Contents

Chapter 1

Cryptography

1.1 What is cryptography

Cryptography is the science of secret communications. It is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents. The term is derived from the Greek word kryptos, which means hidden. It is closely associated to encryption, which is the act of scrambling ordinary text into what's known as ciphertext and then back again upon arrival. In addition, cryptography also covers the obfuscation of information in images using techniques such as microdots or merging. Ancient Egyptians were known to use these methods in complex hieroglyphics, and Roman Emperor Julius Caesar is credited with using one of the first modern ciphers.

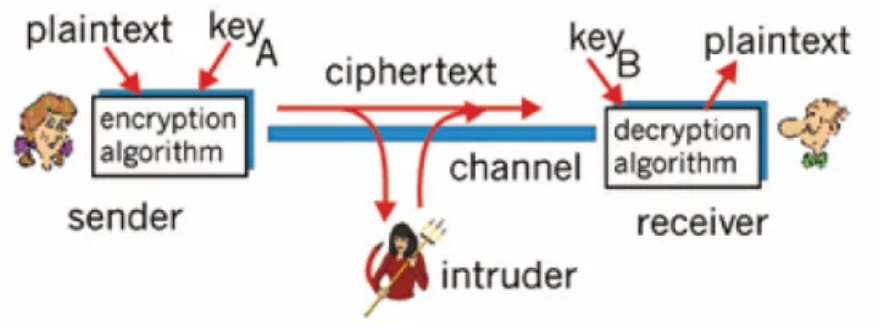
Long before the information age, cryptography was used only to ensure secrecy of information. Encryption was used to ensure confidentiality in communications by spies, military leaders and diplomats. The Egyptian hieroglyphs, the scytale transposition cipher used by the Spartans of Greece, waxed seals and different physical devices to assist with ciphers were used throughout history right up to modern times. These devices underwent further changes when computers and electronics came into the picture, immensely helping in cryptanalysis.Cryptography has become more mathematical now and also finds applications in day-to-day security. It helps you safely transfer or withdraw money electronically and you'd be hard-pressed to come

across an individual without a credit or debit card. The public-key encryption system introduced the concept of digital signatures and electronic credentials. Cryptography has a definitive existence in our lives today and the whole system will crumble in its absence. Let's now discuss the varied uses of cryptography in modern times and its intersection with computer science.

1.2 Usage of cryptography

1.2.1 Secrecy in transmission

The major goal of cryptography is to prevent data from being read by any third party. Most transmission systems use a private-key cryptosystem. This system uses a secret key to encrypt and decrypt data which is shared between the sender and receiver. The private keys are distributed and destroyed periodically. One must secure the key from unauthorized access, because any party that has the key can decrypt the encrypted information.



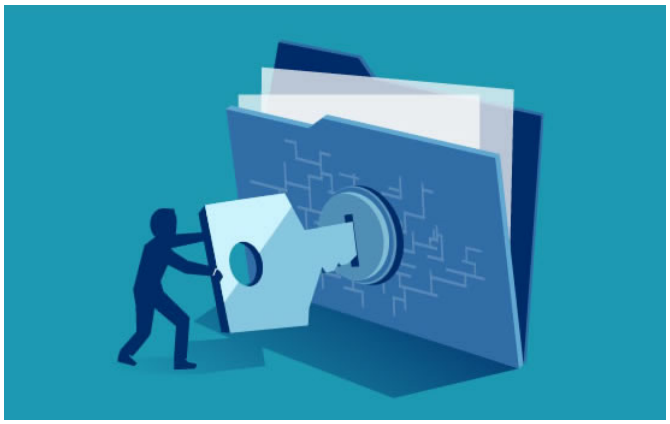
Alternately a key-generating-key, called a master key, can be used to electronically generate a one-time session-key for every transaction. The secrecy of the master-key should be maintained by all parties privy to the information. The disadvantage of this method is there's too much hope riding on the master-key, which if cracked, collapses the entire system.

A better method is to use a public-key cryptosystem. In this system, data can be encrypted by anyone with the public-key, but it can be decrypted only by using the private-key, and data that is signed with the private key can be verified only with the public key. With the development of publickey systems, secrecy can be maintained without having to keep track of a large number of keys or sharing a common master-key. If, say, Alex wants to communicate with Neil, she first generates her public/private key pair and sends the public key to Neil over a non-secure channel. Neil to encrypt information and sends it back to Alex. Only Alex has the private key with which she can decrypt the information. Anyone who intercepts the public key or the encrypted data can't decrypt the message due to the protocols followed during information transfer.

1.2.2 Secrecy in storage

Storage encryption refers to the application of cryptographic techniques on data, both during transit and while on storage media. Storage encryption is gaining popularity among enterprises that use storage area networks (SANs). Secrecy in storage is maintained by storing

data in encrypted form. The user has to provide the key to the computer only at the beginning of a session to access the data and it then takes care of encryption and decryption throughout the course of normal use. Hardware devices can also be used for PCs to automatically encrypt all information stored on disk. When the computer is turned on, the user must supply a key to the encryption hardware. The information is plain gibberish without its key thus preventing misuse if the disk is stolen.



Multiple ciphers can be used for individual files and folders. The ciphers and keys should be changed frequently to ensure security of data. However, if the user forgets a key, all of the information encrypted with it makes no sense and is rendered useless. This is why backups of encrypted information are advised to be stored in plaintext. The data is only encrypted while in storage, not when in use. This leaves a loophole for the attackers. The system is vulnerable to a security breach if the encryption and decryption are done in software, or if the key is stored somewhere in the system.

1.2.3 Integrity in transmission

We can use cryptography to provide a means to ensure that data is not altered during transmission, i.e. its integrity is preserved. In electronic funds transfer, it is very important that integrity be maintained. A bank can lose millions if a transaction is illicitly intercepted.

Cryptographic techniques are employed to prevent accidental or intentional modification of data during transmission, leading to erroneous actions. One of the ways to ensure integrity is to perform a checksum on the information being transmitted and to transmit the checksum in an encrypted form as well.

