A brief history and mathematical deconstruction of:

Ciphers and the RSA Cryptosystem

Kyle Ottmann
Cole Polychronis

Discrete Mathematics
Westminster College
November 24, 2015

Summary

In this paper, we will be exploring the RSA cipher, which is currently one of the most secure and preferred methods of encryption of online data. We will do so by first tracing the major development of ciphers during WWII, and then trace how these developments led to significant commercial interest around the world following the war. Then we will proceed to explore the RSA cipher, the culmination of this commercial interest and the current standard for online encryption. We will explore the RSA cipher by first giving a brief history of its inception at MIT and then by delving into the high-level mathematics and algorithms, which constitute the actual method by which RSA ciphers encrypt data. This math will include using modular arithmetic, finding a totient, and using the Euclidian and extended Euclidian algorithms. In addition to this exploration, we will share our experience in developing a small Java program which makes use of the general principles of the RSA cipher for the purpose of providing small, easy to understand examples of how RSA functions.

Introduction

The primary goal goals of this paper are twofold. The first goal is to provide a brief, but informative history of the development of ciphers that occurred in World War II, which prompted commercial interest leading to the development of the RSA cipher, which we will give a very brief history of. The second goal of this paper is to explain the complicated mathematics that drive the RSA cipher, especially that of the Euclidian and Extended Euclidian Algorithm. In order to do this more effectively and in a way that is more easily understandable, we will create a simple example which will outline every step of the process in creating an RSA cipher. Finally, we will briefly mention our experience in coding a RSA cipher program in Java which will make use of a GUI to demonstrate the mathematical values present in an RSA cipher. Ultimately, RSA ciphers are a worthy topic for study in an academic setting because they represent an incredibly important aspect of cyber security. Cyber security itself is an incredibly lucrative industry, expected to be worth $ 75 million by the end of 2015, and estimated to be worth as much as $ 170 million by the year 2020 [6].

Background

In order to analyze RSA ciphers, there are several key terms and concepts worth defining and reviewing. First and foremost is the concept of encryption and decryption. Simply, encryption is the process of using a certain pattern to convert a message to seemingly meaningless characters and decryption is the process that reverses encryption and converts the random series of characters back into the original message. This decryption is often done through use of a key, which is in essence the algorithm or pattern which describes the manner in which the information was

encrypted. RSA ciphers are unique in that they are the first asymmetric ciphers, which means that they make use of two keys. Each key can only either encrypt or decrypt a message, but not both. This principle is shown later on in Figure 1.

Development of Ciphers in World War II

While ciphers have been recorded being used in warfare all the way back in Roman Times, ciphers experienced incredible development during World War II. This cryptographic arms race began when the German Axis powers developed a machine the likes of which the world had never seen before, a machine that would later be called Enigma. This machine produced incredibly complex (for the time) encryption of sensitive data and while it was put to little use at first, by the 1930's the Enigma was widely used by the German navy, airforce, and army [20]. While a team of mathematicians and cryptographers based in Bletchley Park in England were finally able to crack the Enigma Codes, the world realized that cryptography had taken a new turn in complexity and became a key focus of study during the war.

A unique response to the increasing developments in cryptography came from the United States in the form of the Navajo Code Talkers. While not technically a form of encryption, the Navajo language provided a unique answer to the growing sophistication of code breaking algorithms, as the Navajo language has no comprehensive written form and therefore, the message can only be delivered and interpreted by speakers of the Navajo Language [15]. However, the Navajo Code Talkers exhibited a problem that was shared by all ciphers and cryptography of that time: the process, machine, or human being responsible for the encryption and decryption had to be physically

transferred from one place to another in order to be used across any distance. This physical movement meant that the cipher key (in any form) merely had to be intercepted to be cracked. This prompted significant research into other systems that wouldn't require the transfer a key. However, this research became less of a political focus in the United States upon the end of the war.

History of RSA Ciphers

While government interest in new ciphers was put aside in the wake of World War II, data security became of great interest to corporations around the world. During this period, many more advancements were made in the area of cryptography, they were all centered around the use of symmetric encryption. It wasn't until the The RSA cipher was developed by MIT students Ron Rivest, Adi Shamir, and Leonard Adleman in 1977 that the world was introduced to a new cryptographic method; asymmetric encryption. The RSA cipher is credited with the first public key cryptosystems, otherwise known as asymmetric cryptography [3], a system which involves the creation of two keys: a public key and a private key. An individual's public key is accessible to anyone, and can be used to encrypt a message to be delivered to that individual. However, that individual's private key is not shared with anyone, as well as the private key is the only key that can decrypt a message sent to the individual. This is incredibly powerful, as the key that decrypts a message is never transferred between people, and is therefore far more secure. This process is shown graphically in Figure 1 below.

