SQRL Q&A #176

Description: Following up on last week's "SQRL - Secure QR Login" podcast, this week's Q&A focuses upon the many interesting questions Steve's description of a new approach to secure website login sparked in the minds of the podcast's listeners. And, of course, we also catch up with the week's news.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-425.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-425-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here. We're going to have a follow-up on his amazing SQRL technology. We'll answer your questions, too. I'm back. Steve's back. Let's talk. Security Now! is next.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 425, recorded October 9th, 2013: SQRL & Q&A #176.

It's time for Security Now!, the show that covers your security, your safety, your privacy online, with the - are you - did you just have a muffin? What? Steve Gibson's wiping his mouth off here. Oh, okay, coffee thermos.

Steve Gibson: That's my coffee thermos, and I wiped a drip off.

Leo: Steve Gibson's here, the Explainer in Chief, the man behind SQRL. Steve, I've been gone three weeks, and you announced a massive, major new initiative.

Steve: Yes.

Leo: Last week.

Steve: Yes.

Leo: By the way, was it Iyaz or Tom? Who filled in for me? Was it Tom?

Steve: Tom did all three.

Leo: Thank you, Tom.

Steve: And there was just so much pressure that we were getting from our listeners that, I mean, I wanted - I thought it would be fun to do it. But then I thought, well, I can get you both. I can do it with Tom, and then I can do it with you.

Leo: That's quite all right. No, no. I would never want you to hold back. IBM used to do that. They would hold back technologies because they wanted to have something for next year. And I think that's reprehensible. If you've invented something, if you've innovated something, then why hold it back? And you know what I love about SQRL - and I want you to explain it a little bit to me, for those who missed last week as I did. But what I love about SQRL is you're not encumbering it. You said this is patentable, it's a novel idea, but I'm going to give this to the world. And I think that's great, Steve. Thank you.

Steve: Yeah, well, I mean, there may be some parts about it that are novel. And I'm sure if it came from the typical corporate syndrome there would have been attorneys scouring it, looking for little nooks and crannies to say where it was innovative. One of the things that's happened in this past week, it's hard for me to believe it's only been a week because it's just been an amazing reaction, just phenomenal. I created a newsgroup at GRC on our news server one week ago, in the morning before last week's podcast. And we're at over 1,100 postings.

Leo: Whoa.

Steve: There's a page now of projects that are underway among the SQRL pages at GRC. So a bunch of people are immediately racing to implement this on Android and iOS and web server platforms, in every language you can imagine. I've even been contacted by the W3C, the HTML5 spec editor, who says authentication and login is like a serious problem, no one has solved it yet, this looks wonderful, let's talk. So...

Leo: So let me see if I understand it. And sometimes paraphrasing is the best way. It is a way of using either QR codes or some other secret that's shared, authenticating yourself to web pages without giving - anonymously, effectively, without giving the web page any other information about you - right? - and not using a third-party service. This is a transaction just between you and any - let's say, let's pick a website. Let's say Google decided to implement this, which would be wonderful. Then I would just - this is what I - explain if I've got this right. I would go to Google and say, okay, whip out your phone and snap this QR code, and then I would be logged in.

Steve: Right.

Leo: What? How does that work?

Steve: So, okay. The shortest way to say it is that you take a combination of your own sort of like grand master secret key, which is never used online, never leaves anywhere.

Leo: So it's the equivalent of a private key in a public key cryptography system.

Steve: No.

Leo: No.

Steve: No, because there you have a matching public key. This is an absolute secret that is just - it's randomly generated, just pure randomly generated. And it's yours. And one of the questions that we have today is, well, what if someone else got the same one? So, because in a large population of people, as we know...

Leo: That's going to happen; right.

Steve: ...there's the birthday attack problem, which is where it's surprising how few people you need to have for any two of them to share the same thing, like they have the same birthday and so forth. So there's only 365 days in a year, so there's that many possible birthdays. How many people do you have to have together before any two of them have the same birthday? And the number is smaller than people suspect. But it turns out that's not a problem here. So you get this incredibly random large number which you never disclose.

Leo: Okay.

Steve: That's - it is never disclosed.

Leo: It's kind of like your - is it like your Bitcoin address, kind of, sort of?

Steve: Actually, it's very much. Bitcoin is also 256 bits. I changed that, by the way, since last week. We dropped it from 512 to 256 just because...

Leo: It's plenty, yeah.

Steve: ...there was no need for 512 bits.

Leo: Now, with Bitcoin you could generate multiple addresses. Can I do that, as well?

Steve: Yes, you can. And so you can have multiple users, essentially, defined in one little app. But let's take the case of one user. So you've got this 256-bit, absolutely random thing that is uniquely yours. What this does is it combines that with the website domain, like Google.com. And then it uses that, it uses the combination of your super secret and the website domain where you're going to create an asymmetric key pair, to create the public and private key.

Leo: So it's like a hash of the two?

Steve: It does. It uses actually an HMAC operation, a Hashed Message Authentication Code, which is just, like, it's an extra-secure kind of hash. It just uses hashing operations a couple times to do essentially the same thing, but it's a little stronger in some ways than just a hash because there is a vulnerability potentially because the domain name is being hashed with your secret key. It's better to use a message authentication code where it was just keyed by your secret key, and then you use that to hash the domain name. It's just technically better. So it creates a little more isolation for, like, things the user could control, like the domain name going into the hash. But so the idea is, armed with this unique key, every website you use generates on the fly a unique asymmetric key pair.

And the point is that since your super-secret master key never changes, every time you come back to that website, you get the same asymmetric key pair, a private and a public key. Now, what we do with that is simple, really. The QR code that you mentioned, a phone can snap it, can scan it. And we simply take that and sign it with our public key, and we send that signature and the public key to the site. So the public key is our identifier for that site. So, for example, every time we go to Google we generate the same key pair, and so we're going to have the same public key. So we always look exactly the same to Google. It doesn't - we're completely anonymous. It knows nothing about us except this is our token.

And but then the other cool thing is, since we've signed the QR code with our private key, and we sent that along, Google is able to verify the signature with our public key, which is our identity, which proves to Google that we the person who has this public key also have the private key because we just signed the QR code that we were given. So it's very simple. There's, like, no extra moving parts. Everything is tightly locked together. Yet what this asserts is that the same person coming back to the site any time in the future is the person that was originally identified.

Leo: As long as you keep control of your private key.

Steve: Yes. Yes. And that's one of the most important things. There's been fabulous discussion, for example, like people said, well, what about revocation if a key gets away from me?

Leo: Oh, good one, yeah.

Steve: And so it's absolutely true that there are things we're giving up when we cut out a third party. And that's the other issue is that all of these other authentication-identification systems...

Leo: You're talking about things like Facebook Connect or Google Connect or...

Steve: Well, yes. But also there are, like, there are committees and authentication proposals in the works. And there are so-called "federated identity systems," more like the VeriSign deal where remember that the original football that we talked about years ago, this is my little PayPal football. Well, when I give the number to PayPal, PayPal gives it to VeriSign. So VeriSign is a third-party authenticator.

Leo: Ah, okay. Is that true - that's not true of Google Authenticator.

Steve: No. But, see, the problem with Google Authenticator is that it's a - there you do have a shared secret. I've got that secret token that runs the sequence in my phone, and Google has the same secret that runs their sequence so we can make sure that we agree. Now, while that's nice, and it is two-party, that is, there's no third party, the problem there is we have a shared secret. There is something for Google to lose. And the beauty of my approach is there's nothing for them to lose. They don't have a secret of mine. All they have is my identity, which is only valid for them. Because it comes from their web domain, when I go to a different website, I present a completely different identity in no way connectable to the identity that I have for Google.

So this creates a unique identity for every website on the Internet. Yet every time you go back, you're the same as you have always been for that site. And so, for example, the reason - what's happening with Google Authenticator and the increasing number of sites that support it, is now we end up with a whole - with a growing list of six-digit numbers that are all changing. Why can't we use the same one for multiple sites? Well, we're back to the same password problem. And that is, Google, I mean, people's databases are being hacked all the time. We just had a huge Adobe breach where they lost 1.2 million of their user accounts. So again, you have to, if you're going to share a secret like a password or like your authenticator, then there's the problem of you needing to have separate ones for each site and managing them separately. Here there's no shared secret.

Leo: Okay, good. I'm sorry. You paused. Do you have two different coffee cups? It looks like you're double-dipping there.

Steve: I have this little cup and a warmer. And then I've got my thermos is where the coffee comes from after having been transferred from the coffee pot.

Leo: So as I remember, you kind of had - this all came to you in a flash, like

Archimedes in his bathtub.

Steve: I was having coffee at IHOP while I was having breakfast, of course, on a Thursday morning about six weeks ago. And this was kind of like, oh. I mean, it's like, wait a minute. And I thought, well, okay. Does this work? And so I thought about it for a few more minutes. And then I decided, like, it seemed to. So I kept it to myself for about a week or two while I was continuing to work on wrapping up this phase of work on SpinRite. And then I just - I couldn't - I said to the guys in the SpinRite development newsgroup, I said, okay, look. I have to confess I'm being distracted by something. We're going to get this work we're doing now finished.

Leo: SQRL. It's well named, I might add, yeah.

Steve: What?

Leo: SQRL.

