5-2014

Physical Layer Detection of Hardware Keyloggers

Saptarshi Mallick
Utah State University

PHYSICAL LAYER DETECTION OF HARDWARE KEYLOGGERS

by

Saptarshi Mallick

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Engineering

Approved:

Dr. Ryan Gerdes Major Professor	Dr. Ming Li Committee Member
Dr. Koushik Chakraborty Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2014

Abstract

Physical Layer Detection of Hardware Keyloggers

by

Saptarshi Mallick, Master of Science

Utah State University, 2014

Major Professor: Dr. Ryan Gerdes
Department: Electrical and Computer Engineering

This work is aimed at the detection of the presence of hardware keyloggers (HKL). Here, of the two types of HKLs (active and passive), passive HKLs were dealt with. Passive keyloggers are placed in parallel to the keyboard-PC connection and may not draw power from the host device, unlike active ones that are placed in series with the keyboard-PC connection and may draw power from the host device.

There are certain electrical characteristics (features) of a system which were seen to be affected while the keylogger was attached to the system. The HKL is circuit-modeled in such a way that its effect on the electrical characteristics can be studied and predicted. It was possible to detect the presence of HKL by its effect on the electrical characteristics of the system.

It was seen that the behavior of the features changed with the ambient temperature where the experiment was carried out. So, experimental work was carried out in order to find the dependency of the features on ambient temperature. This was done to restore the stability of features. It was found that temperature is a good predictor of the electrical features of the PS/2 keyboard clock.

Stealthy (drawing power from a place other than the host) and evasive (having the technology to evade detection techniques) keyloggers were also considered; that evade the

primary detection approach mentioned here. It was also found that none of these customized keyloggers were able to make themselves immune to the secondary detection approach mentioned here. From 4 to 100 keystrokes were found necessary in order to detect passive and evasive HKLs attached to the circuit given baseline data (training data, i.e., no HKL attached). Experiments were performed in order to identify the HKL without training data on keyboards. This was done in order to see whether the HKL causes any consistently apparent change in the keyboards, i.e., whether the HKL affects keyboards uniquely.

(77 pages)

Public Abstract

Physical Layer Detection of Hardware Keyloggers

by

Saptarshi Mallick, Master of Science

Utah State University, 2014

Major Professor: Dr. Ryan Gerdes
Department: Electrical and Computer Engineering

This work addresses the problem of detecting devices which are stealthily attached to the computer for logging keystrokes from keyboards. These devices are known as hardware keyloggers (HKL). When an HKL is attached to the keyboard, certain electrical characteristics of the keyboard signal are altered. Based on these characteristics (features), differences have been identified and an accurate assertion was made about the presence of HKL.

The characteristics from which the differences were obtained were used to make distributions and compared with distance-measuring methods. An experiment was done to collect data from a number of keyboards and form two distributions (training and test) to perform the comparison. It was possible to detect the presence of HKL in the keyboard with a minimum of 4 to 100 keystrokes.

For justifying the stability of the features, the temperature of the surroundings was obtained and the dependence of the features on temperature was obtained. Also, an experiment was done to see whether the keyboards were uniquely affected by the HKLs. This was done without using any training data, i.e., the distribution of features which was used did not come from a known state of the system (either with HKL or not with HKL).

To my parents about whom I don't have any words to describe and extend my thanks.....

Acknowledgments

I want to thank my parents Sharmila and Subrata Mallick for their immense love and support. They believed in me and helped me all the way, without whom this journey would have been impossible. I want to thank my friends and seniors for their support all through this academic course.

I am also grateful to Dr. Ryan Gerdes for his immense help all through my master's program and without whom I wouldn't have got the opportunity to learn what I have been able to learn today. I want to thank all my committee members for being patient and helping me out all through my master's program.

Saptarshi Mallick

Contents

List of Tables

List of Figures

Acronyms

HKL	Hardware Keylogger
PLI	Physical Layer Identification
PC	Personal Computer
μC	Micro Controller
GPIO	General Purpose Input Output
PS/2	Personal System/2
KLD	Kull-Back Leibler Divergence
SVM	Support Vector Machines
OneClass-SVM	One Class Support Vector Machines
AWG	Arbitrary Waveform Generator
MOSFET	Metal-Oxide Semiconductor Field Effect Transistor
ADC	Analog to Digital Converter
DAQ	Data Acquisition
PLD	Physical Layer Detection
MCC	Measurement Computing
USB	Universal Serial Bus
pF	Pico Farad
mV	Mili Volt
EER	Equal Error Rate

Chapter 1

Introduction

Keyloggers are devices which log each keystroke from a keyboard and record them. Keyloggers can be of two types, hardware keyloggers (HKL) and software keyloggers. While HKLs are pieces of hardware attached to the host computer and have their own memory, software keyloggers are software programs running on the host system. While the software keyloggers need to be installed, the HKLs do not depend on any application or do not even depend on power from the host device. There are a variety of software keyloggers available on the market and much work has been done on detecting software keyloggers. In contrast, the detection of HKLs is still an outstanding problem. Here, HKLs are dealt with; the scope of this work does not include software keyloggers. One can argue that HKLs can be detected simply by verifying the peripheral devices connected to the PC, but it should be noted that an HKL can be placed inside a keyboard [1] or for that matter inside a PC.

1.1 Motivation

A number of incidents suggest that HKLs are a threat. They have been used by students to perform unethical actions such as changing grades on exams [2]; they have been installed by hackers on victims' computers [3] to steal money from password-protected bank accounts, which represents a threat to the public [4]; and again keylogger-like devices were placed by hackers on credit card readers where customers swipe their cards in a number of stores across the US in order to steal credit card information. There are a number of HKLs which have been made and are available on the market for sale. One can buy these keyloggers for around $12–$30 or make them with the help of microcontrollers [5]. There are two kinds of HKLs: active and passive. Only passive HKLs have been used for this work. Active keyloggers are both input and output devices which are placed in series with

the keyboard and PC (Personal Computer). When the keyboard communicates with the PC, it takes in the signal from the keyboard and outputs it to the PC with the same thing happening when the PC communicates with the keyboard. They can take power from the bus of the keyboard-PC interface or can be self-powered. Passive keyloggers on the other hand are placed parallel to the keyboard-PC line and can be self-powered (more details and figures in Chapter 2). This is just an input device and does not output anything but causes a kind of *loading effect*. Loading effect is the decrease in the voltage available from the source to the load resistor. It is hypothesized that if there is a hardware device attached to a system there will be distortions caused in the circuit signal. This hypothesis has been validated by doing experiments with the electrical signals from the keyboard. This work exclusively focuses on detection of passive HKLs, more importantly leveraging the fact that it causes a loading effect in the line signal (more details in Chapter 2).

As the HKL has been built using a microcontroller (drawing power from a separate USB bus) it has both input resistance of its pins and also an input capacitance which will cause loading effects [6] and transient effects [7] respectively. Further experiments considered a custom HKL which would make it evasive, which means that it will have a very high input resistance and an input capacitance on the order of pF, evading the primary detection approach mentioned here. The HKL considered here is stealthy, which means that it draws power from a different bus other than the bus connecting the keyboard and PC.

In order to perform the comparison between the two states (HKL connected and not connected), sophisticated Physical-Layer Identification (PLI) methods were used. PLI methods have been used previously in wireless devices by Danev et al. [8] and in Ethernet devices by Gerdes et al. [9]. In order to employ the PLI methods [10], device fingerprints should be procured [11, 12], which should remain constant (Reason for fingerprinting the clock line of the keyboard in this case). In this work, PLI was used in detecting hardware devices (HKLs) which is done for the first time and also opens a new domain for PLI. The reason these methods are termed *Physical-Layer Identification* is that the device (keyboards, in this case) is fingerprinted at the physical layer to obtain the data and perform

comparisons, setting a baseline for the calculations. PLI requires that the features be compared with a baseline. The features used here were voltage level when the PS/2 clock signal was in steady state and the transient effects when the PS/2 clock signal was transitioning from a high to low or from a low to high voltage (details in Chapter 2).

The loading effect caused by the passive HKL causes changes in the clock signal. As a result, the PLI techniques used here for detecting small differences in the clock signal represent the first work as far as this area is concerned. The wire connecting the keyboard and the PC was tapped by the custom keylogger and the line voltage was measured when it was connected and when it was not connected. The rise time, fall time, and slew rate (transient response) of the clock signal with and without the HKL attached (detailed discussion of the experimental setup is in Chapter 4) were measured by connecting a high-speed ADC. The line voltage difference being negligible for these evasive keyloggers, their presence was detected by the change in transient response of the circuit.

1.2 Related Work

A software detection method for in-line (active) HKLs was proposed by Mihailowitsch [13]. The author looked at the power and the signal propagation time consumed when the HKL was placed in the circuit. These discrepancies in the circuit will not be the case when passive keyloggers are taken into account. They might not draw power from the host device but sit parallel to the line between PC and keyboard. Ming-Chang et al. [14] took into account that hardware keyloggers could be detected by looking for any unknown hardware piece connected to the peripheral devices attached to the PC. Also, it is a common notion [15] that visual detection is the best way to detect HKLs. This method will not be successful, as discussed previously, when the keylogger is placed inside the keyboard or inside the PC.

Coming to the ways in which the keyloggers can be defeated, Keshet et al. [16] stated that if the keyloggers were incapacitated, which means that the memory of the HKL can be filled up with useless data (affecting the storage capacity of the keylogger), then the keyloggers were not able to capture data anymore. This will not be as effective in cases

where the HKLs have high memory capacity as stated by Baloch [17]. More or less the same idea, except for overwriting the memory, was stated by Greene and Parker [18]. No research has been done on detecting passive HKLs or on detecting HKLs not using software methods, thus motivating the work presented here.

1.3 Organization of Chapters

After the introduction in this chapter, the next chapter discusses the circuit model used in the experiments, along with the threat model and the working of the PS/2 protocol for the HKL. The third chapter discusses the feature extraction algorithm and proposed architecture for detecting HKLs. The features are electrical characteristics, viz. voltage levels and transient effects (rise time, fall time, and slew rate) of the clock signal. These features were used to form distributions which were compared with a baseline training distribution. A threshold was set and both the known and unknown states were determined according to the threshold. The error rates (results) of going above and below the threshold were obtained for the known and unknown states respectively. The distance comparison algorithms discussed here were used to compare the two distributions. The fourth chapter discusses the experimental setup and the results obtained for the distance comparison algorithms. The fifth chapter discusses the experimental setup where the temperature effects were considered in order to prove that the features were unstable with temperature, and results were obtained. The sixth chapter discusses ways to detect an HKL without a baseline training distribution. This was done using data-mining and machine-learning methods, which include Support Vector Machines (SVM) and one-class SVMs. The seventh chapter puts forth conclusions and proposes future work.

Chapter 2

Characterization of a Keylogger: Its Circuit Model and Its Detection

The circuit model of the HKL is discussed here. The model shows that its parallel connection to the keyboard's clock line takes only the clock signal as input. It does not output anything but only causes loading effects and transient effects. The device fingerprinting and the PS/2 protocol are also described in order for easier understanding of the process of tapping the clock line by the HKL and how it is affected by that.

2.1 Keylogger Description and Related Assumptions

In this work, the PS/2 protocol [19] was used rather than USB, considering the fact that the signal recovery and analysis for USB keyboards is far more complex than for PS/2 keyboards. Moreover PS/2 and USB buses act on the same principle, proving that the method works for the principle as a whole and not just for the PS/2 protocol. The PS/2 protocol is described in detail in the next section. As discussed in the previous chapter, the keylogger built here is passive; it is placed parallel to the keyboard-PC line and does not draw power from the PS/2 bus. Figure 2.1 shows, in a generic way, how the connections are done in an active HKL (Figure 2.1(a)) and a passive HKL (Figure 2.1(b)). It can be seen from the figure that the passive HKL only takes the data and the clock from the bus. It can either be self-powered or it can draw power from the bus connecting the keyboard and PC. In this work, a self-powered passive HKL was considered in order to maintain its stealth so that the detection approach, mentioned here, was not biased.

The main reason that active HKLs were not taken into consideration for this work is that Mihailowitsch [13] had already proposed a method for detecting them. In all, there is less research on detecting passive HKLs, passive being the rarest of its kind, so more stress

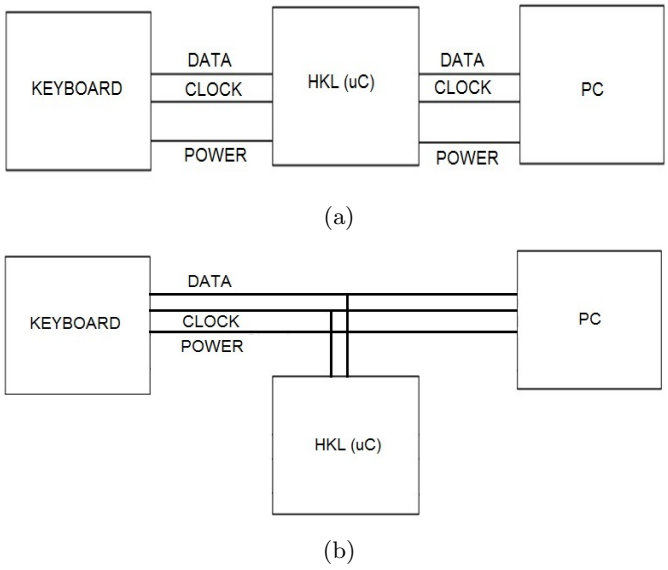


Fig. 2.1: (a) Active HKL acting as both input and output device. It is connected in series with the PC and the keyboard and can draw power from the bus with which it is connected. (b) A passive HKL which only acts as an input device. The input from the keyboard to the PC is only tapped by the HKL and monitored. It is placed parallel to the keyboard and PC and it does not draw power from the bus with which it is connected.

is given on the detection of passive HKLs.

2.2 Threat Model

A μC (microcontroller) based HKL (as it is the most common one [5]) was used for this work. The μCs have General Purpose Input Output (GPIO) ports which serve as input to the place of tap when attached as HKL in the circuit. It is assumed that an attacker places the μC-based keylogger in between the keyboard and the PC that reads the keystrokes. It should be noted here that the point where the keylogger is connected in between the keyboard and the PC does not affect the detection approach. The lumped element model [20] states that $L_c << \lambda$. Here L_c denotes the circuit's characteristic length which is shorter compared to the wavelength, λ. Wavelength, being the inverse of the signaling frequency between the keyboard and the host, is of a higher value as the frequency is less for PS/2 [19]. Thus, the actual point of attachment of the keylogger, be it inside the PC/keyboard or outside, does not matter as long as it is between the keyboard

and the PC interface. For this reason, the model says that a passive HKL is placed anywhere in between the keyboard and PC which logs keystrokes.