The information is received on the other end and again checksummed. The transmitted checksum is decrypted and compared with the previous checksum. If the checksums agree, the information is most likely unaltered. The problem with this scheme is that the checksum of the original message can be known and another message with the same checksum can be generated and sent instead of the original one. This problem can be overcome by using a public-key cryptosystem. After generating the public-key/private-key pair, if we throw away the private-key and use only the public-key to encrypt the checksum, the checksum becomes impossible to decrypt. In order to verify the checksum, we generate a new checksum for the received information, encrypt it using the public-key and match it with the encrypted checksum. This is also known as a one-way function as it is hard to invert.

1.2.4 Integrity in storage

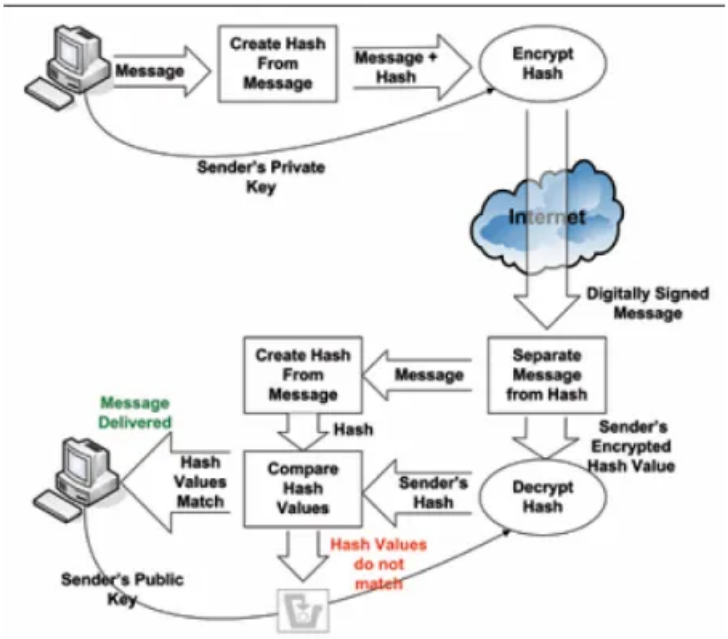
Integrity in storage had been ensured by access control systems with lock and keys and other guards to prevent unauthorized access to stored data. The existence of computer viruses has changed the scenario and the need of integrity against intentional attack has become a problem of epic proportions. Cryptographic checksums to ascertain validity of stored data are of help here. As in the case of transmission, a cryptographic checksum is produced and compared to the expected value. However, storage media are more vulnerable to attacks than transmission channels due to longer exposure and larger volumes of information.

1.2.5 Authentication of identity

Authentication is the process of verifying if the user has enough authority for data access. Simple passwords are used to identify someone. You must also have seen in classic gangster movies, the exchange of keywords to prove identity. Cryptography is similar to the practice of providing passwords for identity authentication. Modern systems use cryptographic transforms in conjunction with other characteristics of individuals to provide more reliable and efficient authentication of identity. Many systems allow passwords to be stored in an encrypted form, with read access available to all programs which may use them. Since passwords are not stored as plaintext, an accidental of data doesn't compromise the system's security.Passwords are analogous to the key in a cryptosystem that allows encryption and decryption of anything the password has access to. The principal element of this system is the password selection process. And that's a whole other subject that we can't cover here. But in a nutshell, the longer the password, the more random it will be and the harder it is to guess. So if you think it's easy for you to remember, you should know that it will be all the easier to crack.

1.2.6 Digital signatures

A digital signature is a mechanism by which a message is authenticated i.e. proving that a message is coming from a given sender, much like a signature on a paper document. To be as effective as a signature on paper, digital signatures must be hard to forge and accepted in a court of law as binding upon all parties to the transaction. The need for digital signatures arises when the parties dealing in a transaction are not physically close, and the volume of paperwork is high, in other words big business dealings. Digital signatures can be created using a public key cryptosystem and hashing process.



Hashing produces a message digest that is a small and unique representation of the original message. Hashing is a one-way algorithm, i.e. the message can't be derived from the digest. Let's say that Alex is sending a message to Neil. Alex first hashes the message to produce a digest, and then encrypts the digest with her private-key to create her personal signature; the public-key and hash algorithm are appended to it. The whole message including the digest is then encrypted using a one-time symmetric-key which is known only to Alex and Neil. Neil decrypts the message using the symmetric-key. He then decrypts the message digest using the public-key. He would then hash the original message using the same hash algorithm (whose name was appended in the message) with which it was previously hashed. If the evaluated digest and decrypted digest match, then the signature has been verified and the recipient would be sure that the message integrity has been preserved.

Another aspect of this system is the non-repudiation of digital signatures. Since the private-key is only privy to the sender, he can't deny signing the message. Also, a digital signature can be verified by anyone using the sender's public-key which is usually included in the

digital signature format.

1.3 Types of cryptography

In general there are three types Of cryptography:

1.3.1 Symmetric Key Cryptography:

It is an encryption system where the sender and receiver of message use a single common key to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner. The most popular symmetric key cryptography system is Data Encryption System(DES).

All cryptographic algorithms involve substituting one thing for another, for exam- ple, taking a piece of plaintext and then computing and substituting the appropriate ciphertext to create the encrypted message. Before studying a modern key-based cryptographic system, let us first get our feet wet by studying a very old, very simple symmetric key algorithm attributed to Julius Caesar, known as the Caesar cipher (a cipher is a method for encrypting data). For English text, the Caesar cipher would work by taking each letter in the plain- text message and substituting the letter that is k letters later (allowing wraparound; that is, having the letter z followed by the letter a) in the alphabet. For example, if k = 3, then the letter a in plaintext becomes d in ciphertext; b in plaintext becomes e in ciphertext, and so on. Here, the value of k serves as the key. As an example, the plaintext message "bob, i love you. Alice" becomes "ere, l oryh brx. dolfh" in ciphertext. While the ciphertext does indeed look like gibberish, it wouldn't take long to break the code if you knew that the Caesar cipher was being used, as there are only 25 possible key values. An improvement on the Caesar cipher is the monoalphabetic cipher, which also substitutes one letter of the alphabet with another letter of the alphabet. However, rather than substituting according to a

regular pattern (e.g., substitution with an offset of k for all letters), any letter can be substituted for any other letter, as long as each letter has a unique substitute letter, and vice versa. The substitution rule in Figure 8.3 shows one possible rule for encoding plaintext. The plaintext message "bob, i love you. Alice" becomes "nkn, s gktc wky. Mgsbc." Thus, as in the case of the Caesar cipher, this looks like gibberish. A monoalphabetic cipher would also appear to be better than the Caesar cipher in that there are 26! (on the order of 10 26) possible pairings of letters rather than 25 possible pairings. A brute-force approach of trying all 10 26 possible pairings.

