NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IDENTITY OBFUSCATION THROUGH THE EXCHANGE OF KEYSTROKE BIOMETRIC DATA

by

Peter T. Goodwin

March 2020

Thesis Advisor: Vinnie Monaco
Second Reader: John D. Fulp

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE IDENTITY OBFUSCATION THROUGH THE EXCHANGE OF KEYSTROKE BIOMETRIC DATA	**5. FUNDING NUMBERS**
6. AUTHOR(S) Peter T. Goodwin	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The study of keystroke dynamics began decades ago, but the field has become increasingly relevant due to current events that have increased public awareness of the erosion of internet privacy. Web servers of 2020 are highly sophisticated at observing and analyzing the behavior of users on their platforms. Current capabilities include detection of fraud and concealed information, predicting demographic traits, and identifying users, all based on idiosyncratic differences in the way users interact with a keyboard. This reality forces the question of whether there is anything that users can do to prevent their own machines from divulging personal information via their keystroke dynamics-driven profile.

This research describes an obfuscation technique capable of hiding a user's keystroke dynamics-based fingerprint through a combination of exchanging and randomizing biometric data generated while typing. Additionally, this obfuscation was designed to be as subtle as possible, so that its use would be minimally detectable. The results of this research indicate that there is an inherent tradeoff between the ability to conceal a user's identity and the ability to conceal the presence of an identity-hiding tool.

14. SUBJECT TERMS cyber security, privacy, keystroke biometrics, human computer interaction	**15. NUMBER OF PAGES** 49
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IDENTITY OBFUSCATION THROUGH THE EXCHANGE OF KEYSTROKE BIOMETRIC DATA

Peter T. Goodwin
Lieutenant Junior Grade, United States Navy
BS, U.S. Naval Academy, 2016

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2020

Approved by: Vinnie Monaco
Advisor

John D. Fulp
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The study of keystroke dynamics began decades ago, but the field has become increasingly relevant due to current events that have increased public awareness of the erosion of internet privacy. Web servers of 2020 are highly sophisticated at observing and analyzing the behavior of users on their platforms. Current capabilities include detection of fraud and concealed information, predicting demographic traits, and identifying users, all based on idiosyncratic differences in the way users interact with a keyboard. This reality forces the question of whether there is anything that users can do to prevent their own machines from divulging personal information via their keystroke dynamics-driven profile.

This research describes an obfuscation technique capable of hiding a user's keystroke dynamics-based fingerprint through a combination of exchanging and randomizing biometric data generated while typing. Additionally, this obfuscation was designed to be as subtle as possible, so that its use would be minimally detectable. The results of this research indicate that there is an inherent tradeoff between the ability to conceal a user's identity and the ability to conceal the presence of an identity-hiding tool.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge my wife, Kaitlin Marie, whose unrelenting support throughout this program has made this the most joyful academic experience of my life. I would also like to thank my advisor and second reader, Dr. John Monaco and Professor John D. Fulp, respectively, who were incredibly helpful and supportive. Finally, I would like to thank my God, who gave me this profound gift of being able to continue my education.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In the year 2020, there are few things as prominent in the public consciousness as internet privacy. Pundits, politicians, and consumers of all stripes make it routinely clear that this is among their foremost concerns, and for good reason. Scandals ranging from the use of Cambridge Analytica in the 2016 election cycle, to the Amazon Alexa listening a little too closely to its owners, have put a long overdue topic of concern in the public spotlight. While policy-based remedies to this issue are discussed in the public arena, everyday people have begun to try and protect their privacy themselves. To illustrate this, look no further than the increased adoption of VPNs, or virtual private networks. In just a few short years, a practice that was largely relegated to sophisticated internet users is projected to be a $70 billion industry by 2025 [1]. In less than two decades, internet users are already shifting from being willing to share seemingly everything about their lives on social media, to demanding ways to keep their personal information from being collected via the Internet.

Computer scientists have long been interested in the subject of masking an identity through a process called obfuscation. Obfuscation can be defined as obscuring the true meaning of something. In computer science, this concept is often thought of in tandem with software development, during which code is written in a manner that makes it difficult to understand, but it can be used in a variety of different ways in both offensive and defensive settings.

In an era where privacy is at a premium, many people would be surprised to know some of the ways in which their identity can be revealed. Among some of these lesser known methods is tracking an individual using their keystroke biometric data. This data is generated while humans type and can be readily used to distinguish an otherwise anonymous group of individuals from one another. Hiding this behavior presents unique challenges, because it is as much a question of how to make a user’s efforts at obfuscation undetectable, as it is about hiding their identity.

This thesis focuses on combining two different forms of obfuscation to more completely hide a user's identity. These forms are camouflage and impersonation. While these terms are often thought of interchangeably, they have important distinctions that ultimately determine their utility in a given setting.

Camouflage is focused on being able to hide by blending in with the current environment, while the goal of impersonation is to take on the appearance of a different entity. As is well known, in the natural world animals ranging from insects to mammals have evolved to protect themselves using both camouflage and impersonation. Whether it is the changing color of a chameleon's skin, or a stick bug being mistaken for a twig, there are plenty of examples of nature using a wide array of methods to hide or confuse. In a similar fashion, this project both investigated and implemented camouflage and impersonation techniques to obfuscate the behavior exhibited by humans while they type.

While some keystroke obfuscation tools have already been proposed and implemented, it is unclear whether they only trade the obfuscation of one digital fingerprint for the formation of another. For example, one commonly used technique is to trade the timing of a user's keystrokes for randomly timed intervals. In this case, the behavior no longer resembles that of the specific user, but from the server's point of view keystroke timing determined by a random number generator is incredibly unique, and therefore identifiable. Because of this, in order to more thoroughly obfuscate a given user's keystroke dynamics one must not only hide their particular behavior, but also ensure that it blends in with the rest of the network traffic.

Bearing all this in mind, this research attempted the development of a new technique that is able to camouflage and impersonate, can be used in real time, and is capable of testing the hypothesis that a user's keystroke dynamics-based identifier, can be obfuscated by exchanging his or her keystroke biometric data with that of another user.

One of the principal focuses of this study is obfuscation, to include the domain of keystroke dynamics obfuscation and the obfuscation of any identifiable behavior generated by middleware implemented to subtly hide user identities.

As stated in the hypothesis, this project is attempting to assess the feasibility of obfuscation of a user's keystroke generated footprint by exchanging his or her behavioral data with that of another user. This will be attempted strictly through the use of a non-parametric method, described in greater detail in Chapter III.

This thesis is broken down into three principal sections. It begins with a basic explanation of keystroke dynamics and the current state of attempts at obfuscation within this field. Next the development and implementation process of this thesis' obfuscation method is described. Finally, the data and results are displayed, analyzed, and used to draw conclusions as well as identify opportunities for future implementation.

THIS PAGE INTENTIONALLY LEFT BLANK

II. KEYSTROKE DYNAMICS OVERVIEW: CURRENT STATE OF RESEARCH

Like so many different types of human behavior such as gait or vocal patterns, the way in which people type can be a used to identify them. Using an individual's typing patterns to verify his or her identity is referred to as keystroke biometrics. Traditionally, biometrics is associated with more *static* physiological characteristics, like a fingerprint. However, when people type it is very common for them to settle into a style or rhythm that becomes highly repetitive and thereby can be used as an identifying behavioral (or *dynamic*) trait. Fabian Monrose and Aviel D. Rubin in their paper "Keystroke Dynamics as a Biometric for Authentication" further note that these "behavioral traits … have some physiological basis, but also reflect a person's psychological makeup" [2]. Because of this, the more experienced, and thus reliant upon muscle memory, a user is; the easier the traits are to identify. It is for this somewhat ironic reason that the two-fingered "hunt and peck" method of typing has the advantageous quality of being nearly impossible to classify, due to the fact that the user's behavior is constantly changing over time.

