

Keystroke Statistical Learning Model for Web Authentication

Cheng-Huang Jiang
capsuleone@gmail.com

Shiuhpyng Shieh
ssp@csie.nctu.edu.tw

Jen-Chien Liu
ljchien.cs94g@nctu.edu.tw

National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan 300

ABSTRACT

Keystroke typing characteristics is considered as one of the important biometric features that can be used to protect users against malicious attacks. In this paper we propose a statistical model for web authentication with keystroke typing characteristics based on Hidden Markov Model and Gaussian Modeling from Statistical Learning Theory. Our proposed model can substantially enhance the accuracy of the identity authentication by analyzing keystroke timing information of the username and password. Results of the experiments showed that our scheme achieved by far the best error rate of 2.54 %.

Keywords

Keystroke, Statistical Learning Theory, Hidden Markov Model, web authentication, Gaussian Model

1. INTRODUCTION

As the Internet becomes powerful and convenient, more and more applications are developed for web-based services instead of for local use only. Web-based services allow people to access information and resources globally and ubiquitously. Unfortunately, it also creates opportunities for malicious attacks and intrusion. As a consequence, the authentication of user identity for web-based services has become an important issue. Conventionally, web-based services employ username-password pair to authenticate the identity of a user. In the authentication phase, an adversary with a stolen username-password pair may gain access to the web-based service by masquerading as the victim. To deal with these problems of impersonation and illegal access, a biometric verification mechanism is needed to complement, but not to replace conventional authentication schemes.

Behavioral biometrics, which requires a user to behave in a consistent manner, includes keystroke dynamics, speech recognition, hand-writing, and mouse movement. Keystroke dynamics, also referred to as keyboard typing characteristics or keyboard typing rhythms, is one of the most novel and creative biometric techniques, and it has the following advantages over

This work is supported in part by the National Science Council (NSC), the Institute for Information Industry (III), the Industrial Technology Research Institute (ITRI), the International Collaboration for Advancing Security Technology (iCAST), and the Taiwan Information Security Center at NCTU (TWISC@NCTU).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASIACCS'07, March 20–22, 2007, Singapore.

Copyright 2007 ACM 1-59593-574-6/07/0003...\$5.00.

others: non-intrusive, transparent, and low-cost.

Keystroke dynamics is based on the assumption that different people have unique habitual rhythm patterns in the way they type. Previous work [4][6] has demonstrated that it is a feasible authentication measure.

Research regarding fixed-text keystroke analysis, which evaluates the typing patterns from short, fixed text or predetermined text, can be divided into two categories. One category focuses on analyzing the keystroke timing information of username-password pair typed by a user. Ru and Eloff [1] used fuzzy logic to characterize the typing behavior of a user based on the keystroke latencies, distance of keys on a keyboard, and typing difficulty of the key combinations. Incorporating the concept of keyboard gridding [2], Magalhaes et al [3] devised a lightweight algorithm that only evaluated one target string, the password. Unfortunately, keyboard gridding is optimized for right-handed users. Haidar et al [5] presented a suite of techniques using neural networks, fuzzy logic, and statistical methods to learn the typing behavior of a user.

The other category utilizes keystroke timing information of predetermined texts typed by the user during a certain period of time. Gaines et al [9] conducted an experiment in which seven professional secretaries were asked to type three passages twice, four months apart. However, they could not create reference profiles because of insufficient sample data. Bergadano et al [10] suggested an approach which measures digraph latencies based on the degree of disorder.

These methods cannot provide high accuracy to differentiate between a legal user and an impersonator. To cope with this problem, a new scheme for user authentication with high confidence is desirable. In this paper, we present a formal statistical model for keystroke dynamics analysis using Hidden Markov Model and Gaussian Modeling. Based on the proposed model, we develop schemes for fixed-text keystroke analysis, which can be applied to web-based services to enhance the security strength of connectional authentication mechanisms.

