ICANN73 | Virtual Community Forum – DNSSEC and Security Workshop (1 of 3)
Wednesday, March 9, 2022 – 12:30 to 14:00 AST

KATHY SCHNITT: Thank you. Hello and welcome to the DNSSEC and Security Workshop, Part 1 of 3. My name is Kathy and I'm joined by my colleague Kim, and we are the remote participation managers for this session.

Please note this session is being recorded and is governed by the ICANN Expected Standards of Behavior. During this session, questions or comments will only be read aloud if submitted within the Q&A pod. We will read them aloud during the time set by the chair or moderator of this session. If you'd like to ask your question or make your comment verbally, please raise your hand. When called upon, you'll be given permission to unmute your microphone. Kindly unmute your microphone at this time to speak.

All participants in the session may make comments in the chat. Please use the drop-down menu in the chat pod and select "Respond to all panelists and attendees." This will allow everyone to view your comment. Please note that private chats are only possible among panelists in the Zoom webinar format. Any message sent by a panelist or a standard attendee to another standard attendee will also be seen by the session hosts, co-hosts, and other panelists.

This session includes automated real-time transcription. Please note the transcript is not official or authoritative. To view the real-time transcription, click on the closed caption button in the Zoom toolbar.

To ensure transparency of participation on the ICANN's multistakeholder model, we ask that you sign into Zoom sessions using your full name, for example, first name and last name or surname. You may be removed from the session if you do not sign in using your full name. And with that, I'd like to hand the floor over to Dan York.

DAN YORK: Greetings, everyone. Thank you for coming into this session today from wherever you are in the world. In this session, as we call it, our DNSSEC and Security Virtual Workshop, we'll be covering a range of topics. Let's actually take a look at what this looks like.

I should say, first, the work to assemble this program happens through a dedicated program committee that has been working on this for a significantly long time. And a number of people who are presenters and also moderators are part of this committee. You can see some of the names here. If you are interested in presenting at the next one of these, which will be coming up at ICANN74, you should pay attention to our call for proposals that will be happening soon after this meeting where we'll be looking for people for the next time. So these are the folks who are involved. This meeting is happening under the auspices of the ICANN Security and Stability Advisory Committee, otherwise

known as SSAC, with some additional support from the Internet Society in what we're doing.

Today, we've got quite an agenda for you. We're going to begin—I have a few bit of remarks around what we're seeing in terms of the current deployment statistics and pieces there. But then we'll be going into a panel discussion around DNSSEC and quantum, and quantum computing, quantum cryptography, what does it have to do? How could it, might it affect the DNS, and what can we do around that? So we've got a great session of presenters that will be there. Then we'll have a Q&A after that. And again, as Kathy mentioned in the opening, we'll be asking you to put your questions into the Q&A pod so we can be able to go and see that.

Then after that, Steve Crocker and Shumon will be here to talk about DNSSEC provisioning automation. This is a topic that if you've been following us for a while, this has been a big topic of how do we automate the actual provisioning of the key signatures into the TLDs and from there to ensure the chain of trust happens there. We have a great panel, a number of different people talking about how do we go and make this happen? What are the newest things? What are the new ways we can go and make all this work?

Then our final panel for the day will be moderated by Russ Mundy who's here, where we'll be talking a little bit differently about some of the other aspects beyond DNSSEC and pieces look a little bit about RPKI, about how some of the things regarding routing security come

back to relate to what we're doing here in DNS and some other surveys and things that are there.

So I want to begin then with some of our routine updates from what we're seeing in terms of DNSSEC deployment. The series of workshops have been going on now for—I forget exactly how long we've calculated this, Russ, but we've been looking at the growing deployment that we've seen and there's two sides to it. There's the validation. Are people checking the signatures of DNSSEC? And the other side is, are people actually signing domains? When we look at the validation sides, we continue to use the metrics that we get from APNIC with their mechanisms that they show there, which show us that we're climbing and overall, on the global side, we're up around 27-30% validation of DNS signatures coming from the networks all around the world that are out there.

You can see on this and you can pull these slides if you have not seen. If you're going into the ICANN scheduling for this session, you can see in the upper right kind of above the participants. You'll be able to download this and all the other slides and the panelists today. We should note, this is by virtue of the way that the ICANN schedule is set up, there are three different sections in the ICANN scheduling app that are system for this workshop. This first block has the presentations for these opening slides and also for the quantum panel. The second one will then have it for the DS automation and then beyond that. Anyway, if you're interested in this, you can go and pull down these slides, see what's here.

You also will note that we include the links at the bottom of all these slides where you can go and see the most recent measurements that are here. What this is showing us is the regions of the world that are seeing the most validation. And you can see obviously, there's several of the places in Oceana where they're getting significantly high rates of validation and you can go on down from there. Western Europe, it's great to see is up around us, we can show there. About 55% of the networks are showing that they're validating DNSSEC in that kind of space.

We also have been doing some work—Wes Hardaker, Viktor had been doing some great work over at the DNSSEC Tools site. And we're showing the number of stats there, if you go to stats.dnssec-tools. You'll see here, this is the count that we're seeing, the continued growth in the number of DS records, which are the records that we use, the sign via ultimate domains that are there and link them all together. So you're seeing those records, it's great to see that kind of growth.

Viktor has also been tracking a lot and Wes with the domains that are signed with MX records in forms of using DANE to go and secure the connections that we have with e-mail. So again, we're continuing to see a nice growth in that with domains that are deploying DANE records for SMTP servers. So great to see that going on.

We also like to show this, which shows that for RPKI, the resource public key infrastructure, which is for routing security, that we are continuing to see the growth of valid RPKI prefixes. So this is a good

thing that we're seeing, the growth, that green line. We want to keep seeing that growing up and continued deployment of that. Likewise, we're seeing, again, growth in the coverage of IPv4 prefixes, again, with RPKI, the number of Route Origin Authorization sort of going on or ROAs. So, all of this is good. We continue to see the growth and to see where that's happening on there.

Since the beginning of this workshop, we've also been tracking the deployment around ccTLDs. It's actually quite similar to what we've had from the last time that we presented this back in last fall at ICANN72. We're pretty much at the same kind of place. So I think we've had one more new one come in. But otherwise, we're still about where we're at with the growth of where things are going. So good to see. We'll continue to see more growth in that space.

If you would like to go and look at these kinds of statistics yourself or use any of the tools, we would encourage you to go for DNSSEC to go to dnssec-tools.org or stats.dnssec-tools.org. We also have stats.labs.apnic. It's there. We also have some RPKI resources that you see there as well. I should have added one more, which is the MANRS Observatory, which is observatory.manrs.org, which also has some of these kinds of statistics around that.

So with that, I'm going to end this segment and I'm going to stop sharing. Okay. Do have any questions? We do not at this time. So I want to begin now with moving directly into our next panel because this could be a lot of interesting discussion in pieces around that.

There's been a lot of conversation around what is quantum computing? What is quantum cryptography? How could this affect the DNS? Paul Hoffman from ICANN recently released a whitepaper that he'll talk about in his section around what is quantum computing? How does it affect DNS? What is that going to be? So, in this panel, we are going to bring together a pool of experts to talk a bit about what this is. So please, you're welcome to ask questions of the individual panelists. We'll also have some time at the end where we'll be doing that. I was told when I looked up on the Internet that there are three types of people in this world: those who understand quantum computing, those who do not understand quantum computing, and those who will simultaneously do and do not understand quantum computing. That was my one joke for this session. We'll leave it at that.

