Physical-Layer Detection of Hardware Keyloggers

Ryan M. Gerdes(✉) and Saptarshi Mallick

Utah State University, Logan, UT 84322, USA
ryan.gerdes@usu.edu, saptarshi.mallick@aggiemail.usu.edu

Abstract. This work examines the general problem of detecting the presence of hardware keyloggers (HKLs), and specifically focuses on HKLs that are self-powered and take measures, such as passively tapping the keyboard line, to avoid detection. The work is inspired by the *observer effect*, which maintains that the act of observation impacts the observed. First, a model for HKLs is proposed, and experimentally validated, that explains how attaching a HKL necessarily affects the electrical characteristics of the system it is attached to. The model then motivates the selection of features that can be used for detection. A comparison framework is put forth that is sensitive enough to identify the minute changes in these features caused by HKLs. Experimental work carried out on a custom keylogger designed to conceal its presence, at the expense of reliability, shows that it is possible to detect stealthy and evasive keyloggers by observing as few as five keystrokes. Optimal attack strategies are devised to evade detection by the proposed approach and countermeasures evaluated that show detection is still possible. Environmental effects on detection performance are also examined and accounted for.

Keywords: Physical layer identification · Device fingerprinting · Keyloggers · Hardware keylogger

1 Introduction

A hardware keylogger (HKL) is a device, situated between the analog interfaces of a computer and its keyboard, that recovers the keystrokes transmitted by a keyboard through the sampling of the electrical impulses transmitted by the keyboard. These devices represent a real and persistent public threat, as evidenced by the discovery that keylogger-like devices inside point-of-sale terminals at 63 stores were used to steal customer credit card information [34]. When installed on public computers, HKLs enable identify theft on a wide scale and allow an attacker to acquire credentials that may be used to gain access to other systems and services (as a Cal State student did to perpetrate voting fraud [1]). On private computers the surreptitious installation of HKLs makes it possible for an attacker to bypass full disk encryption. These devices are inexpensive and readily available (the authors found keyloggers for $30–$400, depending on the features, such as keystroke capacity, point of attachment, size, and wireless transmission

H. Bos et al. (Eds.): RAID 2015, LNCS 9404, pp. 26–47, 2015.
DOI: 10.1007/978-3-319-26362-5_2

of recorded keystrokes, from eight manufacturers). Alternatively, the knowledge required to build an efficient HKL can be obtained in an undergraduate microcontrollers course or instructions can be procured for free online.

The most popular countermeasure against HKLs is simple visual inspection [35]; however, this is impractical for large organizations [3] and is complicated by the fact that keyloggers are increasingly unobtrusive. Indeed, HKLs are available for embedding inside keyboards [18], inside laptops [19], and as PCI cards [20] for the expressed purpose of avoiding casual visual detection. Existing non-visual methods [24] are also only capable of detecting certain types of HKL.

To enable the detection of stealthy and evasive hardware keyloggers we propose an approach based on the *observer effect*, which states that the act of observing must perforce impact the phenomenon being observed [28]. Specifically, the mere fact that an attacker connects a piece of equipment (the HKL) to measure the output of a keyboard affects the output of the keyboard. The mechanism by which this occurs is known as *loading*, a well known problem encountered, for example, when attempting to measure the voltage of a high resistance circuit [25]. For this work we examined a HKL especially designed for stealth and evasion and found that it impacted keyboard signaling to a measurable and detectable degree. In fact, we conjecture that any HKL that recovers keystrokes via direct measurement of the wired keyboard/PC communication channel, even those hidden within a keyboard or PC, should be discoverable using our method.

Within the broader context of the security literature, our work falls into the category of physical layer identification (PLI), also known as device fingerprinting. In PLI hardware and manufacturing inconsistencies that cause minute and unique variations in the signaling behavior of devices are utilized for identification and monitoring purposes [8]. The approach outlined below utilizes PLI techniques for keylogger detection by having the host computer fingerprint the keyboard and compare the fingerprint to baseline fingerprints, which were acquired in the absence of a keylogger, to determine whether a keylogger has been attached.

1.1 Related Work

Countermeasures for HKLs generally fall into one of four categories: avoidance, detection, exhaustion, and obfuscation.

An avoidance strategy involves giving the PC input using another method, such as an onscreen keyboard, whenever sensitive information is called for [35]. This method tends to be tedious, cannot be used while others are nearby, and is potentially vulnerable to screen capture, though methods have been proposed to counter the latter threat [29].

Resource exhaustion, wherein spurious keystrokes and commands are received from/sent to the keyboard so as fill/overwrite its memory, was suggested in [14,24]. While severely resource-constrained HKLs, e.g. a self-powered HKL that wirelessly transmits keystrokes, may be uniquely vulnerable to this type of countermeasure, exhaustion is generally an impractical strategy as HKLs

can have GBs of memory while the clock of the keyboard is on the order of ten KHz (the author of [24] gives 109 min to fill 64 KB at ≈10 keystrokes per second).

Obfuscation refers to the encryption of keystrokes before they are transmitted (the keyboard and PC sharing a secret key) or hiding keystrokes in a continuous flood of random keystrokes (perhaps the PC and keyboard share the seed of a common pseudo-random number generator). The authors are unaware of either technique being used in practice.

Current, non-visual, HKL detection methods rely upon changes in timing or deviations in power caused by the keylogger drawing power from the bus [24]. These methods, however, are only effective against inline keyloggers; i.e. those that are connected in serial with the keyboard/computer and actively intercept and then recreate the signals from the keyboard. Stealthy and evasive keyloggers—i.e. ones that are self-powered, hidden within the keyboard or connectors, and passively tap the keyboard by being connected in parallel with it—are undetectable using these approaches.

The possibility of using PLI to detect taps on lines—i.e. eavesdroppers on wired communications—was first suggested in [13]. Ours is the first work to directly confirm this conjecture, though in [10] it was demonstrated that changes to the communication medium (in that case increasing the length of the Ethernet cable) leads to a perceptible shift in a device's fingerprint. The reader is referred to [5,7,12] for an overview of PLI techniques, issues, and results.

1.2 Paper Structure

In the next section we describe the types of keyloggers and set forth a threat model that characterizes the type of keyloger we hope to detect. We then explain the workings of the PS/2 protocol to the extent necessary to understand the operation of keyloggers. A first-order model that explains how a HKL indubitably affects the system it is connected to concludes the section. In Sect. 3, our architecture for detecting keyloggers is introduced. We then leverage the model set forth in the previous section to select features to detect the presence of a HKL. The methods used for the extraction and comparison of features are also discussed. Experimental validation of the detection methodology is described in Sect. 4. Details of the keylogger designed to test our approach are given and experimental procedures discussed. Section 5 considers feature stability due to changes in the environment and examines the extent to which attacker countermeasures could be employed to evade detection. We conclude with further avenues of research.

2 Theory of Detection

The types and characteristics of HKL are discussed and a threat model is chosen that maximizes an attacker's chances of remaining undetected. The PS/2 protocol and physical layer are described to understand how they are leveraged by HKL designers. The effects a HKL has on transient and steady-state line voltages are examined through the use of a first-order model.

2.1 Threat Model and Assumptions

Hardware keyloggers may be divided into *active* and *passive* types, either of which may be self-powered or use the resources of the host PC for power. The active type, sometimes known as inline, sits between (in series with) the keyboard and host PC and intercepts and regenerates the signaling of the keyboard/PC. According to [24] these are the most common commercial type of keylogger. A passive HKL, on the other hand, sits aside (in parallel with) the keyboard/PC and simply observes the state of the line connecting the two to recover keystrokes. For the purposes of this work, we consider a HKL *stealthy* if it does not draw upon the host PC for power and *evasive* if it takes measures against a detection methodology to avoid discovery. The keylogger we studied (modeled on a commercial HKL design [17] and discussed in Sect. 4.1) was passive and stealthy; evasive variants are considered in Sects. 5.2 and 5.3.

