“Keyboard Acoustic Emanations Revisited”

Li Zhuang, Feng Zhou, and J.D. Tygar

Presenter: Daniel Liu

Overview

- Introduction to Emanations
- Keyboard Acoustic Emanations
- Keyboard Acoustic Emanations Revisited
- Extensions
- Questions?

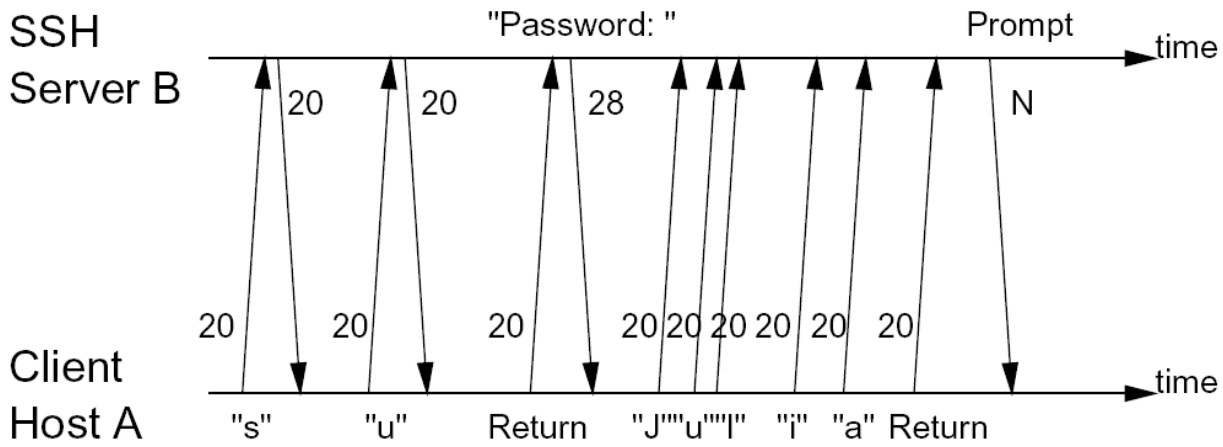
Emanations are Everywhere

- Unintended information leakage
  - Inputs and Outputs
  - Software
  - Hardware
  - Networks
  - TEMPEST

"Timing Analysis of Keystrokes and Timing Attacks on SSH"

D. Song, D. Wagner, X. Tian. UC Berkeley, 2001.

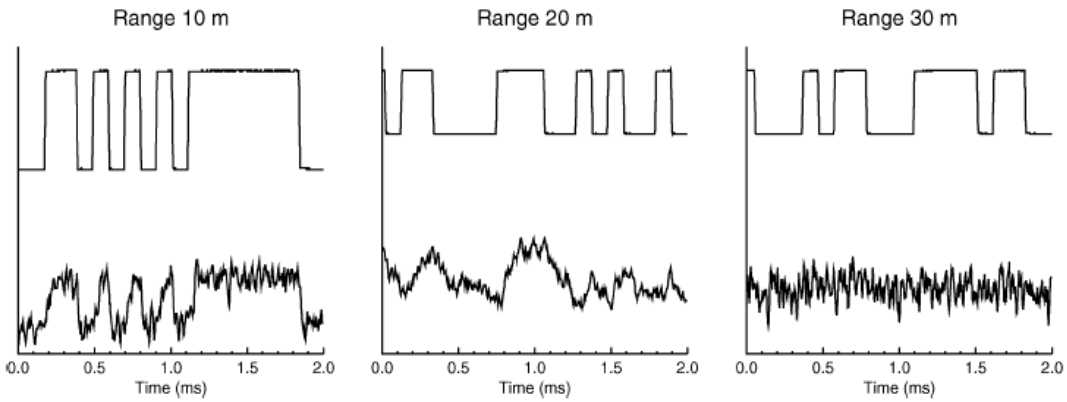
- Interactive mode sends every keystroke in a separate IP packet
- Typing patterns can be analyzed



"Information Leakage from Optical Emanations"

J. Loughry, D. Umphress. 2002.

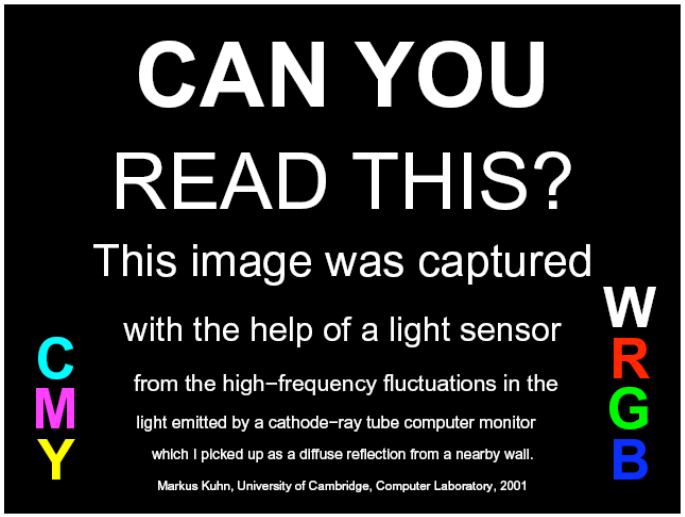
- LED status indicators have been shown to correlate with the data being sent
- Many devices were shown to be vulnerable

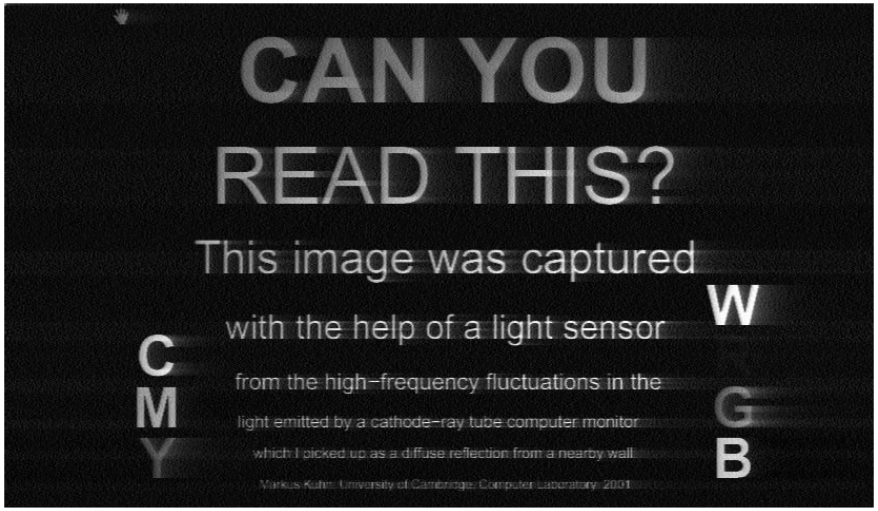


"Optical Time Domain Eavesdropping Risks of CRT Displays"

M. Kuhn, 2002.

- Uses a fast photosensor to deconvolve the signal off of a reflected wall
- Based on phosphor decay times

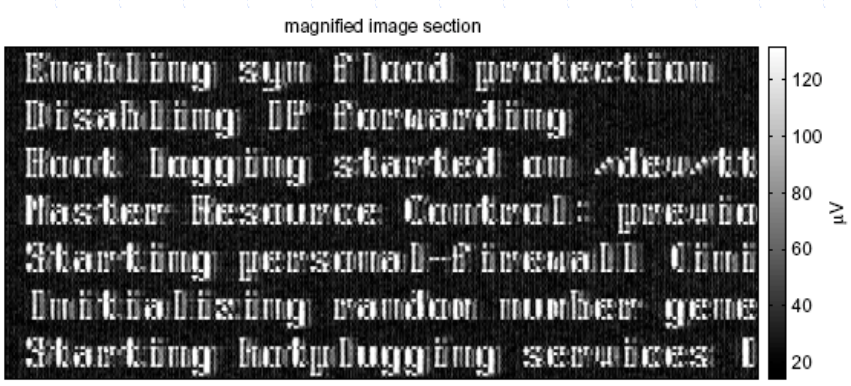


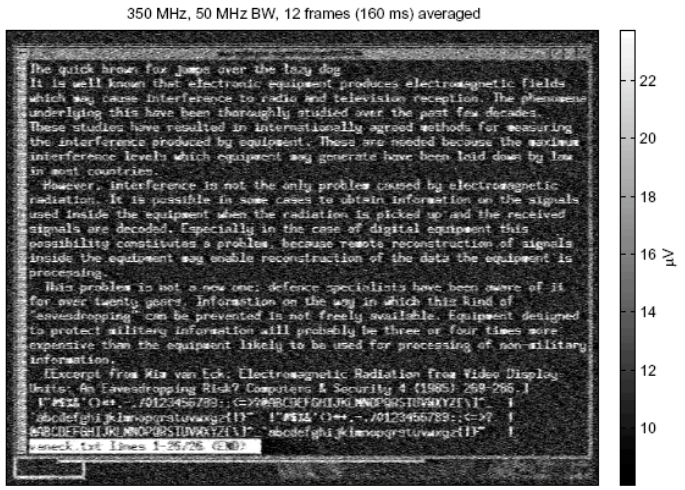


"Electromagnetic Eavesdropping Risks of Flat Panel Displays"

M. Kuhn, 2004.

- Signals can be received with directional antennas and wideband receivers
- Gbit/s digital signals are sent via serial transmissions and are detectable





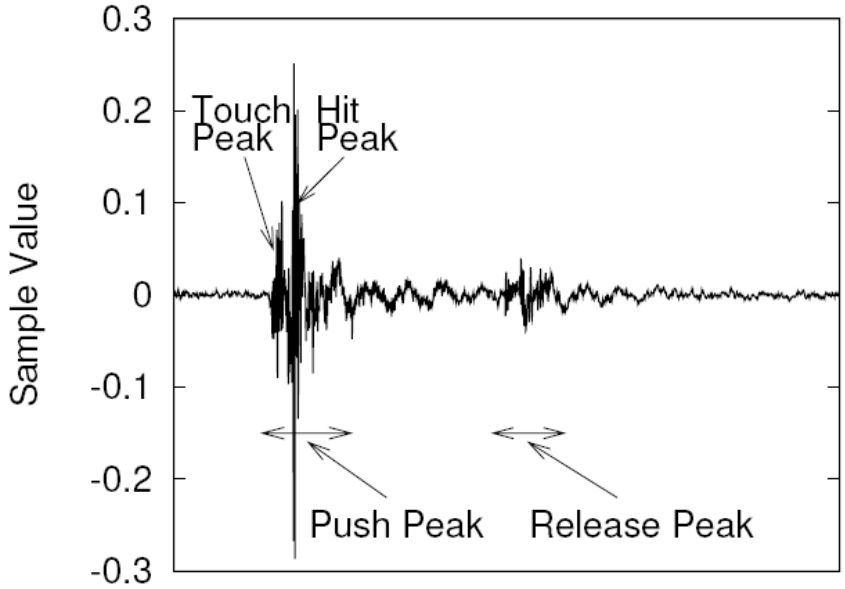
"Keyboard Acoustic Emanations"

D. Asonov, R. Agrawal, 2004.

- Differentiate the sound emanated by different keys to eavesdrop on what is being typed
- Can be done with a standard PC microphone
- Does not require physical intrusion
  - Parabolic Microphones
  - Record remotely without user knowledge
- Recognition is based on using neural nets

