QUANTUM-SAFE TRUST FOR VEHICLES

The Race is Already On

**A DISCUSSION WITH
MICHAEL GARDINER,
ALEXANDER
TRUSKOVSKY,
GEORGE NEVILLE-
NEIL, AND
ATEFEH MASHATAN**

The theory of quantum computing has been with us for nearly three decades, courtesy of a quantum mechanical model of the Turing machine proposed by physicist Paul Benioff in the early 1980s. For most of that time, the notion has seemed more a far-off vision than an impending reality. That changed abruptly with a 2019 claim by Google AI, in conjunction with NASA, that it had managed to perform a quantum computation infeasible on a conventional computer.

While many have eagerly anticipated the new vistas that could open with the arrival of quantum computing, cryptographers and security experts have not generally shared that enthusiasm since one of the most anticipated quantum advantages comes in the area of integer factorization, which is critical to RSA (Rivest-Shamir-Adleman)-based security. Also, as far back as 1994, MIT mathematician Peter Shor developed a quantum algorithm capable of solving the discrete logarithm problem central to Diffie-Hellman key exchange and elliptic curve cryptography. Now that it seems quantum-computing capabilities could

become commercially available within the next decade or two—likely in the form of cloud-based services—security professionals have turned with an intensified sense of urgency to the challenge of how to respond to the threat of quantum-powered attacks.

One domain where this is particularly true is in the automotive industry, where cars now coming off assembly lines are sometimes referred to as “rolling data centers” in acknowledgment of all the entertainment and communications capabilities they contain. The fact that autonomous driving systems are also well along in development does nothing to allay these concerns. Indeed, it would seem the stakes of automobile cybersecurity are about to become immeasurably higher just as some of the underpinnings of contemporary cybersecurity are rendered moot.

To explore the implications of this in the discussion that follows, acmqueue brought together some of the people who are already working to build a new trust environment for the automotive industry: Alexander Truskovsky, director of technical strategy at ISARA Corporation, where efforts are being made to develop quantum-safe cryptographic roots of trust; Mike Gardiner, a solutions architect at Thales who has been central to efforts to tailor quantum-safe protections for the automotive industry; Atefeh Mashatan, director of the Cybersecurity Research Lab at Ryerson University; and George Neville-Neil, director of Engineering Operational Security at JUUL Labs, who is better known to many as Kode Vicious.

Now that our cars are becoming smart, they have essentially turned into data centers on wheels.

ATEFEH MASHATAN What do you see as your greatest concerns when it comes to quantum vulnerability in the automobile industry?

MICHAEL GARDINER One of the big concerns has to do with over-the-air software updates for smart cars—like a Tesla, for example—where somebody with a quantum computer could potentially issue malicious firmware while creating the illusion it comes from the manufacturer. There's also risk associated with the telemetry data the car sends back to the manufacturer, which could be intercepted or tampered with to make it appear the vehicle went somewhere it didn't actually go.

AM Why should we even live with the exposure associated with over-the-air updates?

MG Now that our cars are becoming smart, they have essentially turned into data centers on wheels. Which is to say they're now increasingly composed of software components, all of which contain bugs just by their very nature. Auto manufacturers can use software updates not only to deliver new features that keep the car's general entertainment system up to date, but also to correct defects as they surface in other systems.

ALEXANDER TRUSKOVSKY Just to provide some sense of scale, vehicles such as Ford's F-150 come with more than 100 million lines of code. You can easily imagine a fair number of bugs to deal with there. It's not really a question of *whether* software updates will be necessary, but rather how many and how often. Typically, you would rather not burden the owner of the car with the expense and inconvenience of coming into the dealership each and every time those updates need to be administered. It's also

anticipated that by 2022 each vehicle sold will have some degree of autonomy built into it. This, of course, makes it all the more critical that there be some mechanism in place for updating that software in a prompt and efficient manner.

GEORGE NEVILLE-NEIL It's one thing to say a car has 100 million lines of code, but most people who build systems containing both safety-critical and nonsafety-critical components are smart enough to know they need to separate those things from each other. How confident are we that over-the-air updates for the safety-critical components won't end up getting bundled along with those for the entertainment system? I ask since that entertainment system is just one big hideous Linux box full of every open-source library some clown wanted to include so people would be able to play music, videos, and games in the car.

The brake system, on the other hand, is something that was presumably written by adults and ideally has been firewalled off from everything else—and not just by a digital firewall either, but also by an air gap. I trust there will be software updates for those safety-critical systems that are sent out separately from those delivered for the car's general entertainment system.

