Chapter 1

Historical Introduction

Cryptology, the process of concealing messages, has been used for the last 4,000 years. It started at least as long ago as the Egyptians, and continues today and into the foreseeable future. The term **cryptology** is from the Greek $\kappa\rho\upsilon\pi\tau\omega$ or **kryptós**, meaning secret or hidden, and $\lambda o\gamma o\zeta$ or **logós**, meaning science. The term cryptology has come to encompass **encryption** (**cryptography**, which conceals a message) and **decryption** (revelation by **cryptanalysis**).

In this chapter we give a quick introduction to the main terms and goals of cryptology. Our intent here is not to delve deeply into the mathematics; we'll do that in later chapters. Instead, the purpose here is to give a broad overview using historical examples to motivate the issues and themes. Thus the definitions are less formal than later in the book. As this is a cryptography book, we of course highlight the contributions of the field and individuals in the stories below, though of course this cannot be the entire story. For example, even if you know the enemy's plan of attack, men and women must still meet them in the field of battle and must still fight gallantly. No history can be complete without recalling and appreciating the sacrifices many made.

Below we provide a brief introduction to the history of cryptography; there are many excellent sources (such as [**45**]) which the interested reader can consult for additional details. Later chapters will pick up some of these historical themes as they develop the mathematics of encryption and decryption. This chapter is independent of the rest of the book and is meant to be an entertaining introduction to the subject; the later chapters are mostly mathematical, with a few relevant stories.

For the most part, we only need some elementary number theory and high school algebra to *describe* the problems and techniques. This allows us to cast a beautiful and important theory in accessible terms. It's impossible to live in a technologically complex society without encountering such issues, which range from the obvious (such as military codes and deciphering terrorist intentions) to more subtle ones (such as protecting information for online purchases or scanning purchases at a store to get the correct price

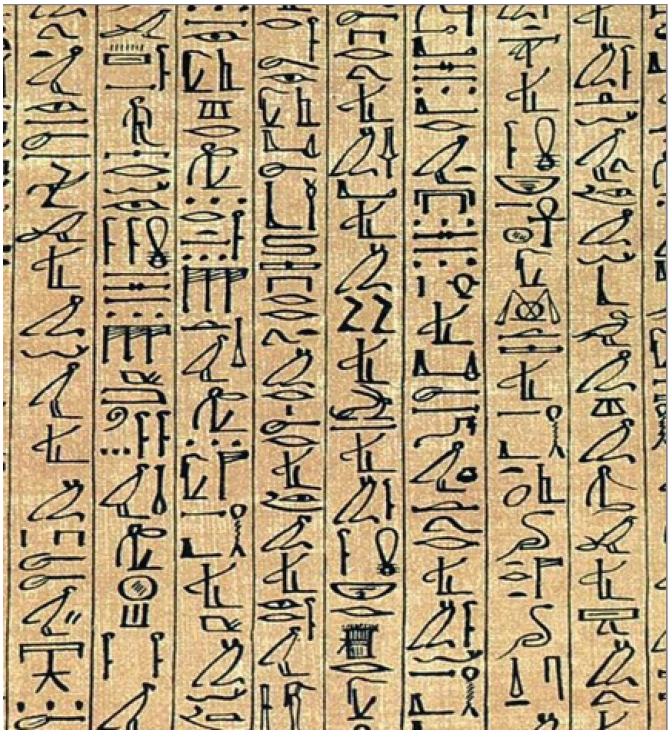

FIGURE 1.1. Hieroglyph on Papyrus of Ani. (Image from Wikipedia Commons.)

and update inventories in real time). After reading this book, you'll have a good idea of the origins of the subject and the problems and the applications. To describe modern attacks in detail is well beyond the scope of the book and requires advanced courses in everything from number theory to quantum mechanics. For further reading about these and related issues, see [**5, 6, 57**].

1.1. Ancient Times

The first practice of cryptography dates at least as far back as ancient Egypt, where scribes recorded various pieces of information as **hieroglyphs** on monuments and tombs to distinguish them from the commonly used characters of the time and give them more importance (see Figure 1.1). These hieroglyphics included symbols and pictures, and were translated by the hierarchy of the country to suit themselves. Thus, the hieroglyphs served the purpose of taking something in writing and masking the text in secrecy. s

The Egyptian hieroglyphs were initially done on stone as carvings and then later on papyrus. The Babylonians and others at about the same time used cuneiform tablets for their writing. One such tablet contained

the secret formula for a glaze for pottery, where the figures defining the ingredients were purposefully jumbled so people couldn't steal the secret recipe for the pottery glaze. This is the oldest known surviving example of encryption.

As important as pottery is to some, when cryptography is mentioned people think of spies and military secrets, not pottery glazes. The first documented use of secret writing by spies occurred in India around 500 BCE. The Indians used such techniques as interchanging vowels and consonants, reversing letters and aligning them with one another, and writings placed at odd angles. Women were expected to understand concealed writings as an important skill included in the *Kama Sutra*.

The Old Testament of the Bible includes an account of Daniel. He was a captive of Babylon's King Nebuchadnezzar around 600 BCE and had won promotion with successfully interpreting one of the king's dreams. He saw a message "Mene, Mene, Tekel, Parsin" written on a wall (Daniel 5:5–28) and interpreted it as *Mene* meaning "God Hath numbered thy kingdom and finished it"; *Tekel* as "Thou art weighed in the balances and art found wanting"; and *Parsin* as "Thy kingdom is divided and given to the Medes and Persians". The king was killed that very night and Babylon fell to the Persians. Other passages of the Old Testament allude to passwords required for entry into various places. Very few knew the passwords, or keys as they were often called.

As time progressed and conflict became more prevalent and important to the spread of boundaries, the need for concealed messages grew. This was also at a time when written records began to be collected. Both the Greeks and the Persians used simple encryption techniques to convey battle plans to their troops in the fifth century BCE. For example, one technique was to wrap a missive written on parchment around rods of specific sizes and with writing down the length of the rod. When unwrapped the letters were not in the right order, but wound around the right size rod they were. Another example is the Greek use of wooden tablets covered with wax to make them appear blank (a steganographic technique discussed in Chapter 10), which were then decrypted by melting the wax to expose the written letters.

Various transmission systems were developed to send messages in the period between 400 and 300 BCE, including the use of fire signals for navigation around enemy lines. Polybius, the historian and cryptographer, advanced signaling and cipher-making based on an idea of the philosopher Democritus. He used various torch signals to represent the letters of the Greek alphabet, and he created a true alphabet-based system based on a 5×5 grid, called the Polybius checkerboard. This is the first known system to transform numbers to an alphabet, which was easy to use. Table 1.1 shows a Polybius checkerboard (note that `i` and `j` are indistinguishable):

Each letter is coded by its row and column in that order; for example `s` is coded as 43. The word "spy" would be coded by 43, 35, 54, while "Abe is a spy" is 11, 12, 15, 24, 43, 11, 43, 35, 54. It's easy to decode: all we have to

TABLE 1.1. The Polybius checkerboard.

	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	x	y	z

do is look in the appropriate table entry to get the letter (remembering, of course, that 24 can be either an `i` or a `j`). For example, 22, 42, 15, 15, 43, 25, 11, 42, 15, 13, 34, 32, 24, 33, 22 decodes to either "Greeks are coming" or "Greeks are comjng"; it's clear from context that the first phrase is what's meant.

A **cipher** is a method of concealment in which the primary unit is a letter. Letters in a message are replaced by other letters, numbers, or symbols, or they are moved around to hide the order of the letters. The word **cipher** is derived from the Arabic **sifr**, meaning nothing, and it dates back to the seventh century BCE. We also use the word code, often interchangeably with cipher, though there are differences. A **code**, from the Latin **codex**, is a method of concealment that uses words, numbers, or syllables to replace original words or phases. Codes were not used until much later. As the Arabic culture spread throughout much of the western world during this time, mathematics flourished and so too did secret writing and decryption. This is when frequency analysis was first used to break ciphers (messages). **Frequency analysis** uses the frequency of letters in an alphabet as a way of guessing what the cipher is. For example, `e` and `t` are the two most commonly used letters in English, whereas `a` and `k` are the two most commonly used letters in Arabic. Thus, "native language" makes a difference. Chapters 4 and 5 include many examples of how frequency analysis can decrypt messages.

Abu Yusef Ya'qab ibn 'Ishaq as-Sabbah al-Kindi (Alkindus to contemporary Europeans) was a Muslim mathematician, who lived in what is now modern day Iraq between 801 and 873 AD. He was a prolific philosopher and mathematician and was known by his contemporaries as "the Second Teacher", the first one being Aristotle [**55**]. An early introduction to work at the House of Wisdom, the intellectual hub of the Golden Age of Islam, brought him into contact with thousands of historical documents that were to be translated into Arabic, setting him on a path of scientific inquiry few were exposed to in that time [**46**].

Al-Kindi was the first known mathematician to develop and utilize the **frequency attack**, a way of decrypting messages based on the relative rarity of letters in a given language. The total of his work in this field was published in his work *On Deciphering Cryptographic Messages* in 750 AD,

one of over 290 texts published in his lifetime [**50**]. This book forms the first mention of cryptology in an empirical sense, predating all other known references by 300 years [**28**]. The focus of this work was the application of probability theory (predating Fermat and Pascal by nearly 800 years!) to letters, and is now called **frequency analysis** [**41**].

