

Hiding Secrets in the Dimensions

Understanding Post-Quantum Cryptography Through Geometry - and Why Businesses Need to Prepare Now

A plain-English guide to post-quantum cryptography, lattice-based encryption, and the practical steps organizations can take to prepare for long-term cryptographic change.

Author

Jeff Campbell

Published

July 26, 2026

Publication

Research Publication No. 1

Category

Post-Quantum Cryptography

Contents

1	Executive Summary	1
2	Why This Matters Now	1
3	The Invisible Trust Layer	2
4	How Key Exchange Really Works	2
5	Why Quantum Changes the Rules	3
6	Harvest Now, Decrypt Later	3
7	Post-Quantum Cryptography in Plain English	4
8	The Three NIST Standards in Simple Terms	4
9	A Dot-Grid Exercise	5
10	Why the Fourth Dimension Feels Strange	5
11	Dimensions as Attributes	6
12	From Dimensions to Lattices	6
13	The Role of Noise	7
14	What This Means for Business Leaders	7
15	Questions CTOs Should Be Asking	8
16	A Practical Readiness Roadmap	8
17	Conclusion	9
18	Author Note	10
19	About the Author	10
20	References	10

Executive Summary

Encryption is the invisible trust layer under modern business. It protects banking, healthcare, remote work, software updates, identity systems, private communications, cloud services, backups, APIs, and vendor integrations.

For decades, much of public-key security has relied on mathematical problems that ordinary computers find extremely difficult to reverse. RSA, Diffie-Hellman, and elliptic-curve cryptography are not protected because their handshakes are hidden. They are protected because the public information exposed during those handshakes should not be enough to recover the private secret.

Quantum computing changes that assumption for specific families of cryptography. Shor's algorithm is not merely faster brute force. It is a different mathematical shortcut that can expose hidden structure in RSA, Diffie-Hellman, and elliptic-curve systems once sufficiently powerful fault-tolerant quantum computers exist.

This does not mean the internet ends tomorrow. It means organizations need to prepare before the emergency. The most urgent issue is not panic-driven replacement. It is visibility: knowing where cryptography lives, which data must remain confidential for years or decades, which vendors control critical trust paths, and which systems are flexible enough to migrate.

Post-quantum cryptography is the response. It runs on ordinary computers but is designed to resist both classical and quantum attacks. NIST finalized its first three post-quantum cryptography standards in 2024: ML-KEM for key establishment, ML-DSA for digital signatures, and SLH-DSA for digital signatures.

The practical conclusion is simple: most organizations do not yet know where all of their cryptography lives. The first step is to find it.

Why This Matters Now

The current technology age is not moving at the speed of the old industrial world. Artificial intelligence, cloud infrastructure, automation, robotics, biotech, and advanced computing are compressing years of progress into months, weeks, and sometimes days.

That pace is exciting. It is also dangerous when the foundations underneath our systems are changing faster than our planning cycles.

Encryption is one of those foundations. It is built into web traffic, VPNs, certificates, software signing, backup systems, APIs, identity providers, databases, SaaS platforms, and third-party services. Most people do not notice encryption when it works. They only notice it when it fails.

The post-quantum discussion is no longer only theoretical. Standards now exist. Vendors are beginning to implement them. Government guidance already recommends readiness roadmaps, cryptographic inventories, vendor engagement, and migration planning.

The question for many organizations is not whether post-quantum cryptography exists. It does. The harder question is whether the organization knows enough about its own environment to migrate when it needs to.

The Invisible Trust Layer

Most people think about encryption only when they see the lock icon in a browser. That is the visible part. The hidden part is much larger.

Encryption helps a browser trust a bank website. It helps a VPN protect a remote worker. It helps a software update prove it came from the real vendor. It helps a cloud service protect stored data. It helps an identity provider sign tokens. It helps a device decide whether firmware is legitimate.

In a modern business, cryptography is not one system. It is everywhere.

It lives in certificates, TLS endpoints, VPN tunnels, IPsec, SSH, SFTP, database connections, backups, EHR systems, payment processors, SaaS platforms, mobile apps, endpoint agents, code-signing pipelines, APIs, service accounts, cloud KMS platforms, HSMs, and old appliances nobody wants to touch.

