1 Trustworthy User Devices[1]

Andreas Pfitzmann[1], Birgit Pfitzmann[2], Matthias Schunter[2], Michael Waidner[3]

1 Technische Universität Dresden, Fakultät Informatik, D-01062 Dresden, Germany, email: pfitza@inf.tu-dresden.de

2 Universität des Saarlandes, Fachbereich Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany, email: pfitzmann@cs.uni-sb.de, schunter@acm.org

3 IBM Research Division, Zürich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland, email: wmi@zurich.ibm.com

1.1 Abstract

There are numerous plans to supply users with personal devices to improve security in areas such as electronic commerce, payment systems, and digital signatures. Most of these plans take only smartcards into consideration. However, there is a rapidly growing market for far more powerful mobile user devices, for example, mobile phones, pagers, gameboys, multifunctional watches, personal communicators, and personal digital assistants (PDAs). Most mobile user devices inherently require security functions for some of their prospective applications.

To a great extent these two separate developments can be expected to merge over the next few years. This opens up new architectural options for security, but also poses new threats. This article surveys the resulting design issues. Although no device can be completely trustworthy, the authors believe that combining security functions and powerful mobile user devices can produce mutual benefits if appropriate measures are taken.

1.2 Introduction

One of the biggest challenges for security in communications and electronic commerce is the security of the devices used. For example, the best digital signature algorithm is of no value if the corresponding signature key cannot be stored and used securely. Even if a PC or workstation were accessible to only one person, and this person were careful not to load any programs from dubious sources onto this computer (both of these requirements are quite restrictive in themselves, considering what normal individuals and small businesses use PCs for), the risk to confidential data stored on the computer is great. Bugs are regularly discovered in standard programs that one cannot live without, like the operating system itself, browsers, emailers, and text editors. All of these programs allow malicious outsiders to gain at least partial control over the user's computer.

To a certain extent, one can manage the risk of insecure devices by non-technical means. For example, simple credit-card payments over the Internet are, actually, not a problem for the user because the merchants carry the risk. Even for applications where simple repudiation of an action is unacceptable to the recipients (e.g., repudiation of the order of a special-purpose merchandise), one can still manage risks and liabilities to quite a large extent; see the SEMPER Electronic Commerce Agreement for detailed proposals [BaZi_99]. Nevertheless, technical security is needed to keep the risk below certain levels. Moreover, such mechanisms only work for financial risks, not where loss of privacy or other direct damage is concerned.

A short-term solution is to introduce smartcards in an attempt to store keys, i.e., the main confidential data of a user, securely. However, even if the smartcard, as such, were perfectly secure, the overall solution is not. As shown in Figure 1.1, all of the control over the smartcard resides in the PC, which is not necessarily trustworthy. If malicious software has gained access to that computer, it can capture the user's PIN and allow the smartcard to perform entirely different functions than those intended by the user. (As early as 1993, the German research group *provet* demonstrated a wide variety of attacks of this type, which, in practice, do not even exploit operating system bugs. The results of this study are only available in German [Pord1_93].)

The situation becomes even worse when users are expected to insert their smartcards in devices other than their own, e.g., point-of-sales terminals at gas stations. There have even been many cases where users were tricked into inputting both their smartcard and their PIN into a fake terminal.

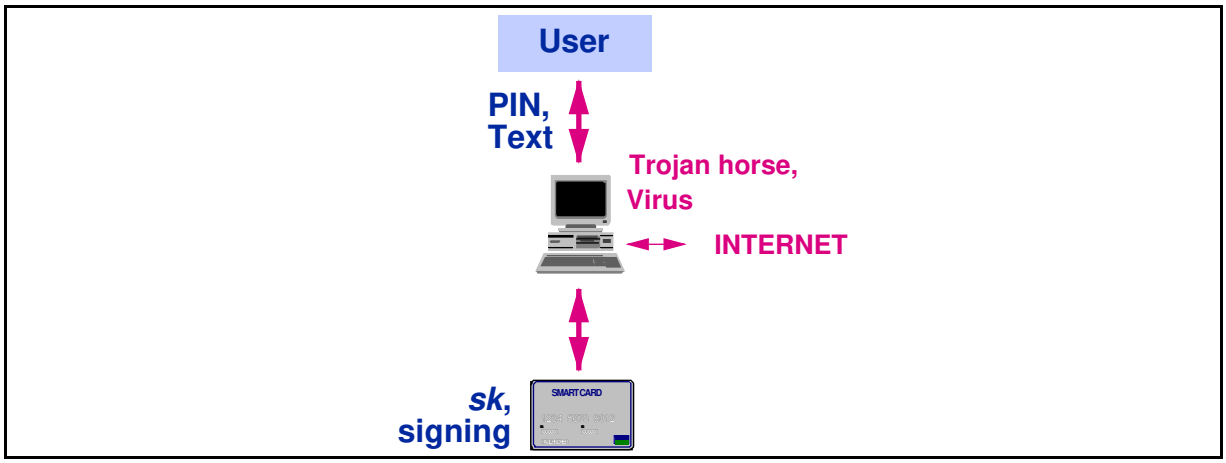

Figure 1.1: Problems with a configuration "PC + smartcard"

Such attacks can only be prevented if the user has a trusted device with a user interface, i.e., keypad and display. Thus, a portable, genuinely personal user device, with somewhat less complexity than a PC, seems to be the long-term solution.2

For quite some time, the idea of using such devices for security, as proposed, e.g., in the payment projects Mondex and CAFE [Rolf_94, BBCM1_94], was considered quite exotic. The argument against such devices was that nobody would want to pay for them. Today, however, a large percentage of the population already has mobile devices which they use for other purposes, e.g., mobile phones, pagers, and personal digital assistants (PDAs). Hence, it seems natural to combine these devices with security functions. Many of these devices already have some special security functions anyway, e.g., those used for billing mobile phone calls and related services.

