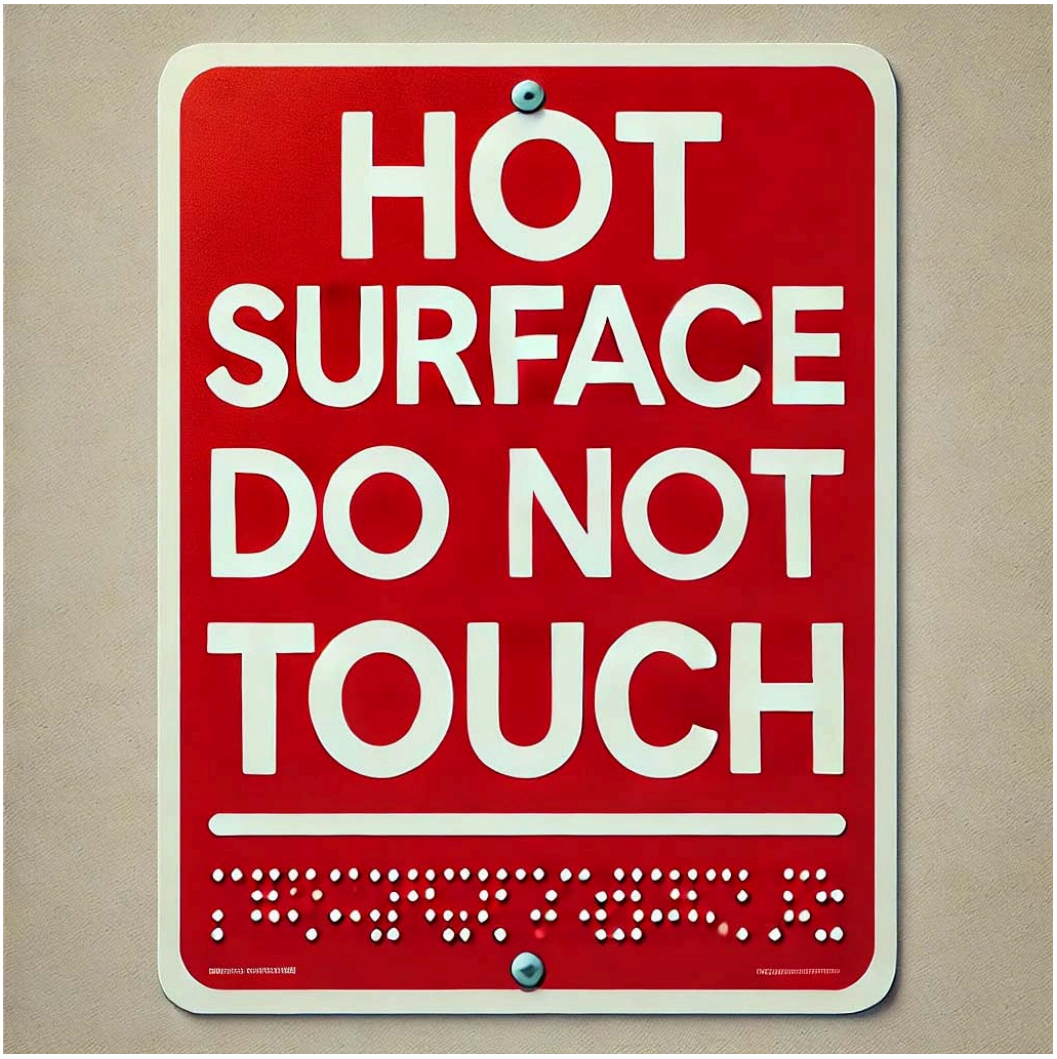
Security Now! #997 - 10-22-24

Credential Exchange Protocol

This week on Security Now!

Did Chinese researchers really break RSA encryption? What did they do? What next-level terror extortion is being powered by the NPD breach data? The EU to hold software companies liable for software security? Microsoft lost weeks of security logs. How hard did the try to fix the problem? The Chinese drone company DJI has sued the DoJ over its ban on DJI's drones. The DoJ wishes to acquire "DeepFake" technology to create fake people. Microsoft has bots pretending to fall for phishing campaigns, then leading the bad guys to their honeypots. It's diabolical and brilliant. A bit of BIMI logo follow-up, then… A look at the operation of the FIDO Alliance's forthcoming Credential Exchange Protocol which promises to create passkey collection portability.

Generic accessibility requirements may not always produce an appropriate outcome.



Security News

"Chinese researchers break RSA encryption with a quantum computer"

I know from having created and written InfoWorld Magazine's TechTalk column for eight years, that the way things work in publishing, the authors of columns and news articles have absolutely no control over works' titles. Why that is true is something I've never understood. But it is. I can't begin to tell you how many times I was distressed to see the headline one of my carefully thought out and crafted columns was given after it left my control and headed to the printer. My column's title often bore no relationship whatsoever to what I had written.

And so, with that understanding, I can forgive the well-meaning author of a piece that appeared last Monday the 14th in CSOnline. The headline of that piece could not possibly been any more misleading than it was, so I can only imagine what its author thought when they saw it in print.

The incredibly provocative headline in question read: "Chinese researchers break RSA encryption with a quantum computer". Did that happen? No. It didn't even remotely happen. It wasn't and still isn't even remotely close to happening and there's no way to characterize what did happen as having **"broken"** RSA encryption. **"Breakage"** in cryptography has a very specific bone-chilling meaning ... and this isn't it.

Fortunately, to regain some sense of order to the universe, one only needs to read past that deliberately fictitious headline to the first sentence of the actual article, which says: *"The research team led by Shanghai University's Wang Chao, found that D-Wave's quantum computers can optimize problem-solving in a way that makes it possible to attack encryption methods such as RSA."* Or phrased another way: "A team of very clever Chinese researchers discovered a better way to employ some characteristics of D-Wave's quantum computers against the prime factoring problem that lies at the heart of RSA's encryption protection." Unfortunately, the truth of their discovery makes for a much less exciting headline.