I'm going to first bring up in Robin Wilton, a colleague of mine from the Internet Society, actually, who is going to give us a brief introduction and help us understand why these other three gentlemen who are presenting today are so concerned about this and what they're doing. So with that, over to you, Robin.

ROBIN WILTON: Thank you, Dan. I'm not sure I have any quantum jokes in my presentation. But I did at least consider retitling it, "Everything You Always Wanted to Know About Quantum Cryptanalysis But Only Had 20 Minutes to Find Out." So this is going to be a little bit of a roller coaster because explaining quantum physics in 20 minutes is tough enough. Explaining how it relates to quantum computing and how

that relates to encryption and cryptanalysis, well, that's quite a high bar. So I will crack on right away. So we share a screen here and hopefully you will see a title slide there, "Encryption and Quantum Computing."

So the first thing to note is that I probably lied in the session title. This is not going to be about quantum encryption. I will mention encryption applications of quantum computing right at the end in one sentence. But really, this is about the potential impact of quantum computing on stuff that is encrypted.

So, first, several disclaimers. First, I'm not a physicist, I'm not even particularly a mathematician or computer scientist. I did philosophy in the modern languages for my degree. But I have been having to look at this for a while and my interaction with encryption systems goes back to the mid 80s as a trainee systems engineer and tech support. Someone far better than me at explaining this, Richard Feynman, put it this way that "If you think you understand quantum mechanics, you don't understand quantum mechanics." So I definitely don't understand quantum mechanics. And therefore, much of this presentation relies on metaphors, which are probably inaccurate but which I hope gives you a picture of what is going on.

So what I want to do here is talk enough about quantum physics to help us understand how it relates to quantum computing, and then enough about quantum computing to help us understand how it relates to cryptanalysis. And then we'll look at the potential impact of quantum computing on the main kinds of cryptography that are used

today. I hope that that will pave the way nicely for Paul and subsequent speakers to carry on and describe the applications of this kind of technology.

So a quick word about quantum physics. I think the key message here is that there are things we can observe in the universe that can't be explained by classical physics. So originally, Newtonian physics, about the movement of objects, and then Einsteinian physics, about space time. Those can take us a certain way to explain everything we see in the universe, but they can't explain everything. And one of the most famous examples of this is the so-called double split experiment, which you can use to illustrate the fact that, although a photon is a particle, light actually behaves as both a particle and a wave. And you can do an experiment to show it behaving in each of those different and apparently incompatible ways. As I say, I'm not going to go into the details of the double split experiment. But the key point about it is that at the quantum level, you can explain something that classical physics can't explain by saying that at the quantum level, what's happening with a photon is that it is superposing two states. It is superposing simultaneously a wave-like state and a particle-like state. And once you understand that that's describing what's happening at quantum level, you can then describe what's happening when you see photons apparently act at one point like particles and another point like waves.

If you really want to find out some more really fascinating things about quantum physics and particularly quantum biology, I really

recommend Jim Al-Khalili's amazing program on this on his studies into quantum biology, and there's a YouTube link there for you.

So how does this relate to computing? Well, classical computing is based on bits, ones and zeros. And using combinations of ones and zeros, we can store information, we can make logic gates, and then we can compute stuff. But any given bit can only represent a value of either one or zero at any point. And so the relevance of quantum computing here is that quantum computing changes that fundamental assumption.

Rather than representing only a one or a zero at any given time, a quantum bit, a qubit, is based on the principle of superposing the states of one and zero at the same time. So just in a photon, you can superpose the state of a wave in a particle. Here in a qubit, what we're doing is using superposition to superpose a value of one and a value of zero at the same time.

So you've probably heard about Schrodinger's cat, even if none of us can generally remember the precise thing of how the experiment was set up and exactly what it was supposed to prove. But when he described it, he said—and I put this in because it was quite colorful, I thought. He said that, until the state of what's happening inside the box is observed, the living cat and the dead cat (if you'll pardon the expression) are mixed or smeared out in equal parts. But what's happening here is that while the coin is up in the air, you can't really tell whether it's representing heads or tails, or one or zero, it's only when it lands that you can determine its state.

I was trying to think of another metaphor for this. It's a little bit like if you get a camera, and your camera can take photos that more than 1/5 of a second, so 100 to the second, 1250 to the second. And you take a photograph of a TV screen or have a fluorescent light strip. While those things are operating normally, because they're operating at 50 hertz, it looks to you like they are a constant stream of either light or images. But if you can capture a short enough snapshot of them, you might see that only part of the fluorescent light tubes lit up or that only part of the TV screen is displaying a particular image. So again, it's just an analogy. But while the qubit is, as it were volatile, it could be representing both a one and a zero simultaneously.

So how does that relate to classical computing? Well, I'm not going to go through this example completely. But suffice to say that the example that I've used here is we've got a really dumb computer. It has 12 bits of storage, it's got a function to add bits, and it's got a compare function. And you wanted a piece of binary arithmetic on to four-bit binary numbers. Basically, what we're trying to do here is we want to find out what number it is that you have to add to 0001 in order to get the result 1000. Now, that's a simple enough sum even binary that you can probably do it in your head. But if we had this really stupid computer trying to do it, the computer would have to take that row of question marks, insert values into them, probably starting at 0000, and then increment it one bit at a time, one binary value at a time until it found the value that matched one plus the end result 1000. So it would be stepping through a series of binary values one at a time comparing.

Suppose instead of four bits, we had four qubits. Each of those qubits could be representing both the one and the zero simultaneously by superposing. Therefore, across our four bits, at one time, each of those four bits is representing both a one and a zero. So somewhere in that mishmash of states must be the correct answer. So all we have to do then is identify which of the possible or probable states is the one that we're looking for, the correct answer.

So, in essence, what's happening here is that we're using qubits to represent multiple values simultaneously, instead of having to search through them exhaustedly, one after the other using binary values of one or zero. We're superposing those values and making sense of the resulting volatile set of values, probabilities. So I've said all you have to do is identify which of the possible states in there is the one that you're looking for. That, conversely, is the trick, and I'll come to that in a minute.

So the next bit that we need to deal with is, how does this all relate to encryption? Well, the two principal kinds of encryption currently in use are symmetric encryption, which uses secret keys, and asymmetrical public key encryption. They rely on the difficulty of solving different kinds of problems.

So for symmetric encryption, where you lock the data up with a key, scramble it, and then you unlock it with a copy of the same key. Basically, what you're doing is you're introducing some random data into the stuff you want to encrypt. And then you're mixing them so thoroughly that the end result is indistinguishable from random data.

So for a well-designed symmetric encryption algorithm, there should be no more efficient way of recovering the plaintext from the ciphertext than to try every single possible key until you hit the right one. So the way that I've summed it up here is exhaustive search of the key space. And if the keys are long enough, the goal is to make that key space infeasibly large. We'll come on to that in a second.

