White Paper

The Quantum Computer and Public-Key Crypto Systems

A look at security of the present-day public-key primitive, as well as implications for the future

ENTRUST

SECURING A WORLD IN MOTION

Table of Contents

INTRODUCTION

"Cryptography" is not a single thing, but a collection of algorithms addressing different aspects of data security: confidentiality, integrity, and authenticity. Elements that encompass some of its main aspects include encryption, key exchange, and signatures.

The public-key cryptographic primitives in common use today rely on the difficulty of one of two mathematical problems for their security. The first one involves finding the factors of a number that is the product of two large primes. The second one involves finding discrete logarithms. That is, given an input element, an output element, and a mathematical operation or function, determine how many times the operation must be applied to the given input to produce the given output. The elements may be either the integers modulo a large prime or points on an elliptic curve modulo a large prime. RSA, Diffie-Hellman, Elliptic-Curve Diffie-Hellman, and the Elliptic-Curve Digital Signature Algorithm have all been shown to be secure as long as these problems are hard.

When mathematicians use the words "problem" and "hard," they mean something quite precise; a problem is like a mathematical puzzle where you are given some input information and have to compute some output. **A problem is described as hard if the amount of work required to compute the output increases exponentially with the size of the problem.** For example, searching a database of n entries is not hard because, if you add one or two new entries, the search only requires one or two more checks.

The famous Traveling Salesman problem, on the other hand, is hard because increasing the number of cities to be visited from n to n+1 or n+2 results in n or n2 new paths to check. In cryptographic systems, the size of the problem is the size of the key, which means adding one bit to the key size, in theory, doubles the amount of work to break the algorithm, and doubling the key size results in the square of the amount of work required.

In order to keep the strength of public-key algorithms out of reach in the face of advancing cryptanalytic techniques and increasing computing power, key sizes must be increased periodically. For most commercial applications, the well-accepted guidance for RSA key size is currently set at 2048 bits, and this is due to rise to 3072 bits by the year 2030. Developments in machine architecture are responsible for many of the cryptanalytic advances that have been made against RSA in recent years, exploiting the parallelism possible by networking computers and using graphics processing units.

Now, engineering advances are making practical a quite different machine architecture – one based on quantum mechanical effects. And, it turns out that solving factoring and discrete logarithms (and therefore breaking RSA, Diffie-Hellman, and Elliptic-Curve public-key algorithms) are not hard for such machines. Stated differently, all three of these algorithms are broken by a quantum computer of sufficient size.

The quantum computer

The idea of building a computing machine based on quantum mechanical principles dates back to the 1950s. It was quickly understood what the properties of this hypothetical machine would be, and the race was on to, first of all, develop the underlying technology required to build a working quantum computer, and secondly, figure out what problems it could solve.

By 1994, a quantum information algorithm called Shor's algorithm was discovered that would make it possible to find the factors of a composite number. Today, Shor's algorithm has been demonstrated in a working quantum computer, but only for very small composite numbers. Several promising technological approaches, including Shor's algorithm, have been demonstrated to produce working quantum computers, albeit at very small scale. But quantum computing technologies must scale up by many orders of magnitude to produce a machine suitable for attacking public-key cryptographic problems.

At the heart of a quantum computer is a register of quantum bits, or qubits, whose name is derived by analogy with the classical digital computer, which is based on the binary digit, or bit. In a digital computer, a bit is always in either one of two states, represented by the voltage at the output of a logic gate, and by measuring the voltage we can discover which state it is in.

In contrast, the properties of a quantum computer derive from the superposition of wave functions on a qubit. There are many ways of realizing a physical qubit, but one well-studied way is the spin of a subatomic particle. In this instance, the state of the qubit is represented by the orientation of the spin. External operations on the particle can modify its spin orientation, and the qubit's state, by superimposing additional wave functions.

