A Matrix Extension of the RSA Cryptosystem

Andrew Pangia

December 12, 2014

Abstract

We propose a variation on the RSA Cryptosystem: namely, an extension of the RSA encryption and decryption methods to matrix values in addition to scalars. We first explore the mathematics behind the RSA Cryptosystem, after which, we investigate the theory of the proposed variation to the system. The concept of extending the RSA Cryptosystem to apply to matrices originated in a paper released by IEEE in 2008 but the method given contained several errors. In our investigation, we correct those errors and establish limitations of the method which we have found.

1 Introduction

Throughout human history, a necessity to secretly exchange messages has existed. The methods of disguising the message are referred to as ciphers, or cryptosystems. Prior to being disguised, the message is referred to as the plaintext, while the disguising process itself is referred to as encryption and typically involves an integer or group of integers called a key. After being encrypted, the message is is referred to as a ciphertext, and the process of returning the ciphertext to the plaintext is referred to as decryption. In the past, cryptosystems were utilised primarily during military endeavors where a commander needed to communicate with his peers without having the opposition learning any information. In this current age of information, however, cryptosystems are far more ubiquitous, ranging from checking email, to making a purchase online, to withdrawing money from an ATM.

Some of the first ciphers were relatively simple, in which decrypting is simply reversing the encryption process. A typical example of an early cipher is the Caesar Cipher. This cipher begins by representing the desired character list in an integer system and selecting as a key a positive integer less than the number of characters. This integer must be known to both the sender and receiver. The sender converts his message into its integer form using the number correspondence and adding the key to each integer in congruence addition. The resulting values are converted to their corresponding characters and the resulting ciphertext is then sent. The receiver decrypts by converting the ciphertext to integers, subtracting the key from each integer, and converting back to character form. As an example, we use the English alphabet as our character list with the number correspondence in Table 1 (on page 2).

Suppose the plaintext *cryptography* is being sent (this example is to show the process of the cipher; we are not overly concerned with the applicability of the example). The sender selects the integer $k = 3$ as his key and converts *cryptography* to the integers $3, 18, 25, 16, 20, 15, 7, 18, 1, 16, 8, 25$

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16

q	r	s	t	u	v	w	x	y	z
17	18	19	20	21	22	23	24	25	26

Table 1: Alphabetic Correspondence

using Table 1. He converts each plaintext integer p using the congruence

$$c \equiv p + k \equiv p + 3 \pmod{26}$$

where c is the ciphertext value. Note that congruence modulo 26 is used since there are twenty-six characters in our list. The integer form of the ciphertext is $6, 21, 2, 19, 23, 18, 10, 21, 4, 19, 11, 2$, which corresponds to *fubswrjudskb*. The receiver recreates the plaintext by use of the congruence

$$p \equiv c - k \equiv c - 3 \pmod{26}.$$

In the past, simple ciphers such as the Caesar Cipher were sufficient to enable private communications. However, as technology has improved, the requirement for improved cryptosystems became apparent. To see the evidence of this need, observe that regardless of which key is used (or how many), the Caesar cipher can be cracked in a matter of microseconds using an exhaustive search method with even an older computer.

2 The RSA Cryptosystem

2.1 Theory

Developed in 1977 by Ronald L. Rivest, Adir Shamir, and Leonard Adleman, the RSA Cryptosystem is an asymmetric cipher; that is, the RSA system decrypts a message by use of a key other than the key used to encrypt the message [2]. This asymmetricity enables the user to publish his key and allow everyone to communicate with him, yet at the same time, prevents anyone from knowing what he has been told. The RSA system involves the receiver choosing two large prime numbers p and q and obtaining the integer $n = pq$. She next calculates $\phi(n) = (p-1)(q-1)$, where $\phi(n)$ is the number of positive integers less than or equal to n which are relatively prime to n. The receiver then selects an integer e such that $\gcd(e, \phi(n)) = 1$. The purpose of selecting the number e in this manner is to ensure that e^{-1} exists modulo $\phi(n)$. The receiver now computes $d \equiv e^{-1} \pmod{\phi(n)}$ by using the Euclidean Algorithm or another preferred method for computing inverses. The ordered pair (e, n) is made public (and is in fact referred to as the public key) while the ordered pair $(d, \phi(n))$ remains private (and is referred to as the private key).

