Keyboard Acoustic Emanations

Dmitri Asonov Rakesh Agrawal
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120, USA
{*dasonov,ragrawal*}*@us.ibm.com*

Abstract

We show that PC keyboards, notebook keyboards, telephone and ATM pads are vulnerable to attacks based on differentiating the sound emanated by different keys. Our attack employs a neural network to recognize the key being pressed. We also investigate why different keys produce different sounds and provide hints for the design of homophonic keyboards that would be resistant to this type of attack.

1. Introduction

Emanations produced by electronic devices have long been a source for attacks on the security of computer systems. Past attacks have exploited electromagnetic emanations [6] as well as optical emanations [10, 13]. Acoustic emanations have also been explored. For example, it is shown in [7] that the acoustic emanations of matrix printers can carry substantial information about the text being printed.

We investigate acoustic emanations of a PC keyboard, the clicks, to eavesdrop upon what is being typed. This attack is based on the hypothesis that the sound of clicks can differ slightly from key to key, although the clicks of different keys sound very similar to the human ear. Our experiments show that a neural network can be trained to differentiate the keys to successfully carry out this attack.

This attack is inexpensive and non-invasive. It is inexpensive because in addition to a computer, the only other hardware required is a parabolic microphone. It is non-invasive because it does not require physical intrusion into the system; the sound can be recorded from a substantial distance. We, therefore, also investigate what can be done to thwart this attack.

In addition to the PC keyboards, we also study attacks on notebook computers, touchtone telephones, and ATM keypads. Our experiments suggest that what is being typed on these devices can also be compromised using an attack based on the sounds produced by clicks.

1.1. Paper Layout

Section 2 presents the details of the attack. We explain how we extract features from the raw acoustic signal produced by the click of a key on the PC keyboard. These features are then used to train the neural network for differentiating the keys. We first show the effectiveness of the attack in distinguishing two keys and then extend the attack to cover multiple keys.

We show that the differences in typing style have little impact on the ability of the network to recognize the keys. This means that the network can be trained on one person and then used to eavesdrop on another person typing on the same keyboard. We also show that it is possible to train the network on one keyboard and then use it to attack another keyboard of the same type, albeit there is a reduction in the quality of recognition.

In Section 3, we examine the physical characteristics of a keyboard that cause the attack to succeed. Specifically, we determine why different keys produce different sounds. These insights provide clues for the design of homophonic keyboards that would be resistant to this type of attack.

In Section 4, we study the vulnerability of different types of push button input devices to the proposed attack. We discuss related work in Section 5 and conclude with a summary in Section 6.

2. The Attack

The proposed attack is based on the hypothesis that the sound of clicks might differ slightly from key to key, although the clicks of different keys sound similar to the human ear. We employ a neural network to classify clicks. We chose to use neural networks for this task as they have been successfully used in solving related problems, such as speaker identification [18].

2.1. Experimental Set-up

We first specify the equipment and the software used in our study.

Keyboards. We used several types of keyboards. Most of PC keyboard experiments were performed with an IBM keyboard S/N 0953260, P/N 32P5100. Experiments with multiple keyboards were performed with three GE Power keyboards HO97798. For experiments with telephones, Siemens RP240 phones (M/N 62001) were used.

Microphones. We used a simple PC microphone for short distances up to 1 meter and a parabolic microphone for eavesdropping from a distance.

Computer omnidirection microphone: serial number $33-3026$ manufactured by RadioShack; frequency response: 30 Hz–15 kHz; impedance: 1000 ohms $\pm 30\%$; sensitivity: -68 dB ± 3 dB; operating voltage: 1.0 to 10 VDC.

Parabolic microphone: 'Bionic Booster' manufactured by Silver Creek Industries; frequency response 100 Hz–10 kHz (-3 dB response); gain amp. cut off at 90 dB; overall system gain: 40 dB; sensitivity: -46 dB (0 dB = 1 V/Pa).

ADC and FFT. The input was digitized using a standard PC sound card with 44.1 kHz sampling rate. Sigview software v.1.81 was used for recording the sound as well as for calculating time-FFT on 2 ms windows, with the Hanning windowing function applied. Window overlap was not available.

Neural Network. We used the JavaNNS neural network simulator [23] to build a backpropagation neural network. The number of input nodes equaled the size of the feature. For example, one value per 20 Hz in the FFT requires 200 input nodes for 0–4 kHz interval. There were 6–10 hidden nodes, depending on the size of the feature and the number of keys. The number of output nodes equaled the number of keys in the experiments with multiple keys. One output node was used in the experiments with two keys.

2.2. Training the Neural Network

The raw sound produced by key clicks is not a good input for training a neural network. Neural networks are recommended to be trained with an input consisting of several dozens to several hundreds of numeric values between 0 and 1 [19], which corresponds to approximately up to 1 kB input. On the other hand, the size of the acoustic signal corresponding to a keyboard click is about 10 kB. We, therefore, extract relevant features from the raw sound.

