simplified, but it still allows us to make a relatively good estimate of the attractiveness of the different types of attack, and thus the probable lines of attack. Naturally, an attacker would normally choose an attack that requires the least effort and expense. The six criteria listed in Table 8.5 will consciously or unconsciously influence the attacker's behavior.

Table 8.5 Criteria for determining the effort and cost of an attack on the hardware or software of the security components, based on the prerequisites for the attack

Degree of attractiveness	Low	Medium	High
level of knowledge and skills required	high	medium	low
number of secrets required	many	moderate	few
amount of time required	much	moderate	little
acquisition of the necessary technical equipment (purchase or access)	difficult	moderate	easy
access to the components to be attacked	difficult	moderate	easy
value of the result (money or prestige)	low	medium	high

The lower the level of specific knowledge or skills required for an attack, the more attractive it is to an individual or an organization. An attack that does not require the knowledge of any secrets is similarly attractive. This is not inconsistent with Kerckhoff's principle, which says that security should depend only on the key and not on the cryptographic algorithm itself. This is because Kerckhoff's principle does not mean that you have to reveal everything you know in order to make a system secure. The presence of many secrets represents an enormous hindrance to the mounting of a successful attack.

Especially in the case of systematically searching for a key, the amount of time required plays an important role. The classic example of this would be breaking a cryptographic algorithm using a brute-force attack that takes an average of 10,000 years. No serious attack could be based on such an approach.

The attractiveness of an attack is equally dependent on the technical equipment that is needed for the attack. This need not necessarily refer only to the purchase of equipment, since it may be sufficient to be able to rent the equipment or somehow acquire access to the equipment. For example, a device that can generate and precisely position focussed ion beams costs several hundred thousand pounds, but such equipment can be rented by the day at research institutes, and some students can use this sort of equipment free of charge for their research work.

The availability of the components to be attacked also strongly influences the attractiveness of a particular type of attack. For example, you could attack a card-based electronic purse system either at home, by analyzing your own personal card and its card-specific keys, or at the system level by trying to analyze a security module with its system-wide master keys. The problem with the latter approach is that access to the security module is protected by multiple security measures.

This is by the way why Smart Cards for pay television are so strongly exposed to attack. The attacker can work undisturbed in his own living room, studying the communications and behavior of his Smart Card in order to try to duplicate them using a computer or a home-made electronic circuit. Nobody observes him or interferes with his work. However,

if he were to attempt to do the same thing with a Smart Card terminal in a supermarket, the cashier would immediately forbid any further experiments and thus interrupt his work. A good review of the subject of security of electronic money with and without Smart Cards can be found in [BIS 96].

The final criterion, which is of decisive importance, is naturally the value of the reward for the attacker's troubles. His efforts must be paid back, either in a financial form or in the form of enhanced prestige. From this, it can be concluded that various field trials of electronic purses are only at risk of being attacked by hackers and academic groups. There are far too few locations where the cards can be used, and the businesses are mostly too simple (bakers, kiosks and the like), for any significant amount of money to be obtained from an attack.

8.2.2 Attacks and defense mechanisms during the development stage

Even in the development stage of microcontroller hardware and software for Smart Card operation systems, a wide variety of security measures are employed. Security, like quality, must be taking into account from the very beginning of a development project; it cannot be designed into a product as an afterthought.

With regard to attacks in the development stage, it can be generally said that access to the facilities in question is very difficult and the required level of expertise is very high. The attractiveness of an attack is thus correspondingly reduced. Nevertheless, the potential danger of a successful attack at this stage is significant, since there are very extensive possibilities for manipulating the hardware and software.

8.2.2.1 Smart Card microcontroller development

The development of the hardware for a Smart Card microcontroller takes many months. It involves only a few persons and takes place in access-controlled, supervised rooms within the facilities of a semiconductor manufacturer. The computer systems used for the design of the semiconductor are usually part of an independent network that is isolated from the rest of the world. This prevents any changes to the chip design to be made from outside, and also prevents outsiders from obtaining information about the internal design of the chip.

A very extensive amount of insider knowledge is needed to undertake manipulations to a chip design that would weaken its security, which means that this type of attack is probably very unlikely. In addition, nowadays the design and the protection mechanisms of almost all Smart Card chips are evaluated by independent testing agencies, so that an insider attack would not go undetected. However, it could be advantageous to an attacker to know the exact design criteria and the arrangement of the functional elements on the chip, since he would then know about the protective mechanisms and sensors present in the chip and the scrambling of busses and memories. This knowledge could later be useful to an attacker with regard to physical chip analysis.

Protection: design criteria

There are a number of basic criteria that apply to the definition of the functions of a Smart Card microcontroller. One of these is that the mechanisms for protection against static and dynamic attacks must actually work. Sensors and other protective elements are of little use if they can be too easily defeated or if they do not work in certain circumstances. An example is a sensor on the microcontroller chip that has such a large area that it can easily be destroyed by a needle, after which it no longer can fulfill its protective role.

Exhibit 1017
Page 065 of 102

A very important design criterion, which is different from the criteria used for standard chips, is that absolutely no undocumented mechanisms or functions must be present in the chip (*'that's not a bug, that's a feature'*). Such undocumented features are usually not completely tested, since only a few people know about them, and thus often have various errors and weak points. Since they are not documented, they could unintentionally be overlooked when the hardware is evaluated, and then possibly be used later for attacks. The use of such undocumented features is thus strictly prohibited, even though they can often be very helpful for the developer.

Protection: unique chip number
When the semiconductor hardware is being developed, all hardware security elements must be first defined and then converted into hardware for the resulting microcontroller. One such element, in addition to sensors and protective coverings, is WORM (*write once, read multiple*) memory, which is also referred to as an OTP (*one time programmable*) memory. When the semiconductor chips are manufactured, a unique chip number is written to this memory. This means that each chip is different and can be uniquely traced, and the Smart Cards can later be unambiguously identified within the system. In addition, chip numbers can also be used for the derivation of keys, and they allow the generation of 'blacklists' that can be used to take suspect cards out of circulation.

It should not be overlooked that although these numbers cannot be altered in the original chips, they naturally provide no protection against imitation chips using freely programmable microcontrollers. This means that security measures cannot be based on the presence of a particular chip number in the WORM memory of a particular chip. Such a unique number can only be used as the basis for true cryptographic security mechanisms. For example, a chip number can be used for the derivation of secret keys, which are in turn used in a challenge–response authentication process.

8.2.2.2 Smart Card operating system development

Software for Smart Cards is developed according to modern software development principles. Regardless of which development model is used (waterfall, spiral or whatever), certain general conditions must be observed.

