Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards

Bruce Schneier
Counterpane Systems
schneier@counterpane.com

Adam Shostack
Netect, Inc.
adam@netect.com

October 19, 1999

Abstract

Smart card systems differ from conventional computer systems in that different aspects of the system are not under a single trust boundary. The processor, I/O, data, programs, and network may be controlled by different, and hostile, parties. We discuss the security ramifications of these "splits" in trust, showing that they are fundamental to a proper understanding of the security of systems that include smart cards.

1 Introduction

Smart cards, credit-card–sized devices with a single embedded chip—CPU and RAM/ROM—are viewed by some as "magic bullets" of computer security. They are being proposed (and used) for access control, electronic commerce, authentication, privacy protection, etc. Unfortunately, there is little analysis of the security risks particular to smart cards, and the unique threat environments that they face.

In this paper, we discuss the security model of a smart card system independently of its application. We look at the fundamental properties of a smart card—a CPU and memory device with no means of communicating with the outside world—and show how these properties make systems based on smart cards riskier than similar systems based on self-contained computers. A clear example is a person carrying a card whose computer is under someone else's control. This is an unusual situation for a typical computer, and a common one for a smart card. We show that for many applications, using a smart card securely means understanding it not as a "trusted" computation platform, but as a data storage device with limited computational abilities.

1.1 From Computers to Smart Cards

The best way to understand the threats facing a smart card is to start with the threats associated with a conventional desktop computer. We believe that the most important security aspect of smart cards, as participants in protocols, is the way in which they differ from other computational devices. By starting with a general purpose computer, and splitting apart its various functions into those that make up a smart card and its operating environment, we can examine each change and how it affects security. Each of these splits adds opportunities for attack. For example, consider a case where the owner of a card does not control the data stored on it. This leads to attacks by the person possessing the card against the data stored within it. This attack simply isn't possible if there is no such split.

Our model of a general purpose computer consists of a CPU, storage, input/output devices, and power supply. The CPU is the primary processor of the computer, responsible for carrying out computation. In a normal computer, it is tightly coupled to its storage, such as RAM, disk drives, or tape, as well as its generalized I/O devices, such as keyboards or mice for input, terminals or printers for output, and various digital communication ports, such as serial ports or Ethernet cards. In this configuration, the computer can be treated as a single unit for most threat models.

We begin by miniaturizing the computer, which adds nothing beyond a useful visualization tool. Consider a computer such as the REX personal organizer. This PC-CARD has a small screen, a PC-CARD interface to communicate with another computer, and a few buttons for input. We will now transform the REX, in stages, into a smart card, showing how each step of the transformation leads to new vulnerabilities.

Consider the I/O port, and replace it with a slow-speed serial port. The system that the card connects to has a limited ability to attack it, since the card is presumably going to be attached only to its owner's computer, or perhaps, for a few moments, to another to trade contact information. The card, throughout, has the ability to send and receive information through its screen and buttons. It would not be difficult to transform something like the REX into a secure electronic checkbook. (There are other engineering challenges, but it is substantially easier than building the same system with a smart card.)

Continue by decoupling the input mechanism, such that user input must go through a keyboard attached to the reader. It is obvious that the keyboard could record PIN and card information for use in a later attack. Lastly, remove the screen, such that the card has no way of communicating with its user except through a screen of indeterminate fealty.

The essential characteristic of a smart card is that its functionality is split in ways unusual for a computer. These splits mean that a smart card is "handicapped," by which we mean "unable to interact with the world without outside peripherals." This is the essential nature of smart cards: one that differentiates them from portable computers such as the Palm Pilot, and that defines the trust model in which they are forced to operate. Other splits may and do occur, but the fundamental one is that of being restricted in their I/O.

Smart card functionality is split in other ways. The cardholder might not have any control of the software running on the card. In the case of multifunction cards, the card issuer might not have any control either. The owner of the data inside the card might not be the cardholder, and the data owner might require that the cardholder not be able to modify, or even view, the data.

In the following sections, we examine the ramifications of the split described above, as well as others commonly found in smart card systems. Our models often include five or six parties. We examine in depth how the parties, when split, might attack each other. We also examine the motivations that cause attackers to engage in the variety of mischief that becomes possible when roles are split apart. And finally, we discuss different resistance models.