Steve: Yeah. SQRL, yes. Now, one of the things that Tom mentioned, one of the first things he mentioned was, but what if I don't want to use my camera? What about just logging in with my laptop?

Leo: Well, that's the question. What does it require for me to participate? First of all, and this might be the biggest hurdle, websites have to adopt it.

Steve: Yes. Well, okay. So that's the other thing is that there's been some confusion. One of the things, the reason I consider this a low friction for adoption is, first of all, it's all open source. It's all open spec. It's all just there. It's also very simple. I mean, we're going to have apps, implementations of this, a few weeks from now. So...

Leo: There's already a Chrome extension. So, I mean, it's moving fast. It's moving fast.

Steve: And it's funny because I made a comment yesterday, or last week, I was wondering how long it would take Google to figure this out. And it was four hours, I think, from the podcast moment where you put SQRL in, and bang, Google knows all about it.

Leo: They scan it fast. So, all right. So presuming there's adoption, and there are good reasons to adopt it, what does it take for me to participate? Do I have to have a smartphone?

Steve: No. That's one of the first things we looked at was how do we eliminate the phone? Because a lot of people like the idea of the phone as a second factor, a physical

second factor. And, for example...

Leo: I love it. I use it at whatever site offers it. It's great.

Steve: Yes. Now, imagine in a library or a public kiosk. What this literally lets you do is snap a QR code that's being displayed on a computer you do not trust. And without entering any of your credentials, you're logged in. So, I mean, so that's really a change. That's really cool.

Leo: And there's no leakage of information at that point. I mean, I could - nobody can come to that terminal and figure it out or anything like that.

Steve: Correct.

Leo: I walk away with my phone and, boom, we're done.

Steve: Yes. That whole computer and its connection got none of your information. What happened was that, from its viewpoint, spontaneously, you were logged in at that site. It just kind of happened behind its back. So, again, it has no secrets to disclose. There's nothing for bad guys to get, no keystrokes to be logged, no malware to catch anything. It's just done. But many of us sitting at our computers at home, we have laptops and so forth, yes, we have a phone by the front door charging, and we'll use it when we're out and about, but right now it'd be nice just to be able to authenticate with our computer.

The effort that's underway is for an app to be installed, like any other app, which registers the SQRL scheme, in the same way that HTTP - if you click a link on your computer which has HTTP, your computer knows to launch the web browser and give it that link to display it. If you click a link that says "mailto," your computer knows your email client should be launched in order to start up a mail note. So, similarly, an SQRL app on desktops, desktops and laptops and, in fact, tablets, for example, it would register the SQRL scheme, which is what those first letters are called. So when on your desktop you go to Google.com or Amazon.com or whatever, and you see this little SQRL QR code, you click on it. You don't use your camera. You actually click the little SQRL QR code because it is also a link. It's an optical link.

Leo: Okay, yeah.

Steve: But it's also an HTML link. So you click on it, and that launches the little authenticator which is on your computer in the same way that your web browser is or your email client is. This is the little SQRL app. So it launches it, giving it that link. And so it is then able to handle all the authentication for you. And again, you just leave the login form blank and click "Login," and you're logged in. So it is, again, completely seamless authentication.

Leo: Wow. You know, it reminds me a little bit of - have you looked at Duo Security?

This is a multifactor system that LastPass is using.

Steve: I haven't.

Leo: It's interesting. It's a new option on LastPass. They've added, in addition to YubiKey and Google Authenticator, something called Toopher, Duo Security, and Transalt [ph]. I think it's Transalt. But the way Duo works is when I open LastPass on a phone, the Duo app on my other phone that's been set up with Duo gives me an alert, and it says, hey, somebody's trying to log into LastPass on this device. Do you want to allow it? I say "Allow," and all of a sudden I'm on. It's very magical and mysterious, and I don't know if it's related or not. But it feels like that. It would be - so would LastPass be able to use something like this as a second...

Steve: They could, yeah.

Leo: So it is, it could be used as second-factor authentication.

Steve: Absolutely.

Leo: Yeah. I'm not sure I completely understand it. But well, here's the good news. It's completely described in great detail on your website at GRC.com/sqrl. And people can read about it. And how has the adoption been? I mean, it is only a week old, but people are...

Steve: Okay. So it couldn't be any, I mean, it couldn't be happening any faster. I was planning on continuing to flesh out the pages that I had been working on. For example, there is a page about attacks where I've enumerated all the different sorts of things that I can think of and anyone has been able to think of that could go wrong. And so that was the next thing I had planned to work on. The problem is we've got about 20 people that, like, want to write it right now. And I said, uh, okay, uh, okay, wait. And so I've switched over to - in fact, if you look, there's a Projects and Implementations page where I'm following all of the projects that have been created. There's about four or five of them on GitHub and various other places. So I've switched over to - there's something called "Implementation Details" page, which is up now, which I spent Monday and Tuesday working on, the last two days, because I need to present a proposed protocol because people insist on writing code right now. And so what will happen, Leo, is that short - oh, and I should tell our listeners that I am going to write one.

Leo: Oh, good.

Steve: I will - I'm going to write...

Leo: You need to at least write a reference one.

Steve: Well, yeah. I'm going to write a reference Windows, because I'm not a mobile platform developer, but I am a Windows coder, as everyone knows. I will write a Windows and Wine, so anyone with Wine, any Linux people, a reference implementation for the desktop and Windows laptop and Wine laptop application that will be, of course, it'll be in assembler, so it'll be 2K bites long. It'll actually be a little longer than that because I'm going to have to bind in some libraries for the crypto. But it'll be super small. I will, of course, sign it with GRC's Authenticode certificate and make it available. So there will be mine. But there will be a whole bunch of them. We're going to guarantee interoperability. And so the clients will be there. There's already a Drupal...

Leo: I saw that. That's great.

Steve: ...and WordPress plugins are coming. So the clients will all be free, and they'll just be there. And so as websites adopt this, people will begin to use it. And an example I gave was, like, how many times have we gone to a place where someone has done an interesting blog, and you think, oh, it'd be fun to contribute to that. Well, but the thing wants you to go through all of this...

Leo: Rigmarole, yeah, yeah.

Steve: ...account creation rigmarole. And so sites are losing all kinds of valuable input because it's just too annoying to have to, like, identify yourself.

Leo: And so we face that. And the way we do it, and it's not completely satisfactory, but we will use Facebook Connect, or Google+, or there are a number of these; right? And that makes it easy because you already have a Facebook account. And you see this on apps, too. You just say, yup sign in with Facebook. The problem is that Facebook's getting all that information. So this would be a similarly simple way to log in...

Steve: One on one.

Leo: ...but no one else gets it. Just it's between you.

Steve: Yup.

Leo: And there's no man-in-the-middle issues with this. I mean, it sounds pretty good. You're using SSL, of course, in the transaction.

Steve: There's also no NSA in the middle.

Leo: Yeah, we like that, too.

Steve: And, see, all these other, like, all these other authentication technologies that are sort of out there churning and haven't really happened yet, they are so-called "federated identity." There's people like VeriSign or someone, I mean, there's a so-called "authentication service" that you authenticate to, and then the site authenticates to, and so you can cryptographically assert that you are who you are. But so can the NSA then offer them a national security letter and say we want to know all the sites that this user has been going to, has been logging into. I think the day has passed where anyone is willing to trust a third party because we now know how vulnerable they are to, unfortunately, our own federal government.

Leo: Yeah. So this is very timely in that respect.

Steve: Yeah. Yeah, we'll be talking, there was some news about Ladar Levison and exactly what happened with Lavabit.

Leo: Oh, yeah, yeah.

Steve: Oh, our worst fears were - actually came to pass.

Leo: Well, I didn't want for - I apologize to everybody who listened last week, I didn't want for you to completely recap the show last week. I just wanted kind of the executive summary of what SQRL is. And I know this is a Q&A episode. We're going to have a lot of questions about it, as well.

Steve: Yeah. There have been - lots of interesting issues have come up. My sense is, I mean, I have no horse in this race. I just sort of - you know as well as anyone, Leo, I have been focused on authentication as THE problem...

Leo: It is.

Steve: ...that needs to be solved.

Leo: It really is. There's no question about it.

Steve: I did Perfect Passwords, Perfect Paper Passwords, Password Haystacks, the Grid, whatever that thing was, the Latin squares, Off The Grid system? I mean, it's just - it's been on my mind. And so here was like, wait a minute, this solves a class of problems where a site wants to collect anonymous connections. And people have said, well, sites want to be able to leverage who you are and market to you and don't want you to remain anonymous. Well, they're sort of confusing two things. This provides the identifier token. But then they could still have you fill out forms and email addresses and first and last name, I mean, whatever it is...

Leo: Oh, yeah. They could attach stuff to it if they wanted to.

Steve: Absolutely. But it's not necessary.

Leo: That I like. So I have to say it's very intriguing. And unfortunately, like a lot of good ideas, it requires a groundswell of support from everybody. I think the need is clear, and I think the idea is good. I just hope that everybody goes, okay, yeah, we'd like to do this. You know, you get a few major - you know, the problem is that neither Google nor Facebook really wants anything like this because they've got their own solutions which give them information, and they want that information. We have to demand this, frankly.