2.3 Overview of the PS/2 Protocol

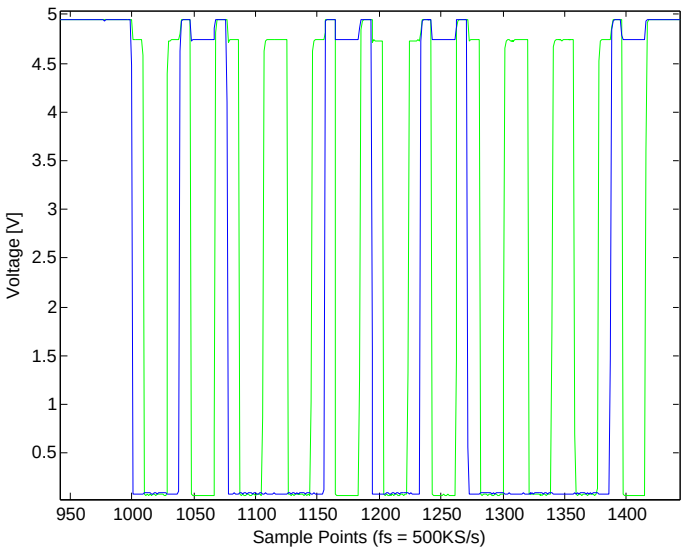
Figure 2.2(a) shows the clock (red in the figure) and data (blue in the figure) signal of a PS/2 keyboard. Here, high was +5 V and low was 0 V. Both the clock and the data lines were at high state when no key was pressed, i.e., no data was being transmitted, which is the idle state. As soon as a key is pressed on the keyboard, the data line goes low and that in turn pushes the clock line to go low. The PC reads data and samples it from keyboard when the clock goes negative. The PS/2 bus of the keyboard has four wires which were used here [19]. These are clock, data, power (+5 V DC at 275 mA), and ground. The frequency of the PS/2 clock is 10 kHz to 16.7 kHz.

The clock and the ground signals, but not the power signal, are needed to connect to the keylogger, as it is stealthy. The data line is needed by the attacker for recovering the keystroke but was not used for detection here in this work. All the calculations here were done using the clock line. This is because the clock line is not only a repetitive signal (satisfying the PLI requirement of fingerprinting as stated in the previous chapter) but also it is the same for all the keystrokes for a particular keyboard and remains almost the same for different keyboards, too.

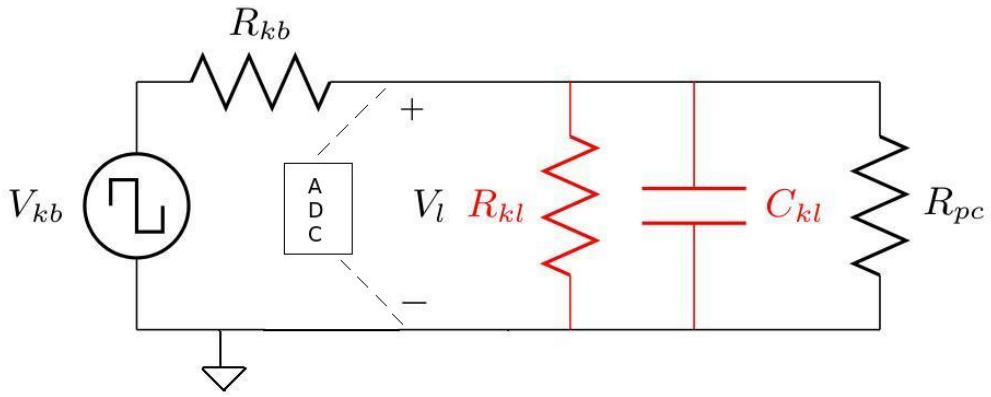
2.4 Circuit Model of the Keylogger

The HKL, which was modeled as a first-order RC-circuit (Figure 2.2(b)), is presented here. The GPIO ports of the microcontroller, used to model the HKL, control the input of the HKL in the circuit. From the figure it can be seen that the HKL can be represented with a resistance R_{kl} and C_{kl} (both being finite quantities) parallel to each other. The resistance is produced from the resistance of the GPIO ports of the μC due to the leakage currents. The capacitor has a capacitance on the order of pF and the HKL input resistance is on the order of kΩ at higher voltages and MΩ at lower voltages [21].

As a property of the PS/2 clock, as soon as the data is transmitted, the clock line



(a)



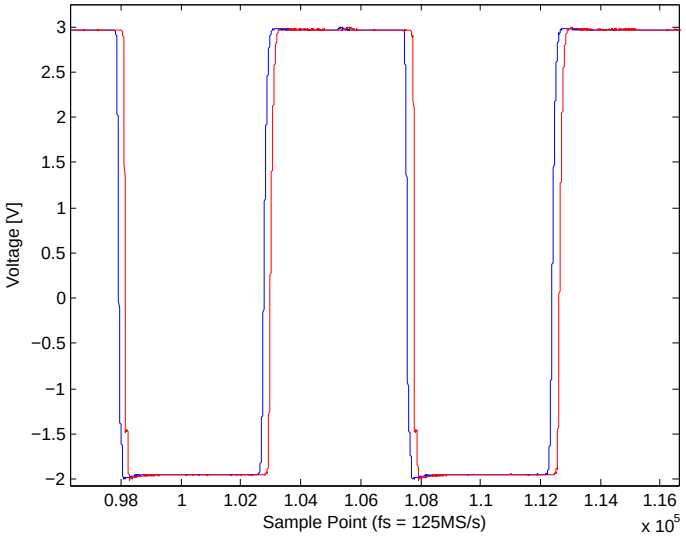
(b)

Fig. 2.2: (a) Electrical signal from the keyboard when the space bar is pressed (green: clock line; blue: data line; the clock is offset by 250 mV to aid visualization). Data is sampled by the host at the falling edge of the clock. (b) A passive HKL modeled in terms of its input capacitance C_{kl} and resistance R_{kl}. The HKL is connected in parallel with the PC (represented by the load R_{pc}) and keyboard (represented by the square-wave voltage source V_{kb}, with output resistance R_{kb}).

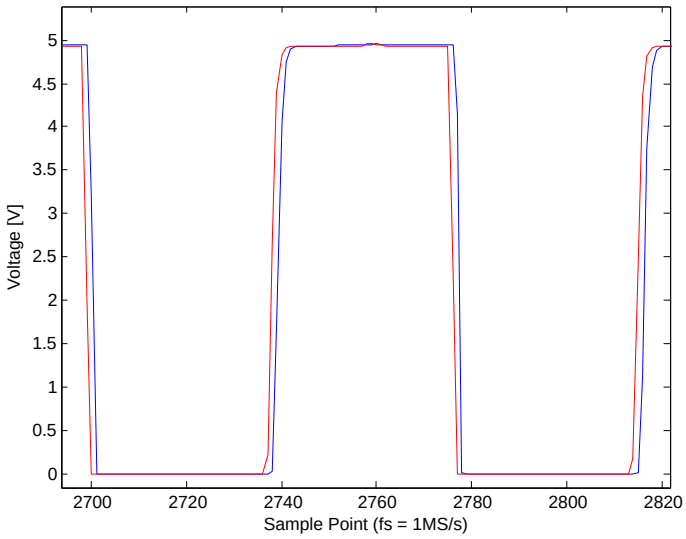
goes from +5 V DC (High) to 0 V DC (Low). Without the presence of HKL, the downward transition from high to low is a fast drop V_l(t). It is still a function of time due to the presence of very small capacitances in the PC (stray capacitances). In the presence of HKL the downward transition becomes a natural response of the RC circuit [20]. R_{kb}, being small as compared to R_{kl} and R_{pc} (as the keyboard is a voltage source of voltage V_{kb}), can be ignored for the moment and the line voltage will be $V_l(t) = 5\exp\left(\frac{-t}{\tau}\right)$ where $\tau = (R_{kl} \parallel R_{pc})\ C_{kl}$. For the rise time the capacitor discharges and the clock goes from low to high with the reversal of the polarity of V_{kb}.

The input impedance of the GPIO pins of the μC helped in detecting the HKL in two ways: (a) measuring the difference in voltage level of the clock signal (level-based detection approach), which was the primary detection technique, and (b) measuring the difference in the transient effects due to the small capacitance offered by the GPIO pins (transient effect-based detection approach), which was the secondary detection technique. As the capacitance of the capacitor was very small (on the order of pF) the resultant transient effects were small. When the transient effect-based detection was taken, the rise-times and the fall times were calculated according to a particular standard [22]. It can be seen from Figure 2.3 that there was a very small change in the rise time and fall time of the clock signal. The rise time and the fall time were less when connected without HKL (blue) than when connected with HKL (red). With the values of R_{kl} and R_{pc} on the order of MΩ and MΩ respectively and the value of capacitance on the order of pF, the time constant came out to be on the order of µs. Thus, the difference was on the order of µs, which was detectable.

Due to the small input capacitance of the GPIO pins, it was ignored while performing the level-based detection. The first two upper and lower clock level voltages were calculated. It can be seen from Figure 2.4 and Figure 2.5 that there was a definite difference on the order of mV for both the lower level (Figure 2.4) and the upper level (Figure 2.5) of the clock. The red color is the one with HKL and the blue color is the one without HKL. From the model discussed previously, the line voltage with (V_l) and without the HKL ($V_l^{'}$) was



(a)



(b)

Fig. 2.3: (a) The falling and rising portions of the first clock period of a keyboard's clock line for sampling rate at 125 MS/s and (b) 1 MS/s. The difference is clearly visible when the HKL (experiment done only using the 3 pF capacitor to see the transient effects) is present (red) compared to when it is absent (blue). Shown for two keyboards with an average of 100 records for each.

calculated.

In the absence of HKL, from Kirchhoff's current law it can be written that

$$\frac{V_l}{R_{pc}} = \frac{V_{kb} - V_l}{R_{kb}}.$$

Simplifying we get

$$V_l = \frac{R_{pc}}{R_{kb} + R_{pc}} V_{kb}. \tag{2.1}$$

$R_{eq} = R_{kl} \parallel R_{pc}$ is allowed with $V_l^{'}$ being the line voltage with HKL connected, which gives through Kirchhoff's current law

$$\frac{V_l^{'}}{R_{eq}} = \frac{V_{kb} - V_l^{'}}{R_{kb}}.$$

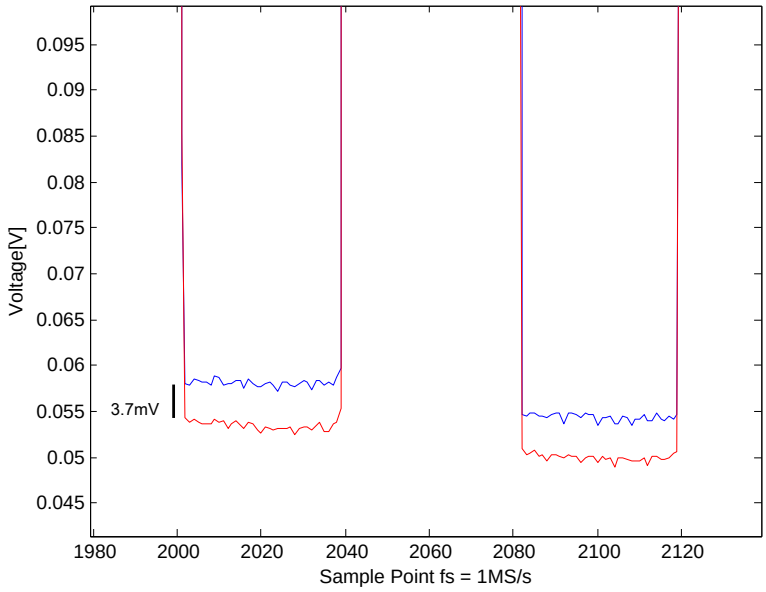
Simplifying, it can be written as

$$V_l^{'} = \frac{R_{eq}}{R_{kb} + R_{eq}} V_{kb}. \tag{2.2}$$

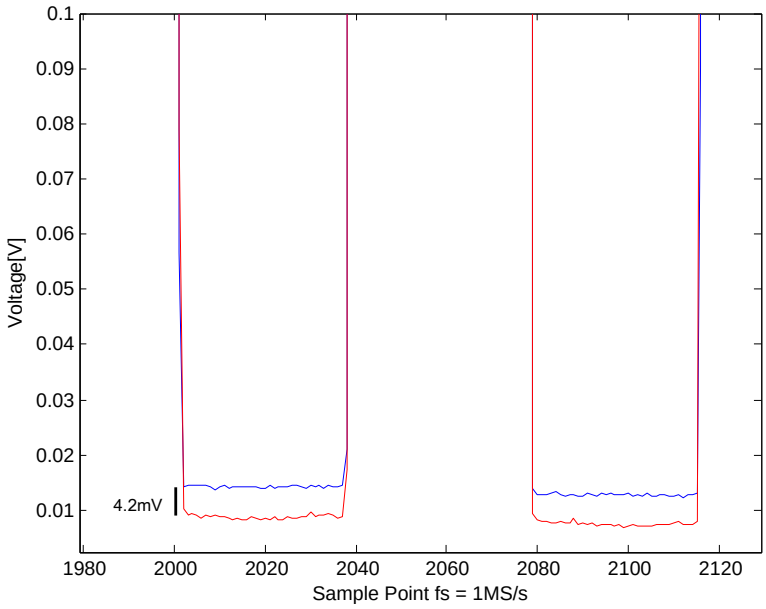
Equation (2.2) is less than or equal to Equation (2.1) when $R_{eq} \leq R_{pc}$.

These equations give the line voltages with the HKL connected ($V_l^{'}$) and without the HKL connected (V_l). The dependence of the line voltage on R_{kl}, which is the input resistance of HKL, suggests that a change in R_{kl} would definitely lead to a change in the line voltage, which fact is leveraged in performing the level-based detection approach. It can be seen that the value of R_{kl} is crucial because if it is too high then it does not have any effect on the difference in line voltage. Calculations were made taking R_{kl} to be as high as $10\,\mathrm{M\Omega}$ and taking R_{pc} to be as high as on the order of $\mathrm{M\Omega}$ (say $1\,\mathrm{M\Omega}$, being a load resistance). The keyboard being a voltage source will give a small resistance compared to R_{kl} and R_{pc} [20]. Taking R_{kb} as $500\,\Omega$ the drop in the line voltage when the HKL was present to when the HKL was absent with $V_{kb} = 5\,\mathrm{V}$ is $V_l - V_l^{'} \approx 250\,\mu\mathrm{V}$, which can be detected by the PLD (Physical-Layer Detection) methods discussed here.

