

Securing Sensor Nodes Against Side Channel Attacks

Kanthakumar Pongaliur¹ Zubin Abraham¹ Alex X. Liu¹ Li Xiao¹ Leo Kempel²

¹ Department of Computer Science and Engineering

² Department of Electrical Engineering

Michigan State University, East Lansing, Michigan 48824, U.S.A.

{pongaliu, abraha84, alexliu, lxiao}@cse.msu.edu, kempel@egr.msu.edu

Abstract

Side channel attacks are non-invasive attacks in which adversaries gain confidential information by passively observing the target computing device. Sensor nodes are particularly vulnerable to side channel attacks due to the lack of protective physical shielding and their deployment in open environments. As sensor nodes are increasingly being deployed in safety critical applications such as power grid, volcano monitoring, and even military applications, protecting sensor nodes from side channel attacks is critical. However, side channel attacks on sensor nodes have not been investigated in previous work. In this paper, we present a taxonomy of side channel attacks on sensor nodes. For each type of the attacks, we provide guidelines and approaches to thwart the attack. We also propose a new technique, called process obfuscation, which can be used as a countermeasure for a variety of side channel attacks on sensor nodes. Furthermore, to demonstrate the feasibility of side channel attacks, we conducted electromagnetic leakage attacks, a type of side channel attack, on popular Tmote-sky sensor nodes using commercially available equipment.

1. Introduction

Wireless sensor networks, which primarily consist of a number of autonomous sensors to collaboratively monitor physical and environmental conditions, have become ubiquitous in today's world, finding applications in the fields of military surveillance, health care monitoring, environmental monitoring, etc. These autonomous sensors, due to their distributed nature and limited physical shielding, are vulnerable to attacks, in particular, non-invasive attacks. Although it is possible for adversaries to launch invasive attacks once they obtain access to sensor nodes, there are many reasons that adversaries prefer non-invasive attacks. Some of these are listed below:

- The adversary may not have direct physical access to the node, for example, the node is embedded within a concrete wall or beneath a road surface.
- The node may not be in physically accessible proximity.
- The node is built with tamper resistant material.
- The adversary does not want his attack to be discovered, especially when sensor nodes are used in critical (such as military) applications.
- The adversary may have access to only one node and would not want to damage it using invasive attacks.

This paper considers side channel attacks, in which an adversary accesses a sensor node in a non-invasive (i.e., non-tampering) manner and gains confidential information by observing the node under normal operation. In such attacks, the goal of the adversary is to deduce the inner workings of the hardware or the software. The adversary may use a variety of techniques such as power analysis (simple power analysis and differential power analysis), execution cycle frequency analysis, timing information (on data movement into and out of the CPU) analysis, electromagnetic leakage analysis, acoustic emission analysis, etc.

Previous work on side channel attacks has studied information leakage from monitors [29], keyboards [6, 10, 39], consumer mobile devices (such as PDAs, pagers) [18], IC chips [37], smart cards [32], etc. However, no previous work has focussed on preventing side channel attacks on sensor nodes. Compared to other computing devices, sensor nodes are particularly vulnerable to side-channel attacks for two major reasons. First, sensor nodes are typically deployed in unsafe environments that could be easily accessible to adversaries. Second, sensor nodes are typically not encapsulated in tamper-proof bodies due to the demand on reducing manufacturing costs. A large number of physical components of a sensor node are usually exposed, thereby

giving the attacker sufficient closeness to the specific modules of the sensor node in certain cases. Under these circumstances, the attacker may initiate attack on a part of the node by collecting leaked information that is available at close proximity. Thus, even a node that is compliant with FCC regulations may still be leaking sufficient information through its side channels for an adversary to obtain confidential information. As sensor nodes are increasingly being deployed in safety critical environments such as battle fields and power grids, preventing side channel attacks is indispensable in securing sensor networks. Because of the non-invasive nature of side channel attacks, they are often used together with other kinds of (possibly invasive) attacks.

The unique characteristics of sensor nodes make the prevention of side channel attacks more challenging. First, sensor nodes have low CPU power. A sensor node typically runs a 4-8MHz microcontroller. Second, sensor nodes have a small amount of memory. A sensor node has RAM and flash memory in the order of few hundred Kilobytes. Third, sensor nodes have limited battery life. Some sensor nodes are powered by two AA batteries [2,22], and some are powered only by type-5 hearing aid batteries [7]. Fourth, sensor nodes have low bandwidth. They have a restrictive wireless bandwidth of 19.2 Kbps. This low bandwidth channel becomes even more important because of the hostile environment where the channel might have lossy characteristics. They generally have an integrated on board antenna with a transmission range of 100 - 300Ft. Fifth, sensor nodes are often deployed in unguarded and even hostile environments. Last but not least, most of their physical components are usually exposed. These physical constraints make previous solutions for preventing side channel attacks on resource rich devices unsuitable for sensor nodes, since most of the solutions increase the power consumption.

