The Perfect Accusation

Description: After covering this month's Patch Tuesday events and catching up with the past week's security news, Steve and Leo examine the week's most troubling and controversial revelations: the NSA's reported ability to crack much of the Internet's encrypted traffic. They explain how different the apparent reality is from the headlines, but why, also, this does form "The Perfect Accusation" to significantly strengthen all future cryptographic standards.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-421.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-421-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. And, yes, normally we would do a question-and-answer session on our odd-numbered episodes. But today we've got a very important story. We're going to talk about breaking news from last week about the NSA breaking all the encryption. Is it true? What does Steve say? We'll talk about it next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 421, recorded September 11th, 2013: The Perfect Accusation.

It's time for Security Now!, the show that covers you and your privacy and security online with the guy, the man, the myth, the legend, the one and only - he's holding up his spyglass - Steve Gibson of GRC.com, our Explainer in Chief. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again. I was looking at the number, 421. Now, of course that's a particularly special number for me because that's the first three digits of binary. So 4, 2, and 1 are the first...

Leo: Oh, wow. You've got a mind that works in mysterious ways.

Steve: I see patterns in things. But...

Leo: I would not have noticed that.

Steve: ...that's a lot of episodes. I mean, just...

Leo: That I noticed. Huh?

Steve: Yeah, it's like...

Leo: Okay. We have to explain because John Slanina, who's a very bright guy, says, "What?" So the binary numbers are one...

JOHN: And zero.

Leo: Yeah, but...

Steve: 1, 2, 4, 8...

Leo: 1, 2, 4...

Steve: ...16, 32, 64 and so forth.

Leo: So 101...

Steve: And so, like, I've been doing a lot of work back in assembly language with the work on SpinRite. And so I'm - and some of the screens that I'm putting up show hex dumps of registers. And so the hex dump will show 05. And so I have to decompose that into 101 to know which bits of the status the bus mastering controller has set and not. And so I'm - so, like, when I look at 421, what I see is, oh, the binary values of the first three bits of - yeah.

Leo: Right. The first bit is 1, 01. The second bit represents the twos. You get it now?

JOHN: He said it was the first three...

Leo: It's reversed. No, no. Okay, got it. He says you said it was the first three binary numbers, and obviously that's not - it's the decimal places on - it would be 010, a hundred thousand in decimal. It's 0, 1, 2, 4, 8, 16 in...

Steve: Right. And the order of them, in this case 4, 2, and 1, is the binary value when it's in a hex representation of each of those digit places.

Leo: And that's what - and the truth is I should recognize it because 421 are the bits that you set with chmod when you're in Linux, when you change file permissions.

Steve: Precisely, exactly. And so, yes, it's a 7.

Leo: It's a 7. Folks, if we haven't baffled you yet, you belong here. This is your show. You found the right show.

Steve: This was not a test to see whether you qualify for the balance of the podcast.

Leo: Yeah, but it - yeah.

Steve: This was just a little diversion.

Leo: [Sighing]

Steve: Now, we were supposed to do a Q&A.

Leo: Yes, we were. We're not, I see.

Steve: How could we? You know?

Leo: I know. It's just endless, isn't it.

Steve: Once again, on the day after the podcast, it was last Thursday...

Leo: We had you on the radio show to talk about it, it was such a big story.

Steve: And I came on with Tom on Tech News Today.

Leo: That's right, yeah.

Steve: In order to talk a little bit about it. Basically, to sort of calm people down. And what I want to do today, we're going to talk about the most recent round of revelations, which, as we've seen before, unfortunately, the press goes crazy.

Leo: Oh, yeah.

Steve: It's not the functions which are hyperbolic, it's the headlines. And so I titled this "The Perfect Accusation" because, as we're going to see, what we actually know is that, like, where - and I've read all these press articles that talk about the NSE - I'm sorry. The NSA has substantially, is substantially able to crack all of the Internet's cryptography and subvert it. And they've been working behind the scenes to weaken it. And there's a lot to be disturbed about. So I've got - so we'll talk about the news. But once again, the real meat here is where I want to focus the podcast because that's what this podcast is for. And in '07 our friend Bruce Schneier wrote a beautiful summary of the one, the only thing that anyone has been able to point to that suggests that maybe the NSA was involved behind the scenes in weakening something that no one cares about. I mean, it absolutely...

Leo: This is the NIST standard.

Steve: Yes. This is, well, this is one fourth of it that is the worst fourth, that is the - it's like the unloved stepchild random number generator. I've never heard of anyone using it. No one would ever use it. It's slower than, like, orders of magnitude, than the other good ones. And it's just - it's, like, bizarre. But the key is, it is perfect because, even if it were cracked, and it isn't, we don't even know that there was influence behind the scenes. But we really do, as an industry, need to be more vigilant. So this accusation could not be more perfect because probably the crypto industry wasn't asking enough questions, wasn't saying, hey, now, wait a minute, where did you come up with these magic numbers in here? And so something was allowed to happen.

And part of the reason it was allowed to happen is everyone knew, well, it doesn't matter. But it's perfect because it serves as an object lesson. The entire crypto industry is aware of this. All of our listeners will be. And everyone can at least breathe a sigh of relief because very much in the same way that I said I - the first day that we talked about the Snowden leaks, and I said, "I'm so glad," and the news was very fresh, and we weren't sure what it was going to come to, similarly, I am exactly so glad about this because this is a wakeup call that we needed. The fact that this happened at all says, okay, we were getting too lax. But it can never again be the case that anyone is accused of too much tinfoil because of essentially what did happen six years ago. Shouldn't have, but it doesn't matter. Anyway, we have...

Leo: Yeah, that's what the professor at Johns Hopkins said in his blog post that Johns Hopkins forced him to pull down.

Steve: Well, yeah. I'm going to talk about Matt. Matt Green is a neat cryptographer. And it's not actually that blog post but the one, I don't remember if it was after or before, but I'm going to share that in its entirety toward the end of this because, more than anyone else I've read, Matthew perfectly sort of lays out the terrain. And our listeners will end up coming away from this podcast saying, okay, I understand exactly what happened, and I feel comfortable with it. I mean, revelations were definitely part of this, with what we found out.

Leo: Oh, yeah. We do know new stuff.

Steve: These documents are creepy when you read them. But I also think, eh, they feel

to me like maybe this is how we get our budget pumped up a little bit. Some of what they're saying is like, eh, okay.

Leo: It's a little - they're overselling the case a little bit.

Steve: Exactly. I think a real close analysis would come to that conclusion.

Leo: Yeah. Steve Gibson, Leo Laporte, Security Now!. Let's get to security news.

Steve: Yeah, so again we are on this side of a Patch Tuesday. So I feel a little bit of an obligation just to note that to our users. When I fired up my Win7 machine that I use only for once a week, or sometimes a little more frequent Skype connections with you guys, I had five updates that it wanted to give me. I'm not sure actually why it was so few because there were 14 - oh, I know why, it's because I don't have...

Leo: You don't have those...

Steve: ...SharePoint Server...

Leo: There you go.

Steve: ...on this machine. So, yeah, so we are this side of Patch Tuesday. Microsoft and Adobe had treats for us. In Microsoft's case, four of their 14 patch bundles are rated critical, collectively fixing 47, at least 47 known security vulnerabilities. And I thought it was interesting. Microsoft is beginning to sense the friction against updates, especially when updates are messing things up. So, I mean, that really hurts them. So they're beginning to prioritize them. And Microsoft recommended, and this is my phraseology, for "reluctant enterprise-class upgraders," that they prioritize and install the Outlook, the IE, and the SharePoint Server fixes with a higher priority because those are a bigger problem. And at least one of the SharePoint vulnerabilities has been publicly disclosed, so it's ripe for exploitation. And in a corporate setting, you don't want anybody getting into your servers that way. So definitely worth doing that.

Adobe was a grand slam - Flash, Acrobat, Reader, Shockwave Player, and Adobe Air. Everything.

Leo: But Microsoft's updating those? Or that's the Adobe Updater?

Steve: Well, Adobe has updated them. Now, IE10 has auto-update now. And of course Google Chrome was the leader of the pack in auto-updating browser technology. OS X - I'm sorry, OS 10. I keep writing "X," and that's why I say it - OS X will block older versions now, so it's aware of that, and tell you you need to update. So it's really only Firefox and Opera users who still may need to deliberately go and get it. And as always, if you do go to the site and grab a download update pack, make sure you disable the default-enabled McAfee Security Scan if you don't wish to have that installed in your

machine. It's just amazing that they think they can get away with that. But I guess they get money from McAfee.

