Side Channel Attacks
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- **TEMPEST** - leaked electromagnetic radiation can directly provide plaintexts and other information.
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- **TEMPEST** - leaked electromagnetic radiation can directly provide plaintexts and other information.
- **Timing** - some crypto computations take different amounts of time e.g. squaring and multiplying
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- **TEMPEST** - leaked electromagnetic radiation can directly provide plaintexts and other information.
- **Timing** - some crypto computations take different amounts of time e.g. squaring and multiplying
- **Power Monitoring** - some microprocessor operations take measurably more power than others.
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- **TEMPEST** - leaked electromagnetic radiation can directly provide plaintexts and other information.
- **Timing** - some crypto computations take different amounts of time e.g. squaring and multiplying
- **Power Monitoring** - some microprocessor operations take measurably more power than others.
- **Acoustic Cryptanalysis** - exploit sound produced during a computation.
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- TEMPEST - leaked electromagnetic radiation can directly provide plaintexts and other information.
- Timing - some crypto computations take different amounts of time e.g. squaring and multiplying
- Power Monitoring - some microprocessor operations take measurably more power than others.
- Acoustic Cryptanalysis - exploit sound produced during a computation.
- Differential Fault Analysis - introduce faults in a computation.
Side Channel Attacks

Exploitation of some aspect of the physical part of a system on which some cryptographic algorithm is run to learn the crypto-variables associated with that algorithm

- **TEMPEST** - leaked electromagnetic radiation can directly provide plaintexts and other information.
- **Timing** - some crypto computations take different amounts of time e.g. squaring and multiplying.
- **Power Monitoring** - some microprocessor operations take measurably more power than others.
- **Acoustic Cryptanalysis** - exploit sound produced during a computation.
- **Differential Fault Analysis** - introduce faults in a computation.
- **Long Distance Observation** – look at monitor screen or a reflection of it in glasses, etc.
Side Channel Attacks

TEMPEST:

Read electromagnetic radiation from the crypto device.

1943 – Bell Labs top secret encrypted teletype terminal Model 131-B2 (Python) for Army and Navy “unbreakable codes” – used one-time tapes

Sigaba M134C
Used alongside the model 131-B2

http://www.jproc.ca/crypto/tempest.html
Side Channel Attacks

Model 131-B2:

Designed by Bell Labs for secure military communication.

After delivery, someone noticed oscilloscope patterns across the lab every time a letter was encrypted.
Side Channel Attacks

Model 131-B2:

Designed by Bell Labs for secure military communication.

After delivery, someone noticed oscilloscope patterns across the lab everytime a letter was encrypted.

Military was alerted – response was “Are you kidding?” We don't believe it – let's run a test.

Bell people were able to intercept 75% of the plaintext from 80 feet away.

Military said fix it – Bell Labs modified design to include shielding (radiation) and filtering (powerlines) resulting in a device (131-A-1) that could not be modified in the field.
Side Channel Attacks

Model 131-B2:

- Designed by Bell Labs for secure military communication.
- After delivery, someone noticed oscilloscope patterns across the lab every time a letter was encrypted.
- Military was alerted – response was “Are you kidding?”
- We don't believe it – let's run a test.
- Bell people were able to intercept 75% of the plaintext from 80 feet away.
- Military said fix it – Bell Labs modified design to include shielding (radiation) and filtering (powerlines) resulting in a device (131-A-1) that could not be modified in the field.
- Military did not accept it – instead, warned commanders to secure an area of radius 100 feet around the device when using it.
Side Channel Attacks

Model 131-B2:

Designed by Bell Labs for secure military communication.

After delivery, someone noticed oscilloscope patterns across the lab everytime a letter was encrypted.

Military was alerted – response was “Are you kidding?”

We don't believe it – let's run a test.

Bell people were able to intercept 75% of the plaintext from 80 feet away.

Military said fix it – Bell Labs modified design to include shielding (radiation) and filtering (powerlines) resulting in a device (131-A-1) that could not be modified in the field.