Basic Notion…

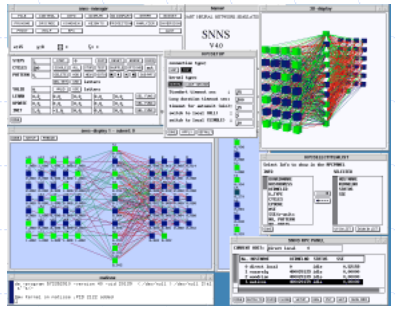
- Not all keys sound the same
- Consider ‘q’ and ‘t’



Experimental Setup

- IBM Keyboards, GE Power Keyboards, Siemens RP240 Phones
- Simple, omni-directional, and Bionic Booster Parabolic microphones
- Standard PC Sound Card and Sigview Software
- JavaNNS Neural Network Software

SIGVIEW



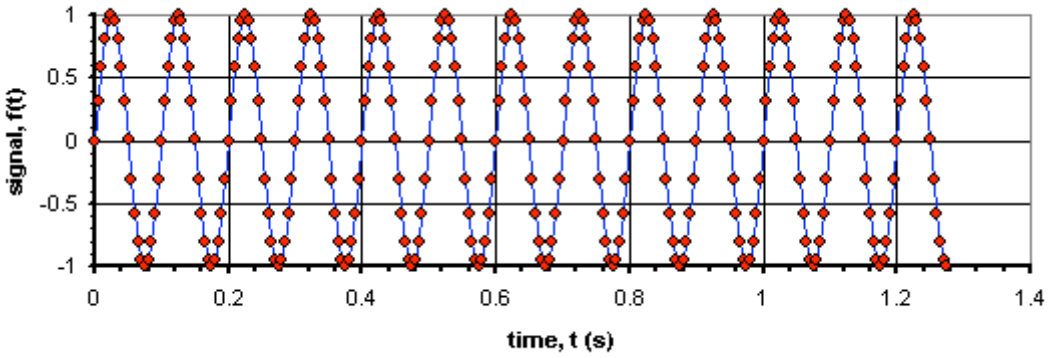
http://www.sigview.com/
http://www-ra.informatik.uni-tuebingen.de/SNNS/

Threat Analysis

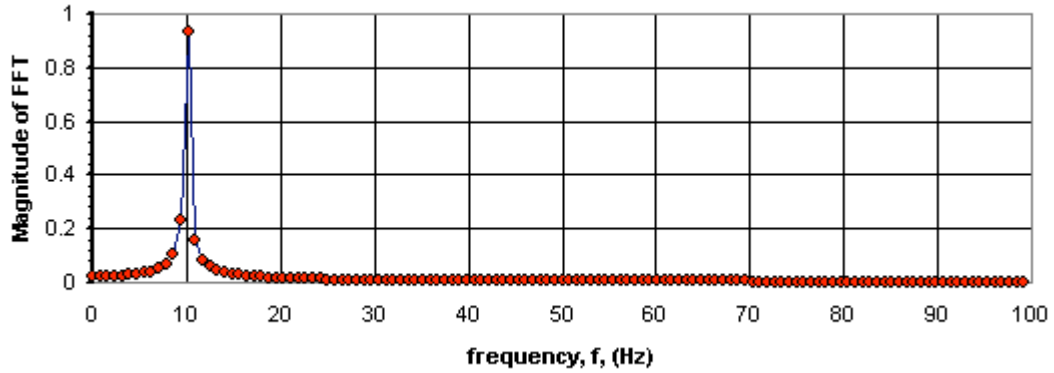
- Attacker must use labeled training data for best results
- Only looked at a few types of keyboards
- No mention of typing rate of the users
- Maximum distance tested with a parabolic microphone was 15 m
- There are many assumptions made!

Fast Fourier Transform (FFT)

- Takes a discrete signal in the time domain and translates it to the frequency domain



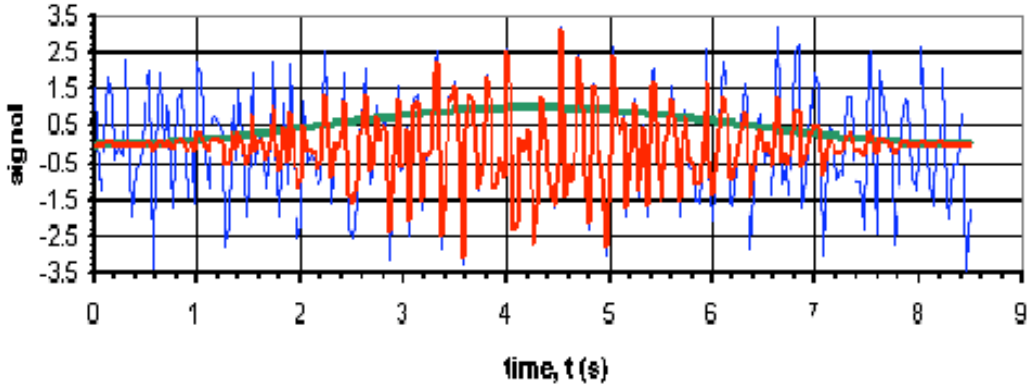
10 Hz Sine Wave
Amplitude 1



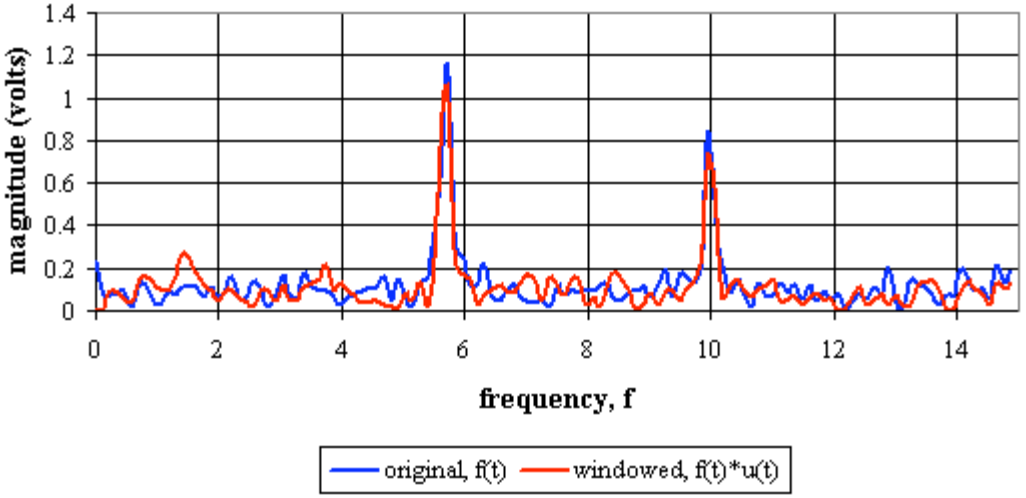
200 samples/sec
Amplitude ~1
(dispersion)

http://www.mne.psu.edu/me82/Learning/FFT/FFT.html

FFT Continued…



Looks like
Random noise

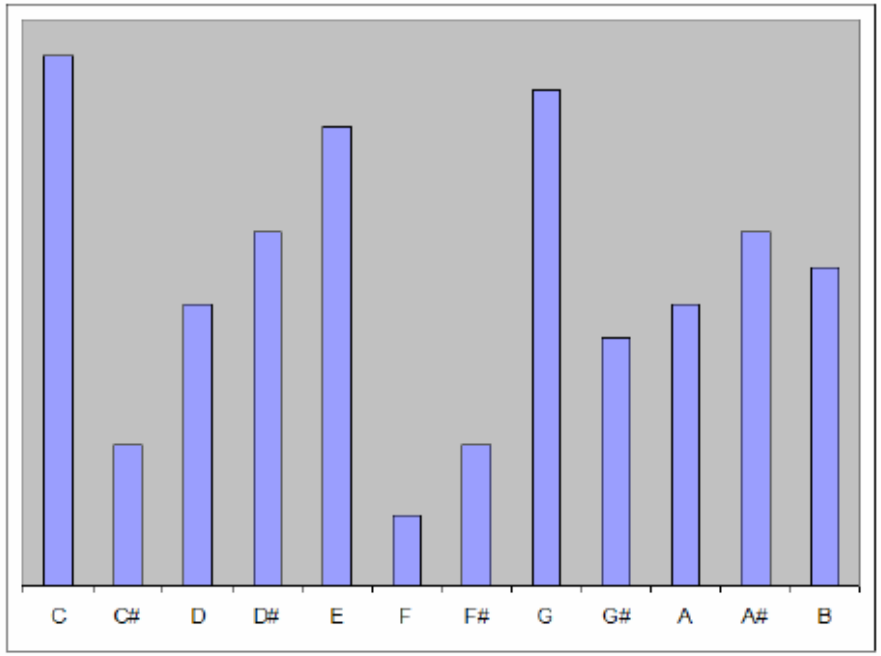


Components at:
5.7 Hz
10 Hz

“Recognizing Chords with EDS”

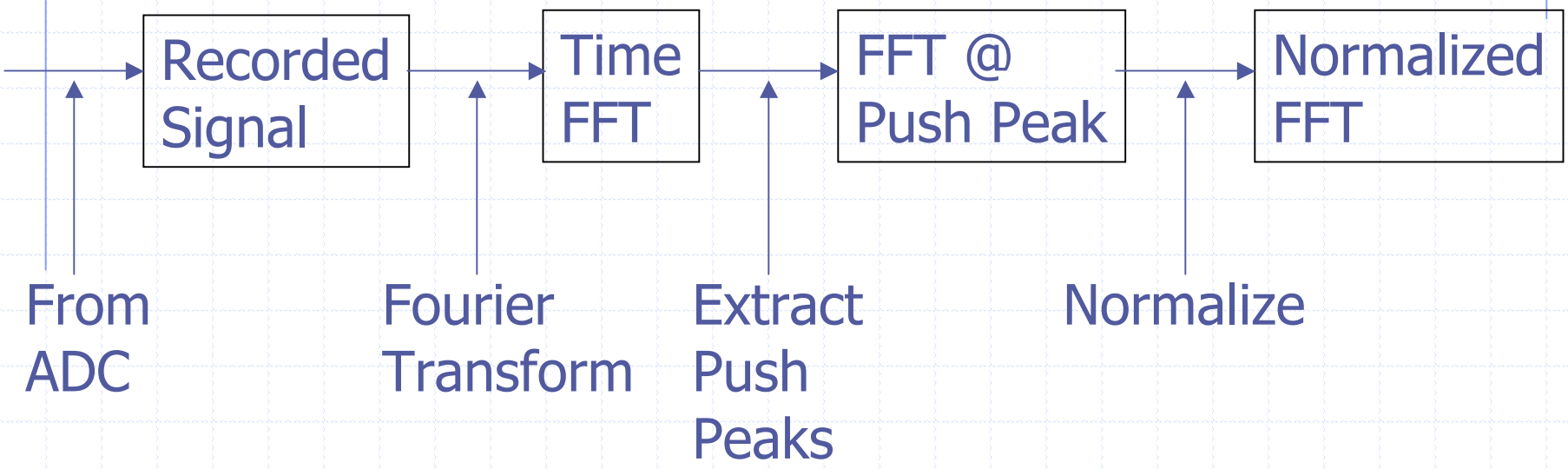
G. Cabral et al, 2005.

