Reverse Engineering of Microcontrollers

Proseminar Microcontrollers and Embedded Systems WS14/15

Fabian Weise
Chair for Robotics and Embedded Systems
Department of Informatics
Technische Universität München
Email: fabian.weise@tum.de

Abstract

In this paper the different techniques of reverse engineering and hardware analysis are explained through the example of Atmel's ATtiny13 microcontroller. Furthermore its protection against vulnerabilities is analyzed, how its fuse bit mechanism can be bypassed and a possibility to defend against such attacks on the software side. Finally, the legal regulations concerning reverse engineering of products are explained.

I. Introduction and motivation

Nowadays, semiconductor chips have become integral parts of our modern living. They are needed when making a call, watching TV, washing clothes and many other use cases.

As the industry began to use silicon chips not only for control purposes but for protection as well, the demand for security has been growing proportionally. Especially since the Edward Snowden Affair the field of IT-security is booming (see Stiennon, 2013 [1]).

The technology moved to everyday life from the military and banking sector: In order to prevent the use of unbranded batteries in smartphones and notebooks, to restrict the servicing of appliances to manufacturer service centres, and to block non-genuine and refilled cartridges for printers (see Skorobogatov, 2005 [2]).

Since manufacturers invent new security solutions for their hardware, the hacker community is also constantly trying to break their protection. In terms of the technological process, reverse engineering has been helping to design compatible products and improve existing ones. But everytime someone tries to steal property, the demand for security implicitly increases.

Therefore both parties consistently gain new knowledge and experience, which not only leads to shifting the front line back and forth regularly, but also to affecting both economics and law. Concomitant, distinguishing reverse engineering from piracy is difficult in the majority of cases (see Skorobogatov, 2005 [2]).

This paper covers the general techniques of the reverse engineering of microcontrollers. As such research is an endless process due to continuous technological progress, I had to choose a narrow area of security analysis in microcontrollers through the example of the ATtiny13.

II. Techniques and tools

A. Toolset

Hardware security analysis is a field in computer science which requires a comprehensive knowledge in a variety of areas.

Apriori, one needs to be able to write test programs in assembler and C, and to use development and debugging tools, universal programmers and test equipment like signal generators, oscilloscopes and logic analysers.

Performing simple attacks then involves using special electronic tools to manipulate the signals applied to the chip. That also requires building interface boards and writing programs on a PC to control them.

More powerful and complex attacks require direct access to the chip surface. For that, knowledge and experience in chemistry, as well as the use of a microscope and mircoprobing station is needed.

Once the access to the chip layout is ensured, some basic knowledge of silicon design from microelectronics is necessary. For most of the information concerning the ATtiny13's design, the interested reader is referred to the documentation published by Atmel itself (see Atmel, 2010 [3]).

Finally, a broad knowledge of physics is needed throughout many of the experiments.

B. Attack categories

In the reality, a potential attacker aims to recover security algorithms and crypto key material stored in a microcontroller. So a successful attack could possibly result in the ability to create own valid credit cards, for instance if the target would be a card reader.

Hardware, that handles such kind of high confidential information usually has to satisfy the security requirements for cryptography modules of the Federal Information Processing Standard 140-2, that is a security standard. For its detailed content, the interested reader is referred to the National Institute of Standards and Technology, 1994 [4].

Throughout this paper we presume, that for such an attacker it is possible to obtain an arbitrary amount of examples of target devices, notwithstanding that in the reality it is not.

We can distinguish four major attack categories (see Wagner, 1999 [5]):

Microprobing techniques can be used to access the chip surface directly in order to observe, manipulate, and interfere with the integrated circuit. It lets someone understand the inner structure of the semiconductor chip and study its functionality.

Software attacks are based on the processor's normal communication interface and exploit security vulnerabilities found in the protocols, cryptographic algorithms or their implementation. An example would be the exploitation a function in order to provoke a Buffer Overflow and read usually not accessible memory.

Using **Eavesdropping** techniques the attacker is able to monitor the analog characteristics of supply, interface connections and any electromagnetic radiation by the processor during normal operation with high time resolution.

Fault generation techniques use abnormal environmental conditions to generate malfunctions in the processor that provide additional access, for instance heat, UV-, photon- or electron-irradiation.

All microprobing techniques are invasive attacks. They require hours or weeks in a specialised laboratory and in the process they destroy the packaging. The other three are non-invasive attacks. The attacked device is not physically harmed during these attacks. The last attack category could also be called semi-invasive. It means that the access to the chip's die is required but the attack is not penetrative and the fault is generated with intensive light pulse, radiation, local heating or other means (see Skorobogatov, 2005 [2]).

