En studie av metoder för manipuleringsresistens

A review of tamper-resistance mechanisms

Pontus Hero Ek
Villiam Rydfalk

Handledare, Marcus Bendtsen
Examinator, Nahid Shahmehri
Abstract

Tamper-resistance is the implementation of some kind of security against physical attacks on a device. Most tamper-resistance methods are focused on preventing the theft of data or keys from a device, while others are focused on detecting or responding to an attack. The focus of our review is mainly on resistance against a number of attack-types that are regarded as the main threats in the area.

In our review we first start by categorising the different tampering attacks and later on listing different methods of resistance that have been of interest in the last decade. We also describe how the different methods of resistance work and which different attacks they counter. Lastly there is a table that contains the various attacks and defences that will give the reader an easy overview of our review.
Acronym list

AES : Advanced Encryption Standard
API : Application Programming Interface
ARM : Advanced RISC Machine
BGA : Ball Grid Array
BIOS : Basic Input/Output System
CPA : Correlation Power Analysis
CPU : Central Processor Unit
DFA : Differential Fault Analysis
DMA : Direct Memory Access
DPA : Differential Power Analysis
DPL : Dual-rail Precharge Logic
EAA : Electromagnetic Analysis Attack
ERIST : Efficient Randomized Instruction inSertion Technique
FIB : Focused Ion Beam
IC : Integrated Circuit
MEMS : Micro-Electromechanical Systems
NEMS : Nano-Electromechanical Systems
OS : Operating System
PAA : Power Analysis Attack
PV : Photo-Voltaic
RAM : Random Access Memory
RISC : Reduced Instruction Set Computing
SCA : Side-Channel Attack
SoC : System on Chip
SPA : Simple Power Analysis
TCG : Trusted Computing Group
TEL : Time-Enclosed Logic
TPM : Trusted Platform Module
## Contents

1 Introduction ................................................................. 1  
  1.1 Aim ................................................................. 2  
  1.2 Delimitations .......................................................... 2  

2 Theory ................................................................. 3  
  2.1 Classes and methods of attacks ........................................... 3  
    2.1.1 Side-channel attacks .............................................. 3  
    2.1.2 Software attacks ................................................ 4  
    2.1.3 Fault generation ................................................ 5  
    2.1.4 Microprobing ..................................................... 5  
    2.1.5 Reverse engineering ............................................ 5  
  2.2 Invasive, non- and semi-invasive ...................................... 6  

3 Method ................................................................. 7  

4 Result ................................................................. 11  
  4.1 Potting ............................................................... 11  
  4.2 Glue logic ............................................................ 11  
  4.3 Ball grid arrays ..................................................... 12  
  4.4 Copy trap ............................................................. 13  
  4.5 Duplicate computation ............................................... 13  
  4.6 Obfuscation .......................................................... 13  
  4.7 ARM architecture, TrustZone and SecurCore ....................... 14  
  4.8 Software encryption on system on chip ........................... 15  
  4.9 Software encryption on internal RAM ............................... 16  
  4.10 Mitigation of SCAs ................................................ 16  
  4.11 Shielding integrated circuits with sensors ....................... 17  
  4.12 Trusted Platform Module .......................................... 18  
  4.13 Result overview table ............................................. 18  

5 Discussion ............................................................ 20  
  5.1 Method .............................................................. 20  
  5.2 Results ............................................................. 21  
  5.3 The work in a wider context ........................................ 21  

6 Conclusion .......................................................... 23
List of Figures

1 Side-channel attack ........................................... 4
2 Potting .......................................................... 11
3 Glue logic design .............................................. 12
4 Ball grid array .................................................. 13
5 Obfuscation ...................................................... 14
6 ARM TrustZone .................................................. 15

List of Tables

1 Search terms 1 .................................................. 9
2 Search terms 2 .................................................. 9
3 Search terms 3 .................................................. 9
4 Search terms 4 .................................................. 9
5 Search terms 5 .................................................. 10
6 Search terms 6 .................................................. 10
7 Search terms 7 .................................................. 10
8 Result overview table ......................................... 19
1 Introduction

In early 2016 Apple released an update for iOS that included a new security measure for iPhone 6 [1]. The iPhone 6 has a finger print sensor and it could be possible to replace the finger print sensor in the iPhone’s home button and thus gain access to a user’s data. To counter this, the home button and the security system in the iPhone are paired in the factory by Apple. If the home button is replaced or altered the finger print sensor will not work unless it is paired again, but the iPhone itself still works. In the iOS software update they included harder control between the sensor and the security responsible processor. Resulting in that if the sensor was no longer trusted, e.g. someone had tampered (“messed with”) with the home button, the iPhone would display error 53 and be rendered unusable. The problem for a lot of users was if they had repaired the home button themselves at some point or had used a third party to repair it, then the sensor was no longer trusted according to the iPhone’s security system. The effect was that the affected phones only showed error 53 on the screen rendering the phones useless. The error 53 was eventually fixed, but the finger print sensor would still not work since the security system in the iPhone did not trust it [2]. The trust between the security system and the iPhone home button is a necessary security measure to make sure the data on the iPhone is not accessible by unauthorised users, this is an example of tamper-resistance.

