Spring 6-3-2017

Encryption Backdoors: A Discussion of Feasibility, Ethics, and the Future of Cryptography

Jennifer A. Martin
Seattle Pacific University

ENCRYPTION BACKDOORS: A DISCUSSION OF FEASIBILITY, ETHICS, AND THE FUTURE OF CRYPTOGRAPHY

by

JENNIFER A. MARTIN

FACULTY ADVISOR, MIKE TINDALL
SECOND READER, ELAINE WELTZ

A project submitted in partial fulfillment
of the requirements of the University Scholars Program

Seattle Pacific University

2017

Approved _____

Date _____

ABSTRACT

In the age of technological advancement and the digitization of information, privacy seems to be all but an illusion. Encryption is supposed to be the white knight that keeps our information and communications safe from unwanted eyes, but how secure are the encryption algorithms that we use? Do we put too much trust in those that are charged with implementing our everyday encryption systems? This paper addresses the concept of backdoors in encryption: ways that encryption systems can be implemented so that the security can be bypassed by those that know about its existence. Many governments around the world are pushing for these kinds of bypassing mechanisms to exist so that they may utilize them. The paper discusses the ethical implications of these policies as well as how our current encryption algorithms will hold up to future technology such as quantum computers.

ENCRYPTION BACKDOORS: A DISCUSSION OF FEASIBILITY, ETHICS, AND THE FUTURE OF CRYPTOGRAPHY

Introduction to Encryption

Information is one of the most valuable commodities in the world today. Because almost everything can be done "online," we have digitized most of our information. The ability to get information from your primary care physician to a specialist is made easier and more efficient by the digitization of medical records. Controlling finances is made convenient by banking applications, spending trackers, and real-time information about the assets that you own. Shopping is made simple by online commerce websites and transactions. All of these everyday conveniences rely on the digitization of information.

While we value convenience and time, we also, as a society, value privacy and security. The Internet is not inherently secure. Thus, we require another entity or practice to ensure that our information is not available to everyone. *Cryptography* is the field that works to make sure that only the people we want to have access to our information have access to it. More specifically, cryptography is "the science or study of the techniques of secret writing, especially code and cipher systems [6]." Under the umbrella of cryptography is *encryption*. Encryption, in essence, is a code or cipher system and "is a means of maintaining secure data in an insecure environment [17]." In another sense, encryption is the process of disguising data so that it cannot be read,

modified, or fabricated by those that don't have the appropriate permissions. It ensures that if you give Amazon your credit card information, that only Amazon gets that information and not some other random person that also happens to be on their website.

There are innumerable encryption systems out there, but only a fraction of those are used in practice because, like anything else, some are better than others. The properties of trustworthy encryption systems include a basis on sound mathematics, extensive analysis by qualified experts with the conclusion that they are sound, and the ability to stand the "test of time [17]." Cryptographic systems are generally split between symmetric and asymmetric types both of which make use of *keys*. In the context of encryption, keys are values (determined by the given algorithm) that encrypt or decrypt blocks of plaintext [19]. *Symmetric encryption* is based on the principle that one key does both the encryption and decryption. *Asymmetric encryption*, on the other hand, is based on the principle that one key encrypts and a different key decrypts. Among strong symmetric cryptographic systems are *DES* (data encryption standard) and *AES* (advanced encryption standard.) However, symmetric systems require that both parties physically (or securely) meet to exchange keys or else the system is no longer secure. Asymmetric cryptographic systems avoid this problem as they rely on a key pair with one private key and one public key.

Clearly, "keys" are the cornerstone of encryption. Mirroring the concept of how security works for buildings, keys enable access to specific places for the people that have permission to be there (those that have the keys.) Recently, the government has been pushing the idea that it needs access to all information (or in other words: government needs to be the keeper of all the keys) in order to be able to keep the country safe from terrorists and criminals. As a result, "a mandatory key escrow has been proposed, whereby government agencies would keep a sort of 'skeleton key' to all encrypted data [10]." A *skeleton key* in concept is a kind of "master key" that would open any lock. Since most encryption algorithms make this mathematically impossible in practice, it is more helpful to think of the concept as a "*backdoor*." The government wants a way of accessing all encrypted data, a way to get at the data without needing to use the specific keys. Thus, they want to use the "back door." So the question is whether backdoors are feasible to implement in modern encryption algorithms and whether their implementation would be ethical. Encryption is not simply a matter of technology…but "a political, social and policy issue that will become more prominent as global electronic commerce increases and as computer networks reach into more and more homes, businesses and government agencies [10]."

This paper examines the feasibility of implementing a backdoor into a popular encryption algorithm and then discusses what the widespread use of backdoors would mean for the privacy of information and other ethical issues. The findings conclude that

while it is possible to put backdoors into the implementation of encryption algorithms, doing so is problematic in that it is hard to control who can take advantage of the door once it is there. Finally, since the fields of technology and subsequently cryptography are constantly evolving, the feasibility of backdoors may change in the future.

RSA Encryption

First, we introduce and analyze a widely used asymmetric encryption algorithm: RSA. Another term for this is *public-key cryptography* and, "one of the elegant aspects of [these] systems [are] that each one is constructed around a fundamental mathematical idea… [and in the] case of RSA, the essential mathematical relationship comes from prime numbers [4]." As with many mathematical areas of study, "the study of prime numbers had essentially no practical applications [4]" at first. With the development of systems like RSA, "prime numbers are now some of the most crucial mathematical objects from a practical standpoint [4]," thus illustrating the importance of studying seemingly less-applicable branches of mathematics.