Through the years of this podcast we've talked a lot about strength of RSA encryption which lies entirely in the still surprisingly intractable challenge of factoring extremely large – and when I say
"extremely large" I mean "humongous" – numbers into their two prime factors. The basis of RSA's extremely clever system is that we first choose a very large, as in 4096-bit, prime number at random. That will be our private key. Then we hide that private key by choosing another similarly large 4096-bit prime number and then multiply those two primes to obtain an 8192-bit product. The product of those two primes is the public key, inside of which is hidden the private key. So if it were possible for some computer system to factor that even more massive 8192-bit public key, then that original private key that was hidden inside the public key could be revealed and RSA's protection would then actually be in trouble.

The Chinese researchers explained in their paper <quote> "Using the D-Wave Advantage, we successfully factored a 22-bit RSA integer, demonstrating the potential for quantum machines to tackle cryptographic problems." That's all they said. "We successfully factored a 22-bit integer, so – NEWS FLASH! – quantum computers can be used to factor integers."

If memory serves, the last time we looked at this a few years ago, other researchers were announcing their breakthrough by factoring a much smaller number. So I have no doubt that this

represents a significant discovery and another breakthrough in the application of quantum computer technology for breaking cryptography. But at today's strength of 8,192 bits which would require factorization, RSA's factorization protection still appears to be entirely safe.

At the same time, these are the sorts of breakthroughs that make cryptographic researchers nervous, which is why it's a good thing that our industry has already designed and is already deploying so-called post-quantum algorithms that no longer rely upon the protection offered by the factorization problem.

In the case of Signal's approach to this, since these new quantum-safe algorithms are still relatively new and unproven, they're taking the belt **and** suspenders approach of using both the old time-proven and the new and hopefully safe algorithms at the same time. In that way Signal's users are already protected against the possibility of some true breakthrough in the use of quantum computers to break old-school asymmetric key crypto, while also hedging their bet by retaining the use of time-proven old-school asymmetric crypto in case some flaw might later be discovered in these newer post-quantum solutions.

So, for all of this podcast's followers who saw this shocking headline and forwarded it to me asking whether the end of the world was nigh... I saw nay.

The next level of terror extortion

A buddy of mine forwarded a scam PDF that had arrived in his email. But the opening line of this particular scam is what caught my attention and made it onto today's podcast. Although his email does not have any aspect of his name in it, the PDF was correctly addressed to him with his full and correct first and last name, then it said:

> ***[Addressed to him by full and correct name]**,*
>
> *I know that calling **[and it had his accurate phone number]**, or visiting you at **[and then it had his full current residential street address]** would be an effective way to contact you in case you don't act. Don't try to hide from this. You have no idea what all I can do in **[and then his city of residence]**. (That opening paragraph was all in bold type.)*
>
> *I suggest you read this message carefully. Take a minute to relax, breathe, and really dig into it. We're talking about something serious here, and I ain't playing games. You don't know me whereas I know you very well and right now, you are thinking how, correct?*
>
> *Well, you've been treading on thin ice with your browsing habits, scrolling through those videos and clicking on links, stumbling upon some not-so-safe sites. I actually placed a Malware on a porn website and you accessed it to watch (if you know what I mean). When you were watching videos, your system started out working as a RDP (Remote Protocol) which provided me with total control over your device. I can look at everything on your display, flick on your camera and mic, and you wouldn't even notice. Oh, and I have got access to all your emails, contacts, and social media accounts too.*
>
> *Been keeping tabs on your pathetic existence for a while now. It is simply your misfortune that I stumbled across your blunder. I gave in more time than I should have exploring into your life. Extracted quite a bit of juicy info from your system, and I've seen it all.*

Yeah, Yeah, I've got footage of you doing embarrassing things in your room (nice setup, by the way). I then developed videos and screenshots where on one side of the screen, there's whatever garbage you were playing, and on the other part, it is someone doing naughty things. With simply a click, I can send this video to every single of your contacts.

... and it goes on for another page like that, until it finally gets around to demanding the transfer of $2,000 USD in Bitcoin to the Bitcoin address it provides.

The reason I'm bringing this up here is that the extremely compelling and terrifying part of this is that the sender of this email is obviously in possession of its recipient's full name, telephone number, residential street address and city. So of course my mind immediately jumped to the data that escaped from the National Public Data breach. This is a textbook classic and perfect example of how the data from such a breach could be – and now clearly is being – used to dramatically increase the grip and believability of these sorts of scam extortion letters. This makes those days of the Nigerian prince who needs our assistance to move funds out of his country seem quaint by comparison.

But the thing that really yanked my heartstrings here, is the idea of how many people are truly going to be terrorized by this. They're going to receive this email and open it to see their name, their phone number and the city and street address where they're probably sitting right then, in an awful and horribly threatening extortion letter. They won't know how anyone could have otherwise obtained their information, which they probably still regard as private, nor will they appreciate that this was likely sent by some cretin in Russia or North Korea who knows absolutely nothing about them and has no capability to, in any way, act upon the information that was obtained from a massive data breach of highly personal data **on everyone.**

I don't think it would be overreacting for this to justify a widespread public service announcement carried by all social media, online and cable news outlets to let people know that this is not real, is the result of an online breach of personal and private information, and should neither be believed nor given a second thought. And I fear that we should all prepare ourselves to be receiving a lot more of this sort of crap. It's clear that the information that escaped from these recent breaches may not only be used to impersonate us, it can also be used to intimidate and terrify us.

The European Union moves on software product liability

As our long time listeners know, one of this postcast's longest standing observations has been over the distortion in the software industry created by software license agreements that universally disclaim any and all responsibility for any consequences of the use and operation of the software. The wheels don't fall off the cars we drive only because it would be the end of any automaker whose cars' wheels did fall off – because the rigid enforcement of product liability would end that company's existence overnight. But that's never been the situation in the software business where software users have no choice other than to contractually sign away all of their rights in a software license agreement in return for the privilege of using the software, regardless of its quality.

