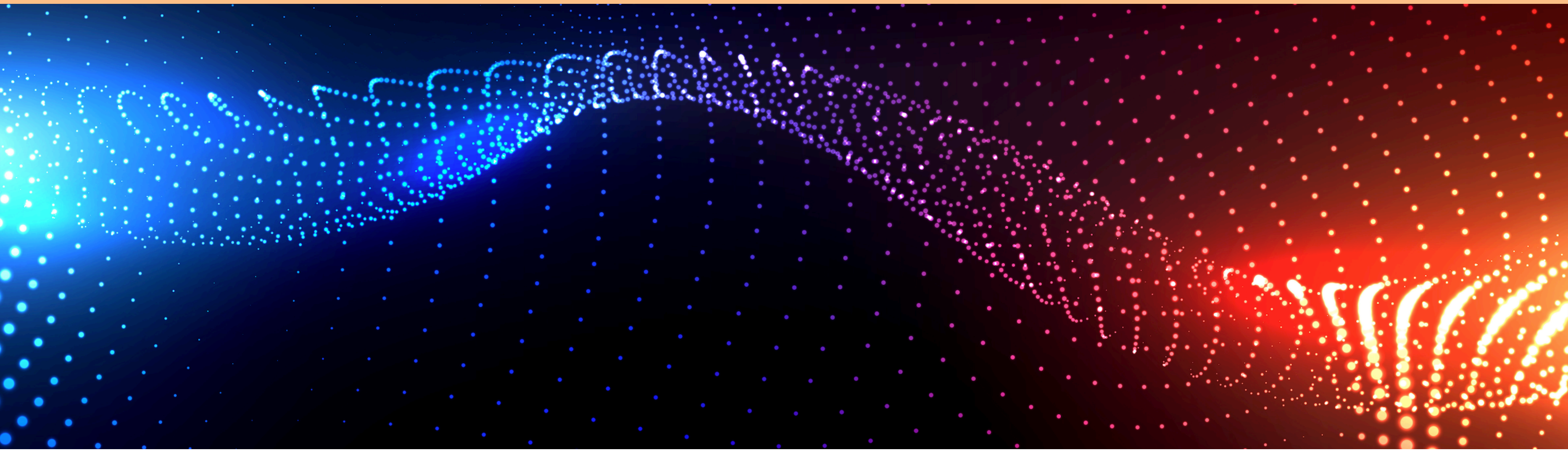
INTRINSIC ID™

White Paper



Intrinsic ID PUFs in a Post-Quantum World

Why should I (not) be concerned about using post-quantum cryptography?

Introduction

In July 2022, NIST announced the first batch of to-be-standardized algorithms for post-quantum cryptography.[1] This announcement has security professionals world-wide scrambling to assess the status of their systems and to evaluate the need for transitioning to these algorithms in the future. Security is critical to all systems, so this has led to some nervousness and uncertainty. Moreover, the discussions around the application of post-quantum cryptography are highly technical, which makes it challenging for security professionals to make a well-informed decision.

Unfortunately, this current situation of temporary uncertainty also creates a brief opening for opportunistic actors to push their "solutions" through tactics of fearmongering (warning against non-existent problems which they claim to solve) and deception (claiming to solve real problems while in fact they do not). This can lead to well-intentioned but misguided investments, and worst-case even to vulnerable systems. In this white paper we want to address two important questions with the goal of helping embedded security architects and engineers weather the current uncertain stage of this evolution by arming them with the right information:

1. **Why should I (not) be concerned about using post-quantum cryptography?**
2. **What is the status of Intrinsic ID hardware security products in a post-quantum world?**

[1] "PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates", NIST Information Technology Laboratory, July 5, 2022, https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4

Credits: Image by starline on Freepik

What precisely is the threat that post-quantum cryptography will protect us from?

The Need for Post-Quantum Cryptography

The Quantum Threat

Before we dive into post-quantum cryptography, we first need to clarify the quantum threat, i.e., what precisely is the threat that post-quantum cryptography will protect us from?[2,3,4] Understanding this is paramount in evaluating one's need for post-quantum cryptography.