To further clarify this, it is important to distinguish the difference between identification and verification. Identification is focused on trying to determine who a user is, while verification is centered around trying to make sure they are who they claim to be. To verify a user's claimed identity a check is made against the known profile for that individual, while identification requires being able to categorize a group of previously unknown users.

The underlying data that is used to generate a keystroke biometric based profile is composed of key logs. In the words of John V. Monaco in his paper "SoK: Keylogging Side Channels," "keystroke logging, or *keylogging* is the practice of recording the characters typed on a keyboard" [3]. From these key logs, the time intervals between the pressing of the different keys can be structured and then analyzed to assemble a user profile.

Keylogging data can differ based on the hardware used while typing. The principle piece of hardware is the keyboard, and keyboards vary in both their logical and physical layout. The logical layout "defines a mapping of key identifiers to physical locations,"

while the physical layout is how the actual keys on the keyboard are arranged and shaped [3]. Additionally, different keyboard models have different keys. Examples of these key types include the rubber dome, scissor switch, and mechanical switch, among others [3]. In short, the material makeup of a keyboard also plays an important role in how a user's keystroke biometric data is formed.

The typical way in which a user's data is collected is through browser-based observation. What is meant by this is the reality that when users open sessions with a web server over the Internet, under certain conditions they begin conveying more behavior than they may realize. For example, when a website has the autocomplete feature, every time a key is pressed, regardless of whether or not return is hit, an http request is sent to the server that transmits all of the information relating not only which character was pressed, but also how the user typed it [4]. An obvious example of this is seen when search engines generate suggestions before return is pressed. Over time, this information can easily be used to build a profile of anyone who regularly uses the browser. All of this results in the unfortunate reality that users who have the tendency to hash out their often-private thoughts by typing them out while they think, are unknowingly thinking out loud.

As is so often the case in computer science today, the rapid improvement in both software and hardware capabilities have opened doors to keystroke dynamic analysis applications that were previously relegated to theoretical discussion. As internet users attempt to hide their web surfing by using "internet privacy companies" or virtual private networks, most of them would be surprised to learn that their identity is still very identifiable [5]. Over the course of the past decade, a variety of commercial tools have been developed that can quickly and efficiently collect, identify, and authenticate a user's keystroke behavioral characteristics. While "it is not clear how many organizations routinely track user behavior and to what extent this is being used" the very presence of these actors is increasingly relevant in a modern society that is highly concerned with privacy on the Internet [6].

Before diving into the tools being used for tracking users, it is important to acknowledge that this technology has a far less controversial application: password hardening. The use of encrypted passwords remains the most commonly used method for protecting sensitive information, but “keeping the decryption key within reach is obviously dangerous” [7]. To address this problem, both commercial companies and academic researchers alike have developed software that uses an individual’s keystroke biometrics to securely verify his or her identity. Most of the publicly available information on methods of keystroke based user identification is provided by the creators of this kind of software. Of the many different methods being employed, the Long Text Identification Approach developed by Charles Tappert, Mary Villani, and Sung-Hyuk Cha of Pace University, along with the development of keystroke detectors by Kevin S. Killourhy and Roy A. Maxion of Carnegie Mellon provide a useful sample of the kinds of tools currently being implemented.

In their paper “Keystroke Biometric Identification and Authentication on Long-Text Input” [8], Charles Tappert, Mary Villani, and Sung-Hyuk Cha of Pace University outline their long text approach to identify users by their keystrokes. Their process begins with capturing user data. The submission is then stored as a text file that details the sequence number, the character, the code text equivalent, the location, the time pressed, and the time released for each key used. Their software then creates a feature vector that describes a user’s typing pattern. In the words of the authors: “the features are statistical in nature and designed to characterize an individual’s keystroke dynamics of writing samples of 200 or more characters” [8]. They then classify the feature vector for identification using a Nearest Neighbor classifier to compare the user-generated sample with the vectors in the training set. The user is identified by seeing which entry in the training set has the closest Euclidean distance to the test subject. Finally, the software classifies the user by authentication using a vector-difference model resulting in a “you are verified” or “you are not verified” binary output. Their results were very accurate, with a 99% success rate in ideal conditions, and a 90% success when presented with some abnormalities [8].

Computer scientists Kevin S. Killourhy and Roy A. Maxion of Carnegie Mellon, describe their comparison of several different keystroke dynamics-based detectors in their

paper “Comparing Anomaly-Detection Algorithms for Keystroke Dynamics” [9]. Much the same as the previously mentioned paper, the purpose of their research was to use “keystroke dynamics… as a biometric identifier” capable of meeting the European standard for access-control systems [9]. In contrast to the work out of Pace University however, Killourhy and Maxion built detectors capable of identifying users based on how they typed their passwords alone. First, they developed a Windows application for test subjects to input their chosen password, and then converted the data from their actions into a *timing vector*. From there they ran the data through 14 different anomaly-detection algorithms that ranged from being Euclidean to neural network based. To measure their success, Killourhy and Maxion used the equal error rate (EER) method. EER can be understood as the optimal performance of a system, a point in operation at which the total of false identification acceptance and false identification rejection is at its lowest. For this reason, the lower the score the better. While none of their detectors performed well enough to meet the European standard, half of them showed plenty of promise with the best performer (a Manhattan (scaled) detector) achieving an EER of 0.096.

These papers only represent a small portion of the wide variety of techniques used to identify individuals by their keystroke biometric data. Furthermore, it is important to note that while these researchers’ methods are likely quite similar to those used on commercial servers, it could very well be the case that corporations have developed far more sophisticated tracking techniques. This is difficult to know for certain because these commercial methods are not made public, but since they have plenty of incentive to refine their identification processes it can be surmised that significantly more effective tools are already used to track internet users.

As is so often the case in the field of computer science, for every defensive capability that is developed a corresponding offensive one is sure to follow, and in the world of keystroke dynamics analysis this certainly remains true. Keystroke biometric identification presents a unique problem because it cannot be deterred by many of the commonly used internet privacy tools. As long as the server side actor can view a user’s input events as they type on a browser, he or she is able to implement tools like the ones previously mentioned to reveal the user’s identity. Professors John V. Monaco and Charles

C. Tappert make this point very clearly when they reference TOR in their paper "Obfuscating Keystroke Time Intervals to Avoid Identification and Impersonation" [6]. As their introduction states, "Masking spatial information, such as IP address through TOR, is futile when temporal information, such as keystroke timings, can be used for identification." This reality has triggered the development of a variety of methods for obfuscating typing characteristics. Of the many different techniques being implemented, the previously mentioned research done by Monaco and Tappert as well as the paper "Author Attribute Anonymity by Adversarial Training of Neural Machine Translation" by Rakshith Shetty, Bernt Schiele, and Mario Fritz, of the *Max Planck Institute for Informatics* provide an excellent sample set of the kinds of measures being taken to attempt to obfuscate a user's keystroke biometric data or general writing style.

In their paper, Monaco and Tappert outline two different "mix" methodologies that randomly introduce delays or adjust the time interval of input events to make user behavior less identifiable. The first of their tools is the delay mix which introduces delays at both the start of the event (pressing a key) and the reception of the event (application observes the key being pressed). As they state in their paper, "the delay mix introduces noise to time intervals of their arrival process by randomly delaying each event" [6]. They also developed an interval mix which uses time intervals instead of delays to hide a user's identity. Instead of merely adding a given amount of time to when the input event is sent to an application, the interval mix transmits the input event within some random point in a set window of time. Both these tools were tested using a data set of a wide variety of typing styles and user characteristics. Their results concluded that while the methods employed were not identifiable to the user (the delays are impossible to notice with the naked eye), they were only able to achieve a 20–50% reduction in the ability to identify the users. While this number is not trivial, it still leaves considerable room for future improvement.