2 Model Trust Environment of a Smart Card

There are many parties potentially involved in any smart card–based system. Usually, there are at least five or six, including the cardholder, the terminal, the data owner, the card issuer, the card manufacturer, and the software manufacturer.

- The **cardholder** is the party who has day to day possession of the smart card. The card is in his wallet; he decides whether and when to use it. In the case of a smart card used as an electronic wallet, he is the person to whom the wallet was issued. He may control the data on the card, depending on the system, but it is highly unlikely that he had control of the protocols, software, or hardware choices made in the creation of the card system. Note that this is in contrast to many personal computer–based systems, where the owner and user usually has some say in the system he is using.

- The **data owner** is the party who has control of the data within the card. In cases such as using a card as a mechanism for carrying digital certificates, the card owner is also the data owner. However, if the card is an electronic-cash card, the issuer of the cash is the data owner, and this split opens the possibility of attack.

- The **terminal** is the device that offers the smart card its interactions with the world. The terminal controls all I/O to and from the smart card: the keyboard by which any data is entered into the smart card, and the screen by which any data from the smart card is displayed. If the card is used as a phone calling card, this is the pay phone owner. If the card is used as an ATM identification card, this is the ATM service provider. If the card is a pay-TV membership card, the terminal is the set-top box. [1]

[1] The previous two examples—ATM identification card and pay-TV membership card—illustrate times when the terminal, as well as the smart card, may be broken into several parties. In the case of an ATM, the use of another bank's ATM network and terminal is common, which means that the bank cannot rely on the terminal to be friendly. In the case of the pay-TV system, the terminal is in the long-term possession of the user, and can be attacked in the safety and comfort of the user's home. In cases where the terminal ownership, programming, possession, or other functions are split, a full analysis needs to be performed to ensure that the security impacts of the splits are understood.

- The **card issuer** is the party who issued the smart card. This party controls the operating system running on the smart card, and any data that is initially stored on the smart card. If the card is a telephone payment card, the issuer is the phone company. If the card is an employee ID card, the issuer is the employer. Sometimes the issuer just issues the card and then disappears from the system; other times he is involved with the system throughout. In some multi-function cards, the card issuer may have nothing to do with the applications running on the card, and may only control the operating system. In other multi-function cards, the same issuer may control all the applications running on the card.

From the security analysis point of view, it is often simplest to view the card issuer, the manufacturer, and the software engineers as the same party; however, they rarely actually are. Hence:

- The **card manufacturer** is the party who produces the smart card. Note that this is a simplification; the manufacturer may or may not own the fabrication facility in which the chips are actually made; they may have subcontracted design functions, and they may be using third-party tools in their work, such as VHDL compilers. However, we model all of these as the card manufacturer. Opportunities to subvert the manufacture of the card come in many places, to a wide variety of individuals.

- The **software manufacturer** is the party who produces the software that resides on the smart card. This is again a simplification of a probably complex array of makers of compilers, utilities, etc. Issues of trusting trust [Tho84] arise here in the same ways they do with the card manufacturer.

3 Examples of Trust Splits in Smart Card Systems

Following are representative smart card–based systems, described in terms of what parties control different aspects of the system. This list is not meant to be exhaustive, and there are both other examples of splits described here and other splits not described here.

- **Digital Stored Value Card.** These are payment cards intended to be substitutes for cash. Both Mondex and VisaCash are examples of this type of system. The card owner is the customer. The terminal owner is the merchant. The data owner and the card issuer are both the financial institution that supports the system.

- **Digital Check Card** This is similar to the card above, except that the card owner is the data owner.

- **Prepaid Phone Card.** These are simply a special-use stored value card. The card owner is the customer. The terminal owner, data owner, and card issuer are all the phone company.

- **Account-based Phone Card.** In this system, the smart card does not store an account balance, but simply an account number which is a pointer into a back-end database. The card owner and data owner is the customer, while the terminal owner and card issuer is the phone company.

- **Access Token.** In this application, the smart card stores a key which is used in a login or authentication protocol. In the corporate case, the cardholder is the employee, and the data owner, terminal owner, and issuer are likely the company. In the case of a multi-use access token, the cardholder and data owner might be the same person, while the terminal owner may be a merchant and the data owner a financial institution.

