Attacking of Embedded Systems by Using Simple Power Analysis

V.P.S. Nikitha.Peddada#1, Kartheek.B.N#2
#Department of Electrical and Electronics Engineering, K.L University
Guntur, India

Abstract— Embedded Systems are being used for the development and implementation of Safety and Mission Critical Systems. Malfunctions of such type of embedded systems will lead to disasters at times. The embedded systems must be fully secured from outside intervention in order to have effective functioning as well as to provide protective environment to these mission critical systems. There are several attacking systems discussed in the literature each requiring a kind of counter attacking system. Power Analysis and variations of power analysis are the significant attacking mechanisms discussed in the literature. Crypto servers are the main areas of attacking as they deal with securing the data that flow in-between several components of the embedded systems. Most of the attacking systems suggested in the literature suffer from lack of experimental models to emulate the attacking system. An attacking system could be amply proved when several samples of data are used for attacking and the samples of data provide for knowledge base. In this paper an experimental setup is proposed which is an embedded system by itself for creation of a Knowledgebase which shall form the basis for attacking. The experimental setup required for undertaking the actual attacking with the usage of the knowledgebase is also presented. Further, the proposed attacking system is applied for mission critical system and the experimental results obtained through the simulation are also presented.

Keywords— Power Analysis, Attacking, Embedded Systems, Knowledge Base, Crypto Server, Data Encryption standard (DES).

I. INTRODUCTION

In cryptography, a side channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than Brute force or theoretical weaknesses in the algorithms. For example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information which can be exploited to break the system. Some side-channel attacks require technical knowledge of the internal operation of the system on which the cryptography is implemented, although others such as differential power analysis are effective as black-box attacks. The most powerful side channel attacks are based on statistical methods pioneered by Paul Kocher.

Attempts to break a cryptosystem by deceiving or coercing people with legitimate access are not typically called side-channel attacks; see Social engineering and rubber-hose cryptanalysis. For attacks on computer systems themselves (which are often used to perform cryptography and thus contain cryptographic keys), see Computer security. The rise of Web applications and software-as-a-service has also raised the possibility of side-channel attacks on these programs, even when transmissions between a Web browser and server are encrypted, according to Microsoft and Indiana University.

General classes of side channel attack include:

- Timing attack — attacks based on measuring how much time various computations take to perform.
- Power monitoring attack — attacks which make use of varying power consumption by the hardware during computation.
- Electromagnetic attacks — attacks based on leaked electromagnetic radiation which can directly provide plaintexts and other information. Such measurements can be used to infer cryptographic keys using techniques equivalent to those in power analysis, or can be used in non-cryptographic attacks, e.g. TEMPEST attacks.
- Acoustic cryptanalysis — attacks which exploit sound produced during a computation (rather like power analysis).
- Differential fault analysis in which secrets are discovered by introducing faults in a computation.

In all cases, the underlying principle is that physical effects caused by the operation of a cryptosystem can provide useful extra information about secrets in the system, for example, the cryptographic key, partial state information, full or partial plaintexts and so forth. The term cryptphthora (secret degradation) is sometimes used to express the degradation of secret key material resulting from side channel leakage.

II. HISTORY

Side channels are a variant of the classic covert-channel problem. Covert channels involve two or more processes collaborating to communicate via a shared resource that they can both affect and measure. Attackers can exploit these channels to bypass operating system protections such as mandatory access control that are intended to keep the processes separate. For example, one process can allocate memory while the other measures the amount of free memory. Through repetition of this behavior, the first process can slowly communicate information to the second. The channel’s signal-to-noise (S/N) ratio measures its quality. For example, memory allocations by unrelated
processes might skew some measurements, so a particularly busy system might have a low S/N ratio. Error correction methods can assist with this case.

Whereas covert channels involve the problem of preventing cooperation, side channel attacks are a purely adversarial problem. Side channels emerge because computation occurs on a non-ideal system, composed of transistors, system, composed of transistors, wires, power supplies, memory, wires, power supplies, memory, wires, power supplies, memory, and peripherals. Each component has characteristics that vary with the instructions and data being processed. When this variance is measurable by an attacker, a side channel is present.