The quantum threat is brought about by a combination of two important evolutions in the last couple of decades:

1. The development of quantum computing technology since the 1980s, with gradually improving operational quantum computers since the 2010s.
2. The discovery in the 1990s of quantum algorithms that affect the security of certain traditional cryptographic techniques.

Let's look a little more into the details of these developments.

A **quantum computer** is a computing system which uses quantum mechanical phenomena like superposition and entanglement to perform basic operations. Based on this, it can *in theory* perform certain very specific computations much more efficiently than any traditional computer could. However, the development of a *meaningful* quantum computer, i.e., one that can *in practice* outperform a modern traditional computer is exceptionally difficult. For the moment there is still ongoing debate as to whether this is just an incredibly complex engineering problem to be solved, or if there are more fundamental limitations at play which could prevent this altogether. Assuming the former case, enormous investments and significant research and development time will still be needed to bring quantum computing technology to a meaningful level. Taking the current state of the art and extrapolating its development into the future, assuming an exponential improvement equivalent to Moore's law for traditional computers, it is estimated by experts that it will still take at least 15 to 20 years before a meaningful quantum computer can become reality.[2,3]

Contrary to popular belief, a quantum computer is *not* a very fast general-purpose supercomputer, *nor* can it magically operate in a massively parallel manner. Instead, it presents a different paradigm to computing, because it is able to efficiently execute so-called **quantum algorithms** for which there are sometimes no known efficient equivalents on a traditional computer. Today, two known quantum algorithms are of concern to us, because they affect the security of classical cryptography:

1. **Shor's algorithm**, invented in 1994 by Peter Shor, is an efficient (polynomial-time) quantum algorithm for factoring large integers, and for solving a few related number-theoretical problems. There are currently no known efficient factoring algorithms for traditional computers, a problem which lies at the basis of several classic public-key cryptographic techniques.

[2] "Report on Post-Quantum Cryptography", NIST Information Technology Laboratory, NISTIR 8105, April 2016, https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf

[3] "2021 Quantum Threat Timeline Report", Global Risk Institute (GRI), M. Mosca and M. Piani, January, 2022, https://globalriskinstitute.org/mp-files/quantum-threat-timeline-report-2021-full-report.pdf

[4] "Quantum Safe Cryptography and the Quantum Threat", SSH Academy, https://www.ssh.com/academy/cryptography/what-is-quantum-safe-cryptography

2. **Grover's algorithm**, invented in 1996 by Lov Grover, is a quantum algorithm which can search for the inverse of a generic function quadratically faster than a traditional computer can. In cryptographic terms, searching for inverses is equivalent to a brute-force attack (e.g., on an unknown secret key value), and its difficulty lies at the basis of security for most symmetric cryptography primitives.

It is clear that these quantum algorithms, if they can be executed on a meaningful quantum computer, will have an impact on the security of current cryptographic techniques.

The impact of quantum computers on current-day public-key cryptography is problematic and needs to be fixed.

Impact on Public-Key Cryptography

By far the most important and most widely used public-key primitives today are based on RSA (for encryption and signing), and discrete-logarithm or elliptic curve cryptography (for key exchange and signing). All these primitives are based on number-theoretic problems which can be efficiently solved by Shor's algorithm. This has major implications. *At the point in time when meaningful quantum computers become operational,* Shor's algorithm can be used to break these primitives on a practical level. This will make virtually all public-key cryptography in current use insecure.

For the affected public-key encryption and key exchange primitives, this problem is already real today. An attacker can capture and store encrypted messages exchanged now and break their decryption in the future when meaningful quantum computers are operational. Highly sensitive and/or long-term secrets communicated today (or in the past) are hence already at risk to be disclosed in the future.

For the affected signing primitives, the problem is a little less problematic when used in short-term commitments. However, if meaningful quantum computers become available then valid signatures can be forged, which hence voids the value of any signature at that point in time. It is, therefore, not recommended to use the affected primitives for signing long-term commitments which still need to be verifiable in 15-20 years or more.

