Final Paper

11 May 2017

Honor pledge

Adam Burbidge

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.

Assignment

The final paper is an individual research assignment on an aspect of Systems Security Engineering that is covered in the class. Topics need to be selected and submitted for approval. The deliverable is a formal research paper of no less than 15 pages. The paper should be presented in a format suitable for publication. Students are encouraged to seek publication and/or conference presentation for their papers.

Response

On Quantum Computing and Behavioral Analytics for the Future of Computer Security

Abstract

The amount of everyone's personal and private data available on the internet is increasing, as are the number of transactions involving that data. Consequently, cybersecurity considerations are rapidly becoming an area that everyone needs to be concerned about. Encryption methods have long been used to protect sensitive information from falling into the wrong hands; the current generation of algorithms typically use some form of public-key cryptography. However, recent technological advancements in quantum computing may eventually render current algorithms obsolete, and new ones will have to be devised. In this paper we look first at a brief history of cryptography and then consider the implications of quantum computers on current-generation algorithms, as well as whether manufacturers should be legally required to provide a way for their devices' security to be bypassed. We also discuss how another emerging technology, behavioral analytics, might be used to identify users and detect suspicious behavior, even if the user's credentials have been stolen. We conclude with a brief discussion of the possible privacy concerns that the use of such technology might invoke.

Introduction

For probably almost as long as there has been written language, there have been people wishing to send messages that can only be read by the intended recipient, and not by any third-party who might intercept the message along the way. Such “secret codes” are often used by militaries and governments to communicate orders and confidential information among soldiers and agencies without an enemy agent knowing what was said. In more modern times they also find applications as part of an

identity management scheme so that both parties in a conversation can be sure that the other really is who they say they are and that their messages are unaltered. In this paper we will first look briefly at some of the history and evolution of secret codes and encryption methods, and then consider how two emerging technologies, quantum computing and behavioral analytics, might be used undermine or enhance existing security, and perhaps help us to come up with new encryption algorithms and be able to identify an individual even if their credentials have been stolen or compromised.

First ciphers - Caesar Cipher and onwards

One of the simplest examples of a secret cipher is the Caesar cipher, so called because Julius Caesar reportedly used it to send confidential information to his soldiers. This is a simple substitution cipher where each letter is replaced by one from a different position in the alphabet. Caesar supposedly used it with a shift value of 3; in the more general case, with the 26-letter English alphabet, a shift of any number up to 25 could be used. (A shift of 0 is simply the original text, not encrypted at all.) However, given the relatively small number of possibilities, if we knew that such a substitution had been used, it would not take long, even by hand, to simply try every combination until we found one that yielded a coherent message; a modern computer could do this within seconds. This type of cipher therefore provides only very minimal actual protection, effectively zero with access to automated tools; even in Caesar's day, the "security" came from the assumptions that many (perhaps even most) enemy agents who might intercept a message would either be illiterate, suppose that the message was written in some unknown foreign language, or simply not know the method to decipher the message. [1] If these assumptions were not correct, it is entirely possible that the enemy could figure out how to read the message within a relatively short amount of time. (It is not known whether there were any successful attempts to break the Caesar cipher during his time as the earliest available records date from the 9th century, however the point is that it would be relatively easy for a determined enemy to do so.) [2] Even without knowing that a substitution cipher was used, if we can study the message itself, we can do things like a frequency analysis, counting the number of times that each letter shows up in the

message. We know that in English, for example, the letter E is the most frequently used letter, so if we see that a particular letter appears many times in the enciphered message, we might guess that it represents the letter E. (Other languages also have their own characteristic letter and word frequencies.) For this purpose, a longer message might actually be easier to crack than a shorter one because it will give us more data to analyze.

An improvement on the Caesar cipher is the Vigenère cipher, where instead of a simple substitution (with a one-to-one correspondence between original and encrypted letters) there is a secret codeword that allows us to select encryptions from multiple shifted alphabets. This helps because a simple frequency analysis of letters in a message will no longer suffice, as there is no longer a unique correspondence between letters. However, if the message is long enough, and the keyword is short enough, there will still be repeating patterns in the encrypted message, and we can perform a modified form of a frequency analysis. The solution to this would be to use a codeword that is at least as long as the message itself; then there will no longer be any repeating patterns, although the enemy might eventually be able to guess the codeword if it is too simple. Ideally, then, the codeword should be at least as long as the message to be encrypted so that there will not be any repeated patterns, and selected randomly so that it cannot be guessed. Such a system is a *one-time pad*, which has been shown to be unbreakable. [3] (A simple example will illustrate why: suppose an intercepted ciphertext reads “KNQPRT”. Using one decryption key, we could recover the (plaintext) message “ATTACK”. Using a different decryption key, however, we might instead recover “GOHOME”. If the keys are truly selected randomly, we have no clues for choosing one over the other, and in fact any plaintext message could be recovered by using a different key. In this example, simply changing keys could yield any number of six-letter messages, with no way to know which is the “real” one.)

While a one-time pad is theoretically cryptographically perfect, there are several practical problems to overcome that make it impractical for most real-world applications. For one thing, the encryption keys

can only ever be used once (hence the name *one-time* (and “pad” because they keys were originally written on physical pads)), otherwise there will start to be repeating patterns, and it might become possible to perform some analysis as described above to recover encrypted messages. Ideally, the encryption keys should be destroyed after use to prevent the enemy from obtaining them. There is also the issue of having the encryption keys actually be truly random; as we will see below, this is not necessarily an easy problem to solve, and if the keys form any kind of predictable pattern then it might be theoretically possible to recover them. Finally, there is the problem of exchanging the keys themselves without the enemy intercepting them. If we have the means to successfully transfer the one-time keys securely while maintaining their absolute secrecy, then presumably we would also have the means to directly transfer the message itself securely and secretly. (The advantage of exchanging the keys is, of course, that we only have to perform the transfer once and then we can securely transmit multiple messages up to the total length of the keypad. We do, however, still have to maintain absolute secrecy of the pad itself for the duration of its useful life. In general, we might not have the means or resources to do this, or to perform multiple key exchanges, so there is constantly a risk that the enemy might discover the key.)

