Tamper Resistance — a Cautionary Note[*]

Ross Anderson
Markus Kuhn

Cambridge University
Computer Laboratory
Pembroke Street
Cambridge CB2 3QG
England
`rja14@cl.cam.ac.uk`

COAST Laboratory
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
U.S.A.
`kuhn@cs.purdue.edu`

Abstract

An increasing number of systems, from pay-TV to electronic purses, rely on the tamper resistance of smartcards and other security processors. We describe a number of attacks on such systems — some old, some new and some that are simply little known outside the chip testing community. We conclude that trusting tamper resistance is problematic; smartcards are broken routinely, and even a device that was described by a government signals agency as 'the most secure processor generally available' turns out to be vulnerable. Designers of secure systems should consider the consequences with care.

1 Tamperproofing of cryptographic equipment

Many early cryptographic systems had some protection against the seizure of key material. Naval code books were weighted; rotor machine setting sheets were printed using water soluble ink; and some one-time pads were printed on cellulose nitrate, so that they would burn rapidly if lit [20].

But such mechanisms relied on the vigilance of the operator, and systems were often captured in surprise attacks. So cryptographic equipment designed in recent years has often relied on technical means to prevent tampering. An example is the VISA security module, commonly used in banks to generate and check the personal identification numbers (PINs) with which customers authenticate themselves at automatic teller machines. It is basically a safe containing a microcomputer that performs all the relevant cryptographic operations; the safe has lid switches and circuitry which interrupts power to memory, thus erasing key material, when the lid is opened [27]. The idea is to deny the bank's programmers access to customer PINs and the keys that protect them; so when a customer disputes a transaction, the bank can claim that the customer must have been responsible as no member of its staff had access to the PIN [6].

Evaluating the level of tamper resistance offered by a given product is thus an interesting and important problem, but one which has been neglected by the security research community. One of the few recent articles that discuss the subject describes the design of the current range of IBM products and proposes the following taxonomy of attackers [1]:

Class I (clever outsiders): They are often very intelligent but may have insufficient knowledge of the system. They may have access to only moderately sophisticated equipment. They often try to take advantage of an existing weakness in the system, rather than try to create one.

Class II (knowledgeable insiders): They have substantial specialized technical education and experience. They have varying degrees of understanding of parts of the system but potential access to most of it. They often have highly sophisticated tools and instruments for analysis.

Class III (funded organisations): They are able to assemble teams of specialists with related and complementary skills backed by great funding resources. They are capable of in-depth analysis of the system, designing sophisticated attacks, and using the most advanced analysis tools. They may use Class II adversaries as part of the attack team.

---

[*] This paper has been published by the USENIX Association in *The Second USENIX Workshop on Electronic Commerce Proceedings*, Oakland, California, November 18–21, 1996, pp 1–11, ISBN 1-880446-83-9.

The critical question is always whether an opponent can obtain unsupervised access to the device [22]. If the answer is no, then relatively simple measures may suffice. For example, the VISA security module is vulnerable to people with occasional access: a service engineer could easily disable the tamper protection circuitry on one of her visits, and extract key material on the next. But this is not considered to be a problem by banks, who typically keep security modules under observation in a computer room, and control service visits closely.

But in an increasing number of applications, the opponent can obtain completely unsupervised access, and not just to a single instance of the cryptographic equipment but to many of them. This is the case that most interests us: it includes pay-TV smartcards, prepayment meter tokens, remote locking devices for cars and SIM cards for GSM mobile phones [4]. Many such systems are already the target of well funded attacks.

So in what follows, we will assume that all attackers can obtain several examples of the target equipment. We will also ignore tampering at the circuit board level (though this has caused losses, for example, with prepaid electricity meters [7]) and rather concentrate on attacks aimed at recovering crypto key material stored in smartcards and other chip-level security processors.

2 Breaking smartcards and microcontrollers

The typical smartcard consists of an 8-bit microprocessor with ROM, EEPROM and RAM, together with serial input and output, all in a single chip that is mounted on a plastic carrier. Key material is kept in the EEPROM.

Designers of EEPROM based devices face a problem: erasing the charge stored in the floating gate of a memory cell requires a relatively high voltage. If the attacker can remove this, then the information will be trapped.

Early smartcards received their programming voltage on a dedicated connection from the host interface. This led to attacks on pay-TV systems in which cards were initially enabled for all channels, and those channels for which the subscriber did not pay were deactivated by broadcast signals. By covering the programming voltage contact on their card with tape, or by clamping it inside the decoder using a diode, subscribers could prevent these signals affecting the card. They could then cancel their subscription without the vendor being able to cancel their service.

Some cards are still vulnerable to this kind of attack, and it gives rise to a sporadic failure mode of some card-based public telephone systems: telephones where the relevant contact is dirty or bent may fail to decrement any user's card. However, the cards used nowadays in pay-TV decoders generate the required 12 V from the normal 5 V power supply using an on-chip oscillator and diode/capacitor network. This can push up the cost of an attack, but does not make it impossible: large capacitors can be identified under a microscope and destroyed with lasers, ultrasonics or focused ion beams. A chip prepared in this way can be investigated at will without the risk of erasing the EEPROM.