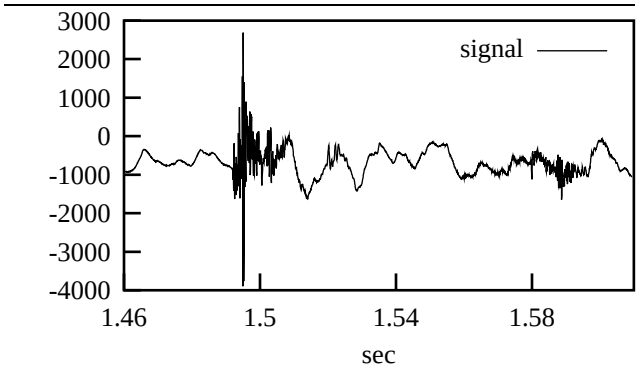

Figure 1. The acoustic signal of one click.

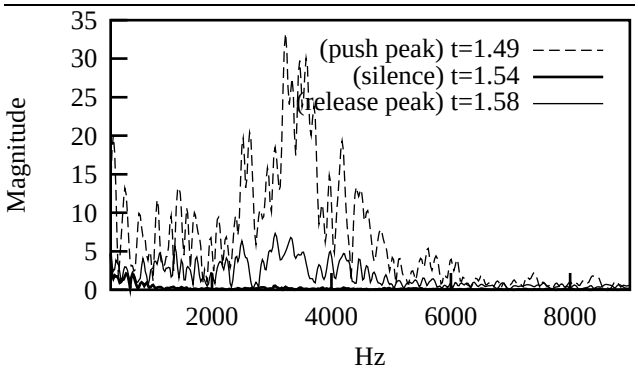

Figure 2. Frequency spectrums corresponding to the push peak, a silence interval, and the release peak.

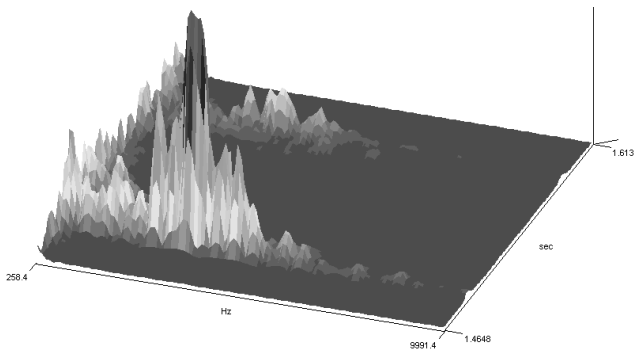

Figure 3. Time FFT of the signal in Figure 1.

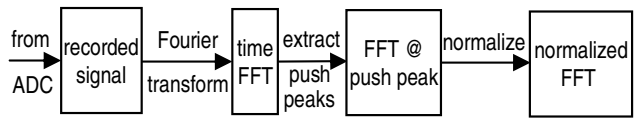

Figure 4. Feature extraction.

kHz	0–9	.3–3.4	0–3	1–4	2–5	3–6	4–7	5–8	6–9
ADCS	1.65	2.70	2.76	3.45	4.36	3.94	5.05	5.94	7.70

Table 1. ADCS value for [0:9] kHz, radio band, and shifting 3 kHz intervals.

We want the features that enable the neural network to differentiate between perceptually similar sound samples. The direct frequency spectrum is known to have significant variation for perceptually similar sounds [8], which makes it particularly attractive for our application. Interestingly, it is this same property of the direct frequency spectrum that causes it not to be used as a feature in the conventional sound classification [8].

We also need to carefully choose the time at which the spectrum is calculated. For this purpose, an understanding of how the signal of a click looks like is instructive. As shown in Figures 1, 2 and 3, the click lasts for approximately 100 ms, and the acoustic signal has two distinct peaks corresponding to pushing the key and releasing the key. There is relative silence between the push and release peaks.

The frequency distribution is best exposed at the peaks. We calculate the frequency distribution at the time of the press peak because the release peak is considerably lower. After calculating the frequency distribution at the press peak, we normalize the vector so that the values in the spectrum fall in the range [0,1] required for a neural network.

Initially, we used the FFT [19] extracted from the 8–10 ms window of the push peak to serve as the feature. Further experimentation, however, led to a refinement. When zoomed, the push peak can be observed to consist of two distinct active intervals at the beginning and the end of the 10 ms interval, with relative silence in the middle. These active intervals correspond to a finger touching the key (the touch peak) and then a finger and the key hitting the keyboard supporting plate (the hit peak). The keyboard plate vibrates in both cases. If the FFT is extracted from a 2–3 ms window corresponding to either of the two active intervals, the recognition improves by several percentage points. The reason is that the noise in the middle of the 10 ms interval and on the edges of touch and hit peaks spoil the feature. The touch peak was expressed much better than the hit peak in many of the clicks. We, therefore, use touch peaks to extract features.