- Compute FFT
- Sum Frequency Bins



CMaj Chord
C, E, G are peaks

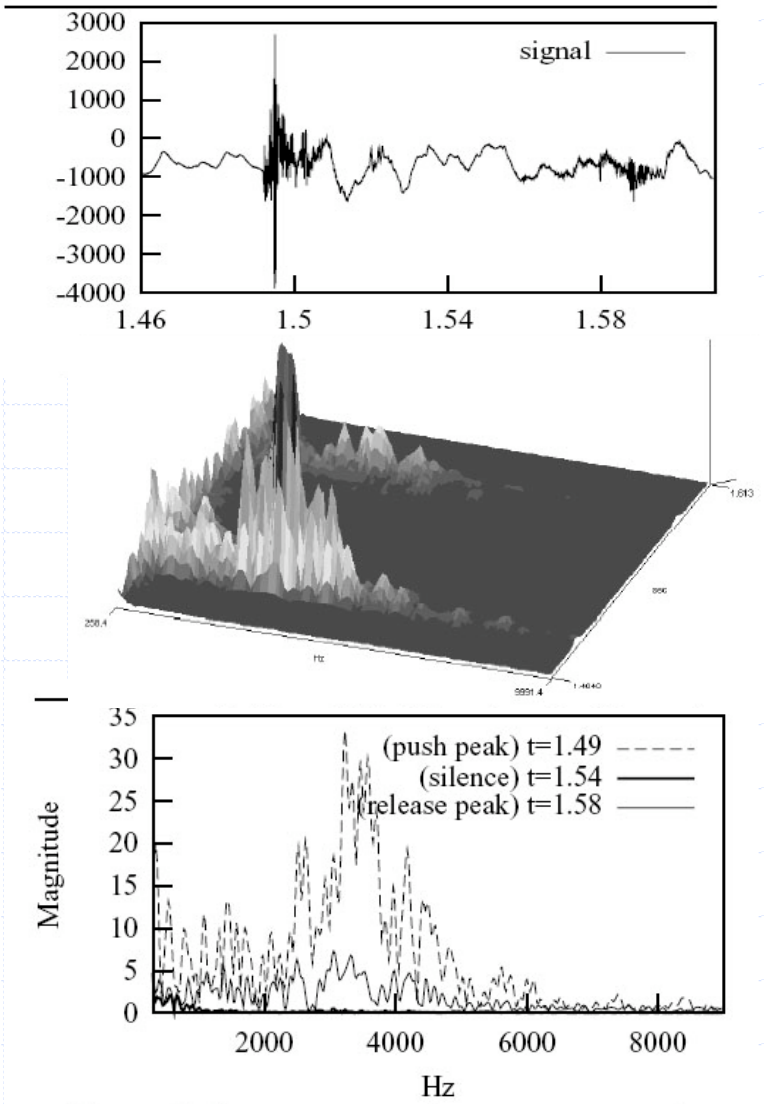
Feature Extraction Design



What about key presses that overlap?

Feature Extraction Reality

- Recorded Signal
- Time FFT
- FFT at Push Peak



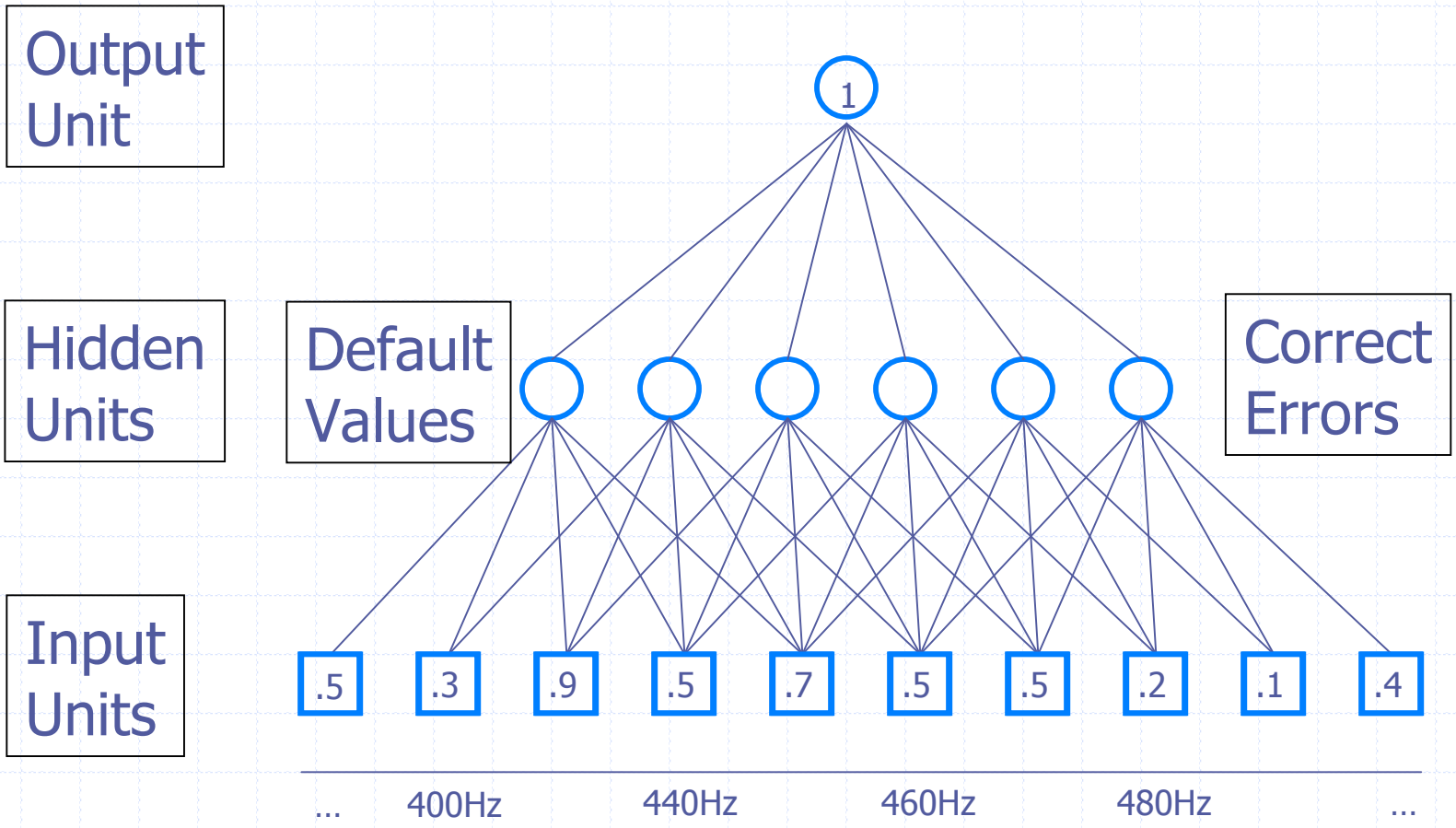
Why Do We Need FFT Here?

- Neural nets typically take dozens to several hundred inputs (all 0 to 1)
- This is about 1kB of input
- The keyboard click signal is 10kB
- FFT is used to extract features of the "touch peak" of the signal (2-3 ms)
- This allows the neural net to be trained

Neural Network

- Backpropagation neural net
- Input nodes, one value per 20 Hz
- Used 6 to 10 hidden nodes
- “Two key” experiments had one output
- Multiple key experiments had an output for each key

Training Neural Net



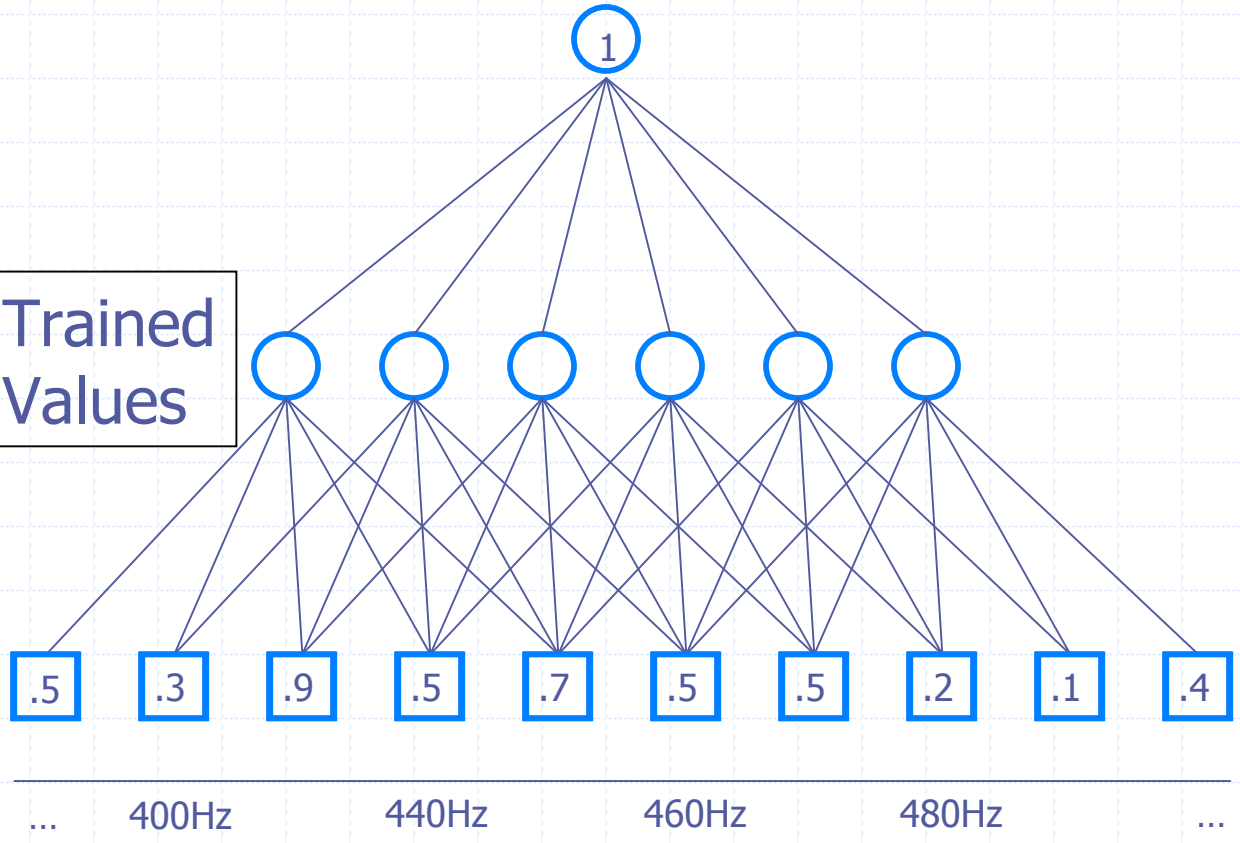
Using the Trained Neural Net

Output Unit

Hidden Units

Trained Values

Input Units



But this training process can be tedious!