So our task is to classify the various logical and physical attacks on security processors and get some idea of the cost involved.

2.1 Non-invasive attacks

Unusual voltages and temperatures can affect EEPROM write operations. For instance, for the PIC16C84 microcontroller, a trick has become widely known that involves raising VCC to VPP − 0.5 V during repeated write accesses to the security bit. This can often clear it without erasing the remaining memory.

For the DS5000 security processor, a short voltage drop sometimes released the security lock without erasing secret data. Processors like the 8752 that can be used with both internal and external memory but that limit the switch between them to resets have been read out using low voltages to toggle the mode without a reset. Low voltage can facilitate other attacks too: at least one card has an on-board analogue random number generator, used to manufacture cryptographic keys and nonces, which will produce an output of almost all 1's when the supply voltage is lowered slightly.

For these reasons, some security processors have sensors that cause a reset when voltage or other environmental conditions go out of range. But any kind of environmental alarm will cause some degradation in robustness. For example, one family of smartcard processors was manufactured with a circuit to detect low clock frequency and thus prevent single-stepping attacks. However, the wild fluctuations in clock frequency that frequently occur when a card is powered up and the supply circuit is stabilising, caused so many false alarms that the feature is no longer used by the card's operating system. Its use is left to the application programmer's discretion. Few of them bother; those who do try to use it discover the consequences for reliability. So many cards can be single-stepped with impunity.

For similar robustness reasons, the under-voltage and over-voltage detection circuitry in many devices will not react to transients. So fast signals of various kinds may reset the protection without destroying the protected information, and attacks of this kind are now known in the community for quite a number of devices.

Power and clock transients can also be used in some processors to affect the decoding and execution of individual instructions. Every transistor and its connection paths act like an RC element with a characteristic time delay; the maximum usable clock frequency of a processor is determined by the maximum delay among its elements. Similarly, every flip-flop has a characteristic time window (of a few picoseconds) during which it samples its input voltage and changes its output accordingly. This window can be anywhere inside the specified setup cycle of the flip-flop, but is quite fixed for an individual device at a given voltage and temperature.

So if we apply a clock glitch (a clock pulse much shorter than normal) or a power glitch (a rapid transient in supply voltage), this will affect only some transistors in the chip. By varying the parameters, the CPU can be made to execute a number of completely different wrong instructions, sometimes including instructions that are not even supported by the microcode. Although we do not know in advance which glitch will cause which wrong instruction in which chip, it can be fairly simple to conduct a systematic search.

A typical subroutine found in security processors is a loop that writes the contents of a limited memory range to the serial port:

```
1  b = answer_address
2  a = answer_length
3  if (a == 0) goto 8
4  transmit(*b)
5  b = b + 1
6  a = a - 1
7  goto 3
8  ...
```

We can look for a glitch that increases the program counter as usual but transforms either the conditional jump in line 3 or the loop variable decrement in line 6 into something else.

Finding the right glitch means operating the card in a repeatable way. All signals sent to it have to arrive at exactly the same time after reset for every test run. Many glitches can be tested for every clock cycle, until one of them causes an extra byte to be sent to the serial port. Repeating it causes the loop to dump the remaining memory, which if we are lucky will include the keys we are looking for.

Output loops are just one target for glitch attacks. Others are checks of passwords, access rights and protocol responses, where corruption of a single instruction can defeat the protection. A possible software countermeasure might be to avoid single-point-of failure instructions. This was common enough in the old days of unreliable hardware: a senior Cambridge computer scientist recalls that in the 1950's a prudent system programmer was someone who, having masked off three bits, would verify that the result did not exceed seven!

Hardware countermeasures include independent internal clock generators that are only PLL synchronized with the external reference frequency.

2.2 Physical attacks

Physical attacks on some microcontrollers are almost trivial. For example, the lock bit of several devices with on-chip EPROM can be erased by focusing UV light on the security lock cell, which is located sufficiently far from the rest of memory.

Current smartcards are slightly harder to attack, but not very much harder. They generally have little to prevent direct access to the silicon; the marketing director of a smartcard vendor claimed that there was simply no demand from their users for anything really sophisticated [21]. The most that appears to be done is a capacitive sensor to detect the continued presence of the passivation layer [23], or an optical sensor under an opaque coating [3]. Similar robustness considerations apply to these detectors as to the ones discussed above; they are often not used, and when they are, they are fairly easy to detect and avoid.

Anyway, the typical chip module consists of a thin plastic basis plate of about a square centimetre with conductive contact areas on both sides. One side is visible on the final card and makes contact with the card reader; the silicon die is glued to the other side, and connected using thin gold or aluminium bonding wires. The chip side of the plastic plate is then covered with epoxy resin. The resulting chip module is finally glued into the card, which is available in ISO credit card format, in miniature format for some GSM systems, or in the case of some prepayment electricity meter systems and pay-TV systems resembles a small plastic key.

