MATH3302

Coding and Cryptography

Cryptography Part 1

2010

Contents

1 Overview of MATH3302

This course explores methods used to communicate *sensitive* and *important* information over an *insecure* and *unreliable* channel. To enable that we have a trade-off between secrecy and reliability versus simplicity and efficiency. The potential cost(s) of message interception or errors in the message determines the price we are willing to pay for message transmission.

Example 1.1 *In World War I, carrier pigeons were used to send messages from the field back to base. The message was tied around the pigeon's leg. Various things could go wrong, for instance the pigeon could be killed (not very reliable) or the pigeon could be intercepted (not very secure). Surpisingly, this method of message transmission was used by many groups of people for a long time. The following quotes are from "Instructions on the use of carrier pigeons in war", a booklet written by U.S. Army Chief Signal Officer in 1918.*

> *"... experience has proved that pigeons very quickly become accustomed to shellfire."*

When extra secrecy was needed the pigeon was induced to swallow the message. In that case

> *"... on arrival at its destination, extracting the message demands a little skill."*

For very important messages, multiple pigeons were sent.

Now we will look at the modern model for secure and reliable communication.

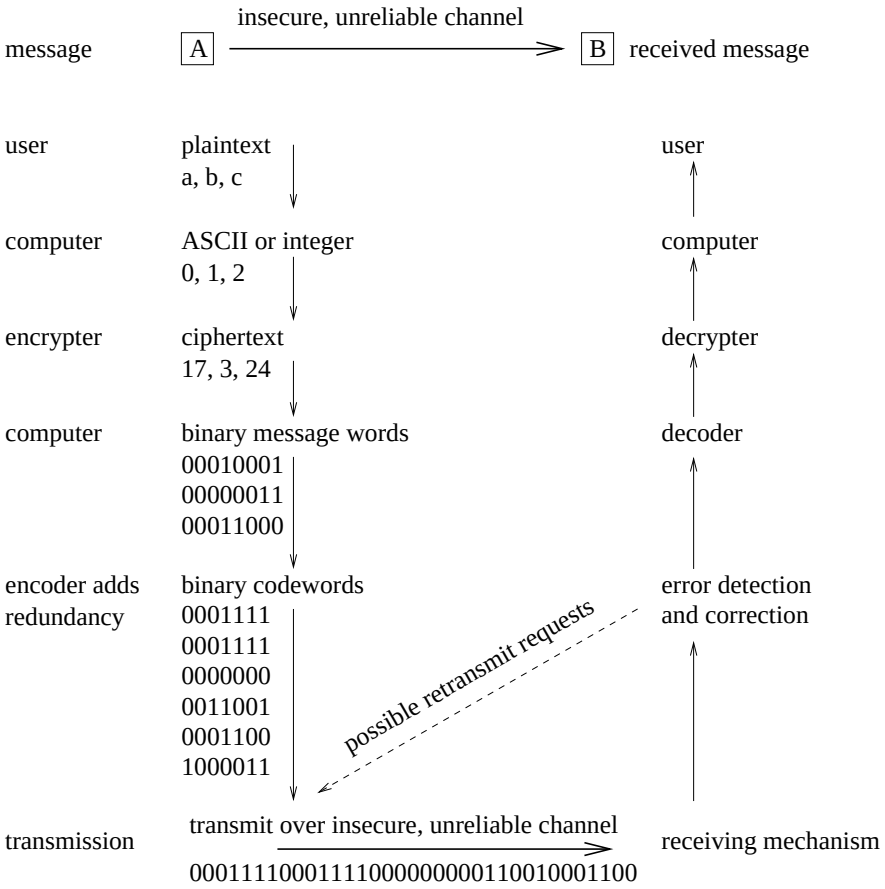


Figure 1: Electronic communication summary

In this course we look at two aspects of the communication process; these are related but distinct.

Coding Theory: concerned with efficient and reliable communication (not secrecy)

Cryptography: concerned with secure, secret communication (not reliability)

1.1 Coding Theory

In coding theory, redundant information is added to the message so that some errors can be detected and maybe even corrected. However, we can never guarantee that the correct message will be received. If too many errors occur in tranmission, then the process of error detection and correction will not give the correct message.

Example 1.2 *The 4-fold repetition code repeats each binary bit 4 times. Thus to send the message 0, you transmit the codeword 0000, and to send the message 1, you transmit the codeword 1111.*

If the received word is anything other than 0000 or 1111, then an error has been detected. Thus this code can detect up to 3 errors in any codeword.

If the received word is something like 0010 or 1000, then it is likely that the codeword was 0000. If the received word is something like 1110 or 1011, then it is likely that the codeword was 1111. So by picking the most likely codeword, we say that this code can correct 1 error.

If the received word is something like 1001 or 0011, then each codeword is equally likely, so the process of error detection and correction does not help.

This code is 3-error detecting and 1-error correcting, but in order to achieve this we needed to send 4 bits for each 1 bit of message, so it is not very efficient.

The process of adding redundant information gives improved reliability but reduces efficiency. The goal of coding theory is to determine how best to balance reliability and efficiency.

1.2 Cryptography

Two people want to communicate over an insecure channel in such a way that an eavesdropper cannot understand their message. There are two models that make this possible.

If it is practical for the two people to exchange a secret key in some secure way (for instance in person or over a secure channel) then they can use **private key cryptography**. The security of the system relies on the secrecy of the key. All classical cryptography systems are private key as well as many modern ones. Almost all the cryptography systems that we look at in this course are private key cryptosystems.

If it is impractical for the two people to exchage a secret key in some secure way, then they can use **public key cryptography**. The security of the system relies on the difficulty of solving a very difficult problem such as factoring a large number or solving a discrete logarithm problem. Several modern public key cryptography systems exist.

The fundamental properties of a good cryptography system include:

- ciphertext must be hard to decrypt without knowledge of the key;
- the system should be easy to implement (for both encryption and decryption) in both hardware and software;
- the system should be efficient;
- the encryption/decryption algorithms should be easy to understand and to verify correctness of;
- when you encrypt X and then decrypt the result, you should get X;
- the key should not be too large;
- the number of possible keys (the keyspace) should be big.

2 Introduction to cryptography

2.1 Basic concepts

cryptography the art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible, and then retransforming that message back to its original form. Crypto means secret and graphy means writing, so cryptography is secret writing.

plaintext the original intelligible message (we will write this in lowercase)

ciphertext the transformed message (we will write this in uppercase)

cipher an algorithm for transforming an intelligible message into one that is unintelligible by transposition and/or substitution methods

key some critical information used by the cipher, (usually) known only to the sender and receiver

encipher (encrypt) the process of converting plaintext to ciphertext using a cipher and a key

decipher (decrypt) the process of converting ciphertext back into plaintext using a cipher and a key

cryptanalysis the study of principles and methods of transforming an unintelligible message back into an intelligible message without knowledge of the key. Unauthorised recovery of the plaintext is sometimes called *code-breaking* or *cracking*.

cryptology a collective term for both cryptography and cryptanalysis

code an algorithm for transforming an intelligible message into an unintelligible one using a code-book

The fundamental objective of cryptography is to enable two people, usually referred to as Alice and Bob (A and B for short), to communicate over an insecure channel in such a way that an opponent, Oscar, cannot understand what is being said. This channel could be a telephone line or computer network, for example.

In the simplest model, Alice encrypts the plaintext message x, using a predetermined key, and sends the resulting ciphertext y over the (insecure) communication channel. Oscar, seeing the ciphertext in the channel by eavesdropping, (hopefully) cannot determine what the plaintext was. However, Bob, who knows the encryption key, can decrypt the ciphertext and reconstruct the plaintext.

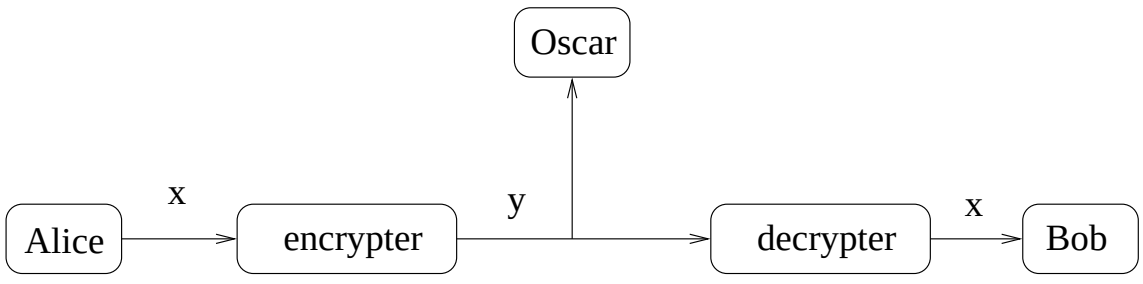


Figure 2: The communication process

Definition 2.3 *A cryptosystem is a five-tuple* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *in which the following conditions are satisfied:*

1. $\mathcal{P}$ *is a set of possible plaintext messages (called the* plaintext space*);*
2. $\mathcal{C}$ *is a set of possible ciphertexts or encrypted messages (called the* ciphertext space*);*
3. $\mathcal{K}$ *is a set of possible keys (called the* key space*);*
4. $\mathcal{E}$ *is a set of functions (encryption rules) indexed by* $\mathcal{K}$ *which map elements of* $\mathcal{P}$ *to elements of* $\mathcal{C}$*.*
5. $\mathcal{D}$ *is a set of functions (decryption rules) indexed by* $\mathcal{K}$ *which map elements of* $\mathcal{C}$ *to elements of* $\mathcal{P}$*.*
6. *For each* $K \in \mathcal{K}$*, there is a function* $e_K \in \mathcal{E}$ *and a corresponding function* $d_K \in \mathcal{D}$ *such that for any plaintext message* $x \in \mathcal{P}$*,*
$$d_K(e_K(x)) = x.$$

Notice that this means d_K *is the inverse of the function* e_K*.*

Definition 2.4 *A function* $f : X \to Y$ *is* **one-one** *or* **injective** *iff*

$$\forall a, b \in X,\ f(a) = f(b) \implies a = b.$$

Clearly, the encryption function e_K *must be one-one.*

2.2 History of cryptography

Cryptography has a fascinating history.

- Cryptography has been used in most cultures
- Records of some type of cryptography go back to 1900 BC
- Cryptography is listed as an 'essential art' in the Kama Sutra
- Cryptography has a major role in military applications

What follows is the *Timeline of Cryptography* from *CryptoBuddy.com*.

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

Upon reaching a level of literacy, cryptography appears in any culture in much the same manner as language and writing. The need for private communication among two or more people leads to forms of cryptography as the need for secrecy arises. Much of this is done by or on behalf of government and/or military, but in the long-term, the best available systems are often invented by civilians. As David Kahn wrote, *"It was the amateurs of cryptology who created the species. The professionals, who almost certainly surpassed them in cryptanalytic expertise, concentrated on down-to-earth problems of the systems that were then in use but are now outdated. The amateurs, unfettered to those realities, soared into the empyrean of theory."*

1900 BC	Egyptians used alternate hieroglyphs while inscribing tablets, the first documented use of written cryptography. The place was Menet Khufu on the Nile River. The writer did not use a completely different set of hieroglyphs, though. Instead, his was a system of partial substitution, with some unusual hieroglyphs (Egyptian writing) here and there.
1500 BC	In Mesopotamia, cryptography surpassed that of Egypt, approaching a very modern level. The first recorded use of cryptography was in 1500 BC with an encrypted formula for pottery glaze. The tablet containing the formula was only three by two inches and was found on the banks of the Tigris river. It used special signs which can have several different meanings.
590 BC	Hebrew scribes writing in the Book of Jeremiah used a reversed-alphabet simple substitution cipher known as Atbash. Jeremiah started dictating to Baruch in 605 BC, but the chapters containing these bits of cipher are attributed to a source labeled C, believed not to be Baruch. This could be an editor writing after the Babylonian exile in 587 BC, someone contemporaneous with Baruch or even Jeremiah himself. Atbash was one of a few Hebrew ciphers of the time.
487 BC	The Spartans of Greece created the first military form of cryptography. Their soldiers used a Skytale, a piece of wood with a strip of leather wrapped around it. They wrote on the leather, unwrapped it from the staff, and wore it as a belt. The recipient of the message would have to have an identical stick for the letters to line up when the leather was re-wrapped.
300 BC	Artha-sastra, a book attributed to Kautilya, was written in India. It recommended varieties of cryptanalysis, the process of breaking codes, to gain intelligence reports.
130 BC	In Uruk, which is now known as Iraq, it was popular for scribes to turn their names into numbers within the Colophon of their works. This was most probably done just to amuse the readers, and served no security-related purpose.
53 BC	Julius Caesar (100-44 BC) used a simple substitution with the normal alphabet, shifting the letters a fixed amount. This cipher was less strong than Atbash by a small amount, but in a day when few people read in the first place, it was good enough.
200 AD	The Leiden Papyrus, a work detailing how to make unusual potions, has crucial portions of the recipes enciphered. Examples of these magic recipes are some to supposedly make a man love a woman, or give a man an incurable skin disease. Incidentally, they do not work.
400 AD	The Kama Sutra of Vatsayana lists cryptography as the forty-fourth and forty fifth of sixty-four arts (yogas) men and women should know and practice. The date of this work is unclear, but is believed to be between the first and fourth centuries, AD. Vatsayana says his Kama Sutra is a compilation of much earlier works, making the dating of the cryptography references even more uncertain. Part I, Chapter III lists the sixty-four arts, and opens with: "Man should study the Kama Sutra and the arts and sciences subordinate thereto [...] Even young maids should study this Kama Sutra, along with its arts and sciences, before marriage, and after it they should continue to do so with the consent of their husbands." These arts are clearly not the province of a government or even of academics, but rather are practices of laymen. In this list of arts, the forty-fourth and forty-fifth read, "The art of understanding writing in cipher, and the writing of words in a peculiar way. The art of speaking by changing the forms of words. It is of various kinds. Some speak by changing the beginning and end of words, others by adding unnecessary letters between every syllable of a word, and so on."