Needless to say, due to Shor's algorithm, **the impact of quantum computers on current-day public-key cryptography is problematic and needs to be fixed**. For some use cases there is already a real risk now due to the capture-and-store scenario, but the situation becomes globally impactful if meaningful quantum computers become available in an estimated 15-20 years from now. Over the last decade, the cryptographic community has worked hard on solving this problem by designing new public-key primitives that are based on mathematical problems which cannot be solved by Shor's algorithm (or any other known efficient algorithm, quantum or otherwise). These algorithms are generally referred to as **post-quantum cryptography**. NIST's announcement on a selection of these algorithms for standardization[1], after years of public scrutiny, is the latest culmination of that field-wide exercise.

Impact on Symmetric Cryptography

Nearly all applications of symmetric cryptography are based on the use of block ciphers (e.g., AES) and hash functions (e.g., SHA-256), and to a lesser extent stream ciphers. For a well-designed symmetric key primitive, its security level is equivalent to the effort needed for brute-forcing the used secret key, or exhaustively searching for collisions. On a traditional computer, the effort of brute-forcing a secret key is directly exponential in the key's length: $O(2^n)$. However, when a

The practical impact of quantum computers on symmetric cryptography is, for the moment, very limited.

meaningful quantum computer can be used, Grover's algorithm can speed up the brute-force attack quadratically. The needed effort remains exponential, though only in half of the key's length: $O(\sqrt{2^n} = 2^{n/2})$. In terms of the security level of a symmetric primitive expressed in number of bits, Grover's algorithm can be said to reduce it by 50%.

However, there are some important remarks to be made regarding the potential impact of Grover's algorithm on symmetric cryptography:

- It was shown that Grover's algorithm is an optimal brute-force strategy (quantum or otherwise),[5] so the quadratic speed-up is the worst-case security impact that needs to be considered. No future developments in quantum computing or quantum algorithms will lead to better brute-force attacks on symmetric crypto.
- There are strong indications that it is not possible to meaningfully parallelize the execution of Grover's algorithm.[3,6,7,8] In a traditional brute-force attack, doubling the number of computers used will cut the computation time in half. Such a scaling is not possible for Grover's algorithm on a quantum computer, which makes its use in a brute-force attack very impractical.
- Before Grover's algorithm can be used to perform practical brute-force attacks, e.g., on 128-bit keys, so with an effort of $O(2^{128/2} = 2^{64})$, the performance of quantum computers needs to improve tremendously. Very modern traditional (non-quantum) supercomputers can barely perform computations with a complexity of $O(2^{64})$ in a practically feasible time (several months). Based on their current state and rate of progress, it will still take a very long time – much, much more than 20 years – before quantum computers could be at that same level.[7]

Despite Grover's algorithm, **the practical impact of quantum computers on symmetric cryptography is, for the moment, very limited**. Worst-case, the security strength of currently used primitives is reduced by 50% (of their key length), but due to the mentioned limitations of Grover's algorithm, that is an overly pessimistic assumption for the near future. The natural conclusion that the solution is to double the length of symmetric keys to withstand quantum brute-force attacks is a very broad blanket measure that will certainly solve the problem but is too conservative. **At this point in time, there are no mandated recommendations for quantum-hardening symmetric-key cryptography, and 128-bit security strength primitives like AES-128 or SHA-256 are considered safe to use now and in the foreseeable future (also see, e.g., reference 8, for the standpoint of NIST about this)**.

[5] "Grover's quantum searching algorithm is optimal", C. Zalka, Phys. Rev. A 60, 2746, October 1, 1999, https://journals.aps.org/pra/abstract/10.1103/PhysRevA.60.2746

[6] "Reassessing Grover's Algorithm", S. Fluhrer, IACR ePrint 2017/811, https://eprint.iacr.org/2017/811.pdf

[7] "NIST's pleasant post-quantum surprise", Bas Westerbaan, CloudFlare, July 8, 2022, https://blog.cloudflare.com/nist-post-quantum-surprise/

[8] "Post-Quantum Cryptography - FAQs: To protect against the threat of quantum computers, should we double the key length for AES now? (added 11/18/18)", NIST Information Technology Laboratory, https://csrc.nist.gov/Projects/post-quantum-cryptography/faqs

Information-theoretically secure constructions are not impacted at all by the quantum threat.