Non-invasive attacks are particularly dangerous in some applications for two reasons. Firstly, the owner of the device might not notice that the secret keys or data have been stolen, therefore it is unlikely that the validity of the compromised keys will be revoked before they are abused. Secondly, non-invasive attacks often scale well, as the necessary equipment can usually be reproduced and updated at low cost (see Anderson, 1997 [6]).

The design of most non-invasive attacks requires detailed knowledge of both the processor and software. On the other hand, invasive microprobing attacks require very little initial knowledge and usually work with a similar set of techniques on a wide range of products. Attacks therefore often start with invasive reverse engineering, the results then help to develop cheaper and faster non-invasive attacks. Semi-invasive attacks can be used to learn the device functionality and test its security circuits. As these attacks do not require establishing any physical contact to the internal chip layers, expensive equipment such as laser cutters and focused ion beam machines are not necessary. The attacker could succeed using a simple off-the-shelf microscope with a photoflash or laser pointer attached to it (see Skorobogatov, 2005 [2]).

Attacks are reversible when the device can be put back into the initial state, or irreversible with permanent changes done to the device. For example, power analysis and microprobing could give the attacker a result without harming the device itself. Certainly microprobing will leave tamper evidence but usually that does not affect further device operation.

On the contrary, fault injection and UV light attacks could very likely put the device into the state where the internal registers or memory contents are changed and cannot be restored. In addition, UV attacks leave tamper evidence as they require direct access to the chip surface (see Anderson, 1997 [6]).

III. Hardware Analysis

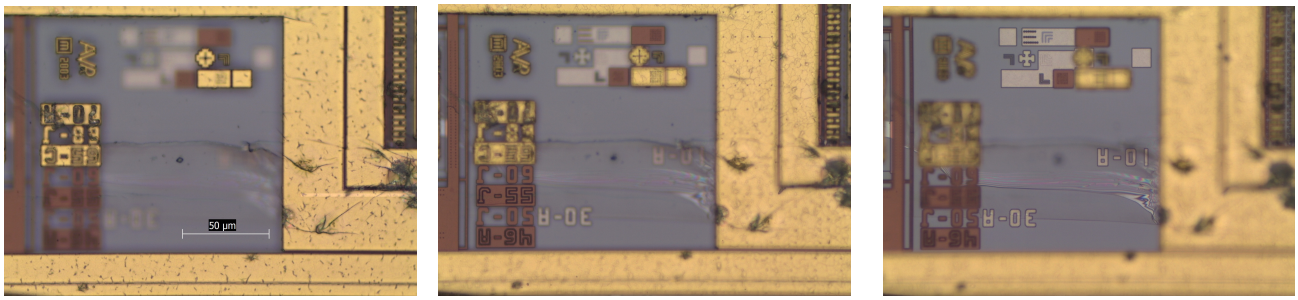
To analyse a microchip one needs to get further information about its actual design. This can be obtained by removing the epoxy packaging of the microchip applying different chemicals. Especially if there are only a few chips available, one is dramatically limited in the amount of attempts (see Beck, 1997 [7]).

Concerning the ATtiny13, I bored a hole in its epoxy packaging with a CNC machine, just as deep as close before exposing the bond wires. Then inside a cleanroom the remaining polyepoxides were removed by dripping 92% fuming nitric acid (HNO_3) into it. In the later process I removed the top-layer using nitro-hydrochloric acid (HNO_3+ 3 HCl).

Contemporary chip design processes also include multiple patterning, that is a lithographical technique that uses multiple metal layers on a microchip in order to enhance its feature density. This is particularly necessary for producing recent 14nm and smaller transistors (see Skorobogatov, 2005 [2]) .

It is noteworthy, that the complexity of refactoring dramatically scales with the number of layers on a chip, because it becomes much more difficult to separate each layer accordingly.

For instance, the modern IBM Power 8 consists of 15 metal layers, whereas the ATtiny13 consists of the three ones shown below.