The development computer always requires a separate, completely isolated network that does not allow any external access. The development tools, such as compilers and simulators, are software packages whose proper operation must be verified in dedicated tests. Sometimes two different compilers are even used, to be sure that the results are correct. Using software whose origins are not completely traceable is fundamentally prohibited, since such software would offer a possible means to manipulate development tools and consequently modify the programs to be generated.

Protection: development principles
Just as with hardware development, no undocumented features may be built into the software. For example, it would certainly be possible to convert the laborious black-box tests that are commonly used with Smart Cards into white-box tests by incorporating commands that could be used to read out arbitrary regions of memory. However, if such commands were inadvertently left in the operating system, it would be possible to use them to read the secret keys in real Smart Cards. In order to eliminate the possibility of such an attack, the creation of dump commands is prohibited as a matter of principle, even if their use would save valuable development time.

An additional principle is that programmers should never work alone on a project. This is already forbidden by considerations of software quality assurance, but the 'four eyes' principle must be observed for reasons of security as well. This effectively hinders attacks by insiders, since at least two developers must agree to work together on any attack. In addition, internal source code reviews are performed regularly, which assures the quality of the code and also supervises the development process.

Once the software development is finished, the entire source code and its functions are often inspected by an independent testing agency, as part of a software evaluation.[6] The main reason for performing this time-consuming and costly testing is to check for software errors, but it also has the effect of making it impossible for a developer to hide a Trojan horse (for example) in the operating system. In practice, such things can be found only by reviewing the entire code, since an experienced programmer is certainly able to finds means and techniques to hide a Trojan horse so that it cannot be found by a black-box test.

Protection: dividing knowledge
If several people work on a task, the result will be significantly more resistant to attack, due to the different opinions and experience of the people involved. The principle of dividing knowledge (shared secrets) is the opposite of the idea that 'everybody knows everything about everything'. In the development of security components, complete knowledge of the component should fundamentally never be vested in a single individual, since that person would then be a target for an attack. As in many military realms, knowledge is divided over several individuals in the development stage, so that although it is possible for experts to discuss particular subjects, there is never any single person who knows everything.

A similar situation exists with regard to completing of the Smart Card operating system, in which tables, program code and configuration data are loaded into the EEPROM. In addition to providing increased flexibility, this procedure also has a security aspect. This is because the card manufacturer, who receives the final, assembled ROM code for producing the fabrication masks, does not have complete knowledge of the operating system. The parts of the operating system that are located in the EEPROM are unknown to the chip manufacturer, so he cannot discover the complete security mechanisms and functions of the operating system by analyzing the ROM code.

8.2.3 Attacks and defense mechanisms during the production stage

Attacks during the production of chips or Smart Cards are typical insider attacks, since the production environments are closed. Access is strictly controlled, and every entry is logged. Nevertheless, security measures cannot be dispensed with in the production stage, since some technically very interesting and effective attacks can be carried out in this stage.

Protection: authentication during the finishing steps
Already at the wafer fabrication stage, the Smart Card microcontrollers are individualized using chip numbers and protected using transport codes. With recent operating systems, the transport code is chip-specific, and an authentication is a mandatory requirement for each access in the finishing process. Although this increases costs and the amount of time required to finish the chips, and naturally requires a security module for every machine, it considerably increases security.

---

6 See also Section 9.3, 'Evaluating and Testing Software'.

Exhibit 1017
Page 067 of 102

An obvious type of attack in the finishing process is to feed in dummy chips or dummy Smart Cards, which behave the same as genuine components but which for example include a 'memory dump' command. The earliest opportunity to exchange a dummy chip for a real chip is of course after the wafer has been divided into individual dice. This type of attack can be illustrated on the basis of a Smart Card for digital signatures.[7] The attacker swaps a dummy Smart Card for a genuine Smart Card at the initialization stage. This card is then initialized with genuine data and afterwards personalized. Since this Smart Card has all the functions of a real Smart Card, the process for generating the key for the asymmetric cryptographic algorithm will also be executed by the microcontroller. It obtains the necessary data for this from the initialization and personalization data. After this, the attacker must manage to recover possession of this card, and then he can read the secret signature key from the card using his special dump command. Since the associated public key has been signed by the trust center and is thus confirmed to be genuine, the attacker now knows everything necessary to produce as many duplicate cards as he wishes, all of which will be recognized as genuine.

This attack is unrealistic, since administrative measures are taken to prevent chips and Smart Cards from being taken into or removed from the finishing stations. In addition, mandatory authentication between the Smart Card and the security module of the finishing machine before every finishing step makes it difficult to swap chips or cards.[8]

8.2.4 Attacks and defense mechanisms when the card is in use

Access to the component to be attacked – the Smart Card – is usually much easier for the attacker after the Smart Card has been issued than in the previous phases of its life cycle. This is why the probability of attack is relatively high when the card is in use.

The idea of a self-destroying Smart Card microcontroller appears again and again in many publications, as a sort of panacea against all sorts of attackers. There are hardware security modules, such as those used for military applications, in which such mechanisms are sometimes employed, but such a defensive measure is not allowed in Smart Cards for a multitude of reasons. When it is without power, a Smart Card first of all has no way to recognize a potential attack, and there is also no possibility of any sort of active defense mechanism, since the Smart Card does not have any reserve source of energy. This is all aside from the fact that true self-destruction probably could not be imposed on cardholders, for purely legal reasons. Who would be responsible for the losses or damage that could occur under unfavorable circumstances, just because a Smart Card has incorrectly destroyed itself? In addition, true self-destruction is not at all necessary, since in almost all cases it is sufficient to erase the secret keys stored in the card.

There is yet another aspect to this subject, which relates to erasing keys or blocking Smart Cards. It is very difficult for a Smart Card to even recognize that it is being attacked. There is simply no sensor that can report 'Attack! Erase everything!' Too low a voltage or too high a clock rate could for example be a sign of an attack, but these conditions also occur in normal operation due to unfavorable ambient conditions. Contaminated contacts result in a high contact resistance and thus a low operating voltage. An excessive clock rate

---

7 See also Section 13.6, 'Digital Signatures'.

8 An extensive and detailed description of the usual cryptographic process for the initialization and personalization of Smart Cards is contained in Section 10.4, 'Phase 3 of the Life Cycle in Detail'.

can occur at a Smart Card terminal that is intended to be used with higher clock rates. Since recognizing an actual attack is so difficult, and usually not even possible, automatic mechanisms for blocking the card or erasing the keys are usually not employed.