Given the difficulties of implementing a one-time pad system, is there perhaps a “better” method that could be used, that would not require such an elaborate setup? The examples discussed above were all forms of “symmetric” encryption; that is, the messages can be decrypted using the same key that was used to encrypt them. However these methods contain a fundamental weakness which is precisely due to this symmetry: once the secret key and/or encryption method is known (whether it’s stolen, guessed, or otherwise compromised somehow), then not only can we decrypt any subsequent message sent using that method, but we can also generate false new messages that may be indistinguishable from the “real” messages. What we need instead is some encryption method where it won’t matter if the enemy captures the keys we exchange; what we need is public key encryption.

Public key cryptography

With public key encryption, what we have is a pair of keys, one of which is used for encryption and the other for decryption. One of the keys is given away publicly, while the other is kept private. A message encrypted with one of the keys can only be decrypted with the corresponding other key. Here, then, is the advantage over the secret-key methods discussed above: it does not matter if the enemy captures the public key (and in fact, we assume that they will, since it is given away publicly and we make no attempt to conceal it); they will not be able to decrypt messages using it unless they have the corresponding private key. The private key is never exchanged, so it is as secure as our storage method. (How secure the storage method is (and especially, how to improve its security) may be the topic of another paper, but the point is that physical security is still an important consideration even in the digital age.)

For this system to work, essentially what we need is some mathematical operation that's easy to perform, but difficult to reverse unless we have some additional "secret" information. (This "secret" information will be our private key.) It turns out that factorization is one such operation: if we have (or can find) two large prime numbers (often called 'p' and 'q'), it is easy to multiply them together to produce their product 'N'. Given only the product, however, it is difficult to determine the factors that created it. When used in conjunction with the modulus function, we can devise an algorithm where we can easily encrypt a message using the product, but we cannot easily decrypt the message unless we already know the factors; in essence, this is the basis of RSA cryptography. There are many excellent textbooks, articles, and tutorials that describe how this works in more detail (for instance, [1] [2]), but without worrying too much about the mathematics involved, suffice it to say that if we could find the factors of the "semi-prime" product N, then we could recover the private key. (For a very simple example, suppose that Alice selects the primes p=7 and q=11; she multiplies them together to get the product N=77, and sends this product to Bob as part of her public key. Bob then performs some mathematical operation using N to encrypt his message and returns it to Alice. Since Alice knows the

factors p and q, she can use this information to decrypt the message. However a third party, Eve, who has been listening in on the conversation, retrieves the encrypted message and the product N, but is unable to decrypt the message until she first determines the factors of N. This example uses only very small prime numbers and so their product can be factored very quickly, but the method of doing so illustrates the challenge of factoring large numbers: in reality p and q could have hundred of digits, so factoring N using known techniques could take years. (It should be noted that this example is greatly oversimplified and intended for illustration only; further details of the mechanics of the encryption algorithm and factoring methods are discussed in [1]. It is also interesting to note that the characters Alice and Bob were first named by Ron Rivest, Adi Shamir, and Leonard Adleman in their paper on the subject, "A method for obtaining digital signatures and public-key cryptosystems" - their initials give the RSA algorithm its name. The third party is typically called Eve, the eavesdropper. While there do exist some forms of attack that would allow a third party to inject their own falsified messages, for the purpose of this example Eve is only a passive listener; she can hear everything that Alice and Bob say to each other, but she cannot alter it.)) The challenge, therefore, is to factorize N. While there are algorithms that can be used to do this, eventually they all come down to some form of trial and error. Since N is a (very) large number, this is difficult on a classical digital computer largely because of the amount of time it will take. On a quantum computer, however, we could use Shor's Algorithm to dramatically reduce the amount of time required to find the factors.

To simplify the details, in essence the "quantum" part of Shor's algorithm boils down to finding the period (call it r) of the modulus function ($a^n \bmod N$), where N is the number we want to factor, a product of two large prime numbers, n is an increasing integer, and a for this algorithm is a number (less than N) that we chose at random (that is, we want to find how long it takes until the output values start to repeat themselves). This is difficult on a traditional silicon-based digital computer because it effectively still requires checking numbers in sequence and then counting how long it takes until the results start repeating. If N is a large number (which, for cryptographic purposes, we already know that

it is; it might have hundreds, or even thousands, of digits), then the period `r` could potentially be nearly as large as `N` itself, and this calculation could take a prohibitively long time. In fact, this (i.e. the amount of time it would take to determine the period of the sequence) is effectively “all” (so to speak) that protects data encrypted with an RSA algorithm: there is currently no known way using a traditional digital computer and classical algorithms to figure out the period within a reasonable amount of time for large values of `N`. [1] If we had access to a quantum computer that could implement Shor’s Algorithm, however, we could speed up this computation dramatically.

Quantum computing

There is a perception, common in popular culture, that a quantum computer is akin to having many traditional digital computers working on a problem in parallel. This notion is somewhat true in the sense that, like the famous Schrodinger's Cat thought experiment, the quantum computer is internally in a state of superposition (with some probability of being in any given state), so in a way it does “sort of” process all possible inputs simultaneously. The problem is that, also like Schrodinger’s Cat, when we try to look at the result the superposition collapses to just a single state which, in the case of the calculation above, may or may not be the answer we wanted. We can, however, apply some additional manipulations to our methodology that will increase the probability that the final output will be the one we want. In this case, the additional manipulation is something called a Quantum Fourier Transform, which will effectively increase the probability of the “right” answer being selected when the waveform collapses and decrease the probability of the “wrong” answer.[11] Once we have the period `r`, and after completing a few more calculations from the remaining steps of Shor’s algorithm, we can find the prime factors of `N`, and once we have those we can recreate the private key, thereby breaking RSA security!

