Tufts University

Computer Systems Security

Computer Science 116

# Quantum Computing: The Risk to Existing Encryption Methods

*Author:*
Zach Kirsch

*Mentor:*
Ming Chow

December 15, 2015

**Abstract**

The basis of modern security relies on encryption methods that are practically impossible, but theoretically possible, to break. The most common is the Rivest-Shamir-Adleman (RSA) cryptosystem, which takes advantage of the concept that it is far easier to multiply two very large primes together than it is to factor the product. It has been regarded as a sound encryption method because breaking it would require factoring that is beyond the capabilities of current computing limits and existing mathematical methods.

Unfortunately, a new technology poses a threat to that assertion: quantum computing. A quantum computer relies on quantum superposition to perform multiple calculations in parallel by creating different states of bit patterns that exist simultaneously. Because of this, a quantum computer can factor a 300 digit number in the same amount of time that an ordinary computer could multiply the factor together, rendering our current encryption methods obsolete.

This paper discusses the threat of quantum computing to RSA and other popular encryption methods, and proposes safe alternatives that can be useful in the post-quantum era.

# Contents

# 1 Introduction

Encryption is at the heart of modern society. Nearly every electronic interaction requires safeguarding information, from securing an email password to protecting missile launch codes. Internet searches, financial transactions, and even democratic elections rely on encryption for security and confidentiality.[1] The importance of encryption cannot be understated, and potential threats to it must be taken seriously.

Quantum computing is one such threat. The processing power alone of quantum computers is an incredible achievement, reaching speeds eight orders of magnitude faster than classic computers and thousands of times faster than modern supercomputers.[2] While there is a endless list of positive and innovative applications for quantum computing, there is a concern that its sheer power could be used for more a malicious intent. Many existing security mechanisms and encryption methods are thought to be secure because a brute-force attack is time prohibitive. However, with quantum computers and their computational speediness on the horizon, it is time to rethink what it means to be time prohibitive, and to develop new encryption algorithms that are resistant to the capabilities of quantum computing.

# 2 To the Community

Our society is fundamentally reliant on encryption. In a post-globalization era, every society that has an open socket to the world requires data to be secure in some way. Without the reassurance that bank accounts are secure, that privacy still exists, that our votes for President are properly cast, it is questionable how much of our society would remain the same. It is not infeasible that the absence, or extreme weakening, of encryption could lead to higher levels of anarchy than the first world is accustomed to. While this paper is not arguing that quantum computers will erase encryption or even that the threat is imminent, the importance of encryption requires that we thoroughly examine new technologies that could alter the security of existing infrastructure.

Despite that fact that quantum computing technology is not yet capable of such attacks, inaction is a massive risk. It is important to investigate the threat for three reasons. First, there is a time lag between when a cryptosystem is proven broken and when it becomes patched or replaced. A classic example is the Heartbleed vulnerability (CVE-2014-0160) in OpenSSL where 300,000

[1] Hayam K. Al-Anie, Mohammad A. Alia, and Adnan A. Hnaif. "E-Voting Protocol based on Public-Key Cryptography". In: *Al Zaytoonah University of Jordan, Amman, Jordan* (July 2011).

[2] Jordan Novet. *Google says its quantum computer is more than 100 million times faster than a regular computer chip*. Dec. 2015. URL: `http://venturebeat.com/2015/12/08/google-says-its-quantum-computer-is-more-than-100-million-times-faster-than-a-regular-computer-chip/`.

servers were still vulnerable two months after its revelation..[3] If a breach is expected or possible, it is preferable to transition away from the vulnerable technology before it is compromised. Second, it would be unknown which data was secure. As will be discussed in this paper, most quantum-based attacks could be interceptions, meaning that it is possible that a message between two legitimate parties could be read without alerting either one. This would cause massive uncertainty for all encrypted messages, because there would be no way to guarantee that data is transferred without interception. Third, when the capable quantum technology arrives, the algorithms for breaking current cryptographic standards will have already been developed. Mathematicians are already devising quantum algorithms for breaching some of the world's most used encryption methods, most notably the Rivest-Shamir-Adleman cryptosystem (RSA) and elliptic curve cryptography (ECC).