2. MODELING AND METHODOLOGY

In our model, a single keystroke will trigger two events, the key press event and the key release event. We define the *duration* in our scheme as the elapsed time between the first and the last consecutive pressed keys, that is $P_l - P_f$, where P_f (or P_l) represent the time the first (or the last) key was pressed. Let Q denote the set of keys of interest. The graph for modeling N consecutive keystrokes is referred to as an n -graph in the literatures, that is, the graph for two consecutive keystrokes is referred to as a digraph and that for three consecutive keystrokes as trigraph, etc. Given a sequence of consecutive keystrokes, $S = \{s_1, s_2, \dots, s_m\}$, where m

is the number of keystrokes in the sequence, the number of n -graphs is $m-n+1$, and the set of n -graphs is denoted as $G = \{g_1, g_2, \dots, g_{m-n+1}\}$. The set of durations of n -graphs is defined as $GD = \{d(g_1), d(g_2), \dots, d(g_{m-n+1})\}$ where $d(g_k) = P_{n+k-1} - P_k$. Our model analyzes the durations of n -graphs as timing features.

2.1 Gaussian Modeling and Maximum Likelihood Estimation

Previous work [6] [9] showed that the duration distribution of a given set of digraphs forms an approximate Gaussian distribution. Therefore it is natural to make the assumption that n -graph $g \in \mathcal{Q}^n$, with duration $d(g)$, forms a Gaussian distribution, such that

$$\Pr[d(g) | g] = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(d(g)-\mu_g)^2}{2\sigma_g^2}},$$

where μ_g is the mean value of the duration $d(g)$ for n -graph g , and σ_g is the standard deviation.

It may not be possible to collect all typing keystrokes of the individual and to calculate the mean and variance for each distinct combination of n -graph durations. However, it is desirable to deduce $\{(\hat{\mu}_g, \hat{\sigma}_g)\}_{g \in \mathcal{Q}^n}$ of n -graph durations and give a keystroke sequence S by the method of maximum likelihood estimation of the parameters. Fortunately, the maximum likelihood estimation of the parameters for Gaussian distribution can compute the sample mean and sample variance as follows.

$$\hat{\mu}_g = \frac{\sum_{i=1}^k d(g)}{k}, \hat{\sigma}_g^2 = \frac{\sum_{i=1}^k [d(g) - \hat{\mu}_g]^2}{k-1}$$

where k is the number of n -graph g appeared in S .

2.2 Hidden Markov Model

Hidden Markov Models (HMMs) [7][8] can be used to model sequential data, such as the sequence of keystroke timing information we intend to analyze in this paper. The HMM used to model the timing information of keystroke sequence is shown in Figure 1. It is a statistical graphical model, where each circle is a random variable. Unshaded circles q_i represent unknown (hidden) state variables we wish to infer, and shaded circles y_i are observed state variables, where t is a specific point in time. A is a state transition matrix holding the probabilities of transitioning from q_i^i to q_{i+1}^j , where q^i (or q^j) means the i -th (or j -th) state. So we have $P(q_{i+1}^j | q_i^i) = A_{ij}$. η is a state emission matrix holding the output probability $P(y_i | q_i^i)$ of i -th state. π_i is the initial state probability of i -th state. A compact notation $\lambda = (A, \eta, \pi)$ is used to indicate the complete parameter set of the model.

In our setting, given a keystroke sequence S , the set of n -graph G , the set of $[n+1]$ -graph G' . The state transition matrix A is the probability of the frequency that the $[n+1]$ -graph appeared in the S as follows.

$$A_{g_t, g_{t+1}} = \frac{|g'_t|}{m-n},$$

where $|g'_t|$ denotes the number of appearances of $[n+1]$ -graph g'_t in G' .

The state emission matrix η here is defined as the Gaussian distribution probability of the n -graph G with duration GD as follows:

$$\eta_g(d(g_i)) = \begin{cases} \Pr[d(g_i) | g] = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{[d(g_i)-\mu_g]^2}{2\sigma_g^2}}, & g = g_i \\ 0, & g \neq g_i \end{cases}$$

The initial probability vector π is the probability of the frequency that the n -graph appeared in S . We make the assumption that each individual has his/her own HMM with $\lambda = (A, \eta, \pi)$ for individual's keystroke timing characteristics.

