Cracking GSM Cellphones

Description: Steve and Leo discuss the state of GSM (Global System of Mobile communications) cracking. Steve shows where to purchase the required hardware, from where to download the software, and just how easy and practical it has become to "crack" the old and very weak "security" employed by the three billion cellphones now in worldwide use.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-213.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-213-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Music and Spinner.com, where you can get free MP3s, exclusive interviews, and more.

This is Security Now! with Steve Gibson, Episode 213 for September 10, 2009: Cracking GSM. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that covers all things secure and peaceful and calm, and all things insecure and horrible and nightmarish. And one guy does it all. He's our expert on security, the one and only Steve Gibson of the Gibson Research Corporation, GRC.com. Hey, Steve, how are you today?

Steve Gibson: Well, Leo, you know we almost didn't have this podcast.

Leo: What?

Steve: Yeah.

Leo: Did you almost get run over by a bus?

Steve: No, I almost didn't finish Michael McCollum's newest book in time.

Leo: Were you reading feverishly? So you read the third volume of the Gibraltar series in less than a week.

Steve: Well, he gave it to me - I got it Friday afternoon. So that's where Labor Day Weekend went.

Leo: You read it in three days.

Steve: And, well, I couldn't put it down, literally.

Leo: Oh, it's that good.

Steve: I mean, yesterday morning from 7:00 to 10:30 I read, and my eyes were a little wet at several points. I mean, it's...

Leo: You cried? Oh...

Steve: Well, it's - he does some good character development, and you care about these people, and it's...

Leo: Oh, that's neat.

Steve: It's just spectacular.

Leo: Oh, I can't wait.

Steve: I mean, it's everything I want in hard sci-fi. It's just great. So, and I found enough little typos, I could sort of see, you know, like when you go and rewrite a sentence, you scan it, and you leave an 's because it used to be possessive and now it's no longer, or you get the tense wrong on part of it. So I'm a slow, careful reader. And I found a bunch of stuff, I mean, enough that it was probably worth his while having me read it. And so it's, as of yesterday evening, it's back to him. And I don't know what his turnaround for turning it into an eBook will be. But, I mean, it's one now. It seems pretty ready to go.

Leo: What's the name of it? Gibraltar...

Steve: "Gibraltar Stars" is the third in the trilogy. The first one was "Gibraltar Earth," "Gibraltar Sun," and then "Gibraltar Stars." And what I like about it so much is, as with all of his things, there are really unique constructions. Obviously, it's fiction, so it's contrived. But he stays very faithful to the rules he sets up. One of the things that really

bugged me about "Star Trek: The Next Generation" was Q, that ridiculous, omnipotent alien. Because if you have someone who's omnipotent, why even bother?

Leo: It's just a plot device. It's like...

Steve: Yeah. Well, it's a cheat. He could just blink you wherever you wants to blink you or do anything to you. It's like, okay, now suddenly, if there aren't any rules, then you don't have any problems.

Leo: It's a total cheat.

Steve: And so Michael McCollum establishes a universe with limitations and then always builds really intriguing plots around them where you're sitting there thinking, oh, god, okay. What are we going to do now? God. Anyway, I don't want to say too much because I just - I can't recommend it highly enough. And I have had enough feedback from people who have heard us talking about him and Peter Hamilton and our other, the other authors that we like so much, that they've been turned on to this stuff and really enjoyed the read. So anyway, this trilogy is finished, and it's really great. And we can have a podcast now.

Leo: That's good. Lucky thing you finished it.

Steve: Yeah.

Leo: We're going to cover - today we're going to cover something you promised last week, which was cracking GSM?

Steve: And we've had people send feedback, wondering about this. And we've even read Q&As where people are saying, hey, you know, if I use a cellular modem...

Leo: How safe is it? How safe is it?

Steve: ...just by itself, how safe is it? And I've known that fundamentally it wasn't safe because I've sort of felt, I mean, I sort of moved through this domain. And I remember seeing somewhere that the encryption was based on three shift registers, which immediately says oh, goodness. And now I know exactly how bad it is, and we're going to talk about it today. Basically I'm glad I'm over on Verizon with - and not using GSM. It's completely cracked. It's completely broken.

Leo: So any bad buy could listen in on your conversations.

Steve: And not for much money. It turns out - oh, I meant to tell you before we started recording, but you can do it now: www.ettus.com is the group that offer a beautiful, I

mean, just spectacular technology, cute little software programmable radio receiving set.

Leo: Oh, neat.

Steve: It's based on the GNU Radio project that John Gilmore has funded for about a third of a million dollars. And basically after a day of sitting here doing the research, if I had any inclination, everything that I need to listen in on someone's cellphone conversation, all the software, it's all open source, it's beautifully designed, you can program it from Python or C, everything is there to do it. And you need about a thousand dollars for the radio receiver equipment, and then any PC. It's just - it's done.

Leo: Oh, man. You know, I remember talking...

Steve: And it's not like...

Leo: Go ahead.

Steve: ...hundreds of thousands of dollars or corporate or government level. That's just not the case.

Leo: I remember talking to Woz some time ago. He used to like to sit and listen to, what was it, he had a little receiver, he would listen to cellphone conversations, I think, or maybe - oh, no, it was long distance calls coming over satellites, unencrypted over satellites. And he would just tune in and listen to the calls. Sounds like this is almost as easy.

