RSA Encryption

Brandon Liang
MAT255 Number Theory
Davidson College

May 13, 2015

Contents

Abstract

This article briefly outlines the history and development of RSA encryption and how it is stands out compared to other earlier cryptographic methods. It then explains the algorithm and provides examples to help readers better understand RSA encryption. In the end, the article will list one well-known attack on RSA cryptosystem.

1 Introduction

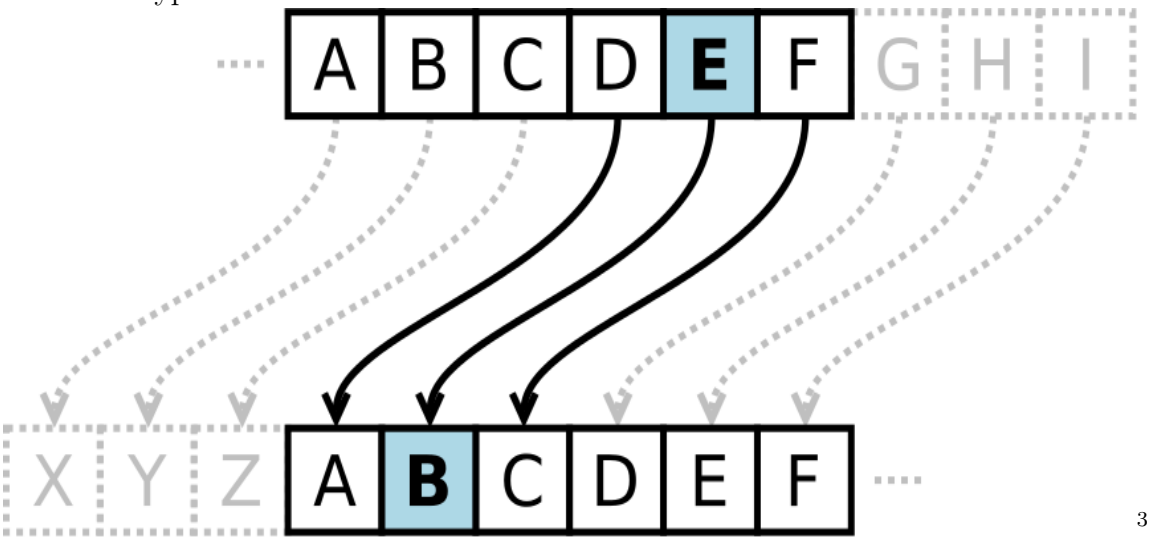
1.1 Early Application of Cryptography

Ever since events are recorded in human civilization, there has been a need of cryptography. To give a formal definition of cryptography, it is the enciphering and deciphering of messages

in secret code or cipher[1]. An informal interpretation could be the protection and secrecy of information and message traveling from the sender to the desired audience or receiver through a public space. The effectiveness of a cryptosystem is measured by the difficulty to decipher the encoded message by the general public. Here the process of enciphering is also known as "Encrypting" and the process of deciphering is also known as "Decrypting".
For clarity purpose, a bit is known as the number of digits by a number and one byte consists of 8 bits.

In CSC121, we had a homework problem called "Caesar Cipher" [2] where we were to crack one of the oldest and most well-known encryption schemes. The given text contains information just like normal words, symbols and punctuations but the only distinction is that every single letter is shifted by the same amount to a new letter. For instance, if the shift amount is 1, then "*a*" would be enciphered to "*b*". The trick here is that since there are only 26 English letters and each letter can only be shifted/enciphered to another letter, one is able to crack the ciphered text by trying at most 26 shift amounts to see which shift amount produces the most English-like text.

In the following illustration, there is a left shift amount of 3 where, for instance, the letter "E" is encrypted as "B".



[3]

Here is an example of a ciphered text and the corresponding deciphered/plain text with a right shift amount of 3[4]:

Ciphertext: WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ
Plaintext: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

[1] "cryptography." Merriam-Webster.com
[2] "It is named after Julius Caesar who is believed to have used the technique to protect important military secrets." Hw6.pdf. CSC121
[3] "Caesar Cipher." Wikipedia
[4] "Caesar Cipher." Wikipedia

Even though the cipher text can be deciphered relatively quickly since there are only 26 possible shift amounts, this example illustrates how early cryptography was first applied and practiced to protect the secrecy of information and messages.