Steve: Yeah. And again, it could be easily added. Look how long it has taken sites to finally adopt two-factor authentication. I mean, one of the stories today is that Evernote just finally added second-factor authentication, after that massive February breach that they suffered. And they promised to do it, and they're finally rolling it out today.

Leo: What? No, no, they've had it for a while.

Steve: Evernote?

Leo: Yeah. I've been using it for at least a month.

Steve: Oh, okay. I just saw the story that they...

Leo: Yeah, no, no, no, it's been going on for a while. And it's great. And I use it. And in fact it uses, which I love, Google Authenticator.

Steve: Yes.

Leo: So it makes it very easy for me to add, yeah, sure, I'll add another layer of authentication. It's not a...

Steve: Yeah, now, if they - if you were all - if the world had evolved a little differently, and you were already using SQRL for other sites, and Evernote added SQRL, then it'd be like, oh, look, I can...

Leo: Yeah, there'd be some incentive, yeah.

Steve: Yeah. So anyway, I mean, my role here is we'll get a spec out. We'll create a bunch of apps. I mean, I want to get back to SpinRite as soon as I can. But this exists,

and we just need to nail it down, make it real, and then see what happens. I mean, it's a proposed alternative for authentication. What I can say is our listeners all got it. I mean, it was phenomenal what the reaction was.

Leo: Well, but they know you, and they trust you. And we have to get the greater world to do the same.

Steve: Well, and it may take a while, which is fine. My app will exist. Other apps on other platforms will exist. Mobile platform apps will exist. There it will be. And then it will be users beginning to bug websites, I want to be able to authenticate with SQRL, go support it. And people will be creating all of the web-side...

Leo: It should be very easy for sites to do. That's one of the advantages Facebook Connect has. It's just a line of code, or Google Connect. It's a line of code. It was very easy for me and others to, even without a big IT department or programmers, to just add that support. And when both Google+ and Facebook added this, uptake was very quick, almost universal. But so I would love to see that. I mean, it would be just great to see, I think. But Facebook is never going to do it, and Google's never going to do it, because they want to collect information. And this is a separate, you know, this kind of deals them out of the loop, potentially.

Steve: Well, they did Authenticator, and there's no...

Leo: That's true.

Steve: ...information being collected there.

Leo: No, and I believe Google's - I believe at least Google, maybe not Facebook...

Steve: Well, and Google is actively looking at this. I mean, they're working closely with Yubico and Stina and the YubiKey.

Leo: Oh, good. Oh, good.

Steve: They've got pilot projects and things. And in fact they did play with QR code login briefly. A couple years ago, for like about a month, there was something where you could - that you could - they would present you with a QR code. You could snap it, and the login sort of jumped over to your phone. It took it away from the website over to your phone. And it's funny, too, because there have been - I've been flooded with people saying, oh, Gibson, this has been done before. And then they'll send me a link to something which has a QR code, but that's the only thing it bears in common. So I also have...

Leo: This is unique, as far as I can tell. There's nothing like this, yeah.

Steve: I do have a page of all of that other stuff that people are finding, just so it has a place to live, so I can say, yeah, we've seen all of that, and none of it is the same. There's even been some people saying, like showing me patents. And if you look at the diagram on the patent, it's got 26 different things all pointing at each other. And it's like, okay, look at my picture, and look at their picture. There's just no comparison.

Leo: All right. So thank you. And I apologize for everybody who's already heard all this. Steve Gibson, Leo Laporte. Before we get to questions and answers, let's get some security news.

Steve: Well, we are, as you mentioned, we've just passed the second Tuesday of the month. This is one of those earliest Tuesdays it could be because the first Tuesday of the month was the first of the month, this being - wait. Yeah, yeah. So yesterday was the 8th, and today's the 9th. So we have eight patches from Microsoft for their October surprise.

Leo: I like that, the "October surprise." I like that.

Steve: Repairing 26 unique identified vulnerabilities. Of these 26, 17 were remote code execution, six were elevation of privilege, two were denial of service, and one was information disclosure. Microsoft is beginning now to prioritize these for administrators because they recognize there is friction on the part of IT companies or IT organizations in large companies because there have been mistakes made before. So Microsoft says immediately patch numbers 80, 81, and 83. Then you should also do 82, 84, 85, and 86. And the least important is 87.

Leo: Yeah, well, of course, as always.

Steve: So we had Internet Explorer that had nine privately reported execution vulnerabilities. That's obviously...

Leo: Nine?

Steve: Nine. Ten memory corruption vulnerabilities when parsing specially crafted web pages. And this is the update that fixes that active exploit that was being patched in the wild where there was a Fixit that I tweeted a couple weeks ago and recommended - the way I said it, for anyone who must use Internet Explorer, if you're using it, you probably ought to apply this Fixit because there's no patch yet. So this is - this was patched in this month's Patch Tuesday, so do that. Also there was a kernel mode driver problem in font parsing causing memory corruption, and a use-after-free vulnerability. Seven privately reported vulnerabilities there: two remote code execution, and five were elevation of privilege. Problems in the .NET framework. Problems back in the old Common Control Library, the old comctl32.dll, a problem was found. Also in SharePoint Server, Excel,

Word, and Silverlight. So it's a grab bag, one of these big months, and definitely worth updating your Windows, especially because we've got 17 remote code execution vulnerabilities in IE. Oh, and several of them are being exploited in the wild now. So that puts an edge on the need to update.

Leo: No kidding. Wow.

Steve: Now, I tweeted - this hit the news days ago. And The New Yorker did a really nice four-page piece summarizing what our friend Ladar Levison went through with the shutdown of Lavabit. As our listeners will well remember, he was gagged and unable to say anything about what had actually happened. And so there was that frustration, that he couldn't even tell us. And what he posted on his site when he shut down his pseudo-secure email service was that he could not in good conscience keep it open lest he be committing a crime against the people of - all of his customers who were using his service. So he chose to shut it down. He was threatened by law enforcement for the act of shutting it down, which that was a little galling.

So I'm not going to read this piece. It's long. But I did tweet it recently, and I recommend it. In fact, I gave it a bit.ly shortcut: bit.ly/grc-lb, all lowercase, as in Lavabit. So grc-lb will take you to this nice New Yorker piece. But here's, in summary, what we learned: They did demand that he turn over his SSL certificate.

Leo: As you had speculated. They wanted him to stay in business, to be basically fully compromised, and have no one know it.

Steve: Yes.

Leo: So his decision was, look, either I continue to run, and every single person who has an account on my site will be open to the feds, an open book, or I just shut down. And he had no choice.

Steve: Yup.

Leo: And he couldn't talk about it.

Steve: Yes. So he was gagged. First he said, look, I can't hand over the master keys. Basically it's the master keys to his business is the way to think about this. I cannot.

Leo: So he didn't do that? He didn't give them?

Steve: Well, not initially.

Leo: He kind of had to; right?

Steve: He said, for $3,500, paying me just for my time, I will write the code you want to give you the access to the individual whose account you want to monitor. I will provide you with that access. They said no. We want everything. And he said no, I can't give you anything. The judge then sanctioned him $1,000 a day...

Leo: Oh, my god.

Steve: ...until he complied with a court order to turn over the keys.

Leo: And he can't say - he can't tell. Can he get a lawyer?

Steve: He had a lawyer working with him. And his lawyer, of course, was bound by attorney-client privilege not to disclose. So, yeah, he...

Leo: Nobody did say anything about this.

Steve: Nothing could be said publicly.

Leo: This is happening completely in secret.

Steve: Yes. So the only way we know all this finally is a court did release some of this information. There was a court order to release some documentation. Then we got some information.

Leo: He in his farewell said, I can't tell you. I have tried, I think he said three times, to get permission to tell you what's been going on, but I can't. Well, apparently he did subsequently get some permission.

Steve: Well, so, yes, and that was just recently that he was able to get some permission.

Leo: Oh, my god. Well, I'm glad I paid for 10 years of Lavabit right before he closed down because that's gone towards his legal defense. He deserves some help on this.

Steve: Well, so what he finally did, because he had this thousand, I mean, he's not a rich person. He was doing all right, but - so he printed out, in text, in four-point type, and it took 11 pages, his certificate key cryptography.

Leo: Here [laughing]. You want it? Here.

Steve: He gave it to them. He gave it to them, and they were not happy.

Leo: No.

Steve: They complained that this was...

Leo: We can't read this.

Steve: ...nearly illegible. In four-point type it could not be OCR scanned. Someone would have to manually type in 11 pages of incredibly tiny text characters.

Leo: That's like paying your taxes in pennies, basically.

Steve: And one single mistake made anywhere would render it completely useless. And his attorney said, well, there it is.

Leo: There you go.

Steve: You wanted it. And so then they upped it to $5,000 a day and held him in contempt of court, upped it to $5,000 a day, and a couple days later he shut down Lavabit because he said, sorry, I can't do this. In fact...

Leo: Did that get him off the hook? I mean...

Steve: I think he gave them the keys and unplugged the servers and wiped them.

Leo: You can have the keys, but I'm going to take the car.

Steve: Precisely. It was, here are the keys, sorry there's nothing for you to plug them into any longer. No more traffic will be encrypted using those keys. So that was the story.

Leo: I hope that he can get away with this. I mean, I hope that this doesn't cost him - are they still prosecuting him? Or are they just unhappy?

Steve: They're not happy.

Leo: That's fine. I don't want them to be happy. They should be miserable.

