Off The Grid

Description: After catching up with the week's news, Steve explains his goals, development process, and operation of the "Off The Grid" paper-based encryption system he developed for use in encrypting website domain names into matching secure website passwords.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-315.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-315-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 315, recorded August 25th, 2011: Off The Grid.

It's time for Security Now!, the show that covers your security and privacy online. And normally we would have Steve Gibson, the host of our show, man in charge of the Gibson Research Corporation at GRC.com, on the line via Skype. But guess what, he's right here.

Steve Gibson: Yay.

Leo: Hey. It's nice to have you here.

Steve: It's so fun. I was here briefly, of course, a month ago. I zipped up just to do the very first podcast ever made out of these brand new studios of yours. And then I thought, hey, I want to come back. And I sort of deliberately missed the big grand opening party because I wanted you all to myself.

Leo: Well, I'm really glad. And you're right, I mean, the grand opening party was fun, and you missed a nice party. But it's kind of a little more normal now; right?

Steve: Yeah, and I wanted to get a chance to hang with the crew and put faces with names. And this has been just wonderful.

Leo: Well, it's really nice to have you.

Steve: And of course this is, for me, a big show. Which is why I set myself a goal of having this project finished. I've been working on this truly since the day after we told everybody about the Password Haystacks. That was a Wednesday, of course, when we recorded. And Thursday morning I set myself up at Starbucks with the project, which was to answer the question, can I come up with a non-technology-based encryption system? Because the whole Password Haystacks deal, it answers the issue of adding entropy in order to make brute-force password-guessing much more difficult. But it doesn't solve the problem, which is really a big problem, which is the per-website password. There's still that burden that we know, especially as websites are being hacked all the time, we know that it's really important for security to use a different password for each website. And the fact is, unless there's something to help with that, it's probably really not being done that often.

Leo: No.

Steve: So I thought, okay. What about some means for encrypting the domain name into a matching password? And doing that without a computer, just because I wanted to see if I could. That way it's universal. And you can't hack paper. If it's off the 'Net, if it's out of your computer, or in this case "Off The Grid," then you're safe from anything that could possibly happen.

So about, maybe 10 days after I started the project, I thought I had something. So I wrote the following posting in - I have a newsgroup at GRC called "Thinktank," which is where I hang out and work with a group of people that are into, like, the early development phase of things. So I wrote, "Gang: I am way behind here in this newsgroup (80 unread messages) because since last week's podcast about Password Haystacks, I've been working on a closely related follow-on project that I believe everyone here will find quite interesting and intriguing.

During last week's Password Haystacks podcast, I mentioned that I thought that I probably had something more up my sleeve. It now appears that I do. For anyone who may have been wondering about that and waiting for the other shoe to drop, here's what I've been up to: I believe that I have a simple, secure, and highly practical pure paper-based keyed enciphering scheme to support the individualized encryption of Internet web domains into short (12 character) high-entropy tokens for use as corresponding login passwords for those web domains.

Such a system has advantages over any alternatives available today. Being "offline," it's secure against any possible - ANY possible - browser-based compromise of browser-generated or browser-stored passwords. Being "per domain," it solves the problem that password padding does not of still needing per-domain passwords for the best security. Being a keyed domain name enciphering system, there's no need to record, store, or remember passwords. This "Simple Paper Cipher," whatever we're calling it, can simply be used to recreate any domain's password on the fly.

Being able to generate 12 high-entropy password characters when given a domain name of any length (the user can easily make these longer or shorter, but 12 would be recommended for security), we get sufficient security without the need for additional low-entropy padding. Being paper-based, the system can be used to generate passwords

absolutely anywhere, since it doesn't rely on any technology that might not be available. So it's both universal and "cloud safe," since nothing is ever stored anywhere. And finally, being easy to use - which was my number two requirement, second only to security - the system requires very little training, yet can produce, I believe, industrial-strength encryption having hash-like features such that, for example, changing any single character of the input results in an entirely different result, and so forth.

And I wrapped up by saying: And no, I don't need any lectures about the fact that only fools invent new encryption systems. I am well aware of the pitfalls of doing so. Well aware. But I'm not trying to displace or replace AES Rijndael or any industrial-grade encryption system. I'm merely wishing to create a new alternative for users who are stuck using the same password everywhere or relying on technology, which can be a mixed blessing, to create and/or store unending numbers of per-domain passwords. And I think I've done so.

Leo: Yay.

Steve: So that's the topic. I'm going to explain the process I went through from the start that Thursday morning, the day after the Password Haystacks was revealed, and explain how the system works. And we've got a whole ton of security news, especially the most high-priority security alert we have had in a long time.

Leo: This one's a shocker. You just told me about it a second ago, and I'm kind of blown away by this one. All right, Steve Gibson. We go "Off The Grid" with Steve in just a second. Okay. Well, I guess we'll start with the security news.

Steve: We've got a bunch of stuff to cover before we get into the topic. Okay. This is mind-boggling. A security researcher named Daniel Garcia - I need everybody to pay attention.

Leo: Pay attention.

Steve: Pay attention. A security researcher named Daniel Garcia discovered that consumer routers made by Edimax, the Linksys routers, Sitecom, and Thomson, which is the SpeedTouch router...

Leo: See, I don't know the others. But Linksys I've heard of.

Steve: Linksys we know.

Leo: That's the most - that's the bestselling router in the business.

Steve: They respond to UPnP, Universal Plug & Play, requests on their WAN interfaces.

Leo: Okay. I understand what that means, but you'd better explain.

Steve: Yes. That's the side of the router which is exposed to the wild, exposed to the Internet. And it means that, if somebody were to send UPnP packets of the proper design at your public IP of your router, which is provided by your ISP, they would be able to log into your router and do the things that Universal Plug & Play is able to do, specifically mapping...

Leo: To configure the router.

Steve: Reconfigure the router, mapping ports through from the outside into your private LAN.

Leo: So open up holes in the router, or the router's firewall.

Steve: At www.toor.do is the freely downloadable scanner which has been created to allow people to scan the Internet for these.

Leo: Oh.

Steve: So, and in a short time, Daniel found 125,000 routers. I mean, Leo, this is like the open ports deal that I created ShieldsUP! to help people with, where people had their hard drives mapped onto the Internet without knowing it. So as long-time listeners of this podcast know, our advice has been disable Universal Plug & Play in your router as one of the things you do to secure yourself. So people who took that advice are safe. If your router has a provision for disabling the WAN side, and you otherwise need Universal Plug & Play, then I would say that's what you need to do. But if it won't, if it doesn't make it clear that you're able to disable the WAN side independently, turn off Universal Plug & Play.

Leo: Wow.

Steve: I mean, this is...

Leo: Well, we've recommended that for years, that that's what you should do.

Steve: Yes.

Leo: But now there's a really compelling reason. In fact, I've always turned off WAN routing, as well, because I don't see any reason for somebody outside of my network to be able to administer my router.

Steve: No, oh, you mean WAN-side administration.

Leo: WAN-side administration.

Steve: You absolutely want that turned off, not just password-protected, because we all know that there are problems with relying on passwords to protect something. If there's no reason...

Leo: And there may be backdoors. If there's no reason - and most people, why would you want to administer from remote?

Steve: Right. Normally you're only logging into your router from your LAN side in order to do whatever, set up its password and perform configuration. Now, gamers will be probably the largest group that have Universal Plug & Play enabled. Or, for example, Skype users, because Skype is now UPnP aware, and it's able to perform port mapping if it's available. So the problem is, it has never been secure. It was not very well thought out. And what we've now found is that hundreds of thousands of routers on the 'Net are exposing their UPnP interface.

