Security Evaluation Schemas for the Public and Private Market with a Focus on Smart Card Systems

Eberhard von Faber

debis IT Security Services, Rabinstraße 8, D-53111 Bonn, Germany
e-vonfaber@itsec-debis.de

Abstract. Even users must have some understanding of the different evaluation schemas. They must be able to rate the outcomes they rely on and use the opportunities to steer the processes. Some evaluation schemas are designed for general purposes others for specific application contexts. The elements of evaluation schemas are introduced first. Then observations about smart card evaluations are discussed demonstrating that the evaluation or approval process itself effects the evidence of the assurance and the value of evaluation verdicts. Especially trade-off situations typical of smart card evaluations are discussed.

Table of Contents

1 Introduction

Modern companies use information technology more and more to support their traditional business activities, to offer them in a better way or to more customers. The commercial goals of a company can only be reached if the information technology operates perfectly. Nowadays information is a critical resource that enables companies to succeed in their business. Therefore, many products and systems provide security functions exercising proper control of the information. Companies

Ç.K. Koç and C. Paar (Eds.): CHES'99, LNCS 1717, pp. 187-203, 1999.

and the individuals using such products expect that the sensitive information remains private and that unauthorised modifications are detected.
It is key to know (i) whether the products and systems actually and properly respond to the security needs in a specific application context and (ii) whether they provide a sufficient level of protection. Customers usually do not wish to rely solely on the promises given by the product vendor. Therefore, IT-products or systems which already exist or which are still in the development stage are to be evaluated by *independent specialists* to proof if and to what extend the security objectives are met. In addition, the developer himself often likes to have some proof and indication how to improve his solution.
Security assessments (evaluations) are sub-contracted to independent (third) parties (Evaluation Facilities or labs). They have to have the knowledge, expertise and resources necessary to judge whether the product or system is "secure". The Evaluation Facility is eventually expected to pass a verdict. For this it is required to formulate the *question* the lab has to answer as precise as possible. If the question or the definition of the Security Target is hazy the evaluation result will not be very helpful. For that reason international evaluation criteria specify requirements for content and presentation of the Security Target.

Too little attention is given to the fact that the evaluation or approval process itself effects the evidence of the assurance and the value of evaluation verdicts. The evaluation schema, its structure and rules, the criteria and their evaluation methodology actually effects the outcome of an evaluation and the information given to the user. There are different evaluation schemas being designed for general purposes or for specific application contexts. After having introduced the elements of an evaluation schema trade-off situations typical of smart card evaluations are discussed. Studying these examples give valuable information to individuals, companies and organizations using evaluated products.

1 Evaluation Schemas

According to almost all security evaluation schemas three parties are involved when an evaluation is being carried out:

- developer and manufacturer of the product or system,
- Evaluation Facility, and
- Overseer (Certification Body or Approval Authority).

The developer and manufacturer applies for a security certificate (to be used in the public market) or an approval (for a closed private market). He has to provide information about all the construction details allowing the Evaluation Facility to assess the security provided. The Overseer (Certification Body or Approval Authority) monitors the evaluation activities, reviews and analyses the evaluation report(s) to assess conformance to the evaluation criteria, the evaluation methodology, and the evaluation schema. Finally, the Overseer issues the certificate or the approval. The decision made by the overseer is based upon the evaluation report prepared by the Evaluation Facility.

In the public market the developer/manufacturer wants to demonstrate to customers that they can have confidence in the security provided by the product or system. The certificate issued by the Overseer (Certification Body in this context) confirms that the evaluation has successfully been performed according to the criteria. The users' decision whether to use the product is additionally based on information given in the Certification Report which contains the non-confidential evaluation results (major findings) and perhaps extra guidelines for operational use.

In the private market the developer/manufacturer wants to get the approval that the product or system can be used in the specific application context given. For instance in banking applications (like the POS debit electronic cash system which uses the eurocheque card in Europe and many smart card based electronic purse systems like GeldKarte in Germany) the card issuers require a successful evaluation before the component can be sold to the banks or payment processors and used in the payment system.

In fact approvals are often used for marketing in other markets since they demonstrate that the developer/manufacturer is able to provide high-quality products. But unlike in public markets the evaluation results are to be accepted by specific institutions. To some extend this changes the security evaluations and the co-operation of the parties being involved in many ways.

Regardless of the market or evaluation schema, the Evaluation Facility uses the same set of information to perform the assessment. This is visualized in Figure 1.