Steve: Well, and back in the day, before we went digital, when we had analog cellphones, I did run across a little scanner, and you could turn it on, and you would only hear one side of the conversation because they were on, the transmitter and receiver, on different frequencies. But it was really embarrassing what you heard. It was like, oh, goodness, I hope this guy's wife isn't listening to this.

Leo: Yeah, exactly.

Steve: I mean, it was really - it was just out there in the open. And in fact I refuse to have important conversations with my attorneys over the cellphone because I knew firsthand that it just wasn't secure. And we'll talk about the various, well, in detail about the technology, why this is so badly broken now and what it means in terms of practical attack scenarios.

Leo: Oh, that's shocking. All right. We're going to get to that in just a second. Before we do, any errata or security news? I guess this is the second Tuesday of the month, isn't it.

Steve: Well, yeah. We're recording a day earlier than we normally do in order to make room for the Mac event which is happening on, what, on the 9th, I guess.

Leo: Yeah, 09/09/09, yeah.

Steve: Is that going to be the tablet, or more iPods?

Leo: Nobody knows, of course, because Apple doesn't say. But the general consensus seems to be the tablet will be next year. These will be just iPods.

Steve: Yeah, that's sort of what I heard, too. So, yes. We're standing on Tuesday. And as of half an hour ago I checked Microsoft's advanced bulletin notification deal. And all they've got is their very generic, five really bad problems, all remote code execution attacks.

Leo: Oh, boy.

Steve: Microsoft has acknowledged what we talked about last week, which was this problem with IIS and the FTP vulnerability. So I want to reiterate to all of our listeners, anyone who's, for example, got IIS, which is the web server with FTP that's installed, you know, the so-called "personal web service," which it's very possible to have running even if you're not some big corporation who's serving these things. FTP is vulnerable for remote attack. So you absolutely want to shut that down. Microsoft is not expected to have a fix for it by today. As far as we know it's not one of these five critical bulletins because it's happened much too quickly for them to respond. They have said they will fix it as soon as they have a patch available, meaning probably an out-of-cycle patch because this is potentially a big enough problem that they're not going to let this thing languish for long. But as we're recording this, we're expecting five critical bulletins from Microsoft.

Leo: I've got them, if you want me to read just the headlines of them.

Steve: Oh, sure.

Leo: Vulnerability in Jscript scripting engine, your favorite, scripting.

Steve: JavaScript.

Leo: Love that. Vulnerability in DHTML editing component, ActiveX control. This is another remote code execution. Vulnerabilities in Windows Media format. Oh, that's not good. Could allow remote code execution. Vulnerabilities in TCP/IP.

Steve: Ooh. Okay.

Leo: Privately reported vulnerabilities in TCP/IP. The vulnerabilities allow remote code execution if an attacker sends specially crafted packets over the network to a computer with a listening service.

Steve: Oh, goodness. That's way [indiscernible].

Leo: Firewall will protect you. Okay, firewall...

Steve: Well, yeah, if you've got one.

Leo: They say firewall best practices and standard default firewall configurations can help protect networks.

Steve: Interesting. I can't wait, well, we'll definitely have some news about that next week when I know what's going on with that because that sounds really important.

Leo: This is Windows Vista, Server 2008, also important for 2000 SP 4, Windows Server 2003.

Steve: Wow.

Leo: And finally, vulnerability in wireless LAN autoconfig service could allow remote code execution. We've been telling people to remove that anyway. That had a problem before.

Steve: And it's just dumb. It's one of these things that it's always upset me that Microsoft has this stuff turned on by default, even though the majority of users, just like Universal Plug & Play, the majority of users aren't using it and don't need it. But it's there just in case. And, whoops, it's vulnerable.

Leo: Almost all of these look like they're Vista specific.

Steve: That's interesting. And so - it's interesting because, before we began recording, you were asking me hypothetically whether I expected, now that we have Windows 7 with its presumably enhanced security, if these problems are going to be going away. And my reaction was, uh, I don't think so. I mean, and so here's Vista that's got new problems in it.

Leo: Yeah. Almost all of these are critical for Vista. Now, I don't think - I don't know. Do they push patches yet for 7?

Steve: I do know that when you install a new 7 it automatically, I mean, there's already updates for it, even though it's not been released. I have installed the RTM, the Release To Manufacturing version. And immediately upon getting it going, it's like, okay, let's go do some updates. Oh, here they are.

Leo: Right. And we do get updates. But I think those are not part of the second Tuesday cycle yet.

Steve: Oh, absolutely not. I don't expect that to happen until it is, in fact, released.

Leo: Okay. Wow.

Steve: Well, now, wait a minute. That's - I'm trying to remember whether I got updates. I think if you have Windows 7 running, then the system says, oh, you're a Windows 7 user. You're an early bird, but here's your updates. So I think it's part of - I think it's already running.

Leo: Maybe they are doing it, yeah. Well, we're only a month away.

Steve: Yes. Time flies.

Leo: Wow.

Steve: Also I wanted to mention to any users of OpenOffice that multiple vulnerabilities have been disclosed in the Word.doc format. They are remote execution, remote code execution attacks. So I know that people who have, like, said, okay, we're not going to follow Microsoft any longer with Office, we're going to go to OpenOffice, and it's a beautiful piece of work, there are problems there. And patches are available. So if you're an OpenOffice user, it's time to go check for updates and keep yourself current because you want to make sure that those don't get you. Although I think the attack target size is smaller for OpenOffice users than for, like, Microsoft Office users. But still, everyone's trying to exploit these things these days.