Rakshith Shetty, Bernt Schiele, and Mario Fritz first published their research on anonymization through the use of an adversarial training model at the USENIX security symposium in 2018. While their research was more catered towards natural language processing and not the obfuscation of keystroke biometric data, their work still outlines a potential framework to hide a user's keystroke dynamics-based identity. Their model is

both fully automatic and is capable of teaching how to obfuscate using pre-loaded data. In the words of the researchers, “we propose a sequence-to-sequence language model, inspired by machine translation, and an adversarial training framework to design a system which learns to transform the input text to obfuscate the author” [8]. This project was built upon their A4NT network which implements an adversarial trained model that is capable of conducting “style transfer without paired data” [8]. Their results were overall quite positive and their model was not only capable of fooling commonly used classifiers, but was also “likely to be effective against previously unknown NLP adversaries” [8].

Although their model was built for text and not keystroke based analysis, their framework could be applied to hiding a user’s typing behavior. Many of the same tools and classifiers that are used to identify the authors based off of their text, are also implemented to identify users based off of their keystroke biometric data. The adversarial training model is ultimately centered around using identifying qualities to strengthen an obfuscation technique. In keystroke dynamics, it is the rhythm of an individual’s typing that creates the fingerprint that is used to identify them. A model that transforms this rhythm by manipulating input events could very well hide a user’s keystroke based identity much the same way that the A4NT network did for text analysis.

While these, and other, approaches present viable means for hiding a user’s keystroke biometric data, none of them fully address the importance of both usability and functionality. These tools may be suitable in an academic setting, but this does not mean they are viable options for the average internet user who wants to better maintain his or her privacy. It is for this reason that this thesis proposes the development of a tool that is as simple as it is effective: one that can be used in real time, hides a user’s identity, and is not easily noticed while being used.

III. IMPLEMENTATION AND OBFUSCATION OF THE CLASSIFICATION PROTOTYPE

This project attempted to investigate aspects of keystroke biometrics based obfuscation that remain largely unexplored, in addition to looking into some of the questions posed in prior research. The focus of this thesis can be broken down into three research goals. The first goal was to develop a means of obfuscation that hides keystroke biometric data such that it renders different users' identities indistinguishable from one another. The second goal was to obfuscate without making it obvious that the individual implementing the technique is trying to hide his or her behavior. More simply put, this goal is centered around trying to ensure the obfuscated data blends in with unchanged user data. The final research goal was to develop a technique that is capable of being implemented in real time. This involves both ensuring that it obeys the natural rules of sequencing dictated by how typing is conducted, and minimizing lag to make it as usable as possible.

To date, most user keystroke obfuscation has been focused on randomizing biometric data to remove its identifying characteristics or implementing a model that obfuscates behavior by reacting to the actions being taken by the observing classifier. While these approaches are certainly effective at making individuals less identifiable, they also introduce several issues.

Firstly, depending on the size of the random number used, masking a user's identity can come at the cost of making the technique functionally unusable in real time. For example, without having an appropriately defined cap on the size of the random numbers used, buffering the keystroke data before transferring the input event to an application could result in a severe lag in the time between when a key is pressed and when the corresponding letter appears on the screen.

Secondly, adding randomness to user data can easily produce the unintended consequence of making the user even more identifiable as a result of their obfuscation. To better understand this phenomenon, imagine that there is one person who is using a ski mask to hide his or her face among a group of people. While it is true that the masked individual's facial characteristics are hidden, her can also be easily distinguished from his

peers as the “ski mask person.” In other words, it is possible for the means of obfuscation to become a flag capable of being used for identification if it is not properly implemented within the proper context. This reality resulted in the development of a two-layered approach to obfuscation for this project. This approach is displayed in Figure 1.

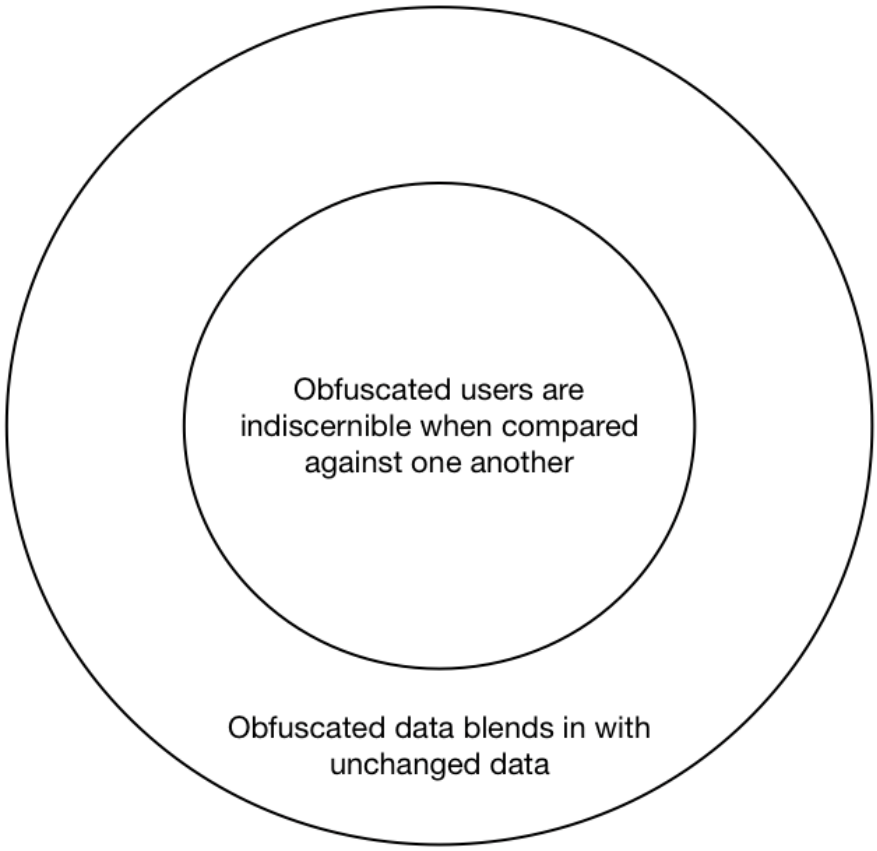

Figure 1. Two-Layered Obfuscation Methodology

The first layer consists of ensuring that the obfuscated data blends in with unchanged keystroke biometric data, as is stated in goal number two. Ideally, this is done well enough that the obfuscation goes completely unnoticed. However, in the event that the monitoring entities do notice that there is abnormal behavior going on, the second layer should have a sufficient number of users implementing the obfuscation technique that when they are grouped together they cannot be distinguished from one another. To this end, it is

important to note the relationship between being subtle and masking your identity, and how accomplishing both these goals requires compromise.

The unfortunate reality is that there is a somewhat inverse relationship between making data appear to be normal and changing it enough that it no longer contains a fingerprint. While the implementation derived was made to mitigate this problem as much as possible, it cannot be entirely undone. It is for this reason, that two things must be considered. First, in order to best accomplish both goals a balance must be struck between making changes to obfuscate the user's identity, and minimizing these changes to have the data set look as similar to an unchanged data set as possible. Second, in the event that the first layer does not effectively work, it is ideal to have as many users as possible attempting to obfuscate their behavior, as an increase in obfuscated users results in a decrease in the ability to guess at random which user is which.

