measure to solve the various authentication issues in online applications. Specifically the objectives of this research were to: determine the effectiveness of keystroke analysis and password security synergy to authenticate users of a system, ensure the avoidance of password sharing compromise and ensure non-invasiveness into computer web-based applications through keystroke analysis and password security synergy.

2. LITERATURE REVIEW
In a study by Monrose and Rubin; utilizing Keystroke dynamics biometrics as an authentication technique [1]. They built up a toolkit utilizing C++ for examination of data based on user keystroke patterns which is a computer desktop application. The toolkit was composed of adjusting the xview library routines, and serves as a front-end to their primary verification engine. The toolkit was useful in diagnosing framework conduct and can create graphical response for both Matlab and Gnuplot frameworks. They addressed the practical relevance of utilizing keystroke analysis as a biometric feature for verifying access to workstations. They surveyed the present conditions of keystroke analysis and present classification methods on basis of template matching and Bayesian probability models. Their designed toolkit was used to simulate the efficiency of using keystroke analysis as an authentication measure.

Another study utilized Keystroke analysis as a biometric for authentication to predict secured password for users to be used as login credentials to users [3]. Their methodology empowers the creation of long-term hardened password that

can be tried for login purposes or utilized for encryption of documents, entry access to a virtual private system [3]. Furthermore, their approach naturally adjusts to progressive changes in a client's keystroke timings while keeping up the same enhanced password over numerous logins, for use in document encryption or different applications requiring for user password. Their approach was mainly to generate password for clients based on their typing 36behavior to be used in other systems and the keystroke dynamic metric used was continuous keystroke technique [3].

In 2008, a study conducted by Brochoux and Clarke on deployment of keystroke analysis on smartphone [6]. In their study they utilized two factors as a technique for verification on smartphones, thus a secret-knowledge (password/PIN) and a keystroke technique. They suggested a simple PIN and a strong alphanumeric password as two streams of which the keystroke dynamics was applied to. Visual Basic .NET and Microsoft .NET Compact Framework 2.0 was used in deploying the keystroke classifier in order for it to run on various mobile operating systems [6]. Also, due to the poor performance of the neural network on smartphones, it was impractical to ascertain performance rates for false acceptance (the rate at which impostors are granted access to the system, FAR) and the false rejection (the rate at which legitimate users are rejected from their system, FRR). The performance for the statistical classifiers based upon entering a PIN and (longer) password was determined with respect to FAR and FRR. They actually compared PIN to password to find out the one that has the highest performance when augmented by keystrokes dynamics as authentication measure on smartphones [6].

In another study keystroke analysis was used as a tool for intrusion detection based on continuous authentication using periodic keystroke dynamics [4]. This is where authentication is performed through the interactive period a user uses the system. They argued that keystroke patterns are determined throughout an entire login session and if one can perform keystroke analysis of free text, there can be deployment of one of the efficient applications of continuous verification thus intrusion detection. They conclude by adapting periodic keystroke dynamics to further verify users who have already gain access to the system thereby raising alarms to the system administrators/operators if they are imposters or legitimate users [4].

Considering the capabilities of the reviewed systems, this study looks at the typing behavior of clients to supplement their login username and password to authenticate them on web applications, thus the verification actually is based on the keystroke patterns determined in the process of invoking the user's login credentials. The keystroke analysis technique adopted for this study is static keystroke dynamics.

2.1 How?

Conventional passwords are a simple path for verification in computer systems and applications yet in some cases their security level may be inadmissible [9]. This study specifically adapts static keystroke dynamics technique to support password security to verify users during login sessions in web applications. Keystroke dynamics is a procedure that overcomes large portions of the challenges different techniques, both customary and new, cannot address. Therefore, keystroke dynamics is a hopeful new answer for an already exiting question: "By what means would we verify clients on web applications?"

3. METHODOLOGY

An exploratory and descriptive design was used to find out the various ways users of the system manage their login credentials and other problems associated with the authentication module of the system. It is for this reason that accentuated that such a design is appropriate for providing a comprehensive understanding of the phenomenon being studied [7].

3.1 Study Population, Sample and Sampling Techniques

A purposive sampling technique was used to select University of Education–Kumasi (UEW-K) for the study because the university is more innovative in leveraging digital technology to drive its operational efficiency especially in managing its student's records and other administrative operations.

The study targeted a population of 50 staff members but 22 staff members were purposively selected in which they represent the various departments who directly work with the system. Their views were collected for the modeling and evaluation of the proposed system, where their profile data was collected to feed the system and then evaluated to prove the efficiency of the said system.

4. SYSTEM DESIGN

In designing the framework, an evolutionary methodology was employed for the fact that there was a likelihood of the framework requirements changing, was considered to be high. The prototype model was adapted because it permitted more flexibility and greater collaboration between the researcher and the case setting. Further, this model made it easy to discover mistakes associated with the system functionalities and the User Interface (UI) at the early stages of the system design. JavaScript and PHP were used for implementing the keystroke dynamics in the authentication module.