This paper represents the first comprehensive study of side channel attacks on sensor nodes. We first formalize a three-phase attack model on sensor nodes. Second, we identify a family of side channel attacks on sensor nodes, which includes electromagnetic radiation leakage attacks, optical side channel attacks, traffic analysis attacks, power analysis attacks, timing attacks, fault analysis attacks, acoustic attacks, and thermal imaging attacks. For each type of the attacks, we present countermeasures to limit vulnerability. We also propose a new technique, called process obfuscation, which can be used to counter a variety of side channel attacks. Last, our experimental results show the feasibility of electromagnetic radiation leakage attacks using readily available equipment.

The rest of this paper is organized as follows. Section 2 summarizes related work. We describe the generic attack model in Section 3. We identify a family of side channel attacks on sensor nodes and present their countermeasures in Section 4. In Section 5, we show the feasibility of a side-

channel attack by taking the electromagnetic attack as a special case. We summarize the paper and lay out future work in Section 6.

2 Related Work

Roosta *et al.* was the first to briefly discuss the possibility of side channel attacks (also referred to as tempest [28]) in sensor networks [34]. Okeya *et al.* demonstrated the feasibility of side-channel attacks by passively monitoring the power consumption waveform of IC chips using an oscilloscope [33]. They presented differential power analysis attacks by selective forgery of MAC and demonstrated how several key bits may be extracted. Similarly, Kocher *et al.* showed that monitoring the electrical power consumption of a smart card running the DES algorithm was sufficient to retrieve the secret key [27].

Gratzner *et al.* successfully demonstrated a side channel attack with neither the knowledge of any information about the algorithms nor the microprocessor power consumption model [21].

Cryptographic techniques to counter side-channel attacks on smart cards and IC chips have been proposed by Micali *et al.* in physically observable cryptography [31]. Alternately, theoretical approaches like building private circuits in a boolean context have been proposed to counter side channel attacks [24]. Using this approach, any n -gate circuit that is allowed to leak up to t bits at a time, can be converted to a perfectly secure circuit of size $O(nt^2)$ with an increase in circuit size by factor of $O(t^2)$. The limitation of the technique is its weakness to probing attacks where the information retrieval is dependent on a limited number of physical wires. Ishai *et al.*, in a follow up paper, proposed a method to detect tampering even in a fully compromised circuit followed by self destruction of the circuit [23].

Chevallier-mames *et al.* proposed a low cost solution in terms of both complexity and computational speed, which creates side channel atomic blocks where the whole code of a process appears as a succession of blocks that are indistinguishable by simple side-channel analysis [14]. This method has the advantages of being inexpensive and generic.

Batina *et al.* talked about fault attacks that reveal secret information by inserting faults into the device cryptosystem and concluded that clear box testing is necessary in the case of side channel attacks and testing cannot be limited to black box [9].

To gauge the effectiveness of the various strategies proposed to counter side-channel attacks, evaluation needs to be performed at the field level. Kocher provided means for testing and evaluating how the various secure strategies to counter side-channel attacks perform in practice [26].

Acoustic side channel attacks with sound emanating

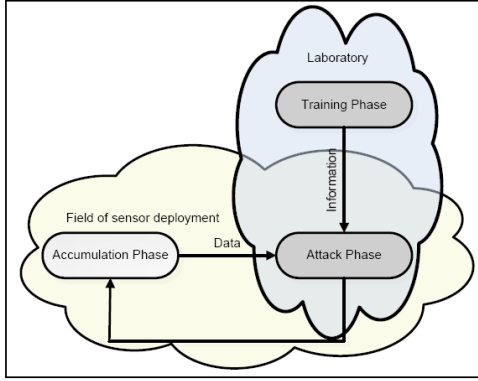


Figure 1. Attack Model

from a keyboard was studied by Asonov *et al.* [6]. This was further revised by Zhuang's group in [39]. Similarly, Shamir and Tromer provided a proof of concept of vulnerable acoustic emissions from computing components [35].

Most of the work discussed here corresponds to systems in general. Among the few that have an architecture of the order of a sensor node is the smart card. The smart card can be considered to have similar limitations as the sensor node though it does not involve any distributed processing as in the case of sensor network, which brings in unique side-channel vulnerabilities.