Yahoo! has joined some of the other very unhappy commercial providers named in the very early Edward Snowden links in suing the federal government and the FISA Court, essentially for the damages they're experiencing over loss of reputation. Yahoo! said on Monday that they've joined other U.S. technology giants in launching legal action against the federal government over the NSA surveillance revealed by whistleblower Edward Snowden. I'm reading from an article in the Guardian, which I thought was interesting because this article, I salute the Guardian, is being pretty rough on the Guardian. It said: "Yahoo! filed a suit in the Foreign Intelligence Surveillance (FISA) Court, which provides the legal framework for NSA surveillance," of course as we know, "to allow the company to make public the number" - so they're saying "to allow the company to make public the number of data requests it receives per year from the spy agency."

Now, again, that seems like a benign thing. They're not saying we want to say who. We just want to say how many. And I think they...

Leo: Google has asked for this, too.

Steve: Yes.

Leo: And I think Facebook, as well. And of course they've said no.

Steve: Yes, Google and Facebook. Google and Facebook are the other. And also Microsoft is in there, too. And one of the Yahoo! guys said: "Yahoo's inability to respond to news reports has harmed its reputation and has undermined its business, not only in the United States but worldwide. Yahoo! cannot respond to such reports with mere generalities." And also it says: "Criticizing news coverage, specifically by the Guardian and the Washington Post, Yahoo! said media outlets were mistaken in claiming that the PRISM program allowed the U.S. government to tap directly into the servers to collect information. It said that claim was 'false.'"

So we still have this cloud of we don't know the details. And that cloud may never be lifted. You'll remember, of course, everyone will remember my original theory was, if we take the denials of the reading that some looking at Snowden's slides gave, and take the position that the NSA has installed taps just upstream of the providers, that it very much solves the same problem, without actually requiring that these companies are complicit in this. So again, we don't know. But to me this is interesting because it is clearly the case that the domestic corporate interests are really being hurt by this notion that they're collaborating with the NSA and people who are upset by the idea that their privacy is being compromised. So another one in the pile.

Also OpenID, or myOpenID, which is a service run by Janrain for, wow, seven years, announced that they were going to be closing on February 1st of 2014. The CEO, Larry Drebes, explained. He said: "In '06, Janrain created myOpenID to fulfill our vision to make registration and login easier on the web for people. Since that time, social networks and email providers such as Facebook, Google, Twitter, LinkedIn, and Yahoo! have embraced open identity standards. And now, billions of people who have created accounts with these services can use their identities to easily register and log into sites across the web in the way myOpenID was intended.

"By '09 it had become obvious that the vast majority of consumers would prefer to utilize an existing identity from a recognized provider rather than create their own myOpenID account. As a result, our business focus changed to address this desire, and we introduced social login technology. While the technology is slightly different from where we were in '06, I'm confident that we are still delivering on our initial promise - that people should take control of their online identity and are empowered to carry those identities with them as they navigate the web.

"For those of you who still actively use myOpenID, I can understand your disappointment to hear this news and apologize if this causes you any inconvenience. To reduce this inconvenience, we are delaying the end of life of the service until February 1st, 2014 to give you time to begin using other identities on those sites where you use myOpenID today." And then he says: "Speaking on behalf of Janrain, I truly appreciate your past support of myOpenID." So we originally covered this.

Leo: Yeah, I used it, frankly.

Steve: When it happened. And, I mean, it had - I guess I would call this early first-generation, an early first-generation attempt. To give our listeners a quick review, you logged in, kind of perversely, with a URL. And it was a URL to a page that you controlled. And so that was the concept, the idea being there's always this notion of something you control. For example, using email to authenticate, you controlled your email account. In this case, the idea was you controlled a web page. And so you logged in with a URL to the web page, and then the site could go there to pick up your login credentials for authentication. And anyway, it just sort of - it was interesting. You could - sometimes you'd run across a website that would say, oh, log in with OpenID.

And so these guys were for people who didn't have their own website and web server and couldn't easily manage their own page. This was a service to provide that identity sort of as a third-party provider. So it was an interesting notion. And what he's talking about, of course, when he refers to social login, is what we have since been covering, which people see as log in with your Facebook account, log in with your Twitter account. And that's the OAUTH approach, where the site you're attempting to authenticate to bounces you over to a site where you are already known. You authenticate there, give permissions as required, and then behind the scenes that site authenticates you to the place you were trying to log in, and your browser is again bounced back to where you originally were, now authenticated. And then behind the scenes is the plumbing to make that secure. We did a podcast on OAUTH quite some time ago [SN-266].

At this point, Facebook accounts for 46% of OAUTH social login use. So it's the leader at 46%. Google is second at 34%. Yahoo! a somewhat distant third at 7. Twitter right behind it at 6. Then a whole bunch of other sort of there, too, also-rans collectively have about 6%, and Microsoft is less than 1%. So Facebook is the clear leader, with Google coming in in second place at 34. And that's where we are. And of course there's a lot of attention being given to login, and I teased everyone last week with the idea that I think I may have a really terrific solution that's better than all of that. So I will be working on that soon.

Many people, as a consequence of the NSA revelations, have tweeted me and said, hey, Steve, what about LastPass? You vetted it. You use it. Leo uses it. Everybody likes it. But where are they relative to the NSA? Responding to that yesterday, Joe posted a blog that I'm going to share with our listeners. He said: "With news that the United States National

Security Agency has deliberately inserted weakness into security products and attempted to modify NIST standards, questions have been raised about how these actions affect LastPass and our customers. We want to directly address whether LastPass has been or could be weakened, and whether our users' data remains secure.

"In short, we have not weakened our product or introduced a backdoor, and haven't been asked to do so. If we were forced by law to take these actions, we would fight it. If we were unable to successfully fight it, we would consider shutting down the service. We will not break our commitment to our customers. Although we are not currently in the position of having to consider closing the service, it is important to note that, if LastPass had to be shut down, our users would be able to export their data or continue using LastPass in 'offline' mode, although online login and syncing would no longer be possible.

"We have consistently reiterated that LastPass cannot share what we cannot access. Sensitive user data is encrypted and decrypted locally with a key that is never shared with LastPass. As always, we encourage our users to create a strong master password to better protect themselves from brute-force attacks. Given our technology and the lack of access to stored user data, it is more efficient for the NSA or others to try to circumnavigate LastPass and find other ways to obtain user information.

"Ultimately, when you use an online service, you are trusting the people behind that service to have your best interests at heart and to fight on your behalf. We have built a tradition of being open and honest with our community, and continue to put the security and privacy of our customers first. We will continue to monitor the situation and change course as needed, with updates to our community when necessary. Thank you to our community for your ongoing use and support of LastPass."

Leo: Now, this is the fundamental problem with any closed-source solution is, yes, they do everything right, but they could be coerced to put a backdoor in the closed source, the binary that you're required to use. Right?

Steve: Well, remember that they're running JavaScript on our browser.

Leo: Oh, all right. So it is open source, then.

Steve: It is open. No, it's open source. And they even - and, I mean, this is why I'm so happy with them is everything that you could do, they have done. It is truly TNO. Otherwise I would never have recommended it as I have. I wouldn't be using it myself. As I said, I don't know any of my passwords anymore. And no one needs to...

Leo: I love LastPass. But I'd be very sad to learn that it could have a backdoor. But it couldn't, you're saying.

Steve: And here's the point. Yes, it can't. Here's the point is that the NSA did go after - I'm blanking now, the email company, the email provider who Snowden used.

Leo: Yeah, yeah, yeah. I'm blanking, too. I've put it out of my mind.

Steve: Yeah. Anyway, we know who we're talking about. A couple weeks ago...

Leo: And he shut down, much to the dismay of the NSA. Lavabit. Thank you, chatroom. Lavabit.

Steve: Because he wasn't actually offering - yes, Lavabit. They did go after Lavabit because he wasn't actually secure. It was fake security. It was completely vulnerable to him receiving a national security letter, which he would have to comply with, and he would have to violate his customers' privacy. All LastPass is storing for us is a preencrypted blob. They do not have the encryption key. Only we do. What they get is a multi-iterated hash of the encryption key and our account name, our email address in this case, and password, hashed a bunch of times, which serves as an opaque token with which to identify who the blob belongs to.

So our browser says, here's a blob and this cryptographic token which means nothing. Save this for us. And they do that. And then when, over on our iPad, we log in, the iPad says, hey, are there any updates to this cryptographic token's blob? And the LastPass server says, oh, I do have a newer blob for you. Here's your new blob. And so that's how synchronization works. Only when the blob gets down to the client, which then has the email address and password, are those used, again, cryptographically multiple iteration hash, to generate a key which decrypts the blob there.

So there's - and this is the model for the future. This is the only way these things can work. That's what we've been talking about for years: Pre-Internet Encryption and TNO, Trust No One. There is no reason for the NSA to bother the LastPass folks because their system is secure. Now, the danger that exists is that they would be compelled to insert in the currently secure system some insecure technology. And we have to acknowledge the possibility. I mean, if I don't say it, I'm going to get flooded with tweets that will say, Gibson, this is possible. And it's like, yes, it's possible that - and unfortunately, one of the things that we know from this last week's revelations is the NSA is not happy that, like, LastPass exists and has done this in such a secure fashion. [The NSA] would love to have access to all of a given user's login passwords for their entire identity.