1.2 From "Caesar Cipher" to "Public Key Cryptography" to "RSA"

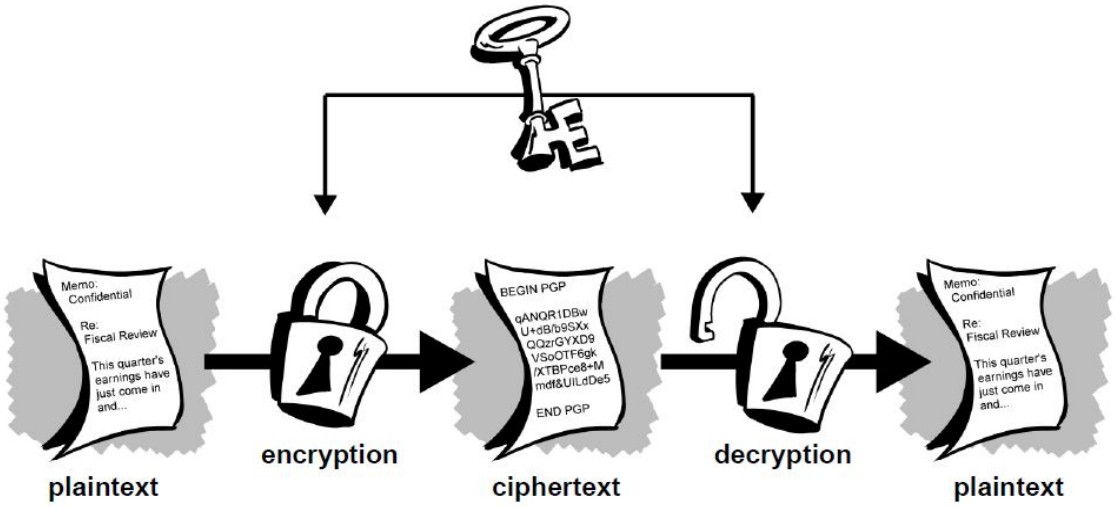
So the essential question arrives: How do we ensure a ciphered message doesn't get deciphered that quickly, or at all? This touches base on the essence of cryptography – protection and secrecy of information.

If a ciphered message can be deciphered rather quickly and effortlessly by anyone, then cryptography becomes pointless. In the "Imitation Game"[5], Alan Turing was able to use his Turing machine and just a slight piece of clue to crack the secret message generated by Enigma and help the Allied defeat the Nazi. However, if the desired audience of a ciphered text is not able to decrypt the message efficiently, it will also hurt the purpose of cryptography, since it may remain an unsolvable puzzle. Thus, the historical development of cryptography had always been trying to achieve these two themes: safety and efficiency. It needs to be difficult for all but some selectively. In other words, before the best encryption method was published (spoiler alert: RSA Encryption!), researchers had tried to come up with a cryptographic method that ensures not anyone but only the desired audience can decrypt the ciphered message.

However, the restraint remained. all the cryptosystems before RSA involve a public key and a private key. The public key is for encryption and the private key is for decryption. As an example, the shift amount in "Caesar Cipher" is both the public and the private key. Since the public key is identical to the private key in "Caesar Cipher", there is no sense of confidentiality in private key, which leads to little difficulty in depichering the "Caesar Cipher". Notice that the public key and private key are inverse functions of each other, because the deciphered message has to match the original message. Hence, in all early cryptosystems, encryption and decryption are considered symmetric, since the process of decryption is simply the opposite or reverse of that of encryption.