- **Web Browsing Card.** In this application, a customer can use his card in his own PC to buy things on the WWW. This is another example of a cash card. The difference is that the cardholder and terminal owner are both the customer (i.e., the owner of the PC). The data owner and card issuer are both the financial institution.

- **Digital Credential Device.** In this application, the smart card stores digital certificates or other credentials for presentation to another party. Here, the cardholder and the data owner are both the same. The terminal owner is either the other party (in an in-store application, for example) or the cardholder (browsing on the WWW). The card issuer is the CA that issued the credentials, or some other party that collects the credentials.

- **Key Storage Card.** In this application, the user stores various (possibly verified) public keys in a smart card to protect them having to be stored on his less secure PC. Here, the cardholder, the data owner, and the terminal owner are the same.

- **Multi-Function Card.** This card is the most complicated. The card manufacturer and card issuer are separate, as are the software manufacturers. The data owner may be the cardholder for some applications, and a separate entity for others. There are multiple terminal owners, depending on which applications are on the card.

4 Smart Card Threat Models

An attack is simply defined as an attempt by one or more parties involved in a smart card transaction to cheat. We consider two classes of attackers, those who are parties to the system, and those who are interlopers. Attacks by participants could be a cardholder trying to cheat a terminal owner, a card issuer trying to cheat a cardholder, etc. Outsider attacks could be mounted by someone who steals a card: a temporary cardholder who steals a card from a legitimate cardholder, or replaces terminal software or hardware. Attacks by outsiders are often similar to attacks on protocols involving general purpose computers; however, they may take advantage of various properties of the system created by the separation of roles.

Motives for attack fall into a few broad categories [Sch97]. First and most obvious are financial thefts, including theft of money or credit, or theft of services sold to the general public, such as telephone cards. There are also impersonation attacks, where the card system is an intermediate target, with the system being attacked to gain access to some computer system, or other access control device. These differ from theft of service in that the user could not purchase the service legitimately. For example, the use of an access card to get into a computer system; computer access is generally available, but access to the particular system is the goal of the attacker. There are attacks on privacy, where one party wants more information than is given by the protocol. Lastly, there are publicity attacks, where the attacker is motivated not by any direct financial gain through attacking the system, but a desire for notoriety.

5 Classes of Attack

Due to the large number of parties involved in any smart card–based system, there are many classes of attacks to consider. Our goal here is to categorize them by function split. That is, we will look at attacks by system participants against one another. Most of these attacks are not possible in conventional computer systems, since they would take place within a traditional computer's security boundary. However, they are possible in the smart card world.

5.1 Attacks by the Terminal Against the Cardholder or Data Owner

These are the easiest attacks to understand. When a cardholder puts his card into a terminal, he is trusting the terminal to relay any input and output from the card accurately. For example, if a user puts a stored value card into a vending machine and makes a $1 purchase, he is relying on the terminal to send a "deduct $1" message to the card, and not a "deduct $10." Similarly, when the card sends a message to the cardholder that says "balance = $1," the cardholder is relying on the terminal's screen to relay that message accurately. The ability for a rogue terminal to do damage in this environment is significant, and it is impossible for the cardholder to detect this kind of fraud in the context of a single terminal. This kind of fraud has been attempted using fake ATM machines [**?**].

Prevention mechanisms in most smart card systems center around the fact that the terminal only has access to a card for a short period of time. Software on the card could limit the amount of damage a rogue terminal could do. A stored-value card could, for example, only allow the terminal to deduct $1 maximum per transaction, and to perform no more than one transaction every minute [KS99]. However, there are prevention mechanisms that involve having the user own the smart card terminal, such as one attached to a personal computer. The real prevention mechanisms, though, have nothing to do with the smart card/terminal exchange; they are the back-end processing systems that monitor the cards and terminals, and flag suspicious behavior.

5.2 Attacks by the Cardholder Against the Terminal

More subtle are attacks by the cardholder against the terminal. These involve fake or modified cards running rogue software, with the intent of subverting the protocol between the card and the terminal. For some examples, see [McC96].

Good protocol design mitigates the risk of these kinds of attacks, which can be made more difficult by hard-to-forge physical aspects of the card (e.g., the hologram on Visa and MasterCard cards), which can be checked by the terminal owner manually. Note that digital signatures on the software are not effective here since a rogue card can always lie about its signature, and there is no way for the terminal to peer inside the card. Defending against this kind of attack requires another function split: the cardholder must not be able to manipulate the data inside the card.