The final goal of this research was to develop a means of obfuscation that was capable of being used in real time. There are two main requirements necessary to accomplish this goal. First, the method of obfuscation must follow the rules of natural human typing behavior. These rules mainly consist of ensuring that any changes made do not violate the necessary sequencing of input events that occur while typing. For example, a press time cannot occur after the press time that follows it. The second requirement is that the obfuscation technique requires minimal computational power to prevent any significant lag that could make the application less usable. To address this, this research used a non-parametric approach to obfuscation that both limits computational complexity and prevents error due to model driven assumptions.

As a final note, it is important to mention that the prototype developed in this project only works in a static environment by making changes to a pre-recorded data set. However, it was still made with the intention of being easily integrated into a real-time application. A description of this future implementation will be described in Chapter V.

A. IMPLEMENTATION

This project's obfuscation method works by manipulating several key aspects of a user's keystroke biometric data. While the data set used for this research had a wide array

of information regarding the users included in it, only the users' identifying number, the name of the key that they pressed, the time that they pressed the key, and the time that they released the key were necessary for accomplishing the obfuscation. Additionally, it is important to note that all of the press and release times were recorded sequentially in the initial data frame. The keystroke data frame format is displayed in Table 1.

Table 1. Keystrokes Data Set

user	key_name	press_time	release_time
1	shift	630364300	630364674
1	t	630364596	630364674
1	h	630364861	630364940
1	e	630364940	630365049
1	space	630365174	630365361
1	i	630365517	630365610
1	t	630366640	630366734
1	s	630368450	630368559

While there are a variety of users' typing characteristics that can be changed to make them less identifiable, the two major items that are used for classification are duration and latency. Duration is the amount of time that a key is pressed, and is calculated by subtracting the press time from the release time. Latency is the total time between keys being pressed. This is calculated by subtracting the current press time from the one that occurred before it.

The two principal means by which to change data can be broken down to changing either the rate or the magnitude. A change in rate refers to how often the change is made. For example, one could make changes to only half of the press times as opposed to all of

them. Changing magnitude refers to the degree to which something is changed. An example of this would be the size of the buffer added to a user's press time.

For this project, obfuscation was accomplished by making changes to both duration and latency with varying degrees of rate and magnitude. For duration, the user's identity was hidden by exchanging their duration times with that of another user. All of these exchanges were made using corresponding keys. For example, if the first user generated a duration while pressing the letter "a," then only a duration time that was also made while typing the letter "a" was used for the swap. The obfuscation program accomplishes this by first extracting all of the duration times in the data set, and compiling them in a dictionary with the key name as the key and the duration time as the value. The program then begins making its way through the data set a second time, during which it checks the current letter of the key that is being pressed and then swaps it with a random duration of the same letter type from the dictionary. The exchange process is displayed in Figure 2.

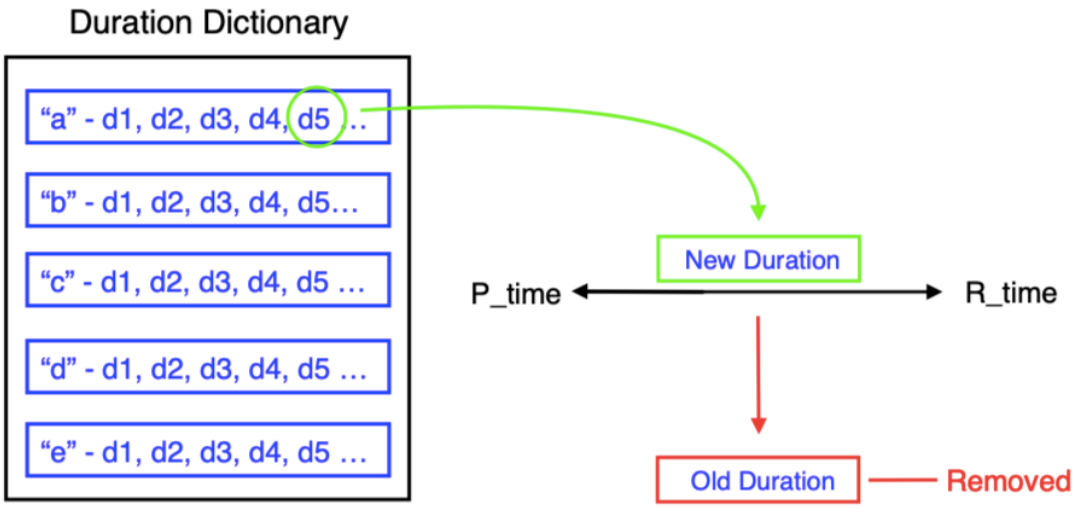

Figure 2. Duration Exchange Process

Changes to latency were made by randomizing the current latency time. This process begins with calculating the latency of the current keystroke by subtracting the next press time from the current press time. From there, a random number is chosen with an

upper bound of the calculated latency time. This ensures that a changed latency will never cause the current key press time to be greater than the press time of the key that follows it, which is not possible in a dynamic setting. The randomization process is displayed in Figure 3.

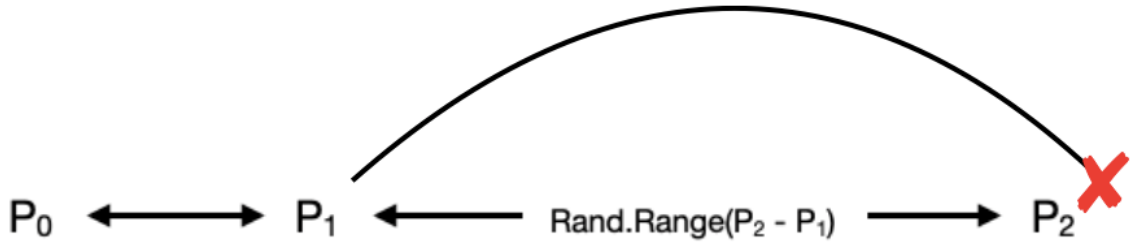

Figure 3. Latency Randomization Range

The rate of change was determined by using different probabilities to randomly manipulate a subsection of the data set. This change rate varied from 0 to 100% of the all of the user keystrokes in the data set.

The magnitude of a change was expressed by the potential amount of change made to latency alone. As previously mentioned, latency was changed by taking a random number that was no larger than the distance between two sequential press times. By increasing the upper bound of this range, the potential random numbers used to replace the original latency were on average larger.

Prior to being classified, all obfuscated data was transformed from its raw form as a series of key logs into a features file using the same technique described in Monaco, Stewart, Cha and Tappert's paper "Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works" [10]. The features file consists of key user characteristics derived from the keystrokes file. These user characteristics are determined by calculating the mean and standard deviation of a wide array of different groupings of a user's keystrokes. For example, the features file finds both the mean and standard deviation of the vowel and consonant keys. These values are

determined on a session-by-session basis, as opposed to deriving a single score for a given user across all of their typing sessions. This features file format is displayed in Table 2.

Table 2. Features File Data Format

user	session	du_all_keys_mean	du_all_keys_std	du_letters_mean	du_letters_std	du_vowels_mean
1	0d4c3d78-	109.670	61.528	98.967	24.597	111.400
1	7b9aa97c-	110.697	62.673	103.433	27.346	121.944
1	c3f4542b-	113.304	93.978	94.730	29.336	108.324
1	e4a6add1-	118.103	79.043	98.882	31.694	108.688
4	095cad3b-	102.976	60.359	96.356	29.620	96.697
4	3ab0e58b-	104.878	77.354	93.353	25.606	91.700
4	7471d0fc-	100.811	70.524	97.057	29.190	98.000
4	b124eeb4-	96.282	45.348	90.965	30.005	98.794

B. CLASSIFICATION

All testing was done using a random forest classifier. Random forests essentially work by constructing a large number of decision trees and averaging their results to come to a final conclusion. They are particularly useful because they handle estimates of error internally which prevents the need for any additional testing. There is precedent for their use in keystroke dynamics-based classification. Examples of this include Roy A. Maxion and Kevin S Killourhy's research titled "Keystroke Biometrics with Number Pad Input," among many others [11]. An illustration of a random forest and its computational technique is displayed in Figure 4.