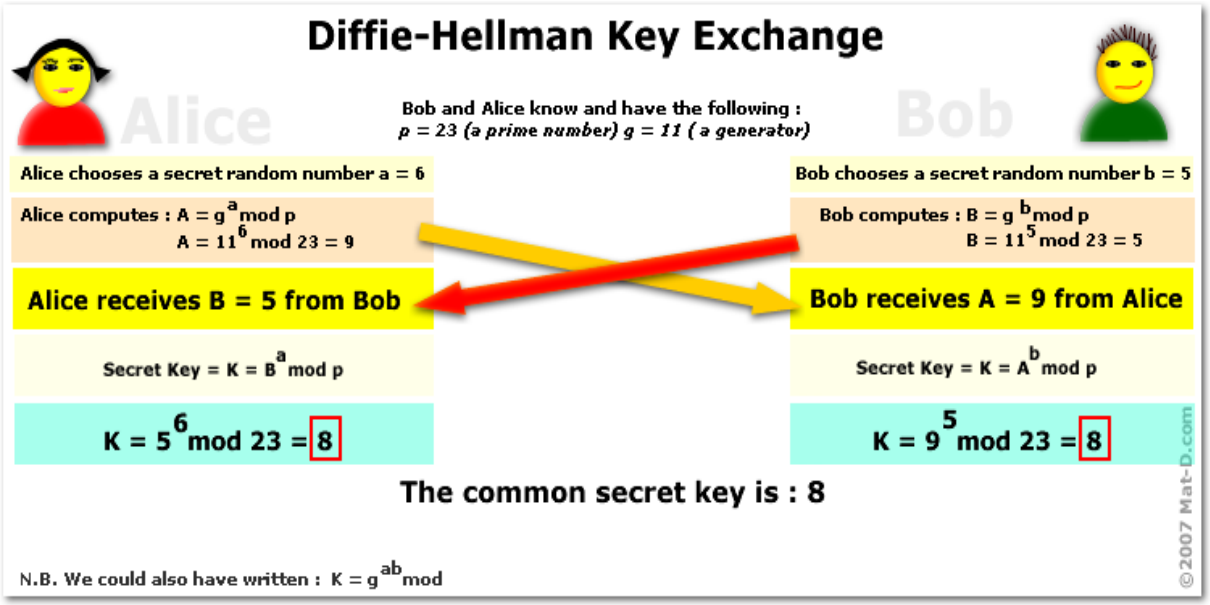
Therefore, the challenge now becomes how to come up with a public and private key system where the private key still reverses the effect of the public key but also preserves confidentiality and secrecy for the receiver of the message against the general public.

[5] "Imitation Game." 2014



[6]

Whitfield Diffie was able to find an asymmetric key, under collaboration with Martin Hellman. In such cryptosystem, the public and private keys are distinct, where the private key is the decryption key and the public key is the encryption key.[7] For example, if Bob wants to send a message to Alice, he is able to encrypt the message using Alices public key, but not able to decrypt an encrypted message, since he doesn't have the private key. Only Alice can decrypt that message, because the private decryption key is asymmetric and thus not open or easy to find. To break it down, this cryptosystem involves a one-way function, "which is irreversible unless the decoder has a special piece of knowledge unknown to the rest of the world–the private key."[8]



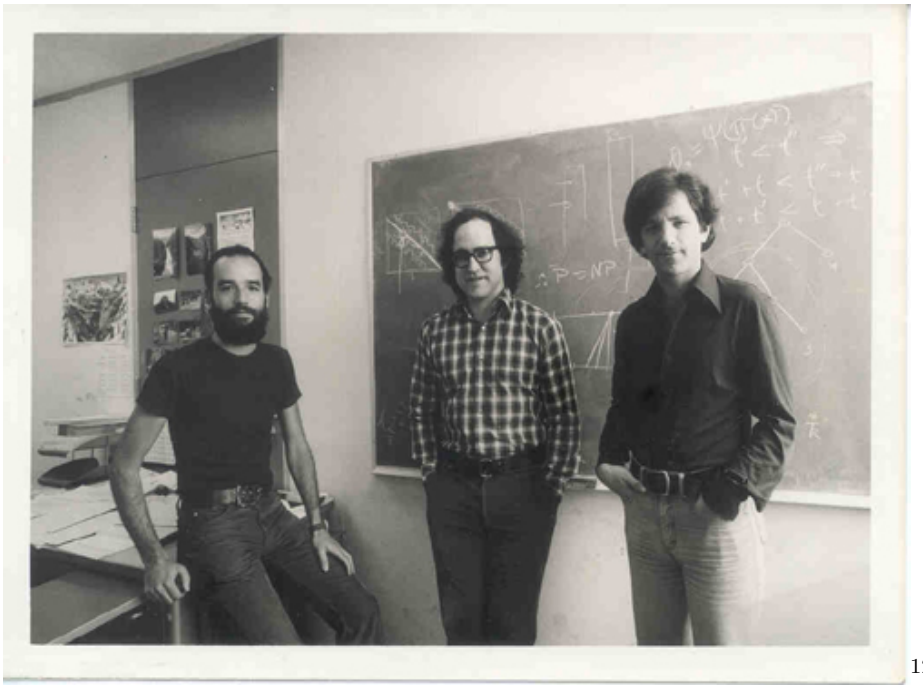
[9]

[6] "Symmetric Key Cryptography (Non-Technical)." abbicabandlng.wordpress.com
[7] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[8] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[9] "Diffie-Hellman Key Exchange." Stackoverflow

However, Diffie never came up with a specific one-way function. Nevertheless, his paper (published in 1975) showed that there was indeed a solution to key distribution with an asymmetric key system that relies on a one-way function and that sparked interest among other mathematicians and scientists.[10] By 1977, three researchers: Ron Rivest, Adi Shamir, and Leonard Adleman, came up with a solution. And that marks the birth of RSA Encryption.