Plaintext letter:	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
Ciphertext letter:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

would require far too much work to be a feasible way of breaking the encryption algorithm and decoding the message. However, by statistical analysis of the plain- text language, for example, knowing that the letters e and t are the most frequently occurring letters in typical English text (accounting for 13 percent and 9 percent of letter occurrences), and knowing that particular two-and three-letter occurrences of letters appear quite often together (for example, "in," "it," "the," "ion," "ing," and so forth) make it relatively easy to break this code. If the intruder has some knowledge about the possible contents of the message, then it is even easier to break the code. For example, if Trudy the intruder is Bob's wife and suspects Bob of having an affair with Alice, then she might suspect that the names "bob" and "alice" appear in the text. If Trudy knew for certain that those two names appeared in the ciphertext and had a copy of the example ciphertext message above, then she could immedi- ately determine seven of the 26 letter pairings, requiring 10 9 fewer possibilities to be checked by a brute-force method. Indeed, if Trudy suspected Bob of having an affair, she might well expect to find some other choice words in the message as well. When considering how easy it might be for Trudy to break Bob and Alice's encryption scheme, one can distinguish three

different scenarios, depending on what information the intruder has.

- emphCiphertext-only attack. In some cases, the intruder may have access only to the intercepted ciphertext, with no certain information about the contents of the plain- text message. We have seen how statistical analysis can help in a ciphertext-only attack on an encryption scheme.
- *Known-plaintext attack.* We saw above that if Trudy somehow knew for sure that "bob" and "alice" appeared in the ciphertext message, then she could have determined the (plaintext, ciphertext) pairings for the letters a, l, i, c, e, b, and o. Trudy might also have been fortunate enough to have recorded all of the cipher- text transmissions and then found Bob's own decrypted version of one of the transmissions scribbled on a piece of paper. When an intruder knows some of the (plaintext, ciphertext) pairings, we refer to this as a known-plaintext attack on the encryption scheme.
- *Chosen-plaintext attack.* In a chosen-plaintext attack, the intruder is able to choose the plaintext message and obtain its corresponding ciphertext form. For the simple encryption algorithms we've seen so far, if Trudy could get Alice to send the message, "The quick brown fox jumps over the lazy dog," she could completely break the encryption scheme. We'll see shortly that for more sophisticated encryption techniques, a chosen-plaintext attack does not necessarily mean that the encryption technique can be broken.

Five hundred years ago, techniques improving on monoalphabetic encryp- tion, known as polyalphabetic encryption, were invented. The idea behind polyalphabetic encryption is to use multiple monoalphabetic ciphers, with a specific

Plaintext letter:	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
$C_1(k = 5)$:	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
$C_2(k = 19)$:	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

Block Ciphers

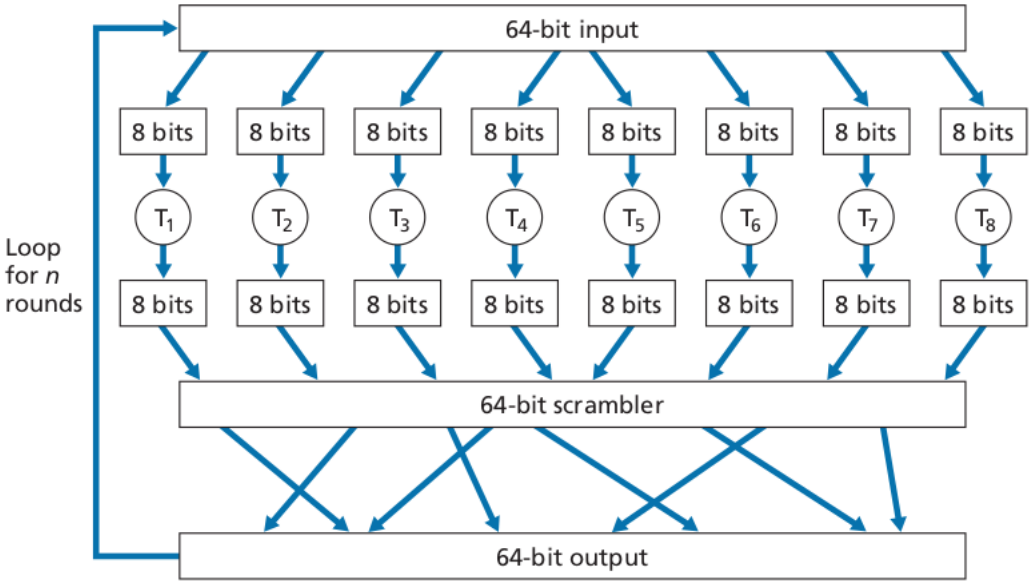
Let us now move forward to modern times and examine how symmetric key encryp- tion is done today. We focus on block ciphers, which are used in many secure Internet protocols, including PGP (for secure e-mail), TLS (for securing TCP connections), and IPsec (for securing the network-layer transport). In a block cipher, the message to be encrypted is processed in blocks of k bits. For example, if k = 64, then the message is broken into 64-bit blocks, and each block is encrypted independently. To encode a block, the cipher uses a one-to-one map- ping to map the k-bit block of cleartext to a k-bit block of ciphertext. Let's look at an example. Suppose that k = 3, so that the block cipher maps 3-bit inputs (cleartext) to 3-bit outputs (ciphertext). One possible mapping is given in Table 8.1. Notice that this is a one-to-one mapping; that is, there is a different output for each input. This block cipher breaks the message up into 3-bit blocks and encrypts each block accord- ing to the above mapping. You should verify that the message 010110001111 gets encrypted into 101000111001. Continuing with this 3-bit block example, How many possible mappings are there? To answer this question, observe that a mapping is nothing more than a permu- tation of all the possible inputs. There are $2^3 (= 8)$ possible inputs (listed under the