That is why post-quantum readiness is not just a cryptography problem. It is an infrastructure problem. It is a vendor problem. It is an application problem. It is a compliance problem. It is a procurement problem. It is a data-classification problem. It is a long-term risk-management problem.

How Key Exchange Really Works

A common misunderstanding is that encryption works because the key exchange is hidden. It does not. Modern encryption assumes an attacker may be watching the entire conversation.

The secret is not protected because the attacker cannot see the handshake. The secret is protected because the handshake only exposes public mathematical material. The actual session key is never sent across the network. Each side uses its own private information and the other side's public information to independently arrive at the same shared secret.

This is like an incomplete recipe.

The public key is like a public pantry. Everyone can see the eggs, flour, bread, syrup, and other basic ingredients. The private key is the hidden cooking method: temperature, timing, sequence, technique, and adjustments that are never spoken aloud.

During the exchange, the systems do not send the finished secret key across the network. They send public mathematical material that is safe for others to see. Each side then combines that public material with its own private information to independently arrive at the same final flavor.

An attacker may see the public pantry and the visible dish moving across the table, but without the hidden cooking method, the attacker cannot reconstruct the final secret flavor. That final flavor is the shared secret used to derive the actual encryption keys.

In practical systems such as TLS or VPNs, the bulk data encryption is often still done with familiar symmetric encryption such as AES or ChaCha20. The post-quantum change usually happens earlier in the trust process: how systems establish the shared secret and how they prove identity.

Why Quantum Changes the Rules

A normal computer uses bits. A bit is either a 0 or a 1. Off or on. Stack enough bits together, and computers can represent documents, pictures, programs, databases, videos, and everything else we use them for.

A quantum computer uses qubits. A qubit is not just on or off in the same simple way. Through quantum behavior, qubits allow certain calculations to be approached in ways ordinary computers cannot efficiently reproduce.

That does not mean quantum computers are magic. It does not mean they make every hard problem easy. It does not mean every password, bank account, and encrypted message becomes public overnight.

But for some mathematical problems, quantum computers change the battlefield.

Shor's algorithm is the reason RSA, Diffie-Hellman, and elliptic-curve cryptography are at risk. It gives a sufficiently powerful quantum computer a shortcut through the mathematical maze those systems rely on.

Using the recipe analogy, classical cryptography is like showing the world a finished dish while keeping the two secret master ingredients hidden. A normal cook may know the dish was made from some combination of ingredients, but recovering the exact secret ingredients would take an unreasonable amount of trial and error.

Shor's algorithm is not just a faster cook guessing more recipes. It is more like a quantum-trained head chef who can detect hidden structure in the dish and recover the secret ingredients directly. For RSA, the hidden structure is related to factoring a huge number back into the secret prime numbers used to create it. For Diffie-Hellman and elliptic-curve systems, the hidden structure is related to a different problem called the discrete logarithm problem.

That distinction matters. The threat is not simply that quantum computers are faster. The threat is that quantum computers can use a different kind of algorithm against the exact mathematical assumptions many current public-key systems depend on.

Harvest Now, Decrypt Later

The most important business risk is not only what a quantum computer can break today. It is what attackers can collect today and decrypt later.

An attacker does not need to decrypt everything immediately. They can capture encrypted traffic, documents, medical records, legal files, financial information, source code, identity data, or government communications today and store it.

If the data remains valuable five, ten, twenty, or thirty years from now, the attacker can wait for better tools.

That changes the timeline. The risk begins before the quantum computer is ready.

Some data loses value quickly. A lunch order does not need to remain secret for decades. A short-term meeting invite may not matter next year. But medical records, legal files, financial

records, intellectual property, source code, identity data, credentials, private communications, and government-related information may remain sensitive for a long time.

Long-lived data should move to the front of the planning line.

Post-Quantum Cryptography in Plain English

Post-quantum cryptography is not encryption that runs on a quantum computer. It is cryptography designed to run on ordinary computers while resisting attacks from both ordinary and quantum computers.