Military did not accept it – instead, warned commanders to secure an area of radius 100 feet around the device when using it.

The war ended, people went home, issue was forgotten.
Side Channel Attacks

Model 131-B2:

But the model 131-B2 was still used after the war!

In 1951 the CIA discovered they could recover plaintext ¼ mile away on power lines!
Side Channel Attacks

Model 131-B2:

But the model 131-B2 was still used after the war!

In 1951 the CIA discovered they could recover plaintext ¼ mile away on power lines!

Filters were added – power line problem was solved but radiated signals continued. Advisories were issued:

1. secure an area of 200 feet when using
2. operate 10 machine simulataneously (masking)
3. get a waiver based on operational necessity
Model 131-B2:

But the model 131-B2 was still used after the war!

In 1951 the CIA discovered they could recover plaintext ¼ mile away on power lines!

Filters were added – power line problem was solved but radiated signals continued. Advisories were issued:

1. secure an area of 200 feet when using
2. operate 10 machine simulataneously (masking)
3. get a waiver based on operational necessity

Then someone realized that other equipment might have the same problem – sure enough lots did, pretty bad
Side Channel Attacks

Model 131-B2:

But the model 131-B2 was still used after the war!

In 1951 the CIA discovered they could recover plaintext ¼ mile away on power lines!

Filters were added – power line problem was solved but radiated signals continued. Advisories were issued:
   1. secure an area of 200 feet when using
   2. operate 10 machine simulataneously (masking)
   3. get a waiver based on operational necessity

Then someone realized that other equipment might have the same problem – sure enough lots did, pretty bad

By 1955:
   1. filtering techniques improved
   2. teletype relays made to energize simultaneously but the size of the spike was enough to read a char.

It was noticed that the sound of the machine was enough
Soundproofing a room made it easier for the attacker

Then, in 1962, someone was walking around the 200 foot perimeter that needed to be secure to use the crypto devices, somewhere in Japan, and noticed a strange, partially hidden antenna on a hospital carport. He notified counter intelligence – they said study it and send back details at first light. But the next day it was gone!
Model 131-B2:

Soundproofing a room made it easier for the attacker

Then, in 1962, someone was walking around the 200 foot perimeter that needed to be secure to use the crypto devices, somewhere in Japan, and noticed a strange, partially hidden antenna on a hospital carport. He notified counter intelligence – they said study it and send back details at first light. But the next day it was gone!

In 1964, 40 microphones and a large metal grid cemented into the ceiling were found in the embassy in Moscow.

See the report for details.
Side Channel Attacks

Other TEMPEST Attacks:

1985 – Van Eck Phreaking proof of concept computer monitors emit radiation related to fonts that appear on a screen
2006 – Dutch government bans use of NewVote voting machines due to fear that election secrecy voided
2009 – Secrecy of Brazil elections compromised
2009 – Most common wireless/wired keyboards vulnerable lately – DVI cables are particularly vulnerable
Side Channel Attacks

Other TEMPEST Attacks:

1985 – Van Eck Phreaking proof of concept computer monitors emit radiation related to fonts that appear on a screen
2006 – Dutch government bans use of NewVote voting machines due to fear that election secrecy voided
2009 – Secrecy of Brazil elections compromised
2009 – Most common wireless/wired keyboards vulnerable lately – DVI cables are particularly vulnerable

Countermeasures:
Filter high-frequency spectral components from fonts before they are rendered on a monitor.
Add noise to drive the eavedropper's equipment into Saturation.
Use shielded keyboards and computers – expensive!
Encrypt transmissions
Side Channel Attacks

Timing Attacks – based on different time operations take on a particular machine – people like Joe Young and Rob Lancaster try to design circuits that improve speed when possible (eliminate unnecessary conditionals, reduce RAM cache hits).