Also the latest update to the Mac OS, so-called Snow Leopard update, which brings us to version 10.6. We were down in the 10.5s up until now. So we knock up to that second digit. And what's in the news is that, unfortunately, when you do bring yourself up to 10.6 with Snow Leopard, it brings along a known vulnerable previous version of Adobe/Macromedia Flash Player, which it installs, which is known to be insecure.

Leo: And downgrades you, even if you have an updated version.

Steve: Precisely. It overwrites the current version with an older one which is known to be insecure. So I wanted to let our listeners know that they're going to want to update Flash after installing Snow Leopard in order to fix that.

Leo: Yeah. There's a bit of debate over whether that was a good practice or not.

Steve: Which?

Leo: Well, the problem is the Adobe patch came out two weeks before they went gold on the Snow Leopard.

Steve: Right.

Leo: So they, I think completely reasonably, said, well, we haven't had time to test this.

Steve: I don't disagree. You can easily imagine that they had basically a ready-to-go, release-to-manufacturing build and image. And it's like, okay, well, look, we'll just let Flash update itself afterwards.

Leo: And then what we don't know is how necessary it was to downgrade the existing version of Flash. I guess that depends on the installer and how the upgrade was performed.

Steve: Exactly. I was going to say that, to my mind, it wasn't deliberate. They were just probably overwriting everything in the system with their own stuff. And so they said, okay, let's just - we're installing OS X 10.6. Stomp on whatever's here and replace it with stuff that is what we now think is current. And in this case it wasn't current.

Leo: Yeah. So no harm, no foul now because you know about it. And I presume Apple will push a - I don't know. Maybe Apple won't. I think they think this is Adobe's issue.

Steve: Oh, yeah. I would say it's Adobe's issue. And of course Flash does inform its users from time to time. I would just say if you're able to - if you know you're potentially a victim of this, then it's worth updating to the latest version of Flash after you have installed Snow Leopard since it will have moved you back a bit. And again, I don't think it's - it's not the end of the world. It's not like a flaw in TCP. So, you know...

Leo: Well, there are exploits. There are exploits out there. One thing, and this is a side note, but I think a really great side note, Firefox in its latest version...

Steve: Mm-hmm. That was my next point, Leo.

Leo: I'll let you go. Because it happened to me, and I was so pleased.

Steve: Yeah. Currently in beta, both the v3.0 chain and the 3.5 chain, upgrade chain of Firefox, the next versions will begin warning users if their version of the Flash plug-in is out of date. So Firefox is beginning to take responsibility, at least in the case of Flash, like a major high-usage plug-in, for making sure that it's current, sort of as an extra benefit, an extra security benefit for its users, which I think is very cool.

Leo: Yeah. That is really nice. And again, not their responsibility. But since you're the browser, and you know what's going on, why not?

Steve: Yeah. And I had no errata this week. I did have just a fun little SpinRite anecdote to share with our listeners. It was a subject that we received through our sales email titled "A Note of Appreciation." He said, "Dear Steve and Co." - this is someone whose name is Barnett. He said, "I purchased SpinRite 6 back in December of '07." So, wow, two and a half years ago. "And up until" - actually two and three quarters years ago - "up until this week I really didn't find a need for it. We had a terrible thunderstorm come through here Monday. And while we didn't take a direct hit, apparently the bolts" - I guess he means the bolts of lightning - "were close enough to produce EMF in the network wiring. The damage was limited to one router and a very important server. Apparently the surge got through the UPS and scrambled the boot volume enough to fill the event log with disk alerts. A closer look revealed that the bad sector table on the disk itself was damaged. I have good backups, but it takes forever to restore that large box. So I gave SpinRite a shot at it first. I set SpinRite to Level 4 and just let it run. Six hours later, SpinRite reported success. The stats in SpinRite showed that there were 1,310 bad sectors recovered."

Leo: Oh. Is that a lot?

Steve: Yeah.

Leo: Okay.

Steve: That's, I mean, to have - I mean, SpinRite can be recovering sectors, but my guess is that this bolt of lightning sort of tipped the machine over, but that it was already close to having problems. To me that feels like long-term accumulated problems that SpinRite came along and said, well, I'm glad you're running me now. Let's fix all of this stuff. So he says, "I rebooted the box, and it's been purring like a kitten ever since. As you can imagine, I'm one very happy user."

Leo: Yay.

Steve: So, yeah. Another fun story.

Leo: Yay, that's really great news. All right, we're going to talk about, in just a second, we're going to talk about getting - cracking GSM.

Steve: Switching back to land lines.

Leo: Yeah. That might be the subtext, the subtitle: Why you don't want to use a cellphone for anything important. So this is - this applies to current GSM phones; right? This is not...

Steve: Yeah. GSM, well, it applies to the world.

Leo: Every, yeah, because everybody uses it.

Steve: The acronym is Global System for Mobile. That's GSM, Global System for Mobile communications, GSM.

Leo: Okay.

Steve: It currently has three billion users worldwide. GSM has 80 percent of the cellphone market spread through 200 countries. There's a GSM alliance that are the group that sort of hold the spec and manage the spec. Everything about this is worrisome. I mean, from day one, the fact that they were keeping this algorithm, their cipher, a secret, rather than allowing it to be exposed publicly, tells you, I mean, it was like the first thing to worry about. We've talked often about the dangers of relying on security through obscurity. It's not that some obscurity can't also be useful. But relying on the obscurity is something you never want because nothing remains obscure forever.