The most important distinction is this: post-quantum cryptography usually does not mean every byte of application data is encrypted with a brand-new quantum-resistant bulk cipher. In many deployments, the actual data still uses symmetric encryption such as AES or ChaCha20. The post-quantum change happens in the key-establishment and signature layers.

The Three NIST Standards in Simple Terms

Standard	Plain-English role	Common places it may matter
ML-KEM	Helps two systems establish a shared secret without sending the secret across the network.	TLS, VPNs, IPsec, API connections, cloud service connections
ML-DSA	Creates digital signatures that prove authenticity and integrity.	Certificates, private PKI, code signing, firmware signing, identity systems
SLH-DSA	Creates hash-based digital signatures, often useful where conservative long-term signature assumptions matter.	Specialized signing, long-term trust, fallback or high-assurance signing use cases

ML-KEM is the easiest to connect to encrypted traffic because it helps create the shared secret that later feeds symmetric encryption. ML-DSA and SLH-DSA are not used to encrypt the payload. They are used to prove identity, authenticity, and integrity.

In other words: post-quantum readiness is not just about keeping messages private. It is also about keeping trust chains trustworthy.

A Dot-Grid Exercise

The word “lattice” is where many people start to check out. A mathematician might explain it with formulas, vectors, matrices, modular arithmetic, probability distributions, and high-dimensional geometry. That explanation may be correct, but for many people, it dies on the first page.

So instead, let's use geometry.

Start with a flat sheet of paper. Draw a simple grid of dots: four dots across and four dots down. That gives you sixteen dots. Each dot has an address. The dot in the second column and first row might be described as:

(2, 1)

That is a two-dimensional address. It tells you where the dot is on the page: across and down.

Now add a third dimension. Make three more copies of that same sheet of paper and stack them on top of the first one. Now you have four layers. Each layer has sixteen dots, so the whole stack has sixty-four dots. A dot can now be described by three coordinates:

(2, 1, 4)

That might mean second column, first row, fourth layer.

You have created a simple three-dimensional grid. You can think of it as a cube made of addressable points. Every point has a location. Every location can be named.

So far, this is easy to picture.

Why the Fourth Dimension Feels Strange

Now add another dimension.

We can write the next coordinate as:

(2, 1, 4, ?)

But where does that fourth axis go?

Left and right are already used. Up and down are already used. Forward and backward are already used. There is no obvious direction left to point in. This is where the human brain starts to struggle.

Three-dimensional space is limited to the first three address identifiers. When you add the fourth, it no longer fits inside the visual constraints we are used to. The mistake is trying to cram the fourth dimension into the same physical box.

Instead, change the visualization.

Imagine each dot in the three-dimensional grid as a container. Everything inside that container already shares the first three coordinates. If we are standing at coordinate (2, 1, 4), everything inside that container has the same 2, 1, and 4 address values.

Now the fourth coordinate can describe an attribute inside the container.

For this teaching example, imagine the fourth dimension is color. The container holds billions of possible colors, all unique. We select red. The address now becomes:

(2, 1, 4, red)

That red object can itself be treated as another container. Everything inside it shares the same first four coordinates: 2, 1, 4, and red.

Now the fifth dimension might be shape. Imagine billions of possible shapes and fractals. We select octagon. The address becomes:

(2, 1, 4, red, octagon)

That octagon can become another container. Inside it, everything shares the same first five coordinates. The sixth dimension might be texture. The seventh might be temperature. The eighth might be density. The ninth might be a frequency. The tenth might be a pattern. The specific attributes do not matter. The point is that each new dimension adds another way to narrow the address.

This can repeat until the address has hundreds or thousands of coordinates.

A thousand-dimensional address does not need to be drawn as a thousand physical directions. It can be understood as a thousand-part description. Each part narrows the location, relationship, or state of the object being described.

In real mathematics, the coordinates are numbers and relationships, not literal colors and octagons. But the visualization helps: after the third dimension, stop looking for another physical direction and start thinking in terms of nested attributes, containers, and increasingly specific addresses.

Dimensions as Attributes

This sounds abstract until you remember that computers already describe things this way.

A person in a database might be described by age, height, zip code, income range, job title, device type, login behavior, purchase history, browser fingerprint, risk score, location pattern, and thousands of other attributes.