The roots of al-Kindi's insight into frequency analysis began while he was studying the Koran. Theologians at the time had been trying to piece together the exact order in which the Koran was assembled by counting the number of certain words in each *sura*. After continual examination it became clear that a few words appeared much more often in comparison to the rest and, after even closer study in phonetics, it became more evident that letters themselves appeared at set frequencies also. In his treatise on cryptanalysis, al-Kindi wrote in [**50**]:

> One way to solve an encrypted message, if we know its language, is to find a different plaintext of the same language long enough to fill one sheet or so, and then we count the occurrences of each letter. We call the most frequently occurring letter the "first", the next most occurring letter the "second", the following most occurring letter the "third", and so on, until we account for all the different letters in the plaintext sample. Then we look at the cipher text we want to solve and we also classify its symbols. We find the most occurring symbol and change it to the form of the "first" letter of the plaintext sample, the next most common symbol is changed to the form of the ''second" letter, and the following most common symbol is changed to the form of the "third" letter, and so on, until we account for all symbols of the cryptogram we want to solve.

Interest in and support for cryptology faded away after the fall of the Roman Empire and during the Dark Ages. Suspicion of anything intellectual caused suffering and violence, and intellectualism was often labeled as mysticism or magic. The fourteenth century revival of intellectual interests became the Renaissance, or rebirth, and allowed for the opening and use of the old libraries, which provided access to documents containing the ancient ciphers and their solutions and other secret writings. Secret writing was at first banned in many places, but then restored and supported. **Nomenclators** (from the Latin *nomen* for name and *calator* for caller) were used until the nineteenth century for concealment. These were pairs of letters used to refer to names, words, syllables, and lists of cipher alphabets.

It's easy to create your own nomenclator for your own code. Write a list of the words you most frequently use in correspondence. Create codewords or symbols for each one and record them in a list. Then create an alphabet, which you will use for all the words that are not included in your list. Try

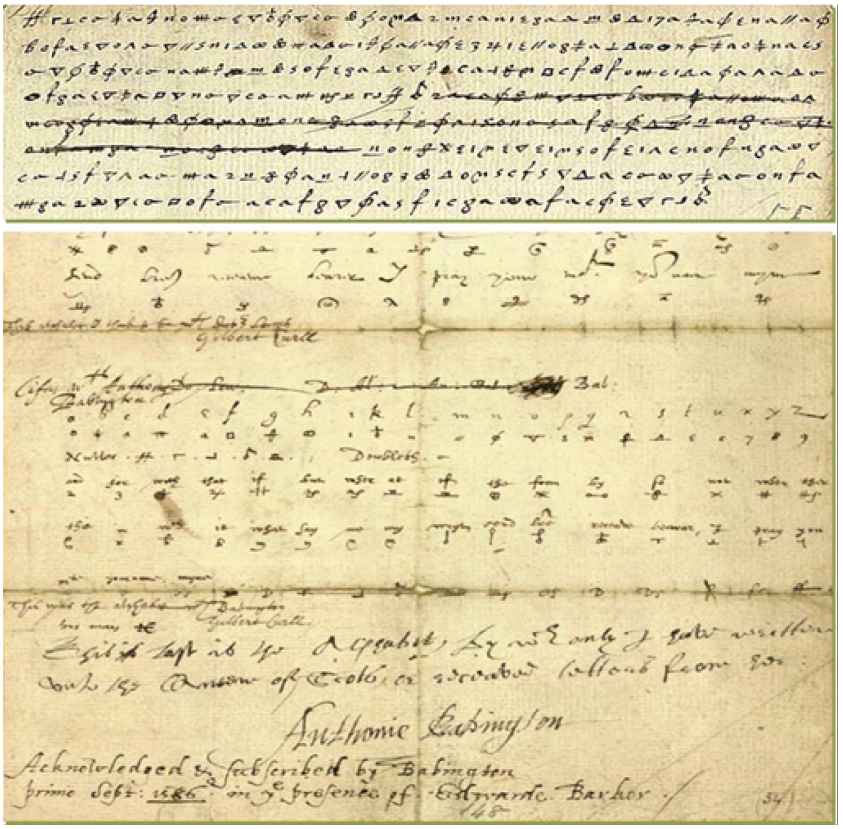

FIGURE 1.2. A forged nomenclator used in the Babington Plot in 1585. (Image from Wikipedia Commons.)

constructing a message to a friend by substituting the codeword for each word in the message that is on your list, and for those not in the list, use the alphabet you created. This should sound quite familiar to those who are used to texting. The difference here is that this uses your own codewords and alphabet, rather than commonly used phrases such as "lol" and "ttyl".

It wasn't until the seventeenth century that the French realized that listing the codewords in alphabetical order as well as the nomenclator alphabet in alphabetical order made the code more readily breakable. Figure 1.2 shows a portion of a fifteenth century nomenclator.

The Renaissance was a period of substantial advances in cryptography by such pioneer cryptographers, mostly mathematicians, as Leon Alberti, Johannes Trithemius, Giovanni Porta, Geirlamo Cardano, and Blaise de Vigenère. Cryptography moved from simple substitutions and the use of symbols to the use of keys (see Chapters 2 to 5) and decryption using probability.

Secrets were kept and divulged to serve many different purposes. Secret messages were passed in many ways, including being wrapped in leather and

then placed in a corked tube in the stoppers of beer barrels for Mary Stuart, Queen of Scots. Anthony Babington plotted to kill Queen Elizabeth I. He used beer barrels to conceal his message, telling Mary Stuart of the plot and his intent to place her, Mary, on the throne. He demanded a personal reply. In doing so, Mary implicated herself when the barrels were confiscated long enough to copy the message. They decrypted the message using letter frequency techniques (see Table 4.1 of §4.1). Mary Stuart was subsequently charged with treason and beheaded.

Double agents began to be widespread, especially during the American Revolution. Indeed, the infamous Benedict Arnold used a particular code called a **book code**. Because he was trusted, his correspondence was never checked and thus never tested. Not knowing whether that would continue to be true, he often used invisible ink to further hide his code.

Aaron Burr, who had at one time worked for Arnold, got caught up in his own scandal after Thomas Jefferson was elected president. Burr had been elected vice president, and he was ambitious and wanted to advance to the presidency. Alexander Hamilton learned of a plot to have New England and New York secede and publicly linked Burr to the plot. This led to the famous Hamilton–Burr duel, where Hamilton was killed. People turned against Burr as a result, and he, in turn, developed an elaborate scheme to get rid of Jefferson. The scheme included ciphers to link all of the many parts and people, some from Spain and England. Despite eventual evidence of deciphered messages, Burr was not convicted of treason.

Telegraphy and various ciphers played key roles during the Civil War. The **Stager cipher** was particularly amenable to telegraphy because it was a simple word transposition. The message was written in lines and transcribed using the columns that the lines formed. Secrecy was further secured by throwing in extraneous symbols and creating mazes through the columns. Consider the following simple example:

j	o	e	i	s
a	n	t	t	o
s	o	r	o	n
o	n	a	r	t

Most likely this would be read as “Joe is ant [antithetical] to soron [General Soron] on art”. But the intent is to read it as “Jason traitor”.

Women have always been directly involved in cryptography. An interesting example occurred during the Battle of Bull Run. A woman called Rebel Rose Greenhow sent messages to the Confederate defenders about Union troop movements and numbers. She used everything from pockets hidden in her clothing to coded designs embroidered into her dresses. She was so effective that the Federal authorities began counterespionage missions and tracked leaks to party and parlor gossip. Greenhow’s chief nemesis turned out to be Allan Pinkerton, the famous detective. He eventually trapped her and had her imprisoned; however, even from her cell she managed to create

new networks and methods of secret communication. In the end, the cryptographic efforts of the South were not as advanced and effective as those of the North. Despite the variety of codes and ciphers applied during the Civil War, none affected the outcome of the war as much as telegraphy did. Telegraphy and Morse code enabled Grant to use broad strategies on many fronts, contributing to Lee's surrender in 1865.

1.2. Cryptography During the Two World Wars

1.2.1. World War I

Cryptography has played an important role in the outcome of wars. The inadequacy of the cryptographic techniques at the beginning of World War I probably contributed to the loss of early potential Allied victories. Early attempts by the Russians, who far outnumbered the Germans, failed because the Russians sent messages in unprotected text that were picked up by German eavesdroppers, who then foiled the attacks.

The Allies were no better at intelligence gathering. Even though they intercepted a radio message from the German warship, *Goben*, in 1914 and deciphered the message, it was too late to prevent the shelling of Russian ports which ultimately caused Turkey to ally with the Germans. In general, decrypted messages were not generally trusted.

It was the hard work of the military and the intelligence gathering of the Allies that initially brought the plot of Zimmerman to the attention of the U.S. During the First World War, British naval intelligence began intercepting German radio messages. They amassed a group of scholars whose job was to decipher these German communications. With the help of the Allied forces and some good luck, they were able to come across German code books. Armed with their knowledge and hard work, the British cryptographers of what became known as **Room 40** decoded a message, called the **Zimmerman telegram**, from the German Foreign Minister Zimmerman.

It described German plans first sent to the German ambassador in the U.S. and then to the German ambassador in Mexico City. The message indicated that Germany was about to engage in submarine warfare against neutral shipping. Zimmerman, fearing that the U.S. would join England, proposed an alliance with Mexico. If the U.S. and Germany were to go to war with each other, Mexico would join forces with Germany, who would support Mexico regaining the land it lost to America in the Mexican-American War of 1846 to 1848. Room 40 analysts intercepted the telegram, deciphered it, and kept it secret for a while. It was then released to the Associated Press. The exposé shocked the U.S. into joining the war as an ally of the British.

