Title:

The circumstances under which quantum cryptography would overtake public-key encryption as the preferred method for securely sending data.

International Baccalaureate extended essay in
Computer Science

Name:

Morgan Haywood

Session number:

001071-005

Word count:

Abstract: 238

Essay: 3997

Submitted June 2009

A discussion of the circumstances under which quantum cryptography would overtake public-key cryptography as the preferred method for securely sending data. Public-key encryption is currently used as a way to securely send data without the need to securely distribute keys prior to this. Public-key encryption is based upon one-way functions, and RSA is a specific example of public-key encryption. Quantum cryptography is based upon the nature of observing spinning photons, and BB84 is a specific example of this type of protocol. RSA has security issues with regard to key size in comparison to current computing power and 'quick' factorising algorithms, especially Shor's algorithm, though this is reliant on the development of quantum computers. Public-key encryption is also used for authentication and digital-signatures, which quantum cryptography cannot be. Additionally, public-key encryption, even when used as hybrid encryption, is more convenient than quantum cryptography, which is a type of key-exchange protocol. The environment influences quantum cryptography and sending multiple photons is a security issue. There have been several implementations of quantum cryptography, though it is still being developed. As quantum cryptography can be implemented in such a way as to be absolutely secure, if it were to be developed fully, quantum cryptography may well replace public-key encryption in some areas. If public-key encryption were to be broken, quantum cryptography would perhaps be the only way to securely transfer data.

Contents

Public-key encryption is currently the standard method for securely sending data over networks. Nevertheless to some extent it has security issues, and future technological developments may render it entirely insecure. Quantum cryptography on the other hand is a developing technology that may provide absolute security. This essay will examine the circumstances under which quantum cryptography would overtake public-key encryption as the preferred method to securely send data. Quantum cryptography still requires a lot of development to be practical and even then would be less convenient than public-key encryption, but would address all security concerns; public-key encryption has issues under certain circumstances and could be broken if algorithms or technologies were to be developed. Quantum cryptography is less convenient than public-key encryption, as the former requires interaction between the parties, while the latter does not. Where security outweighs all other concerns, or if public-key encryption were to become entirely unviable, quantum cryptography may overtake public-key encryption as the preferred method to securely send data.

Alice and Bob are traditionally used as the sender and receiver and Eve as the eavesdropper when talking about cryptography (Schneier, 1996; Singh, 1999), and so will be used here. Also important are Trent (a neutral third party) Carol (another participant in Alice and Bob's communications) and Mallory (who unlike Eve will intercept, modify or otherwise maliciously interfere with Alice and Bob's communications). It should be noted that, although they are referred to as people for illustrative purposes, they are more commonly computers performing actions at the users' request.

Using symmetric-key systems, Alice and Bob must share some information, their key, before any message can be sent. This poses the most substantial issue for modern cryptographers; Alice and Bob must have a way of exchanging information securely before any protected messages can be sent (Singh, 1999). This has become known as the key distribution problem (Singh, 1999). There are several notable protocols of this sort; the Data Encryption Standard, DES, which was the USA standard from 1977 to 2000 (Atreya, Hammond, Paine, Starret & Wu, 2002), the Advanced Encryption Standard, AES, which is the current standard (Atreya, Hammond, Paine, Starret & Wu, 2002), and the one-time pad, which is the only symmetric-key algorithm that has been proven absolutely secure (Biggs, 2008; Kaku, 1998; Schneier, 1996).

To overcome the key distribution problem, public-key encryption has become widely used. Examples of public-key encryption methods include Rabin, RSA and ElGamal (Schneier, 1996), however RSA is the most common (Pountain, 2001; Schneier, 1996), and thus will be referred to henceforth, rather than public-key encryption in general; the same points apply to all methods, and talking about specifics is clearer.

RSA works because of two important one-way functions (Katz & Lindell, 2008; Schneier, 1996; Singh, 1999), prime factorisation, that is finding the prime factors of a number (Katz & Lindell, 2008; Singh, 1999), and modular arithmetic, that is arithmetic using only a limited range of values, as is used for telling the time (Singh, 1999). These are mathematical functions which are very easy to solve in one direction, but very difficult to reverse (Katz & Lindell, 2008; Schneier, 1996; Singh, 1999).