In the following section, some attacks that can be considered to be 'classic' are described and explained. The descriptions of the attacks can be said to represent the 'state of the art'. They are intended primarily to provide people who are inexperienced in the subject of Smart Card security with a reasonably solid basic understanding, so that they will not in their ignorance reuse mechanisms that are already known to be vulnerable. The described defense mechanisms can be used to counter these attacks. These in turn can be avoided by somewhat modified attack scenarios, which results in the well-known cat and mouse game of measures and countermeasures for attacks and defenses.

The scenarios presented here are not an invitation to break the security of a Smart Card system, since they are without exception well known and also public. They do not represent any serious threat to the security of any modern, present-day Smart Card system, since they have long since been dealt with by suitable defense mechanisms. However, a few years ago, it is possible that a certain amount of success could have been achieved.

The attacks are divided into those that are directed against the chip hardware and those in which an attempt is made to break the Smart Card system at the logical level. The physical attacks and analysis methods can also be subdivided into static and dynamic attacks. In a static analysis, the chip is not operating, but it may be electrically powered. In a dynamic analysis, which is much more difficult to carry out, the chip operates with its full range of functions during the analysis.

8.2.4.1 Attacks on the physical level

Manipulations at the semiconductor level require a large amount of technical effort and expense. Depending on the attack scenario, the equipment required can include a microscope, a laser cutter, micromanipulators, focussed ion beams, chemical etching installations and very fast computers for the analysis, logging and evaluation of the electrical processes in the chip. This equipment and the knowledge of how to use it are available to only a few specialists and organizations, which strongly reduces the probability of an attack on the physical level. Nevertheless, a card or semiconductor manufacturer must assume that a potential attacker could employ all the equipment and facilities needed for such an attack, which means that suitable protection must be built into the hardware.

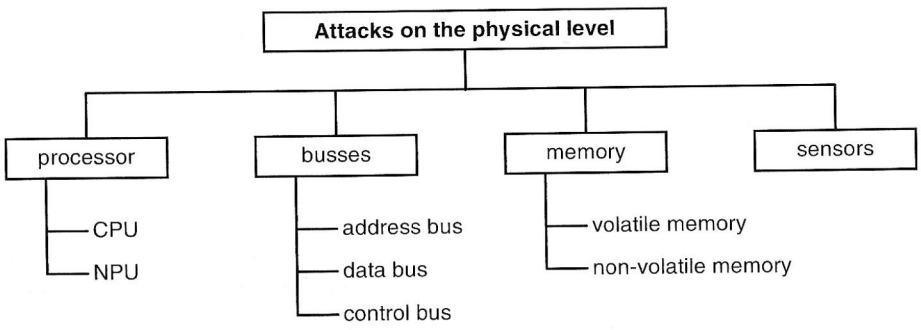


Figure 8.20 Classification of the points of attack at the physical level on a Smart Card microcontroller

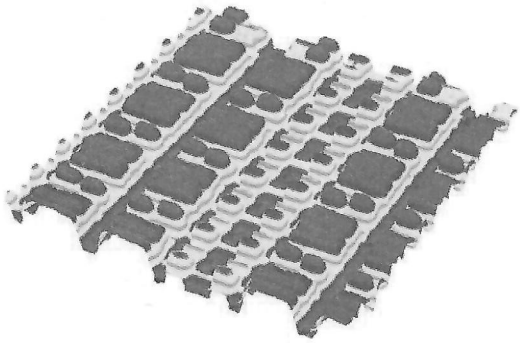


Figure 8.21 Graphic representation of the surface profile of a Smart Card microcontroller, as measured using an atomic force microscope. The maximum surface relief in this illustration is only 2.3 µm (Source: Giesecke & Devrient)

In order to conduct an attack on the physical level, a few preliminary steps are necessary. The thing that has to be done is to remove the module from the card, which can easily be done using a sharp knife. After this, the epoxy resin must be removed from the chip. Anderson and Kuhne [Anderson 96b] used fuming nitric acid for this with an infrared lamp as a heat source, followed by an acetone rinse to clean the chip. After this, the semiconductor chip is free and still fully operational. Many people think that the chip now lies unprotected before them and only has to be 'read out', but this is by no means so. An attacker still has to work through a manifold of security measures before he can gain access to the secrets.

The protective measures in the hardware can be divided into passive and active elements. The passive elements are based directly on the techniques used in semiconductor manufacturing. They include all processes and options that can be used to protect the memory region and the other functional parts of the microcontroller against various types of analysis.

There is a full spectrum of active elements available on a silicon chip to complement the passive possibilities offered by the semiconductor technology. Active protection means the integration of various types of sensors into the silicon crystal. These sensors are polled and evaluated by the Smart Card software as needed. This is naturally only possible when the chip with all its supply lines is fully activated. A chip without electrical power cannot measure any sensor signals, let alone evaluate them. Consequently, the line between useful protective elements and technical gadgetry is frequently particularly thin where sensors are concerned. A light-sensitive sensor that is supposed to prevent an optical analysis of the memory will not respond if the chip is located on the object carrier of an optical microscope without power or a clock signal. In addition, it is very easy to visually identify such a sensor on the chip surface and cover it with a drop of black paint, so that its protective function can easily be neutralized even when the chip is operating.

Long-term functional security is also important. For example, a temperature sensor that causes the Smart Card software to erase the entire EEPROM in response to a brief and non-damaging overheating of the chip, does not by any means contribute to increased functional security or security against an attack. For this reason, only a few sensors are normally employed in Smart Card microcontrollers.

Exhibit 1017
Page 070 of 102

In the following discussion, we explain the protective mechanisms of Smart Card microcontrollers that are the most important and the most often used in practice.

Static analysis of Smart Card microcontrollers

Protection: semiconductor technology

The dimensions of chip structures (lead widths, transistor sizes and so on) approach the limit of what is currently technically possible. The usual structural widths lie in the range of 1 to 0.35 µm, which is in itself no longer technically unusual. However, the density of the transistors on the surface of the silicon belongs to the highest level that can currently be achieved using standard lithographic manufacturing processes. These very fine structures alone make it nearly impossible to extract any information from the chip using analytic procedures, for which reason semiconductor technologies with 1-µm structure sizes are currently considered to be secure. This dimension is sure to be reduced in the future.