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

input columns). These eight inputs can be permuted in 8! = 40,320 different ways. Since each of these permutations specifies a mapping, there are 40,320 possible map- pings. We can view each of these mappings as a key—if Alice and Bob both know the mapping (the key), they can encrypt and decrypt the messages sent between them. The

brute-force attack for this cipher is to try to decrypt ciphtertext by using all mappings. With only 40,320 mappings (when k = 3), this can quickly be accom- plished on a desktop PC. To thwart brute-force attacks, block ciphers typically use much larger blocks, consisting of k = 64 bits or even larger. Note that the number of possible mappings for a general k-block cipher is 2 k !, which is astronomical for even moderate values of k (such as k = 64). Although full-table block ciphers, as just described, with moderate values of k can produce robust symmetric key encryption schemes, they are unfortunately dif- ficult to implement. For k = 64 and for a given mapping, Alice and Bob would need to maintain a table with 2 64 input values, which is an infeasible task. Moreover, if Alice and Bob were to change keys, they would have to each regenerate the table. Thus, a full-table block cipher, providing predetermined mappings between all inputs and outputs (as in the example above), is simply out of the question. Instead, block ciphers typically use functions that simulate randomly permuted tables. An example (adapted from [Kaufman 2002]) of such a function for k = 64 bits is shown in Figure 8.5. The function first breaks a 64-bit block into 8 chunks, with each chunk consisting of 8 bits. Each 8-bit chunk is processed by an 8-bit to 8-bit table, which is of manageable size. For example, the first chunk is processed by the table denoted by T 1 . Next, the 8 output chunks are reassembled into a 64-bit block. The positions of the 64 bits in the block are then scrambled (permuted) to produce a 64-bit output. This output is fed back to the 64-bit input, where another cycle begins. After n such cycles, the function provides a 64-bit block of ciphertext. The purpose of the rounds is to make each input bit affect most (if not all) of the final output bits. (If only one round were used, a given input bit would affect only 8 of the 64 output bits.) The key for this block cipher algorithm would be the eight permutation tables (assuming the scramble function is publicly known). Today there are a number of popular block ciphers, including DES (standing for Data Encryption Standard), 3DES, and AES (standing for Advanced Encryption Standard). Each of these standards uses functions, rather than predetermined tables, along the

lines of Figure 8.5 (albeit more complicated and specific to each cipher). Each of these algorithms also uses a string of bits for a key. For example, DES uses 64-bit blocks with a 56-bit key. AES uses 128-bit blocks and can operate with keys that are 128, 192, and 256 bits long. An algorithm's key determines the specific "mini-table" mappings and permutations within the algorithm's internals. The brute- force attack for each of these ciphers is to cycle through all the keys, applying the decryption algorithm with each key. Observe that with a key length of n, there are 2 n possible keys. NIST [NIST 2001] estimates that a machine that could crack 56-bit DES in one second (that is, try all 2 56 keys in one second) would take approximately 149 trillion years to crack a 128-bit AES key.



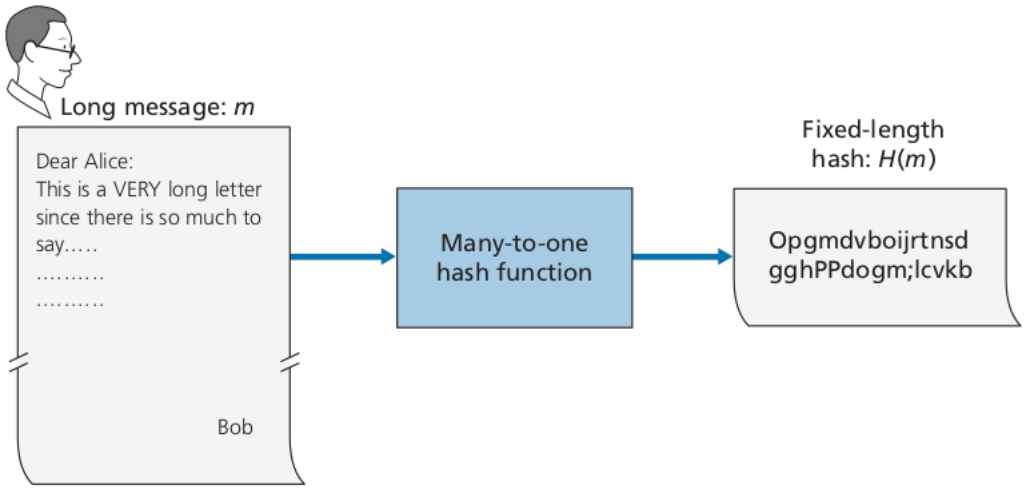
1.3.2 Hash Functions:

There is no usage of any key in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords. a hash function takes an input, m, and computes a fixed-size string H(m) known as a hash. A cryp-

tographic hash function is required to have the follow- ing additional property:

- It is computationally infeasible to find any two different messages x and y such that H(x) = H(y).

Informally, this property means that it is computationally infeasible for an intruder to substitute one message for another message that is protected by the hash function. That is, if (m, H(m)) are the message and the hash of the message created by the sender, then an intruder cannot forge the contents of another message, y, that has the same hash value as the original message. Let's convince ourselves that a simple checksum, such as the Internet checksum, would make a poor cryptographic hash function. Rather than performing 1s comple- ment arithmetic (as in the Internet checksum), let us compute a checksum by treating each character as a byte and adding the bytes together using 4-byte chunks at a time. Suppose Bob owes Alice $100.99 and sends an IOU to Alice consisting of the text string "IOU100.99BOB." The ASCII representation (in hexadecimal notation) for these letters is 49,4F,55,31,30,30,2E,39,39,42,4F,42.



Message	ASCII Representation				
I O U 1	49	4F	55	31	
0 0 . 9	30	30	2E	39	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Message	ASCII Representation				
I O U 9	49	4F	55	39	
0 0 . 1	30	30	2E	31	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

The messages "IOU100.99BOB" and "IOU900.19BOB" have the same checksum. Thus, this simple checksum algorithm violates the requirement above. Given the original data, it is simple to find another set of data with the same checksum. Clearly, for security purposes, we are going to need a more powerful hash function than a checksum. The MD5 hash algorithm of Ron Rivest is in wide use today. It computes a 128-bit hash in a four-step process consisting of a padding step (adding a one followed by enough zeros so that the length of the message satisfies certain conditions), an append step (appending a 64-bit representation of the mes- sage length before padding), an

initialization of an accumulator, and a final loop- ing step in which the message's 16-word blocks are processed (mangled) in four rounds. The second major hash algorithm in use today is the Secure Hash Algorithm (SHA-1) . This algorithm is based on principles similar to those used in the design of MD4 , the predecessor to MD5. SHA-1, a US federal standard, is required for use whenever a cryptographic hash algorithm is needed for federal applications. It produces a 160-bit message digest. The longer output length makes SHA-1 more secure.

