Tracing Cryptography

Kaelan Carney
M429: History of Mathematics
Dr. Bharath Sriraman
April 25, 2025

Abstract

This paper traces the historical evolution of cryptography from its origins in ancient civilizations to the development of modern RSA encryption. It begins with early Arabic contributions, including steganography and the first substitution and transposition ciphers. It then moves to the advancements from Greek and Roman societies, such as the Spartan skytale and the Caesar cipher. This paper then examines the Vigenère/autokey cipher, and the Enigma machine's role in World War II. The advent of computers and the ensuing encoding systems like ASCII and asymmetric key cryptosystems, culminating in the RSA algorithm. RSA's reliance on Euler's totient function and the Euler-Fermat theorem is analyzed. As well as the importance of these principles in modular arithmetic efficiency via the Square and Multiply method.

The History of Cryptography

Arabic Cryptography

In David Kahn's book, *The Codebreakers*, he states that the development of techniques most similar to what we consider cryptography began in Arabic nations and spread outward. These nations employed techniques that were simplistic by today's standards but undeniably significant. Some of these techniques were the earliest examples of steganography, the practice of concealing information within other information. A modern example is embedding a message in a digital image; however, it could be as simple as using the first word of each sentence or employing invisible ink. These techniques were compiled in a 14-volume encyclopedia, written by Shihab al-Din Abu ʿl-Abbas Ahmad ibn ʿAli ibn Ahmad ʿAbd Allah al-Qalqashandi in 1412. This encyclopedia, *Subh al-A'sha*, detailed many methods developed in Arabic nations, including both steganography and, of particular relevance to this paper, cryptography.

The Arabic nations developed a diverse repertoire of cryptographic methods. One of the simplest being the use of a different language, either invented for the purpose of secrecy or an already-known one. More notably, these techniques represent the earliest known examples of both transposition and substitution ciphers. Transposition ciphers retain the same letters as the original message but rearrange their positions according to a specific rule. Substitution ciphers replace the letters of the message with others. Often, these methods are combined to create more secure algorithms. These volumes also mark the first instance of cryptanalysis, the study of cryptography and language. Kahn explains that cryptanalysis likely emerged from the study of the Quran, as scholars analyzed the occurrences of words and letters in the text. Such analysis can lead to methods for breaking ciphers, particularly substitution-based ones.

Greek and Roman Cryptography

Ciphers have obvious value to a military strategist. Being able to anticipate an opponent's moves would ensure that one never falls into a trap or is caught by surprise. Kahn theorizes that this military need partly explains why Chinese dynasties never consistently used or developed ciphers. Thus, it is unsurprising that some of the earliest records of ciphers used in a military context come from the Spartans. Around the 5th century B.C., the Spartans employed a device

called a “skytale”, a round rod or pole of a specific diameter. Both the sender and receiver required a rod of identical diameter, crucial to the encryption and decryption process. The sender would wrap a piece of leather or parchment around the rod and write the message along its length. When unwrapped, the parchment’s letters would be scattered, rendering the message unreadable. Upon receiving the message, the recipient would wrap the parchment around their rod, realigning the letters to make the message readable again. If a spy acquired the parchment, they would need a rod of the exact diameter to decode the message; otherwise, it would remain unreadable.

The Greek writer Polybius devised a unique cryptographic method that could be adapted to send messages over long distances efficiently. He arranged the 24 letters of the Greek alphabet in a 5 × 5 grid. By numbering the rows and columns, he created a system in which each letter was represented by two numbers. This table illustrates the same process applied to the English alphabet.

0	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	ij	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	s	y	z

The astute might notice that the English alphabet contains 26 letters, two more than the Greek alphabet and one more than the 25 cells in a 5 × 5 grid. For this reason, I combined the letters “i” and “j” in a single cell to avoid omitting a letter. Using this table, we can determine that “n” is represented by 33, “w” by 25, and “d” by 41. This method allows the creation of a secret message, such as 32|51|13|13|43, which can only be decoded by someone in possession of the table. This innovation is significant in itself, but Polybius also proposed transmitting numbers over long distances using torches held in both hands. For example, to send the letter “q,” one could hold one torch in the left hand and four in the right. Although Polybius’s method is not as viable today, it introduces a concept common in the modern era: the conversion of letters to numbers. This will be discussed in more detail later in this paper. Never the less, it is worth noting that Morse code relies on translating the English alphabet into a form of binary code. Also, computers rely on this number-to-alphabet conversion.Associating numbers with letters enables the use of a wide range of mathematical tools for data encryption.

