Extended Essay

How are Mathematical concepts used in cryptographic solutions to achieve Information Security?

Subject: Mathematics

Session: May 2023

Date: 09/06/2022

Candidate: jgt165

Word Count: 2913

Contents

1 Introduction

1.1 Motivation

The use of **cryptography**, an evolutionary technique involves the aim to allow two people to exchange private/sensitive information on a public channel - one that may even be infested with malicious users. The use of cryptography to protect secrets dates back to the Caesar cipher amongst the Romans to the Enigma machine in World War II designed to protect confidential military intelligence. In the current world, cyber threats come from all types of places, and with open vulnerabilities, the data integrity from small businesses to large enterprises could be at stake. To put it in context, between February and March of 2014, eBay underwent a data breach of 145 million users which included encrypted passwords, names, e-mail addresses, addresses, date's of birth, etc [1]. Modern cryptography, built to counter such events, heavily relies on number theory and algebra to effectively secure communications and transactions all over the web today.

The subject of this paper aims to investigate: "How are Mathematical concepts is used in cryptographic solutions to achieve information security?". This is done by first looking elementary to intermediate number theory used to build cryptographic ciphers and the connection to the RSA encryption and decryption algorithm.

1.2 Public/Private Key Ciphers

In the field of cryptology, a **cryptographic cipher** (also known as encryption algorithms) is a system to encrypt/decrypt data. Data including bank transactions, military inteligence, sensitive intellectual data are sent from point to point with a low possibility of being meddled with through the use of different ciphers. Two popular techniques are known as **public-key** and **private-key** ciphers, with each having a **encryption key** and **decryption key**. An encryption key is used to convert `plain_text` to `cipher_text`, leading the decryption key to decode `cipher_text` back into `plain_text`.

A **private-key cipher** (also known as symmetric key encryption) utilizes a single key for encrypting and decrypting data. Essentially the process starts with an unencrypted plaintext message which is then ciphered with an encryption key. The receiving end must use the same

key to transform the ciphertext back into the original message.

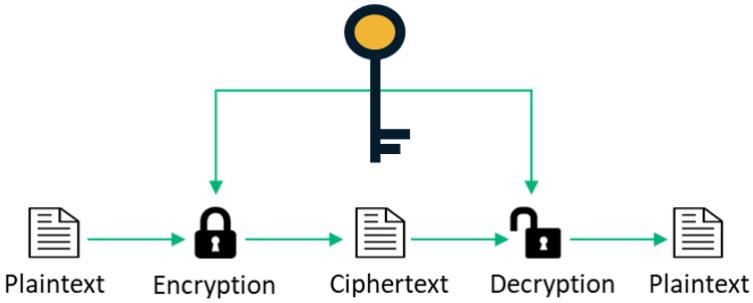

Figure 1: Private-key Encryption

In such case, the confidentiality of keeping the shared key a secret is essential to the cipher's security. The symmetric key encryiption system offers a fast and convenient way to set up, however leveraging on scalability as the secret key has the potential to be lost or stolen to adversaries aiming to decrypt messages.

On the other hand, a **public-key cipher** (also known as asymmetric key encryption) is developed on the basis of two keys: public and private key. The public key, with open access, encrypts the plaintext message before transmission. However, the ciphertext can be only decrypted with the private key. The niche about this system is that the public and private key are mathematically related, but one cannot be derived from the other. For example the RSA cryptosystem/algorithm utilizes the idea that multiplying two moderately large prime numbers, for example p and q is simple but recovering the factors given $n = p \times q$ is a more non-trivial.