Now, it is considered that there is an attacker who can evade this level-based detection

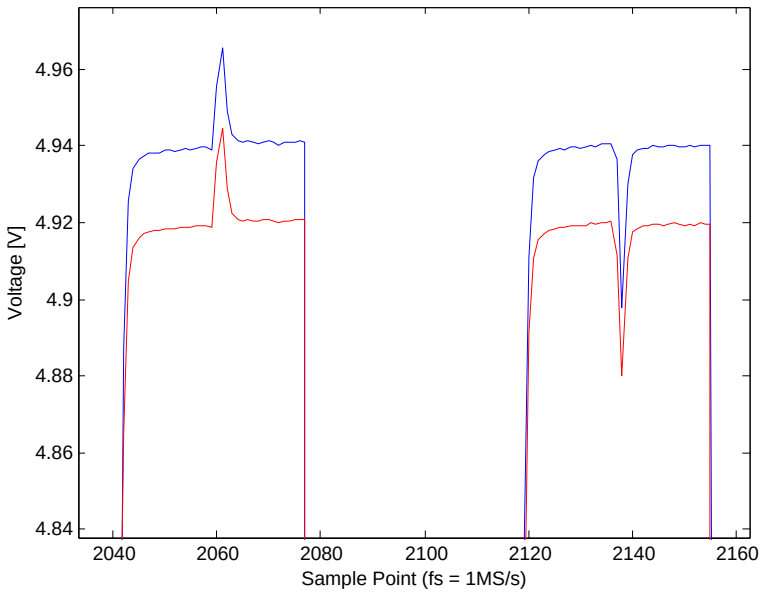


(a)

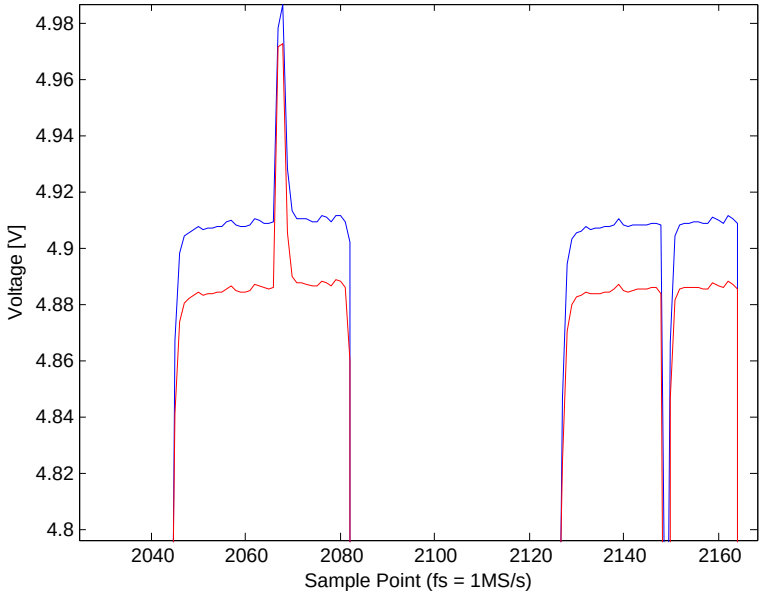


(b)

Fig. 2.4: (a) and (b) show the voltage of the clock line with (red) and without (blue) HKL (μC keylogger) for the lower portion of the first two clock periods. In each case the level is less during HKL's presence due to the loading effects of HKL. These are shown for two different keyboards with an average of 100 records for each.



(a)



(b)

Fig. 2.5: (a) and (b) show the voltage of the clock line with (red) and without (blue) HKL (μC keylogger) for the upper portion of the first two clock periods. In each case the level is less during HKL's presence due to the loading effects of HKL. These are shown for two different keyboards with an average of 100 records for each.

approach. It can be assumed that he/she puts a device with a very high input impedance compared to R_{pc} for which the voltage difference is too small to be measured. The value of R_{kl} (producing the equivalent resistance along with R_{pc}) should be selected by the attacker in such a way that the R_{eq} is large enough not to be resolved by the ADC. The derivation which follows will show the minimum value of R_{eq} for achieving this.

From Equation (2.2) it can be written that

$$R_{eq} = \frac{R_{kb}V_l^{'}}{V_{kb}-V_l^{'}}. \tag{2.3}$$

Adding and subtracting $V_l R_{kb}$ in the numerator and V_l in the denominator, in the above equation, it can be written that

$$R_{eq} = \frac{R_{kb}V_l - R_{kb}V_l + R_{kb}V_l^{'}}{V_{kb}-V_l^{'} + V_l - V_l}, \tag{2.4}$$

$$R_{eq} = \frac{R_{kb}(V_l-(V_l-V_l^{'}))}{V_{kb}-V_l + (V_l-V_l^{'})}, \tag{2.5}$$

$$R_{eq} = \frac{R_{kb}(V_l-r)}{V_{kb}-V_l + r}, \tag{2.6}$$

where r = $V_l - V_l^{'}$ which is the minimum resolvable voltage difference by the ADC which would be used.

From the above equations it is clear that the attacker should cleverly choose the value of R_{kl} while customizing the keylogger such that the drop across R_{eq} is large enough not to be resolved by the ADC in order to evade this level-based detection approach and small enough to minimize the time constant.

An interesting observation which can be made from the above two figures is that the drop in the line voltage due to the presence of the keylogger is different for different keyboards. This is because every keyboard has a different R_{kb}. The higher the value of R_{kb}, the larger the drop in voltage, which is also evident from Equations (2.1) and (2.2). The attacker knows that changes in R_{kl} will definitely be noticeable but the change in R_{kb} can-

not be restricted, as it depends on the keyboard used. This can open a possible area for research about how to change the value of R_{kb} with the introduction of an HKL such that one will not be able to detect whether the change in the keyboard resistance is due to the HKL or due to the change in keyboard type.

Chapter 3

Proposed Physical-Layer Detection Methodology

The previous chapter describes the mechanism for how the HKL affects the clock line. In this chapter an architecture is proposed which will be able to scan the signal and see where the display of signal distortion is greatest. The extraction routine extracts the so-called "features" (voltage, frequency, rise time, fall time, and slew rate) of the signal, leading to the comparison routine. More precisely a distribution of level and transient effects are considered for comparison. The distance-comparison metrics used here are sensitive to very small changes in the state of features (able to detect differences up to the order of ns and mV for transient-based and level-based approaches respectively).

3.1 Procedure

The procedure followed here to detect the keylogger with the help of the features obtained is depicted in Figure 3.1. The clock signal from the PS/2 keyboard was fingerprinted and the features were collected. The features used here are the voltage level of the clock signal for the level-based detection approach and rise time, fall time, and slew rate for the transient effect-based detection approach. A distribution of features, obtained from a number of keyboards, was formed. Two distributions were dealt with here, the training distribution and the test distribution. The test distribution was compared with a baseline distribution known as the training distribution, based on a threshold value that was kept between the training and the test distributions. The test distribution was from an unknown state of the system (it was not known whether or not the HKL is connected). When the test distribution was above the threshold and the training distribution was below the threshold, it was asserted that the test distribution came from a system other than that of the training scenario.

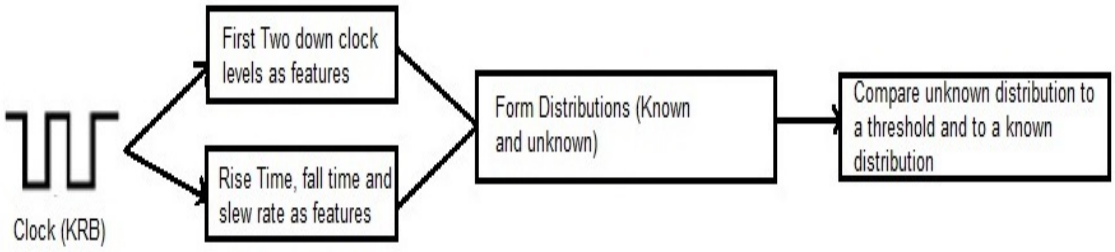


Fig. 3.1: The entire procedure is depicted. The clock signal is fingerprinted and the features are extracted. Both the level and transient effects are considered. Distributions are formed by taking features from a number of keyboards. These distributions are compared using distance-measuring metrics.

3.2 Architecture

Figure 3.2 shows the proposed physical-layer detection engine which was used. This can be placed anywhere between the PC and the keyboard. The figure shows that there were three blocks, represented by db, $f(\cdot)$ and $d(\cdot)$. Block $d(\cdot)$ was used for the comparison of features using distance-measuring algorithms discussed later in this chapter. $f(\cdot)$ extracted the features from the clock signal which were sampled using the ADC (sampler in the figure). The sampler was connected to the clock and ground because, as discussed in the previous chapter, the clock line did not change with every keystroke. "db" is the database where training and test data were stored. Two types of ADCs were used to obtain 1 MS/s and 125 MS/s sampling rates. The first ADC was necessary for measuring the drop in the line voltage and thus needed to have a higher resolution. The second ADC, which had a higher speed, was used in order to obtain a difference and measure the time constant on the order of ns. (The reader should refer to the data collection procedure in Chapter 4 for a detailed description on the ADCs used.)

3.3 Feature Extraction

From the previous chapter it is evident that the changes in the waveform due to the presence of HKL can be seen with the naked eye. From Figures 2.4 and 2.5 it is evident

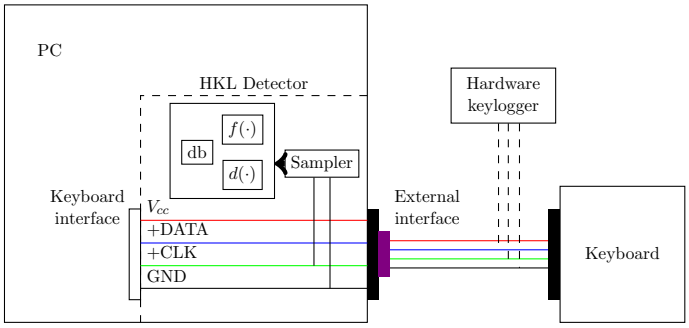


Fig. 3.2: The proposed architecture for detecting HKLs at the physical-layer. A sampler (ADC) measures the voltage of the clock line. When a key is pressed, the corresponding samples are processed to check if they match a baseline acquired in the absence of an HKL. The attachment of an HKL would change the line state by a detectable amount.

that the features which might be effective are the upper and lower clock level voltages for the down keypress.

It is also evident from Figure 2.3 that the rise times, fall times, and slew rates[1] can be taken as features so that a comparison can be made between the signal's transient analysis distribution.

After the sampling was done by the sampler (ADC), the records needed to be aligned with a particular reference signal. The reference signal was taken from the first record of each keyboard without the HKL attached, and the alignment was done by the correlation of the reference signal with a whole record, taking the keydown portion clock signal. The clock signal which was considered as a square wave [23] had the upper and lower levels when it was high (+5 V) and low (0 V) respectively. The next step was the extraction of features ($f(\cdot)$ in Figure 3.2). For extracting the upper and lower levels, the clock signal was traversed, and features were extracted from the lower and upper portions of the clock for the down keypress. When the clock signal became less than a certain voltage threshold value (shown in Figure 3.3), then the sample points corresponding to that region were taken as a lower clock signal, and when it was above a certain voltage threshold value, then the sample points corresponding to that region were taken as the upper clock signal. The lower clock level was the first downward clock signal and the upper clock level was the upward

[1]Slew rate being the rate of change of voltage with time, the rise time and fall time calculations can produce slew rates, i.e., transition of voltage up or down per unit of time.

clock signal which came immediately after the lower level. There were 11 such lower and upper clock levels (as can be seen from the figure), of which only 2 were taken for each record. The voltage threshold value was taken as 3 V (shown as green in the figure).

After the aligned keydown portions were obtained from the sampled signal, the transient responses were obtained. Rise times, fall times, and slew rates were also used as features for the whole down keypress clock signal. In order to extract these, the clock signal was again traversed, and these were obtained using a particular standard [22]. The rise times, fall times, and slew rates were obtained for all the clock levels corresponding to the down keypress (shown in Figure 3.3) and a subset of them (5 out of 11 clock levels) was taken in order to form the distribution and perform the analysis.

Two separate distributions were formed for both rise/fall times and upper/lower clock levels along with the third distribution of slew rate, which were used in the comparison function ($d(\cdot)$ in Figure 3.2).

3.4 Distribution-Formation and Comparison of Features Using Distance-Measuring Metrics

The distributions were formed by extracting the voltage, rise time, fall time, and slew rate for the down clock signal of 25 keyboards (details about experimental setup is given in Chapter 4). The clock signals were first aligned with a single observation (reference signal) from that particular keyboard considered. The reference signal was taken from the keyboard without the HKL connected. These distributions were then compared with the distance-measuring metrics. It was observed that the voltage of the clock signals, obtained from the same keyboards, change with time. Not only that, the changes in the line voltage and the transient effects were very small as discussed in Chapter 2. In fact, the changes were so small that the mere use of Euclidean distance calculation would not suffice, and the change in the line voltage over time would cause a considerable overlap in the distributions found while performing the comparison. These facts prompted the use of two distance metrics which were designed for comparing distributions, known as the Kullback–Leibler divergence (KLD) method [24, 25] and chi-squared distance measurement [26].

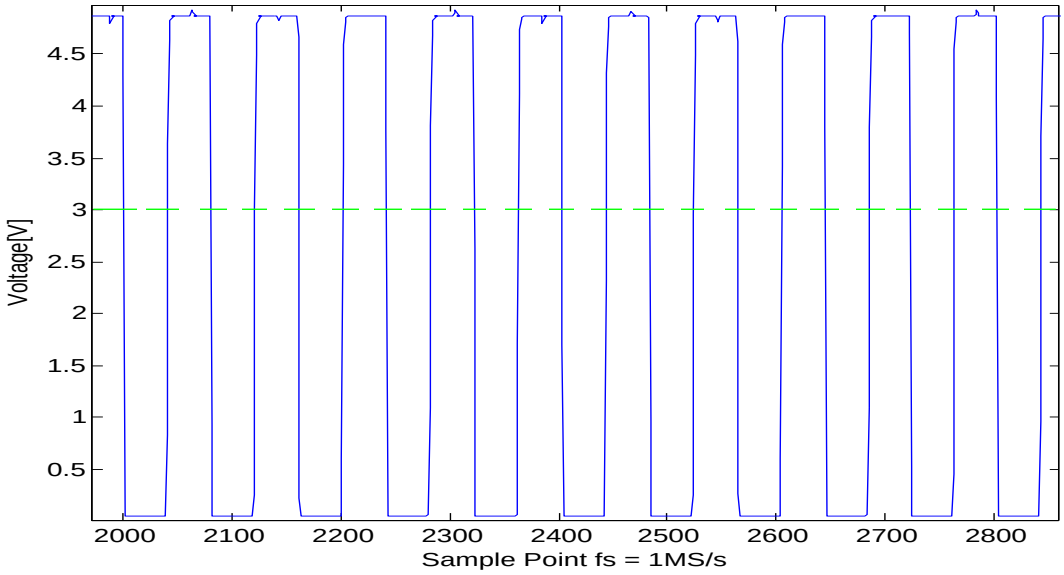


Fig. 3.3: These are the clock levels corresponding to the down keypress. Green line at 3 V shows the threshold value taken for selecting the upper and lower clock levels. Only 2 of these 11 clock levels were taken in order to form the distributions. For the transient effects, 5 out of 11 rise time, fall time, and slew rates were chosen.

3.4.1 Kullback–Leibler Divergence

As discussed before, two distributions were dealt with here, training and test distribution. The general distance between the training distribution and the test distribution was not the same as that of the distance between the test distribution and the training distribution according to KL divergence. This means that the KL divergence is a nonsymmetric measure of distance between two distributions. Let the distributions be P and Q. These two distributions are from the data with the HKL and without the HKL respectively. The distributions here, as discussed before, were the rise times, fall times, slew rates, and the lower level and upper level of the clock. These distributions in an unknown state were compared with a baseline built for the keyboard in a known state (no HKL present).

From the definition, divergence of the distribution with the keylogger from the distribution without the keylogger is the measure of the information lost when the first distribution is used to make an approximation to the second distribution. Also, it can be defined as

the number of extra bits required to approximate from P to Q. It can be asserted that an HKL is attached, for the system from where the test distribution comes, when the number of extra bits required to approximate from P to Q, or the information lost when the test distribution used make an approximation to the training distribution is too high, which means that the test distribution is greater than a particular threshold. This threshold value (T), chosen during the training distribution, is set in such a way that there is an acceptable number of false positives. It can be asserted that no HKL is attached to the system when the test distribution is less than the threshold.