Our listeners also know that I one hundred percent understand that mistakes happen and that the perfect operation of a complex software system can be impossible to achieve. But at the same time, through the years of this podcast we've examined instance after instance of the consequences of deliberate policies – not mistakes – that can only be characterized as enabling continuing egregious conduct on the part of some software producers. This conduct and the policies that enable it are explicitly protected by the license agreements under which such software is used.

I've also often wondered here, when and how this will change. Well, change may be coming. I don't know what to make of this next piece of major earth shaking news, because the changes that the European Union proposes to make in its product liability laws to explicitly include software liability while at the same time eliminating software licensing exemptions seems too radical to actually occur. But time will tell. And the fact that this is moving into law certainly means something. So let me back up a bit and explain what's in the works. The first clue I had about this was from the first item of news in the Risky Business newsletter. Here's what it describes – please listen to this carefully:

The European Union has updated its product liability law to cover software and associated risks, like security flaws and planned obsolescence. The new EU Directive on Liability for Defective Products replaces one of the EU's oldest directives and will provide consumers with the legal tools to hold companies liable in court if they sell defective products.

The biggest change to the old directive is the addition of software products to the list of covered goods. Companies that sell or want to sell in the EU will have to make significant changes to how they are currently doing business if they have failed to invest in proper software development and cybersecurity practices. The new directive extends liability to vendors for software that contains security flaws, where those flaws lead to ***any*** *damage to consumers. This includes both physical damage caused by defective or insecure software but also material damage, such as loss of functionality and features, loss of financial assets, and others.*

The directive also classifies the lack of a software update mechanism to be a product defect and makes the vendor liable. Software vendors are also forbidden to withhold information about a software update's negative impact. The only exemption in liability coverage is when the software update requires the consumer to manually install an update, but generally, the directive sees vendors liable as long as they have control over their product after a sale.

The directive also extends liability to vendors who use any type of planned obsolescence system to artificially reduce the life span of their products. This includes software designed to slow down a device, hardware components engineered to fail after a certain period, or an update that degrades a software's performance in order to entice users to move to a new service, tier, or product.

Companies can also be held liable for misleading consumers about a product's durability, repairability, or expected life span. The directive requires victims to prove a product's defectiveness, but it also adds a new legal mechanism to force vendors to make required evidence available. The new rules exclude free and open-source software (FOSS) from its requirements. The new directive was approved earlier this year by the EU Parliament and earlier this month by the EU Council. It is set to go into effect in 24 months, in the fall of 2026.

https://data.consilium.europa.eu/doc/document/PE-7-2024-INIT/en/pdf

I trust Catalin's reporting, but I needed to see this for myself. So I found the 63-page document from the EU and as far as I can see he didn't get anything wrong. I have the URL of the PDF at the top of page 6 of the show notes for anyone who wants to see this for themselves. So I'm just going to pick and choose a couple of paragraphs from the whole document to give everyone a taste.

After a bit of explanation about how and why the very old previous Directive is no longer useful, this new Directive explains that rather than attempting to edit and amend the old one, it is being replaced in its entirety by this new Directive. That brings us to paragraph #6 which open the replacement Directive and reads:

> *(6) In order to ensure that the Union's product liability regime is comprehensive, no-fault liability for defective products should apply to all movables, including software, including when they are integrated into other movables or installed in immovables.*

Just so that everyone is clear about the legal definition of "no-fault liability", an example I found online reads:

> *No-fault liability is the legal responsibility to compensate someone for an injury, even if you were not negligent or at fault. For example, if you own a dangerous animal and it hurts someone, you are responsible for their injuries, even if you didn't mean for it to happen.*

So it's clear that from the standpoint of a software publisher, unintentional damage will not waive their liability under this new Directive for any damage it may cause. Paragraph #13 explains:

> *(13) Products in the digital age can be tangible or intangible. Software, such as operating systems, firmware, computer programs, applications or AI systems, is increasingly common on the market and plays an increasingly important role for product safety. Software is capable of being placed on the market as a standalone product or can subsequently be integrated into other products as a component, and it is capable of causing damage through its execution. In the interest of legal certainty, it should be clarified in this Directive that software is a product for the purposes of applying no-fault liability, irrespective of the mode of its supply or usage, and therefore irrespective of whether the software is stored on a device, accessed through a communication network or cloud technologies, or supplied through a software-as-a-service model. Information is not, however, to be considered a product, and product liability rules should therefore not apply to the content of digital files, such as media files or e-books or the mere source code of software. A developer or producer of software, including AI system providers, should be treated as a manufacturer.*

And this is followed by paragraph #14 which fully exempts open source software. It reads:

> *(14) Free and open-source software, whereby the source code is openly shared and users can freely access, use, modify and redistribute the software or modified versions thereof, can contribute to research and innovation on the market. Such software is subject to licences that*

allow anyone the freedom to run, copy, distribute, study, change and improve the software. In order not to hamper innovation or research, this Directive should not apply to free and open-source software developed or supplied outside the course of a commercial activity, since products so developed or supplied are by definition not placed on the market. Developing or contributing to such software should not be understood as making it available on the market. Providing such software on open repositories should not be considered as making it available on the market, unless that occurs in the course of a commercial activity. In principle, the supply of free and open-source software by non-profit organizations should not be considered as taking place in a business-related context, unless such supply occurs in the course of a commercial activity. However, where software is supplied in exchange for a price, or for personal data used other than exclusively for improving the security, compatibility or interoperability of the software, and is therefore supplied in the course of a commercial activity, this Directive should apply.