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

725 AD	Abd al-Rahman al-Khalil ibn Ahmad ibn Amr ibn Tammam al Farahidi al-Zadi al Yahmadi, creator of the first Arab dictionary, wrote a book on how he solved a Byzantine cryptographic puzzle written in Greek. His method of attack started on an assumption the puzzle began with "In the name of god," and so he worked out the rest from that assumption. This method of attack is the same one employed in World War II to break German communications.
855 AD	Bakr Ahmad ben Ali ben Wahshiyya an-Nabati published several cipher alphabets traditionally used for magic. A few documents with ciphertext survive from the Ghaznavid government of conquered Persia, and one chronicler reports high officials were supplied with a personal cipher before setting out for new posts. The general lack of continuity of Islamic states, however, and the consequent failure to develop a permanent civil service and set up permanent embassies in other countries militated against cryptography's more widespread use.
1226 AD	A faint political cryptography appeared in the archives of Venice, where dots or crosses replaced the vowels in a few scattered words.
1250 AD	Roger Bacon, an English Monk, wrote his book, Secret Works of Art and the Nullity of Magic. In it, he describes several simple ciphers, such as using consonants only, or magic figures, and wrote, "A man is crazy who writes a secret in any other way than one which will conceal it from the vulgar."
1350 AD	Abd al-Rahman Ibn Khaldun wrote The Muqaddimah, a substantial survey of history which cites the use of "names of perfumes, fruits, birds, or flowers to indicate the letters, or [...] of forms different from the accepted forms of the letters" as a cipher among tax and army bureaus. He also includes a reference to cryptanalysis, noting, "Well-known writings on the subject are in the possession of the people."
1379 AD	Gabrieli di Lavinde at the request of Pope Clement VII, compiled a combination substitution alphabet and small code, the first example of the nomenclator. This class of code/cipher was to remain in general use among diplomats and some civilians for the next four-hundred fifty years in spite of the fact stronger ciphers being invented in the meantime, possibly because of its relative convenience.
1392 AD	The Equatorie of the Planetis, possibly written by Geoffrey Chaucer, contains passages in cipher. The cipher is a simple substitution with a cipher alphabet consisting of letters, digits and symbols.
1412 AD	Shihab al-Din abu 'l-Abbas Ahmad ben Ali ben Ahmad Abd Allah al-Qalqashandi wrote Subh al-a 'sha, a fourteen volume Arabic encyclopedia which included a section on cryptology. This information was attributed to Taj ad-Din Ali ibn ad-Duraihim ben Muhammad ath-Tha 'alibi al-Mausili who lived from 1312 to 1361, but whose writings on cryptology have been lost. The list of ciphers in this work included both substitution and transposition, and for the first time, a cipher with multiple substitutions for each plaintext letter. Also traced to Ibn al-Duraihim is an exposition on and worked example of cryptanalysis, including the use of tables of letter frequencies and sets of letters which can not occur together in one word.
1466 AD	Leon Battista Alberti (a friend of Leonardo Dato, a pontifical secretary who might have instructed Alberti in the state of the art in cryptology) invented and published the first polyalphabetic cipher, designing a cipher disk to simplify the process. This class of cipher was apparently not broken until the nineteenth century. Alberti also wrote extensively on the state of the art in ciphers, besides his own invention. Alberti also used his disk for enciphered code. These systems were much stronger than the nomenclator in use by the diplomats of the day and for centuries to come.
1473 AD	A manuscript by Arnaldus de Bruxella uses five lines of cipher to conceal the crucial part of the operation of making a philosopher's stone.
1518 AD	Johannes Trithemius wrote the first printed book on cryptology. He invented a steganographic cipher in which each letter was represented as a word taken from a succession of columns. The resulting series of words would be a legitimate prayer. He also described polyalphabetic ciphers in the now-standard form of rectangular substitution tables. He introduced the notion of changing alphabets with each letter.

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1553 AD	Giovan Batista Belaso introduced the notion of using a passphrase as the key for a repeated polyalphabetic cipher. This is the standard polyalphabetic cipher operation misnamed Vigenère by most writers to this day.
1563 AD	Giovanni Battista Porta wrote a text on ciphers, introducing the digraphic cipher. He classified ciphers as transposition, substitution, and symbol substitution (use of a strange alphabet). He suggested use of synonyms and misspellings to confuse the cryptanalyst. He apparently introduced the notion of a mixed alphabet in a polyalphabetic tableau.
1564 AD	Bellaso published an autokey cipher improving on the work of Cardano who appears to have invented the idea.
1585 AD	Blaise de Vigenère wrote a book on ciphers, including the first authentic plaintext and ciphertext autokey systems in which previous plaintext or ciphertext letters are used for the current letter's key. Both of these were forgotten and re-invented late in the nineteenth century. The autokey idea survives today in the DES, CBC, and CFB modes.
1623 AD	Sir Francis Bacon advanced a cipher by employing one of the first uses of steganography, or hiding the fact an encrypted message even exists. He hid his messages by slightly changing the typeface of a random text so each bit of the code was hidden within the random text's letters. He described a cipher now bearing his name, a biliteral cipher, known today as a 5-bit binary encoding. He advanced it as a steganographic device by using variation in typeface to carry each bit of the encoding.
1790 AD	Thomas Jefferson, possibly aided by Dr. Robert Patterson, a mathematician at the University of Pennsylvania, invented his wheel cipher. This was re-invented in several forms later, and used in WWII by the US Navy as the Strip Cipher, M-138-A.
1817 AD	Colonel Decius Wadsworth produced a geared cipher disk with a different number of letters in the plain and cipher alphabets, resulting in a progressive cipher in which alphabets are used irregularly, depending on the plaintext used.
1854 AD	Charles Babbage seems to have re-invented the wheel cipher.
1854 AD	Charles Wheatstone invented what has become known as the Playfair Cipher, having been publicized by his friend Lyon Playfair. This cipher uses a keyed array of letters to make a digraphic cipher which is easy to use in the field. He also re-invented the Wadsworth Device, and is known for that one.
1857 AD	Admiral Sir Francis Beaufort's cipher, a variant of what is called Vigenère, was published by his brother, after the admiral's death in the form of a four by five inch card.
1859 AD	Pliny Earle Chase published the first description of a fractionating (tomographic) cipher.
1861 AD	During the Civil War, possibly among other ciphers, the Union used substitution of select words followed by word columnar-transposition while the Confederacy used Vigenère, the solution of which had just been published by Kasiski.
1861 AD	Friedrich W. Kasiski published a book giving the first general solution of a polyalphabetic cipher with repeating passphrase, thus marking the end of several hundred years of strength for the polyalphabetic cipher.
1861 AD	The first US Patent on a cryptographic device was filed. About eighteen-hundred patents in the field have been issued since.
1891 AD	Major Etienne Bazeries invented his version of the wheel cipher, and demonstrated it for the French Army.
1895 AD	Radio was invented. The importance of this to cryptology is immense. During times of War, it allowed for enemy communications to be intercepted en mass. Thus, the profession of cryptanalysis, or the breaking of encrypted messages, was born.

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1901 AD	Major Etienne Bazeries published his version of the design of the wheel cipher after the French Army rejected it.
1913 AD	Captain Parket Hitt reinvented the wheel cipher, in strip form, leading to the M-138-A of WW II.
1914 AD	World War I began, marking a time of intense military use of cryptography. Britain began using messages run through cryptographic squares for their communications, and the French began using four-digit codes representing words.
1915 AD	Germany changed its cipher method to complicated substitution ciphers, using twenty-four possible encryption alphabets, or combinations of them. These ciphers became progressively more complicated.
1916 AD	Major Joseph O. Mauborgne put Hitt's strip cipher back in wheel form, strengthened the alphabet construction, and produced what led to the M-94 cipher device.
1916 AD	The Allies started using a code book in telephone communication. The Germans had been consistently intercepting wired telephone calls because only radio-based communications were previously encrypted. Thus, after a catastrophic loss of life that resulted from a message intercept, General Dubail of France requested these trench codes, which dictated that in normal telephone conversations, certain words be spelled out in code rather than spoken. The Germans did not start doing this in their communications for another year.
1917 AD	British cryptographers broke the Zimmerman Telegram, a secret German communication to Mexico in which the Germans offered Mexico United States territory in return for joining the German cause. When the American public got wind of this, their opinion became in favor of joining the war with the allies. In his book, David Kahn says, "Never before or since has so much turned upon the solution of a secret message."
1917 AD	William Frederick Friedman, later to be honored as the father of US cryptanalysis was employed as a civilian cryptanalyst at Riverbank Laboratories, and performed cryptanalysis for the US Government, which had no cryptanalytic expertise of its own. He went on to start a school for military cryptanalysts at Riverbank, later taking that work to Washington and leaving Riverbank.
1918 AD	The ADFGVX System was put into service by Germany near the end of WW I. This was a cipher which performed a substitution through a keyed array, fractionation, and then transposition of the letter fractions. It was broken by the French cryptanalyst, Lieutenant Georges Painvin.
1918 AD	The United States employed eight American Indians from the Choctaw tribe to relay important messages across insecure communication channels in their native tongue. Since Native American languages are extremely complex and difficult to learn, this allowed for simple and effective encryption.
1919 AD	Arvid Gerhard Damm applied for a patent in Sweden for a mechanical rotor cipher machine. This machine grew into a family of cipher machines under the direction of Boris Caesar Wilhelm Hagelin who took over the business, and was the only one of the commercial cryptographers of this period to make a thriving business. After the war, a Swedish law enabled the government to appropriate inventions it felt important to defense, causing Hagelin to move the company to Zug, Switzerland where it was incorporated as Crypto AG. The company is still in operation, although facing controversy for having allegedly weakened a cipher product for sale to Iran.
1919 AD	Gilbert S Vernam, an AT&T employee, invented a practical polyalphabetic cipher machine to make a non-repeating, virtually random sequence of characters, often called a one-time pad. (US Patent 1,310,719 of 22 July 1919) Using an encryption key the same length as the message, and never using that key again is the only proven method of securely communicating. It is impractical under most circumstances because all parties must have a long and identical key, presenting a logistical nightmare for everyday use. This machine was offered to the government for use in WW I, but was rejected. It was put on the commercial market in 1920.

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1919 AD	Hugo Alexander Koch filed a patent in the Netherlands on a rotor-based cipher machine. He assigned these patent rights in 1927 to Arthur Scherbius the inventor and marketer of the Enigma Machine since about 1923.
1921 AD	Edward Hugh Hebern incorporated Hebern Electric Code, a company making electro-mechanical cipher machines based on rotors which turn, odometer style, with each character enciphered.
1923 AD	Arthur Scherbius incorporated Chiffriermaschinen Aktiengesellschaft to make and sell the Enigma Machine he had invented that same year. He tried to sell it commercially, but had little luck. The German Government would eventually take it over, improve upon it, and use it to encrypt military communications in World War II.
1924 AD	Alexander von Kryha produced his coding machine which was used, even by the German Diplomatic Corps, into the 1950s. However, it was cryptographically weak, having a small period. A test cryptogram of eleven-hundred thirty-five characters was solved by the US cryptanalysts Friedman, Kullback, Rowlett, and Sinkov in two hours and forty-one minutes. Nevertheless, the machine continued to be sold and used, a triumph of salesmanship, and a lesson to consumers of cryptographic devices.
1927 AD	During the US Prohibition years 1927 to 1933 users of cryptography were not limited to legitimate bankers, lovers, experimenters, etc. There were also a handful of criminals. This greatest era of international smuggling created the greatest era of criminal cryptology. To this day, the FBI runs a cryptanalytic office to deal with criminal cryptography. Kahn noted in some of his 1967 writing, "A retired lieutenant commander of the Royal Navy devised the systems for Consolidated Exporters' Pacific operation, though its Gulf and Atlantic groups made up their own as needed. His name was unknown, but his cryptologic expertise was apparent. The smugglers' systems grew increasingly more complicated." "Some of these are of a complexity never even attempted by any government for its most secret communications," wrote Elizabeth Smith Friedman in a report in mid-1930. "At no time during the World War, when secret methods of communication reached their highest development, were there used such involved ramifications as are to be found in some of the correspondence of West Coast rum running vessels."
1927 AD	Hugo Alexander Koch assigned the patent rights on a rotor-based cipher machine to Arthur Scherbius who invented and had been marketing the Enigma Machine since about 1923.
1929 AD	Lester S. Hill published Cryptography in an Algebraic Alphabet in which a block of plaintext is enciphered by a matrix operation.
1930 AD	The American Sigaba Machine (M-134C) was invented, but the name of the inventor is in question. Kahn attributes it to William F. Friedman while Deavours attributes it to an idea of Frank Rowlett, one of Friedman's first hires. It improved on the rotor inventions of Hebern and Scherbius by using pseudo-random stepping of multiple rotors on each enciphering step rather than have uniform, odometer-like stepping of rotors as in Enigma. It also used fifteen rotors (ten for character transformation, five probably for controlling stepping) rather than the Enigma's three or four.
1930 AD	The British developed the Typex encryption machine, based on the commercial Enigma from the 1920s. This machine contained five rotors, each of which changed letters of the alphabet to other letters. After each character of the message being encrypted was typed, the rotors changed positions, creating an entirely new encryption scheme for the next letter. Reversing the process decrypted the message.
1933 AD	The Enigma Machine was not a commercial success, but it was taken over and improved upon to become the cryptographic workhorse of Nazi Germany. It was broken by the Polish mathematician, Marian Rejewski, based only on captured ciphertext and one list of three months worth of daily keys obtained through a spy. Continued breaks were based on developments during the war by Alan Turing, Gordon Welchman, and others at Bletchley Park in England.