Can lead to:
- Information about secret keys – bits, length, RSA key factors, Diffie-Hellman exponents

Requires:
- Some known plaintext
- Almost no resources, no expensive resources
Side Channel Attacks

Timing Attacks – A simple modular exponentiation

Foundation of RSA and DH: \( y^x \mod n \)
y is found by eavesdropping, \( n \) is public

Let \( R_1 = 1 \)
For \( k = w-1 \) downto 0:
    If (bit \( k \) of \( x \)) is 1 then
        Let \( R_0 = R_1 \cdot y \mod n \)
    Else
        Let \( R_0 = R_1 \)
        Let \( R_1 = R_0 \mod n \)
EndFor
Return \( R_0 \)
Side Channel Attacks

Let $R_1 = 1$
For $k = w-1$ downto 0:
    If (bit $k$ of $x$) is 1 then
        Let $R_0 = R_1 \cdot y \mod n$
    Else
        Let $R_0 = R_1$
    Let $R_1 = R_0 \cdot R_0 \mod n$
EndFor
Return $R_0$

**Rough idea:** assume mod multiplication is a lot slower for some $y$ than modular squaring.

Then sometimes $R_0 = R_1 \cdot y \mod n$ is slow.
From system knowledge, $y$ and $n$ that cause this will be known if all previous bits are known.

When attacking, if algorithm is fast on bit $k$ when it should be slow, $k$ is 0. If algorithm is always slow on bit $k$, it is prob 1.
Find first key bit, then second bit, ...
Side Channel Attacks

**Specific Attack**: given $j$ random $y_0, y_1, \ldots, y_{j-1}$ and timing measurements $T_0, T_1, \ldots, T_{j-1}$, and let $e$ be a time that includes measurement error, loop overhead, and such. Let $t_{i,k,b}$ be the time it takes for the $k^{\text{th}}$ iteration on message $y_i$ with keybit $b$. Attacker observes ($w =$ exponent size)

$$T_i = \sum_{0 \leq k \leq w-1} (x_k t_{i,k,1} + (1-x_k) t_{i,k,0}) + e$$

Given guess $g_b$ and guessed bits $g_0 \ldots g_{b-1}$, the attacker can compute

$$T_{i,b} = \sum_{0 \leq k \leq b-1} (g_k t_{i,k,1} + (1-g_k) t_{i,k,0})$$

For each $y_i$, the difference is

$$D_{i,b} = \sum_{0 \leq k \leq b} ((x_k-g_k) t_{i,k,1} + (g_k-x_k) t_{i,k,0}) + \ldots$$

If of first $b$ bits $c$ are incorrect then

$$\text{Var}(D_{i,b}) = c(\text{Var}(t_{k,0}) + \text{Var}(t_{k,1})) + \ldots$$
**Side Channel Attacks**

**Specific Attack**: $n$ is fixed, 512 bit exponent, generate 250 timing measurements.

![Histograms showing time distribution for mod multiplication and mod exponentiation](image)

- **mod multiplication**
- **mod exponentiation**

Probability that subtracting the time for a correct bit is going to reduce the variance more than subtracting time for an incorrect bit is about 0.88.
Side Channel Attacks

**RSA**: Instead of computing \( m^e \mod n \) compute
\[ m^e \mod p \cdot m^e \mod q \]

Choose values for \( m \) that are (believed) close to \( p \) or \( q \) and use timing measurements to determine whether \( m \) is larger or smaller than the actual value. If larger, Subtract \( p \) from \( m \) at least once.

**DSS**: computes \( s = (k^{-1}(H(m)+ x \cdot r)) \mod q \) where
\( k^{-1} \) is precomputed, \( H(m) \) is the hash of the message, \( x \) is the private key, \( r \) and \( q \) are assumed known by all

\( H(m) \) has little effect on timing variance, also \( k^{-1} \). Timings correlate with upper bits of \( x \).
Remedies for Timing Attacks:

Unfortunately, it is hard to make software run in fixed time across platforms due to compiler optimizations and variations, RAM cache hits, etc.

Executing all statements, possibly throwing out some results does not always work but will increase the number of samples needed.
Remedies for Timing Attacks:
Unfortunately, it is hard to make software run in fixed time across platforms due to compiler optimizations and variations, RAM cache hits, etc.

