Advanced Engineering Course on

Cryptographic Engineering

EPFL Premises, Lausanne, Switzerland
June 25-29, 2012

Thursday, June 28, 2012
Pankaj Rohatgi, Cryptography Research, USA

- Electromagnetic Attacks and Countermeasures
- Improved Techniques for Side-Channel Analysis
These lecture notes are solely for the use of the registered course Participants and Instructors teaching in the course.

No part of these notes may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic, photocopying, microfilming, recording or otherwise) without written permission from MEAD Education SA and/or MEAD Microelectronics, Inc.
Electromagnetic Attacks and Countermeasures

Instructor: Dr. Pankaj Rohatgi
Cryptography Research Inc

Course Outline

- Introduction and History
- EM Emanations Background
- Real-life EM Leakage Examples
- EM Capturing Equipment
- Multiplicity of EM-channels and comparison with power channel
- Using EM to bypass power-analysis countermeasures
- Quantifying EM-Exposure
- Countermeasures
EM History: Mostly Classified

- TEMPEST: Transient electromagnetic pulse emanation standard.
- Parts of TEMPEST literature declassified Jan '01 under FOIA.
  - Electromagnetic, electrical, acoustic ....
- Relevant TEMPEST literature:
  - NACSIM 5000 tempest fundamentals.
  - NACSEM 5112 NONSTOP evaluation techniques.
  - NSTISSI no. 7000 TEMPEST countermeasures for facilities.

Published EM Work (peripherals)

- EM Leakages from peripherals known for some time.
  - E.g., Monitors: Van Eck, Anderson & Kuhn.
EM attacks on peripherals: monitor grabbing and tempest fonts

- Standard Font
- Low Pass Filtered "Tempest Font"

EM analysis (CPU): First published results

- J.-J. Quisquater & David Samyde E-smart 2001
- Gemplus Team CHES 2001
  - SEMA/DEMA attacks
- Required careful positioned E and M-field probes on chip surface to isolate signals.
  - Best results required "decapsulating" the chip

Near-field probe kit (NFPS1, Schaffner)
In parallel

- IBM Research investigation, closer to now declassified TEMPEST literature.
  - Systematic study of EM leakages from computing equipment and peripherals:
    - Chip cards.
    - CPUs and Crypto Accelerators.
    - Monitors, keyboards and peripherals
  - Compare EM to other side-channels.
  - Develop methodology for leakage assessment.
  - Develop countermeasures
- Basis for this course.

IBM’s EM Publications

- Deeper understanding of EM leakages.
  - Similar to now declassified TEMPEST literature.
- Plenty of EM signals are available, provided you know what to look for and where.
  - Superior signals and attacks possible without micro-antennas or decapsulation.
  - Multiple signals available from even a single sensor.
  - Some attacks possible from a distance.
- EM side-channel(s) >> Power side-channel
  - EM can break DPA-resistant implementations.
Multiplicity of EM signals

• “The forms in which compromising emanations might appear at an interception point are numerous”

From NACSIM 5000 TEMPEST Fundamentals.

EM Emanations Background

• Origin/Types of EM Emanations
  – Direct Emanations from intended currents.
    • Maxwell’s eqns, Ampere’s and Faraday’s laws.
      – Eg. [Quisquater,Samyde 01], Gemplus [GMO ’01]
  – Unintentional Emanations from coupling effects.
    • Depend on physical factors (e.g. circuit geometry)
    • Typically ignored by circuit designers.
    • Manifest as modulation of carriers (e.g. clock harmonics) present/generated/introduced in device.
      – AM or Angle Modulation (FM, Phase)
    • Compromising signals available via demodulation.
Origin of EM Emanations

“The strongest and most numerous electromagnetic emanations are generated by sharp-rising and current waveforms of short duration.....Also, faster rise times generate additional emanations -- harmonics -- of progressively lower amplitudes from the same pulse source, these harmonics ...... represent, in effect, a great many compromising signals. These signals can be acquired not only by being correctly tuned to the fundamental frequency, but also at any of the harmonic frequencies.... At times, in fact, harmonics are more useful than the fundamental, i.e., Emanations at the fundamental frequency are often lost among other signals of the same frequency, whereas a harmonic might be more easily isolated.”

From NACSIM 5000 tempest fundamentals.

Modulated Carriers

"Modulated spurious carriers (U). - This type of CE is generated as the modulation of a carrier by RED data. ...... The carrier is usually amplitude or angle-modulated by the basic red data signal. Or a signal related to the basic RED data signal, which is then radiated into space or coupled into EUT external conductors."

From NACSIM 5000 Tempest Fundamentals.
EM measurement: Smartcards

Amplitude Modulation: EXAMPLE 1

- 6805-based smart-card using external 3.68MHz clock.
- 3 instruction, 13 cycle loop:
  - Access RAM containing a value B (5 cycles)
  - Check for external condition (5 cycles)
  - Jump back to start of loop (3 cycles)
Raw signal from near-field sensor during 2 iterations of loop (26 cycles)

FFT OF RAW SIGNAL FROM EXPERIMENT 1 (0-250MHZ)
Log\textsubscript{10} FFT OF RAW SIGNAL (0-20MHz)

FFT OF AM-DEMODULATED (150MHz Carrier, 50MHz BW) SIGNAL
Information Leakage in EM Signals

- Information about computation.
- Large time-scale
  - Major computational blocks.
- Medium time-scale
  - Structure of computational blocks.
- Small time-scale
  - Clock cycle level information.
SEMA

- SEMA: Simple ElectroMagnetic Attack
  - Based on analyzing structure within a single sample.
  - EM equivalent of Simple Power Analysis (SPA)
- Structure and locations of computational blocks visible in sample
  - Examples:
    - Key dependent execution sequence.
      - Different computations sequence for different key bits.
      - E.g., Conditional jumps based on key bits
    - Some instructions directly leak information about operands.
SEMA using AM demodulation

- Example 2
  - Demo: RSA on high end smartphone
- Example 3
  - Demo: ECC on a modern PDA

Demo: RSA on Smart-Phone

- Mobile app with naïve RSA implementation on high-end, 4G LTE smart phone
- Magnetic field pickup coil placed behind phone
- Measurements collected during computation of $M^d \mod N$

Iterations of the exponentiation ‘For’ loop
Simple EM attack modern PDA

- ECC on modern PDA
  - App performing ECC using an open source crypto library
  - Point multiplication \((m \times Q)\) over P-571
  - Signal available from 10 feet away
- Naïve double-and-add algorithm to compute \(m\times Q\)

For each bit \(i\) of secret \(m\)
perform “Double”
if (bit \(i\) == 1)
  perform “Add”
endif
endfor

- In ECC, double and add are very different operations
- The double/add execution sequence yields the bits of \(m\)!