While we have skipped over almost all of the mathematics involved, if a quantum computer can theoretically solve a factorization problem, then does this mean that RSA and related algorithms are effectively dead and we have come to the end of cryptography as we know it? Fortunately, for the time

being at least, the answer is ‘no’: as yet there is no existing quantum computer that can actually implement Shor’s algorithm on a scale large enough to factor the numbers used for cryptographic purposes. Currently the largest number known to have been factored with this method is 21, and even in this case the method required prior knowledge of the solution. [13] However, it is likely to be only a matter of time until larger numbers are factored successfully. Significantly, the number 56153 has also been factored using a different method called adiabatic quantum computation (an alternate to Shor’s algorithm), which does not require prior knowledge of the solution, and which was achieved using only four qubits. (A ‘qubit’ is the basic unit of a quantum computer, a “quantum bit”.) The researchers have also shown theoretically (although not yet demonstrated in practice) that 291311 could also be factored using only six qubits. [13] [14] This is important because adiabatic quantum computation is a subclass of a process called quantum annealing, and quantum annealing computers are being actively developed, the most well-known of which is probably the one produced by D-Wave, a Canadian company that recently (Jan 2017) announced they had begun selling a 2000-qubit system. [18] In light of this, should we revise our initial assessment and start worrying anew about the future of cryptography? For now, the answer is still ‘no’: quantum annealing in general is designed to solve optimization problems and find the local minimum of a function. This is good for problems such as the well-known travelling salesman problem, where it may be sufficient to find a “good enough” solution even if it is not necessarily the absolute best. However, even for this class of problem, at this time there are still classical algorithms that can outperform quantum annealing. [13] For breaking cryptography, we need to know the exact factors of a given large number, not just ones that are ‘close’; quantum annealing cannot implement Shor’s algorithm. Quantum computing itself also comes with its own set of challenges. For example, the superposition of quantum states only exists as long as the qubits are unobserved; once observed, the waveform collapses into a single definite value. The environment is always threatening to measure the qubit in some way, so there is a challenge to ensure that the superposition can be maintained until the measurement is needed and the waveform does not collapse prematurely. [15] [20] In addition, a hacker could potentially use a strong pulse to disrupt the detectors

used to capture the quantum states, making them unable to determine when there is something to be measured and thereby invalidating their results. [20] However, these potential problems are not fundamentally unsolvable, so it is likely still only a matter of time until they can be solved and RSA cryptography does become vulnerable. In recognition of this, there is active research into post-quantum cryptography (that is, cryptographic methods that will be resistant to analysis by quantum computers) [23] [25], and as of 2016 the National Institute of Standards and Technology (NIST) currently has a call for proposals for “quantum-resistant cryptographic algorithms” to become the “new public-key crypto standards”. [21]

One such example of a “post-quantum” algorithm is one that Google has been experimenting with in its Chrome browser, dubbed “New Hope”. [22] In their press release, the company stated that they did not yet know whether the algorithm would turn out to be breakable even with existing computers (hence the need to perform trials and experiments), but if it was successful it would be able to protect secure connections even against future quantum computers. [24] The details of the algorithm are discussed in a paper by researchers Erdem Alkim, Léo Ducas, Thomas Pöppelmann and Peter Schwabe [25], and research is currently ongoing to see whether it, or another proposed post-quantum algorithm, will be suitable as a replacement for existing encryption algorithms. [27] [28]

It should be noted that in addition to potentially breaking current public-key cryptographic methods and forcing us to devise new algorithms, there are also ways in which quantum computers might be able to help enhance existing (as well as future) security. For instance, RSA-based cryptography depends in part on the factorization of large numbers, and much of this discussion has concentrated on methods to increase the speed of obtaining the factors, since this information can be used to recover a private key and so decrypt a message. However, another part of cryptography also depends on generating random numbers, a process that has many applications besides just cryptography (ranging from game design to weather prediction). For one, since the security of the encryption depends on the semi-prime N being

difficult to factor, its prime factors must have been randomly selected; if there are two numbers that happen to use the same value as one of their factors, then both can be factored by computing the greatest common divisor between `N1` and `N2`, and there are existing algorithms for doing this efficiently. (If `N1=pq` and `N2=pq′` then `gcd(N1,N2)=p` (their first factor), which will allow us to obtain the other factors (`q` and `q′`) of both `N1` and `N2`, so we can now determine both private keys; algorithms to compute the greatest common denominator of two integers are also discussed in [1].)

However, existing digital computers are unable to generate “true” random numbers, and instead rely on algorithms that will generate “pseudo” random numbers. These numbers may appear to be statistically random, but if the sequence can be in some way predicted, then there is effectively no longer any security and, with a little more work (again well within the capabilities of existing digital computers), data “encrypted” with the compromised sequence could be decrypted by a third party who knew the sequence. This would effectively be a “backdoor” into the encryption algorithm. While we know that the numbers aren’t really “random”, the sequence itself should therefore be unpredictable, because if it isn’t then we can theoretically determine the cryptographic keys, and so decrypt any communications that used the compromised sequence.

[Author’s note: This is a question that I had when I first started learning about cryptographic algorithms over the course of this class: if the security of my public key depends on producing a value of `N` which is the product of two randomly chosen and unknown primes, how can I be sure that “my” primes are unique to me, and do there not exist published lists of known primes? (If so, it seems that the problem could be reduced to a simple lookup table.) As it turns out, there are algorithms that can perform a “primality test” to quickly determine whether or not a given number is prime, without necessarily calculating the exact factors of a non-prime number. Since there are infinitely many primes, part of the “setup” step of software like GPG does involve finding two previously-unknown primes of suitable length. For my public key to remain secure, there must also be a negligible probability that anyone else has selected the same prime number as me for one of their factors. A method to estimate the number of primes in a given interval is discussed in [1]. For the interval (0,10^100] that value is approximately 4.36e97.]