MG In the automotive industry, a lot of these components were coded a long time ago and haven't necessarily been looked at recently or vetted by third parties. So, the entertainment systems you find in cars now generally are able to talk on the same CAN [Controller Area Network] bus that the safety-critical systems use.

GNN That's a little disturbing.

Vehicle design cycles are lengthy, generally five to eight years.

AT I agree, but at this stage, over-the-air updates are mostly used just for the entertainment systems. Some manufacturers such as Tesla can enable some other functionality via software update, but, for the most part, only the entertainment systems are being updated in this way. Also, with the shift to greater computerization, many vehicles are now being switched to different networking systems. Which is to say you're going to see more Ethernet-based communication between components since a CAN bus simply cannot handle the load that comes along with the current autonomous-driving-system requirements.

So today, vehicles are being designed to be updatable as well as to accommodate some more advanced computer systems. But bear in mind that vehicle design cycles are lengthy—generally five to eight years—meaning that vehicles set to debut five years from now have already been designed.

AM What's being done to secure software updates for cars at this point?

AT If handled at a dealership, a mechanic can use a USB key to download the software update, generally without signatures—even though that's not an advisable practice. Over-the-air software updates, in contrast, absolutely require code-signing, and that calls for a public infrastructure with the trust anchor being a root certificate embedded in the vehicle—where the private key belongs to the original equipment manufacturer. With that in place, updates might be delivered to cars in much the same way they're currently sent to mobile phones or laptops in the sense that they would be digitally signed and

the vehicle then would take additional steps to verify the authenticity of the software before applying it.

The problem is that the embedded trust anchors at the heart of this system are based on classic public-key cryptography, which will be easily broken once attackers are able to use quantum computers. Changing out those dated trust anchors for new ones that have been hardened against quantum-based attacks will require vehicles to be brought in for servicing. In some cases, that might be accomplished easily just by updating the public key, but more often, the upgrade will require some sort of hardware replacement.

This takes us to another problem related to the emergence of autonomous vehicles that include sensors that talk to ECUs [engine control units], which in turn talk to the brakes and the steering system so the vehicle knows where to steer and when to brake. In that scenario, there will be messages that are relayed between different components and need to be authenticated so the ECU knows they are indeed coming from the vehicle's actual brake sensor or collision sensor—and *not* being impersonated by some hacker trying to take control of the vehicle.

All of which is to say that this is a zero-trust infrastructure where every single message needs to be authenticated and autonomous driving decisions must be fully authenticated. The cryptography being evaluated for use in this environment has yet to be standardized by NIST [National Institute of Standards and Technology]. Still, while some of the parameters of the core quantum-safe algorithms can be modified, the fundamentals of those

algorithms—that is, the key sizes, the speeds, and the ways in which things are executed—are not going to change. This means these algorithms can start being tested on vehicle components so that auto manufacturers will be able to start releasing new models that include hardware capable of supporting post-quantum cryptography as soon as possible.

Also, in parallel, work can begin on embedding quantum-safe trust anchors in vehicles since the math used for code signing is essentially ready to roll today. Then, a few years from now, once the final standards become available, that quantum-safe software update channel we've been talking about can be used to supplement the trust anchors with any additional quantum-safe functionality developed in the interim, most of which is expected to relate to requirements for autonomous driving.

AM What happens if the NIST standard proves to be not entirely compatible with the quantum-resistant algorithm you're currently working with?

AT You have to hedge your bets, meaning you need to provide for every type of crypto algorithm—lattices, multivariate, code-based, hash-based... you name it. If the lattice-based approaches prove to be broken, then you need to be ready to employ hash-based and multivariate. So, you really need to be able to port all of them.

AM Beyond over-the-air software updates, what should auto manufacturers be particularly concerned about once quantum computing becomes commercially available?

AT Actually, there's another matter related to software updates we should talk about first. In the case of autonomous vehicles, there are occasions when the

manufacturer needs to send various authenticated commands to the vehicle. Providing for the security of those commands is pretty similar to what it takes to protect software updates—which is to say, both need to be authenticated in the same way.

The sorts of commands I'm talking about are those that might be sent to an autonomous vehicle following an accident. In that event, an authenticated command could be sent to the vehicle to direct it to a particular service facility or to get the car to move itself out of traffic, over to the shoulder of the road. Clearly, these commands need to be quantum safe as well. I'm talking largely in terms of authentication here, but encryption also plays a big part since we need to provide privacy protection for the user.

