The transition to post-quantum cryptography, metaphorically

Stefan-Lukas Gazdag[1](✉) and Sophia Grundner-Culemann[2](✉)

[1] genua GmbH, Kirchheim near Munich, Germany
stefan-lukas_gazdag@genua.de
[2] Ludwig-Maximilians-Universität, Munich, Germany
grundner-culemann@nm.ifi.lmu.de

Abstract. *Are we there yet? Are we there yet? No, kids, the road to quantum-safety is long and sturdy. But let me tell you a story:*
Once upon a time, science discovered a great threat to Cryptography World: The scalable quantum computer! Nobody had ever seen one, but everyone understood it would break the mechanisms used to secure Internet communication since times of yore (or the late 20th century, anyway). The greatest minds from all corners of the land were gathered to invent, implement, and test newer, stronger tools. They worked day and night, but alas, when smaller quantum computers already started to emerge, no end to their research was in sight. How could that be?
This paper provides a collection of carefully wrought, more or less creative and more or less consistent metaphors to explain to audiences at all expertise levels the manifold challenges researchers and practitioners face in the ongoing quest for post-quantum migration.

Keywords: Post-quantum cryptography · Transition · Migration

1 Introduction

The potential rise of quantum computers may be one of the most disruptive developments in technology in the somewhat near future. The workings of such machines as well as their feasibility are understood better every day [36], for better and for worse. A major consequence is a cryptographic apocalypse, breaking today's asymmetric encryption and signature algorithms and at least weakening symmetric crypto. By now the need to transition to post-quantum cryptography (PQC) (= alternative algorithms that are not only secure in the classical context but against a quantum adversary) is glaringly obvious and widely accepted as inevitable. Thousands of people are working in this field right now, ranging from academia and researchers, standardization gremiums, and regulating authorities to programmers and engineers. Even salespeople and marketing have hopped on board.

Despite years of hard work and first success stories, the transition advances rather slowly. At the same time, governments and IT-security agencies base their recommendations for cryptographic mechanisms on the working hypothesis that

Fig. 1: The well captured current state of the transition to post-quantum cryptography. On the left one can see where we started, on the right one can clearly see where we are right now. Picture of Sisyphus by Titian via Wikimedia *Punishment_ sisyph.jpg* in the public domain.

quantum computers may be a serious threat by the early to mid 2030s or at least that the first major phase of transition has to be finished by then [20,21,28,36].

So far, the undertaking reminds us of the Greek myth of Sisyphus: This tragic king cheated death twice and got punished by Hades. He now has to move a stone up a mountain for eternity, and each time, just before he arrives at the top, the stone rolls all the way back down for the hardship to begin yet again (see Figure 1[3]). But the task will surely be completed by the end of eternity.

Luckily, in the case of the PQC transition, each of the single undertakings – research projects, PQC experiments, commercial feature development – is an important step forward in the whole picture. Citing the late Hans Rosling [30], *slow change is still change*. The reasons for this slow pace are quite diverse and lie in various problems and challenges.

For example, professional programmers are not necessarily educated cryptographers, while expert cryptographers do not necessarily know how to write good and secure code (with some striking exceptions). Yet, both sides have to cooperate on this complex topic, in which it is not even clear where to start.

In this rather non-technical paper we try to explain aspects of cryptography and of the PQC transition using metaphors and analogies, to help make the topic more accessible.

In Section 2 we introduce basic concepts of cryptography. In Section 3 we explain secure communication. Section 4 discusses the workings and peculiar-

[3] `https://knowyourmeme.com/memes/how-it-started-vs-how-its-going`

ities of secure communication in that ecosystem called the Internet. Section 5 illustrates the post-quantum tools at our disposal. We then briefly talk about the uncertainty of crypto security in Section 6. This leads us to crypto-agility in Section 7. We complete this metaphoric excursion with political remarks in Section 8.

The style of this paper is neither too serious nor very scientific (as you might have noticed) and rather intended to bring a smile to the reader's lips. Due to abstraction not every single detail may be true to the expert's eye. We try to give a veritable account as much as possible, but may omit or over-simplify some aspects hopefully gaining a better comprehensibility in return. That way we hope to give an accessible introduction into this highly interesting research field, that may even be used by the experts to explain the topic to others who may lack a technical background.

2 Basic cryptographic concepts or Let's feast!

Let's learn about basic cryptographic concepts with a culinaric approach. Get some snacks ready for this excursion into the field of trophology and let's feast.

We want to order some food from our favorite pizza place by robot-delivery (see Figure 2). But we don't want anyone to mess with our precious pizza on the way from the pizza place to our home. It's pizza after all! So how do we achieve this?

2.1 Padlocks and Encryption

Symmetric cryptography When we call the pizza place, they tell us a secret four-digit code. The pizza robot arrives and we see that it is locked with a digital combination lock. We open it using the code, et voilà: Pizza! Since we used the same combination as the pizza place, this is an example of symmetric cryptography. Seems to be easy, but we need to exchange the code in a way we consider secure.

Asymmetric cryptography Now imagine living in a time before phones (but with robot delivery, for some reason) and having a broken foot. This means, you cannot go to the pizza place in person, and you can not get a secret code for the delivery robot remotely. Your friend Joey will go and place the order for you, but you're hesitant to trust Joey with the secret code - he loves pizza a bit too much! So you hand him a padlock for which only you have a key you keep secret. Joey places your order and hands over your high-quality padlock. The pizza baker can easily use it to lock the robot's trunk, but it will be hard for even skilled lock pickers to open without the key. This is an example of asymmetric cryptography where users have both, a *public key* (the lock) that anyone may know and use to encrypt messages, and a *private key* (the key) that is hard to find and allows only the owner to decrypt the message.