5.3 Attacks by the Cardholder Against the Data Owner

In many smart card–based commerce systems, data stored on that card must be protected from the cardholder. In some cases, the cardholder is not allowed to know that data. A building access card, for example, could have a secret value inside the card; knowledge of this value could allow the cardholder to make additional access cards. Or knowledge of a secret key in an electronic commerce card could allow the cardholder to make fraudulent transactions. In other cases, the cardholder is allowed to know the value, but not allowed to change it. If the card is a stored-value card, and the user can change the value, he can effectively mint money.

There are two essential characteristics of these attacks. One, the card must act as a secure perimeter, preventing the cardholder from accessing the data inside the card. In this context, the card may need to be fairly confident that it will detect and respond to attacks with a minimum of control over its environment. And two, the attacker has access to the card on his own terms. He is allowed to take the card into his laboratory and perform whatever experiments he wants to. He is allowed to take cards and destroy them in order to learn how they work.

There have been many successful attacks against the data inside a card. These attacks include reverse-engineering and defeating tamper-resistance [AK96], fault analysis [BS97, BDL97], and side-channel attacks such as power and timing analysis [Koc96, Koc98b, KSWH98b, DLK+99].

These attacks have been particularly effective against pay-TV access cards [McC96, Row97], and have been used against digital cellular telephone access cards [BGW98]. They are starting to be used against stored-value cards for electronic commerce [Row97].

5.4 Attacks by the Cardholder Against the Issuer

There are many financial attacks that appear to be targeting the issuer, but this may be illusory. In fact, the attacks are targeting the integrity and authenticity of data or programs stored on the card. These attacks are made possible by the issuer's decision to use a smart card system where the cardholder holds data for the issuer or other party. Using the pay telephone application as an example, if the phone were to use an account-based system, where a simple card holds a very long account number that is used by the phone company to dereference an account stored on a back-end system, then there are account guessing and theft attacks based on the numbers. This sort of system can be enhanced by adding a challenge/response or inverted hash chain mechanism for sending replay resistant passwords. This makes strong use of a simple smart card in conjunction with a back office–managed authorization scheme to resist fraud. If the card issuer chooses to put bits that authorize use of the system in the card, they should not be surprised when those bits are attacked. These bits could be "authenticated" account numbers, or it could be a system with a key buried within the card, on the assumption that this key cannot be extracted, and proper completion of the protocol indicates that the card has not been tampered with. These systems all rest on the questionable assumption that the security perimeter of a smart card is sufficient for their purposes.

5.5 Attacks by the Cardholder Against the Software Manufacturer

Generally, in systems where the card is issued to an assumed hostile user, the assumption exists that the card will not have new software loaded onto it. This is enforced by the use of pre-issuance stages with various one-way transformations being employed by the card manufacturer to ensure that the software is not tampered with. The underlying assumption may be that the split between card owner and software owner is unassailable, and relies on the separation being strong. However, attackers have shown a remarkable ability to get the appropriate hardware sent to them, often gratis, to aid in launching an attack.

5.6 Attacks by the Terminal Owner Against the Issuer

In a system closed to outsiders, such as some prepaid telephone cards, the terminal owner is also the card issuer (the phone company has both roles). In some more open systems, like Mondex, the terminal owner is the merchant and the card issuer is Mondex. The latter split introduces several new attacks.

The terminal controls all communication between the card and the card issuer (generally the back-end of the system). In this system, the terminal can always falsify records that have nothing to do with the smart card, refuse to record transactions, etc. The terminal can also fail to complete one or more steps of a transaction to facilitate fraud or create customer service difficulties for the issuer. By failing to complete the action of debiting a card, a terminal can cheat the issuer, or by completing a transaction and not offering service (i.e., a pay phone) can create a service nightmare.

These attacks are not related to the smart card nature of the system, and are simply attacks against the relationship between the terminal owner and the card issuer. Some systems try to mitigate this threat by having the card and back-end computer make a secure connection through the terminal. Many systems use monitoring on the back end to reduce the effectiveness of these attacks.