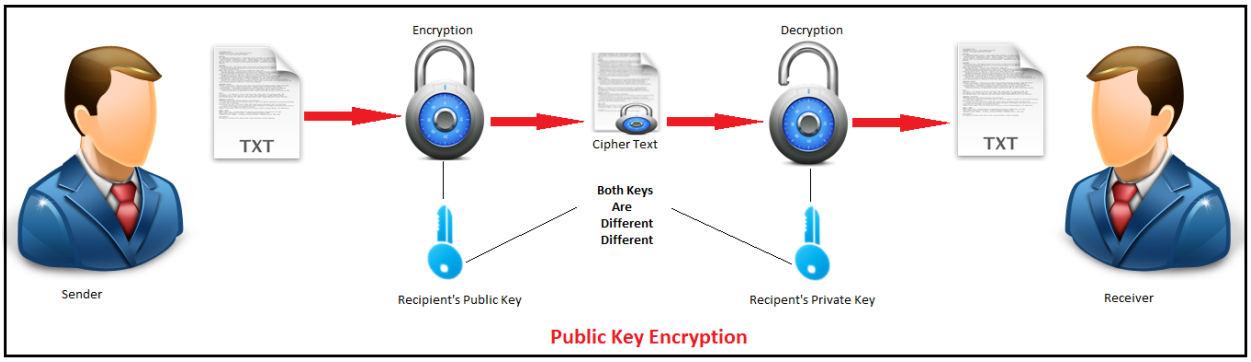

Figure 1: Public Key Encryption [3]

RSA ciphers represent an incredibly important aspect of cyber security, as RSA ciphers and ciphers derived from RSA are the current ways that sensitive information is protected online, such as protecting emails, purchase information, and even sensitive government documents.

The Mathematics Behind RSA Ciphers

As RSA ciphers deal with some complicated mathematical principles such as reversing through a modular operand via the Extended Euclidian Algorithm, we found that it would be best to iterate through each step of the RSA cipher process by using one, simple example, maintained throughout the whole process to illustrate each step. From this point forward, we will give a brief explanation of each step to creating an RSA cipher and then demonstrating that step.

Step One:

The first step in creating an RSA cipher is to pick two large prime numbers. Typically, these prime numbers must be a minimum of 1024 digits to be considered

secure and most RSA ciphers that are currently implemented are actually 2048 digits long. These prime numbers are so large that mathematicians can't say with certainty that they are in fact prime. To this end, there exists Rabin-Miller Primality Test, which given a large number, can predict with high probability whether a number will be prime or not [22]. While this is an incredibly fascinating topic that deals with very abstract mathematical principles, it is well beyond the scope of ourselves and our peers. So for the sake of simplicity, we will ignore the the first step, in that we will be picking two very small numbers. Thus, we arrive at:

Prime Number ***P*** : 5

Prime Number ***Q*** : 13

Step Two:

The next step in creating an RSA cipher is to find the modulus ***n*** to be used in the rest of the algorithm. Simply, ***n*** is found by multiplying out two prime numbers ***P*** and ***Q***. In the real world, because the two prime numbers used to find our modulus are incredibly large, the modulus would be even more so and therefore very difficult to factor. While our example will not be nearly as difficult, it will still adequately show the process.

$n = P * Q = 5 * 13 = 65$

Thus: $n = 65$

Step Three:

The third step in the process is finding the totient, symbolized as ϕ (phi). The totient is based around the principle that for any prime number ***n***, every number from 1 to ***n - 1*** has a greatest common denominator (which we will from now on refer to as gcd) of 1 with the prime number ***n***, which therefore means it has a multiplicative inverse in ***modulus ϕ(n)*** [22], which will be described in greater depth in **Step 5**.

ϕ(n) = ***ϕ(P)*** * ***ϕ(Q)*** = (***P*** - 1) * (***Q*** - 1) = (4) * (12) = 48

Thus: ***ϕ(n)*** = 48

Step 4:

The fourth step in the process is determining the public key, generally expressed as ***e***. To find ***e***, we simply pick a number in the range from 2 to ***ϕ(n)*** (exclusive), that has a gcd of 1 with ***ϕ(n)***.

Let ***e*** = 11

The public key is in fact a pair, in which ***e*** represents an exponent and ***n*** is the modulus. Thus the public key is actually represented as (***e***, ***n***).

Thus: Public Key = (11, 65)

Step 5:

The fifth and final step to the process is determining the private key, represented as ***d***, which is also the most difficult. This step is purposely complex, as the the private key needs to be secure and should therefore not be easily determined. This step can be broken up into two smaller steps. In essence, we must solve the equation

e * ***d*** (mod ***ϕ(n)***) = 1 for ***d***.