And, I mean, right now there are no doubt hackers who are getting ready to have some fun, firing this up. On h-online.com is the article titled "UPnP-enabled routers allow attacks on LANs," which just came out. So I'm sure people will be able to find it. We've got the link in our show notes. And if you click that image to zoom in on that image, Leo, you'll see a sample of his scanner, which is just scanning through a chunk of IP space on the Internet, finding router after router which is wide open because Universal Plug & Play allows you, it's designed to be a software-based, a packet-based administration for the router. The idea that this is exposed by default in Linksys routers, among others, is...

Leo: Has to be a mistake.

Steve: ...unconscionable to me.

Leo: Has to be a mistake.

Steve: Yeah.

Leo: I do think that the problem in turning off UPnP, which is a Microsoft technology, primarily comes from the fact that Microsoft's Xbox complains.

Steve: Heavy gaming, yes.

Leo: Xbox complains. It says, no, I can't, I won't - you may not be able to start, for instance, or allow your friends to start an Xbox live session with you and so forth. So

a lot of people whose kids have Xboxes, the kids say, "Dad, Dad, Dad" - and Microsoft tells you to do this.

Steve: I know.

Leo: "Turn on UPnP, and then my Xbox will work." Otherwise it won't work very well, and that's why they turn it on.

Steve: Yes. For anybody who does, who is in that situation, it's well known that you can manually...

Leo: It's just it's a pain.

Steve: It's a pain.

Leo: Because there's three or four ports, and you've got to open them up.

Steve: Exactly. It's three or four ports that you need to manually route, manually configure on your router. Our listeners, I would say, doing that is not beyond them. So if you have - if you've got an Xbox, you're into gaming, you've got UPnP on, I would say turn it off. I mean, that's always been our advice, turn that off. And then there are plenty of resources on the 'Net that will show you how, with different routers, to map those ports that you need through. And that's really just a much more secure way to run anyway.

Leo: It's not hard, yeah. I mean, it really isn't hard. You just have to know how the software works. And I think it does intimidate people. I'm not saying that. It definitely intimidates people, but it's not hard. Slingbox, too, they're telling me in the chatroom, is another hardware application that wants UPnP. And it would make sense because it needs to make it easy for the consumers to open up those holes so that the Slingbox can communicate with the outside world.

Steve: Yeah. If the router allows you to disable UPnP selectively, then definitely disable it on the WAN side. I don't even know why it's there. That's nutty. I can't imagine that it serves a useful purpose to have that exposed on the WAN side, Leo. I mean...

Leo: Well, I think if you're a sysadmin - let's say our guy Russell, our IT guy, he wants to, from home, fix a problem with one of our routers.

Steve: Yeah...

Leo: But that's - I'm sure he doesn't do that because he knows, if you're a good IT guy, you know the dangers.

Steve: Okay.

Leo: So I think it's a mistake. I think these companies just left it - somehow made a mistake, that's all.

Steve: So we talked about the recent reduction in security of AES. And there's been some continuing discussion about it, as I knew there would be. Bruce Schneier, our favorite security, super-security expert...

Leo: Love him. Love him.

Steve: ...blogged recently about - he said: "This is what I wrote about AES in 2009." Now, that is when that first related key attack was found. So that was, what, three years ago, no, two years ago. And he said: "I still agree with my advice." Which was: "Cryptography is all about safety margins. If you can break 'n' rounds of a cipher, you design it with 2n or 3n rounds. What we're learning is that the safety margin of AES is much less than previously believed. And while there is no reason to scrap AES in favor of another algorithm, NST" - and he probably meant NIST - "should increase the number of rounds of all three AES variants." And he wrote this two years ago.

He said: "At this point, I suggest AES-128, which normally has 10 rounds, be extended to 16. AES using a 192-bit key, which is now 12 rounds, should be extended to 20. And AES-256, which is now at 14 rounds, should be doubled to 28 rounds. Or maybe even more; we don't want to be revising the standard again and again." And he said: "And for new applications, I suggest that people don't use AES-256. AES-128 provides more than enough safety margin for the foreseeable future. But if you're already using AES-256, there's no reason to change."

And then in talking about this story also, SANS Institute Editor William Murray, he wrote, he said: "Since no claim as to the strength of AES has ever been made, this is simply a mathematical claim that the work factor for discovering a key is about four or five times lower than a brute-force attack."

Leo: Interesting.

Steve: "While this is a significant analysis, worthy of a paper, perhaps even a headline, an attack using this information, begun at the Big Bang, would not have yet completed."

Leo: That seems enough time. I could live with that.

Steve: And he said: "Kudos to the analysts." So again, we want to keep perspective.

Leo: So a quarter of the Big Bang, big deal; right? A quarter of the life of the universe, big deal.

Steve: Yes. Exactly. So Bruce also blogged about stealing ATM PINs with a thermal camera: "Researchers from UCSD pointed thermal cameras towards plastic ATM PIN pads and metal ATM PIN pads..."

Leo: Oh, this is clever.

Steve: Yes, "...to test how effective they were at stealing PIN numbers. The thermal cams did not work at all against metal pads."

Leo: Really. Oh, because they probably dissipate the heat very quickly.

Steve: And spread it so quickly.

Leo: Right, right.

Steve: "But on plastic pads the success rate of detecting all the digits was 80 percent after 10 seconds and 60 percent after 45 seconds."

Leo: That's amazing because you really don't touch, when you're using an ATM, you touch those keys very rapidly.

Steve: Yeah.

Leo: And yet these thermal cameras must be sensitive enough to...

Steve: And just that heats them up enough.

Leo: Just enough.

Steve: And so, when they look at it, they can see the relative heat. So the one that you first touched would be cooler, a little cooler...

Leo: Slightly cooler.

Steve: ...than the one you second touched.

Leo: So you even get the order.

Steve: Yes. You get the order of them.

Leo: How far away does this thermal camera, can this thermal camera be?

Steve: I'm sure it was, like, right there. He said, "If you think about your average ATM trip, that's a pretty wide window and an embarrassingly high success rate for thieves to take advantage." So the idea being someone does their transaction. If they're sufficiently quick, they walk away, you run over and take a picture of it with a thermal camera and see if there's still some heat signature left on the PIN pad.

Leo: So you should take your time at the ATM. Or somebody's saying I never touch them anyway, I use a pen.

Steve: That'd be really good, yeah.

Leo: That'd work.

Steve: Don't want to leave your fingerprints behind, either.

Leo: Yeah, no.

Steve: We did get confirmation of Chinese cyberwarfare development aimed at the U.S.

Leo: Oh, interesting.

Steve: For the first time ever. Every time, you know, there's been lots of suspicion of it because IP addresses tend to terminate there. But everyone can say...

Leo: You can spoof it.

Steve: They have plausible deniability. They could say somebody hacked those machines, and they were looping through those machines. A Chinese military video had a few seconds of screenshot which, when uninterpreted, when translated into English, showed the IP addresses and some explicit instructions about attacking U.S. IPs.

Leo: This IP address apparently belongs to an American university.

Steve: Yup.

Leo: It says "Select attack target" in the big letters. And then the IP address to attack. That is amazing. Wow.

Steve: And the nature of where it was made it very clear that it was inadvertently left into a training video that Chinese cyberwarfare developers were using.

Leo: This looks like Linux. Doesn't look like Windows. That's really interesting.

Steve: So it's a goof. It's a goof. Since we last talked, a serious crypto bug was found and fixed in the latest release of PHP, which is already now obsolete. This was a perfect instance of why fixing security software doesn't always provide you with the result you were hoping for. Version 5.3.7 you do not want to use.

Leo: 5.3.7.

Steve: It just came out, and shortly after it came out it was discovered that anyone calling the crypt() function using an MD5 with salt, the function would only return the salt. It would give you back...