One of the things that the protocol I will be shortly proposing solves is all of this. There's no reliance on this kind of vulnerability, which is one of the things that makes it better. But what Joe is saying is he won't do that. He will, like Lavabit, just say, I'm sorry, we're pulling the plug. He won't be able to tell us why. He'll just say, "LastPass has decided we can no longer offer this service. Good luck." As opposed to breaking this trust. Now, then you wonder if him doing that puts him in violation of a letter saying you need to give us access because that's one of the other things that Lavabit said was that there was an implication in the email that his attorney was receiving that shutting down the email system was preventing the government from getting what they wanted from him. It's like, oh, lord.

Leo: And I'm sure the law allows them to compel him in some form. And it wouldn't be surprising if it compelled LastPass to do it secretly.

Steve: Yeah, I mean, so...

Leo: Maybe the solution is to use the open source KeePass. It's not as functional.

Steve: Yeah, and then not update it.

Leo: Right. Well, it's open source.

Steve: I mean, that's the problem, is the only - I guess what we could do is, if there was a way, for example, of freezing the LastPass code base so that it...

Leo: But if it's JavaScript there isn't because you're loading it every single time; right?

Steve: Exactly. You're getting an update from - and the LastPass plugin uses that in order to provide your form fill. So anyway, we're in an interim awkward place right now, and I hope we can move past that quickly.

Leo: Boy, it'd be really a big compromise to get LastPass because that would - then they'd have everything.

Steve: Yeah. So Jenny and I saw "Riddick." And I tweeted "FWIW," which is For What It's Worth, "Riddick was pretty much awful." And I tweeted, "If you're sure you'll like it, then go with that. But it was much less good than the two previous."

Leo: Oh, you liked the other ones.

Steve: Oh, "Pitch Black" was great. I thought that was innovative and new and fun and interesting. And then the second one I thought was really fun, too. This was just kind of a cartoon. It's like, eh, okay. So I just - and many people thanked me for the warning. I mean, it'll be on disk next Tuesday, so - I'm kidding, but, I mean, soon it'll be out. And spend $3 to get it on Apple TV or an Amazon download or something. Don't spend $15 in the theater unless you really...

Leo: I can't tell you how few sequels are any good. Think about "The Matrix."

Steve: Yeah.

Leo: "Star Wars." Though people are going to yell at me for that one. But...

Steve: So, and in response to this, David Busch tweeted from - his handle is @HappySlice. He said: "@SGgrc Riddick movies were always campy." Can't argue that. He said: "I watched the first three episodes of 'Orphan Black' yesterday. Fantastic show."

Leo: Oh, wasn't that good. Yeah, thank you for that recommendation, by the way.

Steve: Yes. And so the reason I brought this up is that I've had a lot of feedback from people. I haven't even poked into it yet. But Jenny also watched it and loves it. And Elaine, who read the, well, created the transcript last week, noted that Season 1, Episode 1 - remember I said that Paul had mentioned in The New York Times that he believed it was being re-aired. It re-airs on BBC America starting September 14th.

Leo: As they prepare for the new season.

Steve: Yeah. So three days from now. So for anyone who wants, it was 13 episodes. Was it 13 or 10? Can't remember now. But anyway, so people are really liking it. So that's good news.

Leo: Good.

Steve: Okay. So I did note, I watched Apple announce the new iPhone 5S. And like everybody, it's like, okay, 64-bit processor, that sounds interesting, motion, or the M7 motion...

Leo: Let me ask you, though, before you go jump on, okay, to the next thing, 64-bit processor. It's hard to get the straight information on this. You're a programmer. You would understand the value or the non-value of 64 bits. It's my understanding, certainly on a desktop, the clear advantage is you could address more memory, so you can break the 4GB RAM barrier. This is not a problem on phones. They don't say how much memory the iPhone has. But I guarantee it's not 4GB. It's much more likely a gig or two. There's one phone that has 3GB. Nothing has more than that.

Steve: The only - the place where 64-bit matters is where you're dealing with math of any kind that can't fit in 32 bits. So the way a 32-bit processor handles it is in pieces. So you add the two lower 32 bits, and maybe there's a carry from that. So you then add with the carry the upper 32 bits. So in terms of performance, and that also...

Leo: For giant numbers, though, only; right?

Steve: Yes, yeah, exactly. For, well, so crypto is giant numbers.

Leo: Ah.

Steve: Graphics oftentimes uses giant numbers. So it's just, I mean, my sense, you know, here I am programming in assembler. I look at...

Leo: Well, that's why I ask, because you know register sizes. Most programmers aren't even aware of this. And that's what 64-bit means; right? It means it has registers of, instead of 8 or 16 or 32...

Steve: They are double the length. They are...

Leo: They're 64-bit registers.

Steve: Yeah. And so...

Leo: A lot of what you do does not require 64 bits.

Steve: Right, like character processing and so forth.

Leo: Right, that's 8 bits.

Steve: Eight bits, yeah.

Leo: Maybe 16, if you're using...

Steve: Right.

Leo: So my question is, is it a marketing term? Or is there some real value to be gained on a mobile platform with 64 bits?

Steve: Oh, I see. I...

Leo: That's a value judgment. I mean, I'm only asking you what are the technical advantages of 64 bits. And you can access more memory, and you can work on giant chunks of - giant numbers.

Steve: Correct.

Leo: I guess if you're doing - moving data, you can move bigger chunks at a time, which is nice. That's why gaming would benefit.

Steve: Yeah. Although, for example, what's happened with our processors is the processor speed has completely outstripped RAM speed. RAM is stuck because of its physics to being relatively slow. So we have a cache on the chip that reads in so-called

"lines" of RAM. It reads chunks of memory at a time because the notion is processors tend to stay sort of within their neighborhood as they're executing code. So doubling the register size doesn't mean that the cache was doubled or that the cache line size was doubled. It probably wasn't. That's probably...

Leo: No, I'm sure it wasn't, yeah.

Steve: Those things are probably all the same. So I agree. I think maybe it's a little bit of specsmanship, more than anything else.

Leo: ExtremeTech, Joel Hruska says on ExtremeTech it's marketing fluff. But I think that is a somewhat subjective judgment. And the other thing I don't fully understand is the difference that ARM might - because we're using a RISC architecture with ARM, there may be a difference. That's different - I'm thinking, 64-bit generally for me is on the Intel CISC stuff. That I'm a little more familiar with. I'm not sure if it changes things in the ARM architecture, the v8.

Steve: Yeah. When we were talking about how processors work, we did our whole processor technology series years ago, one of the things that we made clear was that the reason the complex instruction set, the Intel-style instruction set is inherently difficult to run at low power is you've got all of this real estate taken up with very - with instructions which have very low utilization. You have to have them because you want to be backward compatible to all the way back to an 8088 chip, which Intel to their credit is. But, boy, they're dragging that legacy forward with every single generation. They still have to have all of that old gunk that you just have to know they would love to be able to flush. Whereas the ARM people were able to..

Leo: To start from scratch, yeah...

Steve: ...start from scratch, yeah.

Leo: They do, on the ARM page, talking about their x8 and 64-bit, they do say - or v8, I should say - they do say that - they do mention cryptography specifically. And that does make sense, if you're dealing with giant primes.

Steve: Yeah. I would say look at the phone we have now and what it's able to do with 32 bits. It's like, that seems fine to me. I mean, everything scrolls smoothly. Nothing is jerky and hesitating. I mean, clearly you could engineer the phone around 32 bits if that was what you needed. I just think we're going to see RAM get larger and ROM get larger. And the problem is, with 64 bits, it just in general is more hungry. Even though they apparently have really got power consumption down, too, on this thing.

Leo: Yeah. Oh, I'm sorry, I didn't mean to interrupt, but I knew of all the people I could ask about 64-bit, I thought you'd be the best on that.

Steve: So fingerprints...

Leo: The fingerprints, yeah.

Steve: ...is interesting to me. I think it's 100% cool. I love that it's round because that means it's orientation independent. I sort of wish that it was a linear, drag your finger over it approach because then it's less easy to spoof it, I've always felt. If you have to move your fingerprint over it, you can get a lot of linear resolution and a lot of temporal resolution as you drag your finger across it. But then you certainly don't get rotation neutrality. And it's clearly nice that you can put your finger down on any 360-degree orientation, and it spins it around and figures out what it's seeing. And one of the things that I've been noticing in the buzz about this is people worrying about the security of it.

Well, what I would remind people is it is not its goal, like a fingerprint that you leave behind at a crime scene, to be able to identify you from the population of the world, where we see on all of the crazy TV shows where the fingerprint comes up, and the computer's going [vocalizing], like finding features, and then you see all these faces flashing by, and we find the person that that one fingerprint belongs to. None of that is happening here. All that's happening here is that we've trained the phone to recognize one or a very few, because you could have multiple people trained, fingerprints out of everybody else. So that's a very different problem. That's the problem of here's five fingerprints, or one, typically, if you just have just your phone, here's one fingerprint that we've seen over and over and over and over and over and over and over. Now comes a new one. Is it the same? That's the question we're asking. Is it the same?