1.3 "RSA" and RSA Encryption

"RSA" stands for the acronym of the surnames of the three contributors: Ron Rivest, Adi Shamir, and Leonard Adleman, who were all computer science professors and researchers at Massachusetts Institute of Technology. The order of the names follows that on the original paper they published in 1977, "A Method for Obtaining Digital Signatures and Public- Key Cryptosystems"[11], where their algorithm of cryptosystem was first presented to the world.



[12]

(from left ro right: Adi Shamir, Ron Rivest and Leonard Adleman)

Rivest is a computer scientist with an exemplary ability to apply new ideas in new places. He also kept up with the latest scientific papers, so as to lead ideas for the one-way function.[13] Shamir, also a computer scientist, has a lightning intellect and ability to focus on the core of a problem. He as well as Rivest generated ideas for the one-way function.[14] Adleman, however, is a mathematician with extraordinary stamina, rigor and patience. He was largely

[10] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[11] "A Method for Obtaining Digital Signatures and Public- Key Cryptosystems", Rivest, Shamir, Adleman.
[12] usc.edu
[13] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[14] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank

responsible for spotting the flaws within the ideas of Rivest and Shamir, and ensured that they did not follow false leads.[15] As "inventors", Rivest and Shamir spent a year coming up with ideas, and Adleman, as a "proofreader", spent a year shooting them down.

One night in April, 1977, Rivest found the answer to the question that had bothered the three of them for a long time: Is it possible to find a one-way function that can be reversed only if the receiver has some special information? He then formalized the solution with the help of Shamir and Adleman.[16] The system was later named "RSA", for Rivest, Shamir, and Adleman, and its success in finding a one-way function makes it stand out as the best and most commonly-used cryptosystem.

2 RSA Algorithm

2.1 How It Works

[17] Here we use Bob and Alice again to explain how RSA encryption works. It involves 6 key elements:

$$p, q, n, \phi(n), e, d$$

If Bob wants to send a secret message to Alice, Alice will need to generate the public and private keys first, since Bob will need this public key to encrypt his message.

So Alice first chooses two random primes p and q. The larger the primes, the safer the encryption.[18] Then she multiplies p and q to get $n = pq$ and also the Euler Phi Function, $\phi(n) = (p-1)(q-1)$, which is also known as Totient. Alice then chooses a random exponent e such that $2 \leq e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. This e, is one of the public keys as it is the encryption exponent. She then finds a d such that $ed \equiv 1 \pmod{\phi(n)}$. This d then becomes one of the private keys, since it is the decryption exponent.

Alice then uses the 6 elements to produce the public and private keys:

$$\text{Public Keys: } \{e, n\}$$
$$\text{Private Keys: } \{d, p, q\}$$

Now, Bob has access to the public keys. He uses the provided encryption exponent, e, and n to encipher his message m into a ciphered text, c:

$$c \equiv m^e \pmod{n}$$

Alice then gets the ciphertext, c and uses her decryption exponent in the private keys, d, to decipher the ciphertext c:

$$m \equiv c^d \pmod{n}$$

[15] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[16] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[17] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[18] "A Method for Obtaining Digital Signatures and Public- Key Cryptosystems", Rivest, Shamir, Adleman.

Since the private keys are never public, no one other than Alice, the desired receiver, will be able to perform the computation efficiently and correctly. For others who don't know the private keys or the decryption exponent, the only way is to factor out n into all possible combinations of two prime numbers in order to then find the Totient $\phi(n)$ and thus the decryption exponenet d. The sections below will explain why this is not an easy or possible approach.

2.2 Generating Public and Private Keys

Notice that in the process of generating the public and private keys, the choice of p and q rather random, as long as they are large enough (explained later on). After computing $n = pq$ and $\phi(n) = (p-1)(q-1)$, the trick lies in the choice of the encryption exponenet e.

Since the encryption exponenet e satisfies the condition that $\gcd(e, \phi(n)) = 1$, where $\leq e < n$, all possible values of e form a set for such encryption exponent and the size of the set depends directly on the value of $\phi(n) = (p-1)(q-1)$. Let k be the cardinality of this set, then Alice will have k distinct values available for e.

