Chapter 22

Phones

I rarely had to resort to a technical attack. Companies can spend millions of dollars toward technological protections and that’s wasted if somebody can basically call someone on the telephone and either convince them to do something on the computer that lowers the computer’s defenses or reveals the information they were seeking.

– KEVIN MITNICK

Privacy is not about hiding – privacy is about human growth and agency.

– CHRISTOPHER WYLIE

The protection of phones, the app ecosystem they support and the telecommunications networks on which they rely, is central to the modern world. First, in the decade after the launch of the iPhone, the world moved from accessing the Internet via PCs or laptops to using smartphones instead, and added billions of new users too. Whole business sectors are being revolutionised as they move to apps; of the 5.5bn adults on earth, 5bn have phones, and 4bn of them have smartphones. Second, the new generation of connected devices, from smart speakers to cars, are very much like phones, often using the same platforms and sharing the same vulnerabilities. Third, phones now provide the bedrock for authentication: if you forget your password, you get an SMS to recover it – so someone who can steal an SMS from you may be able to spend your money. Fourth, phone networks are critical to other infrastructure: electricity companies rely on phones to direct their engineers when repairing faults, so if the phone system goes down a few hours after the power does, there’s a real problem. Finally, there’s public policy. While smartphones have revolutionised the lives of the third-world poor by giving access to services such as banking, they also facilitate surveillance and control.

The phone ecosystem is mind-numbingly complex, and to master it the security engineer needs not just general security knowledge such as crypto and access controls, and knowledge of specific platforms such as Android and IoS, but of telecomms too. The history of telecomms security is instructive. Early
attacks were carried out on phone companies by enthusiasts (‘phone phreaks’) to get free calls; then the phone system’s vulnerabilities were exploited by crooks to evade police wiretapping; then premium rate calls were introduced, which brought in large-scale fraud; then when telecomms markets were liberalized, some phone companies started conducting attacks on each other’s customers; and some phone companies have even attacked each other. At each stage the defensive measures undertaken tended to be inadequate for various reasons. The same cycle of exploitation then repeated with the Internet – amateur hackers followed by debates about wiretaps followed by fraud and tussles between companies and users; and as the two came together we’ve seen lots of complex interactions. Now we see rapidly growing phone-based fraud against banking systems, bad apps stealing people’s personal information and high policy debates on the national security implications of 5g infrastructure. How is the security engineer to navigate this?

The security of the phone as a platform depends on a number of things, which I’ll deal with under two main headings.

1. First, there’s whether the network to which it’s attached has somehow been compromised, whether by some kind of wiretap or by a SIM swap attack which undermines its network identity.

2. Second, there’s the question of whether the device itself has been compromised, whether by malware rooting the operating system, or by the installation of a hostile application or library.

### 22.1 Attacks on phone networks

The abuse of communications goes back centuries. Before Sir Rowland Hill invented the postage stamp, postage was paid by the recipient. Unsolicited mail became a huge problem – especially for famous people – so recipients were allowed to inspect a letter and reject it rather than paying for it. People soon worked out schemes to send short messages on the covers of letters which their correspondents rejected. Regulations were brought in to stop this, but were never really effective [1326].

A second set of abuses developed with the telegraph. Early optical telegraphs worked using semaphores or heliographs; people would bribe operators, or ‘hack the local loop’ by observing the last heliograph station through a telescope, to learn which horse had won before the local bookmaker did. Here too, attempts to legislate the problem away were a failure [1651]. The problems got worse when the electric telegraph brought costs down: the greater volumes of communication, and the greater flexibility that got built into and on top of the service, led to more complexity and more abuse.

The telephone was to be no different.

### 22.1.1 Attacks on phone-call metering

Early phone-call metering systems were open to creative abuse.
In the 1950’s, the operator in some systems had to listen for the sound of coins dropping on a metal plate to tell that a callbox customer had paid, so people practised hitting the coinbox with a piece of metal that struck the right note.

Initially, the operator had no way of knowing which phone a call had come from, so she had to ask the caller his number. He could give the number of someone else – who would then be charged. Operators started calling back to verify the number for international calls, so people worked out social engineering attacks (‘This is IBM here, we’d like to book a call to San Francisco and because of the time difference can our Managing Director take it at home tonight? His number’s xxx-yyyy’). So payphone lines had a warning to alert the operator. But the UK implementation had was a bug: a customer who had called the operator from a payphone could depress the rest briefly, whereupon he’d be reconnected (often to different operator), with no warning this time that the call was from a payphone. He could then call anywhere and bill it to any local number.

Early systems also signalled the entry of a coin by one or more pulses, each of which consisted of the insertion of a resistance in the line followed by a brief open circuit. At a number of colleges, enterprising students installed ‘magic buttons’ which could simulate this in a callbox in the student union so people could phone for free. (The bill in this case went to the student union, for which the magic button was not quite so amusing.)

Attacks on toll metering continue. Most countries moved their payphones from coins to chip cards in the 1990s to cut the costs of coin collection and vandalism, but as I remarked in section 18.5, the design was often poor at first and villains sold lots of bogus phone cards until it got fixed.

Other attacks involve what’s called clip-on: physically attaching a phone to someone else’s line to steal their service. In the 1970s through the 1990s, when international phone calls were very expensive, some foreign students would clip a phone on to a residential line in order to call home, and the unsuspecting home owner could get a huge bill. The Norwegian phone company had customer premises equipment authenticate itself to the exchange before a dial tone was given [907].

The UK phone company was not as enlightened as its Norwegian counterpart, and had a policy of denying that wiretaps were possible, so it could just collect the call charges from victim households. This occasionally caused collateral damage, as a family in Crumlington was to find out. The first sign they had of trouble was hearing a conversation on their line. The next was a visit from the police who said there’d been complaints of nuisance phone calls. The complainants were three ladies, all of whom had a number one digit different from a number to which this family had supposedly made a huge number of calls. When the family’s bill was examined, there were also calls to clusters of numbers that turned out to be payphones; these had started quite suddenly at the same time as the nuisance calls. When the family had complained later to the phone company about a fault, their connection was rerouted and this had solved the problem.
A report from the phone company’s maintenance engineer noted that the family’s line had been tampered with at the distribution cabinet, but this was against doctrine and the company later claimed the report was in error. It turned out that a drug dealer had lived close by, and it seemed a reasonable inference that he’d tapped their line in order to call his couriers at the payphones. By using an innocent family’s phone line instead of his own, he not only saved on the phone bill, but also had a better chance of evading police surveillance. But both the police and the local phone company refused to go into the house where the dealer had lived, claiming it was too dangerous – even though the dealer had by now got six years in jail. The Norwegian phone company declined an invitation to testify about clip-on for the defence. The upshot was that the subscriber was convicted of making harassing phone calls, in a case widely believed to have been a miscarriage of justice.

Stealing dial tone from cordless phones was another variant on the theme. In the 1990s, this became so widespread in Paris that France Telecom broke with phone company tradition and announced that it was happening, claiming that the victims were using illegally imported cordless phones which were easy to spoof [997]. That was a bit cheeky, as most equipment seems to simply send a handset serial number to the base station rather than using the DECT security mechanisms, which use cryptography patented by the French company Alcatel. These mechanisms were proprietary but turned out to have multiple weaknesses, as Erik Tews documented in 2012 after reverse engineering them [1697]. DECT authentication is based on a weak block cipher; confidentiality uses a weak stream cipher (a slightly more complicated version of A5/1 which I describe below in section 22.2.1) which can be broken with typically $2^{34}$ effort; there are weak random number generators; while protocol failures include a man-in-the-middle attack, and a replay attack where you make a silent call to collect keystream to decrypt a call you recorded earlier. It’s said that the German intelligence services used DECT to train recruits in signal collection and cryptanalysis. Since Tews’ work was published, the DECT standards body suggests using AES instead but it’s not clear how many vendors can be bothered. The takeaway is that a cordless phone gives you no security against a capable opponent nearby, and as the standard emerged during the Crypto Wars of the 1990s you should have expected nothing else. As for clip-on fraud, it has largely disappeared since services like Skype and WhatsApp made long-distance calls cheap.

Social engineering is also widespread. A crook calls you pretending to be from AT&T and asks whether you made a large number of calls to Peru on your calling card. When you deny this, they say that, in order to reverse out the charges, can you confirm that your card number is 123-456-7890-6543? No, you say (if you’re not really alert), it’s 123-456-7890-5678. Now 123-456-7890 is your phone number and 5678 your password, so that crook can now bill calls to you.

Premium-rate phone services grew rapidly during the 1990s, leading scammers to develop all sorts of tricks to get people to call them: pager messages, job ads, fake emergency messages about relatives, ‘low cost’ calling cards with 0900 access numbers, you name it. Indeed the business of tricking people into calling premium numbers enabled crooks to hone the techniques they now use in phish-
22.1. ATTACKS ON PHONE NETWORKS

ing attacks. The 809 area code for the Caribbean used to be a favourite cover for crooks targeting US subscribers; many people weren’t aware that ‘domestic’ numbers (numbers within the USA’s +1 international direct dialling code) include countries other than the relatively cheap USA and Canada. Even though many people have now learned that +1 809 is ‘foreign’ and more expensive, the introduction of still more Caribbean area codes, such as +1 345 for the Cayman Islands, has made it even harder to spot such scams.

Phone companies advised their customers ‘Do not return calls to unfamiliar telephone numbers’ – but how practical is that? Just as banks now train their customers to click on links in marketing emails and thus make them vulnerable to phishing attacks, so I’ve had junk marketing calls from my phone company – even though I’m on the do-not-call list. Governments typically set up weak regulators who avoid trying to regulate premium-rate operators, claiming it’s too hard; and from time to time it all blows up. In the late 2000s, all the major UK TV companies (including the state-owned BBC) ended up getting fined for getting viewers to phone in and vote, in all sorts of shows. Many of these are recorded, so the calls were futile [1197]. Phone scams by broadcast stations have been a recurring problem worldwide since radio broadcasting took off in the 1920s, and got worse when TV went mainstream in the 1950s [1854]. It’s also a recurring pattern that the biggest scams are often run by ‘respectable’ companies rather than by Russian gangsters.