According to experts, "RSA is by far the most popular public-key encryption algorithm in use [17]." Named after its inventors Ron Rivest, Adi Shamir, and Len Adelman, RSA encryption is considered the "workhorse of public-key cryptography [24]." A basic scenario where Bob wants to send a message to Allen explains it best. First, Allen uses an RSA "key generation" algorithm to generate two keys (which form a

pair): one public, and one private. The private key should only be accessible to Allen (or else the security of the interaction is lost.) Allen then sends Bob the public key that he generated, and this allows Bob to encrypt a plaintext message with the public key that only Allen can decrypt with his private key. It doesn't matter if others can gain access to the public key because both keys are necessary to be able to both encrypt and decrypt messages. While the situation seems trivial, the mathematics behind the process quickly becomes complex.

The mathematical concepts required to understand the RSA encryption algorithm adequately include abstract algebra, probability (specifically, combinatorics), and number theory. In addition, Euclid's theorem (that there are infinitely many primes) is important to the core concept because, without it, we wouldn't be able to generate enough unique primes for the algorithm to be secure, or even feasible. The private key in the key pair is comprised of two distinct odd prime integers, p, and q. The public key in the key pair is composed of a positive integer n that is the product of p and q and a positive integer e such that the greatest common divisor of the integer e and the product of $(p-1)$ and $(q-1)$ is equal to one. The plaintext message converts to an integer that we'll call c (for *ciphertext*.) The process of decryption is essentially finding an integer m such that the product of m and e are congruent to c modulo n. In other words, finding the e^{th} roots modulo a composite integer n. The conditions imposed on the problem parameters n and e ensure that for each integer c in the range of zero to n-1,

there is exactly one m in the range of 0 to n-1 such that the product of m and e are congruent to c modulo n. Equivalently, the function $f: Z_n \rightarrow Z_n$ defined as $f(m) = me(mod\ n)$ is a permutation [15]. Thus, the public key is comprised of e and n while the private key is composed of d and n. The message (m) is encrypted by sending the ciphertext version (converted to numbers) which is equal to the message raised to the e^{th} power modulo n. The message is decrypted by computing c (for ciphertext) to the d^{th} power modulo n.

As an example, let's encode the message "This is a test!" using small prime values for p and q. First, we must convert the message to numbers using the ASCII table of values. Thus, "This is a test!" becomes "84104105115321051153297321161011511633." Now, using prime values $p=11$ and $q=17$, we compute $N=pq=187$ and $r=(p-1)(q-1)=160$. We have thirty candidates for K (a number equal to 1 (mod 160) that can be factored), but we will use 4641. Obviously, this works as a value of K since $4641\ (mod\ 160) = 1$. Next, we want to factor K into integers e and d such that e and d are relatively prime to 187 and for which $ed = 1\ (mod\ 160)$. We can use $e=21$ and $d=221$ since $4641 = (21)(221)$, e is relatively prime to N (since their greatest common divisor is 1), d is relatively prime to N (their greatest common divisor is also 1), and $ed = 4641\ (mod\ 160) = 1$. Thus, our public key is (21, 187) and our private key is (221, 187). To encode the message using these numbers, we need to take chunks of two numbers at a time (so that the value is less than N) and raise that number to the e^{th} power (mod N). The first part of the message is

84, and we would encode this by computing $(84)^{21}$ (mod 187) = 84 and to decode we would simply compute $(84)^{221}$ (mod 187). In this case, they were the same number. In the next case, however, we get $(10)^{21}$ (mod 187) = 142 and then to decrypt we would take $(142)^{221}$ (mod 187) to get 10 back. This procedure is then followed for every chunk to encrypt the entire message. The final encrypted message is "841429682441852182441851391502116142447044173124." In this small example, it is clear why we choose prime numbers that are much larger than 11 and 17. Since there are only thirty choices for K and only a handful of choices for e and d, this example is very easy for a computer to brute-force guess the key-pair.

While our example describes how one would use the RSA algorithm by hand, in practice, a computer does the computation, and the message sender does not actually intentionally choose a p and q. In fact, "each person in the network uses a computer to randomly generate an appropriately large p, q, and d and then computes n, k, and e [14]." Then e and n are given to the public for encoding while p, q, d, and k are kept secret. "There is no practical way for outsiders to determine d without first finding p and q by factoring n [14]."

An important line of questioning is whether or not the implementation of the RSA algorithm is efficient enough to be useful. The first step is choosing prime numbers p and q, and thus, the computer must have efficient ways of finding prime numbers. Fortunately, there are effective ways of finding large prime numbers such as the

application of Fermat's Little Theorem [4]. The mentioned theorem says that "for any prime number p, and any number a which is not a multiple of p, $a^{p-1} = 1\ (mod\ p)$ [4]." If the stated equation is not true, then p is not prime and another number can be tested. Another potentially costly operation is choosing e to be relatively prime to (p-1)(q-1) and computing the inverse of e modulo (p-1)(q-1). But in actuality, these operations can be done using the extended Euclidean algorithm which "allows us to compute modular inverses extremely rapidly (a runtime proportional to e and (p-1)(q-1)) [4]." Finally, there is the matter of *exponentiation* for the actual encryption and decryption. Exponentiation is typically a very costly operation for a computer since it is usually computed by way of repeated multiplication. One of the reasons that the required exponentiation isn't too expensive for the computer to handle in a timely manner is because it uses a method called *fast modular exponentiation*. This approach enables the computation of $m^e (mod\ n)$ in at most $2log_2 e$ steps [4]. Fast modular exponentiation uses repeated squaring and then multiplies together all of the values of the repeated squaring and is thus much more efficient than regular exponentiation. Therefore, the RSA algorithm is concluded to be efficient enough to be useful in practice: just not necessarily ideal for every context.

