Fraunhofer
AISEC

FRAUNHOFER RESEARCH INSTITUTION FOR
APPLIED AND INTEGRATED SECURITY

PROTECTING EMBEDDED SYSTEMS AGAINST PRODUCT PIRACY

TECHNOLOGICAL BACKGROUND AND PREVENTIVE MEASURES

BARTOL FILIPOVIĆ, OLIVER SCHIMMEL

Fraunhofer
AISEC

Protecting Embedded Systems Against Product Piracy

Technological Background and Preventive Measures

Bartol Filipović, Oliver Schimmel

April 2012

Version 1.1 (eng)

Fraunhofer Research Institution for Applied and Integrated Security
www.aisec.fraunhofer.de

Fraunhofer AISEC
Parkring 4
85748 Garching (near Munich)
Germany

Contact: bartol.filipovic@aisec.fraunhofer.de

Contents

List of Figures

Picture credits

Cover picture: fotolia/Martina Berg

Figure 1.1a (Seite 10): iStock-Foto/Aleksandar Ljesic
Figure 1.1b (Seite 10): iStock-Foto/Vladimir Ovchinnikov
Figure 1.1c (Seite 10): iStock-Foto/Alexander Shirokov
Figure 1.1d (Seite 10): iStock-Foto/Alexey Dudoladov
Figure 1.1e (Seite 10): Apple
Figure 1.1f (Seite 10): iStock-Foto/Jill Fromer

Figure 1.2 (page 11): Cfaerber, Wikimedia Commons, licensed by Creative Commons-License namen-transfer under same terms 3.0 Unported (CC BY-SA 3.0), URL: http://creativecommons.org/licenses/by-sa/3.0/deed.de. The original file can be found at http://upload.wikimedia.org/wikipedia/commons/d/d1/Geistiges_Eigentum_und_Wettbewerbsrecht.png

All other figures © authors and Fraunhofer AISEC

Abstract

The negative impact of product piracy is a threat that needs to be taken seriously. It affects not only original equipment manufacturers but also defrauded consumers and damaged economies. Manufacturers lose market shares and suffer damage to their images. Consumers (unwittingly) use substandard products of questionable safety and reliability. Declining investments, job cutbacks and tax losses ultimately even weaken economies.

Embedded systems are an integral part of many modern investment and consumer goods. Unless preventive technological protection is provided, sophisticated technologies permit attacks to be made on hardware and software in embedded systems. The attacks range from targeted modification to complete reverse engineering and product piracy. In this report, we discuss the ways in which attacks can occur as well as the protection measures that are taken to fight product piracy through technological means.

Fraunhofer AISEC develops technological protection measures for fighting product piracy and protecting business assets on the basis of the latest scientific findings. Our application-oriented security specialists have many years of project experience and proven expertise in the field. Our offer includes, among other things:

- A modern lab for hardware security analyses and system evaluations
- Product-specific security solutions
- Hardware- and software-based protection measures and barriers to imitation (protection against manipulation, reverse engineering and product piracy)
- Identification of components and spare parts
- Design security
- Practicable implementation of modern encryption techniques

1 Product and know-how piracy

The piracy of products, components and designs causes enormous economic damage that is constantly reaching new record-breaking levels. The victims are predominantly the companies whose products are being pirated and copied. Copied goods are then sold under conditions that distort competition and markets. The negative effects extend from the loss of market shares and damage to a company's image to job losses. As a direct consequence, this also places an immediate burden on the economy. However, negative consequences also affect consumers—for example, if a substandard counterfeit of questionable safety and reliability is passed off onto an unsuspecting buyer who thinks he is purchasing a brand-name product.

It is estimated that product piracy affects 10% of the global trade in goods, and the economic damage runs to as much as 300 billion euros [18]. In a study published in 2011, the International Chamber of Commerce (ICC) put the full scope of the shadow economy for counterfeits and pirated copies in the G20 countries at as much as a total of 650 billion U.S. dollars in 2008 [3]. Pirated copies of music, movies and software accounts for a share of this figure amounting to 30 to 75 billion U.S. dollars. Negative macroeconomic effects cause damage valued at around 125 billion U.S. dollars, including such problems as tax losses, higher costs of criminal prosecution and loss of foreign investments. In addition, more than 2.5 million jobs have been lost, according to the study. In a study published in 2010, the annual loss in the German engineering sector amounts to an estimated 6.4 billion euros, according to the Verband Deutscher Maschinen- und Anlagenbau (German Engineering Federation) [16]. And the trend is growing.

The range of products affected by piracy is enormous. The problem impacts both consumer and investment goods. Examples of imitations can be found in the textile and clothing industry, digital media (pirated copies of music, movies and software), mechanical assemblies and even electronics (both individual components and entire systems). Modern reverse engineering and rapid prototyping processes make sophisticated methods and tools available for system analysis, manipulation and product cloning [4, 13, 15].

The tools available for protecting products against piracy are as varied as the products affected. To fight piracy, however, an obvious step is to use protection measures that are aimed at securing a company's own core know-how. Effective protection against plagiarism means that companies must take precautions, and a primary focus in protecting embedded systems should be placed on technology. Suitable measures can help counteract reverse engineering or the unwanted manipulation of electronics or software.

1.1 Definitions

Product piracy can be broken down into different categories, which is why a definition of the term is provided below along with a discussion of the systems addressed in this article.

1.1.1 Product piracy

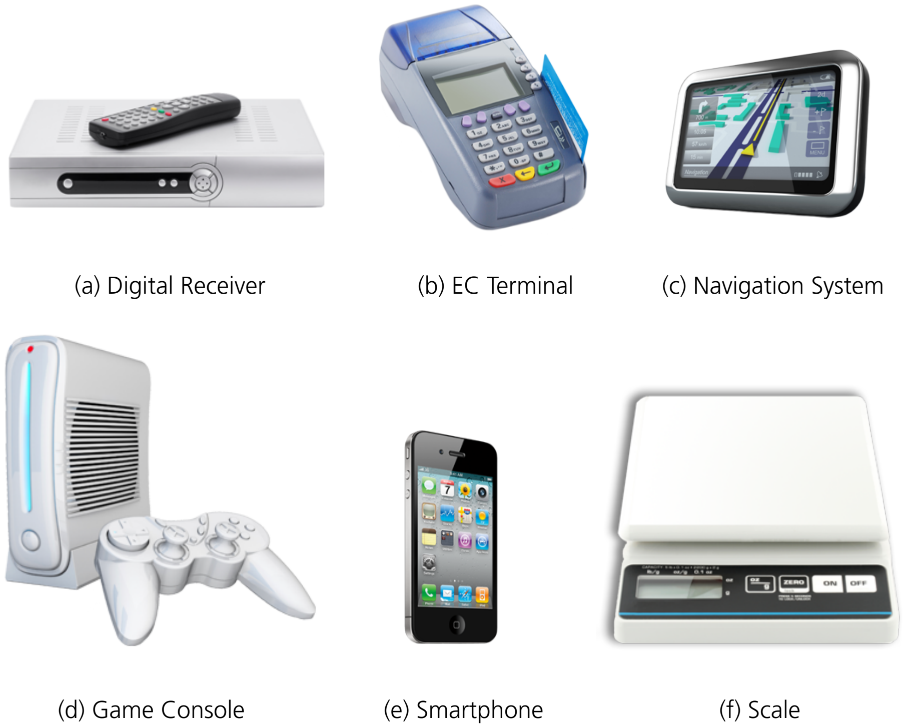
In everyday speech, the term product piracy is used synonymously with other designations, such as product counterfeiting, plagiarism, product imitation and brand piracy. The definition according to [12] makes an initial distinction between imitations and originals: a) An imitation is created after the original; b) the imitation has a similar application functionality to the original from the customer's point of view; c) the imitation is based on the same or a very similar technology as the original; and d) the imitation is created on the basis of an illegitimate use of someone else's technological know-how. Imitations typically copy certain features of an original product in part or in their entirety. According to [12] the two classes, plagiarism and counterfeiting, form special types of imitations. Plagiarism involves passing off someone else's intellectual property as one's own creation. Counterfeiting unlawfully passes off another person's creation as one's own product. The term product piracy covers both counterfeiting and plagiarism. According to [18], product piracy is the forbidden copying and reproduction of goods for which the lawful manufacturer holds intellectual property rights, design rights and process rights.

1.1.2 Embedded systems

An embedded system is an electronic system (computer) that is embedded in a specific application-oriented context. Common functions are measurement, control, regulation, data monitoring and signal processing. The underlying computer system is highly specialized: Both the hardware and software (firmware) are optimized for a specific application scenario. Typical conditions are: minimal costs, limited use of space, energy and memory and long-term usability. The ability to process data in real time is often another important requirement. Examples of embedded systems are shown in Figure 1.1.

1.2 Fighting product piracy

Original equipment manufacturers can take both organizational and legal steps as well as technological ones to protect their interests and fight product piracy.



(a) Digital Receiver

(b) EC Terminal

(c) Navigation System

(d) Game Console

(e) Smartphone

(f) Scale

Figure 1.1: Examples of embedded systems

Protection measures have either an undifferentiated effect (applying to all of a company's products and know-how constituents) or a differentiated effect (applying to only selected product groups and know-how constituents). Strategic measures are investigated in [12]. Von Welser and González [17] discuss legal and organizational measures to fight product piracy. Experience with the practical application of protection measures are described in [9, 2, 1].

The organizational measures include, for example, selecting production facilities as well as access to regulated and secured know-how and managing the chain of supply and innovation.

Legal action includes applying and enforcing legal protection of industrial property (e.g., patents, trademark rights). Note that the company goals must be consistent with competition and antitrust laws. This means that any interests of consumers and competitors that may conflict with the company's own interests are protected under the law: Free market competition is desirable and laws are in place to prevent the formation of monopolies. Figure 1.2 shows an overview of the relevant terminology relating to intellectual property and competition law.

A wide range of technological measures are possible which can take effect on different levels. On the one hand, a company's internal communication infrastructure can be protected against industrial espionage in order to prevent unwanted transfer of valuable company know-how. In addition, companies can use suitable IT security tools (such as intrusion prevention solutions or data leakage preven-