Tamper-resistance prevents an attacker from altering a device or making an unauthorised access to its data, and different methods are used on almost every product which stores valuable information. Tamper-resistance is critical to ensure that the data on a device is not compromised if the device is lost or stolen. Since modern society uses processors, microchips and all sorts of computing devices for everything between the most mundane tasks to bank transactions, it becomes vital to ensure that personal information or company secrets are not available to anyone who can read the memory of a processor.

Looking back at the iPhone, the fingerprint sensor control is not the only tamper-resistance implemented by Apple. They also have special screws holding the case in place that require a special screwdriver to open. This is another type of tamper-resistance that limits the access to the physical parts of the phone and by extension protects the phone’s data. The special tool required is not impossible to find and buy, but it does limit the amateur users from tampering with the iPhone. The architecture of the processor on the iPhone is furthermore a type of tamper-resistance. By partitioning the sensitive data to different encrypted memories it is difficult to receive the data via software attacks or the security keys via physical attacks. There are probably more tamper-resistance measures on the iPhone and the combination of all different methods make it tough to solve, but not impossible as the FBI managed to gain access to an iPhone 5C in 2016 [3].
1.1 Aim
The purpose of this report is to present different methods of tamper-resistance covering methods described by the research community the recent eight years.

There are two main questions we wish to answer:

1. What different methods of tamper-resistance are used?
2. Which types of attacks do the different methods protect against?

1.2 Delimitations
This report is aimed at an audience with little knowledge of tamper-resistance methods and the technical aspects behind them. Therefore we only make an overview of how the methods work. We will not go into deeper details of the implementation or technical attributes of the methods. We have chosen to prioritise publications of methods for defence regarding tamper-resistance before the year of 2008, since the following eight years have had a great increase in minor devices, such as smartphones.

A tamper-attack can happen under many occasions when a product is being produced. We chose to focus on attacks made against a complete device that has not been tampered with during its production.
2 Theory

There are numerous ways of attacking a device depending on what the goal of the attack is. Usually the attacker is after data stored on the device, however the attack can be setting up for another attack or to figuring out how the device works. Furthermore, as time passes and both the field of attacking and protecting advances, some attacks are more common than others.

In this chapter we will describe the types of attacks that tamper proofing attempts to mitigate. In Section 2.1.1 to 2.1.5 we offer a classification of these attacks, and briefly describe the attacks along with an example of each class. Similarly, in Section 2.7 we classify the different types of attackers.

2.1 Classes and methods of attacks

In this review we use the classification of five attacks found in the works of Anderson and Skorobogatov [4] [5]. This will be the foundation of our review and our results will be focused and based around how to prevent these classes of attacks.

2.1.1 Side-channel attacks

*Side-channel attacks* (SCA) could be seen as a type of eavesdropping. By analysing the unintended outputs of a device it is possible to extract the encryption keys used on the device. Figure 1 illustrates a simplified view of the units Alice and Bob sending a message, such as a security key, between them that is encrypted by Alice and decrypted by Bob. Eve in the middle is trying to read it. The different arrows pointing out of Alice and Bob in Figure 1 are some examples of possible side-channel signals that they might leak when communicating. It could be the heat that is generated by the units circuitry, power consumption of the unit or noise made during the encryption or decryption operations. These can all be measured with different tools and give an attacker information not meant to be read, such as the security key sent between Alice and Bob.

Examples of widely used SCAs are the two different types of *Power Analysis Attacks* (PAA). The *Simple Power Analysis* (SPA) and *Differential Power Analysis* (DPA). In SPA the attacker monitors the power output to figure out what instructions are executed and what inputs and outputs are used. This requires knowledge of the implementation of the device and is in that regard weaker than DPA since it requires more information to obtain any results which makes the attack harder to mount properly. In contrast, DPA uses statistical analysis and correction of signal errors to acquire the cryptographic keys. By measuring the power consumption of several operations and comparing the results it is possible to draw conclusions about what keys are used and thus does not require implementation details to obtain them [6]. A similar attack to SPA is the *Electromagnetic Analysis Attack* (EAA) which instead reads the electromagnetic radiation instead of power output [7] [8].
A cold boot attack is an attack where the attacker cools down the Random Access Memory (RAM) which is a component that stores processes temporarily in order for them to run much faster. As the processes are stored on the RAM they are vulnerable if attacked by a cold boot attack. If the RAM can be cooled down to around -50°C the temporarily stored programs and information remains on the cooled RAM and can then be put into the attacker’s computer to be read [9]. By conducting this attack, the attacker may additionally be able to obtain encryption keys stored in the RAM.

A Direct Memory Access (DMA) is a feature that allows hardware such as graphic cards direct access to the memory without the requirement of setting up a connection with the processor. A DMA attack is done by manipulating the DMA controller and thereafter read memory regions [10]. The manipulation is done by inserting an illegitimate unit which will use the same connection and port as a trusted component in order to have the rights to read data and install malware.