In the following, the aim is to illustrate that mobile user devices and security functions are a good combination. However, the synergy of mobility and security cannot be achieved without effort. Following the life cycle of such devices is aimed at identifying all relevant issues. This is done for three different types of trust that may have to be put in a device. These types of trust are introduced in Section 1.3, and the issues related to each type are treated in Sections 1.4, 1.5, and 1.6, respectively. Among them are some unresolved and, probably, unsolvable issues, so that a judicious combination of technical measures with risk and liability management is still appropriate.

1.3 Types of Trust

How can we build trustworthy mobile user devices for applications with legal significance?

An important first point to consider is that "trustworthy" is not an objective term — for widespread, open applications in a free society, the individual users should be as free as possible in deciding whom they trust and under which circumstances. In particular, trustworthy means both that a device *is* good and that this is credible.

The following types of trust in a mobile user device have to be distinguished because they lead to somewhat different and sometimes contradictory requirements on the device. As the device *acts on behalf* of someone, we use the analogy of agents.

1. *Personal-agent trust:* For legally significant applications, a user has to be able to trust that her own device acts according to her wishes while it is in her possession. For instance, it should not sign unintended statements, nor unintentionally delete electronic money.
2. *Captured-agent trust:* The legitimate user may want her mobile user device to protect her even if it is lost or stolen, or if it temporarily leaves her possession, such as for maintenance, or when it is inserted in a point-of-sale terminal. For instance, a thief, or a person who finds the device, should not be able to sign statements in the legitimate user's name.
3. *Undercover-agent trust:* Other parties may want a mobile user device to protect their interests from the legitimate user of the device. An important example is prepaid offline payment systems, where users have so-called "electronic cash" in their mobile user devices and payments can be made in shops without a connection being made to a bank. The shops and the bank want the mobile user device to prevent its legitimate user from spending the same bit string which represents the "electronic cash" in several shops.

The second and third types of trust require tamper resistance, i.e., the device needs physical resistance against attempts to read out, or corrupt its internal data. The entire user device will not be tamper-resistant in all cases. We call a tamper-resistant component a *security module*, while *user device* refers to an entire mobile device. In principle, many configurations of user devices and security modules are conceivable — an entire user device may function as a security module, or one or more security modules might be built in or plugged into a device. It may be possible to use a plug-in security module with one, or more user devices.

In many applications undercover-agent trust, i.e., security of a device against its user, is not needed, e.g., in online payment systems, or general signature applications. The question of whether, in cases where it is needed, the same security module can be used to protect both the user and other parties is discussed below under the name of double-agent trust.

1.4 Personal-Agent Trust

In the following, the life cycle of a personal-agent device is traced in an attempt to identify all issues where design and usage of the device have to be different from other mobile user devices without any security-critical functions. The main phases in this life cycle are trustworthy design, production, shipment and personalisation of the mobile user device, followed by trustworthy interaction between the user and her device. Measures to prevent personal agents from being captured are also discussed in this section.

1.4.1 Recruiting a Good Agent

A number of standard measures, already used for building trustworthy devices for a specific corporate customer, can also be used to make the design, production, and shipment of devices for a variety of end-users trustworthy. More diversity and more inspections of shipped devices should be added to these standard measures.

Design and Production

Standard measures are structured design; quality control of all design and production steps, possibly with verification of small critical modules; and fault-tolerance measures, such as error masking, recovery, and containment, including measures against covert channels.

To enable trust by heterogeneous end users, diversity is important. For instance, end users should have the choice between different manufacturers, and each manufacturer should be supervised by independent evaluators, such as consumer organisations. All parties benefit if the manufacturer is independent of organisations that might later be engaged in lawsuits with the end users, e.g. a bank in a payment application.

Simply having a physically smaller device does not solve the problem of the operating system. However, current mobile phones and PDAs have less functionality than current PCs and users do not expect all their old software to run on them, hence, the chances of building a secure operating system for them are much better than for PCs. Hopefully, this system could then evolve in a modular way as more functionality is expected from PDAs (for some applications specialised security devices will still be needed).

Shipment

End-users have to be sure that they obtain authentic devices from the production process they trust. Naturally, one should take precautions for secure delivery. However, this is much more difficult to survey comprehensively, in particular for diverse evaluators trusted by different end users, than shipment to a few corporate customers. Hence, some checks on devices after shipping are advisable.

In order to exclude the production of fake devices by *outsiders*, all security-critical parts of a device, including the user interface, need at least some physical protection, e.g., holograms. Otherwise, an adversary could reassemble hardware and software parts from authentic devices in ways which would be virtually impossible for users to detect. A security module can additionally prove its authenticity by a challenge-response mechanism. For instance, the user receives a random challenge, c, and a response, r, from the manufacturer through a different channel than the device, e.g., by letter, whereas the device is bought at a retailer. She then verifies that the device correctly answers c by r (Figure 1.2). This measure assumes that the adversary controls, at the most, one of the two channels from the manufacturer to the user. Furthermore, it assumes that the adversary cannot establish a covert channel between the fake device that he gave the user (e.g., a mobile phone) and the real device; otherwise, the fake device could relay the challenge c to the adversary who could input it into the real device and relay the response r back to the fake device.