MG There's another aspect of this: Because users will have connectivity to smart vehicles from their mobile devices, any commands they send and any information the car sends back to their mobile devices will also need to be protected for privacy. There's a lot of potential here for hackers to obtain sensitive private information.

AM Are these communications between users and their vehicles currently protected by some form of encryption?

GNN My impression is that communications within a car are not encrypted as yet, nor are they likely to be in the near future. They really ought to be, given how many people have tapped into the CAN bus and now will start tapping into the Ethernet. But I haven't seen a standard that says encryption for this is going to be required.

MG Yes, that's my understanding as well.

GNN Let's also not forget that a huge amount of location

data goes to and from these newer vehicles, and there definitely are a lot of issues with that.

AM Yes, and that goes beyond privacy concerns since access to that location data could even conceivably be used to enable abductions or other violent crimes. Is that something people working on post-quantum security are thinking about?

MG Quantum-enabled attacks would be able to negate the privacy of that telemetry information completely since it's protected only by asymmetric cryptography. In any event, most security efforts are focused on the integrity of software updates at this point.

AT Yes, that and vehicle safety. In the case of a car with an autonomous driving system, information from other vehicles and other sources about, say, a collision just up the road could alter the vehicle's driving instructions. But that data is forgeable, which clearly represents a threat to the safety of the occupants of that vehicle.

GNN In light of this and the other security risks we've been discussing, what do you see as an optimal timeline for getting quantum-resistant cryptography and PKI [public-key infrastructure] deployed throughout the automotive world? And how does that compare with what actually seems feasible?

MG Even if we were to start deploying right now, we would be at the mercy of supply-chain issues in the automotive industry. It would take roughly five years at the going rate to get these sorts of design changes into a car that's in production.

AT The average on-the-road lifespan of a vehicle is about

11½ years. Then we need to work our way backwards from that to account for the five years or so it takes to design a vehicle. That means the design decisions being made today ought to still make sense 16½ years from now. The recommendation from NIST and other organizations is that mitigations be in place for quantum-computing threats by 2030. Clearly, there's already a fair amount of urgency when it comes to the question of when we should start quantum-proofing vehicles.

For all the pressure to move quickly, the automotive industry can probably be counted upon to proceed stepwise toward the production of vehicles provisioned for quantum-safe trust. Almost without a doubt, the first of those steps will focus on ensuring that the embedded compute devices installed in cars are actually up to the challenge.

That alone will represent quite a departure for automakers that historically have relied on the least expensive off-the-shelf microcontrollers available. But that simply won't suffice when it comes to providing for an array of complex quantum-safe encryption algorithms or the throughput demands that autonomous driving systems are sure to place on controllers required to communicate continuously with a variety of sensors.

GNN Let's talk about some of the challenges the automotive industry will face once it comes to implementing quantum-resistant PKI. It won't be exactly like deploying PKI within a data center or for something

that's always online. What are some of the key differences you've been working through?

MG When you're thinking about something like the trust roots for conducting financial transactions by way of a web browser, all of the trusted CAs [certificate authorities] you're going to encounter are ones that have previously been agreed upon by the browser makers and the CA/Browser Forum [a consortium of CAs and vendors of browser software, operating systems, and other PKI-enabled applications]. The rules around how you can use those certificates and for which keys and for how long have already been established.

In the automotive space, there's no equivalent to that. An automotive manufacturer ought to be able to roll a quantum-resistant PKI into its vehicles early on without first needing to obtain broad industry acceptance.

GNN Does that actually make things easier? Or does it just end up looking like the same problem that surfaced back when all the browser people had to find some way to agree?

MG I don't think so—at least not until the industry starts talking about vehicle infrastructure or vehicle-to-vehicle communications. That will require wider industry agreements on what should and shouldn't be allowed. But, for now, if we're talking just about firmware updates for some particular car model or what it takes to secure telemetry information between the manufacturer and the car or between a user's mobile and the car, that can be handled on a manufacturer-by-manufacturer basis.

GNN Given that, what does the roadmap look like for rolling out these protections? You say we're talking about

defending objects that have an expected 16½-year design span. So, if today is day one, what does the next year or two need to look like for automakers in terms of implementing something along these lines?

AT First, they should be able to do at least a couple of things in parallel. One is that, as they're working on a new vehicle design, they can migrate their CAN bus to Ethernet while also updating all the embedded compute devices that serve as their controllers. These are things they can either design themselves or shop for off the shelf. Either way, it will be necessary to evaluate these devices to make sure they're capable of supporting all the available encryption algorithm families. Some of those algorithms might not be used for years, but the automakers should at least be confident that the hardware they're installing today will be capable of handling them.