1.3.3 Asymmetric Key Cryptography:

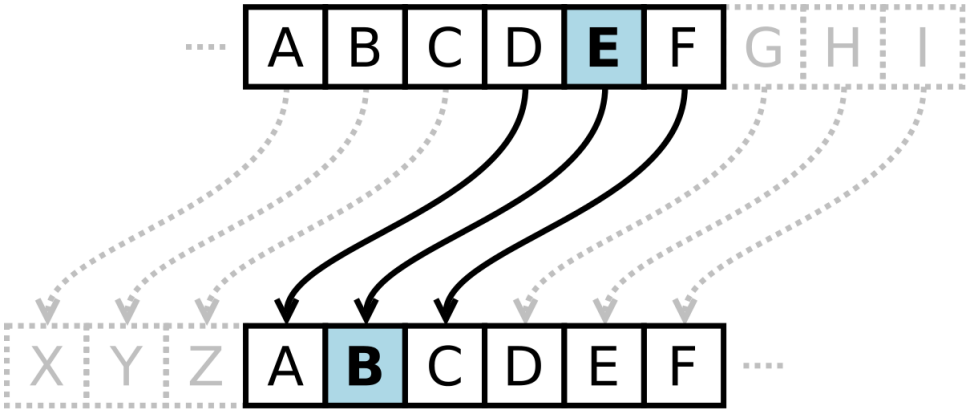
Under this system a pair of keys is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows the private key.Later we will discuss broadly.

1.4 History of cryptography

Cryptology was well established in ancient times, with both Greeks and Romans practicing different forms of cryptography. With the fall of the Roman Empire, cryptology was largely lost in the West until the Renaissance, but it flour- ished in the Arabic world. The Arabs invented the first reliable tool for cryptanaly- sis, frequency analysis. With the end of the Middle Ages and the increase in commerce and diplomacy, cryptology enjoyed a Renaissance of it's own in the West

Julius Caesar, probably the greatest of all Roman generals, was no stranger to cryp- tology. In his famous Commentary on the Gallic Wars, Caesar himself describes using a form of a cipher to hide a message If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to deci- pher these, and get at their meaning, he must substitute the fourth letter of the alpha-

bet, namely D, for A, and so with the others." (Seutonius 1957, Ch. 56) This is the first written description of the modern monoalphabetic substitution cipher using a shifted standard alphabet. Using Caesar's cipher, the cipher alphabet looks like



For 900 years the monoalphabetic substitution cipher was the strongest cipher sys- tem in the Western world. The Romans used it regularly to protect their far-flung lines of communication. But after the fall of the Western Roman Empire in 476 C.E. the knowledge of cryptology vanished from the West and wasn't to return until the Renaissance. Indeed, with the decline of literacy and scholarship in Europe during the Dark Ages following the fall of Rome cryptology turned from a useful technique for keeping communications secret into a dark art that bordered on magic. But interest in cryptology was not dead. In the latter part of the first millennium, there was another place where intellectual curiosity and scholarship flowered and where mathematics and cryptology saw their biggest advances since Caesar – the Arab world. And it was the Arab world from which the next big advance in cryptanalytic techniques would come. The period around the ninth century C.E. is considered to be the beginning of the Islamic Golden Age, when philosophy, science, literature, mathematics, and reli- gious studies all flourished in what was then the peace and prosperity of the Abbasid Caliphate. Into this period was born Abu Yūsuf Ya-qūb ibn Isāq as-Sabbāh al-Kindi (801–873 C.E.), a polymath who was the philosopher of the age. Al-Kindi wrote books in many disciplines including as-

tronomy, optics, philosophy, mathematics, medicine, and linguistics, but his book on secret messages for court secretaries, A Manuscript on Deciphering Cryptographic Messages is the most important to the history of cryptology. It is in this book that the technique of frequency analysis is first described.

Although cryptography has a long and complex history, it wasn't until the 19th century that it developed anything more than ad hoc approaches to either encryption or cryptanalysis (the science of finding weaknesses in crypto systems). Examples of the latter include Charles Babbage's Crimean War era work on mathematical cryptanalysis of polyalphabetic ciphers, redeveloped and published somewhat later by the Prussian Friedrich Kasiski. Understanding of cryptography at this time typically consisted of hard-won rules of thumb; see, for example, Auguste Kerckhoffs' cryptographic writings in the latter 19th century. Edgar Allan Poe used systematic methods to solve ciphers in the 1840s. In particular he placed a notice of his abilities in the Philadelphia paper Alexander's Weekly (Express) Messenger, inviting submissions of ciphers, of which he proceeded to solve almost all. His success created a public stir for some months.[24] He later wrote an essay on methods of cryptography which proved useful as an introduction for novice British cryptanalysts attempting to break German codes and ciphers during World War I, and a famous story, The Gold-Bug, in which cryptanalysis was a prominent element.

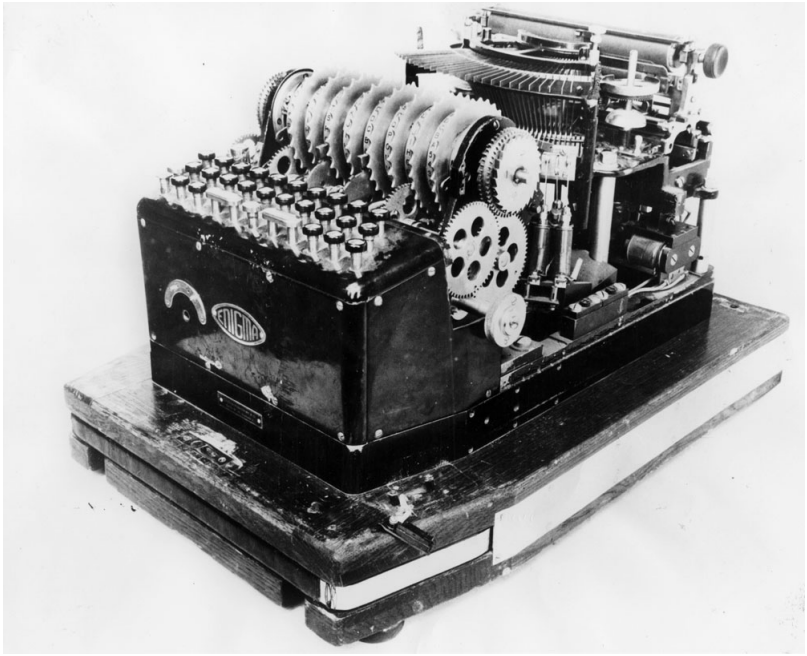
Cryptography, and its misuse, were involved in the execution of Mata Hari and in Dreyfus' conviction and imprisonment, both in the

early 20th century. Cryptographers were also involved in exposing the machinations which had led to the Dreyfus affair; Mata Hari, in contrast, was shot.