Steve: Yup.

Leo: But I want him to not have to pay $5,000 a day or go to jail.

Steve: So in the story it says, he says, "It was the government's insistence on collecting the SSL keys that most deeply disturbed Levison and led to the shutdown of Lavabit. He believes that not only would the FBI have had unfettered secret access to the communications of his 400,000 customers without being required to give Levison a log of what it accessed, but putting his encryption keys in the hands of the government would have opened Lavabit to a more profound exploitation of his service's communications.

"Levison worried that, if he turned the keys over to the FBI, the NSA would have been able to obtain them without his knowledge through a Foreign Intelligence Surveillance Act court order. We know now that the NSA has been systematically cracking encryption across the web. And it has built a database of encryption keys that automatically decode messages. This is dangerous, Levinson said, because it allows the NSA to read encrypted communications as they flow past the agency's taps of the broader Internet infrastructure by simply observing them, leaving no trace of the surveillance, unlike traditional man-in-the-middle attacks.

"This vulnerability, he insists, is not sufficiently understood. And while the Times' initial reporting indicates that the NSA's method of obtaining the keys for its database is shrouded in secrecy, Levison suggests that his case also illustrates one of the ways in which it collects them, by secretly compelling companies to turn them over."

Leo: Now, presumably they've done this to other companies, other people.

Steve: Exactly.

Leo: Who haven't had the integrity to step forward as Ladar did and say I'm not doing it.

Steve: Well, and who also have shareholders. As was observed when this occurred, Ladar was able to say no and give them the finger because it's just him. I mean, he was able to make the decision unilaterally to give up 10 years of his life building this service because of his own integrity. But no CIO or CTO or COO or CEO could do that. They'd just be fired by the board.

Leo: Now, companies like Facebook and Google have said, well, I don't know if they've categorically denied this particular thing. Have they?

Steve: Well, remember that email companies don't have a problem because email is almost never encrypted. And so my original hypothesis is that PRISM could simply tap upstream of Google...

Leo: Right. We know they're doing that, by the way. Thank you, Dianne Feinstein. Senator Feinstein inadvertently last week said that they were doing that, in public

testimony in the Congress, in the Senate. So thank you, Dianne, for letting us know. She's, of course, on the Senate Intelligence Committee, confirming what you've said all along. In fact, Steve, you have been right about this all along. This is exactly what you presumed was happening with Ladar Levison.

Steve: Because ultimately it's all driven by technology. As I said, it's about - it's the politicians are scratching their head, trying to figure out what to do. There will be somewhere there's policy. Ultimately, it's about bits and math. And so we can - we understand bits and math here. And so we can say this is what's probably ultimately happening. Where the hardware meets the software and the data passes through it, this is what's going on.

Leo: Yeah. Wow. Wowie zowie.

Steve: So, but the ultimate takeaway is we now know that companies have been, we know of one specifically, have been asked by the federal government to divulge their SSL keys and have had to decide what to do. Ladar said no. Major companies cannot say no. So maybe, if you parse the exact terminology of the CEOs of these major firms, they don't consider giving their keys up working with the federal government. I don't know how you parse that. But it is certainly what the NSA wants.

Leo: Just makes me weep.

Steve: Really, it almost makes - I remember when I read this the first time I was a little queasy. It's like, oh, crap.

Leo: Well, we should just assume this happens, and that the NSA now has all - there is no secure SSL traffic. But this doesn't mean that PGP is not safe. And really this is why the only really safe way to do email is using PGP or GPG or some form thereof.

Steve: So while we're on the topic, I will take this a little bit out of sequence.

Leo: By the way, just to the Evernote authentication, that is now available to all Evernote users. It was premium users only. That's why I thought - so I'm a premium. Of course I pay for Evernote. And so I've had it for a while. But everybody can get two-factor authentication.

Steve: What I remember reading was free users now had this, but paid users also have something else. There was some other level of or type of authentication.

Leo: Well, I must check.

Steve: Which wasn't free, I think, for Evernote to offer. Whereas authentication, the

Google authentication style, as we know, is just two-party and is free.

Leo: Evernote's just fabulous. They've just added the ability to edit PDFs in Evernote. I mean, it's just a great tool. I've put it on every single device I have. And that's a lot of devices.

Steve: So I have a sci-fi alert. I know nothing about this. But tonight, and for those who don't listen until tomorrow or Friday, also re-airing Friday night, is the premiere of a new series called "The Tomorrow People." It has an eight out of 10 ranking on IMDB. The short description is "The story of several young people from around the world who represent the next stage in human evolution, possessing special powers, including the ability to teleport and communicate with each other telepathically. Together they work to defeat the forces of evil."

Leo: That sounds like heroes.

Steve: Yeah, it's heroes mixed with jumpers mixed with a couple things. Anyway, I don't know what it is. Greg Berlanti is one of the main guys behind it, and he brought us "Everwood" and "Brothers & Sisters" and "Dawson's Creek." So take that for what it's worth. I liked "Everwood" and "Brothers & Sisters" a lot, actually. And this is on the CW, so maybe it's going to be bubblegum. I don't know. I'm just saying it's there. So for anybody who's interested, maybe we'll have something. It was based on a much older British series by the same name, "The Tomorrow People." I think it was back in the '70s, like really old. And so we'll see. Maybe they've updated it and will give us something good. My recorder will be recording it, and so we'll know.

Speaking of the NSA, I got a bunch of tweets from people because I have several times talked about how much I like the series "The Good Wife." And remember, Leo, that many people were recommending it, and I was saying, what? I'm not watching something called "The Good Wife." That just doesn't seem like my kind of thing. Well, I love it. But even if anyone has never seen an episode, go find last Sunday night's. It was Sunday, October 6th. The whole thing, well, okay, three quarters was the NSA listening in on conversations. It was really funny. They just having - they had a huge amount of fun with this, with two geeky 20-somethings, each with these big workstations and data streaming in and eavesdropping and networking people, and, oh, now they've got permission to go from two stages removed to three and so forth. Anyway, I got a bunch of people brought it to my attention even before I had seen it because it was sucked in by my DVR, but I hadn't watched it yet. Anyway, it was really fun. So if you're at all curious, you want to see a show, the first one that I've seen really having fun at the NSA's expense about all of this kind of eavesdropping, "The Good Wife" Sunday, October 6th episode was great.

Now, I was also trying to save any mention of this specifically until you got back, but we had a light news week last week, and so I thought, well, okay. And that is that Jenny just had a book published. I knew you'd get a kick out of it, Leo.

Leo: Oh. Congratulations, that's great.

Steve: It's a children's book titled "Is God Real or Pretend?"

Leo: [Laughing] I love it. And what's the answer, by the way? I'd be curious.

Steve: Well, it's comparative religion for kids. She has a character, Benjamin, and he's got a dog because Jenny's a big dog person, who has, like, his grandmother, Dr. Wendy Knowles, I think, who he talks to about this. She's a professor of astronomy. And he goes to somebody else. Anyway, he ends up meeting through the - it's not long, I think it's 66 pages - meeting the major leaders of the world's five biggest top religions - Hindu, Buddhist, Jewish, Christian, and Muslim. And they explain their religions to Benjamin.

Leo: Oh, that's neat.

Steve: Oh, it's...

Leo: So it's kind of open-minded. It's not saying yes or not.

Steve: Oh, yeah, yeah. And apparently our listeners love it. They've been buying the book. They've been - multiple copies. Several...

Leo: Is it on Amazon? Where can I get it?

Steve: It's on Amazon. And in fact I wanted to ask our listeners, if you did buy it and liked it, of course we all care about people's opinions. And so if you could take a moment to go back and add a review to her site or for the book, that would be great.

Leo: "Is God Real or Pretend?"

Steve: Or pretend. I just love the title. I just, you know, it just - I got a kick out of it. And then Jenny said, "You didn't tell them that you designed the cover." It's like, okay, well, that's really not relevant.

Leo: You did the cover? Wait a minute.

Steve: The concept. An artist did the cover. But I came up with the idea.

Leo: It's a kid looking through a telescope at the world.

Steve: Well, and remember the picture, that painting in the Sistine Chapel where there's a human sort of with an outreached hand, and God has reached down and is sort of touching fingers. Well, so that's what's in the - up in the moon or whatever it is he's looking at, is sort of that interaction.

Leo: Oh, it is, it's the Sistine ceiling. I just noticed that.

Steve: Yeah.

Leo: Yeah. Well, that's cool.

Steve: But anyway, that was my idea.

Leo: It is in paperback, $13.45, available from Amazon right now by Jennifer Horsman - H-o-r-s-m-a-n - illustrated by Julie Leimann Weaver, "Is God Real or Pretend?"

Steve: And Jenny also said, she said, "Steve, there's also been a spike in my other books."

Leo: I didn't know she wrote books.

Steve: Oh, she was a bestselling author.

Leo: What?

Steve: She was doing romance novels.

Leo: Oh, "Forever and a Lifetime." Holy moly. Did you do the cover on this?

Steve: No, no, they're five-star - they're, like, when she and I got back together, she sort of mentioned that. And I said, what?

[Talking simultaneously]

Steve: No, she had Fabio on the cover of one of them.

Leo: It looks like Fabio. It does.

Steve: No, it was.

Leo: It is Fabio.