The training data T_r was built from the features of a certain number of records obtained in the absence of HKL. A test distribution P was then constructed from records collected from one or more keystrokes. To test for the presence of HKL, KL Divergence was employed: if d(P, T_r) $< T$, then the records from the distribution P were said to be acquired in the absence of HKL. If d(P, T_r) $> T$, then the test distribution P came from a system with HKL attached. The Kullback–Leibler Divergence [27] is defined as

$$D_{KL}(P\|Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right) P(i), \tag{3.1}$$

where P and Q are the two distributions with the divergence here defined as from Q to P. The value of i in Equation (3.1) would go from 1 to the number of observations taken for testing.

3.4.2 Chi-Squared Distance Measurement

The second distance measurement metric between distributions is the chi-squared distance measurement. Also known as the chi-squared distance between histograms[2] [26], this is the symmetric measurement of distance between the training distribution and the test distribution. With the same assumptions as before (baseline distribution being the training distribution), the distribution (P, as discussed in case of KLD) was subjected to an unknown state, and difference in the distributions was noted.

[2]The histograms need not be normalized in this case.

Considering the approach taken here, the upper and lower clock levels along with the transient effects were taken as distributions without the HKL connected for building a training distribution. In the same way, a test distribution was constructed from the unknown state of the system to test the presence of HKL. Now, the chi-squared distance was obtained from the following definition [26]

$$\chi^2 = \frac{1}{2}\sum_i \frac{(P(i)-Q(i))^2}{P(i)+Q(i)}, \tag{3.2}$$

where P and Q are the two distributions between which the distance is calculated here. In this equation (3.2), i ranges from 1 to the number of records taken for testing (number of bins in case of the histograms). This includes, in all cases, all 1000 records acquired in the data collection process. As discussed before, a threshold (T) was decided and any value of chi-squared distance for any record above that would assert that the test records were obtained from a system with an HKL attached to it.

Chapter 4

Experimental Setup and Results for the Proposed Distance Metrics

In this chapter, the experimental setup and the design of the keylogger used are discussed. The error rate results obtained from the distance-measuring metrics are presented and the process followed in order to make the evasive keyloggers detectable (also discussed in Chapter 2) is discussed here with the corresponding results and plots.

4.1 Keylogger Design

The keylogger used in the experiment was a microcontroller-based HKL. It was built using a Texas Instruments (TI) Tiva C Series TM4C123G LaunchPad evaluation kit, which is based on the TI TM4C123GH6PM microcontroller [28]. It was powered by a separate USB bus so that no power was extracted from the PS/2 bus. The PS/2 clock was hooked to one of the pins, configured as input, for the μC. When the clock went from high to low, an interrupt service routine was called where the HKL activity took place (all inside the μC). One of the LEDs on the μC was blinked as soon as the ISR was called, to make sure that the negative edge of the clock was correctly read by the keylogger (μC) as the data was sampled by the PC at the negative edge of the clock.

4.2 Data Collection

Along with the voltage of the PS/2 clock line, the temperatures of the main components were also obtained. The temperature data collection is discussed in the next chapter. Certain issues (questions) are part of the problem regarding data collection and should be dealt with. These problems are stated and a workaround for them in the process of data collection are also given.

- As different keyboards are used, the relationship of the HKL might be different with each keyboard. *Solution*: Data was taken from all devices, 25 keyboards (though all types of keyboards available on the market were not tested).

- It was proposed here that the behavior of each keyboard should be different with HKL and without HKL. *Solution*: Data was taken for both cases (with and without HKL).

4.2.1 Evasive Keyloggers

Special keyloggers can be built specifically to evade the level-based detection approach which was proposed here and is discussed in Chapter 2. This can be done by an attacker by placing a high impedance device (e.g., LT1793 op-amp with an input impedance of 1 TΩ [29]) in place of the μC. This avoids the loading effect due to the high impedance and does not cause any difference in voltage. While there is no change in the voltage level, one can still detect the keylogger by measuring the difference in the transient effects as there is a capacitance corresponding to the input of the comparator (or the high-impedance device considered). The attacker needs to keep in mind that the input capacitance for the op-amp is around 1.5 pF [29], and this device, when attached to the clock line, will give a capacitance of more than 1.5 pF. In order to justify this experimentally, a capacitor with 3 pF capacitance was put in place of the μC for experimenting and the transient effects were measured.

4.2.2 Data Collection without Temperature

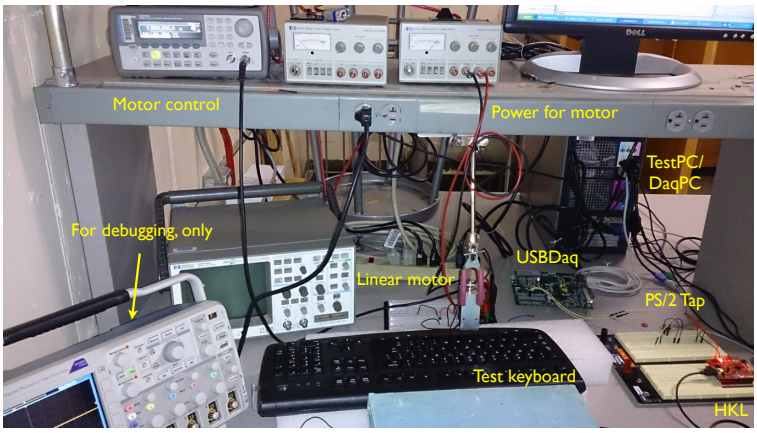
Figure 4.1 has two figures showing the experimental setup without considering the temperature of the surroundings. The PC used here was a Dell Optiplex GX620 with a USB keyboard connected to it. As it did not have a PS/2 port, a USB-to-PS/2 converter was used to connect the test keyboards. This ensured that the controlling USB keyboard did not affect the test keyboards. A linear motor was used to press the space bar in an automated manner for 0.3 s every 1.2 s with the help of a function generator (AWG-Arbitrary Waveform Generator). The function generator along with the MOSFET used to make the

motor work had a 20% duty cycle square wave with a period of 1.5 s. Thus the motor was on within that square wave period and then went off and so on.

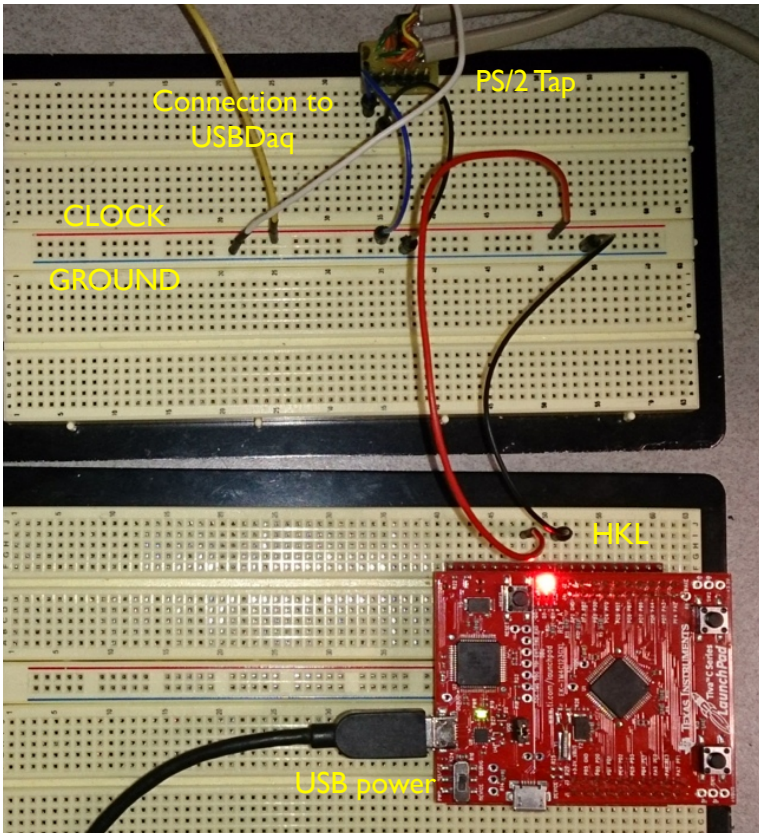
The ADC was one of the main components of the setup. Figure 4.1(a) shows the ADC and Figure 4.1(b) shows how the connection was made from the tap in the PC-keyboard connection to the ADC. Two ADCs were used for examining two feature sets: level difference and transient effect difference. A Measurement Computing (MCC) USB2500 Series DAQ board configured to use at a full scale voltage of 10 V was used to obtain a 1MS/s sampling rate. This was a 16-bit ADC and as a result the minimum voltage resolution was $\approx 153\,\mu$V. The oscilloscope shown in Figure 4.1(a) was used as the other ADC, connected in the same way. It was a Tektronix DPO2024 oscilloscope equipped with a Tektronix P6139B probe (10 MΩ and 8 pF input impedance and capacitance, respectively), which was used at a sampling rate of 125 MS/s. The oscilloscope acting as an 8-bit ADC was configured to use a full-scale voltage of 10 V. An offset of 2 V was used, which made the high voltage measured by the ADC to be 3 V. Thus it was able to measure signals with a resolution of ≈ 39 mV.

Whenever the space bar was pressed by the motor, the μC, acting as the HKL, took in the clock signal and the ADC sampled the input. For the level-based detection, an MCC USB2500 series DAQ board was used for sampling. Each keypress, both up and down keystrokes, was collected for the MCC DAQ at every interrupt for the μC, collecting 350 000 samples altogether. For the oscilloscope, 125 000 samples were obtained during the same time and resulted in collection of only the down clock signal for each keypress.

Data was collected from a total of 25 keyboards from two manufacturers (Dell and Logitech) and two different places of manufacture (China and Thailand). For each keyboard, 1000 keystrokes were recorded without the keylogger attached, followed by another 1000 keystrokes with the keylogger attached. The same procedure was followed with the 3 pF capacitor where the 125 MS/s sampler (Tektronix oscilloscope) took approximately an hour per keyboard to acquire both sets of data and approximately 50 minutes per keyboard for the 1 MS/s sampler to acquire both sets of data.



(a)



(b)

Fig. 4.1: (a) Experimental setup used. The keyboards were secured in a place that the linear motor struck approximately the same place on the space bar for each keyboard. (b) Closeup of the tap connections. The HKL designed is shown in the lower right corner just after observing a keystroke. This shows that the HKL is self-powered and passively taps the clock line.

4.3 Discussion and Results

It was seen in Chapter 2 that the rise time and the fall time for the 3 pF capacitor connected (analogous to HKL connected) is more than when it is not connected. From Figure 2.3(a), it was calculated that the rise time for an average of 100 records is 11.36×10^{-7}s for the first down clock signal without HKL connection (capacitor in this case) and 11.53×10^{-7}s with the HKL connected. Thus there is a difference on the order of ns (specifically 0.017 µs). This increase in the fall and rise time is attributed to the presence of an extra 3 pF capacitor in addition to any other stray capacitors which might be present in the circuit (inherent to the circuitry). As discussed before, the μC was removed and a 3 pF capacitance capacitor was placed for obtaining these readings. Thus here only the capacitance C_{kl} was accounted for rather than both the resistance (R_{kl}) and capacitance. From Figure 2.2(b) it is noted that the time constant without R_{kl} would be $\tau = C_{kl} \times R_{pc}$. At the same time the stray capacitance due to R_{pc} would dictate the time constant when the 3 pF capacitor was not connected. Let us consider the stray capacitance due to R_{pc} to be 2 pF and R_{pc} to be 1 MΩ. In this case the time constant was 2 µs. Now, when the 3 pF capacitor was connected, the equivalent capacitance with the stray capacitance became 5 pF and with the same value of R_{pc} the time constant was calculated to be 5 µs. It can be seen that there was a difference in the time constants on the order of µs (specifically 3 µs). Thus, in turn, the difference in rise time, fall time, and slew rates were on the order of µs. The difference in the actual value and the calculated value of the rise time is due to the average taken of 100 records. The average of 1000 records was taken to compensate for this difference.

Now, when the μC was used as HKL, the average difference of the line voltage in the absence and presence of the HKL (i.e., V_l–$V_l^{'}$) was found to be 4.7 mV for the lower level of the clock and 24 mV for the upper level of the clock. Though differences in the upper clock line have also been calculated, stress was given on the lower line voltage difference due to the fact that the clock went down only when a key was pressed and data was transmitted. The upper portion of the clock was the default position (PS/2 clock remains high during the idle

state). Also, the leakage current remains constant for a greater range of voltage [0,3.3]V for GPIO ports [21]. Thus, examining the lower clock level should give more accurate results than the upper clock levels.

4.3.1 Passive Keyloggers

To detect the differences in the upper and lower clock levels and also in the transient effects obtained from the feature extraction procedure discussed in Chapter 3, distances between the distributions, obtained with the HKL attached and without the HKL attached, were calculated using KLD and chi-squared. A training distribution was used which varied from 1 through 25 records for each keyboard and the test distribution consisted of the rest of records. Out of 350 000 samples, the number of sample points used for these calculations was 280 and 300 for the upper level and the lower clock levels respectively. The whole upper portion of the clock for the first transition fell within 280 sample points and the lower portion for the first transition within 300 points. No more than one transition was taken because in that way the calculation will be tractable.

The *equal error rate (EER)* [30] is reported for measuring the performance of the detection method. EER is the rate at which both the false negatives (falsely detect the absence of HKL) and false positives (falsely detect the presence of HKL) are equal. Tables 4.1 and 4.2 report the average, maximum, and median (the minimum was always zero) EER for training distributions built from $N = 1, 2, 4, 5, 10, 20, 25$ consecutive records (training/test distributions built from the lower level in Table 4.2 and the upper level in Table 4.1) for the KLD measurement and Tables 4.3 and 4.4 for the chi-squared measurement. From the previous discussions it is obtained that there is a higher drop in the upper level of the clock and thus it could seen from the table that it is easier to detect the HKL there. This is true for both the KLD and chi-squared approaches as seen from Tables 4.1, 4.2, 4.3, and 4.4. For KLD and with the small differences in the lower level of the clock, the HKL was detected after 4 keystrokes and for the upper level of the clock after 2 keystrokes. Meanwhile in case of chi-squared for both the lower and upper level of the clock, it is seen that 25 keystrokes were necessary to detect the HKL. In case of the tables, detection means that the test

distribution built from the records captured was greater than the specified thresholds.In all the tables, sample points is the number of sample points used in that particular distance calculation metric.

Figure 4.2 shows the distances between training and test distributions (Figure 4.2(a) shows the KLD measurement and 4.2(b) shows the chi-squared measurement), and $N = 25$ for the lower and upper clock level comparisons respectively, along with their respective EER thresholds. In Figure 4.2(b) for keyboard 13 from the left, a large amount of overlap can be seen, which means that data is corrupted and the distance metric could not calculate the distance from its clock signal. For all cases, test records are all the 1000 records from the unknown state of data.

In order to guarantee equal false positives and false negatives, the threshold was carefully chosen. From the distance-measurement figures and the tables it can be the seen that the threshold should be above the training distribution such that there are minimum or no false negatives. Thus the HKL was detected for all the cases when the threshold used fell below the test distributions (shown in red in all the figures). For the upper and lower levels of the clock it was possible to lower the number of keystrokes, needed to detect the keylogger, to four.

4.3.2 Passive and Evasive Keyloggers

Previously in this chapter and also in Chapter 2 it is pointed out that the HKL can be detected even with rise and fall time differences. Though they are very small, it is shown in this section that PLD methods have been successful in detecting the presence of HKL for keyboards. These calculations have been done in order to show that even a specially-built keylogger which is customized to evade level-based detection cannot do that without increasing the time-constant, in turn increasing the rise time, fall time, and slew rate. In order to validate this claim, the following distance calculations are shown.