At the same time, they can also ensure that their current software/firmware update capabilities will be able to take advantage of the most secure algorithms now available for that purpose—specifically, stateful hash-based signatures.

With both of those goals accomplished, an automaker will have assurance that it has hardware capable of supporting quantum-safe cryptography over the long term, along with a quantum-safe channel through which to push additional quantum-safe functionality over the years to come.

AM Given that 16½-year design span for cars, will the compute devices that auto manufacturers are currently embedding be up to all that?

AT That's a good question. A lot of the algorithms we'll see

The automotive industry is expected to save \$35 billion by doing software updates over the air rather than handling them in person at dealerships.

in the near future will be pretty complicated. And yet, auto manufacturers typically won't spend any more for these compute devices than is absolutely necessary—meaning the devices they buy are usually quite limited. This could prove interesting since the quantum-safe algorithms are definitely going to be far more demanding than the encryption algorithms they're currently running.

AM Can you quantify that?

AT It all depends. Some algorithms, like the lattices, run pretty fast, but they also have large keys and signatures. And then you have algorithms like supersingular isogenies that have much smaller keys but run much slower.

Since one of the considerations here has to do with the operation of autonomous vehicles, some thought will also have to be given to throughput. That is, you need to be able to handle perhaps 100 messages per second since there's always going to be some number of sensors talking to some number of controllers—and all of that has to happen in realtime. Also, some of those messages will likely be encrypted or require signatures, which is going to add considerably to processing time. But the manufacturers still have to make sure they can meet those realtime requirements. That could prove to be quite a challenge.

AM What about that other potential challenge—over-the-air updates? How many automakers are likely to start moving in that direction?

AT I've read that, over the next couple of years, the automotive industry is expected to save \$35 billion by doing software updates over the air rather than handling them in person at dealerships.

AM Has anyone done a risk assessment of the auto

industry's potential exposure to quantum-enabled attacks?

MG We've rated the potential as low in the short term, moving up to medium to high over time. But because we see the impact of any attack as being critical, we're treating this as a medium risk at minimum, even over the short term.

GNN What is the implementation timeline for getting quantum-resistant PKIs out there? What needs to happen first? And when do you think that's going to happen?

MG I think the first thing automotive manufacturers are looking to attain is increased cryptographic agility out of the resources in their cars. Which is to say that right at the top of their list is gaining the ability to handle quantum-safe algorithms, firmware updates, and telemetry communications. In terms of what happens inside the vehicles, that's probably less of a concern for the time being simply because that requires physical access.

Anyway, just the sourcing of components capable of handling the increased load will, in itself, represent a huge change since these organizations are accustomed to looking at just small microcontrollers that offer the essential built-in functions but little beyond that. Now they're going to have to think more about the future without knowing exactly what that future is going to look like or how much extra capacity that's going to require. I imagine it's going to take them four or five years to go from planning to getting something on the road.

GNN Do you think the path is at least somewhat going to resemble what happened back when the concern was embedding secure compute elements in desktop systems? If you were looking to do SSL [Secure Sockets

Layer] 10 or 15 years ago, you had to add specialized cryptographic components to your server. Do you see the first push here being made with the same microcontrollers that automakers were using before, along with some added cryptographic components? Or do you think they're going to need to ditch those microcontrollers and move up to full-on modern processors that include built-in cryptographic instructions? If so, since all those cryptographic instructions now are asymmetric rather than quantum-resistant, how's that going to work?

MG At first the automakers are either going to have to build in more general-purpose compute devices so they can achieve the required flexibility, or they're going to need to look at FPGA [field-programmable gate array] technology in order to solve that requirement. That's because all the ASICs and other hardware out there right now may not be able to handle these new quantum-safe algorithms. There are some other things they could try, but those might not fit with whatever the standard for this proves to be by 2024.

GNN FPGAs—or anything else along those lines—are going to represent quite a cost bump. Do you think the automakers will be willing to take that on?

MG I'm not sure. But I suppose they could consider relying on a symmetric key scheme that's internal to the vehicle and then try to handle integrity and encryption that way. With that approach, they might be able to get away with just one centralized FPGA that's responsible for all the translation between the internal car world and the external world. That still probably wouldn't line up with whatever the standard becomes within the next few years.

Since quantum crypto standards have not yet crystallized, we'll likely see adjustments on many fronts for years to come. But one thing is certain—quantum computing is coming. And it's no longer comfortably far off in the distant future.