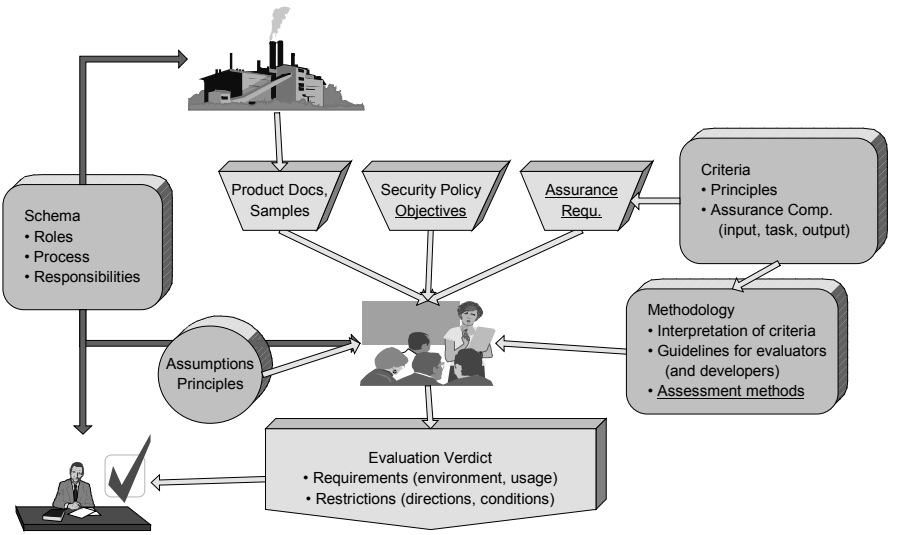

Fig. 1. Evaluation Process as seen by the Evaluation Facility

The basic input for the Evaluation Facility is as follows: (i) All the information about the product's construction is provided by the developer. He also provides samples of the product for penetration and testing. (ii) The Evaluation Facility needs to have a list of the security objectives since the product is checked whether to meet the security

objectives defined for the context given. (iii) The assurance requirements define on what grounds evidence can be given that the product meets its security objectives.
The assurance requirements are defined in the evaluation criteria. One can find them in the "Information Technology Security Evaluation Criteria (ITSEC)" [4], in the "Common Criteria" (Part 3: Security Assurance Requirements) [3] and in the "Trusted Computer System Evaluation Criteria (TCSEC)" [7] as well. When working on the different assurance aspects the Evaluation Facility uses guidelines describing the evaluation methodology (for instance [5] and [6]).
The relations between the three parties are described in a document called Evaluation Schema. The Evaluation Schema defines the process together with the roles and responsibilities of the three parties. Additional regulations may be given in form of assumptions or principles if needed in a specific application context.

1.1 Public Market (Business to Customer)

The developer/manufacturer wants to demonstrate to his customers that they can have confidence in the security provided by the smart card integrated circuit or another product he develops. For that reason he prefers to have an "official" certificate issued by an officially recognised Certification Body. The evaluation carried out by specialised laboratories (as third parties) gives evidence of the product's assurance. Assurance in turn gives the confidence needed by the customers and users.
Independent from the application contexts the products are used, evaluation schemas were set-up by governmental organisations and assurance requirements were defined. Such national evaluation schemas shall meet general market needs. Here criteria such as the "Information Technology Security Evaluation Criteria (ITSEC)" [4] and the "Common Criteria" ([1], [2], [3]) are used and international recognition agreements were signed. In practice recognition of certificates (evaluation results) is a problem especially if the Evaluation Facility is not known and the methods of the laboratory are not clearly documented in the Certification Report being published.
The evaluation shall give assurance (or trust) that the product meets its security target. The corresponding requirements checked during the evaluation are defined in a document called Security Target. The result of the evaluation is given in form of a verdict (pass or fail). If needed, directions for the developer or the user of the product are given. The criteria set out in the ITSEC or Common Criteria permit the developer to define the security functions without any restriction and to choose one out of seven evaluation levels representing increasing confidence in the ability of the product to meet its Security Target. The evaluation level determines the assurance on the verdict. In addition, a minimum strength of the security mechanisms shall be claimed.

There are clear advantages of having an evaluation schema set-up and maintained by governments. Perhaps the most important ones are:

- independence from influence of manufacturers and service providers,
- broad recognition of the evaluation results and the Certificates, and
- possibility to incorporate many institutions, organizations, and laboratories especially for definition and maintenance of the evaluation methodology.

Therefore, a high quality is expected. Again the quality of the assessment depends on the knowledge, expertise and market position of the Evaluation Facilities. The national bodies being responsible for the accreditation of laboratories and the Certification Bodies monitoring the evaluations have to have detailed know-how and enjoy a high reputation. Otherwise the schema will not be valuable but a burden for the laboratories (and the vendors) trying to maintain a high standard.

1.2 Private Market (Banking Applications)

Especially in banking applications controlled by specific providers of payment systems, card issuers or banking associations the approach is little different. For example, the use of a smart card in the German GeldKarte is permitted by Zentraler Kreditausschuß (ZKA, the common organisation of the German credit sector associations) only after successful evaluation of its hardware and software. This evaluation must be carried out by a laboratory (as trusted third party) being accredited by ZKA. The evaluation must show that the ZKA-Criteria ([9] or [10]) are fulfilled.