Especially, and we've also talked about this, when every single cellphone user has a handset which is able to decrypt GSM. I mean, by definition. It's just like DVD players running in your living room that are decrypting Blu-Ray. Well, that didn't last very long, Blu-Ray encryption. Similarly, everyone with a cellphone is holding the technology to do the decryption because it has to in order for them to have the conversation. So it wasn't long before the so-called cipher algorithm in GSM was reverse engineered.

And we've also talked, for example, about the problems that WEP, the Wired Equivalent Privacy, the original oldest version of the WiFi cipher had. The problem was that it was designed at a time when we didn't have today's level of RAM, CPU power, power-saving technology. So the designers deliberately came up with an algorithm-sparse approach. And unfortunately, GSM was designed back with that same philosophy in that same era. Because it's an old spec. It's back from the '80s.

The idea is, again, very much like WiFi, or like WEP's WiFi, it is a pseudorandom bitstream cipher, meaning that it's not a block cipher. And we've talked about various types of crypto many times in the past. It's not a block cipher where you take a block of bits, and a sophisticated algorithm turns it into another block of bits where there's no way on examining it to see what the transform is between those. Instead, this is an XORing approach where you have a generator of pseudorandom data where, bit by bit, you XOR, you exclusive OR the output of this generator with the data you want to encrypt.

And when you, as we've also said before, when you do that, when you exclusive OR, essentially you are pseudorandomly flipping the bits of the so-called plaintext to create

the ciphertext. Then the person at the other end is able to generate exactly the same pseudorandom bitstream, so they flip the bits. And exactly the same bits that you flipped, they flip back; which, again, takes that ciphertext and returns it to plaintext, that is, decrypts it. So it's conceptually simple. And if you have a source of really good pseudorandom bits, that is, if the pseudorandom data generator is high quality, there's really nothing wrong with it except that there are problems with so-called known plaintext attacks. And we've talked about this actually just recently when we were talking about the attacks on WiFi, the sort of the slowly encroaching attacks. Remember two weeks ago we talked about the TKIP - I guess it was last week.

Leo: Yeah.

Steve: The TKIP attacks where they rely on the fact that the attacker knows some of the bytes in the packet. Well, if you know what the bytes in the packet are, and you know what the ciphertext is, since the relationship is just an exclusive OR, you can exclusive OR what you know and what you see as ciphered and get the key stream out of that. So this whole XORing is just not a very secure way, fundamentally not a secure way to do things. But it's incredibly inexpensive. It takes a few transistors, literally, to perform an exclusive OR operation. So it's because it's so economical in terms of hardware implementation - and even, if you did it in software, the same thing - that it tends to get used by older technologies.

So where do we get - well, first of all, I want to say that what happened in the news recently that we talked about a couple weeks ago that caused me to say, okay, I'm finally going to talk about GSM, was there was this news that some - that within a couple months there was going to be publicly available, open source technology to allow anyone to decrypt cellphone conversations. Well, that may well happen. But what's annoying to this hacker group is that these problems have been known for a decade and have been pooh-poohed. And in fact this GSM Alliance is still pooh-poohing these issues. In response to this recent news story, they said among other things that this would require the construction of a large lookup table of approximately two terabytes. This is equivalent to the amount of data contained in a 20-kilometer-high pile of books, they said.

Leo: Oh, yeah. And of course we'll be using books to store those tables.

Steve: And monks to transcribe the data.

Leo: What the hell? That's just FUD. Or what's the opposite of FUD?

Steve: Well, exactly. And I'm thinking, two terabytes. Then I think about your Cottage up there.

Leo: I'm just looking at one hard drive, it's two terabytes.

Steve: Exactly.

Leo: C'mon.

Steve: And then they said that - they said, "However, before a practical attack could be attempted, the GSM call has to be identified and recorded from the radio interface. So far, this aspect of the methodology has not been explained in any detail, and we strongly suspect the team developing the intercept approach has underestimated its practical complexity." So when I saw that, I said, okay, let's - and I wanted for our own listeners to sort of bring this home, to make this real. It's like, okay, how do you get this stuff out of the air? Because of course before we can start deciphering anything, we have to have something to decipher. And we've all got cellphones, but they don't have digital interfaces that send their bitstreams out.

Well, it turns out all of that work has been done for us, Leo. There's an incredibly cool technology called a USRP - I love it that you would tend to say "usurp" - the USRP, the Universal Software Radio Peripheral. It's produced by a company called Ettus. That's the guy's last name. So www.ettus.com will take you to his site. It's open hardware in the same spirit as open software, meaning that he's just producing it, not making a ton of money, but doing all of the hardware engineering work for people who don't want to do it themselves. But somebody who wanted to save some money and had the ability could certainly do that, as well.

It's a hardware platform, literally, about a seven-inch by seven-inch square circuit board. The first iteration, the USRP 1, or just USRP, had a USB 2 interface. You can then get daughter boards that span various ranges of radio frequencies. And this thing runs all the way from zero, that is, from DC essentially, to 5.9 GHz. So that's everything you could want. You can use it to experiment with GPS signals that are at a couple gigahertz, with AM through WiFi and beyond. This is a general purpose radio transceiving peripheral. The second version has a gigabit Ethernet interface rather than USB 2.0 because they wanted to be able to operate at larger bandwidths and so have a greater data flow in and out of this board.