5.7 Attacks by the Issuer Against the Cardholder

In general, most systems presuppose that the card issuer holds the best interests of the cardholder at heart. This is not necessarily the case, and a malicious issuer can launch several attacks against cardholders.

These attacks are typically privacy invasions of one kind or another. Smart card systems that serve as a substitute for cash must be designed very carefully to maintain the anonymity and unlinkability that are a property of cash money. Attacks or design failures can substantially reduce the privacy of the system. Alternately, a system may be sold as having more privacy than it in fact offers, allowing the issuer to gather data surreptitiously about the cardholders.

Features introduced into the card as the system matures may alter initial characteristics of the system with substantial impact on the privacy of the system. This can count as an attack by the issuer because the cardholder is rarely asked or able to discern the security impact of a change to the system made by the issuer. These changes are often not optional from the customer's viewpoint; the only choices are to accept the upgrade or leave the system. Lastly, this type of attack may be carried out by the issuer, or by the hardware or software designer, in collaboration with terminals, without the knowledge or consent of the issuer.

5.8 Attacks by the Manufacturer Against the Data Owner

Certain designs by manufacturers may have substantial and detrimental effects on the data owners in a system. The design of secure multi-user computers is a challenging one, and the security model to use to establish a secure kernel that offers processes protection from each other is not a solved problem. By providing an operating system that allows or even encourages multiple users to run programs on the same card, a number of new security issues are opened up.

The first, and most obvious, is subversion of the operating system and subsequently other programs. This is an area where mainstream operating system manufacturers have failed to provide adequate protection for the last thirty years. The vendors who have announced smart card operating systems recently do not have enviable records. However, even if the smart card operating system can be made secure, issues of user interface security remain and are exacerbated by the smart card's handicaps. How is the user (or the designer) to know what program is running when the card is inserted into a terminal? How to ensure that your program is talking to the terminal, and not through another program? How can a program that believes itself compromised terminate safely, and signal outward the cause for its demise? Or should it even try; what interesting attacks might become possible if a card announces its own imminent suicide? Can the card ensure that once such a message is sent the action of destroying its memory is completed, in the presence of a possibly hostile power supply?

Less obvious would be intentionally poor random number generators [KSWH98a], or other aspects of cryptographic implementation that are difficult and arcane areas to test [Sch97, Sch98a, Koc98a, Sch98b]. The manufacturer is in an admirable position to engage in kleptographic attacks [YY96, YY97a, YY97b]. Of the major smart card vendors, none has an admirable record of creating operating systems that were free of exploitable vulnerabilities. In addition, by providing implementations of various supporting protocols, the vendor may be in a position to leak an application's keys using any of several subliminal channels [Sim84, Sim85, Sim86, Sim94].

And finally, it is possible for one application on a smart card to subvert another application running on the same smart card. It has been shown how to take a secure protocol and to create another protocol, also secure, such that the second protocol breaks the first protocol if both are running on the same device using the same keys [KSW96].

6 Transformative, or Impersonation, Attacks

There is a class of attacks based on separating or changing the roles played by various parties; for example, changing the cardholder by stealing the card may allow access to data that the cardholder has stored, or using ActiveX controls that allow an attacker to become (in essence) the terminal owner, engaging in the set of attacks available to terminal owners.

The essential character of a transformative attack is that a party is transformed, leading to an unexpected set of motivations for that party. When a card is stolen, the new cardholder (i.e., the thief) has lost all interest in maintaining the security of the account, and possibly in the physical integrity of the card. When a terminal is subverted, its desire to participate in a fair manner is replaced by a desire to subvert the protocol (why else subvert the terminal?). Thus, when a system assumes that the data stored on a card is secure because the interests of the cardholder and issuer are aligned, a vulnerability is opened by the theft of the card.

Alternately, we examine a system with a smart card reader attached to a PC, where that PC is acting as part of the terminal. The terminal is presumed to be friendly to its owner; perhaps it is being used to carry Web certificates from home to work. Unfortunately, the terminal can be transformed by the introduction of an ActiveX control that changes the reader software. This attack, by changing the expected behavior of a component, can recast the security of the protocol. The behavioral change here can be active, in the case of changing a request and its associated display, or passive, in the case of monitoring attacks. Monitoring attacks can attack the privacy of the transactions made by the card or the secrecy of PIN or other data. The latter is probably a precursor to an active attack, not necessarily in the domain of the smart card protocol. That is, recall that PINs are often used in more than one system, and that the active attack does not need to attack the smart card system.