Additional details about feature extraction pertain to frequency intervals that go into the feature. We experimented with features extracted from different intervals. We recorded the training and the test set for 30 keys on a single PC keyboard. For each filtered frequency interval, we extracted the features, retrained the network, ran the network over the test set, and observed the recognition rate. Table 1 shows ADCS[1] for different intervals. We find that the best recognition rate is achieved by including the entire active interval in the feature extraction, whereas relatively short intervals produced poorer results.

Another observation that can be made from the experiments is that higher frequencies are generally less informative. Of particular interest is the 300–3400 Hz interval – telephone audio band. The relatively good ADCS for this interval in our experiments suggests that eavesdropping on the clicks over the phone, an attack setting proposed in [12], is potentially possible.

Figure 4 summarizes the sequence of transformations applied to the raw sound of the click for feature extraction.

It is conceivable to use alternative feature extraction algorithms. For example, one may use cepstrum instead of raw FFT [19]. As a matter of fact, one can even experiment with a different type of classifier, such as a support vector machine or a decision tree [14]. As we will see, the current setup was adequate to demonstrate the vulnerability of keyboard and push button devices to attacks based on sound produced from key clicks. It is an interesting topic of future research to explore if these alternatives can enhance further the effectiveness of this attack.

2.3. Distinguishing two keys

Before applying the neural network to the task of distinguishing two PC keyboard keys based on the clicks produced by them, we tried to visualize the difference between the features extracted from the sound produced by the clicks. We applied various aggregations to the features produced from the 10 ms window of a push peak for the two keys, but did not observe significant difference visually. However, features extracted from the 2–3 ms window of a touch peak are visually distinguishable, even if no aggregation is applied (see Figure 5). Note that the visual difference between the touch peak spectrums of different keys differs for different keys.

We next report the neural network results. We chose keys k and l on a standard QWERTY keyboard for this experiment. This and most of the further experiments included the following steps:

1. Preparing the {key, feature} pairs for training the neural network. This step involved recording 100 clicks of each key and extracting the features. Unless noted

1 The average depth of correct symbol (ADCS) is defined in [11]. This measure gives the average position of the correct symbol in the ordered set returned by the network. ADCS parameter can be interpreted as follows. ADCS=1 means a recognition with no errors at all. ADCS=15 (half of the number of the keys in the experiment) means there was no information gain and the recognition was completely unsuccessful.

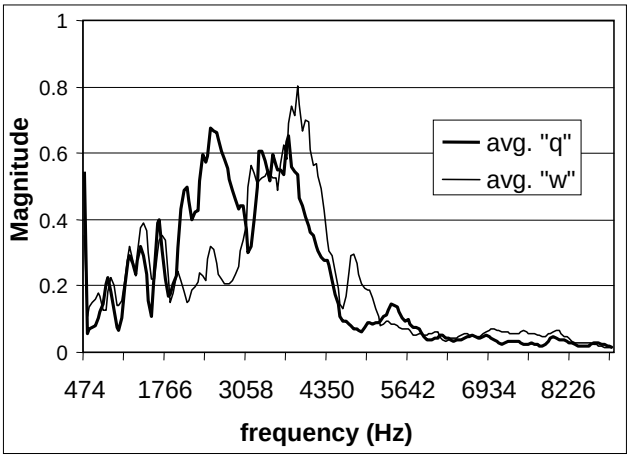

Figure 5. Comparison of the normalized average spectrums (extracted from touch peaks of the clicks).

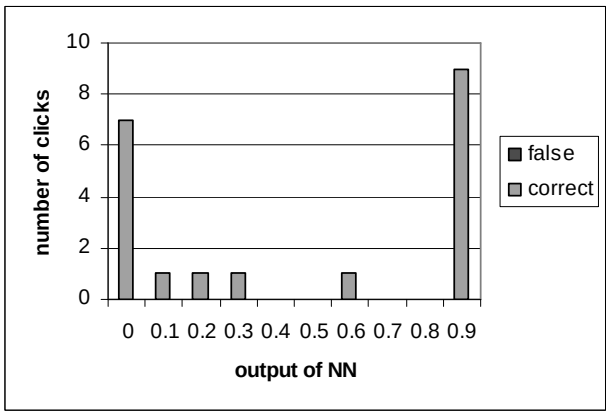

Figure 6. Results of recognizing ten k and ten l clicks each.

otherwise, the clicks were recorded from a distance of about 0.5 meter.

2. Training the neural network with the pairs {key, feature}.
3. Preparing features to test the trained neural network. This step involved recording a set of test clicks (10 clicks per key) and extracting the features.
4. Testing the neural network. In this step, the neural network was provided with a test feature and the output of the network was compared with the identity of the key that was actually pressed.