Leo: It wouldn't hash it.

Steve: It would give you back what you gave it. No matter what the rest of the stuff was you were telling it to hash, it would just ignore that and give you back the salt. Now, that was not the case with Blowfish or DES, only MD5. So it would only affect those sites that were doing it. But because it was such a problem, it was instantly fixed. And just this morning my friend Simon Zerafa, who tweets me all kinds of interesting things, and did in another case about something else today, he sent me the note that 5.3.8 was already out.

Leo: Probably in response to this.

Steve: Oh, absolutely in response. So skip over 5.3.7. Don't use it. Just go to 5.3.8 that has this problem fixed already.

And thanks to Anthony Bosio, who's @abosio in Twitter. He sent me what I thought was a pretty funny original password recovery question, which he encountered when he was creating an account for the National Archives. He wanted to download some content from the National Archives. So one of the questions was rather common: What is the name of your hometown? Or what is your mother's maiden name? Okay, we've all seen that before. What's your pet's name? No surprise there. Who was your childhood hero? I think I've actually encountered that before. But I have never before seen, "What is your preferred Internet password?"

Leo: [Laughing] This is the National Archives?

Steve: Uh-huh. It's one of the things they ask you.

Leo: "Monkey." "Monkey." I love "monkey."

Steve: Now, that would not be the password you're using here because you don't know what it is that you're using this, this password recovery option, to recover.

Leo: Well, maybe it'd be a good way to remember it.

Steve: So hopefully you did not use your preferred Internet password. And of course this podcast is all about not having a preferred Internet password.

Leo: Yes. That's right. We're going to show you how not to do that.

Steve: That's right. Now you don't need to do that.

Leo: Although you could enter "Steve's Off The Grid" as your answer to that question.

Steve: That would be good.

Leo: There you go.

Steve: Yeah. So there was...

Leo: That's funny.

Steve: ...also further good news from Twitter that was tweeted thanks to @sergeantjoe, who tweeted me that - we covered the fact back on March 15th that Twitter had added the option for always-on HTTPS, specifically in response to Firefox, which is why I felt that Firefox on balance was a good thing because it was, as it did with Facebook and has now with Twitter, it was going to push people to start bringing up SSL encryption by default.

The good news now is that Twitter has blogged that they are beginning to roll out HTTPS on by default. They're being cautious about it. They're going to be bringing it out in a few accounts to make sure that it doesn't have some unintended side effects. But they're saying it's our intention to ultimately do this. We just don't want to break anything in the meantime. So that's really good news. I mean, that's really where we want to be headed.

And our listeners know that our background topic that we will be moving forward with, I expect in two weeks we're going to tackle TCP, is we are doing the basics of How the Internet Works from the beginning.

Leo: Oh, look, my Twitter is HTTPS.

Steve: You probably set that on by - you are able to turn that on on your settings page.

Leo: Oh, I was smart enough to turn it on probably.

Steve: Yes.

Leo: I see. I see. I see.

Steve: And our listeners have done that. But the setting will be on...

Leo: By default.

Steve: ...by default in the future, which is really great news.

Leo: Right. So, yes, it is, I do remember that now, it is in the settings page, yeah.

Steve: So Brian Weeden tweeted me a note that the infamous hacker group Anonymous has been using SNMP for pulling off their DDoS attacks.

Leo: Simple Network Management Protocol.

Steve: Simple Network Management Protocol.

Leo: That's a way that businesses, for instance, can manage multiple computers, the IT department manages multiple computers remotely.

Steve: Right. It's UDP based. And we talked about ICMP and UDP in our last podcast. And on our Q&A last week someone asked about couldn't you use the Internet's devices to hide your IP addresses? If you spoofed the source IP, for example, and pinged a router, it would respond with an echo reply aimed at the apparent source IP. But the problem is it would only be responding with about the same bandwidth that you sent. So you'd have to be sending a huge amount of bandwidth out, spraying all those routers. And then they would be sending their ping replies aimed at a focus point, very much like concentrating the sun with a magnifying glass onto the palm of your hand, which doesn't

feel good. It burns you.

Leo: Right.

Steve: Okay. SNMP has the ability to be queried for all its data. There is a "Send me your entire configuration tree." SNMP is organized so that, for example, assets are in known locations. There's something called an OID. And the one that starts with 1.3.6.1 is like all the configuration information that the device has - all the interfaces, all of its IP addresses, its DNS addresses, its bandwidth, its connections. SNMP essentially allows you to query everything you can imagine about an Internet device. And it's also able to say "dump it all." So you send a single UDP packet to a router whose SNMP default password is public. That's one of the annoying things about SNMP is the default password is public.

Leo: Right.

Steve: By spec. It's like, all SNMP devices have a default password to public.

Leo: But I assume that the IT department, IT guy's going to fix that.

Steve: Well, and that's the read-only password.

Leo: Okay.

Steve: And the assumption is we don't mind...

Leo: If you read it.

Steve: ...you doing a read-only query because you're not going to change anything, you're only going to get stuff. So the point is that it turns out there are tens of thousands of pieces of equipment, I think actually the number is more like a hundred thousand plus, that will respond to a single UDP query and generate megabytes of data in reply. So what you end up with is a bandwidth amplification attack. And since it's UDP, which does not use the TCP connection setup...

Leo: ACK and NACK, right.

Steve: Exactly, ACK and SYN ACK response that inherently verifies the endpoint IPs. And we'll be talking about that in two weeks. We're going to do TCP next. This means you can spoof your source address and aim the entire SNMP dump at a target of your choice.

Leo: And it will just send it.

Steve: And that's what Anonymous has been using.

Leo: That's how they're flooding?

Steve: That's what they've been using for doing their super-potent DDoSes.

Leo: So they're going to a bunch of different machines and saying we want an SNMP dump, and sending it to BART or whatever.

Steve: Sending a single UDP off to some router, telling it this is coming from them, and the dump gets sent to them.

Leo: Wow. And it's how many megabytes? Well, I guess it depends on the size of the network.

Steve: Depends upon the size of the SNMP tree. But if you look at it, it just goes on forever.

Leo: Can be quite huge.

Steve: And it's a verbose protocol. It's a messy, verbose protocol. So it just dumps all this data out.

Leo: And of course you do this to thousands and thousands and thousands of machines...

Steve: Yup. Just send a packet every so often. So that's a biggie.

Leo: That's actually clever.

Steve: It is. It's very clever. Unfortunately, evil and diabolical. Simon Zerafa also made a note that just mentioned that Java was just updated to #27. So it's Java v6 Update 27 is where we are now. And so you'll want to check in your control panel, if you've got Java installed, might be a good thing to update that. They just did a bunch of bug fixes. It wasn't any big huge security fix. But it's always good to keep that current.

And Leo, I have confirmed, as if we weren't sure, the supreme geekiness of these listeners, our beloved Security Now! listeners.

Leo: Not a surprise.

Steve: Because the nerdiest joke of all time, which was "I could tell you a UDP joke, but you might not get it"?

Leo: Might not get it, yeah.

Steve: It was so heavily retweeted by people who thought that was funny. So it's like, okay, that's perfect.

Leo: That's so great.

Steve: I did want to mention that anyone who missed out on - this is in just total miscellany. Anyone who missed out on the $99 TouchPad...

Leo: Right. WebOS.

Steve: Don't feel badly.

Leo: You got yours already?

Steve: No. I played with one in a store. It's a piece of junk.

Leo: Oh, it's junk. No, that's right.

Steve: It's a piece of junk. For 500 bucks I would be really upset, if I paid that last week.

Leo: It's slow, it's sluggish, it's not...