In World War I the Admiralty's Room 40 broke German naval codes and played an important role in several naval engagements during the war, notably in detecting major German sorties into the North Sea that led to the battles of Dogger Bank and Jutland as the British fleet was sent out to intercept them. However its most important contribution was probably in decrypting the Zimmermann Telegram, a cable from the German Foreign Office sent via Washington to its ambassador Heinrich von Eckardt in Mexico which played a major part in bringing the United States into the war.

In 1917, Gilbert Vernam proposed a teleprinter cipher in which a previously prepared key, kept on paper tape, is combined character by character with the plaintext message to produce the cyphertext. This led to the development of electromechanical devices as cipher machines, and to the only unbreakable cipher, the one time pad.

During the 1920s, Polish naval-officers assisted the Japanese military with code and cipher development.



Mathematical methods proliferated in the period prior to World War II (notably in William F. Friedman's application of statistical techniques to cryptanalysis and cipher development and in Marian Rejewski's initial break into the German Army's version of the Enigma system in 1932).

1.5 The Math Behind Cryptography

Secrecy systems based on modular arithmetic. The first of these had its origin with Julius Caesar; the newest systems that we will discuss were invented in the late 1970s. In all these systems, we start by translating letters into numbers. We take as our standard alphabet the letters of English and translate them into the integers from 0 to 25, as shown below

Letter	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
Numerical Equivalent	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Of course, if we were sending messages in Russian, Greek, Hebrew, or any other language, we would use the appropriate alphabet and

range of integers. Also, we may want to include all ASCII characters, including punctuation marks, a symbol to indicate blanks, and the digits for representing numbers as part of the message. However, for the sake of simplicity, we restrict ourselves to the letters of the English alphabet. The transformation of letters to numbered equivalents can be done in many other ways (including translation to bit strings). Here we have chosen a simple and easily understood transformation for simplicity.First, we discuss secrecy systems based on transforming each letter of the plaintext message into a different letter (or possibly the same) to produce the ciphertext. The en cryption methods in these cryptosystems are called character, or monographic, ciphers,because each character is changed individually to another letter by a substitution. Alto gether, there are 26 ! possible ways to produce a mono graphic transformation. We will discuss some particular monographic transformations based on modular arithmetic. Julius Caesar used a cipher based on the substitution in which each letter is replaced by the letter three further down the alphabet, with the last three letters shifted to the first three letters of the alphabet. To describe this cipher using modular arithmetic, let P be the numerical equivalent of a letter in the plaintext and C be the numerical equivalent of the corresponding ciphertext letter. Then

$C \equiv P + 3 \pmod{26}$

The correspondence between plaintext and ciphertext is given below

	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
Plaintext	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2
Ciphertext	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

To encrypt a message using this transformation, we first change it to its numerical equivalent, grouping letters in blocks of five. Then we transform each number. The group ing of letters into blocks helps to prevent successful cryptanalysis based on recognizing particular

words.

1.5.1 Affine Transformation

The Caesar cipher is one of a family of similar ciphers described by a shift transformation.

$$C \equiv P + 3 \pmod{26}$$

where k is the key representing the size of the shift of letters in the alphabet. There are 26 different transformations of this type, including the case of $K \equiv 0 \pmod{26}$ Where letters are not altered, because in this case $C \equiv P \pmod{26}$ More generally, we will consider transformations of the type

$$C \equiv aP + b \pmod{26}$$

Where a and b are integers with (a, 26) = 1. These are called affine transformations.Shift transformations are affine transformations with $a = 1$. We require that $(a, 26) = 1$, so that as P runs through a complete system of residues modulo 26, C also does. There are $\phi(26) = 12$ choices for a, and 26 choices for b, giving a total of $12 * 26 = 312$ transformations of this type (one of these is $C \equiv P \pmod{26}$ obtained when $a = 1$ and b = 0). If the relationship between plaintext and ciphertext is described above then the inverse relationship is given by

$$P \equiv a^*(C - b) \pmod{26}$$

where a^* is an inverse of a (mod 26), which can be found using the congruence

$$a^* \equiv a^{\phi(26)-1} = a^{11} \pmod{26}$$

.

Example 1.5.1. To encrypt the message

APPLIED MATHEMATICS

Converting the letters into their numerical equivalents, we obtain

1 16 16 12 9 5 4 -64(this is the space value output by programe) 13 1 20 8 5 13 1 20 9 3 19

Using the Caesar transformation $C \equiv P + 3(mod\ 26)$, this become 4 19 19 15 12 8 7 -61(this is the space value output by programe), 16 4 23 11 8 16 4 23 12 6 22

Translating back to letters, we have

DSSOLHG PDWKHPDWLFV

This is the encrypted message.

Example 1.5.2. To decrypt the message

DSSOLHG PDWKHPDWLFV

encrypted by the Caesar cipher, we first change these letters into their numerical equiv alents, to obtain
4 19 19 15 12 8 7 -61(this is the space value output by programe), 16 4 23 11 8 16 4 23 12 6 22
Next, we perform the transformation $P \equiv C + 3(mod\ 26)$ to change this to plaintext,and we obtain
1 16 16 12 9 5 4 -64(this is the space value output by programe) 13 1 20 8 5 13 1 20 9 3 19

We translate this back to letters and recover the plaintext message.