While all of the HKLs we are aware of are based on microcontrollers (uC), in some circumstances, such as when a special form-factor is called for or in an attempt reduce energy consumption, an attacker might design an application-specific integrated circuit (ASIC) HKL. Without loss of generality, as ASICs and uCs use the same transistor-level technology for interfacing purposes, our keyloggers were constructed using a uC. This simplified development and testing significantly as uCs are commonly equipped with enough features (general-purpose input/output [GPIO] ports, memory, samplers, converters, and computation abilities) to allow for a flexible HKL design.

Given the success of previous PLI work in identifying wired devices [11], we chose not to examine active devices as it was thought that they would be easily discoverable. In fact, a sophisticated PLI approach is probably unnecessary to detect these devices due to the fact the signals they generate are based on GPIO ports that do not attempt to reproduce exactly the analog signaling of the keyboard. This is because GPIO ports know only two outputs, which correspond to the logic high and low voltage levels of the microcontroller.[1] In addition, while detection methods exist for active keyloggers that may or may not draw power from the host PC [24], none do for the passive, stealthy variety.

PS/2 keyloggers are used to illustrate the approach as they are simpler and easier to understand. Because of the electrical and signaling similarities of USB and PS/2 line drivers, comparable loading effects will be observed when a USB HKL is connected, so the approach would still be effective for USB keyboards. In fact, given the relatively higher speed, it should be easier to detect a USB keylogger, as the HKL load would produce greater distortions at higher frequencies (i.e. because of the slow clock speed of the PS/2 protocol, it is actually more difficult to detect the presence of a HKL). Host-to-keyboard communication is also disregarded (both PS/2 and USB keyboard protocols are bi-directional).

Finally, we attached our keyloggers to a tap point in the middle of the PS/2 cable (details given in Sect. 4.2). Because of the low frequencies of the signaling and short distances involved, the lumped element model [26] still holds,

[1] In Sect. 5.2 we do examine the case of an evasive HKL designed to defeat our detection method by reproducing the keyboard's signal exactly.

which implies that the actual point of attachment (i.e. inside or outside the keyboard/PC) is immaterial. Thus, our setup mimics an attacker connecting a HKL to an arbitrary point between the analog interfaces of the keyboard/PC.

In summary, we consider an attacker who has connected a passive PS/2 HKL, designed to conceal its presence, that recovers keystrokes by measuring the line state at any point between the keyboard and the PC.

2.2 Overview of PS/2 Protocol

The PS/2 bus consists of power (+5 V DC at 275 mA), ground, data, and clock lines [4]. During the idle state (i.e. when neither the keyboard or host is transmitting) the clock and data lines are kept at +5 V DC. The keyboard brings the data line low and then the clock line low to signal its intention to transmit. The low state corresponds to ground. The data line is sampled on the falling edge of the clock, which runs between 10–16.7 KHz (Fig. 1a). A passive, stealthy microcontroller-based keylogger, e.g. a self-powered variant of [17], would be connected to the ground, clock, and data lines and configured such that a downward voltage transition on the clock line triggers an interrupt routine in which the data line is sampled to determine whether a one or zero is being transmitted. Data concerning a keystroke is communicated to the host when a key is pressed and again when it is released.

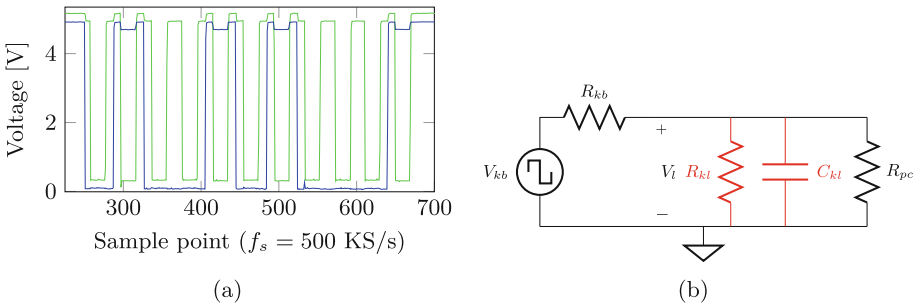


Fig. 1. (a) Electrical signal from the keyboard when the `SPACE` key is pressed (green: clock line; blue: data line; the clock is offset by 250 mV to aid visualization). Data is sampled by the host at the falling edge of the clock. (b) A passive HKL modelled in terms of its input capacitance C_{kl} and 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}) (Color figure online).

2.3 First-Order HKL Model

To understand the effects of connecting a HKL, and hence aid in our selection of features for detecting the presence of a keylogger, we modelled the HKL

as a first-order RC circuit (Fig. 1b). The model is meant to capture the non-zero capacitance, C_{kl}, and finite input resistance, R_{kl}, of a uC's I/O ports and suggests two ways in which a HKL may affect keyboard signaling.

The first is to notice that when data is transmitted the clock line goes from +5 V DC to 0 V DC for each bit; in the presence of a HKL this is roughly equivalent to what is known as the natural response of an RC circuit [26]. The act of bringing the clock line from the high to low state would ideally result in a fast downward drop of the line voltage, $V_l(t)$, however, with a keylogger present, and ignoring the keyboard output resistance for the moment, the line voltage will approach zero according to $V_l(t) = 5\exp(-t/\tau)$, where $\tau = (R_{kl} \parallel R_{pc})C_{kl}$. A similar analysis holds for when the clock is driven high (the step response). The presence of a HKL thus causes changes in the fall and rise time of the circuit.

The differences in fall/rise times in the absence and presence of a HKL, however, are likely to be small, as the parallel combination of R_{kl} and R_{pc} is likely large (on the order of kΩ) but the capacitance C_{kl} very small (on the order of pF), which leads to a time constant $\tau \sim$ ns. To confirm this we sampled the line voltage of a keyboard with and without a HKL at 40 GS/s using a Tektronix DPO7254C oscilloscope (see Sect. 4.2 for setup details). Figure 2a shows the rising portion of the first clock period without (blue) and with (red) a HKL (the figure is composed of an average of 100 time-aligned signals). Using the procedures set forth in [15] and these signals, fall/rise times were calculated without the HKL as $2.0333 \times 10^{-7}/1.3731 \times 10^{-6}$ s and 2.0350×10^{-7} /1.3782×10^{-6} s with. While the fall/rise times are indeed greater in the presence of the HKL, the difference is small; the record-to-record variation is also substantial, with 99 % confidence intervals of $1.9801 \times 10^{-7} \pm 7.5279 \times 10^{-9}$ s/$1.2760 \times 10^{-6} \pm 4.3898 \times 10^{-8}$ s without the HKL and $1.9804 \times 10^{-7} \pm 7.5868 \times 10^{-9}$ s/$1.2815 \times 10^{-6} \pm 5.1446 \times 10^{-8}$ s with. For these reasons, we ignore C_{kl} and examine the effects of R_{kl}, alone.[2]

We note that unless $R_{kl} \gg R_{pc}$, the voltage drop across the load (the PC) as seen by keyboard will be decreased by the parallel combination of R_{kl} and R_{pc} (Fig. 2b). This leads to a second way in which a HKL will perturb the system, namely a decrease of the voltage across the line, V_l. The proof is as follows.