Reassuringly, though, organizations are coming to realize that they can't afford to be caught flat-footed once that day comes. From experience, they already know it takes considerable time and effort just to move from one encryption algorithm to another. The shift to quantum-safe algorithms will involve far more than that, and the stakes when it comes to getting everything right will also be much higher.

AM In terms of anticipating challenges ahead, are there any lessons to be learned from looking at some of the cryptographic changes made in the past?

MG Probably so. SHA-1 (Secure Hash Algorithm 1), for example, has probably been broken for a few years now, and yet there still are things out there in the wild that continue to use it. Unless we start preparing for the post-quantum challenge now, we're going to find ourselves in that same position, where the industry continues to rely upon cryptography that's no longer effective well after the arrival of quantum computing. As a general rule, the cybersecurity industry tends to be quite risk-averse when it comes to tackling new things. But this is one of those cases where people need to consider that they're likely to put themselves at a much higher risk if they don't start preparing now, since there's no doubt that quantum computing is coming.

AT Another good example from the past would be that the NSA [National Security Agency] announced Suite B Cryptography in 2005, and it was mandated that all government agencies should implement it. But then, 10 years later, came another announcement saying that anyone who hadn't already completed the migration to Suite B ought to pause and wait for the new quantum-safe standards to emerge. Which is to say that, over the past 16 years, neither the U.S. government nor the Canadian government has managed to complete its migration to Suite B Cryptography. This gives you a pretty good idea about just how long it takes really large organizations to migrate from one algorithm to another.

Some of these quantum-safe algorithms behave quite differently from what security experts have become accustomed to. That is, a typical signature algorithm has one private key and one public key. The private key signs while the public key verifies, and you can do that as many times as you want. With the quantum-safe algorithms, you also have one public key that verifies the signatures, but the private key is very different. Basically, it's a collection of one-time keys that have been organized in a binary tree, where each key is a leaf and the root of the tree is your public key. During a signing verification operation, you sign with one of those private keys, but then that key has to be discarded.

What you effectively have is a large number of exhaustible private keys that you need to manage and maintain the state for. This just hasn't been done before. So now, PKI organizations that use these schemes to create root certificates and sign entity certificates need to do

a good deal more planning than before. That's because, in the case of a root certificate, the height of the tree determines the number of potential signatures, and there's a tradeoff between that number as it grows and the amount of efficiency that can be achieved. This means you need to plan so you can determine the maximum number of signatures you want to accept from a particular key. You also need to provide for high availability and plan for disaster recovery.

With a large tree of private keys, it's also important to be careful about state. If you back up in multiple locations, you need to share that state across all those locations, which is very impractical, of course. The point is that you end up changing your whole operational plan just because you're now dealing with an exhaustible key. At some point, even though the public key that signs your certificates is still valid, you may run out of private keys. It's easy to see how people who aren't accustomed to these sorts of issues could become pretty frustrated with the quantum-safe algorithms.

MG The other big cryptographic change people need to get used to is the lack of a general-purpose, jack-of-all-trades key such as RSA. Today RSA is used for encrypting and for signing and for exchanging keys. But these newer algorithms don't really allow multiple operations with a single key, or even with a pair of keys.

GNN Just exactly how large are these trees?

AT There actually are both single-tree and multi-tree variants. The single-tree variants range from tree height 5, where you've got 32 possible signing verification key pairs, to tree height 25, where you've got around 32

million keys. You can then nest these sets in a number of multitree formats that allow for an essentially infinite number of keys. But that naturally can lead to significant complications. State management, as already noted, can also quickly become *very* complicated.

These schemes are not recommended for general-purpose use, but they can work really well in those instances where you sign something once and then verify it many times. This applies to root certificates and even intermediate certificates since these are things you create once while also signing a bunch of certs, and then those certs are used over and over again. It also applies to code signing, where you sign once and then many vehicles just need to verify the signature, which actually makes the job a lot easier.

GNN OK, so it's going to be PKI that handles these trees when it comes to handing the keys out. What's the thinking in terms of how that stash of keys is to be maintained in the car?

AT That's a very good question, but it doesn't actually apply to the car. Instead, it applies where the software updates are actually signed, which is in a hardware security module [HSM] located in the auto manufacturer's data center. The corresponding part of the system you'll find on the vehicle side is the public key that serves as the root node of the binary tree, which is only 60 octets [with each octet consisting of eight bits] in length for stateful, hash-based signature schemes—which is to say it's very small.