When measured in a particular direction, the spin appears to be aligned either in that direction or in the opposite direction, depending upon the state at the time of measurement; **it is said to have either "spin-up" or "spin-down" with respect to the measurement direction.** The answer obtained depends probabilistically on the spin orientation with respect to the measurement direction. More generally, the measured state of a qubit is either a 0 or a 1, depending probabilistically upon its internal state at the time of measurement. Measuring a qubit inevitably modifies the state of the qubit. So, if the contents of a quantum register represents the result of a computation, then once read, the register no longer contains the result.

Superposition is the first important concept behind quantum computers. If we simply load a register of qubits with a single value and then read that value back, we have not gained anything. The value of quantum computers comes from the wave properties of quantum mechanical phenomena. These can be used to represent data as a wave, and load all the input data into a single quantum register at the same time by superimposing their wave functions. We can then perform operations on the register and finally read out a single value representing the answer to the problem. In this way, a quantum computer can "act on all input data at the same time."

The second important concept behind quantum computers is entanglement; when two or more qubits become entangled, their internal states are correlated. The measurement of one entangled qubit determines the result of measuring another one.

Properly harnessed, the quantum mechanical mechanisms of superposition and entanglement open up a new paradigm of computing that allows efficient solution of certain types of problems that are hard for classical computers.

Quantum mechanics has been described as the most successful theory in the history of science. Nevertheless, it is decidedly counterintuitive. Objects at the subatomic scale behave quite differently from the objects we encounter in everyday life. As a result, quantum computers are capable of some surprising feats. But it is a mistake to think of them as merely massively parallel or superfast classical computers. In fact, they perform rather poorly at many computing tasks. However, they are capable of tackling certain problems that are out of reach for the computers we are familiar with. This is analogous to the way in which GPUs excel at parallel tasks like rendering millions of pixels, but are very poor at sequential tasks like computing the digits of pi.

Quantum computer designs are based around a quantum register, with quantum gates arranged to establish relationships between the states of the qubits in the register. This means that quantum computers do not work by breaking a problem down into small pieces that can be tackled separately. Instead, the register must be initialized to represent all inputs to the problem. Then the contents of the register evolve in accordance with relationships dictated by the arrangement of gates. This is the fundamental difference between classical digital computers and quantum computers: to process a large dataset, a classical computer must examine data points one at a time and keep records of what it has seen.

A quantum computer, by contrast, loads all data points into the quantum register and then performs operations on the super-imposed data. Once the correct operations have been performed, the contents of the quantum register are output, and each qubit will read either 0 or 1 according to the register state with the highest probability. In this way, quantum computers excel at problems where the answer represents a pattern distributed throughout the input data. Factoring large composite numbers and computing discrete logarithms turn out to be two such problems. The implication of this is that a small quantum computer offers little advantage over a classical computer. But, once a quantum computer of sufficient size can be assembled, it will eclipse classical computers in terms of the efficiency with which it can solve these problems.

Quantum computer developments circa 2019

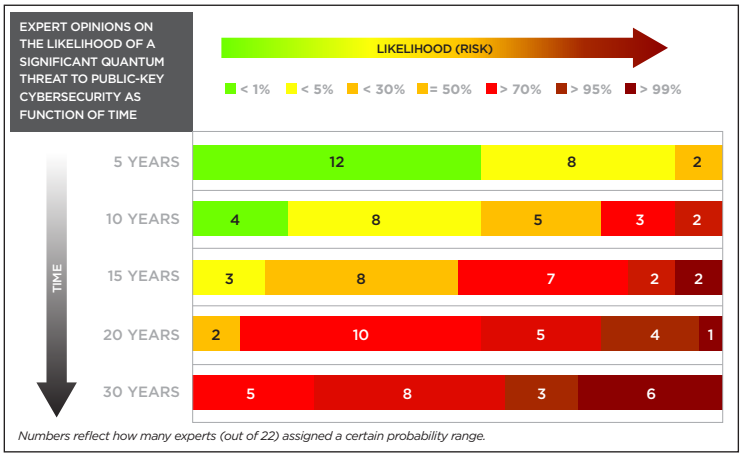
Practical quantum computers are available to researchers today and are beginning to be made available to the general public using cloud-based infrastructures. These machines are not of sufficient size to threaten any public-key algorithms using today's recommended key sizes. A number of engineering obstacles must be overcome before they can be considered a real threat.