While the RSA encryption algorithm is incredibly useful, it isn't well suited for all situations. In particular, "it isn't enough that something is mathematically possible; to be practical we need to be able to do it relatively quickly, ideally in real time [4]." The

problem with RSA encryption is the speed (or lack thereof) and the requirements for using the algorithm. "Messages encrypted by RSA must be numbers, and they must be at most n, where n is the publicly displayed modulus. Unfortunately, if the message is large, this would require lots of encryption [4]." So, every message that we want to send using the RSA encryption algorithm needs to be converted to a number, encrypted, decrypted, and then converted back to plaintext. While it is possible to execute the operations for encrypting with RSA efficiently, “they still take long enough that we don't want to have to perform them thousands or millions of times [4]." It should be apparent from the detailed explanation above that these calculations are costly in terms of computing power and thus, would not be efficient to use for very large messages.

Thus, in practice, RSA is usually used to encrypt and send a *key* so that a more efficient algorithm such as DES or AES can be used for the encryption and decryption of plaintext. The primary uses of the RSA algorithm are then, key exchange, authentication, and signing of digital messages or transactions [17].

The use of the RSA encryption algorithm in producing digital signatures is incredibly vital to the way many people use the internet. Authenticating the identity of someone online is a very challenging task as you can’t just take someone at their word. Business contracts, online shopping transactions, confidential communication, as well as many other transactions require assurance that the person on the other end is who they claim to be. Thus, the introduction of digital signatures in the online world was

vital to its' success. While the standard process of encryption uses the public key to encrypt a message, using the *private key* to encrypt a message provides a useful verification process that the person who encrypted it is who they claim to be. Since the private key is only known by one person, anyone who has access to the public key can verify the first person's identity by decrypting the message that was encrypted using the private key. Thus, the RSA algorithm "provides a very simple solution to these authentication problems [4]." The process described above verifies what is known as a person's "digital signature [3]."

Although the RSA algorithm provides the basis for digital signatures, the process is actually quite a bit more complicated since, in most cases, we want to know more than just the identity of the person. For example, when signing a legal or binding document over the internet, there must be reasonable assurance that the document or message being signed by the person cannot be altered by the receiver of the document (or even a third party.) Thus, the question becomes how can someone "sign" a "reasonably sized number that will nevertheless authenticate the entire message as a genuine agreement by [the signee] [4]?" Because most messages (or documents) that we would want to sign over the internet are much too big for computer-based implementations of RSA, the compromise is to "incorporate information about the message, so we know which message [the person] meant to sign [4]." The security in

document signing can be achieved using hash functions in conjunction with the RSA algorithm.

Hash functions can take a message and return a sample of the message, called a "hash." The "hash" is a sort of "fingerprint" of the message, and without the ability to do this, *digital signatures* would be practically unusable [4]. There are some issues with hash functions because the "fingerprint" of the message isn't necessarily unique. "The situation where two messages have the same hash value is called a *collision*. Collisions are a problem for hash functions because each collision represents multiple messages which the hash function cannot distinguish between [4]." Collisions themselves are an unavoidable problem with hash functions, but secure hash functions simply have collisions that are difficult to find. A popularly used hash function for everyday internet communication is called the SHA-1 (Secure Hash Algorithm 1), and it outputs a 160-bit hash value. "As of 2012, no one knows even a single instance of a collision for the hash function SHA-1 [4]." As an example of how this hash function converts plaintext to a hash value, we take the plaintext "Math is fun!" which gives a hash value of "58652c94b5d73a3caca6a23744bb4a59b0d0f353." It would be hard to imagine another plaintext phrase mapping to such a random value, so it makes sense that there has never been a documented collision. Obviously, this hash algorithm is invaluable to internet communication and subsequently digital signatures, and amazingly, other hash

functions that are believed to collide even less are being put into widespread use as well.

Now that the typical uses of the RSA algorithm have been explained, it's important to understand why it can be trusted to be secure. The level of security of the RSA encryption algorithm is based on the difficulty of prime factorization in a finite field. At the most basic level, the private key is composed of two prime numbers, and the product of these two numbers is the public key. With small enough numbers, RSA encryption wouldn't be very secure because computers could simply brute-force (guess every factor of the public key until it guesses correctly) a solution for the private key. However, RSA encryption requires that the key length is a minimum of 3072 bits [2]. This requirement keeps the method secure because it is impractical to brute-force a solution for that large of a key. The point is hit home with the following fact: "In real-life computer implementations of RSA, the numbers which have to be factored to break the system are hundreds of digits long, and would take even the world's fastest computers thousands or millions of years to factor [4]."

While there is widespread belief that the RSA algorithm is secure, there is no actual concrete proof of its security. In the early 2000's when a math professor at Purdue claimed to have a proof of the Riemann hypothesis, some believed that it threatened the security of the RSA algorithm. The Riemann hypothesis proposed that the distribution of prime numbers could be described by the Riemann Zeta function. Fortunately,

cryptanalysts have always assumed that the Riemann hypothesis was true and thus its proof would have little to no impact on the security of the RSA algorithm in practice [18]. It is possible that there is a way to crack RSA without having to factor n (the RSA problem) but "to date, no one has found a way to circumvent the system by breaking it without factoring integers [4]." Thus, “all we can do is show that it's resistant to a large number of varied attacks and that to date no one has published a successful assault [4]." In fact, cryptanalysts have extensively analyzed the RSA algorithm and have found no serious flaws in the algorithm itself but, rather, possible flaws in the implementation of the algorithm. An example of a flawed implementation is not using a large enough key value or, relatedly, using a flawed method for random number generation. In general, the idea pertains to building flaws in the implementation of the algorithm that are virtually undetectable by users.

Random number generation is one of the main ways that backdoors can be implemented in many encryption algorithms, RSA included. For algorithms like DES and AES, “any appropriately sized string of bits can be a key,” and for RSA and ECC the actual key “consists of the values of a set of parameters and perhaps one or more random numbers [17]." Because good random number generation is one of the main tenants for secure encryption, “a weak random number generator can undo the entire encryption system [25].”