Demo: Extracting the double/add sequence & secret \(m\)

![Graph showing double and add sequence]
AM: Example 4

- PCI-bus based RSA Accelerator S inside a Intel/Linux server.
- Multiple AM modulated carriers.
  - Several carriers at clock harmonics propagate up to 50 feet and through walls.
  - Precise RSA timing available at 50 feet.

EMF Measurement: RSA Accelerator S
Precise Timing (SSL Music)

• S looping with ~3s each of
  – 512-bit RSA
  – 1024-bit RSA
  – 2048-bit RSA
  – 4096-bit RSA

• Music can be heard 50 feet away by AM-demodulating 299MHz clock
  harmonic carrier.

• Enables better timing attacks.
RSA Internals in S

• AM-demodulating an intermodulated carrier at 461.4 Mhz, bandwidth 150Khz.
• Clear signal available upto 3-4 feet.
  – Further distance requires statistical techniques.

S: Two identical 2048-bit exponentiations with 12-bit exponent.
S: Initialization (Fixed modulus, 2 exponents X 2 data)

S: Data and Key Dependent Exponentiation (2 Exponents, 2 Data)
Attacks on S.

- Large class of attack techniques applicable.
  - Such techniques earlier restricted to power analysis attacks on RSA in smart-cards.
  - Earlier foil showed MESD is possible [MDS '99]
  - More powerful attacks will target the Montgomery Multiplication.
- Attacks on Montgomery multiplication
  - Widely used in current RSA hardware.
    - Uses mostly multiplies and avoids the divide operation in implementing modular reduction.
  - Standard implementation has a conditional operation.
    - Gives timing attack [Dhem et al '98, Schindler ‘00]
    - Single exponentiation with known data D and conditional branching information sufficient to obtain key. [WT’01]
      - Even when squares/multiplies are indistinguishable.
      - Power and now EM attack.
    - Conditional branch statistics for squares and multiplies are different for random input data! [WT ‘01].
      - Power and now EM attacks even when using data blinding.
Angle Modulation: Example 1

• Same 6805-based smart-card executing same loop as before, running on variable internal clock (a DPA countermeasure).

• 3 instructions.
  – Check RAM containing value B (5 cycles)
  – Check for external condition (5 cycles)
  – Jump back (3 cycles)

• Varied B and looked at loop frequency.
Angle Modulation: Example 2

- PCI based RSA/Crypto Accelerator R inside an Intel/Linux server.
- AM-demodulate a 99Mhz carrier (clock harmonic).
R performing an RSA operation in a loop

Internals of RSA Exponentiation in R with small exponent
Can we get RSA internals?

- Not directly….
- But, timing of asynchronously generated G affected by ongoing computation due to coupling effects.
  - Timing statistics of G (using ~1000 samples) gives information about internals!!
  - G strong enough to be captured at 10-15 feet.
EM Propagation

- Propagation of EM
  - Radiation
  - Conduction
  - Complex combination of both.
Propagation of EM Emanations

"Modulated spurious carriers (U). - This type of CE is generated as the modulation of a carrier by RED data. .... The carrier is usually amplitude or angle-modulated by the basic red data signal. Or a signal related to the basic RED data signal, which is then radiated into space or coupled into EUT external conductors."

From NACSIM 5000 tempest fundamentals.

More on propagation

- “There are four basic means by which compromising emanations may be propagated. They are: electromagnetic radiation; conduction; modulation of an intended signal; and acoustics. A brief explanation of each follows.
  - a. (C) Electromagnetic Radiation (U). - Whenever a RED signal is generated or processed in an equipment, an electric, magnetic or electromagnetic field is generated. If this electromagnetic field is permitted to exist outside of an equipment, a twofold problem is created; first the electromagnetic field may be detected outside the Controlled Space (CS); second the electromagnetic field may couple onto BLACK lines connected to or located near the equipments, which exit the CS of the installation.
  - b. (C) Line Conduction. - Line Conduction is defined as the emanations produced on any external or interface line of an equipment, which, in any way, alters the signal on the external or interface lines. The external lines include signal lines, control and indicator lines, and a.c. and d.c. powerlines.
  - c. (C) Fortuitous Conduction. - Emanations in the form of signals propagated along any unintended conductor such as pipes, beams, wires, cables, conduits, ducts, etc.
  - d. (C) [Six lines redacted.]” From NACSIM 5000 TEMPEST Fundamentals
EXPERIMENT 3: EM PROPOGATION

- Power-line currents well studied (Power Analysis)
- BUT power line being a conductor must carry conductive EM emanations
  - Faint, AM-Modulated EM signals at low carrier frequencies overwhelmed by large power consuming currents.
  - Faint, AM-modulated EM signals at larger carrier frequencies can be easily extracted!

Raw power line signal during 3 rounds of DES
EM Capturing and Analysis Equipment

- **Capturing Equipment**
  - Antennas (Far Field), Near Field Probes.
  - Current probes, LISNs.
  - Analog Processing: Filters/Amplifiers,
    - Tunable wideband receiver or equivalent ($$).
      - E.g., Dynamic Sciences R-1550
  - Software for invoking device operations and triggering data collection.
  - Digital oscilloscope/sampling board for signal capture.

- **Gross Analysis of Sensor Signal**
  - Spectrum Analyzer

- **Signal Processing and Analysis Software.**
Key Equipment

• Wide bandwidth, Wideband Receiver.
BUILD YOUR OWN: CHEAP, LOW NOISE, BUT INCONVENIENT

COMPROMISE: ICOM 7000: CHEAP, HIGHER NOISE, INTERMEDIATE CONVENIENCE, LIMITED BANDWIDTH
**Capturing EM using low-end receiver**

- Devices
- Antennas
- **GNU Radio** (demodulation, filtering)
- DPAWS™ side-channel analysis software
- Receiver ($350)
- Digitizer, GNU Radio peripheral ($1000)

---

**Key Insights on EM Leakage**

- Several EM carriers:
  - Generated
  - Ambient
  - Introduced (a la NONSTOP)
- Higher frequency, low energy harmonics may carry more information than lower frequency, high energy carriers.
- Different EM carriers carry different information.
- Leakages via some EM side-channels incomparable to Power side-channel leakage
  - Some EM channels may have leakage very similar to power side-channel leakage.
EM vs. Power

• Sometimes, EM is the only side-channel available.
  – Filtered power supplies, restricted access…
  – E.g. Crypto Tokens, SSL Accelerators,…
• EM useful even when power is available
  – Extra information is available in some EM channels
    • For Direct emanations: micro-sensor positioning may better isolate a current flow of interest [QS01, GMO 01]
    • For unintentional emanations: Exploiting Bad Instructions
  – Multiple channels better than one if channel noise is uncorrelated.
    • Covered in next lecture.