The first one costs $700. The second one is $1,400. So we're no longer talking hundreds of thousands of dollars and arcane hardware and stuff that only large corporations and governments can afford. You can go on their site. You can click the button, "Buy This." Then they have daughter boards which configure it for different ranges of frequencies, and there's documentation about which one you want for GSM. So you get one of those. And then you get an antenna with a cord, and you plug it into your laptop. So...

Leo: Is this legal?

Steve: Everything is legal, even decrypting your own conversations, just not somebody else's.

Leo: So buying the equipment and recording the calls is completely legal.

Steve: Buying it, yeah, buying it, the knowledge, the ciphers, every stage of this is legal unless you decrypt somebody else's conversation. And of course you wouldn't want to do that by mistake. So this notion that this is difficult to do just no longer holds any water.

There's also a fantastic project called the GNU Radio project. John Gilmore has invested about a third of a million dollars in funding this. It is a general purpose software radio project developing all of the modules that go behind this piece of hardware. It's, of course, open source also. Lots of people contributing and doing all kinds of cool stuff. So, for example, I mean, you literally could build your own GPS system.

There's a company called Path Intelligence which uses this board, the software from the GNU Radio project, to track people in shopping malls, to aggregate data about the foot traffic patterns. They have a couple of these radios stationed around the mall. And by using literally the timing information from all the cellphones that everybody in the mall is walking around with, they're able to track individual people. And they, of course, don't care who these people are. But cellphones are generating their little handshake with the cell towers constantly. So that allows them, for example, to see how many, like how much traffic the various restrooms get, who stands in front of what window for how long, how many people go up the stairs versus go up the elevator or the escalator. And so they're able to basically track individual people using this technology.

So again, we're now at the hobby level. We're at the level where the hobbyist with a couple thousand dollars can - needs to know nothing about radio and even hardware. And even all of the preprocessing steps for demultiplexing the data and analyzing it and performing spectrum analysis and finding the channels and everything, all of that's been done. There's even some people have taken - they're not at the GPL licensing, but they are - so they're proprietary licenses, but free, but they're open source and free for personal use, where turnkey packages to pull all this data together have been produced. There's even one which abstracts this USRP, this Universal Software Radio Peripheral, making it look like a network device so that Wireshark, our favorite packet capture utility, is able to capture GSM packets and decode them and show you all the bits and all the protocols and everything going on in a stream that you capture.

So, I mean, we're way far along in making this possible. In my opinion, this GSM Alliance is - they're saying what they have to say politically; but, if they really believe what they're saying, that they're in serious denial because this is no longer James Bond government-level sci-fi stuff. It would be entirely possible for a company who wanted to do some surveillance of a competitor to equip a van with some of this equipment, spending only tens of thousands of dollars, park it across the street from a competitor, aim their antennas at the competitor's building, and spend a day just streaming in, sucking in all of the cellphone traffic that is being transacted by the employees within the building, and then drive the van off and decrypt those conversations offline afterwards and find out what was being said. I mean, it is no longer difficult to do. It's entirely possible.

So the problem is that, not surprisingly, this is old technology which was built to be safe enough then. One of the other concepts that we've talked about several times in the last few weeks is this - in fact, it started with this notion of how long was a voting machine secure. We talked about the idea that security has a lifetime. And you'll remember that one of the questions we dealt with in the Q&A last week was some guy said, well, if I stored something that was encrypted today, then waited 10 years or 20 years, assuming that that encrypted data was still valuable, what happens if decryption technology and cracking technology get so much better in the intervening decades that I can then decrypt something from history that's valuable that I wasn't able to decrypt at the time that it was current?

Leo: We had that question last week, didn't we.

Steve: Yup. It's a really good question. And so similarly, here when we talk about this GSM Alliance's pooh-poohing the idea that you would need two terabytes of data, well, back in 1980 that was, you know, terabytes, it's like, wait a minute, how many zeroes is that? Now you're, like, using those things for doorstops, Leo, those drives. So we have seen an increase in the practicality of attacks.

Now, the technology that GSM uses for generating pseudorandom data is unfortunately weak. And they did rely on it being kept secret, which of course is not something you can rely on. All these secrets are going to get out over time. There were assumptions over the years about the exact algorithm which were locked up in the silicon of chips. And at one point someone physically reverse-engineered the algorithm from the chips and figured out exactly what was going on. And it uses a technique that we've never talked about before. It's a so-called Linear Feedback Shift Register, LFSR.

The idea is you have a - first of all, a shift register is a sort of a - you can think of it visually as a long string of bits contained in a hardware register. And when, on the event of a so-called clock pulse, this shift register moves all of the bits, the ones and zeroes, one place to either the right or left, depending upon whether it's shifting right or shifting left. But for the purpose of this, let's imagine that this is shifting to the right. So you have a string of little bit cells. Upon receiving a clock pulse, every one and zero moves one cell to the right.

Well, you need something to fill the gap that was open. That is, if the bit in the first position on the far left moved to the second position, then you need to decide whether now what is the first bit of the shift register is going to be a one or a zero. What they do is they take some few bits stationed in various places in the shift register and exclusive OR those bits. So often, for example, it's the last three, like the far right bits of the shift register, the last three bits. They will be exclusive ORed, meaning that if you, like, if you count up the number of ones in the last three positions, if it's an odd number, then the result is a one. And if it's an even number, or zero, then the result is a zero. And so you feed that back into the front of the shift register.