1.2.2. Native Americans and Code Talkers in World War I and II

A group of **Choctaw Indians** were coincidentally assigned to the same battalion early in World War I, at a time when the Germans were wiretap-

ping and listening to conversations whenever and wherever possible. It thus became critically important for the Americans to send coded messages.

As the Choctaws were overheard in conversation in the command posts, officers thought about using the Choctaw native tongue to send coded messages. They tried successfully using the Choctaw language with two battalions and found no surprise attacks. The officials now knew that this linguistic system could work. For the most part these messages were sent as natural communications without additional coding. There were some issues, as some words were not in the Chocktaw vocabulary. This led to codewords being substituted, such as "big gun" for artillery, "stone" for grenade, and "little gun shoot fast" for machine gun. Telephone and radio were the most efficient means of communication, yet were highly susceptible to penetration; however, the use of the Choctaw language baffled the Germans, who were unable to decipher the language or the coded vocabulary. Some coded written messages in Choctaw were given to runners to protect their secrecy from the Germans, who often captured Americans to steal the valuable information.

The most famous group of **code talkers** were the **Navajos**, who were used in the Pacific during World War II (see Figure 1.3). It all began with an older gentleman, a WWI veteran himself, reading a paper on the massive death tolls encountered by the Americans and their efforts to create a safe encryption code. Philip Johnston was a missionary's son who grew up playing with Navajo children and learned their language as a boy. He was perhaps one of merely 30 non-Navajos who could understand their language. He knew that the U.S. had befuddled the Germans in World War I by using Choctaws to transmit messages in their own language on field phones. Thus, in combination with his war experience and with his intricate knowledge of the Navajo language, he realized that this could be the key to an unbreakable code. The Navajo marines and the few others who understood the language trained like all other marines; their desert and rough lifestyle actually benefited them during rigorous training. But in addition they were trained for radio communications and were tasked to create a unique code that would soon be used on the battlefield. Their language was very complex, which helped the security of their encrypted messages. For example, the Navajo language has at least ten different verbs for different kinds of carrying, depending on the shape and physical properties of the thing being carried. Also, depending on the tone or pitch of the speaker's voice, the same word could have a multitude of meanings. Even prefixes can be added to a verb, as many as ten different ones, to the point where one word in Navajo can take the place of a whole sentence in English.

Although their language seemed quite uninterpretable in its natural form, they took it a step further. To further encrypt the messages, they created the code that would be utilized on the front lines. The Navajo code initially consisted of a 234-word vocabulary, which over the course of WWII grew to some 450 words. Some military terms not found in the Navajo

FIGURE 1.3. Newton's Photograph from the Smithsonian Exhibit on American Indian Code Talkers. `http://www.sites.si.edu/images/exhibits/Code\%20Talkers/pages/privates_jpg.htm`

language were given specific code names, while others were spelled out. For example, "dive bomber" became "*gini*" (the Navajo word for chicken hawk). Even when they would spell words out, the word remained complex. Each English letter was assigned a corresponding English word to represent it and then that word was translated into Navajo. For example, `z` became "zinc" which then became "*besh-do-gliz*", and those letters that were frequently used were given three word variations so that a pattern, if decrypted by the enemy, could not easily be found. As an indication of its complexity, consider the code in a message sent in 1944: "*A-woh Tkin Ts-a Yeh-hes Wola-chee A-chen Al-tah-je-jay Khut*", which translated means, "Tooth Ice Needle Itch Ant Nose Attack Ready or now" corresponding to the decrypted message, TINIAN Attack Ready.

The Navajo code talkers could take a three-line English message and encode, transmit, and decode it in twenty seconds. A machine would take thirty minutes. Their unique skills were an important asset in the victories in WWII. Some Japanese thought it could be a tribal language, and there were cases where Navajo soldiers in POW camps were tortured and forced to listen to these encrypted messages. But all they could tell was that it was just incoherent jumbled words in Navajo. In order to decode the transmission, one had to be fluent in English, Navajo, and know the secret

code. It was never broken, and it wasn't until 1968 that the existence of these codes was released to the public, only after they had become obsolete.

1.2.3. World War II

Winston Churchill became Prime Minister of Great Britain seven months after the start of World War II. As a communications, intelligence, and security specialist in World War I, he was very aware of the importance of breaking German codes and ciphers. To respond to this need, he created a small group of decryption specialists, along with the Government Code and Cipher School at **Bletchley Park**, an estate 45 miles outside of London. Other linguists and mathematicians joined them in subsequent months to break the German encryptions, especially those generated by the **Enigma**. The Enigma, a rotor-based encryption device developed by the Germans, had the potential to create an immense number of electrically generated alphabets. Bletchley staff gave the code name **Ultra** to their deciphering efforts. Ultra was helped by French and Polish sources who had access to the Enigma's workings. The whole of Chapter 3 is devoted to the Enigma and the Ultra efforts.

The U.S. isolationist policies after World War I directed people away from the warning signs of trouble overseas, including some missed opportunities to detect the bombing of Pearl Harbor in December 1941. U.S. cryptographic units were blamed for not reading the signs. The Hypo Center in Hawaii did not have the decipherments of the "J" series of transposition ciphers used by Japan's consulate, despite the fact that one of the Japanese consulates was very near the U.S. naval base at Pearl Harbor. Had the Navy had access to the messages at the Hypo Center, history might have been different. In addition, the information filtering through the cryptoanalysts from the Japanese cipher machine **Purple** was not disseminated widely. They had broken the cipher, Red, from one of the Japanese cipher machines, but Purple was a complicated polyalphabetic machine that could encipher English letters and create substitutions numbering in the hundreds.

Dorothy Edgars, a former resident of Japan and an American linguist and Japanese specialist, noticed something significant in one of the decrypted messages put on her desk and mentioned it to her superior. He, however, was working on the decryption of messages from Purple and ignored her. She had actually found what is called the "lights message", a cable from the Japanese consul in Hawaii to Tokyo concerning an agent in Pearl Harbor, and the use of light signals on the beach sent to a Japanese submarine. After the shocking losses at Pearl Harbor, the U.S. leaders no longer put their faith in an honor code where ambassadors politely overlooked each other's communications. The U.S. went to war once again.

Naval battles became paramount, and cryptoanalysts played a key role in determining the locations of Tokyo's naval and air squadrons. The Navy relied heavily on Australian cryptoanalysts who knew the geography best.

General Douglas MacArthur commanded an Allied Intelligence Unit formed from Australian, British, Dutch, and U.S. units. They contributed to decisive Allied victories by successfully discovering Japan's critical military locations and their intended battles, such as Midway.

Traitors and counterespionage efforts continued to exist through the rest of the war. For example, the attaché Frank Fellers gave too-frequent and detailed reports about the British actions in North Africa, and German eavesdroppers snared various reports, reencrypted them and distributed them to Rommel. However, Fellers' activities were discovered, and Rommel was ultimately defeated after this source of information ceased.

Another aspect of cryptography is misdirection. The end of World War II was expedited through the transmission of codes and ciphers intended to be intercepted by German intelligence. Various tricks were employed to communicate false information and mislead them into believing something else was going on. They even had vessels sent to these bogus locations to give the appearance of an impending battle. We'll discuss some of these in greater detail in Chapter 3.

1.3. Postwar Cryptography, Computers, and Security

After World War II came the Cold War, which many feared could flare into an active war between the Soviets and the U.S. and her allies. It was a time of spies and counterspies, and people who played both sides of the fence. The damage to U.S. intelligence from activities of people like Andrew Lee and Christopher Boyce, the Falcon and the Snowman, was irreparable. They sold vital information to Soviet agents in California and Mexico, including top-secret cipher lists and satellite reconnaissance data in the 1970s. As a result, the Russians began protecting their launches and ballistic missile tests with better encrypted telemetry signals.

Another spy operated in the 1980s, John Walker. He was a Navy radio operator who used the KL-47, a mainstay of naval communications. It was an electronic rotor machine more advanced than the Enigma machine. He provided the Russians with wiring diagrams, and they were able to reconstruct the circuitry and determine with computer searches the millions of possible encrypted variations and read the encrypted messages.

Jewels was the codename for the carefully guarded cipher machines in Moscow used by the CIA and NSA cipher clerks. Many precautions were taken to protect the computer's CPU, and the cipher machines were state of the art with key numbers and magnetic strips that changed daily. Messages were double encrypted; however the Soviets managed to "clean" the power line to the machines so that electronic filters could be bypassed. The results of the subsequent leaks revealed many CIA agents who were then expelled, as well as revealing U.S. negotiating positions.

One of the more famous recent spies was identified in 1994 as Aldrich Ames, a CIA analyst, whose father Carleton had also been a CIA counterspy

in the 1950s. Aldridge Ames had been divulging secrets for at least ten years and had been in contact with many Russians as a CIA recruiter. He applied cryptographic techniques to conceal his schemes, some as simple as `B` meaning meet in Bogota, Columbia, while others involved a series of chalk-marked mailboxes with codenames like "north" and "smile", signaling brief commands like "travel on". At the time of this writing, he is serving a life sentence in prison for treason.