APPLIED MATHEMATICS

Python code that was used for generating that code given below

File Edit View Search Terminal Help

```
1 def char2int(char):
2     trans_number=[]
3     for c in char.lower():
4         trans_number.append(ord(c)-96)
5     return trans_number
~
~
~
~
~
```

```
1 def shifter(arr,num=3):
2     modified_arr=[]
3     for i in arr:
4         modified_arr.append(i+num)
5     return modified_arr
~
~
```

```
1 def int2char(arr):
2     result=""
3     for i in arr:
4         result+=chr(i+96)
5     return result.upper()
```

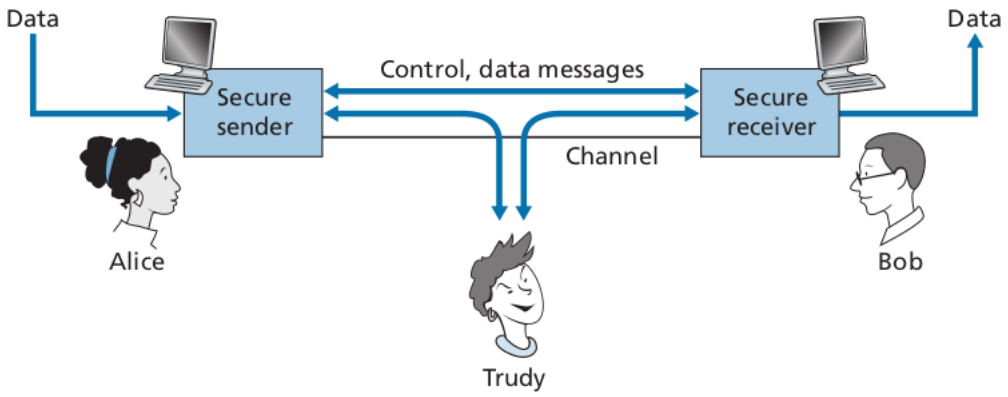
Chapter 2

Application Of Cryptography

2.1 Network Security

Secure communication through computer is very important in our modern days.Cryptography can handle the problem of secure communication.Let us introduce Alice and Bob, two people who want to communicate and wish to do so securely.But what does it precisely mean to communicate securely!!.

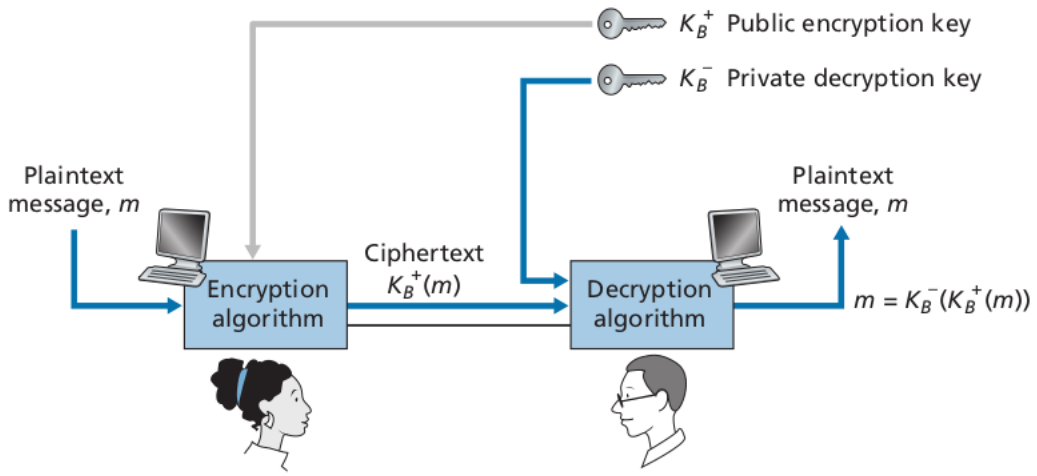
- Alice wants only Bob to be able to understand a message that she has sent, even though they are communicating over an insecure medium where an intruder (Trudy, the intruder) may intercept whatever is transmitted from Alice to Bob.
- Bob also wants to be sure that the message he receives from Alice was indeed sent by Alice, and Alice wants to make sure that the person with whom she is communicating is indeed Bob.



2.2 Public Key Encryption

The use of public key cryptography is conceptually quite simple. Suppose Alice wants to communicate with Bob. Rather than Bob and Alice sharing a single secret key (as in the case of symmetric key systems), Bob (the recipi- ent of Alice's messages) instead has two keys—a public key that is available to everyone in the world (including Trudy the intruder) and a private key that is known only to Bob. We will use the notation K_B^+ and K_B^- to refer to Bob's public and pri- vate keys, respectively. In order to communicate with Bob, Alice first fetches Bob's public key. Alice then encrypts her message, m, to Bob using Bob's public key and a known (for example, standardized) encryption algorithm; that is, Alice computes $K_B^+(m)$. Bob receives Alice's encrypted message and uses his private key and a known (for example, standardized) decryption algorithm to decrypt Alice's encrypted mes- sage. That is, Bob computes $K_B^-(K_B^+(m))$. We will see below that there are encryption decryption algorithms and techniques for choosing public and private keys such that $K_B^-(K_B^+(m)) = m$; that is, applying Bob's public key, K_B^+ , to a message, m (to get $K_B^+(m)$), and then applying Bob's private key, K_B^- , to the encrypted version of m (that is, computing $K_B^-(K_B^+(m))$) gives back m. This is a remarkable result! In this manner, Alice can use Bob's publicly available key to send a secret message to Bob without either of them having

to distribute any secret keys! We will see shortly that we can interchange the public key and private key encryption and get the same remarkable result—that is, $K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m$.



Although public-key cryptography is appealing, one concern immediately springs to mind. Since Bob's encryption key is public, anyone can send an encrypted message to Bob, including Alice or someone pretending to be Alice. In the case of a single shared secret key, the fact that the sender knows the secret key implicitly identifies the sender to the receiver. In the case of public key cryptography, however, this is no longer the case since anyone can send an encrypted message to Bob using Bob's publicly available key. We introduce digital signature to solve this problem.

2.2.1 RSA

While there may be many algorithms that address these concerns, the RSA algorithm (named after its founders, Ron Rivest, Adi Shamir, and Leonard Adleman) has become almost synonymous with public key cryptography. Let's first see how RSA works and then examine why it works. RSA makes extensive use of arithmetic operations using modulo-n arithmetic. So let's briefly review modular arithmetic.