Well, this is - it's an approach that's been known for a long time. It's - once upon a time, before we had really mature cryptography, it was - people looked at that and said, oh, wow, we're never going to be able to figure out what those bits are doing. The idea being that when you set the shift register up, and then you run it, that is, you clock it and clock it and clock it, there's a complex pattern of bits that ends up getting shifted into the front of the shift register. And after 19 clocks, for example, in the case of a shift register that was 19 bits long, well, then you begin to get bits at the end that scramble up what goes in the beginning. And before long it gets pretty complex.

So what GSM uses is three of these shift registers. One is 19 bits long. The second is 22 bits long. And the third is 23 bits long. So you've got three different shift registers. It's important that the period of the shift register, that is, the length of the shift register are different. And they're different in a complex way. This 19, 22, and 23, they came out of, you know, because 19 and 23 are both prime numbers, so they're going to have a very long period before - if you imagine these sort of rotating around before they come back into their original synchronization. So the problem is that what seemed really complex in 1980 and, like, oh, no one's ever going to figure this out, modern cryptographic analysis just looks at it and says, okay, what are we going to do after lunch? Because this is just not difficult to deal with at all.

The people that are doing the cryptography have come up with a whole bunch of approaches for attacking this. There's all kinds of weaknesses in the way this works. The system, by coincidence, 19 plus 22 plus 23, that is, the sum of the lengths, is exactly 64.

So one of the problems is that the entire state of the shift registers at any time has only 64 bits of complexity. Well, we know that that's no longer enough complexity. We're to the point with modern computing technology and modern storage and using, for example, the graphics processing units in graphics cards, 64 bits is worrisome.

It turns out that it is possible to use precomputation attacks against this pseudorandom generator. We've talked about precomputation attacks before, the so-called rainbow tables. A precomputation attack is one where you do a lot of work ahead of time to generate some tables which you're able to then use afterwards to essentially reverse an unreversible function. For example, rainbow tables have been used with hash functions where, as we know, with a hash function you feed a bunch of stuff in, and you end up with a result. Well, for example, if you were to hash a whole bunch of common passwords, you would end up with a rainbow table of the results of the hashing, so you simply - you look for the value you're searching for in the rainbow table, and it tells you what the input was that gave you that value.

Turns out that the same kind of thing can be done with this GSM stream cipher. There's a precomputation attack. And it was published thoroughly, completely, in 2003. A bunch of researchers laid it all out. They said, here's how we cracked GSM. We can either have - I think they had, like, a time-complexity tradeoff. You'd have to listen to two minutes of GSM cellphone traffic, and then you could crack the key that was used to encrypt this. After two minutes you could crack it in one second. Or if you listen to two seconds of GSM cellphone traffic, then you can crack it in two minutes. So if you have more input data, takes less time; less input data, more time. And they use then tables exactly like we were talking about, basically precomputation tables, the so-called two terabytes that the GSM Alliance was pooh-poohing and saying, well, you know, no one's ever going to be able to produce this.

Well, this cracking gang is putting together a project, very much like the SETI@home project, where a bunch of people who've got unused graphics cards, they have code that runs on the NVIDIA chipset graphics, running 32 threads in the graphics card, doing precomputation attacks, putting together essentially these tables, which will then, once they're assembled, be freely available to anyone. They haven't really done any breakthrough work themselves. I congratulate them on taking the theoretical papers and making them practical. But, and they understand this, too. What they'll be putting together is the network and the facility for making this available.

And right now you're able to download this stuff and run it on your machine and join the network and begin cranking out this data. I mean, this is happening today. So it's very clear that even if you didn't go for the distributed hobbyist level approach, that any major corporation that had any need, certainly any government, can now crack GSM. You're able to, due to the availability of this kind of inexpensive hardware, you can just suck in all of the GSM channels that are active in a given area, just stream them onto hard drives, and then crack them at your leisure.

Leo: At your leisure, yeah. Record them now, crack later.

Steve: Yeah. I mean, it is absolutely the case that we've got - we're using old technology, and storage and processing power has advanced to the point that it no longer provides us protection.

Leo: Well, and in the GSM Alliance's defense, I mean, obviously nobody's going to put them in a book. What they're probably trying to say is it's still a bit of a chore. It's not something that some guy with a scanner down the street can do.

Steve: It's certainly the case, you're right, it's not like you buy a scanner at Radio Shack, and you turn it on, and you listen to random conversations. So at this point you have to have some motivation to do it. There are other attacks which do not require this kind of table. I don't want to get into the details of it just because it's really complex. But, for example, if you knew somebody who was using a GSM phone, and you wanted to crack them, you're able to pretend to be a cell tower to their phone. If you monitor them, initiating a conversation, the way the GSM handshake functions is that the cell tower comes up with a 128-bit, pseudorandom, one-time token. It gives it to the customer and says, using the preshared key - in the SIM card is a 128-bit preshared key. The cell tower, who knows the customer's account, knows what SIM card they have with the preshared key. So the cell tower gives them a 128-bit token, which is a one-time token, says use your preshared key to encrypt this that I've given you, and give me the result to prove that you're you.

So there's an authentication phase. And unfortunately the same data is used to produce the session key, which is a big mistake. You never want to use the same data for authentication and encryption, which is a mistake that GSM has unfortunately made. And that's a weakness because it allows someone who's listening to that - this random number that comes from the cell tower is in the clear. So if you're listening to that conversation, you can then subsequently appear to be a cell tower.

There is no protection against re-use, which is another big problem. We know about the problems of re-use. So you can pretend to be a cell tower, give the same key to the user, and cause them, since their preshared key is static, you give them the same challenge, essentially, in this challenge handshake. They will generate the same session key, which now you have. And so you're now able to decrypt a conversation that you had previously without any use of two terabytes of tables.