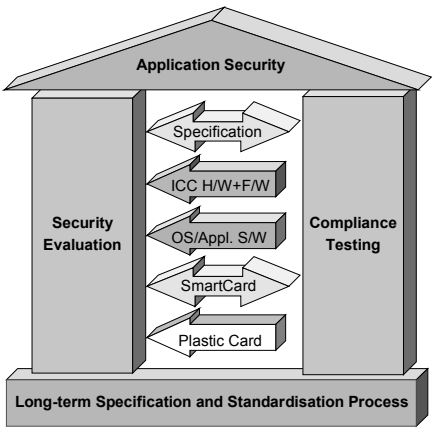

Fig. 2. Foundation of Application Security in Smart Card based Payment Systems

In the late 80's the banks in Germany began to develop their own Evaluation Schema. In the same time governmental organizations developed security evaluation criteria [8]. But the banks decided not to participate in the schema to have the freedom to control the evaluation process independently according to their specific needs.

In banking applications the products (for example smart cards) are designed to provide specific services. The cards are purchased by experts of the banks or their service companies. The specifications are not developed by the card manufacturer but by security experts commissioned by the banks, their associations or working groups. Note that there are approximately 40 million cards in Germany each equipped with the same functionality. So, there are a lot of differences compared to other public markets.

As shown in Figure 2 the security in banking applications like the electronic cash system which uses the eurocheque card or the GeldKarte in Germany is supported by a long-term specification process, security evaluations as well as compliance testing. The specification of the application must be subject to an evaluation. It is used again when performing the exhaustive compliance testing.
There are some advantages of having an independent evaluation schema. Perhaps the most important ones are:

- flexible definition of assessment rulers,
- possibility to require the analysis of specific attack scenarios and
- possibility to require improvements of the products.

The approval is given if all successful attacks considered are so expensive or difficult that the value of gathered information is less than the expenditure. There is some opportunity for interpretation which in turn introduces flexibility since the aspects (i) attacker's skill, (ii) attacker's knowledge, (iii) money and equipment, (iv) time, and (v) availability of samples (components) must somehow be combined to yield an overall verdict.

Since Zentraler Kreditausschuß (ZKA), as the Approval Authority for the banking applications just mentioned, is held responsible for the security it is in the position to demand the improvement of the system. For instance, some years ago a plan has been presented how to attack the Data Encryption Standard (DES) using hardware especially designed for that purpose [12]. Our company worked out all details and presented this information to ZKA [13]. As a result, the German banks decided to move to the Triple-DES. Note that there are more than 250,000 terminals and about 40 million smart cards in the field.

2 Observations

In the following examples are discussed showing difficulties in practical evaluations. First they help to understand the peculiarities of different evaluation schemas. Then the examples shall give indications on how to develop such schemas and the ability of the parties being involved to treat with them.

2.1 Case Study #1: Who Defines the Security Target?

The Security Target shall define the user's requirements because the product or system is checked against it. The evaluations results in turn (verdict, answer to the question defined in the Security Target) guides users whether or not to purchase and use the product. Therefore users (knowing the application context), developers (knowing the product) and third parties (being familiar with the assessment schemas and methodologies) should co-operate to define the Security Target.

The criteria set out in the ITSEC or Common Criteria permit the sponsor to define the security functions without any restriction and to choose the evaluation level. But often

the Security Target does not exactly meet the user's requirements. Then the evaluation will fail to give the evidence exactly needed by the user.
The user finds himself in a bad position: He has either to demonstrate to the developer that the Security Target does not meet exactly his needs or he has to live without having a certified product. Note that only a small percentage of the products have been evaluated.
The Common Criteria [1] allow to define such requirements for a set of products all intended to respond to the same security needs identified for similar environments. Such a set of requirements is called "Protection Profile". A Protection Profile holds for a group of products (implementations). Before using such a Protection Profile in an evaluation process, it must be evaluated and then filled with all the information identifying a special implementation (product).
As a consequence, a Protection Profile like [14] is a helpful tool for manufacturers (when developing their products etc.) and for users (to articulate their security needs). Evaluations based on Security Targets which in turn are based on the same Protection Profile are expected to yield comparable results.
Unfortunately, Protection Profiles are often written by the manufacturers and focus on specific aspects only [14]. It is therefore up to the users to clearly express interests. This can be done by writing Protection Profiles or by defining similar sets of requirements. An outstanding example for the second way are the regulations of the German "Digital Signature Act" [15] and its "Digital Signature Ordinance" [16]. Here standard assurance requirements are used [4]. But functional requirements are also defined for services and components to be provided for a public key infrastructure planned to partly replace the hand written signatures. Users must form consortiums to define functional requirements not only assurance requirements.

2.2 Case Study #2: Logical Versus Physical Security

Software often being evaluated provides security against hostile access on a well-defined interface or on an external channel. The software itself is not subject to an attack. Security function and threat agent are well separated. Software provides logical security. But in many cases one can not guarantee nor even assume that the attakker is not able to attack the security functions themselves. If the module is in a hostile environment and not protected by other means it can be subject to tampering or other types of influencing its behavior. Physical security is required.
In the nineteenth century Kerckhoff stated that secrecy must reside entirely on the key. So, it is assumed that an attacker may have complete knowledge of the cryptographic algorithm and its implementation. For smart cards this assumption does not hold. Especially for hardware the secrecy of design information is important. All details about the design and layout relieve attacks since modern equipment such as a focused ion beam (FIB) can be used to re-wire the chip. If design data are available prior reverse engineering is not required. But Kerckhoff's assumption does not always hold for software too. For instance the concrete software implementation of a cryptographic function may provide measures to avert the Differential Power Analysis (DPA, refer to chapter 2.6). Knowing the method of data coding or data processing the attack can be refined and perhaps successfully be carried out. So, smart card secu-