Executing all statements, possibly throwing out some results does not always work but will increase the number of samples needed.

\[
\begin{align*}
R0 &= 1 \\
R1 &= y \\
&\text{For } k = w-1 \text{ downto } 0: \\
&\quad \text{If (bit } k \text{ of } x \text{ is 0) then} \\
&\qquad R1 = R0 \cdot R1 \mod n \\
&\qquad R0 = R0 \cdot R0 \mod n \\
&\quad \text{Else} \\
&\qquad R0 = R0 \cdot R1 \mod n \\
&\quad R1 = R1 \cdot R1 \mod n \\
&\text{Return } R0
\end{align*}
\]
Side Channel Attacks

Remedies for Timing Attacks:
Injecting random delays increases the number of samples needed.
Side Channel Attacks

Remedies for Timing Attacks:

Injecting random delays increases the number of samples needed.

Before computing the mod exp algorithm, choose random \( v_1, v_2 \), such that \( \text{inv}(v_1) = v_2^x \mod n \)

Multiply message \( m \) by \( v_2 \mod n \) then, when finished with the mod exponentiator multiply by \( v_1 \mod n \).
Side Channel Attacks

Simple Power Analysis Attacks:
Possible because circuit power consumption changes drastically depending on the operations performed and there is equipment that can measure power changes sampling at even 1 GHz speed.
Side Channel Attacks

Simple Power Analysis Attacks:
Possible because circuit power consumption changes drastically depending on the operations performed and there is equipment that can measure power changes sampling at even 1 GHz speed.

SPA trace showing an entire DES operation

SPA trace showing DES rounds 2 and 3
Side Channel Attacks

Simple Power Analysis Attacks:

SPA DES trace showing differences in power consumption of different microprocessor instructions.
Side Channel Attacks

Simple Power Analysis Attacks:

- SPA can reveal sequence of instructions executed
- It can be used to break cryptographic implementations in which the execution path depends on the data being processed
  - DES key schedule
  - DES permutations
  - Comparisons
  - Multipliers
  - Exponentiators
Side Channel Attacks

Preventing Simple Power Analysis Attacks:

Avoid writing procedures that branch depending on some crypto-variable.

Use hard-wired implementations of symmetric crypto Algorithms – most implementations are OK.

The microcode in some microprocessors cause large power consumption differences depending on operands - must fix.
Side Channel Attacks

Differential Power Analysis Attacks:

Statistical analysis similar to timing attacks but looking at power usage instead.

Example: DES

\[
D(C,b,K_i) = \text{computes the value of bit } b \text{ of number } Left \\
\text{on the } 16^{th} \text{ round for ciphertext } C, \text{ and } K_i \text{ is the } 6 \text{ bit key entering the S-box.}
\]

Observe: if \( K_i \) is wrong, \( D(C,b,K_i) \) is wrong with probability \( \frac{1}{2} \) for each ciphertext.

Given: \( m \) encryption operations capturing power traces \( T_{1...m}^{[1..k]} \) containing \( k \) samples each and ciphertexts \( C_{1...m} \). No knowledge of the plaintext is required.

If average over \( C_{1...m} \) of difference \( D(C,b,K_i) - \bar{T} \) tends to 0, then must have the wrong \( K_i \).
The Mangler function: mixes 32 bit input with 48 bit key to produce 32 bits

1. Expansion of input bits:

\[
\begin{align*}
32 \text{ bit } R_i & \\
\oplus & \\
K_i & \\
\oplus & \\
32 \text{ bit } R_{i+1} & \\
\end{align*}
\]

\[
\begin{array}{cccccccc}
4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} & 4 \text{ bits} \\
6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} & 6 \text{ bits} \\
\end{array}
\]

Expanded input (48 bits)

2. Mixing with key:

\[
\begin{align*}
R_i (32 \text{ bits}) & \\
\oplus & \\
K_i (48 \text{ bits}) & \\
\oplus & \\
R_{i+1} (32 \text{ bits}) & \\
\end{align*}
\]
Side Channel Attacks

**DES:**

Hence correct value for $b$ is identified by spikes in the differential trace.
Side Channel Attacks

DES:

Get all 8 8-bit S-box keys for 48 bit round subkey – use exhaustive search to get remaining 8 key bits for a 56 bit 16th round key. Then work upward to get all per round keys. Reveal 3-DES keys by analyzing outer DES 1st.
Side Channel Attacks

Preventing Differential Power Analysis Attacks:

Use shielding around computer. Isolate power line using some auxiliary power supply. This is unfortunately expensive and only means the attacker needs to take more samples.