Finding the decryption exponent d associated with the encryption exponenet e is also a critial step Alice needs to perform. To find d, Alice needs to solve the congruency equation for d:

$$m \equiv c^d \pmod{n}$$

To solve for d, one will need to apply the Extended Euclidean Alogrithm. [19]

2.3 Why It Works–Number Theory behind RSA

1). Fermat's Little Theorem: If p is a prime and p does not divide a, then a is a generator of p, thus $a^{p-1} \equiv 1 \pmod{p}$.[20]
This is the fundation of the next theorem, Euler's Theorem. Its significance is that for any prime number p, it is able to find a generator whose order is $p-1$. Recall that order is the smallest positive integer exponent to which the the base number can be raised to 1 congruency modulu p.

2). Euler's Theorem: If $\gcd(m,n) = 1$, then $m^{\phi(n)} = 1 \pmod{n}$.[21]
Euler's Theorem applies Fermat's Little Theorem that if m and n are relatively prime to each other, than m has an order of $\phi(n)$. Note that here n isn't necessarily a prime number; in fact, in RSA encryption, n is a composite since $n = pq$, which makes $\phi(n) = (p-1)(q-1)$ since p and q are both primes.
If we raise both sides by a power of z ($z \in \mathbb{N}$) and multiply both sides by m, the plain text,

[19] "A Method for Obtaining Digital Signatures and Public- Key Cryptosystems", Rivest, Shamir, Adleman.
[20] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank
[21] "The RSA Cryptosystem: History, Algorithm, Primes." Michael Calderbank

we get

$$m^{z\phi(n)} \equiv 1^z = 1(\text{mod } n)$$
$$m^{z\phi(n)+1} \equiv m(\text{mod } n)$$

In generating the encryption and decryption exponenets, we have

$$c \equiv m^e \pmod{n}$$
$$m \equiv c^d \pmod{n}$$
$$\text{where } ed \equiv 1 \pmod{\phi(n)}$$

We can rewrite ed as $ed = z\phi(n) + 1$ ($z \in \mathbb{N}$), then $c^d = m^{e^d}$ equals to $m^{ed} = m^{z\phi(n)+1} \equiv m \pmod{n}$, which means that the resulting decrypted message is m that matches the original plain text.
Thus, Euler's Theorem ensures the cumulative effect of encrypting the plain text to get a ciphertext ($c \equiv m^e \pmod{n}$) and decrypting the ciphertext ($c^d \equiv m^{e^d} \equiv m \pmod{n}$ would return the same plain text. The signifiance of Euler's Theorem is that it satisfies the inverser-function relationship between encrypting and decrypting. Note that the encryption exponent e and the decryption exponent d are multiplicative inverses of each other modulo $\phi(n)$, since $ed \equiv 1 \pmod{\phi(n)}$.

3). Extended Euclidean Algorithm is applied to find the decryption exponent d, the multiplicative inverse of the given encryption exponent e, such that $ed \equiv 1 \pmod{\phi(n)}$. In other words, Euler's Theorem ensures why RSA works and Extended Euclidean Alogrithm is the mechanics to fulfill it.
Given $ed \equiv 1 \pmod{\phi(n)}$, we can rewrite it as $ed + z\phi(n) = 1$ for some $z \in \mathbb{Z}$. Then we apply the Euclidean Algorithm to find the greatest common divisor of e and $\phi(n)$ and then use the Extended Euclidean Algorithm to find d. The following illustration gives a perfect example of how to use the Extended Euclidean Algorithm for such task:

Find d where $127d \equiv 1 \pmod{589} \rightarrow 127d + 589z = 1$

$$587 = 4 * 127 + 81 \quad t_2 = t_0 - q_1 * t_1 = 0 - 4 * 1 = -4$$
$$127 = 1 * 81 + 46 \quad t_3 = t_1 - q_2 * t_2 = 1 - 1 * -4 = 5$$
$$81 = 1 * 46 + 35 \quad t_4 = t_2 - q_3 * t_3 = -4 - 1 * 5 = -9$$
$$46 = 1 * 35 + 11 \quad t_5 = t_3 - q_4 * t_4 = 5 - 1 * -9 = 14$$
$$35 = 3 * 11 + 2 \quad t_6 = t_4 - q_5 * t_5 = -9 - 3 * 14 = -51$$
$$11 = 5 * 2 + 1 \quad t_7 = t_5 - q_6 * t_6 = 14 - 5 * -51 = 269$$
$$2 = 2 * 1 + 0$$
$$\textit{The modular inverse of } 127 \textit{ mod } 589 \textit{ is } 269 \textit{ mod } 589.$$ [22]