SCAs are dangerous due to how simple and inexpensive they are relative to other attacks since they do not require large laboratory equipment to conduct. Additionally, SCAs are difficult to counter because some countermeasures against specific SCAs are mutually exclusive to the countermeasures against other SCAs. To simplify, by implementing a resistance against PAAs it might be harder to implement a counter for EAAs.

### 2.1.2 Software attacks

Software attacks utilises weaknesses such as bugs or bad implementation of a device’s cryptographic algorithms to gain information or to make way for other types of attacks by messing with the security system or installing malware. It could be some kind of spyware that sends information to the attacker of what the user is doing on the device or a backdoor giving the attacker a way around the security of the device.

A type of tampering software attack is a rootkit which modifies the operating
system of the device. The rootkit grants root-access (administrator access) to the attacker who then can hide functions from the real admin and make other attacks unseen [11].

Another type of software attacks are attacks against the Application Programming Interface (API). These often consist of sending commands to a device in strange orders to confuse and maybe even break the security system and thus give the attacker unauthorised access to information and functions [12].

2.1.3 Fault generation

Fault generation attacks consists of inducing a fault and utilising that fault to gain access to specific states of operation (ex. an administrator access state) or security keys. Sometimes fault generations are accidental, for example radiation residue in the packaging emitting α-rays that flips bits in a device’s memory. Fault generation attacks are in many cases similar, but are not random or accidental. The similarity does make it hard to distinguish and react to them correctly.

There are several methods of fault generation attacks. Varying things like power supply, temperature, or external clock can induce errors. Targeting vulnerable points with laser or X-rays are also methods that have seen use [13]. Some more specific fault generation attacks are the Differential Fault Analysis (DFA) by which an attacker can extract cryptographic keys by analysing the output of a device after injecting faults [14] and the electromagnetic fault injection which induces faults in a device with high precision [15].

2.1.4 Microprobing

Microprobing is an attack using a focused beam of particles to gain physical access to the surface of the chip in order to tamper with the device or simply observe it as preparation for another attack.

An example of equipment and strategy is to use a Focused Ion Beam (FIB). The FIB workstation is a special scanning microscope. With a gas that can be broken down with the ion beam, it will create the possibility to lay down insulators or conductors with a precision of around 50 nanometres. With the new connections laid out it can read the flow of data and gain access to security keys [5] [16].

2.1.5 Reverse engineering

Reverse engineering is to analyse the device in such a way that the attacker could replicate its functions or at least learn what they are.

One methodology includes the removal of all the layers of the chip in reverse order and photographed to know the internal structure of the chip. Reverse engineering can be a time-consuming process, and usually requires the longest time compared to the other attack techniques. When it is eventually completed, the obtained information can be made into a netlist to simulate the device.
Where a netlist is used to describe all the connections of a circuit [4]. This method has been used in the pay-TV system to create illegal clone cards that decrypts the TV channels without the requirement of monthly payment.

2.2 Invasive, non- and semi-invasive

These attacks are in turn divided into three main categories. Invasive attacks are long term attacks that require expensive laboratory equipment, they ruin the packaging of the device resulting in evidence that the unit has been tampered with. All forms of reverse engineering and microprobing are invasive attacks, the difference is that they leave different degrees of evidence of the tampering. Non-invasive attacks on the other hand are relatively inexpensive, easily updated with new equipment and hard to notice since they leave the device physically unharmed. Side-channel, software, and fault generation are all non-invasive with the exception of some fault generation attacks that are known as semi-invasive. Semi-invasive attacks require access to the surface of a chip while they do not require electrical contact with the device. This kind of attack is powerful since it is effective like the invasive attacks but does not require sophisticated laboratory equipment like the non-invasive attacks [4].
3 Method

In this section we will describe the type of method we used in order to gather our resources. First we describe what method we used to find information. Following that we illustrate our search term tables.

A large part of the project was to find literature and articles ranging from general information to technical evaluations of methods regarding tamper-resistance. In order to make it well outlined, a structured search became vital in order to acquire the required information.

We focused our searching around structured search terms to give a good overview of what we have searched for and what we have seen as the important and interesting areas in the subject.

The method we used is a variant of Rumsey’s search method [17] where we put less focus on the last 3 steps depending on the need or subject. Rumsey’s method is as follows:

The first step is to identify search terms. This is done by looking at the hypothesis, subject or concept of the area of interest and finding synonyms and terms that are connected to the original. This can be structured into tables or spider diagrams. We preferred tables over spider diagrams as they give us a more structured overview that is easier to follow and update. Every column corresponds to the main concept or first identified term. The rows are the different ways of expressing it in different words such as synonyms or more specific implementations. An example would be the concept cars with the synonym auto, broad term vehicle, narrow term Impala 67, and related term Top Gear. It should be noted that some of the tables might not be fully expanded with some cells empty. This is because we didn’t fully explore every concept in the later stages of the project. Including terms we actually did not search for would be misleading.