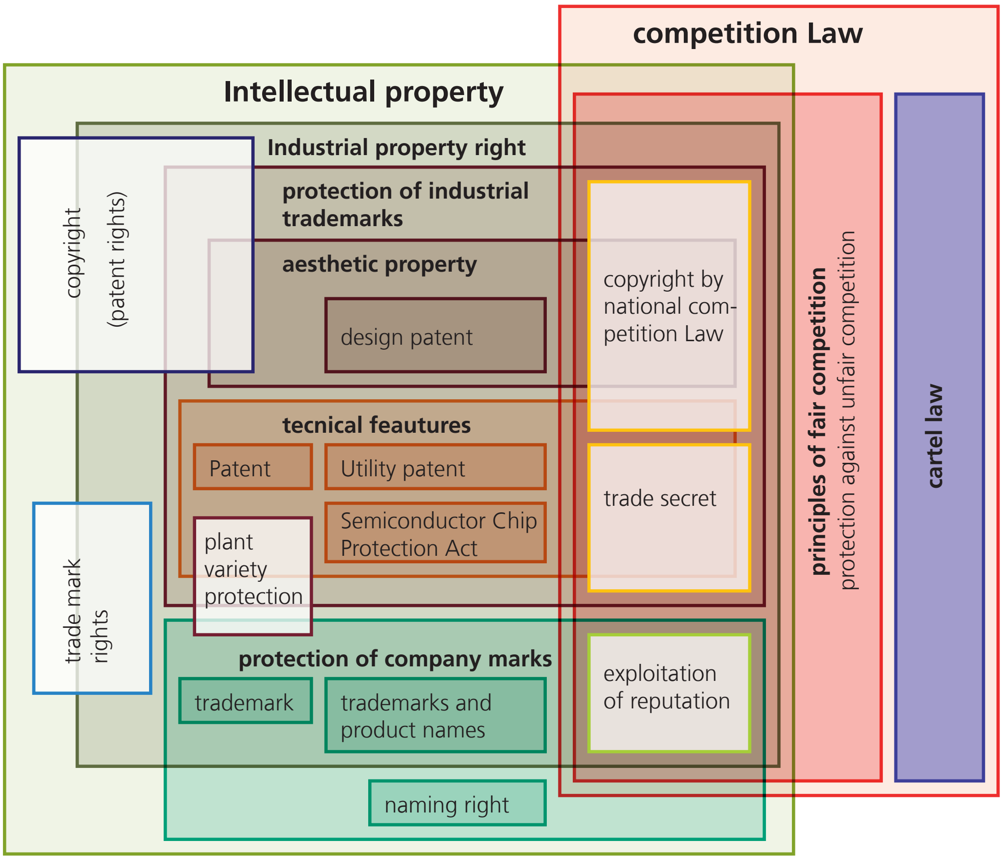

competition Law
Intellectual property
Industrial property right
copyright
(patent rights)
protection of industrial trademarks
aesthetic property
design patent
copyright by national competition Law
tecnical feautures
Patent
Utility patent
Semiconductor Chip Protection Act
plant variety protection
trade secret
trade mark rights
protection of company marks
trademark
trademarks and product names
exploitation of reputation
naming right
principles of fair competition
protection against unfair competition
cartel law

Figure 1.2: Relationship between different types of intellectual property under competition laws

tion measures). On the other hand, technological countermeasures that are integrated right into the products can make it difficult to reverse-engineer products. It is also possible to take suitable precautions that increase protection against manipulation. Additional technological means include labeling and identification methods for distinguishing between original and counterfeit products. A comparison of the primary characteristics of both areas *labeling & identification* and *anti-reverse engineering* can be found in Table 1.1.

To effectively fight product piracy, attack scenarios and protection goals must be coordinated. The implementation of suitable protection measures must ultimately be practical, economical and compliant with the law.

1.2.1 Taking attack scenarios into account

The attack scenarios are product-specific and can vary a great deal. As a result, coordinated protection measures must be selected. For example, the primary risk for a manufacturer of game consoles is not the fact that the product is cloned[1],

[1] In this case, proprietary and exclusive components (such as special graphic chips) are normally used to prevent product cloning.

	Labeling & identification	**Anti-reverse engineering**
Target group for the measure; (work to be performed by . . .)	Distribution parties (customs, dealers) & possibly customers & ggf. Kunde	Competitors, imitators
Testing & target group awareness required	Yes	No
Protection against product-specific know-how transfer and intellectual property	No	Yes
Protection against manipulation	No	Yes

Table 1.1: Comparison of technological measures

but rather that hardware or firmware manipulations make it possible for anyone, even lay people, to play pirated games [8, 14]. The same applies to navigation systems: Manufacturers want to sell the latest maps for their devices. Therefore, they are interested not only in protecting the navigation software against pirated copies but also the map material, in particular. An interesting example involves manipulated card readers (terminals) for bank cards: There have been cases in which defrauders used additional hardware that was secretly *integrated into the devices* (hardware-based Trojan horses) to steal PIN numbers and card data [6]. The information obtained was forwarded directly to the defrauders over the mobile communications network in the form of SMS text messages. The mobile communications technology needed to do this was part of the Trojan horse hardware. The difference between this approach and so-called "skimming" attacks is that the additional hardware is not simply placed as unobtrusively as possible on the terminal housing from the outside, but the special Trojan horse hardware is integrated right into the card terminal as electronic modules. This can be done either in a poorly monitored production or sales chain or by targeted break-ins into the points of sale where high-traffic card terminals are located.

A product's life cycle is also important in assessing the threat of piracy.For example, a mobile telephone that is cutting-edge today loses its attractiveness for a broad customer base after a certain amount of time (approximately two to four years), when the technology is considered outdated. This gives a product pirate a shorter time window in which he can create and market imitations. On the other hand, products exist that have a much longer life cycle and can be sold nearly unchanged within this period of time (e.g., industrial equipment or spare parts in the automotive industry).

It would be plausible to assume that the product pirates take a product- or component-specific approach to their work, i.e., that they select their methods and tools according to their objective. In a product composed of several modular components (such as a housing, mechanics, electronics, software), the product pirates can use a practical method for categorizing the individual constituents: Which components are standard and where is there special or valuable know-how that is worth extracting? They can then decide which cloning variant is most suitable: to adopt the concept or create a slavish one-to-one copy.

Depending on the system complexity, they can also follow specific strategies for individual components. Viewing individual categories or product components permits a reductionist solution approach. According to the principle of divide and rule, pirates can also selectively analyze and understand complex subcomponents. In other words, they can focus their efforts on interesting and valuable components and acquire expert knowledge as needed. This enables them to also grasp and understand the totality of a system. The reductionist approach gives product pirates several ways in which to proceed (standard components are assumed to be available):

Focusing and specialization: One option is to use targeted and specialized methods and tools that make it possible to reverse-engineer relevant subcomponents. For example, 3-D scanners could be used to clone housings in order to determine object dimensions for computer-assisted design and manufacturing methods [13]. On the software level, binary code analysis would be one example for which specialized knowledge, procedures and tools are needed, depending on the computer system [4].

Substitution and modification: If individual components are resistant to reverse engineering, an alternative technology or a functionally similar subcomponent may be used which is easier to clone or may even be available as standard technology. This approach therefore involves substituting technologies or components. As a result, the clone may be lower tech or of poorer quality, since the product pirate replaces a more advanced technology that he has not mastered with another one (for example textbook technology). The loss in quality is often not easy to perceive, i.e., the cloned product may bear a very similar physical resemblance to the original product and also appear to function equally well at first glance. Instead of creating slavishly exact copies, the plagiarist's stated goal may also be to produce "only similar" plagiarized products or conceptual counterfeits. The targeted deviation from the original product can affect both the functional and material properties of the cloned product. Functional modifications are reflected to a greater or lesser degree in the product characteristics. Material-specific deviations, on the other hand, can be viewed as substitution variants that do not necessarily have to have a functional impact—despite such modifications, a clone of this type may under certain circumstances meet the criteria for a slavish copy, depending on how strict an evaluation is made. Modifications can, of course, also affect the outer appearance or the operating concept. Of particular import are safety-relevant deviations that negatively impact operating safety, for example, or which make it impossible to meet safety standards—in the worst case, moreover, falsified seals of quality create the appearance that the products are, indeed, safe.

1.2.2 Determining protection goals

Confiscated imitations are often analyzed to gain a better understanding of how pirates work. The imitations can be used to determine the quality of product clones and, if possible, identify organizational and technological approaches. At the same time, findings on the market penetration of counterfeits and the product pirates' financial resources may be of interest. It may also be possible to gain information about organizational structures from the distribution channels.

In principle, the original product should be examined for "risk of cloning" as a preventive measure against product piracy. The amount of effort it takes to clone a product must also be determined. A product pirate will set his own objectives, depending on the product characteristics. As explained above, he can try to extract interesting knowledge from the product or (if possible) replace complex components with lower-quality alternatives.

A methodical analysis of the components is most useful for copying a product. The product is broken down into its constituent parts in order to discover functional or technological dependencies. Among other things, standard and non-standard components are identified.

In producing a counterfeit, the product pirate has boundless flexibility when it comes to making deviating changes to the original product. In this context, it is important to correctly assess the possibilities and limitations of potential protection goals and measures.

In some situations, general observations can be made on the ways in which the individual subcomponents can be protected. By breaking the product down into its basic components (product partitioning), one ideally obtains a useful means of categorizing the level of protection that individual product components require. Measures should be aimed at the product's value to the company and the need to protect this value. An appropriate focus reduces complexity.

1.2.3 Implementing protection measures

In protecting the product through technological means, local, isolated measures can be taken as well as holistic ones that link a housing, for example, or form of a product with its electronic interior or which interconnect hardware and software cryptographically. The question arises of what measures will offer the totality of a system (the overall product) adequate protection against plagiarism . In certain situations, modular protection provides local and isolated protection mechanisms that can be systematically circumvented or removed. Technology or component substitution enables an alternative technology or a functionally similar subcomponent to be used. This approach would make it possible to copy a concept. A product-specific and holistically coordinated application of multiple protection measures offers a relatively high level of protection (see Figure 1.3).

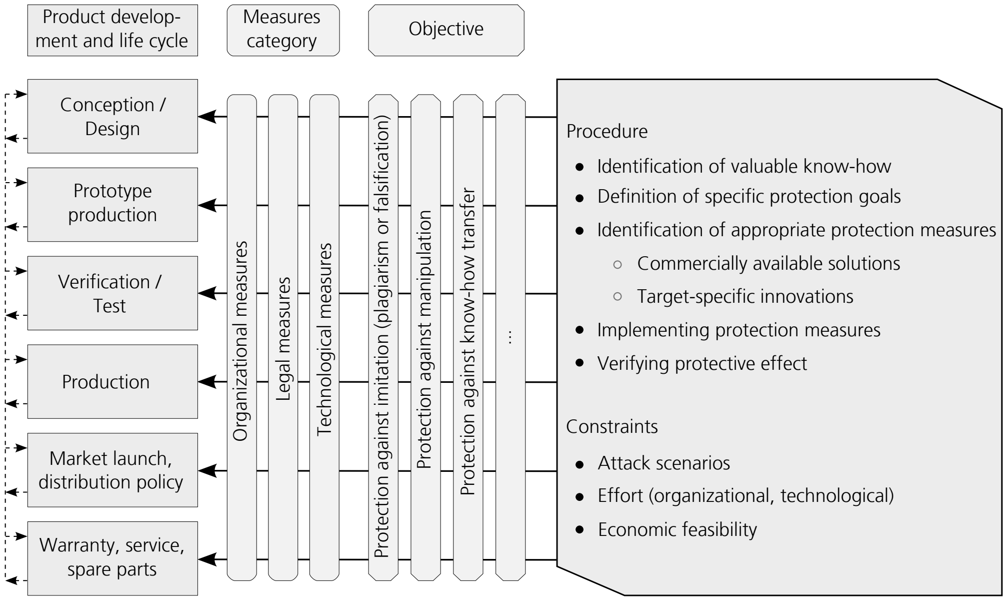

Product develop-ment and life cycle
Measures category
Objective
Conception / Design
Prototype production
Verification / Test
Production
Market launch, distribution policy
Warranty, service, spare parts
Organizational measures
Legal measures
Technological measures
Protection against imitation (plagiarism or falsification)
Protection against manipulation
Protection against know-how transfer
...
Procedure
Identification of valuable know-how
Definition of specific protection goals
Identification of appropriate protection measures
Commercially available solutions
Target-specific innovations
Implementing protection measures
Verifying protective effect
Constraints
Attack scenarios
Effort (organizational, technological)
Economic feasibility

Figure 1.3: Protection measures in a product's life cycle

A product's technological characteristics have a significant influence over the risk potential. The widespread standard technology, which is used in order to reduce costs, is particularly susceptible. Such non-exclusive technology can typically be purchased and used by anyone. Simple textbook technology is easy to clone or can be replaced by functionally similar technology. A comprehensive protection measure alone (that goes beyond the standard technology) appears to be beneficial in this case. Ideally, the core technology offers good protection against cloning per se, since it is difficult to master, requires specialized know-how, and the product characteristics are largely dependent on this unique technology. An example is described in [9, section 3.3.5].