To encrypt a message, the sender converts the plaintext message using a pre-arranged conversion method to an integer (or group of integers) modulo n, here denoted m such that $m < n$, and calculates the ciphertext $c \equiv m^e \pmod{n}$. In the case that the plaintext message is too large for one calculation, the sender simply breaks the message into components (in general i components) and apply the exponentiation to each submessage.

For decryption, the receiver computes $m \equiv c^d \pmod{n}$. Before proving that this decryption process is indeed successful, we state Euler's Theorem, which is integral to the RSA Cryptosystem (for a proof, see [2]).

Theorem 1. ***Euler's Theorem.*** *Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$. If $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.*

In addition to Euler's Theorem, the proof that RSA decryption results in the plaintext also requires the following lemma, which is stated in [3].

Lemma 1. *Let $x, y, a, b \in \mathbb{Z}^+$ such that $\gcd(a, b) = 1$. If $x \equiv y \pmod{a}$ and $x \equiv y \pmod{b}$, then $x \equiv y \pmod{ab}$.*

Proof. Let $x, y, a, b \in \mathbb{Z}^+$ such that $\gcd(a, b) = 1$. Suppose $x \equiv y \pmod{a}$ and $x \equiv y \pmod{b}$. We will show that $x \equiv y \pmod{ab}$. Since $x \equiv y \pmod{a}$, we deduce that $x = y + ka$ and since $x \equiv y \pmod{b}$, we deduce that $x = y + jb$ for some $j, k \in \mathbb{Z}$. By the transitive property of equality

$$y + ka = x = y + jb.$$

By the cancellation property of equality, we obtain that

$$ka = jb,$$

which means that $a|jb$. Furthermore, since $\gcd(a, b) = 1$, we conclude that $a|j$; hence, by definition of divides, $j = la$ for some $l \in \mathbb{Z}$. By substitution, we obtain that $x = y + jb = y + l(ab)$. Consequently by definition of congruence, we deduce that $x \equiv y \pmod{ab}$. □

We now proceed to prove that decryption in the RSA Cryptosystem does indeed result in the plaintext method (stated in the following theorem).

Theorem 2. *Let $n = pq$ where p and q are distinct prime integers, let $m \in \{0, 1, 2, \ldots, n-1\}$, $e, d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(n)}$, and let $c \equiv m^e \pmod{n}$. Then $c^d \equiv m \pmod{n}$.*

Proof. Let e, d, n, m be as defined above and let $c \equiv m^e \pmod{n}$. We will show that $c^d \equiv m \pmod{n}$. Observe that $c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$. Since $ed \equiv 1 \pmod{\phi(n)}$, there exists an integer k such that $ed = k\phi(n) + 1$. Then

$$m^{ed} \equiv m^{k\phi(n)+1} \pmod{n}.$$

Observe that by definition, $n = pq$ where p, q are distinct prime numbers. Then $\phi(n) = \phi(p)\phi(q)$ and $m^{k\phi(n)+1} = m^{k\phi(p)\phi(q)+1}$. Note that $\gcd(m, q) = 1$ or $\gcd(m, q) = q$. If m and q are relatively prime, then by Euler's Theorem $m^{\phi(q)} \equiv 1 \pmod{q}$ so

$$m^{k\phi(n)+1} = m^{k\phi(p)\phi(q)+1} \equiv (m^{\phi(q)})^{k\phi(p)} m \equiv 1^{k\phi(p)} m \equiv m \pmod{q}.$$

If $\gcd(m, q) = q$, then $q|m$, so $m \equiv 0 \pmod{q}$; hence

$$m^{k\phi(n)+1} \equiv 0 \equiv m \pmod{q}.$$

Consequently, $m^{k\phi(n)+1} \equiv m \pmod{q}$.