Impact on Information-Theoretical Security

In addition to public-key and symmetric cryptography, we also want to take a short look at the impact of quantum computing on information-theoretic security. Information-theoretically secure methods are algorithmic techniques for which security claims are proven in a mathematical sense, without any further assumptions on complexity of functions or hardness of mathematical problems (hence it is also called *unconditional* or perfect security). Some important information-theoretically secure constructions and primitives are:

- The Vernam cipher, also called the one-time pad, an information-theoretically secure encryption method. While highly secure, the Vernam cipher is not very practical for most use cases.
- Shamir's secret sharing, a method for securely sharing a secret among multiple parties. It is used in scenarios where high-security secret sharing is needed.
- Quantum key distribution[9] (unrelated to, and not to be confused with post-quantum cryptography) is a secure method for establishing a shared secret between two parties based on quantum-mechanical effects in the communication between them, e.g., using entangled photons.
- Entropy sources and PUFs will generate outputs with guaranteed entropy levels, based on validated stochastic models underpinning their physical operation. They are generally used as secure sources of randomness and for generating root secrets.
- Fuzzy commitment schemes[10] will reliably generate high-entropy secrets from noisy sources (e.g., PUFs, or quantum channels), without leaking any information on that secret.

It is important to realize that all information-theoretical security claims for these constructions remain fully valid in the presence of quantum computers. An information-theoretical proof basically shows that an adversary does not have sufficient information to break the security claim, regardless of its computing power, quantum or otherwise. **Hence, information-theoretically secure constructions are not impacted at all by the quantum threat.**

Executive Summary

Why should I (not) be concerned about using post-quantum cryptography?

The quantum threat to information security is fueled by the development of quantum computing technology, and the discovery of quantum algorithms that affect the security of currently used cryptographic primitives. This threat will become imminent once meaningful quantum computers become available, which is at earliest expected about 15 to 20 years from now. However, quantum computing and quantum algorithms will not break all of cryptography, and the real impact of the quantum threat differs between different primitives:

[9] "Quantum cryptography: Public key distribution and coin tossing", C. H. Bennett and G. Brassard, Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, December, 1984, https://arxiv.org/abs/2003.06557

[10] "A fuzzy commitment scheme", A. Juels and M. Wattenberg, Proceedings of the 6th ACM conference on Computer and Communications Security, November, 1999, https://dl.acm.org/doi/pdf/10.1145/319709.319714

- Popular **public-key primitives** (e.g., RSA, Diffie-Hellman, elliptic-curve crypto) are severely impacted and need to be fixed. This has led to the development of post-quantum cryptography, and several alternative public-key crypto candidates exist today. The standardization process for these new candidates is ongoing.[1]
- **Symmetric cryptography** (e.g., AES, SHA-256) is in theory affected as well, but in practice the impact is very limited. For the moment, there is no need for quantum-hardening existing symmetric-key primitives [8]. If ever needed in the future, a doubling of the used key lengths is guaranteed to solve any known issues.
- **Information-theoretically secure primitives** are not impacted at all and will remain fully secure in the face of quantum computers.

Intrinsic ID PUFs and PUF-Based Key Generation

SRAM PUF

The core technology underpinning all Intrinsic ID products is that of an SRAM physically unclonable function or SRAM PUF. Like other PUFs, an SRAM PUF generates device-unique responses that stem from unpredictable variations originating in the production process of silicon chips. The operation of an SRAM PUF is based on a conventional SRAM circuit and is hence readily available in virtually all digital chips.

The behavior of an SRAM PUF is fundamentally physical in nature. However, as a component in a security solution, it is essential to have this physical behavior captured in a stochastic model that describes the important statistical metrics of the PUF and enables us to accurately quantify them. Based on years of continuous measurements and analysis, Intrinsic ID has developed stochastic models that describe the behavior of SRAM PUFs very accurately,[11] and that go much further than traditional, generic PUF evaluation metrics. Using these models, we can determine tight bounds on the unpredictability of SRAM PUFs, both in terms of uniqueness (how unpredictable is one SRAM PUF instance from any other), and in terms of noisiness (how unpredictable is one SRAM PUF evaluation from another). These unpredictability bounds are expressed in terms of *entropy*, and are fundamental in nature, i.e., they cannot be overcome by any amount of computation, quantum or otherwise.

