Najalia Singh

Applying Blankenship's Method in the RSA Cryptosystem to Test Information Security

Abstract:

The RSA cryptosystem picks keys and uses large prime numbers as an essential tool to allow me to encrypt and decrypt messages. In my research I studied certain linear Diophantine equations and showed how they are used in RSA to encrypt and decrypt information. I also used a technique created by Blankenship, that is rooted in linear algebra and employs matrix row reduction techniques, to efficiently calculate encryption and decryption keys. I employed the Maple software to simulate data and illustrate RSA in action. Using an application of Blankenship's Method enabled me to calculate the decryption key. I then illustrated a way to break RSA for small primes and identified the weaknesses in RSA for larger primes.

Introduction:

Storage of a vast collection of data and all aspects of information heavily relies on technology. Information is always at risk of being hacked, which can compromise this data. An important encryption scheme, RSA, that was published in 1977 by **R**ivest, **S**hamir and **A**dleman, is used to secure Information. Bounds established on the primes p and q made RSA effective and secure. The ease of breaking the codes for various prime pairs (p, q), for small and large pairs, gauged secure RSA was effective.

One can think of the RSA cryptosystem as a numerical encryption system scheme that is used to secure information exchanged between two individuals. Consider, for example, a sender named Bob who wants to send a message to a receiver named Alice. The Encryption Key is denoted by Kb, while the Decryption Key is denoted by Lb. If an eavesdropper, called Eve, wants to obtain the message that only Alice should receive, Bob will have to use a simple algorithm and encrypt the message by using an algorithm. This algorithm is intended for Alice to

decrypt the message with a special key only she has. Eve, on the other hand, will be challenged to crack that encryption since she does not have Alice's key. (Nelson , 2014) Keeping the algorithm secret is not the most critical part of ensuring security; it is the key that must be kept secret.

Other security cryptosystems have been created before RSA, such as the Caesar shift cipher and the Substitution cipher, which are both private-key systems. The Caesar shift cipher is easily compromised by exhausting all the keys, and moreover, this can be done without a computer. The Caesar shift cipher's code is easily broken because there are only twenty-six letters in the alphabet and there is a limit to how those letters can be arranged in a cipher text. The Substitution cipher is readily broken by statistical methods because the probability of a letter occurring on a page is easily determined.

Methods/Materials:

I used the Maple software to simulate data to create various keys. I saw that an essential step in the process requires solutions to linear Diophantine equations of the form ax+by=c, where a, b, c are integer constants and the only allowed solutions for x and y are integers. I employed the special technique known as the Blankenship method to solve ax+by=c. Excel created the chart for p, q, n, m, d, c1, c2 (encrypt and decrypt). RSA is used to send numerical encryption and decryption messages, but I can send text messages by assigning the letters from A to Z with the values 01 to 26.

The setup for RSA involves the following steps:

- Choose two distinct prime numbers, say p and q that are large
- Let n=pq and m=(p-1)(q-1)

- Pick 1<e<m, such that gcd(e,m)=1
- Solve ed =1(mod m) to find d with the restriction that 0<=d<m
- Publish the public key (e,n)
- Keep secret the private key (d,n)
- Forget p and q

To Encrypt a message:

- For example: MAT = 130120
- Look up the public key (e,n). From table I will use (101, 136627)
- Compute M^e mod n to obtain the ciphertext (c) to send
- M < n, to send MAT I will break it up into two parts 130 and 120
- Compute each ciphertext c1 = 130^101 mod 136627 and c2 = 120^ 101 mod 136627 separately, i.e. (c1, c2) = (64397, 83794)

To Decrypt a message:

- Use your secret key (d,n). (d,n) =(100901,136627)
- Reduce c^d mod n to decrypt the message. In this example I have

  c1^d mod n = 64397^100901 mod 136627 and

  c2^d mod n = 83794^100901 mod 136627 have c1 = 130 and c2 = 120, the message is 130120 = MAT
- I have c1 = 130 and c2 = 120, the message is 130120 = MAT

I showed how to Encrypt and Decrypt data with appropriate examples using modular techniques.

