Journal of Cyber Security Technology

ISSN: 2374-2917 (Print) 2374-2925 (Online) Journal homepage: http://www.tandfonline.com/loi/tsec20

Will quantum computers be the end of public key encryption?

William Buchanan & Alan Woodward

JOURNAL OF CYBER SECURITY TECHNOLOGY, 2017
VOL. 1, NO. 1, 1–22
http://dx.doi.org/10.1080/23742917.2016.1226650

Taylor & Francis
Taylor & Francis Group

Will quantum computers be the end of public key encryption?

William Buchanan[a] and Alan Woodward [b]

[a]School of Computing, Napier University, Edinburgh, UK; [b]Department of Computer Science, University of Surrey, Guildford, UK

ABSTRACT
The emergence of practical quantum computers poses a significant threat to the most popular public key cryptographic schemes in current use. While we know that the well-understood algorithms for factoring large composites and solving the discrete logarithm problem run at best in superpolynomial time on conventional computers, new, less well understood algorithms run in polynomial time on certain quantum computer architectures. Many appear to be heralding this next step in computing as 'the end of public key encryption'. We argue that this is not the case and that there are many fields of mathematics that can be used for creating 'quantum resistant' cryptographic schemes. We present a high-level review of the threat posed by quantum computers, using RSA and Shor's algorithm as an example but we explain why we feel that the range of quantum algorithms that pose a threat to public key encryption schemes is likely to be limited in future. We discuss some of the other schemes that we believe could form the basis for public key encryption schemes, some of which could enter widespread use in the very near future, and indicate why some are more likely to be adopted.

ARTICLE HISTORY
Received 10 June 2016
Accepted 17 August 2016

KEYWORDS
Post-quantum cryptography; Shor's algorithm; hidden subset problem

1. Emergence of public key encryption

One of the perennial problems with symmetric encryption has been establishing a secure channel over which to pass the shared secret key. It has always been much easier to compromise the transmission of the keys than to try to find some weakness in the encryption algorithm. Military and other government organisations have put in place elaborate methods of passing secret keys: they pass secrets more generally, so using similar channels to pass an encryption key is not a great leap.

However, as the general public has become more connected, and especially with the commercialisation of the Internet, encryption has become a requirement for the vast majority of networked users. As a result the traditional methods of passing secret keys have become impractical, if only because you might not actually know who you want to communicate with in an encrypted fashion far enough in advance to securely transmit a shared key.

CONTACT Alan Woodward alan.woodward@surrey.ac.uk Department of Computer Science, University of Surrey, GU2 7XH, Guildford, UK

This article was originally published with errors. This version has been corrected. Please see Erratum (http://dx.doi.org/10.1080/23742917.2016.1246282).

As computers increasingly became networked in the 1970s, it was recognised that encryption needed to be more accessible, so a great deal of work was done on algorithms that could ensure that if a key had been passed over relatively insecure channels, it was not compromised. Those most associated with the emergence of public key encryption were Whifield Diffie and Martin Hellman, but they were not the only ones: some were hidden as they operated inside organisations such as GCHQ [1]. However, it was when Diffie and Hellman published their seminal paper entitled 'New Directions in Cryptography' [2] that, as far as the general public was concerned, public key encryption was born. Diffie–Hellman key exchange still features today as part of the protocol in some highly secure messaging applications [3].

When the implementation of the early public key encryption methods was compared to symmetric-key encryption, it was found that public key encryption was significantly slower and hence communication using public key encryption alone was, at the very least, going to require far more processing power than would otherwise have been the case. But, the original problem being studied was secure key transmission so why not use public key encryption to securely transmit the secret key for a symmetric-key algorithm, and then use the faster, more efficient symmetric-key encryption algorithm for the bulk of the communication. In essence, that is how most public key encryption works today.

The majority of public key encryption algorithms rely upon a mathematical function that is easy to compute in one direction but computationally hard to reverse. One of the earliest public key encryption schemes, RSA, named after Ron Rivest, Adi Shamir, and Leonard Adleman [4], was based upon the mathematics of prime numbers. If one has a number, n, that is derived from multiplying two prime numbers, and n is very large, it is practically impossible to calculate what the two constituent prime numbers were (known as factoring) within a time frame that would make the decrypted data useful. To date RSA, and similar public key crypto schemes, have proven to be secure.

Of course, 'secure' is a relative word and there are many ways of recovering the private element that was used to derive, for example, the n used in RSA. These do not involve a direct attack on the RSA algorithm but instead, they use side channel attacks. Research in the past two years that has been able to recover the private keys from the two most popular public key cryptosystems (RSA and Elliptic Curve Digital Signature Algorithm (ECDSA) [5]) in ways ranging from the electronic noise emitted by laptops [6], through to attacks on shared cache memory in virtual machines using the same physical infrastructure [7].

However, there is now an emerging threat that does attack the public key algorithms directly: quantum computers.

2. Breaking RSA

An example of how quantum computers will impact current public key encryption schemes is exemplified by studying how the RSA algorithm is at risk.

2.1. *Factoring algorithms*

Breaking the RSA [4] encryption scheme can be done on conventional computers through factoring. There are several algorithms known. The ways in which RSA might be attacked are well studied and 20 years ago the attacks were well catalogued [8]. The intervening years has seen new forms of attack reported [9] but the problem is always the efficiency with which we can implement these factoring algorithms on conventional computers.

To understand how RSA can be attacked you need only know how the keys for the RSA algorithm are generated, i.e.

(1) Choose two distinct prime numbers p and q
(2) Compute $n = p.q$
(3) Compute $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$, which is known as Euler's Totient. This value is kept private
(4) Choose an integer e such that $1 < e < \phi(n)$ and the greatest common divisor of $e\ \phi(n)$ is 1
(5) Determine $d.e{\equiv}1\text{mod}\ \phi(n)$

The public key consists of the modulus n and the encryption exponent e. The private key consists of the modulus n and the decryption exponent d.

Typically attacks on RSA employ one of three approaches:

(1) Factor $n$and hence compute $\phi(n)$ and thence d; or
(2) Determine $\phi(n)$ directly (without factoring) and thence d; or
(3) Determine d directly.

Current opinion suggests (although we are not aware of any formal proofs) that each is as computationally difficult as the other. However, it is factoring that has the greatest number of algorithms, as they have evolved over the history of the underlying mathematics not just use of composites in cryptography.

The factoring algorithms generally considered when factoring a large composite number, n, on a conventional computer are:

(1) Fermat's difference of squares.
(2) Euler's factoring method where Euler extended Fermat's method.
(3) Kaitchik's method: as late as 1945 Kraitchik revisited Fermat's approach and found a further representation where the difference of squares was a multiple of n, allowing further sieving of results. It is the basis of the quadratic sieve.
(4) Pollard's P-1 method.
(5) Pollard's rho method: published just one year after his P-1 method.
(6) Quadratic Sieve: this was developed in the 1980s.

The General Number Field Sieve is the fastest factoring algorithm we currently have on conventional computers. It relies upon choosing polynomials that have a common root related to n. The efficiency of the algorithm depends very much on the choice of polynomials and there is no known method for optimally making this choice. Hence, it is not always reliable.