An important consideration is certainly also how difficult it is to extract know-how from the product. Both hardware and software solutions may reveal valuable know-how through reverse engineering methods [13]. Once again, the underlying technology (to be analyzed) can be a determining factor in the effort required. Let us take binary code analysis as a specific example: In this regard, many powerful tools (some of them available for free) exist for the x86 computer architecture (such as disassemblers, debuggers, emulators, analysis of file and memory images), while far fewer tools—if any at all—are available for exotic computer architectures. Technological expertise is helpful for specific effort and costbenefit analyses. On this basis, for example, it is possible to assess whether suitable tools are already available for analysis or whether they must first be developed (which would be more expensive and time-consuming).

Useful protection measures may under certain circumstances also apply to the production methods, production facilities and possibly even the distribution channels. Suitable organizational and possibly also technological measures permit constructive and protective action to be taken in these areas (technological differentiation [9, section 3.3.5], track & trace technology, measures to combat the gray market).

The protection goals must be formulated on a product-specific basis and thus relate only to specific product characteristics, production methods or distribution channels (see Figure 1.3). The amount of protection to be achieved with which measures must be determined and an assessment made as to how much of an obstacle to piracy a plagiarist must overcome. The protection will have to cover the core competence in the product, which should be the primary protection goal. Once again, different degrees of distinction can be considered. It is important to identify the limits of the selected or implemented protection measures (evaluation or verification of the level of protection in relation to the protection goals).

1.3 Case study: Digital receiver clone

As the world's oldest and largest manufacturer of antennas and one of the leading suppliers of communications technology, Kathrein fell victim to product pirates in 2008. Kathrein published documentation relating to the clone of its UFS 910 digital receiver [7], on which basis the company brought attention to the differences between the original and the counterfeit. This case study very clearly illustrates how product pirates work.

In many cases, consumer deception begins with imitation of the packaging and housing. Care is taken to ensure that the consumer will identify the counterfeit as the original product without hesitation. Since special attention to these precise points was paid in the case of the digital receiver, this counterfeit was a slavish counterfeit. Differences from the original packaging can be detected only through direct comparison [7]: Certain labels are printed directly on the packaging, the glued-on serial number does not match the one on the device, and the design is that of the previous version. Differences on the receiver housing and the remote control are also not easily discerned by the consumer. Minor details such as paint pigmentation or a different arrangement of ventilation holes are not immediately apparent without making a direct comparison.

Although counterfeits could often be identified by close examination just a few years ago due to poor quality processing, the optical quality has since improved so much that the copy cannot be distinguished from the original at first glance. Counterfeiters benefit from a large selection of tools such as high-resolution cameras, 3-D scanners and similar devices. Manufacturers of the original products cannot easily protect the outer appearance of their products. One way that they can make the design of their packaging or housings distinct from that of copies is to mark them with hard-to-fake code patterns or electronically readable labels. This includes holograms and optically variable inks whose security is based on the confidentiality or limited availability of the technology, closure seals that are destroyed when the product is opened, limited access to microlettering technology (e.g., in euro bank notes), 1-D, 2-D and 3-D barcodes, encrypted security codes and RFID transponders. There are also special paints, microparticles, DNA labeling molecules, laser-based surface scans and digital watermarks that can be checked only with the use of special measuring equipment [2]. Although these markings allow counterfeit products to be identified, they do not prevent their manufacture or the use of someone else's intellectual property, as illustrated in Table 1.1 above.

If we take a closer look at the clone's interior, we can detect a number of substituted standard components [7]. These components are USB and CI slots that are structurally identical but purchased from other manufacturers, as well as power plugs with non DIN-EN-compliant labeling and a different SCART module. In addition, there are other components that come from the same manufacturer but do not match the original design, such as a display with a different character size.

Certain modifications resulting from lack of know-how can also be identified. Thus, a protective film between the power supply unit and the RS232 board, which is used in the original product to meet safety standards, is incorrectly mounted. A deep standby circuit for conserving energy is also integrated into the original power supply unit and missing from the clone. Although the lack of a deep standby circuit does not affect the functionality of the product, viewed from the outside, it does reduce the quality of the clone.

In the following sections, we explain in greater detail how hardware and software reverse engineering gives counterfeiters the opportunity to clone most of the hardware and even the software installed on the device and what can be done to prevent this.

2 Protecting embedded systems

To reach the intended level of protection for embedded systems, the analysis and protection measures are first examined separately for hardware and software. The different levels of the measures inevitably overlap at certain points where the hardware and software must interact. To protect the overall system, hardware and software measures must therefore be coordinated with each other.

2.1 Hardware reverse engineering and countermeasures

There can be different reasons for the desire to analyze a third-party system, or more precisely, to engage in reverse engineering. Apart from illegal cloning and copying of original products, reverse engineering can be used for economic purposes, for example to assess the value of a competitor product as well as the profit margin the product is giving a company on the market. In addition, the knowledge thus obtained can be used to further develop a company's own products or to compare them with the competition.

If a company wishes to expand its product range, reverse engineering a competitor product available on the market is often easier or more efficient than investing time and money in its own research and development work. From a legal standpoint, reverse engineering of electronics lies in a gray area between legal analysis of competitor products in order to make one's own product compatible or exposing plagiarism and illegal copying and thus copyright violations and patent infringement.

Companies such as the Canadian Chipworks have specialized in reverse engineering. In a 2009 publication [15], they reported on current procedures used to analyze semiconductor electronics. They divide reverse engineering into the following steps:

- *Product teardown* - identifying products, packaging, internal boards and components
- *System level analysis* - analyzing functions, timing and signal paths
- *Process analysis* - investigating the technology used
- *Circuit extraction* - reconstructing the in-chip circuitry

2.1.1 Product teardown

Product teardown or, in other words, identifying individual components, is the top level of reverse engineering. The circuit components used are of interest, for example, in estimating material costs. The board is photographed and the components identified on the basis of the printed labels on their housings.

Such analyses cannot be easily prevented, but the product teardown can be made more difficult by modifying the labels used, removing them with lasers after assembling the board (laser erasure, see Figure 2.1) or not printing them in the first place. The circuit can also be encapsulated with opaque epoxy resins, polyurethane resin or silicone rubber. However, encapsulation can cause problems with heat management and system maintenance, since these techniques were originally designed for components under harsh climatic or high-vibration conditions.

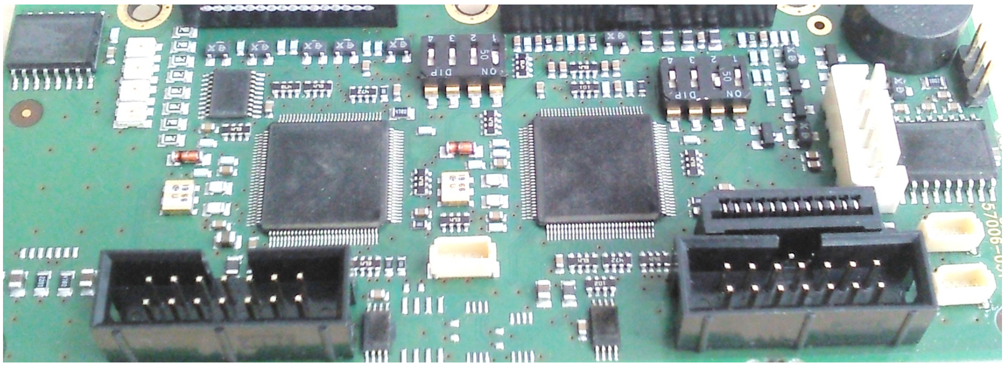


Figure 2.1: Removed package labels make chip identification more difficult

However, this measure is circumvented if the reverse engineer has access to an x-ray machine or a tool for removing the chip package (etching agent, grinding equipment). This reveals the chip designations that the semiconductor manufacturers attached directly to the chip for identification, as shown by a photo of an exposed microcontroller in Figure 2.2. The enlarged image section on the left shows the chip in question.

2.1.2 System level analysis

The next step is to conduct a system analysis, in which the communication between components is analyzed in addition to component identification. A distinction can be made between reverse-engineering the system and functional analysis. When reverse-engineering the system, a photo of the circuit is first taken, just like in product teardown, and the identified components and their connections are noted. The board is thus broken down into its individual parts piece by piece and analyzed. In the case of multilayer boards, each individual

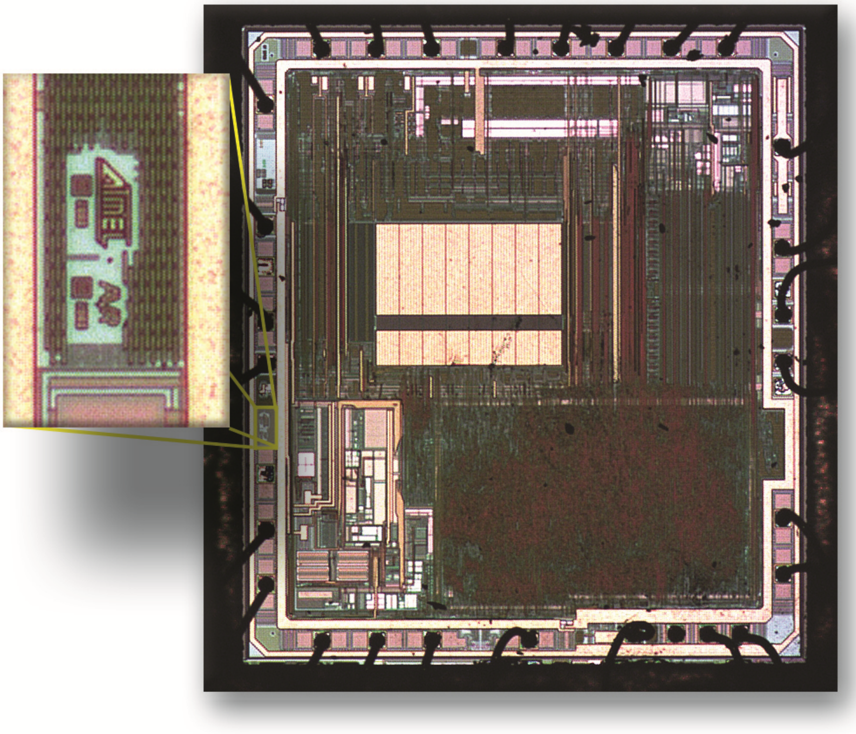


Figure 2.2: Unpacked microcontroller reveals manufacturer name (Atmel AVR)

layer is first removed, digitized and the system functionality reconstructed using software tools.

Another procedure is functional analysis in which the focus is not a precise identification of the component bur rather its functionality. All that is usually needed to do this is a signal generator, a logic analyzer and an oscilloscope [15]. Using certain patterns, the system is excited by the signal generator to perform operations and the functionality of the chip evaluated with the logic analyzer and the oscilloscope. This method can be used for substituting components.

Like with product teardown, the analysis is made more difficult by removing the chip labels or encapsulating the circuit in order to prevent evaluation of the images from being automated with software tools. Designing the board with a multilayer structure can make it more difficult to trace signal paths.

The system analysis can be used to determine not only the functionality of a chip but also the software running on a microcontroller or the implementation of an FPGA. Uncovering this information can be made more difficult on both the software and hardware levels. On the software side, the code can be disguised or its legibility made more complicated. Further details are explained in section 2.2.

The main weak points in the hardware are memory elements for storing the machine code for microcontrollers or the implementation of an FPGA. Special attention must be paid to memory space for sensitive data. In security modules, it should not be possible to simply read secret keys from memory or over corresponding signal lines.