Temporal obfuscation: inject noise into the signal. Unfortunately, some methods for doing this can be circumvented and there are examples where systems have been marked secure but were not.

Cryptosystem design meant for specific hardware
Side Channel Attacks

Differential Fault Analysis:

Observe behavior of a device by causing it to operate outside of its intended specifications.

- High voltage
- Overclocking
- Radiation of various kinds

Encrypt the same piece of data twice, under normal and abnormal operation of the device.
Side Channel Attacks

Differential Fault Analysis:

Observe behavior of a device by causing it to operate outside of its intended specifications.

- High voltage
- Overclocking
- Radiation of various kinds

Encrypt the same piece of data twice, under normal and abnormal operation of the device.

**Example:** DES on a smartcard

**Assume:**
- Random, transient faults are not uncommon in registers
- When a fault occurs it is in one bit in some round
- A fault is a bit inversion
- Fault location and timing are not known to attacker
- For simplicity, assume fault is one of 512 right-half bits
Side Channel Attacks

The Attack:
Attacker encrypts same plaintext twice until different
Side Channel Attacks

The Attack:
Attacker encrypts same plaintext twice until different
The Attack:

Attacker encrypts same plaintext twice until different

Suppose the fault is in $R_{15}$
Side Channel Attacks

The Attack:
Attacker encrypts same plaintext twice until different

Each Sbox:

guess $K_{16,i}$ and check

Most keys are eliminated

Do this about 200 times
**Side Channel Attacks**

**Branch Prediction Analysis:**

Modern superscalar processors (Pentium...):

Pipelining – split instruction into pieces

<table>
<thead>
<tr>
<th>Instr. No.</th>
<th>Pipeline Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>2</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>3</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>4</td>
<td>IF ID EX MEM</td>
</tr>
<tr>
<td>5</td>
<td>IF ID EX</td>
</tr>
</tbody>
</table>

**5 stage pipeline**

IF = instruction fetch
ID = instruction decode
EX = execute
MEM = memory access
WB = register write back
Side Channel Attacks

Branch Prediction Analysis:

Modern superscalar processors (Pentium...):

Pipelining – split instruction into pieces

<table>
<thead>
<tr>
<th>Instr. No.</th>
<th>Pipeline Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>2</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>3</td>
<td>IF ID EX MEM WB</td>
</tr>
<tr>
<td>4</td>
<td>IF ID EX MEM</td>
</tr>
<tr>
<td>5</td>
<td>IF ID EX</td>
</tr>
</tbody>
</table>

5 stage pipeline

IF = instruction fetch
ID = instruction decode
EX = execute
MEM = memory access
WB = register write back

Branch Prediction Unit – predict direction of case split

Branch Target Buffer – record of target addresses of previously executed branches. Limited size – entries may be evicted to allow new address to be stored.

Branch History Registers – store the last $n$ outcomes of a branch – 011001...

Branch Prediction Table – for each branch: states: strongly taken to strongly not taken
Side Channel Attacks

Branch Prediction Analysis:
Modern superscalar processors (Pentium...):

Simultaneous Multithreading:
2 to 8 threads in hardware
Instructions from multiple threads can be executed at the same time
Some “cheap” CPU resources are duplicated to support parallel execution of threaded instructions

**Note:** the BPU is shared among threads – hence the opportunity exists for metadata leaks from one process to another
Side Channel Attacks

Branch Prediction Analysis:

Differential Branch Prediction Analysis Attack outline:

Run a spy process on a thread – assume it is running
In sync with a threaded crypto process