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1937 AD	The Japanese Purple Machine was invented in response to revelations by Herbert O. Yardley. All machines prior to it used rotors to change the position of letters in the alphabet. The Purple Machine used telephone stepping relays instead of rotors, and thus had a totally different permutation at each step rather than the related permutations of one rotor in different positions. This was a totally new concept to cryptography, and thus standard cryptanalytic techniques were useless against it. US cryptographer William Friedman eventually broke the code generated by this machine.
1939 AD	The allies got their first look at a German Enigma machine after Polish intelligence stole it.
1940 AD	The Bombe, a machine to decode Enigma messages, was invented.
1942 AD	The United States began using Navajos in a manner similar to the use of the Choctaws in World War II, to speak important communications in their native language, so the enemy could not understand.
1944 AD	The Allies began using Sigaba to encrypt high level communications. Sigaba was a much-enhanced Enigma, and no one was able to decipher any communications encrypted with it. However, this extremely advanced machine can now allegedly be broken with modern supercomputers in approximately eight seconds.
1970 AD	Dr. Horst Feistel led a research project at the IBM Watson Research Lab in the 1960s which developed the Lucifer Cipher. This later inspired the US DES and other product ciphers, creating a family labeled Feistel Ciphers.
1974 AD	Publication of US FIPS 31 (June) Guidelines for Automatic Data Processing Physical Security and Risk Management
1975 AD	Publication of US FIPS 41 (May) Computer Security Guidelines for Implementing the Privacy Act of 1974 (Withdrawn November 1998)
1975 AD	Publication of US FIPS 65 (August) Guidelines for Automatic Data Processing Risk Analysis (Withdrawn August 1995)
1976 AD	Publication of US FIPS 39 (February) Glossary for Computer Systems Security (Withdrawn April 1993)
1976 AD	Public Key Cryptography is born. Whitfield Diffie and Martin Hellman published New Directions in Cryptography, introducing the idea of public key cryptography. They also put forth the idea of authentication by powers of a one way function, now used in the S/Key challenge/response utility. They closed their paper with the observation, "Skill in production cryptanalysis has always been heavily on the side of the professionals, but innovation, particularly in the design of new types of cryptographic systems, has come primarily from amateurs."
1976 AD	Publication of US FIPS 46A design by IBM, based on the Lucifer Cipher, and with changes (including both S-box improvements and reduction of key size) by the US NSA, was chosen to be the US Data Encryption Standard (DES). It has since found worldwide acceptance, largely because it has shown itself strong against twenty years of attacks. Even some who believe it is past its useful life use it as a component.
1977 AD	Publication of US FIPS 48 (April) Guidelines on Evaluation of Techniques for Automated Personal Identification

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1977 AD	Inspired by the Diffie-Hellman paper, and acting as complete novices in cryptography, Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman had been discussing how to make a practical public key system. One night in April, Ron Rivest was laid up with a massive headache, and the RSA algorithm came to him. He wrote it up for Shamir and Adleman, and sent it to them the next morning. It was a practical public key cipher for both confidentiality and digital signatures, based on the difficulty of factoring large numbers. They submitted this to Martin Gardner on April 4 for publication in Scientific American. It appeared in the September 1977 issue. The Scientific American article included an offer to send the full technical report to anyone submitting a self-addressed, stamped envelope. There were thousands of such requests, from all over the world. Someone at the NSA objected to the distribution of this report to foreign nationals, and for a while, mailings were suspended, but when the NSA failed to respond to inquiries asking for the legal basis of their request, mailings resumed. Adi Shamir believes this is the origin of the current policy, as of August 1995, allowing technical reports and/or papers to be freely distributed. The international journals, Cryptologia and The Journal of Cryptology were founded shortly after this attempt by the NSA to restrain publication. Contrary to rumor, the authors apparently had no knowledge of ITAR or patent secrecy orders. They published before applying for international patents not because they wanted to avoid such restraints on free expression, but rather because they were not thinking about patents for the algorithm. They just wanted to get the idea out.
1978 AD	The RSA algorithm was published in the Communications of the ACM.
1980 AD	Publication of US FIPS 73 (June) Guidelines for Security of Computer Applications
1980 AD	Publication of US FIPS 83 (September) Guideline on User Authentication Techniques for Computer Network Access Control
1980 AD	Publication of US FIPS 81 (December) DES Modes of Operation
1980 AD	The United States Patent Office had by then issued one-thousand seven-hundred sixty-nine patents primarily related to cryptography.
1981 AD	Publication of US FIPS 87 (March) Guidelines for ADP Contingency Planning
1981 AD	Publication of US FIPS 74 (April) Guidelines for Implementing and Using the NBS Data Encryption Standard
1981 AD	Publication of US FIPS 88 (August) Guideline on Integrity Assurance and Control in Database Administration (Withdrawn July 1997)
1983 AD	Publication of US FIPS 101 (June) Guidelines for Life Cycle Validation, Verification, and Testing of Computer Software (Withdrawn February 2000)
1983 AD	Publication of US FIPS 139 (August) Interoperability and Security Requirements for Use of the Data Encryption Standard in the Physical Layer of Data Communications (Withdrawn February 2000)
1983 AD	Publication of US FIPS 102 (September) Guidelines for Computer Security Certification and Accreditation
1983 AD	Publication of US FIPS 94 (September) Guideline on Electrical Power for ADP Installations (Withdrawn July 1997)
1984 AD	The Rot13 Cipher was introduced into USENET news software to permit the encryption of postings in order to prevent innocent eyes from being assaulted by objectionable text. This is the first known example of a cipher with a key everyone knows actually being effective.
1985 AD	Publication of US FIPS 141 (April) Interoperability and Security Requirements for Use of the Data Encryption Standard with CCITT Group 3 Facsimile Equipment (Withdrawn February 2000)
1985 AD	Publication of US FIPS 112 (May) Password Usage
1985 AD	Publication of US FIPS 113 (May) Computer Data Authentication

TIMELINE OF CRYPTOGRAPHY DEVELOPMENT:

1990 AD	Charles H. Bennett, Gilles Brassard, et al published their experimental results on Quantum Cryptography, which uses single photons to communicate a stream of key bits for some later Vernam Encipherment of a message or other uses. Assuming the laws of quantum mechanics hold, Quantum Cryptography provides not only secrecy, but a positive indication of eavesdropping, and a measurement of the maximum number of bits an eavesdropper might have captured. On the downside, Quantum Cryptography currently requires a fiber-optic cable between the two parties.
1990 AD	Xuejia Lai and James Massey in Switzerland published A Proposal for a New Block Encryption Standard, a proposed International Data Encryption Algorithm (IDEA) to replace DES. IDEA uses a 128-bit key, and employs operations which are convenient for general purpose computers, therefore making software implementations more efficient.
1991 AD	Phil Zimmermann released his first version of PGP (Pretty Good Privacy) in response to the threat by the FBI to demand access to the plaintext of the communications of citizens. PGP offered high security to the general citizen, and as such could have been seen as a competitor to commercial products like Mailsafe from RSADSI. However, PGP is especially notable because it was released as freeware, and has become a worldwide standard as a result, while its competitors of the time remain effectively unknown.
1992 AD	Publication of US FIPS 171 (April) Key Management Using ANSI X9.17
1993 AD	Publication of US FIPS 181 (October) Automated Password Generator
1994 AD	Publication of US FIPS 140-1 (January) Security Requirements for Cryptographic Modules
1994 AD	Publication of US FIPS 185 (February) Escrowed Encryption Standard
1994 AD	Publication of US FIPS 188 (September) Standard Security Labels for Information Transfer
1994 AD	Publication of US FIPS 190 (September) Guideline for the Use of Advanced Authentication Technology Alternatives
1994 AD	Publication of US FIPS 191 (November) Guideline for The Analysis of Local Area Network Security
1994 AD	Professor Ron Rivest, author of the earlier RC2 and RC4 algorithms included in RSADSI's BSAFE cryptographic library, published a proposed algorithm, RC5, on the Internet. This algorithm uses data-dependent rotation as its non-linear operation, and is parameterized so the user can vary the block size, number of rounds, and key length. It is still too new to have been analyzed enough to enable one to know what parameters to use for a desired strength, although an analysis by RSA Labs, reported at Crypto'95, suggests w=32, r=12 gives strength superior to DES. It should be remembered, however, this is just a first analysis.
1994 AD	The Blowfish cipher is described by Bruce Schneier. It is based on Feistel rounds, and the design of the f-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software.
1995 AD	Publication of US FIPS 180-1 (April) Secure Hash Standard
1997 AD	Publication of US FIPS 196 (February) Entity Authentication Using Public Key Cryptography
1999 AD	Publication of US FIPS 46-3 (October) Data Encryption Standard (DES) specifies the use of Triple DES
2000 AD	Publication of US FIPS 186-2 (January) Digital Signature Standard (DSS)
2001 AD	Publication of US FIPS 197 (November) Advanced Encryption Standard
2001 AD	Publication of change notice for US FIPS 186-2, Digital Signature Standard (DSS).
2002 AD	Publication of US FIPS 198 (March) The Keyed-Hash Message Authentication Code (HMAC)

2.3 Different types of cryptosystem

We'll mostly cover private key cryptography, with a mathematical foundation, based on substitutions, transpositions and mixing operations. We will also cover a little bit of public key cryptography. There are other systems, including *steganography* and *code books*.

2.3.1 Private key cryptography

- a private-key (or secret-key, or single-key) encryption algorithm is one where the sender and the recipient share a common, or closely related, key
- all traditional encryption algorithms are private-key
- Figure 3 shows an overview of a private-key encryption system and attacker

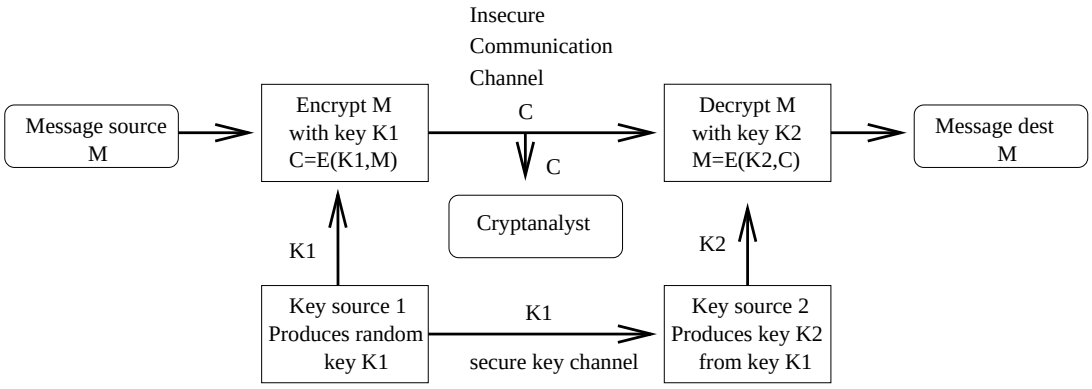


Figure 3: Symmetric private-key encryption system

Alice and Bob will adopt the following protocol:

First, choose a random key $K \in \mathcal{K}$, when they are in the same place and are not observed by Oscar; or when they do have access to a secure channel, in which case they can be in different places.

At a later time, suppose Alice wants to communicate a message to Bob over an insecure channel. We suppose that this message is a string

$$x = x_1 x_2 x_3 \dots x_n$$

for some $n \geqslant 1$, where each plaintext symbol $x_i \in \mathcal{P}$, $1 \leqslant i \leqslant n$.

Each x_i is encrypted using the encryption rule e_K specified by the predetermined key $K \in \mathcal{K}$. Hence, Alice computes $y_i = e_K(x_i)$, $\forall 1 \leqslant i \leqslant n$, and the resulting ciphertext string

$$y = y_1 y_2 y_3 \dots y_n$$

is sent over the channel.

When Bob receives $y_1 y_2 \dots y_n$, he decrypts it using the decryption function d_K, obtaining the original plaintext string, $x_1 x_2 \dots x_n$.

Note that each encryption function e_K is a one-to-one function. So if $\mathcal{P} = \mathcal{C}$ then the function is a *permutation*.

The important aspect of this type of cryptosystem is that a private-key cryptographic system is only as good as the method used for distributing keys.

Distribution methods include:

1. A key can be delivered by one user to the other directly (eg physically) or indirectly (eg over a secure channel).
2. A new key can be delivered by encrypting it with an older key and either using a direct connection or an indirect connection (eg over a regular insecure channel).

First option is awkward. Some form of the second option is widely accepted.

2.3.2 Public key cryptography

The idea behind a *public-key* system is that it might be possible to find a cryptosystem where it is computationally infeasible to determine d_K even if e_K is known. If so, then the encryption rule e_K could be made public by publishing it in a directory.

Consider the following analogy using padlocked boxes: private key schemes involve the sender putting a message in a box and locking it, sending that to the receiver, and somehow securely also sending them the key to unlock the box.

The radical advance in public key schemes was to turn this around. The receiver sends an unlocked box to the sender, who puts the message in the box and locks it (easy - and having locked it cannot get at the message), and sends the locked box to the receiver who can unlock it (also easy), having the key. An attacker would have to pick the lock on the box (hard).

- Public-key/two-key/asymmetric cryptography involves the use of two keys:
  1. a *public-key*, which may be known by anybody, and can be used to encrypt messages, and verify signatures
  2. a *private-key*, known only to the recipient, used to decrypt messages, and sign (create) signatures
- Anyone knowing the public key can encrypt messages or verify signatures, but cannot decrypt messages or create signatures, counter-intuitive though this may seem.
- Public key cryptography works by the clever use of number theory problems that are easy (P type) one way but hard (NP type) the other way, eg exponentiation vs logs, multiplication vs factoring.
- Note that public key schemes are neither more secure than private key (security depends on the keyspace size for both), nor do they replace private key schemes (they are too slow to do so); rather they complement them.

- An asymmetric public-key encryption system is shown in Figure 4. Contrast this with the traditional encryption system shown in Figure 3.

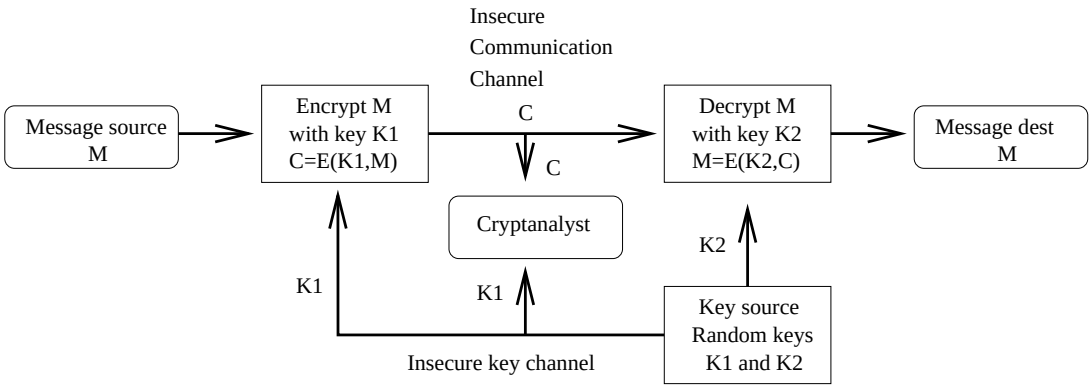


Figure 4: Public-key encryption system

- The development of public key cryptography is probably the single most significant advance in the 3000 year history of cryptography.

The implementation of public key cryptography is based on the idea of a **trapdoor** function $f : X \rightarrow Y$, such that

- f is publicly known, one-one and easy to compute;
- f^{-1} is difficult to compute;
- f^{-1} becomes easy to compute if a trapdoor is known.

First Public-Key scheme was proposed in 1976. Schemes include:

- RSA (named after its inventors: Rivest, Shamir and Adleman)
- Rabin
- ElGamal
- Elliptic Curve

2.3.3 Steganography

Methods of *concealing* the existence of data, in addition to the data itself.

- **Character Marking**: Selected letters of text are overwritten in pencil. The marks are not visible unless the paper is held at an angle to bright light.
- **Invisible ink**: Substances can be used that leave no visible trace until heat or some chemical is applied.
- **Pin punctures**: Small pin punctures on selected letters are not ordinarily visible unless paper is held in front of light.

- **Typewriter correction ribbon**: Used between lines typed with a black ribbon; results of typing visible only under strong light.
- Classical steganography in general requires a lot of physical overhead, and is not very mathematical in nature.

These classical techniques have modern analogies; for example *'Watermarking' digital data*.