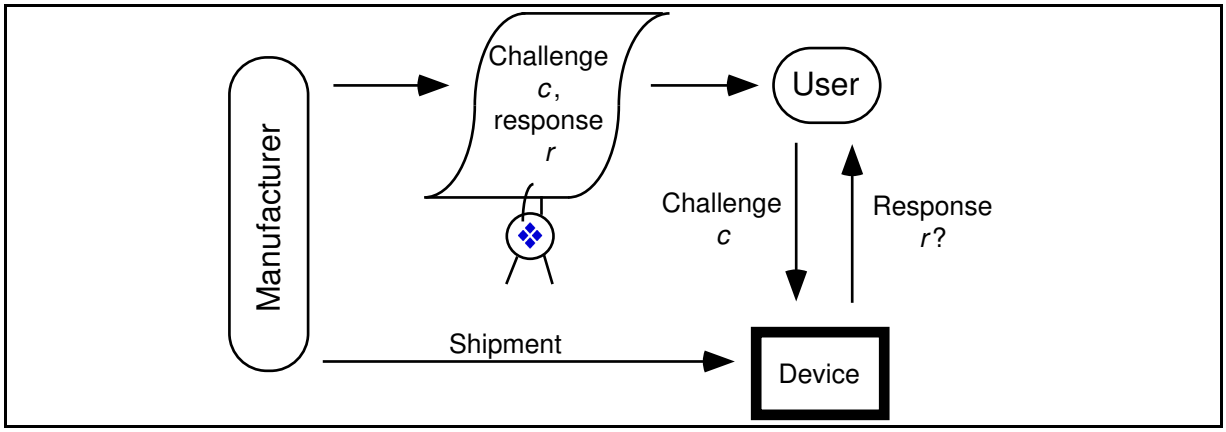

Figure 1.2: Challenge-response mechanism to verify correct shipment of a device. Here, and in the following, figures, devices outlined in bold are security modules.

It is also conceivable that insiders implement devices outside the trusted production process; these devices could pass the above tests. Where only software for a device is concerned, this can be avoided simply, by appending a digital signature of the body that certified the evaluation of this software. (Of course, the user needs a device, or a party she trusts, in order to check the certificate.) However, certificates on the hardware design do not have the same effect, because individual users cannot compare their devices with the certified design. Thus, it is necessary to rely on:

1. functional tests (black-box tests) by the end-users and
2. internal examination (white-box tests) of random samples of shipped devices by better-equipped organisations, e.g., consumer organisations.

Internal examination is a problem for security modules — it can only be carried out on parts that are not really tamper-resistant. The other main problem with functional tests is that they do not address the absence of hidden functionality, such as a 10-digit-activation code that causes a security module to output its secrets. At best, one can test that all intended functionality is present and hope that it fills the available space completely. This should be done, in particular, for the memory of security modules, which can be rewritten, e.g., by inputting random bits that fill the expected free memory completely and asking the security module to output them again (Figure 1.3). Such tests, of course, presuppose that the circuitry components of the security module are correct.

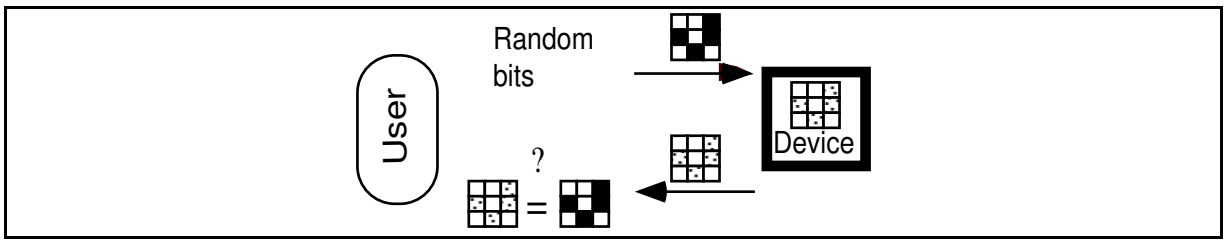

Figure 1.3: Verifying that a device has no hidden functionality in rewritable memory (9 bits shown).

For mobile user devices of different users which are intended to be indistinguishable (to guarantee privacy in applications where it is desired), a special case of hidden functionality arises regarding individual characteristics in their analogue behaviour, e.g., if one considers the exact waveforms of the device output and not only the bits. As this form of hidden functionality can probably not

be excluded, random samples of the devices which interact with the mobile user devices in privacy-critical applications should also be evaluated regularly by consumer organisations to make sure that such individual characteristics are not measured. Establishing deliberate individual characteristics, e.g., holograms, to allow each user to recognise her own device (see below) is not a problem, as long as such characteristics cannot be read by other devices.

1.4.2 Initialising an Agent

For almost all applications with security aspects, a mobile user device has to be personalised. In particular, it needs a secret key which is trusted by its user. To ensure secrecy from the start, the key should be generated in the device and never leave it. Only the corresponding public key should be output and certified at the user's request. Storage space for key generation algorithms is usually not a problem as the code can be deleted afterwards. However, all key generators need an *initial random string*, from a few dozen to a few thousand bits. To ensure the secrecy and randomness of this string, it should be produced by a collective coin flipping protocol. This means that several strings, so-called shares, are combined into one string. If at least one share is completely random and secret, the resulting string is as well. One share should come from the user; highly security-conscious users will roll dice. To protect less cautious users, the manufacturer should also contribute a share. If the device contains a physical random number generator, it should generate another share. Any further parties interested in the quality of the key, e.g., the certification authority, can contribute additional shares. Collective coin flipping in a trusted device can implemented by simply XOR-ing the shares (Figure 1.4).