22.1.2 Attacks on signaling

The term ‘phone phreaking’ refers to attacks on signaling as well as pure toll fraud. Until the 1980s, phone companies used signalling systems that worked in-band by sending tone pulses in the same circuit that carried the speech. The first attack I’ve heard of dates back to 1952, and by the mid-to-late 1960s many enthusiasts in both America and Britain had worked out ways of rerouting calls. One of the pioneers, Joe Engresia, had perfect pitch and discovered as a child that he could make free phone calls by whistling a tone he’d heard in the background of a long-distance call. His less gifted colleagues used home-made tone generators, of which the most common were called blue boxes. The trick was to call an 0800 number and then send a 2600Hz tone that would clear down the line at the far end – that is, disconnect the called party while leaving the caller with a trunk line connected to the exchange. The caller could now enter the number he really wanted and be connected without paying. Phone phreaking was one of the roots of the computer hacker culture that took root in the Bay Area and was formative in the development and evolution of personal computers [1108]. Steve Jobs and Steve Wozniak first built blue boxes before they diversified into computers [656].

Phone phreaking started out with a strong ideological element. In those days most phone companies were monopolies – large, faceless and unresponsive. In America, AT&T was such an abusive monopoly that the courts eventually broke it up; most phone companies in Europe were government departments. People whose domestic phone lines had been involved in a service theft found they were stuck with the charges. If the young man who had courted your daughter was (unknown to you) a phone phreak who hadn’t paid for the calls he made to her,
you would suddenly find the company trying to extort either the young man's name or a payment. Phone companies were also aligned with state security. Phone phreaks in many countries discovered signalling codes or switch features that would enable the police or the spooks to tap your phone from the comfort of their desks, without having to send out a lineman to clip on a wiretap. Back in the days of Vietnam and student protests, this was inflammatory stuff. Phone phreaks were counterculture heroes, while phone companies were hand-in-hand with the forces of darkness.

As there was no way to stop blue-box attacks so long as telephone signalling was carried in-band, the phone companies spent years and many billions of dollars moving to a signaling system called SS7 which is out-of-band, in effect on a private Internet to which normal subscribers had no easy access. Gradually, region by region, the world was closed off to blue-box attacks. This forced attackers to become insiders.

22.1.3 Attacks on switching and configuration

The second wave of attacks targeted the computers that did the switching. Typically these were Unix machines on a LAN in the exchange, which also had machines with administrative functions such as scheduling maintenance. By hacking one of these less well guarded machines, a phreak could go across the LAN and break into the switching equipment – or into other secondary systems such as subscriber databases. For a survey of PacBell’s experience of this, see [360]; for Bellcore’s, see [963].

Using these techniques, unlisted phone numbers could be found, calls could be forwarded without a subscriber’s knowledge, and all sorts of mischief became possible. A Californian phone phreak called Kevin Poulsen got root access to many of PacBel’s switches and other systems in 1985–88: this apparently involved burglary as much as hacking (he was eventually convicted of conspiring to possess fifteen or more counterfeit, unauthorized and stolen access devices.) He did petty things like obtaining unlisted phone numbers for celebrities and winning a Porsche from Los Angeles radio station KIIS-FM. Each week KIIS would give a Porsche to the 102nd caller, so Poulsen and his accomplices blocked out all calls to the radio station’s 25 phone lines save their own, made the 102nd call and collected the Porsche. He was also accused of unlawful wiretapping and espionage; these charges were dismissed. In fact, the FBI came down on him so heavily that there were allegations of an improper relationship between the agency and the phone companies, along the lines of ‘you scratch our backs with wiretaps when needed, and we’ll investigate your hacker problems’ [626].

The FBI’s sensitivity does highlight the fact that attacks on phone company computers are used by foreign intelligence agencies to conduct remote wiretaps. Some of the attacks mentioned in [360] were from overseas, and the possibility that such tricks might be used to crash the whole phone system in the context of an information warfare attack worried the NSA [660, 1005]. Countries that import their telephone exchanges rather than building their own are in an even worse position; a prudent nation will assume that its telephone switchgear has vulnerabilities known to the supplier’s government. (It was notable that during the invasion of Afghanistan in 2001, Kabul had two exchanges: an old
22.1. ATTACKS ON PHONE NETWORKS

electromechanical one and a new electronic one. The USAF bombed only the
first.)

Many real attacks involved insiders, who misconfigured systems to provide
free calls through special numbers. This didn’t matter much when the phone
company’s marginal cost of servicing an extra phone call was zero, but with
the proliferation of value-added services in the 1990s, and with deregulation
giving rise to cash payments between phone companies, it got serious [426]. In
a hack reminiscent of Poulsen, two staff at British Telecom were dismissed after
they each won ten tickets for Concorde from a phone-in offer at which only one
randomly selected call in a thousand was supposed to get through [1732].

As for outsiders, the other ‘arch-hacker’ apart from Poulsen was Kevin Mit-
nick, who got arrested and convicted following a series of break-ins which made
him too the target of an FBI manhunt. They initially thought he was a foreign
agent who was abusing the US phone system to wiretap sensitive US targets. As
I mentioned in Chapter 3, he testified after his release from prison that almost
all of his exploits had involved social engineering. He wrote a book on deception
that became a classic [1199]. In congressional testimony, he came up with the
quote at the head of this chapter: "Companies can spend millions of dollars
toward technological protections and that’s wasted if somebody can basically
call someone on the telephone and either convince them to do something on the
computer that lowers the computer’s defenses or reveals the information they
were seeking". Phone companies, like other firms, are vulnerable to careless
insiders as well as malicious insiders.

Fast forward to 2020, and one worrying development is the growth of switch-
ing exploits. A number of telcos now give SS7 access to corporate customers,
for example if they want to send bulk SMS messages to authenticate customers.
Access to the switch fabric lets them play the kind of games that Poulsen and
Mitnick got up to in the 1980s. For example, if I want to hack your gmail
account, I send a message to your mobile service provider saying that you’ve
roamed into my network. I then start an account recovery at Google, and get
the SMS I need to reset your password. As I noted in sections 3.4.1 and 12.7.4,
this is now in active use for bank fraud; the first instance of its being used to
steal money from bank customers was in Germany in 2016, when they were
moved without their knowledge to another network; there was a similar fraud in
London in 2019 [450]. Most major telcos in developed countries now use some
SS7 firewalls, and allow or deny remote access depending on their roaming
agreements. If there is such an agreement, a firm given SS7 access by the remote
telco can either steal a phone to get its SMS messages, or push to do premium
fraud. Forensics can be hard if there’s a complaint from a single user; the best
you can do may be to look for roaming charges. If there are a thousand cases
the bank might be motivated to go to the operator. But banks and their bulk
SMS contractors are paying operators for SS7 access, opening up the formerly
closed system. In short, we used to think that attacks involving SS7 were the
preserve of nation states, but that is no longer the case.
22.1. ATTACKS ON PHONE NETWORKS

22.1.4 Insecure end systems

The next major vulnerabilities of modern phone networks were insecure terminal equipment and feature interaction.

There have been many exploits of voicemail, whether implemented as an answering machine on customer premises or, as common now, a cloud service. Exploits start with tricking someone into calling a premium-rate number, and escalate to journalists and others hacking voicemail via the default PINs that many people don’t bother to change. The most notorious case was the murder, on the 21st of March 2002, of the English schoolgirl Millie Dowler. In 2011 it transpired that an investigator working for the News of the World, then the UK flagship of the Murdoch empire, had hacked Millie’s voicemail and deleted some of her messages, giving Millie’s family false hope that she might still be alive. This led to the closure of the newspaper, several criminal convictions – including the imprisonment in 2014 of David Cameron’s publicist Andy Coulson, a former News of the World editor – and a public inquiry into press standards.

But the really big frauds that exploit insecure end systems tend to target companies and government departments, as they have the ability to pay big phone bills. Attacks on corporate private branch exchange systems (PBXes) had become big business by the mid-1990’s and cost business billions of dollars a year [431]. PBXes are usually supplied with facilities for refiling calls, also known as direct inward system access (DISA). The company’s sales force could call in to an 0800 number, enter a PIN or password, and then call out again taking advantage of the low rates a large company can get for long-distance calls. As you’d expect, these PINs become known and get traded by villains [1226]. The result is known as dial-through fraud.

In many cases, the PINs are set to a default by the manufacturer, and never changed by the customer. Many PBX designs also have fixed engineering passwords that allow remote maintenance access, and prudent people reckon that any PBX will have at least one back door to give easy access to law enforcement and intelligence agencies (it’s said, as a condition of export licensing). Such features get discovered and abused. In one case, the PBX at Scotland Yard was compromised and used by criminals to refile calls, costing the Yard a million pounds, for which they sued their telephone installer. The crooks were never caught [1696]. One of the criminals’ motivations is to get access to communications that will not be tapped. Businesses who’re the victims of such crimes find the police reluctant to investigate, and the phone companies aren’t helpful – they don’t like having their bills disputed [1475].

In a notorious case, Chinese gangsters involved in labor market racketeering – smuggling illegal immigrants from Fujian, China, into Britain – hacked the PBX of an English district council and used it to refile over a million pounds’ worth of calls to China. The gang was tackled by the police after a number of its labourers died; they were picking shellfish in Morecambe Bay when the tide came in and drowned them. The council had by now discovered the discrepancy in its phone bills and sued the phone company for its money back. The phone company argued that it wasn’t to blame, even although it had supplied the insecure PBX. Here, too, the gangsters were interested not just in saving money but in evading surveillance. (Indeed, they routed their calls to China via a
compromised PBX in Albania, so the cross-border segment of the call, which is most likely to be monitored by the agencies, was between numbers their collection systems wouldn’t touch; the same trick seems to have been used in the Scotland Yard case, where the crooks made their calls via the USA.)