- It is becoming important to be able to establish and protect ownership of digital data, including images, videos and audio.
- One way is to use steganography to digitally 'watermark' the data: that is, to digitally include some signature in the data.
- The signature must be
  - 'perceptually undetectable'
  - resistant to data manipulation, including
    - geometric distortion (rotating, scaling, cropping);
    - common signal processing (filtering, adjustments to colour space);
    - collusion and forgery.
  - extractable by the owner
  - non-destructive of the original data
- Possible methods (each with advantages and disadvantages), include:
  - modification of least significant bit
  - spread-spectrum watermarking

2.3.4 Code books

- A *code*, not a cipher.
- Rather than a mathematical rule for encrypting text, a code contains many pairs of the form (word,number). You must:
  - predict in advance all n words which could ever be required;
  - randomly associate a unique number to each word;
  - create two books, one with words in order and their numbers (used for encrypting), and one with numbers in order (for decrypting);
  - distribute encryption book to sender and decryption book to receiver.
- A code destroys statistical information in the message and may seem to be impossible to crack.
- Codes were heavily used in military applications, especially First World War.

The Zimmerman Telegram was sent from Germany's foreign minister to Germany's ambassador to the USA in 1917, while the USA was neutral. It was encoded using a 2-part code book with 10,000 words.

```
Berlin, January 19, 1917

On the first of February we intend to begin submarine warfare unrestricted.
In spite of this, it is our intention to endeavor to keep neutral the United
States of America. If this attempt is not successful, we propose an alliance
on the following basis with Mexico: That we shall make war together and
together make peace. We shall give general financial support, and it is
understood that Mexico is to reconquer the lost territory in New Mexico,
Texas, and Arizona. The details are left to you for settlement.

You are instructed to inform the President of Mexico of the above in the
greatest confidence as soon as it is certain that there will be an outbreak
of war with the United States and suggest that the President of Mexico, on
his own initiative, should communicate with Japan suggesting adherence at
once to this plan; at the same time, offer to mediate between Germany and Japan.

Please call to the attention of the President of Mexico that the employment
of ruthless submarine warfare now promises to compel England to make peace
in a few months.

Zimmerman (Secretary of State)
```

The British were able to crack the code book, using skill, luck and military intelligence. They had already made great progress, over a long time period. They released the details with a clever misinformation program. The USA joined the war one month later.

2.4 Introduction to cryptanalysis

We'll discuss specific types of attack when we discuss specific cryptographic methods. In each case, the object is to determine the key that was used.

The general assumption that is usually made is that the opponent, Oscar, knows the cryptosystem being used. This is usually referred to as *Kerchhoff's Assumption*.

Kerchhoff's Assumption means that you can't rely on 'secret details' of the algorithm to give security.

There are several different types of attack:

1. ciphertext only
    - Oscar only has access to some enciphered messages y. Cryptanalysis uses statistical information in the ciphertext to find likely plaintexts and it must be apparent when the correct plaintext is found.
2. known plaintext
    - Oscar knows (or strongly suspects) some plaintext-ciphertext pairs (x-y pairs) and uses this knowledge in attacking the cipher.
3. chosen plaintext
    - Oscar has obtained temporary access to the encryption machinery and can select plaintext x and obtain corresponding ciphertext y. The knowledge of algorithm structure is used in attack.
4. chosen plaintext-ciphertext
    - Oscar can select plaintext x and obtain corresponding ciphertext y, or select ciphertext y and obtain plaintext x. This allows further knowledge of algorithm structure to be used to attack the cipher.

Clearly, these four levels of attacks are enumerated in increasing order of strength.

2.5 Unconditional and computational security

Two different measures of security provided by ciphers:

1. A cipher is **unconditionally secure** if no matter how much resource is available, the cipher cannot be broken. (Except by sheer luck if the opponent happens to try the correct key.)
2. A cipher is **computationally secure** or **provably secure** if given practical (and conceivable) resources, the cipher cannot be broken.

Unconditional security is very hard to achieve, and very expensive in terms of key size. (The only known algorithm is Vernam's one-time pad.)

Most discussions are concerned with computational security: cracking is impossible, or at least very hard (with associated cost higher than the probable benefit).

3 Classical Cryptographic Techniques

There are two basic ideas behind classical ciphers: **substitution** and **transposition**.

- **substitution ciphers** - have letters replaced by others
- **transposition ciphers** - rearrange letters into a different order

A substitution cipher might be:

1. **monoalphabetic** - only one substitution alphabet is used, so each plaintext letter is mapped to a unique ciphertext letter; or
2. **polyalphabetic** - several substitution alphabets are used, so letters of the plaintext alphabet are mapped into letters of the ciphertext alphabet depending on their position in the text.
3. **stream ciphers** - a keystream is generated, which may or may not depend on the preceding plaintext, and used to encrypt the plaintext one character at a time.

Several ciphers may be concatenated together to form a **product cipher**. For information on lots of simple substitution and transposition ciphers, see A. Sinkov "Elementary Cryptanalysis", New Mathematical Library, Random House, 1968.

3.1 Reminder of some mathematics: modular arithmetic

You should have previously encountered modular arithmetic. This is a very quick reminder.

Definition 3.1 Let $a, b \in \mathbb{Z}$ and m be a positive integer. Then we write $a \equiv b \pmod{m}$ (sometimes $a = b \pmod{m}$) iff $m \mid (b - a)$. The integer m is called the *modulus*.

Remark 3.2 Equivalently, $a \equiv b \pmod{m}$ iff $\exists k \in \mathbb{Z}$ such that $a = mk + b$.

Theorem 3.3 Quotient-Remainder Theorem *Let $x \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$. Then there exist unique integers q and r such that $x = qm + r$ where $0 \leq r \leq m$.*

Definition 3.4 Suppose we divide a and b by m, obtaining integer quotients and remainders, where the remainders are between 0 and $m - 1$ (say $a = q_1 m + r_1$ and $b = q_2 m + r_2$, $0 \leq r_1, r_2 \leq m - 1$). Then $a \equiv b \pmod{m}$ iff $r_1 = r_2$. We use the notation $a \pmod{m}$ to mean the remainder when a is divided by m, so $a \equiv b \pmod{m}$ iff $a \pmod{m} = b \pmod{m}$. If we replace a by $a \pmod{m}$, we say that a is *reduced* mod m.

Definition 3.5 The set $S = \{0, 1, 2, \ldots, m-1\}$ is called the *least non-negative residue system modulo m*. Every integer is equivalent to exactly one element of S, modulo m.

Definition 3.6 We can now define addition and multiplication modulo m. Let $a, b \in \mathbb{Z}$. Then:

1. $(a + b) \pmod{m} = (a \pmod{m} + b \pmod{m}) \pmod{m}$
2. $(a \times b) \pmod{m} = (a \pmod{m} \times b \pmod{m}) \pmod{m}$

The set S, together with the operations $+$ and $\times$ (often written $\cdot$), form an algebraic structure called a *ring*, which is denoted $(\mathbb{Z}_m, +, \cdot)$, or more simply $\mathbb{Z}_m$. We often write $\mathbb{Z}_m = \{0, 1, \ldots m - 1\}$, with the understanding that the two operations $+$ and $\cdot$ are defined on the set.

Definition 3.7 Addition and multiplication in $\mathbb{Z}_m$ are *stable*. That is,

1. *Stability under addition*: If $a = b \pmod{m}$ and $c = d \pmod{m}$ then $a + c = b + d \pmod{m}$.
2. *Stability under multiplication*: If $a = b \pmod{m}$ and $c = d \pmod{m}$ then $a \cdot c = b \cdot d \pmod{m}$.

Remark 3.8 Thus, addition and multiplication in $\mathbb{Z}_m$ work exactly like addition and multiplication in $\mathbb{Z}$, except that all intermediate calculations can be reduced modulo m at any stage, and the results are reduced modulo m.

Remark 3.9 Note that exponents cannot be reduced. For example,

$$2^6 = 64 = 4 \pmod{6}$$

and

$$2^6 = 2^{3+3} = 2^3 \times 2^3 = 8 \times 8 = 2 \times 2 = 4 \pmod{6}$$

but

$$2^6 \neq 2^0 = 1 \pmod{6}.$$

Exercise 3.10 Evaluate each of: 11×16 in $\mathbb{Z}_{26}$; $(2k+1) \times (2m) \pmod{2}$, $k, m \in \mathbb{Z}_2$; $-15 \pmod{26}$.

Definition 3.11 The definitions of addition and multiplication in $\mathbb{Z}_m$ satisfy most of the familiar rules of arithmetic. These include:

1. addition is *closed*: for any $a, b \in \mathbb{Z}_m$, $a + b \in \mathbb{Z}_m$.
2. addition is *associative*: for any $a, b, c \in \mathbb{Z}_m$, $(a + b) + c = a + (b + c)$.
3. 0 is the *additive identity*: for any $a \in \mathbb{Z}_m$, $a + 0 = 0 + a = a$.
4. the *additive inverse* of any $a \in \mathbb{Z}_m$ is $m - a$. That is, $a + (m - a) = (m - a) + a = 0$ for any $a \in \mathbb{Z}_m$.
5. addition is *commutative*: for any $a, b \in \mathbb{Z}_m$, $a + b = b + a$.
6. multiplication is *closed*: for any $a, b \in \mathbb{Z}_m$, $ab \in \mathbb{Z}_m$.
7. multiplication is *commutative*: for any $a, b \in \mathbb{Z}_m$, $ab = ba$.
8. multiplication is *associative*: for any $a, b, c \in \mathbb{Z}_m$, $(ab)c = a(bc)$.
9. 1 is the *multiplicative identity*: for any $a \in \mathbb{Z}_m$, $a \times 1 = 1 \times a = a$.
10. multiplication *distributes* over addition. That is, for any $a, b, c \in \mathbb{Z}_m$, $(a + b)c = (ac) + (bc)$, and $a(b + c) = (ab) + (ac)$.

Remark 3.12 Properties 1 to 4 say that $\mathbb{Z}_m$ forms an algebraic structure called a *group* with respect to addition. Properties 1 to 10 establish that $\mathbb{Z}_m$ is a *ring*.

Definition 3.13 Since additive inverses exist in $\mathbb{Z}_m$, for $a, b \in \mathbb{Z}_m$ we can define subtraction, $a - b$, to be $a + (m - b)$. Equivalently, we can compute the integer $a - b$ and then reduce modulo m.

Definition 3.14 Suppose $a \in \mathbb{Z}_m$. The *multiplicative inverse* of a, if it exists, is an element $a^{-1} \in \mathbb{Z}_m$ such that $aa^{-1} \equiv 1 \bmod m$. Note that multiplicative inverses do not always exist; for example, there is no multiplicative inverse of 2 in $\mathbb{Z}_4$.

Definition 3.15 Given two integers a and b (not both 0), d is their *greatest common divisor* (denoted $d = \gcd(a, b)$) if d is the largest integer which divides evenly into both a and b.

Definition 3.16 Two integers a and b are called **relatively prime** if $\gcd(a, b) = 1$.

Remark 3.17 In most cases in cryptography and coding theory, we will be concerned with integers a and b for which $\gcd(a, b) = 1$. The reason for this is that:

Theorem 3.18 *An element $a \in \mathbb{Z}_m$ has a multiplicative inverse a^{-1} if and only if $\gcd(a, m) = 1$.*

Remark 3.19 Hence we have:

1. $(\mathbb{Z}_m, \cdot)$ is never a group because 0 never has a multiplicative inverse.
2. $(\mathbb{Z}_m \setminus \{0\}, \cdot)$ is a group whenever m is prime.
3. $(\mathbb{Z}_m \setminus \{0\}, \cdot)$ is never a group if m is not prime.

Remark 3.20 Suppose we wish to find a^{-1} for some element $a \in \mathbb{Z}_m$. We can first apply Euclid's Algorithm to determine $\gcd(a, n)$, and hence determine whether a^{-1} exists. If it does exist, then the Extended Euclid's Algorithm can be used to find it.

Algorithm 3.21 Euclidean algorithm
Given two integers a and b (not both 0), $a, b \geq 0, a \geq b$, $d = \gcd(a, b)$ may be found as follows:

Step 1.	$a = bq_1 + r_1$	$0 \leq r_1 \leq b$
Step 2.	$b = r_1 q_2 + r_2$	$0 \leq r_2 \leq r_1$
Step 3.	$r_1 = r_2 q_3 + r_3$	$0 \leq r_3 \leq r_2$
.	.	.
.	.	.
Step n.	$r_{n-2} = r_{n-1} q_n + r_n$	$0 \leq r_n \leq r_{n-1}$
Step n+1.	$r_{n-1} = r_n q_{n+1}$	(so $r_{n+1} = 0$)

Then $r_n = \gcd(a, b)$. □

Exercise 3.22 Use the Euclidean algorithm to find $\gcd(26, 11)$.

Algorithm 3.23 Extended Euclidean algorithm
Given two integers a and b with $d = \gcd(a, b)$, it is possible to find integers x and y such that $d = ax+by$.

The general method is to apply the Euclidean algorithm *backwards*, solving for d in terms of a and b.

□

Example 3.24 *Noting that gcd$(26, 11) = 1$, find x and y such that $26x + 11y = 1$ (mod 26).*

We have applied the Euclidean algorithm above. Then

$$\begin{aligned} 1 &= 4 - 3 \cdot 1 \\ \implies 1 &= 4 - (11 - 4 \cdot 2) \\ \implies 1 &= 4 \cdot 3 - 11 \\ \implies 1 &= (26 - 11 \cdot 2) \cdot 3 - 11 \\ \implies 1 &= 26 \cdot 3 - 11 \cdot 7 \end{aligned}$$

Exercise 3.25 Find the multiplicative inverse of 11 (mod 26).

3.2 Plaintext space

Unless otherwise stated, we will make the following assumptions about the plaintext space:

- The plaintext language is English.
- We are only concerned with alphabetical characters $abc \ldots z$ and $ABC \ldots Z$.

- Without loss of generality, we'll assume all plaintext is lowercase, and will write it in lowercase.
- We will write ciphertext in uppercase.
- There is a one-one correspondence between the characters $abc\ldots z$ in the plaintext space and $\mathbb{Z}_{26}$, and between the characters $ABC\ldots Z$ in the ciphertext space and $\mathbb{Z}_{26}$.

Example 3.26 *The correspondence between letters in the plaintext and ciphertext spaces and* $\mathbb{Z}_{26}$ *is:*

letter:	a,A	b,B	c,C	d,D	e,E	f,F	g,G	h,H	i,I	j,J	k,K	l,L	m,M
number:	0	1	2	3	4	5	6	7	8	9	10	11	12
letter:	n,N	o,O	p,P	q,Q	r,R	s,S	t,T	u,U	v,V	w,W	x,X	y,Y	z,Z
number:	13	14	15	16	17	18	19	20	21	22	23	24	25

3.3 Monoalphabetic substitution ciphers

The keyspace is a (sub)set of permutations on $\{0, 1, \ldots, 25\}$.