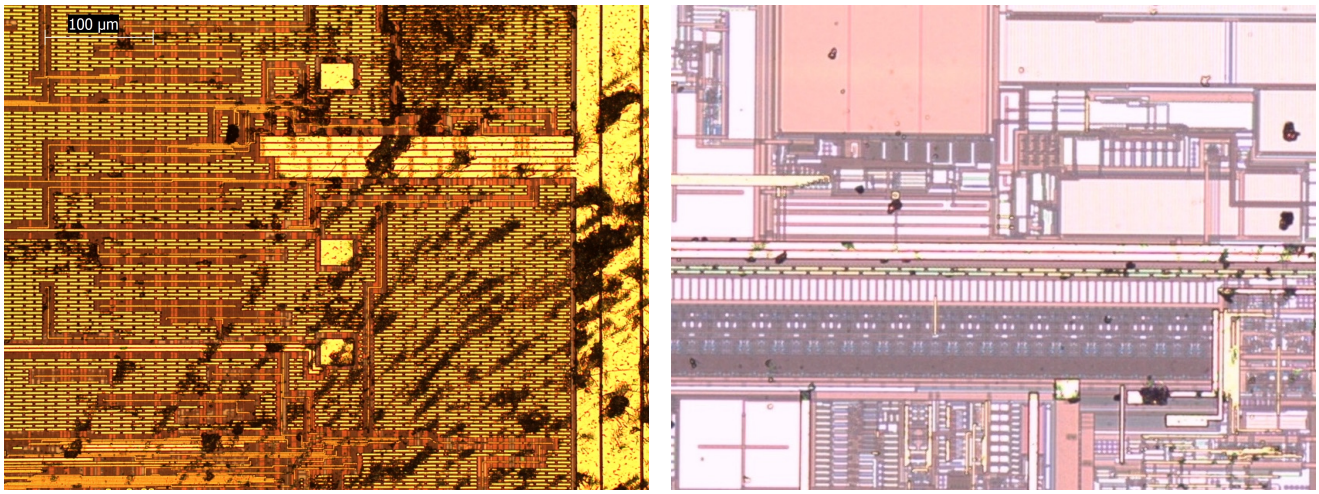


(a) Top layer (b) Middle Layer (c) Bottom layer

Fig. 1

The further hardware analysis process shows a standard method to secure the inner logic through obscurity by Atmel: The top metal layer consists of a regular pattern, that does not provide any functionality to the chip, in order to only hold a potential attacker from further refactoring.

Since one separate layer is very thin, it can be removed putting the die into a bath of nitro-hydrochloric acid for a few minutes, so its actual logic gets exposed.



(a) A fake top metal layer pattern on microchip ATtiny13 microcontroller makes the analysis of the chip die and microprobing attacks more difficult. 200x magnification

(b) A section after removing the top metal layer pattern on microchip ATtiny13 microcontroller to reverse engineer the inner logic. 500x magnification

Fig. 2

If the chip's inner logic is accessible one can to refactor its hole functionality, presuming to not be restricted in the amount of time. Moreover it is possible to inject faults, e.g. in order to break a read protection and access a specific memory (see Skorobogatov, 2005 [2]).

IV. Hardware vulnerabilities

One of the oldest attacks on microcontrollers known, are the UV light attacks, which were released in the middle seventies. They belong to the semi-invasive attacks, because only the chip's packaging has to be removed. This attack can be applied to many microcontrollers using EPROM, since their protection is only designed to withstand low cost non-invasive attacks (see Kume et al., 1987 [8]).

This type of attack can be devided into two stages: Finding the security fuse and resetting it to the unprotected state with a UV light source. The security fuse is usually designed, such that it cannot be erased earlier than the program memory. That is, one cannot apply the UV light to the whole chip. Therefore one can either selectively apply the UV light to the fuse by using a microscope or a UV laser, or cover the memory with opaque material to protect it from being erased (see Kume et al., 1987 [8]).

In order to protect EPROM memory against attackers, the protection mechanisms were improved. Not only was a top metal layer added in order to block the UV light, but also inverted memory cells were used. That is, a set fuse bit corresponds to a non set bit and vice versa.
Also the ATtiny13 benefits from both the top metal protection and inverted fuses, so that the security fuses cannot simply be reset with UV light, because, as aforementioned, the erased fuse corresponds to active security.

However, as the UV light can change the fuse from the non-secure state to secure, it can still be used to find the fuse (see Messerges, 1999 [9]).

Then with a micro probing station one can manipulate the chip's behaviour by injecting a fault and gain access to the chip's data.

In order to find the ATtiny13's fuse bits, I successively covered parts of the chip with opaque material and irradiated it with UV light. Then, if the data was still readable, I recovered its origin state by rewriting a written program. In case the data could not be read, because of the set protection state after irradiation, the fuse bits were found.