For asymmetric encryption, there are two principal kinds of mathematical problem or computing problem that are the basis for those algorithms. One is factorization of large numbers made from prime numbers so that they have as few factors as possible, and the second one is discrete logarithm solutions. Again, I'd have nowhere near the mathematical background sufficient to explain either of those two in any detail whatsoever. But there's plenty online if you really want to delve into the realms where sums contain more letters than numbers.

An important point here is that although quantum computing in both those instances for symmetric and asymmetric encryption can speed things up, it needs some help from something else. And that boils down to the problem I hinted at earlier. While you've got all those qubits lined up, and they're all flickering through all the possible values, you still have the challenge of stopping the reels at the point where they match the answer that you really want. So I'll say a little bit more about that in a second.

Okay. As I mentioned, with symmetric encryption, there should be no more efficient way of recovering the cleartext than to try exhaustively

all of the possible keys. In applied cryptography, Bruce Schneier set out some of the physics constraints on what would be involved in conducting exhaustive search if your keys for symmetric algorithms are long enough. There's a reference there if you want to follow up on that information.

But just to give you an idea, when we talk about symmetric keys, most strong current algorithms should be using at least 128-bit length keys and should be supporting 256-bit keys. Now, 256, in other words, all the possible values of 256-bit binary string is an extremely large number. It's more than the number of estimated atoms in our galaxy. It's more than the number of atoms in our planet, and it's more than the mass of the earth in grams. And as Schneier points out, that's an interesting place, that's an interesting number to stop off at because all of the silicon that we have at our disposal with which to make computers is a subset of the mass of the earth. So actually, if your keys are long enough, there might not be enough silicon in the planet to make enough computers to conduct an exhaustive search.

So with symmetric encryption, one of the key factors is that each time you add one bit to the key length, you're doubling the key space because you're going from, for example, 2 to the 36$^{th}$, 2 to the 37$^{th}$. And 2 to the 37$^{th}$ is twice the value of 2 to the 36$^{th}$. So mind adding one bit, if you double the length of the symmetric key, you're increasing the key space by its square. I have a graphic of that in a second. So much for symmetric encryption.

With asymmetric encryption, the process of encryption is reversible but it's not reversible using the same key that was used to encrypt the data. That's kind of counterintuitive once you've got it in your head the idea of symmetric keys. But with asymmetric encryption, you use one key to lock the data and a different related key to unlock it. The important part about asymmetric encryption is that once you have locked the data with the first key, reversing the operations in the same key does not give you the cleartext back, which means that you can safely publish that key in the knowledge that someone who has access to that key and to a piece of data that has been encrypted with it can't get the cleartext back because they don't have the corresponding private key that only the recipient possesses.

So one of the applications of public key cryptography is to validate the fact that the public key used to lock the data really came from the person you want to send it to. That makes public key cryptography a very interesting target for cryptanalysis, because if you can undermine that mechanism, you can undermine a lot of integrity of subsequent encrypted communications.

Okay. I'm keeping one eye on the clock here. I'm just about keeping up the time here. I mentioned earlier that it's all very well having a quantum computer, but you actually you're going to need some help in order to identify which is the right answer out of all the values of quantum computer can hold, can represent at the same time.

Here, for symmetric encryption, we need an algorithm called Grover's algorithm, which has the effect of reducing the key size for a

symmetric key to its square root. Now, given what I just said about the length of symmetric keys, effectively, what this means is that with quantum computer and Grover's algorithm, you've effectively halved the length of your symmetric key in terms of the security that it's providing.

That looks a bit like this. In the top diagram, you can see that if you had 128-bit key space and you halved the key length, what you've really done is you've reduced the key space to a square root. So in theory, a simple countermeasure to this is to double the length of your symmetric keys, which means that you've just squared the key space again and you've effectively neutralized the effect of Grover's algorithm. So, in theory, that's great. But in practice, replacing encryption systems that you've already deployed with new systems that use twice the key length is known to be a slow process, especially if it means upgrading all the devices that you've deployed and ensuring that all your communicating partners have upgraded their systems, too.

So with quantum cryptanalysis, again, I said there are two kinds of problems here. There's factorization or solving discrete logarithm problems. Here, what you need is a different algorithm called Shor's algorithm, which reduces the complexity of solving those kinds of problems. I'm not going to get into P—and P is a measure of complexity—but that's how the mathematicians will do it. Again, I've put a reference in that, some Wikipedia articles on complexity.

But as you can see from the graph here, you need substantial numbers of qubits in order to mount one of those attacks, whether it's factorization or solving discrete logarithms. Even for elliptic curve cryptography, which you can see in the blue line in the graph on the right, for long elliptic curve keys of 521 bits, you need roughly 10 times that number of qubits in order to mount a successful attack and solve the discrete logarithm problem in a viable time. And 5000 qubits is a heck of a challenge, which I think Paul is going to say a little bit about. But essentially, it's tough enough to get one qubit to stay stable for an appreciable length of time. I think the maximum at the moment is something like a few dozen qubits for a small length of time. But getting them to stay stable for long enough to get workable results out at numbers like 5000 qubits is currently astronomically infeasible.

So what do we do about it? Well, there is research going on into post quantum cryptography, looking for algorithms, looking for mathematical and computing problems that are harder to solve even with a quantum computer than the factorization problem underlying RSA and the discrete log problem underlying elliptic curve. If you want reference information on that, there is copious data available through NIST, which is currently most of the way through a standardization competition for candidate algorithms and their backups.

I guess the bottom line, at least at this stage is it seems likely to me that with classical computers, you'll be able to implement tougher algorithms quicker than quantum computing technology advances to this stage where it can start breaking. But of course, for that to benefit, you have to have migrated off any algorithms that are rendered

obsolete by quantum computing. And the key lesson here, especially in network environments, is implementing that cryptographic algorithm is not the same by any means as deploying it. So you'd factor that into the time calculation of when quantum computing might become a threat.

Okay. So really quickly, I just wanted to mention some other domains of quantum computing. Because quantum computing takes many forms and only one of those is quantum cryptanalysis, some of the others are really interesting and will get commercial traction way before quantum cryptanalysis does. Those include things like sensor technology, medical and diagnostic devices. Did you know MRI machines make use of quantum effects? Navigation by replacements for GPS, extremely accurate time sources that are far less fragile and cumbersome than atomic clocks, and so on.

I promised to mention quantum encryption and all I'm going say about that is if you get a quantum network, in other words, a network that is, for example, exchanging state of photons, you may well be able to use that to provide secure methods of key distribution, and thereby get by some of the symmetric key distribution problems that I mentioned earlier.

Dan, I see you have reappeared. I think I am out of time on that basis, but that's okay, because I'm out of slides, too. So, these slides are all available on the site. In fact, in the deck that I put on the site, there is some more just in case you wanted some more detail. And of course,

they've got links in if you want to take our reference and material, too. Dan, back to you.

DAN YORK: Thank you, Robin. Actually, I was just turning my camera on. Perfect timing, though. That was a great piece that was there. While I was sitting here, I did not know that MRI machines were used to quantum effects. Does that mean they see like part of my body and not part of my body at the same time? I don't know.

ROBIN WILTON: It's weird, isn't it? It took me a while to figure that out. But basically, what's happening is they use classical physics in the sense that they shove you into a huge magnet and then reverse the polarization every time. It's a bit like microwaving you. If you think of it like a lens, you've got classical physics microwaves going through you. In going through you, they produce a quantum effect. And then what comes out the other end has been affected at a quantum level, and so it looks a bit different to what went in.