Because of the necessity to be prepared for the evolution of quantum computing, this paper serves to participate in the existing discussion and to encourage the community to do so as well. While this threat is not imminent, it is a high magnitude risk that the security industry in particular must be prepared for.

# 3 Public-Key Cryptography

Modern cryptographic methods are often based on mathematical functions that are difficult to invert (trapdoor functions). A prime example is RSA encryption. With existing mathematical methods, it is incredibly simple to multiply large primes together.[4]. However, it is considerably more difficult to factor a semiprime (the product of two primes) into its prime factors. It would take a supercomputer weeks or months to factor a semiprime with over 100 digits.[5] Public-key cryptography is usually based on the difficulty of semiprime factoring, where a public key is generated from the semiprime and the private key is generated from the factors. The only way to generate the private key is to factor the semiprime described by the public key. In contrast, symmetric cryptography involves one secret and easily-invertable key that is shared between the parties. The key is used to encrypt the message, and once delivered, the key's inverse is used to decrypt it.

There is a large benefit to public-key cryptography over symmetric cryptography. In both cases, the private key is invaluable and compromising the private key is equivalent to comprising any encrypted data. With public-key cryptography, if Alice has a private key, she alone is responsible for its security. Bob can send Alice message with her published public key that Alice can decrypt with her private key. Comparatively, with symmetric cryptography, Alice must share her key with Bob. With the private key stored in two separate locations,

[3] Dante D'Orazio. *Over 300,000 servers remain vulnerable to Heartbleed after initial wave of patches.* June 2014. URL: `http://www.theverge.com/2014/6/22/5831732/over-300000-servers-vulnerable-to-heartbleed-two-months-later`.

[4] In secure systems, "large" primes are generally hundreds of digits long.

[5] Toni Smith. *Quantum Cryptography.* May 2004. URL: `http://www.math.ucsd.edu/~crypto/Projects/ToniSmith/crypto.html`.

the probability that it is compromised if effectively doubled. Additionally, if Alice wishes to communicate with multiple people using symmetric encryption, she must either create a separate key for each individual, or use the same key for each partner. The latter possibility is especially problematic, because it increases the likelihood that the key is compromised and precludes Alice from communicating with individuals she does not completely trust. Public-key cryptography eliminated these concerns by establishing a single public encryption key that anybody can access, and a private key that only Alice knows. This superiority is why public-key cryptography is more widely used in applications than symmetric cryptography.

To briefly summarize how semiprime numbers (and their difficult factoring) is incorporated in public-key cryptography, consider the following example. Alice chooses two large primes, $p$ and $q$, and multiples them together to create semiprime $N$.[6] She then calculates $\phi(N)$ (which for a semiprime $N = pq$ is $(p-1) \times (q-1)$).[7] Alice then chooses an integer $e$ and a corresponding integer $d$ such that $e \times d = 1 \mod \phi(N)$. $N$ and $e$ are published as the public key, and $d$ is kept as Alice's private key. To send a message $M$ to Alice, Bob encrypts the message by computing $M^e \mod N$. Once Alice receives the encrypted message $M^e$, she can use her private key $d$ to recover the original message by calculating $(M^e)^d \equiv M^{ed} \equiv M^1 \equiv M \mod \phi(N)$. If the encrypted data is intercepted, the best known method for decryption is to calculate $d$, which requires factoring $N$. As long as $p$ and $q$ are chosen large enough, N will not be easily factorable and the encrypted message will remain secure.[8] Because of the high confidence in the security of public-key cryptography and RSA in particular, it is one of the most heavily used forms of encryption.

# 4 Quantum Computing

The current methods for breaking RSA are not very effective. One method is to factor the number $N$ described by the public key. However, with the magnitude of the primes chosen, factoring takes exorbitant and infeasible time with current methods and technologies (factoring time grows exponentially with input length in bits).[9] The second method is to guess a message, encrypting it with the public key, and checking if it matches the encrypted data. Not only is guessing the message incredibly improbable, but most secure algorithms append random bits to the end of a message as a salt to prevent this sort of attack.[10]

[6] This is done with the Rabin-Miller primality test – a mathematical algorithm for determining with high probability whether a number is prime.