The second step is limiting the search. Our main limitation is the date of publication. Since technology changes fast, older publications can be outdated even if they are only a few years old. We chose to limit our searches for specific methods to only include articles from 2008 and onwards, but we include some older works if we assess them to be relevant and if they are referenced by newer works. In our report we have no explicit limitation in the type of publication, however we do try to only use peer-reviewed articles in order to have articles at a professional and trusted level. We have also restricted ourselves to the English language because there are no reasonable articles on a language we know.

The third step is truncation, wildcards, and phrases. This means to stemming the terms, using symbols to tell the database that there may be several spellings and limiting a search to a specific phrase. This step is rather specific and can be considered more like a tip than a necessity in the method. It is favourable to keep in mind that some words have several meanings or are parts of other words. However, in computer science the search terms become rather specific and the risk for different meanings is low.

The fourth and fifth steps are both combining terms. The fourth is to use Boolean logic to combine, for example, tamper AND resistance. This is just to
make the search more specific and make it easier to find relevant information. The Boolean combining is the most common and the combining we used the most. The fifth step is focused on text semantics, more specifically connectors. In some databases you can specify that two terms are to be in the same paragraph or sentence in the article by using specific flags.

Once all steps are completed the actual search is carried out and the results are analysed and evaluated. This process is then repeated consecutively. The new term could be one that is found while searching and thus the tables are expanded as the searching goes on throughout the project.

The databases and libraries we used are the ones connected to the Linköping university library. These are for example ScienceDirect, IEEE Xplore and ACM digital library. We could find what we needed either in physical books, in e-books or published articles. We have also used Google Scholar to some extent for cross referencing dates and checking the version of some publications.
Table 1: Search terms 1

<table>
<thead>
<tr>
<th>Concept</th>
<th>Protection</th>
<th>Mobile device</th>
<th>Data</th>
<th>Tamper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>resistance, guarding, attack countermeasure</td>
<td>portable computing</td>
<td>information</td>
<td>meddle, interfere, alter, change, unauthorized alteration</td>
</tr>
<tr>
<td>Broad term</td>
<td>security</td>
<td></td>
<td></td>
<td>portable computing</td>
</tr>
<tr>
<td>Narrow term</td>
<td>data protection, intrusion detection, intrusion prevention</td>
<td>smartphone, tablet, computer</td>
<td></td>
<td>hacking, middle-man, eavesdropping, anti-tamper, tamper tolerant</td>
</tr>
<tr>
<td>Related term</td>
<td>active, passive, physical, software, hardware, encryption, detection</td>
<td>router, base tower, system</td>
<td>database, statistics, data protection</td>
<td>hacking, middle-man, eavesdropping, anti-tamper, tamper tolerant</td>
</tr>
</tbody>
</table>

Table 2: Search terms 2

<table>
<thead>
<tr>
<th>Concept</th>
<th>Hacking</th>
<th>Memory</th>
<th>Software</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>software attack, unauthorized access</td>
<td></td>
<td>program</td>
<td></td>
</tr>
<tr>
<td>Broad term</td>
<td>attack</td>
<td></td>
<td>computer product</td>
<td>electronics</td>
</tr>
<tr>
<td>Narrow term</td>
<td>footprinting, scanning</td>
<td>RAM</td>
<td>java, C++, anti-virus, firewall</td>
<td>harddrive, mobile device</td>
</tr>
<tr>
<td>Related term</td>
<td>intrusion</td>
<td>CD, harddrive, SSD, cloud, hardware</td>
<td>database, statistics, data protection</td>
<td>hacking, middle-man, eavesdropping</td>
</tr>
</tbody>
</table>

Table 3: Search terms 3

<table>
<thead>
<tr>
<th>Concept</th>
<th>Cryptoprocessor</th>
<th>IBM</th>
<th>Smartcard</th>
<th>Smartphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>&quot;High-end physically secure processors&quot;</td>
<td></td>
<td>attacker categories, cryptoprocessor</td>
<td>android, iPhone, google phone</td>
</tr>
<tr>
<td>Broad term</td>
<td></td>
<td>IT company</td>
<td>credit card, debit card, chipcard</td>
<td>cell phone</td>
</tr>
<tr>
<td>Narrow term</td>
<td>IBM 4758</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Related term</td>
<td>tamper circuitry</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Search terms 4

<table>
<thead>
<tr>
<th>Concept</th>
<th>ARM</th>
<th>Invasive</th>
<th>Semi-invasive</th>
<th>Non-invasive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>processor architecture</td>
<td>destructive attack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broad term</td>
<td>physical attack</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Narrow term</td>
<td>microprobing, reverse engineering</td>
<td>fault injection, side-channel</td>
<td>fault injection, side-channel</td>
<td></td>
</tr>
<tr>
<td>Related term</td>
<td>RISC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5: Search terms 5

<table>
<thead>
<tr>
<th>Concept</th>
<th>Epoxy casing</th>
<th>Short circuit mesh</th>
<th>Obfuscation</th>
<th>UV light attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td>shield</td>
<td>conceal, confuse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broad term</td>
<td>physical protection</td>
<td>physical protection</td>
<td>disable security fuses</td>
<td></td>
</tr>
<tr>
<td>Narrow term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Related term</td>
<td></td>
<td></td>
<td></td>
<td>tamper-tolerant</td>
</tr>
</tbody>
</table>