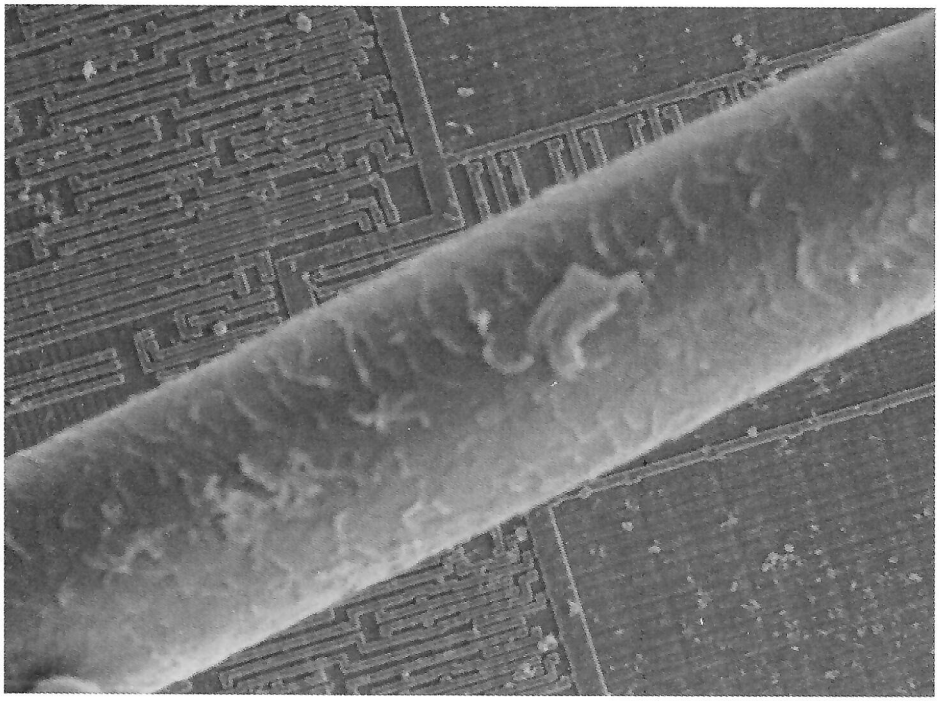


Figure 8.22 Photograph of a human hair in comparison with semiconductor structures of a Smart Card microcontroller, magnified 1000× (Source: Giesecke & Devrient)

Protection: chip design

'Standard cells' are frequently used for the design of semiconductor integrated circuits. These can contain the core elements of a processor or a particular type of memory. The advantage of using standard cells is it allows a semiconductor manufacturer to quickly produce a variety of different types of chips at a high quality level. This process, which has been developed for mass-produced components where security is not an issue, is not

Exhibit 1017
Page 071 of 102

allowed to be used for Smart Card microcontrollers. This is because the designs and functions of standard cells are known, and their use would thus provide a potential attacker with too much information and thus considerably simplify his task. The functional elements of Smart Card microcontrollers are developed especially for this application and are not used for any other purpose.

Protection: dummy structures
The use of dummy structures on the chip is a measure that is the subject of frequently controversial discussions among experts. Dummy structures are elements of the semiconductor that do not have any actual function, but instead are intended to confuse and mislead an attacker. The associated security is based purely on keeping the existence and locations of such structures secret. Dummy structures can also be monitored, so that any changes to them can be detected and can cause the chip to switch off. The main disadvantage of dummy structures is the additional room that they occupy on the chip.

Protection: busses on the chip
All internal busses of the chip, which connect the processor to the three different types of memory (ROM, EEPROM and RAM), are not brought to the outside of the chip. This means that it is not possible to directly make connections to these busses. It is thus not possible for an attacker to tap into the address, data or control bus of the microcontroller or to influence the bus signals in order to read out the memory contents. In addition, the busses on the chip are scrambled in a manner that is either static or chip-specific, so that the functions of the individual bus lines cannot be recognized from the outside.

Protection: memory design
The memory medium used for most programs is the ROM. The contents of ROM that is generally used in the industry can be read bit by bit using an optical microscope. It would not be particularly difficult to assemble these bits into bytes and then arrange these bytes to obtain the complete ROM code. In order to prevent exactly this type of analysis, the ROM is not located in the top level of the chip, which is the most easily accessible layer. It is instead located in the lower layers of the silicon. This impedes an optical analysis.

However, if the chip were to be glued to a carrier upside down and the rear surface were then ground off, it would be possible to read the contents of the ROM. For this reason, only ion-implanted ROM is used in Smart Card microcontrollers, since the contents of such a ROM cannot be seen using either visible or ultraviolet light. This also largely protects against selective etching, which is a process that can be used to attempt to etch the semiconductor to make the contents of the ROM visible again.

Protection: protective layers
The analysis of the electrical potentials on the chip while it is operating represents a threat. With a suitably high sampling resolution, this technique can be used to measure charge potentials (voltages) on very small regions of the crystal. With this information, it is possible to draw conclusions about the contents of the RAM while the chip is operating. This analysis can be very effectively prevented by placing current-carrying metallization layers on top of the memory cells in question. If these metallic layers are removed by chemical etching, the chip will no longer operate properly, since these layers are needed to distribute the supply voltages that the chip needs in order to function.

In addition, the chip can be fabricated with meandering current-carrying structures on top of the entire chip or on top of regions that need special protection, such as an

underfrequency detector. The resistances of such structures can easily be monitored, or they can be incorporated into functional parts of the chip, such that the chip immediately stops working if they are damaged.

It is also conceivable to use opaque protective layers whose integrity is continuously monitored by phototransistors, which are easily implemented in semiconductor devices. As soon as such a layer is removed, this would immediately be detected and the chip could refuse to operate any further.

Attack and defense: reading out the volatile memory

As is well known, a RAM loses its data contents when its power supply is cut off. However, this does not occur if the memory cells are cooled to a temperature of –60 °C. Also, the content of the RAM is not necessarily lost if the stored data remain unchanged for a long time. The background of this effect is described in a paper by Peter Gutmann on the subject of the secure erasing of memory media [Gutmann 96]. Consequently, secret keys are not held in RAM any longer than is absolutely necessary, following which they are immediately erased or overwritten with other values. This minimizes the risk that traces of secret keys may be left in the RAM cells, and weakens attacks based on fixing the RAM contents by freezing or burning.

Reading out RAM cells is very difficult, since it requires detecting the switching states of the transistors involved. However, it is certainly possible to extract stored data from RAM cells using sophisticated electron microscopes and special contrast-enhancing processes. A prerequisite for this is the removal of the passivation layer and the metallization layers underneath the passivation layer, which protect the RAM against exactly this type of attack. Removing the metallization layers unavoidably causes the RAM cells to be destroyed, since part of their function in included in these layers.