And so the beauty of this, if they've done it well, is that every time you use it, it is refining its knowledge. Notice that you may actually put your fingerprint down in a slightly different position. So hopefully that gives it knowledge that it didn't have of a new region of your finger, whereas it has enough of the old, of the region it's seen before, to say, ah, same guy. Look, it's moved over 17 pixels. But we've got some more information here over on the right-hand side. So it expands the map of the finger. I mean, I trust them to have thought about this and to have done it right.

And so it's exciting to think how well this could work because the question it has to answer is same fingerprint or not? And that's why, potentially, it can do a very good job of, eh, this doesn't look the same. And it's going to be, no doubt, somewhere in the world are a collection of other people whose actual fingerprint looks enough like yours that it would say, oh, he's back, when in fact he's not. But that's the same problem that you have - this is actually a weak analogy, but we've all talked about how door keys are not unique. There are not enough combinations for door keys for them to be unique. There are other people in the world whose door keys fit our doors, the front doors of our homes. But they don't know that, and we don't know that. So it's good enough. But it's definitely the case that somebody who you actually encounter at Starbucks who picks up your phone, or someone who steals your phone, their particular fingerprint, none of their 10...

Leo: Highly unlikely, yeah.

Steve: ...or anyone they know are going to fool this thing. Yeah. So anyway...

Leo: Apple says that they're saving the fingerprint as a hash, and encrypted, to boot.

Steve: Perfect. That's exactly what they should do.

Leo: So it would be useless even if the NSA were to get the database.

Steve: Yes. It would be absolutely possible to run the recognizer, determine a bunch of characteristics after derotating it and recentering it and so forth, and then you take those characteristics, and you hash them so that you get, unfortunately we'll reuse the term, you get a fingerprint of the fingerprint. Or a signature of the fingerprint is a better way to put it. And but you can't go backwards and figure out what those features were that generated that signature. And I believe, see, that's one of the other really nice things about what Snowden has done for us, is it's the level of scrutiny was already high. Now it's neurotic.

Leo: Yeah. Everybody's very paranoid, yeah.

Steve: It's good. Thank you.

Leo: Not bad.

Steve: Get some, you know, tinfoil sales is up, and I think that's just fine.

Leo: And I have to point out, and perhaps people forget this, I don't know how many DMVs do it, but certainly in California, they fingerprint you when you get your license. So California has built a massive database of every driver in the state, of their fingerprints. So the NSA really doesn't have to work that hard.

Steve: And probably not encrypted.

Leo: Not encrypted. I'm sure shared with the NSA. That's the reason they collect them.

Steve: Yup. They're actually recording your entire fingerprint image.

Leo: Right, right.

Steve: And of course that is why famously we've said, when you go to Disneyland, and they want to use your fingerprint for access, give them your knuckle instead.

Leo: Right. And then it's just a matter of time before they take a little bit of hair, and then they get your DNA. And, you know, you can't knock it because every crime is solvable then. All you need is a fingerprint or a strand of hair. You could say you were there. You were on the scene.

Steve: Oh, just put gloves on like Dexter, and then you're fine. Okay. So, shoot, there was one more thing. Oh, I did want to say that this scanner is subject to spoofing.

Leo: Ah. Now, this is a good question because there were scanners that would look for an infrared heat signature, and then that would be - you'd have to be a live person. There are some scanners you could use a Play-Doh thumb.

Steve: Now, there's a lot we don't know. You can, for example, you can get someone's pulse from their thumb. And so maybe Apple's doing that. I mean, not a lot - there's a lot we don't know yet. But as I understand it, it's a capacitive sensor which uses the ridge, the difference in capacitance...

Leo: Oh, that's interesting.

Steve: ...between the ridges that are closer and the grooves that are further away. And so it needs to be 3D. But somebody, I guarantee you, they will take - they'll do an experiment. They'll take a thumbprint off of a wineglass, lift it off or just dust it and then photograph it. Then they'll use a 3D printer to make a 3D image of that thumb, and it will unlock the phone. And it's like, okay. So proof of concept, that's interesting. We're going to see that. And then, yeah, you could - so technically, if you've got somebody's fingerprint - I mean, if this works. We don't know for sure that it works. But if I had a lot of spare time on my hands, I would try it. Maybe one of our listeners who gets one of these new phones will be industrious and give it a shot. Somebody's going to post it up on YouTube, and we'll certainly carry it when we find out about it. But maybe Apple's done something to defeat that, where it's got to be a live thumb.

Leo: I'd be curious. Here's the thing that I thought was most interesting. If it's just to unlock the phone, big deal. Then none of this is that important. It's just unlocking the phone.

Steve: Oh, Leo. I'm so - how long are you going to be gone?

Leo: Three weeks.

Steve: Okay. Maybe I can wait.

Leo: Don't wait, don't wait, no, no, no, no, no.

Steve: I really - because one of the things that my solution needs is endpoint security because...

Leo: Authentication is so vital for all this stuff.

Steve: Yes. And so the power is that it's completely anonymous, and but the ease of use is that you would like it not to put you, not to have to have you remember a big long password. So I'm loving that there's now this fingerprint scanner. Doubtless they will export this to an API so that apps can say please put your thumb on the Home key in order so that we know you're here. Well, an implementation...

Leo: That would be so awesome.

Steve: ...of my code for my approach on the iPhone, absolutely will want to do that.

Leo: And the thing I thought was telling is that Apple trusts it enough to use it not just for authentication on the phone, but for eCommerce. And they're putting their - that's putting their money where their finger is because that means they're saying you can buy stuff on the app store just with your fingerprint, not with your password anymore.

Steve: As Sarah said, she apparently has so many apps that she loves the idea. She can say, oh, I want this, and then...

Leo: It's very frustrating. Every time you update an iPad or an iPhone, you have to enter in your password. It's extremely frustrating.

Steve: And if it's a good password, then it upgrades the frustration level.

Leo: So this is, I think, encouraging. I'm not going to run out and buy one until I see that API and third parties adopting it. But that could make the iPhone 5S a must-have until others do the same thing. And it may well be that Apple, because they bought AuthenTec, has the rights to do this that others don't.

Steve: Yeah. I'm up for plan upgrade on my Apple track. I've got a Blackberry and an iPhone. So I'll be there.

Leo: Yeah.

Steve: Okay. So this is really cool, Leo. Net Neutrality is a hard concept to explain. If you don't know about this, make a note, Leo, to check this out. The URL is TheInternetMustGo.com. I tweeted it yesterday. And what I said was "Terrific video that finally explains Net Neutrality. Everyone you share this with will finally get it." Anyway,

it's a spoof of a guy who's hired by the ISPs, essentially to, like, sell why Net Neutrality is a bad idea.

Leo: [Laughing] It says at the beginning, "This is for internal marketing purposes only."

Steve: Yes.

Leo: I love it.

Steve: Anyway, it's really good. I commend our entire audience, TheInternetMustGo.com. Watch the video. It's also on YouTube, so you can just watch it on YouTube if you're interested. And, I mean, he covers the bases. He, like, goes to privacy advocacy groups. And, like, there's a bunch of hippie-like people around the table, and they're, like, looking at him like he's lost his mind as he tries to tell them why it's all good that you'll be able to pay extra, $5 to get these and $5 to get this plan. And someone says, well, that's cable TV. We don't want that. And he goes, oh, yeah, you do. So anyway, highly recommended. TheInternetMustGo.com.

Leo: Love it. I'm sharing it on Facebook right now.

Steve: Good. It's, I mean, because, again, it's a difficult concept to portray. And I found out about this from the EFF that is promoting this video. So they're behind this, too.

A little quick update on SpinRite. We have nailed the high-speed technology. We're generating benchmarks using the new low-level, all in assembler code, which is matching the manufacturer's absolute maximum sustained data throughput rates that they say their own drives can do. So there was one that was reasonably old, I think it was 78MB/s on the outer diameter of the disk Seagate said this particular drive can do. And we're measuring it at 77.4 and actually achieving the absolutely maximum that the drive is capable of. So that's what SpinRite will be doing. And we've got that technology nailed. Everybody, like about a hundred people are playing with that.

We've also uncovered, as I knew we would, some weird boundary cases. There's an older OCZ Vertex 2 SSD that turns out not to be ATA spec compliant. We were telling it - it says it's able to handle transfers of 65536 sectors at a time, but it doesn't, even though everybody else's drives do. We've run across a couple add-on controllers that misbehave and revector the hardware interrupt controller and sort of take it over. So I'm now in the last few days here of, like, dealing with the fringe cases. I've come up with solutions for all of that.