3 The Attack Model

In this section, we present a model of side channel attacks on sensor nodes. This model is not specific to any particular type of sensor node. In this model, we assume that an adversary can get close to a sensor node, but the physical access of the node is restricted. An adversary may use specialized attack strategies for different components of a node. Depending on the type of node or the functionality of the node, an adversary can adapt attack strategies. For example, if an adversary identifies the node as a gatherer of event information, which means the node needs to write the information into memory, then the adversary may focus on obtaining information leaked during memory access cycles. As another example, if a node is identified as an aggregator of some confidential information, then the adversary may want to focus on obtaining the information leaked during the node's execution of the aggregation algorithm.

Example instruments used by an adversary could be Agilent E7401A EMC Analyzer [1], Agilent 11955A biconical antenna [1], and a log-periodic antenna. The biconical antenna is large in size and can measure up to 300MHz while the log-periodic antenna can measure up to 2GHz. Other commercially available equipments could also be used.

Our attack model consists of three phases as depicted in Fig. 1.

- **Training Phase:** Depending on the functionality of the sensor node from which the information leakage is studied, adversaries may train themselves by studying the information leakage from a similar sensor node performing the same functionality. This training process could iterate multiple times, thereby making it easy to decipher the information leaked. This phase can be accomplished by the adversary at its own convenience and does not need the adversary to be in the field of attack.
- **Accumulation Phase:** In this phase, the adversary passively monitors the environment in which the node is deployed and in the due process gathers the data leaked.
- **Attack Phase:** In this phase, the adversary uses the information learned in the training phase to decipher the valid information from the data gathered in the accumulation phase. The attack phase can be either integrated with the accumulation phase and implemented dynamically in the field; or it can be a stand alone phase, implemented in a static manner after the completion of the accumulation phase possibly in a better equipped environment.

4 Taxonomy of Attacks and Countermeasures

In this section, we describe the different types of side-channel attacks on sensor nodes and propose corresponding countermeasures. The goal of these countermeasures is not to fully prevent side channel attacks, but to make the attacks practically infeasible. Before we present the taxonomy of side channel attacks and their specific countermeasures, we first give two general counter measures. These general countermeasures are applicable to multiple side channel attacks, and they may be used in conjunction with other attack-specific countermeasures. In designing countermeasures for side channel attacks, we need to accommodate sensor node limitations such as slow CPU speed, limited battery life, small memory capacity, and low communication bandwidth.

4.1 General Countermeasures

The two general countermeasures are obfuscation and tamper-proofing.

4.1.1 Obfuscation

Obfuscation techniques are used to conceal the meaning of a certain computation by making it obscure and harder to understand. Code obfuscation [15, 30] has been proposed previously as a technique to limit side channel information leakage. In this paper, we propose a new obfuscation technique, called *process obfuscation*, to circumvent side channel attacks.

Code Obfuscation: Code obfuscation is a process that has been studied well previously [5, 15, 30] and it can be described as follows. Suppose that (part of) an algorithm consists of a loop where the execution of a given set of instructions depends on certain input values. If, from some side-channel information (e.g., timing, power consumption etc.), one can distinguish which set of instructions is processed, then one can retrieve some secret data (if any) involved during the course of the algorithm. To prevent this from happening, we can create the code such that the different paths of the execution cycle take the same amount of time or the same execution path over different execution cycles takes different amount of times. This can be achieved by using fake instructions in the code so as to either maintain the execution time over different paths or to randomly differ the execution time over the same path. Code obfuscation techniques for prevention of side-channel attacks has been classified as follows [38]

- **Data Independent Calculations:** The basic idea behind this technique is to make the amount of time for performing operations independent of the data being processed.
- **Blinding:** Blinding is a technique in which the content of the message is disguised before it is signed. This technique is inspired from blinding signatures [13], which was further elaborated by Kocher [25]. It is especially useful for preventing electromagnetic side-channel attacks.
- **Avoid conditional branching:** This technique avoids conditional branching wherever possible. In the case where a conditional branch is required, we take precautions to enforce the same execution time in each branch.

If the sensor node is capable of simultaneously possessing multiple encryption algorithms (depending on memory/space availability), it can randomly select the encryption algorithm it uses. In doing so, we will need to make the side-channel characteristics like power consumption, timing etc. of the same order. Else, it may defeat the purpose of having such an implementation.