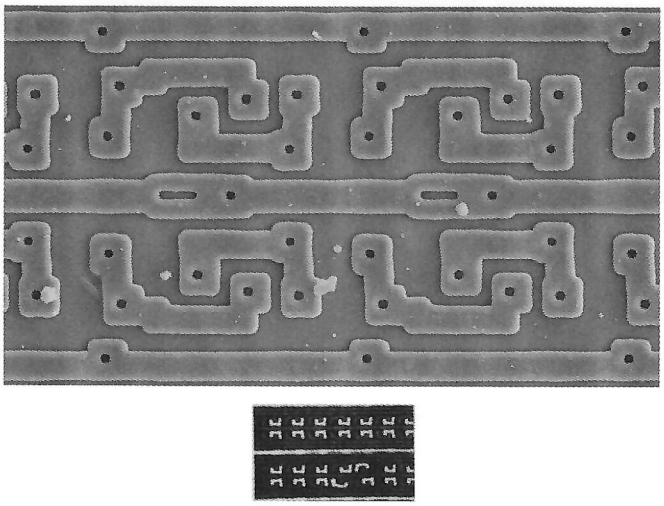


Figure 8.23 Photograph of several RAM cells magnified 3000×, without protection by means of additional metallization layers. The lower picture shows the electrical potentials of the same RAM cells, as measured using an electron beam tester with the chip in operation. The distribution of zeros and ones over the RAM can be clearly recognized. (Source: Giesecke & Devrient)

Exhibit 1017
Page 073 of 102

Protection: memory scrambling

Scrambling the memory on the microcontroller chip, which is similar to the well-established practice of scrambling the busses, is being used increasingly often. The security of this technique is based on the secrecy of the scrambling scheme for the memory cells. Memory scrambling is easily implemented and does not need much additional space on the chip. Without the relevant scrambling information, it is extremely difficult for an attacker to discover how the memory cells are actually addressed.

The EEPROM can also be scrambled using software. However, this requires complicated programming, and all write accesses must be protected by making them atomic processes, since otherwise the system would be very vulnerable to the sudden removal of the supply voltage. Software memory scrambling does however have the advantage that it can be made chip-specific and even dynamic, so that the memory contents can be redistributed within the memory in the course of a session.

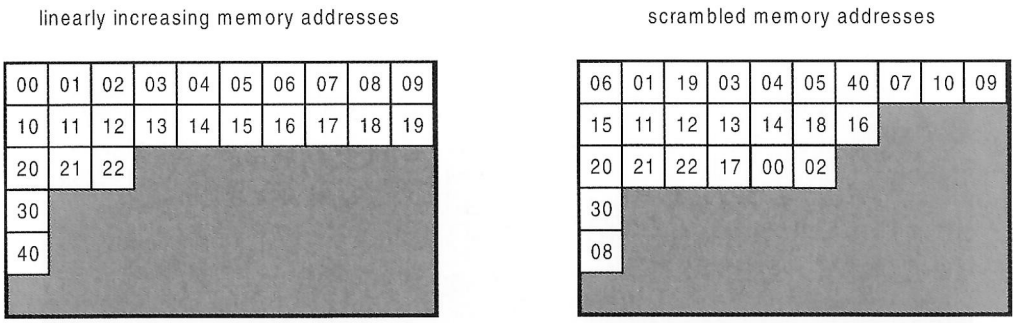


Figure 8.24 Comparison of a conventional semiconductor memory with a scrambled memory used in a Smart Card microcontroller

Attack and defense: encrypted PIN storage

This example of an attack at the physical level is actually a textbook case of an ineffective defensive mechanism. The basic idea of this defense is that if it is possible to read out the EEPROM of a Smart Card, an attacker should at least be prevented from learning the value of the PIN in this manner. There are naturally preventive measures that can be taken. The PIN need only be encrypted using a one-way function, and the result stored in the EEPROM as the reference value for a PIN comparison. A card-specific key could be used for the one-way function, so that the reference data for two identical PINs would be different in two different cards.

If it were now possible to read the reference data out of the EEPROM, it would appear to be impossible to derive the PIN from these data. However, a clever attacker would not even choose such an approach. If normal PINs consisting of four decimal digits are used, the number space of the PINS has a lower boundary of '0000' and an upper boundary of '9999'. This means that the number of possible PINs is exactly 10,000. If the attacker can read the entire memory of the Smart Card, he can also read the one-way function and its associated card-specific key. With this information, he sets about encrypting all possible PINs using the one-way function. After an average of 5000 attempts, he will have obtained a result that matches the reference value in the Smart Card, which means that he knows the

PIN. As can be seen, in this case using a one-way function to store the PIN does not provide any significant advantage.

Dynamic analyses of Smart Card microcontrollers

Protection: passivation layer monitoring

A passivation layer is placed on top of the microcontroller in the silicon at the conclusion of the fabrication process. This layer impedes oxidation (due to atmospheric oxygen) and other chemical processes at the surface of the chip. The passivation layer must always be removed before any sort of manipulation of the chip can be carried out. It should be borne in mind that although it is possible to chemically remove the passivation layer, the chip is then exposed to a large risk of oxidation, which can destroy it relatively quickly. A sensor circuit can employ resistance or capacitance measurements to determine whether the passivation layer is still present. If it is missing or damaged, this can either trigger an interrupt to the chip software or cause the complete hardware of the chip to be shut down, which reliably prevents any sort of dynamic analysis.

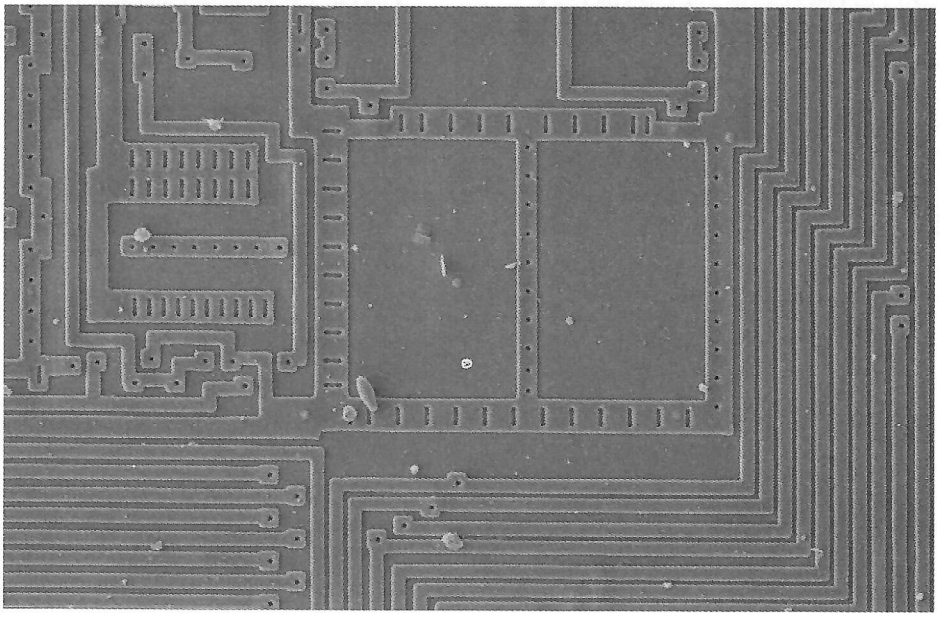


Figure 8.25 Photograph of a passivation layer monitor at 1000× magnification. The passivation layer detector consists essentially of the two rectangular semiconductor elements, and it uses capacitance measurements. (Source: Giesecke & Devrient)