Although the basic mathematics has been studied for hundreds of years, and despite our modern computing power, RSA has so far resisted all published attacks for keys of the size used in modern implementations. Research continues to be published on improved algorithms for factoring composites by considering the fundamental mathematics (for example, [10]) but there is little expectation that it will lead to a fundamental breakthrough in attacking RSA.

2.2. *Quantum computing*

In the computers with which we are familiar, processing is done using bits. A bit has two possible values: 0 and 1. A quantum computer uses qubits, a term introduced by Stephen Weisner when he proposed quantum money [11]. When measured, a qubit also has the values 0 or 1, but during computation a qubit is both 0 and 1 simultaneously. In quantum physics this is called 'superposition' [12]. So, if you have two qubits, you can have four possible states, three qubits gives eight possible states; and all simultaneously. In a bit-based computer you have the same number of possible states but only one exists at any one time. It is the fact that these states, $|0\rangle$ and $|1\rangle$, can exist simultaneously in a quantum computer, which is both counter-intuitive and extraordinarily powerful.

Qubits are manipulated using quantum logic gates in the same way that conventional computation is done by manipulating bits using logic gates. In essence, you can apply a computation to all possible values of the qubits simultaneously, thereby increasing the amount of computation you can undertake in any given time over that you could otherwise do in a conventional computer [13].

The algorithms developed over many years for conventional computers have been optimised for conventional architecture, and different algorithms are needed to run on a quantum computer. Hence, trying to compare speeds of conventional and quantum computers can be spurious. It is important to note that quantum computers do not offer a universal speed advantage over conventional computers. When researchers talk of 'quantum parallelism' it does not apply to all algorithms. In the case of factoring large composites, for example, the algorithm must be tailored to achieve the required speed up.

It is noteworthy that since the development of early quantum algorithms there has not been a large array of new algorithms as was expected. Shor himself commented on this point when he wrote [14] that quantum algorithms appear to fall into one of three classes: those that we now consider part of the Hidden Subgroup Problem (HSP) [15] [16], Search algorithms such as Grover's algorithm [17], and the original quantum systems simulations, which were the original motivation for Feynman's suggestion for a quantum computer.

The US National Institute of Standards and Technology (NIST) maintains a catalogue of known quantum algorithms (known as the Quantum Algorithm Zoo) [18]. This currently contains 57 different algorithms. However, even this could be considered an overestimate as the majority of the algorithms are a specific application of one of the classes suggested by Shor, with the possible addition of another class: solving linear equations. Of the 57 algorithms listed in the Quantum Algorithm Zoo, only 27 provide superpolynomial speedup, and these in turn rely upon a very small number of primitives such as the Quantum Fourier Transform (QFT).

Researchers have also suggested that the list of quantum algorithms should have another class called Quantum Walks [19]. This newest category would incorporate quantum algorithms that are the analogues of conventional algorithms based upon random walks and Markov chains.

Markov chains and random walks have proven to be very powerful tools in speeding up algorithms on conventional computers. However, whether the analogue will extend to quantum computers is still an open question. In any event, it is unlikely that quantum algorithms derived from such an approach would demonstrate the superpolynomial speedup seen in the quantum algorithms derived from solving variants of the HSP.

It appears that whilst the quantum algorithms derived from solving HSP pose a known threat, Grover's algorithm is the only other quantum class that could pose a threat to public key encryption. And, where Grover's algorithm has been proven to pose a threat to a particular scheme, it does not offer the same magnitude of speed-up seen in the solutions to the HSP.

Given the history of quantum algorithm development, it appears highly unlikely that any new class of quantum algorithm is likely to emerge. Thus, the basic test for whether or not an encryption scheme is quantum proof is most likely to remain how it resists algorithms from within the HSP.

2.3. *Factoring using a quantum computer*

One of the earliest algorithms developed for quantum computing was Peter Shor's 1994 algorithm [20]. It was designed to factor composite numbers into their prime number components using Euler's method. However, Shor recognised that part of Euler's method (determining the period of the function) could be computed using a version of the fast Fourier transform modified to take advantage of quantum parallelism.

For those seeking a detailed explanation of Shor's algorithm and how it achieves the quantum speed up, we have included this at Appendices 1 and 2. We describe Shor's algorithm using the Phase Estimation Algorithm as an Order Finding Algorithm, which is a specific instance of the HSP. Appendix 3 shows how the subgroup problem leads not just to an Order Finding Algorithm but also to quantum algorithms for Period Finding and Discrete Logarithms, which are the basis for quantum attacks on elliptic curve cryptography and finite field cryptography.

Shor's algorithm runs on a quantum computer in polynomial time as opposed the General Number Field Sieve running on a conventional computer, which runs in super-polynomial time [21]. Even with the largest values of n in use today in RSA, this means it would be feasible to decrypt messages in meaningful timescale using Shor's algorithm on a quantum computer.

2.4. *Practical considerations*

Current implementations of quantum computers require large, expensive infrastructure for supercooling and electromagnetic shielding, and even then we have been able to assemble only a handful of qubits in a single processor. However, the history of computing shows that although conventional computers began by requiring similar infrastructure, size soon shrank dramatically and the environmental requirements for today's machines allow for domestic operation. The pattern can already be seen with a company in Canada, D-Wave Systems Inc. [22], which offers access to a form of quantum-based computing. The D-Wave system appears just as mainframe computing did 20 years ago but if it follows the same trajectory as conventional computing, routine operation of quantum computers will be in routine, widespread use before 2030.

However, D-Wave's version of a quantum computer uses adiabatic quantum computing, [23] which does not support Shor's algorithm [24].

Despite the engineering difficulties encountered in building a quantum computer, work continues on improving how gate-based quantum computers (the architecture required for Shor's algorithm), might better be used to factor large composites. Work published earlier in 2016 [25] has shown for the first time that quantum computers could be used to factor large prime composites using Shor's algorithm, with 99% confidence, which is a major improvement on previous work. And, even though the number factored is only 15, most thought it would take 12 qubits to run the algorithm successfully, but the same researchers ran Shor's algorithm using only five qubits. These five qubits were only five atoms in an ion trap.

Not only have they shown in [25] that fewer qubits than previously thought can be used, the architecture suggested by their experiment is scalable. That means that, provided researchers can keep the atoms stable in the ion trap, they could expand the size of the numbers being factored. The team at Innsbruck who have been building the equipment have shown particular expertise in achieving just such a scaling of these ion traps. Hence, whilst many believe it will take decades to achieve a scalable quantum computer, this recent work suggests that running Shor's algorithm on a quantum computer will be achieved in the foreseeable future.

The demonstration reported in [25] is believed by some researchers not to support the necessary steps (see Appendix 2) of Continued Fraction Expansion, and Quantum Modular Exponential, in Shor's algorithm [26]. This suggests that it may not be a true implementation of Shor's algorithm. Such doubt demonstrates the difficulties of comparing various implementations of quantum algorithms.

Improvements in the technology used to implement quantum computers, and the efficiency of algorithms that attack the most popular public key encryption schemes, will converge within a very few years. In the foreseeable future we will have programmable, relatively inexpensive quantum computers capable of running more than just Shor's algorithm; the threat to current public key encryption appears high.

Gartner showed in their 2015 Hype Cycle that it will be more than 10 years away [27] and yet IBM, which is the only organisation to have a functioning five-qubit processor available for general use, suggest they will have a 100 qubit based system by 2020 [28]. Also, a large amount of money is being invested in quantum technologies by governments. For example, the UK government announced in 2015 that it would invest £270 million in the field [29].