There's, like, all kinds of problems. As I was reading through the research that's been done about attack after attack after attack on the GSM system, you just sit there sort of with your head in your hands thinking, oh, my goodness. If I were the person who designed this, and I was reading where the state of the art is today in cracking this, I'd just be thinking, whoa, I'm embarrassed. But they did the best job they could at the time with the resources that they had.

Leo: Whoa, I'm embarrassed. I'm embarrassed for you, man.

Steve: I'm embarrassed, oh.

Leo: It's so sad.

Steve: Oh, don't tell anybody else you were the guys that did this.

Leo: But as you point out, how long ago was this? 20 years ago? I mean...

Steve: Yeah.

Leo: As you point out, it might have been okay then. The idea of a two-terabyte table then might have been, you know, considered...

Steve: Oh, it was - oh, my god, back then, Leo, we had paper cards, right, and paper tape and, well, I guess we were beyond that a little bit. But we had, what, 10MB was a big deal. Now we're, you know, you're streaming terabytes of data out of your facility. I've got terabytes. We all have terabytes. It's just that there's been so much change in the technology from then to now that I cut these guys some slack.

The problem is, we're all still using, what is it, three billion people in 200 countries, 80 percent of the cellphone market is GSM, globally. And it's no longer safe. Yes, absolutely. I don't think anybody is going to be spying on their neighbors or caring what random conversations are. But if people depended upon it for real security, that becomes a problem. And we've only talked about voice stuff. But all this applies to SMS. So, for example, there are banks which are now, as we know, using cellphones and SMS tokens for security. And they're not safe.

Leo: I use them all the time.

Steve: Yeah.

Leo: That's how I log into my bank. I ask them to send me a token.

Steve: And again, what's the chance that some random person is going to be going after you? I agree it's slim. But targeted attacks, I wouldn't be surprised if, before long, we begin to see reports of GSM cellphone technology succumbing to specific targeted attacks. It could happen.

Leo: Yeah. Well, and you hit the nail on the head when you said this is the kind of thing a government or a business might do, as opposed to Steve Wozniak.

Steve: Well, hobbyists, motivated hobbyists certainly now have this within their grasp because all the hardware exists. You go to a website; you order the stuff. All the software's open source. The project will be making these rainbow tables available. There's all kinds of more active attacks, not just passive decryption attacks, but active man-in-the-middle sorts of attacks that GSM is also vulnerable to that I didn't even talk about. It's just it's absolutely not something that you could rely on. So at this point I would say to our listener who asked last week about GSM, or about cellphone Internet, I would say, well, this is where you really want to have your own encryption riding on that channel. You want to have your own tunnel, like an SSL connection or a VPN, that will protect you from any kind of snooping. Because otherwise you might as well be using

WEP, unencrypted WiFi.

Leo: Right. What is your sense of other technologies that are used right now? CDMA primarily?

Steve: I remember something similar about CDMA. I haven't looked at it closely for comparison. Like you, I'm curious now to see whether it's the same. But in this research I was just focused on GSM because I wanted to follow up on the news of what these guys had done. And it turns out what they - all they're really doing is they're taking six-year-old research from 2003, and they're saying, okay, the papers are published. Everyone's still ignoring this. Let's make some noise. Let's wake people up to this problem because someone ought to do that. And that's really, I mean, that's the goal of this group is not to foster piracy and hacking, but basically to challenge this GSM Alliance and say, folks, you've got to get your acts together here because this is not secure, and you're in denial.

Leo: You have your heads in the sand.

Steve: Yup.

Leo: How about data? We're talking about voice communications. Data goes over a different channel; right?

Steve: Well, data is using the same system. The GPRS is the packet radio technology. And it unfortunately uses all the same cipher and the same keys.

Leo: Oh, wow.

Steve: One of the things that you're able to do, one of the other attacks is interesting. There is a weaker version of the cipher. There's multiple versions of the stream cipher. The stream cipher is called A5. The authentication algorithm is known as A3. And the key agreement algorithm is A8. Well, this A5 stream cipher can - there are variations. There's A5/0, which says no encryption, just in the clear. There's A5/1, which was the original strong encryption, but it had export restrictions placed on it. So as a consequence, phones also support A5/2, which is a deliberately weakened, exportable encryption.

So get this, Leo, because this also bears on some of the things we've talked about in the past. Even though you may have a phone using the A5/1 strong encryption, it also supports A5/2. Because what if you happened to roam to a carrier that wasn't supporting strong encryption? Well, the phone would downgrade itself to A5/2. Well, it turns out there are active attacks which can be perpetrated where you ping somebody's phone and feign that you're only able to support the /2, the weak encryption, which is much easier to crack than the strong encryption. What we've been talking about is the strongest encryption available. And so you can essentially get the phone to downgrade itself, but A5/1 and /2 use the same keys. So you're able to get the phone to run a weaker cipher, which is much easier to crack, and then you're able to gain access to its key.

Leo: Wow.

Steve: So, I mean, it's very badly broken. It's absolutely not something that we could consider secure. It is far, far shy of state-of-the-art, the kind of state-of-the-art crypto that we're used to having in everything else we do.

Leo: You might have had a hint of that when they gave President Obama a special NSA-encrypted phone to use, that maybe perhaps the government knew there was, you know, some issue.

Steve: Yeah. And, well, they knew it because they have a closet full of equipment which is listening in on everyone's cellphone conversations.