[7] $\phi(x)$ is the Euler totient: the count of numbers less than or equal to $x$ that do not share any prime factors with $x$

[8] Barry Steyn. *How RSA Works With Examples*. May 2012. URL: `http://doctrina.org/How-RSA-Works-With-Examples.html`.

[9] Nikos Drakos. *Shor's Algorithm for Quantum Factorization*. July 2002. URL: `http://tph.tuwien.ac.at/~oemer/doc/quprog/node18.html`.

[10] Paul Fahn. *Answer s To Frequently Asked Questions About Today's Cryptography*. Tech. rep. 100 Marine Parkway, Redwood City, CA 94065: RSA Laboratories, Sept. 1992, p. 7.

In the present day, RSA (when used correctly) cannot be broken. However, while it is secure it practice, it is theoretically vulnerable. If a fast algorithm of semiprime factoring was discovered (it has not been proven that an efficient classical factoring algorithm does not exist), or computing power was rapidly increased, the security of RSA encryption would become questionable.[11] Unfortunately, the latter possibility is all too real with the exploration of quantum computing.

As is apparent in the name, quantum computing has its roots in quantum mechanics. Unlike previous technological advancements where results and efficiency could be easily measured, a quantum computer is incredibly difficult to understand, and even harder to build; despite some companies claiming they have created quantum computers, there is still disagreement over whether quantum computing is even possible.[12]

The purpose of this paper is not to delve deeply into the physics of quantum computers as it is considerably complicated and unnecessary for discussing the security implications of their existence. However, it is helpful to offer a brief explanation to show the relevance of quantum computing in modern cryptography. A core concept of quantum mechanics is superposition – the idea that a particle can exist in multiple states simultaneously (in a way that seems mutually exclusive), but collapses into a single state when it is inspected. Quantum computers take advantage of this principle by creating a superposition of problems, and because each problem (state) exists simultaneously, the computer can solve each problem simultaneously. While a classic silicon-based computer can solve one problem at a time (or a few on a multi-core computer), a quantum computer is much more efficient. Current computers use bits to represent states, and bits are binary (either 0 or 1). Comparatively, quantum computers use qubits to represent states, and it is these qubits that exist in superposition – effectively both 0 and 1 simultaneously. In essence, a quantum computer allows the equivalent of parallelization, but on a level far greater than existing supercomputers.[13]

A one-qubit quantum computer, with the qubit in superposition between two states (0 and 1), could effectively perform two operations at once. A two-qubit quantum computer could represent four states at once (00, 01, 10, 11), and thus could effectively perform four operations at once. In the general case, an $n$-qubit quantum computer could represent $2^n$ simultaneous operations.[14] Google's 1000-qubit processor could support $2^{1000} \approx 10^{301}$ operations at once. While other factors prevent this full potential from being realized, Google's quantum

[11] Drakos, *Shor's Algorithm for Quantum Factorization*, op. cit.

[12] Richard Chirgwin. *Boffins say D-Wave machine could be a classic.* Feb. 2014. URL: `http://www.theregister.co.uk/2014/02/04/boffins_say_dwave_machine_could_be_a_classic/`.

[13] Smith, *Quantum Cryptography*, op. cit.

[14] Ciara Byrne. *The Golden Age of Quantum Computing is Upon Us (Once We Solve These Tiny Problems).* May 2015. URL: `http://www.fastcompany.com/3045708/big-tiny-problems-for-quantum-computing`.

computer is about $10^8$ times faster than the a classical computer simulating the same algorithms.[15]

# 5 Vulnerable Encryption Methods

## Public-Key Cryptography

While there have not recently been tremendous improvements in number theory, the processing power of quantum computers reduces the existing time barrier to semiprime factoring.