Fig. 2: Could this be the truly secure way of pizza delivery? Let the trunk be locked with a padlock of our own that only we have a private key for. In addition the trunk could be sealed by the pizza place with an unforgeable seal. Detail of a picture by Oregon State University, CC BY-SA 2.0, https://flickr.com/photos/oregonstateuniversity/50507269361/

2.2 Key Encapsulation Mechanisms

The pizza place owner, Paula, hears about your broken foot and that you sent Joey over with a padlock to ensure a safe delivery. She anticipates more orders by you over the coming weeks, always relying on Joey and the padlock. She has an idea: She writes a note with a four-digit code and locks it in the delivery robot along with the pizza. Next time Joey is kind enough to place an order for you, he will not have to bring your padlock - from now on, both you and the pizza place can use the shared code to lock and unlock the robot.

Paula has used a *Key Encapsulation Mechanism* [11]: This technique has become popular in PQC research. They allow two parties to establish a shared key in the following way: One party sends the other a public key for which she knows a secret key. The other chooses a shared secret to encapsulate, encrypts it with the public key, and sends this cipher back. When the cipher is decrypted with the secret key, both parties know the shared secret.

To avoid bad behaviour of the party that chooses the key (like Paula choosing the code '1234', which Joey can easily crack), there are mechanisms that enforce good shared secrets (compare [17]). However, a proper, "Diffie-Hellman-style" key exchange mechanism as described in Section 2.4 is missing from the PQC landscape (since the most promising solution turned out to be insecure [8] and the NIST PQC standardization process focused on KEMs[4]).

[4] See old Q14 in the NIST PQC FAQ https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs and the mailing list https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/oc3Xl0ZKLGI/m/m_23qpdtBgAJ

2.3 Signed and Sealed

To assure you that the pizza delivery robot is coming from the right place, Paula can apply a seal to the robot with her stamp on it: As long as no one can forge her stamp, only she will be able to seal the robot to your satisfaction, and you can easily check the validity.

Digital signatures work in the same way: One party can sign a message with her secret key, and everyone else can easily check the signature with the corresponding public key [7].

2.4 Shared secrets: Pizza á la Diffie-Hellman

We know how to securely exchange information or things with people we know or people we can provide with our public key. But what if we have to dynamically set up a transfer with someone we just met? How do we agree on a shared secret? Again we need something, that is easy to do for the owner of a key, but hard to detect and that can be used to establish a shared secret. Imagine two pizza bakers who are said to make the best pizza in the world. Each of them has a secret ingredient that makes their pizza fantastic, but none of them wants their own ingredient ever to be unveiled. In this world, it is impossible to taste ingredients from pizza (dough). This is the *security assumption*.

With the following process, they can make the world's very best pizza combining their secrets without revealing it to the other (or anyone)

1. They agree on a pizza dough and each one prepares dough of the same kind.
2. Each pizza baker includes the secret ingredient.
3. The pizza bakers exchange the dough batches. This can happen over any channel - even if the dough is intercepted, no one can find out the secret ingredient. Then, each adds their own secret ingredient to the received batch.

Both pizza bakers now have the same kind of enhanced pizza dough without disclosing their individual secret ingredients.

This is the mechanism of a Diffie-Hellman style key exchange [12]: Two parties with individual secret keys exchange them in such a way that an outsider listening in can not learn the secrets, but both parties end up with the same shared key. Also, unlike KEMs, a Diffie-Hellman exchange gives both parties equal control over the result.

2.5 Using asymmetric vs. symmetric cryptography

Asymmetric cryptography is usually rather slow but key management is easy - public keys can be broadcast and anyone can use them. There is no need for a secure channel to exchange keys.

Symmetric cryptography is usually fast, but communication parties have to securely share the key beforehand.

In practice, both methods are combined: Communication partners agree on or share a secret key for symmetric encryption by using an asymmetric scheme.

When traditional cryptography can no longer be used, this will become harder, because established routines can not handle PQC keys. It is as if the padlock needs to become bigger and bigger over time. At some point, the robot will no longer be able to carry it. Section 4 explains this problem in greater detail after Section 3 lays out the basics.

3 Securing Internet communication or: Building walls

Providing security in the Internet often means building walls. A common example is the classical *perimeter paradigm*. Computers or whole computer networks are treated like medieval fortresses surrounded by big walls to keep attackers out.

And just like fortresses rely on moats and other additional fortifications, digital high-security relies on several layers of counter-attack mechanisms. This is called the *in-depth security paradigm.*

3.1 Tunnels between fortresses or: Virtual Private Networks

Communication between two fortresses is difficult - sending messages out over the open fields risks them being intercepted or manipulated. In the digital landscape, it is possible to build figurative tunnels between them: Any observer can see the tunnel or bridge - during or after construction -, its beginning and end, and the bricks it is made of. But it is hard to interfere with the construction process and the bricks are very sturdy. The digital analogy are Virtual Private Network (VPNs) which protect Internet traffic.

3.2 Different bricks and adding layers or: Post-quantum VPNs

The problem is: A quantum computer could tell an attacker how to destroy specific bricks or how to weaken some of them. Right now the crypto community is trying to find alternative bricks, but so far they have various drawbacks: Some are too big, others have odd shapes that are not suitable for tunnel-building or are made from material with strange behavior. Often it is unclear how long the material will last being exposed to the ever-lasting rain of cryptanalysis, the art of finding attacks.

Because those novel bricks or the ways they are integrated into the walls are not yet trusted to last on their own, the tunnels are often fortified using a *hybrid approach*, i.e. laying another layer of bricks on top of an old one. However, applying this outer coating can be troublesome as the construct may become too big or heavy to be actually practicable.