Results:

For large data sets, encryption can be very slow and may need outsourcing, which makes easier for it to be compromised. Theoretically I have the mathematical means of cracking RSA, but the choice of large primes renders this approach to be impractical. RSA is secure because n is the product of 2 large primes which has over 200 decimal digits. By computing the solutions to the congruence: ed ≡1(mod m) (0<=d<m), and by rewriting and solving the Diophantine Equation: ed + my = 1 for e using the simulated data in figure 1, then makes solving for d faster. This Blankenship Method is easier and faster to use because of the inherent properties of matrices, and it offers a way to break RSA. At the same time, it is still lengthy. The other algorithms, some of which use the extended Euclidean method, are more cumbersome when used to solve the congruence for d.

p	q	n = p*q	m = (p-1)*(q-1)	e, gcd(e,m)=1, 1<e<m	d, ed≡1 mod m	c1≡130 ^ e mod n	c2≡120 ^ e mod n	c1^d mod n	c2^d mod n
19	23	437	396	101	149	28	435	130	120
31	53	1643	1560	101	1421	1266	1453	130	120
59	41	2419	2320	101	781	936	372	130	120
67	17	1139	1056	101	941	520	273	130	120
73	107	7811	7632	101	4685	1861	3460	130	120
113	131	14803	14560	101	11821	3274	11530	130	120
263	109	28667	28296	101	22973	14921	9278	130	120
431	317	136627	135880	101	100901	64397	83794	130	120
167	61	10187	9960	101	5621	2875	5708	130	120
271	31	8401	8100	101	401	3188	7343	130	120
223	71	15833	15540	101	10001	9993	7363	130	120
89	113	10057	9856	101	8685	9715	9383	130	120
457	97	44329	43776	101	23405	32559	11063	130	120

Figure 1: Equations to encrypt and decrypt a message , and how I got the value of e (public/encryption key) and d (private/decryption key).

$$ed \equiv 1 \bmod m \text{ or } ed + my = 1$$

$$\begin{pmatrix} e & 1 & 0 \\ m & 0 & 1 \end{pmatrix} \xrightarrow{\text{row reduce}} \begin{pmatrix} 1 & d & y \\ 0 & * & * \end{pmatrix}$$

For $e = 101$ and $m = 396$, solve $101d + 396y = 1$

$$\begin{pmatrix} 101 & 1 & 0 \\ 396 & 0 & 1 \end{pmatrix} \xrightarrow{\text{row reduce}} \begin{pmatrix} 1 & 149 & -38 \\ 0 & -396 & 101 \end{pmatrix}$$

- Set up and row reduce the 2 by 3 matrix in terms of e and m until the first number in any row is zero
- Read off the answers for d and y

Figure 2

Analysis and Conclusion:

I can break RSA by factoring out n in terms of its two prime factors and then compute the secret key. Currently, all the known algorithms to factor numbers into their prime factors (large magnitude) are not possible to compute in a reasonable time. I can easily compute the decryption key for small pairs of primes. I employed Blankenship's method which uses matrix computations, (efficient and reduces large numbers easily and quickly by division), to get to the decryption key. Blankenship's method allows for a wider range of numbers that I can use to find the decryption keys. However, with the current size of primes chosen in practice, it is again impractical to decrypt in a reasonable time. Blankenship's method also allows users to create additional algorithms, which can extend the range of numbers that can be factored, and forces users to extend the size of primes chosen to ensure data security. Concerns I have when I use RSA are knowing the sender's identity and ensuring the message is not fraudulent.

When encrypting and decrypting messages in RSA, computations are extremely difficult when n is very large even with a computer. RSA is secure for extremely large primes (more than 200 digits), because I am limited by the ability of computers to do computations at extremely

high levels of precision for the decryption key d**.** I used Blankenship's method, an extended form of the Euclidean algorithm, to find d. Blankenship's Method allows me to calculate the decryption code quickly and efficiently for small primes. This is equivalent to solving linear Diophantine equations, linear algebra and matrix reduction. The study also shows how these different branches of mathematics interact. My studies also introduced me to the cutting-edge field of quantum computers, which I am eager to explore in the future. With the advent of cutting-edge technology such as quantum computers, RSA will become unsafe because quantum computers will break RSA's code in seconds. Current computers encode data using binary bits; quantum computers use qubits (based on the electron's spin) in multiple electron states with complex mathematics. It is not expected for quantum computers to be commercially available for encryption; they are susceptible to thermal and electromagnetic interference and because affordability is also a major consideration. Currently mathematicians are working on "quantum resistant" encryption algorithms to secure data. Blankenship's Method gives a readily accessible way to compute the decryption key and use the power of matrix row reduction over the integers. In addition, it allows me to identify weaknesses in RSA, and as a consequence, gives me a way to strengthen RSA by using large primes.

Works Cited:

A J, M., Van, O. P., & Vanstone, S. A. (1997). *Handbook of Applied Cryptography.* CRC Press.

Nelson , J. (2014, March 19). Intermediate Math Circles Cryptography I. Waterloo, Canada: Center for Education in Mathematics and Computing.

Rosen. (2011). *Elementary Number Theory and its applications, 6th Ed.* New Jersey: Pearson.

Vazzana, E. M. (2008). *Introduction to Number Theory* . CRC Press.

Washington, T. W. (2002). *Introduction to Cryptography with Coding Theory, 2nd Ed.* Pearson.