Steve: That's exactly it. Just, like, scrolling the calendar, you can't, like, it waits, and then it jumps out about an inch, and then it decides to go ahead. It's, oh, my goodness. So...

Leo: Apparently, I was told that one of the problems was a lot of debugging was left on in the OS. And there are ways to go into the settings and turn off a lot of the stuff, which does make it snappier. But this was the same problem WebOS, you remember, had on the Pre. It was just sluggish. And so, yeah, I don't - we do know a lot of people ordered them. In fact, maybe as many as a million people ordered

the...

Steve: And for $99.

Leo: For a hundred bucks? Why not.

Steve: Yeah. But still, don't feel bummed. It is not the equivalent of an iPad.

Leo: There's a reason it's not selling.

Steve: Oh, goodness. And someone either wrote to me or tweeted, I wanted to mention - we were talking about the Lost Fleet series, which was a series of six books. There is a new series beginning by that author called Beyond the Frontier. The first book is "Dreadnaught." And so that has started. It's available in hardcover and Kindle and Audible, so across the board. You have to decide if you want to get sucked into a series that isn't finished yet because you'll be left hanging, waiting for the next book. But they are, I mean, I'm reading it, there's no doubt about it. It's a fun series. In fact, I'll probably do it before I switch over and do the next, that Peter Hamilton series that'll keep me tied up on my stair climber for the rest of my life.

Okay. And many people tweeted that I forgot to tell everyone where to get that really cool embedded processor that we talked about last week. It's the processor, the $30 embedded processor that I'm going to be using for the Portable Sound Blaster project. And I went jumping up and down, talking about how cool it was, and I forgot to tell everybody where to get it.

Leo: It's an ARM-based processor.

Steve: It's an ARM Cortex-3 with a complete development, all the software is free. It itself costs 30 bucks, and all you need is a USB cable to plug it into a Windows, a Mac, or a Linux machine. Anyway, it's LPCTools.com. And you're looking for what they call the LPC 1768, LPCXpresso. So it's called the LPCXpresso. And it actually has two processors. The LPC 1768 is the interface for USB. And then the LPC 1769 is the actual target processor on this little board.

Leo: You'd better hurry, they only have six left.

Steve: Yes. It actually, as I've been responding to people tweeting, giving them this URL, I've been watching...

Leo: That number is going down.

Steve: ...the count dropping down. So people are snapping them up.

Leo: This is pretty amazing. For 30 bucks you get a Cortex-M3, which is the basis for many smartphones.

Steve: Yes, 128MHz.

Leo: 128MHz, yup, 512KB of Flash, 64K of RAM, and Ethernet, 100MB Ethernet, USB - this is pretty amazing.

Steve: It's a fantastic - and all the libraries available, it'll do a 32x32 multiply in a single cycle. There's a DSP library that'll do an FFT. And, I mean, and all the software, a full - I'm trying to think of what's the open platform IDE that's so popular.

Leo: Eclipse?

Steve: Eclipse. A full Eclipse platform made for this, with the files ready to go from Code Red. It's all free. You just sign up, and you're able to get it for free. So the whole thing is $30 out the door...

Leo: That's pretty amazing.

Steve: ...to mess around. Oh, it's got something like 77 I/Os that are uncommitted, and multiple USBs. Anyway, it's a tremendous little development platform, and it's what I'm going to be using because it is also a ton of power. There is the Arduino, but that's way - it's much more - that's heavily interpreted. And then you're writing Arduino scripts. Here you're writing in C, and it compiles directly into ARM Thumb-3 code, which is what the Cortex-M3 uses natively. So it's great.

And lastly, @Boonie_NL on Twitter said, "Steve, there are not 52 weeks in a year.

Leo: What? 52.179? What?

Steve: Leo, that's exactly right.

Leo: Is it?

Steve: Are you, like, how many places of pi did you memorize?

Leo: Well, that's a little pedantic. That's a little pedantic.

Steve: So, yes. I took 365.25 and divided it by seven, and I got 52.178571. So in fact...

Leo: Wow. It was a guess. That was just a weird guess.

Steve: Damn, are you good.

Leo: Maybe I saw it, and I remembered it.

Steve: Well, I guess you did.

Leo: I don't think it's in your notes. Wow.

Steve: Anyway, very, very good. And I did want to share a fun story by a listener of ours in Oslo, Norway, Bent Heier. He said, "Hey, I'm a 17-year-old boy from Norway who loves your show."

Leo: Yay.

Steve: "It was actually what got me into liking computers."

Leo: Double yay.

Steve: "And I found it very easy to understand, even for someone with close to no prior experience with computers. I started listening to Security Now! just a couple of months ago. The first one I listened to was in the fact the one about bitcoins. Then I listened to a few more before I decided to go back to the very beginning, and now I'm at Episode 115 - likely 150 when you read this." And he says, "I love your show, and I have listened to it all summer long - on the plane, on the train, and virtually everywhere.

"Well, I downloaded SpaceMonger to see what it was, and I liked it. But on my two-year-old laptop it was unable to read approximately 20 percent of the drive. I didn't care too much about it at first as I have only used about 40 percent of the drive anyway. I like keeping it clean. But I had some problem with explorer.exe that stopped working, and it's a little hard to use a PC without that.

"This problem escalated after a while, and I decided to go into Safe mode and do a full scan. However, I'd not got into Safe mode manually before, ever. I'd just pressed the power button to shut it off and then on again. Not good for the computer, but I'm a kid. Please excuse me." And he says, "So I figured it was probably in the BIOS menu or something. It isn't, but I found some hard drive scanning tools. Remembering in the back of my head that SpaceMonger had not found my entire hard drive, I decided to run the tests. They told me that 80 percent of my disk was okay. Having 20 percent of my disk being dead was not a good thing.

"I knew what I needed: SpinRite. I had been lent a copy from a relative of a friend of mine." And he says in parens, "I'm poor." And he says, "And it worked perfectly. My disk is now 100 percent working, and it has no problems in the last couple of weeks. Now I

really want my own copy of SpinRite, but my parents are hard to convince. And since I am underage I need their permission and, more importantly, their credit card..."

Leo: Their money, yeah. Forget the permission.

Steve: "...to make online purchases. So I decided that I will get SpinRite for Christmas this year. It will be my best present ever."

Leo: Awww, he's asking for SpinRite for Christmas?

Steve: That's all he wants for Christmas...

Leo: That is nerdy.

Steve: ...is his own copy of SpinRite. So, Bent, thank you so much for your note.

Leo: Good on you, Bent. All right. Let us get right to the meat of the matter because I want to - I'm very curious about this. You do have a web page on GRC that people can go to to follow along, if they...

Steve: Well, yes. Although that shows the result. It'd be better if people sort of closed their eyes...

Leo: Listen first, okay.

Steve: Yeah. And you, too, Leo.

Leo: All right.

Steve: Okay. So...

Leo: My eyes are closed.

Steve: What I set about doing was, as I said at the top of the show, I wanted to see whether it was possible to do something without a computer. Much as I love software and computers, sometimes they're not the right solution. And depending on them can be a problem. For example, we've got the problem when Firefox is updated that plug-ins initially don't work until they get caught up. They may not be compatible. So what if you were depending upon something that wouldn't run with your new copy of your browser? That would be a problem. And how many people, for example, have something that's truly legacy, like a 33-1/3 RPM LP? Well, what are you going to play that on now? I

mean, you end up with having things - our technology does move forward. So I wanted something...

Leo: Paper is good. It doesn't change. People have been using this for a long time.

Steve: ...that was absolutely, in three or four decades from now, it would still work. So, first of all, we talked really in the early stages of our encryption series about the simple Caesar cipher, which was a substitution cipher, where you would...