4.1 Use Case Diagram of the Authentication Module

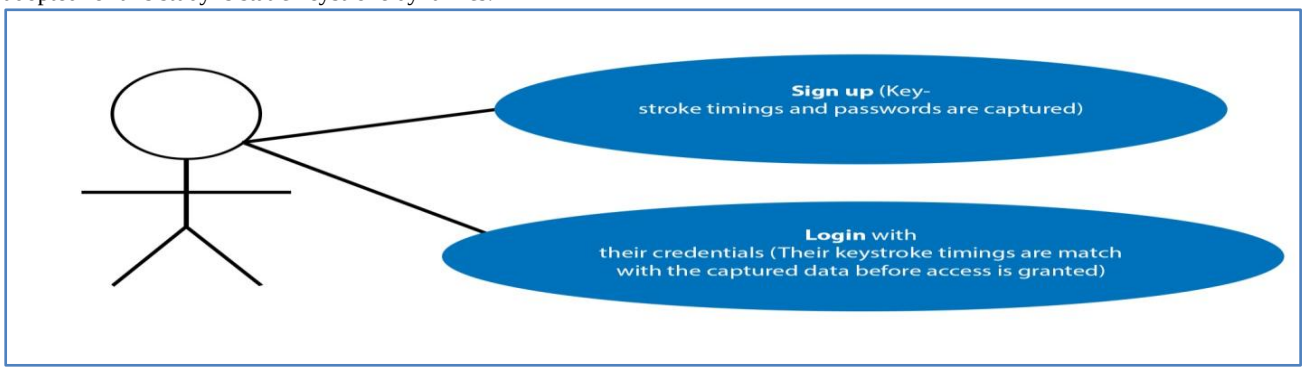

Figure 4.1: Use case diagram of the propose system

4.2 Data Flow Diagram

The figure below shows the dataflow of the proposed system

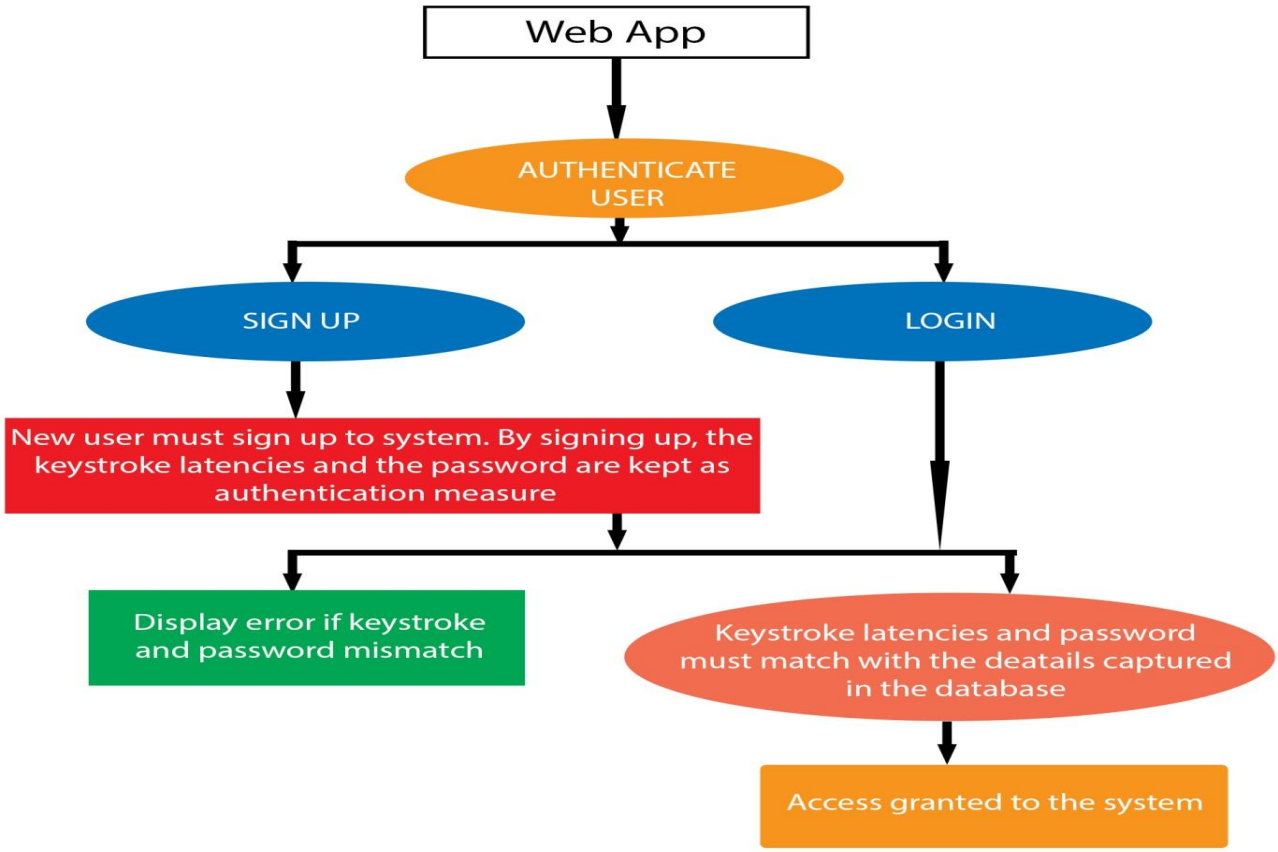

Figure 4.2: Dataflow diagram of the proposed system

4.3 Pages Design of the Proposed Authentication Module

The Authentication module consists of a sign up and login pages where the sign up page is linked with the login page. The sign up allows users of the system to create their accounts where their keystrokes latencies are determined and stored for the enhancement of the password security. Users' credentials thus password and username are entered five times during the accounts creation (signup) in which their keystrokes are captured. During the login, the user keystrokes latencies captured together with the password and username are matched for consistency before access is granted or denied. The pages are illustrated below;