### Table 6: Search terms 6

<table>
<thead>
<tr>
<th>Concept</th>
<th>Optical fault injection</th>
<th>Advanced imaging techniques</th>
<th>Optical side-channel attacks</th>
<th>Fingerprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broad term</td>
<td>induce transient fault with laser</td>
<td>observe rear IR light</td>
<td>observe photon emission on transistor</td>
<td></td>
</tr>
<tr>
<td>Narrow term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Related term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 7: Search terms 7

<table>
<thead>
<tr>
<th>Concept</th>
<th>Watermark</th>
<th>Fusible links</th>
<th>Copier trap</th>
<th>Duplicate computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synonyms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broad term</td>
<td>copyright</td>
<td>wire between sensor and trigger</td>
<td>conceal transistor</td>
<td>reproduce the same result</td>
</tr>
<tr>
<td>Narrow term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Related term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4 Result

In this section we will present methods used as countermeasures against tampering attacks. We will give an overview of the methods, and in Table 8 offer an overview of which methods counter which attacks.

4.1 Potting

Integrated circuits (IC) are commonly protected by some kind of packaging, typically an epoxy or silicon casing (very strong chemical materials when hardened). Potting basically means to put the chip in a ”pot” and filling it up with the protective material (Figure 2). The casing is mainly to protect against dust, corrosion, and vibrations but it does prevent an attacker from easily reading the bus of the IC as well, since it is difficult to cut into the epoxy to get a measuring instrument on the wires. The bus is the communication system between components. To be further resistant to tampering this can be combined with a mesh of wires that short circuit the device if cut. By layering several of these meshes inside the potting it becomes a difficult job to gain access to the device without destroying it in the process. This makes microprobing and reverse engineering a well-potted IC expensive, time consuming and risky even with high precision lasers and drills [5] [18].

![Figure 2: Potting](image)

4.2 Glue logic

In a chipset the Central Processing Unit (CPU), RAM and other components are often easy to see as different ”blocks” on the card. The wires going between them are also easy to see and thus easy to attack. If an attacker wanted to attack a specific part of the chipset it would not be a risk to damage another
part during that attack. *Glue logic* makes it possible to randomise placement of the component and the route of the communication.

Glue logic, as the name implies, can be seen as the glue that makes components work together in an electrical device. It is a digital (as opposed to analogue) circuitry that makes components work closer together primarily as a communication interface designed to be efficient and quick.

Instead of having the components in specific building blocks with easily identified buses for communication, the glue logic design makes it impossible to discern different components in a device and manually attack them as they are built much like an integrated circuit. A simplified image of the two different designs are shown in Figure 3 with the separate blocks to the left and a glue logic block to the right. To reverse engineer a system that uses glue logic requires either sophisticated layout reconstruction software or extensive knowledge of the device [4] [5].

![Figure 3: Glue logic design](image)

### 4.3 Ball grid arrays

Another method of making reverse engineering difficult is hiding the circuitry connections. On normal chip cards, the connections can be followed from one component to another even with the naked eye. The *Ball Grid Array* (BGA) is a way of hiding the connections by making the possible connections so numerous it cannot be guessed.

In short BGA is a sort of casing used on a chip to hide the connection between different chips on a chipset. The capsule has a square grid of lead soldering balls (Figure 4). Once soldered into place in an oven all the balls are connected to the card and hide the actual connections and makes it harder to reverse engineer the chipset since they are difficult to remove without proper equipment [4].
4.4 Copy trap

This method uses the huge amount of transistors that make up the logic functions in the electronic world to its advantage. When an attacker starts to reverse engineer an IC, it is common that they begin by mapping the structure of the unit to achieve a netlist. When an attacker mount a reverse engineer attack, there are a few transistors that are of interest.

This method exploit their idea and lure intruders by placing something that looks like a transistor when it is in fact something else [19]. One such design can be to have a 3-input NOR gate which function only as 2-input NOR gate. This results in a much more time consuming process when trying to reverse engineer the IC.

4.5 Duplicate computation

When an IC makes a computation there is a moment where a bit could be flipped in the memory where the result is stored and then a faulty number is used. The fault could be induced by an attacker with the intention of making the device use the wrong commands to reach a different state.

So when an IC makes a computation the result can be controlled by making the same computation again and comparing the results. This is simply called duplicate computation. Since the chance of inducing a fault at the exact same place is low it is a viable way of detecting faults and handling it whether it is a random fault or a fault injection attack [20].

4.6 Obfuscation

Obfuscation has the purpose of concealing acts. This is a broad method and its methods include bus scrambling, junk code insertions, re-ordering, random dead code, unconditional jumps and suppression of constants. Obfuscation is additionally meant to confuse the attacker when it is in its idle state. It does so by sending dummy data all the time to avoid activity monitoring [21].

Bus scrambling can set up new connection routes when communicating, ensuring that the data is harder to read for an attacker since it could be on any one of the many different connections. The scrambling can be altered after either a
portion of a session or a whole session. In Figure 5 we illustrate the idea of bus scrambling [22].