rity is also based upon concealing information. If the attacker is able to manipulate or intentionally disturb the card's operation, the construction of the countermeasures in hardware (and sometimes in software) must be kept secret. Then the attacker must carry out a costly reverse-engineering. But hiding information has a disadvantage. The more experts had analyzed a solution the higher is to estimate the assurance of its security. Restricting the availability of information reduces the number of experts having the possibility to approve, disapprove or improve the mechanisms considered.

It seems that the differences between hardware and software measures have not been thoroughly taken into account. The ITSEC and the Common Criteria and their evaluation guidelines do not consider the study time needed to prepare the attack. The ZKA approach distinguishes between first and following attacks.

2.3 Case Study #3: Information Versus Protection

In fact all security mechanisms realized in the hardware can be disabled or bypassed by direct manipulation based on previous reverse-engineering. The hardware developer may describe his measures built in to counter direct manipulation or reverse-engineering in the Security Target. This would inform the user about such important security measures but discloses the details to the public. If such kind of information is kept secret it is up to the Evaluation Facility to assess them. But for the same reason it is difficult to inform the user about the effectiveness of countermeasures against direct manipulation and reverse-engineering, for instance. But especially card issuers taking the risk for fraud in payment systems have legitimate interests to be informed about the existence and the effectiveness of the security measures built in.

All kind of information about such details implicitly may yield information about things not thoroughly being addressed and therefore give hints to possible attacks (possible weaknesses). And it will not only inform attackers but competitors. Of course, security shall not be founded on confidentiality of design details, but not publishing details often supports security. This can be understood by evaluators working for chip manufacturers when they read papers on chip card security being published by experts not having the same level of detailed information.

2.4 Case Study #4: Smart Cards as Composite Components

Smart card hardware and software is developed by different companies. These components are assembled in a particular way. Then security relevant data are injected into the card. The process is depicted in Figure 3.

Different entities are involved when producing a smart card: software development, chip and mask manufacturing, module manufacturing, card manufacturing (plastic #1 and #2), and personalization in two steps. Even for the smart card chip itself there are at least two companies providing components.

Hence for practical reasons and to prevent disclosure of details of the design to the other party, hardware and software are evaluated separately.[1] Apart from such confidentiality aspects each company must take the responsibility for his own developments or services.

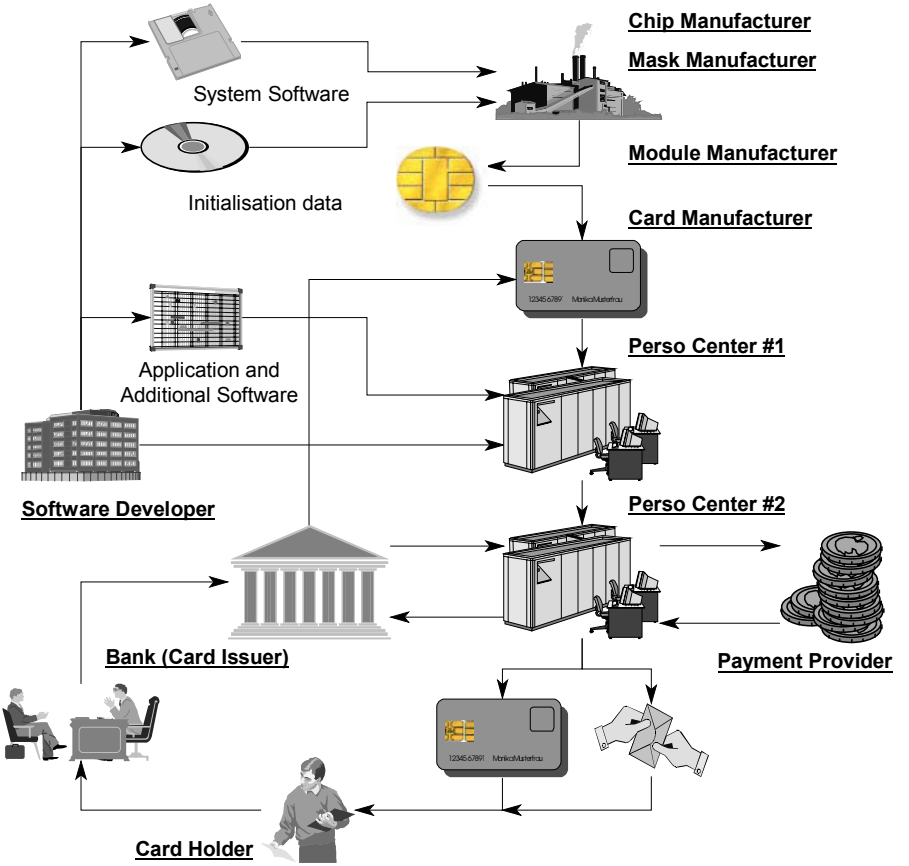

Fig. 3. Process of Producing Smart Cards

The security objectives for a smart card are twofold:

- Ensure "security" for the card when being in the field.
- Maintain "security" throughout the development and production process.