Only Need up to 9 kHz

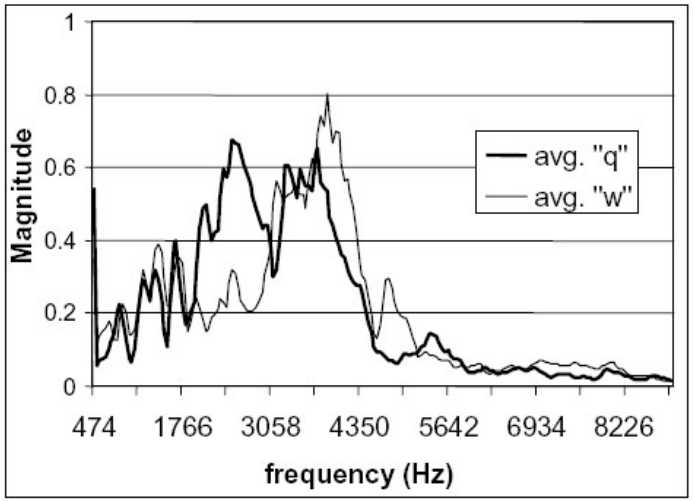
- Average depth of correct symbol is best with 0 – 9 kHz
- 300 – 3400 Hz still gives decent accuracy (telephone audio band)

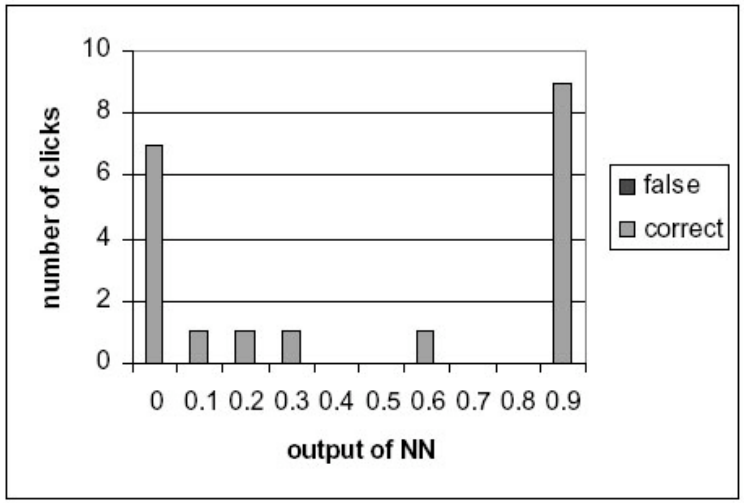
kHz	0–9	.3–3.4	0–3	1–4	2–5	3–6	4–7	5–8	6–9
ADCS	1.65	2.70	2.76	3.45	4.36	3.94	5.05	5.94	7.70

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

First Test: Distinguishing Two Keys

- Record and extract features
- Trained the neural net to two keys
- Record new features for the neural net
- Test the neural net and check accuracy
- No decrease in recognition quality even at 15 meters





Testing with Multiple Keys

- Trained to recognize 30 keys, 10 clicks each
- Correct identification: 79%
- Counting second and third guesses: 88%

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

Realistic Typing Model?

- Each key is individually typed
- “hunt and peck” typist
- Very few people type like this
- Not a significant threat to touch typists

Testing with Multiple Keyboards

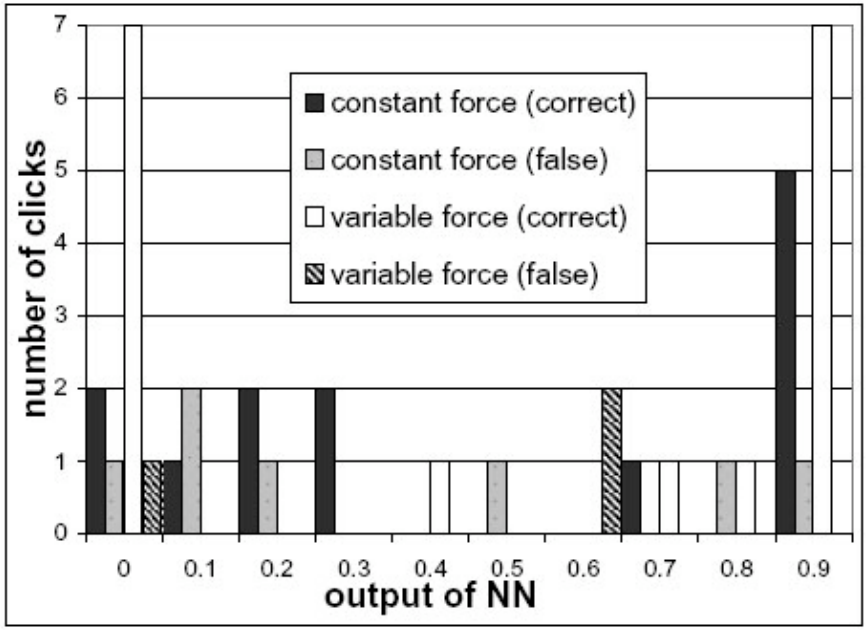
- Training done with another keyboard (A)
- Four candidate guesses (28%, 12%, 7%, 5%)
- Keyboard B and C are ~50% accurate (4 guesses)
- This test uses three different GE keyboards(?)

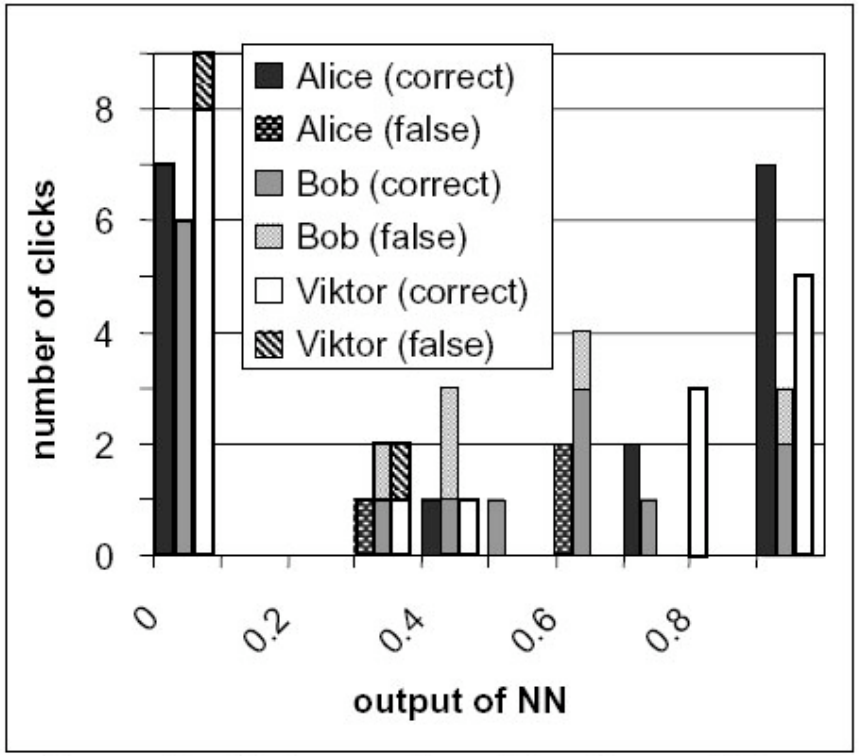
Keyboard B, ADCS: 9.24						
key pressed	q	w	e	r	t	y
recognized	6,1,1	4,1,1	0,1,0	0,2,1	5,1,1	1,0,0
key pressed	u	i	o	p	a	s
recognized	1,2,1	4,1,1	4,3,1	4,1,1	4,1,0	2,1,0
key pressed	d	f	g	h	j	k
recognized	1,4,0	0,0,0	1,0,1	5,1,1	9,0,0	1,0,2
key pressed	l	;	z	x	c	v
recognized	5,0,1	3,2,0	1,0,2	0,0,0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

Different Typing Styles (Two Key)

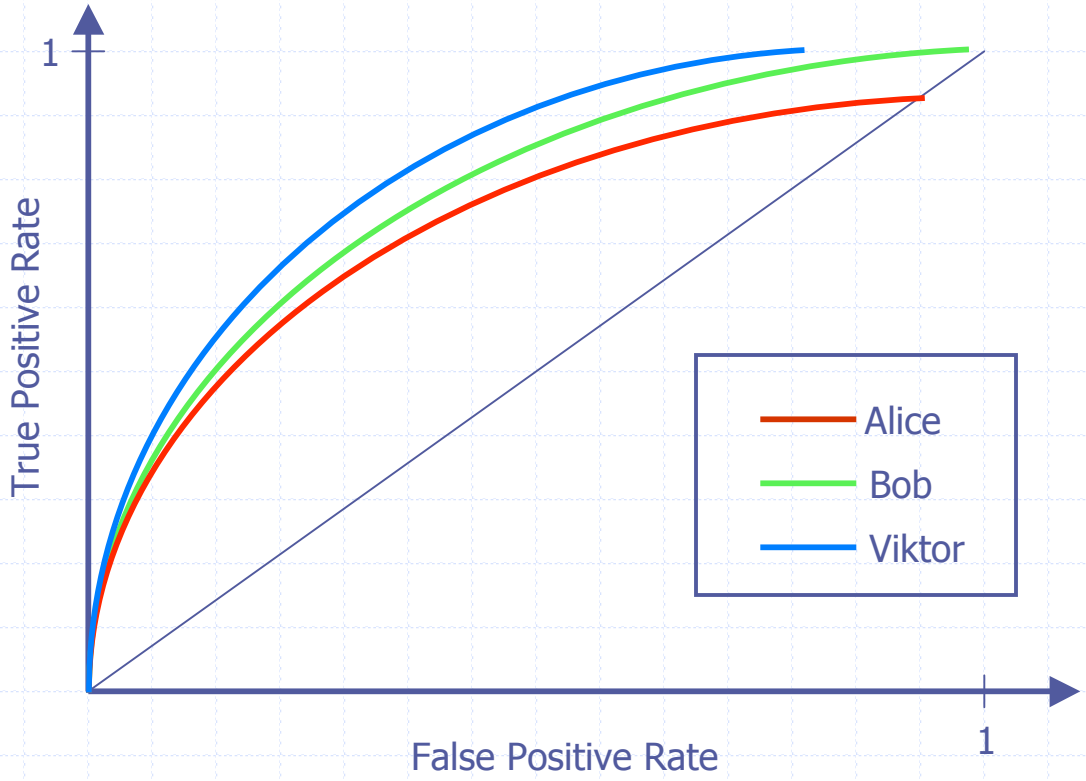
- Variable Force Typing
- Comparison of Three Different Typists





ROC Curves

- Shows the multiple keyboards test
- But we lose the exact output values

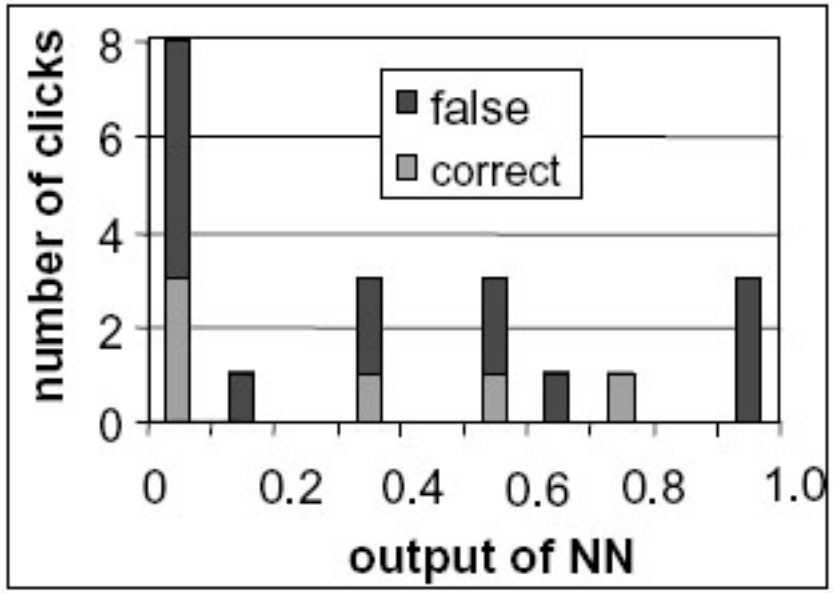


Why Clicks Produce Different Sounds

- Three Possibilities
  - Surrounding environment of neighboring keys
  - Microscopic differences in construction of keys
  - Different parts of the keyboard plate produce different sounds