DEMA and information leakage

• DEMA: Differential Electro Magnetic Attack
  – Statistical attack using multiple samples.
  – EM analogue of DPA.
• Principle:
  – Just like power signals, EM emanations are correlated to each active bit in the state of device at an instant in time.
    • DEMA works just like DPA.
  – DEMA attack useful to illustrate how a particular bit in an implementation leaks in a EM channel.
DEMA Attack (EM channel 1) for a Sbox output bit for DES on 6805

DEMA Attack (EM channel 2) for same Sbox output bit.
DEMA Attack (EM channel 3), for same Sbox output bit.

DPA Attack the same Sbox output bit
Comparing various EM & Power Leakages

- A simple comparison possible by aligning the three different DEMA correlation plots and the DPA plot in time.
Bad Instructions

- Instructions where some EM leakage >> Power leakage.
- All architectures have BAD Instructions.
  - Typically CPU intensive rather than bus intensive.
- Caution: Bad Instructions can break power analysis resistant implementations.
- Bad Instruction Example: Bit-test on several 6805 based systems leaks tested bit.
TESTED BIT = 0 IN BOTH TRACES

TESTED BIT DIFFERENT
Bad instructions can break power analysis countermeasures

- Assumption behind power analysis countermeasures:
  - Minimize information leakage (from power) from each execution sequence.
    - Additional techniques [KJJ, C et al, GP] can amplify uncertainty.
- Bad instructions in DPA-resistant implementations violate the assumption and create vulnerabilities.
  - Large EM leakage $\rightarrow$ SEMA.
  - Moderate EM leakage $\rightarrow$ Higher-order EM attacks on share-based DPA countermeasures [C et al, GP].
    - If shares are manipulated using BAD instructions, SEMA or Higher order DEMA becomes possible.
      - Some attacks work even when code unknown! [AARR '02]

Vulnerability Assessment: A Prerequisite for validating EM Countermeasures

- Huge Number of Signals
  - Multiple sensors X multiple demodulations
- Many classes of attacks.
  - Low cost attacks:
    - Only one signal captured at a time. Use best available signal for attack.
  - Expensive attacks:
    - Equipment available to acquire several signals at a time.
    - Optimal signal selection and combination strategy within these constraints.
  - Unbounded attacks:
    - Sensors specified.
    - No limit on the number of signals that can be collected.
    - No limit on the signal processing ability.
- Need a sound methodology to assess vulnerability in these cases
  - More details in next lecture.
**Countermeasures**

- Ideally EM analysis countermeasures should include circuit/board redesign to reduce unintentional emanations.

---

**Reducing Leakage**

"The prevention of TEMPEST problems can best be accomplished by being attentive to the problem throughout every stage of the equipment or system design and development. Due to the many ways that information is processed in an equipment, there are many ways that compromising emanations can be generated. It is nearly impossible to completely prevent the generation of such compromising emanations. Therefore, the TEMPEST design objectives should be to (a) keep the amplitude and frequency spectrum of compromising emanations as low as possible (i.e., below the applicable limit); (b) prevent RED signals from coupling from RED to BLACK lines or circuits; and (c) to prevent emanations from escaping from the equipment through electromagnetic or acoustical radiation or through line conduction. When involved in retrofitting non-TEMPEST designed equipments, many of the methods identified herein, in addition to encapsulation techniques, may be useful in meeting design objectives."

From NACSIM 5000 TEMPEST Fundamentals
Why that’s unlikely to occur in practice

“In typical baseband communication or data processing circuit designs, minimum attention is given to suppression of unintentional emanations. Design engineers do not realize the importance of component selection, interconnections, or layout in minimizing signal emanations. Draftspersons, who are unfamiliar with electrical engineering fundamentals, are frequently employed in the design of PC boards and interconnecting leads. Occasionally, this chore is delegated to a computer, which follows a minimal number of rules governing circuit applications and circuit interconnections. As a result, undesired signal emanations will probably be detected when the equipment must be proven TEMPEST hazard-free.”

From NACSIM 5000 TEMPEST Fundamentals

Countermeasures (cont)

• Other means of reducing S/N of EM leakage
  • EM Shielding
  • Noise introduction
  • Physically secure zones.

• Many DPA countermeasures are also EM countermeasures
  – Signal reduction by removing sensitive conditional operations, timing variations, use of instructions that leak excessively (in EM)
  – Once leakage from Power/EM leakage has been made low other countermeasures can be used effectively
    • Incorporating Randomness (e.g., blinding/masking)
    • Protocol level countermeasures
      – Limit usage of any particular key
Improved Techniques for Side-channel Analysis

Instructor: Dr. Pankaj Rohatgi
Cryptography Research Inc

Side-channel Analysis

• Many Side channels:
  – Timing [K], Power [KJJ], EM [QS,GMO], multiple EM channels [AARR]
• Many targeted implementations:
  – Software: DES, AES, RSA, Diffie-Hellman, Elliptic curves, COMP128, SHA,…
  – Hardware: Smart Cards, SIMs, PDAs, SoCs/Asics, FPGAs, SSL accelerators,…
• Attack Techniques:
  – SPA/SEMA, DPA/DEMA, Higher-order DPA/DEMA, IPA, Partitioning Attacks,
    Correlation Power Analysis, Mutual Information, MESD, Collision Attacks, Hidden
    Markov Model,…
• Most attack techniques use only a small fraction of information present in the side-channel(s)!
  – Noise ignored (actually removed by averaging).
  – Multiple related leakages (at different times) are not combined.
• Multiple side-channels are usually not combined.
Improved attacks via better information extraction

- CMOS Device Information Leakage.
- Maximum Likelihood.
  - Gaussian Noise Assumption.
- Applications:
  - Template Attacks.
    - Classical Template Attack
    - Single-bit Template Attacks
    - Template-Enhanced DPA Attacks
  - Improved DPA/DEMA.
  - Multi-channel attacks.
  - Information Leakage Assessment.

CMOS Devices: Side-Channel Perspective

- Simplified view: Many types of current flows
  - Intentional current flows when device changes logic state.
  - Leakage currents.
  - Coupling effects between currents flowing in device.
- State transitions and intentional current flows at clock edge determined by few bits (relevant bits) in logic state (relevant state)
  - Bits of the logic state that change.
  - Bits that influence the actual bits that change.
- Intentional current flows correlated to relevant state.
  - couplings effects also (differently) correlated to relevant state.
  - Leakage currents uncorrelated.
- Power/EM signal depends on current flows.
- An implementation variable (bit) leaks in a side-channel when:
  - It is part of the relevant state.
  - Correlated current flow picked up by the side channel.
Leakage of single bit in an execution sequence in different side-channels

DEMA/DPA results on different side-channels
DEMA Attack (EM channel 1) for a Sbox output bit for DES on 6805

DEMA Attack (EM channel 2) for same Sbox output bit.
DEM Attack (EM channel 3), for same Sbox output bit.