DAN YORK: All right, well, maybe I shouldn't have asked that question. But I think the key point I took out of that certainly is that there's a lot that's involved with this on a whole range of areas. So I think the question that we have really here—thank you, Robin, for that and for all the links you provided. And for folks who are asking about those links in

the chat, if you download Robin slides, you can go and click those links and follow them right there. Oh, do we have a question?

ROBIN WILTON: We do but Paul already answered.

DAN YORK: Okay. I'm hearing—

KATHY SCHNITT: Jacques, you're coming through funny. We can understand you.

DAN YORK: That was a very high pitched—I don't know if that was a quantum entangled Canadian or something like that. I don't know.

ROBIN WILTON: I think so. Quantum warbling. The question in the chat was when I said astronomically impossible, did I really mean it? Well, no, not really. As Paul said, it's really a matter of how long it takes to make what's currently impossible possible, and the question is, how much time?

If you look at some of the graphs for advances in quantum computing and the number of qubits that can be kept stable for how long, you will see that it's on the upward trend. But my conclusion was having looked at how many qubits you need to stack up, we're still quite a

way off, even if there's a kind of step change. See how I avoided saying quantum change there?

JACQUES LATOUR: Yes. So that was a question. Can you hear me okay now?

DAN YORK: Now we can.

JACQUES LATOUR: Yeah. I had to turn off my quantum encryption.

DAN YORK: There we go.

JACQUES LATOUR: So impossible … So do you think it's more than a lifetime? Or is it 10 years, 100? Because having 5000 stable qubits is like having 1000 pigeon on electric wire, right? It's possible, but it'll never happen.

ROBIN WILTON: Yeah. I think it's like any other cryptanalysis question. It's going to depend partly on who is prepared to throw how many resources at it. So one of the comments I was making about other fields of quantum computing is that things like quantum metrology are going to be a far greater commercial applicability and attractiveness. But nation states are still going to be very interested in quantum cryptanalysis. So I

would expect them to be devoting significant time and effort and resources to improving quantum computing, therefore, quantum cryptanalysis.

DAN YORK: We've got one more question, and then we'll jump into our next presentation. Nils, if you want to unmute yourself, you're welcome to ask it.

NILS WISIOL: Robin, thanks for your talk. I've seen the graph that you've shown with the how many qubits it requires to be it broken. And I thought it was less for ECC than for RSA. So I would ask, do you expect it to be broken earlier? Sooner, or is there no essential difference?

DAN YORK: Robin, you're muted.

ROBIN WILTON: Nils, to be honest, I don't think I'm mathematically or physically qualified to answer that. One thing I did note was that there's a difference in the number of qubits that you have to apply to each bit of key. So, for example, for elliptic curve, you need roughly 10 qubits lined up in order to attack one bit of the key. And for RSA, you only need, I think, roughly 2n plus one where n is the number of qubits.

I think what I would recommend is—in the slide I put a link to a piece of Microsoft research on that topic that goes way into detail about exactly what kind of logic gates you need to assemble with your qubits in order to meet an attack, and they look at both discrete log problems and factorization. So if you're comfortable enough with the maths, I would highly recommend looking at that because it will give you a far more accurate raw data than I can give in my metaphorical way.

DAN YORK: Thank you, Robin. We'll have you stay around for the Q&A when we're done here. But with that, I want to bring it to the question of, so what does this have to do with DNS and why are we here? I'm going to turn that over to Paul Hoffman to give us a little bit of a session around that.

PAUL HOFFMAN: Greetings. Is that all visible and such?

DAN YORK: Yes, it is. We see you there.

PAUL HOFFMAN: Very good. So I'm going to start by answering Nils's last question, which is, look in the links that I'm going to be giving you later, this is definitely covered. And I guess I'm supposed to start with the obligatory joke, which Robin set me up for. By the way, Robin,

excellent description of quantum and such. Because, Robin, you started with talking about Schroedinger's cat and there's a wonderful quote from Stephen Hawking that says, "When I hear somebody talking about Schroedinger's cat, I reach for my gun."

So given that, let's bring it to the DNS. As Robin was saying close to the end there, what does this have to do with the DNS? Really, our major concern is the quantum computers in the future are going to be able to break some of the things that we're doing now, for those of us who are using DNS over TLS, anyone who has stored those sessions who has a cryptographically relevant quantum computer, one that's big enough, will be able to see what was in the TLS section. And, of course, with DNSSEC, which is based on signatures, somebody will be able to find the public keys for DNSSEC and be able to then forge DNSSEC responses. I'll talk about that more in a few slides.

Because there's all these questions, ICANN just published a couple months ago a paper that specifically is honed in on DNS, because as we saw from Robin's discussion, if you want to talk quantum, you can talk for years and years. So we tried to look at that. And don't worry, I've got links on a slide about halfway through a bunch of different links and I'll describe them for people who want to know more. If you don't need to know more, hopefully I can give you in the next 10 or so minutes a quick overview of why you should care about this if you care about the DNS.

So the major threat is that in the future, when someone can build a very large quantum computer, they can pick out the private keys that

were used in DNSSEC and TLS. Not possible today. So we are not concerned about whether this is conceptually possible. We've known for over 25 years that if a large enough quantum computer can be built, it will work for this problem. So if that happens in DNSSEC, this would allow somebody who has such a large computer to impersonate any zone because they know the private key of the DNSKEY, they can create their own RRSIG records they can impersonate.

For TLS, which is very different—again, DNSSEC, we're talking about signing TLS, we're talking about encrypting. Somebody who has recorded a session could figure out the key that was used to encrypt the session and then what was said. So these are the two parts that are the threat to the DNS, but it is really important to understand these quantum computers are not only not ready now but they're not even ready soon. We just don't know how long it will be. We know that it won't be anytime real soon. But because it's far off enough in the future, we can't tell.

When people are saying, "Well, we want to know exactly," think of it in terms of this. In 1960, we already had a little bit of VLSI happening and such like that. If someone said to you in 1960, "Will somebody be able to build a computer like this?" I'm holding up my phone. Unfortunately, Zoom is not cooperating on the focus. "Will somebody be able to build a phone like that in, say, 50 or 60 years?" And you could say no or yes. And you would be right only in one of the two. In 1960, you could not say, "I could build this phone in 2010." But obviously this phone got built in 2020. So we're really at that very early

part of being able to guess the future for when will somebody be able to build one of these computers, which will be a threat to the DNS. And to be very clear—I covered this a little bit later—it's a threat to everything, not just to the DNS. We're sort of the tail end of the economic part of the threat.

So why is this so hard? As Robin had said, it's very hard to get the qubits to do the things you want, and so what you need to do is to throw more qubits at it. So using today's cryptography, we would need a quantum computer that would have more than 10 million qubits. Now, I'm talking about physical qubits. As Robin pointed out, the number of essential qubits that you need, but you need a whole lot of qubits for each essential qubit. And we don't even know this number very accurately now. We're talking about needing computer with more than 10 million qubits. We just don't know how to make those now. The problems for making those are really, really, really hard. Not only will it take a long time to develop but building such a computer is now just financially infeasible. It's going to get better over time but we have no idea when it's going to be worthwhile.