Cryptography advanced further during the Roman era, notably with the Caesar cipher. As Kahn writes, Caesar used this cipher to write letters to friends, such as Cicero. His frequent use

of the cipher is why it is attributed to him as the Caesar cipher. The algorithm is simple and exists in many variations. It involves shifting each letter of the intended message by a fixed number of positions in the alphabet. For example, with a shift of 3, the letter "a" becomes "d", "b" becomes "e", and so forth. By convention, any alphabet shifted by a non-zero amount is called a Caesar alphabet.

The Caesar cipher is not commonly used today due to the advent of computers, which have made several methods for breaking it more viable. For example, analyzing letter frequency or systematically trying every possible shift on an encrypted message are both effective techniques. However, a modern variant of the Caesar cipher, known as ROT13 (short for "rotate 13"), has emerged. As the name suggests, ROT13 is a Caesar cipher with a shift of 13. This variant is unique because it evenly divides the 26-letter English alphabet. Consequently, encrypting a message with ROT13 and then encrypting it again yields the original message. To mitigate vulnerabilities such as letter frequency analysis and the predictability of Caesarian ciphers, a cipher must exhibit a more randomized structure. Ideally, each letter in an encrypted message should appear randomized to reveal as little information as possible.

The Vigenère cipher

Blaise de Vigenère was born in 1523 in Saint-Pourçain, a village in France located approximately halfway between Paris and Marseille. At the age of 26, he traveled to Rome, where he first encountered cryptography. Kahn reasonably suggests that, while in Rome, Vigenère studied the works of Trithemius, Bellaso, Alberti, and other cryptographic experts. Inspired by these works, he developed his own cryptographic method, now known as the autokey cipher. At 47, while raising his one-year-old daughter, he published his complete method in a book titled *Traict'e des Chiffres*. Unfortunately, his method faded into obscurity until its rediscovery in the 19th century, when cryptography gained renewed interest. This rediscovery led to confusion: what is today called the Vigenère cipher is not his original autokey cipher. Although we know his original method—which will be described in the following paragraph—the origins of the device called the Vigenère cipher remain unclear. Kahn attributes the confusion to 19th-century cryptographic writers who misrepresented the autokey cipher as a simpler method. Some sources suggest that the Vigenère cipher is actually Bellaso's method, misattributed to Vigenère, though no reliable evidence could be found to support this claim. Nevertheless, given the similarities between the algorithms, this assumption is not unreasonable.

The autokey and Vigenère ciphers, along with their variations, share fundamental similarities. They all utilize the same "rotating" alphabet, as shown in *Figure 1*, and apply keys in the same way. However, they differ in their key generation methods. This bears a striking resemblance to the recent history of RSA encryption, a topic we will explore later. We will examine the autokey method because it is most directly attributable to Vigenère, unlike the method commonly referred to as the Vigenère cipher. Vigenère method requires two components: a plaintext message, such as "hello," and a short primer known to both the sender and the receiver. Vigenère reportedly favored single-letter primers, though the method can easily accommodate

longer ones. For this example, we will use the message “hello” and the primer “p.” To begin the algorithm, we write the message and the primer as shown in the following table.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1: The modern Vigenère table

h	e	l	l	o
p				

To proceed, we use the first letter of the plaintext message to select a column in the table that follows and the primer to select a row. The intersection of the column and row yields the first letter of the encrypted message. We write this letter below the primer to begin the encrypted message and also to the right of the primer to serve as the next letter in the key. Thus, with each letter we encrypt, we extend the key by one letter. This means that once we are done, we will have an encryption key that is long enough to encrypt the whole message. Applying this process to the entire message produces the table shown below.

h	e	l	l	o	

p	w	a	l	w	k
w	a	l	w	k	

To decrypt the message we simply follows the same process in reverse. We write the encrypted message above the primer. Then, we follow the row designated by the key until we find the encrypted letter. The column that the encrypted letter is in is the decrypted message that we write below. Repeating this process for the rest of the letters results in this table.

w	a	l	w	k	
p	w	a	l	w	k
h	e	l	l	o	

This method of generating the key as one encrypts or decrypts the message creates an added layer of security. This feature means that if the one decrypting were to make a mistake or have the wrong primer, the rest of the message would be unreadable.