Process Obfuscation: In this paper, we propose the technique of process obfuscation in which we take reactive decisions during runtime to mask the process being executed. For example, process obfuscation can be attained by executing *process₀* followed by a fake execution of *process₁* when *process₀* must be executed, and by executing a fake execution of *process₀* followed by *process₁* when *process₁* must be executed. Obfuscating a process can be achieved in two ways, namely a proactive methodology in which we obfuscate each and every process and a reactive methodology in which we obfuscate only those processes that can end up leaking information through the side-channel.

Given the resource constraints of a sensor node, the proactive approach is not suitable. Hence, we suggest using the reactive approach. Next, we discuss a simple methodology with which we can attain process obfuscation while consuming just the minimum resources. We will base our reactive method on the past transmissions or computations. If the past n computations had a recurring pattern p and the frequency of recurrence of the pattern p in n is greater than the leakage threshold λ , then we will need to take a deviation from the actual processing and perform obfuscation. Unless some countermeasure is taken, the recurring pattern will leak information. The value of λ can be varied and will be based on the sensitivity of data under consideration.

4.1.2 Tamper-proofing

A sensor node in a tamper-proof body ensures that any attempt to break the body results in all confidential information in the node being destroyed. The main hindering factor of building sensor nodes in tamper-proof bodies is cost. Indeed, there are many advantages of tamper-proofing other than obstructing side channel attacks. For example, a tamper-proof body could prevent wear and tear due to environmental factors, thereby increasing the overall life of the node assuming we have sufficient power availability. In practice, the encapsulation body of a sensor node may not need to be tamper resistant for preventing side channel attacks. A simple cover over the sensor node sometimes suffices and reduces the strength of certain side channel leakages, while also preventing unrestricted access to internal components.

4.2 Taxonomy of Attacks

In this section, we classify side channel attacks on sensor nodes into eight categories, namely power analysis attacks, electromagnetic leakage attacks, optical side channel attacks, traffic analysis attacks, timing attacks, fault analysis attacks, acoustic attacks, and thermal imaging attacks. For each type of attack, we present corresponding countermeasures.

4.2.1 Power Analysis Attacks

Power analysis attacks were first studied by Kocher *et al.* [27]. In power analysis attacks, an adversary studies the power consumption of devices, especially cryptographic modules. Power analysis attacks require close proximity to a sensor node, such that an adversary can measure the power consumption of the sensor node. There are two types of power analysis, namely simple power analysis (SPA) and differential power analysis (DPA). In differential power analysis, the adversary studies the power analysis and is able to apply mathematical and statistical principles to determine the intermediate values.

Countermeasures: The countermeasures are implemented by either preventing or complicating power analysis attacks. Prevention of power analysis attack can be easily achieved by introducing a tamper resistant body. This increases the one time cost of the sensor node, but will allow the node to conserve the power usage when compared with other countermeasures. Complicating power analysis attacks can be achieved by several strategies, which we list as follows:

- **Power randomization:** Power randomization is a technique in which a hardware module is built into the chip that adds noise to the power consumption [17]. This countermeasure is simple and easy to implement, but is not energy efficient. Also, it adds to the fabrication cost of the device.
- **Novel Circuit designs:** In [23, 24], Wagner *et al.* gave solutions to design hardware circuits that mask the changes in power consumption so as to increase the cost of a power analysis attack. Again, this increases the cost of manufacturing sensor nodes.
- **Obfuscation:** Obfuscation is a good solution to prevent SPA, but is susceptible to DPA [27].

4.2.2 Electromagnetic Leakage Attacks

Electromagnetic radiations are emitted and propagate following Maxwell's equations. The electromagnetic radiations attain importance when they are hardware generated emissions, especially emissions from the cryptographic module. Electromagnetic leakage attacks have been shown to be more successful than power analysis attacks on chips [3]. In the case of sensor nodes, the circuitry is exposed and hence leads to stronger emanations of EM radiations. Since the circuitry is exposed, it provides an easier environment to study the electromagnetic emanations from each individual component of the sensor node by appropriate probe placement.

Similar to power analysis, electromagnetic analysis can be performed at two levels of sophistication, namely a simple analysis called simple electromagnetic analysis (SEMA) and a differential analysis called DEMA. Furthermore, the emanations can be either direct emanations from a component or a mixture of emanations from a group of components. We performed experiments on EM emanations from sensor nodes, which will be discussed further in Section 5.