4 Limits of the ecosystem or: Running a train.

A cryptographer's life is full of conceptual work: Some invent crypto schemes, scrutinize their security, find and fix errors. Others implement such schemes

including hardening and testing the actual software code. Others adapt protocols to be compatible with this new cryptography.

But even if this important work is done meticulously, it may fail to take restrictions imposed by the real-world and the present ecosystem into account. This is not uncommon: Even networking experts with years of experience who know many pitfalls sometimes stumble over strange behavior of the Internet. The latter basically acts as a railway system for messages with billions of trains traveling the whole world every day.

4.1 The Internet Protocol or: Railway Infrastructures

If we want to send data via the Internet, the railway network is already there. There are tracks of fixed width. Depending on these, the train may reach a specific top speed. There are railway control centers that take care of routing trains to their destination. The locomotive knows where it has to go. It signals all the necessary information to the control center. It's the head of possibly many railway cars to follow, each car carrying a certain amount of cargo.

This can be compared to communication via the Internet Protocol [1]: Every message has a header (the locomotive) containing information about the message's (train's) destination and length. This information is processed by the routers that forward the message through the network. A message can have several parts, each carrying a certain amount of data. But, invisible to the inexperienced eye, the Internet exhibits a few historically grown quirks, explained in the next section.

4.2 Maximal Transport Unit (MTU) or: Train in Transit

In our metaphoric ecosystem, the length of a single train car is capped, corresponding to the Maximum Transmission Unit (MTU), a limit that exists of different networking layers and protocols. Sometimes a car may be allowed to be quite long with a relaxed MTU. But in other cases, especially when many long trains with long cars want to pass, the network may reduce the maximum allowed length dynamically and drastically.

The control centers (routers and other devices on the way) may also operate in a way that deviates from the network regulations (as specified by standardization gremiums): For example, upon learning that your car length used to be too long and you split it into smaller cars (= *IP fragmentation*), some systems (the *middleware*) may simply push your cars off the tracks without informing you[5]. Other times, systems wait for a specific number of cars to arrive and only forward the cars once there are enough, as if to ferry them across a river (*buffering*).

[5] There are just so many quirks about MTU, IP fragmentation and so on. For example, have a look here `https://lwn.net/Articles/960913/`

4.3 Connection Timeout or: No exit here!

Finally, the train may be close to its final destination, the train station aka computer or server is in sight. But train stations have restrictions of their own:

Maybe the station master only allows a specific number of cars behind your locomotive. Or your locomotive and the cars have to pull into the station within a given time. If you try to get too many cars into the station or you take too long, the station master may simply deny further entry and either send you all the way back home or message your station of origin that your train is now to be considered lost[6].

With post-quantum cryptography these problems might increase one order of magnitude: Let a normal train with one car represent a classical key exchange fitting into one packet of up to 1.5 kB including all headers.

To use the PQC encryption mechanism McEliece, a highly trusted approach, with the most secure parameter set and a 1.5 MB public key, the final train station would need to reserve space for a train with 1,000 times the usual length. This would mean missing out on hosting 1,000 classical trains instead.

4.4 Never change a running (train) system

Replacing classical cargo with PQC cargo is a difficult endeavour: Only few PQC approaches are interoperable with the current trains and network; others require more or much longer cars, or they significantly slow the trains down.

However, renewing the whole railway network with better tracks would be extremely costly and may be unfeasible[7].

Thus, we will likely have to continue finding workarounds to deal with the limits imposed by the tracks. This is arguably the biggest hindrance in the smooth introduction of post-quantum solutions or *fancier* crypto mechanisms [31, 35].

This is just a limited insight into the dirty details about the workings of the Internet as of today. Even many experienced computer scientists are not aware of the full extent of the problems and even skilled practitioners and implementors sometimes quarrel with it. Yet it is extremely important to understand that especially tentative implementors at companies, standardization organizations and so on are not simply nay-sayers who refuse new technolgoy. Usually, when very experienced practitioners become edgy, it's these practical issues that make them nervous and cautious.

4.5 Excursion: VPNs on literal trains

Let us support this metaphorical journey with a real-world example: Maybe you have travelled by train and luckily had access to the Internet via WLAN provided by the railroad company. You were able to connect to the network

[6] Compare, for example, `https://tldr.fail`

[7] There are proposals for new network architectures, though, e.g. SCION `https://scion-architecture.net/`

and the infotainment websites provided by the train. You might even have had smooth access to some websites via your browser. But for some reason, the VPN to your company network did not work at all. How come?

Well, just as described above, the MTU has to be taken into account: A single packet can only have a maximum data size that is allowed on the route to its destination. Each system on the path to the destination may have a different limit. So the lowest limit counts. The limit is usually considered to be around 1.5 kB. Quite often it's a bit lower than that, more like 1.2 to 1.4 kB. For the WLAN connection, this value is fine: You can access the websites provided by the server on the train.

But once your communication leaves the train via mobile protocols, the MTU may decrease drastically to 800-900 B or even less (especially when the train is moving fast). For some cases, this suffices; in other cases, the application or protocol has some mechanisms to detect and adapt to the MTU. But some applications have a hard time to suitably reduce package size.

In the latter case, your computer or smartphone may report a perfectly good WLAN connection, but some applications fail because only some or none of the packets can leave the train network. Then all of a sudden, the connection may be perfectly fine, e. g. because the train is slowing down, until there's hiccup soon after, and the mobile connection turns bad again.