DPA Attack the same Sbox output bit
The Challenge Of Side Channel Cryptanalysis

- How can an adversary best determine (all or part of) the relevant state given side-channel signal(s)?
  - E.g. bits of the data bus during a LOAD instruction
  - Can be formulated as a hypothesis testing problem.
Adversarial Model: Profiling Phase

- A training device (identical to the target device)
- An elementary operation (e.g., cycle/instruction) that can be invoked.
  - Generalizes to larger operation sequences.
- k Probability distributions: \( B_1, \ldots, B_k \) on the relevant states from which the operation can be invoked
- Adversary: repeatedly performs the following:
  - chooses one of the k distributions \( B_1, \ldots, B_k \): say \( B_i \)
  - invokes elementary operation on training device drawn from chosen distribution \( B_i \) on relevant state.
  - Collect resulting side-channel signal(s).
- AIM: Use training device and signals to prepare for an attack.

Adversarial Model: Hypothesis Testing Phase

- Target device
- Adversary:
  - Can make a bounded number of invocations to the same elementary operation on the target device
  - Relevant state that is drawn independently according to exactly one of k distributions \( B_1, \ldots, B_k \) (say \( B_i \)), unknown to the adversary
- Goal: Adversary to select the correct distribution: \( B_i \)
- Measure of success: Success probability expressed as a function of the number of invocations.
A Sophisticated Strategy For The Adversary

- Collect $L$ samples: $O_1, \ldots, O_L$ from target.

- Assume that there are $K$ equally likely hypotheses, $H_1, \ldots, H_k$ corresponding to each of the distributions.

- Let $p(O|H)$ be the probability distribution of the sensor signals under hypothesis $H$.

Maximum Likelihood Test

- The Maximum Likelihood Hypothesis Test:

$$k = \arg\max_{1 \leq k \leq K} \prod_{i = 1}^{L} p(O_i|H_k)$$

- It is optimal!

- Problem:
  - How to compute the probability distribution of the sensor signals for each hypothesis: $P(O | H_i)$?
Principle

- Sample O: K different hypothesis.
- Sample = Fixed Mean Signal + Sample Noise drawn from a Noise probability distribution:

  \[ O = \hat{S}_1 + N_1(.) \]
  \[ O = \hat{S}_2 + N_2(.) \]
  \[ O = \hat{S}_K + N_k(.) \]

  \( \hat{S}_i \) = Signal (hyp i)
  \( N_i(.) \) = Noise sample (hyp i)

- Physical properties of device make \( \hat{S}_i \) distinct from \( \hat{S}_j \)
- To estimate \( \hat{S}_i \):
  - Set distribution to i in training device
  - Collect many samples.
  - Average out noise \( N_i(.) \).

  \[ \text{NOTE: Most side-channel attacks (DPA) focus on mean signal by averaging out noise !} \]
  - Requires lots of samples.
  - Wastes information.

- How to estimate Noise probability distribution \( N_i(.) \)?

Estimating Noise Probability Distribution \( N_i \) for hypothesis i

- Assume O sampled at n-time instants.
- Mean Signal \( \hat{S}_i \): a single vector over \( R^n \)
  - Estimated by averaging many O’s.
- Noise \( N_i \): Prob. distribution function over \( R^n \)
  - Need to assign a probability to each n-dimensional hypercube (region) in \( R^n \)
  - Infeasible to compute without further assumptions on noise!
    - Infeasible to estimate a general p.d.f over even \( 2^n \) (large n)
    - or just on \( R \)
Signal Detection Theory To The Rescue

- Many Noise Models have been developed.
- In many cases noise is Gaussian:
  - E.g: At time instant t, the noise probability distribution is Gaussian:

\[ N_t(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{\frac{-x^2}{2 \sigma^2}} \]

In Practice: Good First Approximation
What about multiple time-instants?

- Noise at each time instant is Gaussian and also *independent* of noise at other time instant?
  - Usually a false assumption.
  - Could be used but don’t expect good results.
- Noise at each time instants is Gaussian but *dependent* (correlated) to noise at other time instants.
  - Fairly accurate assumption.
  - Correlation particularly strong for noise at adjacent time instants.

Multivariate Gaussian Noise Assumption

- Noise is a multivariate Gaussian distribution with mean 0 and a noise covariance matrix \( \Sigma_H \). For any noise vector \( \vec{N} \):

\[
p(\vec{N}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_H|}} \exp\left(-\frac{1}{2}(\vec{N})^T \Sigma_H^{-1}(\vec{N})\right)
\]
Maximum Likelihood and Gaussian Assumption

• Under the Multivariate Gaussian assumption, if \( \mu_H \) is the mean signal for hypothesis H then:

\[
p(O | H) = \frac{1}{\sqrt{(2\pi)^n | \Sigma_H |}} \exp\left(-\frac{1}{2} (O - \mu_H)^T \Sigma^{-1}_H (O - \mu_H)\right)
\]

• Computing \( p(O | H) \) now feasible: Both \( \mu_H \) and \( \Sigma_H \) can be estimated from training device.

Special Case: Binary Hypothesis Testing

• Maximum likelihood test reduces to:

\[
\begin{align*}
(O - \mu_{H_0})^T \Sigma^{-1}_{H_0} (O - \mu_{H_0}) - (O - \mu_{H_1})^T \Sigma^{-1}_{H_1} (O - \mu_{H_1}) \\
\geq \\
\ln(|\Sigma_{H_1}|) - \ln(|\Sigma_{H_0}|)
\end{align*}
\]

• In many cases: \( \Sigma_{H0} = \Sigma_{H1} = \Sigma_N \)
• Probability of Error: \( P_e = \frac{1}{2} \text{erfc}(\frac{\Delta}{2\sqrt{2}}) \)

\[
\Delta^2 = (\mu_{H_1} - \mu_{H_0})^T \Sigma^{-1}_N (\mu_{H_1} - \mu_{H_0})
\]
Application 1

TEMPLATE ATTACKS

Template Attacks Motivation

- Sometimes only a single (or a few) side-channel sample is available.
  - Stream ciphers, Ephemeral keys.
  - “System Level Countermeasures” to side channel attacks:
    - Higher level protocols limit key usage.
- Are these inherently immune to side-channel attacks?
  - Immune to traditional simple/differential attacks
    - Easy to secure implementations against SPA/SEMA
      - Ensure signal differences $<$ noise level.
    - DPA/DEMA and higher order DPA/DEMA inapplicable.
      - Cannot remove noise by averaging over multiple samples.
  - Not immune against Template Attacks (with some assumptions).
Template attacks: overview

- Suppose we have a crypto implementation on many “identical” devices and
  - We can freely experiment with a test device.
  - We get a single (few) side-channel sample(s) from a target device to be attacked.