To guard against software attacks, there are tamper-tolerance algorithms which have the purpose to conceal methods that are being used. A typical algorithm that is used as obfuscation works by breaking up the application that shall be protected into multiple blocks. The blocks are then duplicated and rearranged within the program. It is then possible to add code which should diversify the blocks while still being functionally equivalent compared to the previous version [23]. These steps can be repeated to increase the tamper-resistance further.

![Figure 5: Obfuscation](image)

### 4.7 ARM architecture, TrustZone and SecurCore

The Advanced RISC Machine (ARM) is a line of reduced instruction set computing (RISC) architectures. RISC is basically a design approach to have as few transistors as possible to make the chip smaller and more energy efficient. This results in a harder and more expensive process for attackers to keep up with the progress due to chips not having as much power deviation when performing PAA. Additionally, the decrease in area also affects the analysis because the attacker has less space to inspect.

This makes the small ARM processors ideal for smartphones and tablets and they are used by most manufacturers [24]. It should be noted that the ARM is an architecture and that all licenced companies implement their own version in their products.
ARM Holdings has released a security extension called TrustZone. TrustZone is a concept of making two distinct worlds in a processor, a secure world and non-secure world. The idea is that software that require to be safe is run in the secure world and the rest in the non-secure world. Figure 6 illustrate how ARM TrustZone works on a fundamental level. Every action goes per default into the non-secure world. If an application want to gain access to the secure world, it needs to go through a gatekeeper which verifies if the application is trusted and can therefore have more rights [25]. This concept can be used together with the methods described in Section 4.8 and Section 4.9.

With the use of a secure world and a non-secure world, it is possible to resist against a DMA attack. The resistance is done by letting the secure world have control over the memory that the secrets are stored on [10]. Furthermore, the secure world deny DMA request from the non-secure world.

![ARM TrustZone](image)

Figure 6: ARM TrustZone

### 4.8 Software encryption on system on chip

The system on a chip (SoC) is a widely used term and can simply be interpreted as a chip which integrates components, for instance graphic card, processor, RAM. Together with a software, a tweaked Advanced Encryption Standard (AES) and SoC it is possible to resist against a cold boot attack.

AES is used to encrypt data safely with software whenever the phone goes to the locked state. Additionally, the data is later decrypted when it goes into its awaken state. To be able to do this, the cache must be able to be locked. This is solved by storing the locked state on the cache of the CPU.
When the AES is tweaked and used on SoC it does so without the leakage of secrets to a memory where it can be read by an attacker. This have been a problem with the commonly used AES and have therefore led to it being adjusted [10].

The encryption with the help of software can be made customisable to specifically target particular applications and save their sensitive memory page, such as Google Maps coordinates. By saving the sensitive memory page to the CPU cache, it will insure that there is no data that can be read if a power loss occur. Leading to an outcome where it will be possible to protect the sensitive memory pages.

4.9 Software encryption on internal RAM

Instead of using the method described in Section 4.8, which is not always available, there is internal RAM. Internal RAM (IRAM) is the range of addresses on the RAM that is internal to the CPU. It is often of high speed and it acts comparable to a CPU cache, the difference is that it can always be addressed by software.

This method is preferred if cache locking is not supported or if the usage is focused on a device that require little memory capacity and high speed. IRAM is used to ensure that keys or other secrets can be stored more safely by allocating an amount of memory on the CPU. IRAM is favourable due to its fast speed and that it have no memory remanence, meaning that there is no data left behind when a power loss occur [10].

4.10 Mitigation of SCAs

When it comes to SCAs like PAA there are two main countermeasure paradigms: masking and hiding. Masking means making the power consumption mean average constant and varying the variance with a random element. In contrast hiding simply means making the power consumption constant.

Dual-rail precharge logic (DPL) is a method of hiding the circuit activity from a potential attacker by encoding information with two differential wires (a method of reducing noise) and by dividing the clock period into two phases: precharge and evaluation. Simplified, instead of having low and high voltage signal 0 and 1 they are signalled by a combination of the wires. The technical aspect is less important than the fact that it is a widely used method, that there are several methods based on DPL and that it has been shown to have a weakness against certain PAAs. The method time-enclosed logic (TEL) was proposed as a replacement [26].

TEL is also a hiding method that with a high-frequency filter makes the readable current vary much less and thus be many times harder to draw conclusions about it through spikes and patterns. A more specific type of PAA is the correlation power analysis (CPA) which is the one TEL proved resistant against. Without going into the technical aspects of CPA it is a method of
seeing a correlation between measurements and has been proven several times
to be an efficient attack by extracting security keys [27] [28].

In the article about TEL it was suggested (not tested) that TEL would
be sufficient, or at least equal to earlier methods, security measures against
electromagnetic analysis attacks (EAA) as well [26].

A countermeasure against SCAs that could be argued to be a form of masking
is the Efficient Randomized Instruction inSertion Technique (ERIST). ERIST
is a framework of a chip build on an ARM7 (see Section 4.7 for information
about ARM) and consists of a few modules. The basic idea of ERIST is that
a module generates random, but runnable, instructions that are injected and
executed when the processor works with protected code. Since these random
instructions are actual logical instructions it makes them indiscernible from the
cryptographic instructions in a power sequence reading [29].