Using RSA, Alice first picks two large prime numbers, p and q, which she then multiplies together to give N *(*Kurose & Ross, 2003; Schneier, 1996; Singh, 1999). Alice then chooses a smaller number, e*,* which should be relatively prime to $(p\text{-}1)\times(q\text{-}1)$ (Biggs, 2008; Kurose & Ross, 2003), that is these numbers should have no common factors other than 1. It should be noted that while it is not a requirement that the value that Alice chooses for e is unique, her value of N and thus her combination of values for p and q must be unique (Singh, 1999). From e, Alice can then work out d, which should be found such that $(e\times d)$ (mod $(p\text{-}1)(q\text{-}1)$) = 1, for which she may use the Euclidean algorithm (Biggs, 2008; Katz & Lindell, 2008; Singh, 1999). N and e are now Alice's public key and *d* is Alice's private key, though *p* and *q* must also be kept confidential as they are needed to find *d* (Kurose & Ross, 2003).

Alice then sends her public key to Bob or publishes it in a place where he can find it. This is the important difference between symmetric-key and public-key encryption; while using a symmetric-key method, Alice would be forced to ensure that only Bob knew her key, whereas using public-key encryption means that Alice wants as many people as possible to know her (public)key (Schneier, 1996; Singh, 1999).

To encrypt his message, Bob computes $C=M^e(\text{mod } N)$, where *C* is the cypher text and *M* is the message. Upon receiving the cypher text, Alice can compute $\mathrm{M}=\mathrm{C}^d(\text{mod } N)$ to recover the message (Biggs, 2008; Kurose & Ross, 2003; Schneier, 1996). If Eve intercepts the message she cannot decrypt it using Alice's public-key, as the private-key must be used to decrypt (Chesbro, 2000), and she cannot find Alice's private-key from her public-key due to the one-way functions (Biggs, 2008). Thus Alice and Bob can communicate securely without the need to securely share information prior to this (Singh, 1999). To allow Alice to send Bob a message, Bob must produce his own public- and private-key set.

Unlike public-key encryption, quantum cryptography relies not on mathematical concepts but on quantum mechanics, which deal with individual particles. Light is made up of photons, which can vibrate, or spin, in any direction, but which can be polarised so that all photons are spinning in the same direction (Schneier, 1996). When this is done, all photons that were spinning at a right angle to the angle of polarisation are stopped, and others will at random either be stopped or change direction to the angle of polarisation (Schneier, 1996). So it is possible, by using a second filter, to tell the direction of polarisation in polarised light, so long as one is measuring in the same direction, or at a right angle to this polarisation. Yet, if any other angle is used one can no longer tell what the polarisation was initially, as the polarisation of the light has changed (Schneier, 1996). This property of observing spinning photons is used in quantum cryptography.

While other protocols exist, the most commonly known quantum cryptography protocol is BB84, developed by Bennett and Brassard in 1984 (Barnett & Phoenix, 1996), and thus will be discussed here. Again, the same points apply to all protocols, but this protocol is used for clearer illustration. Under this protocol, Alice uses two schemes for representing the data she is sending (Barnett & Phoenix, 1996), with the spin on the photons denoting the bit. Each scheme represents the bits using two directions at right angles, that is it represents them in one basis (Schneier, 1996; Singh, 1999). For example, a particle spinning up and down could represent 1 and left to right 0, or using another scheme, right diagonal represents 1 and left diagonal 0. Alice sends Bob a series of random bits, or qubits as they are known when bits are sent with quantum techniques, randomly switching between two schemes (Barnett & Phoenix, 1996; Schneier, 1996; Singh, 1999). Upon receiving the qubits, he too randomly switches between the two schemes, independently of Alice (Schneier, 1996). After all qubits have been transmitted, Alice calls Bob on an unsecured line and he tells her which scheme he used for each qubit, and Alice tells Bob which qubits he read using the correct scheme (Schneier, 1996; Singh, 1999), on average half the bits (Schneier, 1996). Note that the actual content of each bit is never discussed. Bob and Alice then discard all bits that Bob did not read correctly. The rest now form a shared secret key, as both will now have the same sequence of bits (Barnett & Phoenix, 1996; Schneier, 1996; Singh, 1999), unless Eve has listened into the conversation.[1]

Eve is in much the same position as Bob. She too must randomly switch between schemes to read the qubits (Barnett & Phoenix, 1996; Schneier, 1996; Singh, 1999), so she, like Bob, correctly reads about half the qubits, and can listen into the phone conversation to find out which bits will