In the absence of a HKL the line voltage is given as

$$V_l = \frac{R_{pc}}{R_{kb} + R_{pc}} V_{kb} \tag{1}$$

Allowing $R_{kl} = \beta R_{pc}$, the parallel combination $R_{eq} = R_{kl} \parallel R_{pc} = \frac{\beta}{1+\beta} R_{pc}$ results in a new line voltage

$$V_l' = \frac{R_{eq}}{R_{kb} + R_{eq}} V_{kb} = \frac{R_{pc}}{\frac{1+\beta}{\beta} R_{kb} + R_{pc}} V_{kb} \leq V_l \tag{2}$$

Eq. 2 is strictly less than Eq. 1 when $\beta \neq \infty$.

[2] In Sect. 5.3 we show that HKLs that do not affect line voltage—i.e. those with high input impedance—can still be detected because of their affect on the transient response of the system.

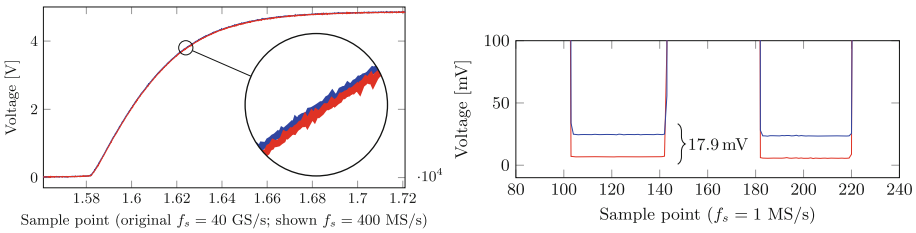


Fig. 2. (a) The rising portion of the first clock period of a keyboard's clock line. It takes the signal longer to transition to the low level when a HKL is present (red) than when it is not (blue); the same holds for the falling portion. (b) The voltage of the clock line with (red) and without a HKL (blue) for the lower portion of the first two clock periods. The level is less due to the loading effects of the HKL (Color figure online).

Given that both the PC and a uC-based HKL likely use the same transistor-level technology to measure the state of line, we take $\beta \approx 1$. Furthermore, to measure a voltage we would expect both the PC and HKL to present a very high resistance, while the keyboard, acting as a voltage source, would present a comparatively low resistance [26]. Assuming that R_{kl} and R_{pc} are approximately 1 MΩ and R_{kb} approximately 500 Ω the difference in the line voltage when a HKL is present at $V_{kb} = 5$ V would be $V_l - V_l' = 2.5\,\text{mV}$.

Figure 2b shows the lower portion of the first two clock periods of the clock line in the absence (blue) and presence (red) of a HKL (1000 records were time-aligned and averaged). As the figure indicates, the line voltage when a HKL is connected is indeed lower than when it is not and the difference is commensurate with the above calculation (the difference is also apparent and slightly greater for the upper portion of the signal). We also observed differences in the change of voltage for the HKL and no HKL case (i.e. $V_l - V_l'$) between keyboards, which can be explained by assuming that keyboard resistance, R_{kb}, differs between keyboards, where a lower R_{kb} leads to a smaller change in voltage. Similarly, the low voltage level of clocks for keyboards probably differs due to the fact that keyboards have different ground path resistances.

Finally, we note that our model assumes that the resistance and capacitance for a HKL are constant for all frequencies and line voltages, which, in general, is not the case. Given the low frequency of the PS/2 clock, frequency-dependent effects are apt to be slight. Changes in the resistance of the HKL, R_{kl}, for different line voltages could, however, be noticeable because of the constancy of the HKL's input port leakage current over a range of input voltages. For example, the maximum leakage current of a popular microcontroller is 1 µA over the input voltage range of [0,3.3] V [33]. A HKL built using this uC would present a resistance of 25 kΩ at 25 mV and 3.3 MΩ at 3.3 V. This suggests that in searching for the decreases in line voltage that signal the presence of a HKL, we should focus on the upper level of a signal, as by Eqs. 1 and 2, a larger relative drop would be produced for larger values of V_{kb}. The input-voltage dependency

of resistances also opens another avenue for possible detection: a HKL may be present if the observed deviation of the line voltage for the high and low levels of the clock is not equal.

3 Physical-Layer Detection of Keyloggers

Having proposed a mechanism by which a HKL may be detected, we introduce an anomaly detection architecture meant to leverage the mechanism to determine if a HKL has been attached. We describe its main components, including feature extraction and feature comparison. The extraction routine will focus on those areas of the signal most likely to display differences in the presence of a HKL, while the comparison routine will be sensitive to the slight changes our theory predicts will result from a HKL but still be robust to noise.

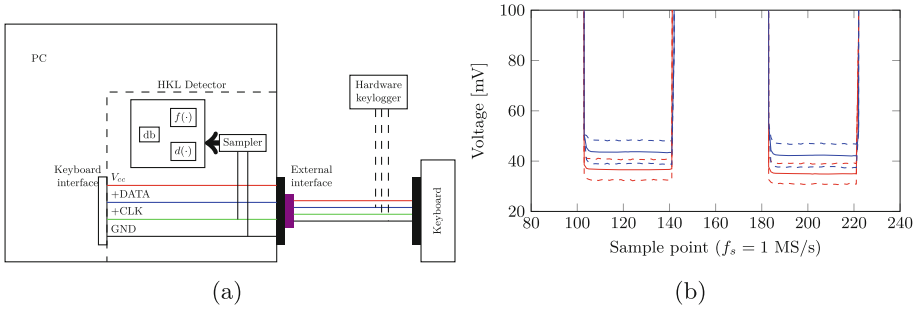
3.1 Proposed Architecture

To detect HKLs using the loading effects outlined above, we propose to incorporate a physical layer detection engine within the PC to perform anomaly detection based on the state of the clock line (Fig. 3a). The engine would be situated between the external keyboard interface of the PC and the internal keyboard interface so as to detect a HKL connected at any point between, or even inside, the PC and keyboard. The clock line is monitored because, while the data signal depends on the keypress, the clock signal is invariant with respect to the key being pressed; i.e. it is ubiquitous and repetitive. The detection engine consists of (1) a high-resolution analog-to-digital converter (ADC) or sampler to measure the clock line, (2) a routine $f(\cdot)$ that extracts features from the sampled data, (3) a metric $d(\cdot)$ by which to compare features of a newly sampled keypress to a baseline feature set, and (4) a database to store training and test data. Feature extraction and comparison are described in the following sections.

As the effect of a HKL on the line state amounts to a few millivolts or tens of millivolts decrease, it is necessary to employ a high-resolution sampler in the detector. By excluding transient effects—i.e. changes in fall and rise times—from the feature set, in addition to the fact that the PS/2 clock is less than 20 KHz, a comparatively low-speed ADC should prove sufficient. Given an ADC with an allowable input range of 0–5 V, a 12-bit ADC would achieve a resolution of $\approx$1.25 mV. Such an ADC can be had for as cheaply as $3.00 [2].

3.2 Feature Extraction

Our detection theory suggests, and is borne out by data, that a HKL will produce macroscopic effects on the line voltage. As such, it is sufficient to use the raw voltage measurements for features. We note that principal component analysis, factor analysis, or linear discriminant analysis could be used to reduce the



(a) (b)

Fig. 3. The proposed architecture for detecting hardware keyloggers at the physical layer. A sampler 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 a HKL. (b) The mean of two periods of the lower portion of the clock signal, using 1000 records, without (blue) and with (red) a HKL. The dashed lines give the 99 % confidence intervals for the means, which indicates that the line voltage for the two cases can be seemingly equivalent at times (Color figure online).

number of features or find the most powerful features in the future, though we did not find these techniques necessary to detect our HKL. As indicated in part by Fig. 2, we have found that a HKL affects the higher and lower levels of the clock to a different degree. Because of this, we have opted to extract samples from the lower and upper portions, and consider each set separately. The latter effect implies that it is only necessary to use a subset of the samples from each level for detection purposes. We use the same reasoning to justify the use of only the keydown portion of the keystroke for detection purposes.