Although many specialists concentrate on the security in the field since the smart card is delivered into a hostile environment without any security regulations and may be subject to tampering, the security in the development, production and personalization process is also important. More precise, the organization of the personalization

[1] Of course, for instance for DPA both companies have to co-operate to provide samples for analysis soon.

process turns out to affect the security functionality to be provided by the smart card. So, the security objectives, a smart card component is being assessed against, depend very much on the application context which in this case includes the production and personalization process.
Therefore, one has to start with a concept of the smart card supply chain (refer to Figure 3). It addresses security and should be a subject to a separate assessment. From this a list of requirements for the individual components (and processes using other equipment) is gained including information about the number, extend, rigor, and depth of the evaluations to be carried out.
In principle, the evaluation criteria used in public markets (ITSEC and Common Criteria) cover all the aspects of a product's life-cycle (especially: development, system generation, delivery, configuration, and effectiveness in the field). Nevertheless, a central authority is required being responsible to ensure that all the measures fit together. Note that there are several components and processes. The published result of an ITSEC or Common Criteria evaluation (certification report) usually do not contain enough information to decide whether continual security is guaranteed.
In case of complicate composite products evaluation schemas like that of Zentraler Kreditausschuß (ZKA), as the Approval Authority for the banking applications mentioned above, are very efficient. Because of nowadays rapid changes in technology, evaluation overhead should be avoided. Simultaneously, requirements must be defined soon. ZKA is in the position both to approve components and processes (based on evaluation results provided by the laboratories) and to represent and improve the "user's requirements" since the payment systems are operated for the banking community the ZKA belongs to.

2.5 Case Study #5: Security Target Definition for Hardware

The hardware's countermeasures are often characteristics of the device which can not easily be described. Sometimes there are countermeasures not being designed as such. Nevertheless, the evaluator has to check whether the device has vulnerabilities. For example, an attacker may try to cause faulty operations of a smart card processors to compromise the modules security [17]. It is well-known that RSA-keys (Bellcore attack) or DES-keys (Differential Fault Analysis, DFA) can be read out if the attacker succeeds in causing specific faults by exposing the device to radiation or changing the environmental conditions in another way.
The principle of the Differential Fault Analysis (DFA) is shown in Figure 4 (last round of the DES, one S-Box i and the associated lines are considered). The attacker looks for the key component K(i) but he does not know C(i). Due to a single bit fault in R_{15} one has one or two values i with $I(i)' \neq I(i)$. (The faulty values are marked with a prime.) Comparing error-free and faulty values one has

$$S_i [I(i) \oplus K(i)] \oplus S_i [I(i)' \oplus K(i)] = O(i) \oplus O(i)' \quad \textbf{(1)}$$

The unknown constant value C(i) disappeared.
There are lots of environmental conditions which may cause erratic operation. For instance our laboratory read out keys by superimposing glitches on the power supply. Of course, there are rather simple measures to be implemented in the DES calculation

to prevent this kind of attack. But what are the issues to be checked by the evaluator when he considers the bare hardware only. According to the security evaluation criteria mentioned above (ITSEC and Common Criteria) the developer must explicitly define security functions or mechanisms designed to avert threats.

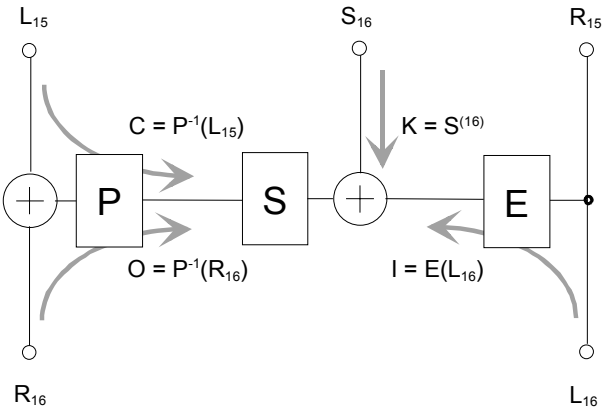

Fig. 4. Differential Fault Analysis: Last Round of the DES

Hardware and software may be evaluated separately to avoid disclosure of design details. In addition, each company must take the responsibility for his own developments. But it is often difficult to define a threat for the hardware since the cryptographic algorithm is realized in the software (outside the Target of Evaluation). General statements like robustness against failures are hard to check since there are many ways to affect the chip and the effect of a malfunction caused by an attacker is difficult to rate without knowing the application context.

But listing the security measures to be implemented in the hardware does not solely solve the problem. The smart card must withstand attacks. The analysis of such attack scenarios require to assess the suitability, binding and strength of a set of many security measures and characteristics of the hardware (layout for example). Even the latter are often not claimed to be a security measure. A smart card hardware offering many state of the art security measures may have fundamental vulnerabilities. If defining such detailed security requirements then the user (not the hardware developer) unexpectedly will design smart card security.[2]

User groups like banking consortiums shall carefully use lists of security measures. They are helpful as a first guidance. But its again the Evaluation Facility performing the detailed investigations. The evaluation schema operated for the users shall ensure that skill and knowledge of the laboratories is developed. Lists of attack scenarios and methods are mandatory.