[1] see appendix A

form the final key. Given that Eve is unlikely to read correctly the same qubits that Bob does she will not know the key in its entirety, but she may learn a significant amount of information. Furthermore, Eve's presence on the quantum line has one important effect; as she reads the qubits, she changes them (Schneier, 1996). So, if Eve is reading the qubits, even where Bob used the correct scheme he will not read all the qubits correctly (Barnett & Phoenix, 1996). Alice and Bob can then detect Eve's presence by discussing openly the first few bits of the final key, which will then be discarded (Schneier, 1996; Singh, 1999). If Bob has all these bits correct they can be certain that Eve was not on the line, and thus their key is secure. If this error checking shows discrepancies then they must discard the key and start over, with Alice once again sending random bits to Bob, until they produce a key that Eve has not interfered with (Singh, 1999). Using a more sophisticated method, they can agree on a random sub-set of the bits and perform a parity check on them, then discard an agreed single bit of the sub-set. While this will not show an even number of errors, the process may be repeated with other sub-sets to prevent this possibility and thus find any errors (Mosca, Jozsa, Steane & Ekert, 2001). This way they need to discard fewer bits if the key is found to be secure (Schneier, 1996).

Their shared key can now be used as the key for a symmetric-key algorithm, for example ADS or a one-time pad (Schneier, 1996).

Both of these cryptographic methods have some problems, however. While RSA is widely used today, it is impossible to tell if it is secure. Anyone who has found a way to factorise *N*, which allows them to find *p* and *q* and so *d*, can find the private-key from the public-key.[2] Likewise, if there is any way to obtain *d* from *e* and *N*, that is the public-key, without use of *p* and *q* as when the key was generated, then the public-key is enough information to find the private-key (Kaufman,

[2] see above for explanation of N, p, q, e and d.

Perlman & Speciner, 1995). Based upon current knowledge there is no way to carry out the latter, and so the former is generally the focus of security concerns when using public-key encryption.

To this end, one of the major decisions that Alice must make is the size of *N* (Singh, 1999)*,* as a larger value of *N* is harder to factorise and thus creates more secure encryption. Still, this also slows down the process and produces greater cypher text compared to the size of the message. Any value of *N* may be factorised by simply trying all possible factors, but this is time intensive and relies largely on current computing power in order to be practical. For example, in 1994 a challenge was issued to break a key in the order of 10^{129} (Singh, 1999), approximately a 429-bit key, and this was completed in 1997, by a team of 600 volunteers (Singh, 1999); prior to 2000, various teams of crypto-analysts and mathematicians, using several computers, took 3 months to break a 512-bit key (Chesbro, 2000); in 2001 a 128-bit key was still considered secure (Pountain, 2001); a more current standard is key lengths up to roughly 4000 bits (Graham-Rowe, 2007). While this value is obviously beyond the means of the average citizen at present, advances in computing power, or having enough computers to throw at a problem, will eventually break Alice and Bob's security. So while Eve, a school friend jealous of Alice and Bob's relationship, will probably never be able to read their love letters until old age, Eve's alternate reality twin, the head of the secret service, may well be able to break through Alice and Bob's security to find out the details of the heist they are plotting in reasonable time.

Given that Alice has chosen a sufficiently large value of *N* that Eve cannot factorise it by exhaustive search in a reasonable amount of time, Eve may still be able to decode Alice's encryption if there is a quicker way to factorise *N.* With the exception of Shor's algorithm (Graham-Rowe, 2007; Kaku, 1998; Mosca, Jozsa, Steane & Ekert, 2001) no one seems to have yet come up with a factorising algorithm to significantly reduce the time required, but conversely no one has yet proven that one does not exist (Beutelspacher, 1991/1994). Thus there is always the possibility that such an algorithm has already been found but not been released publicly (Singh, 1999). If an algorithm were to exist that could factorise a number without requiring a large number of calculations, then RSA could theoretically be broken using current technology up to a key length large enough as to be impractical to use.

Shor's algorithm, the only good factorising algorithm to have been found, has one major drawback, that it can only be run on a quantum computer (Graham-Rowe, 2007). This is because it uses the ability of quantum computers to perform a large number of calculations in parallel (Graham-Rowe, 2007). So while it could reduce the number of calculations required from $10^{\sqrt{n}}$ (the number needed for the current best known factorising algorithm) to n^3, where n is the length of the key in digits (a significant enough reduction to make factorisation practical) (Mosca, Jozsa, Steane & Ekert, 2001),[3] it is reliant upon technology that is not currently advanced enough to run it.