Removing the chip is easy. First, we use a sharp knife or hand lathe to cut away the plastic behind

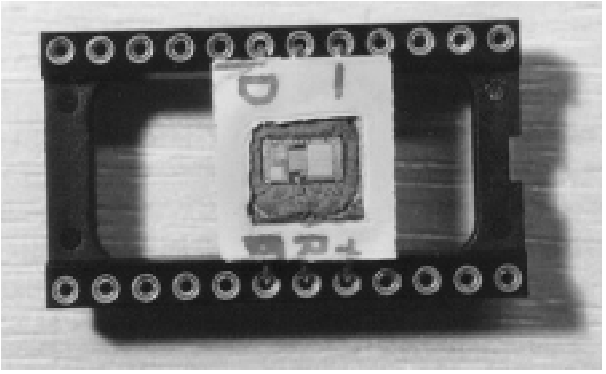

Figure 1: Fully functional smartcard processor with covering plastic removed for microprobing experiments. All tools necessary for this preparation were obtained for US$30 in a pharmacy.

the chip module until the epoxy resin becomes visible. Now we settle a few drops of fuming nitric acid (>98% HNO_3) on the resin and wait a few minutes until some of it has dissolved (the process can be accelerated by heating up the acid with an infra-red radiator). Before the acid dissolves too much epoxy and gets solid, we wash acid and resin away by shaking the card in acetone. We repeat this procedure around five to ten times until the silicon surface of the die is fully exposed. The chip can then be washed and will be fully functional unless one of the bonding wires has been damaged.

Functional tests with pay-TV and prepaid phone smartcards have shown that EEPROM content is not affected by hot nitric acid. No knowledge beyond school chemistry is necessary; the materials are easily available in any chemistry lab, and several undergraduate students have recently reported the successful application of this method on an Internet mailing list dedicated to amateur smartcard hacking. Fuming nitric acid is an aggressive oxidant and should be handled carefully (especially when using flammable liquids such as acetone), but it does not affect silicon, silicon oxide, silicon nitride, or gold as used on the chip and its contacts. The aluminium used in the metal layer of the chip is covered at once with a thin oxide layer and is also unaffected. Nitric acid is commonly used anyway to clean chip surfaces during manufacture.

There are commercial IC package removal machines used in process quality control, which expose the chip to an HNO_3 vapor stream that not only dissolves the resin but also transports away the waste products. This leaves a somewhat cleaner die surface than our manual method, but these machines use a lot of acid and need to be cleaned after use. So even professional chip analysis laboratories extract the chip manually if only a few packages have to be opened.

Most chips have a passivation layer of silicon nitride or oxide, which protects them from environmental influences and ion migration. This is not affected by nitric acid; chip testers typically remove it using dry etching with hydrogen fluoride, a process that is not as easily performed by amateur hackers.

But dry etching is not the only option. Another approach is to use microprobing needles that remove the passivation just below the probe contact point using ultrasonic vibration. Laser cutter microscopes commonly used in cellular biology laboratories have also been used to remove the passivation locally. Some testing laboratories have sets of nine microprobes so that the card bus can be read out during real time operation [12].

It is also normal to remove the passivation before using an electron beam tester to access on-chip signals, because the secondary electrons emitted by the chip surface accumulate a positive charge on the passivation layer which causes the signals to disappear after a few seconds. One might therefore think that such attacks would require dry etching facilities. However, in some experiments with an electron beam tester, we have found that the charge accumulation effect is less serious when the chip is still covered with a thin dirt layer of HNO_3 and resin remains, which is probably weakly conductive. We suggest that a suitable weakly conductive layer might be deposited on top of the passivation layer as an alternative way of preventing the charge build-up.

2.3 Advanced attack techniques

The techniques described above have been successfully used by class I attackers — amateur pay-TV hackers, students and others with limited resources. We will now briefly describe some of the techniques available in professionally equipped semiconductor laboratories, of which there are several hundred worldwide. Some of these are situated in universities (three in the UK, for example), and it has happened that class I attackers get access to professional equipment in the course of student projects.

A recent article [11] gives an overview of a technique developed for reverse engineering chips at the Cavendish Laboratory in Cambridge. The authors of that paper first developed techniques for cleanly etching away a layer of a chip at a time. One innovation is a technique to show up N and P doped layers using the Schottky effect: a thin film of a metal such as gold or palladium is deposited on the chip creating a diode which can be seen with an electron beam. Images of successive layers of a chip are then fed into a PC with image processing system software that reduces the initially fuzzy image to a clean polygon representation and identifies common chip features.

The system has been tested by reverse engineering the Intel 80386 and a number of other devices. The 80386 took two weeks, and it usually takes about six instances of a chip to get it right. The output can take the form of a mask diagram, a circuit diagram or even a list of the library cells from which the chip was constructed.