The first step of our feature extraction procedure ($f(\cdot)$ in Fig. 3a) is to obtain the samples corresponding to the keydown portion of the keystroke from a record. To accomplish this an alignment routine takes the maximum of the correlation between a record and a reference signal for the keyboard to indicate the point in the record at which the reference is best aligned with the keydown signal, and then returns a contiguous subset of the record containing just the sample points encompassing the keydown clock signal. The reference signal consists of the keydown portion of a single record obtained in the absence of a HKL.

From the keydown portion of the record, roughly the first 1.5 periods of the clock (the entire first period and the upper half of second) are then used with Algorithms 1 and 2 to obtain the sample points of the lower and upper portions, respectively, of the truncated the clock signal. These sample points form two separate distributions to be used in our comparison function ($d(\cdot)$ in Fig. 3a), discussed next. The extraction procedure allows for the inclusion of some points belonging to the transient; this allows us to include transient effects not captured by our model but that could nonetheless serve as distinguishing features.

Algorithm 1. Extract lower level sample points from clock signal

Input : $\mathbf{R}$ (a sample point-by-record matrix of line measurements for the clock signal)
Output : S (sample points of $\mathbf{R}$ in the clock's lower level)

$S = \varnothing$;
foreach $R_i \triangleq R_{*,i} \in \mathbf{R}$ **do**
  $\{X \subset R_i : \forall x \in X < mean(R_i)\}$;
  $\{Y \subset R_i : \forall y \in Y \leq mean(X)\}$;
  $\{Z \subset R_i : \forall z \in Z \leq \mu(Y) + \sigma(Y)\}$;
  //$\mu(\cdot)$ and $\sigma(\cdot)$ compute the mean and standard deviation of the elements
  $S \leftarrow S \cup Z$;

Algorithm 2. Extract upper level sample points from clock signal

Input : $\mathbf{R}$ (a sample point-by-record matrix of line measurements for the clock signal)
Output : S (sample points of $\mathbf{R}$ in the clock's upper level)

$S = \varnothing$;
foreach $R_i \triangleq R_{*,i} \in \mathbf{R}$ **do**
  $\{X \subset R_i : \forall x \in X > mean(R_i)\}$;
  $\{Y \subset R_i : \forall y \in Y \geq mean(X)\}$;
  $\{Z \subset R_i : \forall z \in Z \geq \mu(Y) - \sigma(Y)\}$;
  //$\mu(\cdot)$ and $\sigma(\cdot)$ compute the mean and standard deviation of the elements
  $S \leftarrow S \cup Z$;

3.3 Feature Comparison

Figure 3b shows the mean (solid) and 99 % confidence intervals (dashed) of 1000 records acquired for a keyboard with a keylogger present (red) and in its absence (blue). The signals vary with respect to time and that individual signals with and without the HKL overlap, but that the means, and possibly the variances, are different when the HKL is connected compared to when it is not. Because of the overlap and variation observed, a simple distance metric, such as the Euclidean one, would require a large threshold to keep false positives low, but would also produce an unacceptable number of false negatives. To accommodate both variation and overlap we propose to use a distance metric designed for comparing distributions known as the *earth mover's distance* (EMD).[3]

Put simply, the EMD is a measure of the cost of transforming one histogram to another [27]. In our case, the sample points extracted from the lower, or upper, portion of the first 1.5 clock periods serve as the distribution, and we are interested in the cost of transforming the distribution of a record(s) when the line is in an unknown state to a baseline built for the keyboard in a known state (keylogger absent). If the cost is too high—i.e. if the distribution is too far from the baseline—we assert a HKL has been attached.

Specifically, considering samples from only one of the levels, we build a training distribution D_{trn} from the extracted features of a number of records procured in the absence of a HKL. A test distribution D_{tst} is then constructed from records collected from one or more keystrokes. To test for the presence of a HKL we employ the EMD: if $d(D_{tst}, D_{trn}) \leq T$, where T is a threshold, established to during a training phase, that results in an acceptable number of false positives, the records are said to have been acquired in the absence of a HKL. This procedure is followed for every keystroke or series of keystrokes.

The reference signal necessary to extract features from records and the training distribution for comparing those features to a threshold are stored in the database of our proposed detection engine.

[3] Properly speaking, we use a variant of the EMD for non-normalized histograms, where we have selected the l_1 norm for the ground distance metric [27].

4 Experimental Setup and Results

Results validating the first-order model given in Sect. 2 are presented. The HKL designed to conduct experiments on is explained and an overview of our experimental setup and procedures given.

4.1 Keylogger Design

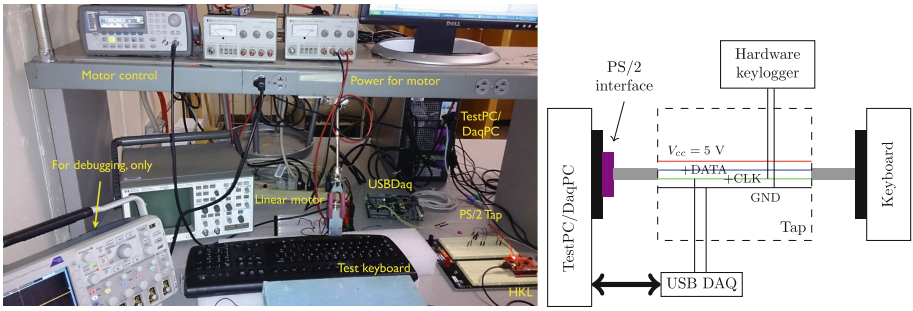
Our HKL was built using a Texas Instruments (TI) Tiva C Series TM4C123G LaunchPad, which is based on the TI TM4C123GH6PM microcontroller [32]. It is modeled on [17] (the only commercially available passive HKL we are aware of) and has similar specifications (e.g. the input leakage currents are the same order of magnitude). A passive HKL is *ipso facto* maximally evasive with respect to current active HKL detection methods. As our methodology relies only on observing deviations present on the clock line, we did not configure the uC to sample the data line or even connect it to the data line. One pin on the uC was set as an input and the uC was configured to issue an interrupt on the falling edge of the pin; an LED was blinked for each keystroke to verify proper operation.

In keeping with the premise of the work—to detect passive and stealthy keyloggers—the uC was powered using the USB bus, not the PS/2 bus; the input pin was also kept floating to maximize its impedance (i.e. R_{kl}) and make the HKL nominally evasive with regards to our detection approach. A floating input pin is generally discouraged as the input can be easily shifted by environmental factors such as noise, leading to spurious readings. It was felt, however, that activating the internal pull-up or pull-down resistors would affect the line voltage noticeably and therefore bias the experiments in favor of our approach.

4.2 Data Collection

Our experimental setup (Fig. 4a) consisted of a single PC for test and measurement purposes; i.e. the PC measured its own clock line voltage (mimicking our proposed architecture [Fig. 3a]). As the PC (a Dell Optiplex GX620) lacked a PS/2 port, a USB-to-PS/2 converter was used to connect the test keyboards. This had the side benefit of allowing us to attach a USB keyboard to control the system without interfering with the keyboard under test. To automate the data collection process a linear motor was setup to press the space bar every 1.2 s for 0.3 s (a 20 % duty cycle square wave with a period of 1.5 s was used with a switch to turn the motor on and off).