Cryptology continued to use codes and ciphers but was intensified, and it became more sophisticated with the improvements in computer technology. Horse Feistel of IBM in the 1970s developed a process of computer enhanced transposition of numbers using binary digits. It began as a demonstration cipher. Known as **Demon**, and then **Lucifer**, this DES cipher is a complicated encrypting procedure built upon groups of 64 plaintext bits, six of which were parity bits to guarantee accuracy. Simultaneously, Professor Martin Hellman and students Whitfield Diffie and Ralph Merkle collaborated to present the public key as a solution to the problem of distributing individual keys. This system had a primary basis of two keys. One was published and the other was kept private (see §8.5). For a while this system proved unbreakable, but in 1982 a trio of mathematicians from MIT broke it. They, Leonard Adleman, Ronald Rivest, and Adi Shamir, created another two-key procedure based on prime numbers. Their public key version is called **RSA**, and it is discussed in Chapter 8. RSA is slower to implement than DES because of its many computations, but is useful in networks where there are many communicants and the exchange of keys is a problem.

Today, matters of security are ever present as Social Security numbers, bank account numbers, employment data, and others are digitized on a daily basis. Some of the alphanumeric components used include door openers, passwords, health plan numbers, PIN numbers, and many more. Even though these are not intended as encryptions, they are nonetheless to be kept hidden for privacy and security reasons. The U.S. government became obsessed with a system developed in the 1990's called **Pretty Good Privacy (PGP)** for email, because they could not access emails when they thought they needed to. PGP has since been replaced by a system not nearly as good. A system called **key escrow** involved sending and receiving equipment that electronically chose algorithms from millions of available keys to encrypt conversations or data exchanges. The keys were to be held by two secure agencies of the federal government and required court-approved permission to access. It never gained public approval.

As computer technology improves, new codes and ciphers are developed for encryption, and attempts are made at decryption, often successfully. In some cases, old techniques, such as **steganography**, are made even better. Steganography is the technique of passing a message in a way that even the existence of the message is unknown. The term is derived from the Greek *steganos* (which means covered) and *graphein* (to write). In the past, it was often used interchangeably with cryptography, but by 1967 it became

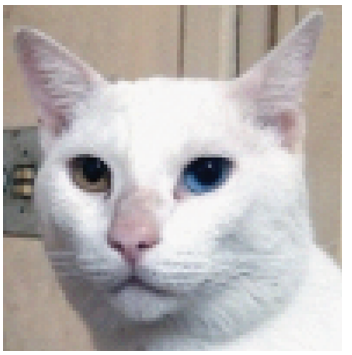

FIGURE 1.4. An embedded digital image that says "Boss says we should blow up the bridge".

used exclusively to describe processes that conceal the presence of a secret message, which may or may not be additionally protected by a cipher or code. The content of the message is not altered through the process of disguising it. The use of wax tablets discussed in §1.1 is an example of ancient steganography. Modern steganography, discussed in Chapter 10, not only conceals the content of messages, but hides them in plain sight in digital images, music, and other digitized media. The computer has provided a modern day invisible ink as these messages are not discernable by the naked eye or ear (see Figure 1.4).

Quantum computing has made **quantum cryptography** possible. Quantum cryptography uses quantum mechanical effects, in particular in quantum communication and computation, to perform encryption and decryption tasks. One of the earliest and best known uses of quantum cryptography is in the exchange of a key, called **quantum key distribution**. Earlier cryptology used mathematical theorems to protect the keys to messages from possible eavesdroppers, such as the RSA key encryption system discussed in Chapter 8. The advantage of quantum cryptography is that it allows fast completion of various tasks that are seemingly impractical using only classical methods, and it holds forth the possibility of algorithms to do the seemingly impossible, though so far such algorithms have not been found. Chapter 10 includes a longer discussion of quantum cryptography and the mathematics and physics behind it.

1.4. Summary

In this chapter we encountered many of the issues and key ideas of the subject (see [**12**] for an entertaining history of the subject). The first are various reasons requiring information protection. The case of Mary Stuart, Queen of Scots, and Anthony Babington show the grave consequences when ciphers are broken. While the effects here are confined to individuals, in

Chapter 3 we'll see similar issues on a larger scale when we explore Enigma and Ultra.

Another important takeaway is the need for speed and efficiency. In a battle situation, one does not have thirty minutes to leisurely communicate with headquarters. Decisions need to be made in real time. It's precisely for such reasons that the Navajo code talkers played such an important role, as they allowed U.S. units the ability to quickly communicate under fire. Of course, this code was designed to communicate very specific information. In modern applications we have a very rich set of information we want to encode and protect, and it becomes critically important to have efficient ways to both encrypt and decrypt.

Another theme, which will play a central role throughout much of this book, is replacing message text with numbers. We saw a simple recipe in the work of Polybius; we'll see more involved methods later. The monumental advances in the subject allow us to use advanced mathematical methods and results in cryptography.

We end with one last comment. Though there are many threads which we'll pursue later, an absolutely essential point comes from the Soviet efforts to read our ciphers. Even though the cipher machines in Moscow used double encryption, the Soviets were able to circumvent electronic filters by "cleaning" the power lines. This story serves as a powerful warning: in cryptography you have to defend against all possible attacks, and not just the expected ones. We'll see several schemes that appear safe and secure, only to see how a little more mathematics and a different method of attack are able to quickly break them.

1.5. Problems

EXERCISE 1.5.1. *Use the Polybius checkerboard to encode:*
(a) Men coming from the south.
(b) King has called a cease fire.

EXERCISE 1.5.2. *Use the Polybius checkerboard to encode:*
(a) Fire when ready.
(b) Luke, I am your father.

EXERCISE 1.5.3. *Use the Polybius checkerboard to decode:*
(a) $13, 54, 13, 32, 11, 14, 15, 44$.
(b) $33, 45, 35, 32, 54, 33, 41, 51, 44$.
(c) $23, 15, 32, 15, 34, 35, 21, 45, 43, 35, 54$.

EXERCISE 1.5.4. *Use the Polybius checkerboard to decode:*
(a) $43, 44, 15, 11, 31, 23, 34, 32, 15$.
(b) $35, 34, 31, 54, 12, 24, 45, 43$.

EXERCISE 1.5.5. *Use the Polybius checkerboard to decode*
$23, 22, 22, 22, 33, 25, 43$.

EXERCISE 1.5.6. *Come up with two messages that encrypt to the same text under the Polybius checkerboard but have different meanings; each message should make sense. Note there are not too many possibilities as almost all letters have a unique decryption.*

EXERCISE 1.5.7. *One difficulty in using the Polybius checkerboard is that it only has* 25 *squares, but there are* 26 *letters in the English alphabet. Show how we can overcome this by either increasing the size of the board or by considering a cube. What is the smallest cube that works?*

EXERCISE 1.5.8. *Create a nomenclator code book and alphabet, and use it to encrypt the message: "Meet me at our favorite restaurant at* 6 *PM."*

EXERCISE 1.5.9. *Using a Stager cipher, encode the message "Do you believe in miracles?"*

EXERCISE 1.5.10. *Using a Stager cipher, encode the message "It was early spring, warm and sultry glowed the afternoon." (*Note: *there is an interesting history to this quote, which can be uncovered by an internet search.)*

EXERCISE 1.5.11. *The following message was encrypted with a Stager cipher; what is it?*

d	*f*	*n*	*o*	*t*	*i*	*f*	*r*
o	*i*	*t*	*u*	*h*	*t*	*t*	*e*
n	*r*	*i*	*s*	*e*	*e*	*h*	*y*
o	*e*	*l*	*e*	*w*	*s*	*e*	*e*
t	*u*	*y*	*e*	*h*	*o*	*i*	*s*

EXERCISE 1.5.12. *The following message was encrypted with a Stager cipher; what is it?*

a	*s*	*l*	*l*	*u*	*p*	*o*
s	*i*	*i*	*l*	*t*	*t*	*r*
k	*x*	*o*	*a*	*a*	*f*	*m*
f	*m*	*n*	*r*	*c*	*o*	*o*
o	*i*	*d*	*s*	*c*	*u*	*r*
r	*l*	*o*	*b*	*e*	*r*	*e*

EXERCISE 1.5.13. *Deciphering the code in Exercises* 1.5.11 *and* 1.5.12 *is fairly easy* if *you know to read it in columns. We can increase the security by hiding the number of columns and writing it as* `d f n o t i f r o i t u h t t e n r i s e e h y o e l e w s e e t u y e h o i s`. *While this initially masks the number of columns, assuming we have at least two columns and at least two rows, show there are only six possibilities.*

EXERCISE 1.5.14. *Suppose your friend is considering encrypting a message to you through a Stager cipher. Having done Exercise* 1.5.13*, she knows that it would be a mistake to write the message in columns, as then it can be readily deciphered. She therefore decides to write it as a string of letters, and only the two of you will know the number of rows and columns. If there*

are r rows and c columns, this means she can send a message of rc letters. In terms of security and frustrating an attacker, which of the following is the best choice for rc and why: $1331, 1369, 1800$, *or* 10201?

EXERCISE 1.5.15. *Research and write a brief description about one of the following:*

- *The Black Chamber.*
- *The technological treason in the Falcon and the Snowman case.*
- *Cryptography during Prohibition and the role of Elizabeth Smith Friedman.*
- *Echelon.*
- *The Kryptos sculpture at the NSA.*

EXERCISE 1.5.16. *A major theme of this book is the need to do computations quickly. The Babylonians worked base* 60*; this meant they needed to know multiplication tables from* 0×0 *all the way to* 59×59*, far more than we learn today (since we work base* 10*, we only go up to* 9×9*).*

(a) Calculate how many multiplications Babylonians must memorize or write down.