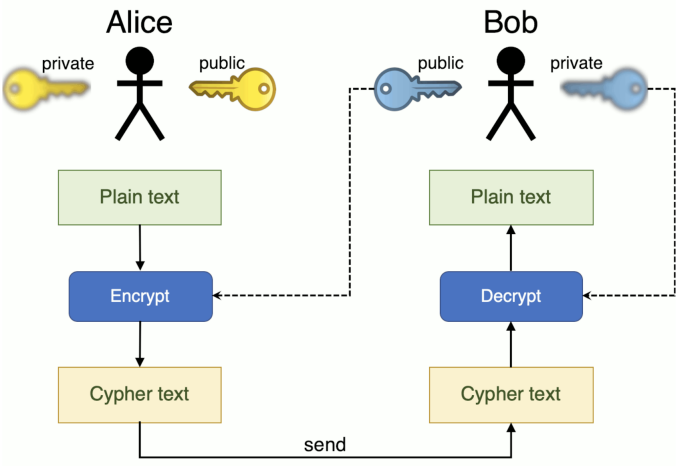

Figure 2: Public-key Encryption

The process and benefits of design of public-key cryptography can be seen with the following

example, referencing Figure 2. Let the process start with two people, Alice and Bob, where Alice is sending a message to Bob.

1. First, the receiver, Bob sends Alice his `public-key`, *publicly*, as in it's confidentiality is insignificant.
2. Alice, once receiving Bob's `public-key`, encrypts her message, `plain_text`, into `cipher-text` and sends it to Bob. To decrypt the message, a hacker would need Bob's private key, which is designed to only be in Bob's possession.
3. Thus, after receiving the `cipher-text`, Bob uses his `private_key` to decrypt the message.

The public-key encryption system offers a more secure and thorough transfer of data, when compared to symmetric encryption. In such case, Bob can save his private key to decrypt messages, even from multiple users. However, public-key encryption algorithms are computationally costly, which will be discussed later as to exactly why. A third convention used in cryptography is to use public-key encryption to share a *private-key* between two users securely, and then use the public-key encryption technique.

2 Number Theory Essentials

In this section, the pre-requisites and essential topics from number theory that have connection to cryptographic ciphers will be explored. The following section focuses on the RSA cryptosystem, of which the creators had to rely on theoretical topics to create public-key exchange system so that `private_key` could not be mathematically derived from the `public_key` effortlessly.

2.1 Prime Numbers and Divisibility

Definition 2.1.1. If a and b are integers, and $a \neq 0$, and there is an integer k such that $b = ak$, we say a **divides** b, or write $a \mid b$. Conversely, if a is not a divisor of b, we write $a \nmid b$ [2].

Definition 2.1.2. The **greatest common divisor** of two non-zero integers a and b, $gcd(a, b)$, is the largest integer d ($d \in \mathbb{Z}^+$) such that $d \mid a$ and $d \mid b$.

Proposition 2.1.1. For two integers $a, b \in \mathbb{Z}$, where $a \neq 0$, there exists a unique pair of integers q and r satisfying $b = aq + r$ and $0 \leq r < a$.

Also known as the *quotient-remainder theorem*, q is called the quotient and r the remainder.

Definition 2.1.3. If two positive integers m and n have no common divisors greater than 1, they are said to be **relatively prime** or **co-prime**. Or, it can be said that (the greatest common disivor equals 1) $\gcd(m, n) = 1$.

Euclidean Algorithm:

One method used to find the gcd of two non-negative integers is prime factorization, while another exploits repeated division known as the Euclidean Algorithm.

Let us suppose we have two positive integers a and b, where $a \neq b, a > b$. The algorithm works by initially dividing a by b, and by Proposition 2.1.1:

$$a = b \cdot q_0 + r_0 \tag{2.1.1}$$

where (q_0, r_0) are a pair of unique integers, with $q_0 \in \mathbb{Z}$ and $0 \leq r < b$ [3]. By repeated divisions the algorithm continues [3]:

$$\begin{aligned}
a &= q_0 \cdot b + r_0, \text{ where } 0 \leq r_0 < b, \\
b &= q_1 \cdot r_0 + r_1, \text{ where } 0 \leq r_1 < r_0, \\
r_0 &= q_2 \cdot r_1 + r_2, \text{ where } 0 \leq r_2 < r_1, \\
r_1 &= q_3 \cdot r_2 + r_3, \text{ where } 0 \leq r_3 < r_2, \\
&\vdots \\
r_{k-2} &= q_k \cdot r_{k-1} + r_k, \text{ where } 0 \leq r_k < r_{k-1}, \\
r_{k-1} &= q_{k+1} \cdot r_k + 0, \text{ and } r_{k+1} = 0
\end{aligned}$$