Countermeasures: The first countermeasure to prevent electromagnetic information leakage is to encompass the node with a casing so as to prevent access to individual components in a sensor node. The second countermeasure would be the use of secret shares in which the original computation is divided probabilistically such that the power subset of shares is statistically independent. The use of secret shares was proposed by Goubin *et al.* [20]. One of the major drawbacks of this solution is the increase in the power consumption due to the number of operations that are almost doubled. The third countermeasure is to use masking methods [4, 16, 19]. Masking is a scheme in which the intermediate variable is not dependent on an easily accessible subset of secret key [16]. This results in making it impossible to deduce the secret key with partial information gathered through EM leakage.

4.2.3 Optical Side Channel Attacks

The intensity of light emissions from a monitor or liquid crystal display could be used to study the contents of the last displayed screen. Given the form-factor optical side-channel attacks on sensor nodes are formulated differently from the attacks on devices that use a visual display to output information. The sensor nodes have light emitting diodes (LED), which have two primary purposes. The first purpose is in debugging the application program while programming the node. The second use of the LED is for the purpose of signaling. LEDs are externally visible to both a user as well as an adversary, unless the node is used for an application in which they are not in the line of sight.

There are often multiple sets of LEDs on a sensor node having different functionality. For instance, one set lights up during the programming of the node; one set of LEDs display the transmission or reception of data; one set of LEDs are programmed to display information for various application purposes. These LEDs can leak valuable information. For example, the adversary can program his attacking station to listen to the channel only when there is an optical signal from the node, thereby conserving a lot of its (the adversary's) battery. Another case is during node re-programming by the base station. The adversary is alerted about the occurrence of changes to the node by just observing the optical side-channel.

Countermeasures: We first need to prevent LEDs from leaking any confidential information by the way they light up. Any signaling information, which can be used by legitimate users for their application, can also be used by adversaries with malicious intent. Secondly, sensor node programmers should remove the debugging information for which the programmer gets a feedback from the LED, before sending it to production. This is an important requirement because in the event of a malfunction, the execution of the debug code may result in information leakage from the optical side-channel. We recommend to disable all the LEDs on sensor nodes prior to deployment.

4.2.4 Traffic Analysis Attacks

Traffic analysis attacks are attacks that analyze traffic flow to gather topological information. This traffic flow could divulge information about critical nodes, such as the aggregator node in a sensor network. Such attacks primarily relate to the intermittent transmissions that are inherent in sensor networks. Due to the limited energy capacity of nodes and the fact that the transceiver component of a node consumes the most power, the nodes in a sensor network limit the use of the transceiver to transmit or receive information either at a regulated time interval or only when an event has been detected. This generally results in an architecture comprising some aggregator nodes (also called supersensor or actor) within a sensor network. Aggregator nodes are the sensor nodes whose primary purpose is to relay transmissions from nodes toward the base station in an efficient manner, instead of monitoring events like a normal node.

The added functionality of acting as a hub for information gathering and preprocessing before relaying makes aggregator nodes an attractive target to side channel attacks. A possible side channel attack could be as simple as monitoring the occurrences and duration of computing activities at an aggregator node. If a node is frequently in active states (instead of idle states), there is high probability that the node is an aggregator node and also there is a high probability that the communication with the node is valid. Such leakage of information is highly undesirable because the leaked information could be strategically used by adversaries in the accumulation phase of an attack.

Countermeasures: It is practically inefficient to prevent adversaries from identifying aggregator nodes because camouflaging traffic in sensor networks is power intensive. Consequently, we focus on preventing adversaries from identifying valid aggregation cycles of aggregator nodes. One solution to counter such attacks is to have each aggregator node execute dummy operations that resemble the average power consumption curve observed during the normal operation of the aggregator node. This additional requirement

on aggregator nodes, despite the fact that it would result in additional power consumption, can be met, because unlike ordinary nodes in a sensor network, aggregator nodes are typically equipped with longer battery lives owing to their integral nature and functionality with the sensor network.

Apart from simulating the power consumption of a genuine process execution, the two necessities that the execution of the dummy process must incorporate to be successful in thwarting the accumulation phase are to use a different dummy execution process each time or have a low repetition rate. This should help prevent the attacker from finding a pattern that would differentiate the execution of a dummy process from the normal execution of an aggregator node. The second requirement relates to the timing of the execution of the dummy process. Depending on whether there is a pattern to the timing of the execution of a dummy process, an adversary may be able to identify and disregard the dummy process. For example, if an adversary is capable of identifying the presence or absence of a radio transmission, the attacker can disregard any power consumption curve computed during the absence of transmission signal. Similarly, if the dummy process is not executed every time the aggregator node receives a transmission, the attacker will be able to identify invalid transmission. Hence, to ensure the effectiveness of this scheme, the dummy process must be executed each time the aggregator receives a transmission as well as randomly during idle periods. The advantage of incorporating dummy processes in an aggregator is to minimize the ease of identifying transmission flow in a sensor network that can be used to identify the base station of the sensor network, which could be highly confidential in critical applications (such as military applications).