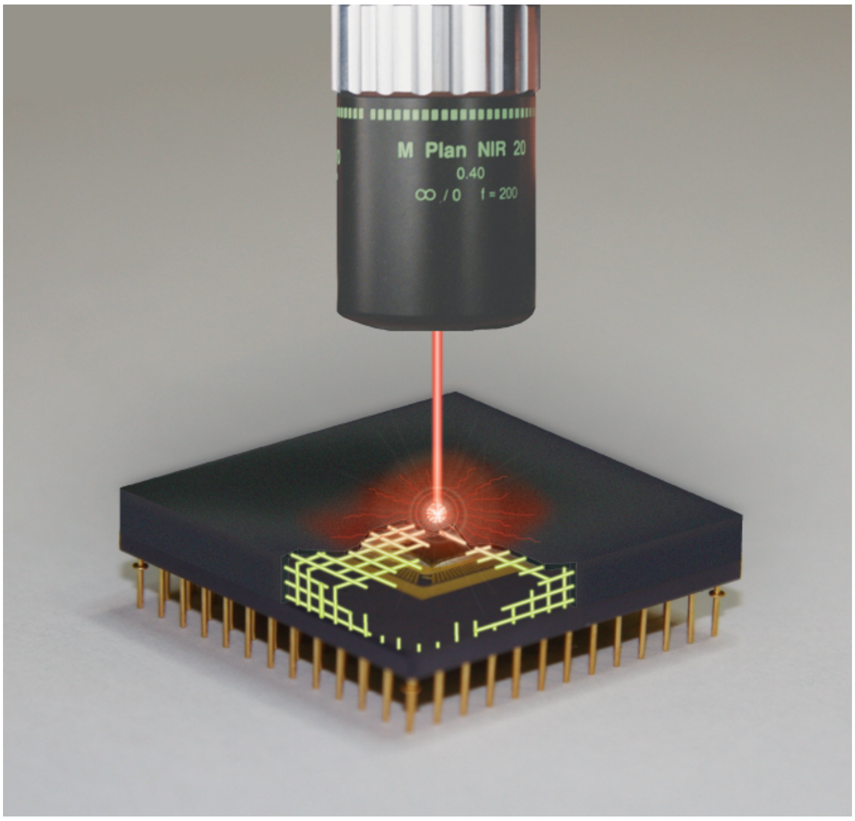


Figure 2.3: Anti-tamper protection: Special chip package permits physical detection of an attack

These memory elements can be read, among other things, over the programming and debugging interfaces of a chip, which are actually provided for programming and troubleshooting purposes and which are present almost exclusively in the terminal devices [1]. So-called software and hardware fuses, which are intended to prevent the memory element from being read, are therefore installed more and more frequently in chips. This is accomplished, for example, by setting certain bits in the chip or by fusing corresponding control lines. These fuses are not always resistant enough to withstand manipulations with targeted methods. Fuse bits that are set can be reset by laser charge injection, while fused lines can be localized by IC microsurgery, process analysis and circuit extraction and bridged directly on the chip. Once the fuse is reset, the system analysis can continue. Effective countermeasures are wire meshes and reactive membranes that are placed around the chip or module and electrically connected to it. If the chip package is then opened for circuit extraction, which destroys the wire mesh or membranes, this will result in destruction of the entire chip functionality (see Figure 2.3).

[1] The following are relevant, for example: Joint Test Action Group (JTAG), Universal Asynchronous Receiver Transmitter (UART), Trace Analyser, In-System Programming (ISP).

Due to the rapid increase in integration of programmable chips into embedded systems, FPGA manufacturers, in particular, are facing the challenge of developing concepts for effective protection against copying. In principle, there are two types of FPGAs: SRAM-based FPGAs, which load their configurations from an external, non-volatile memory at system startup, and non-volatile (flash or antifuse) FPGAs, whose configuration remains the same after a one-time write operation. When using SRAM-based FPGAs, care must be taken to ensure that the configuration data cannot be read over the data bus while it is being loaded to the FPGA. All FPGA manufacturers achieve this by relying on proprietary bit streams, which are intended to make it difficult or impossible to decipher the underlying code. However, this does not prevent the FPGA from being cloned, since the bit stream can be loaded one-to-one to a structurally equivalent FPGA. As a result, the bit stream is often stored in memory in encrypted form, transmitted to the FPGA at system startup and then decrypted in the FPGA. A suitable key management system is needed for this purpose so that the key cannot be read from the chip. Fraunhofer AISEC in Munich is currently researching a new key management method. This is done by taking advantage of minor inaccuracies during chip manufacture which inevitably arise from process fluctuations, in order to generate a unique, electronic fingerprint. This fingerprint can then be used as the key (see Figure 2.4).

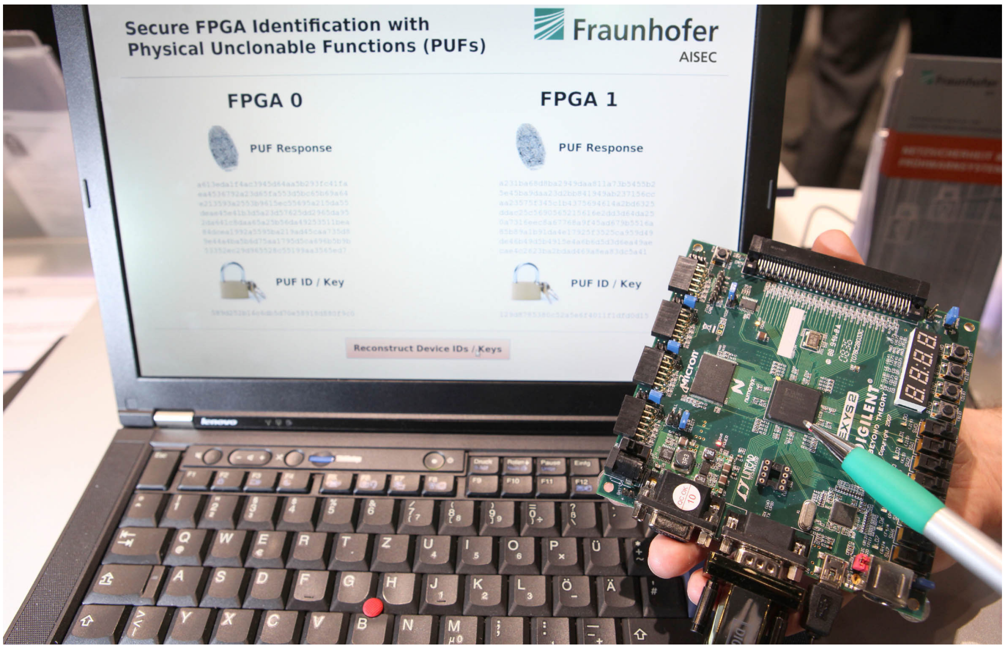

Secure FPGA Identification with
Physical Unclonable Functions (PUFs)
Fraunhofer
AISEC
FPGA 0
PUF Response
PUF ID / Key
FPGA 1
PUF Response
PUF ID / Key
Reconstruct Device IDs / Keys

Figure 2.4: Using *physical unclonable functions* for generating secret keys

The advantage of this method is that the key can be generated at runtime and does not have to be stored in a memory. In addition, the inaccuracies in the circuitry cannot be reconstructed and the fingerprint thus cannot be copied. This is therefore also known as a physical unclonable function (PUF). Scientists have been researching how to integrate PUFs into circuits since 2002. During that time, different variants for implementing PUFs have been developed. One of them is described in [5]: A PUF is integrated on a single chip in silicon, together with the associated measuring electronics.

In many cases, it is not enough to simply store the key. A further analysis is added which enables an attacker to calculate the secret key with the help of physical information emitted during encryption. This analysis is known as side channel analysis and makes use of the physical variables of time and power consumption. An "unprotected" cryptographic circuit may require different amounts of calculation time, for example, depending on the message to be encrypted and the key used, since different functions are performed or different numbers of memory accesses made, depending on the data. If the message, for example the encrypted bit stream for an FPGA, is known, the key used can be determined with the aid of statistical calculations. In the same manner, the key can be calculated using the power consumption or electromagnetic radiation of the chip. Almost all circuits today are manufactured in CMOS technology. This technology consumes current only when the transistors in the chip change from one state (0 or 1) to the other state (1 or 0). Different numbers of transistors must change states to achieve the different combinations of messages and keys. The difference can be detected in the power profile and provides information about the key.

In addition to the side channel analysis, the cryptographic circuit can also be forced to make calculation errors due to external influences (heat irradiation, changing the supply voltage or clock frequency, laser charge injection, etc. – Figure 2.5). The key can then be determined by comparing the incorrect and correct outputs. This is therefore referred to as an fault attack.

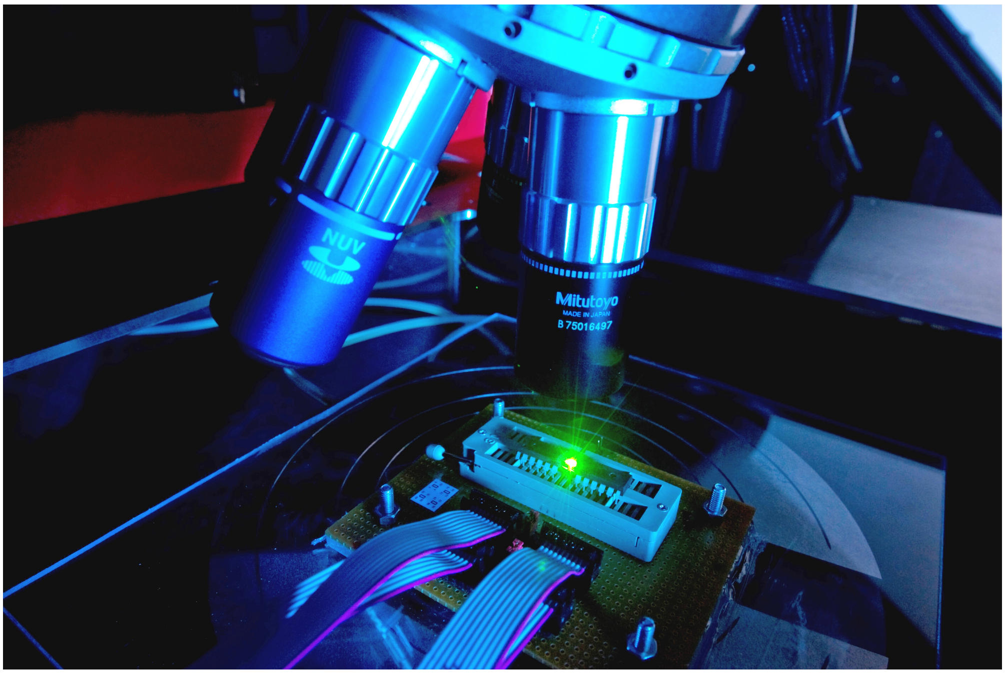


Figure 2.5: Error attack by means of laser charge injection

Countless countermeasures already exist to make it more difficult to perform side channel and fault attacks that are based on randomly installed dummy operations, data masking, dual rail technology[2] or even built-in sources of noise. However, the physical security vulnerabilities and countermeasures associated with them are always dependent on the chip and implementation. The attack methods and countermeasures are therefore always in competition. Research groups such as Fraunhofer AISEC therefore are constantly on the lookout for new security vulnerabilities and are developing specific protection measures for this purpose.

Special hardware modules exist which are hardened against the attacks described here (also see [11]). In most cases, however, no protection measures have yet been integrated into today's widely available and cost-effective standard chips.

Information about the component technology used is needed for some of the system analysis steps presented here. Process analysis and the circuit extraction usually associated with this analysis are used for this purpose.