The facts indicate that a quantum computer capable of threatening public key encryption systems will exist by 2025. The fact that quantum computers, when they emerge, will not be universally available affects the threat model posed by these devices. Whilst governments might be the first to operate viable quantum computers, the way in which D-Wave and IBM have introduced their systems shows that access to quantum computing may not be restricted only to government agencies. Shared system will emerge rapidly once viable quantum computers can be reliably produced.

Table 1. NIST impact analysis of quantum computing on encryption schemes [31].

Cryptographic algorithm	Type	Purpose	Impact from large-scale quantum computer
AES-256	Symmetric key	Encryption	Larger key sizes needed
SHA-256, SHA-3		Hash functions	Larger key sizes needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECHD (Elliptic Curve Cryptography)	Public key	Signatures, key establishment	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key establishment	No longer secure

2.5. *Public key encryption schemes affected*

RSA is only one of the public key encryption schemes in use today. However, following his factoring algorithm, Shor quickly developed a quantum algorithm for solving the discrete logarithm problem [30], and in so doing showed that quantum computers pose a similar risk to other popular public key encryption schemes. Further quantum algorithms derived from the HSP have shown that there are algorithms that pose a threat to all current popular public key encryption schemes (see Appendix 3).

A recent report from the NIST [31] contained a table (Table 1) that indicates quantum computer will be the end for current public key encryption.

Importantly, the NIST review shows that if the key lengths are sufficient, symmetric key encryption (specifically the Advanced Encryption Standard AES) is quantum resistant. Although Grover's algorithm poses a threat to some symmetric key schemes, most analysis indicates that simply doubling the key length will be sufficient protection. Thus it is the key exchange element of public key encryption that is under threat.

As the symmetric encryption schemes remain quantum resistant, it has been suggested that using Kerberos for key management may be an immediate answer to the threat posed by quantum computers. It is already in widespread use and could be deployed publicly if a suitable post-quantum public key system is not found [32]. However, several factors militate against Kerberos being used as a truly public key exchange mechanism:

(1) Kerberos would need to be integrated into web services;
(2) Open source implementations, with a royalty free license, would be required;
(3) Service providers would need to adopt a common federation, certification and auditing standard;
(4) Binding agreements with service providers would be required to engender trust.

Hence, most believe that a post-quantum asymmetric candidate needs to be developed.

3. Post-quantum public key candidates

3.1. *Quantum key distribution*

Many years before Shor developed his algorithm, there was a method, also based upon quantum principles, which many believe is the answer to the risk posed by quantum computers. The method is Quantum Key Distribution (QKD).

QKD first appeared in 1984 when Bennett and Brassard developed their BB84 protocol [33], although it was not until 1992 that it was reliably demonstrated experimentally [34]. It relies upon the fact that photons can be polarised to form qubits but that the polarisation that represents the basis states |0 |1, can be alternated between two pairs of polarisations. If anyone attempts to intercept a polarised photon, it will affect the polarisation and make it impossible to consistently recover the key that is encoded in the stream of qubits [35] [36].

Since the BB84 protocol, other protocols have been developed, most notably that by Ekert in 1991, which uses entangled pairs of photons [37]. However, all fundamentally rely upon the principle that interception disturbs the quantum state.

QKD is unlikely to be a universal answer to the risk posed by quantum computers for three reasons:

(1) The security relies upon quantum mechanical principles and only recently has work been publicly reported on developing formal proof that QKD is semantically secure; [38]
(2) QKD requires an expensive infrastructure;
(3) QKD can operate over relatively short distances: the longest to date being approximately 150 km although some reports suggest over 300 km has now been achieved [39].

Cryptographers will provide further proofs of the semantic security of QKD, and the engineering challenges associated with using QKD at range will be accommodated. In 2016 architectures are already being suggested for an extensive ‘quantum network’ [40]. However, all such architectures rely upon repeaters which, from a security perspective, will always be weak points thereby undermining the original end-to-end security that attracted many to QKD.

Assuming such large-scale networks could be implemented and proven secure, they will always be based upon a physical infrastructure for which costs scale with size. Hence, whilst it might be appropriate for portions of the backbone of a large network, it is unlikely to be suitable for multiple end point security.

The principles behind QKD was showing advantages over other post-quantum candidates in Unconditional Secure Signature (USS) schemes. This was because it made fewer assumptions than conventional USS schemes, and particularly did not have the same reliance on trusted third parties. However, recently a USS scheme was proposed [41] that does not rely upon a quantum distribution scheme (QDS), makes the same limited assumptions as QDS and has the added advantages that it requires fewer secret bits to be shared whilst generating a shorter signature.

3.2. *Mathematically based solutions*

There is a range of alternative mathematical problems to those used in RSA and ECDSA that have already been implemented as public key cryptographic schemes, and for which the HSP does not apply. That is, they appear to be quantum resistant. These implementations include:

(1) Buchmann–Williams Key Establishment [42]
(2) The NTRU Cryptosystem [43]
(3) The Goldreich–Goldwasser–Halevi Cryptosystem [44]
(4) The Ajtai–Dwork Cryptosystem [45]
(5) The McEliece Cryptosystem [46]

Early analysis of the levels of quantum resistance of these crypto schemes [47] dismissed the Buchmann–Williams scheme as a post-quantum candidate. Although the scheme was based upon a mathematical problem (Pell's equation) for which the solution is exponentially slower than the best known factoring algorithms, it is susceptible to a quantum-based attack. It serves as a good example of why simply changing the mathematical problem upon which a scheme is based does not necessarily make it quantum resistant.

The ideal candidate would be not only quantum resistant but would also:

(1) Be based upon a mathematical problem that is NP-hard to solve on conventional computers, that is, something that replaces the one-way functions that underlie the current popular public key encryption systems such as RSA;
(2) Be efficient to implement, requiring minimal computing power without compromising the difficulty in solving the underlying mathematical problem;
(3) Generate a small public key for ease of storage and transmission amongst multiple parties;
(4) Enable perfect forward secrecy.

The mathematical problems that are most actively being investigated are

(1) Lattice-based cryptography
(2) Multivariate-based cryptography
(3) Hash-based signatures
(4) Code-based cryptography
(5) Supersingular elliptic curves-based cryptography.

The existing alternatives, and new schemes emerging, from these areas of mathematics, do not all necessarily satisfy the characteristics of an ideal scheme.

3.2.1. *Lattice-based cryptography*

The Ajtai–Dwork (AD), Goldreich–Goldwasser–Halevi (GGH) and NTRU encryption schemes mentioned above are lattice based.

The AD system has the drawback that the pubic key is large and it causes message expansion. Also, traditional cryptanalysis has shown any practical implementation of AD is not sufficiently secure [48]. Hence, unless further work improves these aspects it is unlikely that the AD system is a realistic candidate.

The GGH scheme is more efficient than the AD system but cryptanalysis indicates that although it is based upon the Closest Vector Problem (CVP) in a lattice, which is known to be NP-hard, decryption can be reduced to solving CVP instances that are much easier than the general problem [49]. Such concerns militate against it being a practical candidate.