Again, going back to the analogy of someone talking about VLSI in the 60s, you can say to them, "When will it be feasible to make a personal computer?" and somewhat no one would have said 20 years. I'm sorry, I shouldn't say that. Some people would have. Saying 20 years at that point would have seen crazy for some people and plenty for others. So this is the problem that we have with building large quantum computers is that the number of qubits needed and all of them to be

working exactly the way you expect is so huge. We just don't have an idea of how to do that today.

So here's the slide with all of the recent relevant publications. I'm going to go through this briefly. The top one is the paper that I wrote for ICANN, which is a sort of longer version of what we've said here. It includes a bunch of the stuff that Robin said and such like that, but it's quite short, like seven or eight pages. Because there were so little available a couple of years ago that was readable about quantum computing and Internet security, ICANN paid Dr. Hilarie Orman, who's a well known cryptographer and quantum physicist, to write an academic level paper on this. So this paper is like 40 or 50 pages. But if you really want to know a lot, this is a great paper. It's very up to date. It was only published a few months ago, and we used this basically as the source for us writing our paper. People expect ICANN write sort of fact-based things and show our work. Dr. Orman showed her work very, very well. You can follow through and such like that.

The next link is for a website, I'm going to show you the best part of it. It's a very short website that shows you where we are now and where we need to get to in order to be using quantum computing to break cryptography. So hold that but there's a link there.

There's also a great study by Ericsson research on quantum technology, a little bit wider than just the DNS but it's also well written.

So here's the chart that I just referred to, which shows you where we are now and what is needed to break encryption. I hope you can see

this on your screens, depends on the size of your screen. The “we are here” is over in the lower left-hand corner, that little gray area. “RSA is broken” is up towards the far right. There’s other things here. The two things I want you to notice is we’re in the lower left and what is needed is way up in the upper right. Also, both scales are log scales. We have really far to go. This is not like a straight line is going to be going across log, log. So we’re really, really far away.

Nils had asked, “Well, what about elliptic curve?” Elliptic curve is just slightly to the left of the RSA curve. So it is pretty much just as far as what we need. Again, this chart here is on the third bullet point here. Sam is updating it occasionally. This really should give you a feeling for how far away we are but that we know where we’re going.

As Robin said, we would like to prevent the problem. One of the things to note from this slide, if you look at the RSA lines on the upper right, notice how close they are together, even though they are for larger and larger RSA keys. What that tells you is that we can’t avoid the problem just by using larger RSA keys or larger elliptic curve keys. We really need to move to something different, which are the post quantum cryptographic algorithms that we strongly, strongly believe are not susceptible to quantum computers. We know that they are not susceptible to any of the known quantum algorithms today, we’re very sure of that. But of course, someone might discover different quantum algorithms later. But it turns out people have only been developing quantum algorithms very, very slowly. So we’re pretty good with this. But we can’t just drop them in right now, because they have much larger keys, much larger signatures. Also, right now, during the

standardization process, people are arguing a lot about which of them are better, and really, are they as secure as we say.

As a community, again, we're concerned with TLS and DNSSEC. For TLS, let's just follow along with the TLS folks. They have much bigger problems than we do, we don't need to participate in their arguments, we'll just do that. For DNSSEC, waiting for a good post quantum signing algorithms is probably our best bet because we have so little experience with the ones now. There are a lot of arguments and they're going to cause us problems by having large keys and such like that. The other problem we have right now is that all the cryptographers are looking at the TLS part, not the signing part. So we're going to have to wait for them to catch up a bit. We'll know much more in the next 10 years.

Here's my last slide. The threat is real. But it will only become actually real for us when these large quantum computers become useful, it really comes down to economics. The initial computer is going to be so big, so expensive, and so expensive to run that we in the DNSSEC world are not going to be the initial target. The initial target, of course, is going to be breaking state secrets, being able to trick people to go to the wrong website and such. We in the DNSSEC world can wait longer while the other parts of the cryptography world get better. Then we can follow along and change the post quantum algorithms.

So that's it for me. And I think I just hit the end of the time so let me stop sharing and hand it back to Dan.

DAN YORK: Thank you, Paul. That's a really good overview. I appreciate all the links you sent in there of why we should be concerned, what we should be thinking about. I have not seen any questions yet, and that's okay. We've cued up some time at the end for questions that are here. So, Paul, you can stay around here.

What I want to move to next is our presentation of one of—let's see. Are we ready here? One of the possible ways that we could defend against this—Paul and Robin both mentioned a number of different algorithms in development and some different ways around this. We have Nils Wisiol here to talk to us a little bit about one of the possibilities that's under consideration. I'll pass it to you, Nils.

NILS WISIOL: Thank you, Dan. Can you guys see my slides?

DAN YORK: We can.

NILS WISIOL: Awesome. Paul said you can't just put post quantum crypto into the DNS, right? Let's say it's a challenge. So we did just that and we implemented FALCON-512, which is one of the proposals in NIST competition into PowerDNS, just to see how it goes. I think I can skip the motivation. We've heard there's a thing called Shor's algorithm and it can factorize and that will be a problem in the future. I want to emphasize that DoH and DoT is not a solution to this problem because

it only provides transport security. It doesn't give us the security guarantees that we expect from DNSSEC. DoH and DoT can play a role—we will see that later—but not for security.

Why is it so difficult to find a replacement for the algorithms used in DNSSEC? Well, what we want from a signature algorithm in DNSSEC is fast validation because people running resolvers would say, "We need this fast. Otherwise, we need more machines." I've heard people are skeptical initially about deploying validation for DNSSEC because they fear that they don't have enough CPU time to do all the validation and such. So it is important that we have fast validation algorithms. It's important that we have short signatures because the packet size in DNS is somewhat limited. At least in older implementations or in some device, it's very limited. So this is something we need to keep an eye on. Also, we want short-ish public keys. I think that's a general thing when you look at post quantum cryptography in some signature variants, the pieces just explode if you compare it to what we're used to so far.

With this motivation, we just set out and we said, "Okay, our mission now is to create a real world working example of how it could look like." Why did we choose FALCON? Well, we choose FALCON-512 because Moritz Mueller and others, they looked at all the candidates and finalists that made it to the third round of the NIST competition. And they looked at the public key size and the signature size that you can see here in the table, and they also looked at the signing speed and the verification speed. Well, they determined FALCON-512 is really the only thing that comes close to match our requirements that we

have in DNSSEC today. The most important requirement that we're looking at is the packet size, and it should stay under 1200 bytes. That's because the maximum transmission unit is limited in some devices. Because usually, requests and responses are sent over UDP, and if responses get too large, there will be a fallback to TCP and that is problematic for a number of reasons because it may be not supported so well and so on.

So FALCON-512 has the smallest signatures and public keys combined. There's other candidates like Dilithium and Sphincs, I think. There's going to be a talk later about the hash space candidates here that have very small public keys but larger signatures. Based on the work of Moritz, we chose FALCON not because we think it's a good choice, not because we think it should be standardized, just because we think that's the best option that is available.