Much like with supercomputers, algorithms for quantum computers have to be devised differently than they are for classical computers. The primary algorithm for factoring with quantum computers is Shor's Algorithm, devised originally by mathematician Peter Shor. While the running time of classical factoring algorithms increase exponentially with input length,[16] Shor's Algorithm is considerably more efficient. The time complexity of Shor's algorithm for factoring an $n$-bit integer is: $O((\log n)^2 \times (\log\log n) \times (\log\log\log n))$.[17] The algorithm is based on quantum Fourier transforms and modular exponentiation via repeated squarings.[18]

Shor's algorithm is built to efficiently factor distinct odd primes. Thus, the smallest factorable number under Shor's algorithm is 15 (the product of 3 and 5). In 2001, the algorithm was tested and successfully factored 15 using 7 qubits.[19] In 2012, 21 was successfully factored with Shor's Algorithm.[20] While the factored numbers are clearly nowhere near the magnitude of RSA numbers, the success of the algorithm on small inputs shows promise that larger quantum computers will be capable of factoring larger ones. The current recommendation of a 2048-bit RSA number would require 4096 qubits to break.[21]

The danger to RSA is quite clear. The existence of fast factoring algorithms will completely invalidate any data encrypted under RSA. When encrypted traffic is intercepted, the public key could be determined by examining the destination of the traffic. Using Shor's algorithm or an equivalent, the private key could be derived from the public key. Recall that for a public key $N$ and $e$, the private key is $d$ such that $e \times d = 1 \mod \phi(N)$. When N is factored into primes

[15] Novet, *Google says its quantum computer is more than 100 million times faster than a regular computer chip*, op. cit.

[16] Drakos, *Shor's Algorithm for Quantum Factorization*, op. cit.

[17] David Beckman et al. "Efficient networks for quantum factoring". In: *Phys. Rev. A* 54 (2 Aug. 1996), pp. 1034–1063. DOI: 10.1103/PhysRevA.54.1034. URL: http://link.aps.org/doi/10.1103/PhysRevA.54.1034.

[18] An in-depth explanation and classical implementation of Shor's algorithm is available at http://tph.tuwien.ac.at/~oemer/doc/quprog/node18.html

[19] Vandersypen LM et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." In: *Nature* 414 (Dec. 2001), pp. 883–887.

[20] Enrique Martín-López et al. "Experimental realization of Shor's quantum factoring algorithm using qubit recycling". In: *Nature Photonics* 6 (Feb. 2012), 773–776.

[21] John Proos and Christof Zalka. "Shor's discrete logarithm quantum algorithm for elliptic curves". In: *Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario* (Feb. 2008).

$p$ and $q$, $\phi(N)$ is easily determined to be $(p-1) \times (q-1)$, and as a result, the private key $d$ can be calculated trivially. Once quantum computers are stable and large enough, a quantum computer will therefore be able to factor an RSA semiprime in the same amount of time that a classical computer can multiply the prime factors together.[22]

With quantum algorithms like Shor's, it will be trivial to derive a private key from a published public key; publishing the public key would be equivalent to posting the private key as well. Additionally, because data can be intercepted while in transit and then decrypted, it is possible that the message could be read without alerting either the sender or the receiver. Not only would all data encrypted with this method be vulnerable, but no message could be guaranteed to be secure, effectively destroying the purpose of the encryption.

## Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a oft recommended form of cryptography that encrypts data using maps within elliptic curves.[23] Unfortunately for its proponents, ECC is under the same threat from quantum computing that RSA is. A modification of Shor's algorithm could solve the discrete logarithm problems behind ECC and decipher data that is ECC-encrypted.[24] In fact, because smaller keys are needed for an elliptic curve cryptography system than for an equivalent RSA system, smaller quantum computers could break ECC before they could break RSA. This contradicts the common sentiment that ECC should be the cryptography of the future. It would require between 1300 and 1600 qubits to break 224-bit ECC, which is equivalently secure to 2048-bit RSA (which requires 4096 qubits to break).[25]

With a classical processor, an ECC cipher with $n$ bits in the key takes $2^{n/2}$ steps to crack. With a quantum computer, the complexity is constant and does not grow with key length. Once they are large enough, a quantum computer could break any ECC cipher almost instantaneously. Like with RSA, decrypting ECC-encrypted data with a quantum computer would require no more time than the classical encryption process.[26]

## Symmetric Cryptography

Fortunately, the aforementioned superiority of public-key cryptography over symmetric cryptography does not apply to the threat from quantum comput-

[22] Smith, *Quantum Cryptography*, op. cit.