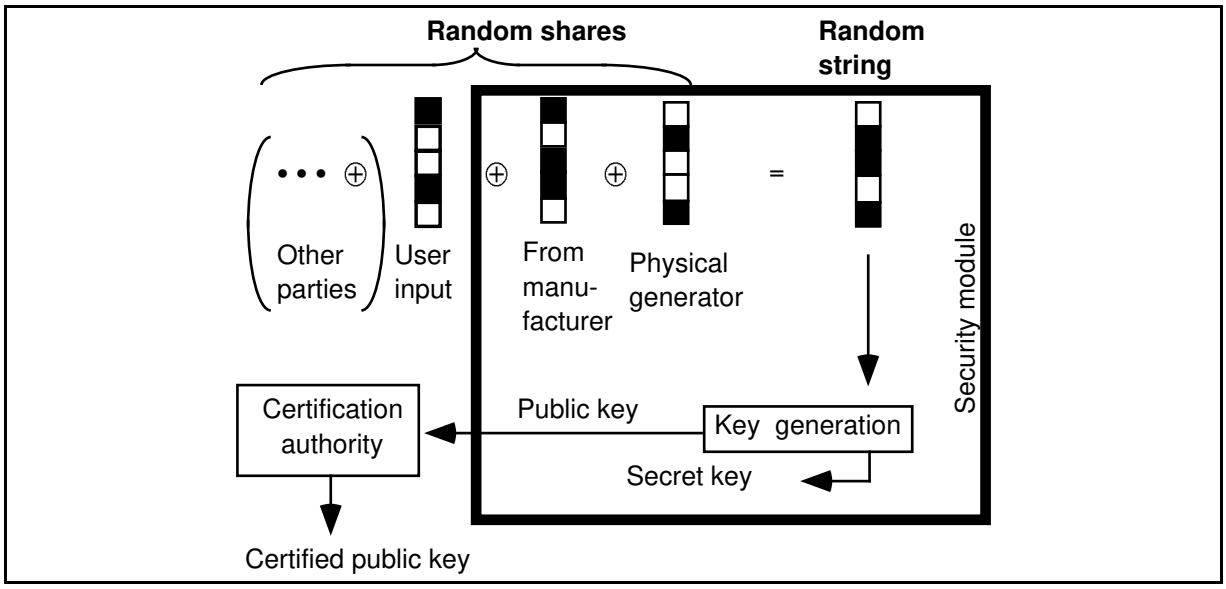

Figure 1.4: Key generation and certification using collective coin flipping in a security module.

1.4.3 Briefing and Reporting

Once a user has a personalised, trustworthy mobile user device, how can she communicate her wishes to it? Generally, the answer lies in the good design and documentation of the user interface. We do not go into ergonomics here, but we argue that some standardisation is necessary for security and consider the important example of signing with mobile user devices.

Need for Standardisation

For ordinary users, devices need a familiar and consistent user interface. Thus, the interfaces for legally significant applications have to be fairly constant, both over different applications and over time, just as the placement of the brake pedal in cars is largely standardised.

For instance, the legal significance of hand-written signatures is not only to show who produced a document, but also to warn the signer that she is making a legally binding statement. This is not only helpful for the signer, but also necessary for legally establishing that a signed document represents a declaration of intent by the signer, in contrast to being a draft, or part of a longer text. In digital applications, this warning function has to be provided by a standardised user interface. For example, the SEMPER Tinguin (Trusted Graphical User Interface) was designed for such a function [LaWW_99].

Mobile User Devices as Periscopes

A specific question, but important because of the central role of signing, is how users can be sure what they are signing if the mobile user device does not have the screen size one would usually need to compose or display a document.3 Note that a user must browse through such a document with her trusted mobile user device, otherwise this user could be tricked into signing something different where the displayed portions are identical.

Smaller documents. Obviously, application designers and users should try to keep the overall size of documents small by omitting all details which have no legal significance. Where possible, the *variable* part of a document should be reduced by using templates. The fixed part of the document, which the user does not need to look at, may include information such as a reference to an external text, e.g., a section of a statute; forms that the mobile user device obtained together with a certificate from a party the user has decided to trust, e.g., a consumer organisation; or forms the user uses repeatedly and has already read.

Good output channels. The smaller the screen is, e.g., on a mobile phone or a watch, the more important high resolution is for scrolling a small window over a larger document, in contrast to the usual small character displays,. Additional types of output channels, such as voice, may also be helpful.

Enforced browsing. To ensure that the warning function of digital signatures is not lost, the mobile user device should force the user to look at all significant parts of a document, even if the user would prefer not to. For instance, for a document with templates, it might show the fields one-by-one and ask the user to acknowledge each field individually, or even to copy the most important items correctly. For large amounts of data, e.g., a 10-page contract, this may prove infeasible. If such documents cannot be signed on paper instead, the device should at least show random samples, so that the user can detect major modifications in comparison to the full document viewed on a larger screen, and also allow additional browsing. Non-linear data, such as hypertext, would need special user guidance based on a standard for the scope of a signature in such data.

Additional partially-trusted output channels. For additional outputs in larger format, the mobile user device should contain an interface to standard output devices, such as fax machines, printers, TV sets, or computer monitors. This solution is not perfect, but quite trustworthy because there is a large choice of such devices. It is certainly better than using the computer of the recipient of the signature to view the data. An exotic measure to improve such solutions by means of

diversity is visual authentication [NaPi_97], where the user has a transparency unknown to the output device which allows him a high probability of recognising unauthorised modifications to the data output by the trusted device. Alternatively, a handscanner, which is easier to build into a mobile user device than a printer, could be used to check the printed output.

1.4.4 Keeping the Agents Free

The first rule is that users should not hand over security-critical devices to others. In particular, there should be no maintenance on personalised and functioning mobile user devices.