Leo: Right. They can crack it, so we might assume the other guys can, too.

Steve: And it's worth mentioning, too, that all of this is only the in-the-air cipher. That is, if our government wanted to listen in on our phone calls - I guess we know that after 9/11 that was being done - it's much easier to just wait until the cell tower has performed all of the decryption and turned this back into analog signals and pick it up there. I mean, you could certainly do that.

The problem, of course, is, as with everything, we've talked about this in the context of WiFi many times, wireless is tempting because this stuff is in the air. And so there are, like I said, you park a van across the street from your competitor's office and suck in all of the cellphone conversations going on and see what you can glean. Who knows what you'll overhear? It's just it's not the case that it's as insecure as analog. But you absolutely should never depend upon its security, I mean, in any place where you've got super high valuable conversation and there's some reason to believe somebody else might love to know what you're talking about.

Leo: Especially if you're sending your bank key over your SMS uplink.

Steve: Yeah. And again, it's also worth mentioning that you could just use a big parabolic microphone, parabolic reflector and a microphone, and listen to somebody who's in visual range. You might not hear the other side of the conversation, but you would get theirs. So there are other sort of analog, real-world ways to do this.

Leo: Gosh, yes. And...

Steve: But it's certainly the case...

Leo: ...I presume that we're moving to newer technologies anyway over time. And really mostly, I mean, look, you're not going to redesign GSM and retrofit all the

towers and retrofit all the phones. That's not going to happen.

Steve: That's the problem. Now, 3G is a stronger technology. But the problem is the phones are all able to fall back to the earlier technology, and that provides a backdoor for the encryption. What you'd really want to do is be able to tell your phone, for example, no longer allow any weak encryption.

Leo: Oh, that's good.

Steve: Unfortunately, the phones are just open, and they're designed to roam and to work wherever they happen to find themselves.

Leo: When there's a secure phone, like the NSA-encrypted phone that the President uses, they probably use the same GSM or CDMA frequencies and channels and technologies, but they encrypt the data. They scramble it.

Steve: Yes, exactly. They're running an encrypted tunnel inside of the regular carrier.

Leo: Got it.

Steve: So if somebody decrypts that, all they're still going to get is highly encrypted...

Leo: Gibberish.

Steve: ...really, really pseudorandom noise. Just gibberish, yes. And they'll have no way to go any further. They're blocked by the tunnel that is running inside of the GSM channel.

Leo: But of course as with VPN or a scrambler technology, both ends have to support it. And that's why it's not generally used.

Steve: Well, not only do both ends have to support it, but again, once it comes out the other end, all of that encryption has been stripped off, and it's back to plaintext again.

Leo: Right, right.

Steve: So part of the mitigating aspect of this is, okay, so what's someone really going to do who wants to know what you're talking about? Maybe they're just going to be in the booth next to you with their ear cocked with, exactly, just overhearing your conversation in the old analog world.

Leo: Just listen. Steve, great, really an interesting subject. Fascinating. And of course ties in, if some of this stuff like rainbow tables leaves you scratching your head, we've covered all of the fundamental technologies in previous episodes.

Steve: Yup.

Leo: So you can go back, and I know that there are now 212 and this one, so 213 episodes. That's a lot of listening. But you can go back and look at rainbow tables. We talked about that. We talked about XORing in the past. We've talked about crypto in general. So you can really get a fundamental education on all this stuff from previous episodes.

Steve: Well, and we do have the transcripts at GRC, and a search for the transcripts. So you could put in "rainbow tables" or "XOR" into the search...

Leo: Exactly.

Steve: ...and quickly find those instances where we've talked about this stuff before.

Leo: Steve, as always, a pleasure. You'll find the transcripts, the 16KB versions of the show, the show notes and more at Steve's site, GRC.com. That's also where you'll find SpinRite, the absolute must-have, there is but one, hard drive maintenance utility, the one to get. And, by the way, recovery, too, as kind of a side effect of it. It does a great job. And all of his freebies, lots of security information and lots of programs like ShieldsUP! and Shoot The Messenger, DCOMbobulator and Wizmo, it's all at GRC, Gibson Research Corp., GRC.com.

And we'll be back - normally we record on Wednesdays. So if you want to watch us live at live.TWiT.tv, tune in at 2:00 p.m. Eastern time, 11:00 a.m. Pacific time, Wednesdays, 1800 UTC. And you can watch the show then. And then of course we offer it the next day, on Thursdays, iTunes and Zune and other downloads, as a podcast. So anybody can get it who has podcatching implements, including Listen on the Android phone. You can find out more about that at TWiT.tv/sn. All the protocols are there. Steve, thank you so much.

Steve: Next week we will do our 75th Q&A.

Leo: Wow.

Steve: So anyone who has questions, please by all means go to GRC.com/feedback and tell me what's on your mind, what you want to hear about, topics, suggestions, questions, and things that I've skipped over or forgot to mention, so forth. And we'll deal with them next week.

Leo: And this just in, Windows 7 updates were just pushed out, and one of our chatters is downloading them now. So that answers the question. Second Tuesdays for everybody now.

Steve: Thought that was the case. And we will talk next week about what happened in the world of Microsoft updates. I want to find out what that TCP/IP flaw is. That sounds like a bad one. So we'll have the news of that next week.

Leo: Thank you, Steve. Thank you all for joining us.

Steve: Thanks, Leo.

Leo: We'll see you next time on Security Now!.