Additionally to looking at what Moritz said before, we also looked at algorithms they did not consider. So we looked outside the third round of the NIST competition, and we looked at qTESLA and MQDSS and LUOV and BLISS and SQISign. You can see pretty much the same picture here. Either the public key is too large or the signature is too large. We really liked the idea of using LUOV but then it came close to being broken, I think, summer last year so we had to drop it off the list. There's BLISS. But BLISS hasn't even been submitted to the competition, and I think that's because of patent problems or something. It also has similar metrics to FALCON so BLISS couldn't really win here.

One thing that you can see is, if you want to go to higher security levels, FALCON-512 is security level one. If you want to increase that, you could use BLISS to go to level three with almost no penalty and public key size and signature size. But again, it's not part of the competition, so it may not receive as much research attention. That's probably also not what you want.

Then there's SQISign which has small signatures and public keys, but it's extremely slow. Of course, maybe one day, there is a better implementation and speed could go generally up with time, but this is really extremely slow. Considering everything, we chose FALCON-512 as our proof of concept.

Let's compare beginning with key storage. We just used the FALCON-512 implementation from the Open Quantum Safe Project and we used that in combination with PowerDNS, because Open Quantum Safe Project has a fork. They maintain of the open SSL library and it knows FALCON-512, and PowerDNS also can be used against OpenSSL. So that was easy-ish to do. We just looked at the database storage in PowerDNS that was actually used and we compared it to the other choices that DNSSEC knows. You can see, it's larger than anything else. But I think given the circumstances, this is okay, it's feasible, it's not dramatic, even though it might be inconvenient.

I think more interesting is the performance. And here in the performance, you can see three different scenarios. On the left, we have how long it takes to generate a new key pair. In the middle, you can see how long it takes to sign something, and on the right, you can

see how long it takes to verify something. I've done those performance tests on my laptop. They might run differently on different CPU architectures and so on. I've used the PowerDNS integrated benchmark for that. But the verification I also tested in a network-based setup that is, so to say, more end to end, and for the verification, that shows exactly the same order. Of course, it has totally different times but it's the same principal result.

For key generation, you can see we're faster than RSA with a reasonable key size so that would be a win, I would say, for signing stuff. We're slower than ECDSA but we're also still faster than RSA. And for verification, we're faster than all the other choices except RSA. So I would say in terms of performance, this shouldn't be a problem because we've used RSA successfully in the past and we are also using ECC, so that is not a significant decrease compared to either of those.

The real problem is the packet size. We've done a couple of different scenarios here, and we just looked at the packet size that arrived at the responsible resolver. You can see that if you have, let's say, two IPv4 addresses or one IPv6 address and they carry a signature, then you stay below the magical boundary of 1200-ish bytes. So that's okay. There might be trouble with amplification attacks. I've have not looked very close into that but they are below the threshold. The problem is everything else we tested is above the threshold. So you can't just request IPv4 addresses and do nothing else. The resolver also needs to obtain the public key, and that will be a problem if there's no TCP support or packets get lost along the way, and so on.

Because the packet sizes are larger than the MTU, we need to consider there could be problems when we deploy this to the real world. We're planning to look at this to see how much of a problem it actually is. First of all, I wanted to show you—you can actually try it out yourself. Our mission was to provide a proof of concept. So here, you can test our domain name against Google resolver. You will get an answer with some signature and it won't be authentic data, because, of course, Google resolver doesn't speak FALCON. But you can also use our resolver that is available here. Then it will actually get validated and it will tell you, "Hey, I validated this and it actually worked." You can also play around with our website.

To conclude this, we looked at the storage, it's more but you can handle it. We looked at the performance, it's okay. It's either faster than RSA used to be or it's faster than elliptic curve used to be. We have our online proof of concept and the playground that you can test against. For compatibility, we think there will be a lot of TCP fallback so that might be a problem.

I just want to point out that in DoT and DoH, there's no such limitation. So that might help us to get larger keys or larger signatures into DNSSEC by just using DoH or DoT. For compatibility also, I think we shouldn't just rely on saying, "Okay, the packets are too large, we can't do anything about it." We actually need to find out is this a problem. So "How many validating resolvers do not support TCP?" is an interesting question or "How much of the existing DNSSEC infrastructure has a problem with such large packets if we just send them to them?"

I think other future work should include what other post quantum algorithms can be used. I mean, you've seen the table, it doesn't look really good. But I think the next talk in this panel is going to be on hash-based signatures.

I've talked about BLISS, the Merkle Tree is a hash-based thing so we will hear about that in a second. Then I think we need to talk about how do we make the root actually post quantum secure because it will require an algorithm rollover. That will be a huge project just by itself, I think.

Finally, I found Jason Goertzen who also did a FALCON-512 implementation in BIND. I've talked to him and we're planning to test if those implementations are actually compatible. With that, I have my last slide. That's it. Thank you, guys.

DAN YORK: Wow. That was fascinating. I see Jason is actually in the chat. He's there on that. Fascinating. I had one quick question. You were saying DoH and DoT, etc., are not done with the issue, and that's because they're already working over TCP is what you mean. They don't have the fallback issue because they're already over TCP?

NILS WISIOL: Yes, they don't have the packet size limitation.

DAN YORK: Okay. That's what I thought you were saying. I was just not clear on that. Well, thank you. It's great that you folks are out there doing this kind of research and digging into it. When Paul talked earlier about packet sizes in DNS and the challenges there, it was very interesting to see you showing exactly how big those packets can be, which is huge and perhaps beyond what we can really do. And certainly, with the realm of amplification attacks and so on, so forth there, it could be quite there. Any specific questions? I'm not seeing any, Kathy, at the moment. So let me go on then. And if you could stop sharing, we'll bring up our last panelist.

Andrew, frankly, is going to bring us through some of the work that they've been doing at Verisign, which overlaps with some of these other pieces here but gets into some specific areas around all of this. So, Andrew, I will turn it to you.

ANDREW FREGLY: All right, great. I'll bring up my presentation. Okay. Can you see it?

DAN YORK: No.

ANDREW FREGLY: All right. Let's see. Now you should be able to.

DAN YORK: Now we do.

ANDREW FREGLY: All right. What great talks by my fellow panelists. I'm going to try and minimize there, restating what they've stated, but I'll just touch maybe a little bit on them. First, what this talk will cover is, one, why is Verisign and our CTO organization considering DNSSEC for PQ algorithms for it right now? And then I'll talk about what we're focused on, which is right now is hash-based signatures, why we're looking at them. Now I'll talk about a little bit of our work related to hash-based signatures. Let me see if I can get my slides to advance. You know what? There we go.

Okay. I think this is very well covered by the others as to why we're doing this, the potential advent of cryptographically relevant quantum computers breaking the current DNSSEC algorithms. So that was interesting to listen to, Dan's and to the other talks. We talked about, well, why look at this now?

Well, first, we look at Mosca's formula, he says, basically, if you look at the time it takes to migrate to the new algorithms and the shelf time, and then you look at information that might be exposed before you transition, that gives you the timeline estimate. The biggest variable in this, which has been discussed, is when will this cryptographically relevant quantum computing come online? Now, in my research, I've seen everything from 15 years to 50 years. At the 15th year, if that were to somehow magically happen, that would be—we're right now in a period where we almost could be exposed and would have to be working at right now. If it's 50 years, we got plenty of time.

