**Chapter 19**

**Side Channels**

**The hum of either army stilly sounds,**  
 **That the ﬁxed sentinels almost receive**

**The secret whispers of each others’ watch;**

**Fire answers ﬁre, and through their paly ﬂames**

**Each battle sees the other’s umber’d face.**

– WILLIAM SHAKESPEARE, KING HENRY V, ACT IV

**Optimisation consists of taking something that works, and replacing**

**it with something that almost works but is cheaper.**

– Roger Needham

**19.1** **Introduction**

Electronic devices such as computers and phones leak information in all sorts of  
 ways. A *side channel* is where information leaks accidentally via some medium  
 that was not designed or intended for communication; a *covert channel* is where  
 the leak is deliberate. Side channel attacks are everywhere, and 3–4 of them  
 have caused multi-billion dollar losses.

1. First, there are conducted or radiated electromagnetic signals, which can

compromise information locally and occasionally at longer ranges. These  
 ‘Tempest’ attacks led NATO governments to spend billions of dollars a  
 year on shielding equipment, starting in the 1960s. After the end of the  
 Cold War, people started to realise that there had usually been nobody  
 listening.

2. Second, side channels leak data between tasks on a single device, or be-

tween devices that are closely coupled; these can exploit both power and  
 timing information, and also contention for shared system resources. The  
 discovery of Differential Power Analysis in the late 1990s held up the de-  
 ployment of smartcards in banking and elsewhere by 2–3 years once it was  
 realised that all the cards then on sale were vulnerable.

3. The third multibillion-dollar incident started in January 2018 with the an-

nouncement of the ‘Spectre’ and ‘Meltdown’ attacks, which exploit spec-  
 ulative execution to enable one process on a CPU to snoop on another,  
 for example to steal its cryptographic keys. This will probably force the  
 redesign of all superscalar CPUs over 2020–5.

4. There are attacks that exploit shared local physical resources, such as when

a phone listens to keystrokes entered on a nearby keyboard, or indeed on  
 a keyboard on its own touch screen – whether that sensing is done with  
 microphones, the accelerometer and gyro, or even the camera. Another  
 example is that a laser pulse can create a click on a microphone, so a  
 voice command can be given to a home assistant through a window. So  
 far, none of the side-channel attacks on phones and other IoT devices has  
 scaled up to have major impact – but there are ever more of them.

5. Finally, there are attacks that exploit shared social resources. An example

is identifying someone in a supposedly anonymous dataset from patterns  
 of communications, location history or even just knowing when they went  
 on holiday. This has led to many poor policy decisions and much wishful  
 thinking around whether personal data can be anonymised sufficiently  
 to escape privacy law. There have been both scandalous data leaks, and  
 complaints that data should be made more available for research and other  
 uses. It’s hard to put a dollar value on this, but it is signiﬁcant in ﬁelds  
 such as medical research, as we discussed in chapter 11.

We have known about side channels for years but have consistently underesti-

mated the importance of some, while spending unreasonable sums on defending  
 against others. A security engineer who wants to protect systems long-term  
 without either overlooking real and scalable threats, or wasting money chasing  
 shadows, needs to understand the basics.

**19.2** **Emission security**

*Emission security*, or Emsec, is about preventing attacks using *compromising*  
 *emanations*, namely conducted or radiated electromagnetic signals. It’s mostly  
 military organizations that worry about *Tempest*, where the stray RF emitted  
 by computers and other electronic equipment is picked up by an opponent and  
 used to reconstruct the data being processed. It has become an issue for voting  
 machines too, after a Dutch group found they could tell at a distance which  
 party a voter had selected on a voting machine, and attacks have also been  
 demonstrated on automatic teller machines (though these don’t really scale).

Both active and passive emission security measures are closely related to

*electromagnetic compatibility* (EMC) and *radio frequency interference* (RFI),  
 which can disrupt systems accidentally, as well as *electromagnetic pulse* (EMP)  
 weapons, which disrupt them deliberately. (I discuss these in more detail in the  
 chapter on electronic warfare.) As more and more everyday devices get hooked  
 up to wireless networks, and as devices acquire more sensors, all these problems  
 – RFI/EMC, side channels and electronic warfare threats – may get worse.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**19.2.1** **History**

Crosstalk between telephone wires was well known to the 19th century tele-  
 phony pioneers, whose two-wire circuits were stacked on tiers of crosstrees on  
 supporting poles. They learned to cross the wires over at intervals to make each  
 circuit a twisted pair. Crosstalk ﬁrst came to the attention of the military in  
 1884–85, and the ﬁrst known combat exploit was in 1914. Field telephone wires  
 were laid to connect units bogged down in the mud of Flanders, and often ran  
 for miles, parallel to enemy trenches a few hundred yards away. An early WWI  
 phone circuit was a single-core insulated cable which used earth return in order  
 to halve the cable’s weight and bulk. It was soon discovered that earth leakage  
 caused crosstalk, including messages from the enemy side. Listening posts were  
 quickly established and protective measures were introduced, including the use  
 of twisted-pair cable. By 1915, valve ampliﬁers had extended the earth leakage  
 listening range to 100 yards for telephony and 300 yards for Morse code. People  
 found that the tangle of abandoned telegraph wire in no-man’s land provided  
 such a good communications channel, and leaked so much traffic, that clearing  
 it away become a task for which lives were spent. By 1916, earth return circuits  
 had been abolished within 3000 yards of the front [1380].

The intelligence community discovered side-channel attacks on cryptographic

equipment around World War 2, when Bell sold the US government a mixer to  
 add one-time tapes to telegraph traffic and discovered plaintext leaking out  
 in ciphertext. Through the 1950s, both the USA and the UK struggled to

suppress electromagnetic and acoustic emanations from their own cipher ma-  
 chines; from 1957 there was a machine, the KW-27, which was ‘reasonably well  
 protected’ against Tempest emissions. In 1960, after the UK Prime Minister  
 ordered surveillance on the French embassy during negotiations about joining  
 the European Economic Community, his security service’s scientists noticed that  
 the enciphered traffic from the embassy carried a faint plaintext signal, and con-  
 structed equipment to recover it. By the 1960s, NATO started work on Tempest  
 standards; America and Britain gave their European allies selective and incom-  
 plete security advice, so they could continue to spy on them. Meanwhile the  
 Russians developed serious proﬁciency at exploiting spurious emissions and spied  
 on all of them. When the Americans and British realised this, they used manual  
 one-time pads as a stopgap for traffic at Secret and above, then started putting  
 crypto equipment in shielded rooms in vulnerable embassies [600]. There was a  
 brief public reference to the possibility that computer data might leak in Rand  
 Corporation reports by Willis Ware in 1967 and 1970 [1985, 1986]. After that,  
 emission security became a classiﬁed topic, with secret NATO standards set by  
 1980 that were only declassiﬁed in 2000.

Meanwhile the stray RF leaking from the local oscillator signals in domestic

television sets was being targeted by direction-ﬁnding equipment in ‘TV detector  
 vans’ in Britain, where TV owners must pay an annual license fee to support  
 public broadcast services. The fact that computer data might also leak came  
 to public attention in 1985 when Wim van Eck, a Dutch researcher, published  
 an article describing how to reconstruct the picture on a VDU at a distance  
 using a modiﬁed TV set [601]. The story of the leaky French cipher machine  
 was leaked by the security service whistleblower Peter Wright in 1987 [2047].  
 Published research in emission security and related topics took off in the 1990s,

|  |  |  |
| --- | --- | --- |
|  |  |  |

as I’ll discuss shortly.

**19.2.2** **Technical Surveillance and Countermeasures**

Before we dive into the details of Tempest attacks, it is worth noting that the  
 simplest and most widespread attacks that use the electromagnetic spectrum  
 are not those exploiting unintended RF emissions of innocuous equipment, but  
 where a listening device is introduced by the attacker, or (more recently) when  
 a target’s device is compromised by malware. No matter how well it is pro-  
 tected by encryption and access controls while in transit or storage, most highly  
 conﬁdential information comes into being either as speech or as keystrokes on a  
 laptop or phone. If it can be captured by the opponent at this stage, then no  
 subsequent protective measures are likely to help very much.