NTRU has no known feasible attack despite many years of research attempting to develop one. There is also a related but newer NTRU signature based and BLISS [50] scheme. Researchers have found sub-exponential time attacks on NTRU-like schemes but they are still not practical attacks. The most effective attack to date is a hybrid attack [51].

The project sponsored by the European Commission to study post-quantum candidates [52] suggested that the Stehle–Steinfeld variant of NTRU be considered for standardisation rather than the patented NTRU algorithm [53] [54]. However, several of the NTRU algorithm patents are due to expire in 2017 [55], so the patents alone will no longer be an issue.

The security of the NTRU encryption scheme and the BLISS signature is believed to depend on the CVP. However, we are not aware that the security of NTRU and BLISS is provably reducible to the CVP.

Implementations of lattice encryption may not be subject to attacks using Shor's algorithm, but Grover's quantum algorithm [17] might still pose a threat. For example, if you consider the BLISS Ring–LWE (Learning with Errors) Signature Scheme [56], Grover's algorithm can be used to mount an attack against the random oracle element of the scheme. The original implementation used an oracle that is not collision resistant. Hence conducting a preimage search would be a suitable form of attack, and preimage searches (using, say, Grover's algorithm) are where quantum computers do provide dramatic speed advantages.

It is noteworthy that the D-Wave quantum computer does run Grover's algorithm. It is this very ability to improve searching large datasets that prompted Google to buy one of these systems and with which they are already producing results that they claim are 100 million times faster than conventional computers [57].

Other concerns about lattice-based encryption emerged when it was shown that the Soliloquy lattice-based scheme developed by CESG [58] was vulnerable to a quantum algorithm from those based on solving the Abelian Hidden Subgroup Problems. The authors concluded in their paper that their finding indicated that this quantum algorithms posed a wider threat than had 'traditionally' been documented. That appears to be an oversimplification for the reasons given in the dialogue that was posted to the cryptographic algorithms mailing list, entitled 'What does GCHQ's 'cautionary tale' mean for lattice cryptography?' The key points were made in the following extracted points [59]:

(1) 'Lattice-based' cryptosystems are not based on 'lattices' per se, but rather on certain *computational problems* on lattices. There are many kinds of lattice problems, not all of which appear to be equally hard – therefore, not all lattice-based cryptosystems offer the same qualitative level of security. For ideal lattices, the choice of *problem* matters at least as much as the choice of *ring*.

(2) Soliloquy is *not* representative of modern ideal/lattice-based cryptography. In contrast to the vast majority of works in the area, Soliloquy does not come with any meaningful security proofs/reductions. Moreover, it relies on a more specialised – and hence potentially easier to break – lattice problem.

(3) The GCHQ attack works because Soliloquy unexpectedly turns out to rely on 'weak ideals' that can be broken more efficiently than general ideals. There is a direct link between the lack of security reduction and this unforeseen reliance on weak ideals.

(4) The GCHQ attack *does not* affect the vast majority of ideal-lattice-based cryptosystems. In particular, it does not work against any system that has a 'worst-case' security proof via the Ring–LWE or Ring–SIS problems.

(5) This state of affairs underscores the importance of worst-case security proofs in ideal/lattice-based cryptography. Among other advantages, such proofs ensure that a cryptosystem does not have any unexpected reliance on 'weak' instances.

Other lattice based schemes are rapidly developing such as LWE and Ring–LWE. The use of ideal lattices combined with LWE is currently producing significant results [60] [61]. Since 2014 we have seen this mature into a ciphersuite for TLS [62] and further modified to form the New Hope scheme [63], which notably has been chosen by Google as their experimental implementation for the Chrome browser.

The lower bound on the security of these Ring-LWE schemes reduces to the Shortest Vector Problem (SVP), which is known also to be NP-hard [61].

Recently published, tight security analysis of NTRU indicates that it is theoretically weaker than Ring-LWE, although this does not translate into a practical attack. However, the researchers concluded that the effort in transforming an NTRU scheme to a Ring-LWE-based system is small enough for key exchange mechanisms as to be worth doing [64].

3.2.2. *Multivariate-based cryptography*

This relies upon the difficulty of solving systems of multivariate equations. Many attempts to build encryption schemes based on this principle have failed. However, the Rainbow scheme does show promise as a quantum resistant signature scheme, but a patent has been filed [65].

The Rainbow Multivariate Equation Signature Scheme is a member of a class of multivariate quadratic equation cryptosystems called Unbalanced Oil and Vinegar (UOV) Cryptosystems which has been shown to reduce to a generic multivariate quadratic UOV systems [66]. We know that the Multivariate Quadratic Equation Solving problem is NP-hard but the public key is relatively large, although it is still an order of magnitude less than that for the McEliece Goppa code based scheme.

3.2.3. *Hash-based signatures*

Hash-based signature schemes were introduced around the same time as RSA and have been studied as a possible alternative to schemes that rely upon number theory. The most widely studied are Lamport signatures and the Merkle signature scheme.

There is a proof that the Merkle Hash Tree signatures reduce to the security of the underlying hash function, [67] which is a reason why the EU project recommended it as a quantum resistant candidate.

Until interest was renewed because of their possible quantum resistance, these schemes had been dismissed due to the limitations on the number of signatures that can be created from any one private key. This has not changed and so is likely to impede its widespread adoption.

3.2.4. *Code-based cryptography*

This relies upon error-correcting codes. It includes the McEliece scheme as well as the variants Niederriter encryption scheme and Courtois, Finiasz and Sendrier signature schemes.

The original method McEliece developed in the 1970s relied upon random Goppa codes. No fundamental flaw has been found since it was first developed. The McEliece Encryption System depends upon the Syndrome Decoding Problem (SDP) [68] which we know is NP-hard [69].

The most effective attacks on McEliece are the information-set decoding algorithms. It has been suggested that Grover's algorithm may been a means of conducting such attacks [70]. However, the claims for the speed-up were shown to be based upon incorrect assumptions about Grover's algorithm [71] and the speed-up was not significant enough to pose a serious threat. Hence, the McEliece scheme is currently considered quantum resistant.

The main disadvantage of the McEliece scheme is the length of the public key. Whereas other post-quantum candidates have public keys as low as 1 kB in size, the public key for McEliece, when based on Goppa codes, is approximately 1 MB. There have been alternative codes suggested in order to reduce this issue but concerns have been raised about the security of these alternatives. It remains an area of active research and it may yet produce a candidate that is as robust as the original Goppa code based scheme whilst being efficient to implement.

3.2.5. *Supersingular elliptic curve isogeny cryptography*

This cryptographic system relies on the properties of supersingular elliptic curves and the difficulty of constructing an isogeny between two such curves with the same number of points. The idea of applying this problem for cryptographic purposes was first considered in 1999 [72] and the early cryptographic primitives published have so far remained unbroken.

The technique is attractive as it works in a similar way to existing Diffie–Hellman implementations [73]. However, we are unaware of any security reduction of this scheme to a known NP-hard problem. There is a fundamental assumption made about the equivalence of how hard it is to solve the Computational Supersingular Isogeny (CSSI) problem, the Supersingular Computational Diffie–Hellman (SSCDH) problem, and the Supersingular Decision Diffie–Hellman (SSDDH) problem, and a conjecture that these problems cannot be solved in polynomial time [74].