Such cases apart, dial-through fraud is mostly driven by premium rate services and the crooks are in cahoots with premium line operators. Most companies don’t understand the need to guard their ‘dial tone’ and don’t know how to even if they wanted to. PBXes are typically run by company telecomms managers who know little about security, while the security manager often knows little about phones. This is changing as VOIP technologies take over and the company phone network merges with the data network. Estimates of the losses from PBX fraud sustained by business worldwide fell from $4.96bn in 2011 to $3.88bn in 2017, with about half the latter figure now actually VOIP rather than classical PBX [84].

Exploits of insecure end-systems affect domestic subscribers too. Premium-rate mobile malware arrived in 2006, when the Red Browser worm cashed out by sending $5 SMSs to Russia [861]; this scaled up after Android came along, and we’ll discuss mobile malware in section 22.3.1.4 below. And now that phones are used more and more for tasks such as voting, securing entry into apartment buildings, checking that offenders are observing their parole terms, and authenticating financial transactions, more motives are created for ever more creative kinds of mischief, and especially for hacks that defeat caller line ID. Since the early 2000s, there have been warnings that caller-line ID hacks, SMS spoofing and attacks on the SS7 signaling could be used for fraud. This is now reality, and we’ll discuss it in more detail below.

22.1.5 Feature interaction

Phone manipulation often involves feature interaction.

- Inmates at the Clallam Bay Correctional Center in Washington state, who were only allowed to make collect calls, found an interesting exploit of a system which the phone company (‘Fone America’) introduced to handle collect calls automatically. The system would call the dialled number and a synthesised voice would say: “If you will accept a collect call from...(name of caller)...please press the number 3 on your telephone twice.” Prisoners were supposed to state their name for the machine to record and insert. The system had, as an additional feature, the ability to have the greeting delivered in Spanish. Inmates did so, and when asked to identify themselves, said “If you want to hear this message in English, press 33.” This worked often enough that they could get through to corporate PBXes and talk the operator into giving them an outside line. The University of Washington was hit several times by this scam [632].

- Many directory-enquiry services will connect you to the number they’ve just given you, as a service to motorists who can’t dial while driving. But this can be used to defeat mechanisms that rely on endpoint identification. Naughty children use it to call sex lines despite call barring, while naughty
22.1. ATTACKS ON PHONE NETWORKS

grown-ups use it to prevent their spouses seeing lovers’ numbers on the family phone bill [1323].

- Call forwarding is a source of many scams. In the old days, it was used for pranks, such as kids social-engineering a phone company operator to forward their teacher’s calls to a sex line. Nowadays, it can be both professional and nasty. For example, a fraudster may tell a victim to confirm their phone number with the bank by dialing a sequence of digits – which forwards incoming calls to a number controlled by the attacker. So the bank’s callback mechanisms are defeated.

- Conference calls can be exploited in all sorts of ways. For example, football hooligans in some countries are placed under a curfew that requires them to be at home during a match, and to prove this by calling the probation service, which verifies their caller ID. So you get your partner to set up a conference call with the probation service and your mobile. If the probation officer asks about the crowd noise, you tell him it’s the TV and you can’t turn it down or your mates will kill you. (And if he wants to call you back, you get your partner to forward the call.)

22.1.6 VOIP

In voice over IP (VOIP), voice traffic is digitised, compressed and routed over the Internet. This had experimental beginnings in the 1970s; products started appearing in the 1990s, and it became big business from the mid-2000s. Nowadays, most normal phone calls are digitized and sent over IP networks belonging to the phone companies, so in a technical sense almost all phone calls are now ‘VOIP’. But though my home phone pretends to be a POTS device, my lab phone is now a born-VOIP device that offers conference calling and all sorts of other complicated features that I don’t understand.

The most popular VOIP protocol, the Session Initiation Protocol (SIP), has had its share of vulnerabilities [1870] but is mostly attacked through poor configurations, for which many actors are constantly scanning; a PBX can get over a million messages a day trying to register as an extension, and then attempting to call high-cost numbers in less developed countries [1150]. As I noted in section 22.1.4 above, the VOIP segment of frauds against corporate PBX systems was about $2bn p.a. by 2017 [84]. The broader interaction with security is complicated. Corporate security policies can result in firewalls refusing to pass VOIP traffic. The current political tussle is over robocalls, which can hide caller ID more easily if they go over VOIP. The FCC voted in 2020 to insist that telcos implement by the end of June 2021 a suite of protocols, STIR/SHAKEN, which authenticate callers over SIP [301]. Another regulatory issue is that governments want emergency calls made through VOIP services to work reliably, and provide information about the location of the caller. But an IP packet stream can be coming from anywhere, and no-one owns enough of the Internet to guarantee quality of service. And although a VOIP handset looks like a phone and works like a phone, if the power goes off, so does your service. Then you’re forced to fall back on the mobile network.
22.1. Frauds by phone companies

Phone fraud is not just a story of crooked customers committing toll fraud against telcos, and defrauding other customers by exploiting mechanisms that the telcos have no real incentive to harden. There are many scams by unscrupulous telcos. The classic scam is cramming, where a rogue phone company bills lots of small sums to unwitting users. Billing was designed in the days when phone companies were monopolies, usually state-owned, and assumes that phone companies trust each other: if company A creates a call data record (CDR) saying that a customer of telco B called their subscriber, they just pass it on to telco B, which pays up. (It has no incentive to quibble, as it gets a cut.)

I was myself the victim of an attempt at cramming. On holiday in Barcelona, my wife’s bag was snatched, so we called up and cancelled the phone that she’d had in it. Several months later, we got a demand from that MNO to pay a few tens of dollars roaming charges recently incurred by that SIM card in Spain. In all probability, the Spanish phone company was simply cramming a few charges to a number that they’d seen previously, in the knowledge that they’d usually get away with it. My MNO insisted that even though we’d cancelled the number, we were still liable for calls billed to it months afterwards and had to pay up. I got out of the charges only because I’d met the company’s CEO at an academic seminar and was able to get his private office to fix the problem. Customers without such access usually get the short end of the stick. Indeed, UK phone companies’ response to complaints has been to offer customers ‘insurance’ against fraudulent charges. That they can get away with this is a clear regulatory failure. There are many variants: if you call an 800 number in the USA, the company may say “Can we call you right back?” and if you agree then you’re deemed to have accepted the charges, which can be at a high premium rate. The same can happen if you respond to voice prompts as the call progresses.

Another problem is slamming – the unauthorized change of a subscriber’s service provider without their consent. It would be a mistake to assume that cramming and slamming are just done by small fly-by-night operators. AT&T was one of the worst offenders, having been fined not only for slamming, but for forging signatures of subscribers to make it look as if they had agreed to switch to their service. They got caught when they forged a signature of the deceased spouse of a subscriber in Texas.

Yet another is the short termination of international calls. The abuse of premium-rate numbers led regulators in many countries to force phone companies to offer premium-rate number blocking to subscribers. The telcos get round this by disguising premium rate numbers as international ones. I mentioned scams with Caribbean numbers in section 22.1.1 above. Now many other phone companies from small countries have got into the act, and offer sex line operators a range of numbers on which they share the revenue. For example, calls for the small Pacific country of Tuvalu went via Telstra in Perth, Australia, where they were forwarded by satellite. However, the sex line numbers were marked as invalid in Telstra’s system, so they were automatically sent via the second-choice operator – a company in New Zealand. This is a standard operating procedure: in the Moldova scam I mentioned earlier, the calls didn’t
go to Moldova but to Canada. Legally, it is hard to challenge as there is an international agreement (the Nairobi Convention) that stops phone companies selectively blocking international destinations. So if you want to stop your staff phoning the sex line in Tuvalu, you have to block all international calls.

The interaction between scams and regulation is increasingly complex; all sorts of new services are being made possible by the smartphone revolution and the regulators are usually somewhat captured by the phone companies – who want a cut of high-value service delivery, ranging from parking meters in London to ferry tickets in Finland. As malware became widespread on mobile phones, the botnet herders who control subverted phones can pay for all sorts of goods and services by SMS. I predicted in the second edition of this book that such exploits would be industrialised; they soon got to the point that neither Google nor Apple allows normal apps to send or receive text messages. We will discuss malware later, but first we might pause to think of the industry’s economics. Why are telcos so antagonistic towards their customers?

### 22.1.8 Security Economics of Telecomms

Phone and cable companies have extremely high fixed costs and very low marginal costs. Building a nationwide network costs billions and yet the cost of handling an additional phone call or movie download is essentially zero. As I discussed in the chapter on Economics, this has a couple of implications.

First, there’s a tendency towards dominant-firm markets. For many years telephone service was considered in most countries to be a ‘natural monopoly’ and operated by the government; the main exception was the USA where the old AT&T system was heavily regulated. After the breakup of AT&T following an antitrust case, and Margaret Thatcher’s privatisation of BT, the world moved to a different model, of regulated competition. The details vary from one country to another but, in general, some sectors (such as mobile phones) had a fixed number of allowed competitors; others (such as long-distance provision) were free for companies to compete in; and others (such as local loop provision) remained monopolies but were regulated.

Second, the competitive sectors (such as long-distance calling) saw prices drop quickly to near zero. Some sectors were made competitive by apps: Skype and WhatsApp made international calls essentially free.

In many telecomms markets, the outcome is confusion pricing – products are continually churned, with new offerings giving generous introductory discounts to compete with the low-cost providers, but with rates sneakily raised afterwards. There is constant bundling, of broadband access with mobile service and TV offerings. If you can be bothered to continually check prices, you can get really good deals, but often at the cost of indifferent service. If you don’t have the time to keep scrutinising your broadband and mobile phone bills, you can get some unpleasant surprises.

---

1Finally, twelve years after I pointed this out in the second edition of this book, my university has now blocked all outgoing international calls on my mobile.
22.2 Going mobile

Since their beginnings as an expensive luxury in 1981, mobile phones have become one of the big technological success stories. By 2020, we now have over five billion subscribers; it’s said that over a billion phones will be sold this year. In developed countries, most people have at least one mobile, and many new electronic services are being built on top of them. Growth has been rapid in developing countries too, where the wireline network is often dilapidated and people used to wait years for phone service to be installed. In many places it’s the arrival of mobile phone service that’s connected villages to the world. This has brought many benefits, and new crimes too. They developed steadily as the technology was evolved and deployed.