An extraordinary range of bugs is available on the market:

*•* At the low end, a few tens of dollars will buy a simple radio microphone  
 the main constraint on these devices. They typically have a range of only  
 a few hundred yards, and a lifetime of days to weeks.

*•* At the next step up are devices that draw their power from the mains, a  
 indeﬁnitely. As a historical example, the UK Security Service got entry to  
 the Egyptian embassy in London during the Suez crisis and modiﬁed the  
 telephone to listen in when the clerk was entering the day’s key settings  
 into the cipher machine [600]. Some modern equivalents clip into a key-  
 board cable and look like a connector; others look like electrical adaptors  
 but send audio and video back to their owner. Police covert-entry teams  
 install such bugs in the homes and cars of serious crime suspects. Most  
 now use mobile-phone technology: they can be seen as custom handsets  
 that listen and watch when called.

*•* One exotic device, on show at the NSA Museum in Fort Meade, was pre-  
 dren. It was a wooden replica of the Great Seal of the United States, and  
 the ambassador hung it on the wall of the office in his residence. In 1952,  
 it was discovered to contain a resonant cavity that acted as a microphone  
 when illuminated by microwaves from outside the building, and retrans-  
 mitted the conversations that took place in his office. Right up to the  
 end of the Cold War, embassies in Moscow were regularly irradiated with  
 microwaves, so variants of the technique presumably remained in use.

*•* Bugs are also implanted in equipment. In 1984, sixteen bugs were dis-  
 stored eight key presses and then transmitted them in a single burst.  
 There have been many *keyloggers* designed and ﬁelded since then in key-  
 boards and keyboard cables, using a wide variety of sensors and side chan-  
 nels [1331].

|  |  |  |
| --- | --- | --- |
|  |  |  |

*•* Laser microphones work by shining a laser beam at a reﬂective or partially  
 conversation is taking place. The sound waves induce vibration in the  
 surface which modulates the reﬂected light, and this can be picked up and  
 decoded at a distance.

*•* However it’s now possible that the bulk of surveillance worldwide is done  
 by a skilled attacker, or by a coercive or manipulative family member, or  
 sometimes even as a condition of employment.

An expert in *technical surveillance countermeasures* (TSCM) will have a

whole bag of tools to provide protection against such attacks.

*•* The better *surveillance receivers* sweep the radio spectrum from about 10  
 be explained as broadcast, police, air traffic control and so on. Direct-  
 sequence spread spectrum can be spotted from its power spectrum, and  
 frequency hoppers will typically be observed at different frequencies on  
 successive sweeps. Burst transmission does better. But the effectiveness  
 of surveillance receivers is limited by the bugs that use the same frequencies  
 and protocols as legitimate mobile phones. Many organizations tried to  
 forbid the use of mobiles, but most have given up; even the Royal Navy  
 eventually had to allow sailors to keep their phones on board ship as too  
 many of them left.

*•* The *nonlinear junction detector* can ﬁnd hidden devices at close range. It  
 when the transistors, diodes and other nonlinear junctions in the equip-  
 ment rectify the signal. However, if the bug has been planted in or near  
 legitimate equipment, then the nonlinear junction detector is not much  
 help. There are also expensive bugs designed not to re-radiate at all.

*•* Breaking the line of sight, such as by planting trees around your laboratory,

*•* It’s possible to detect hidden wireless cameras that just use the normal  
 for this purpose [415].

*•* Some facilities have shielded rooms, so that even if bugs are introduced  
 material is supposed to be kept in a *secure compartmented information*  
 *facility* (SCIF) that has both physical security and acoustic shielding, and  
 is swept regularly for bugs; a SCIF may have electromagnetic shielding too  
 if a threat assessment suggests that capable motivated opponents might  
 get close enough. Shielded rooms are required in the UK for researchers  
 to access sensitive personal data held by government, such as tax records.  
 There are vendors who sell prefabricated rooms with acoustic and electro-  
 magnetic shielding. But this is harder than it looks. A new US embassy

|  |  |  |
| --- | --- | --- |
|  |  |  |

building in Moscow had to be abandoned after large numbers of micro-  
 phones were found in the structure, and Britain’s counterintelligence ser-  
 vice decided to tear down and rebuild a large part of a new headquarters  
 building, at a cost of about $50m, after an employee of one of the building  
 contractors was found to have past associations with the Provisional IRA.

*•* After the Obama administration kicked out three dozen Russian diplomats  
 Russians had even picked up conversations in unshielded SCIFs by hacking  
 officials’ phones [579].

Technological developments are steadily making life easier for the bugger

and harder for the defender. As more and more devices acquire intelligence

and short-range radio or infrared communications – as the ‘Internet of Things’  
 becomes the ‘Internet of Targets’ – there is ever more scope for attacks via  
 equipment that’s already there rather than stuff that needs to emplaced for the  
 purpose. It’s not just that your laptop, tablet or mobile phone might be running  
 creepware that records audio and uploads it later. The NSA banned Furby toys  
 in its buildings, as the Furby remembers (and randomly repeats) things said in  
 its presence. The Cayla talking doll was banned in Germany as strangers could  
 use it to listen to a child remotely, and speak to them too.

But there are many more subtle ways in which existing electronic equipment

can be exploited.

**19.3** **Passive attacks**

We’ll ﬁrst consider passive attacks, that is, attacks in which the opponent ex-  
 ploits electromagnetic signals that are presented to him without any effort on  
 his part to create them. I’ll exclude optical signals for now, and discuss them  
 along with acoustic attacks later.

Broadly speaking, there are two categories of electromagnetic attack. The

signal can either be conducted over some kind of circuit (such as a power line  
 or phone line), or it may be radiated as radio frequency energy. These are

referred to by the military as ‘Hijack’ and ‘Tempest’ respectively. They are

not mutually exclusive; RF threats often have a conducted component. For

example, radio signals emitted by a computer can be picked up by the power  
 main and conducted into nearby buildings.

**19.3.1** **Leakage through power and signal cables**

Every hardware engineer knows that high-frequency signals leak everywhere and  
 you need to work hard to stop them causing problems. Conducted information  
 leakage can be suppressed by careful design, with power supplies and signal  
 cables suitably ﬁltered. But civilian equipment only needs to be well-enough  
 shielded that it doesn’t interfere with radio and TV; it’s a much harder task to  
 prevent any exploitable leak of information.

|  |  |  |
| --- | --- | --- |
|  |  |  |

In military parlance, *red* equipment (carrying conﬁdential data) has to be

isolated by ﬁlters and shields from *black* equipment (that can send signals di-  
 rectly to the outside world). Equipment with both red and black connections,  
 such as cipher machines, is tricky to get right, and shielded equipment tends to  
 be available only in small quantities, made for government markets. But the  
 costs don’t stop there. The operations room at an air base can have hundreds of  
 cables leading from it; ﬁltering them all, and imposing strict conﬁguration man-  
 agement to preserve red/black separation, can cost millions. The contractors  
 are expensive, as the staff all need clearances – the NATO standard SDIP-20  
 for emission security (formerly AMSG 720B) is classiﬁed.

**19.3.2** **Leakage through RF signals**

When I ﬁrst learned to program in 1972 at the Glasgow Schools’ Computer Cen-  
 tre, we had an IBM 1401 with a 1.5 MHz clock. A radio tuned to this frequency  
 in the machine room would emit a loud whistle, which varied depending on  
 the data being processed. Some people used this as a debugging aid. A school  
 colleague had a better idea: he wrote a set of subroutines of different lengths  
 so that by calling them in sequence, the computer could play a tune. It never  
 occurred to us that this could be used for mischief as well as fun.