The line voltage was measured by connecting a sampler to a tap point midway between the two ends of the PS/2 cable (Fig. 4b). Our choice of sampler was a Measurement Computing USB-2500 Series DAQ board. The DAQ was configured to use a full-scale voltage of 10 V and sample at 1 MS/s. Given the board's 16 bit ADC, we were able to measure signals with a resolution of $\approx$153 µV. Upon detecting the first falling edge of the clock, the sampler would acquire data for the next 35 ms. This sampling period allowed us to capture the clock for both



(a) (b)

Fig. 4. (a) Experimental setup: the keyboards were secured in place so that the linear motor struck approximately the same place on the spacebar for each keyboard. (b) A schematic diagram of the setup. The dashed box represents an electrical tap in the PS/2 cable that was created by cutting the cable, striping the wires, and then soldering the exposed wires to binding posts.

the keydown and keyup signals sent by the keyboard, though only the clock corresponding to the keydown press was used in our analysis.

We collected data from a total of 25 keyboards, consisting of eight different models, 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, followed by another 1000 keystrokes with the keylogger attached. It took approximately 50 min per keyboard to acquire both sets of data.

The reference signal, used for aligning signals in the feature extraction procedure (Sect. 3.2), for each keyboard was obtained from the first record captured for the keyboard without the HKL. Because the clock signal does not vary substantially from keyboard-to-keyboard, the negative (falling) threshold-based trigger that was set on the sampler to detect the beginning of the clock would consistently initiate the sampling sequence at nearly the same point of the clock signal. This enabled us to use the same set of sample points for the reference signal and extraction of the first 1.5 periods of the clock, again, described in Sect. 3.2, for each keyboard.

4.3 Discussion

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 23.7 mV for the upper level of the clock and 4.11 mV for the lower level. We attribute the difference in the voltage drop between the two levels to a change in the input resistance. Indeed, according to [33] the nominal and maximum leakage currents of the uC at 5 V are 30 µA and 60 µA, respectively, while at 50 mV they are 1 nA and 1 µA. This suggests that R_{kl} =[83.3 kΩ,166.6 kΩ] at 5 V while at 25 mV, R_{kl} =[25 kΩ,25 kΩ]. Using the

maximum leakage currents and assuming a $\beta = 1$ with $V_{kb} = 500\Omega$ the predicted differences, by (1) and (2), would amount to 30 mV and 471 µV for the high and low level, respectively.

While the observed drop in line voltage for the high level roughly corresponds to the predicted drop, the lower level differs by an order of magnitude. The observed drop for the lower level could be explained if the leakage current were 10 µA, which would produce an expected drop of 3.4 mV. The documentation for the uC ([33], p. 4) suggests that the leakage current for *most* GPIO pins is less than 1 µA so perhaps the GPIO pin used in our HKL has a higher than average leakage current. Another possibility is that, as the datasheet indicates, for input voltages between –0.3 V to 0 V the maximum leakage current is given as 10 µA. Mismatches in the internal biasing of the uC due to the use of a separate power supply for the uC, intended to maintain the HKL's stealth, could conceivably make the input appear in this range to the uC.

To detect the differences in the line state, distances between a training distribution, built for each keyboard from fixed a number of records, and test distributions based on varying numbers of records were computed using the EMD metric. Individual training distributions for the keyboards were built from 25 randomly selected records captured without the keylogger attached. The EMD implementation we used requires that the number of sample points in the training and test distributions be equal. To satisfy this requirement we removed randomly selected samples from the larger distribution to make it equal in size to the smaller distribution. For all the test cases—i.e. whatever the number of records used to build the test distribution—the maximum number of sample points used was limited to 256 in order to keep the EMD calculation tractable.

Table 1. The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution fixed at 25 records). The left part of the table gives results for distributions built using the lower clock level while the right gives results for the upper clock level. We are able to reliably detect the presence of the HKL, for all 25 keyboards, after 25 keystrokes by observing the lower level and only 10 keystrokes by observing the upper. Sample points is the nominal number of sample points used in the EMD calculation.

N	EER (%)			T			Sample	EER (%)			T			Sample
	mean	max	median	mean	max	min	points	mean	max	median	mean	max	min	points
1	7.56	31.6	2.8	0.001	0.004	0.001	34	2.42	8.40	2.2	0.016	0.125	0.006	32
2	2.92	13.6	0.6	0.002	0.008	0.001	68	0.67	3.20	0.4	0.039	0.250	0.019	65
3	1.86	16.5	0	0.004	0.016	0.002	104	0.22	1.20	0	0.064	0.500	0.031	97
4	0.98	6.4	0	0.004	0.016	0.003	135	0.12	0.08	0	0.084	0.500	0.031	129
5	0.72	7.0	0	0.006	0.031	0.004	167	0.06	1.50	0	0.104	0.500	0.057	161
10	0.32	5.0	0	0.011	0.063	0.004	256	0	0	0	0.173	0.500	0.063	256
15	0.24	4.6	0	0.012	0.063	0.004	256	0	0	0	0.178	0.500	0.125	256
20	0.16	4.0	0	0.012	0.063	0.004	256	0	0	0	0.193	0.500	0.063	256
25	0	0	0	0.012	0.063	0.004	256	0	0	0	0.175	0.500	0.125	256

To evaluate the efficacy of our approach, we calculated the equal error rate (EER) on a per keyboard basis. Table 1 reports the average, maximum, and median (the minimum was always zero) EER for test distributions built from $N = \{1, 2, 3, 4, 5, 10, 15, 20, 25\}$ consecutive records (training/test distributions built from the lower level on the left and the upper level on the right). As we observed a larger voltage drop for the upper level of the clock, we anticipated that it would be easier to detect the HKL at the higher voltage, and indeed this was so. However, even with the small differences observed at the lower level, our approach is able to reliably detect (i.e. achieve an EER = 0) the presence of the HKL after 25 keystrokes, while for the upper level this same feat is achieved with only 10 keystrokes. Figure 5a and b show the distances between training and test distributions, using $N = 10$ for the high level and $N = 25$ for the low level comparisons, along with their respective EER thresholds.

We were able to further lower the number of keystrokes needed to detect the keylogger to five by fusing the outputs of the upper and lower distance calculations using unanimous voting. That is, for a set of records to be declared free of the HKL, the distances for both the upper and lower level distributions would need to fall within their respective thresholds. To evaluate the fusion of the distance tests, we established the thresholds needed to guarantee zero false-positives for each test distribution. Thus, the keylogger could be detected if the distance for either test distribution built from records captured with the HKL attached was greater than the specified thresholds. Zero false negatives were achieved when $N = 5$ for both high and low level distributions.

5 Feature Stability and Countermeasures

It is shown that while the features used for HKL detection are dependent on the environment, this dependency can be modelled and thus accounted for. Attacker countermeasures, both active and passive, are also considered and neutralized.

5.1 Stability of Features

The variability apparent in Figs. 3b, 5a, and b suggests that the line voltage is a stochastic process. This begs the question: can we track the state of the line using training data acquired at an earlier time? In an attempt to offer a partial answer to this question we acquired a second round of data without the keylogger attached and used the training distributions for the first dataset to calculate the distance between the two. We found that the distances calculated using the upper clock level were within the thresholds established for the earlier dataset; i.e. were able to successfully re-identify that the line was not encumbered with the HKL. In the case of the lower level, however, the distance between the training and test distribution were greater than the previously established thresholds; i.e. we falsely identified the line as having the HKL attached.