4. Conclusion

Gate-based quantum computers do pose a significant threat to the public key encryption schemes most widely used today via, for example, Shor's algorithm, even though quantum computers do not offer a generalised speed up in computing all algorithms. The threat is likely to move from the theoretical to the practical within the next 10 years.

It is unlikely that QKD will be a complete answer to the threat, primarily because of cost and scale. However, there are a number of alternatives, some of which have already begun experimental deployment. Whilst some of these alternatives may prove vulnerable to quantum attack using an algorithm derived from one of the existing classes of quantum algorithm, there is nothing to suggest that a new class of quantum algorithm will be discovered that might pose a threat.

Of the existing post-quantum candidates the most likely to succeed are the lattice-based cryptography schemes, specifically the Ring–LWE schemes, which appear to have some advantages over the older NTRU schemes.

The development of gate-based quantum computers will *not* be the end of public key encryption, but the open question remains as to whether a truly quantum resistant system, that is cost effective at scale, will be in widespread use before the threat is in operation.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

William Buchanan is a professor in the School of Computing at Edinburgh Napier University, and a Fellow of the BCS and the IET. He currently leads the Centre for Distributed Computing, Net-works, and Security, and works in the areas of security, next generation user interfaces, Web-based infrastructures, e-Crime, intrusion detection systems, digital forensics, e-Health, mobile computing, agent-based systems and simulation. Bill has one of the most extensive academic sites in the World, and is involved in many areas of novel research and teaching in computing. He has published over 27 academic books, and over 120 academic research papers, along with several awards for excellence in knowledge transfer, and for teaching, such as winning at the I ♥ **my Tutor Awards** (Student voted), Edinburgh Napier University, 2011, and has supervised many award winning student projects. He has also led a project which is a finalist for the BCS/ Computing Award for Technological Excellence.

Alan Woodward is a visiting professor in the Department of Computer Science at the University of Surrey, and Fellow and chartered member of the British Computer Society, Institute of Physics and the Royal Statistical Society. He began as a physicist. However, he developed an interest in computing early on through signal processing for gamma ray burst detectors, and so switched to engineering after his BSc. His postgraduate research at the Institute of Sound and Vibration Research (ISVR), University of Southampton, was in adaptive filtering, and novel methods of recovering corrupted signals. After leaving the ISVR Alan worked for the UK government for many years. He still provides advice to government organisations through advisory roles in, for example, Europol. He has many years of experience in, and continues to conduct research into, cyber security, covert communications, forensic computing and image/signal processing.

ORCID

Alan Woodward http://orcid.org/0000-0002-8472-4836

References

[1] Ellis JH. The possibility of secure non-secret digital encryption. Communications Security & Evaluation Group, Government Communications Headquarters; 1970.

[2] Diffie W, Hellman M. New directions in cryptography. IEEE Trans Inf Theory. 1976;22(6):644–654. DOI:10.1109/TIT.1976.1055638.

[3] Open Whipser Systems. Advanced cryptographic ratcheting. 2013. Available from: https://whispersystems.org/blog/advanced-ratcheting/.

[4] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Commun ACM. 1978;21(2):120–126. DOI:10.1145/359340.359342.

[5] Miller VS. Use of elliptic curves in cryptography. In: Williams HC, editor. Advances in Cryptology — CRYPTO '85 Proceedings; 1985. Available from: http://link.springer.com/chapter/10.1007%2F3-540-39799-X_31

[6] Genkin D, Pachmanov L, Pipman I, et al. ECDSA Key extraction from mobile devices via nonintrusive physical side channels. 2016. Cryptology ePrint Archive, (Report 2016/230).

[7] Inci MS, Gulmezoglu B, Irazoqui G, et al. Cache attacks enable bulk key recovery on the cloud. 2016. Cryptology ePrint Archive, (Report 2016/596).

[8] Boneh D. Twenty years of attacks on the RSA cryptosyste. Not AMS. 1998;46(2):203–213.

[9] Nitaj A, Ariffin MRK, Nassr DI, et al. New attacks on the RSA cryptosystem. 2014. Cryptology ePrint Archive, (Report 2014/549).

[10] Deng Y, Pan Y. An algorithm for factoring integers. 2012. Cryptology ePrint Archive, (Report 2012/097).

[11] Weisner S. Conjugate coding. ACM Spec Interest Group Algorithms Comput Theory. 1983;15:78–88.

[12] Dirac PAM. The fundamental equations of quantum mechanics. In: Proceedings of the Royal Society A. Vols. Mathematical, Physical and Engineering Sciences; 1925.

[13] Nielson MA, Chuang IL. Quantum computation and quantum information. Cambridge: Cambridge University Press; 2010.

[14] Shor PW. Why haven't more quantum algorithms been found? J Acm. 2003;50(1):87–90. DOI:10.1145/602382.602408.

[15] Lomonaco SJ, Kauffman LH. Quantum hidden subgroup problems: a mathematical perspective. 2002. CERN quant-ph/0201095. DOI:10.1044/1059-0889(2002/er01). Available from: http://arxiv.org/abs/quant-ph/0201095

[16] Wang F. The hidden subgroup problem. 2010. arXiv:1008.0010.

[17] Grover LK. A fast quantum mechanical algorithm for database searches. In: Proceedings, 28th Annual ACM Symposium on the Theory of Computing; 1996. p. 212. Available from: http://dl.acm.org/citation.cfm?id=237866

[18] National US. Institute of standards & technology. Quantum Algorithm Zoo. Available from: http://math.nist.gov/quantum/zoo/.

[19] Montanaro A. Quantum algorithms: an overview. NPJ Quantum Inf. 2016;2:15023. doi:10.1038/npjqi.2015.23.

[20] Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. Sante Fe; 1994. Available from: http://dl.acm.org/citation.cfm?id=1398518

[21] Lenstra AK, Lenstra HW. The development of the number field sieve. Berlin, Heidelberg: Springer-Verlag; 1993. Available from: http://link.springer.com/book/10.1007%2FBFb0091534

[22] Dwave. Available from: http://www.dwavesys.com/.

[23] Van SHW, Der Ploeg A, Izmalkov A, et al. Adiabatic quantum computation with flux qubits, first experimental results. IEEE Trans Appl Supercond. 2007;17(2):113–119. doi:10.1109/TASC.2007.898156.

[24] Beauregard S. Circuit for Shor's algorithm using 2n+3 qubits. Quantum Inf Comput. 2002;3 (2):175–185.

[25] Monz T, Nigg D, Martinez EA, et al. Realization of a scalable Shor algorithm. Science. 2016;351 (6277):1068–1070. doi:10.1126/science.aad9480.

[26] Cao Z, Liu L. Comment on "Realization of a scalable Shor algorithm". 2015. Cryptology ePrint Archive, (Report 2015/1133).

[27] Gartner. Technology hype cycle. 2015. Available from: http://www.gartner.com/newsroom/id/3114217.

[28] IBM. Quantum computing primer. 2016. Available from: https://www.research.ibm.com/quantum/expertise.html.

[29] EPSRC. Quantum technologies. 2015. Available from: https://www.epsrc.ac.uk/research/ourportfolio/themes/quantumtech/.