5 New cryptographic tools or: Cleaning the stables

In the classical Greek myth about Heracles, the divine hero has to clean the Augean stables. He is faced with the poisonous feces of 3,000 cattle - a seemingly impossible task. But Heracles achieves the impossible by digging ditches, filling them with the unwelcome residues and redirecting the rivers Alpheios and Pineios to wash them out[8].

Today's Internet - an ever-growing laboratory experiment that got slightly out of hand - is a similar mess in need of care to make it secure or safe to cross for messengers and messages. To deal with this task, we usually use pitchforks, spades and shovels - our classical cryptography. Those are tools we are familiar with. We have years of experience.

Pitchforks are easy to handle, but it is strenuous work. Analysis and tests tell us that the pitchforks' quality and strength will be good enough at least for a while, though we know from experience that it is necessary to increase the size of the pitchforks every once in a while: attackers want to hinder us doing the chores by messing with our tools which have to be built sturdier.

Sooner or later, pitchforks will probably become too big to handle properly, but increasing the size is currently our best approach.

Scalable quantum computers are looming on the horizon, though - and they will break pitchforks of any reasonable size.

[8] `https://en.wikipedia.org/wiki/Heracles`

So curvy... **McEliece?**

Fig. 3: Digging into the dirt? You're used to spades and shovels? Well, they are all broken, because attackers know how to break the shafts easily. But we've got some alternatives. Now choose your tool wisely! *A selection of Georgian-Victorian English sterling silver tableware spoons (c. 1790 – c. 1850)* by Grenadille - Own work, CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=74660583` and By © Raimond Spekking / CC BY-SA 4.0 (via Wikimedia Commons), CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=26908766` Both via Wikimedia Commons.

PQC offers a variety of other tools to help with the mess: There are spoons and big mesh colanders - and for extreme cases, gigantic bucket-wheel excavators (see Fig. 3). They are all suitable for the task in principle, but with various advantages and disadvantages:

Most experts agree that the giant excavator, the McEliece crypto system [23] (the name of the famous mathematician, not a crypto mining company) can be trusted to do the job even in the long-term future. But moving this excavator to the relevant locations can be difficult due to its size.

The big mesh colanders are quite efficient and can serve as drop-in replacements for pitchforks to some extent. But choosing the right structure of the mesh (doesn't it remind you of a lattice?) is critical for its effectiveness and security [29].

Spoons with a certain curvature (namely that of an elliptic curve as used for isogeny-based cryptography) are also suitable for the task. They are easily transported, but it might take a lot of time to do the actual work. And already, a promising spoon design was shown to be faulty in the material [8]. This has decreased trust - though for other tools it is also unclear when material fatigue will set in. Evidently, choosing the right tool is a daunting task - but residue is piling up fast, so choose we must.

Fig. 4: Ok, we're almost there. Now where's this paper that tells us or gives us an idea on how to break crypto? Is it in conference proceedings of a different mathematical domain? Or even in engineering? Luckily more and more old books and proceedings are digitalized to make them more accessible and searchable, but what if we use the wrong search terms? Some experts we've talked to believe such a scenario to be rather unlikely compared to other advances in cryptanalysis, but it is not negligible according to some. By Matthew Petroff, CC BY-SA 3.0, https://upload.wikimedia.org/wikipedia/commons/c/c1/George-peabody-library.jpg

6 Trust in cryptography or: The infinite, chaotic library

Allegedly, famous cryptographer Adi Shamir said something like *If you need to keep secrets for 10 years, use RSA. If it's 40 years, don't use public key at all!* at the RSA Conference 2024, according to various sources and personal accounts (e. g. [24]). This paraphrases a problem in cryptography: We just have no idea what crypto is secure or for how long!

Trust in a cryptographic problem builds over time when many brilliant minds have studied it and found no practical attacks. However, the chance of a sudden insight is never zero. There's always the possibility that someone might come along with an unexpected idea to completely break our crypto.

Looking for attacks can be like checking an infinite, chaotic library for the existence of a certain book (Figure 4): The more shelves you go through without finding it, the stronger your hunch that it is not there - but anyone might find it suddenly and prove your conjecture wrong. Some regulating authorities therefore refrain from predictions spanning more than very few years (for example [2]).

For RSA, a lot of shelves have been searched to no avail; trust is high that it will be safe for the foreseeable future (unless, of course, large enough quantum computers emerge). In PQC, however, most approaches are still relatively new.

6.1 Best estimates

In practice, there is more to this: A good indicator of the estimated security are the run times or complexity of the best known attack algorithms for classical and quantum computers.

Also, formal proofs of security are important: They only supply a model of a rather idealized world; absolute proofs of security do not exist. They are also hard to handle, but help to avoid common mistakes and give a better understanding of the scheme.

Besides theoretic shortcomings, practitioners face implementation pitfalls. Years of experience using a scheme increase trust, that the overall security and proper use are well-understood (e.g., best choices for parameter sets).

As an example, there is no full formal proof for the RSA scheme, but many (partial) proofs and proofs for security properties of specific instantiations of RSA exist. Currently, the best classical attack against the concept are widely thought to be the *general number field sieve* and the best quantum algorithm seems to be Shor's algorithm [32]. A 829 bit key (RSA-250) was broken in 2020 [37] which gives us a hint on what computational power is able to do today. There are also known attacks against RSA implementations, e.g. the Coppersmith attack [10] or ROCA [25]. Despite those uncertainties, it is considered one of the best cryptographic tools we have today.

6.2 Thinking outside the box

It might seem implausible that new attacks should be found so suddenly rather than being developed gradually. An important aspect for this sort of occurrences are domain thinking and pure chance: Sometimes, new ideas emerge because someone unexpectedly crosses a bridge between two seemingly unrelated fields of expertise - as Craig Gentry, a lawyer turned computer scientist, showcased with his groundbreaking PhD thesis [16].