- Can we extract secrets or keys of the target device?
  - May be impossible using classical side-channel techniques like SPA/SEMA etc.

- But:
  - Can build precise models of noise AND expected signal using the test device.
  - Use methods like maximum likelihood to extract ALL information from the target sample using these models
  - This is the template attack.

Classical Template Attack
Template Attacks: Plan

- **Test case: RC4 on smart card**
  - Ideal candidate to highlight template attacks.
    - At most one sample can be captured during state initialization with key.
    - State changes on each invocation

- **Noise Modeling**
- **Classification Technique**
  - Variants
  - Empirical Results
- **Related Work and Implications**

---

Example: RC4 on a smart-card

- At best, single trace during RC4 state initialization with ephemeral key available.

```c
STATE INITIALIZATION
i = j = 0;
for (ctr=0; ctr < 256; ctr++)
{
    j = key[i] + state[ctr] + j;
    SwapByte(state[ctr], state[j]);
    i = i + 1;
}
```

- Can avoid SPA by having key independent code and low leakage instructions.
- No DPA style attack possible.
- One key byte used per iteration.
- Is a single sample enough to recover the whole key?
  - Can two fixed keys different in 1st byte be distinguished during 1st iteration?
An Actual Implementation of RC4 on a 6805 Smart Card

Power Sample showing 6 iterations of loop
No SPA: Averaging 5 samples

Distinguishing different first key bytes by signal averaging needs ~ 50 samples
RC4 Template Attack: Base Step

- Collect **single sample** of key initialization from device under test.
- With experimental device, collect large number (100s) of samples for **all values of first key byte**.
- Identify points on samples directly affected by first key byte.
  - These are points where the **mean signals for different first byte differ significantly**. Assume L such points
  - *For these L points only, for each different key byte compute the mean signal and a characterization of the noise probability distribution.*
  - These are the templates for the first byte of RC4.
- Use maximum-likelihood test to classify target sample (same L points) with respect to the templates for the first byte.
  - Classification result is our best bet for the first byte of key.

**Essential Detail: Significant Points.**

- Select points where signal depends on the key.
- Build template **only** at selected points.
- Point selection impacts results

Variation in Mean signals for different key bytes
**ATTEMPT 1: Univariate Gaussian Noise Model**

- Assume that the noise at the L points is Gaussian but independent
  - $\Sigma_H$ is diagonal; no need to compute covariance between different sample points.

- **Noise characterization:** Given L-point samples for a particular value of key compute
  - Averages: L point average $\hat{S}[i], i=1,..,L$
  - Variances: L point variances $V[i], i=1,..,L$.
    - Diagonal noise variance matrix $\Sigma_H$ (L x L)
      $$\Sigma_H [i,i] = V[i]$$

**Empirical Results with Univariate Gaussian Assumption for 5 key hypothesis**

<table>
<thead>
<tr>
<th>Actual Key Byte</th>
<th>1111110</th>
<th>11101110</th>
<th>11011110</th>
<th>10111110</th>
<th>00010000</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111110</td>
<td>0.86</td>
<td>0.04</td>
<td>0.07</td>
<td>0.03</td>
<td>0</td>
</tr>
<tr>
<td>11101110</td>
<td>0.06</td>
<td>0.65</td>
<td>0.10</td>
<td>0.19</td>
<td>0</td>
</tr>
<tr>
<td>11011110</td>
<td>0.08</td>
<td>0.16</td>
<td>0.68</td>
<td>0.09</td>
<td>0</td>
</tr>
<tr>
<td>10111110</td>
<td>0.10</td>
<td>0.11</td>
<td>0.08</td>
<td>0.71</td>
<td>0</td>
</tr>
<tr>
<td>00010000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Cross Classification probability.**

- **significant error** if key bytes are similar (low hamming distance).
ATTEMPT 2: Multivariate Gaussian Noise Model

- Multivariate statistics:
  - Use full noise correlation matrix $\Sigma_H$
- Noise characterization: Given L-point samples for a particular value of key compute
  - Averages L point average $\hat{\Sigma}$
  - Noise correlation matrix $\Sigma_H$ \((L \times L)\)
    - $\Sigma_H[i,j] = \text{covariance}(T[i]-\hat{\Sigma}[i], T[j]-\hat{\Sigma}[j])$ for samples $T$

Results with Multivariate Model

EMPirical RESULT:
Among these 10 key bytes, correct classification probability:

<table>
<thead>
<tr>
<th>Key byte</th>
<th>0xFE</th>
<th>0xEE</th>
<th>0xDE</th>
<th>0xBE</th>
<th>0xFD</th>
<th>0xFB</th>
<th>0xF7</th>
<th>0xED</th>
<th>0xEB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>98.62</td>
<td>98.34</td>
<td>99.16</td>
<td>98.14</td>
<td>99.58</td>
<td>99.64</td>
<td>100</td>
<td>99.76</td>
<td>99.94</td>
</tr>
</tbody>
</table>

- These 10 keys deliberately chosen to be very close.
- Estimated that only a 5-6% error probability in identifying the correct 1st byte out of 256 possibilities.
Multi-byte keys: simple attack

Target smartcard performing RC4 state initialization with unknown key

Simple Attack: Multi-byte keys

Test card

Target

Byte 1 Byte 2 Byte 3

0

1

MAXIMUM LIKELIHOOD TEST

255
Template Attack: Multi-byte Keys

Byte 1  Byte 2  Byte 3

0

Byte 1 = 7

1

... = 100

255

Template Attack: Multi-byte Keys

Byte 1  Byte 2  Byte 3

0

Byte 1 = 7

1

... = 100

255

Mean signal

Noise
Template Attack: Multi-byte keys

Attacking multi-byte keys

BUT.

- 5-6% error in identifying single byte is too much.
  - Any single error in classification will render attack useless.
  - Over multiple bytes, probability of a single error can be significant.
Modifying Maximum Likelihood

- *Maximum Likelihood*: Retains hypothesis predicting max probability for observed noise ($P_{\text{max}}$)
- *Approximation*: Retain ALL hypotheses predicting probability at least ($P_{\text{max}}/c$), $c$ constant.
  - Retain more than 1 hypothesis for each byte.
  - Tradeoff between number of hypothesis retained and correctness.