Algorithm 3.27 General monoalphabetic substitution cipher
For a given key (permutation) π*, and* $x_1, x_2, \ldots, x_n \in \mathcal{P}$*,* $y_1, y_2, \ldots, y_n \in \mathcal{C}$*,*

$$e_\pi(x_1 x_2 \ldots x_n) = \pi(x_1)\pi(x_2)\ldots\pi(x_n),$$

and

$$d_\pi(y_1 y_2 \ldots y_n) = \pi^{-1}(y_1)\pi^{-1}(y_2)\ldots\pi^{-1}(y_n),$$

where π^{-1} *is the inverse permutation to* π*.* □

3.3.1 The Affine cipher

- An affine cipher has an encryption function of the form

  $$e(x) = (ax + b) \mod 26,$$

  where $a, b \in \mathbb{Z}_{26}$, $x \in \mathcal{P}$. The key is (a, b).
- For decryption to work, we need to ask: when is an affine function injective (ie one-one)? In other words, for every $y \in \mathbb{Z}_{26}$, we want the congruence

  $$ax + b \equiv y \mod 26$$

  to have a unique solution for x. This congruence is equivalent to

  $$ax \equiv y - b \mod 26.$$

Theorem 3.28 *The congruence* $ax \equiv c \pmod m$ *has a unique solution* $x \in \mathbb{Z}_m$ *for every* $c \in \mathbb{Z}_m$ *if and only if* $\gcd(a, m) = 1$*.*

Exercise 3.29 Explain why it is not sensible to choose $a = 0$ in a key for the Affine cipher.

Exercise 3.30 Comment on the validity of choosing $b = 0$ in a key for the Affine cipher.

Exercise 3.31 Demonstrate, with an example, why it is invalid to choose $a = 13$ in a key for the Affine cipher.

Exercise 3.32 What is the number of permissible keys in the Affine Cipher over $\mathbb{Z}_{26}$?

Algorithm 3.33 Affine cipher
Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ *and let*

$$\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : \gcd(a, 26) = 1\}.$$

For $K = (a, b) \in \mathcal{K}$, *define*

$$e_K(x) = (ax + b) \mod 26$$

and

$$d_K(y) = a^{-1}(y - b) \mod 26,$$

where $(x, y \in \mathbb{Z}_{26})$. □

Exercise 3.34 Let $K = (a, b) = (7, 3)$. Find the ciphertext for the plaintext *dog*.

Exercise 3.35 Let $K = (a, b) = (3, 5)$. Find the plaintext corresponding to ciphertext *LFK*.

You may have encountered two special cases of the affine cipher.

- The affine cipher with $(a, b) = (1, 3)$ is the Caesar cipher, reputedly used by Julius Caesar.
- An affine cipher with $a = 1$ is called a shift cipher. A shift cipher has 26 possible keys (but only 25 sensible ones).

3.3.2 Cryptanalysis of the Affine cipher

We consider a **known-plaintext** attack on the Affine cipher. Consider the following ciphertext obtained from the Affine Cipher.

FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSHVUFEDK
APRKDLYEVLRHHRH

Suppose we know that R is the encryption of e and K is the encryption of t. So we have $e_K(4) = 17$ and $e_K(19) = 10$. Recall that $e_K(x) = ax + b$, where a and b are unknowns. So we have

$$4a + b = 17 \pmod{26}$$
$$19a + b = 10 \pmod{26}.$$

Exercise 3.36 Solve the simultaneous congruences given above.

Using $d_K(y) = 9y - 19$ we find the following plaintext:

algorithmsarequitegeneraldefinitionsofarit
hmeticprocesses

Using a known-plaintext attack with knowledge of two plaintext-ciphertext pairs made it very easy obtain the key. However, an affine cipher is also easy to crack using a ciphertext-only attack. A guess and check approach would involve making a sensible guess as to two potential plaintext-ciphertext pairs, based on some knowledge of the most frequently used letters in the English language, and then solving simultaneous congruences as we did above. If the first guess didn't work out, then we would try with another pair, and repeat this process until we obtained the key. Note that the most frequently used letters in the English language are e and t and the most frequent ciphertext characters are: R (8 occurrences), D (6 occurrences), E, H, K (5 occurrences), and F, S, V (4 occurrences each).

3.3.3 Mixed alphabets

The most general form of a monoalphabetic substitution cipher is an arbitrary mixed alphabet. Here each plaintext letter is assigned a unique random ciphertext letter, hence the key is 26 letters long, and specifies the permutation of the 26 letters. An affine cipher is a special case of a mixed alphabet cipher and because of the extra structure, the 26 letter long key can be described by the ordered pair (a, b).

Example 3.37 *Let π be the following permutation:*

```
Plain:   abcdefghijklmnopqrstuvwxyz
Cipher:  DKVQFIBJWPESCXHTMYAUOLRGZN
```

Then the following plaintext encrypts to:

```
Plaintext:   ifwewishtoreplaceletters
Ciphertext:  WIRFRWAJUHYFTSDVFSFUUFYA
```

3.3.4 Cryptanalysis of mixed alphabet ciphers

Exercise 3.38 Determine the total number of keys for a mixed alphabet cipher.

- With so many keys (more than 4×10^{26} keys), you might think this is secure, but it isn't!
- The problem is not the number of keys, rather:
  - there is lots of statistical information in message;
  - can solve the problem piece by piece;
  - use informed guesses and knowledge of the plaintext language;
  - can get key nearly right, and nearly get message.
- In most languages, letters don't occur with equal frequency: for example, in English, 'e' is by far most common. Thus we can use statistical properties of the plaintext language, such as letter frequency analysis, to crack the cipher.
- Various people have estimated relative frequencies of the 26 letters by compiling stats from numerous novels, magazines and newspapers. Figure 5 gives estimates from Beker and Piper.
- It may also be useful to consider sequences of two or three consecutive letters called *digrams* and *trigrams*, respectively. The 30 most common digrams are (in decreasing order) TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI and OF. The twelve most common trigrams are (in decreasing order) THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR and DTH.
- Note that letter frequencies are different for different languages (see Appendix A in Seberry and Pieprzyk), so knowledge of the plaintext language is important.

- Since a monoalphabetic substitution does not change relative letter frequencies, you can use the tables to compare letter frequencies in the plaintext language to letter frequencies in the ciphertext and guess the permutation letter by letter.
- You do need a moderate amount of ciphertext (100+ letters) for letter frequencies to become apparent.

letter	freq	letter	freq	letter	freq	letter	freq
A	0.082	N	0.067	E	0.127	M	0.024
B	0.015	O	0.075	T	0.091	W	0.023
C	0.028	P	0.019	A	0.082	F	0.022
D	0.043	Q	0.001	O	0.075	G	0.020
E	0.127	R	0.060	I	0.070	Y	0.020
F	0.022	S	0.063	N	0.067	P	0.019
G	0.020	T	0.091	S	0.063	B	0.015
H	0.061	U	0.028	H	0.061	V	0.010
I	0.070	V	0.010	R	0.060	K	0.008
J	0.002	W	0.023	D	0.043	J	0.002
K	0.008	X	0.001	L	0.040	Z	0.001
L	0.040	Y	0.020	C	0.028	X	0.001
M	0.024	Z	0.001	U	0.028	Q	0.001

Figure 5: Estimated frequencies of letters in English text (unsorted at left, sorted at right)

The following graph (Figure 6) shows relative frequencies of characters in English plaintext.

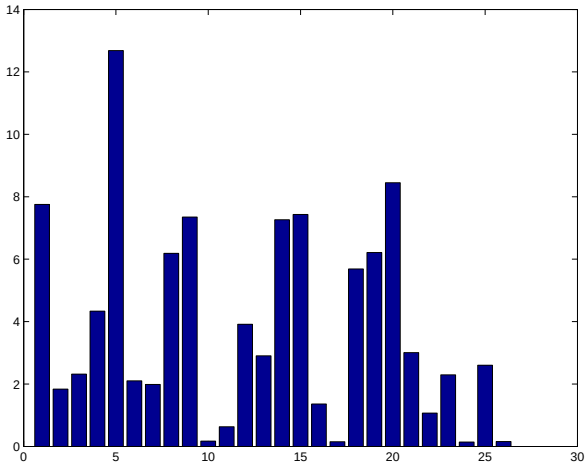
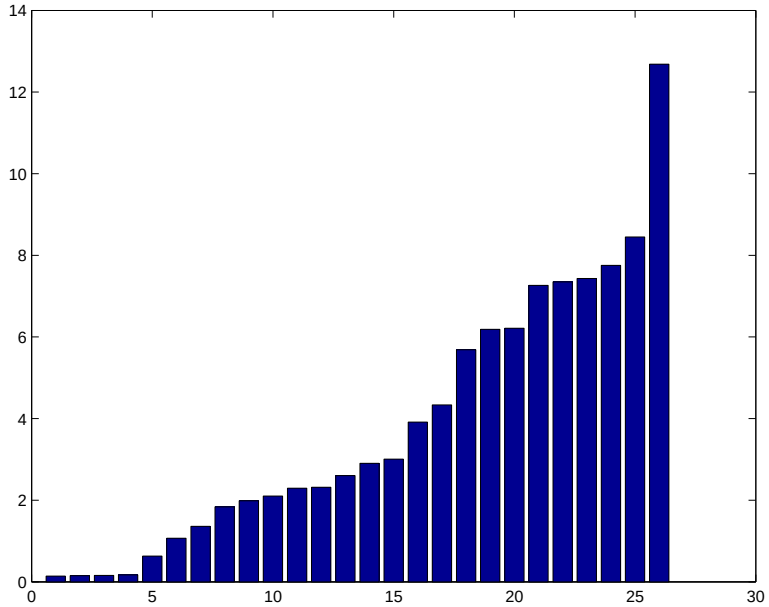
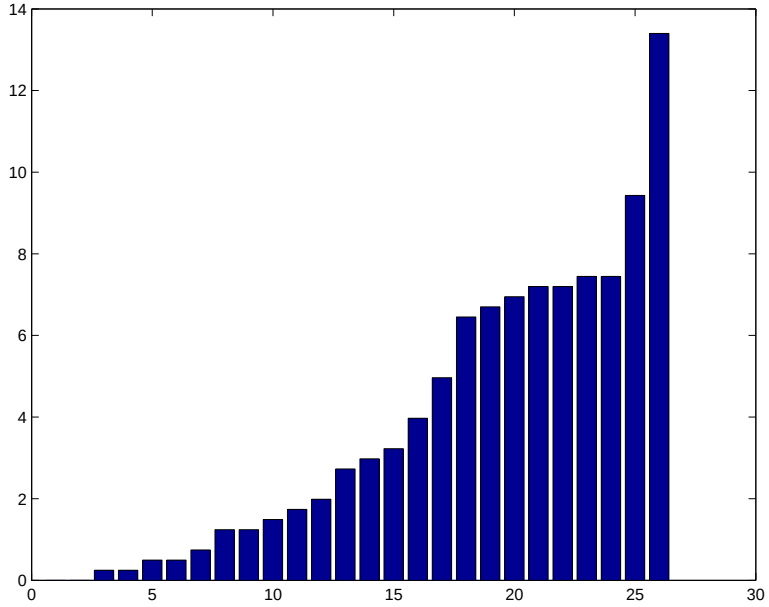


Figure 6: Relative frequencies of characters in plaintext

By reordering, we can obtain a graph of the sorted frequencies of letters in English. If some ciphertext has a similar distribution of sorted letter frequencies, then is probably English text, encrypted with a monoalphabetic substitution cipher. The two graphs below show how comparison could be done. The plaintext and ciphertext come from different files.



Sorted relative frequencies of characters in plaintext



Sorted relative frequencies of characters in (mixed-alphabet) ciphertext

Exercise 3.39 Decrypt the following ciphertext, which was encoded using a mixed alphabet. To help, Figures 7 and 8 give the sorted frequencies of letters in the ciphertext. Figure 9 gives the sorted relative frequencies of the 20 most frequent digrams.

```
CBCPRBWVBTVGCMXWPRJJDRFLGZUJXASXAB
TRBRPCGSJXQRGCGKZPPXPPCZGZERSZZAEZ
WBVGXQVPBYXCGURGBZERUCEXTZUXMXWJCB
BJXLGZUGBTXEXXJCGSPZWMCXUPZEPVFTRQ
RGQRDYXZGTCPECWPBXGBXWCGSRGXCSTYZV
WTZZABTCPBWVBTCPPZUXJJECOXACGBTXQC
GAPZEBTXPVWWZVGACGSERQCJCXPBTRBTXC
PFZGPCAXWXARPBTXWCSTBEVJKWZKXWBDZE
PZQXZGXZWZBTXWZEBTXCWARVSTBXWP
```

X:	34	C:	27	Z:	27	B:	26	P:	24	G:	23	T:	19
W:	19	R:	17	E:	13	J:	11	V:	11	A:	10	S:	9
Q:	7	U:	7	F:	3	K:	3	M:	3	Y:	3	D:	3
L:	2	O:	1	N:	0	H:	0	I:	0				

Figure 7: **Absolute frequencies of characters in the ciphertext**

X:	0.1126	C:	0.0894	Z:	0.0894	B:	0.0861	P:	0.0795	G:	0.0762	T:	0.0629
W:	0.0629	R:	0.0563	E:	0.0430	J:	0.0364	V:	0.0364	A:	0.0331	S:	0.0298
Q:	0.0232	U:	0.0232	F:	0.0099	K:	0.0099	M:	0.0099	Y:	0.0099	D:	0.0099
L:	0.0066	O:	0.0033	N:	0.0000	H:	0.0000	I:	0.0000				

Figure 8: **Relative frequencies of characters in the ciphertext**

BT:	0.0381	CG:	0.0277	TX:	0.0242	XW:	0.0242	ZE:	0.0208	CP:	0.0173
PB:	0.0173	PZ:	0.0173	GB:	0.0138	GS:	0.0138	RG:	0.0138	WZ:	0.0138
XA:	0.0138	XC:	0.0138	ZG:	0.0138	ZU:	0.0138	BX:	0.0104	ER:	0.0104
GX:	0.0104	JX:	0.0104								

Figure 9: **Frequently occurring digrams in the ciphertext**

3.4 Polyalphabetic substitution ciphers

- The ciphers discussed so far have been monoalphabetic, with single substitution alphabets.
- In general more than one substitution alphabet is used → **polyalphabetic cipher**
- This makes cryptanalysis harder:
  - more alphabets to guess;
  - flattens frequency distribution (as each plaintext letter may/will get replaced by several ciphertext letters) (see for example Figure 10 on page 32).

3.4.1 Vigenere Cipher

- The Vigenere cipher uses multiple mixed alphabets, each is a shift cipher.
- The key is m letters long (usually a word), and the ith letter specifies the ith shift to use.
- Use each shift in turn to encrypt one letter, repeating from the start after m letters in the plaintext.
- The number of possible keywords in the Vigenere Cipher is 26^m.
- For centuries it was believed to be uncrackable: 'Le Chiffre Indechiffrable'

Exercise 3.40 Explain why the number of possible keywords in the Vigenere Cipher is 26^m.