Moving now to more modern equipment, the VDUs used as monitors until

the early 2000s naturally emit a TV signal – a VHF or UHF radio signal modu-  
 lated with the image currently being displayed. The beam current is modulated  
 with the video signal, which contains many harmonics of the dot rate, some of  
 which resonate with metal components and radiate better than others. Given a  
 broadband receiver, these emissions can be picked up and reconstituted as video.  
 Wim van Eck discovered this and made it public in 1985 [601]; equipment de-  
 sign is discussed in his paper and in much more detail in [1105]. Contrary to  
 popular belief, the more modern ﬂat displays are also generally easy to snoop  
 on; a typical laptop has a serial line going through the hinge from the system  
 unit to the display and this carries the video signal (Figure 19.1).

Other researchers started to experiment with snooping on everything from

fax machines through shielded RS-232 cables to ethernet [534, 1796]. Hans-

Georg Wolf demonstrated a Tempest attack that could recover card and PIN  
 data from a cash machine at a distance of eight meters [1095]. Most business  
 sectors just ignored the problem, as countermeasures such as shielding and jam-  
 ming are difficult and expensive to do properly [143]. The military’s expertise  
 and equipment remained classiﬁed and unavailable outside the defence world.  
 Finally, in October 2006, a Dutch group opposed to electronic voting machines  
 demonstrated that the machine used to collect 90% of the election ballots in  
 the Netherlands could be eavesdropped from a distance of several tens of me-  
 ters [785]. This led to a Dutch government requirement that voting equipment  
 be Tempest-tested to a level of ‘Zone 1 - 12dB’.

The *zone* system works as follows. Equipment certiﬁed as Zone 0 should

not emit any signals that are exploitable at a distance of one meter; it should  
 protect data from electronic eavesdropping even if the opponent is in the next  
 room, and the wall is something ﬂimsy like plasterboard. Zone 1 equipment  
 should be safe from opponents at a distance of 20 meters, so the Dutch ‘Zone 1

|  |  |  |
| --- | --- | --- |
|  |  |  |

350 MHz, 50 MHz BW, 12 frames (160 ms) averaged

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  | |  |  | µV |
|  |
|  |
|  |
|  |
|  |
|  |

Figure 19.1: – RF signal from a Toshiba laptop reconstructed several rooms  
 away, through three plasterboard walls (courtesy of Markus Kuhn [1104]).

- 12dB’ criterion means that a voting machine should not leak any data on what  
 vote was cast to an eavesdropper 5 meters away. Zone 2 and Zone 3 mean 120  
 and 1200 meters respectively. Technical details of zoning were brieﬂy published  
 by the Germans in 2007, as [343]. This document was then withdrawn, perhaps  
 because the Americans objected. But everything in it was already in the public  
 domain except the zone limit curves, which are worst-case relative attenuations  
 between distances of 20m, 120m and 1200m from a small dipole or loop antenna,  
 taking into account the difference between nearﬁeld and farﬁeld dropoff. Any  
 competent RF engineer can reverse engineer the rest of it.

The zone system has come into wide governmental use since the end of the

Cold War slashed military budgets. Governments faced up to the fact that there  
 are almost no attacks, except on high-value targets to which an opponent can  
 get really close, such as diplomatic missions. The Snowden papers revealed that  
 the US’s principal Tempest target was the UN diplomatic missions in New York,  
 and even there, such techniques were only used against the handful of nations  
 whose computers couldn’t be compromised using malware.

Governments realised they had been wasting billions on shielding everything,

and cost cuts forced them to use commercial off-the-shelf (COTS) equipment for  
 almost everything. COTS equipment tends to be zone 2 when tested, with some  
 particularly noisy pieces of kit in zone 3. By knowing which equipment radiates  
 what, you can keep your most sensitive data on equipment furthest from the  
 facility perimeter, and shield stuff only when you really have to. Zoning has  
 greatly cut the costs of emission security.

Markus Kuhn and I developed a lower-cost protection technology, called ‘Soft

Tempest’, which was deployed for a while in some products, from email encryp-

|  |  |  |
| --- | --- | --- |
|  |  |  |

tion programs to Dutch voting machines [1105]. It uses software techniques to  
 ﬁlter or mask the information-bearing electromagnetic emanations from a com-  
 puter system. We discovered that most of the information-bearing RF energy  
 from a VDU was concentrated in the top of the spectrum, so we removed the  
 top 30% of the Fourier transform of a standard font by convolving it with a  
 suitable low-pass ﬁlter (see Figures 19.3 and 19.4).

|  |  |
| --- | --- |
|  |  |
| **Figure 19.3 – normal text** | **Figure 19.4 – text low-pass ﬁltered** |

This has an almost imperceptible effect on the screen contents as seen by

the user. Figures 19.5 and 19.6 display photographs of the screen with the two  
 video signals from Figures 19.3 and 19.4.

|  |  |
| --- | --- |
|  |  |

**Figure 19.5 – screen, normal text** **Figure 19.6 – screen, ﬁltered text**

However, the difference in the emitted RF is dramatic, as illustrated in the

photographs in Figures 19.7 and 19.8. These show the potentially compromising  
 emanations, as seen by a Tempest monitoring receiver.

Using Soft Tempest techniques on VDUs translated to a difference of a

zone [108]. Less can be done for modern ﬂat screens, but for some devices,

there may still be useful gains to be had.

|  |  |
| --- | --- |
|  |  |

**Figure 19.7 – page of normal text** **Figure 19.8 – page of ﬁltered text**

However, the attacker can use active as well as passive techniques. The

phenomenon we observed with the IBM 1401 – that a suitable program would  
 turn a computer into a radio broadcast transmitter – is easy to reimplement  
 on a modern computer. Figures 19.9 and 19.10 show what the screen on a PC  
 looks like when the video signal is an RF carrier at 2 MHz, modulated with pure  
 tones of 300 and 1200 Hz.

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
|  |  |

**Figure 19.9 – 300 Hz AM signal** **Figure 19.10 – 1200 Hz AM signal**

Using such tricks, malware can infect a machine that’s air-gapped from the

Internet and exﬁltrate data to a radio receiver hidden nearby [1105]. And the  
 intelligence community knew this: there had been a report of the CIA us-  
 ing software-based RF exploits in economic espionage in a TV documentary  
 in 1995 [1062]. Material declassiﬁed by the NSA in response to a FOIA re-  
 quest [986] revealed that the codeword *Teapot* refers to“the investigation, study,  
 and control of intentional compromising emanations (i.e., those that are hostilely  
 induced or provoked) from telecommunications and automated information sys-  
 tems equipment.” The possibility of malware is one reason why Tempest testing  
 involves not just listening passively to the device under test, but injecting into  
 it signals that simulate the worst-case attack in which the opponent has used a  
 software exploit to take over the device and tries to set up a covert channel [252].

The ﬁnal class of classical Emsec attacks is the exploitation of RF emana-

tions that are accidentally induced by nearby RF sources, called *Nonstop* by  
 the US military [132]. If equipment processing sensitive data is used near a mo-  
 bile phone, then the phone’s transmitter may induce currents in the equipment  
 that get modulated with sensitive data by the nonlinear junction effect and re-  
 radiated. For this reason, it used to be forbidden to use a mobile phone within  
 5 meters of classiﬁed equipment. Nonstop attacks are also the main Emsec con-  
 cern for ships and aircraft; here, an attacker who can get close enough to do a  
 passive Tempest attack can probably do much more serious harm than eaves-  
 dropping, but as military ships and aircraft often carry very powerful radios and  
 radars, one must be careful that their signals don’t get modulated accidentally  
 with something useful to the enemy. In one case, Soviet spy ships were found  
 to be listening to US military data in Guam from outside the 3-mile limit.

**19.3.3** **What goes wrong**

As Ed Snowden conﬁrmed, the Emsec threats to embassies in hostile countries  
 are real. The UK embassy in one hostile Arab country used to be on the sec-  
 ond ﬂoor of an office block whose ﬁrst and third ﬂoors were occupied by the  
 Mukhabarat, the local secret police; if that’s what you get given as diplomatic  
 premises, then shielding all electronic equipment (except that used for decep-  
 tion) will be part of the solution. It won’t be all of it; your cleaning staff will  
 be in the pay of the Mukhabarat so they will helpfully loosen your equipment’s

|  |  |  |
| --- | --- | --- |
|  |  |  |

Tempest gaskets, just as they change the batteries in the room bugs.