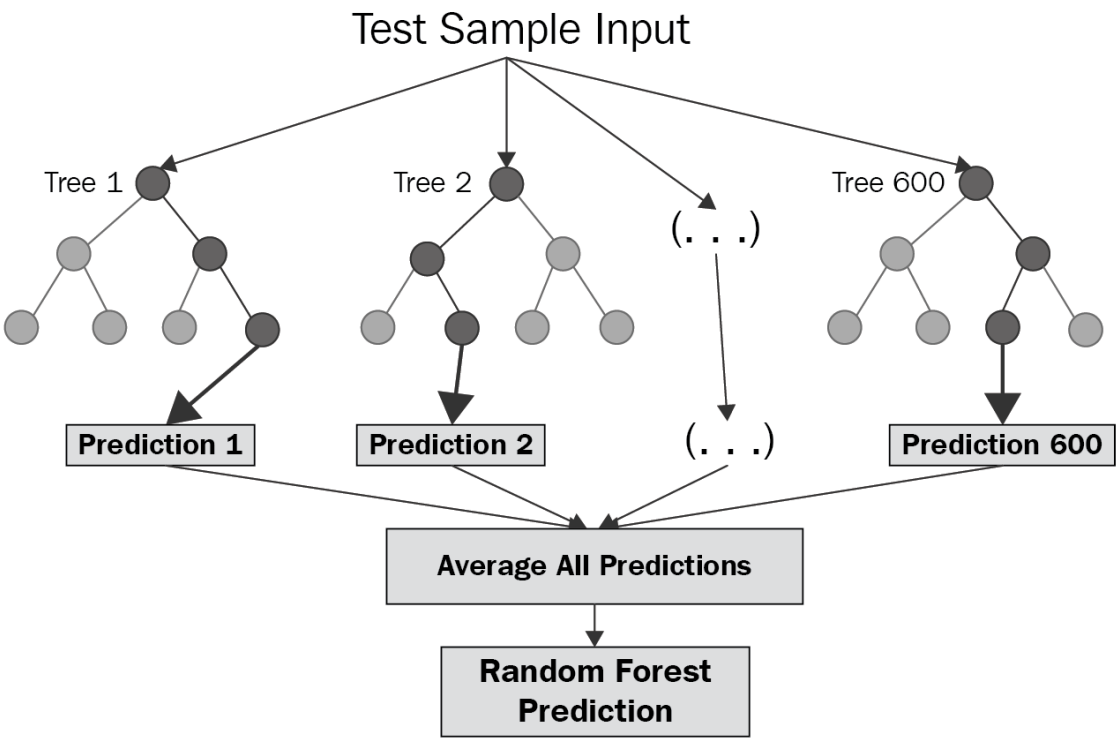

Figure 4. Random Forest Diagram. Source [12].

The training sets the random forests used were determined by assuming that the entity attempting to do identification had access to the obfuscation technique, and was able to generate its own obfuscated data set to compare against unchanged keystroke data. As a result, all classification was done using a 50-50 split between obfuscated and unchanged keystrokes, with each of them properly labeled as either changed or unchanged data.

IV. RESULTS AND ANALYSIS

The classifier's output scores were gathered over the course of several hundred simulations run with different parameters for both latency and duration to test how changes to rate and magnitude affect the identifiability of users and the detectability of the obfuscation technique. These tests also helped reveal how changes made to latency and duration affect the overall accuracy of the classifier, as well as how these parameters impact one another.

Changes to rate were made by modifying the number of duration exchanges completed. All exchanges were selected randomly. Modifications to magnitude were made by using different percentages of the potential random ranges for a given latency (recall that this range is determined by calculating the difference between subsequent press times). For example, a test done with 50% of the latency range would have greater magnitude than one done with 25%.

Since the obfuscation is intended to implement changes to both latency and duration, all simulations were completed with the variable (either duration or latency) that is not currently being tested at either 0% or 100%. For the discussion of these results, the variable that is not currently being tested will be called the co-variable. Including the co-variable helped illuminate how the interaction between latency and duration affect the accuracy of the random forest classifier.

The results are displayed in sets of two tables that list the classifier's accuracy scores for identifying users and detecting whether obfuscation was being used, against the rate or magnitude tested. These tables are displayed in Table 3, Table 4, Table 5, and Table 6. The data from both of these sets is then plotted on the same chart to display the correlation and effect between the two variables. These plots are displayed in Figure 5, Figure 6, Figure 7, and Figure 8.

A. DATA

Table 3. Rate Change with 0% Latency Data Set

Identifiability vs. Rate of Change (0% Latency)

RATE AS % OF DURATIONS CHANGED	IDENTIFIABILITY
0.00	0.78
4.80	0.75
9.77	0.77
24.35	0.70
48.07	0.59
71.96	0.50
96.57	0.40

Detectability vs. Rate of Change (0% Latency)