Even in closely related research fields, useful knowledge for a certain problem may remain hidden within another subdomain until someone stumbles accross it - compare, for example, the recent attacks found against the SIDH crypto scheme, which rely on a known idea but were overlooked for a comparatively long time [8].

6.3 The role of intelligence agencies

Of course, new mechanisms or attacks may be found by someone with little interest in making them public - like intelligence agencies. However, the common worldwide knowledge is usually considered to advance at a similar pace, meaning: Even if a single person or a specialised group should gain an intellectual

advantage with new developments, others will follow with similar ideas relatively quickly. For example, the British GCHQ invented public key schemes about four years before RSA, the first globally known asymmetric encryption scheme, was proposed [27]. Yet, such agencies may have other capacities like influence (compare backdoors in crypto) or *store now, decrypt later* attacks where encrypted Internet data is stored until the encryption may be broken by means of advances in computational power, cryptanalysis or quantum computers emerging.

7 Finding trade-offs or: Approaching the coffin corner

There are some hurdles, but also ideas and a lot of research already, so how hard can this PQC migration thing be? Answer: Yes.

As pointed out in Section 5, picking the right tools is not straightforward: What cryptographic scheme(s) to use? What parameter set to take? Does it fit into the communication protocol? Can this be implemented well and does it work in real-world use? Section 6 highlights another layer: What if we choose a solution that turns out to be insecure? As explained in Section 4, the ecosystem has limits and it is hard to change a running railway system once it is established.

We therefore need to think long and hard about the flexibility of our future systems (their *crypto-agility*) and the trade-offs involved in designing them. We will probably not want our first PQC migration steps to stick forever, though there is a viable risk they will. So we better have good migration steps, each a worthwhile and important move in the right direction.

The following sections explain crypto-agility per se, the "coffin corner" problem in aviation, and why this problem arises in PQC migration.

7.1 Crypto-agility

Crypto-agility treats a broad variety of questions like:

- Does a system have an update mechanism? If so, is it possible to switch on the go when it's supposed to run 24/7?
- Can ever-increasing key sizes [5] of the same approach be fitted into concepts and implementations designed around smaller values without having to remodel the whole thing?
- Can the migration of a scheme to a successional version - like MD4 to MD5 to SHA-1 to SHA-2, and last but not least, SHA-3 - ever be smooth?
- (How) Can a crypto apocalypse caused by an implementation bug easily be mitigated by switching to a built-in alternative? Or will only long nights of fixing and distributing updates solve the problem - just to find some admins and end-users not applying the provided patches[9]?
- Can doing more than five key exchanges out of a choice of ten crypto schemes in arbitrary order ever be practical?

[9] Compare the Heartbleed bug: https://www.securityweek.com/heartbleed-still-affects-200000-devices-shodan/

In the PQC world, crypto-agility considers not only exchanging mechanisms, but also the use of *hybrid solutions* (employing several cryptographic schemes at once), ideally with a choice of the preferred algorithms with which to do so.

Of course, crypto-agility may introduce a lot of complexity: Previous design principles used to follow the KISS principle ("Keep it simple, stupid!") wherever possible to deal with performance, memory requirements and attack vectors. Also, minimalist approaches in software design reduce the complexity of testing significantly. More agile designs aim to account for increased flexibility, but can hardly be tested with the same scrutiny and are more likely subject to the ecosystem's restrictions.

This raises the question which trade-offs can be achieved in practice. The next section illustrates why some design problems have very little wiggle room.

7.2 The coffin corner

Consider an analogy from aerodynamics or aeronautics where problems can arise at high altitudes for some fixed-wing aircraft:

With higher altitude, the air gets thinner and colder. The minimum speed required for the aircraft to keep flying (*stall speed*) increases in thin air as the aerodynamic lift must stay strong enough. At the same time, these machines are not resistant to the forces and shock waves from air flow over the wings exceeding the speed of sound - and the speed of sound decreases in lower temperatures.

So, with rising altitudes, the aircraft must fly faster to stay in the air, but if it rises too high and flies too fast, it breaks. This speed range is called *the coffin corner*, as flying only a bit too fast or too slow at the maximum flight altitude would end catastrophically.

In the case of the PQC migration, the coffin corner is this sweet spot where we achieve the optimal trade-off between practicability and flexibility (see Figure 5): We need a solution that is as agile as possible while keeping the complexity at bay on account of the rigid requirements of our ecosystem (which might change dynamically - the fluctuating MTUs from Section 4 send their regards).

The more flexible our solutions become, the more difficult they are to employ. If what we propose is well-suited for exchanging faulty crypto schemes, it is likely so demanding that it may be unusable in everyday communication. If, however, our protocols are very similar to the traditional ones (i.e., designed around the crypto that fits in nicely but hardly flexible) it will fold as soon as the cryptanalysis winds blow a little too hard and break the crypto we rely on – and the migration struggle starts again.