Random numbers generated by computers are not “truly” random. Normally, random number generation is based on taking random constants from "a neutral source such as digits in the definition of π [17]." In any case, for a random number generator to be considered useful, “[it] will produce a new key that an attacker cannot readily infer, even knowing the previous one or even previous thousands of keys [17].” However, if there is a pattern for the source of “random numbers,” then they are no longer random. In the case of the RSA algorithm, if the numbers that the keys are based on are not random, it may be possible to figure out what they were, thus causing the security of the algorithm to fail. As a principle, “without assured randomness, an attacker can predict what the system will generate and undermine the algorithm [25].” The next section will examine the use of a flawed random number generator and how it created a backdoor in a government-promoted encryption scheme.

NSA Backdoor Controversy

One relatively well-known instance of a backdoor in an encryption implementation was discovered in 2007 and confirmed to be a calculated act in 2012. One purpose of the National Security Agency (NSA) is to “lead the U.S. Government in cryptology …in order to gain a decisive advantage for the Nation and our allies under all circumstances [13].” Part of their role in the security community is to make recommendations to the National Institute of Standards and Technology (NIST) “for

securing U.S. government sensitive and unclassified communications [17].” In 2005, the NSA recommended what is known as Suite B (a set of advanced cryptography algorithms) that included a random number generator known as *Dual-EC-DRBG* [17]. The following year, NIST proposed these recommendations as the standard for the United States. This action majorly increased the number of systems that implemented Suite B so that they would be acceptable for interaction/implementation with U.S. government.

On the surface, this seemed like a normal interaction. The Dual-EC-DRBG random number generator was “based on an elliptic curve cryptosystem [17].” In 1985, *elliptic curve cryptography* was pioneered by Victor Miller and Neil Koblitz “as an alternative mechanism for implementing public-key cryptography [17]." The algorithms are based on logarithms in finite fields, and the mathematics behind them are fairly sophisticated.

In essence, the cryptosystem examines elliptic curves and uses their properties as the basis of the algorithm. These curves are composed of (x,y) coordinates of points “that satisfy the equation

$$y^2 = x^3 + ax + b \tag{1}$$

for constants a and b [17]." Due to properties of these curves, we know that any "nonvertical straight line passes through at most three points on the curve and given any two points we can find the third point through which the line passes [17]." Given

points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, and the calculated slope, $s = (y_P - y_Q)/(x_P - x_Q)$, we get the following set of equations that lead to information about the third point, *R*, and its *x* and *y* values.

$$P + Q = R \tag{2}$$

$$x_R = s^2 - x_P - x_Q \tag{3}$$

$$y_R = -y_P + s(x_P - x_R) \tag{4}$$

According to the algorithm, elliptic curve operations are done in a finite group, namely the integers, mod *p* for some prime *p* [17]." Restricting the arithmetic operations to a finite field makes reversing the problem difficult even though we know that both values for *x* and *y* increase without bound in the simple equation. Thus, the problem is to find how many steps (base *P* of the logarithm of *Z*) it takes to get from starting point *P* to the end point *Z* with the constraint that the results are in a finite field [17]. Generating every intermediate point between *P* and *Z* is currently the quickest known way to solve the problem. Since brute-forcing this would take a significant amount of time, the algorithm is believed to be secure.

There are advantages to using the elliptic curve cryptography system over another system such as RSA. One of these advantages is that one can get "equivalent security with [relatively] short key lengths [17]." Another is that the algorithm is open-source and thus costs nothing for someone to implement. In contrast, RSA is a patented the algorithm which means that those that want to use its encryption technique "may be

required to pay a license fee [17]." The ECC algorithm was enticing to the NSA because of “its strong security, efficiency, and scalability over public-key cryptography algorithms [17]" which is why when it was chosen for the NIST standard, there were no immediate alarms raised.

Despite its allure, Dual_EC_DRBG is notably slower than other random number generation approaches by somewhere between “two [and] three orders of magnitude [17]." Even more concerning (although it is not immediately obvious why) is the fact that the generation is dependent on the constants a and b (shown in equation (1)). In the appendix of the NIST publication for Suite B, these constants can be found, but there is no accompanying explanation for why the specific constants were chosen. The year following the NIST recommendations, two Microsoft researchers, Shumow and Ferguson, presented findings about the Dual_EC_DRBG and their concerns about its security [17]. Their presentation illuminated a relationship between the constants in the NIST appendix and a “second, secret set of numbers that can act as a kind of skeleton key [22].” If someone knew about the secret numbers, the output of the random-number generator could be pretty easily predicted. Even more alarming was the fact that the predictions could happen with just 32 bytes (or one TLS internet encryption connection) of its output [22]. Although an uncommon occurrence in cryptography, the backdoor in the algorithm for the Dual_EC_DRBG is not immediately obvious. Thus, the researchers did not present the information as if the security recommendation had gone through

despite this knowledge, but rather came from the standpoint of concern because the backdoor that they discovered “makes Dual_EC_DRBG very fragile [25].” After the flaw had been published by the two researchers, NIST proposed that Dual_EC_DRBG be removed from the U.S. standard for encryption, and although there was general suspicion, no major developments in the story were made until a few years later.

Documents from a former NSA employee came to light in 2012 after he had defected to Russia with a significant amount of classified material. The documents that he released "appear to detail actions of the NSA that weaken the design or implementation of popular cryptography [17]." It became clear from the information released that the backdoor in Dual_EC_DRBG was put there intentionally by the NSA as “part of a $250-million, decade-long covert operation by the agency to weaken and undermine the integrity of a number of encryption systems used by millions of people around the world [25].” Adding to the questionable activities of the NSA is the reported information that the “NSA paid RSA $10 million ostensibly so that Dual_EC would be the default random number generator distributed in RSA's cryptographic toolkit BSAFE [17]." The reason this is such a significant action is that during that time, these cryptographic toolkits were considered to be trustworthy and well-vetted and as a result, a significant number of people used them. If the NSA acted with intent and knew about the backdoor at the time it was recommended to be part of the encryption standard; then it would have made it incredibly easy to cripple the encryption and

observe users unnoticed. Even more alarming is the fact that “even if it’s not the default algorithm for encryption on a system… an intruder, like the NSA, can get into the system and change the registry to make it the default algorithm used for encryption [25].” Thus, there was a legitimate reason for NSA to push its inclusion as part of the encryption standard. So, since there is overwhelming evidence that the NSA acted with intention, the question this begs is whether or not the intentional inclusion of backdoors (especially in encryption standards) is an ethical practice.