4.11 Shielding integrated circuits with sensors

Detecting an attack is also useful when protecting a device from physical attacks.
A burglar alarm in a house only tells a security system that something has
happened, the sensors used in tamper detection work the same way. This gives
the security system a chance to react to the attack. Depending on the needs
of the owner of the device it could trigger a module for erasing data or small
charges destroying the device.

The attacks that sensors shield against can be put in two general categories:
mechanical and optical. Where they can be represented as pressure and photo
detectors respectively. One strength of sensors is that they are small so it is
often possible to use various different kinds of sensors in the same IC.

Sensors can be embedded within an IC in order to detect attacks. The
sensor is commonly linked to an erasure device which erase the confidential
information. However, most ICs uses a battery for its power usage which leads
to a decrease in its ability to stop an attack if the counterpart can remove the
battery. Additionally, a battery’s longevity and size limits the effectiveness in
systems. This can be fixed by replacing the battery with an energy harvesting
device.

The energy harvesting device can function as a sensor to detect fluctuations
and to simultaneously generate power for triggering the erasure device. Two
such devices are Photo-Voltaic (PV) and Nano- and Micro-Electromechanical
Systems (NEMS/MEMS). PV absorbs light and is used together with an anti-
fuse that will erase the content connected to the circuit [30].

Another approach is to use NEMS/MEMS which protects against a mechan-
ical deconstruction of layers of an IC. These deconstructions of circuits involve
mechanical polishing or grinding of the IC. It detects the mechanical force that
is exerted in the process and trigger the same type of erasure device as PV [30].

Some microprobing and fault generation attacks are based around electro-
magnetic methods. Algorithmic and circuit countermeasures are weak against
this type of attacks, but using an EM-sensor they can be detected and averted
[31].
DPL (explained in Section 4.10) can also be used together with sensors as a fault injection countermeasure by making one of the wire combinations of the DPL a signal that can be used by tampering sensors to alarm and reset the device [32] [33].

4.12 Trusted Platform Module

A Trusted Platform Module (TPM) is a cryptographic microcontroller specification developed by the Trusted Computing Group (TCG) to be a support in trusted computing platforms. Trusted computing platform refers to a device that is strict with what hardware can be used and what software can run on it. There are many different implementations of TPMs and they are widely used by various companies as a method of validation and ensuring integrity of systems [34].

The trusted computing platform is basically a device that requires that the software and underlying systems are trusted, i.e. does what is expected in a predictable way. Provided that a device is handling sensitive data it should be trusted in order to be secure.

A TPM has to provide a few basic features defined by the TCG [35]. It should have protected capabilities, integrity measurement and integrity reporting. Simply put, the TPM should have a safe place to store and operate data, a way to measure whether or not a process is trusted and a means to obtain that measurement.

By being trusted itself, the TPM acts as a root of trust which it then uses to start other processes that then are trusted if the measurement is valid. When booting a device that has a TPM the Basic Input/Output System (BIOS) and Operating Systems (OS) are measured so that the booting process is trusted. During the run the TPM refreshes its measurements to see that no unauthorised changes are made during the run and thus prevents malicious external software from executing [34].

4.13 Result overview table

This table illustrates an overview of what our results have reached. The table can be read by identifying an attack in the upper row and thereafter pinpoint the type of method that can be used to resist it in the left column.

The legend at the bottom indicates what the letters in the table refers to. As a clarification we will explain the different markers. With resistance we mean that it hinders an attacker but does not guarantee security. Detection refers to the sensors ability to detect a type of attack. Theoretical resistance means that the method has not been tested against the attack, but is argued to be able to resist it. The star (*) marks a combination of two methods, we only have one such occurance.
Table 8: Result overview table

<table>
<thead>
<tr>
<th>Type</th>
<th>Software</th>
<th>Side Channel</th>
<th>Fault generation</th>
<th>Microprobing</th>
<th>Reverse engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PAA</td>
<td>EAA</td>
<td>Cold Boot</td>
<td>DMA</td>
</tr>
<tr>
<td>Potting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glue logic design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ball grid array</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy trap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Obfuscation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicate computation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM Architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES on SoC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES on IRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R - Resistance  
D - Detection  
(R) - Theoretical resistance  
* - DPL and Sensor needed
5 Discussion

In this section we discuss the method, results and our work in general to give our view of what we have done and what could be worked on in the future.

5.1 Method

We were early influenced by Ross Andersons (author of Security Engineering [5]) view on the subject. This may have made us a bit too focused on his idea of tamper-resistance. While he is a leading expert on the subject, other views are important to include. Even though they were all similar we did find a multitude of attack classifications, attacker types and countermeasure categories during our research. Some left out things that others did not. Since Ross Anderson and Sergei Skorobogatov both used the same categories and are experts cited by many, leading to the use of their definitions. A similar review could probably be made but with a different base to stand on.