Once $r_{k+1} = 0$, by the Euclidean algorithm:

$$\gcd(a, b) = r_k \tag{2.1.2}$$

Example: Find $\gcd(1071, 462)$. Applying the Euclidean algorithm:

$$1071 = 2 \cdot 462 + 147$$

$$462 = 3 \cdot 147 + 27$$

$$147 = 7 \cdot 21 + 0$$

The last remainder is 0, thus $\gcd(1071, 462) = 21$.

Definition 2.1.4. An integer p, where $p > 1$ is **prime** if it's only divisors are 1 and p itself. Conversely, it is said to be **composite**.

Definition 2.1.5. If a and b are non-zero integers and $\gcd(a, b) = 1$, then a and b are said to be **coprime** or **relatively prime**.

Theorem 2.1.1. The **The Fundamental Theorem of Arithmetic** states that every integer greater than 1 can be written as a **unique** product of primes: $p_1^{n_1} p_2^{n_2} p_3^{n_3} \dots p_k^{n_k}$, where p_i are prime numbers and $n_i > 0$.

2.2 Modular Arithmetic

Modular arithmetic is a mathematical system dealing with integers $0, 1, 2, \dots, m-1$, also known as integers **modulo** m, where m is referred to as the **modulus** [4].

Definition 2.2.1. Let m, where $m \in \mathbb{Z}^+$ be the modulus. Two integers a and b are **congruent** if $m \mid (a - b)$ or equivalently both share the same remainder when divided by m. We write this relation as $a \equiv b \pmod{m}$ when they are congruent, and $a \not\equiv b \pmod{m}$ otherwise.

Example: 17 is congruent to 7 modulo 10 since $10 \mid (17 - 7) \rightarrow 10 \mid 10$, so $17 \equiv 7 \pmod{10}$. On the contrary, 17 is not congruent to 8 modulo 10 since $10 \nmid (17 - 8)$.

By Proposition 2.1.1, there are a number of commmon modular arithmetic properties. If $a, b \in \mathbb{Z}$, $m \in \mathbb{Z}^+$, and $a \equiv b \pmod{m}$, the following are true for any integer $c \geq 0$ [5].

1. $a + c \equiv b + c \pmod{m}$
2. $a - c \equiv b - c \pmod{m}$
3. $ac \equiv bc \pmod{m}$
4. $a^c = b^c \pmod{m}$

Definition 2.2.2. Given an integer b and modulo m, the **modular inverse** of b is an integer b^{-1} such that $b \cdot b^{-1} \equiv 1 \pmod{m}$.

Proposition 2.2.1. The **modular multiplicative inverse** or "modular inverse" of an integer b modulo m, b^{-1}, does not exist when $\gcd(b, m) > 1$ [4].

Proof. If b^{-1} exists, it is the solution to the congruence: $bx \equiv 1 \pmod{m}$, which can be re-written as $bx = km + 1$ or $bx - km = 1$, $k \in \mathbb{Z}$. If $d = \gcd(b, m)$, we have $d \mid bx$ and $d \mid km$ which implies $d \mid (bx - km)$. Since $bx - km = 1 \Rightarrow d \mid 1 \Rightarrow d = 1$. Thus for b^{-1} to exist, $\gcd(b, m)$ must be 1, and cannot be greater than 1.

2.3 Fermat's Little Theorem

Theorem 2.3.1. (Fermat's Little Theorem (FLT) [6]) Let p be some prime number, and a some integer ($a \in \mathbb{Z}$). If a is not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$ which is equivalent to:

$$a^p = a \pmod{p} \textbf{ or } p \mid a^p - a \tag{2.3.1}$$

Proof. (Proof By Induction [6]) Let $P(a)$ be the proposition that $a^p \equiv a \pmod{p}$, where p is a fixed prime number.