2.1.3 Process analysis

Information on the process variables, material and structure of a chip is obtained with the aid of the process analysis. The analysis methods and tools are commonly available, since every semiconductor manufacturer needs them for process control and production error analysis. The first step is to remove the package (depackaging). This is done by etching with different acids, also by grinding in the case of ceramic packages. Once the chip has been exposed, various microscopes (scanning and transmission electron microscope, scanning probe microscope, etc.) and chemicals can be used to expose material transitions, structure sizes, number of layers and p/n-doped zones. There is no protection against this analysis.

2.1.4 Circuit extraction

Following depackaging and after determining the individual layer thicknesses on the basis of the process analysis, layer after layer of the chip can be removed by means of different etching and polishing steps. It can then be photographed and the information obtained finally reassembled with software.

The photographing step is followed by the actual circuit analysis. All transistors, coils, resistors, capacitors and conductors must be identified in the individual layers and connected to each other. Depending on the circuit complexity, this is done either manually or with tool support. A final verification checks whether all components are connected and no vias (Vertical Interconnect Access) are open.

[2] Inserting a second encryption operation that processes precisely the opposite item of data. The objective is to ensure that the power consumption always remains the same.

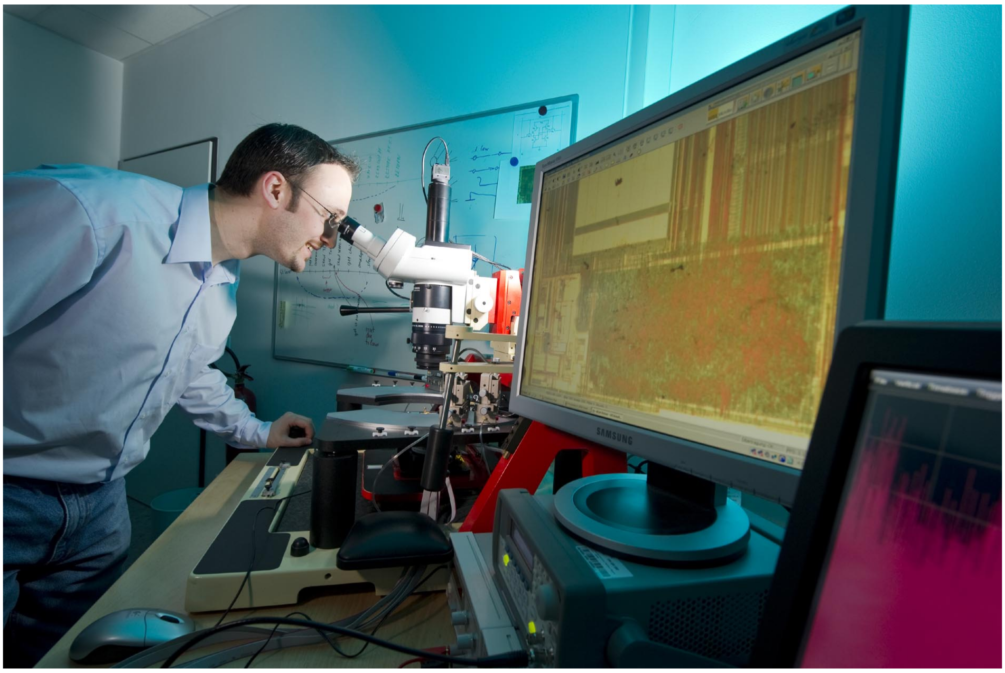


Figure 2.6: Microscope makes the inner circuitry of a microchip visible

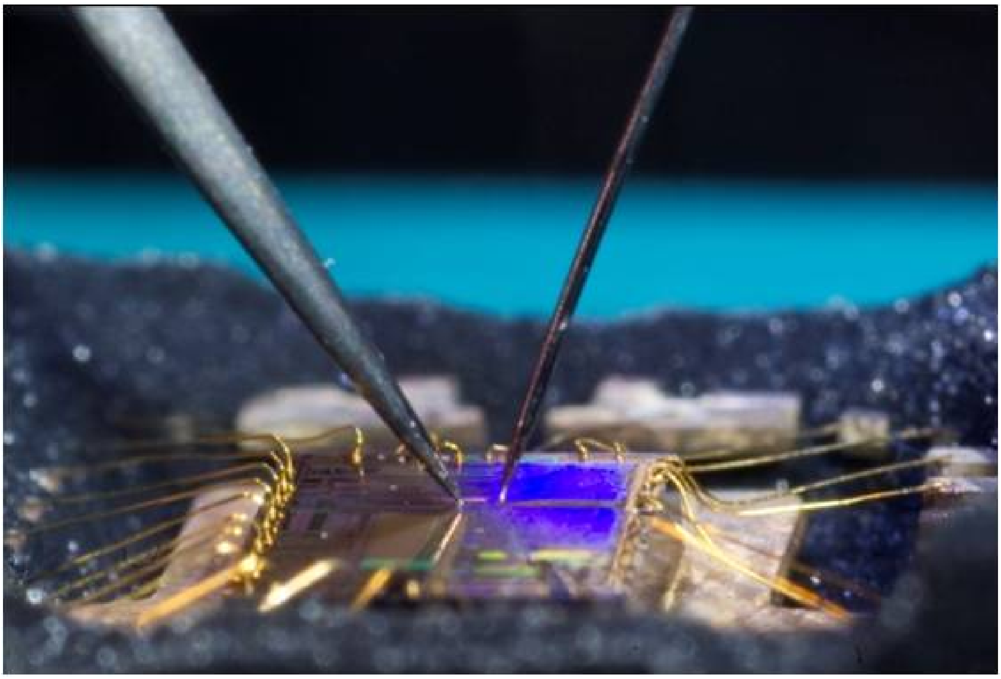


Figure 2.7: Microprobing in an opened chip

The result is a circuit diagram that may, however, be very unclear, due to today's high function block density. Experienced analysts therefore examine the diagram more closely, identify the function blocks and thus simplify the representation.

The result of this analysis can be a list of circuit diagrams for each layer, a complete network list, circuit simulations, block and timing diagrams or even circuit equations [15].

An example of the practical application of circuit extraction is described in [10].

2.2 Software reverse engineering and countermeasures

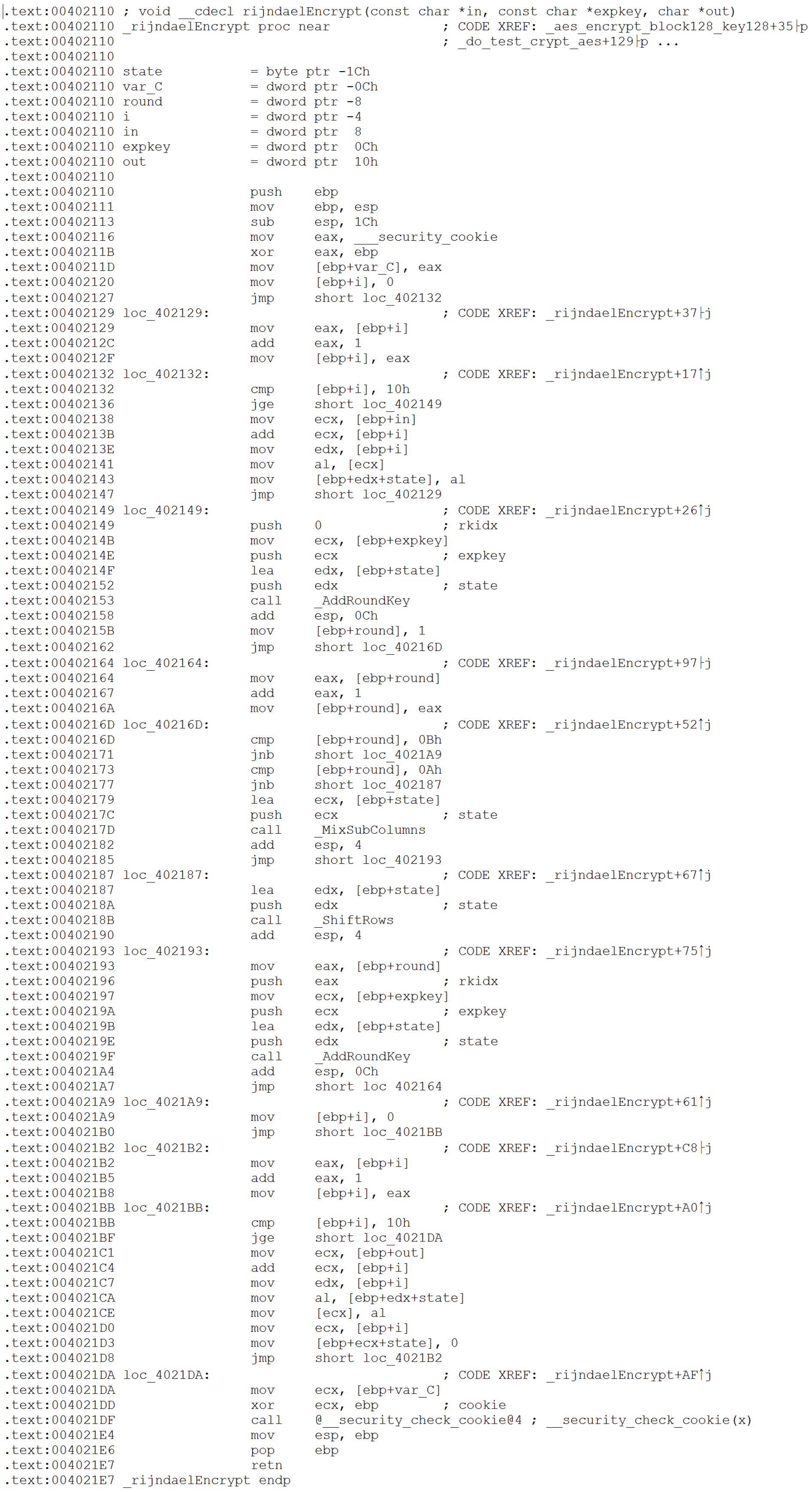
Access to the machine code of a microcontroller (hex dump) or the FPGA implementation raises the possibility that the entire know-how invested in the chip may be extracted. Knowledge of the processor used and its command set enables the machine code to be converted back into assembler code with the aid of disassemblers. Since assembler code is usually not clear enough, additional tools (decompilers) exist which convert the assembler code to a more understandable pseudo code (see Figures 2.8, 2.9 and 2.10).

```
00402110  55 8B EC 83 EC 1C A1 84  6C 40 00 33 C5 89 45 F4  Uïýâý.íäl@.3+ëE¶
00402120  C7 45 FC 00 00 00 00 EB  09 8B 45 FC 83 C0 01 89  ÃE³....Ù.ïE³â+.ë
00402130  45 FC 83 7D FC 10 7D 11  8B 4D 08 03 4D FC 8B 55  E³â}³.}.ïM..M³ïU
00402140  FC 8A 01 88 44 15 E4 EB  E0 6A 00 8B 4D 0C 51 8D  ³è.êD.õÙÓj.ïM.Qì
00402150  55 E4 52 E8 B8 FD FF FF  83 C4 0C C7 45 F8 01 00  UõRÞ©²  â-.ÃE°..
00402160  00 00 EB 09 8B 45 F8 83  C0 01 89 45 F8 83 7D F8  ..Ù.ïE°â+.ëE°â}°
00402170  0B 73 36 83 7D F8 0A 73  0E 8D 4D E4 51 E8 BE F4  .s6â}°.s.ìMõQÞ¥¶
00402180  FF FF 83 C4 04 EB 0C 8D  55 E4 52 E8 10 F2 FF FF    â-.Ù.ìUõRÞ.=
00402190  83 C4 04 8B 45 F8 50 8B  4D 0C 51 8D 55 E4 52 E8  â-.ïE°PïM.QìUõRÞ
004021A0  6C FD FF FF 83 C4 0C EB  BB C7 45 FC 00 00 00 00  l²  â-.Ù+ÃE³....
004021B0  EB 09 8B 45 FC 83 C0 01  89 45 FC 83 7D FC 10 7D  Ù.ïE³â+.ëE³â}³.}
004021C0  19 8B 4D 10 03 4D FC 8B  55 FC 8A 44 15 E4 88 01  .ïM..M³ïU³èD.õê.
004021D0  8B 4D FC C6 44 0D E4 00  EB D8 8B 4D F4 33 CD E8  ïM³ãD.õ.ÙÏïM¶3-Þ
004021E0  E2 18 00 00 8B E5 5D C3  CC CC CC CC CC CC CC CC  Ô...ïÕ]+¦¦¦¦¦¦¦¦
```