The first generation of mobile phones (1G) used analog signals and the handset simply sent its serial numbers in clear over the air link\(^2\). So villains built devices to capture these numbers from calls in the neighborhood, or reprogrammed phones to steal ID from other phones nearby. One of the main customers was the call-sell operation that would steal phone service and resell it cheaply, often to immigrants or students who wanted to call home. The call-sell operators would hang out at known pitches with cloned mobiles, and their customers would queue up to phone home for a few dollars. The call-sell market was complemented by the criminal market for anonymous communications: people hacked mobile phones to use a different identity for each call. Known as tumblers, these were particularly hard for the police to track [864]. The demand for serial numbers grew rapidly and satisfying it was increasingly difficult, even by snooping at places like airports where lots of mobiles get turned on. So prices rose, and as well as passive listening, active methods started to get used.

Mobile phones are cellular: the operator divides the service area up into cells, each covered by a base station. The mobile uses the base station with the strongest signal, and there are protocols for handing off calls from one cell to another as the customer roams. Early active attacks consisted of a fake base station, typically at a place with a lot of passing traffic such as a freeway bridge. As phones passed by, they heard a stronger signal and attempted to register by sending their serial numbers and passwords.

Various mechanisms were tried to cut the volume of fraud. Most operators ran intrusion-detection systems to watch out for suspicious patterns of activity, such as too-rapid movement or a rapid increase in call volume or duration. Vodafone also used RF fingerprinting, a military technology in which signal characteristics arising from manufacturing variability in the handset’s radio transmitter are used to identify individual devices and tie them to the claimed serial numbers [707].

22.2.1 GSM

The second generation of mobile phones (2G) adopted digital technology. The Global System for Mobile Communications (GSM) was founded when 15 coun-

\(^2\)In the US system, there were two of them: one for the equipment, and one for the subscriber.
panies signed up to the GSM Association in 1987 and secured political support from the EU; service was launched in 1992. The designers of GSM set out to secure the system against cloning and other attacks: their goal was that GSM should be at least as secure as the wireline system. What they did, how they succeeded and where they failed, make an interesting case history.

The industry initially tried to keep secret the cryptographic and other protection mechanisms which form the core of the GSM protocols. This didn’t work: some eventually leaked and the rest were discovered by reverse engineering. I’ll describe them briefly here. Mobile networks consist of a radio access network (RAN) and a core network (CN), and each mobile network has two databases, a home location register (HLR) that contains the location of its own mobiles, and a visitor location register (VLR) for the location of mobiles which have roamed in from other networks. These databases enable incoming calls to be forwarded to the correct cell.

The handsets are commodity items, personalised using a subscriber identity module (SIM) – a smartcard you get when you sign up for a network service, and which you load into your handset. The SIM can be thought of as containing three numbers:

1. there may be a personal identification number that you use to unlock the card;
2. there’s an international mobile subscriber identification (IMSI), a unique number that maps on to your mobile phone number;
3. finally there is a subscriber authentication key $K_i$, a 128-bit number that serves to authenticate that IMSI and is known to your home network.

There is also a handset serial number, the international mobile equipment identification (IMEI). The protocol used to authenticate the handset to the network runs as follows (see Figure 2021). On power-up, the SIM may request the customer’s PIN; if this isn’t configured, or once it’s entered correctly, the SIM emits the IMSI, which the handset sends to the nearest base station along with the IMEI. The IMSI is relayed to the subscriber’s HLR, which generates five triplets. Each triplet consists of:

- RAND, a random challenge;
- SRES, a response; and
- $K_c$, a ciphering key.

The algorithm is that RAND is encrypted under the SIM’s authentication key $K_i$, giving SRES concatenated with $K_c$:

$$\{RAND\}_K = (SRES|K_c)$$

The encryption method is up to the issuer; an early standard called Comp128 turned out to be insecure [1781, 1782], so issuers nowadays use hash functions or constructions using AES.

Security Engineering 664 Ross Anderson
22.2. GOING MOBILE

Anyway, the triplets are sent to the base transciever station (BTS), which now presents the first RAND to the mobile. It passes this to the SIM, which computes SRES. The mobile returns this to the base station and if it’s correct the mobile and the base station can now communicate using the ciphering key $K_c$. So the whole authentication protocol runs as in Figure 22.2.

```
SIM → HLR IMSI
HLR → BSC (RAND, SRES, $K_c$), ...
BSC → SIM RAND
SIM → BSC SRES
BSC → mobile {traffic}$K_c$
```

Figure 22.2 – GSM authentication protocol

There are several vulnerabilities in this protocol. First, the base station isn’t authenticated, so it’s easy for a wiretapper to use a false base station to intercept calls. Such devices, known as IMSI catchers in Europe and Stingers in the USA, are now standard law-enforcement equipment. Second, in most countries the communications between base stations and the VLR pass unencrypted on microwave links\(^3\). This allows bulk interception by intelligence agencies, and in many cases access to the triples needed to spoof or decrypt traffic.

The introduction of GSM caused significant shifts in patterns of crime. The authentication mechanisms made phone cloning difficult, so the villains switched to buying phones using stolen credit cards, using stolen identities or bribing insiders [1839]. Robbery was the next issue, with a spate of media stories about kids being mugged for their phones. Mobile phone crime did indeed increase 190% between 1995 and 2002, but to keep this in context, the number of subscribers went up 600% in the same period [792]. Some of the theft is bullying – kids taking smaller kids’ phones; some is insurance fraud by subscribers who’ve dropped their phones in the toilet and report them as stolen as their insurance doesn’t cover accidental damage; but there is a hard core of theft where muggers take phones and sell them to fences. Many of the fences either work at mobile phone shops that have authorised access to tools for reprogramming the IMEI, the serial number in the handset, or else have links to organised criminals who ship the handsets abroad.\(^4\).

Prepaid mobile phones appeared from about 1997, enabling the industry to expand rapidly to people without credit ratings, including both poor people

---

\(3\)The equipment can encrypt traffic, but the average phone company has no incentive to switch the cryptography on.

\(4\)In recent smartphone designs, the IMEI is supposed to be unalterable; some Android phones keep it in TrustZone.
in rich countries and everyone in poor countries. By 2008, prepaids made up 90% of the market in Mexico but 15% in the USA. During the 2010s, billions of people got access not just to calls and texts but to online information and payment services.

Prepaid phones also made anonymous communication practical. The issues include not just evading police wiretapping but fraud, stalking, extortion, bullying and other kinds of harassment. However, prepaid phones only protect you from the police if they don’t try very hard. Most criminals don’t have any clue of the level of operational discipline needed to stop traffic analysis. As I already remarked, one alleged 9/11 mastermind was caught when he used a prepaid SIM from the same batch as one that had been used by another Al-Qaeda member; and after the failed 21/7 London bombings, one would-be bomber fled to Rome, where he was promptly caught. He had changed the SIM in his mobile phone en route; but call records show not just the IMSI from the SIM, but also the IMEI from the handset. If you’ve got all the world’s police after you, just changing the SIM isn’t anything like enough. Operational security requires a detailed technical understanding of how networks operate, and as the 2019 impeachment hearings into President Trump showed, even US ambassadors are vulnerable if capable foreign intelligence agencies are after their traffic.

In addition to authentication, 2G was supposed to provide two further kinds of protection – location security and call content confidentiality.

The location security mechanism is that when a mobile is registered to a network, it is issued with a temporary mobile subscriber identification (TMSI), which acts as its address in that network. This is a lightweight mechanism; it is defeated trivially by IMSI catchers, which pretend to be a base station in a different network.

2G GSM also provides some call content confidentiality by encrypting the traffic between the handset and the base station once authentication and registration are completed. The speech is digitized, compressed and chopped into packets; each packet is encrypted by xor-ing it with a pseudorandom sequence generated from the ciphering key \(K_c\) and the packet number. The algorithm commonly used in Europe is A5/1. This is a stream cipher that, like Comp128, was originally secret; like Comp128, it was leaked and attacks were quickly found on it [225]. By the mid-2000s, law enforcement suppliers were selling devices that would break the key in under a second, enabling a surveillance team to hoover up all the GSM traffic and decrypt it, so they could then pick out conversations of interest. Phones also supported an even weaker algorithm called A5/2, which was licensed for export to non-EU countries\(^5\) and which can be broken almost instantly. As I mentioned above in section 22.1.1, the DECT standard for cordless phones is somewhat similar, and also weak. The embassies of major powers round the world have roof structures that indicate antennas for capturing local telephone traffic, and the Snowden papers confirm that the NSA collects local phone traffic at US diplomatic missions.

In addition to passive bulk collection, targeted active collection can exploit protocol tricks.

GSM vendors introduced a third cipher, A5/3, which is based on a strong

---

\(^5\)There was a row when it emerged that Australia was using A5/2.
22.2. GOING MOBILE

block cipher known as Kasumi and became standard in third-generation mobile phones. But there’s the bidding-down attack which exploits the fact that the initial algorithm negotiation is in plaintext. The IMSI catcher simply tells the handset to use a weaker cipher. Elad Barkan, Eli Biham and Nathan Keller realised that this can be done retrospectively [157]. If you’re following a suspect who uses his mobile, you record the call, including the initial protocol exchange of challenge and response. Once he’s finished, you switch on your IMSI-catcher and cause him to register with your bogus base station. The IMSI-catcher tells his phone to use A5/2 rather than A5/1, and a key is duly set up – with the IMSI-catcher sending the challenge that was used before. So the mobile phone generates the same key $K_c$ as before. As this is now being used in a weak cipher, it can be cracked quickly, giving access to the conversation already recorded. A5/2 has now been retired; handsets that cannot use A5/1 or A5/3 communicate in plaintext. However A5/1 is easy to break with modern equipment.

Phone companies, equipment vendors and ISPs are now compelled to provide for local law enforcement access, but other countries often want access too and the wiretap facilities are often so poorly engineered that they can be abused [1552]. In 2004-5, persons unknown (but presumed to be from the NSA or CIA) tapped the mobile phones of the Greek Prime Minister and about a hundred of that country’s political, law enforcement and military elite during the Athens Olympics, by subverting the wiretapping facilities built into Vodafone’s Greek network. Both Vodafone, and their equipment supplier Ericsson, were heavily fined [1412]. Colleagues and I warned about this problem years ago [4] and the Snowden disclosures suggest that it has got steadily worse. I’ll discuss it at greater length in Part III.