Algorithm 3.41 Vigenere cipher
Let m be some fixed positive integer.

Define $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$. For a key $K = (k_1, k_2, \ldots, k_m)$, we define

$$e_K((x_1, x_2, \ldots, x_m)) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$$

and

$$d_K((y_1, y_2, \ldots, y_m)) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m),$$

where all operations are performed in $\mathbb{Z}_{26}$. Repeat key after m characters. □

Exercise 3.42 Suppose $m = 5$ and the keyword is FISHY. Find the ciphertext for the following plaintext: *ericthehalibut.*

Exercise 3.43 Suppose $m = 6$ and the keyword is CIPHER. Find the plaintext for the following received message: *VPXZGIAXIVWPUBTTMJPWIZITWZT.*

3.4.2 Cryptanalysis of the Vigenere cipher

Example 3.44 The following text was encrypted using the Vigenere cipher with unknown key length.

TBWOMYLUYXSSLIFZFBULFPEFOHDRSAEGAEISRIEFIEYNGGEZOM
LVSAVYFBIATXALXNNWWYSETBWRSHNADTHVEMOASJELWNWHAFDR
XRMJLXHGHCLAIETBJXIBRZGNVGIGWLEJEYCMSCASLAIVRXMMCG
ONZXMEAOFMEADNGTQVLFAGIESMZHTWUMLHZRRNZXANYNZXXJOS
GNRTEMLHJGHYXTQVLSUTXUELAGINNXDRHVAQWKICALLBGHLUJE
CSRYINIATCFMLRSYSMXRNNAHRFTBWBVZIHVLARRYEHVRVUUTRG
TBSGXUECJLMFTYJLEADQZXRAONZBRTBYLMIEOZXXVRDUOTPXTI
EXVLTIFPEFNYUXWFALQMSNMOKXXUECJFSENCFZLBULKTRQFOJG
MFHWGGZRRMSMMBNZGKXUEYNXRVNASGHUOQWOIEBUJXSSNYOLXU
EWGNRGRSAGKRNYJTPZIAZMFRTBWREYWUQLGBNNJBZRDNGEINRH
KHQRFLGFXUECJTYATULIVRSYFMMADYWWXUESOXVRWYDEWHPJDB
IQBILAAVTBFXAFAHVAECPCFXWFBSLAIEEWWGXNRLAOEYOZSFMY
INATVRGCEXRGIHLAIAECYAFBULZHSQINOTWGOLWFEVNNZXAUOF
WPMATYJTRQMYJRXBNQSLXUEBWTHDUUJMIES

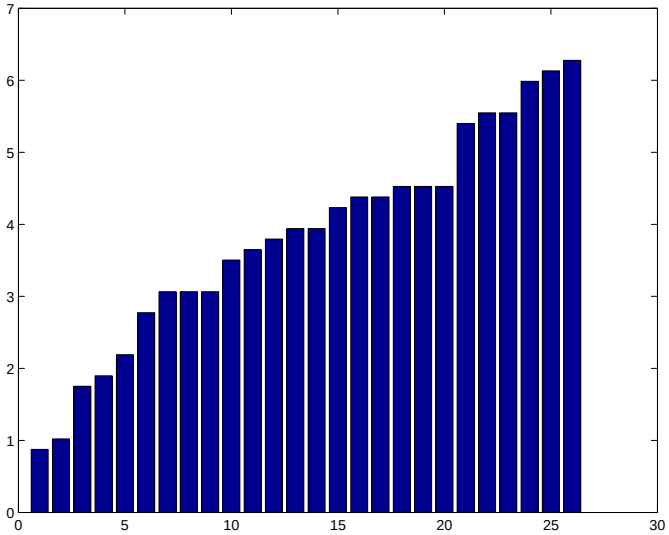


Figure 10: Sorted relative frequencies of characters in (Vigenere) ciphertext

Remark 3.45 Note that now, a given letter in the plaintext may encrypt to multiple distinct letters in the ciphertext. This makes cryptanalysis harder. However, it is certainly still not too hard, using *Kasiski's test* and Friedman's *Index of coincidence*. To crack the Vigenere cipher we must first determine the key length m and then decrypt each of m shift ciphers.

To determine the key length m, we can use any of three methods:

- Kasiski's Test (Kasiski announced this in 1863 but Babbage had discovered it in 1854);
- Friedman's Test method 1 (Friedman invented this in 1925);
- Friedman's test method 2.

Kasiski's Test

The test is based on Kasiski's Observation that two identical pieces of plaintext will encrypt to the same ciphertext whenever their (plaintext) occurrence is x positions apart, where $x \equiv 0 \pmod{m}$.
To apply Kasiski's test, we search the ciphertext for pairs of identical strings of length ≥ 3 and record the distances between starting positions. Let d_1, d_2 ..., be such distances. Then we guess that $m|\gcd(d_1, d_2, \dots)$.

Exercise 3.46 Apply Kasiski's test to guess the most likely value of m for the text in Example 3.44. Here is some information about the position of repeated strings in the ciphertext. The numbers give the positions at which the corresponding substring starts.

tbw:	1 73 277 475	xss:	10 442	bul:	18 390	rsa:	28 457
eyn:	42 421	iat:	59 257	etb:	72 114	lai:	111 141 615
ivr:	143 520	onz:	151 325	nzx:	152 188 194 644	mea:	155 160
ead:	161 317	dng:	163 493	agi:	171 225	ies:	173 683
mlh:	182 206	zrr:	185 407	zxx:	195 338	gnr:	201 453
hva:	233 566	yin:	254 600	rsy:	264 522	tyj:	313 655
rtb:	329 474	mie:	334 682	xvr:	340 538	xwf:	364 574
bul:	390 624	trq:	394 658	yjt:	464 656	aec:	568 618
lai:	579 615	fbul:	17 623	fpef:	21 357	tbwr:	73 475
laie:	111 579	slai:	140 578	tqvl:	166 214	nzxa:	188 644
lxue:	448 670	xuecj:	305 377 509	xue:	221 305 377 419 449 509 533 671		

Friedman's Test

Friedman's test is stronger than Kasiski's test and it uses the *index of coincidence* which is based on the question "If a pair of letters is selected from a text, what is the probability that the two letters are equal?"
Given a text of length n, let f_i be the frequency of letter i, for $i = 0, 1, \ldots, 25$.
The number of pairs of letters in the text is $\binom{n}{2} = \frac{n(n-1)}{2}$.
The number of pairs of the letter A in the text is $\binom{f_0}{2} = \frac{f_0(f_0-1)}{2}$.

Exercise 3.47 Let I_c be the probability that a randomly chosen pair of letters in the text are equal; I_c is called the index of coincidence of the text. Determine a formula for I_c in terms of f_i and n.

Now let p_i be the expected probability of occurrence of letter i, for $i = 0, 1, \ldots, 25$ in English text. The probability that any randomly chosen pair of letters is 'AA' is p_0^2 and the probability that any randomly chosen pair of letters is 'BB' is p_1^2, etc.

Exercise 3.48 Determine the probability of a randomly chosen pair of letters being equal when chosen from English text.

Exercise 3.49 Determine the probability of a randomly chosen pair of letters being equal when chosen from truly random text.

Exercise 3.47 can be used to calculate the index of coincidence for a piece of ciphertext, Exercise 3.48 gives the expected index of coincidence for English, and Exercise 3.49 gives the expected index of coincidence for random text. So given a piece of ciphertext, evaluate I_c. If it is approximately 0.065, then it was probably encrypted using a monoalphabetic substitution cipher (and you can use frequency analysis to crack it) or a transposition cipher (more about that later). If your I_c is not approximately 0.065, then it may have been encrypted using a polyalphabetic substitution cipher and Friedman's tests give two ways to proceed to find the key length m.

Friedman method 1.

Guess m, partition the ciphertext into m groups, each of length n/m and evaluate I_c for each group. If all are approximately 0.065, then you have the correct m. If not, try a different m.

Example 3.50 Apply Friedman's method 1 to Example 3.44 to determine the most likely value of the key length m.
For each group of letters, we need to count the letter frequency f_i for each letter in the group and also count the total number of letters n in the group. Then $I_c = \dfrac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}$ for that group.

For $m = 1$, $I_c = 0.042754$
For $m = 2$, $I_c = 0.045795, 0.046355$
For $m = 3$, $I_c = 0.058377, 0.049733, 0.050815$
For $m = 4$, $I_c = 0.045152, 0.048848, 0.047953, 0.044100$
For $m = 5$, $I_c = 0.040897, 0.041327, 0.038858, 0.048411, 0.043366$
For $m = 6$, $I_c = 0.069108, 0.065363, 0.067691, 0.071883, 0.073125, 0.062723$
For $m = 7$, $I_c = 0.041658, 0.046707, 0.048811, 0.042710, 0.038923, 0.038292, 0.039734$
For $m = 8$, $I_c = 0.045691, 0.048974, 0.045691, 0.040219, 0.045144, 0.047899, 0.052381, 0.047339$

Friedman method 2.

Use the formula

$$m = \frac{0.027n}{(n-1)I_c - 0.038n + 0.065} \quad \text{where } I_c = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}.$$

How did Friedman come up with the above formula? Consider the arrangement of the ciphertext under the keyword $k_1k_2\ldots k_m$.

$$\begin{array}{ccccc} k_1 & k_2 & k_3 & \ldots & k_m \\ c_1 & c_2 & c_3 & \ldots & c_m \\ c_{m+1} & c_{m+2} & c_{m+3} & \ldots & c_{2m} \\ \vdots & & & & \end{array}$$

The text in each column has been encrypted using the same shift. Thus we expect the index of coincidence within a column to be around 0.065. There are approximately $\frac{n}{m}$ letters in each column. Thus, within a column we expect $0.065\binom{\frac{n}{m}}{2}$ equal pairs of letters, and within the entire body of ciphertext, we expect the number of equal pairs in which the two letters come from the same column to be

$$0.065\binom{\frac{n}{m}}{2}m = 0.065\frac{\frac{n}{m}(\frac{n}{m}-1)}{2}m = 0.065\frac{n(n-m)}{2m}.$$

Now consider pairs of letters from distinct columns. Assume that each column has been encoded using a different shift, so $k_i \neq k_j$ for $i \neq j$. We expect the index of coincidence for pairs of letters from different columns to be around 0.038. Thus within the entire body of ciphertext, we expect the number of equal pairs from different columns to be

$$\binom{m}{2}\left(\frac{n}{m}\right)\left(\frac{n}{m}\right)(0.038) = 0.038\frac{m(m-1)}{2}\frac{n^2}{m^2} = 0.038\frac{n^2(m-1)}{2m}.$$

Adding these two values gives the expected number of equal pairs in the entire ciphertext

$$P = 0.065\frac{n(n-m)}{2m} + 0.038\frac{n^2(m-1)}{2m} = \frac{0.065n(n-m) + 0.038n^2(m-1)}{2m}.$$

The expected proportion of equal pairs is

$$\begin{aligned}\frac{P}{\binom{n}{2}} &= \frac{0.065n(n-m) + 0.038n^2(m-1)}{2m}\frac{2}{n(n-1)} \\ &= \frac{0.065(n-m) + 0.038n(m-1)}{m(n-1)} \\ &= \frac{0.027n + m(0.038n - 0.065)}{m(n-1)}\end{aligned}$$

and this (by definition) is equal to I_c. By rearranging we see that

$$m = \frac{0.027n}{(n-1)I_c - 0.038n + 0.065} \quad \text{where } I_c = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}.$$

Note that if n is large, then you can use the approximation

$$m \approx \frac{0.027n}{nI_c - 0.038n} = \frac{0.027}{I_c - 0.038}.$$

Exercise 3.51 Apply Friedman's method 2 to find the most likely value of m for the text in Example 3.44.

Now that we have found the most likely value of the key length m, we need to find the actual key. There are two ways to do this. The first is to use standard methods of cryptanalysis for shift ciphers on each column. Either use letter frequency analysis or rotate each one of the m shift ciphers (columns) in turn until the message looks sensible.

Exercise 3.52 Writing the ciphertext from Example 3.44 in 6 columns, the second column of ciphertext with its letter frequencies is given below.

BUILHGINMYXWBAMLFJCBZGYSXNONFMMNNSMYSLXQLUYCYNBHYUBCYQNYZ
UIIYLOCCLOWMZYAQUYWSYABUNNHLCUYYSYJIBHCSWLZNCHCLNLNFYYQBU

Letter frequencies:												
A	B	C	D	E	F	G	H	I	J	K	L	M
3	8	9	0	0	3	2	5	5	2	0	10	6
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
12	3	0	4	0	6	0	8	0	4	3	17	4

Determine the most likely value of the second character of the keyword.

The second method uses mutual index of coincidence to find the relative shift between two characters of the keyword. Let x and x' be strings of text of lengths n and n' respectively and let $MI_c(x, x')$ be the probability that a random element of x equals a random element of x'. For $i = 0, \ldots 25$, let f_i and f'_i be the frequencies of the character i in x and x' respectively. Then the *mutual index of coincidence* of the two pieces of text is

$$MI_c(x, x') = \sum_{i=0}^{25} \frac{f_i f'_i}{nn'}.$$

Since we know the value of our keyword length m, we can write our ciphertext in m columns to obtain m sub-ciphertexts $y_1, y_2, \ldots, y_m$. Assume the key is $K = k_1 k_2 \ldots k_m$ and that column y_i is the result of encryption using the shift k_i. We will now work out a formula for the mutual index of coincidence for the sub-ciphertexts y_i and y_j.
Choose two ciphertext columns y_i and y_j and choose a character from each column at random, c_i from y_i and c_j from y_j. What is the probability that $c_i = c_j = A$?
If $c_i = A$, the corresponding plaintext had to be character $(0 - k_i) \pmod{26}$, and the probability that this was the plaintext is p_{0-k_i}. Similarly, the probability that $c_j = A$ is p_{0-k_j}. Thus the probability that they are both A is $p_{0-k_i} \times p_{0-k_j}$. Similarly, the probability that they are both B is $p_{1-k_i} \times p_{1-k_j}$. Thus we can calculate the mutual index of coincidence for columns y_i and y_j.

$$MI_c(y_i, y_j) = \sum_{h=0}^{25} p_{h-k_i} p_{h-k_j} = \sum_{g=0}^{25} p_g p_{g+k_i-k_j} \text{ or alternatively } \sum_{g=0}^{25} p_g p_{g+k_j-k_i}.$$

Our goal is to find the relative shift between the columns, so we want to find $k_j - k_i$. For each $0 \leq i < j \leq m$, consider columns y_i and y_j. Let columns y_i and y_j have n and n' characters respectively, and let f_g and f'_g be the frequencies of character g in columns y_i and y_j respectively. For each possible relative shift $s = 0, 1, 2, \ldots, 25$, calculate the mutual index of coincidence

$$\sum_{g=0}^{25} \frac{f_g f'_{g+s}}{nn'}.$$

If this value is around 0.065, then it is probable that $s = k_j - k_i$. Note that for the English language the expected value of MI_c for an incorrect relative shift is between 0.031 and 0.045, so it should be clear when the correct one has been found.