RATE AS % OF TOTAL CHANGED	DETECTABILITY
0.00	0.50
4.80	0.52
9.77	0.57
24.35	0.73
48.07	0.89
71.96	0.97
96.57	0.96

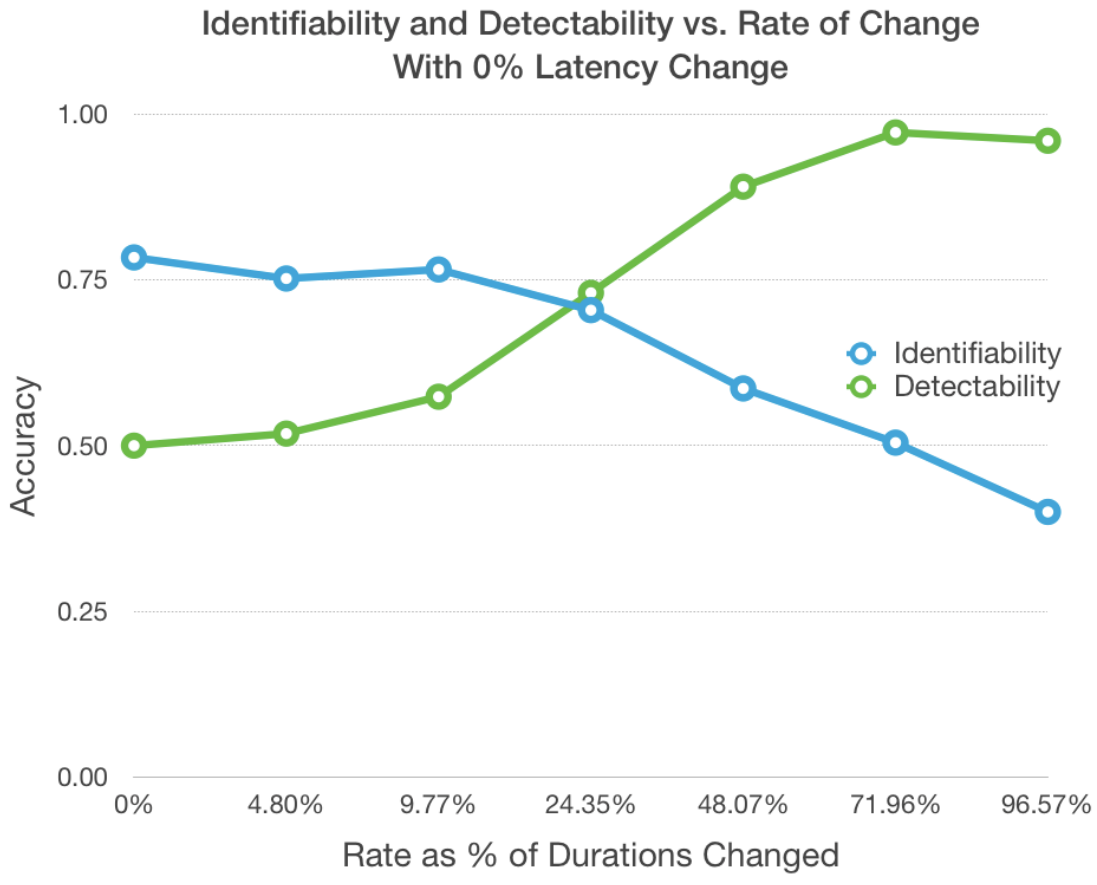

Figure 5. Change in Rate With 0% Latency Plot

Table 4. Rate Change with 100% Latency Data Set

Identifiability vs. Rate of Change (100% Latency)

RATE AS % OF DURATIONS CHANGED	IDENTIFIABILITY
0.00	0.39
5.00	0.34
9.90	0.35
24.27	0.28
48.28	0.20
71.90	0.24
96.57	0.19

Detectability vs. Rate of Change (100% Latency)

RATE AS % OF TOTAL CHANGED	DETECTABILITY
0.00	0.99
5.00	1.00
9.90	1.00
24.27	0.98
48.28	1.00
71.90	1.00
96.57	0.95

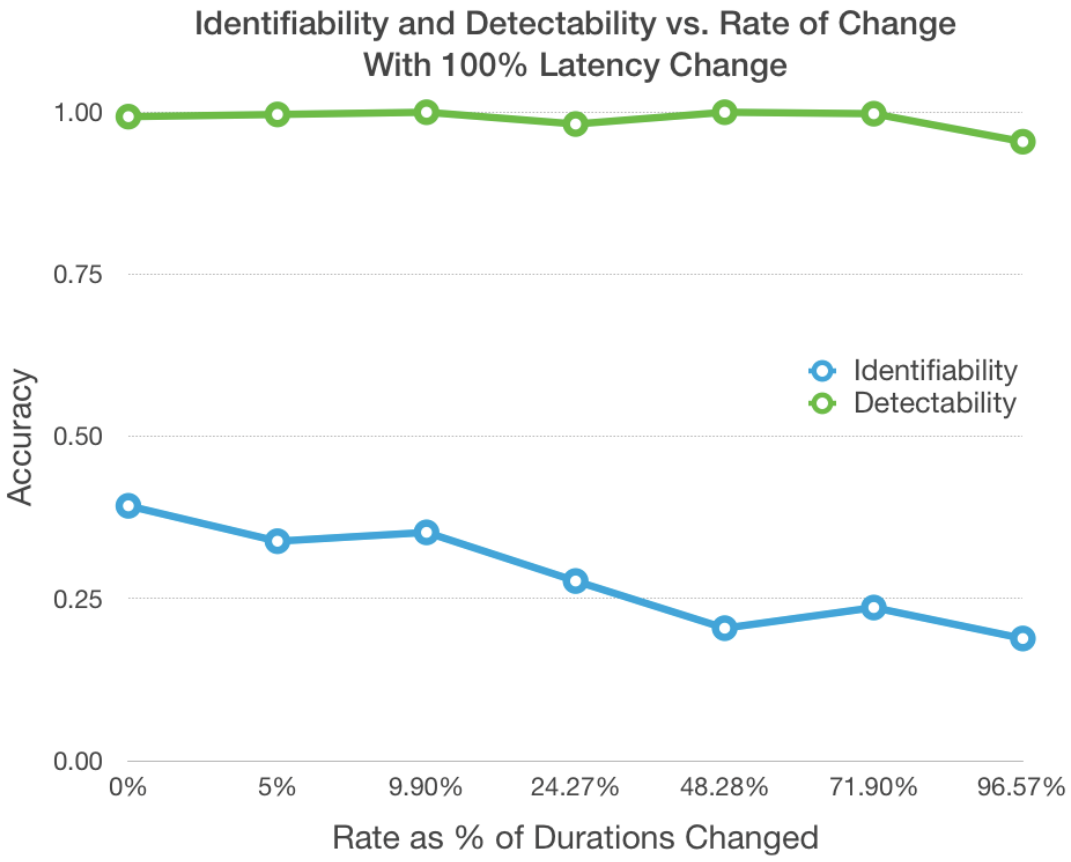

Figure 6. Change in Rate With 100% Latency Plot

Table 5. Magnitude Change with 0% Duration Data Set

Identifiability vs. Magnitude of Change (0% Duration)

MAGNITUDE AS % LATENCY CHANGE	IDENTIFIABILITY
0.00	0.77
5.00	0.78
10.00	0.74
25.00	0.72
50.00	0.59
75.00	0.41
100.00	0.33

Detectability vs. Magnitude of Change (0% Duration)

MAGNITUDE AS % LATENCY CHANGE	DETECTABILITY
0.00	0.50
5.00	0.57
10.00	0.67
25.00	0.87
50.00	0.94
75.00	0.98
100.00	0.99

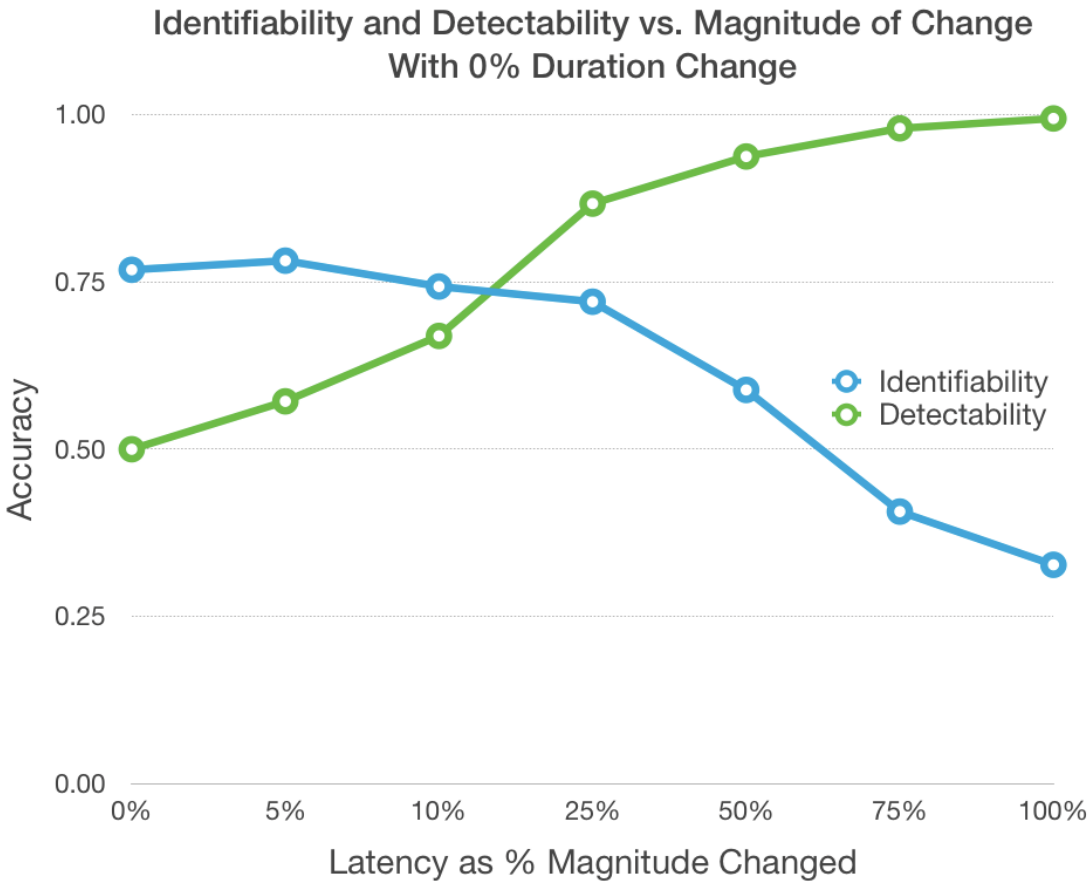

Figure 7. Change in Magnitude with 0% Duration Plot

Table 6. Magnitude Change with 100% Duration Data Set

Identifiability vs. Magnitude of Change (100% Duration)