4.2.5 Timing Attacks

The attack involves exploiting the variance in execution time for different branches in the cryptosystem. This becomes all the more important in sensor networks subject to the slower processors used in the nodes. The slower processors will enhance even small difference in computation time over different branches. Timing attacks could also study the number of memory accesses and the time variance in doing the same.

Branch attack: There are studies that deal with branch attacks separately, but we consider branch attacks to be a subset of timing attacks since it is the time variation that is exploited in the case of branch attacks.

Countermeasures: One of the most relevant countermeasure for such attacks is using more clock cycles such that branching does not effect the execution time. Also, the memory access times should be standardized to be the same over all accesses. Since time as a resource is available in

abundance in a sensor network (as compared to power), we can slow down the access times by adding sufficient delay to normalize the access times. These countermeasures have the same drawback that we have reiterated so far: increased power consumption.

4.2.6 Fault Analysis Attacks

Fault analysis attacks (also called fault induction attacks) are attacks in which useful information gets leaked out due to occurrence of fault in the cryptosystem. The faults may occur naturally or be induced by adversaries. Adversaries may inject faults in two ways. One is to give programs invalid input, which may cause buffer overflows. The other is to use equipment such as a laser pointer to illuminate SRAM (EPROM and EEPROM can also be modified) to flip some bits in memory [36]. Fault analysis attacks have been discussed in detail by Biham and Shamir [11]. The importance of checking cryptographic protocols for faults was detailed by Boney *et al.* [12].

Countermeasures: To counter fault analysis attacks, we suggest to use redundancy to catch injected faults. The idea is similar to N-version programming [8]. For certain critical functions, we deploy multiple implementations of the same function. Given an input, we process it using the various implementations and compare the outputs. A selection module could be incorporated to decide the valid output. Although sensor nodes have limited resources, critical regions usually comprise the crypto functions, which need to be secured.

4.2.7 Acoustic attacks

There are two types of acoustic emissions that have been studied previously: acoustic emissions from keyboards [6, 39] and acoustic emissions from computing components such as CPU and memory [35]. Acoustic emissions are produced by a keyboard when different keys are pressed and can be used to identify the keys being pressed with extra triangulation information. Acoustic emissions from computing components have been demonstrated by Shamir *et al.* to be exploitable [35]. As sensor nodes typically do not have keyboards, the acoustic emission from its exposed computing components is a real concern.

Countermeasures: One practical and effective countermeasure for acoustic attacks is to encapsulate a sensor node in sound absorbing material. Another countermeasure is to introduce random acoustic noise of similar frequency to obfuscate acoustic emissions from sensor nodes.

4.2.8 Thermal Imaging attacks

Thermal imaging attacks differ from acoustic attacks in that the emission being exploited is heat instead of sound. Such attacks often exploit the infrared images emanating from CPUs.

Countermeasures: To counter thermal imaging attacks, one approach is to use a dual layered case with the inner layer a highly conducting surface and the outer layer made of a non-conducting material. When heat is generated from internal computing components, the inner, highly conducting surface will quickly dissipate the heat around. The outer layer prevents accesses to the temporary hot spots formed on the inner layer.

5 Electromagnetic Leakage Attacks - A case study

As a proof of concept, in this section, we demonstrate the feasibility of side channel attacks on sensor nodes. We choose electromagnetic leakage attacks in our case study due to the availability of requisite equipment.

5.1 Experimental Setup

The experiments were conducted in a radio frequency anechoic chamber to suppress the electromagnetic wave analogy of echos and also reduce the interference of electromagnetic radiations from other devices. We use the following setup for our measurements.

- Tmote-sky sensor node: Program Space 48kB, RAM 10kB, Frequency 2400-2483 MHz. These types of sensor nodes are popular and commercially available [2].
- Agilent E7401A EMC Analyzer. We use it to analyze the electromagnetic radiations captured by the antenna.
- Log-periodic antenna. We use it to capture electromagnetic emission up to 2GHz frequency.

The sensor node was programmed to send consecutive zeros or consecutive ones at different intervals of time. We conducted experiments by placing the antenna at different distances of 10 feet, 3 feet, and less than 1 feet from the sensor node. We chose to send zeros and ones because we wanted to study the electromagnetic radiation under the fundamental but antipodal data. Different distances were also selected to see if there was any prominent variance in the electromagnetic radiation strength over distance.