Leo: Is that named after Julius Caesar?

Steve: Yeah, I think so because he was known to have used that cipher.

Leo: But that was a substitution code; right?

Steve: Yeah, a simple substitution code, the idea being that you would take the alphabet - and we'll talk about the Roman alphabet for ease of conversation, A through Z. And you would, in one case - remember the simple decoder rings would have one ring of A through Z, and then an inner ring of A through Z, and you would rotate it to a certain setting, and then your so-called plaintext would be on the outer ring, and your cipher text would be the inner ring. And so all it's doing is it's essentially sliding down by a fixed offset any character over to a different one, so that it's doing a one-for-one transposition of the characters.

Leo: It does in fact come from Julius Caesar. In fact, according to this book, which is a great book I'm sure you've read, David Kahn's "The Codebreakers," Julius Caesar himself wrote about the code in his book "Gallic Wars," you know, "Omnis Gallia in divisa partes tres," that's the book. And he talks about Cicero and a letter that needed to be delivered to Cicero, and it was enciphered in this exact method.

Steve: Yeah. Now, the problem we have with that in this day and age is that we know that the frequency of the occurrence of letters varies dramatically. There's lots of E's and T's and S's. There's very few Q's. And but we also know that the pair "QU," like in "quest," occurs often. And so if you did a frequency analysis...

Leo: Very easy, in fact...

Steve: ...of a corpus of English, you'd immediately get this standard distribution of the occurrence of letters. If you then did...

Leo: "ETAOIN SHRDLU," by the way.

Steve: There again, there you go again, Leo.

Leo: In case you're interested. That's in English, though.

Steve: Yes.

Leo: Yeah, "ETAOIN SHRDLU."

Steve: And so if you took a ciphertext that was ciphered in, in a simple substitution cipher, there would be some letter that would occur the most. And so the chances are...

Leo: That's going to be an "e."

Steve: ...that's going to be an "e." And the thing that occurs second the most is going to be a "t."

Leo: So it's easy to find "the," for instance, and that's going to give you the "h."

Steve: And suddenly you, exactly, and then you would also see - you would see three-letter groupings that had a "T," a something, and an "E." And you'd go, well, that one most be an "H." So you could imagine how simple it is to crack that.

Leo: Right.

Steve: Okay. Knowing that, a smart person back in the 1800s in England, somebody who anyone who knows electricity has heard of, named Charles Wheatstone, of the Wheatstone Bridge. The Wheatstone Bridge is four resisters connected in an "H" fashion where you have two coming down from a positive source of voltage, connected to two going down to the ground, and then you put a voltmeter at the junction of those pair. And Charles Wheatstone figured out - and actually he wasn't the originator of this. It was mentioned 10 years earlier, but no one really paid attention to it. And he sort of brought it back and publicized it, and it ended up with his name. Well, he had a good friend who lived near him named Lord Playfair.

Leo: Great name.

Steve: And he and Lord Playfair used to walk around town, talking about ciphers and puzzles and things of that ilk. Wheatstone shared with Playfair a paper-based cipher that he had invented, which Lord Playfair really got a kick out of and showed some people. And it ended up catching on. This thing was used throughout World War I by a number of militaries, and British Intelligence was known to have used it during World War II. It's called the Playfair Cipher. Now, and what's interesting is it's the Wheatstone Bridge. It should really be the Wheatstone Cipher. Except that everyone knows Wheatstone invented it, but Playfair got so excited about it, he just kept talking about it. And so everyone called it Playfair Cipher.

Leo: The Playfair; right.

Steve: So what happened was...

Leo: It's kind of funny because Microsoft's encryption, or actually Apple's encryption for - its DRM for the iTunes is called "FairPlay." And the guy who cracked it, DVD Jon, ended up making a program called PlayFair. But I don't know if he knew. He probably did. He's a smart boy.

Steve: Well, and you can Google it, and you'll find lots of information about it on Wikipedia. But I want to describe it to our listeners because it also - okay. So when I discovered it that Thursday morning, I said, oh, maybe my - I didn't want to invent something. I wanted to find something that had already been well vetted, was understood, people had looked at. And so I thought, maybe the PlayFair cipher or a variant of it was something I could use for this paper enciphering system. So the way it works is we have a 26-character Roman alphabet, A through Z, 26 characters.

Leo: So we don't allow for numbers or punctuation.

Steve: We don't do numbers and punctuation. The idea was to get messages in a secure fashion, delivered by couriers in the war, in such a fashion that, if someone intercepted the courier, they'd look at this page of gibberish, and they couldn't decipher it. What's interesting is for a long time this was believed to be uncrackable. So here it is.

You technically have a passphrase which you fill in in upper left and across, then the next row and down, order. And that's the key for this square, the 5x5 grid. So it's very small and convenient. So there would be a passphrase, and you simply fill in the letters of that passphrase. Any that occur a second time you ignore. So that basically you start filling in this square from the top down, in raster order, until you're done with the passphrase. Then in alphabetic order any letters you haven't used already you fill in till you're done.

Leo: So you then go A through Z.

Steve: Then, exactly, then you go A through Z. But for our purposes, just imagine all the letters of the alphabet in a random order occurring in there. Because actually it weakens the cipher to have it be a passphrase because it turns out...

Leo: Ironically.

Steve: Exactly, because things that are probably not in the passphrase, like "Z" and "V" and things, they're going to end up being pushed down, down to the bottom. And it'd be better if they were in an unknown location in this grid.

Leo: "Z" always tends to be at the end.

Steve: Exactly. It's always going to be the last character.

Leo: Right. Unless you have a passphrase with a "Z," but...

Steve: So what's so clever about this, this is the, as far as anyone knows, the world's first digraph cipher. The Caesar cipher is called a "monograph" cipher, meaning a single-character substitution. This is a "digraph" substitution cipher. You take the message that you want to encrypt, and you break it up into pairs of characters. So you have, whatever it is, you blow off the spacing between words, and basically you respace it in character pairs. You locate any two characters, like in the first pair, you find them on the grid. And if they form two corners of a rectangle, then the encrypted characters are simply the other two corners of the rectangle.

Leo: Oh, that is tricky.

Steve: Yeah. And if they are on the same line, then you encipher by going to the character to the right of each of them. And if you wrap off the end of the grid, you come back around. If they're in a vertical column, then you use the characters below and also wrap around. And that's it. Oh, and if you have a double-character pair, obviously if you had "AA" that occurred together, you can't, you know, it's just itself. Part of the setup is you just stick an "X," you separate a character pair with an "X" when you're doing your original grouping by twos. And anyway, so you simply write down the characters that are occurring in the opposite corners of a rectangle, or if it's same characters on a line or a row, the ones that are either to the right of them or underneath them. And that's all there is to it. Now, the...

Leo: That would be difficult to decipher, I think.

Steve: It would. It would be difficult. And...

Leo: Maybe not impossible with computer computation.

Steve: Well, and that's the problem is that what this does is it has the effect of dramatically blunting the frequency analysis. That is, if you think of a frequency analysis as like having a peak at "E," and then a lesser peak at "T," and a lesser peak at "S," there's going to be a strong peak-y response to a monograph substitution cipher.

Leo: A Caesar cipher.

Steve: Yes, a Caesar cipher. There will still be one for a digraph cipher. So in fact, quoting Wikipedia about this, Wikipedia says "Playfair is no longer used by military forces

because of the advent of digital encryption devices. Playfair is now regarded as insecure for any purpose because modern computers could easily break the cipher in seconds."

Leo: Very quickly, yeah.

Steve: Yes. There's just nothing to it. But I wanted to...

Leo: Isn't that funny. But that just shows you, computational speed is key in a lot of these things.