So anyway, it's going really well. And of course I will remind everybody that anyone who has SpinRite now, SpinRite 6, will be able to get a free upgrade to this hot new version as soon as I have it ready. And actually we'll make it available probably before it's officially released for people who want to beta test.

We are getting new people joining the process. This is - it's so fun to work in this mode because GRC has a newsgroup server, an old-style newsgroup, NNTP, at news.grc.com. But that's not a web browser. You can't put news.grc.com into your web browser. You

need an NNTP client. Thunderbird is one. Outlook actually even has news capability. I use Gravity, which is a nice free one that's been around forever on Windows. And the Mac's got a news reader. iOS has one called NewsTap that I use.

But it's really interesting because, when we're in this mode, when I'm, like, there's people in the newsgroup, we're working on stuff, somebody will have a problem. I'll say, oh, and I'll be back in two minutes with another attempt, and they'll run it, and it fixes the problem. And people are just not used to, like, interactive software development, where it's a cycle of, like, a few minutes, and something is resolved. And some guys will, like, come back after being gone for three days, and they'll go, oh, my god, I can't believe what's happened in the three days. And it's because...

Leo: How fun is that for you? That must be great.

Steve: Oh, it's just incredible. I mean, I'm dead by the end of the day. But I can cycle so fast that way that people are able to say, okay, this didn't work, or it hung. And I'll go, okay, hold on a second, and I'll put some code in where it apparently hung. That person will run it, it'll spit out some diagnostics, I'll go, oh...

Leo: This is such a...

Steve: ...shoot, okay. And then I'll say, I'll bet that your BIOS is, like, leaving interrupts turned off.

Leo: Awesome. Awesome.

Steve: And so I just add - I turn interrupts back on, problem solved, okay, move on to the next thing. And so it just allows us to move so fast.

Leo: You should write this up because very few people are doing development in this - or have the luxury of doing development in this fashion. You have a devoted group of people who are literate and smart, and you're interacting with your beta testers in real time.

Steve: Yes. It is real-time development.

Leo: That's awesome.

Steve: And it's just incredibly powerful.

Leo: Write this up because this is something that other developers should pay attention to. And yet again, another really important point proving the value of community. Having an interactive real-time community as we do with the chatroom

and so forth, so valuable.

Steve: Yeah. Well, exactly. You see exactly that.

Leo: Yeah. I'm doing interactive, iterative broadcasting.

Steve: Yup.

Leo: Yeah. It's really cool. That's really cool. All right, Steverino. It's time to talk about a perfect accusation.

Steve: One thing I forgot to wrap up with is people have asked how they participate in that process I was just describing.

Leo: Oh, oh, of course, yeah. How can we do that?

Steve: GRC.com/discussions will take you to our page. Or up under Services, I think, on the main menu is Discussions. And so that's the page that lays out how to participate - the domain name of the news server, news server configuration in order to get there. We have a test group, GRC.test, where people can attempt to do posts. It has a five-day expiration, so we generally host little random dialogues there. Everything is saved forever in the newsgroups. And so I'm operating in GRC.SpinRite.dev, as in short for "development." That's where I am. But there's a SpinRite group. There's a Security Now! newsgroup where there's the discussion of the podcast topics, sort of like for people who want to go much deeper. There's all kinds of stuff going on. I mean, it's a little-known bastion of no - it's like, there's no flaming, there's no spam, it's just serious people who are interested in this topic. And so, if you're interested in participating, we'd love to have, you know, the more the merrier. It ends up being fun for everybody.

Leo: Oh, I'm going to learn Turkish. I found the book I want.

Steve: Okay. So the perfect accusation. One of the things that annoys me, our listeners know, is when the headlines are clearly designed to attract readers. Or as I said, when not only the algorithms are hyperbolic, but the headlines are. CNBC had a headline, "Internet Experts Want Security Revamp After NSA Revelations."

Leo: Okay.

Steve: And it's like, what? And then, so it starts off - I won't bother everybody with the whole thing. But "Internet security experts are calling for a campaign to rewrite web security" - what? No, they're not - "in the wake of disclosures that the U.S. National Security Agency has developed the capability to break encryption protecting millions of sites." Okay. There are no such disclosures. No one said that. I mean, it's just like, okay.

Leo: Just make up the news. You'll get more hits that way.

Steve: "But they acknowledged the task won't be easy, in part because Internet security has relied heavily on brilliant government scientists who now appear suspect to many."

Leo: Oh.

Steve: It's like, oh, gosh, just shoot me now.

Leo: Where is that from?

Steve: CNBC.

Leo: Oh, well, no wonder.

Steve: I know.

Leo: Terrible.

Steve: Anyway...

Leo: This is what happens when you have people who don't understand technology, which is most people, writing about highly technical subjects and trying to get links.

Steve: Now, unfortunately, The New York Times did, I mean, it's as if they took what scant information they have and read it the worst possible way, knowing nothing about the subject. So, yes, it creates an interesting, inflammatory story which, as we were just saying, generates hits. But, boy, I mean, it leaves the wrong impression. At the same time, the raw data is also deeply disturbing because The New York Times story linked to secret documents which were - this was this next level of rollout. And in the actual secret document release - and now I'm reading from the source material - there's weird acronym stuff that the intelligence community uses, TS/SI/NF, whatever that stands for. Then we know that SIGINT is Signals Intelligence.

So this says: "The SIGINT Enabling Project actively engages the U.S. and foreign IT industries to covertly influence and/or overtly leverage their commercial products' designs. These design changes make the systems in question exploitable through SIGINT collection, e.g., Endpoint, Midpoint, et cetera, with foreknowledge of the modification. To the consumer and other adversaries, however, the systems' security remains intact. In this way, the SIGINT Enabling approach uses commercial technology and insight to manage the increasing cost and technical challenges of discovering and successfully exploiting systems of interest within the ever more integrated and security focused global communications environment."

Okay. So that's political gobbledygook, but it also says - or, in fact, now, off to the side, The New York Times has summarized this, saying: "The NSA's SIGINT Enabling Project is a $250 million-a-year program that works with Internet companies to weaken privacy by inserting backdoors into encryption products." Now it's like, okay. But no examples, no names, no companies, no, like, we found one of these. No one's ever found one. But that's what we're saying. "This excerpt from a 2013 budget proposal outlines some methods the agency uses to undermine encryption used by the public."

So again, what I just read is what The New York Times summarizes that way. And so it's true that what I read is unsettling. But they're trying to get money. And that's one of the things I would like to remind people is they talk about things coming online, or we're making progress on this. So we need a new facility in Utah with lots of water to cool our supercomputers, and then we're going to rub our hands together, and magic is going to happen. So, again, this is - there are no specifics anywhere.

Also, same document, different topic: "Basic resources in this project are used to" - and there's two that are extra worrisome - "insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communication devices used by targets." So they're saying that. But, again, no specifics. No other information. Just that that's what they say they're doing. And then the other one is "Influence policies, standards, and specification for commercial public key technologies." So it's interesting that they say that because that's what sort of - that raised the alarm among the crypto community that remembered a controversy from seven years ago where it's like, oh, I kind of remember something about that. And we're going to talk about that next.

And then also here, two more bullets is "Reach full operating capability for SIGINT access to a major Internet peer-to-peer voice and text communications system," and that's been suggested maybe to be Skype, that there was that reengineering of Skype that we know was specifically done so that they could - someone could respond to wiretap requests. And then also "Complete enabling for" - and then it's blanked out here, redacted - "encryption chips used in Virtual Private Network and web encryption devices." So it's like, again, no specifics, just strong and worrisome intent.

Okay. So, now let's look at someone who understands this exactly to sort of get our bearings. And this is Bruce Schneier, who back in '07 - so I'm going to read what he wrote. But he's talking about this just happened, this is new. So remember that this is then. This is seven years ago, six years ago that he's talking about. And so it's history. So he says: "Random numbers are critical for cryptography, for encryption keys, random authentication challenges, initialization vectors, nonces, key-agreement schemes, generating prime numbers and so on." And those are, of course, things we've talked about on this podcast often. We all understand we've got to have a source of really good random numbers and what happens when you don't.

Continuing, Bruce says: "Break the random number generator, and most of the time you break the entire security system. Which is why you should worry about a new random number standard" - remember, new in '07 - "that includes an algorithm that is slow, badly designed, and just might contain a backdoor for the National Security Agency." So what I'm reading is the only thing anyone knows about. And all of this has sort of bubbled up from this, what I'm reading.

"Generating random numbers isn't easy, and researchers have discovered lots of problems and attacks over the years. A recent paper found a flaw in the Windows 2000 random number generator. Another paper found flaws in the Linux random number generator. Back in '96, an early version of SSL was broken because of flaws in its random number generator. With John Kelsey and Niels Ferguson in 1999, I co-authored," says

Bruce, "Yarrow, a random number generator based on our own cryptanalysis work. I improved this design four years later and renamed it Fortuna in the book "Practical Cryptography," which I co-authored with Ferguson." I'm afraid we've got a weed whacker going on outside.