### Approximate Approach Tradeoff
(Estimates based on Experiments)

<table>
<thead>
<tr>
<th></th>
<th>c=1</th>
<th>c=$e^6$</th>
<th>c=$e^{12}$</th>
<th>c=$e^{24}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success probability (retaining correct byte hypothesis out of 256)</td>
<td>95.02</td>
<td>98.67</td>
<td>99.37</td>
<td>99.65</td>
</tr>
<tr>
<td>Avg. number of hypothesis retained</td>
<td>1</td>
<td>1.29</td>
<td>2.11</td>
<td>6.89</td>
</tr>
</tbody>
</table>
RC4 Template Attack.
Iteration : Extend and prune

• For each remaining possible value of first byte
  – For each value of second byte
    • Build template independently ONLY for second iteration (less accurate)
    • OR Build template for first 2 iterations together (twice as large)
  – Classify using new template to reduce choices for first 2 bytes

Iteration: Estimated Result

• Base case: Narrow candidates for first byte to a small number (e.g., on average 1.3 possibilities with 98.67% correctness for c=e)
• Assume T_i candidates for first i key bytes.
  – Extend: For each candidate, build larger templates for all 256 possible values of next byte.
  – Prune: Use approximate approach to reduce 256 T_i candidates down to T_{i+1} < 1.3 * T_i (diffusion).
• Tedious but feasible for reasonable sized keys.
  – N bytes: Failure probability 1.33N% (N=16, failure prob.= 21.5%)
  – Number of remaining candidates < 1.3^N (N=16, candidates < 67)
• Substantially better when templates include sample for all iterations up to now
  – Error rates of not retaining correct hypothesis is almost same as single byte case.
  – Number of retained hypothesis is smaller
When do template attacks work best?

- When we have good contamination and good diffusion.
- Contamination:
  - Variable that needs determination leaks in the side channel in different ways at several different times.
  - E.g., in RC4 each key byte affects many cycles in an iteration.
  - Good contaminations result in good classification probability.
- Diffusion:
  - Small differences in values of a variable being determined result in major differences in values used for subsequent processing.
  - E.g., in RC4, small differences in first key byte translate to very large differences in subsequent iterations.
  - Good diffusion rapidly eliminates choices that were retained in earlier iterations because the values were very similar.

Signals (and signal difference) for two fixed keys with different first byte

![Graph showing signals and signal difference](image)
Other cases

- Template attacks verified in other cases
  - EM emanations from hardware SSL accelerators
    - Single sample noisy analogue of earlier work.
  - Hardware based DES
    - Attacking key checksum verification steps
  - AES

Related work

- [Messerges,Dabbish,Sloan][Walter] Use signal based iterative method based to extract exponent of device implementing RSA.
- [Fahn, Pearson] Use profiling of experimental device before attack on device-under-test.
- [Reshberger, Oswald, WISA 2004.] Further practical details and improvements on template attack methodology.
- [Archambeau, Peeters, Standaert Quisquater, CHES 2006] Template Attacks in Principal Subspaces
  - Use top few principal components of profiled signals for template building instead of picking the L significant points.
Single-bit template attack

Classical Template Attack Drawbacks

- Methodology: Iteratively make test device’s computation identical to target device’s.
  - *Tedious*: 256 templates for each unknown byte
  - *Iterative and on-line*: Templates for later bytes requires attack on earlier bytes.
  - Cannot handle randomized implementations
    - Attacker cannot get test device to produce the same randomness as target.
Single-bit template attacks

- Based on an empirical observation
  - After DPA attack on algorithmic bit $b$, binary templates can be built to extract $b$ directly from signals

- These binary templates are good classifiers for bit $b$ for signals with random inputs

Example:

- Typical DPA attack:
  - Round 1, $s_{1b0}$:
    - Bit 0 of output of Sbox 1.
### DES: Experimental results

#### Single-bit prediction probabilities

<table>
<thead>
<tr>
<th>Sbox bit</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>0.91</td>
<td>0.88</td>
<td>0.93</td>
<td>0.77</td>
<td>0.72</td>
<td>0.80</td>
<td>0.84</td>
</tr>
<tr>
<td>1</td>
<td>0.98</td>
<td>0.88</td>
<td>0.92</td>
<td>0.94</td>
<td>1.00</td>
<td>0.92</td>
<td>0.97</td>
<td>0.77</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>0.89</td>
<td>0.99</td>
<td>0.92</td>
<td>0.95</td>
<td>0.83</td>
<td>0.90</td>
<td>0.79</td>
</tr>
<tr>
<td>3</td>
<td>0.90</td>
<td>0.91</td>
<td>0.72</td>
<td>0.85</td>
<td>0.83</td>
<td>0.86</td>
<td>1.00</td>
<td>0.89</td>
</tr>
</tbody>
</table>

**Entropy Loss**

- 2.57
- 2.10
- 2.13
- 2.30
- 2.28
- 1.50
- 2.61
- 1.35

**Total** = 16.8/32

Templates built using 1400 signals, 50 significant points

### Single-bit template attacks: Direct Applications

- Build large number of single bit templates targeting different (key and input data dependent) bits
  - E.g., in DES,
    - 32 Sbox output bits in 1st round and 32 “data” bits in 2nd round depend only on 1st round key and input data.
- Use templates to classify these (key and input data dependent) bits in a single trace.
- Significant entropy loss for key used in trace.
  - Entropy loss speeds up key search attacks.
    - E.g. using **Serial List Viterbi Algorithm**.
Template-Enhanced DPA: overcoming random masking countermeasure

Using single-bit template attack as a building block

How to overcome random masking?

• Single bit templates classify bits from signals with random/unknown data/key.
  – But single-bit template building requires knowing value of the bit within the test device

• Attacker only needs defective RNG in test device for building templates.
  – Getting defective RNG in test device
    • Physical attack on test device RNG.
    • But other means could be easier ...
Insider attack: Patching a test card EEPROM

Signal from Test card with EEPROM patch to disable RNG vs Signal from Target card

Some cards in the field have defective (biased) RNGs!

Results of DPA on a "DPA-protected card" in the field
The problem: 3-5% bias in the RNG!
Template-enhanced DPA: The steps

- Perform complete classical DPA on “DPA-protected” test card (with defective RNG) to extract the key.
  - DPA attacks on all predicted algorithm bits \( b \) will succeed.
- Build single-bit templates for each bit \( b \) using key and inputs.
  - Actual traces use \( b \) biased mask instead of \( b \)
  - Errors if RNG is biased (rather than 0) but these don’t matter.
- Perform modified DPA on target card with perfect RNG
  - Modification: Replace DPA selector for bit \( b \) with DPA selector \( b \) (data, key-hypothesis) \( \oplus \) template classification for \( b \) (signal).
- (Highest) Peaks indicate the correct key hypothesis.