Tables 4.5, 4.6 and 4.7 report the average, maximum, and median/minimum (minimum was not always zero in this case, stated later) EER for training distributions built from $N = 1, 2, 4, 5, 10, 20, 25$ consecutive records (training/test distributions built from the rise

Table 4.1: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 25 records). The table gives results for distributions built using the upper clock level. The presence of HKL was reliably detected, for all 22 keyboards, after 4 keystrokes by observing the upper level.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	0.5927	9.87	0.095	0.6454	0.8	0.5	300
2	0.1661	1.46	0.0300	0.0225	0.032	0.016	300
4	0.0295	0.19	0	0.0433	0.06	0.02	300
5	0.0122	0.19	0	0.0655	0.08	0.06	300
10	0	0	0	0.15	0.2	0.1	300
20	0	0	0	0.15	0.2	0.1	300
25	0	0	0	0.25	0.25	0.25	300

Table 4.2: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 25 records). The table gives results for distributions built using the lower clock level. The presence of the HKL was reliably detected, for all 22 keyboards, after 4 keystrokes by observing the lower level.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	9.6291	92.5	1.45	0.9583	1.05	0.7	280
2	0.4791	3.2000	0.2000	0.0242	0.0265	0.0240	280
4	0.0208	0.2	0	0.034	0.04	0.025	280
5	0.0041	0.1	0	0.0508	0.0600	0.0500	280
10	0	0	0	0.1	0.1	0.1	280
20	0	0	0	0.2	0.2	0.2	280
25	0	0	0	0.4	0.4	0.4	280

Table 4.3: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 25 records). The table gives results for distributions built using the upper clock level. The presence of HKL was reliably detected, for all 22 keyboards, after 5 keystrokes by observing the upper level.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	2.4227	9.99	0.5450	0.0011	0.002	0.0005	300
2	0.9205	4.94	0.2900	0.0009	0.0017	0.0003	300
4	0.2018	2.42	0.0250	0.0008	0.0012	0.0006	300
5	0.1039	1.92	0	0.0006	0.0008	0.0005	300
10	0.0095	0.18	0	0.0009	0.0013	0.0008	300
20	0.0009	0.02	0	0.0009	0.0012	0.0007	300
25	0	0	0	0.0009	0.0346	0.0008	300

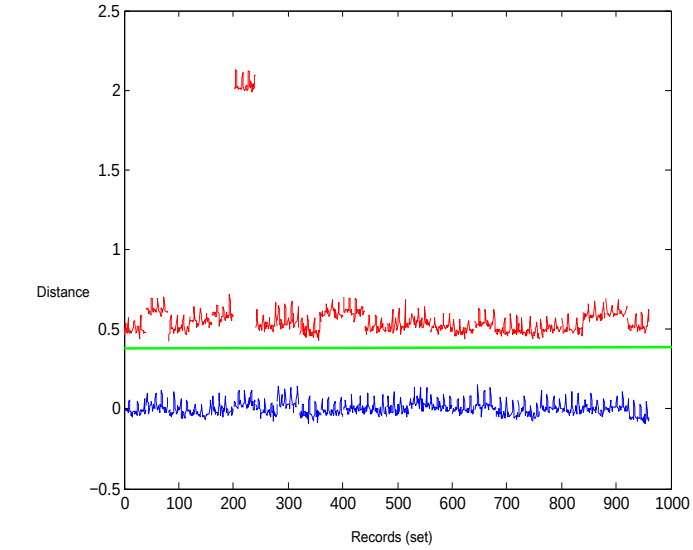
Table 4.4: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 25 records). The table gives results for distributions built using the lower clock level. The presence of the HKL was reliably detected, for all 22 keyboards, after 5 keystrokes by observing the lower level.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	5.3245	5.21	1.4	0.0012	0.0021	0.0004	280
2	0.4891	2.34	0.02	0.0009	0.0017	0.0002	280
4	0.0124	0.32	0.01	0.0009	0.0013	0.0006	280
5	0.0041	0.25	0	0.0005	0.0008	0.0001	280
10	0.0076	0.2	0	0.0005	0.0009	0.0002	280
20	0.0006	0.01	0	0.0006	0.0012	0.0002	280
25	0	0	0	0.0006	0.0009	0.0004	280

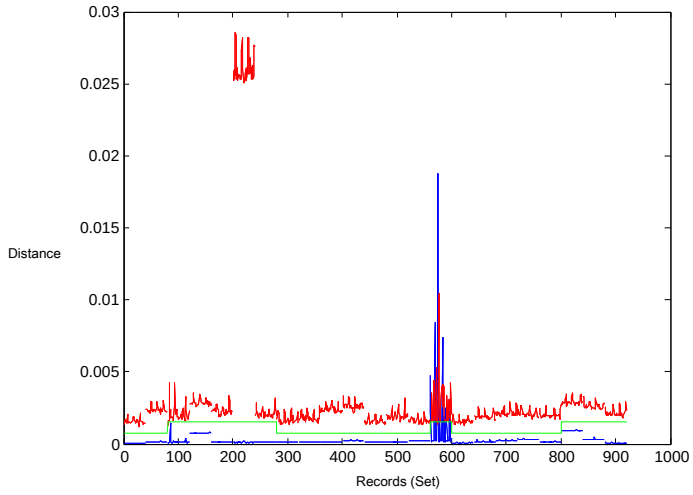
times, fall times, and slew rates) for KLD measurement. Tables 4.8, 4.9, and 4.10 report the average, maximum, and median/minimum EER for chi-squared measurement. As it was difficult to discern a difference in rise times when the 1 MS/s sample rate was used (details in Chapter 2), the data obtained with the sampling rate of 125 MS/s is shown here. Previous discussions showed that the difference between the rise time, fall time, and slew rate are small, and thus it could be seen from the tables that it took more keystrokes to detect the HKL in these cases. In case of rise time it is easier to detect the HKL (50 keystrokes for both KLD and chi-squared) than for fall time or slew rate. Here, out of 125000 samples, 7500 sample points have been used, so that the transient effects for all the clock levels could be considered.

The minimum values of EER for KLD using rise time distribution were 0.02, 0.03, and 0.01 for 1, 2, and 4 keystrokes, respectively. The minimum value of EER for KLD using fall time distribution was 0. The minimum value of EER for KLD using slew rate distribution were 0.18, 0.67, 0.08, and 0.01 for 1, 2, 4, and 5 keystrokes respectively. The minimum value of EER for chi-squared using rise time distribution were 0.02, 0.03, and 0.01 for 1, 2, and 4 keystrokes respectively. The minimum value of EER for chi-squared using fall time distribution were 0.08, 0.07, and 0.05 for 1, 2, and 4 keystrokes respectively. The minimum value of EER for chi-squared using slew rate distribution were 0.01, 0.18, 0.02, 0.37, 0.01, 0.16, and 0.02 for 1, 2, 4, 5, 10, 20, and 25, keystrokes, respectively. Figures 4.3(a) and 4.3(c) show the KLD distances and Figure 4.3(b) show the chi-squared distance between training and test distributions, and $N = 25$ for the rise time, fall time, and slew rate comparisons, along with their respective EER thresholds.

It is important to note here that the distance values with HKL are more than those without HKL because the rise time, fall time, and slew rate were more when the capacitor was connected than when it was not. The false negatives were lower in this case than the false positives because the KLD deals with the logarithm of the distributions of the rise time. As the logarithm of a smaller decimal value leads to a large number, it can be seen that the false negatives were at a lower distance from the false positives. The chi-squared distance



(a)



(b)

Fig. 4.2: (a) The KLD between the training and a test distribution built from records without a keylogger attached (blue) and with a keylogger attached (red) for 22 (x-axis; records are grouped) keyboards. EER threshold is shown in green. Since there is no overlap, HKL can be detected. Features were extracted from the lower level of clock with $N = 25$ and (b) shows the same thing for chi-squared calculation with features extracted from the upper level of the clock with $N = 25$.

measurement does not deal with logarithm, so the false positives are lower than the false negatives. However, if an absolute value of the distance is taken, then the false negatives go up and the false positives would remain where they are for the KLD. Figures showing large amount of overlap between the distances are probably corrupted data. Tables 4.8, 4.9, and 4.10 report the same metrics for the chi-squared calculation. It took a minimum of 50 (100 in some cases) keystrokes to assert that an HKL is attached to the system. For all cases, the test records are all 1000 records from the known and unknown states of data.

From the tables, it can be seen that for some of the transient effects, viz. the slew rate for KLD and chi-squared calculation, and the rise time and fall time for chi-squared calculation, it was not possible to detect the keylogger even with 50 keystrokes. It took 100 keystrokes to detect them. Using a sampling-rate of 125 MS/s with 50 keystrokes, a median EER of 0 was obtained for the KLD calculation in case of the rise time and fall time, and with 100 keystrokes, a median EER of 0 was obtained for the chi-squared measurement in case of the rise time and fall time. Thus, following the previous discussions, HKL was detected for all the cases when the threshold used fell above the test distribution, giving almost zero false negatives.

Table 4.5: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the rise time for 5 clock levels. The presence of HKL was reliably detected, for all 25 keyboards, after 50 keystrokes by observing the rise time.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	2.0013	8.87	3	-2.5×10^{-9}	-2.5×10^{-9}	-5×10^{-9}	7500
2	1.8696	4.81	1.5	-1.5×10^{-8}	-1.5×10^{-8}	-8×10^{-9}	7500
4	1.0056	2.48	0.73	-2.12×10^{-8}	1×10^{-8}	-5×10^{-8}	7500
5	0.4174	1.52	0.21	-3.36×10^{-8}	-2.43×10^{-8}	-4.43×10^{-8}	7500
10	0.1492	0.96	0.03	-4.56×10^{-8}	-2×10^{-8}	-6×10^{-8}	7500
20	0.0954	0.54	0.02	-3.59×10^{-7}	-2.3×10^{-7}	-5×10^{-8}	7500
25	0.0244	0.18	0.01	-1.59×10^{-7}	-1.497×10^{-7}	-9×10^{-8}	7500
50	0.0187	0.02	0	-2.56×10^{-7}	-2×10^{-7}	-6×10^{-7}	7500
100	0.01	0.1	0	-4.48×10^{-7}	-1×10^{-7}	-7×10^{-7}	7500

Table 4.6: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the fall times of 5 clock levels. The presence of the HKL was reliably detected, for all 25 keyboards, after 50 keystrokes by observing the fall time.

N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	12.6291	11.55	9.51	-1.5×10^{-9}	1×10^{-9}	-3×10^{-9}	7500
2	3.0132	6.05	3.59	-1.8×10^{-9}	2×10^{-9}	-4×10^{-9}	7500
4	0.8748	2.48	0.71	-3.18×10^{-9}	-1.5×10^{-9}	-5.5×10^{-9}	7500
5	0.5940	1.75	0.51	-2.52×10^{-9}	-1×10^{-9}	-4×10^{-9}	7500
10	0.2750	0.92	0.2250	-3.52×10^{-9}	-1×10^{-9}	-5.5×10^{-9}	7500
20	0.1005	0.52	0.01	-4.56×10^{-8}	-1.4×10^{-8}	-5.6×10^{-8}	7500
25	0.0772	0.38	0.02	-1.56×10^{-8}	1×10^{-8}	-2.6×10^{-8}	7500
50	0.0632	0.21	0	-5.46×10^{-8}	-1×10^{-8}	-9×10^{-8}	7500
100	0.0112	0.1	0	-6.48×10^{-8}	-5×10^{-8}	-8×10^{-8}	7500

Table 4.7: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the slew rate for each clock level. The presence of the HKL was reliably detected, for all 25 keyboards, after 100 keystrokes by observing the slew rate.

N	EER(%)			T			sample points
	mean	max	min	mean	max	min	
1	5.6291	9.67	0.18	-6.28×10^4	5×10^4	-16×10^4	7500
2	2.5624	4.98	0.67	-5.48×10^4	7×10^4	-17×10^4	7500
4	1.9708	2.5	0.08	-0.66×10^5	1.5×10^5	-3×10^5	7500
5	1.2644	2	0.01	-2.2×10^5	3.6×10^5	-6.4×10^5	7500
10	0.4240	0.98	0	-3.6×10^5	2×10^5	-6×10^5	7500
20	0.2392	0.54	0	-5×10^5	-4×10^5	-7×10^5	7500
25	0.1204	0.33	0	-6.76×10^5	-2×10^5	-16×10^5	7500
50	0.0332	0.19	0	-2.048×10^6	-0.8×10^6	-3.2×10^6	7500
100	0.0096	0.1	0	-4.15×10^5	-3×10^5	-4.8×10^5	7500

Table 4.8: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the rise time for 5 clock levels. The presence of HKL was reliably detected, for all 25 keyboards, after 50 keystrokes by observing the Rise time.

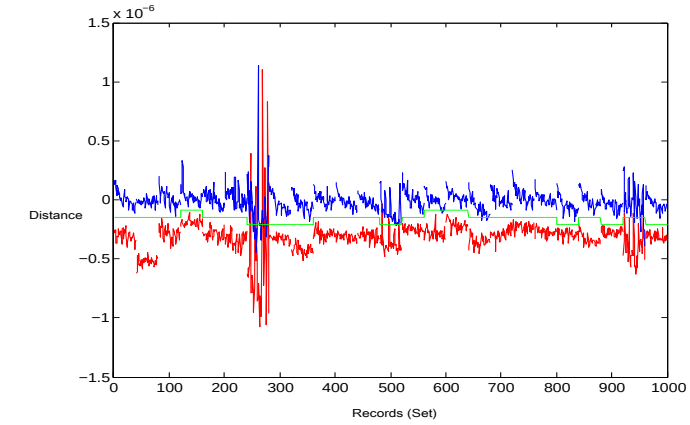
N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	2.1321	8.97	2.5	2.5×10^{-10}	4.5×10^{-10}	1.5×10^{-10}	7500
2	1.8696	2.7	2	5.5×10^{-10}	8×10^{-10}	2×10^{-10}	7500
4	1.4352	2.5	1.51	2.12×10^{-10}	4×10^{-10}	1×10^{-10}	7500
5	1.0932	2	1.15	2.24×10^{-10}	3×10^{-10}	1.5×10^{-10}	7500
10	0.6116	1	0.75	2.06×10^{-10}	3.5×10^{-10}	0.5×10^{-10}	7500
20	0.2323	0.54	0.52	3.59×10^{-10}	2.2×10^{-10}	5.1×10^{-10}	7500
25	0.1632	0.4	0.13	2.0320×10^{-10}	2.96×10^{-10}	0.56×10^{-10}	7500
50	0.0613	0.2	0.01	2.055×10^{-10}	2.72×10^{-10}	1.12×10^{-10}	7500
100	0.01621	0.1	0	4.48×10^{-10}	1×10^{-10}	7×10^{-10}	7500

Table 4.9: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the fall times of 5 clock levels. The presence of the HKL was reliably detected, for all 25 keyboards, after 50 keystrokes by observing the fall time.