Leo: That's fine. It's not loud. But it's good to note it so that people don't think it's their...

Steve: Don't think it's, yes, their earphones are falling out. "The U.S. government released a" - so here it comes. "The U.S. government released a new" - back in '07 again - "official standard for random number generators this year, and it will likely be followed by software and hardware developers around the world. Called NIST Special Publication 800-90, the [130]-page document contains four different approved techniques called DRBGs, 'Deterministic Random Bit Generators.' All four are based on existing cryptographic primitives. One is based on hash functions, one on HMAC, one on block ciphers, and one on elliptic curves. It's smart cryptographic design to use only a few well-trusted cryptographic primitives, so building a random number generator out of existing parts is a good thing.

"But one of those generators, the one based on elliptic curves, is not like the others. Called Dual_EC_DRBG" - that stands for Dual Elliptic Curve and then Deterministic Random Bit Generator - "not only is it a mouthful to say, it's also three orders of magnitude slower than its peers."

Leo: Oh, that's good. That's an improvement.

Steve: Yeah, so it's like, oh, let's hurry up and use that one.

Leo: That's a thousand times slower.

Steve: Yes, yes. Because the other ones are - they're hash functions, or they're a symmetric cipher that you run a counter through.

Leo: That's terrible.

Steve: Yeah. "It's in the standard only because it's been championed by the NSA, which first proposed it years ago in a related standardization project at the American National Standards Institute (ANSI). The NSA has always been intimately involved in U.S. cryptography standards. It is, after all, expert in making and breaking secret codes. So the agency's participation in the NIST" - the NIST is the U.S. Commerce Department's National Institute of Standards and Technology - "standard is not sinister in itself. It's only when you look under the hood at the NSA's contribution that questions arise."

Now, I should stop for one second, just to remind people that, for example, IBM many years ago developed DES, the Data Encryption Standard. And IBM proposed it as a standard. And it was a technology where there are these things called S-Boxes. An S-Box

is a sort of a - it's a pattern box. You put in a byte, and a different byte comes out. So there's a mapping inside between incoming and outgoing bytes. And DES has a bunch of these, which the cryptographers at IBM specified and said this is really good. The NSA changed the design of the S-Boxes and never said why, but they did that. They just said, uh, this is better. And that then got standardized.

Now, we know that DES is weak, but it wasn't as a consequence of that. It was because the key length was 56 bits, which is no longer secure. Thus 3DES uses three different keys, each of that length, and does the DES thing three times. Much later, when our academic understanding of cryptography improved, people looked at what the NSA had done when we were at a position to understand it, and they had fixed it. If the original design had been left alone, DES was already broken. It was bad. And so without saying anything, without giving away their secrets, the NSA said, uh, change it like this. Just trust us.

And it turns out they fixed it. They, the NSA secretly, without telling us why, fixed the broken crypto that was used universally. DES was the banking crypto that was universally used for a long time, until we got up to modern times. So certainly there are cryptographers and mathematicians and a lot of smart people at the NSA. It would be wrong to assume that their only goal is to crack Internet crypto. They are equally responsible for helping us to have strong crypto so that foreign governments and terrorists and bad guys are unable to crack the crypto. So I just wanted to insert that here.

Continuing with Bruce, remember, because when you look under the hood, he was saying, this Dual_EC_DRBG is in the standard because of the NSA: "Problems with Dual_EC_DRBG were first described" - okay, now, he's writing this in '07 - "first described in early 2006. The math is complicated, but the general point is that the random numbers it produces have a small bias. The problem isn't large enough to make the algorithm unusable, and Appendix E of the NIST standard describes an optional workaround to avoid the issue, but it's cause for concern. Cryptographers," Bruce writes, "are a conservative bunch. We don't like to use algorithms that have even a whiff of a problem.

"But today there's an even bigger stink brewing around Dual_EC_DRBG" - "today" meaning '07 still, remember. So there were problems. Two papers were published that raised some concerns. Then in '07, so a year later: "In an informal presentation at the CRYPTO 2007 conference in August, Dan Shumow and Niels Ferguson showed that the algorithm contains a weakness that can only be described as a backdoor. This is how it works: There are a bunch of constants - fixed numbers - in the standard used to define the algorithm's elliptic curve. These constants are listed in Appendix A of the NIST publication, but nowhere is it explained where they came from.

"What Shumow and Ferguson showed is that these numbers have a relationship with a second secret set of numbers that can act as a kind of skeleton key. If you know the secret numbers, you could predict the output of the random number generator after collecting just 32 bytes of its output. To put that in real terms, you only need to monitor one TLS Internet connection" - now remember, if it was encrypted using this pseudorandom number generator that nobody has ever used ever. But if it were, then you'd "only need to monitor one TLS Internet connection in order to crack the security of that protocol. If you know the secret numbers, you can completely break any instantiation of Dual_EC_DRBG.

"The researchers don't know what the secret numbers are; but, because of the way the algorithm works, the person who produced those constants might know. He had the

mathematical opportunity to produce the constants and the secret numbers in tandem." Now, think public/private key. It's very much like that. We understand that you produce a public key and a private key together, and the point is one doesn't expose the other. Yet you need to use the other to undo the effect of the one. So the analogy isn't perfect.

But imagine that we're using, essentially, a public key in the form of these constants, which actually are just - an elliptic curve is a parametric curve. It's a curve described by y cubed equals x cubed plus x3, blah blah blah, that kind of thing, algebraic curve, where the specific instance of the curve matters. So these numbers describe a single curve which is in the standard. And so we could think of it like the public key. Maybe what these researchers discovered is it's theoretically possible that there could be the equivalent of a private key matching those numbers which are in the standard, which are public, which would completely break the random number generator. So it's, again, theory.

Now, I've looked at the standard. I've also looked at the presentation that these guys gave. And they specifically say in their conclusion, Slide No. 8 of 9, under "Conclusion," it says, all caps: "WHAT WE ARE NOT SAYING: NIST intentionally put a backdoor in this PRNG. WHAT WE ARE SAYING: The prediction resistance of this PRNG as presented in [the standard] is dependent on solving one instance of the elliptic curve discrete log problem. And we do not know if the algorithm designer knew this beforehand." So what they're being very careful in saying is there's a theoretical weakness we've discovered. Maybe it's news to everybody, including the NSA. Maybe it's not. And so we're wondering more now about the behind-the-scenes politics of this, sort of, today.

So continuing with Bruce's post, he said: "Of course, we have no way of knowing," Bruce writes, "whether the NSA knows the secret numbers that break Dual_EC_DRBG. We have no way of knowing whether an NSA employee working on his own came up with the constants and has the secret numbers. And we don't know if someone from NIST, or someone in the ANSI working group, has them. Maybe nobody does." Maybe they don't exist. "We don't know where the constants came from in the first place. We only know that whoever came up with them could have the key to this [theoretical] backdoor. And we know there's no way for NIST - or anyone else - to prove otherwise. This is scary stuff, indeed," writes Bruce.

"Even if no one knows the secret numbers, the fact that the backdoor is present [theoretically] makes Dual_EC_DRBG very fragile. If someone were to solve just one instance of the algorithm's elliptic curve problem, he would effectively have the keys to the kingdom." Remember, only if people ever used this. But, "He could then use it for whatever nefarious purpose he wanted, or he could publish his result and render every implementation of the random number generator completely insecure.

"It's possible to implement Dual_EC_DRBG in such a way as to protect it against this backdoor, by generating new constants with another secure random number generator and then publishing the seed. This" - now, get this. "This method is even in the NIST document, in Appendix A. But the procedure is optional, and my guess is that most implementations" - if there are any - "of the Dual_EC_DRBG won't bother.

"If this story leaves you confused, join the club," he says back in '07. "I don't understand why the NSA was so insistent about including Dual_EC_DRBG in the standard. It makes no sense as a trap door: It's public and rather obvious. It makes no sense from an engineering perspective: It's too slow for anyone to willingly use it. And it makes no sense from a backwards-compatibility perspective: Swapping one random number generator for another is easy.

"My recommendation," says Bruce, concluding, "if you're in need of a random number generator, is not to use Dual_EC_DRBG under any circumstances. If you have to use something in [this standard] SP 800-90, use [the counter] CTR_DRBG or [the hash] Hash_DRBG. In the meantime, both NIST and the NSA have some 'splaining to do," says Lucy, or says Ricky.

Leo: Says Lucy [laughing].

Steve: So that's the story.

Leo: That's pretty funny, I have to say.

Steve: Yeah. It's just - it's weird. And so this is what people thought of, they remembered, when they read these assertions in the budget request for the way the NSA wants to spend their money. And I want to - we've got about 15 minutes left before 1:00 o'clock, when you turn into a pumpkin, Leo. So I want to share what Matthew Green said because he's a PhD, he's got his master's, he's at Johns Hopkins, a cryptographer. He originally was thinking he wanted to write a book on cryptography, and instead he just took to blogging. His blogs are excellent. He blogs at blog.cryptographyengineering.com. Or just CryptographyEngineering.com, and then you can see the link to his blog.