[30] Shor P. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. 1994. Available from: http://dl.acm.org/citation.cfm?id=1398518

[31] Chen L, Yi-Kai L, Jordan S, et al. Report on Post-Quantum Cryptography (NISTIR 8105). Gaithersburg (MD): National Institute of Standards and Technology; 2016. Available from: http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf

[32] Campagna M, Hardjono T, Pintsov T, et al. Kerberos revisited quantum-safe authentication. ETSI Quantum-Safe-Crypto Workshop. 2013. September 26–27; Sophia Antipolis, France.

[33] Bennett CH, Brassard G. Quantum cryptography: Public key distribution and coin tossing. In: IEEE International Conference on Computers, Systems and Signal Processing; New York; 1984.

[34] Bennett CH, Bessette F, Brassard G, et al. Experimental quantum cryptography. J Cryptology. 1992;5,(1):3–28. doi:10.1007/BF00191318.

[35] Woodward A. Quantum cryptography at the end of your road. *Scientific American*. 2012. Guest Blog. Available from: http://blogs.scientificamerican.com/guest-blog/quantum-cryptography-at-the-end-of-your-road/.

[36] Woodward A. Privacy through uncertainty: quantum encryption. *Scientific American*. 2012. Guest Blog. Available from: http://blogs.scientificamerican.com/guest-blog/privacy-through-uncertainty-quantum-encryption/.

[37] Ekert E. Quantum cryptography based on Bell's theorem. Phys Rev Lett. 1991;67:661–663. doi:10.1103/PhysRevLett.67.661.

[38] Alagic G, Broadbent A, Fefferman B, et al. Computational security of quantum encryption. 2016. Cryptology ePrint Archive, (Report 2016/424).

[39] Korzh B, Ci C, Wen Lim R, et al. Provably secure and practical quantum key distribution over 307 km of optical fibre. Nat Photonics. 2015;9:163–168. doi:10.1038/nphoton.2014.327.

[40] Epping M, Kampermann H, Bruß D. Large-scale quantum networks based on graphs. New J Phys. 2016;18:053036. DOI:10.1088/1367-2630/18/5/053036.

[41] Amir R, Abidin A, Wallden P, et al. Unconditionally secure signatures. 2016. Cryptology ePrint Archive, (Report 2016/739).

[42] Buchmann JA, Williams HC. A key-exchange system based on imaginary quadratic fields. J Cryptology. 1988;1(2):107–118. DOI:10.1007/BF02351719.

[43] Hoffstein J, Pipher J, Silverman JH. NTRU: A ring-based public key cryptosystem. Lect Notes Comput Sci. 2006;1423:267–288.

[44] Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problem. In: Advances in Cryptology - CRYPT0 '97; 1997.

[45] Ajtai M, Dwork C. A pubfic-key cryptosystem with worst- case/average-case equivalence. In: 29th A GM Symposium on Theory of Computing, p. 284-293; 1997. Available from: http://dl.acm.org/citation.cfm?id=258604

[46] McEliece R. A public-key cryptosystem based on algebraic coding theory. 1978. The Deep Space Network Progress (Report, DSN PR 42-44) p. 114–116. Available from: http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF

[47] Perlner RA, Cooper DA. Quantum resistant public key cryptography: a survey. In: Proceedings of the 8th Symposium on Identity and Trust on the Internet; 2009. Available from: http://dl.acm.org/citation.cfm?id=1527028&CFID=669534288&CFTOKEN=94893828

[48] Nguyen P, Stern J. Cryptanalysis of the Ajtai-Dwork cryptosystem. In: CRYPTO '98; 1998. p. 223–242. Available from: http://link.springer.com/chapter/10.1007/BFb0055731

[49] Nguyen P. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem. In: CRYPTO 99; Berlin; 1999. Available from: http://link.springer.com/chapter/10.1007/3-540-48405-1_18

[50] Ducas L, Durmus A, Lepoint T, et al. Lattice signatures and bimodal gaussians. 2013. Cryptology ePrint Archive, (Report 2013/383).

[51] Howgrave-Graham N. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. Adv Cryptol – CRYPTO 2007 Lect Notes Comput Sci. 2007;4622:150–169.

[52] Commission EU. Post-quantum cryptography for long-term security. 2015. *Horizon 2020: PQCRYPTO*, vol. Project reference: 645622.

[53] Augot D. Initial recommendations of long-term secure post-quantum system. 2015. PQCRYPTO Project number: Horizon 2020 ICT-645622. Available from: https://pqcrypto.eu.org/docs/initial-recommendations.pdf

[54] Stehlé D, Steinfeld R. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. 2013. Cryptology ePrint Archive, (Report 2013/004).

[55] Hoffstein J, Pipher J, Silverman JH. Public key cryptosystem method and apparatus. US Patent 6,081,597. 1997. US Patent Office.

[56] Ding J, Xie X, Lin X. A simple provably secure key exchange scheme based on the learning with errors problem. 2012. Cryptology ePrint Archive, (Report 2012/688).

[57] Denchev VS, Boixo S, Isakov SV, et al. What is the computational value of finite range tunneling. 2016. arXiv:1512.02206.

[58] Campbel, P, Groves M, Shepherd D. Soliloquy: a cautionary tale. Cheltrenham: 2014. Available from: https://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf.

[59] Peikert C. What does GCHQ's "cautionary tale" mean for lattice cryptography? Available from: https://web.eecs.umich.edu/~cpeikert/soliloquy.html.

[60] Lyubashevsky V. Lattice-based identification schemes secure under active attacks. In: Lecture Notes on Computer Science 4939; Springer; 2008. p. 162–179. Available from: http://link.springer.com/chapter/10.1007%2F978-3-540-78440-1_10

[61] Lyubashevsky V, Peikert C, Rege O. On ideal lattices and learning with errors over ring. Eurocrypt. 2010. Available from: http://link.springer.com/chapter/10.1007/978-3-642-13190-5_1

[62] Bos a JW, Costello C, Naehri M. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. 2014. Cryptology ePrint Archive, (Report 2014/599).

[63] Alkim E, Ducas L, Pöppelmann T, et al. Post-quantum key exchange – a new hope. 2015. Cryptology ePrint Archive, (Report 2015/1092).

[64] Kirchner P, Fouque P-A. Comparison between subfield and straightforward attacks on NTRU. 2016. Cryptology ePrint Archive, (Report 2016/717).

[65] Ding J, Schmidt D. Rainbow, a new multivariable polynomial signature scheme. Lect Notes Comput Sci. 2005;3531:164–175.

[66] Bulygin S, Petzoldt A, Buchmann J. Towards provable security of the unbalanced oil and vinegar signature scheme under direct attacks. In Progress in Cryptology – INDOCRYPT 2010; 2010. Available from: http://indocrypt2010.uwaterloo.ca/

[67] Pereira GC, Et A. Shorter hash-based signatures. J Syst Software. 2016;116:95–100. DOI:10.1016/j.jss.2015.07.007.

[68] Chabaud, Stern. The cryptographic security of the syndrome decoding problem for rank distance codes. Lect Notes Comput Sci Adv Cryptol ASIACRYPT '96. 2005;1163:368–381.

[69] Blaum M, Farrell PG, van Tilborg HCA. Information, coding and mathematics. New York: Springer; 2002. Available from: http://www.springer.com/gb/book/9781402070792

[70] Barg A, Zhou S. A quantum decoding algorithm of the simplex code. In: 36th Allerton Conference on Communication, Control and Computing; Monticello; 1998. Available from: http://allerton.csl.illinois.edu/

[71] Bernstein DJ. Grover vs McEliece. Lect Notes Comput Sci Post Quantum Cryotograp. 2010;6061:73–80.