Once the layout and function of the chip are known, there is an extremely powerful technique developed by IBM for observing it in operation, without even having to remove the passivation layer. The tester places a crystal of lithium niobate over the feature whose voltage is to be monitored. The refractive index of this substance varies with the applied electric field, and the potential of the underlying silicon can be read out using an ultraviolet laser beam passed through the crystal at grazing incidence. The sensitivity of this technique is such that a 5 V signal of up to 25 MHz can be read [30], and we understand that it is a standard way for well funded laboratories to recover crypto keys from chips of known layout. When attacking a smartcard, for example, we would read the EEPROM output amplifiers.

The response of the protection community to attacks of this kind has been to develop 'conformeal glues', chip coatings that are not merely opaque and conductive but which also strongly resist attempts to remove them, usually damaging the underlying silicon in the process. These coatings are referred to in a FIPS standard [17] and are widely used by the U.S. military, but are not generally available.

In addition to chip coatings, silicon features may be used to obscure the design. We have heard of design elements that look like a transistor, but are in reality only a connection between gate and source; and 3-input NORs which function only as 2-input NORs. Such copy traps may use holes in isolating layers or tricks done in the diffusion layer with ion implantation. However, the layer etching and Schottky techniques described above can detect them.

Another possibility is to introduce complexity into the chip layout and to use nonstandard cell libraries. However the chip still has to work, which limits the complexity; and nonstandard cells can be reconstructed at the gate level and incorporated in the recognition software.

A more systematic approach was employed in the U.S. government's Clipper chip. This had a fusible link system in which the links that created a classified encryption algorithm and a long term device key from an unclassified mask were fused after fabrication, and were made of amorphous silicon to make microscopy more difficult. In addition to this, the surface of the chip was 'salted' with oscillators to make electromagnetic sensor attacks more complicated.

Details of the fusible link technology can be found in a paper in the relevant data book [18], and from the scanning electron micrographs there, it is clear that — given enough effort — the secret information can be recovered by sectioning the chip (this technique has been used by the Cambridge team on obscure features in other chips). We are reliably informed that at least one U.S. chipmaker reverse engineered the Clipper chip shortly after its launch. However the attacks that discredited the Clipper chip used protocol failures rather than physical penetration [10] — a topic to which we will return later.

Sectioning is not the only way to reverse engineer a chip whose surface is well protected. For example, a recently declassified technique invented at Sandia National Laboratories involves looking through the chip from the rear with an infra-red laser using a wavelength at which the silicon substrate is transparent. The photocurrents thus created allow probing the device's operation and identification of logic states of individual transistors [2].

The use of sectioning leads us to a more general, and relatively unexplored, topic — attacks that involve actively modifying the target chip rather than merely observing it passively. It is well known that active opponents can mount much more severe attacks on both cryptographic protocols and algorithms than passive opponents can, and the same turns out to be true when reverse engineering chips.

We understand, for example, that production attacks carried out by some pay-TV pirates involve the use of a focussed ion beam (FIB) workstation. This device can cut tracks in a chip's metallisation layer, and deposit new tracks or isolation layers. It can also implant ions to change the doping of an area of silicon, and it can even build vias to conductive structures in the lowest layers of the chip. These machines cost several million U.S. dollars, but low-budget attackers can rent time on them from various semiconductor companies.

Armed with such a tool, attacks on smartcards become much simpler and more powerful. A typical attack involves disconnecting almost all of the

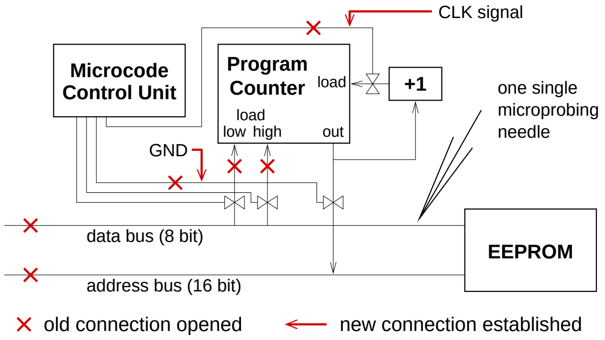

Figure 2: Read-out attack modifications on a security processor performed with a focused ion beam workstation allow easy access to secret EEPROM content with a single microprobing needle.

CPU from the bus, leaving only the EEPROM and a CPU component that can generate read accesses. For example, the program counter may be left connected in such a way that the memory locations will be accessed in order as the device is clocked (see Fig. 2). Once this has been done, the attacker needs only a single microprobing needle or electro-optical probe to read the entire EEPROM contents. This makes the program much easier to analyse than in passive attacks, which typically yield only an execution trace; it also avoids the considerable mechanical difficulties of keeping several probes simultaneously located on bus lines that are perhaps a micrometre wide.

In conclusion, it is imprudent to assume that the design of silicon chips, or the information stored in them, can be kept from a capable motivated opponent. So how can we protect key material from such an opponent?

2.4 Advanced protection techniques

One application in which capable motivated opponents may be assumed, and where billions of dollars are spent on thwarting them, is the security of nuclear weapons. The threat model here is unequivocally class III — rogue states fronted by "terrorist" commando teams operating in cahoots with subverted military personnel. The U.S.A. has led the development of a control technology, now partially shared with other nuclear and near-nuclear nations, and the following account has been pieced together from a number of open sources.