[22] "Public-Key Cryptography for Dummies (RSA Version)." dengfengli.com

2.4 Example

Now we give an example that involves small prime numbers p and q.(In real-world application, p and q would be a lot larger as explained.)
We pick $p = 5$ and $q = 11$, thus $n = pq = 55$ and $\phi(n) = (p-1)(q-1) = 40$.
Now we find e. Since gcd(e,40) = 1 and $2 \leq e < 40$, we pick $e = 13$. Now we need to find d. $13d \equiv 1 \pmod{40} \rightarrow 13d + 40z = 1$. Use the Extended Euclidean Algorithm:

$$\begin{aligned} 40 &= 3 * 13 + 1 \\ 13 &= 13 * 1 + 0 \\ 1 = 40 - 3 * 13 \rightarrow d &= -3 \equiv 37 \pmod{40} \end{aligned}$$

Now we have the public keys: $\{e = 13, n = 55\}$ and the private keys: $\{d = 37, p = 5, q = 11\}$.
Let $m = 49$ and Bob first encrypts this plain text using the encryption exponenet e and n from the public keys: $c \equiv m^e = 49^{13} \equiv 4 \pmod{55}$.
Now Alice gets the ciphertext $c = 4$ and she uses the decryption exponent d from her private key to decipher the text: $m \equiv c^d = 4^{37} \equiv 49 \equiv m \pmod{55}$.
Thus Alice gets the correct plain text from Bob through RSA encryption and decryption.