MAGNITUDE AS % LATENCY CHANGE	IDENTIFIABILITY
0.00	0.42
5.00	0.40
10.00	0.38
25.00	0.30
50.00	0.24
75.00	0.24
100.00	0.17

Detectability vs. Magnitude of Change (100% Duration)

MAGNITUDE AS % LATENCY CHANGE	DETECTABILITY
0.00	0.96
5.00	0.96
10.00	0.97
25.00	0.96
50.00	0.94
75.00	0.94
100.00	0.96

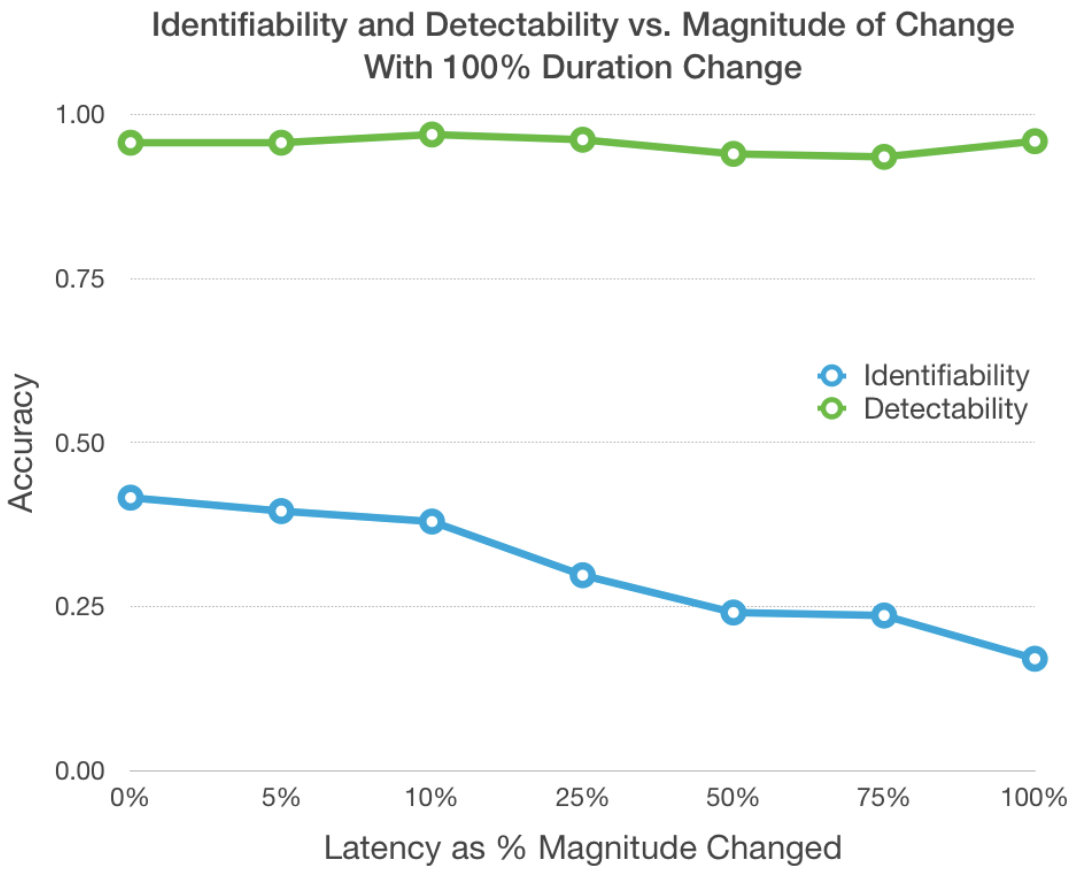

Figure 8. Change in Magnitude with 100% Duration Plot

B. ANALYSIS

The data set used for testing the obfuscation technique consisted of 22 different users. Each of these users participated in four different sessions during which they were

recorded typing nursery rhymes. For each of these sessions, the users typed approximately 153 keystrokes.

The obfuscation method had varying degrees of success at achieving the aforementioned research goals. To begin with, the technique generally performed well at obfuscating users' identities. The classifier's baseline identification accuracy was 0.78, since the random forest was able to correctly identify users 78% of the time on a data set that had no obfuscation done to it. Thus, any obfuscation that resulted in a score lower than 0.78 succeeded in making users less identifiable, and anything above 0.78 increased their identifiability. With both the rate of duration exchanges and the magnitude of latency changes set to a maximum, the classifier's ability to identify users dropped nearly four-fold to between 17% and 19% accuracy.

Changes to rate and magnitude done independently also lowered the random forest's identification accuracy, but it was not nearly as effective as when they were used together. Conducting duration exchanges at 100% and latency changes at 0% resulted in the identification accuracy being lowered to 40%, while completing the inverse with latency changes at 100% and duration exchanges at 0% caused the accuracy score to drop to 33%.

The random forests did a better job of detecting changes made to the keystroke data. The classifier's baseline detection accuracy was 0.5 or 50%, since there is a 50% chance of guessing correctly when an unchanged and changed data set are compared side by side. For changes made to rate alone, when just 25% of the potential duration exchanges were completed, the detection accuracy rose to 73%. For magnitude changes made independently, the obfuscation was even more detectable. With the latency magnitude at 25%, the random forest was able to detect that a change was made 87% of the time. When rate and magnitude changes were tested with either of the co-variables set at 100%, the obfuscation became highly detectable. In both series of tests, detectability never went below 94% accuracy.

Overall, the duration exchanges were less detectable than the random ranges used to manipulate the latencies, but in both cases the program was not able to obfuscate enough

to thoroughly hide users' identities before being detected. For the duration exchange, while around 10% of the durations were able to be swapped before the behavior became highly noticeable, it took approximately a 50% rate of change to see a substantive drop in the classifier's identification accuracy. Unfortunately, at a 50% rate of change the obfuscation became detectable 89% of the time. Similarly, a 50% latency magnitude change was required to have a meaningful impact on user identifiability, but at this magnitude the changes were 94% detectable.

THIS PAGE INTENTIONALLY LEFT BLANK

V. FUTURE WORK AND CONCLUSION

A. FUTURE IMPLEMENTATION

This research's prototype has room for significant improvement, particularly in optimizing the duration exchange to make its use less detectable. While significantly lowering users' identifiability came at the cost of making their efforts at obfuscation obvious, this prototype did successfully make a non-trivial amount of change that went unnoticed. This was particularly true for the duration exchanges. Nearly 10% of all potential exchanges were able to be completed while only increasing the detectability of this behavior from the baseline score of 50% to just 57%. With over 13,000 keystrokes logged, this implies that over 1000 exchanges on the data set were able to be made while remaining largely unnoticed by the classifier. Were these exchanges to be made more precisely and focused on keystroke groupings that have an outsized impact on generating a fingerprint, they could provide enough of a chink in the classifier's armor to successfully obfuscate users' identities in a subtle manner.