7.3 Aircraft and tunnels

How about crypto-agility in VPNs? Does it feel like sending an aircraft through a tunnel[10]? Consider the Diffie-Hellman Key Exchange used in IPsec [22]. The left

[10] Reality is often worse than one's own imagination. Please don't try this at home: `https://www.youtube.com/watch?v=19fQAxys9q8`

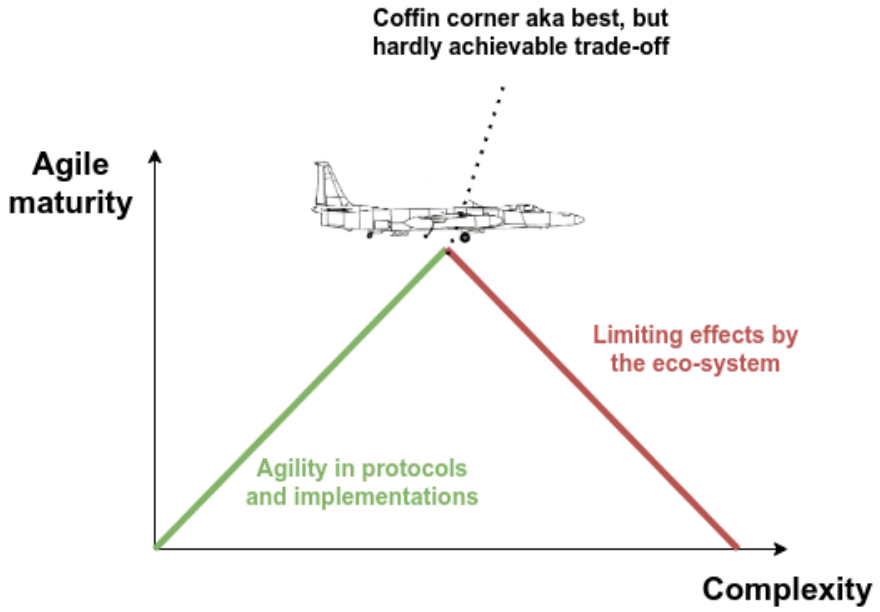

Fig. 5: More agile solutions usually come with a higher complexity. We wish to fly high with agility, but need a minimum amount of complexity to do so. But the environment only allows for a maximum complexity before solutions become impractical in real-world use. Thus a coffin corner exists that is the point of a specific kind of agility with the maximum complexity that is still practical enough. U-2 plane used as a part of work by Marcin Zieliński - Own work, CC BY 2.5, `https://commons.wikimedia.org/w/index.php?curid=1404361`.

part of Figure 6 shows the neat traditional version with two rounds of messages between the initiator and the responder.

If the comparatively small Diffie-Hellman keys usually used in this protocol are replaced or combined with a small and efficient quantum-safe scheme (say, sNTRU Prime [6]) the protocol does not change a lot: This usually fits within the MTU and other constraints and adds only a little complexity. However, this approach is not flexible: Larger schemes can not be used, and if sNTRU Prime is broken - well, tough luck!

We can extend the protocol to allow the use of more and bigger schemes: The right part of Figure 6 shows the PQC version of IKEv2 with up to seven rounds of messages. While this can be done and is a standardized approach [33, 34], it adds a lot of overhead and complexity compared to the original protocol design. Thus its practicability may be limited in some cases, e.g. in a congested network with high packet loss rates [15, 18, 19]. The quest for the sweet spot between crypto-agility and practicability therefore remains an open research topic.

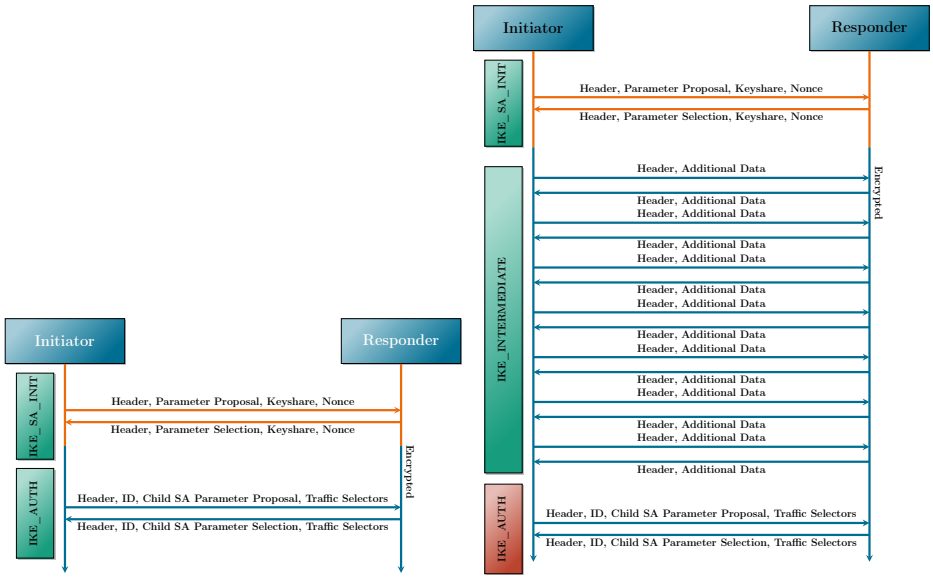

Fig. 6: The left diagram shows a classical IKEv2 for key establishment and authentication [13]. Even a hybrid combination of a small ECDH and a small scheme like sNTRU Prime 761 [6] or Kyber 768 [4] may fit in, though a small MTU might be a problem. The changes that are necessary to provide a quite agile post-quantum solution adds a lot of complexity to be seen on the right [33,34]. Please note that you do not need to do all the steps shown to provide a post-quantum key establishment, but you could. Also, post-quantum authentication has not yet been adressed explicitly. Original tikz by Daniel Herzinger, PQ-IKEv2 adapted

8 A balancing act

Will PQC migration get a bad reputation? Will it feel like the development of fusion reactors (or quantum computers) - always about ten years away?