(b) The number in part (a) can almost be cut in half, as $xy = yx$. Using this observation, how many multiplications must be memorized or written down?

(c) As it is painful and expensive to lug clay tablets around, there was a pressing need to trim these numbers as much as possible. The Babylonians made the remarkable observation that

$$xy = \frac{(x+y)^2 - x^2 - y^2}{2}.$$

Show this formula is true, and thus reduces multiplication to squaring, subtracting, and division by 2*.*

REMARK. The above formula shows that the Babylonians need only learn the squares and can deduce the remaining products by elementary operations. This is an early example of a "look-up table", where some calculations are done and stored, then used to deduce the rest. This exercise shows that the *standard* way to do a problem is sometimes not the most practical.

Chapter 10

Modern Cryptography

In this chapter we look at encrypting and decrypting today. In particular we consider the traditional cryptographic tools in the context of available technology, as well as new tools. First, we revisit steganography, talked about briefly in Chapter 1, and consider how messages can be hidden in digital text, audio, and visual media. Then we consider the complicated field of quantum cryptography from an elementary perspective. It is hoped and feared that someday quantum cryptography will advance to the point that there are truly unbreakable encryption schemes. Lastly, we look at how cryptography, especially decryption, plays an important role in not only wars, but in securing the homeland against attacks such as those that occurred on September 11, 2001.

10.1. Steganography—Messages You Don't Know Exist

10.1.1. Introduction

Invisible ink dates back to Roman times and was used extensively and successfully as recently as during World War II. German spies used invisible ink to print very small dots on letters, blocks of text or images scaled down to the size of a regular dot, now called microdots. Even as late as 2008, al Qaeda used the invisible ink form of steganography. A British man, Rangzieb Ahmed, was alleged to have a contact book with al Qaeda telephone numbers, written in invisible ink. He was subsequently convicted of terrorism and is currently in jail.

Steganography is the technique of passing a message in a way that even the existence of the message is unknown. The term steganography is derived from the Greek *steganos*, which means covered, and *graphein*, to write. In the past, it was often used interchangeably with cryptography, but since 1967 it has been used exclusively to describe processes that conceal the presence of a secret message, which may or may not be additionally protected by a cipher or code. The content of the message is not altered

through the process of disguising it. There are two types of steganography: technical or physical steganography, and linguistic.

Invisible ink is an example of technical steganography, as are hollowed out books, umbrella handles, and other things that occur frequently in spy novels. The use of technical steganography goes back to at least the fifth century B.C.E. The Greek historian Herodotus described the revolt against Persian rule that succeeded with the use of technical steganography. Two leaders of the revolt communicated secretly by shaving the head of a slave and tattooing a secret message on it. After the slave's hair grew back, he was sent to coconspirators who read the message by shaving his head. The Greeks were able to overthrow the Persians using the information in the concealed message.

Fifty years earlier, a second steganographic method was used by the Greeks against the Persians to turn back an invasion of Xerxes and his men. Demartus used a makeshift device created by scraping wax from two wooden tablets to alert the Spartans. He inscribed what he knew of the intentions of the Persians on the tablets and then replaced the wax covering. The seemingly plain tablets were passed to the Spartans untouched, who in turn scraped off the wax to read the message.

Why is encryption insufficient and steganography needed? The obvious answer is to hide the mere existence of files. Often no one can even prove such files exist, whereas courts can demand access to encrypted files, or a person can be tortured to give the password to encrypted data. With steganography we have the possibility that people are unaware of the existence of the message.

10.1.2. The Processes of Steganography

Let's start with some terminology. Steganography has various component parts.

- The **carrier** or **cover image** is the original file or image in which the secret message is to be embedded: text, image, audio, or video.
- The **payload** is the data (the image or message) to be hidden.
- The **package** or **stego-object** is the information embedded in the carrier.
- The **key** unlocks the message so it can be viewed by the receiver.

We depict the process in Figure 10.1. In functional notation, we can describe a **stego-function** as

$$f(C, P, K) \;=\; C + P + K \;=\; S,$$

where C is the carrier, P is the secret image, K is the key, and S is the result called the stego-object.

The **encoding density of a stego-image** is the percentage of bits in the payload to bits in the carrier. In other words the lower the density, the more hidden the file.

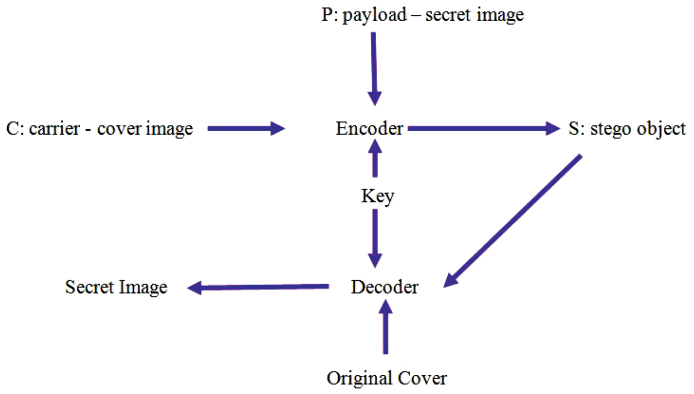

FIGURE 10.1. Schematic of steganography process.

Activity: Equipment required: deck of cards, fine point pen.

- Step 1: Choose two students.
- Step 2: These two students take the deck of cards and decide on an order to the cards.
- Step 3: Off to the side of the room they arrange the pack in that order.
- Step 4: They then write a message (to do something) on the side of the deck with a fine point pen, making sure no one sees them do it.
- Step 5: They shuffle the deck and pass the deck to another group of students.
- Step 6: Have the group do what the message indicated.

Before reading on, identify the carrier, payload, key, and stego-object in the activity.

In this activity, the **carrier** is the deck of cards, the **payload** is the message, the **stego-object** is the deck of cards with message on it, and the **key** is knowledge of the "right" order of the cards. Note that the encoding density of the stego-object is quite high, since the only thing added was the writing.

What things can be used today as carriers? Audio, video, still imagery, plain text, microdots, covert channels using Internet Control Message Protocol can all be used as carriers. Let's look at examples for each possible carrier. First we consider examples using plaintext carriers.

Using **plaintext**, one option is for the sender to send a series of numbers along with the message. There is prior agreement (which becomes the key) that the numbers correspond to the letter in the word in that place. A 0 corresponds to skipping a word. For example, consider the message "The men protest including raw meat. Some protest at least weekly. Beef

undercooked is leading soldiers' illness complaints to grow", with number sequence 3111121415211114701 which is delivered separately. The receiver takes the message and highlights the third letter of the first word, the first letter of the second word, the first letter of the third word and so on as follows:

> The **m**en **p**rotest **i**ncluding **r**aw m**e**at. **S**ome pro**t**est **a**t leas**t** w**e**ekly. **B**eef **u**ndercooked **i**s **l**eading sol**d**iers' **i**llness complai**n**ts to **g**row.

Using only the highlighted letters, the message becomes "EmpireState-Building", indicating a potential terrorist target. Note that the 0 indicates that the word "to" is not considered.

Null ciphers (unencrypted messages) don't require a key. The letter in each word is uniform and is known ahead of time, for example the second letter of each word. The message is innocent sounding to anyone but the recipient. Null ciphers were used extensively during WWII, but have not been used much since.

Horizontal line (word) shift coding is a form of hiding in plain text, which uses horizontal spacing between each line (word). The lines (words) are shifted slightly left or right, the decoding is given in a codebook. An extreme example is shown in the following using the sentences of a paragraph:

> Some kids used vinegar or milk to write a message to friends
> that others couldn't read; an iron would reveal it. Some used
> invisible ink (such as milk, vinegar, juice or other liquids
> that are clear on paper, but darken when heated).

Assume our codebook is 1 left = tomorrow, 2 left = today, 3 left = next week, 1 right = bomb, 2 right = DC and 3 right = al Qaeda. The paragraph above is 3 right, 1 left, 1 right, 2 right, which yields: `al Qaeda tomorrow bomb DC`. Note that the last line is flush so encodes nothing.

Vertical line (word) shift coding uses vertical shifting of lines(words) so only part of the message is seriously read.

Feature coding uses extra white space characters, between consecutive words for example. The key is a lookup table available to each of the communicants, which matches the numbers of spaces to words or letters. Obvious short words are filled in by the receiver. For example, take the message:

```
My  wife    and   I are having troubles with our children.
```

The number of extra spaces are 2,4,3,1,1,1, and the key is 1 = tomorrow, 2 = al Qaeda, 3 = attack, and 4 = NYC. The decoding is "al Qaeda in NYC to attack tomorrow". Tomorrow is repeated in order to make the spacing less obvious, and "in" is filled in by the recipient.

10.2. Steganography in the Computer Age

Steganography processes in the computer age make use of the newest technologies to hide messages in plain sight. We saw some examples of tools that could be used with or without a computer. Now we look at examples that are particularly designed for computer tools. Indeed, the most used steganography today consists of digital images. These visual digital images come in many formats.