Figure 2.8: Hex dump of an x86 machine code

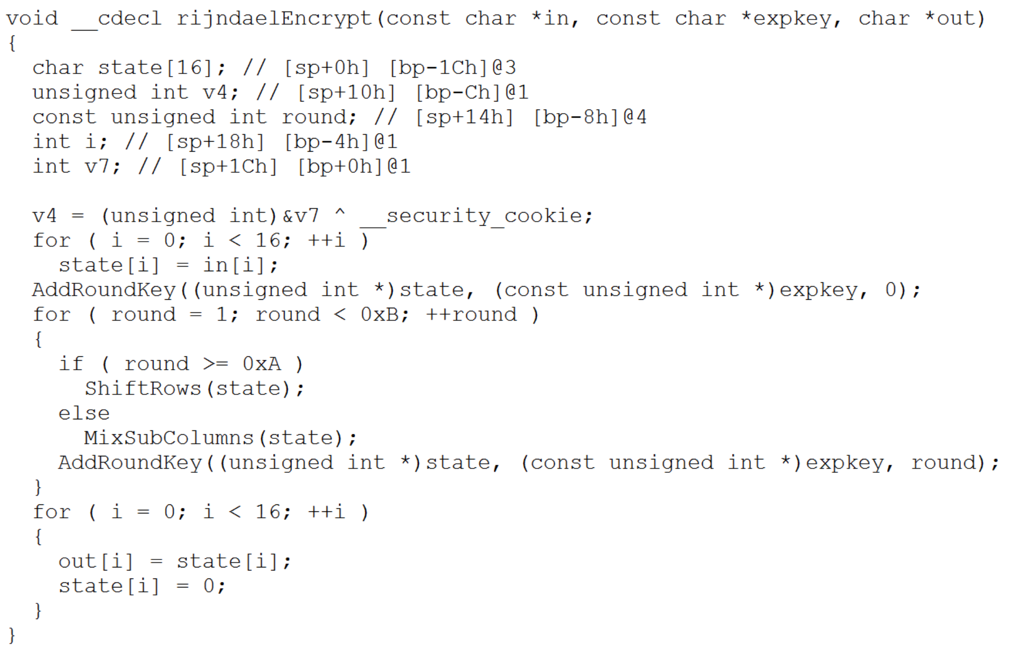
If one manages to read non-volatile memories such as flash chips or to otherwise obtain firmware code (such as update files provided online), knowledge of the associated file system and reverse engineering often make it possible to obtain important information, (configuration) files or secret parameters. This information also allows targeted manipulation of the firmware. Security queries may thus be compromised or enhancements made to the firmware. Practical examples include unauthorized unlocking of computer software (privilege escalation, expansion of rights) by means of so-called jail breaks (in the case of smart phones or game consoles, for instance).

Once access to the code is gained, it can be easily transferred to structurally identical chips. Although verifying a code copy is not a simple process, there are ways to provide digital watermarks (known from image and music protection) on the code level or to prove through statistical methods that the probability of the same code sequence randomly recurring is too slight. If a copy can be verified, there are legal remedies for taking action against the counterfeiter.



```
.text:00402110 ; void __cdecl rijndaelEncrypt(const char *in, const char *expkey, char *out)
.text:00402110 _rijndaelEncrypt proc near              ; CODE XREF: _aes_encrypt_block128_key128+35↓p
.text:00402110                                         ; _do_test_crypt_aes+129↓p ...
.text:00402110
.text:00402110 state           = byte ptr -1Ch
.text:00402110 var_C           = dword ptr -0Ch
.text:00402110 round           = dword ptr -8
.text:00402110 i               = dword ptr -4
.text:00402110 in              = dword ptr  8
.text:00402110 expkey          = dword ptr  0Ch
.text:00402110 out             = dword ptr  10h
.text:00402110
.text:00402110                 push    ebp
.text:00402111                 mov     ebp, esp
.text:00402113                 sub     esp, 1Ch
.text:00402116                 mov     eax, ___security_cookie
.text:0040211B                 xor     eax, ebp
.text:0040211D                 mov     [ebp+var_C], eax
.text:00402120                 mov     [ebp+i], 0
.text:00402127                 jmp     short loc_402132
.text:00402129 loc_402129:                             ; CODE XREF: _rijndaelEncrypt+37↓j
.text:00402129                 mov     eax, [ebp+i]
.text:0040212C                 add     eax, 1
.text:0040212F                 mov     [ebp+i], eax
.text:00402132 loc_402132:                             ; CODE XREF: _rijndaelEncrypt+17↑j
.text:00402132                 cmp     [ebp+i], 10h
.text:00402136                 jge     short loc_402149
.text:00402138                 mov     ecx, [ebp+in]
.text:0040213B                 add     ecx, [ebp+i]
.text:0040213E                 mov     edx, [ebp+i]
.text:00402141                 mov     al, [ecx]
.text:00402143                 mov     [ebp+edx+state], al
.text:00402147                 jmp     short loc_402129
.text:00402149 loc_402149:                             ; CODE XREF: _rijndaelEncrypt+26↑j
.text:00402149                 push    0               ; rkidx
.text:0040214B                 mov     ecx, [ebp+expkey]
.text:0040214E                 push    ecx             ; expkey
.text:0040214F                 lea     edx, [ebp+state]
.text:00402152                 push    edx             ; state
.text:00402153                 call    _AddRoundKey
.text:00402158                 add     esp, 0Ch
.text:0040215B                 mov     [ebp+round], 1
.text:00402162                 jmp     short loc_40216D
.text:00402164 loc_402164:                             ; CODE XREF: _rijndaelEncrypt+97↓j
.text:00402164                 mov     eax, [ebp+round]
.text:00402167                 add     eax, 1
.text:0040216A                 mov     [ebp+round], eax
.text:0040216D loc_40216D:                             ; CODE XREF: _rijndaelEncrypt+52↑j
.text:0040216D                 cmp     [ebp+round], 0Bh
.text:00402171                 jnb     short loc_4021A9
.text:00402173                 cmp     [ebp+round], 0Ah
.text:00402177                 jnb     short loc_402187
.text:00402179                 lea     ecx, [ebp+state]
.text:0040217C                 push    ecx             ; state
.text:0040217D                 call    _MixSubColumns
.text:00402182                 add     esp, 4
.text:00402185                 jmp     short loc_402193
.text:00402187 loc_402187:                             ; CODE XREF: _rijndaelEncrypt+67↑j
.text:00402187                 lea     edx, [ebp+state]
.text:0040218A                 push    edx             ; state
.text:0040218B                 call    _ShiftRows
.text:00402190                 add     esp, 4
.text:00402193 loc_402193:                             ; CODE XREF: _rijndaelEncrypt+75↑j
.text:00402193                 mov     eax, [ebp+round]
.text:00402196                 push    eax             ; rkidx
.text:00402197                 mov     ecx, [ebp+expkey]
.text:0040219A                 push    ecx             ; expkey
.text:0040219B                 lea     edx, [ebp+state]
.text:0040219E                 push    edx             ; state
.text:0040219F                 call    _AddRoundKey
.text:004021A4                 add     esp, 0Ch
.text:004021A7                 jmp     short loc_402164
.text:004021A9 loc_4021A9:                             ; CODE XREF: _rijndaelEncrypt+61↑j
.text:004021A9                 mov     [ebp+i], 0
.text:004021B0                 jmp     short loc_4021BB
.text:004021B2 loc_4021B2:                             ; CODE XREF: _rijndaelEncrypt+C8↓j
.text:004021B2                 mov     eax, [ebp+i]
.text:004021B5                 add     eax, 1
.text:004021B8                 mov     [ebp+i], eax
.text:004021BB loc_4021BB:                             ; CODE XREF: _rijndaelEncrypt+A0↑j
.text:004021BB                 cmp     [ebp+i], 10h
.text:004021BF                 jge     short loc_4021DA
.text:004021C1                 mov     ecx, [ebp+out]
.text:004021C4                 add     ecx, [ebp+i]
.text:004021C7                 mov     edx, [ebp+i]
.text:004021CA                 mov     al, [ebp+edx+state]
.text:004021CE                 mov     [ecx], al
.text:004021D0                 mov     ecx, [ebp+i]
.text:004021D3                 mov     [ebp+ecx+state], 0
.text:004021D8                 jmp     short loc_4021B2
.text:004021DA loc_4021DA:                             ; CODE XREF: _rijndaelEncrypt+AF↑j
.text:004021DA                 mov     ecx, [ebp+var_C]
.text:004021DD                 xor     ecx, ebp        ; cookie
.text:004021DF                 call    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:004021E4                 mov     esp, ebp
.text:004021E6                 pop     ebp
.text:004021E7                 retn
.text:004021E7 _rijndaelEncrypt endp
```

Figure 2.9: Disassembling the hex dump yields the assembler listing belonging to Figure 2.8



```
void __cdecl rijndaelEncrypt(const char *in, const char *expkey, char *out)
{
  char state[16]; // [sp+0h] [bp-1Ch]@3
  unsigned int v4; // [sp+10h] [bp-Ch]@1
  const unsigned int round; // [sp+14h] [bp-8h]@4
  int i; // [sp+18h] [bp-4h]@1
  int v7; // [sp+1Ch] [bp+0h]@1

  v4 = (unsigned int)&v7 ^ __security_cookie;
  for ( i = 0; i < 16; ++i )
    state[i] = in[i];
  AddRoundKey((unsigned int *)state, (const unsigned int *)expkey, 0);
  for ( round = 1; round < 0xB; ++round )
  {
    if ( round >= 0xA )
      ShiftRows(state);
    else
      MixSubColumns(state);
    AddRoundKey((unsigned int *)state, (const unsigned int *)expkey, round);
  }
  for ( i = 0; i < 16; ++i )
  {
    out[i] = state[i];
    state[i] = 0;
  }
}
```

Figure 2.10:
To gain a better understanding, the assembler listing in Figure 2.9 can be converted to a pseudo code listing by decompilation

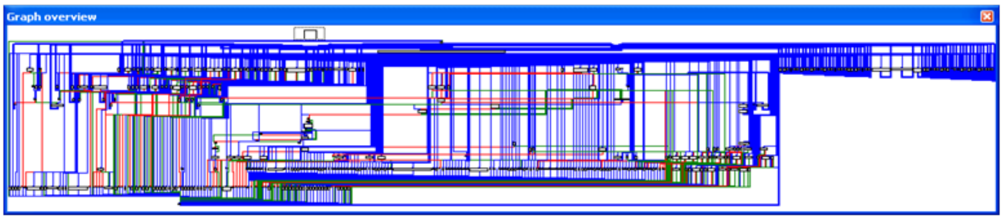
Many countries make it illegal to circumvent the copy protection mechanism of software and implementations for commercial purposes. In addition to protecting illegal copies of the software code, many companies do not want to disclose the know-how they have integrated into the software or hardware. Methods or calculation specifications developed in-house are often a decisive competitive advantage and therefore need to be kept secret. The original manufacturers therefore attach great importance to enforcing the protective goals of *copy protection* and *anti-reverse engineering* for their products. For this reason, both software developers and circuit designers increasingly seek to protect their developments against espionage.