As for the defensive side of things, there was a scandal in April 2007 when it

emerged that Lockheed Martin had ignored Tempest standards when installing  
 equipment in US Coast Guard vessels. Documents were left on the web site  
 of the Coast Guard’s Deepwater project and ended up on an activist website,  
 cryptome.org, which was closed down for a while. The documents tell a story  
 not just of emission security defects – wrong cable types, violations of cable  
 separation rules, incorrect grounding, missing ﬁlters, red/black violations, and  
 so on – but of a more generally botched job. The ships also had hull cracks,  
 outdoor radios that were not waterproof, a security CCTV installation that did  
 not provide the speciﬁed 360 degree coverage, and much more [501]. This led  
 to a Congressional inquiry. The documents provide some insight into Tempest  
 and Nonstop accreditation procedures.

The most recent development has been Tempest attacks on smartphones.

Such devices do not have a design requirement to withstand a capable motivated  
 opponent sitting in the next room with decent radio equipment; so it should have  
 been no surprise when, in 2015, Gabriel Goller and Georg Sigl described how to  
 go about extracting private keys from smartphones at a distance using passive  
 RF monitoring [778]. The main difficulty with such attacks is that a phone’s  
 clock frequency typically varies with workload; if this frequency can somehow be  
 ﬁxed (e.g. by malware) then attacks become much easier – in fact, they reduce  
 to a standard timing attack, of a kind I will now describe.

**19.4** **Attacks between and within computers**

In the chapter on multilevel security, I remarked that Butler Lampson pointed  
 out in 1973 covert channels may allow a process at high to signal down to  
 low [1125]. As a simple example, the high process can keep some shared resource  
 busy at time *ti* to signal that the *i*-th bit of a secret key is 1. If a machine is  
 shared between high and low, and resources are not allocated in ﬁxed slices,  
 then the high process can signal by ﬁlling up the disk drive, or by using a lot of  
 CPU cycles (some people call the former case a *storage channel* and the latter a  
 *timing channel*, though in practice they can often be converted into each other).  
 There are many others such as sequential process IDs, shared ﬁle locks and last  
 access times on ﬁles – reimplementing all of these in a multilevel secure way  
 is an enormous task. It’s also possible to limit the covert channel capacity by  
 introducing noise. Some machines have had randomised system clocks for this  
 purpose. But some covert channel capacity almost always remains [808].

In classical multilevel-secure systems, it was considered a good result to get

covert channel bandwidth down to one bit per second. This would make it hard  
 to leak many Top Secret satellite images, but of course it would be trivial to  
 leak a 256-bit crypto key. This is one of the reasons the NSA was traditionally  
 suspicious of crypto in software. And covert channels are even harder to analyse  
 and block in distributed systems where the software can initiate communications  
 on the network. DNS supports covert channels, for example, which are hard  
 to block because of the service’s legitimate use, but which have been used by  
 malware to exﬁltrate credit card numbers [1371]. Such channels have easily

|  |  |  |
| --- | --- | --- |
|  |  |  |

enough bandwidth to smuggle out crypto keys.

In the mid-1990s, side-channel research was invigorated by the discovery of

novel attacks on smartcards and other crypto implementations.

**19.4.1** **Timing analysis**

In 1996, Paul Kocher showed that many implementations of public-key algo-  
 rithms such as RSA and DSA leaked key information through the amount of  
 time they took [1064]. When doing exponentiation, software typically steps

through the secret exponent one bit at a time, and if the next bit is a one it  
 does a multiply. Paul’s idea was to guess the exponent one bit at a time, work  
 through the consequences of this guess for the timing measurements, and see if  
 it reduced their variance. This clever signal-processing technique was steadily  
 reﬁned. By 2003, David Brumley and Dan Boneh implemented a timing attack  
 against Apache using OpenSSL, and showed how to extract the private key from  
 a remote server by timing about a million decryptions [330]. Some implemen-  
 tations of public-key algorithms use blinding to prevent such attacks (OpenSSL  
 did offer it as an option, but Apache didn’t use it). In fact, there was a whole  
 series of timing attacks on SSL/TLS; despite this protocol’s having been proven  
 secure in the late 1990s, there has been about one attack a year since on its  
 implementation, mostly using side channels.

Symmetric-key block ciphers are vulnerable too. John Kelsey, Bruce Schneier,

David Wagner and Chris Hall had pointed out in 1998 that Rijndael, the algo-  
 rithm that later became AES, is vulnerable to timing attacks based on cache  
 misses [1034]. The attacker can verify guesses about the output of the ﬁrst round  
 of the cipher by predicting whether the guessed value would cause a cache miss  
 on S-box lookup, and verifying this against observation. A number of researchers  
 improved this attack steadily since then, and a na¨ıve implementation of AES  
 can be broken by observing a few hundred encryptions [1489, 232, 1483]. Many  
 crypto libraries and toolkits are vulnerable; you need to work out whether they  
 are an issue for your application and if so what you’re going to do. And it’s  
 not just the algorithms that leak; protocol and implementation features such as  
 padding and error handling leak secrets too.

**19.4.2** **Power analysis**

Timing attacks can work from a distance, but if you can get up close to the  
 target equipment, there’s a lot more you can do. Smartcard makers were aware  
 from the 1980s that information could leak through the power line and patented  
 various defences; by the early 1990s, it appears to have been known to pay-TV  
 hackers and to some government agencies that information could be gathered  
 by simply measuring the current a card drew. Known as *power analysis* or *rail*  
 *noise analysis*, this may involve as little as inserting a resistor in the ground line  
 and connecting a digital storage scope across it to observe the device’s current  
 draw. An example of such a power trace can be seen in Figure 19.11. This  
 shows how a password can be extracted from a microcontroller by guessing it a  
 byte at a time and looking for a different power trace when the correct byte is

|  |  |  |
| --- | --- | --- |
|  |  |  |

guessed.

wrong inputs  
 correct input  
 difference

mA

−5

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

µs

Figure 19.11: plot of the current measured during 256 single attempts to guess  
 the ﬁrst byte of a service password stored in the microcontroller at the heart of  
 a car immobilizer (courtesy of Markus Kuhn and Sergei Skorobogatov).

Different instructions have quite different power proﬁles, and, as you can

see, the power consumption also depends on the data being processed. The  
 main data-dependent contribution in many circumstances is from the bus driver  
 transistors, which are quite large (see the top of Figure 18.7). Depending on the  
 design, the current may vary by several hundred microamps over a period of sev-  
 eral hundred nanoseconds for each bit of the bus whose state is changed [1298].  
 Thus the Hamming weight of the difference between each data byte and the  
 preceding byte on the bus (the *transition count*) is visible to an attacker. In  
 some devices, the Hamming weight of each data byte is available too [1303].  
 EEPROM reads and writes can give even stronger signals. If a wrong PIN

guess leads to a PIN-retry counter being decremented, this may cause a sharp  
 increase in current draw as a charge pump prepares to write memory (at this  
 point, an attacker might even reset the card and try another PIN).

The effect of this leakage is not limited to password extraction. An attacker

who understands (or guesses) how a cipher is implemented can obtain signiﬁcant  
 information about the card’s secrets and in many cases deduce the value of the  
 key in use. This was brought forcefully to the industry’s attention in 1998 by  
 Paul Kocher, when he adapted the signal-processing ideas developed for timing  
 attacks into an efficient technique to extract the key bits used in a block cipher  
 such as DES from a collection of power traces, without knowing any implemen-  
 tation details of the card software [1065]. This technique, known as *differential*

|  |  |  |
| --- | --- | --- |
|  |  |  |

*power analysis*, involves partitioning a set of power traces into subsets, then  
 computing the difference of the averages of these subsets. If the subsets are  
 correlated with information of interest, the difference should be nonzero [1067].

As a concrete example, the attacker might collect several hundred traces of

transactions with a target card, for which either the plaintext or the ciphertext  
 is known. They then guess some of the cipher’s internal state. In the case

of DES, each round of the cipher has eight table look-ups in which six bits of  
 the current input are xor’ed with six bits of key, and then used to look up a  
 four-bit output from an S-box. So if it’s the ciphertext to which the attacker  
 has access, they will guess the six input bits to an S-box in the last round. The  
 power traces are then sorted into two sets based on this guess and synchronized.  
 Average traces are then computed and compared. The difference between the  
 two average traces is called a *differential trace*.