Figure 4.3: Sign up page of the proposed authentication module

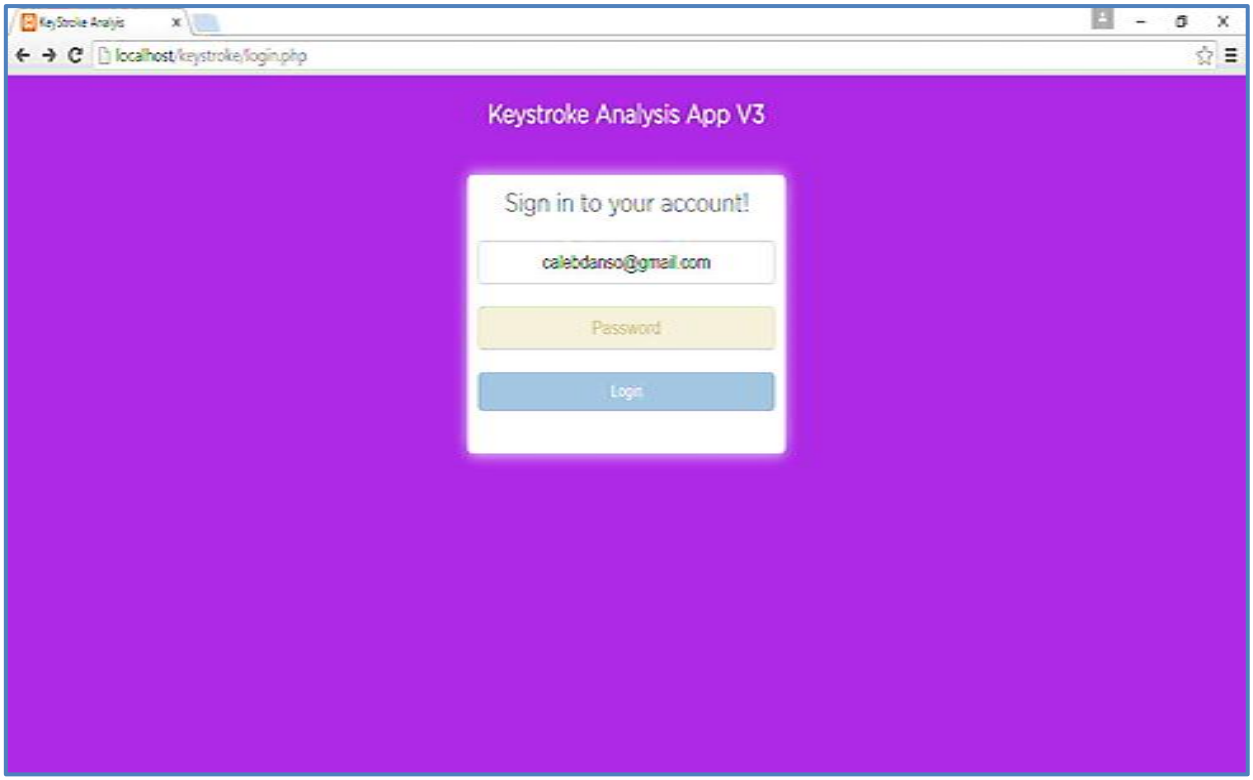

Figure 4.4 Login page of the proposed authentication system

5. SYSTEMS ANALYSIS AND EVALUATION

In order to use the keystroke dynamics to verify users, this study is based on some features, which were captured from keystroke events. These features which are "Latency", "Source" and "Terminus" can be acquired from processing the embedded JavaScript and PHP script. In description, *Latency* is the time interval between "Pressing time" and "Releasing time" of a same keystroke while *Source* is the period between "Releasing time" of one keystroke and "Pressing time" of the next keystroke. Also, *Terminus* is referred as the period between "Releasing time" of previous key and "Pressing time" of current key. All these features was put together to determine the time every character ways keyed to system by a user and used to draw the typing of every participant enrolled in the system. These features are captured in milliseconds and are illustrated in Figure 5.1 below.

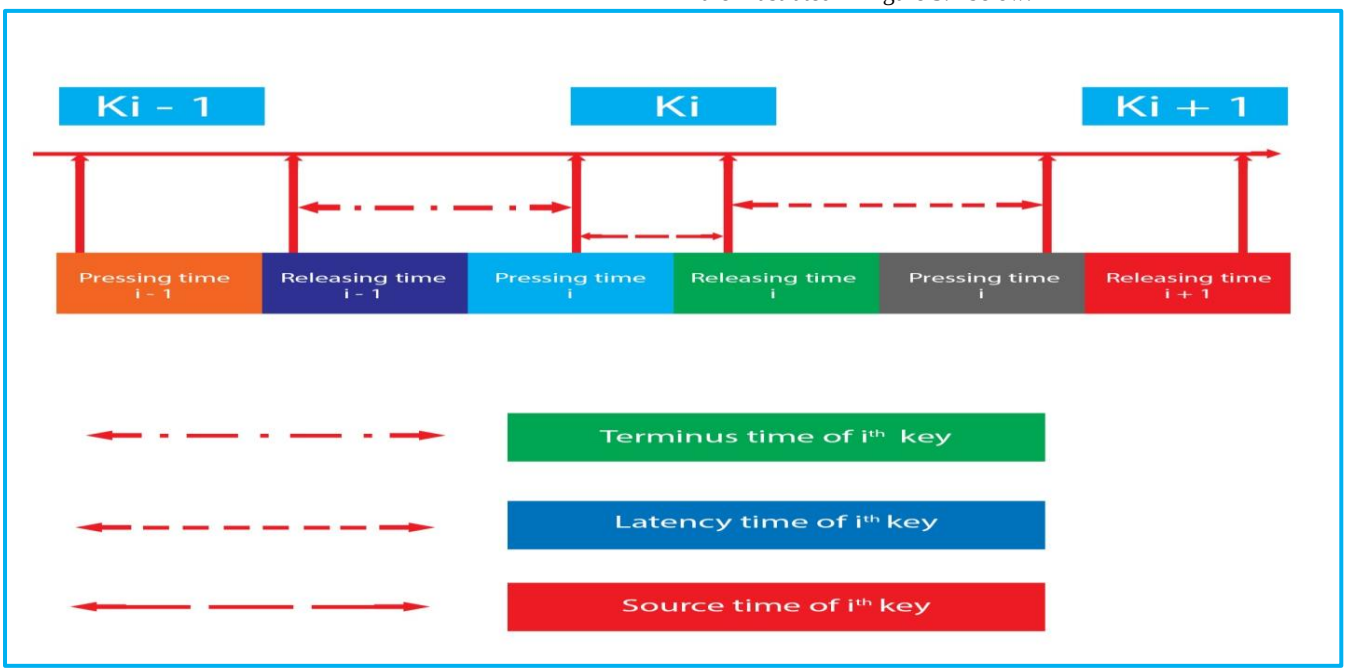

Figure 5.1: The proposed system data features

5.1 Data Processing of the proposed System

A level of irregularity exists in clients typing rhythms. While a few people might be exceptionally reliable, others are most certainly not. This causes issues amid verification of users because of large unwanted outliers. Keeping in mind the end goal to decrease these issues, the information is pre-processed before the required keystroke pattern is stored. Figure 5.2 below illustrates sample keystroke trait pattern in milliseconds captured from a user during sign up session where the upper-bound and lower-bound are determined using (username: mikiosei13@gmail.com, password: ember@25). The classifier/algorithm generated by JavaScript denies imposters of access based on their keystroke timings during authentication.