While there are several research groups around the world currently working on building a quantum computer, one capable of running Shor's algorithm with large numbers still seems some years away. In 2007, the most powerful quantum computers to have been produced were developed by two teams independently of each other, from the University of Brisbane, Australia, and the University of Science and Technology of China, in Hefei. Both of these only use four qubits, though they had been used to run Shor's algorithm (Graham-Rowe, 2007). To factorise a number of the scale currently used in RSA it is estimated that 50 trillion qubits would be needed (Graham-Rowe, 2007). Though the two functioning quantum computers prove the concept, a quantum computer of this size is still a long way off (Graham-Rowe, 2007).

Even if a quick method of decoding RSA were to be found, by using Shor's algorithm or otherwise, RSA is not the only public-key encryption method and not all methods rely on the same one-way functions (Graham-Rowe, 2007; Schneier, 1996). For example, Rabin uses square roots

[3] see appendix B

and ElGamal uses discrete logarithms (Schneier, 1996). So, if RSA became insecure these other public-key algorithms would be just as secure as they are today as there are currently no fast algorithms that may be used against them, on a quantum computer or otherwise (Graham-Rowe, 2007).

Nevertheless, if a quantum computer were to be built, the computing power needed to break one-way functions would be available and so other public-key encryption methods would become more vulnerable. Additionally, many other encryption algorithms, for example DES, rely on the difficulty of factorisation for their security (Chesbro, 2000).

Both public-key encryption and quantum cryptography are vulnerable to the man-in-the-middle attack. In the case of public-key encryption, this is because Alice cannot verify that Bob's public-key is, in fact, Bob's (Schneier, 1996). Mallory may well intercept Bob's message when Bob sends Alice his public key, and replace the key with his own. Alice then thinks that Mallory's public-key is Bob's, and then Mallory can read all the messages that Alice sends to Bob, and anything signed by Mallory, Alice thinks came from Bob (Schneier, 1996). In the case of quantum cryptography, this is because Alice cannot be certain that the person she is sending qubits to is Bob. Public-key encryption has ways around this issue of authentication; certification authorities are a common method (Schneier, 1996). They are a neutral third party which everyone on the network trusts: Trent. Trent signs Bob's public-key and some personal information about Bob. When Alice looks up Bob's public-key, she can check Trent's signature and the personal information to make sure she has the correct key. Alternatively, distributed key management can be used, whereby Alice and Bob might meet, and she sign his key. If Carol trusts Alice, she can then check Alice's signature and be sure that it is, in fact, Bob's key. This removes the need for a single person everyone on the network trusts, but means that if Carol doesn't know Alice or anyone else Bob has had sign his key, she can't trust Bob's key (Schneier, 1996).

Along with encrypting messages, the second main use of public-key encryption is to verify that a message came from a certain person, known as digital signatures (Schneier, 1996). This relies on the fact that either key may be used for encryption, on the proviso that the other key is used for decryption. Not all public-key encryption methods possess this property, although RSA, among others, does (Schneier, 1996). Alice first encrypts her reply with her private-key, then with Bob's public-key (Schneier, 1996), which all know. When he receives her message, once he has decrypted it using his private-key, Bob can then further decrypt the message using Alice's public-key (Schneier, 1996). If this second decryption produces meaningful text, Bob knows that it came from Alice, as it must have been encrypted using her private-key and she is the only one who knows this key. The double encryption is necessary, as anyone can decrypt anything encrypted with Alice's private-key, and so signing provides no security (Singh, 1999).

Be that as it may, in the real world public-key encryption is very rarely, if ever, used alone. This is because public-key encryption is far slower than symmetric-key methods, symmetric-key methods being roughly 1000 to 10 000 times faster in hardware and 100 times faster in software (Schneier, 1996). Also, to a much lesser extent, because public-key encryption is vulnerable to chosen-plain text attacks (Schneier, 1996), as Eve knows both the key and method used to encrypt messages. Thus, public-key encryption is generally used in hybrid encryption to securely send keys, which can be used with a symmetric-key system to encrypt the message. The so-called session-key is then discarded. This overcomes the above problems to some extent while still allowing Alice and Bob to communicate securely without having to exchange any information beforehand (Schneier, 1996).