Figure 6 shows a sample experiment of applying a trained neural network to recognize 10 clicks produced by each of the keys k and l. The network recognizes

Keyboard A, ADCS: 1.99						
key pressed	q	w	e	r	t	y
recognized	9,0,0	9,1,0	1,1,1	8,1,0	10,0,0	7,1,0
key pressed	u	i	o	p	a	s
recognized	7,0,2	8,1,0	4,4,1	9,1,0	6,0,0	9,0,0
key pressed	d	f	g	h	j	k
recognized	8,1,0	2,1,1	9,1,0	8,1,0	8,0,0	8,0,0
key pressed	l	;	z	x	c	v
recognized	9,1,0	10,0,0	9,1,0	10,0,0	10,0,0	9,0,1
key pressed	b	n	m	,	.	/
recognized	10,0,0	9,1,0	9,1,0	6,1,0	8,1,0	8,1,0

Table 2. The neural network is tested with 300 clicks, 10 clicks per key.

that a click is produced by the key k (l) if it assigns an output node a value between 0 and 0.5 (0.5 and 1). The histogram displays the number of correct and false recognitions per each 0.1 interval of the output range of the node. In Figure 6, there are no false recognitions, meaning that all 20 clicks were recognized correctly.

On average, there were only 0.5 incorrect recognitions per 20 clicks, which shows the exposure of keyboard to the eavesdropping using this attack.

2.4. The effect of distance

In the above experiment, clicks were recorded from a short distance of less than 1 meter. We repeated the experiment of distinguishing between two keys by recording clicks from different distances to study the influence of the distance on the quality of recognition. We used an inexpensive parabolic microphone to record the clicks. The microphone was placed behind the person typing. The person was sitting in a cubicle in a hall, with substantial background noise.

The maximum distance we experimented with was approximately 15 meters. There was no decrease in recognition quality even at this distance.

2.5. Multiple keys

We next studied the effect of multiple keys on the quality of recognition.

We trained a network to recognize 30 keys on a keyboard ('q–p', 'a–;', 'z–/'). We then recorded 10 test clicks per key. The neural network had 30 output nodes, each node corresponding to one of 30 keys. The network was trained to recognize a key by assigning a unique output node a value close to 1, while other nodes were assigned values close to

0. At the time of testing, a certain key was deemed recognized if the output node corresponding to this key was assigned largest value when the feature corresponding to the key was provided as input to the neural network.

The results are presented in Table 2. For each of the 10 test clicks of a single key we collected three numbers: how many times the node corresponding to the key had the largest value, the second largest value, or the third largest value among the 30 nodes. We observe that if the node corresponding to the pressed key does not have the largest value among the 30 nodes, then with high probability, it has the second (or third) largest value.

In summary, a key was recognized correctly with the largest value assigned to the correct node in 79% of the clicks (out of 300 test clicks). The network assigned the second (the third) largest value to the correct node in 7% (%2) of the test clicks. So, the correct key was not found among the three candidates proposed by the network in only 12% of the tests. This experiment further reinforces the vulnerability of the keyboard to eavesdropping.

2.6. Multiple PC keyboards

We next investigated the feasibility of attacking a keyboard with a network trained on another keyboard of the same type. We performed our experiments using three GE keyboards. After training a network on keyboard A, we applied this network to recognize the clicks produced by the other two keyboards B and C.

The results are presented in Table 3. As expected, the quality of recognition is lower compared to the case where the network is used to attack the same keyboard as it was trained on (Table 2). We see that 28%, 12%, 7%, and 5% of the clicks were recognized correctly as the first, second, third, or fourth candidate respectively for keyboard B. So, the correct key was found among the four guesses made by the network in 52% of the tests. For keyboard C, the same statistics is 50%.

These results show that the quality of recognition in this setup might be insufficient to eavesdrop on the plain text being typed, however, the information gain is significant for password snooping.

2.7. Handling different typing styles

In the previous experiments, all the clicks used in the training set as well as the test set were generated by the same person, using the same finger and approximately the same force.

We next studied the effect on recognition if a person types with variable force. Initially, the network was trained on clicks produced with an approximately constant typing

Keyboard B, ADCS: 9.24						
key pressed	q	w	e	r	t	y
recognized	6,1,1	4,1,1	0,1,0	0,2,1	5,1,1	1,0,0
key pressed	u	i	o	p	a	s
recognized	1,2,1	4,1,1	4,3,1	4,1,1	4,1,0	2,1,0
key pressed	d	f	g	h	j	k
recognized	1,4,0	0,0,0	1,0,1	5,1,1	9,0,0	1,0,2
key pressed	l	;	z	x	c	v
recognized	5,0,1	3,2,0	1,0,2	0,0,0	2,0,0	0,2,2
key pressed	b	n	m	,	.	/
recognized	3,3,1	3,1,1	5,1,1	0,2,1	2,1,0	7,2,1
Keyboard C, ADCS: 9.10						
key pressed	q	w	e	r	t	y
recognized	1,1,3	0,0,1	0,0,1	4,3,1	0,0,0	0,0,0
key pressed	u	i	o	p	a	s
recognized	2,3,0	1,3,0	3,3,3	1,1,1	0,1,0	1,2,0
key pressed	d	f	g	h	j	k
recognized	2,0,1	0,1,0	2,0,4	2,4,1	0,3,1	3,1,0
key pressed	l	;	z	x	c	v
recognized	1,0,0	1,1,0	2,2,0	0,1,1	10,0,0	1,0,2
key pressed	b	n	m	,	.	/
recognized	7,1,1	7,1,1	5,0,2	1,1,3	4,1,0	2,1,1

Table 3. Keyboards B and C are attacked using a network trained on the keyboard A of the same type. There are 300 test clicks per keyboard, 10 clicks per key.