As noted earlier, several variations of the autokey cipher exist. This paper considers the simplest variation to be an autokey cipher that uses a word as the primer instead of a single letter. This approach makes the key more resistant to brute-force attacks while maintaining the autokey's partial encryption feature. Departing from the autokey cipher allows us to use other methods, such as the one mistakenly called the Vigenère cypher, which employs a repeating key that extends for the length of the message. For example, to encrypt the message "hello there" using the primer "key," the primer is repeated approximately 3.33 times, resulting in the key "keykeykeyk." This method is less secure because it lacks the autokey cipher's feature of incorporating the message itself into the key. This makes the resulting key and message repetitive and somewhat predictable.

World War 2

One of the most renowned cryptographic devices in history is the German Enigma Machine. Invented by Arthur Scherbius, a German engineer, the Enigma initially failed commercially in the 1920s. However, recognizing its potential, the German armed forces adopted it during Hitler's rearmament in the 1930s. The machine operated in two stages, utilizing rotors and a pegboard. The rotors were metal disks with electrical contacts that swapped each input letter with another. Throughout World War II, various rotors were created, offering extensive options for message encryption. The first stage of encryption used three to five rotors. Each rotor would swap the input letter with another, meaning that for three rotors, the input letter would change three times. After a rotor switched a letter, it was also rotated, adding a more randomized component to the system. Although the transformation of a single letter was a simple substitution, the rotation of each rotor makes this stage more similar to a substitution operation. Yet, the vast number

of possible configurations—arising from the combinations of rotor selections, their orderings, and their initial rotations—was deemed insufficient. Consequently, a second stage employed a pegboard to further transform the rotor output. By connecting wires between pairs of pegs, the user could specify that any letter output from the rotors be swapped with another. This pegboard mechanism exponentially increased the number of possible settings, creating an extraordinarily complex encryption system. That the Enigma was ever deciphered is truly remarkable.

The Beginnings of RSA

Following World War II, computers became increasingly prevalent. This surge in computational power significantly reduced the time of repetitive tasks, enabling faster brute-force attacks on keys and rapid message encryption. A notable consequence of this shift was that computers operate exclusively in numerical formats, specifically binary numbers. Consequently, each letter of the alphabet is represented by a number, akin to the method developed by Polybius. Today, various encoding standards facilitate this translation between numbers and letters. The two most widely used standards are ASCII, shown in the table below, and Unicode, which is too long to list here. This ability to convert numbers to letters opened the door to new cryptographic algorithms that could utilize number theory.

b_7 b_6 b_5 → / Bits					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Row ↓ / Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	¦
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Figure 2: ASCII character table

The first modern cryptographic methods emerged in the late 1970s, generally falling into one of two categories: symmetric or asymmetric key algorithms. Symmetric key algorithms, such as the Vigenère and autokey cipher, use a single cryptographic key for both encryption and decryption. This approach has the disadvantage of requiring both parties to securely share the key. In 1976, during the post-Cold War era, Whitfield Diffie and Martin Hellman published a groundbreaking method now recognized as the first public-private, or asymmetric, key cryptosystem. This system derives its name from the use of two keys: a public key, which is accessible to everyone, and

a private key, known only to the recipient. Asymmetric cryptosystems are widely regarded as more secure than their symmetric counterparts but demand greater computational power. At the time of its publication, Diffie and Hellman's work was not immediately adopted because it lacked a practical one-way function—a method to encrypt a message such that decryption by an eavesdropper would be computationally infeasible.

This problem is where Rivest, Shamir, and Adleman made progress. They worked together at MIT to expand the work of Diffie and Hellman, resulting in the RSA cryptosystem that they released in 1977. Their method allows the sender to encrypt a message as long as they know the receiver's public key, and only the receiver's private key can decrypt it. They relied heavily on developments in number theory, especially from Euler and Fermat. There was also another man, Clifford Cocks, who was working for the British intelligence agency and created a similar algorithm. Despite completing his research in 1973, a few years before RSA was created, the secrecy of his work meant that he was never credited for the discovery until after it was released, well into the 90′s.

RSA at its core relies heavily on the work of Pierre de Fermat, a lawyer from France. He was born in 1601, in the middle of the Renaissance and around the time of the founding of the British East India Company. He went to school and primarily worked as a lawyer, but he nevertheless made large contributions to analytical geometry, calculus, and—what this paper is interested in—number theory. Fermat made many discoveries in Pell equations, perfect numbers, another subset called Fermat numbers, and others. This work resulted in what is now called *Fermat's Little Theorem*, which is crucial in RSA encryption. A substantial portion of his work is known because of letters he sent to friends that contained his discoveries. However, his work generally did not include proofs, which created room for the next generation of mathematicians, namely Euler.