In the case of logical security it can rather easily be decided whether a solution is "secure". For example the effort for an exhaustive key search can be calculated and

[2] In addition, if a solution has been disapproved the developer may lead this back to the requirements he is responsible for.

then the probability for a successful attack can turn out to be almost zero. The identification of individual mechanisms often supports this analysis. For hardware the rating is more difficult. The effort for a successful attack is often significantly smaller. When evaluating a complex of many mechanisms, characteristics and properties the verdict can be quite clear. But if one considers individual mechanisms they may be assessed to be rather weak. So, the result of the strength of functions/mechanisms analysis is to some extend pre-determined by the complexity of the functions or mechanisms identified on the Security Target level.

2.6 Case Study #6: Differential Power Analysis (DPA)

In July 1998 for instance, the new attack scenario Differential Power Analysis (DPA) [11] has been published.[3] Although it was known that an external observable like the power consumption or radiation contain information about the secrets being processed by a smart card or any other device, such kind of attacks have not thoroughly been considered before. In July 1998 our lab read out keys from smart cards using DPA the first time.
The principle (using the last round of the DES) is shown in Figure 5.

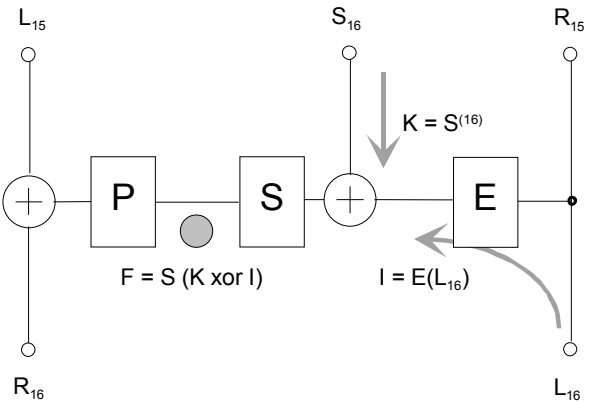

Fig. 5. Differential Power Analysis (DPA): Last Round of the DES

The pseudo code below shows the analysis process. The analysis is carried out for one or two bits (denoted by b) of each S-Box (denoted by i). K(i) and I(i) are the values of the lines associated with the input lines of that S-Box i.

```
locate "time interval" first
procedure start
choose: bit line b, key hypothesis K(i)
for (very much input values) do
   calculate F(b) = S { I(i) ⊕ K(i) }
   if F(b) = 0 then V(m) = -1 else V(m) = +1
   m = m+1
```

[3] The attack has been announced in spring 1998.

```
      measure power consumption over time S(m,t)
   for (each time in the interval) do
      calculate linear correlation
      between V(m) and S(m,t) giving CR(t)
      check if CR(t) shows significant "peaks"
   end start.
   for (each key hypothesis K(i) & other bit lines b) do
      execute "procedure start"
```

Results measured by our lab are shown in Figure 6. The upper curve is the co-variance CR(t), the power consumption over time S(m,t) is the curve below.

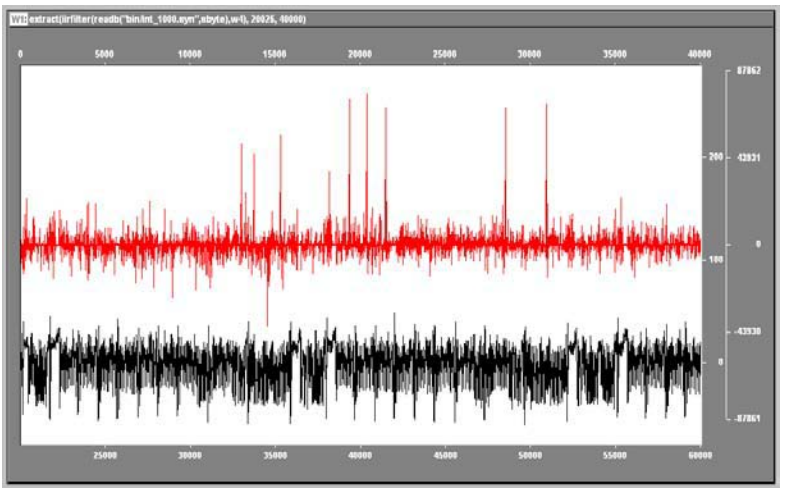

Fig. 6. Linear correlation (co-variance) CR(t) and power consumption over time S(m,t)

All manufacturers of smart card hardware and software were requested immediately by Zentraler Kreditausschuß (ZKA) to add countermeasures against the DPA (both in hardware and software) soon. ZKA has the knowledge and as the responsible organization is in the position to demand such improvement of the system. This again shows the immediate reaction of this evaluation schema.