Protection: voltage monitoring

A voltage monitor is present in every Smart Card microcontroller. It provides a well-defined shutdown of the IC in case the supply voltage exceeds its allowed lower or upper limits. This gives the software the assurance that it is not possible to operate the chip in marginal regions, in which the chip does not function properly. If such a voltage monitor were not present, it could for example happen that the program counter becomes unstable

Exhibit 1017
Page 075 of 102

when the chip is operated outside of its specified supply voltage range, which in turn could lead to uncontrolled jumps within the program or simply to processing errors. This faulty behavior could be used as the basis for determining secret keys by using the technique of differential fault analysis (DFA), which is described elsewhere in this book.

Another sensor that is partly based on the voltage detector is the power-on detector. This detector, which is also present in all chips, recognizes a power-on condition independently of the reset signal, and ensures that the chip is always placed in a defined initial state when power is first applied. The reasons for doing this are similar for those for voltage monitoring.

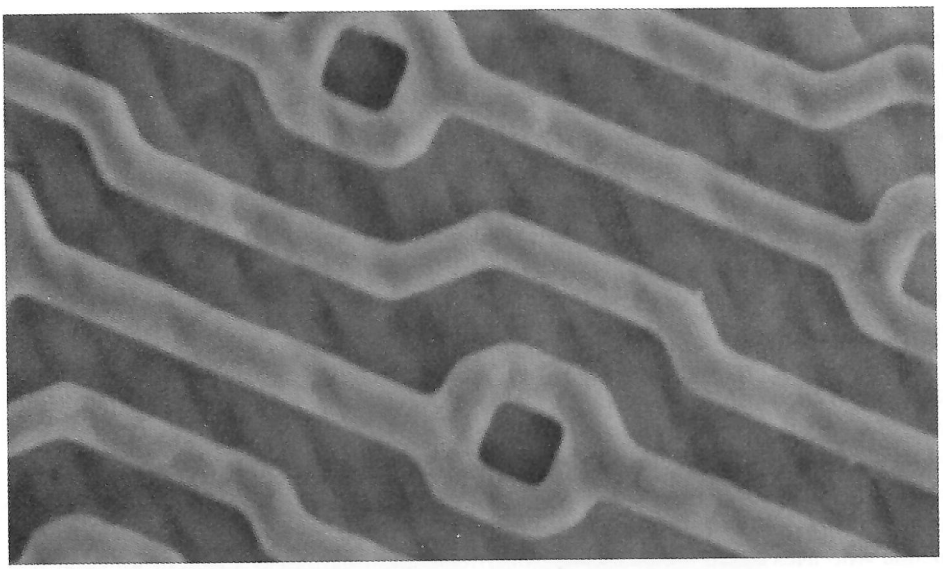


Figure 8.26 A modern passivation layer monitor at 8000× magnification. The track separation is 4 µm. This detector presumably uses resistance measurements. (Source: Giesecke & Devrient)

Protection: frequency monitoring

A Smart Card is always driven by an external clock, so its processing speed is completely determined outside the card. This means that, at least in theory, it is possible to run the microcontroller in single-step mode. This would provide outstanding opportunities for analyzing its operation, in particular with regard to measuring the operating current consumption (power analysis) and electrical potentials on the surface of the chip. In order to prevent such attacks, an underfrequency detector is built into the chip. This eliminates the possibility of reducing the clock rate to unallowed levels. The minimum clock rate given in most specifications is 1 MHz. However, the underfrequency detector has a wide tolerance range for technical reasons, so the chip usually stops working at around 600 kHz. This assures that the chip can will always work at the minimum specified clock rate of 1 MHz. Overfrequency detectors are also sometimes used, but nowadays the hardware is frequently constructed such that the chip cannot be used if the clock rate is too high.

Exhibit 1017
Page 076 of 102

In order to protect the microcontroller against the dangers of single-step operation, it is naturally necessary to secure the underfrequency detector with protective layers, so that any attempt to manipulate the detector will be noticed.

Protection: temperature monitoring
A temperature sensor is used in some types of chips, but the benefit of such a sensor is controversial. The chip is not damaged if the temperature briefly rises above the specified maximum value, and this does not in itself represent an attack. Shutting down the chip in this marginal situation, however, could result in an artificially increased failure rate, while it does not provide the operator of the Smart Card system with any additional security.

Protection: bus scrambling
In many Smart Card microcontrollers, the internal busses that drive the memory (and are not externally accessible) are scrambled. This means that the individual bus lines are not laid out next to each other in increasing or decreasing order, but are instead arranged randomly next to each other and 'swapped' several times, or even arranged in several layers on top of each other. This represents an additional hurdle for a potential attacker, who thus does not know which bus line is associated with which address bit or function.

Scrambling the bus lines was originally introduced only in a static version, with the same scrambling scheme used on every chip. This means that it would probably not be all that difficult for an attacker to discover the scrambling scheme over a moderate length of time, and thus be able to take it into account when tapping the busses.

The security provided by this technique can be improved by using chip-specific scrambling. This is naturally not achieved by using a different set of exposure masks for the busses of each chip, since this is currently either not technically possible or not affordable. Instead, scrambling is performed by randomizer circuits located just ahead of the memory. These can be driven by the chip serial number, for example. This technique is not difficult in terms of semiconductor technology, and it makes life considerably more difficult for someone who tries to tap the bus. Chip-specific scrambling that is different for each session would also be possible if a variable initial value is used for the randomizer.