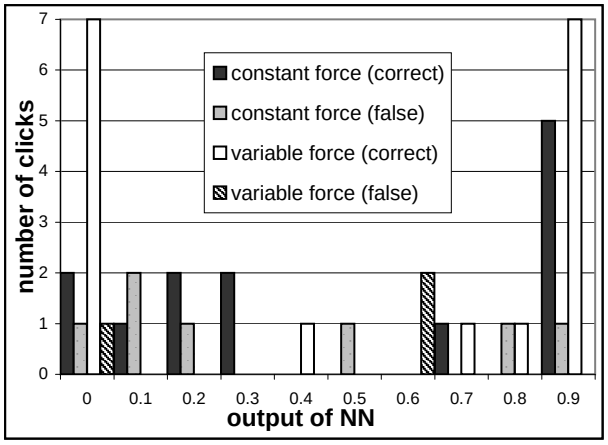

Figure 7. Test clicks produced with variable force are classified by two networks. For one network a constant force was used to produce the training set of clicks. For another network, variable force was used.

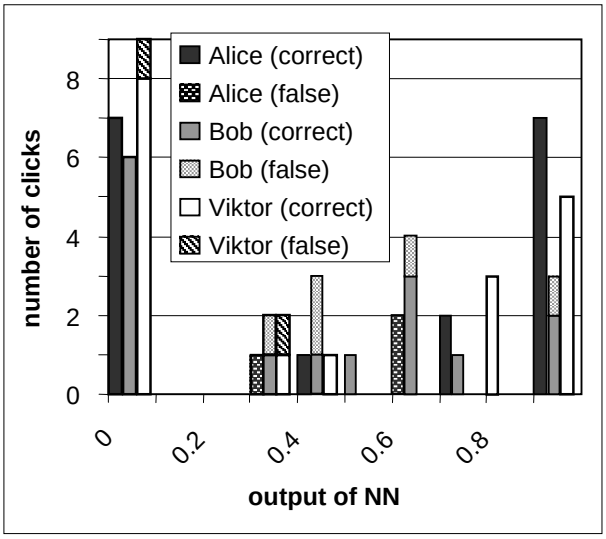

Figure 8. Clicks produced by three different persons are tested on the network trained by another person.

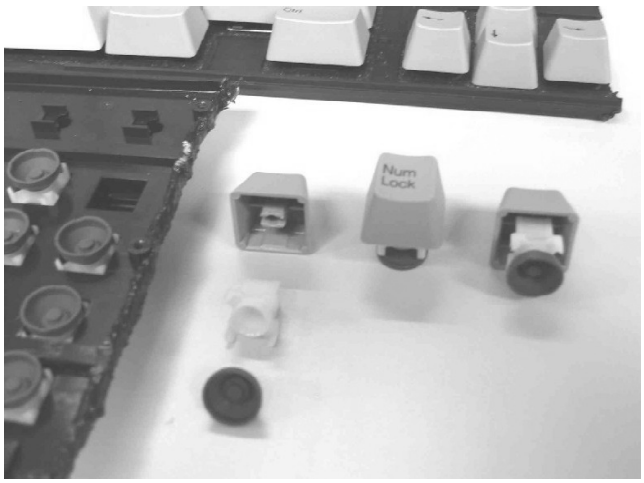

Figure 9. The architecture of a mechanical keyboard.

force. The results of recognizing the test clicks produced with variable force were poor (Figure 7).

Next, we generated a training set in which clicks were produced by typing with a variable force and trained the network again. The testing results (Figure 7) now are as good as in the basic experiment, meaning that the network can be trained to recognize the clicks produced with different forces. Other experiments show that the same conclusion is valid for typing with one vs. many fingers. Namely, if trained with one finger only, the clicks produced by different fingers are recognized with high error rate. But a network trained with many fingers is as good as the basic one in recognizing the clicks: approx. 1 click (out of 20 clicks) is recognized incorrectly.

We also investigated if different typing styles affect the quality of recognition. The answer to this question is particularly important for practical attacks, where the network might be trained by one person (an attacker himself) and then applied to the keyboard in use by another person.

The network was trained using the training set produced with variable typing force by one person. Then, the test sets were recorded from three different people. They were free to use any typing style. The results of recognition (Figure 8) show that it is possible for the network to be trained on one person and then applied to attack the same keyboard in use by another person. The difference in typing style affects the quality of the classification of the clicks only slightly.