Anyway, the net effect is while the 2G GSM security mechanisms were designed to provide slightly better protection than the wireline network in countries allowed to use A5/1, and somewhat worse protection elsewhere, they now provide slightly worse protection everywhere because of the range of exploits that can be industrialised by third parties.

22.2.2 3G

The third generation of digital mobile phones was initially known as the Universal Mobile Telecommunications System (UMTS) and now as the Third Generation Partnership Project (3gpp, or just 3G). The acronym 3gpp is still used for the standards body working on 4G, 5G and beyond. 3G entered service in 2003–2004 and is due to be retired in 2022, after which mobile networks that cannot use 4G or 5G for some reason are supposed to fall back to 2G. 3G uses spread-spectrum technology on the radio access network, and its main advantage over 2G is higher data rates: instead of the 9.6kb/s of standard 2G and the tens of kilobits per second of the 2.5G variant (GPRS), 3G data rates are in the hundreds of thousands of bits per second. 3G’s vision was to enable all sorts of mobile services, from mobile TV to laptops that just go online anywhere. It laid the foundation for the smartphone revolution.

The overall security strategy is described in [1786], and the security architecture is at [1773]. The crypto algorithms A5/1 and A5/2 are replaced by A3, based on a block cipher called Kasumi [932], which in turn is based on a design...
by Mitsuru Matsui called Misty, which has now withstood public scrutiny for two decades [1127]. All keys are now 128 bits. Cryptography is used to protect the integrity and confidentiality of both message content and signalling data, rather than just content confidentiality, and the protection runs from the handset to the core network, rather than simply to the local base station. So picking up the keys, or the plaintext, from the base station or microwave backhaul is no longer an attack. The authentication is now two-way rather than one-way. The theory was that this would end the vulnerability to rogue base stations, and IMSI catchers wouldn’t work any more. In practice, they work fine as they just tell the target handset to fall back to 2G operation. 3G also has also a proper interface for local interception [1774].

In the basic 3G authentication and key agreement (AKA) protocol, the authentication runs from the handset to the visitor location register. The home location register is now known as the home environment (HE) and the SIM as the UMTS SIM (USIM). The home environment chooses a random challenge RAND as before and enciphers it with the USIM authentication key $K$ to generate a response RES, a confidentiality key $CK$, and integrity key $IK$, and an anonymity key $AK$.

$\{RAND\}_K = (RES|CK|IK|AK)$

There is also a sequence number SEQ known to the HE and the USIM. A MAC is computed on RAND and SEQ, and then the sequence number is masked by exclusive-or’ing it with the anonymity key. The challenge, the expected response, the confidentiality key, the integrity key, and the masked sequence number made up into an authentication vector AV which is sent from the HE to the VLR. The VLR then sends the USIM the challenge, the masked sequence number and the MAC; the USIM computes the response and the keys, unmasks the sequence number, verifies the MAC, and if it’s correct returns the response to the VLR.

- USIM $\rightarrow$ HE \hspace{1cm} IMSI (this can optionally be encrypted)
- HE $\rightarrow$ VLR \hspace{.2cm} RAND, XRES, $CK$, $IK$, $SEQ \oplus AK$, $MAC$
- VLR $\rightarrow$ USIM \hspace{.2cm} RAND, $SEQ \oplus AK$, $MAC$
- USIM $\rightarrow$ VLR \hspace{.2cm} RES

Fig 20.4 – 3gpp authentication protocol

The 3G standards set out many other features, including identity and location privacy mechanisms, backwards compatibility with 2G, mechanisms for encrypting authentication vectors in transit from HEs to VLRs, and negotiation of various optional cryptographic mechanisms.

As with 2G, its design goal was that security should be comparable with that of the wired network [844] and the net effect was a modest improvement: bulk eavesdropping on the air link is prevented by higher-quality mechanisms, although targeted attacks by IMSI catchers still work by exploiting fallback. In a number of countries, third-generation mobiles were hard for the police to tap in the first few years, as they had to integrate their systems with those of the network operators to operate at any scale greater than tactically.
22.2. GOING MOBILE

22.2.3 4G

Fourth-generation mobile networks were first rolled out in 2009, and accounted for most mobile subscriptions (4.2bn of the 8bn) by 2019 [896]. They use IP throughout, unlike 2G and 3G which had circuit-switched core networks. The radio access network changed from spread spectrum to frequency-domain equalization (FDE) schemes, making very high bit rates possible despite multi-path radio propagation (echoes). The higher data rates made apps such as Google Maps and Snapchat possible. There is actually a family of standards that has evolved during the 2010s, supporting bandwidths in the megabits up to tens of megabits per second. The 4G security standards rowed back from 3G by limiting encryption to the link between the handset and the base station. The authentication and key agreement (AKA) protocol is very similar to 3G, although the nomenclature has changed: The handset is now the UE or user equipment while the HE/HLR is now the home subscriber server (HSS), and the base station functionality is split into an Evolved NodeB (eNodeB) base station and a smaller number of Mobility Management Entities (MMEs), which handle the AKA exchange, make admission decisions, supply session keys to the base stations and handle law enforcement access. The idea was that the MMEs can be housed in protected spaces or at least made tamper-resistant (people talked about TPMs but no operator seems to have implemented them).

The three main weaknesses in 4G are that local traffic at a base station (or MME) can still be monitored by anyone who can take it over; that the user equipment’s identity is sent to the network in the clear, or masked using a Globally Unique Temporary Identity (GUTI) which is fairly weak, like its predecessor the TMSI [840]; and that the home network delegates authentication to the serving network [335]. SS7 is replaced by a control protocol suite called Diameter, where messages can be optionally encrypted, but as the operators trust each other it’s vulnerable to many of the same types of attack [397]. It started off with fewer abusable functions, but they got put back in following business pressure.

On top of 4G is coming Rich Communications Services (RCS), which became widely available during 2019 thanks to support from Google in its Messages app. It is intended to replace SMS with richer chat features including geolocation exchange, social presence information and voice-over-IP. Also known as SMS+, +Message or joyn, it provides many of the same services as WhatsApp, but without the end-to-end encryption, as it’s a telco hosted product. Many of the initial implementations are insecure as the telcos haven’t configured them correctly [1541].

For decades, phone security has been kept weak at the behest of the security and intelligence community. Yet this strategy blew back when it turned out that Russian agents in the USA compromised the communications of FBI counterintelligence agents who used push-to-talk cellphones [531]. We haven’t been told whether they were 3G or 4G, or what the specific exploits were, but it was so bad that in December 2016 the Obama administration kicked out three dozen Russian diplomats. They had also been obsessed with getting premises with line of sight to the CIA HQ at Langley, Virginia.
22.2. GOING MOBILE

22.2.4 5G and beyond

Fifth-generation networks entered service in 2019, promising a further significant improvement over 4G in terms of bandwidth and latency. The main driver at present is bandwidth; mobile traffic grew by 68% between Q3 2018 and Q3 2019, mostly from video, and growth at over 25% is anticipated up till 2025, by which time almost half the traffic worldwide will be 5G [896]. Again, there’s an evolving family of standards, with complexity increasing still further. Initial deployments use non-standalone mode (NSA) which reuse the 4G control plane (and even the 4G towers) but boost the data rate. The real excitement is about standalone mode (SA) which will follow. 5G makes it cheaper and easier for mobile network operators to build new capacity, not just at existing frequencies, but at millimeter-wave frequencies over 20GHz, which will mean much larger numbers of small base stations on lamp posts, bus stops and so on (this will also limit the time available to do authentication handshakes). Network energy efficiency and area traffic capacity could be up two orders of magnitude, while latency, connection density, mobility and data rates could go up one order. Availability is a high priority; after the 2017 Brussels bombings, the police couldn’t get network service because of congestion, and had to find wifi hotspots to talk to each other.

The terminology changes yet again. Each tiny base station is now a distributed unit (DU) and is controlled by a centralised unit (CU), which is also in the field but counted as part of the core network. The encryption goes from your device to the CU, and from there it’s protected using IPSec to the access management function (AMF), which replaces the MME boxes. The authentication and key agreement protocols are much the same (XRES is renamed HXRES). One material improvement is that your device identity is sent to your home network encrypted under its public key, so location privacy will be harder to break; and we’re told that IMSI catchers won’t work any more6. Passive and active attacks by fake base stations seem still possible, including man-in-the-middle attacks that downgrade a device to a previous generation of technology, and could be used to deplete the batteries of energy-critical devices [1557].

However the whole core network moves to the cloud, including all the law-enforcement access mechanisms. Instead of defending familiar technologies, mobile network operators will depend on new ones that they don’t understand and which most will just buy from the cheapest vendor. One mistake in configuration, and things could be world readable; and unless something like SGX can be made to work, the cloud providers’ governments may well be able to get access by serving warrants on them rather than on the operators. The use of SDN in the core cloud network opens up still more questions, of which the most troublesome long-term may be whether 5G becomes an end-run round net neutrality, enabling network operators to customise offerings to each application by performance (and price). Meanwhile the specifications are complex and the implementations are still flaky. As the standards evolve, one fight is between the big data carriers who want to manipulate traffic to break net neutrality and claw their way up the value chain, versus the big mobile network operators who want end-to-end trust. In theory traffic edits will be signed by the firm that

---

6well, we heard that before with 3G: the wiretappers just forced fallback to 2G. We hear that the intelligence agencies are lobbying to break this, in alliance with the big data carriers.
does the editing, but nobody seems to know how that will work. Another is that the US government is trying to prevent Huawei getting a critical mass of installations outside China; the 2019 annual report of the UK National Cyber Security Centre (part of GCHQ) noted that significant supply-chain risks have developed over 2010–19, for which market drivers were insufficient to ensure an adequate response [1262]. In 2020, with anti-Chinese sentiment rising during the coronavirus pandemic, the UK Prime Minister Boris Johnson decided to ban Huawei from the 5G network by 2023, then reversed the decision when the telcos told him they couldn’t even replace the existing Huawei kit by then. A longer-term resolution may depend on a third tussle, between the ‘bellheads’ and the ‘netheads’: between firms like Nokia and Huawei who take a phone-industry approach and culture, and insurgents such as Rakuten whose culture is from the computer industry and will happily virtualise everything in sight once it’s in the cloud [559].