We hypothesize that our inability to track the lower line voltage is due to temperature-induced variations, as such small voltages ($\approx$25 mV for the lower

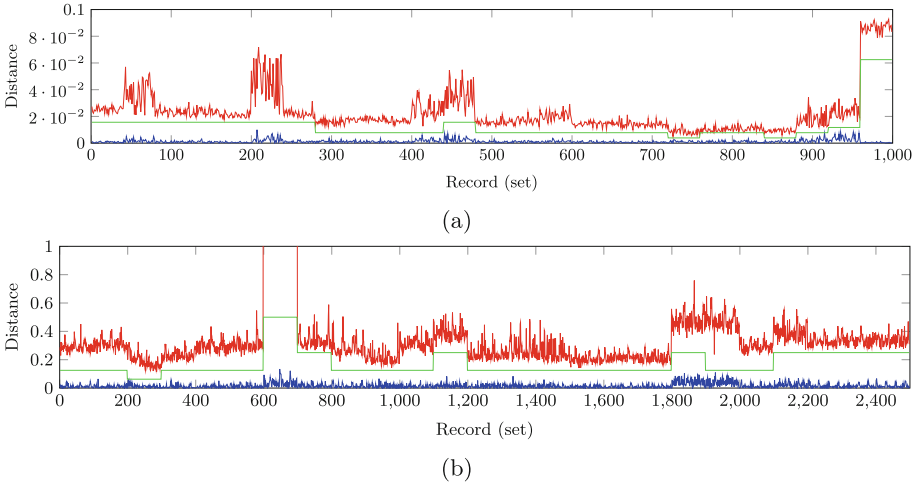


Fig. 5. The earth mover's distance between a training distribution and a test distribution built from records without a keylogger attached (blue) and with a keylogger attached (red) for all 25 keyboards (x-axis; records are grouped). EER thresholds shown in green. Since there is no overlap we are able to detect the HKL. (a) Features extracted from lower level of clock with $N = 25$ and (b) features extracted from upper level of clock with $N = 10$. The spike in the distance for records 601–700 results from a faulty keyboard (Color figure online).

clock level) could be shifted by thermal noise over time. However, temperature-induced changes to the line voltage could be compensated for by employing noise models to equalize line measurements taken at different temperatures. This implies that temperature readings need to be recorded when data is taken in order to take into account the discrepancy between the temperature of new data and the temperature at which the training data was acquired.

To test the above hypothesis, we performed an experiment wherein temperature sensors (the TI LM35DT [31]) were placed next to four suspected points of influence; viz. the keyboard under test, the Measurement Computing DAQ board, the site of PS/2 cable tap, and the uC-based HKL. A National Instrument USB-6008 series DAQ was used to record the output of the sensors. Every 30 s for 24 h[4] a key was pressed and the output of the sensors were measured 100 times, in addition to the voltage of the PS/2 clock line. Using 20 % of the captured data, selected at random, for each of the 23 keyboards as training data we performed a linear regression on the remaining 80 % using the model $F \sim 1 + T$,

[4] A slight change was made to our experimental setup to accommodate the duration of the data runs. Instead of the space bar being manually pressed, a program was written that toggled the `NUMLOCK` state. Since the OS state of this key and the `NUMLOCK` LED must be consistent, the PC would signal the keyboard that it had a scancode to send by bringing the clock line low, which would then cause the keyboard to generate a clock signal that we were able to subsequently capture.

where T denotes the average measured temperature during which the record was captured and F the mean of the lower portion of the clock signal of the record (our feature of interest). An average $R^2 = 0.99$ indicates that the line voltage is a strong function of temperature; the sensor that provided the best fit was nearest the PS/2 tap point.

5.2 Active and Evasive Keyloggers

In our threat model (Sect. 2.1) we pointed out that active keyloggers that rely on GPIO ports to capture keystrokes from the keyboard and then replay them to the PC should be easily detectable as the input/output ports do not capture the nuances of the keyboards signaling. In this section we argue that even a specially built keylogger that took pains to accurately measure and reproduce the keyboard's signals would be unlikely to remain undetected.

It has been demonstrated [6,9] that, under some circumstances, physical layer identification systems are vulnerable to an attacker replaying a signal from a device using an arbitrary waveform generator (AWG) or digital-to-analog converter (DAC). In the experiments carried out in these works, an attacker acquires a digital copy of a device's signal using an analog-to-digital converter (ADC) and then reproduces it using a DAC. As most uCs are equipped with an ADC and DAC (or can be easily outfitted with them), we could imagine an attacker attempting to mount a similar attack on our proposed PLD system by creating an active HKL that samples the keyboard's signal using the ADC and then replays it to the PC using the DAC. Leaving aside the exact characteristics of each converter necessary to carry out the attack (sampling rates and resolution, chiefly), we point out that the ADC/DAC would still cause loading effects that would make it detectable.

Firstly, the finite resistance associated with the ADC input would cause a drop in the line voltage, which would mean that an attacker cannot know the true value of the keyboard's output. Secondly, the non-zero output impedance of the DAC would cause a decrease in the voltage measured by the PC (this can be seen by the replacement of V_{kb} and R_{kb} with V_{dac} and R_{dac} in Eq. 1). Now, the attacker could attempt to compensate for these loading effects by calibrating the HKL to the system they wish to attach it to. However, this procedure is quite invasive, and noticeable, as it requires that the attacker obtain the resistances of the PC and keyboard. The measurements required to deduce these values require that both the PC and keyboard be powered, as their port impedance would change in the absence of power. To accomplish this would require that the attacker sever at least the clock line between the two, which our PLD could be programmed to notice.

Additionally, we note that measuring and replaying the line state continuously using an ADC/DAC would be more energy intensive than simply measuring and replaying the binary state of the line via GPIOs, leading to a shorter period of keylogging. Also, a simple active HKL would cut off bi-directional communication between the PC and keyboard as the replay is one-way. It may be possible to design an HKL that senses the keyboard taking control of the line,

but this seems nontrivial and could introduce delays in signal propagation that are detectable.

Finally, it may also be possible to detect/counter a self-powered active HKL employing a DAC by shorting the keyboard line. The short works as a counter because a stealthy HKL will have a limited power supply, as it is self-powered, so drawing the maximum amount of current possible via a short would increase its power consumption and decrease its operational lifetime. In addition, the DAC could probably not sustain a significant current draw without damage. Detection is also possible using a short: as the keyboard draws its power from the PC a short should result in a spike of current on the V_{CC} line, of a known amount. The presence of a HKL could be deduced by the absence of such a spike, or a spike of equivalent magnitude.

5.3 Passive and Evasive Keyloggers

As noted in Sect. 2.3, a HKL can be detected due to differences in fall/rise time (transient response) or voltage drops. In this work we focused on voltage drops because transient effects are small for the HKL we considered (time constant on the order of a nanosecond). An attacker attempting to evade our level-based detection approach could equip their HKL with a high input impedance comparator, based on the LT1793 op-amp [22], for example, at the input stage to ensure an undetectable voltage drop. With an input impedance of 10 TΩ, such an op-amp would effectively make $R_{eq} = R_{pc}$ (order of MOhm), which would result in $V_l = V_l'$. A high input impedance comparator would, however, produce a time constant ($\tau = R_{eq}C_{kl}$, $C_{kl} \sim$ pF) on the order of microseconds, which would distort the clock voltage to a noticeable degree (due to changes in the rise/fall time of the circuit). This does beg the question: can an attacker select a R_{kl} such that the transient response is unchanged and the drop immeasurable? We would argue no, as follows.