Exercise 3.53 Apply the method of mutual index of coincidence to find the most likely shift between columns 2 and 3 for the ciphertext in Example 3.44.
Here is the sub-ciphertext from column 3 along with the letter frequencies.

WYFFDAEGLFAWWDOWDLLJGWCLMZFGAZLZZGLXUADWLJIFSAWVEUSJJZZLX
OEFUQKJFKJGSGNSWJOGAJZWQJGKGJLFWODDLFVFLWASAELYZOWZWJJSWJ

Letter frequencies:												
A	B	C	D	E	F	G	H	I	J	K	L	M
8	0	1	6	4	10	9	0	1	13	3	12	1
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	5	0	2	0	6	0	3	2	14	2	2	9

Example 3.54 Determine the keyword used to obtain the ciphertext in Example 3.44.
Here are the sub-ciphertexts for the ciphertext in Example 3.44 with a keyword length of 6:

1: TLLUOERYOVTNTNEEAMHTRIEAROADLSURYOEHLENAALRTSNTIRVTETDOBO
DTTNAMENUFHRNENOBNERNITWNDRFETSDEWPBTAPBEROIGIEUIONOTMNEUS

2: BUILHGINMYXWBAMLFJCBZGYSXNONFMMNNSMYSLXQLUYCYNBHYUBCYQNYZ
UIIYLOCCLOWMZYAQUYWSYABUNNHLCUYYSYJIBHCSWLZNCHCLNLNFYYQBU

3: WYFFDAEGLFAWWDOWDLLJGWCLMZFGAZLZZGLXUADWLJIFSAWVEUSJJZZLX
OEFUQKJFKJGSGNSWJOGAJZWQJGKGJLFWODDLFVFLWASAELYZOWZWJJSWJ

4: OXZPREFGVBLYRTANRXAXNLMAMXMTGHHXXNHTTGRKBENMMHBLHTGLLXBMX
TXPXMXFZTGGMKXGOXLNGTMRLBEHFTIMWXEBAXAXAGOFTXAAHTFXPTRLTM

5: MSFESIIESIXSSHSWXHIIVESICMEQITZAXRJQXIHIGCILXRVAVRXMERRIV
PVEWSXSLRMZMXRHISXRKPFEGZIQXYVMXVWIAAEWIXEMVRIFSWEAMRXXHI

6: YSBFASEZAANEHVJHRGEBGJCVGEAVEWRNJTGVUNVCHSARRFZRRGUFAATER
XLFFNUEBQFRBUVUESUGRZRYBRNRUARAURHQVFCFENYYRGABQGVUAQBUDE

Here are some values for the mutual index of coincidence between pairs of columns (columns labels match the labels of the sub-ciphertexts above).

cols 1, 2:	shift 7: 0.0513	shift 20: 0.0673	cols 2, 3:	shift 11: 0.0500	shift 24: 0.0700
cols 2, 4:	shift 21: 0.0532	shift 25: 0.0702	cols 2, 5:	shift 6: 0.0507	shift 10: 0.0707
cols 2, 6:	shift 6: 0.0500	shift 19: 0.0675			

In Exercise 3.52 we decided that u was most likely to be the second character of the key.

- The most likely shift between columns 1 and 2 is 20, so the first character is likely to be a.
- The most likely shift between columns 2 and 3 is 24, so the third character is likely to be s.
- The most likely shift between columns 2 and 4 is 25, so the third character is likely to be t.
- The most likely shift between columns 2 and 5 is 10, so the third character is likely to be e.
- The most likely shift between columns 2 and 6 is 19, so the third character is likely to be n.

The most likely keyword is *austen* and the corresponding plaintext is:

thevillageoflongbournwasonlyonemilefrommerytonamostconvenientdistancefortheyoung
ladieswhowereusuallytemptedthitherthreeorfourtimesaweektopaytheirdutytotheiraunt
andtoamillinersshopjustoverthewaythetwoyoungestofthefamilycatherineandlydiawere
particularlyfrequentintheseattentionstheirmindsweremorevacantthantheirsistersand
whennothingbetterofferedawalktomerytonwasnecessarytoamusetheirmorninghoursand
furnishconversationfortheeveningandhoweverbareofnewsthecountryingeneralmightbethey
alwayscontrivedtolearnsomefromtheirauntatpresentindeedtheywerewellsuppliedbothwith
newsandhappinessbytherecentarrivalofamilitiaregimentintheneighbourhooditwasto
remainthewholewinterandmerytonwastheheadquarters

3.4.3 Playfair cipher

- The Playfair cipher encrypts *pairs* of characters together, effectively creating a much larger alphabet (with digrams).
- It is still a substitution cipher.
- It was invented in 1854 by Charles Wheatstone (but promoted by Baron Playfair).
- It can be chained with other ciphers to increase security.
- The Playfair cipher was used in the Boer War, WW1 and WW2 as a field cipher; and Australian coastwatchers used it in WW2 to rescue J.F.K.

Example 3.55 We construct a 5×5 array using some keyword (with repeated letters deleted) followed by the remaining letters of the alphabet. Let the keyword be CHARLESWHEATSTONEPLAYFAIREXAMPLE.
The corresponding Playfair array would look like

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

To encrypt the message "old cipher", we group the plaintext in pairs of letters (ignoring spaces), and inserting an X if a pair consists of a repeated letter. We also put an X at the end if a single character is left after pairing. Then we encrypt the pairs as follows:

- If the pair are in the same column, then each letter is replaced by the letter directly below it (wrapping around to the top if necessary).
- If the pair are in the same row, then each letter is replaced by the letter directly to the right of it (wrapping around if necessary).
- If the pair are in different rows and columns, then they correspond to the opposite corners of a rectangle and each letter maps to the letter in the same row that forms the other corner of the rectangle.

Hence, ol encrypts to IO, dc encrypts to XR, ip encrypts to NY, he encrypts to CS and rx encrypts to CD. So our ciphertext is IOXRNYCSCD. To decrypt, the same procedure is followed, with below and right replaced by above and left.

Exercise 3.56 Using the Playfair array given above, decrypt the ciphertext HRTVCTRBTC.

By encrypting digrams rather than single letters, the letter frequency profile is flattened, making this quite a good cipher. However, it can be still be cracked using frequency tables for digrams and the fact that if ab maps to $\alpha\beta$ then ba maps to $\beta\alpha$. Double Playfair with carefully selected keywords can get around some of the weaknesses of single Playfair.

3.4.4 The Hill Cipher

A polyalphabetic substitution cipher which encrypts m characters at a time. It was invented in 1929 by Lester S. Hill.

Algorithm 3.57 Hill cipher
Let m be some fixed positive integer. Let $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$ and let

$$\mathcal{K} = \{m \times m \text{ invertible matrices over } \mathbb{Z}_{26}\}.$$

For a key K and $x \in \mathcal{P}$, we define

$$e_K(x) = xK$$

and for $y \in \mathcal{C}$

$$d_K(y) = yK^{-1},$$

where all operations are performed in $\mathbb{Z}_{26}$. □

Exercise 3.58 Let

$$K = \begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix}.$$

Find the ciphertext for the plaintext *bond*.

Remark 3.59 Assume we are working mod 26. Let

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

Then the matrix inverse of K, written K^{-1}, is

$$K = (ad - bc)^{-1} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

where all operations are performed mod 26, and $(ad - bc)^{-1}$ is the multiplicative inverse of $(ad - bc)$ in $\mathbb{Z}_{26}$.

Exercise 3.60 Let

$$K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}.$$

Find the plaintext for the ciphertext *DELW*.

3.4.5 Cryptanalysis of the Hill Cipher

We consider **Known Plaintext** attacks (so the opponent possesses a string of known plaintext x and the corresponding ciphertext y).

Assume that the opponent has determined the value of m being used. Suppose he has at least m distinct pairs of m-tuples, $x_j = (x_{1,j}, x_{2,j}, \ldots, x_{m,j})$ and $y_j = (y_{1,j}, y_{2,j}, \ldots, y_{m,j})$ such that $y_j = e_K(x_j)$, $1 \leqslant j \leqslant m$.

Define two $m \times m$ matrices $X = (x_{i,j})$ and $Y = (y_{i,j})$. Then we have the matrix equation $Y = XK$, where the $m \times m$ matrix K is the unknown key.

So $K = X^{-1}Y$. (If X is not invertible then it will be necessary to try other sets of m plaintext-ciphertext pairs.)

Example 3.61 *Suppose the plaintext* friday *is encrypted using a Hill Cipher with* $m = 2$ *to give the ciphertext* PQCFKU.

So

$$\begin{aligned} e_K(5, 17) &= (15, 16) \\ e_K(8, 3) &= (2, 5) \\ e_K(0, 24) &= (10, 20). \end{aligned}$$

From the first two plaintext-ciphertext pairs we get $Y = XK$, so

$$\begin{bmatrix} 15 & 16 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 5 & 17 \\ 8 & 3 \end{bmatrix} K.$$

So

$$K = \begin{bmatrix} 5 & 17 \\ 8 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 15 & 16 \\ 2 & 5 \end{bmatrix}.$$

Exercise 3.62 Determine K for the above example.

3.5 Permutation Ciphers

Transposition or *permutation* ciphers keep the plaintext characters unchanged, but cipher the message by rearranging the order of the characters.

3.5.1 Scytale cipher

- an early Greek transposition cipher
- strip of paper was wound round a staff
- message written along staff in rows, then paper removed
- leaves a strip of seemingly random letters

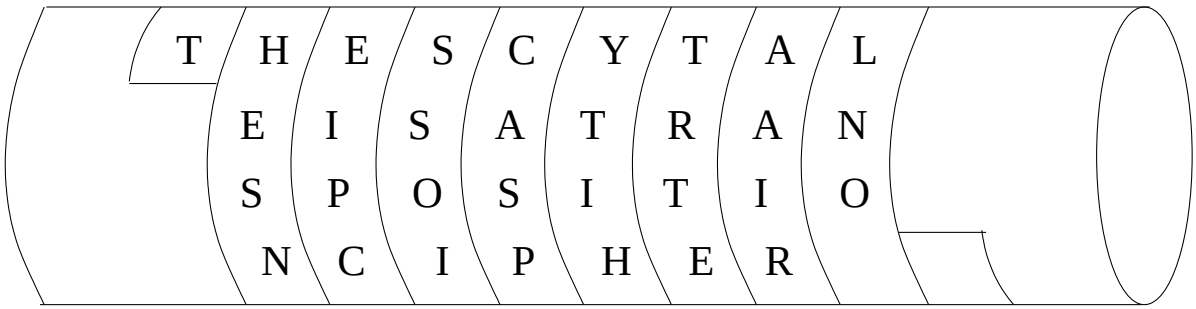


Figure 11: The Scytale cipher

- not very secure: key is width of staff

3.5.2 Reverse cipher

- write the message backwards

```
Plain:   i came i saw i conquered
Cipher:  DEREU QNOCI WASIE MACI
```

3.5.3 Rail Fence cipher

- write plaintext with letters on alternate rows
- read off ciphertext row by row

```
Plain:  i a e s w c n u r d
         c m i a i o q e e
Cipher: IAESW CNURD CMIAI OQEE
```

- can use more than 2 rows

3.5.4 Geometric Figure

- write message following one pattern
- read message out with another
- for example, write in rows, read out in columns

3.5.5 Row transposition ciphers

Algorithm 3.63 General row transposition cipher

Let m be some fixed positive integer.

Let $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$ and let $\mathcal{K}$ consist of all permutations of $\{1, 2, \ldots, m\}$.

For a given key (i.e. a permutation) π, we define

$$e_\pi(x_1, x_2, \ldots, x_m) = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(m)})$$

and

$$d_\pi(y_1, y_2, \ldots, y_m) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \ldots, y_{\pi^{-1}(m)}),$$

where π^{-1} is the inverse permutation to π. □

Note that the key is repeated every m characters, and it may be necessary to pad the plaintext to an even multiple of m.

(Note: can think of the general mixed alphabet as maintaining the positions of characters within text but permuting their values. Here, we are permuting the positions of characters within the text but maintaining their values.)

Theorem 3.64 *Every permutation can be written as a product of disjoint cycles.*

To obtain the inverse of a permutation, reverse the order of each disjoint cycle in the permutation.

Exercise 3.65 Suppose $m = 6$ and the plaintext was encoded with the permutation $\pi = (1\,3)(2\,5\,4\,6)$. Decrypt EESLSHSALSESLSHBLEHSYEETHRAEOS.

3.5.6 Implementation of row transposition ciphers

Here is a simple, practical implementation of a row transposition cipher:

1. Pick a word of length m as the key and write it down.
2. Label the letters in the key with numbers from 1 to m, with the alphabetically first letter labelled 1, the second letter labelled 2, and so on. Each letter (even repeats) must have a different label.
3. Write the labels of the letters in the same order as they occur in the keyword.
4. Write the plaintext in rows under the labels with m characters per row.
5. Reorder labels and columns of text so that the labels are in order from 1 to m.
6. Read ciphertext off one row at a time.
7. Important: do not leave ciphertext in groups matching the size of the key!

Example 3.66 *Encrypt the following plaintext using a row transposition cipher and the key* COMPUTER.

Plaintext: aconvenientwaytoexpressthepermutation

Key	C	O	M	P	U	T	E	R
Key	1	4	3	5	8	7	2	6
Text	a	c	o	n	v	e	n	i
	e	n	t	w	a	y	t	o
	e	x	p	r	e	s	s	t
	h	e	p	e	r	m	u	t
	a	t	i	o	n	z	z	z

When the columns are reordered, the ciphertext becomes:

ANOCNIEVETTNWOYAESPXRTSEHUPEETMRAZITOZZN

Finally, you might like to split ciphertext into groups of size different to m:

ANOCN IEVET TNWOY AESPX RTSEH UPEET MRAZI TOZZN

Exercise 3.67 Determine the permutation that corresponds to the encryption process used in the previous example.

Exercise 3.68 Determine the permutation that corresponds to using a row transposition cipher with the key SORCERY.

Exercise 3.69 Show that a row transposition cipher is a special case of the Hill Cipher.

3.5.7 Cryptanalysis of row transposition ciphers

- A frequency count will show a normal language profile, hence we can guess that letters have been rearranged.
- The basic strategy is to guess the period, then look at all possible permutations in period, and search for common patterns.
- Can use lists of common pairs and triples, and other features.