Then we have the question of products that are enhanced by or dependent upon external services. Where does liability lie then? Paragraph #17 says:

(17) It is becoming increasingly common for digital services to be integrated into, or inter-connected with, a product in such a way that the absence of the service would prevent the product from performing one of its functions. While this Directive should not apply to services as such, it is necessary to extend no-fault liability to such integrated or inter-connected digital services as they determine the safety of the product just as much as physical or digital components. Those related services should be considered components of the product into which they are integrated or with which they are inter-connected where they are within the control of the manufacturer of that product. Examples of related services include the continuous supply of traffic data in a navigation system, a health monitoring service that relies on a physical product's sensors to track the user's physical activity or health metrics, a temperature control service that monitors and regulates the temperature of a smart fridge, or a voice-assistant service that allows one or more products to be controlled by using voice commands. Internet access services should not be treated as related services, since they cannot be considered as part of a product within a manufacturer's control and it would be unreasonable to make manufacturers liable for damage caused by shortcomings in internet access services. Nevertheless, a product that relies on internet access services and fails to maintain safety in the event of a loss of connectivity could be found to be defective under this Directive.

And, finally, what about the limitation of liability exemptions that are currently present in virtually all software license agreements? Paragraph 56 of legislation says:

(56) The objective of protecting natural persons would be undermined if it were possible to limit or exclude an economic operator's liability through contractual provisions. Therefore no contractual derogations should be permitted. For the same reason, it should not be possible for provisions of national law to limit or exclude liability, such as by setting financial ceilings on an economic operator's liability.

Not being trained in the law, I cannot render any opinion about the eventual impact of what the European Union has just done. But I can read. And what should be abundantly clear is that a sea change is coming to the product liability side of the software industry.

Even if this is met with a great deal of industry pushback, and it's difficult to imagine that it won't be, it appears that the past half-century of software publishing operating with impunity in a world without accountability or consequences may be approaching its expiration date. Over the past 50 years software and the Internet have gradually grown to become truly mission-critical. But many older aspects of the way things have always been done have remained in place due to inertia and no immediate need to change. Newer tools have been created that could enable software to be significantly more robust than it is today. But programmers still choose to recklessly code in crazy unsafe and unmanaged languages like C and Assembly! Imagine that!

We've seen reports of major projects being deliberately recoded in fast and safe languages which will, at least, be able to deal with ridiculously persistent errors, such as use-after-free, that continue to plague today's code. But these deliberate and expensive recoding efforts remain the few and far between exceptions. This needs to become the norm. So it may be that legislation such as the EU has just put into place, having a 24-month grace period before it goes into effect, will up the ante and finally induce serious consideration of how future coding should be accomplished to reduce the incidents that might subject its publisher to warranted product liability claims.

And since I just dissed two of my favorite languages, let me be clear that it is entirely possible to write safe and secure code in C or Assembly – it's just far more expensive to really do so. The flight computers controlling both the American Shuttle program and the two Voyager space probes were hand coded in assembly language and they both proved to be extremely reliable accomplishments. It all boils down to economics. We know that I write everything in assembly language and that none of what I produce has ever had a problem with bugs. I rarely revise my final product other than to add new features. But I also have the unusual freedom of not having a boss, and more importantly, not writing under any sort of delivery deadline. That's not a luxury most of the world's coders enjoy. So for nearly everyone else, the thing that makes the most economic sense, is using next-generation memory safe languages. That's the only strategy that makes sense for keeping uncaught errors from turning into exploitable security vulnerabilities.

I'm going to be keenly interested to see what comes of the EU's new software liability legislation. Stay tuned!

Microsoft lost weeks of customer's security logs.

BleepingComputer's headline was: *"Microsoft warns it lost some customer's security logs for a month"* and TechCrunch reported this under their headline *"Microsoft said it lost weeks of security logs for its customers' cloud products"* Since going to the source is usually best, I tracked down Microsoft's own report. Under the section titled "What happened?" They wrote:

> *Starting around 23:00 UTC on 2 September 2024, a bug in one of Microsoft's internal monitoring agents resulted in a malfunction in some of the agents when uploading log data to our internal logging platform. This resulted in partially incomplete log data for the affected Microsoft services. This issue did not impact the uptime of any customer-facing services or resources – it only affected the collection of log events. Additionally, this issue is not related to any security compromise.*

The issue was detected on 5 September. Following detection, our engineering teams began investigating and implemented a temporary workaround to reduce the impact of these failures beginning on 19 September.

Those dates are what caught my eye. They say that the issue was detected on the 5th of September and that their engineering teams began investigating and implemented a temporary workaround to reduce the impact of these failures beginning on September 19th. In other words, two weeks lapsed between their initial detection of this issue and their beginning to investigate and implement a temporary workaround. It sounds as though logging is not an urgent priority for them, though after all of the problems they've had surrounding a lack of logging for their customers one would imagine that it might receive more attention. I guess not.

DJI sues the DoJ

DJI, Chinese manufacturer of what are arguably the best small consumer drones in the world, has sued the United States Department of Defense over the DoJ's listing of them as agents of the Chinese military. Reuter's News Service carried the news which contained some interesting detail, writing:

WASHINGTON, Oct 18 (Reuters) - China-based DJI sued the U.S. Defense Department on Friday for adding the drone maker to a list of companies allegedly working with Beijing's military, saying the designation is wrong and has caused the company significant financial harm.

DJI, the world's largest drone manufacturer that sells more than half of all U.S. commercial drones, asked a U.S. District Judge in Washington to order its removal from the Pentagon list designating it as a "Chinese military company," saying it "is neither owned nor controlled by the Chinese military."

Being placed on the list represents a warning to U.S. entities and companies about the national security risks of conducting business with them. DJI's lawsuit says because of the Defense Department's "unlawful and misguided decision" it has "lost business deals, been stigmatized as a national security threat, and been banned from contracting with multiple federal government agencies." The company added "U.S. and international customers have terminated existing contracts with DJI and refuse to enter into new ones."

DJI said on Friday it filed the lawsuit after the Defense Department did not engage with the company over the designation for more than 16 months, saying it "had no alternative other than to seek relief in federal court." Amid strained ties between the world's two biggest economies, the updated list is one of numerous actions Washington has taken in recent years to highlight and restrict Chinese companies that it says may strengthen Beijing's military.