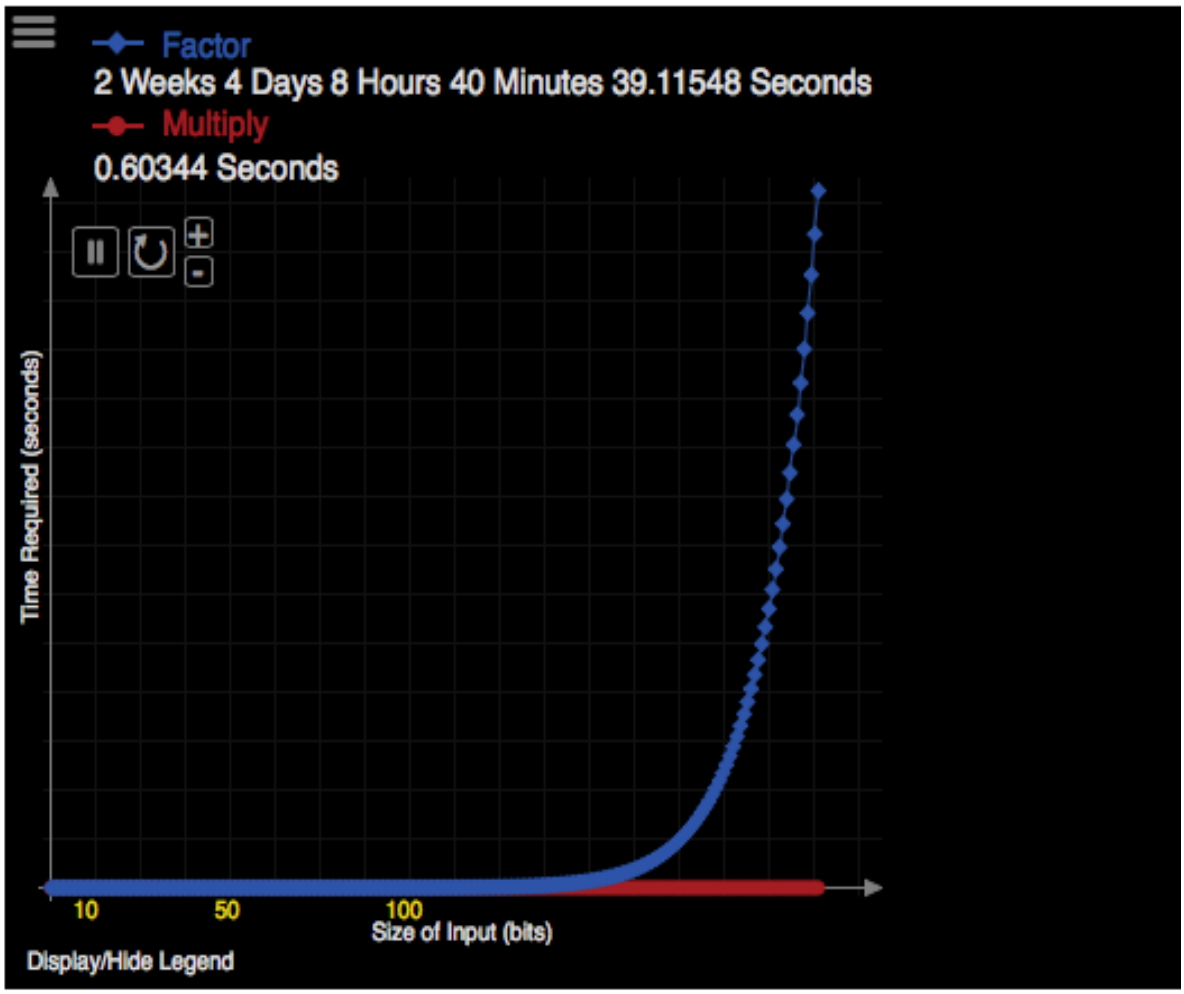
2.5 Why RSA is Safe

In the previous section we explained that RSA encryption is efficient for the desired audience/receiver who has the information of the private key and the decryption exponent, since in the example we have shown that it's a one-step squaring and modulo calculation (a large number would take longer to compute, but still a constant number of step).

Then what about the general public who don't know the decryption exponent and the private keys? The answer is that they would most likely not going to crack the cipher text.

Suppose Emily somehow gets the ciphertext from Bob, $c = 4$, but she doesn't know d, p or q but $e = 13$ and $n = 55$. To get d, she first needs to know $\phi(n)$ which depends on $p - 1$ and $q - 1$, where p and q are the prime factors of n. In this case specifically, $n = 55$, it's fairly easy to compute the possible pairs of two prime factors and then compute d from $\phi(n)$. This is why in real-world application, someone would never choose two prime numbers too small and trivial to encrypt the plain text. As a matter of fact, if n is relatively large composite, it takes countless effort and time to get p and q. While such factorization isn't impossible, it certainly is hard, as the difficulty of factoring is based on and depends on the size of the two prime numbers. In fact, factorization of n costs $O(e^{log(n)^3})$ run-time complexity.[23] Therefore, when n is large enough, Emily will have a hard time finding p and q and thus $\phi(n)$ and d. The following graph of run-time complexity analysis compares the run-time complexity of multiplying and factoring a prime number with respect to the size of that prime number (in bits):

[23] "Public Key Cryptography: RSA and Lots of Number Theory." Winlab.rutgers.edu



[24]

The mathematical principle behind this is the fact that prime multiplication is easy, but prime factorization is hard, especially factoring a large number. This principle is essentially what makes RSA encryption a one-way function, which is Rivest's breakthrough on that night in April 1966.

3 Further Exploration of RSA

3.1 Attacks in History

There has been attacks to the RSA encryption system. One famous one is called Wiener's attack and also known as Low Private Exponenet attack since cryptologist Michael J. Wiener shows that a small decryption exponenet d could result in a total collapse of the RSA cryptosystem.[25]

[24] "Time Complexity of Modern Cryptography." Khanacademy.com
[25] "Twenty Years of Attacks on the RSA Cryptosystem." Dan Boneh.

Wiener's attack can be summarized as a theorem: [26]

$$\text{Let } N = pq \text{ with } p < q < 2p. \text{ Let } d < \frac{1}{3}N^{\frac{1}{4}}.$$

Given (N, e) with $ed \equiv 1 \pmod{\phi(N)}$ Marvin can efficiently recover d.

Proof. (The following proof is a complete citation from "Twenty Years of Attacks on the RSA Cryptosystem.") [27]
Given $ed \equiv 1 \pmod{\phi(N)}$, according to the Extended Euclidean Algorithm, there exists a k such that $ed - k\phi(N) = 1$. Therefore:

$$|\frac{ed - k\phi(N)}{d\phi(N)}| = \frac{1}{d\phi(N)}$$
$$|\frac{e}{\phi(N)} - \frac{k}{d}| = \frac{1}{d\phi(N)}$$