You cannot draw all of that as a normal physical shape, but mathematically it can still be treated as a point in a high-dimensional space.

The same idea appears in machine learning, recommendation systems, fraud detection, image recognition, and data analytics. High-dimensional spaces are not science fiction to computers. They are ordinary working territory.

That is the bridge to lattice cryptography.

From Dimensions to Lattices

Imagine a vast grid, not in two dimensions, not in three dimensions, but in hundreds or thousands of dimensions. Every point in that grid has an address. But the address is not simple like row two, column three. It may be a long list of numbers describing a location across many axes.

A lattice is a structured collection of points in such a space. The structure is regular enough that legitimate users can perform useful operations, but complex enough that certain problems become extremely hard without secret information.

The secret is not simply hidden at one visible dot. That would be too simple. In lattice-based cryptography, the legitimate user has secret information that makes the high-dimensional structure manageable. An attacker sees public information, but without the secret, the attacker faces a hard geometry problem inside a space too large and complex to search directly.

This is why the dimensional analogy matters. It gives us a way to understand how an address can have hundreds or thousands of components without needing to draw hundreds or thousands of physical directions.

The Role of Noise

Lattice-based cryptography also adds another important idea: noise.

Noise means the answer is intentionally nudged away from the clean, obvious point.

Go back to the flat sheet of paper. If I tell you the secret is at coordinate (2, 1), you can put your finger directly on that dot. But what if I place the secret slightly between dots? Maybe it is near (2, 1), but not exactly on it. Maybe it has been moved a little to the right, a little up, and at a slight angle.

If you know the secret adjustment, you can still find it.

If you do not, you are only near the answer.

In two dimensions, that is not very secure. You could look around and probably find it. In three dimensions, it gets harder. In a thousand dimensions, near becomes a nightmare.

Using the container analogy, imagine being told that the secret is in container (2, 1, 4, red, octagon), but the actual value has been nudged through additional hidden dimensions. You are not just looking for the right dot, the right color, or the right shape. You are trying to recover a precise relationship across a huge number of attributes, with deliberate noise added to blur the target.

With the right secret, the noise is manageable. Without the secret, the public information is not enough to reconstruct the answer efficiently.

That is the basic intuition behind lattice-based post-quantum cryptography.

What This Means for Business Leaders

The business takeaway is not that every leader needs to become a cryptographer. The takeaway is that the foundation under many current trust systems is changing.

Post-quantum cryptography changes the hard problem. It moves away from relying on RSA, Diffie-Hellman, and elliptic-curve assumptions that Shor's algorithm threatens. It moves toward

mathematical problems, including high-dimensional lattice problems, that are believed to resist both classical and quantum attacks.

But having standards is not the same as being migrated.

Old certificates, VPNs, applications, hardware devices, blockchains, APIs, identity providers, databases, development frameworks, and third-party platforms all need to be inventoried and upgraded carefully. Some systems will support hybrid post-quantum options quickly. Others will lag. Some old devices may never support them at all.

This is the part many CTOs are not thinking about yet, especially accidental CTOs by default: the leaders who inherited technical responsibility without becoming cryptographers, network architects, or PKI specialists.

They do not need a math lecture. They need a practical way to ask the right questions.

Questions CTOs Should Be Asking

The first business step is not buying a product. It is understanding exposure.

A post-quantum readiness conversation should begin with questions like these:

- Where are we using public-key cryptography?
- Where are we using RSA, Diffie-Hellman, or elliptic-curve cryptography?
- Where do we rely on certificates, certificate authorities, or digital signatures?
- Where are keys generated, stored, rotated, and retired?
- Which systems use hardcoded cryptographic libraries?
- Which vendors manage encryption or key management on our behalf?
- Which VPNs, firewalls, API gateways, SaaS platforms, and identity providers sit in critical trust paths?
- Which devices or applications cannot be upgraded easily?
- Which data would still matter if stolen today and decrypted ten years from now?
- Do our procurement and vendor review processes ask about post-quantum readiness and crypto-agility?

For many organizations, the answers will be incomplete.

That is not a reason to ignore the problem. It is the reason to start.

A Practical Readiness Roadmap

The right response is not panic. The right response is preparation.