Intelligence agencies have often relied on side channel attacks to monitor their foes. In one clever incident, the Soviet Union provided a large wooden seal to the American consulate in Moscow.

The US ambassador proudly hung it in his office after it had been examined for covert transmitters. It appeared to be clean. Unbeknownst to the ambassador, the seal contained a carefully designed cavity that vibrated in response to sounds in the room. The spies transmitted a radio beam at the seal, measured the beam’s modulation, and recreated the conversations in the room. This listening method went undetected for years. More recently, side channel attacks have become a powerful threat to cryptography. One of the first papers on side channel attacks showed how to recover an RSA private key merely by timing how long it took to decrypt a message. 1 This was possible because RSA and other public-key cryptosystems work with large numbers (for example, 2,048 bits), whereas modern CPUs have a smaller word size. Crypto implementations compensate by using multiprecision arithmetic, representing large numbers by an array of words and using a loop to carry overflows from one word to the next.

To raise a multiprecision number to an exponent, systems such as RSA commonly use square-and-multiply. This optimization decomposes an exponentiation into a series of squarings (x2) and conditional multiplies (* x), which occur if the bit in question is a one.

This is similar to pencil-and-paper multiplication, in which trailing zeros mean that you shift the result one decimal place to the left while nonzero digits are multiplied and added to the result.

Because the multiply step is conditional, an attacker gains information about the total number of one bits with each decryption. By measuring the total time to perform a multiprecision exponentiation with different input messages, the attacker can eventually recover the entire private key or enough to brute-force the rest.

Timing attacks have continually improved, even being performed against an SSL (Secure Sockets Layer) implementation over a network.2 New ways to filter jitter have improved the distinguishability to 200 ns over a LAN and 30 ms over the Internet.3 Attacks have also exploited new side channels. Power consumption, RF and electromagnetic emissions, sound, vibration, and even heat give away information about secret computations. These attacks aren’t merely the subject of research papers. Smartcards used for payment, Smartcards used for payment, transit, and satellite TV have been compromised by both active fault induction attacks (“glitching”) and side channel attacks. Hackers used a timing attack against a secret key stored in the Xbox360 CPU to forge an authenticator and load their own code.

Embedded-systems designers are no longer the only ones who are no longer the only ones who must prevent side channel attacks. Previously, network-based timing attacks against SSL were the only side channel attack most software developers needed to consider. But today, virtualization and application-hosting services such as Amazon S3 have given attackers a more privileged vantage point of running code on the same system (possibly even at the same privilege level) as the target’s code. Also, high-speed multicore CPUs with large caches and complicated instruction- and data-dependent behavior provide more possibilities for side channels and greater precision for measurements.

To illustrate side channel attacks against software cryptography, I analyze three recent attacks. Each is increasingly more powerful, to the point where the attacker can recover an entire RSA key by measuring the behavior of a single decryption operation.

A. We need hardware security because:

- Theft of service
  - attacks on service providers (satellite TV, electronic meters,
    access cards, software protection dongles)
- Access to information
  - information recovery and extraction
  - gaining trade secrets (IP piracy)
  - ID theft
- Cloning and overbuilding
  - copying for making profit without investment in development
  - low-cost mass production by subcontractors
- Denial of service
  - dishonest competition
  - electronic warfare

B. Industry needs secure chips:

- car industry
  - anti-theft protection, spare parts identification
- accessory control
  - mobile phone batteries, printer toner cartridges, memory modules
- service and access control
  - RFID tags, access cards, payment tokens, software dongles
- home entertainment and consumer electronics
– consumables, accessories, game consoles
• intellectual property protection
– software copy protection
– protection of algorithms
– protection against cloning and reverse engineering

III. PRINCIPLE OF POWER ANALYSIS

• Semiconductors use current while switching
• Shape of power consumption profile reveals activity
• Comparison of profiles reveals processes and data
• Power is consumed when switching from 1 0 or 0 1