Random number generation

In 2007, researchers from Microsoft published work that suggested the possibility that such a backdoor might indeed exist in the NIST SP800-90 DUAL_EC_PRNG algorithm. [30] This algorithm works by

using two points on an elliptic curve, but the researchers showed that if the relationship between these two points was known, then it might be possible to predict the sequence of (no-longer-random) numbers. Indeed, after experimental verification of their method, they stated that “in every experiment 32 bytes of output was sufficient to uniquely identify the internal state of the PRNG [pseudo-random number generator]”. [30] Since it was not known how the points on the curve were initially chosen, the researchers could not say whether or not the algorithm designer knew the relationship between the points, and consequently they could not speculate whether or not NIST had intentionally built a backdoor into the algorithm. However in their conclusions, they did say that “the prediction resistance of this PRNG (as presented in NIST SP800-90) is dependent on solving one instance of the elliptic curve discrete log problem. (And we do not know if the algorithm designer knew this beforehand.)” [30] Evidence subsequently emerged that the algorithm might indeed contain an intentional backdoor; in light of growing concerns, NIST withdrew its Special Publication 800-90A with the note that “the specification of the Dual_EC_DRBG has been removed. The remaining DRBGs [Deterministic Random Bit Generators] (i.e., Hash_DRBG, HMAC_DRBG and CTR_DRBG) are recommended for use.” [29]

Quantum Random Numbers?

At this point an interesting question arises: we have discussed quantum computers in the context of possibly breaking existing cryptographic algorithms by making it easy to solve problems that are currently difficult (or rather, extremely time-consuming, if not technically challenging) such as the factorization problem. We have also discussed another way in which an algorithm might be compromised, by feeding it a predictable sequence of numbers, since digital computers are unable, on their own, to generate “true” random numbers (but rather, they can only produce a supposedly-unpredictable sequence of ‘pseudo-random’ numbers). So, if the security of cryptographic methods relies partly on the generation of random numbers, the question becomes whether quantum computers could generate “true” random numbers. There is some evidence to suggest that the answer might be ‘yes’: quantum processes may, in fact, generate random results which are measurably

different from those produced by known pseudo-random number generators. [33] If this is correct, and if we can generate a long enough sequence of random numbers, then the one-time pad discussed above might become a viable possibility. Indeed, this has been stated to be “the ultimate goal of establishing a long secret key” [32]; we already know that the one-time pad is theoretically unbreakable, so if it were possible to generate and securely transfer a sufficiently long truly random secret key, we could “thus obtain transfer of data in absolute secrecy”. [32] (The problem then becomes one of exchanging the key securely, which was the challenge with using the one-time pad in the first place, but quantum mechanics might be able to help there as well, perhaps by taking advantage of phenomena such as entangled particles.) [15]

Alternate methods to retrieve private data

Today’s cryptographic algorithms are secure only by virtue of the amount of time it would take to solve the underlying mathematics; as discussed above, quantum computers may eventually render them obsolete, but at the same time may also pave the way for truly secure encryption. If true “absolute secrecy” does become a possibility, are there other methods that could be used for obtaining seemingly-secured data? One relatively low-tech possibility that some government agencies are pursuing is simply asking the manufacturers of devices implementing these algorithms to give them access to the contents of the device; in effect, asking the companies to create an intentional backdoor, known to them but not to the general public, and placing legal pressure on them to comply.

In 2016 there was a widely-publicized case in which the FBI recovered an encrypted (work-issued) iPhone from a confrontation in which the suspect to whom the phone was issued was killed in a standoff with police. Being unable to recover the phone’s contents, they subsequently attempted to legally require Apple to create a special build of their software that would allow them to bypass the phone’s security. Apple refused, partly on the grounds that being forced to create software with certain features would be tantamount “to compelled speech and viewpoint discrimination in violation of the First

Amendment" [54], and in their open letter to customers warned of the possible dangers of intentionally weakening security. [52][53] The concern is that once security has been weakened for one user, it has effectively been weakened for all users; there is no way to weaken only one specific instance while keeping all others as secure as they were before. (In the case of number factorization, for example, if a new method is developed, there is no way to limit its use to only one specific instance of factorization; thus, if the quantum computer discussed above can implement a faster factorization algorithm, then *all* encryption methods that depend on the difficulty of factorization for their security have been weakened since they will all be vulnerable to this technique.) In the words of Apple's open letter to its customers, "The FBI may use different words to describe this tool, but make no mistake: Building a version of iOS that bypasses security in this way would undeniably create a backdoor. And while the government may argue that its use would be limited to this case, there is no way to guarantee such control. […] Once created, the technique could be used over and over again, on any number of devices." [52]

Continuing this line of thought, since quantum computers do not yet exist on a scale large enough to be useful for solving real-world problems, and since there is ongoing legal debate about whether companies such as Apple can or should be compelled to create software that intentionally weakens security, perhaps there is another easier way: we could ask the user directly.