The process is repeated for each of the 64 possible six-bit inputs to the target

S-box. The correct input value – which separates the power traces into two sets  
 each with a different S-box output value – will typically give a differential trace  
 with a noticeable peak. Wrong guesses, however, give randomly-sorted traces,  
 so the differential trace looks like random noise. In this way, the six keybits  
 that go to the S-box in question can be found, followed by the others used in  
 the last round of the cipher. In the case of DES, this gives 48 of the 56 keybits,  
 so the remainder can be found trivially.

The industry had not anticipated this attack, and all smartcards then on

the market appeared vulnerable [1065]. As it is a noninvasive attack, it can  
 be carried out by modiﬁed terminal equipment against a bank card carried  
 by an unsuspecting customer. So once the attacker has taken the trouble to  
 understand a card and design a Trojan terminal, a large number of cards may  
 be compromised at little marginal cost.

Paul’s discovery held up the deployment of smartcards in banking for two or

three years while people worked on defences. In fact, his company had patented  
 many of the best ones, and ended up licensing them to most crypto vendors.  
 Some work at the protocol level; for example, the EMV protocol for bank cards  
 mandates (from version 4.1) that the key used to compute the MAC on a trans-  
 action be a session key derived from an on-card master key by encrypting a  
 counter. In this way, no two ciphertexts visible outside the card are ever gener-  
 ated using the same key. Other defences include randomised clocking, to make  
 trace alignment harder, and masking, where you introduce some offsets in each  
 round and recalculate the S-boxes to compensate for them. This way, the im-  
 plementation of the cipher changes every time it’s invoked. With public-key  
 algorithms, there are even stronger arguments for masking, because they also  
 help mitigate fault attacks, which I’ll discuss below. The more expensive cards  
 have dedicated crypto engines for modular multiplication and for DES/AES.  
 Testing a device for DPA resistance is not straightforward; there is a discussion  
 by Paul Kocher at [1066] and a 2011 survey article that discusses the practical-  
 ities of attack and defence at [1067].

There are many variants on the theme. Attacks based on cache misses can

measure power as well as the time taken to encrypt, as a miss activates a lot  
 of circuitry to read nonvolatile memory; you can’t stop cache attacks on AES

|  |  |  |
| --- | --- | --- |
|  |  |  |

just by using a timer to ensure that each encryption takes the same number  
 of clock cycles. Another variant is to use different sensors: David Samyde and  
 Jean-Jacques Quisquater created *electromagnetic analysis*, in which they move  
 a tiny pickup coil over the surface of the chip to pick up local signals rather than  
 relying simply on the whole device’s current draw [1568]. And, as I noted in the  
 last chapter, DPA can be combined with optical probing; Sergei Skorobogatov’s  
 optically-enhanced position-locked power analysis uses a laser to illuminate a  
 single target transistor for half of the test runs, giving access not just to a  
 Hamming weight of a computation, but a single targeted bit [1771].

|  |
| --- |
| A spectacular demonstration of power analysis arrived in 2016 when Eyal  Ronen, Colin Oˆa˘A´ZFlynn, Adi Shamir and Achi-Or Weingarten demonstrated |
| a worm that could take over Philips Hue lamps, after they developed an im-  proved power-analysis attack to retrieve the AES key that these lamps used to  authenticate ﬁrmware updates [1614]. Philips had made several other mistakes:  relying on a single AES key, present in millions of low-cost devices, to protect  updates, using the same key for CBC and MAC, and having two bugs in the  light link protocol they used. As updates could propagate by ZigBee, malware  could spread in a chain reaction from one lamp to the next; the authors showed  that in a city such as Paris, there were enough lamps for such a chain reaction  to be self-sustaining, like nuclear ﬁssion. |

The state of the art in 2019 is probably the *template attack* where the attacker

studies a device’s current draw closely for the instructions of interest and builds  
 a multivariate Gaussian distribution giving the probability distribution for an  
 observed trace given the instruction, the operands, the results and the state.  
 For details, see for example Marios Choudary and Markus Kuhn [419]. It is also  
 possible to use special hardware tools to capture a power trace with less noise,  
 a signiﬁcant factor in power analysis [1783].

**19.4.3** **Glitching and differential fault analysis**

In 1996 Markus Kuhn and I reported that many smartcards could be broken  
 by inserting transients, or *glitches*, in their power or clock lines [106]. For

example, one smartcard used in early banking applications had the feature that  
 an unacceptably high clock frequency only triggered a reset after a number of  
 cycles, so that transients would be less likely to cause false alarms. You could  
 replace a single clock pulse with two much narrower pulses without causing a  
 reset, but forcing the processor to execute a NOP instead of the instruction it  
 was supposed to execute. This gives rise to a *selective code execution* attack  
 where the attacker can step over jump instructions to bypass access controls, or  
 construct his own program out of gadgets found in the card’s own code.

The following year, Dan Boneh, Richard DeMillo and Richard Lipton noticed

|  |  |  |
| --- | --- | --- |
| that a number of public key cryptographic algorithms break horribly if a random  error can be induced [285]. For example, when doing an RSA signature the secret  computation *S* = *h*(*m*)*d* (mod *pq*) is carried out mod *p*, then mod *q*, and the  results are then combined, as this is much faster. But if the card returns a  defective signature *Sp* which is correct modulo *p* but incorrect modulo *q*, then  we will have | | |
|  |  |  |

*p* = gcd(*pq, Sep ffi h*(*m*))

which breaks the system at once.

Also in 1997, Eli Biham and Adi Shamir pointed out that if we can set a

given bit of memory to zero (or one), and we know where in memory a key is  
 kept, we can ﬁnd out the key by just doing an encryption, zeroising the leading  
 bit, doing another encryption and seeing if the result’s different, then zeroising  
 the next bit and so on [246]. Optical probing turned out to be just the tool  
 for this [1648], and using a laser to set key bits to zero one at a time has now  
 become a routine reverse-engineering technique.

Glitches induced by lasers are not limited to attacks on chips. It turns out

that if you ﬁre a laser at a MEMS microphone, as used in phones and in voice-  
 controlled digital assistants such as Google Home and Amazon Alexa, it records  
 a click. Kevin Fu and colleagues found that by modulating a laser pointer with  
 spoken commands, they could activate such devices from tens of meters away  
 – so they could order Alexa to unlock a house’s front door by shining a laser  
 pointer through the window from the garden [1844].

Many real-world attacks now use a combination of active and passive meth-

ods. In section 19.3 above, I discussed optically enhanced position-locked power  
 analysis, which uses a laser to partially ionise a target transistor during power  
 analysis. And you can use a power glitch to greatly increase the optical emis-  
 sions from a chip for a short period of time, in order to distinguish speciﬁc  
 memory writes, as I discussed in section 18.5.5.

**19.4.4** **Rowhammer, CLKscrew and Plundervolt**

One very serious chip-level side channel is when DRAM memory contents can  
 leak into adjacent rows. In 2014, Yoongu Kim and colleagues at CMU found that  
 DRAM manufactured in 2012 and 2013 was vulnerable to disturbance errors;  
 repeatedly accessing a row in a modern DRAM chip causes bit ﬂips in physically-  
 adjacent rows at consistently predictable bit locations, an attack now known as  
 Rowhammer [1048]. The following year, Mark Seaborn and Thomas Dullien  
 found how this hardware fault could be exploited by application code to gain  
 kernel privileges [1694]. By the year after that, Kaveh Razavi and colleagues  
 had shown how to use the technique to replace a strong public key with a  
 weak one – with the effect that one virtual machine could attack a co-hosted  
 target machine by subverting its OpenSSH public-key authentication, and also  
 compromise the software update mechanism by forging GPG signatures from  
 trusted keys [1587]. The vulnerable type of DRAM is still in such wide use  
 and the attacks can target so many different software mechanisms, that they  
 may be around for some time. The ﬁrst generation of hardware mitigation from  
 vendors includes *target row refresh* (TRR) where the DRAM chip controller  
 refreshes rows to block the most common hammering patterns; Pietro Frigo and  
 colleagues built a fuzzer to analyse 42 chips with TRR defences, and found other  
 patterns that gave attacks on 13 of them [725]. And in 2020, Andrew Kwong  
 and colleagues found that the mechanism could be used to read as well as write;

|  |  |  |
| --- | --- | --- |
|  |  |  |

an attacker can exploit the dependence between Rowhammer-induced bit ﬂips  
 and the bits in adjacent rows to deduce those bits – and what’s more, this works  
 even when ECC memory detects and corrects each bit ﬂip [1114].