In the same exact manner, we obtain that $m^{k\phi(n)+1} \equiv m \pmod{p}$. Because p, q are distinct primes (and thus relatively prime), by Lemma 1 we conclude that $m^{k\phi(n)+1} \equiv m \pmod{pq}$, or $m^{k\phi(n)+1} \equiv m \pmod{n}$. Ergo, since $c^d \equiv m^{k\phi(n)+1} \pmod{n}$ and $m^{k\phi(n)+1} \equiv m \pmod{n}$, by transitivity we conclude that

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

□

The encryption key (e, n) is publicly known (so that all can send messages), whereas the decryption key $(\phi(n), d)$ is kept private. For decryption, the receiver exponentiates the ciphertext (the message after being encrypted) as demonstrated in the proof of the theorem to obtain

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

and reconverts m to the message in the pre-determined manner.

RSA Cryptography is not the quickest encryption method currently in use due to the processing time of calculating such large exponents; as such, RSA is more useful for transmitting small amounts of data. One method is to use RSA Cryptography to encrypt a symmetric key used in a swifter encryption method and send the symmetric key between communicants. This results in the symmetric key being reasonably well protected from interception while still allowing for fast communications.

Another use of the RSA cryptosystem lies in the verification of digital signatures. Under the digital signature method, a person encrypts a message (his name for example) for his friend by use of his own private decryption exponent. To verify that the message is indeed genuine, the receiver of the message applies the encryption exponent which corresponds to the supposed sender. If the message decrypts correctly, then the receiver knows that her friend did indeed send her the message.

2.2 Example

As an example of how the RSA Cryptosystem operates, take *cryptography* as our plaintext; we apply the correspondence stated with the Caesar Cipher.

The receiver selects prime integers

$$p = 503, \text{ and } q = 499,$$

which yields

$$n = pq = 250997.$$

Next, the receiver computes

$$\phi(n) = (p-1)(q-1) = 249996$$

and chooses her encryption exponent to be $e = 19$. Finally, she computes her decryption exponent to be $d = 210523$ where $ed \equiv 1 \pmod{\phi(n)}$. The encryption key which the receiver makes public is $(e, n) = (19, 250997)$.

If the sender wishes to send the message *cryptography* (as we assume he does for the purposes of our example), he obtains $m = 318251620150718011608 25$ using Table 1.

He breaks m into four submessages less than n to obtain

$$m_1 = 31825, m_2 = 162015, m_3 = 71801, m_4 = 160825.$$

The sender then calculates the following ciphertexts and forwards them to his friend.

$$\begin{aligned}
c_1 &\equiv {m_1}^e \equiv 31825^{19} \equiv 92363 \pmod{n};\\
c_2 &\equiv {m_2}^e \equiv 162015^{19} \equiv 13977 \pmod{n};\\
c_3 &\equiv {m_3}^e \equiv 71801^{19} \equiv 165966 \pmod{n};\\
c_4 &\equiv {m_4}^e \equiv 160825^{19} \equiv 56661 \pmod{n}.
\end{aligned}$$

The receiver calculates the plaintexts by computing

$$\begin{aligned}
m_1 &\equiv {c_1}^d \equiv 92363^{210523} \equiv 31825 \pmod{n};\\
m_2 &\equiv {c_2}^d \equiv 13977^{210523} \equiv 162015 \pmod{n};\\
m_3 &\equiv {c_3}^d \equiv 165966^{210523} \equiv 71801 \pmod{n};\\
m_4 &\equiv {c_4}^d \equiv 56661^{210523} \equiv 160825 \pmod{n}.
\end{aligned}$$

We chose to use the correspondence of a to 01, b to 02 and so on for the sake of simplicity. The American Standard Code for Information Interchange (ASCII), however, is frequently used as the numerical correspondence; ASCII has integers $\{0, 1, ..., 127\}$ to represent 128 characters, visible characters beginning at 32 (for 'space') and ending at 126 (for '$\sim$') with the remaining integers referring to computer commands (backspace, newline, etc.).