[72] Galbraith SD. Constructing isogenies between elliptic curves over finite fields. LMS J Comput Math. 1999;2:118–138. DOI:10.1112/S1461157000000097.

[73] Jao D, De Feo L. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: PQCrypto; 2011. Available from: http://troll.iis.sinica.edu.tw/pqc11/

[74] Delfs C, Galbraith SD. Computing isogenies between supersingular elliptic curves over F_p. 2013. arXiv:1310.7789.

Appendices

Appendix 1. Shor's Algorithm Explained

In 1760, Euler showed that if n is the product of two primes p & q then for $x^1 \bmod n, x^2 \bmod n$, $x^3 \bmod n$, $x^4 \bmod n$, … (provided x is not divisible by p *or* q) the sequence repeats with some period that evenly divides $(p-1)(q-1)$, that is, $F(a) = x^a \bmod n$ is a periodic function with period, r where x is an integer coprime to n, and a is an integer $< n$.

We know from number theory that

$$x^r \equiv 1 \bmod n, \tag{1}$$

$$\text{so } \left(x^{r/2}\right)^2 = x^r \equiv 1 \bmod n \tag{2}$$

$$\therefore \left(x^{r/2}\right)^2 - 1 \equiv 0 \bmod n \tag{3}$$

If r is an even number then $\left(x^{r/2}-1\right)\left(x^{r/2}+1\right) \equiv 0 \bmod n$

that is, $\left(x^{r/2}-1\right)$ $\left(x^{r/2}+1\right)$is an integer multiple of n

If $x^{r/2} \neq 1$, then at least one of $\left(x^{r/2}-1\right)$ or $\left(x^{r/2}+1\right)$ must have a nontrivial factor in common with n.

Thus by computing gcd($\left(x^{r/2}-1\right)$, n) and gcd($\left(x^{r/2}+1\right)$, n), we will obtain a factor of n.

So the algorithm for factoring is:

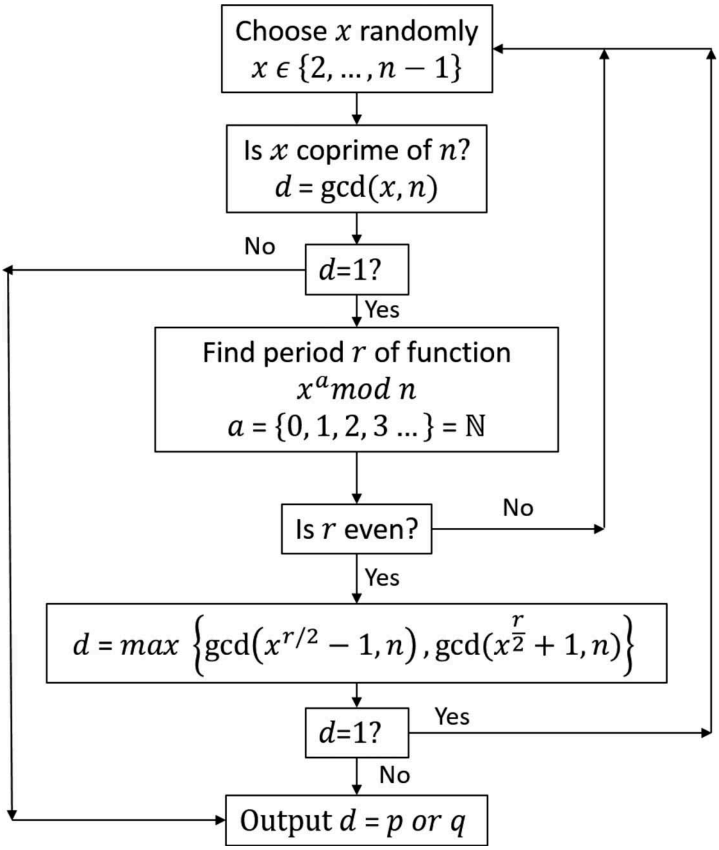
Choose x randomly
$x \in \{2, \dots, n-1\}$
Is x coprime of n?
$d = \gcd(x, n)$
No
d=1?
Yes
Find period r of function
$x^a mod\ n$
$a = \{0, 1, 2, 3 \dots\} = \mathbb{N}$
Is r even?
No
Yes
$d = max\ \left\{\gcd\left(x^{r/2}-1, n\right), \gcd(x^{\frac{r}{2}}+1, n)\right\}$
d=1?
Yes
No
Output $d = p\ or\ q$

Figure 1. Factoring algorithm.

For a small composite this algorithm can be computed manually to see how it operates. For example, the trivial case of factoring 15 would be as follows using the above:

(1) Pick a coprime – for 15 the obvious number is 11. Now raise 11 to increasing powers (1, 2, 3 …) modulo 15, i.e. compute $x^a \equiv 1 \bmod n$:
   (a) Divide 11 by 15 to get 0 with a remainder of 11
   (b) Divide 121 by 15 to get 8 with a remainder of 1
   (c) Divide 1331 by 15, to get 88 with a remainder of 11
(2) Proceed this way, raising 11 to higher powers: the remainder when we divide by 15 alternates between 11 or 1
(3) Hence, the period of 11 with respect to being divided by 15 is 2
(4) Now, raise 11 to the power given by its period, which is 2, the answer is 121 and now take the square root to get 11, i.e. compute $x^{r/2}$
(5) Add 1 to 11 and subtract 1 from 11 to get a pair of numbers: 10 and 12, i.e. compute $\left(x^{r/2} - 1\right)$ and $\left(x^{r/2} + 1\right)$
(6) Find the greatest common denominator of 10 and 15 and 12 and 15. The former is 5 and the latter is 3, which we know are the prime numbers that compose 15.

The problem is that as n increases it becomes increasingly difficult to determine the period $r$of the function $F(a)$. Whilst the Fourier transform runs in conventional computers using highly efficient fast Fourier transform implementations, it is still impractical to run this algorithm on a conventional computer when n exceeds relatively small numbers.

Shor devised a quantum computing algorithm for determining the period. This is the only part of this algorithm that requires a quantum computer, and it is because of this one stage that we obtain the quantum speed up over a conventional computer.

At the heart of Shor's quantum period finding method is the quantum Phase Estimation Algorithm. This relies upon the fact that for a unitary operator U with eigenvector $|u\rangle$ with eigenvalue $e^{2\pi i\varphi}$ you can determine φ.

To do this two quantum registers are created. The first register has t qubits in initial state $|0\rangle$. The second register begins in state $|u\rangle$

The first step is to apply a Hadamard transform to first register and the operator U to the second register, with U raised to successive powers of 2.
The first register becomes

$$\frac{1}{2^{t/2}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right), \tag{4}$$

which equals

$$\frac{1}{2^{t/2}}\sum_{k=0}^{2^t-1} e^{2\pi i\varphi k}|k\rangle. \tag{5}$$

If we apply the inverse quantum Fourier transform and read out the state of the first register, we find a value $\approx \varphi$.

As a quantum gate circuit the Phase Estimation Algorithm could be considered as shown in Figure 2.