What about 6G and 7G? Telecomms researchers talk about the former seeing evolution in the radio access network to support a diversity of apps with different requirements for peak bandwidth, latency, service quality and power consumption [1321]; and the latter having thousands of micro-satellites to deploy 200Mbps broadband over all the earth’s surface. The arrival of streaming games, augmented reality and (perhaps) autonomous vehicles will create demand for ultra-low-latency cloud services, so rather than having our data shipped off to a few dozen data centres run by Google, Facebook, Microsoft and Amazon, we may see edge clouds with clusters of servers in each town, perhaps even in the buildings that used to house the old telephone exchanges. Then, just as the dotcom boom in the late 1990s forced us to partition web services into the active processes at the core and the rest that could be served more or less statically and thus cached locally in CDNs, we’ll have to host some of the active stuff locally too.

22.2.5 General MNO failings

Regardless of the generation of radio link technology in use, there are some common failings of MNOs whose root causes lie in the economics and regulation of the industry. One is the rapidly growing attacks on authentication functions supported by mobile phones. In addition to the SS7 security issues we discussed in section 22.1.3, which apply also to wireline telcos, the mobile world has brought us SIM swapping, channel jacking and the theft of cookies from authenticator apps. Many of these have security economics at their root: there is some misalignment of incentives between the various principals in the system.

In section 3.4.1 we introduced SIM swap attacks, where the attacker persuades the victim’s telco to issue a new SIM card on the victim’s account. This can open the door to all sorts of mayhem; individuals can have their lives trashed by attackers who take over their online accounts. Celebrities are targets: in August 2019, Twitter CEO Jack Dorsey had his account taken over for an hour and used to send racist and antisemitic tweets, causing commentators to wonder whether someone who took over President Trump’s twitter account might start World War 3 [1215]. As I mentioned in 12.7.4, SIM swap attacks are
mostly used in 2020 against the customers of banks and bitcoin exchanges, and often involve phone company insiders. Yet the response of phone companies has been at best patchy. Although the first of them (MTN in South Africa) adopted countermeasures in 2003, the only major US MNO making SIM swapping harder is Verizon [646]. But not all countermeasures help all users: if they are optional, then the company can more easily disclaim losses by the customers who don’t opt to use them. The first MNO to take action was MTN in South Africa in 2003, which enabled users to designate a second SIM to authorise SIM swap fraud case in 2007, which I described in section 12.7.4. We discussed the often adversarial attitude of phone companies toward their customers in section 22.1.8; MNOs are no different in this respect from legacy wireline phone companies. Indeed, they may be worse because most of their customers in most countries are prepayment customers.

Another example of MNOs and their suppliers feeling unable to do customer security properly is SIMjacking. In 2013, Karsten Nohl warned that many SIMs in use were easy to hijack, because of features built in to facilitate over-the air software update. The industry retorted that it wasn’t a problem as SIM cards could run only signed software [1438]. In 2019, it emerged that governments had been using this for surveillance [1006]. We will discuss government surveillance, and the tensions it has generated with security since the crypto wars, in section 26.2.7.3.

22.3 Platform security

Another broad theme in phone security is whether the platform itself is trustworthy, or whether your phone might act against your interests. This has been a growing concern since programmable phones came along in the early 2000s. For more back story see the second edition of my book which describes the state of play in 2007. Briefly, before the iPhone came along, security was fragmented along the supply chain, with chip designers, chip makers, OS vendors, handset OEMs and MNOs passing the buck while they tussled over DRM and over control. MNOs did not want OEMs to have any relationship with the customer. As I remarked in the chapter on Access Control, Arm launched TrustZone in 2004; by 2007, several hundred viruses and worms were being detected in Symbian phones each year, and vendors responded with access controls, code signing, and so on. Apple changed the world in several ways at once. First, it broke the taboo on OEMs have a relationship with the customer, and made it much easier for third party vendors to write apps. They made the iTunes store central to a platform strategy, which they monetised by taking a share of both music downloads and software. This entailed a semi-closed platform. Devices could go online either through the MNO or via wifi, and could switch easily between the two as needed. The effect was to shift power from the MNO to Apple. Google launched Android the following year, with a strategy of making the platform as open as possible7, allowing anyone to write apps for Android phones. They aimed to provide a minimum level of trust, to enable the ecosystem to grow.

7given the regulators’ insistence that the baseband software which controls the device’s RF behaviour had to be locked down
They remembered that Microsoft had grabbed most of the PC software market from Apple in the early 1980s by offering a more open platform that got the network effects going in their favour and hoped to do the same with phones, leaving the iPhone as a niche product for the rich. This did not in the end happen, and we now have two large ecosystems that have converged in a number of ways.

Both the iPhone and Android launched with security architectures I describe in the chapter on Access Control; both approaches aim to separate apps from each other and to prevent them from subverting the platform itself. I also discussed in the chapter on Side Channels how a bad app could, for example, use the phone’s accelerometer and gyro to work out a password or PIN being entered into another app, even if denied direct access to the screen. The combination of rich sensors and a huge range of applications makes security and privacy services at the platform level rather complex. Both the Android and iPhone security mechanisms have been refined over time, with more controls added to block or mitigate the more flagrant abuses. However they can best be understood as an ecosystem, rather than as a list of protection options.

This ecosystem is truly immense. By 2019, 56% of all Internet access globally was from mobile devices, but 63% in the USA and 80% in India [1134]. It consists at the very least of the apps that run on the two families of mobile devices themselves, and the back-end services they rely on. The boundaries are hard to define. We probably have to include the ad ecosystems that app developers bundle with their products. Do we include the web services that mobile devices access from browser apps? Do we include voice telephony, now that this is migrating to apps like WhatsApp, Skype and Signal? What about other devices, from watches to cars, that run mobile operating systems and apps? It may be simplest to start with the app families.

22.3.1 The Android app ecosystem

Android is the most widely deployed end-user operating system, found not just in phones but in tablets, watches, TVs, cars and other devices – a total of over 2bn monthly active devices. Its platform security model is described by René Mayrhofer and colleagues from Google in [1134], and in section 6.2.8 I discussed the technical architecture. Actions are based on three-party consent: the user, the developer and Google should all agree. The implementation is that rather than giving a userid to the end user, as in a conventional *nix system, Android runs each app in a separate userid; data in private app directories is controlled by the app, while data in shared storage is controlled by the end user, and there are mandatory access control mechanisms to ensure that critical system data remain under the control by the platform, unless it’s rooted. So long as this does not happen, the user cannot be tricked into letting a bad app access or overwrite the data of other apps. The threat model includes everything from physical attacks and wiretapping through the exploitation of vulnerabilities in the operating system, libraries and other apps; it’s assumed that users will be tricked into installing malicious apps [1134]. This risk is mitigated to some extent by the Google Play Protect (GPP) service which scans those apps that are downloaded via the Play Store; Google does allow Android users to download
22.3. PLATFORM SECURITY

apps from other sources but since 2014 has uploaded them for scanning when
they’re first run. But the risk of evil apps is ever present, and many more apps
are somewhat predatory, even if they’re distributed by apparently respectable
businesses such as hardware vendors, MNOs and security firms. The sad fact
is that user data has become a major commodity; little else might have been
expected given that most apps are free and the ecosystem is driven as much by
ad revenue as anything else.

22.3.1.1 App markets and developers

App markets mitigate some security problems while amplifying others. As the
Android ecosystem is open, anyone can be a developer and distribute the soft-
ware they write through the Play Store. This makes a huge market available to
novice developers, who can get simple apps running with little effort (we now
start off first-year CS students on Java, the Android language of choice). The
fact you have to use the framework with the Android SDK constrains developers
in potentially useful ways. Although fragmentation greatly impedes the update
process for operating systems, app updates are easy if you use an app store that
pushes updates.

However the developer rapidly encounters both technical and business com-
plexity. Some simple apps are little more than a customised browser for an
online back end; others exercise a single feature of the phone in new ways, as
flashlight apps do. But how uniform is that feature? How many versions of
Android do you need to support? Do you need to test on hundreds of different
handsets? There are now test frameworks to help, but fragmentation is a real
issue if your app uses the rich hardware features on many modern phones. For
example, people developing contact-tracing apps for coronavirus have struggled
with the variation in bluetooth performance between different handsets. An-
other example is where developers want to protect really sensitive information,
such as key material in banking apps. Arm hoped that developers would use
TrustZone but this turned out to be so hard given the variation between OEMs,
handsets and software versions, that most turned to obfuscation instead.

Business complexity can come from the application itself, or from the ecosys-
tem’s underlying economics: platform companies, device vendors, app develop-
ers, app publishers (who add all sorts of ads), ad networks, toolsmiths and end
users all have different incentives. There are different rules for paid apps, apps
allowing in-app purchases and free apps. The rules for identifying users are com-
plex: the user’s consent is needed to use some UIDs (IMEI, IMSI, phone number
and ad ID) but not others such as MAC address and hardware fingerprint.

22.3.1.2 Bad Android implementations

The first bundle of systemic security problems to become obvious as Android
became widespread around 2010 was the poor quality of the engineering work
by many of the OEMs who licensed it. One example was factory reset. There’s
a thriving trade in second-hand phones, as rich users buy the latest models
and their old phones end up being sold. You might think that when you do a
factory reset on your phone, that clears all your personal information, not just
from shared storage but from app storage as well. But it’s hard to get this right because of all the interactions with how Flash memory is organised on a typical phone; there may be an embedded multimedia card (eMMC) and virtual SD card, with their own wear-levelling mechanisms. If the OEM’s engineers don’t take the trouble to implement secure deletion, then the all-too-common outcome is that someone who buys your phone second-hand can retrieve the Google master cookie and access the Gmail account associated with the phone [1592]. For several years now I’ve always bought Google’s own-brand Nexus phones and never sold them after use, but many people get phones subsidised by a contract and locked to the MNO, which sells them in second-hand markets afterwards – often in less developed countries. (It is prudent to assume that phones in LDCs have been rooted and had RATs installed by local distributors.)