The method used to find our results is not a complicated one. It is mostly there to make our searching easier to follow and for us to structure it in a way so we do not use the same search terms repeatedly throughout the project. Using the same search terms as we did, similar articles should be found provided that the same search engine is used. We realised that we could not only use peer-reviewed articles for every method. Some methods were referenced in several places but not explained. When we then found an article that did explain the method we used it whether it was peer-reviewed or not. We also included some white-papers for company specific methods that were standardised.

The database search engine UniSearch at Linköping University was a great tool for finding resources, however it could have been a great idea to take a step back from it and use something completely different a bit more. Even though most relevant article databases like IEEE and ScienceDirect are linked to UniSearch there might be articles relevant for our review that we missed due to not specifically searching for other databases and search motors other than the ones provided to us by the university.

The biggest delimitation of our work was the decision to not include attacks during production or methods to control authenticity of a device. Watermarking and fingerprinting, self explained by the names, are both such countermeasures and we scrapped them to avoid making our review too broad. This does give a possible follow-up of this review since we have not looked into the possibility of hardware trojans and other production line based attacks that are also directed at hardware.

Some sources were not available to us due to not being free. We made one order for a chapter out of a book from the university library service, but otherwise we have only used free resources. If this is not an issue there might be some methods we have not been able to find enough material to include. One such example we know of is elliptic curve cryptography that was referenced in some places, but rarely explained in a way we could understand it or referenced in a satisfactory way.
5.2 Results

As in most security solutions, nothing is completely guaranteed to be safe just because some kind of security mechanism is implemented. This is why tamper-resistance is a term used more than tamper-protection since the methods presented mostly make it harder to attack, not theoretically impossible. It is, as always, a question of time and resources. With enough of both any defence can be broken. The goal is to make this process slow and costly enough to make the result of the attack less valuable compared to the invested effort. If a product is broken after four years of attacking but is replaced after three it is a victory for the manufacturer.

Our result is not a guide of which defences should or should not be implemented on a device. Depending on a specific device’s situation and needs some tamper-resistance might already be implemented following a design choice or unavailable due to some technical requirements. Some methods are even mutually exclusive and makes every attack and countermeasure a research area by themselves. This review intends to give the reader a basic overview of some common methods and the mentality when building a tamper-resistant device. Further reading and researching in the specific area of interest is encouraged if the intention is to build a tamper-resistant device.

Even though we do define tampering attacks as either changing data or accessing it without authorisation it still is still debatable what is a tampering attack and what is not. The SCAs are the ones that stood out to us since they do not really appear like tampering at all but rather just reading output. Tampering implies changing something while the SCAs give the impression of eavesdropping. Almost every single article and book about tampering attacks bring up SCAs along with microprobing and reverse engineering which are more direct tampering attacks. This is where the semi-invasive attacks come into the picture. Some of the SCAs do require some physical tampering before they can be conducted which makes them a sort of tampering attacks. Also, looking back at our definition it definitely is accessing data without authorisation. So we think SCAs as a whole category does fit into our scope even though some of them require no physical alteration of the device at all.

We wanted to be as general as possible and not use company specific implementations of methods in our results. But in some cases the standard has been set by a company and everyone uses that specific solution and it would then have been tedious to not use it as a means of explanation.

5.3 The work in a wider context

Security is a hot topic in both media and politics. As technology grows and takes more place in society so does the risks. Pirating, theft, counterfeiting and privacy intrusions are just a few of the problems that have become digital in the last decades. In the branch of hardware these are not small problems either. How to store certain information is sensitive and certain numbers are even illegal to know and share due to being used in some security solutions, for example
the encryption keys of DVDs and Blu-rays. Some types of secure hardware are even illegal to use in certain countries due to the governments laws requiring the ability to read information stored on all computers. As always in security, it becomes a question of how ethical it is to retain something that no one can read even though there may be lives at stake. The situation between Apple and the FBI is a prime example.
6 Conclusion

We have given a broad overview on some tamper-resistance methods and what specific attacks they may withstand. There is a multitude of different ways to make a unit more tamper-resistant and a few of them that have only changed slightly since the 1990s. Most methods have roots in older versions with many similarities, but incremental changes that outdate the older ones. So even though most methods are old, the area is very fast paced.

To answer the questions we posed in Section 1.1 we have first found a number of methods used to make devices resistant against attacks and made a summery of what their goals are as our result. We have also found a few specific attacks that the methods are designed to be resistant against and show this in the table.

Our results show that there are a multitude of different approaches to almost every security threat. And since the attack categories have not changed much in the last decades it is probably safe to say that there will not be any groundbreaking new ways of tampering with a device in the near future either.

We also found that some of the tamper-resistance methods we presented are not applicable on all platforms or situations due to the practicality of their implementations. It is as usual a question of necessities.

Several companies have specific tamper-resistant processors as a separate family. These so called cryptoprocessors use several different methods to counter as many threats as possible and much work and research is put into their production. And we can conclude that most of the methods we have presented are used by such processors in one form or another.
References


(visited on 19/04/2016).


Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-komersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

Copyright

The publishers will keep this document online on the Internet – or its possible replacement – from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/her own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.

© Pontus Hero Ek and Villiam Rydfalk