3. Countermeasures

An obvious candidate for countermeasure is a silent keyboard. It can be a keyboard made of rubber [5], or a keyboard based on a touchscreen or touchstream technologies [4]. Recently, virtual keyboards have appeared that can be projected on a flat surface [1] or in the air [3].

These choices are more expensive than the standard mechanical keyboard. Typing on a standard keyboard is much more comfortable than typing on a touchscreen or a rubber keyboard.[2]

In this section, we try to determine experimentally the reasons for clicks to sound different. Conclusion made from this study can help in designing a mechanical keyboard that produces indistinguishable clicks.

3.1. Mechanical Keyboard

Figure 9 shows the schematic of a mechanical keyboard. Each key consists of three parts: a head, a dome-shaped rubber part, and an intermediate plastic part that interconnects the head and the rubber. The keys are kept together by a plastic plate of a size of the keyboard, with the intermediate parts of the key going through this plate.

Under the rubber part there is an electrical switch corresponding to the key. When the key is being pressed, the dome-like rubber part of the key is squeezed, and the top of the dome forces the switch under it to close the circuit.

2 The silent "chicklet" keyboard was an important factor in the non-acceptance by the market of IBM PC Jr. [2].

3.2. Why the clicks produce different sounds

We hypothesized three reasons for clicks to produce different sounds:

1. The interaction of the sound produced by a key click with the surrounding environment such as neighborhood keys might be the reason for each key to sound slightly different.
2. Microscopic differences in the construction of the keys might cause them to sound different.
3. Different parts of the keyboard plate might produce different sounds when the nearby key is pushed. By analogy with a drum, striking a key at different locations on the plastic plate provides different timbres.

The first hypothesis was ruled out by the following experiment. After the neural network was trained to distinguish between two keys, the surrounding environment was changed by removing several neighborhood keys on the keyboard. This modification should have changed the way the clicks sound if the environment is the underlying reason. However, the recorded test clicks were successfully classified by the trained network, invalidating the hypothesis.

To check the second hypothesis, we trained the network to recognize two keys: k and l. We then interchanged them on the keyboard plate and recorded their test clicks. The network identified the test clicks of the k element as l, and the test clicks of the l element as k (Figure 10).

This experiment led us to conclude that the second hypothesis can be ruled out: The microscopic differences in key elements play no or minor role in making the clicks to sound different. This conclusion does not reject the obvious observation that macroscopic differences play role in causing the keys to sound different. For example, the space key may sound different from a standard key partially because of the difference in size between the keys.

To check the third hypothesis we cut out several pieces of the keyboard plate with one key in each piece using a milling machine. Obviously, this operation should render the third hypothesis irrelevant, because there is not any notion of the position of the keys on the keyboard plate. If the third hypothesis were true, we expect the network not to be able to distinguish between clicks produced by the keys mounted in these pieces. Indeed, after we trained the network, the network was unable to recognize the keys based on the test clicks (Figure 11), thus supporting our third hypothesis.

These experiments suggest that the clicks sound different because the keys are positioned at different positions on the keyboard plate. Knocking at different positions on the plate makes different sounds. Neither microscopic differences between the key elements nor a surrounding environment plays a significant role in differences between the sounds produced by different keys.

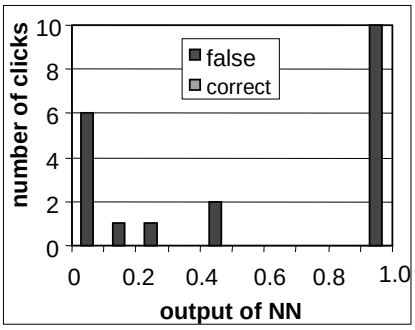

Figure 10. Results of recognizing two keys exchanged on the keyboard. First 10 clicks are produced by the l key element, the next 10 clicks are produced by the k key element. The network is trained to recognize k with 0 and l with 1 before the keys were exchanged.

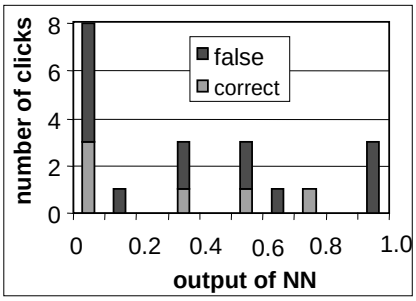

Figure 11. Results of recognizing two keys cut out of the keyboard plate. The network was trained to recognize the keys after the keys were cut out.

The construction of a homophonic keyboard should be engineered to obliterate the above identified cause for keys to sound differently. Possibilities include, for example, not placing the keys in one plate, or producing the keyboard plate from a material that does not conduct vibrations to prevent the plate from acting as a "drum".