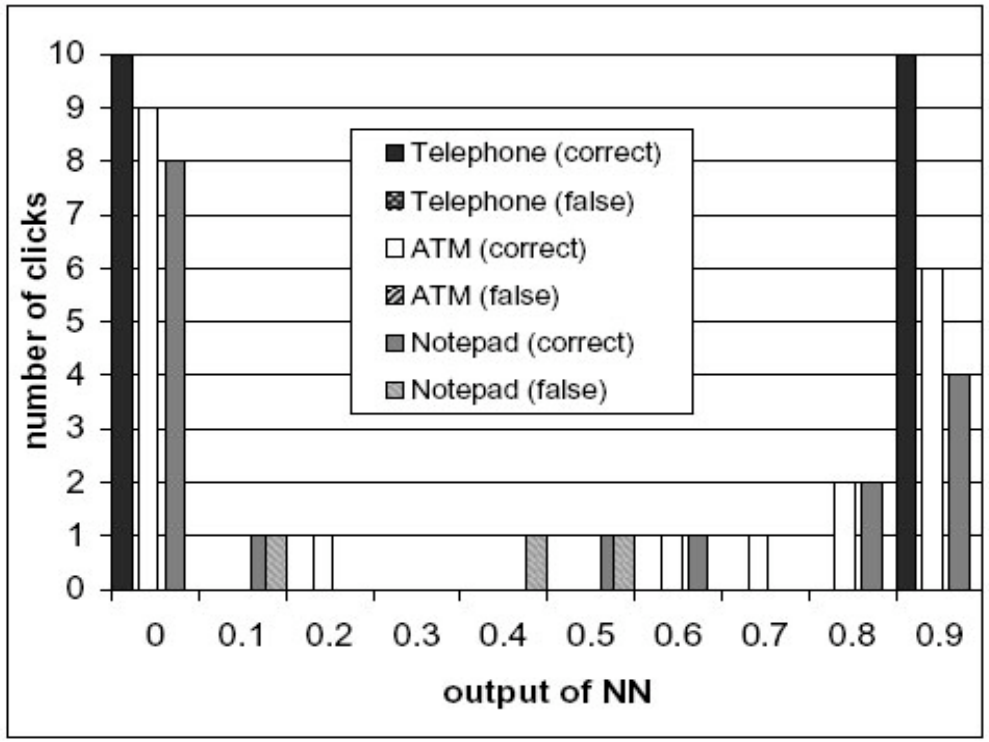
Milling Out Pieces

- Several pieces of the keyboard plate were removed
- Neural net was unable to pass the two key test



Notebook, ATM, and Phone Pads

- Notebook keys are not quite as vulnerable
- ATM and Phone Pads are vulnerable



Countermeasures

- Grandtec rubber keyboard

- Fingerworks Touchstream

- Gaze based selection?

Can We Do Better?

- Can this be done without recording and using labeled training data?
- Are FFTs a good way to represent features?
- Very poor recognition with multiple keyboards
- Typing styles slightly reduce accuracy
- Are there ways to take advantage of English language structure?

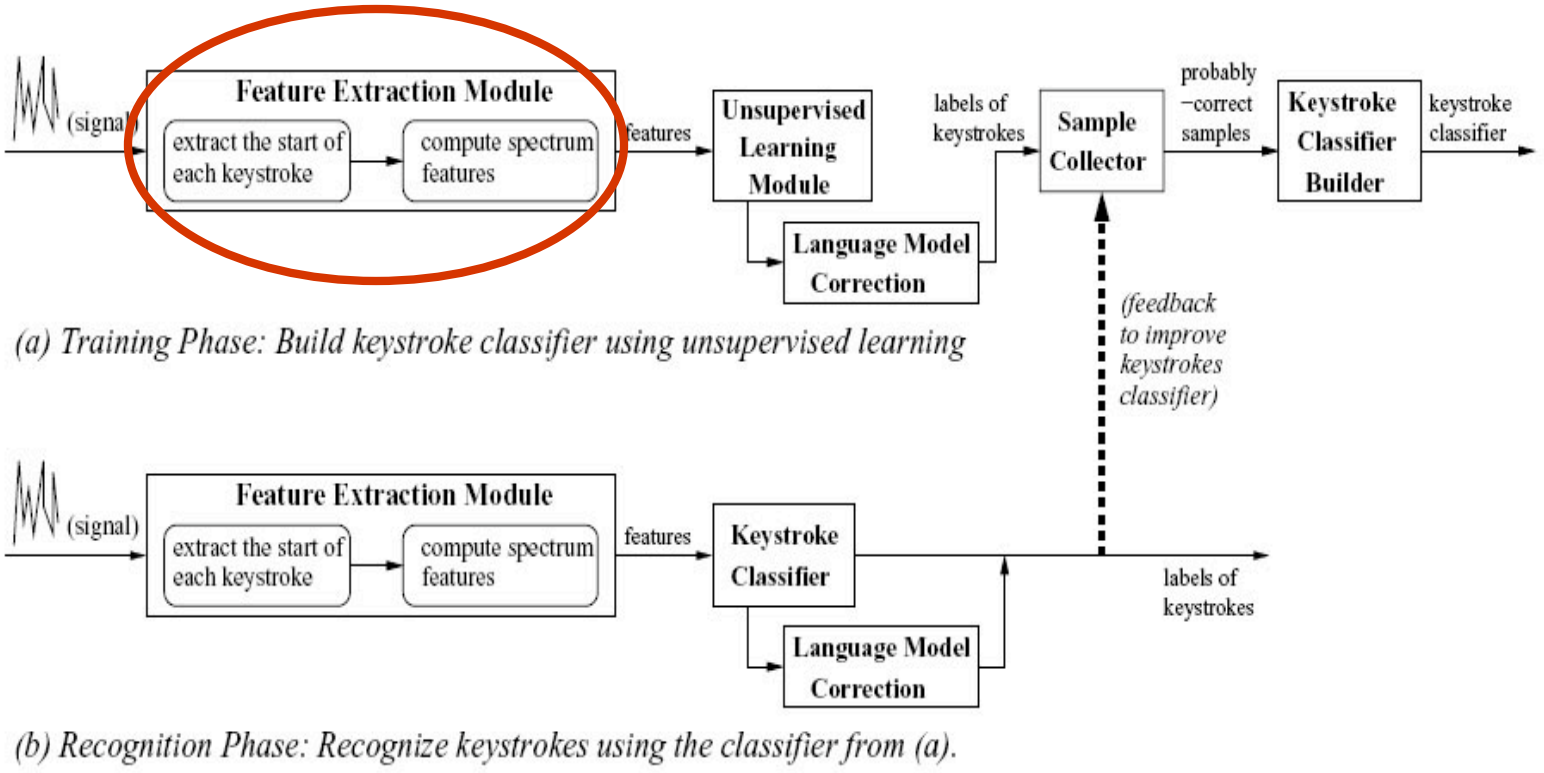
“Keyboard Acoustic Emanations Revisited”

Li Zhuang, Feng Zhou, J.D. Tygar, 2005.

-
- “We Can Do Better!!!”




High Level Overview



(a) Training Phase: Build keystroke classifier using unsupervised learning

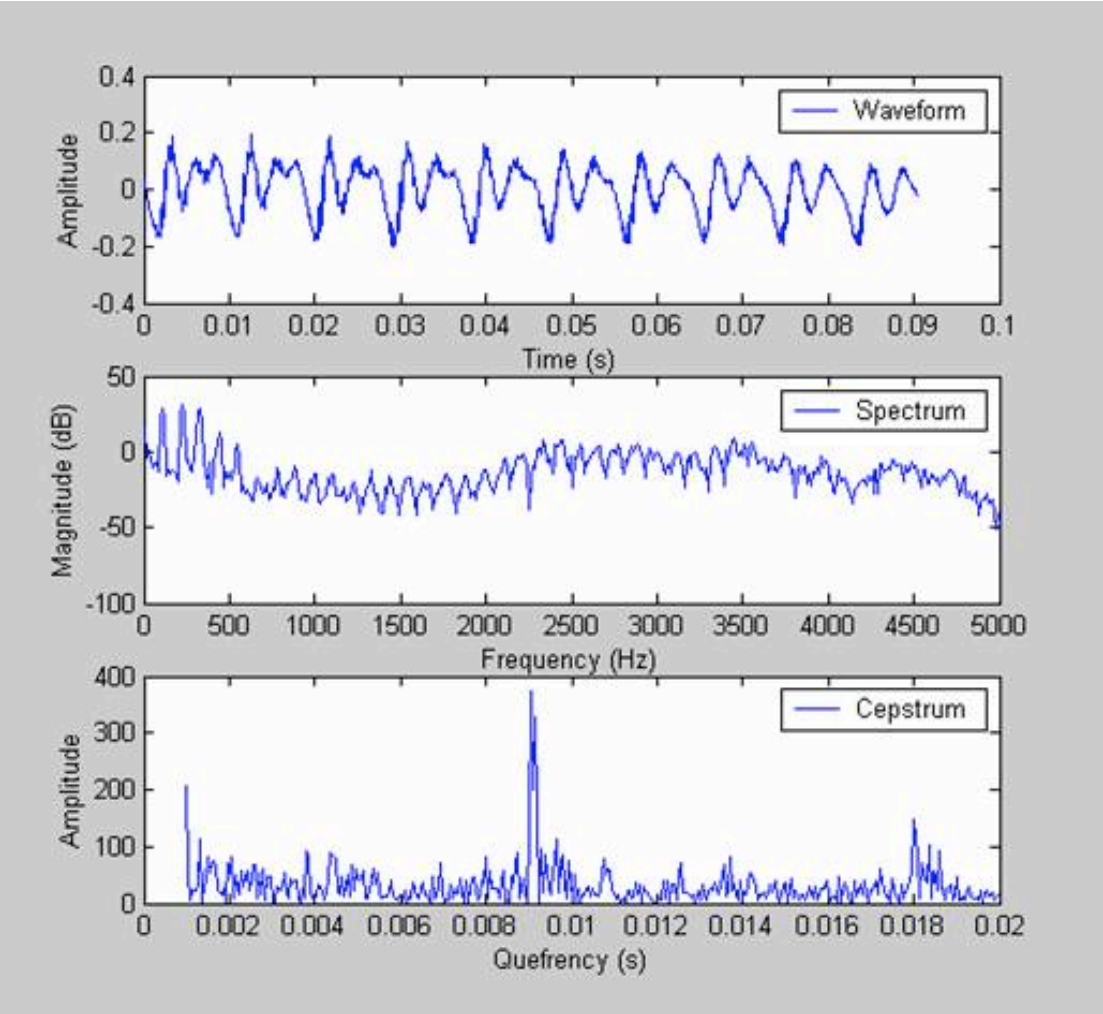
(b) Recognition Phase: Recognize keystrokes using the classifier from (a).

Figure 2: Overview of the attack.