![Power consumption during clock cycle](image)

**Fig1:** Power consumption of a system during a clock cycle.

A. Side channel analysis tools

- Probes
- Power: Intercept power circuitry with small resistor
- EM: Coil with low noise amplifier
- Digital storage oscilloscope
- High bandwidth amplifier
- Computer with analysis and control software

B. Simple Power Analysis (SPA) working scheme Attack on Smartcards:

An SPA attack, as described in , involves directly observing a system’s power consumption. Different attacks are possible depending on the capabilities of the attacker. In some situations the attacker may be allowed to run only a single encryption or decryption operation. Other attackers may have unlimited access to the card. The most powerful attackers not only have unlimited access, but also have detailed knowledge of the software and hardware running in the card. If an attacker can determine where certain instructions are being executed, it can be relatively simple to extract useful information. For example, during the PC1 permutation in DES, we could determine the Hamming weight of each key byte by measuring the pulse height at the cycle of the instruction that accesses this data. In an 8-bit microprocessor, knowing the Hamming weight of all eight DES key bytes reduces the brute-force search space from 256 to keys depending on whether or not the parity bits are used. This was just an example; with algorithms using more key bits than DES or with triple-DES, knowing Hamming weight information alone does not help much with this type of brute-force attack.

A more powerful attack can result if the attacker can see Hamming weight information about the key bytes and also information about shifted versions of the key bytes.

In DES, such information can be leaked when shifting the C and D registers. In fact, given the weight of each byte for eight of the C and D shifts there is enough information to solve for the value of every key bit using the equation $AK=W$

Where $K, W$ are vectors (1) where is a vector of Hamming weights, $wi$; a binary vector of the key bits, $kj$; and $A$ is a binary matrix such that $Aij$ is 1 if and only if weight $wi$ includes key bit $kj$. Even algorithms with more than 56 key bits, such as triple-DES would be vulnerable to this attack. If transition count information rather than Hamming weight information is available, an SPA attack can still weight information is available, an SPA attack can still be mounted, but it may be more difficult. The attacker would need to know the contents of the data bus before or after the data being sought is accessed. Many times this data is easy to determine because it is a fixed address or an instruction opcode. Attackers with access to the code can easily get this data and set up equations similar to Equation $AK=W$. Other less knowledgeable attackers may need to resort to trial and error to determine the correct equations. Due to the limited number of possibilities such an approach is reasonable. Our experiments confirmed that poor implementations of DES will almost always be vulnerable to SPA attacks. Shifting the key bytes or the use of conditional branches to test bit values can be especially vulnerable. Also, if the code or portions of the code run in variable time, power analysis could be used to enable a timing attack. Fortunately, implementors of cryptographic algorithms have known about these issues for a number of years. Kocher et al. reported that it is not particularly difficult to build SPA-resistant devices. However, the attackers keep getting smarter so further research and vigilance will always be necessary.
These results show how the data affects the power levels. The nine overlayed waveforms correspond to the power traces of different data being accessed by an LDA instruction. These results were obtained by averaging the power signals across 500 samples in order to reduce the noise content. The difference in voltage between traces of different data being accessed by an LDA instruction. These results were obtained by averaging the power +1 transitions is about 6.5 mV.

**IV. POWER ANALYSIS ATTACKS ON CMOS**

Most digital circuits are based today on CMOS technology, using complementary transistors as a basic element. When a CMOS gate changes its state, it charges/discharges a parasitic capacitive load and causes a dynamic short circuit of the gate. The more gates changing their state, the more power is dissipated. The current consumed by a circuit can be measured by placing a small resistor or a transformer in the power supply line. Drivers on the address and data bus consist of many parallel inverters per bit, each driving a large capacitive load. During transition they cause a significant power surge in the order of 0.5–1 mA per bit, which is sufficient to estimate the number of bus bits changing at a time. By averaging the measurements of many repeated identical operations, smaller transitions can be identified. Of particular interest for cryptographic algorithms is the state change of a carry bit. In order to reconstruct the algorithm or find the secret key, power consumption measurements for different input data are normally done with a digitizing oscilloscope at a few hundred megahertz. After that, the acquired traces are transmitted to a computer for comparison and post processing.