These quality problems extend to TrustZone and its Trusted Execution Environment (TEE), as implemented by various chipset vendors. For example, Qualcomm’s TEE system lets a trusted app (TA) can map in memory regions of the host OS, and as a result any insecure TA can let an adversary root the device. Other problems allow attacks on the TEEs of the other four vendors: the software security mechanisms used in trusted environments lag the state of the art by several years, with absent or weak ASLR, excessively large TCBs, information leaks through debugging channels, no execution prevention, multiple side channels and no good ways to revoke or vulnerable wicked TAs – of which there are plenty to let the attackers in. See David Cerdeira and colleagues for a survey of these issues [374].

However the biggest problem with Android implementations is poor after-sales support. Many OEMs only support the version that’s currently being actively marketed; they are reluctant to spend engineer time backporting fixes to old versions. A 2015 survey revealed that 87% of active devices were insecure, averaged over 2011–15, because they were running versions of the operating system that contained known vulnerabilities. In many cases, the OEM simply did not make fixes available. This had already been identified as a problem by Google by 2011; the company offered OEMs access to cut-price components if they undertook to patch their systems, but this got little traction. Google now offers certification programs for both vendors and apps. But the problems go deeper than just OEM engineering effort. If a vulnerability is found in, say, the OpenSSL or Bouncy Castle cryptographic library, this fix has to propagate to Linux, then to Android, then to each OEM, and then in many cases to each mobile network operator – as the MNOs control updates, at least for phones that are locked to the network. Each of these steps can take several months, and each can be neglected for commercial reasons [1703]. This raises thorny issues around coordinated disclosure, which we’ll discuss in section 27.4.1.1, and regulation, which we’ll discuss later in the chapter on Assurance.

22.3.1.3 Permissions

Consent has been a wicked problem from the beginning, as we noted in the chapter on Access Control. In early versions of Android, an app’s manifest specified the access rights it demanded and the user would have to approve them all on installation in order to run it. This led to widespread abuse, as
most users would just click approval to get the installation done, and a lot of utility apps became machines for harvesting and reselling your address book, browser history and other personal data. Already in 2012, research showed that only 17% of users paid attention during installation, and only 3% could answer basic questions about what was going on [612]. In 2015, Android 6 moved to the Apple model of approving access to such resources on first use. Indeed, progressive restrictions of the more dangerous permissions have driven platform evolution more than anything else. Android 6 also made fine-grained location access a separate permission; Android 7 limited apps’ access to the metadata of other apps; Android 8 randomised MAC addresses and mandated the use of a single Advertising ID for monetisation; Android 9 limited access to sensors when an app is in background mode and restricted access to the phone and call logs; and Android 10 restricted location access in background mode.

Google now provides several dozen permissions, and developers can define their own custom permissions when making services available to other apps; thousands of these are defined by hardware vendors, MNOs, security firms and Internet browsers [673]. These further balkanise the ecosystem and make it even harder for users (and developers) to understand.

An analysis of the consent problem by Yasemin Acar and colleagues breaks it up into comprehension of permissions, and attention to permissions, by both users and developers [10]. There are both usability and incentive failures on both sides. It’s clear enough why a predatory flashlight app wants access to my address book; many failures are more subtle. Developers are just trying to make stuff work so they can ship it, while users are just trying to access some service or other. Developer usability is a significant source of bugs; we’ve noted this elsewhere (e.g. in section 5.5) but it looms larger in appified ecosystems as the developers have to drive the application framework APIs to get useful work done. A substantial minority of developers request more permission than they need out of ignorance or confusion, and this holds even for system apps whose developers should know better. Google failed to implement fail-safe defaults; the APIs are confusing and poorly documented. This drove developers to copy each others’ code via fora such as stackexchange, to an even greater extent than with conventional development.

22.3.1.4 Android malware

As Android is an open platform, for which anyone can write apps, it has attracted a lot of harmful software. As we mentioned in section 22.1.4, premium-rate phone malware arrived in 2006 with the Red Browser worm; Android’s arrival turned mobile malware from a niche activity into a mainstream problem. Definitions here are hard, as many apps are harmful in different ways to at least some people; here I focus on apps that act secretly against the interests of the user that installed them. I’ll discuss bad programs installed by OEMs and MNOs later in section 22.3.1.6.

Malware can be bulk or targeted, and it can come from private-sector crim-

---

8It also drove Acar and her colleagues to look at usability from the developers’ viewpoint, creating an important new area of security research which I mentioned in the research problems section at the end of the chapter on Access Control.
inal or state actors. Most of it by volume is of the bulk private-sector variety, and most of that comes through regular distribution channels. As well as the millions of apps in the Play Store, alternative markets are widely used, especially in countries like China and Iran where the Play Store is censored. The largest single source of malware has been the Play store, with over 20% of apps being harmful at some times, while some alternative markets have on occasion removed most of their apps for being harmful. Apps may be born harmful, or libraries on which they rely may become bad, or the bad guys may buy failing app companies, just as they snap up domains of former banks. One of the biggest crime rings exposed recently did hundreds of millions of dollars of ad fraud by buying Android apps and using their user data to train bots that then clicked on ads [1575]; such scams exploit other kinds of malware too. The measurement problems are non-trivial, as over 60 anti-virus firms label apps using different criteria and classify them into different families. There are several hundred families active at any one time.

A 2018 survey by Guillermo Suarez-Tanguil and Gianluca Stringhini analysed 1.2m samples collected over 2010–17, and classified them into over a thousand families [1673]. Since 2012, most of them have involved repackaging, where the malware dev takes a legitimate app (the carrier and adds harmful code (the rider). This is industrialised by repackaging many benign carriers with variants of the same malicious rider. The riders may try to root the phone for persistent access, and drop a remote access Trojan (RAT) that can earn money at the direction of a command-and-control server, just as with regular PC malware. The monetisation strategies have evolved; in 2010 the focus was on making premium-rate calls, but by 2018 it had shifted to ad fraud and the exfiltration of personal information. The great majority of riders use obfuscation tricks such as encryption, while only a quarter of benign apps do this (Facebook’s app uses obfuscation as a defence against user data and keys being stolen by malware, particularly RATs that root phones). Riders also now mostly native code rather than java.

Banking Trojans stand out among the more targeted varieties of private-sector malware. A common approach is the overlay attack where the malware tricks the user into allowing it to use Android Accessibility Services, which enables it to build an overlay over (for example) your banking app so it can capture the screen and input data, under the control of a remote command server [368]. Android malware has been stealing bank SMSes for some time, and Google has pushed back by allowing only approved apps the permission to read SMSes; the latest development in 2020 is that the Cerberus banking malware can now steal Google authenticator cookies too [402].

States already used targeted malware in intelligence and law-enforcement missions, and by 2012 vendors such as Gamma had produced mobile-phone versions of their products that were found in multiple jurisdictions [1116]. Such malware also seeks root access but implants spyware. Recent examples of bulk malware deployment come from Turkey, which in 2018 was using man-in-the-middle devices on the Türk Telekom network to deploy spyware [1104], and China, which sets website traps for Uighurs’ phones [365]. Bulk state-actor malware can include doctored versions of apps in some jurisdictions; Skype was available in China from 2005 only through a local distributor, Tom Online, which
repackaged it to scan for words forbidden by Chinese censors. After Microsoft bought Skype, they took back control from 2013, but the app was banned from app stores accessible in China from 2017 [1222].

There are other technical abuses whereby apps defeat the permission framework while stopping short of rooting your phone. Joel Reardon and colleagues ran 88,000 Android apps in an instrumented virtual environment to look for abuses of side channels by apps to communicate with each other [1444]. They found two large Chinese companies, Baidu and Salmonads, using the SD card as a covert channel, so that ads which could read the phone’s IMEI could store it for those which could not. They also found 42 apps getting the IMEI when they shouldn’t, using ioctl system calls, and over 12,000 with the code to do so.

22.3.1.5 Third-party services

Mobile phone apps typically incorporate third-party services to support ads, social network integration and analytics for a range of purposes from crash reporting to A/B testing. Such services can track users across multiple apps, even without their consent. An example of what can go wrong comes from CamScanner, an app downloaded by over 100m people for scanning and managing documents. At some point, the app was updated to add a new advertising network that contained a malicious module. Negative reviews led antivirus researchers to take a look, and it turned out that the module was dropping Trojans on to people’s phones [726].

Third-party services are a fairly opaque part of the ecosystem, as they are not directly visible to the user. Some light has been shed by a survey carried out by Abbas Razaghpanah and colleagues, using a VPN app used by 11,000 volunteers to monitor traffic to and from their phones [1442]. They mapped over 2,000 advertising and tracking services (ATS), including hundreds that had not previously been reported, and found that a substantial minority (39%) did cross-device tracking; 17 of the top 20 had a presence on the web as well as in the app ecosystem. Eight of the top ten reserved the right, in their privacy policies, to share data with other organisations. The largest of all were Alphabet and Facebook, but firms whose whole business consists of ATS, such as Chartboost, Vungle and Adjust, have a significant share and are relatively unknown to users. App developers often use several such services simultaneously. Paid apps have the fewest trackers, free apps have more, and free apps that allow in-app purchases, often of premium services, tend to have the most.

Mutual trust issues are discussed by Yasemin Acar and colleagues [10]. App developers have to trust ad networks, as they execute in the app sandbox and inherit its permissions. Ad libraries exploit apps in various ways, such as loading insecure code from web services and stealing users’ private information; app developers return the compliment by stealing money from the networks with fake click events, just like malware developers. (The boundaries are a bit fuzzy, as they were before in the world of the PC; there’s predatory behaviour at just about every layer of the stack.)