How it works

Example: Attacking Sbox lookup in a masked implementation
S-box lookup: Regular vs. Masked

Regular S-box lookup

\[ K \oplus D \rightarrow Sbox \rightarrow O \]

Masked S-box lookup

\[ K \oplus D \oplus imask \rightarrow Masked Sbox (imask, omask) \rightarrow O' = O \oplus omask \]

DPA: with defective RNG

Regular S-box lookup

\[ K \oplus D \rightarrow Sbox \rightarrow O \]

Masked S-box lookup with defective RNG

\[ K \oplus D \oplus imask \rightarrow Masked Sbox (imask, omask) \rightarrow O' = O \oplus omask \]

DPA predicts O. But for right hypothesis O' correlates to O. Peaks where O' gets used.

Single-bit templates build using peaks and hypothesis above.

Templates will classify bits of O' if (template building errors notwithstanding)
Template Enhanced DPA on Target

Masked S-box lookup with perfect RNG

$$K \oplus D \oplus \text{imask}$$

DPA predicts bits of $O$

$$\text{Masked Sbox (imask, omask)}$$

$$O' = O \oplus \text{omask}$$

For correct hypothesis:

Template enhanced DPA selector for “sbox output bit i”

= DPA selector [output bit i] \oplus Template classification [output bit i]

= $O [bit \ i] \oplus O'[bit \ i]$

= $\text{omask[bit \ i]}$

Peaks appear wherever $\text{omask}$ is processed!

Template-enhanced DPA: experiments

- Two algorithms on two smart-cards.
  - DES on Smartcard A (ST19: 6805 architecture)
  - AES on Smartcard B (Atmel ATMega163: AVR architecture)
- Build implementations with masking countermeasure
  - with RNG that could be switched on/off.
- Templates built using RNG off, attack with RNG on.
  - Building templates with RNG off vs biased RNG not an issue
  - Experimental, simulation and analytical result: this only impacts signal requirements for template creation NOT effectiveness of the attack.
Protected DES on Smartcard A: The cards

Test Card: Serial No: 000285
Switchable RNG

Target Card: Serial No: 000298
Switchable RNG

Results for protected DES on smartcard A

Test Card 285 RNG OFF
DPA sbox1-bit0
Source for templates

Test Card 285 RNG ON
Template-DPA sbox1-bit0-mask

Target Card 298 RNG ON
Template-DPA sbox1-bit0-mask
Results: Protected AES (smart-card B)

DPA: Sbox 0 bit 1: RNG off

Template-enhanced DPA: Sbox 0 bit 1 mask: RNG on

Intuition: Why biased RNG templates are as good as broken RNG templates
Under simplifying assumptions
Signal/Noise for bit=0/1 signals in L-dimensional space and the max likelihood test

Max Likelihood test

0 signals 1 signals

“1” signals used in template creation with biased RNG

Minority of 0 signals Majority of 1 signals
Gaussian “fitted” on the “1” signals.

Mean closer to mean of 0 signals.
Variance larger than variance of 1 signals

Max likelihood test for the “0” and “1” signals

MAX likelihood test for 0 and 1 signals
Converges to test for 0 and 1 signals
Templates build using enough “0”/ “1” samples give same results as templates build using 0/1 samples

Experiments and Monte Carlo simulation with actual noise matrices
Template-based DPA (Further Reading)

  - Template-DPA on masked HW implementation

- E. Oswald and S. Mangard, Template Attacks on Masking–Resistance is Futile, CT-RSA 2007
  - Strategies for applying Template-DPA to masked implementations
    - Templates to extract hamming weights of intermediates

---

Application 2

Improved DPA/DEMA
The situation thus far:

- **Adversary:**
  - has a training device identical to the target device
  - goes through a profiling phase using the training device.
- **But DPA is more popular:**
  - No training devices needed.
  - No profiling required.
  - **BUT,** same techniques can be used to further optimize DPA-style attacks

**DPA**

- Decision metric:

\[ M_H[j] = (\mu_{H,0}[j] - \mu_{H,1}[j])^2 \]
Recall: Binary Hypothesis Testing

- Maximum likelihood test reduces to:

\[
\begin{align*}
(O - \mu_{H_0})^T \Sigma^{-1}_{H_0}(O - \mu_{H_0}) - (O - \mu_{H_1})^T \Sigma^{-1}_{H_1}(O - \mu_{H_1}) \\
\geq \\
\ln(|\Sigma_{H_1}|) - \ln(|\Sigma_{H_0}|)
\end{align*}
\]

- In many cases:

\[
\Sigma_{H_0} = \Sigma_{H_1} = \Sigma_N
\]

- Probability of Error:

\[
P_e = \frac{1}{2} \operatorname{erfc}\left(\frac{\Delta}{2\sqrt{2}}\right) \\
\Delta^2 = (\mu_{H_1} - \mu_{H_0})^T \Sigma_N^{-1}(\mu_{H_1} - \mu_{H_0})
\]

Improving DPA

- Let \( \mu_{H_i}[j] = \mu_{H_i,0}[j] - \mu_{H_i,1}[j] \)

- Assume a null hypothesis \( H_v \) corresponding to a random bifurcation of the samples into two bins.

- Using the Gaussian assumption, the metric of hypothesis \( H_i \) wrt \( H_v \) (log likelihood ratio) is given by:

\[
M_{H_i}[j] = \frac{(\mu_{H_i}[j] - E(\mu_{H_i}[j]))^2}{V(\mu_{H_i}[j])} - \frac{(\mu_{H_i}[j] - E(\mu_{H_i}[j]))^2}{V(\mu_{H_i}[j])} - \ln\left(\frac{V(\mu_{H_i}[j])}{V(\mu_{H_i}[j])}\right)
\]
Improved DPA: Computing $E(\mu_H[j])$ and $V(\mu_H[j])$

- Simplification: use sample estimates:
  
  $E(\mu_H[j]) = \mu_H[j] = \mu_{H,0}[j] - \mu_{H,1}[j]$  
  
  $V(\mu_H[j]) = \frac{\sigma_{H,0}^2[j]}{N_0} + \frac{\sigma_{H,1}^2[j]}{N_1}$

- So the new metric is:
  
  $M_{H,j} = \frac{(\mu_{H,j} - \mu_{H,j})^2}{\frac{\sigma_{H,0}^2[j]}{N_0} + \frac{\sigma_{H,1}^2[j]}{N_1}} - \ln \left( \frac{\sigma_{H,0}^2[j]}{N_0} + \frac{\sigma_{H,1}^2[j]}{N_1} \right)$

<table>
<thead>
<tr>
<th>S-box hypothesis</th>
<th>Min # of samples (Diff of means)</th>
<th>Min # of samples (Max likelihood)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1, B3</td>
<td>640</td>
<td>350</td>
</tr>
<tr>
<td>S2, B3</td>
<td>630</td>
<td>210</td>
</tr>
<tr>
<td>S7, B3</td>
<td>110</td>
<td>40</td>
</tr>
<tr>
<td>S8, B3</td>
<td>130</td>
<td>90</td>
</tr>
</tbody>
</table>
Application 3

Multi-channel Attacks

There Is A Lot Of Information Out There!