In 2019, practical quantum computers have a register size in the double digits. And, in order to break 2048-bit RSA, approximately 4,000 ideal qubits would be required. Furthermore, if the chosen quantum technology demands error-correcting techniques, then each ideal qubit would need to be composed of a sufficiently large array of physical qubits in order to achieve acceptable levels of fault-tolerance.

However, some of the techniques proposed for implementing a quantum computer exploit the very-large-scale-integration techniques used to manufacture state-of-the-art silicon chips. So, the future growth curve may well outpace Moore's Law.

A further obstacle to overcome is the power required to operate a large-scale quantum computer, which, for some of the front-runner technologies, is considerably higher than that required by a conventional digital computer.

When 22 experts were asked about the likelihood of a significant quantum threat to public-key cybersecurity in the next 10 years, 5/22 experts reported that they expected a more than 50% chance. Even more telling is the answers for the next 15 years: 11/22 expected a more than 50% chance of a significant threat occurring. Therefore, it is expected a quantum computer capable of breaking today's public-key algorithms will be available in the late 2020s or early 2030s.[i]



Numbers reflect how many experts (out of 22) assigned a certain probability range.

As published in the Global Risk Institute's Executive Summary Report on the Global Threat Timeline.

This picture can be confused by the existence of a different kind of quantum computer; one called an adiabatic quantum computer[ii]. These are commercially available today with a register of several thousand qubits. While they are effective at addressing optimization problems, they are not suitable for solving cryptanalytic problems.

Cryptographic implications of a large-scale quantum computer

Cryptographic algorithms are generally divided into symmetric and asymmetric (aka "public key") categories. Two algorithms have been discovered that exploit the properties of a quantum computer and that have implications for cryptography: Grover's algorithm[iv] for performing searches, and Shor's algorithm[iii] for performing quantum Fourier transforms.

Symmetric cryptography is vulnerable to a certain quantum algorithm (running on a quantum computer) called Grover's Search. Grover's algorithm searches a dataset and selects the entry that matches a specified set of constraints or satisfies a specified equation. If n is the number of records in the dataset, then a classical computer cannot take any shortcut; in the worst case, it must make n queries. Grover's search, on the other hand, can find the solution with only √n queries.

Cryptographic applications of Grover's algorithm can speed up a search of all possible AES-256 keys to find the one that successfully decrypts a given message, or a search of all possible input strings for one that hashes to a given SHA-256 output. In both cases, Grover's algorithm can do this in 2^{128} queries compared to the 2256 required by a classical computer, thereby effectively halving the security strength of symmetric ciphers and hash functions. For this reason, the effect of a large-scale quantum computer is to double the size of symmetric keys and hash functions. But, AES-256 and SHA-256 remain secure against such an attack today. Nevertheless, given the amount of memory and compute power available to a modern computer, doubling the size of symmetric keys and hash values has a relatively minor impact on performance. So, we generally consider symmetric cryptography to be immune to quantum computers.

Examples of symmetric cryptography:

- Block cipher encryption (3DES and AES).
- Stream cipher encryption (RC4 and Salsa20/ChaCha)
- Hash functions, or cryptographic message digests (SHA-256)

The second, and more significant, quantum algorithm is Shor's algorithm applied to factoring integers and finding discrete logarithms, by taking advantage of the quantum Fourier transform algorithm. Asymmetric cryptography is vulnerable to Shor's Algorithm and all the asymmetric crypto we use today will need to be completely replaced with new "quantum-resistant" or "post-quantum" cryptographic algorithms. Shor's algorithm can find the prime factors of an n-bit number in polynomial(n) time. Like Grover's algorithm, there can still be significant computation involved for a large n, but considering that the leading classical algorithm for integer factorization – the general number field sieve – requires almost 2^n1/3 time, Shor's algorithm reduces the problem of breaking an RSA or ECC key to something that can be achieved in several months, down from centuries or millennia. Breaking an RSA key requires a quantum computer with roughly 2n qubits, whereas breaking an ECC key requires a quantum computer with roughly 6n qubits. But, because ECC keys are smaller than RSA keys, ECC keys can be compromised by a smaller quantum computer than the one required to compromise an RSA key of equivalent strength.