10.2.1. Visual Images

The most used are **still images**. These are messages hidden in digital imagery utilizing the limits of human vision. Human vision cannot detect differences in luminance (intensity) of color at the high frequency end of the color spectrum. A digital picture (or video) is stored as an image file using an array of colored dots, called pixels. An image file is merely a file containing a binary representation of the color or light intensity of each picture element (pixel) that makes up the image. Each pixel typically has three numbers associated with it, one for the red intensity, one for the green intensity, and one for the blue intensity. Images typically use 8-bit or 24-bit color. An 8-bit color scheme means that eight bits are used per pixel. These numbers or values range from 0 to 255 (for 256 or 2^8 values). Each number is stored as eight bits of 0's and 1's. An 8-bit bitmap representing the three colors, or intensities, at each pixel will have 2^8 different values at blue, 2^8 different values at red, and 2^8 different values at green. A difference of 1 in intensity is imperceptible. For example, if 11111111, which represents blue, is changed to 11111110, then this is undetectable to the human eye.

The 24-bits per pixel provide a much better set of colors. Using 24-bits per pixel for color means that changes are even more minimal and indiscernible to the human eye. Thus, in a standard 24-bit bitmap you have three color components per pixel: red, green, and blue. Each component has 2^8 or 256 values. When using a computer printer to construct the image, color intensity is given as a number value for each of red, green, and blue for each pixel, and then converted by the computer to 24-bit representations. The value designation is shown in Figure 10.2.

Using the color coding described in Figure 10.2, 191 converts to 10111111 for red's intensity, 29 is represented by 00011101 and 152 is represented in binary by 10011000, which produces a magenta color as shown for a color pixel.

Using 8-bit color a 640×480 pixel image using 256 colors requires a 307 KB file, whereas one twice as large using 24-bit color needs a 2.36 MB file. These are large files and need some form of compression to be workable.

The simplest way of hiding a message within an image file is to use what is called the **least significant bit insertion method**. The **least significant bit** (**LSB**) of a binary string is the last (furthest to the right) bit of the string, the one representing the unit's value. The least significant

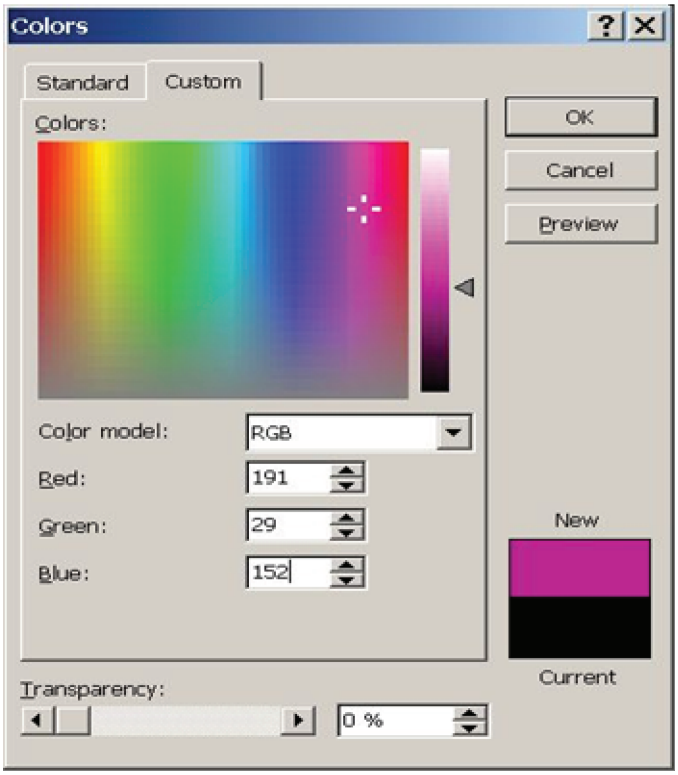

FIGURE 10.2. Color triples.

bit insertion method takes the binary representation of the hidden data and overwrites the least significant bit of the cover image with the LSB of the hidden image. Especially if we use 24-bit color, the change is imperceptible to the human eye.

Example 1: Three color pixels of the cover image are represented by the three 24-bit RGB words (9 bytes) below:

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

Now let's hide the payload or secret image 101101101 in the above cover image by overlaying them on the LSB of the cover image. For example the second of the nine is 00001101 which with the overlay of the second bit of 101101101 changes the last 1 to a 0. We get the following nine bits of data. The bits that have been changed are in bold.

10010101 00001100 11001001
10010111 00001110 11001011
10011111 00010000 11001011

We have hidden nine bits by changing only four of the LSB's.

FIGURE 10.3. Cover image (left). Extracted image (right).

Example 2: The following nine bytes of data are in the cover image:

00100111	11101001	11001000
00100111	11001000	11101001
11001000	00100111	11101001

Suppose the binary value for the letter `A` in the hidden message is (10000011). Inserting the binary value of `A` into the three pixels, starting from the left byte, results in the changed bytes shown in bold:

00100111	1110100**0**	11001000
0010011**0**	11001000	1110100**0**
1100100**1**	00100111	11101001

For example, if the cover image is the image of the trees, adding the hidden message "A" to every pixel yields a new image of a cat (see Figure 10.3). The cat is observable only to the receiver, who knows to drop all but the last two bits of each color component in each pixel and enhance the brightness 85 times.

There are numerous ways to conceal data in images. **DES** encryption algorithms are used by encrypting groups of 64 message bits using keys that are also 64-bits long. Every 8^{th} key bit is ignored in the algorithm making the key size actually 56 bits.

One can also use **Sudoku puzzles** to conceal data by using a key to hide the data within an image. There are as many keys as there are possible solutions to the puzzle. There are $6.71 \cdot 10^{21}$ possible solutions to the standard 9×9 Sudoku puzzle. This is equivalent to around 70 bits, making it much stronger than the DES method which uses a 56-bit key.

Masking and filtering are used to hide images within images. We can manipulate the luminance of particular areas of the image in order to

FIGURE 10.4. Effect of changing luminance.

encode data. The contribution of each pixel is varied to achieve desired effects. Consider for example the image in Figure 10.4. We display an adjusted luminance, where the letters AMS appear throughout.

Another WWII technique that is still used today to hide images is **microdots**. The microdot technique uses photographs the size of a printed period that contain the information of an 8×11 printed page. The message is hidden by its small size and needs not be encrypted. The recipient merely needs to know the message is coming and that microdots are used. The recipient has to have a way of enlarging the period to create the image.

10.2.2. Audio and Video Embedding

Audio signals are encoded using sampling rates (frames) and a frequency range. The human ear can only distinguish from 20Hz to 20KHz. However, when there are two sounds created simultaneously with one louder than the other, generally only the louder sound is heard. Most digital audio is created by sampling the signal and quantizing the sample with a 16-bit quantizer. The rightmost bit, or low order bit, of each sample can be changed from 0 to 1 or 1 to 0. This modification from one sample value to another is not perceptible by most people and the audio signal still sounds the same. This is called low bit coding.

Phase coding, on the other hand, relies on the relative insensitivity of the human auditory system to phase changes and substitutes the initial phase of an audio signal with a reference phase that represents the data. It is more

complex than low bit coding, but it is much more robust and less likely to distort the signal that is carrying the hidden data. Video encoding uses the human eye, which can only perceive approximately 24 frames per second. You can increase the frame rate and hide the payload in the superfluous frames. Since this may be detected visually, you can also encode each frame using each frame as a still image.

10.2.3. Digital watermarking

Watermarking has taken on a new importance in the digital era. Still images, video, music, text, and software are all easily copied and illegally distributed, causing the authors to lose out on considerable income in royalties. Digital watermarking is one other form of steganography used today to essentially copyright digital information. A message, which is just an identifier, is hidden in an image so that its source can be tracked or verified. Digital watermarking is used to protect against piracy or to identify an image's owner. Unfortunately, not all watermarking software is created equal. Some watermarks are easily removed or destroyed by manipulating various characteristics of the file, so companies are looking for more and more ways of securing the watermarks.

10.2.4. Steganography and Email, Chaffing and Winnowing, and Avatars

What was once thought of as junk email that got sent to your spam folder may be actually much more. This junk email may indeed be a message that is sent out, where only the recipient knows it is a message. The sender covers his tracks using techniques called chaffing and winnowing, rather than encryption.

In **chaffing and winnowing**, Alice, the sender, wants to send a message to Bob, the receiver. Alice lists the bits in her message and sends out each bit in a separate packet. Each packet contains the bit's serial number in the message, the bit itself (both unencrypted), and an identification code (**MAC**), whose secret key Alice shares with Bob. Charlie, the transmitter of Alice's packets to Bob, interleaves in a random order the packets with corresponding bogus packets (which are called **chaff**) with corresponding serial numbers, the bits inverted, and a random number in place of the MAC. Charlie does not need to know the key to transmit the message. Bob uses the MAC to find the authentic messages and drops the chaff. Bob's part of the process is called **winnowing**, thus the terminology chaffing and winnowing.

Now an eavesdropper located between Alice and Charlie can easily read Alice's message. But an eavesdropper between Charlie and Bob would have to tell which packets are bogus and which are real (i.e., to winnow, or "separate the wheat from the chaff"). That is infeasible if the MAC used is secure and Charlie does not leak any information on packet authenticity.