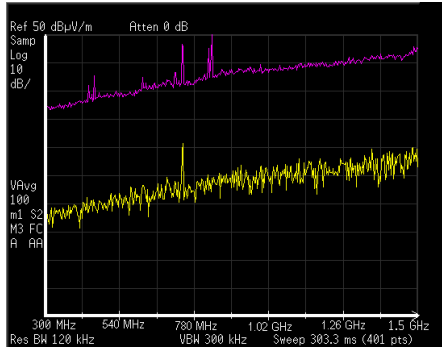


Figure 2. Transmit 0's

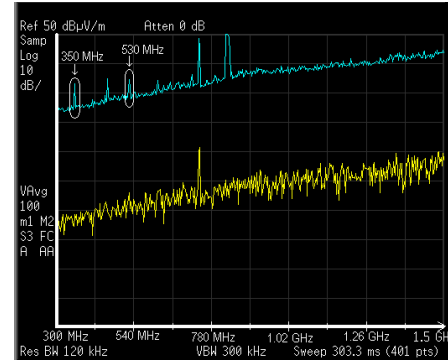


Figure 3. Transmit 1's

5.2 Observations

We observed unique peaks in the frequency readings that correspond to the input. These readings were consistent over multiple runs emphasizing the correlation between the input and the unique peaks. There was no significant impact of distance over signal strength. Consequently, we only present results from one set of observations depicted in Figs. 2 and 3. The bottom plot (in yellow color) in the Figs. 2 and 3 represents the idle state, while the top plot in Fig. 2 represents the resultant electromagnetic radiations while transmitting data packets containing zeroes. Similarly, the top plot in Fig. 3 represents the electromagnetic radiation while transmitting data packets containing ones. As depicted in the graphs, distinctive peaks are observed at 350MHz and 530MHz while transmitting packet containing ones (Fig. 3) that are absent while transmitting packets containing only zeroes (Fig. 2). This leads to the conclusion that there exists a strong correlation between the electromagnetic radiation and the data processed in the sensor. We noticed the use of an anechoic chamber highly reduced the ambient noise.

6 Conclusions and Future Work

The lack of physical shielding and the deployment in open environments make sensor nodes particularly vulnerable to side channel attacks. This paper represents the first comprehensive study of side channel attacks on sensor nodes. We make five key contributions. First, we present an attack model applicable to side-channel attacks. Second, we present a taxonomy of side channel attacks on sensor nodes. Third, for each type of attack, we provide guidelines and approaches to thwart the attack. Fourth, we propose the technique of process obfuscation, which can be used as a countermeasure for a variety of side channel attacks. Last, we show our experimental results on conducting electromagnetic leakage attacks on Tmote-sky sensor nodes. Our results show the feasibility of launching side channel attacks on sensor nodes.

This paper opens a large domain of possibilities for future work. For example, the impact of electromagnetic leakages on the security of sensor nodes is certainly worthy exploring.

References

- [1] <http://www.home.agilent.com/agilent/home.jsp>.
- [2] <http://www.moteiv.com>.
- [3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-Channel(s). In *CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 29–45, London, UK, 2003. Springer-Verlag.
- [4] M.-L. Akkar and C. Giraud. An implementation of des and aes, secure against some attacks. In *CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, pages 309–318, London, UK, 2001. Springer-Verlag.
- [5] A. Alarifi and W. Du. Diversify sensor nodes to improve resilience against node compromise. In *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pages 101–112, New York, NY, USA, 2006. ACM.
- [6] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy*, pages 3–11, 2004.
- [7] B. Atwood, B. Warneke, and K. Pister. Preliminary circuits for smart dust. *Southwest Symposium on Mixed-Signal Design*, pages 27–29, 2000.
- [8] A. Avizienis and L. Chen. On the implementation of N-version programming for software fault-tolerance during program execution. In *Proceedings of Intl. Computer software and Appl. Conf.*, pages 145–155, 1977.
- [9] L. Batina, N. Mentens, and I. Verbauwhede. Side-channel issues for designing secure hardware implementations. In *Proceedings of 11th IEEE International On-Line Testing Symposium*, pages 118–121, Washington, DC, USA, 2005. IEEE Computer Society.
- [10] Y. Berger, A. Wool, and A. Yeredor. Dictionary attacks using keyboard acoustic emanations. In *CCS '06: Proceedings of*