Unlike Grover's algorithm, Shor's algorithm is not defeated by simply increasing the RSA or ECC key size. The existence of a polynomial time attack against an algorithm is considered to break that algorithm, because the falling cost of compute resources over time favors the cryptanalyst over the legitimate user.

Examples of asymmetric cryptography (aka "public key"):

- Key exchange (Elliptic Curve Diffie-Hellman – ECHDE)
- Public key encryption (RSA)
- Digital signatures (RSA, DSA, ECDSA)

Current public-key algorithms are deployed for authentication, digital signature, data encryption, and key establishment purposes. So, once quantum computers of sufficient size become a reality, we will need replacement schemes for each of these functions. Data encryption and key-agreement algorithms are susceptible to a recorded-cipher-text attack, in which an adversary today records exchanges protected by pre-quantum algorithms and stores the cipher text for analysis in the future, once they have access to a large-scale quantum computer, at which point they will be able to recover the plaintext. So, for these key purposes, depending on the required algorithm security lifetime, pre-quantum cryptography will become vulnerable sooner.

Properly designed digital signature schemes used for authentication will remain secure until the day a suitable quantum computer actually comes online. But, once a suitable quantum computer does exist, a signer could repudiate signatures created earlier, claiming that they were forged using a private key broken later by a quantum computer.

Time frame for a working large-scale quantum computer

As mentioned earlier, today's quantum computers are limited in size and therefore pose no threat to present day cryptography. And several significant engineering obstacles must be overcome before the threat becomes real. Nevertheless, experts are of the opinion that these obstacles will be overcome in time. As pointed out by the U.S. National Security Agencyv, many experts predict that a quantum computer capable of breaking today's standard public-key algorithms will be available within the planned life of systems currently under development.

The question of "when" is a hot topic among researchers working on quantum computers. As mentioned above, most expert opinions seem to be converging on the late 2020s to early 2030s. These estimates are based on the rate of advancement of quantum technologies in academic labs. It should be assumed that closed government agencies are ahead of the public domain, and that by the time RSA-2048 is first broken publicly, governments will have had that capability for several years. Expert consensus suggests that organizations that rely on public-key cryptography to protect the confidentiality of information assets should be aware of their data security lifetime and the time required for a cryptographic migration of their infrastructure, and should be working with their public-key infrastructure vendors to ensure they are protected.

In 2017, NIST started a post-quantum cryptography standardization process with 69 algorithms submitted. In July 2020, the third round of reviews left seven finalists and eight alternates for standardization, with draft standards expected to become available between 2022 and 2024.

While the requirement for crypto-agility is already a given, for the reasons mentioned above, crypto-agility, or the ability to switch from one cryptographic scheme and parameter set to another at little or no cost, will become an even more essential feature of deployed systems.