[23] Nick Sullivan. *ECDSA: The digital signature algorithm of a better internet*. Mar. 2014. URL: `https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/`.

[24] Proos and Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves", op. cit.

[25] Ibid.

[26] Lamont Wood. *The Clock Is Ticking for Encryption*. Mar. 2011. URL: `http://www.computerworld.com/article/2550008/security0/the-clock-is-ticking-for-encryption.html`.

ing. While a quantum computer can derive the private key from a public key, symmetric cryptography has no public key. The only method of attack is brute-force; the quantum computer must generate possible private keys and attempt to decrypt the message.

As an example, a 128-bit AES (symmetric) cipher has $2^{128}$ (about $10^{38}$) possible keys. A classical computer, which generally executes 1 trillion instructions per second, would take about 10.79 quadrillion years to test every possibility. Conversely, for an $n$-bit cipher, a quantum computer operates on the order of $2^{n/2}$. For a 128-bit cipher, this is $2^{64}$ (about $(10^{19})$ steps and it would take about 6 months to test every possibility.[27]

# 6 Risk Assessment

There are a number of barriers that prevent quantum computers to becoming fully developed in the status quo:

**Accuracy** A quantum computer is a probabilistic machine, which means that in a single trial it might return the correct solution along with 10,000 other possibilities.[28] Higher accuracy can be done with numerous trials of the same problem, but this diminishes the speed advantage of quantum computing.[29]

**Environmental factors** Qubits can be altered by heat, noise, stray magnetic couplings. In order to minimize this, the qubits need to be totally isolated and in near-absolute zero temperatures. When doing so, there is still some extraneous noise, but another issue arises: when the qubits are totally isolated, it is difficult to control or examine them without contaminating the environment and contributing additional noise and heat to the system.[30]

**Phase error** In addition to the errors that plague regular bits like bit flip error, qubits are susceptible to other changes in data, like phase error, which can incorrectly flip the superposition sign of the phase relationship and cause errors in measurement.

Despite these hurdles, quantum computers have had some success. Google recently announced that its D-Wave quantum computer was functional and contained over 1,000 qubits.[31] While this is a large development, we are most likely still years, or even decades, away from quantum computers that are capable of

[27] Ibid.

[28] Andrew Tarantola. *The Quantum D-Wave 2 Is 3,600 Times Faster than a Super Computer*. Mar. 2014. URL: `http://gizmodo.com/the-quantum-d-wave-2-is-3-600-times-faster-than-a-super-1532199369`.

[29] Byrne, *The Golden Age of Quantum Computing is Upon Us (Once We Solve These Tiny Problems)*, op. cit.

[30] Ibid.

[31] Novet, *Google says its quantum computer is more than 100 million times faster than a regular computer chip*, op. cit.

the aforementioned cryptanalysis. However, this is not a reason to delay the discussion. It is entirely possible that there could be an unexpected breakthrough in quantum computing technology that accelerates its development faster than predicted. Or, even if this is not the case, once quantum computers reaches the necessary threshold, they could be used to retroactively decipher encrypted messages. Therefore, any secret transmissions that do not expire in the near future should not be encrypted with methods that could be broken by quantum computing.[32]

Additionally, as part of risk analysis, it is important to discuss the potential enemies behind quantum-based attacks. Fortunately, while these attacks could cause inordinate damage, the cost of building and maintaining a quantum computer is cost prohibitive to web thieves and other black hat hackers. Who would be behind attacks? Based on who is pursuing research and development currently, large corporations and governments are the best guesses.[33] While these organizations will not be stealing credit cards and bank accounts, it is improbable that their intentions are entirely noble. Empirically, the National Security Agency uses tools at its disposal to eavesdrop in some form on the conversations and interactions of foreigners and U.S. citizens. Large corporations often have perverse incentives to view more of their customers data to better target ads and increase revenue. Inevitably, the largest harm that will emerge from quantum computing is an extreme loss of privacy.