Many major Chinese firms are on the list, including aviation company AVIC, memory chip maker YMTC, China Mobile , and energy company CNOOC. DJI is facing growing pressure in the United States. Earlier last week DJI told Reuters that Customs and Border Protection is stopping imports of some DJI drones from entering the United States, citing the Uyghur Forced Labor Prevention Act.

DJI said no forced labor is involved at any stage of its manufacturing.

U.S. lawmakers have repeatedly raised concerns that DJI drones pose data transmission, surveillance and national security risks, something the company rejects.

Last month, the U.S. House voted to bar new drones from DJI from operating in the U.S. The bill awaits U.S. Senate action. The Commerce Department said last month it is seeking comments on whether to impose restrictions on Chinese drones that would effectively ban them in the U.S. - similar to proposed Chinese vehicle restrictions.

We've talked about this previously so this is not surprising. And this is one of those situations where it's entirely possible to see the logic being applied by each side. It cannot be argued that nothing could ever make a more perfect spying device than camera-equipped flying drones. They are by definition flying cameras and DJI's are among the best. We previously talked about how DJI's drones are being actively used within U.S. Military bases and even on secret military bases. And DJI drones receive software updates. So it's theoretically possible for the Chinese government to order DJI to alter their firmware so as to turn their drones into active spying cameras. And, whether or not it's fair, "theoreticals" are what keep our military planners up at night.

The only way I see for this to work would be for DJI to essentially create a wholly separate US version of DJI as an independent U.S. based division. DJI China could produce the drone chassis and all hardware – which is where the majority of the cost and value lies. But the sole exception would be the drone's circuit board which would be manufactured using U.S.-known components sourced for that purpose. And that US DJI drone control board would be flashed with firmware that had been audited and inspected by technical representatives of the United States. DJI would need to establish camera footage uploading cloud servers in the U.S. without any ties to China and the only connection would be the receipt of brainless drone chassis.

This would all obviously represent a huge burden and cost for DJI. But I can't see reaching any other compromise. It's not strictly fair, but the danger, even if only theoretical, is so great that I think DJI will need to consider some solution along these lines.

The DoJ wants to acquire sophisticated "Deep Fake" capability

Meanwhile, The Intercept reports that this same U.S. Department of Justice is in the market for sophisticated "Deep Fake" technology. The Intercept's headline was *"The Pentagon Wants to Use AI to Create Deepfake Internet Users"* with the subhead *"The Department of Defense wants technology so it can fabricate online personas that are indistinguishable from real people."* Once again I find that the details are quite interesting. Here's the start of The Intercept's coverage of this:

The United States' secretive Special Operations Command is looking for companies to help create deepfake internet users so convincing that neither humans nor computers will be able to detect they are fake, according to a procurement document reviewed by The Intercept.

The plan, mentioned in a new 76-page wish list by the Department of Defense's Joint Special Operations Command, or JSOC, outlines advanced technologies desired for the country's most elite, clandestine military efforts. The entry reads: "Special Operations Forces (SOF) are interested in technologies that can generate convincing online personas for use on social media platforms, social networking sites, and other online content."

The document specifies that JSOC wants the ability to create online user profiles that "appear to be a unique individual that is recognizable as human but does not exist in the real world," with each featuring "multiple expressions" and "Government Identification quality photos."

In addition to still images of faked people, the document notes that "the solution should include facial & background imagery, facial & background video, and audio layers," and JSOC hopes to be able to generate "selfie video" from these fabricated humans. These videos will feature more than fake people: Each deepfake selfie will come with a matching faked background, "to create a virtual environment undetectable by social media algorithms."

The Pentagon has already been caught using phony social media users to further its interests in recent years. In 2022, Meta and Twitter removed a propaganda network using faked accounts operated by U.S. Central Command, including some with profile pictures generated with methods similar to those outlined by JSOC. A 2024 Reuters investigation revealed a Special Operations Command campaign using fake social media users aimed at undermining foreign confidence in China's Covid vaccine.

Last year, Special Operations Command, or SOCOM, expressed interest in using video "deepfakes," a general term for synthesized audiovisual data meant to be indistinguishable from a genuine recording, for "influence operations, digital deception, communication disruption, and disinformation campaigns." Such imagery is generated using a variety of machine learning techniques, generally using software that has been "trained" to recognize and recreate human features by analyzing a massive database of faces and bodies. This year's SOCOM wish list specifies an interest in software similar to StyleGAN, a tool released by Nvidia in 2019 that powered the globally popular website "This Person Does Not Exist." Within a year of StyleGAN's launch, Facebook said it had taken down a network of accounts that used the technology to create false profile pictures. Since then, academic and private sector researchers have been engaged in a race between new ways to create undetectable deepfakes, and new ways to detect them. Many government services now require so-called liveness detection to thwart deepfaked identity photos, asking human applicants to upload a selfie video to demonstrate they are a real person — an obstacle that SOCOM may be interested in thwarting.

Oh Leo... Not only are we nowhere near Kansas, we're also a long way from Mayberry. I suppose this is the new reality. I've said before that I hope our cyber forces are able to give as well as they get. I suppose this is what that looks like. But to modify an old saying, seeing the way the sausage gets made can really put one off of sausage for a while.

Microsoft has bots providing faked credentials to phishing sites

While we're on the subject of things being faked, get a load of this: Microsoft is running a massive deception campaign that is providing phishing sites with fake credentials. The credentials lead to Azure tenants for fake companies. So, in other words, Microsoft has bots reading email to detect phishing. When such phishing is detected, bots visit the phishing site pretending that they are actual people who have been fooled by the phishing campaign. But the

phishing victim bots provide fraudulent login credentials which, in turn, lead to fake company sites which have been established in Azure cloud tenants. That is so cool. Microsoft says that threat actors use the credentials to log into the Azure honeypots in 5% of cases. Microsoft then uses the data collected from the honeypots to learn of, discover and document new techniques. It takes around 20 days for threat actors to catch on to the deception and stop logging into the accounts, but by then Microsoft has collected all the data they need.