A. Fault injection

Now, that the fuse bits are found, one can perform probing or modification attacks with a microprobing workstation. Its major component is a special optical microscope with a long working distance objective lens. The microscope usually has several more objectives to accommodate different magnification and depths of focus, that is the more important the more metal layers the chip consists of. Additionally, several micropositioners are installed on a stable platform around the chip test socket in order to allow the probe arms' movement with submicron precision over the chip surface. At the end, a probing tip with an elastic hair is installed on each arm, allowing electrical contact to on-chip bus lines without damaging them (see Kocher et al., 1999 [10], Skorobogatov, 2005 [2]).

Furthermore, the probe tip can be either passive or active. The passive tip has low impedance and high capacitance and is used for both eavesdropping and injecting signals. It cannot be used for probing internal signals on the chip, whereas an active tip offers a high bandwidth with low loading capacitance (see Kocher et al., 1999 [10]).

Finally, I could break the ATtiny13's protection, applying a passive probe tip in order to manipulate the already found fuse bits. It is noteworthy, the chip still worked even though the top layer was completely removed.

V. Discussion

For applications or devices that include cryptography, U.S. and Canadian federal government agencies are required to use a cryptographic products that have been FIPS 140 (Federal Information Processing Standards) validated or Common Criteria validated. Most Common Criteria protection profiles rely on FIPS validation for cryptographic security. Within the FIPS 140-2 validations, there are four possible security levels for which a product may receive validation. Even companies of other nations are oriented towards these standards, since they are the highest known (see National Institute of Standards and Technology, 1994 [4], Kelsey, 2000 [11]).

In our experiment, we have shown how to break the protection of an ATtiny13 microcontroller, but under real conditions it is much less likely to perform such an attack successfully. The reasons for it are, one is restricted by the amount of chips, other security mechanisms and the equipment; i.e. there are light or temperature sensors, that measure irregularities and in case erase all the chips data immediately. Hence, attackers try out non-invasive methods first and apply invasive ones just in case of failure, since the chip's state will then become irreversible.

A. Possible defenses

Not only is it the manufacturer, who is able to ensure security, but also the customer. For instance, by using standard C functions it is possible to exploit a program through provoking a buffer overflow. One example are calls of functions, that expect a certain string with an according length; they should be wrapped into conditions to check whether the string really has the length to be expected and otherwise throw an exception.

Another example is to check whether a certain value has been corrupted. A simple way is to compare it with a checksum:

```
byte result = someValue;
byte resultChecksum = BITWISE_NOT(someValue);
if ((BITWISE_XOR(result, resultChecksum)) != 0xFF) fail();
```

The crucial fact is, that also the software has to be robust in order to be able to protect against possible attackers. Furthermore, it is much more expensive for an attacker to perform a hardware than a software attack, because of the needed tools. Hence, the number of theoretically successful attackers shrinks proportionally if the system is software secure (see Skorobogatov, 2005 [2])

VI. Legality

In the majority of cases it is difficult to distinguish the terms legal and illegal regarding the reverse engineering of a soft- or hardware product. It is noteworthy, that the analysis and reverse engineering of a competing car or a weapon is never legally challenged, nor was reverse engineering software a few decades ago. The crucial fact is that the software or hardware is not subjected to critical scrutiny (see Samuelson and Scotchmer, 2002 [12]).

Moreover it is null and void to prohibit the analysis of an owning product, because it is the user's right to validate the application's security, i.e. in case of a trojan horse; often those terms to act as a deterrent.

Nowadays it is usual to use encryption and obscurity methods in the field of computer technology. One aim is to make the reverse engineering process more difficult, but the more important is that the amanded

copyright and related laws have an effect regarding its copy protection and treats of punishement. That is, breaking a copy protection is prohibited by law, but exceptions exist i.e. in order to archive digital products and therefore the necessary techniques like reverse engineering (see Library of Congress, 2001 [13]).

Especially in the EU, the Art. 6 of the 1991 EU Computer Programs Directive governs reverse engineering in the European Union. It states that in general, "unauthorised reproduction, translation, adaptation or transformation" is unlawful (see European Council [14]). A similar exemption exists for reverse engineering as in the U.S., when this is performed for interoperability purposes, but the law prohibits use of the knowledge gained, in a way that prejudices the right holder's position or legitimate interests, i.e. using reverse engineering to create a competing product. It also prohibits the public release of information obtained through reverse engineering of software.

Therefore one also does have to discuss industrial property rights, such as patents and related rights.