2.7 Case Study #7: Hardware Versus Software

The design hierarchy of a smart card is shown in Figure 7. Things like the process technology are changed not very often but the customer's software may change rather rapidly. The higher the level the more effort and time must be invested to make a change. So, one likes to assess the hardware and the application software separately.

Modern smart card hardware is equipped with special security mechanisms like detectors etc. For the mechanisms to be effective often the software has to properly take advantage of them. The mechanisms must be enabled or initialized. Register bits

(flags) signaling a possible attack must be evaluated, interrupts must be served, and the software must properly respond to such events to bring the card into a secure state.

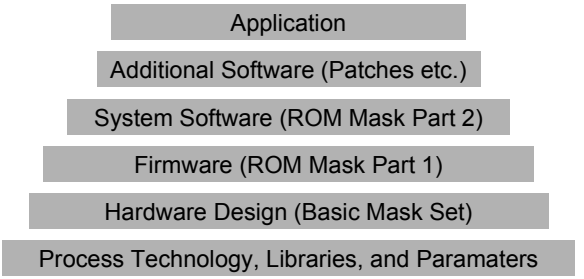

Fig. 7. Design Hierarchy of a Smart Card

In some cases, just the evaluator of the *hardware* formulated requirements how to use the hardware's security characteristics best possible. Restrictions and conditions were discovered during the evaluation of the hardware. Or security relevant software had to be changed since the characteristics of the memories (especially E^2PROM) showed that a vulnerability might have been introduced. The other way around it was just the evaluator of the *software* who discovered that special characteristics of the hardware are required to maintain security. But of course he could not check if the hardware fulfills these requirements.

Such findings must be listed as guidelines for the evaluator of the other component. Zentraler Kreditausschuß (ZKA), as the Approval Authority, ensured that these lists are checked before final approval.
These issues could not have been discovered by locking at either hardware or software only. In addition, it is often not feasible to have such information on a security target (or requirements) level. In many cases, the guidelines required (restrictions or conditions) were outcomes of an evaluation. Communication between different Evaluation Facilities (if needed), mediated by an experienced Approval Authority or Certification Body, is needed when assessing different components which together built a secure system.
It is hard to force a company to disclose the details needed to the another company. Usually, they will not co-operate in order to have one evaluation for a composite product. In addition, this information comes often from an assessment the developer normally did not performed by himself. The information being published in evaluation reports are not sufficient. If such information would have been included secret information is disclosed. Therefore, the evaluation schema must support the technical communication between the evaluators and force them to look beyond his target of evaluation.

2.8 Case Study #8: Whom Do You Trust?

Security evaluation criteria like the ITSEC and the Common Criteria are designed to assess technical measures. But obviously, security can not be guaranteed by technical

means alone. They have to be supplemented by organizational, personnel and other measures.

Obviously, if the developer and the Evaluation Facility collaborate backdoors and exploitable vulnerabilities may exist. More general it is up to the evaluators to guarantee assurance. Therefore, laboratories with long-term experience and good reputation shall be used.

Many organizations define requirements for the smart card design and production process. They consider traceability aspects as well as *technical* measures to proof the authenticity of the components being approved. Some of those concepts are reasonable but other ideas go too deep. It is always important to consider carefully before requiring technical measures. In smart card production processes this may introduce too much overhead and overtax the possibilities of the vendors. One should focus instead on the hostile actions for the card in the field and on the most risky actions like (i) the injection of keys and other critical data, (ii) the mechanisms needed to protect components not being ready to be issued and (iii) the security provided by the hardware and software of the card.

So, there is again a trade-off: Technical measures help to reduce the trust needed when services are delegated to other parties. But complex technical solutions can be too difficult and expensive to realize and may overtax the possibilities of the vendors.

3 Summary

Presently, too little attention is taken on the fact that the evaluation or approval process itself effects the evidence of the assurance and the value of evaluation verdicts. There are different evaluation schemas being designed for different purposes. After having introduced the elements of an evaluation schema trade-off situations typical of smart card evaluations were discussed. Studying these examples give value information to individuals, companies and organizations using evaluated products:

The criteria set out in the ITSEC or Common Criteria permit the sponsor to define the security functions. If the evaluations are directed by the developer, then the evaluation may often fail to give the evidence exactly needed by the user. Users must form consortiums to define functional requirements not only assurance requirements.

Software provides security against hostile accesses on a well-defined interface or external channel. Security function and threat agent are often well separated. Software provides logical security. Hardware modules with embedded software used in a hostile environment can be subject to tampering or other types of influencing its behavior. Physical security is required. Kerckhoff assumed that an attacker may have complete knowledge of the cryptographic algorithm and its implementation. For smart cards this assumption does not hold. Especially for hardware the secrecy of design information is important. But Kerckhoff's assumption does not always hold for software too. But restricting the availability of information reduces the number of experts having the possibility to approve, disapprove or improve the mechanisms considered.

Differences between hardware and software measures have not been thoroughly taken into account. The ITSEC and the Common Criteria and their evaluation guidelines do not consider the study time needed to prepare the attack. The ZKA approach distinguishes between first and following attacks.