In principle, no exclusively software-based methods exist to protect products against third-party attacks. Effective protection against data espionage or reverse engineering of software products requires a certain amount of hardware support. If the complete binary code of the software is available, the protection measures contained therein can be analyzed, provided that they are only software-based. For example, it is of little use to incorporate a cryptographic process and associated key into the software code in order to use it to protect other software components. This is due, among other things, to the fact that software cannot be protected against modification, emulation or monitored and controlled execution on a virtual machine without hardware support. Suitable hardware support can be used to provide cryptographic protection of certain software components, for example in a dedicated hardware chip. The necessary cryptographic processes and keys are always located within the hardware module and are not present in the software binary code and can thus also not be extracted from the software - neither through statistical analysis of the binary code nor by emulating the software in a virtual environment. In this constellation, the hardware and software form a functional unit, and it is not possible to change or analyze the encrypted software components in any useful way without the special-purpose hardware.

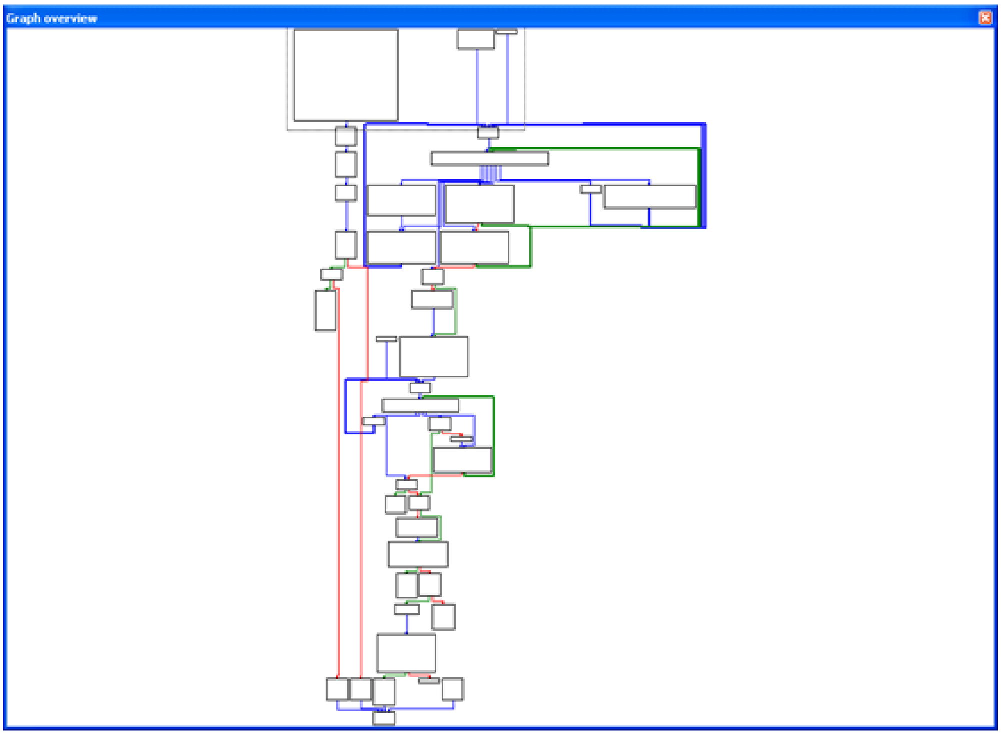
Hardware-based protection measures for software are also not automatically secure. The first dongle solutions are an example of this: The objective is to permit only authorized users to use the software products, that is, users who are in physical possession of the dongle. The weakness of the first dongle solutions was the fact that the software binary code could be easily analyzed. If the query for the existence of the dongle could be located in the binary code, this query could simply be skipped, for example by changing the assembler command *Branch Equal* to *Branch Not Equal*. An executable file modified in this way would run without any dongle at all. Modern solutions are not so easy to outsmart.

From these types of examples, one can learn that software products need to be protected against manipulation in order for the hardware-based protection to even take effect. This can be difficult to do in typical PC environments. Such systems are a worthwhile target for crackers, due to their mass implementation, and have been well documented and analyzed. A wide range of sophisticated tools, which are relatively easy to obtain on the Internet, also exist for software reverse engineering and manipulation. A product that is based on embedded hardware is easier to protect because the manufacturer usually has more hardware and software configuration options at his disposal, in particular if all components come from a single source and can be coordinated with each other (from a security standpoint).

A software product is also, in principle, an item of intellectual property that is worthy of protection. A basic goal may therefore be to prevent the attacker from understanding the purpose and functionality of the software components and internal functions. So-called obfuscators, which disguise the software code, are used for this purpose. Protection of this type is important for programming languages such as Java and C#, since these languages do not generate native machine code but rather map the source file to an intermediate language. It is possible to perform a binary code analysis on this intermediate language and thus reconstruct the original source code from the assemblies of Microsoft .Net Framework with the aid of suitable class browsers. This characteristic is an undesirable status where product piracy is concerned. Therefore, program packages exist for Java and the .Net languages that convert all functions, variables, objects, classes and types to names that are as meaningless as possible. In addition, the program packages disguise the calls and links between different components, ultimately encrypt all character strings and generate tangled spaghetti code to give the attacker as few reference points as possible for his analysis (see Figure 2.11). These measures are intended to deter potential attackers as well as increase the cost and time expenditure to the extent that they are sensitive and too difficult to calculate for an attacker. However, any troubleshooting that may be necessary after the program transformation cannot be realistically performed. In addition, due to the obfuscator, the transformation can even produce additional errors in the software product, which may remain unnoticed until the product reaches the customer as a result of the more difficult maintainability and then generate unforeseen costs. Problems may also arise due to the program's modified runtime behavior, increased memory requirements and more difficult handling of updates or patches.



(a) Tangled code flow of an obfuscated function



(b) Simple code flow of the above function after it has been automatically untangled

Figure 2.11: Effects of (de)obfuscation on the code flow

Encryption and compression processes (some of which are proprietary) are also carried out in protecting software applications. Although the level of protection can be very high if suitable methods and expert implementation are used, this may go hand in hand with performance losses and a rather great amount of know-how needed for viable implementation (related catch-words are: key management, selection of optimum methods, implementation aspects, side channel attacks). If complex, practical requirements are involved, a non-specialized developer may not be able to justify this additional effort.

The statical analysis of the code can reveal whether and where cryptographic calculations take place. Corresponding points on the program binary code can be located by means of characteristic code signatures (similar to the functionality of virus scanner programs). In addition to the standard procedures, a certain number of proprietary methods are used or an attempt is made to disguise the cryptographic code with the aid of obfuscation techniques. Dynamic code analyses and memory monitoring at program runtime can, under certain circumstances, compromise the program or important data, even if encryption is used, if the attacker manages to access the relevant cryptographic parameters in the computer system. Cryptographic techniques are therefore ideally combined with specialized hardware.

Without specialized hardware, it is possible to attack the binary code (possibly at runtime, using a debugger or a virtual runtime environment). This can also be seen in modern copy protection mechanisms for PC software, which according to experience can be analyzed and circumvented in practice.

In contrast to the standardized PC systems, customer-specific embedded systems offer a great deal more flexibility in optimally integrating cryptographic processes—in both software and hardware that have been coordinated with each other from a security standpoint.

2.3 Combining hardware and software protection

A dongle (copy protection connector) is used primarily to protect software against unauthorized copying. They are sold by certain manufacturers together with a development package. In principle, a dongle is a *hardware security module* in which the cryptographic keys are protected against both physical attacks (such as side channel attacks) and software-based attacks. A modern dongle for USB ports is approximately the same size as a memory stick and frequently contains the necessary drivers in a separate memory. The internal memory enables cryptographic techniques to be integrated directly into the dongle. Using encryption makes it more difficult for an attacker to localize and suppress the queries for the dongle hidden in the program. A challenge response authentication mechanism (Figure 2.12) can be used to check for the presence of a dongle. A cryptographic dongle can also be used during program runtime to protect the software against manipulation (for example, by using a cryptographic integrity check of the binary code instructions in the memory)—this at least makes reprogramming the software much more difficult. The current dongle generation is therefore very difficult to fool and also offers good implementation. Encryption also helps prevent the intellectual property from being viewed, since the attacker must first break the encryption in order to begin his analyses.

A *secure memory device* functions in much the same way as a cryptographic dongle. Instead of being connected to an external port, a secure memory device is integrated as a module into the hardware design. Along with the internal

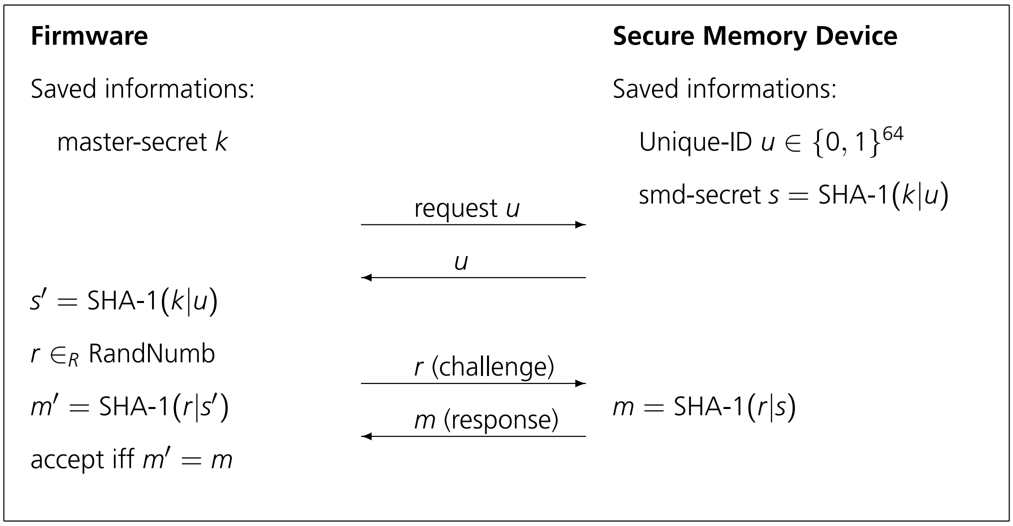


Figure 2.12:
Challenge response authentication for checking for the presence of a hardware module

memory, which can also contain individual parameters or have a unique chip ID, these modules have useful cryptographic basic functions that are used to protect the overall system (hardware and software – see Figure 2.12).

In modern PCs, special modules have found their application under the designation of *trusted platform modules* (TPM), where they are used primarily to prevent manipulation of the BIOS and the operating system boot program. The chip is passive and cannot directly influence either bootup or operation. It contains a unique ID and can therefore be used to identify the system. As a cryptographic device, the chip is able to perform cryptographic processes and is used as a secure key memory and random number generator.

The majority of manufacturers of programmable logic circuits (such as FPGAs) offer protection mechanisms for certain product series (*design security*). Some use an additional secure memory device for this purpose. A number of FPGAs have certain protection measures and/or cryptographic processes integrated into them.

The degree to which the integrated chips are resistant to non-invasive and, in particular, (semi)-invasive attacks must be considered in particular. To obtain worthwhile knowledge, some institutions or competitor companies are perfectly willing to pay the cost of the latter costly attacks and to hire commercial service providers with well-equipped labs who specialize in the reverse engineering of hardware modules and boards. The complexity of reconstruction in practical terms usually depends on the manufacturing technology of the hardware modules. The specific physical characteristics of a module must therefore be taken into account in assessing the level of protection.