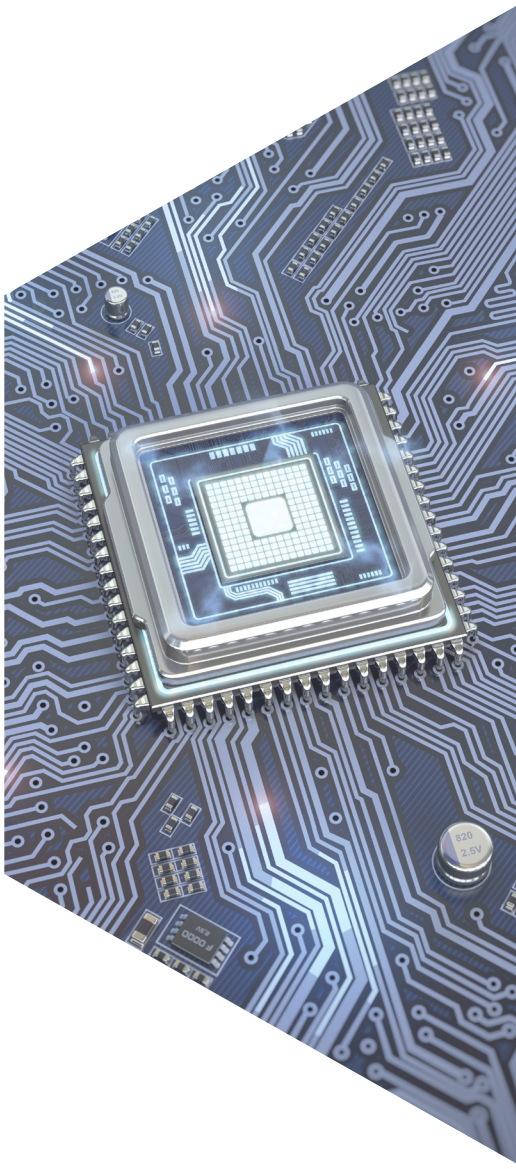
Many experts predict that a quantum computer capable of breaking today's standard public-key algorithms will be available within the planned life of systems currently under development.

Quantum-resistant algorithms have some characteristics that will make agility less straightforward. In the case of some algorithms, key and block sizes are significantly larger than they are for their present-day counterparts. Hash-based signature schemes commonly require their internal state to be preserved so that sub-keys are never reused, even in the face of unexpected loss of power or a disk crash.

Some key-agreement schemes can fail, resulting in no agreement and requiring a retry. In addition, until quantum-resistant algorithms have undergone sufficient expert scrutiny, some schemes will use a combination of a present-day algorithm, and a supposed-quantum-resistant algorithm, such that the combination is at least as strong as the stronger of the two, dubbed by Professor Daniel Bernstein as the "P+Q approach"[vi]. In the absence of a standard solution to this, there will inevitably be interoperability challenges.

Quantum-resistant algorithms are required for a wide variety of settings, including personal computers, smart cards, constrained physical devices, cryptographic hardware modules, high-throughput transaction servers, etc. Different cryptographic primitives are likely to be optimal for each of these settings. And, these will all require standardization.

Any new public-key paradigm has historically taken about 15 years to find acceptance in the marketplace. These timelines were achieved without the time pressure created by a visible and emerging threat. The timeline demanded by the quantum threat means that the usual sequential phasing of mathematical scrutiny, standardization, and industry adoption will have to overlap in the coming years.



Conclusions

The quantum computer represents a revolution in computer science. Its implications will, one day, be far-reaching, especially for information security. While that day is still some way off, it is important to start preparing now. Designers and operators of information systems based on public-key technology should be aware of the implications and consult with suppliers to ensure that quantum-resistant solutions will be in place with sufficient time to test and deploy before the vulnerability can be practically exploited.

What remains to be done

A number of steps must be undertaken before the use of quantum-resistant public-key algorithms becomes routine. The typical process for introducing new cryptographic primitives includes a five- to 10-year period of scrutiny by qualified independent mathematical specialists, followed by a standardization process that involves selecting suitable parameters and data encodings, followed by standardization of protocols and cipher suites, and finally of adoption by product vendors and service providers, and deployment by users.

At the end of 2020, post-quantum algorithm standardization is underway. NIST has started a post-quantum standardization process that is expected to be completed in the next two to three years. It is expected that once completed, these algorithms will become part of the FIPS-140 algorithm suite. ETSI is responsible for the project in Europevii and it is expected there will be close cooperation between the European and US projects.

So the race is on; quantum engineers are racing to overcome the obstacles to a practical large-scale computer, and cryptographers are racing to define, evaluate, implement, and deploy quantum-resistant solutions in time to address the threat posed by a working quantum computer of sufficient scale.