Steve: Yes.

Leo: Including RSA.

Steve: Yes. So, okay. So I struggled with it for a couple days, like is there a way to make it secure. There are variations on Playfair. There's one called Four-Square, where you use four 25x25 grids. You locate the two characters in diagonally opposite 5x5 grids, and then you use the opposite corners for your output, your ciphertext. And that allows you to have more independence because you just got much more entropy, essentially, that you're able to store. And I also liked it because one of the other problems with Playfair is that you only could get out ASCII. And we know that high-entry passwords, we like them to have special symbols and characters and things. And there was no way to encrypt numbers and the dot and hyphen, which do occur in domain names.

Leo: Ah, good point.

Steve: So I thought, well, no, I just - I can't do it. But here was the other problem. One of the things I wanted, one of the powerful features of modern encryption, and we've talked about this when we were talking about encryption technology, is there are these things called "encryption modes," like CBC is Cipher Block Chaining. And what cipher block chaining does is it keeps each operation from standing alone.

There is a fantastic example on Wikipedia. If you go to Wikipedia and look up "encryption modes," the first link will be a link to Wikipedia. And if you scroll down, there's a picture of the Linux penguin showing you the danger of not using a mode of encryption because even though you are encrypting - for example, with AES, we take a 128-bit block, and we encrypt it with a key. But if we simply then take the next chunk of data and encrypt it with the same key, there's something of the content leaks through. And the way to solve that is you take something from the previous encryption and mix it in with the next one. That is, so you make your downstream encryption dependent upon everything that came first.

And so that was what I wanted. See, because the other problem with the Playfair cipher, not to mention any substitution cipher, is that each character pair stood by itself. Each pair stood alone. And that's just - there's no way for that to be safe. But more importantly, say that you were encrypting, like all domain names have .com on the end. So that would mean that, no matter what I did, the ".c" would encrypt to a character

pair, and the "om" would encrypt to the same character pair. And so every domain name that had .com would always encrypt to the same four characters.

Leo: You'd start to recognize that.

Steve: Well, yes. And no matter what came before, it would always be the same. So suddenly you lose four characters, four characters' worth of the strength of your password.

Leo: You in fact create patterns, which is what happened to Tux here.

Steve: Exactly. Okay. So here is where what actually was a breakthrough happened. You know the old cartoon with Einstein where he's trying to work out e=mc2, and he gets lost, and he says, "And then a miracle happens." So there's a cloud, and then out comes the result.

Leo: Eureka.

Steve: What I wanted was I wanted some way for something simple, a simple paper cipher, to have memory. It had to have state. It had to, in order for the past to affect the future, you had to have state. It had to have memory. You had to, like, be in a location somehow, like on a grid. And then I thought, okay, that means that the domain name, like the character of the domain name moves you to a next place on the grid. And then the next character moves you to the next place. So that where you are is a function of the history of the characters you have encrypted so far. And so I then thought, okay, for that to work, I need to have a grid which is 26x26, where on every row and column, each letter only occurs once.

Leo: Okay.

Steve: So that, if you think about it, every row would have the alphabet in a mixed-up order. But you want to be able to locate, like, "A" of Amazon at some point, say on the top row. Then you want to look up "M" of Amazon in that column that starts with "A." So that means you can only have one "M" in that column. Then you want to, wherever that "M" is, now you switch to rows, and you want to look up the "A," "AMA," the "A," the second "A" of Amazon, somewhere on that line, on that row. So that meant - and I see someone in the chatroom just said "A lot like Sudoku," and that's exactly right.

Leo: Oh, Sudoku is exactly that, yes.

Steve: There are no digits that occur in any row or column. Well, and then I thought, okay, so I'm like, okay. I'm on to something here. So, and I thought, somebody somewhere must have thought about things where there's only one of them in any row or column. And, oh, boy.

Leo: Have they.

Steve: It's called a "Latin Square."

Leo: Sudoku is actually a special case of a Latin Square.

Steve: It is. It's actually a restricted Latin Square because within the 3x3 sort of sub-squares, you also have the same...

Leo: Conditional...

Steve: The same constraint.

Leo: Yeah, constraint, yeah.

Steve: So I stumbled into a massive world about the Latin Square. And...

Leo: You know what I love is you reverse-engineered crypto, in a sense, and came up with the same thing, in effect. By thinking about it, you realize, well, we have to solve it this way.

Steve: We need cipher block chaining.

Leo: We need to do this, yeah.

Steve: We need somehow for the future to affect the past.

Leo: And you know you're on the right track because you came up with something that is now, it turns out, widely used.

Steve: Although never for crypto. As far as I know, this is the first cryptographic application of a Latin Square.

Leo: How interesting.

Steve: Now, Leo, you've got to go to GRC.com/latinsquare.htm.

Leo: Okay.

Steve: Because I needed to, early in my research, and there's a link off of that one page, if you want to...

Leo: The Workbench here?

Steve: Scroll down to the Workbench, and you can now start clicking things.

Leo: We can play with the Latin Squares.

Steve: I needed to understand how Latin Squares operate.

Leo: So of course he wrote some code.

Steve: Yes, that is a - the Latin Square Workbench, which is available on GRC, is a - it allows you to experiment with manipulating Latin Squares because I realized that's going to be the heart of the Off The Grid Cipher.

Leo: Right, right.

Steve: So, okay. So Latin Squares have fascinated mathematicians for decades. And one of the things that I learned, and Wikipedia's got a great page about Latin Squares, they show, among other things, how many Latin Squares there are. And what's interesting is the number of possible Latin Squares gets big so fast that today, Leo, today nobody knows how many there are that are bigger than 11x11. Nobody knows how many 12x12...

Leo: We can't compute it.

Steve: No, we can't compute it. The mathematicians, if you scroll back up on that page, above that chart, you'll see there's a formula that shows that there are - that we know that there are at least a certain number. And that number is big when you get to 26x26.

Leo: Very big.

Steve: In fact, that number is 9.337 times 10^426.

Leo: That's big.

Steve: Okay. The log2 of that, that is, the number of bits that is, is 1,418 bits.

Leo: Wow, that's a lot.

Steve: So that is an incredible amount of entropy.

Leo: Plenty.

Steve: Okay. So here's how the Off The Grid cipher works. And for our listeners, I mean, I know this is a lot to take in. I wanted to give everybody sort of a rich background so you'd be able to understand how to use this thing. The idea is that you will be able to go to GRC, and it's GRC.com/offthegrid.htm. And there's a whole set of - there's a complete tutorial in using it. There's a - I've written a JavaScript random grid generator. And you wouldn't believe how much technology is in this thing because, I mean, just to get this thing to generate the grid quickly - in fact, if you want to, you can see it work by clicking on it, watch it work, and then regenerate one. And it'll get stuck at some point as it's trying to solve the grid because it's got to create a grid that obeys this Latin Square property.

Leo: It's using brute-force almost; right?

Steve: I'm doing a highly constraint-based tree search with a ton of heuristics in order to allow it to not spend forever because - but one of the indications that it's able, that there are so many grids, is that it's just able to give you grids on the fly very quickly. And no two of these will ever be the same because we've got so much entropy in this thing, it's not even crazy. I mean, it is crazy. So the way...

Leo: You slow it down to let everybody watch it work.

Steve: Yes. Under "Watch It Work" I actually use a different algorithm than - on the "Get It Done," I use a "restart the row" algorithm because sometimes it takes too long, going too far back.

Leo: It's instantaneous.

Steve: It is, it's virtually instantaneous.

Leo: It's instantaneous, yeah.

Steve: And that took - there's some serious technology in there to get that going.

Leo: Did you obfuscate the JavaScript, or can we look at the source of this and...