3 Matrix Modification

3.1 Prior Work

Next we consider modification to the RSA Cryptosystem which involves storing the plaintext inside a 2×2 matrix. This modification still utilises an integer $n = pq$ where p, q are distinct prime integers, but involves replacing the plaintext scalar value m with a 2×2 matrix, M and, rather than relying on the value $\phi(n) = (p-1)(q-1)$, instead relies on the value

$$(p^2-1)(p^2-p)(q^2-1)(q^2-q)$$

which is created from the product of the orders of $GL_2(\mathbb{Z}_p)$ and $GL_2(\mathbb{Z}_q)$, general linear groups of degree 2. A general definition of the General Linear Group is given below.

Definition 1. *The **General Linear Group** of order s, denoted $GL_s(\mathbb{Z}_n)$, is the monoid of invertible $s \times s$ matrices containing elements from $\mathbb{Z}_n$ with respect to multiplication such that the determinants of the matrices and n are relatively prime.*

Note that the identity matrix I is in $GL_s(\mathbb{Z}_n)$. Furthermore, matrix multiplication is associative for all matrices. Since the determinants of the matrices and n are relatively prime, every element of $GL_s(\mathbb{Z}_n)$ has a mulitplicative inverse in $GL_s(\mathbb{Z}_n)$. Consequently, $GL_s(\mathbb{Z}_n)$ is a group.

The inspiration for this method comes from [1], which proposed this method for all square matrices and stated the use of the General Linear Group. The exponentiation modulus proposed in [1] was taken to be

$$g = \prod_{k=0}^{s-1}(p^s - p^k) + \prod_{k=0}^{s-1}(q^s - q^k)$$

where s is the degree of the General Linear Group over $\mathbb{Z}_p$ and the General Linear Group over $\mathbb{Z}_q$; furthermore, [1] claimed that

$$\prod_{k=0}^{s-1}(p^s - p^k)$$

is the order of the General Linear Group of degree s. As an example, [1] took $p = 43$, $q = 47$. The encryption modulus is $n = pq = 2021$, while the exponentiation modulus is

$$g = (p^2 - p)(p^2 - 1) + (q^2 - q)(q^2 - 1) = 8111184.$$

The encryption exponent was chosen to be $e = 17$ such that $\gcd(e, g) = 1$ and the decryption exponent was found to be $d = 954257$.

When we refer to a matrix $A \pmod n$ for a positive integer n, we mean that each entry of A belongs to $\mathbb{Z}_n$. We take as an example

$$M = \begin{bmatrix} 13 & 1 \\ 20 & 7 \end{bmatrix}.$$

Note that the matrix corresponds to the plaintext *math*. Then, according to the method proposed in [1], the ciphertext is

$$C \equiv M^e \equiv \begin{bmatrix} 1473 & 884 \\ 1512 & 211 \end{bmatrix} \pmod n.$$

However, on decrypting this matrix, we obtain that

$$C^d \equiv \begin{bmatrix} 791 & 1460 \\ 906 & 115 \end{bmatrix} \pmod n.$$

Note that this matrix is not the original matrix M.

3.2 Conjecture

Aboud et al took their exponentiation modulus to be the sum of the orders of the general linear groups of degree s over $\mathbb{Z}_p$ and $\mathbb{Z}_q$. However, on noticing that this does not decompose to Euler's Phi Function for the 1×1 (scalar) case, we modified the exponentiation modulus to be

$$g = \prod_{k=0}^{s-1}(p^s - p^k) \cdot \prod_{k=0}^{s-1}(q^s - q^k).$$

To simplify investigation of our modification, we fixed $s = 2$ to obtain

$$g = (p^2 - p)(p^2 - 1)(q^2 - q)(q^2 - 1).$$

In order for our modified RSA system to be operable, we first prove that the order of $GL_2(\mathbb{Z}_n)$ be relatively calculable.

Theorem 3. *The order of the General Linear Group $GL_2(\mathbb{Z}_p)$ is given by*

$$|GL_2(\mathbb{Z}_p)| = (p^2-1)(p^2-p),$$

where p is a prime integer.