Leonhard Euler was a Swiss mathematician born in 1707, about 50 years before the Seven Years' War and the American Revolution. He is credited with his work in geometry, calculus, mechanics, number theory, and astronomy. Euler's mathematical ability earned him the respect of Johann Bernoulli and his two sons. He also befriended many rulers, such as Frederick the Great, ruler of Prussia, and Catherine II, empress of Russia. Euler was a regular powerhouse of mathematical ability; however, he lost sight in one eye in 1735, then in the other in 1766. Nevertheless, he continued to work, sustained by his incredible memory and mental mathematical skills. A couple of years after losing his eyesight, he wrote a letter to a German princess describing the fundamentals of many fields that are now grouped under the name classical mechanics. Some of his papers focused on the work of Fermat, often with the intent to prove a theorem that Fermat had created but not proved. This particular work resulted in a paper, published in 1758, called *Theoremata arithmetica nova methodo demonstrata.*It was the result of Euler proving and generalizing Fermat's Little Theorem, hence why it is sometimes called the Euler-Fermat theorem.

RSA

As this paper will show, the encryption technique itself does not directly relate to Euler or Fermat; however, the key generation process heavily relies on concepts tied to their work. To use RSA encryption, you start by selecting two prime numbers, typically denoted as p and q. Ideally, these primes should be very large—exceeding 300 digits—to ensure they are resistant to brute-force attacks. You then multiply them together to obtain n, a crucial component in the encryption and decryption equations. Next, you determine two values: e and d. The value e must be between 1 and $\varphi(n)$ and must be coprime with both n and $\varphi(n)$. Realistically, it should be the largest value that matches this description; however, any value that does will work. Here, $\varphi(n)$ is Euler's totient function, which represents the number of integers less than n that are relatively prime to n. For instance, if $n = 6$, the numbers coprime to 6 between 1 and 6 are 1 and 5, so $\varphi(6) = 2$. The totient function can be calculated more efficiently if n can be factorized as $n = q_1^{e_1} \times q_2^{e_2} \times ... \times q_k^{e_k}$, in which case the totient is:

$$\varphi(n) = \left(q_1^{e_1} - q_1^{e_1-1}\right) \times \left(q_2^{e_2} - q_2^{e_2-1}\right) \times \cdots \times \left(q_k^{e_k} - q_k^{e_k-1}\right)$$

Alternatively, for n as a product of distinct primes, it can be expressed as:

$$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

Now that we have e and $\varphi(n)$, we can solve for d using the following equation: $e \cdot d \equiv 1 \bmod(\varphi(n))$. With these values, you can create your public key (typically used for encryption), which is the pair (e, n), and your private key (typically used for decryption), which is the pair (d, n). Given a numerical message c, one can encrypt the message with this equation:

$$\text{Encrypted message} = c^e \bmod n$$

and then decrypt the message with this one:

$$\text{Decrypted message} = c^d \bmod n$$

RSA encryption relies on several key axioms: (1) Encryption and decryption with the correct public-private key pair are inverses of each other. This ensures that encrypting a plaintext with the public key and then decrypting it with the private key (or vice versa, depending on the use case) recovers the original plaintext. (2) Encrypting a message with the public key does not yield the original message. This is obviously a critical point of encryption. (3) If you have a message c, a public key e, a private key d, and a third (unrelated) key K, encrypting c with the public key and then attempting to decrypt it with K will not yield the original message. This axiom ensures that only the corresponding private key can decrypt the message. Together, these axioms form the basis of a robust cryptographic system. With this background, this paper can examine how Euler and Fermat's work contributed to RSA's development.

Let's walk through an example. Say we have a message that we want to send to a friend. First, we need to calculate a pair of primes $p = 11783$ and $q = 40763$, which are not even close

to large enough for real security but perfect for the example, and compute $n = pq = 480310429$. We also need $\varphi(n) = (11782)(40762) = 480257884$. Using those, we choose an e for a public key, and then compute a d for a private key. Let's say that $e = 3$, which fulfills $1 < e < \varphi(n)$ and is coprime with n and $\varphi(n)$. Then, we get to choose a d which fulfills $3 \cdot d \bmod(\varphi(n)) = 1$; let's go with $d = 320171923$. This gives us our public $(3, n)$ and our private $(320171923, n)$ keys. Now, we take our message, $c = 4321$, and encrypt it $4321^3 \bmod n = 465726518$. This number is the encrypted message, which we send to our friend who will then decrypt it $465726518^{320171923} \bmod n = 4321$. As you can see, our friend gets the original message back.