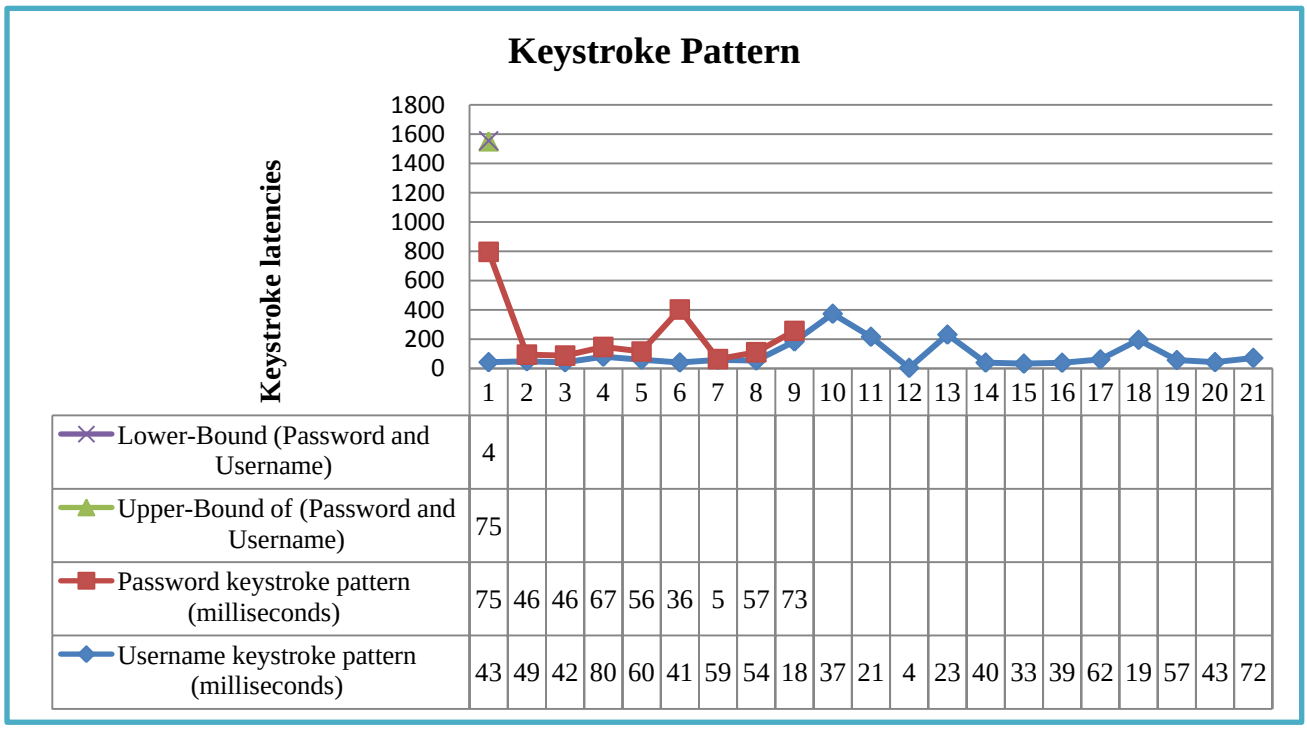

Figure 5.2: Processing of Keystroke timings of user's password and username

5.2 The FAR and FRR of the Classifier of the Proposed System

To assess the execution of our proposed approach in the part of grouping precision, two further approaches were utilized: false rejection rate (FRR) and false acceptance rate (FAR). These two approaches are defined below:

FRR shows the likelihood of recognizing/rejecting a legitimate client as an imposter.

FAR shows the likelihood of recognizing/accepting a fraud as a legitimate user.

Mathematical Equation of the FAR and FRR

$$FAR = \frac{x}{y}$$

X = Number of successful imposter attempts

Y = Total number of overall imposter attempts

$$FRR = \frac{m}{n}$$

M = Number of denied legitimate users

N = Total number of legitimate users

Figure 5.3 below indicates the outcome of average FAR and average FRR, where both were essentially figured utilizing each of the 44 participants' data which include legitimate (22) and imposter users (22). This figure demonstrates that the average FRR extends from 0% to 9% and FAR range from 0% to 4.5% which means that out of the 22 legitimate users; there is 9 percent possibility of denying a legitimate user of access and 4.5 percent possibility of granting illegitimate access to an imposter. These outcomes show that the possibility of the system accepting an imposter is lower than the possibility of denying legitimate user. This supports the study conducted by Gagbla that any authentication system having low FAR than FRR is of high performance rate.

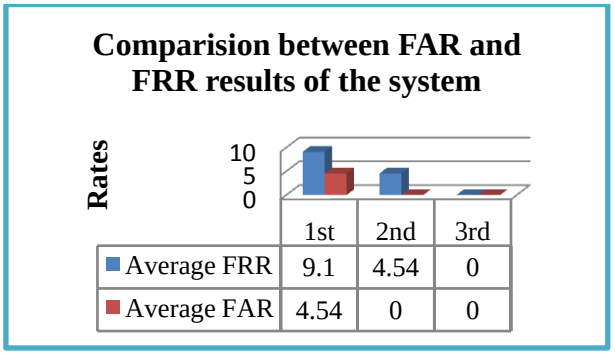

Figure 5.3: Results of average FAR and FRR

5.3 Scenario Evaluation of the Proposed System

The effectiveness of the system is tested and analyzed per desirable threshold (the user credentials and their keystroke timings) using statistical analysis.

The evaluation was organized under four circumstances of verification.

1. Legitimate client: the clients attempted to be verified in their own records captured.
2. Impostor client: the clients attempted to be verified in other client's records, knowing the strings (Username and Password) written by the legitimate users.
3. Observer impostor client: the clients watched the typing behaviour of legitimate users, to know how they invoke the keystrokes of their passwords and they attempted to authenticate to the system.
4. Guess imposter client: the user tries to speculate the password and the typing behavior of legitimate users.

5.3.1 Legitimate Users Verification

The system is designed to capture legitimate users' keystroke timings, process and use them to authenticate the users together with their password and username strings. Every legitimate user is enrolled to the authentication module 5 times in order to get the accurate typing patterns for evaluation. 22 legitimate users were successful enrolled to the system capturing their usernames and passwords as well recording their typing behaviour by their keystroke patterns. Upon authentication almost all the users successfully had their typing behaviour accurate and two enrolled clients had issues with their keystroke patterns. Those who did not get their typing behaviour accurate were not granted access to the system. This is illustrated in figure 5.4 where legitimate users were granted access five times after correctly specifying their Username and Password and correctly typing within their Lower bound and Upper bound keystrokes patterns as shown in figure 5.4. Also in figure 5.4, it can be realised that some legitimate users correctly specified their Username and Password but failed to type within the recorded Lower bound and Upper bound keystrokes patterns and for this reason were denied access. Therefore it could be recommended that users of the system should try their best to type within the lower bound and upper bound of their keystroke patterns that were captured during the sign up process.