CPUs are also vulnerable to hardware fault injection, using dynamic scaling

of frequency and voltage. To save power, many modern CPUs change frequency  
 in response to load, and scale the voltage appropriately. In 2017 Adrian Tang,  
 Simha Sethumadhavan, and Sal Stolfo discovered the CLKscrew attack, where  
 they overclocked the Arm processor on a Nexus 6 to defeat TrustZone, extracting  
 crypto keys and escalating privilege [1858]. In 2019, Kit Murdock and colleagues  
 discovered Plundervolt: here an undocumented voltage scaling interface in Intel  
 Core processors is exploited to cause an undervoltage that induces faults in  
 multiply and AES-NI operations that allow RSA and AES keys to be extracted  
 using fault analysis, as well as mistakes in pointer arithmetic that leak arbitrary  
 memory contents from SGX exclaves [1366].

Although Arm and Intel released microcode patches for CLKscrew and Plun-

dervolt, we may expect other CPU attacks of the same genre. Rowhammer /  
 RAMBleed attacks remain an issue. In the long term, hardware security will re-  
 quire more defensive design. This will not be trivial: just increasing the DRAM  
 refresh rate increases device power consumption, as would less aggressive fre-  
 quency scaling. Two of the scientists who discovered Rowhammer, Onur Mutlu  
 and Jeremie Kim, suggest that when the memory controller closes a row, then it  
 refreshes the adjacent rows with a probability tuned to the dependability of the  
 chip [1369]. This may in turn add more complexity at the system level. Given  
 that ever more side channels will lurk in new chip technologies as ﬁrms push  
 devices ever closer to the boundaries set by physics, a more principled approach  
 is needed to semiconductor security. Chip vendors are learning the hard way  
 that they need to involve good security engineers at design time, rather than  
 just hoping to patch stuff later. When failures emerge at the level of a popular  
 semiconductor process, or a widely-used CPU, remediation is expensive.

**19.4.5** **Meltdown, Spectre and other enclave side channels**

The latest tsunami to hit the chipmakers (and indeed the whole information  
 security world) is a family of attacks based on CPU microarchitecture. The  
 story starts in 2005, when Colin Percival found that AES cache misses could be  
 used by an attacker to observe an encryption operation in another hyperthread  
 on the same Intel CPU; by pulling data into the L1 cache, then measuring a  
 moment later how long it takes to access the same data, you can see whether  
 your data were evicted by the other hyperthread [1508]. Two years later, Onur  
 Acıi¸cmez, ¸Cetin Kaya Ko¸c and Jean-Pierre Seifert invented branch prediction  
 analysis (BPA). Modern high-performance CPUs have a superscalar architecture  
 in which the CPU no longer fetches and executes one instruction at a time,  
 but has a pipeline that fetches as many as a dozen instructions ahead, and  
 tries to predict which branch the code will take. BPA enabled a spy thread  
 to extract a secret key from a parallel crypto thread by observing the CPU’s  
 branch-prediction state; a misprediction imposed a penalty of 20 cycles at the  
 time; in the best circumstances, an RSA private key could be extracted from  
 observing a single signature [13]. Others explored other cache behaviour; in

|  |  |  |
| --- | --- | --- |
|  |  |  |

2015, Fangfei Liu, Yuval Yarom and colleagues showed that the L3 cache gave  
 practical *prime and probe* cross-core attacks that enabled the recovery of GPG  
 private keys [1176]. By 2017, the Cachezoom attack allowed an attacker to

extract keys from SGX enclaves [1328]. The most recent such attack is the

Membuster attack by Dayeol Lee and colleagues, which uses OS privilege to  
 induce cache misses that leak data [1134]. (Intel’s response has been simply to  
 declare such attacks to be out of scope.) This was a ﬁeld in which, over more  
 than a decade of work, many ideas came together; the CPU vendors should have  
 been paying more attention.

The most impactful attacks were Meltdown and Spectre, disclosed in early

2018. They both exploit speculative memory reads, and build on the previous  
 work on prime-and-probe, branch prediction and cache side-channels. They are  
 so serious that both Intel and Arm announced that they will redesign their CPUs  
 to block them; but that will take years, and in the meantime software mitigations  
 (where available) may cause a 15% performance hit with some workloads, and  
 occasional reboots. Given that the world’s data centres consume perhaps 3% of  
 all electric power, this is potentially a big deal.

Meltdown creates a race condition between memory access and privilege

checking, and reads out forbidden memory via a cache side channel. It was  
 discovered independently by multiple researchers who disclosed their ﬁndings  
 responsibly to the chip makers and then consolidated their results [1172]. The  
 chip makers spent much of 2017 working secretly on bug ﬁxes.

Spectre was disclosed at the same time, having also been discovered by many

of the same teams. It’s actually a (growing) family of vulnerabilities exploiting  
 the branch prediction logic that is a special case of speculative execution. This  
 logic tries to guess which code path will be taken after a conditional jump, and  
 rogue software can train it to mispredict. The CPU will then fetch instructions  
 that will never be executed, and if some of these perform forbidden operations –  
 such as when a user program reads protected kernel memory – then the protected  
 pages may be fetched from cache. Even if they are never read – so the access-  
 control check is never done – this gives a reliable timing side-channel that enables  
 an attacker to observe crypto key material [1069]. In short, even if a CPU’s  
 execution is formally correct, all sorts of lower-level optimisations can make the  
 timing depend on secret data, and a whole series of Spectre variants have come  
 along to exploit this. While Meltdown reads a target process’s data directly,  
 Spectre tricks the target process into revealing its data via side-channels.

The Spectre family of attacks keeps on growing; shortly after Spectre was an-

nounced, researchers discovered a variant called Foreshadow that cracks many of  
 the features on Intel processors that Spectre didn’t, including SGX and system  
 management mode [338]. The 2019 security conferences brought a whole series  
 of other attacks that exploit subtle microarchitectural features: Zombieload,  
 Fallout, Smotherspectre and RAMBleed to name but four, while 2020 brought  
 Load Value Injection, which combines ideas from Meltdown and Spectre [339],  
 and CrossTalk, which enables one core in a CPU to attack another [1570]. Pretty  
 well all CPUs now use branch prediction – except the tiniest – and have become  
 so complex that there are lots of side channels. Finding them at design time isn’t  
 easy, as the tools the chipmakers developed for verifying their designs merely  
 check that the logic gives the right answer – not how long it takes. The reason

|  |  |  |
| --- | --- | --- |
|  |  |  |

they’re now being found is that the formerly sleepy backwater of microarchi-  
 tectural covert channels suddenly became the hottest topic in security research,  
 and hundreds of bright research students are suddenly looking hard. Fixing  
 everything they ﬁnd will take years, and given the nature of the technology I  
 doubt that everything will ever be ﬁxed. Arm, for example, has introduced new  
 barrier instructions CSDB, SSBB and PSSBB. After CSDB appears in code, for  
 example, no instruction may be speculatively executed using predicted data or  
 state [131]. There’s also a new data ﬁeld CVS2 from v8.5A onwards to indicate  
 the presence of mitigations against adversarial prediction training. It will take  
 perhaps four years to get this all into silicon, and several more for the neces-  
 sary support to appear in software toolchains – and longer still for programmers  
 to learn to use it all. Many programmers won’t bother, and many managers’  
 reaction to such wicked and complex problems will be denial.