4. Notebook Keyboards, Telephone Pads, and ATM Pin Pads

We repeated the experiment of distinguishing between two keys for a notebook keyboard, a telephone pad, and an ATM pin pad. For the notebook keyboard the two keys were again k and l. The telephone pad and the ATM pad were tested with keys 1 and 2.

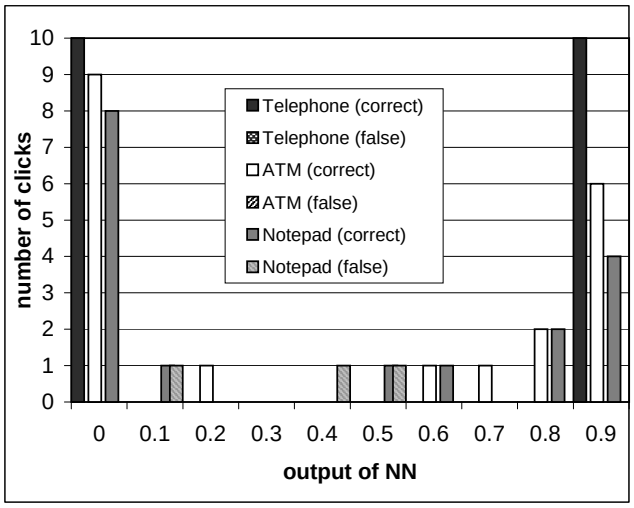

Figure 12. Results of recognizing two keys on a notepad keyboard (the keys are k and l) and on a telephone and ATM pads (the keys are 1 and 2)

Keypad A			Keypad B			Keypad C		
1	2	3	1	2	3	1	2	3
9,1,0	10,0,0	10,0,0	10,0,0	5,4,1	3,2,2	6,4,0	0,0,0	0,1,0
4	5	6	4	5	6	4	5	6
9,0,1	10,0,0	10,0,0	0,0,0	9,1,0	6,2,2	0,1,2	0,0,0	2,3,1
7	8	9	7	8	9	7	8	9
10,0,0	10,0,0	10,0,0	10,0,0	0,0,1	6,3,1	9,1,0	1,1,3	4,3,1
ADCS: 1.03			ADCS: 2.59			ADCS: 4.08		

Table 4. Telephone pads A, B and C, all of the same type, are attacked using a network trained on the pad A. There are 90 test clicks per keypad, 10 clicks per key.

The results are shown in Figure 12. With 2 incorrect recognitions out of 20 clicks, the notebook keyboard shows less vulnerability to the attack than the standard PC keyboard. Interestingly, all the 20 test clicks of the telephone pad as well as all the 20 test clicks of the ATM pad were recognized correctly.

4.1. Multiple telephone pads

We repeated the experiment described earlier for multiple keyboards for telephone pads. Namely, we trained the network using one telephone pad and applied this network to recognize clicks of other pads.

The data in Table 4 is presented in the same format as in Section 2.6. The results of this experiment are similar to the results of the experiment with keyboards:

- It is feasible to attack a telephone with a network trained on another telephone of the same type. However, the quality of recognition is lower compared to the case where the network is trained on the same telephone.
- The quality of recognition is different from pad to pad. For example, the clicks of telephone B are recognized considerably better than the clicks of telephone C.

5. Related Work

A Computerworld article [16] discusses the computer security in general, and suggests that "secrecy is an illusion" by mentioning different exotic ways of breaking into the systems. The "keyboard trick" is included as one of the approaches. Unfortunately, the author could not recall any references for this "trick" [17].

TEMPEST documents NACSEM 5103, 5104, and 5105 are about acoustic emanations, but are (unfortunately) classified according to the partially unclassified NACSIM 5000 [15]. This document also says that "Keyboards, printers, relays – these produce sound, and consequently can be sources of compromise", while going no further beyond this statement.

The authors of [20] observed that an eavesdropper can collect timing information from the traffic of an interactive secure shell session. In particular, this timing information reveals the delays between keys typed. The distribution of the inter-keystroke delays differs slightly for different pairs of keys. Thus, partial information about the identity of the typed keys is revealed to an eavesdropper. The latency distributions for different key pairs highly overlap, so the information gain is relatively low. Moreover, different users may demonstrate different inter-keystroke timings that further reduces information gain. However, we can imagine combining the timing analysis with the acoustic attack described herein to make decisions about the key clicks that were not unambiguously recognized based on the acoustic data alone. A related software–only timing attack is described in [21].

Wireless keyboards can be eavesdropped by using another receiving station. To prevent this disclosure, several keyboard producers offer the keyboards with over-the-air encryption.

Two attacks using electromagnetic emanations of the keyboard are briefly mentioned in [6]. The authors also explain how to counter them by modifying the keyboard device driver, and the firmware in the keyboard microcontroller. The emanations from the LEDs, in particular keyboard LEDs, are studied in [13]. The use of sounds emitted by a Hagelin rotor machine for a side channel attack

has been documented in [22]. Acoustic emissions from matrix printers were shown to be compromising in [7].