The vehicle is actually responsible only for doing the easy part here. It just needs to verify the signature, and that's relatively easy since the scheme relies on hashing,

which is something all the current automotive hardware nodes are capable of.

With that said, it still amounts to somewhat more hashing than the automakers are accustomed to. But then that's really all there is to it. The whole signature verification process involves doing only a couple hundred hashes. This is why that scheme is so suitable for deployment today. In fact, it could be used for software updates right now since even the computer hardware currently found in vehicles is capable of supporting it. Meanwhile, the real difficulties that come along with the new private key can be relegated to a data center where a couple of HSMS can be used to handle all the signing and backup requirements.

GNN Taking all this into account, what would you say are the quantum-related issues people should be most concerned about right now? And why?

AT There's certainly no need to worry about everything at the same time. It really gets back to that matter of product life spans and design cycles. For example, the financial services industry has its own quantum concerns to address, but there's no need for those folks to drop everything and start rethinking credit-card security just yet. That's because credit-card transactions are short-lived, and the cards themselves are replaced every few years. Even in the worst case, a new credit card could be issued within just a few days. In any event, any credit card you have in your wallet right now is likely to have been replaced a number of times before universal quantum-compute capabilities are made available to potential adversaries.

Auto manufacturers, on the other hand, need to account for much longer product life cycles. By the same token, they don't have to contend with the truly daunting product life-span concerns faced by the aerospace industry, where jet engines often are in service for several decades and, of course, cannot be readily replaced. That's a field already well into a timeframe where they need to be deeply concerned about looming quantum security threats—meaning they'll soon need to have answers for all aspects of that problem.

Auto manufacturers can just turn their focus initially to ensuring that whatever engine hardware is designed and built today is capable of handling the cryptography that will become essential once attackers are able to take advantage of quantum-compute capabilities. Once the auto industry has a handle on that, it can turn its attention to making sure it also has the ability to deliver software and firmware updates in a quantum-safe manner.

GNN Can you think of any industries or organizations that already seem to be approaching this correctly?

MG From an industry perspective, you have ETSI [European Telecommunications Standards Institute], which has already started to figure out how its standards are going to evolve with the addition of both quantum-resistant cryptography and quantum-key distribution in parallel with the work that's being done at NIST. There also are efforts going on in the CA/Browser Forum on standardizing post-quantum certificates, with DigiCert being a particularly loud voice in that space.

At a more organizational level, there are some great examples coming to light now at Microsoft, Google,

A hybrid of quantum and classical schemes might prove necessary in order to meet compliance targets.

and Cloudflare. They've all been doing some very public experiments on integrating quantum-resistant cryptography into TLS [Transport Security Layer], SSH [Secure Shell], and VPN [virtual private network] connections. A lot of the code they have built so far can be found in open-source repositories, so others can take advantage of it. I'd say these organizations also have a great focus on the practicalities of the upcoming transition and what needs to be done to ensure that real-world systems will be ready for whatever is ultimately standardized. One of the promising takeaways from these experiments is that they've shown overall performance in these schemes is still dominated by network transmission, meaning there's probably no need for concern that user experience is going to suffer unduly.

Among those who are currently tackling this challenge, the common thread seems to be a focus on cryptographic agility rather than on attempting to anticipate which of the proposed schemes is going to end up being certified. This suggests that, at an organizational level at least, there's an understanding that the efforts to modify these standards can continue to move forward in parallel so long as they're all built to be agile. Organizations also seem to be taking a pragmatic approach with their own efforts by assuming that a hybrid of quantum and classical schemes might prove necessary in order to meet compliance targets.

This would be my advice for the automotive industry as well: Keep cryptographic agility as a primary focus and don't overoptimize for any specific implementation. This should help with the quantum transition while also allowing for adaptability to changing regional

requirements. For companies with limited resources that need to focus their efforts, I'd say integrity and identity issues are the ones to concentrate on. Roots of trust tend to be the most difficult to swap out since people believe they can be trusted over a long period of time, and yet, in the end, all issues of integrity in a distributed system rely on shared roots of trust.

Copyright © 2021 held by owner/author. Publication rights licensed to ACM.

SHAPE THE FUTURE OF COMPUTING!

We're more than computational theorists, database managers, UX mavens, coders and developers. We're on a mission to solve tomorrow. ACM gives us the resources, the access and the tools to invent the future.

Join ACM today at acm.org/join

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

Association for
Computing Machinery

Advancing Computing as a Science & Profession