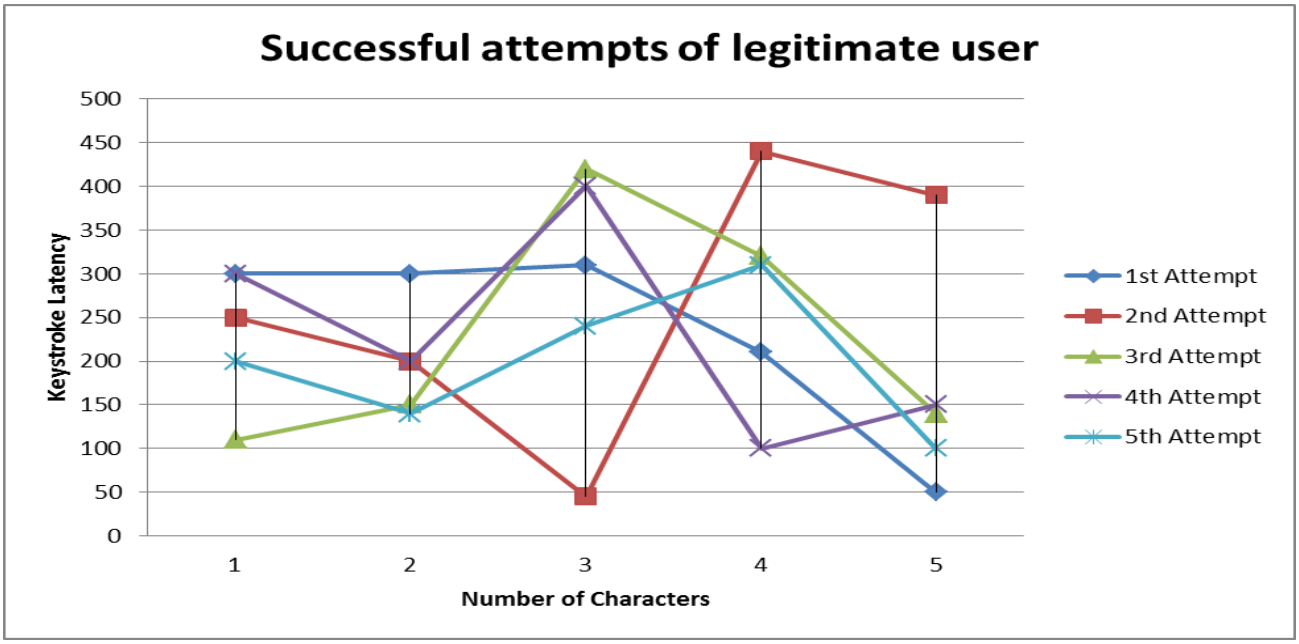

Figure 5.4: Successful Login attempts for legitimate user

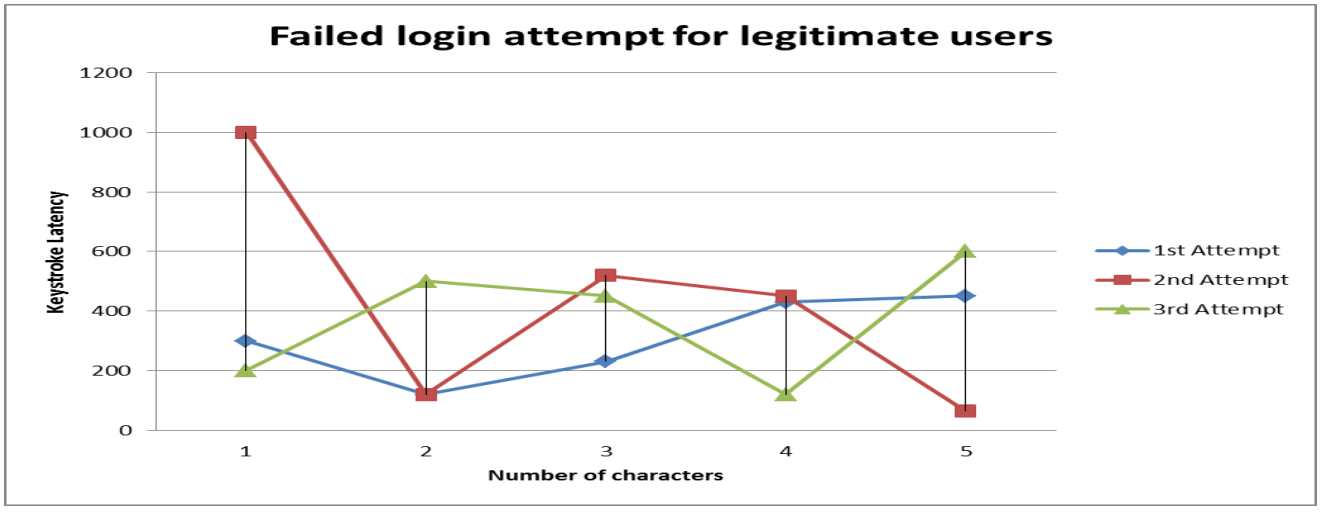

Figure 5.5: Failed login attempts for legitimate users

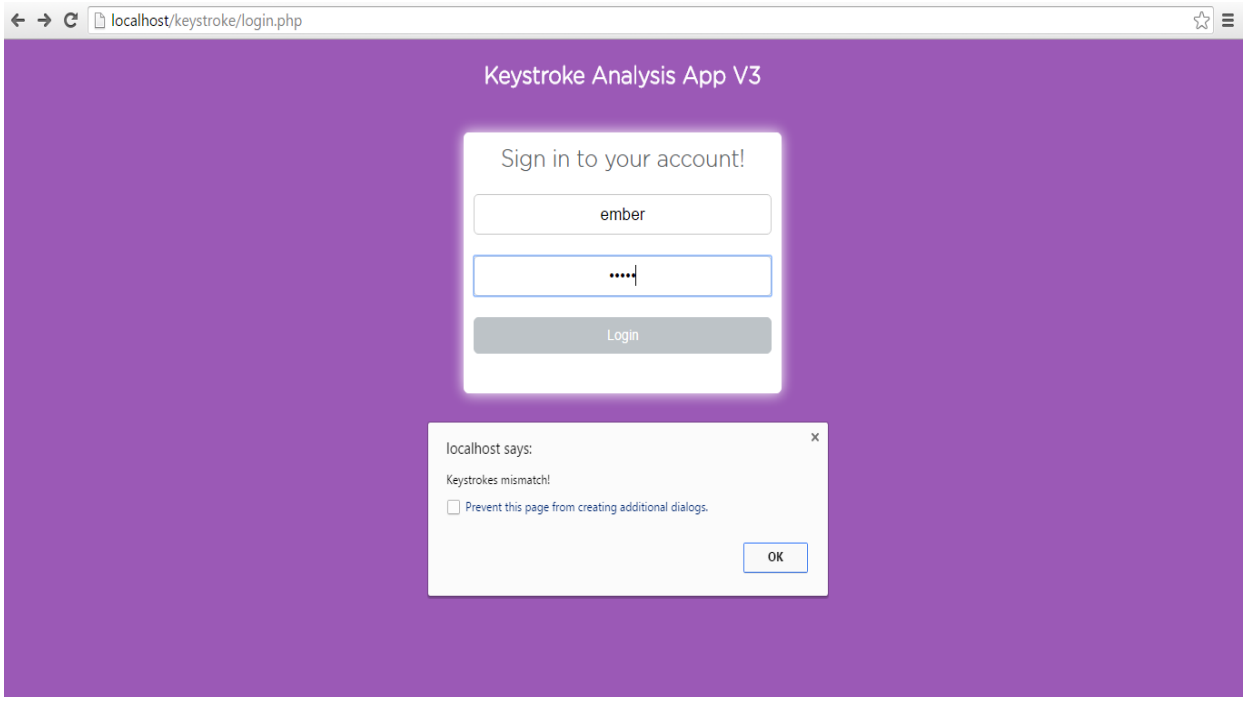

Figure 5. 5: Legitimate user verification

Figure 5.6 above depicts how a legitimate client is refused access to the system based on the typing pattern not matching the initial keystrokes captured to the database of the system no matter getting the credentials correct, it indicates "Keystrokes mismatch". The user is only granted access when both the user details in addition to keystroke patterns are the same.