In fact all security mechanisms realized in the hardware can be disabled or bypassed by direct manipulation based on previous reverse-engineering. According to the ITSEC and the Common Criteria the countermeasures must be described in the Security Target. But this would discloses the details to the public. But especially card issuers taking the risk for frauds in payment systems have legitimate interests to be informed about the existence and the effectiveness of the security measures built in.

Different entities are involved when producing a smart card. For practical reasons and to prevent disclosure of details of the design to the other party, hardware and software are evaluated separately. Apart from such confidentiality aspects each company must take the responsibility for his own developments or services. And the organization of the personalization process turns out to affect the security functionality to be provided by the smart card. In principle, the evaluation criteria used in public markets (ITSEC and Common Criteria) cover all the aspects of a product's life-cycle. But sometimes the security objectives for a smart card component depend very much on the production and personalization process. A central authority is required being responsible to ensure that all the measures fit together. The published result of an ITSEC or Common Criteria evaluation (certification report) usually do not contain enough information to decide whether continual security is guaranteed.

Unfortunately, it is often difficult to define a threat or security objective for the hardware as required by the criteria. General statements like robustness against failures are hard to check since there are many ways to affect the chip and the effect of a malfunction caused by an attacker is difficult to rate without knowing the application context. For software it can rather easily be decided whether a solution is "secure". For hardware the rating is more difficult. The effort for a successful attack is often significantly smaller. In addition, the result of the strength of functions/mechanisms analysis is to some extend pre-determined by the complexity of the functions or mechanisms identified on the Security Target level.

Zentraler Kreditausschuß (ZKA) has the knowledge and is in the position to demand improvement of the system. Examples given show the immediate reaction of this evaluation schema. A plan has been presented and worked out how to attack the Data Encryption Standard (DES). As a result, the German banks decided to move to the Triple-DES. Some years later all manufacturers of smart card hardware and software were requested by ZKA to add countermeasures against the DPA (both in hardware and software) soon. The criteria were extended.

In some cases, just the evaluator of the *hardware* formulated requirements how to use the hardware's security characteristics best possible. Restrictions and conditions were discovered during the evaluation of the hardware. The other way around it was just the evaluator of the *software* who discovered that special characteristics of the hard-

ware are required to maintain security. Such findings must be listed as guidelines for the evaluator of the other component. Therefore, the evaluation schema must support the technical communication between the evaluators and force them to look beyond his target of evaluation.

Obviously, security can not be guaranteed by technical means alone. They have to be supplemented by organizational, personnel and other measures. Many organizations focus on technical measures for the smart card design and production process. In fact, technical measures help to reduce the trust needed when services are delegated to other parties. But complex technical solutions can be too difficult and expensive to realize and may overtax the possibilities of the vendors.

References

1. Common Criteria for Information Technology Security Evaluation; Part 1: Introduction and General Model; Version 2.0, May 22nd, 1998
2. Common Criteria for Information Technology Security Evaluation; Part 2: Security Functional Requirements, Part 2: Annexes; Version 2.0, May 22nd, 1998
3. Common Criteria for Information Technology Security Evaluation; Part 3: Security Assurance Requirements; Version 2.0, May 22nd, 1998
4. Information Technology Security Evaluation Criteria (ITSEC); Provisional Harmonised Criteria, Version 1.2, June 1991
5. Information Technology Security Evaluation Manual (ITSEM); Provisional Harmonised Methodology, Version 1.0, September 1993
6. ITSEC Joint Interpretation Library (JIL), Information Technology Security Evaluation Criteria; Version 2.0, November 1998
7. Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD, December 1985 ("Orange Book")
8. German Information Technology Security Criteria (ITSK), "Green Book"
9. Criteria for the Security of electronic cash Systems, Zentraler Kreditausschuß (ZKA)
10. Criteria for the Security of Smart Card based Payment Systems, Zentraler Kreditausschuß (ZKA)
11. Paul Kocher, Joshua Jaffe, and Benjamin Jun: Introduction to Differential Power Analysis and Related Attacks, Cryptography Research, July 31st, 1998
12. M. Wiener: Efficient DES Key Search, Manuscript, Bell-Northern Research, Ottawa, 1993 August 20
13. debis IT Security Services: Brute-Force-Attack on the Data Encryption Standard (DES), March 1996
14. Protection Profile Smart Card Integrated Circuit, Version 2.0, Issue September 1998, Registered at the French Certification Body under the number PP/9806
15. Act on Digital Signature (Digital Signature Act - Signaturgesetz - SigG), in: Article 3 Federal Act Establishing the General Conditions for Information and Communication Services - Information and Communication Services Act - (Informations- und Kommunikationsdienste-Gesetz - IuKDG); Federal Ministry of Education, Science, Research and Technology, 22 July 1997
16. Digital Signature Ordinance (Signaturverordnung - SigV), On the basis of § 16 of the Digital Signature Act of 22 July 1997 (Federal Law Gazette I S. 1870, 1872)
17. Guidelines for Implementing and Using the NBS Data Encryption Standard; FIPS PUB 74-1; April 1st, 1981