As said before, the primes chosen to start this process would need to be massive (>300 digits), and likewise for e and d. This would make the process significantly harder for humans, but modern computers are powerful enough to make the process effective. Imagine that someone wanted to discover another person's private key given that they already know the public key; the malicious actor would have to find and test an abhorrent number of prime numbers until they found one of them. The computational impossibility of this task is the backbone of RSA's design and why it has been the reigning champion of cryptography for so long. However, the development of blockchain has produced some worthy contenders.

Euler's Totient Function

As previously noted, Euler's totient function is fundamental to cryptography because it calculates the number of integers relatively prime to a given value. Such a value is extremely useful for calculating the private and public keys. Euler's totient function, also known as Euler's phi function, was first introduced in his 1763 paper *Theoremata arithmetica nova methodo demonstrata* (Demonstration of a New Method in the Theory of Arithmetic). At that time, Euler did not assign it a specific symbol or name. In 1784, he published another paper, *Speculationes circa quasdam insignes proprietates numerorum* (Speculations About Certain Outstanding Properties of Numbers), further exploring the function and designating it with the Greek letter π and describing it as the count of numbers less than a given integer d that share no common divisors with d (other than 1). Later mathematicians refined his definition to specify that $\varphi(1) = 1$, and in the 1800s the symbol shifted from π to the Greek letter φ (phi), which remains in use today. By then, the function had become widely known as Euler's totient function, or simply Euler's totient. The term "totient" aligns with the function's definition, while the "cototient" is defined as $n - \varphi(n)$, intuitively counting the number of positive integers up to n that share at least one prime factor with n.

The Euler-Fermat Theorem

RSA relies on the modulo operator and the properties of modular numbers. Our modern understanding of modulus arithmetic owes much to two theorems: Euler's Theorem and its predecessor, Fermat's Little Theorem. Euler's Theorem (sometimes called the Fermat-Euler Theorem, though this paper refers to it as Euler's Theorem for clarity) also emerged from his 1763 paper from earlier. That paper was the third time that Euler had proven Fermat's Little Theorem. This proof was considered to be Euler's favorite, and in it he generalized Fermat's Little Theorem

into what is now known as Euler's Theorem. Euler's Theorem states that for two relatively prime numbers a and n, $a^{\varphi(n)} \equiv 1 \bmod(n)$. There is also a second version of the theorem that relaxes the coprime condition, stating that if $n = p \cdot q$, $a < n$, and k is an integer, then $a^{k \cdot \varphi(n)+1} \equiv a(\bmod(n))$. Euler's Theorem has several notable properties: First, if n is prime, then $\varphi(n) = n - 1$. Second, the totient function is multiplicative, meaning if m and n are coprime, $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$. Third, as seen above, if n is a product of distinct primes $p_1^{e_1}, p_2^{e_2}, ...$, then $\varphi(n) = n \cdot \prod\left(1 - \frac{1}{p_i}\right)$.

Fermat's Little Theorem was created by Pierre de Fermat in the 17th century. Fermat never proved the theorem in his lifetime, which was possibly the reason that Euler was interested in it. Fermat's Little Theorem became the precursor to Euler's work and states that for an integer a and a prime n, $a^n \equiv a(\bmod(n))$. If a is not divisible by n (i.e., they are coprime), it simplifies to $a^{n-1} \equiv 1(\bmod(n))$. For example, if $n = 5$ and $a = 2$, then $2^5 = 32 \equiv 2(\bmod(5))$ $(32 - 6 \cdot 5 = 2)$, and $2^{5-1} = 16 \equiv 1(\bmod(5))$ $(16 - 3 \cdot 5 = 1)$. In the 1700s, Euler published his first proof of Fermat's Little Theorem. In 1763, he revisited the topic, not only reproving it but also extending it into the generalized form now known as Euler's Theorem. Therefore, Fermat's Little Theorem was the basis for Euler's Totient Theorem and Formula, and by extension, the RSA algorithm.