Assume that an attacker can arbitrarily set the resistance of the HKL. It is the attacker's prerogative to select an R_{kl} that produces an equivalent resistance as small as possible (to minimize the time constant), yet large enough so that the resulting voltage drop across R_{eq} is less than can be resolved by the ADC. The minimum equivalent resistance to accomplish this is

$$R_{eq} = \frac{V_l - r}{V_{kb} - V_l + r} R_{kb} \tag{3}$$

where r is the minimum resolvable voltage drop (see Appendix for derivation). For the ADC used in the paper $r = 150\,\mu\text{V}$, which yields $R_{eq} = 943\ \text{k}\Omega$. Ignoring the capacitance of the additional resistors needed to effect the target resistance, the capacitance of a LT1793 op-amp is 1.5 pF, which produces a time constant on the order of microseconds. Even an $r = 1\,\text{mV}$ requires $R_{eq} = 714\ \text{k}\Omega$, which still produces a time constant on the order of a microsecond. Additionally, attaching the op-amp to the clock line is likely to produce more than 1.5 pF of capacitance.

To validate the above claim we replaced our HKL with a resistor (representing R_{eq}) and 3 pF capacitor and acquired 1000 clock line measurements for each

keyboard; fresh comparison data without the resistor and capacitor was also collected. The resistor value was selected experimentally for each keyboard such that the resulting voltage drop could not be detected by our Measurement Computing DAQ. On average, the minimum equivalent resistance for our collection of keyboards was 6 MΩ; i.e. an attacker able to tune the input resistance of their HKL to 6 MΩ would ensure that it is undetectable to our level-based approach, while at the same time minimizing the time constant (and hence rise/fall times) of the circuit. The 3 pF capacitance was used to represent the capacitance of the HKL input pin and the connection to the PS/2 line. An estimate of 3 pF was made as a lower bound based on the assumption that the HKL would be mounted on a printed circuit board (PCB) to accomodate lower capacitance surface mount components, which introduces parasitic capacitances due to the groundplane (0.5 pF cm^{-2} [30]), traces (0.8 pF cm^{-1} [30]), and bondwires (0.1 pF to 0.15 pF for 2 mm wire lengths [16]). Our detection approach consisted of extracting the rise and fall times (calculated according to [15]) of the first five edges of the portion of the clock relating to the down keypress. Instead of the EMD, which was found to be unable to distinguish between the HKL and no HKL cases, the Kullback-Leibler (KL) divergence [21] was employed for comparing training and test distributions.

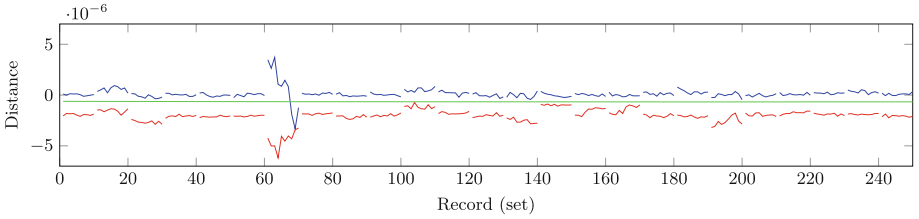


Fig. 6. The Kullback-Leibler divergence between a training distribution and a test distribution built from records without a keylogger attached (blue) and with a keylogger attached (red) for 23 keyboards (x-axis; records are grouped). EER thresholds shown in green. Since there is no overlap, aside from keyboard seven, we are able to detect the HKL. Features consist of the rise times for the first five rising edges of the clock with $N = 100$. The spike in the distance for records 61–70 results from a faulty keyboard (Color figure online).

Given expected rise/fall times on the order of a microsecond, we did not expect to be able to discern a difference in rise/fall times using a 1 MS/s ADC. As such, a Tektronix DPO2024 oscilloscope equipped with a Tektronix P6139B probe (10 MΩ and 8 pF input resistance and capacitance, respectively) was used. Using a sampling rate of 125 MS/s[5] with 25 keystrokes for training/detection using the rise time resulted in an average ERR of 0.02 %, while 100 keystrokes for

[5] We note that while 125 MS/s ADCs are more expensive than the 1 MS/s variety, they can still be had for less than $15, e.g. the LTI LTC2251 [23].

Table 2. The equal error rate, and corresponding thresholds, achieved using N records to build the test distribution (training distribution fixed at 25 records). The left part of the table gives results for distributions built using the fall times of the first five falling edges, while the right gives results for distributions built from the rise times of the first five rising edges. We are able to reliably detect the presence of the HKL, for most keyboards, after 50 keystrokes by observing the fall times and only 25 keystrokes by observing the rise times.

N	EER (%)			T			EER (%)			T		
	mean	max	median	mean	max	min	mean	max	median	mean	max	min
1	4.59	9.98	4.31	−0.74	−0.66	−0.86	3.75	9.69	3.51	−6.04	−2	−12
2	2.73	5	3.16	−0.08	−0.07	−0.19	2.11	4.94	1.93	−2.4	−1.5	−4.5
4	1.12	2.5	1.28	−0.26	−0.05	−0.45	0.7	2.24	0.61	−3.62	−0.5	−7.5
5	1	1.93	1.15	−0.39	−0.03	−0.93	0.54	1.68	0.37	−5.82	−3.5	−9.5
10	0.39	0.99	0.41	−0.275	−0.1	−0.45	0.15	0.81	0.53	−8	−7	−9
20	0.10	0.5	0.04	−1.5	−1	−2.5	0.05	0.42	0.01	−17	−11	−26
25	0.09	0.4	0.06	−1.6	−0.6	−2.6	0.02	0.18	0	−21.5	−14	−29
50	0.02	0.1	0	−2.8	−1.8	−3.8	0.03	0.08	0	−32.5	−30	−35
100	0.01	0.1	0	−6.1	−5.7	−6.5	0.001	0.03	0	−64.5	−62	−67

training/testing yielded an average EER of 0.001 % (Fig. 6). This suggests that either an increase in the sampling rate of the ADC or the number of keystrokes used for detection would be sufficient to detect the presence of a HKL designed to evade a level-based detection approach. We note that for all of the keyboards considered, using both rise and fall times, the HKL stand-in was eventually and definitely detected; i.e. the KL divergence for the resistor/capacitor samples were significantly greater than the largest distance for the non-resistor/capacitor samples (Table 2).

6 Conclusion and Future Work

Inspired by the observer effect, we hypothesized that a HKL would have a measurable effect on its host system. Specifically, we built a detection methodology based on the theory that the HKL would cause the voltage of the clock line to drop. This prediction was substantiated through experiments wherein it was shown that 25 keystrokes were necessary to identify the presence of a HKL when the lower level of the clock was used for detection, while the upper level required only 10 keystrokes and was shown to be more consistent across time. A combined approach based on unanimous voting reduced the detection time to five keystrokes. It was found that the features used to identify the presence of a HKL are sensitive to temperature. Furthermore, it was shown experimentally that an attacker cannot escape detection by modifying the input resistance of the HKL, if the transient characteristics of the clock line are monitored.

Future work includes the long-term observation of keyboard signals to understand and incorporate the effects of ageing. Adaptive thresholding schemes may prove useful in this regard. Secondly, to complement detection, investigations of

active countermeasures against HKL should be undertaken, including the permanent disabling of HKLs through electrical means. Finally, research should be undertaken to identify features that are not based on the clock signal level. The ultimate aim of this work should be to discover features in the keyboard signaling that are sensitive to the presence of a HKL but invariant with respect to the keyboard resistance/voltage and the PC resistance.