- Multiple EM channels + Power Channel
- Different side-channels carry different information!
- How can multiple side-channels be used for better attacks?
Exploiting Multiple Side Channels

• Choice of channels:
  – What sensors to use?
  – Which samples to collect?
  – Where should the sensors be placed?

• Method of combination:
  – How should the information in the samples be combined?

• Advantage:
  – Is there any way to quantify the advantage gained by combining these channels?

Special Case: Binary Hypothesis Testing

• The test reduces to:

\[(O - \mu_{H_0})^T \Sigma_{H_0}^{-1} (O - \mu_{H_0}) - (O - \mu_{H_1})^T \Sigma_{H_1}^{-1} (O - \mu_{H_1}) \geq 0\]

\[\ln(|\Sigma_{H_1}|) - \ln(|\Sigma_{H_0}|)\]

• In many cases: \[\Sigma_{H_0} = \Sigma_{H_1} = \Sigma_N\]

• Probability of Error:

\[P_e = \frac{1}{2} \text{erfc} \left( \frac{\Delta}{2\sqrt{2}} \right)\]

\[\Delta^2 = (\mu_{H_1} - \mu_{H_0})^T \Sigma_N^{-1} (\mu_{H_1} - \mu_{H_0})\]
Multiple Channel Selection

• Question:
  – How to choose the channels?

• Answer:
  – Minimize the probability of error

• More concretely:
  – Choose channels so that $\Delta^2$ is maximized

But picking two channels with best SNR may not work!

Sample from two channels $=[O_1, O_2]^T$

$H_0: O_k = N_k, \quad H_1: O_k = S_k + N_k$

$\mu_N = 0, \quad \Sigma_N = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$

SNR for the two channels $S_1^2$ and $S_2^2$

$$\Delta^2 = \frac{(S_1 + S_2)^2}{2(1 + \rho)} + \frac{(S_1 - S_2)^2}{2(1 - \rho)}$$

Case 1: $S_1 = S_2$ but Noise Correlated $\Delta^2 = \frac{2S_1^2}{(1 + \rho)}$

Case 2: $S_2 < S_1$ but Noise Uncorrelated $\Delta^2 = S_1^2 + S_2^2$

Case 2 is better when $\frac{S_2^2}{S_1^2} > \frac{1 - \rho}{1 + \rho}$
Multi-channel template attacks

• For a given invocation, concatenate samples from multiple channels to get a larger sample.
• Experiment:
  – Invoke operation on smart card with two different bytes
  – Collect power and EM samples for only 3 cycles

Multi-channel template attacks (contd)

<table>
<thead>
<tr>
<th>Correct Hypothesis</th>
<th>Error (Power)</th>
<th>Error (EM)</th>
<th>Error (Power+EM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_0$</td>
<td>9.5%</td>
<td>15.1%</td>
<td>2.8%</td>
</tr>
<tr>
<td>$H_1$</td>
<td>20.1%</td>
<td>15.2%</td>
<td>6.6%</td>
</tr>
</tbody>
</table>

• MORAL: There is insecurity in numbers!
Multi-channel DPA

- Generalization of single channel DPA
- Each observation is a vector of $L$ samples from $L$ side channels
  - Since DPA treats each time instant independently, multiple side channels can be pooled together only if they have similar leakage characteristics
- Caveat: DPA-style attacks cannot use leakages that occur at different times in the channels.

Multi-channel DPA

- Same approach as improved DPA: Start with metric
  \[ \bar{\mu}_H[j] = \mu_{H,0}^0[j] - \mu_{H,1}^0[j], \ldots, \mu_{H,0}^L[j] - \mu_{H,1}^L[j] \]
  \[ \text{def} \]
  \[ = \mu_H^0[j], \ldots, \mu_H^L[j] \]
- Null hypothesis $H_v$: random division of samples into two bins.
- Hypothesis $H_i$: Observed differences in means in the 0/1 bins along each of $L$ dimensions is not by chance.
- Use multivariate ($L$ dimensional) Gaussian assumption to derive the log likelihood ratio of $H_i$ with respect to $H_v$.

The $L$ dimensions are now related by Covariance matrix, that can be computed (using simplification) from covariance matrix of between the samples at time $j$. 
### Experimental Results

<table>
<thead>
<tr>
<th>S-box hyp</th>
<th>Min # of samples (Power)</th>
<th>Min # of samples (EM)</th>
<th>Min # of samples (Power +EM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1,B1</td>
<td>170</td>
<td>640</td>
<td>150</td>
</tr>
<tr>
<td>S1,B2</td>
<td>(1000)</td>
<td>340</td>
<td>60</td>
</tr>
<tr>
<td>S1,B3</td>
<td>350</td>
<td>160</td>
<td>110</td>
</tr>
<tr>
<td>S2,B2</td>
<td>50</td>
<td>230</td>
<td>30</td>
</tr>
<tr>
<td>S2,B3</td>
<td>210</td>
<td>340</td>
<td>120</td>
</tr>
<tr>
<td>S4,B0</td>
<td>200</td>
<td>340</td>
<td>60</td>
</tr>
<tr>
<td>S6,B1</td>
<td>180</td>
<td>190</td>
<td>180</td>
</tr>
<tr>
<td>S7,B3</td>
<td>40</td>
<td>520</td>
<td>30</td>
</tr>
<tr>
<td>S8,B3</td>
<td>90</td>
<td>140</td>
<td>60</td>
</tr>
</tbody>
</table>

### Application 4

Information Leakage Assessment
Scenario

• Attacker has K sensors giving multiple side channel signals.
• Each sensor has finite bandwidth.
• Assume each sensor signal represented with arbitrary precision.
• Let \( O(t) = [O_1(t), O_2(t), \ldots, O_K(t)] \) be the concatenation of K sensor signals.
• Let \( \Omega(t) \) be the space of all possible \( O(t) \).
• Need to determine how much information about the relevant state (in an elementary operation, e.g., clock cycle in an instruction) can be extracted from the information captured by the sensors given \( L \) independent invocations.
  – Simple case binary hypothesis testing. Is relevant state from Distribution D0 or Distribution D1.
• Technique: Maximum Likelihood.

For more information

• See Reference:
References


References

• NSA. NACSIM 5000 (1982). Available at: http://cryptome.org/nacsim-5000.htm