Steve: She had THE Fabio was on the cover of one. Anyway, apparently there has been a spike in the sale of those, which are, now, they're out of print, but they are available on Kindle also.

Leo: She also wrote the "Vegetarian Weight Loss Plan."

Steve: She did.

Leo: But "A Kiss in the Night," "Magic Embrace," "The Ice Queen: A Christmas Romance," "Passion's Joy." Wow. And "Please Don't Eat the Animals." "Virgin Star," "Awaken My Fire," "Passion Flower." Holy moly.

Steve: But apparently they're good. I have not, I have to confess, I have not read any of them. But I was curious, so I went...

Leo: You know, just a word of advice, man to man, don't read them because, if you do, then you have to give an opinion.

Steve: Yeah, well, other people have. And they're, like, breathless, like this is the best, you know, when are you going to write more? I mean, people are just going nuts over her romance novels. So I said, well, yeah.

Leo: And she's all yours. Well, are you a lucky guy or what? Does she wear bodices?

Steve: What?

Leo: Nothing. Moving on. All right. Let's go. We've got questions. You've got answers; right?

Steve: Yep.

Leo: Yep [nonsense riff]. These probably have a lot to do with SQRL.

Steve: They all do. In fact, we can skip the first one because he was just asking, he says he loves the SQRL idea, but he doesn't have a smartphone. So we've covered that. You will be able to use desktop clients. Oh, and other advantage of the desktop client, because people have asked about browser plugins to do SQRL, well, first of all, browser plugins are kind of scary because they're in the browser, and you wonder about the browser's security.

Leo: Anytime it's a binary ball, you've got to worry about it.

Steve: Well, and the beauty of this is, since we'll register the sqrl:// scheme, then you install one client, and then all of the different browsers in your machine get to share it. So you have the advantage of it being outside the browser, separately; and, if you have Windows, written probably by me in assembler, and signed, and done. Or in other platforms for Windows and Wine, and I'm sure people will do one for Mac, too. But the advantage is - and it's cross-browser, and everybody gets to share that one.

Leo: Cool. All right. Well, then, we'll go to, well, credit, by the way, actually that wasn't any one person. Many users, it says.

Steve: It was many users, yes.

Leo: So we'll go to Hojune Kim with a question about usability: I heard you talk about SQRL. While my interest is piqued, I do have a question about its usability. If the master key is heavily encrypted on the device, wouldn't the users have to decrypt the master key every time they want to authenticate? If this is so, SQRL would never be as usable as you describe it. The user experience wouldn't be just scan a QR code and be done with it. It would be scan a QR code, type an ultra-strong password on the phone. Pain in the butt; right? Only then would they be logged in. So no smoother than using Keepass or LastPass on mobile. Have you thought of this? Thanks. Of course you have. Of course you have. What's your answer?

Steve: Okay. Of course. And you and I haven't, in our start of this, discussed the question of passwords. That's the other note. If you think about it, nowhere in what you and I talked about for the first half hour did I mention passwords because the system doesn't need a password. That is, the security of the SQRL system itself is perfect just using public key crypto. No password. But the security of your phone we know is not perfect. And so what we need is we need a way of having the SQRL app in your phone authenticate you. It has no problem authenticating itself to all the websites in the world. And people are very casual with their phones. It's like, hey, can I borrow your phone for a second? You hand them to people. Kids play with their parents' phones. And so we need to lock access to the app itself in a way which is strong, however strong the user wants. So, I mean, they could define a weak password, if they chose to, for their phone, I mean, and understand the consequences of using a weak password.

Leo: So what you're basically saying is you secure the phone, and then the app is on the phone, and so that's your choice is how you secure the phone. And if you've got an iPhone 5, for instance, you're in luck because you could use the fingerprint scanner, and that's pretty good.

Steve: Well, see, and you just - right. And you said "pretty good." Is it good enough? Maybe it's good enough. It's really up to the user how complex a password they want to use. No matter what your password complexity, we use a password-based key derivation function, PBKDF. We use a memory hard one, Scrypt, so that it is immune to acceleration by GPUs and FPGA arrays. So that when you enter your password, it has to think about it for one second. That's not long. But the beauty of that is, it is an algorithm to decrypt your password that cannot be sped up, that takes one second per guess. So even if a bad guy breached your phone security and got the contents of your phone, your master key

would be encrypted by a technology that resists acceleration so that they would be limited to one guess per second.

Now, unfortunately, if you used "himom" or "monkey" or something as your password, well, it's going to be on a password list, and it may still not take them that many seconds, even at one guess per second. So you'd still want to use a good password. But so we basically, we're solving the problem as much as we can. It's up to the user to decide, like, based on their environment. If it's a school teacher, and she hands her phone around to the kids in her classroom every day, well, that's a very insecure use of her phone. If you're in a bunker in some way that your phone never - you would have never to worry about your phone being compromised, then you could back off on how strong you want your particular SQRL password to be.

Leo: So if I put it on an iPhone, like this is the iPhone 5s, I currently have it so that when I put my fingerprint on it, it unlocks it. And but you could add a long password to that. Can you add - could you - I guess there's no app, so we don't know yet. But you could, in theory, as LastPass does, add an additional password to unlock the SQRL app; right?

Steve: Well, yes. Oh, and that's exactly the idea is - and so we've had a couple different ideas.

Leo: Because I choose, for instance, I keep my password in LastPass so I don't have to reenter it every time. But I put a PIN in LastPass. So you'd have to unlock my phone, and you'd have to know my PIN on LastPass, and then you could get my passwords. And I consider that sufficient, but it isn't the only way to do it. You could make it much more secure, if you wish.

Steve: Remember that the difference here is we're using LastPass because we have given up. We're using LastPass as a database of all the different passwords we use. This, if this were adopted, ends that. You only have one password to prove who you are to the app. And then it handles across the entire Internet worth of authentication for you with that one password. So what's different here is one password is all you need.

Now, another thing we're considering that will probably be a feature of the app is when you first use it, for example, after unlocking it, you would have to enter your entire long secure password which protects you against hackers getting your phone. That's its purpose is, I mean, because if someone's guessing on your keyboard, well, there'll be a five-guess lockout. Obviously it's going to say, sorry, and wipe the key so that no one can access it. So the idea would be the first time you use it after unlocking it, you have to say this is really me and enter a long password. But then, if you haven't used it for some length of time, or you've switched back to the app, or who knows what rule we could have, then you only have to enter the first four characters of your super long password in order to say, yes, it's still me who wants to authenticate now to the site.

The point is you're giving it a lot of power. It has the power to instantly log you into every site where you are using it as your authentication. So that does need to be protected. And it's still the case that something you know is the best protection. Remember that the authorities can force you to put your thumb on your iPhone 5s Unlock button. They cannot force you to give a password. That's considered testimony.

Leo: Isn't that funny?

Steve: It's self-incriminating testimony against yourself.

Leo: Yeah. I thought that was quite interesting, yeah. Okay. Moving on, Clay Cross with a tweet. He says: Why does SQRL use domain names to generate the private key? Wouldn't using a SQRL ID be better, like using a hashtag versus a domain name? Like a keyword, an AOL keyword. Somebody in the chatroom asked the same question, or a similar question: What happens if a website changes its domain name? Doesn't this screw it up?

Steve: Right. So, and I think we actually have that question later on...

Leo: Oh, sorry.

Steve: ...because that has come up. But that's a great question. The reason is that that is the way the site identifies itself to the Internet. I mean, it's relying on DNS. We know, I mean, DNS security is something that a lot of attention has been given to. We've talked about DNS spoofing. I've got my whole DNS spoofability thing. Dan Kaminsky famously found that there was like a lack of entropy in the way that ports were being used on DNS servers. I mean, so DNS has a long huge history of, like, we understand the security of it. It is generally extremely secure. So that's the thing that cannot be changed.

If a website made up an ID, like for themselves, then some other evil website could use Amazon's ID, and you would be generating your authentication for Amazon and giving it to the evil website, which it could then use to log on, to impersonate you to Amazon. So it is we bind - the way to talk about this is that the user's identity presented to a website is bound tightly to that website's name. There's a one-to-one binding between the user's identity and the domain name so that any change to it changes their identity. So that brings up the next problem, and we'll skip over that question when we come to it because we've already asked it in the chatroom, and that is, what if a site changes its name? Well, that's a problem.

Leo: Well, you just have to reauthenticate with the new site.

Steve: That's what you would have to do.

Leo: Like if you changed your name. I mean, you've got to...

Steve: Yeah. So normally, if a site wanted to change its domain name, it's because it's come up with a better one.

Leo: Right.

Steve: And so first of all, notice that no one ever does. I mean, Amazon is never going to change...

Leo: It's rare, yeah.

Steve: Because of the huge reputation cost of changing that name. But typically they could hold onto the old one and start using the new one preferentially. Now, in that case, if someone authenticated to their new domain, they would say, oh, we've never seen you before because you would be unknown under that new domain name. And so but they would know they're in transition, and they would say, if you have an account with us, please scan this SQRL code so we can link you. And so they would present the SQRL code for the old domain. You would scan it, and they'd go, okay, and they'd just transfer your identity over. So all of these problems can be solved. I mean, there are problems created by the simplicity of this, but that's also what makes it so robust.

Leo: Yeah. I don't think these are - these are not intractable problems. These are completely...