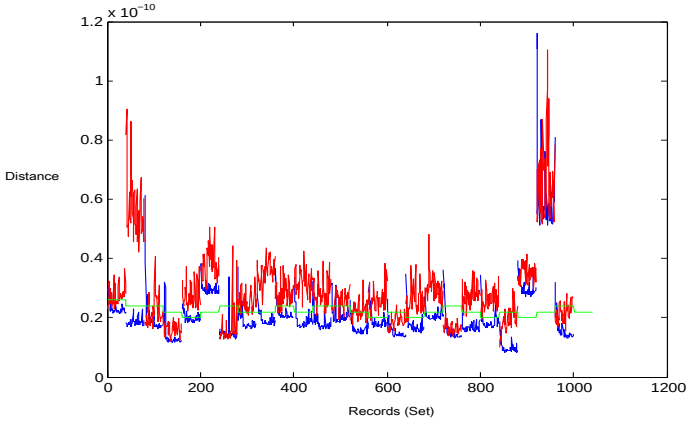
N	EER(%)			T			sample points
	mean	max	median	mean	max	min	
1	9.6291	9.55	8.51	1.5×10^{-11}	1×10^{-11}	3×10^{-11}	7500
2	3.0121	3.05	2.01	2.89×10^{-11}	4×10^{-11}	1.2×10^{-11}	7500
4	1.5484	2.89	1.57	2.32×10^{-11}	3×10^{-11}	1.5×10^{-11}	7500
5	1.2840	2	1.31	-2.02×10^{-11}	2.7×10^{-11}	1.2×10^{-11}	7500
10	0.5640	1	0.63	-2.268×10^{-11}	2.7×10^{-11}	1.8×10^{-11}	7500
20	0.1005	0.52	0.01	-4.56×10^{-8}	-1.4×10^{-8}	-5.6×10^{-8}	7500
25	0.1736	0.4	0.1	2.22×10^{-11}	2.4×10^{-11}	2×10^{-11}	7500
50	0.1236	0.2	0.14	2.27×10^{-11}	2.8×10^{-11}	1.6×10^{-11}	7500
100	0.0563	0.1	0	-2.44×10^{-11}	3.3×10^{-11}	1.8×10^{-11}	7500

Table 4.10: The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution varied from 1 through 100 records). The table gives results for distributions built using the slew rate for each clock level. The presence of the HKL was reliably detected, for all 25 keyboards, after 100 keystrokes by observing the slew rate.

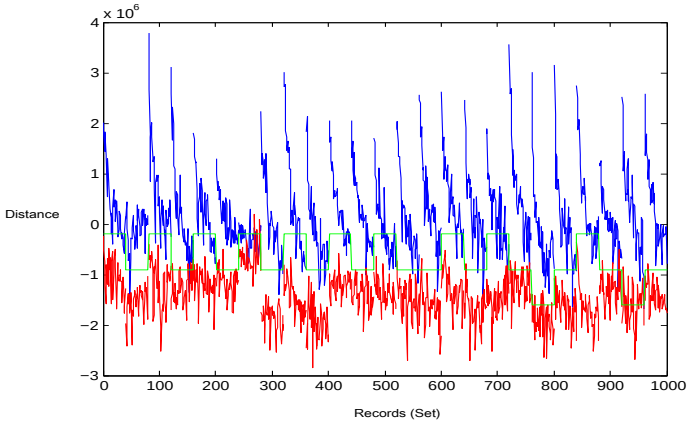
N	EER(%)			T			sample points
	mean	max	min	mean	max	min	
1	5.2291	9.68	0.01	-6.38×10^3	5×10.5^3	-16.5×10^3	7500
2	2.6524	4.98	0.18	-5.48×10^3	7×10^3	-17×10^3	7500
4	1.7908	2.5	0.02	-4.64×10^3	2×10^3	-8×10^3	7500
5	1.6080	2	0.37	-4×10^3	0	-8×10^3	7500
10	0.8064	1	0.01	-5.7×10^3	-2×10^3	-10×10^3	7500
20	0.2392	0.54	0.16	-5×10^5	-4×10^5	-7×10^5	7500
25	0.2248	0.4	0.02	-5582	-3230	-7430	7500
50	0.13	0.2	0	-5.86×10^3	-2×10^3	-10×10^3	7500
100	0.0624	0.1	0	-6.16×10^3	-3.5×10^3	-8.5×10^3	7500



(a)



(b)



(c)

Fig. 4.3: (a) The KLD between the training and a test distribution built from records without a keylogger attached (blue) and with a keylogger attached (red) for 25 (x-axis; records are grouped) keyboards. EER threshold is shown in green. Since there is separation, HKL can be detected for most of the keyboards. Features were extracted from the rise times of the clock and (b) shows the same thing with the features extracted from the fall times of the clock, and (c) shows the same thing with the features extracted from the slew rate of the clock.

Chapter 5

Stability of Features

The variability in the line voltage apparent in Figures 4.2 and 4.3 suggests that the line voltage is a random process. This leads to an important question about whether it is possible to identify the presence of HKL using data gathered over time. In order to answer this question, a round of data were run without the keylogger attached, and the training data from the previous data run were used to calculate the distance between the two. Figure 5.1 shows that for the lower level of the clock, the distance between the two distributions is negligible and there is maximum overlap, which suggests that the clock line is not getting distorted with time.

However, the last keyboard in the figure shows a huge difference in the distance, which might be due to the effect of temperature on the data. The dependence of this feature on temperature produces unwanted effects as can be seen from the figure.

The effect of temperature on the data is much more pronounced in Figure 5.2, where it can be seen that there is a definite difference between the two training data with one of them taken at an earlier time. The difference is on the order of mV, which suggests that if data is kept over time then the data affected by HKL and the training data will be at the same level, which might not allow one to detect the HKL successfully. Thus this escalates the need to obtain data affected by the temperature of the surroundings.

In an attempt to further prove this point, regression tests were done using the temperature data, and it was shown that the line voltage is a function of temperature. This test was done to see how the voltage of the clock line varies with temperature. It is proposed as future work that the voltage of the clock line with the HKL attached should be scaled according to the temperature when the HKL is not attached, and then comparison tests should be performed between the two voltage levels, with and without the HKL. For this

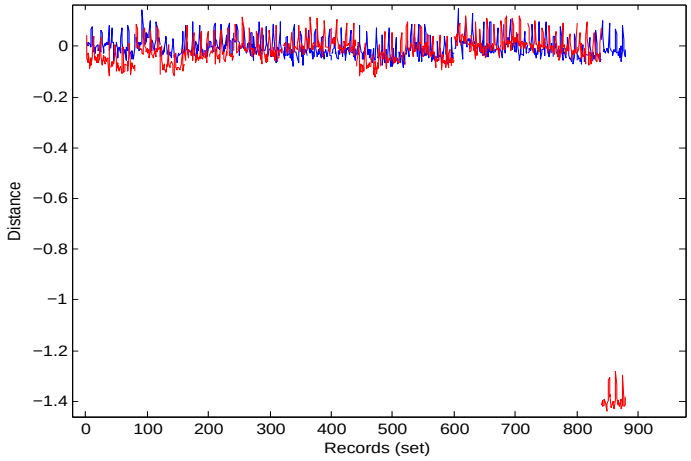


Fig. 5.1: The KLD between the training and a test distribution built from records without a keylogger for an earlier data set (blue) and without a keylogger attached taken later (red) for 22 (x-axis; records are grouped) keyboards. Since there is no separation it was asserted that both come from no HKL attached.

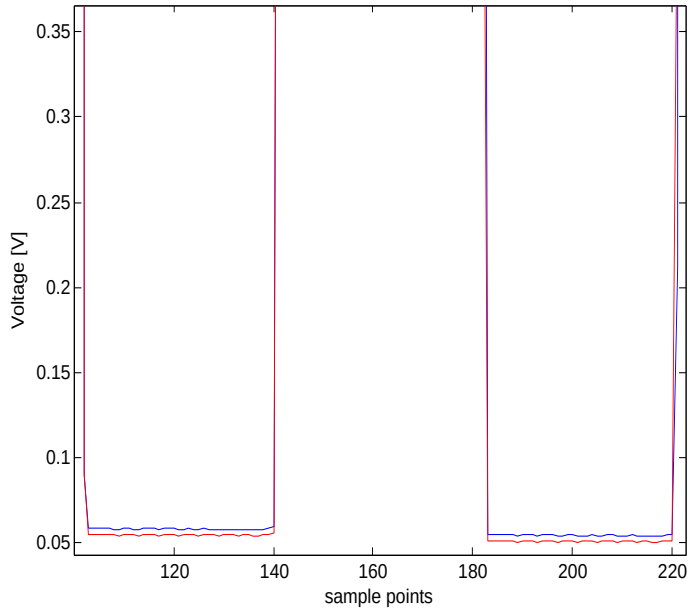


Fig. 5.2: Average of 1000 records taken for 1 keyboard's clock signal. Both the clock signals were taken from records without HKL connected with the one from an earlier data set (blue) and from a data set taken later (red). There is a difference between the two clock signals, which suggest that temperature affects the data.

reason it is necessary to obtain the dependence of the voltage on the temperature when the HKL is not attached.

5.1 Data Collection with Temperature

It bears repeating that this forms the motivation of the data collection. A part of the problem is put forward (issue or question) and should be dealt with regarding the temperature data collection. The problem is stated and a workaround is also given, by the process of data collection which is stated later in this chapter.

- For the data collection with temperature, ranges of temperatures should be applied to the location of the experiment. *Solution*: Data were taken twice for each keyboard. For the first data run, more time was covered without HKL, and for the second data run more time was covered for with HKL. This ensured that the temperature for almost the whole day was taken into account for both states of the system (with and without HKL).

It is possible that such a small difference in the lower clock level ($\approx$ 5mV from Chapter 2) could be shifted by thermal noise over time. The variations in the line voltage can be compensated by taking and equalizing the line voltage measurements at different temperatures.

The experimental setup for the data collection without temperature is kept the same, and the setup used for temperature sensing is as shown in Figure 5.3. The temperature per voltage is calculated by the temperature sensor, and the voltage measurement is done using a sampler that was placed after the temperature measuring circuit shown in Figure 5.3. Here, the ADC (sampler) is a National Instrument (NI) USB-6008 series DAQ board. The DAQ, used in differential input mode, has a full-scale voltage of 40 V and can sample at a rate of 10 kS/s. Given that the board is a 12-bit ADC, the minimum voltage that could be measured is with a resolution of ≈ 9.76 mV.

Four temperature sensors (transducers) were used which are low voltage LM35DT temperature transducers with a temperature coefficient of 10 mV/°C and which have a low

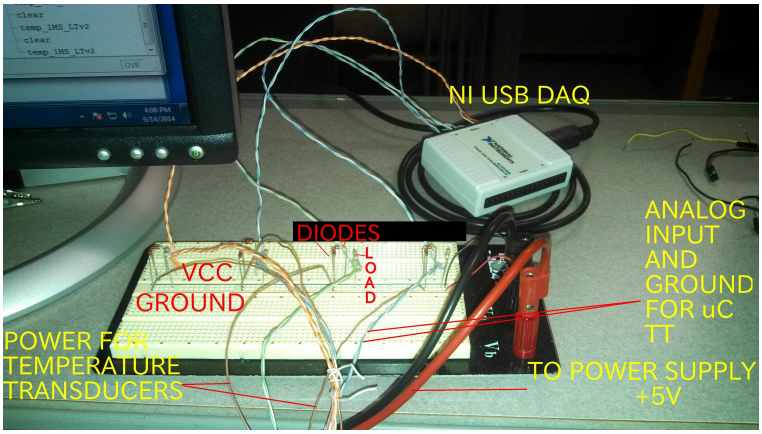
output impedance of 0.1 Ω for a 1 mA load [31]. These are used to measure temperatures of four different places in the experimental setup, i.e., the test keyboard, MCC DAQ board, the site of tapping into the PS/2 cable, and lastly, the μC. The temperature transducer with a 5 V supply has an output voltage of $V_{out} = 0.5+(0.01)$T for a temperature of T(°). The pin configuration is shown in Figure 5.4. These four temperature sensors are connected to a voltage divider circuit with resistances from diodes, in order to bias the sensors and proper usage. The connections are made as shown in Figure 5.3(b). Load resistances are connected for proper functioning of the circuit.

In this data collection process, instead of using a linear motor, a Numlock key toggling algorithm (sets it on if it is off and vice versa) was used which would send out information to the test keyboard connected to the system to change the state of the Numlock light (LED) in accordance with the controlling keyboard. Using this toggling algorithm, the Numlock key was virtually simulated, making the OS turn off the Numlock LED if on and turn it on if off accordingly.

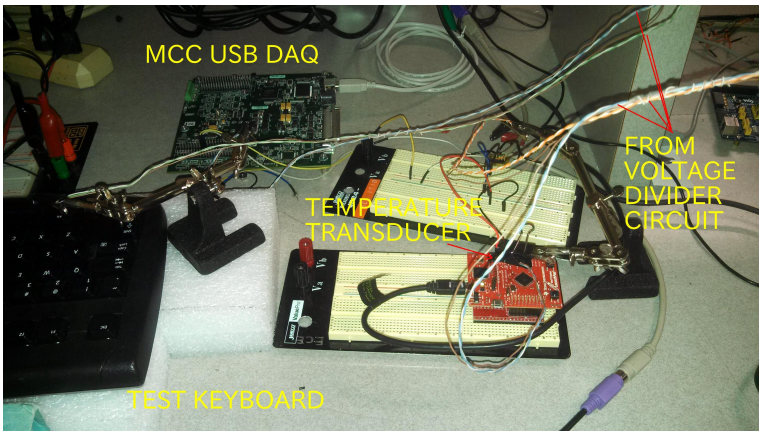
The toggling was done every 30 s, and for each falling edge of the clock the NI USB-6008 would collect temperature data and the MCC USB DAQ would collect the voltage data for the next 30 s. For each keyboard, data collection was done for 16 h with no HKL and for 5 h with HKL. The second data run were done for 5 h with no HKL and for 16 h with HKL. The data collection done for 16 hours was started at approximately 5pm and the data collection for 5 hours started at approximately 10am. Table 5.1 will give a better understanding of the data collection procedure. A third round of data were run where 24 hours of data without the keylogger was obtained, starting at approximately 10am.

5.2 Results

100 temperature readings, from all the temperature sensors, were taken for each triggering of the Numlock key LED. Linear regression (detailed description in Chapter 6) [32] was performed between the average of the temperature readings from one of the sensors, taken from the experimental setup for 23 keyboards, and the voltage of the clock line without the HKL. The linear model was first trained with a training set randomly chosen as



(a)



(b)

Fig. 5.3: (a) This figure shows the temperature setup used here. The sampler is shown on the top which takes input from the voltage divided circuit which has thermistors as resistors. The temperature is fed from the temperature transducers shown in (b). It also shows the four places where the temperature is taken in the original experimental setup. The test keyboard is shown which was taking the NOLOG data due to which the HKL was disvonnected from the PS/2 tap. TT = Temperature Transducer.

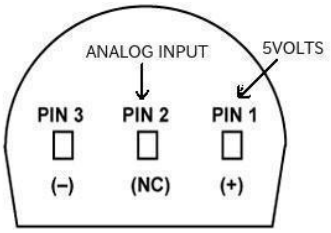


Fig. 5.4: This is the bottom view of the temperature transducer used in the experimental setup. This sensor is powered by the 5 V supply voltage and records the temperature at its location.

13 271 out of 66 263 records (for all 23 keyboards), which constitutes 20% of the records, and then the new responses were predicted based on the rest of the records (52 992) taken as the test records.