With the base case, $P(1)$, we get:

$$1^p \equiv 1 \pmod{p} \tag{2.3.2}$$

which is true.

Next, assume/suppose the following congruence is true for some $P(a = k)$, $k \in \mathbb{Z}^+$:

$$k^p \equiv k \pmod{p} \tag{2.3.3}$$

(Inductive Step) For $P(a = k + 1)$, by the binomial theorem, our left side is:

$$(k+1)^p = \sum_{i=0}^{p} \binom{p}{i} k^{p-i} \tag{2.3.4}$$

$$= k^p + \binom{p}{1} k^{p-1} + \binom{p}{2} k^{p-2} + \cdots + \binom{p}{p-1} k + 1 \tag{2.3.5}$$

Taking $\pmod p$ to 2.3.5, we see that p divides all $\binom{p}{i} = \frac{p!}{i!(p-i)!}$ for $1 \leq i \leq p-1$, as $p \mid p!$. With simplification to 2.3.5, we get:

$$(k+1)^p \equiv k^p + 1 \pmod p \tag{2.3.6}$$

From 2.3.3, we know that $k^p \equiv k \pmod p \Rightarrow k^p + 1 \equiv k + 1 \pmod p$. Substituting into 2.3.6, we get:

$$(k+1)^p \equiv k + 1 \pmod p \tag{2.3.7}$$

Thus, the proposition holds for $a = k + 1$. If the proposition is true for $a = k$, $k \in \mathbb{Z}$, and is true for $a = k + 1$, it shows $P(k) \Rightarrow P(k + 1)$. Since the proposition is true for $a = 1$, by the principle of mathematical induction it is true for all $a \in \mathbb{Z}$.

Using FLT to find Modular Inverse:

For the specific case of a **prime modulus**, Fermat's Little Theorem can be used to the modular inverse of an integer with respect to the modulus.

Let a be an integer, and m a prime modulus, where $\gcd(a, m) = 1$ and a^{-1} is the modular multiplicative inverse of a. From Theorem 2.3.1 we have:

$$\begin{aligned} a^{m-1} &\equiv 1 \pmod m \\ a^{m-1} \cdot a^{-1} &\equiv a^{-1} \pmod m \\ a^{m-2} \times \underbrace{(a \cdot a^{-1})}_{1 \text{ from Definition 2.2.2}} &\equiv a^{-1} \pmod m \\ \Rightarrow a^{m-2} &\equiv a^{-1} \pmod m \end{aligned} \tag{2.3.8}$$

To find a^{m-2} from 2.3.8 computationally, one can utilize Binomial Exponentiation [7].

Binomial Exponentiation:

Problem Statement: Find a^n where a and n are positive integers, and n can tend to very large numbers i.e $\approx 10^{18}$.

Thoughts: Computationally, multiplication is a highly intensive task. In some cases, it could take up to years to calculate the powers of certain large expressions. Binomial exponentiation suggests an idea to caluclate a^n in utmost $\lfloor \log_2 n \rfloor + 1$ operations [7]. To take this into

perspective, if $n = 10^{18}$, binomial exponentiation could solve it in 60 multiplication operations.

Idea:

Write n in terms of base 2. For example given $a = 4$ and $n = 11$.

$$4^{11} = 4^{1011_2} = 4^8 \cdot 4^2 \cdot 4^1$$

Specifically, we are finding the following expression: $a^1, a^2, a^4, a^8, \ldots, a^{2^{\lfloor \log n \rfloor}}$ [7]. As for calculating the powers in the stated list, each can be calculated by squaring the previous (i.e $a^8 = (a^4)^2$), resulting in fewer multiplications as well.

Thus, 4^{11} can be found with 3 multiplications:

$$4^{11} = 65536 \cdot 16 \cdot 4 = 4194304$$

2.4 Chinese Remainder Theorem

Theorem 2.4.1. (Chinese Remainder Theorem (CRT) [8]) If $m_1, m_2, \ldots, m_k$ are pairwise coprime positive integers, ($\gcd(m_i, m_j) = 1$ for $i \neq j$), and $a_1, a_2, \ldots, a_k$ are arbitrary integers, then the following system of congruences:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

have a solution, unique modulo $m_1 m_2 \ldots m_k$.