Appeal to the user directly

In many security applications it is often the case that the user is the weakest link. After all, there would be no need to break a lock, or even to weaken it, if the user can be convinced, compelled, or otherwise tricked into opening it themselves. In fact, this is one technique often used by malicious opponents to target unsuspecting users: if a well-crafted phishing campaign can appear to be sufficiently legitimate, there will be a certain percentage of users who fall victim to it. (The reason why is, again, a matter of simple mathematics: if a phishing attack goes to, say, 1000 users, then even if only 0.1% of users fall victim to it, that will still be, on average, one person out of the 1000; from the hacker's perspective it

may only take one to achieve the results they want, whether their objective is stealing personal information or penetrating deeper into a corporate network.) There will, after all, be no need to attempt to employ sophisticated techniques such as the ones described above to attempt to discover secret keys or bypass security if the user can somehow be persuaded to voluntarily reveal their private information. [55] [56] While phishing awareness training can help with these cases, there is some evidence that suggests it may in fact be largely ineffective: users who already know not to click on links in suspicious email feel that they don't really learn anything new from the training, and users who routinely fall victim to simulated phishing attempts don't seem to absorb the lessons that the exercise was supposed to teach them. [57] [58] Perhaps better training methods can be devised, but from a technological point of view, are there any emerging technologies that might help to prevent end-users from seeing phishing attempts in the first place, and to limit the amount of damage if they do fall victim? As it turns out, there are.

Behavioral analytics

One particularly interesting emerging technology in cybersecurity is behavioral analytics. At this point we have probably all heard of artificial intelligence and machine learning; searching through large amounts of repetitive data quickly and identifying patterns is one type of problem that classical digital computers are particularly good at, and these types of technologies are already in use, for instance, in the spam filters of email messages. It seems that it would be a relatively small step to expand on this to improve detection of potential phishing emails, and some implementations do make use of the technology. However, not all phishing attempts come via email: some attackers may also employ social engineering methods through phone calls or even face-to-face conversations. These types of attacks may be more difficult to defend against, because if the attack is well-executed, the victim may not even realize that they have given away confidential or private information. Once the attackers have the information, however, they can leverage it as part of a larger attack, with the ultimate goal often being to obtain access to an individual's or a company's private files. For security analysts attempting to

mount a defence against this type of social engineering, given that an attack of this nature tends to have a different activity profile than a legitimate use-case, there are again patterns that a sophisticated computer might be able to detect. To this end, IBM has offered its Watson computer to aid in the search for, and detection of, new cyber threats. [41] [43]

In 2011, Watson famously defeated former *Jeopardy!* champions Ken Jennings and Brad Rutter. It did this by employing many language analysis algorithms to examine its input (*Jeopardy!* clues in this case) and determine statistical similarities, and then generate probable responses with some minimum confidence threshold in the selected answer. [42] With some modifications to its response-generation techniques, Watson could also, for example, analyze network log data to identify anomalous behaviour. It would appear that this is indeed the sort of task that Watson is performing in its new job as a cybersecurity analyst. For example, after studying one security analyst's data, Watson correctly identified "attacks on 34 of the company's laptops linked to a new strand of malware" which had initially appeared to be simply false alarms to the human analyst. [43]

Watson, and machine learning and pattern-matching algorithms in general, can perform well at these sorts of tasks because they can search through large amounts of data much faster than any human analyst, and in many cases the suspicious activity or anomalous behaviour will be readily apparent compared to benign activity because humans, in general, tend to be relatively predictable in their actions. For instance, many people have a daily and/or weekly routine, and tend to get up and go out at around the same time every day, participating in the same activities on the same day of every week. By analysing the relevant data, It would not take an artificial intelligence long to notice such trends. If applied in the workplace, the AI might recognize that a particular employee usually logs in and out at about the same time every day, and performs similar tasks related to their job function. Suppose instead that this employee logs in from a different location one day, and/or at a different time than usual. By themselves those actions don't necessarily indicate a problem (perhaps the employee has

decided to work from home one day, or is on a business trip elsewhere), but they could prompt a more careful monitoring of activities to see if the user subsequently starts attempting to perform tasks outside of their normal activity, or taking actions that they have never done before. If so, suspicions should be raised because it might indicate that the user has, in fact, had their account compromised. A possible countermeasure at this point might be, for instance, to require the user to respond to increasingly restrictive security challenges to attempt to prove that they really are the legitimate owner of the credentials being used. (For example, if the user possesses a company cellphone or other secondary device, require them to respond to a prompt on it; while still not completely foolproof, this type of protective measure could help to reduce the impact of a malicious entity using stolen credentials.) Similar tactics could also be used in private life to help reduce the occurrence and impact of identity theft. Watson (or another powerful computer dedicated to this task) could search through hundreds or thousands of documents, and so find suspicious trends much faster than a human researcher could. For example, if a user is observed logging into a website from a location they have never been to before, suddenly changing several personal details, and/or making some unusual requests for information, it could raise a few red flags that their credentials have been compromised. If an artificial intelligence such as Watson can detect this and raise appropriate safeguards, it may help to protect users from crimes such as identity theft, despite their own often less-than-perfect cybersecurity habits.

Potential challenges for behavioral analytics

While there are some legitimate use-cases for behavioral analysis and activity tracking such as the ones discussed here for identifying potentially compromised credentials and helping to reduce identity theft, there are also some concerns about the implications about user privacy. [46] Such user profiling is already commonly used, for example, to provide targeted advertizing; many smartphone apps also track the user's location in real-time. Some of the potential risks of advancements in behavioral analytics include surveillance by governments or companies, which is already a growing concern in some areas. The tracking data could also be used for price or service discrimination (being able to

accurately predict what an individual will do is valuable information in many situations, and could potentially be exploited by agents on either side). While, in the context of an employer and their employees, many companies already have policies in place stating that company assets are monitored and the users should not necessarily have any expectations of privacy, clearly a balance has to be reached for the more widespread deployment of such technologies among the general public. Part of this balancing act may involve implementing initiatives to educate users about cybersecurity topics and protecting their online privacy. [60] As with the quantum computers discussed previously, behavioral analytics are a powerful tool that can be used to improve security (especially when employed to detect known attack methods and identify users deviating from their normal behavior) [50], but could also be misused in the wrong hands. Suppose, for instance, that a malicious entity somehow obtained the analytic data: it could potentially form the basis of a more convincing identity theft scenario if the attack could be crafted to deviate only slightly, at least at first, from the user's normal behavior. Then the legitimate user might not necessarily know straight away that their information had been compromised. There are ongoing discussions that attempt to address some of these concerns (for example, [47]), but additional efforts are still needed. It is also worth noting that malicious entities will not hesitate to use any technologies at their disposal, including artificial intelligence and advanced data collection methods, so cyber security experts need to be prepared to respond with similar technologies. [51]