First, we will examine what the NSA is tasked with doing and whether they stayed true to their mission statement. In relation to cryptography, the NSA has three primary roles: developing codes and ciphers for the United States military, supporting NIST, and performing what they call signals intelligence missions. These missions “collect, process, and disseminate intelligence information from foreign signals for intelligence and counterintelligence purposes to support military operations [17]." While part of their role is ensuring the security of encryption systems for U.S. citizens, sometimes the signals intelligence missions can run counter to that aspect. In addition, the NSA lists their core values as honesty, integrity, respect for the law, and transparency [13]. Expanding on transparency, they explicitly state that, "we embrace transparency to the fullest extent possible. We never forget that we, too, are Americans and that every activity we engage in is aimed at ensuring the safety, security, and liberty of our fellow citizens [13]." The problem is that by their own standards and

goals, the NSA put the security and liberty of United States citizens at risk by pushing an encryption scheme that had a backdoor. If we look further back in its past, it seems that the NSA has a history of putting the security of information for its U.S. citizens at risk.

One example of a questionable practice (regarding ethics) was in the 1990s when the NSA restricted the export of “most cryptographic hardware and software to products using a 40-bit or shorter key [17]." The issue with using a key that short is that the brute-force attack method of breaking encryption becomes feasible. Many thought that the reasoning behind this restriction was to make it easier for the agency itself to hack into encrypted information as a part of its signals intelligence mission. However, by knowingly implementing such a practice, the agency was also subjecting its citizens to outside attack as well.

Another example that brought into question the NSA’s practices and motives was their support of a cryptographic chip called Clipper. The point of the chip was to implement strong cryptography that also allowed for “key recovery.” However, the key recovery was only possible by the government because the chip backed up “a copy of every encryption key used [and] stored [them] with two government agencies [17].” The point was that each government agency would get half the key and they would able to retrieve the keys if needed with a court order. From these examples, it is made clear that the NSA has a history of promoting insecure practices for its citizens so that

their signals intelligence mission is made easier when it comes to accessing information from United States citizens.

This discussion shows that clearly backdoors in algorithm implementation are feasible and have been put into practice numerous times. The question now is really whether or not these practices are ethical. The evidence generally points to these practices as not being ethical. While it is important for government agencies to be doing their jobs and protecting the country, supporting and promoting flawed encryption standards puts United States citizens at more risk than it is worth. Information is incredibly valuable, and the NSA has no control over who discovers the backdoors that they utilize (including hackers or outside government agencies.)

The Role of Tech Companies and Government Agencies

A recent court case has been the subject of much scrutiny and interest in relation to backdoors in technology and provides a good segue into the role that technology companies play in regards to backdoors in encryption. The Federal Bureau of Investigation (FBI) recently won a court case against technology giant, Apple, requiring that they essentially create a backdoor to the security on an iPhone. There are actually quite a few pieces to this discussion, and a good start is with the way Apple handles the security on their iPhone products.

More recent versions of the iPhone operating system (iOS 8 and above) use encryption based around a *key-pair* derived from the user's passcode as well as "a [unique] *hardware key* embedded in the device." In order to unlock the phone, "the passcode must be combined with the *device key* in a chip on the phone, and that chip rate limits passcode attempts to make a brute force attack slower [16]." Since Apple doesn't have the user's passcode and newer devices have the hardware key embedded in the device with means that make it unrecoverable, they can't actually hack into the phones that they produce. While this is a selling point for their phones and a positive for the consumer, government agencies like the FBI want a way into the phones and they want technology giants such as Apple to build them.

Reflecting on the case (*Apple v. FBI*, 2016), the FBI specifically wanted to be able to access the data on the 2015 San Bernardino shooter's iPhone 5c as they believed it would give them further insight into the crime committed. Since iPhone passcodes are composed of options for either a four-digit pin or a six-digit pin, it would be possible to hack into the phone in a matter of somewhere between hours and weeks- but a finite amount of time nonetheless. The problem is that "by default, the phone would have wiped its contents after 10 failed login attempts, preventing the FBI from simply guessing the PIN via a brute-force approach [8]." Thus, the agency turned to the courts and won a case that technology giants are trying to overturn for reasons that will be examined later. Specifically, the "FBI wants a new version of the operating system

designed to allow the FBI to brute force attack the phone [16],” and they want Apple to put it on the shooter’s phone. Because "Apple's public key is built into every iPhone, [it also has] a secret key that it uses to sign software updates [8].” So, the flawed operating system would need to have Apple’s “signature” for the iPhone to recognize it as legitimate and not wipe the phone’s contents.

Even more recently, the attack on London in 2017 in which a man “drove a car into pedestrians, killing three of them, and then fatally stabbed a police officer [23]" added to the debate on cracking encryption for terrorist investigations. The London attacker used WhatsApp (a secure messaging app owned by Facebook), and lawmakers in Britain want the encryption broken to aid in their investigation. "After several terrorist attacks in Europe and elsewhere, the region's lawmakers and regulators want Silicon Valley companies to do more to tackle potential threats [23].” The country has already pushed legislation through that gives “law enforcement greater powers to make telecommunications and technology companies hand over digital information relating to intelligence operations [as well as] bypass encryption protocols, where possible, to aid investigations [23]." The British home secretary reiterated to the technology companies in the case of the London attacker, "they have a responsibility to engage with the government [and] to engage with law enforcement agencies when there is a terrorist situation [23].” However, there are huge implications for this kind of “backdoor.”