Steve: No, they're not showstoppers.

Leo: Not at all. @jmwhitty on Twitter tweeted: For SQRL, why not just add a server-signing component to prevent evil site session - let me say that again - evil site session jacking. And it says: Steve, talk all about the antispoofing, antiphishing work.

Steve: So a lot of time has been given to the question of phishing sites because, for example, you could go to EvilSite.com, and it could say hi, and present you with a SQRL code, saying this is the code to log on. But it could be showing you the SQRL code from Amazon. That is, when you went to their page, it could have gone to Amazon and pretended to want to log on. Amazon would have given it a logon page with a SQRL code, which it in turn shows to you. So when you scan that code, you're actually authenticating yourself to Amazon, but you're authenticating the login session that the evil site started.

So we solve that problem by showing the user on the screen of their smartphone or on the user interface of their app, if they're using a desktop or laptop, we show them, you are about to authenticate to www.amazon.com, meaning you're about to provide your credentials for Amazon.com. Well, you'd say wait a second, I'm at EvilSite.com. I'm not giving EvilSite.com my credentials. So we do need to show the user the domain name in the SQRL code because it's a QR code. People can't read those. Machines can read them. So we show that to confirm this is the site they think they're on that they intend to authenticate to, and then they move on.

Now, the other possibility is that you're at Amazin.com, that is, it's the phishing case where you believe you're at Amazon, but you're actually not. So there are two things that we're able to do. One of the other possibilities is, see, what we want to do to block phishing attacks is we want not to allow a phishing site or a man-in-the-middle site to acquire our credentials. Now, I'll note that going to Amazin.com or Amazon.cn or something, that presents you with a perfectly normal looking Amazon.com page where you type in your username and password, this is a standard phishing attack. So it would

be nice if SQRL had some way to defeat that, but it's sort of - it's, like, not our problem. I mean, it's like, well, okay, this is a problem with logging into a site that pretends to be something that it isn't. This is a problem we've always had. It turns out that we actually do get some leverage, though, because what we can do is we can - what we want to prevent is this evil site session from being logged in, so that it's logging in as us.

Well, we can ask Amazon to return to the client a logon link, that is, a link to a logged on session, rather than logging on the session that's displayed the SQRL code, so essentially cutting the spoofing site out of the loop completely, cutting this phishing site out. We then click on the link in the app, and it gives us a logged-on Amazon page. And so it actually is complete protection against phishing for the first time. And this is all being written up on the site. So I know that it's - I'm running it out quickly because I'm looking at the clock, and we're not making much progress here.

Leo: That's all right. That's all right. I think this is also, for some people, difficult to understand. And so a lot of the questions that are coming up are questions that come from really mostly like, I don't get it. As opposed to you, you know. So we can - I don't want to give them short shrift, but read up. Read more. Go to GRC.com/sqrl. All will become clear, Grasshopper.

Steve: Yup.

Leo: Rich Baldry wonders about poor server implementations. He says: SQRL's a great idea, sounds pretty sound, and I like how you've already started to collate potential attacks or weaknesses to get everything out on the table. And by the way, I do want to say this, and I'm sure people know this, but I'll say it again, the whole point of this, in going public with this, is to have it be vetted.

Steve: Yes.

Leo: You're not saying, oh, this is the answer, it's done. You're saying, hey, look at this, security gurus, experts. Take a look. Bang on this. Let's find out what's wrong with it. So you're certainly not, you're not saying, hey, it's perfect. We want to find what's wrong with it. If there is.

Steve: It's one week ago. It survived the weekend, where some crypto guys looked at it before last week's podcast. It has survived the last week of, I mean, we're getting 12,000 hits a day on the SQRL page at GRC. So it's come to people's attention.

Leo: So, as an example, Rich says, one weakness I don't think you've covered, risk of a poor server-side implementation of a random number generator, as you've discussed often. You can trust the math, but the implementation may not be so good. I can see in extreme cases that could create QR code collisions which could mess up the login process. Would it also allow an attacker to analyze the encrypted data sent by the client and potentially discover the private key of users? Would it even be a route for a malicious site with a specifically designed random number generator to attack the system? Also, why does the site-specific key need to be

derived from a user's master key and the domain? Oh, this is kind of like that previous question. Why not just use another random value for that key? If there is no need for the site-specific identities to be related, why create this relation? Well, it is. That's the whole point.

Steve: Yes. And so I'll just finish that last point first. What Rich is sort of saying, and a number of people have asked this question in different ways, well, why not just make up a random key for every site you visit? And then the problem is you have a database of random keys that you are required to never lose or forget. The beauty of this is everything derives from one master key. And I don't know if you picked up on this, but we have a way of exporting those from, like, out of phones into - by using a QR code. So, Leo, if you started to use it, you could have the phones face each other...

Leo: Oh, that's good.

Steve: ...and transfer the master key from phone to phone.

Leo: Here's my master keys. Oh, that's great, yeah.

Steve: Yes. And also printed out on paper so that you're able to physically back up your identity, put it in a safety deposit box, put it somewhere safe, so that if anything ever happened, if the phone got lost, well, you just install your master key into a replacement phone.

Leo: You know, this reminds me of SuperGenPass, which I've used for years, which hashes the website URL with your master password and creates a unique password for every site you visit that can easily be recreated.

Steve: Yes. In fact, it's funny because I remembered you talking about this after I - when I was, like, deep in the documentation. I thought, you know, I remember Leo talking about something, I didn't remember what the name of it was, that did use...

Leo: I've used it for years.

Steve: Yup. And so basically we're talking about that tied to - where instead of a password - see, the problem with a password is a password does not have enough entropy. And we're coming to a question about that in a second, which is really interesting. But to answer Rick's question here, or Rich's question, random number generation on the server. Remember that there's a so-called "nonce," an n-o-n-c-e, which is crypto speak for a number used once, nonce stands for number once, where the idea is there's a random value in this QR code which is what we're signing and sending back to the server to prove that we have the private key that matches the public key that we have given the server as our identity. The identity is our public key. The signature is our proof that we're the owners of the matching private key.

So it turns out that the requirement for randomness is not very extreme. For example, say that a really, really bad server implementation never changed the nonce, it used 12345678910; okay? So the risk is a replay attack because every time we signed that, the signature would be the same. So even though we're using SSL to send that back to the server, an employee could capture it. Someone maybe could hack somehow like an SSL proxy, man-in-the-middle attack of some sort. I mean, again, my point is that the only danger is of a replay. So it behooves the server to change that every time.

But in the spec we're working on, and it's called "Implementation Details," and there's a lot of it already there on the website that wasn't there last week, we, the client, add our own nonce, as well, specifically to protect against this case. So if a poor server or even a completely broken server were giving us the same SQRL code every time, we're appending our own nonce, which will not repeat, in order to guarantee that the signature we're sending is different every time. So we've solved that.

Leo: In other words, it doesn't have to be that robust. The requirements [indiscernible] that necessary.

Steve: Correct. It actually isn't - there's not a huge problem, if it just used an incrementing counter. It just has to be different.

Leo: Right, yeah.

Steve: Its predictability is not a problem. It just has to not be the same thing we signed before, or an older signature could be used as a current signature. By using a counter, that's prevented.

Leo: Prevents replay attacks. All right, Steve. Our next question comes from Mr. Tickle [laughing]. I hope he's somebody's boss, and that that person says, "Yes, I have to ask Mr. Tickle if I can do this." Steve, I wonder - he's got to be British; right? I wonder if you considered implications of this SQRL with rapidly advancing augmented reality technology. Maybe Daniel Tickle is his Second Life handle. Mine is - you know what mine is? Pruneface Spatula. I'm not kidding. I wish I were. While phones and tablets and such will be very handy, with upcoming devices like Google Glass and Meta (Spaceglasses.com), this would be a perfect method of authentication when dealing with such a hands-off technology. Oh, I like this. You look at a web page, you look at a QR code, and now you're logged in. You don't even have to pull out the phone. You just look. I'm here. It's a wonderful idea. I hope SQRL gets the attention it needs and deserves. Had you thought about that? That's great.

Steve: I just thought that was a cool idea. You're right, it's a - yeah, yeah.

Leo: I can see people doing that. Kehnin Dyer wonders about master password one minute unnecessary. I don't know what that means. Let's read. Why, he says, is it necessary to have offline verification of proper decryption of the master QR code to get your master secret? In my mind, silently failing, that is, generating a real, but

not YOUR, master code is a much better option. The only way to verify the key was properly imported in this case is to try to log into a website that has your credentials. If it fails, well, that's not it. This makes it an online attack. It also makes having your QR master code no better than random guessing master codes to someone without your password. You're going to have to figure out what that means. I have no idea what he's talking about.

Steve: Oh, fortunately I do.

Leo: Good [laughing].

Steve: So what we decided was that, if you export this super-secret master key in any fashion out of the app, it needs to be deeply encrypted because bad guys could get it.

Leo: Right. That's a weakness.

Steve: Yes. And so the idea is that we use this PBKDF function with Scrypt where we require a second of authentication that is processing, essentially, decrypting a second of decryption when you type it into your phone. To encrypt and then decrypt an exported version is 60 seconds.

Leo: Oh, that slowdown. I get it.

Steve: Yes, yes.

Leo: I get it.