Proof. Let $S = \left\{ \begin{bmatrix} a & b \\ c & d \end{bmatrix} : a,b,c,d \in \mathbb{Z}_p \right\}$. Note that $GL_2(\mathbb{Z}_p) \subset S$. We will count the number of matrices in $GL_2(\mathbb{Z}_p)$ by subtracting the number of matrices not in the linear group from the total possible number of matrices in S. Let $A \in S$; since there are p choices for each entry in A, we deduce that $|S| = p^4$. Let k denote the number of matrices not in $GL_2(\mathbb{Z}_p)$; then there will be $p^4 - k$ matrices in $GL_2(\mathbb{Z}_p)$. Suppose that A is not in $GL_2(\mathbb{Z}_p)$. Then $\det(A) \equiv 0 \pmod{p}$; that is, $ad - bc \equiv 0 \pmod{p}$ or $bc \equiv ad \pmod{p}$. Either $bc \equiv 0 \pmod{p}$ or $bc \not\equiv 0 \pmod{p}$. We begin counting by taking $bc \equiv 0 \pmod{p}$; then there are $2p-1$ ways to choose b and c such that $bc \equiv 0 \pmod{p}$ and accordingly $2p-1$ ways to choose a and d such that $ad \equiv bc \pmod{p}$. Hence, there are $(2p-1)^2$ ways to obtain $ad \equiv bc \equiv 0 \pmod{p}$.

Now take $bc \equiv i \pmod{p}$ where $i \in \{1,2,...,p-1\} = \mathbb{Z}_p^*$. Observe that $\mathbb{Z}_p^*$ is a group with respect to multiplication, so by the uniqueness of solutions in groups, we obtain that for a single i, there are $p-1$ ways to obtain $bc \equiv i \pmod{p}$ and accordingly $p-1$ ways to obtain $ad \equiv bc \pmod{p}$. Note that there are $p-1$ unique choices for i, so by applying the basic multiplication rule of combinatorics, we obtain that there are $(p-1)(p-1)(p-1) = (p-1)^3$ different ways to obtain $bc \equiv ad \pmod{p}$ for all nonzero bc.

Since zero and nonzero bc choices are disjoint, we obtain that $k = (2p-1)^2 + (p-1)^3$, and substituting yields that

$$|GL_2(\mathbb{Z}_p)| = p^4 - k = p^4 - ((2p-1)^2 + (p-1)^3).$$

Simplifying then yields that

$$|GL_2(\mathbb{Z}_p)| = (p^2-1)(p^2-p).$$

□

Having obtained the order of $GL_2(\mathbb{Z}_p)$ for all prime integers p, we are now able to construct a modulus which will act in a similar fashion as $\phi(n)$ does in the RSA cryptosystem. We choose to use

$$g = (p^2-1)(p^2-p)(q^2-1)(q^2-q)$$

where p and q are the prime factors of the aforementioned n and we obtain a result analogous to Theorem 2. However, in order to do this, we require Lagrange's Theorem in regards to group order (for a proof, see [4]):

Theorem 4. ***Lagrange's Theorem.*** *Let H be a subgroup of a finite group G. Then $|H|$ divides $|G|$.*

We also require the following lemma.

Lemma 2. *Let $n = pq$ where p and q are distinct prime integers. If $M \in GL_2(\mathbb{Z}_n)$, then $M \in GL_2(\mathbb{Z}_p)$ and $M \in GL_2(\mathbb{Z}_q)$.*

Proof. Let $n = pq$ where p and q are distinct prime integers and suppose $M \in GL_2(\mathbb{Z}_n)$. We proceed by contradiction. Assume that $M \notin GL_2(\mathbb{Z}_p)$. Then by definition of a general linear group, $\gcd(\det(M), p) \neq 1$. Since p is prime, we have that $\gcd(\det(M), p) = p$; then $p|\det(M)$. Recall that $p|n$ by definition of n. Then $\gcd(\det(M), n) \geq p \neq 1$, which contradicts that $M \in GL_2(\mathbb{Z}_n)$. Thus, we conclude that if $M \in GL_2(\mathbb{Z}_n)$, then $M \in GL_2(\mathbb{Z}_p)$. By the same reasoning, $M \in GL_2(\mathbb{Z}_q)$. □

For our modification, we use M to denote the matrix representation of the plaintext, C to denote the resulting ciphertext, and e and d to denote our encryption and decryption exponents respectively. We calculate $C \equiv M^e \pmod{n}$ and $M \equiv C^d \pmod{n}$ using the fact that $ed \equiv 1 \pmod{g}$.