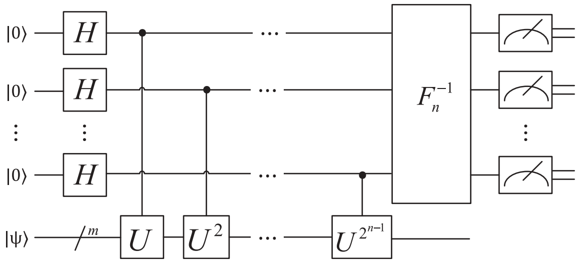

Figure 2. A possible quantum circuit for phase estimation algorithm.

In Shor's algorithm, we begin by creating initial conditions in a two register quantum computer:

(1) $t = 2L + 1 + \log\left(2 + \frac{1}{\epsilon}\right)$ qubits initialised to $|0\rangle$ as register 1,
(2) L qubits initialised to $|1\rangle$ as register 2.

Next we put the registers into superposition, which creates the following state:

$$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|1\rangle. \tag{6}$$

We then employ a unitary function such that

$$U_{x,n} : |j\rangle|k\rangle \rightarrow |j\rangle|x^j k \bmod n\rangle, \tag{7}$$

where x is co-prime to the L-bit number n. This creates the state

$$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle|x^j \bmod n\rangle, \tag{8}$$

which approximates to

$$\approx \frac{1}{\sqrt{r2^t}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi isj/r} |j\rangle|u_s\rangle. \tag{9}$$

Which means that by applying the inverse Fourier transform, we effectively create this state:

$$\rightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\widetilde{s/r}\rangle|u_s\rangle. \tag{10}$$

By measuring the values in first register we will obtain:

$$\rightarrow \widetilde{s/r} \tag{11}$$

where s is equally likely to be any of the possible values of s. However, by using the continued fractions algorithm you can determine r.

The remainder of the algorithm can now be determined using a conventional computer.

Appendix 2. Continued Fractions Algorithm

The Phase Estimation Algorithm produces only $\varphi \approx s/r$, which we know to $2L + 1$ bits. However, we also know that it is a rational number as it is the ratio of two bounded integers. By computing the nearest fraction to φ we can obtain r.

Every possible rational number ξ can be written as an expression of the form:

$$\xi = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\cdots + \cfrac{1}{a_N}}}}}, \tag{12}$$

where $a_{0,} a_{1,\cdots,} a_N$ is a non-negative integer and $a_N > 1$. For simplicity we write this as $[a_0, a_1, \cdots, a_N]$. Assuming ξ is rational then the following recurrence relation always terminates:

$$\begin{cases} a_0 = \xi \\ \xi_0 = \xi - a_0 \end{cases}, \text{and if } \xi_n \neq 0, \text{then} \begin{cases} a_{n+1} = {}^1\!/_{\xi_n} \\ \xi_{n+1} = \frac{1}{\xi_n} - a_{n+1} \end{cases} \tag{13}$$

The n-th *convergent* $(0 \leq n \leq N)$ of this continued fraction is defined as the rational number ξ_n given by $[a_0, a_1, \cdots, a_n]$.

Each convergent can be written in the form $\xi_n = \frac{s_n}{r_n}$, where $\gcd(s_n, r_n) = 1$, and s_n, r_n are determined by the recurrence relation:

$$s_0 = a_0, s_1 = a_1 a_0 + 1, s_n = a_n s_{n-1} + s_{n-2},$$

$$r_0 = 1, r_1 = a_1, r_n = a_n r_{n-1} + r_{n-2}. \tag{14}$$

Suppose s/r is a rational number such that

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2r^2}. \tag{15}$$

Then s/r is a convergent of the continued fraction for φ. Thus it can be computed in $O(L^3)$ using the continued fraction algorithm.

Since φ is an approximation of s/r accurate to $2L+1$ bits, it follows that

$$\left| \frac{s}{r} - \varphi \right| \leq 2^{-2L-1} \leq \frac{1}{2r^2} \text{ because } r \leq n \leq 2^L. \tag{16}$$

Hence, we can be sure that the continued fractions algorithm produces s' and r' with no common factors such that $\frac{s'}{r'} = \frac{s}{r}$. We can check if r' is the correct order by ensuring $x^{r'} \bmod n = 1$.

Appendix 3. Hidden Subgroup Problem

If a group G is finite and Abelian, we can use the quantum Fourier transform to create a superposition over the group elements to which we can also apply a quantum black box function f to give

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle. \tag{17}$$

We can rewrite $|f(g)\rangle$ using the Fourier basis:

$$|f(g)\rangle = \frac{1}{\sqrt{|G|}} \sum_{t=0}^{|G|-1} e^{2\pi i l g/|G|} |\hat{f}(l)\rangle \tag{18}$$

where $e^{-2\pi i l g/|G|}$ represents $g \in G$ indexed by l. However, this can be simplified if f is constant and distinct on cosets of K (a subgroup of G):

$$|\hat{f}(l)\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} e^{-2\pi i l g/|G|} |f(g)\rangle, \tag{19}$$

which has nearly zero amplitude for all values of l other than where

$$\sum_{h \in K} e^{-2\pi i l h/|G|} = |K|. \tag{20}$$

By determining l we can determine the elements of K, which being Abelian enables the generation of a set for the whole hidden subgroup. This does rely upon using the continued fraction expansion algorithm, which is possible only where in determining l from $l/|G|$ there is a high probability that there are no common factors.

However, finite Abelian groups are isomorphic to a product of cyclic groups of prime order power, i.e.

$$G = \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \cdots \times \mathbb{Z}_{p_M}, \tag{21}$$

where p_i are primes and $\mathbb{Z}_{pi}$ is the group over integers $\{0, 1, \cdots, p_i - 1\}$ with addition modulo p_i. Thus we can express the phase in Equation (18) as

Table A1. The components used in the hidden subgroup problem to generate three important quantum algorithms.

Algorithm	G	K	X	f
Period-finding	$\mathbb{Z}, +$	$\{0, r, 2r, \cdots\} \in G$	Any finite set	$f(x + r) = f(x)$
Order-finding	$\mathbb{Z}, +$	$\{0, r, 2r, \cdots\} r \in G$	$\{a^j\} j \in \mathbb{Z}_r \; a^r = 1$	$f(x) = a^x f(x + r) = f(x)$
Discrete logarithm	$\mathbb{Z}_r \times \mathbb{Z}_r + (\text{mod } r)$	$(l, -ls) l, s \in \mathbb{Z}_r$	$\{a^j\} j \in \mathbb{Z}_r \; a^r = 1$	$f(x_1, x_2) = a^{kx_1 + x_2} f(x_1 + l, x_2 - ls) = f(x_1, x_2)$

$$e^{2\pi i l g/|G|} = \prod_{i=1}^{M} e^{2\pi i l'_i g_i/p_i}, \tag{22}$$

where $g_i \in \mathbb{Z}_p$. The phase estimation algorithm now gives us l'_i from which we can determine l.

By using the following for G, K along with a defined set X and function f, we see that the Hidden Subgroup Problem is the basis of the algorithms indicated, which are used as the basis for the quantum attacks on RSA, elliptic curve cryptography and DSA:

The importance of the Abelian Hidden Subset Group as a standard model is further illustrated by noting that the quantum attack on the Buchmann–Williams scheme mentioned in Section 3 is also a solution to the hidden subgroup problem: where G is the reals $\mathbb{R}$.