Step 5A:

By substituting what we have found in previous steps, we thus know that we are trying to solve the equation 11 * ***d*** (mod 48) = 1 by finding ***d***. In order to find ***d*** by using the Extended Euclidian Algorithm, we must first use the Euclidian Algorithm to rewrite the equation 11 * ***d*** (mod 48) = 1. In essence, the Euclidian Algorithm finds the gcd of two number by subtracting the smaller number from the larger number and repeating this process, as the gcd for these smaller numbers is the same as the gcd for the original numbers. This process is repeated until the remainder is 1. This is best demonstrated by the example below.

The Euclidian representation of 11 * ***d*** (mod 48) = 1 is: **48x + 11y = 1**

We then set 48 equal to its smaller components, in terms of the smaller number 11 and a remainder: **48 = 4(11) + 4**

We then repeat the process, setting 11 equal to its smaller components, in terms of the smaller number 4 and a remainder: **11 = 2(4) + 3**

We repeat this process once again: **4 = 1(3) + 1**

We can now stop, as the remainder is 1.

Step 5B:

Now that we have performed the Euclidian Algorithm on our modular equation, we can now use the Extended Euclidian Algorithm to find what ***d*** equals.

To use the Extended Euclidian Algorithm, we start at the "bottom" of **Step A**,

rearranging the bottom equation to be equal to 1.

This gives us: **1 = 4 - 1(3)**.

We want this equation in terms of our two original terms (11 and 48), so we next solve our "second from the bottom" equation of **Step A** for 3, so that we can

substitute it the equation we arrived at in our first iteration of the Extended Euclidian Algorithm. Solving the “second from the bottom” equation for 3, we get: **3 = 11 - 2(4)**.

Substituting this into the first equation we got from **Step A**, we get:

1 = 4 - (11 - 2(4)), and combining like terms, we get: **1 = 3(4) -1(11)**

Repeating this process once again, solving for 4, we get from the top equation in **Step A** that: **4 = 48 - 4(11)**.

Substituting this into our equation once again, we get: **1 = 3(48-4(11)) - 1(11)**, and combining like terms, we get: **1 = 3(48) - 13(11)**.

By applying (mod 48) to our equation, we know that 3(48) (mod 48) is 0, so we are left with -13 * 11 (mod 48) = 1. However, we can’t have negative numbers as a key, so we must use the properties of modular equivalence classes to find an equivalent positive number to -13. Such a number is 35. Thus, we can say 35 * 11 (mod 48) = 1.This is in the same form as our previous equation 17 * ***d*** (mod 35) = 1, so therefore we know ***d*** = 35.

Thus: ***d*** = 35

Similar to our Public Key, the Private Key is actually a key pair, where ***d*** represents an exponent and ***n*** represents a modulus. Thus the Private Key is actually represented as (***d***, ***n***).

Thus: Private Key = (35, 65)

Java Experience

Above, we have just shown what an applied process it is to find the Public and Private Keys for an RSA cipher, even when incredibly small (in comparison) prime numbers are used. As we strived to come up with the example to show the RSA cipher process in this paper, we realized that a tool to help display all this information through a few simple inputs would be incredibly helpful. This inspired us to write a small Java program, which implemented a GUI. This program would simply take in two small prime numbers and display ***n,*** the totient ***ϕ(n)***, and the ***Public*** and ***Private Keys***. While this program is limited to small numbers, and therefore not meant for practical use dealing with numbers thousands of digits long, this program will serve as a great teaching tool to simply demonstrate the basic principles of an RSA cipher, as well as to help guide the exploration of RSA.

Conclusion

From World War II encryption methods to the present day RSA algorithm, the underlying principle has remained unchanged - to secure message transmission. That is not to say that the methods used for encryption haven't changed, we have seen remarkable advancements in cryptography from the symmetric encryption used in the German Enigma to the more modern and secure encryption methods used now. RSA is unique in the respect that it uses asymmetric encryption, providing a public and private key allowing for description only by the intended recipient. The unique algorithm behind this mathematics marvel is supported by the Euclidian and Extended Euclidian

Algorithm as we saw earlier in this paper. Rivest, Shamir, and Adleman pioneered the public key cryptosystems that now guard our computers and protect sensitive data, no easy feat in today's attack prone world. This makes it an advantageous field of study for those in the mathematics or computer science disciplines. There is no doubt why RSA plays such a significant role in the now multi-million dollar cyber security industry.

Throughout this paper we have demonstrated two learning objectives, the understanding of rigorous mathematics and an original thought. Our rigorous mathematics are displayed in the Mathematics Behind RSA Ciphers section where we demonstrate the use of the Euclidian algorithm and its related and supportive algorithms. Original thoughts have been revealed in our java code allowing us to demonstrate the mathematics behind RSA on a smaller, easier to grasp scale.