Steve: So that any time that code is outside of your phone, if you've printed it out, it will require 60 seconds of constant crunching per guess for a single password entry to be decrypted and then turned into your code.

Leo: That really effectively prevents brute-forcing.

Steve: It completely does. It makes it absolutely impractical. And there's no way around it. But part of what is exported is a check to see if you've entered the right password. So what this questioner is asking is, well, the check is what allows it to be brute-forced, even though it takes a minute, he says. And he's saying why even bother with that? Why take a minute? Why not just use whatever password the user puts in? Technically you could. That is, the password is mixed with a secret key, and that generates the proper identity. But from a usability standpoint, in terms of like regular moms and dads using this thing, they would enter a password. And if they made a mistake in entering it, if there wasn't the ability to check the password, then it would say, okay, fine, but then the site wouldn't know who they were.

Leo: It just wouldn't work. Or it would think you're somebody else.

Steve: It would be, yeah, you would be someone else. And so we decided we have to verify the password for usability. But an option, when you export the password, is not to include the verifier. Then it's on you. If you're someone who you don't want the verifier in there, it's like, okay, that's fine. But you'd better enter it right, or it's not going to work. Now, there's one problem, is there's another way to verify. Kehnin was saying that it had to be an online attack. It turns out that's not the case. If you've got any authentication for a user, their use of SQRL on any website, then you've got that site's domain name, and you've got their identifier, their public key for that site. That's all you'd need to run an attack, trying every possible password to see if, for that domain, you get the proper public key. So there is an offline brute-forcing attack that does not require you to actually use the Internet except just once to capture one authentication for one website.

So we've come to the conclusion that the right thing to do is just deeply encrypt this so that it takes 60 seconds to decrypt it, and let the user know, whoops, sorry, that was the wrong password, we'll tell you a minute from now, and then try again. I think we found the right set of interactive, like, tradeoffs.

Leo: Sounds good. You've obviously thought about this a lot.

Steve: It's all I've been doing, Leo.

Leo: Murray McEwan wonders about online forum user culpability and the anonymous poster, to wit: If this SQRL makes personal identification unnecessary, would this have the effect of increasing slanderous and hostile postings by anonymous users who feel they are safely hidden? I don't even want to go on. I would think that a scrupulous forum manager would still want traceability of users who post on his forum, and this means some sort of account set-up of the forum posters would be required. Yes, that's right. Also, hey, can SQRL help defeat spambots and spammers who want to sign up on forums?

Steve: So this has actually come up a lot.

Leo: No, it's good to answer the question. I agree.

Steve: Yeah. It has come up a lot.

Leo: It's kind of obvious, but okay.

Steve: Yes, well, but the idea is that I'd promoted it last week, and even in the teasing weeks before, as anonymous. I mean, it is.

Leo: Well, it allows anonymity. But it's not a requisite. I mean...

Steve: Correct, correct. So, for example, it's a token that never changes that represents a user. A forum could require nothing but it, or they could still require an email address loop confirmation, or, more probable, a CAPTCHA. I mean, you might still require a CAPTCHA. Or you might use a CAPTCHA just once per ID, per SQRL ID. And as long as it's not abused, as long as there are not too many incoming posts, then you would, like, not require a CAPTCHA every time. What this does is it provides a secure assertion of who you are to a website. What they choose to do with it is up to them.

Leo: It doesn't change any of the stuff that a website would normally do. It could, or it doesn't have to. Just as Facebook Connect, same thing. I mean, this is all - yeah. This is not new stuff. Now, I do find the spam question interesting. Is it possible, it would be, wouldn't it, to robotically generate these logins?

Steve: Yes. So it does, yes, it does nothing to defeat spam or spammers. You could just invent keys and just come in as a billion different individual people.

Leo: In fact, that might be a little bit of an argument against it because it would in fact make it easier to automate mass logins. You could log into a site thousands and thousands of times. A site would still have to implement some sort of antispam technology.

Steve: Well, yeah. And remember that the first thing that's going to happen is it's going to say, I don't know you. And so you then say, oh, crap, so you need to know me before I can post something.

Leo: Right. But that's up to the site to do.

Steve: Correct.

Leo: Obviously, yeah.

Steve: So right now, if a site had no login requirement, spambots could go crazy.

Leo: Anyway, right. It just means they can go crazy faster.

Steve: Actually, it doesn't change the speed of crazy.

Leo: No. They're still pretty fast, yeah.

Steve: They can go just as fast crazy.

Leo: The speed of crazy. What is the speed of crazy, after all?

Steve: We answered No. 9 already.

Leo: Changing domains, okay. So let's go to No. 10, John in Illinois. He wonders about not saving the key: What if the SQRL password on the phone was used to generate the key each time so that there would be no key for an attacker to get? Multiple people could use the same SQRL app under the same device but have different SQRL passwords every time. Also, every password entered would generate a QR code. The attacker would not get a failure message and would have to try each QR code on the website. Is this a good addition to your excellent idea? What say you?

Steve: So, okay. So to paraphrase that, John is suggesting that the password that the user chooses is their identity for a given site. So I choose a password. That is hashed with the domain name to create the key. Sort of interesting, if it was impossible for two people ever to use the same password.

Leo: Which of course is ridiculous because...

Steve: And that's the problem. We already know most people use "monkey," having followed your excellent example. And so...

Leo: Hey, I use "monkey123," get it straight.

Steve: So we obviously need more entropy because what we don't want is a collision. And as I mentioned at the top of this, the "birthday attack" is what this has been famously named because there's only 365 birthdays possible. And I think it's, what is it, is it 23 is the number?

Leo: It's a very low number.

Steve: Surprisingly low number where...

Leo: But that's not - so it's not like, if you say, hey, whose birthday is November 29, that's not what we're talking about. Two people in the room having the same birthday, it needs only be around 23 people before it's almost certain that two of them will have the same birthday. But you don't get to say which birthday it is.

Steve: Yes, exactly. Nor...

Leo: That's kind of a critical thing to understand.

Steve: Yes. But that is this problem because, for example, how many people would it take before two of them might have the same password?

Leo: Right. And it's a surprisingly low number.

Steve: Probably five.

Leo: It's not 365, yeah, right.

Steve: Yeah. Okay. So our question that we're posing is we're using a pseudorandomly generated master key. And in my proposal, in the case of a smartphone, we use entropy from the platform, from iOS or from Android. But I also have the person wave the phone around, and we stream data in from the camera.

Leo: Brilliant. Love this.

Steve: Yes.

Leo: And this is just a reference implementation. Others could do something different; right?

Steve: Yeah.

Leo: But this is a nice idea. I like it.

Steve: Yeah, because it absolutely frees you from the possible lack of entropy or NSA involvement in the pseudorandom number generator in the phone and generates the key. So yesterday, preparing for the podcast, I posed a question to my gang, fabulous group, over in the newsgroup. And I said, "Gang, would someone like to do some birthday collision math? What we really want to know is, for a given number of people, what is the statistical chance that any two of them have the same public key identity for any single website, which would be an identity collision at Amazon or Facebook or whatever, which is the same as any two having the same identity master key, since both are 256 bits." I said, "As we know, assuming 2^256 randomly distributed master key identities," okay, so 2^256, that's how many bits - there's 256 bits in the user's super-secret, like, grand master key. Okay, well, that ends up being 116 times 10^75.

Leo: And that's because the key is so long.

Steve: Well, because it's 256 bits. How many...

Leo: Yeah. The birthday thing, there's only 365 chances, possibilities, 366 with leap year.

Steve: Right, right, right. So 2^256 is equal to 116 times 10^75.

Leo: It's good.

Steve: So that's, okay...

Leo: That's a big number.

Steve: ...116 and 75 zeroes. It's huge.

Leo: It's more people than have been alive in the lifetime of the world and then some.

Steve: Well, okay. So I said, "That many total possible public key identities," I said, "as I understand it, a rough estimate of the number of people required for there to be a 50% chance of collision is the square root of the size of the total key space."

Leo: That's right, yeah.

Steve: In our case, that's simple, since the square root of 2^256 is 2^128.

Leo: Yeah.

Steve: I said, "So in order for there to be a 50% chance of two people having the same identity master key, we would need to have 2^128 users on a single website. So that's 340 times 10^36 people, all using this system on a single website. And then there's be a 50% chance of a collision."

Leo: That's only a 50% chance.

Steve: I know. Exactly. I said, "What I think would be very useful would be to see the collision probability calculation for the Earth's current population."

Leo: Yeah. If all seven billion people logged into the same website, what's the

chance of a collision?

Steve: Yes.

Leo: Pretty low, I guess.

Steve: If everyone on Earth were using SQRL, what would be the chance that any two would have the same identity master key? So one of our great contributors, Sam, posted. With a 2^256 key size and seven billion keys, he used the same population figure you just cited, the odds of a collision are one in 4.73 times 10^57.

Leo: I don't even know.

Steve: The entire population of the earth, one in 4.73 times 10^57. I'm sure there's the same...

Leo: That's sufficiently unlikely.

Steve: ...chance of a huge asteroid...

Leo: Oh, much bigger.

Steve: ...hitting within the next five seconds.

Leo: Oh, much bigger, yeah.

Steve: Than that.

Leo: I think we're safe.

Steve: So then...

Leo: I love that.