A practical readiness roadmap might look like this:

1. Create a cryptographic inventory. Identify where encryption, certificates, key exchange, signing, VPNs, and cryptographic libraries are used.

2. Classify data by secrecy lifetime. Prioritize data that must remain confidential for years or decades.
3. Identify systems exposed to harvest-now-decrypt-later risk. Pay special attention to externally transmitted sensitive data, long-term archives, backups, and high-value intellectual property.
4. Ask vendors for their post-quantum roadmaps. Track the answers and make them part of procurement, renewal, and risk-management discussions.
5. Evaluate crypto-agility. Determine which systems can change algorithms easily and which ones are brittle, legacy-bound, or vendor-locked.
6. Test hybrid post-quantum options where available. Look for performance, compatibility, certificate-size, network, and operational impacts.
7. Update procurement language. New systems should not be purchased with cryptographic dead ends built into them.
8. Document exceptions. Some systems will not be upgradeable and will need compensating controls, isolation, replacement plans, or formal risk acceptance.
9. Build a staged migration plan. Start with the highest-value and longest-lived data, then expand outward.
10. Revisit the plan regularly. Standards, vendor support, implementation guidance, and threat timelines will continue to evolve.

This does not mean every business needs to become a cryptography research lab. It means leaders need to understand that encryption is infrastructure.

Conclusion

Quantum computing does not mean the internet ends.

It does not mean every encrypted system fails at once. It does not mean companies should rip out every VPN, certificate, firewall, application, and identity platform tomorrow.

But it does mean the assumptions under much of today's public-key cryptography are changing.

Modern encryption does not work because attackers cannot see the handshake. It works because the handshake is designed so that public information is not enough to recover the shared secret. Quantum computing threatens some of the old assumptions behind that design.

Post-quantum cryptography changes the hard problem. It does not simply make RSA bigger. It moves trust toward mathematical structures, such as high-dimensional lattices, where the secret path remains manageable for the legitimate user and brutally difficult for the attacker.

The future of encryption may be built inside high-dimensional geometry, where the correct answer is easy to recover if you have the secret path and extremely difficult to find if you do not.

The secret may be hidden in the dimensions.

But the business risk is hidden in the blind spots.

Organizations that prepare now will treat post-quantum migration as infrastructure modernization. Organizations that wait may discover that their encryption was not a permanent wall. It was a countdown.

The math may be complicated. The responsibility is not.

Find where encryption lives. Understand what must stay secret. Ask vendors hard questions. Build crypto-agility. Test early. Migrate deliberately.

The next generation of digital security will not be won by the organizations that panic first. It will be won by the ones that prepare before the emergency.

Author Note

This paper is intended as an educational overview for business leaders, technical leaders, and accidental CTOs who are beginning to think about post-quantum cryptography and long-term encryption risk.

For organizations that want to move from awareness to action, the next step is usually a structured readiness assessment: identifying where cryptography is used, which data must remain confidential over long timelines, which vendors control critical encryption paths, and which systems may need modernization before post-quantum migration becomes urgent.

The author is available for questions and limited consulting discussions by request.

About the Author

Jeff Campbell is an IT and cloud infrastructure consultant currently working with a high-quality Managed Service Provider supporting healthcare, enterprise, and regulated business environments. His background includes cloud architecture, database platforms, disaster recovery, security operations, infrastructure modernization, firewall governance, Microsoft 365, AWS, and enterprise systems integration.

Jeff previously spent seven years at AWS, including work with Amazon RDS, Aurora PostgreSQL, AWS Database Migration Service, and large-scale database modernization efforts. His current work focuses on helping organizations understand operational risk, modernize infrastructure, and prepare for technology shifts before they become emergencies.

For questions or follow-up discussion regarding this paper, Jeff can be reached at: research@solvefor42.com.

References

- NIST, “NIST Releases First 3 Finalized Post-Quantum Encryption Standards,” August 13, 2024.
- NIST FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard.
- NIST FIPS 204, Module-Lattice-Based Digital Signature Standard.
- NIST FIPS 205, Stateless Hash-Based Digital Signature Standard.
- CISA, NSA, and NIST guidance on quantum-readiness migration to post-quantum cryptography.