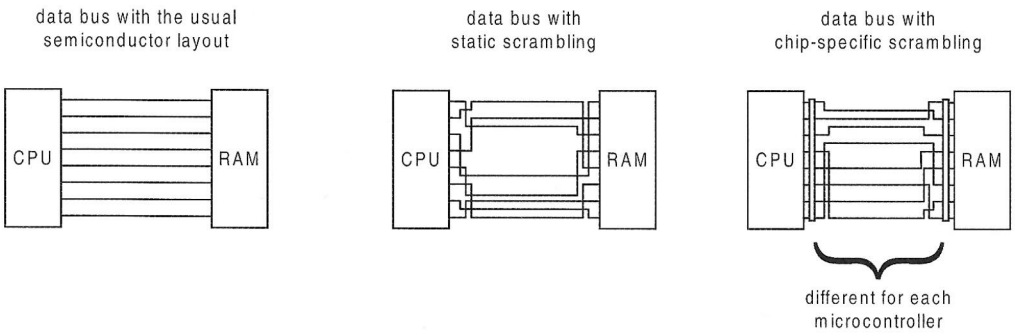


Figure 8.27 Bus scrambling in a Smart Card microcontroller illustrated using an 8-bit data bus between the CPU and the RAM. The data bus lines shown here represent information flows rather than electrical leads.

Exhibit 1017
Page 077 of 102

Protection: irreversible switching from the test mode to the user mode
All microcontrollers have a test mode that is used for verifying the chips during the fabrication process, and for executing internal test programs while the semiconductors are still in the wafer or after they have been packaged in modules by the manufacturer. The test mode allows types of access to the memory that are strictly forbidden when the chips are later in actual use. For technical reasons, however, it is an unavoidable requirement to be able to read data from the EEPROM in this mode.

The change from the test mode to the user mode must be irreversible. This can be realized by using a polysilicon fuse on the chip. In this case, a voltage is applied to a test point on the chip that is provided for this purpose, and this voltage causes the fuse to melt through. The chip is thus switched into the user mode using hardware. Normally, this cannot be reversed. However, a fuse is by its nature a relatively large structure on the surface of the chip. It is conceivable that the fuse could be mechanically bridged after the passivation layer has been partially removed where it covers the fuse. This would put the microcontroller back into the test mode, and the memory could be read out using the extended access options available in this mode. If the complete content of the memory is known, it is then easy to clone the Smart Card that has been read out.

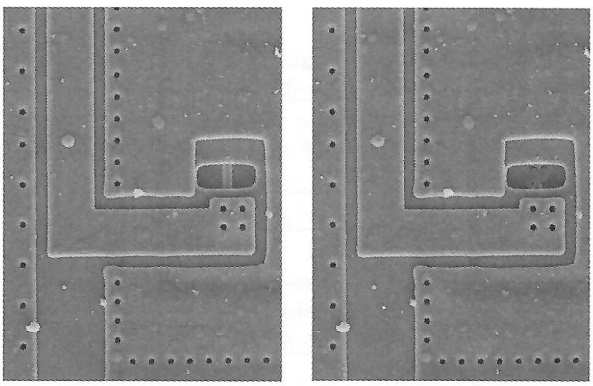


Figure 8.28 Photograph of a polysilicon fuse magnified 2000×. The picture on the left shows a fuse that is still intact, while that on the right shows a blown fuse. (Source: Giesecke & Devrient)

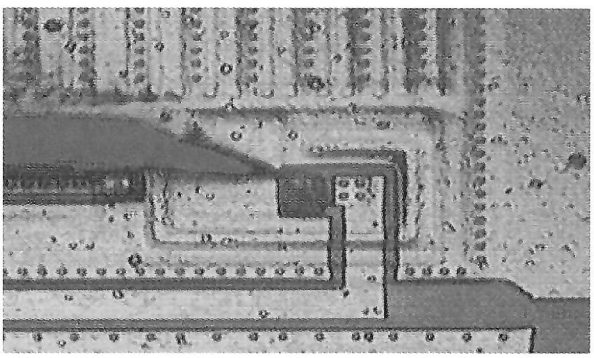


Figure 8.29 Photograph of a polysilicon fuse together with a microprobe needle, magnified 500×. A blown fuse could be bridged using a microprobe needle. (Source: Giesecke & Devrient)

In order to defend against this type of attack, most semiconductor manufacturers have adopted the practice of reserving a portion of the EEPROM for the switchover mechanism, in addition to using a fuse. If a certain unalterable value is located in this part of the memory, the chip has been irreversibly switched to the user mode. Even if the fuse is bridged over, the chip will not return to the test mode, since the additional logical switch in the EEPROM prevents this.

The security of the switchover from the test mode to the user mode can be increased even further by a very simple measure. The microcontroller chip is laid out on the wafer such that the test pads needed for making contact with the chip for performing the tests are simply sawed off when the wafer is divided into individual dice. In this case, neither a fuse nor any EEPROM cells are needed to switch between the modes, since the elements needed for the test mode are no longer present. In the same way, it would also be possible to replace the fuse that switches from the test mode to the user mode by a track that is irreversibly broken when the dice are sawed from the wafer. With present-day technology, it is not possible to make a connection to a sawn-through trace on the edge of a chip.

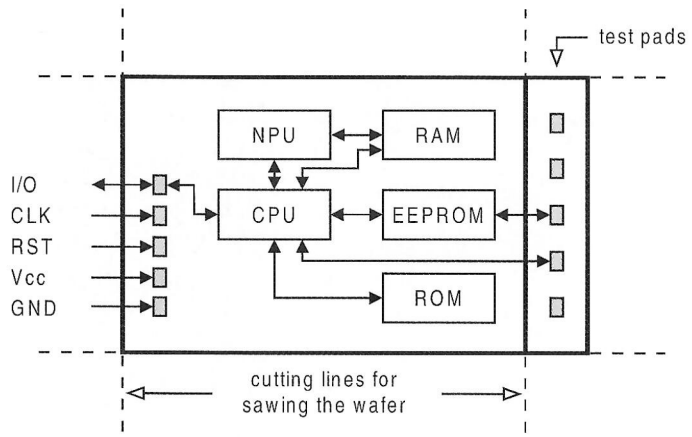


Figure 8.30 One of several possible ways to irreversibly remove the test pads used for testing the CPU and memory of a Smart Card microcontroller

Dynamic analysis and defense: tapping the memory busses of the microcontroller

Before the busses between the CPU and the memories of the microcontroller (ROM, EEPROM and RAM) can be tapped, the chip must be exposed and the passivation layer on the top surface of the chip must be removed. The passivation layer protects the chip against oxidation on the one hand, but it also protects the chip against attack, since its integrity is monitored by sensors. According to Anderson and Kuhn [Anderson 96b], it can be removed by etching with hydrofluoric acid. A laser cutter[9] can be used just as well to selectively cut openings in the passivation layer at the necessary locations.

After the passivation layer has been removed from the entire surface of the chip, or only from selected locations, it would be at least theoretically possible to make contact with the

9 A laser cutter is a device for drilling and cutting using a high-power laser beam. It has an accuracy of a fraction of a micron.