There are many examples of children’s apps collecting personal data without
parental consent, contrary to the US Children’s Online Privacy Protection Act (COPPA), and other practices contrary to the EU GDPR and its ePrivacy Directive. This may pose a challenge to EU regulators, as the ATS industry is overwhelmingly based in the USA, and amounts to a substantial invisible export. Even from the viewpoint of the US authorities, most of the ATS specialists don’t even have a COPPA policy, leaving regulatory compliance to their customers.

22.3.1.6 Pre-installed apps

Julien Gamba and colleagues studied the firmware distributed by over 200 vendors worldwide [673]. Distributions typically reflect a partnership between a handset OEM and an MNO, with various affiliated developers, ad networks and distributors. They can be poorly controlled; there have been multiple cases of malware finding its way in, as well as software to do mass-scale data collection for commercial or regulatory reasons. Some phones also have diagnostic or support modes that could be exploited by wicked apps. Most of the pre-installed apps are not available in the Play store and thus appear to fall outside the conventional framework. Some are from firms like Facebook and AccuWeather which are known to collect personal data aggressively; many of these are not the public versions of these firms’ apps; and many pre-installed apps use mobile analytics or targeted advertisement libraries. What’s more, 74% of the non-public apps do not seem to get updated, and 41% remained unpatched for 5 years or more [673]. Many have sensitive custom permissions in order to perform such tasks as mobile device management for enterprise customers, call blocking, and VPN services. Behavioral analysis showed that a significant proportion of pre-installed apps could access and disseminate user and device identifiers, configuration and current location. The domains most contacted by such apps were Alphabet, Facebook, Amazon, Microsoft and Adobe.

22.3.2 Apple’s app ecosystem

Apple’s ecosystem has always been more closed. When the Mac was competing with the PC it was one hardware platform against many OEMs; the same pattern followed with the iPod, where Apple demanded 30% of music sales, and it continued when Apple launched the iPhone. The business model was much the same as a gaming console. Apple is the only hardware vendor and demands 30% of software revenues, as well as 30% of in-app purchases9. The company used its control of the hardware and the operating system to implement rights-management mechanisms to protect its aftermarket revenue. Apple does due diligence on developers; the app vetting process is substantially tougher than Google’s; iOS apps submitted to the App Store are only allowed to use the publicly-documented APIs [1645]. Researchers have dug into the iOS ecosystem a lot less, but nevertheless a few things can be said.

The overall protection against malware is the best of any mass-market system. Indeed, when our own university’s finance division has asked for advice on how to protect high-value transactions against phishing, my advice has been

9A potential future antitrust issue, though it was relaxed slightly in April 2020 when Amazon was allowed to sell movies on Apple devices without giving Apple a cut [765].
simple: buy an iPad on which you run the bank’s authenticator app to release payments, use it only for payments, and keep it in a safe the rest of the time.

However, the protection isn’t entirely bulletproof, and various actors have found workarounds.

First, there’s a long history of of hobbyists and others ‘jailbreaking’ Apple devices, starting with people who objected to DRM or who wanted to sideload their own apps, as they can with Android. As jailbreaks come out, Apple patches them; so at least the company has an incentive to patch its devices up to date, rather than abandon them after sale as the typical OEM does. Sometimes this isn’t possible, as when the exploit is of the device’s boot ROM; for example, the 2019 Checkra1n jailbreak will liberate most devices sold before 2017 [728]. And the forensics industry lives by the Checkm8 jailbreak, which exploits the boot ROM of all iPhones from the 4S to the X, enabling police and law enforcement to bypass the user PIN [728]. This was used by the FBI to access a phone seized from one of the San Bernardino terrorists, as I describe in section 26.2.8, and now more widely in the forensic ‘kiosks’ sold to the world’s police forces, as I describe in 26.5.1. There’s also a market for carrier unlocking, and state actors are willing to pay millions of dollars for such exploits in order to install malware on targets’ iPhones. Cybercriminals also do it: in 2019, Google’s Project Zero revealed iOS exploits that were being used in the wild to infect iPhones indiscriminately [187].

Second, Apple let large firms have ‘enterprise certificates’ which let iOS developers do as they pleased. This led to abuse and spats, with Facebook’s enterprise cert being suspended until their app stopped infringing App store policy; Google’s app on the iPhone had a similar experience, and suddenly lots of abuse by porn, gambling and spyware apps came to light. They had been abusing enterprise certificates and hiding in plain sight in the app store [1542]. Many had got their enterprise certs by pretending to be helpline apps from MNOs in less developed countries [1061].

Overall, the malware issues are less serious than with Android. But the same market forces apply, and so ad abuse still happens. Many popular apps (including dating apps such as Grindr and OkCupid) share a lot of data with advertisers, and are still allowed in the Apple ecosystem [1597]. The same holds for apps you might expect to be more privacy conscious, such as VPNs and ad blockers – where the privacy exploits come in through embedded ad networks, as in the Android ecosystem [1576]. And Apple, too, has been progressively tightening up the permissions apps need; for example, iOS13 refines geodata from ‘allow’ on installation to ‘allow once’ and ‘allow while using app’, and also curtails the use of wifi and Bluetooth to determine location – causing the same kind of complaints from developers [403]. The two stores share some political problems, such as the fact that they both allowed an app used by men in Saudi Arabia to control the movements of their wives, daughters and servants, as I discussed in section 2.5.4. Occasionally, they do diverge. Apple is more aggressive than Google at removing ‘bad’ apps, though this can sometimes get them a bad press. During the 2019 protests in Hong Kong, Apple banned a crowdsourced protest safety app that demonstrators were using to avoid the police, claiming “Your app contains content – or facilitates, enables, and encourages an activity – that is not legal ... specifically, the app allowed users to evade law
enforcement”, while Google left the Android version up [1136].

### 22.3.3 Cross-cutting issues

The convergence of the two ecosystems is leading to a growing number of cross-cutting issues, in addition to the issues such as SIM swapping that we discussed in section 22.2.5 and that largely stem from the underlying MNOs themselves.

These largely relate to poorly engineered apps and rapacious ad ecosystems. Quite simply, when billions of people entrust their financial lives, their social lives and even their sex lives to apps, then poorly-written apps can cause real harm. The specific application issues have been discussed in many other chapters of this book. Here, one example may suffice to put things in context. It illustrates a problem that many app developers just don’t think through – that of revocation. In fact, when assisting in the design of a payment app, we spent about half of the security-engineering time working out in detail how we’d cope with stolen phones: how payments could be blocked quickly when alerts came in from different stakeholders, what would happen when the crime victim walked into a shop the following day and bought a new phone, whether you’d rely on the phone shop to authenticate them or make them call a bank contractor, how you’d deal with phone OEMs who had their own backup and recovery services – an absolute mass of mind-numbing detail. That’s what real engineering comes down to: working with your supply chain and thinking through both the customer experience and the possible abuse cases.

My example of what can happen when you don’t pay enough attention is FordPass, an app that enables you to control a rental car so you can track it, lock and unlock it, and start the engine – even several months after you’ve returned it to the rental lot [724]. There are many more cases, but this is enough to illustrate that poorly designed apps can expose other systems, including safety-critical ones.

The threats from poorly written apps cover the whole spectrum of confidentiality, integrity and availability. The consequences of goods relying on apps that are no longer maintained are such that the EU passed a law in 2019 requiring vendors of goods with digital components to maintain these components for at least two years and for longer if that is a reasonable expectation of the customer. This means that car apps have to be maintained for ten years after the last vehicle leaves the showroom. We’ll discuss sustainability further in the last chapter of this book.

### 22.4 Summary

Phone security is a fascinating case study. People have been cheating phone companies for decades, and since deregulation the phone companies have been vigorously returning the compliment. To start off with, systems were not really protected at all, and it was easy to evade charges and redirect calls. The mechanism adopted to prevent this – out-of-band signalling – proved inadequate as the rapidly growing complexity of the system opened up many more vulnerabili-
22.4. SUMMARY

ities. These range from social engineering attacks on users through poor design
and management of terminal equipment such as PBXes to the exploitation of
various hard-to-predict feature interactions. The main disruptive force was the
development of premium-rate services that enabled people to steal real money.

On the mobile front, the attempts to secure GSM and its third, fourth and
fifth generation successors make an interesting case study. Their engineers con-
centrated on communications security threats rather than computer security
threats, and on the phone companies’ interests at the expense of the customers’.
Their efforts were not entirely in vain but have led to an immensely complex
global ecosystem that has become the subject of significant political tussles,
particularly over the control of 5G infrastructure.

The dominating factor, however, is the nature of the mobile app ecosys-
tems. The Android ecosystem has attracted hundreds of thousands of develop-
ers, ranging from firms like Uber that have built apps into major international
businesses, through apps offered by many established businesses and a host of
specialist tools, to a substantial criminal fringe. The Apple ecosystem is some-
what more regulated but similar in a number of respects. Many apparently
innocuous apps in both ecosystems can be abused in interesting ways, and the
ad networks they use are a pervasive threat to privacy.

Research Problems

The interaction between communications, mobility, platform security, and the
protection distributed systems, continues to be fertile ground for both interesting
research and expensive engineering errors. We have explored a lot of the issues
over the past ten years in the mobile app ecosystem, mostly in the Android part
of it where most of the problems occur. Mobility is now extending to all sorts
of other devices, from your watch to your car, and many of the issues around
app ecosystems are arising with smart speakers and other domestic devices.
Given the sheer scale of these new emerging ecosystems, we will need innovative
security scanning tools to find out what’s going on. One approach is to build
honeypots and look for the attack traffic that will appear in due course; a
somewhat more forward defence may be to analyse the companion apps used to
control IoT devices and infer vulnerabilities from them [1788].

Further Reading

Information about the world’s phone systems is scattered across a large number
of standards documents that can be rather heavy going, while app platforms at
least have official guides, white papers and developer communities. Keeping up
with the latest exploits is a matter of following the security blogs and tech press.
There are some good surveys of specific subproblems, which I’ve cited in the
relevant sections, but I’m not aware of any good books or survey papers of the
overall phone security scene. Perhaps that’s inevitable; now that more people go
online via mobile devices then from laptops or desktops, mobile security touches
one way or another on much of the subject matter of this book.