6.1 Attacks by Third Parties Using Stolen Cards

There are two differences between this attack and an attack by the cardholder. One, the thief does not have access to any secret information required to activate the card. And two, the thief has only a limited amount of time to carry out his attack before the cardholder will notice that his card has been stolen.

Hence, all the attacks by the cardholder are possible with the following addition: the thief is not concerned with any long-term repercussions against the legitimate cardholder. For example, a low-value stored-value card might deal with the potential of cardholder fraud by simply keeping records of cardholder transactions, and billing (or prosecuting) any discrepancies. A thief who steals a card would not be deterred by this defensive measure.

It is possible to build defenses into the system either at the card's or at the issuer's level. At the card level, there are perimeter and anomaly defenses available. The perimeter defense is that the card can consider several bad PIN attempts to be indicative of attack. (Note that this opens the card to a denial of service driven by a malicious terminal.) The anomaly detection defense would be for the card to store history information and detect a pattern change in its use. This is an aggressive requirement, but in those cases where a card can be used offline, it may make sense to raise a flag of some type, possibly requiring contact with its issuer before additional use to allow the back end system a chance to make a more elaborate or sophisticated decision, or perhaps simply to defend the system against card duplication.

6.2 Eve and Mallet

If we assume that the use of a smart card is to allow protocol interactions between mutually distrusting parties, or at least parties whose interests diverge, then the protocols must resist the same set of attacks that they would if the systems were implemented with general purpose computers. Thus, most attacks based on eavesdropping or malicious protocol manipulation may be modeled as the case of one party attacking another. Assuming that the protocol is well designed, it will resist these attacks equally well if the attacker is internal or external.

6.3 Collaborative Attacks

Systems that rely on the split between various components being maintained as a hostile boundary without cooperation may find themselves surprised when roles they had thought split are brought together. The smart card and set top box, supposedly representing different interests, may collaborate in obtaining unauthorized service for the owner of the television. Similarly, the terminal's owner may be surprised to discover that both the card and the terminal, made and programmed by the same shop, have certain undocumented features. The number of possible collaborations and interesting models for attack grows with the number of parties to the system. Those who forget that most attacks are perpetrated by insiders will likely be reminded (assuming their fraud detection models are good enough.)

7 Resistance Models

There are, broadly, two ways to resist attacks against smart card systems. The first is to make specific attacks harder: use strong cryptographic protocols, increase tamper-resistance, etc. We don't discuss these methods in detail; we believe they are less effective and more prone to implementation and design failure than the second, which is to make entire classes of attack ineffective. This can be done most effectively by reducing the number of parties, or increasing the transparency of a party's role to the point where carrying out an attack is difficult. The easiest way to reduce the number of parties is to combine roles so that there are fewer hats to wear. If, for example, the cardholder is also the data owner, all attacks by by the cardholder against the data owner are simply irrelevant. Or, if the terminal owner is also the issuer, then attacks by the terminal owner on the issuer are only possible in the transformative case, where an attacker takes control of the terminal.

7.1 Fewer Splits

Each time a system has the design role of two or more parties merged into one, the avenues of attack that are available to one of those parties against the other disappears. For example, if the cardholder and trusted terminal are merged by adding screen and data entry to the card, then the keysniffing and untrusted display problems simply disappear.

Contrariwise, adding parties to the system opens new venues of attack which need to be considered. The separation of the terminal and card from each other creates a venue which could scarcely have been designed better to enable man-in-the-middle attacks. The combination of physical encasement of the card, and terminal's control of the user interface and network allow most any such attack documented to be carried out if the protocol is not designed to handle it. Experience has shown that even many security products are released without consideration given to MITM, replay, and reflection style attacks. [Sho96, Sho97] Even if these attacks are considered, the addition of parties to a transaction makes managing keys, nonces, sequence numbers, and other defenses substantially more difficult.

Considering the smart card's inability to communicate with the outside world, the simplest (and also usually the least expensive) split reduction is to ensure that the cardholder and data owner are one. This will eliminate attacks by the cardholder on the data that currently plague most existing systems. The other extremely effective change to be made, adding screen and input devices to the card, involves a substantial increase in the cost of the card.