One of the major flaws with the FBI's argument is their suggestion that "this tool could only be used once, on one phone." However, once the backdoor is created, "the technique could be used over and over again, on any number of devices [9]." When that kind of technique is out there at all, there are chances for exploitation by a number of parties- and not just the government because "any circumvention technique developed by Apple or another firm proves that such circumvention is possible, which leads to clever outside parties using the bare information revealed to develop their own [9]." In addition, once the precedent is set that a company "can be forced to build one skeleton key, the inevitable flood of similar requests—from governments at all levels, foreign and domestic—could effectively force Apple and its peers to develop internal departments dedicated to building spyware for governments, just as many already have full-time compliance teams dedicated to dealing with ordinary search warrants [21]." Furthermore, once these types of requests for hacking their own encryption increases, "the tools will leak [and] the techniques will be exploited by criminals [causing] our collective security [to] decline [16]."

Recognizing the potential for abuse as well as the ethical concerns, technology giants such as Google, Amazon, Facebook, and Microsoft filed a large number of court briefs in support of Apple in relation to this case. With good cause, "industry leaders worry the government is trying to create a permanent 'back door' that would allow investigators or spies to circumvent encryption or password protection [12]."

Effectively, this would create significant ethical issues within companies as well as undermine the privacy of the consumers.

Many computing professionals are a part of the Association for Computing Machinery which has a dedicated code of ethics for computing practices. One of the general moral imperatives that they list is to respect the privacy of others. Specifically, “it is the responsibility of professionals to maintain the privacy and integrity of data describing individuals. This includes taking precautions to ensure the accuracy of data, as well as protecting it from unauthorized access or accidental disclosure to inappropriate individuals [1].” Thus, if professionals working for these large technology companies are forced to allow unauthorized (at least by the user) access to their information, that is a breach of the code of ethics.

In addition, companies that already work hard to secure their products in the first place would also have to work to “undermine that security—and the better it does at the first job, the larger the headaches it creates for itself in doing the second [21].” Obviously, the stakes are not as low as the government and its agencies present. These companies have made their positions clear in that "such efforts would infringe on human rights because providing the authorities with access to such messaging services would require weakening their overall levels of encryption [and] would leave people who use their services vulnerable to outsiders [23]." The case is not simply about “whether the federal government can read one dead terrorism suspect's phone, but

whether technology companies can be conscripted to undermine global trust in our computing devices [21]." Once the precedent is set, and the backdoor or special firmware is created and out there, everyone's data security is at risk just as in the case of the NSA discussion.

Developments are being made every day in technology and science fields, and while it is pretty clear from the previous discussion of ethics that implementing algorithms with backdoors is generally not a great idea, one wonders if backdoors will be an issue for future encryption algorithms.

Quantum Cryptography: The Future of Encryption?

Quantum cryptography is what some consider to be the future of encryption. In the 1980s, Stephen Wiesner was the first to explore the possibility of this technique, and it was later refined by Charles Bennett [17]. While most encryption schemes use mathematics as their basis, quantum cryptography uses physics- specifically the behavior of light particles [17]. Similar to how the RSA algorithm is used in practice, *quantum encryption* would be most useful for transmitting keys that would be used in the encryption of the actual message. For this reason, many refer to it as "*quantum key distribution* [11]" or "QKD" instead. In essence, this encryption technique uses *photons* coupled with the Heisenberg uncertainty principle to secure communication using an agreed upon technique for converting the polarized photons to a more readable version

(such as bits.) After the key is transmitted securely, normal methods of coding and encoding are used to communicate [4].

As stated above, the basis of QKD is light particles (also known as photons.) As these light particles travel through space they are "vibrating in all directions, " but we simplify the matter by saying that "they have the directional orientation of their primary vibration [17]." Because of this, we count four different directional orientations that are used. These are denoted with the symbols -, |, /, and \ and are calculated "by rounding each actual orientation to the nearest 90 degrees [17]." These light particles are then polarized using one of two different kinds of polarizing filters (denoted either + or x.) The filter "is a device or procedure that accepts any photons as input but produces only certain kinds of photons as output [17]." As an example, a "+ filter" can differentiate between | and – photons (with a fifty-percent chance of incorrectly counting a / or \ photon instead [17].)

In order to physically create these photons, "quantum cryptographers use light emitting diodes as a source of unpolarized light [4]" that allows them to produce the necessary single photon at one time. Then by using polarizing filters, the photon can be forced to take on a specific state or be polarized (the superposition of 1 and 0) [4]. As the photon passes through the filter, "information is attached to [its] spin [4]," and by assigning binary code to each photon sent, a message can be formed that only the receiver will be able to discern.

The most interesting aspect of this type of encryption is that "once a photon is polarized, it can't be accurately measured again, except by a filter like the one that initially produced its current spin [4]." This is due to the Heisenberg uncertainty principle which states that the speed and location of a particle at a given time cannot both be known since they cannot both be measured at the same time. Thus, “when we measure any property of a particle, it affects other properties [17].” In practice in interpreting these messages, the sender and receiver do not need to use the same filters for measuring the photons; the receiver just needs to know whether the filter that they used matched the filter that the sender used in order to correctly interpret the message. If they both used the same filter, then the recorded information should match what the sender intended. On the other hand, if the receiver used the opposite filter, they can use that information to correctly interpret what the orientation of the photon was supposed to be. The result should be a record of photon orientations that is interpreted with binary code to reveal the message (or more typically, the key.)