Another potential means of optimization would be to use a large data set that better prevents the possibility of overlap in which a user's data is exchanged with one of his or her own duration times. While it is unlikely this played an outsized role in this research's results, the increased variance caused by using a larger number of keystrokes would almost certainly help better mask a user's identity while keeping the obfuscation undetectable.

While the prototype developed in this research only makes changes to static data, as previously mentioned, everything was constructed so that it was capable of being used in a live setting. There are a number of potential methods for accomplishing this, but two of the most straightforward ones are to use an executable or a device driver.

Before continuing, it is important to understand how the operating system logs keystrokes and subsequently transmits these keystrokes to any connected applications that require them. The process begins when a user presses a key which generates an interrupt. The kernel sees the flag and processes the interrupt by creating the input event in which the keyboard scan code is mapped to the key code for the letter pressed. This information

is then written to a virtual file called the keyboard device file. Other applications, such as an internet browser, are able to view this file to see what keys the user is pressing.

To implement this project's obfuscation technique in real time using an executable in user space, the middleware would have to lock the keyboard device file while reading from it, change the duration and latency times for certain keystrokes as required to accomplish the obfuscation, and then unlock it for other applications to view the now altered keystroke biometric data. Middleware made using a device driver would operate in kernel space and make the necessary changes right when the initial interrupt is being converted into an input event. The middleware itself could be located in a hardware device that sits between the keyboard and the computer, or as software that resides on the computer itself. A diagram of the obfuscation middleware as hardware is seen in Figure 9.

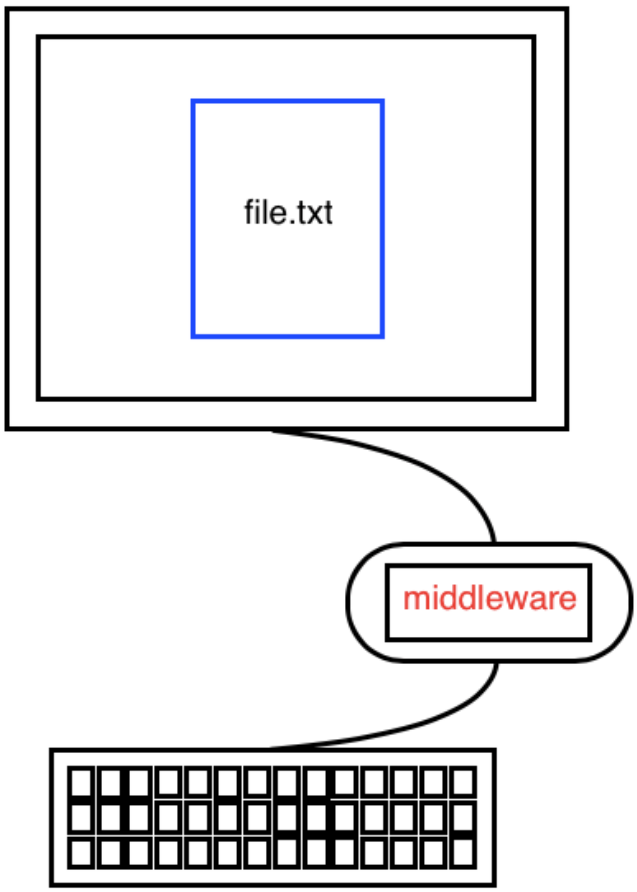

Figure 9. Middleware as Hardware

B. CONCLUDING THOUGHTS

At its core, this research attempted to strike a balance between two seemingly inversely correlated characteristics. Using camouflage to decrease a user's identifiability and impersonation to limit the detectability of their attempts at obfuscation presents the unique challenge of trying to make changes that not only hide an identity, but essentially forge a new one. By exchanging individuals' keystroke biometric data, changes were made in a manner that prevented the obvious abnormalities that strictly machine-driven obfuscation like randomization create. However, while this did successfully generate a patchwork identity that no longer resembled the original user, the obfuscation technique failed to create a typical human keystroke biometric profile. Instead of a state of the art

mask or a thorough yet subtle disguise, this technique seems to have resulted in a somewhat zombie-like identity that is constructed of human elements, but when viewed in totality is clearly not a regular human profile. In this way, the results of this project revealed the tradeoff between obfuscating an identity, and concealing the presence of an obfuscation technique. In the future, better calibrating this newly minted identity could lead to successfully finding the balance between camouflage and impersonation necessary for subtly yet thoroughly hiding a user's keystroke biometric profile.

Perhaps this process of obfuscation through the generation of new identity charts a potential path forward for all internet privacy advocates. Since the dawn of online accounts, humans have been drawn to the internet's anonymity that allows them to cultivate an alternate identity. Unlike other domains where an individual's potential seems only as great as their physical appearance, background, or education, the Internet allows users to explore a world in which their personality alone defines who they are and where they stand amongst their peers. It is this phenomena of creating a new internet identity that may be at the center of ensuring internet privacy moving forward. If at every technological layer of the web a user's data is transformed to appear to be a new human profile, their real-world identity can be hidden while their efforts at securing their privacy remain undetectable.

LIST OF REFERENCES

[1] Kenneth Research, “Virtual Private Network (VPN) Market: Global Industry Analysis, Size, Share, Growth, Trends, and Forecast, 2018–2025: With CAGR More than 18.26%,” *MarketWatch*, 19-Dec-2019. [Online]. Available: https://www.marketwatch.com/press-release/virtual-private-network-vpn-market-global-industry-analysis-size-share-growth-trends-and-forecast-2018-2025with-cagr-more-than-1826-2019-12-19

[2] F. Monrose and A. D. Rubin, “Keystroke dynamics as a biometric for authentication,” *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351–359, 2000.

[3] J. V. Monaco, “SoK: Keylogging Side Channels,” presented at *2018 IEEE Symposium on Security and Privacy (SP)*, 2018.

[4] J. V. Monaco, “Feasibility of a Keystroke Timing Attack on Search Engines with Autocomplete,” presented at *2019 IEEE Security and Privacy Workshops (SPW)*, 2019.

[5] Duck Duck Go, “About DuckDuckGo,” 2020. [Online]. Available: https://duckduckgo.com/about

[6] J. V. Monaco and C. C. Tappert, “Obfuscating Keystroke Time Intervals to Avoid Identification and Impersonation,” Pace University, Sep. 2016.

[7] R. W. F. Lai, C. Egger, M. Reinart, S. S. M. Chow, M. Maffei, and D. Schroder, “Simple Password-Hardened Encryption Services,” *USENIX*, vol. 27, Aug. 2018.

[8] C. C. Tappert, M. Villani, and S.-H. Cha, “Keystroke Biometric Identification and Authentication on Long-Text Input,” *Behavioral Biometrics for Human Identification*, pp. 342–367, Jan. 2009.

[9] K. S. Killourhy and R. A. Maxion, “Comparing anomaly-detection algorithms for keystroke dynamics,” presented at *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, Aug. 2009.

[10] J. V. Monaco, J. C. Stewart, S.-H. Cha, and C. C. Tappert, “Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works,” presented at *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2013.

[11] R. A. Maxion and K. S. Killourhy, “Keystroke biometrics with number-pad input,” *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, Chicago, IL, 2010, pp. 201–210.

[12] A. Chakure, “Random Forest and its Implementation,” *Medium*, 10-Jan-2020. [Online]. Available: https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California