References

1. American Cryptogram Association, Cipher Types (2005); available at http://cryptogram.org/cipher_types.html, accessed on October 07, 2015.

2. BBC, “How the Modern World Depends on Encryption,” October 25, 2013, available at http://www.bbc.com/news/technology-24667834, accessed on October 07, 2015.

3. C# Corner, RSA Algorithm With C# (2013); available at http://www.c-sharpcorner.com/UploadFile/75a48f/rsa-algorithm-with-C-Sharp2/, accessed on November 04, 2015.

4. Yan Chen, “Cryptography: asymmetric encryption (RSA)” (2005), available at http://www.cs.northwestern.edu/~ychen/classes/cs395-w05/lectures/class3.ppt, accessed October 07, 2015.

5. Yan Chen, “Cryptography: symmetric encryption (DES/AES algorithms)” (2005), available at http://www.cs.northwestern.edu/~ychen/classes/cs395-w05/lectures/class2.ppt, accessed October 07, 2015.

6. CSO, “Worldwide cybersecurity market continues its upward trend,” July 09, 2015, available at http://www.csoonline.com/article/2946017/security-leadership/worldwide-cybersecurity-market-sizing-and-projections.html, accessed on October 07, 2015.

7. Dr Michael Evans, AMSI, “Math Delivers! RSA Encryption” (2013), available at http://www.amsi.org.au/teacher_modules/pdfs/Maths_delivers/Encryption5.pdf, accessed on November 04, 2015.

8. Greg Goebel, Vectors, Introduction to Codes, Ciphers, & Codebreaking (2014); available at http://www.vectorsite.net/ttcode_01.html, accessed on October 07, 2015.

9. Maria D. Kelly, “The RSA Algorithm: A Mathematical History of the Ubiquitous Cryptological Algorithm” (2009), available at http://www.sccs.swarthmore.edu/users/10/mkelly1/rsa.pdf, accessed on November 03, 2015.

10. Khan Academy, Ciphers vs. Codes (2015); available at https://www.khanacademy.org/computing/computer-science/cryptography/ciphers/a/ciphers-vs-codes, accessed on October 07, 2015.

11. Khan Academy, Modern Cryptography (2015) available at https://www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/, accessed on October 21, 2015.

12. John Matson, Scientific American, Record 232-digit number from cryptography challenge factored (2010); available at http://blogs.scientificamerican.com/observations/record-232-digit-number-from-cryptography-challenge-factored/, accessed on November 04, 2015.

13. Nicholas G. McDonald , “Past, Present, and Future Methods of Cryptography and Data Encryption” (2009), available at http://www.eng.utah.edu/~nmcdonal/Tutorials/EncryptionResearchReview.pdf, accessed on October 07, 2015.

14. Kate Mulcahy, Listverse, 10 Codes & Ciphers (2012); available at http://listverse.com/2012/03/13/10-codes-and-ciphers/, accessed on October 07, 2015.

15. Navajo Code Talkers, Cryptography (2015); available at http://navajocodetalkers.org/category/cryptography/, accessed on October 21, 2015.

16. Abderrahmane Nitaj, “The Mathematical Cryptography of the RSA Cryptosystem” (2012), available at http://www.math.unicaen.fr/~nitaj/RSAnitaj1.pdf, accessed on November 21, 2015.

17. Martin Ouwehand, École polytechnique fédérale de Lausanne, The (simple) mathematics of RSA (2001); available at http://certauth.epfl.ch/rsa/, accessed on November 21, 2015.

18. Real Time Logic, Certificate Management for Embedded Systems (2013); available at https://realtimelogic.com/blog/2013/10, accessed on November 03, 2015.

19. Jake Salterberg, "An Introduction to the RSA Encryption Method" (2012), available at http://math.arizona.edu/sites/math.arizona.edu/files/webfm/undergrad/uta/Spring12UTATalkSalterbergJake.pdf, accessed on November 03, 2015.

20. ShoreTel, Encoded Communications of World War II (2015); available at https://www.shoretel.com/encoded-communications-world-war-ii, accessed on October 07, 2015.

21. Huzaifa Sidhpurwala, Red Hat Security Blog, A Brief History of Cryptography (2013); available at https://securityblog.redhat.com/2013/08/14/a-brief-history-of-cryptography/, accessed on October 07, 2015.

22. Barry Steyn, Doctrina, How RSA Works with Examples (2015); available at http://doctrina.org/How-RSA-Works-With-Examples.html, accessed on November 03, 2015.

23. Word Games, Cryptoquote (2010); available at http://www.eastoftheweb.com/cgi-bin/top_scores.pl?game=cryptoquote, accessed October 08, 2015.