So the conclusion from that is basically, we don't really know when quantum computers will break current algorithms. My own opinion is most likely to be a longer time period, but we really don't know. And we also probably won't know when this capability to achieve due to the factors we would actually be pursuing this. So this leads to a conclusion that knowing what the PQC solution for DNSSEC will be is desirable, just in case we have the possible near-term availability quantum computing.

Then what I'll state here, I want to be clear that distinguish between preparation and adoption. So we really believe that there are two different things. And the second thing is that you can adopt without preparation. So suppose we're not ready and not prepared and then all of a sudden we discover, "The quantum computers can crack these algorithms," it would be impossible to shift right over to them. We believe that preparation should be done to provide agility for adoption.

Another reason for doing this is that we think looking at the DNSSEC use case at the same time other algorithms are being looked at for other use cases like the TLS use cases or the certificate use cases is important because we don't want to have to take algorithms or implementation decisions that are based on just those use cases and then try and make them work with DNSSEC.

A second reason is that algorithm diversity. So even without considering quantum computing—and I don't know if you heard Dustin Moody's talk yesterday on the IACR PKC, but he said that that's

the main goal that NIST has in their algorithm selection process is diversity in algorithm. And there's a reason for that. In case current algorithms get broken by attacks, those can be, in addition to quantum, there actually can be classical attacks on the current algorithms such as the attacks that were recently done against the Rainbow algorithm that was part of the NIST signature finalist.

So we have some observations that are driving our activities. Some of NIST candidate algorithms have—these have been discussed—these have large public keys, large signatures, have different CPU memory requirements. So we're looking at that. We've talked about the UDP MTU limits that as you exceed them, the unreliability, the transport goes way up, even with EDNS(0). And other surveys that show that EDNS(0) even support for that is not everywhere right now. And then, as noted before, we prefer small signatures. If we had to make a trade off in small signatures and key sizes, we prefer small signatures.

Then somehow note, too, this is something we've encountered. As we look at the transition time or having multiple algorithms at the same time, which is highly likely during a transition and possible even post quantum, is that having all the RRSIGs returned for all the different keys on a signed response, it exasperates the problem, the same thing for having large DNSKEY RRsets. So we've been looking at hash-based signature schemes as an option for PQC transition and resilience. And one of the reasons is we think the NIST approved the algorithms. There'll be plenty of traction for people to look at adopting them to the DNSSEC or usage and writing specifications for that. But we realized that when we look at the hash-based signatures, it's

interesting. These are already PQC algorithms, I guess. It's funny, they're looking at new algorithms, they're dropping replacements. And you can say, well, hash-based signatures aren't necessarily dropping replacements but they are post quantum algorithms. So they're based on basically hash algorithms that have been out there for a while such as SHA-2 and the newer SHAKE algorithms. These algorithms have had many years of crypto analysis so they're considered very strong algorithms and they're considered post quantum resistant.

So when we started looking at the hash-based algorithms, we started looking at the ones that NIST has approved, and these are HSS/LMS, XMSS/XMSS MT. You can look at NIST SP 800-208 which specifies these algorithms. And also there's some IETF RFCs to specify these algorithms. In addition to these algorithms, we've looked at SPHINCS+, which is a stateless hash-base signature algorithm that is still in the NIST competition which we think will actually get adopted. We'll know very soon as Dustin said the announcements are coming out any day now.

We also have efforts related to a novel scheme first proposed by Burt Kaliski, which we refer to as synthesized zone signing keys, and I'll discuss that a little later on and I'll discuss the other ones. Not much time I've got. I'm running a little one time.

DAN YORK: You've got about five minutes left there.

ANDREW FREGLY: All right. The Merkle trees—I'll just talk to this. So for those of you, I'll just say for a stateful hash-based signature thing, they have what's called a one-time signature for key pair, that's private and public keys. The thing is about the one-time P cares, is they can only be used once because repeated use will expose the private key. Actually, elements of the private key are a part of the signature. Lamport invented that technology. So Ralph Merkle invented the Merkle tree approach, which is basically, well, how do we solve that instead of sending them one at a time? We'll put a Merkle tree and put hashes of them as the leaves of a Merkle tree.

The Merkle tree is a binary tree with each node as a hash. And so you hash the leaves together and create the parent nodes until you get to the root public key of the tree, it becomes the overall public key. And you can verify a public key that's given in a signature is an element of the overall public key just by re-traversing that path using what we call an authentication path and rehashing it. And then at the end, seeing if the key you come up with matches the public key.

Then we have another concept. That was the original Merkle tree but it had some issues because you have to generate it all at one time, and it doesn't work well in distributed environments. Particularly, there's subject to restrictions that you can't export private key information, so you can't have it across multiple signing systems. So in this case, you can have a top-level tree that you generate at the start of key generation, you generate all the OTS public and private keys. And then

you generate lower level trees and sign them with those keys. You generate them as needed. This approach works well for distributing signing operations across multiple signing systems or, for instance, if we were signing com, we want to sign it multiple locations. It's an approach we would have to follow.

So what we've done just in the last few days, we've submitted a draft for stateful hash-based signatures. That'll be something we'll hopefully cover at the IETF meetings coming up next week. It's just a basic draft focusing on the two algorithms I mentioned.

Now, one of the things we haven't done, as we realized there's implementation issues with these hash-based signatures managing state as a challenge, how to do the hierarchical tree structures, how to deal with things like ceremonial signing, and that sort of thing, and we're considering writing a future draft that covers those issues. But I think a little experience with the actual algorithms will help with that, too.

I'm going to talk about the method Burt came up with. It's what we call synthesized zone signing key. So this is kind of an interesting approach. Basically, for DNSSEC, you would take all the RRsets that you would normally sign and you hash them. And then you put those hashes as leaves of the Merkle tree. And then using the same authentication path mechanism used for HSS/LMS or XMSS MT, the root node of that tree then becomes a public key for all of those RRsets and for signatures on them. So that's kind of a neat approach because it cuts the way down on a signature size, because you don't

have the one-time signatures in the signatures. All you have is the authentication path that's really needed.

So with that, I'll talk about going forward. What we're going to do, so we're continuing to look at across all the NIST candidates plus these algorithms I've mentioned, their characteristics, their resources. And we're looking at creating test beds in cooperation with other resolver providers and open-source providers, and focusing on operational experience and ecosystem readiness, too. This will take planning for the collaborative activities. You can see our first effort and standards. Also, transition is going to be something to be thought through, too, as we know from the root key rollover, that's going to take a lot of coordination and effort. So I guess with that, you can read the rest of the slide. I'm done.

DAN YORK: All right. Thank you, Andrew.

ANDREW FREGLY: A lot to cover.

DAN YORK: Yeah, you did. Again, thank you for providing slides that have lots of links that people can follow up on.

ANDREW FREGLY: Let me show you that. Yeah. This is a good basic set of references for the material I covered.

DAN YORK: I think one of the challenges we all have right in this space is that there's so much to learn and so much is there. Well, thank you for that. So if you want to stop sharing, then I would also ask if the other panelists want to turn their cameras on. We will open it up to questions. I want to also thank the 90 people or so who have stayed on through this whole session to listen to all of this information in pieces. So this is time for questions and concerns and things. While we're waiting, I will throw one out. Paul, Andrew, Nils raised the question of, what about PQC for the root?