Example: Solve the following simultaneous congruences:

$$\begin{aligned} x &\equiv 3 \pmod{5} \\ x &\equiv 1 \pmod{7} \end{aligned}$$

Solving for x separately we get:

$$x \equiv 3 \pmod 5 \Rightarrow x \in \{3, \boxed{8}, 13, 18, 23, \dots\}$$

$$x \equiv 1 \pmod 7 \Rightarrow x \in \{1, \boxed{8}, 15, 22, 29, \dots\}$$

We see the first solution is 8, and by CRT, all solutions are in the form: $x \equiv 8 \pmod{35}$ or $8 + 35k$.

Proposition 2.4.1. Let α and β be two co-prime positive integers. If $x \equiv a \pmod \alpha$ and $x \equiv a \pmod \beta$, then by CRT, $x \equiv a \pmod{\alpha\beta}$.

Example: If $53 \equiv 5 \pmod 6$ and $53 \equiv 5 \pmod 8$, we know $53 \equiv 5 \pmod{48}$.

3 The RSA Cipher

3.1 About the RSA

The RSA (Rivest-Shamir-Adleman) is a public-key cryptographic cipher/cryptosystem, and named after the surnames of it's authors, first published in 1977 [9]. The security and strength of the system stems from the notion that factorizing large integers into their prime number factorizations is unfeasible in a computational sense.

The revolutionary development of the RSA algorithm led to countless practical applications, to which most can associate with. Any website labeled `https` uses a security protocol system called SSL (Secure Sockets Layer) [10]. Intended to protect two-way data communication between web serves, the protocol relies on RSA Ciphers. A `https` website can also be identified with the padlock icon, to the left of the site url [10]. It is also used for the primary goal of *Information Security*, such as authenticity in emails, bank transactions, site logins, etc.

3.2 Euler's Theorem

Definition 3.2.1. The **Euler totient-function** [12], or $\varphi(n), n \in \mathbb{Z}^+$, is defined to be the number of positive integers less than or equal to n, that are coprime to n. Formally: $\varphi(n)$ is the number of $m \in \mathbb{Z}^+$, where $1 \leq m \leq n$ and $\gcd(m, n) = 1$.

Example: Find $\varphi(9)$:

$$\gcd(1,9) = 1,\ \gcd(2,9) = 1,\ \gcd(3,9) = 3$$

$$\gcd(4,9) = 1,\ \gcd(5,9) = 1,\ \gcd(6,9) = 3$$

$$\gcd(7,9) = 1,\ \gcd(8,9) = 1,\ \gcd(9,9) = 9$$

By inspection, we can see that $\varphi(9) = 6$.

Proposition 3.2.1. If p is a positive integer ($p \in \mathbb{Z}$), and prime, $\varphi(p) = p - 1$.

Proof. We know from Definition 2.1.3 that a prime number p has divisors 1 and p. By analyzing $\gcd(k, p)$ for $1 \leq k \leq p$, we can see that except for $\gcd(k = p, p) = p$, $\gcd(k, p) = 1$ for all other k. Thus, $\varphi(p) = p - 1$. □

Proposition 3.2.2. If n is a positive integer, where n's prime-factorization is $n = p_1^{e_1} p_2^{e_2} p_3^{e_3} \dots p_m^{e_m}$ [12]:

$$\varphi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right) = n \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_m}\right) \tag{3.2.1}$$

Theorem 3.2.1. (**Euler's Theorem** [13]) If positive integers a and n are coprime, and $\varphi(n)$ is the Euler's Totient Function:

$$a^{\varphi(n)} \equiv 1 \pmod{n} \tag{3.2.2}$$

Example: Find $3^6 \pmod{14}$.