He weighed in on this, and I'm going to skip down a little bit, saying: "All of this is a long way of saying that I was totally unprepared" - so this is he's just written this about the end of last week's revelations. "All of this is a way..."

Leo: This is the post, I should just mention, that Johns Hopkins initially forced him to take down.

Steve: I don't think it was this one, actually.

Leo: Oh, it was the one prior to that.

Steve: Yes, yeah.

Leo: Okay, yeah. He left it on blogspot. He didn't take it down there. But the university compelled him, in a really shameful display of kowtowing to the NSA...

Steve: Of academic censorship, yes.

Leo: Really horrible, yeah.

Steve: Yeah. And I'm sort of ignoring that because it's like, okay, fine. And but his take is compelling. He said: "All of this is a long way of saying that I was totally unprepared

for today's bombshell revelations describing the NSA's efforts to defeat encryption. Not only does the worst possible hypothetical I discussed appear to be true, but it's true on a scale I couldn't even imagine. I'm no longer the crank. I wasn't even close to cranky enough. And since I never got a chance to see the documents that sourced the New York Times/ProPublica story," and, he says, "and I would give my right arm to see them, I'm determined to make up for this deficit with sheer speculation. Which is exactly what this blog post will be." So then he talks about Bullrun and Cheesy Name, which are two of the project names. He says...

Leo: [Laughing]

Steve: I know. Cheesy Name, they named it. "If you haven't read the ProPublica/New York Times or Guardian stories, you probably should. The [takeaway] is that the NSA has been doing some very bad things. At a combined cost of $250 million per year, they include: Tampering with national standards (NIST is specifically mentioned) to promote weak, or otherwise vulnerable cryptography; influence standards committees to weaken protocols; working with hardware and software vendors to weaken encryption and random number generators; attacking the encryption used by 'the next generation of 4G phones'; obtaining cleartext access to 'a major Internet peer-to-peer voice and text communications system,'" and he writes in parens, "(Skype?); identifying and cracking vulnerable keys; establishing a Human Intelligence division" - the so-called HUMINT - "to infiltrate the global telecommunications industry; and, worst of all, to me, somehow decrypting SSL connections.

"All of these programs go by different code names, but the NSA's decryption program goes by the name Bullrun, so that's what we'll use here." So he says, "How to break a cryptographic system: There's almost too much here for a short blog post, so I'm going to start with a few general thoughts. Readers of this blog should know that there are basically three ways to break a cryptographic system. In no particular order, they are: One, attack the cryptography. This is difficult and unlikely to work against the standard algorithms we use..."

Leo: As Schneier says, trust the math.

Steve: Trust the math. He says: "...though there are exceptions like RC4. However, there are many complex protocols in cryptography, and sometimes they are vulnerable. Two, go after the implementation. Cryptography is almost always implemented in software, and software is a disaster. Hardware isn't that much better. Unfortunately, active software exploits only work if you have a target in mind. If your goal is mass surveillance, you need to build insecurity in from the start. That means working with vendors to add backdoors. Three, access the human side. Why hack someone's computer if you can get them to give you the key?"

And he says, he writes: "Bruce Schneier, who has seen the documents, says that math is good, but that code has been subverted. He also says that the NSA is cheating. Which, assuming we can trust these documents, is a huge sigh of relief. But it also means we're seeing a lot of two and three here."

So he says: "So which code should we be concerned about, and which hardware? This is probably the most relevant question. If we're talking about commercial encryption code, the lion's share of it uses one of a small number of libraries. The most common of these

are probably the Microsoft CryptoAPI and [embodied in] Microsoft SChannel" - the so-called Secure Channel - "along with the OpenSSL library. Of the libraries above, Microsoft is probably due for the most scrutiny. While Microsoft employs good - and paranoid - people to vet their algorithms, their ecosystem is obviously deeply closed source. You can view Microsoft's code if you sign enough licensing agreements, but you'll never build it yourself. Moreover, they have the market share. If any commercial vendor is weakening encryption systems, Microsoft is probably the most likely suspect.

"And this is a problem because Microsoft IIS powers around 20% of the web servers on the Internet, but nearly 40% percent of the SSL servers. Moreover, even third-party encryption programs running on Windows often depend on the CAPI components" - which is the Microsoft secure API, the Crypto API - "including the random number generator. That makes these programs somewhat dependent on Microsoft's honesty.

"Probably the second most likely candidate is OpenSSL. I know it seems like heresy to imply that OpenSSL, an open source and widely developed library, might be vulnerable. But at the same time, it powers an enormous amount of secure traffic on the Internet, thanks not only to the dominance of Apache SSL, but also due to the fact that OpenSSL is used everywhere," he has in italics. "You only have to glance at the FIPS validation lists to realize that many commercial encryption products are just thin wrappers around OpenSSL. Unfortunately, while OpenSSL is open source, it periodically coughs up vulnerabilities." I like that, like a fur ball. "Part of this is due to the fact that..."

Leo: [Coughing]

Steve: Yeah, "...that it's a patchwork nightmare originally developed by a programmer who thought it would be a fun way to learn bignum division."

Leo: Isn't that funny. He was studying C.

Steve: I know, yeah. "Part of it is because crypto is unbelievably complicated. Either way, there are very few people who really understand the whole codebase.

"On the hardware side, and while we're throwing out baseless accusations, it would be awfully nice to take another look at the Intel Secure Key integrated random number generators that most Intel processors will be getting shortly. Even if there's no problem, it's going to be an awfully hard job selling these internationally after today's news." Anyway, and he goes on - which standards, which people are involved, if it's HUMINT. And finishing up, it says: "So what does it all mean? I honestly wish I knew. Part of me worries that the whole security industry will talk about this for a few days. Then we'll all go back to our normal lives without giving it a second thought." I don't think that's the case, I'm saying.

And he says: "I hope we don't, though. Right now there are too many unanswered questions to just let things lie. The most likely short-term effect is that there's going to be a lot less trust in the security industry, and a whole lot less trust for the U.S. and its software exports. Maybe this is a good thing. While we've been saying for years that you can't trust closed code and unsupported standards, now people will have to verify.

"Even better, these revelations may also help to spur a whole burst of new research and redesigns of cryptographic software. We've ... been saying that even open code like

OpenSSL needs more expert eyes. Unfortunately, there's been little interest in this, since the clever researchers in our field view these problems as 'solved' and thus somewhat uninteresting." They're not interested in, like, some stinky implementation of the algorithms they've created. It's the math that turns them on. "What we learned today is that they're solved, all right. Just not exactly the way we thought."

So that's where we are. My feeling is that this is great. This, I mean, this brouhaha is, I mean, this is a perfect sort of second echo of the original Snowden shockwave that really hits the core of the crypto industry. We now have absolute evidence that there was influence a long time ago by the NSA on the standards body, NIST, that got this standard introduced. I also did read in all of the research that this particular standard got into Vista. So apparently it's in Windows Vista and 7 and 8. I haven't done any further research to track it down. But again, nobody uses it. In the same way that we've talked about how SSL, your browser has a whole bunch of possible security protocols that it knows, the server has a whole bunch, and they negotiate the best that they both know.

Similarly, this is - it's a random number generator that is weird and untrusted and a thousand times slower and nobody would choose to use it. But it's sort of there. So the point is, though, that as Bruce says, cryptography is ultraconservative. Cryptographers are. Never again will this be allowed to happen. And that's why this is a good thing. Elliptic curves themselves are fine. I mean, there are many good elliptic curve algorithms. We've moving towards them because they're faster; they use smaller keys. So there's, like, there's nothing wrong with an elliptic curve. It's just a - it's a way of implementing the discrete logarithm problem that we'll be talking about a little bit more in the future. So that's not it. It's when a standard says use this one particular curve from the family, and it needs to be in the standard. And then Appendix A says, oh, but you could also use random values, if you wanted to, as long as you also use the one in the standard.

So, I mean, it is highly suspicious. The good news is that's enough to be an object lesson for all time to never accept something like this and just shrug our shoulders. I mean, we now wish we had put our foot down and said absolutely not. Unless you can tell us where these numbers came from, we're not using them. Again, so this is great for that reason. I don't think this will ever happen again.

Leo: Good.

Steve: And my feeling is, what we've seen is a lot of hyperbole, a lot of glaring headlines to draw readers, the scariest possible conclusions drawn from admittedly scary intent. Certainly there is budgetary intent in, like, if you give us money, we're going to be able to do these things. So Congress, open your checkbook. So I think that creates some motivation for them overstating what they can do. But again, if they can install a keystroke logger in someone's machine, you don't have to break the encryption of what's leaving their machine because you get it beforehand.