I suppose that if this is what they were doing instead of fixing their broken logging I ought to cut them a bit of slack since this sure does seem wonderfully proactive. :)

Closing the Loop

Justin Long

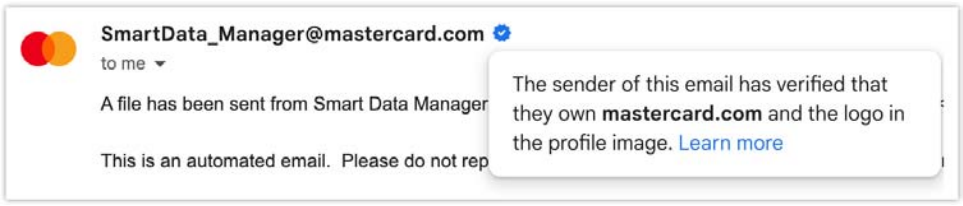
Hey Steve, After listening to your coverage of BIMI, the technology behind BIMI seems solid. However, I would never even tell a user about this, let alone have them rely on it. It is the kind of thing that gets simplified down to "It's easy! Just look for the logo and you'll know it's safe!" My fear is that the scammers will start including logo files in the body of the email, with "Verified by BIMI!" next to it. Then Gary in Accounting sees a logo and thinks it's safe to click. In my opinion, BIMI doesn't do anything to help the problem. If anything, it provides a false sense of security to most risky users. Thank you for everything you do! This podcast helps me every episode. -Justin

I completely agree with Justin. I like the idea of having GRC's logo appearing in those inboxes where anyone's BIMI logo might appear. And while, as we saw last week, it can be quite a royal pain in the butt to get that to happen, for GRC's weekly podcast mailings and for our other much less frequent software update mailings, for the moment at least, if only as an experiment, it's worth it to me. But I think it's clear that email is so messed up that this is all it can ever be – a way for those who care enough to make it happen have their corporate identity represented in the inboxes of any recipients whose clients will do so – and nothing more.

The reason I believe BIMI is surrounded by all this nutty over-the-top security and authentication is that anything we do moving forward should be as secure as we can make it. As I noted toward the end of last week's exploration of BIMI, our industry has continually set the bar too low out of a fear of low adoption from setting the bar high. So the designers behind BIMI – who don't know the future any more than the rest of us – decided that if they were going to create a new facility it would at least be built upon a state-of-the-art secure foundation. And in a world with determined North Korean identity fakers and AI, which have raised the bar on identify verification, doing less would be prone to failure. And if, in addition, it's used as it might well be, as another spam filtering signal, I'll be glad to have it help GRC's BIMI and domain validated.

And note that it's our domain that's being validated for the display of the BIMI logo. Our email already passes SPF, DKIM and DMARC verifications. So once the Internet Archive gets back on its feet, which has still not happened, and DigiCert is able to verify GRC's historic use of our logo, any existing email that's sitting in our listener's inboxes should obtain the BIMI logo retroactively.

Another listener, **Kevin de Smidt**, is currently the Head of Technology at CURE International and was earlier at ValiMail, so he is well acquainted with BIMI. He sent a sample writing: *"This is how Mastercard emails appear in my Google Workspace email - notice the blue checkmark and the text when hovering over it:"*



The image Kevin sent shows a little blue seal with a checkmark and hovering over the checkmark to inquire about it pops up an overlay which reads: "The sender of this email has verified that they own mastercard.com and the logo in the profile image." Then it has a highlighted link labeled "Learn more". So it does appear that at least Google Workspace is surfacing something extra when a fully functioning BIMI verification is present. We'll see what happens once GRC is able to join that group.

That's all folks! :(

When I start producing today's podcast I was hoping to spend more time on listener feedback since I am receiving so much fantastic feedback from our (now) more than 11,300 Security Now! email list subscribers. But there was so much engaging news that I wanted to cover that we've run short of time and space. Let's hope that things are quiet this next week and that we'll be able to spend more time closing the loop with our amazing listeners.

So let's take our last break and then take a close look at the design of the FIDO Alliance's Credential Exchange Protocol which will be used to move Passkey collection between providers.

Let's take a break, then we'll plow into our main topic!

Credential Exchange Protocol

Today, as planned, we're going to examine the operation of the recently announced and widely anticipated Passkeys Credential Exchange Protocol. I should caution everyone that all we have so far is an outline of that protocol. The most recent version of the specification still has a long way to go before it 's ready for the world. For example, what I found in the most recent documents looks like this:

4. Usage Guidelines
[Offer guidelines for using the CXF format to import and export credentials securely.]

4.1. Importing Credentials
[Explain the steps and considerations for importing credentials using the CXF format.]

4.2. Exporting Credentials
[Provide instructions for exporting credentials to the CXF format.]

5. Examples
[Present practical examples of importing and exporting credentials using the CXF format.]

5.1. Importing a Credential Set
[Walk through the process of importing a set of credentials using CXF.]

5.2. Exporting a Credential Set
[Provide an example of exporting a credential set to the CXF format.]

6. IANA Considerations
[Outline considerations related to IANA registrations, including the CXF media type.]

6.1. CXF Media Type
[Specify the media type for CXF and its registration details.]

7. Security Considerations
[Provide an in-depth analysis of the security aspects of the CXF format and its use.]

In other words, we have an almost comical lack of meat on this particular bone. I have no doubt that the various participants are all rowing in the same direction, and that they fully intend to turn this into an actionable specification document at some point. But at this moment what we mostly have is evidence of good intentions. However, scant though it may be, there is enough here to piece together a coherent picture of the system's operation. We're far short of having sufficient information to create a working implementation, but we're going to be able to get a feel for how the system works.

So let's begin with the news coverage WIRED offered eight days ago on October 14th, the day of the big FIDO Alliance Authenticate Conference held in Carlsbad, California. WIRED wrote:

The password-killing tech known as "passkeys" has proliferated over the past two years, developed by the tech industry association known as the FIDO Alliance as an easier and more

secure authentication alternative. And although superseding any technology as entrenched as passwords is difficult, new features and resources launching this week are pushing passkeys toward a tipping point.