# 7 Defenses

While there are a number of encryption methods that are susceptible to quantum-based attacks, there are some that are built to be resistant. Collectively, these are dubbed *post-quantum cryptography*. The ideal post-quantum cryptographic method is not merely practically secure (e.g. RSA), but theoretically secure against any attacks. The NSA has already announced plans to migrate their cryptographic standards to post-quantum cryptography.[34] The following are examples of methods that have been or can be engineered that are resistant to quantum-based attacks.[35]

**Hash-based public-key signature system** This is proposed as an alternative to RSA and other quantum-vulnerable public-key signature systems. Common cryptographic hashing functions are very difficult to invert, even for quantum computers, and could provide a reasonable method of verifying au-

[32] Alexander V. Sergienko. *Quantum Communications and Cryptography*. CRC Press, 2005.

[33] Google, Microsoft, and the National Security Agency (NSA) are deeply investing in quantum computing research and technology

[34] National Security Agency. *Cryptography Today*. Jan. 2009. URL: `https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml`.

[35] See *Bernstein 2009* for more exhaustive explanations and examples

thenticity.[36] Hashing is already often used to provide signatures for software that adopting it on a larger scale is entirely possible.

**Lattice-based Cryptography** This is a form of public-key cryptography that avoids the weaknesses of RSA. Rather than multiplying primes, this encryption method involves multiplying matrices. Matrix inverses are computationally very difficult to produce and multiplication thus provides a good trapdoor function. This method is relatively unpopular due to its long public keys, but its quantum resistance could make up for this drawback.[37]

**Symmetric Key Cryptography** As previously stated, symmetric keys like the AES cipher must be brute-forced. With long enough key lengths, even quantum computers will not be able to break the ciphers in a reasonable amount of time.

**Quantum Cryptography** A subset of post-quantum cryptography is quantum key distribution and quantum cryptography: cryptography that relies on quantum mechanics much like quantum computing attacks do. The message transfer begins by one party sending a stream of photons to another; the state and characteristics of each proton are used to generate the key. If the photons are examined at any point between the sender and the receiver, the receiver's detector will notice an error rate in the photon values and alert the two parties. If the key is generated correctly, the key is used to encrypt and send the message.[38] Because silent interception is not possible, and the key is completely random, the quantum key is virtually unbreakable and it "is considered the most powerful data encryption scheme ever developed."[39] While there presently are issues with implementation that prevent it from being uncrackable, this form of encryption is ideal because it is theoretically unbreakable and its security does not depend on the state of existing technology.[40]

[36] Daniel J. Bernstein, Johannas Buchmann, and Erik Dahmen. *Post-Quantum Cryptography*. Springer-Verlag Berlin Heidelberg, 2009.

[37] Ibid.

[38] Wood, *The Clock Is Ticking for Encryption*, op. cit.

[39] Los Alamos National Laboratory. *Quantum Cryptography*. URL: `http://www.lanl.gov/science/centers/quantum/cryptography.shtml`.

[40] Adam Mann. *Laws of Physics Say Quantum Cryptographi is Unhackable. It's Not.* June 2013. URL: `http://www.wired.com/2013/06/quantum-cryptography-hack/`.

# 8 Conclusion

Quantum computing is an exciting new technology. It has the potential to perform computation at an unprecedented rate which will have exceptional benefits for society. However, as with every new and powerful technology, we must analyze the security implications involved. Quantum computing poses serious risks to widely-used encryption methods, most notably RSA and ECC. Rather than slow the pace of innovation and stifle growth, the reaction to these concerns should be to migrate our encryption standards to post-quantum cryptography. The goal should be to stop the use of theoretically unsecure encryption methods, such as RSA, and instead use methods that are proven computationally hard to solve.