The spy process repeats the following:
• executes a series of branch instructions
• measures the time it takes to execute them

The branch instructions are designed to cause BPT evictions of targets previously stored by the crypto

If crypto process results in branch-taken:
Spy process branch must be evicted from the BPT
Spy process re-executes to evict the crypto process
The extra time means crypto had taken the branch

Otherwise:
Nothing evicted from the BPT
Spy process re-executes in less time – crypto process non-branch is revealed
Side Channel Attacks

Branch Prediction Analysis:

Simple Branch Prediction Analysis:

Acquire a key in a *single execution* of RSA signing, say

Attack Outline:

Reason for statistical analysis in DBPA is *noise* due to one or more other threaded processes running as well. Assume that running of another process happens randomly. Then there will be moments when no other process is running and times for this moment will be clear.

A target address may be stored in more than one place in the BPT – hence it is necessary to evicted all these occurrences and the spy branch instructions are designed to do this. The number of such branch instructions is an important parameter.
Branch Prediction Analysis:

Simple Branch Prediction Analysis:

Acquire a key in a *single execution* of RSA signing, say Square-and-multiply exponentiation, 512 bit key, 508 bits recovered successfully.

Similar results for Montgomery Powering Ladder.
Side Channel Attacks

Acoustic Cryptanalysis:

- Keyboard keypresses sound slightly different
- Keys on telephones and ATM machines can emit distinctive sound patterns
- CPU may emit a “hum” which changes depending on operations it is performing
- Impact printers may emit distinctive sounds
- Different times for operations under different inputs may enable timing attacks

Method:

- Record the sounds for off-line processing
- Apply filtering (FFT, neural network, ...) to translate to characters and/or numbers
Side Channel Attacks

Acoustic Cryptanalysis:
The “hum” of a CPU can be heard over fan and other computer and room noise because the “hum” is high freq.
The microphone can be placed as far away as 6 feet from the computer

Experiments:
Condenser microphone ($170)
Mixer for amplification and equalization ($55)
Creative Labs soundcard ($70)
Motherboard has Intel Celeron at 666MHz
PC case open
Microphone at 8” from computer
Signal analysis software
GunPG signing a message with a 4096 bit RSA key. Exponentiation mod p then mod q.
GunPG signing several messages with 4096 bit RSA keys
Side Channel Attacks

Result of freezing a bank of 1500 μf capacitors on the motherboard during while executing a loop of MUL instructions
Side Channel Attacks

Monitor Eavesdropping, Optical, Short/Long Distance:

Examples:
  Reflections from teapots, eyeglasses, eyes, glasses, even a wall that is in the vicinity of a monitor which is turned away from an attacker.

  Telescopic observations from a helicopter or blimp of a monitor facing a window.
The following are some experiments using the above test pattern on a 15” laptop screen pointing away from a window in a typical office.

Cite: Backes, Dürmuth, Unruh “Compromising Reflections or How to Read LCD Monitors Around the Corner,” Saarland University, Saarbrücken, Germany.
Side Channel Attacks

Monitor Eavesdropping, Optical, Short Distance:

Reflections from coke bottle and eyeglasses at 15 feet, 10 MegaPixel camera, indoors, 2 second exposure time (while eyeglasses worn)
Side Channel Attacks

Monitor Eavesdropping, Optical, Short Distance:

Reflections from a spoon and wineglass at 15 feet, 10 MegaPixel camera, indoors, 2 second exposure time (while eyeglasses worn)
Side Channel Attacks
Monitor Eavesdropping, Optical, Long Distance:

Reflections from a teapot at 120 feet, 10 MegaPixel camera, indoors, 2 second exposure time, small telescope

Reflections from a teapot at 90 feet with a larger telescope
Side Channel Attacks

Monitor Eavesdropping, Optical, Long Distance:

Reflections off a wall (middle) and after processing with Deconvolution, gamma reduction, and edge detection algorithms

An LCD monitor pixel produces slightly directed light
Monitor Eavesdropping, Optical, Noise Reduction:

1. take many shots with short exposure time, choose 10% of the best and add them to get properly exposed image

2. Deconvolution and other Image Processing algorithms:
   Weiner: $F(s) \times H(s) = R(s)$, estimate $H(s)$ and use $H^{-1}(s)$
   Non-blind, Adaptive blind, No/Nearest neighbor:
   2D blind, 2D real-time,...