Leo: Right. It just, you know, this is exactly what you'd expect. They're trying every method they can to get through the darknet. And they've got, apparently, budgeted a quarter of a billion dollars a year to do it. So, you know. Hey, if they want to subvert me, give me a few million dollars a year, I'd do it.

Steve: Well, and again, the technologies we're actually relying on, this is not this wacky

random number generator nobody uses. So maybe...

Leo: Not even ECC. People keep saying ECC's using this elliptical curve RNG.

Steve: No, no, no.

Leo: It's too slow. Why would you use something a thousand times slower?

Steve: Correct. Correct.

Leo: There's no motivation to do that.

Steve: And I'm a little worried that people are going to misunderstand ECC. Elliptic Curve Cryptography has nothing to do with this except it uses the term "elliptic curve." And the idea is that...

Leo: It's just the same name, I guess.

Steve: Right, it is. And so, for example, you could take - there are many ways to generate random numbers. You could take a hash function and take the output of the hash function and feed it back in. And then you're going to get a different output. And you feed that back in, you get a different output. And you feed that back in, you get a different output. Well, there's a random number generator using a hash function. Similarly, you could take a cipher, and you feed its output back to its input. Oh, look, now you've got random numbers. Or you could use an elliptic curve in exactly the same way. And that's what this does is it uses - it just sort of feeds it back to itself. The question is why that particular one because there's an infinity of them. And in Appendix A it says you could also use a really good random number and make up your own curve. Except the standard supplies one, and that's the point.

Leo: Right, that's a problem, yeah.

Steve: Yes.

Leo: And if you use an OpenPGP implementation, when you generate your - one of the concerns somebody had is, if Intel does put a backdoor, let's say, into a chip that's in hardware, these hardware RNGs, every PGP, OpenPGP implementation I've used of course uses random number generators, but also gathers entropy from mouse moves and keystrokes. Does that mean we don't have to worry, in that case?

Steve: Yes. Yes. For example, what I will be coming up with soon also needs random numbers because everything does. I'm going to have the person wave their camera phone around and stream all of the data from the camera into a hash. And so we'll take

the random number that iOS gives, but then XOR it with one we make locally so we get the benefit of both.

Leo: So even if the random number generator's flawed, and they're all - by the way, we should say pseudorandom number generators, that's the issue is it's...

Steve: Actually, no. We've passed that now.

Leo: Are we better at that? Oh, good, okay.

Steve: Intel is true quantum random numbers.

Leo: Got it.

Steve: It is not algorithmic. And I've read several articles now about what Intel's doing. And, I mean, it's wonderful. It's great. But again...

Leo: But even if it were compromised, hashing it with chaotic information produces an unpredictable result.

Steve: Yes, just turn the microphone on and digitize the noise and mix that in. And then even if there was some bias, you've washed that away.

Leo: Right. Okay. So you can trust the math, and you can trust open source implementations of things like PGP. I use GNU Privacy Guard, and I love it.

Steve: Yes. It is - and that's the key. Nothing actually mainstream, not SSL, TLS, nothing anyone is actually using has in any way been hurt. Just this bizarre, weird, seven-year-old, one particular elliptic curve random number generator that nobody would ever choose.

Leo: But it is a smoking gun that says, look, they are trying to subvert.

Steve: It's a lesson, yes, exactly. It's a fabulous object lesson. And that's why I named this podcast whatever I named it.

Leo: A Perfect Accusation.

Steve: The Perfect Accusation. Because, okay, you fooled us once. We're not accepting numbers that we don't know where they came from ever again.

Leo: No, no, no. And one can presume that they are attempting to subvert Microsoft and Google individual engineers or corporately or with NSLs. There is a constant assault on corporations. That's, by the way, the damage that this does.

Steve: Yes.

Leo: It makes us not trust anybody.

Steve: You heard me last week saying I have a problem with using Bit whatever it is, Microsoft's encryption.

Leo: BitLocker.

Steve: BitLocker. I just, like...

Leo: Because we don't know.

Steve: How could I trust that?

Leo: It's a binary blob, and it could be compromised.

Steve: Yes. Whereas we've got stories of people using TrueCrypt and Brazil sending drives to the FBI because they just...

Leo: They can't figure out what's in there.

Steve: They can't crack it, yeah.

Leo: Right.

Steve: Oh, and I forgot to mention also, I should have, because so many people tweeted, I said last week that Greenwald's partner Miranda had the password written down on paper. I'm sure you remember, Leo, that Greenwald is absolutely denying that. He's saying, whoa...

Leo: Oh, good.

Steve: It was not written down. That was made up. And, by the way, they have not decrypted the documents. Apparently there was some small cache maybe that weren't

encrypted in the first place. So, again, I wanted to make sure that I just said that all I was relating was the news that we had at the time. But Greenwald has said, absolutely believe me, we were - this was not written down on paper. So that sounds like GCHQ or whatever they're called, trying to excuse themselves for forcing, being able to claim poor...

Leo: See, we found a Post-it note; right.

Steve: ...security, yeah, claim poor security, and thus we're going to trash all your hard drives in the basement.

Leo: Sad.

Steve: Never a dull moment in security, Leo.

Leo: It is fascinating. And if you are interested, this is the show for you, every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 18:00 UTC on TWiT.tv, if you want to watch live. Steve offers 16Kb audio, for those who are bandwidth impaired but would like to still listen, on his site, GRC.com. You'll also find full English language transcriptions, thanks to Elaine Farris and Steve, who makes those available. GRC.com. When you get there you might want to buy SpinRite the world's finest hard drive maintenance and utility.

Steve: Pays my bills. Thank you.

Leo: Yeah. Thank Steve by buying SpinRite. And you'll get the benefit of this great tool. He's also got a lot of freebies, and you can check them all out. In fact, are we going to do questions next week? I guess so, huh.

Steve: Yeah.

Leo: Barring breaking news. You can leave your questions there, too: GRC.com/feedback. Steve does not do email. Don't try. People keep saying, what's Steve's key? I say, I don't know. He doesn't do email, dude. You can get full bandwidth audio and video of the show at our site, TWiT.tv/sn, or subscribe to any of our feeds on your favorite podcatcher, like iTunes, et cetera, et cetera. And you'll get it each week automatically. Steve, thanks so much. I guess...

Steve: Do we have you, or are you gone now?

Leo: I guess this is farewell, my friend.

Steve: For three weeks; right? So three...

Leo: I will be in Venice a week from today.

Steve: And do I know, are we going to have Iyaz? Or is Tom going to do it from his lair? Or who's going to...

Leo: Who's hosting next week? Tom Merritt will be hosting from the...

Steve: Okay.

Leo: From the Merritt lair.

Steve: Cool. I think that'll be fun, from one Skype to the next.

Leo: Yes, from one Skype to another. Yeah, so I'll be back October 9th, three weeks hence. But don't hold back, dude. Do it, man. I'll listen. I'll be listening to the shows. I'll be listening.

Steve: Okay.

Leo: All right?

Steve: Okay. I'll - okay.

Leo: You do what you want. You always do. I mean, I can't...

Steve: I expect that I'm a few days away from being able to put the work that we've got on data throughput aside. Then I need to work on communicating this.

Leo: Yeah, I'm sure...

Steve: And so we will just see how it goes.

Leo: It all takes longer than one expects.

Steve: It always takes longer. It's why I don't do schedules.

Leo: Yes.

Steve: I just show up here every week.

Leo: That's the schedule. And just a heads-up, we are looking at moving, I think, to Thursday; right? What did we...

Steve: Tuesday at 1:00 was the last I heard from Liz.

Leo: We're working on our next, our new schedule, which will debut next year, after the Christmas breaks. And some shows will be moved, including, I believe, this one. So I just want to warn people because I know everybody hates change. But it will help me because I will then get two consecutive days off instead of the doughnut that I get right now.

Steve: Okay. So have a great trip the next three weeks, and we'll be talking then when you get back.

Leo: I've got my Venice guidebook. I've got my Istanbul guidebook. I got my Turkish language lessons at Audible.

Steve: You got Audible, exactly.

Leo: I'm ready. I know enough Italian to be dangerous, so I'm - you know, it's fun, you can actually ask Google Now, you can ask, you know, you can say, "How do you say where's the bathroom in Italian?" and it'll tell you. It's kind of cool.

Steve: No kidding.

Leo: Yeah, you want to see? Okay, Google Now. Oh, it's locked. Is it locked? Yeah, let me unlock it first. Okay, Google Now. How do you say "Where is the bathroom" in Italian?

FEMALE VOICE: "Dov' il bagno?"

Leo: Did you hear it?

Steve: Uh-huh.

Leo: Dov' il bagno? That, see, that's the future, right now. Right here, right now. I love that. All right, Steve. We'll see you in a month.

Steve: Yes.

Leo: On Security Now!.

Steve: Thanks, Leo.

Leo: Bye-bye.