In its most commonly used form, then, RSA is very similar to key exchange. Key exchange, as opposed to actual encryption, is a method to solve the key distribution problem rather than a way to securely send data and is used with a symmetric encryption method. Quantum cryptography is really just a key exchange algorithm rather than an encryption method, as the requirement that about half the bits at random are discarded makes it useless for sending data. Although hybrid encryption is very similar to key exchange it is far more convenient and easy to implement as it does not require Alice and Bob to interact (Atreya, Hammond, Paine, Starret & Wu, 2002). Using hybrid encryption,, Bob can send Alice both the information she needs to read his message and the message at once to use at her leisure, provided that she has published her public-key.

Conversely, using quantum cryptography Alice must also send Bob some information before he can encrypt his message, that is he must know which bits she has read correctly and that the key has not been interfered with before he knows the final key. This makes quantum cryptography far less convenient.

Sending information using spinning protons at all is inherently difficult, as the environment that they are sent through may interfere with them (Kaku, 1998; Singh, 1999), causing Bob to incorrectly read bits that he has used the correct scheme for. This is the quantum version of noise on the line (Mosca, Jozsa, Steane & Ekert, 2001). However, if parity checking is used to detect Eve's presence, then this will also eliminate errors caused by noise, because if errors are found in the parity then the bit Alice and Bob discard may be the incorrect one, and as the parity checking continues Alice and Bob will eventually eliminate these errors. This allows a small chance of a bit being incorrect due to noise to be tolerated (Mosca, Jozsa, Steane & Ekert, 2001). Fibre optic has become the preferred medium for quantum cryptography because of this, as it can carry light with a minimum of outside interference.

When more than one photon is sent at a time, all with the same spin, Eve may read both photons separately, each with a different filter, thus obtaining with certainty the spin that was on them, while Bob reads another photon. As the photon Bob receives has the same spin as the one Alice sent, neither Bob nor Alice will be able to detect Eve's interference (Willis, 2009), and thus their messages will be insecure. Researchers are currently experimenting with using nano diamonds with a single imperfection to overcome this problem of sending multiple photons (Willis, 2009), nonetheless it is still one of the major security issues when implementing quantum cryptography.

Quantum cryptography cannot be used for everything that public-key encryption is used for today. Applications such as authentication and digital signatures are possible because there is a widely-published key that can link the secret one to a specific user, and that can itself be permanently linked to a single user. Quantum cryptography, however, relies on faith that the person one is receiving documents from is who they claim to be, and that a published document was authored by the person from whom it supposedly came. Neither of these things can be verified the way they can be with public-key encryption. This means that quantum cryptography is vulnerable if Mallory impersonates Alice when communicating with Bob and Bob when communicating with Alice, causing both Alice and Bob to believe that they have established a secure key with the other, while Mallory can act as a middle man, passing messages between them, while reading or altering messages at will.

Authorities have always regulated encryption to ensure the law-enforcers can break through it if need be, and there has been heated discussion surrounding the use of so-called ‘strong’ encryption, for example RSA. For instance, DES keys were originally restricted to 56-bits for this purpose (Singh, 1999). Given this, there is likely to be much debate over the use of quantum cryptography, as when used with a one-time pad it is theoretically unbreakable (Singh, 1999).

Public-key encryption is currently widely used. It secures banking, business and e-commerce (Graham-Rowe, 2007) when such data needs to be sent over networks, as well as e-mails and files that users wish to send securely (Atreya, Hammond, Paine, Starret, & Wu, 2002).

Quantum cryptography, though still in the research stage, has been implemented on a number of occasions. The first demonstration was performed by Bennett and Brassard in 1992 over a distance of 30 cm (Barnett & Phoenix, 1996; Schneier, 1996). British telecom achieved 10 km using fibre-optic cable later in the 1990's (Schneier, 1996), and in 1995 the University of Geneva performed the key-exchange over 22.7 km, between Geneva and the nearby town of Nyon, again using fibre-optics (Kaku, 1998; Singh, 1999). In attempting to create a method to communicate with satellites using quantum cryptography, Los Alamos National Laboratory attained a distance of 1 km through the air (Singh, 1999). All of these demonstrate that it is a realistic technology, not just a theoretical concept, though none are yet on a large enough scale to be viable for any sort of practical use. Additionally, repeaters would be needed over distances greater than about 100 km, and as currently used repeaters will not work with single photons networks using present-day technology would be limited to this distance (Salleh, 2005). So for quantum cryptography to ever overtake public-key encryption repeaters would need to be built and the ability to re-transmit using satellite developed (Mosca, Jozsa, Steane & Ekert, 2001).