The authors of [9] tackle the problem of protecting the users from thieves that use video camera or binoculars to spy for the telephone card numbers and ATM pins remotely. The suggested solution is to install an eye tracking system at the terminals, so that the users use the motions of their eyes to input the numbers.

6. Summary

We explored acoustic emanations of keyboard-like input devices to recognize the content being typed. After providing a detailed description of the basic attack on a PC keyboard, we successfully applied this attack to other types of push button input devices, such as notebook keyboards, telephone pads, and ATM pads.

A sound-free (non-mechanical) keyboard is an obvious countermeasure for the attack. However, it is neither comfortable for users nor cheap. We identified possible reasons that cause the keys to sound slightly different to draw preliminary conclusions on how a homophonic mechanical keyboard that produces indistinguishable clicks can be constructed.

The work presented in this paper points to many avenues for further research. One can explore to quantify the environmental variables under which the proposed attack can succeed. One can also investigate the vulnerability of other push button devices such as push button locks found on many doors and push button garage door openers installed in many houses. Only by measuring and analyzing the vulnerability of sound producing devices can we hope to develop countermeasures and make them secure.

Acknowledgements. We wish to profusely thank Markus Kuhn of Cambridge University and Malcolm Slaney of IBM Research for their invaluable suggestions and advice for improving the paper. Roberto Bayardo, Christos Faloutsos, Kevin McCurley, Marc McSwain, Barton Smith, Ramakrishnan Srikant, and Shumin Zhai from IBM Research, as well as anonymous reviewers provided useful feedback on the preliminary draft of the paper. Dave Altknecht and Alan Melton helped us with our work with a milling machine at the IBM Research model shop.

References

[1] Canesta keyboards. `http://www.canesta.com/products.htm`.

[2] Chicklet keyboard from IBM PC Junior. `http://www.digibarn.com/collections/devices/pcjr-chicklet-keyboard/`.

[3] HoloTouch technology. `http://www.holotouch.com`.

[4] TouchStream keyboards. `http://www.fingerworks.com/`.

[5] The virtually indestructible keyboard. `http://www.grandtec.com/vik.htm`.

[6] R. J. Anderson and M. G. Kuhn. Soft tempest – an opportunity for NATO. In *Proceedings of Protecting NATO Information Systems in the 21st Century, IST Symposium, Washington DC, USA*, Oct. 1999.

[7] R. Briol. Emanation: How to keep your data confidential. In *Symposium on Electromagnetic Security For Information Protection, SEPI'91, Rome, Italy*, Nov. 1991.

[8] M. A. Casey. *Introduction to MPEG-7: Multimedia Content Description Language*, chapter Sound Classification and Similarity Tools. J. Wiley, 2001.

[9] M. D. Flickner, Q. Lu, and C. H. Morimoto. Gaze-based secure keypad entry system. Patent US6282553, 2001.

[10] M. G. Kuhn. Optical time-domain eavesdropping risks of CRT displays. In *Proceedings of IEEE Symposium on Security and Privacy, Berkley, California, USA*, May 2002.

[11] M. G. Kuhn. Compromising emanations: eavesdropping risks of computer displays. Technical Report UCAM-CL-TR-577, Computer Laboratory, University of Cambridge, 2003.

[12] M. G. Kuhn. Personal communication. Feb. 2004.

[13] J. Loughry and D. A. Umphress. Information leakage from optical emanations. *ACM Transactions on Information and System Security*, 5(3):262–289, Aug. 2002.

[14] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[15] NSA. NACSIM 5000 TEMPEST fundamentals. National Security Agency, Fort George G. Meade, Maryland, `http://cryptome.org/nacsim-5000.zip`.

[16] N. Petreley. Secrecy is an illusion. *Computerworld*, `http://www.computerworld.co.nz/webhome.nsf/0/7237CE66D15E3BFFCC256B8A00%0C087B`, Apr. 2002.

[17] N. Petreley. Personal communication. Sept. 2003.

[18] R. Price, J. Willmore, and W. Roberts. Genetically optimised feedforward neural networks for speaker identification. Technical Report DSTO-TN-0203, Defence Science and Technology Organisation (Australia), 1999.

[19] S. W. Smith. *The Scientist and Engineer's Guide to Digital Sound Processing*. California Technical Publishing, 1997.

[20] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and SSH timing attacks. In *Proceedings of 10th USENIX Security Symposium, Washington DC, USA*, Aug. 2001.

[21] J. Trostle. Timing attacks against trusted path. In *Proceedings of IEEE Symposium on Security and Privacy, Berkley, California, USA*, May 1998.

[22] P. Wright. *Spycatcher*. Random House Value Pub, 1989.

[23] A. Zell, N. Mache, T. Sommer, and T. Korb. Recent developments of the SNNS neural network simulator. In *Applications of Neural Networks Conf., SPIE, volume 1469, Orlando Florida*, 1991.