Theorem 5. *Let $n = pq$ where p and q are distinct prime numbers, let $M \in GL_2(\mathbb{Z}_n)$ be a matrix made up of nonnegative integers less than n, let $g_p = |GL_2(\mathbb{Z}_p)| = (p^2-1)(p^2-p)$ and $g_q = |GL_2(\mathbb{Z}_q)| = (q^2-1)(q^2-q)$ and define $g = g_p g_q$. Further let $e, d \in \mathbb{Z}^+$ such that $ed \equiv 1 \pmod{g}$, and let $C \equiv M^e \pmod{n}$. Then $C^d \equiv M \pmod{n}$.*

Proof. Suppose we have all variables and matrices as defined above. We will show that $C^d \equiv M \pmod{n}$. Note that $C^d \equiv (M^e)^d \equiv M^{ed} \pmod{n}$ by definition of C. Since $ed \equiv 1 \pmod{g}$, we obtain by definition of congruence that $ed = 1 + kg$ for some integer k. Hence we obtain that

$$M^{ed} \equiv M^{1+kg} \pmod{n},$$

so we obtain that

$$M^{ed} \equiv M^{1+kg} \equiv M \cdot ((M^{g_p})^{g_q})^k \pmod{p}$$

by definition of g. But we have that g_p is the order of $GL_2(\mathbb{Z}_p)$; hence, since $\langle M \rangle$ is a subgroup of $GL_2(\mathbb{Z}_p)$, we have by Lagrange's Theorem that $x := |\langle M \rangle|$ divides g_p. Hence we obtain that $g_p = jx$ where $j \in \mathbb{Z}$, so

$$C^d \equiv M^{ed} \equiv M \cdot ((M^{g_p})^{g_q})^k \equiv M \cdot ((M^{xj})^{g_q})^k \equiv M \cdot (M^x)^{jg_qk} \equiv M \cdot I^{jg_qk} \equiv M \pmod{p}.$$

In the exact same manner, we obtain that $C^d \equiv M \pmod{q}$. Hence we have that $C^d \equiv M \pmod{n}$ by elementwise application of Lemma 1. □

3.3 Example

As an example, we revisit the example of encrypting and decrypting *cryptography*. As before, our hypothetical message receiver selects prime integers $p = 503$ and $q = 499$ and calculates $n = 250997$. She now calculates

$$g_p = (p^2-1)(p^2-p) = 63886038048,$$

$$g_q = (q^2-1)(q^2-q) = 61876998000,$$

and

$$g = g_p g_q = 3953076248524019904000.$$

Finally, she selects encryption exponent $e = 241$ and finds her decryption exponent $d = 1016973972649332921361$ such that $ed \equiv 1 \pmod{g}$. The publicised key (which comes with

information on how to store the values in the matrix) is still (e, n) while the private key is (d, g). The sender converts his message *cryptography* into four submessages (the same four he attained in the example of RSA Cryptography) and creates the encryption matrix

$$M = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} = \begin{bmatrix} 31825 & 162015 \\ 71801 & 160825 \end{bmatrix}.$$

The sender forwards the ciphertext matrix

$$C \equiv M^e \equiv \begin{bmatrix} 31825 & 162015 \\ 71801 & 160825 \end{bmatrix}^{241} \equiv \begin{bmatrix} 153377 & 104497 \\ 76449 & 55902 \end{bmatrix} \pmod{n}.$$

Meanwhile, the receiver decrypts by computing

$$M \equiv C^d \equiv \begin{bmatrix} 153377 & 104497 \\ 76449 & 55902 \end{bmatrix}^{1016973972649332921361} \equiv \begin{bmatrix} 31825 & 162015 \\ 71801 & 160825 \end{bmatrix} \equiv M \pmod{n}.$$