Steve: I did obfuscate it. But anyone who is interested is welcome to it. And of course obfuscation doesn't help you much. You can easily decrypt it because the browser's having to do that in order to run it.

Leo: Right.

Steve: Okay. So the way this operates is pretty simple. You take the first six characters of the domain name, and those are expanded into 12 characters for that domain name's password. You look up the first character, somewhere along the first row, the first line. And we'll take "A." I use "Amazon" as my example in the tutorial. And Leo, if you go to the second page there on the "How It Works," you'll be able to see it on the site. Oh, and actually I think the second page is all the criteria that I had, the goals I had, because I wanted any single change to a character to give you a completely different result. I wanted really strong crypto out of this thing.

And if people say, well, is this as strong as AES-128, like commercial-grade crypto, it's different than commercial-grade crypto. Commercial-grade crypto, like the Rijndael cipher that we've talked about, where you put in 128 bits, and you get a completely pseudorandom 128 bits out, you could - a bad guy could look at those in and out, in and out, in and out all day long and gain no information about your key. As we're about to see, the Off The Grid cipher does leak a little bit of its structure with every password. Except there is so much entropy in this thing that you - and we have done an analysis, and I have a security analysis of it, looking at the amount of entropy that exists in the grid. But let me get to how you use it.

So you would locate "A" of Amazon across the top. Then you basically follow the path for the first six characters of the domain name. So you locate "A." You go down to "M." You go over to "A." You go up or down to "Z." You go over to "O," and then up or down to "N." Basically you follow A-M-A-Z-O-N. And what that does is that will bring you to the starting point for what I call Phase 2: The Cryptographic Generation part. What we've basically done is we have hashed the domain name Amazon using - by following the path, we've hashed the entire name so that where we are is dependent upon every letter of the domain name.

Leo: But it has more characters than the actual domain name.

Steve: Well, the encryption portion does. We're about to do that.

Leo: You're going to pad it.

Steve: Yeah. Well, no. There's two phases. Phase 1 just - we sort of run through the domain name to get to the starting point...

Leo: Got it.

Steve: ...for Phase 2.

Leo: That's the starting point.

Steve: Yes. And the reason we do that is, if, like...

Leo: So we have a unique starting point for every domain.

Steve: Exactly. Exactly. So that's like hashing. Now we find "A" of Amazon again on whatever row, or sorry, whatever line we were left on. And we find the "A." And then we go, we output the two characters that follow it in the grid. So we output the two characters that are next, that are, like, after the "A." We call it "skipping over it," or "overshoot" is the term I use. We find the two characters after the "A," and those we write down. And so that's where we're now located. We then find the "M" somewhere in the row that we're now in, and output the two characters overshooting it.

Leo: Ah, got it.

Steve: So for each character of the domain name, we output the two characters that follow that. And then we simply go through the grid for the six characters of "Amazon," and that will give us a 12-character output. Or if you took GRC.com, we have "GRC.co." Now, the way I handle the letters and dash and dot is that the top of the grid, in the center, the top of the grid, are the numerals 0 through 9 and dash and dot. And that forms a simple little translation table that allows you to translate those into alphabetic characters, which you then look up on the grid. So what we have - and again, I've got all this documented on the site. A user - and I need to say that I've got all the documentation there. I have not finished with the final grid printer. You could use one of these grids and print it out.

The problem is, I want to let people change the font, change the size, change the color. They might want to print it again in a different size. So I need to add the ability to capture the key that's used so that the user can save that safely and then dump that into the page later. Right now all of that's just happening behind the scenes. So in the next couple days, I mean, it's the next thing I'm going to do, but I wanted to introduce this while I was here in the studio with you, Leo.

So the idea is you print this grid. You then take it offline. I mean, it is just - it's a grid that is a 26x26 grid. And you're then able to use it to encrypt, I mean, anything you want it to. You could mail it to a friend in New York, and then you could use this to create passwords which you then send in email, along with a file that's encrypted with it. And nobody who intercepted your email could decrypt it. I mean, so you could use it for other things. I mean, it's a general-purpose, paper-based cipher, that doubles the length of whatever you put in, that has a phenomenal amount of entropy. And I go into all this on the pages, the security analysis of it.

Leo: You're going to mark it up, though. If you use it, you're going to mark it up, right, as you use this.

Steve: Really?

Leo: So you print out a new - you can do it with your finger? You get fast enough?

Steve: Yes. You're fast enough. And what is really weird is, before long you sort of memorize it. You sort of...

Leo: So you can reuse your grid over and over.

Steve: Yes. Yeah, the idea is you just sort of scan along with your finger, A-M-A-Z-O-N, to find the start. And then you're able to just type or write down the output, two characters out for every character in. And...

Leo: So I've got this grid which I generated right from your page.

Steve: Yup.

Leo: I go to "A," the first "A" is right there in the upper left-hand corner.

Steve: Okay.

Leo: So now I go - oops, it changed. It finished. Okay. Let me find - there is the first "A." So now I go down to the first "M." And I don't care about upper or lowercase?

Steve: No. Now, that's one thing I didn't mention is that I have upper and lowercase in the grid. That'll be an option that users may want to turn off because the comment's been made, and I feel this way also, that lowercase is just easier to visually scan. But the idea is you ignore the case while you're looking for the letter.

Leo: While you're generating. But you use this - got it.

Steve: But you use the case when you're outputting them. That way we get alpha characters with upper and lowercase.

Leo: So really, as I go, I start with the first letter "A," and then I'll just take the first two to the left of it.

Steve: Exactly.

Leo: So my first two letters of my password are lowercase "M," uppercase "G."

Steve: Yup.

Leo: Then I know - I remember where my "A" was, so I go back, and I go down to the "M."

Steve: Yup.

Leo: First two letters to the left, lowercase "o," lowercase "c."

Steve: Yup.

Leo: Now I'm going to go over to the next "A." There it is. And again, capital "K," lowercase "r."

Steve: Yeah, you got it.

Leo: So you're right, you don't really have to...

Steve: You don't really. And what I have found, 'cause I had one that I printed out, it's...

Leo: It's pretty quick.

Steve: It really seems - some of your memory seems to get engaged, and you find that you get quicker with a given grid. And the idea would be that a user would have their grid. It's impossible to ever get the same one again. We've analyzed the threat model. The threat model would be that somehow a bad guy gets a collection of your passwords generated by a grid and the matching domain name.

Leo: They can start to make some sense of it.

Steve: And they start to sort of piece it together. And the analysis is, what, if any, is the danger of that? And there is so much entropy - again, it's easy to say, well, 1,418 bits of entropy. But consider, remember that what we just read about the beginning of the Big Bang and that you still couldn't find a 128-bit key, okay, we have 10^388 times that much entropy.

Leo: That's a lot more.

Steve: Times that much entropy. We've got it to give away. And so the secret of this is that, while it is a leaky cipher, because think about it, your password is actually composed of pieces, little two-character pieces...

Leo: You could start to build a table, if you knew the cleartext.

Steve: Well, and so you would have a digit of, a character of the domain name and two characters that follow. You don't know if it's above it, below it, to the right or to the left. But you know that there's a - I call them a "triple." You would know that there were those three. You also don't know where they are relative to the other guys. But I have done an analysis that I will suggest our listeners look at, which explains what information gets freed. And the idea is, of course, that you behave yourself so that nobody gets all your passwords. Turns out you get nothing if your password gets stolen or even if a bunch of them are stolen because there are so many possible Latin Squares that the bad guy still has to guess. And no matter how much constraint they put on it...

Leo: There's a lot of guessing.

Steve: ...all the other ones are still not known.

Leo: Wow.