Just like any other encryption scheme, there are strengths and weaknesses associated with QKD. The major strength is that it is the “first cryptographic scheme to safeguard against passive interception [4].” Because of the Heisenberg uncertainty principle, it is fairly easy to tell whether or not a third party has measured the photon because by measuring it at all, the orientation is altered [4]. Thus, when the sender and receiver are verifying the filters used, if they find that the receiver used the correct filter

but got a bad result, there was interference. Another positive is that there is no need to encrypt the conversation about filter usage between the sender and receiver. Since the measurements aren't actually communicated but the type of filter used is instead, "a third party listening in on the conversation can't determine the actual photon sequence [4]" from the conversation.

Unfortunately, QKD has downsides as well. More than "twice the bits transmitted are not used in [the actual] cryptography [17]" which means that it isn't optimal in terms of efficiency. In addition, "single photon detection is hard, and the technology to process more than a few photons is expected to be developed in the future [4]" so to put this scheme into wider practice, there would need to be more developments made. Finally, there is the severe technical limitation of distance. Because "a photon's spin can be changed when it bounces off other particles, it may no longer be polarized the way it was originally intended to be [upon arrival] [4]." So, the actual distance that protons can be transmitted with a reasonable degree of accuracy isn't terribly far [4]. New developments are being made for a "scalable architecture that includes a Trusted Node to bridge the gap between successive quantum key distribution systems [11]," and it would allow the sharing of keys over a far-reaching network. Effectively, this architecture would make a larger scale implementation not only possible but practical.

While not currently in widespread use, there are working implementations of the quantum cryptographic scheme. The U.S. National Institute for Standards and Technology has one working over a distance of "one kilometer over glass fiber at a rate of four megabits per second [17]." There is even a joint network between BBC Communications and Harvard University that has "six servers and covers a distance of ten kilometers, but reliable communications up to twenty kilometers have been achieved [17]." Like so many other things, QKD is solid in theory, but technical difficulties in its implementation stage need to be worked out before it is more widely adopted. Despite this, many experts believe that this method will be the "best technically feasible means of generating secure encryption [11]" once it can be more widely integrated.

Although this encryption technique relies on physical processes, it can be implemented in such a way that allows for a backdoor. The widespread belief about the improved security of QKD is based on "assumed behavior of implemented equipment," and furthermore, it is assumed that "as long as [any] deviations [from modeled behavior] are properly characterized and proofs updated, [that] implementations are unconditionally secure [20]." Recently, a team of researchers examined the consequences of laser damage on two different practical quantum communication systems and found that not to be the case. In both systems, they found that "laser damage altered the characteristics of security critical components in such a way that

resulted in a compromise of security [20]." Just as in the case of Dual_EC_DRBG, any flaws in the implementation of an encryption algorithm can be utilized in the breaking of its security. The researching team found this to be the case and concluded that "any alteration of system characteristics might compromise the security either directly by leading to an attack or indirectly by shifting some parameter in the security proof so it would no longer apply [20]." While difficult to envision (since it is not widely implemented), laser damage is a real possibility for a backdoor into QKD that would weaken its security.

Quantum Computers

Quantum computing is an active area of research which threatens the security of our modern encryption algorithms and any algorithms that depend on computers not being fast enough to break algorithms with the brute-force method. As discussed earlier, a cornerstone of the security of algorithms currently in use is the generation of random numbers and the difficulty of factoring to get back to these original numbers after computations have been done to them. The reason that current computers cannot brute-force these factorizations is due to their slow speed. In practice, the number of combinations that must be tried to brute-force guess a key is phenomenally large. In the United States, typically "128-bit keys are used and are virtually impossible to crack by brute force methods using current computing technologies [because it] would take a

conventional PC about 1,000,000,000,000,000,000,000 years to crack [10]. With the development of quantum computers, however, "scientists could slice through sophisticated encryption schemes, model quantum systems with unprecedented accuracy, and filter through complex, unstructured databases with unparalleled efficiency [7]." In other words, the efficiency and power of quantum computers would make brute-force factorization for large fields viable in a much smaller amount of time. Regarding the RSA algorithm that we discussed earlier, since there already exist "extremely fast factorization algorithms built for quantum computers, all that remains is to actually build these machines [4]." Unfortunately, according to leading experts in the field, basically, all of public-key cryptography is vulnerable to the possibility of quantum cryptography.

Analogous to the bit for classical computers, quantum computers work on *qubits*. Qubit stands for "quantum bit" and can exist normally as a 0 or 1 or as both 0 and 1 simultaneously [7]. These computers could use quantum concepts such as superposition of states or quantum entanglement which would "enable it to perform many calculations simultaneously [8]." While classical computers work on the concept of either 1s or 0s, superposition of states includes 1 and 0 at the same time. In practice, rather than "considering one solution to a problem at a time, you can consider multiple possible solutions simultaneously [7]." As Matthias Steffen (the manager of the experimental quantum computing research team at IBM) puts it, "you start with a sea of

all possible answers in your quantum states, and you design your algorithm to peel away the wrong answers so that the right answer emerges [7]."

While the sheer power of these computing systems is daunting, there are quite a few difficulties in the process of their implementation. Quantum computing is based around the qubit as previously mentioned but there are many different ways researchers are going their actual building. A few of these include artificial atoms, tapping electron spin, and trapping ions [7]. Some of these building schemes have severe physical limitations. For example, qubits require "working in extremely low temperatures, often bordering on absolute zero [7]" which is an environment that is not only hard to create but to maintain as well. There is also the issue of the ability to measure quantum states. Coherence time is defined as the "amount of time the quantum system is available to be read by the computer before the quantum state collapses [7]" and is measured in microseconds due to how quickly this happens. In order to measure quantum states at all requires a "mastery of quantum correlation or entanglement [7]" so that the systems can be measured without accidentally destroying them in the process. Finally, there is "an intrinsic margin of error in quantum computation in general [7]" which makes it a requirement for the systems to regularly rid themselves of.