Fifteen years ago it was hard to get funding for PQC because the public didn't see the problem yet. Now it feels like getting funding may become harder because the PQC community hasn't presented the *answer to the ultimate question of life, the universe and everything* [3]. Recent funding calls focus on the transition to quantum-safe crypto (compare [9,14]). This is very important in order to achieve a rather quick migration. We see more and more success stories and they actually are successes. Quite often we see impressive progress. But in the big picture, people may wonder why we're not done yet. And will the support continue after the first steps are taken? Will the migration be considered fully done once the first iteration of quantum-safe solutions is out there?

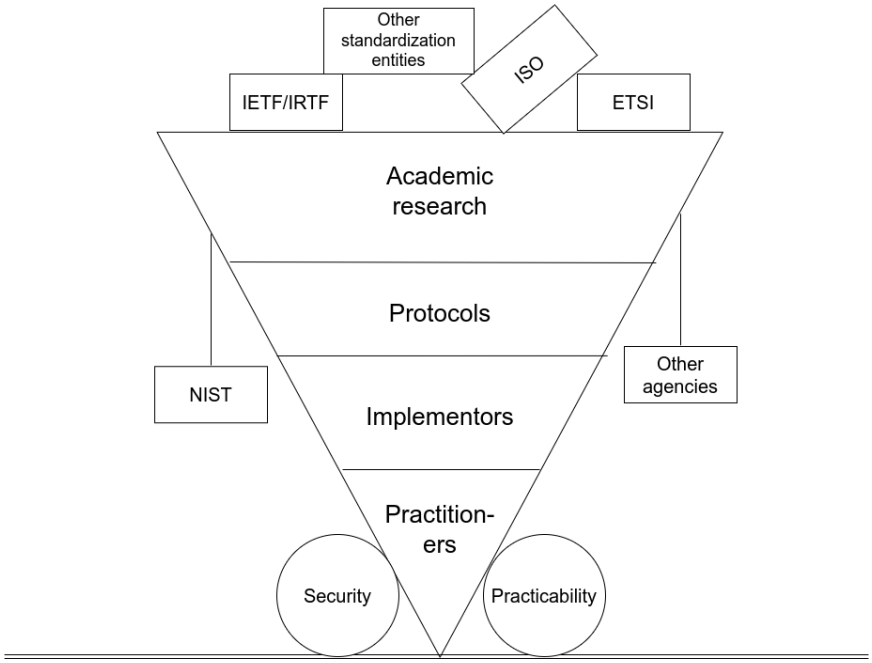

Fig. 7: The progress of the PQC migration is a balancing act. It is important to understand that so many different players and experts are needed alike. There's not one entity or aspect more important than all others. We have to find useful answers for the questions on all levels. This can only be done by continuous collective effort of all affected parties.

For players and stakeholders it may sometimes be easy to blame others. *The implementors don't want our fantastic new academic results! – Academic researchers don't want to understand the pain we practitioners experience every day! – Everything could be done easily if only the regulating authorities would tell us what to do!* But what everyone at least subliminally knows is that we are talking about a huge challenge and mission that we can only accomplish in a collective effort (see Figure 7). The PQC migration is a modern role model of applied cryptography and a key driver of crypto-agility, going all the way from highly theoretic research to talking about bits and bytes on the line.

Now that the publication of the first NIST PQC standards is finally there [26], it feels like starting running from the starting block, but still not being too sure where the finish line is.

What we can expect in the near future are broader first steps into the migration (yes, some have been done already). But the solutions introduced now will likely not be the ones we need in ten to fifteen years. PQC migration means continued development and improvement. Just as we had to learn that regular software updates are a necessity (though not everyone seems to have heard it),

we need to understand that technology updates are an important part of the Internet's future, as hard and laborsome as change is.

Being open-minded to new approaches while understanding and giving respect to the problems of practical networking will help us profit from many advantages of a more agile and resilient infrastructure that will enhance the overall security. We might learn a lot from the transition and shape our future communication, not only regarding the threat of quantum computers but rethinking how we do things on the Internet in general.

Dear users, sponsors and funding givers, dear everyone: Please bear with us!

Acknowledgements

We thank metaphors, analogies and irony for their existence. But without any irony we thank all the motivated and skilled people in the post-quantum community, who help making this transition not only possible but worthwhile.

When working on and with metaphors it is sometimes hard to distinguish *original work* and *prior art* that may have found its way into the subconscious. If you think we didn't give proper credit for some ideas despite our best efforts and let us know, we will try to update this humble publication.

We want to encourage discussion, new ideas and the use of these collected metaphors: Raising awareness and explaining the benefits and stumbling blocks of the transition will help smoothing our journey to a secure post-quantum world.

We will also need you when we found a start-up for mounted messenger delivery of code books, transfer of One-Time Pad codes, and truckloads of PQ-RSA [5] keys on hard drive - of course aiming for environmentally friendly solutions. Going analog might be the future of secure communication ;-)

All provided links have been checked for availability on 2024-09-09.

References

1. Internet Protocol. RFC 791 (Sep 1981). https://doi.org/10.17487/RFC0791, https://www.rfc-editor.org/info/rfc791
2. Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Technical Guideline BSI TR-02102-1, Federal Office for Information Security, Bonn, DE (February 2024)
3. Adams, D.: The hitchhiker's guide to the galaxy (1995)
4. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber. Submission to the NIST Post-Quantum Cryptography Standardization (2017)
5. Bernstein, D.J., Heninger, N., Lou, P., Valenta, L.: Post-quantum rsa. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography. pp. 311–329. Springer International Publishing, Cham (2017)
6. Bernstein, D.J., Lange, T., Chuengsatiansu, C., van Vredendaal, C.: NTRU Prime. NIST submissions (2017)
7. Buchmann, J.A.: Digital Signatures, pp. 249–275. Springer New York, New York, NY (2004). https://doi.org/10.1007/978-1-4419-9003-7_12, https://doi.org/10.1007/978-1-4419-9003-7_12

8. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Paper 2022/975 (2022), https://eprint.iacr.org/2022/975
9. Commission, E.: Post-quantum cryptography transition HORIZON-CL3-2024-CS-01-02 (Jun 2024), https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/horizon-cl3-2024-cs-01-02
10. Coppersmith, D.: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. Journal of Cryptology **10**(4), 233–260 (Sep 1997)
11. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33**(1), 167–226 (2003). https://doi.org/10.1137/S0097539702403773
12. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (November 1976)
13. Eronen, P., Nir, Y., Hoffman, P.E., Kaufman, C.: Internet Key Exchange Protocol Version 2 (IKEv2). RFC 5996 (Sep 2010). https://doi.org/10.17487/RFC5996, https://www.rfc-editor.org/info/rfc5996
14. für Bildung und Forschung, B.: Bekanntmachung Post-Quanten-Kryptografie in die Anwendungen bringen (Dec 2022), https://www.bmbf.de/bmbf/shareddocs/bekanntmachungen/de/2022/12/2022-12-30-Bekanntmachung-PostQuantenKryptografie.html
15. Gazdag, S.L., Grundner-Culemann, S., Heider, T., Herzinger, D., Schärtl, F., Cho, J.Y., Guggemos, T., Loebenberger, D.: Quantum-resistant macsec and ipsec for virtual private networks. In: Günther, F., Hesse, J. (eds.) Security Standardisation Research. pp. 1–21. Springer Nature Switzerland, Cham (2023)
16. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. p. 169–178. STOC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440, https://doi.org/10.1145/1536414.1536440
17. Giacon, F., Heuer, F., Poettering, B.: Kem combiners. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography – PKC 2018. pp. 190–218. Springer International Publishing, Cham (2018)
18. Heider, T.: Towards a Verifiably Secure Quantum-Resistant Key Exchange in IKEv2. Master's thesis, Institute of Informatics of the Ludwig–Maximilians–University Munich (2019)
19. Herzinger, D., Gazdag, S.L., Loebenberger, D.: Real-World Quantum-Resistant IPsec. In: 2021 14th International Conference on Security of Information and Networks (SIN). vol. 1, pp. 1–8 (2021). https://doi.org/10.1109/SIN54109.2021.9699255
20. Jr., J.R.B.: Executive Order on Improving the Nation's Cybersecurity. Tech. rep., The White House (May 2021), https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
21. Jr., J.R.B.: National Security Memorandum on Promoting United States Leadership in Quantum Computing While Mitigating Risks to Vulnerable Cryptographic Systems (NSM-10). Tech. rep., The White House (May 2022), https://www.whitehouse.gov/briefing-room/statements-releases/2022/05/04/national-security-memorandum-on-promoting-united-states-leadership-in-quantum-computing-while-mitigating-risks-to-vulnerable-cryptographic-systems/

22. Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., Kivinen, T.: Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296 (Oct 2014). https://doi.org/10.17487/RFC7296, https://datatracker.ietf.org/doc/html/rfc7296.txt
23. McEliece, R.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Coding Thv **4244**, 114–116 (1978)
24. Millman, R.: Five things we learned from the 2024 RSA Conference (May 2024), https://www.itpro.com/security/5-takeaways-from-rsa-conference-2024, Last access: 2024-09-09
25. Nemec, M., Sys, M., Svenda, P., Klinec, D., Matyas, V.: The return of coppersmith's attack: Practical factorization of widely used rsa moduli. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 1631–1648. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3133969, https://doi.org/10.1145/3133956.3133969
26. NIST: NIST Releases First 3 Finalized Post-Quantum Encryption Standards (Aug 2024), https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards
27. NSA: Clifford Cocks, James Ellis, and Malcolm Williamson (2021), https://www.nsa.gov/History/Cryptologic-History/Historical-Figures/Historical-Figures-View/Article/3006218/clifford-cocks-james-ellis-and-malcolm-williamson/, last Access: 2024-09-09
28. NSA: Announcing the Commercial National Security Algorithm Suite 2.0. Tech. rep., National Security Agency (Sep 2022), https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF
29. Player, R.: Parameter selection in lattice-based cryptography. Ph.D. thesis, Royal Holloway, University of London (2018)
30. Rosling, H., Rönnlund, A.R., Rosling, O.: Factfulness. Sceptre (2018)
31. Schmidt, R.: Post-quantum Security At Signal (Jun 2024), https://newtpqc.org/public/rolfe_schmidt.pdf
32. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011
33. Smyslov, V.: Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2). RFC 9242, Internet Engineering Task Force (May 2022). https://doi.org/10.17487/RFC9242, https://www.rfc-editor.org/info/rfc9242
34. Tjhai, C., Tomlinson, M., Bartlett, G., Fluhrer, S., Geest, D.V., Garcia-Morchon, O., Smyslov, V.: Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2). RFC 9370 (May 2023). https://doi.org/10.17487/RFC9370, https://www.rfc-editor.org/info/rfc9370
35. Westerbaan, B.: post-quantum fancy cryptography (Jun 2024), https://newtpqc.org/public/bas_westerbaan.pdf
36. Wilhelm, F., Steinwandt, R., Zeuch, D., Frey, J.: Status of quantum computer development. Tech. rep., Federal office for Information Security (August 2023), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Quantencomputer/Entwicklungstand_QC_V_2_0.pdf
37. Zimmerman, P.: Factorization of RSA-250 (Feb 2020), https://web.archive.org/web/20200228234716/https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2020-February/001166.html, last Access: 2024-09-09