5.3.2 Impostor and Legitimate User

Figure 5.7 and Figure 5.8 depict the distribution for keystroke latency of the target string "ember@25" and "mikiosei" as password and username respectively for a legitimate account user and an impostor.

An imposter can gain illegitimate access if his typing behaviour is similar to that of the legitimate user. It can be seen from figure 5.7 that there is a high correlation among the 3rd and the 6th keystrokes. Hence increasing the number of samples collected during enrolment will reduce the level of correlation thereby reducing imposter guessing chances. In line with this the system was designed to accept enrolment details five times before the upper bound and the lower bound is recorded as the benchmark.

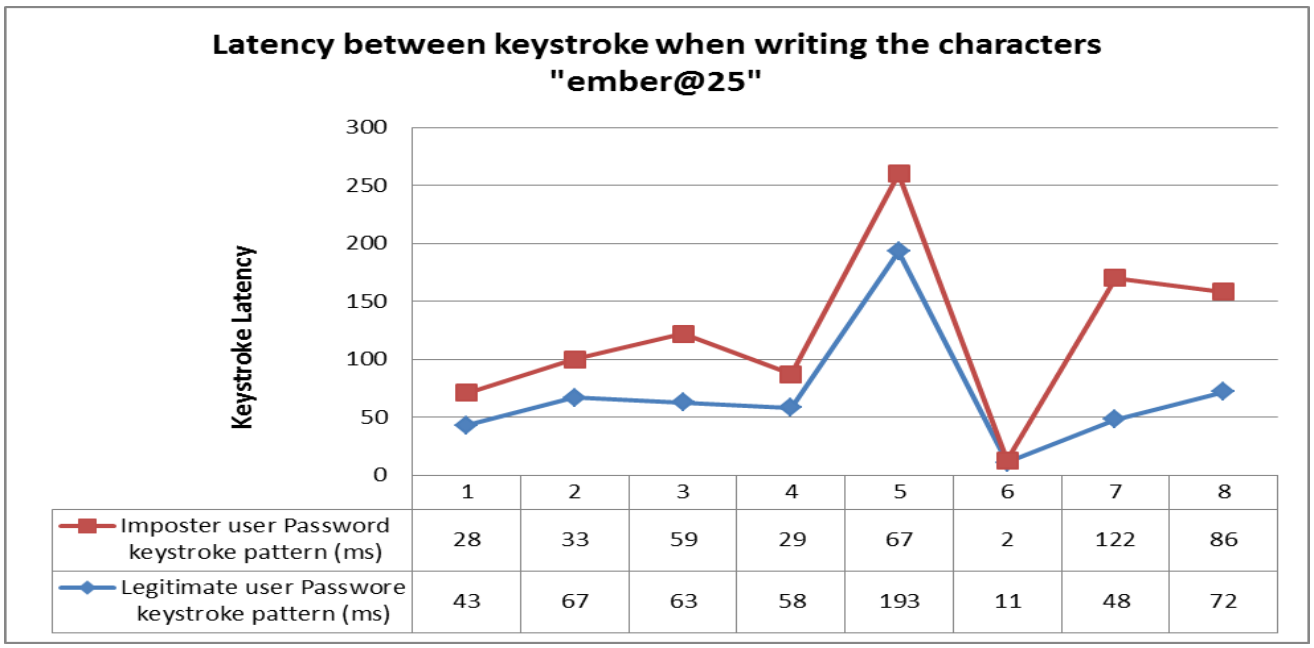

Figure 5.6: Comparison of latency between keystrokes for legitimate and impostor user

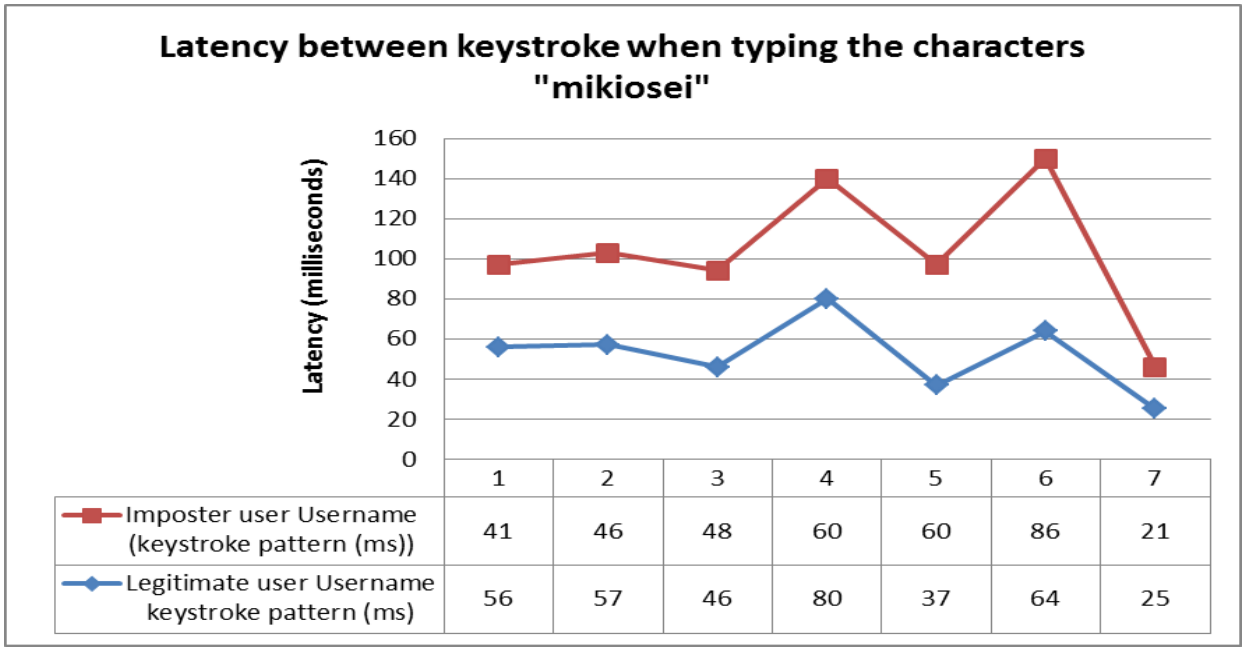

Figure 5.7: Keystrokes for legitimate and impostor

Figure 5.7 and Figure 5.8 above illustrates the keystrokes of a legitimate and an imposter which clearly shows the differences in their typing behavior. An imposter may try several times during the login session but may have to get the credentials and their corresponding keystrokes accurate before access could be granted. This shows that enhancing password security by using keystroke dynamics is important and helps to avoid password sharing compromise.