We know $\gcd(3, 14) = 1$, so 3 and 14 are therefore coprime. We also see that $\varphi(14) = 6$, which is the exponent 3 is raised to. By Theorem 3.2.1, where $a = 3, \varphi(n) = 6, n = 14$, the answer must be 1.

3.3 Encryption Process

For the sake of simplicity, we will refer back to the example in 2, where Bob is the receiver and Alice is the sender. This section will focus on the direct step-by-step mathematical process that RSA undergoes. The following will reinforce the steps with mathematical proofs.

ASCII Character Encoding: The RSA cryptosystem requires messages to be transcribed as numbers - numerical values. ASCII (American Standard Code for Information Interchange), is a character encoding system where each unique alphanumeric character is represented with an integer.

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	fl	255	ˇ

Figure 3: Basic ASCII Table [14]

For example consider the message "box":

$$\text{"box"} = 98 + 111 + 120 = 98111120$$

Algorithm (Key Generation) [11]:

1. Bob chooses two **secret** primes, p and q, relatively large, and computes his modulus $n = pq$.

2. Next, Bob forms his **public encryption key** e, by picking an integer e such that $\gcd(e, \varphi(n)) = 1$. Note that since $n = pq$, where p, q are primes, $\varphi(n) = (p-1)(q-1)$.

3. The **secret decryption key**, d, is computed such that $ed \equiv 1 \pmod{\varphi(n)}$. Thus, d is the modular multiplicative inverse of e, since $\gcd(e, \varphi(n)) = 1$.

4. Bob publishes his public key as a pair: (e, n), and keeps his secret key as (d, n). Note p and q are also kept hidden.

Algorithm (Encryption and Decryption) [11]:

In our case, where Alice will send a message (`plain_text`) to Bob (m), it must be convert to ASCII i.e consider the example of "box" outlined in an earlier section.

For simplicity, the assumption is made that m fits the inequality $2 \leq m < n$, where n is the modulus outlined above.

When Alice sends the message to Bob, she must have the message m, Bob's public-key (e, n). The `cipher_text` c is calculated as:

$$c \equiv m^e \pmod{n} \tag{3.3.1}$$

Alice then sends c to Bob, who can then decrypt the `cipher_text` back to `plain_text` with his secret key d and n:

$$m \equiv c^d \pmod{n} \tag{3.3.2}$$

It is interesting to point out that what RSA is suggesting is: $m^{ed} = m \pmod{m}$, where the public/private keys e and d cannot be derived from one another.

Example:

1. Bob chooses his prime numbers as $p = 23$ and 37. n is calculated as $n = pq = 851$.

2. We know that $\varphi(n) = (p-1)(q-1) = (23-1)(37-1) = 792$. One value of e that satisfies $\gcd(e, \varphi(n)) = 1$ is $e = 5$.

3. Bob can calculuate his secret key d as the modular inverse of 5, with the modulus as $\varphi n = 792$. Thus, we have: $d = e^{-1}$ where $e \cdot e^{-1} \equiv 1 \pmod{\varphi(n)}$. By calculating modular inverse, we get $d = 317$.

4. Alice wants to send the message "BED" to Bob. For simplicity, as m needs to be less than n, we assume the mapping $A \to 1$, $B \to 2, \ldots$ Box then equals: $B + E + D =$ "2" + "5" + "4" = 254.

5. Alice encrypts the message m to cipher text as follows: $c \equiv m^e \pmod{n}$. Thus, we get: $c \equiv 254^5 \equiv 760 \pmod{851}$.

6. Bob can decrypt the cipher text back into plain text as follows: $c \equiv c^d \pmod{n}$. Thus, we get: $d \equiv 760^{317} \equiv 254 \pmod{851}$.