# References


Al-Anie, Hayam K., Mohammad A. Alia, and Adnan A. Hnaif. "E-Voting Protocol based on Public-Key Cryptography". In: *Al Zaytoonah University of Jordan, Amman, Jordan* (July 2011).

Beckman, David et al. "Efficient networks for quantum factoring". In: *Phys. Rev. A* 54 (2 Aug. 1996), pp. 1034–1063. DOI: 10.1103/PhysRevA.54.1034. URL: http://link.aps.org/doi/10.1103/PhysRevA.54.1034.

Bernstein, Daniel J., Johannas Buchmann, and Erik Dahmen. *Post-Quantum Cryptography*. Springer-Verlag Berlin Heidelberg, 2009.

Byrne, Ciara. *The Golden Age of Quantum Computing is Upon Us (Once We Solve These Tiny Problems)*. May 2015. URL: http://www.fastcompany.com/3045708/big-tiny-problems-for-quantum-computing.

Chirgwin, Richard. *Boffins say D-Wave machine could be a classic*. Feb. 2014. URL: http://www.theregister.co.uk/2014/02/04/boffins_say_dwave_machine_could_be_a_classic/.

D'Orazio, Dante. *Over 300,000 servers remain vulnerable to Heartbleed after initial wave of patches*. June 2014. URL: http://www.theverge.com/2014/6/22/5831732/over-300000-servers-vulnerable-to-heartbleed-two-months-later.

Drakos, Nikos. *Shor's Algorithm for Quantum Factorization*. July 2002. URL: http://tph.tuwien.ac.at/~oemer/doc/quprog/node18.html.

Fahn, Paul. *Answer s To Frequently Asked Questions About Today's Cryptography*. Tech. rep. 100 Marine Parkway, Redwood City, CA 94065: RSA Laboratories, Sept. 1992.

LM, Vandersypen et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." In: *Nature* 414 (Dec. 2001), pp. 883–887.

Los Alamos National Laboratory. *Quantum Cryptography*. URL: http://www.lanl.gov/science/centers/quantum/cryptography.shtml.

Mann, Adam. *Laws of Physics Say Quantum Cryptographi is Unhackable. It's Not*. June 2013. URL: http://www.wired.com/2013/06/quantum-cryptography-hack/.

Martín-López, Enrique et al. "Experimental realization of Shor's quantum factoring algorithm using qubit recycling". In: *Nature Photonics* 6 (Feb. 2012), 773–776.

National Security Agency. *Cryptography Today*. Jan. 2009. URL: https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml.

Novet, Jordan. *Google says its quantum computer is more than 100 million times faster than a regular computer chip*. Dec. 2015. URL: http://venturebeat.com/2015/12/08/google-says-its-quantum-computer-is-more-than-100-million-times-faster-than-a-regular-computer-chip/.

Proos, John and Christof Zalka. "Shor's discrete logarithm quantum algorithm for elliptic curves". In: *Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario* (Feb. 2008).

Sergienko, Alexander V. *Quantum Communications and Cryptography*. CRC Press, 2005.

Smith, Toni. *Quantum Cryptography*. May 2004. URL: http://www.math.ucsd.edu/~crypto/Projects/ToniSmith/crypto.html.

Steyn, Barry. *How RSA Works With Examples*. May 2012. URL: http://doctrina.org/How-RSA-Works-With-Examples.html.

Sullivan, Nick. *ECDSA: The digital signature algorithm of a better internet*. Mar. 2014. URL: https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/.

Tarantola, Andrew. *The Quantum D-Wave 2 Is 3,600 Times Faster than a Super Computer*. Mar. 2014. URL: http://gizmodo.com/the-quantum-d-wave-2-is-3-600-times-faster-than-a-super-1532199369.

Wood, Lamont. *The Clock Is Ticking for Encryption*. Mar. 2011. URL: http://www.computerworld.com/article/2550008/security0/the-clock-is-ticking-for-encryption.html.