Conclusions

As more of our personal and private information becomes available online, and as the state of the art of modern technology continues to advance, cybersecurity is inevitably becoming an area that everyone needs to be concerned about. Even users who may feel they have "nothing to hide" could find themselves targets of cyber attacks, if only for the additional access their credentials could provide. In this paper we have studied some of the various encryption methods to protect private information, ranging from simple substitution ciphers up to current-generation public-key encryption. In the future, it is possible that quantum computers may eventually render these methods obsolete. There is current

research being performed to study potential new cryptographic algorithms that will be resistant to attacks by a quantum computer, and work should continue in this area regardless of whether or not quantum computers eventually become a reality, because such work will ultimately improve security for everyone. Quantum computers might also eventually offer improvements for another area of cryptography, that of generating random numbers. Currently existing digital computers are unable to generate “true” random numbers and instead rely on algorithms to produce “pseudo-random” numbers. As discussed above, if the generating algorithm is compromised then cryptographic methods that depend on it are also compromised, but it might be possible to take advantage of quantum effects to produce numbers that are, in fact, “truly” random.

While waiting for these technologies to become realities, there is also increasing use of artificial intelligence and machine learning techniques for behavioral analysis and trend monitoring. This type of data analysis can identify suspicious activity, so its increased use may help to protect users who have had their credentials stolen by identifying out-of-character activity. In this case, however, the challenges may be more of a legal and legislative nature rather than technological ones if there is resistance among the general public against having their activities tracked and studied by such technologies. Continued work is also need in this area to address the concerns and reach a suitable balance for the implementation of these methods. In the meantime, companies and individuals need to stay vigilant and assess their own risks for cyber attacks, taking appropriate actions to prepare for the inevitable day when they will be the victim of an attack. [59] [60]

References

Cryptography

[1] Pieprzyk, Josef; Thomas Hardjono; Jennifer Seberry (2003). *Fundamentals of Computer Security*. Springer. ISBN 3-540-43101-2.

[2] Singh, Simon (2000). *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography.* ISBN 0-385-49532-3.

[3] Pro-Technix, Cryptology and Data Secrecy : The Vernam Cipher. http://www.pro-technix.com/information/crypto/pages/vernam_base.html

[4] Tony Lawrence, *GPG/PGP Basics.* 2014-10-29. http://aplawrence.com/Basics/gpg.html

[5] Keith W. Ross, *Fundamentals of Cryptography.* https://faculty.iima.ac.in/~jajoo/ec/cryptography.htm, 1996

[6] Jennifer Beddoe, *How to Find the Prime Factorization of a Number.* http://study.com/academy/lesson/how-to-find-the-prime-factorization-of-a-number.html

[7] Michael Stum et al. *Why are primes important in cryptography?* http://stackoverflow.com/questions/439870/why-are-primes-important-in-cryptography, Jan 13, 2009

[8] Alison DeNisco, *5 reasons your company can't hire a cybersecurity professional, and what you can do to fix it.* March 29, 2017. http://www.techrepublic.com/article/5-reasons-your-company-cant-hire-a-cybersecurity-professional-and-what-you-can-do-to-fix-it/

Quantum Computing

[9] Peter W. Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.* https://arxiv.org/pdf/quant-ph/9508027v2.pdf, 25 Jan 1996

[10] SAMUEL J. LOMONACO, JR, *A LECTURE ON SHOR'S QUANTUM FACTORING ALGORITHM.* https://arxiv.org/pdf/quant-ph/0010034.pdf, 9 Oct 2000

[11] Scott Aaronson. *Shor, I'll do it*, http://www.scottaaronson.com/blog/?p=208, Feb 24, 2007

[12] Cambridge University Press. 978-0-521-86485-5 - *Quantum Cryptography and Secret-Key Distillation.* Gilles Van Assche, http://gva.noekeon.org/QCandSKD/QCandSKD-introduction.pdf

[13] Lisa Zyga, *New largest number factored on a quantum device is 56,153.* November 28, 2014. https://phys.org/news/2014-11-largest-factored-quantum-device.html

[14] Nikesh S. Dattani and Nathaniel Bryans, *Quantum factorization of 56153 with only 4 qubits.* Dec 1, 2014. https://arxiv.org/pdf/1411.6758.pdf

[15] Josh Clark. *How Quantum Cryptology Works* http://science.howstuffworks.com/science-vs-myth/everyday-myths/quantum-cryptology.htm

[16] Anastasia Marchenkova, *What's the difference between quantum annealing and universal gate quantum computers?* Feb 27, 2016 https://medium.com/quantum-bits/what-s-the-difference-between-quantum-annealing-and-universal-gate-quantum-computers-c5e5099175a1

[17] Luke Graham, *Quantum computer worth $15 million sold to tackle cybersecurity.* 26 Jan 2017. http://www.cnbc.com/2017/01/26/quantum-computer-worth-15-million-sold-to-tackle-cybersecurity.html

[18] D-Wave Systems Inc., *D-Wave Announces D-Wave 2000Q Quantum Computer and First System Order.* JAN 24, 2017 https://www.dwavesys.com/press-releases/d-wave%C2%A0announces%C2%A0d-wave-2000q-quantum-computer-and-first-system-order

[19] KLINT FINLEY, *Quantum Computing Is Real, and D-Wave Just Open-Sourced It.* Jan 11, 2017. https://www.wired.com/2017/01/d-wave-turns-open-source-democratize-quantum-computing/