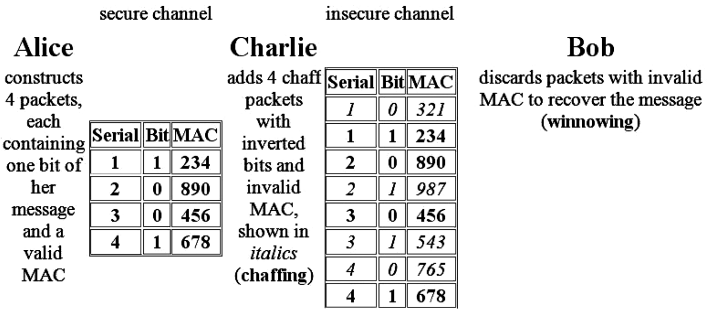

FIGURE 10.5. An example of chaffing and winnowing. (Image from Wikipedia Commons.)

An example is shown in Figure 10.5, where Alice wishes to send the message 1001 to Bob. Assume that all even MACs are valid and all odd ones are invalid.

When an adversary requires Alice to disclose her secret key, she can argue that she used the key simply for authentication. If the adversary cannot force Alice to disclose an authentication key, thereby allowing them to forge messages from Alice, then her message remains confidential. Charlie, on the other hand, does not even possess any secret keys that he could be forced to disclose.

An **avatar** in computing is any 2-dimensional or 3-dimensional graphical representation of a person, or object that represents the user. Avatars are used extensively in computer games, in internet forums, internet chats and blogs, in internet messaging, or even in artificial intelligence. These avatar graphical representations themselves can contain hidden messages. For example, the image in Figure 10.6 contains the message: "The invasion will take place December 12 at noon", using RedandBlueWing34 as a password. Be sure to do Exercise 10.6.9 to create your own avatar with hidden message.

The image shown in Chapter 1 of the cat was an avatar image also created using `mozaiq.org`.

10.3. Quantum Cryptography

> Nobody understands quantum theory.
> –Richard Feynman, Nobel Prize-winning physicist.

Quantum cryptography is based on quantum physics whose foundation is the **Heisenberg Uncertainty Principle**. In essence this principle says that it is impossible to know both an object's position and velocity at the same time. For cryptography, this uncertainty is used to secure the communication of messages. Once the secure key is transmitted using quantum

FIGURE 10.6. An avatar image constructed using `http://mozaiq.org/encrypt`.

cryptographic techniques, coding and encoding using the normal secret-key method can take place.

How is the secret key transmitted? Photons are used to transmit the key. You may have heard of **qubits**, the quantum computing analog of a bit. A qubit is a unit of quantum information. Each photon carries a qubit of information. A **bit** can take on the value of 0 or 1, but a qubit can take on the value of 0 or 1 ***or*** the superposition of the two. To create a photon, quantum cryptographers use light emitting diodes, (**LEDs**), as a source of unpolarized light to create one photon at a time. A string of photons are created through this process. Using polarization filters, we can force the photon to take one state or another, or polarize it (superposition of 0 and 1).

Spins and filters: Information is attached to the photon's spin as it emerges from a polarizing filter. Once a photon is polarized, it can't be accurately measured again, except by a filter like the one that initially produced its current spin. So if a photon with a vertical spin | is measured through a diagonal filter $\times$, either the photon won't pass through the filter (we use ( ) to denote this), or the filter will affect the photon's behavior, causing it to take a diagonal spin (either $\diagup$ or $\diagdown$). In this sense, the information on the photon's original polarization is lost, and so too is any information attached to the photon's spin.

The questions become the following: How does a photon become a key? How do you attach information to a photon's spin? Binary codes come into play. Each type of a photon's spin represents one piece of information, usually a 1 or a 0, for a binary code. As we saw before, a binary code uses strings of 1's and 0's to create a coherent message. So a binary code can be assigned to each photon—for example, a photon that has a vertical spin | can be assigned a 1.

Let's consider what is happening. Alice can send her photons through randomly chosen filters and record the polarization of each photon. She will then know what photon polarizations Bob should receive. When Alice sends Bob her photons using an LED, she'll randomly polarize them through either of two filters, so that each polarized photon has one of four possible states: |, —, ⁄ or \. As Bob receives these photons, he decides whether to measure each with either his + or × filter; he can't use both filters together. Keep in mind, Bob has no idea what filter to use for each photon, he's guessing for each one. After the entire transmission, Bob and Alice have a nonencrypted discussion about the transmission.

The reason this conversation can be public is because of the way it's carried out. Bob calls Alice and tells her which filter he used for each photon, and she tells him whether it was the correct or incorrect filter to use. Their conversation may sound a little like this: "Bob: Plus; Alice: Correct. Bob: Plus; Alice: Incorrect. Bob: ×; Alice: Correct." Since Bob isn't saying what his measurements are, only the type of filter he used, a third party listening in on their conversation can't determine the actual photon sequence.

For example, Alice sent one photon as a ⁄ and Bob says he used a + filter to measure it. Alice will say "incorrect" to Bob. But, if Bob says he used an × filter to measure that particular photon, Alice will say "correct." A person listening will only know that that particular photon could be either a ⁄ or a \, but not which one definitively. Bob will know that his measurements are correct, because a — photon traveling through a + filter will remain polarized as a — photon after it passes through the filter. After their odd conversation, Alice and Bob both throw out the results from Bob's incorrect guesses. This leaves Alice and Bob with identical strings of polarized protons. It might look like this: — \ | | \ — — | | — \ | ⁄ ⋯ and so on. To Alice and Bob, this is a meaningless string of photons. Now if the binary code is applied, the photons become a message. Bob and Alice can agree on binary assignments, say 1 for photons polarized as \ or — and 0 for photons polarized as ⁄ or |. This means that their string of photons is equivalent to 110001110001100, which can, in turn, be translated into English, Spanish, Navajo, prime numbers, or anything else Bob and Alice use as codes for the keys used in their encryption.

The goal of quantum cryptology is really to thwart attempts by a third party to eavesdrop on the encrypted message; we call the eavesdropper Eve. In modern cryptology, Eve can passively intercept Alice and Bob's encrypted

message; she can get her hands on the encrypted message and work to decode it without Bob and Alice knowing she has their message. Eve might accomplish this by wiretapping their phones, or reading emails thought to be secure, hacking into their computers, or countless other ways.

Now the Heisenberg Uncertainty Principle comes into play when Eve makes her own eavesdrop measurements; it makes quantum cryptology the first cryptographic scheme to safeguard against passive interception.

Let's go back to Alice and Bob and bring Eve into the picture. If Alice sends Bob a series of polarized photons, and Eve has set up a filter of her own to intercept the photons, Eve is in the same boat as Bob—neither of them has any idea what are the polarizations of the photons Alice sent. Like Bob, Eve can only guess which filter orientation (for example, an $\times$ filter or a $+$ filter) she should use to measure the photons. After Eve has measured the photons by randomly selecting filters to determine their spin, she will pass them down the line to Bob using her own LED with a filter set to the alignment she chose to measure the original photon. She does this to cover up her presence and the fact that she intercepted the photon message. By measuring the photons, Eve inevitably altered some of them.

Now suppose Alice sends to Bob one photon polarized to a — spin, and Eve intercepts this photon. Eve incorrectly chose to use an $\times$ filter to measure the photon. If Bob randomly (and correctly) chooses to use a $+$ filter to measure the original photon, he will find it's polarized in either a $\diagup$ or $\diagdown$, or it doesn't pass through, and he records ( ). Bob will believe he chose incorrectly until he has his conversation with Alice about the filter choice. After all of the photons are received by Bob, and he and Alice have their conversation about the filters used to determine the polarizations, discrepancies will emerge if Eve has intercepted the message. In the example of the — photon that Alice sent, Bob tells her that he used a $+$ filter. Alice tells him this is correct, but Bob knows that the photon he received didn't measure as — or |. Because of this discrepancy, Bob and Alice know that their photon has been measured by a third party, who inadvertently altered it. This is a powerful advantage of quantum cryptography over classical methods; a third party *must* leave evidence upon intercepting the signal.

Alice and Bob can further protect their transmission by discussing some of the exact correct results after they've discarded the incorrect measurements, by using a parity check on them. If the chosen examples of Bob's measurements are all correct, meaning the pairs of Alice's transmitted photons and Bob's received photons all match up, then their message is secure. Bob and Alice can then discard these discussed measurements and use the remaining secret measurements as their key. If discrepancies are found, they should occur in 50% of the parity checks. Since Eve will have altered about 25% of the photons through her measurements (half the time she guesses right and thus doesn't change anything, while for the other half of the time it passes through the incorrect filter and gets the correct orientation half of

the time), Bob and Alice can reduce the likelihood that Eve has the remaining correct information down to a one-in-a-million chance by conducting 20 parity checks.

Not all is perfect, however, in quantum cryptography. The distance that the key can be transmitted is a technical limitation, approximately 67 kilometers. The distance possible with quantum cryptology is short because of interference. A photon's spin can be changed when it bounces off other particles, and so when it's received, it may no longer be polarized the way it was originally intended to be. This means that a 1 may come through as a 0, which is really the probability consideration in quantum physics. As the distance a photon must travel to carry its binary message is increased, so, too, is the chance that it will meet other particles and be influenced by them. Single photon detection is hard, and the technology to process more than a few photons is expected to be developed in the future. Numerous companies are doing the research to develop technologies for quantum cryptography, largely because they see quantum cryptography as a technique to combat hacking.