---

**C. Simple power analysis (SPA): difference in instruction flow.**

- 8-byte password check in Freescale MC908AZ60A microcontroller
- 1 byte at a time, 1 of 256 attempts leads to distinctive power trace
- full password recovery in 2048 attempts (less than 10 minutes)
Each type of instruction executed by a CPU causes different levels of activity in the instruction decoder and arithmetic unit, therefore it can often be clearly distinguished, making it possible to reconstruct passwords and parts of algorithms. Power traces acquired from two identical samples may be different because of fabrication variations between devices – different location on the wafer, slight size variations in transistors and wires, variations in leakage currents and parasitic capacitance. This results in a large residual difference when two traces from different samples are compared. Therefore, power traces from the same device are more easily compared.

Even for the same portion of code running on the same device the power traces can be different if the code is executed from different memory locations, because address decoders also contribute to the power consumption.

When looking for data, for example in a cryptographic operation, it is important to understand that the power fluctuations are affected by the number of bits set or reset. As a result, only the Hamming weight of data (number of bits set) can be estimated, rather than the actual value.

However, because of small variations between transistors on the chip and different lengths of wires, there is still a small difference in power traces, even for the same Hamming weights. For some Hamming weights, such as one or the bus width minus one, it is practical to distinguish between data values by averaging a large number of traces, thus reducing the noise and increasing the resolution.

The waveforms of the four respective analysis are given in the next page

1. Difference between instructions in a PIC16F84 microcontroller
2. Guessing the password in a HC908AZ60A microcontroller
3. Difference between traces from two samples of PIC16F84
4. Difference between the same code run at different addresses.

V. CONCLUSIONS

• There is no such a thing as absolute protection
  – given enough time and resources any protection can be broken
• Technical progress helps a lot, but has certain limits
  – do not overestimate capabilities of the silicon circuits
  – do not underestimate capabilities of the attackers
• Defence should be adequate to anticipated attacks
  – security hardware engineers must be familiar with attack technologies to develop adequate protection
  – choosing the correct protection saves money in development and manufacturing
• Attack technologies are constantly improving, so should the defence technologies
• Many vulnerabilities were found in various secure chips and more are to be found posing more challenges to hardware security engineers.

ACKNOWLEDGEMENT

A successful and satisfactory completion of any significant task is the outcome of invaluable aggregate combinations of different people in the radial direction, explicitly and implicitly.

I would therefore take this opportunity to thank and express my gratitude to all those without whom this document wouldn’t be possible.

I would like to thank our project in-charges Mr. Kartheek. B. N for his unfailing encouragement, invaluable guidance and suggestions given to us.
FUTURESCOPE

Since their invention in 1998, many variants of power-analysis attacks on public-key cryptosystems have been published. As a matter of fact, all types of public-key cryptosystems contain operations that are susceptible to power-analysis attacks. Fortunately, also a broad variety of countermeasures is available. It seems that more countermeasures are available for cryptosystems based on elliptic curves, simply because of the diverse arithmetic they offer and the many representations of points and curves themselves. To conclude the observations made in the previous sections and referenced papers: RSA-type cryptosystems are less likely to be susceptible to simple power-analysis attacks, but offer less possibilities for countermeasures against differential power-analysis attacks. EC-based cryptosystems are highly susceptible against simple power-analysis attacks but many different countermeasures can be implemented. Some countermeasures are probably subject to a patent, and some will not give a sufficient protection on their own. In future work, micro-architectural countermeasures against DESs will be investigated. The real challenge will be to achieve high security performance without compromising requirements of embedded crypto-systems.

REFERENCE

[2]. Side channel attacks and counter measures for embedded systems by Job De Hass
[3]. Smartly Analyzing the Simplicity and the Power of simple power analysis on Smart cards by Rita Mayer Sommer