[20] Adam Mann, *Laws of Physics Say Quantum Cryptography Is Unhackable. It's Not*, https://www.wired.com/2013/06/quantum-cryptography-hack/, Jun 7 2013

Post-Quantum Cryptography

[21] Dustin Moody, *Post-Quantum Cryptography: NIST's Plan for the Future.* 2016. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/pqcrypto-2016-presentation.pdf

[22] Bruce Schneier, *Google's Post-Quantum Cryptography.* Jul 12, 2016, https://www.schneier.com/blog/archives/2016/07/googles_post-qu.html

[23] Daniel J. Bernstein and Tanja Lange et al., *Post-quantum cryptography.* Jan 22, 2017. https://pqcrypto.org/

[24] Matt Braithwaite, *Experimenting with Post-Quantum Cryptography.* July 7, 2016. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

[25] Erdem Alkim et al., *Post-quantum key exchange – a new hope.* Nov 1, 2015. https://eprint.iacr.org/2015/1092.pdf - https://cryptojedi.org/papers/newhope-20151101.pdf

[27] Ian Malloy and Dennis Hollenbeck, *Inversions of New Hope.* https://arxiv.org/ftp/arxiv/papers/1608/1608.04993.pdf

[28] Andrea Russo et al., *Is the "New Hope" Lattice Key Exchange vulnerable to a lattice analog of the Bernstein BADA55 Attack?* May 21, 2016. https://crypto.stackexchange.com/questions/35488/is-the-new-hope-lattice-key-exchange-vulnerable-to-a-lattice-analog-of-the-ber

Random Number Generation

[29] Elaine Barker and John Kelsey, *NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators.* January 2012. http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf (Note: WITHDRAWN)

[30] Dan Shumow and Niels Ferguson, *On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng.* http://rump2007.cr.yp.to/15-shumow.pdf

[31] Matthew Green. *How does the NSA break SSL?* Dec 3, 2013. https://blog.cryptographyengineering.com/2013/12/03/how-does-nsa-break-ssl/

[32] Mario Stipčević and Çetin Kaya Koç, *True Random Number Generators.* http://cs.ucsb.edu/~koc/cren/docs/w06/trng.pdf

[33] Emerging Technology from the arXiv, *First Evidence That Quantum Processes Generate Truly Random Numbers.* April 13, 2010. https://www.technologyreview.com/s/418445/first-evidence-that-quantum-processes-generate-truly-random-numbers/

Emerging Cybersecurity Technologies

[34] John P. Mello Jr., *5 emerging security technologies set to level the battlefield.* https://techbeacon.com/5-emerging-security-technologies-set-level-battlefield

[35] Robert Westervelt, *10 Emerging Security Technologies Gaining Interest, Adoption.* June 17, 2013. http://www.crn.com/slide-shows/security/240156647/10-emerging-security-technologies-gaining-interest-adoption.htm

Machine Learning and Artificial Intelligence in Cybersecurity

[36] TechTarget, *Machine learning in security explodes: Does it work?* March 2017. http://searchsecurity.techtarget.com/ezine/Information-Security-magazine/Machine-learning-in-security-explodes-Does-it-work

[37] Sean Martin, *It's a Marketing Mess! Artificial Intelligence vs Machine Learning.* https://itspmagazine.com/from-the-newsroom/its-a-marketing-mess-artificial-intelligence-vs-machine-learning

[38] Andrew Thomson, *10 HOT STARTUPS USING ARTIFICIAL INTELLIGENCE IN CYBER SECURITY.* MARCH 11, 2016. http://blog.ventureradar.com/2016/03/11/10-hot-startups-using-artificial-intelligence-in-cyber-security/

[39] Rebecca Linke, *IBM Watson: Regular A.I. by day, cybercrime fighter by night.* FEB 14, 2017. http://www.computerworld.com/article/3169556/artificial-intelligence/ibm-watson-regular-ai-by-day-cybercrime-fighter-by-night.html

[40] Ian Sherr, *IBM built a voice assistant for cybersecurity.* February 13, 2017. https://www.cnet.com/news/ibm-built-a-voice-assistant-for-cybersecurity-hayvn-watson/

[41] Tim Greene, *IBM's Watson teams up with its SIEM platform for smarter, faster event detection.* FEB 14, 2017. http://www.networkworld.com/article/3169884/security/ibm-s-watson-teams-up-with-its-siem-platform-for-smarter-faster-event-detection.html

[42] Miles Brundage and Joanna Bryson. *Why Watson Is Real Artificial Intelligence* http://www.slate.com/blogs/future_tense/2014/02/14/watson_is_real_artificial_intelligence_despite_claims_to_the_contrary.html

[43] Alex Konrad , IBM Turns Watson Into A Cybersecurity Weapon Amid White House Interest. FEB 13, 2017. https://www.forbes.com/sites/alexkonrad/2017/02/13/ibm-turns-watson-to-cyber-security/

Behavioral Analytics

[44] Brett DiNovi, *3 Ways Behavior Analysis & Artificial Intelligence Will Change The World.* FEBRUARY 21, 2017. http://www.bsci21.org/3-ways-behavior-analysis-artificial-intelligence-will-change-the-world/

[45] Kenneth Corbin, *Feds to battle cybersecurity with analytics.* MAR 29, 2017. http://www.cio.com/article/3185881/government/feds-to-battle-cybersecurity-with-analytics.html

[46] ENISA, *Privacy considerations of online behavioural tracking.* https://www.enisa.europa.eu/publications/privacy-considerations-of-online-behavioural-tracking/at_download/fullReport

[47] Michelle Baddeley. *A Behavioural Analysis of Online Privacy and Security.* http://www.econ.cam.ac.uk/dae/repec/cam/pdf/cwpe1147.pdf