Following the Cuban missile crisis, there was concern that a world war could start by accident — for example, by a local commander under pressure feeling that 'if only they knew in Washington how bad things were here, they would let us use the bomb'. There was also concern that U.S. nuclear weapons in allied countries might be seized by the ally in time of tension, as U.S. forces there had only token custodial control. These worries were confirmed by three emergency studies carried out by Jerome Wiesner, the presidential science adviser.

President Kennedy's response was National Security Action Memo no. 160, which ordered that America's 7,000 nuclear weapons then in countries from Turkey to Germany should be got under positive control, or got out [25].

The U.S. Department of Energy was already working on safety devices for nuclear weapons, the basic principle being that a unique aspect of the environment had to be sensed before the weapon would arm. For example, missile warheads and some freefall bombs expected to experience zero gravity, while artillery shells expected to experience an acceleration of 20,000 g. There was one exception though: atomic demolition munitions. These are designed to be taken from their storage depots to their targets by jeep or helicopter, or even hand carried by special forces, and then detonated using time fuses. So there is no scope for a unique environmental sensor to prevent accidental detonation.

The solution then under development was a secret arming code, which activated a solenoid safe lock buried deep in the plutonium pit at the heart of the weapon. The main engineering problem was that when the lock was exposed, for example by a maintenance engineer replacing the power supply, the code might become known (as with the VISA security module mentioned above). So it was not acceptable to have the same code in every weapon, and group codes had to be used; the same firing code would be shared by only a small batch of warheads.

But, following the Kennedy memo, it was proposed that all nuclear bombs should be protected using code locks, and that there should be a 'universal unlock' action message that only the president or his legal successors could give. How could this be securely translated to a large number of individual firing codes, each of which would enable a small batch of weapons? The problem became worse when the Carter administration's policy of 'measured response' created a need for a wide variety of 'selective unlock' messages, giving the president options such as enabling the use of nuclear artillery and air defence weapons against a Soviet incursion into Germany. It became worse still with concern that a Soviet decapitation strike against the U.S. national command authority might leave the arsenal intact but useless. As is now well known, the solution lies in the branch of cryptomathematics known as 'secret sharing' [24], whose development it helped to inspire, and which enables weapons, commanders and options to be linked together with a complexity limited only by the available bandwidth.

In modern weapons the solenoid safe locks have been superseded by PALs — prescribed action links — about whose design details we have been able to find no useful open source material. However, it is known that PALs are considered sufficient only when they can be buried in the core of a large and complex weapon. With simple weapons (such as atomic demolition munitions) it is not considered feasible to deny access to a capable motivated opponent. These weapons are therefore stored in sensing containers called PAPS (prescribed action protective system) which provide an extra layer of protection.

Both the big-bomb and PAPS-enhanced systems include penalty mechanisms to deny a successful thief access to a usable weapon. These mechanisms vary from one weapon type to another but include gas bottles to deform the pit and hydride the plutonium in it, shaped charges to destroy components such as neutron generators and the tritium boost, and asymmetric detonation that results in plutonium dispersal rather than yield. Whatever the combination of mechanisms used in a given design, it is always a priority to destroy the code in the switch; it is assumed that a renegade government prepared to deploy "terrorists" to steal a shipment of bombs would be prepared to sacrifice some of the bombs (and some technical personnel) to obtain a single serviceable weapon.

To perform authorised maintenance, the tamper protection must be disabled, and this requires a separate unlock code. The devices that hold the various unlock codes — for servicing and firing — are themselves protected in similar ways to the weapons. We understand, for example, that after tests showed that 1 mm chip fragments survived the protective detonation of a control device carried aboard airborne command posts, the software was rewritten so that all key material was stored as two separate components, which were kept at addresses more than 1 mm apart on the chip surface.

This highlights the level of care that must be taken when developing security processors that are to withstand capable attack. This care must extend to the details of implementation and operation. The weapons testing process includes not just independent verification and validation, but hostile 'black hat' penetration attempts by competing laboratories or agencies. Even then, all practical measures are taken to prevent access by possible opponents. The devices (both munition and control) are defended in depth by armed forces; there are frequent zero-notice challenge inspections; and staff may be made to re-sit the relevant examinations at any time of the day or night.

These mechanisms and procedures have so far succeeded in preventing rogue governments from stealing (as opposed to making) atomic weapons.

The nuclear business also supplies the only examples known to us of tamper resistant packages designed to withstand a class III opponent who can obtain unsupervised physical access. These are the missile sensors developed to verify the SALT II treaty [26] — which was never deployed — and the seismic sensor package developed for test ban treaty verification, which was. In this latter system, the seismic sensors are fitted in a steel tube and inserted into a drill hole that is backfilled with concrete. The whole assembly is so solid that the seismometers themselves can be relied upon to detect tampering events with a fairly high probability. This physical protection is reinforced by random challenge inspections.