*At the FIDO Alliance's **Authenticate Conference** in Carlsbad, California, researchers announced two projects that will make passkeys easier for organizations to offer—and easier for everyone to use. One is a new technical specification called Credential Exchange Protocol (CXP) that will make passkeys portable between digital ecosystems, a feature that users have increasingly demanded. The other is a website, called Passkey Central, where developers and system administrators can find resources like metrics and implementation guides that make it easier to add support for passkeys on existing digital platforms.*

Andrew Shikiar, CEO of the FIDO Alliance, told WIRED: "To me, both announcements are part of the broader story of the industry working together to stop our dependence on passwords. And when it comes to CXP, we have all these companies who are fierce competitors willing to collaborate on credential exchange."

*CXP comprises a set of draft specifications developed by the FIDO Alliance's "**Credential Provider Special Interest Group**." Development of technical standards can often be a fraught bureaucratic process, but the creation of CXP seems to have been positive and collaborative. Researchers from the password managers 1Password, Bitwarden, Dashlane, NordPass, and Enpass all worked on CXP, as did those from the identity providers Okta as well as Apple, Google, Microsoft, Samsung, and SK Telecom.*

The specifications are significant for a few reasons. CXP was created for passkeys and is meant to address a longstanding criticism that passkeys could contribute to user lock-in by making it prohibitively difficult for people to move between operating system vendors and types of devices. In many ways, though, this problem already exists with passwords. Export features that allow you to move all of your passwords from one manager to another are often dangerously exposed and essentially just dump a list of all of your passwords into a plaintext file.

It's gotten much easier to sync passkeys across your devices through a single password manager, but CXP aims to standardize the technical process for securely transferring them between platforms so users are free—and safe—to roam the digital landscape. Importantly, while CXP was designed with passkeys in mind, it is really a specification that can be adapted to securely exchange other secrets as well, including passwords or other types of data.

Christiaan Brand, identity and security group product manager at Google, told WIRED: "In the future, this could apply to mobile driver's licenses, say, or passports—any secrets that you want to export somewhere and import into another system. We've got most of the rough edges sanded down with passkeys, but one of the main pieces of negative feedback over the past year has been around portability and potential vendor lock-in. I think with this, we are signaling to the world that passkeys are growing up."

The goal of "Passkey Central", a resource repository, is similarly to help the ecosystem expand and mature. Product leads or security professionals who want to implement passkeys for their user base may need to make a business case to executives to get budget for the project. The FIDO Alliance is basically aiming to help them with the pitch—providing data and communications materials—and then support their rollout with prefab materials like implementation and roll-out guides, user experience and design guidelines, documentation around accessibility, and troubleshooting.

FIDO's Shikiar said: "We've made amazing progress on passkeys. Usability and user experience are pretty much there. But we do have a punch list and we're actively working on it. Portability is an important feature on that list. And while the biggest brands on the planet are now using passkeys at scale, there's a very long tail of companies that haven't gotten started yet. So we want to offer resources and the assets they need to be successful."

Craig Newmark Philanthropies' Cyber Civil Defense coalition provides some funding to advance passkeys. In an interview with WIRED, Newmark said he believes that passkeys can make a real difference both for the digital security of individual people and for internet security overall. Craig said: "There are a lot of vulnerable systems out there. You need to make it a lot harder for bad actors to defeat password schemes. You need to make everything more secure, and passkeys is part of that."

Okay. Now having noted that there was very little meat on this bone, there was some. The specification we have today has a useful introduction to the problem and application space this protocol is expected to fill; and it turns out to be more than just passkey credential transport.

Here's how today's Credential Exchange Protocol specification introduces the problem:

Individuals and organizations use credential providers to create and manage credentials on their behalf as a means to use stronger authentication factors. These credential providers can be used in browsers, on network servers, and on mobile and desktop platforms, and often sharing or synchronizing credentials between different instances of the same provider is an easy and common task.

However, the transfer of credentials between two different providers has traditionally been an infrequent occurrence, such as when a user or organization is attempting to migrate credentials from one provider to another. As it becomes more common for users to have multiple credential providers that they use to create a manage credentials, it becomes important to address some of the security concerns with regard to migration currently:

- *Credential provider applications often export credentials to be imported in an insecure format, such as CSV, that undermines the security of the provider and potentially opens the credential owner to vulnerability.*
- *Credential providers have no standard structure for the exported credential CSV, which can sometimes result in failure to properly migrate one or more credentials into a new provider.*
- *Some credentials might be unallowed to be migrated, due to device policy or lack of algorithmic capability by the importing credential provider.*
- *Because organizations lack a secure means of migrating user credentials, often they will apply device policy that prevents the export of credentials to a new provider under any circumstances, opting to create multiple credentials for a service.*

In order to support credential provider interoperability and provide a more secure means of credential transfer between providers, this document outlines a protocol for the import and export of one or more credentials between two credential providers on behalf of a user or

organization in both an offline or online context. Using Diffie-Hellman key exchange, this protocol allows the creation of a secure channel or data payload between two providers.

That introduction paints a picture of a more generalized secret exchange protocol. It's clearly useful and, surprisingly, it's also completely lacking in our industry today. Somehow we've managed to come this far without a universal definition for how the owner of some secrets could move them elsewhere. The fact that this is finally being proposed demonstrates the arrival of some much-needed maturity. Up to this point, much of our industry has relied upon closed and proprietary ecosystems. That closure was first pried somewhat open by the promise and delivery of competitive open source software and open development. But the profit motive runs deep and we've seen how shaky some of open source software's foundation can be.

The CXP document noted that the name was subject to change. I'd vote for something that's explicitly more generic than "Credentials" to more explicitly invite the system's wider use. Perhaps simply the "Secrets Exchange Protocol" which can be used to exchange Passkeys but also other things, like password collections and OTP private keys for inter-authenticator secret exchanges. The next subsection under "Scope" adds:

1.1. Scope

This protocol describes the secure transmission of one or more credentials between two credential providers on the same or different devices managed by the same credential owner, capable of function in both online and offline contexts. This protocol does not make any assumptions about the channels in which credential data is passed from the source provider to the destination provider. The destruction of credentials after migration by the credential provider source is out of scope as well.