At the end of 2020, post-quantum algorithm standardization is underway.

Candidate quantum-resistant algorithms

Researchers have explored a variety of approaches to avoid an attack using Shor's algorithm. Some of these are revivals of crypto systems invented prior to the discovery of Shor's algorithm, and some are newly invented to address the quantum threat. In late 2019, the leading candidates fall into five broad categories: code-based encryption and key agreement, lattice-based encryption and key agreement and signature, hash-based signature, multivariate signature, and supersingular elliptic curve isogeny key agreement. Clear leaders have yet to emerge from these fields.

The history of code-based schemes dates back to 1978 when McEliece described a public-key encryption scheme based on error-correcting codes[viii], with a significant optimization proposed in 1986 by Niederreiter[ix]. This approach was uncompetitive compared with the leading public-key crypto systems of the time, as it results in much larger public keys. So, the field did not receive serious attention. Interest, however, has revived because of its apparent resistance to quantum attacks.

Another area of research is based on mathematical objects known as lattices[x xi]. Crypto schemes that derive their security from one of several well-studied lattice problems have also been around for some time, and they include the NTRU crypto system introduced in 1996[xii], and the (ring) learning-with-errors family of algorithms introduced by Regev in 2005[xiii xiv].

The lattice problems are believed to be hard in the classical model of computing, and there are no known quantum solutions. But, they also result in large data structures. In order to overcome this limitation, optimizations have been introduced; at the cost of foregoing a formal security proof. The absence of a security proof forces proponents to fall back on the argument that independent researchers have not been able to discover a weakness, despite the length of time during which the schemes have been available for study. This argument supposes that researchers have actually devoted effort to the problem; an argument that is difficult to quantify and verify.

Nevertheless, optimized lattice-based systems are fast and result in relatively small data structures. Naturally, interest in these schemes has taken on new urgency, and researchers are actively working to develop security proofs.

Signature schemes based on hash-functions also have a long history dating back to work done by Merkle and Lamport in the late 1970s[xv]. These schemes rely entirely on the well-understood properties of hash functions to create a digital signature scheme, and since modern hash functions are resistant to an attack using Grover's search, so too is the signature scheme as a whole resistant to a quantum attack. The scheme's private key is expanded into a very large hash-tree in which each leaf node can be used to sign just one message. The downside of hash-based schemes is that they result in a large signature block and some variants require reliable tracking of state, in order to ensure that each leaf node is used no more than once.

More recently, schemes based on elliptic-curve isogenies[xvi] have been proposed. These are attractive because they share implementation characteristics with familiar elliptic-curve schemes, their mathematics are comparatively simple, and their runtime and block sizes are relatively small. However, being a very new set of algorithms, they have yet to receive the independent scrutiny necessary to build confidence in their security. The field of multivariate polynomial-based cryptography[xvii] is in a similar state; there have been several promising results, but their theory still requires exploration and independent scrutiny.

Finally, to the practical: In 2019, many practical experiments in post quantum encryption used one of the lattice-based schemes, or hash-based scheme. Of notable mention is the New Hope key exchange algorithm[xviii], an implementation of ring learning-with-errors[xix]. The hash-based signature scheme SPHINCS, which can use well-known hashing algorithms and Merkle trees, while not requiring state management was considered a favorable candidate. It became one of the eight alternate algorithms selected in June 2020 by NIST as part of their third round of post-quantum algorithm standardization process.

The seven current finalists of NIST's third round of reviews are Classic McEliece, CRYSTALS -KYBER, NTRU, SABER, CRYSTALS-DILITHIUM, FALCON, and Rainbow. Five of them use lattice-based schemes, and the NIST declared that, in their current view, these structured lattice schemes appear to be the most promising general-purpose algorithms for public-key encryption/key encryption management and digital signature schemes.

Entrust's strategy

Entrust follows these developments closely and participates in industry efforts to develop standard solutions. Our priority is to ensure that customers can build solutions and access services that are not vulnerable to a quantum attack in advance of quantum-computer developments.