QuiddiKey

Intrinsic ID offers a hardware security solution based on SRAM PUF technology in the form of its QuiddiKey product. Figure 1 shows, on a high level, the basic internal architecture of QuiddiKey.

The central component of QuiddiKey is a *fuzzy commitment scheme*[10] that protects a root key with an SRAM PUF response and produces public helper data. The operation of a fuzzy commitment scheme is such that the root key can always be reliably regenerated later based on the helper data and a noisy version of the PUF response. It is information-theoretically proven that the helper data discloses zero information on the root key, so the fact that the helper data is public has no impact on

[11] "An Accurate Probabilistic Reliability Model for Silicon PUFs", R. Maes, Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, 2013, https://www.iacr.org/archive/ches2013/80860176/80860176.pdf

the root key's security. This no-leakage proof relies on the unpredictability bounds of the PUF employed by the system, as expressed by its stochastic model. Over the years, Intrinsic ID has spent significant effort on optimizing the fuzzy commitment scheme to robustly deal with the stochastic behavior of SRAM PUFs in a very broad sense, always with a focus on keeping the no-leakage proof intact.[12]

Where the goal of a fuzzy commitment scheme is to be able to securely regenerate the same root secret, the goal of an *entropy source* is rather to generate guaranteed fresh entropy every time it is evaluated.[13] The actual entropy which is collected by an entropy source needs to originate from a physically noisy process. QuiddiKey implements an entropy source that relies on an SRAM PUF acting as a physical noise source. In this case, it is the noisiness of the SRAM PUF that plays an important role, and the PUF's stochastic model is seminal for assessing the amount of generated noise entropy. Very importantly, QuiddiKey uses its entropy source to initialize its root key for the very first time, which is subsequently protected by the fuzzy commitment scheme.

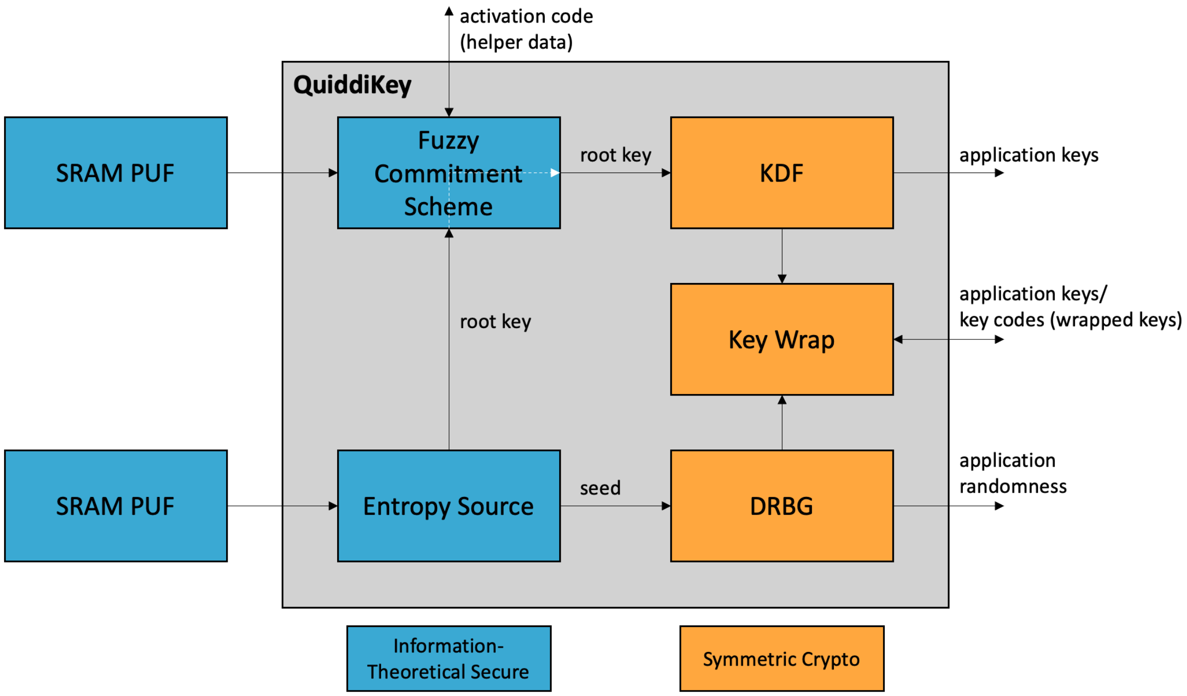


Figure 1 – High-level Architecture of QuiddiKey.