Quantum cryptography cannot be used for the authentication and signature purposes that make public-key encryption so useful for everyday uses, especially on the internet. Thus quantum cryptography has enough significant disadvantages when compared to public-key encryption that, unless public-key encryption becomes insecure, the former will probably never replace the latter in common use. Nonetheless, should quantum cryptography be developed to the point where it is a viable technology, it would certainly be useful for applications where the data must be absolutely secure, despite the inconvenience of having to verify each others identity, for example military or government applications. On the other hand, if all public-key encryption methods were to be broken, which would most likely only happen if a quantum computer were to be built, then quantum cryptography may well have to become the standard method for securely sending data.

Bibliography

Atreya, M., Hammond, B., Paine, S., Starret, P. & Wu, S. (2002). *Digital signatures*. Berkeley, California: McGraw-Hill.

Barnett, S. M. & Phoenix, S. J. D. (1996, March). The principles of quantum cryptography. *Philosophical transactions: mathematical, physical and engineering sciences, 354,* pp. 793-803. Retrieved October 26, 2008, from the JSTOR database.

Beutelspacher, A. (1994). *Cryptology: an introduction to the art and science of enciphering, encrypting, concealing, hiding and safeguarding described without any arcane skulduggery but not without cunning waggery for the delectation and instruction of the general public.* (J. C. Fisher, Trans.) Washington DC: The Mathematical Association of America. (Original work published 1991).

Biggs, N. L. (2008). *Codes: an introduction to information communication and cryptography.* London: Springer-Verlag.

Chesbro, M. E. (2000). *The complete guide to e-security: protect your privacy on the internet.* New York: Kensington Publishing Corp.

Grahman-Rowe, D. (2007, Sep. 15). Qubits poised to reveal our best-kept secrets. *New Scientist, 2621,* pp. 30-31.

Kaku, M. (1998). *Visions: how science will revolutionize the twenty-first century.* New York: Oxford University Press.

Katz, J. & Lindell, Y. (2008). *Introduction to modern cryptography.* Boca Raton, Florida: Taylor & Francis Group.

Kaufman, C., Perlman, R. & Speciner, M. (1995). *Network security: private communication in a public world.* Englewood cliffs, New Jersey: Prentice Hall PTR.

Kurose, J. F. & Ross, K. W. (2003). *Computer networking: a top-down approach featuring the internet* (2nd ed.). Location unkonown: Addison-Wesley.

Mosca, M., Jozsa, R., Steane, A. & Ekert, A. (2001). Quantum-enhanced information processing. In J. M. T. Thompson (Ed.), *Visions of the future: physics and electronics*. Cambridge: Cambridge University Press.

Pountain, D. (2001). *The new penguin dictionary of computing.* London: Penguin books.

Salleh, A. (2005). *Diamonds are a spy's best friend.* Retrieved April 24, 2009, from ABC television. Web site: http://www.abc.net.au/science/articles/2005/04/29/1356182.htm?site=catalyst&topic=latest

Schneier, B. (1996). *Applied cryptography: protocols, algorithms and source code in C* (2nd ed.). New York: John Wiley & Sons.

Singh, S. (1999). *The code book.* London: Fourth Estate.

Willis, P. (Reporter). (2009, March 19). *Catalyst* [Television broadcast]. Hobart, Tasmania: Australian Broadcasting Corporation.

Bibliography: Appendices

Courtland, R. (2009). *Universe's age ereased from texas school science standards.* Retrieved May 10, 2009, from Short Sharp Science. Web site: http://www.newscientist.com/blogs/shortsharpscience/2009/03/universes-age-erased-from-texa.html

Gonsalves, A. (2007). *IBM bluegene is the world's fastest computer once again.* Retrieved May 10, 2009, from EE Times. Web site: http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=202805721

History of supercomputing timeline text. (n.d.). Retrieved May 10, 2009, from Computer Science and Mathematics. Web site: http://www.csm.ornl.gov/ssi-expo/histext.html

Schneier, B. (1996). *Applied cryptography: protocols, algorithms and source code in C* (2nd ed.). New York: John Wiley & Sons.