To prevent loss or theft of mobile user devices, they should be designed in such a way that users are encouraged to wear them rather than leaving them in a pocket or bag, or even in the office or a hotel room. Multifunctional watches are the only complete mobile device where this is currently possible. This may change if roll-up displays become reality [Brow1_94]. At present, however, we usually need small *removable security modules*. There are two types:

- *Plug-in security modules* are inserted into the mobile user device for operation. When removed from the device, most of them must be stored in small special containers.
- *External security modules* are always kept separate from the mobile user device. Hence, they need a secure communication channel to it. Radio frequency, infrared, or induction are possible options, but they restrict the places where the security module can be worn on the body.

As removable security modules only have to communicate with special mobile user devices, there is no reason to restrict their format to ISO smartcards. Modules that are smaller, sturdier, or that contain a larger chip or several chips, are already on the market. Examples include the smartcard chips on a small card, which are part of the European mobile phone standard (GSM), pendants for medical emergency data, and steel “buttons” (Figure 1.5: Different).

External security modules and containers for plug-in modules could be built into watches, bracelets, necklaces, earrings, lockets on key rings (although many people are not sufficiently careful with keys) or even rings to be worn on the fingers. Several variants should be offered, so that most people find one they like.

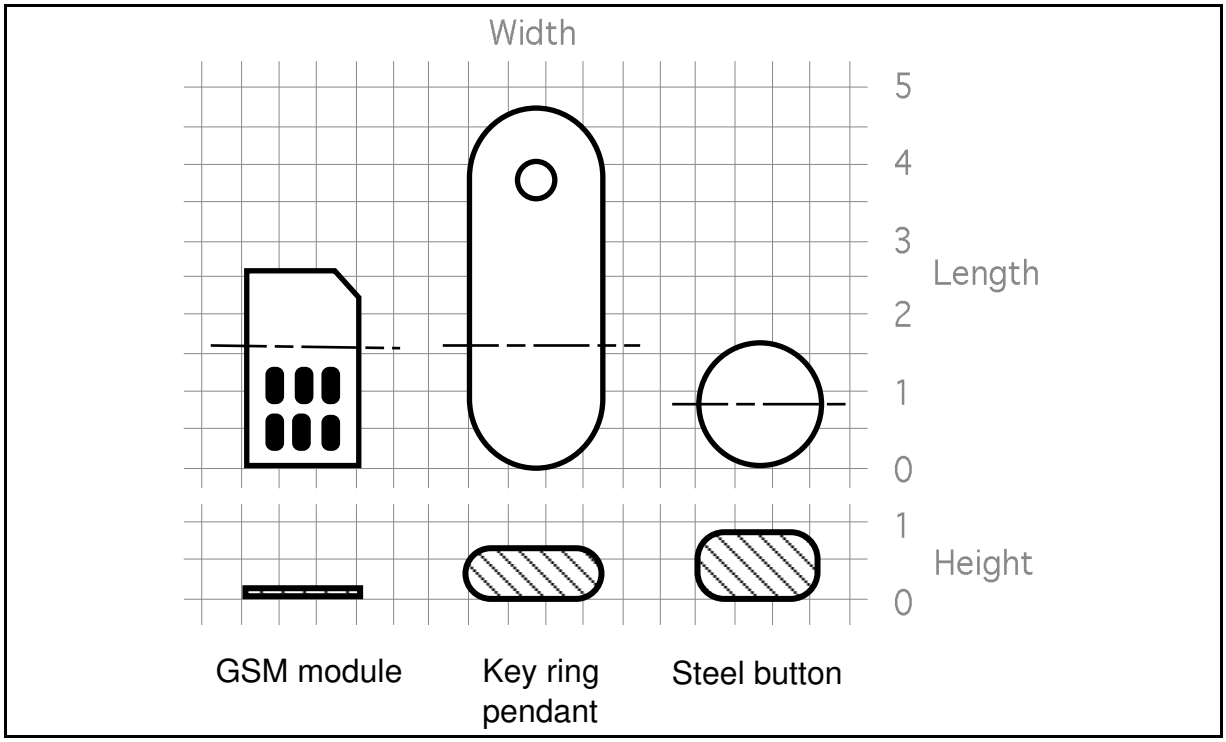

Figure 1.5: Different shapes of existing security modules.

1.5 Captured-Agent Trust

The following describes additional measures for captured-agent trust, i.e., measures to be taken if a user device is lost or stolen, in spite of the precautions suggested in Section 1.4.4. Obviously, some of the measures must already have been taken during earlier phases of the life of the device, so that they are in place when the theft or loss occurs.

1.5.1 Friend or Foe

First, a device that has fallen into the wrong hands has to be alerted. Hence, it needs user identification. The relative advantages of techniques like PINs (personal identification numbers), passphrases, and biometrics have been described in many other papers, e.g., [DaPr_89] for an overview and [ShKh_97, Lawt_98] for the current status of biometrics, therefore, the focus here is on the integration of such techniques into applications.

Ideal Times for User Identification

In principle, user identification is needed for each security-critical user command because the device could have been lost or stolen immediately before this takes place. Important cases are whenever:

- a signed statement is sent, for which the user may be made liable, or
- private data are released.

Furthermore, an adversary may be interested in keeping a stolen device running, even if no critical operation is in process, because tamper resistance is easier to break during operation. This can be avoided by additional user identification at fixed time intervals. For external security modules worn directly on the body, one can try to find a cheap form of biometrics, such as

temperature or heart beat monitoring, that only needs to notice if the security module has been removed from the body.

In practice, user identification will typically be required far less frequently, as explained in the following two paragraphs.