7. Bob gets the text message as 254, from which he can determine the original message using ASCII, resulting in "BED".

3.4 Proof of Correctness

From the encryption process, we see that $\varphi(n) = (p-1)(q-1)$. Since $n = pq$ (the prime factorization), we can utilize Proposition 3.2.2 to see why this is true:

$$\begin{aligned}\varphi(n) &= n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)\\ &= pq\left(1-\frac{1}{p}\right)\left(1-\frac{1}{q}\right)\\ &= (p-1)(q-1)\end{aligned}$$

The overall correctness of the RSA Algorithm stems from the theorem:

Theorem 3.4.1. For defined integers e, d and n, from **Algorithm (Key Generation)**, $m^{ed} \equiv m \pmod{n}$ for $m \in \mathbb{Z}$.

Proof. Consider two cases:

Case 1: When m is coprime to n [15]: We know that $ed \equiv 1 \pmod{\varphi(n)}$, which can be rewritten as:

$$\begin{aligned}ed \equiv 1 \pmod{\varphi(n)} &\Rightarrow \varphi(n) \mid (ed-1)\\ &\Rightarrow (ed-1) = \varphi(n)k \text{ where } k \in \mathbb{Z}\\ &\Rightarrow ed = 1 + \varphi(n)k \text{ where } k \in \mathbb{Z}\end{aligned}$$

Since m and n are coprime, this means that $\gcd(m, n) = 1$, and by Euler's Theorem (Theorem 3.2.1), $m^{\varphi(n)} \equiv 1 \pmod{n}$.

Next, consider the following:

$$m^{ed} \equiv m^{1+\varphi(n)k} \equiv m^1 \cdot (m^{\varphi(n)})^k \equiv m \cdot 1^k \equiv m \pmod{n}$$

Thus, for case 1, $m^{ed} \equiv m \pmod{n}$ for $m \in \mathbb{Z}$.

Case 2: When m is not coprime to n [11]:

From the described algorithm, we know that e and d satisfy $ed \equiv 1 \pmod{\varphi(n)}$ which can be written as:

$$ed = 1 + k(p-1)(q-1), \text{ where } k > 0$$

Since $n = pq$, and p and q are co-prime, by the Chinese Remainder Theorem, specifically Proposition 2.4.1, it suffices to show:

$$m^{ed} \equiv m \pmod{p} \textbf{ and } m^{ed} \equiv m \pmod{q}$$

Taking the first congruence $m^{ed} \equiv m \pmod{p}$, consider two cases:

1. If $m \equiv 0 \pmod{p}$, then $m^{ed} \equiv 0 \pmod{p}$, as m^{ed} is simply a multiple of m.
2. If $m \not\equiv 0 \pmod{p}$, we know $m^{p-1} \equiv \pmod{p}$ by Theorem 2.3.1, and we get:

$$m^{ed} \equiv m^{1+k(p-1)(q-1)} = m \cdot (m^{p-1})^{q-1} \equiv m \cdot 1^{q-1} \equiv m \pmod{p}$$

From both cases, $m^{ed} \equiv m \pmod{p}$ holds. By symmetry, substituting q for p yields the same result. Thus, for case 2, $m^{ed} \equiv m \pmod{n}$, for $m \in \mathbb{Z}$.

The co-prime Case:

Analyzing Case 1, we see the assumption that n and m are co-prime. Many published papers on RSA use the Chinese Remainder Theorem for the general case, regardless of coprime properties. This is because the probability of m and n **not** being co-prime ($\gcd(m, n) \neq 1$) is quite low.

We know that $m < n$ from the encryption process. For $\gcd(m, n) \neq 1$ for $2 \leq m < n$, means that m must be a multiple of p, q or both. Considering the multiples of p and q within the range of n we get:

$$p \to p, 2p, 3p, \ldots, qp$$

$$q \to q, 2q, 3q, \ldots, pq$$

Since p and q are prime we know that thedre are $p + q$ possibilities with the common pq, thus $p + q - 1$ unique possibilities. Thus, we get the probability as:

$$\frac{p}{pq} + \frac{q}{pq} - \frac{1}{pq} = \frac{1}{p} + \frac{1}{q} - \frac{1}{pq}$$

As p and q are choosen as relatively large prime numbers, it is highly improable that m and n will not be co-prime. However, even if they are, the algorithm is still proved by case 2.