Steve: So that's Off The Grid. We now have an arguably secure, 100 percent low-tech - after you get your grid generated it's low tech - paper-based cipher. And my job here is done.

Leo: Wow. That's really interesting. Really interesting. So I would print out this cipher. And I would follow your steps.

Steve: Yeah. Now, people may not - the idea would be that any time you go back to a site, you can recreate the password you used from that site. So you might want to use it for dumb sites that make you create an account, like just to post a response to a blog, where it's like, okay, I just need a throwaway password. So I'm saying that I guess maybe you don't want to use it for your BofA login that you're using all the time, just because you wouldn't want to have to do this all the time. Or you might use this as the input to a password storage system.

Leo: Oh, this you only use once for maybe LastPass.

Steve: Yes. But it's always in your wallet. So if for any reason you don't have your storage system, you can fall back to this.

Leo: Ah.

Steve: So you could use this to generate passwords, really secure 12-character passwords, which you then store...

Leo: In your safe.

Steve: ...in your safe. But if anything ever happened to your safe...

Leo: You could recreate it.

Steve: ...they all came originally from your Off The Grid password.

Leo: Of course the first thing I ask is, well, why don't we just create a computer program to do this? But that is not what we're trying to do. We're trying to do this without technology.

Steve: Right. Exactly. The idea was...

Leo: You could have a computer program that did this.

Steve: I will probably do one. But I don't know how to make it secure. Maybe we could run it in a phone, but then there's a problem with it getting loose from your phone. I mean, the idea is really to be Off The Grid so that no...

Leo: And there's a step of obfuscation here because you have - you begin your square in a kind of - in a way only known to you.

Steve: Yes. Oh, one of the other cool things?

Leo: We left that out.

Steve: Leo, imagine that you're working with a server that requires you to change your password.

Leo: You change your system.

Steve: You start on a different line. All you have to do, instead of starting at the top line, you start at the second line. You'll get a completely different password if you start at the

second line. And then, if you're allowed to go back to the first line...

Leo: You have to make a note somewhere.

Steve: ...or you could do it like what month of the year, you know, zero...

Leo: Oh, I like that.

Steve: Yes. What month...

Leo: That way you could recreate that, as well.

Steve: That way, exactly, you're always able to recreate it. And so actually you could have up to 52 different passwords, depending on whether you start on which line or start on which row because you could also change the orientation and do columns first instead of lines. So the idea is it's...

Leo: It's pretty flexible.

Steve: It is. And what's cool is it's a Latin Square. And there are so many of them. And so it's a - and actually it's a state machine. I forgot to say it's a Latin Square-driven, finite state machine...

Leo: That was one of the criteria that you had.

Steve: ...that moves you through the grid.

Leo: You needed some memory.

Steve: Yes. And so we get a 676-state state machine because that's 26x26.

Leo: Awebby (sp) in the chatroom says, could you improve the statefulness by starting your next cipher from where you left off the last time on the grid instead of row number one?

Steve: You could, but then you would need to know...

Leo: You have to remember it.

Steve: You'd be chaining ciphers. And that maybe is overkill.

Leo: And you would have to somehow remember where you left off.

Steve: Yes.

Leo: I mean, that's not something you could deduce.

Steve: Yeah. And really, also remember that we're only really choosing one of 26 rows, or columns. So it's not like to start Phase 2, the Phase 1, where we follow the path, we're only going to be, like, in one of 26 rows. So there aren't - it's not like it's anyplace on the grid because we're going to then look for the first character of that domain name when we start Phase 2. But what I like about it is it's flexible. It's open. It's public domain. And it's just very cool. It's never been done before.

Leo: People are grappling with the idea of how do we prevent losing it.

Steve: Yes. Now, okay. And that's why what I don't have yet is the page which will allow you to copy and paste the master key. So I will have that.

Leo: Oh, so there is a key that you use to generate.

Steve: Yes. There will be a key. And that you put in a...

Leo: So you could regenerate - if you save the key, you can regenerate the square.

Steve: And the idea is you'll be able to choose fonts. You might want to make it bigger. You might want to change the color. I'll have all kinds of options that are driven by that key.

Leo: Got it.

Steve: And so we will generate the key for you randomly. But then you'll be able to save the key in order to recreate the grid at any time.

Leo: That's excellent. Very, very...

Steve: Yeah. It's just, you know, now we have it. We never had one before.

Leo: I had all my crypto books pulled out here to follow along. Wow, that is very interesting. And you don't have the issues of statistical analysis giving you spikes on certain letters or any of that stuff.

Steve: Correct.

Leo: It's completely, really completely random.

Steve: Yes, it is. There are so many grids that there is no way, even with someone knowing a bunch of your passwords and the related domain names, there's no way for them to piece it back together. They can know all that, and all the rest of the characters have so much entropy still, it's like - I calculated it, and I'll show the math on my page. But we've got so much entropy that you - even Rijndael, with 128 bits, we have 1,418 bits of entropy.

Leo: Wow.

Steve: Yeah.

Leo: GRC.com/offthegrid.htm, if you want to take a look at all of Steve's work and create your own secure square.

Steve: Yeah. Play with it for now. I'll have this thing finished. And I'm sure by next week I will announce that the actual grid creation page is ready.

Leo: But you could copy and paste...

Steve: You could use the one that I'm showing right now.

Leo: Yeah, yeah, because it generates a new one every time you go to that page; right?

Steve: Yup, yup.

Leo: Steve Gibson is at, and this is at, GRC.com. That's the place, absolutely the place to go.

Steve: Whoops, the Xpresso board is sold out.

Leo: Yes, it was sold out almost immediately, by the way. The chatroom is telling us, almost immediately after you gave that address, the last six...

Steve: Yeah, that's just so cool.

Leo: But I'm sure they'll make more. I hope they can make more.

Steve: Or get more, or, yes, yes.

Leo: Or get more or something. The company that makes them will make more. If you go to GRC.com, all the details are there. Steve puts the podcasts up there, as well, in 16KB form; full transcriptions, as well. We have audio and video available at TWiT.tv/sn, Sierra November, if you want to go there for a copy, as well. And of course on iTunes and the Zune Marketplace and everywhere else you can get shows like this. Get Security Now!, Subscribe to Security Now!. That way you'll get it every week. You won't miss an episode. And you could go back in time if you wanted, 315 episodes, they're all there, both at the TWiT site and at Steve's site.

Steve: However many years that is.

Leo: Well, we're now in our seventh year.

Steve: 52.187 or...

Leo: 179, yeah, yeah. We're in our 5.96th year or something. But...

Steve: Yeah, I think we're in our seventh year. We're six point something.

Leo: Seventh year? You're right. You're right.

Steve: We're north of six.

Leo: We've completed six.

Steve: Yeah.

Leo: That doesn't seem right.

Steve: So we'll have our Q&A next week. And then, oh, boy, the week after is TCP.

Leo: Awesome. Awesome. Steve is also the creator of SpinRite, the world's best hard drive maintenance and recovery utility. While you're at GRC pick up a copy. You never know when you're going to need it.

Steve: Yeah, don't wait for Christmas.

Leo: Don't wait for Christmas. Bent's parents can get it right now. Just go to GRC.com. We do this show normally - we're doing it a different day because Paul Thurrott wanted to switch. But normally we do this show on Wednesdays, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC at live.twit.tv, so you can watch it live if you prefer. I think it's fun to watch live because we talk before and after and so forth. Steve, thank you so much for being here. It's really nice having you in-studio.

Steve: I'll be back.

Leo: Yeah, come back. We love having you hear. Dinner's on me anytime you come up here. And the Cabernet.

Steve: Fantastic.

Leo: And we'll see you all next time on Security Now!.