Managing the Risk of Observability

Some types of user identification can be observed by adversaries. In particular, PIN or password entry can be observed, and keyboards emanate radiation while the PIN is being entered. An additional problem occurs if the same identification is used not only with a person's own device, but also with devices belonging to other parties. For instance, parcel services have recipients sign on pen-pads and can, therefore, observe the dynamic signatures of a significant part of the population. In spite of the fact that biometric data are harder to input correctly than digital data, even after observation, there is still an element of risk. Better organisation can decrease, but not entirely rule out, such observability. (Fingerprints, for example, can be observed independent of their use for identification.)

At least for PINs and passwords, the risk of observation must, therefore, be weighed up against the security gained. The security-critical commands should be sorted into different risk classes, with a different type of user identification for each class. For instance, a password which allows high-value money transfers should not be put at risk by using it for low-value payments in shops. Users who can only remember one good PIN, or password, should use it exclusively for the highest risk class. For lower risk classes, even a very simple PIN is better than none. It still reduces the profit expected from stealing devices at random.

In some situations, the risk may be reduced by performing user identification before the critical command. For instance, at a gas station, one could unlock the expected amount while still sitting in the car and, later, merely confirm the exact amount by "ok," a trivial PIN, or a one-time PIN chosen in the car.

User-Friendliness

Often the frequency and strength of user identification is a compromise between user friendliness and security. The choice should largely be left to the individual user. For instance, users may be too lazy to use a PIN for spending "small" amounts. Each user should decide what she currently considers "small" by unlocking a total amount that can be spent before the PIN is needed again.

Users who like long PINs or passwords should not be restricted to 4 or 6 digits just because other users can only remember that many. This can be supported by alphanumeric PINs.

If an inexpensive type of biometrics that works with most, but not all people, is available it can be implemented. The biometrics can then be deactivated on the devices of people for whom it is ineffective, or they can be offered even less expensive devices with PINs instead. Biometrics that fail when a user is ill or in other extreme situations are not suitable for normal ecommerce applications. More generally, biometrics in such applications should be used with a high tolerance, so that the authentic users are never excluded. PINs can be used in addition for high-risk inputs.

There may be an additional emergency PIN that can be used if extortion is attempted. It should enable transactions which appear valid, but are not.

Device Identification

So far, the ways in which the mobile user device identifies its user have been discussed. For protection against fake-device attacks, the user also has to identify her device, unless she always carries the entire device around. Standard methods against the simple swapping of devices are physical characteristics like holograms, or implanted photographs, which can be recognised by the user.

If the user has a second trusted device, in particular a plug-in security module, more secure cryptographic protocols for mutual identification of devices can be used. During personalisation, the security module receives a special device identification number, DIN, from the user. For mutual identification, security module and mobile user device first identify each other by cryptographic means (Figure 1.5). If this is successful, the security module, which has no display, shows the DIN to the user via the mobile user device. If the user sees the correct DIN, she enters her PIN into the mobile user device. Now the security module can be removed. For this technique, the larger device also needs some tamper resistance. Furthermore, just as in Section 4.1, one has to ensure that an adversary cannot establish a channel between a fake device and a real device, e.g., by physical protection or so-called distance bounding protocols [BrCh_94].

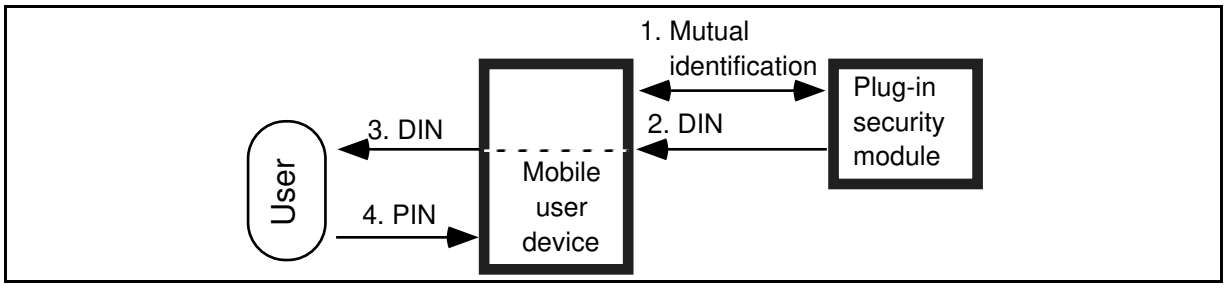

Figure 1.5:A protocol for identification of a mobile user device

These two techniques can be combined. For instance, a user might normally be satisfied with weak identification by a hologram, but use strong identification with a plug-in security module if the mobile user device has not been under her control for some time, e.g., during maintenance or in a hotel room.

Nevertheless, inadequacies in user and device identification are currently among the reasons why one should not put complete confidence in electronic transactions.

1.5.2 The Agent's Armoury

If the person who finds or steals a device cannot use it in a normal way because user identification is required, she may try to extract its secrets. For such cases, the device must be equipped with tamper resistance.

Levels of tamper resistance are not precisely measurable. Certainly, no device is tamper-*proof*. Given typical articles [Krus_91, Pate_91, AnKu_96, KoJJ_98], one can assume the following design rules for devices and applications employing security modules.

Design Issues

Tampering is much easier while a security module is operating. To minimise the risk if a device is lost in an operational state, auto shutdown should occur at the end of transactions and after a

certain period of time with no user input. However, even a stolen device must still, for example, allow three PIN entries. The information which can be observed from attempted PIN entries is minimised if each bit of the stored secret PIN is addressed and compared individually.

Different types of memory have different degrees of tamper resistance. In particular, reading the ROM of a smartcard-like security module appears relatively easy for a well-equipped attacker. Hence, security should not rely on the secrecy of algorithms, or keys, stored in ROM. Secrets in erasable memory, such as EEPROM, or battery-powered RAM, are more secure. If the security module has an internal power supply, secrets can even be deleted if an attack is detected. Fast deletion is crucial, for which battery-powered RAM is optimal. An internal battery also reduces the risk of differential power analysis [KoJJ_98]. This is a further advantage of security modules which are larger than a smartcard chip.