Exhibit 1017
Page 079 of 102

address, data and control busses for the memory using microprobe needles. If it is possible to make electrical connections to all the lines of these three busses, it is very easy to address the individual memory cells and to read any desired regions of the ROM and EEPROM. The chip does not have to be powered for this, and any desired type of connection jig can be used. The consequences of a successful attack using this method would be serious, since in principle it would make all the secret data in the non-volatile memory readable.

This method could be extended by making connections to the busses and then operating the chip in the normal manner. In this way, it would be possible to eavesdrop on the complete data traffic between the CPU and the memories, and this could be recorded using a sufficiently fast logic analyzer.

As already indicated, it is very difficult to make electrical contact with the individual traces on the chip. The number of connections needed for this attack is 16 for the address bus, 8 for the data bus and 1–4 for the control bus. In total, at least 25 simultaneous connections would have to be created between an external analysis computer and the traces on the chip. Even with modern micromanipulator technology, this is currently not possible, due to the very small dimensions of the semiconductor structures. However, it would be possible to use a focused ion beam generator, which is a tool that is commonly used in the semiconductor industry, to implant a sort of electrically conductive contact surface for each bus line. These then could be used as contact point for a microprobe needle. The effort required for this is however enormous.

Even if an attacker succeeded in making these connections, he would still have to determine how the busses have been scrambled before he could read the data successfully. This is because the individual bus traces are not arranged on the chip in an orderly fashion next to each other, but are instead arranged in an externally unrecognizable manner.

If markedly improved technology in the future should make it possible to make connections to the busses of current microcontrollers, this would probably not have any affect on security, since by that time the semiconductor structures will have become significantly finer than they presently are. In addition, micromechanical technology will probably always lag behind that of semiconductor technology, which is based on optical processes. This means that, even in the future, this sort of attack will probably not by suitable for significantly weakening the security of Smart Cards.

Dynamic analysis and defense: measuring the current consumption of the CPU

Already in 1995, in the first edition of this book, the following statement appeared at this point: 'The design of the processor is also critical with regard to security. A Smart Card processor must have nearly the same current consumption for all machine instructions. Otherwise, conclusions can be drawn regarding the instruction being processed, based on the current consumption. A certain amount of secret information can be deduced from these conclusions.' The fact that conclusions can be drawn about the instructions being executed by a processor, and even the data being processed, by analyzing the current consumption of the processor while the instructions are being executed, was thus already well known for years when Paul Kocher, Joshua Jaffe and Benjamin Jun published a paper on simple power analysis (SPA) and differential power analysis (DPA) in June of 1998 [Kocher 98].

The working principle of simple power analysis is relatively uncomplicated. The current consumption of a microcontroller is determined by measuring the voltage drop across a resistor connected in series with the power supply. The measurements are made with high time resolution using an analog-to-digital converter. With a high-performance processor, such as a Pentium or a PowerPC, no conclusions could be drawn about the instructions

being executed, due to the complexity of the internal processes. The relatively simple designs of the 8051 and 6085 CPUs used for Smart Card microcontrollers, on the other hand, result in measurable and thus interpretable variations in the current consumption, depending on the instructions and data that are being processed.

Differential power analysis can reveal even finer differences in the current consumption of a microcontroller than can simple power analysis. With the DPA technique, the current consumption is first measured while the microcontroller is processing known data and then while it is processing unknown data. The measurements are repeated many times in order to eliminate the effects of noise by taking average values. The differences are calculated once the measurements have been completed, and conclusions regarding the unknown data are drawn from the results.

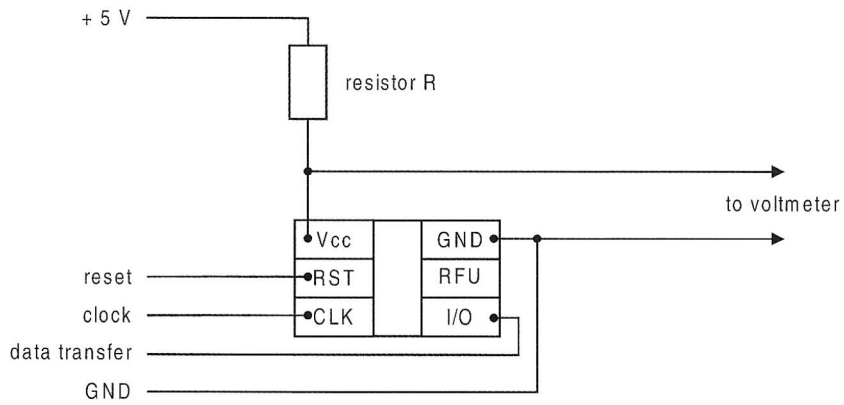


Figure 8.31 Circuit diagram of the connections to a Smart Card microcontroller needed to make simple current measurements using a series resistor

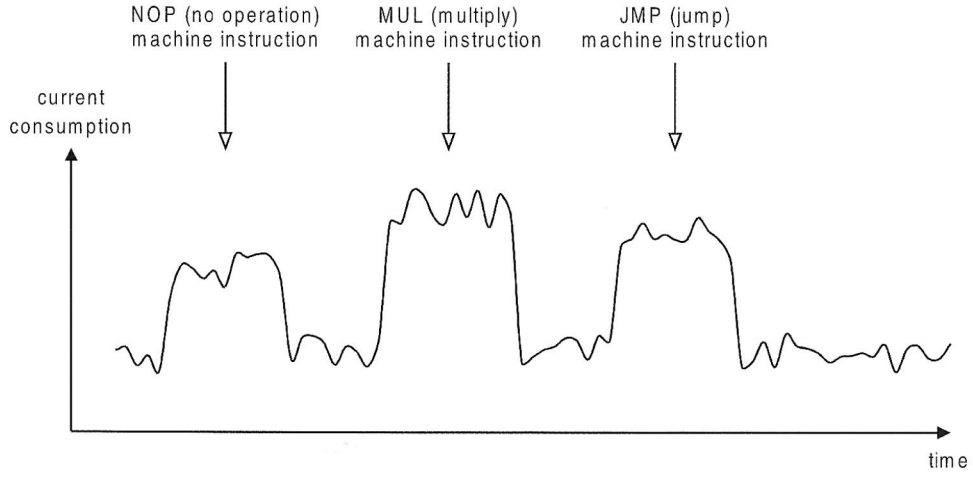


Figure 8.32 Simplified representation of variations in the current consumption of a Smart Card microcontroller with the execution of several different machine instructions. The current consumption may depend on both the data being processed and the machine instruction being executed.

Exhibit 1017
Page 081 of 102