PAUL HOFFMAN: Any algorithms that the DNSSEC community chooses to add would only be valuable if we also put them at the root. That is that if we choose algorithms that are not useful at the root, that doesn't help because then in this future where someone has a cryptographic irrelevant quantum computer, it doesn't matter if .com is signed by a better algorithm. If the root is not, they can use that to falsify DNSSEC responses. So whatever we're going to choose, whether it's one algorithm or many post quantum algorithms, at least one of them has to be useful in the root.

DAN YORK: Yeah. This is a good reason why we're having this workshop now, because it might take us 10 years to get to the point where we can do all that into the root at some point, right?

PAUL HOFFMAN: I'm going to jump in just wearing a different hat, which is that I work with the IANA folks on some of these questions. It is not clear yet whether doing an algorithm change is actually much harder than doing the key change that we did in 2017-2018. It might be harder, it might not be. I know that IANA really cares a lot about this and is looking into it. It's not like that they're pushing it off. This isn't accurate thought for them. But we aren't going to be in a rush. And therefore, we can do plenty of things beforehand, including setting up test beds and such like that, that we'll be able to use to figure out how well is this going to go.

Andrew said—and I put something in the chat—about if we have an implementation considerations document that's active and full for all of the various proposed algorithms. That's going to help a lot because we can actually test bed a lot of the things. I mean, Nils already started doing that, right? He started off early, but that kind of thing where—Nils could have come back and given a presentation going like, "Oh my God, this was horrible," and that would be valuable information. So us looking now, without any thought of choosing anytime soon, is still very good.

ANDREW FREGLY: Yeah. This gets back to a point I was making about the stateful hash-based signatures and also the synthesized hash-based signatures that Burt came up with. You look at them, they're very different operationally and they really vary based on the zone size. The issues you would have for root are very different than you would have for com because of how slow root changes and the fact that it's ceremonially signed, you might have a small zone operator—I talked about stateful. And even hierarchal trees and the small zone operator not subject to these standards may say, "Hey, I just need a single hash Merkle tree and I'm not subject to miss standards. So I'll back it up if that machine crashes. I'll just reload my private keys on another machine." But other operators can't do that. So there's a lot of issues like that to be fleshed out.

DAN YORK: Yeah. Some of the people involved in this workshop put together a draft number of years back about the observations on the challenges of rolling out new algorithms across the DNS infrastructure, and certainly on the verification side, that has large challenges. Although as things move toward DNS over HTQ, whatever, some of those challenges become less for those resolvers. But that's a whole other thing. Robin, you've got your hand up. Go ahead.

ROBIN WILTON: Yeah, just a question. This probably should have occurred to me sooner. But this thought about the implications of replacing an algorithm of the root rather than just the key. I mean, I know, even not

having been involved in it, the key rollover process is quite an involved one and needs to be carefully documented and needs to be carefully audited, and auditable and recorded and so on. Does this mean that in the migration time that Andrew included in his equation of when you need to start doing things, during that migration period, we need to factor in, have we actually written and tested auditable algorithm rollover ceremonies? Is that a procedural thing that we need to do?

PAUL HOFFMAN: It would be if at the time that we roll the algorithm to a post quantum algorithm, if we haven't already done another algorithm change in the root. But given that we have plenty of time to do it, it may be—and again, I am absolutely not speaking for IANA here, they get to do their own thing. They're in a different group and I think we're both happy about that.

DAN YORK: Paul Hoffman said.

PAUL HOFFMAN: Yeah, right. But if IANA has already done an algorithm roll, for example, to an elliptic curve algorithm or to as yet undefined new way of doing RSA, then all of that will be documented, Robin, and so at that point, it will not be an issue. In my mind—and, Andrew, tell me if you think this is correct because you've been looking at this, again, the operational issues. In my mind, the biggest issue for the root and for the large or the very public zones is whether you can get a hardware

security modules, HSMs, that will have the new algorithm in them. That is that that takes a while and you're sort of stuck with whoever the HSM vendors who you trust are to care about you as compared to, to care about the web PKI and such like that. Andrew, since you were the one, you put up more on that, is that in your thinking as well?

ANDREW FREGLY: Yeah, that's part of it. We have our forward plans. This really needs to be an industry-wide collaborative effort with all the different players in here, from open-source, HSM operators, resolver operators, and the registries for the different zones. If we don't all get on the same page, it's going to be really hard. So the sooner we start thinking about that, the better.

DAN YORK: We've got just a couple of minutes left. I've got a question but, Steve, you raised your hand, so come on in.

STEVE CROCKER: Thank you. I just want to add about this algorithm rollover. Certainly, the hardware security modules are a major factor. But with respect to changing an algorithm for the root key, in my mind, a really big factor is the support for the algorithm in all of the resolvers around the world. And in order for that to happen, it has to be in the libraries. So you've got a supply chain issue kind of thing in which the algorithm has to be defined and everybody has to be comfortable with that, then it has to make it into the libraries. And then those libraries have to be

incorporated into the resolvers that are then operating around. When you talk about all the systems that are out in the field, how often are they upgraded? So now your timeline is quite long with respect to trying to migrate a new algorithm. So I think Paul's point that we definitely want experience of just doing—I'll call it the easy algorithm changes before you get to the quantum issues. That's my key point.

ANDREW FREGLY: One last thing. Moritz Mueller is going to give a talk, I think, in one of the next sessions. He studies things like algorithm support across resolvers, and that might be worth listening to give you a feel for just how tough those problems are.

DAN YORK: Yeah. Certainly, one of the things we're looking for. I have a question for the panelists. Maybe I'll go with Nils on this first. We've highlighted this issue of key length and the struggle with there. Do you have any thoughts about how we can help address that kind of issue?

NILS WISIOL: I would like to first find out how big the issue is. Because I think it's known that some resolvers, some authoritative servers do not support TCP. But maybe those are the ones we don't really care about. I mean, if we deploy a new algorithm and some authoritative server doesn't support TCP, then they won't support the new algorithm either, right? So that's not a loss in terms of this upgrade. So what I would like to do—Andrew, you mentioned that earlier—I would like to cooperate

with resolver vendors and see how is their support status in terms of very large packets.

ANDREW FREGLY: Yeah. I agree with you. So overall thing, the trade offs of all the industry players, when you're operating a really large zone where you have to deliver billions of responses a day, our considerations are maybe different than smaller operators. So getting us all together and working through this is critical.

DAN YORK: All right. Well, with that, I don't have any more quantum jokes, because if I did, they would be both funny and unfunny at the same time. So we'll leave it at that and thank our panelists for this. We are at the top of the hour and at the approved time when we are supposed to end the session for ICANN73. So thank you very much for this. We are now going to be going into a break period for 30 minutes. And so we will resume in the DNSSEC Workshop Session 2. Mr. Crocker will be back in here to lead us on a panel around DS automation with a host of characters around that.

So I would encourage you, if you listen to this, please, these materials are out here, take a look. Look at them, share the links, help other people understand around this. As we've all said here, it's going to be a long path to get there and it's the time to be starting thinking about this and certainly now. I would encourage you to stay around and listen to Session 2 and Session 3 today. Thank you all for your

attention, and thank you to our panelists for a great set of sessions. Thank you.

KATHY SCHNITT: Thank you, Dan. Please stop the recording.

**[END OF TRANSCRIPTION]**