While multi-vendor interoperable solutions must await industry consensus and widely adopted standards, closed systems can be protected in the near future.

Key to successfully navigating the next decade will be a plan for crypto agility, as the list of favored algorithms and parameter sets can be expected to be somewhat dynamic.

Naturally, FIPS-140-compliance for systems employing purely quantum-resistant algorithms will not be possible until NIST has concluded their standardization of quantum-resistant algorithms. However, both FIPS compliance and quantum safety can be simultaneously achieved today by pairing a FIPS-approved algorithm with a quantum-resistant algorithm in a so-called "P+Q scheme."

Resources

i M. Mosca. Cybersecurity in an era with quantum computers: Will we be ready? Cryptology ePrint Archive, Report 2015/1075, 2015. http://eprint.iacr.org/2015/1075.

ii Tamir, B., & Cohen, E. (2015). Notes on Adiabatic Quantum Computers. arXiv preprint arXiv:1512.07617.

iii Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 212-219). ACM.

iv Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, 41(2), 303-332.

v https://cryptome.org/2016/01/CNSA-Suite-and-Quantum-Computing-FAQ.pdf

vi D. J. Bernstein, (2016, February). The Post-Quantum Internet. Invited Talk at PQCrypto2016 conference, Fukuoka, Japan.

vii https://pqcrypto.eu.org/

viii McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. Coding Thv, 4244, 114-116.

ix Niederreiter, H. (1986). Knapsack-type cryptosystems and algebraic coding theory. PROBLEMS OF CONTROL AND INFORMATION THEORY-PROBLEMY UPRAVLENIYA I TEORII INFORMATSII, 15(2), 159-166.

x Peikert, C. (2016). Decade of Lattice Cryptography. World Scientific.

xi Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In Post-quantum cryptography (pp. 147-191). Springer Berlin Heidelberg.

xii Hoffstein, J., Pipher, J., & Silverman, J. H. (1998, June). NTRU: A ring-based public key cryptosystem. In International Algorithmic Number Theory Symposium (pp. 267-288). Springer Berlin Heidelberg.

xiii Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In Post-quantum cryptography (pp. 147-191). Springer Berlin Heidelberg.

xiv O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In Proc. 37th ACM Symp. on Theory of Computing (STOC), pages 84–93, 2005.

xv Merkle, R. C. (1989, August). A certified digital signature. In Conference on the Theory and Application of Cryptology (pp. 218-238). Springer New York.

xvi De Feo, L., Jao, D., & Plût, J. (2014). Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Journal of Mathematical Cryptology, 8(3), 209-247.

xvii Kipnis, A., Patarin, J., & Goubin, L. (1999, May). Unbalanced oil and vinegar signature schemes. In International Conference on the Theory and Applications of Cryptographic Techniques (pp. 206-222). Springer Berlin Heidelberg.

xviii Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2015). Post-quantum key exchange-a new hope (pp. 2015-1092). Cryptology ePrint Archive, Report 2015/1092, 201 5. http://eprint. iacr. org.

xix M. Braithwaite. (2016, July 7). Experimenting with Post-Quantum Cryptography [Blog post]. Retrieved from https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

For more information

888.690.2424
+1 952 933 1223
info@entrust.com

entrust.com

ABOUT ENTRUST CORPORATION

Entrust keeps the world moving safely by enabling trusted identities, payments and data protection. Today more than ever, people demand seamless, secure experiences, whether they're crossing borders, making a purchase, accessing e-government services, or logging into corporate networks. Entrust offers an unmatched breadth of digital security and credential issuance solutions at the very heart of all these interactions. With more than 2,500 colleagues, a network of global partners, and customers in over 150 countries, it's no wonder the world's most entrusted organizations trust us.

Learn more at
entrust.com

ENTRUST

U.S. Toll-Free Phone: 888 690 2424
International Phone: +1 952 933 1223
info@entrust.com