Taking into account all of these difficulties, researchers are starting small. Most are pouring all of their resources and brainpower into "developing a single, stable

qubit--and eventually strings of tens, then hundreds, and then thousands and tens of thousands of qubits [7].” Thus, the overall goal for the moment is to figure out a way to not only stabilize qubits…but also “to program them to find and fix any errors [5].” Recently, “researchers from UCSB and Google have figured out how to stabilize an array of nine qubits [5].” While this is an exciting development and a step toward implementing a working quantum computer, there is still a long way to go. The experiment with the nine-qubit array can at present only protect against errors showing up in regular computers, but they gained valuable insight into the stability of groups of qubits that “yield hope for a successful quantum computer someday [5].”

No matter where we are in developing a quantum computer, to many experts, it seems the implementation is inevitable. Thus, it will be important for the continued security of information that new encryption algorithms emerge that cannot be made obsolete by the emergence of quantum computing power.

Conclusion

Commonly used RSA encryption is an example of an encryption scheme that is incredibly useful in the online realm because it not only allows users to verify their identity, but also allows the secure exchanging of keys so that a more efficient encryption algorithm can be utilized. However, upon closer examination of the algorithm, it is clear that backdoor implementation is feasible and is put in the

implementation of the algorithm rather than the actual algorithm itself. With flawed random number generation among other techniques, backdoors can be implemented without the knowledge of the algorithm user. Although it is clear that we can, this begs the question of whether or not we "should." Many important cases are coming to light about the privacy of information versus what the government wants to be able to access. Finally, there is the question of what quantum computers will mean for the security of modern encryption algorithms in the future as well as being able to implement backdoors.

The field of technology and its security is constantly evolving, and the two main things that we must always keep in mind are how will we plan for the future security of our valuable information and the thought that just because something is possible-doesn't necessarily mean we should do it.

BIBLIOGRAPHY

[1] ACM. ACM Code of Ethics and Professional Conduct. https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct

[2] Boxcryptor. AES and RSA Encryption https://www.boxcryptor.com/en/encryption

[3] T. Bell, H. Newton. Coding-Encryption http://csfieldguide.org.nz/en/chapters/coding-encryption.html

[4] M. Cozzens, S. Miller, *The Mathematics of Encryption: An Elementary Introduction*, American Mathematical Society, 2013.

[5] K. Dickerson. Google just hit a milestone in the development of quantum computers. http://www.businessinsider.com/googles-quantum-computing-milestone-2015-3

[6] Dictionary.com "cryptography," in *Dictionary.com Unabridged*. Source location: Random House, Inc. http://www.dictionary.com/browse/cryptography

[7] C. Dillow. How It Would Work: Creating A Quantum Computer. http://www.popsci.com/science/article/2012-04/unleashing-unparalleled-power-quantum-computer

[8] M. Eddy. Crypto-Wars: Why the Fight to Encrypt Rages On http://www.pcmag.com/article/348329/crypto-wars-why-the-fight-to-encrypt-rages-on

[9] G. Fleishman. Skeleton keys open doors for all governments, not just ours. http://www.macworld.com/article/3034319/legal/skeleton-keys-open-doors-for-all-governments-not-just-ours.html

[10] K. Gheamian. Simple Concept, Complex Technology. http://www.governing.com/templates/gov_print_article?id=100552974

[11] D. Hayford. The Future of Security: Zeroing in on Un-hackable Data with Quantum Key Distribution. https://www.wired.com/insights/2014/09/quantum-key-distribution/

[12] A. Khamooshi. Breaking Down Apple's iPhone Fight With the U.S. Government. https://www.nytimes.com/interactive/2016/03/03/technology/apple-iphone-fbi-fight-explained.html?_r=0

[13] NSA. Mission & Strategy. https://www.nsa.gov/about/mission-strategy/

[14] T. Hungerford, *Abstract Algebra: An Introduction*. Third Edition. Cengage Learning, Boston, MA, 2014.

[15] A. Menezes, P.C. van Oorschot, S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[16] R. Mogull. Why the FBI's request to Apple will affect civil rights for a generation. http://www.macworld.com/article/3034355/ios/why-the-fbis-request-to-apple-will-affect-civil-rights-for-a-generation.html

[17] C. P. Pfleeger, S.L. Pfleeger, J. Margulies, *Security in Computing*. Fifth edition. Prentice Hall Press, Upper Saddle River, NJ, USA, 2015.

[18] J. Quain. Encryption Skeleton Key? http://web.a.ebscohost.com.ezproxy.spu.edu/ehost/pdfviewer/pdfviewer?sid=97604351-9af3-408e-a74a-2363463e9ca6%40sessionmgr4008&vid=1&hid=4101

[19] M. Rouse. Key http://searchsecurity.techtarget.com/definition/key (updated September 2005)

[20] S. Sajeed, S. Kaiser, P. Chaiwongkot, M. Gagne, J. Bourgoin, C. Minshull, M. Legre, T. Jennewein, R. Kashyap, V. Makarov, Laser damage creates backdoors in quantum cryptography. https://obj.umiacs.umd.edu/extended_abstracts/QCrypt_2016_paper_52.pdf (2016)

[21] J. Sanchez. This Is The Real Reason Apple Is Fighting The FBI. http://time.com/4229601/real-reason-apple-is-fighting-the-fbi/

[22] B. Schneier. Did NSA Put a Secret Backdoor in New Encryption Standard? https://www.schneier.com/essays/archives/2007/11/did_nsa_put_a_secret.html

[23] M. Scott. In Wake of Attack, U.K. Officials to Push Against Encryption Technology. https://www.nytimes.com/2017/03/27/technology/whatsapp-rudd-terrorists-uk-attack.html

[24] R. Smith. Understanding encryption and cryptography basics http://searchsecurity.techtarget.com/Understanding-encryption-and-cryptography-basics (updated January 2003)

[25] K. Zetter. How A Crypto 'Backdoor' Pitted the Tech World Against the NSA. https://www.wired.com/2013/09/nsa-backdoor/