Calculating Modulo

Euler's Theorem and Fermat's Little Theorem play a key role in calculations involving the modulo operator. A technique known as the Square and Multiply method is widely used to compute expressions of the form $a^n \bmod m$, particularly with large exponents. This method is prudent to the topic of RSA because of the similarity between the encryption and decryption formulas. The Square and Multiply algorithm speeds up modular arithmetic by breaking down exponentiation into squaring and multiplying steps based on the binary representation of the exponent. This is the equation form of the method:

```
INPUT: a, x, n (n is an integer)
OUTPUT: b = a^x mod n
{
    b=1;
    for (i = 0 to m - 1) // m is the number of bits in $x$
    {
        if (x_i = 1) then
            y = y * a mod n
            a = a^2 mod n
        else skip
    }
    return b
}
```

This algorithm illustrates how the square and multiply operates in terms of the number of binary digits of the exponent x. While a detailed explanation of the Square and Multiply method exceeds this paper's scope, it's worth noting that the exponent (x) in $a^x \bmod n$ determines the number

of iterations required. Because the number of operations is equal to the number of bits required to represent x, we can easily calculate the number of iterations using this equation: $\lceil \log_2(x + 1)$. This table shows the increase in operations depending on the exponent calculated with this equation.

x	operations (also m)
10	3
20	5
30	5
40	6
50	6

From this table, we can see that if we reduce the value of x, we can reduce the number of iterations of the Square and Multiply algorithm. Therefore, we can make the whole algorithm more efficient if we can efficiently shrink the exponent. Interestingly, both Fermat's Little Theorem and Euler's Theorem can be applied to achieve this.

In the paper textit(Exponential Simplification Using Euler's and Fermat's Theorems) by Mohamed et al., the authors demonstrate how these theorems can reduce the exponent and therefore optimize the Square and Multiply method. For instance, under Fermat's Little Theorem, if $n > p - 1$ (where p is a prime and a and p are coprime), then the exponent can be simplified from n to $n - (p - 1)$. Applying the Square and Multiply method with this simplified exponent will result in the same value and require fewer multiplications. The paper also outlines additional conditions and describes a similar, though slightly more complex, application of Euler's Theorem. Both approaches reduce computational complexity by lowering the exponent. The primary distinction between the two theorems lies in their applicability: Euler's Theorem applies when a and n are positive integers that are relatively prime, using $a^{\varphi(n)} \equiv 1(\mathrm{mod}(n))$, while Fermat's Little Theorem requires n to be prime, resulting in $a^{n-1} \equiv 1(\mathrm{mod}(n))$ when a and n are coprime. This is the main difference between them; otherwise, they operate with similar time complexity and achieve functionally equivalent results.

Conclusion

The evolution of cryptography, as shown in this paper, follows from the early steganographic and cipher techniques of Arabic scholars to the military applications of the Spartan skytale and the Enigma machine. The introduction of the Vigenère cipher's autokey method was a significant advancement, bringing a randomized result that fixed previous weaknesses. Also, the advent of computers and numerical encoding systems like ASCII were crucial to modern cryptographic paradigms' development. Rivest, Shamir, and Adleman's algorithm, was created with those encodings in mind. Their work used that freedom to make use of Euler's totient function and the Euler-Fermat theorem. Even after its creation, others have found use in the Euler-Fermat theorem to aid the Square and Multiply method. To this day, RSA is a powerful tool that ensures safety and security for governments, corporations, and individuals on the internet.

References

- Mohan, M., Devi, M. K. K., & Jeevan Prakash V. (2016). *Exponential Simplification Using Euler's and Fermat's Theorem*. Procedia Computer Science, 78, 714–721. (https://doi.org/10.1016/j.procs.2016.02.029).
- Euler, L. (1763). *Zahlentheoretische Theoreme, mit einer neuen Mehthode bewiesen.*
- Euler, L. (2007). *Speculations on some characteristic properties of numbers* (arXiv:0705.3929). arXiv. (https://doi.org/10.48550/arXiv.0705.3929).
- Diffie, W., Hellman, M.. (1976). *New directions in cryptography*. (https://doi:10.1109/tit.1976.1055638).
- Kahn, D. (1996). *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet (2nd ed)*. Scribner.
- By an unknown officer or employee of the United States Government - MIL-STD-188-100, pg. B-2, Fig 1, 1972. (different scan), Public Domain, https://commons.wikimedia.org/w/index.php?curid=63485656