Feature Extraction: Cepstrum Features

- The cepstrum can be seen as information about rate of change in the different spectrum bands
- Use the signal spectrum as another signal, then look for periodicity in the spectrum itself
- signal → FT → log → FT → cepstrum
- cepstrum of signal = FT(log(FT(the signal)))

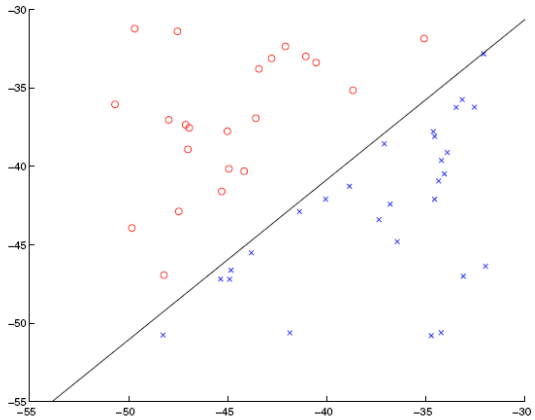
Cepstrum Example



http://www.phon.ucl.ac.uk/courses/spsci/matlab/lect10.html

Linear Classification

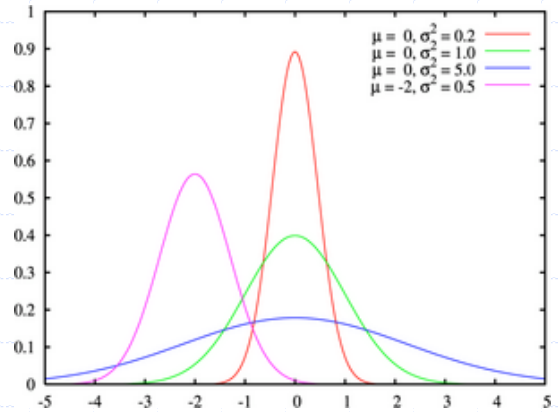
- Simple example with only two dimensions



- Output score = f((vector of weights)·(feature vector))

- Training process finds the best vector of weights to use

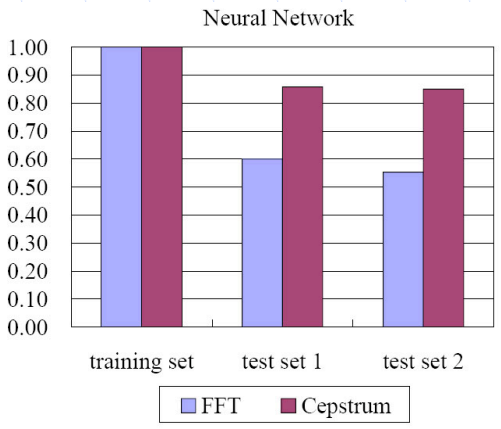
Gaussian Mixtures

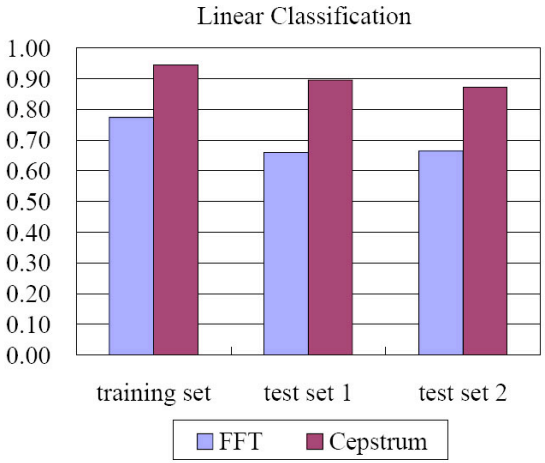


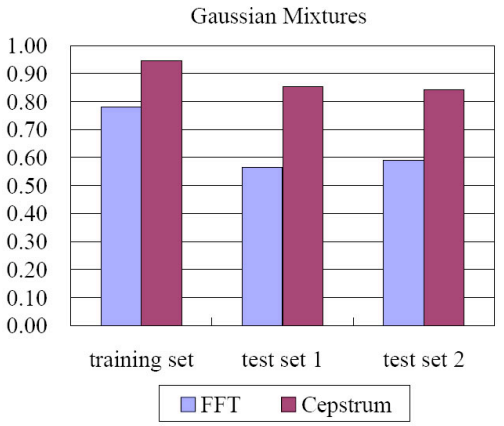
- Used to model many PDFs as a mixture
- Through experimentation they decided to use five gaussian distributions
- When a new feature is analyzed, use the EM algorithm to calculate potential membership

Cepstrum vs FFT

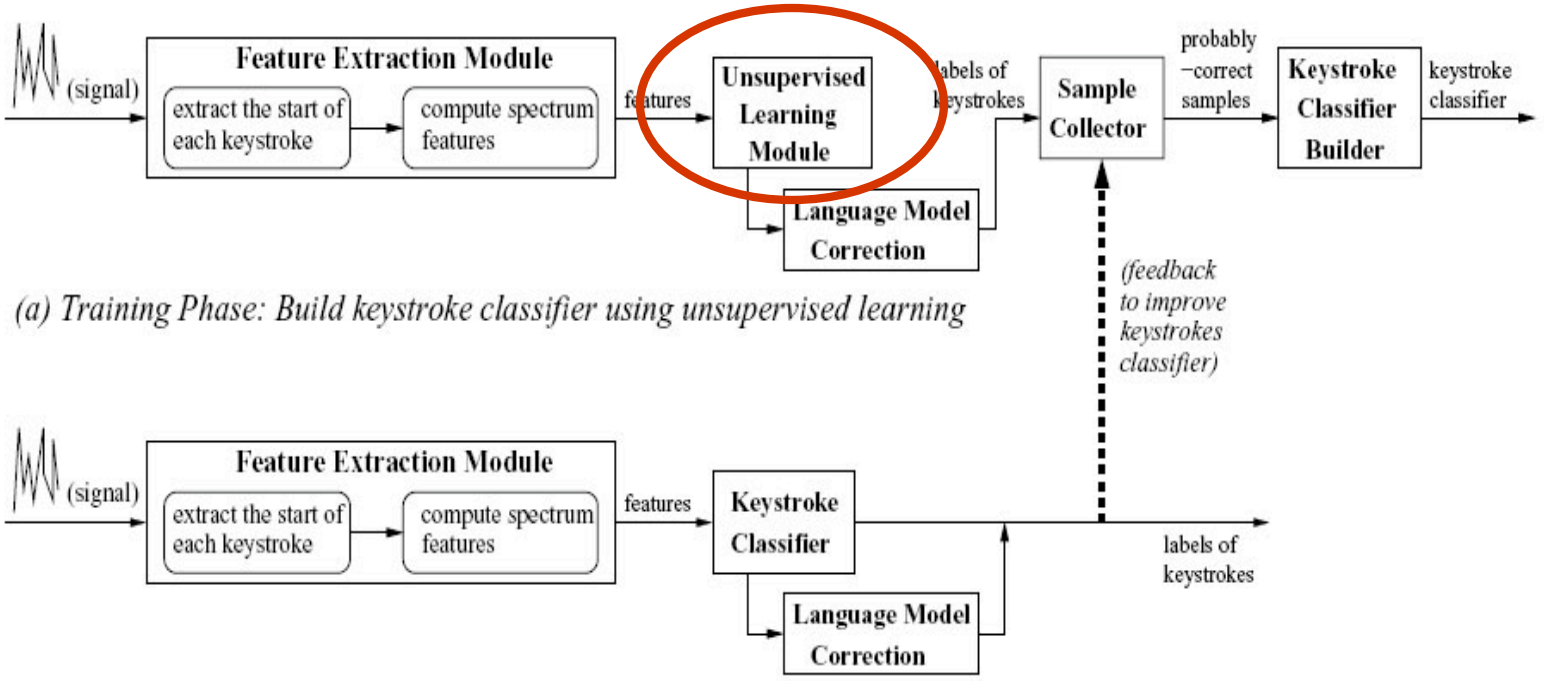
- Linear Classification seems to be the best of the three methods for recognition
- Converted to Mel-Frequency Cepstral Coefficients (scaled to human hearing)
- Done with Matlab newpnn function







High Level Overview



(a) Training Phase: Build keystroke classifier using unsupervised learning

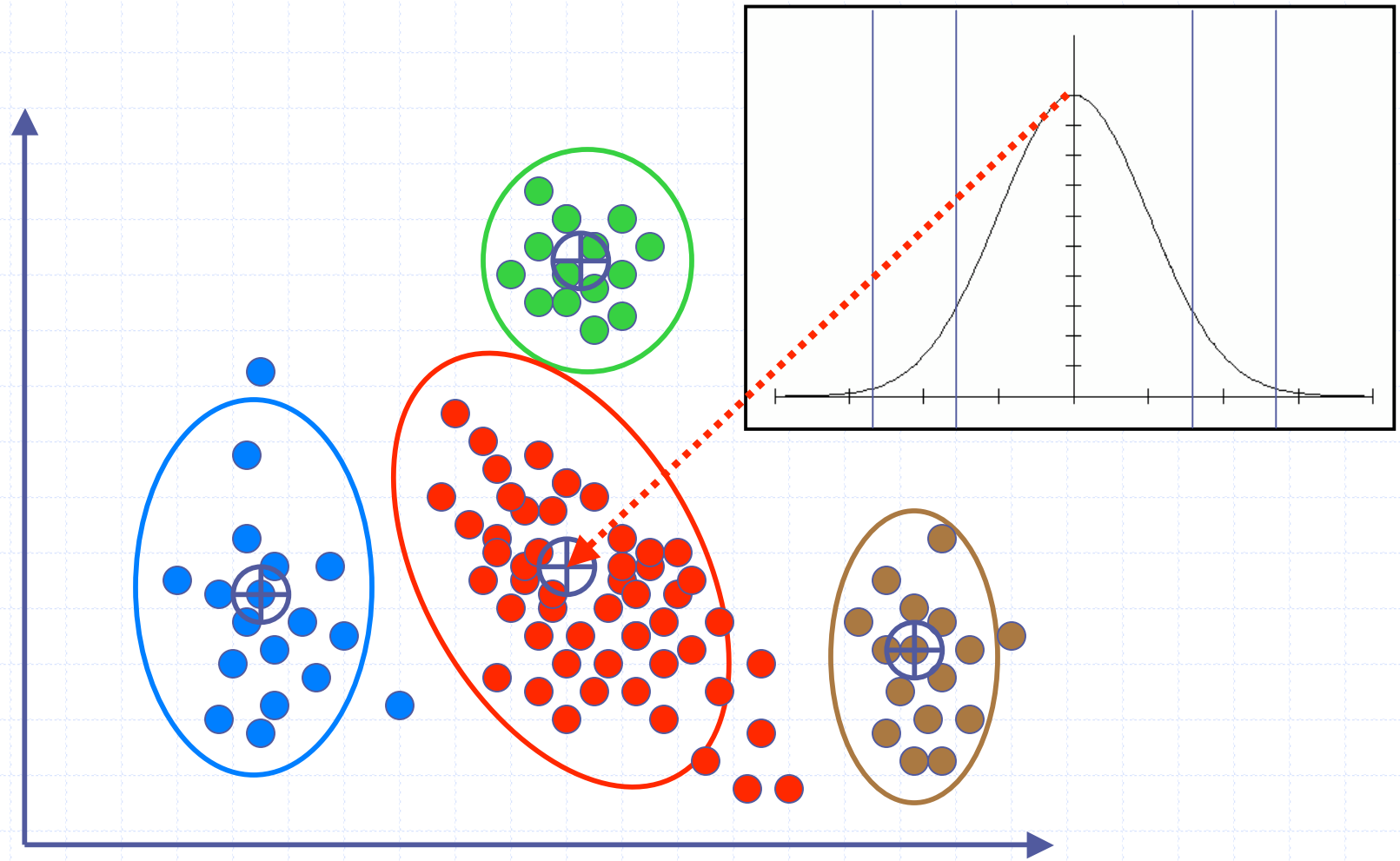
(b) Recognition Phase: Recognize keystrokes using the classifier from (a).

Figure 2: Overview of the attack.