Recall that x mod n simply means the remainder of x when divided by n; so, for example, 19 mod 5 = 4. In modular arith- metic, one performs the usual operations of addition, multiplication, and exponentiation. However, the result of each operation is replaced by the integer remainder that is left when the result is divided by n. Adding and multiplying with modular arithmetic is facilitated with the following handy facts:

$$[(a \ mod \ n) \ + \ (b \ mod \ n)] \ mod \ n = (a \ + \ b) \ mod \ n$$

$$[(a \ mod \ n) \ - \ (b \ mod \ n)] \ mod \ n = (a \ - \ b) \ mod \ n$$

$$[(a \ mod \ n) * (b \ mod \ n)] \ mod \ n = (a * b) \ mod \ n$$

It follows from the third fact that (a mod n)^d mod n = a^d mod n, which is an identity that we will soon find very useful. Now suppose that Alice wants to send to Bob an RSA-encrypted message, let's always keep in mind that a mes- sage is nothing but a bit pattern, and every bit pattern can be uniquely represented by an integer number (along with the length of the bit pattern). For example, suppose a message is the bit pattern 1001; this message can be represented by the decimal integer 9. Thus, when encrypting a message with RSA, it is equivalent to encrypting the unique integer number that represents the message. There are two interrelated components of RSA

- The choice of the public key and the private key
- The encryption and decryption algorithm

To generate the public and private RSA keys, Bob performs the following steps:

1. Choose two large prime numbers, p and q. How large should p and q be? The larger the values, the more difficult it is to break RSA, but the longer it takes to perform the encoding and decoding. RSA Laboratories recommends that the product of p and q be on the order of 1,024 bits. For a discussion of how to find large prime numbers.

2. Compute $n = pq$ and $z = (p-1)(q-1)$.

3. Choose a number, e, less than n, that has no common factors (other than 1) with z. (In this case, e and z are said to be relatively prime.) The letter e is used since this value will be used in encryption.

4. Find a number, d, such that ed - 1 is exactly divisible (that is, with no remainder) by z. The letter d is used because this value will be used in decryption. Put another way, given e, we choose d such that

$$ed \; mod \; z = 1$$

5. The public key that Bob makes available to the world, K_B^+ , is the pair of numbers (n, e); his private key, K_B^- , is the pair of numbers (n, d).

The encryption by Alice and the decryption by Bob are done as follows:

- Suppose Alice wants to send Bob a bit pattern represented by the integer number m (with m < n). To encode, Alice performs the exponentiation m e , and then computes the integer remainder when m e is divided by n. In other words, the encrypted value, c, of Alice's plaintext message, m, is

$$c = m^e \; mod \; n$$

  The bit pattern corresponding to this ciphertext c is sent to Bob.

- To decrypt the received ciphertext message, c, Bob computes

$$m = c^d \; mod \; n$$

Example 2.2.1. suppose Bob chooses p = 5 and q = 7.

Then n = 35 and z = 24. Bob chooses e = 5, since 5 and 24 have no common factors. Finally, Bob chooses d = 29, since 5 * 29 - 1 (that is, ed - 1) is exactly divisible by 24. Bob makes the two values, n = 35 and e = 5, public and keeps the value d = 29 secret. Observing

these two public values, suppose Alice now wants to send the letters l, o, v, and e to Bob. Interpreting each letter as a number between 1 and 26 (with a being 1, and z being 26), Alice and Bob perform the encryption and decryption respectively. Note that in this example, we consider each of the four letters as a distinct message.

Encryption Phase

Plaintext Letter	m: numeric representation	m^e	Ciphertext $c = m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Decryption Phase

Ciphertext c	c^d	$m = c^d \bmod n$	Plaintext Letter
17	481968572106750915091509141 1825223071697	12	l
15	127834039403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	10000000000000000000000000000000	5	e

2.2.2 How RSA Works!!

In order to understand why RSA works, again denote n = pq, where p and q are the large prime numbers used in the RSA algorithm. Recall that, under RSA encryption, a message (uniquely represented by an integer), m, is exponentiated to the power e using modulo-n arithmetic, that is,

$$c = m^e \ mod \ n$$

Decryption is performed by raising this value to the power d, again using modulo-n arithmetic. The result of an encryption step followed by a decryption step is thus $(m^e \ mod \ n)^d \ mod \ n$. Let's now see what we can say about this quantity. As mentioned earlier, one important property of modulo arithmetic is $(a \ mod \ n)^d \ mod \ n = a^d \ mod \ n$ for any values a, n, and d. Thus, using $a = m^e$ in this property, we have

$$\mathbf{(m^e \ mod \ n)^d \ mod \ n = m^{ed} \ mod \ n}$$

It therefore remains to show that m ed mod n = m. Although we're trying to remove some of the magic about why RSA works, to establish this, we'll need to use a rather magical result from number theory here. Specifically, we'll need the result that says if p and q are prime, n = pq, and z = (p - 1)(q - 1), then x^y mod n is the same as $x^{(y \ mod \ z)} \ mod \ n$. Applying this result with x = m and y = ed we have

$$m^{ed} \ mod \ n = m^{(ed \ mod \ z)} \ mod \ n$$

But remember that we have chosen e and d such that ed mod z = 1. This gives us

$$m^{ed} \ mod \ n = m^1 \ mod \ n = \ m$$

which is exactly the result we are looking for! By first exponentiating to the power of e (that is, encrypting) and then exponentiating to the power of d (that is, decrypting), we obtain the original value, m. Even more wonderful is the fact that if we first exponentiate to the power of d and then exponentiate to the power of e—that is, we reverse the order of encryption and decryption, performing the decryption operation first and then applying the encryption operation—we also obtain

the original value, m. This wonderful result follows immediately from the modular arithmetic:

$$(m^d \ mod \ n)^e \ mod \ n = m^{de} \ mod \ n = m^{ed} \ mod \ n = (m^e \ mod \ n)^d \ mod \ n$$

The security of RSA relies on the fact that there are no known algorithms for quickly factoring a number, in this case the public value n, into the primes p and q. If one knew p and q, then given the public value e, one could easily compute the secret key, d. On the other hand, it is not known whether or not there exist fast algorithms for factoring a number, and in this sense, the security of RSA is not guaranteed. With recent advances in quantum computing, and published fast factoring algorithms for quantum computers, there are concerns that RSA may not be secure forever. But the practical realization of these algorithms still appears to be far in the future.