So, during the 2020s, any crypto that you do on CPUs that also run untrust-

worthy processes is potentially at risk. Quite possibly all CPUs of any size will  
 acquire cryptoprocessors, with hardware engines that do AES, ECDH, ECDSA  
 and so on in constant time. (But that then opens up several new cans of worms,  
 as we’ll discuss in the chapter on Advanced Cryptographic Engineering.)

**19.5** **Environmental side channels**

The past twenty years have seen a host of side-channel attacks that exploit hu-  
 man behaviour and the environment of the device. Such attacks exploit acous-  
 tics, optics, device motion and combinations too; once attackers ﬁgure out how  
 to recover text from the sound of someone typing, they can apply the same tech-  
 niques to keystroke timings observed by other means, such as on the network  
 or by measuring device motion.

**19.5.1** **Acoustic side-channels**

Acoustic security has a long history in terms of preventing people or devices  
 eavesdropping on sensitive conversations, as I mentioned in section 19.2.2. As  
 for listening to machines, the ﬁrst case may have been during the Suez crisis  
 in 1956, when the British ﬁgured out the settings of the Egyptian embassy’s  
 Hagelin cipher machine using a phone bug. There was later a ‘folk rumour’  
 that the agencies were able to tell what someone was typing on the old IBM  
 Selectric typewriter by just recording the sound they made, and that data could  
 be recovered from the noise made by dot matrix printers [323]. It later turned  
 out that the KGB had indeed bugged IBM typewriters in the US embassy in  
 Moscow from 1976 to 1984, though they used magnetic bugs rather than micro-  
 phones [790].

In 2001, Dawn Song, David Wagner and Xuqing Tian showed that the timing

of keystrokes contained enough information for an opponent to recover a lot of  
 information merely by observing traffic encrypted under SSH. As each keystroke  
 is sent in a separate packet when SSH is used in interactive mode, encrypted  
 packet timing gives precise inter-keystroke timing and even a simple hidden  
 Markov model gives about one bit of information per keystroke pair about the

|  |  |  |
| --- | --- | --- |
|  |  |  |

content; they noted that this would enable an attacker about a factor of 50  
 advantage in guessing a password whose encrypted value he’d observed [1803].

In 2004, Dmitri Asonov and Rakesh Agrawal showed that the different keys

on a computer keyboard made sufficiently different sounds. They trained a neu-  
 ral network to recognise the clicks made by key presses on a target keyboard and  
 concluded that someone’s typing could be picked up from acoustic emanations  
 with an error rate of only a few percent [136]. In 2005, Li Zhuang, Feng Zhou,  
 and Doug Tygar combined these threads to come up with an even more powerful  
 attack. Given a recording of someone typing text in English for about ten min-  
 utes on an unknown keyboard, they recognised the individual keys, then used  
 the inter-keypress times and the known statistics of English to ﬁgure out which  
 key was which. Thus they could decode text from a recording of a keyboard to  
 which they had never had access [2072]. Other researchers quickly joined in; by  
 the following year, Yigael Berger, Avishai Wool, and Arie Yeredor had shown  
 that with improved signal-processing algorithms, acoustic reconstruction could  
 be made much more efficient [228].

Others took acoustic analysis down to a much lower level: Eran Tromer

and Adi Shamir showed that keys leak via the acoustic emanations from a PC,  
 generated mostly at frequencies above 10KHz by capacitors on the mother-  
 board [1908].

The deep neural network revolution that began in 2012 enabled much more

information to be wrung out of such signals, and by 2016 Alberto Compagno  
 and colleagues had shown that if you type while talking to someone over Skype,  
 they can reconstruct a lot of what you’re typing [464]. Also in 2016, Mengyuan  
 Li and colleagues had shown that when you type on a smartphone, your ﬁnger  
 motions interfere with the RF signal in ways that change the multipath be-  
 haviour on wiﬁ enough to modulate the channel state information; this enables  
 a rogue wiﬁ hotspot to infer keystroke information [1162]. By 2017, Ilia Shu-  
 mailov had ﬁgured out how one app on a mobile phone could recover passwords  
 and PINs typed into another app by listening to the taps on the screen, using the  
 two microphones in the device [1731]. Such *time-difference-of-arrival* (TDOA)  
 processing had previously been the domain of sophisticated electronic-warfare  
 kit; here was an application in your pocket, and that would enable a rogue  
 app to steal your online banking password, even despite the protection avail-  
 able if the password entry mechanism is implemented in the Trusted Execution  
 Environment, so malware cannot tap it directly.

**19.5.2** **Optical side-channels**

Turning now to optics, there are obvious optical side-channels such as *shoulder*  
 *surﬁng*, where someone watches your PIN over your shoulder at an ATM and  
 then picks your pocket; ATM crime gangs have also used CCTV cameras in  
 shop ceilings above a PIN entry device, and even in furniture vans parked next  
 to a cash machine. And now that everyone has a camera in their pocket and a  
 3-d printer in their den, physical keys are easy to duplicate – even by someone  
 watching at a distance. But there is much, much more.

Have you ever looked across a city at night, and seen someone working late

|  |  |  |
| --- | --- | --- |
|  |  |  |

in their office, their face and shirt lit up by the diffuse reﬂected glow from their  
 computer monitor? Did you ever stop to wonder whether any information might  
 be recovered from the glow? In 2002 Markus Kuhn showed that the answer was  
 ‘pretty well everything’: he hooked up a high-performance photomultiplier tube  
 to an oscilloscope, and found that the light from the blue and green phosphors  
 used in common VDU tubes decays after a few microseconds. As a result,

the diffuse reﬂected glow contains much of the screen information, encoded in  
 the time domain. Thus, given a telescope, a photomultiplier tube and suit-

able image-processing software, it was possible to read the computer screen at  
 which a banker was looking by decoding the light scattered from his face or his  
 shirt [1103]. (According to Ed Snowden, this was one of the techniques the NSA  
 used to spy on foreign embassies, and went under the code-name ‘Ocean’.)

The next headline was from Joe Loughry and David Umphress, who looked at

the LED status indicators found on the data serial lines of PCs, modems, routers  
 and other communications equipment. They found that a signiﬁcant number of  
 them were transmitting the serial data optically: 11 out of 12 modems tested,  
 2 out of 7 routers, and one data storage device. The designers were just driving  
 the tell-tale light off the serial data line, without stopping to realise that the  
 LED had sufficient bandwidth to transmit the data to a waiting telescope [1189].

The latest discovery, by Ben Nassi and colleagues in 2020, is the lamphone

channel. Speech or music in a room induces vibration in a hanging lightbulb,  
 which can be read from across the street using a telescope and a suitable pho-  
 todiode [1387]. Unlike a laser microphone that picks up sound from a window,  
 this is entirely passive, and the direction is less sensitive.

**19.5.3** **Other side-channels**

Thermal covert channels arrived in 2006, when Steven Murdoch discovered that  
 a typical computer’s clock skew, which can be measured remotely, showed di-  
 urnal variation, and realised this was a function of ambient temperature. His  
 experiments showed that unless a machine’s owner takes countermeasures, any-  
 one who can extract accurate timestamps from it can measure its CPU load;  
 and this raises the question of whether an attacker can ﬁnd where in the world  
 a hidden machine is located. The longitude comes from the time zone, and the  
 latitude (more slowly) from the seasons. So hiding behind an anonymity service  
 such as Tor might not be as easy as it looks [1356, 1358].

It had long been known that oily ﬁngerprint residues can compromise ﬁn-

gerprint scanners, as we discuss in the chapter on biometrics. However they  
 also leave traces on touchscreens, and after these started being used on phones,  
 Adam Aviv documented the *smudge attack*: these residues are a very effec-  
 tive way of breaking the pattern lock commonly used on Android devices [145].  
 (Smudges also help guess the PINs used on all sorts of touchscreen devices –  
 even your Tesla.)