Unsupervised Key Recognition

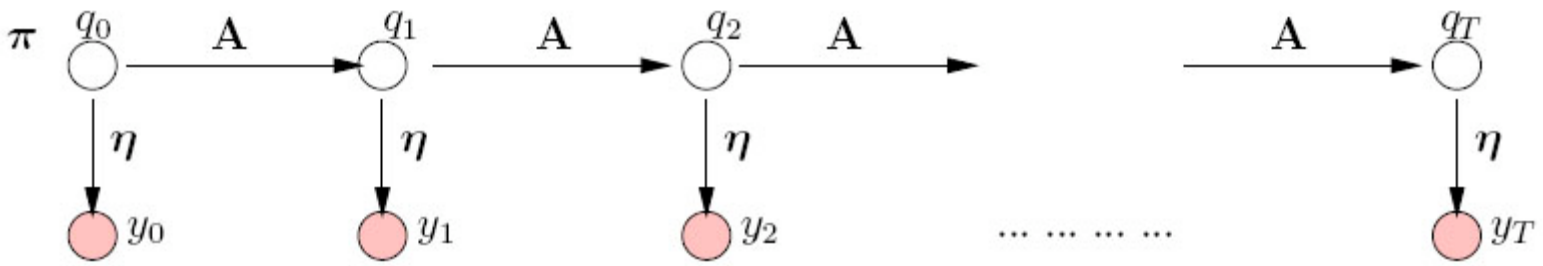
- Cluster each keystroke into K classes
- A particular key will be in each class with a certain probability
- Given a sequence of these keystrokes, they use standard HMM algorithms to identify keys
- 60% accuracy for characters and 20% for words

Simplified K-means



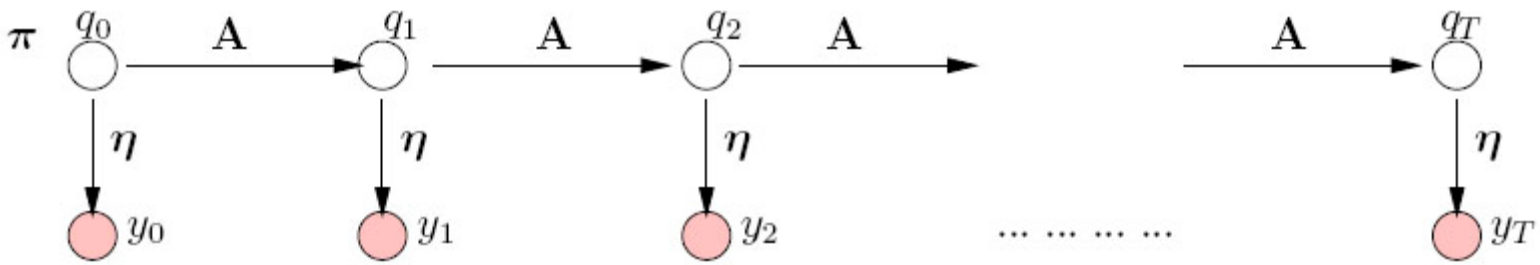
HMM Design

- Shaded circles are observations and unshaded circles are unknown state variables
- A is the transition matrix based on English language
- n is an output matrix (probability of q_i being clustered into class y_i)



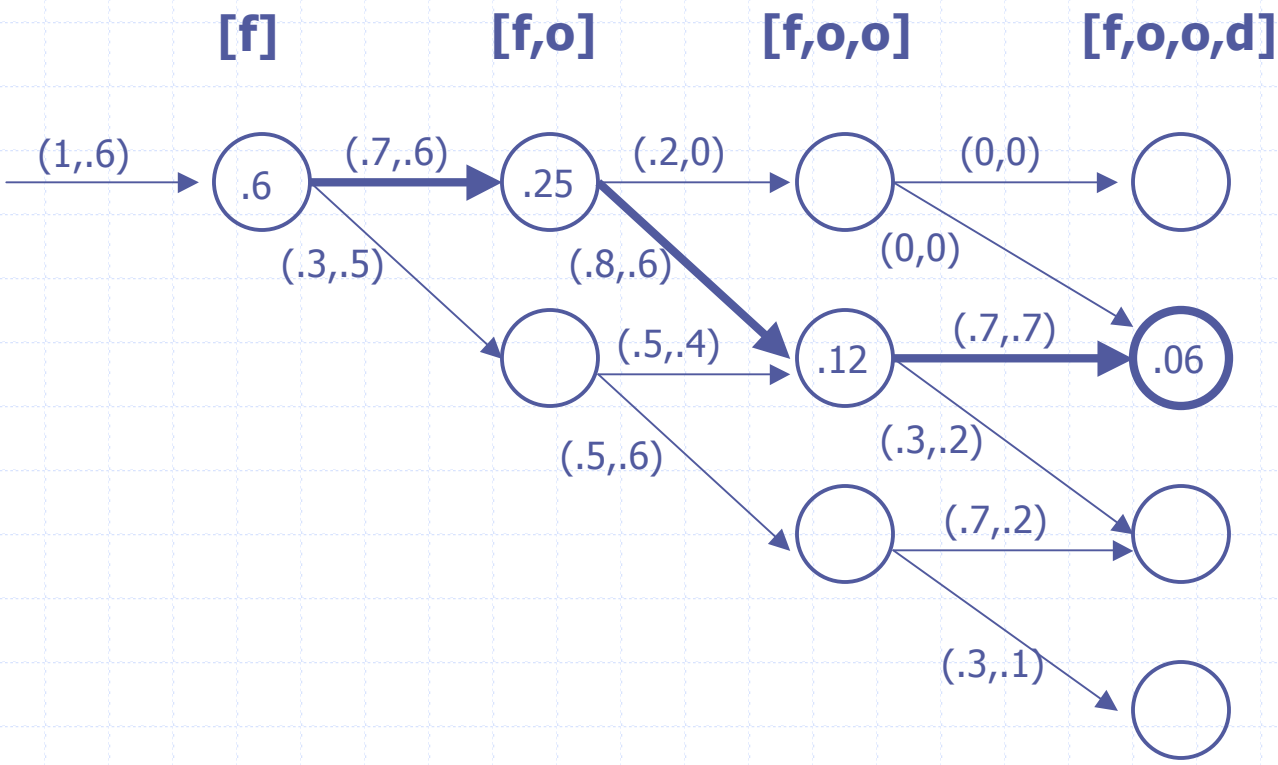
HMM Algorithm

- Expectation Maximization (EM) is used to refine values for the n matrix
- Next the Viterbi algorithm is used to infer the sequences of keys q_i



Viterbi Algorithm

- Finds most probable state that outputs a sequence
- Keeps track of only the most probable states



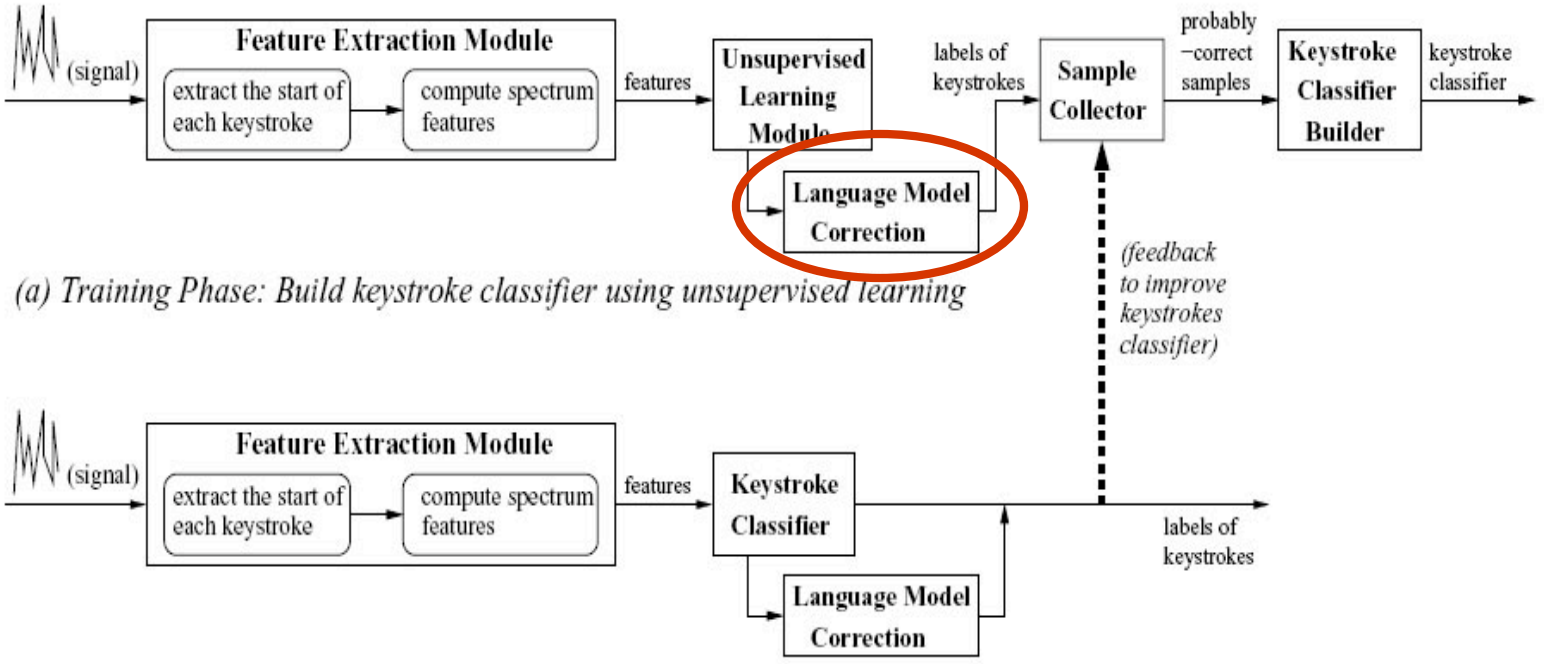
Sample of Original Text

the big money fight has drawn the support of dozens of companies in the entertainment industry as well as attorneys gnnerals in states, who fear the file sharing software will encourage illegal activity, stem the growth of small artists and lead to lost jobs and dimished sales tax revenue.

Detected text

the big money fight has drawn the shoporo od dosens of companies in the entertainment industry as well as attorneys gnnerals on states, who fear the fild shading softwate will encourage illegal acyivitt, srem the grosth of small arrists and lead to lost cobs and dimished sales tas revenue.

High Level Overview



(a) Training Phase: Build keystroke classifier using unsupervised learning

(b) Recognition Phase: Recognize keystrokes using the classifier from (a).

Figure 2: Overview of the attack.