7.2 More Transparency

It is widely understood by the security community that the best way to ensure the security of a system is to allow widespread public examination of it. It has been shown repeatedly that interested attackers will obtain specifications or attack the system without them [Sho96, Bla94], and that open publication leads to review and analysis. (Examples are IPSec, PGP, and S/MIME.) Combining the mechanisms of simplicity and openness greatly simplifies the task of reviewers who choose to examine a system. Thus, reducing the number of parties not only eliminates entire classes of attacks as shown above, but it also makes the task of analyzing the system simpler. The simplicity of the security analysis will likely cause the analysis to occur sooner, as well as giving it a higher likelihood of success.

The transparency defense involves cleanly separating roles so that attacks are more difficult to execute. For example, the Mondex system includes a variety of terminal types (some portable) that allow a user to check certain parameters independent of a merchant terminal. This allows a class of attacks on the cardholder or be discovered much more quickly. Access to the full set of Mondex stored parameters (i.e., the data owner's data) would presumably make the system that much more secure by increasing the audit-ability of the system. Similarly, an attack by the software manufacturer is made more difficult by the presence of strong and clear specifications, and/or open source implementations.

7.3 Design for Security

This defensive model of design is focused on designing systems to be secure from the architecture down [SSS+98]. Adding security to a system after the design phase has been shown to be difficult, expensive, and failure prone. Therefore, we offer a model where careful design from the start eliminates the need for many costly and complex attempts to bolt security on at a later phase. The reductionist model not only simplifies the process of design and implementation, but is fairly difficult to implement incorrectly. We have seen that implementation failures are a primary cause of cryptosystem failure in the field [And94, Sch97, Sch98a, Koc98a, Sch98b].

Another facet to the transparency defense is to avoid the complexities and risk of multi-application smart cards. Not using a multi-application smart cards both reduces the number of parties involved and creates a simpler operating environment with less complexity and potential for bugs. The reduction in the number of parties using the card (from N to 2) means that the issues of OS subversion and cross application attacks are practically eliminated.

8 Conclusions

We have shown that the splitting of the security perimeter is a difficult task. In particular, having a user carry a computer on behalf of a data owner he may wish to attack is a very risky situation for the data owner. We have also shown that the card's handicap of being unable to communicate makes it highly vulnerable to attacks by the terminal. These vulnerabilities are part of smart card systems by design, and require substantial effort to combat.

We have outlined a pair of fundamental defenses for cards, that operate at the system design level, offering system designers a new model in which to evaluate their systems. This model encourages pushing security into the earliest phases of system design. We offer as a prime candidate for improvement placing the user interface under the control of the user. System designs that re-combine the roles into more capable systems will likely find their investment results in fewer points of weakness.

References

[And94] R. Anderson, "Why Cryptosystems Fail," *Communications of the ACM*, v. 37, n. 11, Nov 1994, pp. 32–40.

[AK96] R. Anderson and M. Kuhn, "Tamper Resistance – A Cautionary Note," *Second USENIX Workshop on Electronic Commerce Proceedings*, USENIX Press, 1996, pp. 1–11.

[BDL97] D. Boneh, R.A. Demillo, R.J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," *Advances in Cryptology—EUROCRYPT '97 Proceedings*, Springer-Verlag, 1997, pp. 37–51.

[BGW98] M. Briceno, I. Goldberg, D. Wagner, "Attacks on GSM Security," work in progress.

[BS97] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," *Advances in Cryptology—CRYPTO '97 Proceedings*, Springer-Verlag, 1997, pp. 513–525.

[Bla94] M. Blaze, "Protocol Failure in the Escrowed Encryption Standard," *Pro-*

ceedings of Second ACM Conference on Computer and Communications Security, ACM Press, 1994.

[DLK+99] J.-F. Dhem, F. Koeune, P.-A. Leroux, P. Mestre, J.-J. Quisquater, and J.-L. Willerns, "A Practical Implementation of the Timing Attack," *CARDIS '98 Proceedings*, Springer-Verlag, 1999, to appear.

[Jon93] K. Johnson, "One Less Thing to Believe in: High-Tech Fraud at an ATM," *The New York Times*, 13 May 93, pp. 1,B9.