So if systems have to be protected against class III opponents, we might hazard the following summary:

- if our goal is to merely detect tampering with a positive probability (as with treaty verification), then we can allow unsupervised access provided we are allowed to use a massive construction and to perform challenge inspections;
- if we wish to prevent the loss of a cryptographic key with near certainty (as with firing codes), then we had better use explosives and we had better also guard the device.

The above analysis convinced us that military agencies have limited confidence in the ability of tamper-resistant devices (and especially portable ones) to withstand a class III opponent with unsupervised access. Having read an early draft of this paper, a senior agency official confirmed that chip contents cannot be kept from a capable motivated opponent; at most one can impose cost and delay. A similar opinion was ventured by a senior scientist at a leading chip maker.

Furthermore, the expense and inconvenience of the kind of protection used in the nuclear industry are orders of magnitude greater than even major banks would be prepared to tolerate. So what is the state of the art in commercial security processor design? They may be vulnerable to a class III opponent, but how about class II and class I?

3 Commercial security processors

Many commercial systems use either security module or smartcard technology. However, a growing number of designs consist of a composite package containing processor, memory, tamper detection circuitry and a battery.

An early example, whose design rationale was published in detail, is the μABYSS coprocessor developed by IBM. A variety of tamper resistant packages were tested for ease of penetration and ease of manufacturing, including stannic oxide lines on glass, piezo-electric sheets and a number of wire winding techniques. The designers settled on a four layer wrapping of 40 gauge (80 μm diameter) nichrome wire surrounding the processor, battery, memory and sensor circuitry, and embedded in a hard, opaque epoxy filled with silica to make it harder to machine and more likely to crack under UV laser ablation [28] [29].

This appears to be a promising technology, and increasingly so as circuit sizes and power consumption shrink. The μABYSS design protected 260 cm^2 of card circuitry, but much less is used in many recent designs. 128 kilobyte SRAM chips are available today with room temperature data retention currents of less than 1 μA. A small 3 V lithium cell can easily provide this for a decade.

Many aggressive chemicals used to remove opaque chip packages (such as fuming nitric acid) have a low electrical resistance and can easily be detected as long as battery power is available; indeed, they will often cause critical breaks and short-circuits directly. Power supply networks could be made from a variety of different conductive and isolating materials such that practically any useful chemical solvent will cause at least one part to fail.

Suitable packaging can make it difficult for the attacker to strip away the protection one layer at a time, so that a successful attack might require a highly laborious process of manually shorting out the protective wire winding, guided by X-rays and precise measurements of the voltage at various points along its length.

There are some subtleties though. One might think that the protection mechanisms only have to deactivate the power supply; but low-power SRAM chips remember bit values without a supply voltage at room temperature reliably for many seconds. By cooling the whole circuit with liquid nitrogen or helium, an attacker can extend the reliable power-off memory time to minutes or even hours, which could be enough to disable the alarm system and reapply power. Longterm exposure to a constant bit pattern can cause some SRAM cells to adapt their prefered power-up state accordingly, an effect that can remain for several days without any supply voltage [19]. Possible countermeasures include SRAM cells with a well-defined power-up behavior.

Recent examples of battery-backed security module assemblies are the IBM Transaction Security System [1] and the Dallas Semiconductor DS5000 series [16]. The latter devices have been described by a European signals security agency as the most secure processors available on general sale; we will now report a successful attack on them.

3.1 The Dallas DS5002FP Secure Microcontroller

One might want to make the tamper resistant module as small as possible, since hermetic sealing limits power dissipation, because larger packages are more vulnerable to false alarms, and for simple cost reasons.

But many applications require much more RAM than can be conveniently included in a small package, and one established technique is bus encryption [13] [14] [15]. The CPU contains hardware for on-the-fly encryption of both the external address and the data bus. External RAM contains only encrypted data stored at encrypted addresses. The secret key is stored in a special battery buffered register on the CPU chip.

The Dallas Semiconductor DS5002FP microcontroller uses this bus encryption strategy. This Intel 8051 compatible processor is used in a number of financial transaction terminals and pay-TV access control systems to store secret keys and execute confidential cryptographic algorithms. On-chip bootloader firmware allows users to upload unencrypted software; it is then encrypted and stored in the external memory. The secret key is unique to each device, which has a special self-destruct pin that allows external alarms to erase it. A special version (DS5002FPM) features an additional metal layer die top coating designed to prevent microprobe attacks.

According to the manual, this layer is a *"complex layout that is interwoven with power and ground which are in turn connected to logic for the Encryption Key and Security Logic. As a result, any attempt to remove the layer or probe through it will result in the erasure of the security lock and/or the loss of encryption key bits"*. Additional security is provided by pseudo-random dummy accesses performed on the external bus whenever the CPU core does not access external memory. In addition, 48 bytes including the reset and interrupt vectors are located on chip. Access to them also results in external dummy RAM access cycles, such that anyone observing the external bus cannot know when the internal RAM is accessed.

The security features of the DS5002FP are at first glance quite impressive and the manufacturer describes them as *"the most sophisticated security features available in any microcontroller"*.