3. Take high definition pictures of small areas and then piece them together like a jigsaw puzzle
Monitor Eavesdropping, EM:

Graphics cards need to generate pixel images with sharp rise and fall times.

Has the effect of multiplying a (relatively low frequency) display signal by pulses at high frequency – hence there are many aliased sections of the frequency spectrum of a signal directed to a monitor.

Tune a receiver to one of those sections, demodulate, and recover the base signal.

Factors in choosing where to tune:

- Radiation efficiency increases with frequency.
- Emitted harmonics decrease with increasing frequency.
- Chosen band should be free of broadcast signals.
Side Channel Attacks

Monitor Eavesdropping, EM:

Video graphics adapter pixel shapes

- Matrox Millennium II (157.5 MHz)
- ATI 3D Rage Pro (157.5 MHz)
- Toshiba 440CDX (49.5 MHz)
Side Channel Attacks

Monitor Eavesdropping, EM:

Video graphics adapter signal spectra (100 kHz bandwidth)

- Matrox Millennium II (157.5 MHz)
- ATI 3D Rage Pro (157.5 MHz)
- Toshiba 440CDX (49.5 MHz)
Test Pattern:

Toshiba 440CDX, IIYAMA Vision Master Pro450 monitor, 800x600 resolution @ 75MHz, 6x13 font (in Xterm?), Antenna placed about 10 feet from the monitor's side.
Side Channel Attacks

Monitor Eavesdropping, EM:

2.92 MHz center frequency, 20 MHz bandwidth, 256 (16) frames averaged, 3 m distance

2.92 MHz center frequency, 10 MHz bandwidth, 256 (16) frames averaged, 3 m distance
Side Channel Attacks

Monitor Eavesdropping, EM:

700 MHz center frequency, 100 MHz bandwidth, 256 (16) frames averaged, 3 m distance

It is well known that electronic equipment produces electromagnetic fields which may cause interference to radio and television reception. The phenomena underlying this have been thoroughly studied over the past few decades. These studies have resulted in internationally agreed methods for measuring the interference produced by equipment. These are needed because the maximum interference levels which equipment may generate have been established by law in most countries. (From: Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?)

740 MHz center frequency, 200 MHz bandwidth, 256 (16) frames averaged, 3 m distance
Side Channel Attacks

Monitor Eavesdropping, EM:

480 MHz center frequency, 50 MHz bandwidth, 256 (16) frames averaged, 3 m distance

480 MHz center frequency, 50 MHz bandwidth, magnified image section
Side Channel Attacks

Sending Hidden Messages via Dithered Patterns:

Humans can't tell small differences in color shades - particularly between medium gray and checkered patterns of black and white pixels.

But compromising emissions tend to be strong in higher frequency components – and that is what an eavesdropper will generally see more strongly.

Hence, embedding a dithered image inside a “smooth” one will show a different image to an eavesdropper.
Side Channel Attacks

Sending Hidden Messages via Dithered Patterns:

How to throw off an eavesdropper
Side Channel Attacks

Sending Hidden Messages via Dithered Patterns:
Side Channel Attacks

Sending Hidden Messages via Dithered Patterns:

445 MHz center frequency, 10 MHz bandwidth, 1024 frames averaged, 3 m distance

Hidden analog transmission of text and images via the compromising emulations of a video display system can be achieved by amplitude modulation of a dither pattern in the displayed cover image.

What the eavesdropper might recover
Side Channel Attacks

Sending Hidden Messages via Dithered Patterns:

Hidden analog transmission of text and images via the compromising emanations of a video display system can be achieved by amplitude modulation of a dither pattern in the displayed cover image.

What the hidden message was