Another way to protect intellectual property is to manufacture special chips, known as ASICs. This solution is very expensive, due to the retooling costs, and only pays off in extremely large volumes. An approach of this type is therefore only practical for mass-market products. However, once again the effort required for reconstructing the circuit depends on the manufacturing process or structure sizes. This can mean that an ASIC may under certain circumstances be easier to analyze than an FPGA that is protected against microscopic attacks. Security is achieved by giving such FPGAs a fully uniform microscopic structure that remains the same even after the FPGA is configured. This technology currently provides a high degree of protection against intellectual property theft.

In addition to the above-mentioned hardware modules, cryptographically enabled microcontrollers and CPUs also exist which may be used as modules for implementing a protection concept. Before using such modules, it is necessary to clarify whether their hardware-implemented cryptographic processes are state of the art and thus cannot be compromised by side channel attacks or (semi)-invasive techniques.

3 Fraunhofer AISEC Profile

3.1 General Description

As a specialist in IT security, Fraunhofer AISEC develops instantly deployable solutions which are fully aligned with the needs of the client. This tailored treatment is made possible through the efforts of more than 70 highly-qualified employees working in all relevant areas of IT security. At our location in Munich/Garching, Fraunhofer AISEC is currently building up three new research and development departments which specialize in hardware-related security and the protection of complex networks and services. At the same time, a test center is being established for security and reliability testing of hardware and software applications and components; additionally, the test center will provide services for functional, interoperability and conformity testing.

3.1.1 Network Security and Early Warning Systems

The security of distributed network services is a significant challenge for companies. On one hand, they require a mechanism with which to identify and authorize participating entities. And on the other hand, they need the means to guarantee privacy and confidentiality. Additionally, businesses must also comply with data protection regulations. The Network Security and Early Warning Systems department, in cooperation with industry partners, develops solutions which enable the secure operation of networks and services. To this end, Fraunhofer AISEC develops concepts, procedures and protocols in, for example, security in All-IP networks and Future Internet, Personal Area Networks, detection and prevention of malicious software and attacks, combined machine learning techniques, image understanding, and sensor fusion.

3.1.2 Embedded Security & Trusted Operating Systems

The development of hardware-based security solutions must start with the design of chips and circuits. In this context, the Embedded Security and Trusted OS department concerns itself with methods for self-protection of systems through targeted system-hardening and the transfer of security functionality into hardware. In order to make this possible, Fraunhofer AISEC, together with industry partners, develops and tests trustworthy (mobile) platforms, advanced virtualization concepts, and component identification methods, as well as new testing methods for embedded components. To this end, a hardware development and testing laboratory will be built at our Munich location.

3.1.3 Secure Services and Quality Testing

The Secure Services and Quality Testing department, in cooperation with its project partners, develops solutions and testing tools for the development, composition, hardening, provisioning and operation of safe and reliable service-based business software. Solutions tailored to the particular requirements of specific domains, such as automotive engineering, logistics, e-Government (for example, standardized formats for data exchange in SOA-based processes), and health care are particularly important in this respect. For example, the department develops secure value-added services and service platforms for intelligent environments (Smart Factory, Smart Office), safety components, risk and compliance management processes, and Web Services using Enterprise Service Bus infrastructure.

3.2 Overview of product and intellectual property protection services

Fraunhofer knows the state-of-the-art for technological protection measures. On behalf of its customers, the Institute determines how a particular product can best be protected against imitations.

Products containing unprotected components are an easy target for forgers and threaten the investments of innovative companies. Fraunhofer AISEC points out risks and supports manufacturers of embedded systems in designing products which are robust against piracy. Companies are assisted in the selection of appropriate protective measures and the optimal integration into their products.

If protection requirements cannot be fulfilled by standard market measures for performance or production reasons, the Institute develops innovative techniques, such as the protection of embedded systems. The solutions are effective and customizable and provide substantial protection against reverse engineering as well as proactively concealing the functions of a protected product. Through these techniques, the imitation of products becomes more difficult, and valuable business know-how is protected.

To achieve a high level of hardware security, product-specific conditions, such as seamless integration into existing business and production processes, are considered. Fraunhofer AISEC already provides solutions implemented as portable C implementations and suitable for low-cost microcontroller architectures. Moreover, hardware-based reference designs are offered for programmable logic devices (synthesizable code for low-cost FPGAs). The software and hardware solutions can be combined and customized by parameters.

Furthermore, Fraunhofer AISEC tests embedded systems and IT processes for vulnerabilities and piracy-robustness, testing designs and prototypes as well as completed products. Complete system checks can be carried out as well as analyses of selected components. The Institute's experience in the area of IT-security and design security facilitates effective tests whose results can be incorporated directly into further development and product maintenance.

Services, assets and areas of competence:

- Modern laboratory for hardware security analysis and system evaluation[1]
- Product-specific security solutions
- Hardware- and software-based protection measures (protection against tampering, reverse engineering and product piracy)
- Component and replacement part identification
- Design security: protection using hardware components [2]
- Practical implementation of modern encryption techniques
- Binary code analysis (various architectures, including x86, ARM and CIL-Code/.NET byte code)

Expertise in the aforementioned areas (for both extant products and those in the design phase):

- Analysis of application scenarios, threat and risk analysis
- Custom design and implementation of innovative protective measures
- Support in the integration of established protective measures
- Proof-of-concept and prototype implementations; reference implementations (programming language/hardware description language: VHDL or Verilog)
- Analysis of the state-of-the-art
- Preparation of market overviews for commercial protection measures
- Technology consulting and assessment, action recommendations
- Evaluation of protection measures (protection level), assessment of utility (estimation of circumvention cost)
- Design of tamper-proof products and processes
- Testing of systems and processes for vulnerabilities and piracy-robustness

[1] Individual microchips as well as complete embedded systems are tested.

[2] For example, based on FPGA, Secure Memory Devices, Hardware Security Modules, Trusted Platform Modules, or cryptographically-enabled microcontrollers.

Bibliography

[1] Abele, E. (Hrsg.) ; Albers, A. (Hrsg.) ; Aurich, J.C. (Hrsg.) ; Günthner, A. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 3: *Wirksamer Schutz gegen Produktpiraterie im Unternehmen: Piraterierisiken erkennen und Schutzmaßnahmen umsetzen*. Frankfurt am Main : VDMA Verlag, November 2010. – 223 S. – Mit Ergebnissen aus den Projekten: ProOriginal, KoPira, KoPiKomp, ProAuthent. Weitere Bände der Reihe: [2, 9] 10, 37, 38

[2] Abramovici, M. (Hrsg.) ; Overmeyer, L. (Hrsg.) ; Wirnitzer, B. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 2: *Kennzeichnungstechnologien zum wirksamen Schutz gegen Produktpiraterie*. Frankfurt am Main : VDMA Verlag, November 2010. – 196 S. – Mit Ergebnissen aus den Projekten: MobilAuthent, O-Pur, EZ-Pharm. Weitere Bände der Reihe: [1, 9] 10, 17, 37, 38

[3] BASCAP: *Estimating the global economic and social impacts of counterfeiting and piracy*. London : Frontier Economics Ltd, February 2011. – URL `www.iccwbo.org/bascap` 8

[4] Eagle, Chris: *The IDA Pro Book: The unofficial guide to the world's most popular disassembler*. First Edition. No Starch Press, Inc., 2008 8, 13

[5] Gassend, Blaise ; Clarke, Dwaine ; Dijk, Marten van ; Devadas, Srinivas: Silicon physical random functions. In: *Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA : ACM, 2002 (CCS '02), S. 148–160 23

[6] Heise News: *Visa: Regelmäßiges Wiegen der Kartenterminals schützt vor Manipulationen*. Online News-Meldung vom 09.07.2010 12:16. 2010. – URL `http://heise.de/-1035169` 12

[7] Henke, Andreas ; Janssen, Mark: *Vergleich Kathrein UFS 910 Original mit China Klon*. Online Publikation (PDF-Datei). 2008. – Bebilderte Beschreibung der Unterscheidungsmerkmale zwischen original Kathrein UFS 910 und eines Imitats dieses Produktes 16, 17

[8] Huang, Andrew: Keeping Secrets in Hardware: The Microsoft Xbox^{TM} Case Study. In: Kaliski, Burton S. Jr. (Hrsg.) ; Koç, Çetin Kaya (Hrsg.) ; Paar, Christof (Hrsg.): *Cryptographic Hardware and Embedded Systems – CHES 2002: 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*. Berlin, Heidelberg : Springer-Verlag, 2003 (Lecture Notes in Computer Science 2523), S. 213–227 12

[9] Kleine, O. (Hrsg.) ; Kreimeier, D. (Hrsg.) ; Lieberknecht, N. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 1: *Piraterierobuste Gestaltung von Produkten und Prozessen*. Frankfurt am Main : VDMA Verlag, November 2010. – 192 S. – Mit

Ergebnissen aus den Projekten: PiratPro, Protactive, Pro-Protect. Weitere Bände der Reihe: [1, 2] 10, 15, 16, 37

[10] Krissler, Jan ; Nohl, Karsten ; Plötz, Henryk: Chiptease: Verschlüsselung eines führenden Bezahlkartensystems geknackt. In: *c't – Magazin für Computertechnik* 8 (2008), S. 80–85. – Heise Zeitschriften Verlag, Hannover 26

[11] National Institute of Standards and Technology: Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules / National Institute of Standards and Technology. 5 2001 (FIPS PUB 140-2). – U.S. government computer security standard. Last updated December 3, 2002 24

[12] Neemann, Christoph Wiard ; Klocke, F. (Hrsg.) ; Schmitt, R. (Hrsg.) ; Schuh, G. (Hrsg.) ; Brecher, C. (Hrsg.): *Berichte aus der Produktionstechnik*. Bd. 13: *Methodik zum Schutz gegen Produktimitationen*. 1. Auflage. Shaker Verlag, 2007 9, 10

[13] Raja, Vinesh (Hrsg.) ; Fernandes, Kiran J. (Hrsg.): *Reverse Engineering: An Industrial Perspective*. First Edition. London : Springer-Verlag, 2008 8, 13, 15

[14] Steil, Michael: *17 Mistakes Microsoft Made in the Xbox Security System*. Presented at the 22nd Chaos Communication Congress, December 29th 2005, 18:00, Berliner Congress Center, Berlin, Germany. 2005. – URL `http://events.ccc.de/congress/2005/fahrplan/events/559.en.html` 12

[15] Torrance, Randy ; James, Dick: The State-of-the-Art in IC Reverse Engineering. In: Clavier, Christophe (Hrsg.) ; Gaj, Kris (Hrsg.): *Cryptographic Hardware and Embedded Systems – CHES 2009: 11th International Workshop Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Berlin, Heidelberg : Springer-Verlag, 2009 (Lecture Notes in Computer Science 5747), S. 363–381. – Invited Talk 3 8, 18, 20, 26

[16] VDMA: *VDMA-Umfrage zur Produkt- und Markenpiraterie 2010*. Verband Deutscher Maschinen- und Anlagenbau, 2010. – URL `www.vdma.org/produktpiraterie` 8

[17] Welser, Marcus von ; González, Alexander: *Marken- und Produktpiraterie: Strategien und Lösungsansätze zu ihrer Bekämpfung*. 1. Auflage. Wiley-VCH, 2007 10

[18] Wildemann, Horst ; Ann, Christoph ; Broy, Manfred ; Günthner, Willibald A. ; Lindemann, Udo: *Plagiatschutz: Handlungsspielräume der produzierenden Industrie gegen Produktpiraterie*. München : TCW Transfer-Centrum GmbH & Co. KG, 2007 8, 9