Williams, R. (2008). *Mathematics: sandra arithmetic, crypto, microsoft excel.* Retrieved May 10, 2009, from Techgage.Web site: http://techgage.com/article/intel_core_i7_performance_preview/9

Appendix A

Illustration of BB84 protocol

Alice sends (binary)	1	1	1	1	0	1	0	0	0	1	0
Alice uses basis	×	+	×	×	+	+	×	×	+	+	+
Alice sends	∕	｜	∕	∕	—	｜	\	\	—	｜	—
Bob uses basis	×	+	+	+	+	×	×	+	×	+	+
Bob reads	∕	｜	—	｜	—	∕	\	—	∕	｜	—
Shared key	∕	｜			—		\			｜	—
Shared key (binary)	1	1			0		0			1	0

(adapted from Schneier, 1996)

Only the basis that Alice and Bob used are ever discussed, and this is only done after all bits have been sent.

Appendix B

Calculation of relatives speeds of factorisation

In 2007, Blue Gene was the fastest computer, with a performance of 478.2 teraFLOPS (Gonsalves, 2007), that is 478.2×10^{12} or $10^{14.8}$ flops.

Given that $10^{\sqrt{n}}$ calculations are required to factorise an n bit key, $10^{\sqrt{155}}$ or $10^{12.41}$ flops are needed to factorise a 512-bit (or roughly 155-digit) key when using the best known factorising algorithm that can be run on a conventional computer.

Then Blue Gene could do this in $10^{12.41} \div 10^{14.8}$, or $10^{-2.39}$ seconds, which is 4.07×10^{-3} or about 4 milliseconds.

For a 1024-bit (or roughly 308-digit) key, $10^{\sqrt{309}}$ or $10^{17.59}$ FLOPS are required, which would take Blue Gene, $10^{17.59} \div 10^{14.8} = 10^{2.79}$ seconds,

which is $10^{2.79} \div 10^{1.78} = 10^{1.01}$ = about 10 minutes.

A 4096-bit (1234-digit) key, the largest currently used, would require $10^{\sqrt{1234}}$ or $10^{35.13}$ FLOPS and take Blue Gene $10^{35.13} \div 10^{14.8}$, or $10^{20.33}$ seconds,

which is $10^{20.33} \div 10^{1.78} = 10^{18.55}$ minutes,

which is $10^{18.55} \div 10^{1.78} = 10^{16.77}$ hours,

which is $10^{16.77} \div 10^{1.38} = 10^{15.39}$ days,

which is $10^{15.39} \div 10^{2.56} = 10^{12.83}$ years.

By contrast, the universe is about 13.73 billion years old (Courtland, 2009), that is 13.73×10^{9} (taking 'billion' to mean 10^{9}), or $10^{10.14}$ years. This means that Blue Gene would take about 490 times the age of the universe to factorise a 4096-bit key.

On the other hand, if Blue Gene were capable of running Shor's algorithm, for a 512-bit (155-digit) key this would be reduced to 155^{3}, or $10^{6.57}$ flops, which would take $10^{6.57} \div 10^{14.8}$, or $10^{-8.23}$ seconds, that is 5.90×10^{-9} seconds, or 5.90 nanoseconds. For a 1024-bit key it would take 46.8

nanoseconds, and for a 4096-bit key it would take 2.98 microseconds. That is, factorising a 4096-bit key using Shor's algorithm is several times faster than factorising a 512-bit key using the best known algorithm that can currently be run.

Even assuming that a computer capable of running Shor's algorithm could only run at 1 megaFLOPS, which given that more recent PC processors run at 30-70 gigaFLOPS (Williams, 2008), would be a particularly slow speed by today's standards (this being 30000 to 70000 times faster), and one ninth the speed of the fastest computer in 1964 ("History of supercomputing", n.d.), it would still only take about half an hour to factorise a 4096-bit key. ($1234^3 \div 10^6 = 10^{3.27}$ seconds, which is $10^{3.27} \div 10^{1.78} = 10^{1.49} = 31$ minutes).

Then, $10^{12.83}$ years, many times the age of the universe, for the fastest computer (using the fastest known traditional algorithm) compared to half an hour for an exceptionally slow computer (using Shor's algorithm) to factorise a 4096-bit key is a significant reduction in time and allows the necessary calculations to be performed in a reasonable amount of time.