3.6 Product ciphers

- In general, ciphers based on just substitutions or just transpositions are not secure.
- What about using several ciphers together (so apply one then apply the other)?
  - two substitutions really form only one more complex substitution
  - two transpositions really form only one more complex transposition
  - but a substitution followed by a transposition makes a new, much harder cipher
- Product ciphers consist of substitution-transposition combinations chained together. In general they are far too hard to do by hand, however one famous product cipher, the *ADFGVX* (originally ADFGX) cipher was used in WW1.
- The widespread use of product ciphers had to wait for the invention of cipher technology, particularly the rotor machines (eg Enigma, Hagalin).

3.6.1 ADFGVX cipher

- It was named this because only the letters ADFGVX are used in the ciphertext.
- These letters were chosen since they have very distinct morse codes:

$$A\ \cdot - \qquad D\ - \cdot \cdot \qquad F\ \cdot \cdot - \cdot \qquad G\ - - \cdot \qquad V\ \cdot \cdot \cdot - \qquad X\ - \cdot \cdot -$$

- The cipher uses a fixed substitution table to map each plaintext letter to a letter pair (row-col index) and then uses a keyed block transposition.
- It was used by Germany as a field cipher in WW1.
- It was invented by Nebel and originally used only ADFGX; later V was added so that the alphabet and the digits 0 – 9 could be encoded.

The key consists of a 6×6 array of the characters of the plaintext space (alphabet and digits) and a keyword. The array is used in the substitution step and the keyword in the transposition step.
The substitution step of the encryption involves replacing each character of the plaintext with TWO ciphertext characters (the row label followed by the column label).
The transposition step of the encryption involves writing the keyword (with repeated letters removed) as the heading of columns, then writing the encoded text from the substitution step underneath the "keyword", and finally rearranging the columns so that the column headers are in alphabetical order.

Exercise 3.70 Construct a 6×6 array by placing the 36 characters in the blank cells.

	A	*D*	*F*	*G*	*V*	*X*
A						
D						
F						
G						
V						
X						

Use your array and the keyword *pumpkin* to encrypt the message *red*.

The ADFGX cipher was broken by Painvin, a Frenchman. He observed similar messages with repeated substrings. Reputedly it took only 28 hours to break the new 6-letter version when it was introduced.

3.7 Stream Cipher

- All cryptosystems that we have seen so far are *block ciphers* in the sense that they encrypt blocks of plaintext. Successive plaintext blocks are encrypted using the same key and the key is completely determined before the encryption starts. The resulting ciphertext for any particular bit of the plaintext depends only on the position of the plaintext characters.
- An alternative approach is to use a *stream cipher*. A stream cipher encrypts a stream of plaintext on the fly, that is, one character (or bit) at a time, with the current key calculated by a keystream generator. The current key may depend not only on the initial key fed into the keystream generator, but also on the preceding plaintext. Thus the keystream may not be completely determined before the encryption process starts.
- The basic idea is to generate a keystream $l = l_1 l_2 l_3 \cdots$ and use it to encrypt a plaintext string $x = x_1 x_2 x_3 \cdots$ according to the rule

$$y = y_1 y_2 y_3 \cdots = e_{l_1}(x_1) e_{l_2}(x_2) e_{l_3}(x_3) \cdots .$$

- Definition 3.71 (below) is similar to the definition of a cryptosystem given in Definition 2.3; now we have added $\mathcal{L}$ and $\mathcal{F}$.

Definition 3.71 A *stream cipher* is a *seven-tuple* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{L}, \mathcal{F}, \mathcal{E}, \mathcal{D})$, where the following conditions are satisfied:

1. $\mathcal{P}$ is a finite set of possible *plaintexts*.
2. $\mathcal{C}$ is a finite set of possible *ciphertexts*.
3. $\mathcal{K}$, the *keyspace*, is a finite set of possible (initial) *keys*.
4. $\mathcal{L}$ is a finite set called the *keystream alphabet*.
5. $\mathcal{F} = (f_1, f_2, \ldots)$ is the *keystream generator*. For $i \geqslant 1$, f_i is a function which generates the ith element of the keystream $l_i \in \mathcal{L}$ as a function of the initial key $K \in \mathcal{K}$ and the first $i-1$ plaintext characters.
6. For each $l \in \mathcal{L}$, there is an *encryption rule* $e_l \in \mathcal{E}$ and a corresponding *decryption rule* $d_l \in \mathcal{D}$. Each $e_l : \mathcal{P} \to \mathcal{C}$ and $d_l : \mathcal{C} \to \mathcal{P}$ are functions such that $d_l(e_l(x)) = x$ for every plaintext $x \in \mathcal{P}$.

Remark 3.72 For a stream cipher, with initial key $K \in \mathcal{K}$ and plaintext string $x_1x_2\cdots$, the function f_i is used to generate the keystream element l_i which is then used to encrypt x_i, yielding $y_i = e_{l_i}(x_i)$. To encrypt the plaintext string $x_1x_2x_3\cdots$ we would successively compute

$$l_1, y_1, l_2, y_2, \ldots.$$

Decrypting the cipher string $y_1y_2y_3\ldots$ can be accomplished by successively computing

$$l_1, x_1, l_2, x_2, \ldots.$$

Definition 3.73 The initial key for a stream cipher is often called the *seed*.

Definition 3.74 A stream cipher is ***synchronous*** if the keystream is independent of the plaintext (that is, the next keystream element l_i is generated only as a function of the initial key k and previously generated keystream elements $l_1, l_2, \ldots, l_{i-1}$). A stream cipher is ***asynchronous*** if the keystream is not independent of the plaintext.

Definition 3.75 A stream cipher is ***periodic*** with *period* d if $l_{i+d} = l_i$ for all $i \geq 1$.

Remark 3.76 We can think of block ciphers as special cases of stream ciphers in which the keystream is constant or periodic. For example, the Vigenere cipher with keyword length m is a synchronous, periodic stream cipher with period m.

3.7.1 Autokey cipher

The Autokey cipher is an asynchronous, non-periodic stream cipher. For the Autokey cipher, the key is a value $k \in \mathbb{Z}_{26}$, and the keystream generator sets $l_1 = k$ and generates subsequent keystream elements by shifting the plaintext characters by one position, that is $l_i = x_{i-1}$, where the plaintext is $x_1, x_2, \ldots$. The ciphertext is obtained by adding plaintext and keystream element mod 26.

Algorithm 3.77 Autokey cipher
Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathcal{L} = \mathbb{Z}_{26}$. Let the initial key be $k \in \mathcal{K}$ and the plaintext be $x_1, x_2, \ldots$. Let $l_1 = k$ and $l_i = x_{i-1}$ (for $i \geqslant 2$). For $j \geq 1$ and $l_j, x, y \in \mathbb{Z}_{26}$ define

$$e_{l_j}(x) = x + l_j \mod 26$$

and

$$d_{l_j}(y) = y - l_j \mod 26.$$

□

Example 3.78 *Let $k = 8$. Using the Autokey cipher, encrypt the plaintext* rendezvous.

plaintext:	r	e	n	d	e	z	v	o	u	s
plaintext in $\mathbb{Z}_{26}$:	17	4	13	3	4	25	21	14	20	18
key:	8	17	4	13	3	4	25	21	14	20
ciphertext in $\mathbb{Z}_{26}$:	25	21	17	16	7	3	20	9	8	12
ciphertext:	Z	V	R	Q	H	D	U	J	I	M

Thus the ciphertext is *ZVRQHDUJIM*.

Exercise 3.79 The Autokey cipher has been used to encrypt a message with key 5. Decrypt the ciphertext $MLPWZ$.

3.7.2 LFSR-based stream cipher

Remark 3.80 Stream ciphers are often described in binary arithmetic and used when transmitting data using bits. The plaintext is $x_1, x_2, \ldots$ where each $x_i \in \{0, 1\}$, the keystream is $l_1, l_2, \ldots$ where each $l_i \in \{0, 1\}$, and the ciphertext is $y_1, y_2, \ldots$ where $e_{l_i}(x_i) = y_i = x_i + l_i \pmod 2$. Decryption uses modulo 2 subtraction, which is equivalent to modulo 2 addition, so $d_{l_i}(y_i) = x_i = y_i + l_i \pmod 2$. These operations are equivalent to the bit-wise exclusive or operation (XOR), and so binary arithmetic stream ciphers are easy to implement in both hardware and software. Using the XOR operation for encryption and decryption was originally proposed by Gilbert Vernam in 1917, so stream ciphers that utilise the XOR operation as outlined above are called Vernam ciphers.

Remark 3.81 The security of a stream cipher depends on the keystream $l_1, l_2, \ldots$ appearing to be random. Generating random numbers may seem trivially easy but is in fact quite difficult. Because computers are deterministic, the best you can do with a computer is to use a seed and generate a sequence of *pseudo-random* numbers. Two examples of pseudo-random number generators are linear congruential generators and linear feedback shift registers.

A keystream generated by a linear feedback shift register can be described by a linear recurrence relation. Here is an example of a synchronous keystream generated via a linear recurrence relation of degree 4.

Example 3.82 Let the initial key be (k_1, k_2, k_3, k_4) and let $l_j = k_j$ for $j = 1, 2, 3, 4$. The keystream is generated using the rule

$$l_{i+4} = l_i + l_{i+1} \pmod 2, \text{ for } i \geqslant 1.$$

For example, starting with an initial key of $(1, 0, 0, 0)$, the keystream is

$$1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, \ldots.$$

Here is a more general description of a synchronous keystream generated via a linear recurrence relation of degree m.

Example 3.83 Let the given key (of length m) be $(k_1, k_2, \ldots, k_m)$ and let $l_j = k_j$ for $1 \leqslant j \leqslant m$.

Now we generate the keystream using a linear recurrence relation of degree m. For $i \geqslant 1$:

$$l_{i+m} = \sum_{a=0}^{m-1} c_a l_{i+a} \pmod 2 = c_0 l_i + c_1 l_{i+1} + \cdots + c_{m-1} l_{i+m-1} \pmod 2,$$

where $c_0, c_1, \ldots, c_{m-1} \in \mathbb{Z}_2$ are predetermined constants.

Remark 3.84 A keystream generated by a linear recurrence relation of degree m will repeat as soon as a sequence of m bits in the keystream is repeated. Thus, since the number of distinct sequences of m bits is finite, every keystream generated this way will be periodic. A good linear recurrence relation with a good initial key are ones that make this period as long as possible.

In Example 3.82, if the keystream is initialized with any vector other than $(0, 0, 0, 0)$, then we obtain a keystream of period 15, which is the longest possible.

In a general linear recurrence relation, if the constants $(c_0, c_1, \ldots, c_{m-1})$ are chosen in a suitable way, then any non-zero initialization vector $(k_1, k_2, \ldots, k_m)$ will give rise to a keystream having period $2^m - 1$.

Exercise 3.85 Explain why we need $c_0 = 1$ to generate a useful keystream.

Exercise 3.86 Explain why we need $c_i = 1$ for at least one $i > 0$ to generate a useful keystream.

3.7.3 Cryptanalysis of LFSR-based stream cipher

We consider **known plaintext** attacks. Recall that the ciphertext is the sum modulo 2 of the plaintext and keystream. The keystream is produced from $l_1, l_2, \ldots, l_m$ using the linear recurrence relation

$$l_{m+i} = \sum_{a=0}^{m-1} c_a l_{i+a} \pmod 2 = c_0 l_i + c_1 l_{i+1} + \cdots + c_{m-1} l_{i+m-1} \pmod 2$$

for $i \geqslant 1$, where $c_0, c_1, \ldots, c_{m-1} \in \mathbb{Z}_2$ and $c_0 = 1$.

Example 3.87 Suppose that Oscar knows that the keystream was generated using a 5-stage LFSR, and suppose also that he obtains the ciphertext string

$$101101011110010$$

corresponding to the plaintext string

$$011001111111001.$$

Then he can compute the keystream bits:

$$110100100001011.$$

Using the first 10 bits, he gets the following equations:

$$\begin{array}{llll}
c_0l_1 + c_1l_2 + c_2l_3 + c_3l_4 + c_4l_5 = l_6 & \text{so} & c_0 + c_1 + c_3 = 0 \\
c_0l_2 + c_1l_3 + c_2l_4 + c_3l_5 + c_4l_6 = l_7 & \text{so} & c_0 + c_2 = 1 \\
c_0l_3 + c_1l_4 + c_2l_5 + c_3l_6 + c_4l_7 = l_8 & \text{so} & c_1 + c_4 = 0 \\
c_0l_4 + c_1l_5 + c_2l_6 + c_3l_7 + c_4l_8 = l_9 & \text{so} & c_0 + c_3 = 0 \\
c_0l_5 + c_1l_6 + c_2l_7 + c_3l_8 + c_4l_9 = l_{10} & \text{so} & c_2 = 0
\end{array}$$

These equations can be easily solved by hand, so the linear recurrence relation used was

$$l_{i+5} = l_i + l_i + 3 \pmod 2$$

for $i \geqslant 1$ and the initial key was $(1, 1, 0, 1, 0)$.

Remark 3.88 The above example could have been solved in matrix form as follows. Using the first 10 keystream bits he finds:

$$(0, 1, 0, 0, 0) = (c_0, c_1, c_2, c_3, c_4) \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

This yields

$$(c_0, c_1, c_2, c_3, c_4) = (0, 1, 0, 0, 0) \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} = (1, 0, 0, 1, 0).$$

Thus the recurrence relation used to generate the keystream is

$$l_{i+5} = l_i + l_{i+3} \pmod 2.$$

In general, to carry out a known-plaintext attack on an LFSR-based stream cipher, assume that Oscar knows the value of m. Suppose he has a plaintext string $x_1x_2 \ldots x_n$ and the corresponding ciphertext string $y_1y_2 \ldots y_n$, for some $n \geq 2m$. He can compute the key stream bits $l_i = x_i + y_i \pmod 2$, $1 \leqslant i \leqslant n$. Then Oscar needs only to compute $c_0, c_1, \ldots, c_{m-1}$ in order to be able to reconstruct

the entire keystream. Since $n \geq 2m$, he can write a system of m linear equations in the unknowns $c_0, c_1, \ldots, c_{m-1}$ as follows:

$$(l_{m+1}, l_{m+2}, \ldots, l_{2m}) = (c_0, c_1, \ldots, c_{m-1}) \begin{bmatrix} l_1 & l_2 & \cdots & l_m \\ l_2 & l_3 & \cdots & l_{m+1} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ l_m & l_{m+1} & \cdots & l_{2m-1} \end{bmatrix}.$$

So (noting that the matrix inverse here is in $\mathbb{Z}_2$),

$$(c_0, c_1, \ldots, c_{m-1}) = (l_{m+1}, l_{m+2}, \ldots, l_{2m}) \begin{bmatrix} l_1 & l_2 & \cdots & l_m \\ l_2 & l_3 & \cdots & l_{m+1} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ l_m & l_{m+1} & \cdots & l_{2m-1} \end{bmatrix}^{-1}.$$