If the user interface of a mobile user device is outside the security module there is a risk of fake-interface attacks, i.e., an adversary replaces, or manipulates, the interface part of the device. A low-tech variant might be to exchange the lettering of the "ok" and "cancel" buttons; a high-tech variant to program the fake device to store the PIN and then to steal the fake device and the security module again. Some attacks of this sort are likely to happen; however, for mass fraud, they seem less profitable than observing PIN entry directly, or selling fake devices. Note that a secure display may be useful even without a full secure keyboard, as the user can confirm commands simply with a secure "ok" button. It is even possible to input a PIN input without any keyboard [ABKL_93]. The display shows a random number, and the user "corrects" it into the PIN.

Some applications require a reliable clock and a real random number generator. Thus, a universally applicable security module should also provide these functions.

Risk Analysis

Even with well-designed security modules, risk analysis is necessary: Do the efforts needed to break such a module and the risk of detection stand in good relation to the advantage the attacker would gain?

The risk that someone will steal and *break* devices with security modules in order to defraud individual users of low values seems relatively low, or at least lower than the risk of PIN observation. If users can transfer high values via mobile user devices, breaking tamper resistance may become worthwhile. Hence, caution is recommended in connection with home banking of large sums and general signature applications. Violating a particular user's privacy will usually not be worth the effort of breaking a security module. Security agencies, however, may be prepared to do so if the data which could be recovered would make retrospective surveillance possible.

1.6 Undercover-Agent Trust

In the following the trustworthiness of undercover agents for the third party they are designed to protect is discussed. In this case, only design and production, personalisation, and tamper-resistance have to be considered. In particular, the third party only communicates with this device via other devices, e.g., the bank via its central computers, hence, user identification and ergonomic problems do not occur. The extent to which the same mobile user device can protect both the user and a third party is considered.

1.6.1 Recruiting

The standard measures for the trustworthy design and production of undercover agents have already been mentioned above, as they are a subset of those needed for personal agents and, in practice, are actually better-developed.

Double-Agent Security Module?

In many applications, both the user and a third party have to trust a mobile user device. We speak of double-agent trust, if both trust the same security module, and of multi-agent trust if several third parties trust a user's security module , e.g., the user's bank, phone company, and employer.

The main argument for double- and multi-agent trust is cost. However, smartcard-like security modules are very cheap compared to mobile user devices, hence, the cost argument is not important for them. Furthermore, multi-agent trust is convenient, because users do not have to swap security modules between different applications. The main counter argument is that, it is a fact of life that it is unlikely that many parties will agree on one security module. In respect of double-agent trust, it is difficult to imagine that banks would trust security modules if users had the same rights to choose manufacturers and evaluators that the banks usually have. Thus, multi-agent trust is ok, if it can be achieved, but it is pointless to insist on double-agent trust, because that only decreases the number of security modules of a user interacting with n other parties from $n + 1$ to n, and forces her to trust n other parties that do not trust each other.

Furthermore, it may be impossible to implement all the required applications on a single security module. This can be supported, however, if security modules can transfer signed applications, or even encrypted data, into untrusted storage in the remainder of the mobile user device.

Interaction of Opposing Agents

If an undercover agent and a personal agent are separate parts of a mobile user device, protocols for their interaction are needed.

Basically, actions are only valid if authorised, e.g., signed, by both agents. If the undercover agent can communicate with other devices only via the personal agent, the personal agent's authorisation may simply consist of letting messages pass.

If privacy is desired for some applications, the undercover agent should not have covert channels to other devices (see Section 1.4.1). Thus, if messages from the undercover agent contain randomness, e.g., in the digital signatures, the personal agent must transform them. Examples of such protocols are the so-called wallet-with-observer protocols [ChPe1_93].

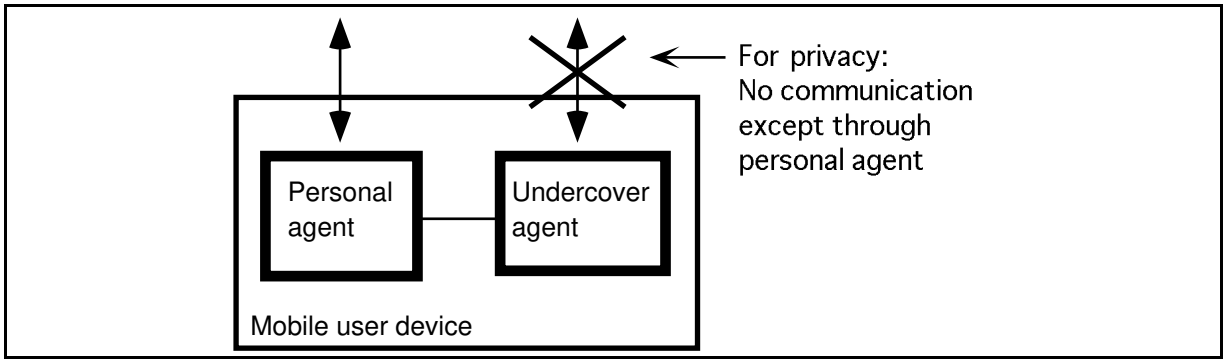

Figure 1.6: A mobile user device with separate security modules trusted by the user and another party.

1.6.2 Initialising

Double-agent trust in personalisation can be achieved simply by letting the other party contribute an input string to collective coin flipping.

1.6.3 Armoury

Tamper resistance for undercover agents is the usual example cited in the related literature. Recall that the only major difference to captured-agent trust is that an undercover agent has to operate continuously in a hostile environment.