A. Patent Law

A patent is a set of exclusive rights granted by a sovereign state to an inventor or assignee for a limited period of time in exchange for detailed public disclosure of an invention. An invention is a solution to a specific technological problem and is a product or a process. Because patents are a form of intellectual property, the exclusive right of owning them is the right to prevent others from using it. That is, commercially making, using, selling, importing, or distributing a patented invention without permission (see Bundesrepublik Deutschland, 2013 [15]).

Hence, just because the reverse engineering process itself is not culpable, one does not grant the rights to make use of its information, for instance, to build competing products using a competitor's patented technology.

B. End User License Agreements

In software products, the End User License Agreement is the contract between the licensor and purchaser. It establishes the right of the purchaser to use the product. The license may define ways under which the copy can be used, additionally to the general terms and conditions of sale (see Rössler, 2012 [16]).

VII. Conclusions

The aim of this paper was to highlight some potential problems of hardware security in microcontrollers, and give an introduction to various attack methods and possible protections against such attacks.

Our introduction to attack technologies and tools included already known non-invasive, invasive and semi-invasive attacks, such as UV irradiation. Like invasive attacks, they require the depackaging of the chip to get access to its surface, but the passivation layer of the chip remains intact as these methods do not require electrical contact to internal metal wires.

In future prospects, new semi-invasive attacks are becoming more attractive as they do not require very expensive tools and give results more quickly. This is especially important, because with the technological progress, invasive attacks are becoming constantly more demanding and expensive, with shrinking feature sizes and increasing device complexity. Being applied to a whole transistor or even a group of transistors, semi-invasive attacks are less sensitive to the small feature size of modern chips.

Semi-invasive attacks are not entirely new. The old EPROM-hacking trick of exposing the microcontroller's fuse bit to UV light requires depackaging it. Semi-invasive attacks could in theory be performed using such tools as UV light, X-rays, lasers, electromagnetic fields and local heating. They could be used individually or in conjunction with each other.

REFERENCES

[1] S. Richard, "It security industry to expand tenfold," *Forbes*, pp. 1–2, 2013.

[2] S. P. Skorobogatov, "Semi-invasive attacks: A new approach to hardware security analysis," *University of Cambridge Computer Laboratory*, 2005.

[3] ATMEL, "Attiny13," 2010.

[4] N. I. of Standards and Technology, "Security requirements for cryptographic modules," *FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION*, 1994.

[5] L. C. Wagner, "Failure analysis of integrated circuits: Tools and techniques," *Kluwer Academic Publishers*, 1999.

[6] M. G. K. Ross J. Anderson, "Low cost attacks on tamper resistant devices," *M.Lomas et al. (ed.), Security Protocols*, 1997.

[7] F. Beck, "Integrated circuits failure analysis: A guide to preparation techniques," *John Wiley and Sons*, 1997.

[8] T. H. T. T. H. Kume et al., S. Meguro, "A flash-erase eeprom cell with an asymmetric source and drain structure," *IEEE IEDM Technical Digest*, 1987.

[9] R. H. S. Thomas S. Messerges, Ezzy A. Dabbish, "Investigations of power analysis attacks on smartcards," 1999.

[10] B. J. Paul Kocher, Joshua Jaffe, "Differential power analysis," *CRYPTO99, LNCS, Vol. 1666, Springer-Verlag, pp. 388397*, 1999.

[11] D. W. C. H. John Kelsey, Bruce Schneier, "Side channel cryptanalysis of product ciphers," *Journal of Computer Security, Vol. 8(23), 2000, pp. 141158*, 2000.

[12] P. Samuelson and S. Scotchmer, "The law and economics of reverse engineering," *111 Yale L.J. 1575*, 2002.

[13] L. O. CONGRESS, "Exemption to prohibition on circumvention of copyright protection systems for access control technologies," *Copyright Office 37 CFR Part 201*, 2002.

[14] E. Council, "Council directive 91/250/eec of 14 may 1991 on the legal protection of computer programs," *Official Journal L 122 , 17/05/1991 P. 0042 - 0046*, 1991.

[15] B. Deutschland, "Patentgesetz in der fassung der bekanntmachung vom 16. dezember 1980 (bgbl. 1981 i s. 1), das zuletzt durch artikel 1 des gesetzes vom 19. oktober 2013 (bgbl. i s. 3830) gendert worden ist," 2013.

[16] M. Rössler, "Lizenzvertragsrecht," *Skriptum, RWTH Aachen Universitt*, 2012.