5.3.3 Guess Imposter User

The classifier is designed to only grant a login success only when both the user credentials and keystroke timings match with the captured data during enrollment. For this reason a guess imposter have to guess multiple of times in accessing the credentials and as well as the keystroke patterns of a particular legitimate user. After three wrong attempts the system is designed to refresh after 10 seconds and this adds a level of security to password protected systems. The figure 5.9 illustrates a guess imposter user session login failure.

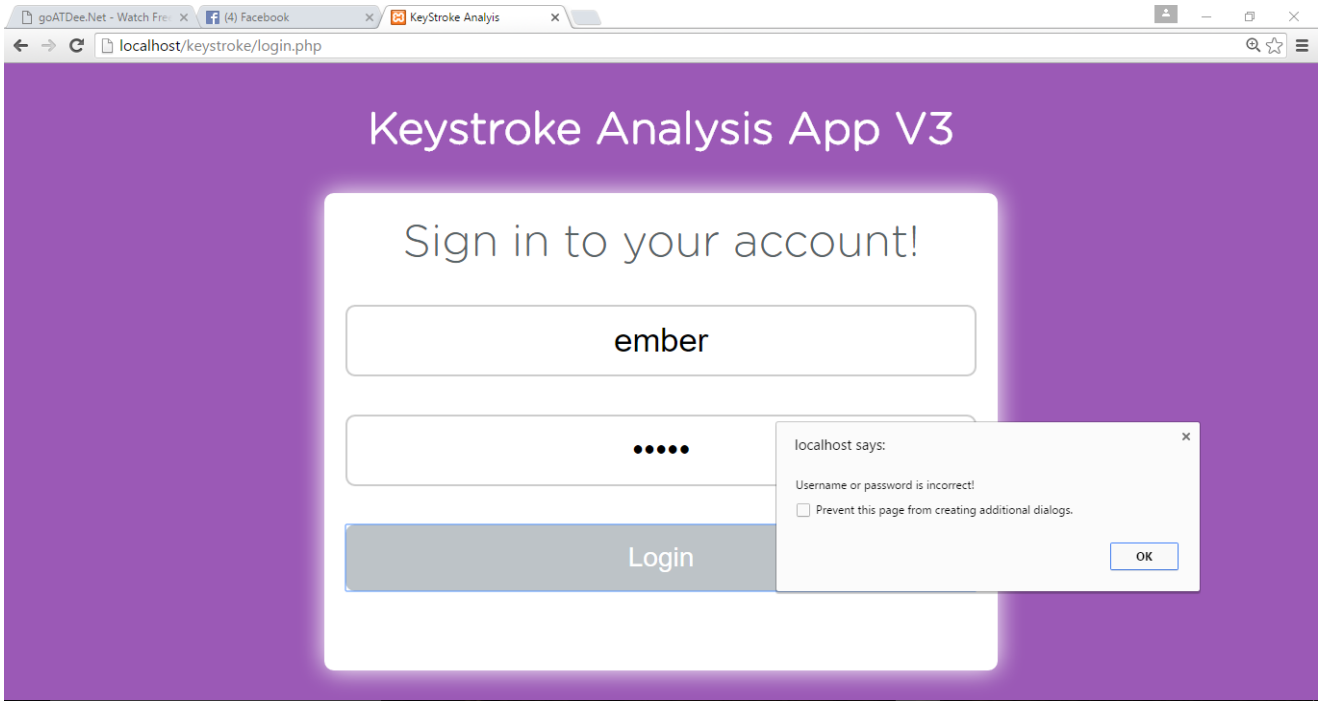

Figure 5.8: Guess imposter user

The figure 5.9 above illustrates how a guess imposter is rejected from gaining access to a legitimate user account. This authentication system executes its function without the requirement of any special device unlike other physical biometrics such as fingerprint.

5.3.4 Observer Imposter Client

The clients observe the typing behavior of legitimate users, to know how they invoke their keystrokes of their username and passwords and attempts to authenticate to the system. The system only grants users access when there is match with both characters typed and the keystroke patterns stored as upper bound and lower bound in the system. For this reason an imposter may know the typing behavior of a legitimate user but would be denied access to the system when he does not get the credentials (username and password) strings accurate.

5.3.5 Evaluating Character Timings

This evaluation was conducted to determine how an imposter is likely to guess the keystroke timings of legitimate user based on the combination of characters used to create the credentials and the choice of hands used in invoking the characters.

Figure 5.10 below demonstrates the latency of character sets for every classification. The entire latency range is split into six containers (<100, 100-150, 150-200, 200-250, 250-300, 300+) as appeared in the horizontal-axis. Inside every classification, every character pair is put into the comparing receptacle if its mean latency falls within the scope of the container. Character sets that involve two letter keys and one number key, keyed utilizing the same hand, take the longest time in invoking the keystrokes.

This is essentially on the grounds that two hands offer a specific measure of parallelism, while character sets typed with one hand require a specific level of consecutive movement of hand in typing the keys and subsequently tend to take longer time. This is very clear on account of two letters and one number sets, where keystrokes that are invoked utilizing one hand definitely take the longest time