Acknowledgements. The authors would like to thank Li Yin and Heidi Harper of Utah State University for their assistance in collecting data.

Appendix: Optimal Selection of HKL Input Resistance

The attacker seeks to minimize the difference between the line voltage with and without the HKL in order to evade the level-based detection approach, while simultaneously minimizing the time constant associated with the HKL to lessen the increase of the rise/fall times of the clock signal. The former goal can be realized by choosing $R_{kl} \gg R_{pc}$ to ensure that $R_{eq} = R_{kl} \parallel R_{pc} = R_{pc}$. This, however, is achieved at the expense of the latter goal, as the time constant $R_{eq}C_{kl}$ can only be decreased by selecting R_{kl} such that $R_{eq} < R_{pc}$, due to the fact that the HKL capacitance is fixed. The minimum value of R_{eq}, and by extension the optimal input impedance of the HKL, necessary to evade the level-based approach while minimizing the time constant of the HKL is calculated as follows.

Allow r to represent the minimum resolvable voltage drop of the ADC employed in the detector. Evading the level-based detection approach requires $V_l - V_l' = r$, where r may be expressed in terms of the quantities controllable and/or known by the attacker as

$$r = V_l - \frac{R_{eq}}{R_{kb} + R_{eq}} V_{kb} \tag{4}$$

Defining

$$V_m = \frac{R_{eq}}{R_{kb} + R_{eq}} V_{kb} \tag{5}$$

and rearranging terms yields

$$V_l - r = V_m \tag{6}$$

Furthermore, manipulation of (5) gives

$$R_{eq} = \frac{V_m}{V_{kb} - V_m} R_{kb} \tag{7}$$

By substituting (6) into (7) we arrive at

$$R_{eq} = \frac{V_l - r}{V_{kb} - V_l + r} R_{kb} \tag{8}$$

□

References

1. ABC News: Former Cal State student gets year in prison for rigging campus election (2013). http://abcnews.go.com/US/cal-state-student-year-prison-rigging-campus-election/story?id=19682401
2. Analog Devices: AD7265 Differential/Single-Ended Input, Dual 1 MSPS, 12-Bit, 3-Channel SAR ADC (2006), datasheet
3. Chahrvin, S.: Keyloggers–your security nightmare? Comput. Fraud Secur. **2007**(7), 10–11 (2007)
4. Chapweske, A.: The ps/2 mouse/keyboard protocol (2003). http://www.computer-engineering.org/ps2protocol
5. Danev, B.: Physical-layer Identification of Wireless Devices. Ph.D. thesis, ETH Zurich, Zurich, Switzerland (2011)
6. Danev, B., Luecken, H., Capkun, S., Defrawy, K.E.: Attacks on physical-layer identification. In: Proceedings of the Third ACM Conference on Wireless Network Security (WiSec 2010), pp. 89–98. ACM, New York (2010)
7. Danev, B., Zanetti, D., Capkun, S.: On physical-layer identification of wireless devices. ACM Comput. Surv. (CSUR) **45**(1), 6 (2012)
8. Daniels, T.E., Mina, M., Russell, S.F.: A signal fingerprinting paradigm for physical layer security in conventional and sensor networks. In: Proceedings of the International Conference on Security and Privacy for Emerging Areas in Communnication Networks (SecureComm), pp. 219–221. IEEE Computer Society (2005)
9. Edman, M., Yener, B.: Active attacks against modulation-based radiometric identification. Technical report, Rensselaer Polytechnic Institute, Department of Computer Science (2009), technical Report
10. Erbskorn, J.W.: Detection of Intrusions at Layer ONe: The IEEE 802.3 normal link pulse as a means of host-to-network authentication A preliminary performance analysis and survey of environmental effects. Master's thesis, Iowa State University, Ames, IA (2009)
11. Gerdes, R., Mina, M., Russell, S., Daniels, T.: Physical-layer identification of wired ethernet devices. IEEE Trans. Inf. Forensics Secur. **7**(4), 1339–1353 (2012)
12. Gerdes, R.M.: Physical layer identification: methodology, security, and origin of variation. Ph.D. thesis, Iowa State University, Ames, IA (2011)
13. Gerdes, R.M., Daniels, T.E., Mina, M., Russell, S.F.: Device identification via analog signal fingerprinting: a matched filter approach. In: Proceedings of the Network and Distributed System Security Symposium (NDSS). The Internet Society (2006)
14. Greene, M., Parker, M.: Method and system for detecting a keylogger that encrypts data captured on a computer, 25 July 2006, US Patent App. 11/492,581
15. IEEE: Standard for transitions, pulses, and related waveforms (2011), IEEE Std 181–2011
16. Karim, N., Agrawal, A.: Plastic packages electrical performance: reduced bond wire diameter. In: NEPCON WEST, pp. 975–980 (1998)
17. KeeLog: Open source DIY hardware keylogger (2012). http://www.keelog.com/diy.html
18. KeeLog: Keygrabber Module (2013). http://www.keelog.com/
19. KeyCarbon: Keycarbon Raptor (2012). http://www.keycarbon.com/
20. KeyCarbon: Keycarbon PCI (2013). http://www.keycarbon.com/
21. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **52**, 79–86 (1951)
22. Linear Technology: LT1793 JFET Input Op Amp (1999), datasheet

23. Linear Technology: LTC2251/LTC2250 ADCs (2005), datasheet
24. Mihailowitsch, F.: Detecting hardware keyloggers, November 2010. https://deepsec.net/docs/Slides/2010/DeepSec_2010_Detecting_Hardware_Keylogger.pdf. [DeepSec 2010 Presentation]
25. Nakra, B.C., Chaudhry, K.K.: Instrumentation Measurement and Analysis. McGraw-Hill Education (India) Pvt Limited (2009)
26. Nilsson, J.W., Riedel, S.: Electric Circuits. Prentice Hall, Upper Saddle River (2010)
27. Pele, O., Werman, M.: A linear time histogram metric for improved SIFT matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 495–508. Springer, Heidelberg (2008)
28. Salkind, N.: Encyclopedia of Research Design. SAGE Publications, Thousand Oaks (2010)
29. Sapra, K., Husain, B., Brooks, R., Smith, M.: Circumventing keyloggers and screendumps. In: 2013 8th International Conference on Malicious and Unwanted Software: "The Americas" (MALWARE), pp. 103–108, October 2013
30. Texas Instruments: High Speed Analog Design and Application Seminar: High Speed PCB Layout Techniques (2004), presentation
31. Texas Instruments: LM35 Temperature Sensors (2013), datasheet
32. Texas Instruments: Tiva TM4C123GH6PM microcontroller (2013), datasheet
33. Texas Instruments: Use conditions for 5-v tolerant gpios on Tiva C series TM4C123x microcontrollers (2013), application Report
34. The New York Times: Credit card data breach at Barnes & Noble stores (2012). http://www.nytimes.com/2012/10/24/business/hackers-get-credit-data-at-barnes-noble.html?_r=3&
35. Zaitsev, O.: Skeleton keys: the purpose and applications of keyloggers. Netw. Secur. **2010**(10), 12–17 (2010)

http://www.springer.com/978-3-319-26361-8

Research in Attacks, Intrusions, and Defenses
18th International Symposium, RAID 2015, Kyoto, Japan,November 2-4, 2015. Proceedings
Bos, H.; Monrose, F.; Blanc, G. (Eds.)
2015, XIII, 638 p. 147 illus. in color., Softcover
ISBN: 978-3-319-26361-8