Steve: ...Taylor jumped in, and he said - oh, somebody responded, how many people will have to sign up for there to be one collision? Just wondering from the perspective of a non-math genius. And Taylor, who's a very crypto-savvy person, said for there to definitely be a collision, definitely, 2^256 plus 1; right?

Leo: Yeah. Right.

Steve: Because every - you could have 2^256...

Leo: They'd all have to sign up, yeah.

Steve: ...all with a separate key, and they - one more...

Leo: That one extra guy puts you right over the top.

Steve: Got to be, yep. Now that's the guarantee. And he cited a Wikipedia article called "The Pigeonhole Principle."

Leo: Well, it just makes sense that's how many possible choices there are. For 100% certainty you'd have to have one more than the total number of possible.

Steve: Now, not being satisfied...

Leo: In other words, you'd have to have 366 people in the room to guarantee a birthday collision.

Steve: Precisely. Exactly. But not being satisfied with that, he said we will never - because Taylor is nothing if not thorough - we will never have to worry about that, though.

Leo: Why not?

Steve: Since you can't fit 2^256 people within a light lifetime of each other.

Leo: What's a light lifetime?

Steve: Well, remember this information cannot travel faster than the speed of light.

Leo: Right.

Steve: So information is the limiting factor when you have a huge ball of people.

Leo: There's just not enough bandwidth to do this.

Steve: Well, light can't get from one part of the diameter of the people ball to the other. So Assumption No. 1: A person occupies one meter cubed, one cubic meter of space. Assumption No. 2: The speed of light is 299,792,458 meters per second.

Leo: Yes, we know this, yes.

Steve: Okay.

Leo: Thanks to Michelson and Morley. Okay, go ahead.

Steve: So, yeah, just short of three whatever it is.

Leo: Three thousand kilometers per second, okay.

Steve: Yes. So 2^256 people would then occupy 2^256 cubic meters of space.

Leo: Yes. That's the people ball.

Steve: And if you arranged them in a sphere, its diameter would be...

Leo: Which, by the way, is the most compact way to arrange them.

Steve: All this ball of people.

Leo: It's not random. That's...

Steve: Big.

Leo: Yeah, sphere.

Steve: Big mother ball of people.

Leo: Yeah.

Steve: So that would be three times 10^25 meters diameter.

Leo: Okay. Got you so far [laughing].

Steve: It would take light, let's see, I didn't even put commas in here, so looks like 3 million, 198 thousand, 109 - wait, no, 3 billion.

Leo: Three billion.

Steve: Three billion, 198 million, 179 thousand, 120 years...

Leo: To cross the people ball.

Steve: ...to traverse from one side to the other.

Leo: Okay.

Steve: Even then, it wouldn't matter. He's really going to take this all the way. Since the people in the center of the sphere would be undergoing a nuclear fusion reaction...

Leo: There's enough mass now to actually create a sun.

Steve: [Indiscernible] components turned into heavier elements. This is why it's fun to be in the GRC newsgroups.

Leo: Taylor, by the way, you got the job at Google. They're going to be on the phone to you right now. That's a good one. That's...

Steve: He says, I don't think they will care about their SQRL ID.

Leo: Because they're now thorium.

Steve: Taylor is at Defuse.ca, if anyone is interested, D-e-f-u-s-e dot c-a.

Leo: Oh, that's the Defuse guy. Oh, he's great.

Steve: Yeah.

Leo: Oh, that's...

Steve: That's Taylor.

Leo: Yeah.

Steve: He's wonderful.

Leo: Defuse.ca. Yeah, we've talked about him before. Oh, Steve. What?

Steve: He's firexware, is his handle, but he unmasked himself a while ago as Taylor.

Leo: That is - Taylor, well done.

Steve: And so are we, my friend.

Leo: We are well done. Well and thoroughly done. What a fun show. What an interesting idea. Again, the whole purpose of this, of making it public, I know Steve would say this, is to get these arrows shooting at him, shooting at not him, but the idea, so that we can validate it. And some arrows have more wood than others. But pretty cool. I'm very encouraged.

Steve: People have worried about what if I lost it, what if it got loose. It's like, yes, that's a problem.

Leo: But that's a problem with everything.

Steve: Well, it is. And, see, if there was an intermediary, if there was a third-party, you could call them and say oh, I lost my SQRL ID. Please cancel it. Well, that would be convenient. But it would also represent a huge liability for the NSA, who says - or I mean an opportunity for the NSA, a liability for the user, saying we want to know every time this person logs in somewhere. So part of the fantastic benefit of this is it is one to one, and it is cryptographically secure. There is some responsibility. What I said in that original page was who do you want to have be responsible? You could farm that out. You could outsource the responsibility.

Leo: No.

Steve: But post-Snowden, I don't think anybody wants to do that.

Leo: That's the reason we need this. We've got - that's been solved, if that's what you care about. But this is something that doesn't have that as an issue.

Steve: And with that comes some responsibility. My solution is to empower the user with all the tools they need to protect themselves, to back themselves up securely, to clone between devices, to keep people from abusing their phone. We'll give them those tools. With that comes some responsibility.

[Talking simultaneously]

Steve: ...comes really good security.

Leo: We're assuming you're grownups. This is a solution for grownups. If you want to be a kid, use Facebook Connect. It's fine. That's still...

Steve: And you don't have to use this everywhere. You could just use it on which sites supply it where you feel comfortable using it.

Leo: Well, and it's my great hope that sites will supply it. I mean, that's, I'm telling you, it all comes down to that. You've got, now, I think you're going to vet - I think we're vetting it. I think it's going to, knowing you and knowing your great group there...

Steve: The HTML5 editor of the W3C already has a dialogue open with me to talk about making it a standard.

Leo: So I think the key really is to get that implemented so that sites have that. And then they'll probably still give you Google+ and Facebook Connect and all the other ways of authenticating. But it'd be really nice if they'd add a little QR code, a little SQRL QR code, and make that be one of the options. I love it.

Steve: And my guess is people will start asking for it. They'll say, hey, add this.

Leo: I would. I would. I'll ask for it.

Steve: Yeah. We'll make it easy.

Leo: Steve Gibson is at GRC.com. If you want to participate in this conversation, GRC.com/sqrl. You can also ask questions at GRC.com/feedback. You could follow Steve on Twitter: @SGgrc. Don't email him because he doesn't want to know. Go to the website. There's forums. There's plenty of ways you can have this conversation. He wants to have this in public, as it should be. It should be...

[Talking simultaneously]

Leo: ...channel, it's got to be in public.

Steve: We do maintain old-style, because they're great, Internet network news, you know, NNTP-style forums.

Leo: Right on, yeah.

Steve: Thunderbird is a good newsreader. I use Gravity on Windows. And there's something called NewsTap for iOS devices which is very nice. If you go to GRC.com/discussions, or you can just find it under Services on the main menu, that'll show you everything you need to know about how to set up and join. We've had a whole bunch, an influx of new people, all coming onboard in the grc.sqrl newsgroup, just in the last week, who are saying, hey, I want to write some code, or I want to talk about this. So everybody's welcome.

Leo: You can get copies of this show there, as well. He's got 16Kb audio for the bandwidth-impaired. He's got transcriptions written by an actual human being, Elaine Farris, so that's really the most compact. And a lot of people like to read along while they listen. We have higher quality audio and even video available at our website, TWiT.tv/sn. And of course you always can subscribe wherever you get your podcasts, from iTunes or Zune or whatever. And in fact, do subscribe. That way you get every episode. You have a collection. This is, of all the shows we do, the one that you want to have the full, the complete set. You want all, what is it, 300 and, what is it, 400...

Steve: Four hundred twenty-five.

Leo: Hmm, 420, you want all of them because it's like having the Encyclopedia Britannica. It's just you want to have that on your shelf. And you can always go back and say, hey, what was that Honeypot Monkey thing that we - what was it? What was that?

Steve: HoneyMonkeys.

Leo: HoneyMonkeys, Episode 1, about 80 years ago.

Steve: [Sighing]

Leo: [Sighing] We do this show 11:00 a.m. Pacific, that's 18:00 UTC, on TWiT.tv every Wednesday. Please stop by, say hi. Thanks once again to Tom Merritt for filling in during my vacation. It won't happen again for another year. Or thereabouts. I'll be watching TV tomorrow night with you, Steve. Anything else we need to say? I think that's it. Thanks for joining us.

Steve: Yeah. I'll be seeing "Gravity" tonight, Sandra Bullock and George...

Leo: I hear good things. I'm dying to see this.

Steve: I'm so surprised. It looks so dumb from the previews. It's like, oh, you know...

Leo: I thought, how are they going to get two hours out of Sandra Bullock twirling in space? But apparently they did.

Steve: Exactly. It's, well, actually it's only an hour and a half, so the burden was a little bit lower. But, I mean, everybody is raving about it. So...

Leo: All the geeks love this. It's apparently great.

Steve: Jenny loves 3D, so we're going to see it in 3D, apparently. And she was on Jon Stewart a couple nights ago. And if we're to believe Jon, he really, really liked it, too.

Leo: And all the geeks are loving this. It's - I was surprised. I agree. But it's a great director, and who doesn't love Sandra Bullock and George Clooney and space?

Steve: Yes. Yeah, exactly.

Leo: All right, Steven. We'll get your review next week. We will talk then. Thanks for joining us.

Steve: Thanks, my friend.