The chip uses two block ciphers that are loosely modelled on DES. The first encrypts addresses and acts on 15-bit blocks; the second encrypts data and acts on 8-bit blocks. The key of the second cipher is salted with the address of the byte being encrypted, but its small block size (which was no doubt dictated by the fact that the controller is byte oriented) turns out to be a useful feature for the attacker.

On closer examination, the algorithms show statistical weaknesses that might allow key recovery using differential cryptanalysis. We have not studied this in detail yet. In any case the algorithm strength is a purely economic issue; more rounds can buy more strength, but at a cost in either clock frequency or transistor count. Much more interesting is a weakness of the bus encryption system that is independent of the quality of the encryption algorithms.

3.2 Breaking the Dallas chip

One of us (Kuhn) has designed and demonstrated an effective practical attack that has already yielded all the secrets of some DS5002FP based systems used for pay-TV access control and also broken a code lock provided as a challenge by the German Federal Agency for Information Technology Security (BSI). The attack requires only a normal personal computer, a special read-out circuit built from standard electronic components for less than US$100, and a logic analyzer test clip for around US$200. It was performed in a student hardware laboratory at the University of Erlangen-Nürnberg using only common laboratory tools. Designing the hardware and software and performing the experiments leading to the final approach required less than three months. Thus, in the IBM taxonomy, this attack was carried out by a class I opponent.

The basic idea is simple, but was clearly not considered by the designers or evaluators of this processor. We call it the "cipher instruction search attack": it works by feeding the CPU with suitably chosen enciphered instructions and looking to see which of the resulting plaintext instructions we can recognise from their effects.

For example, we can recognise the three byte instruction

```
MOV 90h, #42h
```

encoded 75h 90h 42h, as it outputs byte value 42h on parallel port P1 (address 90h) two bus access cycles later.

So we reset the CPU and wait until some target instruction is about to be fetched. Then our read-out hardware replaces it, and our control software observes the reaction for a few more clock cycles. Then we repeat the procedure — which we can do over 300 times per second — and systematically work through all 2^{16} combinations for the first two encrypted instruction bytes.

We eventually find a two byte combination that sends a bijective function of the following byte to the parallel port. Assuming the first two bytes are the ciphertext corresponding to 75h 90h (which has to be confirmed by further tests), this gives the data bus decryption function at the address from which the third instruction byte was fetched. By testing all 2^8 values for this byte, we can tabulate the data decryption for one address.

Now we repeat the whole process. However this time we will search for a one-byte no-operation command (NOP) followed by the same MOV instruction as before. This effectively increases by one the address from which the third MOV instruction byte will be fetched.

Although we are now searching for a combination of four encrypted bytes representing two machine instructions, the search complexity has not been increased. We know already the correct encrypted value for one byte (port address 90h) from the previous tabulation of the encryption function at this address. The first instruction does not have to be a NOP, as any one-byte instruction that does not affect the following MOV will do. So the second search loop requires considerably less than 2^{16} iterations — in fact we typically need less than 2,500 attempts.

This search process becomes steadily faster as more addresses are tabulated, and we quickly tabulate the encryption function for a number of consecutive but still unknown addresses. We are now able to encrypt and send to the processor a sequence of machine instructions that simply dumps all the memory and special registers to one of the I/O ports.

The attack is in reality somewhat more complicated than presented in this brief overview. The details will be presented in a separate publication, together with a discussion of possible countermeasures for future bus encryption based systems. Our point is that a class I attacker could circumvent the physical protection of the 'top' commercial system with modest effort. As the attack did not exploit either physical or cryptographic weaknesses, it might be considered a kind of protocol attack [8].

4 Conclusion

It is prudent engineering practice to avoid single points of failure, and especially so where the likelihood of failure is unknown. This makes it all the more remarkable that the tamper resistance claims made for smartcards and other commercial security processors have gone untested for so long. The reader will by now be convinced that these claims should be treated with circumspection.

Public key techniques offer some solace, as the number of universal secrets can be greatly reduced — ideally, to a small number of certification keys, that can then be protected in depth. However, public key protocols have their own problems [9], and we should never forget that the great majority of actual security failures result from simple blunders in design, construction and operation [6] [7]. There is no silver bullet.

A prudent engineer will see to it that the penetration of a moderate number of accessible devices, such as smartcards or payment terminals, will not be disastrous for the system. As most current electronic wallet systems use symmetric cryptography with universal secrets stored in retailers' terminals, they should be designed to keep on working after these secrets have been compromised — such as by supporting a fallback processing mode similar to that for credit cards, with full reconciliation, intrusion detection, hot lists and security recovery.

But although it is necessary to design commercial security systems with much more care, it is not sufficient. They must also be subjected to hostile testing. Readers may compare the nuclear weapons community's insistence on independent verification and validation with the aversion of the banking industry to any hostile review of their systems [5]. It is encouraging to note that some other sectors, such as the prepayment electricity meter industry, are starting to recognise the value of hostile review [7]. Other industries, such as pay-TV, got their hostile review once their systems were fielded.