10.4. Cryptography and Terrorists at Home and Abroad

Chapter 1 talked about the history of encryption through the end of World War II. In this section, towards the end of the book, we consider various applications and possible applications of cryptographic techniques since 2000. For security reasons, the use of cryptographic techniques is often only hinted at in the press or in federal reports, such as the 2006 *Federal Plan for Cyber Security and Information Assurance Research and Development.*

10.4.1. Steganography as a Terrorist Tool

Steganography as a terrorist tool appeared in the news as early as February 5, 2001, in *USA Today* in two different articles, "Terrorist instructions hidden online" and "Terror groups hide behind Web encryption", and then in July of the same year, in an article title, "Militants wire Web with links to jihad". The *USA Today* articles were written by veteran foreign correspondent Jack Kelley, who in 2004 was fired after allegations emerged that he had fabricated stories and sources, so the validity of the information in these articles has been called into question. Later, after 9/11, *The New York Times* published an article claiming that al Qaeda had used steganography to encode messages into images, and then transported these via e-mail to prepare and execute the September 11, 2001, terrorist attack. The *Federal Plan for Cyber Security and Information Assurance Research and Development* of April 2006, highlighted the U.S. fears:

- "... immediate concerns also include the use of cyberspace for covert communications, particularly by terrorists but also by foreign intelligence services; espionage against sensitive but poorly defended

data in government and industry systems; subversion by insiders, including vendors and contractors; criminal activity, primarily involving fraud and theft of financial or identity information, by hackers and organized crime groups. . . ." (pp. 9–10)

- "International interest in R&D for steganography technologies and their commercialization and application has exploded in recent years. These technologies pose a potential threat to national security. Because steganography secretly embeds additional, and nearly undetectable, information content in digital products, the potential for covert dissemination of malicious software, mobile code, or information is great." (pp. 41–42)
- "The threat posed by steganography has been documented in numerous intelligence reports." (p. 42)

It is likely that steganography techniques are in constant use by both sides in the war against terrorism today.

10.4.2. Visual Cryptography

Visual cryptography, a modern variant of steganography, is a method developed by Naor and Shamir in 1995 of encrypting information by breaking up an image into a given number of ciphertext sheets, called **transparencies**. The hidden message is revealed by stacking the transparencies and viewing the result. (Recall the activity with the stack of cards—this is a computer version of that one.) Decryption is performed directly by the human eye with no special calculations required, so even those with no cryptographic experience or means of cryptographic computation can make use of this system. In a basic visual cryptography scheme in which the original message is split into two shares, each share on its own provides no information on the content of the other, thus ensuring perfect secrecy. This guarantee extends to any situation in which an image is divided into n shares, of which k must be assembled to reveal the image; any combination of $k-1$ shares will contain no information on the appearance of the missing transparency. Further research has yielded more advanced methods of visual cryptography that permit the sender to specify qualified and forbidden subsets of n participants for image reconstruction, maximize the contrast of the image, and even conceal the existence of a hidden message by designing each transparency such that it exhibits some meaningful image rather than a random collection of pixels.

The most basic method of visual cryptography involves the division of a black and white image into two ciphertext transparencies. The original message is separated into its component black and white pixels, and each pixel is further divided into smaller sections. The system of pixel subdivision can be done in various ways, including a simple split into two rectangular halves, division into four blocks or replacement as a divided circle. Whatever the method, each combination that forms a full pixel always contains an equal

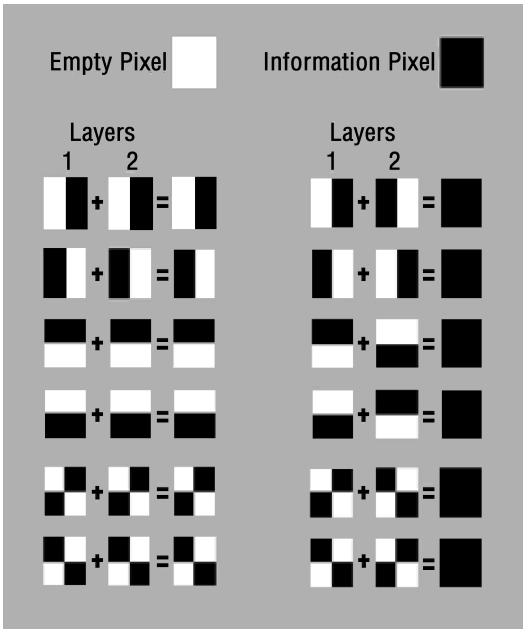

FIGURE 10.7. Blocks of pixels.

number of black and white (or, more precisely speaking, transparent) blocks. For example, a pixel divided into four square blocks can have six different states, each of which consists of two transparent blocks and two black blocks. With the image divided into pixels and a system of subdivision in place, the first transparency is made by assembling an image equal in size to that of the original message that consists of randomly selected subdivided pixels. The second layer is identical to the first except that those pixels that are black in the original image have an opposite state to that of the corresponding pixel in the first transparency, such that stacking the two layers yields an all-black pixel, and therefore the original message becomes visible. If the pixel states of layer one were selected at random, then both the empty and the information pixels in the second layer will have random states. Anyone who intercepts either of the images by itself will be unable to distinguish the character of the message by examination of its component pixels, so if the intended recipient of the message has been given a randomly generated transparency by the sender, the information layer can be transmitted publicly. This technique can be seen as a visual form of a one-time pad system; if the image on the first transparency contains truly random pixels, then the system guarantees near perfect security.

But while basic visual cryptography promises absolute secrecy, it inevitably comes with a few drawbacks. Like any one-time pad system, this method requires that all parties interested in communicating by these means must meet beforehand to exchange the randomly created image and that the

key is never reused, which makes it a poor choice for groups who want to communicate without meeting or are interested in extended exchanges of information. Additionally, encryption by visual cryptography necessarily reduces the contrast of the reconstructed image, since each empty pixel in the final image will be visually interpreted as a shade of grey. In the case described above, an empty pixel in the final image will be composed of two black blocks and two transparent blocks, so there will be a 50% loss of contrast.

An important tool in ferreting out the "bad guys" is commonly called **text extraction**. Millions of emails and other messages are transmitted daily between people, some of which contain important information about the future plans of terrorists, some hidden in text and some out in plain sight. One of the most important software tools to be developed in the early part of the twenty-first century can process a million pages of text and extract what is the "most important" information, which helps prevent attacks in the U.S. and abroad. These programs, such as **Jigsaw**, provide a visual analytical approach to text extraction.

10.5. Summary

This chapter focused on cryptography in the twenty-first century, primarily on various forms of digital steganography and the "hope of the future", quantum cryptography. Quantum cryptography enjoys many advantages over classical methods, one of the most important being that the sender and receiver can detect when a third party has viewed the message. Even if the attacker cannot decode it, it is often very important to know that someone is on the prowl. This is not the end of the interplay between quantum mechanics and cryptography. An active area of research is **quantum computing**. These quantum computers work on qubits, and have the potential of solving many problems significantly faster than classical computers. An important example is factorization. When implementing RSA, it was essential that factorization is hard; if it were easy, then the encryption would not be secure. There exist extremely fast factorization algorithms built for quantum computers; all that remains is to actually build these machines! This leads to the subject of the final chapter. As these quantum computers are still years away, we are forced to use classical computers to attack the factorization problem. In Chapter 11 we describe *some* of the many factorization algorithms.

10.6. Problems

• From §10.1: Steganography—Messages You Don't Know Exist

EXERCISE 10.6.1. *Construct an example of plaintext that decodes to "A battalion will invade the island on Sunday."*

EXERCISE 10.6.2. *Using a null cipher, encode a message that decodes to "A battalion will invade the island on Sunday."*

EXERCISE 10.6.3. *Give an example of how you could encode and decode the statement "A battalion will invade the island tomorrow" using horizontal line shifting, and another using vertical line shifts.*

EXERCISE 10.6.4. *Give an example of how you could encode and decode the statement "A battalion will invade the island tomorrow" using horizontal word shifting, and another using vertical word shifts.*

EXERCISE 10.6.5. *Create an example of how to use feature coding to encode and decode the statement "A battalion will invade the island tomorrow."*

• From §10.2: Steganography in the Computer Age

EXERCISE 10.6.6. *The following nine bytes of data are in the cover image:*

11010011 00001111 00110011
10101010 00110010 01010101
10010010 00010001 10010011

Hide the payload, which is given by 110101001*, in the cover image. How many bits did you have to change?*

EXERCISE 10.6.7. *The following nine bytes of data are in the cover image:*

11010011 00001111 00110011
10101010 00110010 01010101
10010010 00010001 10010011

Hide the payload, which is given by 110101000*, in the cover image. How many bits did you have to change?*

EXERCISE 10.6.8. *Instead of a* 3×3 *image, imagine now you have an* $n \times n$ *image with* n *a large number. You want to hide a payload in it. If the payload is* n^2 *digits, approximately how many of the entry's* n^2 *final digits in the image do you expect to have to change?*

EXERCISE 10.6.9. *Use* `http://mozaiq.org/encrypt` *to create your own message and an avatar image using your own password and share it. Note you can do it directly on your own computer.*

EXERCISE 10.6.10. *Write a short report on the software Jigsaw for visual analytics. What is the mathematics behind Jigsaw?*

• From §10.3: Quantum Cryptography

EXERCISE 10.6.11. *Research quantum computing. When was it first proposed, and by whom?*

EXERCISE 10.6.12. *Research the Heisenberg Uncertainty Principle, and write a short note on it.*

• From §10.4: Cryptography and Terrorists at Home and Abroad

EXERCISE 10.6.13. *Research the program Carnivore, and write a short note on it.*