[48] Till Johnson, Johna. *User behavioral analytics tools can thwart security attacks.* http://searchsecurity.techtarget.com/feature/User-behavioral-analytics-tools-can-thwart-security-attacks

[49] Richards, Kathleen. *User behavior analytics: Conquering the human vulnerability factor.* http://searchsecurity.techtarget.com/feature/User-behavior-analytics-Conquering-the-human-vulnerability-factor

[50] Joseph Busch, *User Behavior Analytics and Privacy: It's All About Respect.* Oct 24, 2016. https://community.rapid7.com/community/insightidr/blog/2016/10/24/respecting-employee-privacy-when-deploying-user-behavior-analytics

[51] Roman V. Yampolskiy, *AI Is the Future of Cybersecurity, for Better and for Worse.* MAY 08, 2017. https://hbr.org/2017/05/ai-is-the-future-of-cybersecurity-for-better-and-for-worse

Electronic Device "Forced Backdoor" Dispute

[52] Tim Cook, *A Message to Our Customers.* https://www.apple.com/customer-letter/, Feb 16, 2016

[53] Tim Cook, *Answers to your questions about Apple and security.* https://www.apple.com/customer-letter/answers/, Feb 16, 2016

[54] KIM ZETTER AND BRIAN BARRETT, *Apple to FBI: You Can't Force Us to Hack the San Bernardino iPhone.* https://www.wired.com/2016/02/apple-brief-fbi-response-iphone/, Feb 25, 2016

Individual Security Awareness Training

[55] UNITED STATES COMPUTER EMERGENCY READINESS TEAM, *Security Tip (ST04-014) Avoiding Social Engineering and Phishing Attacks.* https://www.us-cert.gov/ncas/tips/ST04-014, Oct 22, 2009

[56] Nate Lord, *Phishing Attack Prevention: How to Identify & Avoid Phishing Scams.* https://digitalguardian.com/blog/phishing-attack-prevention-how-identify-avoid-phishing-scams, Feb 28, 2017

[57] Howard Solomon, *Phishing awareness training almost futile, say Canadian bank infosec pros.* http://www.itworldcanada.com/article/phishing-awareness-training-almost-futile-say-canadian-bank-infosec-pros/383842, June 3, 2016

[58] KnowBe4, Inc. *Point-Of-Failure Phishing Training Does Not Work.* https://www.knowbe4.com/resources/point-of-failure-phishing-training-does-not-work/

[59] Deloitte, *Five essential steps to improve cybersecurity.* https://www2.deloitte.com/content/dam/Deloitte/ca/Documents/risk/ca-en-risk-cyber-5-steps.pdf

[60] Cindy Brodie, *The Importance of Security Awareness Training.* June 30, 2008. https://www.sans.org/reading-room/whitepapers/awareness/importance-security-awareness-training-33013

Appendix - Tutorials and lectures on the topics discussed in this paper

Encryption

SciShow, *Cryptography: The Science of Making and Breaking Codes,* Aug 6, 2015, https://youtu.be/-yFZGF8FHSg

Android Authority, *How does encryption work?* Sep 13, 2016, https://youtu.be/clWEKq8CVOk

Public key encryption

Art of the Problem (Khan Academy), *Public key cryptography - Diffie-Hellman Key Exchange*, July 30, 2012, https://youtu.be/YEBfamv-_do
Art of the Problem (Khan Academy), *Public Key Cryptography: RSA Encryption Algorithm*, July 30, 2012, https://youtu.be/wXB-V_Keiu8

Numberphile, *Encryption and HUGE numbers*, Dec 9, 2012, https://youtu.be/M7kEpw1tn50
Numberphile, *How did the NSA hack our emails?*, Dec 22, 2013, https://youtu.be/ulg_AHBOIQU
Numberphile, *NSA Surveillance (an extra bit)*, Dec 23, 2013, https://youtu.be/1O69uBL22nY
Computerphile, *Public Key Cryptography*, Jul 22, 2014, https://youtu.be/GSIDS_lvRv4

Android Authority, *How does public key cryptography work*, Oct 4, 2016, https://youtu.be/rLiEA06Bcic

Quantum Cryptography

Physics Girl, *Quantum Cryptography Explained*, March 1, 2016, https://youtu.be/UiJiXNEm-Go

Computerphile, *Quantum Computing 'Magic'*, November 11, 2016, https://youtu.be/BYx04e35Xso

Veritasium, *How Does a Quantum Computer Work?*, Jun 17, 2013, https://youtu.be/g_IaVepNDT4

Kurzgesagt – In a Nutshell, *Quantum Computers Explained – Limits of Human Technology*, Dec 8, 2015, https://youtu.be/JhHMJCUmq28

PBS Infinite Series, *The Mathematics of Quantum Computers*, February 16, 2017. https://youtu.be/IrbJYsep45E
PBS Infinite Series, *How to Break Cryptography*, April 20, 2017. https://youtu.be/12Q3Mrh03Gk

PBS Infinite Series, *Hacking at Quantum Speed with Shor's Algorithm*, April 27, 2017.
https://youtu.be/wUwZZaI5u0c

Apple vs FBI dispute (in general terms)
CGP Grey, *Should all locks have keys? Phones, Castles, Encryption, and You.*, April 14, 2016
https://youtu.be/VPBH1eW28mo
CGP Grey, *Footnote *: I, Phone*, April 14, 2016, https://youtu.be/e-ZpsxnmmbE

Data collection
Barnacules Nerdgasm, *Windows 10 Busted - Leaked Hacking Tool from NSA*, Apr 16, 2017,
https://youtu.be/z8_Tc_DnJy0

Artificial Intelligence
DARPAtv, *A DARPA Perspective on Artificial Intelligence*, Feb 15, 2017, https://youtu.be/-O01G3tSYpU