As for risk analysis, defrauding large organisations, such as banks, may be worthwhile, even at a very high price. Hence, systems should be designed in such a way that breaking a certain number of mobile user devices does not break the system completely. This is a particular danger with off-line payment systems. Systems based on so-called digital coins, signed individually by the bank, avert this danger as far as possible [Bran_94].

1.7 Conclusion

The fact that a large variety of mobile user devices are currently entering the market for other reasons presents a great opportunity for their use in the field of security applications. These devices offer direct user interfaces, which are an important prerequisite for security, and better performance and versatility than the specialised security devices which are currently proposed. The authors differentiate between the point of view of the user of a device, using the notions of personal-agent trust and captured-agent trust, and the point of view of other parties where such parties also have to trust the mobile user device, using the notion of undercover-agent trust.

As no entirely tamper-resistant mobile user devices are currently available, it is reasonable to use less secure devices which are available from a variety of manufacturers and to supplement them with security modules. For security applications, one should concentrate on developing protocols in the hope that acceptance will be best if some people run the application in mobile phones, others in personal digital assistants, children in game boys, and only some are run in specialised security devices.

1.8 Acknowledgements

We are pleased to thank Joachim Biskup, Hannes Federrath, Phil Janson, Torsten Polle, Kai Rannenberg, Michael Schneider, and Arnd Weber for helpful discussions and comments.

Furthermore, we would like to thank our editor Dale Whinnett for considerably improving the readability of this article.

1.9 References

ABKL_93 M. Abadi, M. Burrows, C. Kaufmann, B. Lampson: Authentication and delegation with smart-cards; Science of Computer Programming 21/2 (1993) 93-113.

AnKu_96 Ross Anderson, Markus Kuhn: Tamper Resistance - a Cautionary Note; 2nd USENIX Workshop on Electronic Commerce, 1996, 1-12.

BaZi_99 Birgit Baum-Waidner, Rita Zihlmann: Legal Framework; in: Gérard Lacoste, Michael Steiner, Michael Waidner (ed.): SEMPER Final Report; to appear in LNCS, Springer-Verlag, Berlin 1999.

BBCM1_94 Jean-Paul Boly, Antoon Bosselaers, Ronald Cramer, Rolf Michelsen, Stig Mjølsnes, Frank Muller, Torben Pedersen, Birgit Pfitzmann, Peter de Rooij, Berry Schoenmakers, Matthias Schunter , Luc Vallée, Michael Waidner: The ESPRIT Project CAFE — High Security Digital Payment Systems; ESORICS 94 (Third European Symposium on Research in Computer Security), LNCS 875, Springer-Verlag, Berlin 1994, 217-230.

Bran_94 Stefan Brands: Untraceable Off-line Cash in Wallet with Observers; Crypto '93, LNCS 773, Springer-Verlag, Berlin 1994, 302-318.

BrCh_94 Stefan Brands, David Chaum: Distance-Bounding Protocols; Eurocrypt '93, LNCS 765, Springer-Verlag, Berlin 1994, 344-359.

Brow1_94 Julian Brown: Roll up for the flexible transistor; New Scientist 143/1944 (24.09.1994) 5.

ChPe1_93 David Chaum, Torben Pryds Pedersen: Wallet Databases with Observers; Crypto '92, LNCS 740, Springer-Verlag, Berlin 1993, 89-105.

DaPr_89 Donald W. Davies, Wyn L. Price: Security for Computer Networks, An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer; (2nd ed.) John Wiley & Sons, New York 1989.

KoJJ_98 Paul Kocher, Joshua Jaffe, Ben Jun: Introduction to differential power analysis; News, June 9th, 1998, see <http://www.cryptography.com/dpa>

Krus_91 Dietrich Kruse: The new Siemens computer card; Selected Papers from the Second International Smart Card 2000 Conference, North-Holland, Amsterdam 1991, 3-7.

Lawt_98 George Lawton: Biometrics: A New Era in Security; Computer 31/8 (1998) 16-18.

LaWW_99 Gérard Lacoste, Michael Waidner, Arnd Weber: Secure Electronic Commerce; in: Gérard Lacoste, Michael Steiner, Michael Waidner (ed.): SEMPER Final Report; to appear in LNCS, Springer-Verlag, Berlin 1999.

NaPi_97 Moni Naor, Benny Pinkas: Visual Authentication and Identification; Crypto '97, LNCS 1294, Springer-Verlag, Berlin 1997, 322-336.

Pate_91 Mike Paterson: Secure single chip microcomputer manufacture; Selected Papers from the Second International Smart Card 2000 Conference, North-Holland, Amsterdam 1991, 29-37.

Pord1_93 Ulrich Pordesch: Risiken elektronischer Signaturverfahren; Datenschutz und Datensicherung DuD 17/10 (1993) 561-569.

Rolf_94 Richard Rolfe: Here Comes Electronic Cash; Credit Card Management Europe, January/February 1994, 16-20.

ShKh_97 Weicheng Shen, Rajiv Khanna: Scanning the Special Issue on Automated Biometrics; Proceedings of the IEEE 85/9 (1997) 1343-1347.

---

[1] This is a revised and extended version of our article ”Trusting Mobile User Devices and Security Modules” in IEEE Computer 30/2 (1997) 61-68.

[2] Secure smartcard readers with display and buttons connected to the user’s PC are devices of the same physical complexity, only mono-functional and non-portable. The following considerations also apply to them.

[3] In addition, one should not forget that the user interface semantics of the bitstrings signed must be made clear in the signature, so that the same signed bitstring cannot have different meanings when displayed to different people. However, this does not have much to do with the devices as such.