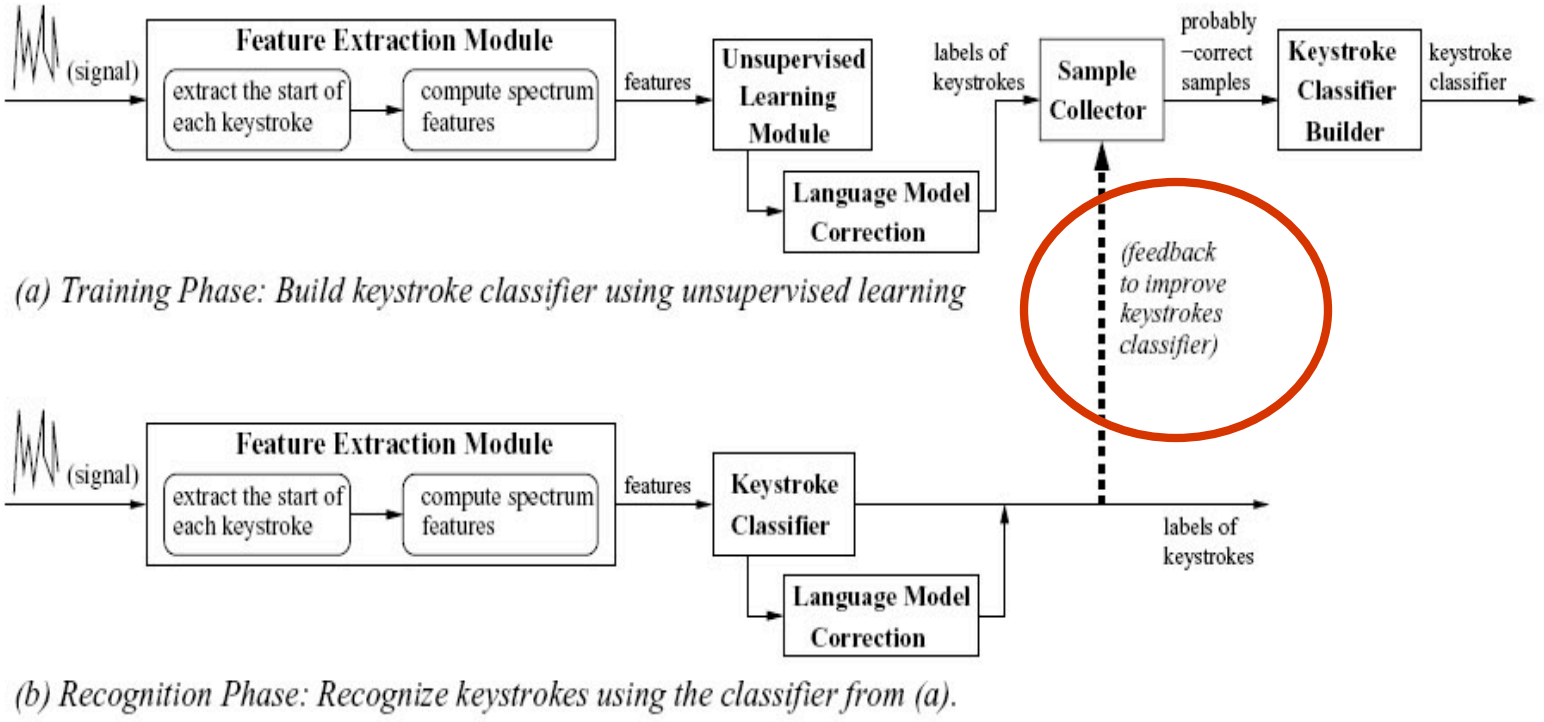
Applying Spelling and Grammar

- Dictionary based spelling (Aspell)
- Applied a simple statistical model of English (n-gram language)
- 70% accuracy for characters and 50% for words

Detected text: Language Model

the big money fight has drawn the support of dozens of companies in the entertainment industry as well as attorneys generals in states, who fear the film sharing software will encourage illegal activity, stem the growth of small artists and lead to lost jobs and finished sales tax revenue.

High Level Overview



(a) Training Phase: Build keystroke classifier using unsupervised learning

(b) Recognition Phase: Recognize keystrokes using the classifier from (a).

Figure 2: Overview of the attack.

Feedback Based Training

- Allows for random text recognition
- Words that were mostly correct are used to train the classifier
- Assume that we know words are mostly correct because the language model only made small corrections

Refine the Classifier

- Run the training set again and use the language model to measure improvement
- Repeat the recognition phase until no improvement is seen (~three times)
- Turn off the language correction and try random character recognition
- Character accuracy improved to 90%

Testing Sets

	Recording Length	Number of Words	Number of Keys	
Set 1	12m 17s	409	2514	Quiet Environment
Set 2	26m 56s	1000	5476	
Set 3	21m 49s	753	4188	Noisy Environment
Set 4	23m 54s	732	4300	

Results: Single Keyboard Recognition

- Language model greatly improves accuracy
- Several rounds of feedback help in noisy environments

		Set 1		*Set 2*		*Set 3*		*Set 4*	
		words	chars	words	chars	words	chars	words	chars
unsupervised learning	keystrokes	34.72	76.17	38.50	79.60	31.61	72.99	23.22	67.67
	language	74.57	87.19	71.30	87.05	56.57	80.37	51.23	75.07
1st supervised feedback	keystrokes	58.19	89.02	58.20	89.86	51.53	87.37	37.84	82.02
	language	89.73	95.94	88.10	95.64	78.75	92.55	73.22	88.60
2nd supervised feedback	keystrokes	65.28	91.81	62.80	91.07	61.75	90.76	45.36	85.98
	language	90.95	96.46	88.70	95.93	82.74	94.48	78.42	91.49
3rd supervised feedback	keystrokes	66.01	**92.04**	62.70	**91.20**	63.35	**91.21**	48.22	**86.58**
	language	**90.46**	**96.34**	**89.30**	**96.09**	**83.13**	**94.72**	**79.51**	**92.49**

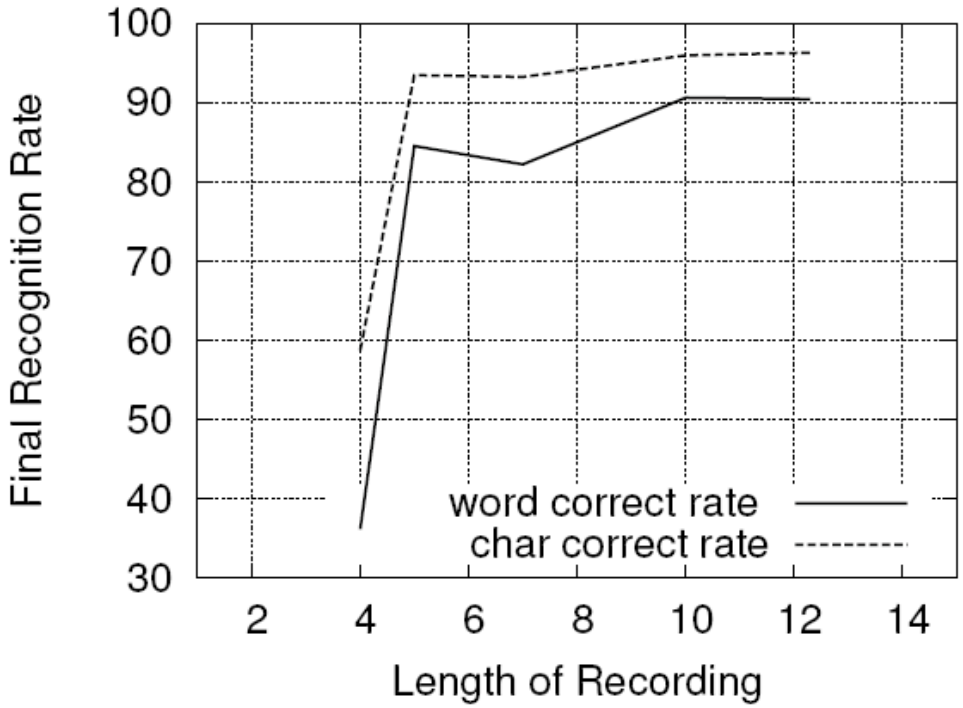
Comparison of Supervised Feedback

- Linear classification performs the best
- Any reason why?

		Neural Network		*Linear Classification*		*Gaussian Mixtures*	
		words	chars	words	chars	words	chars
1st supervised feedback	keystrokes	59.17	87.07	58.19	89.02	59.66	87.03
	language	80.20	90.85	89.73	95.94	78.97	90.45
2nd supervised feedback	keystrokes	70.42	90.33	65.28	91.81	66.99	90.25
	language	81.17	91.21	90.95	96.46	80.20	90.73
3rd supervised feedback	keystrokes	71.39	**90.81**	66.01	**92.04**	69.68	**91.57**
	language	**81.42**	**91.93**	**90.46**	**96.34**	**83.86**	**93.60**

Length of Recording vs. Recognition Rate

- Only need five minutes of recording data to get good recognition rates



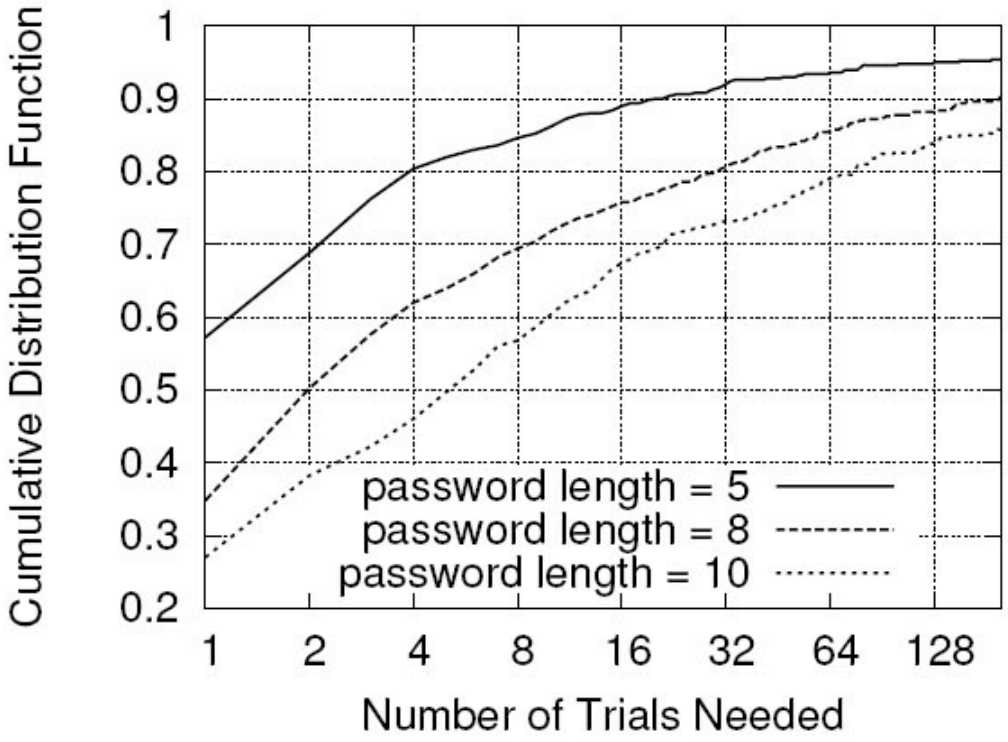
Testing with Multiple Dell Keyboards

- Linear classification was used
- Extra cell phone noise with keyboard 3

		Keyboard 1		*Keyboard 2*		*Keyboard 3*	
		words	chars	words	chars	words	chars
unsupervised learning	keystrokes	30.99	71.67	20.05	62.40	22.77	63.71
	language	61.50	80.04	47.66	73.09	49.21	72.63
1st supervised feedback	keystrokes	44.37	84.16	34.90	76.42	33.51	75.04
	language	73.00	89.57	66.41	85.22	63.61	81.24
2nd supervised feedback	keystrokes	56.34	88.66	54.69	86.94	42.15	81.59
	language	80.28	92.97	76.56	91.78	70.42	86.12
Final result	keystrokes	60.09	**89.85**	61.72	**90.24**	51.05	**86.16**
	language	**82.63**	**93.56**	**82.29**	**94.42**	**74.87**	**89.81**

Random Text Recognition (Got Root?)

- Trained with Set 1 and used with randomly generated sequences



Attack Improvements

- Extra keys (i.e. tab, backspace, shift)
- Other language models
- Application specific (IDEs, editors)
- Remove backgound noise
- Hierarchical Hidden Markov Model

Defenses

- Physical Security
- Use of “quieter” keyboards
- Introduce background noise
- Two-Factor authentication

Extensions

- What about overlapping keystrokes or very fast typists? Dvorak keymapping?
- Do long fingernails play a role?
- Possible for someone to snoop your keyboard remotely through IM or VoIP?

Related Ideas

- Emotive Alert: HMM-Based Emotion Detection in Voicemail Messages (Z. Inanoglu, R. Caneel)
- Statistical Identification of Encrypted Web Browsing Traffic (Q. Sun et al)
- Questions?