For obtaining the models, temperature was taken as the predictor and the output of the model was taken as the line voltage of the PS/2 clock. The models describe the dependence of the line voltage without the HKL (V_l) on temperature of the experimental setup (T). The relation between them is shown in the following equations. The two models are as follows, which were obtained for 23 keyboards.

$$V_l \sim 1 + T \tag{5.1}$$

Table 5.1: This shows the data run details. The first data run was taken in the presence of HKL for 5 hours and without HKL for 16 hours; the second data run was done with the times reversed. The third data run was taken for 23 keyboards for 24 hours each without the HKL. The approximate start times are also given.

Data Run 1	Data Run 2	Data Run 3
5 keyboards, with HKL for 5 hours (start 10AM)	5 keyboards, without HKL for 5 hours (start 10AM)	23 keyboards, without HKL for 24 hours (start 10AM)
5 keyboards, without HKL for 16 hours (start 5PM)	5 keyboards, with HKL for 16 hours (start 5PM)	

$$V_l \sim 1 + T + T^2 + T^{1.5} \tag{5.2}$$

For the model shown in Equation (5.1) the R-squared obtained is 0.999 and for the second model shown in Equation (5.2) the value of R-squared obtained is 0.994. The P-value was 0.0023 for the predictors with the standard error going to as low as 0.1 for the first model. The P-value was 0.01 for the predictors with the standard error going to as low as 0.7 for the second model. The highest estimated coefficient was 10.453 and 11.231 for the first and the second models respectively with the t-statistic value going to as low as 0.542 and 0.621 for the first and second models respectively (detailed definitions of these coefficients are given in Chapter 6). The root mean square error of 0.0646 suggests the high efficiency of the method used to obtain the dependency of voltage on temperature. It is clear from the models that the line voltage is a function of temperature and the correctness of the models is evident from the R-squared value.

5.3 Discussion

It can be seen from the above equations that the voltage is linearly dependent on and also has a quadratic relationship to the temperature. This explains the stability of the clock line voltage with time. It was seen previously in this chapter that the clock line voltage varies with time, giving rise to a larger difference in voltage distributions than the previously stored ones. The linear dependence of voltage on temperature explains that the clock line voltage decreased with the decrease in temperature (with the approaching season) giving a higher difference than the previously collected data.

Chapter 6

Unique Identity of a Keylogger on Keyboards

The proposed feature comparison methods discussed in the previous chapters used training records (obtained from a baseline state without HKL) alongside test records to come to an assertion. In this chapter, different data-mining applications were used to perform the comparison with no training records. As an answer to the problem statement, the experiment sought to detect the presence of HKL given any record from any keyboard. More specifically, the distribution constructed from the features with both the known and unknown state, i.e., with and without HKL respectively, were fed to the PLD machine in this case and the algorithm tried to separate the two classes (states). The introduction of the data mining applications such as classification and clustering led to the use of a number of machine-learning algorithms for a large quantity of data. A number of classification methods were used. Classification techniques are discussed in detail. support vector machines (SVM) and one-class support vector machines are the most popular ones among them. Some of the methods discussed here are linear regression and logistic regression as generalized linear regression.

6.1 Feature Extraction

Chapter 2 indicated that the rise times, fall times, and slew rates were the ones which were most affected by the HKL, including the upper and lower transition of the clock signal for the down keypress. It was also noted from the data that due to the change in the rise time and fall time, there should be a change in the duration of the signal, which prompted its consideration as one of the features. The inclusion of duration as one of the features prompted another question: What does the response of the system look like in the frequency domain? This resulted in taking the Fourier transform (FFT in MATLAB) of

the clock signal for the down keypress. The power and index of the resulting frequency bins were taken as the other set of features. In some of the classification algorithms, the difference between the power of the frequency bins was taken as a feature. It should be noted here that the clock signal (which is a square wave) gives a frequency domain response with frequency spikes at every odd harmonic [23].

Following is the summary of features which were used for the PLD using data mining applications:

- Rise time of the individual clock transitions,
- Fall time of the individual clock transitions,
- Duration of the individual clock transitions,
- Power of the frequency bins of the Fourier transformed clock signal,
- Difference in power of the frequency bins of the Fourier transformed clock signal,
- Index of the frequency bins of the Fourier transformed clock signal.

Rise times, fall times, and slew rates were calculated according to a particular standard [22] for the keydown portion (of the clock signal) of the record. For the same portion of the clock signal the Fourier transform was taken according to Algorithm 1 and the power and index for each frequency bin was calculated. The position of the fundamental frequency bin is used to calculate the position of the remaining frequency bins as far as the algorithm is concerned.

The output of Algorithm 1 formed two separate distributions, which were used in the classification algorithms discussed later in this chapter. It was also noted that there could be more features which could be extracted and dealt with, but the negative result of these classification algorithms was an indication that one should not look for any more features.

The set of features which were extracted and discussed previously (in Chapters 3 and 4), viz., rise time, fall time in the transient effects, and the power and indices of the frequency bins of the Fourier transformed clock signal, were used for linear regression. A logarithmic

Algorithm 1: Extracting the power and indices of the frequency bins of the sampled lower level of the clock signal.

Input : **R** (a sample point-by-record matrix of line measurements for the first downward transition of the clock signal)
Output: *SP,SI* (power and indices of the frequency bins of **R** after the Fourier transformation of the clock's lower level)

$SP = \varnothing$;
$SI = \varnothing$;
foreach $R_i \triangleq R_{*,i}$ **do**
  $\{X \subset FFT(R_i) : \forall\ x \in X = FFT(R_i/2)\}$; `// As the second half is the mirror image of the first half`
  $\{P = Power(X)\}$;
  $\{I = Index(X)\}$;
  $SP \leftarrow SP \cup P$;
  $SI \leftarrow SI \cup I$;

axis is used in the y-axis in order to look at the power of the frequency bins as they are in general lower frequencies due to the lower frequencies of the PS/2 clock. Using the FFT function from MATLAB, it can be seen from Figure 6.1(a) that the frequency domain signal was unsmooth. Smoothing of the signal was necessary so that there is a clear distinction between the frequency bins along with the aliasing frequency components in between. Thus, for smoothing of the signal a special type of filter, a Gaussian filter, known as Gaussian smoothing was used [33]. Looking at the unsmooth signal one can easily infer that the harmonics (frequency spikes) after every odd interval are crowded with a band of unwanted signals, which calls for resolving two issues—noise removal and edge (frequency spike) preservation. While the use of low-pass filters would eliminate the high frequency components, the Gaussian filter has a special capability to act as low-pass filter without a sharp cutoff but with a decaying tail of cutoff frequency that reduces as the frequency of the signal increases. This takes care of both the issues (shown in Figure 6.1(b)) and is capable of doing so because the Fourier transform of the Gaussian function is the Gaussian function itself and has the minimum time-bandwidth product.

It is important to note in Figures 6.1(a) and 6.1(b) the presence of the aliasing frequency components between the actual frequency components. These occur when the sampling of the digital signal (PS/2 clock in our case) is low at a frequency comparable to or smaller

than the signal being measured. These become more pronounced for higher harmonics. It should be noted here that the aliasing frequencies can also be used as a feature set as the frequency components were different for with and without HKL.

The Gaussian filter [34] is often considered the ideal time domain filter used mostly in oscilloscopes and digital telecommunication systems. It is given by the equation

$$g(x) = \frac{1}{\sqrt{2\Pi}.\sigma}.e^{\frac{-x^2}{2\sigma^2}}. \tag{6.1}$$

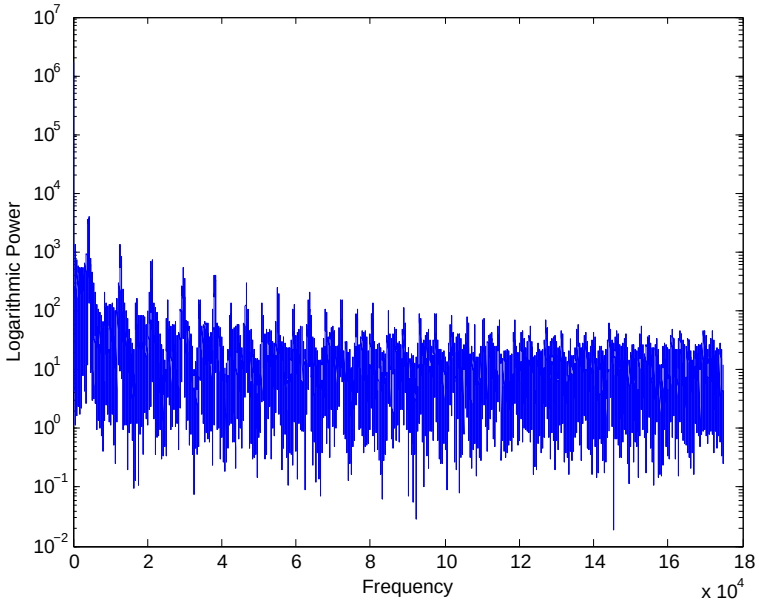
In Equation (6.1), σ is the standard deviation and the calculations were done keeping standard deviation as the parameter, keeping it as 10 and the linspace as 600. Features are extracted after the frequency-domain signal is filtered.

6.2 Classification Methods

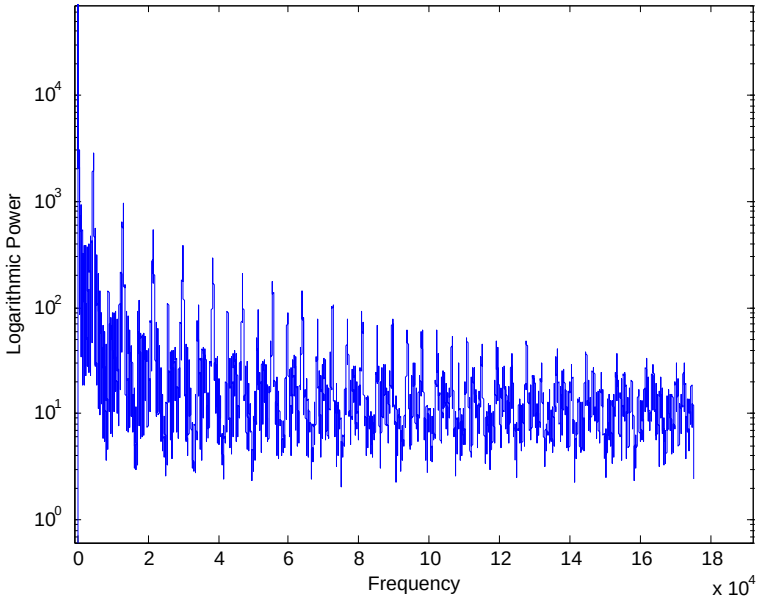
Classification [35] can be defined as a predictive class where the response takes the values across discrete categories, which in this case were keyboards with HKL and without HKL. Classification was chosen as one of the ways to differentiate between the data because a label could be obtained for each class which can be observed. The corresponding predictor value can be looked at based on the class. In classification, only pairs (x_i,y_i) were observed, where $i = 1, 2, .., n$ where n is the number of observations and y_i gives the class of the i^{th} observation and $x_i \in \mathbb{R}$ are the measurements of the predictor variables.

6.2.1 Classification using Linear Regression

The main idea behind the linear regression model is that the relation between the response variable and the predictor variable is linear. Keeping in mind that there are two classes (HKL and no HKL), binary classification is considered with linear regression [32]. Thus, it can be obtained that $y_i \in 1$ and 2. Calculations were done in MATLAB and fitting functions were used. The inputs to the fitting function are x_i and y_i where y is the response variable and x is the predictor variable. Treating the response as continuous, this function finds the linear regression coefficients of the response vector $y \in \mathbb{R}$ on to the predictors. It



(a)



(b)

Fig. 6.1: (a) Fourier-transformed lower level clock signal before using the Gaussian filter for smoothing. (b) Fourier-transformed signal after the Gaussian filter is applied and the signal smoothed.

can be written in equation form in the following way

$$y_i = \beta_1 x_{i1} + ... + \beta_1 x_{ip} = \mathbf{x_i^T}\boldsymbol{\beta} + \varepsilon_i, \tag{6.2}$$

where ^{T} indicates the transpose matrix, i = 1,2,.....,n and ε_i is the noise factor or the error term in the technique employed. The equation can be more clearly written as follows

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ y_n \end{pmatrix}, \mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ . \\ . \\ . \\ . \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & .... & x_{1p} \\ x_{21} & .... & x_{2p} \\ . & . & . \\ . & . & . \\ . & . & . \\ . & . & . \\ x_{n1} & .... & x_{np} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ . \\ \beta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ . \\ . \\ \varepsilon_n \end{pmatrix}. \tag{6.3}$$

- y_i is the response variable also known as the endogenous variable or regressand. The variable that in a data set has to be modeled as a dependent variable and the one that has to be modeled as an independent variable are based on the presumption that the value of one of the variables is directly influenced by another variable.

- x_{i1}, x_{i2}, ..., x_{ip} are the predictor variables, regressors, or independent variables. The matrix $\mathbf{X}$ is known as the design matrix. The predictors are used everywhere, be it MATLAB, R, or SAS, they have a constant variable among the regressor variables and the corresponding β value is known as the intercept. Some of the regressors may be independent of each other, but the system is linear as long as all the regressor variables are linearly dependent on the coefficient variable β.

- $\boldsymbol{\beta}$ is the matrix of regression coefficients which is mainly the focus of the linear regression methodology.

- The ε is the error or noise term that estimates the factors influencing the response variable other than predictor variables. It has to be kept in mind that the relationship between this error term and the independent variables is one of the most important things while doing classification using linear regression because the error term is the expected change in y for a particular change in x_j.

Through this fitted linear regression model one can identify the relationship between a single predictor variable and the response variable while keeping all the predictor variables constant. Thus the β_j is the measure of the expected change in y for one unit of change in x_j when the other covariates are held fixed, that is the expected value of the partial derivative of y with respect to x_j.

The features obtained were fed to the linear model fitting function as input where the predictor variables are the feature matrix obtained after feature extraction, and the response matrix is the expected value as another input. Several models of the fitting function were used, of which some will be described. The point to be noted here is that results were different when different models for the fitting function were used. The quality of the results (how well two different classes are separated) was decided based on the number of false negatives and the number of false positives. This is because the input given contained both the known (without HKL) and unknown system state values, which needed to be put in the right class. The prediction of the class using the fitting model were used to see how accurately the two classes are separated.

There are a number of models for the fitting functions which were used, viz. the constant model, the linear model, the quadratic model, and the pure quadratic model. While the constant model contains just the constant intercept term, the quadratic model contains not only the linear terms and the squared terms but also all the products of pairs of distinct predictors. This clearly states that the constant model gives the smallest number of terms used for fitting and the quadratic model gives the largest number of terms for fitting.

It is important to mention the use of the robust functions [36] while using the models in the fitting function, because the data drawn from the probability distributions have different

standard deviations as discussed in Chapter 2. Refraining from the use of all the functions as the increase of the number of terms was avoided, only three were used. The default function is taken as the bisquared function with the other important ones being Cauchy's function and the logistic function. These functions were chosen at random as they have different responses for different types of errors in a distribution.

6.2.2 Logistic Regression as a Classification Algorithm

While linear regression can be used in the case where more than two classes can be used for predicting, logistic regression is used specifically where the dependent variable is binary (two classes). This classifier is used to predict the response of a dependent variable based on the features collected. Logistic regression [37], known as a multinomial logistic regression, can be used for cases with more than two classes. Unlike the linear regression technique where the residuals needed to be normally distributed, in logistic regression things are based on maximum-likelihood estimation. It is impossible to find the coefficient in the closed form to maximize the likelihood function, and as a result one needs to use the iterative method. This is the reason why logistic regression can be used as a generalized linear model.