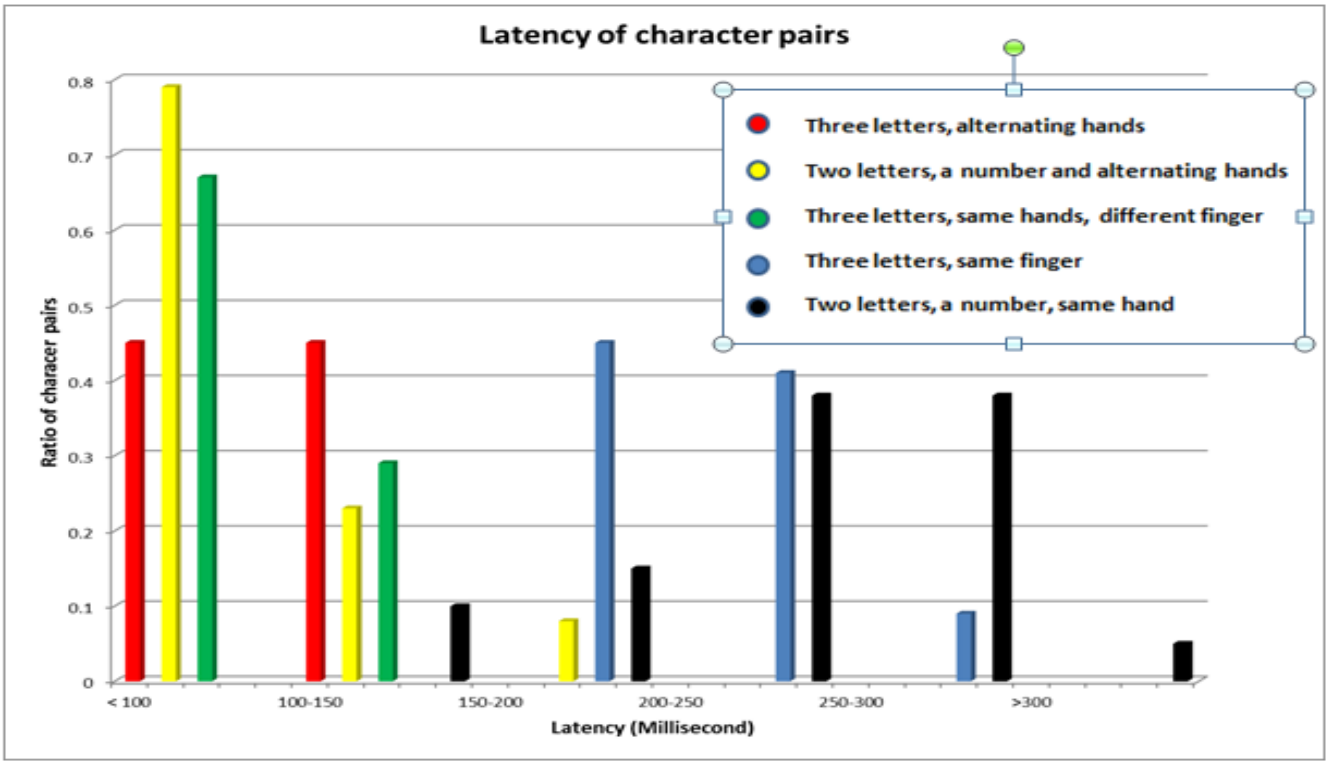

Figure 5. 9: Inter-Keystroke timings of character pairs

In this way, if the imposter studies a character pair keyed with latency more than 150 milliseconds, he can figure with high likelihood of accomplishment that the character pair is not invoked utilizing two alternating hands and consequently can assume around 1 bit of data about the substance of the character pair. This makes it less stressful for an imposter to guess the strings and typing behavior of the legitimate user accurate. Therefore, it is advisable for legitimate users of a system utilizing keystroke dynamics to be enrolled to system using their both hands in creating their keystroke profiles in determining their typing behavior. This further makes it very difficult for an imposter to guess the right strings and keystroke patterns of a legitimate user.

6. CONCLUSION

This research proposed a novel verification approach to computer web applications based on behavioral biometrics thus keystroke dynamics. The proposed technique derives the possibility of performing complicated biometrics without extra equipment, but only a keyboard device that happens to be part of every computer system. The study demonstrated that it can be utilized to handle the general client verification situation and give a moderately secure environment to be more secured against computer threats and attacks. The impact of the various biometric elements and template profiles of participants were tested. The framework was designed and tried with live information from user profiles captured during enrollment process.

Test results were demonstrated taking into account records gathered from individuals of various typing behavior by enrolling them into the system capturing their username and password five times in order to obtain their lower bound and upper bound keystrokes patterns as when they invoke their keystrokes. This study concludes that keystroke dynamics can be used as add up security measure for securing web applications. Although there is no unique trend in users' keystroke patterns, access is granted to the legitimate user once the keystroke values are within the range of the lower bound and upper bound keystroke timings.

6.1 How can this be useful?

From the research, the design and implementation of the framework the following suggestions are recommended:

Online examinations represent a special issue for Distance-learning education, in that it can be extremely hard to give genuine client verification. Because of the inborn obscurity of being on the web, contrasted with taking an examination in a classroom domain, students may endeavor to falsely support their scores in online examinations by having another individual take the exam for them, which the traditional password/username verification scheme can't recognize. This study suggests that a continuous keystrokes dynamics could be implemented to trace such activity on the web. The universities undertaking distance education online examination can adopt this strategy to ensure check and balances of student's performance.

1. Utilizing keystroke analysis as a part of web-based verification systems brings out results that is quick, exact, and adaptable to a huge number of clients, requires no adjustment in client conduct (the behavior) and is promptly deployable over the Internet without the requirement for costly tokens, cards or other specific expensive equipment. A user's keystroke pattern is difficult to be shared, compromise, forged and forgotten. For the utilization of keystroke analysis to enhance password authentication to screen and verify clients, various institutions can easily and cost viably deploy secure access, also observe administrative policies, and avoid vulnerabilities or fraud activities.
2. This study was able to cater for most outliers such as backspace and delete key times that may be recorded as part of the keystroke patterns of an individual but was not able to include that of the navigational keys of the computer keyboard. Therefore this study suggests that further studies should consider eliminating outliers like the keystroke timings of navigational keys on the computer keyboard.

7. REFERENCES

[1] Monrose, R., & Rubin, A., (1999). "Keystroke Dynamics as a Biometric for Authentication". Future Generation Computer Systems, 16(4) pp. 351- 359. .

[2] Monrose, F., Rubin, A., (1997)."Authentication via Keystroke Dynamics", Proceedings of the 4th ACM Conference on Computer and Communications Security, pp. 48-56.

[3] Monrose, F., Reiter, M. K and Wetzel S. (2001). Password hardening based on keystroke dynamics. Bell Labs, Lucent Technologies, Murray Hill, N.J., US Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] Gunetti, D. and Picardi, C. (2012). Keystroke Analysis as a Tool for Intrusion Detection. "Continuous Authentication Using Biometrics: Data, Model, and Metrics". Issa Traore: IGI Global.

[5] Li, J. (2003). Keystroke Analysis: A Novel Type of Authentication. Accessed online through: http://individual.utoronto.ca/jamyli/writing/keystroke.

[6] Brochoux, A. and Clarke, N. L. (2008). Deployment of Keystroke Analysis on a Smartphone. Australian Information Security Management Conference, Edith Cowan University: Perth, Western Australia. Source: http://ro.ecu.edu.au/ism/48.

[7] Bryman, A. and Bell, E. (2007): "Business research methods", 2nd ed. Oxford: Oxford University Press.

[8] Gagbla, K. G. (2005). Securing E-Business Applications. Using Keystroke Dynamics as a Biometric Authentication Technique.

[9] Attila M., Zoltán B., and László C. (2007). Strengthening Passwords by Keystroke Dynamics IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. Dortmund, Germany