- the 13th ACM conference on Computer and communications security, pages 245–254, New York, NY, USA, 2006. ACM Press.
- [11] E. Biham and A. Shamir. Differential fault analysis of secret key cryptosystems. volume 1294, pages 513–525, 1997.
 - [12] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults. volume 1233, pages 37–51, 1997.
 - [13] D. Chaum. Blind signatures for untraceable payments. *CRYPTO 82: Advances in Cryptology*, pages 199–203, 1982.
 - [14] B. Chevallier-Mames, M. Ciet, and M. Joye. Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity. volume 53, pages 760–768, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
 - [15] C. Collberg, C. Thomborson, and D. Low. A taxonomy of obfuscating transformations. Technical Report 148, July 1997.
 - [16] J.-S. Coron and L. Goubin. On boolean and arithmetic masking against differential power analysis. In *CHES '00: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems*, pages 231–237, London, UK, 2000. Springer-Verlag.
 - [17] J. Daemen and V. Rijmen. Resistance against implementation attacks: A comparative study of the aes proposals. In *The Second AES Candidate Conference*, pages 122–132, Gaithersburg, MD, 1999. National Institute of Standards and Technology.
 - [18] C. H. Gebotys, C. C. Tiu, and X. Chen. A countermeasure for EM attack of a wireless PDA. In *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I*, pages 544–549, Washington, DC, USA, 2005. IEEE Computer Society.
 - [19] L. Goubin. A sound method for switching between boolean and arithmetic masking. In *CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, pages 3–15, London, UK, 2001. Springer-Verlag.
 - [20] L. Goubin and J. Patarin. DES and differential power analysis (the duplication method). In *Cryptographic Hardware and Embedded Systems*, pages 158–172, 1999.
 - [21] V. Gratzer and D. Naccache. Blind attacks on engineering samples. Cryptology ePrint Archive, Report 2005/468, 2005.
 - [22] J. Hill, R. Szwedczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.
 - [23] Y. Ishai, M. Prabhakaran, A. Sahai, and D. Wagner. Private circuits 2: Keeping secrets in tamperable circuits. In *Proceedings of Eurocrypt*, pages 308–327, May 2006.
 - [24] Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In *Proceedings of CRYPTO*, pages 463–481, 2003.
 - [25] P. C. Kocher. Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems. In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 104–113, London, UK, 1996. Springer-Verlag.
 - [26] P. C. Kocher. Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks. In *NIST Physical Security workshop*, sept 2005.
 - [27] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.
 - [28] M. G. Kuhn and R. J. Anderson. Soft tempest: Hidden data transmission using electromagnetic emanations. In *Information Hiding*, pages 124–142, 1998.
 - [29] J. Loughry and D. Umphress. Information leakage from optical emanations. In *ACM Transactions on Information and System Security (TISSEC)*, pages 262–289, 2002.
 - [30] D. Low. Protecting java code via code obfuscation. *Crossroads*, 4(3):21–23, 1998.
 - [31] S. Micali and L. Reyzin. Physically observable cryptography. In *Theory of Cryptography*, 2004.
 - [32] S. Moore, R. Anderson, and M. Kuhn. Improving smart-card security using self-timed circuit technology. In *Fourth ACiD-WG Workshop, Grenoble*, pages 211–218, 2002.
 - [33] K. Okeya and T. Iwata. Side channel attacks on message authentication codes. *IPSI Digital Courier*, 2:478–488, 2006.
 - [34] T. Roosta, S. Shieh, and S. Sastry. Taxonomy of security attacks in sensor networks and countermeasures. In *The First IEEE International Conference on System Integration and Reliability Improvements*, December 2006.
 - [35] A. Shamir and E. Tromer. Acoustic cryptanalysis: on nosy people and noisy machines. In *Proceedings of EUROCRYPT*, 2004.
 - [36] S. P. Skorobogatov and R. J. Anderson. Optical fault induction attacks. In *CHES*, pages 2–12, 2002.
 - [37] K. Tiri, D. Hwang, A. Hodjat, B. Lai, S. Yang, P. Schau-mont, and I. Verbauwhede. A side-channel leakage free co-processor IC in 0.18 μ m CMOS for embedded AES-based cryptographic and biometric processing. In *DAC '05: Proceedings of the 42nd annual conference on Design automation*, pages 222–227, New York, NY, USA, 2005. ACM Press.
 - [38] www.discretix.com. Introduction to side channel attacks.
 - [39] L. Zhuang, F. Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 373–382, New York, NY, USA, 2005. ACM Press.