[Koc96] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *Advances in Cryptology—CRYPTO '96 Proceedings*, Springer-Verlag, 1996, pp. 104–113.

[Koc98a] P. Kocher, "Hidden Flaws: Avoiding Unexpected Weaknesses," *The 1998 RSA Data Security Conference Proceedings*, RSA Data Security, Inc., 1998.

[Koc98b] P. Kocher, "Differential Power Analysis," available online from `http://www.cryptography.com/dpa/`.

[KS99] J. Kelsey and B. Schneier, "Authenticating Secure Tokens Using Slow Memory Access," in preparation.

[KSW96] J. Kelsey, B. Schneier, and D. Wagner, "Protocol Interactions and the Chosen Protocol Attack," *Security Protocols, International Workshop April 1997 Proceedings*, Springer-Verlag, 1998, pp. 91-104.

[KSWH98a] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Cryptanalytic Attacks on Pseudorandom Number Generators," *Fast Software Encryption, 5th International Workshop Proceedings*, Springer-Verlag, 1998, pp. 168–188.

[KSWH98b] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Side Channel Cryptanalysis of Product Ciphers," *ESORICS '98 Proceedings*, Springer-Verlag, 1998, pp. pp 97–110.

[McC96] J. McCormac, *European Scrambling Systems*, Waterford University Press, 1996.

[Row97] T. Rowley, "How to Break a Smart Card," *The 1997 RSA Data Security Conference Proceedings*, RSA Data Security, Inc., 1997.

[Sch97] B. Schneier, "Why Cryptography is Harder than it Looks," *Information Security Bulletin*, v. 2, n. 2, March 1997, pp. 31–36.

[Sch98a] B. Schneier, "Security Pitfalls in Cryptography," *CardTech/SecureTech Conference Proceedings, Volume 1: Technology*, CardTech/SecureTech, Inc., 1998, pp. 621–626.

[Sch98b] B. Schneier, "Cryptographic Design Vulnerabilities," *IEEE Computer*, v. 31, n. 9, September 1998, pp. 29–33.

[Sho96] A. Shostack, "Observed Weaknesses in the Security Dynamics Client/Server Protocol", *Network Threats Workshop, Dec 2-4 1996*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 38, R.N. Wright and P.G. Neumann, eds., American Mathematical Society, 1996.

[Sho97] A. Shostack, "Low Hanging Fruit: A Replay Attack on the TIS FWTK," presentation at the CRYPTO '97 rump session.

[SSS+98] C. Salter, O. Saydjari, B. Schneier, and J. Wallner, "Toward a Secure System Engineering Methodology," *New Security Paradigms Workshop 1998 Proceedings*, IEEE Computer Society Press, to appear.

[Sim84] G.J. Simmons, "The Prisoner's Problem and the Sublimianal Channel, *Advances in Cryptology: Proceedings of CRYPTO '83*, Plenum Press, 1984, pp. 364–378.

[Sim85] G.J. Simmons, "The Subliminal Channel and Digital Signatures," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 364–378.

[Sim86] G.J. Simmons, "A Secure Sublinimal Channel (?)" *Advances in Cryptology: Proceedings of CRYPTO 85*, Springer-Verlag, 1986, pp. 33–41.

[Sim94] G.J. Simmons, "Subliminal Channels: Past and Present," *European Transactions on Telecommunications*, v. 4, n. 4, 1994, pp. 459–473.

[Tho84] Ken Thompson, "Reflections on Trusting Trust," *Communications of the ACM* Vol. 27, No 8, August 1984, pp. 761-763.

[YY96] A. Young and M. Yung, "The Dark Side of Black Box Cryptography," *Advances in Cryptology — CRYPTO '96 Proceedings*, Springer-Verlag, 1996, pp. 89–103.

[YY97a] A. Young and M. Yung, "Kleptography: Using Cryptography against Cryptography," *Advances in Cryptology — EUROCRYPT '97 Proceedings*, Springer-Verlag, 1997, pp. 62–74.

[YY97b] A. Young and M. Yung, "The Prevalence of Kleptographic Attacks on Discrete-Log Based Cryptosystems," *Advances in Cryptology — CRYPTO '97 Proceedings*, Springer-Verlag, 1997, pp. 264–276.