The distributions obtained from the feature-extraction algorithms discussed previously were taken as one of the inputs, the predictor matrix, the other one being the response matrix consisting of 0 s and 1 s. The logistic regression can be used as a type of generalized linear model [38] in the following way:

$$\text{logit}(\mathbb{E}[Y_i|x_{1,i},....,x_{m,i}]) = \text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + ... + \beta_m x_{m,i}. \tag{6.4}$$

In compact notation it can be obtained that

$$\text{logit}(\mathbb{E}[\mathbf{Y_i}|\mathbf{X_i}]) = \text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta}.\mathbf{X_i}. \tag{6.5}$$

This generalized model predicts variables with different probability distributions by

fitting a linear predictor function of the above form. The logit function is the natural logarithm of odds, which has the potential of converting the probability values from (0,1) to $(-\infty,+\infty)$. It needs to be noted that both one-step and stepwise regression were used for classification purposes as the step-by-step regression adds and removes the predictor variables at every step of the regression process. *Features*: The same set of features as in case of linear regression were used, along with the aliasing frequencies. The aliasing frequencies were added because in case of a dedicated method for two-class classification, the increase in the input distribution size results in better classification results.

6.3 Support Vector Machines (SVMs)

Formally, an SVM works by building a hyperplane or a number of hyperplanes which can be used for classification. A *hyperplane* is a plane which is one-dimension-less subspace than the ambient space where it is placed. As it was always the case before, for this classification algorithm, two classes need to be considered, and the best separation for any plane is the one with the largest distance from the closest point which was taken as training data. It can be seen that the support vector machine is trained and is ready to classify the new data points. Each data point is considered to be a vector that can be separated by a plane. As there could be a number of planes capable of classifying the data, the best choice is the one which was farthest from the nearest data point in each of the classes. The fact that the classification engine was trained with the data first and then tested with another set of data answers the question of why this method were used for classification.

A library for support vector machines was used [39] for the SVM calculations in MATLAB. The training set was obtained from one-tenth (100) of the number of total records (1000) with a known baseline system (no HKL). The model obtained from the training data took the test data from both known and unknown system states for it to be classified into the two classes. The decision algorithms were applied to the test data to predict the label (with or without HKL) of the data. The training data was of the form $X_i \in \mathbb{R}$ for the number of features taken and the label matrix being $Y_i \in 0,1$ for all features taken into consideration. The training data was linearly separated and there could be two hyperplanes

for separating HKL and no HKL. A primal optimization problem needed to be solved in order to minimize the difference between the two hyperplanes [40]. As the distance between the hyperplanes was minimized (converge to one hyperplane from two) and the data was separated, the optimal objective value and ρ, which was the bias term in the decision function for the hyperplane separation, were obtained. Using the same set of features for linear and logistic regression, accuracy of the classification method was measured by putting the test data, which consisted of all records from all keyboards in the experiment done here, in the object model which was defined as:

$$Accuracy = \frac{\#\ correctly\ predicted\ data}{\#\ total\ testing\ data} \times 100\%. \tag{6.6}$$

6.4 One-Class Support Vector Machines

The major difference between support vector machines and one-class support vector machine is that the latter deals with only the positive class, not the negative class. This is so because it takes into consideration that the negative classes can be negative in their own way and they can belong to a different class altogether. Thus in order to solve the problem it is taken that only one class needs to be considered and that is the class of no HKL.

The same library which was used for support vector machines was used for one-class support vector machines. The training data and the testing data both were the known state of the keyboard (no HKL) and an unknown state of the keyboard, which was the same in the case of support vector machines. Taking the same set of features into consideration, one can conceptually imagine that a hypersphere is taken to fit all the positive training data into it. The hypersphere tries to obtain the support of the place where the positive data is clustered in order to separate them from the rest of the data which in turn may cause over-fitting due to the inclusion of most of the data in one class (if not classified correctly).

6.5 Discussion and Results

These results are discussed in a dedicated section because they do not validate the first-order model given in Chapter 2, and it cannot be asserted from here that an HKL is

attached to the system. A dedicated section is given for the results obtained from the KLD and chi-squared discussed in Chapter 4. It should be noted that while discussing the results of a statistical analysis one should always define the metrics used to judge the efficacy of the results. Thus the definitions of some of the terms used to present the results are given as follows

- P-Value: This value gives the null hypothesis, which means that if a particular model has a low p-value then there is a relation between the predictor variables and response variables. The higher the value, the less the relation is between the predictor and the response variable.
- Coefficient of Determination: The coefficient of determination is the statistical measure of how close the data is to the fitted regression line. This is also known as R^2 (R-squared), which indicates that the lower its value, the lower variability of response data around its mean.
- F-Statistic: This gives the test statistic for testing the significance of the components in the model. These tests show that there is considerable difference in the variance of the means of the two distributions (with and without HKL).

6.5.1 Linear Regression

For the quadratic model, the efficiency values are intimidating as the false positive percentage goes to 60% and the false negatives go to the remaining 40%. This can be explained by the fact that the quadratic and pure quadratic models have a constant value, a linear term, and a square term, resulting in a large number of terms. With the other models too (constant model, interactions model, and pure quadratic), in most cases with a large number of terms the false positives were comparatively fewer. This was further augmented by the value of adjusted R-squared obtained in the linear model, which is 31.7% (less than 50%), and the R-squared of 32.2%. Also the R-squared value in the quadratic model is 60%. It can be concluded from this that the regression method was unable to classify the data into two classes and it cannot be asserted that the class formed apart

from the baseline has HKL attached. The residuals for the quadratic model are given in Figure 6.2.

6.5.2 Logistic Regression

One-step regression and step-by-step regression were performed as discussed earlier and a value of R-squared reported for the one-step process is 60%. In both these cases, for the pure quadratic model, the value of R-squared did not go above 60%. The pure quadratic model and the linear model did not yield the desired R-squared because for the linear model only the linear terms (31.66% R-squared) were considered and for the pure quadratic model even though the square terms were considered (giving an increase in R-squared), the value of R-squared was still low. Even though the quadratic model takes into consideration all the product terms including the constant term, it reported an R-Squared value of 84.5% with the number of terms exceeding 850 in the model. Figure 6.2 shows the residuals for the quadratic model of the linear regression.

Some of the main reasons behind the failure of the above classification methods (linear and logistic regression) are discussed as follows.

- A large number of predictor variables were taken to only two cases (HKL and no HKL), leading to non-convergence.
- High correlations between predictor variables were obtained. The multicollinearity increases and the increase in standard error make the likelihood of convergence much lower.
- In this case there is no complete separation, as is evident from our accuracy (R-squared) measurements. Thus the problem of complete separation is not the case here.

6.5.3 Support Vector Machines

Putting the training data and the features in the SVM, the accuracy obtained is 63% for the multiclass support vector machine. This suggests that the machine was unable to

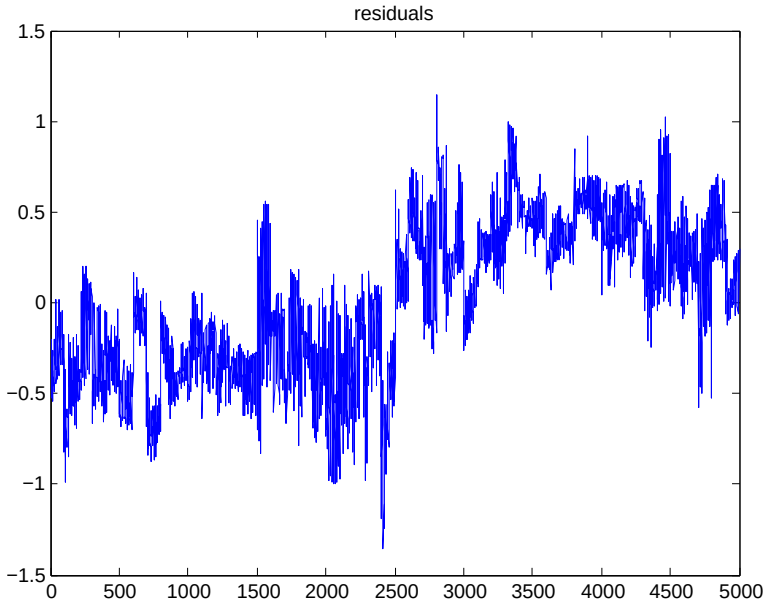


Fig. 6.2: Residuals for the quadratic model for linear regression. For both plots, the x-axis shows the number of records used and the number of features used for each record, and the y-axis shows the values of the features used.

separate the data from the two system states provided as input to the machine, as discussed earlier. This indicates that the data from a particular system state cannot be predicted correctly by the support vectors created. Thus, no assertion about the presence of the HKL can be made here.

6.5.4 One-Class Support Vector Machines

Accuracy comes down to as low as 2% with less training data. This occurs because the data appear to be scattered in such a way that the virtual hypersphere created was not able to fit the positive training data at one go. Again, no assertion can be made here about the presence of the HKL.

Chapter 7

Conclusion and Future Work

The HKL produced a measurable distortion in the clock of the PS/2 keyboard attached for testing. A comparison methodology was built to compare the line voltages with and without the attachment of HKL. Experiments were done and it was shown that 4–5 keystrokes were necessary to identify the presence of an HKL with the lower level and upper level of the clock used for detection and 50–100 keystrokes were necessary to identify the presence of HKL with the rise time, fall time, and slew rate. In order to detect the state of a keyboard without the use of any training records, data mining methods were used. This did not prove to be useful, mostly due to the inability to identify features from the clock line.

Future work includes the use of methods to identify any state of a keyboard without the use of training records. For this, the feature set should be wisely chosen which might produce differences which are present only for a particular state of the keyboard. This can be done by identifying features from the data or any other places which are not affected by the type of keyboard placed for testing (in other words independent of the resistance and voltage of the keyboard). Second, the temperature data should be taken and the voltage of the clock line with the HKL attached should be scaled according to the temperature when the HKL is not attached and then the comparison tests should be performed between the two voltage levels with and without the HKL.

References

[1] Keelog, "Keygrabber module," 2013. [Online]. Available: http://www.keelog.com/hardware_keyboard_logger.html

[2] J. E. Dunn, "Usschool expels pupils for using hardware keyloggers to change grades," 2014. [Online]. Available: http://news.techworld.com/security/3500558/us-school-expels-pupils-for-using-hardware-keyloggers-to-change-grades/

[3] E. Kabelmas, "Brute forcing hardware keyloggers," 2012. [Online]. Available: http://kabelmast.wordpress.com/2012/10/26/brute-forcing-hardware-keyloggers/

[4] L. King, "Hackers in court for 229m keylogger scam," 2009. [Online]. Available: http://www.computerworlduk.com/news/security/12923/hackers-in-court-for-229m-keylogger-scam/

[5] KeeLog, "Open source diy hardware keylogger," 2012. [Online]. Available: http://www.keelog.com/diy.html

[6] B. Nakra and K. Chaudhry, *Instrumentation, Measurement and Analysis*. New Delhi: Tata McGraw-Hill Education, 2004.

[7] P. W. Smith, *Transient Electronics*. New York: Wiley, 2002.

[8] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proceedings of the third ACM conference on Wireless network security*. ACM, 2010, pp. 89–98.

[9] R. M. Gerdes, T. E. Daniels, M. Mina, and S. Russell, "Device identification via analog signal fingerprinting: A matched filter approach." in *NDSS*, 2006.

[10] R. M. Gerdes, "Physical layer identification: methodology, security, and origin of variation," Ph.D. dissertation, Iowa State University, Ames, IA, 2011.

[11] B. Danev, "Physical-layer identification of wireless devices," Ph.D. dissertation, ETH Zurich, Zurich, Switzerland, 2011.

[12] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 6, 2012.

[13] F. Mihailowitsch, "Detecting hardware keyloggers," *HITB SecConf, Kuala Lumpur, Malaysia (October 2010); cirosec GmbH. Hack In The Box*, 2010.

[14] S. Ming-Chang, W. Han, and P. Wu, "Techniques for detecting keyloggers in computer systems," Patent US 8 707 437, Apr. 22, 2014.

[15] Wikipedia, "Hardware keylogger — wikipedia, the free encyclopedia," 2013, [Online; accessed 19-September-2014]. [Online]. Available: \url{http://en.wikipedia.org/w/index.php?title=Hardware_keylogger&oldid=582501391}

[16] I. Keshet, P. Berengoltz, and L. Dorrendorf, "System and method for incapacitating a hardware keylogger," Patent US 20 110 219 457, Oct. 11, 2011.

[17] R. Baloch, *An Introduction To Keyloggers, RATS And Malware.* e-knjiga, 2011.

[18] M. Greene and M. Parker, "Techniques for detecting keyloggers in computer systems," Patent US 20 070 169 191, Jul. 19, 2007.

[19] A. Chapweske, "The ps/2 mouse/keyboard protocol," 2003. [Online]. Available: http://www.computer-engineering.org/ps2protocol

[20] J. W. Nilsson and S. Riedel, *Electric Circuits.* New Jersey: Prentice Hall, 2010.

[21] Texas Instruments, "Use conditions for 5-v tolerant gpios on Tiva C series TM4C123x microcontrollers," Application Report, 2013.

[22] IEEE, "Standard for transitions, pulses, and related waveforms," IEEE Std 181–2011, 2011.

[23] Y. Wei and N. Chen, "Square wave analysis," *Journal of Mathematical Physics*, vol. 39, no. 8, pp. 4226–4245, 1998.

[24] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, pp. 79–86, 1951.

[25] S. Kullback, *Information Theory and Statistics.* Gloucester: Courier Dover Publications, 1997.

[26] O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *Computer Vision–ECCV 2010.* New York: Springer, 2010, pp. 749–762.

[27] J. M. Joyce, "Kullback-leibler divergence," in *International Encyclopedia of Statistical Science.* New York: Springer, 2011, pp. 720–722.

[28] Texas Instruments, "Tiva TM4C123GH6PM microcontroller," datasheet, 2013.

[29] Linear Technology, "LT1793 Low Noise, Picoampere Bias Current, JFET Input Op Amp," datasheet, 1999.

[30] B. Efron, "Estimating the error rate of a prediction rule: improvement on cross-validation," *Journal of the American Statistical Association*, vol. 78, no. 382, pp. 316–331, 1983.

[31] Ladyada, "Tmp36 temperature sensor," 2013. [Online]. Available: https://learn.adafruit.com/downloads/pdf/tmp36-temperature-sensor.pdf

[32] G. A. Seber and A. J. Lee, *Linear regression analysis.* New York: John Wiley & Sons, vol. 936, 2012.

[33] A. P. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, vol. 9. IEEE, 1984, pp. 150–153.

[34] H. Blinchikoff and H. Krause, *Filtering in the time and frequency domains.* The Institution of Engineering and Technology, 2001.

[35] X. Yin and J. Han, "Cpar: Classification based on predictive association rules," in *SDM*, vol. 3. SIAM, 2003, pp. 369–376.

[36] P. J. Huber, *Robust Statistics.* Heidelberg: Springer, 2011.

[37] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Introduction to the logistic regression model.* Wiley Online Library, 2000.

[38] C. E. McCulloch and J. M. Neuhaus, *Generalized linear mixed models.* New York: Wiley Online Library, 2001.

[39] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[40] D. S. Sayad, "Support vector machine - classification (svm)," 2010-2014. [Online]. Available: http://www.saedsayad.com/support_vector_machine.htm