So they're explicitly keeping this general and it's significant that it can be an offline system. The spec does sketch an overview of how this protocol would work; and frankly, it's nothing special. And that's not criticism. Quite the opposite, in fact, because we're past the point where crypto should be surprising. We now have established and well proven ways of accomplishing pretty much anything we need. The sketch looks like this:

The planned recipient of the credential collection is asked to create an "export request" which will then be provided to the credential provider. That request includes the necessary details including a challenge, the details of the type of information that the recipient wishes to receive, the set of encryption schemes it's able to use and, unless it has access to the credential provider's public key, it will also include the public side of a Diffie-Hellman key agreement.

Remember that Whit Diffie and Martin Hellman invented this brilliant scheme which allows two parties to exchange public keys in full view of any attackers, and upon receipt of each other's public Diffie-Hellman keys, each is able to construct and arrive at the same shared secret key. It's counter-intuitive but it works. I used their system in several places in the design of SQRL.

So the credential importer uses a cryptographic grade random number generator to create a unique Diffie-Hellman key pair. It stores the private half internally and includes the public half in the credential "export request" it's been asked to generate.

If an end-user and/or other authorizing party approves and provides the "export request", the exporter uses the information to create an encrypted payload. The exporter will similarly synthesize its own Diffie-Hellman pair, but it doesn't need to retain any record of it. It will combine its own private half with the importer's public half to create a secret and automatically shared "migration key" that it can use to encrypt the payload using the other parameters provided in the importer's "export request". It signs the challenge provided by the importer and includes the public half of the Diffie-Hellman keypair it just created in the exported response package.

The exported response package is then provided to the credential importer. It includes the encrypted credential data, the signed challenge response, and the public half of the credential provider's Diffie-Hellman keypair which was used to create the shared migration key. The credential importer has been holding onto the secret half of the Diffie-Hellman pair it generated for the export request. So it validates the challenge then combines the secret it's been holding with the exporter's public key that was provided in the export packet. This will recreate the "migration key" which it's able to use to securely decrypt the contents of the exported package.

So what we have is a straightforward application of Diffie-Hellman key agreement where the two parties creating the shared secret key use it to exchange an encrypted package containing the user's credentials.

What's currently missing from the specification is ... everything else to make it actually go. But the overall mechanism is clear, proven and will work. We have no idea what the user experience will be. Whether the Internet will be used in some way by both sides to rendezvous, or whether that might only be an option. This could all be done with a USB thumb drive and "sneakernet."

But the gist is that the user asks the credential recipient to create an export request for the credential sender. That export request is then provided to the credential sender which uses it to prepare an encrypted package. And when that encrypted package is returned to the credential recipient, the residual information the recipient retained from the original export request allows it to securely decrypt the sender's package.

A well known characteristic of Diffie-Hellman is any lack of protection from man-in-the-middle attacks. While Diffie-Hellman brilliantly creates a mechanism for secret key agreement between two parties, it has no mechanism for authentication. Nothing I've described prevents an attacker who's somehow able to interpose themselves between the parties from impersonating each end to the other. Doing so would allow the impersonator to decrypt the package as it moves past.

All we know from the specification is that the credential importer will include a challenge for the exporter to sign. That's all we know today. We know that the signer of the challenge needs to use a private key and that the credential recipient needs to verify the signature with its matching public key. But from where and how does the credential recipient obtain the credential sender's public key? From DNS? From some sort of central FIDO registry? We'll need to wait and see. But the system is not safe to use until one end is able to cryptographically authenticate the other, and that needs to be done with some form of secure public keys directory.
https://fidoalliance.org/specs/cx/cxp-v1.0-wd-20240522.html

Okay. So what of the other part of Monday before last's announcement of the new Passkey Central website?

https://www.passkeycentral.org/home

Having read through the site it's clear that more than anything else, it's intended to be Passkey Adoption Lubricant. It's taken a few years, but passkeys have matured to the point that if any sort of friction is holding an organization back, now might be the time to apply some lubrication.

In the early days, anyone could be forgiven for feeling that passkeys were not here yet, or were not ready yet, or hadn't been proven, or might turn out to be another FIDO failure like their first attempt that never managed to achieve critical mass. Or even that what we already had that was well proven and working with multi-factor authentication or with password managers that made the use of super-strong passwords effortless, had already solved the remote identity authentication problem and was good enough.

The Passkey Central site and its companion Passkeys.dev developer site make a very strong case for passkeys having arrived, and for those who do not get busy with its adoption being left behind and eventually being seen as "doing it wrong." Meta was recently excoriated for storing their users' passwords in the clear without any hashing. The difference between the inherent insecurity of any traditional secret-keeping authentication system – such as static passwords or one-time passwords – and the extreme security of an asymmetric key authentication system like passkeys – which requires no secrets to be kept – means that at some point anyone who is **not** employing the free to use, widely available and increasingly ubiquitous passkeys asymmetric system will similarly cause some eyebrows to be raised.

The point is, passkeys have made the grade. The benefits of the system have proven to be sufficiently strong that the question has moved from "whether" to "when". And **"when"** should be ***"as soon as possible, what are you waiting for?"*** because there is no longer any rationally supportable argument to be made for waiting any longer. The Passkey Central site should now provide sufficient lubrication to help overcome any residual adoption friction. The Passkeys.Dev site provides sample code in Rust, TypeScript, Java, .Net, Go, Python and Ruby and Passkeys test sites are available at WebAuthn.io and WebAuthn.me, with Yubico and Akamai also offering testing facilities.

Once passkeys' Credential Exchange Protocol has been fleshed out – and make no mistake, it does still have quite a ways to go though its overall shape is now clear – the last piece of the passkeys solution-set will have been put in place. Given that all of the major players have signed onto supporting CXP, the last roadblock to further passkeys adoption use has been removed.