Adam also developed the use of the smartphone’s accelerometer as a side-

channel, ﬁnding that the phone’s rocking motion as the user typed would reveal  
 signiﬁcant information. Even in uncontrolled settings, while users were walk-  
 ing, his model could classify 20% of PINs and 40% of unlock patterns within 5

|  |  |  |
| --- | --- | --- |
|  |  |  |

attempts [146]. The accelerometer had already been used by Philip Marquardt  
 and others to decode the vibrations from a nearby conventional computer key-  
 board [1229]. Liang Cai and Hao Chen then studied using both the accelerom-  
 eter and gyro, ﬁnding that the latter was more effective, and allowed a 4-digit  
 PIN to be guessed about 80 times better than by chance [365]. Laurent Simon  
 and I then played with turning the camera into a virtual gyroscope, as the phone  
 tilts when you tap in a PIN; we found that camera plus microphone was just as  
 good as the gyro for keystroke inference [1756]. Gesture typing also leaks; text  
 entered into one app can be read by others, although this is a technical side  
 channel that exploits shared interrupt state [1759].

The arrival of the Apple watch in 2015 inspired more people to study smart-

watch side channels; by the end of the year, Xiangyu Liu and colleagues had  
 shown that a smartwatch not only allows you to do the accelerometer inference  
 attacks on smartphone PIN entry, but also to reconstruct text typed at a normal  
 keyboard – though if you wear it on your left wrist you get more accuracy with  
 the left-hand letters [1177].

Are these side channels a big deal? The answer appears to be ‘not yet’. Joel

Reardon and colleagues studied 88,000 apps from the Google Play Store and  
 reported in 2019 that while over 12,000 had the means to exploit side channels  
 to observe other apps or system data, or to communicate in ways that they  
 shouldn’t, only 61 actually did so [1588]. However, the security engineer must  
 remain aware that as we move to devices such as smartphones with a rich set  
 of sensors, we get a rich set of side channels that make it ever more difficult to  
 conﬁne information to speciﬁc apps and contexts. As we move to a world with  
 gazillions of smart objects, the number and type of side channels will multiply.  
 We might expect this to give us a nasty surprise one day.

**19.6** **Social side channels**

Many side channels occur at the application layer, and are often overlooked.  
 One classic example is an increase in pizza deliveries to the Pentagon leaking  
 the fact of a forthcoming military operation. A more subtle example is that  
 personal health information derived from visits to genitourinary medicine clinics  
 is considered specially sensitive in the UK, and can’t be shared with the GP  
 unless the patient consents. In one case, a woman’s visit to a GUM clinic leaked  
 when the insurer failed to recall her for a smear test that her GP knew was  
 due [1310]. The insurer knew that a smear test had been done already by the  
 clinic, and didn’t want to pay twice.

I’ve already discussed such issues at length in the chapter on Inference Con-

trol and don’t propose to duplicate that discussion here. I’ll merely note that  
 this is also a high-impact family of side channels. Policymakers and the tech  
 industry have both pretended for years to believe that de-identiﬁcation of sen-  
 sitive data such as medical records makes it non-sensitive and thus suitable to  
 be treated as an industrial raw material. This is emphatically not the case, as  
 one scandal after another has brought home – leading among other things to  
 the EU General Data Protection Regulation.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Social side channels also play a role on the philosophical side of technology

policy debates; for example, Helen Nissenbaum has gone so far as to deﬁne  
 privacy as ‘contextual integrity’. Most privacy failures that do real harm result  
 from information from one context (such as the clinic) ending up in another  
 (such as a newspaper). Ubiquitous devices with complex side channels are not  
 the only issue; the mass collection of data that’s used for advertising without  
 effective opt-outs leads to much more leakage. I’ll discuss this later in the

chapter on ‘Surveillance or Privacy?’

**19.7** **Summary**

Side-channel attacks include a whole range of threats in which the security of  
 systems can be subverted by compromising emanations, whether from uninten-  
 tional radio frequency or conducted electromagnetic signals, to leakage through  
 shared computational state, to the wide range of sensors found in modern mo-  
 bile phones and other consumer devices and to leakage via social context too.  
 Side channel leakage is a huge topic and it will get more complex still as we get  
 software and sensors in just about everything. Which side channels pose a real  
 threat will of course depend on the application, and most of them will remain  
 of academic interest most of the time. But occasionally, they’ll bite. So the  
 security engineer needs to be aware of the risks.

**Research Problems**

Many of the research papers in the top security conferences in 2019 are about  
 side channels, particularly side-channel attacks on processors that undermine  
 access controls and enclaves, and side-channel attacks on security chips that  
 enable TPMs or payment cards to be defeated. Back in 2015, the emphasis was  
 on side-channel attacks on phones, smart watches and other physical devices.  
 Social side-channels continue to be of interest and drive research into privacy.

Side-channel vulnerabilities are becoming ubiquitous as systems get more

complex. More complex supply chains made bug ﬁxes harder, and sometimes  
 vulnerabilities just won’t be ﬁxed as it would cost too much in terms of per-  
 formance, effort or cash. Attacks become easier as techniques are honed and  
 software gets passed around. This applies to classical Tempest attacks too,

as *software radios* – radios that digitize a signal at the intermediate frequency  
 stage and do all the subsequent processing in software – are no longer an ex-  
 pensive military curiosity [1117] but are now ubiquitous in cellular radio base  
 stations, GPS receivers, IoT devices, and even hobbyists’ bedrooms. The ex-  
 plosion of interest in machine learning is bound to have an effect, improving  
 attacks everywhere from Tempest through power analysis to the exploitation of  
 social channels. It’s hard to predict which side channels will scale up to become  
 another billion-dollar issue, but it’s a good bet that some of them will.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Further Reading**

A recent history of Tempest by David Easter tells of the Cold War struggles  
 between Russia, the USA, the UK and their European allies [600]. The classic  
 van Eck article [601] is still worth a read, and our work on Soft Tempest, Teapot  
 and related topics can be found in [1105]. For power analysis, see the papers  
 by Paul Kocher [1065] and Thomas Messergues [1298]. For timing and power  
 analysis, the original papers by Paul Kocher and his colleagues are the classic  
 references [1064, 1065]; there’s a textbook by Stefan Mangard, Elisabeth Oswald  
 and Thomas Popp that covers all the major aspects [1214], while Paul Kocher’s  
 2011 survey paper, “Introduction to differential power analysis” explains the  
 engineering detail of both attack and defence [1067]. A 2020 survey by Mark  
 Randolph and William Diehl covers more recent work [1576].

To keep up with progress in timing and power attacks on security chips,

you really need to follow the current research literature, as attack techniques  
 improve all the time. For example, in November 2019, Daniel Moghimi, Berk  
 Sunar, Thomas Eisenbarth and Nadia Henninger found timing attacks on a  
 TPM made by STM that had been certiﬁed secure to Common Criteria EAL4+  
 and on a virtual TPM in Intel CPUs, enabling them to extract ECDSA keys;  
 the latter case led to a real attack on a VPN product [1329]. More than twenty  
 years after timing attacks came along, you still can’t rely on either certiﬁed  
 products or big brand names to withstand them.

Attacks on mainstream computer hardware are still developing quickly. For

attacks on memory, see the 2019 survey paper on Rowhammer by Onur Mutlu  
 and Jeremie Kim [1369]. As for attacks on CPUs exploiting speculative ex-

ecution, the Meltdown and Spectre attacks attracted so much publicity that  
 microarchitectural security turned overnight from a backwater into one of the  
 hottest research areas in the ﬁeld. For years the CPU designers (and almost  
 everyone else) had assumed that if hardware had been veriﬁed, then it did what  
 it said in the manual, so there was no point looking for bugs. Now we know that  
 the veriﬁcation tools had nothing to say about side channels, there are hundreds  
 of smart people beating up on CPUs. The bug reports just keep on coming, and  
 CPUs have meanwhile got so complex that it may take years before we get some  
 stability. The best starting point in 2019 is probably the survey paper by Clau-  
 dio Canella and colleagues at the Usenix Security Symposium [380]. Claudio  
 and colleagues have also broken the ﬁrst-generation Meltdown mitigations with  
 an attack called EchoLoad [381].

|  |  |  |
| --- | --- | --- |
|  |  |  |