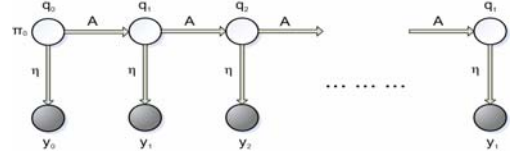


Figure 1: The Hidden Markov Model for keystroke analysis.

Given a keystroke sequence S and its timing information, we have to choose one from the number of HMMs which has the highest probability to generate the keystroke sequence S . Consequently, we have to calculate the probability of keystroke sequence S for each HMM. We will show how to solve the problem with Forward Algorithm in the next section.

2.3 Forward Algorithm

The problem of finding the probability of keystroke sequence S can be viewed as how well a given HMM $\lambda = (A, \eta, \pi)$ would score on S . We use the Forward Algorithm [8] to calculate the probability of an m long keystroke sequence S with n -graph G , and n -graph durations GD .

The state probabilities α 's of each state can be computed by first calculating α for all states at $t = 1$.

$$\alpha_1(g_1) = \pi(g_1) \cdot \eta_{g_1}(d(g_1))$$

Then for each time step $t = 2, \dots, k$, the state probability α is calculated recursively for each state.

$$\alpha_{t+1}(g_{t+1}) = \alpha_t(g_t) \cdot A_{g_t, g_{t+1}} \cdot \eta_{g_{t+1}}(d(g_{t+1}))$$

Finally, the probability of keystroke sequence S given a HMM $\lambda = (A, \eta, \pi)$ is as follows:

$$\Pr[S, G, GD | \lambda] = \alpha_k(g_k) = \alpha_{k-1}(g_{k-1}) \cdot A_{g_{k-1}, g_k} \cdot \eta_{g_k}(d(g_k)).$$

2.4 General Modules for Keystroke Analysis

In this section, we devise two modules to authenticate the user: *Profile Building Module* and *Authentication Module* underlying the model and algorithm described in the previous sections.

In the Profile Building Module, we first build the reference profile for each user, which requires the user to provide reference samples. After collecting a sufficient number of reference samples, we use the maximum likelihood estimation for Gaussian Modeling to calculate the parameters of each n -graph's duration. We also must

compute the transition probability matrix and initial probability vector with respect to HMM. The parameters calculated for HMM are treated as the basic element of the reference profile for each user.

In the Authentication Module, given a keystroke sequence S of target string from a user with claimed identity ID , we wish to examine the possibility that ID generated S . First, we transform the keystroke sequence S to n -graph combinations G and calculate the timing information of n -graph duration GD as usual. Now we produce a vector GDT , such that

$$GDT = \{\mu_{g_1} - \varepsilon\sigma_{g_1}, \mu_{g_2} - \varepsilon\sigma_{g_2}, \dots, \mu_{g_{m-1}} - \varepsilon\sigma_{g_{m-1}}\},$$

where ε is the weighting factor, μ_{g_k} and σ_{g_k} is ID 's duration mean and standard deviation of n -graph g_k respectively. GDT is the n -graph duration vector to evaluate the threshold value of the probability produced by modified Forward Algorithm. With the inputs GD , GDT , and λ_{ID} , we can apply modified version of Forward Algorithm to obtain two probability value $\Pr[S, G, GD | \lambda_{ID}]$ and $\Pr[S, G, GDT | \lambda_{ID}] \cdot \Pr[S, G, GD | \lambda_{ID}]$ can be viewed as the possibility that all the n -graphs durations in G deviate ε times of duration σ from duration μ . $\Pr[S, G, GDT | \lambda_{ID}]$ is the threshold value of probability used to decide that the acceptance of the keystroke sequence S is confirmed if following expression is true.

$$\Pr[S, G, GD | \lambda_{ID}] \geq \Pr[S, G, GDT | \lambda_{ID}]$$

The weighting factor ε can be specified with respect to different level of security strength.