3.5 Security of RSA

RSA relies on p and q – the prime numbers choosen – where $n = pq$, to be incredibly large. This is since in order to make n difficult to factor into p and q. The beauty within this is that there is no concrete process offered by Mathematics to simply provide a list of prime numbers below n. Despite having multiple primality tests published, RSA remains secure with the notion that n is difficult to factor without an insane amount of computational power.

4 Conclusion

Information security is essentially the practice and method in which individuals securely keep a ledger's of everyones information. This can range from simple greetings passed over thorugh email in Gmail, to million-dollar transacations occuring daily on the Bitcoin cryptocurrency channel. To this end, with the invaluable time and effort of numerous computer scientists and mathematicians, we have developed methods of encryption and decryption through the innovative and creative use of Mathematics. The RSA (Rivest-Shamir-Adleman) method is a clever interlay of the topics within principal number theory and algebr, manipulated to create an "one-way" easy mathematical process. Notably, this included topics such as prime numbers, divisibility, modular arithmetic, Fermat's littl theorem, and Chinese remainder theorem.

Speaking to the proclaimed security of RSA, according to [16], it is clear that computers of

today's age will struggle with the main factorization present in this algorithm. According to [16]: "classic computers [would take] around 300 trillion years to break a RSA-2048 bit encryption key". However, while this may feel "safe", it doesn't talk to the full picture. In recent times, quantum computers – exceedingly fast numerical machines – are on the rise. Capable of processing operations at insane speeds, this poses what may be a big threat to the cryptography world, but in response to this there are new cryptographic ciphers on the rise.

For example, take a look at symmetric ciphers. As discussed previously, these are not usually favoured due to the fact that messengers have to establish a secret key over a public channels which may be tampered with. However, new mathematics has worked to solve in an effective way. With the use of the Discrete Logarithm Problem (DLP) on finite groups (beyond the topics within this paper), there have been two new cryptographic ciphers invented known as: Diffie-Hellman Key Exchange, and Elgamal Encryption. It is important to acknowledge both, due to the nature of information security being a real-world problem present on online-browsers, emails, and social media accounts. Both of these are typically used for applications that need to be scalable and fast. Since a public-key cryptographic cipher is a prolongued process, modern computer scientists favour the latter two algorithms for efficiency.

References

[1] https://www.washingtonpost.com/news/the-switch/wp/2014/05/21/ebay-asks-145-million-users-to-change-passwords-after-data-breach/

[2] https://iqti.iisc.ac.in/wp-content/uploads/2021/06/stark_number_theory_chap1-3.pdf

[3] http://www.math.toronto.edu/burbulla/lecturenotes246/Chapter7.pdf

[4] https://artofproblemsolving.com/ebooks/intro-number-theory-ebook/c12s2

[5] https://www.cse.cuhk.edu.hk/~taoyf/course/bmeg3120/notes/rsa-proof.pdf

[6] https://artofproblemsolving.com/wiki/index.php/Fermat%27s_Little_Theorem

[7] https://cp-algorithms.com/algebra/binary-exp.html

[8] https://brilliant.org/wiki/chinese-remainder-theorem/

[9] https://www.cse.cuhk.edu.hk/~taoyf/course/bmeg3120/notes/rsa.pdf

[10] https://vknight.org/Computing_for_mathematics/Assessment/IndividualCoursework/PastCourseWorks/2015-2016/weaving2015-2016.pdf

[11] https://people.engr.tamu.edu/andreas-klappenecker/alg/rsa.pdf

[12] https://artofproblemsolving.com/wiki/index.php/Euler%27s_totient_function

[13] https://artofproblemsolving.com/wiki/index.php/Euler's_theorem

[14] http://www.haghish.com/statistics/stata-blog/stata-programming/ascii_characters.php

[15] https://www.gcsu.edu/sites/files/page-assets/node-808/attachments/maxey.pdf

[16] https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art#:~:text=It%20would%20take%20a%20classical,RSA%2D2048%20bit%20encryption%20key.

[17] https://artofproblemsolving.com/wiki/index.php/Relatively_prime