Acknowledgements: This paper was largely written while the authors were guests at the Isaac Newton Institute, Cambridge. We also acknowledge useful comments from Bob Morris, Gus Simmons, Louis Guillou and a number of sources who prefer to remain anonymous; and support from the Fondazione Ugo Bordoni.

References

[1] DG Abraham, GM Dolan, GP Double, JV Stevens, "Transaction Security System", in *IBM Systems Journal* v 30 no 2 (1991) pp 206–229

[2] C Ajluni, "Two New Imaging Techniques Promise To Improve IC Defect Identification", in *Electronic Design* v 43 no 14 (10 July 1995) pp 37–38

[3] A Anderson `<aha@apollo.HP.COM>`, message posted to USENET `sci.crypt` 26 Apr 1994, message-ID `<CovCG9.581@apollo.hp.com>`

[4] RJ Anderson, "Crypto in Europe — Markets, Law and Policy" in *Cryptography: Policy and Algorithms*, Springer LNCS v 1029 pp 75–89

[5] RJ Anderson, "Liability and Computer Security: Nine Principles", in *Computer Security — ESORICS 94*, Springer LNCS v 875 pp 231–245

[6] RJ Anderson, "Why Cryptosystems Fail", in *Communications of the ACM* v 37 no 11 (Nov 94) pp 32–40

[7] RJ Anderson, SJ Bezuidenhoudt, "On the Reliability of Electronic Payment Systems", in *IEEE Transactions on Software Engineering* v 22 no 5 (May 96) pp 294–301

[8] RJ Anderson, RM Needham, "Programming Satan's Computer", in *Computer Science Today*, Springer LNCS v 1000 pp 426–441

[9] RJ Anderson, RM Needham, "Robustness Principles for Public Key Protocols", in *Advances in Cryptology — CRYPTO 95*, Springer LNCS v 963 pp 236–247

[10] M Blaze, "Protocol Failure in the Escrowed Encryption Standard", in *Proceedings of the 2nd ACM Conference on Computer and Communications Security* (2–4 November 1994), ACM Press, pp 59–67

[11] S Blythe, B Fraboni, S Lall, H Ahmed, U de Riu, "Layout Reconstruction of Complex Silicon Chips", in *IEEE Journal of Solid-State Circuits* v 28 no 2 (Feb 93) pp 138–145

[12] E Bovenlander, RL van Renesse, "Smartcards and Biometrics: An Overview", in *Computer Fraud and Security Bulletin* (Dec 95) pp 8–12

[13] RM Best, "Microprocessor for Executing Enciphered Programs", U.S. Patent No. 4,168,396, September 18, 1979

[14] RM Best, "Preventing Software Piracy with Crypto-Microprocessors", in *Proceedings of IEEE Spring COMPCON 80*, pp 466–469

[15] RM Best, "Crypto Microprocessor for Executing Enciphered Programs", U.S. Patent No. 4,278,837, July 14, 1981.

[16] *'Soft Microcontroller Data Book'*, Dallas Semiconductor, Dallas, Texas, 1993

[17] *'Security Requirements for Cryptographic Modules'*, FIPS PUB 140-1, Federal Information Processing Standards Publication, National Institute of Standards and Technology, U.S. Department of Commerce, January 11, 1994

[18] KE Gordon, RJ Wong, "Conducting Filament of the Programmed Metal Electrode Amorphous Silicon Antifuse", in *Proceedings of International Electron Devices Meeting*, Dec 93; reprinted as pp 6-3 to 6-10, *QuickLogic Data Book* (1994)

[19] P Gutmann, "Secure Deletion of Data from Magnetic and Solid-State Memory", in *Sixth USENIX Security Symposium Proceedings*, San Jose, California, July 22–25, 1996, pp 77–89

[20] D Kahn, *'The Codebreakers'* (Macmillan 1967)

[21] P Maes, marketing director, Gemplus, *comment during a panel discussion at Cardis 94*

[22] R Morris, *talk given to Cambridge Protocols Workshop*, 1994

[23] W Rankl, W Effing, *'Handbuch der Chipkarten'*, Carl Hanser Verlag, 1995; ISBN 3-446-17993-3

[24] B Schneier, *'Applied Cryptography – Protocols, Algorithms, and Source Code in C'* (second edition), John Wiley & Sons, New York, 1996

[25] GJ Simmons, invited talk at the *1993 ACM Conference on Computer and Communications Security*, Fairfax, Virginia, Nov 3–5, 1993

[26] GJ Simmons, "Subliminal Channels; Past and Present", *European Transactions on Telecommunications* v 5 no 4 (Jul/Aug 94) pp 459–473

[27] *'VISA Security Module Operations Manual'*, VISA, 1986

[28] SR White, L Comerford, "ABYSS: A Trusted Architecture for Software Protection", in *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press pp 38–51

[29] SH Weingart, "Physical Security for the μABYSS System", in *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp 52–58

[30] JM Wiesenfeld, "Electro-optic sampling of high-speed devices and integrated circuits", in *IBM Journal of Research and Development* v 34 no 2/3 (Mar/May 1990) pp 141–161; *see also subsequent articles in the same issue*