Based on approximations using continued fractions, $\frac{k}{d}$ is an approximation of $\frac{e}{\phi(N)}$, thus Marvin may use N to approximate $\phi(N)$.
Since $\phi(N) = N - p - 1 + 1$ and $p + q - 1 < 3\sqrt{N}$, we have $|N - \phi(N)| < 3\sqrt{N}$.
Replace $\phi(N)$ with N, we obtain:

$$|\frac{e}{N} - \frac{k}{d}| = |\frac{ed - k\phi(N) - kN + k\phi(N)}{Nd}$$
$$= |\frac{1 - k(N - \phi(N))}{Nd} \leq |\frac{3k\sqrt{N}}{Nd} = \frac{3k}{d\sqrt{N}}$$

Now, $k\phi(N) = ed - 1 < ed$. Since $e < \phi(N)$, we see that $k < d < \frac{1}{3}N^{\frac{1}{4}}$. Hence we get:

$$|\frac{e}{N} - \frac{k}{d}| \leq \frac{1}{dN^{\frac{1}{4}}} < \frac{1}{2d^2}.$$

Then Marvin just needs to compute the log$N$ convergents of the continued fraction for $\frac{e}{N}$. One of these will eventually equal $\frac{k}{d}$. Since $ed - k\phi(N) = 1$, we have gcd(k,d)=1, and hence $\frac{k}{d}$ is a reduced fraction. This is linear-time algorithm for recovering the secret decryption exponent d. □

From this we can see that a low decryption exponent may lead to system attack. Therefore, it's suggested that d, the decryption exponent must be at least 256 bits long in order to avoid this attack, since typically N is 1024 bits. [28]

Moreover, we can see that even though RSA has been the best cryptosystem there has been, it still has potential flaws that can lead to information insecurity. Researchers and developers are still working on making necessary implementations to fix the bugs that have been found by the attackers over the past few decades.

[26] "Twenty Years of Attacks on the RSA Cryptosystem." Dan Boneh.
[27] "Twenty Years of Attacks on the RSA Cryptosystem." Dan Boneh.
[28] "Twenty Years of Attacks on the RSA Cryptosystem." Dan Boneh.

4 References and Bibliography

1. “Cryptography.” Merriam-Webster.com. 2015
http://www.merriam-webster.com/dictionary/cryptography
2. Hw6.pdf. Tabitha Peck. CSC121. Davidson College
3-4. “Caesar Cipher.” Wikipedia.com. 2015
http://en.wikipedia.org/wiki/Caesarcipher
5. “Imitation Game.” Nora Grossman. Ido Ostrowsky. Teddy Schwarzman. (Producer), Morten Tyldum. (Director). (2014). United States: Black Bear Pictures. Bristol Automotive.
6. ”Symmetric Key Cryptography (Non-Technical).” abblcabandlng.wordpress.com
https://abbicabanding.wordpress.com/2009/04/30/symmetric-key-cryptography-non-technical/
7-8. “The RSA Cryptosystem: History, Algorithm, Primes.” Michael Calderbank. University of Chicago. Math Department.
http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Calderbank.pdf
9. ”Diffie-Hellman Key Exchange.” Stackoverflow.com
http://stackoverflow.com/questions/28015433/is-it-possible-to-hack-diffie-hellman-by-knowing-the-prime-number-and-the-gene
10.“The RSA Cryptosystem: History, Algorithm, Primes.” Michael Calderbank. University of Chicago. Math Department.
http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Calderbank.pdf
11. “A Method for Obtaining Digital Signatures and Public- Key Cryptosystems”. Ron Rivest, Adi Shamir, Leonard Adleman. Drexel University. Computer Science Department. 1977.
12. http://www.usc.edu/dept/molecular-science/RSApics.htm
13-17. “The RSA Cryptosystem: History, Algorithm, Primes.” Michael Calderbank. University of Chicago. Math Department.
http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Calderbank.pdf
18-19. “A Method for Obtaining Digital Signatures and Public- Key Cryptosystems”. Ron Rivest, Adi Shamir, Leonard Adleman. Drexel University. Computer Science Department. 1977.
20-21. “The RSA Cryptosystem: History, Algorithm, Primes.” Michael Calderbank. University of Chicago. Math Department.
http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Calderbank.pdf
22. “Public-Key Cryptography for Dummies (RSA Version).”
http://dengfengli.com/?p=6
23. “Public Key Cryptography: RSA and Lots of Number Theory.”
http://www.winlab.rutgers.edu/ trappe/Courses/AdvSec05/RSAandNumTheory.pdf
24. “Time Complexity of Modern Cryptography.”
https://www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/p/time-complexity-exploration
25-28.“Twenty Years of Attacks on the RSA Cryptosystem.” Dan Boneh.
http://crypto.stanford.edu/ dabo/papers/RSA-survey.pdf