3. EXPERIMENTS AND RESULTS

We conducted the experiment via web browser using a client-side JavaScript to gather the timing information of keystrokes. In our experiment, we utilized a timing accuracy of one millisecond and the digraph as the segment size of keystroke sequence. To generate reference samples, 58 volunteers supplied 20 samples of two familiar strings (username and password). These volunteers provided 15 attempts each to authenticate their own accounts, thus, 870 test samples were used to evaluate false reject rate. Another 257 anonymous volunteers tried authenticate the accounts of legitimate users. Each account was attacked between 44 and 82 times for a total of 3528 imposter test samples.

We evaluate the value of standard deviation weighting factor ε between 0.2 and 3.5 with interval of 0.1. The equal error rate (ERR) with the minimum target length of 9 and the reference sample size of 20 could be lower to 2.54% (Figure 2). Since it is difficult for a user to remember long username and password, the EER herein close to 2% with a minimum target length of 9 is the best result so far in literature.

4. CONCLUSION

Conventional password authentication mechanism is insufficient for providing strong security and reliability for identity verification of web-based applications. Our proposed scheme can offer users with a strong defense against adversaries who have access to stolen passwords. In our experiments, we restricted the length of username and password to a minimum of 9 characters. Our approach achieved an EER of 2.54%, which is by far the lowest for this type of system. The experimental results

demonstrated that our scheme is feasible and of practical use as a complement, but not replacement, for conventional authentication schemes. For future work, we may combine the proposed scheme with the analysis of the surfing route to the login page. The proposed model can be also extended to devise a scheme for free-text keystroke analysis, such as continuous real-time identity verification.

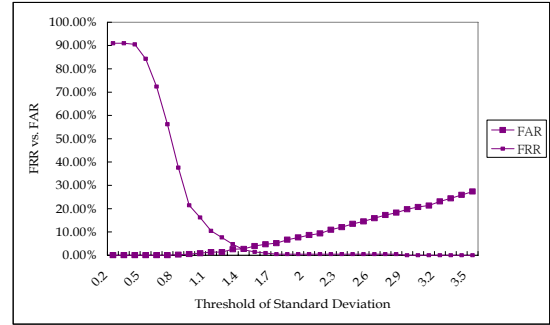


Figure 2: Minimum target string length = 9, reference sample size = 20, EER = 2.54% (FRR: false reject rate, FAR: false accept rate)

5. REFERENCES

- [1] W. G. de Ru and J. H. P. Eloff, "Enhanced Password Authentication through fuzzy logic," IEEE Expert, vol. 17, no. 6, pp. 38–45, Nov. 1997.
- [2] K. Revett and A. Khan, "Enhancing Login Security Using Keystroke hardening and Keyboard Gridding", Proceedings of the IADIS MCCSIS, 2005.
- [3] S. T. Magalhaes, H. M. D. Santos, "An Improved Statistical Keystroke Dynamics Algorithm", Proceedings of the IADIS MCCSIS, 2005.
- [4] A. Peacock, X. Ke, and M. Wilkerson, "Typing Patterns: A Key to User Identification", IEEE Security & Privacy, vol. 2, no. 5, pp. 40-47, Sep 2004.
- [5] S. Haidar, A. Abbas, and A. K. Zaidi, "A multi-technique approach for user identification through keystroke dynamics," in Proc. IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 1336–1341, 2000.
- [6] F. Monroe and A. Rubin, "Authentication via Keystroke Dynamics", Proceedings of the 4th ACM conference on Computer and Communication Security, pp. 48-56, Apr. 1997.
- [7] S. Russell and P. Norvig, "Artificial Intelligence, A Modern Approach", Prentice Hall, 1995.
- [8] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, No. 2, Feb. 1989.
- [9] R. S. Gaines, W. Lisowski, S.J. Press, and N. Shapiro, "Authentication by Keystroke Timing: Some Preliminary Results", Rand Report R-256-NSF. Rand Corporation, 1980.
- [10] F. Bergadano, D. Gunetti, and C. Picardi, "User Authentication through Keystroke Dynamics", ACM Transactions on Information and System Security (TISSEC), vol. 5, no. 4, pp. 367-397, Nov 2002.