3.4 Counterexample

Suppose, now, that we have a plaintext matrix whose determinant is zero modulo n and the sum of the cross-elements is a multiple of one of the factors of n, that is, let plaintext

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

such that

$$a + d = jp = b + c$$

or

$$a + d = kq = b + c.$$

Return to the prime integers used in [1]. The receiver opts to use $p = 43, q = 47$. The encryption modulus is $n = pq = 2021$, while the exponentiation modulus is

$$g = g_p g_q = (p^2 - p)(p^2 - 1)(q^2 - q)(q^2 - 1) = 3337488 \cdot 4773696 = 15932153115648.$$

The receiver chooses encryption exponent $e = 17$ such that $\gcd(e, g) = 1$ and calculates the decryption exponent $d = 14994967638257$. Suppose her friend sends her the message *uvuv*, which becomes the matrix

$$M = \begin{bmatrix} 21 & 22 \\ 21 & 22 \end{bmatrix} \text{ (note that } 21 + 22 = 43\text{)}.$$

The sender computes the ciphertext

$$C \equiv M^e \equiv \begin{bmatrix} 1634 & 172 \\ 1634 & 172 \end{bmatrix} \pmod{n}.$$

However, on decrypting this matrix, the receiver obtains that

$$C^d \equiv \begin{bmatrix} 1290 & 774 \\ 1290 & 774 \end{bmatrix} \pmod{n}.$$

4 Conclusion

While we demonstrated the Matrix RSA Cryptosystem for $GL_2(\mathbb{Z}_n)$, there is no reason that the Cryptosystem is not extendable to be used on a general linear group of general size s. The Matrix RSA cryptosystem extends the RSA system by exponentiating a square matrix as opposed to a scalar modulo n. This has a slight benefit of further complicating the brute force attack on the decryption exponent since g, the value we use in place of $\phi(n)$ (indeed, applying our cryptosystem to a scalar yields $g = \phi(n)$), is vastly larger than $\phi(n)$. Both the RSA and the Matrix RSA cryptosystems require roughly $\log_2(ed)$ exponentiations due to the technique known as successive squaring. However, whereas a scalar multiplication is one quick action for a computer, an $n \times n$ matrix multiplication requires at most n^3 scalar multiplications and $n^2(n-1) = n^3 - n^2$ scalar additions. With special algorithms these numbers can become slightly smaller but still much slower than the scalar multiplication. Hence the Matrix RSA cryptosystem is much slower than the scalar RSA system. In addition, the RSA Cryptography method is generally considered to have factoring $n = pq$ as the weakest point since the easiest method of calculating d is using $\phi(n)$ which is easiest calculated using p and q. The Matrix RSA cryptosystem is still completely reliant on p and q, so, similar to RSA, factoring n is still a reliant method of attack.

In addition, we still need to perform more tests on the Matrix RSA system: We found that there exists a time when the method fails to decrypt the ciphertext. However, does the decryption hold when the cross-elements of the plaintext do not sum to a multiple of the factors yet the determinant is zero? Also, we gave a specific example of when the matrix will not decrypt, but this example is unlikely to occur in reality. As such, we must investigate the probability of an indecryptable matrix actually occurring.

References

[1] S. Aboud et al, An Efficient RSA Public Key Encryption Scheme, *Fifth International Conference on Information Technology: New Generations*, (2008), 127-130

[2] Burton, David M., *Elementary Number Theory* McGraw-Hill Higher Education: 1221 Avenue of the Americas, New York, NY 10020 [c2007].

[3] Marshall, David C., Edward Odell, and Michael Starbird, *Number Theory Through Inquiry* Mathematical Association of America: MAA Service Center, Washington, DC 20090 [c2007]

[4] Nicholson W. Keith, *Introduction to Abstract Algebra* John Wiley & Sons, Inc., 111 River Street, Hoboken NJ 07030 [c2012].

[5] Trappe, Wade and Lawrence C. Washington, *Introduction to Cryptography with Coding Theory* Pearson Education Inc., Upper Saddle River, NJ 07458 [c2006]