[12] "Secure Key Generation from Biased PUFs", R. Maes, V. van der Leest, E. van der Sluis and F. Willems, Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, 2015, https://www.iacr.org/archive/ches2015/92930497/92930497.pdf

[13] "NIST SP800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation", NIST Information Technology Laboratory, January 2018, https://csrc.nist.gov/publications/detail/sp/800-90b/final

In addition to the fuzzy commitment scheme and the entropy source, QuiddiKey also implements a number of cryptographic operations for conveniently serving an application. These include:

- A key derivation function (KDF) which uses the root key protected by the fuzzy commitment scheme as a key derivation key. The KDF is accessible to an application for the reproducible generation of device-unique, cryptographically secure application keys.
- A deterministic random bit generator (DRBG) which is initially seeded by a high-entropy seed coming from the entropy source. The DRBG is accessible to an application for the generation of cryptographically secure randomness.
- Key wrapping functionality, essentially a form of authenticated encryption, for the protection of externally provided application keys using a key-wrapping key derived from the root key protected by the fuzzy commitment scheme.

All these application-accessible cryptographic operations are based on certified standard-compliant constructions making use of standard symmetric crypto primitives, particularly AES and SHA-256.[14]

QuiddiKey does not deploy any public-key cryptography primitives for which a near-future update would be imminent.

Post-Quantum Assessment of QuiddiKey (and Apollo)

As schematically shown in Figure 1, all security components of QuiddiKey are either backed by information theory or based on symmetric cryptography. In that sense, the impact of the *quantum threat* described in the previous section on the security of QuiddiKey is very limited.

All the core components leading up to the generation and protection of the QuiddiKey root key are shown to be information-theoretically secure and are hence impervious to quantum attacks. Essential elements which back this information-theoretical security are: the accurate stochastic model of the SRAM PUF, which describes and quantifies its unpredictability in terms of entropy; the noise entropy assessment of the entropy source based on that model; and the no-leakage proof of the fuzzy commitment scheme, also based on that model.

All application-accessible functionality of QuiddiKey is based on established symmetric cryptography. As discussed in the previous section, in theory symmetric cryptography is mildly affected by quantum attacks due to Grover's algorithm, but in practice the impact is very limited, and no remediation is required nor recommended at this time. In addition, QuiddiKey can be delivered in several variants, including a 256-bit security strength variant, which reflects on the length of the root key. The use of this 256-bit variant of QuiddiKey will offer strong quantum-resistance, even in a far future when Grover's algorithm becomes feasible. However, it should be reiterated that for now and the foreseeable future, 128-bit symmetric cryptography remains safe to use,[8] which includes the 128-bit variant of QuiddiKey.

The Apollo product, the FPGA-based security solution of Intrinsic ID, is constructed following a very similar architecture and using identical building blocks. All the post-quantum assessments above for QuiddiKey equally hold for Apollo.

Importantly, QuiddiKey and Apollo do not deploy any public-key cryptography primitives for which a near-future update would be imminent.

[14] NIST Information Technology Laboratory, Cryptographic Algorithm Validation Program CAVP, validation #A2516, https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=35127

Executive Summary

What is the status of Intrinsic ID hardware security products in a post-quantum world?

The security architecture of QuiddiKey is completely based on information-theoretically secure components for the generation and protection of a root key, and on established symmetric cryptography for the application-accessible functionality. As discussed in the previous section, information-theoretically secure constructions are impervious to quantum attacks. The impact of the quantum threat on symmetric cryptography is very limited and does not require any remediation now or in the foreseeable future. **Importantly, QuiddiKey does not deploy any quantum-vulnerable public-key cryptographic primitives.**

Concluding, all variants of QuiddiKey are quantum-secure and in accordance with recommended post-quantum guidelines. **The use of the 256-bit security strength variant of QuiddiKey will offer strong quantum-resistance, even in a distant future, but also the 128-bit variant is considered perfectly safe to use now and in the foreseeable time to come.**